User Manual

From Red Pitaya Wiki

Red Pitaya is an open source project developed around a reconfigurable measurement instrument in size of a credit card. It can replace many expensive laboratory measurement and control instruments. The users can start using the applications available within the Bazaar free of charge marketplace. This can be achieved with a single click. At the same time they can view and modify the published source code in order to develop new applications and share their results with the community.

The Red Pitaya unit is a network attached device based on Linux operating system. It includes Radio Frequency signal acquisition and generation technologies, FPGA, Digital Signal Processing and CPU processing. Red Pitaya enables everyone to start using technologies, yesterday available only to advanced research laboratories and industry.

Contents

What is in the box

Instrument:

- Red Pitaya

Optional accessories:

- 5 V / 2 A micro USB power supply
- Preloaded micro SD card
- 2 x SMA – BNC adapters
- 2 x oscilloscope probes

Abbreviations

Abbreviations used in this document are listed in Table 1.

Table 1: Abbreviations

Abbreviation	Description
ADC	Analog-to-Digital Converter
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
FIFO	First In First Out (queue)
FPGA	Field Programmable Gate Array
IP	Internet Protocol address
MAC	Media Access Control address
OS	Operating System
PC	Personal Computer
RF	Radio Frequency
BW	Bandwidth
SMA	SubMiniature version A connector
DAC	Digital-to-Analog Converter
SD	Secure Digital
LED	Light-emitting Diode
COM	Communication port
SSH	Secure Shell
SFTP	Secure File Transfer Protocol
NFS	Network File System

Safety symbols and terms

The Exclamation symbol.png symbol on the instrument indicates the user should refer to the operating instructions located in this document.

The **CAUTION** heading in this manual explains hazards which could damage the instrument. Such damage may invalidate the warranty.

The **NOTE** heading in this manual gives important explanations on the usage to avoid misunderstandings.

If the device is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Support

Please address your technical support questions here: Technical support (https://web.archive.org/web/20150812084942/https://redpitaya.zendesk.com/hc/en-us/requests/new)

Hardware

Hardware interfaces

Red Pitaya features several measurement, control, communication and storage interfaces. They are shown in Figure: Interfaces.

Interfaces.png

Figure: Interfaces.

Table 2: Interfaces and their description

Name	Type	Connector	Description
IN1	Input	SMA-F	RF input (High-Z, 1 MΩ // 10 pF)
IN2	Input	SMA-F	RF input (High-Z, 1 MΩ // 10 pF)
OUT1	Output	SMA-F	RF output (50 Ω)
OUT2	Output	SMA-F	RF output (50 Ω)
Ethernet	Full-duplex	RJ45	1000Base-T Ethernet connection
USB	Full-duplex	A USB	Used for standard USB devices
Micro USB (Console)	Full-duplex	Micro B USB	Used for console connection
Micro USB (Power)	Input	Micro B USB	5 V / 2 A power supply
Micro SD	Full-duplex	Micro SD slot	Micro SD memory card

Gain setting for the input channels

Gain can be individually adjusted for both input channels. The adjustment is done by bridging the jumpers located behind the corresponding input SMA connector.

Jumper settings.png Jumper settings photo.png

Figure: Jumper setting

Left setting (LV) adjusts +/- 1 V full scale.

Right setting (HV) adjusts +/- 20 V full scale.

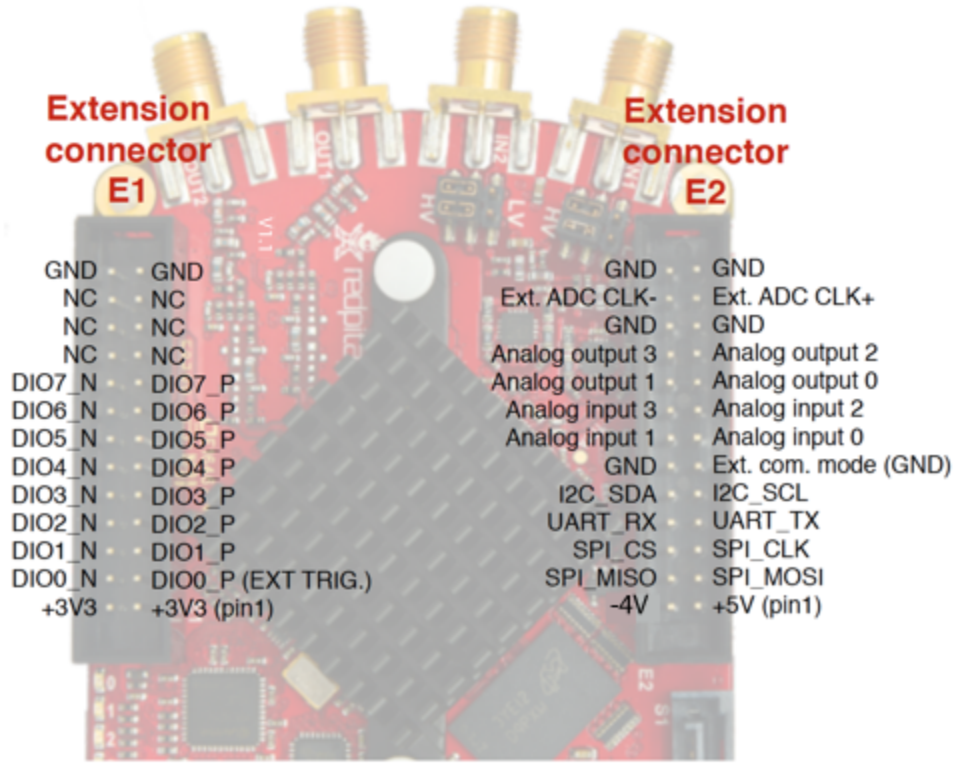
CAUTION: Jumper settings are limited to the described positions. Any other configuration or use of different jumper type may damage the product.

LEDs

Table 4: LED pin description

LED	Description	FPGA pin number	FPGA pin description
Yellow 0		F16	IO_L6P_T0_35
Yellow 1		F17	IO_L6N_T0_VREF_35
Yellow 2		G15	IO_L19N_T3_VREF_35
Yellow 3		H15	IO_L19P_T3_35
Yellow 4		K14	IO_L20P_T3_AD6P_35
Yellow 5		G14	IO_0_35
Yellow 6		J15	IO_25_35
Yellow 7		J14	IO_L20N_T3_AD6N_35
Yellow 8		E6	PS_MIO0_500
Red		D8	PS_MIO7_500
Green	Power Good	K18	
Blue	FPGA Done	R11	DONE_0

Extension connectors



Extension connector E1

- 3v3 power source

- 16 single ended or 8 differential digital I/Os with 3,3V logic levels

Table 5: Extension connector E1 pin description

Pin	Description	FPGA pin number	FPGA pin description	Voltage levels
1	3V3			
2	3V3			
3	DIO0_P	G17	IO_L16P_T2_35 (EXT TRIG)	3.3V
4	DIO0_N	G18	IO_L16N_T2_35	3.3V
5	DIO1_P	H16	IO_L13P_T2_MRCC_35	3.3V
6	DIO1_N	H17	IO_L13N_T2_MRCC_35	3.3V
7	DIO2_P	J18	IO_L14P_T2_AD4P_SRCC_35	3.3V
8	DIO2_N	H18	IO_L14N_T2_AD4N_SRCC_35	3.3V
9	DIO3_P	K17	IO_L12P_T1_MRCC_35	3.3V
10	DIO3_N	K18	IO_L12N_T1_MRCC_35	3.3V
11	DIO4_P	L14	IO_L22P_T3_AD7P_35	3.3V
12	DIO4_N	L15	IO_L22N_T3_AD7N_35	3.3V
13	DIO5_P	L16	IO_L11P_T1_SRCC_35	3.3V
14	DIO5_N	L17	IO_L11N_T1_SRCC_35	3.3V
15	DIO6_P	K16	IO_L24P_T3_AD15P_35	3.3V
16	DIO6_N	J16	IO_L24N_T3_AD15N_35	3.3V
17	DIO7_P	M14	IO_L23P_T3_35	3.3V
18	DIO7_N	M15	IO_L23N_T3_35	3.3V
19	NC			
20	NC			
21	NC			
22	NC			
23	NC			
24	NC			
25	GND			
26	GND			

