

NeptuneSDR

Zyn Q 7020 AD9363 Development

Platform K230 User Manual



Catalogue

Copyright Statement.....	3
disclaimer.....	4
Declaration of China Micro-Power Short-range Device.....	5
I. Introduction to the Development Board.....	7
II. ZYNQ Chip.....	9
III. DDR Chip.....	10
4 、 QSPIFlash.....	10
V. PS System Clock Source.....	11
VI. Structural Diagram.....	12
VII. BTB Connector Pin Definition.....	12
8. USB 2.0 Interface.....	13
IX. Gigabit Ethernet Interface.....	13
10. UART Interface.....	14
11. SD card slot.....	14
12. AD9363.....	15

Copyright Statement

Nep Technology holds the copyright to this document. Without formal written authorization from Nep Technology, no company, organization, website, media outlet, or individual may modify, distribute, copy, or excerpt any content of this document in any manner or form. Such content includes, but is not limited to, text, images, charts, tables, and other materials. The intellectual property rights of this document belong to Nep Technology, and no individual or organization may use it for commercial purposes through any channel. Nep Technology reserves the right to pursue legal action against infringers.

We place great importance on the intellectual property rights of this document, reflecting our respect for intellectual property laws and for creators. We hope all users of this document will respect these rights and comply with the aforementioned statements to avoid unnecessary legal disputes and risks arising from infringement. At the same time, we encourage everyone to respect their own intellectual property rights and create more valuable works.

Through our efforts, we hope to enhance public awareness and respect for intellectual property rights, jointly safeguard their legitimate interests, and ensure better protection and development of intellectual property. Thank you for your cooperation and support!

Disclaimer

This product is for research and educational purposes only. The purchaser shall use it strictly within legal and regulatory boundaries. Any disputes arising from the use of this product and all resulting consequences shall be borne solely by the purchaser, with Neop Technology assuming no liability. This is hereby declared.

Nep Technology provides no warranties of any kind—whether express or implied—including but not limited to warranties of suitability for specific purposes—for the code, software, documentation, or other materials included in this product. Users assume all risks. Recognizing the inherent uncertainties of technology and the potential challenges users may encounter during use, we continuously offer enhanced technical support to help users overcome technical difficulties and deliver a superior user experience through our professional expertise.

Without authorization from Nepu Technology or certification from the relevant authorities, any unauthorized modification of this product will render its wireless compliance invalid and revoke your operational privileges. We recommend that users thoroughly review the applicable regulations and laws before using this product to avoid unnecessary complications.

We hope users will use this product in a lawful and compliant manner to jointly foster a healthy technical ecosystem. If you encounter any issues or have suggestions during use, please feel free to contact us—we are committed to serving you diligently.

China Micro-Power Short-Distance Device Declaration

This product possesses the micro-power short-range radio transmission capability stipulated by the Radio Management Regulations of the People's Republic of China. In accordance with the requirements of Announcement No.52 of 2019 issued by the Ministry of Industry and Information Technology, the following specifications are hereby stated:

(1) The product complies with the technical requirements for Category F general low-power equipment specified in the "Catalog and Technical Requirements for Low-Power Short-Range Radio Transmitters," and is intended for research and educational purposes. The product does not include an antenna; refer to the user manual for details on control switches and other specifications.

(2) The usage scenario or conditions shall not be altered without authorization; the transmission frequency range shall not be expanded; the transmission power shall not be increased (including the installation of additional RF power amplifiers); and the transmission antenna shall not be modified without authorization.

(3) No harmful interference shall be caused to other legitimate radio stations, nor shall any claim for protection against harmful interference be made;

(4) Interference from industrial, scientific, and medical (ISM) application equipment that is exposed to radio frequency energy, or from other legitimate radio stations;

(5) If harmful interference is generated to other legitimate radio stations, usage shall be immediately discontinued, and measures must be taken to eliminate the interference before resuming operation.

(6) The use of low-power equipment within aircraft, as well as within electromagnetic environment protection zones of military and civilian radio stations (including radio observatories, weather radar stations, satellite earth stations, and stations for tracking, ranging, reception, and navigation) designated in accordance with laws, regulations, national provisions, and standards, and within airports, shall comply with the regulations on electromagnetic environment protection established by the relevant competent authorities.

(7) The use of all types of model remote controls is prohibited within a 5,000-meter radius centered on the airport runway's midpoint.

(8) The operating voltage of the low-power device is 5 volts, and it functions normally at room temperature.

Nep Technology Co., Ltd. has officially released the 2022 version of its development board (Model: K210), built on the XILINX Zynq7020+ AD9361 development platform. To help you quickly familiarize yourself with this platform, we have prepared this user manual.

This SDR development platform features an integrated design approach, enabling users to easily reuse the development board for secondary applications. The board employs the XILINX Zynq 7020+ AD9363 chip solution, which combines Processing System (PS) and Programmable Logic (PL) technologies to integrate a quad-core ARM Cortex-A9 processor and FPGA programmable logic onto a single chip. The core board's PS side includes two 1GB high-speed DDR3 SDRAM chips and one 256MB QSPI FLASH chip, while the PL side handles communication with the AD9361 chip, supporting either LVDS or CMOS modes.

