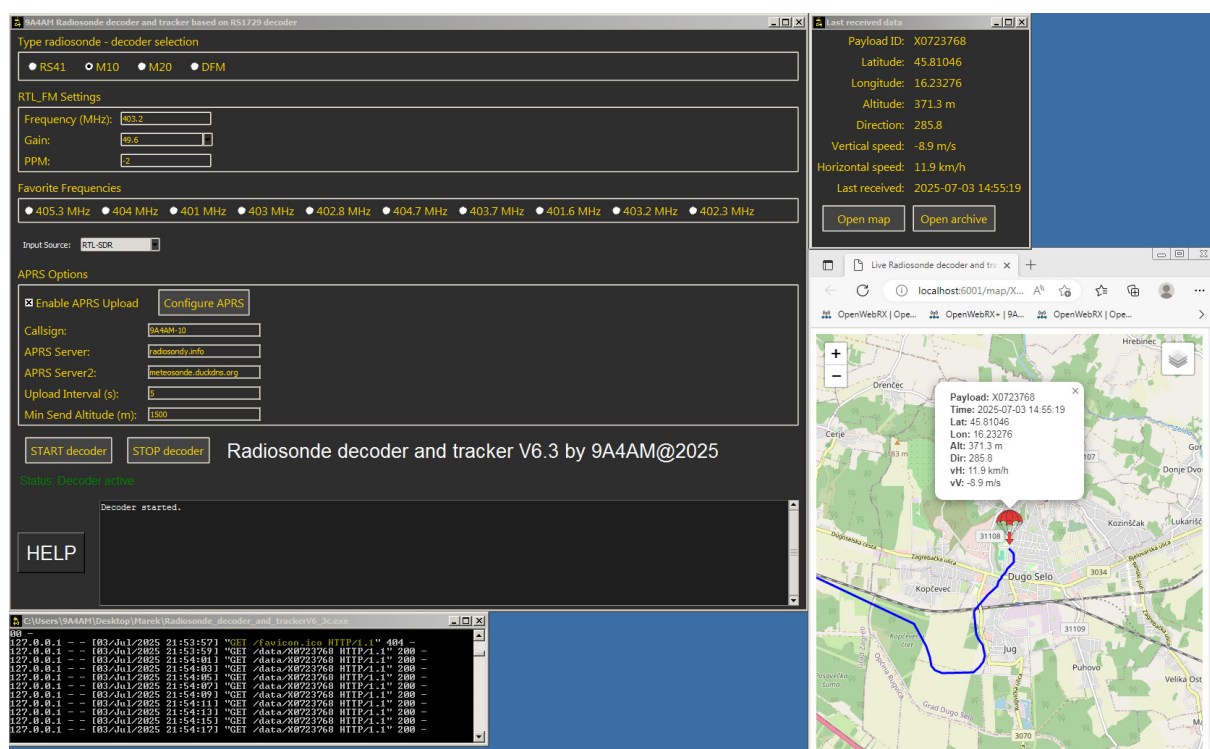# Uputstvo za korištenje
Radiosonde decoder and tracker
by 9A4AM za Windows OS 64bit

(trenutna verzija 6.3 – 03.07.2025 – 9A4AM, Mario Ančić)

(HRVATSKI)

UVOD

Radiosonde decoder je izrađen za Windows 64 bit operativni sustav. Zamišljen je da se distribuira kao jedan jedini EXE koji nakon pokretanja postavi sve što je neophodno za rad dekodera. Sami dekoder je baziran na RS1729 dekoderu meteoroloških sondi i podržava 4 tipa sonde i to: RS-41, M10, M20 i DFM (nekoliko podtipova). U samom početku je zamišljen da radi samostalno sa popularnim SDR uređajem RTL-SDR baziran na RTL2832. Da bi pravilno radio treba instalirati drivere i to mora biti ZADIG driver. Tko se već koristio sa SDR, zna već kako ide i ako ima instalirano, ne treba ništa više za korištenje Radiosonde decoder and tracker. Kasnije sam dodao i opciju da se dekoder koristi i preko Audio ulaza koje ima PC (potrebno koristiti neke od Virtual Audio i SDR# ili SDR++ ili nešto treće). Tu kvaliteta dekodiranja jako ovisi o audio postavkama, pa se treba malo potruditi, dok kod izravnog korištenja RTL-SDR-a nema potrebe za podešenjima, a i bolje dekodira, jer koristi IQ signale i njih obrađuje. Sami dekoder je napravljen kako bi pratili meteorološke radiosonde, te njihove podatke – telemetriju prikazali na prozoru namijenjenom za to, zatim za svaku primljenu sondu imamo spremljeno sve u log datoteku koja se nalazi u LOG direktoriju unutar direktorija gdje je aplikacija. Također se može prikazati let i podaci na karti u realnom vremenu. Prikaz na karti je moguć i učitavanjem neke log datoteke. Na karti će se prikazati cijeli let i marker na zadnje primljenom paketu sa zadnjim podacima u tzv Popup. Karta se prikazuje u Web pregledniku koji je postavljen u Windows OS kao zadani preglednik. Također je implementirana opcija za slanje na neki od servera (podržava slanje na 2 APRS servera -opcija), APRS (meteosonde.duckdns.org ili radiosondy.info). Sve je podesivo unutar GUI-a, ali i preko datoteka u direktoriju aplikacije (settings.ini, aprs_settings.ini i favlist.txt).

Struktura programa:

App direktorij:

                         \<LOG>

                         Radiosonde_decoder.exe

                         settings.ini

                         aprs_settings.ini

                         favlist.txt

Izgled glavnog prozora i prozora s podacima o zadnje primljenom paketu

Pokretanje aplikacije je na ikonu, ukoliko je stavljena na radnu površinu:


Radiosonde...
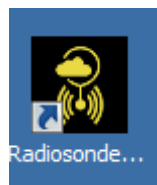
Sada malo detaljnije objašnjenje i kako podesiti sustav.

Na samom početku imamo odabir **tipa dekodera** po tipu sonde koju želimo dekodirati. Kao što sam već u uvodu naveo, dekoder podržava slijedeće tipove sondi: **RS-41, M10, M20 i DFM(PS-15, DFM17 i DFM09)**. Odabir je jednostavan klikom na opciju kraj opisa dekodera.
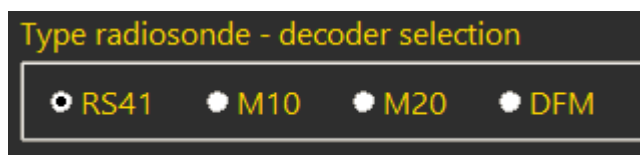

Važno: **Nakon odabira dekodera potrebno je zaustaviti dekoder, ukoliko je bio pokrenut, te ponovno pokrenut dekoder. Inače gumb Start decoder ima i funkciju snimanja izmijenjenih opcija.**
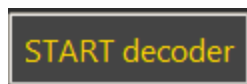

Slijedeće je dio sa podešenjima SDR prijemnika (koristi se RTL_FM). Tu upisujemo **frekvenciju** u obliku 405.3 u MHz. **Gain** se, ovisno o potrebi korisnika, upisuje u slijedeće polje, tj tu je opcija odabira u koracima koje RTL-SDR ima. Za najbolji prijem je najbolje staviti na 49.6, ali treba napomenuti da, **ako se koristi LNA treba smanjiti Gain da ne dođe do preopterečenja ulaza ili čak oštećenja SDR-a**.

