Mini51DE Series CMSIS 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 4.5.0
NuMicro Mini51DE Series CMSIS BSP Revision History.pdf	This document shows the revision history of Mini51DE BSP.
NuMicro Mini51DE Driver Reference Guide.chm	This document describes the usage of drivers in Mini51DE BSP.

2 Library Information

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by ARM® Corp.
Device	CMSIS compliant device header file.
Nu-LB-Mini51	Library for Mini51DE Learning Board
StdDriver	All peripheral driver header and source files.

3 Sample Code Information

Hard_Fault_Sample	Show hard fault information when hard fault happened.
Nu-LB-Mini51	Sample codes for Mini51DE Learning Board
NuTiny-Mini51	Same codes for Mini51DE Tiny Board
RegBased	Sample codes implemented without access standard library but access registers directly.
Semihost	Show how to print and get character with IDE console window.
StdDriver	Demonstrate the usage of Mini51DE MCU peripheral driver APIs.
Template	A project template for Mini51DE MCU.

4 \SampleCode\Nu-LB-Mini51

ADC_PWM	This sample adjusts the PWM output duty according to ADC conversion result where the input voltage is control by VR. The PWM output connects to a buzzer so user can control the buzzer tone with VR.
DeepSleep	This sample code demonstrates how to let system enter and exit deep sleep mode with external interrupt.
I2C_FIFO_EEPROM	This sample demonstrates how to read/write EEPROM via I²C interface using FIFO mode.
I2C_Polling_EEPROM	This sample demonstrates how to read/write EEPROM via I²C interface using polling mode.
I2C_Software_GPIO	This sample code demonstrates how to use GPIO pins to simulate an I²C interface.
I2C_Software_GPIO_Timer	This sample demonstrates how to read/write EEPROM via GPIO pins which simulate I²C interface.
Idle	This sample code shows how to wake system up from idle mode with WDT interrupt.
Interrupt	This sample code demonstrates how to let system enter and exit deep sleep mode with GPIO interrupts.
LCD	This sample code demonstrates how to control a LCD module via SPI interface.
StartKit	This is a starter kit sample enables all peripherals on learning board. Peripherals enabled are UART, SPI, I²C, Timer, ADC, and PWM.
Timer_WDT	This sample demonstrates how to configure timer in periodic mode and watchdog timer. The interrupt status of timer and WDT is shown on LCD control via SPI interface.

5 \SampleCode\NuTiny-Mini51

LED	This sample toggles P3.6 to turn on board LED on and off.

6 \SampleCode\RegBased

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.
ADC_Compare	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
ADC_Convert	Demonstrate ADC function by repeatedly convert the input of ADC channel 0 (P5.3) and shows the result on UART console.
FMC_RW	Show FMC read flash IDs, erase, read, and write functions.
GPIO_Debounce	Demonstrate GPIO de-bounce function.
GPIO_Interrupt	Shows the usage of GPIO interrupt function.
GPIO_Toggle	Show how to toggle GPIO pin.
GPIO_Wakeup	Show how to wake up system from Power-down mode by GPIO interrupt.
I2C_Interrupt_EEPROM	Read/write EEPROM via I²C interface using interrupt mode.
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.
PWM_DeadZone	Demonstrate the dead-zone feature with PWM.
PWM_DoubleBuffer	Demonstrate the PWM double buffer feature.

SPI_LoopBack	Demonstrate SPI function by connect MOSI (P0.5) with MISO (P0.6).
SPI_MasterFIFOMode	Demonstrate how to communicate with an off-chip SPI slave device using FIFO mode.
SPI_MasterMode	Demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with SPI_SlaveMode.
SPI_SlaveFIFOMode	Demonstrate how to communicate with an off-chip SPI master device using FIFO mode.
SPI_SlaveMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterMode.
Timer_EventCounter	Use pin P3.4 to demonstrates timer event counter function.
Timer_FreeCountingMode	Use the timer pin P3.2 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
Timer_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
Timer_ToggleOut	Demonstrate the timer 0 toggle out function on pin P3.4.
Timer_TriggerCountingMode	Use the timer pin P3.2 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
Timer_Wakeup	Use Timer to wake up system from Power-down mode periodically.
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.

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.

7 \SampleCode\StdDriver

ACMP	Demonstrate Analog comparator (ACMP) comparison by comparing CPP0 (P1.5) with Band-gap voltage and shows the result on UART console.
ADC_Compare	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
ADC_Convert	Demonstrate ADC function by repeatedly convert the input of ADC channel 0 (P5.3) and shows the result on UART console.
FMC_IAP	This sample code includes LDROM image (fmc_ld_iap) and APROM image (fmc_ap_main). It shows how to branch between APROM and LDROM. To run this sample code, the boot mode must be “Boot from APROM with IAP”.
FMC_RW	Show FMC read flash IDs, erase, read, and write functions.
GPIO_Debounce	Demonstrate GPIO de-bounce function.
GPIO_Interrupt	Shows the usage of GPIO interrupt function.
GPIO_Toggle	Show how to toggle GPIO pin.
GPIO_Wakeup	Show how to wake up system from Power-down mode by GPIO interrupt.
I2C_Interrupt_EEPROM	Read/write EEPROM via I²C interface using interrupt mode.
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.
PWM_DeadZone	Demonstrate the dead-zone feature with PWM.
PWM_DoubleBuffer	Demonstrate the PWM double buffer feature.

SPI_LoopBack	Demonstrate SPI function by connect MOSI (P0.5) with MISO (P0.6).
SPI_MasterFIFOMode	Demonstrate how to communicate with an off-chip SPI slave device using FIFO mode.
SPI_MasterMode	Demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with SPI_SlaveMode.
SPI_SlaveFIFOMode	Demonstrate how to communicate with an off-chip SPI master device using FIFO mode.
SPI_SlaveMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterMode.
SYS	Demonstrate how to get PDID, get and clear reset source, configure BOD, and output system clock to CKO pin with the system clock / 4 frequency.
Timer_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay
Timer_EventCounter	Use pin P3.4 to demonstrates timer event counter function.
Timer_FreeCountingMode	Use the timer pin P3.2 to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
Timer_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
Timer_ToggleOut	Demonstrate the timer 0 toggle out function on pin P3.4.
Timer_TriggerCountingMode	Use the timer pin P3.2 to demonstrate timer trigger counting mode function. And displays the measured input frequency to UART console.
Timer_Wakeup	Use Timer to wake up system from Power-down mode periodically.
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.
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.

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.