Extension connector E2

- +5V & -3V3 power source
- SPI, UART, I2C
- 4 x slow ADCs
- 4 x slow DACs
- Ext. clock for fast ADC

Table 6: Extension connector E2 pin description

Pin	Description	FPGA pin number	FPGA pin description	Voltage levels
1	+5V			
2	-4V (50mA)*			
3	SPI(MOSI)	E9	PS_MIO10_500	3.3V
4	SPI(MISO)	C6	PS_MIO11_500	3.3V
5	SPI(SCK)	D9	PS_MIO12_500	3.3V
6	SPI(CS#)	E8	PS_MIO13_500	3.3V
7	UART(TX)	C8	PS_MIO08	3.3V
8	UART(RX)	C5	PS_MIO09	3.3V
9	I2C(SCL)	B9	PS_MIO50_501	3.3V
10	I2C(SDA)	B13	PS_MIO51_501	3.3V
11	Ext com.mode			GND (default)
12	GND			
13	Analog Input 0			0-3.5V
14	Analog Input 1			0-3.5V
15	Analog Input 2			0-3.5V
16	Analog Input 3			0-3.5V
17	Analog Output 0			0-1.8V
18	Analog Output 1			0-1.8V
19	Analog Output 2			0-1.8V
20	Analog Output 3			0-1.8V
21	GND			
22	GND			
23	Ext Adc CLK+			LVDS
24	Ext Adc CLK-			LVDS
25	GND			
26	GND			

* Red Pitaya Version 1.0 has -3.3V on pin 2. Red Pitaya Version 1.1 has -4V on pin 2.

- External ADC clock
- Powering Red Pitaya through external connector

JTAG

Table 7: JTAG header pin description

Pin	Description	FPGA pin number	FPGA pin description
1	3V3		
2	GND		
3	TCK	F9	TCK_0
4	TDO	F6	TDO_0
5	TDI	G6	TDI_0
6	TMS	J6	TMS_0

Getting Started

Connecting to Red Pitaya

Red Pitaya is a network attached device acting as a WEB server. Users can connect to Red Pitaya by simply typing its IP address in the WEB browser address bar. Furthermore, the Red Pitaya IP discovery service (https://web.archive.org/web/20150812084942/http://discovery.redpitaya.com/) helps you automatically connect to your Red Pitaya(s) in your local network. The following paragraphs provide detailed information about Red Pitaya connection.

Network DHCP configuration

By default, Red Pitaya obtains its network configuration using the DHCP protocol (see Figure: Default network configuration). If no reply from the DHCP server is obtained within 20 seconds, Red Pitaya will configure its network interface to:

- IP address: 192.168.1.100
- Netmask: 255.255.255.0

I case you connected your computer directly to Red Pitaya with network cable follow this tutorial to setup the network interface.

- Linux (Ubuntu)
- Windows
- Macintosh OSX

With dhcp.png

Figure: Default network configuration

If default configuration is not suitable for you, please follow instructions for Manual network interface configuration.

Red Pitaya SD card preparation

In case you ordered the Red Pitaya kit without the micro SD card you need to install the Red Pitaya software to an empty Micro SD card (Class 4 or better). Follow this procedure:

- Download the Red Pitaya SD card image from our website: redpitaya.com/turniton (https://web.archive.org/web/20150812084942/http://redpitaya.com/turniton/).
- Insert the Micro SD card into the PC's SD slot. Use an adapter if required.
- Make sure the Micro SD card's capacity does not exceed 32 GB, it is empty and formatted as a FAT32 file system.
- Extract the downloaded Red Pitaya SD card image zip file and copy its content to the Micro SD card.

OS installation.png

Figure: OS installation to Micro SD card.

- Final directory structure in the Micro SD card should look like:

```
/bin
/etc
/sbin
/src
/www
boot.bin
devicetree.dtb
uImage
uramdisk.image.gz
version.txt
```

- Unmount and remove the Micro SD card from your PC and insert it into Red Pitaya.

Manual network interface configuration

To set-up a custom network configuration, unplug the Micro SD card from Red Pitaya and plug it into your computer or tablet. Use a text editor and edit the /etc/network/interfaces file. Follow instructions given in the file.

To set a **static** configuration:

- Comment (with #) the iface eth0 inet dhcp line.
- Uncomment the lines below including iface eth0 inet static. Set up the network configuration matching your network.

To set a **dynamic** (DHCP) configuration:

- Uncomment the iface eth0 inet dhcp line.
- Comment the lines below including iface eth0 inet static.

```
# Red Pitaya network configuration
#


###########################
```

```
# lo: Loopback interface  #
###########################

auto lo
iface lo inet loopback


######################################
# eth0: Wired Ethernet - 1000Base-T  #
######################################
#
# Uncomment only one: dynamic or static configuration.
#

# Dynamic (DHCP) IP address
iface eth0 inet dhcp
    udhcpc_opts -t7 -T3

# Static IP address
#iface eth0 inet static
#    address 192.168.1.101
#    netmask 255.255.255.0
#    gateway 192.168.1.1


################################
# wlan0: Wireless USB adapter  #
################################

auto wlan0
iface wlan0 inet dhcp
    pre-up wpa_supplicant -B -D wext -i wlan0 -c /opt/etc/network/wpa_supplicant.conf
    post-down killall -q wpa_supplicant
    udhcpc_opts -t7 -T3
```

- Save the file and plug the Micro SD card back into Red Pitaya before powering it on.

- Direct cable connection - Windows 7

Power on

To power ON your Red Pitaya, follow the sequence of these steps exactly:

- Step 1: Place your Red Pitaya on a firm flat surface in a well ventilated environment.
- Step 2: Make sure the Micro SD card is inserted into Red Pitaya.
- Step 3: Plug in the network cable.
- Step 4: If required, plug in the USB cable for console connection.
- Step 5: Connect the power cable and check the status of LED diodes: Blue (D) and Green (P) LED diodes should be lit. The Yellow LED 0 should blink.

Interfaces and steps mentioned above are indicated in the figure below.

Bootup procedure side.png

Figure: Boot procedure in steps.

Red Pitaya command line access

Console (USB) connection

Serial console connection is independent from the Ethernet connection. Use a Micro USB cable to connect your computer with Red Pitaya. Connection instructions will be given for Windows, Linux and OS X users separately. The serial port configuration is given in Table 3.

