MA-3 Sound Middleware
Sample Source
Installation Guide

Ver.1.3.15.5

April 22, 2003

Yamaha Corporation

[Notes]
This document is the installation guide of MA-3 Sound Middleware as sample source code.
This explains the installation guide of Sound Middleware, but doesn’t guarantee operation of sample middleware.

Contents

Revision

Ver.	Date	Description
0.50	April 23, 2001	Initial edition
0.80	June 1, 2001	ma3cnvprf.h, ma3phrcnv.h and ma3phrcnv.c were added to the file configuration. Example of registration of Stream Converter was deleted. Description of ma3cnvprf.h was added. Description of MA3_VSEL was added.
0.90	June 18, 2001	ma3rmdcnv.h, ma3rmdcnv.h, ma3audcnv.h and ma3audcnv.c were added to the file configuration. Machdep_CheckStatus () and machdep_WaitValidData () were added.
1.00	June 29, 2001	Converters were deleted from the file configuration. File name was changed.
1.1.0	July 11, 2001	Only change of applicable versions
1.2.0	Oct.05.2001	3.7 Cancellation processing of delayed recovery function was added.
1.3.0	Nov. 14, 2001	3.8 Measure for reduction of processing load at implementation was added.
1.3.2	Nov. 14, 2001	3.2.4 Setting of sleep processing function was added. 3.2.5 Setting of delay system FIFO Empty check processing function was added. 3.2.6 Setting of delay system FIFO Empty waiting processing function was added. 3.2.7 Setting of real time system FIFO Empty waiting processing function was added.
1.3.2.7	Nov. 30, 2001	3.1.1 Setting of sleep processing at stop was added. 3.1.2 Setting of address of MA-3 access port on the system was changed. 3.1.3 Setting of time at MA-3 status check was added. 3.8.2 Arrangement of functions in the built in RAM was changed.
1.3.4.2	Dec. 20, 2001	MA_PLAYMODE_CHECK was added.
1.3.6.5	Feb.27, 2002	3.1.6 The setting of the 3 of MA3_VSEL in the speaker amplifier part center voltage was deleted. 3.6.1 Clerical error was corrected.
1.3.7.5	Mar.27, 2002	3.5 The description about LED / Motor control was added.
1.3.8.1	May.29, 2002	3.1.6 The clerical error used as intermediate register BANK=1 ID#9 was corrected to BANK=0 ID#10 in the explanation of MA_VSEL. 3.1.8 The setting change of FM operation mode of SMF and RealtimeMIDI was added. 3.1.9 The setting change of NoteOn Velocity curve of SMF and RealtimeMIDI was added.
1.3.8.2	September 20, 2002	3.1.10 Change of KCS interpretation setup of SMAF/MA-1 was added.
1.3.11.5	September 30, 2002	Version was changed in accordance with source version number.
1.3.12.3	November 15, 2002	Version was changed in accordance with source version number.
1.3.13.6	December 13, 2002	Version was changed in accordance with source version number.
1.3.14.1	January 16, 2003	Version was changed in accordance with source version number.
1.3.15.5	April 22, 2003	3.1.10 It was changed from SMAF/MA-1 to SMAF/MA-1/MA-3.

1 Introduction

This document provides supplementary explanations for installation of Sample Codes Ver.1.3.15.5 of Sound Middleware for mobile audio LSI "MA-3" (hereafter called MA-3).

2 File configuration

This section presents the file configuration of this Sample Source Codes.

File name	Content
mamachdep.h mamachdep.c	Definition of system dependency System dependent function group
masound.h	Definition of MA Sound API
madefs.h	Definition of MA-3 related constants
madebug.h	Definition of debug message output control
madebug.c	MA-3 Hardware file Format type data output function group for debugging
matable.h	Definition of arrangement data
macnvprf.h	Definition of Stream Converter
masndseq.h masndseq.c	Definition of MA Sound Sequencer module Body of MA Sound Sequencer module
masnddrv.c masnddrv.h	Definition of MA Sound Driver module Body of MA Sound Driver module
maresmgr.h maresmgr.c	Definition of MA Resource Manager module Body of MA Resource Manager module
madevdrv.c madevdrv.h	Definition of MA Device Driver module Body of M A Device Driver module

3 System dependency

For Sound Middleware, principle parts that depends on the system are collected in mamachdep.h and mamachdep.c. Perform setting and/or change in accordance with the system conditions.

