ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® 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

Revision 3.00.005 (Released 2019-11-11)

1. Added ISP Sample codes to bsp\SampleCode\ISP folder.
2. Supports GNU GCC.
3. Added Mass Storage sample code to support SD Card.
4. Fixed PWM_DisableCaptureInt of PWM driver.
5. Fixed CLK_SetHCLK() bug of CLK driver.
6. Fixed CLK_EnablePLL() wrong PLL default setting value of CLK driver.

Revision 3.00.004 (Released 2017-11-28)

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

Revision 3.00.003 (Released 2017-10-24)

1. Added CLK_SysTickLongDelay() for long delay.
2. Fixed clear Receive Line Status interrupt flag bug in UART_ClearIntFlag().
3. Modified to disable debug message when enabling semihost without NuLink connecting.
4. Fixed PLL clock source selection bug in CLK_SetCoreClock().
5. Fixed UART_SelectLINMode() clear enable bit setting bug.
6. Fixed a bug of u32RptDescLen calculation in USBD_GetDescriptor().
7. Added new function to control systick and select systick clock source, CLK_EnableSysTick() and CLK_DisableSysTick().
8. Fixed wrong SC1 and SC2 clock source select shift position in MODULE constant definitions.
9. Removed some combinations of I2C control bit settings. To avoid STOP and START write to control bit at the same time.
10. Revised I2C_START(). When set STA bit, ths SI doesn't need set at the same time.
11. 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 SYSCLK_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 WWDT_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.