NUC472/NUC442 BSP Directory

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference manual and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.
ThirdParty	Library from third party, including FatFs, LibMAD, lwIP, uIP, LibMAD, and FreeRTOS™.

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

www.nuvoton.com

1 Document Information

CMSIS.html	Document of CMSIS version 4.5.0
NuMicro NUC472_NUC442 Series CMSIS BSP Revision History.pdf	This document shows the revision history of NUC472/NUC442 BSP.
NuMicro NUC472 NUC442 Driver Reference Guide.html	This document describes the usage of drivers in NUC472/NUC442 BSP.

2 Library Information

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by ARM® Corp.
Device	CMSIS compliant device header file.
SmartcardLib	Smartcard library binary and header file.
StdDriver	All peripheral driver header and source files.
UsbHostLib	USB host library source code.

3 Sample Code Information

CortexM4	Cortex®-M4 sample code.
FreeRTOS	Simple FreeRTOS™ demo code.
FreeRTOS_lwIP_httpd	A simple HTTP server demonstrates LwIP under FreeRTOS™. This HTTP server's IP address could be configured statically to 192.168.0.2, or assign by DHCP server.
FreeRTOS_lwIP_TCP_ EchoServer	A TCP echo server which is implemented with LwIP under FreeRTOS™. This echo server listens to port 80, and its IP address could be configured statically to 192.168.1.2 or assigned by DHCP server. This server replies "Hello World!!" if the received string is "nuvoton", otherwise replies "Wrong Password!!" to its client.
FreeRTOS_lwIP_UDP_ EchoServer	A UDP echo server which is implemented with LwIP under FreeRTOS™. This echo server listens to port 80, and its IP address could be configured statically to 192.168.1.2 or assigned by DHCP server. After receiving any string from its peer, this server echoes that string back.
Hard_Fault_Sample	Show hard fault information when hard fault happened.
ISP	
ISP_Updater	Sample ISP updater that reads firmware from pen driver and updates to APROM.
NUC472-NuTiny	Sample code for NUC472 Tiny Board.
RTX_Blinky	Simple RTX demo code that blinks the on-board LED.
Semihost	Show how to debug with semi-host message print.
StdDriver	Sample code to demonstrate the usage of NUC472/NUC442 MCU peripheral driver APIs.
Template	A project template for NUC472/NUC442 MCU.

4 Third Party Information

FATFS	A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html.
FreeRTOSV8.2.1	A real time operating system available for free download. Its official website is: http://www.freertos.org/.
LibMAD	A MPEG audio decoder library which currently supports MPEG-1 and the MPEG-2 extension to lower sampling frequencies, as well as the de facto MPEG 2.5 format. All three audio layers — Layer I, Layer II, and Layer III (i.e. MP3) are fully implemented. This library is distributed under GPL license. Please contact Underbit Technologies (http://www.underbit.com/) for the commercial license.
lwip-1.4.1	A widely used open source TCP/IP stack designed for embedded systems. Its official website is: http://savannah.nongnu.org/projects/lwip/.
uip-0.9	uIP is a very small implementation of the TCP/IP stack that is written by Adam Dunkels <adam@sics.se>. More information can be obtained from the uIP homepage at http://www.sics.se/~adam/uip/.

5 \SampleCode\CortexM4

BitBand	Demonstrate the usage of Cortex®-M4 Bit-band.
DSP_FFT	Demonstrate how to call ARM CMSIS DSP library to calculate FFT.
MPU	Demonstrate the usage of Cortex®-M4 MPU.

6 \SampleCode\ISP

ISP_CAN	Sample ISP firmware communicated with ISP tool through an CAN interface.
ISP_DFU	Sample ISP firmware communicated with ISP tool through USB DFU(Device Firmware Upgrade) class.
ISP_HID	Sample ISP firmware communicated with ISP tool through a USBD HID interface.
ISP_I2C	Sample ISP firmware communicated with ISP tool through an I^2C interface.
ISP_RS485	Sample ISP firmware communicated with ISP tool through a RS485 interface.
ISP_SPI	Sample ISP firmware communicated with ISP tool through a SPI interface.
ISP_UART	Sample ISP firmware communicated with ISP tool through a UART interface.

7 \SampleCode\NUC472-NuTiny

LED	Toggle PB.10 to turn on / off the board LED.

