VS1010 Developer board quick start

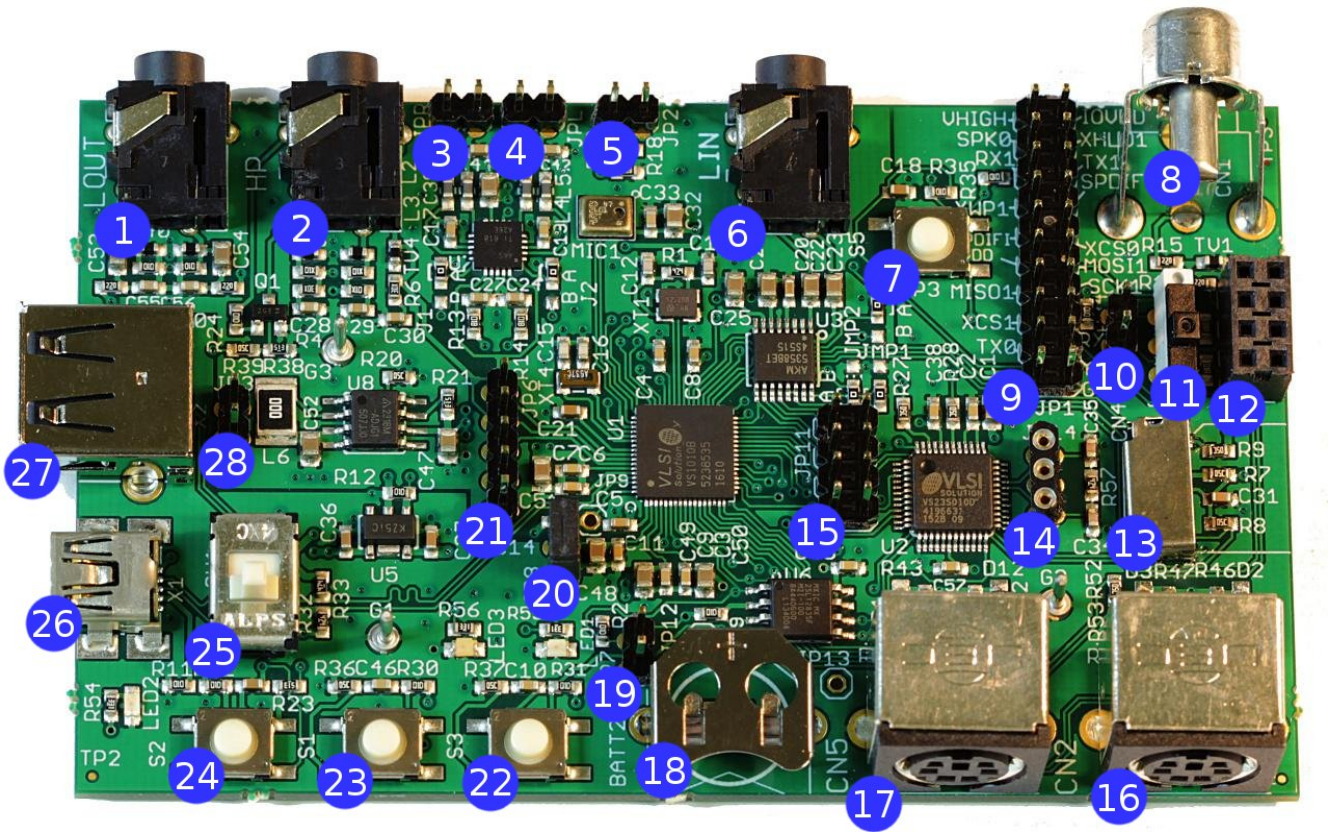

Illustration 1: VS1010 Developer board version 1.2

Disclaimer

This is a preliminary document version. All properties and figures are subject to change.

Table of Contents

1 Required equipment

- Developer board (ICs: VS1010, VS23S010, audio power amplifier, SPI EEPROM, I2S ADC)
- VSIDE USB-UART cable
- microSD card
- USB mini-B cable
- VSIDE compatible computer

2 Input and output

Table 2.1 includes the various connectors, jumpers, buttons, and switches that the Developer board has. Highlighted with gray are buttons and switches. See Illustration 1 to locate them on the actual board.

