

# M4521 Series CMSIS 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.

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

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

*All data and specifications are subject to change without notice.*

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

[www.nuvoton.com](http://www.nuvoton.com)

## TABLE OF CONTENTS

<b>1</b>	<b>DOCUMENT.....</b>	<b>4</b>
<b>2</b>	<b>LIBRARY .....</b>	<b>5</b>
<b>3</b>	<b>SAMPLECODE .....</b>	<b>6</b>
<b>4</b>	<b>SAMPLECODE\CORTEXM4.....</b>	<b>7</b>
<b>5</b>	<b>SAMPLECODE\ISP .....</b>	<b>8</b>
<b>6</b>	<b>SAMPLECODE\REGBASED .....</b>	<b>9</b>
	System Manager (SYS) .....	9
	Clock Controller (CLK).....	9
	Flash Memory Controller (FMC).....	9
	General Purpose I/O (GPIO).....	9
	PDMA Controller (PDMA) .....	10
	Timer Controller (TIMER).....	10
	Watchdog Timer (WDT) .....	10
	Window Watchdog Timer (WWDT) .....	10
	Real Timer Clock (RTC) .....	10
	PWM Generator and Capture Timer (PWM).....	11
	UART Interface Controller (UART).....	11
	Smart Card Host Interface (SC).....	12
	Serial Peripheral Interface (SPI) .....	12
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	12
	I <sup>2</sup> S Controller (I <sup>2</sup> S).....	13
	External Bus Interface (EBI) .....	13
	CRC Controller (CRC) .....	13
	Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	14
<b>7</b>	<b>SAMPLECODE\STDDRIVER.....</b>	<b>15</b>
	System Manager (SYS) .....	15
	Clock Controller (CLK).....	15

Flash Memory Controller (FMC) .....	15
General Purpose I/O (GPIO) .....	15
PDMA Controller (PDMA) .....	16
Timer Controller (TIMER) .....	16
Watchdog Timer (WDT) .....	16
Window Watchdog Timer (WWDT) .....	16
Real Timer Clock (RTC) .....	17
PWM Generator and Capture Timer (PWM) .....	17
UART Interface Controller (UART) .....	17
Smart Card Host Interface (SC) .....	18
Serial Peripheral Interface (SPI) .....	18
I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	18
I <sup>2</sup> S Controller (I <sup>2</sup> S) .....	19
USB Device Controller (USBD) .....	19
USB 1.1 Host Controller (USBH) .....	21
External Bus Interface (EBI) .....	21
CRC Controller (CRC) .....	22
Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	22

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 4.5.0
<b>NuMicro M4521 BSP Revision History.pdf</b>	This document shows the revision history of M4521 BSP.
<b>NuMicro M4521 Driver Reference Guide.html</b>	This document describes the usage of drivers in M4521 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>	Smart card 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>CardReader</b>	CCID <sup>[1]</sup> smart card reader sample code.
<b>CortexM4</b>	Cortex®-M4 sample code.
<b>FreeRTOS</b>	Simple FreeRTOS™ demo code.
<b>ISP</b>	Sample codes for In-System-Programming.
<b>Hard_Fault_Sample</b>	<p>Show hard fault information when hard fault happened.</p> <p>The hard fault handler show some information included program counter, which is the address where the processor was executing when the hard fault occur. The listing file (or map file) can show what function and instruction that was.</p> <p>It also shows the Link Register (LR), which contains the return address of the last function call. It can show the status where CPU comes from to get to this point.</p>
<b>RegBased</b>	The sample code able to access control registers directly.
<b>Semihost</b>	Show how to print and get character through IDE console window.
<b>StdDriver</b>	Sample code to demonstrate the usage of M4521 series MCU peripheral driver APIs.
<b>Template</b>	A project template for M4521 series MCU.

1. Circuit card interface device (CCID) is USB device that interface with integrated circuit cards.

## 4 SampleCode\CortexM4

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

## 5 SampleCode\ISP

<b>ISP_CAN</b>	In-System-Programming sample code through an I <sup>2</sup> C interface.
<b>ISP_DFU</b>	In-System-Programming sample code through USB DFU( Device Firmware Upgrade) class.
<b>ISP_HID</b>	In-System-Programming sample code through a USB 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.



