

## VS1000 Audio Module

### Features

- Plays audio from microSD card or SPI Flash
- Uses powerful Ogg Vorbis audio format
- Firmware and content can be customized
- Controlled via UART or IO pins
- Very simple to use, UART is enough to send commands
- High-performance CD quality analog out
- Supports low bit rates and various sample rates
- Operates from a single supply (VCC = 3.6...6.0V) (Note: other pins are not 5V-tolerant. Do not drive other pins over DVDD!)
- Line output
- Headphone output
- SPI FLASH for code and/or data storage
- MicroSD connector for data storage
- Most internal nodes are routed to pin-headers for easy access and customization

### Applications

- Elevators
- Ticket machines
- Audio user's manuals
- Vending machines
- Car accessories
- Alarm systems
- PC accessories
- Speaking hi-tech toys
- Development board for VS1000 chip

### Description

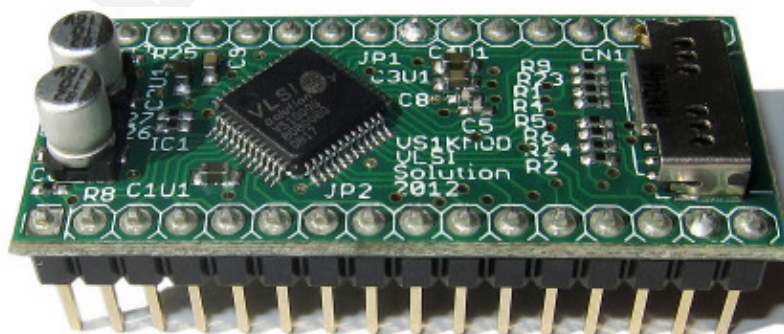
The VS1000 Module is a small, low-cost, high performance, easy to use audio player that is controlled from the IO-pins or UART. It can be used as a "plug-in" audio board in electronic systems or as a standalone small audio player.

The product is supported by design services, audio content preparation and pre-programming. It is also fully configurable by the user by using free Integrated Development Tools (VSIIDE) for the VS1000 IC. The use of the module does not require any advanced information from the user.

Pin headers of the PCB are compatible with DIL32 footprint 2.54 pitch 15.24 mm wide (100 mils pitch, 600 mils wide). This makes it possible to use standard DIL32 ZIF sockets or solder it by using DIL32 footprint.

The left row of the PCB has digital connections for external buttons, LEDs or other circuitry. The right row has power connections, analog connections, a serial port and USB. These can be used to interface the board with a PC and VSIIDE or a host microcontroller.

The VS1000 Module operates from a single power supply. The board has 16 Mbits of on-board FLASH and a microSD card connector for playing license free Ogg Vorbis audio files. The module can boot from the on-board FLASH memory.



## Contents

<b>VS1000 Audio Module</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>3</b>
<b>1 Disclaimer</b>	<b>4</b>
<b>2 Definitions</b>	<b>4</b>
<b>3 Placement and Pinout</b>	<b>5</b>
<b>4 Dimensions</b>	<b>6</b>
<b>5 Characteristics &amp; Specifications</b>	<b>7</b>
5.1 Absolute Maximum Ratings . . . . .	7
5.2 Recommended Operating Conditions . . . . .	7
5.3 Analog Characteristics of Audio Outputs . . . . .	8
5.4 Analog Characteristics of Regulators . . . . .	9
5.5 Analog Characteristics of VCC voltage monitor . . . . .	9
5.6 Analog Characteristics of CVDD voltage monitor . . . . .	10
5.7 Analog Characteristics of USB . . . . .	10
5.8 Power Consumption . . . . .	10
5.9 Digital Characteristics . . . . .	11
<b>6 PCB and Component Layout</b>	<b>12</b>
<b>7 PCB Schematics</b>	<b>13</b>
7.1 Notes about the Schematic . . . . .	15
<b>8 Default Firmware (version 0.70)</b>	<b>16</b>
<b>9 Application Examples</b>	<b>22</b>
9.1 Headphone Connection . . . . .	22
9.2 Line Out Connection . . . . .	23
9.3 USB Controlled and Powered Device with Headphone Output . . . . .	24
9.4 UART Control . . . . .	25
9.5 GPIO-Controlled Player . . . . .	26
9.6 USB controlled LEDs and Relays . . . . .	27
<b>10 Document Version Changes</b>	<b>28</b>
<b>11 Contact Information</b>	<b>29</b>

## List of Figures

1	Pinout. . . . .	5
2	Top layer PCB layout . . . . .	6
3	Top layer PCB layout . . . . .	6
4	Top layer and silkscreen of the PCB (Top view) . . . . .	12
5	Bottom layer and silkscreen of the PCB (Top view) . . . . .	12
6	Schematic 1.4 of the VS1000 Module. . . . .	13
7	Schematic 1.61 of the VS1000 Module. . . . .	14
8	Headphone Connection. . . . .	22
9	Line Out Connection. . . . .	23
10	USB Controlled and Powered Device with Headphone Output. . . . .	24
11	RS232 Control. . . . .	25
12	Button Controlled Player (idle high). . . . .	26
13	USB Controlled LEDs and Relays. . . . .	27

## 1 Disclaimer

All properties and figures are subject to change.

## 2 Definitions

**B** Byte, 8 bits.

**b** Bit.

**Ki** “Kibi” =  $2^{10}$  = 1024 (IEC 60027-2).

**Mi** “Mebi” =  $2^{20}$  = 1048576 (IEC 60027-2).

**VS\_DSP** VLSI Solution’s DSP core.

**W** Word. In VS\_DSP, instruction words are 32-bit and data words are 16-bit wide.

### 3 Placement and Pinout

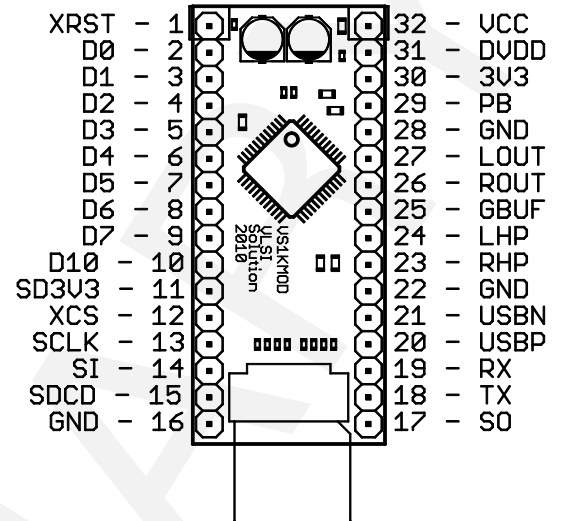


Figure 1: Pinout.

Pin	Name	Description	Pin	Name	Description
1	XRESET	Active Low reset for VS1000	32	VCC	Input Voltage. Power supply for the board. 3.6V - 6.0V allowed
2	D0	General purpose I/O pin	31	DVDD	Output of DVDD regulator, for external circuitry
3	D1	General purpose I/O pin	30	3V3	Output of 3V3 regulator, for external circuitry
4	D2	General purpose I/O pin	29	PB	Powerbutton. For turning the VS1000 on and off
5	D3	General purpose I/O pin	28	GND	Ground. Connected to ground plane
6	D4	General purpose I/O pin	27	LOUT	Left channel line out (AC-coupled)
7	D5	General purpose I/O pin	26	ROUT	Right channel line out (AC-coupled)
8	D6	General purpose I/O pin, 100K pulldown	25	GBUF	Headphone common output (note DC-bias)
9	D7	General purpose I/O pin, 100K pullup	24	LHP	Headphone left channel output (note DC-bias)
10	D10	General purpose I/O pin	23	RHP	Headphone right channel output (note DC-bias)
11	SD3V3	Controls the 3.3V regulator on board, 100K pulldown	22	GND	Ground. Connected to ground plane
12	XCS	SPI bus Chip Select, 100K pullup	21	USBN	USB negative signal
13	SCLK	SPI bus clock input	20	USBP	SUB positive signal
14	SI	SPI bus data input	19	RX	Serial port receive, 100K pullup
15	SDCD	SD card detected	18	TX	Serial port transmit
16	GND	Ground. Connected to ground plane	17	SO	SPI bus data output

