

Sound MCU: S1C31D41

Demonstration Software Guide

"2. SOUNDPLAY (Sound Playback Example)"



Rev1.00

- Purpose : This document describes “SOUNDPLAY” software.

1. Evaluation Board

- 1.1. Jumper Settings
- 1.2. PC Connection

2. Demonstration Software

- 2.1. Build and Download
- 2.2. Operation Flow

3. HWP Library Specification(SOUNDPALY function)

- | | |
|------------------------------------|----------------------------------|
| 3.1. seSoundPlayInit | 3.12. seSoundPlayFinish |
| 3.2. seSoundPlayInit_ELBUZZER | 3.13. seSoundPlayFinish_ELBUZZER |
| 3.3. seSoundPlayInit_PIBUZZER | 3.14. seSoundPlayFinish_PIBUZZER |
| 3.4. seSoundPlaySetPCM | |
| 3.5. seSoundPlaySetParameter | |
| 3.6. seSoundPlaySetParameter_Speed | |
| 3.7. seSoundPlaySetParameter_Pitch | |
| 3.8. seSoundPlayRunCommand | |
| 3.9. seSoundPlayGetState | |
| 3.10. seSoundPlayStartTone | |
| 3.11. seSoundPlayStopTone | |

1. Evaluation Board

1.1. Jumper Settings

Set the jumpers on the evaluation board according to each mode.

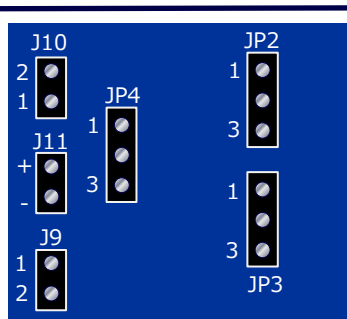


Table1.1 Output Mode Setting

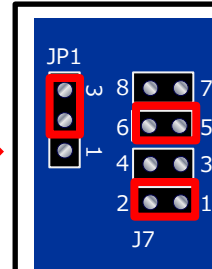
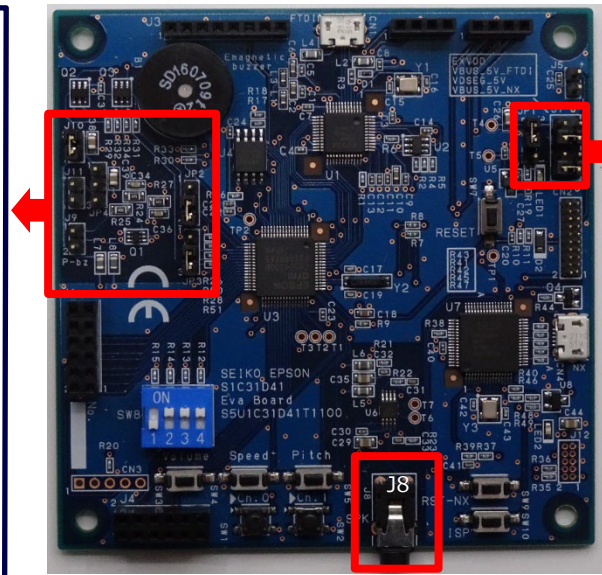


