

PYTTIAPRS:

UN SOFTWARE PER APRS, ESSENZIALE LEGGERO, PENSATO PER LA RADIO (Lorenzo IU1BOT)

```
CALL IU1BOT TOCALL CQ PATH WIDE2-2 LAT 44.3102 LON 9.3359 SYM /- ACK OFF
m:msg p:pos c:cfg x:clr msgs h:clr heard d:raw t:rep raw r:repeat 1:QSL? 73 2:QSL! 73 a:ack on/off q:quit
Received packets:
12:22:09 IU1BOT > CQ WIDE2-2
: !4418.61N/00920.15E-https://github.com/vash909/PyttiAPRS

12:24:18 IR1ZXE-11 > APMI04 WIDE3-3
: @311124z4424.56N/00902.04E#PLXDigi 1K2 M.Fasce 834mslm U=12.7V.

12:24:28 IR1ZXE-11 > APMI04 WIDE1-1
: @311124z4424.56N/00902.04E#PLXDigi 1.1 I-Gate M.Fasce

12:24:36 IR1ZXE-11 > APMI04 WIDE1-1
: }IQ1ZW-10>APLHI0,TCPIP,IR1ZXE-11*:!4424.77N!00858.65E#LoRa APRS iGate Gruppo Radio Liguria

12:24:52 IR1ZXE-11 > APMI04 WIDE1-1
: }IR1ZYD>APRS,TCPIP,IR1ZXE-11*:;ER-IR1ZYD*11111z4424.53N!00902.03E#0430.737MHz T110 R22k +5 MHz -Gr
```

Nel mondo dell' APRS non mancano certo i software: mappe, finestre, livelli, filtri, configurazioni complesse. Tutto molto potente, ma non sempre pratico, soprattutto quando l'obiettivo è fare radio, magari durante un'attività portatile, una sessione satellitare o semplicemente da un terminale remoto.

Sperimentando ampiamente sui satelliti APRS (SONATE-2, ISS PKT, ecc.) mi sono reso conto della mancanza di un client leggero e che facesse rapidamente quello di cui avevo bisogno, o meglio, la mancanza di un client multipiattaforma (sono un assiduo utente linux). Senza soffermarsi a parlare di cosa è già presente su altre piattaforme mi sono dato da fare per avere “un qualcosa” che potesse girare quasi ovunque consumando meno risorse possibili ed ecco qua PyttiAPRS!

Da questa esigenza nasce PyttiAPRS, un piccolo progetto open-source sviluppato in Python che si propone un obiettivo chiaro:

offrire un client APRS semplice, immediato e leggero, concentrato sulle funzioni fondamentali, senza fronzoli.

Un'idea “vecchia scuola”, ma attuale

PyttiAPRS riprende una filosofia che molti colleghi apprezzano:

pochi strumenti, ben fatti, che fanno esattamente quello che serve.

L'interfaccia è completamente testuale (TUI – Text User Interface), simile a quella dei vecchi programmi da terminale, ma con un tocco moderno: finestre ben organizzate, colori, supporto al mouse e comandi rapidi da tastiera. Nessuna mappa, nessuna grafica pesante: solo ciò che serve per comunicare.

Questo approccio lo rende ideale in molte situazioni:

- collegamenti APRS via satellite (ISS, digipeater orbitali)
- attività portatile o QRP
- utilizzo su Raspberry Pi o mini-PC
- connessioni remote via SSH
- sperimentazione e didattica

PyttiAPRS permette di:

- inviare e ricevere messaggi APRS tra radioamatori

- trasmettere beacon di posizione con simbolo e commento configurabili
- visualizzare un log cronologico dei pacchetti ricevuti e inviati
- mantenere una lista delle stazioni ascoltate (“Heard”)
- rispondere rapidamente con messaggi predefiniti.

Tutto questo avviene in un’unica schermata, chiara e leggibile anche su terminali piccoli.