In the design of the development board, we have provided users with a comprehensive array of peripheral interfaces, including one BTB interface, one USB 2.0 OTG interface, one Gigabit Ethernet interface, one UART interface, one SD card slot, and one JTAG interface. These features meet diverse requirements for high-speed data exchange, data storage, video transmission and processing, deep learning, artificial intelligence, and industrial control, making it a "professional-grade" ZYNQ SDR development platform. It enables both preliminary verification and subsequent application of high-speed data transmission, exchange, and processing. This product is particularly suitable for students and engineers engaged in communication algorithm development.



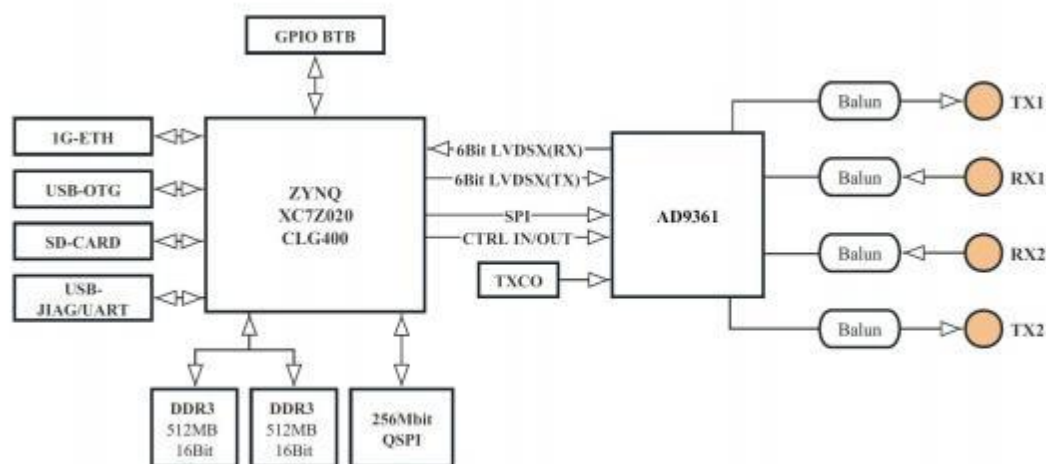
I. Introduction to the Development Board

Here is a brief functional overview of the K230 SDR development platform.

The development board primarily consists of the XC7Z020 chip, paired with two DDR3 memory modules, an AD9361 processor, and a QSPI FLASH memory module. The main chip is the XC7Z020—1CLG400I from Xilinx's Zynq 7000 series. The Zynq7020 chip comprises a Processor System (PS) and a Programmable Logic (PL) unit. Two DDR3 modules, each with a capacity of up to 512 MB, are connected to the PS, enabling independent data processing and storage for both the ARM and FPGA systems. The 256 MB QSPI FLASH module on the PS stores the Linux operating system, file system, and user data in static memory.

The development board features a comprehensive array of peripheral interfaces, including one BTB interface, one USB 2.0 OTG interface, one Gigabit Ethernet port, one UART port, one SD card slot, one JTAG interface, and several LED indicator buttons.

The figure below shows a schematic diagram of the entire development system:



Through this schematic diagram, we can see the interfaces and functions available on our development platform.

Zynq7020 Minimum System

The system consists of a Zynq7020 microcontroller, 1 GB DDR3 memory (for the Power Supply/Control Unit, PS), and 256 MB QSPI FLASH memory. Additionally, a single-ended 33.333 MHz crystal oscillator supplies the clock signal to the PS system.

USB 2.0 interface

The USB interface type is TYPE C, operating in OTG mode. It is used to connect external USB hosts, such as...

Host devices such as PCs and Raspberry Pi.

Gigabit Ethernet interface

1 The port supports 10/100M/1000M Ethernet RJ45 interfaces and is connected to the PS end. It is used for Ethernet data exchange with computers or other network devices.

USB DEBUG Interface

The USB interface is of TYPE C, featuring one UART and one JTAG port for communication with computers, facilitating user debugging. The interface chip employs the FT2232 chipset.

Micro SD card slot

1 A micro SD card slot for storing the operating system image and file system.

BTB Expansion Port

1 The user-defined BTB features an expansion port that allows connection of custom user-subboards or various BTB modules provided by us (such as HDMI input/output modules, camera modules, Ethernet modules, etc.).

JTAG Debug Port

1 The JTAG debugging port is intended for factory debugging and shares I/O resources with the FT2232. It is not recommended for user-defined use; for JTAG debugging, utilize the USB DEBUG interface instead.

LED lamp

5 One light-emitting LED, one custom LED connected to the PS end, one Done configuration indicator, one power indicator, and two data transmission indicators.

key

2 One button, one user-customizable button connected to the PS end, and one global reset button.

DIP switch

1 A 2-bit digital switch enables mode switching, supporting three boot modes: QSPI, JTAG, and SD card.

Radiofrequency SMA interface

4 One low-loss SMA interface (internal hole), two for receiving radio signals, and two for transmitting radio signals.

External reference clock input interface

- 1 An external reference clock input interface provides a clock reference for the AD9363 chip.