Table 1.2 Power Supply Setting

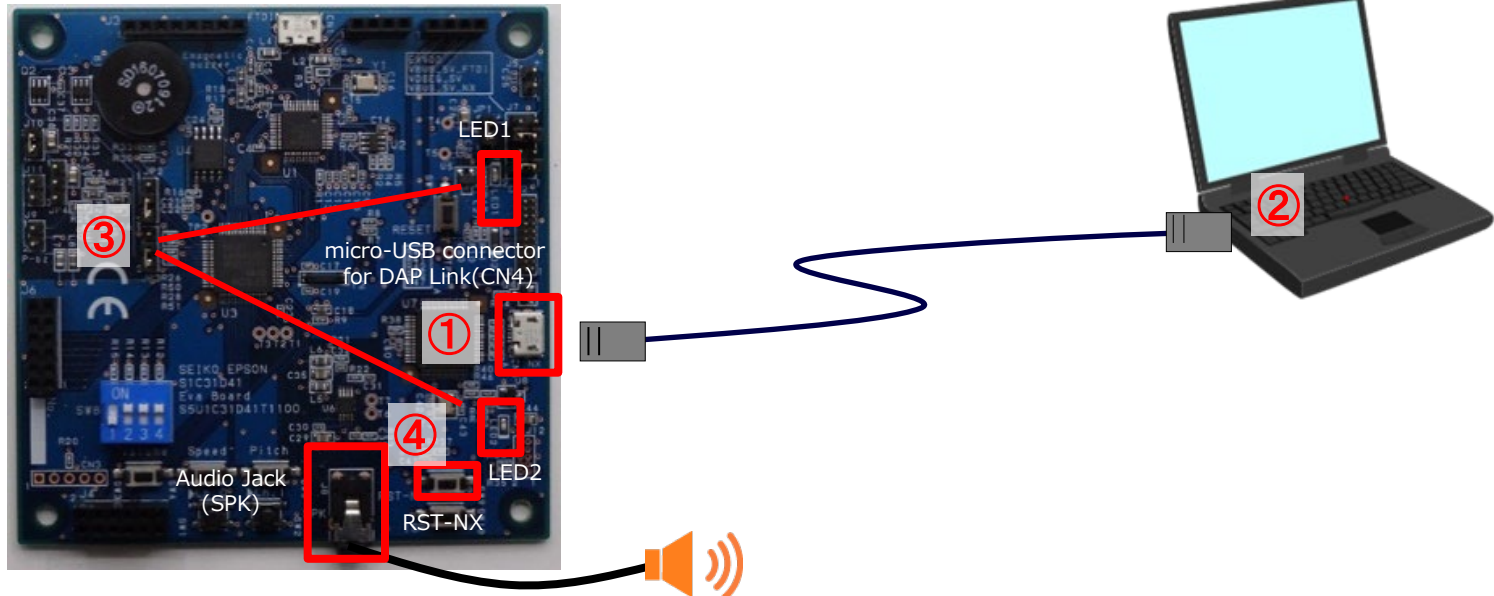
Power	J7	JP1
USB 5V supply	1-2 short 3-4 open 5-6 short 7-8 open	2-3 short

Output Mode	SW8-1	SW8-2	JP2	JP3	JP4	J10	J8	J9	J11
Speaker	OFF	OFF	-	-	1-2 open	1-2 open	Speaker	-	-
Electromagnetic Buzzer	OFF	ON	2-3 short	2-3 short	1-2 open	1-2 short	-	-	-
Piezoelectric Buzzer(USB 5V)	ON	OFF	1-2 short	1-2 short	1-2 short	1-2 open	-	Buzzer	-
Piezoelectric Buzzer(Ext. Voltage)	ON	OFF	1-2 short	1-2 short	2-3 short	1-2 open	-	Buzzer	Ext. Power

1.2. PC Connection

Follow the steps below to connect the PC to the evaluation board.

- ① Connect the micro USB cable to “CN4” connector on the evaluation board.
- ② Connect the micro USB cable to the USB port of the PC with IDE installed.
- ③ Make sure LED1 and LED2 on the evaluation board are lit.
- ④ Push “RST-NX” button.



2. Demonstration Software

2.1. Build and Download

Build **SOUNDPLAY** project in S1C31D41 peripheral sample software package and download the software to the evaluation board.

1. Double click the SOUNDPLAY workspace file to launch the IDE.
2. Set the output mode setting constant in the preprocessor symbol definition. (See Table 2.1)
3. Build this project and download the built software to S1C31D41 on the evaluation board.

* For details on software build and download, refer to the following documents.