**Note: The maximum voltage for all IO pins is 3.6V! Only VCC can be 3.6V - 6.0V.**

## 4 Dimensions

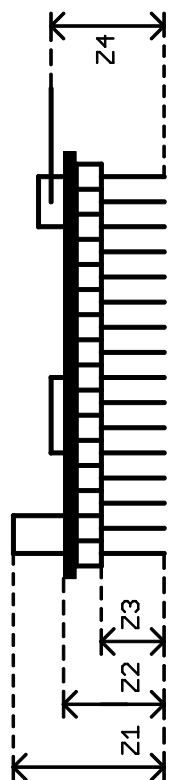


Figure 2: Top layer PCB layout

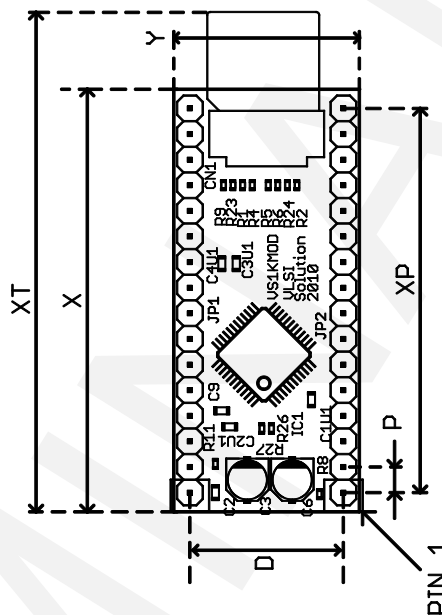


Figure 3: Top layer PCB layout

Parameter	Symbol	Min	Typ	Max	Unit
Width of PCB	Y		18.40		mm
Length of PCB	X		41.91		mm
Total length (includes USB connector)	XT		40.65		mm
Length of pin connector	XP		38.10		mm
Maximum height of module	Z1		16.60		mm
Minimum height of module	Z2		11.60		mm
Pitch of pins	P		2.54		mm
Pin height	Z3		6.80		mm
Distance from lower pin row to upper pin row	D		15.24		mm
Distance to microSD card centerline	Z4		14.40		mm
Pin type and size	PT		0.65 square		mm

### 5 Characteristics & Specifications

#### 5.1 Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Supply voltage	VCC	-0.3	6.0	V
Voltage at Any Digital Input		-0.3	IOVDD+0.3 <sup>1</sup>	V
Total Injected Current on Pins			±200 <sup>2</sup>	mA
Operating Temperature		-40	+85	°C
Storage Temperature		-65	+150	°C

<sup>1</sup> IOVDD is output of the programmable regulator of the VS1000 chip. Must not exceed 3.6 V

<sup>2</sup> Latch-up limit

#### 5.2 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
Operating temperature		-40		+85	°C
Ground <sup>1</sup>	GND		0.0		V
Supply voltage <sup>2</sup>	VCC	3.6		6.0	V
Analog positive supply <sup>3</sup>	AVDD	2.75	2.8	3.6	V
Digital positive supply <sup>3</sup>	CVDD	2.2	2.3	2.65	V
I/O positive supply <sup>3</sup>	IOVDD	1.8	2.8	3.6	V
Input clock frequency <sup>4</sup>	XTALI	12	12 <sup>5</sup>	13	MHz
Internal clock frequency, USB active	CLKU	48		48	MHz
Internal clock frequency, USB inactive	CLKI	12		48	MHz
Master clock duty cycle		40	50	60	%

<sup>1</sup> Do not float ground for latch-up immunity.

<sup>2</sup> At least 4.0 V is required for compliant USB level.

<sup>3</sup> Regulator output of the VS1000 device. Can be controlled by the firmware.

<sup>4</sup> The maximum sample rate that can be played with correct speed is XTALI/256. With 12 MHz XTALI sample rates over 46875 Hz are played at 46875 Hz.

<sup>5</sup> To be able to use USB, XTALI must be 12 MHz.

### 5.3 Analog Characteristics of Audio Outputs

Unless otherwise noted: AVDD=3.3V, CVDD=2.4V, IOVDD=2.8V, TA=-40..+85°C, XTALI=12 MHz, Internal Clock Multiplier 3.0×. DAC tested with full-scale output sinewave, measurement bandwidth 20..20000 Hz, analog output load: LHP to GBUF 30 Ω, RHP to GBUF 30 Ω, LOUT: 10k Ω, LOUT: 10k Ω.

Parameter	Symbol	Min	Typ	Max	Unit
DAC Resolution			18		bits
Dynamic range (DAC unmuted, A-weighted, min gain)	IDR		96		dB
S/N ratio (full scale signal, no load <sup>1</sup> )	SNR		92		dB
S/N ratio (full scale signal, 30 ohm load <sup>2</sup> )	SNRL	75	90		dB
Total harmonic distortion, max level, no load <sup>1</sup>	THD		0.01		%
Total harmonic distortion, max level, 30 ohm load <sup>2</sup>	THDL		0.1	0.3	%
Crosstalk (LOUT/ROUT to ROUT/LOUT), no load <sup>1</sup>	XTALK0		75		dB
Crosstalk (LHP/RHP to RHP/LHP), 30 ohm load, without GBUF <sup>3</sup>	XTALK1		75		dB
Crosstalk (LHP/RHP to RHP/LHP), 30 ohm load, with GBUF	XTALK2		54		dB
Gain mismatch (LOUT/ROUT to ROUT/LOUT)	GERR	-0.5		0.5	dB
Frequency response	AERR	-0.05		0.05	dB
Full scale output voltage	LEVEL	450	530	600	mVrms
Deviation from linear phase	PH		0	5	°
Analog output load resistance	AOLR		30 <sup>4</sup>		Ω
Analog output load capacitance	AOLC			100	pF
DC level (GBUF, LHP, RHP)		1.1		1.3	V
GBUF disconnect current (short-circuit protection)			130	200	mA

<sup>1</sup> Characteristics with no load are measured from LOUT/ROUT outputs towards GND such that LHP/RHP outputs are not loaded.

<sup>2</sup> Characteristics with 30 Ω load are measured from LHP/RHP outputs towards GBUF such that LOUT/ROUT outputs are not loaded.

<sup>3</sup> Loaded from LHP/RHP pin to analog ground via 100 μF capacitors.

<sup>4</sup> AOLR may be lower than *Typical*, but distortion performance may be compromised. Also, there is a maximum current that the internal regulators can provide.



### 5.4 Analog Characteristics of Regulators

Parameter	Symbol	Min	Typ	Max	Unit
<b>IOVDD</b>					
Recommended voltage setting range		1.7		3.6	V
Voltage setting step size		50	60	70	mV
Default setting, reset mode <sup>1</sup>			1.8		V
Default setting, active mode <sup>2</sup>			1.8/3.6 <sup>3</sup>		V
Load regulation			4.0		mV/mA
Line regulation from VCC			2.0		mV/V
Continuous current			30 <sup>4</sup>	40	mA
<b>CVDD</b>					
Recommended voltage setting range		1.8		2.6	V
Voltage setting step size		35	48	55	mV
Default setting, reset mode <sup>1</sup>			1.8		V
Default setting, active mode <sup>2</sup>			2.2		V
Continuous current			30 <sup>4</sup>	35	mA
Load regulation			2.0		mV/mA
Line regulation from VCC			2.0		mV/V
<b>AVDD</b>					
Recommended voltage setting range		2.6		3.6	V
Voltage setting step size		35	46	55	mV
Default setting, reset mode <sup>1</sup>			2.5		V
Default setting, active mode <sup>2</sup>			2.7		V
Continuous current			30 <sup>4</sup>	70	mA
Load regulation			1.5		mV/mA
Line regulation from VCC			2.0		mV/V

<sup>1</sup> Device enters reset mode when XRESET pin is pulled low.

<sup>2</sup> Device enters active mode when XRESET pin is pulled high after reset mode. Regulator settings can be modified by custom firmware (in on-board FLASH).

<sup>3</sup> Depends on GPIO0\_7 pin status in boot.

<sup>4</sup> Device is tested with a 30 mA load.

### 5.5 Analog Characteristics of VCC voltage monitor

Parameter	Symbol	Min	Typ	Max	Unit
Trigger voltage	AMON		1.07×AVDD		V
Hysteresis			50		mV

VCC voltage monitor causes interrupt when it is activated. The purpose is to shut down the system by the firmware before voltage level causes functionality error.

### 5.6 Analog Characteristics of CVDD voltage monitor

Parameter	Symbol	Min	Typ	Max	Unit
Trigger voltage	CMON	1.40	1.53		V
Hysteresis			2		mV

CVDD voltage monitor causes internal hardware reset of the system.

