

Robust Chat — Complete Instruction Manual

1. Overview

Robust Chat v2.20 is an integrated amateur-radio communication suite designed for reliable digital chat under weak-signal or narrow-bandwidth conditions. It unites text messaging, beaconing, position sharing, automatic relaying, CAT transceiver control, and AI-assisted operations within a dark, compact interface.

Supported hardware includes the LiNK500 TNC, Teensy RPR TNC, and SCS Tracker(untested). The LiNK500 is ideally suited, offering both an internal robust modem and a separate CAT control interface. Bluetooth connectivity is also available RF data exchange (RF testing is required in BT).

All operational data—settings, messages, link graphs, and positions—are stored in JSON files beside the executable, in the same folder, ensuring portability between computers.

2. Installation & Startup

System Requirements:

- Windows 10 or 11
- Python 3.9+ (for .py version)
- Packages: pyserial, cryptography
- Hardware: TNC (LiNK500, Teensy RPR, SCS Tracker), GPS receiver

Installation:

1. Create a folder named Robust Chat and place Robust_Chat.exe and JSON files in that same folder.
2. Run via Python (install dependencies if using python) or simply double-click the EXE Windows version.

Startup:

On launch, the program loads fonts, settings, COM ports, and saved data. The window applies dark mode automatically. Logs appear in the console as [LOG] entries, which can be found in the Chat window under Settings (if required) .

Auto TNC: v2.25 introduces an auto connect TNC check box, this is in case the user wants to put the program in the startup folder to auto open on power interruption, so that the tnc auto connects and the station resumes as the program opens i.e for 24/7 operation.

3. Main Interface Layout

The interface is divided into three main areas:

- Header Toolbar: Contains TNC COM selection, MYCALL, GRID, R300/R600 buttons, frequency dropdown, and navigation tabs (Main, Groups, GPS, Beacon, Locator, CAT, Misc).
- Sidebar: Displays chats and the 'Beacons Heard' list with colour-coded freshness (green, amber, red).
- Main Page: Chat messages, input field, Encrypt/Decrypt buttons, Send button.

Outgoing messages have a green background, incoming messages directed to your callsign are orange text, and all timestamps are local to your position.

4. Connecting to TNC

Before connecting, enter MYCALL (your callsign) and GRID (Maidenhead locator, 6 (5KM) or 8 (500M) characters). These must be set before the Connect button becomes active.

Supported TNCs:

- LiNK500 TNC: Dual USB ports (Port A for data, Port B for CAT) and Bluetooth (connection tested:OK, RF TX RX needs testing in Bluetooth mode to confirm no lags in data).
- Teensy RPR TNC and SCS Tracker: Standard KISS-compatible devices.

Select COM port, click Connect. Successful KISS mode initialization is logged as [TNC] KISS mode active. Optional CAT control supports Icom (CI-V), Kenwood/ELECRAFT ASCII, and Yaesu Binary protocols. R300 and R600 symbol rates are selectable at runtime.

5. Message System

Messages are composed in the Main tab using the To: field, text input, and Send button at the bottom of the screen.. Directed messages use ACK confirmation; broadcasts use CQ.

Messages are stored automatically in messages_v1.json for EMCOM and general message history. CLEAR MESSAGES button will clear the window and the messages in the messages_v1.json as well as any beacons in the link_graph.json, for a fresh start.

Groups Tab: In the groups tab you can enable certain callsign messages and beacons only in the received window, think whitelist. This would be beneficial if you wanted to form a group or net without CQ calls or non-member messages or beacons. In the Groups page you will see fields to type in the callsigns allowed, and then click Enable. The status of the Filter is shown in the Chat list, either 'All' for filter off, or FILTER ON, for members only.

WARNING ENCRYPTION MAY BREACH LICENSE CONDITIONS: Encryption uses AES-256-GCM. The ACK system provides one-tick (sent) and two-tick (confirmed) indicators with retries. Auto-relay silently forwards unacknowledged messages once. AI-assisted chat generates replies through LM Studio or compatible endpoints. Message colours and UTC

timestamps improve readability.

