NUC029xAN Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro™ Family

Directory Information

Please extract the “NUC029xANSeries_BSP_CMSIS_V3.00.006.zip” file firstly, and then put the “NUC029xANSeries_BSP_CMSIS_V3.00.006” folder into the working folder (e.g. .\Nuvoton\BSP Library\).

This BSP folder contents:

Document	Device driver reference manual and reversion history.
Library	Device driver header and source files.
SampleCode	Device driver sample code.

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

www.nuvoton.com

TABLE OF CONTENTS

1 DOCUMENT 4

2 LIBRARY 5

3 SAMPLE CODE 6

4 SAMPLECODE\ISP 7

5 SAMPLECODE\REGBASED 8

System Manager (SYS) 8
Flash Memory Controller (FMC) 8
General Purpose I/O (GPIO) 8
Timer Controller (TIMER) 8
Watchdog Timer (WDT) 9
Window Watchdog Timer (WWDT) 9
PWM Generator and Capture Timer (PWM) 9
UART Interface Controller (UART) 9
Serial Peripheral Interface (SPI) 10
I²C Serial Interface Controller (I²C) 10
External Bus Interface (EBI) 11
Analog-to-Digital Converter (ADC) 11
Analog Comparator Controller (ACMP) 12
Hardware Divider (HDIV) 12

6 SAMPLECODE\STDDRIVER 13

System Manager (SYS) 13
Flash Memory Controller (FMC) 13
General Purpose I/O (GPIO) 13
Timer Controller (TIMER) 13
Watchdog Timer (WDT) 14
Window Watchdog Timer (WWDT) 14
PWM Generator and Capture Timer (PWM) 14
UART Interface Controller (UART) 14
Serial Peripheral Interface (SPI) 15
I²C Serial Interface Controller (I²C) 15

External Bus Interface (EBI) 16
Analog-to-Digital Converter (ADC) 16
Analog Comparator Controller (ACMP) 17
Hardware Divider (HDIV) 17

1 Document

Revision History.pdf	The revision history of NUC029xAN Series BSP.
NuMicro NUC029xAN Driver Reference Guide.chm	The usage of drivers in NUC029xAN Series BSP.

2 Library

CMSIS	CMSIS definitions by ARM® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.

3 Sample Code

Hard_Fault_Sample	Show hard fault information when hard fault happened. 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. 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.
ISP	Sample codes for In-System-Programming.
Semihost	Show how to print and get character through IDE console window.
RegBased	The sample codes which access control registers directly.
StdDriver	Demonstrate the usage of NUC029xAN series MCU peripheral driver APIs.
Template	A project template for NUC029xAN series MCU.

4 SampleCode\ISP

ISP_I2C	In-System-Programming Sample code through I2C 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\RegBased

System Manager (SYS)

SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.

Flash Memory Controller (FMC)

FMC_IAP	Show how to call LDROM functions from APROM. The code in APROM will look up the table at 0x100E00 to get the address of function of LDROM and call the function.
FMC_RW	Demonstrate how to read/program embedded Flash by ISP function.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.

Timer Controller (TIMER)

TIMER_Capture	Demonstrate how to use timer2 capture event to capture timer2 counter value.
TIMER_Counter	Demonstrate how to use timer1 counter input function to count the input event.
TIMER_PeriodicINT	Demonstrate how to perform timer counting in periodic mode.

TIMER_PowerDown	Demonstrate how to use timer0 toggle-output interrupt event to wake-up system.

Watchdog Timer (WDT)

WDT_PowerDown	Demonstrate how to use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Select one WDT time-out interval period time to generate time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to cause WDT time-out reset system event while WDT time-out reset delay period expired.

Window Watchdog Timer (WWDT)

WWDT_CompareINT	Select one WWDT window compare value to generate window compare match interrupt event.

PWM Generator and Capture Timer (PWM)

PWM	Demonstrate how to use PWM to generate different frequency (Tenor C Do ~ Si) waveform.
PWM_Capture	Demonstrate how to use PWMB Channel 2 captures PWMB Channel 1 Waveform.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Use PWM Double Buffer function to change duty cycle and period of output waveform.

UART Interface Controller (UART)

UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.

UART_IrDA_Master	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Transmit LIN frame including header and response in UART LIN mode.
UART_RS485_Master	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.

Serial Peripheral Interface (SPI)

SPI_Loopback	Demonstrate the data transfer between a SPI master and a SPI slave.
SPI_MasterFifoMode	Demonstrate how to communicate with an off-chip SPI Slave device with FIFO mode. This sample code needs to work with SPI_SlaveFifoMode.
SPI_SlaveFifoMode	Demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with SPI_MasterFifoMode.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM through a I2C interface
I2C_GCMode_Master	Demonstrate how a Master uses I2C address 0x0 to write data to I2C Slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_GCMode_Slave	Demonstrate how to receive Master data in GC (General Call) mode. This sample code needs to work with

	I2C_GCMode_Master.
I2C_Master	Demonstrate how a Master accesses a Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Demonstrate how to set I2C in slave mode to receive data from a Master. This sample code needs to work with I2C_Master.
I2C_Wakeup_Master	Demonstrate how to wake-up MCU from power-down. Needs to work with I2C_Wakeup_Slave sample code.
I2C_Wakeup_Slave	Demonstrate how to set I^2C to wake-up MCU from power-down mode. Needs to work with I2C_Wakeup_Master sample code.