I na kraju imamo polje s **ppm**, a to služi ukoliko SDR odstupa s frekvencijom, najčešće kod jeftinih kineskih kopija koji nemaju TXCO, ako je u pitanje RTL-SDR V3 ili V4, tada nije potrebno, već ostaviti na 0.
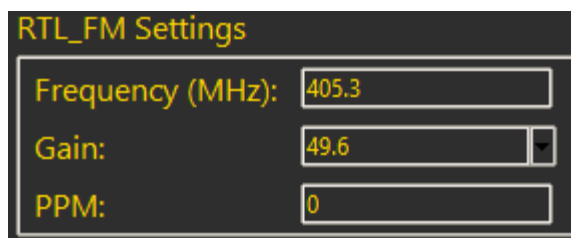
Jedna od korisnih opcija je korištenje favorit frekvencija gdje su stavljene najčešće korištene frekvencije za brzi izbor. Po pokretanju dekodera automatski se generira 10 frekvencija koje će biti dostupne za brzi odabir. Izmjena tih frekvencija je moguća uređivanjem datoteke favlist.txt, te nakon restarta će biti prikazane one koje je korisnik definirao.
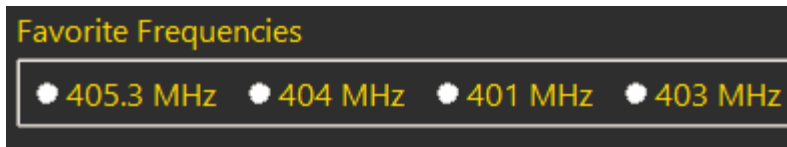

Slijedeće je odabir izvora za dekoder. Opcija je RTL-SDR ili neki od audio ulaza koji nude Windows-i. Moja preporuka je svakako korištenje RTL-SDR direktno, jer je tako najbolje dekodiranje pošto koristi direktno IQ signale i ne ovisi o podešenjima kao na primjer ako se koristi SDR#. Za audio ulaz koristiti neki od Virtual Audio (koristi SOX za obradu audio signala koji je dio aplikacije, te nije potrebna instalacija).
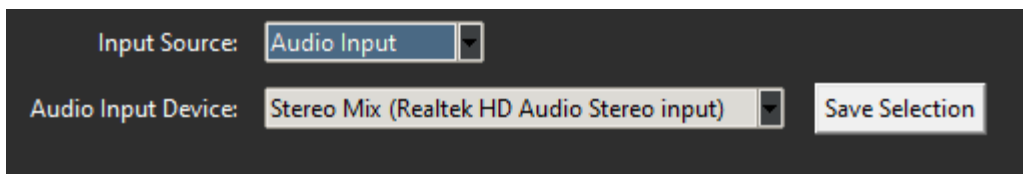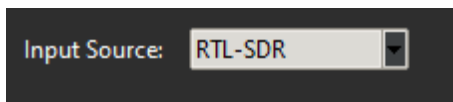
Nakon odabira ili izmjene audio ulaza potrebno je potvrditi tj snimiti odabir na gumb **Save Selection**. Izmjene će spremiti u settings.ini i ostat će kao odabir kod pokretanja dekodera ili do izmjene.
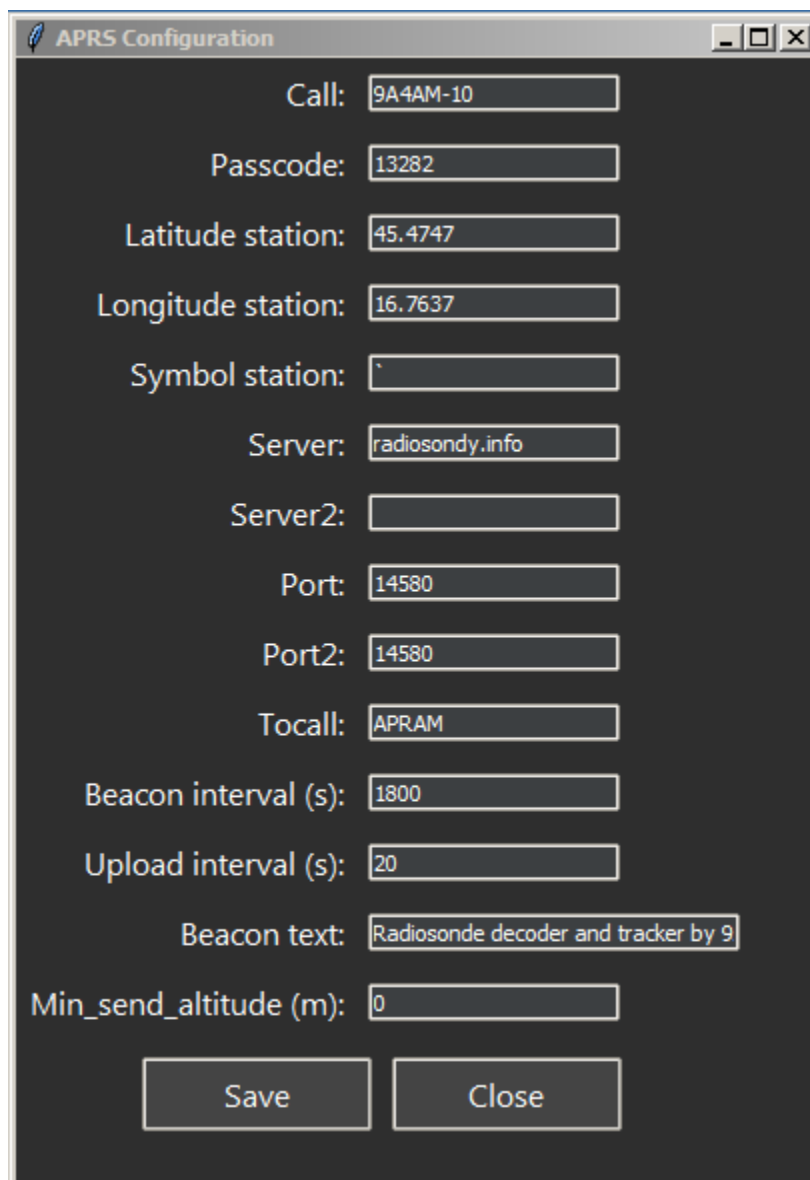
Dolazimo i do dijela koje služi za uključivanje slanja na neki od APRS servera (meteosonde.duckdns.org ili radiosondy.info ili wettersonde ili neki drugi).
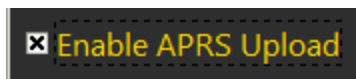
Ako je **X** na **Enable APRS Upload** i podešene postavke za korisnika, telemetrija će biti poslana na server u odabranom intervalu. Ovdje je bitno napomenuti da interval ne bude premali, jer ima i drugih korisnika pa da se ne preoptereti server. Neki preporučeni interval za slanje je 20 sekundi. U donjem djelu je samo prikaz trenutnih parametara pohranjenih u aprs_settings.ini i tu nije moguće mjenjati, već za izmjenu treba kliknuti na gumb **Configure APRS**. Ukoliko nema upisanih podataka za spajanje na APRS server, nije moguće pokrenut tu opciju! **Upload mora biti ISKLJUČEN!!**

Klikom na gumb se otvara prozor.

