

NUC121/NUC125 Series BSP Directory

Directory Introduction for 32-bit NuMicro™ Family

Directory Information

Document	Driver reference manual and reversion 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 and microprocessor 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

1 Document Information

CMSIS.html	Document of CMSIS version 5.1.1.
NuMicro NUC121_125 Series CMSIS BSP Driver Reference.chm	This document shows the usage of drivers in NUC121/NUC125 BSP.
NuMicro NUC121_125 Series CMSIS BSP Revision History.pdf	This document shows the revision history of NUC121/NUC125 BSP.

2 Library Information

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V5.1.1 definitions by Arm® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.

3 Sample Code Information

Hard_Fault_Sample	<p>Show hard fault information when hard fault happened.</p> <p>The hard fault handler shows some information including program counter, which is the address where the processor is executed when the hard fault occurs. The listing file (or map file) can show what function and instruction that is.</p> <p>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.</p>
ISP	Sample code for Nuvoton NuMicro ISP Programming Tool.
NuTiny-SDK-NUC121_125	Sample code for NUC121/NUC125 Tiny Board.
RegBased	Sample codes which access control registers directly.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Demonstrate the usage of NUC121/NUC125 series MCU peripheral driver APIs.
Template	A project template for NUC121/NUC125 series MCU.

4 \SampleCode\ISP

ISP_DFU	Sample ISP firmware communicated with DFU (Device Firmware Upgrade) tool through a USB DFU interface.
ISP_HID	Sample ISP firmware communicated with ISP tool through a USB HID interface.
ISP_I2C	Sample ISP firmware communicated with ISP tool through an I ² C interface.
ISP_MSC	Sample ISP firmware communicated with ISP tool through a USB MSC interface.
ISP_RS485	Sample ISP firmware communicated with ISP tool through a RS485 interface.
ISP_SPI	Sample ISP firmware communicated with ISP tool through a SPI interface.
ISP_UART	Sample ISP firmware communicated with ISP tool through a UART interface.

5 \SampleCode\NuTiny-SDK-NUC121_125

LED_Toggle

Toggle PB.4 to turn on / off the board LED.

6 \SampleCode\RegBased

ADC_BurstMode	Use burst mode to finish more than 16 ADC conversions for the specified channel.
ADC_ContinuousScanMode	Use continuous scan mode to finish two cycles of ADC conversion for the specified channels.
ADC_MeasureAVDD	Use band-gap voltage to calculate AVDD voltage.
ADC_PDMA_SingleCycleScanMode	Use single-cycle scan mode to finish one cycle of ADC conversion for 4 specified channels and transfer conversion results by PDMA.
ADC_PWMTrigger	Trigger single mode ADC by PWM0 channel0.
ADC_ResultMonitor	Use digital compare function to monitor the ADC conversion result of channel2.
ADC_SingleCycleScanMode	Use single-cycle scan mode to finish one cycle of ADC conversion for 4 specified channels.
ADC_SingleMode	Use single mode to finish the ADC conversion of the specified channel.
BPWM_Capture	Use BPWM1 channel2 to capture BPWM0 channel0 waveform.
BPWM_DoubleBuffer_PeriodLoadingMode	Use BPWM0 channel0~5 and double buffering function to output waveforms. This sample code changes duty cycle of each channel and the period shared by 6 channels.
BPWM_DutySwitch	Change duty cycle of BPWM output waveform by configured period.
BPWM_OutputWaveform	Use BPWM counter to output waveform.
BPWM_SyncStart	Use BPWM counter synchronous start function.

