

Nano100AN CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference guide and revision history.
Library	Driver header and source files.
SampleCode	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

TABLE OF CONTENTS

1	DOCUMENT.....	3
2	LIBRARY	4
3	SAMPLECODE	5
4	SAMPLECODE\ISP	6
5	SAMPLECODE\STDDRIVER.....	7
	System Manager (SYS)	7
	Flash Memory Controller (FMC).....	7
	External Bus Interface (EBI)	7
	General Purpose I/O (GPIO).....	7
	PDMA Controller (PDMA)	7
	Timer Controller (TIMER).....	8
	Watchdog Timer (WDT)	8
	Real Timer Clock (RTC)	8
	PWM Generator and Capture Timer (PWM).....	9
	UART Interface Controller (UART).....	9
	Smartcard Host Interface (SC)	10
	Serial Peripheral Interface (SPI)	10
	I ² C Serial Interface Controller (I ² C)	10
	I ² S Controller (I ² S).....	10
	USB Device Controller (USB).....	10
	Analog-to-Digital Converter (ADC)	12

1 Document

CMSIS.html	Document of CMSIS version 4.5.0.
NuMicro Nano100AN Series Driver Reference Guide.html	This document describes the usage of drivers in Nano100AN BSP.
NuMicro Nano100AN Series CMSIS BSP Revision History.pdf	This document shows the revision history of Nano100AN BSP.

2 Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by Arm® Corp.
Device	CMSIS compliant device header file.
SDCardLib	Library for accessing a SD Card via SPI interface.
SmartcardLib	Smartcard library binary and header file.
StdDriver	All peripheral driver header and source files.

3 SampleCode

CardReader	Smartcard reader sample code.
Hard_Fault_Sample	Show hard fault information when hard fault happened.
ISP	ISP firmware samples.
PowerDown_Chk	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.
PWRDWN_DEMO	Demonstrate system enters Power-down mode.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of Nano100AN series MCU peripheral driver APIs.
Template	A project template for Nano100AN series MCU.

4 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 ² 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.

5 SampleCode\StdDriver

System Manager (SYS)

SYS_Control	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.
SYS_TrimIRC	Demonstrate how to use LXT to trim HIRC.

Flash Memory Controller (FMC)

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

External Bus Interface (EBI)

EBI_NOR	Configure EBI interface to access NOR Flash connected to EBI interface.
EBI_SRAM	Configure EBI interface to access SRAM connected to EBI interface.

General Purpose I/O (GPIO)

GPIO_IOTest	Use GPIO driver to control the GPIO pin direction and the high/low state, and show how to use GPIO interrupts.
GPIO_PowerDown	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.

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.
RTC_Wakeup	Demonstrate how to wake up system periodically with RTC interrupt.

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_FIFO_FLASH	Access SPI Flash using FIFO mode.
SPI_SDCARD	Access a SD card formatted in FAT file system.
SPI_TxRxLoopback_PDMA	Demonstrate SPI loop back transfer with PDMA.

I²C Serial Interface Controller (I²C)

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

I²S Controller (I²S)

I2S_NAU8822	An I ² S demo using NAU8822 audio codec, used to play back the input from line-in or MIC interface.
-------------	--

USB Device Controller (USBD)

USBD_Audio_Speaker	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.
USBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device. It 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.
USBD_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.
USBD_HID_Transfer_And_Keyboard	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.
USBD_HID_Transfer_And_MSC	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.
USBD_HID_Transfer_CTRL	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.
USBD_Mass_Storage_CDROM	USB Mass Storage Device CD-ROM Emulation.
USBD_Mass_Storage_Flash	Use internal Flash as back end storage media to simulate a USB pen drive.
USBD_Mass_Storage_SDCard	Demonstrate how to implement a SD card reader.
USBD_Micro_Printer	Demonstrate how to implement a USB micro printer device.
USBD_VCOM_And_HID_Keyboard	Demonstrate how to implement a composite device.(VCOM and HID keyboard)
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual com port device.

Analog-to-Digital Converter (ADC)

ADC_Compare	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
ADC_ContinuousScan	Convert ADC channel 0, 1, 2 in Continuous Scan mode and print conversion results.
ADC_PDMA	Use PDMA channel 1 to move ADC channel 0, 1, 2 converted data to SRAM.
ADC_Single	Convert ADC channel 0 in Single mode and print conversion results.
ADC_SingleCycleScan	Convert ADC channel 0, 1, 2 in Single Cycle Scan mode and print conversion results.
ADC_TimerTrigger	Configure Timer0 to ADC and move converted data to SRAM using PDMA.

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.*