Receiving the Meteor-M N2 LRPT Downlink

with an RTL-SDR Dongle

Raydel Abreu Espinet, CM2ESP, Havana, Cuba (cm2esp@gmail.com)

Recently, on July 8, 2014, the latest Russian polar-orbiting weather satellite was launched. It is **Meteor-M N2**, which came with new upgrades and some fixes learned from its predecessor, **Meteor-M N1**, as well as with improved radiometers and new technologies to be tested.

Similar to the N1 spacecraft (no longer operational), **Meteor-M N2** transmits all six channels obtained by its radiometers in HRPT format on 1700 MHz. However, the frame format is different from other HRPT satellites, so a suitable decoder is required. Alongside with these HRPT transmissions, there is also an LRPT (Low Rate Picture Transmission) digital downlink in the 137 MHz band, the primary channel on 137.100 MHz and the secondary one on 137.900 MHz.

From the six available channels (three visual and three infrared), the ground control can select a combination of three to be sent on the LRPT VHF downlink. A special unit called the *LRPT Adapter* takes 8 scan lines from the HRPT raw data from each of the chosen three channels, and applies a JPEG compression algorithm to create a packet structure which is then digitally transmitted at a pre-established speed of 72 or 80 kiloSymbols per second. Unfortunately, this unit is of a new design—slightly different from the one previously flown in *Meteor-M N1*—and it exhibits a failure which produces data dropout at equal time intervals (usually 6.5 minutes). This is seen as white bands in the received LRPT pictures, though this trouble does not affect the HRPT data. Finally, the LRPT signal is QPSK modulated, and a 5-watt transmitter feeding a QFH (Quadrifilar Helicoidal) antenna broadcasts it to direct readout users on the ground.

However, due to the digital nature of the signal, an analogue FM receiver such as those used for APT is not suitable to receive this new satellite. Previous satellites, like Metop-A, have flown with LRPT transmitters, but the Russian LRPT format, although based on Metop's LRPT structure, is not exactly the same: so professionals decoders designed for Metop LRPT cannot be used for *Meteor-M N2*.

A technology named SDR (Software Defined Radio) can, however, be used to replicate part of the typical hardware. This system may be easily programmed and adapted to multiple kinds of signal. During the time of *Meteor-M N1*, software applications were designed to decode its transmissions. These programs could read quadrature recordings from WAV files obtained from any suitable SDR hardware, and then demodulate the signals and extract the images. But as the *Meteor-M N2* LRPT format is slightly different, the pictures obtained are not perfect. The program, actually a combination of experimental tools, is therefore not suitable for fully decoding *Meteor-M N2* transmissions. However, one of these tools, **LrptRx.exe**, can still be used to extract the QPSK constellation coordinates. These coordinates are known as **Soft-Symbols**.

Another satellite enthusiast, Oleg Bekrenev, developed his own program, **LRPTOffLineDecoder**, which reads a soft-symbol input file and processes it to extract the images, specifically for *Meteor-M N2*. This program can also combine the three visible channels to produce a colour composite image, a very useful way to obtain some resemblance to true colour RGB imagery.

How, then, can we actually proceed to receive *Meteor-M N2* signals that can be processed with Oleg's decoder?

Receiving the LRPT Signals

Well, there are two approaches. One is based on existing *Windows* software and includes many different steps; another procedure makes use the free Open Source software platform named **GNU-Radio** and the **Linux** environment. Many users unfamiliar with the *Linux* operating system may initially be afraid to try the second approach, but as we are going to see during this article, the *Linux* procedure is actually easier, simpler and faster.

For the Windows procedure, almost any SDR device can be used, as long as its bandwidth is enough to handle the almost 100 kHz wide signals from the LRPT downlink, and its software supports base-band WAV file recordings.

Many different types of SDR hardware can be used to record *Meteor-M N2* signals prior to its processing. As far as this author knows, the *FunCubeDongle Pro+*, *SDR-IQ*, and *Perseus-SDR* have all been used successfully. But we are going to focus in one particular kind of SDR hardware, the cheap **RTL-SDR** dongle based on the **RTL2832U** chip.

The RTL-SDR Dongle

This low-priced dongle was designed for DVB-T digital TV reception, connection to a PC being done using a USB port. However, it was later discovered that dongles based on the RTL2832U chip can be commanded to stream raw I/Q data to a host application and therefore be used as a simple and inexpensive SDR receiver. The use of these devices for amateur weather satellite images in the APT domain has been discussed previously in GEO-Quarterly 40 and 41. But with Meteor's LRPT transmissions marking the arrival of the 'digital age' for direct readout users, this dongle has come to the fore once again.

In order to obtain images from the *Meteor-M N2* LRPT downlink with an RTL-SDR dongle there are two steps: first, receive and record the transmission; second, process the recording to extract images. The second step is common to whichever procedure you have chosen to record the signal.

Step One: Recording with RTL-SDR under Windows

There are many diverse software packages compatible with RTL-SDR, but **SDRSharp** [1] is probably the simplest and most intuitive. Installing and configuring *SDRSharp* is not included here, but much information may be found on-line, as well as in my article published in *GEO Quarterly No 40*.

Due to the internal characteristics of the RTL-SDR dongle, the best results are obtained when using sample-rate values between 0.900 Msps and 2.0 Msps, 1.024 Msps being a good, balanced value. In slow computers 0.900 Msps can be a better choice.