External local oscillator input interface

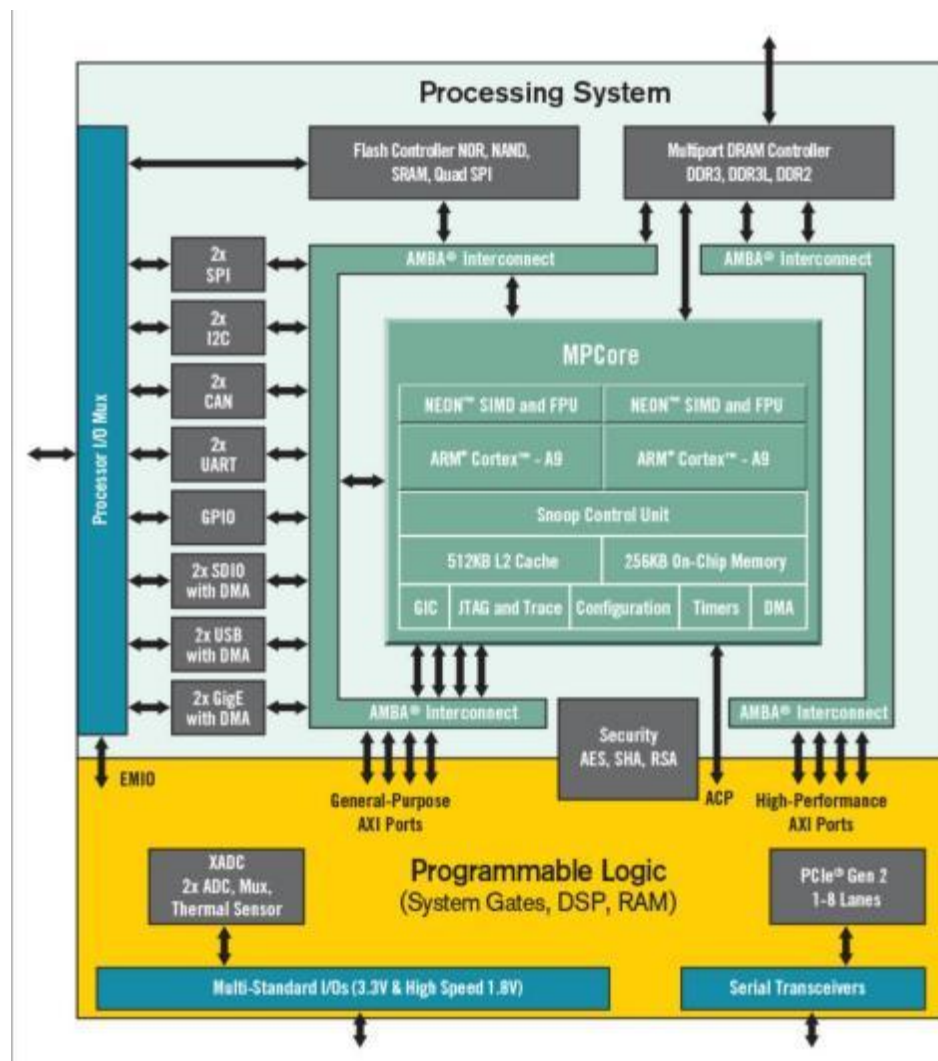
- 2 An external local oscillator input interface for receiving and transmitting external local oscillator signals.

II. ZYNQ Chip

The development board utilizes chips from Xilinx's Zynq7000 series, specifically the XC7Z020-1CLG400I model. The PS system of the Zynq7020 chip integrates two ARM Cortex-A9 processors operating at speeds up to 1 GHz.

The Zynq7020 chip supports 32-bit DDR3, DDR3L, and LPDDR3 memory chips, featuring extensive interface ports on the PS side.

These interfaces include SPI, UART, ETH, USB, QSPI, SD, among others. The PL side incorporates abundant programmable logic units, DSP, and internal RAM. The overall block diagram of the Zynq7020 chip is shown below:



Logic Cells 85K

Look-Up Tables (LUTs) 53,200

CLB Flip-Flops 106,400

DSP Slices 220

Block RAM 4.9Mb

94 I/Os, 8 MIOs

47 Pairs LVDS

100 Voltage Adjustable I/Os

AD Converter x 2, 1Mbps

Speed Grade -1, Industrial Grade

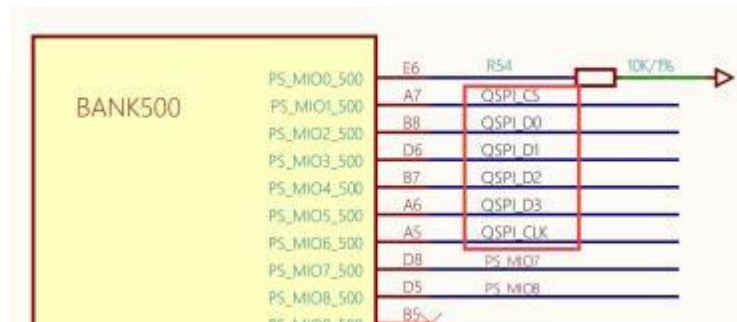
Working Temperature -40 °C to 85 °C

III. DDR Chip

The K230 development board features two Micron 512MB DDR3 chips (model MT41K256 M16TW-107), offering a 32-bit data bus width and 1GB capacity. The DDR3 SDRAM on the PS side achieves a maximum operating speed of 533MHz (data rate of 1066Mbps), with both DDR3 memory modules directly connected to the BANK502 memory interface on the PS.

