**Freescale Semiconductor**
Application Note

Document Number: AN4523
Rev. 0, 01/2014

# Audio player on KwikStik

*by: Michael Galda*

*Freescale Semiconductor, Inc.*

**Contents**

## 1 Introduction

This demo application describes an implementation of the simple Audio player on the Kinetis ARM Cortex™ -M4 based MCU.

The audio files are stored on the SD card. Supported are .wav files containing uncompressed WAV audio samples in LPCM format.

WAV format has a simple structure and is well known. When compressed audio support is needed, an audio software codec can be easily imported to the existing software application, for instance to create a true MP3/WMA player.

This shows an excellent solution for those applications using message playback or voice recording. The Demo software is managed by the Freescale MQX Real-Time Operating System (RTOS). Bare-metal drivers are used for direct access to MCU peripherals like timers and DAC.

The hardware of the demo is based on the Freescale KwikStik (version 5), an ultra low-cost, all-in-one development tool for evaluating, developing and debugging Kinetis MCUs. It features the K40X256VLQ100 (144LQFP) MCU with USB, an On-Board J-Link USB

Programmer, Capacitive Touch Sensing, Segment LCD, Microphone, 3.5 mm Audio Output Jack, Micro SD Card slot functionality, etc.

The solution may be reused for the following applications:

- Voice recorders
- Answering Machines
- Smart Medical Appliances
- Safety systems
- Electronic toys

## 2 Player Demo Overview

The Demo software reads the .wav files stored on SD card, using the Freescale MQX software stationary and MFS (MQX File System). SD-card has to be formatted to FAT file system. The default SD directory location is set to “\Songs” it can be changed in the software. The supported .wav formats are either Mono or Stereo records, tested with 22050 kHz sample rate. Audio records are stored in the raw-form as 16-bit signed integer PCM samples.

DAC (Digital to Analog Converter) is used to generate the audio output. KwikStik HW supports only one DAC out channel, so it can provide single channel audio output (Mono).

DAC output can generate very smooth signals. The DAC current output capability is very low (max. 1mA), so it has to be amplified externally. The DAC output register has to be periodically reloaded with the new sample by the timer channel IRQ. MCU FlexTimer (FTM0) peripheral generating the periodic IRQ with the period equal to audio sampling rate is used for triggering the DAC. In the simplest case the DAC output register is filled by the audio samples inside the timer IRQ routine. In order to offload the CPU, DMA (Direct Memory Access) peripheral can be used for filling the audio output by data with no CPU assistance needed.

Moreover a Double Buffering method (known as Ping-Pong buffers) is used for the smooth playback without jamming.

At any one time, one buffer is actively being played (the front buffer), while the second (background) buffer is filled with the new audio samples. When playing is completed, the roles of the two buffers are switched. This is usually accomplished by switching the pointers.