It is recommended to use **off-set tuning** to avoid any DC spike, as well to avoid any loss of information derived from the I/Q imbalance. Choosing a **manual gain** value and **disabling AGC** (Automatic Gain Control) also allows getting better results. The best gain value depends of the dongle's tuner, the antenna, and the surrounding environment. To overcome the poor sensitivity and selectivity of the RTL-SDR dongle, installing a mast-mounted, properly tuned preamplifier with associated band-pass filter is a good idea. However, in low-noise and interference-free locations, with a good quality antenna, the satellite's downlink may be strong enough at least near TCA (time of closest approach). Of course, in locations suffering from heavy in-band interference, the use of a preamp can make matters a lot worse. Frequency calibration is not really required as long as the frequency difference does not exceed 5 kHz. But if you need to adjust it, the best way is to use a reference station of known frequency, and change the ppm until the displayed frequency of the station matches the closest to its true value.

Always be sure that 'Correct IQ' is selected and enabled on the Radio Panel in the left hand section of *SDRSharp*.

Prior to attempting a recording, several passes should be studied in order to find the optimal gain values and configurations. To successfully receive data, the satellite downlink should have a signal to noise level exceeding 15 dB, and solid decoding is obtained above 20 dB.

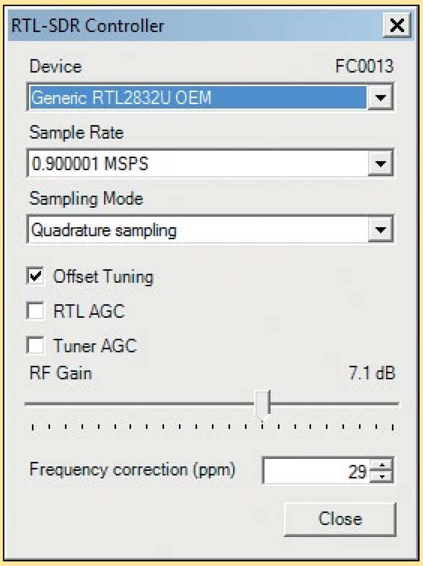

Figure 1
The SDRSharp RTL-SDR configuration window, showing a typical configuration suitable for both LRPT and APT. Off-set tuning is selected, AGC is disabled and the Gain value was adjusted to get the best SNR without over-loading.

There is no need to perform any sort of frequency correction due to the Doppler effect because the carrier recovery and tracking algorithm in the QPSK demodulation process, which will be explained later, will sweep and lock into the signal, and follow it as it shifts in frequency.

Once we have everything ready to record, one important element which cannot be forgotten is to select 137.100 MHz (or 137.900 MHz if the secondary transmitter is active instead) as the '**centre frequency**': the VFO frequency is not important.

In order to do this, perform a 'left-click' over the spectrum panel (not the waterfall), and keep holding the mouse button down. Now drag the mouse pointer left or right and a small label will appear next to it indicating the 'centre frequency'. When this shows the correct value, release the left-click button.

Figure 2 (next page) shows the main SDRSharp window, with the *Meteor-M N2* LRPT downlink at the centre of the spectrum.

Because we are going to record the received RF signal and the entire spectrum, we don't have to worry about which audio demodulator is selected. Volume is not important either.

Proceed now to the **Recording plug-in** on the left panel, select 'Baseband', leave 'Audio' unmarked and select '8 bit PCM' as the sampling format. The RTL2832U chip's internal ADC (Analogue to Digital Converter) has only 8 bit resolution, so recording at higher values will simply consume higher space on your hard-disk without any improvement in quality.

Wait until a *Meteor-M N2* pass begins and, after the signal has exceeded the 10 dB SNR value, start the recording. Once the signal has dropped or the satellite pass ended, stop the recording. *SDRSharp* may be closed now if desired, because there is no need for it until the next pass. The plug-in usually stores the recorded WAV file in the SDRSharp folder.

Processing the WAV File

Before starting to process the recorded WAV file, you must 'condition' it. Because we recorded the entire pass-band, only a small fraction of it contains the *Meteor-M N2* signal: the rest is just noise or other unwanted signals, so some sort of filtering is required. To do so we are going to perform a type of Low-Pass Filter, which means reducing the sample-rate of the WAV file to remove the unwanted higher frequency components. When handling quadrature recordings, the centre of the recorded spectrum is represented by lower frequencies and the edges by higher frequencies. If we reduce the sampling-rate to 130 kHz, we are going keep only a band-span of such value, which will be mostly occupied by the desired *Meteor-M N2* signal, and only a small portion of it being noise.

There are plenty of audio handling tools which may be used, but **Audacity** [2] is a simple, free program, and it can be installed or used as a 'stand-alone' application.

To resample the audio with *Audacity*, first load the WAV file into it. You will find at the program's bottom-left corner a textbox labelled '*Project Rate Hz*' which indicates the current sampling rate of the WAV file. Click on it and change this value to '130000' (i.e. 130 kHz), and then press the ENTER key. Now save the file by opening the *File Menu* and clicking on '*Export*'. Select '**WAV Signed 16 bit PCM**' as the file format, change the file name if desired, and click '*Save*'. Figure 3 shows *Audacity* with a typical *Meteor-M N2* WAV file after the resampling process. You can now discard and delete the original file because only the resampled version is needed.

Next in the requirements is to process the WAV file to **demodulate the QPSK** transmission and extract the 'soft-symbols'. For this purpose we are going to use one of the original tools designed for *Meteor-M N1*, the program **LrptRx.exe**, which can be downloaded from

https://www.dropbox.com/s/qq1fjyitpa3j14o/software.zip

as a file called software.zip, which contains three programs: but only **LrptRx.exe** is needed. This software can read the WAV file and process it: the program's output is later analysed by Oleg's decoder to create the image.

LrptRx

The interface of **LrptRx** is quite complex, but there are only a few parameters which need to be adjusted. First, go to the left section and open the WAV file at the '*Input Filename*' box. Although not essential, you can change the output filename and its location, otherwise the program will simply store it on drive C:\ of your hard drive, with the indicated name. Right between the two boxes there is a check mark labelled '**I/Q Swap**'. It is **essential** to enable this.

