

## M5531 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Please extract the “M5531\_Series\_BSP\_CMSIS\_V3.01.003.zip” file firstly and then put the “M5531\_Series\_BSP\_CMSIS\_V3.01.003” folder into the working folder.

To experience the powerful features of M5531 series in few minutes, please select the sample code to download and execute on the NuMaker-M5531 board. Open the project files to build them with Keil® MDK, IAR, NuEclipse or VS Code, and then download and trace them on the NuMaker board to see how it works.

Please note that M5531 Series CMSIS BSP enables CPU Level-1 I/D-Cache by default but does not guarantee Cache coherence.

### Directory Information

<b>Document</b>	Driver reference guide and revision history.
<b>Library</b>	Driver header and source files.
<b>SampleCode</b>	Driver sample code.
<b>ThirdParty</b>	Libraries from third parties.
<b>Tool</b>	Utility programs.

*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 and microprocessor 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](http://www.nuvoton.com)

## TABLE OF CONTENTS

1	DOCUMENT .....	5
2	LIBRARY .....	6
3	SAMPLECODE .....	7
4	THIRDPARTY .....	9
5	TOOL .....	11
6	SAMPLECODE\CORTEXM55 .....	12
7	SAMPLECODE\CRYPTO .....	13
8	SAMPLECODE\FREERTOS .....	14
9	SAMPLECODE\ISP .....	15
10	SAMPLECODE\NUMAKER .....	16
11	SAMPLECODE\POWERDELIVERY .....	18
12	SAMPLECODE\POWERMANAGEMENT .....	19
13	SAMPLECODE\SECUREAPPLICATION .....	20
14	SAMPLECODE\STDDRIVER .....	21
	Analog Comparator Controller (ACMP) .....	21
	Awake Filter (AWF) .....	21
	Basic PWM Generator and Capture Timer (BPWM) .....	21
	Controller Area Network with Flexible Data-Rate (CAN FD) .....	22
	Camera Capture Interface Controller (CCAP) .....	22
	Clock Controller (CLK) .....	23
	CRC Controller (CRC) .....	23
	Cryptographic Accelerator (CRYPTO) .....	23

Digital to Analog Converter (DAC).....	24
Digital Microphone Inputs (DMIC).....	24
Debug Protection Mechanism (DPM) .....	25
Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	25
External Bus Interface (EBI) .....	25
Enhanced Input Capture Timer (ECAP) .....	26
Ethernet MAC Controller (EMAC) .....	26
Enhanced PWM Generator and Capture Timer (EPWM) .....	26
Enhanced Quadrature Encoder Interface (EQEI) .....	27
Flash Memory Controller (FMC).....	27
Graphic DMA Controller (GDMA) .....	28
General Purpose I/O (GPIO).....	28
High Speed USB 2.0 On-The-Go (HSOTG) .....	28
High Speed USB 2.0 Device Controller (HSUSBD) .....	29
USB 2.0/1.1 Host Controller (HSUSBH/USBH).....	30
I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	31
I <sup>2</sup> S Controller (I <sup>2</sup> S).....	32
I3C Serial Interface Controller (I3C).....	33
Key Derivation Function (KDF) .....	34
Keypad Interface (KPI).....	34
Key Store (KS).....	34
Low Power Analog-to-Digital Converter (LPADC) .....	34
Low Power General Purpose I/O (LPGPIO) .....	35
Low Power I2C Serial Interface Controller (LPI2C).....	35
Low Power PDMA Controller (LPPDMA) .....	36
Low Power Serial Peripheral Interface (LPSPI) .....	36
Low Power Timer Controller (LPTMR) .....	37
Low Power UART Interface Controller (LPUART) .....	38
USB 1.1 On-The-Go (OTG) .....	38
PDMA Controller (PDMA) .....	38
Programmable Serial I/O (PSIO).....	39
Quad Serial Peripheral Interface (QSPI) .....	39
Random Number Generator (RNG).....	40
Real Timer Clock (RTC) .....	40

Smart Card Host Interface (SC).....	40
Secure Digital Host Controller (SDH).....	41
Serial Peripheral Interface (SPI).....	41
SPI/Hyper Bus Synchronous Serial Interface Controller (SPIM).....	42
System Manager (SYS).....	42
Timer Controller (TIMER).....	43
Tick Timer Controller (TTMR) .....	44
UART Interface Controller (UART).....	44
USB 1.1 Device Controller (USBD) .....	45
Universal Serial Control Interface Controller - I <sup>2</sup> C Mode (USCI-I2C) .....	47
Universal Serial Control Interface Controller - SPI Mode (USCI-SPI).....	48
Universal Serial Control Interface Controller - UART Mode (USCI-UART) .....	48
Watchdog Timer (WDT) .....	49
Window Watchdog Timer (WWDT) .....	49
<b>15 SAMPLECODE\TRUSTZONE .....</b>	<b>50</b>
<b>16 SAMPLECODE\XOM .....</b>	<b>51</b>

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 6.1.0.
<b>NuMicro M5531 Series CMSIS BSP Driver Reference Guide.chm</b>	This document describes the usage of drivers in M5531 Series CMSIS BSP.
<b>NuMicro M5531 Series CMSIS BSP Revision History.pdf</b>	This document shows the revision history of M5531 Series CMSIS BSP.
<b>VSCode Quick Start Guide.pdf</b>	This document guides to install, configure and use VS Code.

## 2 Library

<b>CMSIS</b>	<p>Cortex® Microcontroller Software Interface Standard (CMSIS) V6.1.0 definitions by Arm® Corp.</p> <p>M5531 CMSIS-Drivers do not guarantee thread safety and Cache coherence. The source and RTE_Device header files are in the <a href="#">Driver\Source</a> sub-folder. Please add source files and copy RTE_Device header files into your project. Projects can define PRJ_RTE_DEVICE_HEADER macro to include the private RTE_Device.h.</p>
<b>Commu</b>	Helper functions of communication protocols, e.g., xmodem.
<b>CryptoAccelerator</b>	Crypto accelerator source code for MbedTLS library.
<b>Device</b>	CMSIS compliant device header files.
<b>JpegAcceleratorLib</b>	SIMD accelerator library binary and header files for libjpeg.
<b>PowerDeliveryLib</b>	Power delivery library binary and header files for dual, source and sink role.
<b>SmartcardLib</b>	Smart card library binary and header files.
<b>StdDriver</b>	All peripheral driver header and source files.
<b>Storage</b>	Disk I/O modules for FatFs.
<b>UsbHostLib</b>	USB host library source code.

### 3 SampleCode

<b>CortexM55</b>	Cortex®-M55 sample codes.
<b>Crypto</b>	Crypto sample codes using MbedTLS library.
<b>FreeRTOS</b>	Simple FreeRTOS™ demo codes.
<b>Hard_Fault_Sample</b>	<p>Show hard fault information when hard fault happened.</p> <p>The hard fault handler shows some information including program counter, which is the address where the processor is executed when the hard fault occurs. The listing file (or map file) can show what function and instruction that is.</p> <p>It also shows the Link Register (LR), which contains the return address of the last function call. It can show the status where CPU comes from to get to this point.</p>
<b>ISP</b>	Sample codes for In-System-Programming.
<b>NuMaker-M5531</b>	Sample codes for NuMaker-M5531 board.
<b>PowerDelivery</b>	Sample codes for power delivery on UTCPD board.
<b>PowerManagement</b>	Sample codes for power management.
<b>SecureApplication</b>	<p>Sample codes for secure application.</p> <p>VS Code projects require Python 3.12 at least for post-build.</p>
<b>Semihost<sup>1</sup></b>	Show how to print and get character through IDE console window.
<b>StdDriver</b>	Sample codes to demonstrate the usage of M5531 series MCU peripheral driver APIs.
<b>Template</b>	Project template for M5531 series MCU.

<sup>1</sup> Semihost sample does not provide VS Code project because semihosting is insufficiently supported in VS Code.