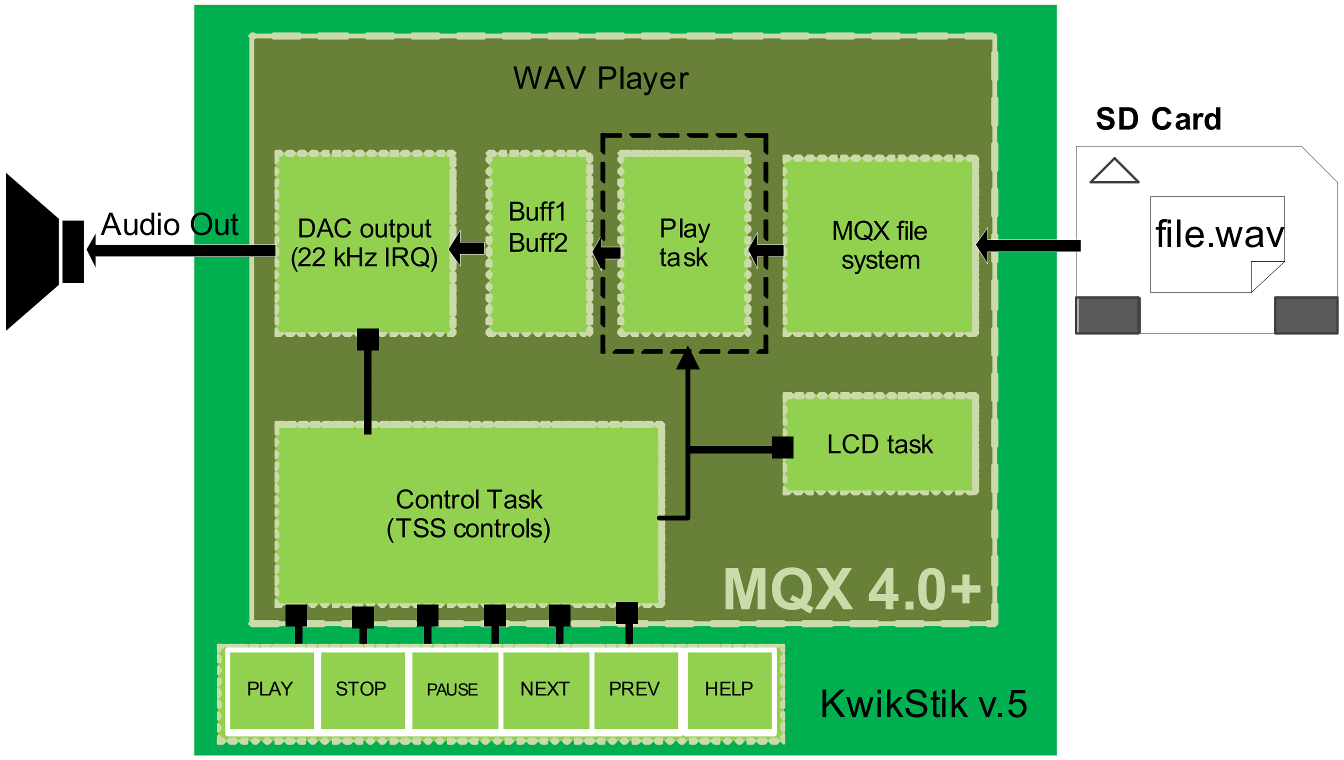


**Figure 1. Audio player block scheme**

## 3 KwikStik Player application description

The software of the application is based on the block scheme shown in Figure 2. The application is controlled by capacitive (TSS library) buttons located on both sides of the KwikStik (www.freescale.com/KwikStik). Touch buttons (E1, E3, E5) on the left side are used for the PLAY / STOP and NEXT / PREV functions and touch buttons on the other side are used for the VOLUME controls.

