Renesas Synergy™ Platform

Getting Started with the Audio Player Application using SSP

R30AN0248EU0102
Rev.1.02
Oct 11, 2018

Introduction

This application note describes the Audio Player demonstration application for Renesas Synergy™ S7G2 MCU based kits. Audio Player is a Human Machine Interface (HMI) application that uses Synergy Software Package (SSP) with GUIX™, USBX™, and FileX™. The program consists of a simple file browser with audio playback capabilities. The interface employs several GUIX widgets and touch controls to manage the underlying Audio Playback Framework and Mass Storage Controller (MSC).

Audio Player is based on SSP and uses Express Logic, Inc. ThreadX® Real-Time Operating System (RTOS), GUIX, USB, and FileX through the e² studio Integrated Solutions Development Environment (ISDE) or IAR Embedded Workbench for Synergy. The application code and configuration can be used as a reference for event-driven HMI and MSC systems. The program includes code based on GUIX and the SSP for fast graphics rendering and display, and uses the graphics accelerators available on the S7G2 devices. Additionally, it implements USBX and FileX to allow it to access data on FAT-formatted USB Mass Storage Devices.

Required Resources

- Any of the following kits
  - Renesas Synergy SK-S7G2 Starter Kit (currently supported)
  - Renesas Synergy DK-S7G2 Development Kit
  - Renesas Synergy PE-HMI1 Product Example for HMI applications
  - Renesas Synergy PK-S5D9 Promotional Kit
- e² studio ISDE v6.2.1 or later
- Synergy Software Package (SSP) 1.5.0 or later
- IAR EW 8.23.1 or later
- SSC v6.2.1 or later
- SEGGER J-link® USB driver
- Micro USB cables
- USB 2.0 Flash drive

Download all the required Renesas software from the Renesas Synergy™ Software page at https://synergygallery.renesas.com.

Supported File Formats

Audio Player application is compatible with uncompressed WAVE files in PCM format. Make sure that the audio stream is signed mono with 16 bits per sample and sampling frequency of 44.1 kHz. Audio Player will prevent incompatible files from playing and produce a warning message that is visible on a status bar.

Contents

1. Supported Boards and Files

The Audio Player demonstration application using the Audio Playback Framework supports all versions of the S7G2 boards and the PK-S5D9 board. Files supported for playback are mono, signed, 16-bit sample, 44.1 kHz PCM streams stored in a WAVE file format. Creating compatible files is explained in section 3.4, Compatible Audio Formats.

2. Running the Audio Player Application

2.1 Connecting to the Board and Importing the Project into e^2 studio

This section describes how to connect the board to power, the J-Link debugger to the PC, and the board to the PC.

More information can be obtained as follows:

- To set up the power connection and the J-Link debugger connection from your PC to the JTAG connector on the target board, refer to the S7G2 Starter Kit (SK-S7G2) Quick Start Guide, S7G2 Development Kit (DK-S7G2) Quick Start Guide HMI Product Example (PE-HMI) Quick Start Guide or S5D9 Promotional Kit(PK-S5D9)Quick Start Guide.
- For instructions on importing the project into e^2 studio and building/running the application, refer to the Synergy Project Import Guide (r11an0023eu0120_synergy_ssp.pdf, included in this package).

2.2 Running and Verifying the Application

- Before running the application, copy the audio files (bundled as part of the project) to the USB Flash drive and connect to the USB Host connector of the board.
- You will see the Splash screen on bootup. If the USB drive with the audio files to be copied to is inserted, names of the audio files will be displayed on the LCD screen. Select the audio file for playing and press the play button. Detailed screen operation is explained in the section 3.
- For boards which do not have the in-built speaker, connect the head phones.
- **On DK-S7G2**, verify that switch array S5 is configured to enable JTAG, PBs, and SDRAM.
- The audio quality varies based on the audio speaker connected. The quality is better when it is played using audio headphones

3. Application Features

3.1 Splash Screen

You will see the GUIX-rendered splash screen just after you start the program. After several seconds, the screen changes to the next screen (determined by presence of the USB mass storage device).

Figure 1 Splash Screen (for PK-S5D9/SK-S7, DK-S7/PE-HMI1)

3.2 USB Screen

The USB screen is displayed when the USB mass storage device is not detected. Certain devices are recognized fast enough for the application to skip this screen and jump straight to the main screen. Others will require additional time to fully initialize before being recognized by the Synergy board. It will also be displayed when no mass storage devices are plugged in.

