M251/M252 Series 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

1 Document Information

CMSIS.html	Document of CMSIS version 5.1.1.
NuMicro M251_252 Series CMSIS BSP Revision History.pdf	This document shows the revision history of M251/M252 BSP.
NuMicro M251_252 Series CMSIS BSP Driver Reference.chm	This document describes the usage of drivers in M251/M252 BSP.

2 Library Information

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

3 Sample Code Information

CardReader	USB CCID Smartcard Reader 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 code for Nuvoton NuMicro ISP Programming Tool.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of M251/M252 MCU peripheral driver APIs.
Template	A project template for M251/M252.
XOM	Demonstrate how to create XOM library and use it.

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^2C 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\StdDriver

ACMP_ComapreDAC	Demonstrate ACMP comparison by comparing ACMP0_P0 input and DAC voltage and shows the result on UART console.
ACMP_ComapreVBG	Demonstrate ACMP comparison by comparing ACMP0_P0 input and VBG voltage and shows the result on UART console.
ACMP_Wakeup	Use ACMP to wake up system from Power-down mode while comparator output changes.
ACMP_WindowCompare	Show how to monitor ACMP input with window compare function.
ACMP_WindowLatch	Demonstrate how to use ACMP window latch mode.
BPWM_Capture	Use BPWM0 channel 0 (PA.0) to capture the BPWM1 channel 0 (PE.13) waveform.
BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM double buffer function.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SwitchDuty	Change duty cycle of output waveform by configured period.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.
CLK_ClockDetector	Demonstrate the usage of clock fail detector and clock frequency range detector function.
CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Implement CRC in CRC-8 mode and get the CRC checksum result.
CRC_CRC32	Implement CRC in CRC-32 mode and get the CRC checksum result.

DAC_ExtPinTrigger	Demonstrate external pin trigger DAC convert sine wave outputs.
DAC_PDMA_TimerTrigger	Show Timer trigger DAC to fetch data with PDMA and convert sine wave outputs.
DAC_SoftwareTrigger	Demonstrate software trigger DAC to convert sine wave outputs.
DAC_TimerTrigger	Demonstrate Timer trigger DAC to convert sine wave outputs.
EADC_Accumulate	Demonstrate how to get accumulate conversion result.
EADC_ADINT_Trigger	Use ADINT interrupt to trigger the EADC conversion.
EADC_Average	Demonstrate how to get average conversion result.
EADC_BandGap	Convert band-gap (Sample module 16) and print conversion result.
EADC_OffsetCancel	Demonstrate how to modify final EADC conversion result by offset cancellation.
EADC_PDMA_PWM_Trigger	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
EADC_Pending_Priority	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
EADC_PWM_Trigger	Demonstrate how to trigger EADC by PWM.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Trigger EADC by writing EADC_SWTRG register.
EADC_TempSensor	Convert temperature sensor (Sample module 17) and print conversion result.
EADC_Timer_Trigger	Show how to trigger EADC by Timer.
EADC_VBat	Convert VBAT/4 (Sample module 18) and print conversion result.

EBI_NOR	Configure EBI interface to access NOR Flash connects on EBI interface.
EBI_SRAM	Configure EBI interface to access SRAM connects on EBI interface.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash.
FMC_IAP	Demonstrate FMC IAP boot mode and show how to use vector remap function. LDROM image was embedded in APROM image and be programmed to LDROM Flash at run-time. This sample also shows how to branch between APROM and LDROM.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM or LDROM by VECMAP.
FMC_MultiWordProgram	Show FMC multi-word program ISP command to program APROM 0x18000~0x20000 area.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify APROM or LDROM pages are all 0xFFFFFFFF or not.
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.
FMC_XOM	This sample code shows how to configure and setup an XOM region then perform XOM function.
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 and output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.

