

Sound MCU: S1C31D41 Sensor demo(fire alarm)

“Reference Design Guide”

Rev1.00

Evaluation board/kit and Development tool important notice

1. This evaluation board/kit or development tool is designed for use with engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the design requirements of finished products.
2. This evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by its use. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. Parts used for this evaluation board/kit or development tool may be changed without any notice.

NOTICE : PLEASE READ THE FOLLOWING NOTICE CAREFULLY BEFORE USING THIS DOCUMENT

The contents of this document are subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation (“Epson”).
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson’s official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties’ intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson’s web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) and applications individually listed in this document (“General Purpose”). Epson products are NOT intended for any use beyond the General Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below. Therefore, you are advised to use Epson products only for General Purpose uses. Should you desire to buy and use Epson products for a particular purpose other than a General Purpose uses, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any particular purpose. Please be sure to contact our sales representative and obtain approval in advance.
【Particular purpose】
Space equipment (artificial satellites, rockets, etc.)
Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
Medical equipment (other than applications individually listed in this document) / Relay equipment to be placed on ocean floor
Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment
Other applications requiring similar levels of reliability as those listed above
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
Company names and product names listed in this document are trademarks or registered trademarks of their respective companies. Arm, Cortex, Keil and μ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. IAR Systems, IAR Embedded Workbench, C-SPY, I-Jet, IAR and the logotype of IAR Systems are trademarks or registered trademarks owned by IAR Systems AB. SEGGER and J-Link are trademarks or registered trademarks of SEGGER Microcontroller GmbH & Co. KG. All rights reserved. All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies. “Reproduced with permission from Arm Limited. Copyright © Arm Limited”

1. Overview
2. Evaluation Board “S5U1C31D41T” and external Parts
3. Startup
 1. Writing Software
 2. Writing sound data
 3. Sensor module connection
 4. Board setting & Demonstration start
4. Function introduction
 1. Fire detection
 2. Other function introduction
5. Operation overview of Sound MCU
 1. Low power consumption design by intermittent operation
6. Appendix
 1. Switching audio devices
 2. Switching sound data
 3. Threshold setting for sensor module
 4. Threshold setting for detect low voltage
 5. Self test of Flash area (checksum)
7. Reference manual

Overview

1. Overview

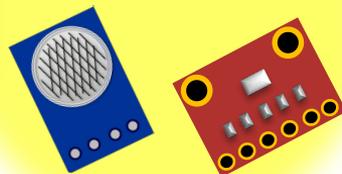
This reference design guide describes the demonstration software that reproduces the operation as a fire alarm by using the evaluation board of the Sound MCU “S1C31D41” and the sensor module.

S5U1C31D41T



Sound MCU “S1C31D41”
evaluation board

Sensor module

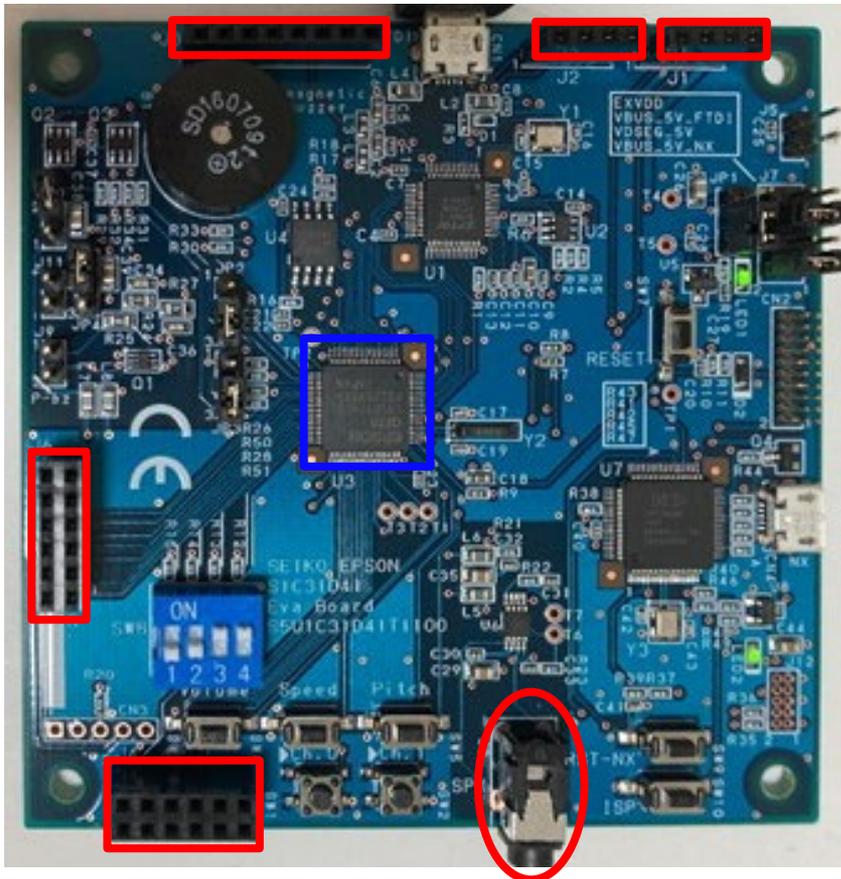


Fire alarm



Evaluation Board “S5U1C31D41T” and external Parts

2. Evaluation Board “S5U1C31D41T” and external Parts



Main parts of evaluation board

-  Sound MCU “S1C31D41”
-  Connector for sensor module
※With the UPMUX (Universal Port Multiplexer) function of EPSON MCU, each connector can be set by software for GPIO and each serial communication (SPI, I2C, etc.).
-  Audio jack for *speaker
*Default audio device
Please refer to Appendix to switch the audio device.