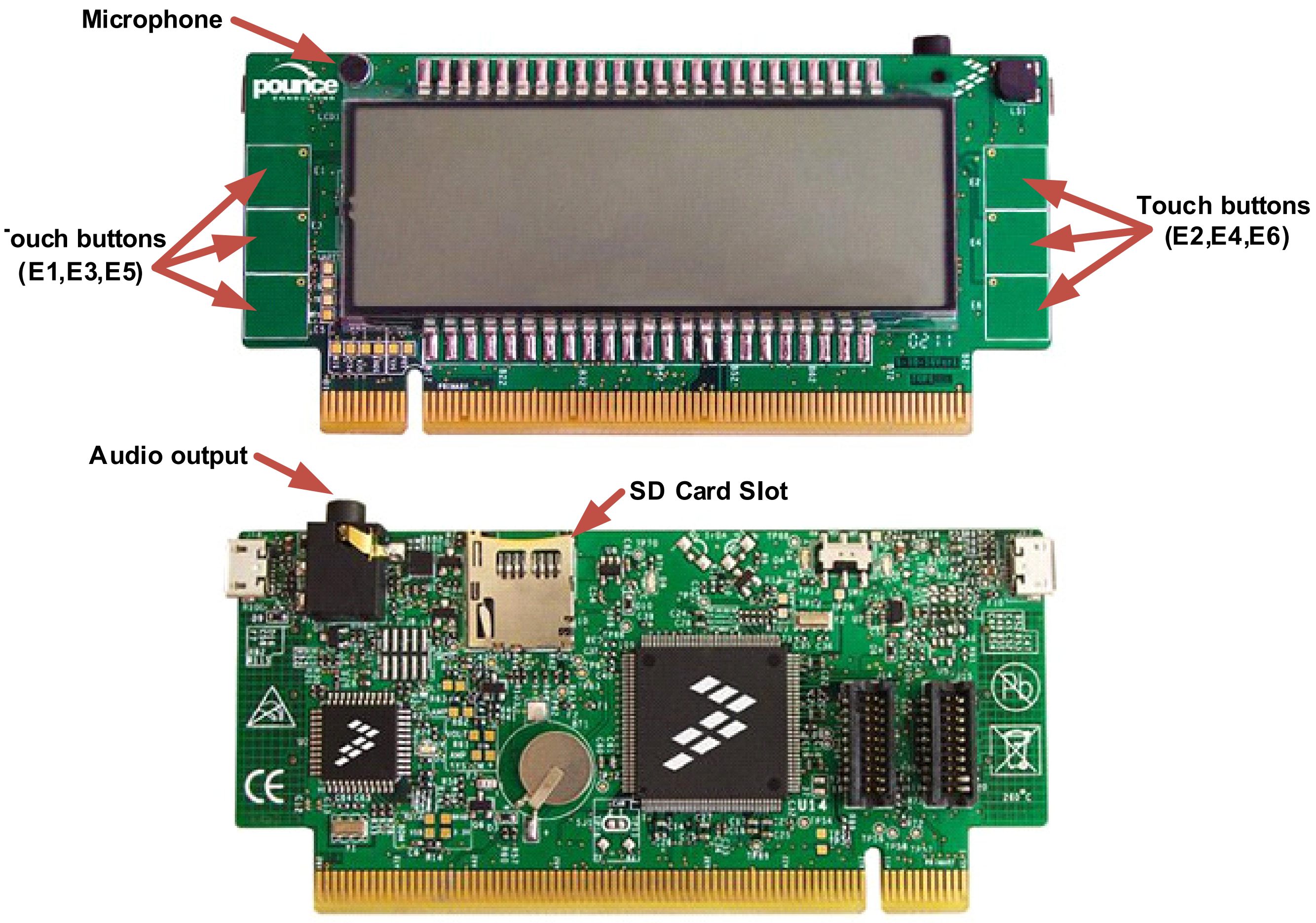


**Figure 2. KwikStik HW description**

[1] The application demo software was tested on KwikStik HW **ver.5**

[2] Due known HW issues the demo may not run properly on the older versions of HW

## 4 Playback

During message playback, the audio data is read from the SD card file to one of the audio output buffers. The samples are reproduced using the internal 12-bit DAC. Audio data paths and file IO operations are double-buffered to avoid the loss of frames. So that once the first buffer (containing the processed audio data) has been played, the application switches to the second buffer and reuses the first one for reading.

Audio output (DAC) is driven via the FTM interrupt handler which is periodically fired based on the audio sample rate (tested with 22050 kHz, stereo records).

The Freescale MQX Real-Time Operating System (RTOS) provides real-time performance within a small, configurable footprint. The easy-to-use API and out-of-box experience ensure that first-time RTOS users can start developing their applications on the day the software is installed. The MQX RTOS is targeted mainly at embedded applications supporting most of the Freescale 32-bit MCUs. By using MQX, we can run several parallel tasks on a single CPU core. Moreover, MQX includes a complete file system (MFS) and peripheral drivers, such as the SD-card driver used in this demo application.

