

M031 Play WAV File from SPI Flash

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses external codec NAU88L25 to play WAV file stored in external SPI Flash W25Q32.
BSP Version	M031 BSP CMSIS V3.00.001
Hardware	NuMaker-M032KG Ver1.0

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

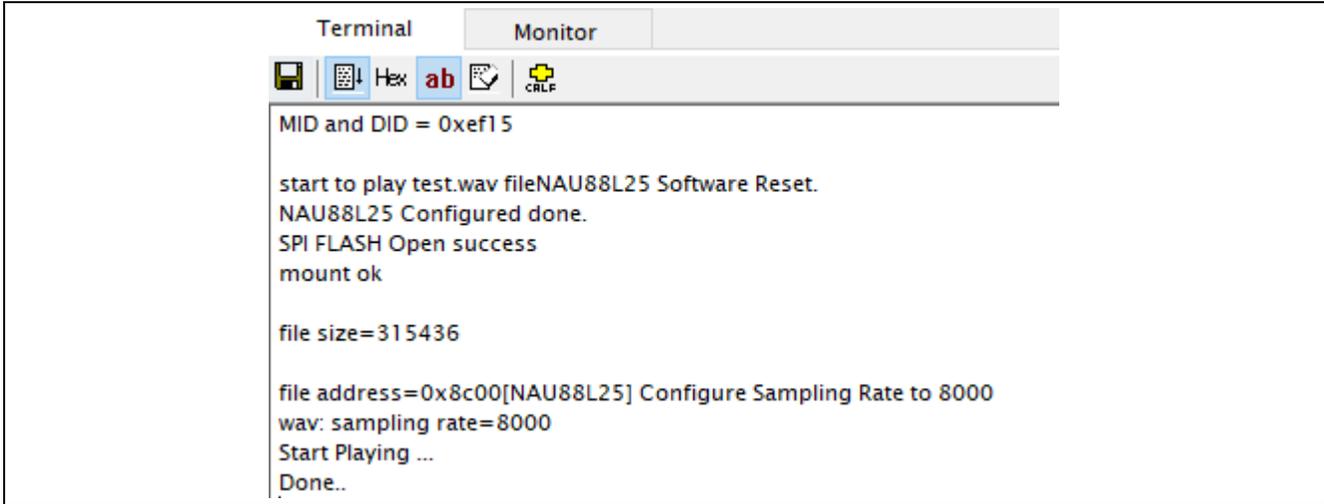
1 Overview

This sample code uses external codec NAU88L25 to play WAV file stored in external SPI Flash W25Q32. First, user needs to store the WAV file into SPI Flash first with file system. Then, the user can run the USB_Mass_Storage_QSPI_Flash sample code first and put the WAV file into QSPI Flash.

The Play WAV file supports mono/stereo mode. The data length is 16 bits and sampling rate is 8 kHz.

1.1 Demo Result

The demo result is shown below.



```
Terminal Monitor
MID and DID = 0xef15
start to play test.wav fileNAU88L25 Software Reset.
NAU88L25 Configured done.
SPI FLASH Open success
mount ok
file size=315436
file address=0x8c00[NAU88L25] Configure Sampling Rate to 8000
wav: sampling rate=8000
Start Playing ...
Done..
```

