

## Come ricevere Oscar 100 con Linux

### Premessa

Come qualcuno avrà compreso leggendo i miei scritti in questo sito, utilizzo solo Linux, avendo abbandonato qualsiasi sistema operativo “proprietario” (cioè realizzato da organizzazioni commerciali) da oltre un ventennio. Grazie, infatti, alla grande abbondanza di software ed alla grande diversità di distribuzioni di questo sistema operativo libero, non esiste alcuna necessità di spendere soldi per licenze software usando sistemi operativi “a pagamento”. E per noi Radioamatori, da sempre a sperimentare e provare cose nuove, dovrebbe essere quasi un obbligo avere Linux nel nostro shack.

Ma veniamo al titolo di questo scritto. Cosa usare e che programma installare per ascoltare il nuovo Oscar 100 con il sistema operativo del Pinguino? Ma, anche e soprattutto, che hardware usare per riceverlo?

### Due parole su Oscar 100

Cominciamo subito col dire che, del nuovo satellite, si parla tantissimo: non starò qui, dunque, a dirvi le sue caratteristiche che potreste trovare su tanti altri siti dedicati a lui. Tengo solo a precisare che si tratta di un satellite commerciale pesante diverse tonnellate dove, tra gli altri trasponder, gli amici OM tedeschi sono riusciti ad “imbarcare” anche un trasponder per noi Radioamatori. Il trasponder consente ben 500 KHz di banda bassa (dedicata ad traffico in fonia, telegrafia e alcuni modi digitali) ed una banda alta di alcuni Mhz dedicata esclusivamente a trasmissioni digitali e video.

Inoltre, il nuovo satellite geostazionario permette di usare hardware di facile reperibilità; per la parte ricevente, possiamo riciclare tutto quanto in uso per la TV via satellite e, in generale, qualsiasi parabola da 80 cm in su. Pertanto, anche ad acquistare tutto nuovo, non si va oltre i 50 euro per un impianto decente di ricezione. Per la trasmissione, invece, la spesa è un pochino maggiore ma, in questo caso, è possibile riciclare il materiale informatico usato per le reti Wi-Fi a 2,4 Ghz, dato che l'uplink sul satellite usa proprio la banda dei 13 cm; tuttavia, in questo documento, non parleremo della parte relativa alla trasmissione, cui dedicherò un documento a parte.

### Hardware ricevente

Nel mio caso, stimolato dall'amico Gabriele IK7FMQ, primo collega di ARI Lecce a spingersi sul satellite, ho acquistato il necessario in un grande magazzino, non avendo sotto mano un impianto TV dismesso da riciclare; il materiale comprendeva una parabola da 80 cm con vari accessori di aggancio (25 euro), un illuminatore economico della Fracarro UX-S LTE (appena 5,30 Euro: nonostante il costo esiguo, diventa ben stabile dopo 15/20 minuti), 10 metri di cavo a bassa perdita per satellite e alcuni connettori di tipo “F”. Con un pochino di inventiva ho realizzato un piccolo

circuito per alimentare l'illuminatore attraverso il cavo coassiale (il "bias-tee", visibile in foto), recuperando lo schema dai tanti disponibili in rete; personalmente, ho seguito quanto realizzato dal collega Ivo, I6IBE (<http://www.radioamatoripeligni.it/i6ibe/qo100/qo100.htm>). Gli illuminatori per la ricezione della TV satellitare, infatti, hanno bisogno di essere alimentati attraverso il cavo coassiale: una normale tensione sino da 12 a 14 volt consente di attivare l'LNB nella banda bassa (quella che ci interessa per la ricezione di Oscar 100).



*Il bias tee completo e pronto per il collegamento, realizzato con un pezzo di quadrello*

Una volta montati parabola ed illuminatore, connesso il sistema di alimentazione e predisposta la discesa con il corretto connettore per il dongle SDR USB (nel mio caso, uno per TV), è necessario puntare in maniera corretta l'antenna: per farlo, basta ricorrere ad uno dei tanti siti dedicati al puntamento delle antenne satellitari. Il migliore è certamente [www.dishpointer.com](http://www.dishpointer.com) che permette di spostare sulla mappa il segnaposto per "centrare" la nostra installazione; la riga gialla, tracciata dal software verso il satellite (identificato con "26E ARABSAT 5B") è utile per puntare la parabola usando i riferimenti "a vista" disponibili sul posto; nel mio caso, ho tralasciato il balcone del vicino a 20 metri da me e, su di esso, ho puntato la parabola: dopo, a 42.000 km, in alto sull'orizzonte a sud, c'è il nostro satellite. Il software, in base alle coordinate del proprio QTH fornisce anche la necessaria elevazione. È necessario tener ben presente che un errore di puntamento, anche minimo, si traduce in una drastica riduzione del segnale: pertanto, ho preferito prima testare la ricezione e, se la prova fosse stata soddisfacente, avrei continuato la personale esperienza con la più impegnativa trasmissione. Recentemente, un forte vento a spostato la parabola di pochissimi cm: beh, anche in questo caso, è stato necessario riposizionarla.

