



High Performance Software Defined Radio

Open Source (GNU type) Hardware and Software Project
Project Description: <http://hpsdr.org>



Hardware Project #2

JANUS Board

Part #1 How JANUS works

Schematics / Board Design

Phil Harman, VK6APH
Bill Tracey, KD5TFD
Lyle Johnson, KK7P
Philip Covington, N8VB

Text

Phil Harman, VK6APH
Horst Gruchow, DL6KBF
HPSDR WIKI

Graphics and Layout

Horst Gruchow, DL6KBF

Project Coordinator

Eric Ellison, AA4SW

Contents

Where to find everything

Contents

Where to find everything	2
--------------------------	---

JANUS - the Audio IN-OUT

About the JANUS Module	3
------------------------	---

JANUS - the Board

Block Diagram	6
PCB Layout - Component Side	7

JANUS - the Circuit

Circuit Description	8
BUS Interface and CPLD	8
Power Supply	11
ADC and Buffer Amplifiers	13
CODEC and PWM	15

JANUS - the Protocol

HPSDR - USB Data Protocol	19
---------------------------	----

JANUS - the Information

Useful Information and Links	24
------------------------------	----

Revision History

25

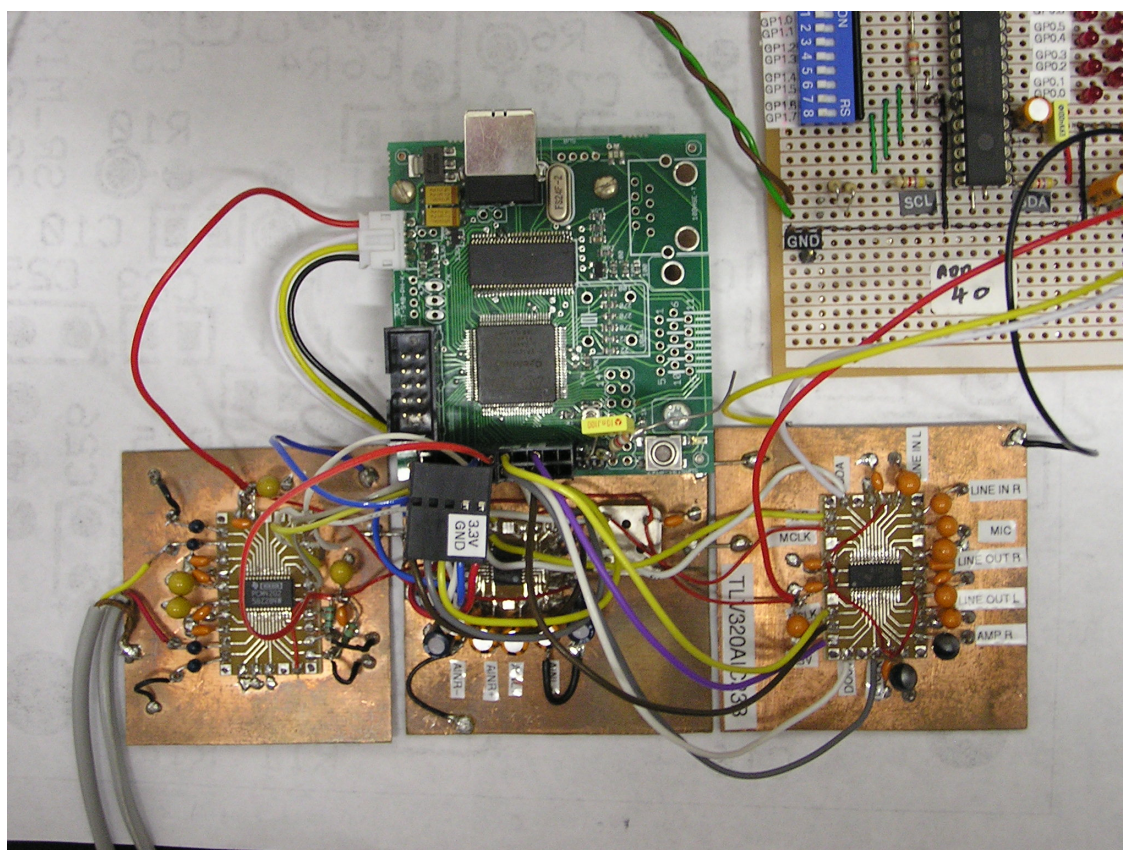
JANUS - the Audio IN-OUT

About the JANUS Module

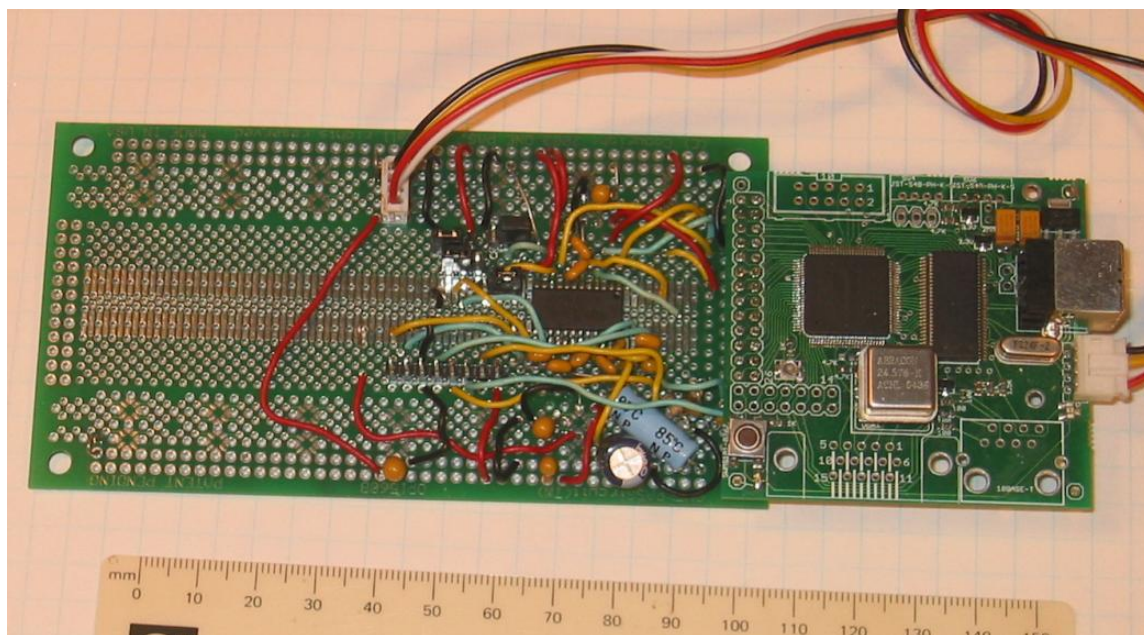
The roman god JANUS was considered to be watching heaven's gate in realtime. This gate in those times must have had an entrance as well as an exit to escape from heaven. Therefore all images show him as two-faced looking at the entrance and exit simultaneously. This mythological picture has been adopted for the JANUS board functions of handling all audio ins and outs in full duplex mode.

The whole game started with some guys buying a Xylo board in order to play with the FPGA (Field Programmable Gate Array) sitting on this board. Since these guys were in a fever of excitement for Software Defined Radio - especially the SDR 1000 - and already had contributed a lot to the free PowerSDR software which runs this radio the first application was born: We make a high performance sound card.

There are some very high-end A/D and D/A converters on the market for audio applications so the choice was not easy. An intense reviewing and prototyping started.



Phil, VK6APH, breadboarding some audio chips for evaluation



Bill, KD5TFD, breadboarding a Cirrus Logic chip

At the end of the bake-off there was a big surprise. Only one chip could meet the high demand which was put on the evaluated chips: the performance at 192 kHz sampling rate. The clear winner was an audio chip which is used in very high-end commercial audio equipment: the AK5394A from AKM.

From the HPSDRWiki:

The Janus module is a very high performance, dual, full duplex, A/D and D/A converter board. While the M-Audio Delta 44 has become the de-facto standard for A/D sound cards for use with an SDR, there are a number of advantages to rolling your own. These include having complete control of any software drivers needed to communicate with the A/D chips as well as optimization of sampling rates and bit depths for individual signals. It's also possible to cost effectively develop a board which approaches the performance of professional high end sound cards.

The consumer demand for high quality PC sound cards has resulted in the availability of a number of very high performance, and low cost, A/D and D/A converter chips that are ideal candidates for this project.

...

The results of the ADC bake-off are in and the AK5394A is the clear winner for our particular application due to its flat noise floor at 192kps.

