M3331 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro™ Family

Directory Information

Please extract the “M3331_Series_BSP_CMSIS_V3.00.002.zip” file firstly, and then put the “M3331_Series_BSP_CMSIS_V3.00.002” folder into the working folder (e.g. .\Nuvoton\BSP Library\).

To experience the powerful features of M3331 in few minutes, please select the sample code of your interest to download and execute on the M3331 board. For example, you can try the TrustZone® technology in M3331 that provides system-wide hardware isolation for trusted software with the sample code in the folder BSP\SampleCode\TrustZone. This folder includes the sample code of TrustZone® for collaborative secure software development, Hard Fault handling and a TrustZone® template. You can open the project files to build them with Keil® MDK, IAR or VSCode, and then download and trace them on the M3331 board to see how the TrustZone® works.

This BSP folder contents:

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

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

1 DOCUMENT 4

2 LIBRARY 5

3 SAMPLE CODE 6

4 THIRDPARTY 7

5 SAMPLECODE\ISP 8

6 SAMPLECODE\NUMAKER-M3334KIV2 9

7 SAMPLECODE\POWERMANAGEMENT 10

8 SAMPLECODE\STDDRIVER 11

System Manager (SYS) 11

Clock Controller (CLK) 11

Flash Memory Controller (FMC) 11

General Purpose I/O (GPIO) 12

PDMA Controller (PDMA) 12

Timer Controller (TIMER) 13

Watchdog Timer (WDT) 14

Window Watchdog Timer (WWDT) 14

Real Timer Clock (RTC) 14

Basic PWM Generator and Capture Timer (BPWM) 14

Enhance PWM Generator and Capture Timer (EPWM) 14

Enhanced Quadrature Encoder Interface (EQEI) 15

UART Interface Controller (UART) 15

I^2S Controller (I^2S) 16

I^3S Controller (I^3S) 16

Serial Peripheral Interface (SPI) 16

Quad Serial Peripheral Interface (QSPI) 17

I^2C Serial Interface Controller (I^2C) 18

Universal Serial Control Interface Controller - UART Mode (USCI-UART) 18

Universal Serial Control Interface Controller - SPI Mode (USCI-SPI) 18

Universal Serial Control Interface Controller - I²C Mode (USCI-I2C) 19
External Bus Interface (EBI) 20
High Speed USB 2.0 Device Controller (HSUSBD) 20
USB 1.1/2.0 Host Controller (HSUSBH) 21
High Speed USB On-The-Go (HSOTG) 22
CRC Controller (CRC) 22
Enhance 12-bit Analog-to-Digital Converter (EADC) 22
Analog Comparator Controller (ACMP) 23
Controller Area Network (CAN) 23
Enhanced Input Capture Timer (ECAP) 24
Enhanced LED Light Strip Interface (ELLSI) 24
LED Light Strip Interface (LLSI) 24
Secure Digital Host Controller (SDH) 24
Parallel Dimming Communication Interface (PDCI) 25

9 SAMPLECODE\TEMPLATE 26

10 SAMPLECODE\TRUSTZONE 27

11 SAMPLECODE\XOM 28

1 Document

CMSIS.html	Introduction of CMSIS version 5.9.0 CMSIS components included CMSIS-CORE, CMSIS-Driver, CMSIS-DSP, etc. ● CMSIS-CORE: API for the Cortex-M0 processor core and peripherals. ● CMSIS-Driver: Defines generic peripheral driver interfaces for middleware making it reusable across supported devices. ● CMSIS-DSP: DSP Library Collection with over 60 Functions for various data types: fix-point (fractional q7, q15, q31) and single precision floating-point (32-bit).
Revision History.pdf	The revision history of M3331 Series BSP.
NuMicro M3331 Series Driver Reference Guide.chm	The usage of drivers in M3331 Series BSP.

2 Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V5.9.0 definitions by ARM® Corp.
CMSIS-DSP	CMSIS DSP Library.
Device	CMSIS compliant device header file.
LIsiYcableLib	LLSI library source code.
NuMaker	Specific libraries for M3331 NuMaker board.
StdDriver	All peripheral driver header and source files.
UsbHostLib	USB host library source code.

3 Sample Code

FreeRTOS	Simple FreeRTOS™ demo code.
Hard_Fault_Sample	Show hard fault information when hard fault happened. The hard fault handler show some information included program counter, which is the address where the processor was executing when the hard fault occur. The listing file (or map file) can show what function and instruction that was. It also shows the Link Register (LR), which contains the return address of the last function call. It can show the status where CPU comes from to get to this point.
ISP	Sample codes for In-System-Programming.
NuMaker-M3334KI	Sample codes for NuMaker-PFM-M3334 board.
NuMaker-M3334KIv2	Sample codes for NuMaker-PFM-M3334KI_V2.0 board.
PowerManagement	Power management sample code.
StdDriver	Demonstrate the usage of M3331 series MCU peripheral driver APIs.
Template	A project template for M3331.
TrustZone	Includes the demo of secure codes and non-secure codes.
XOM	Demonstrate how to create XOM library and use it.