### 5.7 Analog Characteristics of USB

Parameter	Min	Typ	Max	Unit
Drive low level, 2.32 mA load	0.065		0.102	V
Drive low level, $6.1 \times AVDD$ mA load	$0.171 \times AVDD$		$0.270 \times AVDD$	V
Drive low level, $10.71 \times AVDD$ mA load	$0.300 \times AVDD$		AVDD	V
Drive high level, -2.32 mA load	AVDD-0.165		AVDD-0.065	V
Drive high level, $-6.1 \times AVDD$ mA load	$0.650 \times AVDD$		$0.829 \times AVDD$	V
Drive high level, $-10.71 \times AVDD$ mA load	0		$0.700 \times AVDD$	V
USBP level, with 15 k $\Omega$ pull-down	2.7		$0.943 \times AVDD$	V
High-Level input voltage (single-ended)	$0.7 \times AVDD$		AVDD+0.3	V
Low-Level input voltage (single-ended)	-0.2		$0.3 \times AVDD$	V
Differential input common voltage, AVDD $\geq$ 3.3V	0.8		2.5	V
Differential input signal level, AVDD $\geq$ 3.3V	200			mV
Input leakage current	-2.0		2.0	$\mu$ A

### 5.8 Power Consumption

Parameter	Min	Typ	Max	Unit
Current Consumption of Reset mode (XRESET=0V) @ 25 °C		24	48	$\mu$ A
<b>UART player application</b> VCC=6.0V IOVDD=3.3V AVDD=2.8V CVDD=2.5V				
Total Power, play mode, LHP and RHP with 30 $\Omega$ load to GBUF		120		mW
Total Power, play mode, LOOUT and ROUT with 10k $\Omega$ load		120		mW
<b>UART player application</b> VCC=4.0V IOVDD=3.3V AVDD=2.8V CVDD=2.5V				
Total Power, play mode, LHP and RHP with 30 $\Omega$ load to GBUF		120		mW
Total Power, play mode, LOOUT and ROUT with 10k $\Omega$ load		120		mW
<b>UART player application</b> VCC=6.0V, IOVDD=2.7V AVDD=2.6V CVDD=2.2V				
Total Power, pause mode		10		mW
Total Power, play mode, LHP and RHP with 30 $\Omega$ load to GBUF		80		mW
Total Power, play mode, LOOUT and ROUT with 10k $\Omega$ load		80		mW
<b>UART player application</b> VCC=4.0V, IOVDD=2.7V AVDD=2.6V CVDD=2.2V				
Total Power, pause mode		10		mW
Total Power, play mode, LHP and RHP with 30 $\Omega$ load to GBUF		80		mW
Total Power, play mode, LOOUT and ROUT with 10k $\Omega$ load		80		mW

### 5.9 Digital Characteristics

Parameter	Min	Typ	Max	Unit
High-Level Input Voltage	$0.7 \times \text{IOVDD}$		$\text{IOVDD} + 0.3$	V
Low-Level Input Voltage	-0.2		$0.3 \times \text{IOVDD}$	V
High-Level Output Voltage, -1.0 mA load <sup>1</sup>	$0.7 \times \text{IOVDD}$			V
Low-Level Output Voltage, 1.0 mA load <sup>1</sup>			$0.3 \times \text{IOVDD}$	V
XTALO high-level output voltage, -0.1 mA load	$0.7 \times \text{IOVDD}$			V
XTALO low-level output voltage, 0.1 mA load			$0.3 \times \text{IOVDD}$	V
Input leakage current	-1.0		1.0	$\mu\text{A}$
Rise time of all output pins, load = 30 pF <sup>1</sup>			50	ns

<sup>1</sup> Pins GPIO0\_[14:0], GPIO1\_[5:0].

## 6 PCB and Component Layout

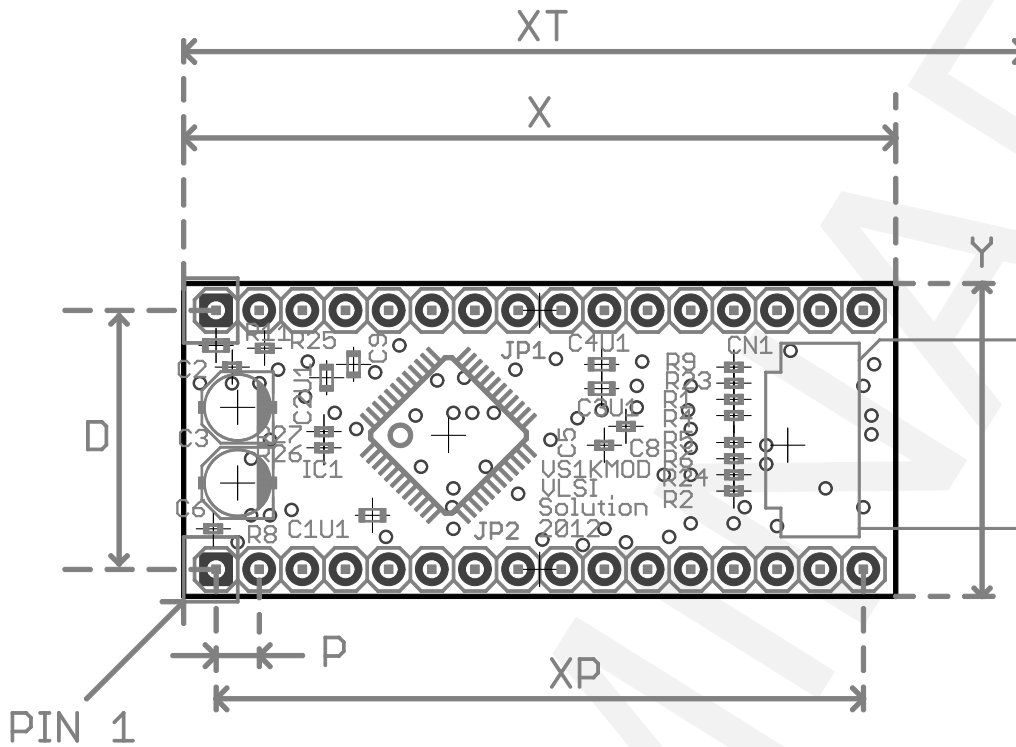


Figure 4: Top layer and silkscreen of the PCB (Top view)

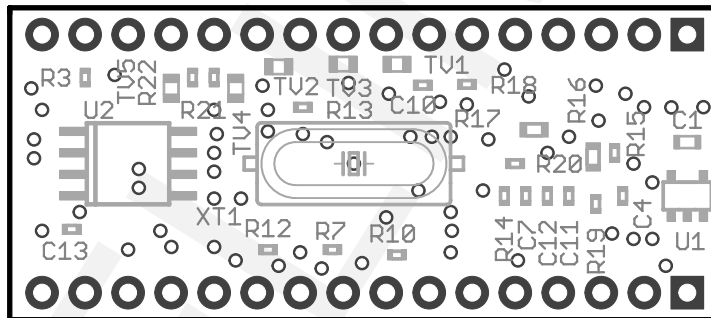


Figure 5: Bottom layer and silkscreen of the PCB (Top view)