3.1 mamachdep.h

3.1.1 Setting of Sleep processing at stop

When API require and interrupt request are processed in parallel by task processing, sleep processing is preformed by the stop processing at API request processing side, and stop processing side performs waiting processing until the interrupt processing stops. On the other hand, when API require and interrupt request are processed sequentially, set MA_STOPWAIT to (0).

Name	Value to be set
MA_STOPWAIT	0: Sleep processing is not performed at stop processing. 1: Sleep processing is performed at stop processing (default).

3.1.2 Error processing of playback times setup at the start

When the number of playback times specified at the first start after standby is called again with the value from which the number of times differs before next standby at the start, designates to ignore it or to make an error.

Name	Value to be set
MA_PLAYMODE_CHECK	0: Ignores (Default). 1: Makes an error.

3.1.3 Setting address of MA-3 access port on the system

The access to MA-3 is executed through 8-bit parallel interface. Sets the address of the port on the system.

Name	Value to be set
MA_STATUS_REG	Value of address of MA-3 status flag register (Adr #0)
MA_DATA_REG	Value of address of MA-3 write / read data register (Adr #1)

This function assumes memory mapped I/O. If not so, it is necessary to add codes for access to the following functions of MA Driver.

Function name	Content of processing
MaDevDrv_WriteStatusFlagReg MaDevDrv_ReadStatusFlagReg	Writes 1 byte data into status flag register. Reads 1 byte data from status flag register.
MaDevDrv_WriteDataReg MaDevDrv_ReadDataReg	Writes 1 byte data into write data register. Reads 1 byte data from read data register.

3.1.4 Setting of time-out at MA-3 status check

Time out is to be set at status check. Set this time to 1.5 ms or over.

Name	Value to be set
MA_STATUS_TIMEOUT	Value that makes the time out 1.5ms.

3.1.5 Setting PLL output frequency in MA-3

To perform setting of PLL output frequency in MA-3, it is necessary to perform clock setting for MA-3 intermediate registers BANK=1 ID#5 and #6. It is necessary to set values corresponding to the frequency of signal connected to CLKI terminal of MA-3.

Name	Value to be set
MA_ADJUST1_VALUE	Value to be set in Intermediate registers BANK=1 ID #5. Bit 4 through bit 0 are valid.
MA_ADJUST2_VALUE	Value to be set in Intermediate register BANK=1 ID #6. Bit 6 through bit 0 are valid.
MA_PLL_OUT	Value of PLL output (Example: Set 55295 when the frequency is 55.295 MHz.

3.1.6 Setting center voltage of speaker amplifier section

To set the center voltage with which the speaker amplifier in MA-3 operates, it is necessary to set for MA-3 Intermediate register BANK=0 ID#10. It is set by using a combination of voltages VDD and SPVDD.

Name	Value to be set
MA_VSEL	Value to be set for intermediate register BANK=0 ID #10 VSEL1, 2. bit 1 ~ bit 0 are valid.

MA_VSEL	SPVDD (V)	VDD (V)	SPAMP center potential (V)
0	3.6	3.0	0.6 * VDD
1	3.0	3.0	0.5 * VDD
2	3.6	2.7	0.67 * VDD

3.1.7 Definition of environmental dependency

The type that is used by each function is used by using the following definition. It is necessary to define them in accordance with the environment.

Definition type	Meaning	Definition type	Meaning
UINT8	8 bits without code	SINT8	8 bits with code
UINT16	16 bits without code	SINT16	16 bits with code
UINT32	32 bits without code	SINT32	32 bits with code

3.1.8 The setting change of FM operation mode of SMF and RealtimeMIDI

The setting of FM operation mode of SMF and RealtimeMIDI can be changed from the default FM4OP 16 tone mode into FM2OP 32 tone mode.

Name	Value to be set at change
RMD_DEF_FM_MODE	Can be changed into FM_2OP_MODE.
SMF_DEF_FM_MODE	Can be changed into FM_2OP_MODE.