Now hit the '*Run*' button and manually move the progress slider to its mid-point, where the signal should be the best. If you obtain a well defined 4-dots constellation, each dot at the centre of its quadrant, everything is OK. Close the program, re-open it and process the entire file.

If the constellation instead looks like an 'X' or a square, try changing the symbol rate from 72000 to 80000 (in the Symbol Loop configuration panel). If the constellation is now achieved with 4 dots, then the satellite is transmitting on 80 K. Close the program and adjust to this value prior to processing the file. It is important to know that, when the end of the file has been reached, **LrptRx** will continue to run forever unless you manually stop it; so keep an eye on the progress slider and manually click on '*Stop*' as soon as the process is complete, and close the program. You can see **LrptRx** in operation in figure 5 on page 10, with a good-quality constellation (4 green dots) showing in the square window at centre right.

Step 2: Decoding the Soft Symbol file into Images

Now that the symbol extraction is complete, we can proceed to the imaging step. As mentioned earlier, in order to process a soft symbol file and extract the images sent in LRPT format, we need Oleg Bekrenev's **LRPTOffLineDecoder** [5]. It is highly recommended to use the latest version published on his website.

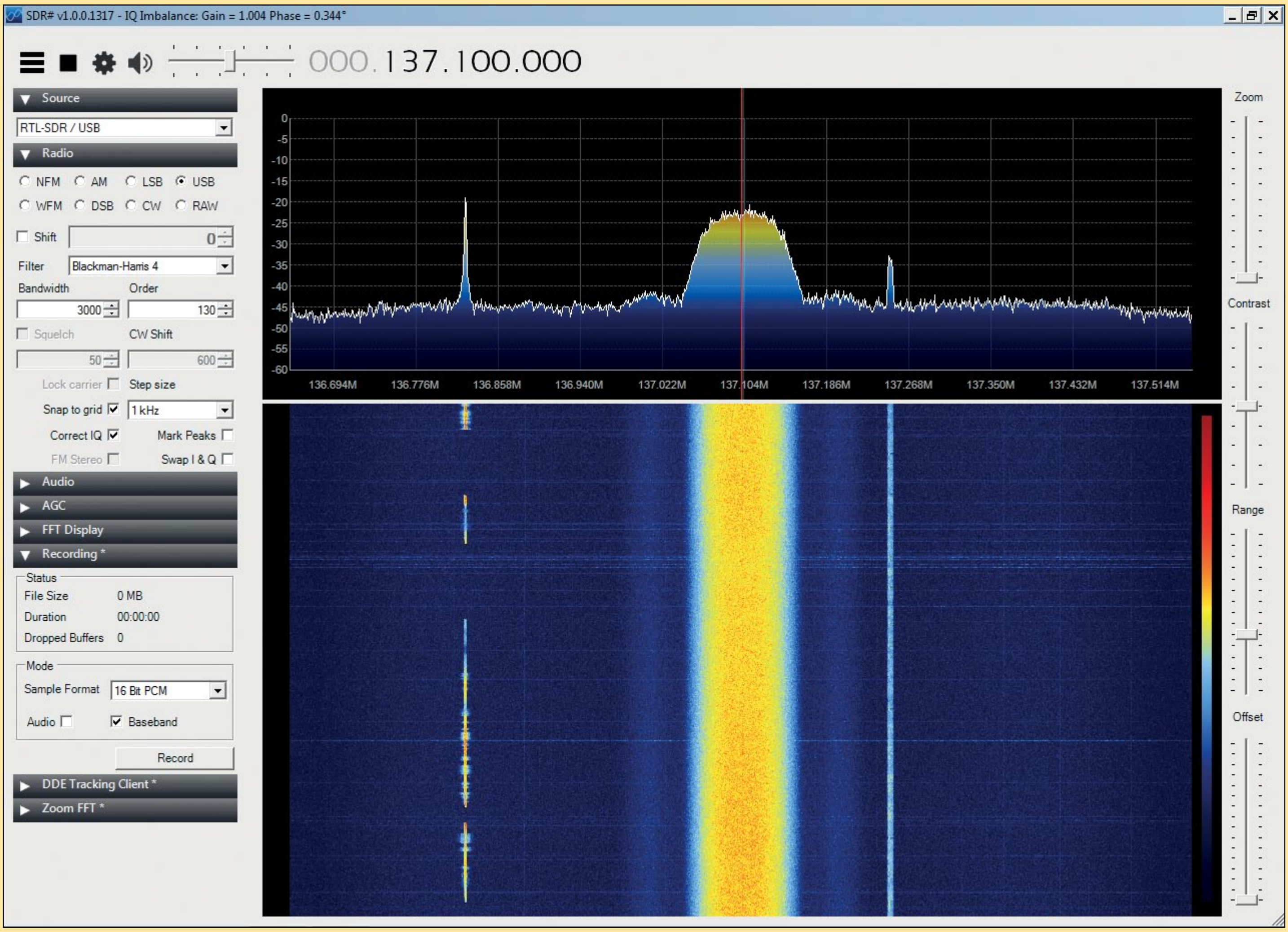

Figure 2 - The SDRsharp window with the Meteor-M N2 LRPT downlink at the centre of the spectrum.