Ovdje treba podesiti postavke za APRS. Call je odabir kako će se zvati i prikazivati prijemna stanica. Najčešće pozivni znak, jer APRS je namijenjen radioamaterima sa licencom i osobnim pozivnim znakom), no međutim da prijem radiosondi ne treba biti radioamater pa je moguće staviti i drugačije ime. Bilo bi dobro da nije dulje od 9 karaktera. Passcode se generira na nekom Online servisu za odabrano ime tj pozivni znak (npr https://apps.magicbug.co.uk/passcode/ ili bilo koji drugi).

VAŽNO: Ne koristiti javne tj regularne APRS servere tipa finland.aprs2.net ili druge, ako se ne koristi validni radioamaterski pozivni znak. Također ne slati telemetriju na te servere, jer sondama nije mjesto na tim servisima. Jedino ako se pusti reprogramirana sonda s radioamaterskim važećim znakom dopušteno je slanje podataka na te servere.
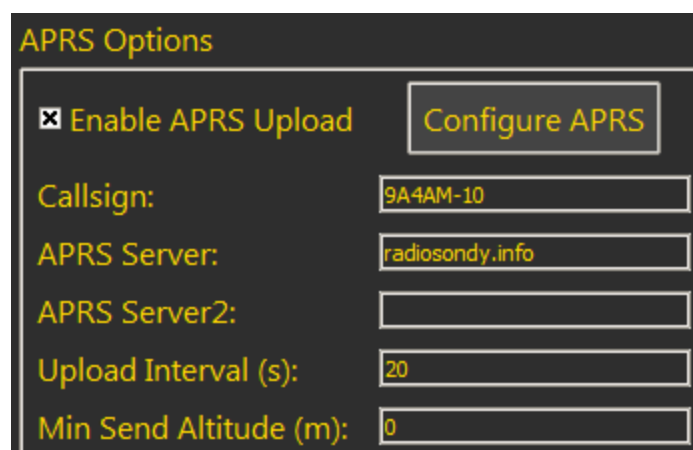
Korisiti isključivo servere: meteosonde.duckdns.org ili radiosonde.info ili drugi namjenjen za prijem i praćenje radiosondi.

Latitude i longitude unijeti za lokaciju prijemene stanice i to u decimalnom obliku (npr. 45.55555 i 16.55555). Simbol station je APRS simbol za tzv Dish Antenna, a to je potrebno da serveri raspoznaju prijemnike za radiosonde. Tu opciju nije moguće mijenjati i ona je fiksna u kodu, kao i Tocall i Beacon text. Server odabrati kao što je ranije objašnjeno, a port je **standard za APRS 14580**, međutim neki serveri koji su pisani za osobne potrebe ili slično, mogu koristiti i druge portove, pa je dopuštena izmjena. Beacon interval je 1800 sekundi tj 30 minuta, izmjene su moguće, ali nema potrebe. Beacon je slanje podataka o prijemnoj stanici na server kako bi bila vidljiva na karti i popisu prijemnika. Uplaod interval je moguće mijenjati, ali je preporuka 20 sekundi ili više. To je vrijeme koje definira interval slanja telemetrije dekodirane sonde na server.

Od verzije 6.1 uveden je i drugi, opcionalni APRS server, pod stavkom Server2 i Port2. Ova opcija služi za slanje na dva APRS servera istovremeno, beacon i telemetrija. Uključuje se kad i prvi tj glavni APRS server s Enable APRS Upload.
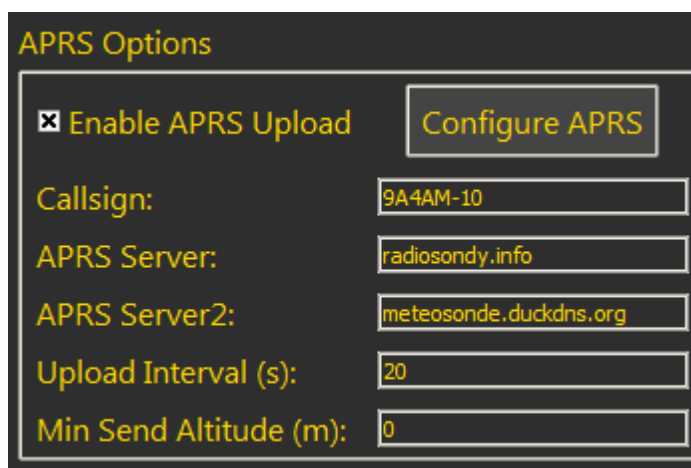
VAŽNO: Ako se ne koristi drugi server, već samo jedan, onda taj mora biti pod Server i Port definiran, a Server2 ostaviti prazno (ništa ne smije biti u polju, jer inače generira grešku, pošto misli da je server upisan ispravno), dok u polju Port2 neka ostane upisano.

Slanje samo na Radiosondy:

Ili slanje na Radiosondy i na Meteosonde server:

I posljednja opcija je **Min Send Altitude**, to je opcija ako se želi staviti neki donji limit kod kojega, ispod zadane vrijednosti, ne šalje na server telemetriju. Ako je 0, tada šalje do kraja leta.

Klikom na **SAVE** pohranjuje se u aprs_settings.ini kako bi kod pokretanja ostalo spremljeno. Na **CLOSE** se zatvara prozor i odabir posprema u lokalne varijable za daljnje korištenje bez potrebe za restartom dekodera.