Figure 2 USB Screen (top-left: SK-S7/PK-S5D9, top-right: DK-S7, bottom: PE-HMI1)

3.3 Main Screen

3.3.1 Overview

The main audio player screen appears once a USB mass storage device has been detected. This window consists of a status bar, file browser, and 3 buttons (5 buttons on PE-HMI1). The interface dynamically adjusts to reflect the current playback status and provide necessary functionality. The file browser allows the user to explore the file system and open PCM WAVE files for playback.

All screen contents are rendered by D/AVE2D drawing engine and are displayed on the TFT screen connected to the Graphic LCD Controller.

Figure 3 Main Screen (from the top: SK-S7/PK-S5D9, DK-S7, and PE-HMI1)

3.3.2 Playback Controls

The Graphical User Interface (GUI) behaves differently for each playback status. This allows screen space to be re-used in order to provide complete audio player functionality.

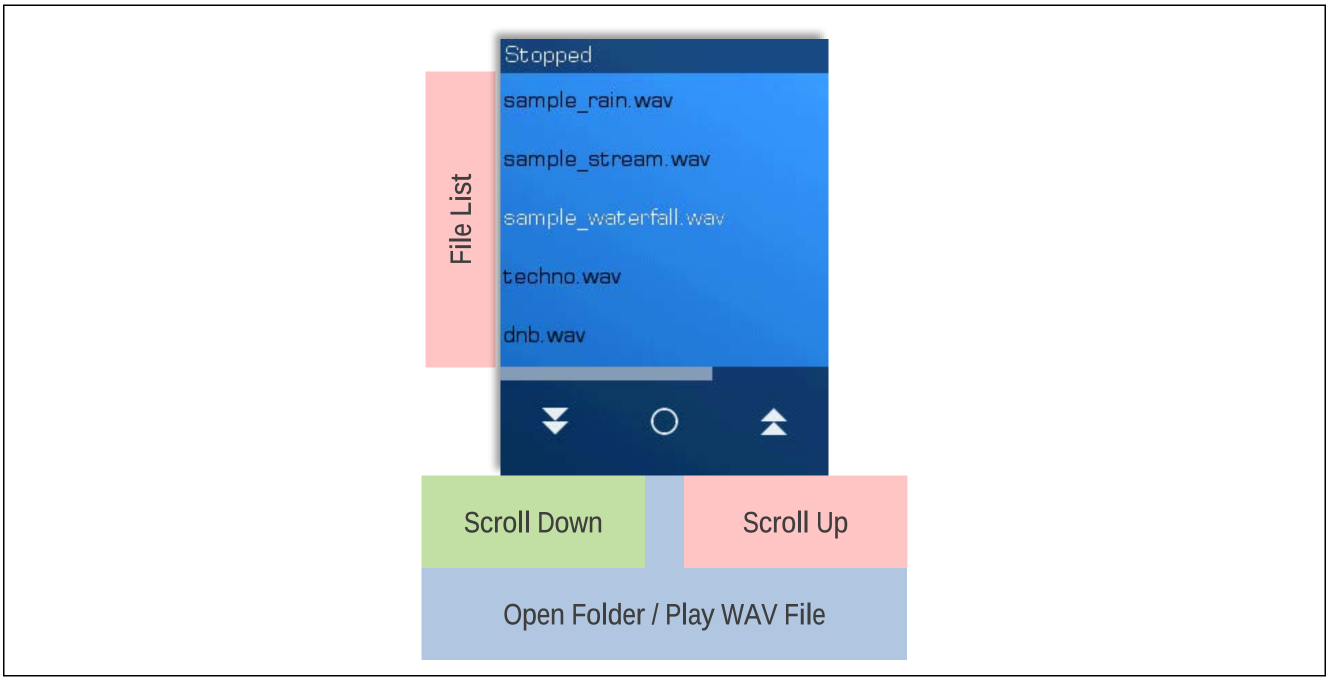


Figure 4 Main Screen (1) GUI While Playback is Stopped

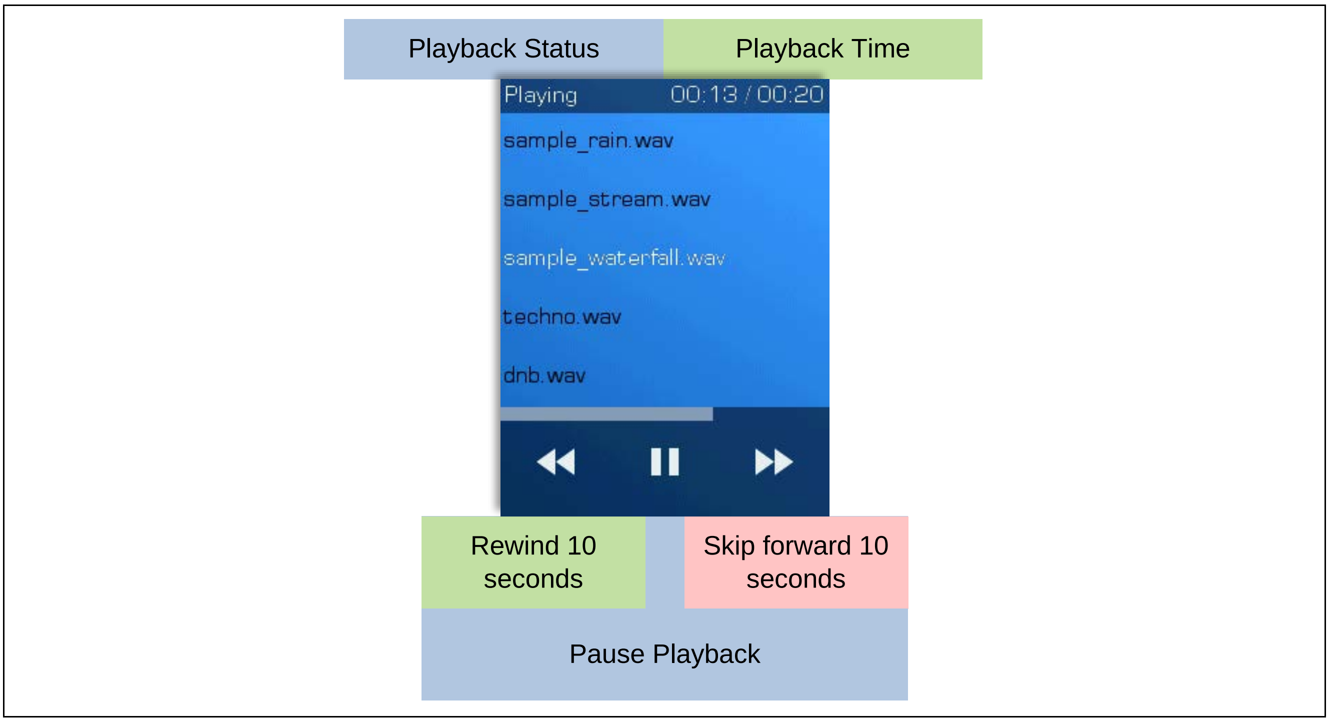


Figure 5 Main Screen (2) GUI When Audio is Playing

Figure 6 Main Screen (3) GUI When Playback is Paused

Figure 7 Main Screen on PE-HMI1 with On-screen Volume Controls

