YAMAHA

CONFIDENTIAL

DSE Mode Control Microprocessor Sample Program Manual

version 1.00.00

Sep. 3, 2012

Yamaha Corporation
Semiconductor Division

Contents

1 Introduction

1.1 Overview

This is a sample program for controlling YDA174/YSS952 (DD-2SP/SPR-2) with it embedded in a microprocessor. In order to run DSE mode on DD-2SP/SPR-2, this program executes the following: initialization, control, register access and on-chip RAM access. In addition, this program allows preset pattern on on-chip ROM in DD-2SP/SPR-2 to be set.

Refer to the following documents as well as this manual.

- "YDA174 Application Manual" or "YSS952 Application Manual"
- "DD-2SP/SPR-2 Firmware DSE Mode Manual"

[Notes]

- For details of processing in this program, see description in the program.
- Hereafter, microprocessor may be referred to as "MCU".

1.2 File Configuration

This program is composed of the following files.

MCUProgram
└─ Sample

common.h	: Common Header File
dsp_access.c	: DD-2SP/SPR-2 Access Processing
dsp_access.h	: DD-2SP/SPR-2 Access Processing Header File
dsp_coef_data.h	: MDSP2/SDSP Coefficient Data File
dsp_control.c	: DD-2SP/SPR-2 Control Processing
dsp_control.h	: DD-2SP/SPR-2 Control Processing Header File
dsp_inst_data.h	: MDSP2 Instruction Code File
init.c	: Initialization Processing
init.h	: Initialization Processing Header File
ioctrl.c	: MCU I/O Block Control Processing
ioctrl.h	: MCU I/O Block Control Processing Header File
main.c	: Main Processing
main.h	: Main Processing Header File

2 Operating Environment

This program is designed with the following operating environment.

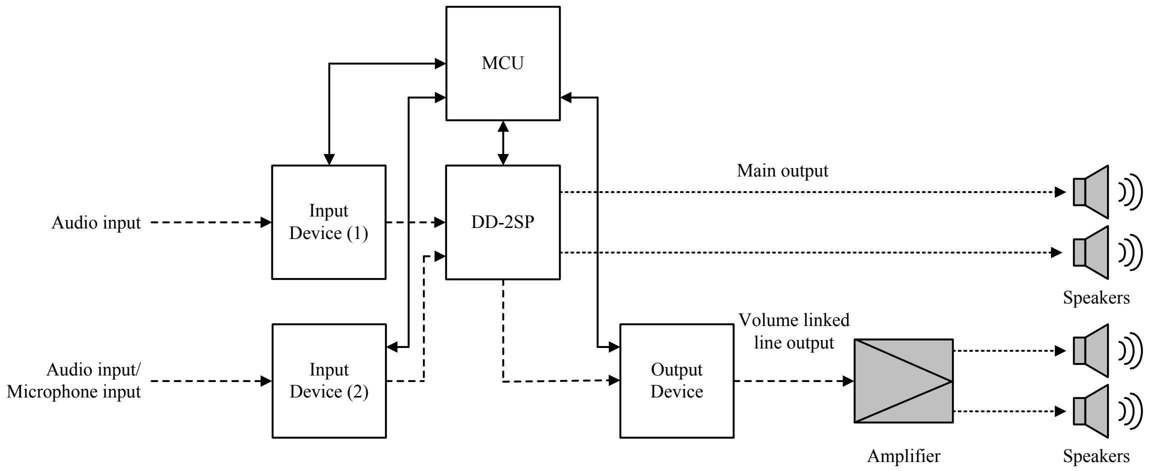
2.1 Microprocessor

This program is designed with being embedded into MCU with I^{2}C communication function.

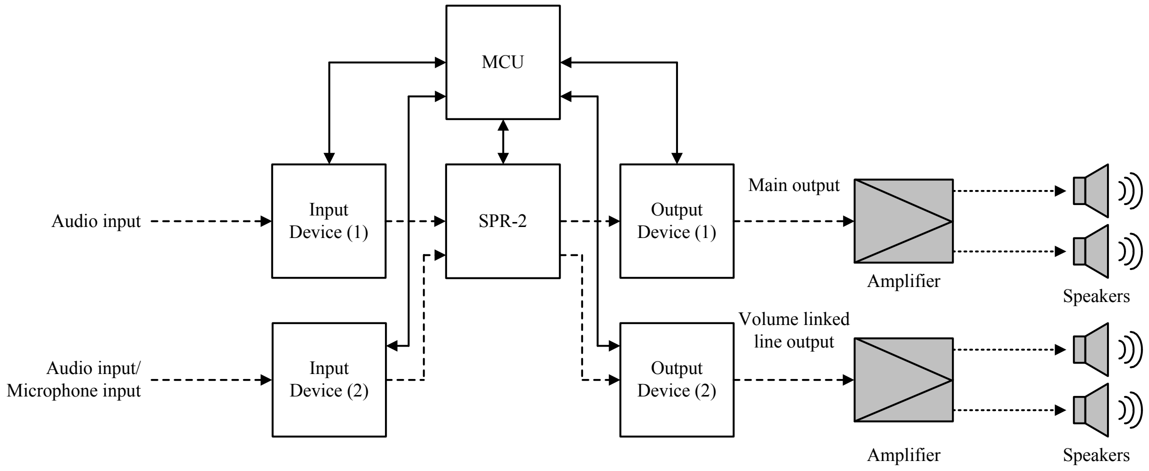
2.2 System Configuration

The expected system configurations is shown as examples below.

