

ARM® Cortex®-M0
32-bit Microcontroller

NuMicro® Family
NANO100BN 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

Revision 3.02.002 (Released 2019-11-07)

1. Added ISP related samples.
2. Added sample USB_D_MassStorage_SDCard.
3. Minor bug fix.

Revision 3.02.002 (Released 2018-05-30)

1. Added Eclipse project support.
2. Minor bug fix.

Revision 3.02.002 (Released 2017-03-10)

1. Fixed INTR_T structure base address definition error.
2. Fixed smartcard driver and library behaviors that do not comply with EMV2000 spec.
3. Fixed CCID sample bug that incorrect error code is returned.
4. Fixed the bug that time-out interrupt flag is cleared at wrong time in I2C_Loopback sample code.
5. Fixed USB_D_Audio_Speaker sample code interrupt control error.
6. Added Interface Association Descriptor(IAD) in USB_D_Audio_Speaker_And_HID_Transfer sample code.
7. Updated SC_ReadSimPhoneBook sample code to support SIM card with CHV1 disabled.

Revision 3.02.001 (Released 2016-07-28)

1. Updated CMSIS to v4.5.0.
2. Added sample codes including USB_D_HID_Transfer_CTRL, USB_D_Vendor_LBK, and USB_D_Mass_Storage_SDCard.
3. Fixed the HIDTransferTest.exe bug to use correct sector size to compare data.
4. Updated HIDTransferTest.exe to support the connection of the composite device with HID interface number other than 0.
5. Fixed the USB_D_VCOM_SerialEmulator, USB_D_VCOM_DualPort, USB_D_VCOM_And_Mass_Storage, USB_D_VCOM_And_HID_Transfer, and USB_D_VCOM_And_HID_Keyboard samples bug to ensure SET_LINE_CODE command is properly handled.
6. Minor bug fixes.

Revision 3.02.000 (Released 2015-08-07)

1. Removed FMC driver's FMC_SetBootSource(), FMC_DisableAPUpdate(), FMC_DisableConfigUpdate(), FMC_DisableLDUpdate(), FMC_EnableAPUpdate(), FMC_EnableConfigUpdate(), FMC_EnableLDUpdate() in fmc.h, because there exist functionally identical macros.
2. Removed SYS_IRCTIMCTL_SEL_MASK, SYS_IRCTIMCTL_LOOP_MASK and SYS_IRCTIMCTL_RETRY_COUNT in sys.h.
3. Removed SPI_ENABLE_DUAL_MODE() in spi.h.
4. Modified SPI_ENABLE_DUAL_INPUT_MODE() and SPI_ENABLE_DUAL_OUTPUT_MODE() to enable dual I/O with direction.
5. Modified USB_D driver to pass USB Command Verify test in usbd.c and usbd.h.
6. Modified UART_SelectIrDAMode() to reload UART clock before calculating baudrate, in uart.c.
7. Modified SD card clock speed from 24 MHz to 5 MHz to make SPI operations stable in SDCard.c.
8. Modified MMC_FLASH_Init() to retry SD CMD0 command until success in SDCard.c.

