NUC122 CMSIS 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.

TABLE OF CONTENTS

1 Document Information

Revision History	Show all the revision history about specific BSP.
NuMicro NUC122 Driver Reference Guide.chm	The usage of drivers in NUC122 Series BSP.

2 Library Information

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

3 Sample Code Information

Hard_Fault_Sample	Show hard fault information when hard fault happened.
ISP	Sample codes for In-System-Programming.
Template	Software Development Template.
Semihost	A sample code to show how to debug with semihost message print.
RegBased	The sample codes which access control registers directly.
StdDriver	NUC122 Series Driver Samples

4 SampleCode\ISP

ISP_DFU	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
ISP_HID	In-System-Programming Sample code through USB HID interface.
ISP_I2C	In-System-Programming Sample code through I2C interface.
ISP_RS485	In-System-Programming Sample code through RS485 interface.
ISP_SPI	In-System-Programming Sample code through SPI interface.
ISP_UART	In-System-Programming Sample code through UART interface.

5 SampleCode\RegBased

System Manager (SYS)

SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode

Flash Memory Controller (FMC)

FMC_RW	Demonstrate how to read/program embedded flash by ISP function.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use 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 form Power-down mode by GPIO interrupt.

Timer Controller (TIMER)

TIMER_Counter	Demonstrate how to use timer1 counter input function to count the input event.
TIMER_PeriodicINT	Demonstrate how to perform timer counting in periodic mode.

Watchdog Timer (WDT)

WDT_PowerDown	Demonstrate how to use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Select one WDT time-out interval period time to generate time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to cause WDT time-out reset system event while WDT time-out reset delay period expired.

PWM Generator and Capture Timer (PWM)

PWM_Capture	Demonstrate how to use PWMB Channel 2 captures PWMB Channel 1 Waveform.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Use PWM Double Buffer function to change duty cycle and period of output waveform.

UART Interface Controller (UART)

UART_Autoflow_Master	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Master.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Slave.

UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Demonstrate how UART 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.

Serial Peripheral Interface (SPI)

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	Demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with SPI_SlaveMode sample code.
SPI_SlaveMode	Demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with SPI_MasterMode sample code.

I^{2}C Serial Interface Controller (I^{2}C)

I2C_EEPROM	Demonstrate how to access EEPROM by I^{2}C interface.
I2C_GCMode_Master	Demonstrate how a Master uses I^{2}C address 0x0 to write data to I^{2}C Slave. Needs to work with I2C_GCMode_SLAVE sample code.
I2C_GCMode_Slave	Demonstrate how to receive Master data in GC (General Call) mode. Needs to work with I2C_GCMode_MASTER sample code.
I2C_Master	Demonstrate how a Master access Slave. Needs to work with I2C_SLAVE sample code.
I2C_Slave	Demonstrate how to set I^{2}C in slave mode to receive the

	data of a Master. Needs to work with I2C_MASTER sample code.

PS/2 Controller (PS/2)

PS2	Demonstrate how to control PS/2 mouse movement on the screen.

6 SampleCode\StdDriver

System Manager (SYS)

SYS	Demonstrate how to change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode

Flash Memory Controller (FMC)

FMC_RW	Demonstrate how to read/program embedded flash by ISP function.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Demonstrate how to use GPIO external interrupt function and de-bounce function.
GPIO_INT	Demonstrate how to use 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 form Power-down mode by GPIO interrupt.

Timer Controller (TIMER)

TIMER_Counter	Demonstrate how to use timer1 counter input function to count the input event.
TIMER_PeriodicINT	Demonstrate how to perform timer counting in periodic mode.

Watchdog Timer (WDT)

WDT_PowerDown	Demonstrate how to use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Select one WDT time-out interval period time to generate time-out interrupt event.
WDT_TimeoutReset	Demonstrate how to cause WDT time-out reset system event while WDT time-out reset delay period expired.

UART Interface Controller (UART)

UART_Autoflow_Master	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Slave.
UART_Autoflow_Slave	Demonstrate how to transmit and receive data with auto flow control. The sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Demonstrate how to transmit and receive data in UART IrDA mode. The sample code needs to work with UART_IrDA_Master.
UART_RS485_Master	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Demonstrate how to transmit and receive data in UART RS485 mode. The sample code needs to work with UART_RS485_Master.
UART_TxRx_Function	Demonstrate how UART 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.

Serial Peripheral Interface (SPI)

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 with FIFO mode. This sample code needs to work with SPI_SlaveMode sample code.
SPI_SlaveMode	Configure SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with SPI_MasterMode sample code.

I^{2}C Serial Interface Controller (I^{2}C)

I2C_EEPROM	Demonstrate how to access EEPROM by I^{2}C interface.
I2C_GCMode_Master	Demonstrate how a Master uses I^{2}C address 0x0 to write data to I^{2}C Slave. Needs to work with I2C_GCMode_SLAVE sample code.
I2C_GCMode_Slave	Demonstrate how to receive Master data in GC (General Call) mode. Needs to work with I2C_GCMode_MASTER sample code.
I2C_Master	Demonstrate how a Master access Slave. Needs to work with I2C_SLAVE sample code.
I2C_Slave	Demonstrate how to set I^{2}C in slave mode to receive the data of a Master. Needs to work with I2C_MASTER sample code.

PS/2 Controller (PS/2)

PS2	Demonstrate how to control PS/2 mouse movement on the screen.

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.