Using M487 to Play AMR File from MicroSD Card

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code uses the M487 microcontroller and external audio Codec NAU88L25 to decode and play an AMR audio file in a MicroSD card.
BSP Version	M480_BSP_CMSIS_V3.05.003
Hardware	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 Overview

This example code uses the M487 microcontroller and external audio Codec NAU88L25 to decode and play an AMR audio file stored in a MicroSD card. Before running the example code, user needs to store the AMR file into a MicroSD card and insert the MicroSD card into the card slot on the NuMaker-PFM-M487 board.

Note: An AMR file “audio.amr” has been stored in the following file path. During the test, the user can copy it to the MicroSD card.

EC_M487_AMRPlayer_SD_V1.00\SampleCode\ExampleCode\M487_AMRPlayer_SD

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

1.1 Demo Result

The demo result is shown below.

```
+-------------------------------------------------------------------------+
|                   AMR Player Example with NAU88L25 Codec                |
+-------------------------------------------------------------------------+
 Please put AMR files on SD card
 Please insert SD card...
***** card insert !
====[AMR Info]======
FileSize 282150 bytes
AMR Mode 7
PacketSize 32 bytes
PlayTime 176 s
====================
NAU88L25 Software Reset.
NAU88L25 Configured done.
[NAU88L25] Configure Sampling Rate to 8000
Start AMR playing ...
Start Playing ...
Stop ...
```

Figure 1-1 Demo Result

2 Code Description

Because the AMR Codec library allocates memory dynamically, it has to reserve more memory for heap. The project file has been set as 8 Kbytes for heap. If you create a new project, remember to add the following setting to your project.