Due to limited screen space, the application uses two on-board pushbuttons (S5 and S4 on SK-S7/PK-S5D9, S2 and S3 on DK-S7) for playback volume control (down and up, respectively). PE-HMI1 version of the application implements on-screen volume controls due to lack of on-board pushbuttons.

3.4 Compatible Audio Formats

The audio player application will stop any playback and take the user back to the USB Screen if the USB device is removed. RIFF header recognition allows the program to prevent playback of incompatible WAV files. In order to play music on your Renesas Synergy™ S7G2 board, make sure that the USB mass storage device is formatted to FAT16/32 and the WAVE audio files are uncompressed (PCM), signed, 16-bit per sample, mono, 44.1-kHz streams. Status bar will display an error message once incompatible file attempts playback. Compatible audio files can be created from most input formats using the ffmpeg codec with the following command:

```
ffmpeg -i %infile% -acodec pcm_s16le -ac 1 -ar 44100 %outfile%
```

4. Application Design

4.1 Source Code Layout

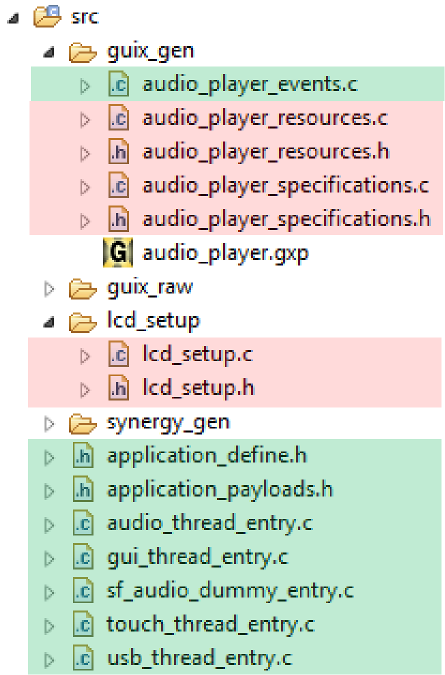
Application code for Audio Player is split between four thread entry files as well as the GUIX Event Handler and a custom application-defined header file and payload definition file. This creates robust building blocks that can easily be ported into another application (if the configuration for the thread and its modules is also copied).