#	**Name**	**Description**
1	LOUT	Line-out
2	HP	Headphones
3	JPR	Amplified signal from a D class amplifier to speakers
4	JPL	Same as above
5	JP2	Connects to VS1010 SAR channel AUX2.
6	LIN	Line-in
7	S5	Reset button
8	TV1	Video-out
9	JP1	Pin header. See sub-chapter “JP1 Header” for details.
10	JP4	Provides access for XCS1 and MOSI1 signals from VS1010. Close jumper to enable video signal.
11	JP5	Video output switch. See sub-chapter “Video output” for more.
12	JP10	Module port for external module. Gives access to the VS1010 GPIO 2_7 pin, second UART and VS23S010 GPIO pins
13	CN4	microSD slot. MicroSD card is used by the VS1010 as both the media storage and also for storing custom application code and console commands that can be run by the VS1010. Use the FAT32 file system.
14	JP14	Video crystal header. See sub-chapter “Video output” for more.
15	JP11	I2S Digital audio
16	CN2	PS/2 Keyboard connector
17	CN5	PS/2 Mouse connector
18	BATT2	Battery for real time clock

19	JP7	Close jumper to ground XCS0
20	JP8	Close jumper to enable VS1010 power. (Closed by default.)
21	JP6	Analog audio signals straight from the VS1010, analog power, ground.
22	S3	Pause/unpause, set runlevel. See chapter “”
23	S1	Next track, set runlevel. See chapter “Buttons”
24	S2	Power on. See chapter “Buttons”
25	SW1	USB connection selection switch. Positions correspond with the locations of the connectors.
26	X1	USB Mini-B connector for connecting the board to a PC. Is currently supported.
27	X2	USB Type A connector. Planned to be used for connecting a USB flash memory when the software drivers become available.
28	JP3	Jumper to enable the D class power amplifier for the speakers. Connects the VHIGH power for a regulator which provides power for the output amplifier. (Closed by default.)

Table 2.1: Connectors, jumpers, switches and buttons. Gray highlight for switches and buttons.

2.1 JP1 Header

The JP1 header provides access to many signals connected to VS1010. It has digital audio inputs and outputs, two UARTs, SPI and powers and grounds. See Illustration 2 for exact pin-out.

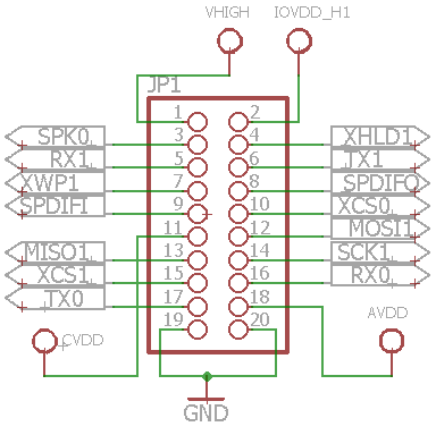

Illustration 2: JP1 Header pin-out

2.2 Video output

The Developer board provides two different video outputs from the TV1 video-out connector, VS1010 internal video debug console and the VS23S010 color video signal. To see the VS1010 console, it's enough to close the jumper JP4, and to set the video selector switch to the position closer to the TV1 connector. See Illustration 3 for an example.

Illustration 3: VS1010B internal debug console shown on an old industrial CRT TV monitor

To see the color video signal, a color carrier crystal for the VS23S010 memory/video controller IC must be connected to the header JP14. Proper frequencies are for PAL 4.433618 MHz and NTSC 3.579545 MHz. With the crystal and custom software, a color video signal can be generated with the board. The video selector switch must be placed at the position farther from the video-out connector. See Illustration 5 for an example.

2.3 Buttons

The Developer board has four buttons. Close to the top, between LIN and JP1, is S5. It's connected to XRESET signal and pressing it resets the VS1010. At the bottom are the buttons S2, S1 and S3. Notice the non-numerical order. S2 is used to turn on the board, but it cannot power it down. Use S5 whenever a reboot is needed. In the player mode S1 is used for skipping to the next track, and S3 for pausing and un-pausing.

