VS1005 VSOS SHELL

VS1005g

Revision History

Rev.	Date	Author	Description
1.20	2015-07-17	HH	Added Rec - an audio recorder.
1.11	2015-06-01	HH	PlayDir developed further.
1.10	2015-05-29	HH	Initial release of documentation.

Contents

List of Figures

1 Introduction

The VS1005 VSOS Shell is a powerful tool that allows controlling VS1005 from an external interface, e.g. the UART.

This document will explain how to install and use the VSOS Shell Environment, as well as the UART Controlled Audio Player that is included in this package.

After the disclaimer and definitions in Chapters 2 and 3, an overview of the Shell is given in Chapter 4, *Overview*. It is followed by requirements in Chapter 5, *Requirements*, and shell installation instructions in Chapter 6, *Installing the VSOS Shell Environment*.

Instruction on using the shell is given in Chapter 7, *Using the Shell Environment*.

The UART Player is explained in Chapter 8, *Using the UART Controlled Player*.

The document ends with Chapter 9, *Latest Document Version Changes*, and Chapter 10, *Contact Information*.

2 Disclaimer

VLSI Solution makes everything it can to make this documentation as accurate as possible. However, no warranties or guarantees are given for the correctness of this documentation.

3 Definitions

DSP Digital Signal Processor.

I-mem Instruction Memory.

LSW Least Significant (16-bit) Word.

MSW Most Significant (16-bit) Word.

RISC Reduced Instruction Set Computer.

VS_DSP^4 VLSI Solution's DSP core.

VSIDE VLSI Solution's Integrated Development Environment.

VSOS VLSI Solution's Operating System.

X-mem X Data Memory.

Y-mem Y Data Memory.

4 Overview

The VS1005 VSOS Shell is a new, powerful tool that allows controlling VS1005 from an external interface, e.g. the UART.

Using the VSOS Shell interface it is possible to create an audio player and recorder system (recorder not yet available) that doesn't require any specific VS1005 programming skills.

Nevertheless, for those who are familiar with programming VS1005, the VSOS Shell Environment makes it possible to create convenient shell commands / programs. By combining these building blocks it is possible to create more complex systems, controlled either by an external microcontroller or VS1005 itself.

5 Requirements

Before installing the VSOS Shell Environment, you need to have the following building blocks:

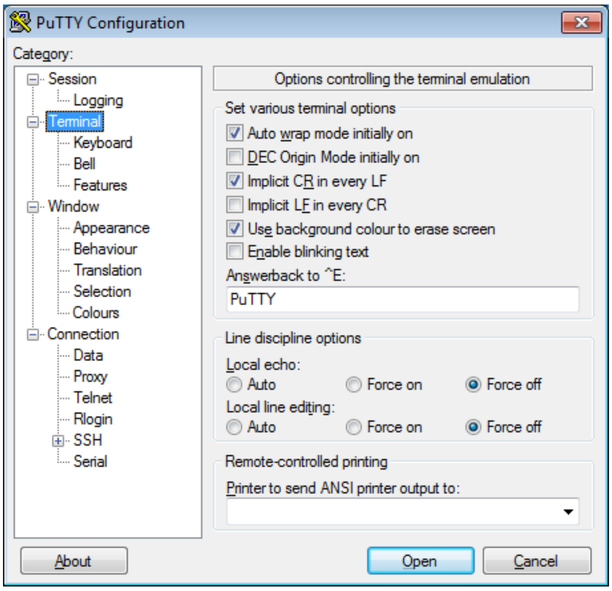
- VS1005g Developer Board. The VS1005g BreakOut Board should work, too, but these instructions have been tested with the DevBoard.
- Latest version of VSOS installed (at least 3.22).
- USB cable between DevBoard and PC for uploading new software.
- UART or USB->UART cable connected between DevBoard and PC for using the UART interface. Data speed is 115200 bps, format is 8N1.
- Your favorite UART Terminal Emulation program installed on the PC. Note that the current VSOS Shell uses only the line feed character (0x0a) for line feed, and no carriage return (0x0d), so the terminal emulation program needs to be able to handle that (see examples in Chapter 5.1, *Terminal Program Settings*). For Microsoft Windows computers, PuTTY has been tested and found working.

When all of this is in order, you are ready to install the VSOS Shell Environment.