I2C_EEPROM	Read and write EEPROM via 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_Loopback	Demonstrate how a master accesses slave.
I2C_Master	An I²C master mode demo code. This sample code needs to work with I2C_Slave sample code.
I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with I2C_Slave.
I2C_PDMA_TRX	Demonstrate I²C PDMA mode, which need to connect I²C0 (master) and I²C1 (slave).
I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with I2C_Slave.
I2C_Slave	An I²C slave mode demo code.
I2C_Wakeup_Slave	Demonstrate how to set I²C to wake up MCU from Power-down mode. This sample code needs to work with I2C_Master.
OPA_Control	Show how to control OPA.
PDMA_BasicMode	Use PDMA channel 2 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA channel 4 to transfer data from memory to memory by scatter-gather mode.
PDMA_ScatterGather_ PingPongBuffer	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).
PSIO_1Wire	Use PSIO to access MAXIM DS18B20 digital thermometer.

PSIO_DMX512	Use PSIO to implement DMX512 protocol.
PSIO_HDQ	Use PSIO to access TI BQ2028 EEPROM.
PSIO_IR	Use PSIO to implement NEC IR protocol.
PSIO_LED	Use PSIO to control Worldsemi WS2812 LED.
PSIO_Microwire	Use PSIO to access Atmel T93C46D EEPROM.
PSIO_PS2_Device	Use PSIO to implement PS/2 device.
PSIO_PS2_Host	Use PSIO to implement PS/2 host.
PSIO_Wiegand	Use PSIO to access HZ1050 RFID reader.
PWM_Brake	Demonstrate how to use PWM brake function.
PWM_Capture	Capture the PWM1 channel 0 waveform by PWM1 channel 2.
PWM_DeadTime	Demonstrate how to use PWM dead-time insertion function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM double buffer function.
PWM_OutputWaveform	Demonstrate how to use PWM output waveform.
PWM_PDMA_Capture	Capture the PWM1 channel 0 waveform by PWM1 channel 2, and use PDMA to transfer captured data.
PWM_SwitchDuty	Change duty cycle of PWM output waveform by configured period.
PWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.
QSPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
QSPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.
QSPI_Slave3Wire	Configure QSPI0 as Slave 3 wire 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.
RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Use RTC alarm interrupt event to wake up system.
RTC_Spare_Access	Show how to access RTC spare registers in supported chip.
RTC_Static_Tamper	Show how to use RTC static tamper function in supported chip.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.
SC_ReadATR	Read the smartcard ATR from Smartcard interface.
SC_ReadSimPhoneBook	Demonstrate how to read phone book information in the SIM card.
SC_Timer	Demonstrate how to use SC embedded timer.
SCUART_TxRx	Demonstrate Smartcard UART mode by connecting PB.4 and PB.5 pins.
SPI_Flash	Access SPI Flash through SPI interface.
SPI_HalfDuplex	Demonstrate SPI half-duplex mode. Configure SPI0 as master mode and SPI1 as slave mode. Both SPI0 and SPI1 are half-duplex mode.
SPI_Loopback	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
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_LoopTest	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as slave mode and QSPI0 will be configured as master 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.
SPII2S_Master	Configure SPI0 as I²S master mode and demonstrate how I²S works in master mode. This sample code needs to work with SPII2S_Slave sample code.
SPII2S_PDMA_Codec	An I²S demo with PDMA function connected with audio codec.
SPII2S_PDMA_Play	An I²S demo for playing data and demonstrating how I²S works with PDMA.
SPII2S_PDMA_PlayRecord	An I²S demo for playing and recording data with PDMA function.
SPII2S_PDMA_Record	An I²S demo for recording data and demonstrating how I²S works with PDMA.
SPII2S_Slave	Configure SPI0 as I²S slave mode and demonstrate how I²S works in slave mode. This sample code needs to work with SPII2S_Master sample code.
SYS_BODWakeup	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.
SYS_DPDMode_Wakeup	Demonstrate how to wake up system from Deep Power-down mode by Wake-up pin (PA.0), Wake-up Timer, RTC Tick, RTC Alarm, or RTC Tamper 0.
SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_TrimHIRC	Demonstrate how to use LXT to trim HIRC.
SYS_TrimMIRC	Demonstrate how to use Timer to trim MIRC.
TIMER_ACMPTrigger	Use ACMP to trigger Timer reset mode.
TIMER_CaptureCounter	Show how to use the Timer2 capture function to capture Timer2 counter value.