External parts

-  Gas sensor module with MQ-2
-  Temperature sensor module with BME280
-  LED

Startup

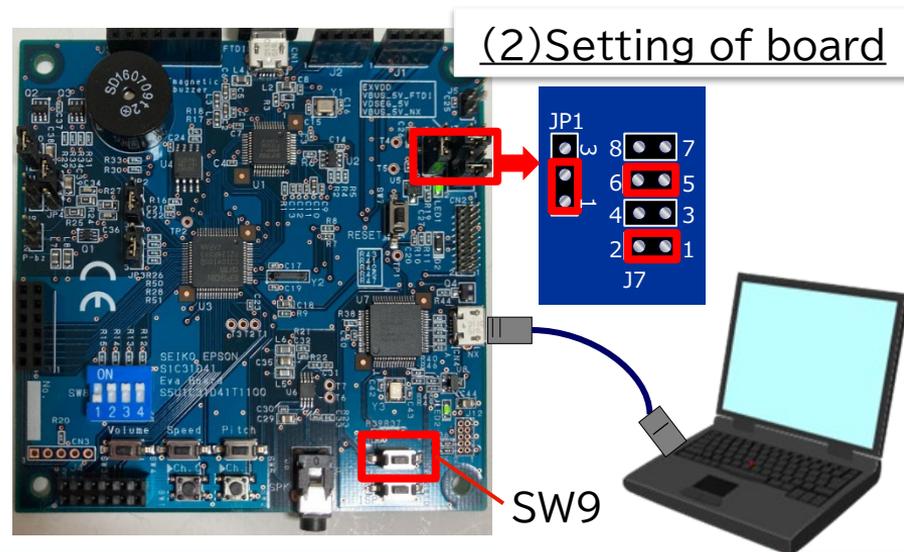
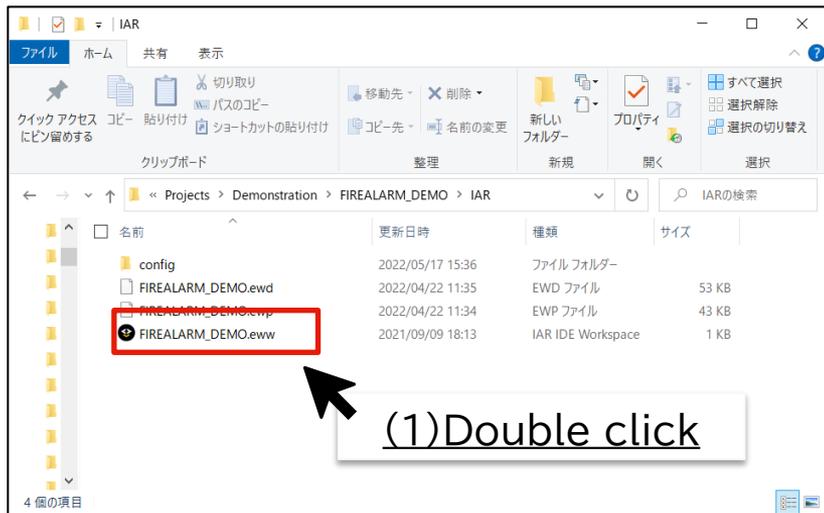
~From software writing to demonstration start~

3.1 Writing Software

(1) Double-click “FIREALARM_DEMO.eww” from sample software package (s1c31d5xd41sp_ver5_00).

Folder path : s1c31d5xd41sp_ver5_00¥Projects¥Demonstration¥FIREALARM_DEMO¥IAR

(2) After setting the jumper as shown in the figure below, connect the PC to the evaluation board and push SW9.



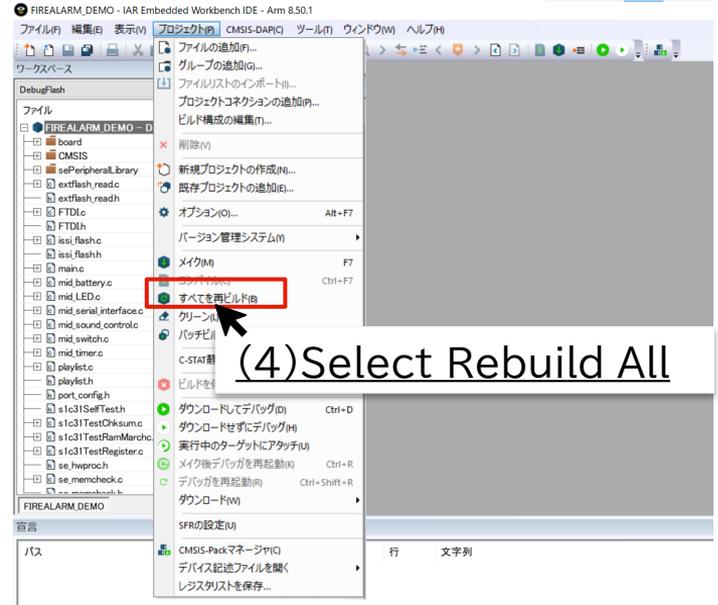
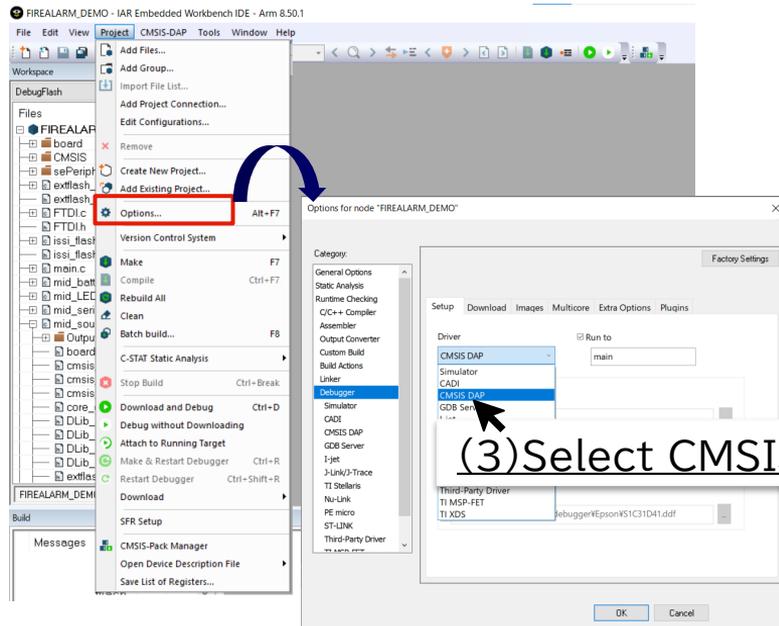
3.1 Writing Software

(3) When EWARM starts up, change the debugger to CMSIS DAP.