Table 3: Console connection parameters

Parameter	Value
Speed	115200
Data bits	8
Stop bits	1
Parity	None
Flow control	None

Neither username nor password is required to use the serial console. Boot-up sequence finishes with a UNIX shell command prompt:

```
redpitaya>
```

Windows users

Download and install the FTD driver (https://web.archive.org/web/20150812084942/http://www.ftdichip.com/Drivers/VCP.htm) to your PC. After installation, a new COM port will appear in the Device Manager you can use in Hyperterminal or another terminal utility to connect to Red Pitaya.

Win7 hyper.png

- Windows 7 users

Linux users

To access the serial console use one of the serial communication tools e.g. minicom (https://web.archive.org/web/20150812084942/http://en.wikipedia.org/wiki/Minicom) or GTKTerm (https://web.archive.org/web/20150812084942/https://fedorahosted.org/gtkterm/). Follow these steps:

- Connect the USB cable between your PC and Red Pitaya.
- Open a terminal window and check the USB devices (look for »Future Technology Devices«):

```
lsusb | grep Future
Bus 005 Device 002: ID 0403:6015 Future Technology Devices International, Ltd
```

The ttyUSBx device must be listed in /dev:

```
user@ubuntu:~$ ll /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 2013-11-27 08:09 /dev/ttyUSB0
```

- Following that, run 'minicom' as shown below:

```
user@ubuntu:~$ minicom -D /dev/ttyUSB0
```

- Serial console will open.

NOTE: Some older Linux distributions (e.g. Ubuntu versions 10.04 and 11.04) do not recognize the FTD USB device automatically. To work arround this, issue the following command in the terminal window:

```
sudo modprobe ftdi_sio vendor=0x0403 product=0x6015
```

To load the driver automatically at boot, add this line to /etc/modules file:

```
ftdi_sio vendor=0x0403 product=0x6015
```

OS X users

Download and install the MAC OS X FTDI driver (https://web.archive.org/web/20150812084942/http://www.ftdichip.com/Drivers/VCP.htm) on your Mac.

Then, follow these steps:

- Run XOS terminal (Launchpad → Other → Terminal) and list usbserial device:

```
localhost:/ user$ ls /dev/cu.*
/dev/cu.usbserial-DN003N3N
```

- Connect to Red Pitaya using the screen tool:

```
localhost:~ user$ screen /dev/cu.usbserial-DN003N3N 115200 8N1

redpitaya>
```

To quit the screen tool press the sequence: [CTRL + A] [K] [Y].

SSH connection

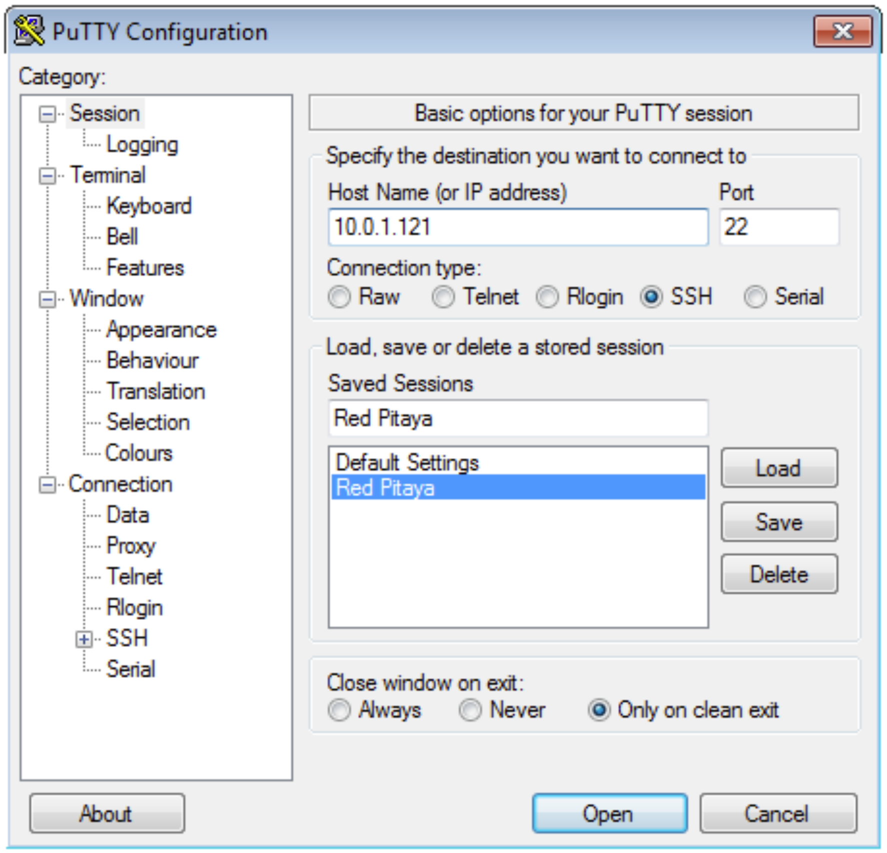
SSH connection can be established using standard SSH clients such as OpenSSH (https://web.archive.org/web/20150812084942/http://www.openssh.com/) (Linux, OS X) or PuTTy (https://web.archive.org/web/20150812084942/http://www.putty.org/) (Windows). Access information for SSH connection:

- Username: root
- Password: root

Connection examples will be given for Windows, Linux and OS X users separately.

Windows users

For this example, PuTTy tool was used on Windows XP and Windows 7 Starter OS. Run PuTTy and enter the Red Pitaya's IP address to »Host Name (or IP address)« field as shown in figure below.



Specify the destination you want to connect to
Host Name (or IP address): 10.0.1.121
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Load, save or delete a stored session
Saved Sessions: Red Pitaya
Default Settings
Red Pitaya
Close window on exit: Always Never [✓ Only on clean exit]

Figure: PuTTy SSH connection settings.

If you attempt to connect to Red Pitaya for the first time, a security alert will pop-up asking you to confirm the connection. At this time, the ssh-key will be added to the registry in your computer. Command prompt pops-up after login is successful (see figure below).

Win putty logged.png

Figure: SSH connection via PuTTy

Linux users

Start Terminal and type:

```
user@ubuntu:~$ ssh root@192.168.1.100
root@192.168.1.100's password: root

Red Pitaya GNU/Linux/Ecosystem version 0.90-299

redpitaya>
```

OS X users

Run XOS terminal: Launchpad → Other → Terminal and type:

```
localhost:~ user$ ssh root@192.168.1.100
root@10.0.3.249's password: root

Red Pitaya GNU/Linux/Ecosystem version 0.90-299

redpitaya>
```

Web browser connection

Applications running on Red Pitaya are client-server based and can be accessed with a standard WEB browser. Simply type the Red Pitaya's IP address to your WEB browser address bar and wait for the main screen to load (see figure below).

Main screen.png

Figure: Main screen of WEB interface.

Red Pitaya as a WiFi Access point (and router)

You can configure your Red Pitaya to act as a WiFi (IEEE 802.11 a/b/g) access point using the recommended Edimax EW7811Un USB WiFi dongle. This will allow you to connect to you Red Pitaya directly with your PC/ tablet/smartphone via your wireless link and will save you the trouble of finding out the IP address assigned to Red Pitaya by your LAN in a classic client mode network configuration.

To enable the Acces Point (AP) mode on Red Pitaya, set the WIFI to:

```
WIFI=ap
```

in sd://etc/network/config file.

```
Note: Access Point (AP) mode is the default mode for Ecosystem builts 0.92-388-d36be19 and later.
```

The default Red Pitaya WiFi AP parameters:

- SSID: Red Pitaya
- Password: redpitaya
- Encryption: WPA2-PSK
- WiFi channel: 6

Once connected to "Red Pitaya" WiFi, you can simply access your Red Pitaya device by IP: 192.168.128.1 or a hostname: redpitaya.

