ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® Family
NUC230/NUC240 Series BSP
Revision History

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

www.nuvoton.com

Revision 3.01.007 (Released 2025-12-23)

1. Update ISP_SPI sample to add interrupt mask protection for ParseCmd().
2. Add software disconnect and reconnect API in USBD sample code.
3. Add a macro to wait for the I2C bus status to be cleared.
4. Add the macro to the sample code to avoid re-entering the interruption before the register write is complete, which could lead to an incorrect I2C_STATUS0 value of 0xf8.
5. Add read/write byte related functions to the I2C driver.
6. Disable I2C IRQ until ParseCmd() is completed to prevent ISP from triggering an interruption and fetching data prematurely in ISP_I2C sample code.
7. Add USBD_HID_Mouse2 sample code.
8. Add Schmitt trigger configuration for I2C pins.
9. Fix infinite loop.

Revision 3.01.006 (Released 2024-05-22)

1. Added timeout handler for infinite loop.
2. Added CAN_BasicMode_SilentMode and CAN_NormalMode_SilentMode sample code.
3. Used "volatile" with the function pointer to disable compiler optimizations in I2C Sample code.
4. Updated the sc library to solve SC_ReadATR and USBD_CCID running fail.
5. Waited all Tx data transmission finished before changing parity setting in UART RS485 sample code.
6. Fixed inverse definition of RTS active level in UART RS485 sample code.
7. Enabled LVR in SYS_PowerDown_MinCurrent sample code to prevent power on/off fail.
8. Updated USBD HID keyboard sample code to support LED status.
9. Modified VendorName of Nu_DFU.inf
10. Added SYS_PowerDown_MinCurrent sample code.
11. Called UART FIFO size from uart.h in USBD VCOM sample code.

Revision 3.01.005 (Released 2022-08-17)

1. Fixed USBD_MassStorage_CDROM crash on Linux.
2. Added I2C hang up and recover mechanism for I2C Master and Slave sample code.
3. Updated project settings of timer capture sample code.
4. Fixed UART TX FIFO control issue in USBD_VCOM sample code.

Revision 3.01.004 (Released 2021-01-22)

1. Fixed SetMultiRxMsg MsgCount in Library/StdDriver/src/can.c
2. Fixed data access fail issue of USBD_Mass_Storage_CDROM sample code.
3. Added SPI_TRIGGER_TX_RX_PDMA API.
4. Added to pass USB-IF CV-Chapter 9 & Class test of all USBD Sample code.
5. Fixed warnings of adc driver in Library/StdDriver/src/adc.c
6. Added Apache-2.0 license declaration in driver source.
7. Added README.md file.

Revision 3.01.003 (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 3.01.002 (Released 2017-10-24)

1. Fixed PLL clock source selection bug in CLK_SetCoreClock().
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. Added CAN_SetRxMsgObjAndMsk() function.
5. Added CLK_SysTickLongDelay() for long delay.
6. Added ADC_MeasureVADD() sample code.

Revision 3.01.001 (Released 2015-08-18)

1. Fixed the clock selection bug in SCUART_TxRx() sample code.
2. Fixed the reset handler from __iar_program_start to Reset_Handler.
3. Fixed the bug of clearing wrong enable bit in UART_SelectLINMode() of UART driver.
4. Fixed CAN_STATUS_LEC_Msk from 0x03 to 0x07.
5. Fixed SC_UACTL_UA_MODE_EN_Msk define error from (3ul << SC_UACTL_UA_MODE_EN_Pos) to (1ul << SC_UACTL_UA_MODE_EN_Pos).
6. Fixed the wrong SC1 and SC2 clock source select shift position in MODULE constant definitions.
7. Fixed the wrong definition bug of PS2_DISABLE_OVERRIDE() and PS2_ENABLE_OVERRIDE().
8. Fixed the bug of PS2_Write() in PS2 driver.
9. Removed ReadDID() function from FMC driver as It is no longer supported.
10. Updated USB driver to improve reliability and compatibility.
11. Added NuEdu sample code.
12. Added INT 'MCUIRQ' and 'MCUIRQCR' bit field definitions.
13. Added CLK_EnableSysTick() and CLK_DisableSysTick() to control SysTick and select SysTick clock source.
14. Added SPI_FIFO_SIZE constant definition.
15. Added USB device sample code.
16. Improved USBD driver for adding more USB sample code.

