ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® Family
M051 Series BSP
Revision History

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

www.nuvoton.com

Revision 3.01.005 (Released 2024-08-02)

1. Set [clear ADF flag operation] to the first operation in ADC_IRQHandler.
2. Fixed multi-function pin setting of SPI sample code.
3. Used "volatile" with the function pointer to disable compiler optimizations in I2C sample code.
4. Enabled LVR in SYS_PowerDown_MinCurrent sample code to prevent power
5. on/off fail.
6. Added time-out handler for infinite loop.
7. Fixed uart.c parentheses issue.
8. Added SYS_PowerDown_MinCurrent sample code.
9. Updated SPI_Loopback sample code.
10. Added I2C hang up & recover mechanism for I2C Master and Slave sample code.

Revision 3.01.003 (Released 2019-11-07)

1. Added ISP Sample codes to bsp\SampleCode\ISP folder.
2. Revised to do ADC_POWER_ON() before ADC_Open() for ADC StdDriver sample codes.
3. Added GCC projects.
4. Fixed timer frequency inaccurate issue.
5. Fixed PWM_DisableCaptureInt error.
6. Fixed CLK driver CLK_SetHCLK() bug: to enable HIRC before switch HCLK and disable HIRC if it is not enabled before.
7. Fixed CLK_EnablePLL() wrong PLL default setting value.
8. Revised MFP coding style and deleted useless files.
9. Modified UART/I2C MFP Setting.
10. Revised multi-function pins settings:
    1. To avoid overwrite the previous correct settings.
    2. Impact on EBI, TIMER, WDT ,WWDT and SPI sample codes.
11. Fixed MFP configuration in Hard Fault sample code.

Revision 3.01.002 (Released 2017-10-05)

1. Fixed clear Receive Line Status interrupt flag bug in UART_ClearIntFlag().
2. Modified to ignore debug message when enabling semihost without connecting NuLink ICE.
3. Fixed PLL clock source selection bug in CLK_SetCoreClock().
4. Add CLK_SysTickLongDelay() for long delay in clk.h.

Revision 3.01.001 (Released 2015-08-04)

1. Fixed the bug about wrong TIMER_Delay pre-scale setting in TIMER driver.
2. Fixed the bug about wrong P4.5 mode setting in main() of GPIO_INT sample code.
3. Fixed the bug about wrong I^2C clock rate calculation in I2C_EEPROM, I2C_GCMode_Slave and I2C_GCMode_Master sample code.
4. Fixed the active level definition bug of UART_RTS_LEVEL_TRIGGER in UART driver.
5. Fixed the implement bug of SYS_CLEAR_RST_SOURCE() in SYS driver.
6. Fixed the bug that SPI peripheral clock is not disabled in SPI_Close( ) of SPI driver.
7. Fixed the shift position bug in PWM_ConfigCaptureChannel() of PWM driver.
8. Fixed the PWM output bug in PWM driver that when PWM duty is set to 0 by PWM_ConfigOutputChannel() in PWM driver.

9. Fixed the pin mask control bug of GPIO_ENABLE_DOUT_MASK() and GPIO_DISABLE_DOUT_MASK() in GPIO driver.
10. Fixed the bug that IRQ is not disabled after chip wake-up in I2C_Wakeup_Slave sample code
11. Fixed IAR reset entry as Reset_Handler for all sample code.
12. Fixed the HIRC clock switching bug of CLK_SetCoreClock() in CLK driver.
13. Fixed the bug that FMC_Erase() disables ISP function when erase error occurred in FMC driver.
14. Fixed the bug that delay time may not be correct in CLK_SysTickDelay() of CLK driver.
15. Fixed the bug that CLK_EnableModuleClock() does not enable clock source when enabling the WWDT module.
16. Fixed the bug of RS485_HANDLE() in the UART_RS485_Slave sample code to only clear one flag at one time.
17. Fixed the bug of UART_RS485_CLEAR_ADDR_FLAG() in UART driver to only clear one flag at one time.
    Fixed the bug of UART_ClearIntFlag() in UART driver to only clear one flag at one time.
18. Fixed the macro implement bug of ADC_IS_DATA_OVERRUN in ADC driver.
19. Fixed a bug on SPI_CLR_UNIT_TRANS_INT_FLAG( ) definition, and removed a redundant right parenthesis.
20. Added UART_Wakeup sample code to show how to wake up system by UART.
21. Added SYS_IS_LVR_RST() to support LVR reset status in SYS driver.
22. Added SPI_Loopback sample code for SPI0 simple data loopback demo.
23. Added RX1, TX1, and VARCLK registers to SPI_T to support M05xxBN in the header file.
24. Added multi-function constant definitions SYS_MFP_P02_TXD, SYS_MFP_P03_RXD, SYS_MFP_P30_RXD, SYS_MFP_P31_TXD, SYS_MFP_P32_nINT0, and SYS_MFP_P33_nINT1.
25. Added GC mode constant definitions for I2C_SetSlaveAddr() in I^2C driver.
26. Added FMC_MultiBoot_SwReset sample code to show how to boot from a different application using VECMAP and software jump.
27. Added CLK_GetPCLKFreq() function to CLK Driver to support getting PCLK clock frequency.
28. Removed unsupported register MCUIRQ from GCR_INT_T.