The buttons S1 and S3 are also used for selecting different boot modes. These are described in the chapter “Booting the VS1010”.

3 Powering the VS1010 Developer board

Developer board can be powered through either the mini-USB plug or JP1 header. It also provides VHIGH to USB-A connector. VHIGH should be 5 volts.

Illustration 4 shows the proper way to power the developer board using the VSIDE UART cable. Use Illustration 2 to see the exact pin locations.

Red wire is 5V. (pin 1)

Green wire is connected to RX0 (pin 16)

White wire is connected to TX0 (pin 17)

Black wire is ground (pin 19 or 20)

Note that TX0 and RX0 are not side by side, RX0 is the *third* pin from the bottom of the connector. Once the correct cables are connected, the board can be turned on by pressing the button S2.

The UART signals have 3 volt levels. **There are no 5 volt tolerant I/O pins on the board.**

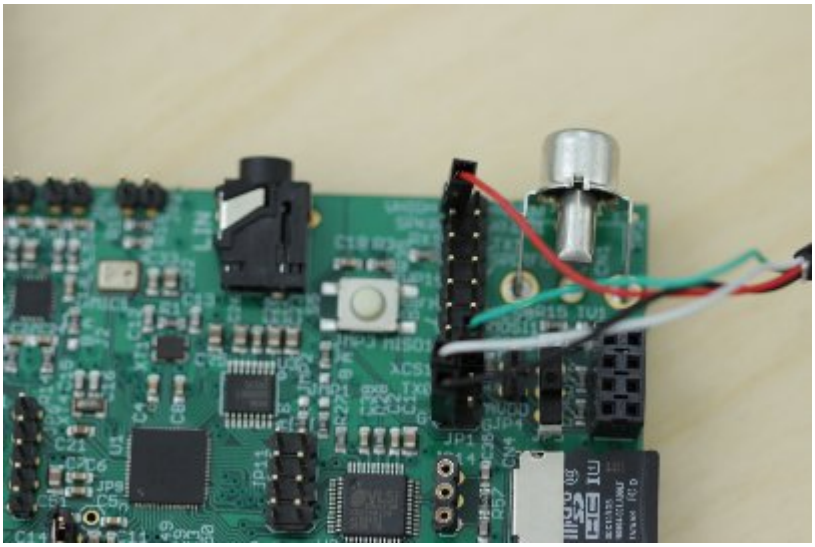

Illustration 4: VSIDE UART cable connected to UART0

4 Connecting with the UART

The provided USB-UART cable should install its drivers with Windows by simply plugging it in. If not, you may try installing the drivers manually. See chapter “Additional Information” for instructions.

To communicate with the Developer board, you also need a terminal program with serial port support. There is a variety of such programs available, for example, Tera Term. Before you can communicate with the board, you need to set up the connection settings.

For the settings, you will need the name of the COM port the USB-UART cable is using. On Windows, go to the Device manager, find the group “Ports (COM and LPT)”, and under it “Prolific USB-to-Serial Comm Port (COMx)”, where x is a positive integer. The COM port number is what you need.

To set up the connection with Tera Term, for example, go to File → New Connection, choose “Serial”, and pick the correct COM port from the list. Then go to Setup → Serial Port. The correct settings are as follows:

Port: the COM port
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none
Transmit delays: 0

The serial connection should be now running. The next chapter has some examples on how to use it.

Illustration 5: VS1010B demo setup with PS/2 keyboard and the VS23S010 color video output

5 Booting the VS1010

Note: SPI booting on the current VS1010 developer boards is unreliable due to chip select conflict between the VS23S010 and the boot flash. This can be fixed in 2 ways: 1) adding a pull-up resistor to the VS23S010 chip select signal or 2) running a patch software from the SD card. Developer boards shipped by VLSI only support booting from the SD card by default.

VS1010 boot order:

1. SPI EEPROM
2. SPI Flash
3. UART0 VS3EMU monitor.
4. SD card
5. Internal ROM player and the command line

If JP7 is closed with jumper, SPI boot is skipped. This is required if SPI memory was flashed with bad image which prevents proper booting.