---

<b>TrustZone</b>	Demo of secure codes and non-secure codes.
<b>XOM</b>	Demonstrate how to create XOM library and use it.

---



## 4 ThirdParty

<b>FatFs</b>	<p>An open-source FAT/exFAT file system library. A generic FAT file system module for small embedded systems.</p> <p>Its official website is <a href="http://elm-chan.org/fsw/ff/00index_e.html">http://elm-chan.org/fsw/ff/00index_e.html</a>.</p>
<b>FreeRTOS</b>	<p>Real-time operating system for microcontrollers.</p> <p>Its official website is <a href="http://www.freertos.org/">http://www.freertos.org/</a>.</p>
<b>libjpeg</b>	<p>A software implements JPEG baseline, extended-sequential, and progressive compression processes maintained and published by the Independent JPEG Group (IJG).</p> <p>Its official website is <a href="http://ijg.org/">http://ijg.org/</a>.</p>
<b>libmad</b>	<p>A MPEG audio decoder library that 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.</p> <p>This library is distributed under GPL license. Please contact <a href="#">Underbit Technologies</a> for the commercial license.</p>
<b>lwIP</b>	<p>A widely used open-source TCP/IP stack designed for embedded systems.</p> <p>Its official website is <a href="http://savannah.nongnu.org/projects/lwIP/">http://savannah.nongnu.org/projects/lwIP/</a>.</p>
<b>mbedtls</b>	<p>Mbed TLS offers an SSL library with an intuitive API and readable source code, so you can understand what the code does.</p> <p>Its official website is <a href="https://tls.mbed.org/">https://tls.mbed.org/</a>.</p>
<b>openmv</b>	<p>An open-source, low-cost machine vision platform. It supports image processing functions such as face detection, key-pointes descriptions, color tracking, QR and Bar code decoding and more.</p> <p>Its office website is <a href="https://openmv.io/">https://openmv.io/</a>. This BSP only includes imlib (core library). Python interface is not included.</p>

<b>paho.mqtt.embedded-c</b>	Eclipse Paho MQTT C/C++ client for Embedded platforms. Its official website is <a href="https://www.eclipse.org/paho/clients/c/embedded/">https://www.eclipse.org/paho/clients/c/embedded/</a> .
<b>shine</b>	A blazing fast MP3 encoding library implemented in fixed-point arithmetic. Its official website is <a href="https://github.com/toots/shine">https://github.com/toots/shine</a> .

## 5 Tool

<b>imgtool.exe</b>	<p>Used to perform the operations that are necessary to manage keys and sign images.</p> <p>It is converted from Python script, <code>imgtool.py</code> or <a href="https://github.com/mcu-tools/mcuboot/tree/main/scripts/imgtool">https://github.com/mcu-tools/mcuboot/tree/main/scripts/imgtool</a>.</p>
<b>OTAServerDemo_v2.2.1.apk</b>	<p>Secure OTA server Android APP to download firmware images with Secure OTA sample codes.</p> <p>Its GitHub repository is <a href="https://github.com/OpenNuvoton/NuOTADemo-Android">https://github.com/OpenNuvoton/NuOTADemo-Android</a>.</p>

## 6 SampleCode\CortexM55

<b>Helium_RGB565to888</b>	RG565 to RGB888 with Helium.
<b>MPU_Cache_UpdatePolicy</b>	Configure memory region as write-back or write-through region.
<b>MPU_Memory_AccessPermission</b>	Configure memory region as read-only region.
<b>MPU_Memory_ExecutePermission</b>	Configure memory region as non-executable region.
<b>TCM_Benchmark</b>	Performance comparison of CRC32 implementation on I/D-TCM vs. Flash/SRAM.

## 7 SampleCode\Crypto

<b>MBEDTLS_AES</b>	Show how mbedtls AES function works.
<b>MBEDTLS_ECDH</b>	Show how mbedtls ECDH function works.
<b>MBEDTLS_ECDSA</b>	Show how mbedtls ECDSA function works.
<b>MBEDTLS_HKDF</b>	Show how mbedtls HKDF function works.
<b>MBEDTLS_RSA</b>	Show how mbedtls RSA function works.
<b>MBEDTLS_SHA256</b>	Show how mbedtls SHA256 function works.

## 8 SampleCode\FreeRTOS

<b>Blinky</b>	This project provides two demo applications. A simple blinky style project, and a more comprehensive test and demo application.
<b>TicklessIdle</b>	Show how to enable FreeRTOS™ tickless idle mode, and enable RTC for long duration task idle.
<b>TrustZone</b>	Show how to enable FreeRTOS™ on TrustZone application.

## 9 SampleCode\ISP

<b>common</b>	Simplified system files, drivers and ISP command handle function due to LDROM code size limitation.
<b>ISP_CAN</b>	In-System-Programming Sample code through CAN interface.
<b>ISP_DFU</b>	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
<b>ISP_DFU_20</b>	In-System-Programming Sample code through HSUSB interface and following Device Firmware Upgrade Class Specification.
<b>ISP_HID</b>	In-System-Programming Sample code through USB HID interface.
<b>ISP_HID_20</b>	In-System-Programming Sample code through HSUSB HID interface.
<b>ISP_I2C</b>	In-System-Programming Sample code through I <sup>2</sup> C interface.
<b>ISP_MSC</b>	In-System-Programming Sample code through USB interface and following Mass Storage Class Specification.
<b>ISP_RS485</b>	In-System-Programming Sample code through RS485 interface.
<b>ISP_SPI</b>	In-System-Programming Sample code through SPI interface.
<b>ISP_UART</b>	In-System-Programming Sample code through UART interface.

## 10 SampleCode\NuMaker

<b>AWF_GSensor_Wakeup</b>	Demonstrate how to use G-senor to wake-up by AWF through low power domain IP: LPTMR, LPPDMA, LPI2C.
<b>lwIP</b>	Common drivers for lwIP samples.
<b>LwIP_httpd_netconn</b>	A simple HTTP server that demonstrates lwIP netconn API under FreeRTOS™. This HTTP server's IP address can be configured statically to 192.168.1.2, or assigned by DHCP server.
<b>LwIP_httpd_socket</b>	A simple HTTP server that demonstrates lwIP socket API under FreeRTOS™. This HTTP server's IP address can be configured statically to 192.168.1.2, or assigned by DHCP server.
<b>LwIP_MQTT</b>	A MQTT client sample. The lower level MQTT client functions are from eclipse paho.
<b>LwIP_SSL_Client</b>	A simple HTTPS client that sends a fixed request and displays the response.
<b>LwIP_SSL_Server</b>	A simple HTTPS server that sends a fixed response. It serves a single client at a time.
<b>LwIP_TCP_EchoClient</b>	A TCP echo client which is implemented with lwIP under FreeRTOS™. This client sends "nuvoton" string to server.
<b>LwIP_TCP_EchoServer</b>	A TCP echo server which is implemented with lwIP under FreeRTOS™. The server listens to port 80, and its IP address can 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.
<b>LwIP_tftp_client</b>	A TFTP client sample that can receive a file from TFTP server or send a file to TFTP server.
<b>LwIP_tftp_server</b>	A TFTP server sample that communicates with TFTP client.
<b>LwIP_UDP_EchoClient</b>	A UDP echo client which is implemented with lwIP under FreeRTOS™. This client sends "Hi there..." string to the server.



<b>LwIP_UDP_EchoServer</b>	A UDP echo server which is implemented with lwIP under FreeRTOS™. The echo server listens to port 80, and its IP address can 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.
<b>MP3_Recorder</b>	MP3 recorder sample records sound to MP3 files stored on SD memory card or USB storage device, and press button to play it.
<b>WiFi_bypass</b>	A simple Wi-Fi demo for NuMaker board.
<b>Xmodem</b>	Demonstrate how to transfer data with UART Xmodem.