4 、QSPIFlash

The K230 development board features a single 256MB Single SPI FLASH chip that provides a 4-bit bandwidth data bus. The FLASH chip model is W25Q256FVEIG and operates on a 3.3V CMOS voltage standard. Due to the non-volatile nature of QSPI FLASH, it can serve as the system's boot device to store the boot image during operation. This image primarily includes FPGA bit files, ARM application code, and other user data files. The connection between QSPI FLASH and FPGA is illustrated in the figure below.

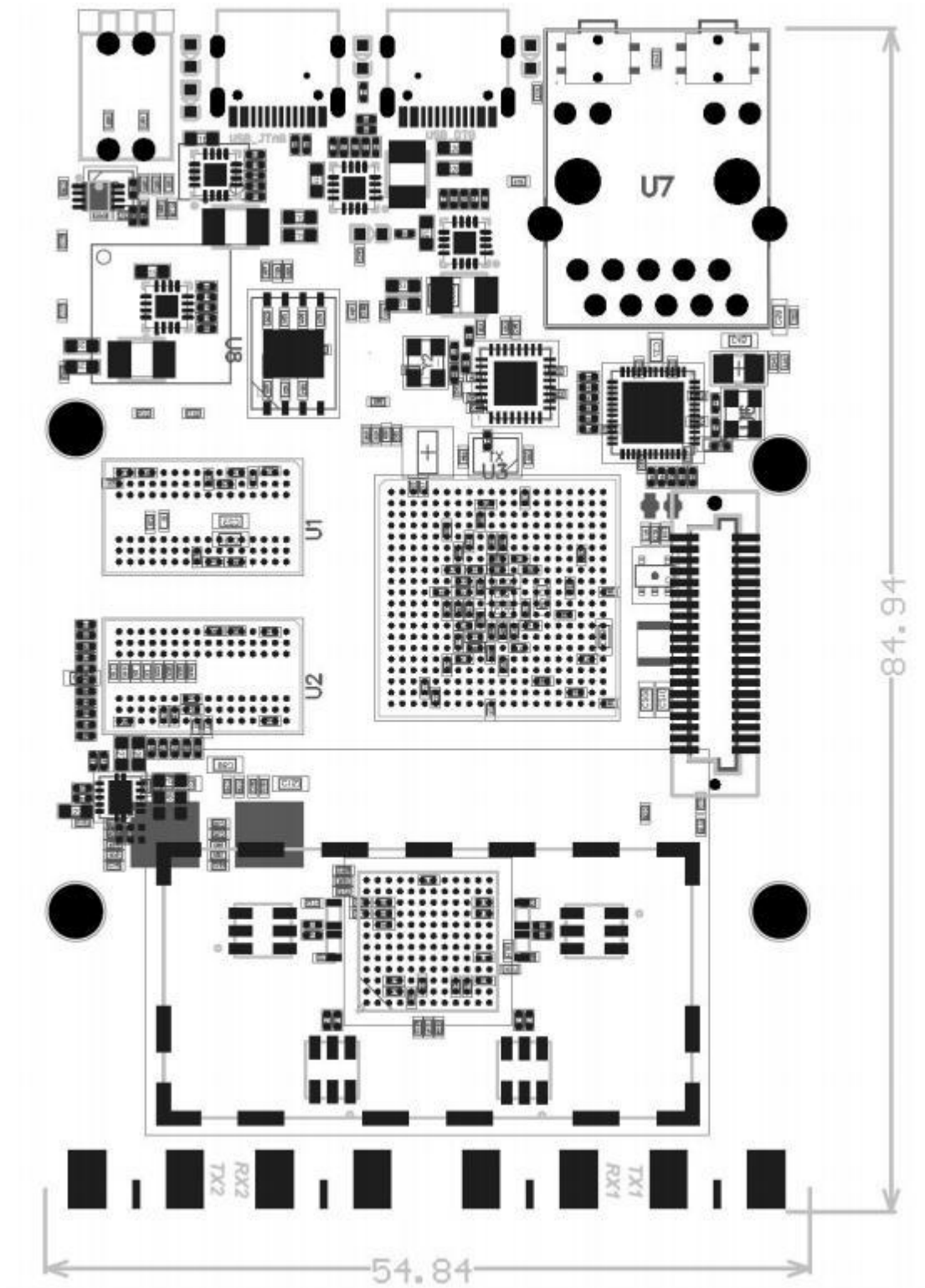


V. PS System Clock Source

The X1 crystal oscillator on the core board provides a 33.333 MHz clock input for the PS section. The clock input is connected to the PS_REF_CLK_500 pin on BANK500 of the ZYNQ chip. The schematic diagram is shown below:



VI. Structural Diagram



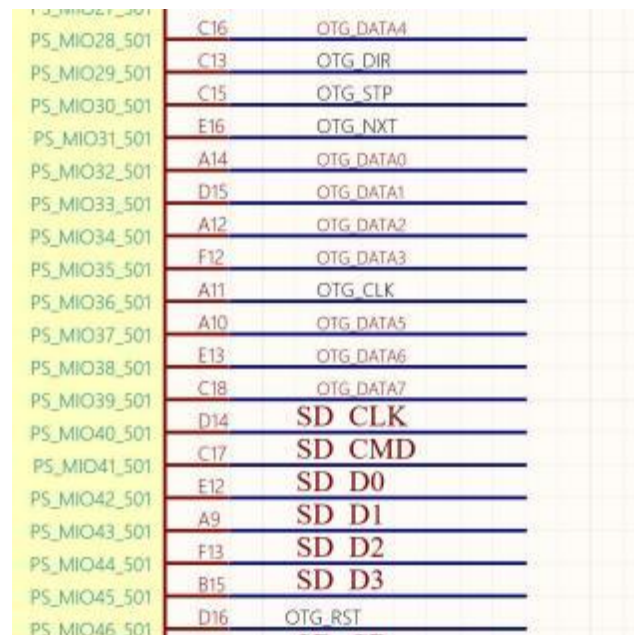
VII. BTB Connector Pin Definition

The BTB connector is connected to Zynq's BANK35 with a voltage specification of 3.3 V. The connector model is 61083-044402LF.

Pin number	FPGA pin	Pin number	FPGA pin	Pin number	FPGA pin	Pin number	FPGA pin
1	VCC3V3	11	PL_RXD0	21	L19	31	H20
2	GND	12	USB_ID	22	E18	32	F16
3	VCC3V3	13	H15	23	L20	33	G19
4	GND	14	H18	24	E19	34	D19
5	VCC3V3	15	G15	25	K19	35	G20
6	GND	16	J18	26	E17	36	D20
7	VCC3V3	17	M19	27	J19	37	F19
8	GND	18	H17	28	D18	38	C20
9	PL_TXD0	19	M20	29	J20	39	F20
10	USB_5V	20	H16	30	F17	40	B20

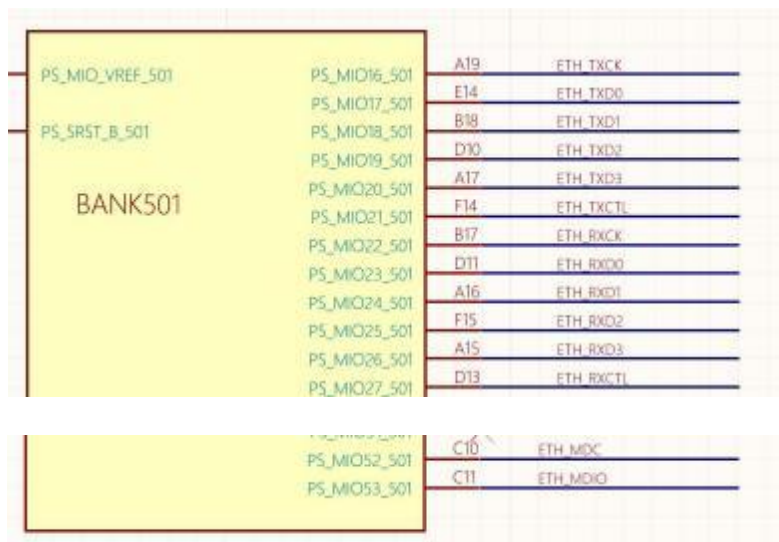
8. USB 2.0 Interface

The K230 development board features a USB 2.0 interface supporting OTG mode, with the USB PHY chip model being USB3320C. The connection to Zynq is illustrated below:



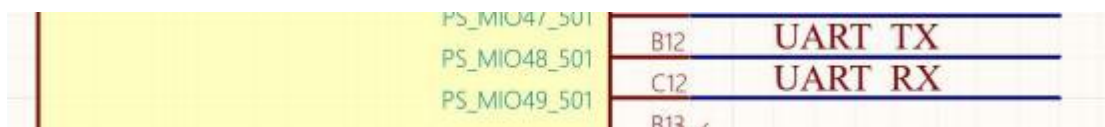
IX. Gigabit Ethernet Interface

The K230 development board features a Gigabit Ethernet interface supporting 10/100M/1000Mbps network transmission rates. The ETH PHY chip model is RTL8211E-VL, with an RGMII interface connected to the PS port. This interface enables Ethernet data exchange with computers or other network devices, as illustrated below for connection to Zynq:



10. UART Interface

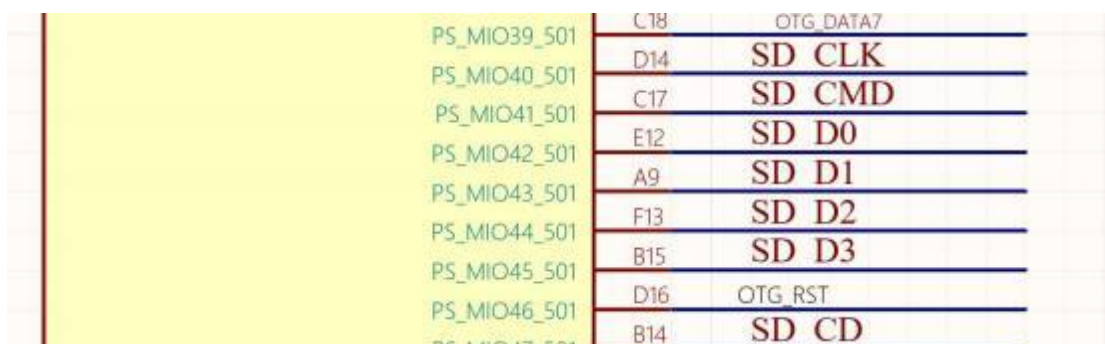
The K230 development board features a UART interface on the PS side, designated as UART1, with TXD connected to MIO48 and RXD to MIO49. The connection to Zynq is illustrated below:

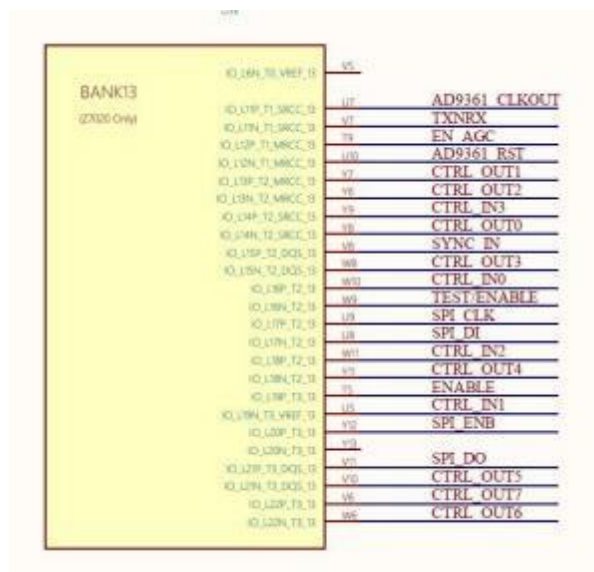


11. SD Card Slot

The K230 development board features a Micro SD card slot, enabling users to access the SD card memory for storing the Zynq7020 chip's BOOT program, Linux operating system kernel, file system, and other user data files.

The SDIO signal is connected to the IO signal of the Zynq7020's PS BANK501, as the VCCIO of Bank501 is set to 1.8V while the SD card's data level is 3.3V. The CD pin is connected via the TXS02612 level converter. The schematic diagram of the Zynq7020 PS and SD card connectors is shown below.





The radio frequency technical specifications are as follows:

1. Radio frequency: 325 MHz to 3.8 GHz
2. ADC/DAC acquisition: 12-bit
- 3.2 Route TX, Route RX, operating modes: duplex and half-duplex
4. RF matching resistor: 50 ohms
5. Instantaneous bandwidth: 20MHz
6. sampling rate

In CMOS mode:

-Maximum data rate: 122.88 Msps (dual-port half-duplex, DDR mode)

The maximum DATA_CLK rate is 61.44 MHz.

In LVDS mode:

-Maximum data rate: 122.88 Msps (dual-port full-duplex).

The maximum DATA_CLK rate is 245.76 MHz.

Radio frequency transmission technical specifications:

Local oscillator range: 325 MHz-3.8 GHz

Transmission power: <5 dBm in 1 GHz CW mode

Bandwidth: 200 kHz-20 MHz

EVM: <2% 5dBm @20MHz

Gain range:>80 dB

Gain step: 0.25 dB

Edge band suppression: 40dbc

Local oscillator suppression: 60 dBc

Radio frequency reception technical specifications:

Local oscillator range: 325 MHz-3.8GHz

Reception sensitivity: -90 dBm@20 MHz

Bandwidth: 200 kHz-20 MHz

EVM: <2% 5dBm @20MHz

Gain range:>60 dB

Gain step: 1 dB

Noise coefficient: <8 dB

IIP3: -25dbm

The xdc definition in Vivado is as follows

1. lvds25

```
# constraints
# ad9363

set_property -dict {PACKAGE_PIN N18 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_clk_in_p] ;
set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_frame_in_p] ;
set_property -dict {PACKAGE_PIN P18 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_frame_in_n] ;
set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_p [0]] ;
set_property -dict {PACKAGE_PIN W16 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[0]] ;
set_property -dict {PACKAGE_PIN Y16 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_p [1]] ;
set_property -dict {PACKAGE_PIN Y17 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[1]] ;
set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_p [2]] ;
set_property -dict {PACKAGE_PIN Y19 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[2]] ;
set_property -dict {PACKAGE_PIN V20 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_p [3]] ;
set_property -dict {PACKAGE_PIN W20 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[3]] ;
set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
```

```

[get_ports rx_data_in_p [4]] ;
set_property -dict {PACKAGE_PIN U20 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[4]] ;
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_p [5]] ;
set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVDS_25 DIFF_TERM TRUE}
[get_ports rx_data_in_n[5]] ;
set_property -dict {PACKAGE_PIN U18 IOSTANDARD LVDS_25} [get_ports
tx_clk_out_p] ;
set_property -dict {PACKAGE_PIN U19 IOSTANDARD LVDS_25} [get_ports
tx_clk_out_n] ;
set_property -dict {PACKAGE_PIN W14 IOSTANDARD LVDS_25} [get_ports
tx_frame_out_p] ;
set_property -dict {PACKAGE_PIN Y14 IOSTANDARD LVDS_25} [get_ports
tx_frame_out_n] ;
set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [0]] ;
set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[0]] ;
set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [1]] ;
set_property -dict {PACKAGE_PIN P16 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[1]] ;
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [2]] ;
set_property -dict {PACKAGE_PIN R17 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[2]] ;
set_property -dict {PACKAGE_PIN U14 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [3]] ;
set_property -dict {PACKAGE_PIN U15 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[3]] ;
set_property -dict {PACKAGE_PIN V12 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [4]] ;
set_property -dict {PACKAGE_PIN W13 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[4]] ;
set_property -dict {PACKAGE_PIN T17 IOSTANDARD LVDS_25} [get_ports
tx_data_out_p [5]] ;
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVDS_25} [get_ports
tx_data_out_n[5]] ;

set_property -dict {PACKAGE_PIN Y8 IOSTANDARD LVCMOS25} [get_ports
gpio_status [0]] ;
set_property -dict {PACKAGE_PIN Y7 IOSTANDARD LVCMOS25} [get_ports
gpio_status [1]] ;
set_property -dict {PACKAGE_PIN Y6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [2]] ;
set_property -dict {PACKAGE_PIN W8 IOSTANDARD LVCMOS25} [get_ports
gpio_status [3]] ;
set_property -dict {PACKAGE_PIN Y11 IOSTANDARD LVCMOS25} [get_ports
gpio_status [4]] ;

```

```

set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS25} [get_ports
gpio_status [5]]
;
set_property -dict {PACKAGE_PIN W6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [6]]
;
set_property -dict {PACKAGE_PIN V6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [7]]
;
set_property -dict {PACKAGE_PIN W10 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [0]]
;
set_property -dict {PACKAGE_PIN U5 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [1]]
;
set_property -dict {PACKAGE_PIN W11 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [2]]
;
set_property -dict {PACKAGE_PIN Y9 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [3]]
;
set_property -dict {PACKAGE_PIN T5 IOSTANDARD LVCMOS25} [get_ports
enable]
;
set_property -dict {PACKAGE_PIN V7 IOSTANDARD LVCMOS25} [get_ports
txrx]
;
set_property -dict {PACKAGE_PIN T9 IOSTANDARD LVCMOS25} [get_ports
gpio_en_agc]
;
set_property -dict {PACKAGE_PIN V8 IOSTANDARD LVCMOS25} [get_ports
gpio_sync]
;
set_property -dict {PACKAGE_PIN U10 IOSTANDARD LVCMOS25} [get_ports
gpio_resetb]
;

set_property -dict {PACKAGE_PIN Y12 IOSTANDARD LVCMOS25 PULLTYPE PULLUP}
[get_ports spi_csn]
;
set_property -dict {PACKAGE_PIN U9 IOSTANDARD LVCMOS25} [get_ports
spi_clk]
;
set_property -dict {PACKAGE_PIN U8 IOSTANDARD LVCMOS25} [get_ports
spi_mosi]
;
set_property -dict {PACKAGE_PIN V11 IOSTANDARD LVCMOS25} [get_ports
spi_miso]
;
# clocks

create_clock -name rx_clk -period 4 [get_ports rx_clk_in_p]

```

2. cmos25

```

# constraints

# ad9361 (SWAP == 0x1)

set_property -dict {PACKAGE_PIN N18 IOSTANDARD LVCMOS25 } [get_ports
rx_clk_in]

set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVCMOS25 } [get_ports

```

```

rx_frame_in]

set_property -dict {PACKAGE_PIN W16 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[0]]

set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[1]]

set_property -dict {PACKAGE_PIN Y17 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[2]]

set_property -dict {PACKAGE_PIN Y16 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[3]]

set_property -dict {PACKAGE_PIN Y19 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[4]]

set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[5]]

set_property -dict {PACKAGE_PIN W20 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[6]]

set_property -dict {PACKAGE_PIN V20 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[7]]

set_property -dict {PACKAGE_PIN U20 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[8]]

set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[9]]

set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[10]]

set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS25 } [get_ports
rx_data_in[11]]

