

## NUC505 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>	Third party source code, including FatFs, LibMAD, 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](http://www.nuvoton.com)

**TABLE OF CONTENTS**

1 DOCUMENT.....3

2 LIBRARY .....4

3 SAMPLECODE .....5

4 SAMPLECODE\BOOTTEMPLATE.....6

5 SAMPLECODE\ISP\_SAMPLE .....7

6 SAMPLECODE\CORTEXM4.....8

7 SAMPLECODE\STDDRIVER.....9

    System Manager (SYS) .....9

    SPI Master (SPIM) .....9

    General Purpose I/O (GPIO).....9

    Timer Controller (TIMER).....9

    Watchdog Timer (WDT) .....10

    Window Watchdog Timer (WWDT) .....10

    Real Timer Clock (RTC) .....10

    PWM Generator and Capture Timer (PWM).....10

    UART Interface Controller (UART).....11

    Serial Peripheral Interface (SPI).....12

    I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C) .....12

    I<sup>2</sup>S Interface Controller (I<sup>2</sup>S).....13

    USB 2.0 Device Controller (USBD) .....14

    USB Host Controller (USBH) .....14

    Analog-to-Digital Converter (ADC) .....15

    SD Card Host Interface (SDH).....15

8 THIRDPARTY..... 16

## 1 Document

<p><b>CMSIS.html</b></p>	<p>Introduction of CMSIS version 4.5.0. CMSIS components included CMSIS-CORE, CMSIS-Driver, CMSIS-DSP, etc.</p> <ul style="list-style-type: none"> <li>● CMSIS-CORE: API for the Cortex-M4 processor core and peripherals.</li> <li>● CMSIS-Driver: Defines generic peripheral driver interfaces for middleware making it reusable across supported devices.505</li> </ul> <p>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).</p>
<p><b>NuMicro NUC505 Series Driver Reference Guide.chm</b></p>	<p>This document describes the usage of drivers in NUC505 BSP.</p>
<p><b>NuMicro NUC505 Series CMSIS BSP Revision History.pdf</b></p>	<p>This document shows the revision history of NUC505 BSP.</p>

## 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>StdDriver</b>	All peripheral driver header and source files.
<b>UsbHostLib</b>	USB host library source code.

### 3 SampleCode

<b>BootTemplate</b>	Different boot sample code.
<b>CortexM4</b>	Cortex®-M4 sample code.
<b>FreeRTOS</b>	Simple FreeRTOS™ demo code.
<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>ISP</b>	This demo code provides a solution for firmware update.
<b>ISP_Sample</b>	ISP firmware samples.
<b>Semihost</b>	Show how to print and get character through IDE console window.
<b>StdDriver</b>	Sample code to demonstrate the usage of M25 series MCU peripheral driver APIs.
<b>Template</b>	A project template for M25 series MCU.

## 4 SampleCode\BootTemplate

<b>CriticalOnSRAM</b>	Demonstrate how to locate a program mainly on SPI Flash for typical use, except critical parts on SRAM for fast execution.
<b>FullOnSRAM</b>	Demonstrate how to locate a program fully on SRAM.
<b>Loader</b>	Demonstrate how to launch a program via loader which is located fully on SRAM.
<b>MainOnSRAM</b>	Demonstrate how to locate a program mainly on SRAM for fast execution, except startup parts on SPI Flash for initialization.
<b>Overlay</b>	Demonstrate how to use overlay to run a large program in small size SRAM, through which a large program is divided into smaller ones which are located in the same SRAM address for execution.

## 5 SampleCode\ISP\_Sample

MTP_IAP_HID	Sample ISP firmware communicated with the ISP tool through a USB HID interface.
MTP_IAP_UART	Sample ISP firmware communicated with the ISP tool through a UART interface.

## 6 SampleCode\CortexM4

BitBand	Demonstrate the usage of Cortex®-M4 BitBand.
DSP_FFT	Demonstrate how to call ARM CMSIS DSP library to calculate FFT (Fast Fourier Transform).
MPU	Demonstrate the usage of Cortex®-M4 MPU.

## 7 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS</b>	<ol style="list-style-type: none"> <li>1. Demonstrate delay function by systick.</li> <li>2. Demonstrate core clock switching.</li> <li>3. Demonstrate how to enable module clock and set module clock divider.</li> </ol>
<b>SYS_PowerDownConsumption</b>	Demonstrate how to save power consumption in Power-down mode.