9. Modified `TIMER_Open()` to not start timer in `timer.c`.
10. Updated `TIMER_Open()` and `TIMER_Delay()` to support extreme high clock input, in `timer.c`.
11. Renamed sample CRC to `CRC_CCITT` in `StdDriver`.
12. Renamed sample GPIO to `GPIO_IOTest` in `StdDriver`.
13. Renamed sample PDMA to `PDMA_Memory` in `StdDriver`.
14. Renamed sample SYS to `SYS_Control` in `StdDriver`.
15. Renamed `SYS_Int_xxx_Msk` to `SYS_xxx_Msk` in `sys.h`.
16. Renamed `GP_DBNCECON_PUEN_*` to `GP_DBNCECON_DBCLKSEL_*` in `Nano100Series.h`.
17. Renamed `SYS_IRCTRIMINT_32KERR_ENNT` to `SYS_IRCTRIMINT_32KERR_INT` in `sys.h`.
18. Fixed the bug that `RTC_AER` enable flow may be interrupted by interrupt service routine in `rtc.c`.
19. Fixed the bug that `PWM_ENABLE_OUTPUT_INVERTER()` does not clear register field before writing input parameter to it in `pwm.h`.
20. Fixed the bug that `TIMER_Delay()` sets prescale to wrong register in `timer.c`.
21. Fixed `SCUART_Open()` and `SCUART_SetLineConfig()` baudrate calculation prescale setting error in `scuart.c`.
22. Fixed bugs of `SPI_EnableAutoSS()` and `SPI_SetBusClock()` in `spi.c`, and cleared bit mask of register field before writing input parameter to it.
23. Fixed the `CLK_SysTickDelay()` bug that continuously calling `CLK_SysTickDelay()` may imply an incorrect delay time by clearing control register on each call in `clk.c`.
24. Fixed implementation errors of `CLK_PLLCTL_FB_DV_Msk` and `CLK_APBCLK_I2C0_EN` in `Nano100Series.h`.
25. Fixed `SYS_CLEAR_RST_SOURCE` implementation error in `sys.h`.
26. Fixed `CLK_WK_INTSTS_IS` implementation error in `clk.h`.
27. Fixed "GPIO_DISABLE_DOUT_MASK" and "GPIO_ENABLE_DOUT_MASK" implementation errors in `gpio.h`.
28. Fixed `SC_SET_STOP_BIT_LEN` implementation error in `sc.h`.
29. Fixed `PDMA_IS_CH_BUSY` implementation error in `pdma.h`.
30. Fixed `LCD_CPUMP_DIV128` implementation error in `lcd.h`.
31. Fixed `RTC_CLEAR_TAMPER_FLAG()` implementation bug in `rtc.h`.
32. Fixed `ADC_SET_DMOF()` implementation error in `adc.h`.
33. Fixed `CRC_SET_SEED()` implementation error in `crc.h`.
34. Disabled Rx before raising RST high during cold reset in `SmartCardLib` library.
35. Checked `SC_RST` and `SC_DAT_O` pin status during deactivation in `SmartCardLib` library.
36. Added `SYS_PA_H_MFP_PA9_MFP_LCD_S7` macro in `sys.h`.
37. Added `I2C_ClearIntFlag()` and `I2S_SetFIFO()` functions in `i2c.c`.
38. Added bit definitions of `MCLKO` (Module Clock CKO) register in `clk.h`.
39. Added `LCD_MODULE` and `DAC_MODULE` macro sets in `clk.c`.
40. Added `CLK_EnableSysTick()` and `CLK_DisableSysTick()` in `clk.c`.
41. Added macros `CLK_PLLCTL_*MHz_HXT` and `CLK_PLL_*MHz_HIRC` for setting `PLLCTL` value in `clk.c`.
42. Added `SYS_EnableIRCTrim()` and `SYS_DisableIRCTrim()` functions in `sys.c`, and added macros `SYS_GET_IRCTRIM_INT_FLAG()` and `SYS_CLEAR_IRCTRIM_INT_FLAG()` in `sys.h`.
43. Added `UART_SelectLINMode()` in `uart.c`, and added `UART_FUNC_SEL_LIN` in `uart.h`.
44. Added a sample `USBD_Audio_Speaker_And_HID_Transfer` to `Nu-LB-NANO130`.

45. Added samples SYS_MCLKO, SYS_PLLClockOutput, and SYS_TrimIRC to StdDriver.
46. Added a sample Timer_Wakeup to StdDriver.
47. Added samples USBD_HID_Keyboard, USBD_HID_MouseKeyboard, USBD_HID_Touch, USBD_HID_Transfer_And_Keyboard, USBD_HID_Transfer_And_MSC, USBD_Mass_Storage_CDRom, USBD_Micro_Printer, USBD_Printer_And_HID_Transfer, USBD_VCOM_And_HID_Keyboard, USBD_VCOM_And_HID_Transfer, USBD_VCOM_And_Mass_Storage, USBD_VCOM_DualPort, and USBD_VCOM_SerialEmulator to StdDriver.

Revision 3.01.000 (Released 2014-09-19)

1. Renamed register TESTCLK to MCLKO.
2. Renamed registers PDSSR0 and PDSSR1 to DSSR0 and DSSR1.
3. Renamed USBD_ENABLE_INT() to USBD_ENABLE_INT().
4. Renamed I2S_Enable_MCLK()/I2S_Disable_MCLK() to I2S_EnableMCLK()/I2S_DisableMCLK().
5. Renamed CLK_CLKSEL1_PWM1_CH01_S_Msk/CLK_CLKSEL1_PWM1_CH23_S_Msk to CLK_CLKSEL2_PWM1_CH01_S_Msk/CLK_CLKSEL2_PWM1_CH23_S_Msk.
6. Renamed RTC_RIIR_SNOOPIS_Msk to RTC_RIIR_SNOOPIF_Msk.
7. Renamed PDMA_IER_BLKD_IE_Msk to PDMA_IER_TD_IE_Msk.
8. Modified PWM_EnablePDMA() function prototype, and added one more parameter to select captured edge.
9. Modified PWM capture interrupt flag relative macro definitions to improve performance.
10. Added ADC clock source bit position and mask definition.
11. Added ADC_SET_REF_VOLTAGE() macro and RES/REF definitions.
12. Added DAC driver.
13. Added ADC_Compare, ADC_TimerTrigger, ADC_PDMA, DAC_PDMATrigger, DAC_SoftwareTrigger, DAC_TimerTrigger, GPIO_PowerDown, Hard_Fault_Sample, PWM_CapturePDMA, SPI_LoopbackPDMA, UART_FlowCtrl, UART_TxRxPDMA, and USBD_HID_Transfer samples.

Revision 3.00.000 (Released 2014-02-20)

1. Update major version number from 2 to 3.
2. Renamed RTC_GetDatAndTime() to RTC_GetDateAndTime().

Revision 2.00.000 (Released 2014-01-11)

1. Primary release version.

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.