The measured figures of the prototype Janus using the AK5394A are:

Noise Floor = -160dBm (in an 11Hz FFT bandwidth) Dynamic Range = 120dB ENOB = 20 bits

...

The selection of A/D converter for microphone and line inputs and D/A converters for audio out and I/Q signals for the transmitter was somewhat simpler. It was hard to go past the TI TLV320AIC23B. This remarkable chip contains a microphone amplifier, with bias feed for an electret microphone, stereo line in and out, stereo 16-bit A/D and D/A converters together with 35mW stereo headphone amplifier. The price for this chip is \$7 (US) in single quantities. We have a lot to thank MP3 players for!

...

The Verilog code for the Xylo board has been changed to support the AK5394A ADC in I2S Slave mode. In addition, the ADC can be switched between 192/96/48kps under control of PowerSDR. The clocking for the TLV320 has been altered so that its ADC and DAC always run at 48kHz. The TLV320 code has also been altered to enable it to run in I2S mode. The clocks for the ADC and DAC are now obtained by dividing down the 24.576MHz rather than running the AK5394A in Master mode.

Bill, KD5TFD, has modified PowerSDR so that it can accept ADC data from the AK5394A at 192/96/48kps and send received DAC data back to the TLV320 at 48kHz. He has also modified the code to up-sample the 48kHz microphone/line-in data from the TLV320 to 192/96kps.

Bill's modifications to the PowerSDR software provides 3 A/D inputs (I, Q, and line/microphone) and 4 outputs (left/right receiver audio and I/Q audio for the exciter). These are both full duplex so VOX etc. operation works just fine. He has also added PTT and CW inputs to the FPGA and these appear to provide very low latency inputs to PowerSDR.

The document that describes the protocol used over the USB to/from the PC will be updated shortly and posted on hamsdr.com

Phil...VK6APH

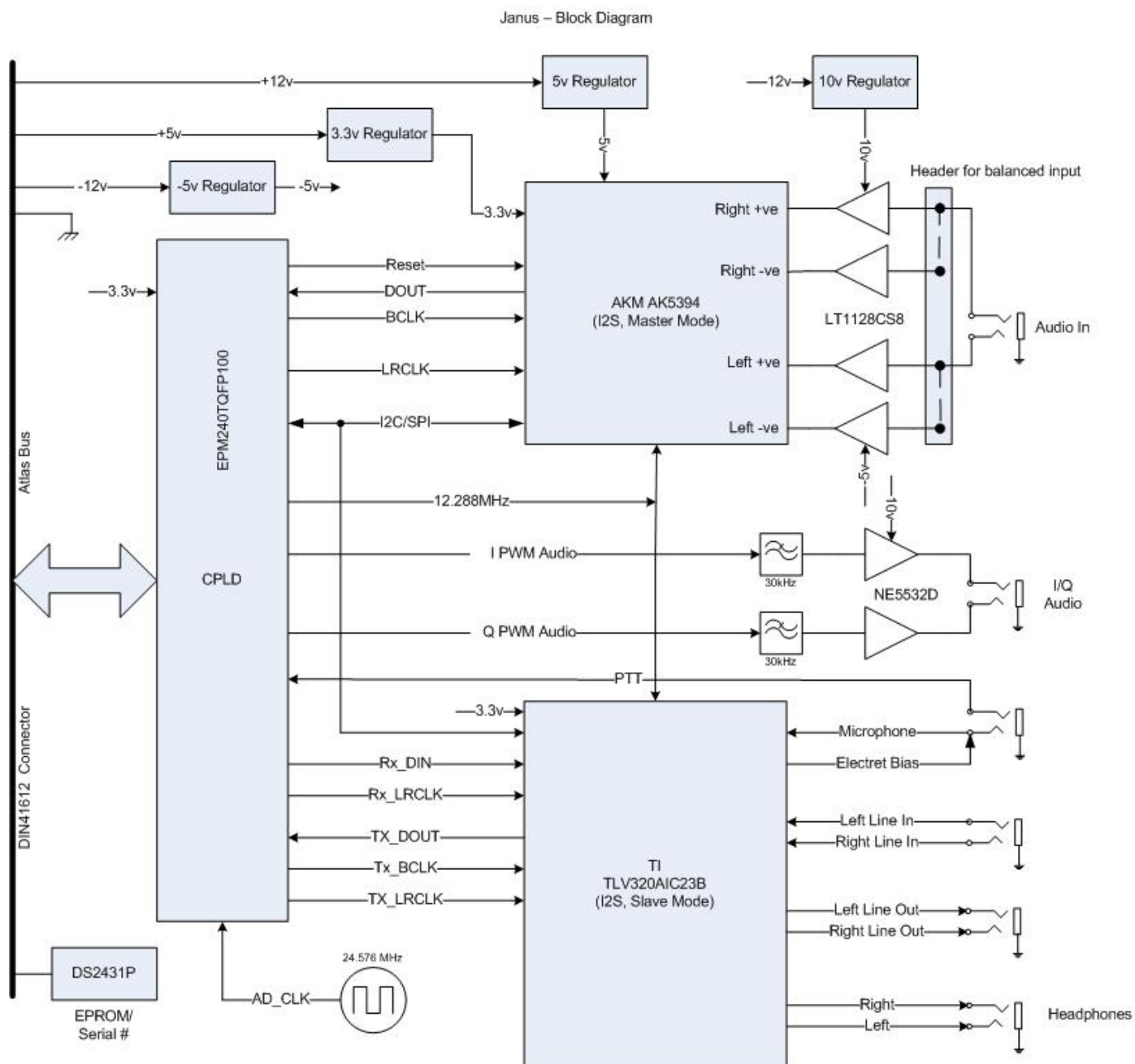
The PCB for the Janus board was laid out by Lyle, KK7P. The latest version of the schematic may be found at <http://www.hamsdr.com/personaldirectory.aspx?id=276> and the PCB layout is at <http://www.hamsdr.com/personaldirectory.aspx?id=275> for viewing or download.

Note that the PCB follows Phil Covington's 120mm x 100mm suggestion. Minor edits have been made to the schematic to incorporate the latest changes from Phil Harman.

You will note that copyrights have been asserted to both the schematic and the PCB layout. This is to protect the content while we figure out how to make this open source. Some of us have had the unpleasant experience in the past of a third party taking our work, claiming it as their own, and then demanding compensation! Note that there is no guarantee that this design does not infringe on someone's patents, designs or claims; use it at your own risk.

JANUS - the Board

Block Diagram



JANUS - the Circuit

Circuit Description

This description applies to the Rev XB.10 schematic, released for review on May 29, 2006.

The latest schematic can be found at <http://www.hamsdr.com/personaldirectory.aspx?Id=276>