Furthermore, while using the AP mode, Red Pitaya will act as a full router performing NAT between the WiFi (LAN) and the 1000Base-T Ethernet (WAN).

In addition to the ability to access the Red Pitaya in a simple way via the WiFi connection, this allows you to access the Internet, including the Bazaar and its functionality.

Possible configurations

```
                       WiFi
PC/tablet/smartphone  <---->  Red Pitaya
       client
```

```
                       WiFi                ETH
PC/tablet/smartphone  <---->  Red Pitaya  <----->      Your LAN
       client                                       or the Internet
                            (192.168.128.1)       (all except 192.168.128.x)
```

Red Pitaya is authoritative for its 192.168.128.0/24 network on the WiFi segment, while it acts as a client (DHCP or static) on the 1000Base-T Ethernet (WAN) connection.

You can connect the Red Pitaya ETH port either:

- Directly to your ISP, as you would normally do with a home router (WAN connection) or
- To your LAN, as long as your LAN is not the 192.168.128.0/24 network segment.

The default 129.168.128.0/24 WiFi network segment is configurable and can be modified to resolve the routing conflicts, should you be that unfortunate to have your existing LAN segment on 192.168.128.0/24 (most of the home LANs are on 192.168.1.0/24 segment)

Several AP or network segment related parameters can be modified in:

sd://etc/network/hostapd.conf
sd://etc/network/interfaces.ap
sd://etc/network/dnsmasq.conf

Web applications

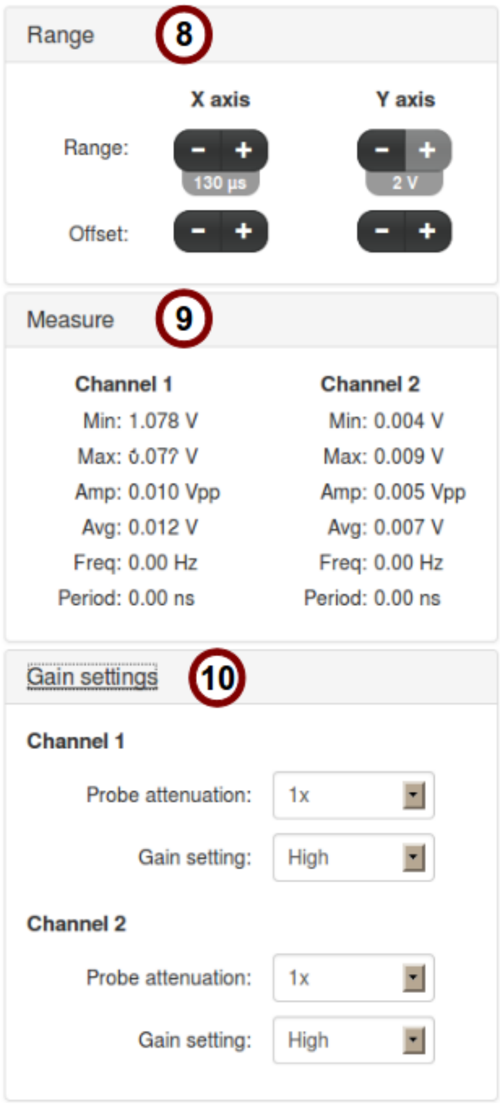
When user connects to a Red Pitaya unit through a WEB browser, a list of locally installed applications is visualized (see Figure: Main screen of WEB interface). Applications can be started by clicking the “RUN” button. To install new applications refer to Installing applications section.

Oscilloscope application

This application turns your Red Pitaya into two channel time domain signal visualization system. The main part of the application is the plot area where 2 RF input signals are presented. The user can control how the traces are visualized through the buttons on the top of the plot area (see figure below). Other options are available in the drop down menus on the right side, see Table 4 for details. Screenshots were taken from a PC browser. The layout may differ on other devices/browsers.

Oscilloscope 1.png

Figure: Oscilloscope application



	X axis	Y axis
Range:	130 µs	2 V

Channel 1	Channel 2
Min: 1.078 V	Min: 0.004 V
Max: 0.077 V	Max: 0.009 V
Amp: 0.010 Vpp	Amp: 0.005 Vpp
Avg: 0.012 V	Avg: 0.007 V
Freq: 0.00 Hz	Freq: 0.00 Hz
Period: 0.00 ns	Period: 0.00 ns

Channel 1

Probe attenuation: 1x

Gain setting: High

Channel 2

Probe attenuation: 1x

Gain setting: High

Figure: Oscilloscope application drop down menus

Table 4 contains description of Oscilloscope functionalities referring to the annotations in Figure: Oscilloscope application and Figure: Oscilloscope application drop down menus.

Table 4: Functions of Oscilloscope application

Annotation number	Description
1	Autoscale: When clicked it sets the voltage range in order for the current input signals to optimally fit into the plot area.
2	Reset zoom: When clicked it resets the time range to the interval 0 to 130 μs and sets the voltage range to full scale.
3	Diagram tools: zoom in / zoom out / pan
4	Channel enable buttons: Enable or disable visualization of specific channel.
5	Averaging: Enables sample averaging during long acquisitions. Signal to noise ratio can be improved through averaging, but the information concerning high frequencies is lost. Refer to the specification documentation for details concerning decimation.
6	AUTO button: Click this button to automatically select the optimal visualization range for the currently applied signals. This functionality is limited to input signals from 3 Hz to 50 MHz.
7	Trigger menu (click on the “Trigger” bar to expand the trigger menu, Figure: Oscilloscope application drop down menus): ▪ Source: trigger source selection (channel1, channel2, external) ▪ Mode: trigger mode ▪ Auto: Continuously acquires sequences of samples (asynchronously to the input signal). ▪ Normal: Acquires traces when trigger conditions are met (time t=0 in the plot area represents the trigger event) ▪ Single: A single trace is acquired when “Single” button is clicked. ▪ Edge: Selects the trigger edge (rising, falling). ▪ Level: Selects the trigger level.
8	Range menu (Figure: Oscilloscope application drop down menus): Range and offset buttons enable quick diagram visualization area setup.
9	Measure menu: It gathers voltage statistics and frequency/time characteristics of both input signals. This functionality is limited to work reliably with input signals from 3 Hz to 50 MHz.
10	Gain settings menu (Figure: Oscilloscope application drop down menus): ▪ Probe attenuation: Should be set according to the probe being used. ▪ Gain setting: Should be set according to the current jumper setting.

To exit the Oscilloscope application, click on Back button on the top left of the screen.

Signal generator application

The signal generator functionality is available as an extension of the Oscilloscope application in order to be able to use both of them simultaneously.