6. Encryption System

In the UK certain 'User services' can request encrypted messages be sent in emergencies or high priority situations, by RAYNET. So it is included for such emergency use. Use outside of this may breach license conditions.

Messages can be encrypted using AES-256-GCM. The encryption workflow:

1. Type Message in send message field and enter To: CALLSIGN. Press PADLOCK to Encrypt before sending.
2. Enter password; press Encrypt; ciphertext replaces plaintext with [ENC] prefix. SEND
3. Receiver presses Decrypt and enters same password. For operational purposes it would be suggestible that groups have a daily changing password based on a table of day date month, words assigned to each. In this case a user can quickly look at the table i.e Monday = aPple , 14 = rEd, January = JupiTter > aPplerEdJupiTer , for example.

No keys are stored; security depends on user password integrity.

7. Acknowledgment (ACK) Reliability System

Directed messages include an ACK ID (e.g., [ACK:84JQ]) 3 retry system. The receiving station automatically replies with an identical ACK frame.

Indicators:

- ✓ Sent (awaiting ACK)
- ✓✓ Delivered (ACK received)
- ! Failed (no ACK after retries)

Pending ACKs persist after restart until confirmed or expired.

Note: If a station has its Group filter on and you are not in the membership your message will appear delivered, but won't be viewable until they turn the filter off.

8. Auto-Relay System

Stations may silently relay third-party messages once if both sender and receiver are known from recent beacons. Logged as [AutoRelay] events. Auto-relay reduces delivery failures across multi-hop paths. Can be toggled in settings.json. This is triggered on conditions that a call is heard to a callsign which is not ack'd, if the To callsign is in your beacon heard list, auto relay sends a one shot TX of the original message, it returns the same way but may use an alternative users system if they have the destination in their beacon heard list.

9. Beacon System

Automatic beacons broadcast callsign and position at intervals: Off, 5, 10, or 15 minutes. Each received beacon updates link_graph.json, forming a network topology with coloured freshness (green/amber/red). Entries older than the set threshold are pruned automatically. Beacons are sent with co-ordinates with a priority order of live GPS first and then Fixed co-ordinates and finally if none of those exist then the Maidenhead Grid is converted to lat lon co-ordinates. Note if you can find your maidenhead grid in 6 figures you have a 5KM precision.. This info is used to plot stations in the LOCATOR cardinal radar system tab.

10. Locator and Mapping

The Locator tab prioritizes GPS, but falls back to Maidenhead system to plot nearby stations as compass markers showing distance and bearing from your station. positions.json stores known coordinates, which are removed after 24 hours. Your own position appears in the centre.

11. GPS Integration

Supports dual GPS sources: PC GPS and LiNK500 TNC and TEENSY RPR TNC GPS. NMEA sentences are parsed to update latitude and longitude fields. Fixed manual coordinates can be entered if no GPS is available. Data saved to positions.json. The PC GPS is refreshed every 5 minutes. If using the GPS from the TNCs this is a manual GET GPS process, as it requires the Chat system to exit KISS mode get GPS and reset KISS mode each time.

12. CAT (Computer-Aided Transceiver) Control

The CAT tab provides radio control via Icom CI-V, Kenwood/ELECRAFT ASCII, or Yaesu Binary protocols.

Auto-QSY: For 24hour stations, switches between day and night bands(default 05:00–21:00 / 21:01–04:59). Manual frequency selection is available via dropdowns for each portion of Day and Night..

ALE-Lite: Ale-lite is a scanning feature frequency selectable, essentially this hunts for Robust Chat program beacons. If you enable a frequency it will be part of the scan frequencies. The system will dwell for 10 seconds on each frequency listening for a beacon, and when it finds one will stop scanning and settle on that frequency.

13. Offline AI Assistant Mode

An optional AI enable checkbox connects to a local LM Studio or compatible API (127.0.0.1:1234). When enabled, it can auto-reply or summarize messages using a 'ham operator' personality. AI settings (endpoint, model, prompt) are adjustable in settings.json.