## 11 SampleCode\PowerDelivery

<b>UTCPD_DualRole</b>	Demonstrate Dual Role Power Device. Support Keil® MDK only.
<b>UTCPD_SinkRole</b>	Demonstrate Sink Power Device. Support Keil® MDK only.
<b>UTCPD_SourceRole</b>	Demonstrate Source Power Device. Support Keil® MDK only.

## 12 SampleCode\PowerManagement

The M5531 series MCU provides some power modes with different power consumption level and wake-up time. For more information, please refer to the application note of M55M1/M5531 Series Power Management and Low Power Design.

<b>PMC_DPDMode_Wakeup</b>	Show how to wake up system form DPD Power-down mode by different wakeup sources.
<b>PMC_NPDMode_Wakeup</b>	Show how to wake up system form NPD Power-down mode by different wakeup sources.
<b>PMC_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering Power-down mode.
<b>PMC_PowerMode</b>	Show how to set different core voltage and main voltage regulator type.
<b>PMC_SPDMode_Wakeup</b>	Show how to wake up system form SPD Power-down mode by different wakeup sources.
<b>PMC_SPDMode_WakeupAndReturn</b>	Show how to continue executing code after wake-up form SPD Power-down mode by SRAM data retention function.
<b>PMC_SPDMode_WakeupVTOR</b>	Show how to execute code after wake-up form SPD Power-down mode by VTOR function.
<b>PMC_SRAMPowerMode</b>	Show how to select SRAM power mode.

## 13 SampleCode\SecureApplication

<b>SecureBootDemo</b>	Demonstrate how to generate the first booting image, NuBL2. After NuBL2 runs, NuBL2 will authenticate NuBL32 and NuBL33 then jump to execute in NuBL32.
<b>SecureISPDemo</b>	Demonstrate how to initialize a SecureISP client mode. This sample code needs to work with <a href="#">USBH_SecureISP</a> sample code.
<b>SecureOTA_SPIM_Cipher</b>	Demonstrate to do secure OTA update for NuBL32 and NuBL33 firmware on SPI Flash by NuBL2 with OTFC enabled.
<b>SecureOTABankRemapDemo</b>	Demonstrate to do secure OTA update for NuBL32 and NuBL33 firmware by NuBL2 with bank remap.
<b>SecureOTADemo</b>	Demonstrate to update NuBL32 and NuBL33 firmware securely over the air (OTA) by NuBL2.
<b>USBH_SecureISP</b>	Demonstrate how to initialize a SecureISP server mode via USB host connected to SecureISPDemo. This sample code needs to work with <a href="#">SecureISPDemo</a> sample code.

## 14 SampleCode\StdDriver

### Analog Comparator Controller (ACMP)

<b>ACMP_ComapreDAC</b>	Demonstrate ACMP comparison by comparing ACMP1_P0 input and DAC voltage and show the result on UART console.
<b>ACMP_ComapreVBG</b>	Demonstrate ACMP comparison by comparing ACMP1_P0 input and VBG voltage and show the result on UART console.
<b>ACMP_Wakeup</b>	Use ACMP to wake up system from Power-down mode while comparator output changes.
<b>ACMP_WindowCompare</b>	Show how to monitor ACMP input with window compare function.
<b>ACMP_WindowLatch</b>	Demonstrate how to use ACMP window latch mode.

### Awake Filter (AWF)

<b>AWF_Wakeup</b>	Demonstrate how to use AWF wake-up function.
-------------------	--

### Basic PWM Generator and Capture Timer (BPWM)

<b>BPWM_Capture</b>	Use BPWM0 channel 0 to capture the BPWM1 channel 0 waveform.
<b>BPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by BPWM double buffer function.
<b>BPWM_OutputWaveform</b>	Demonstrate how to use BPWM counter output waveform.
<b>BPWM_SwitchDuty</b>	Change duty cycle of output waveform by configured period.
<b>BPWM_SyncStart</b>	Demonstrate how to use BPWM counter synchronous start function.

## Controller Area Network with Flexible Data-Rate (CAN FD)

<b>CANFD_CAN_Loopback</b>	Use CAN mode function to do internal loopback test.
<b>CANFD_CAN_MonitorMode</b>	Use CAN Monitor mode to monitor the CAN bus communication test.
<b>CANFD_CAN_TxRx</b>	Transmit and receive CAN messages through CAN interface.
<b>CANFD_CAN_TxRxINT</b>	An example of interrupt control using CAN bus communication.
<b>CANFD_CANFD_Loopback</b>	Use CAN FD mode function to do internal loopback test.
<b>CANFD_CANFD_MonitorMode</b>	Use CAN FD Monitor mode to monitor the CAN bus communication test.
<b>CANFD_CANFD_TxRx</b>	Transmit and receive CAN FD messages through CAN interface.
<b>CANFD_CANFD_TxRxINT</b>	An example of interrupt control using CAN FD bus communication.

## Camera Capture Interface Controller (CCAP)

<b>CCAP_Mono_1Bit_Luma</b>	Use luminance 8-bit to 1-bit conversion to store captured image from HM01B0 sensor to SRAM.
<b>CCAP_MotionDetection_Wake up</b>	Demonstrate how to set up CCAP motion detection function to wake up system when motion is detected in specified regions under Power-down mode.
<b>CCAP_Packet_DownScale</b>	Use packet format (all the luma and chroma data interleaved) to store captured image from sensor to SRAM.
<b>CCAP_Packet_JpegEncode</b>	Use packet format (all the luma and chroma data interleaved) to store captured image from sensor to SRAM and encode image to jpeg.
<b>CCAP_Planar_JpegEncode</b>	Use planar format (split the luma and chroma data) to store captured image from sensor to SRAM and encode the image to jpeg.

## Clock Controller (CLK)

<b>CLK_ClockDetector</b>	Show the usage of clock fail detector and clock frequency monitor function.
--------------------------	---

## CRC Controller (CRC)

<b>CRC_CCITT</b>	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
<b>CRC_CRC8</b>	Implement CRC in CRC-8 mode and get the CRC checksum result.
<b>CRC_CRC32</b>	Implement CRC in CRC-32 mode and get the CRC checksum result.
<b>CRC_POLYNOMIAL</b>	Demonstrate how to use polynomial mode and get the CRC checksum result.

## Cryptographic Accelerator (CRYPTO)

<b>CRYPTO_AES</b>	Show Crypto IP AES-128 ECB mode encrypt/decrypt function.
<b>CRYPTO_AES_CCM</b>	Demonstrate how to encrypt/decrypt data by AES CCM.
<b>CRYPTO_AES_GCM</b>	Demonstrate how to encrypt/decrypt data by AES GCM.
<b>CRYPTO_ECC_Demo</b>	ECDSA signature and verification demo.
<b>CRYPTO_ECC_ECDH</b>	Demonstrate how ECDH calculates share key by private key A and private key B.
<b>CRYPTO_ED25519</b>	ED25519 signature and verification demo.
<b>CRYPTO_HMAC</b>	Show Crypto IP HMAC function.
<b>CRYPTO_RSA</b>	Show how to use Crypto RSA engine to sign and verify signatures.
<b>CRYPTO_RSA_AccessKeyStore</b>	Use Crypto RSA engine accesses key from key store to sign and verify signatures.

<b>CRYPTO_RSA_CRTBypass</b>	Show how to use Crypto RSA engine CRT/CRT bypass mode to sign two signatures.
<b>CRYPTO_RSA_CRTBypassAccessKeyStore</b>	Use Crypto RSA engine CRT/CRT bypass mode accesses key from key store to sign and verify signatures.
<b>CRYPTO_SHA</b>	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.