```

set_property	-dict	{PACKAGE_PIN	U18	IOSTANDARD	LVC MOS25}	[get_ports tx_clk_out]
set_property	-dict	{PACKAGE_PIN	W14	IOSTANDARD	LVC MOS25}	[get_ports tx_frame_out]
set_property	-dict	{PACKAGE_PIN	T15	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [0]]
set_property	-dict	{PACKAGE_PIN	T14	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [1]]
set_property	-dict	{PACKAGE_PIN	P16	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [2]]
set_property	-dict	{PACKAGE_PIN	P15	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [3]]
set_property	-dict	{PACKAGE_PIN	R17	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [4]]
set_property	-dict	{PACKAGE_PIN	R16	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [5]]
set_property	-dict	{PACKAGE_PIN	U15	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [6]]
set_property	-dict	{PACKAGE_PIN	U14	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [7]]
set_property	-dict	{PACKAGE_PIN	W13	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [8]]
set_property	-dict	{PACKAGE_PIN	V12	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [9]]
set_property	-dict	{PACKAGE_PIN	R18	IOSTANDARD	LVC MOS25}	[get_ports tx_data_out [10]]
set_property	-dict	{PACKAGE_PIN	T17	IOSTANDARD	LVC MOS25}	[get_ports

```

tx_data_out [11]]

set_property -dict {PACKAGE_PIN Y8 IOSTANDARD LVCMOS25} [get_ports
gpio_status [0]]

set_property -dict {PACKAGE_PIN Y7 IOSTANDARD LVCMOS25} [get_ports
gpio_status [1]]

set_property -dict {PACKAGE_PIN Y6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [2]]

set_property -dict {PACKAGE_PIN W8 IOSTANDARD LVCMOS25} [get_ports
gpio_status [3]]

set_property -dict {PACKAGE_PIN Y11 IOSTANDARD LVCMOS25} [get_ports
gpio_status [4]]

set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS25} [get_ports
gpio_status [5]]

set_property -dict {PACKAGE_PIN W6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [6]]

set_property -dict {PACKAGE_PIN V6 IOSTANDARD LVCMOS25} [get_ports
gpio_status [7]]

set_property -dict {PACKAGE_PIN W10 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [0]]

set_property -dict {PACKAGE_PIN U5 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [1]]

set_property -dict {PACKAGE_PIN W11 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [2]]

set_property -dict {PACKAGE_PIN Y9 IOSTANDARD LVCMOS25} [get_ports
gpio_ctl [3]]

```

```
set_property -dict {PACKAGE_PIN T9 IOSTANDARD LVCMOS25} [get_ports
gpio_en_agc]
```

```
set_property -dict {PACKAGE_PIN U10 IOSTANDARD LVCMOS25} [get_ports
gpio_resetb]
```

```
set_property -dict {PACKAGE_PIN T5 IOSTANDARD LVCMOS25} [get_ports enable]
```

```
set_property -dict {PACKAGE_PIN V7 IOSTANDARD LVCMOS25} [get_ports txnrx]
```

```
#NO USE
```

```
set_property -dict {PACKAGE_PIN R19 IOSTANDARD LVCMOS25 PULLTYPE PULLUP}
[get_ports iic_scl]
```

```
set_property -dict {PACKAGE_PIN T19 IOSTANDARD LVCMOS25 PULLTYPE PULLUP}
[get_ports iic_sda]
```

```
set_property -dict {PACKAGE_PIN Y12 IOSTANDARD LVCMOS25 PULLTYPE PULLUP}
[get_ports spi_csn]
```

```
set_property -dict {PACKAGE_PIN U9 IOSTANDARD LVCMOS25} [get_ports spi_clk]
```

```
set_property -dict {PACKAGE_PIN U8 IOSTANDARD LVCMOS25} [get_ports
spi_mosi]
```

```
set_property -dict {PACKAGE_PIN V11 IOSTANDARD LVCMOS25} [get_ports
spi_miso]
```

```
#NO USE
```

```
set_property -dict {PACKAGE_PIN T10 IOSTANDARD LVCMOS25} [get_ports
pl_spi_clk_o]
```

```
set_property -dict {PACKAGE_PIN T12 IOSTANDARD LVCMOS25} [get_ports
pl_spi_miso]
```

```
set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS25} [get_ports
pl_spi_mosi]
```

```
set_property -dict {PACKAGE_PIN U7 IOSTANDARD LVCMOS25} [get_ports
```

```
clk_out]

create_clock -name rx_clk -period 16.27 [get_ports rx_clk_in]

# probably gone in 2016.4

create_clock      -name      clk_fpga_0      -period      10      [get_pins
"i_system_wrapper/system_i/sys_ps7/inst/PS7_i/FCLKCLK[0]" ]

create_clock      -name      clk_fpga_1      -period      5      [get_pins
"i_system_wrapper/system_i/sys_ps7/inst/PS7_i/FCLKCLK[1]" ]

create_clock -name spi0_clk      -period 40      [get_pins -hier */EMIOSPIOSCLK0]

set_input_jitter clk_fpga_0 0.3

set_input_jitter clk_fpga_1 0.15

set_property SEVERITY {Warning} [get_drc_checks NSTD-1]

set_property SEVERITY {Warning} [get_drc_checks RTSTAT-1]

set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
```

For any further inquiries, please contact us at neptunesdr@163.com.

Thank you for your support of Hangzhou NIEPU Technology Co., Ltd. Wishing you smooth use of our services in the future!

Neptune SDR

©Copyright Neptune SDR 2023

Hangzhou Nepu Technology Co., Ltd.

Address: China, Zhejiang, Hangzhou

Email: neptunesdr@163.com

Taobao: Neptune Technology SDR