



Datasheet USBPAL

OEM AUDIO Interface Reference

System Design Description for USBPAL

Document Version:
1.3

Document Revision:
2011-08-28

Copyright 2011 by RigiSystems AG, Zug, Switzerland. All rights reserved. Reproduction of part or all of the contents in any form is expressly prohibited without the prior written consent of RigiSystems AG.

RigiSystems AG has used its discretion, best judgments and efforts in preparing this document. Any information contained in this document is provided as is, without warranty of correctness or fitness for any particular purpose. RigiSystems AG may make improvements and/or changes of this document at any time, without prior notice.

Contents

1	Overview	5
1.1	Introduction	5
1.2	Specification	5
1.3	USBPAL Components	7
1.1	References	8
2	Supported Operating Systems	9
3	USB High Speed Streaming Interface	10
3.1	USB controller CY7C68013	10
3.2	USBPAL Startup	11
3.3	Startup Reset Signals	11
3.4	General purpose IO Signals	12
4	OEM Hardware Interface (OHI)	13
4.1	OHI Header pin assignment	13
4.2	OHI Signals Description	13
4.3	I2C Master	14
4.4	SPI	14
4.5	General purpose IO Signals	14
5	Audio Interfaces	16
5.1	Audio header pin assignment	16
5.2	AUDIO bus Signals	17
5.3	PCM Serial Audio Interface (I2S) channel format	17
5.4	DSD Audio	18
5.5	SPDIF	19
5.6	MIDI	20
5.7	Audio Clock generator	20
6	Configuration	21
6.1	Microsoft Windows	21
6.1.1	Configure serial Audio Format	21
6.1.2	Configure Channel Configuration	21
6.1.3	Configure Serial Audio Interface Input	22
6.1.4	Configure S/PDIF Audio Interface Input	23
6.1.5	Configure Serial Audio Interface Output	23
6.1.6	Configure S/PDIF Audio Interface Output	24
6.1.7	Configure MIDI I/O	24
6.2	Apple OS X	24

7	Reference Base Board.....	26
8	Electrical	27
9	Dimensions.....	28
10	USBPAL Main Components	29
11	Notes	30
12	Schematics	31

Figures

Figure 1: USBPAL USB audio interface reference	7
Figure 2: USB high speed interface (USB side).....	11
Figure 3: OHI header pin assignment	13
Figure 4: Audio bus header pin assignment	16
Figure 5: USBPAL Reference board physical	17
Figure 6: PCM audio formats	17
Figure 7: DSD audio formats.....	18
Figure 8: SPDIF format jumper	20
Figure 9: USBPAL Reference board physical	26
Figure 10: USBPAL Reference board dimension	28

1 Overview

1.1 Introduction

USBPAL is the solution for highest quality multi-channel audio product with high speed USB 2.0 connectivity. Based on just two off-the-shelf silicon components and a custom driver stack up to 36 channels of 24bit digital audio at 44.1 to 192kHz can be streamed.

The high speed, low latency drivers for Windows support XP to Windows 7. Support concurrent ASIO and WDM interface on windows. Mac OS-X Core Audio drivers support popular Apple applications such as GarageBand and Logic-Pro.

Through a Windows SDK the driver has a private software API for your custom applications and control panel. The SDK contains all required source code for customization and reference code for a full-featured control panel.

For customization no embedded software or hardware development is required. Replacing logos and audio channel configuration doesn't even require programming skills..

Special care in the design has been taken to support lowest jitter audio. As matter of fact USBPAL can easily be slaved to an existing high quality clock design. In this case no additional jitter is introduced in the system. With this PLL free clocking, the PC applications are slaved to the HW audio clock and not vice vers, thus both deterministic and random jitter are fully under control of the OEM clock design.

The reference hardware board includes two separate oscillators for the generation of the base clocks. These high quality oscillators are then digitally divided down to generate audio clocks. The jitter performance is good for the highest in audio quality.

USBPAL is a turnkey solution, and is enabling rapid product development by OEMs and ODMs.

1.2 Specification

Input channels	16, 8 @ 192KHz	44.1 – 192KHz	24bit
Output channels	16, 8 @ 192KHz	44.1 – 192KHz	24bit
S/PDIF Input	2	44.1 – 96KHz	24bit IEC / 32bit transparent
S/PDIF Output	2	44.1 – 96 KHz	24bit IEC / 32bit transparent
DSD Input	8	2.8MHz, 5.6MHz	1bit
DSD Output	8	2.8MHz, 5.6MHz	1bit
I2C		400KHz	Single master
SPI		~200Khz	Master only
MIDI IO		31.250KHz	

GPIO	10		
ASIO	34 channels	PCM / DSD	Configurable latency
WDM	34 channels	PCM	
MIXER	34x34channels		
Sample rate converter	2 channels		S/PDIF only
Sample rate detector	SPDIF Input		S/PDIF only
Input Latency	➤ 2.5 ms		IF/IF ASIO round trip ~6ms (PC dependent)
Output Latency	➤ 2.5 ms		

