

## Nano102/112 CMSIS BSP Directory

Directory Introduction for 32-bit NuMicro™ Family

### Directory Information

<b>Document</b>	Driver reference manual 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)

## 1 Document Information

<b>NuMicro Nano102_112 Series CMSIS BSP Revision History.pdf</b>	This document shows the revision history of Nano102/112 BSP.
<b>NuMicro Nano102_112 Series Driver Reference Guide.chm</b>	This document describes the usage of drivers in Nano102/112 BSP.

## 2 Library Information

<b>CMSIS</b>	Cortex <sup>®</sup> Microcontroller Software Interface Standard (CMSIS) V3.01 definitions by ARM <sup>®</sup> Corp.
<b>Device</b>	CMSIS compliant device header file.
<b>LCDLib</b>	Library for controlling LCD module.
<b>SmartcardLib</b>	Library for accessing a smartcard.
<b>StdDriver</b>	All peripheral driver header and source files.

### 3 Sample Code Information

<b>Hard_Fault_Sample</b>	Show hard fault information when hard fault happened.
<b>Nu-LB-NANO112</b>	Sample codes for Nano112 Learning Board
<b>NUTINY-EVB-NANO112</b>	Sample codes for Nano112 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 with IDE console window.
<b>StdDriver</b>	Demonstrate the usage of Nano102/112 series MCU peripheral driver APIs.
<b>Template</b>	A project template for Nano102/112 series MCU.

## 4 \SampleCode\Nu-LB-NANO112

<b>COMMON</b>	Common files for Learning Board sample code.
<b>StartKit</b>	This sample code displays 'NANO' on LCD.

## **5 \SampleCode\NUTINY-EVB-NANO112**

<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.

## 6 \SampleCode\StdDriver

<b>ACMP_Sigma_Delta</b>	Use Sigma-Delta mode to measure input voltage.
<b>ACMP_Slope</b>	Use ACMP slop mode to measure capacitor discharge time.
<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.
<b>CRC_CCITT</b>	Calculate the CRC-CCITT checksum value by CRC DMA mode.
<b>FMC_IAP</b>	Demonstrate IAP (In-Application Programming) function. To run this sample, the boot mode must be "Boot from APROM with IAP".
<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write function.
<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 form Power-down mode by GPIO interrupt.
<b>I2C_EEPROM</b>	Read/write EEPROM via an I <sup>2</sup> C interface.
<b>I2C_Loopback</b>	An I <sup>2</sup> C master/slave demo by connecting I <sup>2</sup> C0 and I <sup>2</sup> C1 interface.

<b>I2C_Wakeup</b>	Demonstrate how to wake up system form Power-down mode by I <sup>2</sup> C interrupt.
<b>LCD_Blinking_Test</b>	Demonstrate LCD blinking function on LCD panel of NuTiny-SDK-Nano112-LQFP_TNLCD board.
<b>LCD_Pixel_OnOff_Test</b>	Show how to control pixel on and off on LCD panel of NuTiny-SDK-Nano112-LQFP_TNLCD board.
<b>LCD_Print_Text_Test</b>	Show how to print text on LCD panel of NuTiny-SDK-Nano112-LQFP_TNLCD board.
<b>PDMA_Memory</b>	Use PDMA channel 2 to demonstrate memory to memory transfer.
<b>PWM_Capture</b>	Demonstrate PWM Capture function by using PWM0 channel 2 to capture the output of PWM0 channel 0.
<b>PWM_CapturePDMA</b>	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.
<b>PWM_DeadZone</b>	Demonstrate the dead-zone feature with PWM0.
<b>RTC_Alarm_Test</b>	Demonstrate the RTC alarm function which sets an alarm 10 seconds after execution.
<b>RTC_Time_Display</b>	Demonstrate the RTC function and display the current time to the UART console.
<b>SC_ReadATR</b>	Read the smartcard ATR from smartcard 0 interface.
<b>SC_ReadSimPhoneBook</b>	Demonstrate how to read phone book information in the SIM card.
<b>SCUART_TxRx</b>	Demonstrate smartcard UART mode by connecting PC.4 and PC.6 pins.
<b>SPI_FIFO_Flash</b>	Access SPI Flash using FIFO mode.
<b>SPI_LoopBack</b>	Demonstrate SPI loop back transfer
<b>SPI_TxRxLoopback_PDMA</b>	Demonstrate SPI loop back transfer with PDMA.
<b>SYS_Control</b>	Demonstrate how to change different PLL settings for the system clock source, and output system clock to CLKO (PB.12) pin with the system clock / 4 frequency.



<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLKO pin.
<b>SYS_TrimIRC</b>	Demonstrate how to use LXT to trim HIRC.
<b>Timer_Delay</b>	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
<b>Timer_EventCounter</b>	Use the pin PB.8 to demonstrate timer event counter function.
<b>Timer_FreeCountingMode</b>	Use the timer pin PD.11 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
<b>Timer_InterTimerTriggerMode</b>	Use the timer pin PB.8 to demonstrate inter timer trigger mode function. Also display the measured input frequency to UART console.
<b>Timer_Periodic</b>	Use the timer periodic mode to generate timer interrupt every 1 second.
<b>Timer_ToggleOut</b>	Demonstrate the timer 0 toggle out function on pin PB.8.
<b>Timer_TriggerCountingMode</b>	Use the timer pin PD.11 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
<b>Timer_Wakeup</b>	Use timer to wake up system from Power-down mode periodically.
<b>UART_AutoBaudRate</b>	Demonstrate how to use auto baud rate detection function.
<b>UART_FlowCtrl</b>	Transmit and receive data using auto flow control.
<b>UART_IrDA</b>	Show how to transmit and receive UART data in UART IrDA mode.
<b>UART_LIN</b>	Demonstrate how to transmit LIN header and response.
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.
<b>UART_RS485_Receive</b>	Demonstrate how to receive data in UART RS485 mode.
<b>UART_RS485_Transmit</b>	Demonstrate how to transmit data in UART RS485

	mode.
<b>UART_Rx_Wakeup</b>	Demonstrate how to wake up system form Power-down mode by UART interrupt.
<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>WDT_Polling</b>	Use Polling mode to check WDT time-out state and reset WDT after time-out occurs.
<b>WDT_Wakeup</b>	Use WDT to wake up system from Power-down mode periodically.
<b>WWDT_Reload</b>	Demonstrate the WWDT counter reload function.

**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.*