- **System with DD-2SP**



- **System with SPR-2**



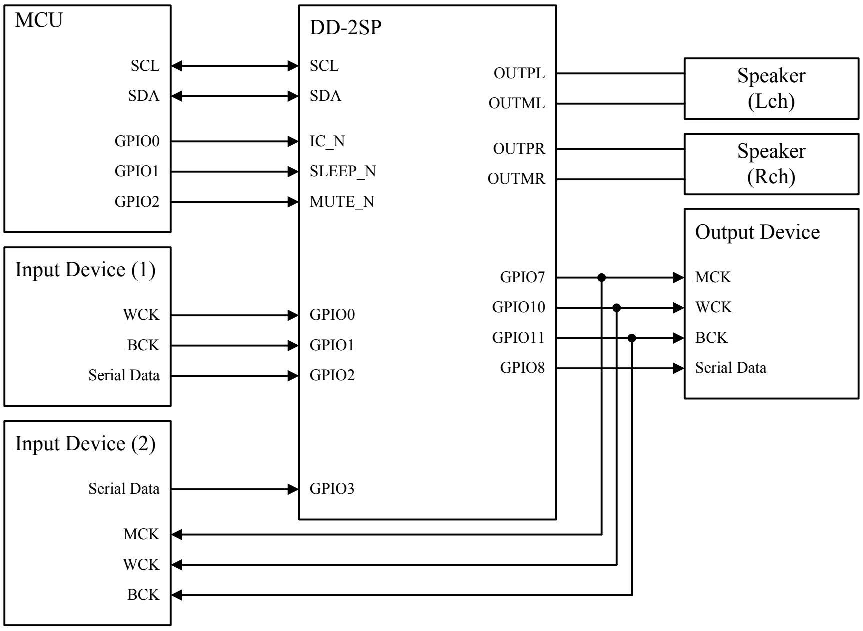
[Note]

- The solid line indicates the control signal path and the dotted line indicates the audio signal path.

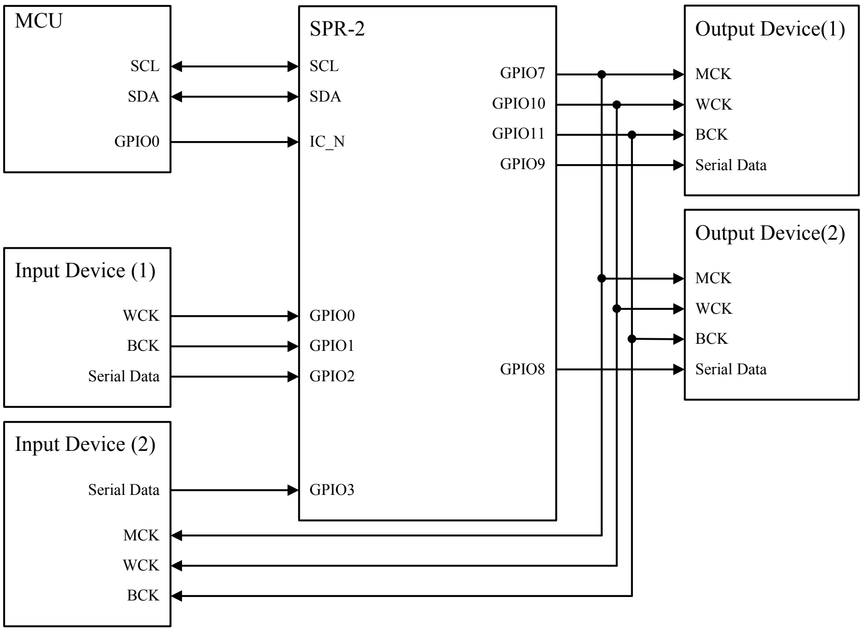
2.3 Connection between Devices

This program assumes DD-2SP/SPR-2 operates in AIF mode (=0) and devices are connected as shown below:

- **DD-2SP**



- **SPR-2**



3 Main Function

3.1 Overview

The codes in this section provide sample of main function that controls DD-2SP/SPR-2.

- **main**

(Source File)

main.c

(Calling Format)

void main(void);

(Description)

Controls DD-2SP/SPR-2 by using “4 Access Function”, “5 Initialization Function”, and “6 Control Function” according to “3.2 Processing Flow”.

(Argument)

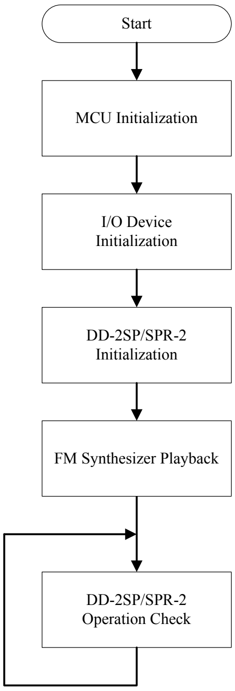
None

(Return Value)

None

3.2 Processing Flow

This section describes processing flow and its details of main function:


Start
MCU Initialization
I/O Device Initialization
DD-2SP/SPR-2 Initialization
FM Synthesizer Playback
DD-2SP/SPR-2 Operation Check

- **MCU Initialization**

MCU is initialized. Fill in setups to initialize your MCU.

- **Input and Output Device Initialization**

Input and output devices are initialized. Fill in setups to initialize your devices.