### Digital to Analog Converter (DAC)

<b>DAC_EPWMTrigger</b>	Demonstrate how to trigger DAC by EPWM.
<b>DAC_ExtPinTrigger</b>	Demonstrate how to trigger DAC conversion by external pin.
<b>DAC_GroupMode</b>	Demonstrate that DAC0 and DAC1 work in group mode.
<b>DAC_PDMA_EPWMTrigger</b>	Demonstrate how to use PDMA and trigger DAC0 by EPWM.
<b>DAC_PDMA_TimerTrigger</b>	Demonstrate how to PDMA and trigger DAC by Timer.
<b>DAC_SoftwareTrigger</b>	Demonstrate how to trigger DAC conversion by software.
<b>DAC_TimerTrigger</b>	Demonstrate how to trigger DAC by Timer.

### Digital Microphone Inputs (DMIC)

<b>DMIC_I2S_Play</b>	Demonstrate how to use DMIC as audio input (MIC) and I2S as audio output (SPK). User can process audio data before output. Data have been transferred via LPPDMA/PDMA.
<b>DMIC_VAD_Wakeup</b>	Demonstrate how to use DMIC VAD function to wake up system from Power-down mode periodically.
<b>DMIC_WAVRecorder</b>	Demonstrate how to use DMIC as audio input (MIC) and record to WAV file.



## Debug Protection Mechanism (DPM)

<b>DPM_UpdatePassword</b>	Demonstrate how to update and compare Secure DPM password.
---------------------------	--

## Enhanced 12-bit Analog-to-Digital Converter (EADC)

<b>EADC_Accumulate</b>	Demonstrate how to get accumulated conversion result.
<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to do the EADC continuous scan conversion.
<b>EADC_Average</b>	Demonstrate how to get average conversion result.
<b>EADC_AverageCMP</b>	Demonstrate how to compare average conversion result.
<b>EADC_BandGap</b>	Convert Band-gap and print conversion result.
<b>EADC_EPWM_Trigger</b>	Demonstrate how to trigger EADC by EPWM.
<b>EADC_PDMA_EPWM_Trigger</b>	Demonstrate how to trigger EADC by EPWM and transfer conversion data by PDMA.
<b>EADC_Pending_Priority</b>	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SWTRG_Trigger</b>	Trigger EADC by writing EADC software trigger register.
<b>EADC_TempSensor</b>	Convert temperature sensor and print conversion result.
<b>EADC_Timer_Trigger</b>	Show how to trigger EADC by Timer.
<b>EADC_VBat</b>	Convert VBAT/4 and print conversion result.

## External Bus Interface (EBI)

<b>EBI_NOR</b>	Configure EBI interface to access NOR Flash connected on EBI interface.
<b>EBI_SRAM</b>	Configure EBI interface to access SRAM connected on EBI

interface.

## Enhanced Input Capture Timer (ECAP)

ECAP_GetEQEIFreq	Show how to use ECAP interface to get EQEI frequency.
ECAP_GetInputFreq	Show how to use ECAP interface to get input frequency

## Ethernet MAC Controller (EMAC)

EMAC_Iwiperf	A lwIP iperf sample.
EMAC_TxRx	This Ethernet sample tends to get a DHCP lease from DHCP server. After IP address configured, this sample can reply to PING packets. Show how to use EMAC driver to simply handle RX and TX packets, it is not suitable for performance and stress testing.

## Enhanced PWM Generator and Capture Timer (EPWM)

EPWM_AccumulatorINT_TriggerPDMA	Demonstrate how to use EPWM accumulator interrupt trigger PDMA.
EPWM_AccumulatorStopMode	Demonstrate EPWM accumulator stop mode.
EPWM_Brake	Demonstrate how to use EPWM brake function.
EPWM_Capture	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2.
EPWM_DeadTime	Demonstrate how to use EPWM Dead Zone function.
EPWM_DoubleBuffer	Change duty cycle and period of output waveform by EPWM Double Buffer function.
EPWM_OutputWaveform	Demonstrate how to use EPWM output waveform.
EPWM_PDMA_Capture	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2 and use PDMA to transfer captured data.
EPWM_SwitchDuty	Change duty cycle of output waveform by configured

	period.
EPWM_SyncStart	Demonstrate how to use EPWM counter synchronous start function.

## Enhanced Quadrature Encoder Interface (EQEI)

EQEI_CompareMatch	Show the usage of EQEI compare function.
EQEI_PeriodicINT	Show the usage of EQEI Unit Timer function.

## Flash Memory Controller (FMC)

FMC_APPROT	Demonstrate how to use FMC APROM Protect function.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.
FMC_DualBank	Demonstrate how dual processes work in dual bank flash architecture.
FMC_DualBankFwUpgrade	Implement a firmware update mechanism based on dual bank flash architecture.
FMC_ExecInSRAM	Implement a code and execute it in SRAM to program embedded Flash.
FMC_FwUpgradeApplication	Bank remap sample code.
FMC_IAP	Show how to call LDROM function from APROM.
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_MultiWordProgram	Implement FMC multi word program function executed in ITCM to program embedded APROM.
FMC_OTP	Demonstrate how to program, read and lock OTP.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to check whether the specified APROM or LDROM region are all 0xFFFFFFFF or not.

<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write functions.
<b>FMC_XOM</b>	Show how to configure and set up an XOM region then perform XOM function.

## Graphic DMA Controller (GDMA)

<b>GDMA_2DCopy</b>	Use GDMA channel 0 to do 2D transfer (Rotate and Mirror).
<b>GDMA_2DFill</b>	Use GDMA channel 0 to fill an area with given value.
<b>GDMA_2DWrap</b>	Use GDMA channel 0 to do WRAP for 2D.
<b>GDMA_CommandLink</b>	Use GDMA channel 0 to do Command linking transfer.
<b>GDMA_MemCopyINT</b>	Use GDMA channel 0 to transfer data from memory to memory.
<b>GDMA_TemplateTransfer</b>	Use GDMA channel 0 to do template transfer.

## General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and de-bounce function.
<b>GPIO_INT</b>	Show the usage of GPIO interrupt function.
<b>GPIO_OutputInput</b>	Show how to set GPIO pin mode and use pin data input and output control.
<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.

## High Speed USB 2.0 On-The-Go (HSOTG)

<b>HSOTG_Dual_Role_UMAS</b>	An OTG sample code that 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.
-----------------------------	---

<b>HSOTG_HNP</b>	Show HID mouse with OTG HNP protocol.
------------------	---------------------------------------

## High Speed USB 2.0 Device Controller (HSUSBD)

<b>HSUSBD_Audio10_Codec</b>	An UAC1.0 sample used to record and play the sound sent from PC through the USB interface.
<b>HSUSBD_Audio10_Headset</b>	An UAC1.0 sample and used to plays the sound send from PC through the USB interface.
<b>HSUSBD_Audio20_Codec</b>	An UAC2.0 sample used to record and play the sound sent from PC through the USB interface.
<b>HSUSBD_Audio20_Headset</b>	An UAC2.0 sample used to play the sound sent from PC through the USB interface.
<b>HSUSBD_HID_Mouse</b>	Simulate a USB mouse and draws circle on the screen.
<b>HSUSBD_HID_Mouse_BC12</b>	Demonstrate how to implement a USB mouse device with BC1.2 (Battery Charging). This sample code shows different type of charging port after connected USB port. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
<b>HSUSBD_HID_MouseKeyboard</b>	Simulate a USB mouse and a USB keyboard.
<b>HSUSBD_HID_Transfer</b>	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.
<b>HSUSBD_HID_Transfer_And_MSC</b>	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB device and PC through the USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>HSUSBD_MassStorage_CDROM</b>	USB Mass Storage Device CD-ROM Emulation.
<b>HSUSBD_MassStorage_Flash</b>	Use embedded Data Flash as storage to implement a USB Mass-Storage device.
<b>HSUSBD_MassStorage_Sactt</b>	Demonstrate the usage of USB DMA scatter gather

erGather	function.
HSUSBD_MassStorage_SD	Implement a SD card reader.
HSUSBD_MassStorage_Short Packet	Implement a mass storage class sample to demonstrate how to receive a USB short packet.
HSUSBD_MassStorage_SRAM	Use internal SRAM as backend storage media to simulate a USB pen drive. This sample code requires enough SRAM.
HSUSBD_RNDIS	Demonstrate how to implement a Remote Network Driver Interface Specification (RNDIS) device.
HSUSBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual com port device.
HSUSBD_VENDOR_LBK	Implement a proprietary Vendor LBK device. This sample code requires a USB host running sample <a href="#">HSUSBH_USBH_VENDOR_LBK</a> to be connected.
HSUSBD_Video_CAM	Demonstrate the implementation of USB UVC (USB Video Device Class) functionality.