La ricezione avviene sulla banda dei 10 Ghz; in realtà, l'illuminatore funge anche da down converter: infatti, riceve la frequenza del satellite, sottrae quella del proprio oscillatore locale e la rimanda sul cavo convertita intorno a 1750 Mhz. Pertanto, alla frequenza del beacon di Oscar 100 (10.489,550 Mhz) bisognerà sottrarre la frequenza dell'oscillatore locale (9.750); ascolteremo, quindi, il beacon del satellite sulla frequenza di 739,550 Mhz e, per farlo, possiamo usare qualsiasi ricevitore. Per praticità ed economicità useremo un Dongle SDR USB da collegare al nostro PC.

## Verifica hardware

Per usare con soddisfazione Linux è necessario verificare, prima di fare un acquisto, se l'hardware è supportato. Nel caso delle chiavette SDR, è sufficiente verificare se sono basate sul chip Realtek RTL2832U. Al contrario, per verificare che la chiavetta che già si possiede sia supportata, installare il pacchetto “rtl-sdr”. Come al solito, fornisco i comandi per le distribuzioni basate su Debian/Ubuntu e derivate:

```
sudo apt-get install rtl-sdr
```

Sarà richiesta la password ed in pochi secondi il pacchetto sarà installato. Inserendo la chiavetta in una porta USB libera e lanciando “rtl\_test -t” avremo:

```
rtl_test -t
Found 1 device(s):
 0: Generic RTL2832U OEM

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7
16.6 19.7 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4
43.9 44.5 48.0 49.6
No E4000 tuner found, aborting.
```

Come si vede, è stato trovato il chip che ci interessa mentre non è un problema il fatto che non è stato trovato il tuner E4000: la chiavetta –quindi– è perfettamente supportata.

Sostanzialmente, il chip RTL2832U permette il campionamento del segnale; la sintonia è invece in carico al chip del tuner: nel mio caso l'integrato R820T della Rafael Micro, in grado –secondo i data sheet– di sintonizzarsi da 24 sino ad oltre 1700 Mhz.

## Il software

Ora che abbiamo la “chiavetta giusta” per il nostro Linux, è necessario installare il software occorrente sulla nostra distribuzione; in breve, tutto sarà affidato alle librerie messe a disposizione da un progetto importante: GNU Radio. Questo progetto è un insieme di librerie che implementano i blocchi necessari per realizzare una SDR, cioè una “radio definita via software”.

Per avere una copia di GNU Radio funzionante è necessario installare, oltre al programma principale, anche tutte le necessarie “dipendenze” (le dipendenze, in Linux, sono tutti i software “di complemento” che servono a far funzionare uno specifico programma, comunemente chiamate “librerie”); nel mio caso, usando Linux Mint, ho verificato quanto era necessario installare ed ho diviso i comandi in modo da intervenire se qualcosa non andava per il verso giusto. Facendo riferimento alla wiki del progetto (<https://wiki.gnuradio.org/index.php/UbuntuInstall>), ecco i comandi da digitare per installare le tante dipendenze di GNU Radio (versione 3.7.2):

Prima di tutto, eliminiamo eventuali librerie non più necessarie

```
sudo apt autoremove
```

Poi passiamo ad installare le nuove

```
sudo apt install cmake git g++ libboost-all-dev python-dev python-mako
sudo apt install python-dev python-mako
sudo apt install python-numpy python-wxgtk3.0 python-sphinx python-cheetah swig
sudo apt install libzmq3-dev python-numpy python-sphinx python-cheetah swig libzmq3-dev
sudo apt install libfftw3-dev libgsl-dev libcppunit-dev doxygen libcomedi-dev
sudo apt install libfftw3-dev libgsl libcppunit-dev doxygen libcomedi-dev libqt4-opengl-dev
sudo apt install libfftw3-dev libgsl libcppunit-dev doxygen libcomedi-dev libqt4-opengl-dev
sudo apt install python-qt4 libqwt-dev libsdl1.2-dev libusb-1.0-0-dev python-gtk2
sudo apt install pkg-config python-sip-dev libqt4-opengl-dev python-lxml
```

Ricordo che il comando “sudo” permette di eseguire i comandi come amministratore: la prima volta che viene usato in una finestra di shell, richiede la password. Se non ci sono errori, un ulteriore comando installerà gnuradio:

```
sudo apt install gnuradio
```

In assenza di errori, possiamo verificare la versione installata del nostro gnu-radio, usando il comando “apt-cache”:

```
apt-cache policy gnuradio
```

che ci informerà sui dettagli del pacchetto:

```
gnuradio:
  Installato: 3.7.2.1-5ubuntu0.1
  Candidato: 3.7.2.1-5ubuntu0.1
```

È anche possibile verificarne il funzionamento lanciando il comando “gnuradio-companion”: se anche questo non restituisce alcun errore, possiamo chiudere la relativa finestra e procedere con l'installazione del programma che useremo come ricevitore.

