

PREPPER DOCK



v.2024-11-29 ADVENT

Manuale d'uso

PrepperDock è l'applicazione computer che s'interfaccia con il firmware PrepperRadio.

Permette di leggere informazioni o portare modifiche specifiche che non è possibile fare dalla radio stessa.

1. Installazione

Windows

Non ha bisogno d'installazione, semplicemente fare doppio click sul file PrepperDock-Win.exe.

Linux (Debian e derivate)

Con il pulsante destro selezionare "PrepperDock-Linux-install.run" e scegliere l'opzione "Lanciare come Programma".

Si apre brevemente un terminale e si chiude da solo.

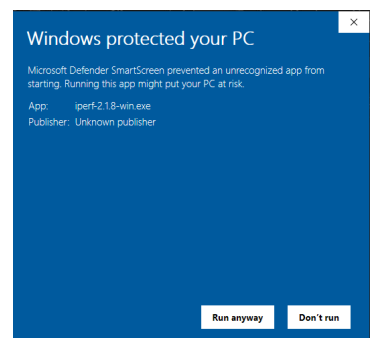
Premi il tasto windows e scrivi "prep" e ti appare PrepperDock.

Oppure cerca nel menu Radioamatori (HAM) e troverai PrepperDock.

Se mancassero alcune librerie nel sistema, il terminale ti chiederà la password e le installerà prima di chiudersi automaticamente.

2. Utilizzare PREPPER DOCK

⚠ Attenzione, all'avvio in Windows, "Defender" potrebbe avvertire che si tratta di un programma sconosciuto, indicate di voler proseguire comunque.



2.1 PrepperDock

È la finestra principale. Essa è composta da due schermate, Home e Advanced.

Home

Questa schermata è il centro di partenza, da qui potete accedere a tutto. PrepperRadio non è solo un firmware, ma un progetto completo destinato a creare un protocollo di emergenza tra persone locali.



In alto ci sono tre pulsanti che aprono le finestre delle funzioni principali.

Nella parte sottostante troverete diversi link che rimandano a tutta la documentazione PrepperRadio.

Advanced

Qui sono presenti funzioni destinate agli esperti o ancora in fase di sviluppo.

2.2 Q Connect

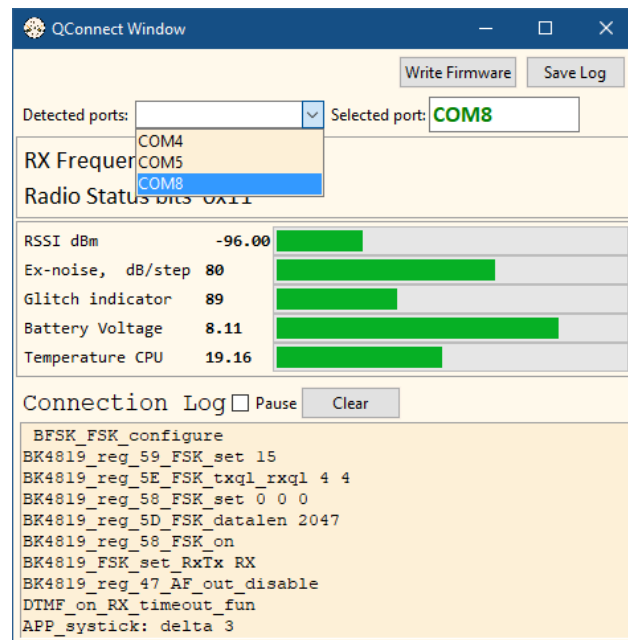
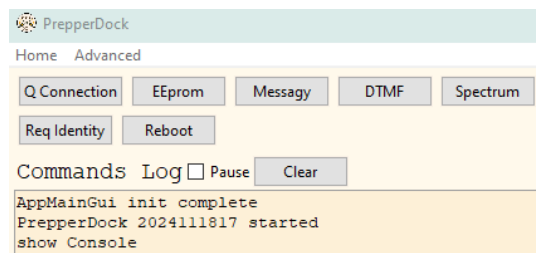
Questa finestra permette di caricare il nostro firmware. Ci permette anche di leggere tutta l'attività della radio in tempo reale e salvare il "log" tramite il primo tasto "Save Log".

Nella stringa "Detected Port", selezionare la porta COM dove sta collegato il cavetto USB.