Revision 3.00.002 (Released 2015-05-11)

1. Fixed the wrong baud rate returned by SCUART_SetLineConfig() in SCUART driver.
2. Fixed SCUART_Open() of SCUART driver for wrong clock calculation and return value.
3. Fixed SC_SET_STOP_BIT_LEN define error.
4. Fixed the bug of ADC_IS_DATA_OVERRUN() that the input parameter is channel number but channel bit mask.
5. Updated CAN clock setting in CAN_Open() of CAN driver to comply with different system clocks.
6. Fixed the wrong return value bug of CLK_SetCoreClock() in CLK driver.
7. Fixed CLK_SetModuleClock() error for PWM clock selection in CLK driver.
8. Fixed the bug of CLK_SysTickDelay() that COUNTFLAG may not be cleared in CLK driver.

9. Fixed the GPIO_ENABLE_DOUT_MASK() and GPIO_DISABLE_DOUT_MASK() implement inverse error in GPIO driver.
10. Fixed the close wrong I^{2}C bug of I2C_Close() in I^{2}C driver.
11. Fixed API declare name from I2C_SetClockBusFreq() to I2C_SetBusClockFreq() in I^{2}C driver.
12. Fixed the clear RS-485 address byte detection flag bug to clear one flag at one time in RS485_HANDLE() of UART driver.
13. Added one more zero packet when BULK IN transfer is end by max packet size packet at last packet in VCOM sample code.
14. Fixed UA_LIN_CTL[4] bit field name as 'MUTE_EN' not 'WAKE_EN' in UART LIN_CTL bit field definitions of header file.
15. Fixed the wrong mask definition of SC_TRSR_TX_POINT_F_Msk and SC_TRSR_RX_POINT_F_Msk in header file.
16. Fixed the wrong bit definition of WAKEUP_EN in USB_INTEN register of header file.
17. Fixed the channel 0 trigger disabled bug when channel 1~3 trigger enabled in PWM_EnableADCTrigger() of PWM driver.
18. Fixed the bug that channel 0 trigger will be disabled when channel 1~3 trigger are enabled in PWM_EnableADCTrigger() of PWM driver.
19. Fixed SCUART_PARITY_NONE/SCUART_PARITY_EVEN/SCUART_PARITY_ODD definition bug in SCUART driver.
20. Fixed four macro definitions of SPI driver to avoid affecting non-target SPI_SS pin including SPI_SET_SS0_HIGH() SPI_SET_SS1_HIGH() SPI_SET_SS0_LOW() SPI_SET_SS1_LOW().
21. Fixed the clear flag bug to clear one flag at one time in UART_ClearIntFlag(). It should be '(uart)->FSR = UART_FSR_RS485_ADD_DETF_Msk' but '(uart)->FSR |= UART_FSR_RS485_ADD_DETF_Msk'.
22. Fixed the clear wrong flag bug in UART_RS485_CLEAR_ADDR_FLAG().
23. Fixed UA_LIN_CTL[4] bit field name as 'MUTE_EN' not 'WAKE_EN' in UA_LIN_CTL constants definitions in UART driver.
24. Added SPI_SET_SS_LEVEL() macro definition in SPI driver to allow user to set both SPI_SS pins.
25. Added a lack macro SYS_IS_LVR_RST() to SYS driver.

Revision 3.00.001 (Released 2014-05-05)

1. Supported NUC230/NUC240
2. Supported latest NuMicro Family Driver API.
3. Supported register based and driver based sample code.
4. Added USB CCID Card Reader sample code

Revision 1.01.001 (Released 2013-07-25)

1. Supported NUC230_240 BN version.
2. Split driver samples into small one for easy reference.
3. Added readme file in root directory for quick reference for BSP directory and sample code.

Revision 1.00.002 (Released 2012-11-09)

1. Fixed CAN, Smart Card, and GPIO driver bugs.
2. Modified the unlock function to do retry when fail to unlock.

Revision 1.00.001 (Released 2012-08-04)

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.