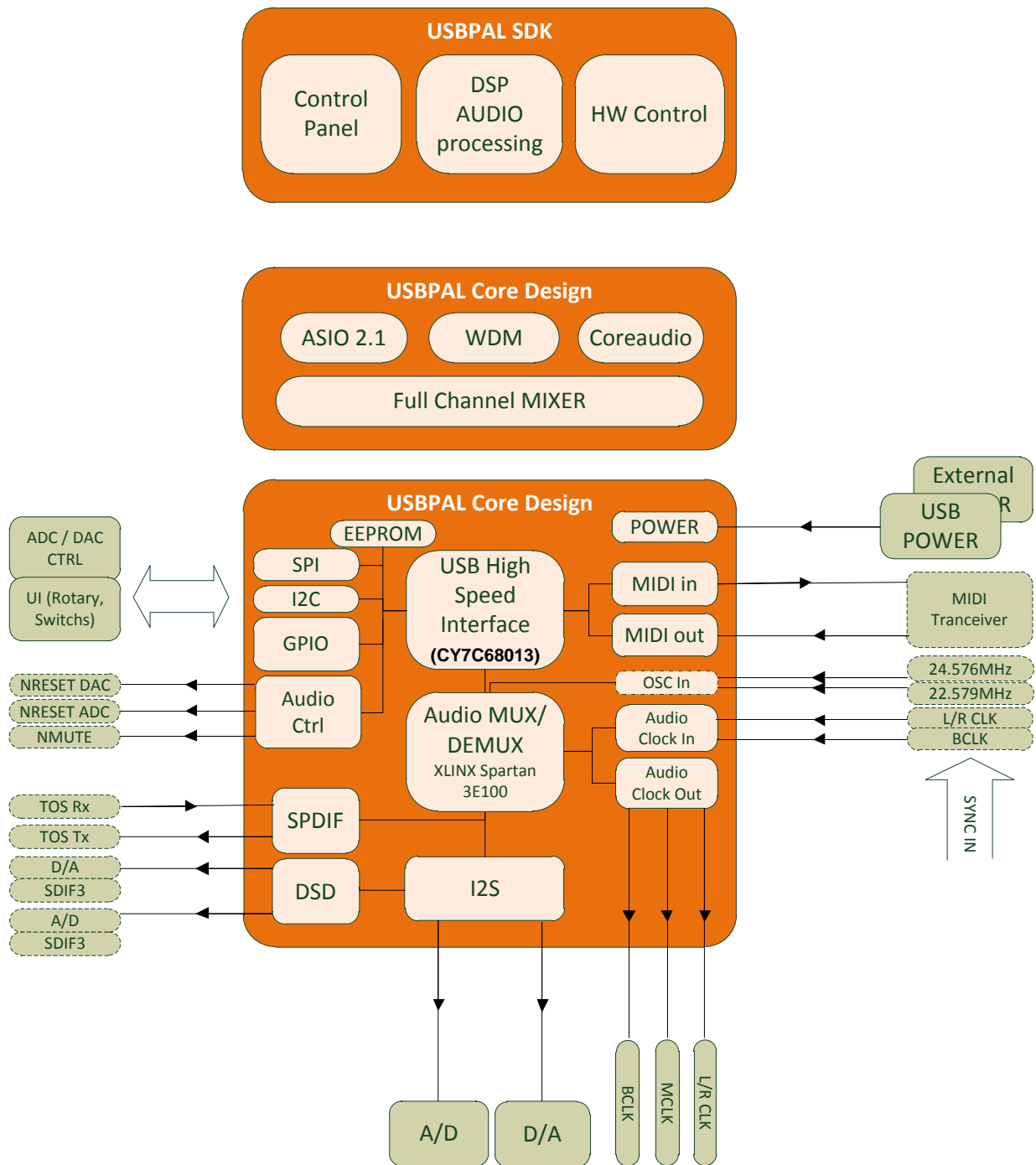


Figure 1: USBPAL USB audio interface reference

1.3 USBPAL Components

The USBPAL is composed of the following components.

Core Design:

- Microsoft Windows kernel drivers, including full channel low latency mixer
- ASIO 2.1 audio API

- WDM audio API
- Apple OS X Coreaudio driver
- Windows control service
- Hardware reference design (Schematics, Layout, BOM)
- FPGA image (automatically through driver)
- FX-2 USB interface firmware and bootloader (automatically loaded through driver)

USBPAL SDK

- Software SDK for customization
- Software Interface for in kernel audio processing plugins, like mixer effects etc.
- Software SDK for HW customization, such as setup of DAC / ADC through SPI or I2C.

Tools

- Software installer for Windows
- Software installer package for OS X (standard apple format)
- Production tool software, to initialize EEPROM and test USBPAL connectivity

1.1 References

- [1] Cypress, EZ-USB Technical Reference Manual, V1.4
http://download.cypress.com/publishedcontent/publish/design_resources/more_resources/contents/ez_usb_r___technical_reference_manual__trm__14.zip
- [2] Cypress, EZ-USB FX2LP USB Microcontroller Data Sheet, Document #38-08032, Rev.K,
http://download.cypress.com/publishedcontent/publish/design_resources/datasheets/contents/cy7c68013a_8.pdf
- [3] Cypress, Streaming Data through Isochronous/Bulk Endpoints on EZ-USB FX2(TM) and EZ-USB FX2LP(TM), AN4053
- [4] Philips Semiconductors, I2S Bus Specification, February 1986, Revised June 5, 1996
- [5] TA Document 2001024, Audio and Music Data Transmission Protocol V2.1, May 24, 2002
- [6] Windows DDK, Microsoft, 2003
<http://www.microsoft.com/ddk/>
- [7] USBPAL Schema v1.3, RigiSystems AG, USBPAL_013_SCH.pdf

2 Supported Operating Systems

USBPAL requires normal of the shelf HW and works with virtually any PC and MacIntosh computer.

OS	Architecture
Microsoft Windows XP	x86
Microsoft Vista	x86, x64
Microsoft Windows 7	x86, x64
Apple OS X 10.4	Intel only
Apple OS X 10.5	Intel only, 32 and 64bit kernel
Apple OS X 10.6	32 and 64bit kernel

For high channel count the following minimal configuration is recommended

Processor	Core Duo 2GHz or better
Chipset	Intel, VIA*
USB 2.0 host	Integrated high speed

USBPAL has been tested also with ATOM class netbooks. Many devices are not be able to support datarate of the full channel count, but work fine with lower channel configurations.