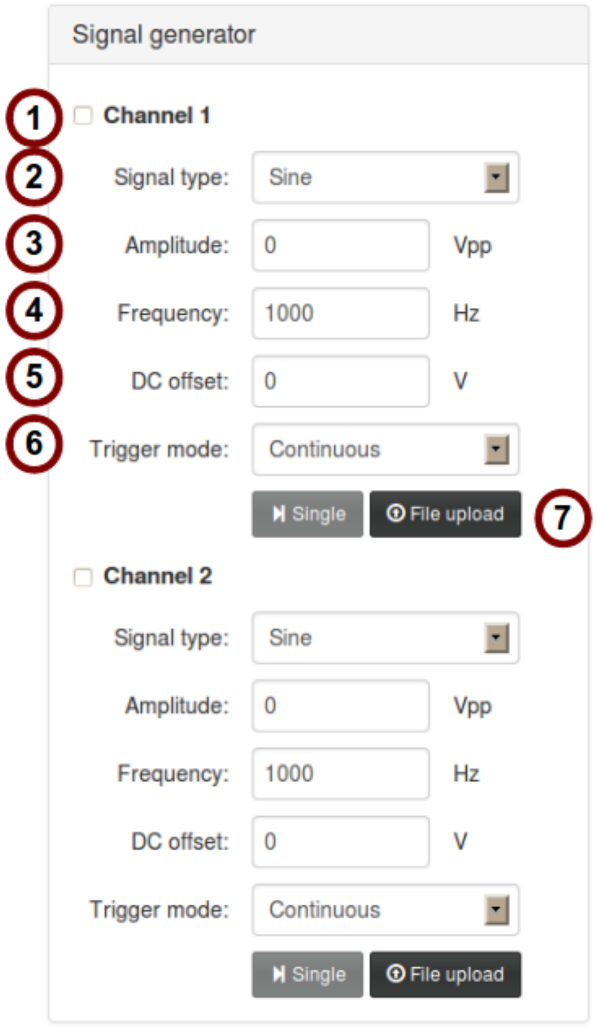


Figure: Signal generator drop down menu.

Table 5 contains description of Oscilloscope functionalities referring to annotations in Figure: Signal generator drop down menu.

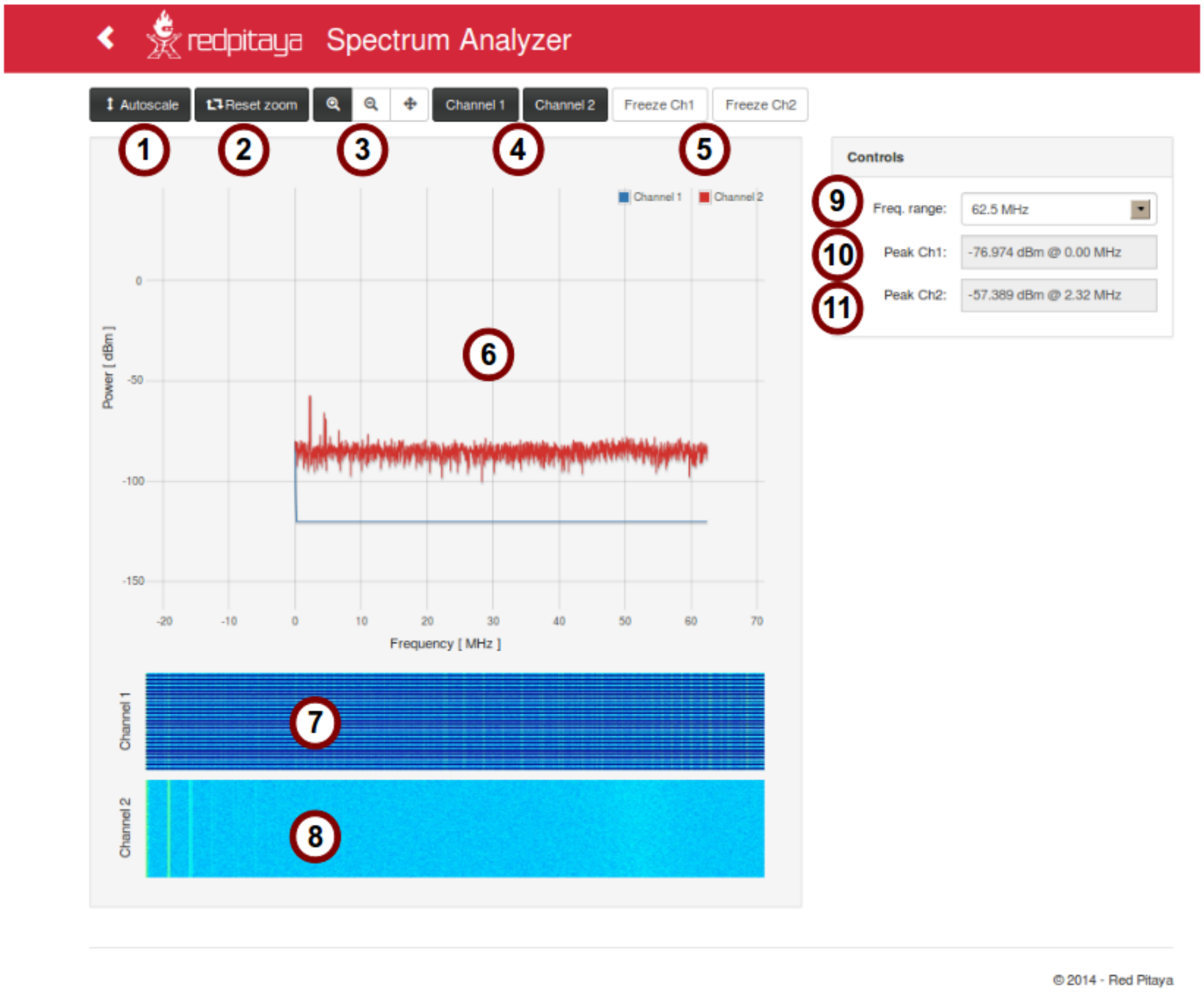
Table 5: Functions of the signal generator

Annotation number	Description
1	Signal enable checkbox: It enables/disables the signal generation on a specific channel.
2	Signal type: It defines the output signal type (sinusoidal, rectangular, triangular, and arbitrary - from file).
3	Amplitude: It defines the output signal amplitude.
4	Frequency: It defines the output signal frequency.
5	DC offset: It defines the signal DC offset.
6	Trigger mode: It defines how the signal is triggered (Continuous, Single, External) “Single” button: When single trigger mode is configured, the click of this button triggers a single sequence of samples with duration equal to the predefined signal frequency. (The last sample in the sequence is applied after the sequence is completed.)
7	“File upload”: It enables the upload of the arbitrary signal samples in the form of a CSV file. The application supports single column text files formatting. The maximum number of samples is 16383. The signal values are normalized to the predefined signal generator range. (See: File upload example).

Note: The output channels are designed to drive 50 Ω loads. Terminate outputs when channels are not used. Connect parallel 50 Ω load (SMA tee junction) in high impedance load applications.

Spectrum analyzer application

This application turns your Red Pitaya into a two channel frequency domain analysis system. The plot area represents the input signal power in dBms versus frequency. The user can control the diagram area using the buttons on top of plot area and the control menu on the right. See Table 6 for details. Screenshots were taken from a PC browser. The layout may differ on other devices/browsers.



Freq. range: 62.5 MHz
Peak Ch1: -76.974 dBm @ 0.00 MHz
Peak Ch2: -57.389 dBm @ 2.32 MHz

Figure: Spectrum analyzer application.