## USB 2.0/1.1 Host Controller (HSUSBH/USBH)

HSUSBH_USBH_AudioClass	Demonstrate how to use USBH Audio Class driver. It shows the mute, volume, auto-gain, channel, and sampling rate control.
HSUSBH_USBH_DEV_CONN	Use connect/disconnect callback functions to handle of device connect and disconnect events.
HSUSBH_USBH_Firmware_Update	Automatically search and read new firmware from USB drive, if found, update APROM Flash with it.
HSUSBH_USBH_HID	Use USB Host core driver and HID driver. Demonstrate how to submit HID class request and read data from interrupt pipe. This sample code supports dynamic device plug/un-plug and multiple HID devices.
HSUSBH_USBH_HID_Keyboard	Demonstrate reading key inputs from USB keyboards. This sample code includes an USB keyboard driver which is based on the HID driver.

<b>HSUSBH_USBH_HID_MouseKeyboard</b>	Demonstrate how to support USB mouse and keyboard input.
<b>HSUSBH_USBH_MassStorage</b>	Use a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it work as a disk driver under the FATFS file system.
<b>HSUSBH_USBH_PowerSaving</b>	Use power down to do power saving with USB host. System wake up after, user need restore and reset SUB host. It also shows device information represented in UDEV_T.
<b>HSUSBH_USBH_SPIM_Writer</b>	Provide a command line interface for reading files from USB disk and writing to SPIM Flash. This sample code also provides functions of dump SPIM Flash, compares USB disk file with SPIM Flash, and branches to run code on SPIM Flash.
<b>HSUSBH_USBH_UAC_HID</b>	Show how to use USBH Audio Class driver and HID driver at the same time. The target device is a Game Audio (UAC+HID composite device).
<b>HSUSBH_USBH_UAC_Loopback</b>	Receive audio data from an UAC device, and immediately send back to the UAC device.
<b>HSUSBH_USBH_VCOM</b>	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
<b>HSUSBH_USBH_VCOM_MassStorage</b>	This sample uses a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it working as a disk driver under FATFS file system. It also demonstrates how to use CDC driver to connect a CDC class VCOM device.
<b>HSUSBH_USBH_VENDOR_LBK</b>	Show how to do transfer on a known device with a vendor driver. This sample code requires a USB device running sample <a href="#">HSUSBD_VENDOR_LBK</a> or <a href="#">USBD_VENDOR_LBK</a> to be connected.

## I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

<b>I2C_Double_Buffer_Slave</b>	Demonstrate how to set I <sup>2</sup> C two-level buffer in Slave mode to receive data from a master. This sample code needs to
--------------------------------	---

	work with <a href="#">I2C MultiBytes Master</a> sample code.
<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<b>I2C_Loopback</b>	Demonstrate how to set I <sup>2</sup> C Master mode and Slave Mode and show how a master accesses a slave on a chip.
<b>I2C_Master</b>	Show how a master accesses a slave. This sample code needs to work with <a href="#">I2C Slave</a> sample code.
<b>I2C_MultiBytes_Master</b>	Demonstrate how to use multi-byte API to access slave. This sample code needs to work with <a href="#">I2C Slave</a> sample code.
<b>I2C_PDMA_TRX</b>	Demonstrate I <sup>2</sup> C PDMA mode and need to connect I2C0 (master) and I2C1 (slave).
<b>I2C_SingleByte_Master</b>	Demonstrate how to use single byte API to access slave. This sample code needs to work with <a href="#">I2C Slave</a> sample code.
<b>I2C_Slave</b>	Demonstrate how to set I <sup>2</sup> C in Slave mode to receive data from a master. This sample code needs to work with <a href="#">I2C Master</a> sample code.
<b>I2C_SMBus</b>	Demonstrate how to control SMBus interface and use SMBus protocol between Host and Slave.
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode via the I <sup>2</sup> C interface. This sample code could work with <a href="#">I2C Master</a> sample code.

## I<sup>2</sup>S Controller (I<sup>2</sup>S)

<b>I2S_Codec</b>	This is an I <sup>2</sup> S demo using NAU8822/NAU88L25 audio codec, and used to play back the input from line-in.
<b>I2S_Codec_PDMA</b>	This is an I <sup>2</sup> S demo with PDMA function connected with codec.
<b>I2S_MP3Player</b>	MP3 player sample plays MP3 files stored on SD memory card.
<b>I2S_WAVPlayer</b>	This is a WAV file player which plays back WAV file stored



in SD memory card.

## I3C Serial Interface Controller (I3C)

<b>I3C_Master_IBI</b>	Demonstrate how to use I3C Master to process In-Band Interrupt request from an I3C Slave. This sample code needs to work with <a href="#">I3C Slave IBI</a> sample code.
<b>I3C_MasterRW</b>	Demonstrate how to use I3C Master to transmit and receive the data from a Slave. This sample code needs to work with <a href="#">I3C SlaveRW</a> sample code.
<b>I3C_MasterRW_PDMA</b>	Demonstrate how to use I3C Master to receive and transmit the data through PDMA to a Slave. This sample code needs to work with <a href="#">I3C SlaveRW</a> sample code.
<b>I3C_SecondaryMaster</b>	Demonstrate how to switch the role between Master and Slave. This sample code requires the use of two boards.
<b>I3C_Slave_HotJoin</b>	Demonstrate how to use I3C Slave to initiate a Hot-Join request to I3C Master. This sample code needs to work with <a href="#">I3C MasterRW</a> sample code.
<b>I3C_Slave_IBI</b>	Demonstrate how to use I3C Slave to initiate an In-Band Interrupt request to I3C Master. This sample code needs to work with <a href="#">I3C Master IBI</a> sample code.
<b>I3C_Slave_Wakeup</b>	Wake up the MCU from Power-down mode after receiving read/write request from I3C Master. This sample code needs to work with <a href="#">I3C MasterRW</a> sample code.
<b>I3C_SlaveRW</b>	Demonstrate how to use I3C0 as I3C Slave to receive and transmit the data from a Master. This sample code needs to work with <a href="#">I3C MasterRW</a> sample code.
<b>I3C_SlaveRW_PDMA</b>	Demonstrate how to use I3C0 Slave to receive and transmit the data through PDMA from a Master. This sample code needs to work with <a href="#">I3C MasterRW</a> sample code.

## Key Derivation Function (KDF)

KDF_DeriveKey	Show how to use KDF to derive new key.
KDF_DeriveKeyToKeyStore	Show how to use KDF to derive new key saved in Key Store.

## Keypad Interface (KPI)

KPI_Keyboard	Show how to set scan key board by KPI.
--------------	--

## Key Store (KS)

KS_AESKey	Demonstrate how to use the AES in Key Store.
KS_ECDH	Demonstrate how to use ECC ECDH with Key Store.
KS_ECDSA	Demonstrate how to use the ECC ECDSA with Key Store.
KS_KeyStatus	Show how to check Key Store status.

## Low Power Analog-to-Digital Converter (LPADC)

LPADC_ACMP_Trigger	Show how to trigger LPADC by ACMP.
LPADC_ADINT_Trigger	Use ADINT interrupt to do the LPADC Single-cycle scan conversion.
LPADC_AutoOperation	Demonstrate how to enable LPADC Auto-operation mode to convert when chip enters Power-down mode.
LPADC_BandGap	Convert Band-gap and print conversion result.
LPADC_BurstMode	Perform A/D Conversion with LPADC burst mode.
LPADC_ContinuousScanMode	Perform A/D Conversion with LPADC continuous scan mode.
LPADC_EPWM_Trigger	Demonstrate how to trigger LPADC by EPWM.
LPADC_LPPDMA_EPWM_Trigger	Demonstrate how to trigger LPADC by EPWM and transfer conversion data by LPPDMA.