Revision 3.00.002 (Released 2014-07-18)

1. Fixed constant definitions of Timer 2 and Timer 3 in clk.h
2. Fixed SYS_Init() GPIO initial bug of \SampleCode\StdDriver\ACMP.

Revision 3.00.001 (Released 2014-02-14)

1. Updated all driver, API, and relative sample code.
2. Updated CMSIS to v3.01.
3. Changed directory structure.

Revision 2.02.004 (Released 2014-02-14)

1. Modified ACMP driver sample code to show message by semihost to avoid UART0/ACMP I/O conflict.
2. Fixed no sound issue of ADC_PWM of learning board sample code.
3. Fixed no message on LCD issue of Idle of learning board sample code.

4. Fixed FMC and UART_LIN target device setting.

Revision 2.02.003 (Released 2014-02-11)

1. Moved semihost relative code to retarget.c and add semihost sample code.
2. Fixed macro definition bug of PWM driver.

Revision 2.02.002 (Released 2013-08-22)

1. Supported M05xxDE.
2. Modified ACMP names (e.g. Renamed ACMPA to ACMP01, ACMPB to ACMP23).
3. Renamed Hardware divider prefix name from DIV to HDIV.

Revision 2.02.001 (Released 2013-06-10)

1. Supported new hardware functions of M05xxDN.
2. Modified and add more driver sample code for M05xxDN.
3. Added ACMP.h
4. Removed 24MHz macro setting for PLLCON.

Revision 2.01.002 (Released 2012-08-10)

1. Fixed the _TIMER_RESET definition.
2. Fixed the SYSCLK_CLKSEL1_PWM23_HCLK definition.
3. Removed the IARv6 directory and supported only IAR v6.21 and higher versions.
4. Modified PLL enable procedure to avoid unstable condition caused by PLL frequency changes.

Revision 2.01.001 (Released 2012-04-24)

1. Provided a new driver for faster performance with smaller code size.
2. Created a new directory hierarchy.

Revision 1.02.003 (Released 2012-03-12)

1. Fixed the UART driver bug.

Revision 1.02.002 (Released 2011-08-26)

1. Added the NuvotonPlatform_IARv6 directory to support IAR tool v6.10 and later.
2. Supported semihosted input by IAR.

Revision 1.02.001 (Released 2011-07-08)

1. Updated the header to support 32-bit word access for control registers.

Revision 1.01.005 (Released 2011-06-20)

1. Fixed the bug about wrong clock source of timer driver.

Revision 1.01.004 (Released 2011-06-08)

1. Fixed the _PORT_DOUT definition of GPIO header.
2. Fixed the GPIO example in the Driver Reference Guide.

Revision 1.01.003 (Released 2011-05-31)

1. Added new sample code of learning board in KEIL environment: Smpl_I2C_SW, Smpl_I2C_SW_I, Smpl_Idle.

2. Fixed driver bugs.
3. Updated the clock diagram in the Driver Reference Guide.

Revision 1.01.002 (Released 2011-01-05)

1. Fixed bugs and added more samples.
2. Added sample code for the learning board.

Revision 1.01.001 (Released 2010-12-01)

1. Fixed bugs and added more samples.
2. Defaulted the NMI_SEL to 0x31 (assigned NMI to reserved IRQ).
3. Updated the clock diagram in the Driver Reference Guide for the PLL source.
4. Removed the second parameter (E_ADC_INPUT_MODE) from DrvADC_SetADCChannel().
5. Removed multi-function pin configurations from SPI driver (DrvSPI_Open()).
6. Removed multi-function pin configurations from ADC driver [DrvADC_Open( ) & DrvADC_SetADCChannel( )]. User needs to control the multi-function I/O by GPIO driver before they can use the I/O of ADC or SPI.
7. Added sample code for M051-LB_004 (Learning Board).
8. Fixed the gau32ClkSrcTbl setting to avoid wrong system clock calculation.
9. Fixed the bug about EINT0, EINT1 handler cleaning status.
10. Removed unused register definitions in startup_M051Series.s.
11. Fixed the GCR_INT_T definition in M051Series.h.
12. Fixed the wrong definition of DRVUART_STOPBITS_1_5 and DRVUART_STOPBITS_2.

Revision 1.00.001 (Released 2010-08-20)

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.