Table 6: Functions of Spectrum analyzer

Annotation number	Description
1	Autoscale: It sets the Y axis scaling factor in order to optimally fit the visualized spectra into the plot area.
2	Reset zoom: It displays the whole frequency range and sets the Y axis to the range from -120 dBm to 20 dBm.
3	Diagram tools: zoom in / zoom out / pan (tablet devices do not visualize these tools).
4	Channel enable buttons: They enable or disable channel visualization.
5	Channel freeze buttons: They freeze the current spectrum of a channel.
6	Main display.
7	Waterfall diagram for channel 1.
8	Waterfall diagram for channel 2.
9	Frequency range selection: The main frequency range covers DC to 62.5 MHz. Additional frequency ranges starting from DC are available in order to observe signal behavior at lower frequencies. For details concerning the frequency range check the specification document.
10	Peak Ch1: It numerically displays the peak marker for channel 1.
11	Peak Ch2: It numerically displays the peak marker for channel 2.

To exit the Spectrum analyzer application, click on Back button on the top left of the screen.

IST RealProbe Flow (contributed)

NewAppIST.jpg

This application demonstrates a system for characterization of the IST RealProbe sensor - measuring the velocity of liquids.

The complete and original description of the method and of the system was supplied by Flavio Ansovini (Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni (DITEN), University of Genoa.

His description can be opened from File:Article RedPitaya + IST realprobe sensor ENmin.pdf or it can be downloaded from HERE (https://web.archive.org/web/20150812084942/https://www.dropbox.com/s/0o2v34cr6fp7zx5/Article%20RedPitaya%20%2B%20IST%20realprobe%20sensor_ENob.pdf).

LCR meter (contributed)

LCR meter is a measurement instrument for measuring impedance (Z), inductance (L), capacitance (C) and resistance (R) of DUT (device under test). It can be obtained from the Bazaar. LCR meter **needs an additional shunt resistor**, which should have a value in range (0.1*|Z(dut)|< R < 10*|Z(dut)|) to allow more precise measurements. Minimum value of shunt resistor is 10Ohm. Connect your Red Pitaya as shown in the picture below.

Lcr conn.png

After a successful installation, command line utility **lcr** and WebUI is available. The WebUI is similar and build upon other Red Pitaya applications. An example image is shown below.

Lcr gui.png

The plot is user-configurable. The X-axis can be either in linear or logarithmic scale and the plotted results on the Y-axis can be one of the following: impedance amplitude (|Z|) or phase, admittance amplitude (|Y|) or phase, serial reactance (Xs) or inductance (Ls) or capacitance (Cs) or resistance (Rs), parallel reactance (Xp) or inductance (Lp) or capacitance (Cp) or resistance (Rp), quality (Q) or dissipation (D) factor.

Command line utility manpage shows that all of the measurement parameters from the WebUI are also usable through the CLI:

```
redpitaya> /opt/www/apps/lcr_meter/lcr
LCR meter version 0.25, compiled at Sat Oct 11 13:38:36 2014

Usage:  lcr [channel] [amplitude] [dc bias] [r_shunt] [averaging] [calibration mode] [z_ref real] [z_ref imag] [count/steps

        channel            Channel to generate signal on [1 / 2].
        amplitude          Signal amplitude in V [0 - 1, which means max 2Vpp].
        dc bias            DC bias/offset/component in V [0 - 1].
                           Max sum of amplitude and DC bias is (0-1]V.
        r_shunt            Shunt resistor value in Ohms [>0].
        averaging          Number of samples per one measurement [>1].
        calibration mode   0 - none, 1 - open and short, 2 - z_ref.
        z_ref real         Reference impedance, real part.
        z_ref imag         Reference impedance, imaginary part.
        count/steps        Number of measurements [>1 / >2, dep. on sweep mode].
        sweep mode         0 - measurement sweep, 1 - frequency sweep.
        start freq         Lower frequency limit in Hz [3 - 62.5e6].
        stop freq          Upper frequency limit in Hz [3 - 62.5e6].
        scale type         0 - linear, 1 - logarithmic.
        wait               Wait for user before performing each step [0 / 1].

Output: frequency [Hz], PhaseZ [deg], Z [Ohm], Y [S], PhaseY [deg], R_s [Ohm], X_s [H], G_p [S], B_p [S], C_s [F], C_p [F],
```

Note that the LCR meter **cannot measure at frequencies under 3Hz**. An example of **lcr** executable call is as follows:

```
redpitaya> /opt/www/apps/lcr_meter/lcr 1 1 0 100 1 0 0 0 5 1 1000 2000 1 0
```

In the first column of the output, a frequency at which the DUT was measured is printed, the second column is the phase and the third is the amplitude of measured impedance. Other columns are additional calculated parameters, listed on the manpage with the formulas written in code comments.

Installing applications

Red Pitaya applications of the current release can be downloaded and installed from Bazaar – the application marketplace. To review and manage (install, upgrade or remove) the available applications, connect to your Red Pitaya through a WEB browser (refer to Web browser connection section), make sure your Red Pitaya has access to the Internet and click on the Bazaar link. A list of applications will appear (see figure below). By clicking the “Install” button, the selected application will be installed on your Red Pitaya. The applications that are already installed on your Red Pitaya can be removed by clicking the »Uninstall« button, or upgraded if newer versions are available, by clicking the »Upgrade« button.

Bazaar.png

Figure: Bazaar.

Older versions

Older versions of the Ecosystem and the applications are available at http://archives.redpitaya.com and can be manually installed:

- by extracting the content of the <ecosystem>.zip archive to the root directory of an empty SD card;
- by extracting the content of the <application>.zip archive to the /www/apps/ directory of the SD card.

Latest development versions

If you prefer to live on the bleeding-edge and cannot wait until the next release, the latest development builds of the Ecosystem and applications from Red Pitaya GitHub repository (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya) are available at http://archives.redpitaya.com/devel/ (https://web.archive.org/web/20150812084942/http://archives.redpitaya.com/devel) and can be manually installed:

- by extracting the content of the <ecosystem>.zip archive to the root directory of an empty SD card;
- by extracting the content of the <application>.zip archive to the /www/apps/ directory of the SD card.

Please note that with these latest development versions you get the latest features, but on the other hand, things can be broken at times and proper operation cannot be guaranteed.

Red Pitaya command line utilities

Note: Command line utilities must not be used in parallel with a WEB application.

Signal generator utility

The Red Pitaya signal generator can be controlled through the generate (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/tree/master/Test/generate) command line utility, but be aware it interferes with the GUI based Oscilloscope & Generator application. Usage instructions (see Table 7 as well):

```
redpitaya> generate
generate version 0.90-299-1278

Usage: generate   channel amplitude frequency <type>

       channel     Channel to generate signal on [1, 2].
       amplitude   Peak-to-peak signal amplitude in Vpp [0.0 - 2.0].
       frequency   Signal frequency in Hz [0.0 - 6.2e+07].
       type        Signal type [sine, sqr, tri].
```

Table 7: Parameters of Signal generator utility

Name	Type	Range	Description
channel	int	1 / 2	Output channel selection
amplitude	float	0 - 2 [V]	Maximal output signal is 2 V peak to peak
freq	float	0 - 62000000* [Hz]	Frequency can be generated from 0 Hz (DC signal) on*.
<type>	string	sine / sqr / tri	Optional parameter. Signal shape type (sine – sine wave signal, sqr – square signal, tri – triangular signal). If omitted, sine is used.

* To generate smooth signals, not exceeding Back-End bandwidth, limitations are:

- 62 MHz (62000000) for sine wave
- 10 MHz (10000000) for square and triangular waves

The output can be disabled by setting the amplitude parameter to zero.

Example (2 Vpp square wave signal with 1 MHz on channel 1):

```
redpitaya> generate 1 2 1000000 sqr
```

Note that the signal generator output impedance is 50 Ω. If user wants to connect the output of the signal generator (OUT1, OUT2) to the Red Pitaya input (IN1, IN2), 50 Ω terminations should be connected at the Red Pitaya inputs through the T-type connector.

