# M480暂停与回复MP3播放器

NuMicro® 32位系列微控制器范例代码介绍

## 文件信息

| | |
|---|---|
| 应用简述 | 本范例代码使用 M480 外部音频编解码器 NAU88L25 播放在 SD 卡里 MP3 格式的档案，此 MP3 播放器提供播放、停止、暂停、恢复功能。 |
| BSP 版本 | M480 Series BSP CMSIS V3.05.001 |
| 开发平台 | NuMaker-PFM-M487 v3.0 |

*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. 概述

本范例代码使用 M480 外部音频编解码器 NAU88L25 拨放在 SD 卡里 MP3 格式的档案，此 MP3 播放器提供播放、停止、暂停、恢复功能。

首先使用者需将 MP3 格式的档案放进 SD 卡里，并将修改檔名为 test.mp3 或是修改 mp3.c 中的 #define MP3_FILE 至正确的名称。此范例程序将会自动读取 SD 卡里的 MP3 档案、使用 NAU88L25 音频编解码器、并且自动透过 I²S 接口播放音乐。

## 1.1 执行结果

本范例代码执行暂停与恢复音乐的功能如下图 1-1 所示，使用者可以按压 SW3 按钮暂停音乐，再次按压 SW3 按钮则可以恢复音乐到暂停前。

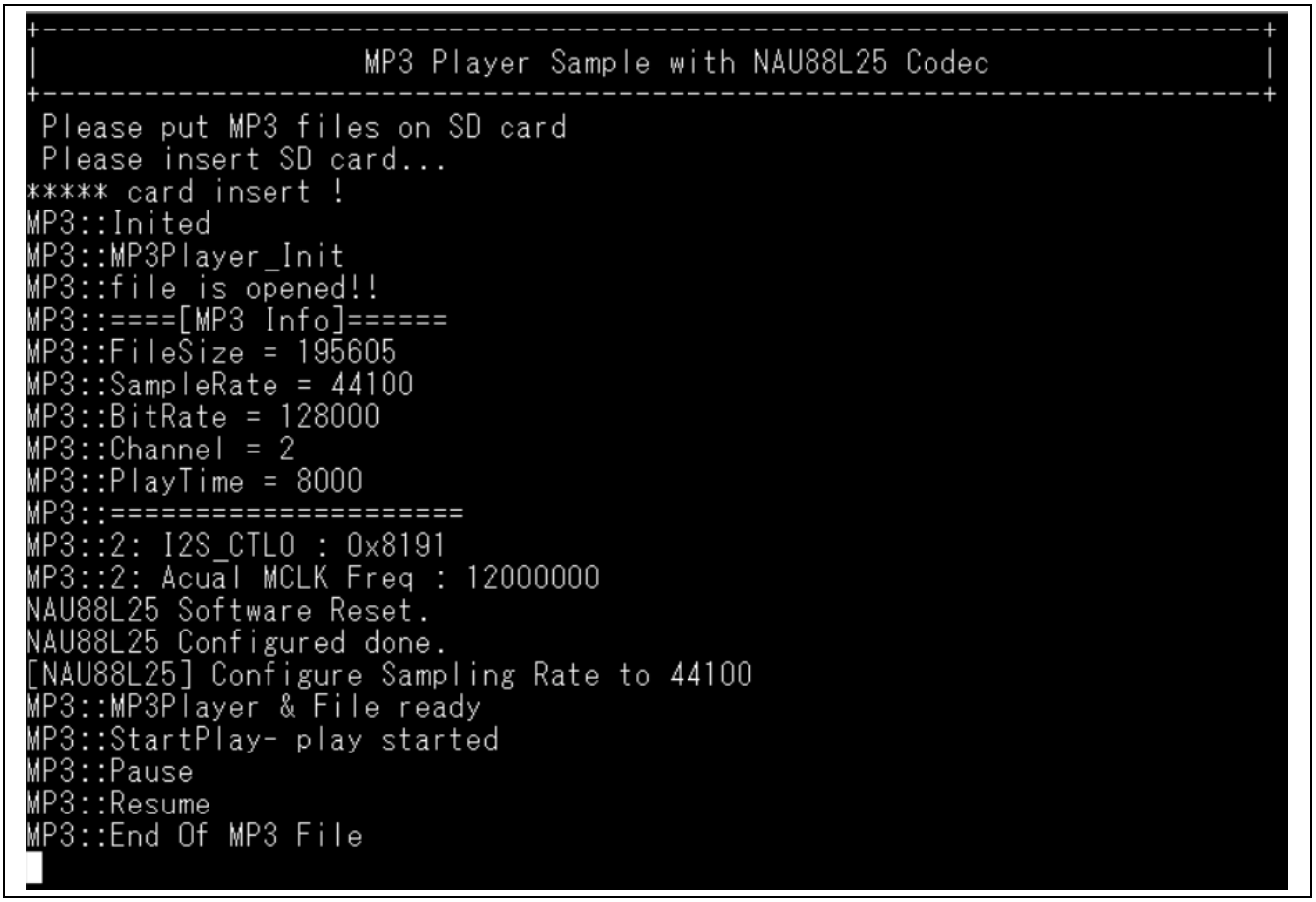


图 1-1 暂停与恢复音乐展示结果

第二个展示停止与重新播放音乐结果如下图 1-2 所示，使用者可以按压 SW2 按钮停止播放音乐，再次按压 SW2 按钮则可以重新播放音乐。

```
+-------------------------------------------------------------------------+
|                    MP3 Player Sample with NAU88L25 Codec                |
+-------------------------------------------------------------------------+
 Please put MP3 files on SD card
 Please insert SD card...
***** card insert !
MP3::Inited
MP3::MP3Player_Init
MP3::file is opened!!
MP3::====[MP3 Info]======
MP3::FileSize = 195605
MP3::SampleRate = 44100
MP3::BitRate = 128000
MP3::Channel = 2
MP3::PlayTime = 8000
MP3::====================
MP3::2: I2S_CTL0 : 0x8191
MP3::2: Acual MCLK Freq : 12000000
NAU88L25 Software Reset.
NAU88L25 Configured done.
[NAU88L25] Configure Sampling Rate to 44100
MP3::MP3Player & File ready
MP3::StartPlay- play started
MP3::Stop
MP3::Start
MP3::Inited
MP3::MP3Player_Init
MP3::file is opened!!
MP3::====[MP3 Info]======
MP3::FileSize = 195605
MP3::SampleRate = 44100
MP3::BitRate = 128000
MP3::Channel = 2
MP3::PlayTime = 8000
MP3::====================
MP3::2: I2S_CTL0 : 0x8191
MP3::2: Acual MCLK Freq : 12000000
NAU88L25 Software Reset.
NAU88L25 Configured done.
[NAU88L25] Configure Sampling Rate to 44100
MP3::MP3Player & File ready
```

图 1-2 停止与重新播放音乐展示结果

## 2. 代码介绍

*main.c* 会初始化 I²C、SD、GPIO 接口与 NAU88L25，后续将会调用 *mp3.c* 来播放音乐，直到音乐停止。

```
int32_t main (void)
{
    TCHAR    sd_path[] = { '0', ':', 0 };    /* SD drive started from 0 */

    MP3RESULT u8MainMP3Result = MP3_OK;

    /* Init System, IP clock and multi-function I/O */
    SYS_Init();

    /* Configure PG.15 as Quasi-bidirection mode and enable interrupt by falling edge
trigger */
    GPIO_SetMode(PG, BIT15, GPIO_MODE_QUASI);
    GPIO_EnableInt(PG, 15, GPIO_INT_FALLING);
    NVIC_EnableIRQ(GPG_IRQn);

