

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

---

**Revision 3.02.002** (Released 2020-10-6)

---

1. Added Apache-2.0 license declaration into driver source code.
2. 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\_Id\_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.**

---

*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.*