Se la porta COM è quella giusta e se la radio è in modalità "FlashFirmware", PrepperRadio apre automaticamente la finestra per caricare il firmware all'avvio.

Il tasto "Write Firmware" permette di caricare il firmware PrepperRadio sulla vostra radio. Per maggior dettagli, riferirsi al manuale d'uso: [user-manual-PrepperRadio--.pdf](#).

Per Linux, il nome del dispositivo seriale è tipicamente `/dev/ttyUSB0`



2.3 EEPROM

In questa sezione è possibile modificare certi parametri memorizzati nella EEPROM della radio.

Anzi tutto bisogna leggerli tramite il tasto "Read". In seguito modificarli e per ultimo, scriverli tramite il tasto "Write".

Read: Legge il contenuto della EEPROM della radio, ovvero tutti i suoi parametri.

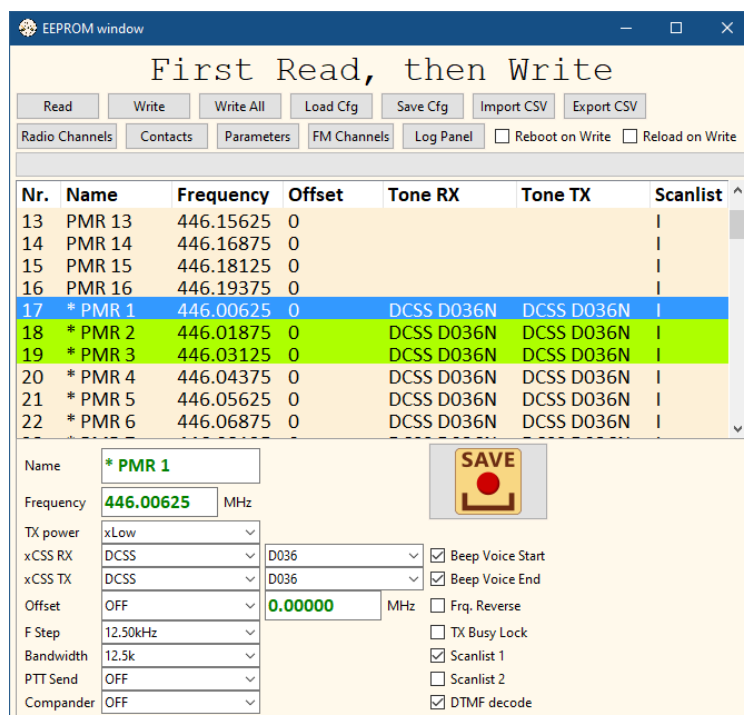
Write: Sovrascrive sulla radio solo i parametri che sono stati modificati.

Write All: Sovrascrive sulla radio tutta la EEPROM con i parametri presenti in PrepperDock. Utile se si vuole configurare diverse radio nello stesso modo.

Load Cfg: Carica un file di configurazione in PrepperDock.

Save Cfg: Salva la configurazione della radio in un file.

Import CSV: Carica sulla radio canali di memoria frequenze e i parametri principali.



Export CSV: Esporta la lista dei canali memorizzati e i parametri principali in un file CSV.

La schermata è suddivisa in diverse schede: Radio Channels, Parameters, FM Channels, Log Panel.

■ 2.3.1 Radio Channels

Qui si possono configurare tutti i canali memorizzati nella radio. È simile al menu Channel Config della radio, ma con parametri supplementari.

Ogni parametro modificato va confermato con il tasto "Save", questo salva i cambiamenti solo sul computer. Per caricarli in seguito sulla radio premere il tasto "Write".

La sezione dei canali riconosce le scorciatoie tastiera : Ctrl + C, X, V e la selezione multipla tramite il tasto Shift.

■ 2.3.2. Contacts

Rubrica del telefono. Possibilità di modificare o aggiungere i contatti.

Ogni parametro modificato va confermato con il tasto "Save", questo salva i cambiamenti solo sul computer. Per caricarli in seguito sulla radio premere il tasto "Write".

■ 2.3.3. Parameters

My ID: idem menu radio.

Message from: idem menu radio.

Call Channel: Canale di chiamata, idem menu radio.

PMR TX: Attiva la trasmissione per la banda PMR446 (UHF).

