

## NUC100 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

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

## 2 Library Information

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

### 3 Sampel Code Information

<b>\SampleCode\CardReader</b>	CCID <sup>[1]</sup> Smart Card reader Sample Code.
<b>\SampleCode\Hard_Fault_Sample</b>	Show hard fault information when hard fault happened.
<b>\SampleCode\ISP</b>	Sample codes for In-System-Programming.
<b>\SampleCode\Template</b>	Software Development Template.
<b>\SampleCode\Semihost</b>	Show how to debug with semi-host message print.
<b>\SampleCode\RegBased</b>	The sample codes which access control registers directly.
<b>\SampleCode\StdDriver</b>	NUC100 Series Driver Samples

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

## 4 \SampleCode\ISP

<b>ISP_DFU</b>	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
<b>ISP_HID</b>	In-System-Programming Sample code through USB HID interface.
<b>ISP_I2C</b>	In-System-Programming Sample code through I2C interface.
<b>ISP_RS485</b>	In-System-Programming Sample code through RS485 interface.
<b>ISP_SPI</b>	In-System-Programming Sample code through SPI interface.
<b>ISP_UART</b>	In-System-Programming Sample code through UART interface.

## 5 \SampleCode\RegBased

<b>ACMP</b>	Demonstrate how ACMP <sup>[1]</sup> works with internal band-gap voltage.
<b>ACMP_Wakeup</b>	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
<b>ADC_ContinuousScanMode</b>	Perform A/D Conversion with ADC continuous scan mode.
<b>ADC_MeasureAVDD</b>	Measure AVDD voltage by ADC.
<b>ADC_PwmTrigger</b>	Demonstrate how to trigger ADC by PWM.
<b>ADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>ADC_SingleCycleScanMode</b>	Perform A/D Conversion with ADC single cycle scan mode.
<b>ADC_SingleMode</b>	Perform A/D Conversion with ADC single mode.
<b>CRC_8</b>	Implement CRC in CRC-8 mode and get the CRC checksum result.
<b>CRC_CCITT</b>	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
<b>FMC_IAP</b>	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.
<b>FMC_MultiBoot_SwReset</b>	Show how to use software reset to implement multi-boot system to boot from different applications in APROM.
<b>FMC_RW</b>	Show how to read/program embedded flash by ISP function.
<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and debounce function.
<b>GPIO_INT</b>	Show the usage of GPIO interrupt function.
<b>GPIO_OutputInput</b>	Show how to set GPIO pin mode and use pin data input/output control.

<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.
<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<b>I2C_GCMode_Master</b>	Show how a Master uses I <sup>2</sup> C address 0x0 to write data to Slave. This sample code needs to work with I2C_GCMode_Slave.
<b>I2C_GCMode_Slave</b>	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
<b>I2C_Master</b>	Show a I <sup>2</sup> C Master how to access Slave. This sample code needs to work with I2C_Slave.
<b>I2C_Slave</b>	Show how to set I <sup>2</sup> C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with I2C_Wakeup_Master.
<b>I2S_Master</b>	Configure I <sup>2</sup> S as Master mode and demonstrate how I <sup>2</sup> S works in Master mode. This sample code needs to work with I2S_Slave sample code.
<b>I2S_PDMA</b>	Demonstrate how I2S works with PDMA in Master mode. Both TX PDMA function and RX PDMA function will be enabled.
<b>I2S_Slave</b>	Configure I <sup>2</sup> S as Slave mode and demonstrate how I <sup>2</sup> S works in Slave mode. This sample code needs to work with I2S_Master sample code.
<b>PDMA</b>	Use PDMA channel 6 to transfer data from memory to memory.
<b>PS2</b>	Demonstrate how to emulate a PS/2 mouse by moving mouse pointer when connecting to PC by PS/2 interface.

<b>PWM_Capture</b>	Capture the PWMB Channel 1 waveform by PWMB Channel 2.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM Dead Zone function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM Double Buffer function.
<b>RTC_PowerDown</b>	Use RTC alarm interrupt event to wake-up system.
<b>RTC_TimeAndTick</b>	Get the current RTC data/time per tick.
<b>SPI_Loopback</b>	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.
<b>SPI_MasterFifoMode</b>	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.
<b>SPI_PDMA_Loopback</b>	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.
<b>SPI_SlaveFifoMode</b>	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.
<b>SYS</b>	Change system clock to different PLL frequency and output system clock from CLK0 pin.
<b>TIMER_Capture</b>	Show how to use the timer2 capture function to capture timer2 counter value.
<b>TIMER_Counter</b>	Implement timer1 event counter function to count the external input event.
<b>TIMER_PeriodicINT</b>	Implement timer counting in periodic mode.
<b>TIMER_PowerDown</b>	Use timer0 toggle-output time-out interrupt event to wake