TIMER_Delay	Demonstrate the usage of TIMER_Delay( ) API to generate a 1 second delay.
TIMER_EventCounter	Use pin PB.4 to demonstrates Timer event counter function.
TIMER_FreeCountingMode	Use the Timer pin PA.11 to demonstrate Timer free counting mode function, and display the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the Timer pin PB.5 to demonstrate inter-timer trigger mode function, and display the measured input frequency to UART console.
TIMER_Periodic	Use the Timer periodic mode to generate Timer interrupt every 1 second.
TIMER_PeriodicINT	Implement Timer counting in periodic mode.
TIMER_PWM_ChangeDuty	Change duty cycle and period of output waveform in PWM down count type.
TIMER_PWM_OutputWaveform	Demonstrate output different duty waveform in Timer0~3 PWM.
TIMER_TimeoutWakeup	Use Timer0 periodic time-out interrupt event to wake up system.
TIMER_ToggleOut	Demonstrate the Timer0 toggle out function on pin PB.5.
UART_AutoBaudRate	Show how to use auto baud rate detection function.
UART_AutoFlow	Transmit and receive data using auto flow control.
UART_IrDA	Transmit and receive UART data in UART IrDA mode.
UART_LIN	Demonstrate how to send data to LIN bus.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_SingleWire	Transmit and receive data in UART single-wire mode.

UART_TxRxFunction	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.
USBD_Audio_Codec	Demonstrate how to implement a USB audio class device.
USBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
USBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
USBD_HID_MouseKeyboard	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
USBD_HID_RemoteWakeup	Simulate a HID mouse supports USB suspend and remote wakeup.
USBD_HID_Transfer	Demonstrate how to transfer data between a USB device and PC through a USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_And_ Keyboard	Demonstrate how to implement a composite device of HID transfer and keyboard. Transfer 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_HID_Transfer_And_ MSC	Demonstrate how to implement a composite device of HID transfer and mass storage. Transfer 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_HID_Transfer_CTRL	Use USB host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.

USBD_Mass_Storage_CDROM	Demonstrate the emulation of USB mass storage device, CD-ROM.
USBD_Mass_Storage_Flash	Use internal flash as backend storage media to simulate a USB pen drive.
USBD_Mass_Storage_SRAM	Use internal SRAM as backend storage media to simulate a USB pen drive.
USBD_Micro_Printer	Demonstrate how to implement a USB micro printer device.
USBD_Printer_And_HID_ Transfer	Demonstrate how to implement a composite device of USB micro printer and HID transfer. Transfer 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_HID_ Keyboard	Demonstrate how to implement a composite device of VCOM and HID keyboard.
USBD_VCOM_And_HID_ Transfer	Demonstrate how to implement a composite device of VCOM and HID transfer. Transfer 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_Mass_ Storage	Demonstrate how to implement a composite device of VCOM and mass storage.
USBD_VCOM_DualPort	Demonstrate how to implement a USB dual virtual COM port device.
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual COM port device.
USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Lookback	Show an I²C master how to access 7-bit address slave via loopback of 2 USCI ports.
USCI_I2C_Loopback_10bit	Show an I²C master how to access 10-bit address slave