<b>LPADC_MaxSps_ContinuousScanMode</b>	Demonstrate how to use PCLK4 as LPADC clock source to achieve the maximum conversion rate.
<b>LPADC_ResultMonitor</b>	Monitor the conversion result of channel 1 by the digital compare function.
<b>LPADC_SingleCycleScanMode</b>	Perform A/D Conversion with LPADC single cycle scan mode.
<b>LPADC_SingleMode</b>	Perform A/D Conversion with LPADC single mode.
<b>LPADC_STADC_Trigger</b>	Show how to trigger LPADC by STADC pin.
<b>LPADC_SWTRG_Trigger</b>	Trigger LPADC by writing LPADC software trigger register.

### Low Power General Purpose I/O (LPGPIO)

<b>LPGPIO_OutputInput</b>	Show how to set LPGPIO pin mode and use pin data input/output control.
---------------------------	--

### Low Power I2C Serial Interface Controller (LPI2C)

<b>LPI2C_AutoOperation</b>	Demonstrate LPI2C Auto-operation mode when chip enters Power-down mode. This sample code needs to work with <a href="#">LPI2C_Slave</a> sample code.
<b>LPI2C_EEPROM</b>	Show how to use LPI2C interface to access EEPROM.
<b>LPI2C_LPPDMA_TRX</b>	Demonstrate LPI2C LPPDMA mode and need two boards to connect LPI2C0 (Master) and LPI2C0 (Slave).
<b>LPI2C_Master</b>	Show how a master accesses a slave. This sample code needs to work with <a href="#">LPI2C_Slave</a> sample code.
<b>LPI2C_MultiBytes_Master</b>	Show how to use LPI2C multi-bytes API to access Slave. This sample code needs to work with <a href="#">LPI2C_Slave</a> sample code.
<b>LPI2C_SingleByte_Master</b>	Show how to use LPI2C single-byte API to access Slave. This sample code needs to work with <a href="#">LPI2C_Slave</a> sample code.

<b>LPI2C_Slave</b>	Show how to set LPI2C in Slave mode and receive the data from Master. This sample code needs to work with <a href="#">LPI2C_Master</a> sample code.
<b>LPI2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through LPI2C interface. This sample code needs to work with <a href="#">LPI2C_Master</a> sample code.

## Low Power PDMA Controller (LPPDMA)

<b>LPPDMA_BasicMode</b>	Use LPPDMA to transfer data from memory to memory.
<b>LPPDMA_ScatterGather</b>	Use LPPDMA to transfer data from memory to memory by scatter-gather mode.
<b>LPPDMA_ScatterGather_PingPongBuffer</b>	Use LPPDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

## Low Power Serial Peripheral Interface (LPSPI)

<b>LPSPI_AutoOperation</b>	Demonstrate how to do LPSPI loopback test in Auto-operation mode when chip enters Power-down mode
<b>LPSPI_Loopback</b>	LPSPI read/write demo by connecting LPSPI MISO and MOSI pins.
<b>LPSPI_LPPDMA_LoopTest</b>	LPSPI read/write demo in LPPDMA mode. Connecting LPSPI MISO and MOSI pins. Both TX LPPDMA function and RX LPPDMA function will be enabled.
<b>LPSPI_MasterFIFOmode</b>	Configure LPSPI 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 <a href="#">LPSPI_SlaveFIFOmode</a> sample code.
<b>LPSPI_SlaveFIFOmode</b>	Configure LPSPI 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 <a href="#">LPSPI_MasterFIFOmode</a> sample code.

## Low Power Timer Controller (LPTMR)

<b>LPTMR_ACMPTrigger</b>	Use ACMP to trigger LPTMR counter reset mode.
<b>LPTMR_AutoOperation</b>	Demonstrate LPTMR Auto-operation with LPPDMA when chip enters Power-down mode.
<b>LPTMR_CaptureCounter</b>	Show how to use the LPTMR capture function to capture LPTMR counter value.
<b>LPTMR_Delay</b>	Demonstrate the usage of LPTMR_Delay() API to generate a 1 second delay.
<b>LPTMR_EventCounter</b>	Use LPTM0 pin to demonstrate LPTMR event counter function.
<b>LPTMR_FreeCountingMode</b>	Use the LPTMR LPTM0_EXT pin to demonstrate timer free counting mode function. Also display the measured input frequency to UART console.
<b>LPTMR_Periodic</b>	Use the LPTMR periodic mode to generate timer interrupt every 1 second.
<b>LPTMR_PeriodicINT</b>	Implement LPTMR counting in periodic mode.
<b>LPTMR_PWM_AccumulatorINTStopMode</b>	Demonstrate LPTMR PWM accumulator interrupt to stop counting.
<b>LPTMR_PWM_AccumulatorINTTriggerLPPDMA</b>	Demonstrate LPTMR PWM accumulator interrupt to trigger LPPDMA transfer.
<b>LPTMR_PWM_AutoOperation</b>	Demonstrate LPTPWM Auto-operation with LPPDMA when chip enters Power-down mode.
<b>LPTMR_PWM_ChangeDuty</b>	Change duty cycle and period of output waveform in PWM up count type.
<b>LPTMR_PWM_OutputWaveform</b>	Demonstrate output different duty waveform in LPTMR0~1 PWM.
<b>LPTMR_TimeoutWakeup</b>	Use LPTMR to wake up system from Power-down mode periodically.
<b>LPTMR_ToggleOut</b>	Demonstrate the LPTMR toggle out function on LPTM0 pin.

## Low Power UART Interface Controller (LPUART)

<b>LPUART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>LPUART_AutoFlow</b>	Transmit and receive data using auto flow control.
<b>LPUART_AutoOperation</b>	Demonstrate how to enable LPUART Auto-operation mode to transmit and receive data when chip enters Power-down mode.
<b>LPUART_LPPDMA</b>	Demonstrate LPUART transmit and receive function with LPPDMA.
<b>LPUART_RS485</b>	Transmit and receive data in LPUART RS485 mode.
<b>LPUART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>LPUART_Wakeup</b>	Show how to wake up system from Power-down mode by LPUART interrupt.

## USB 1.1 On-The-Go (OTG)

<b>OTG_Dual_Role_UMAS</b>	An OTG sample code that 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.
<b>OTG_HNP</b>	Show HID mouse with OTG HNP protocol.

## PDMA Controller (PDMA)

<b>PDMA_BasicMode</b>	Use PDMA0 channel 2 to transfer data from memory to memory.
<b>PDMA_ScatterGather</b>	Use PDMA0 channel 4 to transfer data from memory to memory by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

<b>PDMA_TimeOut</b>	Demonstrate PDMA0 channel 1 get/clear timeout flag with UART1.
---------------------	--

## Programmable Serial I/O (PSIO)

<b>PSIO_1Wire</b>	Demonstrate how to implement 1-Wire protocol by PSIO.
<b>PSIO_DM512</b>	Demonstrate how to implement DM512 protocol by PSIO.
<b>PSIO_HDQ</b>	Demonstrate how to implement HDQ protocol by PSIO.
<b>PSIO_IR</b>	Demonstrate how to implement NEC IR protocol by PSIO.
<b>PSIO_LED</b>	Demonstrate how to light up the WS1812B LED array.
<b>PSIO_Microwire</b>	Demonstrate how to implement Microwire protocol by PSIO.
<b>PSIO_PS2_Device</b>	Demonstrate how to implement PS/2 slave protocol by PSIO.
<b>PSIO_PS2_Host</b>	Demonstrate how to implement PS/2 host protocol by PSIO.
<b>PSIO_Wiegand</b>	Demonstrate how to implement Wiegand26 protocol by PSIO.