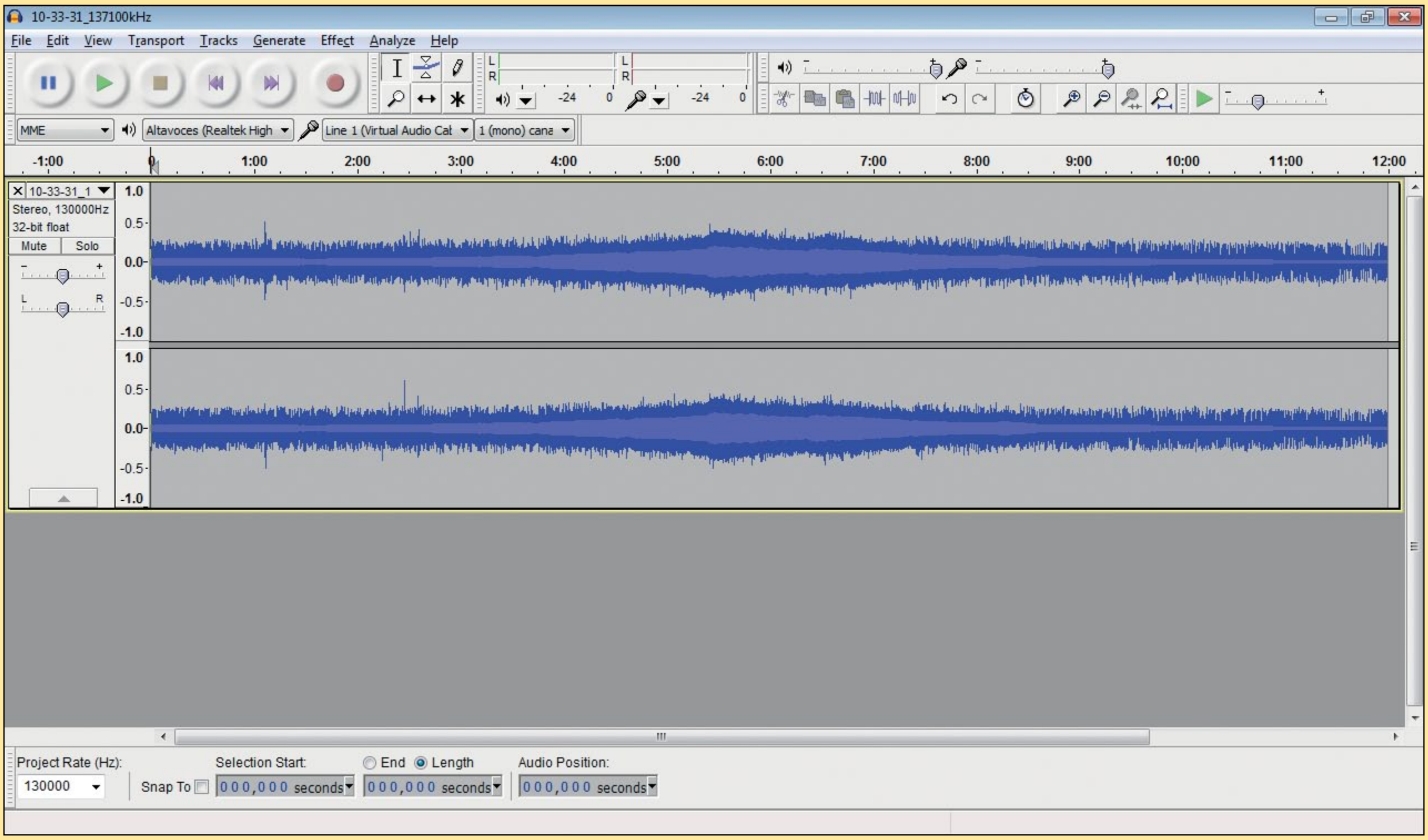

Figure 3 - A view of Audacity and a WAV recording of Meteor-M N2 after the re-sampling process

Open Oleg's decoder and proceed to click on either the '*72K*' or the '*80K*' button, depending on which speed the satellite was using. Normally it should be 72K. The program now asks for the file to open, browse and look for the **.RAW** file generated by **LrptRx**. If you cannot immediately see this file in the browser panel, select '**All *.***' in the file type filter dialogue, and the program will show all the files in the specified folder. Figure 6 shows **LRPTOffLineDecoder** during processing, with images building up.

Right after opening the file, the program will start processing it and the constellation points will be shown as a reference. There are two progress bars: the upper one indicates the current position in the file, while the lower one shows the status of the internal buffer. Under normal conditions the buffer indicator should be full, or almost full, until the end of the file, and it should take around one second of processing to decode 20 seconds of data. But the amount of time required does depend of the data quality. It may take up to 2 minutes to process a typical 12-minute pass. The process can be aborted any time just by clicking at the '*Stop*' button which will be disabled when processing is complete.

LRPTOffLineDecoder only allows you to save single images derived from the output of the combination of three channels. Therefore, if you want to save a colour RGB image, you must select the desired channel for each colour and then click on the **Generate RGB** button. If the transmitted images were the three visual channels, they are chosen by default; otherwise you will need to assign a channel to each colour using the RGB selection panel.

If you want to save an individual channel in grayscale, just select that same channel for R, G and B. This channel selection and image manipulation can be done many times without the need to reprocess the file again. When you press the *Generate RGB* button, a preview window will be shown, where you can inspect the output of the colour composite process. If you want to write this image to your hard disc, it is essential that you click 'Save' at the top left corner of this frame. The picture will be saved in BMP format, in the same folder as the input file. The composite output preview window, showing an RGB colour composite from the three visual channels is shown in figure 7. The two white bands are caused by a defect in the LRPT Adapter unit aboard *Meteor M N2*.

Whether decoding takes place successfully or not, several log files will be written in the same folder for debugging purposes. These files are not needed and can be deleted unless you need to inspect them to troubleshoot decoding issues.

Decoding Revisited

A simpler faster alternative using Linux

Users unfamiliar with the *Linux* environment usually think it may be just too difficult to use and install. But luckily, the **GNURadio** community [3] has compiled a preconfigured version which already possesses all the required dependencies, and which can be installed in an empty flash drive without the need to change anything on your computer: not even installing it on the hard-disk.