CLK_ClockDetector	Show how to use clock fail detector for HXT / LXT and clock frequency monitor for HXT.
CLK_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
FMC_ExecInSRAM	Run code in SRAM by using scatter loading description file.
FMC_IAP	Switch both codes between LDROM and APROM.
FMC_RW	Read and program embedded flash by ISP function.
FMC_SPROM	Show how to change SPROM to secured mode to protect SPROM contents.
GPIO_EINTAndDebounce	Use GPIO external interrupt function.
GPIO_INT	Use GPIO interrupt function.
GPIO_OutputInput	Set GPIO pin mode and use pin data input / output control.
GPIO_PowerDown	Wake up form Power-down mode by GPIO interrupt.
HIRC_Trim¹	Show how to trim HIRC by LXT.
I2C_EEPROM	Access EEPROM by an I ² C interface.
I2C_GCMode_Master	I ² C master uses I ² C address 0x00 to write data to slave. Need to work with the I2C_GCMode_Slave sample code.
I2C_GCMode_Slave	I ² C slave receives master data in General Call mode. Need to work with the I2C_GCMode_Master sample code.

¹ This sample requires LXT on board.

I2C_Master	Demonstrate how an I ² C master accesses slave. Need to work with the I2C_Slave sample code.
I2C_MultiBytes_Master	Use I ² C multiple-byte function to read and write data to slave. Need to work with the I2C_Slave sample code.
I2C_PDMA_TRX	Demonstrate I ² C PDMA mode, which need to connect I ² C0 (master) and I ² C1 (slave).
I2C_SingleByte_Master	Use I ² C single-byte function to read and write data to slave. Need to work with the I2C_Slave sample code.
I2C_Slave	Set I ² C in slave mode to receive the data of master. Need to work with the I2C_Master, I2C_MultiBytes_Master or I2C_SingleByte_Master sample codes.
I2C_Wakeup_Master	Show how to wake up I ² C slave from Power-down mode. Need to work with the I2C_Wakeup_Slave sample code.
I2C_Wakeup_Slave	Show how to wake up system in Power-down mode through an I ² C interface. Need to work with the I2C_Wakeup_Master sample code.
I2S_Master	Demonstrate how I ² S works in master mode. Need to work with the I2S_Slave sample code.
I2S_PDMA_NAU8822	I ² S playback and record with NAU8822 codec by PDMA with scatter-gather mode
I2S_PDMA_Play	I ² S playback by PDMA with scatter-gather mode
I2S_PDMA_PlayRecord	I ² S playback and record by PDMA with scatter-gather mode
I2S_PDMA_Record	I ² S record by PDMA with scatter-gather mode

I2S_Slave	Demonstrate how I ² S works in slave mode. Need to work with the I2S_Master sample code.
PDMA	Use PDMA channel0 to transfer data from memory to memory.
PDMA_Scatter_Gather	Use PDMA channel4 to transfer data from memory to memory with scatter-gather mode.
PDMA_ScatterGather_PingPongBuffer	Use PDMA to implement ping-pong buffer from memory to memory with scatter-gather mode.
PWM_Capture	Use PWM1 channel2 to capture PWM1 channel0 waveform.
PWM_DeadZone	Use PWM dead zone function.
PWM_DoubleBuffer_PeriodLoadingMode	Use PWM0 channel0~1 and double buffering function to output waveforms. This sample code changes duty cycle of each channel and the period shared by 2 channels.
PWM_DutySwitch	Change duty cycle of PWM output waveform by configured period.
PWM_OutputWaveform	Use PWM counter to output waveform.
PWM_PDMA_Capture	Capture PWM1 channel0 waveform by PWM1 channel2, and use PDMA to transfer captured data.
PWM_SyncStart	Use PWM counter synchronous start function.
SPI_MasterFIFOmode	Communicate with an off-chip SPI slave device with FIFO mode. Need to work with the SPI_SlaveFIFOmode sample code.
SPI_PDMA_LoopTest	SPI loopback test with PDMA.
SPI_SlaveFIFOmode	Communicate with an off-chip SPI master device with FIFO mode. Need to work with the SPI_MasterFIFOmode sample code.

