

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

**NuMicro<sup>®</sup> Family**  
**NUC200 Series 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 3.00.004** (Released 2017-11-28)

1. Fixed USB zero packet issue.
2. Modified MFP setting style.

---

**Revision 3.00.003** (Released 2017-10-24)

3. Added CLK\_SysTickLongDelay() for long delay.
4. Fixed clear Receive Line Status interrupt flag bug in UART\_ClearIntFlag().
5. Modified to disable debug message when enabling semihost without NuLink connecting.
6. Fixed PLL clock source selection bug in CLK\_SetCoreClock().
7. Fixed UART\_SelectLINMode() clear enable bit setting bug.
8. Fixed a bug of u32RptDescLen calculation in USB\_GetDescriptor().
9. Added new function to control systick and select systick clock source, CLK\_EnableSysTick() and CLK\_DisableSysTick().
10. Fixed wrong SC1 and SC2 clock source select shift position in MODULE constant definitions.
11. Removed some combinations of I2C control bit settings. To avoid STOP and START write to control bit at the same time.
12. Revised I2C\_START(). When set STA bit, the SI doesn't need set at the same time.
13. Added ADC\_MeasureVADC() sample code.

---

**Revision 3.00.002** (Released 2015-05-13)

1. Fixed SC\_SET\_STOP\_BIT\_LEN define error.
2. Fixed all IAR samples to set entry point from \_\_iar\_program\_start to Reset\_Handler.
3. Fixed all samples that run faster than 50MHz. (NUC100 series only support up to 50MHz).
4. Fixed the wrong shift position for HCLK divider in main() of SYS sample code.
5. Fixed PLLCON\_SETTING constant define from SYSCLOCK\_PLLCON\_50MHz\_XTAL to CLK\_PLLCON\_50MHz\_HXT.
6. Fixed UA\_LIN\_CTL[4] bit field name is "MUTE\_EN" not "WAKE\_EN" in UART LIN\_CTL Bit Field Definitions.
7. Fixed CLK\_SetCoreClock() core lock range from "25~50MHz" to "25~72MHz".
8. Fixed CLK\_SysTickDelay() bug, that COUNTFLAG(SysTick\_CTRL[16]) may not be cleared after write SysTick\_VAL.
9. Fixed UA\_LIN\_CTL[4] bit field name of UART driver. It is "MUTE\_EN" not "WAKE\_EN" in UA\_LIN\_CTL constants definitions.
10. Fixed API declare name from I2C\_SetClockBusFreq() to I2C\_SetBusClockFreq() in I2C driver.
11. Fixed SYS\_IS\_SYSTEM\_RST() bug in SYS driver, it is "SYS\_RSTSRC\_RSTS\_SYS\_Msk" not "SYS\_RSTSRC\_RSTS\_MCU\_Msk".
12. Fixed definition bug of PDMA\_IS\_CH\_BUSY().
13. Fixed clear Time-out flag method bug in I2C\_ClearTimeoutFlag() of I2C driver.
14. Removed unused PWRCON, FREQ\_72MHZ constant definitions from clock driver.
15. Added WWDAT\_MODULE definition for CLK\_DisableModuleClock() and CLK\_EnableModuleClock().
16. Added SPI\_SET\_SS\_LEVEL() macro definition. This macro allows user to set both SPI\_SS pins.
17. Added a lack macro, SYS\_IS\_LVR\_RST() to SYS driver.
18. Added UART FIFO size constants definitions to UART driver.

19. Added CLK\_PLLCON\_25MHz\_HXT, CLK\_PLLCON\_25MHz\_HIRC, CLK\_PLLCON\_24MHz\_HXT, and CLK\_PLLCON\_24MHz\_HIRC constant definitions to CLK driver.
20. Added FMC\_MultiBoot\_SwReset sample code to show how to boot to different AP.
21. Modified time-out counter to a fix value and not to use SystemCoreClockUpdate() in CLK\_WaitClockReady() to improve compatibility.
22. Revised the following four macro definitions to avoid affecting another SPI\_SS pin, SPI\_SET\_SS0\_HIGH(), SPI\_SET\_SS1\_HIGH(), SPI\_SET\_SS0\_LOW() and SPI\_SET\_SS1\_LOW().

#### **Revision 3.00.001** (Released 2014-11-27)

---

1. First 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.*