7 PCB Schematics

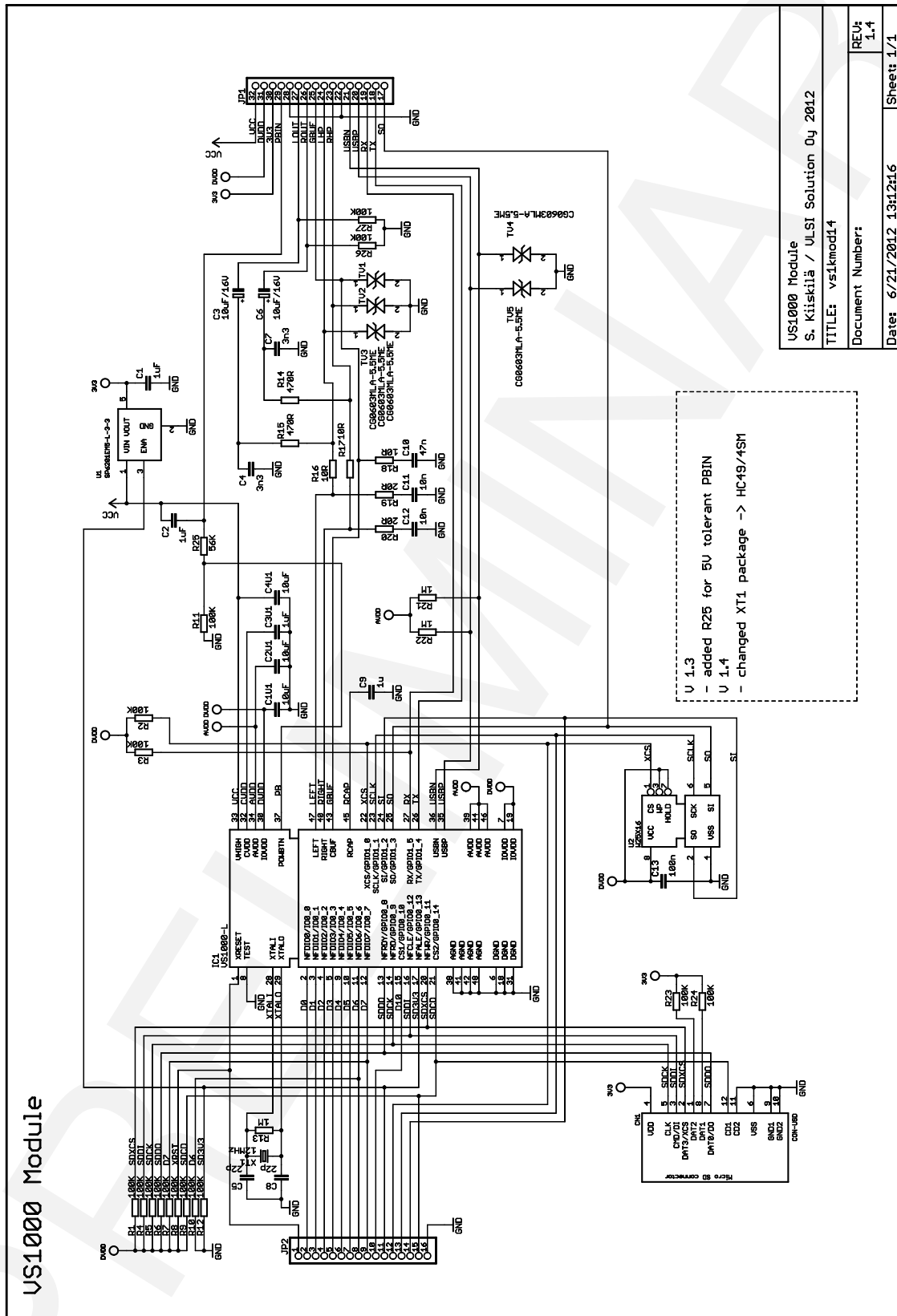


Figure 6: Schematic 1.4 of the VS1000 Module.

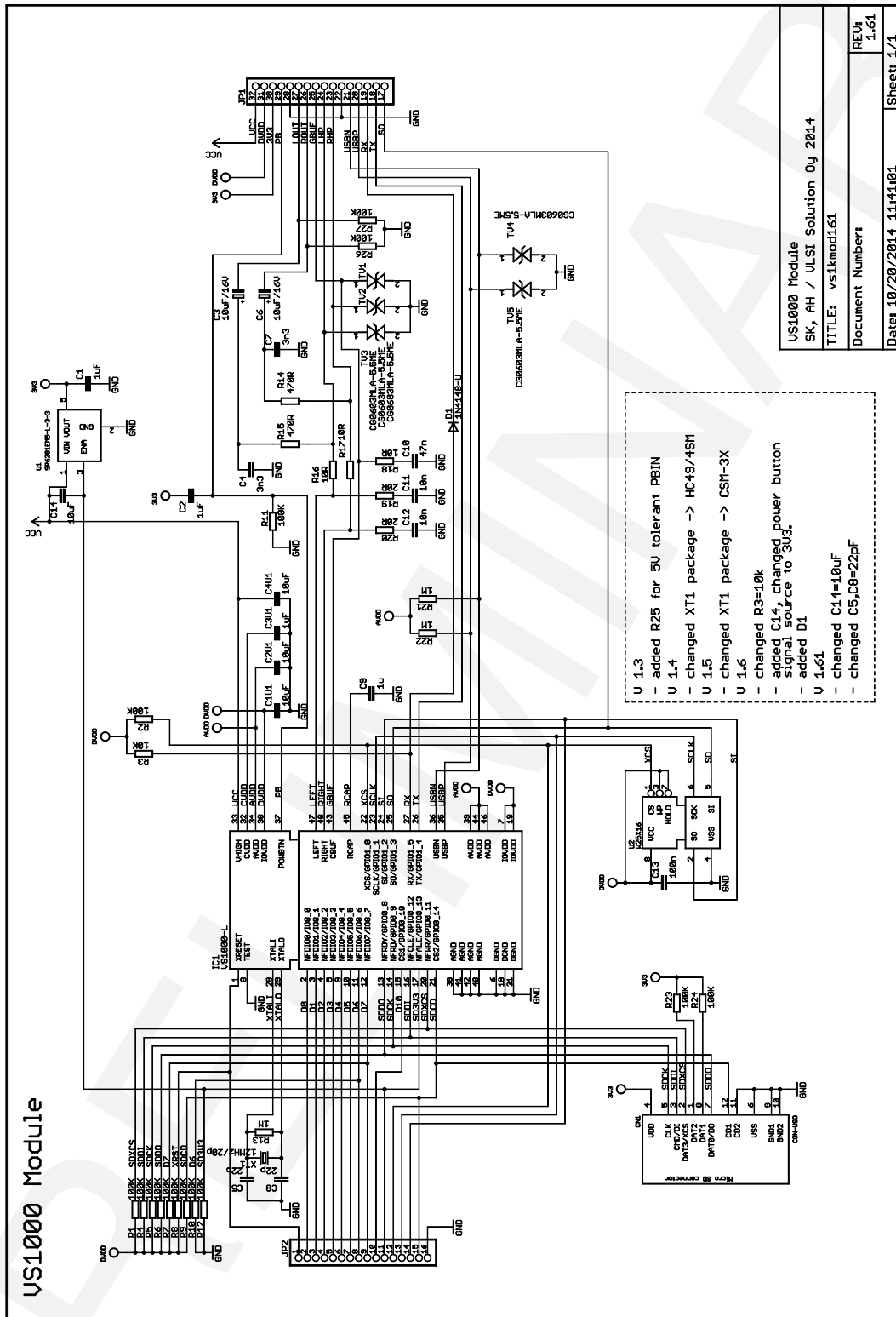


Figure 7: Schematic 1.61 of the VS1000 Module.

## 7.1 Notes about the Schematic

- PWRBTN has resistor divider to prevent too high voltages on the input pin.
- Line outputs have a first order RC low-pass filter that removes some of the DA converter quantization noise. Useful when connecting to a limited bandwidth amplifier.
- Line outputs are AC-coupled, so they can be connected directly to an amplifier.
- Headphone outputs (including common buffer) are DC-biased to 1.2V. Do not connect the common buffer output to ground! Do not connect Left/Right headphone outputs directly to an amplifier!
- Input power supply is connected to pin 1 of JP1.

### v1.61

- For 1.61 the powerbutton pulse is generated using the 3.3 V regulator. This way the pulse should never exceed specification and be high enough to start the module.
- The RX pin now uses a diode, and is thus 5 V-tolerant.

## 8 Default Firmware (version 0.70)

The VS1000 Audio Module allows several uses in addition to audio playback. For normal audio use, the content can reside in either the SPI FLASH or external SD card. The default firmware implements UART-controller player, which plays files from the SPI FLASH or  $\mu$ SD.

UART speed 115200 bps is used. The module boots up by default in *continuous* playing mode, which plays all available files sequentially and continues until powered off. If SD/MMC card insertion is detected, files are played from SD/MMC instead.

When power is applied and the firmware starts, a version string is printed ("v0.7uart" or "v0.70gpio"). If you don't see a versions string, the firmware is older than 0.70, or it is compiled neither with UART nor GPIO control. If a valid FAT filesystem is detected, you will get the string "fat", a 4-byte FAT size (most significant byte first), then a newline character (0x0a). For example "fat", 0x00, 0x00, 0x0f, 0xc0, 0x0a. If there is not a valid FAT disk, you will get "nofat" and a newline (0x0a).

If there is a valid disk, the string "files" is sent, followed by a two-byte value containing the number of playable files (for example 0x00, 0x04 for 4 files), and a newline (0x0a).

When a file has been located and starts playing, you will get the string "play" and a two-byte file number, then a 8.3-character filename (for example "AUDIO01 OGG" and a newline (0x0a). After a file is finished playing (or interrupted), you will get "done" and a newline (0x0a). If the player is in the *continuous* playing mode, a new file will be automatically selected and played.