VAŽNO: Nakon bilo koje promjene u APRS koniguraciji, obavezno restart programa kako bi se učitale nove postavke iz konfiguracije.
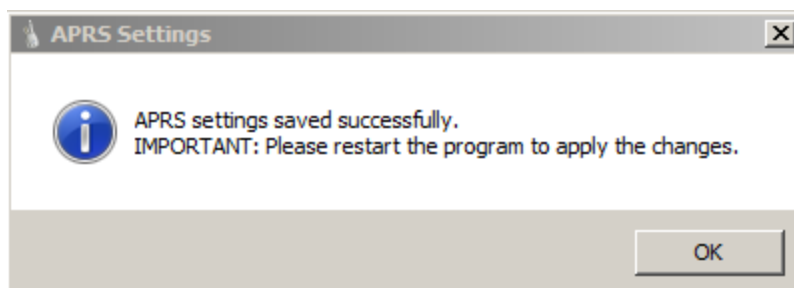
Na kraju su dva gumba **Start decoder** i **Stop dekoder**, koji služe za pokretanje tj zaustavljanje dekodera, inače **Start decoder** ima ranije opisanu funkciju spremanja odabira za SDR i dekoder. Nakon pokretanja dekodera i ako nema nekih grešaka, status se mijenja iz

Ili kad nije pokrenut

I u prozoru


```
Starting with device: Microphone (Realtek HD Audio Mic input)
Decoder started.
```

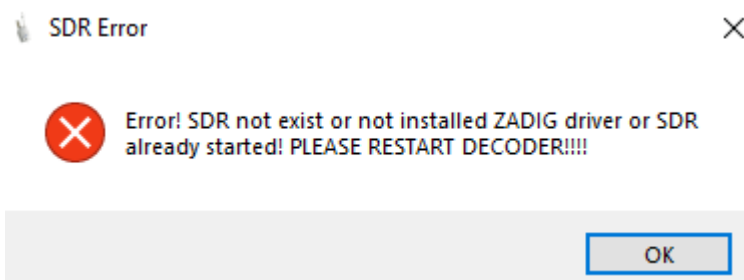Ili kad nije pokrenut


```
Decoder stopped.
```

Ako postoji neka greška, pojaviti će se ovako nešto.


```
Starting with device: Microphone (Realtek HD Audio Mic input)
Decoder started.
[ERR] error: wav header
```

To znači da je dekoder uspješno pokrenut, ali je problem s Audio ulazom, tu sam već napomenuo da treba koristiti neki Virtual Audio, a ovdje je greška jer je odabran Mic ulaz.

Na samome kraju u prozoru je moguće vidjeti dekodirane poruke, DEBUG poruke i dosta nekih informacija koje mogu biti zanimljive u otkrivanju poteškoća.

Ako nije priključen SDR (RTL-SDR) ili nije instaliran ZADIG Driver ili je neki program pokrenut koji koristi već SDR (može se dogoditi da je dva ili više puta pokrenuta i sama Radiosonde decoder aplikacija), dobijemo poruku:

SDR Error ✕

Error! SDR not exist or not installed ZADIG driver or SDR already started! PLEASE RESTART DECODER!!!!

OK

Ako se prikaže ova poruka, potrebno je otkriti problem s RTL-SDR-om i riješiti ga, te nakon toga OBAVEZNO restart programa.

Tipka HELP otvara ovaj dokument. Na PC treba imati instaliran neki od PDF preglednika preko kojega će aplikacija prikazati ovaj dokument.

HELP
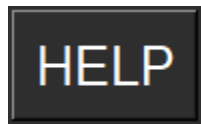
Drugi prozor sadrži podatke o zadnjem primljenom paketu sonde, ako je u LIVE modu dekodiranja, a ako je odabrana datoteka iz LOG-a, onda je prikaz zadnjeg podatka iz log datoteke. Payload ID je serijski broj sonde, za RS-41 je njen original, dok za M10 i M20 se koristi APRS broj (prilagođen zbog limita APRS servera na 9 karaktera), a DFM ima prije serijskog broja karakter D. To je isto standard za slanje na APRS servere za praćenje radiosondi. Ostali parametri prikazuju trenutnu lokaciju, smjer, brzine i datum i vrijeme zadnjeg dekodiranog paketa.