3 USB High Speed Streaming Interface

3.1 USB controller CY7C68013

USBPAL is based on the Cypress CY7C68013 EZ-USB® FX2™ USB microcontroller high-speed USB peripheral controller. Cypress's EZ-USB® FX2™ is the world's first USB 2.0 integrated microcontroller. By integrating the USB 2.0 transceiver, SIE, enhanced 8051 microcontroller, and a programmable peripheral interface in a single chip, Cypress has created a very costeffective solution that provides superior time-to-market advantages. The ingenious architecture of FX2 results in data transfer rates of 56 Mbytes per second, the maximum allowable USB 2.0 bandwidth, while still using a low-cost 8051 microcontroller in a package as small as a 56 SSOP. Because it incorporates the USB 2.0 transceiver, the FX2 is more economical, providing a smaller footprint solution than USB 2.0 SIE or external transceiver implementations. With EZ-USB FX2, the Cypress Smart SIE handles most of the USB 1.1 and 2.0 protocol in hardware, freeing the embedded microcontroller for application-specific functions and decreasing development time to ensure USB compatibility.¹

For USBPAL the controller is configured to run in high-speed (480Mbps) isochronous transfer mode only. The data throughput on full-speed and lower is insufficient for 24b high sample rate. E.g. 192Khz transfer. Thus the device will not stream audio on a full-speed (12Mbps) or slower USB host interface. However, the device will enumerate and the condition is signaled to the software and indicated on the control panel.

Audio data is transferred is using isochronous transfer on the USB bus and reserves the required bandwidth in advance. This gives audio data priority on the bus and reduces buffer requirements on the device to allow for very low latency.

The CY7C68013 has an integrated high speed USB phy. Thus little external components are required. However, the layout of the high-speed signals on the board must take the 480MHz frequency into account and layouted accordingly. The USB clock and CPU clock are generated from a 24MHz crystal connected directly to the controller.

The CY7C68013 also needs a little EEPROM to boot in the appropriate mode and with correct vendor / device identification. With the USBPAL production tool the EEPROM can be written with a correct bootloader software and license information.

¹ Cypress Semiconductor Corporation CY7C68013 Document #: 38-08012 Rev. *B

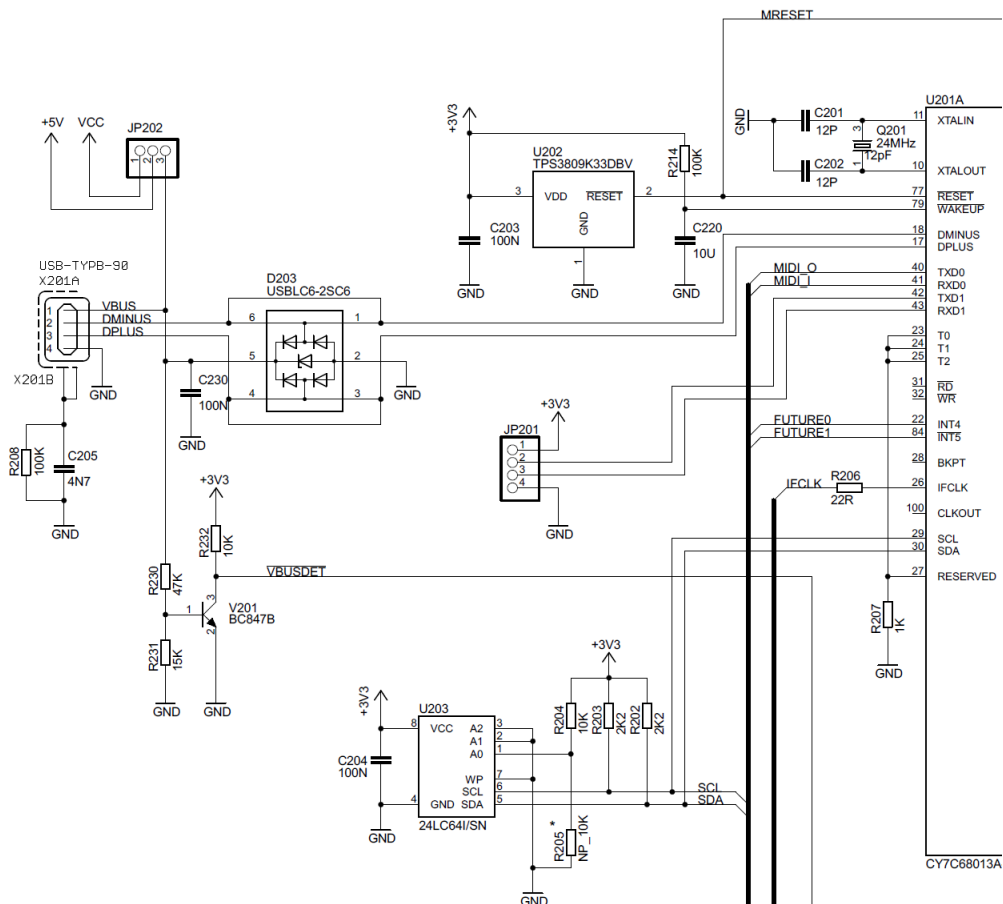


Figure 2: USB high speed interface (USB side)

3.2 USBPAL Startup

Each time the device is connected to USB the firmware is reloaded into controller. Also the FPGA is reloaded in case it was not initialized before. The process takes less than 100ms and is not noticed by the end-user as it is significantly faster than USB enumeration on even the fastest PCs.

