

M451 Series BSP Directory

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference manual and reversion history.
Library	Driver header and source files.
SampleCode	Driver sample code.
ThirdParty	Library from third party

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

1 Document Information

Revision History.pdf	Show all the revision history about specific BSP.
CMSIS.html	Describe all of the information of CMSIS library, including CMSIS-CORE, CMSIS-DSP, CMSIS-RTOS API and CMSIS-SVD.
NuMicro M451 Series Driver Reference Guide.chm	Describe the definition, input and output of each API.

2 Library Information

CMSIS	CMSIS definitions by ARM® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.
SmartcardLib	Library for CCID smart card reader.
UsbHostLib	Library for USB Host.
NuEdu	Library for NuEdu board.

3 Sample Code Information

CardReader	CCID ^[1] smart card reader sample code.
FreeRTOS	Simple FreeRTOS™ demo code.
Hard_Fault_Sample	Show hard fault information when hard fault happened.
Nu-LB-M451	Sample code for M451 learning board.
NuTiny-SDK-M451	Sample code for M451 Tiny Board.
Template	Software Development Template.
Semihost	Show how to debug with semi-host message print.
RegBased	The sample code able to access control registers directly.
NuEdu	Sample code for M451 NuEdu Evaluation Board.
StdDriver	M451 Series Driver Samples

1. Circuit card interface device (CCID) is USB device that interface with integrated circuit cards.

4 \SampleCode\Nu-LB-M451

ADC_Knob	Demonstrate how to use EADC to measure voltage which is controlled by variable resistor and show conversion result on LED and TFT-LCD.
CAN_NormalMode_TxRx	Demonstrate how to transmit or receive message in Normal mode of CAN. This sample code needs to work with another Nu-LB-M451 board.
I2C_EEPROM	Demonstrate how to access EEPROM through the I2C interface and print test results.
I2C_MPU6050	Demonstrate how to access MPS6050 and display relative information on TFT LCD panel.
I2S_NAU8822	Configure SPI1 as I2S Slave mode and demonstrate how I2S works in Slave mode with NAU8822 and play a sound.
IR_NEC	Demonstrate remote control function based on NEC IR (Infrared receiver) protocol and display relative information on TFT LCD panel when system receives data from NEC IR.
SPI_Flash	Demonstrate how to read from and write to SPI Flash.
SPI_TFT_LCD	Demonstrate how to display an image on TFT LCD panel via the SPI interface.
SPI_TFT_LCD_String	Demonstrate how to print a message on TFT LCD panel via the SPI interface.
StartKit	Demonstrate how to display an image and print a message on TFT LCD panel via the SPI interface.
USB_HID_Mouse	Demonstrate how to implement a USB mouse device. The mouse cursor will move automatically when the mouse device is connected to PC by USB.
USB_MassStorage_SDCard	Demonstrate how to use a SD card as storage to implement a USB Mass-Storage device.

5 \SampleCode\NuTiny-SDK-M451

LEDToggle PC.9 to turn on / off the board LED.

6 \SampleCode\NuEdu

Smpl_Basic01_7_Segment	Demonstrate how to set GPIO pin mode and use pin data output control 7 Segment.
Smpl_Basic01_ACMP	Demonstrate how ACMP[1] works with internal band-gap voltage.
Smpl_Basic01_ADC_Knob	Demonstrate how to use ADC to measure variable resistor and change the LED brightness with PWM according to the ADC conversion results.
Smpl_Basic01_Button	Demonstrate how to set GPIO pin mode and detect pin input signal.
Smpl_Basic01_CRC_CRC8	Implement CRC in CRC-8 mode with CRC write data reverse function and get the CRC checksum result.
Smpl_Basic01_EEPROM	Demonstrate how to access EEPROM through I2C interface and print the test results on PC via NUCOM1 port of the NuEdu-Basic01 board.
Smpl_Basic01_FMC_IAP	Demonstrate a simple IAP function to show three independent programs including main routine, independent interrupt handler and updating or switching to another program with IAP function.
Smpl_Basic01_FMC_ISP	Demonstrate LDROM updated through ISP function by branching to LDROM by software reset, and show debug messages via UART.
Smpl_Basic01_IrDA_NEC	Demonstrate remote control function based on NEC IR protocol and changes LED display via NuEdu-EVB-NUC240 and NuEdu-Basci01 boards when system receives data of NEC IR.
Smpl_Basic01_LED	Demonstrate how to set GPIO pin mode and use pin data output control LED.
Smpl_Basic01_PWMDAC	Demonstrate PWM function to simulate a DAC output.
Smpl_Basic01_RGBLED	Demonstrate how to set GPIO pin mode and use pin data output control RGBLED.