4 ThirdParty

FatFs	An open source FAT/exFAT file system library. A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html.
FreeRTOS	A real time operating system available for free download. Its official website is: http://www.freertos.org/.
LibMAD	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 (http://www.underbit.com/)for the commercial license.

5 SampleCode\ISP

ISP_CAN	In-System-Programming Sample code through CAN interface.
ISP_DFU_20	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
ISP_HID_20	In-System-Programming Sample code through USB HID interface.
ISP_I2C	In-System-Programming Sample code through I^2C interface.
ISP_RS485	In-System-Programming Sample code through RS485 interface.
ISP_SPI	In-System-Programming Sample code through SPI interface.
ISP_UART	In-System-Programming Sample code through UART interface.

6 SampleCode\NuMaker-M3334KIv2

Blinky	Use the timer periodic mode to generate timer interrupt every 1 second.
emWin_GIFDecoder	To utilize emWin library to demonstrate interactive feature.
LVGL_WidgetsDemo	To utilize LVGL v9.4.0 library to demonstrate built-in widgets and material theme.
MP3_Player	MP3 player sample plays MP3 files stored on SD memory card.
NuvotonAudioGUI_Firmware	NAU88C22 configuration tool main program.

7 SampleCode\PowerManagement

SYS_DPDMode_Wakeup	Show how to wake up system form DPD Power-down mode by Wake-up pin(PC.0), Wake-up Timer, RTC Tick, RTC Alarm or RTC Tamper 0.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.
SYS_PowerDownMode	Show how to enter to different Power-down mode and wake-up by RTC.
SYS_PowerMode	Show how to set different core voltage and main voltage regulator type.
SYS_SPDMode_Wakeup	Show how to wake up system form SPD Power-down mode by GPIO pin(PC.0), Wake-up Timer, Wake-up ACMP, RTC Tick, RTC Alarm, RTC Tamper 0, BOD or LVR.
SYS_SPDMode_WakeupAndReturn	Show how to continue executing code after wake-up form SPD Power-down mode by SRAM data retention function.
SYS_SPDMode_WakeupVTOR	Show how to continue executing code after wake-up form SPD Power-down mode by VTOR function.

8 SampleCode\StdDriver

System Manager (SYS)

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

Clock Controller (CLK)

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

Flash Memory Controller (FMC)

FMC_APPROT	Demonstrate how to use FMC APROM Protect function.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP Command to calculate the CRC32 checksum of APROM and LDROM.
FMC_DualBankFwUpgrade	Implement a firmware update mechanism based on dual bank flash architecture.
FMC_ECC	Demonstrated FMC ECC status and ECC error fault address.
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash (support KEIL MDK only).
FMC_FwUpgradeApplication	Bank remap sample code.
FMC_IAP	Show how to call LDROM function from APROM.
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_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify APROM/LDROM pages are all 0xFFFFFFFF or not.

FMC_RW	Demonstrate how to read/program embedded Flash by ISP function.
FMC_XOM	This sample code shows how to configure and setup an XOM region the perform XOM function.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_EINTTriggerEPWM	Show the usage of GPIO EINT trigger EPWM function.
GPIO_EINTTriggerPDMA	Show the usage of GPIO EINT trigger PDMA function.
GPIO_EINTTriggerTPWM	Show the usage of GPIO EINT trigger Timer PWM 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.

PDMA Controller (PDMA)

PDMA_BasicMode	Use PDMA0 Channel 2 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA0 channel 4 to transfer data from memory to memory by scatter-gather mode.
PDMA_ScatterGather_PingPongBuffer	Use PDMA0 to implement Ping-Pong buffer by scatter-gather mode (memory to memory).
PDMA_StrideRepeat	Use PDMA0 channel 0 to transfer data from memory to memory with stride and repeat.
PDMA_TimeOut	Demonstrate PDMA0 channel 1 get/clear timeout flag with UART1.

Timer Controller (TIMER)

TIMER_ACMPTrigger	Use ACMP to trigger Timer0 counter 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	Demonstrates the timer event counter function.
TIMER_FreeCountingMode	Use the timer0 pin PA.11 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the timer pin PB.5 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_AccumulatorINTStopMode	Demonstrate TIMER PWM accumulator interrupt to stop counting.
TIMER_PWM_AccumulatorINTTriggerPDMA	Demonstrate TIMER PWM accumulator interrupt to trigger PDMA transfer.
TIMER_PWM_Brake	Demonstrate how to use Timer0 PWM brake function.
TIMER_PWM_ChangeDuty	Change duty cycle and period of output waveform in PWM down count type.
TIMER_PWM_DeadTime	Demonstrate Timer PWM Complementary mode and Dead-Time function.
TIMER_PWM_OutputWaveform	Demonstrate output different duty waveform in Timer0~Timer5 PWM.
TIMER_TimeoutWakeup	Use timer to wake up system from Power-down mode periodically
TIMER_ToggleOut	Demonstrate the timer0 toggle out function on pin PB.5.

