FUJITSU

AN706-00040-2v0-E

32-BIT MICROCONTROLLER
FM3 family Application Note

Simple AV System Board

User Manual

The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.

The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of Fujitsu semiconductor device; Fujitsu does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. Fujitsu assumes no liability for any damages whatsoever arising out of the use of the information.

Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of Fujitsu or any third party or does Fujitsu warrant non-infringement of any third-party's intellectual property right or other right by using such information. Fujitsu assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).

Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.

Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Revision History

Rev	Date	Remark
1.0	Aug.23,2011	First Edition
2.0	Feb.06,2012	Correction format Correction lineup of FM3 Correction by RoHS c compliant for board, parts change, and software change

Table of Contents

Target products

This application note is described about below products;

(TYPE0)

Series	Product Number (not included Package suffix)
MB9B500B	MB9BF504NB,MB9BF505NB,MB9BF506NB MB9BF504RB,MB9BF505RB,MB9BF506RB
MB9B300B	MB9BF304NB,MB9BF305NB,MB9BF306NB MB9BF304RB,MB9BF305RB,MB9BF306RB

1 INTRODUCTION

This user manual contains specifications and information on how to use the simple AV system board.

2 OVERVIEW OF SIMPLE AV SYSTEM BOARD

The simple AV system board plays the following 2 types of music media files and performs 1 type of image output. Music media files are however not played simultaneously; The format of played music file is set in accordance with compile options.

① MP3/JPEG

JPEG files in a USB memory are imported by the file system, and the data decoded by the JPEG encoder/decoder are output and displayed on the LCD.

The MP3 file to be played is selected on the touch panel. It is then imported from the USB memory by the file system, and is played by outputting the data decoded by the MP3 decoder to the DAC. You can select and play MP3 files by switch operation as well.

② AAC

The AAC file to be played is selected by switch operation and the AAC file is imported from the USB memory by the file system. The imported AAC file is decoded by AAC decoder and the data is output to the DAC and played.

If playing an AAC file, image output to the LCD and touch panel control do not work.

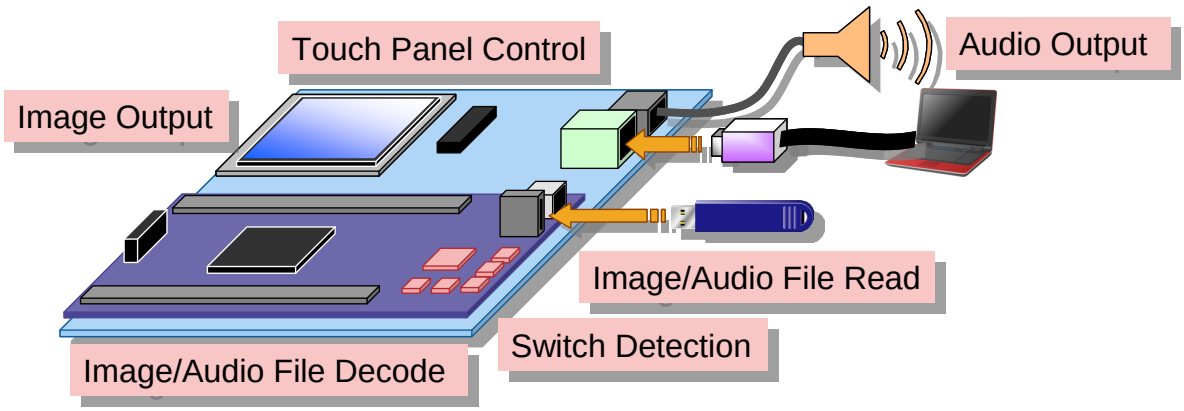

Figure 1 System Overview

3 PREPARATIONS

3.1 System Items

A list of system items of the simple AV system board is given in Table 1, a photograph the external appearance of the AV system board is shown in Figure 2, a photograph the external appearance of ICE is shown in Figure 3, a photograph of the external appearance of the USB memory is shown in Figure 4 and a photograph the external appearance of the USB cable is shown in Figure 5.

Table 1 List of System Items

No.	Name	Pcs.	Remarks
1	Simple AV System Board	1	Consists of microcontroller board (KEIL MCB9BF500), LCD board and power supply cable
2	2 Pin power connection line	1	2.54mm pitch
3	ICE	1	KEIL ULINK2
4	USB memory	1	BUFFALO RUF-C2GS-BL/U2
5	USB cable	1	ELECOM U2C-B07BK

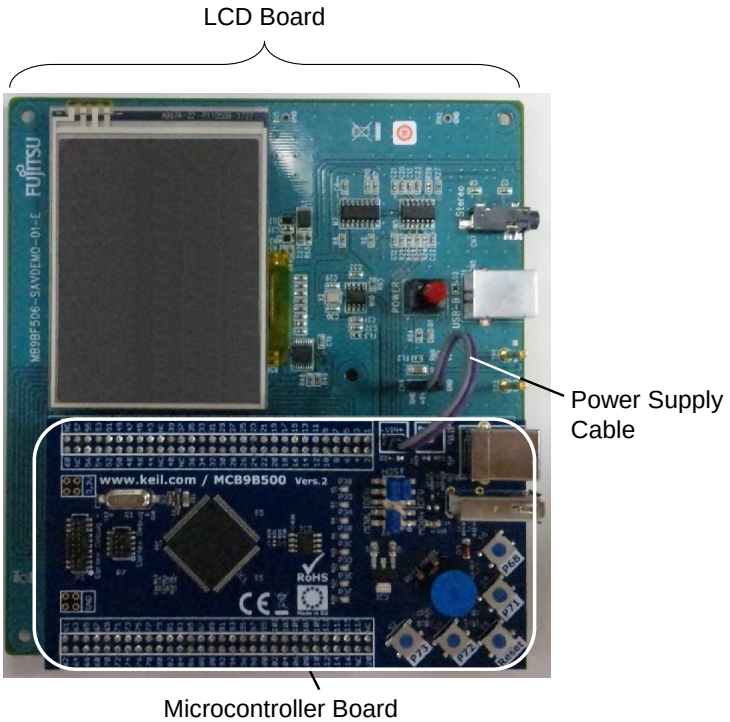

Figure 2 Simple AV System Board

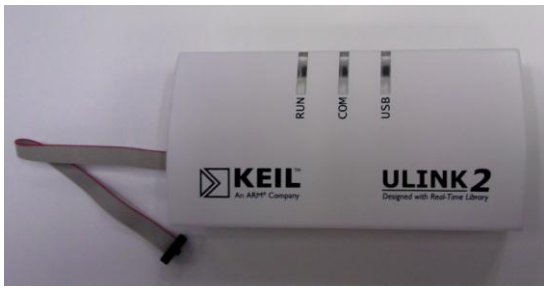

Figure 3 ICE

Figure 4 USB memory

Figure 5 USB cable

3.2 Equipment Other than System Item Required

A list of equipment other than system items required is given in Table 2, and equipment used as need is given in Table 3.

Table 2 Equipment other than system items required

No.	Name	pcs.	Specifications / Remarks
1	PC	1	Use USB host port for power supply. If using ICE, use USB host port.

Table 3 List of Equipment Used as Needed

No.	Name	pcs.	Specifications / Remarks
1	Earphones or speakers	1	For audio output

3.3 Board Appearance

A photograph of the external appearance of the simple AV system board is shown in Figure 6.

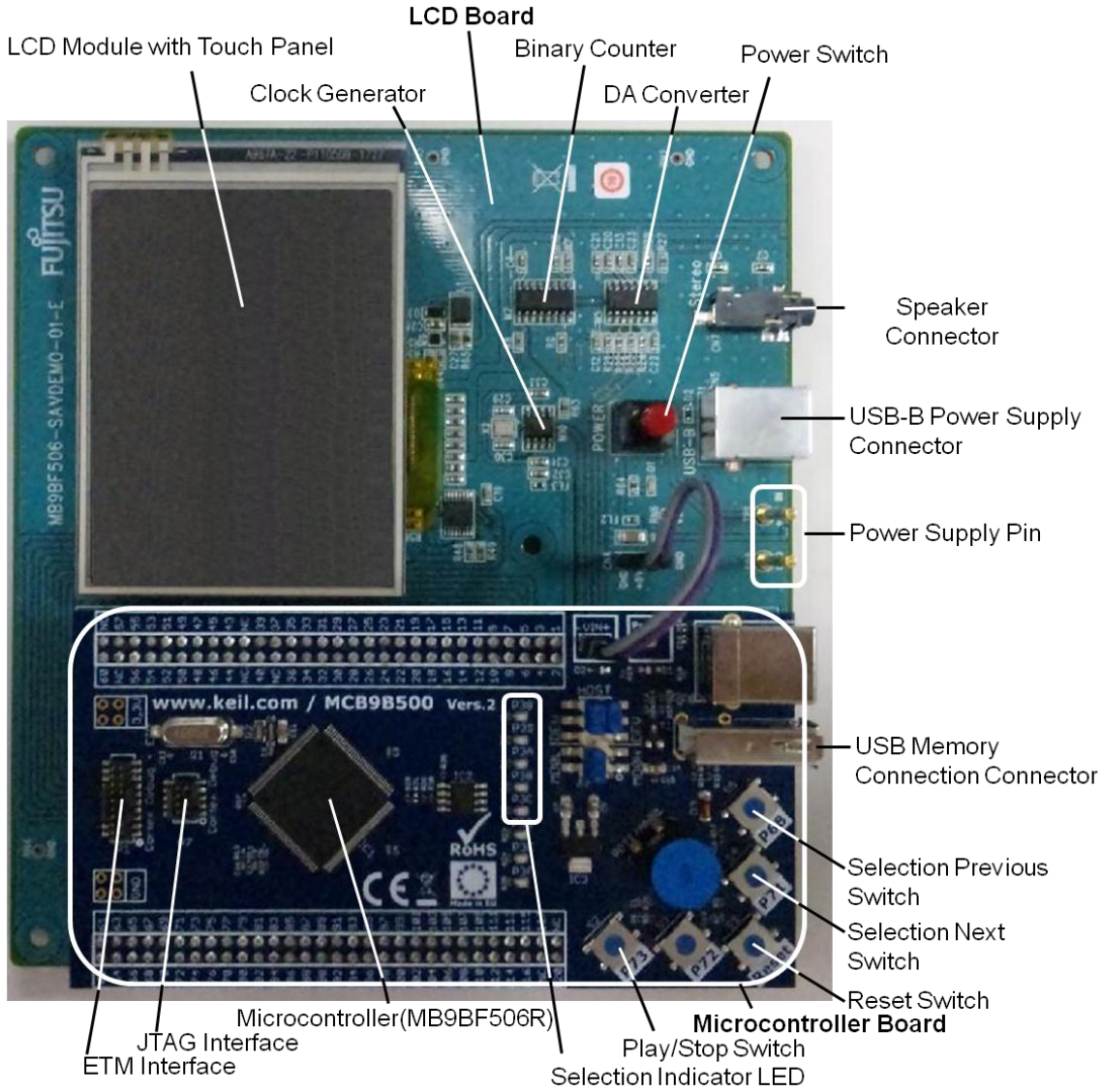

Figure 6 Photograph of External Appearance of Simple AV System Board

3.4 Power Supply Method

USB bus power of the PC is used as the power supply for the simple AV system board.

Connect the Type A side of the USB cable (with Type A - Type B connector) to the USB port of the PC and connect the Type B side to the USB power supply connector of the simple AV system board.

Press down the power switch with the USB cable connected. The power supply switch is self-locking and stays depressed while the power is on. To turn the power off, press the power switch again.

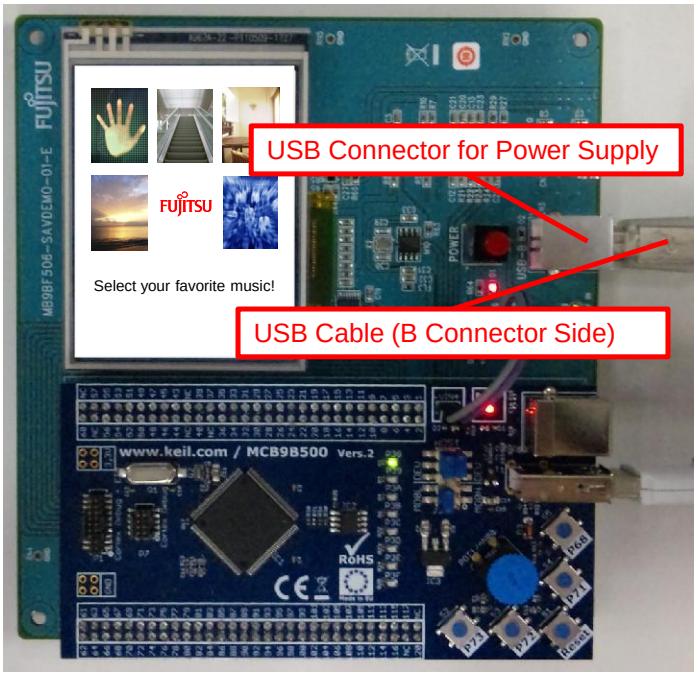

Figure 7 USB Cable Insertion Position for Power Supply

4 PROGRAM EXECUTION METHOD

4.1 Program Execution Using Debugger

4.1.1 Activation of KEIL Integrated Development Environment

Double-clicking “AV_demo.uvproj” inside the Project folder activates the KEIL integrated development environment and opens the simple AV system project.

(1) Designation of Compile Options (MP3 and AAC Switch)

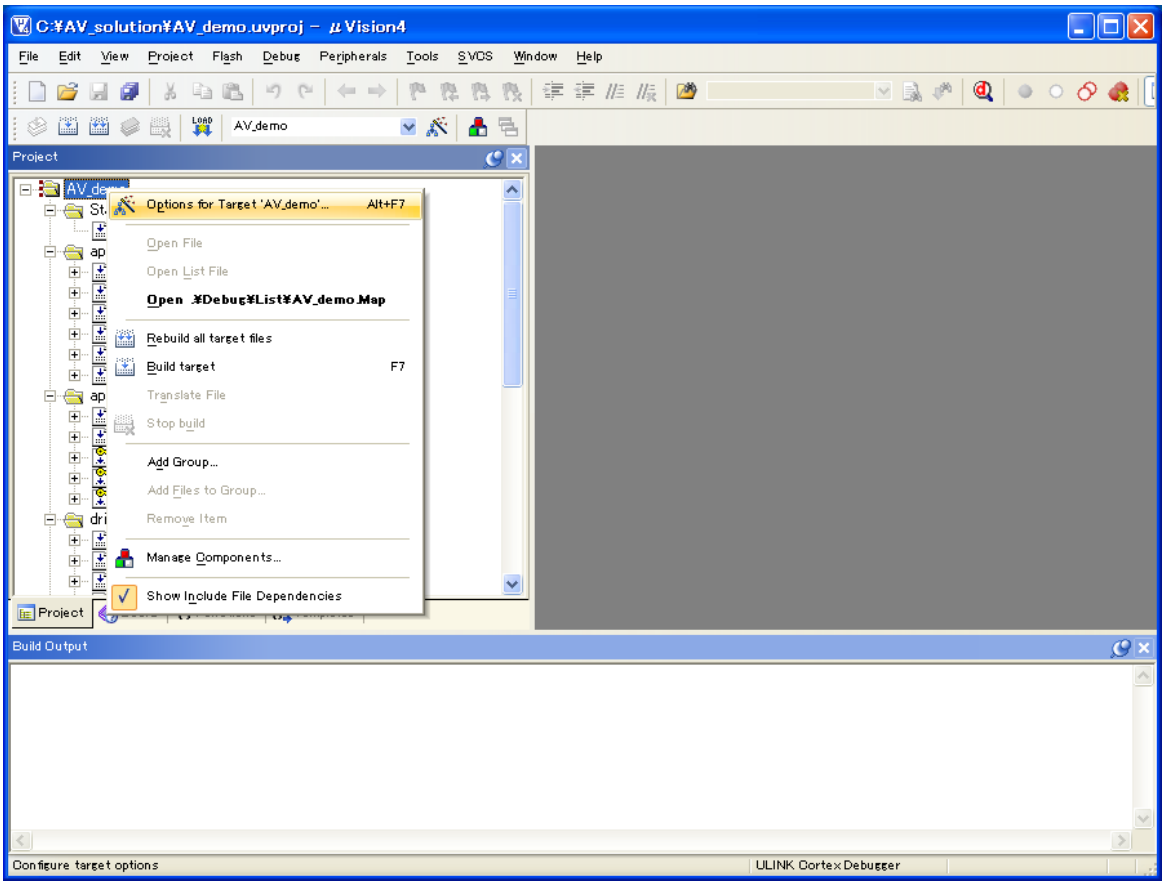

Figure 8 Designation Operations of Compile Options

As shown in Figure 8, right-clicking on “AV_demo” of the project and then left-clicking “Options for Target ‘AV_demo’…” opens the option settings screen.

In the case of MP3, set “MUSIC_MP3” for “Define” of the “C/C++” and “Asm” tabs of the option screen; in the case of AAC, set “MUSIC_AAC”. (Figure 9, Figure 10)

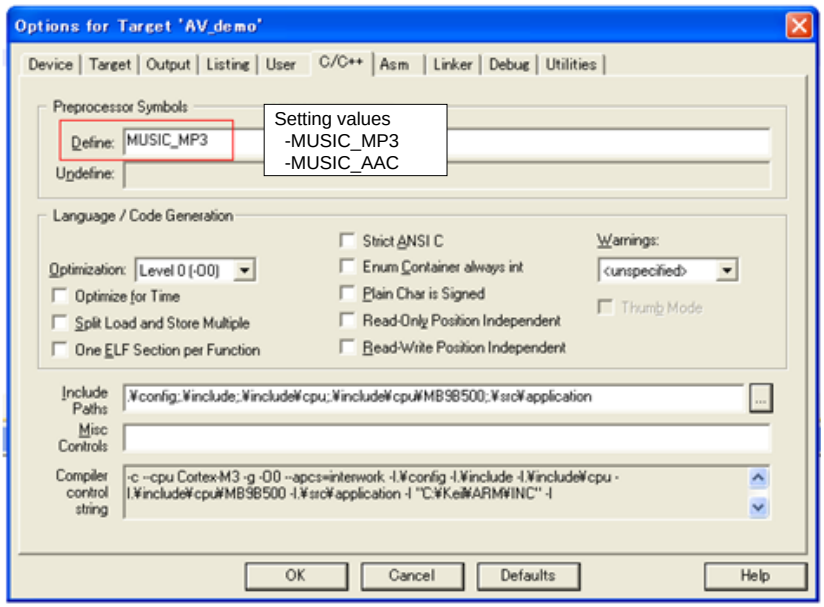

Figure 9 C/C++ Compile Options Designation Method

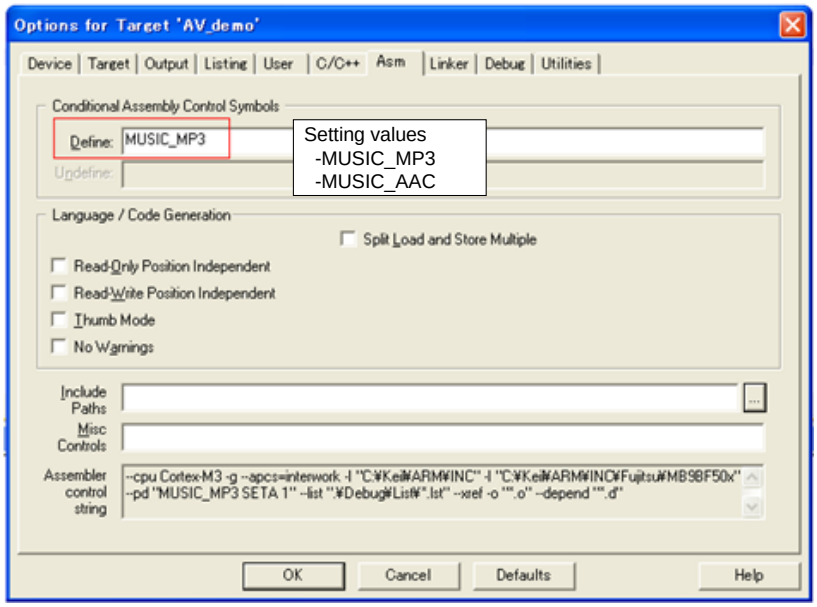

Figure 10 ASM Compile Options Designation Method

(2) Program Build

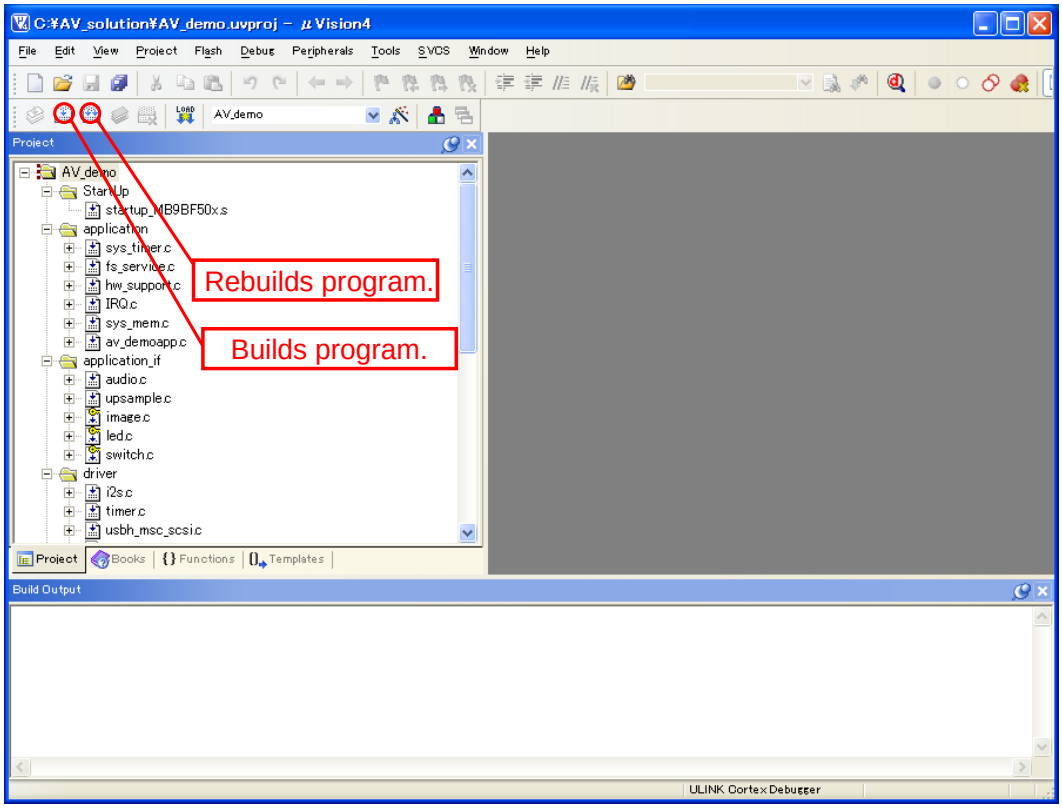

Figure 11 Simple AV System Build

Click the location shown in Figure 11 to build or rebuild the program. After successfully building or rebuilding, the sample program can be exported to the microcontroller.