SYS_BODWakeup	Wake up system from BOD event.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.
TIMER_CaptureCounter	Timer2 uses Timer3 toggle output to capture an event counter value from Timer0 toggle output.
TIMER_Delay	Use timer to implement delay cycle.
TIMER_EventCounter	Use Timer1 counter input function to count the input event.
TIMER_PeriodicINT	Perform timer counting in periodic mode.
TIMER_TimeoutWakeup	Use Timer0 toggle-output interrupt event to wake up system.
UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. Need to work with the UART_AutoBaudRate_Slave sample code.
UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. Need to work with the UART_AutoBaudRate_Master sample code.
UART_Autoflow_Master	Transmit and receive data with auto flow control. Need to work with the UART_Autoflow_Slave sample code.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. Need to work with the UART_Autoflow_Master sample code.
UART_IrDA_Master	Transmit and receive data in UART IrDA mode. Need to work with the UART_IrDA_Slave sample code.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. Need to work with the UART_IrDA_Master sample code.
UART_LIN	Transmit LIN frame including header and response in LIN function mode.

UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485_Master	Transmit and receive data in RS485 function mode. Need to work with the UART_RS485_Slave sample code.
UART_RS485_Slave	Transmit and receive data in RS485 function mode. Need to work with the UART_RS485_Master sample code.
UART_TxRx_Function	Demonstrate how UART transmits and receives data from PC terminal through a RS232 interface.
UART_Wakeup	Wake up system form Power-down mode by nCTS / data interrupt.
UART_Wakeup_LXT	Wake up system form Power-down mode by Rx threshold and time-out / RS485 interrupt. This sample requires LXT on board.
USCI_I2C_EEPROM	Access EEPROM by an I ² C interface.
USCI_I2C_GCMode_Master	I ² C master uses I ² C address 0x00 to write data to I ² C slave. Need to work with the USCI_I2C_GCMode_Slave sample code.
USCI_I2C_GCMode_Slave	I ² C slave receives master data in General Call mode. Need to work with the USCI_I2C_GCMode_Master sample code.
USCI_I2C_Master	Demonstrate how an I ² C master accesses slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Demonstrate how an I ² C master accesses 10-bit address slave. Need to work with the USCI_I2C_Slave_10bit sample code.
USCI_I2C_Monitor	Use UI2C to monitor and log I2C bus traffic.
USCI_I2C_MultiBytes_Master	Use UI2C multiple-byte function to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.

USCI_I2C_SingleByte_Master	Use I2C single-byte function to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Slave	Set USCI-I ² C in slave mode to receive the data of master. Need to work with the USCI_I2C_Master, USCI_I2C_MultiBytes_Master or USCI_I2C_SingleByte_Master sample codes.
USCI_I2C_Slave_10bit	Show a 10-bit address slave how to receive data from master. Need to work with the USCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Master	Show how to wake up I ² C slave from Power-down mode. Need to work with the USCI_I2C_Wakeup_Slave sample code.
USCI_I2C_Wakeup_Slave	Show how to wake up system in Power-down mode through an I ² C interface. Need to work with the USCI_I2C_Wakeup_Master sample code.
USCI_SPI_MasterMode	Communicate with an off-chip SPI slave device. Need to work with the USCI_SPI_SlaveMode sample code.
USCI_SPI_PDMA_LoopTest	USCI-SPI loop back test with PDMA.
USCI_SPI_SlaveMode	Communicate with an off-chip SPI master device. Need to work with the USCI_SPI_MasterMode sample code.
USCI_UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. Need to work with the USCI_UART_AutoBaudRate_Slave sample code.
USCI_UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. Need to work with the UACI_UART_AutoBaudRate_Master sample code.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. Need to work with the

	USCI_UART_Autoflow_Slave sample code.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. Need to work with the USCI_UART_Autoflow_Master sample code.
USCI_UART_PDMA	Transmit and receive UART data with the PDMA.
USCI_UART_RS485_Master	Transmit and receive data in RS485 function mode. Need to work with the USCI_UART_RS485_Slave sample code.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 function mode. Need to work with the USCI_UART_RS485_Master sample code.
USCI_UART_TxRx_Function	Demonstrate how USCI-UART transmits and receives data from PC terminal through a RS232 interface.
USCI_UART_Wakeup	Wake up system form Power-down mode by nCTS / data interrupt.
WDT_PowerDown	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	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.