Ham TX: Attiva la trasmissione per tutte le altre bande che la radio può supportare in Rx. Attenzione, può farlo teoricamente, ma i terminali non sono previsti per quello, ci sono dunque rischi di bruciarli.

MemCh WR: Abilita la scrittura nei canali di memoria della radio.

Always CallCh: La radio si accenderà sempre sul canale impostato come Call.

Filter Msgs: Attivarlo per filtrare solo i messaggi indirizzati a se e non riceverli tutti.

White Blip: Il LED bianco della radio fa un lampo ogni tanto per indicare che è sempre accesa, utile se si sta al buio. Da usare tipo boa (Beacon)

Mic AGC: Automatic Gain Control, guadagno automatico del microfono.

FSK syled: Il LED bianco lampeggia quando viene ricevuto un SYNC FSK. *La radio è sempre in ascolto, il DSP del chip radio cerca in continuazione il "preambolo" (una serie di 0/1 di trasmissione, e se lo trova "si sincronizza") e poi cerca una "chiave" (serie di bit) che indicano inizio di pacchetto. A quel punto genera un "interrupt" di SYNC e si accende il LED.*

Keys Locked: Blocco tastiera.

Keys Autolock: Blocco tastiera automatico.

DTMF sidetone: Mentre si preme PTT, è possibile sentire i toni DTMF inviati.

Beep Keys: Beep della tastiera.

Beep Message: Suoneria alla ricezione di un messaggio.

Beep SelCalls: Suoneria alla ricezione di una chiamata selettiva.

Msg ack: Attiva la conferma di ricezione del messaggio inviato.

Defaults applied: All'avvio la radio regola certi parametri importanti per il buon funzionamento di PrepperRadio.

Nation Code: Imposta la radio secondo la ragione del mondo in cui vi trovate. (Non attivo per il momento)

T Tone len cs: Tail Tone length: Durata del tono di fine chiamata in centesimi di secondo. Default 30 (3 decimi di secondo).

Beep dT cs: Durata dei beep delle suonerie in centesimi di secondo. Default 0.

MIC gain: sensibilità del microfono. Consigliato 16. Max = 31. (0.5dB a step)

Keys lock tout: Keys lock time out. Tempo prima che si inneschi il blocco tastiera. Da moltiplicare per 6 secondi.

POn Time ds: Power On Time, tempo di visualizzazione della schermata di avvio. Decimi di secondi.

Beep vol: Beep volume. Cambia il volume per tutte le suonerie della radio.

LEDW2 on ds: Nella posizione 2 del LED bianco (Flash Light), decide quanto tempo rimane accesa.

LEDW2 off ds: Nella posizione 2 del LED bianco (Flash Light), decide quanto tempo rimane spenta.

ScL N Audio ds: Durata di ripresa scansione dopo aver trovato un segnale in modalità noaudio. Min 1, max 100, default 6.

ScL W Audio ds: Durata di ripresa scansione dopo sparizione del segnale in modalità **5s**. min 1, max 100, default 100.

Welcome 1: Seconda linea nel messaggio di avvio.

Welcome 2: Terza linea nel messaggio di avvio.

UpCode: idem menu radio.

DownCode: idem menu radio.

VOX

Audio VOX: Attiva il VOX.

Detect delay: Detection Delay Time in 128ms. Durata di tempo prima di considerare un volume di voce. Default 8, max 15.

Detect interval: Detection Interval Time. Durata media dove individua se c'è voce. Default 8, max 15.

Off level: Soglia del volume ascoltato che considera del silenzio.

On level: Soglia del volume ascoltato che considera voce. Visibile nell'indicazione del livello del microfono sul display a sinistra.

Min time ds: Tempo minimo dell'attivazione della trasmissione in VOX.

Battery Save in centesimi di secondo.

BS1 off	20	BS2 off	40	BS3 off	60	BS4 off	80
BS1 on	80	BS2 on	60	BS3 on	40	BS4 on	20

2.3.4. FM Channels

Elenca le frequenze FM Broadcast memorizzate nella radio.

2.3.5. Log Panel

Log degli eventi della finestra EEPROM.