BUS Interface and CPLD

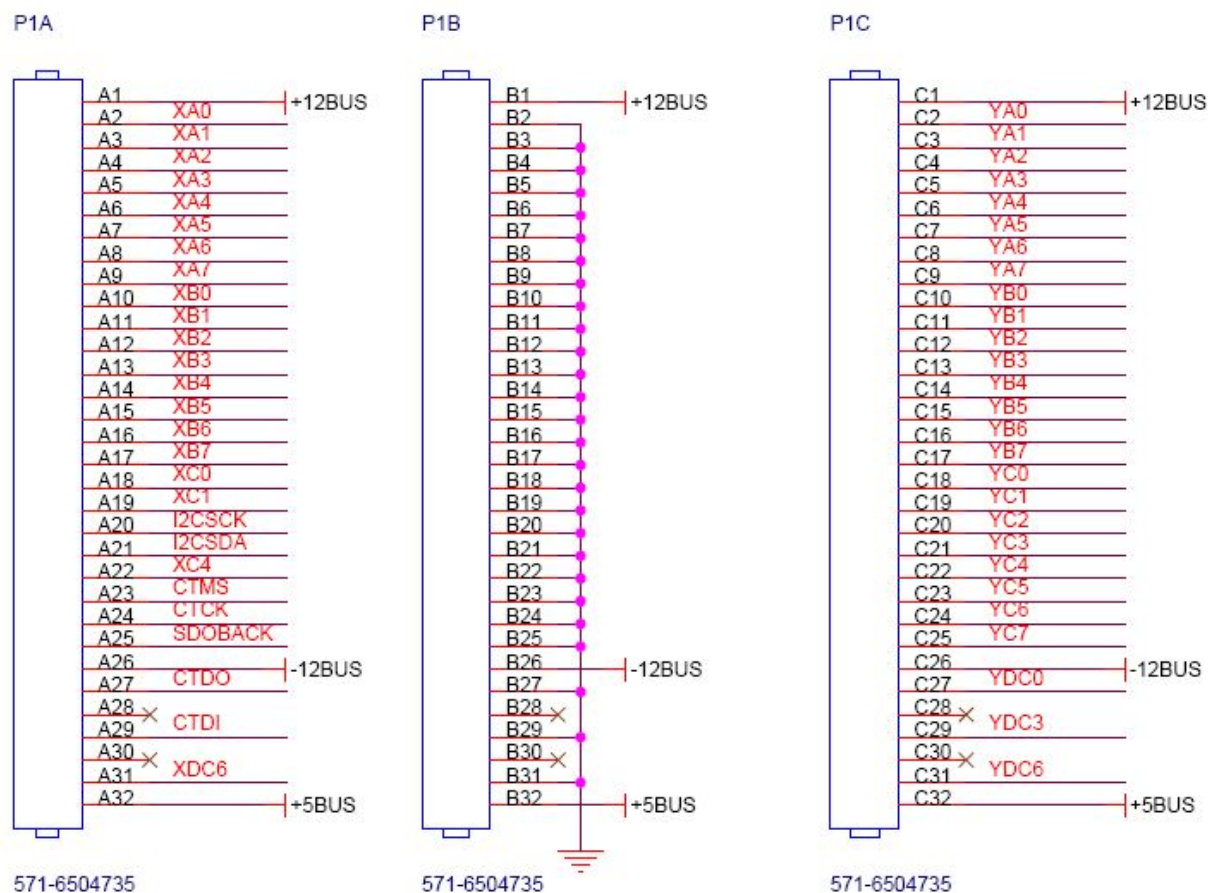


Fig. 1

The 96-pin DIN connector which plugs into the ATLAS backplane is shown in **Fig. 1**. It is labeled P1A...P1C. The interconnection with the OZYMANDIAS control board can be found in the ATLAS documentation on pages 8-9.



Fig. 2

U14 provides a unique 64-bit serial number as well as 128 bytes of EEPROM on a Dallas 1-wire bus (**Fig. 2**) which runs on pins A18 of the ATLAS backplane. This is to support remote identification and configuration of JANUS, for example by OZYMANDIAS. Link to the DS2431P datasheet: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/4272

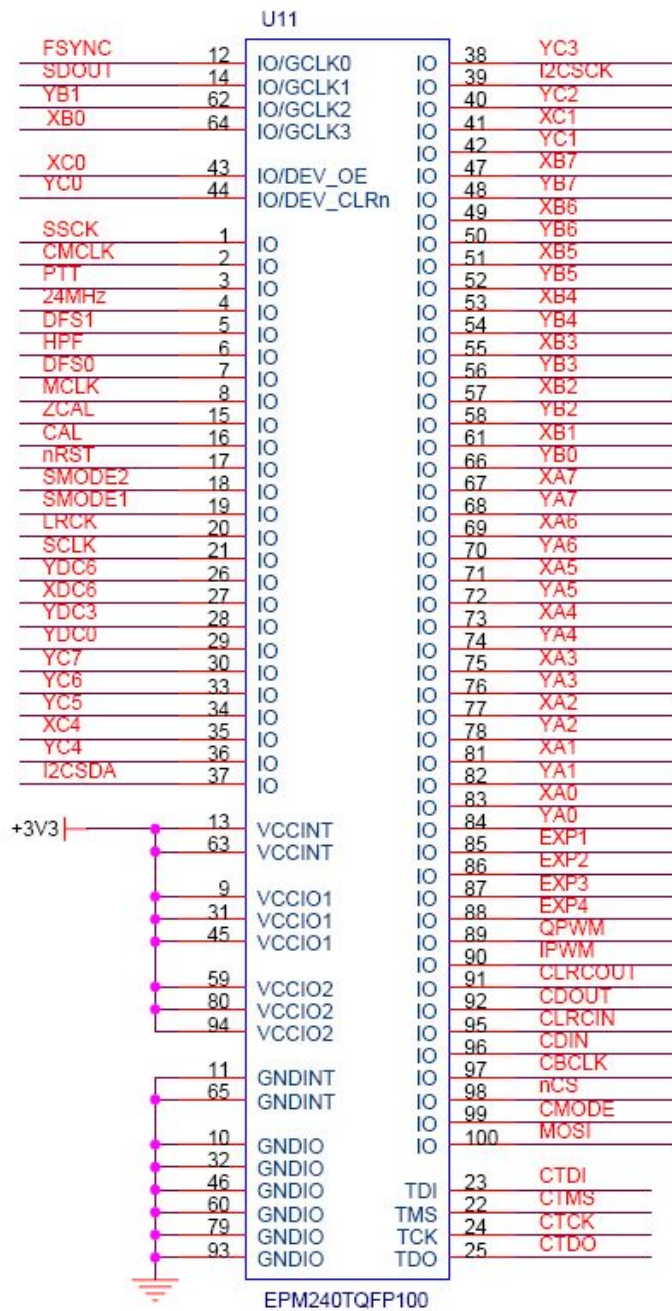


Fig. 3

U11 is an Altera EPM240 CPLD (**Fig. 3**). This is actually a small, SRAM-based FPGA similar to the Cyclone series. It has an on-chip flash configuration memory. U11 is used to map the ATLAS buses to the ADC, CODEC and PWM on the JANUS board.

U11 also provides clocks to the ADC and CODEC, derived from XO1 (**Fig. 4**), or accepts an external clock if multiple JANUS boards are installed.

Four unused pins on the device are brought to connector J7 (**Fig. 5**) to assist in "blue wire" modifications to the board.

These lines should not be brought off the board without proper bypassing and protection.

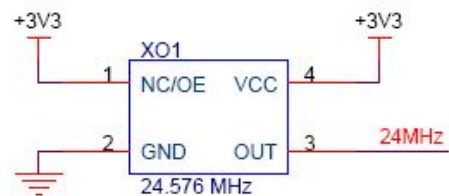


Fig. 4

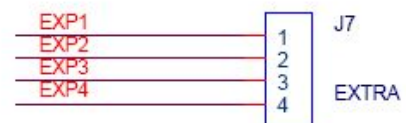


Fig. 5

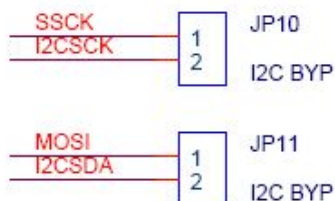


Fig. 6

An I²C interface is provided to configure the CODEC as well as to the CPLD if an I²C function is programmed into it.

JP10 and JP11 (**Fig. 6**) may be placed to directly connect the CODEC to the ATLAS I²C bus pins.

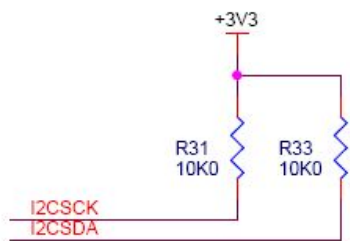


Fig. 7

R31 and R33 (**Fig. 7**) are local pull ups to enable to I²C open-collector logic.

U11 is used to map the ATLAS bus to the ADC, CODEC and PWM on JANUS (**Fig. 8**).

P2 (**Fig. 9**) allows local configuration of U11 by a standard Altera-compatible JTAG programmer. U11 may also be programmed over the ATLAS bus by another device, such as OZYMANDIAS.

For the latter, JP12 must be placed if JANUS is the last card from the master.

There must be no empty slots between OZYMANDIAS and JANUS for this to work, and OZYMANDIAS must be “upstream” from Janus.