After VSOS loads, console output can be seen in UART0, and in TV1 if the board is set up for it. Sample console output:

```
B#0d1d
VS1010B VSOS 2.78
Files:6. Buffers:3.
Runlevel 0
SD:3796 MB
```

The first line is a boot status message:

“B” means VS1010B.

“0” means that SPI bus 0 (internal SPI) is selected.

“d” means “done”, e.g. no boot image is found.

“1” means that SPI bus 1 (external SPI bus 0) is selected,

“d” again means “done”, no boot image is found.

If a boot image would have been found, there would be “f” stating a flash boot or “e” stating an eeprom boot.

Next is the chip name and the VSOS version. The line after shows the amount of file handles that can be opened (6), and the amount file buffers (3), e.g. you can open up to 3 simultaneous disk data files (the remaining three can be character device streams, for example).

Runlevel shows the boot state of the chip, i.e. what the chip is attempting to do. Runlevel 0 is the default boot, and it is the player mode. The other boot modes are explained in the sub-chapter “Runlevels”.

In the default configuration, audio files from the SD card are played. If you have configured the serial connection, you can drop fromt the player to the VSOS shell prompt by sending carriage return (13) to UART0. Doing so will also stop the player.

Note for VS1010B: if the player is paused during this, the shell won't open. Instead, an un-pause is required, after which the shell will open immediately, even without another carriage return.

The VS1010B ROM has two built-in programs, “x:D” to list files and “x:T” to type text files. Below is a sample run of them.

```
VS1010>x:D s:doc
File list of s:doc/*
1: .
2: ..
3: README.txt
 3 file(s) found.

VS1010>x:T s:doc/README.txt
Here will be documentation for the VS1010 developer board.
```

More programs can be added to the SD card or, in the future, to the flash. The Developer board also has a selection of programs which can be found in the chapter “Included system programs”.

5.1 Runlevels

The runlevel is set by holding the buttons S1, S3 or both of them, during boot or reset. See Table 2.

Runlevel	Mode	Buttons to hold
0	normal player mode	None (default boot)
1	USB ramdisk debug mode	S1
2	USB SD card reader mode	S3
3	normal player mode	S1 and S3

Table 2: Runlevels

Pay attention to the order of the buttons, as S1 and S3 are the two rightmost buttons, while S2 is the leftmost. Runlevels 1 and 2 both require the mini-USB cable to have any useful functionality.

Note: if during boot the board has a microSD card attached, with the provided SD bugfix as boot.dlx, runlevel 3 won't launch normally. However, you can work around this either by replacing or removing the boot.dlx, or by simply removing the microSD just for the duration of the boot.

6 Using VSIDE with templates

You can find directions to the VSIDE download in the chapter “Additional documentation”.

VSIDE has a VS1010 template for a typical Hello World program.

1) To use it, launch VSIDE and navigate to File → New → Project/Solution.

2) This launches a “New Project” window with two tabs, “Solution” and “Project”.

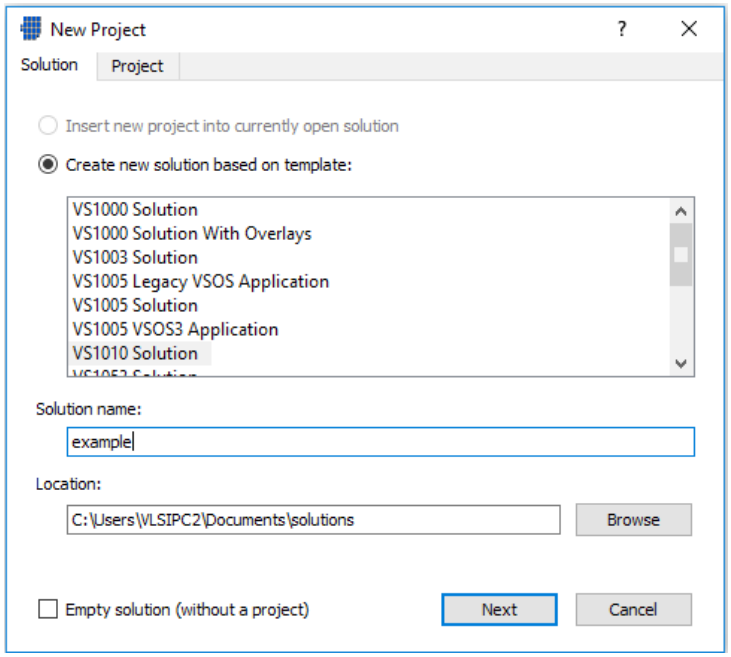

Illustration 6: VSIDE New Project window - Solution tab

3) Make sure you are in the “Solution tab” (Illustration 6) and that “Create new solution based on template:” is selected