### SPI Master (SPIM)

<b>SPIM_CheckIF</b>	Check which interface SPI Flash is attached to.
<b>SPIM_DMA</b>	Demonstrate how to read/write SPI Flash in SPIM DMA mode.
<b>SPIM_DMM</b>	Demonstrate how to read SPI Flash in SPIM DMM (Direct Memory Map) mode.
<b>SPIM_IO</b>	Demonstrate how to read/write SPI Flash in SPIM (SPI Master) I/O mode.
<b>SPIM_SPIROM</b>	<p>Special notes for code running on SPI Flash:</p> <ol style="list-style-type: none"> <li>1. Switch to different clock safely, especially higher system clock.</li> <li>2. Embed MTP signature in the predefined location for security function.</li> </ol>

### General Purpose I/O (GPIO)

<b>GPIO</b>	Show the usage of GPIO external interrupt function and de-bounce function.
<b>GPIO_PowerDown</b>	Show how to wake up system form Power-down mode by GPIO interrupt.

### Timer Controller (TIMER)

<b>TIMER_Delay</b>	Demonstrate the usage of TIMER_Delay() API to
--------------------	---

	generate a 1 second delay.
TIMER_EventCounter	Use the pin PA.12 to demonstrate the timer event counter function.
TIMER_FreeCountingMode	Use the timer pin PA.13 to demonstrate timer free counting mode function. Also display the measured input frequency on the UART console.
TIMER_Periodic	Use the timer periodic mode to generate timer interrupt per one second.
TIMER_ToggleOut	Demonstrate the timer 0 toggle out function on pin PA.12.

### Watchdog Timer (WDT)

WDT_Polling	Use polling mode to check WDT time-out state and reset WDT after a time-out occurs.
WDT_Wakeup	Use WDT to wake up system from Power-down mode periodically.

### Window Watchdog Timer (WWDT)

WWDT_Reload	Demonstrate the WWDT counter reload function.
-------------	---

### Real Timer Clock (RTC)

RTC_Alarm_Mask_Test	Demonstrate the RTC alarm function. This sample code sets a minute alarm mask after execution.
RTC_Alarm_Test	Demonstrate the RTC alarm function. This sample code sets an alarm 10 seconds after execution.
RTC_Time_Display	Demonstrate the RTC function and display the current time on the UART console.

### PWM Generator and Capture Timer (PWM)

PWM_Capture	Demonstrate PWM Capture function by using PWM
-------------	---

	channel 2 to capture the output of PWM channel 0. Please connect PB.10 and PB.12 to execute this code.
<b>PWM_DeadZone</b>	Demonstrate the dead-zone feature with PWM.

**UART Interface Controller (UART)**

<b>UART_AutoBaudRate_Master</b>	Demonstrate how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Slave.
<b>UART_AutoBaudRate_Slave</b>	Demonstrate how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Master.
<b>UART_Autoflow_Master</b>	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
<b>UART_Autoflow_Slave</b>	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
<b>UART_IrDA_Master</b>	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
<b>UART_IrDA_Slave</b>	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
<b>UART_LIN</b>	Demonstrate how to transmit LIN header and response.
<b>UART_RS485_Master</b>	Demonstrate how to transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
<b>UART_RS485_Slave</b>	Demonstrate how to transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
<b>UART_TxRxFunction</b>	Demonstrate how UART transmits and receives data from PC terminal through RS232 interface.

UART_Wakeup	Demonstrate how to wake up system form Power-down mode by UART interrupt.
-------------	---

**Serial Peripheral Interface (SPI)**

SPI_Flash	Access the SPI Flash through a SPI interface.
SPI_Loopback	Demonstrate SPI master loop back transfer. This sample code needs to connect the SPI0_MISO0 pin and the SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterMode	Demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with SPI_SlaveMode.
SPI_SlaveMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterMode.

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

I2C_EEPROM	Show how to use I <sup>2</sup> C interface to access EEPROM.
I2C_Master	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Show how to set I <sup>2</sup> C in Slave mode and receive the data from Master. This sample code needs to work with I2C_master.
I2C_Wakeup_Master	Show how to wake up MCU from Power-down mode. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode via the I <sup>2</sup> C interface. This sample code needs to work with I2C_Master.