- S1C31 Family Peripheral Circuit Sample Software Manual

表2.1 Output Mode Definition

Constant for output mode	Defined
PLAY_AMPSPEAKER	Output the sound amplified by the audio amplifier from the speaker
PLAY_ELBUZZER	Output the sound amplified by the discrete circuit from the electromagnetic buzzer
PLAY_PIBUZZER	Output the sound amplified by the discrete circuit from the piezoelectric buzzer

```
[s1c31d5xd41sp_verx_xx]
- [Licenses] : License group
- [Drivers] : Driver group
  - [board] : Drivers related to the evaluation board
    - [S5U1C31D41T1]
      - [ARM]
      - [IAR]
      - board.c/.h : Evaluation board setting program
      - settings.h : Definition file for setting the function of the evaluation board
    - ...
  - [CMSIS] : CMSIS Drivers
    - [Device]
      - [S1C31D41]
        - [Include]
          - S1C31D41.h : CMSIS peripheral circuit access layer header file
          - ...
        - [Source]
          - [ARM]
          - [IAR]
          - startup_S1C31D41.s : CMSIS startup program
          - system_S1C31D41.c : CMSIS peripheral circuit access layer program
        - ...
      - [Driver]
      - [SVD]
    - [sePeripheralLibrary] : Peripheral circuit library
      - se_clg.c/.h
      - se_i2c.c/.h
      - ...
  - [Middlewares] : Middleware group
    - [seHWPProcLibrary] : HW Processor library
    - [seEepromLibrary] : EEPROM emulation library
    - [seFlashLibrary] : Self-programming library
  - [Projects] : Sample software group
    - [Demonstration] : Various demonstration software
      - [SOUNDPLAY] : Sound Playback Example Project
        - [ARM] : MDK-ARM Project
          - soundplay_D41.uvmpw : Workspace file
        - [IAR] : IAR EWARM Project
          - SOUNDPLAY_D41.eww : Workspace file
        - main.c
        - ...
      - ...
    - [Applications] : Application software group
    - [Examples] : Peripheral example software group
  - README.txt
```

2.2. Operation Flow

SOUNDPLAY executes sound playback such as 2-channel mixing, speed conversion, pitch conversion and tone playback using the sound playback function provided by the HW processor.

1. Set System Clock as OSC3 16MHz for SOUNDPLAY. (See InitSCLK function in main.c.)
2. Initialize the port for amplifier control(*1). (See InitAMP function in main.c.)
3. Set the starting address and the size of Sound ROM in internal Flash.
4. Play the sentence composed of sound files in internal Flash. (See RunSoundPlay function in main.c)
 - 4.1. Call seSoundPlayInit() function to initialize the SOUNDPLAY.
 - 4.2. Call seSoundPlaySetParameter() function to set a sentence number, channel, volume, repeats, speed and pitch.
 - 4.3. Call seSoundPlayRunCommand() function to start playing the specified sentence.
 - 4.4. Wait for an interrupt indicating the sound playback is complete.
 - 4.5. Repeat steps 4.2 to 4.4 if necessary.
 - 4.6. Call seSoundPlayFinish() function to finish the SOUNDPLAY.
5. Initialize QSPI for access to external QSPI-Flash. (See InitExtFlash function in extflash_read.c)
6. Set the starting address and the size of Sound ROM in external QSPI-Flash.
7. Play the sentence composed of sound files in external QSPI-Flash. (See RunSoundPlay function in main.c)
 - 7.1. Call seSoundPlayInit() function to initialize the SOUNDPLAY.
 - 7.2. Call seSoundPlaySetParameter() function to set a sentence number, channel, volume, repeats, speed and pitch.
 - 7.3. Call seSoundPlayRunCommand() function to start playing the specified sentence.
 - 7.4. Wait for an interrupt indicating the sound playback is complete.
 - 7.5. Repeat steps 4.2 to 4.4 if necessary.
 - 7.6. Call seSoundPlayFinish() function to finish the SOUNDPLAY.

*1: This process is valid only when the output mode is "PLAY_AMPSPEAKER".

3. HWP Library Specification (SOUNDPALY function)

3.1. seSoundPlayInit

Syntax	<pre>void seSoundPlayInit (unsigned long rom_start_address, unsigned long rom_size, unsigned long keycode)</pre>
Arguments	<p>rom_start_address SoundROM start address</p> <ul style="list-style-type: none">- Integrated Flash: 0x00 0000, ..., 0x02 FFF0 (16byte alignment)- External QSPI Flash: 0x00 0000 + OFFSET 0x10 0000 + OFFSET 0x20 0000 + OFFSET ... 0xE0 0000 + OFFSET 0xF0 0000 + OFFSET ➔ OFFSET : 0x04 0000. This is start address of memory mapped access area for external QSPI-Flash. <p>rom_size SoundROM data size</p> <ul style="list-style-type: none">- Integrated Flash: ≤ 0x03 0000 Byte (192 KByte)- External QSPI Flash: ≤ 0x10 00000 Byte(16 MByte) <p>keycode 32bit code Epson provides, requires to decode eov file in SoundROM *This value must be same with the keycode on Epson Voice Data Creation PC Tool(ESPER2)</p>
Return value	-
Explanation	Initializes SOUNDPLAY function. Call this function first when using SOUNDPLAY in output mode with amplifier and speaker.