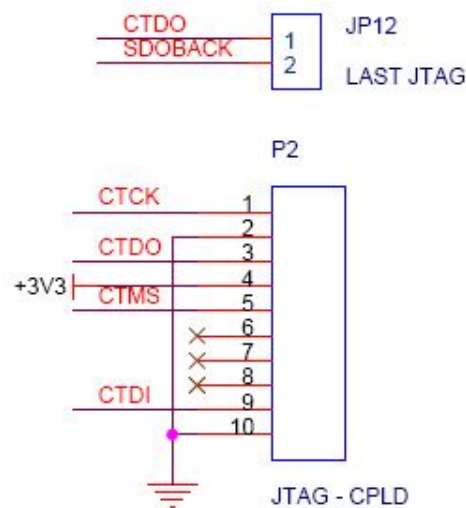


Fig. 9

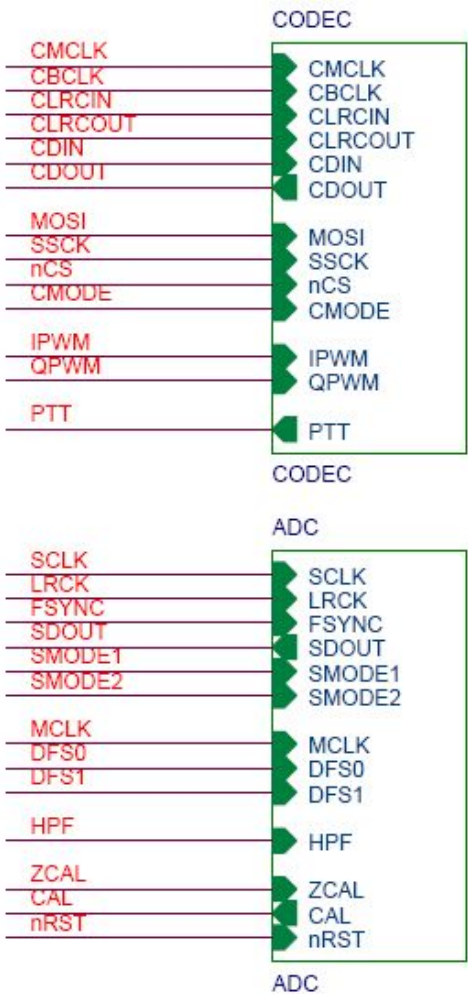


Fig. 8

If all slots are filled, then the “upstream” requirement is no longer in force.

Power Supply

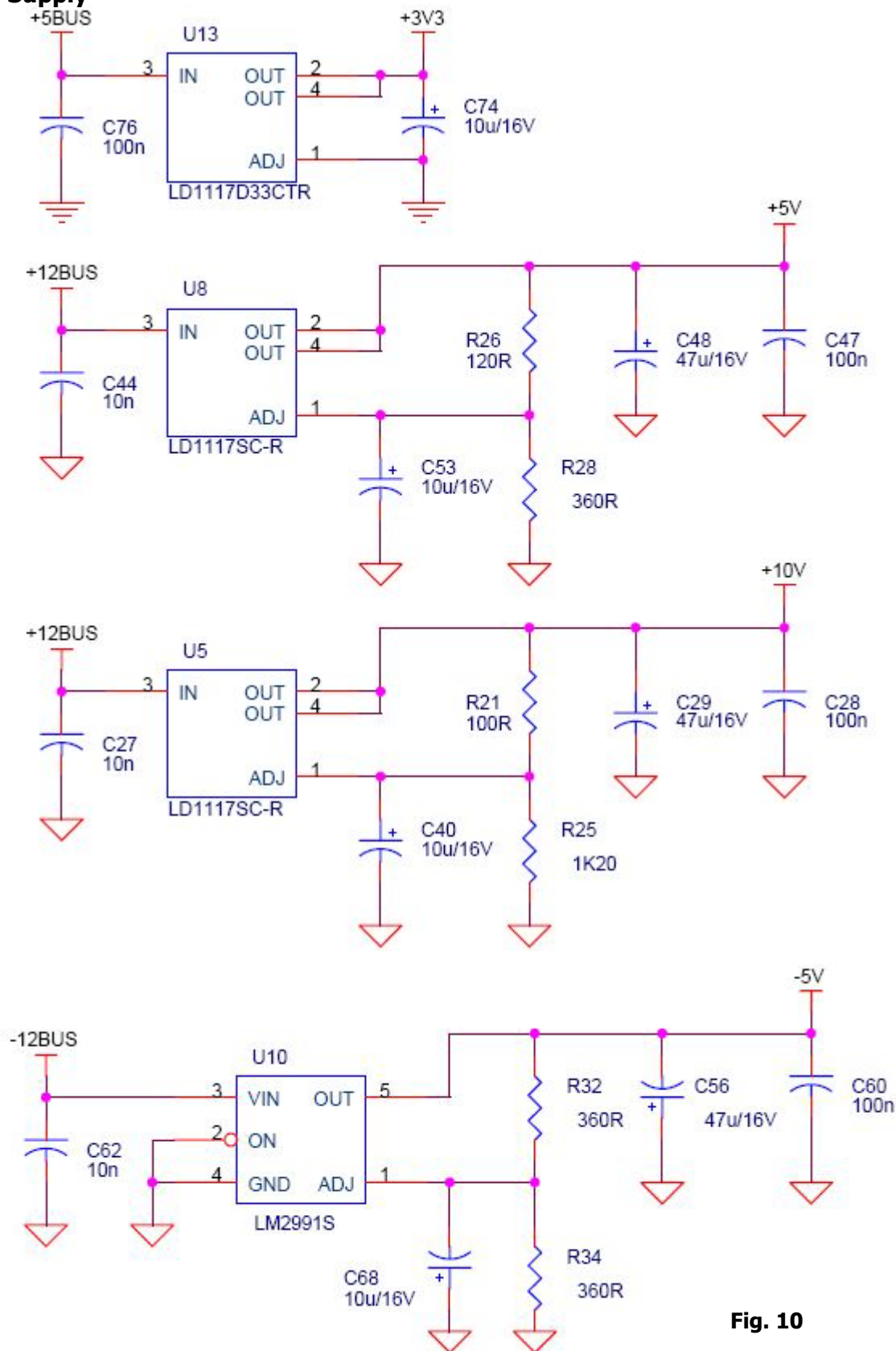
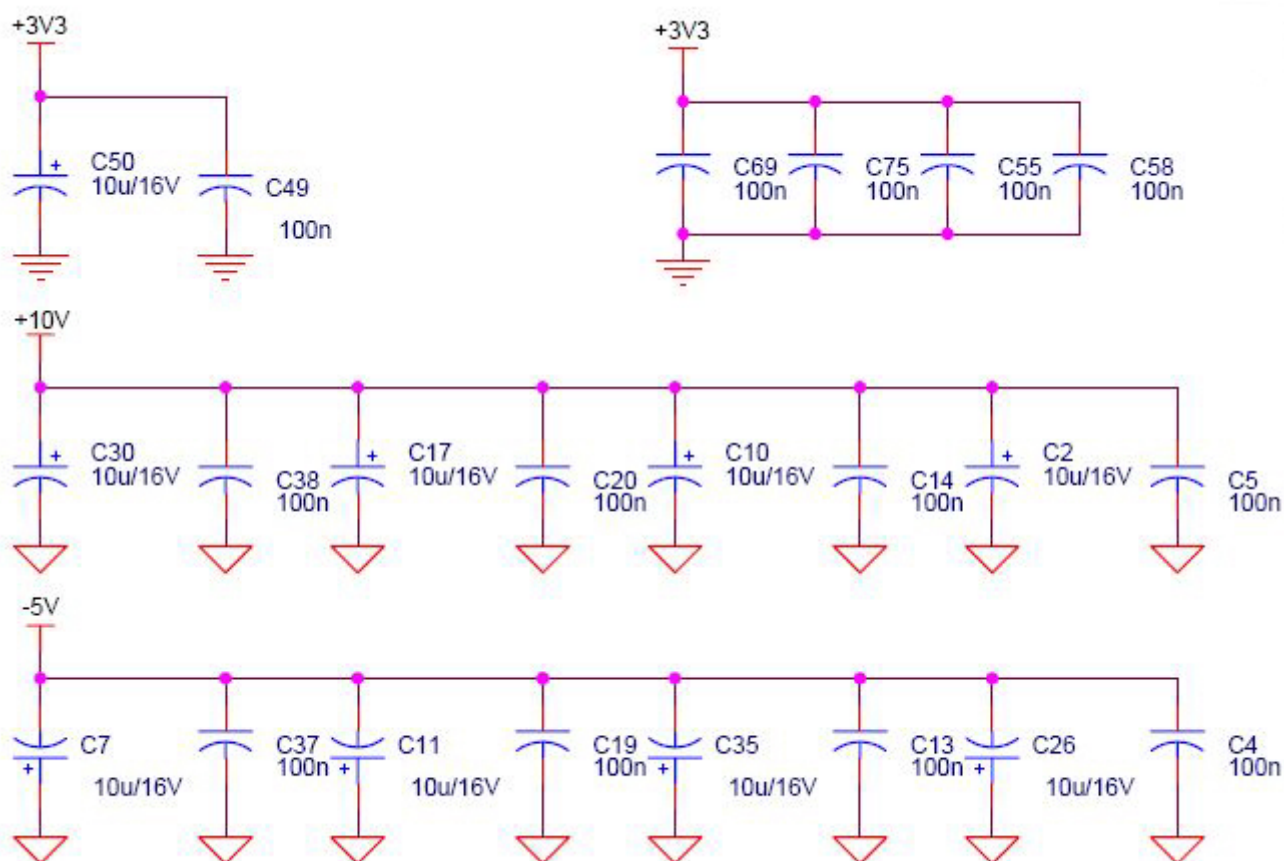


Fig. 10

**Fig. 11**

U13 and associated bypass capacitors provide +3V3 (**Fig. 10 and 11**). This is the primary digital power supply, as well as the CODEC analog supply.