	up system.
<b>UART_Autoflow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
<b>UART_Autoflow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
<b>UART_LIN</b>	Transmit LIN Frame including header and response in UART LIN mode.
<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system form Power-down mode by UART interrupt.
<b>WDT_PowerDown</b>	Use WDT time-out interrupt event to wake-up system.
<b>WDT_TimeoutINT</b>	Implement periodic WDT time-out interrupt event.
<b>WDT_TimeoutReset</b>	Show how to generate time-out reset system event while WDT time-out reset delay period expired.
<b>WWDT_CompareINT</b>	Show how to reload the WWDT counter value.

1. Analog Comparator (ACMP).

## 6 \SampleCode\StdDriver

<b>ACMP</b>	Demonstrate how ACMP works with internal band-gap voltage.
<b>ACMP_Wakeup</b>	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
<b>ADC_ContinuousScanMode</b>	Perform A/D Conversion with ADC continuous scan mode.
<b>ADC_MeasureAVDD</b>	Measure AVDD voltage by ADC.
<b>ADC_PwmTrigger</b>	Demonstrate how to trigger ADC by PWM.
<b>ADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>ADC_SingleCycleScanMode</b>	Perform A/D Conversion with ADC single cycle scan mode.
<b>ADC_SingleMode</b>	Perform A/D Conversion with ADC single mode.
<b>CRC_8</b>	Implement CRC in CRC-8 mode and get the CRC checksum result.
<b>CRC_CCITT</b>	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
<b>FMC_IAP</b>	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.
<b>FMC_RW</b>	Show how to read/program embedded flash by ISP function.
<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and de-bounce function.
<b>GPIO_INT</b>	Show the usage of GPIO interrupt function.
<b>GPIO_OutputInput</b>	Show how to set GPIO pin mode and use pin data input/output control.

<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.
<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<b>I2C_GCMode_Master</b>	Show how a Master uses I <sup>2</sup> C address 0x0 to write data to Slave. This sample code needs to work with I2C_GCMode_Slave.
<b>I2C_GCMode_Slave</b>	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
<b>I2C_Master</b>	Show a I <sup>2</sup> C Master how to access Slave. This sample code needs to work with I2C_Slave.
<b>I2C_Slave</b>	Show how to set I <sup>2</sup> C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with I2C_Wakeup_Master.
<b>I2S_Master</b>	Configure I <sup>2</sup> S as Master mode and demonstrate how I <sup>2</sup> S works in Master mode. This sample code needs to work with I2S_Slave sample code.
<b>I2S_PDMA</b>	Demonstrate how I2S works with PDMA in Master mode. Both TX PDMA function and RX PDMA function will be enabled.
<b>I2S_Slave</b>	Configure I <sup>2</sup> S as Slave mode and demonstrate how I <sup>2</sup> S works in Slave mode. This sample code needs to work with I2S_Master sample code.
<b>PDMA</b>	Use PDMA channel 6 to transfer data from memory to memory.
<b>PS2</b>	Demonstrate how to emulate a PS/2 mouse by moving mouse pointer when connecting to PC by PS/2 interface.

<b>PWM_Capture</b>	Capture the PWMB Channel 1 waveform by PWMB Channel 2.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM Dead Zone function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM Double Buffer function.
<b>RTC_PowerDown</b>	Use RTC alarm interrupt event to wake-up system.
<b>RTC_TimeAndTick</b>	Get the current RTC data/time per tick.
<b>SC_ReadATR</b>	Read the smartcard ATR from smartcard 0 interface.
<b>SPI_Loopback</b>	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.
<b>SPI_MasterFifoMode</b>	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.
<b>SPI_PDMA_Loopback</b>	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.
<b>SPI_SlaveFifoMode</b>	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.
<b>SYS</b>	Change system clock to different PLL frequency and output system clock from CLKO pin..
<b>TIMER_Capture</b>	Show how to use the timer2 capture function to capture timer2 counter value.
<b>TIMER_Counter</b>	Implement timer1 event counter function to count the external input event.
<b>TIMER_PeriodicINT</b>	Implement timer counting in periodic mode.

<b>TIMER_PowerDown</b>	Use timer-0 toggle-output interrupt event to wake-up system.
<b>UART_AutoFlow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_AutoFlow_Slave.
<b>UART_AutoFlow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_AutoFlow_Master.
<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
<b>UART_LIN</b>	Transmit LIN Frame including header and response in UART LIN mode.
<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.
<b>USBD_Audio_NAU8822</b>	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.
<b>USBD_HID_Keyboard</b>	Show how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.

### **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.*