4.1.2 Start of Sample Program Export and Debug

(1) ICE Connection

The equipment connection diagram for exporting the program to the microcontroller is shown in Figure 12.

The PC and simple AV system board are connected via ICE.

Figure 12 Equipment Connections for Exporting Program

(2) Exporting the Program to the Microcontroller

Click the location shown in Figure 13 to export the program to the microcontroller.

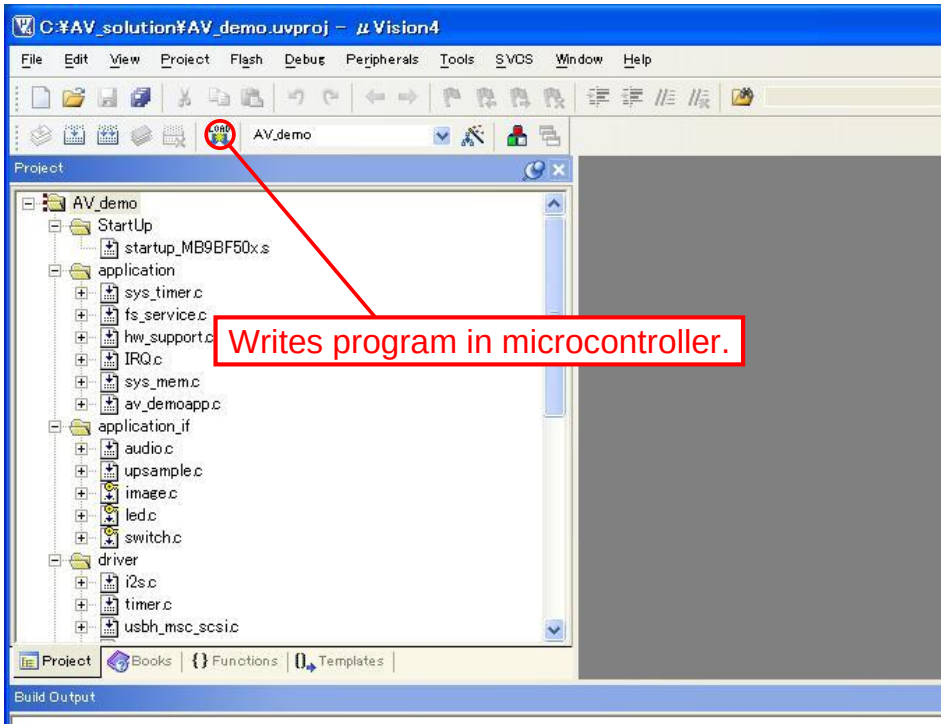

Figure 13 Exporting the Program to the Microcontroller

(3) Debug Activation

Click the location shown in Figure 14 and activate the debugger. Debugging can then be started.

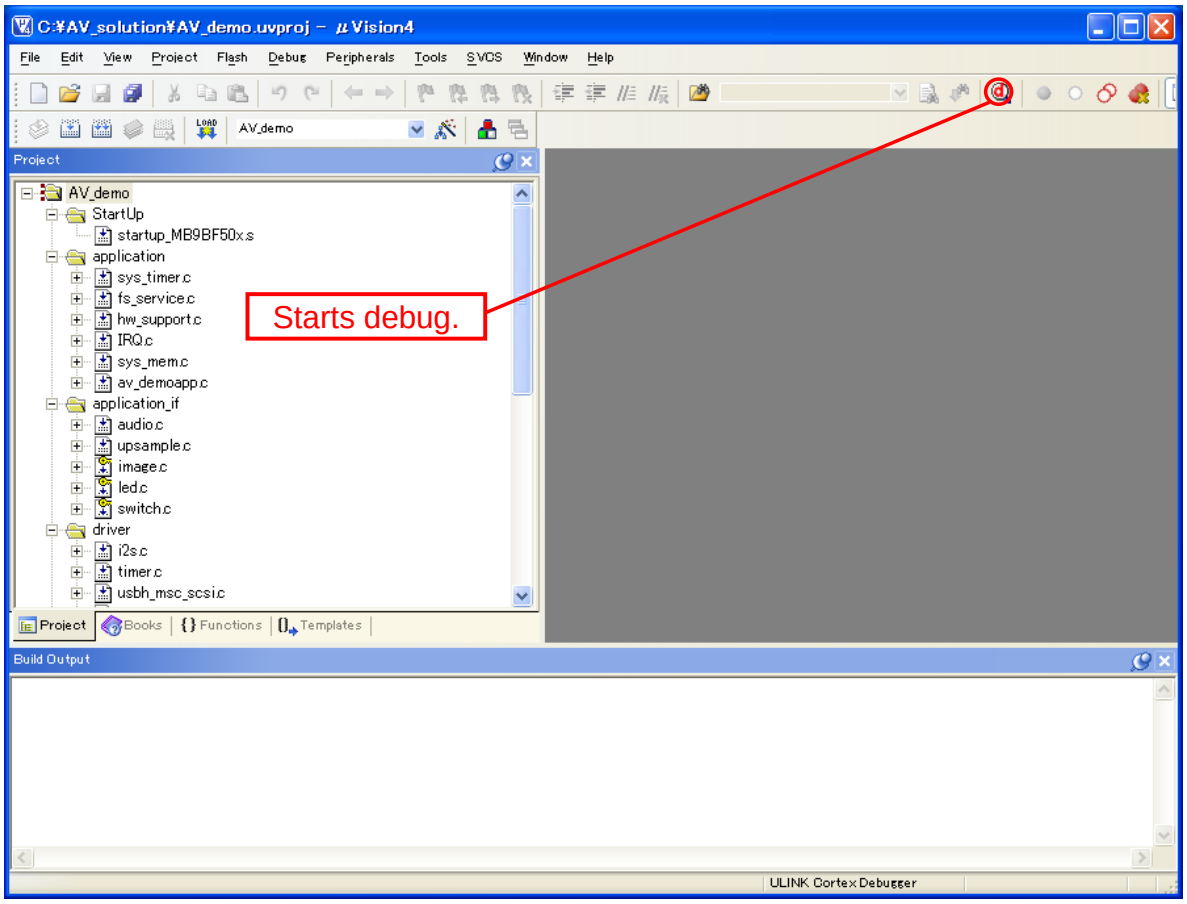

Figure 14 Operation of Debug Activation

4.1.3 Program Execution Using Debugger

When the debugger is activated, the screen appears as shown in Figure 15. Press “Execute” to start executing the program.

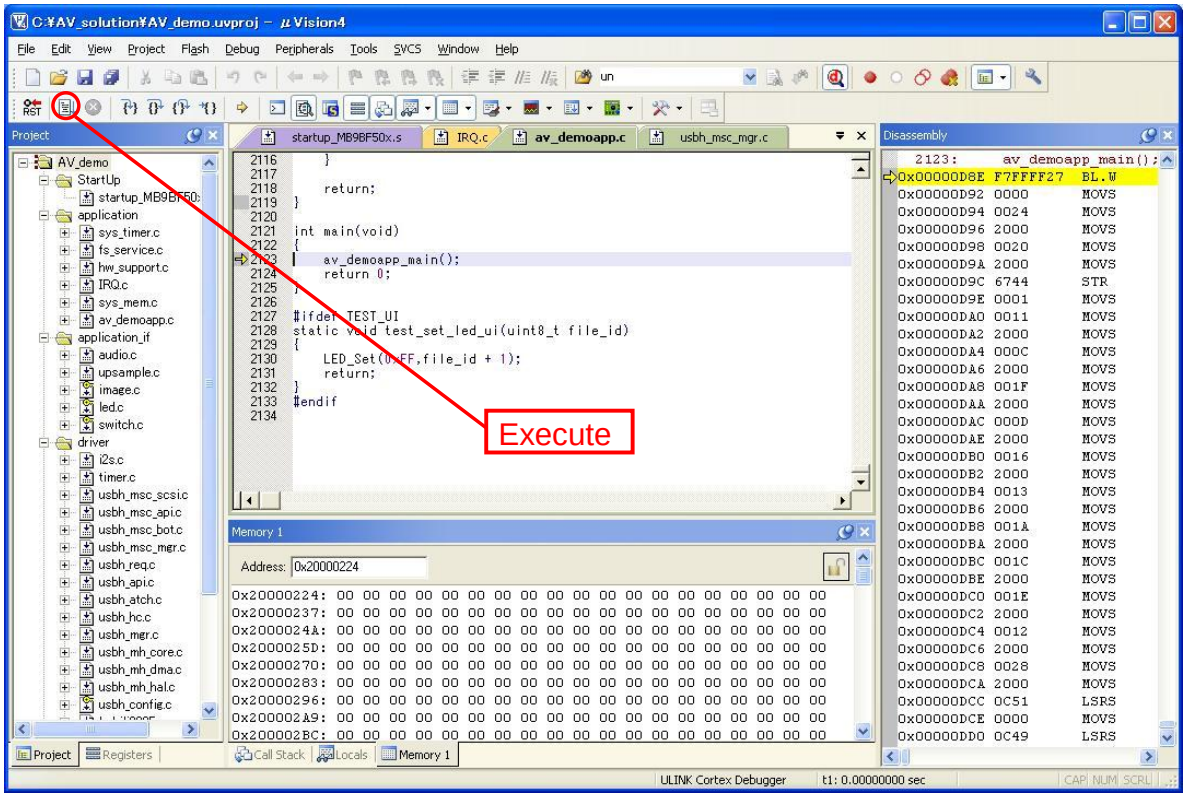

Figure 15 Screen When Debugger is Activated

For other debugger operations, see help.

4.2 Standalone Program Execution

When the program is exported and debugging is completed by the procedure given in “4.1 Program Execution Using Debugger,” turn the board's power off with the power switch and remove ICE from the simple AV system board.

After doing so, the program is run by standalone by turning the power on with the power switch on the simple AV system board.

5 SUPPORTED MEDIA

5.1 Formats that can be Used for USB Memory

The USB memory formats that can be used by the file system are given in Table 4. If using a file system (middleware), a separate contract is required.

Table 4 Recognition Media Capacity and Applicable Formats

No.	Recognition Media Capacity	Applicable Formats	Remarks
1	Max. 256Mbyte	FAT16 / FAT32	
2	256M to 8Gbyte	FAT32	

5.2 Audio Data

The items of audio data for the software development environment offered as a sample are given in Table 5. If using MP3/AAC file decoder (middleware), a contract is required.

Table 5 Corresponding Audio Data

No.	Item	Description	Remarks
1	File format	MP3 or AAC	MP3 and AAC cannot be handled simultaneously. (*)
2	File name	“music1.mp3” to “music5.mp3” “music1.aac” to “music5.aac”	
3	Sampling rate	8k/11.025k/12k/16k/22.05k/24k/ 32k/44.1k/48k	Unit: Hz
4	Bit rate	8k to 320kbps	VBR not supported Sampling rate is up to 160kbps in the case of 44.1kHz/48kHz.
5	Channel mode	[MP3] joint stereo Intensity stereo MS stereo dual channel [AAC] stereo(MPEG2/MPEG4)	

* Cannot be assembled simultaneously due to restriction of microcontroller built-in RAM size.

5.3 Image Data

The items of image data supported by the sample program are given in Table 6. If using a JPEG encoder/decoder (middleware), a contract is required.

Table 6 Image Data Supported by the Sample Program

No.	Item	Description	Remarks
1	File format	JPEG	
2	File name	“picture1.jpg” to “picture5.jpg” “select.jpg”	
3	Resolution	240×320 to 3840×5120	

[Note] If AAC decoder can be assembled for audio data processing, the JPEG encoder/decoder cannot be assembled due to restriction of the microcontroller built-in RAM size.

6 OPERATION METHOD

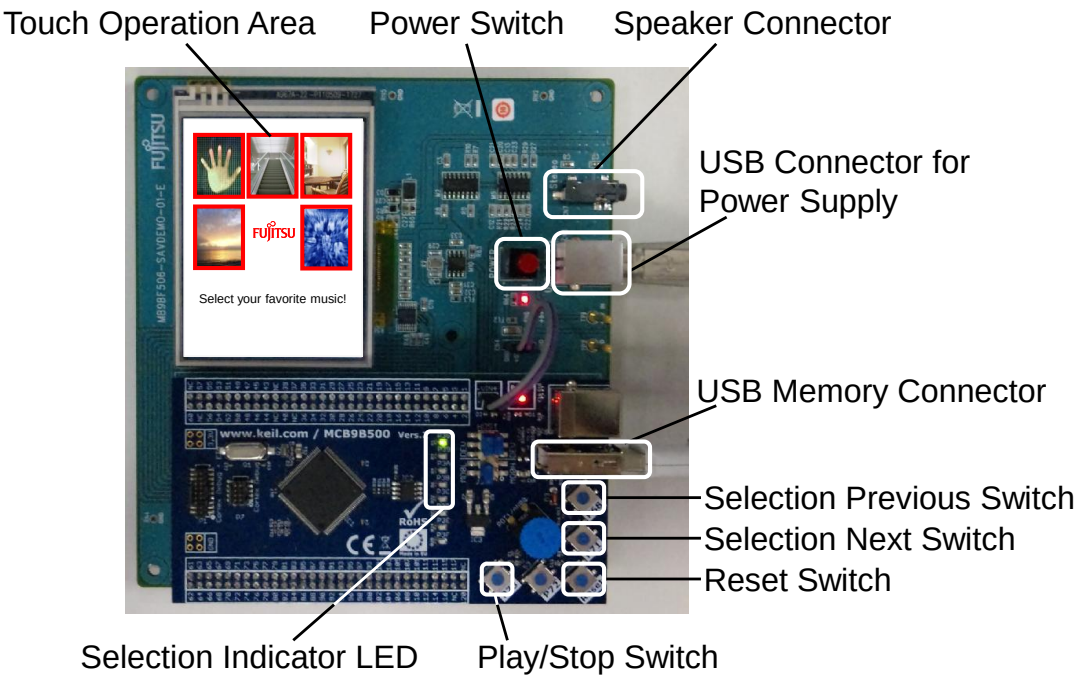

Figure 16 Operation Explanation Diagram

6.1 MP3 File Play

(1) From Preparation to Power On

① Audio data (music1.mp3 to music5.mp3) and image data (picture1.jpg to picture5.jpg, select.jpg) are contained in the root directory of the USB memory.

② Insert USB memory in USB memory connector of simple AV system board.

③ Connect earphones or speakers to the speaker connector.

④ Connect USB cable to the USB connector for power supply.

⑤ Press down power switch (power on).

(2) Song Selection Operation

Pressing the song selection next switch selects audio file in the following order.

music1.mp3 => music2.mp3 => ... => music5.mp3

If the next switch is pressed while music5.mp3 is selected, music5.mp3 is then selected.

Pressing the song selection previous switch selects audio file in the following order.

music5.mp3 => music4.mp3 => ... => music1.mp3

If the next switch is pressed while music1.mp3 is selected, music1.mp3 is then selected.

LED1 to LED5 correspond to music1.mp3 to music5.mp3; the lit LED indicated the file currently selected.

Songs can be selected while music is not played.

Figure 17 Song Selection by Previous/Next Switch

(3) Play Start Operation

While music is not yet played, the selected music can be played by the following operation.

① Press down the Play/Stop switch.

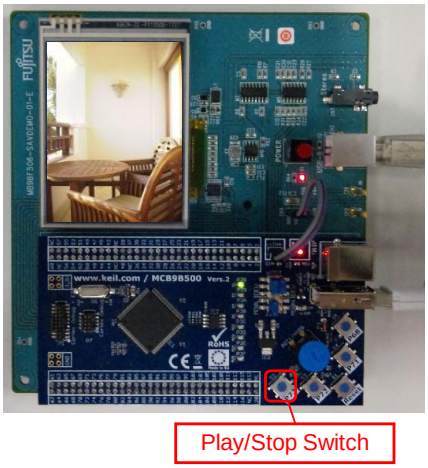

Figure 18 Play Start Operation by Play/Stop Switch

② Touch the thumbnail on the touch panel (LCD) (MP3).

(picture1.jpg to picture5.jpg corresponds to music1.mp3 to music5.mp3)

Touch the touch operation area image with your finger.

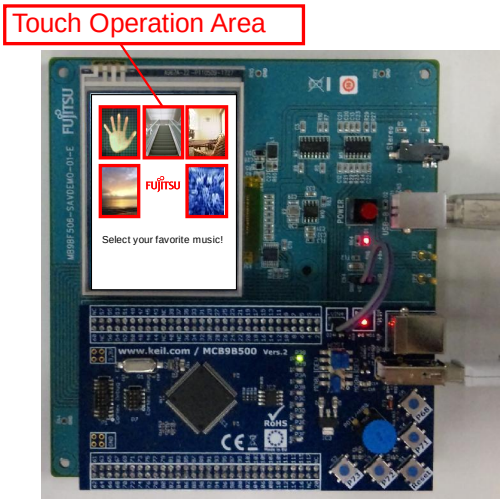

Figure 19 Play Start Operation by Touch Panel

The touched image is enlarged and the music linked with the image starts playing.

(Example) If the touched image were picture3.jpg, music3.mp3 would be selected.

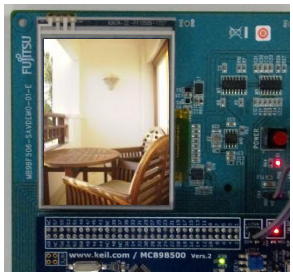

Figure 20 LCD Display While Playing

(4) Play Stop Operation

While music is playing, it can be stopped by the following operation.

① Press the Play/Stop switch.

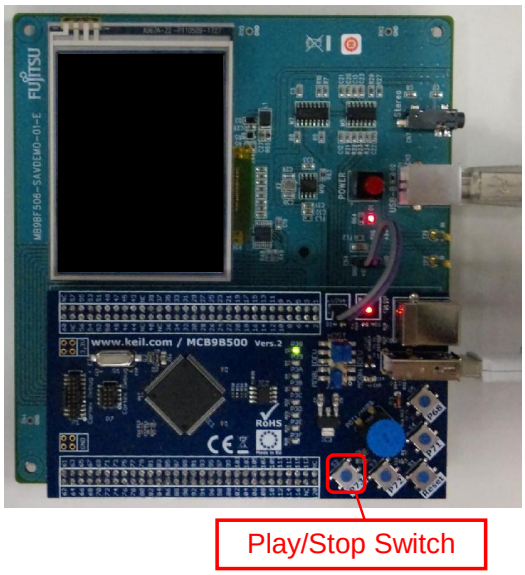

Figure 21 Play Stop Operation by Play/Stop Switch

② Touch the touch panel.

When you touch the enlarged image with your finger, the music stops playing.

Figure 22 Play Stop Operation by Touch Panel

When music is stopped the image simultaneously switches the thumbnail display.

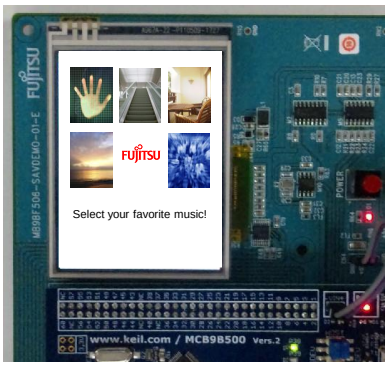

Figure 23 Thumbnail Display

6.2 AAC File Play

(1) From Preparation to Power On

Audio data (music1.aac to music5.aac) is contained in the root directory of the USB memory.

Insert USB memory in USB memory connector of simple AV system board.

Connect earphones or speakers to the speaker connector.

Connect USB cable to the USB connector for power supply.

Press power switch (power on).

(2) Song Selection Operation

Pressing the song selection next button selects audio file in the following order.

music1.aac => music2.aac => ... => music5.aac

If the next switch is pressed while music5.aac is selected, music5.aac is then selected.

Pressing the song selection previous switch selects audio file in the following order.

music5.aac => music4.aac => ... => music1.aac

If the next switch is pressed while music1.aac is selected, music1.aac is then selected.

LED1 to LED5 correspond to music1.aac to music5.aac; the lit LED indicated the file currently selected.

Songs cannot be selected while music is being played. Stop playing to select.

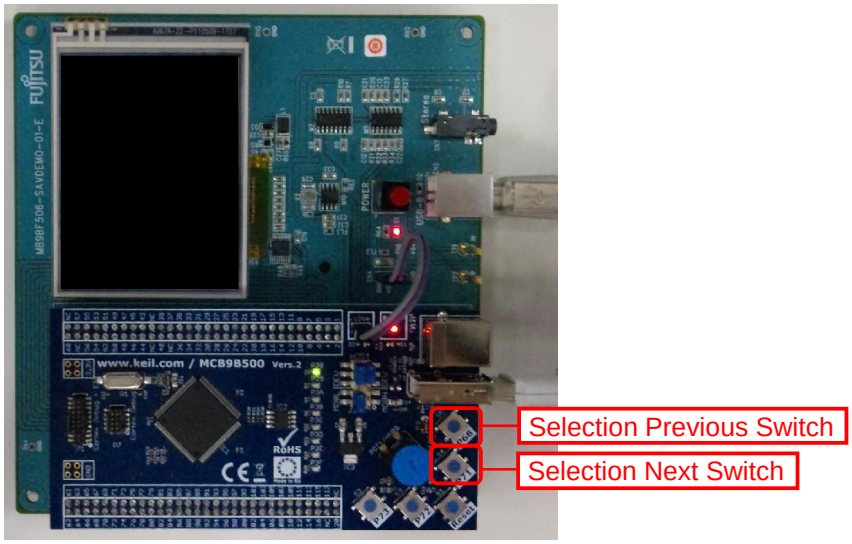

Figure 24 Song Selection by Previous/Next Switch

(3) Play Start Operation

While music is being played, the selected music can be played by the following operation.