3.2. seSoundPlayInit_EL buzzer

Syntax	<code>void seSoundPlayInit_EL buzzer(unsigned long rom_start_address, unsigned long rom_size, unsigned long keycode)</code>
Arguments	Same as seSoundPlayInit
Return value	-
Explanation	Initializes SOUNDPLAY function. Call this function first when using SOUNDPLAY in output mode with electromagnetic buzzer and recommended drive circuit.

3.3. seSoundPlayInit_PIBuzzer

Syntax	<code>void seSoundPlayInit_PIBuzzer(unsigned long rom_start_address, unsigned long rom_size, unsigned long keycode)</code>
Arguments	Same as seSoundPlayInit
Return value	-
Explanation	Initializes SOUNDPLAY function. Call this function first when using SOUNDPLAY in output mode with piezoelectric buzzer and recommended drive circuit.

3.4. seSoundPlaySetPCM

Syntax	<pre>void seSoundPlaySetPCM(unsigned long pcm_start_address, unsigned long pcm_size)</pre>
Arguments	<p>pcm_start_address Start address of the PCM data to be stored in memory.</p> <ul style="list-style-type: none">- In case of RAM: 0x15 0000, ..., 0x15 1FFF 0x15 3000, ..., 0x15 67FF- In case of internal Flash: 0x00 0000, ..., 0x02 FFF0 (16byte alignment)- In case of external QSPI-Flash: 0x00 0000 + OFFSET 0x10 0000 + OFFSET 0x20 0000 + OFFSET ... 0xE0 0000 + OFFSET 0xF0 0000 + OFFSET * OFFSET : 0x04 0000. This is start address of memory mapped access area for external QSPI-Flash. <p>pcm_size Size of the PCM data</p>
Return value	-
Explanation	<p>Sets the memory area where the PCM data is stored.</p> <p>By using this function, you can play sound data (PCM format only) that is not included in SoundROM.</p> <p>To play the PCM data set by this function, set "0" to the sentence number that is the argument of the seSoundPlaySetParameter function.</p>

3.5. seSoundPlaySetParameter

Syntax	<pre>void seSoundPlaySetParameter (unsigned char ch, unsigned short sentenceNo, unsigned char volume, unsigned char repeat, unsigned char speed, unsigned char pitch)</pre>
Arguments	<p>ch Channel to play 0: channel0, 1: channel1</p> <p>sentenceNo Sentence Number to play 1(0x0001)~65535(0xFFFF) : Play the sentence in the Sound ROM 0(0x0000) : Play the PCM data set by the seSoundPlaySetPCM function</p> <p>volume Sound Volume 0x7F:0dB, 0x7E:-0.5dB, 0x7D:-1.0dB, ..., 0x02:-63dB, 0x01:-63.5dB, 0x00:No Sound</p> <p>repeat Repeat times for Sentence Play 0x01 to 0xFE: Repeat time, 0xFF: keep repeating until Stop command receive</p> <p>speed Sound Speed(valid on ch0:voice, invalid on ch1:background music to keep original) 85:85%(Slow), ..., 100:100%(Normal), ..., 115: 115%(Fast) *5% Step</p> <p>pitch Sound Pitch(valid on ch0:voice, invalid on ch1:background music to keep original) 90:90%(Slow), ..., 100:100%(Normal), ..., 110: 110%(Fast) *5% Step</p>
Return value	-
Explanation	Set the parameters for SOUNDPLAY. Use this function to use both conversion of speed and pitch at the same time.

3.6. seSoundPlaySetParameter_Speed

Syntax	<pre>void seSoundPlaySetParameter_Speed (unsigned char ch, unsigned short sentenceNo, unsigned char volume, unsigned char repeat, unsigned char speed)</pre>
Arguments	<p>ch Channel to play 0: channel0, 1: channel1</p> <p>sentenceNo Sentence Number to play 1(0x0001)~65535(0xFFFF) : Play the sentence in the Sound ROM 0(0x0000) : Play the PCM data set by the seSoundPlaySetPCM function</p> <p>volume Sound Volume 0x7F:0dB, 0x7E:-0.5dB, 0x7D:-1.0dB, ..., 0x02:-63dB, 0x01:-63.5dB, 0x00:No Sound</p> <p>repeat Repeat times for Sentence Play 0x01 to 0xFE: Repeat time, 0xFF: keep repeating until Stop command receive</p> <p>speed Sound Speed(valid on ch0:voice, invalid on ch1:background music to keep original) 75:75%(Slow), ..., 100:100%(Normal), ..., 125: 125%(Fast) *5% Step</p>
Return value	-
Explanation	Set the parameters for SOUNDPLAY. Use this function if you want to use only the speed conversion without using the pitch conversion.