4) From the list of solutions, find “VS1010 Solution”, choose an approriate name and save location, and make sure that the checkbox “Empty solution (without a project)” is **not** checked

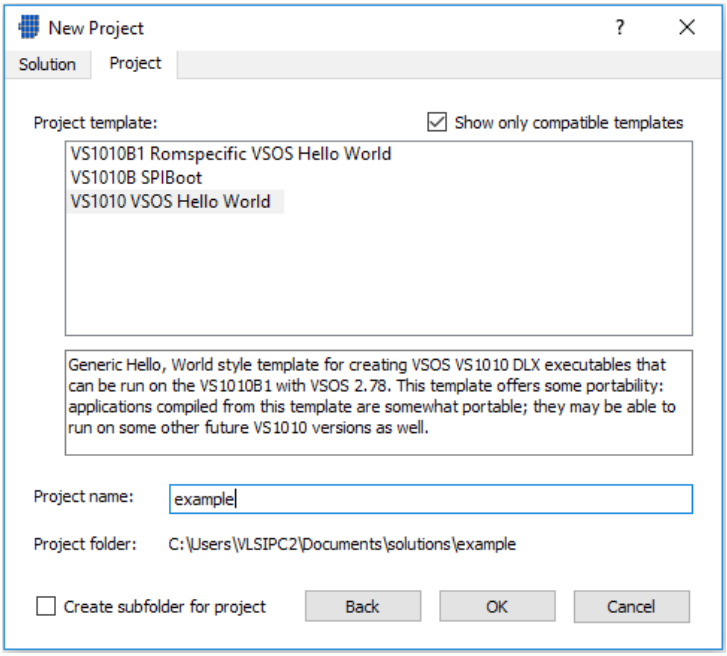

Illustration 7: VSIDE New Project window - Project tab

5) Once satisfied, click next. This will move you to the “Project” tab (Illustration 7)

6) From the list of templates, choose “VS1010 VSOS Hello World”. If there are templates for boards other than VS1010, you can check the box “Show only compatible templates”.

   Sidenote: it's possible to give the project a name that differs from the solution's. This is somewhat useful if you wish to test multiple programs with the same name, e.g. “vsplayer.dlx”. Multiple solutions of the same name are obviously not allowed in the same “solutions” folder (if you use the default location), but projects can share names, provided they are each inside their own solutions. Alternatively, you can simply rename programs manually when needed.

7) Once you click “OK”, the necessary files will be created. You can see the typical VSIDE interface in Illustration 8. Note: the Hello World template has more includes than the screenshot, as most of them are unnecessary to it, however, every VS1010 program should begin by including vo_stdio.h, even if you can build the program without it.