1. The level 1 bootloader is stored in EEPROM and is loaded by the controller at power-on. It contains the the vendorID and product ID and serial number.
2. The level 2 bootloader is loaded through USB by the driver. The level 2 bootloader loads the FPGA image and configuration from the PC.
3. In the last step the application firmware is loaded into the controller.

Loading of firmware is performed using the Cypress firmware load mechanism (EZ-USB Technical Reference Manual v1.4)

3.3 Startup Reset Signals

During startup ADC and DAC are kept in reset to prevent any glitch on the audio output. Audio reset and audio mute are implemented by the following pins:

Signal	CY7C68013- TQFP100	OHI Header Reference Board
NRESET ADC (main codec reset)	GPIO0 (pin 57)	OHI pin 21
NRESET DAC	GPIO1 (pin 58)	OHI pin 22
NMUTE	GPIO2 (pin 59)	OHI pin 23
FPGA CORE POWER	GPIO9	OHI pin 30

GPIO0-2 can be freely customized in the SDK software. The above assignment is the default behavior implemented in the USBPAL SDK.

GPIO9 goes high when core power is enabled. The signal should be used for enabling additional circuitry, which consumes power at power up in order to avoid excessive power consumption at startup when bus powered.

3.4 General purpose IO Signals

OHI provides general purpose I/O pins. These pins are either provided by the FX2 8051 CPU or by the FPGA (accessible via registers). Each GPIO pin can be configured either as input or output.

PIN	NAME	DIRECTION	FUNCTION	REMARK
57	GPIO0	OUT	NRESET_ADC	External pull-down
58	GPIO1	OUT	NRESET_DAC	External pull-down
59	GPIO2	OUT	NMUTE	External pull-down
60	GPIO3	OUT	GP-OUT	
61	GPIO4	OUT	GP-OUT	
62	GPIO5	OUT	GP-OUT	
63	GPIO6	OUT	GP-OUT	
64	GPIO7	OUT	GP-OUT	
86	SPI_CS	OUT	SPI_NCS_FPGA	

GPIO0-7 can be freely customized in the SDK software. The above assignment is the default behavior implemented in the USBPAL SDK.

4 OEM Hardware Interface (OHI)

This section defines the hardware interface between USBPAL module and the OEM components in detail. Hardware requirements, especially for the FPGA, can be derived from this specification.

4.1 OHI Header pin assignment

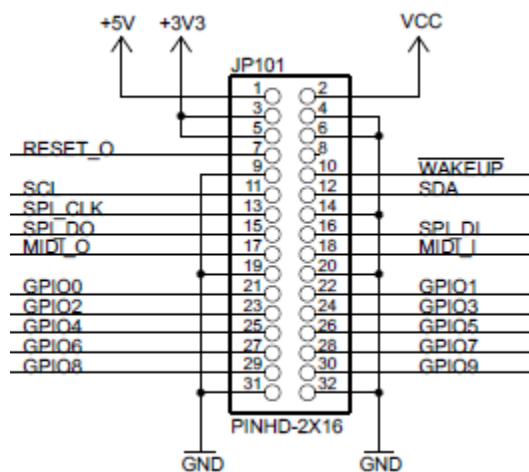


Figure 3: OHI header pin assignment

4.2 OHI Signals Description

Signal	Dir	OHI Header	Connected to	Description
RESET_O	OUT	7	FX2	RESET of OEM hardware Active while USB device is un-configured or in suspend state
NWAKEUP	IN	10	FX2	External interrupt
SDA	IN	12	FX2	I2C master data line (open drain output + input)
SCL	OUT	11	FX2	I2C master clock line (open drain)
SPI_CLK	OUT	13	FX2	SPI clock
SPI_DO	OUT	15	FX2	SPI data out
SPI_DI	IN	16	FX2	SPI data in
MIDI_I	IN	18	FX2	MIDI data in
MIDI_O	OUT	17	FX2	MIDI data out
GPIO0..9	IN/OUT	21..30	FX2	General Purpose I/O or SPI chip select

4.3 I2C Master

OHI provides one I2C master. The USBPAL module acts as a master. An OEM is able to connect one or more I2C slave devices, e.g. DACs, ADCs, etc. A unique I2C slave address has to be assigned to each device.

The I2C master is implemented internally in the FX2. So I2C access is handled by the 8051 CPU. The I2C bus is not routed through the FPGA.

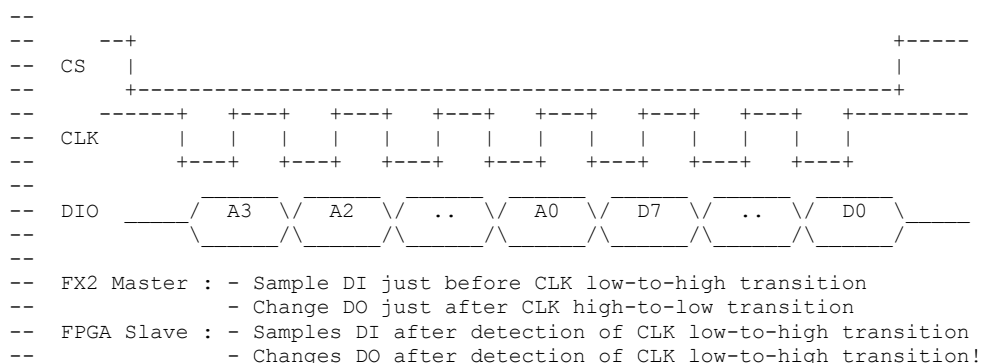
The device driver provides transparent access to I2C slave devices. So OEMs can implement custom control panels, mixer panels, etc.