3.7. seSoundPlaySetParameter_Pitch

Syntax	<pre>void seSoundPlaySetParameter_Pitch (unsigned char ch, unsigned short sentenceNo, unsigned char volume, unsigned char repeat, unsigned char pitch)</pre>
Arguments	<p>ch Channel to play 0: channel0, 1: channel1</p> <p>sentenceNo Sentence Number to play 1(0x0001)~65535(0xFFFF) : Play the sentence in the Sound ROM 0(0x0000) : Play the PCM data set by the seSoundPlaySetPCM function</p> <p>volume Sound Volume 0x7F:0dB, 0x7E:-0.5dB, 0x7D:-1.0dB, ..., 0x02:-63dB, 0x01:-63.5dB, 0x00:No Sound</p> <p>repeat Repeat times for Sentence Play 0x01 to 0xFE: Repeat time, 0xFF: keep repeating until Stop command receive</p> <p>pitch Sound Pitch(valid on ch0:voice, invalid on ch1:background music to keep original) 75:75%(Slow), ..., 100:100%(Normal), ..., 125: 125%(Fast) *5% Step</p>
Return value	-
Explanation	Set the parameters for SOUNDPLAY. Use this function if you want to use only the pitch conversion without using the speed conversion.

3.8. seSoundPlayRunCommand

Syntax	void seSoundPlayRunCommand (unsigned char ch, unsigned short command)	
Arguments	ch Channel to play 0: channel0 1: channel1 2: all channel command Control command for SOUNDPLAY 0x01 (sp_command_start): 0x02 (sp_command_stop): 0x03 (sp_command_stop_after_phrase): 0x04 (sp_command_pause): 0x05 (sp_command_pause_after_phrase): 0x06 (sp_command_unpause): 0x07 (sp_command_mute): 0x08 (sp_command_mute_after_phrase): 0x09 (sp_command_unmute):	 Sound Play Start Sound Stop immediately Sound Stop after current phrase Sound Pause immediately Sound Pause after current phrase Release Pause Sound Mute Immediately Sound Mute after current phrase Release Mute
Return value	-	
Explanation	Run SOUNDPLAY Control Command. Before executing "Sound Play Start" command with this function, call the seSoundPlaySetParameters function.	

3.9. seSoundPlayGetState

Syntax	unsigned short seSoundPlayGetState (unsigned char ch)
Arguments	ch Channel to play 0: channel0 1: channel1
Return value	SOUNDPLAY Status 0x0000 (sp_state_init): On initializing 0x0001 (sp_state_idle): Idle 0x0002 (sp_state_play): Playing 0x0003 (sp_state_pause): Pausing 0x0004 (sp_state_mute): Muting
Explanation	Get the current status of SOUNDPLAY function.

3.10. seSoundPlayStartTone

Syntax	void seSoundPlayStartTone (unsigned short tone)
Arguments	tone - Frequency division value of tone signal (TONEDIV) Calculate the frequency division value from the following formula. tone frequency= fSDAC2CLK / [(4 × TONEDIV + 4) × 2] fSDAC2CLK: SDAC2 operating clock frequency set using the SDAC2CLK register [Hz] TONEDIV: Value set in the SDAC2TONE.TONEDIV[15:0] bits
Return value	-
Explanation	Start the output of the tone signal.

3.11. seSoundPlayStartTone

Syntax	void seSoundPlayStopTone (void)
Arguments	-
Return value	-
Explanation	Start the output of the tone signal.

3.12. seSoundPlayFinish

Syntax	void seSoundPlayFinish (void)
Input argument	-
Return value	-
Explanation	Finish SOUNDPLAY function in output mode with speaker and amplifier.

3.13. seSoundPlayFinish_EL buzzer

Syntax	void seSoundPlayFinish_EL buzzer (void)
Input argument	-
Return value	-
Explanation	Finish SOUNDPLAY function in output mode with electromagnetic buzzer and recommended drive circuit.

3.14. seSoundPlayFinish_PIBuzzer

Syntax	void seSoundPlayFinish_PIBuzzer (void)
Input argument	-
Return value	-
Explanation	Finish SOUNDPLAY function in output mode with piezoelectric buzzer and recommended drive circuit.

EPSON
EXCEED YOUR VISION