Klikom na gumb **Open archive**, otvara se sadržaj LOG direktorija i ako ima spremljenih log-ova sondi, prikazuje se, te odabirom na neku datoteku (logovi su bazirani na principu Payload_ID.log, npr X12345678.log ili ME1234567.log ili D12345678.log) i klikom na **Open** pojave se podaci iz loga i to zadnji paket u prozoru Last_received_data.
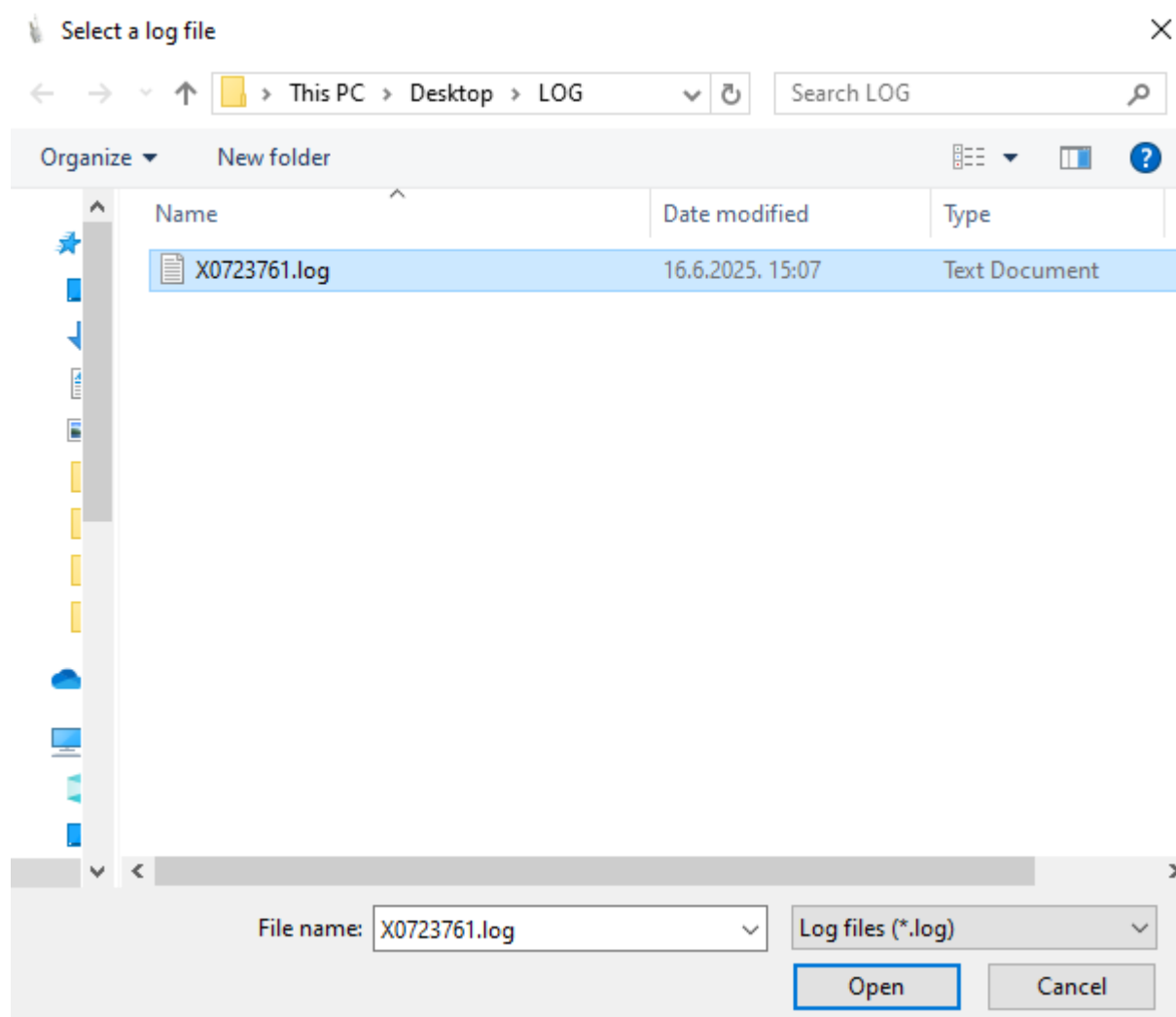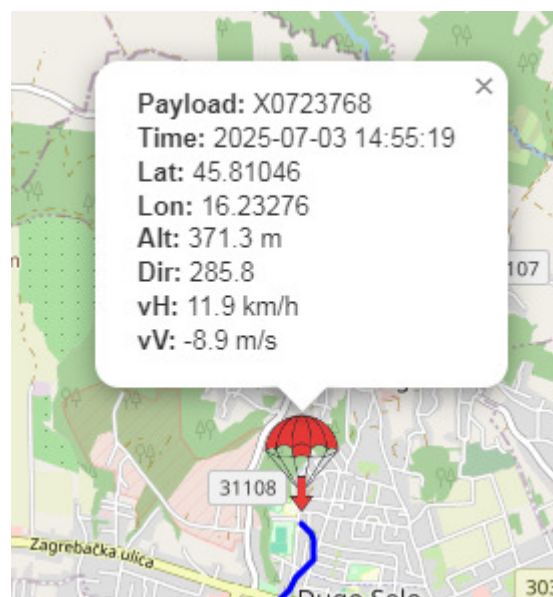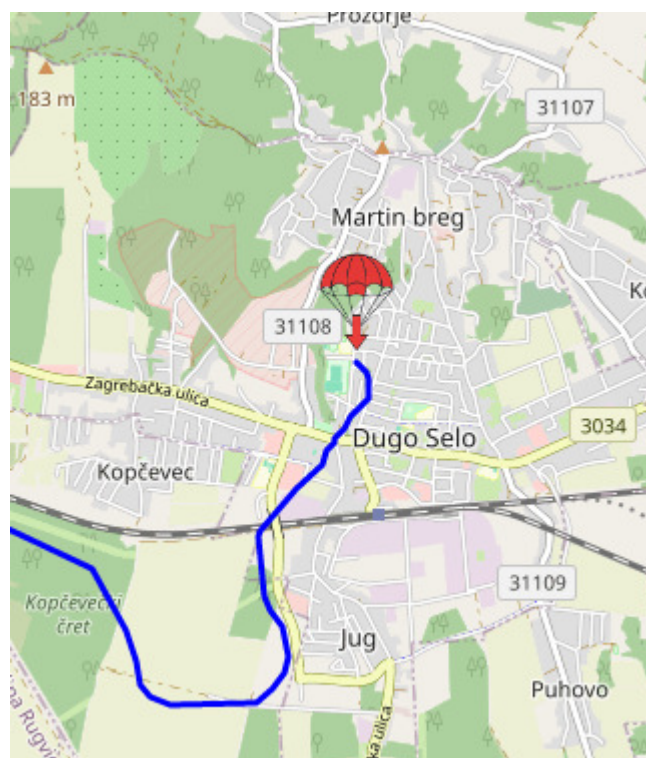
I prikaz u prozoru:
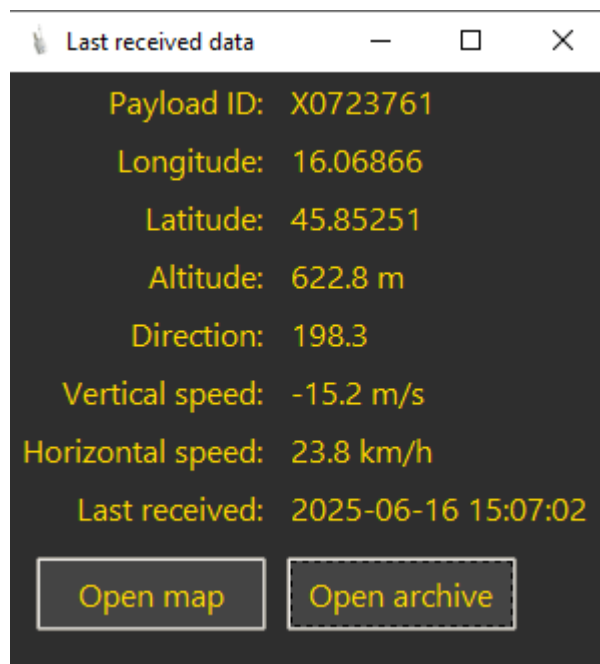
Kada postoje podaci u prozoru Last_received_data, dali LIVE ili učitani iz arhive tj LOG-a, klikom na **Open map** otvara se karta.

Karta (Leaflet) se otvara kao Web prikaz u zadanom internet pregledniku. Tu je samo bitno da neki drugi program ne koristi port 6001, jer aplikacija pokreće u pozadini server na tom portu, a pristup karti je na localhost:6001

Ako je LIVE dekodiranje sonda je prikazana na karti na trenutnoj lokaciji kao balon (ako je u uspinjanju) ili kao padobran (ako je u padu). Klikom na marker otvara se tzv Popup prozor sa aktualnim podacima. Osvježavanje je svake 2 sekunde. Karta se automatski pomiče, ako sonda dolazi do ruba karte u pregledniku tako da je uvijek u vidnom polju.

Eh sad jedna napomena, u LOG direktoriju, ako se koristi prikaz karte, kreira se html datoteka, npr X0723761_map.html. Te datoteke možete obrisati ako vam smetaju, ali i ne morate. Možda, u nekoj budućoj verziji stavim i brisanje, ali za sada ne, jer ne želim dodatna pisanja brisanja po disku korisnika. Inače Windows OS nema ugrađeni RAM-based temp direktorij kao Linux, pa zato smještam u LOG direktorij.

Prikaz na karti.