The symbol-extraction program currently available for *Linux* is a modified version made by the author, based on a decoding script originally written by Martin Blaho. The great advantage of this method is that the three stages done under Windows—recording, resampling and symbol extraction—are all done simultaneously, in real time, during the satellite pass. So we reduce the amount of time and effort by a factor of three. Just run the program at the beginning of the pass, then close it and copy the output file for subsequent processing. No other step is needed other than to adjust PLL tracking sensitivity and the common RTL parameters.

You will need a USB flash drive with a capacity of at least 4 Gigabytes. Make sure it is empty because it will be formatted and all data on it will be lost. The *Linux* operating system can be downloaded from this URL:

http://downloads.gnuradio.org/releases/gnuradio/iso/

Just, look for the most recent file with the '**.ISO**' extension.

Next, you require a program named **Universal USB Installer** [4] which will allows you to install the *Linux* system on the pen-drive. Connect the empty flash-drive and run the installer program. It first asks you to agree with its terms of use, so click the 'I Agree' button to proceed. In the following window, first select '**Ubuntu**' from the drop-down list then click the 'Browse' button and select the ISO file previously downloaded. Next, select the drive letter of the flash drive where you are going to make the installation. Enable the '*Format*' check box and move the slider to create a '*persistence file*' which will store changes made during operation; otherwise any configuration will be lost after reboot. This file should not be too big or *Linux* will run slowly. A recommended size is between 500 - 1000 MB. Finally, select '*Create*' and wait for the installation process (which will take several minutes) to run. Once completed, you can run Linux any time you want just by configuring your computer to boot from the USB port while the *Linux* flash drive is connected.

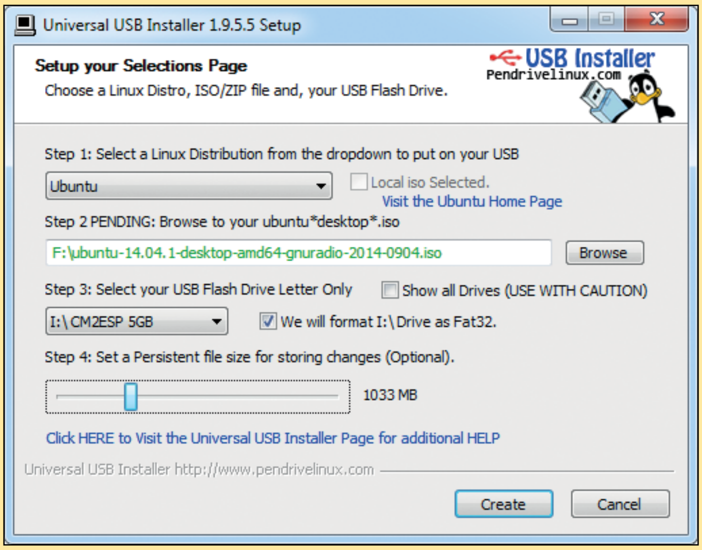

Figure 4 - Universal USB Installer main window with parameters selected

However, the Meteor-M LRPT receiving program is not included in the Linux system and you have to download the **LRPT Soft-Symbol Receiver** from

https://www.dropbox.com/s/8kc89wriludrrb8/meteor_qpsk_rx_rtl.zip

Provisionally, store this on the computer's hard-disk (**not** on the *Linux* flash drive): later, it will be copied into *Linux* (see later).

Now, turn off your computer and connect both your RTL-SDR dongle and the *Linux* flash drive. Turn the power on again and configure your PC to boot from USB. Depending of your mother-board manufacturer and model, different key combinations may be required, so check your PC's instruction manual if you are unsure how to do this.

If everything is correct, *Linux* will start loading from the USB flash drive; when completed, you will find yourself at the *Ubuntu/Linux* desktop. Now you must copy the receiving program (temporarily stored on the PC hard-disk) into *Linux*. Look at the left panel of the *Ubuntu* desktop and click the second button from the top (the one with the file cabinet icon). This will launch the *File Explorer*, which is very similar to *Windows Explorer*. Access your hard-disk and copy the receiving program (the **.PY** file) and paste it into the '**Home**' folder. You only have to do this once, because it will be stored in the persistence file on the pen-drive and can be easily accessed through a short-cut icon that will be created later.

Testing the Receiving Program

To successfully decode any signal from *Meteor-M N2*, you must first test the **LRPT Soft-Symbol Receiver** program and your RTL-SDR dongle under the *Ubuntu/Linux* environment. It is important to adjust the proper RF gain values and the frequency correction parameter. Return to the left panel and click on the third icon from top, which

