

Ricetta per un UV-K5 quasi perfetto.

Per utenti esperti e smanettoni, scritto come mio promemoria, ma che può essere utile anche a qualcun altro.

Ingrediente principale: Firmware Piotr ultima versione (attualmente alla 78, ma sempre in continuo aggiornamento)

Qui: https://github.com/piotr022/UV_K5_playground/releases

Io uso il firmware con la RSSI bar ovvero questa: uv_k5_26_rssi_sbar_encoded_v78.bin
Scaricatelo, ma aspettate a caricarlo sulla radio (con il programma di aggiornamento fornito da Quansheng, qui relative istruzioni e link: <https://www.youtube.com/watch?v=ctIDHpQKjD4&t=24s>)

Aspettate, perchè al già ottimo firmware è possibile aggiungere qualche altra caratteristica, disponibile qui: <https://whosmatt.github.io/uvmod/>

[fbclid=IwAR0YxfXkJNuD4vzKJ2TbMYAjry3PK-Eu2ubKs-lvNIz6Tu8FiAqhzRqpPG0](https://whosmatt.github.io/uvmod/?fbclid=IwAR0YxfXkJNuD4vzKJ2TbMYAjry3PK-Eu2ubKs-lvNIz6Tu8FiAqhzRqpPG0)

Selezionate le opzioni di vostro interesse e (inutile aggiungere quelle già presenti nel firmware di Piotr), sulla destra deselezionate "Using stock v26 firmware", cliccate su "Browse" e selezionate il firmware di Piotr scaricato precedentemente.

Cliccate su "Patch firmware" e verrà generato il firmware personalizzato che a questo punto potete scaricare sul pc e poi caricare nella radio con la procedura descritta nel video linkato sopra.

Attenzione a non sforare il massimo consentito di dimensioni del file, in tal caso il compilatore vi avvertirà che è troppo grande per la memoria della radio.

In particolare il firmware di Piotr più Bootscreen personalizzato non sono compatibili, mentre lo sono tutte le altre opzioni a 0B.

Attenzione: dopo aver caricato il firmware ricordatevi di cliccare su "disconnect", altrimenti la seriale resta impegnata e non vi permette di fare le operazioni successive.

Già qui siamo a buon punto, ma ci sono ancora alcune cosette da sistemare.

Per esempio lo squelch che in queste radio è particolarmente aggressivo ovvero interviene a livelli di segnale che sarebbero ancora perfettamente discernibili facendo perdere parti di eventuali comunicazioni con segnali bassi, ma ancora perfettamente comprensibili.

Qui vengono in aiuto le "Python utils" che trovate qui:

https://github.com/amnemonic/Quansheng_UV-K5_Firmware/tree/main/python-utils

Scaricate i seguenti files:

squelch_table_read.py

squelch_table_write.py

libuvk5.py

batt_calibrator.py (questo ci serve dopo)

Metteteli in una cartella facilmente raggiungibile con il prompt di comandi, per esempio direttamente sotto la radice del disco nella cartella "bidone" ovvero c:\bidone

Adesso è necessario scaricare e installare Python dal sito ufficiale

<https://www.python.org/downloads/>

Installatelo in modo personalizzato con tutte le opzioni attivate.

A questo punto serve aprire il prompt dei comandi (tasto windows poi CMD) e digitare il comando "pip install pyserial" (senza le virgolette) quindi aspettare che tutto venga caricato.

Collegate la radio e accendetela normalmente (senza premere il PTT) e sempre dal

prompt dei comandi portatevi nella cartella creata prima c:\bidone, con il comando "cd c:\bidone" quindi digitate il comando: "sqelch_table_read COMXX" (al posto di XX mettete il numero della COM a cui è collegata la radio e sempre senza virgolette come per i comandi successivi).

Se state usando un cavo USB, troverete la COM a cui è collegata la radio tramite "gestione dispositivi" di Windows, ma qui siamo all'ABC e non mi dilungo troppo.

Se tutto funziona, vi apparirà una serie di dati relativi ai parametri dell sqelch memorizzati nella radio.

A noi interessano i numeri relativi a "uhf_sqelch_open_rssi", "uhf_sqelch_close_rssi", "vhf_sqelch_open_rssi", e "vhf_sqelch_close_rssi" che nella lista generata a video sono quelli indicati in dbm.

Gli altri numeri meglio non toccarli.

