

# Operare sui satelliti con FT817 e SDR-RTL

## I satelliti: due parole

Non c'è nulla di nuovo in questo articolo, ma visto che ho faticato a distinguere consigli obsoleti e qualche volta proprio errati, ho pensato di fare cosa gradita a chi volesse ripetere l'esperienza soffrendo meno.

Satelliti oggi ce ne sono veramente tanti – troppi? – anche nel nostro mondo amatoriale, e non è sempre semplice capire quali ancora funzionano: una pagina preziosa è <https://www.amsat.org/status/> in cui si registrano le osservazioni di ogni radioamatore che ha sentito (o non sentito) ogni satellite AMSAT e su cui si può vedere quali sono affidabili e quali in fine vita.

I satelliti amatoriali sono ripetitori nella quasi totalità lineari e quindi adatti al traffico SSB e CW, alcuni più recenti ripetono canali FM, il che semplifica non poco la vita anche se tolgono un po' di fascino... Il funzionamento più diffuso è nelle bande 70 cm in ingresso (uplink) e 2 m in uscita (downlink), più raramente l'inverso o altro ancora. Un discorso a parte per il satellite geostazionario AO100, meno sfidante operativamente e più interessante tecnicamente, e per cui questo articolo non ha senso.

Un satellite in orbita più o meno alta e più o meno inclinata sull'eclittica *deve essere inseguito*. Prima ancora che con le antenne (io infatti sto usando una turnstile) per compensare l'effetto doppler che shifta la frequenza durante il passaggio, e non poco: +/-3khz e rotti sui 144 e ovviamente il triplo in 70cm alle quote impiegate (dalla quota dipende la velocità e quindi lo scostamento: più bassa la quota, più veloce il passaggio e più alto l'effetto doppler).

## Requisiti

- Windows; capisco le lamentele, ma su Linux potete usarlo come suggerimento, non altro. Se invece usate Apple, non era meglio risparmiare un \$ per la radio?
- Windows 10; dovrebbe funzionare tutto anche più facilmente su Windows 7, ma non ci ho provato; comunque nulla è UWP, quindi non dovrebbero esserci problemi.
- Firma dei driver disabilitata e qui cominciano i problemi (non su Windows 7)

## Firma del driver

Windows 10 ha introdotto l'obbligo assoluto che un driver venga firmato da Microsoft per proteggere i sistemi da potenziali malware che si installassero fingendosi driver, sulla falsa riga di Apple, che controlla tutto. Disabilitare il signing richiede un poco più di prudenza in futuro. Si potrebbe pensare di abilitare e disabilitare usando i programmi, o disabilitare temporaneamente, ma ogni volta si deve riavviare, quindi... poco pratico.

Le istruzioni sono qui: <https://windowsreport.com/driver-signature-enforcement-windows-10/#:~:text=Press%20and%20hold%20the%20Shift,select%20Disable%20driver%20signature%20enforcement>. Seguite le istruzioni per disabilitare permanentemente.

Oddio, se leggete su carta è dura la vita. In due parole: aprite una console amministrazione – con click destro su Start, → prompt dei comandi (amministratore) – e lanciate “*bcdedit.exe /set nointegritychecks on*”, rebootsrappate ed è fatto.

**Purtroppo** diversi produttori di PC proteggono i loro prodotti dai casini di noi esperti hobbisti mediante l'avvio protetto nel BIOS e il comando precedente vi restituisce un errore. Come disabilitarlo dipende dal produttore, il mio HP mi ha costretto ad aprire il BIOS (accensione con F10) e cercare tra le impostazioni di sicurezza l'avvio protetto (che chiamano in altro modo...)

Questa è una operazione preliminare, ma già siete a metà dell'opera.

