

## Mini58 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

### Directory Information

<b>Document</b>	Driver reference guide 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)

## TABLE OF CONTENTS

<b>1</b>	<b>DOCUMENT.....</b>	<b>4</b>
<b>2</b>	<b>LIBRARY .....</b>	<b>5</b>
<b>3</b>	<b>SAMPLECODE .....</b>	<b>6</b>
<b>4</b>	<b>SAMPLECODE\ISP .....</b>	<b>7</b>
<b>5</b>	<b>SAMPLECODE\REGBASED .....</b>	<b>8</b>
	System Manager (SYS) .....	8
	Flash Memory Controller (FMC) .....	8
	General Purpose I/O (GPIO) .....	8
	Timer Controller (TIMER) .....	8
	Watchdog Timer (WDT) .....	9
	Window Watchdog Timer (WWDT) .....	9
	PWM Generator (PWM) .....	9
	UART Interface Controller (UART) .....	9
	Serial Peripheral Interface (SPI) .....	9
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	10
	Analog-to-Digital Converter (ADC) .....	10
	Analog Comparator Controller (ACMP) .....	10
<b>6</b>	<b>SAMPLECODE\STDDRIVER.....</b>	<b>12</b>
	System Manager (SYS) .....	12
	Flash Memory Controller (FMC) .....	12
	General Purpose I/O (GPIO) .....	12
	Timer Controller (TIMER) .....	12
	Watchdog Timer (WDT) .....	13
	Window Watchdog Timer (WWDT) .....	13
	PWM Generator (PWM) .....	13
	UART Interface Controller (UART) .....	13
	Serial Peripheral Interface (SPI) .....	14
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	14

Analog-to-Digital Converter (ADC) .....	14
Analog Comparator Controller (ACMP) .....	15

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 4.5.0.
<b>NuMicro Mini58 Driver Reference Guide.html</b>	This document describes the usage of drivers in Mini58 BSP.
<b>NuMicro Mini58 Series CMSIS BSP Revision History.pdf</b>	This document shows the revision history of Mini58 BSP.

## 2 Library

<b>CMSIS</b>	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by Arm® Corp.
<b>Device</b>	CMSIS compliant device header file.
<b>StdDriver</b>	All peripheral driver header and source files.

### 3 SampleCode

Hard_Fault_Sample	Show hard fault information when hard fault happened.
ISP	ISP firmware samples.
RegBased	Sample codes implemented without access standard library but access registers directly.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Demonstrate the usage of Mini58 MCU peripheral driver APIs.
Template	A project template for Mini58 MCU.

## 4 SampleCode\ISP

ISP_I2C	In-System-Programming sample code through I <sup>2</sup> C interface.
ISP_RS485	In-System-Programming sample code through RS485 interface.
ISP_SPI	In-System-Programming sample code through SPI interface.
ISP_UART	In-System-Programming sample code through UART interface.

## 5 SampleCode\RegBased

### System Manager (SYS)

<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLKO pin.
---------------------------	---

### Flash Memory Controller (FMC)

<b>FMC_RW</b>	Show FMC read flash IDs, erase, read, and write functions.
---------------	--

### General Purpose I/O (GPIO)

<b>GPIO</b>	Use GPIO driver to control the GPIO pin direction, control their high/low state, and how to use GPIO interrupts.
-------------	--

### Timer Controller (TIMER)

<b>Timer_EventCounter</b>	Use pin P3.4 to demonstrates timer event counter function.
<b>Timer_FreeCountingMode</b>	Use the timer pin P3.2 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
<b>Timer_Periodic</b>	Use the timer periodic mode to generate timer interrupt every 1 second.
<b>Timer_ToggleOut</b>	Demonstrate the timer 0 toggle out function on pin P3.4.
<b>Timer_TriggerCountingMode</b>	Use the timer pin P3.2 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
<b>Timer_Wakeup</b>	Use Timer to wake up system from Power-down mode periodically.



## Watchdog Timer (WDT)

WDT_Polling	Use polling mode to check WDT time-out state and reset WDT after time out occurs.
WDT_Wakeup	Use WDT to wake up system from Power-down mode periodically.

## Window Watchdog Timer (WWDT)

WWDT_Reload	Demonstrate the WWDT counter reload function.
-------------	---

## PWM Generator (PWM)

PWM_DeadZone	Demonstrate the dead-zone feature with PWM.
PWM_DoubleBuffer	Demonstrate the PWM double buffer feature.
PMW_MaskAlign	Show how to generate 0%, 50% and 100% PWM duty cycle.
PWM_PreciseCenterAlignedMode	Demonstrate PWM precise center aligned feature.

## UART Interface Controller (UART)

UART_AutoFlow	Show how to transmit and receive data using auto flow control.
UART_IrDA	Show how to transmit and receive UART data in UART IrDA mode.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.

## Serial Peripheral Interface (SPI)

SPI_LoopBack	Demonstrate SPI function by connect MOSI (P0.5) with MISO (P0.6)
--------------	--

<b>SPI_MasterFIFOmode</b>	Demonstrate how to communicate with an off-chip SPI slave device using FIFO mode. This sample code could work with <a href="#">SPI_SlaveFIFOmode</a> sample code.
<b>SPI_SlaveFIFOmode</b>	Demonstrate how to communicate with an off-chip SPI master device using FIFO mode. This sample code needs to work with <a href="#">SPI_MasterFIFOmode</a> sample code.