Signal acquisition utility

The signal from Red Pitaya can be acquired through the acquire (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/tree/master/Test/acquire) command line utility. It will return raw samples from the ADC buffer to standard output, with no calibration compensation. Usage instructions (see Table 8 as well):

```
redpitaya> acquire
acquire version 0.90-299-1278

Usage: acquire  size <dec>

       size     Number of samples to acquire [0 - 16384].
       dec      Decimation [1,8,64,1024,8192,65536] (default=1).
```

Table 8: Parameters of Signal acquisition utility

Name	Type	Range	Description
size	int	0 - 16384	The number of samples to read.
dec	int	1, 8, 64, 1024, 8192, 16384	Optional parameter. It specifies the decimation factor. If omitted, 1 is used (no decimation).

Acquire utility will return the requested number of samples with decimation factor for both input channels (column 1 = Channel1; column 2 = Channel2).

Example (acquire 1024 samples with decimation 8):

```
redpitaya> acquire 1024 8
  -148     -81
  -143     -84
  -139     -88
  -134     -82
   ...
```

Saving data buffers

It is recommended to use an NFS share to store any temporary data (e.g. the measured signals using the acquire utility). Use a standard *mount* command to mount your NFS share (example):

```
redpitaya> mount -o nolock <ip_address>:/<path>  /mnt
```

The /opt file-system on Red Pitaya, representing the SD card, is mounted read-only. To save the data locally on Red Pitaya redirect the acquisition to a file in the /tmp directory. The /tmp directory resides in RAM and is therefore volatile (clears on reboot).

```
redpitaya> acquire 1024 8 > /tmp/my_local_file
```

Alternatively, save the data directly to the NFS mount point:

```
redpitaya> acquire 1024 8 > /mnt/my_remote_file
```

Copying data - Linux users

In case NFS share is not available, you can use secure copy:

```
redpitaya> scp my_local_file <user>@<destination_ip>:/<path_to_directory>/
```

Alternatively Linux users can use graphical SCP/SFTP clients, such as Nautilus for example (explorer window). To access the address line, type [CTRL + L] and type in the following URL: sftp://root@<ip_address>

Nautilus address bar.png

Figure: Nautilus URL/address bar.

Type the Red Pitaya password (next Figure). The default Red Pitaya password for the root account is »root«. For changing the root password, refer to buildroot configuration (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/blob/master/OS/buildroot/config) - a mechanism for building the Red Pitaya root file-system, including the /etc/passwd file hosing the root password.

Nautilus password window.png

After logging in, the main screen will show the directory content of Red Pitaya's root filesystem. Navigate to select your stored data and use the intuitive copy-paste and drag & drop principles to manipulate the files on Red Pitaya (see next Figure).

Nautilus root fs.png

Copying data - Windows users

Windows users should use an SCP client such as WinSCP (https://web.archive.org/web/20150812084942/http://winscp.net/download/winscp518setup.exe). Download and install it, following its installation instructions. To log in to Red Pitaya, see example screen in next Figure.

WinSCP login screen.png

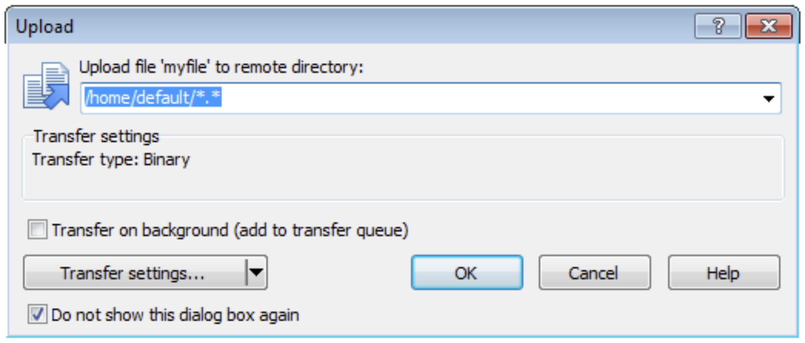
Figure: WinSCP login screen.

After logging in, the main screen will show the content of the Red Pitaya root filesystem. Navigate to select your stored data and use the intuitive copy-paste and drag & drop principles to manipulate the files on Red Pitaya (see next Figure).

WinSCP directory content.png

Figure: Directory content on Red Pitaya.

Select the destination (local) directory to save the data file to (see next Figure).


Upload

Upload file 'myfile' to remote directory:

/home/default/*.*

Transfer settings
Transfer type: Binary

☐ Transfer on background (add to transfer queue)

☑ Do not show this dialog box again

Figure: Select file copy destination.

Alternative access to the instrument

Apart from the usual WEB access, Red Pitaya can be accessed from a computer using standard utilities such as scp and ssh. Accessing Red Pitaya from Matlab® can be done using the Plink (PuTTy Link) (https://web.archive.org/web/20150812084942/http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) utility, a free and open-source command line network connection tool.

Signal generator from Matlab

The generator control can be done by executing the following command:

```
unix('plink -l <username> -pw <password> <IP_address> /opt/bin/generate <channel> <amplitude> <frequency> <type>');
```

The example below presents the setting of the Red Pitaya generator from a Linux PC.

```
unix('plink -l root -pw root 10.0.1.221 “/opt/bin/generate 1 2 500000 sine”');
```

The example below presents the setting of the Red Pitaya generator from a Windows PC.

```
dos('plink -l root -pw root 10.0.1.221 “/opt/bin/generate 1 2 500000 sine”');
```

Signal acquisition from Matlab

The signal acquisition can be done by executing the following command:

```
file=['plink -l <username>  -pw <password> <IP_address> /opt/bin/acquire <size> <dec>'];
```

The example below presents the acquisition of the signals from Red Pitaya to a Linux PC.

```
command=['plink -l root -pw root 10.0.1.221 “/opt/bin/acquire 16384 1”'];
[c,data]=unix(command);
data=str2num(data);
```

Acquisition of the signals from Red Pitaya to a Windows PC:

```
command=['plink -l root -pw root 10.0.1.221 “/opt/bin/acquire 16384 1”'];
[c,data]=dos(command);
data=str2num(data);
```

Acquisition of the signals from Red Pitaya to an NFS share:

```
command=['plink -l root -pw root 10.0.1.221 “/opt/bin/acquire 16384 1 > /mnt/data.txt”'];
unix(command);
```

Accessing system registers

The system registers can be accessed through the monitor (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/tree/master/Test/monitor) utility. Usage instructions:

```
redpitaya> monitor
monitor version 0.90-299-1278

Usage:
       read addr: address
       write addr: address value
       read analog mixed signals: -ams
       set slow DAC: -sdac AO0 AO1 AO2 AO3 [V]
```

Example (system register reading):

```
redpitaya> monitor -ams
#ID     Desc            Raw     Val
0       Temp(0C-85C)    a4f     51.634
1       AI0(0-3.5V)     1       0.002
2       AI1(0-3.5V)     13      0.033
3       AI2(0-3.5V)     1       0.002
4       AI3(0-3.5V)     2       0.003
5       AI4(5V0)        669     4.898
6       VCCPINT(1V0)    55c     1.005
7       VCCPAUX(1V8)    9a9     1.812
8       VCCBRAM(1V0)    55d     1.006
9       VCCINT(1V0)     55b     1.004
10      VCCAUX(1V8)     9ab     1.813
11      VCCDDR(1V5)     809     1.507
12      AO0(0-1.8V)     2b0000  0.496
13      AO1(0-1.8V)     150000  0.242
14      AO2(0-1.8V)     2b0000  0.496
15      AO3(0-1.8V)     220000  0.392
```