## Software

Questi software che servono: aggiungo il link, ma fate più in fretta a googlarli (orribile neologismo!) che a ricopiare i link – se leggete su carta:

- Orbitron <http://www.stoff.pl/downloads.php>
- SDR# <https://airspy.com/download/>
- WiSP DDE <http://www.amsat.org/amsat/ftp/software/win32/radio-control/wispdde40.zip>
- plugin CalicoCAT <https://tagloomis.com/sdr-plugins/>
- com0com <https://sourceforge.net/projects/signed-drivers/files/com0com/v3.0/>

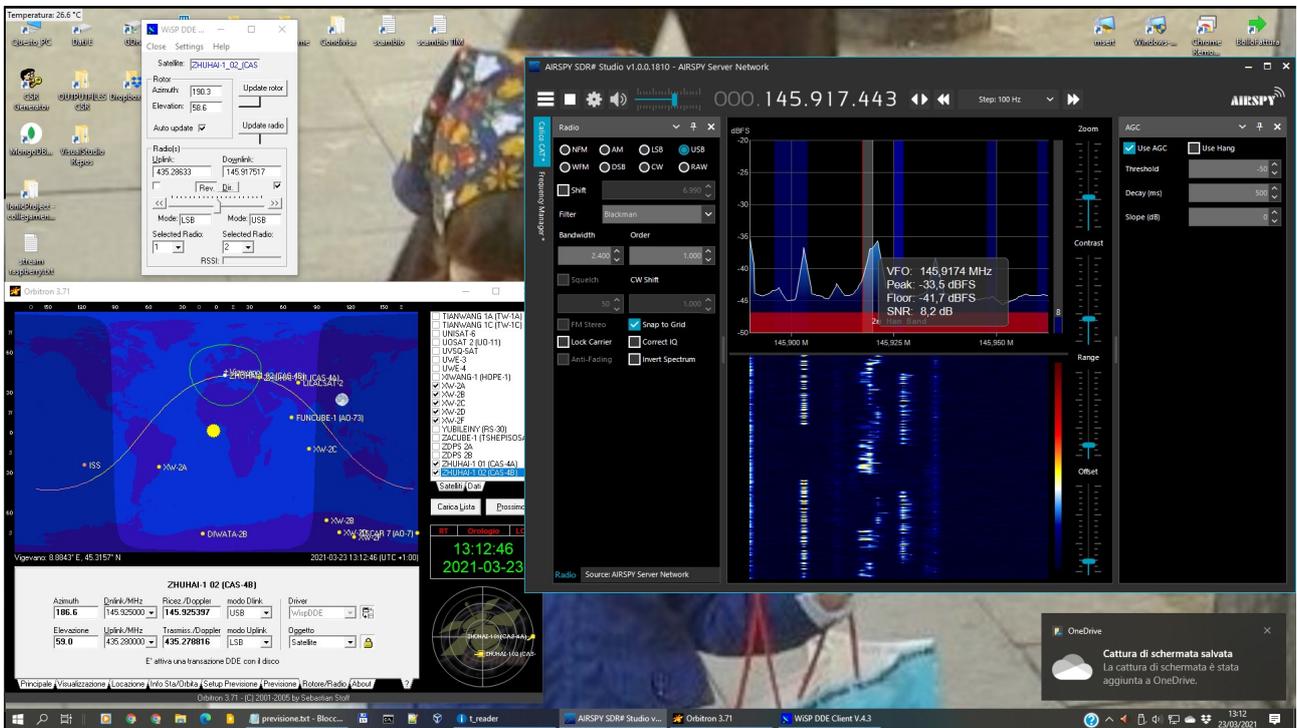


Figura 1: i software coinvolti

## Orbitron

E' un potente e divertente SW di previsione della posizione dei satelliti. Specificando la vostra posizione tramite locatore o latitudine e longitudine o direttamente con il locatore, e l'altezza s.l.m. calcolerà data, ora, direzione e azimuth del satellite selezionato. Odio inventare l'acqua calda e il programma ha un'ottima interfaccia interpretabile: guardate questa introduzione e via: <http://www.radioamatoripeligni.it/i6ibe/orbitron/orbitron.htm>, giocateci un po'