## I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

<b>I2C_FIFO_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using FIFO mode.
<b>I2C_Interrupt_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using interrupt mode.
<b>I2C_Polling_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using polling mode.
<b>I2C_Software_GPIO</b>	Demonstrate how to use GPIO pins to simulate I <sup>2</sup> C interface.

## Analog-to-Digital Converter (ADC)

<b>ADC_Compare</b>	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
<b>ADC_Convert</b>	Demonstrate ADC function by repeatedly convert the input of ADC channel 0 (P5.3) and shows the result on UART console.
<b>ADC_PWMTrigger</b>	Demonstrate PWM0 channel 0 trigger ADC function.
<b>ADC_SequentialMode</b>	Demonstrate ADC PWM Sequential Mode conversion and shows the result on UART console.

## Analog Comparator Controller (ACMP)

<b>ACMP</b>	Demonstrate Analog comparator (ACMP) comparison by comparing CPP0 (P1.5) with Band-gap voltage and shows the result on UART console.
-------------	--

### ACMP\_TriggerTimerCapture

Show how to use Analog comparator (ACMP) state change to trigger timer capture function. P1.5 is used as comparator positive input and Band-gap voltage as negative input.

## 6 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS</b>	Show how to wake up system form Power-down mode by brown-out detector interrupt.
<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLK0 pin.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.

### Flash Memory Controller (FMC)

<b>FMC_RW</b>	Show FMC read flash IDs, erase, read, and write functions.
---------------	--

### General Purpose I/O (GPIO)

<b>GPIO</b>	Use GPIO driver to control the GPIO pin direction, control their high/low state, and how to use GPIO interrupts.
-------------	--

### Timer Controller (TIMER)

<b>Timer_Delay</b>	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay
<b>Timer_EventCounter</b>	Use pin P3.4 to demonstrates timer event counter function.
<b>Timer_FreeCountingMode</b>	Use the timer pin P3.2 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
<b>Timer_Periodic</b>	Use the timer periodic mode to generate timer interrupt every 1 second.
<b>Timer_ToggleOut</b>	Demonstrate the timer 0 toggle out function on pin P3.4.

<b>Timer_TriggerCountingMode</b>	Use the timer pin P3.2 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
<b>Timer_Wakeup</b>	Use Timer to wake up system from Power-down mode periodically.

### Watchdog Timer (WDT)

<b>WDT_Polling</b>	Use polling mode to check WDT time-out state and reset WDT after time out occurs.
<b>WDT_Wakeup</b>	Use WDT to wake up system from Power-down mode periodically.

### Window Watchdog Timer (WWDT)

<b>WWDT_Reload</b>	Demonstrate the WWDT counter reload function.
--------------------	---

### PWM Generator (PWM)

<b>PWM_DeadZone</b>	Demonstrate the dead-zone feature with PWM.
<b>PWM_DoubleBuffer</b>	Demonstrate the PWM double buffer feature.
<b>PMW_MaskAlign</b>	Show how to generate 0%, 50% and 100% PWM duty cycle.
<b>PWM_PreciseCenterAligned Mode</b>	Demonstrate PWM precise center aligned feature.

### UART Interface Controller (UART)

<b>UART_AutoFlow</b>	Show how to transmit and receive data using auto flow control.
<b>UART_IrDA</b>	Show how to transmit and receive UART data in UART IrDA mode.
<b>UART_RS485</b>	Transmit and receive data in UART RS485 mode.

<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
---------------------------	---

## Serial Peripheral Interface (SPI)

<b>SPI_LoopBack</b>	Demonstrate SPI function by connect MOSI (P0.5) with MISO (P0.6)
<b>SPI_MasterFIFOmode</b>	Demonstrate how to communicate with an off-chip SPI slave device using FIFO mode. This sample code could work with <a href="#">SPI_SlaveFIFOmode</a> sample code.
<b>SPI_SlaveFIFOmode</b>	Demonstrate how to communicate with an off-chip SPI master device using FIFO mode. This sample code needs to work with <a href="#">SPI_MasterFIFOmode</a> sample code.

## I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

<b>I2C_FIFO_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using FIFO mode.
<b>I2C_Interrupt_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using interrupt mode.
<b>I2C_Polling_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface using polling mode.
<b>I2C_Software_GPIO</b>	Demonstrate how to use GPIO pins to simulate I <sup>2</sup> C interface.

## Analog-to-Digital Converter (ADC)

<b>ADC_Compare</b>	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
<b>ADC_Convert</b>	Demonstrate ADC function by repeatedly convert the input of ADC channel 0 (P5.3) and shows the result on UART console.
<b>ADC_PWMTrigger</b>	Demonstrate PWM0 channel 0 trigger ADC function.
<b>ADC_SequentialMode</b>	Demonstrate ADC PWM Sequential Mode conversion

and shows the result on UART console.

## Analog Comparator Controller (ACMP)

### ACMP

Demonstrate Analog comparator (ACMP) comparison by comparing CPP0 (P1.5) with Band-gap voltage and shows the result on UART console.

### ACMP\_TriggerTimerCapture

Show how to use Analog comparator (ACMP) state change to trigger timer capture function. P1.5 is used as comparator positive input and Band-gap voltage as negative 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.*