4.4 SPI

OHI provides an SPI interface. The USBPAL module acts as master. An OEM is able to connect one or more slave devices, e.g. DACs, ADCs, etc. A separate SPI chip select line has to be routed to each slave device.

The SPI interface is implemented in the FX2 in software. GPIO pins are used as SPI chip selects.

The device driver provides transparent access to SPI slave devices. So OEMs can implement custom control panels, mixer panels, etc.



4.5 General purpose IO Signals

OHI provides general purpose I/O pins. These pins are either provided by the FX2 8051 CPU or by the FPGA (accessible via registers). Each GPIO pin can be configured either as input or output.

OHI header	PIN FX2	NAME	DIRECTION	FUNCTION	REMARK
21	57	GPIO0	OUT	NRESET_ADC	External pull-down
22	58	GPIO1	OUT	NRESET_DAC	External pull-down
23	59	GPIO2	OUT	NMUTE	External pull-down

24	60	GPIO3	OUT	GP-OUT	
25	61	GPIO4	OUT	GP-OUT	
26	62	GPIO5	OUT	GP-OUT	
27	63	GPIO6	OUT	GP-OUT	
28	64	GPIO7	OUT	GP-OUT	
29-30	86	SPI_CS	OUT	SPI_NCS_FPGA	

GPIO0-7 can be freely customized in the SDK software. The above assignment is the default behavior implemented in the USBPAL SDK.

5 Audio Interfaces

5.1 Audio header pin assignment

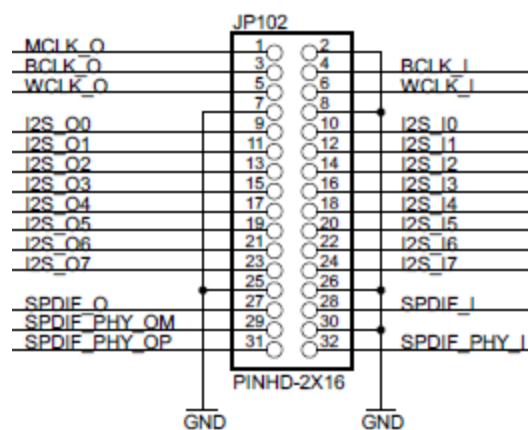


Figure 4: Audio bus header pin assignment

5.2 AUDIO bus Signals

Signal	Audio pin	Dir	Connected to	Description
MCLK_O	1	OUT	FPGA	Master clock output when internal clock generator is used
BCLK_O	3	OUT	FPGA	Bit clock output of internal clock generator
WCLK_O	5	OUT	FPGA	L/R clock output of internal clock generator
BCLK_I	4	IN	FPGA	Bit clock reference clock input (this clock is used to sample audio, connect to BCLK_O to use internal clock generator)
WCLK_I	6	IN	FPGA	L/R clock reference clock input (this clock is used to sample audio, connect to WCLK_O to use internal clock generator)
I2S_I0..7	9,11,...,23	IN	FPGA	Serial audio input signals (terminate with pull down)
I2S_O0..7	10,12,...,24	OUT	FPGA	Serial audio output
SPDIF_O	27	OUT	FPGA	CMOS SPDIF Signal
SPDIF_I	28	IN	FPGA	CMOS SPDIF Signal
SPDIF_PHY_OM	29	OUT	FPGA	IEC compliant SPDIF signal
SPDIF_PHY_OP	31	OUT	FPGA	IEC compliant SPDIF signal
SPDIF_PHY_I	32	IN	FPGA	IEC compliant SPDIF signal

Figure 5: USBPAL Reference board physical

5.3 PCM Serial Audio Interface (I2S) channel format

The below maximum configurations are possible. Any configuration with less channels can be configured in the registry through the *.inf file(s).

Sample rate	Serial Input	Serial Output	SPDIF In	SPDIF Out
44100 Hz	16	16	2	2
48000 Hz	16	16	2	2
88200 Hz	16	16	2	2
96000 Hz	16	16	2 ²	2
176400 Hz	8	8		
192000 Hz	8	8		
352800 Hz	4	4		
38400 Hz	4	4		

Figure 6: PCM audio formats

² The frequency of the input must not exceed the 96KHz as input on WCLK_i

Sample formats:

- **24 bit LPCM, 1 sample per DWORD**
- 32 bit transparent, 1 sample per DWORD (special mode)

All PCM IN and OUT channels are synchronized to the I2S sample clock.

The serial audio interface can be configured into 2 different modes:

- I2S compliant (1 bit data offset)
- **Left justified (0 bit data offset) this is default**

The number of channels and serial audio format used can be configured in the driver information file *.inf (see section audio channel configuration)

5.4 DSD Audio

Sample rate	DSD	Serial Input	Serial Output	SPDIF In	SPDIF Out
2.8 MHz (44100 Hz)	64x	8	8	2	2
(48000 Hz)	n/a	8	8	2	2
5.6 MHz (88200 Hz)	128x	8	8	2	2
96000 Hz	n/a	8	8	2 ²	2

Figure 7: DSD audio formats

Sample formats:

- 1-bit linear DSD

DSD and PCM audio can be mixed. The following mapping of sample rate / DSD bit rate applies

44100 Hz results in 2.8MHz (64x) DSD

88200 Hz results in 5.6MHz (128x) DSD

Each DSD audio channel requires the equivalent of 2 PCM audio channel of bandwidth. Therefore the total number of combined audio channels is reduced from 16 channels in mixed mode. For example a combination could have 14 PCM channels + 1 DSD channel, or 12 PCM audio channels + 2 DSD audio channels.

