RXAMAPITAYA - DIGITAL SSTV RECEPTION WITH THE RED-PITAYA

Ties Bos - PA0MBO

April 15th, 2016

Abstract

This paper describes the software RXAMAPITAYA that turns the credit card sized open source instrument Red-Pitaya (oscilloscope, specrum analyzer) into a full fledged Software Defined Radio (SDR) hf-receiver fully equiped to decode digital SSTV signals (hamDRM standard). It can be used for monitoring DRM SSTV channels continously on a power budget of less than 10 W.

Keywords: Digital SSTV - Hamradio - DRM - Linux - Red-Pitaya

INTRODUCTION

The popularity of Slow Scan Television (SSTV), the exchange of pictures in QSO's on the ham bands in a bandwidth of less than 3 kHz, is rising. This rise can be attributed to a number a favourable factors:

- the growth of digital photography (every smartphone now has a good camera).
- digimodes in general (SSTV is but one of the many digimodes) have become much easier to implement: commercial ham equipment generally offers standard interfacing to computer equipment for audio and control signals.
- a wealth of (mostly free) programs is available for the various digimodes
- the ham Digital Radio Mondiale (DRM) standard for SSTV delivers the pictures at the receiving end in the same quality as the originals (if reception is successful).

Quite a number of hams nowadays maintain a web page where they display their received pictures live. These pages are not only nice to look at, but they also are valuable to monitor the band conditions and open up the possibility to obtain reception reports instantaneously.

Running such a web page requires continuous operation of a receiver and a computer connected to the internet. This document describes how both these functions can be performed by a single low cost measurement system called the Red-Pitaya. This credit card sized board incorporates a fast Analog to Digital Converter (ADC), a Field Programmable Gate Array (FPGA) and a ZynQ 7010 processsor. The ADC and the FPGA are programmed to operate as a SDR-receiver. The Zynq processor runs the operating system Linux and a program to decode the pictures from the signal the FPGA is tuned to.

The design builds on the extensive work of Pavel Demin [1] for the SDR receiver part and runs a modified version of the program RXAMADRM [2] for the decoding of the signal.

For DRM-SSTV three different programs are available and they are compatible with each other: EasyPal for windows [3] , QSSTV for linux [4] and TRXAMADRM also for linux [2]. All three programs use audio signals to and from the soundcard of a computer to send and receive the binary picture data. With the advent of SDR receivers/transmitters this audio route of the picture data (or other digital data for the other digimodes) still has to be followed due to the requirements of the programs. A trick to circumvent the use of real cables is to use socalled ”virtual audio cables”, software that behaves like an audio device (soundcard) that a program can send data to or read data from. In this way two separate programs can exchange binary data.

In the setup described here this inefficient double conversion to and from audio is circumvented: the SDR-receiver in the FPGA produces Inphase and Quadrature signals, I/Q - data pairs which are directly processed to the picture data in the modified RXAMADRM program which is now called RXAMAPITAYA.

OVERVIEW OF RXAMAPITAYA

The SDR receiver is implemented in the FPGA of the Red-Pitaya. Its schematic diagram is given in figure 1. The binary code is in the file **myrx31.bit**. RXAMAPITAYA loads this code into the fpga, sets the frequency and starts the receiver.

The DSP code in the FPGA comprises the following processes. The ADC (analog to digital converter) digitizes the antenna signal from channel 0 at a sample rate of 125 MHz. This digitized signal is

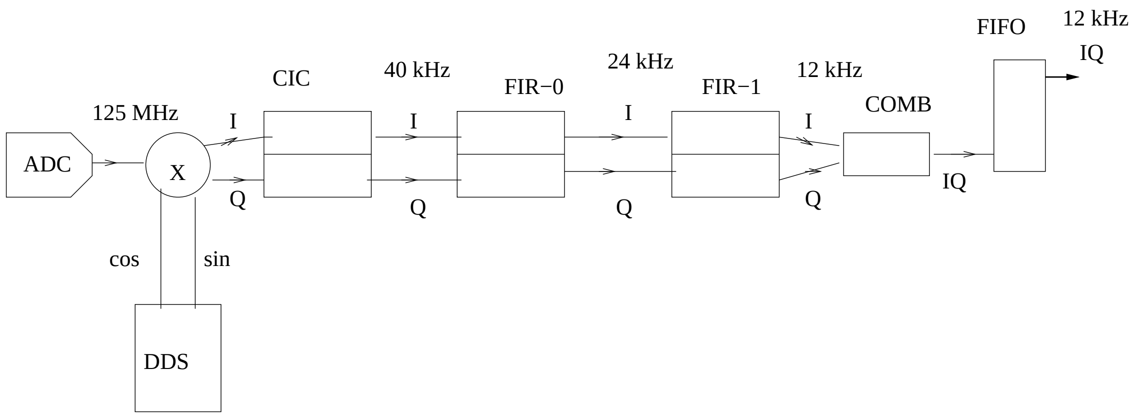

Figure 1: SDR receiver in FPGA

multiplied by a cosine and sine signal produced in the DDS (Direct Digital Synthesizer). In the next stages the sample rate is reduced in three steps from 125 MHz to the final 12 kHz. Each reduction of the sample rate is preceeded by lowpass filtering to comply with the Nyquist criterion. The final bandwidth at the output of the FPGA is around 3 kHz. The filtering is identical for the I (Inphase) and Q (Quadrature) signals. The output is delivered to the Zynq processor via a FIFO buffer (First In First Out). The FPGA delivers this output to the Zynq processor at a fixed memory address together with a count of the number of I/Q samples that are available in the FIFO at the time of reading this count. In this way synchronization with the data stream can be accomplished.