Payload: X0723768
Time: 2025-07-03 14:55:19
Lat: 45.81046
Lon: 16.23276
Alt: 371.3 m
Dir: 285.8
vH: 11.9 km/h
vV: -8.9 m/s

Izvori korišteni pri realizaciji ovog projekta:

https://github.com/rs1729/RS – RS decoder za radiosonde

https://github.com/rtlsdrblog/rtl-sdr-blog – RTL_FM program za SDR

https://github.com/chirlu/sox – program za obradu audio signala

https://github.com/projecthorus/radiosonde_auto_rx - pomoć u realizaciji, ovo je bila inspiracija za projekat uz RS1729 koji i sam AutoRX koristi

https://github.com/dl9rdz/rdz_ttgo_sonde – pomoć u realizaciji

http://oe5dxl.hamspirit.at:8025/aprs/c/ - dxlAPRS, pomoć u realizaciji

https://www.aprs.org/doc/APRS101.PDF – APRS protokol

https://apps.magicbug.co.uk/passcode - APRS passcode generator Online

https://radiosondy.info/ - APRS server i baza radiosondy, pomoć u testiranju i realizaciji

http://meteosonde.duckdns.org:1184 – APRS server i baza by 9A4AM, pomoć u testiranju

https://www.google.com/ - istraživanje i pomoć u realizaciji
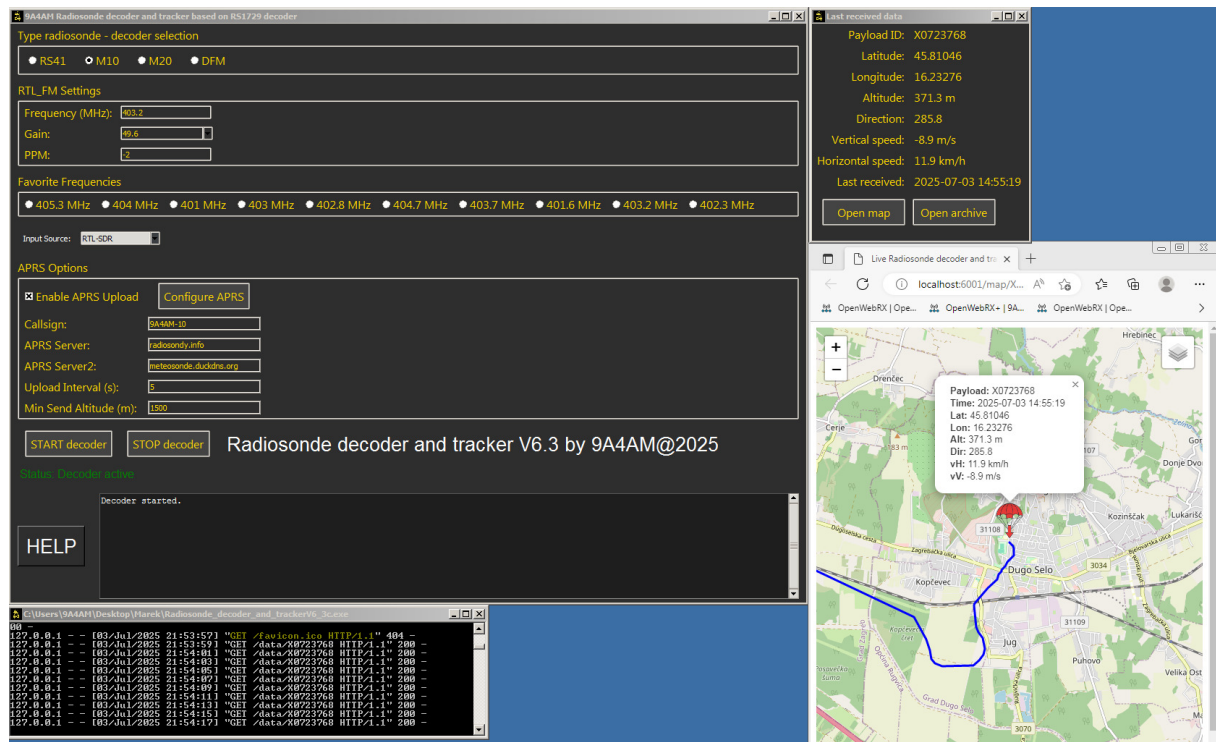
https://chatgpt.com/ - istraživanje i pomoć u realizaciji

Posebna zahvala ekipi iz Viber grupe **WX sonde** na pomoći i testiranju, te savjetima,a bez koje ovaj projekt ne bi bio nikada napravljen, a isto tako i za server.

User Manual

Radiosonde decoder and tracker
by 9A4AM for Windows 64-bit OS

(Current version 6.3 – 03.07.2025 – 9A4AM, Mario Ančić)

(ENGLISH)

INTRODUCTION

The Radiosonde decoder and tracker is designed for the Windows 64-bit operating system. It is distributed as a single EXE file which, upon launch, sets up everything necessary for operation. It is based on the RS1729 decoder for meteorological sondes and supports the following four sonde types: **RS-41**, **M10**, **M20**, and **DFM** (with subtypes).

Originally, it was intended to run standalone with the popular RTL-SDR device based on the RTL2832 chip. In order to function correctly, you must install the **ZADIG** driver. If you have already used SDRs, you probably have everything set up.

Later, I added support for decoding via the PC's audio input using tools such as **Virtual Audio Cable** with SDR#, SDR++, etc. In this mode, decoding quality depends on audio settings. However, when using RTL-SDR directly, no configuration is needed and IQ signals are processed for better performance.

The decoder tracks radiosondes and displays their telemetry in a dedicated window. All data is saved into a **log file** located in the LOG directory inside the application folder.

It also supports **real-time map display** in your default web browser, showing the full flight path and the last packet's data via marker popup. The map can also be loaded from a log file.

The application can optionally **upload telemetry to APRS servers** like (supports 2 APRS servers - optional):

- meteosonde.duckdns.org
- radiosondy.info

All settings can be configured via the **GUI** or configuration files:

- settings.ini
- aprs_settings.ini
- favlist.txt

PROGRAM STRUCTURE

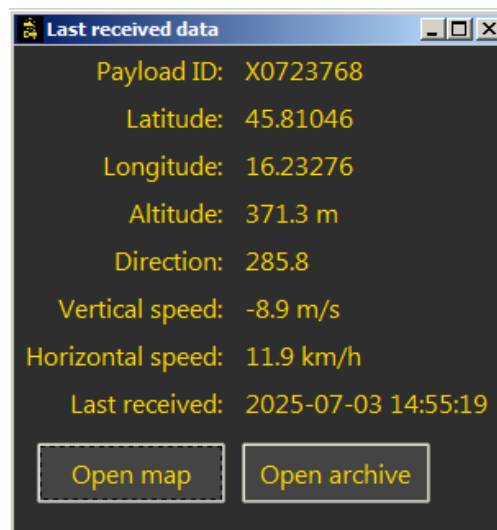**Application directory contents:**

<LOG>

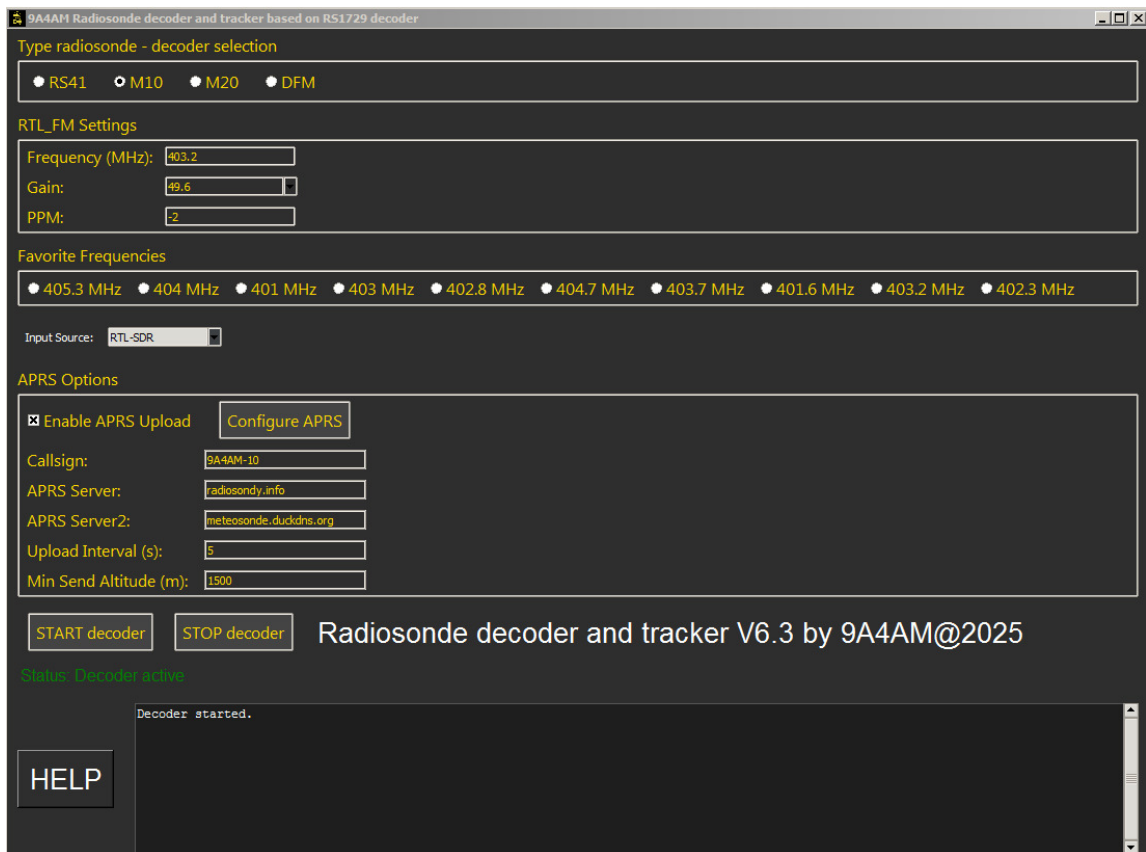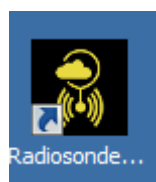Radiosonde_decoder.exe

settings.ini

aprs_settings.ini

favlist.txt

DECODER INTERFACE & CONTROL

To run the application, click the icon (e.g. desktop shortcut).

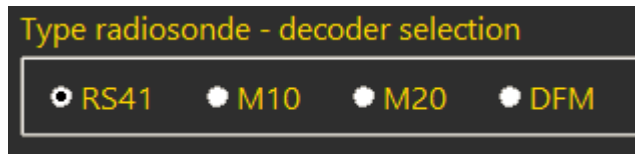Desktop icon

- **DECODER TYPE SELECTION**
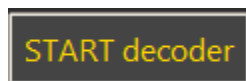
Supported radiosonde types:
- **RS-41**
- **M10**
- **M20**
- **DFM (PS-15, DFM17, DFM09)**

After changing the decoder type, always restart the decoder.
The Start Decoder button also saves settings.

SDR RECEIVER SETTINGS (using RTL_FM)

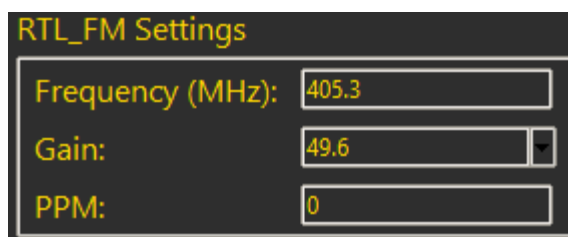In this section, you enter the **frequency** in MHz, for example 405.3.
The **Gain** is entered in the next field and depends on the user's needs — this is selectable in steps that the RTL-SDR supports.

For optimal reception, it is recommended to set the gain to **49.6**.

⚠ **Note:** If you are using an **LNA** (Low-Noise Amplifier), reduce the gain to avoid overloading the input or potentially damaging the SDR.

At the end, there's a field for **PPM**, which is used if your SDR deviates from the set frequency — this is common with **cheap Chinese clones** that do not have a **TXCO** (temperature-compensated oscillator).
If you're using **RTL-SDR V3 or V4**, no correction is needed — just leave the value at **0**.

FAVORITE FREQUENCIES

One of the useful features is the use of **favorite frequencies**, which allows quick selection of commonly used frequency values.

When the decoder starts, it automatically generates **10 default frequencies** that will be available for quick access.

You can customize these frequencies by editing the **favlist.txt** file.
After restarting the decoder, it will display the frequencies defined by the user.
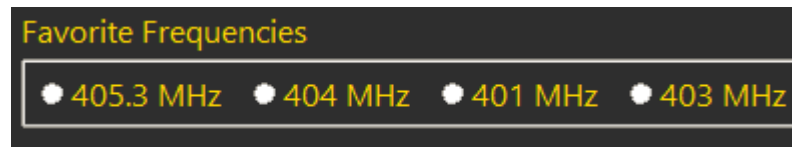
INPUT SOURCE SELECTION

The next step is choosing the input source for the decoder. The options are either **RTL-SDR** or one of the **audio input devices** available in Windows.

My strong recommendation is to use **RTL-SDR directly**, as this provides the best decoding performance. It processes **IQ signals directly**, which results in more accurate decoding and does not depend on external software settings like when using **SDR#**.

If you choose to use audio input instead, select one of the **Virtual Audio devices**. The application uses **SOX** (included) for audio signal processing — **no separate installation is required**.
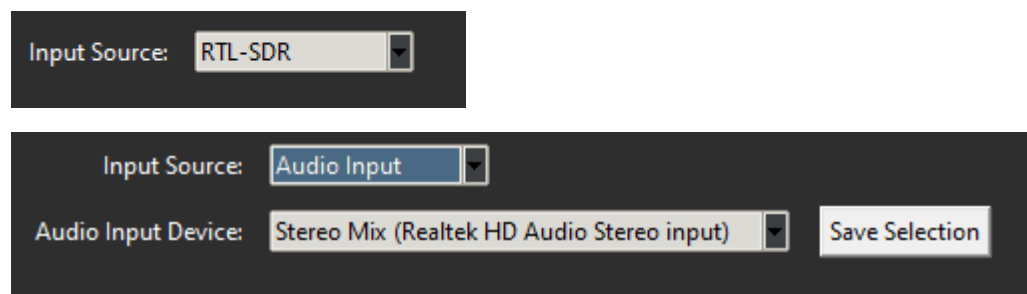
After selecting or changing the audio input, you need to confirm the selection by clicking the **Save Selection** button.
The changes will be saved to the **settings.ini** file and will remain active when the decoder starts — until you make a new selection.