Smpl_Basic01_RTC	Demonstrate RTC application and wake-up function.
Smpl_Basic01_SPI_Flash	Demonstrate how to access SPI Flash through SPI interface and print the test results on both 7-Segments and PC via NUCOM1 port of NuEdu-Basic01 board.
Smpl_Basic01_SPI_Flash_w_PDMA	Demonstrate how to access SPI Flash through a SPI interface with combing PDMA function and print the test results on both 7-Segments and PC via NUCOM1 port of the NuEdu-Basic01 board.
Smpl_Basic01_StartKit	Demonstrate functions of NuEdu-Basic01 via NuEdu-EVB-NUC240.
Smpl_Basic01_Timer	Demonstrate the timer periodic mode application and increase the number display on 7-segments from 0 to 99 via NuEdu-EVB-NUC240 and NuEdu-Basci01 boards when timer interrupt occurs.
Smpl_Basic01_UART	Demonstrate a print message through UART port
Smpl_Basic01_UART_printf	Demonstrate a simple printf function to replace the standard printf library for reducing the code size issue.

7 \SampleCode\RegBased

ACMP	Demonstrate how ACMP ^[1] works with internal band-gap voltage.
ACMP_Wakeup	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
CAN_Set_MaskFilter	Use MaskFilter to receive message in Normal mode. This sample code needs to work with CAN_Test_MaskFilter.
CAN_Test_MaskFilter	Use message object No.1 to send message objects (ID=0x700~0x70F). This sample code needs to work with CAN_Set_MaskFilter.
CLK_ClockDetector	Show the usage of clock fail detector and clock frequency monitor function.
CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Implement CRC in CRC-8 mode and get the CRC checksum result.
DAC_PDMA_PWMTrigger	Show how to use PWM trigger DAC conversion with PDMA.
DAC_PDMA_ScatterGather_PWMTrigger	Show how to use PWM trigger DAC conversion with PDMA scatter gather mode.
DAC_PDMA_TimerTrigger	Show how to use timer trigger DAC conversion with PDMA.
DAC_PWMTrigger	Show how to use PWM trigger DAC conversion.
DAC_TimerTrigger	Show how to use timer trigger DAC conversion.
DAC_WriteDataTrigger	Show how to write a data to DAC_DAT to trigger DAC conversion.
EADC_ADINT_Trigger	Use ADINT interrupt to do the ADC continuous scan conversion.

EADC_PDMA_PWM_Trigger	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
EADC_PWM_Trigger	Demonstrate how to trigger ADC by PWM.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Trigger ADC by writing EADC_SWTRG register.
EADC_Timer_Trigger	Show how to trigger ADC by timer.
EBI_NOR	Configure EBI interface to access MX29LV320T (NOR Flash) on EBI interface.
EBI_SRAM	Configure EBI interface to access BS616LV4017 (SRAM) with PDMA transfer on EBI interface.
FMC_ExecInSRAM	Implement a code and execute in SRAM to program embedded Flash. (Support KEIL® MDK Only)
FMC_IAP	Show how to call LDROM functions from APROM. The code in APROM will look up the table at 0x100E00 to get the address of function of LDROM and call the function.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM. A LDROM code and 4 APROM code are implemented in this sample code.
FMC_RW	Show how to read/program embedded flash by ISP function.
GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.