U8 (**Fig. 10**) provides +5V which is used by the AKM AK5394A ADC.

U5 (**Fig. 10**) and associated bypass capacitors (**Fig. 11**) provide +10V, used in the ADC, PWM and CODEC analog buffer amplifiers.

U10 (**Fig. 10**) and associated bypass capacitors (**Fig. 11**) provide -5V, used by the ADC analog buffer amplifiers.

U2 (**Fig. 12 on next page**) provides bias voltage for the ADC input buffer amplifiers and R34 (**Fig. 10**) is selected to match the buffer amplifier offset to the ADC inputs.

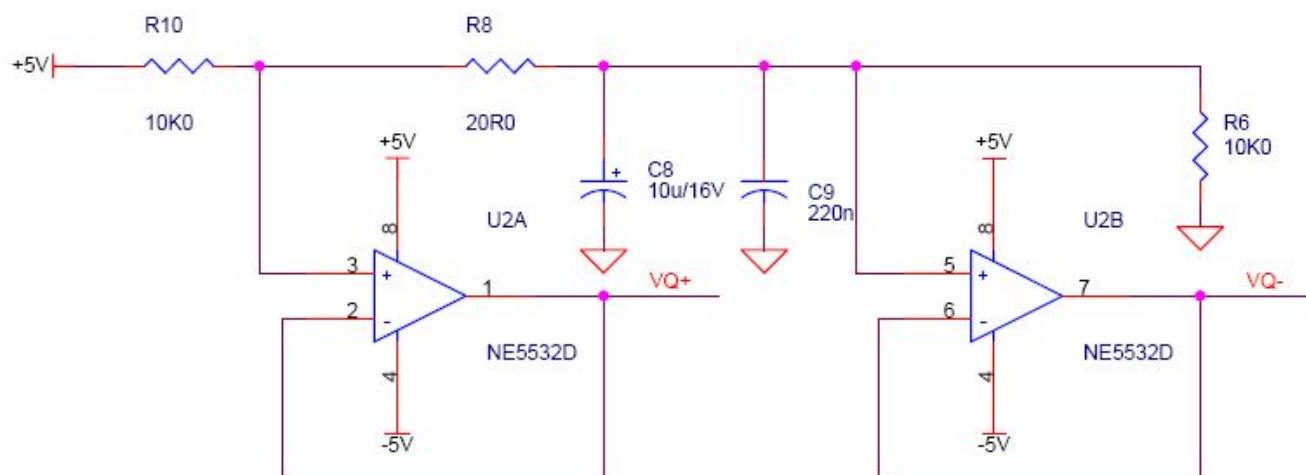


Fig. 12

ADC and Buffer Amplifiers

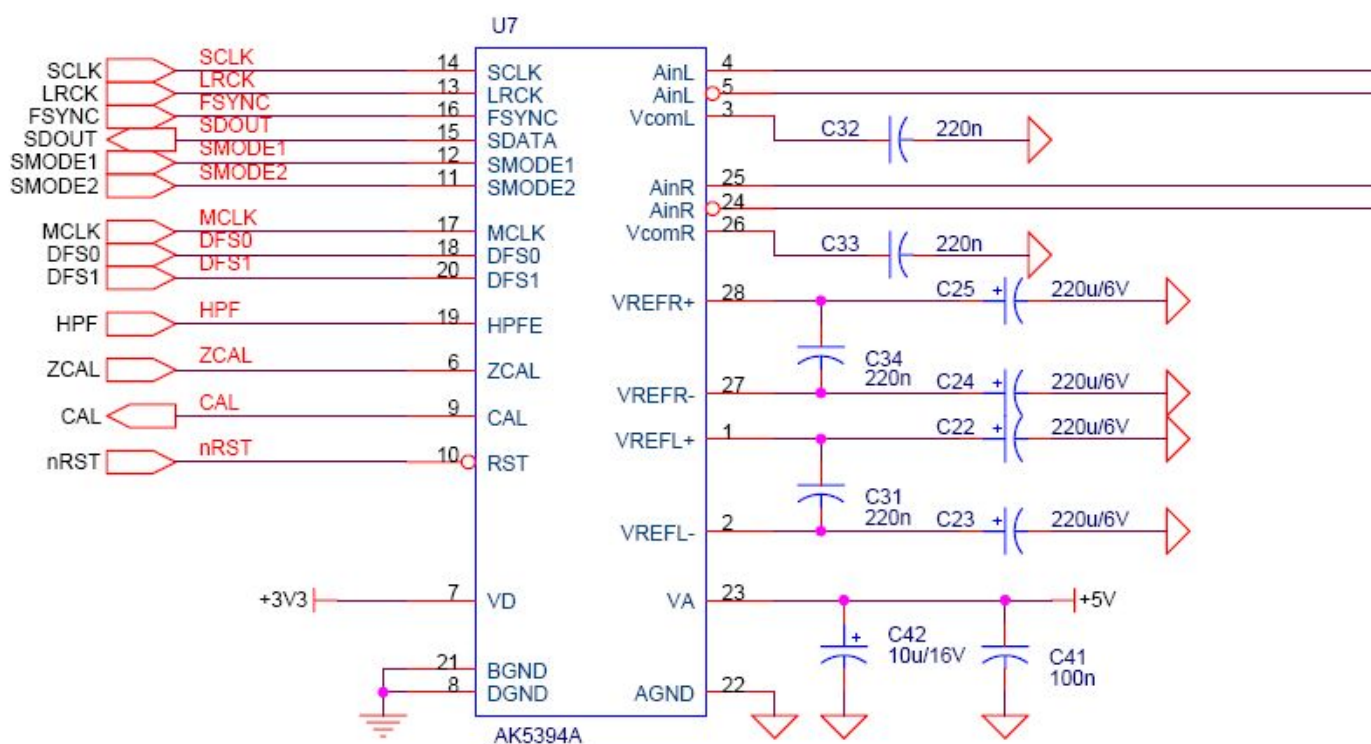


Fig. 13

U7 is the AKM AK5394A ADC chip (**Fig. 13**). The datasheet can be viewed at <http://www.asahi-kasei.co.jp/akm/en/product/ak5394a/ak5394a.html>

This is a 2-channel, 24-bit component with excellent specifications and capable of operation beyond 200 kilosamples/second. Significant bypassing is required on various pins of this device.

There are two channels of input amplifier (**Fig. 14a and 14b**), the left is discussed, the right is similar.

