YAMAHA

CONFIDENTIAL

Format Conversion Tool
dd-1fixcnv
Manual

Version 1.00.05

April 4, 2011

Yamaha Corporation
Semiconductor Division

List of Contents

1 Introduction 3

2 Install and Uninstall Methods 3

3 Usage 4

3.1 Command-line Syntax 4

3.2 Input Data Format 5

3.3 Input File Format 6

3.4 Execution 7

4 Error and Warning Message 10

1 Introduction

The dd-1fixcnv is a command-line tool that converts 2’s complement fixed-point format into real-number format and its opposite direction mutually. This tool is used to prepare coefficient data of the YDA155/YDA156（DD-1/DD-1SP） or YDA174（DD-2SP） or YSS951(SPR-1) or YSS954(SPR-2) firmware in 2’s complement fixed-point format. For details of 2’s complement fixed-point format, see the “YDA155/YDA156 Application Manual” or “YDA174 Application Manual” or “YSS951 Application Manual” or “YSS952 Application Manual”.

- Converts real-number format into 2’s complement fixed-point format. On the contrary, 2’s complement fixed-point format is converted into real-number format.
- Two kinds of conversion modes:
  File Conversion Mode : handles input and output data as a text file.
  Direct Conversion Mode: specifies an input data to an option directly and displays an output data on the window.
- Recommended Operating Environment:
  Windows XP SP3 32bit Japanese, English Versions
  Windows 7 32bit/64bit Japanese, English Versions

2 Install and Uninstall Methods

1) Install Method

Copy the “dd-1fixcnv.exe” onto an appropriate folder.
Registry update, file association, etc. are not performed.

2) Uninstall Method

Delete the “dd-1fixcnv.exe” file.

3 Usage

3.1 Command-line Syntax

● File Conversion Mode

```
prompt>dd-1fixcnv -i<file name> -o<file name> [-r<type>|-x<type>] [-s<scale>]  [-c] [-h]
```

● Direct Conversion Mode

```
prompt>dd-1fixcnv -d<data> [-r<type>|-x<type>] [-s<scale>]  [-c] [-h]
```

Option:

-i<file name> Specifies an input file name. When this option is specified, the file conversion mode is selected.

-o<file name> Specifies an output file name. When this option is specified, the file conversion mode is selected.

-d<data> Specifies only one input data. When this option is specified, the direct conversion mode is selected.

-r<type> Specifies a format before the conversion. Converts the specified format into real number format.
0: 16-bit 2's complement fixed-point format
1: 28-bit 2's complement fixed-point format

-x<type> Specifies a format after the conversion. Converts the real number format into the specified format.
0: 16-bit 2's complement fixed-point format
1: 28-bit 2's complement fixed-point format

-s<scale> Specifies a scaling value in real number. The result of an input data multiplied by this value is converted.
When this option is not specified, it is regarded as 1.0.

-c Adds an input data etc. to the end of the line as a comment.

-h Displays the usage.

[Notes]

- < > indicates the content must be input. And, [ ] indicates the content can be omitted.
- When both options of –r and –x are omitted, it corresponds to -x0 option.
- For a comment to added to –c option, see "3.4 Execution".
- The order to assign the options is arbitrary.
- When only "dd-1fixcnv" is input, its usage is displayed. (It corresponds to -h option.).

3.2 Input Data Format

This section describes the notation system of an input data.

[Note]

- When performing a format conversion, its format is checked and if it cannot be converted, an error message will be output.

3.2.1 Real Number Format

The following shows the notation system of the data in real number format.

- Only numeric character ｛0123456789｝ and ｛.｝ can be used as an input data.
- {+-} that indicates a sign of an input data can be used immediately before numeric characters.
- {eE} that indicates an exponent character of an input data, and an exponent value can be used immediately after numeric characters.
- Use half-size characters only.

Examples of the notation:

Data that can be converted	Data that cannot be converted	Reasons for not being able to convert
-1.0	-　1.0	A sign must be placed immediately before the numeric character.
2.22	２．２２	The numeric character must be half-size characters.
3.33E+21	3.33 E+21	The exponent must be placed immediately after the numeric character.

3.2.2 2’s Complement Fixed-point Format

The following shows the notation system of the data in 2’s complement fixed-point format.

- Only alphanumeric characters ｛0123456789abcdefABCDEF｝can be used as an input data.
- In file conversion mode, only 4 digits can be used as an input data. And, when 28-bit 2’s complement fixed-point format is selected, data is handled as two lines per data.
- In direct conversion mode, when 16-bit 2’s complement fixed-point format is selected, only 4 digits can be used as an input data.
- In direct conversion mode, when 28-bit 2’s complement fixed-point format is selected, only 7 digits can be used as an input data.
- Use half-size characters only.

Examples of the notation:

1) 28-bit 2’s complement fixed-point format in direct conversion mode

Data that can be converted	Data that cannot be converted	Reasons for not being able to convert
1234567	1234	The data must be composed of 7 digits.
1234AbC	１２３４AbC	The numeric character must be half-size characters.

2) Other then the above

Data that can be converted	Data that cannot be converted	Reasons for not being able to convert
1234	123	The data must be composed of 4 digits.
01Ab	０１Ab	The numeric character must be half-size characters.

3.3 Input File Format

The following shows the notation system of the data for an input file.