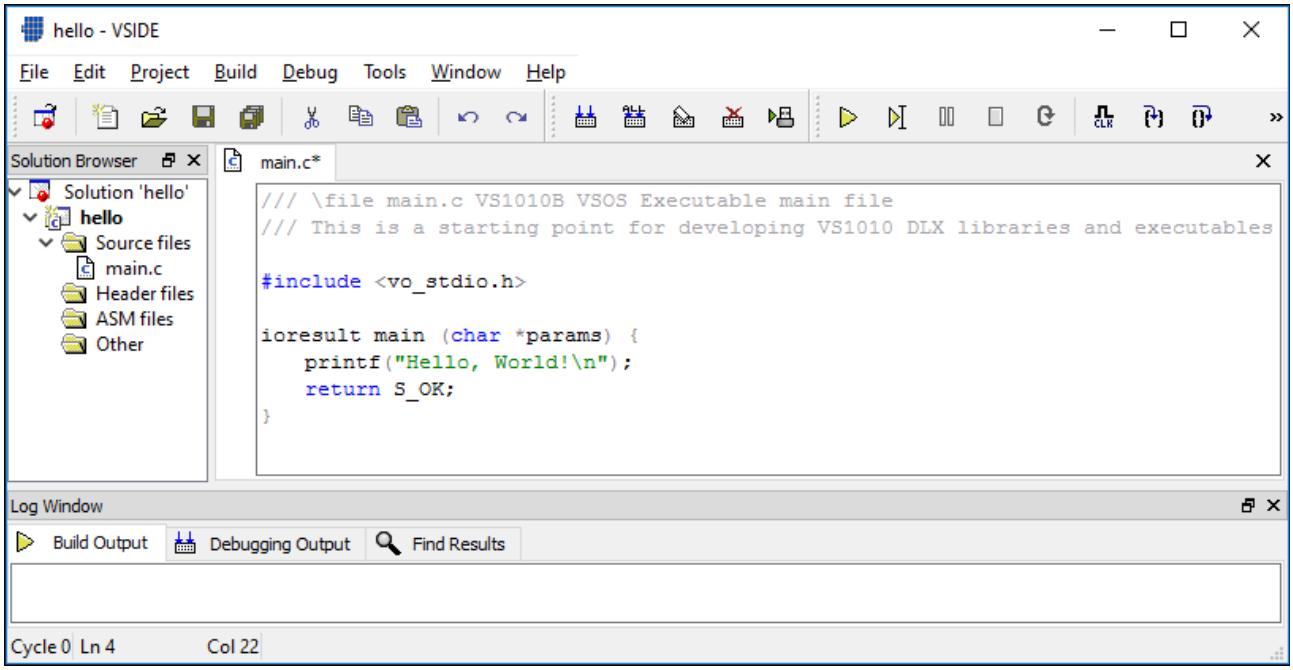

Illustration 8: VSIDE graphical user interface

8) You can now either build the solution from Build → Build Solution, and then move the .dlx to the microSD card by hand, or alternatively first set the target drive by clicking the icon in the toolbar, and then build. The target drive icon is the rightmost icon on the build toolbar (Illustration 9). If the build toolbar is not visible, right-click somewhere on the toolbar area and select it from the list. The icon for the target drive may also have a red X on it (Illustration 10), instead of the green play symbol. This changes depending on whether or not the target drive is set to something.

   Note: If you set the SD card as the target drive, the .dlx file will automatically go to the SYS folder, not the root of the SD. With most programs this is quite useful, but some need to be in the root for them to run automatically, mainly boot.dlx and vsplayer.dlx

Illustration 9: Build toolbar

Illustration 10: Target drive

9) To execute the binary, drop to shell and write the executable name. If the executable file is in the folder S:SYS/, then typing the base name of the file suffices. Otherwise, type the full path of the file.

   Example:

```
VS1010>hello
Hello, World!
VS1010>s:sys/hello.dlx
Hello, World!
VS1010>
```

7 Executing your program

In addition to the previously described UART shell, there are many other ways to execute your code from the SD card.

7.1 CONFIG.TXT

The supplied SD card should come with a text file named “config.txt” which prints some short welcome messages and instructions. The programs entered to config.txt are executed before the ROM player is started.

To execute programs placed in the SYS folder, you can use the same logic as with the UART shell. Either use the name of the function, or write the full path. You can run multiple programs after one another, as long as each of them are on their own lines.

You can also run a different set of programs for different runlevels, by typing the runlevel in square brackets before the programs.

For example:

```
[0]
echo Hello
echo Runlevel is 0
[3]
echo Runlevel is 3
[0]
echo Hello again
```

will print

```
Hello
Runlevel is 0
Hello again
```

for runlevel 0, and

```
Runlevel is 3
```

for runlevel 3.

7.2 \BOOT.DLX

At boot time, after initializing the SD card, if a file named “BOOT.DLX” is found in the root of the SD card, it's automatically executed. This can be used to execute custom programs immidiately after booting. It's also a convenient way to load patch code to the VS1010 before continuing with the normal operation.

In the VS1010B ROM code, there is an off by one error in reporting the number of available sectors on the SD card to the PC in USB mass storage mode. As a consequence, some operating systems fail to show the SD card contents correctly via USB. The VS1010B Developer Board's SD card has a BOOT.DLX file, which fixes the problem, allowing correct USB SD card reader operation.