5.1 Terminal Program Settings

This chapter shows recommended settings for two example terminal programs, tested by VLSI. Other terminal programs can also be used, provided that the crucial parameters are set in a similar way.

5.1.1 Settings for PuTTY



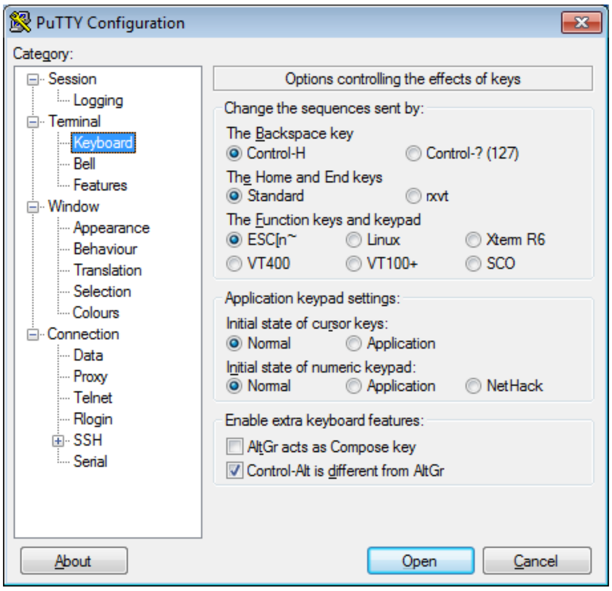
Options controlling the terminal emulation

Set various terminal options
☑ Auto wrap mode initially on
☐ DEC Origin Mode initially on
☑ Implicit CR in every LF
☐ Implicit LF in every CR
☑ Use background colour to erase screen
☐ Enable blinking text
Answerback to ^E: PuTTY

Line discipline options
Local echo: ○ Auto ○ Force on ● Force off
Local line editing: ○ Auto ○ Force on ● Force off

Remote-controlled printing
Printer to send ANSI printer output to:

Figure 1: PuTTY Configuration: Terminal



Options controlling the effects of keys

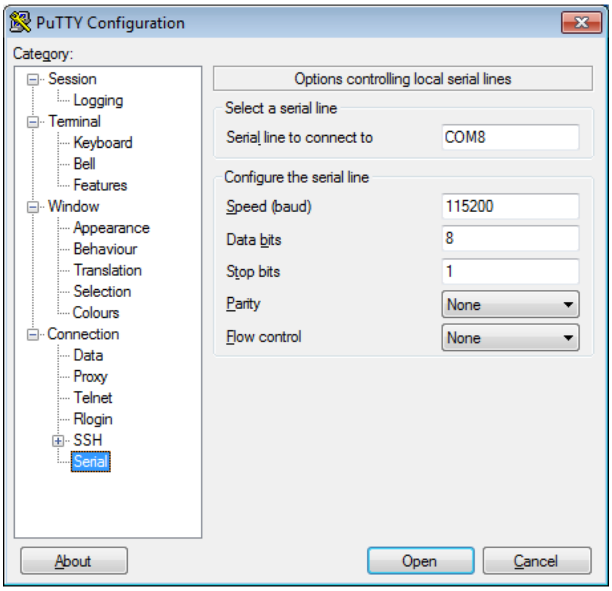
Change the sequences sent by:
The Backspace key: ● Control-H ○ Control-? (127)
The Home and End keys: ● Standard ○ rxvt
The Function keys and keypad: ● ESC[n~ ○ Linux ○ Xterm R6 ○ VT400 ○ VT100+ ○ SCO

Application keypad settings:
Initial state of cursor keys: ● Normal ○ Application
Initial state of numeric keypad: ● Normal ○ Application ○ NetHack

Enable extra keyboard features:
☐ AltGr acts as Compose key
☑ Control-Alt is different from AltGr

Figure 2: PuTTY Configuration: Keyboard

Figure 1 shows the terminal emulation settings for PuTTY. Make sure you check the “Implicit CR in every LF” box. In the keyboard settings in Figure 2, you may set Backspace key to either Control-H or Control-?.