7 \SampleCode\StdDriver

ADC_BurstMode	Use burst mode to finish more than 16 ADC conversions for the specified channel.
ADC_ContinuousScanMode	Use continuous scan mode to finish two cycles of ADC conversion for the specified channels.
ADC_MeasureAVDD	Use band-gap voltage to calculate AVDD voltage.
ADC_PDMA_SingleCycleScanMode	Use single-cycle scan mode to finish one cycle of ADC conversion for 4 specified channels and transfer conversion result by PDMA.
ADC_PWMTrigger	Trigger single mode ADC by PWM0 channel0.
ADC_ResultMonitor	Use digital compare function to monitor the ADC conversion result of channel2.
ADC_SingleCycleScanMode	Use single-cycle scan mode to finish one cycle of ADC conversion for 4 specified channels.
ADC_SingleMode	Use single mode to finish the ADC conversion of the specified channel.
BPWM_Capture	Use BPWM1 channel2 to capture BPWM0 channel0 waveform.
BPWM_DoubleBuffer_PeriodLoadingMode	Use BPWM0 channel0~5 and double buffering function to output waveforms. This sample code changes duty cycle of each channel and the period shared by 6 channels.
BPWM_DutySwitch	Change duty cycle of BPWM output waveform by configured period.
BPWM_OutputWaveform	Use BPWM counter to output waveform.
BPWM_SyncStart	Use BPWM counter synchronous start function.

CLK_ClockDetector	Show how to use clock fail detector for HXT / LXT and clock frequency monitor for HXT.
CLK_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
FMC_ExecInSRAM	Run code in SRAM by using scatter loading description file.
FMC_IAP	Switch both codes between LDROM and APROM.
FMC_RW	Read and program embedded flash by ISP function.
FMC_SPROM	Show how to change SPROM to secured mode to protect SPROM contents.
GPIO_EINTAndDebounce	Use GPIO external interrupt function.
GPIO_INT	Use GPIO interrupt function.
GPIO_OutputInput	Set GPIO pin mode and use pin data input / output control.
GPIO_PowerDown	Wake up form Power-down mode by GPIO interrupt.
I2C_EEPROM	Access EEPROM by an I ² C interface.
I2C_GCMode_Master	I ² C master uses I ² C address 0x00 to write data to slave. Need to work with I2C_GCMode_Slave sample code.
I2C_GCMode_Slave	I ² C slave receives master data in General Call mode. Need to work with I2C_GCMode_Master sample code.
I2C_Master	Demonstrate how an I ² C master accesses slave. Need to work with the I2C_Slave sample code.
I2C_MultiBytes_Master	Use I ² C multiple-byte API to read and write data to slave. Need to work with the I2C_Slave sample code.

I2C_PDMA_TRX	Demonstrate I ² C PDMA mode, which need to connect I ² C0 (master) and I ² C1 (slave).
I2C_SingleByte_Master	Use I ² C single-byte API to read and write data to slave. Need to work with the I2C_Slave sample code.
I2C_Slave	Set I ² C in slave mode to receive the data of master. Need to work with the I2C_Master, I2C_MultiBytes_Master or I2C_SingleByte_Master sample codes.
I2C_Wakeup_Master	Show how to wake up I ² C slave from Power-down mode. Need to work with the I2C_Wakeup_Slave sample code.
I2C_Wakeup_Slave	Show how to wake up system in Power-down mode through an I ² C interface. Need to work with the I2C_Wakeup_Master sample code.
I2S_Master	Demonstrate how I ² S works in master mode. Need to work with the I2S_Slave sample code.
I2S_PDMA_NAU8822	I ² S playback and record with NAU8822 codec by PDMA with scatter-gather mode
I2S_PDMA_Play	I ² S playback by PDMA with scatter-gather mode
I2S_PDMA_PlayRecord	I ² S playback and record by PDMA with scatter-gather mode.
I2S_PDMA_Record	I ² S record by PDMA with scatter-gather mode.
I2S_Slave	Demonstrate how I ² S works in slave mode. Need to work with the I2S_Master sample code.
PDMA	Use PDMA channel0 to transfer data from memory to memory.
PDMA_Scatter_Gather	Use PDMA channel4 to transfer data from memory to memory with scatter-gather mode.

