

# Nano100BN 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.

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

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

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

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

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

## TABLE OF CONTENTS

1	DOCUMENT.....	4
2	LIBRARY .....	5
3	SAMPLECODE .....	6
4	SAMPLECODE\CARDREADER.....	7
5	SAMPLECODE\ISP.....	8
6	SAMPLECODE\NUEDU .....	9
7	SAMPLECODE\NU-LB-NANO130.....	11
8	SAMPLECODE\NUTINY-EVB-NANO130.....	12
9	SAMPLECODE\STDDRIVER.....	13
	System Manager (SYS) .....	13
	Flash Memory Controller (FMC).....	13
	External Bus Interface (EBI) .....	13
	General Purpose I/O (GPIO).....	13
	PDMA Controller (PDMA) .....	14
	Timer Controller (TIMER).....	14
	Watchdog Timer (WDT) .....	14
	Window Watchdog Timer (WWDT) .....	15
	Real Timer Clock (RTC) .....	15
	PWM Generator and Capture Timer (PWM).....	15
	UART Interface Controller (UART).....	15
	Smartcard Host Interface (SC) .....	16
	Serial Peripheral Interface (SPI) .....	16
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	16
	USB Device Controller (USB).....	16
	LCD Display Driver (LCD).....	18
	CRC Controller (CRC) .....	18

Analog-to-Digital Converter (ADC) .....	18
Digital-to-Analog Converter (DAC) .....	19

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 4.5.0.
<b>NuMicro Nano100BN Series Driver Reference Guide.html</b>	This document describes the usage of drivers in Nano100BN BSP.
<b>NuMicro Nano100BN Series CMSIS BSP Revision History.pdf</b>	This document shows the revision history of Nano100BN 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>LCDLib</b>	Library for controlling LCD module.
<b>NuEdu</b>	Library for NuEdu board.
<b>SDCardLib</b>	Library for accessing a SD Card via SPI interface.
<b>SmartcardLib</b>	Smartcard library binary and header file.
<b>StdDriver</b>	All peripheral driver header and source files.

### 3 SampleCode

<b>CardReader</b>	Smartcard reader sample code.
<b>Hard_Fault_Sample</b>	Show hard fault information when hard fault happened.
<b>ISP</b>	ISP firmware samples.
<b>NuEdu</b>	Sample code for Nano130 NuEdu Evaluation Board.
<b>Nu-LB-NANO130</b>	Sample code for Nano130 Learning Board.
<b>NUTINY-EVB-NANO130</b>	Sample code for Nano130 Tiny Board.
<b>PowerDown_Chk</b>	Sample code which implements a function to test system state before entering power-down mode. If a system consumes more power than expected in power-down mode, this function can be used to check if there is any system setting that may cause power leakage.
<b>Semihost</b>	Show how to print and get character through IDE console window.
<b>StdDriver</b>	Sample code to demonstrate the usage of Nano100BN series MCU peripheral driver APIs.
<b>Template</b>	A project template for Nano100BN series MCU.

## **4 SampleCode\CardReader**

**USBD\_CCID**

CCID (Circuit card interface device) smart card reader sample code.

## 5 SampleCode\ISP

ISP_DFU	In-System-Programming sample code through USB DFU( Device Firmware Upgrade) class.
ISP_HID	In-System-Programming sample code through a USB HID interface.
ISP_I2C	In-System-Programming sample code through I <sup>2</sup> C 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\NuEdu

<b>Smpl_Basic01_ADC_Knob</b>	Demonstrate how to use ADC to measure variable resistor and change the LED brightness with PWM according to the ADC conversion results.
<b>Smpl_Basic01_FMC_IAP</b>	Demonstrate a simple IAP function to show three independent programs including main routine, independent interrupt handler and updating or switching to another program with IAP function.
<b>Smpl_Basic01_FMC_ISP</b>	Demonstrate LDROM updated through ISP function by branching to LDROM by software reset, and show debug messages via UART.
<b>Smpl_Basic01_I2C_EEPROM</b>	Demonstrate how to access EEPROM through I2C interface and print the test results on PC via NUCOM1 port of the NuEdu-Basic01 board.
<b>Smpl_Basic01_PWM_IrDA_NEC</b>	Demonstrate remote control function based on NEC IR protocol and changes LED display via NuEdu-EVB-Nano130 and NuEdu-Basci01 boards when system receives data of NEC IR.
<b>Smpl_Basic01_RTC</b>	Demonstrate RTC application and wake-up function, and print the results on the LCD glass via NuEdu-EVB-Nano130 and NuEdu-TNLCD boards.
<b>Smpl_Basic01_SPI_Flash</b>	Demonstrate how to access SPI Flash through SPI interface and print the test results on both 7-Segments and PC via NUCOM1 port of NuEdu-Basic01 board.
<b>Smpl_Basic01_SPI_Flash_w_PDMA</b>	Demonstrate how to access SPI Flash through a SPI interface with combing PDMA function and print the test results on both 7-Segments and PC via NUCOM1 port of the NuEdu-Basic01 board.
<b>Smpl_Basic01_StartKit</b>	Demonstrate RTC function and display the results on the LCD glass via NuEdu-EVB-Nano130 and NuEdu-TNLCD boards.
<b>Smpl_Basic01_Timer</b>	Demonstrate the timer periodic mode application and increase the number display on 7-segments from 0 to 99 via NuEdu-EVB-Nano130 and NuEdu-Basci01 boards