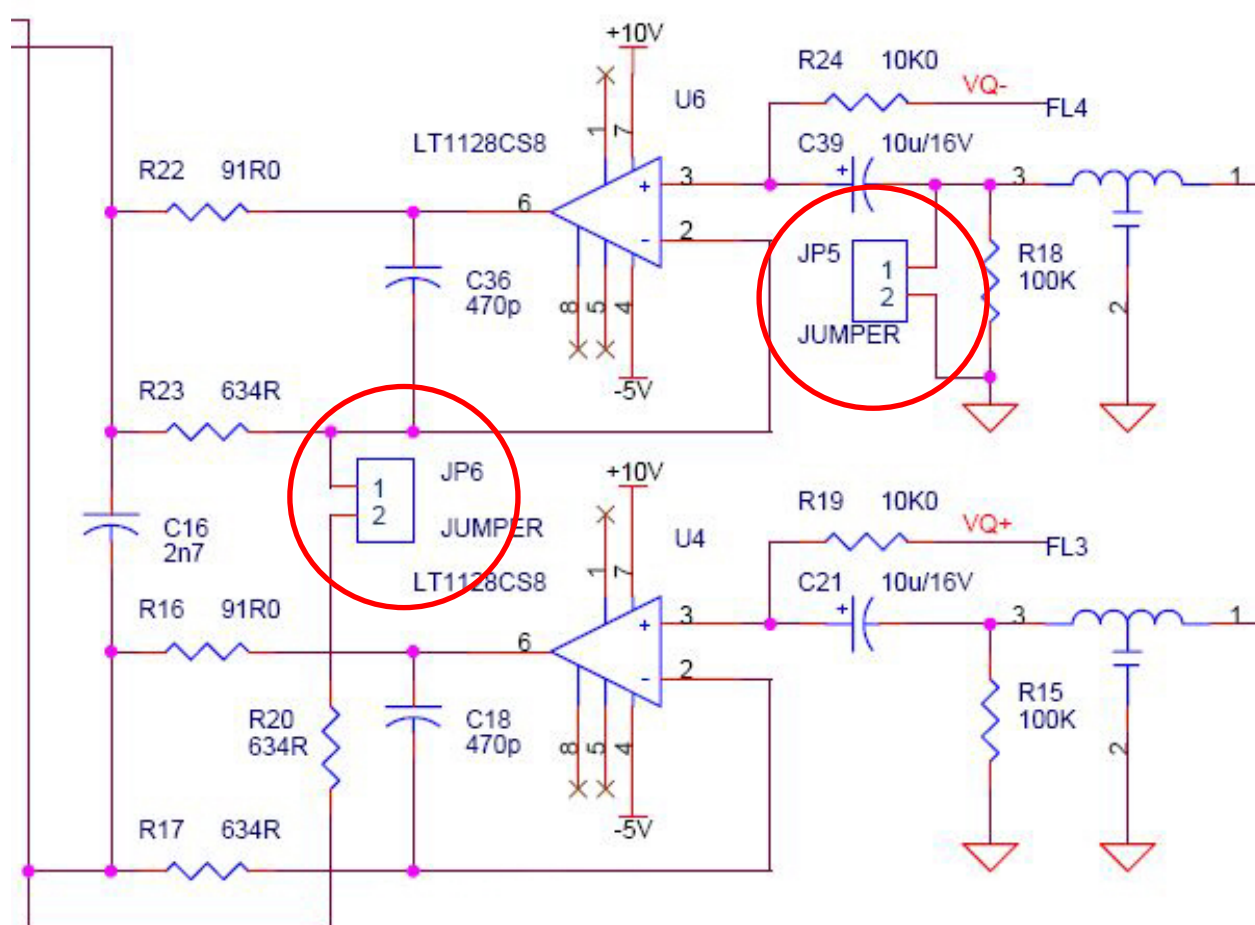


Fig. 14a

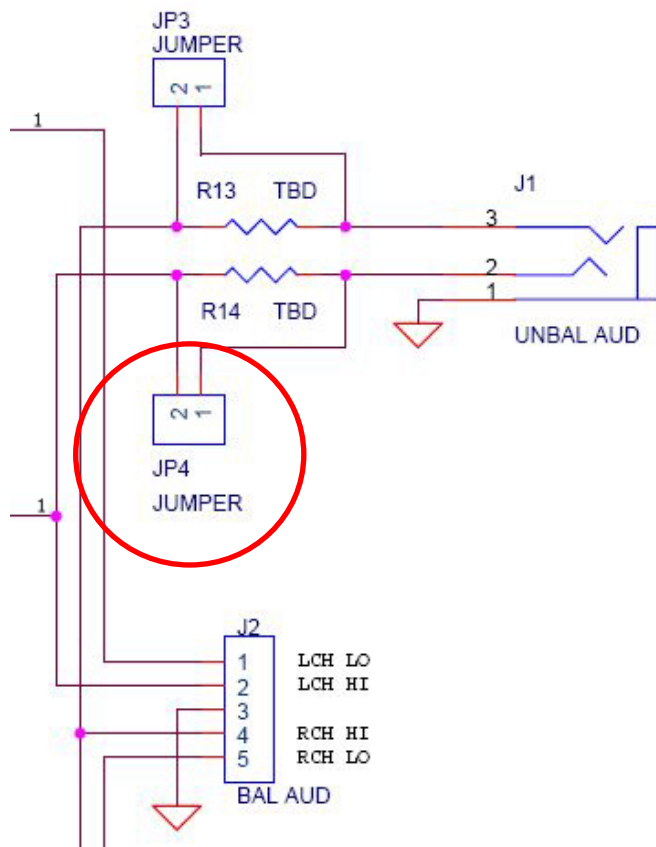


Fig. 14b

Assuming a balanced input applied to J1 (**Fig. 14b**), one side is coupled through JP4 (normally placed) and FL3 to U4 (**Fig. 14a**). FL3 helps to isolate noise on the input, perhaps stray RF. R19 (**Fig. 14a**) introduces the bias, C21 couples the audio signal. U4 (**Fig. 14a**) buffers the signal, with R16, R17 and C18 arranged to maximize dynamic range and minimize noise. The other side of the balanced input signal is similarly buffered by U6 (**Fig. 14a**).

This input configuration was chosen for high input impedance to minimize loading on a QSD-style detector that might drive JANUS. This circuit is not compatible with "professional audio" signal levels or impedance.

U6/U4 outputs are bypassed by C1, working in conjunction with R16/R17/R22/R23 to provide

rolloff for the sigma-delta ADC convertor within U7.

If an unbalanced signal is required, the jumpers JP5 and JP6 are placed (**Fig. 14a**). The input signal is applied to U4, and the output from U4 is coupled to the inverting input of U6 through R20/JP6 so that U6 provides inverting, unity gain. The non-inverting input of U6 is shunted by JP5.

If a higher level signal input tolerance is required, JP4 (**Fig. 14b**) is removed and R14 reduces the input amplitude, working against the nominal input impedance of the buffer amplifier.

CODEC and PWM

U9 is a TI TLV320AIC23B 2-channel CODEC (Coder/DECoder) (**Fig. 15**) which has ADC and DAC functions.

The datasheet can be viewed at

<http://focus.ti.com/docs/prod/folders/print/tlv320aic23b.html>

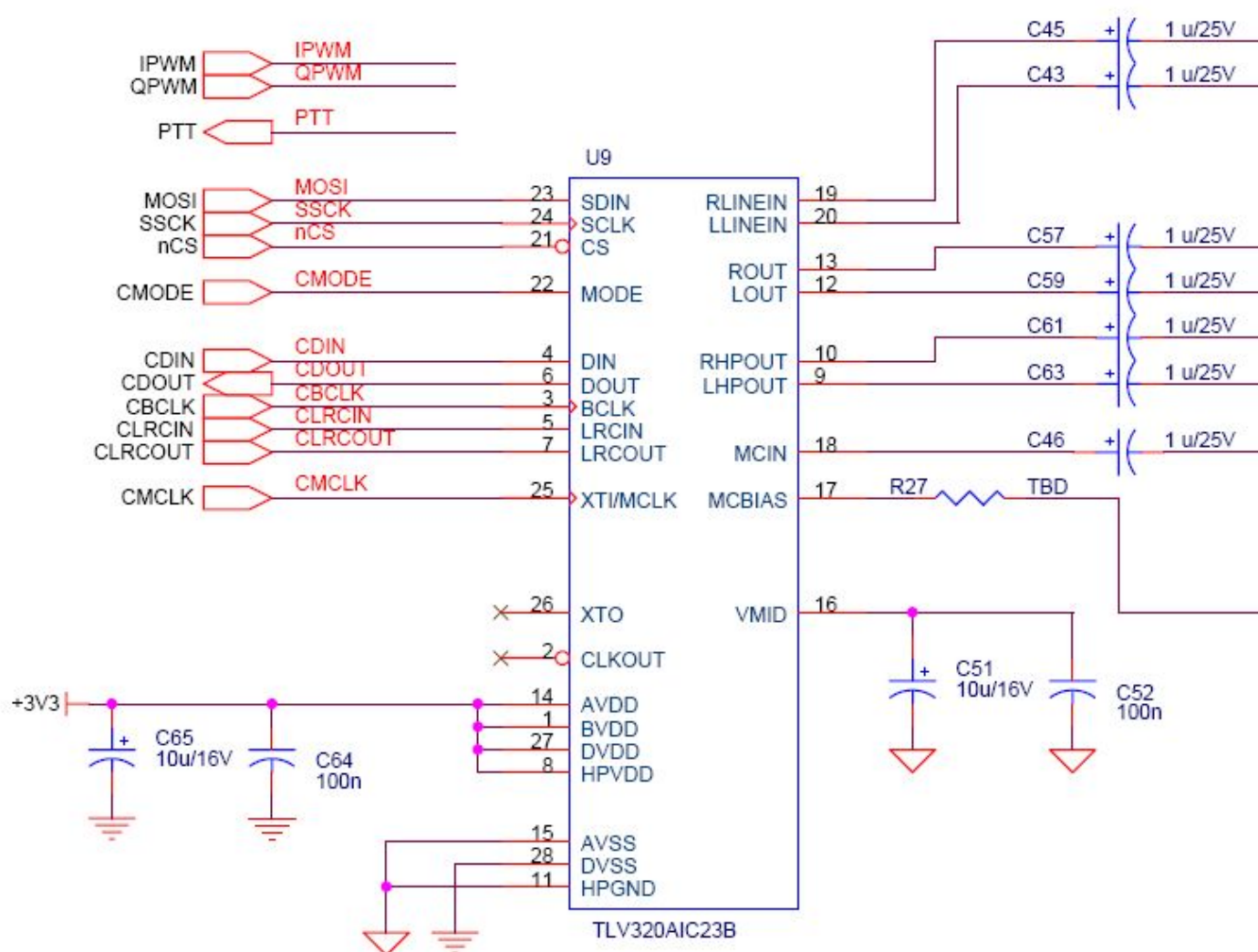


Fig. 15

It is fully configured by the FPGA. Signals are bypassed for RF and AC coupled to the analog inputs or from the analog outputs of U9. U9 is intended to provide "baseband" audio I/O for the HPSDR. Note that no speaker amplifier is provided on JANUS.