- Press down the Play/Stop switch.

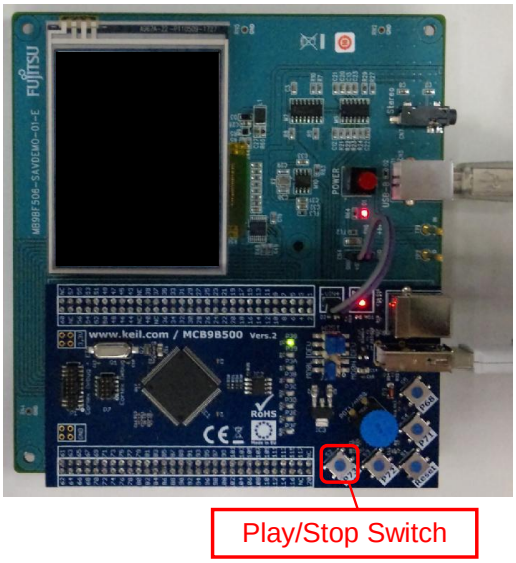

Figure 25 Play Start Operation by Play/Stop Switch

(4) Play Stop Operation

While music is playing, it can be stopped by the following operation.

- Press the Play/Stop switch.

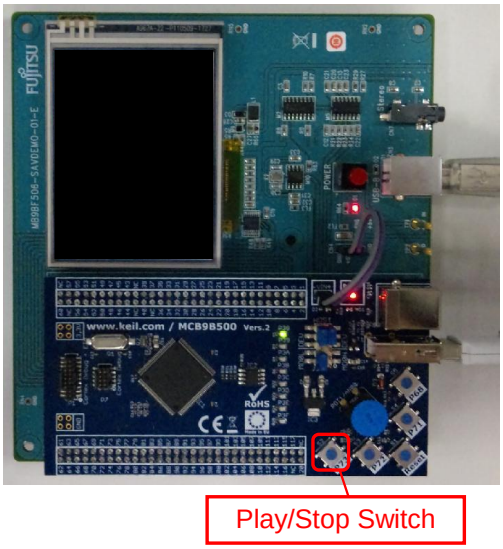

Figure 26 Play Stop Operation by Play/Stop Switch

7 SPECIFICATIONS

7.1 Hardware

7.1.1 General Specifications

General specifications of the simple AV system board are given in Table 7.

Table 7 General Specifications

No.	Item		Description	Remarks
1	Microcontroller		MB9BF506R(Fujitsu Semiconductor)	
2	Power supply		USB bus power (+5V)	
3	Current consumption		300mA(typ.)	Does not include external connected equipment.
4	Switch	Key input switch	Pushbutton switch x 4	
5		Reset switch	Pushbutton switch x 1	
6		USB function Switch	Jumper socket x 2	
7	Indicator	LCD panel with touch panel	[LCD] ・2.4” TFT ・240×320 pixels [Touch panel] ・4-wire resistive touch panel	
8		LED	8 units	
9	External I/F	Audio	Speaker jack x 1ch	
10		USB I/F	USB A type×1ch	For USB memory connection
11			USB B type×2ch	
12		JTAG I/F	For KEIL ICE connection x 1ch	
13		ETM I/F	For KEIL ICE connection x 1ch	
14	Environment conditions		0~60 degrees C	
15	Dimensions (WxD)		130×135mm(typ.)	
16	Weight		164g(typ.)	

7.1.2 Hardware Block Diagram

The hardware block diagram is shown in Figure 27.

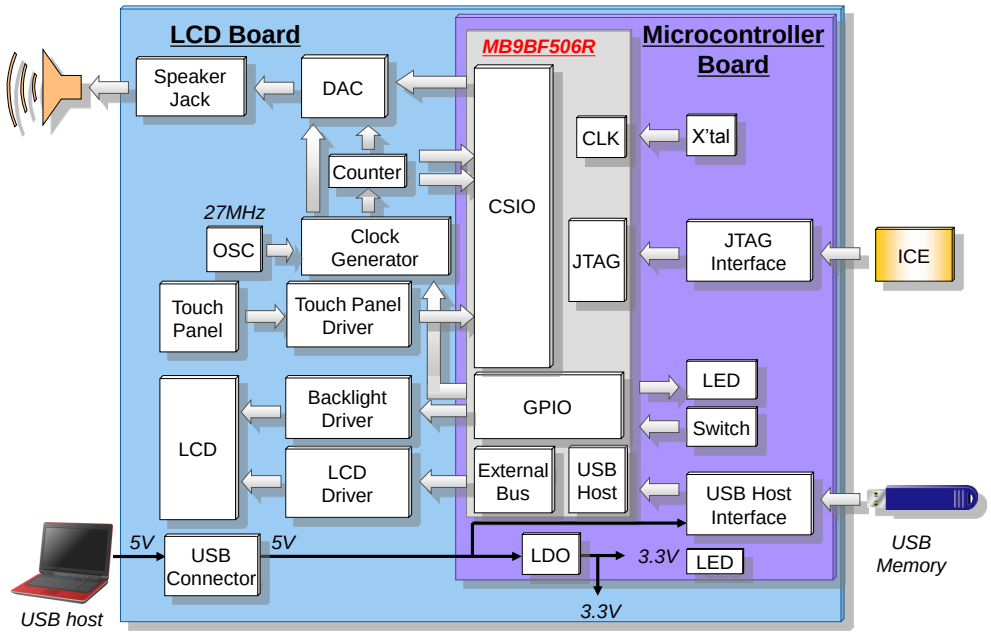

Figure 27 Hardware Block Diagram

7.1.3 Main Components

A list of main components is given in Figure 8.

Figure 8 Main Components of Simple AV System Board

No.	Part Number	pcs.	Remarks
1	Microcontroller	1	Fujitsu Semiconductor MB9BF506R Microcontroller Board: KEIL MCB9B500
2	LCD with touch panel	1	2.4” TFT AMPIRE AM-240320D4TNQW-00H®
3	LCD Driver	1	ILITEK ILI9320
4	Backlight Driver	1	ROHM BD6067GU
5	Touch Panel Driver	1	MAXIM MXB7843EUE+
6	Clock Generator	1	IDT MK2705SLF
7	OSC	1	Kyocera CX3225SB27000D0GEJZ1
8	Counter	1	On Semiconductor MC74AC161DG
9	DA converter	1	Audio data conversion, CIRRUS CS4354-CSZ
10	Speaker connector	1	For earphones or speakers connection Kycon STX-3500
11	USB-B power supply connector	1	For supplying power from external USB host

(*)Refer to the following URL for the schematics of the microcontroller board.

http://www.keil.com/mcb9b500/mcb9bf500-schematics.pdf

The above URLs may be changed without notice.

7.1.4 Microcontroller

7.1.4.1 Pin Connection

A photograph of the external appearance of the microcontroller is shown in Figure 28 and a list of pin connections is given in Table 9.

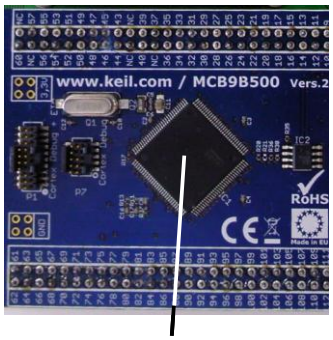

Figure 28 Photograph of Microcontroller External Appearance

Table 9 List of Microcontroller Pin Connections

Pin No.	Pin Name (Use Function)	Connection Destination	I/O	Remarks
01	VCC	+3.3V power supply	—	
02	MDATA00	LCD controller (D0)	I/O	
03	MDATA01	LCD controller (D1)	I/O	
04	MDATA02	LCD controller (D2)	I/O	
05	MDATA03	LCD controller (D3)	I/O	
06	MDATA04	LCD controller (D4)	I/O	
07	MDATA05	LCD controller (D5)	I/O	
08	SPI_MISO	Touch panel controller (DOUT)	I	
09	SPI_MOSI	Touch panel controller (DIN)	O	
10	SPI_SCK	Touch panel controller (DCLK)	O	
11	(Not used)	—	—	
12	(Not used)	—	—	
13	(Not used)	—	—	
14	MDATA06	LCD controller (D6)	I/O	

Pin No.	Pin Name (Use Function)	Connection Destination	I/O	Remarks
15	MDATA07	LCD controller (D7)	I/O	
16	(Not used)	—	—	
17	(Not used)	—	—	
18	(Not used)	—	—	
19	(Not used)	—	—	
20	P36	Touch panel controller (xCS)	O	
21	INT10_1	Touch panel controller (xPENIRQ)	I	
22	(Not used)	—	—	
23	(Not used)	—	—	
24	(Not used)	—	—	
25	(Not used)	—	—	
26	(Not used)	—	—	
27	(Not used)	—	—	
28	(Not used)	—	—	
29	(Not used)	—	—	
30	VSS	GND	—	
31	VCC	+3.3V power supply	—	
32	(Not used)	—	—	
33	(Not used)	—	—	
34	(Not used)	—	—	
35	(Not used)	—	—	
36	(Not used)	—	—	
37	(Not used)	—	—	
38	C	4.7 µF capacitor	—	
39	VSS	GND	—	
40	VCC	+3.3V power supply	—	
41	(Not used)	—	—	
42	(Not used)	—	—	
43	(Not used)	—	—	
44	(Not used)	—	—	
45	(Not used)	—	—	

Pin No.	Pin Name (Use Function)	Connection Destination	I/O	Remarks
46	(Not used)	—	—	
47	(Not used)	—	—	
48	(Not used)	—	—	
49	(Not used)	—	—	
50	(Not used)	—	—	
51	(Not used)	—	—	
52	(Not used)	—	—	
53	(Not used)	—	—	
54	(Not used)	—	—	
55	(Not used)	—	—	
56	MD1	GND	—	
57	MD0	Jumper pin	—	H/L can be switched
58	X0	Crystal oscillator (4MHz)	I	
59	X1	Crystal oscillator (4MHz)	I/O	
60	VSS	GND	—	
61	VCC	+3.3V power supply	—	
62	(Not used)	Connected to potentiometer	—	
63	(Not used)	—	—	
64	P12	LCD backlight controller (EN)	O	LCD backlight control H: Enable, L: Disable
65	MAD08	LCD controller (RS)	O	
66	P14	LCD controller (/RESET)	O	L: Reset active
67	MCSX0	LCD controller (/CS)	O	
68	MOEX	LCD controller (/RD)	O	
69	MWEX	LCD controller (WR/SCI)	O	
70	AVCC	+3.3V power supply	—	
71	AVRH	+3.3V power supply	—	
72	AVSS	GND	—	
73	MDATA8	LCD controller (D8)	I/O	
74	MDATA9	LCD controller (D9)	I/O	
75	MDATA10	LCD controller (D10)	I/O	
76	MDATA11	LCD controller (D11)	I/O	

Pin No.	Pin Name (Use Function)	Connection Destination	I/O	Remarks
77	MDATA12	LCD controller (D12)	I/O	
78	MDATA13	LCD controller (D13)	I/O	
79	MDATA14	LCD controller (D14)	I/O	
80	MDATA15	LCD controller (D15)	I/O	
81	(Not used)	—	—	
82	(Not used)	—	—	
83	(Not used)	—	—	
84	(Not used)	—	—	
85	(Not used)	—	—	
86	(Not used)	—	—	
87	(Not used)	—	—	
88	(Not used)	—	—	
89	(Not used)	—	—	
90	VSS	GND	—	
91	VCC	+3.3V power supply	—	
92	TRSTX	N.C.	—	
93	TCK	JTAG connector, ETM connector	I	
94	TDI	JTAG connector, ETM connector	I	
95	TMS	JTAG connector, ETM connector	I/O	
96	TDO	JTAG connector, ETM connector	O	
97	(Not used)	ETM connector	I	
98	(Not used)	ETM connector	I	
99	(Not used)	ETM connector	I	
100	(Not used)	ETM connector	I	
101	(Not used)	ETM connector	I	
102	(Not used)	—	—	
103	SOT4_0	DA converter (LRCK)	O	
104	SCK4_0	DA converter (SCLK/DEM), Binary Counter (Q2[8 divisions output])	I	
105	(Not used)	—	—	
106	(Not used)	—	—	

Pin No.	Pin Name (Use Function)	Connection Destination	I/O	Remarks
107	(Not used)	—	—	
108	(Not used)	—	—	
109	(Not used)	—	—	
110	(Not used)	—	—	
111	SCK5_1	DA converter (SCK), Binary Counter (Q2[8 divisions output])	I	
112	SOT5_1	DA converter (SDIN)	O	
113	P63	Clock Generator(S0)	O	
114	(Not used)	—	—	
115	P61	Clock Generator(S1)	O	
116	(Not used)	—	—	
117	USBVCC	+3.3V power supply	—	
118	UDM0	USB jumper pin for selection	—	
119	UDP0	USB jumper pin for selection	—	
120	VSS	GND	—	

7.1.4.2 Microcontroller Mode Switch and Function Switch

The microcontroller has the following two modes, which can be switched by external jumper socket.

（1） Serial writer mode

（2） User mode

This section contains an overview of the two modes and a description of the setting method.

（1） Serial writer mode

[Overview]

Built-in flash serial programming of the microcontroller mounted on the microcontroller board can be carried out.

For details, see the “Flash Programming Manual” of the microcontroller.

[Setting Method]

Insert the jumper for switching modes of the microcontroller in the silk notation “MD0H” side.

The Operating Mode off the microcontroller is decided after releasing power on reset, low-voltage detection reset and INITX pin input reset.

When selecting this mode, set to USB function by switching to USB function function of the microcontroller.

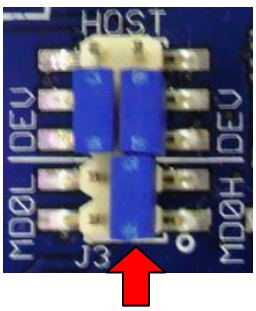

Figure 29 Method of Setting Software Writing Mode

（2） User Mode

[Overview]

Mode that activates the internal ROM (flash) of the microcontroller mounted in the microcontroller. Operation starts when the CPU obtains reset vector from the flash memory. Setting the MD0 pin of the microcontroller to “L” level switches to this mode.

[Setting Method]

Insert the jumper for switching modes of the microcontroller in the silk notation “MD0L” side. The Operating Mode off the microcontroller is decided after releasing power on reset, low-voltage detection reset and INITX pin input reset. When selecting this mode, set to USB function by switching USB host function of the microcontroller.

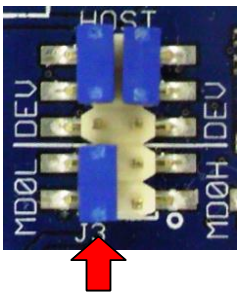

Figure 30 Normal Mode Setting Method

7.1.4.3 USB Interface Function Switching

The microcontroller has the following two USB interface functions, which can be switched by external jumper socket.

（1） USB Host Interface Function

（2） USB Function Interface Function

This section contains an overview of the two interface functions and a description of the setting method.

（1） USB Host Interface Function (Set when User Mode is Selected)

[Overview]

The USB interface of the microcontroller is used as the host.

[Setting Method]

Insert the jumper for switching modes of the microcontroller in the silk notation “HOST” side.

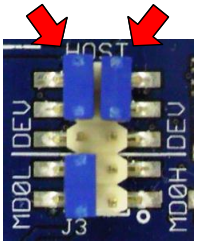

Figure 31 USB Host Function Setting Method

（2） USB Function Interface Function (Set when Serial Writer Mode is Selected)

[Overview]

The USB interface of the microcontroller is used as the function.

[Setting Method]

Insert the jumper for switching modes of the microcontroller in the silk notation “DEV” side.

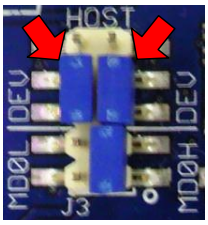

Figure 32 USB Function Function Setting Method

7.1.5 DA Converter

The LCD board is equipped with a DA converter for I^2S conversion. The external appearance and connection diagram of the DA converter are shown in Figure 33.

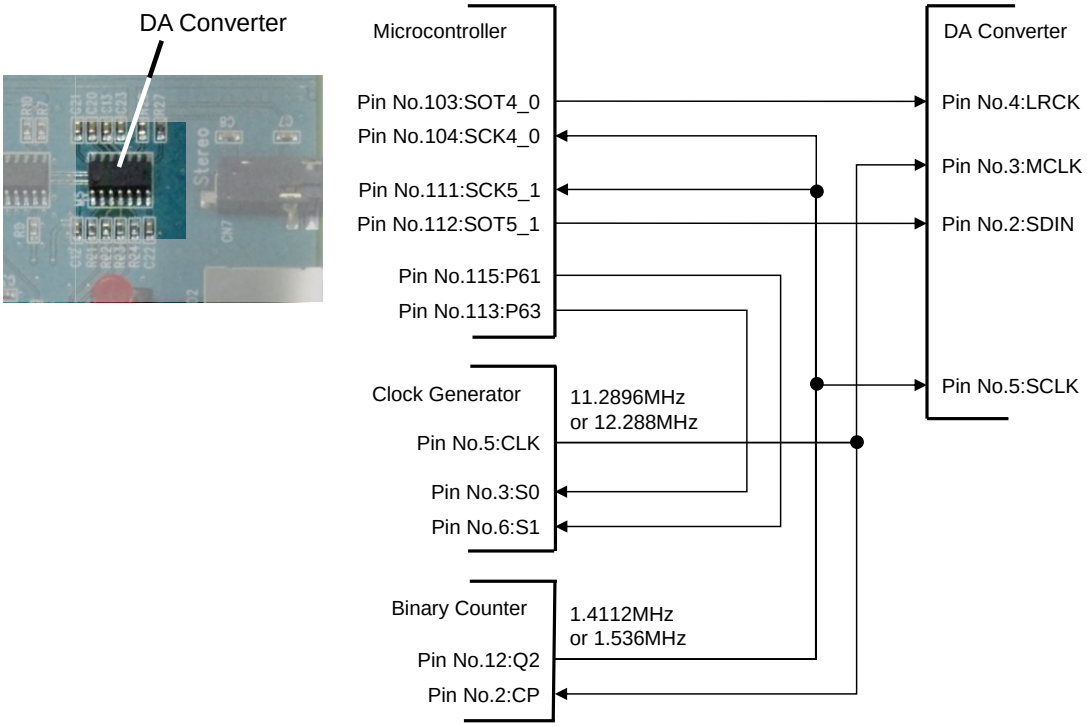

Figure 33 DA Converter External Appearance and Connection Diagram

7.1.6 Switches

7.1.6.1 Power switch

The LCD board is equipped with a power switch. The external appearance of the power switch is shown in Figure 34.

For power supply method, see section 3.4.

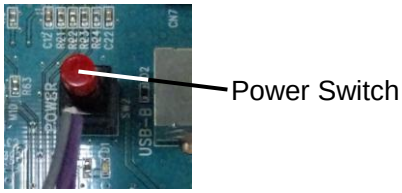

Figure 34 Power Switch External Appearance

7.1.6.2 Key Input Switch

The LCD board is equipped with a power switch. The external appearance and connection diagram of the key input switch are shown in Figure 35.

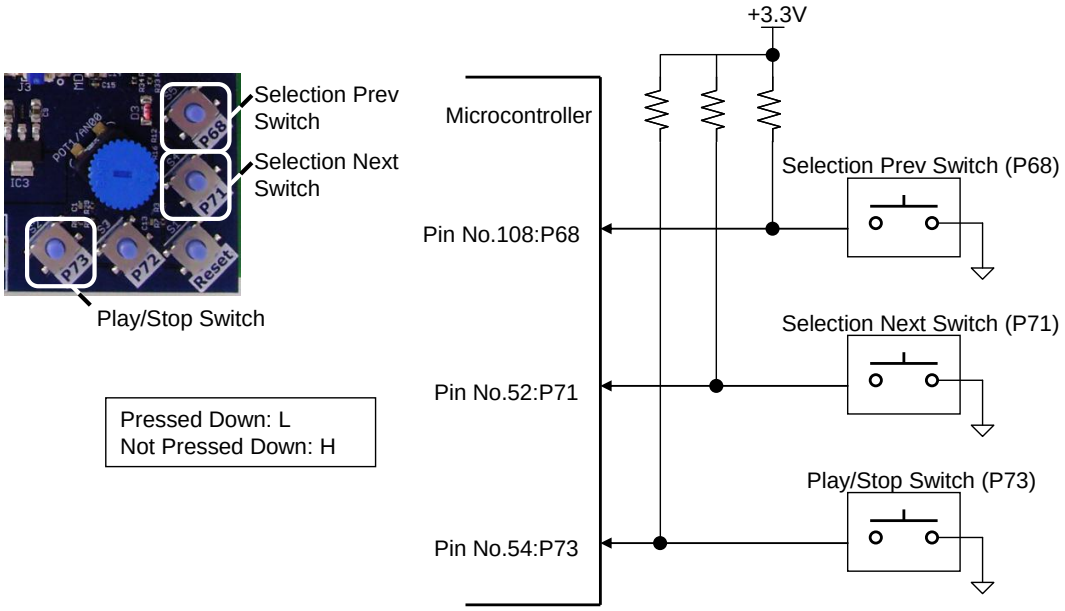

Figure 35 Key Input Switch External Appearance and Connection Diagram

7.1.6.3 Reset Switch

The microcontroller board is equipped with a reset switch. The external appearance and connection diagram of the reset switch are shown in Figure 36.