Figure 1-1 Demo Result

2 Code Description

First, initialize the M031 in main.c, initial I²C, UART, SPI-I2S, QSPI and NAU88L25. Then, call play wave function.

```
int32_t main(void)
{
    uint32_t MidDid;
    TCHAR sd_path[] = { '0', ':', 0 };    /* SD drive started from 0 */
    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();

    /* Init UART to 115200-8n1 for print message */
    UART_Open(UART0, 115200);
    QSpiInit();
    MidDid = QSpiReadMidDid();
    printf("\nMID and DID = 0x%x\n\r", MidDid);
    printf("\nstart to play test.wav file");
    f_chdrive(sd_path);    /* set default path */
    /* Init I2C1 to access NAU8822 */
    I2C1_Init();

    /* Reset NAU88L25 codec */
    NAU88L25_Reset();

    /* Set PE.13 low to enable phone jack on NuMaker board. */
    SYS->GPE_MFPH &= ~(SYS_GPE_MFPH_PE13MFP_Msk);
    GPIO_SetMode(PE, BIT13, GPIO_MODE_OUTPUT);
    PE13 = 0;

    /* Initialize NAU88L25 codec */
    CLK_SysTickDelay(20000);
    NAU88L25_Setup();

    /* Configure PDMA and use Scatter-Gather mode */
    PDMA_Init();

    WAVPlayer();
    while(1);
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
 - ◆ M031 Series BSP CMSIS v3.00.001
- IDE version
 - ◆ Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
 - ◆ NuMaker-M032KG Ver1.0
 - ◆ SPI FLASH W25Q32
 - ◆ Audio codec NAU88L25

- Pin Connect

	V _{DD}	↔	V _{DD}
	GND	↔	GND
	PA.5	↔	WP
	PA.4	↔	HOLD
	PA.3	↔	SPI_FLASH_SS
	PA.2	↔	SPI_FLASH_CLOCK
	PA.0	↔	SPI_FLASH_SI
	PA.1	↔	SPI_FLASH_SO
QSPI			SPI Flash
	V _{DD}	↔	V _{DD}
	GND	↔	GND
	PB0	↔	I2C_DAT
	PB1	↔	I2C_CLK
	PF10	↔	I2S_MCLK
	PF9	↔	I2S_LRCLK
	PF8	↔	I2S_BCLK
	PF7	↔	I2S_DO
	PF6	↔	I2S_DI
NuMaker-M032KG			NAU88L25

Figure 3-1 Pin Connect

4 Directory Information

The directory structure is shown below.

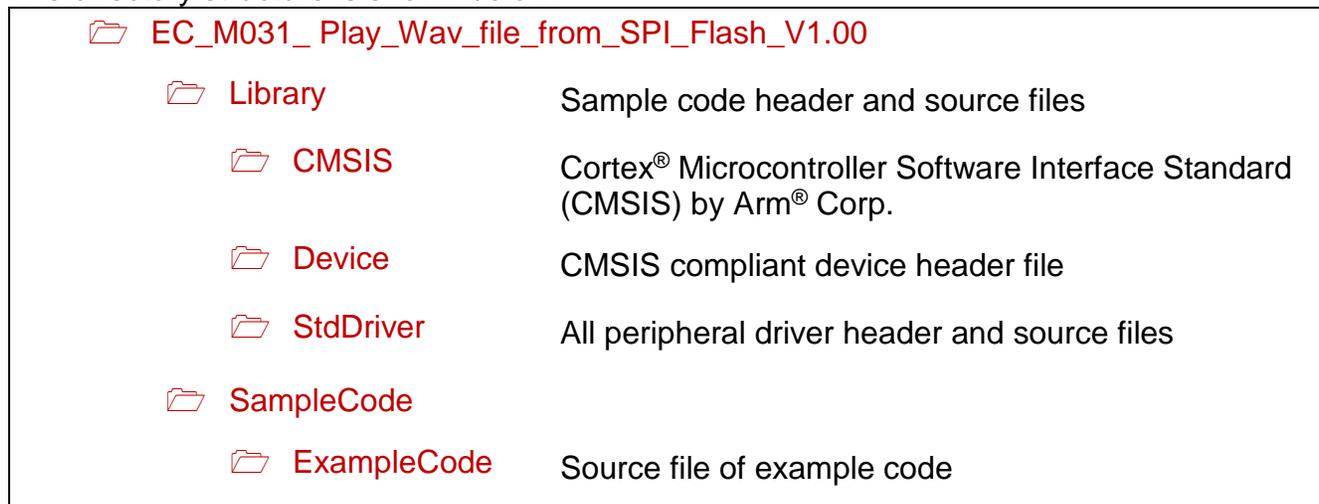


Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *I2S_PDMA_NAU88L25_QSPI_FATFS.uvproj*.
2. Enter Keil compile mode.
 - Build
 - Download
 - Start/Stop debug session
3. Enter debug mode.
 - Run

6 Revision History

Date	Revision	Description
2021.05.18	1.00	1. Initially issued.

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.*