- Writes input data line by line.
- Characters following “//” are a comment.
- A line only with a comment or a blank line can be used.
- Blank and tab characters can be used as characters placed between the line head and an input data.
- Blank and tab characters can be used as characters placed between an input data and a comment.
- The number of characters per line is up to 256 including a comment.
- Use half-size characters only; however, full-size characters can be used as a comment.

[Note]

- When performing a format conversion, its format is checked and if it cannot be converted, an error message will be output.

Examples of the notation:

Data that can be converted	Data that cannot be converted	Reasons for not being able to convert
-1.0	-1.0, -3.0	One data must be described in a line.
1234 // comment	1234 ／／ comment	// must be half-size character.

3.4 Execution

3.4.1 File Conversion Mode

Specify an input file and output file by –i and -o options, respectively. After the execution, the end message appears and an return value returns.

Execution Result	End Message	Return Value
Successful conversion	Complete.	0
Occurrence of any error or warning	Warning or Error.	1

3.4.1-1 An example of format conversion: real number to 16-bit 2's complement fixed point

```
dd-1fixcnv -iinput.txt -ooutput.txt –x0
```

input.txt:

```
0.5
0..5
```

output.txt:

```
4000
????
```

output.txt: (-c option applied)

```
4000                    // 0.5
????                    // 0..5 [Error] Invalid input.
```

[Note]

- The data that cannot be output is displayed as "????".

3.4.1-2 An example of format conversion: real number to 28-bit 2's complement fixed point

```
dd-1fixcnv -iinput.txt -ooutput.txt –x1
```

input.txt:

```
0.999934554
0..999934554
```

output.txt:

```
7FFD
0DB0
????
????
```

output.txt: (-c option applied)

```
7FFD                    // 0.999934554
0DB0
????                    // 0..999934554 [Error] Invalid input.
????
```

[Notes]

- The upper 16 bits of 28-bit data is displayed on the first line.
- The lower 12 bits of 28-bit data is displayed on the second line. The data is displayed with LSB justified.
- The data that cannot be output is displayed as "????".

3.4.1-3 An example of format conversion: 16-bit 2’s complement fixed point to real number

```
dd1fixcnv –iinput.txt -ooutput.txt –r0
```

input.txt:

```
4000
400x
7FFC
```

output.txt:

```
 0.5
????
 0.9998779296875
```

output.txt: (-c option applied)

```
 0.5                   // 4000
????                   // 400x [Error] Invalid input.
 0.9998779296875       // 7FFC
```

[Note]

- The data that cannot be output is displayed as “????”.

3.4.1-4 An example of format conversion: 28-bit 2’s complement fixed point to real number

```
dd1fixcnv –iinput.txt -ooutput.txt –r1
```

input.txt:

```
1234
0ABC
1234
0ABx
```

output.txt:

```

 0.14223238825798

????
```

output.txt: (-c option applied)

```
                        // 1234
 0.14223238825798      // 0ABC
                        // 1234
????                   // 0ABx [Error] Invalid input.
```

[Notes]

- The result of the following formula becomes an input data.
  ({data in the first line} * 0x1000) | ({data in the second line} & 0x0FFF)
- The data that cannot be output is displayed as “????”.

3.4.2 Direct Conversion Mode

Specify an input data by –d option. After the execution, an output data appears and an a return value returns.

Execution Result	Return Value
Successful conversion	0
Occurrence of any error or warning	1

3.4.2-1 An example of format conversion: real number to 2's complement fixed point

e.g.:

```
C:\>dd-1fixcnv -d0.25 –x0
2000

C:\>dd-1fixcnv -d0.25 –x1
2000000

C:\>dd-1fixcnv -d0.25 –x0 -c
2000                    // 0.25

C:\>dd-1fixcnv -d0..25 –x0
???
[Error] Invalid input.
```

［**Note**］

- The data that cannot be output is displayed as "????".

3.4.2-2 An example of format conversion: 2's complement fixed point to real number

e.g.:

```
C:\>dd-1fixcnv –d4000 –r0
 0.5

C:\>dd-1fixcnv –d4000000 –r1
 0.5

C:\>dd-1fixcnv –d4000 –r0 -c
 0.5                    // 4000

C:\>dd-1fixcnv –d12345 –r0
????
[Error] Invalid input.
```

［**Note**］

- The data that cannot be output is displayed as "????".

4 Error and Warning Message

1) Conversion from real number format into 2's complement fixed-point format:

Message	Description
[Error] Invalid input.	Character strings that cannot be converted are used as an input data.
[Warning] Overflowed.	The output data exceeded the maximum value that can be converted. When an input data is positive, the output data was corrected to "0x7FFF(FFF)", and when negative, the output data was corrected to "0x8000(000)".
[Warning] Underflowed.	The output data fell below the minimum value that can be converted. And, the output data was corrected to "0x0000(000)".

2) Conversion from 2's complement fixed-point format into real number format:

Message	Description
[Error] Invalid input.	Character strings that cannot be converted are used as an input data.

3) Conversion from 28-bit 2's complement fixed-point format into real number format:

Message	Description
[Warning] bit15-12 ignored.	MSB 4 bits of the data in the second line was converted, regarded as 0x0.

4) Others

Message	Description
Denied access to xxx.	The data cannot be written into an output file.
Wrong syntax.	An option is not specified correctly.
xxx was not found.	Cannot find any input file.
xxx : No such option or illegal option parameter.	Illegal option was specified. Or, the value of the option was out of the range.