The *continuous* playing mode understands the following single-byte commands:

- f - switch to *file* play mode
- c - switch to *continuous* play mode
- C - cancel play, return to play loop, responds with c
- + - volume up, responds with two-byte current volume level
- - - volume down, responds with two-byte current volume level
- = - pause on, responds with =
- > - play (normal speed), responds with >
- >> (0xbb) - faster play, responds with the new play speed
- n - next song, responds with n
- p - previous song, responds with p
- ? - play info, returns play seconds (4 bytes, high byte first), and one-byte file left indicator (255 to 0).

In *file* play mode files are not played automatically. To change into *file* play mode, send f, then you can send c to cancel playback. (The play mode that will be active after power-on can be configured, see the VSIDE solution.)



The *file* play mode commands are single-line commands ending with the newline character (0x0a), which is marked with in the text below as `\n` . The following commands are currently implemented:

- `OFF\n` - powers down
- `c\n` - switch to *continuous* play mode
- `L\n` - list files
- `PFILENAMEOGG\n` - play by name (capital P), a 8.3-character uppercase name without the "." .
- `pnumber\n` - play file by number (small-case p)

Continuous play mode commands are available during the playing of the song.

If SD card is inserted during SPI FLASH playback, "SD" and newline (0x0a) are sent, and the SD/MMC player program is loaded. It will send FAT (or no-FAT) and file number information in the same way as the SPI FLASH player program before starting playback.

If SD card is removed during SD playback or mass storage operation, "!SD" and newline (0x0a) are sent and the SPI FLASH player program is loaded.

If USB attach is detected during play mode, USB mass storage mode is started. "USB Attach SPI" or "USB Attach SD" and newline (0x0a) is displayed before going into mass storage mode (SPI FLASH and SD, respectively). When USB is detached, "USB Detach SPI" or "USB Detach SD" and newline is displayed before returning to play mode.

The selected play mode is preserved when switching between SPI FLASH and SD/MMC player programs. The same applies to the volume setting.

## Optional Features

Note: Some of these features are not active by default, depending on the firmware type. You need to adjust preprocessor definitions from `system.h`, recompile, and reprogram the module to add or remove these features.

The UART-controlled firmware is the default available in the web store. Other configurations, and specific old versions are available with additional unit cost. Check the web store for a list of available custom versions.

## UART\_CONTROL

When **UART\_CONTROL** is defined in `system.h` (this is the default), the player prints status messages to UART and expects user commands from UART. The default speed is 115200 bps with 8N1 format.

UART\_CONTROL and GPIO\_CONTROL should not be enabled at the same time.

## GPIO\_CONTROL

When **GPIO\_CONTROL** is defined in `system.h` (instead of `UART_CONTROL`), GPIO pins select the song to play. The normal polarity for the GPIO pins is idle/inactive when pulled low, and selected/active when pulled high.

By default `GPIO_MASK` is defined to `0x1f` in `gpioctrl.c`, and specifies that `GPIO0_0` to `GPIO0_4` are used as a binary-coded number. If any combination of the pins are pulled high, the file corresponding to the binary-coded value is played. While the value remains the same, the same file is played in a loop. If none of the pins are pulled high, the current song is played to the end before playback is stopped. For example if only `GPIO0_0` is pulled high, the first song is played, if both `GPIO0_1` and `GPIO0_2` are pulled high (`0b00110 = 6`), the 6th file is played.

Instead, `GPIO_PRIORITIES` can be defined to `0x4ff` in `gpioctrl.c`, and specifies that each GPIO pin from `GPIO0_0` to `GPIO0_7` and `GPIO0_10` select files directly. If any pin is pulled high, the corresponding file is played. If multiple pins are pulled high, the one with the lower number is played. If none of the pins are pulled high, the current song is played to the end before playback is stopped.

The polarity of the GPIO pins can be reversed by enabling `GPIO_INVERTED` define from `gpioctrl.c`. When this define is active, the idle state of the GPIO pins is high, and the active state is to pull the pins low.

Note: The firmware first drives the pins to a known state before scanning their state. This means that pull-up and/or pull-down should be used instead of directly connecting the pins to `IOVDD` or `GND`.

`UART_CONTROL` and `GPIO_CONTROL` should not be enabled at the same time.

## USE\_WAV

When **USE\_WAV** is defined in `system.h` (this is the default), the player also considers files with the `.WAV` suffix and plays mono and stereo 8-bit and 16-bit linear RIFF-WAV files. Comment out the define and recompile to disable the function.

## START\_IN\_FILE\_MODE

When you define **START\_IN\_FILE\_MODE** in `system.h`, the player starts in file mode instead of continuous play mode.

## Optional Commands

Note: None of these features are active by default. You need to adjust preprocessor definitions from `system.h`, recompile, and reprogram the module to be able to use these features.

### USE\_TYPE

When you define **USE\_TYPE** in `system.h`, command 'T' (type) will be included in the firmware. The 'T' command takes a 8.3-character filename as parameter. If the file is found, the number of the file is printed (two bytes), then the content of the file is printed out. Otherwise 0xff 0xff are returned.

### USE\_INFO

When you define **USE\_INFO** in `system.h`, command 'i' and 'I' (information) will be included in the firmware. The 'i' command takes a number as parameter, while the 'I' command takes a 8.3-character filename just like the play commands. For Ogg Vorbis files the info command returns the play length of the file in seconds without playing it. If the file does not exist, "not found" will be returned, so the 'I' command can also be used to determine if a file exists.

### USE\_QUEUE

When you define **USE\_QUEUE** in `system.h`, you can add files to a play queue before starting file playback.

Files are only added by name using the 'Q' command. If the named file is found, "add " followed by the file name and newline is printed. If the named file is not found, "nof " followed by the file name and newline is returned. If the queue is full "ful" and a newline is returned.

As long as there are files in the queue, whenever a play of a file is ended (end of file, next or previous commands), the next file from the queue will be played. "queue ", the number of files in the queue (this one included) and a newline character will be printed before starting playing the file with any normal play printouts.

The additional one-character command for the play mode is R, which clears the queue and stops playing, possibly returning to the file mode or continuing normal play from the next file.

### EXTCLOCK

If you use XTALI which is not 12.0 MHz, you can set the actual XTALI by defining **EXTCLOCK** in kHz. For example with 13MHz XTALI you would

```
#define EXTCLOCK 13000
```

## GAPLESS

When **GAPLESS** is defined in `system.h`, the firmware tries to speed up change from one Ogg file to the next, provided they have been encoded with the same parameters.

With the gapless feature, UART responses during file change are reduced. Only one byte is sent for the file number when play starts and a single 'd' is sent when the play ends.

In gapless mode the internal PLL is set to  $3.5 \times$  clock. (In normal mode  $3.0 \times$  clock is used.)

## WAV Playback

Since the 0.60 version the firmware uses a very compact wav decoder, and by default plays both linear 8-bit and 16-bit PCM WAV files and Ogg Vorbis files. However, the wav decoder is very stripped and optimized, for example fast play is not supported by the WAV decoder.

Due to a lot of code optimizations also elsewhere in the audio module firmware, WAV decoder can now be used together with USB code.

If you want, you can disable WAV decoding (and perhaps also USB) from the configuration file `system.h` to get more free instruction memory for your own changes.

Since version 0.70 the SD USB mass storage handling is a separate program, so there is more available memory in `sdplayer`.

## Updating

Since version 0.3 SD/MMC card can be used for firmware update. If a program named "SDUPDATE.PRG" is found on the SD/MMC card, it is automatically executed. The program can then for example open another file on the SD card and program the contents to SPI FLASH.

Both the boot firmware and audio content can be programmed in this way.

The default name of the firmware file is FIRMWARE.RAW . Copy spiall.spi from the VSIDE project and rename it to FIRMWARE.RAW.

The default name of the content file is CONTENT.RAW .

## Source Code

If you look at the source code (which is now also available as a VSIDE project), the software consists of four parts:

1. usbmass, which is executed after power-on and handles USB Mass Storage for SPI FLASH if USB is attached. Otherwise loads spiplayer or sdplayer depending on the SD/MMC insertion switch.
2. spiplayer, which plays audio from SPI FLASH, loads sdplayer if SD/MMC insertion switch is triggered, and loads usbmass (using watchdog reset) if USB attachment is detected.
3. sdplayer, which plays audio files from SD/MMC, loads sdmass if USB Attach is detected, loads spiplayer if SD/MMC insertion switch is deasserted.
4. sdmass, which handles USB Mass Storage, loads sdplayer if USB is detached, loads spiplayer if SD/MMC insertion switch is deasserted.

The parts are compiled and linked separately (usbmass.bin, spiplayer.bin, sdplayer.bin, sdmass.bin), and SPI boot images are generated from them. Then the boot images are combined into one image file (spiall.spi) that is programmed into SPI FLASH.