I2C_EEPROM	Show how to use I ² C interface to access EEPROM.
I2C_GCMode_Master	Show how a Master uses I ² C address 0x0 to write data to Slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_GCMode_Slave	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Master	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Show how to set I ² C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
I2C_SMBus	Show how to control SMBus interface and use SMBus protocol between Host and Slave.
I2C_Wakeup_Master	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode through I ² C interface. This sample code needs to work with I2C_Wakeup_Master.
I2S_Master	Configure SPI1 as I ² S Master mode and demonstrate how I ² S works in Master mode. This sample code needs to work with I2S_Slave sample code.
I2S_Slave	Configure SPI1 as I ² S Slave mode and demonstrate how I ² S works in Slave mode. This sample code needs to work with I2S_Master sample code.
PDMA	Use PDMA channel 2 to transfer data from memory to memory.
PDMA_Scatter_Gather	Use PDMA channel 5 to transfer data from memory to memory by scatter-gather mode.
PDMA_ScatterGather_PingPong Buffer	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

PWM_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.
PWM_PDMA_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2, and use PDMA to transfer captured data.
RTC_AlarmWakeup	Use RTC alarm interrupt event to wake up system.
RTC_SpareRegisterRW	Show how to access RTC spare registers.
RTC_TimeAndTick	Get the current RTC data/time per tick.
SC_ReadATR	Read the smartcard ATR from smartcard 0 interface.
SCUART_TxRx	Show smartcard UART mode by connecting PA.0 and PA.1 pins.
SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterMode	Configure SPI0 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. This sample code needs to work with SPI_SlaveMode.
SPI_PDMA_LoopTest	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
SPI_SlaveMode	Configure SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device. This sample code needs to work with SPI_MasterMode.
SYS_BODWakeup	Show how to wake up system form Power-down mode by brown-out detector interrupt.

SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
TIMER_CaptureCounter	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_EventCounter	Implement timer1 event counter function to count the external input event.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_TimeoutWakeup	Use timer0 periodic time-out interrupt event to wake up system.
UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Slave.
UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Master.
UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Transmit LIN header and response.
UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485_Master	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Transmit and receive data in UART RS485 mode. This sample code needs to work with

	UART_RS485_Master.
UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.
WDT_TimeoutWakeupAndReset	Implement WDT time-out interrupt event to wake up system and generate time-out reset system event while WDT time-out reset delay period expired.
WWDT_CompareINT	Show how to reload the WWDT counter value.

1. Analog Comparator (ACMP).

8 \SampleCode\StdDriver

ACMP	Demonstrate how ACMP works with internal band-gap voltage.
ACMP_Wakeup	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
CAN_BasicMode_Receive	Implement receive message in Basic mode. This sample code needs to work with CAN_BasicMode_Transmit.
CAN_BasicMode_Transmit	Implement transmit message in Basic mode. This sample code needs to work with CAN_BasicMode_Receive.
CAN_NormalMode_Receive	Implement receive message in Normal mode. This sample code needs to work with CAN_NormalMode_Transmit.
CAN_NormalMode_Transmit	Implement transmit message in Normal mode. This sample code needs to work with CAN_NormalMode_Receive.
CAN_Wakeup	Show how to wake up system form Power-down mode by detecting a transition.
CLK_ClockDetector	Show the usage of clock fail detector and clock frequency monitor function.
CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Implement CRC in CRC-8 mode and get the CRC checksum result.
DAC_PDMA_PWMTrigger	Show how to use PWM trigger DAC conversion with PDMA.
DAC_PDMA_ScatterGather_PWMTrigger	Show how to use PWM trigger DAC conversion with PDMA scatter gather mode.
DAC_PDMA_TimerTrigger	Show how to use timer trigger DAC conversion with PDMA.
DAC_PWMTrigger	Show how to use PWM trigger DAC conversion.

DAC_SoftwareTrigger	Show how to use software trigger DAC conversion.
DAC_TimerTrigger	Show how to use timer trigger DAC conversion.
DSP_FFT	Demonstrate how to call ARM CMSIS DSP library to calculate FFT.
EADC_ADINT_Trigger	Use ADINT interrupt to do the ADC continuous scan conversion.
EADC_PDMA_PWM_Trigger	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
EADC_PWM_Trigger	Demonstrate how to trigger ADC by PWM.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Trigger ADC by writing EADC_SWTRG register.
EADC_Timer_Trigger	Show how to trigger ADC by timer.
EBI_NOR	Configure EBI interface to access MX29LV320T (NOR Flash) on EBI interface.
EBI_SRAM	Configure EBI interface to access BS616LV4017 (SRAM) with PDMA transfer on EBI interface.
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash. (Support KEIL® MDK Only.)
FMC_IAP	Show how to reboot to LDROM functions from APROM. This sample code set VECMAP to LDROM and reset to re-boot to LDROM.
FMC_RW	Show how to read/program embedded flash by ISP function.
GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data