Options controlling local serial lines

Select a serial line

Serial line to connect to: COM8

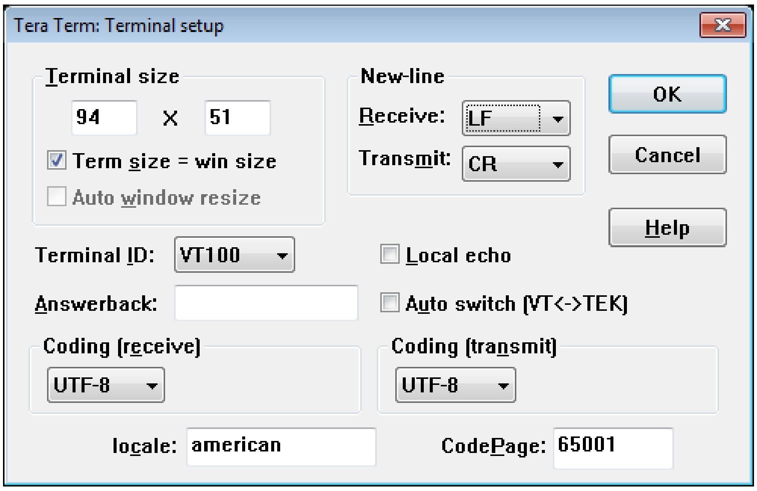
Configure the serial line

Speed (baud): 115200
Data bits: 8
Stop bits: 1
Parity: None
Flow control: None

Figure 3: PuTTY Configuration: Serial

Figure 3 shows the serial communication parameters for 115200 bps, 8N1. For binary file transfers to work, it is important to disable flow control.

5.1.2 Settings for Tera Term



Terminal size: 94 X 51
☑ Term size = win size
☐ Auto window resize

New-line
Receive: LF
Transmit: CR

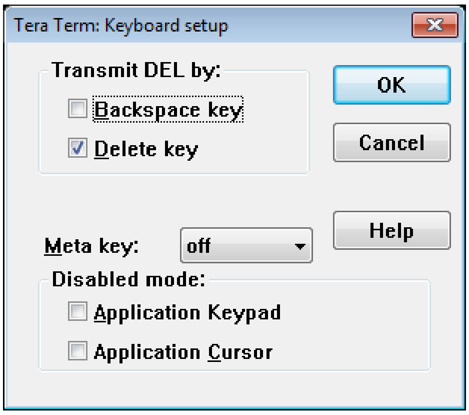
Terminal ID: VT100
Answerback:
☐ Local echo
☐ Auto switch (VT<->TEK)

Coding (receive): UTF-8
Coding (transmit): UTF-8

locale: american
CodePage: 65001

Figure 4: Tera Term: Terminal Setup

Figure 4 shows how to set line feeds and basic terminal emulation mode (VT100) in Tera Term.


Tera Term: Keyboard setup

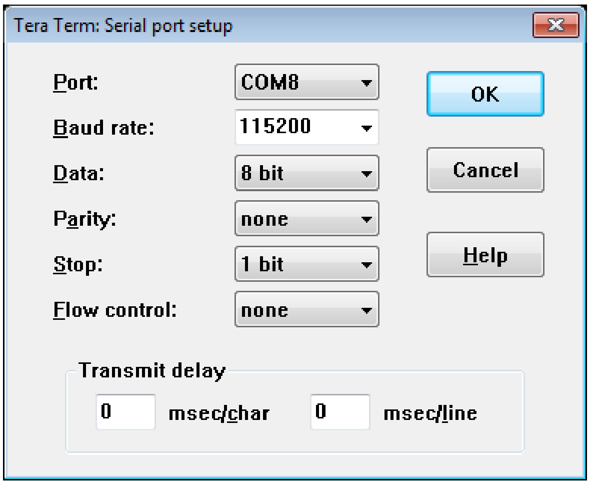
Transmit DEL by:
☐ Backspace key
☑ Delete key

Meta key: off

Disabled mode:
☐ Application Keypad
☐ Application Cursor

Figure 5: Tera Term: Keyboard Setup

Figure 5 shows a working keyboard setup for Tera Term.


Tera Term: Serial port setup

Port: COM8
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char
0 msec/line

Figure 6: Tera Term: Serial Port Setup

Figure 6 shows how to set serial parameters 115200 bps, 8N1, without flow control.