Operators with AI, their beacons appear in lime green with [A.I] tag. LM studio is free and when downloaded <https://lmstudio.ai> allows you to download A.I models and when enabled in Robust Chat a local OFFLINE A.I is available and responds to the MYCALL set, the idea is that those with modern PCs can install local A.I with no internet required after download to serve other Robust Chat users on any topic they require i.e survival, mechanical, electronic, or anything you can think of! Further instructions are on the A.I page of Robust Chat.

Note: L M Studio is required to be open and running while in A.I mode

14. Miscellaneous Features

- R300 / R600 speed toggle
- Auto-save of all settings
- Ctrl + Alt + Q diagnostic hotkey
- Unicode fonts (Segoe UI Emoji / Noto Color Emoji)
- Dark theme
- Compact, field-optimised layout

15. File System and Persistence

JSON files stored alongside executable:

- settings.json — configuration
- messages_v1.json — chat history
- link_graph.json — beacon links
- positions.json — known coordinates
- radios.json — radio definitions
- freqs.json — frequency list

Copy the entire folder to transfer to another system. Backup messages_v1.json regularly.

16. WIFI Tablet/Phone control (as of v2.26)

This feature is to enable the wire free use of controlling a nearby Robust Chat transceiver via your tablet or phone. For example if you had the Robust Chat PC, TNC, Transceiver in a case in the trunk of your car and wanted a Tablet upfront to send and receive messages wirelessly.

-Enable the Remote Wifi Tablet Control in the CAT page

-In Windows 11, search 'Mobile Hotspot Setting'

-Turn on 'Mobile Hotspot', edit the Properties, change name to 'Robust Chat' and set 'Password' to something memorable, perhaps your callsignx2 i.e 'K1ABCK1ABC'

- On Tablet or Phone, connect to 'Robust Chat Wifi', goto the IP address shown in the CAT page by entering it in your Tablet or Phone's browser.

-Bookmark it for future use in your toolbar

-Log in by entering your callsign in CAPS as your Password (this is a separate log in to the wifi passwords, this one on screencast needs your callsign once)

[Optional : If you want to have your Windows PC have the 'robust chat hotspot' on at start up as a permanent feature you can do that in the Windows task scheduler (You can open the Task Scheduler in Windows 11 by searching for "Task Scheduler" in the Start menu.

Alternatively, you can press Win + R, type taskschd.msc, and press Enter to open it with administrator privileges.)

Download the instructions to set this up and EnableHotspot.ps1 file in the folder named [HOTSPOT] in the links you will find via the www.robust-chat.com site]

17. Troubleshooting and Diagnostics

Common issues:

- Missing pyserial/cryptography → reinstall dependencies.
- COM port busy → close other serial apps.
- CAT failure → check CIV address and baud rate.
- GPS missing → verify port and NMEA output.
- Messages missing → confirm JSON file paths in EXE build.
- Radio list/frequencies missing from drop downs > Check Radios.json and freq.json are present in the same folder as the program.

18. Version History and Credits

Version 2.27 Highlights:

- CAT integration
- Dual GPS (PC + LiNK500)
- Bluetooth detection for LiNK500
- ACK + Auto-Relay subsystems
- AES-256-GCM encryption

- AI assistant integration (requires LM Studio)
- ALE-Lite; band scanning
- Auto Start TNC; option to enable TNC to connect automatically on startup
- Remote Tablet/Phone control added over Wifi (same wifi network required)
- Fixed LAB599 TX-500MP CAT Control (which was not working in 2.26)

Credits:

Development: Grok4/GPT5 and Oli M0OLI.

Documentation: 13th Nov (2025)

Acknowledgments: Thank you to those initial testers of Robust Chat and promoters of the testing group Julian OH8STN, the details of which can be found on his website. Designer of the LiNK500 TNC Oliver Harms DL4KA. Robert Wolters DM4RW designer of the Teensy RPR TNC, and SCS Tracker team.