ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® Family
NUC1261 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 3.00.006 (Released 2026-07-07)

1. Fixed RegBased/FMC_IAP sample code VSCode project run code issue.
2. Updated ISP_SPI sample to add interrupt mask protection for ParseCmd.
3. Added software disconnect and reconnect API in USBD Sample code.
4. Supported VSCode multi-toolchain project.
5. Added a check for the STOP bit when I2C transmission is complete.
6. Sent NAK instead of 0 to fix host not going to sleep issue in USBD_HID_Mouse2 sample code.
7. Added a macro to wait for the I2C bus status to be cleared.
8. Added the macro to the sample code to avoid re-entering the interrupt before the register write is complete, which could lead to an incorrect I2C_STATUS0 value of 0xf8.
9. Fixed infinite loop.

Revision 3.00.005 (Released 2024-05-28)

1. Revised some SC sample codes and update SC library.
2. Fixed the issue causing execution failure in the USCI-I2C sample code by correcting the erroneous status return.
3. Added FMC_SPROM sample code.
4. Added slew rate control for SPI sample codes.
5. Used "volatile" with the function pointer to disable compiler optimizations in I2C/UI2C Sample code.
6. Fixed UI2C_Multibytes_Master and Singlebyte_Master bug issue.
7. Modified SYS_PowerDown_MinCurrent and USCI_UART_TxRxFunction sample code.
8. Updated USBD HID keyboard sample code to support LED status.
9. Added driver and sample code related to USCI_SPI PDMA function.
10. Added SYS_PowerDown_MinCurrent sample code.
11. Modified UART_Read() to return received data count when time-out.
12. Modified SYS_UnLockReg() time-out handler.
13. Called UART FIFO size from uart.h in USBD VCOM sample code.
14. Fixed TX hardware FIFO size to 16 in USBD VCOM sample code.
15. Added SYS driver PE0MFP(SYS_GPE_MFPL[3:0]) bit field definition for NUC1261.

Revision 3.00.004 (Released 2022-08-17)

1. Added PE.0 in Library/Device/Nuvoton/NUC1261/Include/NUC1261.h.
2. Fixed return in main() cause hardfault issue.
3. Added timeout handler for infinite loop.
4. Fixed USBD_MassStorage_CDROM crash on Linux.
5. Changed the pin selection in USBD sample code.
6. Fixed build errors of FMC and I2S sample code.
7. Changed the pin selection in the relevant EBI and TIMER sample code.
8. ACMP output is changed from PD6 to PD7 in ACMP sample code.
9. Rebuilt IAR smartcard lib by IAR 8.xx.
10. Added I2C hang up and recover mechanism for I2C Master and Slave sample code.
11. Modified GPIO number and MFP in Library/Device/Nuvoton/NUC1261/Include/NUC1261.h.
12. Enabled schmitt trigger of I2C sample code.
13. Fixed UART TX FIFO control issue in USBD_VCOM sample code.

Revision 3.00.003 (Released 2021-01-21)

1. Modified to pass USB-IF CV-Chapter 9 & Class test of all USBD sample code.
2. Fixed Semihost sample code.
3. Added SPI_TRIGGER_TX_RX_PDMA and SPI_DISABLE_TX_RX_PDMA API.
4. Added Apache-2.0 license declaration in driver source.
5. Added README.md file.
6. Fixed USCI_SPI driver issue in Library/StdDriver/src/usci_spi.c
7. Fixed USPI_SET_SS_HIGH macro in Library/StdDriver/inc/usci_spi.h
8. Corrected adc definitions for conversion sampling time selection in Library/StdDriver/inc/adc.h

Revision 3.00.002 (Released 2019-11-12)

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. Added FMCIDLE_MODULE definition in CLK driver.

Revision 3.00.001 (Released 2018-09-21)

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.

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.