Project → Options → Debugger → Driver → CMSIS DAP

(4) Execute “Rebuild All”

Project → Rebuild All



3.1 Writing Software

(5) Execute “Download and Debug”. If the writing is completed normally, the debug screen will be displayed automatically.

The screenshot shows the IAR Embedded Workbench IDE interface. The 'Project' menu is open, and the 'Download and Debug' option is highlighted with a red box. A blue arrow points from this option to the 'Busy' dialog box in the center, which displays 'Starting debugger session Flashing' and 'Programming flash memory' with a progress bar. A second screenshot on the right shows the IDE with the cursor positioned at the beginning of the 'int main(void)' line in the main.c file, with a blue arrow pointing to it from the text 'Confirm that the cursor is coming at the beginning of the main statement'.

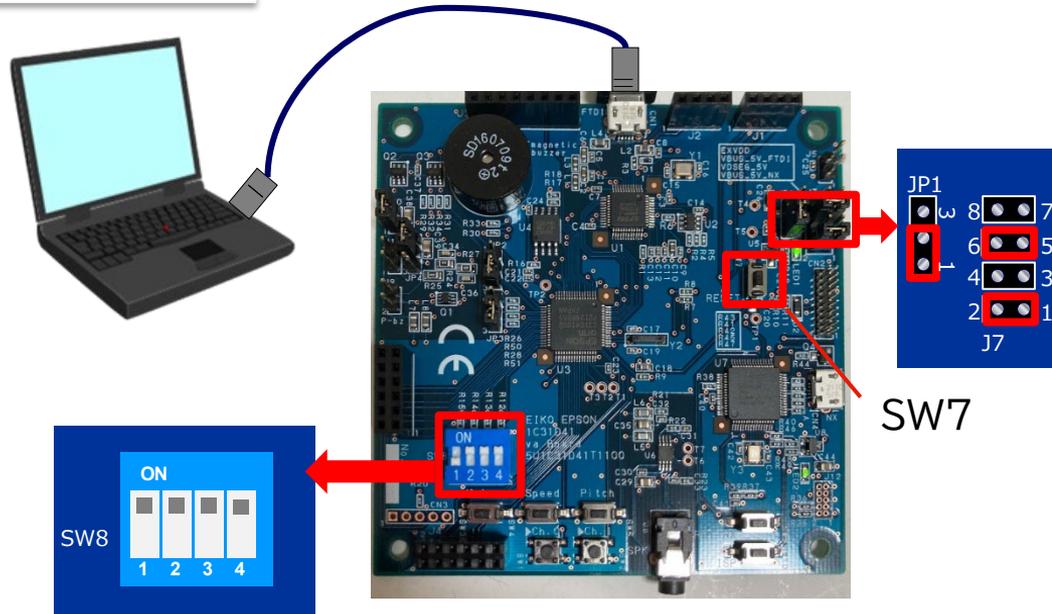
(5) Select Download and Debug

Confirm that the cursor is coming at the beginning of the main statement

3.2 Writing sound data

(1)As shown in the figure below, after setting the jumper and DIP switch, connect the PC to the evaluation board and push SW7.

(1)Setting of board

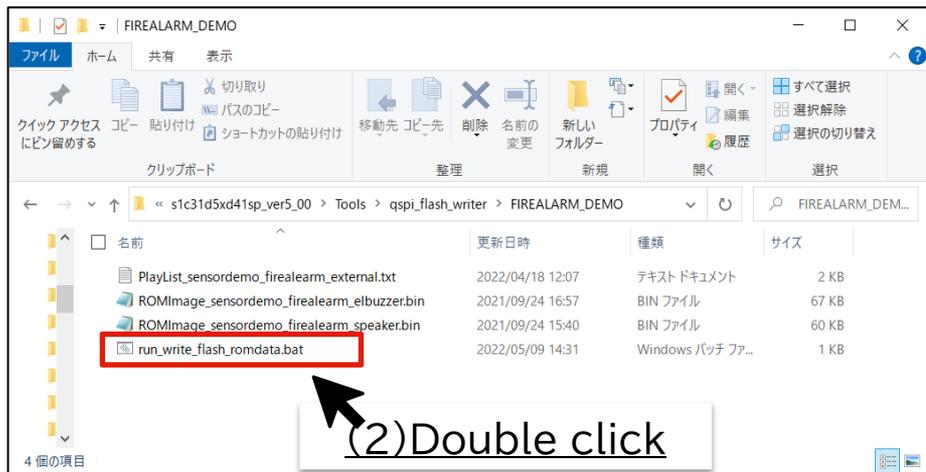


3.2 Writing sound data

(2) Double-click “run_write_flash_romdata.bat” from the sample software package (s1c31d5xd41sp_ver5_00).

