

**ARM<sup>®</sup> Cortex<sup>®</sup>-M0**  
**32-bit Microcontroller**

**NuMicro<sup>®</sup> Family**  
**Mini55 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](http://www.nuvoton.com)

### Revision 3.01.001 (Released 2017-04-21)

1. Updated CMSIS to v4.5.0.
2. Updated CLK\_Idle() to clear CLK\_PWRCTL\_PDEN\_Msk flag before entering idle mode.
3. Updated I2C\_STOP() to clear I2C\_CTL\_SI\_Msk bit and check I2C\_CTL\_STO\_Msk until it is cleared before return.
4. Updated retarget.c to disable uart console when semihost is enabled.
5. Fixed UART1 clock setting error in UART\_Open().
6. Fixed ADC\_Open() channel setting error.
7. Added P2.4 and P2.5 pin definition for UART1\_RX1 and UART1\_TX1.

### Revision 3.01.000 (Released 2015-07-10)

1. Updated TIMER\_T structure, and moved CCAPCTL and CCAP[3] to a new structure named TIMER\_AC\_T in Mini55Series.h.
2. Updated the bit field name of PWM\_CTL register from CMPINV[1~5] to PINV[1~5] in Mini55Series.h.
3. Updated mask and position of GPIO bit field to be consistent with Mini55 Technical Reference Manual in Mini55Series.h.
4. Updated UART\_INTSTS\_MODENIF\_Pos and UART\_INTSTS\_MODENIF\_Msk to UART\_INTSTS\_MODEMIF\_Pos and UART\_INTSTS\_MODEMIF\_Msk in Mini55Series.h.
5. Updated CLK\_EnableXtalRC() to support run time change of XTLEN(CLK\_PWRCTL[1:0]) in clk.c.
6. Updated PWM\_GetADCTriggerFlag() to report a flag of each trigger source instead of a combinational result.
7. Updated SPI\_Close() so it does not disable SPI peripheral clock in spi.c.
8. Updated SYS\_EnableBOD() to support all the eight BOD levels in sys.c.
9. Updated TIMER\_Delay() and replaced the variable type used for calculation from "float" to "long long" to get more accurate value in timer.c.
10. Removed DID related functions and macros in fmc.[ch].
11. Removed CHIP\_RST and CPU\_RST definition, but called SYS\_ResetCPU() and SYS\_ResetChip() directly in sys.h.
12. Removed SYS\_BODCTL\_BOD\_DISABLE definition in sys.h.
13. Fixed ACMP\_SET\_NEG\_SRC() syntax error in acmp.h.
14. Fixed ACMP\_CRV\_SEL() implementation error, and updated incorrect mask name to fix the compilation error in acmp.h.
15. Fixed ACMP\_ENABLE\_RISING\_EDGE\_TRIGGER() implementation error that disables rising edge trigger in acmp.h.
16. Fixed CLK\_SysTickDelay() bug that delay time is sometimes incorrect in clk.c.
17. Fixed CLK\_CLKSEL1\_UART1SEL\_HIRC, UART1\_MODULE definition error in clk.h.
18. Fixed GPIO\_DISABLE\_DOUT\_MASK() and GPIO\_ENABLE\_DOUT\_MASK() error in gpio.h.
19. Fixed I2C\_GetBusClockFreq() bug that always gets divider from I2C0 in i2c.c.
20. Fixed I2C\_CLEAR\_WAKEUP\_FLAG() bug that may clear more than one bit in i2c.h.
21. Fixed I2C\_WAIT\_READY() bug that polls a wrong bit in i2c.h.
22. Fixed PWM\_FB1\_ACMP0 definition error in pwm.h.
23. Fixed PWM\_ENABLE\_OUTPUT\_INVERTER() bug that sets incorrect bit field in pwm.h.
24. Fixed PWM\_EnablePeriodInt(), PWM\_DisableCenterInt(), PWM\_ClearCenterIntFlag(), and PWM\_GetCenterIntFlag() implementation errors in pwm.c.

25. Fixed SPI\_EnableAutoSS() bug that old settings are not cleared before updating new value in spi.c.
26. Fixed SPI\_SetBusClock() bus clock calculation error in spi.c.
27. Fixed SPI\_SET\_xSB\_FIRST() implementation error in spi.h.
28. Fixed SYS\_IsRegLocked() return value inverted bug in sys.c.
29. Fixed TIMER\_Open() and TIMER\_Delay() time-out calculation if timer clock is faster than 0x1FFFFFF Hz in timer.c.
30. Fixed TIMER\_EnableCaptureDebounce() and TIMER\_DisableCaptureDebounce() implementation errors in timer.h.
31. Fixed TIMER\_CAPTURE\_RISING\_\* definition error in timer.h.
32. Fixed UART\_ClearIntFlag() bug that clears more flags than desired in uart.c.
33. Fixed UART\_Open(), UART\_SelectIrDAMode(), and UART\_SetLine\_Config() baudrate calculation bugs in uart.c.
34. Fixed UART\_Write() bug that data is not actually sent out in uart.c.
35. Fixed UART\_CLEAR\_RTS() implementation error in uart.h.
36. Fixed sample code bug that XTL is enabled before configuring multi-function pins.
37. Fixed ACMP\_TriggerTimerCapture sample bug that calculates input frequency based on incorrect frequency.
38. Fixed UART\_RS485 sample MODEM register and address setting error.
39. Added ADC\_SeqModeEnable() and ADC\_SeqModeTriggerSrc() APIs in adc.c.
40. Added CLK\_EnableSysTick() and CLK\_DisableSysTick() to control SysTick and select SysTick clock source in clk.c.
41. Added CLK\_CLKDIV\_UART0(), CLK\_CLKDIV\_UART1(), and CLK\_CLKDIV\_HCLK() macro definitions in clk.h.
42. Added SPI\_GET\_TX\_FIFO\_FULL\_FLAG() macro in spi.h.
43. Added PWM\_EnableRiseInt(), PWM\_DisableRiseInt(), PWM\_ClearRiseIntFlag(), PWM\_GetRiseIntFlag(), PWM\_GetCenterIntFlag(), and PWM\_ClearCenterIntFlag() functions in pwm.c.
44. Added SYS\_GET\_IRCTRIM\_INT\_FLAG() and SYS\_CLEAR\_IRCTRIM\_INT\_FLAG() macro definitions in sys.h.
45. Added SYS\_EnableIRCTrim() and SYS\_DisableIRCTrim() functions in sys.c.
46. Added CLK\_SwitchHCLK, I2C\_FIFO\_EEPROM, Timer\_Wakeup, SPI\_FIFO\_FLASH, and SYS\_TrimIRC samples.

#### Revision 3.00.000 (Released 2014-09-26)

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.*