3.1.9 The setting change of NoteOn Velocity curve of SMF and RealtimeMIDI

The setting of NoteOn Velocity curve of SMF and RealtimeMIDI can be changed from the default 20 * Log(vel/127) into 40 * Log(vel/127).

Name	Value to be set at change
RMD_DEF_VELOCITY_MODE	Can be changed into VELOCITY_40LOG_MODE.
SMF_DEF_VELOCITY_MODE	Can be changed into VELOCITY_40LOG_MODE.

3.1.10 The setting change KCS interpretation of SMAF/MA-1/MA-3

The interpretation of KCS (Key Control Status) in SMAF/MA-1/MA-3 data can be changed. When the following definition is set to “0”, KCS is referred to as usual and Key Control is controlled. In SMAF/MA-1, key control is interpreted as ON, and in SMAF/MA-3, key control is interpreted as Nonspecific.

Name	Value to be set at change
MA13_KCS_IGNORE	0: Usual (normal) interpretation. (Default) 1: KCS is disregarded.

Note) When the above definition is set to “1”, be careful since key control becomes effective also for drum sound in SMAF/MA-1.

* When key control is operated on drum sound, the voice itself may collapse and tone generation may change extremely.

3.2 mamachdep.c

3.2.1 Setting of Wait processing function

In the procedure of power management, wait processing for some time is needed. Function that performs wait for specified time is defined as follows.

```
void machdep_Wait  ( UINT32 time );
```

Description

Performs wait processing for designated time.

Argument

time	Designates wait time [ns]。

Returned value

None

3.2.2 Setting of Status check processing function

Verification of status is required in the procedure for access to MA-3 registers. At this time, time out processing by using a timer is needed to avoid endless loop. Function that performs the status check is defined as follows.

```
SINT32 machdep_CheckStatus  ( UINT8 flag );
```

Description

Performs status check processing.

Argument

flag	Value of flag to be checked

Returned value

0	Successful
-1	Time out

3.2.3 Setting of Valid data wait processing function

In the procedure for acquiring the values of MA-3 registers, it is necessary to verify the status whether valid data has been prepared or not. At this time, time out processing that uses a time is needed to avoid endless loop. Function that performs the status check is defined as follows.

```
SINT32 machdep_WaitValidData  ( UINT8 flag );
```

Description

Performs status check processing.

Argument

flag	Value of flag to be checked

Returned value

0	Successful
-1	Time out

3.2.4 Setting of sleep processing function

Function for sleeping for a period designated by the stop processing is needed. The function that performs sleep is defined below.

```
void machdep_Sleep ( UINT32 sleep_time );
```

Description

Sleeps for a designated period.

Argument

sleep_time	Sleep time [ms]

Returned value

None

3.2.5 Setting of delay system FIFO Empty check processing function

This function checks if the delay system FIFO is Empty or not. No change is required in regular operation.

```
SINT32 machdep_CheckDelayedFifoEmpty ( void );
```

Description

Checks if the delay system FIFO is EMPTY.

Argument

None

Returned value

0	Not Empty
1	Empty

3.2.6 Setting of delay system FIFO Empty waiting processing function

This function waits until the delay system FIFO becomes Empty. Time out processing by using timer is needed to avoid endless loop.

```
SINT32 machdep_WaitDelayedFifoEmpty ( void );
```

Description

Waits until the delay system FIFO becomes EMPTY.

Argument

None

Returned value

0	Successful
-1	Time out

3.2.7 Setting of real time system FIFO Empty waiting processing function

This function waits until the real time system FIFO becomes Empty. Time out processing by using timer is needed to avoid endless loop.

SINT32 machdep_WaitImmediateFifoEmpty (void);

Description

Waits until the real time system FIFO becomes EMPTY.

Argument

None

Returned value

0	Successful
-1	Time out

3.3 macnvprf.h

This file describes the definition of registration of Stream Converter. Change this definition when changing the contents of converter registration.

3.4 Registration of MA-3 interrupt processing function

Set the function that processes MA-3 hardware interruption to MaDevDrv_IntHandler.