    /* Enable interrupt de-bounce function and select de-bounce sampling cycle time is
1024 clocks of LIRC clock */
    GPIO_SET_DEBOUNCE_TIME(GPIO_DBCTL_DBCLKSRC_LIRC, GPIO_DBCTL_DBCLKSEL_1024);
    GPIO_ENABLE_DEBOUNCE(PG, BIT15);

    /* Configure PF.11 as Quasi-bidirection mode and enable interrupt by falling edge
trigger */
    GPIO_SetMode(PF, BIT11, GPIO_MODE_QUASI);
    GPIO_EnableInt(PF, 11, GPIO_INT_FALLING);
    NVIC_EnableIRQ(GPF_IRQn);

    /* Enable interrupt de-bounce function and select de-bounce sampling cycle time is
1024 clocks of LIRC clock */
    GPIO_SET_DEBOUNCE_TIME(GPIO_DBCTL_DBCLKSRC_LIRC, GPIO_DBCTL_DBCLKSEL_1024);
    GPIO_ENABLE_DEBOUNCE(PF, BIT11);

    /* BOD Level Set */
    /* Unlock protected registers before setting Brown-out detector function and Power-
down mode */
    SYS_UnlockReg();

    /* Enable Brown-out detector function */
    SYS_ENABLE_BOD();

    /* Set Brown-out detector voltage level as 2.4V */
    SYS_SET_BOD_LEVEL(SYS_BODCTL_BODVL_2_4V);

    /* Enable Brown-out detector interrupt function */
    SYS_DISABLE_BOD_RST();

    /* Enable Brown-out detector and Power-down wake-up interrupt */
    NVIC_EnableIRQ(BOD_IRQn);
```

```
    /* Init SD */
    SD_Inits();

    /* Init UART to 115200-8n1 for print message */
    UART_Open(UART0, 115200);

    printf("+-----------------------------------------------------------------------
+\n");
    printf("|                    MP3 Player Sample with NAU88L25 Codec
|\n");
    printf("+-----------------------------------------------------------------------
+\n");
    printf(" Please put MP3 files on SD card \n");

    printf(" Please insert SD card... \n");
    while(1)
    {
        if (SDH_Open_Disk(SDH0, CardDetect_From_GPIO) == Successful)
            break;
    }
/* set default path */
f_chdrive(sd_path);

    /* Init I2C2 to access WAU88L25 */
    I2C2_Init();

    /* select source from HXT(12MHz) */
    CLK_SetModuleClock(I2S0_MODULE, CLK_CLKSEL3_I2S0SEL_HXT, 0);

    SetMP3PlayerStatus(eMP3PLAYER_STATUS_START);

    while(1)
    {
        /* play mp3 */
        if (SD0.IsCardInsert == TRUE)
        {
            u8MainMP3Result = TASK_MP3Player();
            if(u8MainMP3Result > MP3_MAD_BUFLEN_ERR)
            {
                printf("MP3 Play Error code : %d\n\r", u8MainMP3Result);
                printf("Exit MP3 Player\n\r");
                break;
            }
        }
    }
}
```