Channel configuration is provided in the driver information file *.inf (see section audio channel configuration)

5.4.1 ASIO DSD Interface

ASIO is implemented according to the ASIO standard ASIO 2.1 and higher. The format used is

ASIOSTDSDInt8MSB1

Some notes on how to use ASIOIoFormatType.

The caller will fill the format with the request types. If the board can do the request then it will leave the values unchanged. If the board does not support the request then it will change that entry to Invalid (-1).

So to request DSD then:

```
ASIOIoFormat NeedThis={ ASIOSTDSDInt8MSB1 };
if(ASE_SUCCESS != ASIOFuture(kAsioSetIoFormat,&NeedThis) )
{
    If the driver did not accept one of the parameters then the whole call will fail and the failing
    parameter will have had its value changes to -1.
}
```

5.5 SPDIF

The USBPAL module provides:

- One SPDIF input
SPDIF IN is *not* synchronized to the I2S sample clock. The SPDIF IN channel is transferred to the PC separately. The driver implements sample rate conversion (SRC) to synchronize SPDIF IN to the PCM channels
On SPDIF IN any sample rate up to the current sample rate set on the module is accepted. Higher sample rates are not accepted.
Thus, the input can accept 48KHz SPDIF on a 88.2KHz base clock. The output SPDIF will always be the same as the base clock (e.g. 88.2KHz).
- One SPDIF output
SPDIF OUT is synchronized to the I2S sample clock.
So on SPDIF OUT the currently selected sample rate is available.

SPDIF IN and SPDIF OUT channels are transferred in 32 bit transparent mode. Parsing (on input) and formatting (on output) of SPDIF compliant data streams is implemented by the device driver.

SPDIF is available both as a CMOS signal, e.g. to attach an optical transceiver (POF) or as SPDIF electrical conforming to IEC signal levels.

This can be jumpered on the JP 401

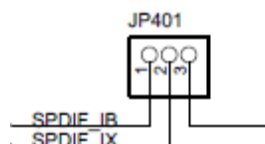


Figure 8: SPDIF format jumper

Connect: Jumper 2-3 for IEC, connect jumper 1-2 for CMOS

5.6 MIDI

The USBPAL module provides:

- One MIDI input channel
- One MIDI output channel

A MIDI channel is implemented as a UART configured as follows:

- 31'250 bits per second, 10 bits per data byte (1 start, 1 stop, 8 data bits)
- resulting transfer rate: 3125 bytes per second

5.7 Audio Clock generator

The following default clock ratio are used.

SAMPLING RATE	MCLK	BCLK	WCLK
44.1 KHz	11.2896 MHz	2.8224 MHz	44.1 KHz
48.0 KHz	12.2880 MHz	3.0720 MHz	48.0 KHz
88.2 KHz	11.2896 MHz	5.6448 MHz	88.2 KHz
96.0 KHz	12.2880 MHz	6.1440 MHz	96.0 KHz
176.4 KHz	22.5792 MHz	11.2896 MHz	176.4 KHz
192.0 KHz	24.5760 MHz	12.2880 MHz	192.0 KHz
352.8 KHz	22.5792 MHz	22.5792 MHz	352.8 KHz
384.0 KHz	24.5760 MHz	24.5760 MHz	384.0 KHz

6 Configuration

USBPAL behavior can be easily customized by software. Most configuration settings don't need any programming at all, as they can be conveniently set over registry keys.

The following section lists the most important sections of the driver *.inf file. The settings can be changed by modifying the file and reinstalling the driver.

6.1 Microsoft Windows

```
;*****
; Registry sections
;-----
[_AddReg_HwSetup]
; Various options as an OR combination of the bit masks described below.
; No flag is set by default.
```

6.1.1 Configure serial Audio Format

```
HKR, "HwSetup\Settings", Flags, %REG_DWORD%, 0x00000000
;
; Flags:
; 0x00000001 -- I2S format: if set: I2S compatible, if not set: MSB justified
; 0x00000002 -- WCLK source: if set: generate WCLK internally, if not set:
use external WCLK
```

6.1.2 Configure Channel Configuration

Different set of audio interface configurations can be specified. Assign a configuration to each sample rate. **Attention**, the total bandwidth limit has to be regarded (16x16 up to 96KHz) (8x8 up to 192KHz).

```
;
; Define the supported sample rates.
; For each sample rate specify the hardware port setup to be used.
;
HKR, "HwSetup\44100", SetupId, %REG_SZ%, "LowFs"
HKR, "HwSetup\48000", SetupId, %REG_SZ%, "LowFs"
HKR, "HwSetup\88200", SetupId, %REG_SZ%, "LowFs"
HKR, "HwSetup\96000", SetupId, %REG_SZ%, "LowFs"
HKR, "HwSetup\176400", SetupId, %REG_SZ%, "HighFs"
HKR, "HwSetup\192000", SetupId, %REG_SZ%, "HighFs"
HKR, "HwSetup\352800", SetupId, %REG_SZ%, "DxDfs"
HKR, "HwSetup\384000", SetupId, %REG_SZ%, "DxDfs"
```

6.1.3 Configure Serial Audio Interface Input

```
; Hardware port setup for low sample rates (44100..96000)
;
; I2S inputs
HKR, "HwSetup\LowFs\I2S_I0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I1",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I2",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I3",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I4",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I5",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I6",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I7",    Type,    %REG_DWORD%,    %I2S_PORT%

;
; Hardware port setup for high sample rates (176400..192000)
;
; I2S inputs
HKR, "HwSetup\HighFs\I2S_I0",   Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_I1",   Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_I2",   Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_I3",   Type,    %REG_DWORD%,    %I2S_PORT%

;
; Hardware port setup for high sample rates (352800..384000)
;
; I2S inputs
HKR, "HwSetup\DxDfs\I2S_I0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\DxDfs\I2S_I1",    Type,    %REG_DWORD%,    %I2S_PORT%
```