MFS is an embedded FAT file system compatible with the Microsoft Windows and MS-DOS file systems. It can format, read, write and exchange files with any operating systems running a FAT-12, FAT-16 or FAT-32 file system. It is fully re-entrant and uses the Freescale MQX RTOS file device driver to access disk devices. See freescale.com/mqx for detailed information on MQX.

Bare-Metal software routines are used for direct access to some MCU peripherals in order to save CPU cycles and take control over the HW during the time critical tasks.

TSS: Freescale Touch Sensing software library is now fully integrated to MQX BSP package, precompiled and delivered in library "tss.lib" with the MQX installation.

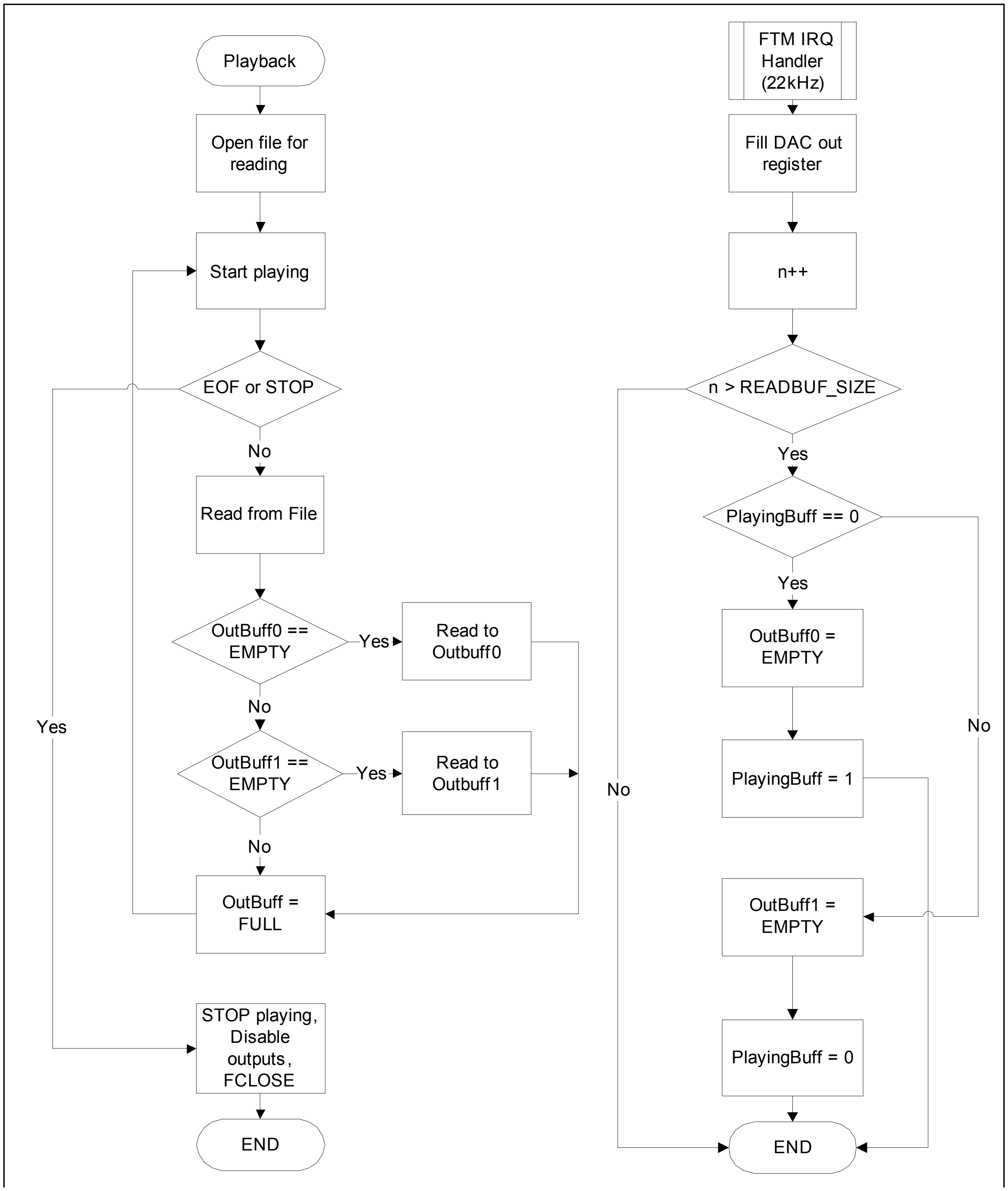


**Figure 3. Playback block diagram**

## 5 MQX BSP setup

The demo software project uses the MQX BSP and file system. Before the Player software project can be compiled under the KEIL environment, the MQX BSP, PSP and MFS libraries has to be compiled. Then the precompiled libraries have to be imported to the project. The project considers the MQX default installation location is:

c:\Freescale\Freescale_MQX_4_0\. When the MQX path is different, the compiler / linker paths have to be changed in the Player software project settings.

Follow the steps below. The procedure is valid for KEIL uVision-4 compiler IDE, but it will be analogical for IAR EWARM or CodeWarrior

1. Download the AN4523SW.zip package.
2. Install the MQX 4.0 or higher from Freescale web: www.freescale.com/mqx
3. Copy the application software demo to you MQX installation directory: C:\Freescale\Freescale_MQX_4_0\demo\
4. Find the MQX build libraries workspace directory located in: MQX_DIR\config\kwikstikk40x256\uv4\

