

M0518 Board Supporting Package Directory Introduction

Rev.3.00.001

Directory Information

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

Document Information

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

Library Information

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

Sample Code Information

\SampleCode\Template	Software Development Template.
\SampleCode\Semihost	The sample code to show how to debug with semihost message print.
\SampleCode\RegBased	The sample code able to access control registers directly.
\SampleCode\StdDriver	Driver Samples

\SampleCode\RegBased

ADC_ContinuousScanMode	Demonstrate how to use continuous scan mode and finishes two cycles of conversion for the specified channels.
ADC_PwmTrigger	Demonstrate how to trigger ADC by PWM.
ADC_ResultMonitor	Demonstrate how to use the digital compare function to monitor the conversion result of channel 2.
ADC_SingleCycleScanMode	Demonstrate how to use single cycle scan mode and finishes one cycle of conversion for the specified channels.
ADC_SingleMode	Demonstrate how to use single mode and finishes the conversion of the specified channel.
BPWM_Capture	Demonstrate how to use BPWM0 Channel 0 to capture the BPWM1 Channel 0 waveform.
BPWM_DoubleBuffer	Demonstrate how to use BPWM Double Buffer function to change duty cycle and period of output waveform.
FMC_IAP	Demonstrate 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	Demonstrate how to implement multi-boot system to boot from different applications in APROM. A LDROM code and 4 APROM code are implemented in this sample code.
FMC_RW	Demonstrate how to read/program embedded flash by ISP function.
GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use GPIO interrupt function.
GPIO_OutputInput	Demonstrate how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Demonstrate how to wake up system form Power-down mode by GPIO interrupt.

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. 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_Master	Demonstrate how a Master access Slave. This sample code needs to work with I2C_SLAVE.
I2C_Slave	Demonstrate how to set I2C in Slave mode to receive the data of a Master. This sample code needs to work with I2C_MASTER.
I2C_Wakeup_Master	Demonstrate how to wake up MCU from power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Demonstrate how to set I2C to wake up MCU from Power-down mode. This sample code needs to work with I2C_Wakeup_Master.
PWM_Capture	Demonstrate how to use PWM0 Channel 0 to capture the PWM1 Channel 0 waveform.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Demonstrate how to use PWM Double Buffer function to change duty cycle and period of output waveform.
SPI_Loopback	Demonstrate SPI master loop back transfer. Needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterFifoMode	Demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with SPI_SlaveFifoMode.
SPI_SlaveFifoMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterFifoMode.

SYS_PLLClockOutput	Demonstrate how to change system clock to different PLL frequency and output system clock from CLK0 pin.
TIMER_Capture	Demonstrate how to use the timer0 capture event to capture timer0 counter value.
TIMER_Counter	Demonstrate how to use timer0 counter input function to count the input event.
TIMER_PeriodicINT	Demonstrate how to perform timer counting in Periodic mode.
TIMER_PowerDown	Demonstrate how to use timer0 periodic time-out interrupt event to wake up system.
UART_AutoFlow_Master	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_AutoFlow_Slave.
UART_AutoFlow_Slave	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_AutoFlow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Demonstrate how to transmit LIN header and response.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485 mode. This 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.

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 WDT time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to generate 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.

\SampleCode\StdDriver

ADC_ContinuousScanMode	Demonstrate how to use continuous scan mode and finishes two cycles of conversion for the specified channels.
ADC_PwmTrigger	Demonstrate how to trigger ADC by PWM.
ADC_ResultMonitor	Demonstrate how to use the digital compare function to monitor the conversion result of channel 2.
ADC_SingleCycleScanMode	Demonstrate how to use single cycle scan mode and finishes one cycle of conversion for the specified channels.
ADC_SingleMode	Demonstrate how to use single mode and finishes the conversion of the specified channel.
BPWM_Capture	Demonstrate how to use BPWM0 Channel 0 to capture the BPWM1 Channel 0 waveform.
BPWM_DoubleBuffer	Demonstrate how to use BPWM Double Buffer function to change duty cycle and period of output waveform.
DAC_PWMTrigger	Demonstrate how to trigger DAC by PWM.
DAC_SoftwareTrigger	Demonstrate how to trigger DAC conversion by software method.
DAC_TimerTrigger	Demonstrate how to trigger DAC by timer.

EADC_ADINT_Trigger	Demonstrate how to use ADINT interrupt to do the ADC continuous scan conversion.
EADC_PWM_Trigger	Demonstrate how to trigger ADC by PWM.
EADC_ResultMonitor	Demonstrate how to use the digital compare function to monitor the conversion result of channel 2.
EADC_SWTRG_Trigger	Demonstrate how to trigger ADC by writing EADC_SWTRG register.
EADC_Timer_Trigger	Demonstrate how to trigger ADC by timer.
FMC_IAP	Demonstrate how to reboot to LDROM functions from APROM. This sample code set VECMAP to LDROM and reset to re-boot to LDROM.
FMC_RW	Demonstrate how to read/program embedded flash by ISP function.
GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use GPIO interrupt function.
GPIO_OutputInput	Demonstrate how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Demonstrate how to wake up system form Power-down mode by GPIO interrupt.
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. 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_Master	Demonstrate how a Master access Slave. This sample code needs to work with I2C_SLAVE.

I2C_Slave	Demonstrate how to set I2C in Slave mode to receive the data of a Master. This sample code needs to work with I2C_MASTER.
I2C_Wakeup_Master	Demonstrate how to wake up MCU from power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Demonstrate how to set I2C to wake up MCU from power-down mode. This sample code needs to work with I2C_Wakeup_Master.
PWM_Capture	Demonstrate how to use PWM0 Channel 0 to capture the PWM1 Channel 0 waveform.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Demonstrate how to use PWM Double Buffer function to change duty cycle and period of output waveform.
SPI_Loopback	Demonstrate SPI master loop back transfer. Needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterFifoMode	Demonstrate how to communicate with an off-chip SPI slave device. Needs to work with SPI_SlaveFifoMode sample code.
SPI_SlaveFifoMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterFifoMode.
SYS_PLLClockOutput	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
TIMER_Capture	Demonstrate how to use the timer0 capture event to capture timer0 counter value.
TIMER_Counter	Demonstrate how to use timer0 counter input function to count the input event.
TIMER_Delay	Demonstrate how to use timer0 to create a precise delay loop.

TIMER_PeriodicINT	Demonstrate how to perform timer counting in periodic mode.
TIMER_PowerDown	Demonstrate how to use timer0 periodic time-out interrupt event to wake up system.
UART_Autoflow_Master	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Demonstrate how to transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Demonstrate how to transmit LIN header and response.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485 mode. This 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.
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 WDT time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to generate WDT time-out reset system event while WDT time-out reset delay period expired.

WWDT_CompareINT

Select one WWDT window compare value to generate the 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.*