## 6 SampleCode\RegBased

### 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 CLK0 pin.

### 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_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 or LDROM by VECMAP.
<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write functions.

### General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and debounce 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 and output control.

<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.
-----------------------	--

## PDMA Controller (PDMA)

<b>PDMA</b>	Use PDMA channel 2 to transfer data from memory to memory.
<b>PDMA_ScatterGather</b>	Use PDMA channel 5 to transfer data from memory to memory by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

## Timer Controller (TIMER)

<b>TIMER_CaptureCounter</b>	Show how to use the Timer2 capture function to capture Timer2 counter value.
<b>TIMER_EventCounter</b>	Implement Timer1 event counter function to count the external input event.
<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_TimeoutWakeup</b>	Use Timer0 periodic time-out interrupt event to wake up system.

## Watchdog Timer (WDT)

<b>WDT_TimeoutWakeup AndReset</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_AlarmWakeup</b>	Use RTC alarm interrupt event to wake up system.
------------------------	--

<b>RTC_SpareRegisterRW</b>	Show how to access RTC spare registers in supported chip.
<b>RTC_TimeAndTick</b>	Demonstrate the RTC function and display current time to the UART console.

## PWM Generator and Capture Timer (PWM)

<b>PWM_Capture</b>	Capture the PWM1 channel 0 waveform by PWM1 channel 2.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM dead-time insertion function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM double buffer function.
<b>PWM_PDMA_Capture</b>	Capture the PWM1 channel 0 waveform by PWM1 channel 2, and use PDMA to transfer captured data.

## UART Interface Controller (UART)

<b>UART_AutoBaudRate_Master</b>	Show how to use auto baud rate detection function. This sample code needs to work with <a href="#">UART_AutoBaudRate_Slave</a> .
<b>UART_AutoBaudRate_Slave</b>	Show how to use auto baud rate detection function. This sample code needs to work with <a href="#">UART_AutoBaudRate_Master</a> .
<b>UART_AutoFlow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">UART_Autoflow_Slave</a> .
<b>UART_AutoFlow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">UART_Autoflow_Master</a> .
<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with <a href="#">UART_IrDA_Slave</a> .
<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with <a href="#">UART_IrDA_Master</a> .
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.
<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with <a href="#">UART_RS485_Slave</a> .

<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with <a href="#">UART_RS485_Master</a> .
<b>UART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.

### Smart Card Host Interface (SC)

<b>SC_ReadATR</b>	Read the smartcard ATR from Smartcard interface.
<b>SCUART_TxRx</b>	Demonstrate Smartcard UART mode by connecting PA.0 and PA.1 pins.

### Serial Peripheral Interface (SPI)

<b>SPI_Loopback</b>	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
<b>SPI_MasterMode</b>	Configure SPI0 as master mode and demonstrate how to communicate with an off-chip SPI slave device with FIFO mode. This sample code could work with <a href="#">SPI_SlaveMode</a> sample code.
<b>SPI_PDMA_LoopTest</b>	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as slave mode and QSPI0 will be configured as master mode. Both Tx PDMA function and Rx PDMA function will be enabled.
<b>SPI_SlaveMode</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_MasterMode</a> sample code.

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

<b>I2C_EEPROM</b>	Read and write EEPROM via I <sup>2</sup> C interface.
<b>I2C_GCMode_Master</b>	Demonstrate how a master uses I <sup>2</sup> C address 0x0 to write data to I <sup>2</sup> C slave. This sample code needs to work with <a href="#">I2C_GCMode_Slave</a> sample code.

<b>I2C_GCMode_Slave</b>	Demonstrate how to receive master data in GC (General Call) mode. This sample code needs to work with <a href="#">I2C_GCMode_Master</a> sample code.
<b>I2C_Master</b>	An I <sup>2</sup> C master mode demo code. This sample code needs to work with <a href="#">I2C_Slave</a> sample code.
<b>I2C_Slave</b>	An I <sup>2</sup> C slave mode demo code. This sample code needs to work with <a href="#">I2C_Master</a> sample code.
<b>I2C_SMBus</b>	Show how to control SMBus interface and use SMBus protocol between host and slave.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with <a href="#">I2C_Wakeup_Slave</a> .
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with <a href="#">I2C_Wakeup_Master</a> .