*mp3.c* 将会初始化 I²S 与显示读取到的 MP3 档案信息，*TASK_MP3Player* 函数将会读取并且设定 MP3 播放器的状态标志，并且执行相对应的函数。

```
MP3RESULT TASK_MP3Player()
{
    MP3RESULT mp3Result = MP3_OK;
    /* Set MP3 Player Status by Button and Reset Flag */
    switch (Status_Flag)
```

```
    {
    case eMP3PLAYER_STATUS_RESUME:
        SetMP3PlayerStatus(eMP3PLAYER_STATUS_RESUME);
        Status_Flag = eMP3PLAYER_STATUS_NONE;
        break;
    case eMP3PLAYER_STATUS_PAUSE:
        SetMP3PlayerStatus(eMP3PLAYER_STATUS_PAUSE);
        Status_Flag = eMP3PLAYER_STATUS_NONE;
        break;
    case eMP3PLAYER_STATUS_STOP:
        SetMP3PlayerStatus(eMP3PLAYER_STATUS_STOP);
        Status_Flag = eMP3PLAYER_STATUS_NONE;
        break;
    case eMP3PLAYER_STATUS_START:
        SetMP3PlayerStatus(eMP3PLAYER_STATUS_START);
        Status_Flag = eMP3PLAYER_STATUS_NONE;
        break;
    default:
        break;
    }
    newplayerstatus = u8MP3PlayerStatus;
    if(currentplayerstatus != newplayerstatus)
    {
        currentplayerstatus = newplayerstatus;
        switch(newplayerstatus)
        {
        case eMP3PLAYER_STATUS_START :
            NAU88L25_Unmute();
            MP3Player_DeviceInit();
            mp3Result = MP3Player_DataInit();
            printf("MP3::MP3Player & File ready\n\r");
            break;
        case eMP3PLAYER_STATUS_STOP:
            NAU88L25_Mute();
            MP3Player_PlayStop();
            break;

        default:
            break;
        }
    }
    if(eMP3PLAYER_STATUS_STOP != currentplayerstatus)
        mp3Result = MP3Player_PlayStart(newplayerstatus);
}
```

# 3. 软件与硬件需求

## 3.1 软件需求

- BSP 版本
  - M480 Series BSP CMSIS V3.05.001
- IDE 版本
  - Keil uVersion 5.26

## 3.2 硬件需求

- 电路组件
  - NuMaker-PFM-M487 v3.0
  - SD 卡
- 示意图
  - NuMaker-PFM-M487 与 SD 卡、NAU88L25 连接示意图

| | | | | |
|---|---|---|---|---|
| | $V_{DD}$ | ↔ | $V_{DD}$ | |
| | GND | ↔ | GND | |
| | PE.7 | ↔ | SD_CMD | |
| | PE.6 | ↔ | SD_CLK | |
| | PE.5 | ↔ | SD_DATA3 | |
| | PE.4 | ↔ | SD_DATA2 | |
| | PE.3 | ↔ | SD_DATA1 | |
| | PE.2 | ↔ | SD_DATA0 | |
| | PD.13 | ↔ | SD_nCD | |
| | NuMaker-PFM-M487 | | SD CARD | |
| | $V_{DD}$ | ↔ | $V_{DD}$ | |
| | GND | ↔ | GND | |
| | PD.0 | ↔ | I2C_SCL | |

| | | | | |
|---|---|---|---|---|
| | PD.1 | ↔ | I2C_SSA | |
| | PF.10 | ↔ | I2S_BCLK | |
| | PF.9 | ↔ | I2S_MCLK | |
| | PF8 | ↔ | I2S_DI | |
| | PF7 | ↔ | I2S_DO | |
| | PF6 | ↔ | I2S_LRCK | |
| | NuMaker-PFM-M487 | | NAU88L25 | |

图 3- 1 脚位连接示意图

# 4. 目录信息

| | |
|---|---|
| EC_M480_Pause_and Resume_MP3_Player_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 |
| ThirdParty | |
| FatFs | A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html. |
| LibMAD | A MPEG audio decoder library which 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. This library is distributed under GPL license. Please contact Underbit Technologies (http://www.underbit.com/) for the commercial license |

图 4-1 目录信息

# 5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *Pause_and_Resume_MP3_Player.uvproj*。
2. 进入编译模式接口
   - 编译
   - 下载代码至内存
   - 进入 / 离开除错模式
3. 进入除错模式接口
   - 执行代码

# 6. 修订纪录

| Date | Revision | Description |
|---|---|---|
| 2022.06.21 | 1.00 | 初始发布。 |

## 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.*