External Bus Interface (EBI)

EBI_NOR	Demonstrate how to read/program external NOR Flash device (W39L040P) through EBI bus.
EBI_SRAM	Demonstrate how to read/program external SRAM device (BS616LV4017) through EBI bus.

Analog-to-Digital Converter (ADC)

ADC_BurstMode	Demonstrate A/D conversion with burst mode. In burst mode, ADC will sample and convert a specified channel continuously and store the conversion result in FIFO buffers.
ADC_ContinuousScanMode	Demonstrate how to use continuous scan mode and finishes two cycles of conversion for the specified channels.
ADC_PwmTrigger	Demonstrate how to trigger ADC by PWM.
ADC_ResultMonitor	Demonstrate how to use the digital compare function to monitor the conversion result of channel 2.
ADC_SingleCycleScanMode	Demonstrate how to use single cycle scan mode and finishes one cycle of conversion for the specified channels.
ADC_SingleMode	Demonstrate how to use single mode and finishes the conversion of the specified channel.

Analog Comparator Controller (ACMP)

ACMP	Demonstrate how ACMP[1] works with internal band-gap voltage.

1. Analog Comparator (ACMP).

Hardware Divider (HDIV)

HDIV	Demonstrate how to user divider API and how to use hardware divider by control registers.

6 SampleCode\StdDriver

System Manager (SYS)

SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.

Flash Memory Controller (FMC)

FMC_IAP	Demonstrate how to reboot to LDROM functions from APROM. This sample code set VECMAP to LDROM and reset to re-boot to LDROM.
FMC_RW	Demonstrate how to read/program embedded flash by ISP function.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.

Timer Controller (TIMER)

TIMER_Capture	Demonstrate how to use timer2 capture event to capture timer2 counter value.
TIMER_Counter	Demonstrate how to use timer1 counter input function to count the input event.
TIMER_PeriodicINT	Demonstrate how to perform timer counting in periodic

	mode.
TIMER_PowerDown	Demonstrate how to use timer0 toggle-output interrupt event to wake-up system.

Watchdog Timer (WDT)

WDT_PowerDown	Demonstrate how to use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Select one WDT time-out interval period time to generate time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to cause WDT time-out reset system event while WDT time-out reset delay period expired.

Window Watchdog Timer (WWDT)

WWDT_CompareINT	Select one WWDT window compare value to generate window compare match interrupt event.

PWM Generator and Capture Timer (PWM)

PWM	Demonstrate how to use PWM to generate different frequency (Tenor C Do ~ Si) waveform.
PWM_Capture	Demonstrate how to use PWMB Channel 2 captures PWMB Channel 1 Waveform.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Use PWM Double Buffer function to change duty cycle and period of output waveform.

UART Interface Controller (UART)

UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.

UART_IrDA_Master	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Transmit LIN frame including header and response in UART LIN mode.
UART_RS485_Master	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.

Serial Peripheral Interface (SPI)

SPI_Loopback	Demonstrate the data transfer between a SPI master and a SPI slave.
SPI_MasterFIFOMode	Demonstrate how to communicate with an off-chip SPI Slave device with FIFO mode. This sample code needs to work with SPI_SlaveFifoMode.
SPI_SlaveFIFOMode	Demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with SPI_MasterFifoMode.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM through a I2C interface
I2C_GCMode_Master	Demonstrate how a Master uses I2C address 0x0 to write data to I2C Slave. This sample code needs to work with I2C_GCMode_Slave.

I2C_GCMode_Slave	Demonstrate how to receive Master data in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Master	Demonstrate how a Master accesses Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Demonstrate how to set I2C in slave mode to receive the data from a Master. This sample code needs to work with I2C_Master.
I2C_Wakeup_Master	Demonstrate how to wake-up MCU from power-down. Needs to work with I2C_Wakeup_Slave sample code.
I2C_Wakeup_Slave	Demonstrate how to set I^2C to wake-up MCU from power-down mode. Needs to work with I2C_Wakeup_Master sample code.

External Bus Interface (EBI)

EBI_NOR	Demonstrate how to read/program external NOR Flash device (W39L040P) through EBI bus.
EBI_SRAM	Demonstrate how to read/program external SRAM device (BS616LV4017) through EBI bus.

Analog-to-Digital Converter (ADC)

ADC_BurstMode	Demonstrate A/D conversion with burst mode. In burst mode, ADC will sample and convert a specified channel continuously and store the conversion result in FIFO buffers.
ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_PwmTrigger	Demonstrate how to trigger ADC by PWM.
ADC_ResultMonitor	Monitor the conversion result of Channel 2 by the digital compare function.
ADC_SingleCycleScanMode	Perform A/D Conversion with ADC single cycle scan mode.

ADC_SingleMode	Perform A/D Conversion with ADC single mode.

Analog Comparator Controller (ACMP)

ACMP	Demonstrate how ACMP works with internal band-gap voltage.

Hardware Divider (HDIV)

HDIV	Demonstrate how to user divider API and how to use hardware divider by control registers.

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.