## WiSP DDE

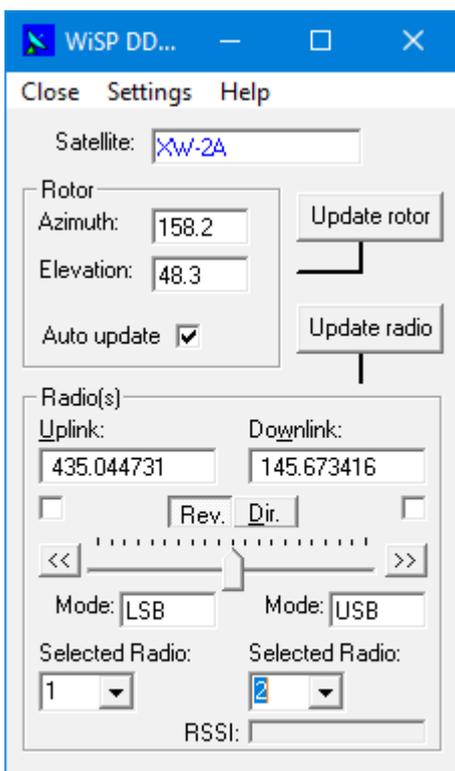


Fig. 2 WiSP window

È l'interfaccia tra Orbitron e la radio, può pilotare due apparecchiature, una in ricezione e una in trasmissione mantenendole allineate con priorità se scegliere la trasmissione (CQ CQ...) o la ricezione (cercate voi chi chiama) e compensando il doppler. Ovvio che se volete usare due radio, qui vi fermate...

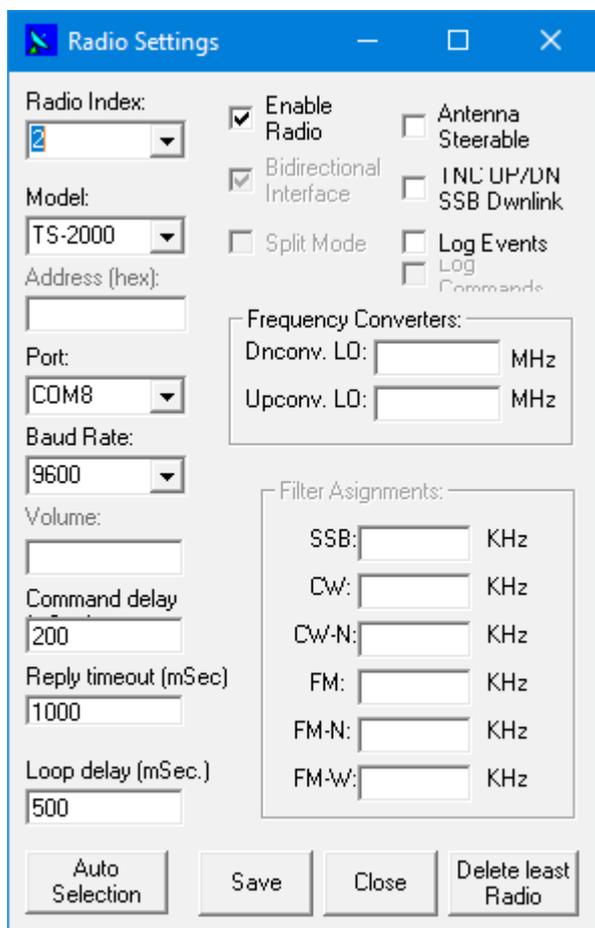
WiSP DDE: questa finestra si apre anche automaticamente quando su Orbitron abilitate l'inseguimento (dopo aver scelto WiSP come interfaccia DDE).

Il satellite viene compilato da Orbitron con il nome dell'oggetto inseguito, la radio "selected" in ricezione e trasmissione nel mio caso sono un FT818nd e SDR# tramite il plugin Calico CAT.