Un aspetto particolarmente curato è la gestione dei messaggi APRS con conferma (ACK): quando attivata, PyttiAPRS aggiunge automaticamente l’identificativo di messaggio e riconosce le conferme ricevute. Allo stesso tempo, questa funzione può essere disattivata con un tasto, caratteristica molto utile nei collegamenti satellitari, dove non sempre gli ACK tornano a terra.

Pensato per l’uso radio, non solo per il protocollo

Uno dei punti di forza di PyttiAPRS è che non nasce come esercizio di programmazione, ma come strumento operativo.

Ogni scelta è guidata dall’esperienza pratica in radio, in particolare durante le attività sui transponder APRS satellitari dove il tempo è poco e i segnali non sempre forti.

Il programma è stato pensato per avere a portata di tastiera i comandi più comuni che diventano accessibili con un solo tasto, selezionare con il mouse immediatamente il corrispondente al quale rispondere e avere una configurazione che viene salvata ad ogni chiusura del programma.

Il programma non pretende di sostituire i grandi client APRS grafici, ma si affianca a essi come strumento rapido e affidabile, perfetto quando “meno è meglio”.

Sotto il cofano, ma senza spaventare

Dal punto di vista tecnico, PyttiAPRS dialoga con un TNC compatibile KISS (ad esempio Direwolf) e gestisce direttamente i pacchetti APRS.

Per l’utente, però, tutto questo è trasparente: basta avviare il TNC, lanciare lo script e iniziare a operare.

Il fatto che sia composto da un solo file Python, senza librerie esterne, lo rende anche un ottimo punto di partenza per chi vuole capire come funziona “davvero” APRS oppure studiare la struttura dei pacchetti ed il funzionamento ad un livello più basso. Ovviamente essendo un programma completamente aperto alla comunità è possibile personalizzarlo come si vuole e ogni contributo è più che ben accetto.

Chiunque può scaricarlo, usarlo, studiarlo e contribuire.

Non è un prodotto “chiuso”, ma un progetto vivo, che cresce grazie ai suggerimenti e all’esperienza condivisa dei radioamatori.

Concludendo...

In un’epoca in cui spesso il software radio diventa sempre più complesso, PyttiAPRS va in controtendenza e dimostra che semplicità ed efficacia possono convivere.

È uno strumento che non fa rumore, non distrae e non pretende attenzione:

sta lì, pronto, e fa quello che deve fare. Proprio come una buona radio.

Per chi ama l’APRS, la sperimentazione e il “fare radio” nel senso più autentico del termine, PyttiAPRS è sicuramente un progetto da provare.

Si ringraziano: IU6HMO Matteo e IU2RRT Pietro per le innumerevoli prove via ISS

DL7NDR Daniel per i preziosi bug report

IU8DKG Alex per il beta-testing

PyttiAPRS è rilasciato come software libero e il codice sorgente è disponibile su GitHub:

<https://github.com/vash909/PyttiAPRS>

```

DF2ET audio level = 36(7/3)  —||||_
[0.5] DF2ET>APK003,ARISS:DF2ET :gnm[9<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "9", from "DF2ET" to "DF2ET", Kenwood TH-D72
gnm

DF2ET audio level = 31(6/3)  —|||_
[0.4] DF2ET>APK003,ARISS:DF2ET :ngnknlnngn@ngn@ndw[10<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "10", from "DF2ET" to "DF2ET", Kenwood TH-D72
ngnknlnngn@ngn@ndw

[0L] DF2ET>APZ001,J51YPT:DG3NGN :QSL? 73{002
[0L] DF2ET>APZ001,J51YPT:DG3NGN :QSL? 73{002

DF2ET-4 audio level = 33(6/3)  —||||_
[0.4] DF2ET-4>APK003,ARISS:DF2ET :gnm[9<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "9", from "DF2ET-4" to "DF2ET", Kenwood TH-D72
gnm

DF2ET-4 audio level = 32(7/3)  —||||_
[0.4] DF2ET-4>APK003,ARISS:DF2ET :ngnknlnngn@ngn@ndw[10<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "10", from "DF2ET-4" to "DF2ET", Kenwood TH-D72
ngnknlnngn@ngn@ndw

DF2ET-4 audio level = 24(7/3)  —||||_
[0.5] DF2ET-4>APK003,ARISS:DF2ET :gnm@wjngpn[11<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "11", from "DF2ET-4" to "DF2ET", Kenwood TH-D72
gnm@wjngpn

[0L] DF2ET>APZ001,J51YPT:sfsdfds
[0L] DF2ET>APZ001,J51YPT:15129.64N/00714.43E>Florian N18 Bochum
[0L] DF2ET>APZ001,J51YPT:15129.64N/00714.43E>Florian N18 Bochum

DF2ET-4 audio level = 33(7/3)  —||||_
[0.5] DF2ET-4>APK003,ARISS:DF2ET :ngnknlnngn@ngn@ndw[10<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "10", from "DF2ET-4" to "DF2ET", Kenwood TH-D72
ngnknlnngn@ngn@ndw

DF2ET-4 audio level = 33(7/3)  —||||_
[0.4] DF2ET-4>APK003,ARISS:DF2ET :gnm@wjngpn[11<0x0d>
The APRS protocol specification says nothing about a possible carriage return after the
message id. Adding CR might prevent proper interoperability with with other applications.
APRS Message, number "11", from "DF2ET-4" to "DF2ET", Kenwood TH-D72
gnm@wjngpn

[0L] DF2ET>APZ001,J51YPT:DG3NGN :QSL? 73{003
[0L] DF2ET>APZ001,J51YPT:CQ :3031{004

```

```

CALL DF2ET TOCALL APZ001 PATH J51YPT LAT 51.4940 LON 7.2404 SYM /> ACK ON
m:msg p:pos c:cfg x:clr msg: h:clr heard d:raw t:rep raw r:repeat i:QSL? 73
Received packets:
17:55:09 DF2ET > APK003 ARISS
: DF2ET :gnm[9
17:55:11 DF2ET > APK003 ARISS
: DF2ET :ngnknlnngn@ngn@ndw[10
17:55:25 DF2ET > DG3NGN J51YPT
: DG3NGN :QSL? 73{002
17:55:29 DF2ET > DG3NGN J51YPT
: DG3NGN :QSL? 73{002
17:56:09 DF2ET -4 > APK003 ARISS
: DF2ET :gnm[9
17:56:11 DF2ET -4 > APK003 ARISS
: DF2ET :ngnknlnngn@ngn@ndw[10
17:56:11 DF2ET -4 > APK003 ARISS
: DF2ET :gnm@wjngpn[11
17:56:22 DF2ET > APZ001 J51YPT
: sfsdfds
17:56:38 DF2ET > APZ001 J51YPT
: 15129.64N/00714.43E>Florian N18 Bochum
17:56:41 DF2ET > APZ001 J51YPT
: 15129.64N/00714.43E>Florian N18 Bochum
17:57:12 DF2ET -4 > APK003 ARISS
: DF2ET :ngnknlnngn@ngn@ndw[10
17:57:14 DF2ET -4 > APK003 ARISS
: DF2ET :gnm@wjngpn[11
17:57:18 DF2ET > DG3NGN J51YPT
: DG3NGN :QSL? 73{003
17:57:23 DF2ET > CQ J51YPT
: CQ :3031{004

```

transiti attraverso ISS e BOTAN in test da DF2ET

Lorenzo IU1BOT - iu1bot@xzgroup.net

AmateurRadioSociety

HOME ASSOCIAZIONE SERVIZI CIRCOLI ISCRIVITI LA RADIO DOWNLOAD NEWS CONTATTI

IQ2GSF
il futuro della radio
adesso

"LA RADIO" Primavera 2026 - A.R.S. - <https://www.arsitalia.it>