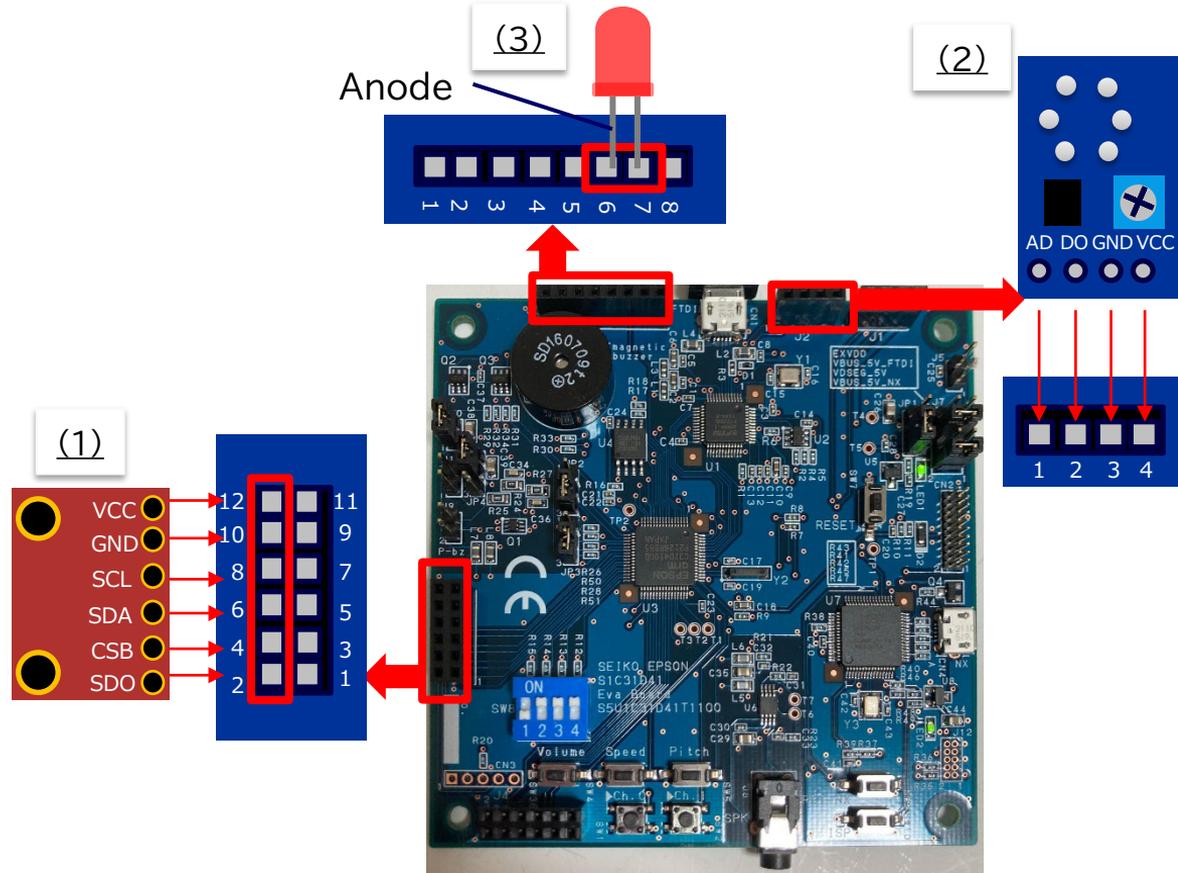
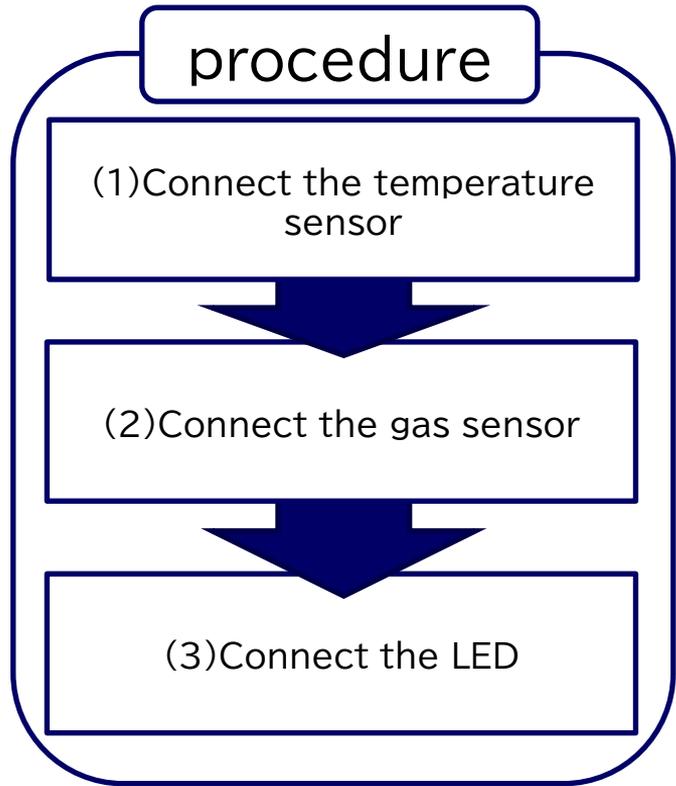
Folder path : s1c31d5xd41sp_ver5_00¥Tools¥qspi_flash_writer¥FIREALARM_DEMO

(3) As shown in the lower right figure, when three “Serial Flash write complete normally” are displayed, the sound data writing is complete.



(3) Writing is completed when displayed “SerialFlash write complete normally”