	via loopback of 2 USCI ports.
USCI_I2C_Master	Show an I²C master how to access 7-bit address slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Show an I²C master how to access 10-bit address slave. This sample code needs to work with USCI_I2C_Slave_10bit sample code.
USCI_I2C_Monitor	Use USCI_I2C to monitor and log I2C bus traffic.
USCI_I2C_MultiBytes_Master	Use UI2C multiple-byte functions 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 functions to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Slave	Show an I²C 7-bit address slave how to receive data from master.
USCI_I2C_Slave_10bit	Show an I²C 10-bit address slave how to receive data from master. This sample code needs to work with USCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Slave	Demonstrate how to set I²C to wake up MCU from Power-down mode. This sample code needs to work with USCI_I2C_Master sample code.
USCI_SPI_Loopback	Implement USCI_SPI0 master loop back transfer. This sample code needs to connect USCI_SPI0_MISO pin and USCI_SPI0_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI0 as master mode and demonstrate how to communicate with an off-chip SPI Slave device. Needs to work with USCI_SPI_SlaveMode sample code.
USCI_SPI_PDMA_LoopTest	Demonstrate SPI data transfer with PDMA. USCI_SPI0 will be configured as master mode and USCI_SPI1 will be configured as slave mode. Both Tx PDMA function and Rx PDMA function will be enabled.

USCI_SPI_SlaveMode	Configure USCI_SPI0 as slave mode and demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with USCI_SPI_MasterMode sample code.
USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Slave sample code.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Master sample code.
USCI_UART_PDMA	This is a USCI_UART PDMA demo and need to connect USCI_UART Tx and Rx.
USCI_UART_RS485_Master	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Slave sample code.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Master sample code.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.
WDT_TimeoutWakeupAndReset	Implement WDT time-out interrupt event to wake up system and generate time-out reset system event while WDT time-out reset delay period expired.
WWDT_CompareINT	Show how to reload the WWDT counter value.

6 \SampleCode\XOM

XOMLib	Demonstrate how to create XOM library.
XOMLibDemo	Demonstrate how to use XOMLib.

7 Sample Code Compatibility List

Category	C	D	E	G
Part Number / Sample Code	M251EC2AE M251FC2AE M251ZC2AE M252EC2AE M252FC2AE M252ZC2AE	M251LC2AE M251LD2AE M251SC2AE M251SD2AE M251ZD2AE M252LC2AE M252LD2AE M252SC2AE M252SD2AE M252ZD2AE	M251KE3AE M251LE3AE M251SE3AE M252KE3AE M252LE3AE M252SE3AE	M251KG6AE M251LG6AE M251SG6AE M252KG6AE M252LG6AE M252SG6AE
ACMP_ComapreDAC	-	-	-	√
ACMP_ComapreVBG	-	√	√	√
ACMP_Wakeup	-	√	√	√
ACMP_WindowCompare	-	√	√	√
ACMP_WindowLatch	-	√	√	√
BPWM_Capture	-	√	√	√
BPWM_DoubleBuffer	-	√	√	√
BPWM_OutputWaveform	-	√	√	√
BPWM_SwitchDuty	-	√	√	√
BPWM_SyncStart	-	√	√	√
CLK_ClockDetector[1]	√	√	√	√
CRC_CCITT	√	√	√	√
CRC_CRC8	√	√	√	√

[1] CLK_ClockDetector requires HXT.

Category	C	D	E	G
CRC_CRC32	√	√	√	√
DAC_ExtPinTrigger	-	-	-	√
DAC_PDMA_TimerTrigger	-	-	-	√
DAC_SoftwareTrigger	-	-	-	√
DAC_TimerTrigger	-	-	-	√
EADC_Accumulate	√	√	√	√
EADC_ADINT_Trigger	√	√	√	√
EADC_Average	√	√	√	√
EADC_BandGap	√	√	√	√
EADC_OffsetCancel	√	√	√	√
EADC_PDMA_PWM_Trigger	√	√	√	√
EADC_Pending_Priority	√	√	√	√
EADC_PWM_Trigger	√	√	√	√
EADC_ResultMonitor	√	√	√	√
EADC_SWTRG_Trigger	√	√	√	√
EADC_TempSensor	√	√	√	√
EADC_Timer_Trigger	√	√	√	√
EADC_VBat	-	-	√	√
EBI_NOR	-	-	√	√
EBI_SRAM	-	-	√	√
FMC_CRC32	√	√	√	√
FMC_ExeInSRAM	√	√	√	√
FMC_IAP	√	√	√	√
FMC_MultiBoot	√	√	√	√