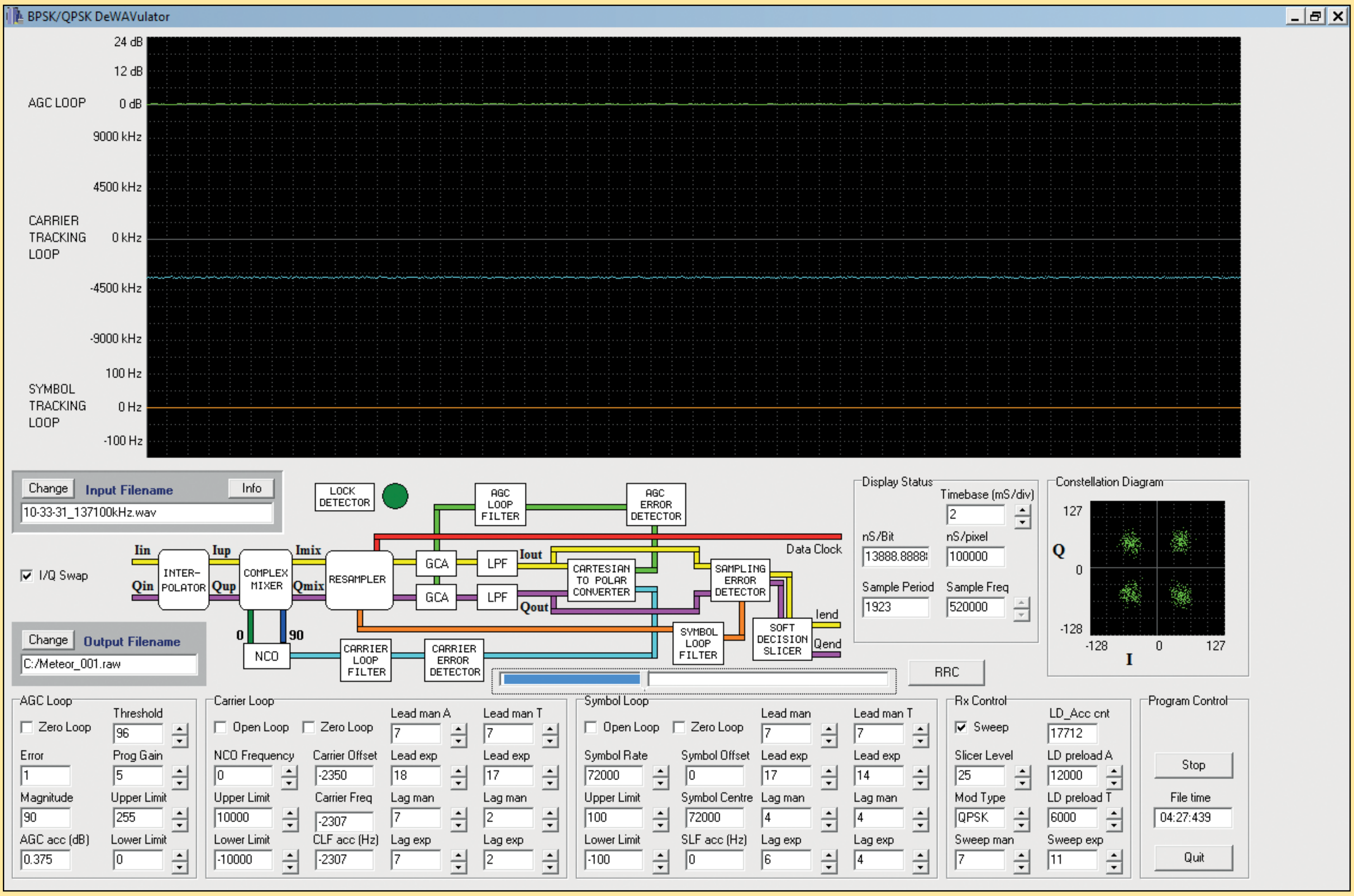

Figure 5 - LrptRx in operation and showing a good quality constellation (4 green dots in the square window at centre right)