6 Installing the VSOS Shell Environment

To install the VSOS Shell Environment, you need to do the following:

1. You need to do this first step only if you want to modify and compile the examples in the solutions folder of this package, and if your VSIDE is v2.34 or older!
   Copy libvs1005_vsos3/consolestate.h in this package to your PC's VSIDE installation directory libvs1005_vsos3/ .
2. Boot your DevBoard in USB Mass Storage Device mode (push S1 while resetting the board).
3. Make a backup of your DevBoard config.txt file in case you want to return to it later.
4. Copy the following files from the root/ directory of this package:
   - config.txt and shell.ap3 to the root directory of your DevBoard.
   - All files in the SYS/ directory to the SYS/ directory of your DevBoard.
5. Safely remove / unmount the VS1005 USB device.
6. Start the UART terminal emulation program on your PC. Set parameters to 115200 bps, 8N1.
7. Reset the DevBoard.

The LCD on the VS1005 DevBoard should stay blank, but you should now see output like the following on your PC's terminal screen:

```
Hello.
VSOS 3.22 build Mar 10 2015 15:51:56
VLSI Solution Oy 2012-2015 - www.vlsi.fi

Starting the kernel..
Starting Devices...
External SPI Flash

Installed system devices:
S: SPI Flash c814, handled by FAT.
Load drivers, config 0...
Driver: RUN... FC00,D Y:0xfc00: 0x2000-13 -> 0x0
Driver: RUN... FC00,C Y:0xfc00: 0x0-12 -> 0x0
Driver: SDSD... D: SD card in SD mode

Driver: UARTIN...
Driver: S:SHELL.AP3...
VSOS SHELL
S:>
```

If this is the case, you have successfully started the VSOS Shell Environment.

7 Using the Shell Environment

The shell environment contains the following features:

- Very small (roughly 1/4 kilowords code size).
- Executes commands from the S:SYS/ folder.
- Can also execute .dl3 and .ap3 programs from other locations if full path is provided.
- Interactive command line editing.
- Previous command can be retrieved with up arrow (vt100 terminal).
- Concept of current directory.
- Buffered, interrupt handled UART stdio driver.
- Ctrl-C to notify programs that they should close.
- Ctrl-C three times to hard reset VS1005.
- Interactive (echo +e) or machine-controlled silent (echo -e) mode.

If the shell is in interactive mode, it will present a command prompt.

If in non-interactive mode, the shell will display a line containing the hash character “#” when expecting input from the user. After receiving the full command line, it will output a line containing “:” before executing the command.

This package contains several programs for the shell, many of which take command line arguments.

7.1 Shell Programs

This section presents the shell programs included in this package.

The program are in rapid development, so while this documentation is valid as of writing this (2015-05-29), it will probably soon be out of date. Many programs offer a “-h” command line option to show what options they can handle.

7.1.1 cd

```
Usage: cd [-h] [dir1 [dir2 [...]]]
-h      Show this help
Note: cd without parameters shows the current directory
      cd :: lists all available devices
```

Change or print current directory or list all devices. Only one directory step may be defined in one parameter, but there may be several parameters.

7.1.2 dir

```
Usage: dir [-s|+s|-a|+a|-f|+f|-h] [fullpath]
-s      Sort files
+s      Don't sort files (faster with large directories)
-a      Only audio files
+a      All files
-f      Fast listing (if -a selected)
+f      Slower listing (if -a selected)
-h      Show this help page
```

dir lists the contents of a directory. It can optionally examine each file and list only those ones that are audio media files.

If “-s” is defined but the directory is too large to be sorted in memory, dir lists the files in file system order.

The output of dir is as e.g. as follows:

```
D:ManMachine/>dir +s
D    1. .                     0 .
D    2. ..                    0 ..
-    3. 03_MET~1.WAV   63755708 03_Metropolis.wav
-    4. 04_THE~1.WAV   39508940 04_TheModel.wav
-    5. 05_NEO~1.WAV   94268204 05_NeonLights.wav
-    6. 06_THE~1.WAV   58616588 06_TheManMachine.wav
-    7. 01_THE~1.WAV   65656124 01_TheRobots.wav
-    8. 02_SPA~1.WAV   62633804 02_Spaceleb.wav
```

The fields are as follows:

1. Entry type. ’D’ is a directory, ’-’ is a normal file, and ’L’ is a volume label.
2. The file system order of the entry.
3. Short name.
4. File size in bytes.
5. Long name.