2.4. Messagy via PrepperDock

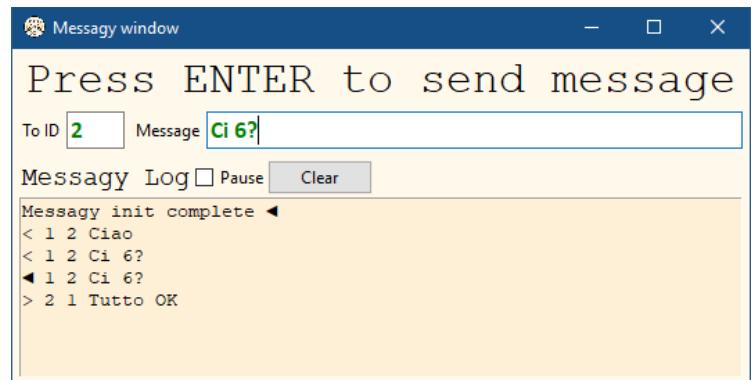
La versione Dock di Messagy può scrivere fino a 40 caratteri contro i 15 della radio.

In pratica la radio diventa un modem per inviare messaggi comodamente dal computer.

> Messaggio ricevuto.

< Messaggio inviato.

◀ Messaggio inviato con conferma di ricezione.



Video illustrativi:

PrepperDock-Messenger

[https://odysee.com/@maxlinux2k:c/PrepperDockMessenger:d?
r=H1SPCWjv2TqR2kqnKP42NCRUoVdwuyLk](https://odysee.com/@maxlinux2k:c/PrepperDockMessenger:d?r=H1SPCWjv2TqR2kqnKP42NCRUoVdwuyLk)

2.5. DTMF

Simile alla finestra di Messagy, permette d'inviare codici DTMF.

La struttura dei "comandi" DTMF

Poiché i comandi sono indirizzati a un utente o gruppo specifico, essi DEVONO:

- SEMPRE avere [to-id]*[from-id]*
- I parametri sono SOLAMENTE numerici e terminano con un *.
- La fine della linea di comando, è sempre e solo A*.
- Il singolo * serve per delimitare parametri di un comando.

Si potranno mettere più comandi in una riga separandoli da AA.

Esempi di comandi gestiti: di seguito il destinatario è 123 e il mittente 345, il codice di gruppo **222**.

- Semplici numeri, eg 123232 non danno attivazione della gestione chiamate
- NON si può chiamare un utente senza mittente: 123* il mittente non appare e non si può ricevere ACK.
- Chiamata a un utente specifico, senza cambiare canale: **123*345A***
- Chiamata a un gruppo senza cambiare canale: **222#345A***

Il comando codice 3 è usato per indicare su quale canale spostarsi, ipotizziamo di spostarsi sul 23, chiamata personale: **123*345AA3*23A***

Definizione di canale [99] frequenza 466.1100, DCSS 1 comando 4.

Richiesta indirizzata al corrispondente 123:

123*345AA4*99*4461100*1A*

ACK su richiesta di ring, notare come 345 sia adesso il destinatario

345*123AA1A*

NACK su richiesta di ring, da vedere di introdurre le motivazioni del NAK

345*123AA2*11A*

Richiesta di allarme con codice di allarme 123

345*123AA4*123A*

La parte "difficile" è fare qualcosa di ragionevolmente affidabile, estensibile, con una struttura logica "costante" e possibilmente compatibile.

Il punto di partenza è che destinatario e mittente vanno SEMPRE identificati, già ci sono cani e porci, in aria, se neppure si chiede un minimo di ID è il caos totale.

Un mix di DCSS + ID ... forse rende le selective call DTMF decentemente immuni dal primo che passa.

```
/**
 * You CANNOT change them at will, since they are documented AND sent around
 */
enum Dtmf_Cmds
{
    DTMF_CMD_NULL,           // ALWAYS use zero as NULL value
    DTMF_CMD_ACK,            // Will do what requested
    DTMF_CMD_NACK,           // Cannot do what requested, one code for the NAK reason
    DTMF_CMD_USECH,          // request to use a specific channel
    DTMF_CMD_TALKNOW,        // one of the two signal to the other that it is going to move to the given channel now, use param, t
    DTMF_CMD_SETCH,          // configure the given channel with F and dcss
    DTMF_CMD_ALARM,         // Emit an alarm tone one param, the tone to emit
};
```

Stay calm and switch on the radio!



<https://t.me/+3S1rKwPf-2AxMTJk>