PDMA_ScatterGather_PingPongBuffer	Use PDMA to implement ping-pong buffer from memory to memory with scatter-gather mode.
PWM_Capture	Use PWM1 channel2 to capture PWM1 channel0 waveform.
PWM_DeadZone	Use PWM dead zone function.
PWM_DoubleBuffer_PeriodLoadingMode	Use PWM0 channel0~1 and double buffering function to output waveforms. This sample code changes duty cycle of each channel and the period shared by 2 channels.
PWM_DutySwitch	Change duty cycle of PWM output waveform by configured period.
PWM_OutputWaveform	Use PWM counter to output waveform.
PWM_PDMA_Capture	Capture PWM1 channel0 waveform by PWM1 channel2, and use PDMA to transfer captured data.
PWM_SyncStart	Use PWM counter synchronous start function.
SPI_MasterFIFOmode	Communicate with an off-chip SPI slave device with FIFO mode. Need to work with the SPI_SlaveFIFOmode sample code.
SPI_PDMA_LoopTest	SPI loopback test with PDMA.
SPI_SlaveFIFOmode	Communicate with an off-chip SPI master device with FIFO mode. Need to work with the SPI_MasterFIFOmode sample code.
SYS_BODWakeup	Wake up system from BOD event.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.
TIMER_CaptureCounter	Timer2 uses Timer3 toggle output to capture an event counter value from Timer0 toggle output.

TIMER_Delay	Use timer to implement delay cycle.
TIMER_EventCounter	Use Timer1 counter input function to count the input event.
TIMER_PeriodicINT	Perform timer counting in periodic mode.
TIMER_TimeoutWakeup	Use Timer0 toggle-output interrupt event to wake up system.
UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. Need to work with the UART_AutoBaudRate_Slave sample code.
UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. Need to work with the UART_AutoBaudRate_Master sample code.
UART_Autoflow_Master	Transmit and receive data with auto flow control. Need to work with the UART_Autoflow_Slave sample code.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. Need to work with the UART_Autoflow_Master sample code.
UART_IrDA_Master	Transmit and receive data in UART IrDA mode. Need to work with the UART_IrDA_Slave sample code.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. Need to work with the UART_IrDA_Master sample code.
UART_LIN	Transmit LIN frame including header and response in LIN function mode.
UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485_Master	Transmit and receive data in RS485 function mode. Need to work with the UART_RS485_Slave sample code.
UART_RS485_Slave	Transmit and receive data in RS485 function mode. Need to work with the

	UART_RS485_Master sample code.
UART_TxRx_Function	Demonstrate how UART transmits and receives data from PC terminal through a RS232 interface.
UART_Wakeup	Wake up system form Power-down mode by nCTS / data interrupt.
UART_Wakeup_LXT	Wake up system form Power-down mode by Rx threshold and time-out / RS485 interrupt. This sample requires LXT on board.
USBD_Audio_HID_NAU8822	USB audio class device with HID key. 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_Audio_HID_Transfer	USB composite device with USB audio class and HID data transfer function.
USBD_Audio_NAU8822	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	USB keyboard device. It supports to use GPIO to simulate key input.
USBD_HID_Mouse	USB mouse device. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
USBD_HID_Mouse2	USB mouse device. It uses PC.0~5 to control mouse direction and mouse key. It also supports USB suspend and remote wakeup.
USBD_HID_MouseKeyboard	A USB mouse and a USB keyboard on the same USB device. The mouse cursor will move automatically when this mouse device connecting to PC. This sample code uses a GPIO pin to simulate key input.
USBD_HID_Transfer	Transfer data between USB device and PC through HID interface. A windows tool is