- **DD-2SP/SPR-2 Initialization**

DD-2SP/SPR-2 is initialized according to procedure described in “YDA174 Application Manual” or “YSS952 Application Manual”. With this program, DD2SPSetDSEMode function (5 Initialization Function) is used for this initialization. Change the function to either one of functions described in “5 Initialization Function” as needed.

● FM Synthesizer Playback

FM synthesizer contents [32: Power_On_Short_01] are played back.

● DD-2SP/SPR-2 Operation Check

DD-2SP/SPR-2 operation is judged on status of the following three register bits:

- MDSPERR bit (0x1B MDSPST register)
- SDSPERR bit (0x20 SDSPST register)
- PLLERR bit (0x23 PLLERR register)

[Note]

- See “YDA174 Application Manual” or “YSS952 Application Manual” for detail of the above register bits.

4 Access Function

The codes in this section provide functions that implement accesses to DD-2SP/SPR-2 registers and DD-2SP/SPR-2 on-chip RAM. In addition, functions for setting preset patterns are also provided.

4.1 Register Access

To write data to DD-2SP/SPR-2 register, use DSPWriteReg function. To read data from the register, use DSPReadReg function. And, use DSPPartWriteReg function to set the register in bit by bit.

● DSPWriteReg

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPWriteReg(unsigned char bAdd, unsigned char bData);

(Description)

Write data to DD-2SP/SPR-2 register.

(Argument)

unsigned char	bAdd:	Write address. Set an address ranging from 0x00 to 0x3F.
unsigned char	bData:	Write data. Set a data value ranging from 0x00 to 0xFF.

(Return Value)

If writing to the register is successful, “0” will be returned.

If writing to the register fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Incorrect write address (bAdd)

● DSPReadReg

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPReadReg(unsigned char bAdd, unsigned char *pbData);

(Description)

Read data from DD-2SP/SPR-2 register.

(Argument)

unsigned char	bAdd:	Read address. Set an address ranging from 0x00 to 0x3F.
unsigned char	*pbData:	A pointer to the variable in which read data is stored.

(Return Value)

If reading from the register is successful, “0” will be returned.

If reading from the register fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Incorrect read address (bAdd)

● DSPPartWriteReg

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPPartWriteReg(unsigned char bAdd, unsigned char bData, unsigned char bMask);

(Description)

Set DD-2SP/SPR-2 register in bit by bit.

(Argument)

unsigned char	bAdd:	Write address. Set an address ranging from 0x00 to 0x3F.
unsigned char	bData:	Write data. Set a data value ranging from 0x00 to 0xFF.
unsigned char	bMask:	Mask. Set a value ranging from 0x00 to 0xFF. Only bits (out of the write data) set to 1 in this argument are write to the register.

(Return Value)

If writing to the register is successful, “0” will be returned.

If writing to the register fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Incorrect write address (bAdd)

4.2 On-chip RAM Access

Use DSPWriteMem function to write data directly into DD-2SP/SPR-2 on-chip RAM. Use DSPReadMem function to read data directly from the RAM,.

In order to write data into CRAM (MDSP2 coefficient RAM) by runtime transfer, use MDSPCoefRTDL function. On the other hand, to read data from CRAM by runtime transfer, execute MDSPCoefRTUL function.

And, use SDSPCoefBT function to write data into PEQ Coefficient RAM via PEQ transfer buffer.

The word sizes on DD-2SP/SPR-2 on-chip RAM are specified in this program as shown below:

No.	**DD-2SP/SPR-2 On-Chip RAM**			
	Area Name	**Initial Address**	**Size [word]**	**Word Length [bit]**
0	WRAM0	0x0000	0x20	16
1	WRAM1	0x0020	0x3E0	32
2	WRAM2 + Reserved	0x0400	0x3C00	32
3	CRAM	0x4000	0x400	16
4	MPRAM	0x4400	0x600	40
5	Reserved	0x4A00	0x3600	32
6	PEQ Coefficient RAM + Reserved	0x8000	0x4000	24
7	Other than the above	0xC000	0x4000	16

The following functions access an area defined above at a time.

● DSPWriteMem

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPWriteMem(unsigned short wAdd, unsigned char *pbData, unsigned short wWordSize);

(Description)

Write data directly into DD-2SP/SPR-2 on-chip RAM.

(Argument)

unsigned short	wAdd:	Top address of write area. Set an address ranging from 0x0000 to 0xFFFF.
unsigned char	*pbData:	A pointer to the buffer in which to store write data.
unsigned short	wWordSize:	Write data size (Word unit). Set a value ranging from 0x0001 to 0x07FF.

(Return Value)

If writing to the memory is successful, “0” will be returned.

If writing to the memory fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Top address of write area + write data size (wAdd + wWordSize) is outside the scope of the area.

● DSPReadMem

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPReadMem (unsigned short wAdd, unsigned char *pbData, unsigned short wWordSize);

(Description)

Read data directly from DD-2SP/SPR-2 on-chip RAM.

(Argument)

unsigned short	wAdd:	Top address of read area. Set an address ranging from 0x0000 to 0xFFFF.
unsigned char	*pbData:	A pointer to the buffer in which read data is stored.
unsigned short	wWordSize:	Read data size (Word unit). Set a value ranging from 0x0001 to 0x07FF.

(Return Value)

If reading to the memory is successful, “0” will be returned.

If reading to the memory fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Top address of read area + read data size (wAdd + wWordSize) is outside the scope of the area.

● MDSPCoefRTDL

(Source File)

dsp_access.c

(Calling Format)

unsigned char MDSPCoefRTDL(unsigned short wAdd, unsigned char *pbData, unsigned short wWordSize);

(Description)

Write data into CRAM by using runtime transfer mode.