User-created source files:

- application_define.h
- application_payloads.h
- audio_thread_entry.c
- gui_thread_entry.c
- touch_thread_entry.c
- usb_thread_entry.c
- sf_audio_dummy_entry.c
- guix_gen/audio_player_events.c

User-generated or referenced source files:

- audio_player_specifications.c/.h
- audio_player_resources.c/.h
- lcd_setup/lcd_setup.c/.h (only on SK-S7/PK-S5D9)



4.2 Thread Layout

4.2.1 Module Hierarchy

Figure 8 illustrates the hierarchy and dependency of the modules belonging to each thread. Conventional design implements GUI and Touch functionality on the same thread (usually called HMI). The Audio Player application separates the two to emphasize independency between touch and display peripherals, and to provide separate subscriber threads for touch and callback messages, both of which are converted and forwarded to GUIX.

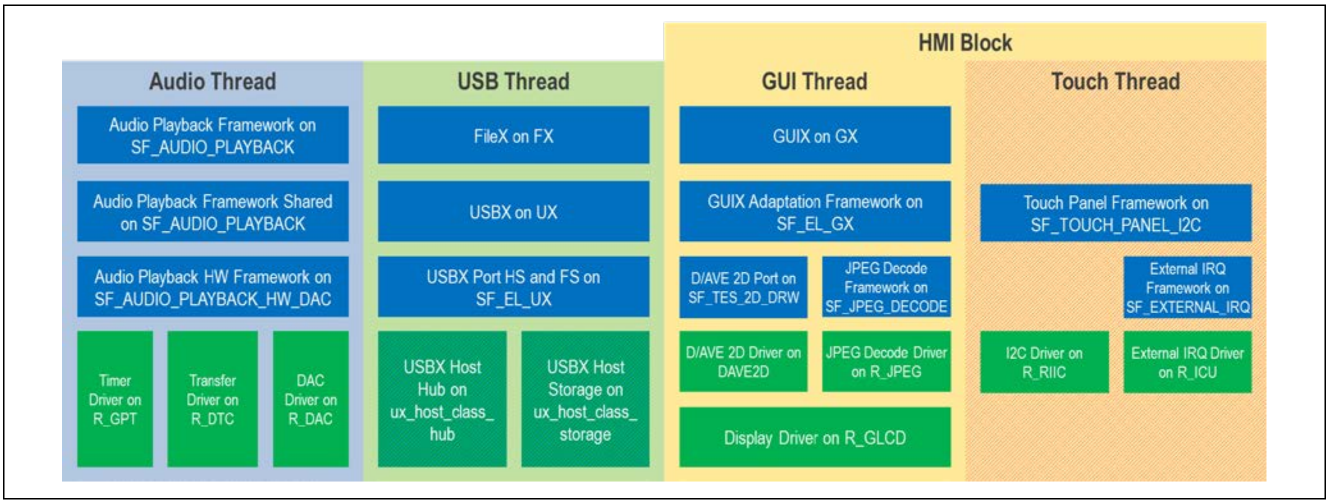


Figure 8 Module Hierarchy

4.2.2 Thread Modules and Objects

In addition to core module blocks visible on the previous diagram, the Audio Player application employs additional ThreadX objects such as queues and semaphores. These objects enable communication between threads and control the code execution without blocking the processor.

4.3 Thread Initialization

Even though different functionalities are isolated into separate threads, minimal dependency between each thread is still present. When launching the application, make sure that modules, objects, and variables that are not initialized yet are not accessed from another thread. Fundamental functionality (Mass Storage Controller and Audio Thread) must be initialized before proceeding to enable GUI and other HMI features (Touch). Audio Player using Audio Playback Framework launches its threads in the order shown in Figure 9.