	input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.
I2C_EEPROM	Show how to use I ² C interface to access EEPROM.
I2C_GCMode_Master	Show how a Master uses I ² C address 0x0 to write data to Slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_GCMode_Slave	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Master	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Show how to set I ² C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
I2C_SMBus	Show how to control SMBus interface and use SMBus protocol between Host and Slave.
I2C_Wakeup_Master	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode through I ² C interface. This sample code needs to work with I2C_Wakeup_Master.
I2S_Master	Configure SPI1 as I ² S Master mode and demonstrate how I ² S works in Master mode. This sample code needs to work with I2S_Slave.
I2S_Slave	Configure SPI1 as I ² S Slave mode and demonstrate how I ² S works in Slave mode. This sample code needs to work with I2S_Master.
PDMA	Use PDMA channel 2 to transfer data from memory to memory.
PDMA_Scatter_Gather	Use PDMA channel 5 to transfer data from memory to memory by scatter-gather mode.

PDMA_ScatterGather_PingPongBuffer	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).
PWM_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.
PWM_PDMA_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2, and use PDMA to transfer captured data.
RTC_AlarmWakeup	Use RTC alarm interrupt event to wake up system.
RTC_SpareRegisterRW	Show how to access RTC spare registers.
RTC_TimeAndTick	Get the current RTC data/time per tick.
SC_ReadATR	Read the smartcard ATR from smartcard 0 interface.
SCUART_TxRx	Show smartcard UART mode by connecting PA.0 and PA.1 pins.
SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 pin together. It will compare the received data with transmitted data.
SPI_MasterMode	Configure SPI0 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. Needs to work with SPI_SlaveMode.
SPI_PDMA_LoopTest	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
SPI_SlaveMode	Configure SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device. This sample code needs to work with SPI_MasterMode.
SYS_BODWakeup	Show how to wake up system form Power-down mode by

	brown-out detector interrupt.
SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
TIMER_EventCounter	Implement timer1 event counter function to count the external input event.
TIMER_CaptureCounter	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_Delay	Show how to use timer0 to create various delay time.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_TimeoutWakeup	Use timer0 periodic time-out interrupt event to wake up system.
UART_AutoBaudRate_Master	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Slave.
UART_AutoBaudRate_Slave	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Master.
UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_LIN	Transmit LIN header and response.
UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485_Master	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.

UART_RS485_Slave	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system form Power-down mode by UART interrupt.
USBDAudioHIDNAU8822	Implement a USB audio class device with HID key. NAU8822 is used in this sample code to play the audio data from Host. It also supports to record data from NAU8822 to Host.
USBDAudioNAU8822	Demonstrate how to implement a USB audio class device. NAU8822 is used in this sample code to play the audio data from Host. It also supports to record data from NAU8822 to Host.
USBDAIDKeyboard	Show how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
USBDAIDMouse	Show how to implement a USB mouse device. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
USBDAIDMouseKeyboard	Demonstrate how to implement a USB mouse function and a USB keyboard on the same USB device. The mouse cursor will move automatically when this mouse device connecting to PC. This sample code uses a GPIO to simulate key input.
USBDAIDTransfer	Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with USB device.
USBDAIDTransferandKeyboard	Demonstrate how to implement a composite device (HID Transfer and keyboard). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with USB device.
USBDAIDTransferandMSC	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB

	device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_MassStorage_CDROM	Demonstrate how to simulate a USB CD-ROM device.
USBD_MassStorage_DataFlash	Use embedded data flash as storage to implement a USB Mass-Storage device.
USBD_Micro_Printer	Show how to implement a USB micro printer device.
USBD_Printer_and_HID_Transfer	Demonstrate how to implement a composite device (USB micro printer device and HID Transfer). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_and_HID_Keyboard	Implement a USB composite device with virtual COM port and keyboard functions.
USBD_VCOM_and_HID_Transfer	Demonstrate how to implement a composite device (VCOM and HID Transfer). It supports one virtual COM port and transfers data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_and_MassStorage	Implement a USB composite device. It supports one virtual COM port and one USB Mass-Storage device.
USBD_VCOM_DualPort	Demonstrate how to implement a USB dual virtual COM port device.
USBD_VCOM_SinglePort	Implement a USB virtual COM port device. It supports one virtual COM port.

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.*