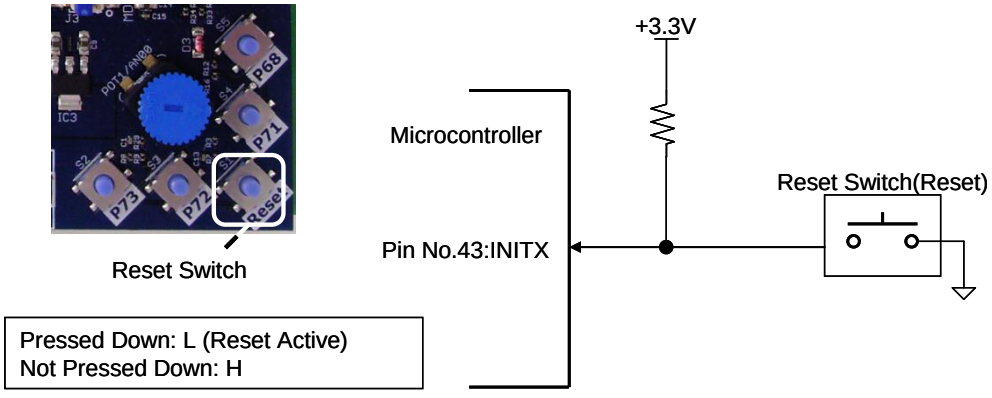

Figure 36 Reset Switch External Appearance and Connection Diagram

7.1.7 Indicator

7.1.7.1 LCD Module with Touch Panel

The LCD board is equipped with an LCD module with touch panel. The external appearance of the LCD module with touch panel is shown in Figure 37 and the specifications of the LCD module with touch panel are given in Figure 10. For connection of microcontroller and LCD module with touch panel, see Table 9.

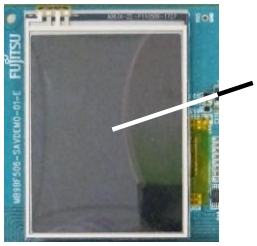

Figure 37 External Appearance of LCD Module with Touch Panel

Figure 10 Specifications of LCD Module with Touch Panel

No.	Item	Description	Remarks
1	LCD driver model	ILI9320(ILI Technology)	
2	Touch panel driver	MXB7843EUE+(MAXIM)	
3	LCD backlight driver	BD6067GU(ROHM)	
4	LCD	2.4” TFT with 4-wire resistive touch panel 240×320 pixels 16-bit parallel	

7.1.7.2 Power LED

The microcontroller board is equipped with a power LED that indicates power supply status. The external appearance of the power LED is shown in Figure 38 and the specifications are given in Table 11.

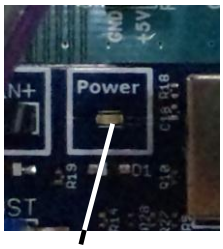

Figure 38 Power LED external appearance

Table 11 Power LED Specifications

No.	Item	Color	Specifications	Connection Destination
1	Power LED	Red	Power ON: On Power OFF: Off	+5V power supply

7.1.7.3 Selection Indicator LED

The microcontroller board is equipped with a selection indicator LED that indicates selection status. The external appearance and connection diagram of the selection indicator LED are shown in Figure 39 and the selection indicator LED specifications are given in Table 12.

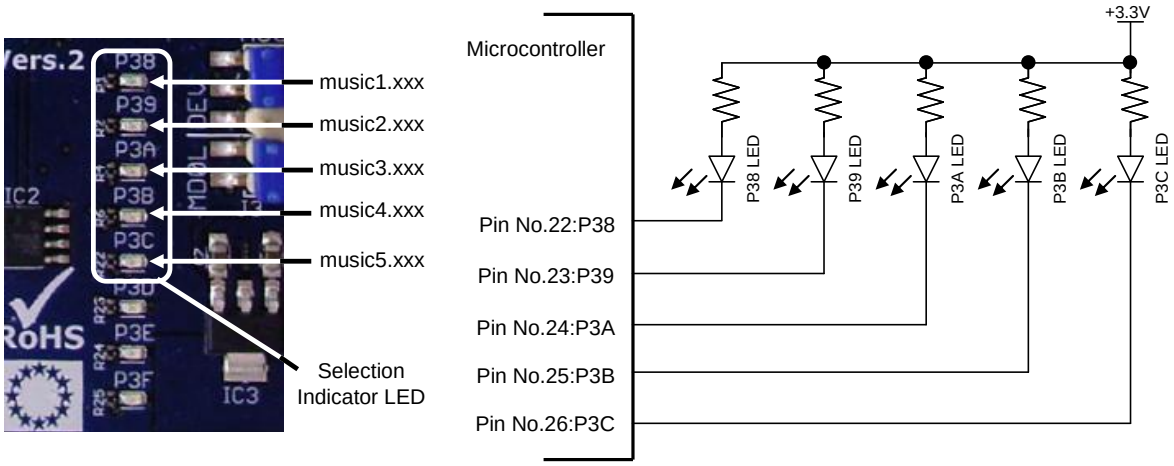

Figure 39 Selection Indicator LED External Appearance and Connection Diagram

Table 12 Selection Indicator LED Specifications

No.	Item	Color	Microcontroller connection destination	Specifications
1	P38 LED	Green	Pin No. 22: P38	Microcontroller H output: Off Microcontroller L output: On
2	P39 LED	Green	Pin No. 23: P39	Microcontroller H output: Off Microcontroller L output: On
3	P3A LED	Green	Pin No. 24: P3A	Microcontroller H output: Off Microcontroller L output: On
4	P3B LED	Green	Pin No. 25: P3B	Microcontroller H output: Off Microcontroller L output: On
5	P3C LED	Green	Pin No. 26: P3C	Microcontroller H output: Off Microcontroller L output: On

7.1.8 External Interface

7.1.8.1 Power Supply Interface

The LCD board is equipped with a USB connector that functions as a power supply interface.

Power is supplied by connecting the USB cable to the USB connector for power supply.

The external appearance of the USB connector for power supply is shown in Figure 40.

The USB connector for power supply uses a conventional USB-B Type connector, but the USB signal line is not connected. Only VBUS and GND are connected.

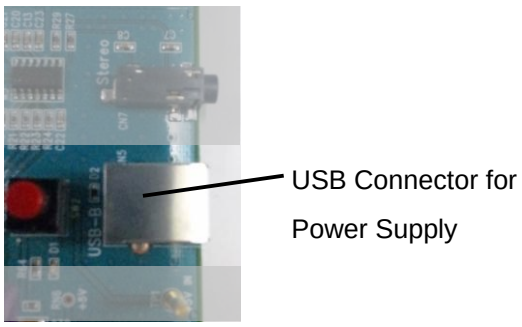

Figure 40 USB Connector for Power Supply

7.1.8.2 Audio Interface

The LCD board is equipped with a speaker connector that functions as an audio interface.

The external appearance and connection diagram of the speaker connector are shown in Figure 41.

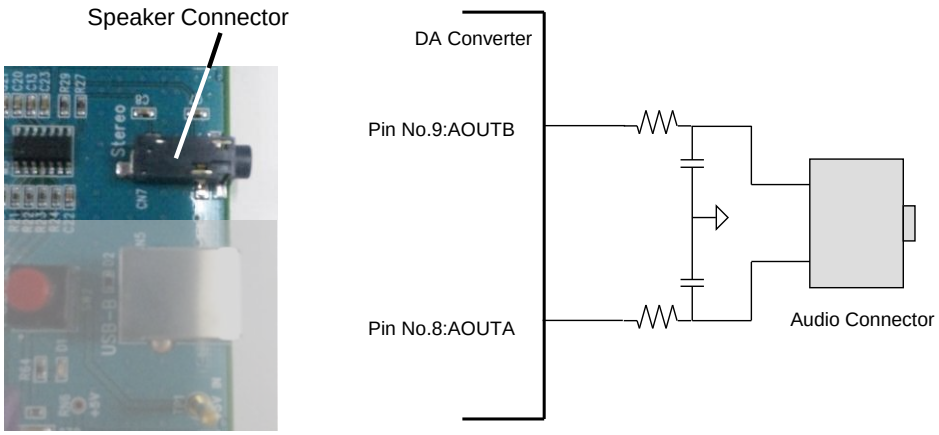

Figure 41 Speaker Connector External Appearance and Connection Diagram

7.1.8.3 USB Interface

The microcontroller board is equipped with a USB memory connector that functions as a USB interface.

The external appearance of the USB memory connector is shown in Figure 42 and the USB memory connector specifications are given in Table 13.

Figure 42 USB Memory Connector External Appearance

Table 13 USB Memory Connector Specifications

Pin No.	I/O (*)	Microcontroller connection destination	
		Pin No.	Signal Name
1	—	—	+5V（VBUS）
2	I/O	118	UDM0（D-）
3	I/O	119	UDP0（D+）
4	—	—	GND
5	—	—	GND

* I/O as seen from the microcontroller.

7.1.8.4 JTAG Interface

The microcontroller board is equipped with a JTAG interface connector for software debugging by JTAG interface.

The external appearance of the JTAG interface connector is shown in Figure 43 and the pin assignment is given in Table 14.

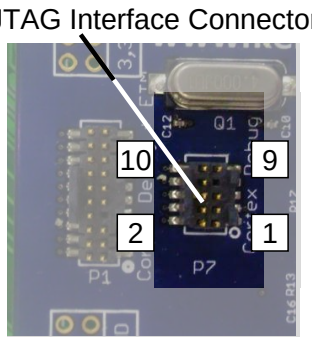

Figure 43 JTAG Interface Connector External Appearance

Table 14 JTAG Interface Connector Pin Assignment

Microcontroller connection destination		I/O (*)	Pin No.		I/O (*)	Microcontroller connection destination	
Pin No.	Signal Name					Pin No.	Signal Name
—	+3.3V	—	1	2	I/O	95	TMS
—	GND	—	3	4	I	93	TCK
—	GND	—	5	6	O	96	TDO
—	GND	—	7	8	I	94	TDI
—	GND	—	9	10	I	43	RESET

* I/O as seen from the microcontroller.

7.1.8.5 ETM Interface

The microcontroller board is equipped with an ETM interface connector for software debugging by ETM interface.

The external appearance of the ETM interface connector is shown in Figure 44 and the pin assignment is given in Table 15.

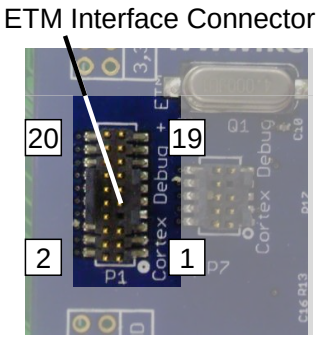

Figure 44 ETM Interface Connector External Appearance

Table 15 ETM Interface Connector Pin Assignment

Microcontroller connection destination		I/O (*)	Pin No.		I/O (*)	Microcontroller connection destination	
Pin No.	Signal Name					Pin No.	Signal Name
—	+3.3V	—	1	2	I/O	95	TMS
—	GND	—	3	4	I	93	TCK
—	GND	—	5	6	O	96	TDO
—	GND	—	7	8	I	94	TDI
—	GND	—	9	10	I	43	RESET
—	GND	—	11	12	O	101	TRACECLK
—	GND	—	13	14	O	97	TRACEDATA0
—	GND	—	15	16	O	98	TRACEDATA1
—	GND	—	17	18	O	99	TRACEDATA2
—	GND	—	19	20	O	100	TRACEDATA3

* I/O as seen from the microcontroller.

7.2 Software

7.2.1 Software Block Diagram

The software block diagram is shown in Figure 45.

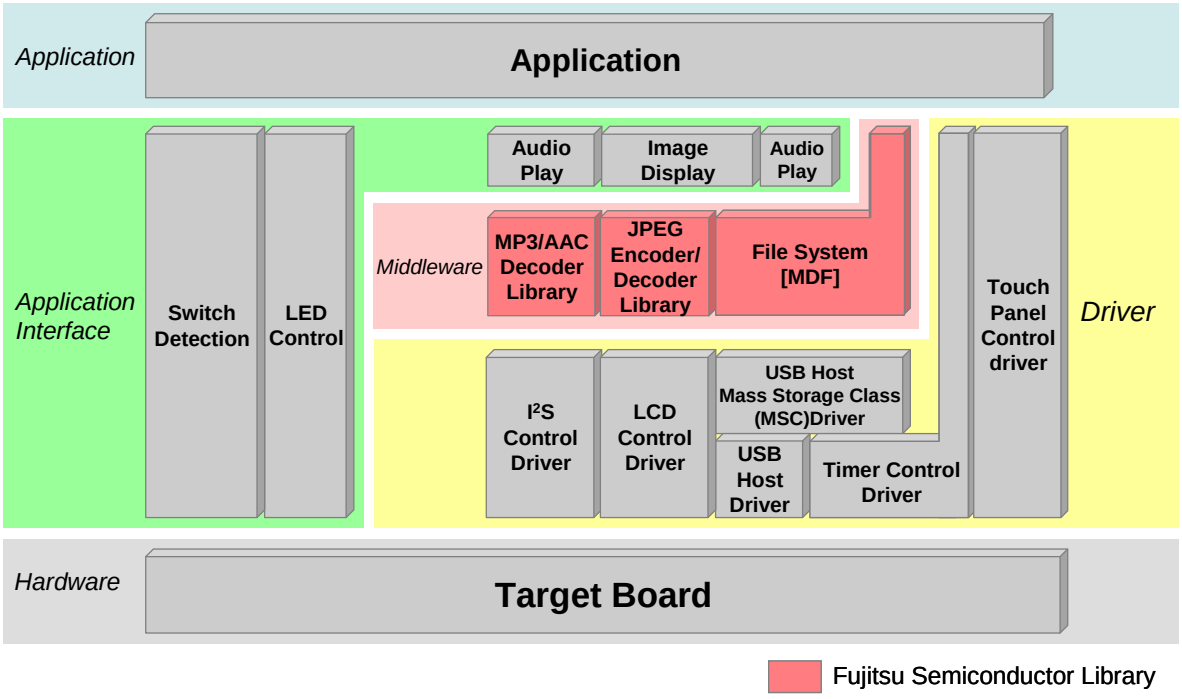

Figure 45 Software Block Diagram

7.2.2 Software Library

(1) File System

- Name

  Multi Device File Access Library V03L01 (object for small MCU, Evaluation)

- Overview

  File system library for embedded system (abbreviated as MDF).

  Used when you want to handle data a directory created by target devices by PC.

  Facilitates data transfer between PC and devices because multi device data can be managed by the same file and directory format used by the PC.

(2) MP3 Decoder

- Name

  MP3 Decoder Library for FM3 V01 Evaluation

- Overview

  MP3 audio decoder library for embedded system.

  Decodes MPEG Audio Layer III (MP3) bit stream data and outputs PCM format bit stream data.

(3) AAC Decoder

- Name

  MPEG-4/2 AAC LC Decoder Library (2ch) for FM3 V01 Evaluation

- Overview

  AAC audio decoder library for embedded system.

  Decodes MPEG-4 and MPEG-2 AAC bit stream data and outputs PCM format bit stream data.

(4) JPEG Encoder/Decoder

- Name

  JPEG Baseline Process Encoder/Decoder Library for FM3 V01 Evaluation

- Overview

  Image encoder/decoder library for embedded system.

  Encodes and decodes image data based on the baseline process of ITU-T T.8 and ISO/IEC 10918-1 standards, which are the still image compression standards.

  Decode function only is used for the simple AV system board.

7.2.3 System Specifications

7.2.3.1 Microcontroller System Specifications

Microcontroller system specifications are given in Table 16.

Table 16 Microcontroller System Specifications

Item	Description	Remarks
Operation Clock	CPU:80MHz APB1 to 3:40MHz	High-speed PLL oscillation Internal 20 multiplier
ROM(FLASH) (*1)	113.0Kbyte	MP3 Vector section: 248 bytes Program section: 112.7 Kbytes
	152.6 Kbytes	AAC Vector section: 248 bytes Program section: 152.4 Kbytes
RAM (*1)	64.0 Kbytes	MP3 Variable: 32.0 Kbytes Stack: 8.0 Kbytes Heap: 24.0 Kbytes
	63.8 Kbytes	AAC Variable: 24.3 Kbytes Stack: 8.0 Kbytes Heap: 31.5 Kbytes
MFS (*2)	Uses 2 ch	For I^2S communication With CSIO (*3) as the slave mode, realizes I^2S by serial output based on clock input from OSC. For details, see “7.2.3.3 MFS System Specifications”.
DMAC	Uses 4 ch	ch. 0/1: For USB host control ch. 2/3: For CSIO data transfer For details, see “7.2.3.4 DMAC System Specifications”.
USB	—	For USB host control
External bus	16bit	For LCD control
Timer	Base Timer ch2	For timer count, 1 ms cycle

(*1) For details, see “7.2.3.2 Memory Map”.

(*2) Multi Function Serial (MFS) Interface

(*3) Clock sync Serial I/O (CSIO) interface

7.2.3.2 Memory Map

The memory maps for ROM (flash) and RAM are shown in Figure 46.

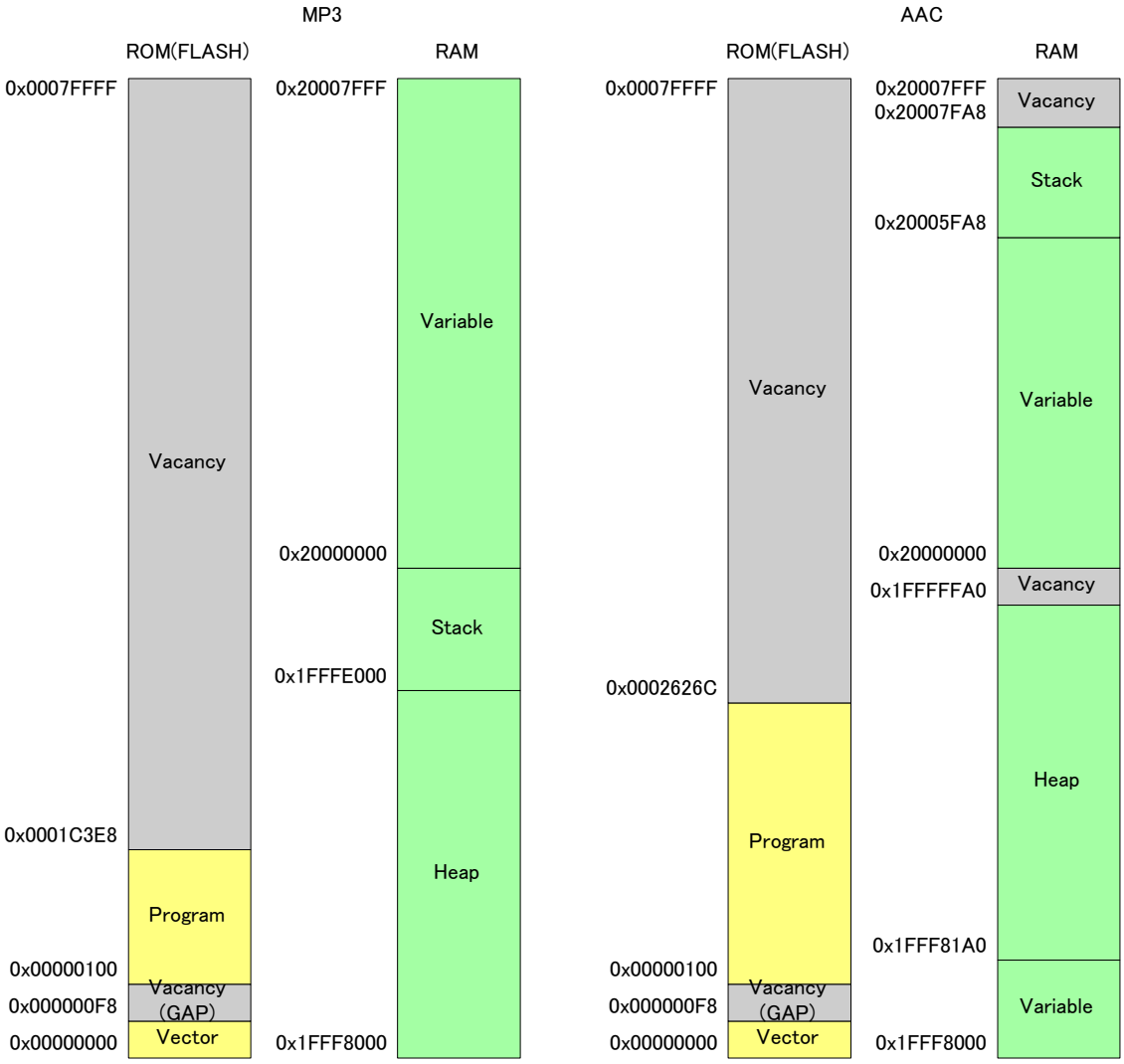

Figure 46 Memory Map

7.2.3.3 MFS System Specifications

MFS system specifications are given in Table 17.

Table 17 MFS System Specifications

Channel No.	Function	Baud rate	Remarks
0	—	—	(Not used)
1	—	—	(Not used)
2	—	—	(Not used)
3	—	—	(Not used)
4	I^2S(*)	1411200/1536000 (*)	LR channel data output to DAC
5	I^2S(*)	1411200/1536000 (*)	Audio data output to DAC
6	—	—	(Not used)
7	—	—	(Not used)

(*) Used as CSIO in slave mode for both channel 4 and 5

Clock uses 8 divisions of 11.2896MHz or 12.288MHz.

Because PCM data of 1 sample output to DAC is 16-bit Stereos (2ch), it is played at the sample rate of 44.1kHz or 48kHz by

- 141120/ 16(bit) / 2ch =44100
- 1536000 / 16 (bit) / 2 ch = 48000.

7.2.3.4 DMAC System Specifications

DMAC system specifications are given in Table 18.

Table 18 DMAC System Specifications

Channel No.	Application	Remarks
0	For data transmission to USB	
1	For data transmission from USB	
2	For I^2S (ch5) output data transfer	Used for interrupt
3	For I^2S (ch4) output data transfer	Set simultaneously with DMAC ch. 2

7.2.3.5 Interrupt System Specifications

Interrupt system specifications are given in Table 19.