6.1.4 Configure DSD Audio Interface Input

```
; I2S inputs
HKR, "HwSetup\LowFs\I2S_I0",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I1",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I2",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I3",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I4",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I5",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I6",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I7",    Type,    %REG_DWORD%,    %DSD_PORT%
```

Any combination of I2S and DSD mix is possible. See example bellow for a Stereo I2S plus 2 channels of DSD:

```
HKR, "HwSetup\LowFs\I2S_I0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I1",    Type,    %REG_DWORD%,    %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I2",    Type,    %REG_DWORD%,    %DSD_PORT%
```

6.1.5 Configure S/PDIF Audio Interface Input

```
; SPDIF input
HKR, "HwSetup\LowFs\SPDIF_I0", Type,    %REG_DWORD%,    %SPDIF_PORT%
; the following flag enables SPDIF IN transparent mode (no deframer, no SRC)
;HKR, "HwSetup\LowFs\SPDIF_I0", Flags, %REG_DWORD%,    0x00000001

;Note that SPDIF Input is not available at frequencies above 96KHz
```

6.1.6 Configure Serial Audio Interface Output

```
; I2S outputs
HKR, "HwSetup\LowFs\I2S_O0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O1",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O2",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O3",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O4",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O5",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O6",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_O7",    Type,    %REG_DWORD%,    %I2S_PORT%

; I2S outputs
HKR, "HwSetup\HighFs\I2S_O0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_O1",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_O2",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\HighFs\I2S_O3",    Type,    %REG_DWORD%,    %I2S_PORT%

; I2S outputs
HKR, "HwSetup\DxDf\I2S_O0",    Type,    %REG_DWORD%,    %I2S_PORT%
HKR, "HwSetup\DxDf\I2S_O1",    Type,    %REG_DWORD%,    %I2S_PORT%
```

6.1.7 Configure DSD Audio Interface Output

```
; I2S outputs
```

```
HKR, "HwSetup\LowFs\I2S_00", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_01", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_02", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_03", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_04", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_05", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_06", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_07", Type, %REG_DWORD%, %DSD_PORT%
```

Any combination of I2S and DSD mix is possible. See example bellow for a Stereo I2S plus 2 channels of DSD:

```
HKR, "HwSetup\LowFs\I2S_I0", Type, %REG_DWORD%, %I2S_PORT%
HKR, "HwSetup\LowFs\I2S_I1", Type, %REG_DWORD%, %DSD_PORT%
HKR, "HwSetup\LowFs\I2S_I2", Type, %REG_DWORD%, %DSD_PORT%
```

6.1.8 Configure S/PDIF Audio Interface Output

```
; SPDIF output
HKR, "HwSetup\LowFs\SPDIF_00", Type, %REG_DWORD%, %SPDIF_PORT%
```

Note that SPDIF Output is not available at frequencies above 96KHz

6.1.9 Configure MIDI I/O

```
;
; MIDI ports
;
; MIDI in
HKR, "HwSetup\MIDI_I0", Type, %REG_DWORD%, %MIDI_PORT%
; MIDI out
HKR, "HwSetup\MIDI_O0", Type, %REG_DWORD%, %MIDI_PORT%
```

6.2 Apple OS X

Apple OS X boots kernel driver at a very early stage of OS initialization. For this reason the USBPAL kernel driver cannot access files yet, as the respective OS service is not available at driver initialization time. For this reason the configuration of the driver requires a recompilation of the OS X driver. All settings are stored in the configuration files `UsbpalCustomExtensionImpl.cpp` and `usbpal_customization.sh`.

```
BEGIN CUSTOMIZATION

export PRODUCT_NAME="USBPAL_RDK"
```

```

export REVERS_WEBPAGE="net.rigisystems"

export MIDI_FACTORY_UUID="4A373765-CD6C-4783-AD7B-60D91CB09B5E"

export USBPAL_PLATFORM="$PRODUCT_NAME"

export USBPAL_CLASS_PFX="net_rigisystems_usbpalrdk"

export USBPAL_VID=0x152a
export USBPAL_PID=0x100

# END CUSTOMIZATION

bool
USBPAL_CLASSNAME(UsbpalCustomExtensionImpl)::OnStart()
{
    TRACE(TRCINF, Print("%s\n", __PRETTY_FUNCTION__));

    switch (mDeviceProperties.usbProductId) {
        case 0x152a:
            mI2sInChannels = 16;
            mI2sOutChannels = 16;
            mSpdifAvailable = true;
            mMidiAvailable = true;
            break; true
        default:
            TRACE(TRCERR, Print("%s invalid usbProductId (%u)\n",
__PRETTY_FUNCTION__,mDeviceProperties.usbProductId));
            return false;
    }

    // success
    return true;
}

```



8 Electrical

Power Estimation		A			
		+3.3V	-3.3V	+2.5V	+1.2V
Crystal Oscillator	2	0.025			
Cypress FX-2 USB IF	1	0.060			
FPGA	1	0.020		0.020	0.100
TPS3809 Reset	1	0.001			
24LC64 EEPROM	1	0.001			
Total current		0.132		0.020	0.100