	when timer interrupt occurs.
<b>Smpl_Basic01_UART_printf</b>	Demonstrate a simple printf function to replace the standard printf library for reducing the code size issue.
<b>Smpl_Basic01_USBD_HID_Mouse</b>	Demonstrate how to implement a USB mouse device. The button on NuEdu-Basic01 can control the cursor when the mouse device is connected to PC by USB.
<b>Smpl_Basic01_WDT_WWDT</b>	Demonstrate the WDT and WWDT application via NuEdu-EVBNano130 and NuEdu-Basci01 boards. The buzzer will beep when WDT interrupt or WWDT interrupt occurs.

## 7 SampleCode\Nu-LB-NANO130

<b>FATFS_SDCard</b>	Access a SD card formatted in FAT file system.
<b>I2C_EEPROM</b>	Read/write EEPROM via an I <sup>2</sup> C interface.
<b>I2S_NAU8822</b>	An I <sup>2</sup> S demo using NAU8822 audio codec, used to play back the input from line-in or MIC interface.
<b>PWM_Capture</b>	Demonstrate PWM Capture function by using PWM0 channel 2 to capture the output of PWM0 channel 0.
<b>SC_ReadSimPhoneBook</b>	Demonstrate how to read phone book information in the SIM card.
<b>SPI_FIFO_Flash</b>	Access SPI Flash using FIFO mode.
<b>Timer_WDT_RTC</b>	Demonstrate the timer, WDT, and RTC function.
<b>USBD_Audio_Speaker</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_Audio_Speaker_And_HID_Transfer</b>	Demonstrate how to implement a USB audio class device and HID transfer. NAU8822 is used in this sample code to play the audio data from Host. It also supports to record data from NAU8822 to Host. 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_Mass_Storage_SDCard</b>	Use SD card as back end storage to implement a USB Mass-Storage device.

## 8 SampleCode\NUTINY-EVB-NANO130

<b>COMMON</b>	Common files for Tiny board sample code.
<b>LCD_DEMO</b>	Demonstrate how to display RTC time on a LCD panel.
<b>PWRDWN_DEMO</b>	Demonstrate how to keep LCD display while system enters Power-down mode.
<b>PWRDWN_LCD_RTC_DEMO</b>	Demonstrate how to keep LCD display while system enters Power-down mode and wake up system periodically with RTC interrupt.
<b>PWRDWN_RTC_DEMO</b>	Demonstrate how to wake up system periodically with RTC interrupt.

## 9 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS_Control</b>	Demonstrate how to change different PLL settings for the system clock source, and output system clock to CLKO (PC.5) pin with the system clock / 4 frequency.
<b>SYS_MCLKO</b>	Demonstrate how to output module clock to PC.0.
<b>SYS_PLLClockOutput</b>	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.
<b>SYS_TrimIRC</b>	Demonstrate how to use LXT to trim HIRC.

### Flash Memory Controller (FMC)

<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write function.
<b>FMC_VECMAP</b>	Show how to branch programs between LDROM, APROM start page, and APROM other page.

### External Bus Interface (EBI)

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

### General Purpose I/O (GPIO)

<b>GPIO_IOTest</b>	Use GPIO driver to control the GPIO pin direction and the high/low state, and show how to use GPIO interrupts.
<b>GPIO_PowerDown</b>	Demonstrate how to wake system up from Power-down

mode by GPIO interrupt.

## PDMA Controller (PDMA)

PDMA_Memory	Use PDMA channel 2 to demonstrate memory to memory transfer.
-------------	--

## Timer Controller (TIMER)

Timer_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
Timer_EventCounter	Use the pin PB.8 to demonstrate timer event counter function.
Timer_FreeCountingMode	Use the timer pin PC.6 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
Timer_InterTimerTriggerMode	Use the timer pin PB.8 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_ToggleOut	Demonstrate the timer 0 toggle out function on pin PB.8.
Timer_TriggerCountingMode	Use the timer pin PC.6 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
Timer_Wakeup	Use timer to wake up system from Power-down mode periodically.

## Watchdog Timer (WDT)

WDT_Polling	Use Polling mode to check WDT time-out state and reset WDT after 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_Test	Demonstrate the RTC alarm function which sets an alarm 10 seconds after execution.
RTC_Time_Display	Demonstrate the RTC function and display the current time to the UART console.

## PWM Generator and Capture Timer (PWM)

PWM_Capture	Demonstrate PWM Capture function by using PWM0 channel 2 to capture the output of PWM0 channel 0.
PWM_CapturePDMA	Demonstrate PWM Capture function by using PWM0 channel 2 to capture the output of PWM0 channel 0 and move captured data to SRAM with PDMA.
PWM_DeadZone	Demonstrate the dead-zone feature with PWM0.