8 \SampleCode\StdDriver

ACMP	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and VBG voltage and shows the result on UART console.
ADC_Compare	Demonstrate ADC conversion and comparison function by monitoring the conversion result of channel 0.
ADC_ContinuousScan	Convert ADC channel 0, 1, 2 in continuous scan mode and prints conversion result.
ADC_PDMA	Use PDMA channel 1 to move ADC channel 0, 1, 2 converted data to SRAM.
ADC_SingleCycleScan	Convert ADC channel 0, 1, 2 in single cycle scan mode and prints conversion result.
ADC_SingleMode	Convert ADC channel 0 in single mode and prints conversion result.
CAN_BasicMode_Rx	Demonstrate CAN bus receive a message with basic mode.
CAN_BasicMode_Tx	Demonstrate CAN bus transmit a message with basic mode.
CAN_BasicMode_Tx_Rx	Demonstrate CAN bus transmit and receive a message with basic mode by connecting CAN0 and CAN1 to the same CAN bus.
CAN_NormalMode_Rx	Demonstrate CAN bus receive a message with normal mode.
CAN_NormalMode_Tx	Demonstrate CAN bus transmit a message with normal mode.
CAN_NormalMode_Tx_Rx	Demonstrate CAN bus transmit and receive a message with normal mode by connecting CAN 0 and CAN1 to the same CAN bus.
CAP_MotionDetection	Implement motion detection with image capture interface.
CAP_Packet_DownScale	Use packet format (all the luma and chroma data interleaved) to store captured image from NT99141

	sensor to SRAM.
CAP_Planar_DownScale	Use planar format (all the luma information for a frame, followed by all the information for one chroma channel, and then the information for the other chroma channel) to store captured image from NT99141 sensor to SRAM.
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.
CRYPTO_AES	Show Crypto IP AES-128 ECB mode encrypt/decrypt function.
CRYPTO_PRNG	Generate random numbers using Crypto IP PRNG.
CRYPTO_SHA	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.
CRYPTO_TDES	Show Crypto IP Triple DES CBC mode encrypt/decrypt function.
EADC_ADINT_Trigger	Demonstrate how to use ADINT interrupt to trigger EADC.
EADC_Compare	Demonstrate EADC conversion and comparison function by monitoring the conversion result of sample module 0 channel 2.
EADC_PWM_Trigger	Show how to trigger EADC0 SAMPLE module 0 by PWM channel 0
EADC_SimultaneousMode	Demonstrate EADC0 sample module 2 and EADC1 sample module 2 in simultaneous sampling mode.
EADC_STADC_Trigger	Demonstrate how to trigger EADC by STADC external signal.
EADC_SWTRG_Trigger	Demonstrate how to trigger EADC by writing EADC_SWTRG register.
EADC_Timer_Trigger	Demonstrate how to trigger EADC by timer.
EBI_SRAM	Configure EBI interface to access SRAM connects on EBI

	interface.
ECAP	Demonstrate the Enhanced capture function using PC.5
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash. (Support KEIL® MDK Only.)
EMAC_TimeStamp	Demonstrate the usage of Ethernet time stamp function. It sets current time to 1000 second and prints out current time every second. It also sets an alarm at 1010 second. And rewind current time by 5 seconds after the alarm.
EMAC_TxRx	This Ethernet sample tends to get a DHCP lease from DHCP server, and use 192.168.10.10 as IP address if failed to get a lease. After IP address configured, this sample can reply to PING packets.
EMAC_uIP_httpd	Implement a HTTP server using uIP.
EMAC_uIP_telnetd	Implement a Telnet server using uIP.
EPWM_Brake	Demonstrate the brake function of EPWM0.
EPWM_DeadZone	Demonstrate the dead-zone feature with EPWM0 channel 0 and channel 1.
FMC_MULTI_WORD_PROG	Show FMC ISP multi-word program function. The loader.bin will load fmc_multi_word_prog.bin to SRAM and execute it.
FMC_RW	Show FMC read flash IDs, erase, read, and write functions.
FMC_VECTOR_REMAP	Show how to branch programs between LDROM, APROM start page, and APROM other page.
GPIO	Use GPIO driver to control the GPIO pin direction, control their high/low state, and how to use GPIO interrupts.
I2C_EEPROM	Read/write EEPROM via I²C interface.
I2C_GSENSOR	Read G-sensor (DMARD08) data via I²C interface.
I2C_Master	An I²C master mode demo code.