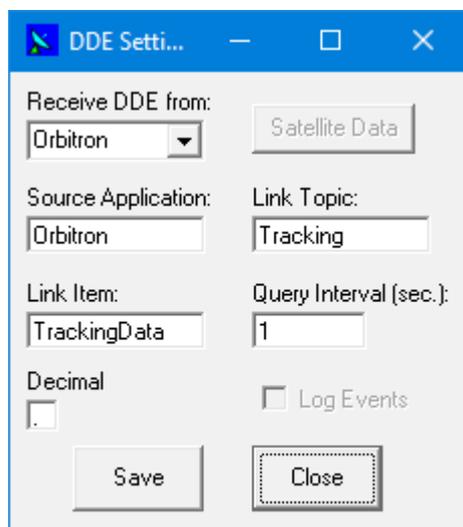
## Setting

Le radio che possono essere memorizzate sono tre, per ognuna si specifica il modello, la porta di comunicazione e se necessari altri parametri funzionali. L'uso del SW SDR# con il plugin simula un TS2000.

Per quanto riguarda l'intervallo di query un secondo per me è sufficiente, non stressate il PC oltre il lecito in quanto deve già decodificare il segnale SDR e supportare Orbitron.

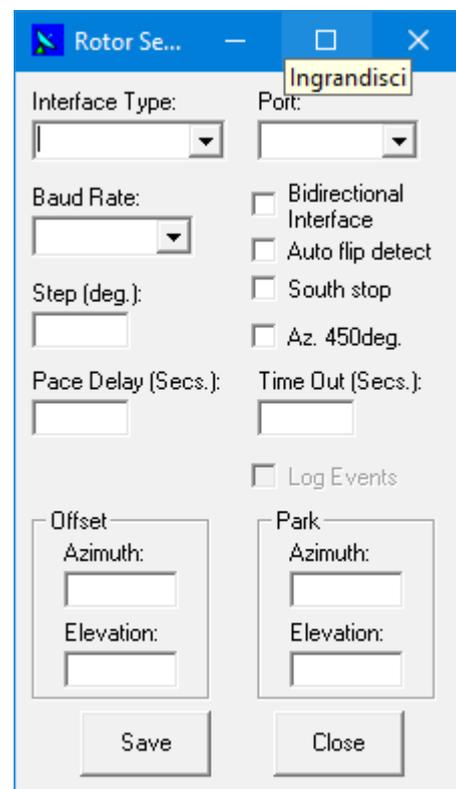


**Fig. 3: settig delle radio**



**Fig. 4 setting DDE sorgente**

Il setting della sorgente è lapalissiano: da Orbitron preleviamo i trakingdata (fig.4)



**Fig 5: setting rotore**

Per quanto riguarda il rotore dovete arrangiarvi fin quando non monterò (se mai monterò) un paio di direttive accoppiate (fig. 5)

## SDR#

Beh, questo non è elementare, ma è documentato sulla rete a iosa, insieme alle sue difficoltà di funzionamento. In realtà io uso un raspberry PI3 come server

piazzandolo all'esterno vicino alla antenna (poca discesa) e raggiungendolo in WiFi perché fa parte di un previsto progetto per AO100, e i setup su Linux sono molto più dominabili rispetto a Windows che tenta di fare tutto lui, ma si sopravvaluta.

Comunque le istruzioni da seguire sono sul sito di AirSpy, la quickstart è chiara: <https://airspy.com/quickstart/>. Segnatevi il nome della directory di installazione!