You can also replace BOOT.DLX with your custom program, if needed. However, in most cases using config.txt should be enough.

7.3 \VSPLAYER.DLX

If there is a program is named “vsplayer.dlx” in the root of the SD card, it will be called by the ROM default music player when it starts or restarts.

The supplied SD card doesn't necessarily have a program with this name, but you can add one regardless.

7.4 Magic sector

This is a way to hide executable code into the SD card so that it is not seen or copied by generic operating systems normal file copy dialogues. For more information contact VLSI support.

8 Included system programs

When you connect to the board using UART, you can run commands from the command line. The commands are stored in the “System Disk”, which is a logical drive letter that can point to different physical storage devices. In the VS1010B developer board, it points to the SD card.

The SYS folder of the system disk (SD card) contains the following programs:

8.1 DIR.DLX

The DIR program lists files.

Example:

```
VS1010>dir
File list of SD/SD Card S:*
     1: VS1010SD
     2: DOC
     3: RomAppsSrc
     4: SYS
     5: A Little Bit (sample).mp3
     6: Angels Crying (sample).mp3
     7: BOOT.DLX
     8: CONFIG.TXT
     9: I wont let the Sun go Down (sample).mp3
    10: Orinoco Flow (sample).mp3
 10 file(s) found.
```

Example:

```
VS1010>dir s:doc/
File list of SD/SD Card s:doc/*
     1: .
     2: ..
     3: Demo10-2017-03-28-16-48-sourcecode.zip
     4: README.TXT
     5: vs1010devboardquickstart.pdf
 5 file(s) found.
```

8.2 ECHO.DLX

ECHO prints the parameter to the command line. This is used e.g. in the CONFIG.TXT to print out a greeting to the user.

Example:

```
VS1010>echo Hello, World!
Hello, World!
```

8.3 TYPE.DLX

TYPE can be used to print ASCII files to the console.

Example:

```
VS1010>type s:config.txt
[0]
echo
echo Welcome to VS1010!
echo Press [Return] to drop to the shell.
echo
# add your own programs here:
echo Exiting to the ROM player.
```

8.4 POKE.DLX

POKE can be used to manipulate memory and peripheral registers.

Example 1:

```
VS1010>poke x:1,2
 X:0x0001: 0x0000->0x0002
```

Example 2:

```
VS1010>poke x:0,1
 X:0x0000: 0x0000->0x0001
X[0]Corrupted
```

In example 2, a guard interrupt notifies that address 0 of the X memory is not zero and is thus corrupted. VSOS needs the first few words of the X memory to be cleared so that any method calls of uninitialized objects have a good chance to be properly trapped by a zero pointer call trap.

8.5 PEEK.DLX

PEEK can be used to read contents of memory locations and peripheral registers.

Example:

```
VS1010>peek 0xbffe
0xbffe X=0x6b36 Y=cbe6

VS1010>peek 0xbfff
0xbfff X=0x5e27 Y=61de
```

The above example reads the X and Y ROM checksums of VS1010B

8.6 LIBLIST.DLX

LIBLIST prints out a list of loaded libraries and the memory ranges that they occupy in instruction, X data and Y data memory spaces.

Example:

```
VS1010>liblist
0: 0xf00:ROMFILE I[0x80..0x169] X[0xf00..0xf4f] Y[0xf00..0xeff]
1: 0xf50:LIBLIST.DLX I[0x16a..0x1b7] X[0xf50..0xf91] Y[0xf00..0xeff]
```

The listing shows that two libraries are loaded. The first one is identified only as “ROMFILE”, meaning it's something loaded from the ROM (it's an MP3 player routine that's automatically executed at startup). The second library is the executable program LIBLIST itself.

8.7 REBOOT.DLX

REBOOT reboots the chip from the command line

8.8 DEVICES.DLX

DEVICES prints out a list of currently loaded device drivers

Example:

```
VS1010>devices
D: SD/SD Card
I: STREAM
S: SD/SD Card
X: ROMFILE
Y: ROMFILE
```

In the above example, D: and S: disks point to the SD card (running in SD mode). I: points to a serial stream receiver device. X and Y point to contents in ROM.