(Argument)

unsigned short	wAdd:	Top address of write area. Set an address ranging from 0x4000 to 0x43FF.
unsigned char	*pbData:	A pointer to the buffer in which to store write data.
unsigned short	wWordSize:	Write data size (Word Unit). Set an address ranging from 0x0001 to 0x000A.

(Return Value)

If writing to CRAM is successful, “0” will be returned.

If writing to CRAM fails, any one of the following three values will be returned:

1: Occurrence of I^{2}C communication error

3: Occurrence of error in DD-2SP/SPR-2 runtime transfer processing (“0x13 MDSPREQ” value is except 0x00)

5: Top address of write area + write data size (wAdd + wWordSize) is outside the scope of the area.

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

● MDSPCoefRTUL

(Source File)

dsp_access.c

(Calling Format)

unsigned char MDSPCoefRTUL (unsigned short wAdd, unsigned char *pbData, unsigned short wWordSize);

(Description)

Read data from CRAM by using runtime transfer mode.

(Argument)

unsigned short	wAdd:	Top address of read area. Set an address ranging from 0x4000 to 0x43FF.
unsigned char	*pbData:	A pointer to the buffer in which read data is stored.
unsigned short	wWordSize:	Read data size (Word Unit). Set a value ranging from 0x0001 to 0x000A.

(Return Value)

If reading from CRAM is successful, “0” will be returned.

If reading from CRAM fails, any one of the following three values will be returned:

1: Occurrence of I^{2}C communication error

3: Occurrence of error in DD-2SP/SPR-2 runtime transfer processing (“0x13 MDSPREQ” value is except 0x00)

5: Top address of read area + read data size (wAdd + wWordSize) is outside the scope of the area.

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

● SDSPCoefBT

(Source File)

dsp_access.c

(Calling Format)

unsigned char SDSPCoefBT(unsigned short wAdd, unsigned char *pbData, unsigned short wWordSize);

(Description)

Write PEQ Coefficients for 1 band into PEQ Coefficient RAM in a batch.

By setting data to PEQ Coefficient Transfer Buffer Area and Transfer Parameter/Transfer Execution Area, the above write processing is realized.

(Argument)

unsigned short	wAdd:	Top address of write area. Set an address ranging from 0x8000 to 0x8045.
unsigned char	*pbData:	A pointer to the buffer in which to store write data.
unsigned short	wWordSize:	Write data size (Word Unit). Set a value of 0x5 or 0xA.

(Return Value)

If writing to PEQ Coefficient RAM is successful, “0” will be returned.

If writing to PEQ Coefficient RAM fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Top address of write area + write data size (wAdd + wWordSize) is outside the scope of the area.

(Note)

Execute this function while SDSP is operating (SDSPSTART=1 in 0x11 DSPSTART register).

4.3 Preset Pattern Setting

Use DSPLoadPreset function to set preset pattern in DD-2SP/SPR-2 on-chip ROM. And, use DSPStopPreset function to stop the preset pattern setting.

● DSPLoadPreset

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPLoadPreset (unsigned short wNum, unsigned short wAdd, bool bSetAdd);

(Description)

Set preset pattern.

(Argument)

unsigned short	wNum:	On-chip ROM block No. Set a value ranging from 0x000 to 0x3FF.
unsigned short	wAdd:	Specify address to write preset coefficient (0x4000 to 0x43FF) when bSetAdd=TRUE. Specify 0 when bSetAdd=FALSE.
bool	bSetAdd:	TRUE: Write address (wAdd) specification enabled. FALSE: Write address (wAdd) specification disabled.

(Return Value)

If setting the preset pattern is successful, “0” will be returned.

If setting the preset pattern fails, either of the following two values will be returned:

1: Occurrence of I^{2}C communication error

5: Illegal on-chip ROM block No. (wNum)
Illegal write address (wAdd) with write address specification enabled (bSetAdd=TRUE)

(Notes)

- Specify write address when setting a preset pattern of on-chip ROM block No. 0x0E3 through 0x0F3.
- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- If executing this function while MDSP2 is stopped, DD-2SP/SPR-2 cannot be accessed form then on. In this case, DSPStopPreset function allows access to it again.

● DSPStopPreset

(Source File)

dsp_access.c

(Calling Format)

unsigned char DSPStopPreset(void);

(Description)

Stops preset pattern setting.

(Argument)

None

(Return Value)

If this preset pattern setting stop function is successfully performed, “0” will be returned.

If this preset pattern setting stop function fails because of I^{2}C communication error, “1” will be returned.

5 Initialization Function

The codes in this section provides functions to initialize DD-2SP/SPR-2.

5.1 DD-2SP Initialization

DD-2SP can be initialized by executing DD2SPSetDSEMode function after resetting DD-2SP.

● DD2SPSetDSEMode

(Source File)

init.c

(Calling Format)

void DD2SPSetDSEMode(void);

(Description)

Initializes DD-2SP DSE mode is set as firmware mode and its related functions are set as shown below:

- I/O data format: I^2S format.
- AIF mode: 0.
- Secures 8-bit headroom.
- FM synthesizer output is mixed.
- The following functions or effects are disabled: small signal detection, surround, band extension, acoustic compensation, loudness, output power limiter, soft clipping.
- The network configuration is the 2.1 channel configuration.
- Master volume is set to -30.0dB. Master volume change smoothly.
- MDSP2 and SDSP will run and mute will be released.