## Plugin

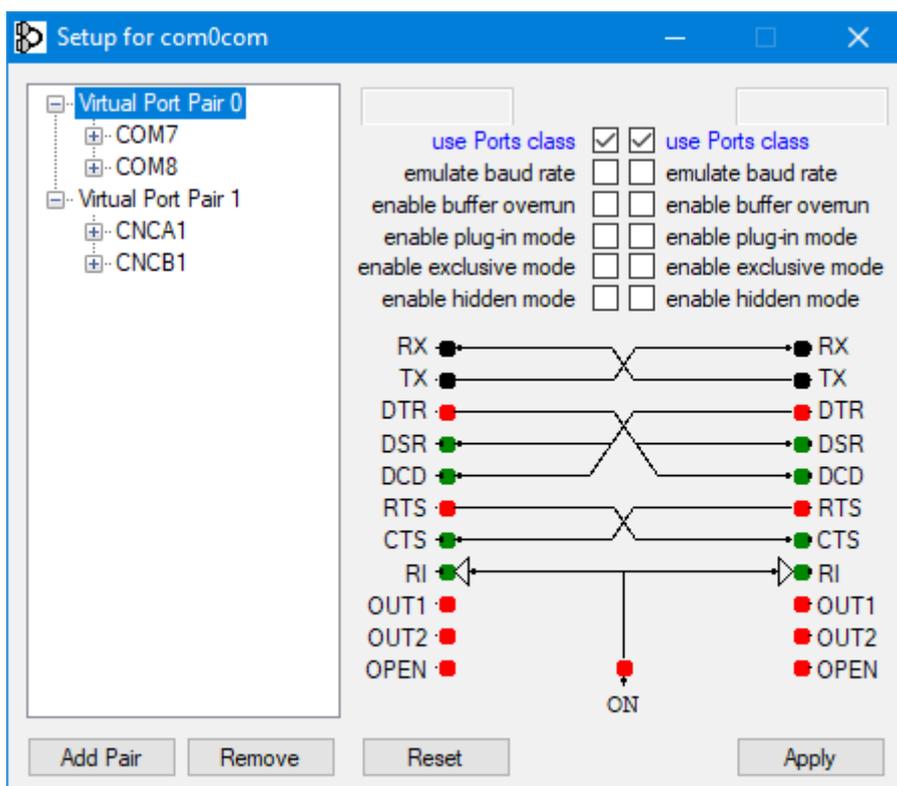
Il plugin CalicoCAT scaricato è uno zip con tre file: andate nella directory bin di SDR#; la avevate copiata vero? Se no cercatela... AirSpy non installa nella classica directory Windows per non litigare coi permessi

Qui copiate la dll; e aprite il file plugin.XML (con Notepad) e aggiungete la riga contenuta in MagicLine.txt. Queste non sono altro che le istruzioni contenute nel terzo file, readme.txt.

## Com0com

Non è proprio elementare, a parte che avete già dovuto eliminare la richiesta di firma dei driver, che è più complesso.

Comcom0 installa un driver di canale virtuale per porte com, accoppiandone due come se esistessero e fossero connesse tra loro in null modem. Detto più semplicemente una porta parla con l'altra anche se non esistono. Metteremo quindi Orbitron a parlare con una e SDR# (attraverso il plugin) con l'altra.



Aprite il setup di comcom0 (chiederà un permesso...) (fig. 6):

Probabilmente vedrete scritto COM# su entrambe le porte e non una porta specifica. Eliminate pure la virtual port con CNCA1 e CNCB1, servono per virtualizzare porte su Ethernet.

Fig. 6

Togliete la spunta su “use port class” su entrambi i lati e rinominate con porte a scelta purché libere sul vostro PC

Dovete ottenere qualcosa di simile alla fig 7, che in figura connette i miei canali virtuali COM7 e COM8.

