A software solution for decoding and extracting files from GOES LRIT/HRIT streams.

Raydel Abreu Espinet, CM2ESP. January 3rd 2018.

(cm2esp@gmail.com)

INTRODUCTION:

GOES-16, formerly known as GOES-R was launched in November 2016. It represents a major leap forward and a huge technology advantage over the previous generation of NOAA's geostationary weather satellites. In comparison with older GOES-NOP birds, these satellite offers 16 spectral bands for imagery and can cover the whole America area from north to south many times faster than its predecessors.

The change of technology also came with an increase of the data rate needed. The legacy LRIT digital stream which once replaced the old analogue WEFAX was now changed with the HRIT stream with a speed of about 400kbit/s, almost four times higher than the 128kbit/s LRIT has.

Luckily for the amateur community and other weather satellite enthusiasts. The RF and software world in general has also evolved, and with the rise and diversification of the SDR technology, nowadays digital software based receivers are available at very affordable and competitive prices. Many SDR designers are focused in the Open Source paradigm so it is possible to easily share, download, modify and use wide range of software tools.

NOAA's and the GOES-R team also share (in part) that principle, and the very initial version of the first HRIT decoder was published in their website (goes-r.gov) alongside with its source code. However, their software was written for a single hardware solution, and the published source code was a very early version with many bugs, errors and outdated. Later, that specific version and its source code were removed from the website and replaced by a Linux "Live" version.

However the software presented on this document is based almost entirely on that old version. A huge number of minor complexity bug fixes and improvements were made. The main difference is that while that old version used only an USRP as input, the presented software compilation maintains only the core GOES frame processing functions, and uses instead as an input a network TCP socket which accepts "hard symbols" (1 and 0). Viterbi and Manchester decoding is done on the software, this allows that any external software with TCP features can be used for BPSK receiving, and even hardware based universal modems can be used.

Therefore the software solution presented here is a combination of OpenSource solutions made by a number of different authors. And Raydel, CM2ESP is ONLY author of the minor improvements and modifications made to those, and credit is wanted and mandatory to be extended to each of those authors.

- Viterbi solution is a simplified variant of Phil Karn, KA9Q original code, plus some minor changes to better support Visual C++ and IL code.
- Manchester (NRZ-M) decoding is from "*Project Desert Tortoise*" and was adapted for stream based sequentially input and for Visual C++ too.
- The old GOES processing base class was written for NOAA by AVTEC, code was released public under GNU/GPL on goes-r.gov website. AVTEC was absorbed by a larger company and disappeared some years ago, and the software solution and source code was removed from the goes-r website just few months ago.

CONCEPT OF OPERATION:

The presented decoding solution is intended to Microsoft Windows users and thought to be a simple install and run solution, to reach a greater number of interested persons on weather satellite receiving. Windows was chosen because many people found complex to begin to dig into the Linux world, although the original software solution was made and is run for Linux. As a radio frontend and BPSK receiving a Windows version of GNURadio is used for interfacing the SDR hardware with the decoding software. R. Alblas's Xrit2Pic is required for visualization of the received LRIT files.

Receiving chain goes as follow:

1- Antenna and L-band feedhorn for GOES LRIT/HRIT (1691/1694 MHz)
2- LNA and optional down-converter.
3- SDR hardware.
4- GNURadio, with osmocomm source block as input, supports: RTL-SDR, HackRF One, AirSpy, USRP, others might as well, but is not properly documented on the Internet. Personally I have test it with RTL-SDR and HackRF One. But HackRF needs special attention due to the big DC peak it has.
5- XRITDecoder software, opens a TCP client connection with GNURadio and takes the hard symbol synced BPSK stream, does Viterbi decoding and performs Manchester decoding (only needed for GOES-16 HRIT, not used for GOES-NOP LRIT), then CADU frames are processed and files are generated.
6- Xrit2Pic is used to display files and save images or animations.

REQUIRED HARDWARE:

All of the following requirements are "general expectations", less or more depends in great deal of the user specific location, and RF environment. RF noisy environments or in vicinity of LTE towers may need special filtering. A total minimum of 5dB SNR at the input of the SDR is needed for proper decoding, so as long as you can reach that minimum you are OK. However 6 to 7 dB SNR can provide best results.