Category	C	D	E	G
FMC_MultiWordProgram	√	√	√	√
FMC_ReadAllOne	√	√	√	√
FMC_RW	√	√	√	√
FMC_XOM	√	√	√	√
GPIO_EINTAndDebounce	√	√	√	√
GPIO_INT	√	√	√	√
GPIO_OutputInput	√	√	√	√
GPIO_PowerDown	√	√	√	√
I2C_EEPROM	√	√	√	√
I2C_GCMode_Master	√	√	√	√
I2C_GCMode_Slave	√	√	√	√
I2C_Loopback	√	√	√	√
I2C_Master	√	√	√	√
I2C_MultiBytes_Master	√	√	√	√
I2C_PDMA_TRX	√	√	√	√
I2C_SingleByte_Master	√	√	√	√
I2C_Slave	√	√	√	√
I2C_Wakeup_Slave	√	√	√	√
OPA_Control	-	-	-	√
PDMA_BasicMode	√	√	√	√
PDMA_ScatterGather	√	√	√	√
PDMA_ScatterGather_PingPongBuffer	√	√	√	√
PSIO_1Wire	-	√	√	√
PSIO_DMX512	-	√	√	√

Category	C	D	E	G
PSIO_HDQ	-	√	√	√
PSIO_IR	-	√	√	√
PSIO_LED	-	√	√	√
PSIO_Microwire	-	√	√	√
PSIO_PS2_Device	-	√	√	√
PSIO_PS2_Host	-	√	√	√
PSIO_Wiegand	-	√	√	√
PWM_Brake	√	√	√	√
PWM_Capture	√	√	√	√
PWM_DeadTime	√	√	√	√
PWM_DoubleBuffer	√	√	√	√
PWM_OutputWaveform	√	√	√	√
PWM_PDMA_Capture	√	√	√	√
PWM_SwitchDuty	√	√	√	√
PWM_SyncStart	√	√	√	√
QSPI_DualMode_Flash	√	√	√	√
QSPI_QuadMode_Flash	√	√	√	√
QSPI_Slave3Wire	√	√	√	√
RTC_Alarm_Test	√	√	√	√
RTC_Alarm_Wakeup	√	√	√	√
RTC_Spare_Access	-	√	√	√
RTC_Static_Tamper	-	√	√	√
RTC_Time_Display	√	√	√	√
SC_ReadATR	√	√	√	√

Category	C	D	E	G
SC_ReadSimPhoneBook	√	√	√	√
SC_Timer	√	√	√	√
SCUART_TxRx	√	√	√	√
SPI_Flash	-	√	√	√
SPI_HalfDuplex	-	√	√	√
SPI_Loopback	-	√	√	√
SPI_MasterFIFOMode	-	√	√	√
SPI_PDMA_LoopTest	-	√	√	√
SPI_SlaveFIFOMode	-	√	√	√
SPII2S_Master	-	√	√	√
SPII2S_PDMA_Codec	-	√	√	√
SPII2S_PDMA_Play	-	√	√	√
SPII2S_PDMA_PlayRecord	-	√	√	√
SPII2S_PDMA_Record	-	√	√	√
SPII2S_Slave	-	√	√	√
SYS_BODWakeup	√	√	√	√
SYS_DPDMode_Wakeup[2]	√	√	√	√
SYS_PLLClockOutput	-	√	√	√
SYS_TrimHIRC	√	√	√	√
SYS_TrimMIRC	√	√	√	√
TIMER_ACMPTrigger	-	√	√	√
TIMER_CaptureCounter	√	√	√	√

[2] SYS_DPDMode_Wakeup does not support tamper pin wakeup function in Category-C parts.

