ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® Family
Mini51DE Series CMSIS BSP
Revision History

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

Revision 3.02.003 (Released 2021-12-16)

1. Minor bug fix.

Revision 3.02.002 (Released 2020-10-6)

2. Added Apache-2.0 license declaration into driver source code.
3. Minor bug fix.

Revision 3.02.001 (Released 2019-11-6)

1. Added ISP related samples.

Revision 3.02.000 (Released 2018-11-15)

1. Added Eclipse project support.
2. Minor bug fix.

Revision 3.01.002 (Released 2017-3-10)

1. Updated CLK_Idle() to clear CLK_PWRCON_PWR_DOWN_EN_Msk flag before entering idle mode.

Revision 3.01.001 (Released 2016-6-15)

1. Updated CMSIS to v4.5.0.
2. Added sample codes including SPI_MasterFIFOMode and SPI_SlaveFIFOMode.
3. Minor bug fixes.

Revision 3.01.000 (Released 2015-10-8)

1. Removed PLL related API declarations in clk.h.
2. Removed DID related functions and macros in fmc.c and fmc.h.
3. Removed FMC_SetBootSource(), FMC_DisableAPUpdate(), FMC_DisableConfigUpdate(), FMC_DisableLDUpdate(), FMC_EnableAPUpdate(), FMC_EnableConfigUpdate(), FMC_EnableLDUpdate() functions in fmc.c and fmc.h because there exist functionally identical macros.
4. Removed SPI_GET_RX_FIFO_FULL_FLAG() macro in spi.h.
5. Removed CHIP_RST and CPU_RST definitions in sys.h.
6. Updated startup_Mini51Series.s and retarget.c to implement a Hard Fault handler that dumps register value before going to an infinite loop.
7. Updated return type of CLK_WaitClockReady() function from void to uint32_t in clk.c.
8. Updated SPI_Close() to remove the code that disable SPI peripheral clock in spi.c.
9. Updated all IAR samples to set Reset_Handler() as application entry point.
10. Updated fmc_ld_iap sample to reduce code size by replacing retarget.c with ldrom_retarget.c.
11. Updated I2C_Interrupt_EEPROM sample to poll STOP bit after STOP command.
12. Updated DeepSleep sample so that CPU is in unlock state before entering Power-down mode.
13. Fixed CLK_SysTickDelay() bug that delay time is sometimes incorrect in clk.c.
14. Fixed FMC_Erase() bug that modifies irrelevant register bits in fmc.c.
15. Fixed GPIO_DISABLE_DOUT_MASK() and GPIO_ENABLE_DOUT_MASK() error that sets and clear wrong bits in gpio.h.
16. Fixed I2C_GetBusClockFreq() bug that reports wrong clock frequency in i2c.c.
17. Fixed PWM_FB1_ACMP0 definition error in pwm.h.
18. Fixed PWM_ENABLE_OUTPUT_INVERTER() bug that set wrong bit in pwm.h

19. Fixed SPI_EnableAutoSS() register setting error in spi.c.
20. Fixed SPI_SetBusClock() clock calculation error in spi.c.
21. Fixed SYS_DISABLE_BOD() and SYS_CLEAR_RST_SOURCE() implementation error in sys.h
22. Fixed UART_EnableFlowCtrl() register setting error in uart.c.
23. Fixed UART_Open() and UART_SetLine_Config() baud rate calculation error in uart.c while UART peripheral clock divider is not 1.
24. Fixed UART_SelectIrDAMode() error that only supports 12MHZ crystal as clock source in uart.c.
25. Fixed UART_CLEAR_RTS() implementation error in uart.h.
26. Fixed sample code bug that XTL is enabled before configuring multi-function pins.
27. Fixed CMSIS path setting error in some sample codes.
28. Fixed UART_RS485 samples that set RTS pin and RS485 mode incorrectly.
29. Added I2C0, SPI0, and UART0 definition which is the same with I2C, SPI, and UART in Mini51Series.h.
30. Added ACMP driver acmp.c and acmp.h.
31. Added CLK_EnableSysTick() and CLK_DisableSysTick() in clk.c.
32. Added CLK_PWRCON_XTL12M, CLK_PWRCON_HXT, CLK_PWRCON_LXT, CLK_CLKSEL0_HCLK_S_LIRC, CLK_CLKSEL0_HCLK_S_HIRC, CLK_CLKSEL0_STCLK_S_HIRC_DIV2, CLK_CLKSEL1_WDT_S_LIRC, CLK_CLKSEL1_ADC_S_HIRC, CLK_CLKSEL1_TMR0_S_LIRC, CLK_CLKSEL1_TMR0_S_HIRC, CLK_CLKSEL1_TMR1_S_LIRC, CLK_CLKSEL1_TMR1_S_HIRC, CLK_CLKSEL1_UART_S_HIRC, CLK_CLKSEL2_FRQDIV_HXT, CLK_CLKSEL2_FRQDIV_LXT, CLK_CLKSEL2_FRQDIV_HIRC definitions in clk.h.
33. Added FMC_CLR_FAIL_FLAG() macro in fmc.h.
34. Added FMC_GetVectorPageAddr() function in fmc.c.
35. Added SPI_GET_RX_FIFO_FULL_FLAG() and SPI_GET_RX_FIFO_FULL_FLAG() macros in spi.h.
36. Added SYS_MFP_P32_CPP1 definition in sys.h.
37. Added Hard_Fault_Sample.
38. Added RegBased samples including ACMP_TriggerTimerCompare, ADC_Compare, , GPIO_Debounce, GPIO_Interrupt, GPIO_Toggle, GPIO_Wakeup, I2C_Master, I2C_Slave, PWM_DeadZone, SPI_MasterMode, SPI_SlaveMode, Timer_EventCounter, Timer_FreeCountingMode, Timer_ToggleOut, Timer_Wakeup, and WDT_Wakeup.
39. Added StdDriver samples including ACMP, ADC_Convert, GPIO_Debounce, GPIO_Interrupt, GPIO_Toggle, GPIO_Wakeup, I2C_Master, I2C_Slave, PWM_DoubleBuffer, SPI_MasterMode, SPI_SlaveMode, Timer_Periodic, Timer_TriggerCountingMode, Timer_Wakeup, and WDT_Polling.

Revision 3.00.002 (Released 2014-02-27)

1. Renamed PWM_*FaultBreak*() functions to PWM_*FaultBrake*().
2. Renamed I2C_GetClockBusFreq() to I2C_GetBusClockFreq().
3. Renamed I2C_SetClockBusFreq() to I2C_SetBusClockFreq().
4. Renamed I2C_SetSlaveMask() to I2C_SetSlaveAddrMask().
5. Added I2C FIFO mode sample
6. Minor bug fixes.

Revision 3.00.001 (Released 2013-10-31)

1. Minor bug fixes

Revision 3.00.000 (Released 2013-10-08)

1. Initial release.

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.