Il software scelto è “gqrx-sdr”, un ottimo ricevitore realizzato dal collega Alexandru Csete OZ9AEC e rilasciato in open source; il suo sito è gqrx.dk. Si tratta di un software basato sulle librerie di GNU Radio (ecco perché era necessario installarle) e sulle librerie grafiche di QT (la piattaforma QT è usata nel desktop di Linux “KDE”: se, invece, la vostra versione di Linux usa un ambiente desktop diverso, ad esempio Gnome, sarà necessario installarle). Tuttavia, eventuali dipendenze dovrebbero essere installate automaticamente: digitiamo il comando di installazione di gqrx-sdr:

```
sudo apt install gqrx-sdr
```

Se, anche in questo caso, non ci sono errori, possiamo lanciare un programma che si occupa di profilare GNU Radio sul nostro hardware: si tratta del comando “volk\_profile” dove la parola “volk” sta per “Vector-Optimized Library of Kernels” ed è una libreria introdotta in GNU Radio dal 2010. L'esecuzione del comando dura qualche minuto: lasciamolo lavorare.

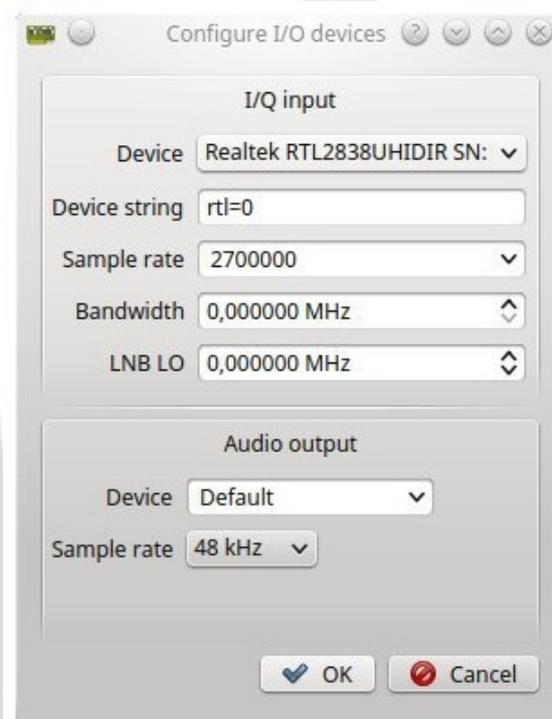
Ora dovremmo quasi essere pronti; tuttavia, dobbiamo fare in modo che, all'inserimento del dongle SDR nella porta USB il sistema del “plug-and-play” di Linux non carichi i driver relativi al sistema

DVB (non dimentichiamo che il chip di Realtech nasce proprio per quello scopo). Per evitare questo inconveniente, dobbiamo mettere in blacklist i relativi driver, in modo che Linux non li carichi: creeremo, quindi un file con alcune righe, con i pochi comandi che seguono:

```
echo "blacklist dvb_usb_rt128xxu" >> /etc/modprobe.d/sdr-blacklist.conf
echo "blacklist dvb_usb_v2" >> /etc/modprobe.d/sdr-blacklist.conf
echo "blacklist rtl2830" >> /etc/modprobe.d/sdr-blacklist.conf
echo "blacklist rtl2832" >> /etc/modprobe.d/sdr-blacklist.conf
echo "blacklist r820t" >> /etc/modprobe.d/sdr-blacklist.conf
```