The processing of the I/Q datastream at a sample rate of 12 kHz is performed in the program **drmtst** that runs in the Zynq processor under linux. This program reads the I/Q data from the FIFO and processes the data to a picture file. Its operation is described in the document [5]. However there are some differences. First of all there is no conversion of audio to I/Q data. This version of **drmtst** can operate directly on the I/Q data. Moreover the SDR receiver in the FPGA has to be tuned. This information is read by **drmtst** as a command line parameter.

The decoded pictures are saved by the program in a subdirectory called **picsrx** and are also presented to the user in a graphical user interface (GUI) together with information about the progress of the reception. This presentation task is performed by a separate program with the name **rxamadrm.tcl** which is written in the script language

Tk/Tcl. It also provides the way to set the rx-frequency and based on this either a lower or upper sideband version of the program drmtst is chosen to process the data.

INSTALLATION AND USE

The software is available as a compressed image with which a 4GB microSDcard can be prepared. This 4GB SDcard can then be inserted into the Red-Pitaya and after applying power it will run a Linux Debian version for the Zynq processor composed by Pavel Demin [1]. After booting it starts an SSH-server and a VNC-server automatically to enable remote access to the system.

Preparation of the 4GB SDcard is performed with the following steps:

1. Insert the card into a PC running some linux distribution
2. Now find out how this card is mounted with the following commands:
   ```
   sudo su <return>
   mount <return>
   ```
   Look for entries like /dev/sdx on /media/.../... where x is some character like b,c,d, etc. If theres is more than one partion on the same device , i.e. /dev/sdf1 and /dev/sdf2 write down both /media entries.
3. dismount these /media-entries, i.e. with
   ```
   umount /media/....
   umount /media/....
   ```
4. Now untar the SD-card image and copy it to the SD-card with:
   ```
   tar -xvzf rxamapitayav0_4.tgz
   dd if=rxamapitayav0_4.img of=/dev/sdx bs=4M
   ```
   with sdx the devicename you saw in the mount command (use a single character, i.e. /dev/sdf in the former example). Take great care to get the correct device name or you will destroy your running linux system. Wait for the dd-command to finish.

Now you can remove the SD-card from the guest system and put it into your Red-Pitaya and power up the latter after connecting it to a local network. Also connect an antenna to the ADC connector at the far left of the front side. To access the system now running on the Red-Pitaya use a VNCViewer on the guest computer. For this you need to know its IP-address it has gotten from the DHCP server of this local network. You have to find out what it is by accessing this DHCP server on your router. The password is ”changeme”.

To start and view the operation of the SDR-receiver and the decoding program **rxamadrm**, use the following commands in the terminal that opened in your VNCVIEWER:

```
cd rxamapitayav0_4/linux <return>
./startdrm.sh <return>
```

This will show a waterfall window and a window which shows the progress during the reception of a DRM-SSTV signal. At the bottom of this window you can make a choice between three standard DRM-SSTV frequencies. Operate this button at least once after starting the program to make sure is has chosen the right sideband program.

DRM-SSTV is not a small signal mode. You will need rather strong signals (S7 and up) for succesful decodes. As the ADC has a high Z input, best results are obtained with a high Z antenna connection. A standard dipole will not give the best results. For a dipole antenne use some stepup hf-transformation.

With the button "FTP" you can set up the communication to a web page on the internet that can display the images you received without user interaction by filling out the form and setting ftp on. If you are afraid to be embarrassed by showing all received pictures you can monitor the pictures coming in yourself and upload only the wanted ones by pressing the "UPLOAD LAST" button of this window.

RESULTS

The system has been used to monitor the 40 meter band DRM-SSTV channel of 7058.00 kHz succesfully over the run of a few weeks. The ADC channel 0 was connected to an endfed wire antenne 20 m long at a height of about 4 - 6 meters. Normal and hybrid mode pictures were received from PA-, DL-, F- and I-land on this antenna. For comparison trxamadrmv3_6 was run on a normal PC connected to a FT2000 via a soundcard interface. The antenna connected to the FT2000 was a W3DZZ dipole also at about 4 - 6 meters height.

Both systems gave more or less the same number successful decodes as can be seen from table 1 for the evening of April 22nd 2016.

References

[1] pavel-demin.github.io/red-pitaya-notes.

[2] http://pa0mbo.nl/ties/public_html/hamradio/rxamadrm.

[3] http://www.vk3evl.com.

[4] users.telenet.be/on4qz.

[5] http://pa0mbo.nl/ties/public_html/hamradio/rxamadrm/rxamanew.pdf.

Table 1: Comparison rx-results Red-Pitaya vs FT2000 (number of pictures received)

Station	Red-Pitaya - end fed 20 m wire	FT2000 - W3DZZ
OE1GOW	33	36
DB2HS	4	5
DD9FY	6	7
DK3UD	12	11
Dl6BL	6	6
DL7VOE	3	-
F4BMT	1	3
F4EFL	2	-
F5LWD	1	-
G8IC	10	18
IZ1MKE	11	7
M0HWM	1	1
M0JGM	1	-
M0KLL	1	2
M0SSBB	5	-
OE3ODW	7	10
PA3ADE	1	-
PA0VER	-	3
PD0CIF	1	-
PE1ACB	8	10
OE2KNS	-	1
IZ3JZP	-	1