I94100 Series Board Supporting Package Directory Introduction

Rev.3.05.002

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

Directory Information

Document	Driver reference manual and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.

Document Information

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

Library Information

CMSIS	CMSIS definitions by ARM® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.

Sample Code Information

\SampleCode\Template	Software Development Template.
\SampleCode\Semihost	The sample code to show how to debug with semihost message print.
\SampleCode\StdDriver	I94100 Series Driver Samples

\SampleCode\StdDriver

CLK_ClockDetector	Demonstrate the usage of HXT clock fail detector.
CRC_CCITT	Demonstrate CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Demonstrate CRC in CRC-8 mode and get the CRC checksum result.
CRC_CRC32	Demonstrate CRC in CRC-32 mode with PDMA transfer.
DMIC_DPWM	Demonstrate how to use ‘DMIC’ to record and ‘DPWM’ to play audio
DMIC_DPMW_PDMA	Demonstrate how to use ‘DMIC’ to record and ‘DPWM’ to play audio, and DMIC and DPWM both use PDAM to transfer data.
DPWM	Demonstrate how to use ‘DPWM’ to play audio file.
EADC_ADINT_Trigger	Demonstrate use ADINT interrupt to do the EADC 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 EADC by PWM.
EADC_ResultMonitor	Demonstrate monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Demonstrate trigger EADC by writing EADC_SWTRG register.
EADC_Timer_Trigger	Demonstrate how to trigger EADC by timer.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM, LDROM.
FMC_IAP	Demonstrate FMC APROM with IAP programs LDROM
FMC_MultiWordProgram	Demonstrate FMC multi-word program function.
FMC_RW	Demonstrate FMC read Flash IDs, erase, read, and write.
GPIO_EINTAndDebounce	Demonstrate the usage of GPIO external interrupt function and

	de-bounce function.
GPIO_INT	Demonstrate the usage of GPIO interrupt function.
GPIO_OutputInput	Demonstrate how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Demonstrate how to wake up system from Power-down mode by GPIO interrupt.
I2C_GCMode_Master	Demonstrate Master uses I2C address 0x0 to write data to Slave.This sample code needs to work with I2C_GCMode_Slave
I2C_GCMode_Slave	Demonstrate Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Master	Demonstrate Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Demonstrate I2C slave mode and need to be tested with a master device.
I2C_SMBus	Demonstrate how to control SMBus interface and use SMBus protocol between Host and Slave.
I2C_Wakeup_Master	Demonstrate how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Demonstrate how to wake up MCU from Power-down mode through I2C interface. This sample code needs to work with I2C_Wakeup_Master.
I2S_Master	Demonstrate how to Implement I2S in Master mode.
I2S_Slave	Demonstrate how to Implement I2S in Slave mode.
I2S_DPWM_85L40	Demonstrate how to implement a record and playback device by using NAU85L40 as microphones, DPWM to drive speakers and I2S to transfer data between I94100 and NAU85L40.
I2S_DPWM_85L40_PDMA	Demonstrate how to implement a record and playback device by using NAU85L40 as microphones, DPWM to drive speakers and I2S to transfer data between I94100 and NAU85L40. PDMA is implemented for data transfer from RAM to DPWM.

PDMA_Control	Demonstrate use PDMA channel 2 to transfer data from memory to memory.
PDMA_Scatter_Gather	Demonstrate PDMA transfer data from memory to memory by scatter-gather mode.
PDMA_ScatterGather_PingPong Buffer	Demonstrate PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).
PWM_Capture	Demonstrate capture the PWM0 Channel 0 waveform by PWM0 Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Demonstrate change duty cycle and period of output waveform by PWM Double Buffer function.
RTC_AlarmWakeup	Demonstrate use RTC alarm interrupt event to wake up system.
RTC_TimeAndTick	Demonstrate get the current RTC data/time per tick.
SPI_I2S_Master	Demonstrate SPI data transfer. (1) Implement how SPI_I2S works in Master mode. (2) This sample code will transmit a TX value 50000 times, and then change to the next TX value.
SPI_I2S_Slave	Demonstrate SPI data transfer. (1) Implement SPI transfer (2) Configure SPI1 as SPI_I2S Slave mode and demonstrate how SPI_I2S works in Slave mode.
SPI_Loopback	Demonstrate 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_MasterSlave	Demonstrate SPI data transfer. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. SPI0 (master) transfer via interrupt and SPI1 (slave) transfer via main loop.