When listing audio files, the output format is different:

```
D:ManMachine/>dir -a
-    7. 01_THE~1.WAV    6:12.2  44100 2 01_TheRobots.wav
-    8. 02_SPA~1.WAV    5:55.1  44100 2 02_Spaceleb.wav
-    3. 03_MET~1.WAV    6:01.4  44100 2 03_Metropolis.wav
-    4. 04_THE~1.WAV    3:44.0  44100 2 04_TheModel.wav
-    5. 05_NEO~1.WAV    8:54.4  44100 2 05_NeonLights.wav
-    6. 06_THE~1.WAV    5:32.3  44100 2 06_TheManMachine.wav
```

Now the fields are as follows:

1. Entry type. ’D’ is a directory, ’-’ is a normal file, and ’L’ is a volume label.
2. The file system order of the entry.
3. Short name.
4. File playback time in m:ss.t format.
5. Sample rate.
6. Number of audio channels.
7. Long name.

Determining playback times for MP3 files requires scanning the whole file, and may be slow. Because of this, unless +f is defined with -a, MP3 file playback times are shown as 0:00.0.

7.1.3 echo

```
Usage: echo [-n|+n|-e|+e|-|-h] string
-n       No newline
+n       Output newline
-e       Turn shell interactive echo mode off
+e       Turn shell interactive echo mode on
-        End of parameters
-h       Show this help
Note: String may contain escape codes such as \" and \n
```

Echoes its input on the screen.

Options “-e” and “+e” switch the global interactive mode off and on, respectively. The non-interactive mode may be easier to handle when the shell is used with a microcontroller.

7.1.4 getcmd

```
Usage: getcmd
```

Reads a command line for the shell.

Not for command line use. To see how to call this function, see the source code for the shell.

7.1.5 ls

```
Usage: ls [path]
```

Show a short listing of a directory in file system file order.

7.1.6 metadata

```
Usage: metadata filename
```

Shows and/or returns metadata for an audio file. The output is in UTF-8 format, and looks as follows (but not necessarily in this order):

```
~0503'Song name
~0504'Album
~0505'Artist
~0506'Year
~050d'Track number
```

Currently metadata can only decode MP3 ID3v1 and ID3v2 tags, but it will later be expanded to read metadata from other file formats, too.

metadata can be used either from the command line, or from another application. For how to use metadata, see the source code for PlayDir.

7.1.7 paramspl

```
Usage: paramspl parameters
```

Split a parameter string to null-terminated strings, taking into account parenthesis and escape characters.

Not for command line use. To see how to call this function, see the source code for echo.

7.1.8 PlayDir

```
Usage: PlayDir [-v|+v|-p|+p|-s|+s|-h]
-v      Verbose
+v      Not verbose
-p      Start in pause mode
+p      Start in play mode
-s      Shuffle mode on
+s      Shuffle mode off
-h      Show this help page
```

Plays all audio files in the current directory.

PlayDir is described in detail in Chapter 8.1.

7.1.9 PlayFile

```
Usage: PlayFile file
```

Plays one file in the file system. Can output some of the metadata that PlayDir does.

7.1.10 PlayFiles

```
Usage: PlayFiles pattern
```

Plays files in the file system in the order they appear in the file system. Can output some of the metadata that PlayDir does.

Example:

```
PlayFiles *
```

7.1.11 Rec

```
Usage: Rec [-cs|-cm|-cl|-cr|-q{x}|-b{x}|-r{x}|-h]
-cs|-cm|-cl|-cr Stereo|Mono|Left/|Right
-qx     Set quality to x (0-10, higher is better)
-bx     Set bitrate to x kbit/s (1-511)
-rx     Set sample rate to x Hz, or x kHz if x<1000
-h      Show this help
outFile The output file name (e.g. D:REC.MP3)
```

Records MP3 to a file.

Rec is described in detail in Chapter 8.2.

7.1.12 run

```
Usage: run cmd param
```

Runs command *cmd* with parameters *param*, then remove *cmd* from memory.

Not for command line use, but useful in the startup script config.txt.

7.1.13 type

```
Usage: type [-c|+c|-h] [file1 [file2 [...]]]
```

```
-c      Print file information line
+c      Don't print file information
-h      Show this help
```

Type a file to screen.

7.1.14 uartin

```
Usage: uartin
```

Interrupt-based UART stdin/stdout driver.

Not for command line use.

7.1.15 ybitclr

```
Usage: ybitclr xxxx,y
```

Clears bit *y* in Y memory register *xxxx*. Both xxxx and y are hexadecimal number.