Table 19 Interrupt System Specifications

Interrupt Factor	Function	Vector Number	Remarks
Reset	Reset_Handler	#01	—
Base timer ch. 2 interrupt	BTIM_IRQHandler	#47	Processes an interrupt by 1ms cycle.
USB host (each status) interrupt	USB_EP0_STA_IRQHandler	#50	Processes an interrupt for USB host interrupt.
DMAC ch. 2 interrupt	DMA2_IRQHandler	#56	Processes an interrupt for DMA ch2 interrupt. If there is play data, set transfer of audio data and LR channel data to I^2S.

7.2.4 API Specifications

This chapter gives I²S driver API used for audio playback processing in addition to API used for application layer. I²S driver API is also described in the application notes for simple AV system solution. For more information, see the notes.

7.2.4.1 Audio Playback API

Function	void AUDIO_Init(void)
Overview	Audio playback processing initialization function Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Argument	None
Return value	None

Function	AUDIO_STAGE_ENUM AUDIO_GetAudioStage(void)
Overview	Audio playback processing status acquisition function
Parameter	None
Return value	Audio playback processing status AUDIO_STAGE_INIT After completion of initialization AUDIO_STAGE_OPEN Audio data file open AUDIO_STAGE_LIBRARY_INIT Decoder library initialization AUDIO_STAGE_ANALYZE Audio data file decode AUDIO_STAGE_READ Audio data file read AUDIO_STAGE_DECODING Audio data decode AUDIO_STAGE_UPSAMPLE Decode data up-sampling AUDIO_STAGE_STOP Decode stop

Function	uint8_t AUDIO_SetAudioStage(AUDIO_STAGE_ENUM Stage)
Overview	Audio playback processing status setting function
Parameter	Audio playback processing status The following status setting only for application. AUDIO_STAGE_OPEN Audio data file open AUDIO_STAGE_STOP Decode stop
Return value	Processing results AUDIO_RET_OK Normal completion AUDIO_RET_INVALID_PARAMETER Parameter error

Function	void AUDIO_PlayTask(void)
Overview	Audio play processing main function Invoked during main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	None

7.2.4.2 Image Display API

Function	void IMAGE_Init(void)
Overview	Image display processing initialization function Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	None

Function	void IMAGE_ClearShow(void)
Overview	Image display clear function Displays entire LCD in white.
Parameter	None
Return value	None

Function	uint8_t IMAGE_Show(const uint8_t *FileName, uint8_t AreaID, uint16_t Color)
Overview	Image display control processing function
Parameter	FileName Filename read from USB memory AreaID LCD display area No. (*) IMAGE_SHOW_AREA1 Area 1 IMAGE_SHOW_AREA2 Area 2 IMAGE_SHOW_AREA3 Area 3 IMAGE_SHOW_AREA4 Area 4 IMAGE_SHOW_AREA5 Area 5 IMAGE_SHOW_AREA6 Area 6 IMAGE_SHOW_AREA7 Area 7 IMAGE_SHOW_AREA8 Area 8 IMAGE_SHOW_AREA9 Area 9 IMAGE_SHOW_AREA_ALL All Color Backgroynd color designation LCD_COLOR_WHITE White LCD_COLOR_BLACK Black LCD_COLOR_GREY Grey LCD_COLOR_BLUE Blue 1 LCD_COLOR_BLUE Blue 2 LCD_COLOR_RED Red LCD_COLOR_MAGENTA Magenta LCD_COLOR_GREEN Green LCD_COLOR_CYAN Cyan LCD_COLOR_YELLOW Yellow
Return value	Processing results IMAGE_RET_OK Normal completion IMAGE_RET_INVALID_PARAMETER Parameter error IMAGE_RET_ILLEGAL_ERROR Field code failure Memory secure error Designated file does not exist

(*) LCD display area corresponds as follows.

Area1	Area2	Area3
Area4	Area5	Area6
Area7	Area8	Area9

Figure 47 LCD Display Area

7.2.4.3 LED Control API

Function	void LED_Init (void)
Overview	LED control processing initialization function Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	None

Function	STATUS LED_Set(uint16_t IdCh, uint8_t LEDStatus)
Overview	LED control processing function Turns specified LED on or off.
Parameter	IdCh Control target LED (defined by bit) bit0 LED1 bit1 LED2 bit2 LED3 bit3 LED4 bit4 LED5 bit5 LED6 bit6 LED7 bit7 LED8 LEDStatus LED on/off control (corresponding bit=0: off, on when it is 1) bit0 LED1 bit1 LED2 bit2 LED3 bit3 LED4 bit4 LED5 bit5 LED6 bit6 LED7 bit7 LED8
Return value	Processing results LED_RET_OK Normal completion LED_RET_INVALID_PARAMETER Parameter error LED_RET_ILLEGAL_ERROR If invoked without being initialized

7.2.4.4 Switch Detection API

Function	void SW_Init (void)
Overview	Switch detection processing initialization function Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	None

Function	STATUS SW_GetStatus(uint8_t ButtonId, uint8_t *pStatus)	
Overview	Switch status acquisition function	
Parameter	ButtonId	Switch No.
	0	Play/Stop switch
	1	Song selection next switch
	2	Song selection previous switch
	pStatus	Pointer for place where switch status is stored
	SW_NO_PUSH	Not pressed
	SW_PUSH	Pressed
Return value	Processing results	
	SW_RET_OK	Normal completion
	SW_RET_INVALID_PARAMETER	Parameter error

7.2.4.5 Timer Control Driver API

Function	void TIMER_Init(void)
Overview	Initialization function for timer control driver Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	None

Function	STATUS TIMER_Start(uint8_t IdCh)
Overview	Timer start function Invoked before entering the main loop and after TIMER_Init invocation (see “7.2.8.1 Main Processing Function”).
Parameter	IdCh Control target timer ch TIMER_CH2 Base timer ch 2 TIMER_CH3 Base timer ch 3 (*) Ch 3 is not used.
Return value	Processing results TIMER_RET_OK Normal completion TIMER_RET_INVALID_PARAMETER Parameter error TIMER_RET_ILLEGAL_ERROR Timer control not initialized Designated ch timer operation

7.2.4.6 Touch Panel Control Driver API

Function	STATUS TOUCH_Init(void)
Overview	Initialization function for touch panel control driver Invoked before entering the main loop (see “7.2.8.1 Main Processing Function”).
Parameter	None
Return value	Processing results TOUCH_RET_OK Normal completion TOUCH_RET_ILLEGAL_ERROR Timer setting failure

Function	STATUS TOUCH_GetAreaStatus(uint16_t* Status)
Overview	Touch panel area status acquisition function
Parameter	Status Pointer to area status acquisition information storage destination If successful, area (see Figure 47) status (touch by 1) is written by bit. bit0 Area 1 bit1 Area 2 bit2 Area 3 bit3 Area 4 bit4 Area 5 bit5 Area 6 bit6 Area 7 bit7 Area 8 bit8 Area 9
Return value	Processing results TOUCH_RET_OK Normal completion TOUCH_RET_INVALID_PARAMETER Parameter error

7.2.4.7 I²S Driver API

Function	void I2S_Init (void)
Overview	Initializes I²S driver. Sets MFS initial settings (CSIO settings) and DMAC initial settings.
Parameter	None
Return value	None

Function	STATUS I2S_Start (uint8 t audio sample rate)
Overview	Starts I²S operation. Sets data for DMAC and activates CSIO. Be sure to initialize by I2S_Init function before function is invoked.
Parameter	Audio sample rate Sampling rate AUDIO_SAMPLE_44100 44.1kHz AUDIO_SAMPLE_48000 48kHz
Return value	I2S_RET_OK Successful I2S_RET_ILLEGAL_ERROR Not yet initialized, operating or stopped

Function	STATUS I2S_Stop (void)
Overview	Completes I²S operation. Stops CSIO.
Parameter	None
Return value	I2S_RET_OK Successful I2S_RET_ILLEGAL_ERROR Not yet initialized or stopped

Function	uint8_t I2S_Get_Status(void)
Overview	Acquires I²S operating status.
Parameter	None
Return value	I2S_STATUS_INIT Initialization status (I²S can be started) I2S_STATUS_START Start (I²S operating) status I2S_STATUS_STOP Stop (I²S cannot be started) status

7.2.5 Operation Limit

The following limitations apply to operation of the sample program used by the simple AV system.

◆Concerning assembly of MP3 and AAC decoder

The MP3 and AAC decoders cannot conduct processing simultaneously because the built-in RAM size is insufficient.

◆Concerning JPEG file processing for AAC decoder assembly

If an AAC decoder is assembled, JPEG file decode processing cannot be conducted because the built-in RAM size is insufficient.

7.2.6 Operation Flow of Entire Application

7.2.6.1 MP3

(1) The application operation flow with audio data playback stopped is as follows.

① USB MSC device connection/disconnection judgment is executed in the main loop.

② If a USB memory is connected, after reading the JPEG files from the USB memory and displaying the images for selection, switch pressing detection and touch panel detection are conducted.

③ If the play/stop switch is detected to be pressed down, or if not detected, but an area of the touch panel is detected to have been touched, the JPEG files corresponding to the selected MP3 file are read from the USB memory and displayed for playback. If the selection previous switch or selection next switch are detected to have been pressed down, MP3 file selection is shifted and LED control is executed.

④ The MP3 selected from the USB memory is then opened.

⑤ The MP3 file header is read, MP3 file header analysis processing is conducted and operation shifts to audio data playback in progress status.

This operation is shown in Figure 48.

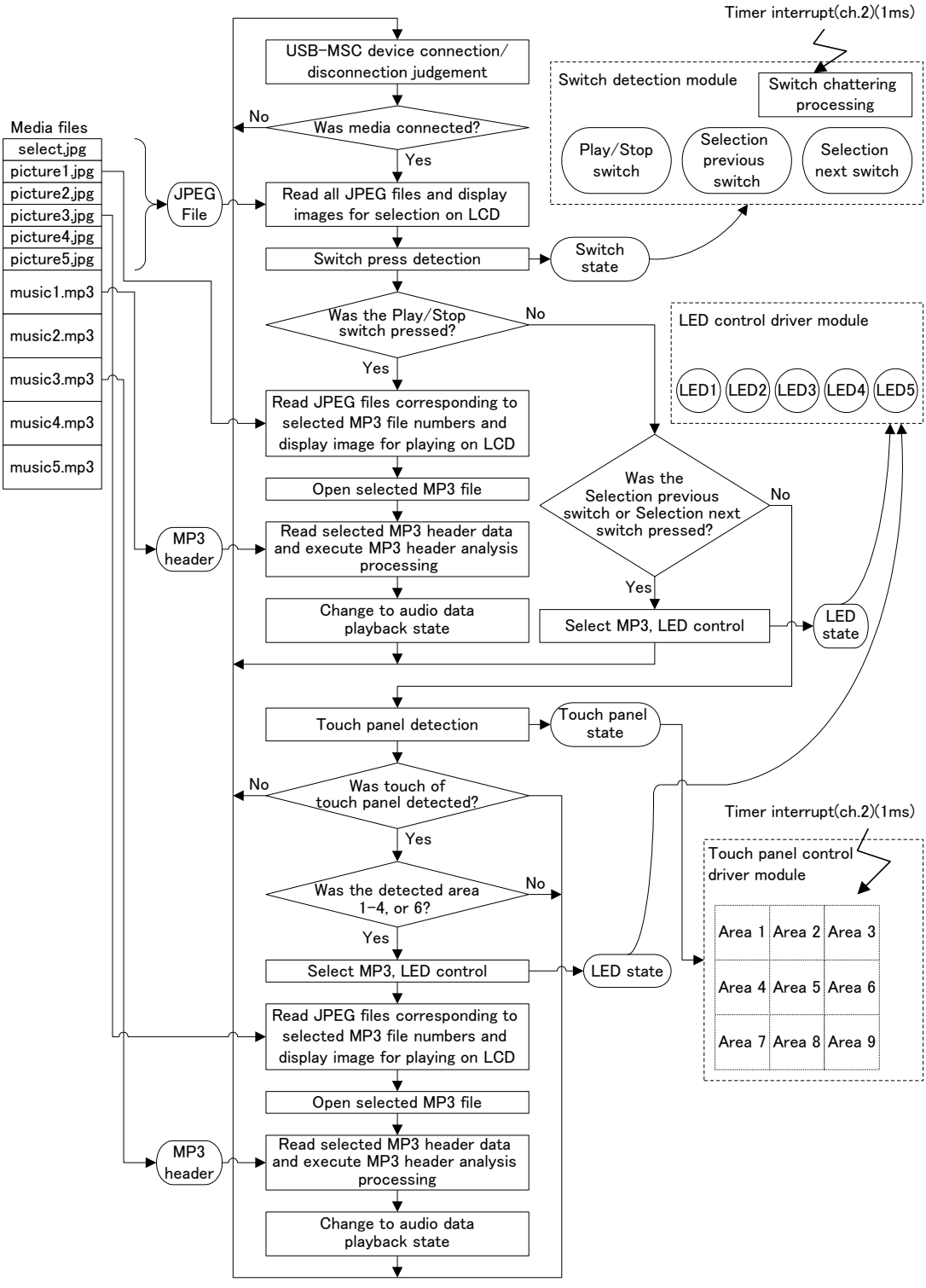

Figure 48 Application Operation Flow (Audio Playback Stopped Status, Case of MP3)

(2) The application operation flow with audio data playback in progress is as follows.

① USB MSC device connection/disconnection judgment is executed in the main loop.

② If the USB memory has been removed, stop playback, close the opened MP3 file, quit the file system and operation shifts to initialization status.

③ If the USB memory is connected, play/stop switch press down detection and touch panel touch detection are executed.

④ If the play/stop switch is detected to have been pressed down, or if not detected, but an area of the touch panel is detected to have been touched, playback is stopped, the MPA file is closed, all JPEF filed are read from the USB memory, that images for selection are displayed and operation shifts to audio data playback stopped status.

⑤ Verify vacancy of input buffer.

⑥ If there is sufficient vacancy, the MP3 file is read from the USB memory and copied in the input buffer.

⑦ One frame of the input buffer is decoded and stored in the RAW buffer.

⑧ When 1 frame had been decoded, the RAW buffer is up-sampled and buried in the output buffer.

⑨ With DMA ch2 interrupt, data is sent from the output buffer to I^2S in sequence.

This operation is shown in Figure 49.

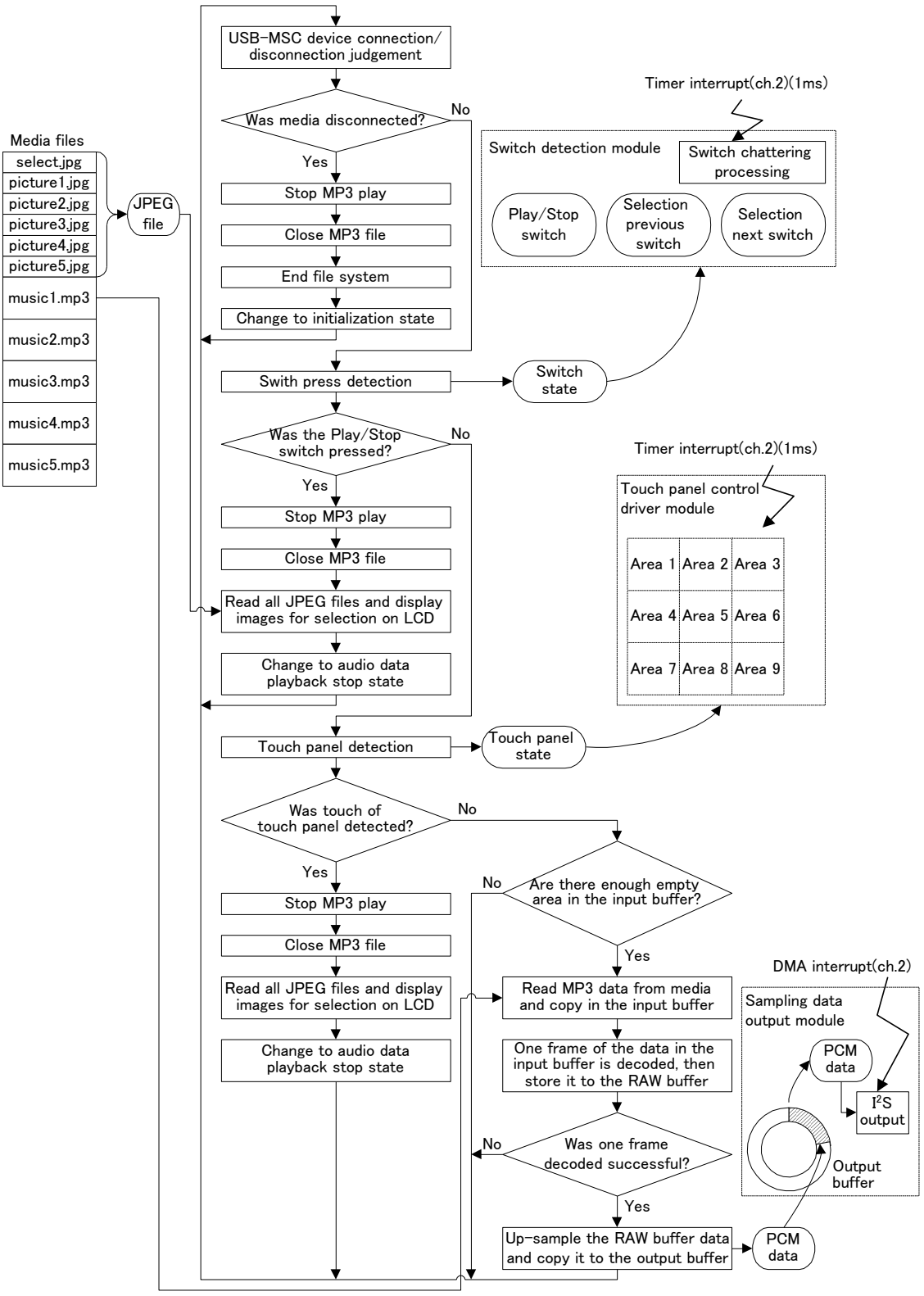

Figure 49 Application Operation Flow (Audio Playback Status, Case of MP3)

7.2.6.2 AAC

(1) The application operation flow with audio data playback stopped is as follows.

① USB MSC device connection/disconnection judgment is executed in the main loop.

② If the USB memory is connected, switch press-down detection is executed.

③ If the play/stop switch is detected to have been pressed down, the AAC file selected from the USB memory is opened. If the selection previous switch or selection next switch are detected to have been pressed down, AAC selection is shifted and LED control is executed.

④ The AAC file header is read, AA file header analysis processing is conducted and operation shifts to audio data playback status.

This operation is shown in Figure 50.

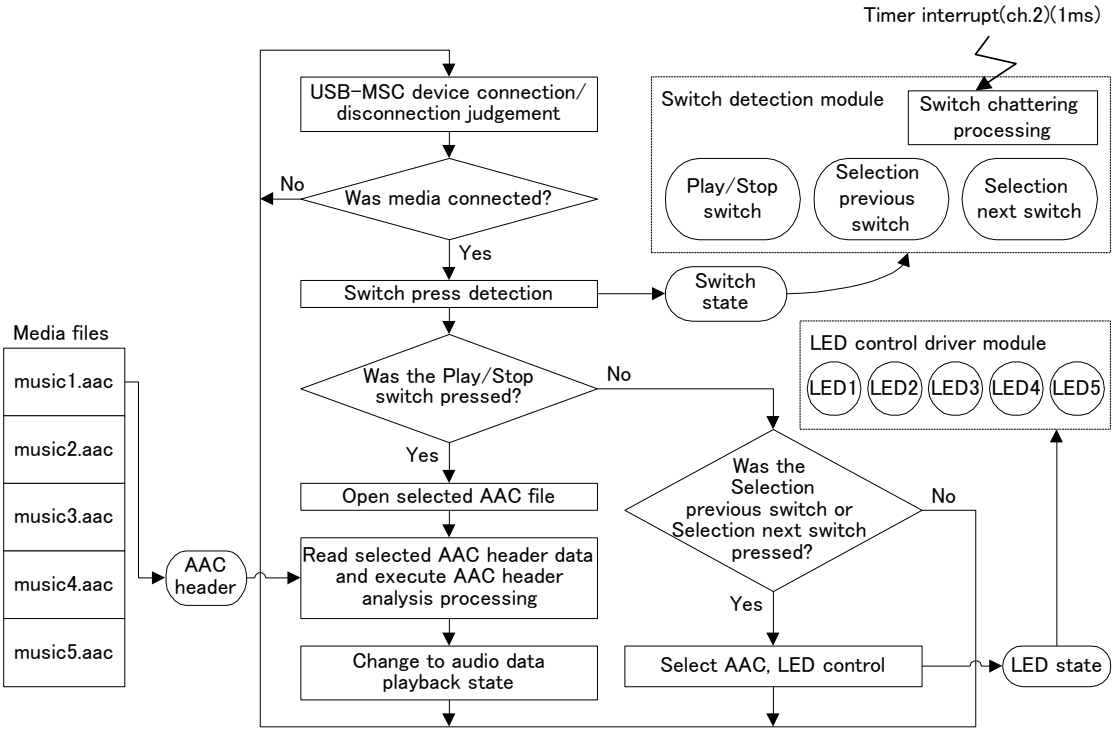

Figure 50 Application Operation Flow (Audio Playback Stopped Status, Case of AAC)

(2) The application operation flow with audio data playback in progress is as follows.

① USB MSC device connection/disconnection judgment is executed in the main loop.

② If the USB memory has been removed, stop playback, close the opened AAC file, quit the file system and operation shifts to initialization status.

③ If the USB memory is connected, play/stop button press-down detection is executed.

④ If the play/stop button is detected to have been pressed down, playback stops, the AAC file is closed, and operation shifts to audio data playback stop status.

⑤ Verify vacancy of input buffer.

⑥ If there is sufficient vacancy, the AAC file is read from the USB memory and copied in the input buffer.

⑦ One elementary stream of the input buffer is decoded and stored in the RAW buffer.

⑧ When 1 elementary stream had been decoded, the RAW buffer is up-sampled and buried in the output buffer.