Just as 8-pin microphone connectors adhere to no standards in Amateur Radio, simple 3.5 mm microphone connectors are also "standards free".

JP7...JP9 (**Fig. 16**) allow configuration of the microphone connector. JP7 may be placed to apply DC bias to the tip or ring of the MIC connector, or left off if no bias is required (e.g., a dynamic mic).

JP8 selects the tip or ring for PTT, or may be left off if the MIC has no PTT function.

JP9 selects the tip or ring for MIC audio. This jumper must be placed.

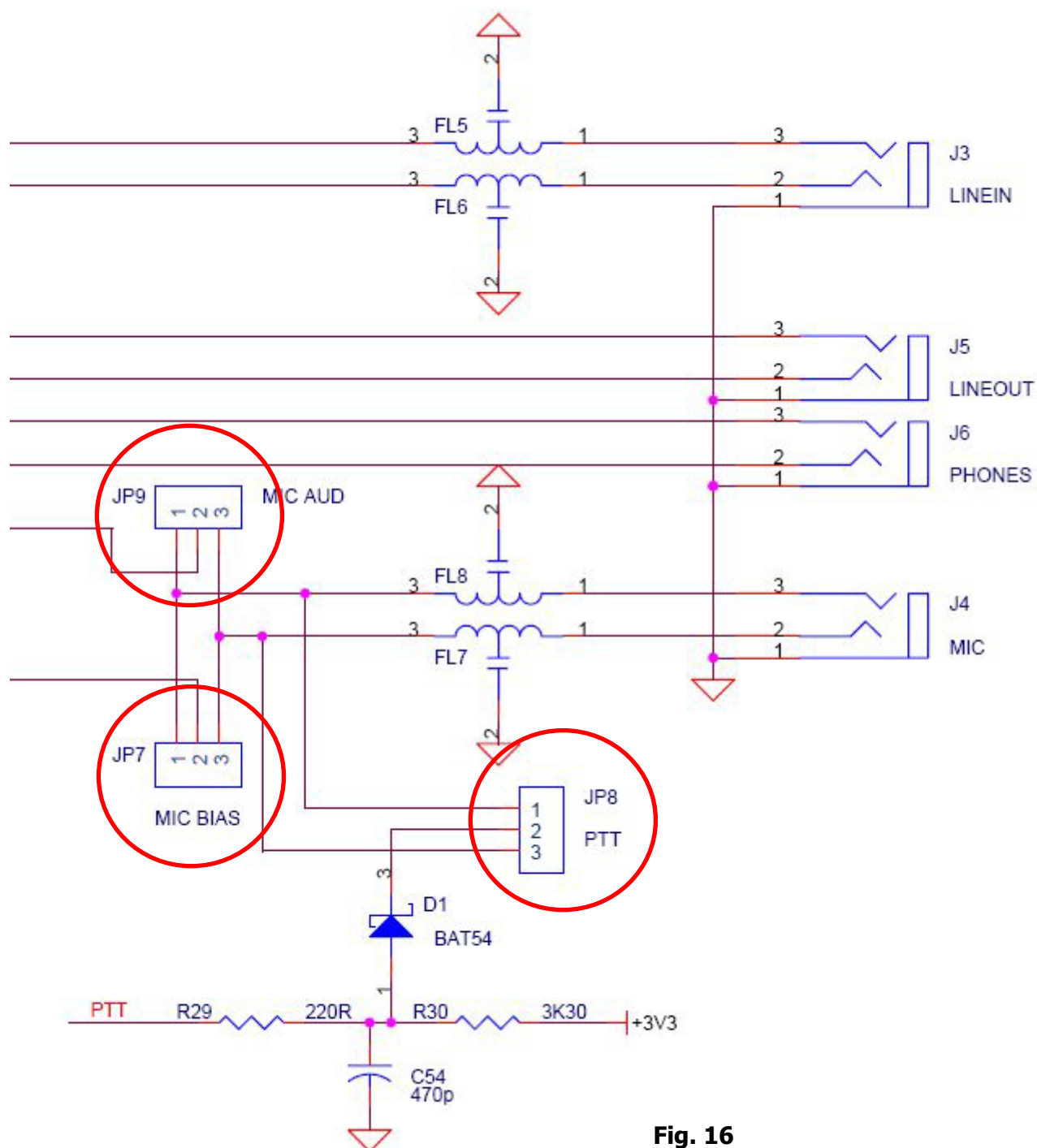


Fig. 16

The PTT signal is pulled high by R30, and the PTT assertion (pulled to ground) is coupled by D1. R29 and C54 provide further decoupling of the PTT input.

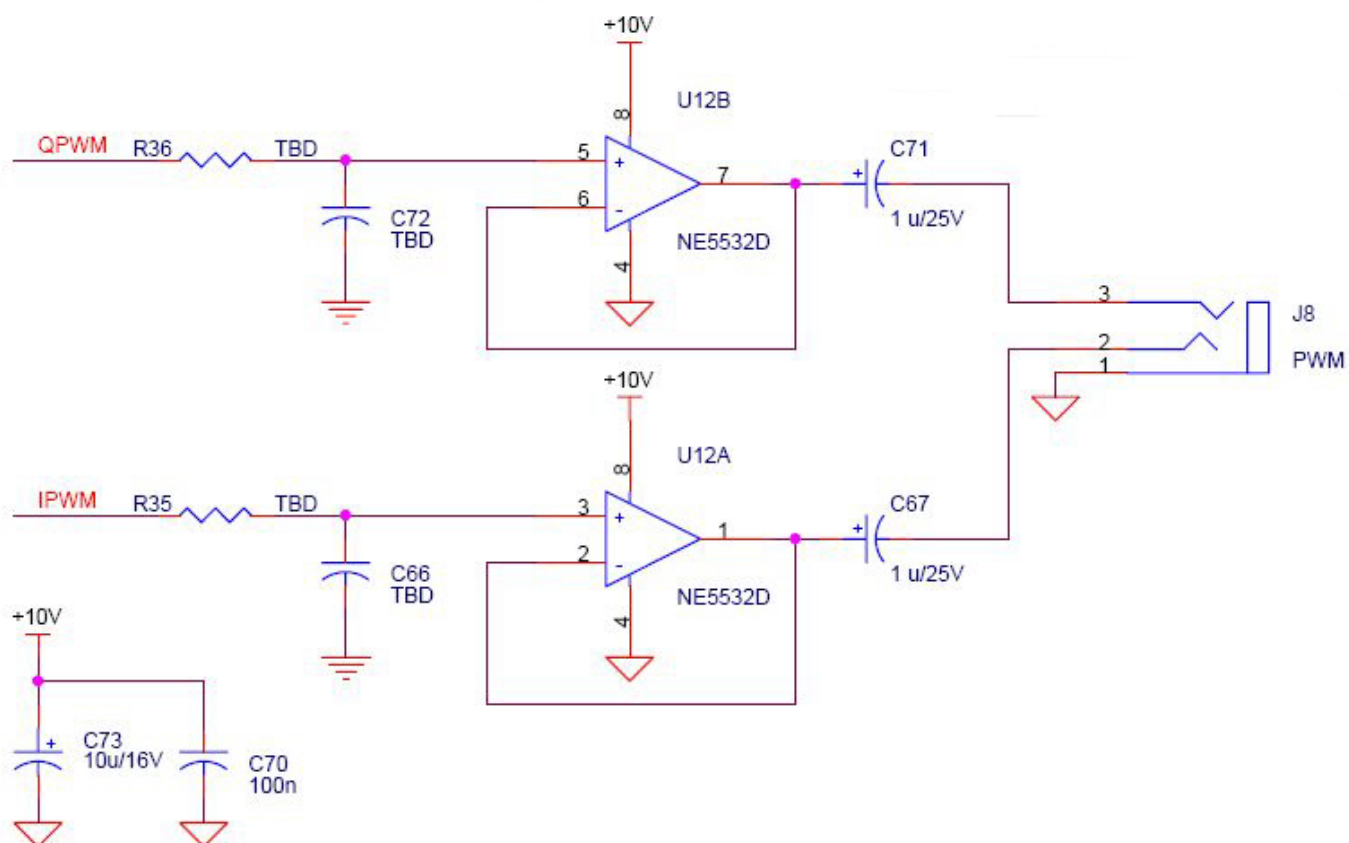


Fig. 17

Finally, U12A and U12B (**Fig. 17**) provide integration and buffering of the PWM signals from OZYMANDIAS or some other source on the ATLAS backplane. These outputs are intended to drive a QSE mixer for transmission.