## Quad Serial Peripheral Interface (QSPI)

<b>QSPI_DualMode_Flash</b>	Access SPI Flash using QSPI dual mode.
<b>QSPI_PDMA_Master</b>	Configure QSPI0 as Master mode and demonstrate how to communicate with an off-chip QSPI Slave device with FIFO mode. This sample code needs to work with <a href="#">QSPI PDMA Slave</a> sample code.
<b>QSPI_PDMA_Slave</b>	Configure QSPI as Slave mode and demonstrate how to communicate with an off-chip QSPI Master device with FIFO mode. This sample code needs to work with <a href="#">QSPI PDMA Master</a> sample code.
<b>QSPI_QuadMode_Flash</b>	Access SPI Flash using QSPI quad mode.

<b>QSPI_Slave3Wire</b>	Configure QSPI as Slave 3-wire mode and demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with <a href="#">SPI_MasterFIFOmode</a> sample code.
------------------------	---

## Random Number Generator (RNG)

<b>RNG_EntropyPoll</b>	Generate entropy from hardware entropy source.
<b>RNG_Random</b>	Generate random numbers.

## Real Timer Clock (RTC)

<b>RTC_Alarm_Test</b>	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
<b>RTC_Alarm_Wakeup</b>	Use RTC alarm interrupt event to wake up system.
<b>RTC_Dynamic_Tamper</b>	Demonstrate the RTC dynamic tamper function.
<b>RTC_Spare_Access</b>	Demonstrate the RTC spare register read/write function and displays test result to the UART console.
<b>RTC_Static_Tamper</b>	Demonstrate the RTC static tamper function.
<b>RTC_Time_Display</b>	Demonstrate the RTC function and display current time to the UART console.

## Smart Card Host Interface (SC)

<b>SC_ReadATR</b>	Read the smartcard ATR from Smartcard interface.
<b>SC_ReadSimPhoneBook</b>	Demonstrate how to read phone book information in the SIM card.
<b>SC_Timer</b>	Demonstrate how to use SC embedded timer.
<b>SCUART_TxRx</b>	Demonstrate Smartcard UART mode by connecting PB.4 and PB.5 pins.



## Secure Digital Host Controller (SDH)

<b>SDH_FATFS</b>	Access a SD card formatted in FAT file system.
------------------	--

## Serial Peripheral Interface (SPI)

<b>SPI_Flash</b>	Access SPI Flash through SPI interface.
<b>SPI_HalfDuplex</b>	Demonstrate SPI half-duplex mode. Configure SPI0 as master mode and SPI1 as slave mode. Both SPI0 and SPI1 are half-duplex mode.
<b>SPI_Loopback</b>	Implement SPI Master loop back transfer. This sample code needs to connect MISO pin and MOSI pin together. It will compare the received data with transmitted data.
<b>SPI_MasterFIFOmode</b>	Configure SPI as master mode and demonstrate how to communicate with an off-chip SPI slave device with FIFO mode. This sample code could work with <a href="#">SPI_SlaveFIFOmode</a> sample code.
<b>SPI_PDMA_LoopTest</b>	SPI read/write demo in PDMA mode. Connect SPI MISO and MOSI pins. Both TX PDMA function and RX PDMA function will be enabled.
<b>SPI_SlaveFIFOmode</b>	Configure SPI 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 <a href="#">SPI_MasterFIFOmode</a> sample code.
<b>SPII2S_Master</b>	Configure SPI as I <sup>2</sup> S master mode and demonstrate how I <sup>2</sup> S works in master mode. This sample code needs to work with <a href="#">SPII2S_Slave</a> sample code.
<b>SPII2S_PDMA_Codec</b>	An I <sup>2</sup> S demo with PDMA function connected with audio codec.
<b>SPII2S_PDMA_Play</b>	An I <sup>2</sup> S demo for playing data and demonstrating how I <sup>2</sup> S works with PDMA.
<b>SPII2S_PDMA_PlayRecord</b>	An I <sup>2</sup> S demo for playing and recording data with PDMA function.
<b>SPII2S_PDMA_Record</b>	An I <sup>2</sup> S demo for recording data and demonstrating how I <sup>2</sup> S

	works with PDMA.
<b>SPII2S_Slave</b>	Configure SPI as I <sup>2</sup> S slave mode and demonstrate how I <sup>2</sup> S works in slave mode. This sample code needs to work with <a href="#">SPII2S_Master</a> sample code.

## **SPI/Hyper Bus Synchronous Serial Interface Controller (SPIM)**

<b>SPIM_Cipher</b>	Show SPIM DMA read/write with cipher enabled. This sample code also dumps SPI Flash content via I/O mode read to prove it is encrypted cipher context.
<b>SPIM_DMA_RW</b>	Show SPIM DMA mode read/write function.
<b>SPIM_DMM</b>	Show SPIM DMM mode read function. This sample code programs SPI Flash with DMA write and verify flash with DMA read and DMM mode CPU read respectively.
<b>SPIM_DMM_RUN_CODE</b>	Show how to make an application booting from APROM with a sub-routine resided on SPI Flash.
<b>SPIM_HYPER_ExeInHRAM</b>	Show how to make an application booting from APROM with a sub-routine resided on HyperRAM.
<b>SPIM_HYPER_RW_MemMap</b>	Show HyperRAM read/write through HyperBus Interface.
<b>SPIM_IO_RW</b>	Show how to issue SPI Flash erase, program, and read commands under SPIM I/O mode.
<b>SPIM_OctalMode_Flash</b>	Show SPIM DMA mode read/write octal flash function.

## **System Manager (SYS)**

<b>SYS_BODWakeup</b>	Show how to wake up system from Power-down mode by brown-out detector interrupt.
<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLKO pin.
<b>SYS_TrimHIRC</b>	Demonstrate how to use LXT to trim HIRC/HIRC48M.

## Timer Controller (TIMER)

<b>TIMER_ACMPTrigger</b>	Use ACMP to trigger Timer counter reset mode.
<b>TIMER_CaptureCounter</b>	Show how to use the Timer capture function to capture Timer counter value.
<b>TIMER_Delay</b>	Demonstrate the usage of TIMER_Delay API to generate a 1 second delay.
<b>TIMER_EventCounter</b>	Use TM0 pin to demonstrate Timer event counter function.
<b>TIMER_FreeCountingMode</b>	Use TM0_EXT pin to demonstrate Timer free counting mode function. And display the measured input frequency to UART console.
<b>TIMER_InterTimerTriggerMode</b>	Use TM0 pin to demonstrate inter-timer trigger mode function and display the measured input frequency to UART console.
<b>TIMER_Periodic</b>	Use the Timer periodic mode to generate Timer interrupt every 1 second.
<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_PWM_AccumulatorINT StopMode</b>	Demonstrate TIMER PWM accumulator interrupt to stop counting.
<b>TIMER_PWM_AccumulatorINT TriggerPDMA</b>	Demonstrate TIMER PWM accumulator interrupt to trigger PDMA transfer.
<b>TIMER_PWM_Brake</b>	Demonstrate how to use TIMER PWM brake function.
<b>TIMER_PWM_ChangeDuty</b>	Change duty cycle and period of output waveform in PWM down count type.
<b>TIMER_PWM_DeadTime</b>	Demonstrate TIMER PWM Complementary mode and Dead-Time function.
<b>TIMER_PWM_OutputWaveform</b>	Demonstrate output different duty waveform in Timer0~Timer3 PWM.
<b>TIMER_TimeoutWakeup</b>	Use Timer to wake up system from Power-down mode periodically.