(Argument)

None

(Return Value)

None

(Note)

To change the digital network to 1 way of 2 way, execute SetCrossover function. See“6.1.3 Setting Crossover Network”, for the detail of the above function.

5.2 SPR-2 Initialization

SPR-2 can be initialized by executing SPR2SetDSEMode function after resetting SPR-2.

● SPR2SetDSEMode

(Source File)

init.c

(Calling Format)

void SPR2SetDSEMode(void);

(Description)

Initializes SPR-2 DSE mode is set as firmware mode and its related functions are set as shown below:

- I/O data format: I^{2}S format.
- AIF mode: 0.
- Secures 8-bit headroom.
- FM synthesizer output is mixed.
- The following functions or effects are disabled: small signal detection, surround, band extension, acoustic compensation, loudness, output power limiter, soft clipping.
- The network configuration is the 2.1 channel configuration.
- Master volume is set to -12.0dB. Master volume change smoothly.
- MDSP2 and SDSP will run and mute will be released.

(Argument)

None

(Return Value)

None

(Note)

To change the digital network to 1 way of 2 way, execute SetCrossover function. See “6.1.3 Setting Crossover Network”, for the detail of the above function.

6 Control Function

6.1 Firmware

6.1.1 Volume Control

Use SetVolume function to set DD-2SP/SPR-2's master volume.
Use Mute function to mute amplifier pin output and digital output. And, use MuteRelease function to release mute of amplifier pin output and digital output.

- **Mute**

(Source File)
dsp_control.c
(Calling Format)
void Mute(void);
(Description)
Mute amplifier pin output and digital output.
(Argument)
None
(Return Value)
None
(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- When executing this function, master volume is muted smoothly. This program sleeps until master volume is muted. The sleep time is 20 ms.
- The time taken from 0dB to mute state of master volume in [692: SV Curve Mid] or its equivalent setting shall be the above sleep time.

- **MuteRelease**

(Source File)
dsp_control.c
(Calling Format)
void MuteRelease(void);
(Description)
Release mute of amplifier pin output and digital output.
(Argument)
None
(Return Value)
None
(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- When executing this function, master volume is released mute smoothly.

● SetVolume

(Source File)

dsp_control.c

(Calling Format)

void SetVolume(unsigned char bVolID);

(Description)

Set master volume by using preset patterns.

(Argument)

unsigned char bVolID: Volume ID code ranging from 0x01 to 0x85.

bVolID	Volume Value [dB]
0x01	-128
0x02	-126
0x03	-124
0x04	-122
0x05	-120
0x06	-118
0x07	-116
0x08	-114
0x09	-112
0x0A	-110
0x0B	-108
0x0C	-106
0x0D	-104
0x0E	-102
0x0F	-100
0x10	-99
0x11	-98
0x12	-97
0x13	-96
0x14	-95
0x15	-94
0x16	-93
0x17	-92
0x18	-91
0x19	-90
0x1A	-89
0x1B	-88
0x1C	-87
0x1D	-86
0x1E	-85
0x1F	-84
0x20	-83
0x21	-82
0x22	-81
0x23	-80
0x24	-79
0x25	-78
0x26	-77
0x27	-76
0x28	-75
0x29	-74
0x2A	-73
0x2B	-72
0x2C	-71
0x2D	-70
0x2E	-69
0x2F	-68
0x30	-67
0x31	-66
0x32	-65
0x33	-64
0x34	-63
0x35	-62
0x36	-61
0x37	-60
0x38	-59
0x39	-58
0x3A	-57
0x3B	-56
0x3C	-55
0x3D	-54
0x3E	-53
0x3F	-52
0x40	-51
0x41	-50
0x42	-49
0x43	-48
0x44	-47
0x45	-46
0x46	-45
0x47	-44
0x48	-43
0x49	-42
0x4A	-41
0x4B	-40
0x4C	-39
0x4D	-38
0x4E	-37
0x4F	-36
0x50	-35
0x51	-34
0x52	-33
0x53	-32
0x54	-31
0x55	-30
0x56	-29
0x57	-28
0x58	-27
0x59	-26
0x5A	-25
0x5B	-24
0x5C	-23
0x5D	-22
0x5E	-21
0x5F	-20
0x60	-19
0x61	-18
0x62	-17
0x63	-16
0x64	-15
0x65	-14
0x66	-13
0x67	-12
0x68	-11
0x69	-10
0x6A	-9
0x6B	-8
0x6C	-7
0x6D	-6
0x6E	-5
0x6F	-4
0x70	-3
0x71	-2
0x72	-1
0x73	0
0x74	1
0x75	2
0x76	3
0x77	4
0x78	5
0x79	6
0x7A	7
0x7B	8
0x7C	9
0x7D	10
0x7E	11
0x7F	12
0x80	13
0x81	14
0x82	15
0x83	16
0x84	17
0x85	18

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.2 Balance Control

Use SetBalance function to adjust balance between left and right channels of output (high range output).

● SetBalance

(Source File)

dsp_control.c

(Calling Format)

void SetBalance(signed char bBalID);

(Description)

Adjust balance between left and right channels of output (high range output).

