# Red Pitaya FPGA Examples Documentation

***Release 1.0***

**Red Pitaya**

**Aug 27, 2017**

# Contents

Red Pitaya is a Zynq7 FPGA – based low cost electronic board with many components such as two core ARM processor, fast ADCs, fast DACs, USB, LAN, etc. In many respects Red Pitaya is similar to the Arduino or Rasbery Pi with large community of enthusiasts and increasing collection of open-source material. What makes Red Pitaya even better are two fast ADCs, two fast DACs and, most of all, the programmable logic or field-programmable-gate-array (FPGA). With on-chip FPGA Red Pitaya could be used for high performance computing, state-of-the-art measurement system, signal processing and much more. Having both linux-based processing system and programmable logic Red Pitaya is an ideal board for introduction to the FPGA programming and ultimately for building powerful professional and non-professional projects such as radar, radio systems, vector-network-analyzer, etc.

CHAPTER 1

# Setup Your Machine

Setup a development platform for FPGA programming using Xilinx's Vivado software.

- `Vivado 2016.4`
- Pavel Demin's page (for Vivado 2016.2).

CHAPTER 2

# Simple Examples

## A simple LED blinker

This is the simplest example – a hardware equivalent of the Hello World – LED blink. There is no need to know any hardware description language ( HDL ), like VHDL or verilog for this example.

More details can be found here.

## Knight Rider Lights

This example introduces the basic concepts of FPGA programming, you will get familiar with Verilog language which will be used to create your own module. Testing will be done in Vivado's simulator.

More details can be found here.

## Stopwatch

This example shows how communication between the Linux processing system and the programmable logic works.

More details can be found here.

## Frequency Counter

This example demonstraits how to use Red Pitaya's 125 Msamples/s 14-bit ADC and DAC peripherals with the FPGA.

More details can be found here.

## Pulse generator

This example demonstraits how to build a simple pulse generator for the Red Pitaya. The design uses standard Xilinx IP blocks and a custom Verilog core that outputs a signal valid that is driven high when the pulse is played on the DAC.

Pulse shape is stored in a Block RAM that can be written from Linux. ADC data is written in a FIFO that can be read from Linux. The FIFO only accepts ADC data when a pulse is played on the DAC.

More details can be found here.

CHAPTER 3

# Advanced Examples

## SDR Receiver

The implementation of the SDR receiver is quite straightforward:

- An antenna is connected to one of the high-impedance analog inputs.
- The on-board ADC (125 MS/s sampling frequency, 14-bit resolution) digitizes the RF signal from the antenna.
- The data coming from the ADC is processed by a in-phase/quadrature (I/Q) digital down-converter (DDC) running on the Red Pitaya's FPGA.
- The I/Q data is transmitted via TCP to the SDR programs such as SDR# and HDSDR.

The tunable frequency range covers from 0 Hz to 50 MHz.

The I/Q data rate is configurable and four settings are available: 50, 100, 250 and 500 kSPS.

More details can be found on Pavel Demins site.

## SDR Transceiver

The SDR transceiver consists of two SDR receivers and of two SDR transmitters.

The implementation of the SDR receivers is quite straightforward:

- An antenna is connected to one of the high-impedance analog inputs.
- The on-board ADC (125 MS/s sampling frequency, 14-bit resolution) digitizes the RF signal from the antenna.
- The data coming from the ADC is processed by a in-phase/quadrature (I/Q) digital down-converter (DDC) running on the Red Pitaya's FPGA.

The SDR receiver is described in more details at this link.

The SDR transmitters consist of the similar blocks but arranged in an opposite order:

- The I/Q data is processed by a digital up-converter (DUC) running on the Red Pitaya's FPGA.

- The on-board DAC (125 MS/s sampling frequency, 14-bit resolution) outputs RF signal.
- An antenna is connected to one of the analog outputs.

The tunable frequency range covers from 0 Hz to 60 MHz.

