NUC505從內部SPI flash撥放WAV檔

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例代碼從NUC505內部SPI flash讀取WAV檔案撥放
BSP 版本	NUC505 Series BSP CMSIS V3.02.000
開發平台	NuTiny-EVB-NUC505 V1.6

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

1 功能介紹

1.1 簡介

範例程式中，NUC505使用內部音訊解碼器，讀取存在內部SPI flash的WAV檔並且撥放。使用者需先燒錄WAV檔至內部SPI flash，並從config.h檔中的WAVFile_address去指定WAV檔儲存位置。請注意WAV檔案不可超過NUC505記憶體位置。程式會先判斷WAV檔頭起始位置是否為RIFF，以及第8位元組是否為WAVE，來驗證儲存是否為WAV檔。

WAV檔案存放地址範圍	播放WAV檔案位置參數
0x2000~0x1F_FFFF	WAVFile_address

NUC505支援單/雙聲道，資料長度為16 bits格式，取樣率為8k, 11025, 12k, 16k, 22050, 24k, 32k, 44100和48k，程式碼會自動依照不同取樣率設定APLL數值。

1.2 執行結果

執行後會打印出以下資訊

```
Relocate vector table in SRAM (0x20000000) for fast interrupt handling, size = 0xbc.
+------------------------------------------------------------+
|      Play WAV file from internal Flash Sample Code         |
+------------------------------------------------------------+
  Please store WAV file first in the internal Flash.
  WAV file storage location cannot exceed flash size
  NOTE: Need head-phone.

wav.bin pre-stored in SPI Flash address from 0x00002000
wav: sampling rate = 44100
wav: Data size = 864 KB
wav: Channel Num = 2

Configure Internal CODEC ... [OK]
Start Playing ...
Play finish... enter while 1
```

2 代碼介紹

首先在main.c中對NUC505做初始化設定，並且把程式碼放入SRAM中執行以增加其執行速度，根據WAVFile_address所設定位置讀取WAV檔撥放。

```
int32_t main(void)
{
    /* Init System, IP clock and multi-function I/O */
    SYS_Init();

    /* Init UART0 to 115200-8n1 for print message */
    UART0_Init();

    /* Relocate vector table in SRAM for fast interrupt handling. */
    {
        extern uint32_t __Vectors[];
        extern uint32_t __Vectors_Size[];
        extern uint32_t Image$$ER_VECTOR2$$ZI$$Base[];

        printf("Relocate vector table in SRAM (0x%08X) for fast interrupt handling, size =
0x%x.\n", Image$$ER_VECTOR2$$ZI$$Base, (unsigned int) __Vectors_Size);
        memcpy((void *) Image$$ER_VECTOR2$$ZI$$Base, (void *) __Vectors, (unsigned int)
__Vectors_Size);
        SCB->VTOR = (uint32_t) Image$$ER_VECTOR2$$ZI$$Base;
    }
    /* Init I2S */
    I2S_Init();

    printf("+------------------------------------------------------------+\n");
    printf("|      Play WAV file from internal Flash Sample Code         |\n");
    printf("+------------------------------------------------------------+\n");
    printf("  Please store WAV file first in the internal Flash.\n");
    printf("  WAV file storage location cannot exceed flash size\n");
    printf("  NOTE: Need head-phone. \n\n");

    /* Play wav from internal flash at WAVFile_address */
    ReadFlashPlay(WAVFile_address);

    while (1);
}
```

在ReadFlashPlay(uint32_t FlashAddr)函式中，程式會先判斷WAV檔頭起始位置是否為RIFF，以及第8位元組是否為WAVE，來驗證儲存是否為WAV檔，並且讀取WAV檔相關資訊，並根據取樣率對APLL做設定。

```
    /* Parsing WAV file header */
    SPIFlash_DMARead(SPIM, FlashAddr, 0, 44, aWavHeader, SPIM_CTL0_CMDCODE_FAST_READ);

    /* Read sampling rate from WAV header */
    u32WavSamplingRate = aWavHeader[6];

    /* Check if the WAV file is correct */
    if (aWavHeader[0] != 0x46464952 | aWavHeader[2] != 0x45564157) {
        printf("WAV File ERROR, please store Wav file at 0x%08x\n", FlashAddr);
        return -1;
    }

    /* Read WAV file total size from WAV header */
    PlayFrameSize = aWavHeader[1] / (PCM_BUFFER_SIZE * 4);
    WavRealSize = aWavHeader[1] / 1024;

    /* Read Channel number from WAV header */
    ChannelNum = (aWavHeader[5] & 0x00030000) >> 16;
    AudioFormat = ChannelNum == 1 ? I2S_MONO : I2S_STEREO;
    /* Check if the MP3 SamplRate frequency is 11025 multiples */
    if (u32WavSamplingRate % 11025) {
        /* APLL is 49152031Hz for 12k, 16k, 24k, 32k, 48k, and 96k sampling rate */
        CLK_SET_APLL(CLK_APLL_49152031);

        if (u32WavSamplingRate == 8000)
            CLK_SetModuleClock(I2S_MODULE, CLK_I2S_SRC_APLL, 1);
    }
```