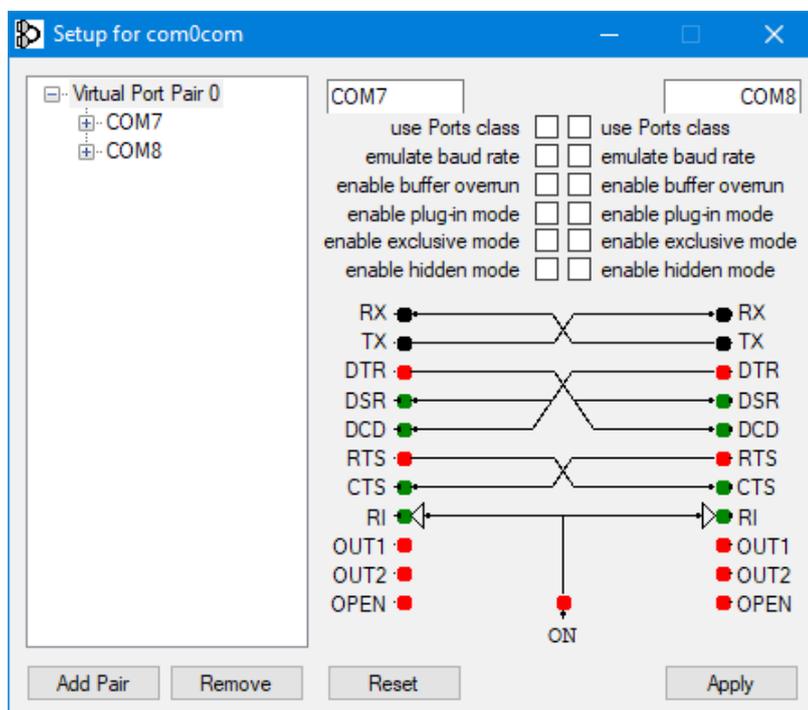


Fig.7

## Setup Finale

Beh, dovrebbe funzionare...

Avviate Orbitron e SDR#. Su SDR# cercate il plugin Calico (numero ed ordine dei plugin cambia a seconda della installazione, dovrete vederne molto pochi in confronto alla mia installazione di test (fig.8)

Cliccando il plugin Calico vedete il suo setup (fig. 9)

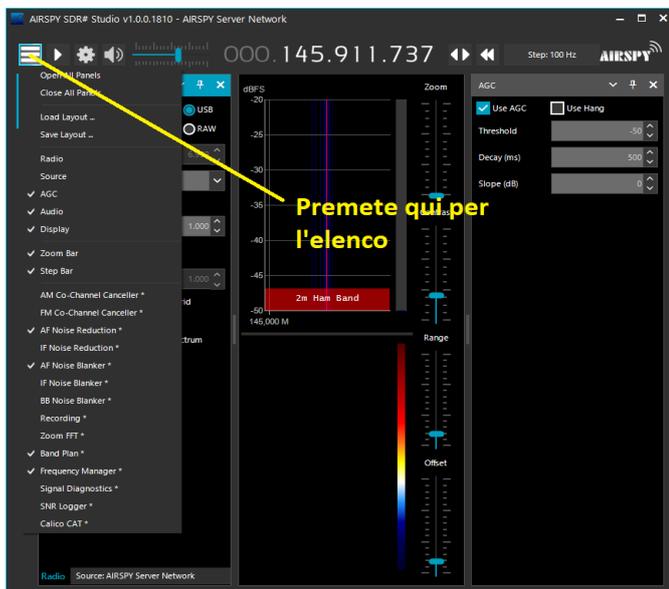


Fig. 8: elenco plugin di SDR#

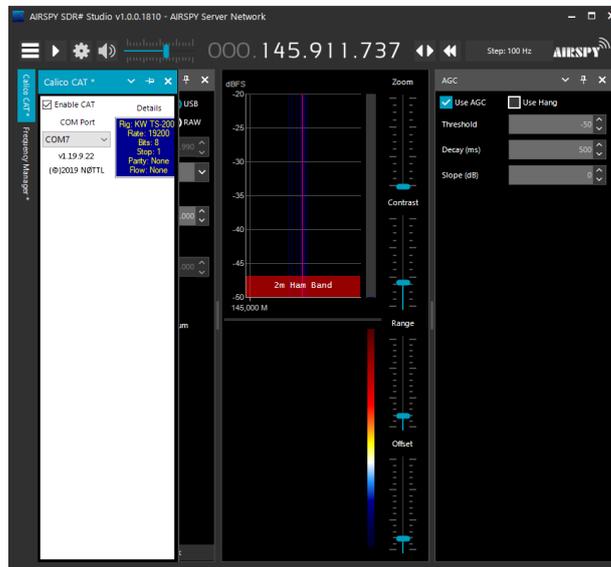


Fig. 9 setup del plugin

Abilitando il CAT scegliete una delle COM virtuali che avete messo in com0com.