8 Using the UART Controlled Player

To test the UART Controlled Player, you need to do the following steps:

1. Copy some audio files to an SD card.
2. Insert SD card to VS1005 DevBoard.
3. Boot DevBoard to the VSOS Shell Environment.
4. Use `cd` to get to your audio content directory, and potentially `dir -a` or `dir -a +f` to list it.
5. Type `playdir` to run PlayDir.
6. Use PlayDir keyboard shortcuts and control commands described in Chapter 8.1.

8.1 PlayDir: Interactive UART Player

```
Usage: PlayDir [-v|+v|-p|+p|-s|+s|-h]
-v       Verbose
+v       Not verbose
-p       Start in pause mode
+p       Start in play mode
-s       Shuffle mode on
+s       Shuffle mode off
-h       Show this help
```

PlayDir plays the files in sorted order in the current directory. If there is not enough memory to sort the files, they are played in file system order.

First it takes a listing of the files in the current directory using *dir*.

Then, it outputs the following line:

```
~0205=Number of audio media files encountered
```

If PlayDir was started with the “-v” command line option, it will then list all audio files with potential metadata.

After the optional list, PlayFile will open the first file, see if it can find metadata, and outputs information for the file. When file playback has finished, PlayFile will play the next file. When it has played all files, it exits.

NOTE!
For systems with a microcontroller, it is strongly recommended that machine-controlled silent mode (echo -e) is switched on before starting PlayDir. For interactive use the default interactive echo mode (echo +e) is recommended.

8.1.1 PlayDir Output

For each file, and depending on whether metadata for the file was found, the following fields may be output (not necessarily in this order):

```
~0308=PlayDir's playlist track number, always first field, always printed
~0205=PlayDir's song number, always second field, always printed
~050b'File name, always third field, always printed
~0503'Song name
~0504'Album
~0505'Artist
~0506'Year
~050d'Track number
```

If not in shuffle mode, the Playlist Track Number and Song Number are the same. If in shuffle mode, the Playlist Track number is increasing, while the Song Number is shuffled.

When the song is playing, the following messages are being sent once a second:

```
~030a'Playback time in seconds
~030b'File position in per cent (0..100), only sent if changed
```

When master volume changes, the following message is shown:

```
~0206=volume
```

where *volume* is the attenuation from maximum volume in 0.5 dB steps. Setting for maximum volume is 0 (-0 dB), minimum volume is 254 (-127 dB).

When pause mode is toggled, the following message is shown:

```
~0104=pauseMode
```

where *pauseMode* is 1 if in pause mode, or 0 if in playback mode.

If the last song of a playlist is playing, and next song is selected, a line containing only the character '>' is printed, and PlayDir exits. If the first song is playing and previous song is selected, a line containing only '<' is printed, and PlayDir exits.

When PlayDir closes, it outputs the following line:

```
~0205=0
```

8.1.2 PlayFile Control Keys

PlayFile has the following controls keys:

n Next song. If last song was passed, '>' is printed and player exits

p Previous song. If first song was passed, '<' is printed and player exits

whitespace Toggle pause mode

< Lower volume by 0.5 dB

> Increase volume by 0.5 dB

. Fast forward 10 seconds

: Fast forward 60 seconds

, Rewind 10 seconds

; Rewind 60 seconds

q or Ctrl-C Exit player

8.1.3 PlayFile Control Commands

In addition to one-character controls, PlayFile also accepts control commands that are of the following format
`~xxxx=y`
where `xxxx` is an exactly 4-character UI Control message in hexadecimal, and `y` is the value as a C formatted number (decimal, hexadecimal, or octal).

An example that sets playback volume to -10 dB of maximum:
`~0206=20`

If the shell is in interactive echo mode (echo +e), then output of file progress messages are suppressed after the command-initiating '~' character to let the user more easily to see what he is typing. In machine-controlled silent mode (echo -e) text output continues as normal.

PlayDir recognizes the following control commands:

Control commands recognized by PlayDir

Msg	Val		Description
	Min	Max	
0104	0	1	Set pause mode (0=off, 1=on).
0183	0	1	Set shuffle mode (0=off, 1=on).
0308	1	32767	Select track from current playlist. If out of bounds (0 or too large), PlayDir exits. Shuffle mode is automatically deactivated.
0206	0	255	Set volume to -`val`/2 dB of maximum. Largest value (255) shuts down analog drivers.
030a	0	65534	Go to `val` seconds in audio file.

