NUC029xAN Board Support Package Directory Introduction

Rev 3.00.005

Directory Information

Document	Driver reference manual and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.

Document Information

BSP Revision History	Show all the revision history about specific BSP.
Driver Reference Guide	Describe the definition, input and output of each API.

Library Information

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

Sample Code Information

\SampleCode\Hard_Fault_ Sample	Show hard fault information when hard fault happened.
\SampleCode\ISP	Sample codes for In-System-Programming.
\SampleCode\Template	Software Development Template.
\SampleCode\Semihost	A sample code to show how to debug with semihost message print.
\SampleCode\RegBased	The sample codes which access control registers directly.
\SampleCode\StdDriver	NUC029xAN Driver Samples

\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.

\SampleCode\RegBased

ACMP	Demonstrate how ACMP[1] works with internal band-gap voltage.
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.
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.
FMC_IAP	Demonstrate 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.
GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use GPIO interrupt function.
GPIO_OutputInput	Demonstrate how to set GPIO pin mode and use pin data input/output control.

GPIO_PowerDown	Demonstrate how to wake-up form Power-down mode by GPIO interrupt.
HDIV	Demonstrate how to user divider API and how to use hardware divider by control registers.
I2C_EEPROM	Demonstrate how to access EEPROM by I^2C interface.
I2C_GCMode_MASTER	Demonstrate how a Master uses I^2C address 0x0 to write data to I^2C Slave. Needs to work with I2C_GCMode_SLAVE sample code.
I2C_GCMode_SLAVE	Demonstrate how to receive Master data in GC (General Call) mode. Needs to work with I2C_GCMode_MASTER sample code.
I2C_MASTER	Demonstrate how a Master access Slave. Needs to work with I2C_SLAVE sample code.
I2C_SLAVE	Demonstrate how to set I^2C in slave mode to receive the data of a Master. Needs to work with I2C_MASTER sample code.
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.
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.
SPI_LoopBackTest	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 sample code.
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 sample code.
SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
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.
UART_Autoflow_Master	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Master.
UART_LIN	Demonstrate how to transmit LIN header and response.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485

	mode. The sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Demonstrate how UART transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system form Power-down mode by UART interrupt.
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.
WWDT_CompareINT	Select one WWDT window compare value to generate window compare match interrupt event.

1. Analog Comparator (ACMP).

\SampleCode\StdDriver

ACMP	Demonstrate how ACMP works with internal band-gap voltage.
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.
CAN_BasicMode_Receive	Demonstrate how to receive message in Basic mode. The sample code needs to work with CAN_BasicMode_Transmit.
CAN_BasicMode_Transmit	Demonstrate how to transmit message in Basic mode. The sample code needs to work with CAN_BasicMode_Receive.
CAN_NormalMode_Receive	Demonstrate how to receive message in Normal mode. The sample code needs to work with CAN_NormalMode_Transmit.
CAN_NormalMode_Transmit	Demonstrate how to transmit message in Normal mode. The sample code needs to work with CAN_NormalMode_Receive.
FMC	Demonstrate how to access embedded flash and switching between APROM and LDROM.
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.
FMC_MultiBoot	Demonstrate how to implement multi-boot system to boot from different applications in APROM.
GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use GPIO interrupt function.
GPIO_OutputInput	Demonstrate how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Demonstrate how to wake-up form Power-down mode by GPIO interrupt.
HDIV	Demonstrate how to user divider API and how to use

	hardware divider by control registers.
I2C_EEPROM	Demonstrate how to access EEPROM by I^2C interface.
I2C_GCMode_MASTER	Demonstrate how a Master uses I^2C address 0x0 to write data to I^2C Slave. Needs to work with I2C_GCMode_SLAVE sample code.
I2C_GCMode_SLAVE	Demonstrate how to receive Master data in GC (General Call) mode. Needs to work with I2C_GCMode_MASTER sample code.
I2C_MASTER	Demonstrate how a Master access Slave. Needs to work with I2C_SLAVE sample code.
I2C_SLAVE	Demonstrate how to set I^2C in slave mode to receive the data of a Master. Needs to work with I2C_MASTER sample code.
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.
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.
SPI_LoopBackTest	Demonstrates the data transfer in SPI master mode.
SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
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.
UART_Autoflow_Master	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Master.
UART_LIN	Demonstrate how to transmit LIN header and response.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Demonstrate how UART transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system form Power-down mode by UART interrupt.
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.
WWDT_CompareINT	Select one WWDT window compare value to generate window compare match interrupt event.

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.