Poi aprite WiSP DDE (è necessario solo facendo il setup) e premete setting per fare setup delle due radio: una sarà il vostro apparato sulla sua COM di controllo, l'altra il TS2000 con la seconda COM di com0com (fig. 10).

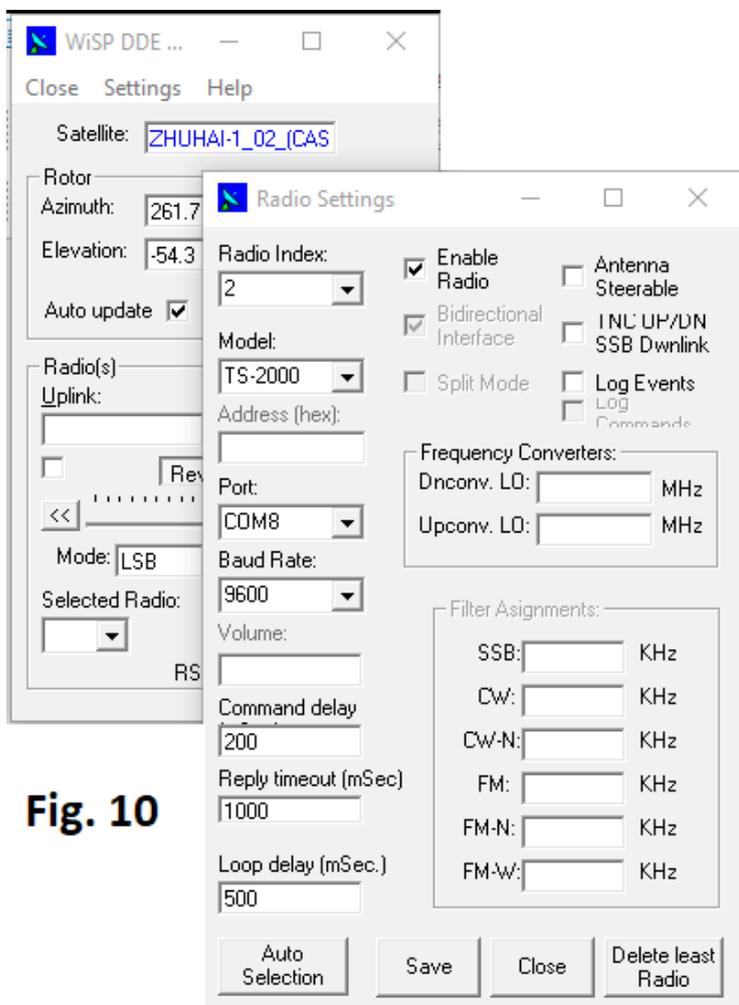


Fig. 10

Quindi sempre in setting indicate Orbitron come DDE link (fig. 11) e chiudete poi WiSP.

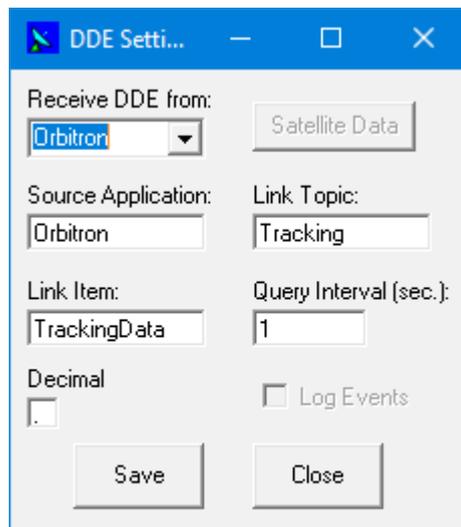


fig. 11

Ora aprite Orbitron e SDR# e premete l'icona DDE (fig 13)