For instance:

C:\Freescale\Freescale_MQX_4_0\config\kwikstikk40x256\uv4\

5. Open the MQX “build_libs” project (file: “build_libs.uvmpw”)

   “bsp_kwikstikk40x256 Debug” has to be selected as active target -Figure 4.
6. Set “bsp_kwikstikk40x256” as the active project (right click on the selected item – Figure 5.
7. Recompile the “bsp” project.
8. Repeat the procedure for “psp” and “mfs” projects as well. (i.e. recompile: bsp, psp and mfs)

You can also run the “Batch build” for all MQX library projects (Project->Batch Build…)

9. Now open the Player project. If you can see the libs already imported under the “MQX Libraries” group (See Figure 6), you can skip that step.
10. (This step is optional.) When needed, import the precompiled MQX libraries to the Player project (you may have to remove the old ones).The pre-compiled libraries are placed in the: MQX_DIR\lib.

For instance:

C:\Freescale\Freescale_MQX_4_0\lib\kwikstikk40x256.uv4\debug\bsp\bsp.lib.

11. Import the “bsp”, “tss”, “psp” and “mfs” lib files: from the debug location which means the “debug target”. See Figure 6.
12. Finally, recompile the Player project.

For more details about the MQX usage under the KEIL compiler read the “MQX-uVision4-Getting-Started.pdf”, located in: c:\Freescale\Freescale_MQX_4_0\doc\tools\uv4\.

13. Go to the section MQX-Viewer TAD Debugger Plug-in.
14. If needed, install the uVision4 TAD, continue per the instructions in the document. See Figure 7 with the MQX TAD tool screenshot.

15. The ".wav" files have to be placed on SD card (formatted with FAT or FAT32 system) and located in the "SD:\Songs\" directory. The default song location can be changed in file "filemenu.h".