**I<sup>2</sup>S Interface Controller (I<sup>2</sup>S)**

<b>I2S_InternalCODEC</b>	An I <sup>2</sup> S demo using internal Audio CODEC used to playback the input from line-in or MIC interface.
<b>I2S_Master</b>	Demonstrate how I <sup>2</sup> S works in Master mode. This sample code needs to work with I2S_Slave.
<b>I2S_MP3PLAYER_SD</b>	A MP3 file player demo using internal Audio CODEC to play back a MP3 file stored in SD card.
<b>I2S_MP3PLAYER_USB</b>	A MP3 file player demo using internal Audio CODEC to play back a MP3 file stored in USB pen drive.
<b>I2S_NAU8822</b>	An I <sup>2</sup> S demo using NAU8822 Audio CODEC to playback the input from line-in or MIC interface.
<b>I2S_NAU8822_MP3PLAYER_SD</b>	A MP3 file player demo using NAU8822 Audio CODEC to play back a MP3 file stored in SD card.
<b>I2S_NAU8822_MP3PLAYER_USB</b>	A MP3 file player demo using NAU8822 Audio CODEC to play back a MP3 file stored in USB pen drive.
<b>I2S_NAU8822_WAVPLAYER_SD</b>	A WAV file player demo using NAU8822 Audio CODEC to play back a WAV file stored in SD card.
<b>I2S_NAU8822_WAVPLAYER_USB</b>	A WAV file player demo using NAU8822 Audio CODEC to play back a WAV file stored in USB pen drive.
<b>I2S_NAU8822_WAVRECORDER_SD</b>	A WAV file recorder demo using NAU8822 Audio CODEC to record a WAV file and save the file to SD card.
<b>I2S_Slave</b>	Demonstrate how I <sup>2</sup> S works in Slave mode. This sample code needs to work with I2S_Master.
<b>I2S_WAVPLAYER_SD</b>	A WAV file player demo using internal Audio CODEC to play back a WAV file stored in SD card.
<b>I2S_WAVPLAYER_USB</b>	A WAV file player demo using internal Audio CODEC to play back a WAV file stored in USB pen drive.
<b>I2S_WAVRECORDER_SD</b>	A WAV file recorder demo using internal Audio CODEC to record a WAV file and save the file to SD card.

**USB 2.0 Device Controller (USB D)**

<b>USB D_Audio_HeadSet</b>	An UAC1.0/UAC2.0 sample code used to record the sound transmitted to PC and play the sound from PC through the USB interface.
<b>USB D_Audio_Microphone</b>	An UAC1.0/UAC2.0 sample code used to record the sound to PC through the USB interface.
<b>USB D_Audio_Speaker</b>	An UAC1.0/UAC2.0 sample code used to play the sound sent from PC through the USB interface.
<b>USB D_HID_MOUSE</b>	Simulate a USB mouse and draw a circle on the screen.
<b>USB D_HID_Transfer</b>	Demonstrate how to transfer user-defined data – Command / Data Read / Data Write by a HID device.
<b>USB D_Mass_Storage_SD</b>	Implement a mass storage class sample for a SD card reader.
<b>USB D_Mass_Storage_ShortPacket</b>	Implement a mass storage class sample to demonstrate how to receive a USB short packet.
<b>USB D_Mass_Storage_SRAM</b>	Use internal SRAM as back-end storage media to simulate a 30KB USB pen drive.
<b>USB D_Mass_Storage_ScatterGather</b>	Demonstrate the usage of USB D DMA scatter gather function.
<b>USB D_VCOM_And_HID</b>	Demonstrate how to implement a USB virtual com port and a HID composite device.
<b>USB D_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual com port device.

**USB Host Controller (USB H)**

<b>USB H_HID</b>	Demonstrate reading inputs from a USB Mouse and displaying the input data (coordinate and button status) on the UART console. This sample includes a USB Mouse driver which is based on the HID driver.
<b>USB H_UAC_HID</b>	Demonstrate how to use an UAC+HID device.

**USBH\_UMAS**

Use a USB Host core driver, a USB mass storage driver, and a FATFS file system to show a disk access shell interface.

**Analog-to-Digital Converter (ADC)****ADC**

1. Demonstrate ADC conversion from channel 0.
2. Demonstrate analog keypad detection from channel 2.

**SD Card Host Interface (SDH)****SD\_FATFS**

Access a SD card formatted in the FAT file system.

## 8 ThirdParty

<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>LibMAD</b>	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 ( <a href="http://www.underbit.com/">http://www.underbit.com/</a> ) for the commercial license.

### 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.*