The starting offsets are specified in Makefile and passed as preprocessor definitions to programs.

In the VSIDE project this image creation is automatic, and the resulting eeprom.img can be programmed into the module with VSIDE's prommer named "VS1000 SPI Flasher (24-bit) prepared image". Note that the normal prommer overwrites eeprom.img .

## 9 Application Examples

### 9.1 Headphone Connection

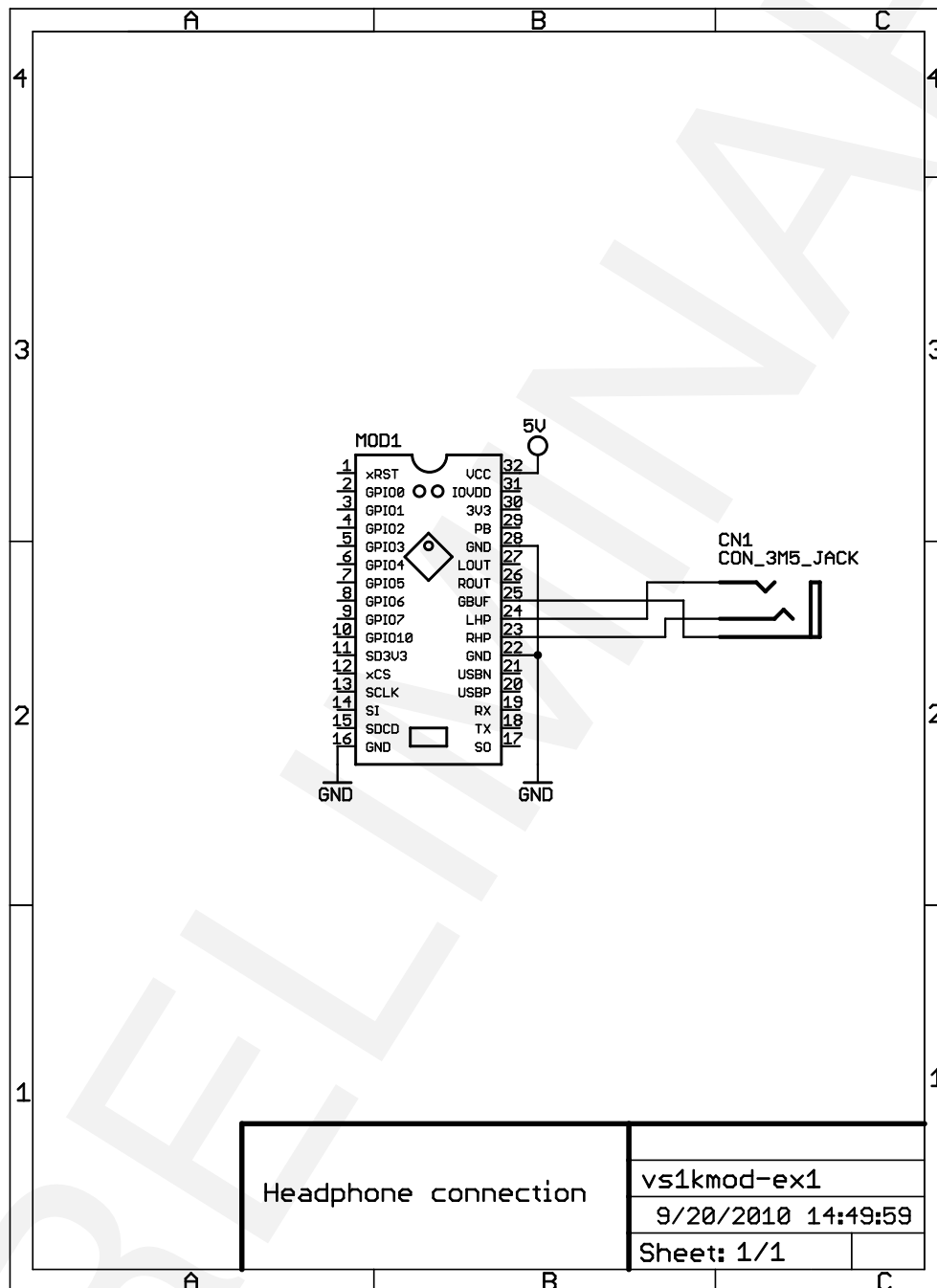


Figure 8: Headphone Connection.

The minimum connections required to get sound output is shown above.

The firmware starts playing the first audio file from the SPI FLASH automatically after power-on.

## 9.2 Line Out Connection

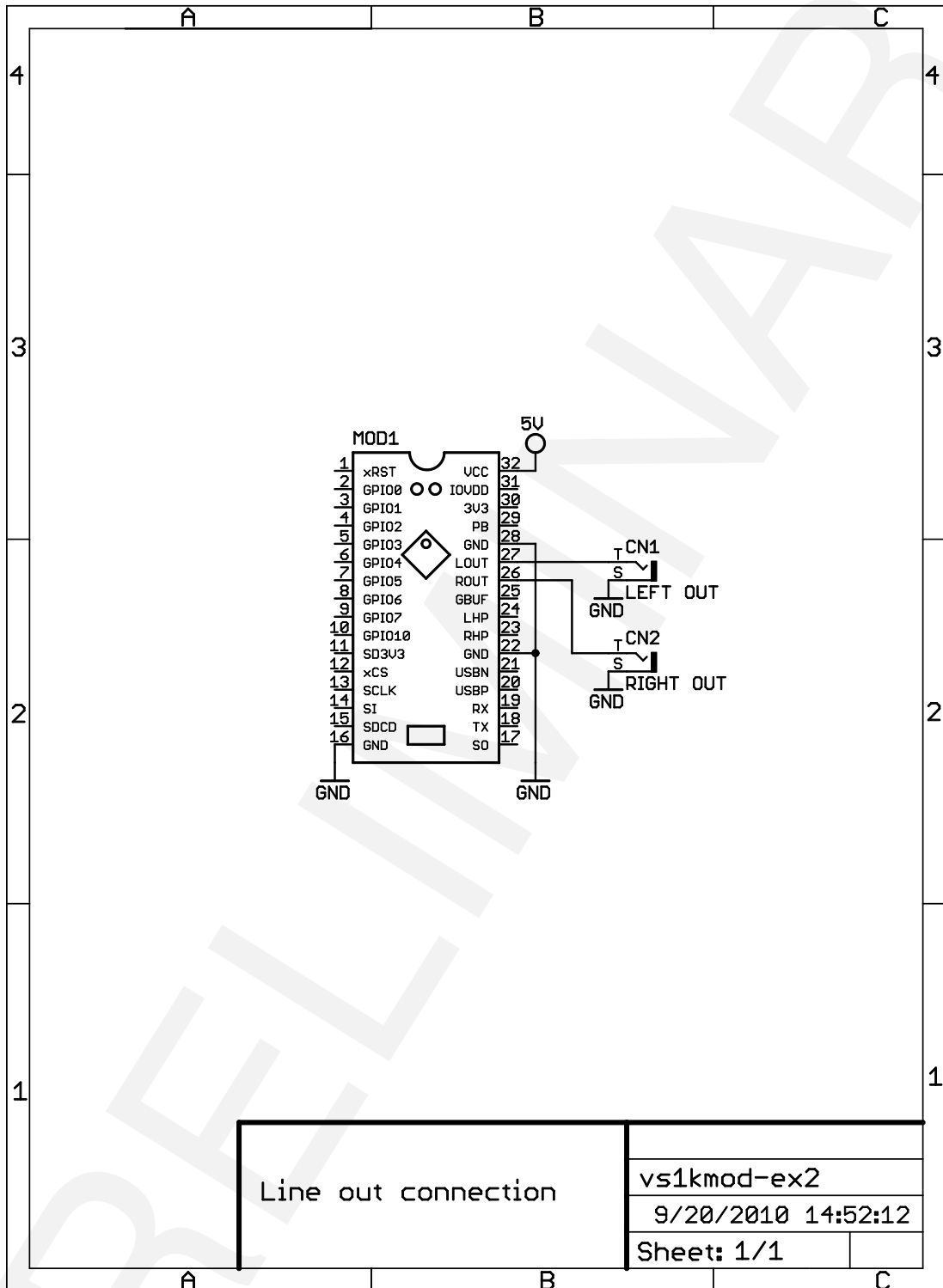


Figure 9: Line Out Connection.