SPI_PDMA_MasterSlave	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_QuadFlash	Demonstrate how to use SPI quad-mode to read/write data to external SPI-FLash. (1) Implement SPI quad-mode data transfer. (2) SPI0 will be configured as Master mode. SPI0(master) will send command via one-bit mode, and read write data through quad-mode. (3)Using W25Q256FV serial flash memory in this demonstration.
SYS_BODWakeup	Demonstrate how to wake up system form Power-down mode by brown-out detector interrupt.
SYS_DPDMode_Wakeup	Demonstrate how to wake up system form DPD Power-down mode by Wake-up pin(PA.15) or Wake-up Timer.
SYS_PLLClockOutput	Demonstrate change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_SPDMode_Wakeup	Demonstrate how to wake up system form SPD Power-down mode by GPIO pin (PA.0) or Wake-up Timer.
TIMER_CaptureCounter	Demonstrate how to use the timer2 capture function to capture timer2 counter value.
TIMER_Delay	Demonstrate how to use timer0 to create various delay time.
TIMER_EventCounter	Demonstrate timer1 counts the external input event.
TIMER_PeriodicINT	Demonstrate timer counting in periodic mode.
TIMER_TimeoutWakeup	Demonstrate timer0 periodic time-out interrupt event to wake up system.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_TxRxFunction	Demonstrate transmit and receive data from PC terminal through RS232 interface.
USBD_HID_Mouse	This sample code shows how to implement a HID device as

	mouse.
USBD_HID_Transfer	Demonstrate how to implement a humane interface device.
USBD_MassStorage_DataFlash	Demonstrate how to implement a USB Mass-Storage device.
USBD_Printer_And_HID_Transfer	Demonstrate how to implement a composite device.(USB micro printer device and HID Transfer).
USBD_UAC_85L40_PDMA	This sample code shows how to implement a USB recording device using codec NAU85L40.
USBD_UAC_85L40_PDMA_4CH_NoVolCtrl	This sample code shows how to implement a USB 4-channel recording device using codec NAU85L40. User can chose how many channels to be enabled by enabling one of the record device and disabling threee other device. Four devices could enable 1, 2, 3 and 4 channels.
USBD_UAC_85L40_PDMA_4CH_VolCtrl	This sample code shows how to implement a USB 4-channel recording device using codec NAU85L40. User can chose how many channels to be enabled by enabling one of the record device and disabling threee other device. Four devices could enable 1, 2, 3 and 4 channels.
USBD_UAC_DMIC_DPWM_PDMA	This sample code shows how to implement a USB recording and playback device using DMIC and DPWM.
USBD_UAC_DMIC_DPWM_PDMA_4CH	This sample code shows how to implement a USB recording and playback device using DMIC and DPWM. (1)Using EVB-I94124ADI-NAU85L40B (DMIC version) to demo. User can chose how many channels to be enabled by enabling one of the record device and disabling threee other device. Four devices could enable 1, 2, 3 and 4 channels.
USBD_UAC_DMIC_PDMA_4CH	This sample code shows how to implement a USB recording and playback device using DMIC and DPWM. (1) Using EVB-I94124ADI-NAU85L40B (DMIC version) to demo. User can chose how many channels to be enabled by enabling one of the record device and disabling threee other device. Four devices could enable 1, 2, 3 and 4 channels.

USBD_UAC_DPWM	This sample code shows how to implement a USB playback device using DPWM.
USBD_VCOM_And_HIDTransfer	Demonstrate how to implement a composite device.(VCOM and HID Transfer)
USBD_UAC_I2S_Output	This sample code shows how to implement a USB playback device using I2S output. The I2S output should be received by a slave device. BSP provides "I2S_Slave_DPWM" as a slave device sample code to playback audio.
USBD_VCOM_And_HIDTransfer	Demonstrate how to implement a composite device.(VCOM and HID Transfer) (1) HID Transfer – 1. Transfer data between USB device and PC through USB HID interface. 2. A windows tool is also included in this sample code to connect with a USB device. (2) VCOM – 1. Implement a USB virtual com port device. 2. A window driver is also include in this sample code to setup before using.
VAD_Wakeup	Demonstrate enter to power down mode and wake-up via VAD(Voice Active Detection).
WDT_TimeoutWakeupAndReset	Demonstrate 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	Demonstrate how to reload the WWDT counter value.

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.