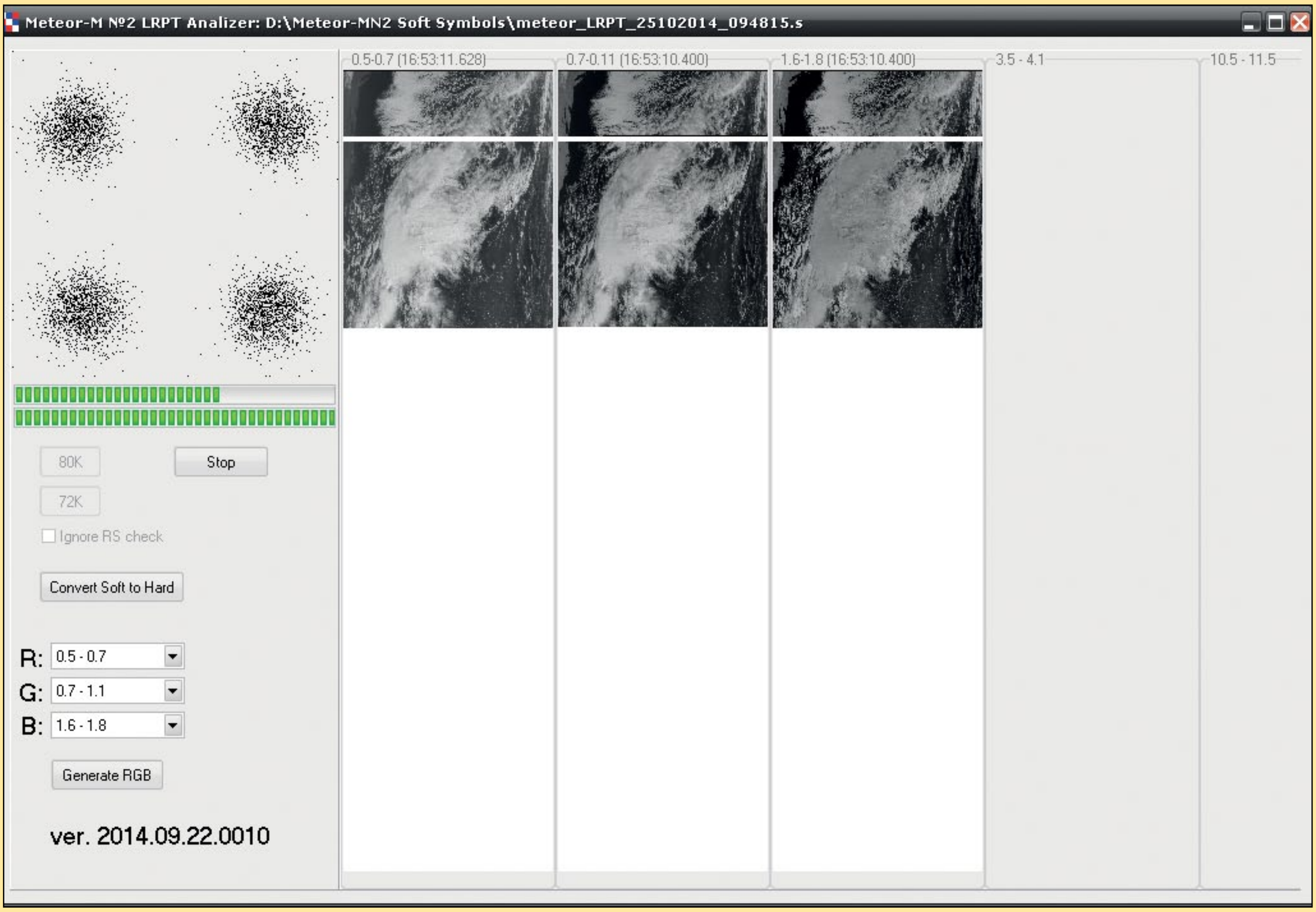

Figure 6 - LRPTOffLineDecoder during the processing of a file, decoding images for channels 1, 2 and 3.

will launch the ‘**Terminal**’. Now type in the following (without the quotes):

“python meteor_qpsk_rx_rtl.py”

and press ‘**Enter**’. If everything is correct, the program will launch. Tune to a known frequency and adjust the gain value and the frequency correction for best results. Take a note of the required values (i.e. write them down somewhere) and close the program. You should be back at the *Terminal*. Now type in the following line again

“python meteor_qpsk_rx_rtl.py --gainrf XX.X --freqcorrect YY”

where ‘XX.X’ is the proper gain value you have just found (42.1, for instance) and ‘YY’ is the frequency correction ppm value (62, for example). Next time you run the receiving program, it will use these values. Check that those numbers are OK, and keep that line in mind, because you will need it to create the short-cut.

Creating the Desktop Short-Cut

The short-cut also only needs to be created the first time: in subsequent boots, all you need do is to double-click on the short-cut and the receiving program will launch.

You need to open a text editor program to write the instructions. Go to the left panel, click the third icon from the top again, type ‘**gedit**’ into the *Terminal* and press *Enter*. A simple text editor, very similar to Window’s *Notepad* will launch. Type in the following lines:

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Type=Application
Name=Meteor-M N2 Rx
Exec=python meteor_qpsk_rx_rtl.py --gainrf XX.X --freqcorrect YY
Categories=Development;
MimeType=application/gnuradio-grc;
Icon=gnuradio-grc
```

If you now hover the mouse pointer over the top edge of the screen, the **File Menu** will appear. Proceed to save the file by selecting **Desktop** from the left hand menu as its location so that the short-cut can be found easily. Write ‘Meteor-Rx.desktop’ into the ‘**Name**’ field as the filename, and click the ‘**Save**’ button at bottom right on the screen. Close the text editor and check if the new file is now present on the Desktop (where it should appear as ‘Meteor-M N2 Rx’). Right-click on it, go to ‘**Properties**’ and open the ‘**Permissions**’ tab and click the ‘**Execute**’ check mark to ‘Allow executing file as program’. Click the ‘**Close**’ button to apply this change.Now you can double-click on it and check that the receiving program launches normally (be patient - it usually takes several seconds to appear).

Once you have carried out this procedure, all you have to do when you want to receive a *Meteor-M N2* pass is to boot your PC from the flash drive, and as soon as the Ubuntu Desktop has loaded, double-click on this short-cut to run the **LRPT Soft-Symbol Receiver** program.

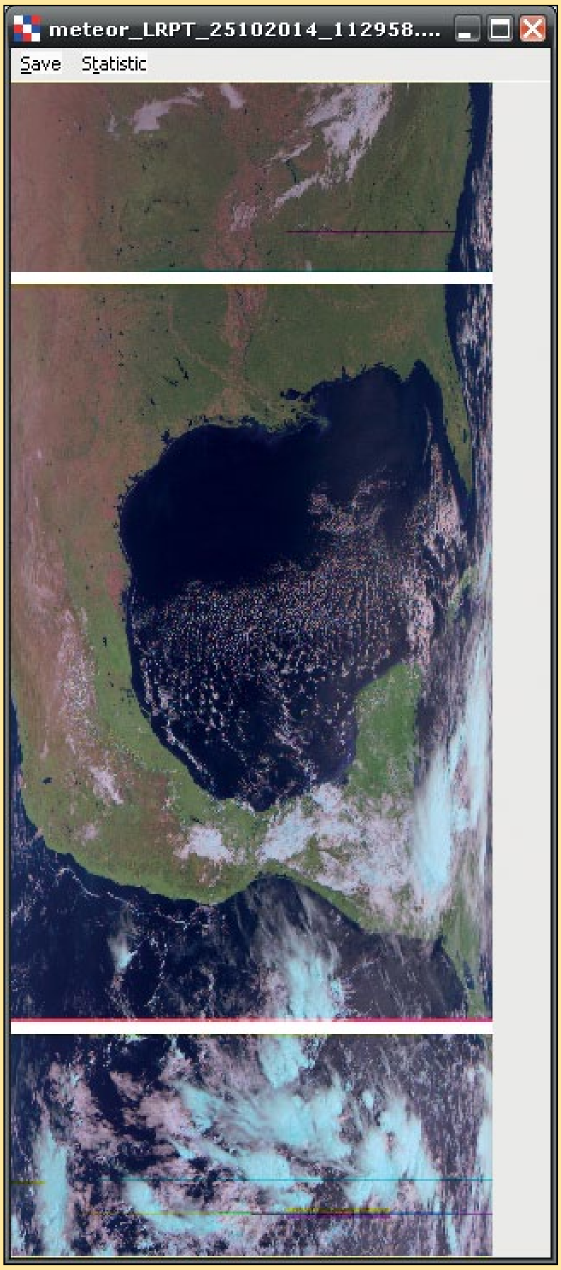

Figure 7 - The image composite output preview window, showing an RGB colour composite created from the three visual channels.

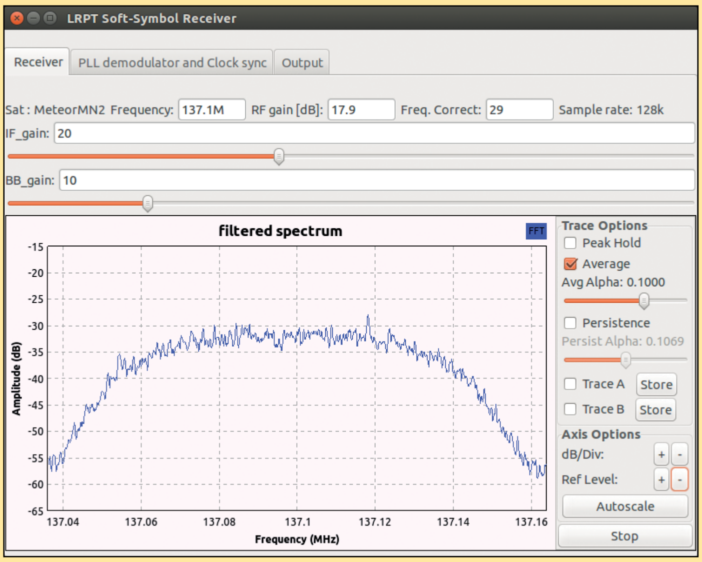

Figure 8 - The spectrum inspection tab of the Meteor-M LRPT receiving program, with a Meteor-M N2 signal present.

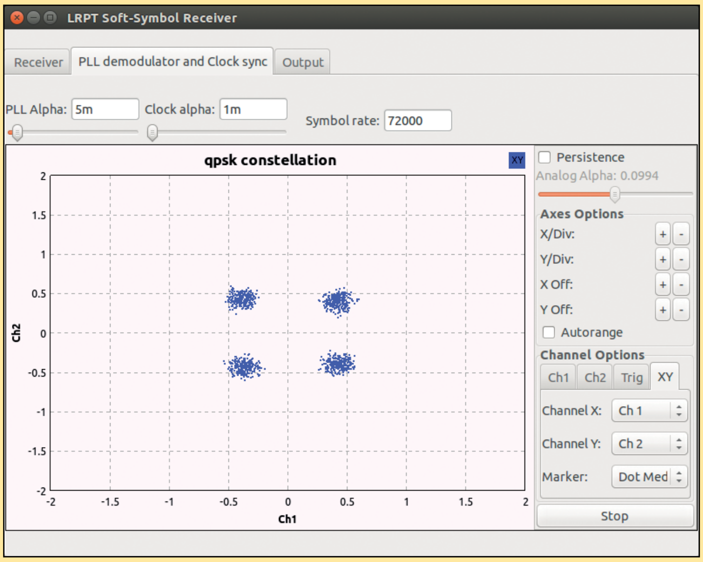

Figure 9 - A well defined and properly decoded 4-dot QPSK constellation displayed in the PLL Demodulator tab.

Recording Meteor .s Files

Every time the **LRPT Soft-Symbol Receiver** program is run it creates a file on the Desktop with extension '**.s**'. During your series of tests, several of these '.s' files will already have been created. Delete them, because '.s' files will not be needed unless a real Meteor-M N2 pass has been received.

When receiving an LRPT signal, initially check the *Receiver* tab which shows the incoming spectrum (figure 8). The *Meteor-M N2* signal will become stronger as the satellite approaches, and once the signal-to-noise ratio is above 10 dB, go to the *PLL modulator and Clock sync* tab which shows the constellation (figure 9). If it is properly formed by its four distinctive dots, even if these are spread out, you are getting a good copy of the LRPT downlink. If it looks like a circle, increase the **PLL-Alpha** value slightly until the constellation forms, then decrease it back to its minimum (1m) and don't change it again. Near the end of the pass, when the signal is too low and the constellation is not forming any more, you can close the program.

The decoded file will be stored on the desktop with a name similar to 'meteor_LRPT_date_timelocal.s'. You need to move this file to the PC hard-disk so that it can be accessed from *Windows*, so right click on it and select 'Cut'. Then open *Ubuntu File Explorer*, and paste the file into a desired location on the hard drive.

Note: In some computers, in order to be able to transfer files in this way, you must first of all enable the *Windows* folder to which you are copying the file for **sharing**. Do this while in *Windows*, and <u>before</u> you boot into *Ubuntu/Linux*.To do this, right-click on the folder concerned, select 'Properties' at the foot of the pop-up menu, open the 'Sharing' tab, and click the 'Share' button.

Note that leaving too many of these files on the *Linux* Desktop will eventually cause the free space in the persistence file of the flash drive to run out (which is why the 'cut' method is recommended).

That's all: Receiving in *Linux* is fast and simple. Just execute the program during a pass then copy the file to your hard-disk, reboot into Windows, and extract the received images using Oleg's **LRPTOffLineDecoder,** which recognises **'.s files'.** This step of the process is exactly the same as that described above for a Windows PC.

Conclusions

During this article we have explored two methods of successfully receiving the digital LRPT downlink from the new *Meteor-M N2* satellite. Although we were focused on the cheap RTL-SDR dongle as a receiving hardware front-end, the *Windows* procedure can be applied to other types of SDR devices, as well as other SDR software. The alternative *Linux* method is only compatible with RTL dongles (so far), but provides a faster and simpler method once we have achieved the slightly difficult first boot and configuration stage.

References

1 http://sdrsharp.com/
2 http://audacity.sourceforge.net/
3 http://gnuradio.org/
4 http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/
5 http://meteor.robonuka.ru/soft/