3.3 Sensor module connection



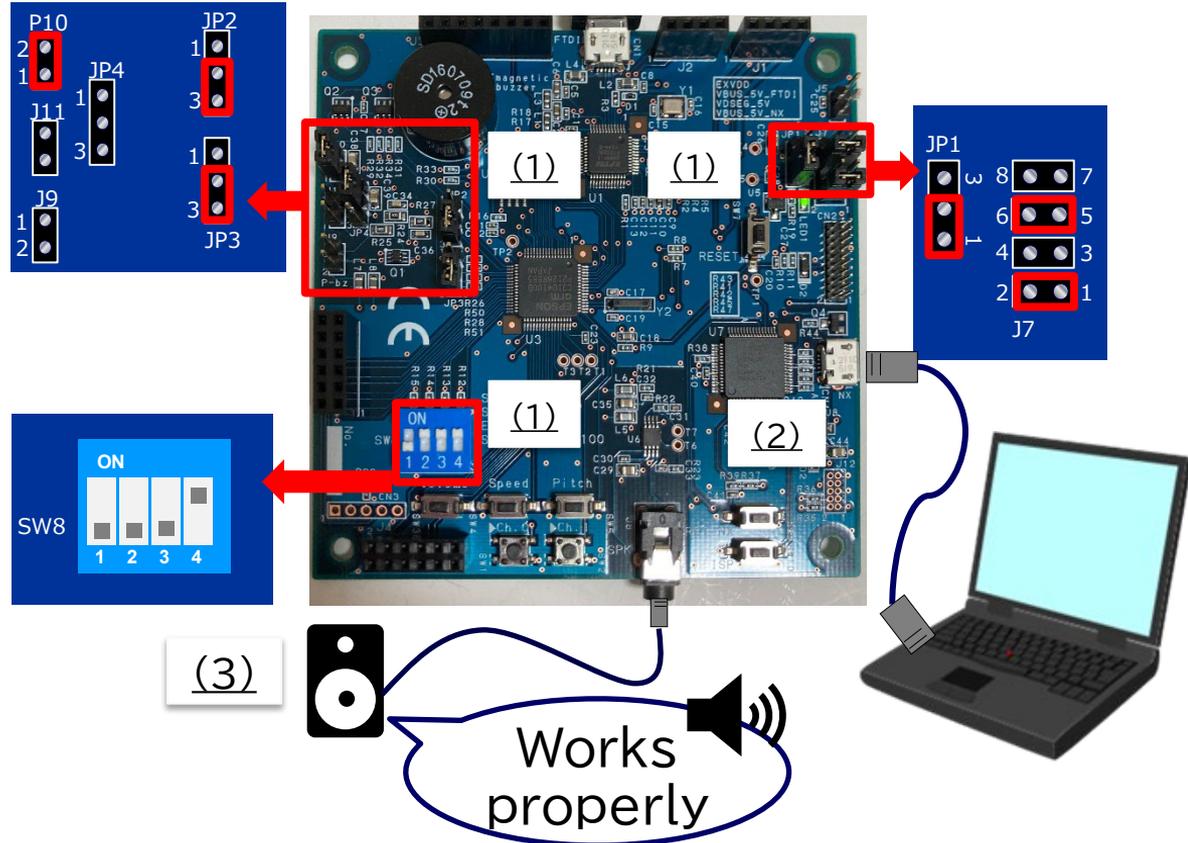
3.4 Board setting & Demonstration start

procedure

(1) Set the jumper and DIP switch of the evaluation board according to the figure.

(2) Connect the USB cable and Demonstration starts when the power is turned on

(3) Startup is complete when you hear a sound from the speaker saying "Works properly".



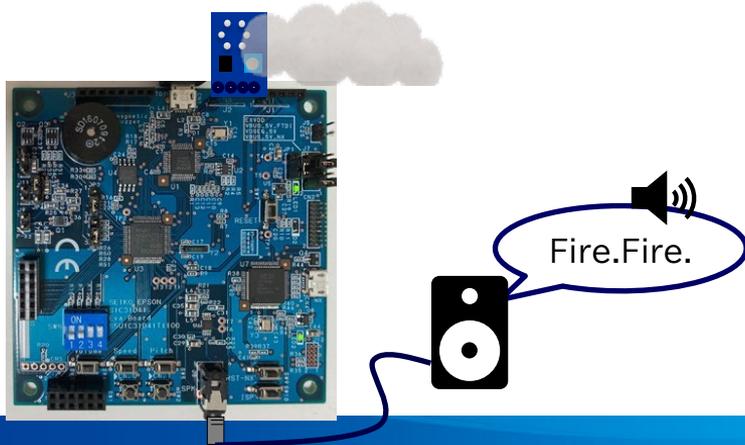
Function introduction

4.1 Fire detection

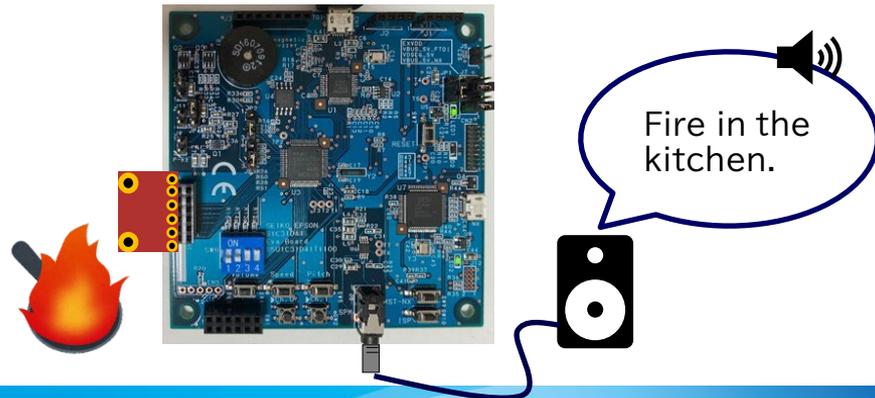
Each sensor module plays audio when it detects gas or heat. Sensing is performed even during playing, and if gas and heat is no longer detected, playing and LED flashing is finished. Audio is played even though the temperature sensor is not connected.

Target	Sentence	Play pattern	LED pattern
Gas	Fire.Fire.	Repeat	Blinking
Heat	Fire in the kitchen.	Repeat	Blinking
No connect temperature sensor	Unable to measure temperature correctly.	Repeat	Blinking

- Image of gas detection



- Image of heat detection



4.2 Other function introduction

- Detects low voltage of power supply

Plays audio when detect low voltage. If the power restored, audio play and LED flushing is finished.

Target	Sentence	Play pattern	LED pattern
Power supply voltage (threshold value of default:2.7V)	Battery low.	Repeat(10sec interval)	Lighting

- Self test

Self test of memory is performed, and the test result is notified playing audio and LED.

Self test is performed automatically when the power is turned on or when the power is reset.

Test result	Sentence	Play pattern	LED pattern
OK	Works properly	One shot	
NG			Blinking

- Stop playing

Push and hold SW1 on the evaluation board for 1 second during playing audio to stop playing. If you push SW1 again after stopping, audio will resume if there is an audio play factor (ex: Power supply voltage is lower power than threshold value). If there is no audio play factor, audio play will finish.