9 Dimensions

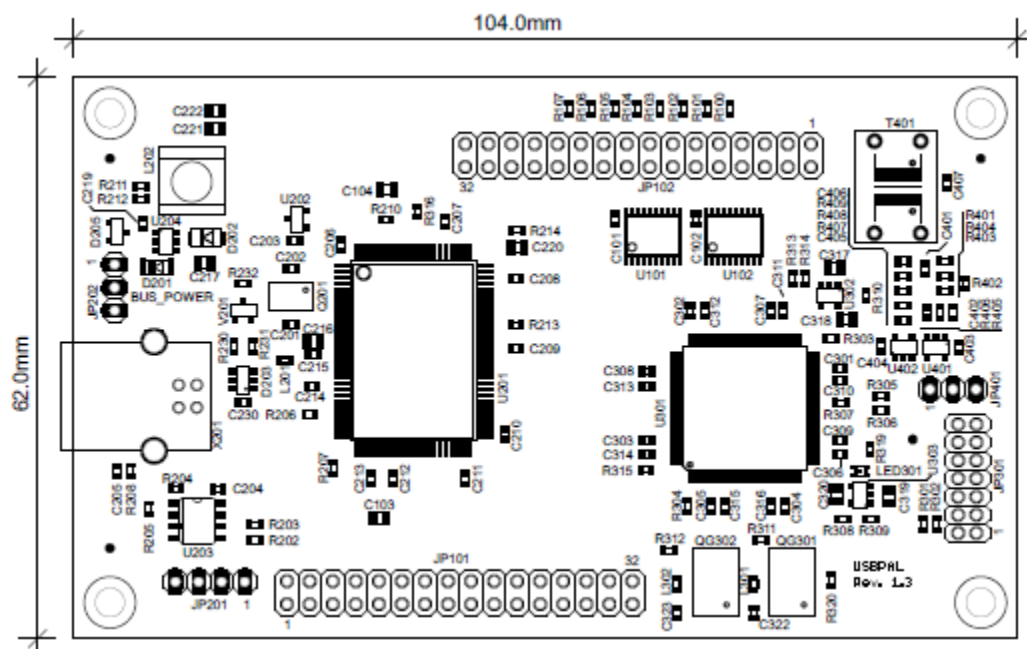
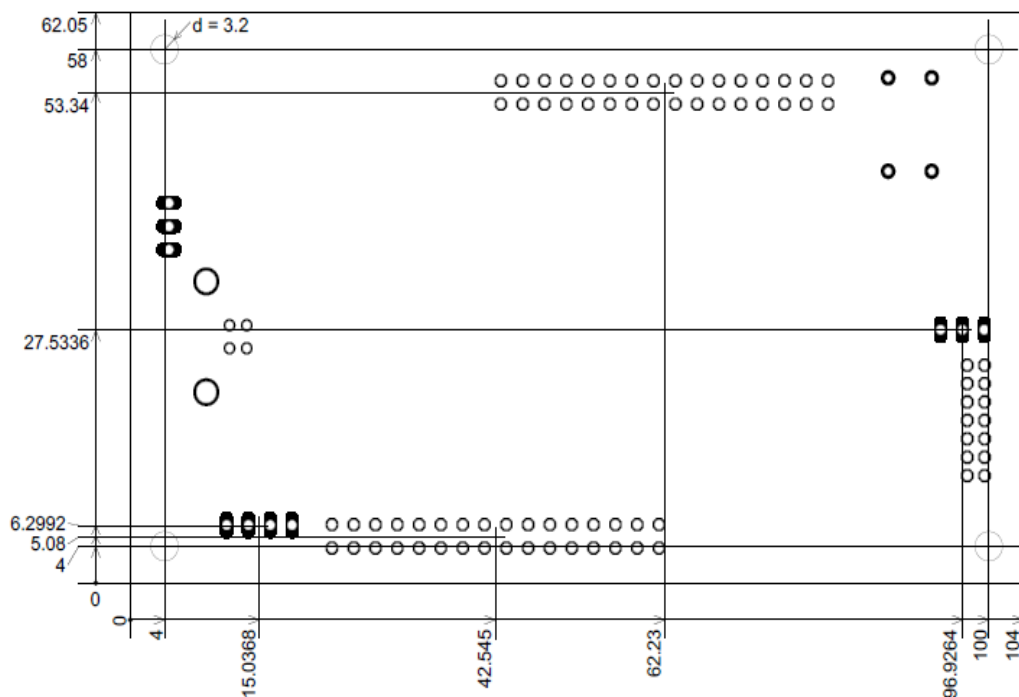


Figure 10: USBPAL Reference board dimension

10 USBPAL Main Components

Qty	Parts	Value	Description	Package	Manufacturer
1	D203	USBLC6-2SC6	USB ESD Protection Device	SOT23-6	ST
1	Q201	24MHz	Quartz Crystal, Case Size 5.0 x 3.2 mm, SMD, 30ppm, 12pF, -20°C to +70°C	CRYSTAL-5.0X3.2	Jauch
1	QG301	22.5792MHz	Crystal Oscillator, Case Size 7 x 5 mm, SMD, 3.3V, 50ppm, 0°C to +70°C	CRYSTAL-7.0X5.0	NSK
1	QG302	24.576MHz	Crystal Oscillator, Case Size 7 x 5 mm, SMD, 3.3V, 50ppm, 0°C to +70°C	CRYSTAL-7.0X5.0	NSK
1	U201	CY7C68013A-100AXC	USB 2.0 Interface, 8051 MCU, Serial Interface Engine	QFP-100-0.65-20.2X16.2	Cypress Semi
1	U202	TPS3809K33DBV	Voltage Supervisor, Delay = 200 ms	SOT23	TI
1	U203	24LC64I/SN	Serial EEPROM, I2C	SO08	Microchip
1	U204	LM2736Y	Step-Down Regulator, 3V to 18V, 0.75A	SOT23-6	NS
1	U301	XC3S100E-4VQG100C	FPGA, Spartan 3E Series, 100K Gates, 100-Pin Package	QFP-100-0.50-16.0X16.0	Xilinx

11 Notes

12 Schematics

