ARM® Cortex®-M

32-bit Microcontroller

# NuMicro™ Family
# NUC123 Series BSP
# Revision History



For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

## Revision 3.01.003 (Released 2021-01-20)

1. Modified to pass USB-IF CV-Chapter 9 & Class test of all USBD Sample code.
2. Added Apache-2.0 license declaration in driver source.
3. Added README.md file.

## Revision 3.01.002 (Released 2019-11-11)

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

## Revision 3.01.001 (Released 2016-06-22)

1. Fix wrong system clock settings in USB device sample codes.
2. Fixed u32PinMask parameter setting error in GPIO_SetMode API.
3. Update CMSIS to v4.5.0.
4. Update USB device driver to improve compatibility.
5. Update USB Virtual COM port driver for WHQL certification.
6. Update I2C driver to add byte write API for I2C master.
7. Update retarget.c to support user defined hard fault handler.
8. Update Hard_Fault_Sample sample code for user defined hard fault handler.
9. Modify USBD_VCOM_SinglePort to support both UART0 or UART1.
10. Rename USBD_VCOM sample code to USBD_VCOM_SinglePort sample code.
11. Remove hardware debounce sample code.
12. Add ADC_MeasureAVDD sample code.
13. Add I2C_MultiBytes_Master and I2C_SingleByte_Master sample code.
14. Add GPIO_SwDebounce sample code.
15. Add USBD_VCOM_DualPort sample code.
16. Add USBD_VCOM_and_MassStorage sample code.
17. Add USBD_VCOM_and_HID_Keyboard sample code.
18. Add USBD_Printer_and_HID_Transfer sample code.
19. Add USBD_Micro_Printer sample code.
20. Add USBD_HID_Transfer_and_MSC sample code.
21. Add USBD_HID_MouseKeyboard sample code.
22. Add USBD_HID_Transfer_and_Keyboard sample code.

## Revision 3.00.003 (Released 2015-07-02)

1. Fix lost parentheses for PDMA_IS_CH_BUSY() of PDMA driver.
2. Fix API declare name from I2C_SetClockBusFreq() to I2C_SetBusClockFreq() in I2C driver.
3. Fix bug of PWM_ConfigOutputChannel() for duty is 0 in PWM driver.
4. Fix clear flag bug to clear one flag in one time in UART_ClearIntFlag() in UART driver.
5. Fix clear RS-485 address byte detection flag bug to clear one flag in one time in RS485_HANDLE() of UART_RS485_Slave sample code.
6. Fix clear RS-485 address byte detection flag bug to clear one flag in one time in UART_RS485_CLEAR_ADDR_FLAG() of UART driver.
7. Fix clear Time-out flag method bug in I2C_ClearTimeoutFlag() of I2C driver.
8. Fix CLK_SysTickDelay() bug of CLK driver, COUNTFLAG(SysTick_CTRL[16]) may not be cleared after write SysTick_VAL.

9. Fix CLKSEL0 setting bug in CLK_SetCoreClock() of CLK driver.
10. Fix GPIO_ENABLE_DOUT_MASK() and GPIO_DISABLE_DOUT_MASK(). Their define are inversed in GPIO driver.
11. Fix I2C_Close implementation error in I2C driver.
12. Fix Multi-Function constant definitions error of PB.2, PB.5, PC.0, PC.3 in SYS driver.
13. Fix PWM_EnableADCTrigger () and PWM_EnablePDMA() implementation error in PWM driver.
14. Fix SYS_IS_SYSTEM_RST() bug, it is 'SYS_RSTSRC_RSTS_SYS_Msk' not 'SYS_RSTSRC_RSTS_MCU_Msk' in SYS driver.
15. Fix UART RTS LEVEL TRIGGER active level definition bug in UART driver.
16. Fix UART transmit data bug in UART_TEST_HANDLE() of UART_TxRx_Function sample code.
17. Modify to support NUC123xxxAEx.
18. modify HCLK clock setting bug in CLK_SetCoreClock() of CLK driver.
19. Remove ADC_ADCHER_PRESEL_INT_TEMPERATURE_SENSO definetion;
20. Remove an extra ')' for GPIO_ENABLE_DOUT_MASK().
21. Replace USBD_SetStall() with USBD_SET_EP_STALL().
22. Revise FMC_Erase()  ISPFF flag clear.
23. Revise the following four macro definitions to avoid affecting another SPI_SS pin. SPI_SET_SS0_HIGH() SPI_SET_SS1_HIGH() SPI_SET_SS0_LOW() SPI_SET_SS1_LOW()
24. Add 144MHz PLL setting (HXT source) definition to CLK driver.
25. Add a lack macro, SYS_IS_LVR_RST().
26. Add a lost ')' for GPIO_DISABLE_DOUT_MASK().
27. Add constant define 'CLK_CLKSEL0_STCLK_S_HCLK' in CLKSEL0 constant definitions for CLK_EnableSysTick() function to select HCLK as sysTick clock source.
28. Add default hard fault handler in retarget.c
29. Add Hard_Fault_Sample sample code for hard fault handler demo.
30. Add lacked 'PUBWEAK HardFault_Handler' to startup_NUC123.s
31. Add new function to control systick and select systick clock source, CLK_EnableSysTick() and CLK_DisableSysTick() of CLK driver.
32. Add new function to control systick and select systick clock source, CLK_EnableSysTick() and CLK_DisableSysTick().
33. Add nonblocking printf implement and use predefine compiler option to enable/disable it in retarget.c
34. Add one more zero packet when BULK IN trasnfer is end by max packet size packet at last packet in USBD_VCOM sample code.
35. Add SPI_SET_SS_LEVEL() macro definition. This macro allows user to set both SPI_SS pins.
36. Add SPI_SlaveDualIOMode sample code to demonstrate SPI dual IO mode.
37. Add UART FIFO size constants definitions in UART driver.
38. Add UART_Wakeup sample code to demonstrate UART wakeup function.
39. Add USBD_Audio_HID_Transfer sample code to support HID Transfer + Audio composite device.

## Revision 3.00.002 (Released 2014-10-17)

1. add USBD_Audio_HID_NAU8822
2. USBD_Audio_HID_NAU8822, Fix length of "UNIT ID 5: Feature Unit"
3. USBD_Audio_HID_NAU8822, Modify the MIC gain to maximum.
4. USBD_Audio_HID_NAU8822, Modify to support keyboard and media key
5. USBD_Audio_HID_NAU8822, Modify I/O configure to support NuEdu-NUC123

6. USBD_VCOM, Modify VID to B002 to match CDC inf driver
7. USBD_VCOM, Modify for win8 certification
8. Modify to support UAC+HID
9. I2C, Modify PLL CLock to 72M
10. Fix FMC_Erase

## Revision 3.00.001 (Released 2014-07-30)

1. Update to support new driver API

## 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.**