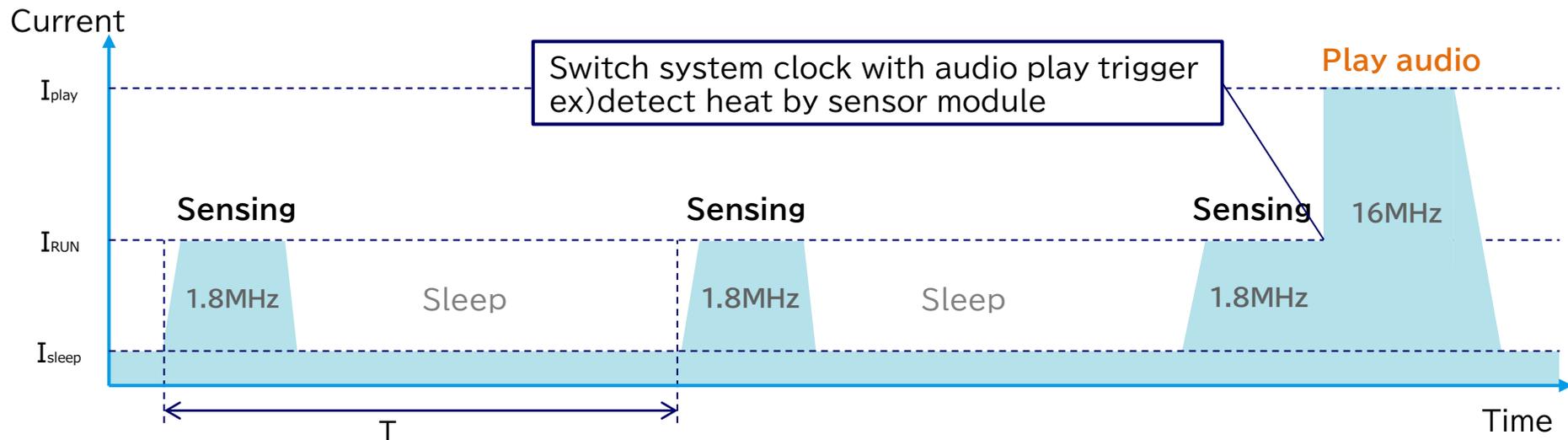
- Reset of software

With no audio playing, push and hold SW1 on the evaluation board for 1 second to reset the software.

Operation overview of Sound MCU (S1C31D41)

5.1 Intermittent operation for low power

By switching the internal system clock of the Sound MCU, efficient intermittent operation is implemented and power is reduced. It operates with a low-speed clock (1.8MHz) during sensing, and operates with a high-speed clock (16MHz) only during audio play.



I_{play} : Current when playing audio
 I_{RUN} : Current when sensing

I_{sleep} : Current when sleep
T : Intermittent operation cycle(1second)

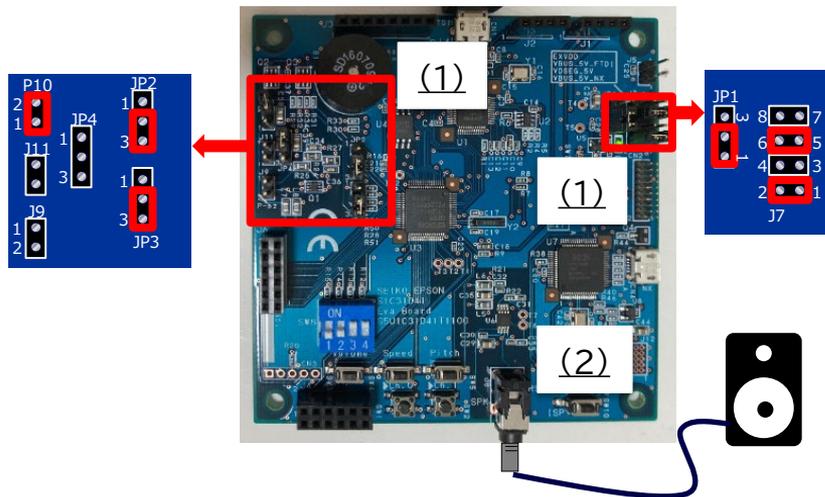
Appendix

6.1 Switching audio devices

With the evaluation board “S5U1C31D41T”, audio can be played from three devices: speaker, electromagnetic buzzer, and piezoelectric buzzer by setting the evaluation board and software.

① Speaker settings

Board setting



- (1) Set jumper
- (2) Connect speaker

Software setting

Enable “SPEAKER” defined in the header file “mid_sound_control.h” and comment out other devices

```
FIREALARM_DEMO - IAR Embedded Workbench IDE - Arm 8.50.1
File Edit View Project CMSIS-DAP Tools Window Help
Workspace main.c mid_sound_control.h x
DebugFlash
Files
  main.c
  mid_battery.c
  mid_LED.c
  mid_serial_interface.c
  mid_sound_control.c
  board.h
  cmsis_compiler.h
  cmsis_iccarm.h
  cmsis_version.h
  core_cm0plus.h
  DLib_Config_Normal.h
  DLib_Defaults.h
  DLib_Product.h
  DLib_Product_string.h
  exiflash_read.h
  FTDI.h
  iccarm_builtin.h
  issi_flash.h
  mid_sound_control.h
  mid_switch.h
  mpu_armv7.h
  playlist.h
  PlayList_Internal.h
  ROMImage.h
  S1C31.h
63 // total number of phrases
64 #define NUMBER_OF_PHRASES 6
65
66 // sound storage
67 #define EXTERNAL_FLASH_WRITE 0xF
68
69 // language
70 #define JAPANESE 0x0
71 #define ENGLISH 0x1
72
73 // sound output device
74 #define SPEAKER 0
75 //#define ELBUZZER 1
76 //#define PIBUZZER 2
77
78 // Structures for soundplay
```

6.1 Switching audio devices

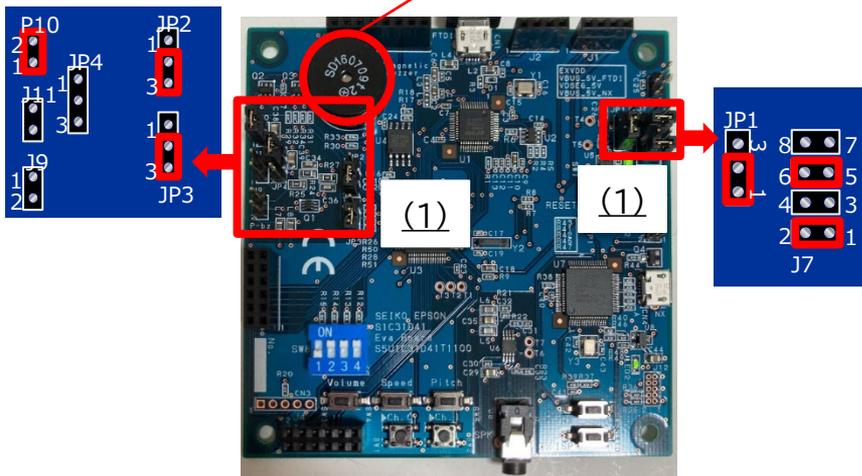
② Electromagnetic buzzer settings

Board setting

Software setting

Electromagnetic
buzzer

Enable “ELBUZZER” defined in the header file
“mid_sound_control.h” and comment out other devices