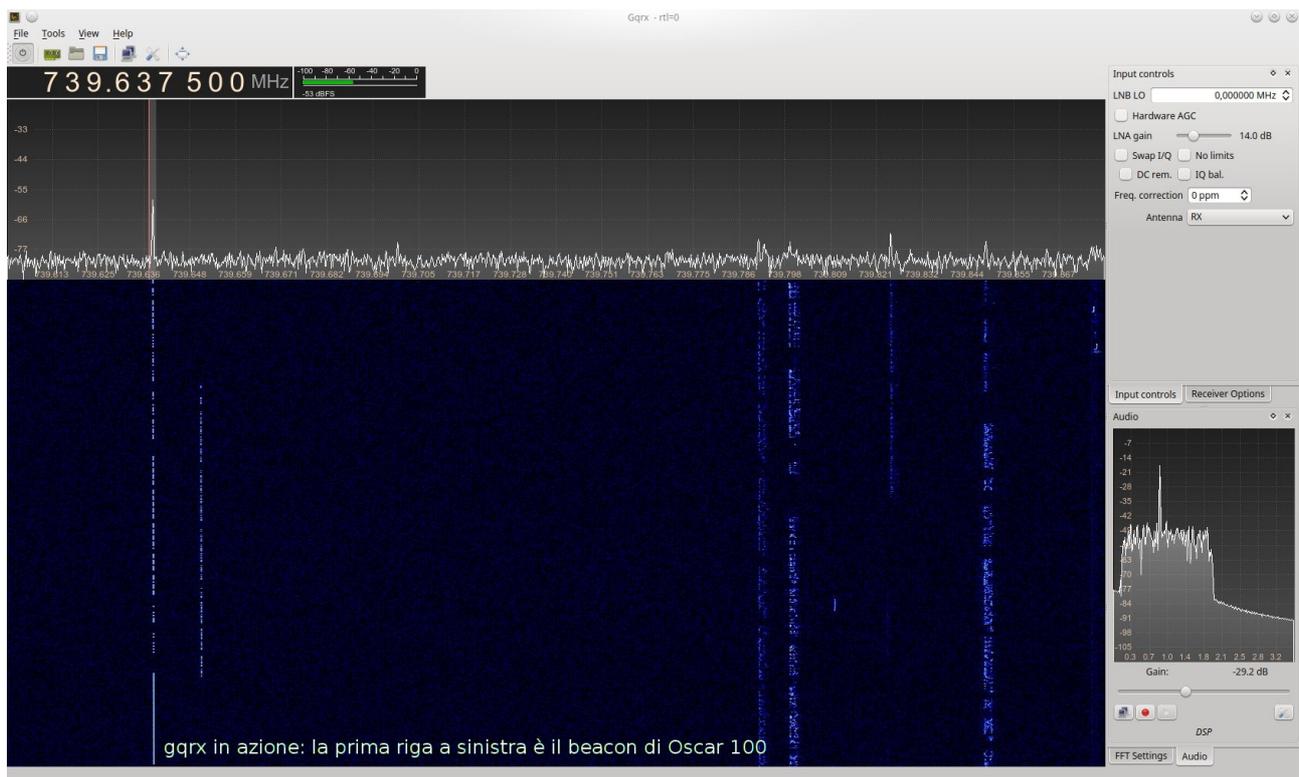
Il comando “echo” stampa una riga sullo schermo; in questo caso, con “>>”, il comando la accoda nel file “sdr-blacklist.conf” salvato nel path “/etc/modprobe.d/” che sarà letto dal sistema nel momento dell’aggiunta di nuovo hardware.

Ora è possibile inserire la chiavetta SDR nella porta USB e lanciare il ricevitore con il comando “gqrX” in una shell (questo ci permette di verificare eventuali errori); avviato il software, compare una piccola finestra a video: impostatela con i valori indicati nell’immagine



*Parametri iniziali di GqrX*

Confermare con il comando OK: apparirà la finestra del ricevitore. Accendetelo (il comando è il primo bottone sotto il “menu file” e sintonizzatelo su 739,638 Mhz e, salvo qualche piccola differenza nella lettura della frequenza, dovremmo vedere una finestra come quella che segue:



*La schermata principale del software di ricezione Gqrx*

La videata è anche personalizzabile: usando la rotella del mouse sulle scale dei Db e delle frequenze è possibile modificarle; inoltre nel tab “input controls” è possibile impostare una frequenza da sottrarre per avere a video la frequenza di RX sul satellite (senza il display 10 Ghz); scriviamo quindi -250,081070 Mhz nella casella di testo identificata da LNB LO:



*Correzione manuale del display della frequenza*

Ora non resta che prendere pratica con il software e verificare tutte le sue potenzialità; il sito internet di Gqrx è davvero prodigo di informazioni (<https://gqrx.dk/>).

## Conclusioni



*La parabola completa, con doppio illuminatore e, dietro, la scatola stagna che contiene up converter e amplificatore*

Questo mio scritto non può (ma non voleva e non poteva neanche esserlo) un trattato completo ed esaustivo su come ricevere Oscar 100 né, tanto meno, su come gestire ed usare Linux; ho però voluto condividere alcune soluzioni che ho adottato per ricevere questo satellite usando il sistema operativo che uso ed i software che mette a disposizione, abbinati con hardware economico; i comandi sono quelli che mi hanno portato ad avere un gqrx funzionante e sintonizzato sul satellite.

Mi auguro che fare un pochino di ascolto sul trasponder spinga altri colleghi italiani a fare prove ed esperimenti anche in trasmissione, attività lievemente più dispendiosa ma certamente più appagante.

Buon Oscar 100 a tutti!

### **Annotazioni:**

*Documento tratto dal sito internet <http://www.ik7xja.it/>, sito internet della stazione radioamatoriale salentina.*