

## M0519 CMSIS BSP Directory

Directory Introduction for 32-bit NuMicro™ Family

### Directory Information

<b>Document</b>	Driver reference manual and reversion 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

NuMicro M0519 Series CMSIS BSP Revision History.pdf	Show all the revision history about specific BSP.
NuMicro M0519 Driver Reference Guide.chm	Describe the definition, input and output of each API.

## 2. Library Information

<b>CMSIS</b>	CMSIS definitions by ARM® Corp.
<b>Component</b>	Library for peripheral components.
<b>Device</b>	CMSIS compliant device header file.
<b>StdDriver</b>	All peripheral driver header and source files.
<b>ThirdParty</b>	Library from the third party.

### 3. Sample Code Information

<b>\SampleCode\FreeRTOS</b>	Simple FreeRTOS™ demo 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>	M0519 Series Driver Samples

#### **4. \SampleCode\ISP**

<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>BPWM_Capture</b>	Capture the BPWM0 Channel 0 waveform by BPWM0 Channel 1.
<b>BPWM_DeadZone</b>	Demonstrate how to use BPWM Dead Zone function.
<b>BPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by BPWM Double Buffer function.
<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to do the EADC continuous scan conversion.
<b>EADC_PWM_Trigger</b>	Demonstrate how to trigger EADC by BPWM.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SimultaneousMode</b>	Show how to converts two different input signal at the same time by simultaneous mode of EADC.(Two ADC converters sample simultaneously.)
<b>EADC_SWTRG_Trigger</b>	Trigger EADC by writing ADSSTR register.
<b>EADC_Timer_Trigger</b>	Show how to trigger EADC by timer.
<b>ECAP</b>	Show how to use ECAP to measure clock frequency
<b>EPWM_DeadZone</b>	Demonstrate how to use EPWM Dead Zone function.
<b>EPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by EPWM Double Buffer function.
<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</b>	Implement a multi-boot system to boot from different applications in APROM. A LDROM code and 4 APROM code are implemented in this sample code.

<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>HDIV</b>	Show how to calculate with hardware divider.
<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 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>MDU_FOC</b>	To show how to use MDU + EPWM to implement FOC and output SVPWM waveform.
<b>OPA</b>	Demonstrate how OPA works with schmitt trigger buffer.
<b>QEI</b>	Show the usage of QEI compare function.

<b>SPI_Flash_With_FIFO</b>	Demonstrate how to access a Winbond 25Q16 SPI flash with FIFO buffers.
<b>SPI_Flash_Without_FIFO</b>	Demonstrate how to access a Winbond 25Q16 SPI flash without FIFO buffers.
<b>SPI_Loopback</b>	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO pin and SPI0_MOSI 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_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>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>BPWM_Capture</b>	Capture the BPWM0 Channel 0 waveform by BPWM0 Channel 1.
<b>BPWM_DeadZone</b>	Demonstrate how to use BPWM Dead Zone function.
<b>BPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by BPWM Double Buffer function.
<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to do the EADC continuous scan conversion.
<b>EADC_PWM_Trigger</b>	Demonstrate how to trigger EADC by BPWM.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SimultaneousMode</b>	Show how to converts two different input signal at the same time by simultaneous mode of EADC. (Two ADC converters sample simultaneously.)
<b>EADC_SWTRG_Trigger</b>	Trigger ADC by writing ADSSTR register.
<b>EADC_Timer_Trigger</b>	Show how to trigger EADC by timer.
<b>ECAP</b>	Show how to use ECAP to measure clock frequency
<b>EPWM_DeadZone</b>	Demonstrate how to use EPWM Dead Zone function.
<b>EPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by EPWM Double Buffer function.
<b>FMC_IAP</b>	Show how to reboot to LDROM functions from APROM. This sample code set VECMAP to LDROM and reset to re-boot to LDROM.
<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>HDIV</b>	Show how to calculate with hardware divider.
<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 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>QEI</b>	Show the usage of QEI compare function.
<b>SPI_Flash_With_FIFO</b>	Demonstrate how to access a Winbond 25Q16 SPI flash with FIFO buffers.
<b>SPI_Flash_Without_FIFO</b>	Demonstrate how to access a Winbond 25Q16 SPI flash without FIFO buffers.
<b>SPI_Loopback</b>	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO pin and SPI0_MOSI 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_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>SPI_SD_Card</b>	Demonstrate how to access a SD card formatted in FAT file system.
<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_Delay</b>	Show how to use timer0 to create various delay time.
<b>TIMER_PeriodicINT</b>	Implement timer counting in periodic mode.
<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.

### **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.*