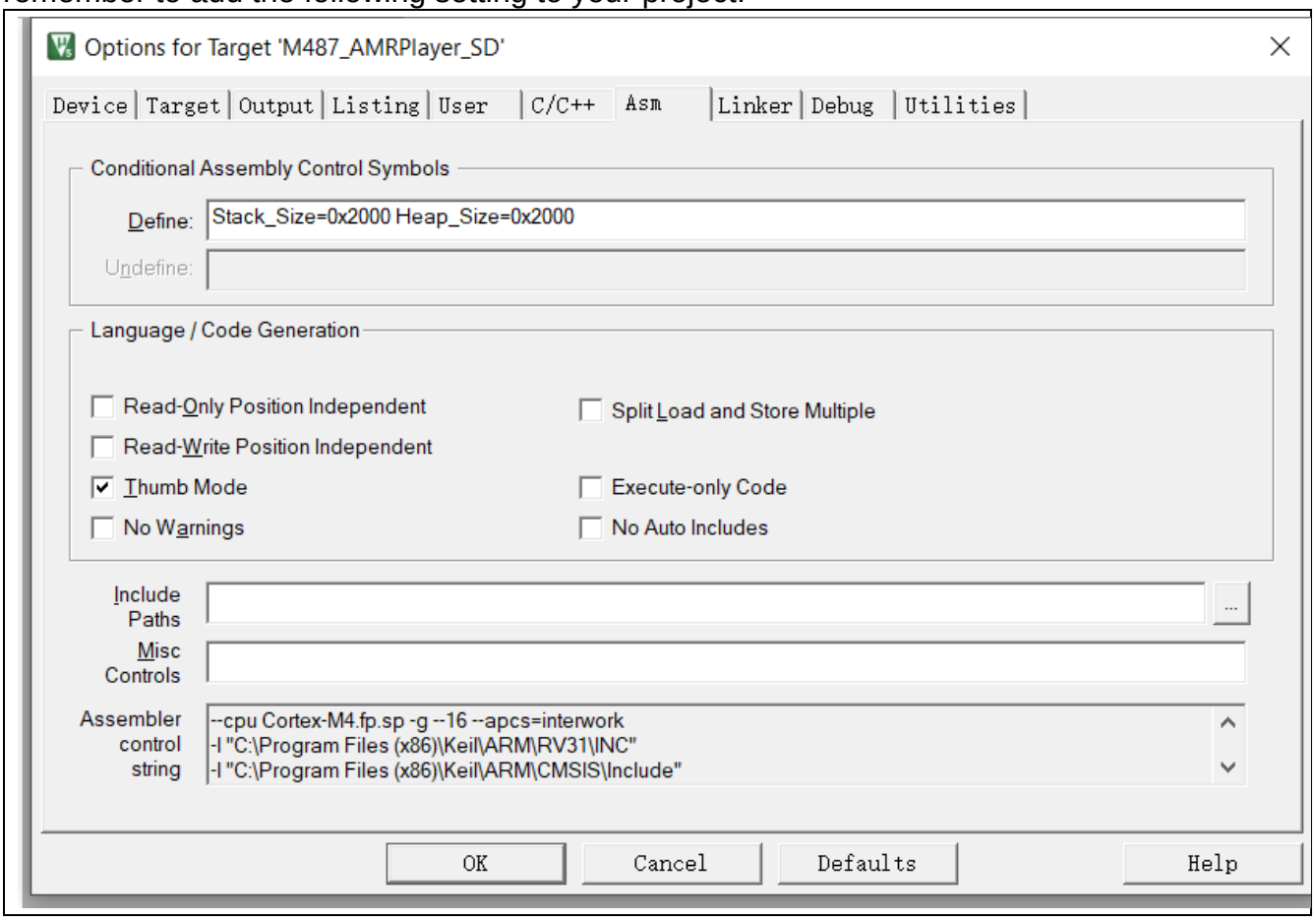

Figure 2-1 Heap Size Settings

Initialize the M480 in main.c, and initialize I²C, UART, and SDH. Then, call AMRPlayer function.

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

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

    /* Init I2C to access WAU88L25 */
    I2C_Init();

    /* Init SD */
    SD_Inits();

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

    printf("+-----------------------------------------------------------------------+\n");
    printf("|                AMR Player Example with NAU88L25 Codec                 |\n");
    printf("+------------ ----------------------------------------------------------+\n");
    printf(" Please put AMR 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);

    /* Set PE.13 low to enable phone jack on NuMaker board. */
    GPIO_SetMode(PE, BIT13, GPIO_MODE_OUTPUT);
    PE13 = 0;
    AMRPlayer();
    while (1);
}
```

Decode the AMR file and duplicate audio data to two channels.

```
if (u8StreamPending == FALSE)
{
    if (!f_eof(&amrFileObject))
    {
        f_read(&amrFileObject, u8amr_buf, packet_size, &ReturnSize);

        if (ReturnSize != packet_size)
        {
            // should not happen. clean the packet.
            printf("ERROR: AMR packet error. Ignore it.\n");
            memset(u8amr_buf, 0, packet_size);
        }

        /* Decode AMR file */
        amrDecode(u8amr_buf, (short *)DecodeBuffer);

        u8StreamPending = TRUE;
    }
    else
    {
        break;
    }
}
else
{
    while (g_u8PCMBuffer_Full[u8PCMBufferTargetIdx]);

    /* Duplicate audio data to two channels */
    p = (u8PCMBufferTargetIdx) ? (g_u16PCMBuffer[1]) : (g_u16PCMBuffer[0]);

    for (i = 0; i < AMR_FRAME_SIZE; i++)
    {
        *p++ = DecodeBuffer[i];
        *p++ = DecodeBuffer[i];
    }

    NVIC_DisableIRQ(I2S0_IRQn);
    g_u8PCMBuffer_Full[u8PCMBufferTargetIdx] = 1;
    NVIC_EnableIRQ(I2S0_IRQn);

    u8PCMBufferTargetIdx ^= 1;

    u8StreamPending = FALSE;
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M480_BSP_CMSIS_V3.05.003
- IDE version
  - Keil uVersion 5.36

3.2 Hardware Requirements

- Circuit components
  - NuMaker-PFM-M487 V3.0
  - 3.5mm headphones
  - A MicroSD card used for file storage
- Pin Connect

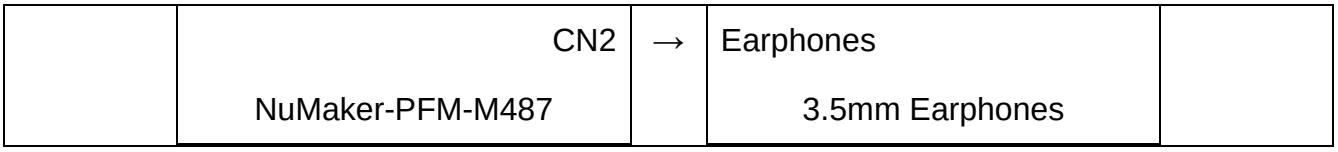

Figure 3-1 M487 and Earphone Connection Diagram

4 Directory Information

EC_M487_AMRPlayer_SD_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.
LibAMR	The AMR Codec library supports Narrow Band of AMR encode and decode.

Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the ExampleCode folder as described in the Directory Information section and double-click *M487_AMRPlayer_SD.uvproj*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2022.03.15	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.