(1) Set jumper

```
main.c mid_sound_control.h
53 // total number of phrases
54 #define NUMBER_OF_PHRASES 6
55
56 // sound storage
57 #define EXTERNAL_FLASH_WRITE 0xF
58
59 // language
60 #define JAPANESE 0x0
61 #define ENGLISH 0x1
62
63 // sound output device
64 // #define SPEAKER 0
65 #define ELBUZZER 1
66 // #define PIBUZZER 2
67
68 // Structures for soundplay
```

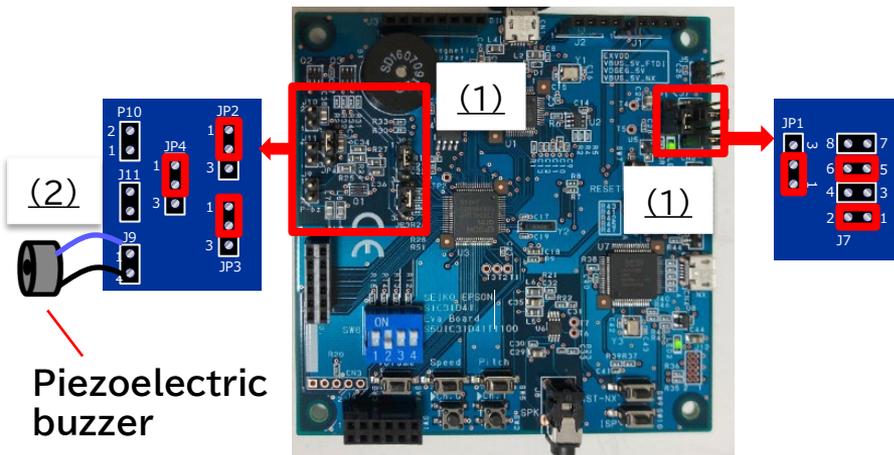
6.1 Switching audio devices

③ Piezoelectric buzzer settings

Board setting

Software setting

Enable “PIBUZZER” defined in the header file “mid_sound_control.h” and comment out other devices



Piezoelectric buzzer

- (1) Set jumper
 - (2) Connect Piezoelectric buzzer to board
- ※Refer to the “S5U1C31D41T Manual” for the settings when supplying power to the piezoelectric buzzer from external.

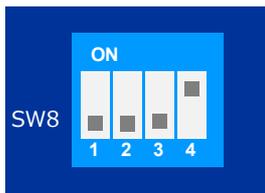
```
main.c mid_sound_control.h x
DebugFlash
Files
- main.c
- mid_battery.c
- mid_LED.c
- mid_serial_interface.c
- mid_sound_control.c
- Output
- board.h
- cmsis_compiler.h
- cmsis_iccarm.h
- cmsis_version.h
- core_cm0plus.h
- DLib_Defaults.h
- DLib_Product.h
- DLib_Product_string.h
- exifflash_read.h
- FTDI.h
- iccarm_builtin.h
- issi_flash.h
- mid_sound_control.h
- mid_switch.h
- mp3_snmv7.h
- playlist.h
- PlayList_Internal.h
- ROMImage.h
- S1C31.h
63 // total number of phrases
64 #define NUMBER_OF_PHRASES 6
65
66 // sound storage
67 #define EXTERNAL_FLASH_WRITE 0xF
68
69 // language
70 #define JAPANESE 0x0
71 #define ENGLISH 0x1
72
73 // sound output device
74 // #define SPEAKER 0
75 // #define ELBUZZER 1
76 #define PIBUZZER 2
77
78 // Structures for soundplay
```

6.2 Switching sound data

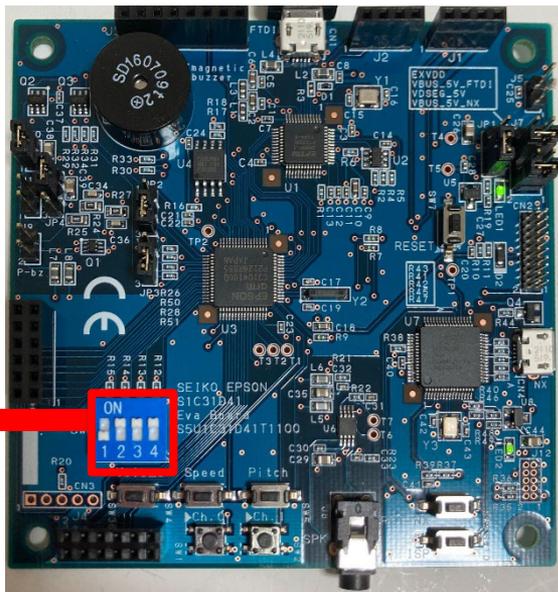
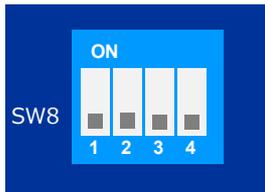
① Select language

Language can be switched by setting the DIP switch (SW8).

English



Japanese



② Switching memory for sound data

Flash memory can be switched by “EXTERNAL_FLASH” defined in the header file “mid_sound_control.h”

- Using SPI-Flash on board

```
mid_sound_control.h
16 * @}
17 */
18 #include "se_common.h"
19 #include "playlist.h"
20
21 // select sound flash
22 #define EXTERNAL_FLASH
23
24 // parameter for QSPI control
25 #define EX_MEMORYMAP_START_ADDRESS 0x00040000
26 #define QSPI_RMDR_MASK_L 0x000FFFFFF
27 #define QSPI_RMDR_MASK_H 0xFFFF0000
28 #define PLAYLIST_ADDR_IN_EXTFLASH 0x700000
```

- Using internal flash of the Sound MCU “S1C31D41”

```
mid_sound_control.h
16 * @}
17 */
18 #include "se_common.h"
19 #include "playlist.h"
20
21 // select sound flash
22 // #define EXTERNAL_FLASH
23
24 // parameter for QSPI control
25 #define EX_MEMORYMAP_START_ADDRESS 0x00040000
26 #define QSPI_RMDR_MASK_L 0x000FFFFFF
27 #define QSPI_RMDR_MASK_H 0xFFFF0000
28 #define PLAYLIST_ADDR_IN_EXTFLASH 0x700000
```

*"3.2 Writing sound data" cannot be executed while using the internal flash.