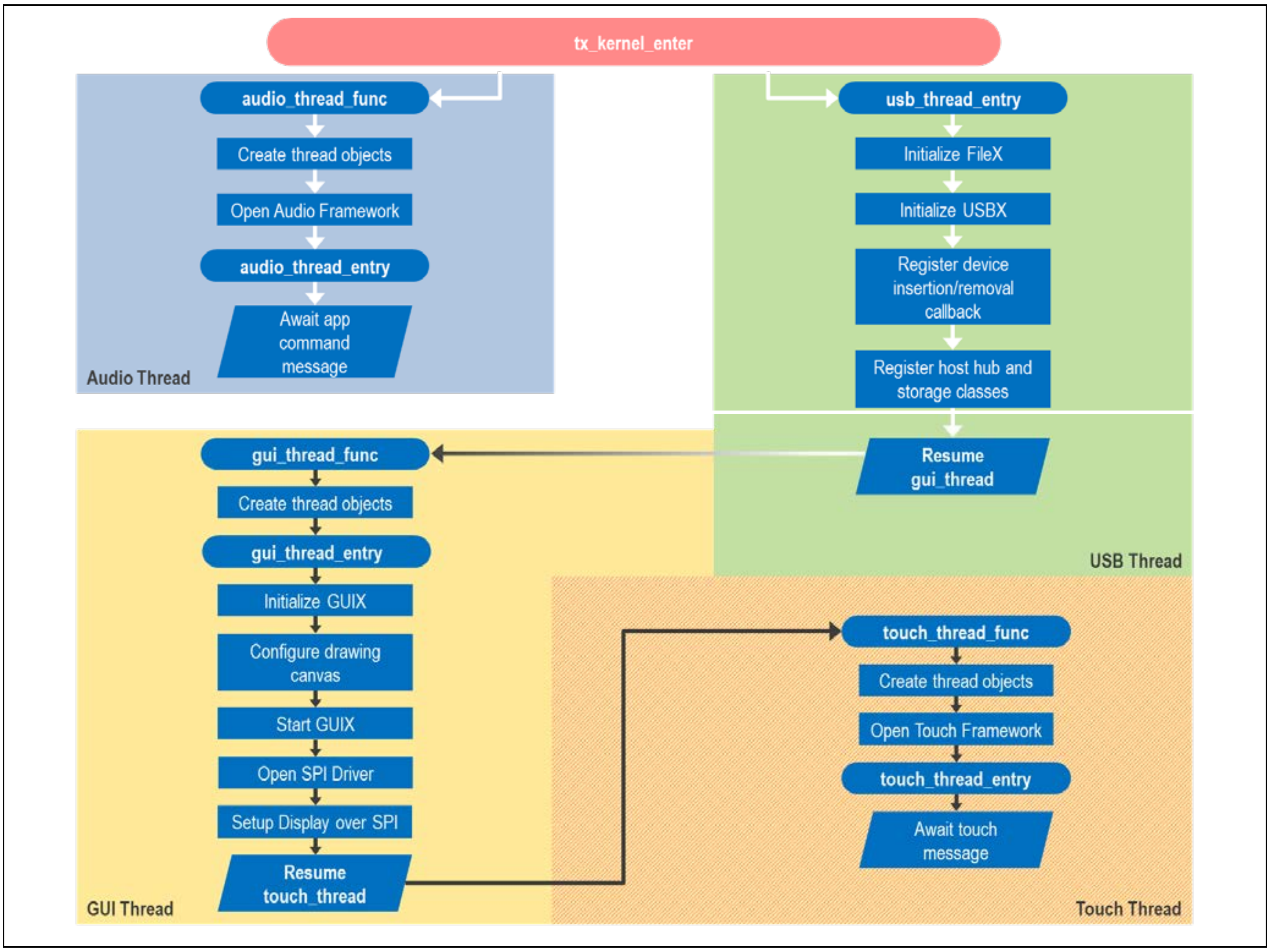


Figure 9 Thread Initialization Order

4.4 Thread Resources

4.4.1 Static Variables

- usb_thread_entry.c
  - Storage instance in **gp_storage*
  - Media pointer in **gp_storage_media*
  - USB host stack in *g_usb_memory*
- audio_thread_entry.c
  - 2 read buffers in *g_file_buffer*[2]
  - Buffered data size in *g_acutal_size*[2]
  - Index of accessed buffer in *g_current_buffer*
- audio_player_events.c
  - Directory file list in *g_file_list*
  - Index of the selected list entry in *g_file_list_index*

4.4.2 Global Variables

- Media Storage pointer in **gp_media*
  - Declared in usb_thread_entry scope
  - Accessed from audio_thread_entry scope
- Audio file RIFF header in *g_file_info* and audio player status in *g_player_status*
  - Declared in audio_thread_entry scope
  - Accessed from audio_player_events scope
- Root window pointer in **p_window_root*
  - Declared in gui_thread_entry scope
  - Accessed from audio_player_events scope