8.9 DEMO10.DLX

This is a very simple button controlled MP3 play file demo. The source code of this program is included below in this document.

8.10 HELLO.DLX

A Hello, World program

8.11 HELLO.MP3

A test MP3 file which contains the OS code author saying the word “hello”. To play it, give the PLAY command (it's an internal command in VS1010 ROM):

```
VS1010>play s:sys/hello.mp3
HELLO.MP3
```

8.12 ROMCRC.DLX

A program that prints out ROM checksums. You can use this to check which version of the VS1010 device you are using. This may be relevant if you choose to optimize your code size by linking directly to symbols in ROM. (If possible, you should use only the VSOS standard interface in your programs – this generates more portable code – instead of accessing any ROM symbols directly.)

Example:

```
VS1010>romcrc
Check ROM CRCs

Calculated IROM: ee14d38f  , ROM @ dfff       contains: ee14d38f  , ok
Calculated XROM: 6b36, 5e27, ROM @ bffe, bfff contains: 6b36, 5e27, ok
Calculated YROM: cbe6, 61de, ROM @ bffe, bfff contains: cbe6, 61de, ok
```

9 Custom player example

Below is an example of a custom player program that plays (about) 3 seconds of music from the start of a file that is selected by developer board buttons. Run it from the command line by typing "DEMO10" or add line DEMO10 to CONFIG.TXT

```
/// \file main.c VS1010B VSOS Executable main file
/// This is a starting point for developing VS1010 DLX libraries and executables

#include <vo_stdio.h>
#include <volink.h>     // Linker directives like DLLENTRY
#include <apploader.h>  // RunLibraryFunction etc
#include <vs1010bRom.h>
#include <vo_gpio.h>
#include <vs1010b.h>
#include <playerinfo.h>
#include <string.h>
#include <protocol.h>
#include <spitv.h>
#include <lcd.h>

SETHANDLER(97, MyPlayerCallback)

s_int32 samplesToPlay;

void MyPlayerCallback(AUDIO_DECODER *auDec, u_int16 samples) {
       // Insert here any user interface code that you want to run
       // while a song is playing, for example to stop the song
       // before it ends. An example is shown below. For sample
       // rate 44100, it plays 3 seconds and then stops the decoder.
       // Note that samplesToPlay is initialized in PlayNamedFile();

       samplesToPlay -= samples;
       if (samplesToPlay <= 0) {
              player.auDec.cs.cancel = 1; //Stop playing current file.
       }
}


u_int16 PlayNamedFile(register const char *filename) {
       player.currentFile = 0;
       strncpy(player.fileSpec,filename,126);
       samplesToPlay = 44100 * 3; //real sample rate is unknown before decoding starts.
       PlayerPlayFile();
}


ioresult main (char *params) {
       printf("VS1010 Buttons Demo\n");
       printf("Press [S1] or [S3] on the devboard.\n");

       while(1) {
              if (!GpioReadPin(0x1c)) {
                     printf("Playing file A...\n");
                     PlayNamedFile("S:A*.MP3");
                     printf("Done.\n");
              }

              if (!GpioReadPin(0x1d)) {
                     printf("Playing file O...\n");
                     PlayNamedFile("S:O*.MP3");
                     printf("Done.\n");
              }
       }

       return S_OK;
}
```

10 Additional documentation

Various additional documentation can be found at http://www.vlsi.fi/:

The VS1010 Developer board schematic: Support → Evaluation Boards → VS1010 Developer Board → Additional Information → Schematic

VS1010 and VS23S010 documentation: Products → (device model) →Additional Information

VSIDE: Support → Software → VSIDE

USB-UART cable drivers: link to a forum topic with the download can be found at the end of the previously described VSIDE page

11 Version history

Version 0.1, 2017-03-28

Version 0.2, 2017-05-18, Added info of programs included in the SD card.

Version 0.3, 2017-05-19, Added illustration.

Version 0.4, 2017-05-19, Added more illustrations and rewrote some text.

Version 0.5, 2017-05-30, Unified formatting. Rewrote, restructured, and added some text. Modified and added some illustrations.

Version 0.6, 2017-06-05, Minor changes to the text, one added illustration

12 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