⑨ With DMA ch2 interrupt, data is sent from the output buffer to I^2S in sequence.

This operation is shown in Figure 51.

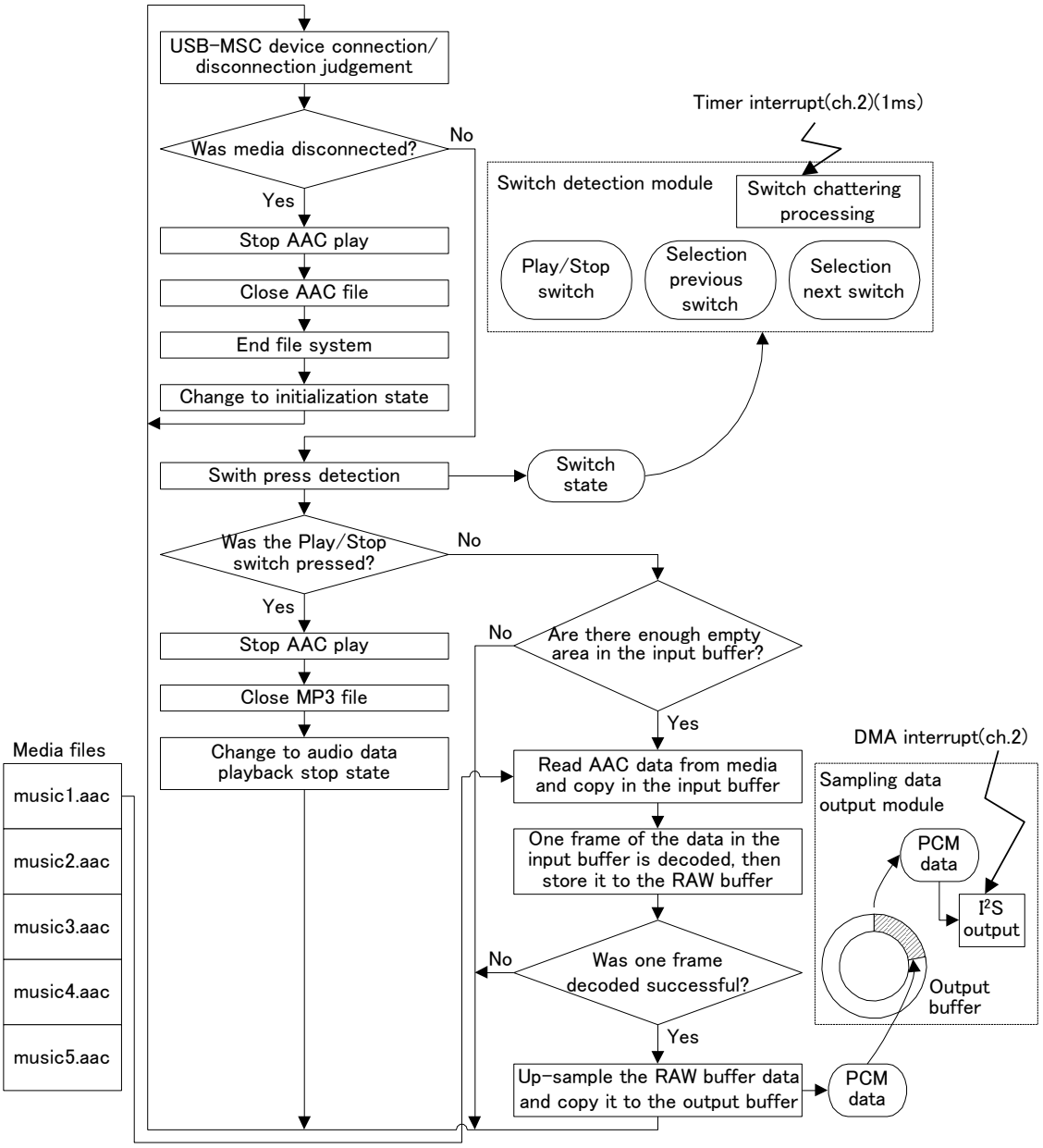

Figure 51 Application Operation Flow (Audio Playback Status, Case of AAC)

7.2.7 Application State Transition

7.2.7.1 MP3

In the case of MP3, the simple AV system consists of the six states shown in Figure 52.

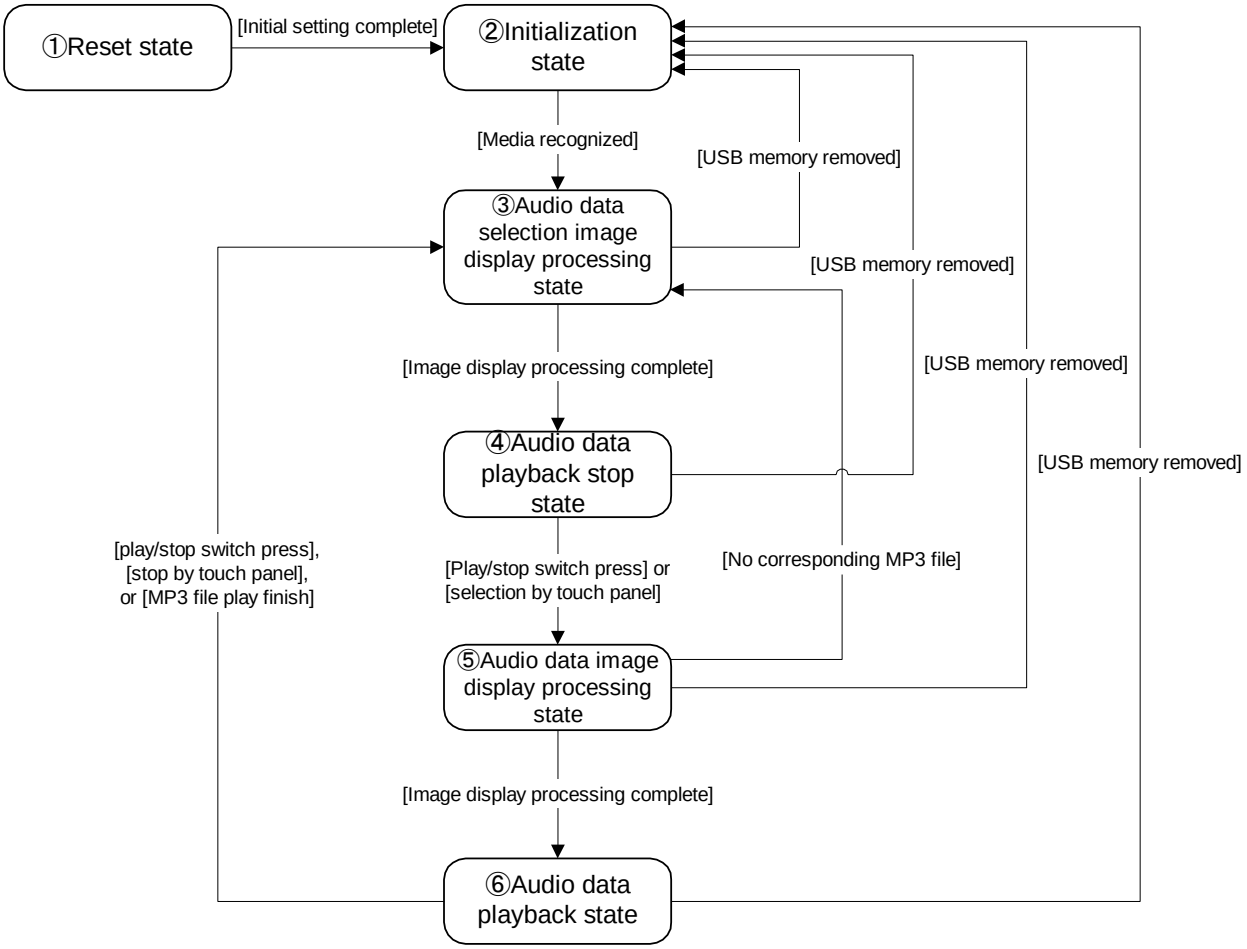

Figure 52 State Transition (MP3)

① Reset State

The simple AV system is in reset state immediately after the power has been turned on.

In the reset state, internal initialization is executed.

② Initialization State

When software internal initialization processing is completed and recognition media (USB memory) is inserted, the simple AV system is in initialization state until the media is recognized.

[Individual State]

Play/stop switch, selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

Does not function even if touched.

LED

All LEDs go off.

LCD

Total screen displayed in white.

USB

State where USB connection connector has not been inserted. When a USB memory has been inserted into the USB memory connector, after being recognized, the system shifts to audio data selection image display processing state.

③ Audio Data Selection Image Display Processing State

After the USB memory is recognized, the system is in audio data selection image display processing state.

Image files are read from the USB memory and images for selecting audio data are displayed on the LCD. After display processing, the system shifts to audio data playback stop state.

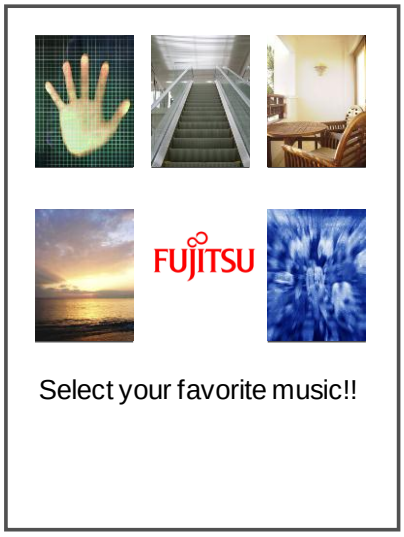

Figure 53 Audio Data Selection Image Display

[Individual State]

Play/stop switch, selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

Does not function even if touched.

LED

LED1 lights.

LCD

Image display is as shown in Figure 53.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, the system shifts to initialization state and waits for the USB to be re-inserted.

④ Audio Data Playback Stop State

After display processing is completed for the LCD in audio data selection image display processing state, the system is in audio data playback stop state.

[Individual State]

Play/Stop switch

With the play/stop switched pressed down, the system shifts to audio data image display processing state.

Selection previous switch, selection next switch

LED 1 to 5 shift lit.

LCD Touch Panel

When the audio data images shown in Figure 53 are touched, the system shifts to audio data image display processing state.

LED

Playback target LED lights. LED 1 lights when USB memory recognition processing is complete.

LCD

Image display is as shown in Figure 53.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, the system shifts to initialization state and waits for the USB to be re-inserted.

⑤ Audio Data Image Display Processing State

When the play/stop switch is pressed down in audio data playback stop state or is selection is made from the touch panel, the system is in audio data image display processing state. Image files are read from the USB memory, and audio data images are displayed on the LCD.

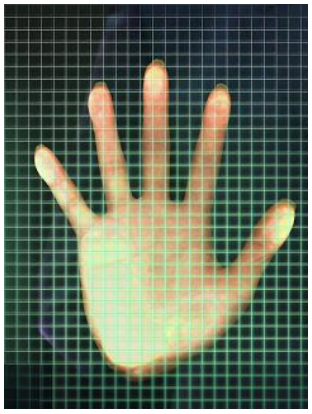

Figure 54 Audio Data Image Display

If the target JPEG file does not exist the entire LCD is displayed in magenta. After display processing is completed for the LCD, if the target MP3 file exists, the system shifts to audio data playback state, If the target MP3 file does not exist, the system shifts to audio data selection image display processing state.

[Individual State]

Play/stop switch, selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

Does not function even if touched.

LED

Playback target LED lights.

LCD

Image display is as shown in Figure 54.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, the system shifts to initialization state and waits for the USB to be re-inserted.

⑥ Audio Data Playback State

After display processing is completed for the LCD in audio data image display processing state, if the target MP3 file exists, the system is in audio data playback stop state.

[Individual State]

Play/Stop switch

Playback is stopped by pressing down the play/stop, and the system shifts to audio data selection image display processing state.

Selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

If the touch panel is touched, playback stops and the system shifts to audio data selection image display processing state.

LED

The LED of the target audio data lights.

LCD

Image display is as shown in Figure 54.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, playback stops and the system shifts to initialization state and waits for the USB to be re-inserted.

7.2.7.2 AAC

In the case of AAC, the simple AV system consists of the four states shown in Figure 55.

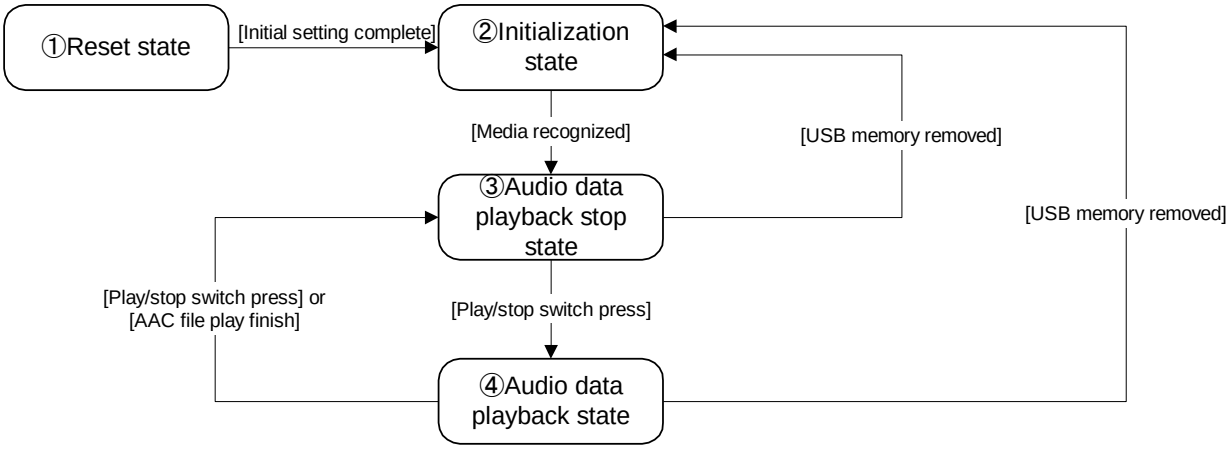

Figure 55 State Transition (AAC)

① Reset State

The simple AV system is in reset state immediately after the power has been turned on.

In the reset state, internal initialization is executed.

② Initialization State

When software internal initialization processing is completed and recognition media (USB memory) is inserted, the simple AV system is in initialization state until the media is recognized.

[Individual State]

Play/stop switch, selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

Does not function even if touched.

LED

All LEDs go off.

LCD

All LEDs go off.

USB

State where USB connection connector has not been inserted. When a USB memory has been inserted into the USB memory connector, after being recognized, the system shifts to audio data playback stop state.

③ Audio Data Playback Stop State

After the USB memory is recognized, the system is in audio data playback stop state.

[Individual State]

Play/Stop switch

With the play/stop switch pressed down, if the target AAC file exists, the system is in audio data playback state.

Selection previous switch, selection next switch

LED 1 to 5 shift lit.

LCD Touch Panel

Does not function even if touched.

LED

Playback target LED lights. LED 1 lights when USB memory recognition processing is complete.

LCD

All LEDs go off.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, the system shifts to initialization state and waits for the USB to be re-inserted.

④ Audio Data Playback State

If the play/stop button is pressed down in the audio data playback state, if the target AAC file exists, the system is in audio data playback state.

[Individual State]

Play/Stop switch

Playback is stopped by pressing down the play/stop, and the system shifts to audio data playback stop state.

Selection previous switch, selection next switch

Does not function even if pressed down.

LCD Touch Panel

Does not function even if touched.

LED

The LED of the target audio data lights.

LCD

All LEDs go off.

USB

State where USB memory is inserted in the USB memory connector. If the USB memory is removed in this state, playback stops and the system shifts to initialization state and waits for the USB to be re-inserted.

7.2.8 Operation Flow

The meanings of the function call points in the flowchart are shown in Figure 56.

Figure 56 Meanings of function call points

7.2.8.1 Main Processing Function

The flowchart of the main processing function (main) is shown in Figure 57.

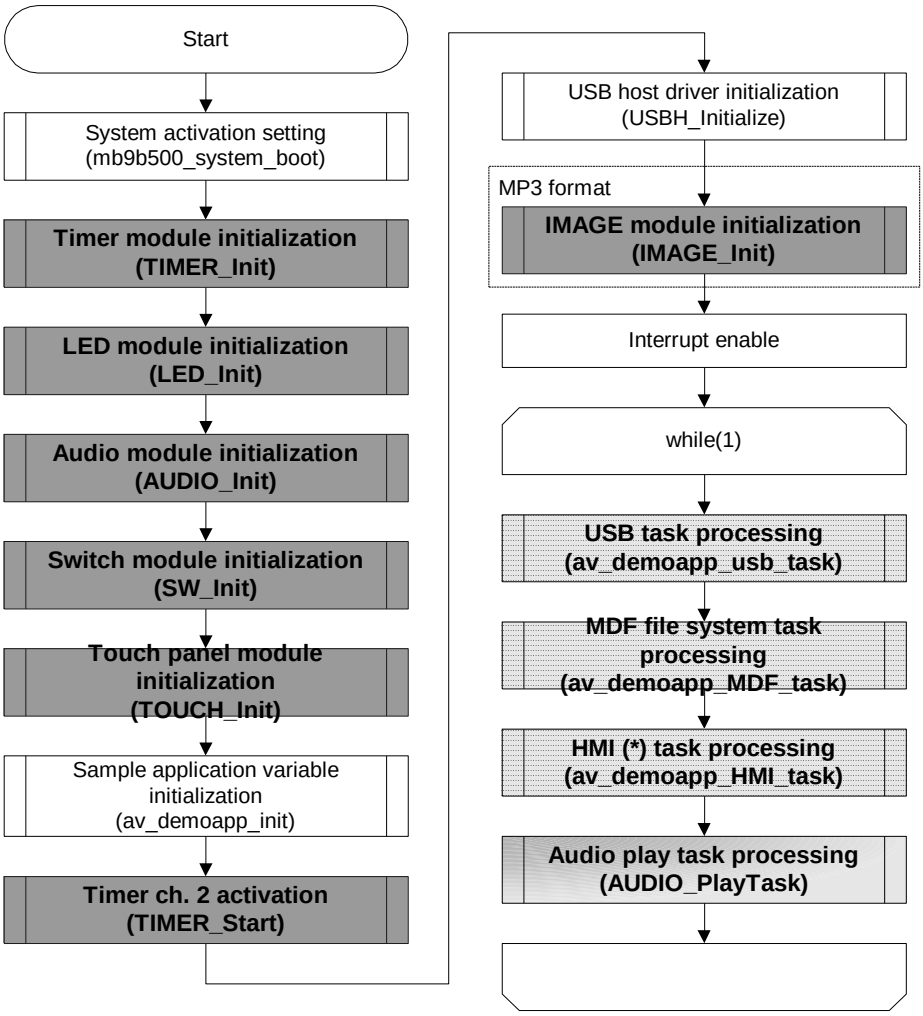

Figure 57 Main Processing Function Flow

(*) Human Machine Interface

7.2.8.2 USB Task Processing Function

After detecting USB device connection/disconnection, the processing shown in Figure 58 is executed.

(av_demoapp_usb_task)

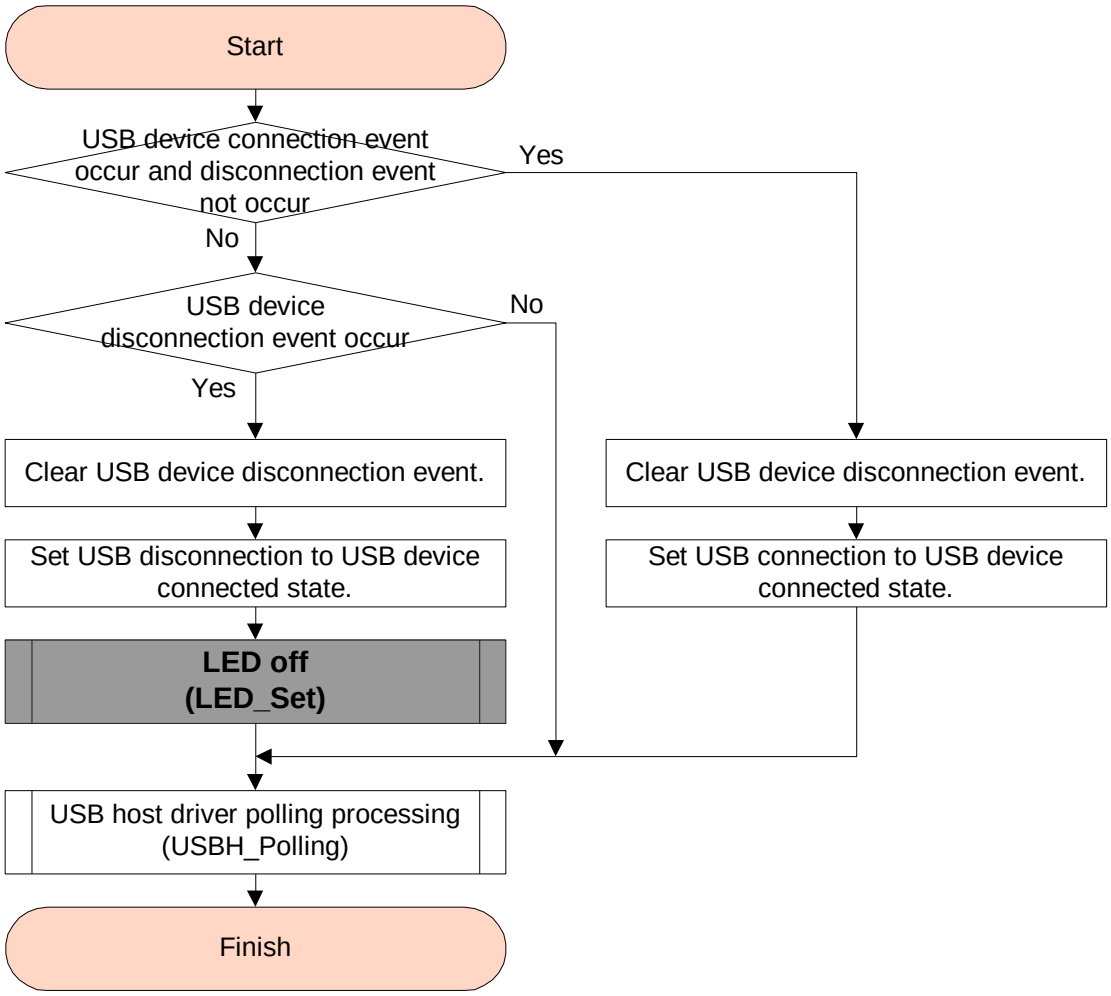

Figure 58 USB Task Processing Function Flow

7.2.8.3 File System (MDF) Task Processing Function

File System (MDF) processing is executed. (av_demoapp_MDF_task)

After reading MBR (*1) and PBR (*2) data, the number of blocks for each sector is calculated.