Per esempio nella mia radio lo sqelch settato a 1 prima interveniva a partire da segnali (FM modulati a 3Khz di deviazione) inferiori a -120dbm circa, tagliando fuori i segnali di poco più deboli, ma ancora perfettamente comprensibili.

Adesso lo sqelch interviene a circa -130dbm rendendo l'ascolto più piacevole e meno "spezzettato" in caso di segnali fluttuanti.

Come si fa?

Va editato il file "sqelch_table_write.py" con blocco note (per sicurezza fate un copia dell'originale).

Riporto qui per esempio la riga da editare per quanto riguarda l'apertura dello squech in Uhf:

```
uhf_sqelch_open_rssi =  
bytearray([0x0a,0x34,0x3a,0x40,0x46,0x4c,0x52,0x58,0x5e,0x64,0xff,0xff,0xff,0xff,0xff  
)
```

Il primo numero esadecimale dopo la parentesi graffa corrisponde allo sqelch 0 e non serve toccarlo, i successivi, separati da virgole, sono rispettivamente i livelli di sqelch 1, 2, 3, ecc. fino a 9. I numeri successivi tutti uguali a 0xff non vanno toccati.

Come si calcola il valore da inserire?

Si somma il livello di dbm a cui si desidera che lo sqelch intervenga al numero 160 e il risultato si moltiplica per due. Il numero decimale ricavato si converte in esadecimale e si scrive dopo 0x nel punto corrispondente al livello di sqelch che vogliamo modificare.

Esempio:

Voglio che lo sqelch a livello 1 si apra in UHF con segnali a partire da -131 dbm

Sommo -131 a 160 -> $-131+160=29$

Moltiplico per due-> $29 \times 2=58$

58 decimale corrisponde in esadecimale a 3A che scrivo nella riga sopra, dove 0x34 diventa 0x3A

Stessa procedura per gli altri valori tenendo presente che i valori delle rispettive righe "close" devono essere leggermente superiori ai valori "open" per dare un po' di cosiddetta "isteresi" al sistema, in modo che, con segnali al limite, non apra e chiuda di continuo.

Fatto tutto, salvate il file.

A questo punto basta collegare la radio, accenderla (senza premere PTT) e dare il comando "sqelch_table_write COMXX" (al posto di XX mettete il numero della COM a cui è collegata la radio come detto sopra).

Per verificare che i livelli siano effettivamente cambiati basta dare il comando "squelch_table_read COMXX" e scorrendo la lista i valori dovrebbero essere quelli nuovi.

Per iniziare mi limiterei a modificare il livello dello squelch 1 per vedere se la ricezione è meno ballerina o meglio ancora, avendo lo strumento adeguato, per misurare la differenza tra prima e dopo.

Con il Test-set ho visto che i valori restituiti a video sono abbastanza aderenti a quelli misurati, mentre quanto indicato sul display con il firmware di Piotr è circa 10 dbm superiore al segnale reale, ma per vedere la differenza tra prima e dopo va comunque bene.

Calibrazione indicazione batteria.

Se siete arrivati fin qui, calibrare la indicazione della tensione della batteria, sarà una passeggiata.

Misurate la tensione della batteria montata sulla radio sui terminali posteriori, con un voltmetro il più possibile affidabile.

Prendete nota del valore e collegate la radio al pc.

Verificate che la tensione sia sempre quella e che non abbia cambiato valore: alcuni cavetti di programmazione inducono una tensione nella radio che si riperquote su quella della batteria. Se così fosse, tenete in considerazione il primo il valore letto a radio scollegata.

Accendete la radio normalmente.

Dal prompt dei comandi spostatevi nella solita cartella "Bidone" di cui sopra e lanciate il seguente comando: "batt_calibrator.py COMXX calibrate" (come al solito al posto di XX ci va il numero della COM a cui è collegata la radio)

Inserite il valore di tensione precedentemente letto per esempio 7,98 senza preoccuparvi di separare le unità dai decimali con il punto o la virgola in quanto vanno bene entrambi.

Aspettate pochi attimi che finisca il processo e spegnete la radio.

Quando la riaccenderete, il valore letto sul display sarà molto simile a quello reale della batteria centesimo di volt più, centesimo di volt meno.

La buona notizia è che i valori di squelch e quelli della calibrazione della batteria restano memorizzati nella radio anche con gli eventuali successivi aggiornamenti di firmware e quindi non è necessario ripetere ogni volta la procedura.

Come sempre, più semplice a farsi che a dirsi.

Fine del pippone.

73 de IK5WWP Lorenzo.