## UART Interface Controller (UART)

UART_AutoBaudRate	Demonstrate how to use auto baud rate detection function.
UART_FlowCtrl	Transmit and receive data using auto flow control.
UART_IrDA	Show how to transmit and receive UART data in UART IrDA mode.
UART_LIN	Demonstrate how to transmit LIN header and response.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_RS485_Receive	Demonstrate how to receive data in UART RS485 mode.
UART_RS485_Transmit	Demonstrate how to transmit data in UART RS485 mode.

UART_Rx_Wakeup	Demonstrate how to wake up system from Power-down mode by UART interrupt.
UART_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.

### Smartcard Host Interface (SC)

SC_ReadATR	Read the smartcard ATR from smartcard 0 interface.
SCUART_TxRx	Demonstrate Smartcard UART mode by connecting PA.8 and PA.9 pins.

### Serial Peripheral Interface (SPI)

SPI_2BIT_Loopback	Demonstrate SPI 2 bit mode loop back transfer.
SPI_TxRxLoopback_PDMA	Demonstrate SPI loop back transfer with PDMA.

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

I2C_Loopback	An I <sup>2</sup> C master/slave demo by connecting I <sup>2</sup> C0 and I <sup>2</sup> C1 interface.
I2C_Wakeup	Demonstrate how to wake up system from Power-down mode by I <sup>2</sup> C interrupt.

### USB Device Controller (USBD)

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 an 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 use GPIO to simulate key input.
USBD_HID_Touch	Demonstrate how to implement a USB touch digitizer



	device. Two lines demo in Paint.
<b>USBD_HID_Transfer</b>	Demonstrate how to transfer data between 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_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 a USB device.
<b>USBD_HID_Transfer_And_MSC</b>	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_HID_Transfer_CTRL</b>	Use USB Host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_Mass_Storage_CDRom</b>	Demonstrate the emulation of USB Mass Storage Device CD-ROM.
<b>USBD_Mass_Storage_Flash</b>	Use internal Flash as back end storage media to simulate a USB pen drive.
<b>USBD_Mass_Storage_SD</b>	Demonstrate how to implement a SD card reader.
<b>USBD_Micro_Printer</b>	Demonstrate how to implement a USB micro printer device.
<b>USBD_Printer_And_HID_Transfer</b>	Demonstrate how to implement a composite device. (USB micro printer device and HID Transfer). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_VCOM_And_HID_Keyboard</b>	Demonstrate how to implement a composite device.(VCOM and HID keyboard)
<b>USBD_VCOM_And_HID_</b>	Demonstrate how to implement a composite

<b>Transfer</b>	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.
<b>USBD_VCOM_And_Mass_Storage</b>	Demonstrate how to implement a composite device.(Virtual COM port and Mass storage device)
<b>USBD_VCOM_DualPort</b>	Demonstrate how to implement a USB dual virtual COM port device.
<b>USBD_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual COM port device.
<b>USBD_Vendor_LBK</b>	A USB device vendor class sample program. This sample program needs to test with USBH_VENDOR_LBK that is executed on NUC442/472.

## LCD Display Driver (LCD)

<b>LCD_Blinking_Test</b>	Demonstrate LCD blinking function on LCD panel of NUTINYSDK- NANO130-LQFP128_TNLCD board.
<b>LCD_Pixel_OnOff_Test</b>	Show how to control pixel on and off on LCD panel of NUTINY-SDK-NANO130-LQFP128_TNLCD board.
<b>LCD_Print_Text_Test</b>	Show how to print text on LCD panel of NUTINY-SDKNANO130-LQFP128_TNLCD board.

## CRC Controller (CRC)

<b>CRC_CCITT</b>	Calculate the CRC-CCITT checksum value by CRC DMA mode.
------------------	---

## Analog-to-Digital Converter (ADC)

<b>ADC_Compare</b>	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
<b>ADC_ContinuousScan</b>	Convert ADC channel 0, 1, 2 in Continuous Scan mode and print conversion results.

<b>ADC_PDMA</b>	Use PDMA channel 1 to move ADC channel 0, 1, 2 converted data to SRAM.
<b>ADC_Single</b>	Convert ADC channel 0 in Single mode and print conversion results.
<b>ADC_SingleCycleScan</b>	Convert ADC channel 0, 1, 2 in Single Cycle Scan mode and print conversion results.
<b>ADC_TimerTrigger</b>	Configure Timer0 to ADC and move converted data to SRAM using PDMA.

### Digital-to-Analog Converter (DAC)

<b>DAC_PDMATrigger</b>	Demonstrate PDMA trigger DAC convert sine wave outputs.
<b>DAC_SoftwareTrigger</b>	Demonstrate software trigger DAC convert sine wave outputs.
<b>DAC_TimerTrigger</b>	Demonstrate timer trigger DAC convert sine wave outputs.

### **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.*