Fig.13

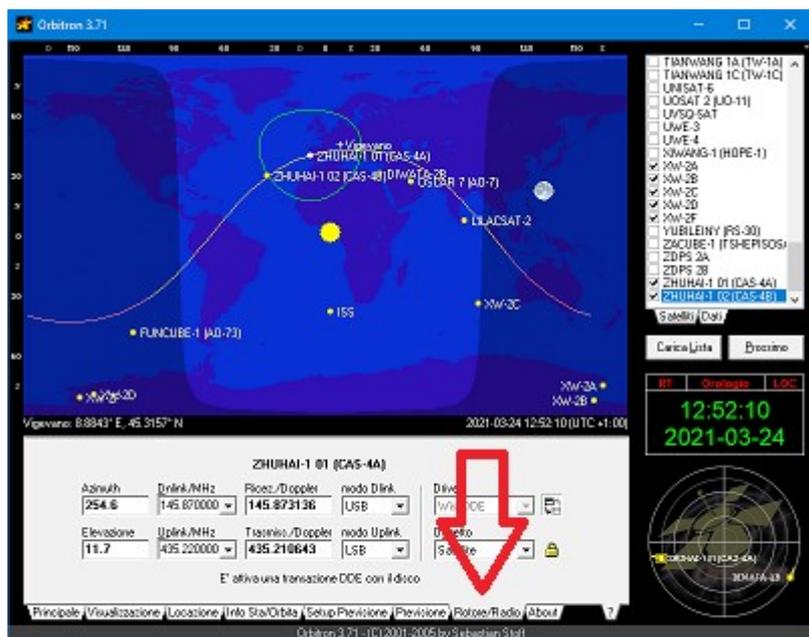


fig. 12 scheda rotori e radio

Se necessario (non è selezionato nessun satellite nell'elenco caricato) selezionate un qualunque satellite.

Si aprirà WiSP (in alto a sinistra) e riporterà il nome del satellite attivo in Orbitron.

I dati di comando delle radio (frequenza, modo) saranno in bianco (fig.13)

Ora potete indicare in downlink il falso TS200 che è poi SDR# e in uplink l'apparato trasmittente.

Decidete se domina il trasmettitore (CQ CQ) o il ricevitore (andate voi a caccia) e l'altro apparato inseguirà.

Dovete indicare se il trasponder è diretto o invertente perché la compensazione sarà diretta o inversa; l'impostazione di frequenza di partenza dovete compilarla – sfortunatamente per ogni satellite – in Orbitron, ma sarà memorizzata; o se preferite impostatela a mano ogni volta sul dominante tra up e downlink (trasmettitore o SDR#).

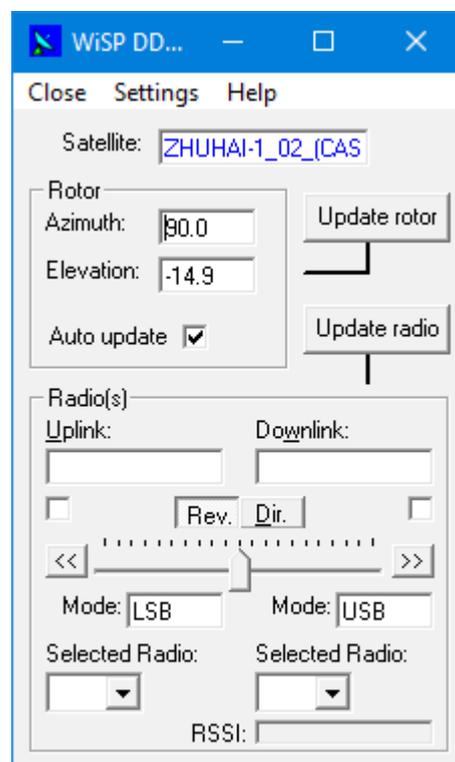


fig. 13