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

<b>I2S_Master</b>	Configure SPI1 as I <sup>2</sup> S Master mode and demonstrate how I <sup>2</sup> S works in Master mode. This sample code needs to work with <a href="#">I2S_Slave</a> sample code.
<b>I2S_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="#">I2S_Master</a> sample code.

## External Bus Interface (EBI)

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

## 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.

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

<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to trigger the EADC conversion.
<b>EADC_PDMA_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
<b>EADC_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM.
<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_Timer_Trigger</b>	Show how to trigger EADC by Timer.

## 7 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 CLK0 pin.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.

### 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_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 or LDROM by VECMAP.
<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write functions.

### General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and debounce 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 and output control.
<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.

## PDMA Controller (PDMA)

<b>PDMA</b>	Use PDMA channel 2 to transfer data from memory to memory.
<b>PDMA_ScatterGather</b>	Use PDMA channel 5 to transfer data from memory to memory by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

## Timer Controller (TIMER)

<b>TIMER_CaptureCounter</b>	Show how to use the Timer2 capture function to capture Timer2 counter value.
<b>TIMER_Delay</b>	Show how to use Timer0 to create various delay time.
<b>TIMER_EventCounter</b>	Implement Timer1 event counter function to count the external input event.
<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_TimeoutWakeup</b>	Use Timer0 periodic time-out interrupt event to wake up system.

## Watchdog Timer (WDT)

<b>WDT_TimeoutWakeup AndReset</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_AlarmWakeup</b>	Use RTC alarm interrupt event to wake up system.
<b>RTC_SpareRegisterRW</b>	Show how to access RTC spare registers in supported chip.
<b>RTC_TimeAndTick</b>	Demonstrate the RTC function and display current time to the UART console.

## PWM Generator and Capture Timer (PWM)

<b>PWM_Capture</b>	Capture the PWM1 channel 0 waveform by PWM1 channel 2.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM dead-time insertion function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM double buffer function.
<b>PWM_PDMA_Capture</b>	Capture the PWM1 channel 0 waveform by PWM1 channel 2, and use PDMA to transfer captured data.

## UART Interface Controller (UART)

<b>UART_AutoBaudRate_Master</b>	Show how to use auto baud rate detection function. This sample code needs to work with <a href="#">UART_AutoBaudRate_Slave</a> .
<b>UART_AutoBaudRate_Slave</b>	Show how to use auto baud rate detection function. This sample code needs to work with <a href="#">UART_AutoBaudRate_Master</a> .
<b>UART_AutoFlow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">UART_Autoflow_Slave</a> .
<b>UART_AutoFlow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">UART_Autoflow_Master</a> .
<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with <a href="#">UART_IrDA_Slave</a> .
<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with <a href="#">UART_IrDA_Master</a> .
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.

<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with <a href="#">UART_RS485_Slave</a> .
<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with <a href="#">UART_RS485_Master</a> .
<b>UART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.

### Smart Card Host Interface (SC)

<b>SC_ReadATR</b>	Read the smartcard ATR from Smartcard interface.
<b>SCUART_TxRx</b>	Demonstrate Smartcard UART mode by connecting PA.0 and PA.1 pins.

### Serial Peripheral Interface (SPI)

<b>SPI_Loopback</b>	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
<b>SPI_MasterMode</b>	Configure SPI0 as master mode and demonstrate how to communicate with an off-chip SPI slave device with FIFO mode. This sample code could work with <a href="#">SPI_SlaveMode</a> sample code.
<b>SPI_PDMA_LoopTest</b>	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as slave mode and QSPI0 will be configured as master mode. Both Tx PDMA function and Rx PDMA function will be enabled.
<b>SPI_SlaveMode</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_MasterMode</a> sample code.