Category	C	D	E	G
TIMER_Delay	√	√	√	√
TIMER_EventCounter	√	√	√	√
TIMER_FreeCountingMode	√	√	√	√
TIMER_InterTimerTriggerMode	√	√	√	√
TIMER_Periodic	√	√	√	√
TIMER_PeriodicINT	√	√	√	√
TIMER_PWM_ChangeDuty	√	√	√	√
TIMER_PWM_OutputWaveform	√	√	√	√
TIMER_TimeoutWakeup	√	√	√	√
TIMER_ToggleOut	√	√	√	√
UART_AutoBaudRate	√	√	√	√
UART_AutoFlow	√	√	√	√
UART_IrDA	√	√	√	√
UART_LIN	√	√	√	√
UART_PDMA	√	√	√	√
UART_RS485	√	√	√	√
UART_SingleWire	√	√	√	√
UART_TxRxFunction	√	√	√	√
UART_Wakeup	√	√	√	√
USBD_Audio_Codec[3][4]	-	√	√	√

[3] Only M252 series support USBD samples.

[4] USBD_Audio_Codec requires I^2S.

Category	C	D	E	G
USBD_CCID[5]	-	√	√	√
USBD_HID_Keyboard	√	√	√	√
USBD_HID_Mouse	√	√	√	√
USBD_HID_MouseKeyboard	√	√	√	√
USBD_HID_RemoteWakeup	√	√	√	√
USBD_HID_Transfer	√	√	√	√
USBD_HID_Transfer_And_Keyboard	√	√	√	√
USBD_HID_Transfer_And_MSC[6]	-	√	√	√
USBD_HID_Transfer_CTRL	√	√	√	√
USBD_Mass_Storage_CDROM	√	√	√	√
USBD_Mass_Storage_Flash	-	√	√	√
USBD_Mass_Storage_SRAM	-	-	-	√
USBD_Micro_Printer	√	√	√	√
USBD_Printer_And_HID_Transfer	√	√	√	√
USBD_VCOM_And_HID_Keyboard	√	√	√	√
USBD_VCOM_And_HID_Transfer	√	√	√	√
USBD_VCOM_And_Mass_Storage	-	√	√	√
USBD_VCOM_DualPort	√	√	√	√
USBD_VCOM_SerialEmulator	√	√	√	√
USCI_I2C_EEPROM	√	√	√	√

[5] Limited by flash size.

[6] Mass storage function of USBD samples requires enough storage size for distinct file systems of operation systems.

Category	C	D	E	G
USCI_I2C_Lookback[7]	-	√	√	√
USCI_I2C_Loopback_10bit	-	√	√	√
USCI_I2C_Master	√	√	√	√
USCI_I2C_Master_10bit	√	√	√	√
USCI_I2C_Monitor	√	√	√	√
USCI_I2C_MultiBytes_Master	√	√	√	√
USCI_I2C_SingleByte_Master	√	√	√	√
USCI_I2C_Slave	√	√	√	√
USCI_I2C_Slave_10bit	√	√	√	√
USCI_I2C_Wakeup_Slave	√	√	√	√
USCI_SPI_Loopback	√	√	√	√
USCI_SPI_MasterMode	√	√	√	√
USCI_SPI_PDMA_LoopTest	-	√	√	√
USCI_SPI_SlaveMode	√	√	√	√
USCI_UART_AutoBaudRate	√	√	√	√
USCI_UART_Autoflow_Master	√	√	√	√
USCI_UART_Autoflow_Slave	√	√	√	√
USCI_UART_PDMA	√	√	√	√
USCI_UART_RS485_Master	√	√	√	√
USCI_UART_RS485_Slave	√	√	√	√
USCI_UART_TxRxFunction	√	√	√	√

[7] USCI_I2C_Lookback and USCI_I2C_Loopback_10bit require two USCI_I2C ports.

Category	C	D	E	G
USCI_UART_Wakeup[8]	√	√	√	√
WDT_TimeoutWakeupAndReset	√	√	√	√
WWDT_CompareINT	√	√	√	√

[8] USCI_UART_Wakeup does not support nCTS wakeup function in Category-C parts.

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.