9.3 USB Controlled and Powered Device with Headphone Output

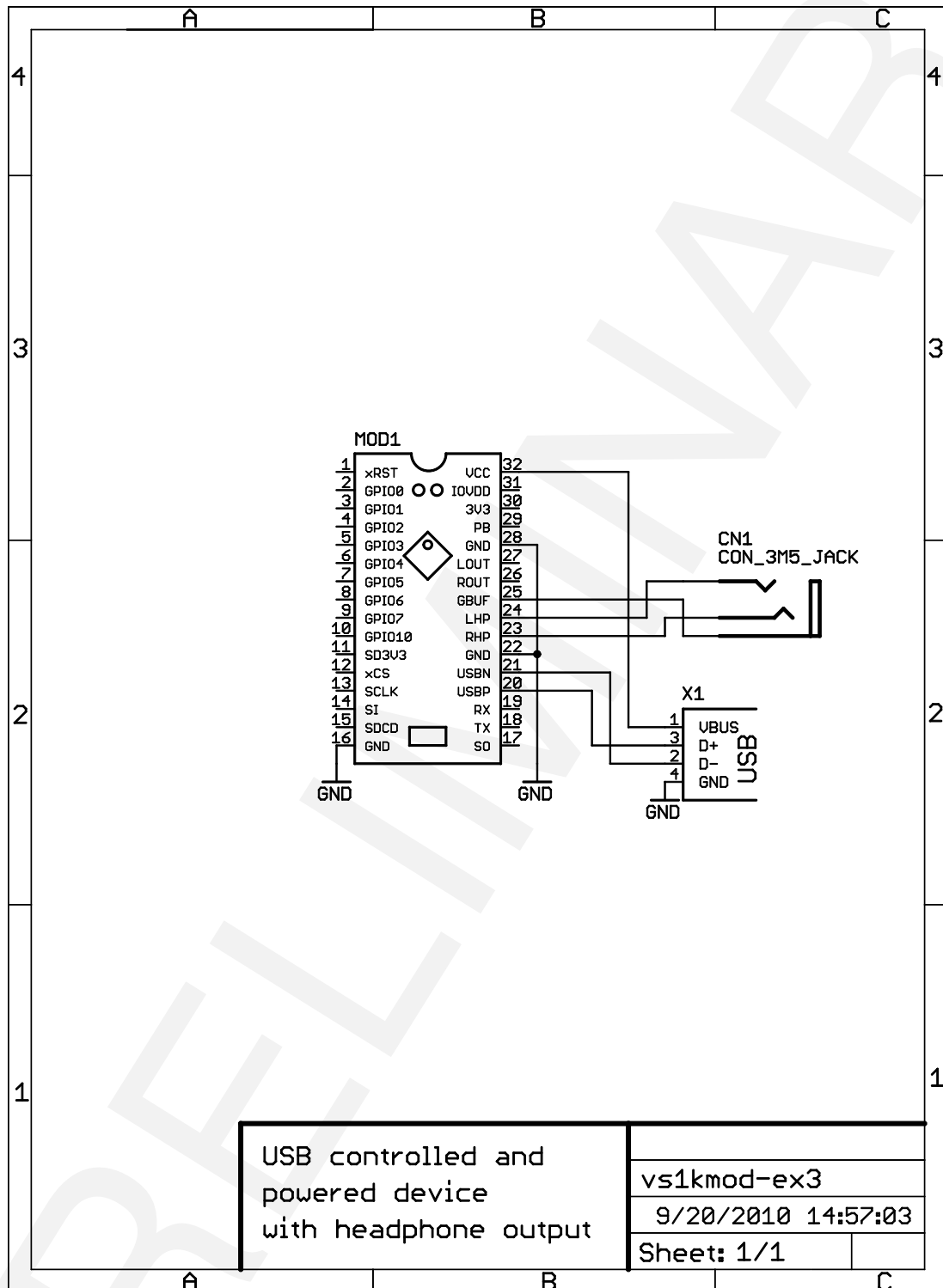


Figure 10: USB Controlled and Powered Device with Headphone Output.



9.4 UART Control

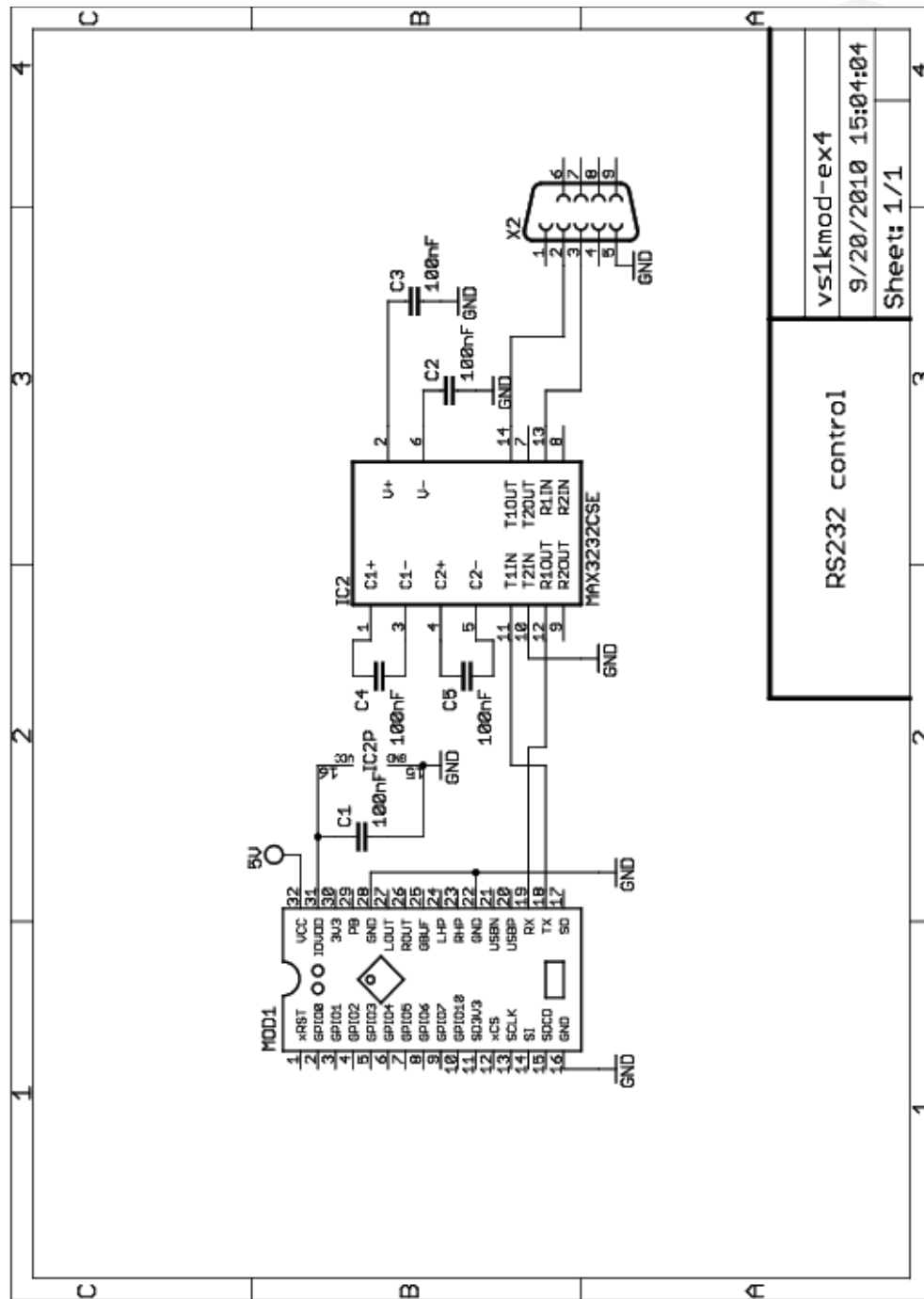


Figure 11: RS232 Control.

The TX and RX pins of the module are logic-level UART pins, so the use of the module with a RS232 interface requires an external RS232 adapter.

Alternatively you can use an logic-level (3.3 V) USB-UART adapter like the VSIDE UART Cable.