JANUS - the Protocol

HPSDR - USB Data Protocol

© Phil Harman VK6APH

All rights reserved

Revisions

Revision	Date	Changes	By
1.0	May 27, 2006	Initial document	VK6APH

Assumptions:

- The USB data will consist of 512 byte Frames
- The sample rate from the receiver A/D converter to the PC will be selectable between 48/96/192kHz at 24 bits
- The sample rate from the microphone to the PC will be 48KHz and 16 bits
- The sample rate from the PC to the speakers/headphones will be 48kHz and 16 bits
- The sample rate from the PC to the I/Q transmit audio will be 48kHz and 16 bits
- Control signals that are high priority will be sent each Frame, lower priority data will be sent less frequently
- I²C data will be sent via a separate pipe and the FX2 chip will provide the I²C Master

Functions required:

- PTT in put active or activated by PowerSDR
- Dot/dash key active
- A/D sampling speed 192/96/48k command
- Forward and Reflected power data
- DDS/NCO tuning phase data
- X2 relay output delay
- User outputs (8)
- User inputs (8)
- GPS based frequency measurement
- SDR1000 parallel port interface

Sync Sequence

A three byte sync sequence consisting of **<0x80><0x00><0x00>** is used. This value represents the most negative value from the 24 bit A/D converter and will only be obtained when the A/D is approaching, or being, overloaded.

If we assume that the signal from the A/D converter is essentially random then the chance of the sync sequence occurring in this data is 1 in 2²⁴ i.e. 1 in ~16 million. Since the maximum

rate that data can be sent over the USB-2 interface is once every 250 μ S then the probability of a false sync decode is approximately once in every 6 seconds. We can reduce this probability by checking that the sync sequence occurs every n bytes (512 in this case) and that the A/D converter is never driven to overload.

Protocol

The protocol consists of a 512 byte frame consisting of a sync sequence, control data and A/D or D/A data.

A frame length of 512 bytes is used since this is the maximum number of bytes that the FIFO in the FX2 USB interface on the Xylo board can hold. (NOTE: This may change when using the OZY board).

High priority control data will be sent as part of each frame e.g. PTT command/request. Lower priority data will be sent as available on a predefined schedule e.g frequency measurements or NCO/DDS output frequency.

Note that ideally the control data should be formatted such that it is not able to simulate a sync sequence.

Protocol – From HPSDR to PC

Since the receive A/D converter will use 24 bits per sample and the Microphone/Line A/D 16 bits per sample then, in order that an integer number of I/Q samples will be included in the 512 byte packed, the maximum number of samples is

$(512 - 8) = 63 \times (3 + 3 + 2)$ bytes i.e. 63 24 bit I/Q samples and 63 16 bit Mic/Line samples

This provides 8 bytes to transfer status data from the FPGA to the PC. The first characters in the 512 byte packet will be sync which is 0x800000. Since 3 bytes are used for the sync character then 5 bytes (C0 to C4) may be used to send status to the PC.

0																16		
80	00	00	C0	C1	C2	C3	C4	I	I	I	Q	Q	Q	M	M			
SYNC			Command and Control															
17																		511
I	I	I	Q	Q	Q	M	M	I	I	I	Q	Q	Q	M	M	

Where Cn is a control byte and

I	Bits 23 – 16 of I sample
I	Bits 15 – 8 of I sample
I	Bits 7 – 0 of I sample
Q	Bits 23 – 16 of Q sample
Q	Bits 15 – 8 of Q sample
Q	Bits 7 – 0 of Q sample
M	Bits 15 – 8 of Mic/Line sample
M	Bits 7 – 0 of Mic/Line sample

Control Bytes

- C0 - bit 0 = 1 PTT/dot active, bit 1 = 1 dash active
 - bits 7-2 describes meaning of C1, C2, C3 and C4 as follows:

C0 bits 7-2	C1	C2	C3	C4
0	'TxFIFO full' counter (debug)	'Rx wait' counter (debug)	'Sequence #' (debug?)	'# bytes in Rx FIFO' (debug?)
1	MSB Fwd Pwr	LSB Fwd Pwr	MSB Rev Pwr	LSB Rev Pwr
2	SDR 1000 Parallel Port (data in)	SDR 1000 Parallel Port (data in)		
3	MSB Freq ¹	Freq	Freq	LSB Freq
4	User Inputs			
5				
...				
63				

¹ Since 32 bits are available a frequency of up to 4GHz can be measured to a resolution of 1Hz.

Protocol – From PC to HPSPDR

The PC sends two audio streams to the HPSPDR. These are

1. 48kHz 16 bit Left/Right received audio
2. 48kHz 16 bit I/Q audio

Since the received audio may also be used to monitor the transmitted audio then these two streams must be available simultaneously.

Since the D/As will use 16 bits per sample then, in order that an integer number of Left/Right and I/Q samples will be included in the 512 byte Frame, the maximum number of samples is

$$(512 - 8) = 63 \times 4 \times 2 \text{ bytes i.e. } 63 \text{ Receiver L/R samples and } 63 \text{ I/Q L/R samples}$$

This provides 8 bytes to transfer status data from the PC to the FPGA. The first characters in the 512 byte packet will be sync which is 0x800000. Since 3 bytes are required for the sync

0																			
80	00	00	C0	C1	C2	C3	C4	L	L	R	R	I	I	Q	Q				
SYNC			Command and Control																
16																			511
L	L	R	R	I	I	Q	Q	L	L	R	R	I	I	Q	Q

W	L	Bits 15 - 8 of Left Rx audio sample
	L	Bits 7 - 0 of Left Rx audio sample
	R	Bits 15 - 8 of Right Rx audio sample
	R	Bits 7 - 0 of Right Rx audio sample
	I	Bits 15 - 8 of I Tx audio sample
	I	Bits 7 - 0 of I Tx audio sample
	Q	Bits 15 - 8 of Q Tx audio sample
	Q	Bits 7 - 0 of Q Tx audio sample

Control Bytes

C0 - bit 0 = 1 MOX active in PowerSDR

C0 7-1	C1	C2	C3	C4																																
0	Bits 1 and 0 encode A/D speed <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td></tr><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>1</td></tr><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>0</td></tr></table> X = unsigned at present	7	6	5	4	3	2	1	0	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0	1	x	x	x	x	x	x	1	0	X2 output delay	User output	
7	6	5	4	3	2	1	0																													
x	x	x	x	x	x	0	0																													
x	x	x	x	x	x	0	1																													
x	x	x	x	x	x	1	0																													
1	NCO/DDS MSB	NCO/DDS	NCO/DDS	NCO/DDS LSB																																
2	SDR 1000 Parallel Port (data out)	SDR 1000 Parallel Port (data out)																																		
3																																				
4																																				
5																																				
...																																				
128																																				

C0 7-1C1 C2 C3 C4
0 Bits 1 and 0 encode A/D speed

JANUS - the Information

Useful Information and Links

Project Description and Information

<http://hpsdr.org>

http://hpsdr.org/wiki/index.php?title=HpsdrWiki:Community_Portal

<http://www.hamsdr.com> (requires registration for full information access)

Discussion List / Reflector

The HPSDR Discussion List (also known as a "reflector") is the major method of intercommunication between all interested persons of this project. At times the number of messages can get large -- other times it may go a day or two without a message. Anyone can view the message traffic in the list archive online. It can be found at

<http://lists.hpsdr.org/pipermail/hpsdr-hpsdr.org/>

Parts Kits and Boards

The TAPR Corporation is distributing parts kits as well as printed circuit boards for the HPSDR project.

TAPR Corporation <http://www.tapr.org>

Revision History

Revision	Date	Changes	Initiator
1.1	June 18, 2006	Cover page changed Schematic description modified	DL6KBF
1.0	June 10, 2006	Initial publication	DL6KBF