

ARM® Cortex®-M4
32-bit Microcontroller

M480 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 3.05.002 (Released 2021-12-28)

1. Add time-out check to infinite loop in drivers and samples.
2. Enable I2C and USCI_I2C pin schmitt trigger.
3. Modify HSUSBD HID samples to pass USBIF.
4. Add remote wakeup to HID mouse sample.
5. Minor bug fix.

Added EPWM_EnableADCTriggerPrescale(), EPWM_DisableADCTriggerPrescale(), Revision 3.05.001 (Released 2020-10-8)

6. Added Apache-2.0 license declaration into driver source code.
7. Added APROM_Loader, CCAP_Packet_JpegEncode, I2C_SMBus, and ISP_UART_SPIFLASH_M487MK sample codes.
8. Added EPWM_EnableADCTriggerPrescale(), EPWM_DisableADCTriggerPrescale(), EPWM_EnableFaultDetect(), EPWM_DisableFaultDetect(), EPWM_EnableFaultDetectOutput(), EPWM_DisableFaultDetectOutput(), EPWM_EnableFaultDetectDeglitch(), EPWM_DisableFaultDetectDeglitch(), EPWM_EnableFaultDetectMask(), EPWM_DisableFaultDetectMask(), EPWM_EnableFaultDetectInt(), EPWM_DisableFaultDetectInt(), EPWM_ClearFaultDetectInt(), and EPWM_GetFaultDetectInt() APIs.
9. Added USPI_DISABLE_TX_RX_PDMA() and USPI_ENABLE_TX_RX_PDMA(), QSPI_TRIGGER_TX_RX_PDMA, QSPI_DISABLE_TX_RX_PDMA(), SPI_TRIGGER_TX_RX_PDMA, SPI_DISABLE_TX_RX_PDMA(), UART_TRIGGER_RX_PDMA(), UART_TRIGGER_TX_PDMA(), UART_DISABLE_RX_PDMA(), and UART_DISABLE_TX_PDMA() macro definitions.
10. Minor bug fix.

Revision 3.05.000 (Released 2020-3-6)

1. Added M487KMCAN support.
2. Added QSPI_Slave3Wire, SPI_HalfDuplex, USBD_Mass_Storage_SD, HSUSBD_Mass_Storage_SD, HSUSBH_USBH_HID_Mouse_Keyboard, HSUSBD_Audio10_Headset, HSUSBD_Audio20_Headset, ISP_SPI, ISP_DFU, and ISP_DFU_20 sample codes.
3. Added SPIM_WinbondUnlock(), EMAC_FillCamEntry(), EMAC_SendPktWoCopy(), EMAC_ClaimFreeTXBuf(), EMAC_GetAvailRXBufSize(), and EMAC_RecvPktDoneWoRxTrigger() APIs.
4. Added TIMER_SET_OPMODE(), UART_PDMA_ENABLE(), UART_PDMA_DISABLE(), UART_PDMA_ENABLE(), UART_PDMA_DISABLE(), EMAC_ENABLE_INT(), EMAC_DISABLE_INT(), EMAC_GET_INT_FLAG(), and EMAC_CLEAR_INT_FLAG() macro definitions.
5. Removed static qualifier of EMAC_PhyInit() which must now be called after EMAC_Open().
6. Updated UsbHostLib to improve USB 2.0 interrupt transfer performance.
7. Minor bug fix.

Revision 3.04.000 (Released 2019-7-5)

1. Added M48xGC/M48xG8 support.
2. Reorganized sample directory for NuMaker-ETM-M487, NuMaker-PFM-M487, and NuMaker-PFM-M487D boards.
3. Minor bug fix.

Revision 3.03.001 (Released 2018-10-5)

1. Added emWin quick start guide.
2. Fixed GCC compilation errors in the Linux environment.

Revision 3.03.000 (Released 2018-05-30)

1. Added emWin sample codes and library.
2. Updated Eclipse project directory structure.
3. Updated audio sample MFP setting to match current NuMaker board design.
4. Minor bug fix.

Revision 3.02.000 (Released 2018-03-20)

1. Updated sample codes to use PB12(RX) and PB13(TX) instead of PD2(RX) and PD3(TX) as console pins for NuMaker-PFM-M487 v3.0 board.
2. Updated FatFs to r0.13a.
3. Updated FreeRTOS to v10.0.0.
4. Updated lwIP to v2.0.3.
5. Added a USBH CDC class driver and a CDC class VCOM sample.
6. Added mbed TLS and test suites using crypto accelerator.
7. Minor bug fix.

Revision 3.01.000 (Released 2017-09-29)

1. Reorganized register definition header files.
2. Added new standard driver reference samples.
3. Added Eclipse project support.
4. Renamed SPI0 to QSPI0, SPI1 to SPI0, SPI2 to SPI1, SPI3 to SPI2, SPI4 to SPI3, ACMP to ACMP01.
5. Rewrote USBH library to optimize memory usage.
6. Minor bug fix.

Revision 3.00.000 (Released 2017-06-15)

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.*