When the MDF file system is initialized, media read and write processing is registered.

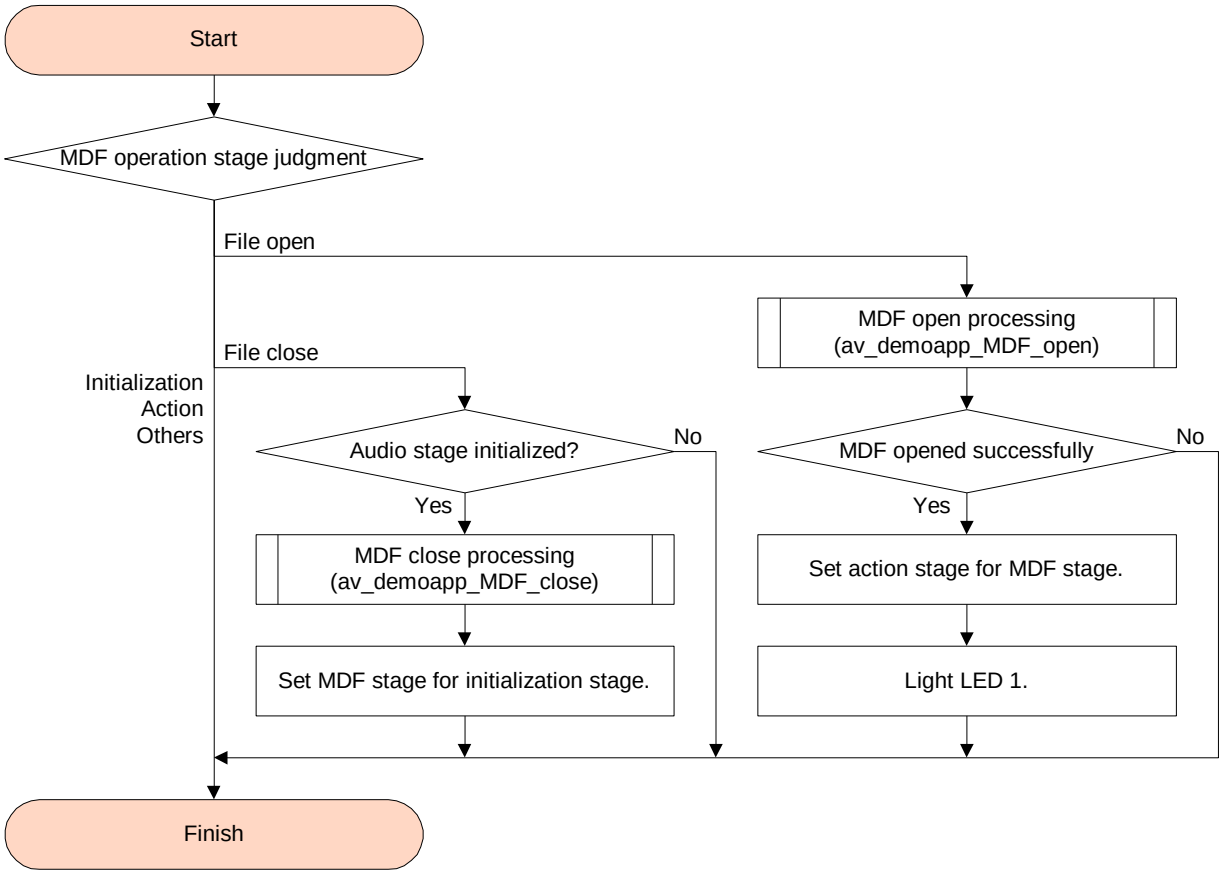

Figure 59 File System (MDF) Task Processing Function

(*1) Master Boot Record

(*2) Partition Boot Record

7.2.8.4 HMI Task Processing Function

HMI processing is executed. (av_demoapp_HMI_task) LCD display, switch detection touch panel detection, etc., user interface processing is executed.

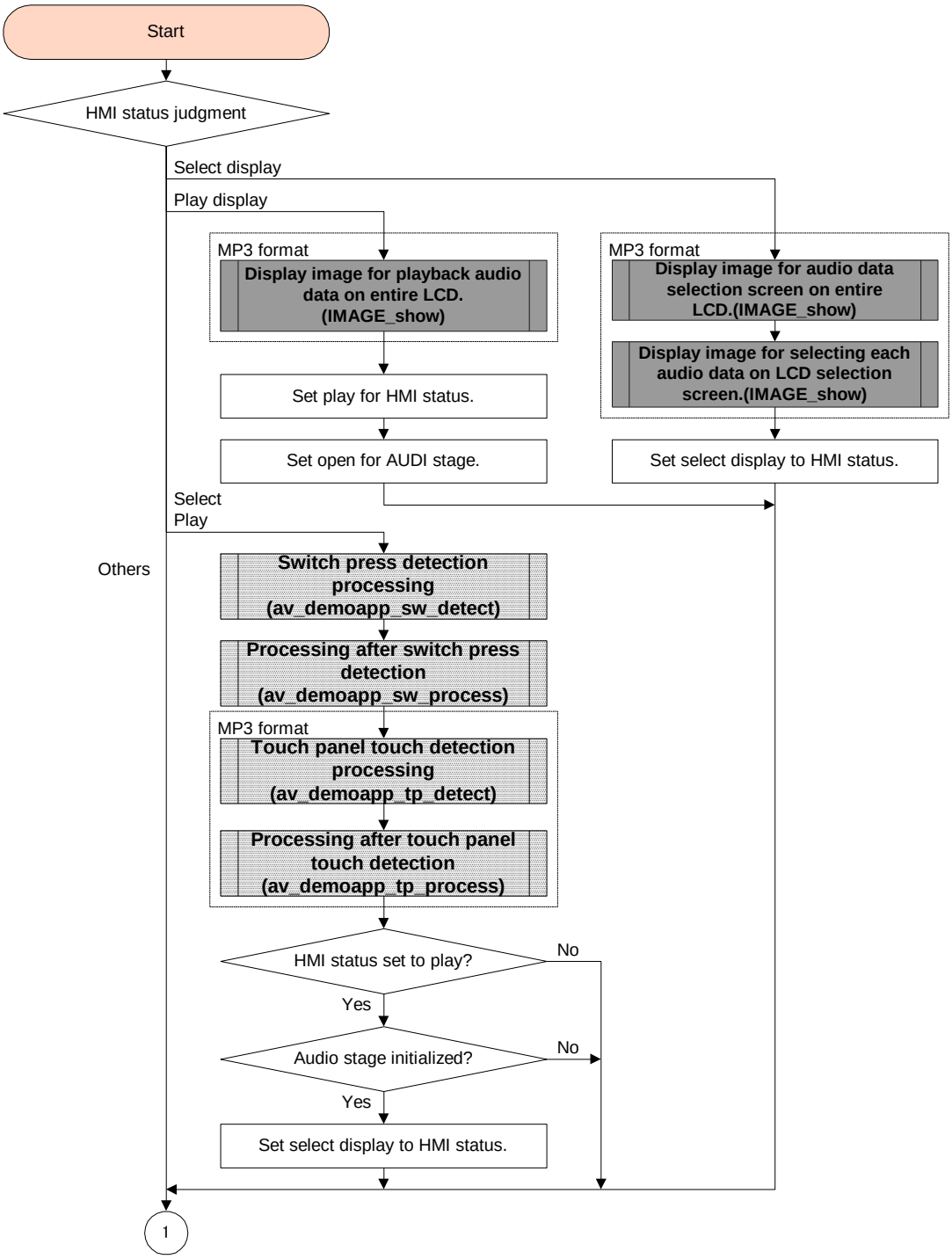

Figure 60 HMI Task Processing Function Flow (1)

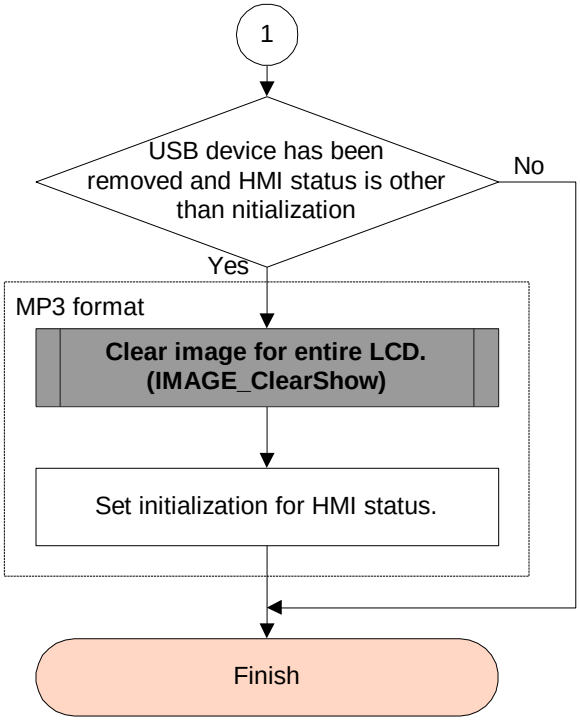

Figure 61 HMI Task Processing Function Flow (2)

7.2.8.5 AUDIO Play Task Processing Function

AUDIO Play processing is executed. (AUDIO_PlayTask)

Audio data read from the USB memory using the MDF file system is decoded and played.

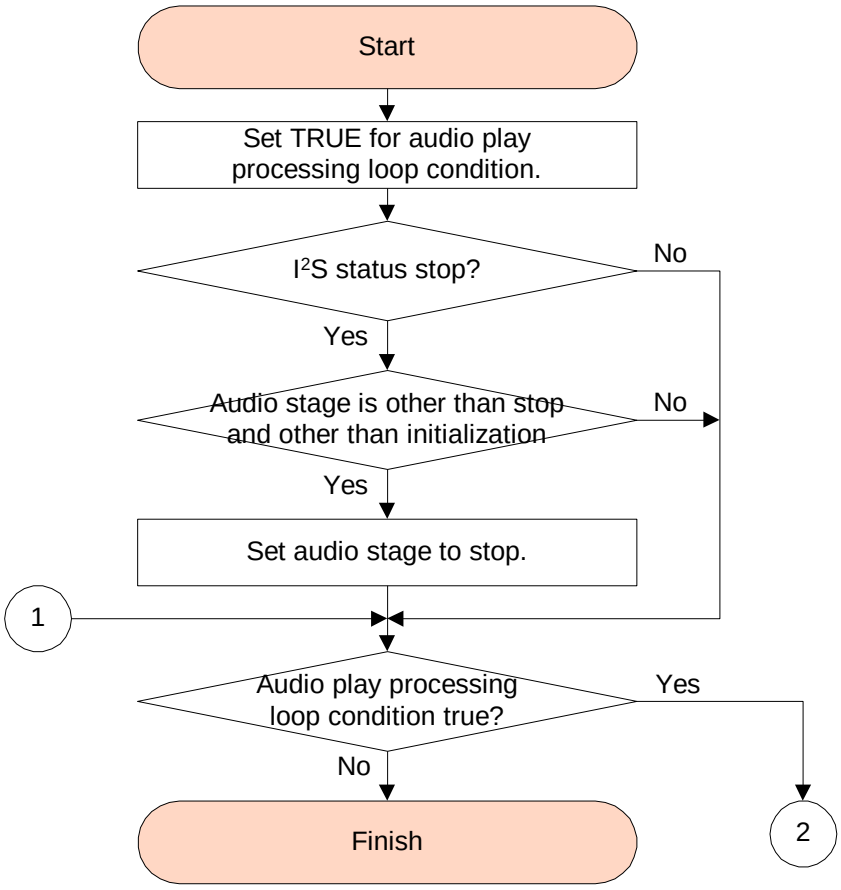

Figure 62 AUDIO Task Processing Function Flow (1)

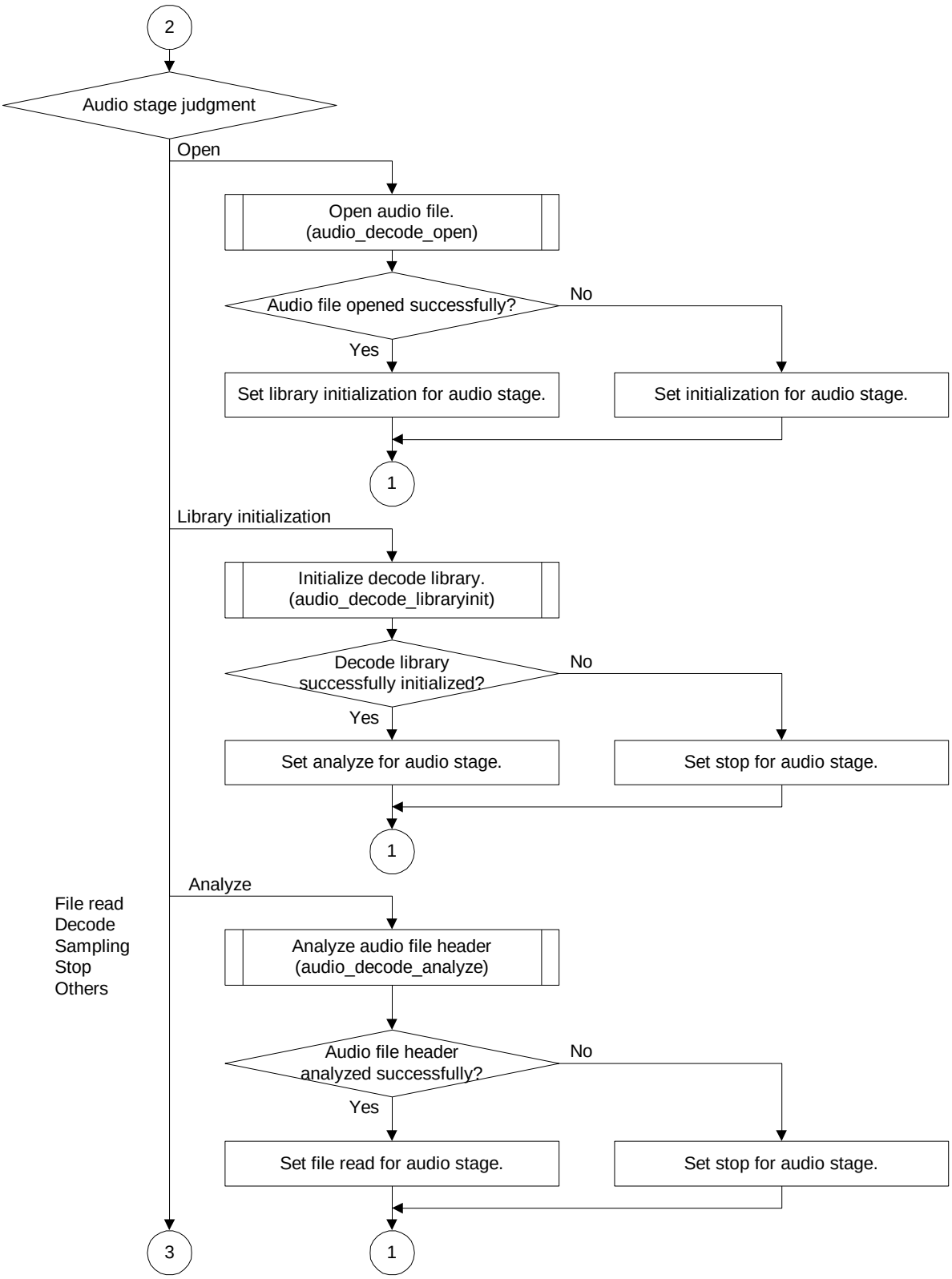

Figure 63 AUDIO Task Processing Function Flow (2)

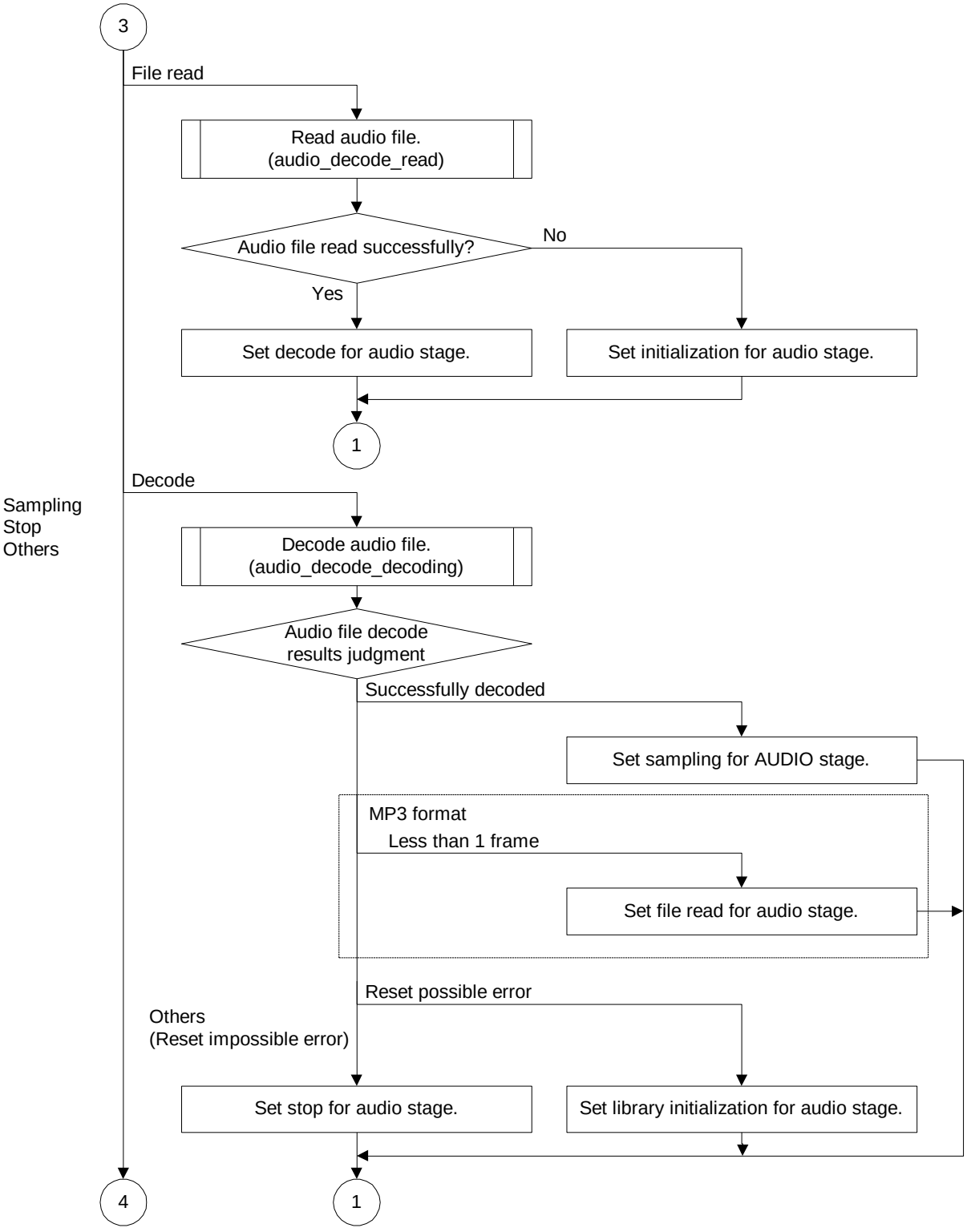

Figure 64 AUDIO Task Processing Function Flow (3)

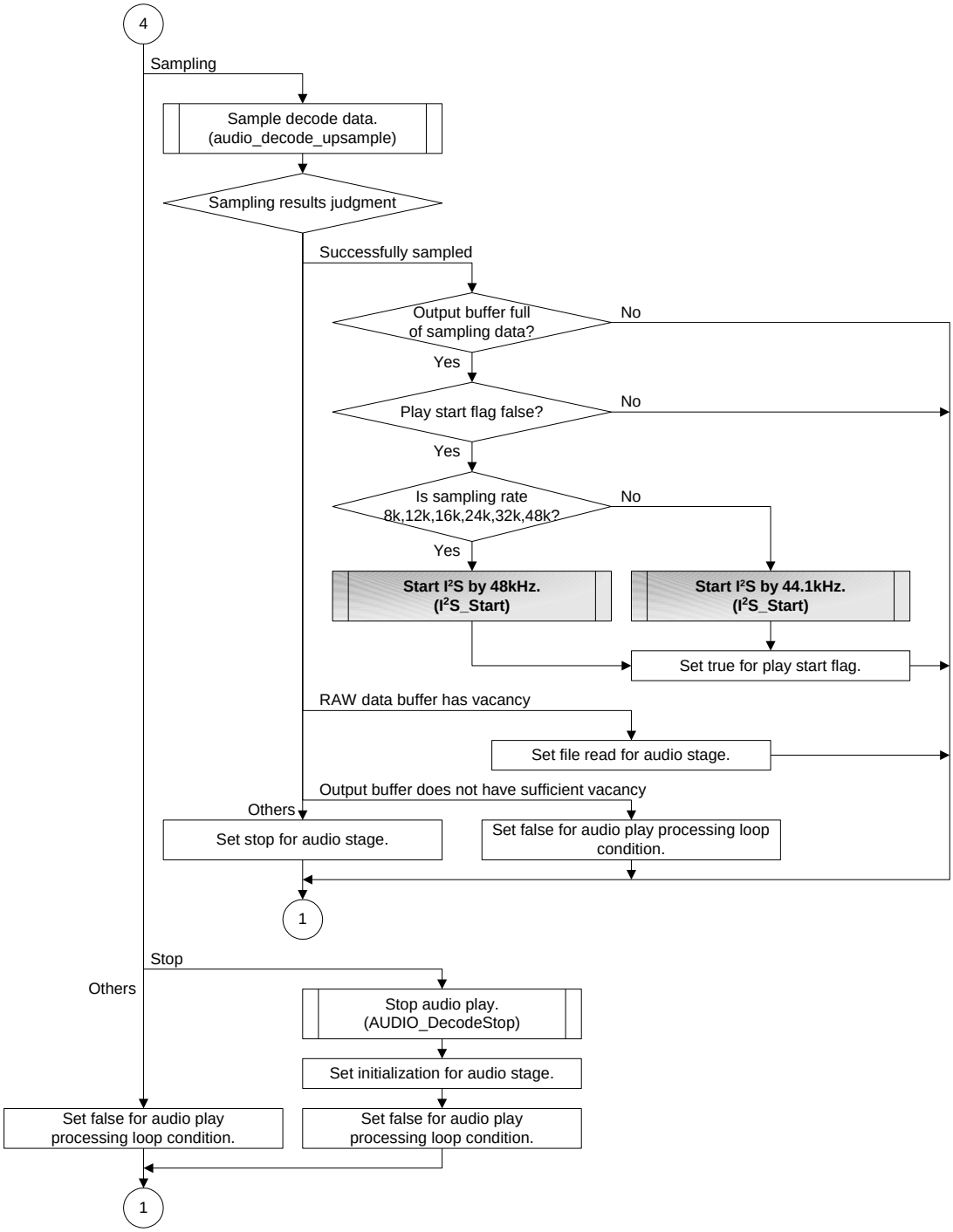

Figure 65 AUDIO Task Processing Function Flow (4)

7.2.8.6 Switch Press Detection Processing Function

Switch press detection processing is executed. (av_demoapp_sw_detect)

Play/stop switch, selection next switch, selection previous switch press detection is executed.