(Argument)

signed char bBalID: Balance ID code ranging from −33 to +33

bBalID	L/R Output Balance
33	To the right (Left − ∞)
32	To the right (Left −32 dB)
31	To the right (Left −31 dB)
30	To the right (Left −30 dB)
29	To the right (Left −29 dB)
28	To the right (Left −28 dB)
27	To the right (Left −27 dB)
26	To the right (Left −26 dB)
25	To the right (Left −25 dB)
24	To the right (Left −24 dB)
23	To the right (Left −23 dB)
22	To the right (Left −22 dB)
21	To the right (Left −21 dB)
20	To the right (Left −20 dB)
19	To the right (Left −19 dB)
18	To the right (Left −18 dB)
17	To the right (Left −17 dB)
16	To the right (Left −16 dB)
15	To the right (Left −15 dB)
14	To the right (Left −14 dB)
13	To the right (Left −13 dB)
12	To the right (Left −12 dB)
11	To the right (Left −11 dB)
10	To the right (Left −10 dB)
9	To the right (Left −9 dB)
8	To the right (Left −8 dB)
7	To the right (Left −7 dB)
6	To the right (Left −6 dB)
5	To the right (Left −5 dB)
4	To the right (Left −4 dB)
3	To the right (Left −3 dB)
2	To the right (Left −2 dB)
1	To the right (Left −1 dB)
0	Center

bBalID	L/R Output Balance
−1	To the left (Right −1 dB)
−2	To the left (Right −2 dB)
−3	To the left (Right −3 dB)
−4	To the left (Right −4 dB)
−5	To the left (Right −5 dB)
−6	To the left (Right −6 dB)
−7	To the left (Right −7 dB)
−8	To the left (Right −8 dB)
−9	To the left (Right −9 dB)
−10	To the left (Right −10 dB)
−11	To the left (Right −11 dB)
−12	To the left (Right −12 dB)
−13	To the left (Right −13 dB)
−14	To the left (Right −14 dB)
−15	To the left (Right −15 dB)
−16	To the left (Right −16 dB)
−17	To the left (Right −17 dB)
−18	To the left (Right −18 dB)
−19	To the left (Right −19 dB)
−20	To the left (Right −20 dB)
−21	To the left (Right −21 dB)
−22	To the left (Right −22 dB)
−23	To the left (Right −23 dB)
−24	To the left (Right −24 dB)
−25	To the left (Right −25 dB)
−26	To the left (Right −26 dB)
−27	To the left (Right −27 dB)
−28	To the left (Right −28 dB)
−29	To the left (Right −29 dB)
−30	To the left (Right −30 dB)
−31	To the left (Right −31 dB)
−32	To the left (Right −32 dB)
−33	To the left (Right −∞)

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.3 Setting Crossover Network

Use SetCrossover function to set the digital crossover network path.

- **SetCrossover**

(Source File)

dsp_control.c

(Calling Format)

void SetCrossover(unsigned char bNetworkID, unsigned char *pbData);

(Description)

Sets any one of the following digital network paths: 1 way, 2 way, or 2.1 channel.

Specific coefficients are used as crossover filter coefficients.

(Argument)

unsigned char	bNetworkID:	0x0:	1 way
		0x1:	2 way
		0x2:	2.1 channel
unsigned char	*pbData:	A pointer to an array. Store coefficient data of HPF.a0[27:12] to LPF.d2[11:0] coefficients (at the addresses from 0x41C3 to 0x41EA) to this array.	

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.4 Surround

Use SetS3D function to apply surround effect.
Use S3DOn function to enable or disable surround effect.

- **SetS3D**

(Source File)
dsp_control.c
(Calling Format)
void SetS3D(unsigned char *pbData);
(Description)
Apply surround effect.
(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of S3DCore.Data0[15:0] to S3DLev[15:0] coefficients (at the addresses from 0x409D to 0x4135) to this array.

(Return Value)
None
(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable surround effect, S3DOn(TRUE) will be executed in this function (For S3DOn(TRUE), see below).

- **S3DOn**

(Source File)
dsp_control.c
(Calling Format)
void S3DOn(bool bOn);
(Description)
Enable or disable surround effect.
(Argument)

bool	bOn:	TRUE:	Enable surround effect.
		FALSE:	Disabl surround effect.

(Return Value)
None
(Note)
Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.5 Band Extension

Use SetHXT function to apply band extension effect.
Use HXTOn function to enable or disable band extension effect.

- **SetHXT**

(Source File)
dsp_control.c
(Calling Format)
void SetHXT(unsigned char *pbData);
(Description)
Apply band extension effect.
(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of HXTCore.Data0[15:0] to WetMix[15:0] coefficients (at the addresses from 0x4149 to 0x4195) to this array.

(Return Value)
None
(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable harmonics enhancer effect, HXTOn(TRUE) will be executed in this function (For HXTOn(TRUE), see below).

- **HXTOn**

(Source File)
dsp_control.c
(Calling Format)
void HXTOn(bool bOn);
(Description)
Enable or disable band extension effect.
(Argument)

bool	bOn:	TRUE:	Enable band extension effect.
		FALSE:	Disabl band extension effect.

(Return Value)
None
(Note)
Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.6 Acoustic Compensation

Use SetAEQ function to apply acoustic compensation effect.
Use AEQOn function to enable or disable acoustic compensation effect.

● SetAEQ

(Source File)

dsp_control.c

(Calling Format)

void SetAEQ(unsigned char *pbData0, unsigned char *pbData1);

(Description)

Apply acoustic compensation effect.

(Argument)

unsigned char	*pbData0:	A pointer to an array. Store coefficient data of AEQCore.Data0[15:0] to RPostLev [15:0] coefficients (at the addresses from 0x41F3 to 0x43F6) to this array.
unsigned char	*pbData1:	A pointer to an array. Store coefficient data of SDSP 10-band PEQ coefficients (at the addresses from 0x8000 to 0x8045) to this array.