6.3 Threshold setting for sensor module

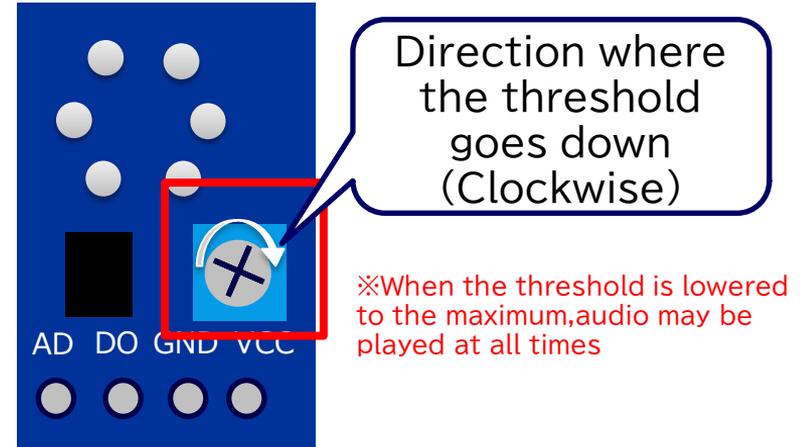
① Temperature sensor module

Change the value defined in the header file “sensor_control.h” and set the temperature threshold that triggers for the audio play. In the case of the figure below, audio play is executed more than 60 ° C.

```
16  * @}
17  */
18  #include "board.h"
19
20  #define TEMP_THRESHOLD 60.00
21
22  #define TEMP_SENSOR_FLAG 1
23  #define SMOKE_SENSOR_FLAG 2
24  #define SENSOR_ERROR_FLAG 3
25
26  // smoke sensor digital input port
27  #define SMOKE_SENSOR_PORT sePPORT_P35
28
29  // SPI error data
30  #define ERROR_DATA 0x2FD
```

② Gas sensor module

Change the gas concentration setting by adjusting the red frame in the figure below. Turning it clockwise lowers the gas concentration threshold that triggers for the audio play, and turning it counterclockwise raises the threshold.



6.4 Threshold setting for detect low voltage

Change the value defined in the header file “mid_battery.h” to set the threshold value for detecting the low power supply voltage that triggers audio play. For the relationship between the set value and the detected voltage value, refer to the S1C31D41 manual.(23.8 Supply Voltage Detector (SVD3) Characteristics)

```
11  * Copyright(C) SEIKO EPSON CORPORATION 2018-2019. All rights reserved.
12  *****
13  */
14
15  /**
16   * @}
17   */
18
19  #define LOWPOWER_THRESHOLD 0x09
20  #define BATTERY_FLAG 1
21
22  extern volatile unsigned short battery_flag;
23
24  /**
25   * @brief Initialize SVD3 function.
26   * @param low threshold:This value is threshold for low bat
```

6.5 Self test of Flash area (checksum)

The self test function can be expanded (checksum in the Flash area) by setting the software. The target Flash area is the internal flash and the SPI-Flash on board. * If the audio Flash memory is set to the internal flash, the SPI-Flash does not perform self test.

* Refer to “6.2 Switching sound data” in this document.

- (1) Delete comment out(//) of “MEMCHECK_FLASH” defined in the source file “main.c” to enable it.
- (2) Start a debug session and follow the steps below to open a register window.

View→Registers→Register 1

- (3) After the register window opens, select “MEMCHECK” from the tab (Group) in the red frame in the figure.

```
36 #include "mid_timer.h"
37 #include "mid_sound_control.h"
38 #include "test_memcheck.h"
39
40 // for OSC3 trimming
41 // #define RUN_TRIMMING
42
43 // for flash check in selftest
44 #define MEMCHECK_FLASH
45
46 // for reset
47
48
49
50 #define MODE0 1
51 #define MODE1 0
52
53 // flag
```

(1) Delete “//” to enable

Register window

(2) Open register window

(3) Select MEMCHECK

Name	Value
MEMCHECK	0x0003

ERROR	0x0000
STATES	0x0001
RESULT	0x00000000
VERSION	0x0001

6.5 Self test of Flash area (checksum)

- (4)As shown in the figure below, set two breakpoints in the source file “main.c” and execute the program.
- (5)Check the register value “RESULT” at the two breakpoints when it is stopped, and rewrite the initial value of the variable in the figure below.

Stop at the top of the breakpoint in the figure below → Check the checksum of the SPI-Flash on the board from “RESULT” and change the initial value of the variable.

Stop at the bottom of the breakpoint in the figure below → Check the checksum of the internal flash from “RESULT” and change the initial value of the variable

- ※If it is not stopped at the breakpoint, it is not necessary to rewrite the initial value.
- ※If enabled self-test in the Flash area and changed the code or audio data, please rewrite the initial value each time.

(4) Set break point (For SPI-Flash on the board)

(4) Set break point (For internal flash)

(5) Set the value displayed in RESULT at each breakpoint to the initial value of the variable for the checksum of each Flash.

- Initial value of checksum variable (checksum_exflash) for SPI-Flash on board
- Initial value of checksum variable (checksum_flash) for internal flash

The screenshot shows the IAR Embedded Workbench IDE with the following components:

- Source Code (main.c):** Lines 610-633 show the `startSelfTest()` function. Breakpoints are set at line 610 (top of the `if` statement) and line 630 (bottom of the `if` statement).
- Registers Window:** Shows the `RESULT` register with a value of `0x0000141F`.
- Memory Window:** Shows the `MEMCHECK` group with values `0x0000141F` and `0x0391`.
- Source Code (main.c):** Lines 61-67 show the `checksum_exflash` variable definition with an initial value of `0xA14F`. Lines 68-76 show the `checksum_flash` variable definition with an initial value of `0x09C6`.

7. Reference manual

- Reference manual

Reference information	Document	Location
Sound MCU 「S1C31D41」	<ul style="list-style-type: none">• S1C31D41 Technical Manual	HP
Evaluation board 「S5U1C31D41T」	<ul style="list-style-type: none">• S5U1C31D41T Manual	
Writing software	<ul style="list-style-type: none">• S1C31 Family Software Development Setup Guide• S1C31 Family Sample Software Manual	

EPSON
EXCEED YOUR VISION