I2C_Slave	An I²C slave mode demo code.
I2S_MP3PLAYER	A MP3 player sample which plays MP3 files stored on SD memory card.
I2S_NAU8822	An I²S demo using NAU8822 audio codec, and used to play back the input from line-in or MIC interface.
I2S_NAU8822_PDMA	An I²S with PDMA demo using NAU8822 audio codec, and used to play back the input from line-in or MIC interface.
I2S_WAVPLAYER	A WAV file player which plays back WAV file stored in USB pen drive.
PDMA	Use PDMA channel 2 to demonstrate memory to memory transfer.
PDMA_Scatter_Gather	Use PDMA channel 5 to demonstrate memory to memory transfer by scatter-gather mode.
PS2	Simulate the behavior of a PS/2 mouse by moving the cursor on the screen.
PWM_Capture	Demonstrate PWM Capture function by using PWM0 channel 2 to capture the output of PWM0 channel 0. Please connect PA.5 and PC.10 to execute this code.
PWM_DeadZone	Demonstrate the dead-zone feature with PWM0.
RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Spare_Access	Show how to access RTC spare registers.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.
SC_ReadATR	Read the smartcard ATR from smartcard 5 interface.
SCUART_TxRx	Demonstrate smartcard UART mode by connecting PA.7 and PA.10 pins.
SD_FATFS	Access a SD card formatted in FAT file system.

SPI_DualMode_Flash	Access SPI Flash using SPI dual mode.
SPI_Flash	Access SPI Flash through SPI interface.
SPI_LoopBack	A SPI read/write demo by connecting SPI0 and SPI1 interface.
SPI_MasterMode	SPI master mode demo code.
SPI_MasterSlave_PDMA	Demonstrate the usage of PDMA transfer. One SPI interface is use as a host, and the other is slave. Totally 4 PDAM channels are used in this sample.
SPI_QuadMode_Flash	Access SPI Flash using SPI quad mode.
SPI_SlaveMode	SPI slave mode demo code.
SPI_TFT_LCD	Display an image on TFT LCD panel via SPI interface.
SPI_TxRxLoopback_PDMA	Demonstrate the usage of PDMA transfer. One SPI interface is enabled in loopback mode. Two PDMA channels are used in this sample, one for transmit, the other for receive.
SYS_Control	Demonstrate the usage of SYS driver by changing different PLL setting for the system clock source. This sample also enables the CLKO (PC.5) output with frequency set to system clock / 4.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.
Timer_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
TIMER_EventCounter	Use pin PB.4 to demonstrates timer event counter function.
Timer_FreeCountingMode	Use the timer pin PC.8 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
Timer_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.

Timer_ToggleOut	Demonstrate the timer 0 toggle out function on pin PB.4.
TIMER_Wakeup	Use Timer to wake up system from Power-down mode periodically.
UART_AutoFlow	Transmit and receive data using auto flow control.
UART_IrDA	Transmit and receive UART data in UART IrDA mode.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.
USBD_Audio_Microphone	An UAC1.0 sample used to record the sound to PC through the USB interface.
USBD_Audio_Speaker	An UAC1.0 sample used to play the sound sent from PC through the USB interface.
USBD_Bulk	Sample USB device bulk transfer code.
USBD_HID_MOUSE	Simulate a USB mouse and draws circle on the screen.
USBD_HID_Mouse_Vender	Simulate a USB mouse that supports vendor command.
USBD_HID_MouseKeyboard	Simulate a USB mouse and a USB keyboard.
USBD_HID_Transfer	Demonstrate how to 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_Mass_Storage_ DataFlash	Use embedded data flash as storage to implement a USB Mass-Storage device.
USBD_Mass_Storage_ ShortPacket	Implement a mass storage class sample to demonstrate how to receive a USB short packet.
USBD_Mass_Storage_ SRAM	Use internal SRAM as back end storage media to simulate a 30 KB USB pen drive.

USBD_Mass_Storage SactterGather	Demonstrate the usage of USBD DMA scatter gather function.
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual com port device.
USBD_VENDOR_LBK	A USB device vendor class sample program. This sample code needs to test with USBH_VENDOR_LBK.
USBH_AUDIO_CLASS	A USB Host sample code to support USB Audio Class.
USBH_HID	Use USB Host core driver and HID driver. It shows how to submit HID class request and how to read data from interrupt pipe.
USBH_HID_KEYBOARD	Show how to use USB Host driver to handle HID keyboard devices
USBH_HID_MULTI	Show how to implement a USB Host and recognize multi-HID devices when devices plug-in.
USBH_UAC_HID	A USB Host sample code to support USB Audio Class with HID composite device.
USBH_UMAS	Use USB Host core driver, USB mass storage driver, and FATFS file system to show a disk access shell interface.
USBH_VENDOR_LBK	A USB host vendor class sample program. This sample code needs to test with USBD_VENDOR_LBK.
USBOTG_Dual_Role_UMAS	An OTG sample code which will become a USB host when connected with a Micro-A cable, and can access the pen drive when plugged in. It will become a removable disk when connected with a Micro-B cable, and then plug into PC.
WDT_Polling	Use polling mode to check WDT time-out state and reset WDT after time out occurs.
WDT_Wakeup	Use WDT to wake system up from power-down mode periodically.
WWDT_Reload	Demonstrate the WWDT counter reload function.

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.