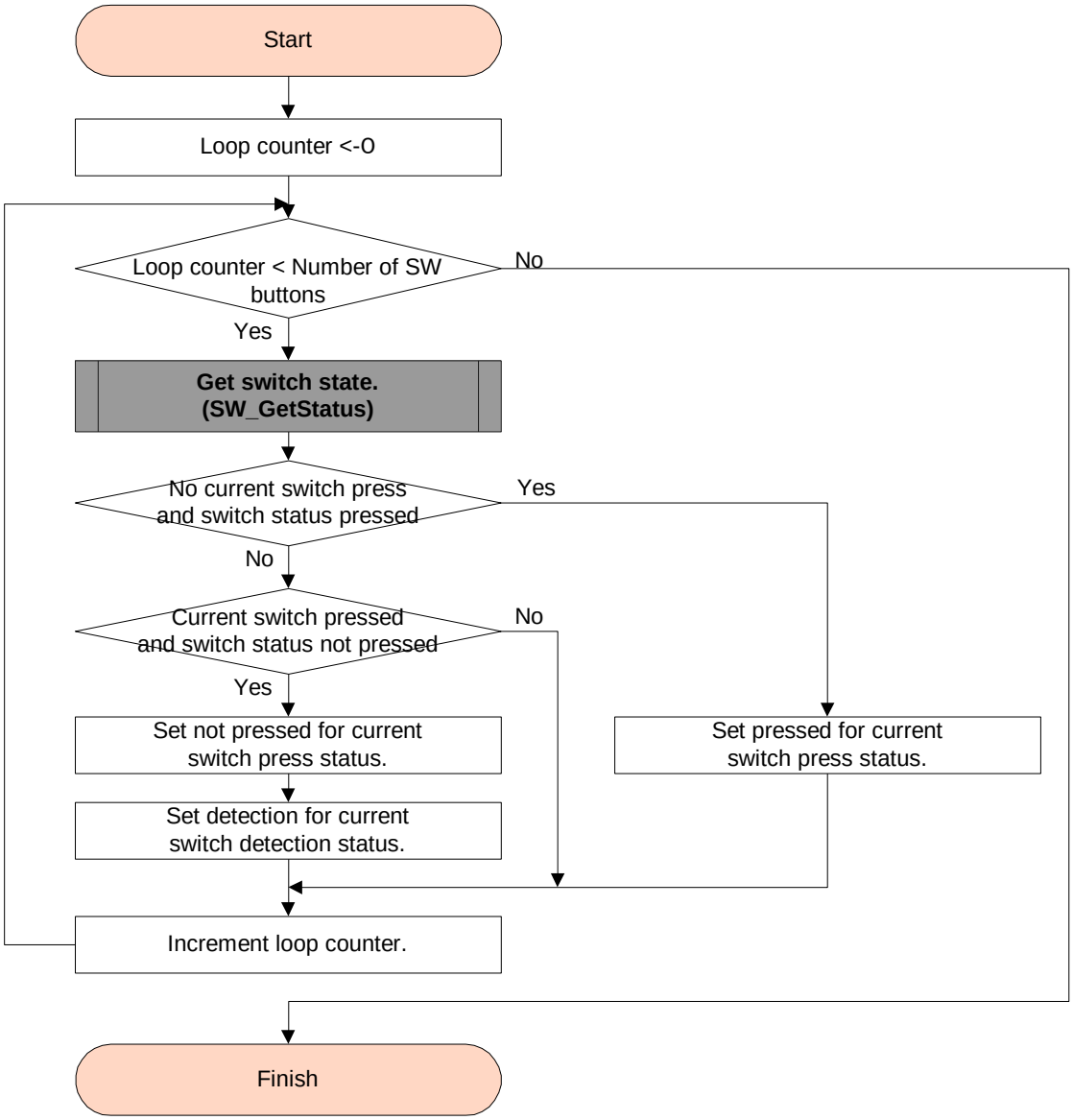

Figure 66 Switch Press Detection Processing Function Flow

7.2.8.7 Processing Function After Switch Press Detection

Processing after switch press detection is executed. (av_demoapp_sw_process)

Play/stop switch, selection next switch, selection previous switch press detection results are checked and if pressing has been detected, respective switch processing is executed.

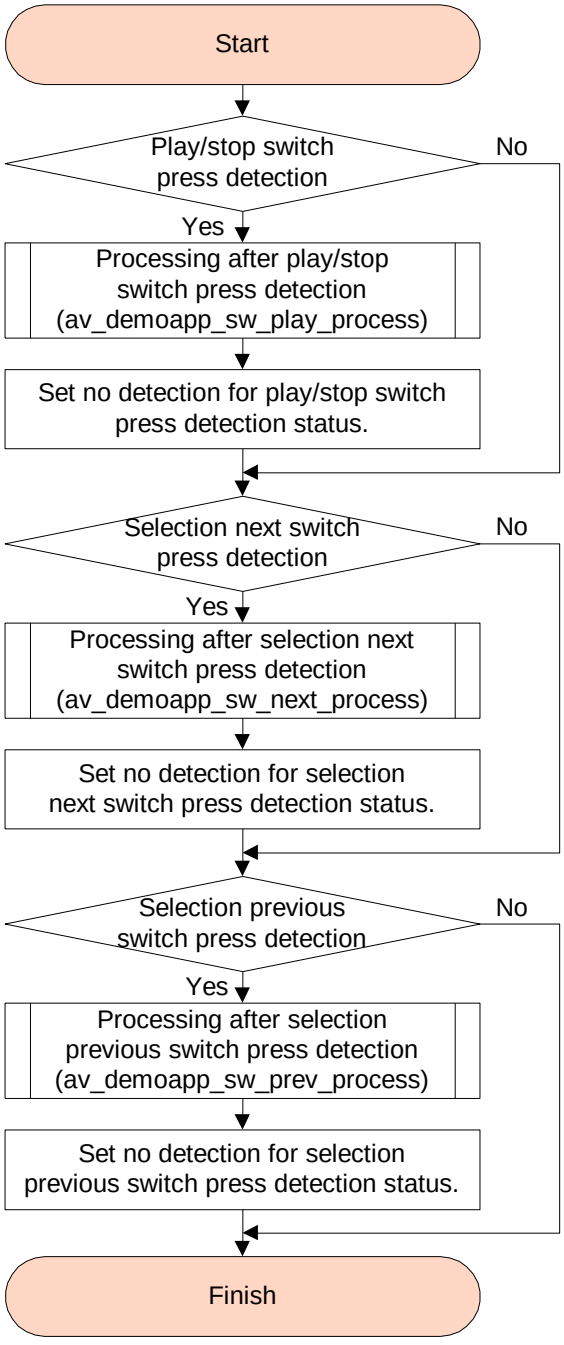

Figure 67 Processing After Switch Press Detection Function Flow

7.2.8.8 Touch Panel Touch Detection Processing Function

Touch panel touch detection processing is executed. (av_demoapp_tp_detect)

Detects which position of the nine areas (see Figure 47) on the LCD has been touched.

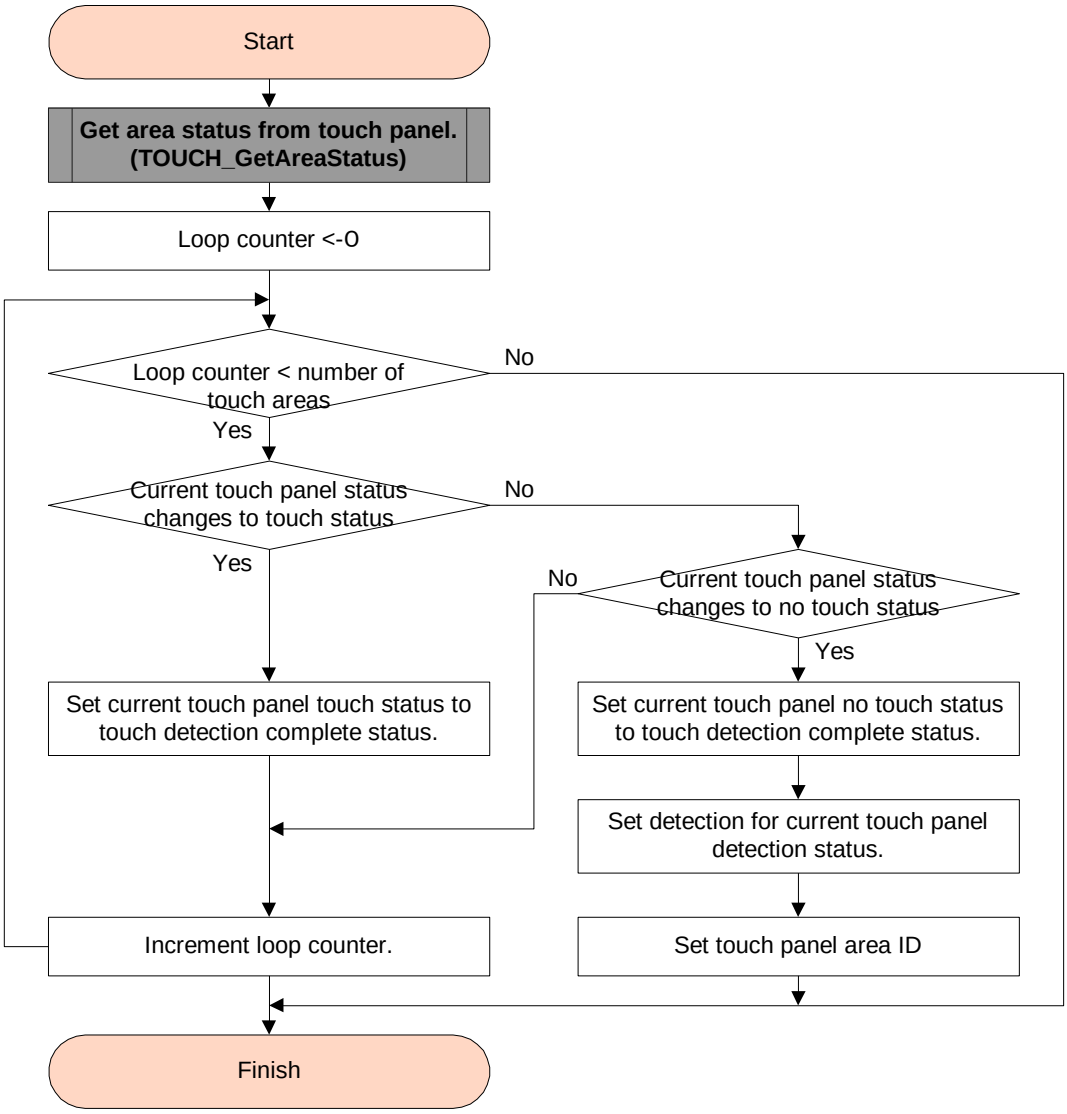

Figure 68 Touch Panel Touch Detection Processing Function Flow

7.2.8.9 Processing After Touch Panel Touch Detection Function

Processing after touch panel touch detection is executed. (av_demoapp_tp_process)

Area state detection results of the touch panel are checked and the processing shown in Figure 69 is executed.

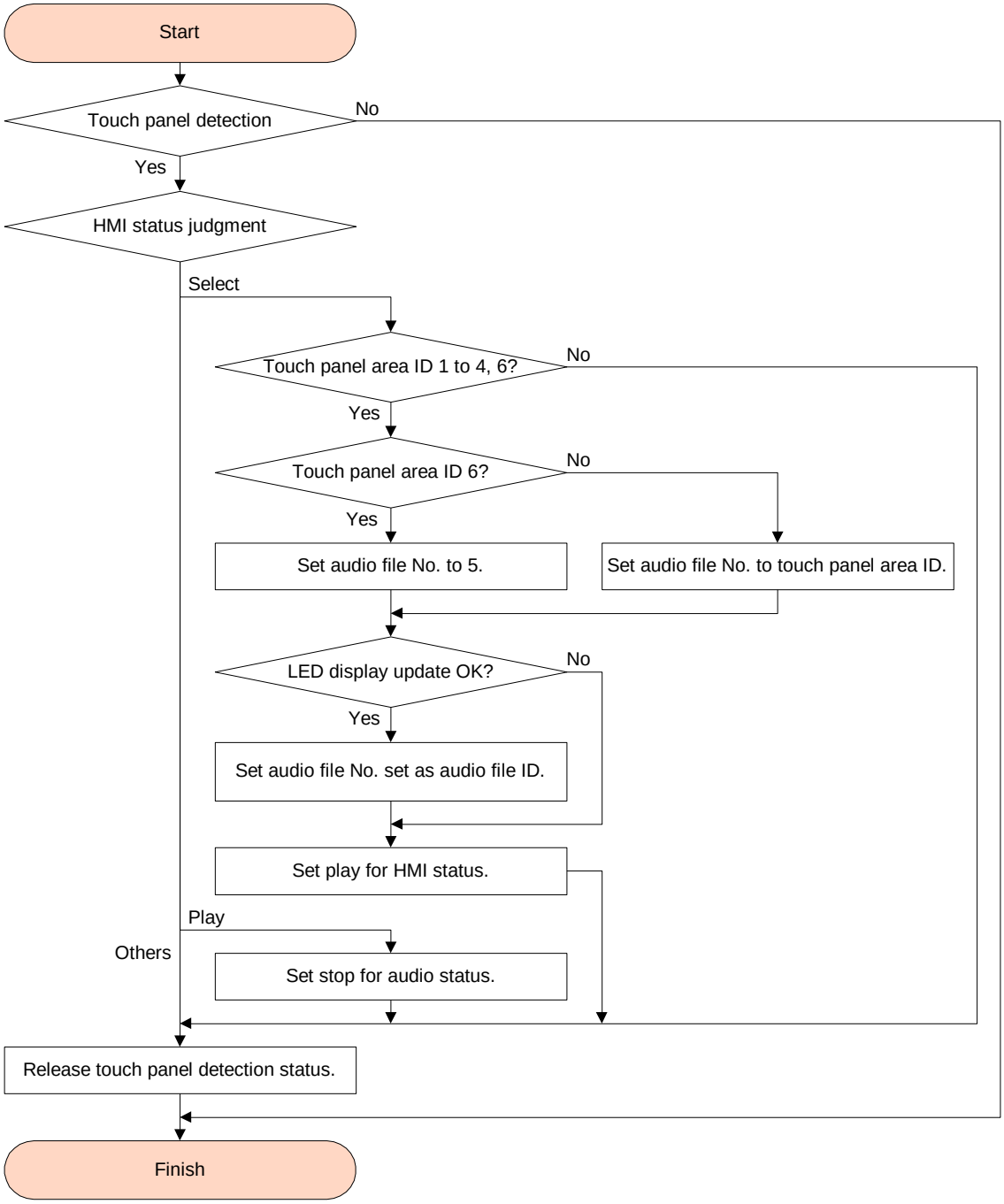

Figure 69 Processing After Touch Panel Touch Detection Function Flow

7.2.9 File Configuration

The following is a software development environment file configuration provided as a sample.

```
+---common                  : Common header file directory
   +---core_cm3.h     : Peripheral access definition source file
+---middle                  : Middleware directory (*)
   +---aac                    : AAC decoder library directory (*)
   +---filesystem     : File System (MDF) library directory (*)
   +---jpeg                   : JPEG decoder library directory (*)
   +---mp3                    : MP3 decoder library directory (*)
+---project                 : Project directory
   +---AV_demo.uvopt          : Option file
   +---AV_demo.uvproj         : Project file
   +---startup_mb9bf50x.s     : Startup assembler file
+---source                  : Sample source directory
   +---application      : Application directory
      +---hw_support.c            : Hardware-dependent processing source file
      +---hw_support.h            : Hardware-dependent processing header file
      +---av_demoapp.c            : Application processing source file
      +---fs_service.c            : File system service source file
      +---sys_mem.c               : System memory processing source file
      +---sys_timer.c             : System timer processing source file
      +---IRQ.c                   : Interrupt processing source file
   +---application_if          : Application interface directory
      +---audio                   : Audio play processing directory
         +---audio.c          : Audio play processing source file
         +---upsample.c             : Up-sampling control processing source file
         +---upsample.h             : Up-sampling control processing header file
      +---image                   : Image display processing directory
         +---image.c                : Image display processing source file
      +---led_sw                  : LED control / switch detection processing directory
         +---led.c                  : LED control processing source file
         +---switch.c               : Switch detection processing source file
```

+---config : User setting file directory
+---usbh_config.c : User setting source file for USB host driver
+---audio_config.h : User setting header file for audio play processing
+---i2s_config.h : User setting header file for I^2S control driver
+---led_config.h : User setting header file for LED control processing
+---switch_config.h : User setting header file for switch detection processing
+---timer_config.h : User setting header file for timer control driver
+---touch_config.h : User setting header file for touch panel control driver
+---usbh_config.h : User setting header file for USB host driver
+---usbh_msc_config.h : User setting header file for USB host mass storage class (MSC) driver
+---drivers : Driver directory
+---i2s : I^2S control driver directory
+---i2s.c : I^2S control driver source file
+---lcd : LCD control driver directory
+---lcd.c : LCD control driver source file
+---lcd_ili9325.c : ILI9325 control processing source file
+---lcd_ili9325.h : ILI9325 control processing header file
+---timer : Timer control driver directory
+---timer.c : Timer control driver source file
+---touch : Touch panel control driver directory
+---touch.c : Touch panel control driver source file

```
+---usb                  : USB host driver directory
   +---usbh_api.c        : USB host driver API source file
   +---usbh_atch.c       : USB connection/disconnection processing source file
   +---usbh_hc.c         : USB host controller processing source file
   +---usbh_mgr.c        : USB host manager processing source file
   +---usbh_mh_core.c    : USB-Mini host controller driver
                           Core processing source file
   +---usbh_mh_dma.c     : USB-Mini host controller driver
                           DMA processing source file
   +---usbh_mh_hal.c     : USB-Mini host controller driver
                           Hardware physical control processing source file
   +---usbh_req.c        : USB host request processing source file
   +---usbh_api.h        : USB host driver API header file
   +---usbh_atch.h       : USB connection/disconnection processing header file
   +---usbh_hc.h         : USB host controller processing header file
   +---usbh_mgr.h        : USB host manager processing header file
   +---usbh_mh_core.h    : USB-Mini host controller driver
                           Core processing header file
   +---usbh_mh_dma.h             : USB-Mini host controller driver
                           Core processing header file
   +---usbh_mh_hal.h     : USB-Mini host controller driver
                           Hardware physical control processing header file
   +---usbh_req.h        : USB host request processing header file
   +---usb_msc           : USB host mass storage class (MSC) driver
                           directory
      +---usbh_msc_api.c    : MSC driver processing API source file
      +---usbh_msc_bot.c    : Bulk Only Transfer processing source file
      +---usbh_msc_api.c    : Manager processing API source file
      +---usbh_msc_scsi.c   : SCSI command processing source file
      +---usbh_msc_api.h    : MSC driver processing API header file
      +---usbh_msc_bot.h    : Bulk Only Transfer processing header file
      +---usbh_msc_api.h    : MSC manager processing API header file
      +---usbh_msc_scsi.h   : SCSI command processing header file
```

+---include : Include file directory
  +---audio.h : Audio play processing header file
  +---avdemoapp.h : Application processing header file
  +---common.h : Common definition header file
  +---gpio.h : GPIO definition header file
  +---i2s.h : I^2S control driver header file
  +---cpu : CPU fixed definition header file directory
  +---image.h : Image display processing header file
  +---lcd.h : LCD control driver header file
  +---led.h : LED control processing header file
  +---switch.h : Switch detection processing header file
  +---system_dependence.h : System-dependent processing header file
  +---timer.h : Timer control driver header file
  +---touch.h : Touch panel control driver header file
  +---typedef.h : Type definition header file
  +---usbh.h : USB host driver header file
  +---usbh_msc.h : USB host mass storage class (MSC) driver header file
  +---cpu : CPU-dependent header file directory
    +---cpu_config.h : CPU setting header file
    +---mb9bf50x : MB9BF500 Series definition header file directory
      +---cpu_define.h : CPU definition header file
      +---cpu_define_mb9bf50x.h : MB9BF500 Series CPU definition header file
      +---cpu_map_mb9bf50x.h : MB9BF500 Series register bit definition header file
      +---mb9bf50x.h : MB9BF500 Series interrupt vector, register definition header file

(*) Only directory provided.
Library is provided by separate contract.

-End-