3.5 LED / Motor Control

Since the control source setup of LED and Motor is turned off by the default, please set up according to a required situation. Moreover, On/Off control of LED and Motor is not performed when there is no Synchronous setup in Sequence data even if it has been set as the sequence synchronization, or when it is not supported by Converter and correspondence sequence. Therefore, please set it suitably from API according to an application situation.

It is necessary to publish in power down release state of device at API setup, and keep in mind that API issue procedure has restriction at the control source setup.

Please refer to “MA-3 Sound Middleware API Specification” and “MA-3 Sound Middleware Release Note” about the setting method of control.

3.6 Others

When change due to compiler dependency or architecture is needed, do it properly.

3.7 Debug mode (madebug.h, madebug.c)

3.7.1 Output setting of MA-3 Hardware file Format

By changing the defined value "1" of MASMW_DEBUG in madebug.h, functions for debugging in madebug.c become valid and MA-3 Hardware file Format type data is outputted. These are functions for debugging, and thus, are not necessary for usual operation.

Before making MA Sound Sequencer operate, it is necessary to set the output buffer with madebug_Open (), and to perform closing processing with madebug_Close () after closing.

void madebug_Open (UINT8 * ptr, UINT32 size);

Description

Designates buffers that output MA-3 Hardware file Format type data as the preparation for the output.

Argument

ptr	Pointer for buffers that output MA-3 Hardware file Format type data.
size	Byte size of buffers that store MA-3 Hardware file Format type data.

Returned value

None

UINT32 madebug_Close (void);

Description

Closes output of MA-3 Hardware file Format type data.

Argument

None

Returned value

Number of output bytes

3.8 Cancellation processing of delayed recovery function

Since there is a delayed recovery function for recovering the delay from absolute time in the sequencer of MA-3, timing lag generally becomes small. Therefore, this mechanism would be useful. However, if data supply stops at a certain reason, it will operate the situation in the direction to worsen.

As cause by which data supply stops, we guess the case where original data serve as excessive message density or the case where it cannot move to processing of sound middleware immediately, etc. In the delayed recovery function of MA-3, when there is delay to an absolute time, it is going to recover the delayed part by subsequent message processing. Therefore, the sequence data on FIFO will be consumed extremely for a short time. This means Empty interruption occurs frequently without vacating time.

For this reason, from MA-3 sound middleware ver.1.2.3, when the sequence data FIFO becomes Empty and sequencer suspends processing and the delayed recovery function becomes effective, the reset processing of sequencer control was added in order to cancel it. This processing was added within the MaDevDrv_Fifo function.

When this portion is changed at the implementation, we would like you to change so that this processing may function.

3.9 Measures for reduction of processing load at implementation

To reduce the processing load at the implementation, the following measures are recommended.

3.9.1 Making functions in-line

It is recommended to make the following functions in line.

File name	Function name
mamachdep.c	machdep_memcpy
	machdep_wait
	machdep_WriteStatusFlagReg
	machdep_ReadStatusFlagReg
	machdep_WriteDataReg
	machdep_ReadDataReg
	machdep_WaitValidData
	machdep_ChecKDelayedFifoEmpty
	machdep_WaitDelayedFifoEmpty
	machdep_WaitImmediateFifoEmpty
mammfcnv.c	Decode_GetBit
	Ev_Set
	Get_EvList
	Get_OffList
	Set_EvList
	Set_OffList

3.9.2 Integrating functions into RAM

For the systems with high speed built-in RAM, it is recommended to integrate the following functions into the RAM.

File name	Priority order	Function name
madevdrv.c	1	MaDevDrv_SendDelayedPacket
		MaDevDrv_SendDirectPacket
		MaDevDrv_SendDirectRamData
		MaDevDrv_SendDirectRamVal
masnddrv.c	2	SearchSlotList
		SendDelayedPacket
		RemakeSlotList
		GetVoiceInfo
		SearchSlot
		GetSlotList
		GetSlot
		MakeDeltaTime
mammfcnv.c	3	Decode_byte
	4	Decode_byte7bit
	5	Search_OffList2
	6	Search_OffList3
	7	Search_OffListA
	8	Note_ON3
	9	Note_ON2
	10	get_flex3
	11	Set_NoteOffEvent
	12	Set_Event