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

<b>I2C_EEPROM</b>	Read and write EEPROM via I <sup>2</sup> C interface.
-------------------	---

<b>I2C_GCMode_Master</b>	Demonstrate how a master uses I <sup>2</sup> C address 0x0 to write data to I <sup>2</sup> C slave. This sample code needs to work with <a href="#">I2C_GCMode_Slave</a> sample code.
<b>I2C_GCMode_Slave</b>	Demonstrate how to receive master data in GC (General Call) mode. This sample code needs to work with <a href="#">I2C_GCMode_Master</a> sample code.
<b>I2C_Master</b>	An I <sup>2</sup> C master mode demo code. This sample code needs to work with <a href="#">I2C_Slave</a> sample code.
<b>I2C_Slave</b>	An I <sup>2</sup> C slave mode demo code. This sample code needs to work with <a href="#">I2C_Master</a> sample code.
<b>I2C_SMBus</b>	Show how to control SMBus interface and use SMBus protocol between host and slave.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with <a href="#">I2C_Wakeup_Slave</a> .
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with <a href="#">I2C_Wakeup_Master</a> .

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

<b>I2S_Master</b>	Configure SPI1 as I <sup>2</sup> S Master mode and demonstrate how I <sup>2</sup> S works in Master mode. This sample code needs to work with <a href="#">I2S_Slave</a> sample code.
<b>I2S_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="#">I2S_Master</a> sample code.

## USB Device Controller (USB D)

<b>USB D_Audio_HID_NAU8822</b>	Implement a USB audio class device with HID key. NAU8822 is used in this sample code to play the audio data from Host. It also supports to record data from NAU8822 to Host.
<b>USB D_Audio_NAU8822</b>	Demonstrate how to implement a USB audio class device. NAU8822 is used in this sample code to play the audio data from

	Host. It also supports to record data from NAU8822 to Host.
<b>USBD_HID_Keyboard</b>	Show how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
<b>USBD_HID_Mouse</b>	Show how to implement a USB mouse device. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
<b>USBD_HID_MouseKeyboard</b>	Demonstrate how to implement a USB mouse function and a USB keyboard on the same USB device. The mouse cursor will move automatically when this mouse device connecting to PC. This sample code uses a GPIO to simulate key input.
<b>USBD_HID_Transfer</b>	Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with USB device.
<b>USBD_HID_Transfer_and_Keyboard</b>	Demonstrate how to implement a composite device (HID Transfer and keyboard). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with USB device.
<b>USBD_HID_Transfer_and_MSC</b>	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_MassStorage_CDROM</b>	Demonstrate how to simulate a USB CD-ROM device.
<b>USBD_MassStorage_DataFlash</b>	Use embedded data flash as storage to implement a USB Mass-Storage device.
<b>USBD_Micro_Printer</b>	Show 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 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>	Implement a USB composite device with virtual COM port and keyboard functions.
<b>USBD_VCOM_and_</b>	Demonstrate how to implement a composite device (VCOM and

<b>HID_Transfer</b>	HID Transfer). It supports one virtual COM port and transfers data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_VCOM_and_MassStorage</b>	Implement a USB composite device. It supports one virtual COM port and one USB Mass-Storage device.
<b>USBD_VCOM_DualPort</b>	Demonstrate how to implement a USB dual virtual COM port device.
<b>USBD_VCOM_SinglePort</b>	Implement a USB virtual COM port device. It supports one virtual COM port.

## USB 1.1 Host Controller (USBH)

<b>USBH_AOA</b>	An Android Open Accessory (AOA) device sample.
<b>USBH_Audio_Class</b>	Show how to implement a USB Host and recognize a complex of audio (speaker, microphone) device.
<b>USBH_HID</b>	Show how to implement a USB Host and recognize a HID device when device plug-in.
<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_MultiDevice</b>	Show how to implement a USB Host and recognize multi-HID devices when devices plug-in.
<b>USBH_UAC_HID</b>	A USB Host sample code to support USB Audio Class with HID composite device.
<b>USBH_UMAS</b>	Show how to implement a USB Host with a file system to read/write a file on USB Mass Storage.
<b>USBH_UMAS_FileRW</b>	Show how to implement a USB Host with a file system to read/write a file on USB Mass Storage.

## External Bus Interface (EBI)

<b>EBI_NOR</b>	Configure EBI interface to access NOR Flash connected on EBI interface.
----------------	---

<b>EBI_SRAM</b>	Configure EBI interface to access SRAM connected on EBI interface.
-----------------	--

### **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.

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

<b>EADC_Accumulate</b>	Demonstrate how to get accumulated conversion result.
<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to trigger the EADC conversion.
<b>EADC_PDMA_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM 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_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM.
<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_Timer_Trigger</b>	Show how to trigger EADC by Timer.

### **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.*