- Antenna with a minimum gain of 21dB, parabolic dish (1 meter, 3 foot minimum) is recommended, but it depends of geographic location.
- Linear polarization feed for 1691/1694 MHz. HRPT circular polarization feeds might be used but an additional loss of 3dB should be expected due to wrong polarization.
- Low Noise Amplifier (LNA), over 20 dB should be fine, less can be used if big dish is used. 1dB Noise Figure.
- OPTIONAL, A downconverter unit is required if the used SDR can't work properly at 1.7 GHz or has a significant loss of sensibility at that high frequencies.
- SDR hardware depends of user availability, RTL-SDR and HackRF One has been tested, and in theory AirSpy, USRP and many others can work as well as supported by Osmocomm source block in GNURadio.
- Computer with 64bit processor. 4 GB RAM and Core i3 processor with at least 1.8 GHz speed, hard disk size depends of the amount of information desired to store. GOES HRIT can consume several GB of information per day.

SOFTWARE REQUIREMENTS:

Software version was tested to work with the present release. Future versions of the utilities and libraries may or may not work.

- Windows 64 bit version. Tested on Windows 8, but should work as well on 7 and 10.
- Microsoft .Net Framework 4.0[1]
- Visual C++ 2010 Runtime[2]
- Gnuradio 3.7.11 for Windows[3]
- Xrit2Pic[4]

1- https://www.microsoft.com/en-us/download/details.aspx?id=17718
2- https://www.microsoft.com/en-US/download/details.aspx?id=5555
https://www.microsoft.com/en-us/download/details.aspx?id=14632
(Both 32bit and 64bit versions are needed)

3- http://www.gcndevelopment.com/gnuradio/downloads.htm
(Use "64-Bit Any CPU" and always latest version)
4- http://www.alblas.demon.nl/wsat/software/winsoft_msg.html
(Use always latest version)

INSTALLATION PROCEDURE:

1. Download the required software, GNURadio and Xrit2Pic
2. Make sure all libraries pre-requirements are already installed (.Net Framework 4 and VC++ 2010 Runtime)
3. Install GNURadio, default installation path directory should be used (C:\Program Files\GNURadio-3.7), and is NOT recommended to change it to avoid additional modifications to the start script.
4. Download and install your required SDR driver/software, example: Zadig.
5. Decompress and copy the decoding software with other accompanying files to (C:\XRITDecoder)
6. Edit LAUNCHER.bat and modify/set frequency, sampling rate according to your SDR hardware.
    a. For RTL-SDR set 1152e3 as sample rate for best results, and frequency as needed.
    b. HackRF and AirSpy devices only has a limited number of sampling rates allowed, verify your hardware and adjust accordingly.
7. Install and configure Xrit2Pic. Set (C:\LRITTEMP) as the input folder for Xrit2Pic as this is where XRITDecoder stores the received files.
8. Run LAUNCHER.bat and monitor for incoming files and signal quality.
9. Run Xrit2Pic and wait for files image segments to complete.

Notes:

- If the installation path of GNURadio is changed, go to the Start Menu, look for “GNURadio Command Prompt” copy the lnk file (right clic, copy), go to XRITDecoder folder, remove file “gnuradio.lnk” and paste the file here, then change the name of it to “gnuradio.lnk”. Alternatively it is possible instead of replace gnuradio.lnk, edit it, by right clic on it, Properties, Shortcut tab and then change carefully the path to the GNURadio installation path.
- It is possible that in some computers the program fails due to the lack of security permissions to write on C: disk. In that case, make sure to run XRITDecoder with administrative privileges, by right clic on LAUNCHER.bat, Run as administrator.

DISCLAIMER AND ACKNOWLEDGEMENT:

- The software is a combination of several Open Source solutions, each one with its corresponding author. Raydel, CM2ESP is ONLY author of the combination of those parts and from the modifications which were made. As previously stated, there is no intention to steal credit for the work made by others.
- The software and its output files are ONLY for AMATEUR NON COMMERCIAL AND DEFINITELY NOT FOR PROFESSIONAL USE. Please don’t charge or ask money for it. Please don’t use the images for operational or Live-Threating situations, unless approved or certified by a qualified authority.
- The assembly, testing and upgrades performed to create this software took over one year of hard work and effort, although it would have been not possible to do it without the work of each of the components authors, it wouldn’t be possible neither without the perseverance and time devoted by the compilation author, Raydel, CM2ESP, and only is requested to give proper credit if the result is being used in any public media, like Internet, social nets, TV, etc.