The –ams switch provides access to analog mixed signals including Zynq SoC temperature, auxiliary analog input reading, power supply voltages and configured auxiliary analog output settings. The auxiliary analog outputs can be set through the monitor utility using the –sadc switch:

```
redpitaya> monitor -sdac 0.9 0.8 0.7 0.6
```

Accessing FPGA registers

Red Pitaya signal processing is based on two computational engines: the FPGA and the dual core processor in order to effectively split the tasks. Most of the high data rate signal processing is implemented within the FPGA building blocks. These blocks can be configured parametrically through registers. The FPGA registers are documented in the RedPitaya HDL memory map (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/blob/master/FPGA/release1/doc/RedPitaya_HDL_memory_map.odt?raw=true) document. The registers can be accessed using the described monitor utility. For example, the following sequence of monitor commands checks, modifies and verifies the acquisition decimation parameter (at address 0x40100014):

```
redpitaya> monitor 0x40100014
0x00000001
redpitaya>
redpitaya> monitor 0x40100014 0x8
redpitaya> monitor 0x40100014
0x00000008
redpitaya>
```

Note: The CPU algorithms communicate with FPGA through these registers. Therefore, the user should

be aware of a possible interference with Red Pitaya applications, reading or acting upon these same FPGA registers. For simple tasks, however, the monitor utility can be used by high level scripts (Bash, Python, Matlab...) to communicate directly with FPGA if necessary.

Application development

For detailed instructions about new application development, please refer to Red Pitaya Development Guide.

Calibration

Due to natural distribution of the electrical characteristics of the analog Front-End and Back-End electronics, their offsets and gains will differ slightly across various Red Pitaya boards and may change during time.

The calibration coefficients are stored in EEPROM on Red Pitaya and can be accessed and modified with the calib (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya/tree/master/Test/calib) utility:

```
redpitaya> calib
calib version 0.90-299-1278

Usage: calib [OPTION]...

OPTIONS:
 -r    Read calibration values from eeprom (to stdout).
 -w    Write calibration values to eeprom (from stdin).
 -f    Use factory address space.
 -d    Reset calibration values in eeprom with factory defaults.
 -v    Produce verbose output.
 -h    Print this info.
```

The EEPROM is a non-volatile memory, therefore the calibration coefficients will not change during Red Pitaya power cycles, nor will they change with software upgrades via Bazaar or with manual modifications of the SD card content.

Example of calibration vector readout from EEPROM with verbose output:

```
redpitaya> calib -r -v
FE_CH1_FS_G_HI = 45870551      # Front-End Channel1 full scale coefficient for High gain (LV) jumper configuration.
FE_CH2_FS_G_HI = 45870551      # Front-End Channel2 full scale coefficient for High gain (LV) jumper configuration.
FE_CH1_FS_G_LO = 1016267064    # Front-End Channel1 full scale coefficient for Low gain (HV) jumper configuration.
FE_CH2_FS_G_LO = 1016267064    # Front-End Channel1 full scale coefficient for Low gain (HV) jumper configuration.
FE_CH1_DC_offs = 78            # Front-End offset of Channel1 [ADC samples].
FE_CH2_DC_offs = 25            # Front-End offset of Channel2 [ADC samples].
BE_CH1_FS = 42755331           # Back-End Channel1 full scale coefficient.
BE_CH2_FS = 42755331           # Back-End Channel2 full scale coefficient.
BE_CH1_DC_offs = -150          # Back-End offset of Channel1 [DAC samples].
BE_CH2_DC_offs = -150          # Back-End offset of Channel2 [DAC samples].
```

Example of the same calibration vector readout from EEPROM with non-verbose output, suitable for editing within scripts:

```
redpitaya> calib -r
         45870551           45870551          1016267064          1016267064                   78                   25
```

You can write the first N elements of the calibration vector (N = 2 in the example below, N <= 10) to EEPROM with:

```
redpitaya> echo "45870552  45870553" | calib -w
```

Should you bring the calibration vector to an undesired state, you can always reset it to factory defaults using:

```
redpitaya> calib -d
```

Front-End offset calibration

Front-End (input) offset calibration is achieved by modifying elements 5 & 6 (FE_CHx_DC_offs) of the calibration vector, executing the following simple steps:

- Terminate the desired channel with 50 Ω termination or short circuit the desired input channel (connect probe to GND).
- Acquire a set of raw ADC samples using the acquire (https://web.archive.org/web/20150812084942/ https://github.com/RedPitaya/RedPitaya/tree/master/Test/acquire) utility and filter out the desired channel/column:

```
acquire 1000 > samples.txt
```

- Calculate the negative mean of the acquired samples, which is the offset required for compensation:

```
redpitaya> CH1offset=$(cat samples.txt | awk '{ sum+=$1} END {printf ("%.0f\n", -sum/1000)}')
redpitaya> CH2offset=$(cat samples.txt | awk '{ sum+=$2} END {printf ("%.0f\n", -sum/1000)}')
```

- Write this offset to calibration EEPROM using the calib utility. You should only change elements 5 and/or 6 of the calibration vector for Channel1 and/or Channel2 respectively. The following will read the current calibration vector, change element 5 (Channel1 Front-End offset) to value 139, and write the whole vector back to EEPROM:

```
calib -r | awk -v offset=${CH1offset} '{$5=offset}1' | calib -w
calib -r | awk -v offset=${CH2offset} '{$6=offset}1' | calib -w
```

- Here is your new calibration vector readback:

```
redpitaya> calib -rv
FE_CH1_FS_G_HI = 45870551
FE_CH2_FS_G_HI = 45870551
FE_CH1_FS_G_LO = 1016267064
FE_CH2_FS_G_LO = 1016267064
FE_CH1_DC_offs = 139
FE_CH2_DC_offs = 25
BE_CH1_FS = 42755331
BE_CH2_FS = 42755331
BE_CH1_DC_offs = -150
```

```
BE_CH2_DC_offs = -150
```

The following code snippet will do all of the above for Channel1:

```
#!/bin/sh
calib -d
calib -rv
N=1000
acquire 1000 > samples.txt
# Offset is the negative mean of ADC samples
CH1offset=$(cat samples.txt | awk '{ sum+=$1} END {printf ("%.0f\n", -sum/1000)}')

calib -r | awk -v offset=${CH1offset} '{$5=offset}1' | calib -w
```

The script above can be expanded to a proper calibration script with user input interaction regarding channel selection and synchronization between manual input termination and signal acquisition. Everyone is welcome to contribute by joining the development through Red Pitaya GitHub (https://web.archive.org/web/20150812084942/https://github.com/RedPitaya/RedPitaya) repository.

Support and contact information

To learn more about technical specifications, signal connections, applications and command line utilities, please visit Red Pitaya webpage (https://web.archive.org/web/20150812084942/http://www.redpitaya.com/) or contact us (https://web.archive.org/web/20150812084942/http://www.redpitaya.com/services/).

Retrieved from "http://wiki.redpitaya.com/index.php?title=User_Manual&oldid=680"

- This page was last modified on 21 July 2015, at 10:51.
- This page has been accessed 114,759 times.
-