Watchdog Timer (WDT)

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

Window Watchdog Timer (WWDT)

WWDT_CompareINT	Show how to reload the WWDT counter value.

Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Use RTC alarm interrupt event to wake up system.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.

Basic PWM Generator and Capture Timer (BPWM)

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

Enhance PWM Generator and Capture Timer (EPWM)

EPWM_AccumulatorINT_TriggerPDMA	Demonstrate EPWM accumulator interrupt trigger PDMA.
EPWM_AccumulatorStopMo	Demonstrate EPWM accumulator stop mode.

de	
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 Zone function.
EPWM_DoubleBuffer	Change duty cycle and period of output waveform by EPWM Double Buffer function (Period loading mode).
EPWM_OutputWaveform	Demonstrate how to use EPWM 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 EPWM counter synchronous start function.

Enhanced Quadrature Encoder Interface (EQEI)

EQEI_CompareMatch	Show the usage of EQEI 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 data in UART IrDA mode.
UART_LIN	Transmit LIN frame including header and response in UART LIN mode.
UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_SingleWire	Transmit and receive data by UART Single-Wire 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.

I²S Controller (I²S)

I2S_Codec	This is an I²S demo using NAU8822/88L25 audio codec, and used to play back the input from line-in.
I2S_Codec_PDMA	This is an I²S demo with PDMA function connected with codec.
I2S_MP3PLAYER	MP3 player sample plays MP3 files stored on SD memory card.
I2S_WAVPLAYER	This is a WAV file player which plays back WAV file stored in SD memory card.

I³S Controller (I³S)

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

Serial Peripheral Interface (SPI)

SPI_Flash	Access SPI flash through SPI interface.
SPI_HalfDuplex	Demonstrate SPI half-duplex mode. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both SPI0 and SPI1 will be configured as half-duplex mode.
SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect MISO pin and MOSI pin together. It will compare the received data with transmitted data.
SPI_MasterFIFOMode	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.
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 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.
SPII2S_Master	Configure SPI0 as I²S Master mode and demonstrate how I²S works in Master mode. This sample code needs to work with SPII2S_Slave sample code.
SPII2S_PDMA_Codec	This is an I²S demo with PDMA function connected with audio codec.
SPII2S_PDMA_Play	This is an I²S demo for playing data and demonstrate how I2S works with PDMA.
SPII2S_PDMA_PlayRecord	This is an I²S demo for playing and recording data with PDMA function.
SPII2S_PDMA_Record	This is an I²S demo for recording data and demonstrate how I2S works with PDMA.
SPII2S_Slave	Configure SPI0 as I²S Slave mode and demonstrate how I²S works in Slave mode. This sample code needs to work with SPII2S_Master sample code.

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	Configure QSPI0 as Slave 3 wire mode and demonstrate how tocommunicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with SPI_MasterFIFOMode sample code.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM through a I²C interface
I2C_Loopback	Demonstrate how a Master accesses Slave.
I2C_Master	Demonstrate how a Master accesses Slave. This sample code needs to work with I2C_Slave sample code.
I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with I2C_Slave sample code.
I2C_PDMA_TRX	Demonstrate I2C PDMA mode and need to connect I2C0(Master) and I2C1(Slave).
I2C_Slave	Demonstrate how to set I²C in slave mode to receive the data from a Master. This sample code needs to work with I2C_Master sample code.
I2C_SMBus	Demonstrate how to control SMBus interface and use SMBus protocol between Host and Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode through I2C interface. This sample code needs to work with I2C_Master sample code.

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

USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function.
USCI_UART_Autoflow	Transmit and receive data using auto flow control.
USCI_UART_PDMA	Transmit and receive UART data with PDMA.
USCI_UART_RS485	Transmit and receive data in RS485 mode.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through a RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.

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

USCI_SPI_Loopback	Implement USCI_SPI1 Master loop back transfer. This

	sample code needs to connect USCI_SPI1_MISO pin and USCI_SPI1_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI1 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. This sample code needs to work with USCI_SPI_SlaveMode sample code.
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.