4.5 Inter-thread Communication

The messaging framework is the core module responsible for communication between threads. It provides a thread-safe way of sharing variables that is built on top of the ThreadX Queue module. It expands the basic functionality by providing a unified scheme for message format that includes its class, code, and payload. Since queues are only used to pass a pointer to the message, the actual message size can be greatly increased to accommodate custom payload structure. The Audio Player application uses four different message classes:

- SSP-defined: *TOUCH* and *AUDIO*
- Application-defined: *APP_CMD* and *APP_CB* (application command and callback, respectively)
- Command messages (APP_ prefix):
  - Posted from GUIX Event Thread and pushbutton interrupt callback function
  - Received and parsed by Audio Thread
- Callback messages (APP_CB and APP_ERR prefixes):
  - Posted from Audio and USB Threads
  - Received by GUI Thread and forwarded to GUIX Event Thread as GX_EVENT

Figure 10 Command and Callback Messages

4.5.1 Command Messages

All command messages in the Audio Player application are processed by the Audio Thread. They lead to a sequence of FileX and SF_MESSAGE API calls in order to control playback and data flow into the internal Audio Framework Thread. The messages can be sent from any thread, allowing the user to replace the GUI with a different interface while retaining full audio playback control capabilities as the underlying USB and Audio Threads are not affected.

Most commands use a payload structure to provide additional data such as the name of the file to be opened or the step size for volume change. Successful execution of the command generates a callback message, which is fed back into the GUIX Event Thread to update status of the interface or display an on-screen message.

4.5.2 Callback Messages

The Audio Thread also posts callback messages to report completion of the task (or an error, if encountered). As the majority of commands are posted from the GUIX Event Thread, the callback messages are routed into the GUIX Event Thread for the GUI to reflect the Audio Thread status (by adjusting the interface and displaying messages).

Every status change generates a callback message. Therefore, the GUI does not have to check global variables regularly. Instead, it only needs to acquire status upon receiving a message. All callback messages are interpreted into GX_EVENTs so that they are handled within the window event process, along with other GUI events such as touchscreen input. When an error callback is posted, the message payload carries a return value from the function that failed to execute.

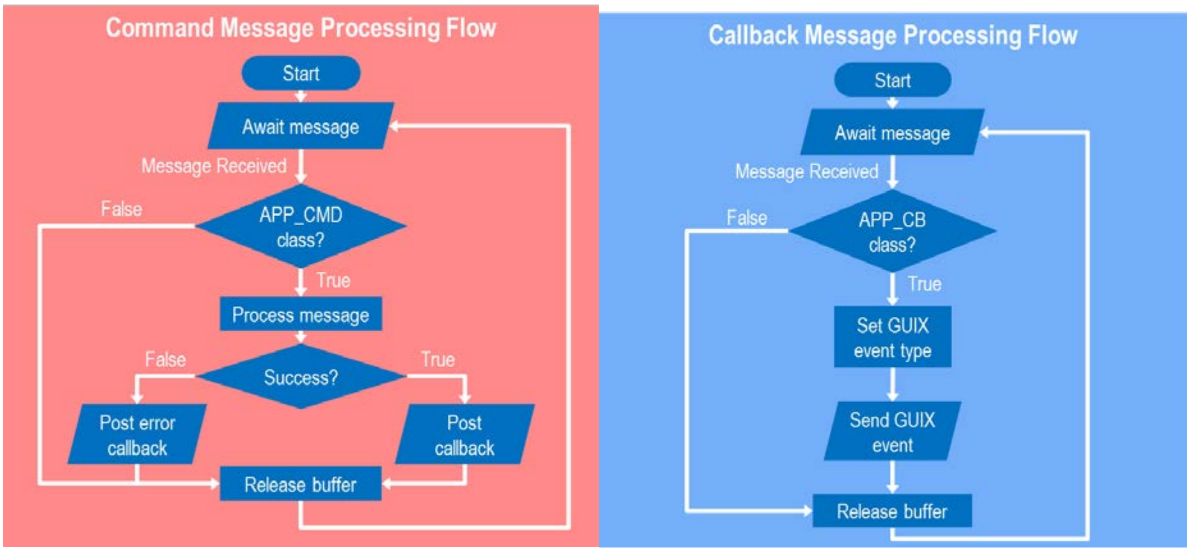


Figure 11 Message Flow

4.6 Message Processing Flow Example

Figure 12 shows an example of the steps taken to process an “open” command and generate callback upon successful completion of the task. Note that the flowchart is simplified and focuses on illustrating message flow between the threads rather than step-by-step execution sequence.