(Return Value)

None

(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable acoustic compensation effect, AEQOn(TRUE) will be executed in this function (For AEQOn(TRUE), see below).

● AEQOn

(Source File)

dsp_control.c

(Calling Format)

void AEQOn(bool bOn);

(Description)

Enables or disables acoustic compensation effect.

(Argument)

bool	bOn:	TRUE:	Enable acoustic compensation effect.
		FALSE:	Disabl acoustic compensation effect.

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.7 Loudness

Use PresetLoudnessSetting function to apply loudness effect by using preset pattern. Use UserLoudnessSetting function to apply specific loudness effect.
Use LoudnessOn function to enable or disable loudness effect.

- **PresetLoudnessSetting**

(Source File)
 dsp_control.c
(Calling Format)
 void PresetLoudnessSetting(unsigned char bEffectID);
(Description)
 Apply loudness effect by using preset pattern.
(Argument)

unsigned char	bEffectID:	0x0:	Apply [694:MVOL Ln DEQ0 Low] effect.
		0x1:	Apply [695:MVOL Ln DEQ0 High] effect.
		0x2:	Apply [696:MVOL Ln DEQ1 Low] effect.
		0x3:	Apply [697:MVOL Ln DEQ1 High] effect.

(Return Value)
 None
(Note)
 Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

- **UserLoudnessSetting**

(Source File)
 dsp_control.c
(Calling Format)
 void UserLoudnessSetting(unsigned char *pbData);
(Description)
 Apply specific loudness effect.
(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of EQ0.a0[15:0] to LnShift[15:0] coefficients (at the addresses from 0x4044 to 0x404F) to this array.

(Return Value)
 None
(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable loudness effect, LoudnessOn(TRUE) will be executed in this function (For LoudnessOn(TRUE), see below).

● LoudnessOn

(Source File)

dsp_control.c

(Calling Format)

void LoudnessOn(bool bOn);

(Description)

Enables or disables loudness effect.

(Argument)

bool	bOn:	TRUE:	Enable loudness effect.
		FALSE:	Disable loudness effect.

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.8 Output Power Limit

Use SetPowerLimit function to limit output power.
Use PowerLimitOn function to enable or disable output power limit.

● SetPowerLimit

(Source File)

dsp_control.c

(Calling Format)

void SetPowerLimit(unsigned char *pbData);

(Description)

Limit output power.

(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of HGC.GainShift[15:0] to LATK.b1 [11:0] coefficients (at the addresses from 0x406A to 0x4080) to this array.

(Return Value)

None

(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable output power limit, PowerLimitOn(TRUE) will be executed in this function (For PowerLimitOn(TRUE), see below).

● PowerLimitOn

(Source File)

dsp_control.c

(Calling Format)

void PowerLimitOn(bool bOn);

(Description)

Enable or disable output power limit.

(Argument)

bool	bOn:	TRUE:	Enable output power limit.
		FALSE:	Disabl output power limit.

(Return Value)

None

(Note)

Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.9 Soft Clipping

Use SetSoftClip function to clip waveform of low freqency singnal smoothly.
Use SoftClipOn function to enable or disable soft clipping.

- **SetSoftClip**

(Source File)
dsp_control.c

(Calling Format)
void SetSoftClip(unsigned char *pbData);

(Description)
Clip waveform of low freqency singnal smoothly.

(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of PreLev[15:0] to CompLev[15:0] coefficients (at the addresses from 0x4089 to 0x408B) to this array.

(Return Value)
None

(Notes)

- Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).
- In order to enable soft clipping, SoftClipOn(TRUE) will be executed in this function (For SoftClipOn(TRUE), see below).

- **SoftClipOn**

(Source File)
dsp_control.c

(Calling Format)
void SoftClipOn(bool bOn);

(Description)
Enable or disable soft clipping.

(Argument)

bool	bOn:	TRUE:	Enable soft clipping.
		FALSE:	Disabl soft clipping.

(Return Value)
None

(Note)
Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.1.10 Detection of Small Signal

Use SmallSignalDetectionOn function to enable detection of small signal.
Use SmallSignalDetectionOff function to disable detection of small signal.

● SmallSignalDetectionOn

(Source File)
dsp_control.c

(Calling Format)
void SmallSignalDetectionOn(unsigned char *pbData);

(Description)
Enable detection of small signal.

(Argument)

unsigned char	*pbData:	A pointer to an array. Store coefficient data of DetLev[15:0] to DetRst[15:0] coefficients (at the addresses from 0x400E to 0x4010) to this array.

(Return Value)
None

(Note)
Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

● SmallSignalDetectionOff

(Source File)
dsp_control.c

(Calling Format)
void SmallSignalDetectionOff(void);

(Description)
Disable detection of small signal.

(Argument)
None

(Return Value)
None

(Note)
Execute this function while MDSP2 is operating (MDSPSTART=1 in 0x11 DSPSTART register).

6.2 FM Synthesizer

6.2.1 Playing Back FM Synthesizer Contents

Use PlayFM function to play back FM synthesizer contents on DD-2SP/SPR-2 on-chip ROM.

- **PlayFM**

(Source File)

dsp_control.c

(Calling Format)

void PlayFM(unsigned char bContentID);

(Description)

Plays back FM synthesizer contents.