The I/Q data rate is configurable and five settings are available: 20, 50, 100, 250, 500 and 1250 kSPS.

The basic blocks of the digital down-converters (DDC) and of the digital up-converters (DUC) are shown on the following diagram:

More details can be found on Pavel Demins site.

## SDR Transceiver Compatible With HPSDR

The High Performance Software Defined Radio (HPSDR) project is an open source hardware and software project that develops a modular Software Defined Radio (SDR) for use by radio amateurs and short wave listeners.

This version of the Red Pitaya SDR transceiver makes it usable with the software developed by the HPSDR project and other SDR programs that support the HPSDR/Metis communication protocol.

This SDR transceiver emulates a HPSDR transceiver with one Metis network interface module, two Mercury receivers and one Pennylane transmitter.

The HPSDR/Metis communication protocol is described in the following documents:

- Metis - How it works
- HPSDR - USB Data Protocol

More details can be found on Pavel Demins site.

## Pulsed Nuclear Magnetic Resonance

The system consists of one in-phase/quadrature (I/Q) digital down-converter (DDC) and of one pulse generator.

The tunable frequency range covers from 0 Hz to 60 MHz.

The I/Q data rate is configurable and five settings are available: 25, 50, 250, 500, 2500 kSPS.

More details can be found on Pavel Demins site.

## Multichannel Pulse Height Analyzer

Spectrometry example.

**Some interesting links on spectrum analysis:**

- Spectrum Analysis Introduction
- Gamma Spectrometry Software

More details can be found on Pavel Demins site.

## Vector Network Analyzer

Some interesting links on network analyzers:

- VNA Operating Guide
- Three-Bead-Balun Reflection Bridge
- Ham VNA
- Fundamentals of Vector Network Analysis Primer
- Introduction to Network Analyzer Measurements

More details can be found on Pavel Demins site.

## Debian With Red Pitaya Ecosystem

More details can be found on Pavel Demins site.

## Signal decimation using a compensated CIC filter

Decimation is the process of converting a time-domain signal sampled at frequency $f_s$ into a signal sampled at frequency $f_s/R$, where R is the decimation rate. Downsampling alone, i.e. keeping one sample every R samples is not sufficient. A low-pass filtering step must precede downsampling to prevent aliasing.

More details can be found here.

## Pulse-Density Modulator

Pulse-density modulation (PDM) is an attractive alternative to pulse-width modulation (PWM) in applications where the PWM technique creates unwanted spikes in the signal spectrum.

More details can be found here.

## Synchronize a cluster of Red Pitayas

In its standard configuration, the Red Pitaya uses an on-board 125 MHz crystal to feed the 125 MSPS ADC and the 125 MSPS DAC. This example shows how to synchronize multiple Red Pitayas on the same clock using the SATA connector (daisy-chain) available on the Red Pitaya.

More details can be found here.

## Examples Utilizing Extension Boards

Red Pitaya becomes even more versatile with when custom extension boards are attached to it, hence we show a couple of examples that utilize extension boards below.

## Simple Coherent Laser Sensor

The guys at Koheron designed a small laser board which fits on top of the Red Pitaya.

More details can be found here.

## Doppler lidar velocimeter

Doppler effect is what we experience when hearing the siren of a moving vehicle: the tone gets higher when the vehicle is approaching.

The frequency of the acoustic wave emitted by the siren is affected by the movement of the vehicle. This effect is not only true for sounds but also for optical waves: when a laser beam with frequency $f_0$ is reflected off a moving target with velocity v, its frequency is shifted of a quantity

$$\Delta f = 2v/c * f_0,$$

where c is the speed of light. The factor 2 occurs because the light is not emitted by the target but reflected off it. In this example laser emits at a frequency of 193.5 THz (i.e. a wavelength of 1550 nm). A target moving at 1 m/s would shift the laser frequency by 1.29 MHz.

More details can be found here.