9.5 GPIO-Controlled Player

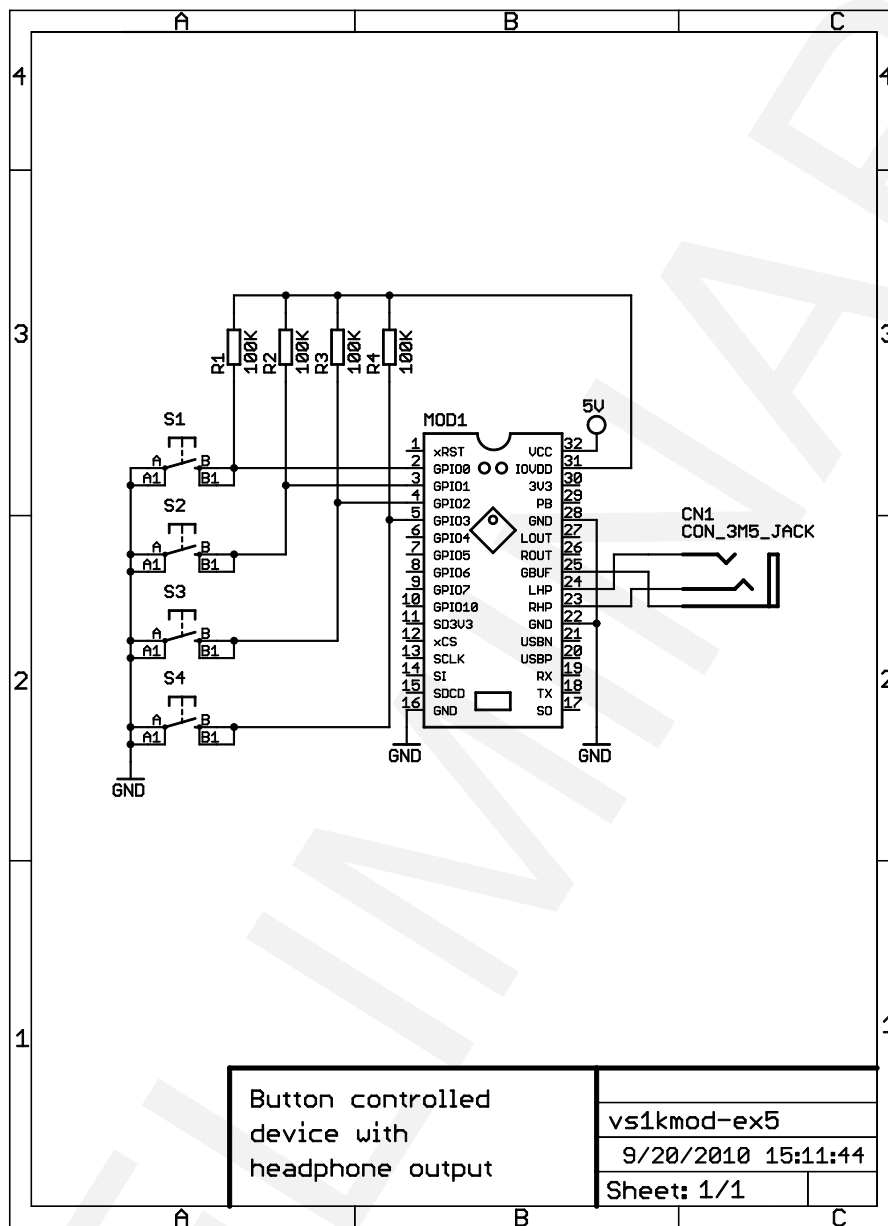


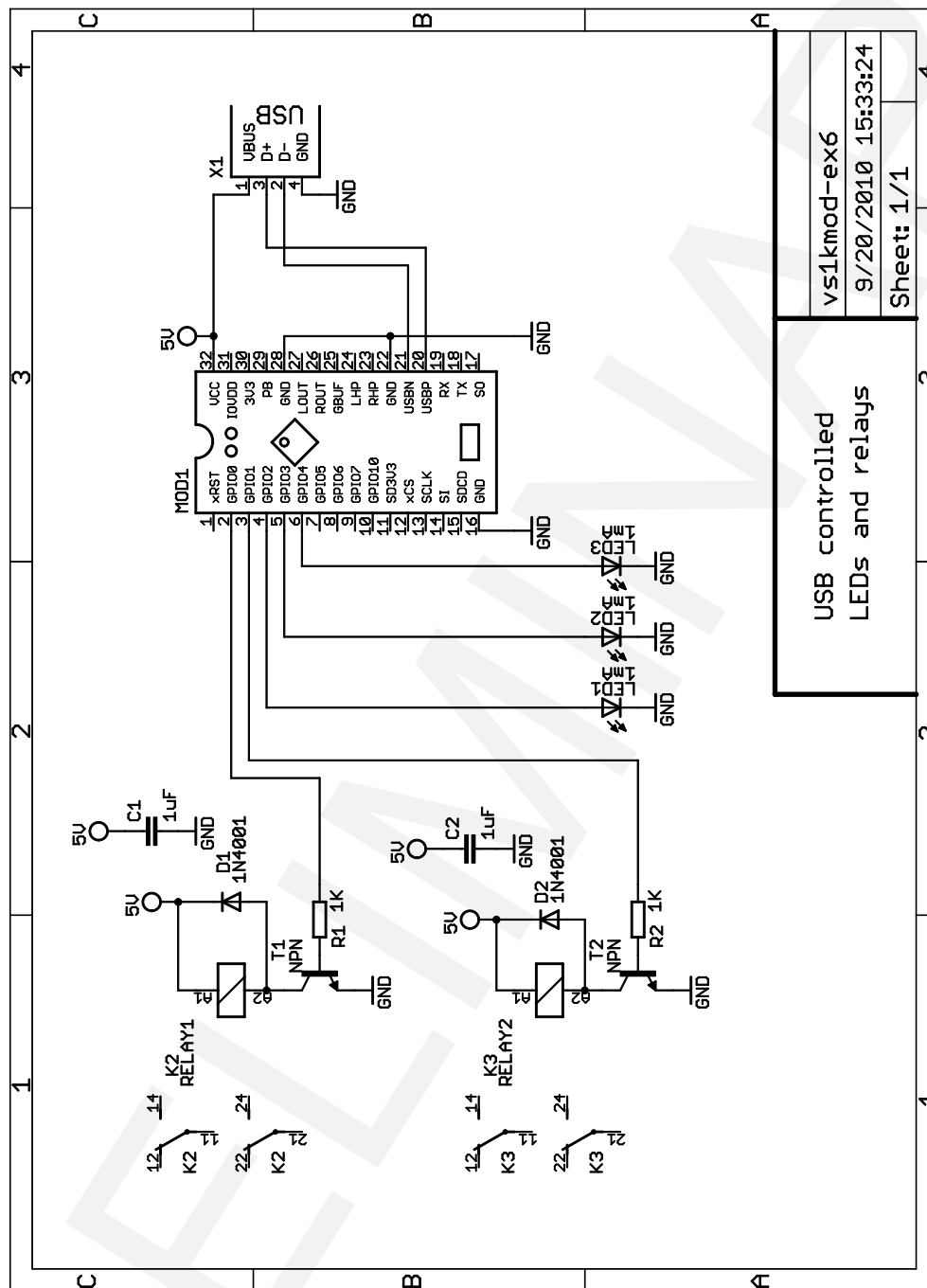
Figure 12: Button Controlled Player (idle high).

Note: The default GPIO-controlled firmware uses idle-low buttons (instead of idle-high like in the schematics above), so that it is compatible with the standard audio module that does not have any buttons connected.

The polarity can be adjusted from the source code (see `gpioctrl.c`).

The GPIO's are driven low (or high, depending on the polarity) by VS1000 before they are scanned, which tries to prevent floating lines from affecting the scanning. This is of course required only when testing, the buttons should have the proper pull-downs and pull-ups in your product. But this drive by VS1000 must be taken account if you connect external controllers (use a series resistor).

9.6 USB controlled LEDs and Relays



vs1kmod-ex6  
9/20/2010 15:33:24  
Sheet: 1/1

USB controlled LEDs and relays

Figure 13: USB Controlled LEDs and Relays.

Currently the default firmware uses USB only for mass storage functions. If a uSD card is inserted, that will be shown as a removable drive on the PC, otherwise the internal SPI FLASH is shown.

With some additional programming the PC could read and control the GPIO pins of the module through USB, allowing very varied applications.

## 10 Document Version Changes

This chapter describes the most important changes to this document.

### Version 0.70, 2015-01-13

- Added schematics for 1.61 .
- Added more firmware specs and talk about the optional features.

### Version 0.61, 2013-05-28

- Updated schematics, component placement and picture.

### Version 0.6, 2013-01-31

- Updates for firmware version 0.6.

### Version 0.41, 2010-11-09

- Corrected image placement.

### Version 0.4, 2010-11-04

- Clarified FLASH / EEPROM storage.
- Corrected component names in Chapter 6, Component Layout.

## 11 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](mailto: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](mailto:support@vlsi.fi)