(Return Value)

unsigned char bContentID: Contents ID code ranging from 0x00 to 0x39

bContentID	Preset Pattern Used
0x00	[32:Power_On_01]
0x01	[33:Power_Off_01]
0x02	[34:Power_On_02]
0x03	[35:Power_On_03]
0x04	[36:Power_On_04]
0x05	[37:Power_On_05]
0x06	[38:Power_On_06]
0x07	[39:Power_On_07]
0x08	[40:Power_On_08]
0x09	[41:Power_On_09]
0x0A	[42:Power_On_10]
0x0B	[43:Power_On_11]
0x0C	[44:Power_On_12]
0x0D	[45:Power_On_13]
0x0E	[46:Power_On_14]
0x0F	[47:Power_On_15]
0x10	[48:Power_On_16]
0x11	[49:Power_On_17]
0x12	[50:Power_On_18]
0x13	[51:Power_On_19]
0x14	[52:Power_On_20]
0x15	[53:Power_On_21]
0x16	[54:Power_On_22]
0x17	[55:Power_On_23]
0x18	[56:Power_On_24]
0x19	[57:Power_On_25]
0x1A	[58:Power_On_26]
0x1B	[59:Power_On_27]
0x1C	[60:Power_On_28]
0x1D	[61:Power_On_29]
0x1E	[62:Power_On_30]
0x1F	[63:Power_On_31]
0x20	[64:Power_On_32]
0x21	[65:Power_On_33]
0x22	[66:Power_On_34]
0x23	[67:Power_On_35]
0x24	[68:Power_On_36]
0x25	[69:Power_On_37]
0x26	[70:Power_On_38]
0x27	[71:Power_On_39]
0x28	[72:Operation_01]
0x29	[73:Operation_02]
0x2A	[74:Operation_03]
0x2B	[75:Operation_04]
0x2C	[76:Operation_05]
0x2D	[77:Comical_01]
0x2E	[78:Pafopafo_01]
0x2F	[79:Chime_01]
0x30	[80:Bicycle_01]
0x31	[81:Telephone_01]
0x32	[82:Announcement_01]
0x33	[83:Announcement_02]
0x34	[84:School_Bel_01]
0x35	[85:Spring_01]
0x36	[86:SL_01]
0x37	[87:Cat_01]
0x38	[88:Cuckoo_01]
0x39	[89:Chime_02]

(Return Value)

None

6.3 Operation Check

6.3.1 Supervising DD-2SP/SPR-2 Operation

Use CheckError function to supervise DD-2SP/SPR-2 operation by monitoring MDSPERR, SDSPERR and PLLER register bits.

- **CheckError**

(Source File)

dsp_access.c

(Calling Format)

unsigned char CheckError(void)

(Description)

Supervise DD-2SP/SPR-2 operation by monitor the following three register bits:

- MDSPERR bit (0x1B MDSPST register)
- SDSPERR bit (0x20 SDSPST register)
- PLLERR bit (0x23 PLLERR register)

(Return Value)

None

(Return Value)

If DD-2SP/SPR-2 is operating normally, “0” will be returned.

If DD-2SP/SPR-2 operation is unusual, “4” will be returned.

If reading the above registers fails, “1” will be returned.

(Note)

For detail of MDSPERR bit, SDSPERR bit and PLLERR bit, see “YDA174 Application Manual” or “YSS952 Application Manual”.

Appendix A Function List

The function of this program are listed in the table below.

Function Name	Source File	Description
DSPWriteReg	dsp_access.c	Write data to DD-2SP/SPR-2 register.
DSPReadReg		Read data from DD-2SP/SPR-2 register..
DSPPartWriteReg		Set DD-2SP/SPR-2 register in bit by bit.
DSPWriteMem		Write data directly into DD-2SP/SPR-2 on-chip RAM.
DSPReadMem		Read data directly from DD-2SP/SPR-2 on-chip RAM.
MDSPCoefRTDL		Write data into CRAM by using runtime transfer mode.
MDSPCoefRTUL		Read data from CRAM by using runtime transfer mode.
SDSPCoefBT		Write PEQ Coefficients for 1 band into PEQ Coefficient RAM in a batch.
DSPLoadPreset		Set preset pattern.
DSPStopPreset		Stops preset pattern setting.
CheckError		Supervise DD-2SP/SPR-2 operation.
DD2SPSetDSEMode	init.c	Initializes DD-2SP DSE mode is set as firmware mode.
SPR2SetDSEMode		Initializes SPR-2 DSE mode is set as firmware mode.
Mute	dsp_control.c	Mute master volume.
MuteRelease		Release mute of master volume.
SetVolume		Set master volume.
SetBalance		Adjust balance between left and right channels.
SetCrossover		Set digital network paths.
SetS3D		Apply surround effect.
S3DOn		Enables or disables surround effect.
SetHXT		Apply band extension effect.
HXTOn		Enables or disables band extension effect.
SetAEQ		Apply acoustic compensation effect.
AEQOn		Enables or disables acoustic compensation effect.
PresetLoudnessSetting		Apply loudness effect by using preset pattern.
UserLoudnessSetting		Apply specific loudness effect.
LoudnessOn		Enables or disables loudness effect.
SetPowerLimit		Limit output power.
PowerLimitOn		Enable or disable output power limit.
SetSoftClip		Clip waveform of low freqency singnal smoothly.
SoftClipOn		Enable or disable soft clipping.
SmallSignalDetectionOn		Enable detection of small signal.
SmallSignalDetectionOff		Disable detection of small signal.
PlayFM		Plays back FM synthesizer contents.