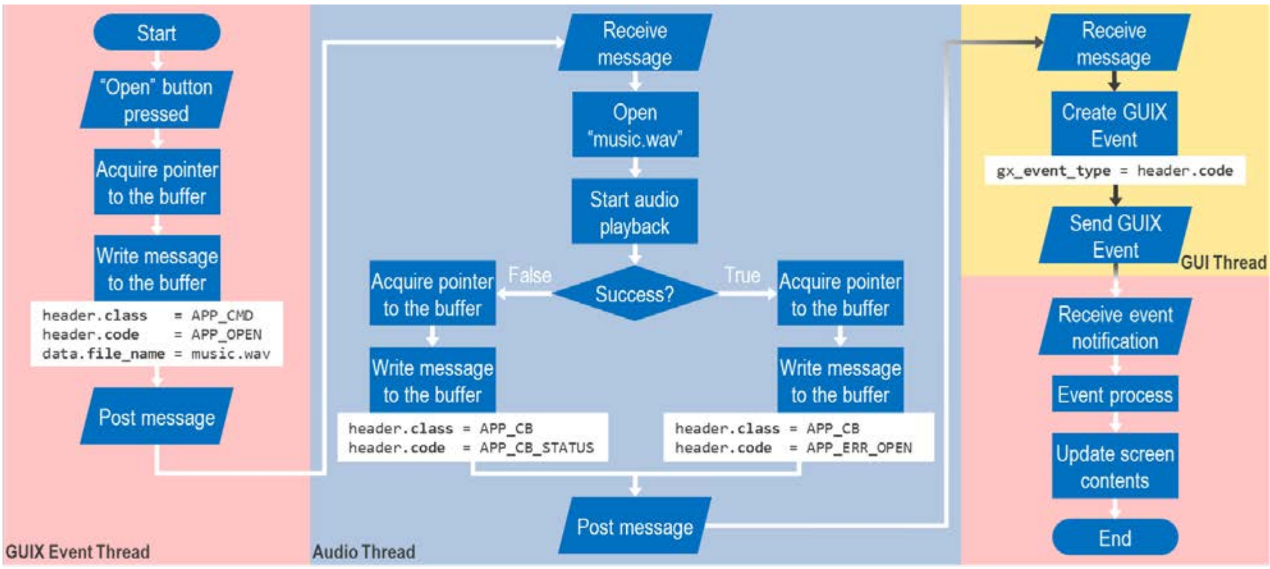


Figure 12 Message Processing Example

For more detail on audio thread processing, see the section 4.7.

4.7 Audio Thread Processing Flow

Audio Thread implements the Audio Playback Framework, creating the foundation of this application. It is fully controlled using the messaging framework. Since messages can be sent and received by any thread, the graphical user interface (GUI) of this application can easily be replaced with an alternative input/output media such as USB CDC AMC device (using USBX), telnet console, or a remote website (both using NetX).

Ping-pong buffering is used to preload audio data while maintaining seamless playback and GUI responsiveness. Buffer size is set to 8 KB, which is enough to store about 90 ms of the 16-bit PCM mono stream.

The thread runs in a permanent loop, first awaiting the message for a single ThreadX tick (default value: 10 ms) before generating timeout and resuming with playback processing. Here, player status is verified (to be *PLAYING*) before the thread attempts to push another set of samples into the buffer. A counting semaphore determines whether the contents of the buffer have been played. When playback is *PAUSED* or *STOPPED*, or semaphore count is zero, the thread hands off processor control to other threads before restarting the loop to await another message.