If shuffle mode is activated with the set shuffle mode command, shuffle mode starts when the current song ends. When shuffle mode is deactivated, playback is continued from the current track.

8.2 Rec: Interactive UART Recorder

```
Usage: Rec [-cs|-cm|-cl|-cr|-q{x}|-b{x}|-r{x}|-h]
-cs|-cm|-cl|-cr Stereo|Mono|Left/|Right
-qx     Set quality to x (0-10, higher is better)
-bx     Set bitrate to x kbit/s (1-511)
-rx     Set sample rate to x Hz, or x kHz if x<1000
-h      Show this help
outFile The output file name (e.g. D:REC.MP3)
```

Rec records MP3 audio to a given file. The user may define the channel modes, sample rate, and either set a quality or a suggested bitrate.

If quality is set, the encoder used Variable BitRate (VBR) encoding, which gives the best quality / file size ratio. If the bitrate is set, the MP3 encoder uses Constand BitRate (CBR).

Sample rate may be set to either 24 or 48 kHz.

NOTE!
For systems with a microcontroller, it is strongly recommended that machine-controlled silent mode (echo -e) is switched on before starting Rec. For interactive use the default interactive echo mode (echo +e) is recommended.

NOTE!
As a safety feature, Rec stop recording whenever the recording file grows to at least one billion bytes (1 GB).

8.2.1 Rec Output

When startingFor each file, and depending on whether metadata for the file was found, the following fields may be output (not necessarily in this order):

```
~050a'fileName
Output buffer size xx bytes
```

When the song is playing, the following message is sent once a second:

```
~030a'Playback time in seconds
```

When pause mode is toggled, the following message is shown:

```
~0104=pauseMode
```

where *pauseMode* is 1 if in pause mode, or 0 if in playback mode.

When Rec closes, it outputs the following line:

```
~010b=1
```

8.2.2 Rec Control Keys

Rec has the following controls keys:

n Next song. (Currently closes the program.)
whitespace Toggle pause mode
q or Ctrl-C Exit recorder

8.2.3 Rec Control Commands

In addition to one-character controls, Rec also accepts control commands that are of the following format
`~xxxx=y`
where `xxxx` is an exactly 4-character UI Control message in hexadecimal, and `y` is the value as a C formatted number (decimal, hexadecimal, or octal).

Rec also accepts report requests that are of the following format
`~xxxx=y`
where `xxxx` is an exactly 4-character UI Control message in hexadecimal.

An example that sets pause mode on:
`~0104=1`

If the shell is in interactive echo mode (echo +e), then output of file progress messages are suppressed after the command-initiating ’~’ character to let the user more easily to see what he is typing. In machine-controlled silent mode (echo -e) text output continues as normal.

Rec recognizes the following control commands:

Control commands recognized by Rec			
Msg	**Val**		**Description**
	Min	**Max**	
0104	0	1	Set pause mode (0=off, 1=on).

Rec recognizes the following report requests:

Report requests recognized by Rec	
Msg	**Description**
030d	Return file position in bytes.
030c	Return average bitrate for file in bps.
0209	Return maximum signal level in dB since last time read.

Request 0209 can be used to build a maximum level VU meter. Such a meter should warn the user (with e.g. yellow vs green colour) whenever the input level exceeds -5 dB, and show a clear overload error (with e.g. red colour or an “OVERLOAD” text) whenever maximum absolute level 0 dB is reached.

9 Latest Document Version Changes

This chapter describes the most important changes to this document.

Version 1.20, 2015-07-17

- Added an audio recorder program, Rec, Chapter 7.1.11.

Version 1.11, 2015-06-01

- Added new functionality to PlayDir, Chapter 7.1.8, and slightly changed its output.
- Added terminal emulation program screen caps to Chapter 5, *Requirements*.

Version 1.10, 2015-05-29

First release with documentation. Added PlayDir.

Version 1.00, 2015-05-20

First release.

10 Contact Information

VLSI Solution Oy
Entrance G, 2nd floor
Hermiankatu 8
FI-33720 Tampere
FINLAND

URL: http://www.vlsi.fi/
Phone: +358-50-462-3200
Commercial e-mail: sales@vlsi.fi

For technical support or suggestions regarding this document, please participate at
http://www.vsdsp-forum.com/
For confidential technical discussions, contact
support@vlsi.fi