設定完畢之後使用Ping-pong buffer機制撥放WAV檔，其流程如下：

```
    while (1) {
        /* Set ping-pong buffer to transmit data */
        if ((aPCMBuffer_Full[0] == 1) && (aPCMBuffer_Full[1] == 1)) {
            /* In beginning, wait all buffers full and start play */
            if (!bAudioPlaying) {
                bAudioPlaying = 1;
                /* I2S DMA settings */
                I2S_SET_TX_TH_LEVEL(I2S, I2S_FIFO_TX_LEVEL_WORD_15);
```

```
                I2S_SET_RX_TH_LEVEL(I2S, I2S_FIFO_RX_LEVEL_WORD_16);
                /* Tx Start Address */
                I2S_SET_TXDMA_STADDR(I2S, (uint32_t) &aPCMBuffer[0][0]);
                /* Tx Threshold Address */
                I2S_SET_TXDMA_THADDR(I2S, (uint32_t) &aPCMBuffer[0][PCM_BUFFER_SIZE - 1]);
                /* Tx End Address */
                I2S_SET_TXDMA_EADDR(I2S, (uint32_t) &aPCMBuffer[1][PCM_BUFFER_SIZE - 1]);
                /* Enable setting -> start to play */
                I2S_ENABLE_TXDMA(I2S);
                I2S_ENABLE_TX(I2S);
                /* Enable interrupt : TX DMA Threshold and TX DMA End Interrupt */
                I2S_EnableInt(I2S, (I2S_IEN_TDMATIEN_Msk | I2S_IEN_TDMAEIEN_Msk));
                NVIC_EnableIRQ(I2S_IRQn);
                printf("Start Playing ...\n");
            }
            /* All buffers are full, wait */
            while ((aPCMBuffer_Full[0] == 1) && (aPCMBuffer_Full[1] == 1));
        }
        /* Read WAV data from internal flash */
        SPIFlash_DMARead(SPIM, FlashAddr + 44, 0, PCM_BUFFER_SIZE * 4,
_UNVMP(&aPCMBuffer[u8PCMBufferTargetIdx][0]), SPIM_CTL0_CMDCODE_FAST_READ);
        /* Move address pointer to next address */
        FlashAddr = FlashAddr + PCM_BUFFER_SIZE * 4;
        count_frame++;
        NVIC_DisableIRQ(I2S_IRQn);
        /* Set buffer full flag */
        aPCMBuffer_Full[u8PCMBufferTargetIdx] = 1;
        NVIC_EnableIRQ(I2S_IRQn);

        if (bAudioPlaying) {
            /* Make sure the current ping-pong buffer is empty */
            while (aPCMBuffer_Full[u8PCMBufferTargetIdx ^ 1]) {
                /* Break if the last audio data played */
                if (count_frame >= PlayFrameSize) {
                    goto stop;
                }
            }
        }
        /* Switch to another buffer */
        u8PCMBufferTargetIdx ^= 1;
    }
```

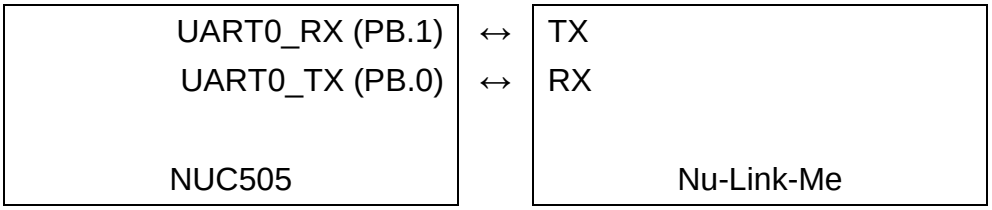
在nuc505_isr.c，當資料傳送完畢時清除緩衝區(buffer)滿溢旗標

```
void I2S_IRQHandler(void)
{
    uint32_t u32I2SIntFlag;

    /* Check interruput flag */
    u32I2SIntFlag = I2S_GET_INT_FLAG(I2S, I2S_STATUS_TDMATIF_Msk |
I2S_STATUS_TDMAEIF_Msk);

    if (u32I2SIntFlag & I2S_STATUS_TDMATIF_Msk) {
        I2S_CLR_INT_FLAG(I2S, I2S_STATUS_TDMATIF_Msk);

        /* Buffer 0 transfer completed, clear buffer full flag */
        aPCMBuffer_Full[0] = 0;
    } else if (u32I2SIntFlag & I2S_STATUS_TDMAEIF_Msk) {
        I2S_CLR_INT_FLAG(I2S, I2S_STATUS_TDMAEIF_Msk);
        /* Buffer 1 transfer completed, clear buffer full flag */
        aPCMBuffer_Full[1] = 0;
    }
}
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC505 Series BSP CMSIS V3.02.000
  - IDE 版本
    - Keil uVersion 5.26
- **硬體環境**
  - 電路元件
    - NuTiny-EVB-NUC505 V1.6
    - 3.5 mm 接頭耳機
  - 示意圖

使用 3.5 mm 接頭耳機連接板上的 HP OUT 底座，並透過 NUC505 的 UART0_RX(PB.1)、UART0_TX(PB.0)連接至 Nu-Link Me，打印訊息。設置終端機的 COM Port 與 Baud，COM Port 的編號可在裝置管理員中找到「NuBridge Virtual Com Port (COMX)」，Baud 設置為 115200。



4 目錄資訊

- EC_NUC505_WavPlayerSPIFlash_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 WavPlayerSPIFlash.uvproj
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Jul. 03, 2019	1.00	1. 初始發布.

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.