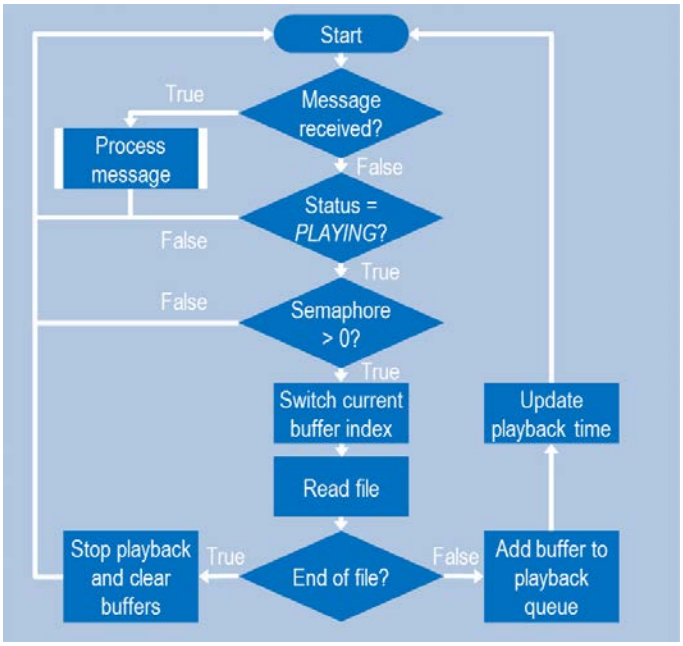


Figure 13 Audio Thread Processing Flow

4.8 GUI Structure

The graphical user interface (GUI) is the primary input and output media for this application. The GUI is split into three screens: splash, USB, and main. Content windows are embedded into, and displayed over their background parent window to allow independent control of the background and overlay pixel maps. Events for all widgets (such as buttons and text labels) are handled by the event process of their parent window.

Audio Player application uses 16 alpha-enabled pixel maps and two 8-bit fonts to provide a smooth GUI experience. These resources add up to just under 440 kB, leaving plenty of room in flash memory for alternative themes and additional features.

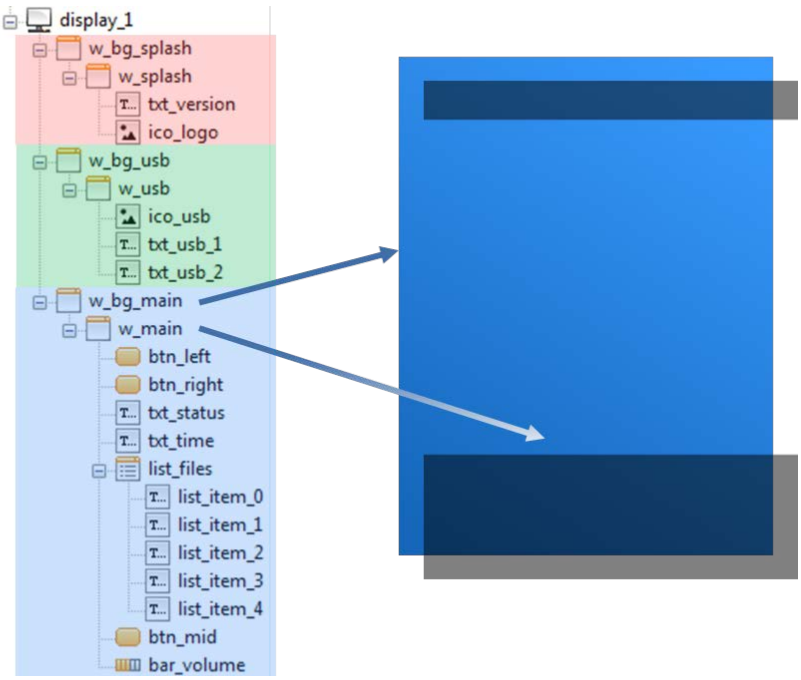


Figure 14 GUI Structure

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	renesassynergy.com/software
Synergy Software Package	renesassynergy.com/ssp
Software add-ons	renesassynergy.com/addons
Software glossary	renesassynergy.com/softwareglossary
Development tools	renesassynergy.com/tools
Synergy Hardware	renesassynergy.com/hardware
Microcontrollers	renesassynergy.com/mcus
MCU glossary	renesassynergy.com/mcuglossary
Parametric search	renesassynergy.com/parametric
Kits	renesassynergy.com/kits
Synergy Solutions Gallery	renesassynergy.com/solutionsgallery
Partner projects	renesassynergy.com/partnerprojects
Application projects	renesassynergy.com/applicationprojects
Self-service support resources:	
Documentation	renesassynergy.com/docs
Knowledgebase	renesassynergy.com/knowledgebase
Forums	renesassynergy.com/forum
Training	renesassynergy.com/training
Videos	renesassynergy.com/videos
Chat and web ticket	renesassynergy.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun 21, 2016	-	Initial version.
1.01	Nov 18, 2016	Title	Minor formatting changes.
1.02	Oct 11, 2018	All	Updated for SSP v1.5.0

All trademarks and registered trademarks are the property of their respective owners.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

© 2018 Renesas Electronics Corporation. All rights reserved.