NUC200 CMSIS BSP Directory

Directory Introduction for 32-bit NuMicro™ Family

Directory Information

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

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

www.nuvoton.com

TABLE OF CONTENTS

1 Document Information

Revision History	Show all the revision history about specific BSP.
NuMicro NUC200 Series Driver Reference Guide.chm	Describe the definition, input and output of each API.

2 Library Information

CMSIS	CMSIS definitions by ARM® Corp.
Device	CMSIS compliant device header file.
SmartcardLib	Library for accessing a smartcard.
StdDriver	All peripheral driver header and source files.

3 Sampel Code Information

CardReader	CCID[1] Smart Card reader Sample Code.
Hard_Fault_Sample	Show hard fault information when hard fault happened.
ISP	Sample codes for In-System-Programming.
Template	Software Development Template.
Semihost	Show how to debug with semi-host message print.
RegBased	The sample codes which access control registers directly.
StdDriver	NUC200 Series Driver Samples

1. Circuit card interface device (CCID) is USB device that interface with integrated circuit cards.

4 SampleCode\ISP

ISP_DFU	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
ISP_HID	In-System-Programming Sample code through USB HID interface.
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	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_MultiBoot_SwReset	Show how to use software reset to implement multi-boot system to boot from different applications in APROM.
FMC_RW	Show 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.

PDMA Controller (PDMA)

PDMA	Use PDMA channel 6 to transfer data from memory to memory.

Timer Controller (TIMER)

TIMER_Capture	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_Counter	Implement timer1 event counter function to count the external input event.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_PowerDown	Use timer0 toggle-output time-out interrupt event to wake up system.

Watchdog Timer (WDT)

WDT_PowerDown	Use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Implement periodic WDT time-out interrupt event.
WDT_TimeoutReset	Show how to generate time-out reset system event while WDT time-out reset delay period expired.

Window Watchdog Timer (WWDT)

WWDT_CompareINT	Show how to reload the WWDT counter value.

Real Timer Clock (RTC)

RTC_PowerDown	Use RTC alarm interrupt event to wake-up system.
RTC_TimeAndTick	Get the current RTC data/time per tick.

PWM Generator and Capture Timer (PWM)

PWM_Capture	Capture the PWMB Channel 1 waveform by PWMB Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.

PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.

UART Interface Controller (UART)

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_PDMA	Transmit and receive UART data with PDMA.
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.

Serial Peripheral Interface (SPI)

SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterFifoMode	Configure SPI0 as Master mode and 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_PDMA_Loopback	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
SPI_SlaveFifoMode	Configure SPI0 as Slave mode and 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.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM by I²C interface.
I2C_GCMode_Master	Demonstrate how a Master uses I²C address 0x0 to write data to I²C 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 I²C 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 I²C to wake-up MCU from power-down mode. Needs to work with I2C_Wakeup_Master sample code.

I²S Controller (I²S)

I2S_Master	Demonstrate how I²S works in Master mode. This sample code needs to work with I2S_Slave sample code.
I2S_PDMA	Demonstrate how I2S works with PDMA in Master mode. Both TX PDMA function and RX PDMA function will be enabled.
I2S_Slave	Demonstrate how I²S works in Slave mode. This sample code needs to work with I2S_Master sample code.

CRC Controller (CRC)

CRC_8	Implement CRC in CRC-8 mode and get the CRC checksum result.
CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.

Analog-to-Digital Converter (ADC)

ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_MeasureAVDD	Measure AVDD voltage by ADC.
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[1] works with internal band-gap voltage.
ACMP_Wakeup	Show how to wake up MCU from Power-down mode by ACMP wake-up function.

1. Analog Comparator (ACMP).

PS/2 Controller (PS/2)

PS2	Demonstrate how to emulate a PS/2 mouse by moving mouse pointer when connecting to PC by PS/2 interface.

6 SampleCode\StdDriver

System Manager (SYS)

SYS	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 execute LDROM functions from APROM. This sample code set VECMAP to LDROM and execute the code in LDROM by functional pointer.
FMC_RW	Show 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.

PDMA Controller (PDMA)

PDMA	Use PDMA channel 6 to transfer data from memory to memory.

Timer Controller (TIMER)

TIMER_Capture	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_Counter	Implement timer1 event counter function to count the external input event.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_PowerDown	Use timer-0 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.

Real Timer Clock (RTC)

RTC_PowerDown	Use RTC alarm interrupt event to wake-up system.
RTC_TimeAndTick	Get the current RTC data/time per tick.

PWM Generator and Capture Timer (PWM)

PWM_Capture	Capture the PWMB Channel 1 waveform by PWMB

	Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.

UART Interface Controller (UART)

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_PDMA	Transmit and receive UART data with PDMA.
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.

Smart Card Host Interface (SC)

SC_ReadATR	Read the smartcard ATR from smartcard 0 interface.

I²S Controller (I²S)

I2S_Master	Demonstrate how I²S works in Master mode. This sample code needs to work with I2S_Slave sample code.
I2S_PDMA	Demonstrate how I2S works with PDMA in Master mode. Both TX PDMA function and RX PDMA function will be enabled.
I2S_Slave	Demonstrate how I²S works in Slave mode. This sample code needs to work with I2S_Master sample code.

USB Device Controller (USBD)

USBD_Audio_NAU8822	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	Show how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.

Serial Peripheral Interface (SPI)

SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterFIFOMode	Configure SPI0 as Master mode and 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_PDMA_Loopback	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
SPI_SlaveFIFOMode	Configure SPI0 as Slave mode and 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.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM by I²C interface.
I2C_GCMode_Master	Demonstrate how a Master uses I²C address 0x0 to write data to I²C 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 I²C 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 I²C to wake-up MCU from power-down mode. Needs to work with I2C_Wakeup_Master sample code.

CRC Controller (CRC)

CRC_8	Implement CRC in CRC-8 mode and get the CRC checksum result.

CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.

Analog-to-Digital Converter (ADC)

ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_MeasureAVDD	Measure AVDD voltage by ADC.
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.
ACMP_Wakeup	Show how to wake up MCU from Power-down mode by ACMP wake-up function.

PS/2 Controller (PS/2)

PS2	Demonstrate how to control PS/2 mouse movement on the screen.

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.