	also included in this sample code to connect with USB device.
USBD_HID_Transfer_and_Keyboard	USB composite device with HID Transfer and keyboard. Transfer data between USB device and PC through HID interface. A windows tool is also included in this sample code to connect with USB device.
USBD_HID_Transfer_and_MSC	USB composite device with HID Transfer and USB Mass-storage. Transfer data between USB device and PC through HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_MassStorage_CDRom	Simulate a USB CD-ROM device.
USBD_MassStorage_DataFlash	Use internal flash as backend storage media to simulate a USB pen drive. GCC project does not support small size flash due to larger code size.
USBD_Micro_Printer	USB micro printer device.
USBD_Printer_and_HID_Transfer	USB composite device with USB micro printer device and HID Transfer. Transfer data between USB device and PC through HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_and_HID_Keyboard	USB composite device with virtual COM port and keyboard.
USBD_VCOM_and_HID_Transfer	USB composite device with virtual COM port and HID Transfer. It supports one virtual COM port and transfers 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_VCOM_and_MassStorage	USB composite device with virtual COM port and USB Mass-Storage.
USBD_VCOM_DualPort	USB virtual COM port device. It supports

	dual virtual COM ports using UART and UART.
USBD_VCOM_SinglePort	USB virtual COM port device. It supports single virtual COM port.
USCI_I2C_EEPROM	Access EEPROM by an I ² C interface.
USCI_I2C_GCMode_Master	I ² C master uses I ² C address 0x00 to write data to I ² C slave. Need to work with the USCI_I2C_GCMode_Slave sample code.
USCI_I2C_GCMode_Master	I ² C slave receives master data in General Call mode. Need to work with the USCI_I2C_GCMode_Master sample code.
USCI_I2C_Master	Demonstrate how an I ² C master accesses slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Demonstrate how an I ² C master accesses 10-bit address slave. Need to work with the USCI_I2C_Slave_10bit sample code.
USCI_I2C_Monitor	Use UI2C to monitor and log I2C bus traffic.
USCI_I2C_MultiBytes_Master	Use UI2C multiple-byte API to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_SingleByte_Master	Use UI2C single-byte API to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Slave	Set USCI-I ² C in slave mode to receive the data of master. Need to work with the USCI_I2C_Master, USCI_I2C_MultiBytes_Master or USCI_I2C_SingleByte_Master sample codes.
USCI_I2C_Slave_10bit	Show a 10-bit address slave how to receive data from master. Need to work with the USCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Master	Show how to wake up I ² C slave from Power-down mode. Need to work with the

	USCI_I2C_Wakeup_Slave sample code.
USCI_I2C_Wakeup_Slave	Show how to wake up system in Power-down mode through an I ² C interface. Need to work with the USCI_I2C_Wakeup_Master sample code.
USCI_SPI_MasterMode	Communicate with an off-chip SPI slave device. Need to work with the USCI_SPI_SlaveMode sample code.
USCI_SPI_PDMA_LoopTest	USCI-SPI loop back test with PDMA.
USCI_SPI_SlaveMode	Communicate with an off-chip SPI master device. Need to work with the USCI_SPI_MasterMode sample code.
USCI_SPI_SlaveModeINT	Communicate with an off-chip SPI master device, transmit and receive data in the interrupt handler. This sample code needs to work with USCI_SPI_MasterMode sample code.
USCI_UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. Need to work with the USCI_UART_AutoBaudRate_Slave sample code.
USCI_UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. Need to work with the UACI_UART_AutoBaudRate_Master sample code.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. Need to work with the USCI_UART_Autoflow_Slave sample code.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. Need to work with the USCI_UART_Autoflow_Master sample code.
USCI_UART_PDMA	Transmit and receive UART data with PDMA.
USCI_UART_RS485_Master	Transmit and receive data in RS485 function mode. Need to work with the

	USCI_UART_RS485_Slave sample code.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 function mode. Need to work with the USCI_UART_RS485_Master sample code.
USCI_UART_TxRx_Function	Demonstrate how USCI-UART transmits and receives data from PC terminal through a RS232 interface.
USCI_UART_Wakeup	Wake up system form Power-down mode by nCTS / data interrupt.
WDT_PowerDown	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	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.

*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.*