Universal Serial Control Interface Controller - I²C Mode (USCI-I2C)

USCI_I2C_EEPROM	Demonstrate how to access EEPROM through a USCI_I2C interface.
USCI_I2C_Master	Demonstrate how a Master access Slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Demonstrate how a Master use 10-bit addressing access Slave. This sample code needs to work with USCI_I2C_Slave_10bit sample code.
USCI_I2C_Monitor	Demonstrate how USCI_I2C Monitors transmission between I2C Master and I2C 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 sample code.
USCI_I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Slave	Demonstrate how to set I²C in slave mode to receive the data from a Master. This sample code needs to work with USCI_I2C_Master sample code.
USCI_I2C_Slave_10bit	Demonstrate how to set I²C in 10-bit addressing slave mode to receive the data from a Master. This sample code needs to work withUSCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Slave	Demonstrate how to set I²C to wake up MCU from Power-

	down mode. This sample code needs to work USCI_I2C_Master sample code.

External Bus Interface (EBI)

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

High Speed USB 2.0 Device Controller (HSUSBD)

HSUSBD_Audio10_Codec	An UAC1.0 sample used to record and play the sound sent from PC through the USB interface.
HSUSBD_Audio10_Headset	An UAC1.0 sample and used to plays the sound send 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_Audio20_Headset	An UAC2.0 sample used to play the sound sent from PC through the USB interface.
HSUSBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device.
HSUSBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
HSUSBD_HID_Mouse_BC12	Demonstrate how to implement a USB mouse device with BC1.2 (Battery Charging) which shows different type of charging port after connected USB port. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
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_HID_Transfer_And_MSC	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.
HSUSBD_Mass_Storage_	USB Mass Storage Device CD-ROM Emulation.

CDROM	
HSUSBD_Mass_Storage_Flash	Use embedded Data Flash as storage to implement a USB Mass-Storage device.
HSUSBD_Mass_Storage SactterGather	Demonstrate the usage of USBD DMA scatter gather function.
HSUSBD_Mass_Storage_SD	Implement a SD card reader.
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_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual com port device.
HSUSBD_VENDOR_LBK	Implement a proprietary Vendor LBK device. This sample requires a M3331 USB host running sample HSUSBD_USBH_VENDOR_LBK to be connected.

USB 1.1/2.0 Host Controller (HSUSBH)

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_HIDMouse_ Keyboard	Demonstrates how to support USB mouse and keyboard input.
HSUSBH_USBH_MassStora	Use a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it work as a disk driver

ge	under the FATFS file system.
HSUSBH_USBH_UAC_HID	Show how to use USBH Audio Class driver and HID driver at the same time. The target device is a Game Audio (UAC+HID composite device).
HSUSBH_USBH_UAC_ LoopBack	Receive audio data from an UAC device, and immediately send back to the UAC device.
HSUSBH_USBH_VCOM	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
HSUSBH_USBH_VENDOR_ LBK	Show how to do transfer on a known device with a vendor driver. This sample requires a M3331 USB device running sample HSUSBD_VENDOR_LBK to be connected.

High Speed USB On-The-Go (HSOTG)

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

CRC Controller (CRC)

CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Implement CRC in CRC-8 mode and get the CRC checksum result.
CRC_CRC32	Implement CRC in CRC-32 mode with PDMA transfer.

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

EADC_Accumulate	Demonstrate how to get accumulate conversion result.
EADC_ADINT_Trigger	Use ADINT interrupt to do the EADC continuous scan conversion.
EADC_Average	Demonstrate how to get average conversion result.
EADC_BandGap	Convert Band-gap (Sample module 16) and print conversion result.

EADC_PDMA_EPWM_Trigger	Demonstrate how to trigger EADC by EPWM and transfer conversion data by PDMA.
EADC_EPWM_Trigger	Demonstrate how to trigger EADC by EPWM.
EADC_Pending_Priority	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Trigger EADC by writing EADC_SWTRG register.
EADC_TempSensor	Convert temperature sensor (Sample module 17) and print conversion result.
EADC_Timer_Trigger	Show how to trigger EADC by timer.

Analog Comparator Controller (ACMP)

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

Controller Area Network (CAN)

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

CANFD_CANFD_TxRxINT	An example of interrupt control using CAN FD bus communication.

Enhanced Input Capture Timer (ECAP)

ECAP_GetInputFreq	Show how to use ECAP interface to get input frequency
ECAP_GetQEIFreq	Show how to use ECAP interface to get QEI frequency.

Enhanced LED Light Strip Interface (ELLSI)

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

LED Light Strip Interface (LLSI)

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

Secure Digital Host Controller (SDH)

SDH_FATFS	Access a SD card formatted in FAT file system.

Parallel Dimming Communication Interface (PDCI)

PDCI_Output	Demonstrate PDCI Mode0 data transfer.
PDCI_PDMA_Output	Demonstrate PDCI Mode0 data transfer with PDMA.

9 SampleCode\Template

Template	A project template for M3331 MCU.

10 SampleCode\TrustZone

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

11 SampleCode\XOM

XOMLib	Demonstrate how to create XOM library.
XOMLibDemo	Demonstrate how to use XOMLib.

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.