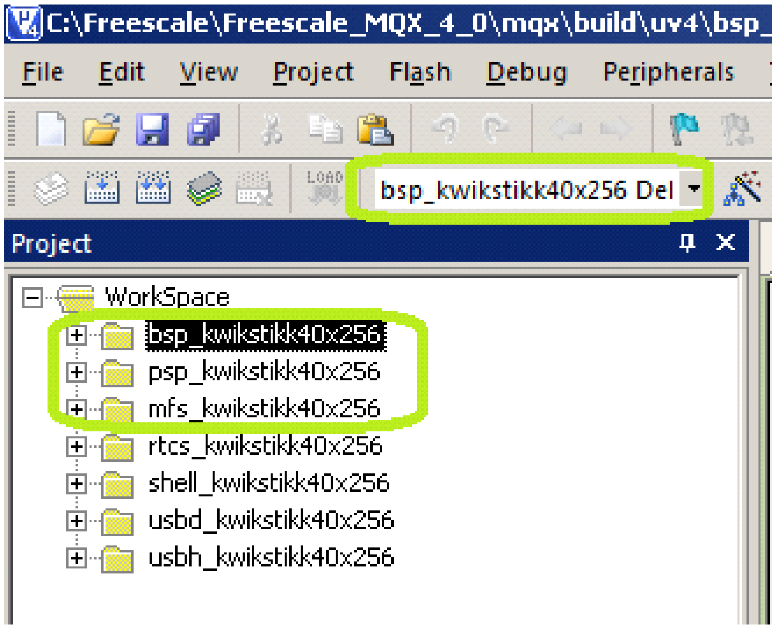


**Figure 4. Building MQX libraries**

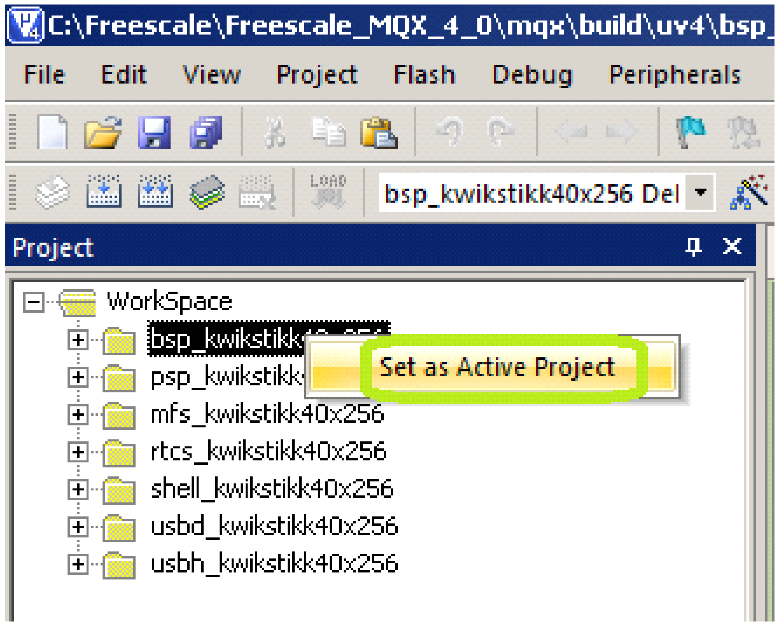


**Figure 5. Set the Active project**

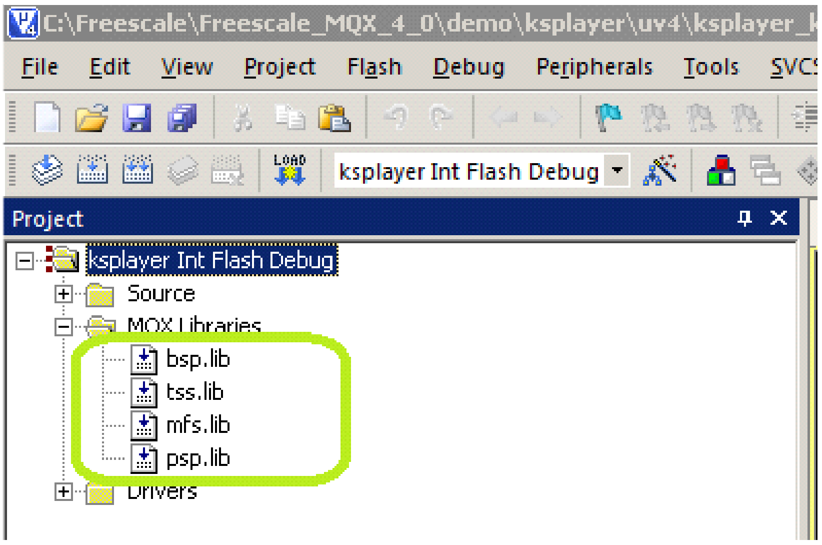
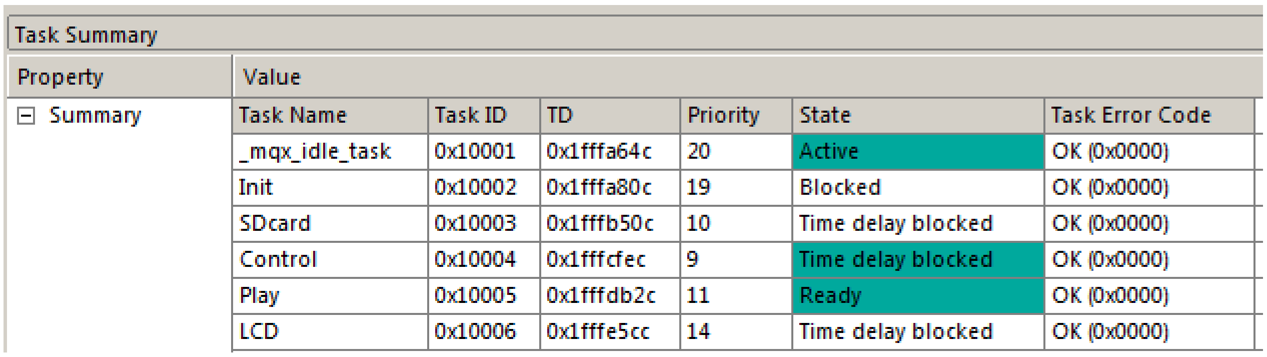

C:\Freescale\Freescale_MQX_4_0\demo\ksplayer\uv4\ksplayer_
File Edit View Project Flash Debug Peripherals Tools SVCS
ksplayer Int Flash Debug
Project
ksplayer Int Flash Debug
Source
MQX Libraries
bsp.lib
tss.lib
mfs.lib
psp.lib
Drivers

Figure 6. Import the MQX libraries to the player project



Task Summary

| Property | Value | | | | | |
|---|---|---|---|---|---|---|
| Summary | Task Name | Task ID | TD | Priority | State | Task Error Code |
| | _mqx_idle_task | 0x10001 | 0x1fffa64c | 20 | Active | OK (0x0000) |
| | Init | 0x10002 | 0x1fffa80c | 19 | Blocked | OK (0x0000) |
| | SDcard | 0x10003 | 0x1fffb50c | 10 | Time delay blocked | OK (0x0000) |
| | Control | 0x10004 | 0x1fffcfec | 9 | Time delay blocked | OK (0x0000) |
| | Play | 0x10005 | 0x1fffdb2c | 11 | Ready | OK (0x0000) |
| | LCD | 0x10006 | 0x1fffe5cc | 14 | Time delay blocked | OK (0x0000) |

Figure 7. MQX TAD tools

[1] MQX TAD tool may be very useful for tuning and debugging the MQX application like this. For instance you can see the periodical switching of the tasks, which task is active and which are blocked or in error state.

## 6 Conclusion

This Application Note presents the simple solution for uncompressed Audio playback based on Kinetis ARM Cortex-M4 MCU with CPU core running @100MHz under the MQX RTOS. The CPU provides enough processing performance, so the application can be additionally extended by audio de-compression codec.

## 7 References

freescale.com/mqx

freescale.com/kwikstik

freescale.com/tss

freescale.com/webapp/sps/site/prod_summary.jsp?code=K40_100

https://ccrma.stanford.edu/courses/422/projects/WaveFormat/

***How to Reach Us:***

**Home Page:**
freescale.com

**Web Support:**
freescale.com/support

Document Number: AN4523
Rev. 0
01/2014