

ARM® Cortex®-M
32-bit Microcontroller

Mini58 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 3.01.003 (Released 2022-03-03)

1. Enable I2C pin schmitt trigger.
2. Fixed sample FMC_IAP compiling error of IAR7.
3. Add time-out check to drivers and samples to prevent from infinite loop.
4. Minor bug fix.

Revision 3.01.002 (Released 2020-10-6)

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

Revision 3.01.001 (Released 2019-11-6)

1. Added ISP related samples.

Revision 3.01.000 (Released 2018-11-14)

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

Revision 3.00.003 (Released 2017-03-10)

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

Revision 3.00.002 (Released 2016-06-15)

1. Updated CMSIS to v4.5.0.
2. Added sample codes including ADC_PWMTrigger, SPI_MasterFIFOmode and SPI_SlaveFIFOmode.
3. Minor bug fixes.

Revision 3.00.001 (Released 2015-06-05)

1. Regenerated register and bit field definitions in Mini58Series.h based on Mini58 Technical Reference Manual Rev 1.0.
2. Removed FMC_ReadDID(), FMC_DisableAPUUpdate(), FMC_DisableSPUUpdate(), FMC_SetBootSource(), FMC_DisableConfigUpdate(), FMC_DisableLDUpdate(), FMC_EnableAPUUpdate(), FMC_EnableSPUUpdate(), FMC_EnableConfigUpdate(), FMC_EnableLDUpdate() functions in fmc.c.
3. Renamed TIMER_TOGGLE_TMXEXT to TIMER_TOGGLE_TMX_EXT in timer.h.
4. Updated all IAR samples to set Reset_Handler() as entry point.
5. Updated FMC_WriteConfig() to check input parameter in fmc.c.
6. Updated TIMER_Open() and TIMER_Delay() to support the time-out calculation while timer clock is faster than 0x1FFFFFF Hz in timer.c.
7. Updated UART_Open() and UART_SetLine_Config() to support using PLL as UART clock source in uart.c.
8. Updated UART_SelectIrDAMode() to support using PLL and HIRC as UART clock source in uart.c.
9. Updated WDT_Open() to support reset delay parameter in wdt.c.
10. Updated Timer_Wakeup sample code to unlock protect register before entering Power-down mode.
11. Added ACMP_SET_FILTER() macro in acmp.h.
12. Added ADC_SeqModeEnable() and ADC_SeqModeTriggerSrc() functions in adc.c.
13. Added CLK_EnableSysTick() and CLK_DisableSysTick() functions in clk.c.
14. Added FMC_GetVectorPageAddr() function in fmc.c.

15. Added PWM_SET_CM RD(), PWM_ENABLE_ASYMMETRIC_MODE(), and PWM_ENABLE_PCAEN() macros in pwm.h.
16. Added PWM_EnableCenterInt(), PWM_DisableCenterInt(), PWM_ClearCenterIntFlag(), PWM_GetCenterIntFlag(), PWM_EnableRiseInt(), PWM_DisableRiseInt(), PWM_ClearRiseIntFlag(), and PWM_GetRiseIntFlag() function in pwm.c.
17. Added SPI_GET_TX_FIFO_FULL_FLAG() macro in spi.h.
18. Added PWM_MaskAlign sample.
19. Fixed ACMP_SET_NEG_SRC() and ACMP_ENABLE_RISING_EDGE_TRIGGER() implementation error in acmp.h.
20. Fixed I2C_GetBusClockFreq() implementation error in i2c.c.
21. Fixed I2C_CLEAR_WAKEUP_FLAG(), I2C_DISABLE_CLOCK_STRETCH() and I2C_ENABLE_CLOCK_STRETCH() implementation error in i2c.h.
22. Fixed TIMER_CAPTURE_RISING_* definition error in timer.h.
23. Fixed TIMER_EnableCaptureDebounce() and TIMER_DisableCaptureDebounce() implementation error in timer.c.
24. Fixed PWM_ENABLE_OUTPUT_INVERTER() implementation error in pwm.h.
25. Fixed PWM_GetADCTriggerFlag() and PWM_ConfigOutputChannel() implementation error in pwm.c.
26. Fixed SPI_SET_xSB_FIRST() implementation error in spi.h.
27. Fixed UART_CLEAR_RTS(), UART_EnableFlowCtrl(), UART_ClearIntFlag() implementation error in uart.c and uart.h.

Revision 3.00.000 (Released 2015-2-27)

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.

*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.*