

**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](http://www.nuvoton.com)

---

**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\_DisableAPUpdate(), FMC\_DisableSPUpdate(), FMC\_SetBootSource(), FMC\_DisableConfigUpdate(), FMC\_DisableLDUpdate(), FMC\_EnableAPUpdate(), FMC\_EnableSPUpdate(), 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\_CMCRD(), 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.*