<b>TIMER_ToggleOut</b>	Demonstrate the Timer toggle out function.
------------------------	--

## Tick Timer Controller (TTMR)

<b>TTMR_AutoOperation</b>	Demonstrate TTMR Auto-operation with LPPDMA when chip enters Power-down mode.
<b>TTMR_Delay</b>	Demonstrate the usage of TTMR_Delay API to generate a 1 second delay.
<b>TTMR_Periodic</b>	Use the TTMR periodic mode to generate timer interrupt every 1 second.
<b>TTMR_PeriodicINT</b>	Implement TTMR counting in periodic mode.
<b>TTMR_TimeoutWakeup</b>	Use TTMR to wake up system from Power-down mode periodically.

## UART Interface Controller (UART)

<b>UART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>UART_AutoFlow</b>	Transmit and receive data using auto flow control.
<b>UART_IrDA</b>	Transmit and receive UART data in UART IrDA mode.
<b>UART_LIN</b>	Transmit LIN frame including header and response in UART LIN mode.
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.
<b>UART_RS485</b>	Transmit and receive data in UART RS485 mode.
<b>UART_SingleWire</b>	Transmit and receive data in UART single-wire mode.
<b>UART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.

## USB 1.1 Device Controller (USB D)

<b>USBD_Audio_Codec</b>	Demonstrate how to implement a USB audio class device.
<b>USBD_Audio_Headset</b>	Demonstrate how to implement a USB audio class device. Codec is used in this sample code to play the audio data from Host. It also supports to record data from codec to Host.
<b>USBD_CCID</b>	USB CCID smart card reader sample code.
<b>USBD_HID_Keyboard</b>	Demonstrate how to implement a USB keyboard device. It supports to use GPIO to simulate key input.
<b>USBD_HID_Mouse</b>	Show how to implement a USB mouse device. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
<b>USBD_HID_MouseKeyboard</b>	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
<b>USBD_HID_RemoteWakeup</b>	Demonstrate how to implement a USB mouse device. It uses PA0 ~ PA5 to control mouse direction and mouse key. It also supports USB suspend and remote wakeup.
<b>USBD_HID_Touch</b>	Demonstrate how to implement a USB touch digitizer device. Two lines demo in Paint.
<b>USBD_HID_Transfer</b>	Demonstrate how to transfer data between a USB device and PC through a USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_HID_Transfer_And_Keyboard</b>	Demonstrate how to implement a composite device of 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 a USB device.
<b>USBD_HID_Transfer_And_MSC</b>	Demonstrate how to implement a composite device of 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.

<b>USBD_HID_Transfer_CTRL</b>	Use USB host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.
<b>USBD_MassStorage_CDROM</b>	Demonstrate the emulation of USB Mass-Storage device, CD-ROM.
<b>USBD_MassStorage_Flash</b>	Use Flash as storage to implement a USB Mass-Storage device.
<b>USBD_MassStorage_SD</b>	Use SD card as storage to implement a USB Mass-Storage device.
<b>USBD_MassStorage_SRAM</b>	Use internal SRAM as backend storage media to simulate a USB pen drive.
<b>USBD_MicroPrinter</b>	Demonstrate how to implement a USB micro printer device.
<b>USBD_Printer_And_HID_Transfer</b>	Demonstrate how to implement a composite device of USB micro printer 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.
<b>USBD_VCOM_And_HID_Keyboard</b>	Demonstrate how to implement a composite device of VCOM and HID keyboard.
<b>USBD_VCOM_And_HID_Transfer</b>	Demonstrate how to implement a composite device of VCOM 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.
<b>USBD_VCOM_And_MassStorage</b>	Demonstrate how to implement a composite device of VCOM and mass storage.
<b>USBD_VCOM_MultiPort</b>	Demonstrate how to implement a USB multiple virtual COM port device.
<b>USBD_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual COM port device.
<b>USBD_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual COM port

<b>_DoubleBuffer</b>	device using double buffer mode.
<b>USBD_VENDOR_LBK</b>	This sample code works as a proprietary Vendor LBK device. It's created for <a href="#">HSUSBH_USBH_VENDOR_LBK</a> of this BSP. Vendor LBK device includes Control, Bulk, Interrupt, and Isochronous in/out endpoint pairs. Each endpoint pair receive data from host via the out-endpoint and send data back to host via the in-endpoint.

## Universal Serial Control Interface Controller - I<sup>2</sup>C Mode (USCI-I2C)

<b>USCI_I2C_EEPROM</b>	Demonstrate how to access EEPROM through a USCI_I2C interface.
<b>USCI_I2C_Master</b>	Demonstrate how a Master accesses Slave. This sample code needs to work with <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_Master_10bit</b>	Demonstrate how a Master uses 10-bit addressing access Slave. This sample code needs to work with <a href="#">USCI_I2C_Slave_10bit</a> sample code.
<b>USCI_I2C_Monitor</b>	Demonstrate how USCI_I2C monitors transmission between I <sup>2</sup> C Master and I <sup>2</sup> C Slave.
<b>USCI_I2C_MultiBytes_Master</b>	Demonstrate how to use multi-byte API to access slave. This sample code needs to work with the <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_SingleByte_Master</b>	Demonstrate how to use single-byte API to access slave. This sample code needs to work with the <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_Slave</b>	Demonstrate how to set USCI_I2C in slave mode to receive the data from a Master. This sample code needs to work with <a href="#">USCI_I2C_Master</a> sample code.
<b>USCI_I2C_Slave_10bit</b>	Demonstrate how to set USCI_I2C in 10-bit addressing slave mode to receive the data from a Master. This sample code needs to work with <a href="#">USCI_I2C_Master_10bit</a> sample code.
<b>USCI_I2C_Wakeup_Slave</b>	Demonstrate how to set USCI_I2C to wake up MCU from Power-down mode. This sample code needs to work with

[USCI\\_I2C\\_Master](#) sample code.

## Universal Serial Control Interface Controller - SPI Mode (USCI-SPI)

<b>USCI_SPI_Loopback</b>	Implement USCI_SPI0 master loop back transfer. This sample code needs to connect USCI_SPI0_MISO pin and USCI_SPI0_MOSI pin together. It will compare the received data with transmitted data.
<b>USCI_SPI_MasterMode</b>	Configure USCI_SPI0 as master mode and demonstrate how to communicate with an off-chip SPI Slave device. This sample code needs to work with <a href="#">USCI_SPI_SlaveMode</a> sample code.
<b>USCI_SPI_PDMA_LoopTest</b>	Demonstrate SPI data transfer with PDMA. USCI_SPI0 will be configured as master mode and USCI_SPI1 will be configured as slave mode. Both Tx PDMA function and Rx PDMA function will be enabled.
<b>USCI_SPI_SlaveMode</b>	Configure USCI_SPI0 as slave mode and demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with <a href="#">USCI_SPI_MasterMode</a> sample code.

## Universal Serial Control Interface Controller - UART Mode (USCI-UART)

<b>USCI_UART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>USCI_UART_Autoflow</b>	Transmit and receive data with auto flow control.
<b>USCI_UART_PDMA</b>	This is a USCI_UART PDMA demo and needs to connect USCI_UART Tx and Rx.
<b>USCI_UART_RS485</b>	Transmit and receive data in RS485 mode.
<b>USCI_UART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>USCI_UART_Wakeup</b>	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.



## Watchdog Timer (WDT)

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.
---------------------------	--

## Window Watchdog Timer (WWDT)

WWDT_CompareINT	Show how to reload the WWDT counter value.
WWDT_ReloadCounter	Demonstrate how to reload the WWDT counter value without resetting the MCU.

## 15 SampleCode\TrustZone

<b>CSSD_LED</b>	Demonstrate how to implement code for Collaborative Secure Software Development in both secure and non-secure code.
<b>HardFault</b>	Show the hard fault usages in both secure and non-secure code.
<b>Template</b>	Demonstrate how to implement code for secure and non-secure.

## 16 SampleCode\XOM

XOMLib	Demonstrate how to create XOM library.
XOMLibDemo	Demonstrate how to use <a href="#">XOMLib</a> .

### **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.*