Now we come to the section used to **enable telemetry upload to one of the APRS servers** (such as meteosonde.duckdns.org, radiosondy.info, wettersonde, or another server).

If the **Enable APRS Upload** box is checked and the user settings are configured, telemetry will be sent to the server at the specified interval.

⚠ It is important that the interval is not set too low, as there are other users and the server should not be overloaded.
A **recommended upload interval** is **20 seconds**.

At the bottom, only the **currently stored parameters** from aprs_settings.ini are displayed — these cannot be edited directly here.
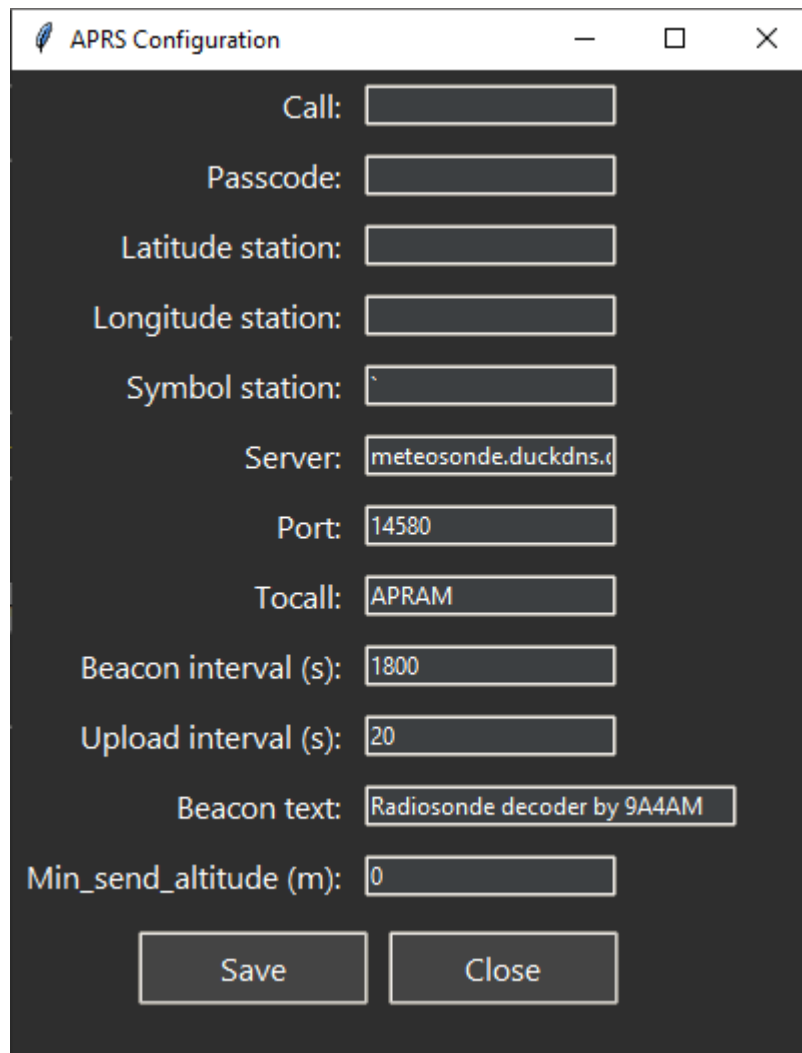To change them, you must click the **Configure APRS** button.

⚠ If connection details for the APRS server are missing, this feature **cannot be activated**.
The **Upload option must remain DISABLED** until configuration is complete!

Clicking the button opens a window.

This window is used to configure the **APRS settings**.
The **Callsign** defines how the receiving station will be identified and displayed.
Typically, this is a radio amateur's licensed callsign, as APRS is primarily intended for licensed amateur radio operators.

However, since receiving radiosondes does **not** require a radio license, you can use a different name — ideally **no longer than 9 characters**.

The **Passcode** is generated online based on the chosen name or callsign, for example at:
☞ https://apps.magicbug.co.uk/passcode/
(or any similar APRS passcode generator).

⚠ IMPORTANT:
Do not use public or standard APRS servers such as finland.aprs2.net or similar unless you are using a valid amateur radio callsign.
Telemetry from radiosondes must not be sent to these servers, as such data does not belong there.

The only exception is if you're using a reprogrammed sonde operating under a valid licensed amateur radio callsign — in that case, data may be sent.

✅ Use only servers specifically intended for radiosonde reception and tracking, such as:

- **meteosonde.duckdns.org**

- **radiosonde.info**

- **or other servers designated for this purpose.**

Enter the **latitude and longitude** of the receiving station in **decimal format** (e.g. 45.55555 and 16.55555).

The **station symbol** is the APRS icon for a **dish antenna**, which is necessary for servers to correctly identify radiosonde receivers.

This option is **not editable** — it is hard-coded into the application, as are the fields **Tocall** and **Beacon text**.

Select the **server** as previously described.
The default **APRS port** is **14580**, but some privately developed or custom servers may use a different port — so port modification is allowed.

- The **Beacon Interval** is **1800 seconds** (30 minutes). This can be changed, but there's usually no need.

  - The beacon transmits information about the receiving station to the server so it appears on the map and in the station list.

- The **Upload Interval** defines how often telemetry from the decoded sonde is sent to the server.

  - This can be adjusted, but the **recommended value is 20 seconds or more**.

The final option is Min Send Altitude — this allows you to set a lower altitude threshold below which telemetry will not be sent to the server.

If set to 0, telemetry will be uploaded until the end of the flight.

Since version 6.1, a second, optional APRS server has been introduced, under the fields Server2 and Port2.
This option allows sending data (beacon and telemetry) to two APRS servers simultaneously.
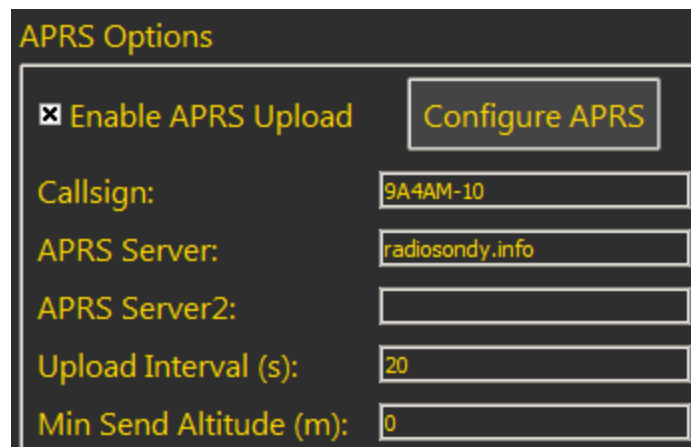It is activated together with the first (main) APRS server by enabling "Enable APRS Upload."

IMPORTANT:

If you are using only one server, it must be defined under Server and Port, while Server2 must be left completely empty (the field must contain nothing, otherwise it causes an error because the system assumes the server is valid).
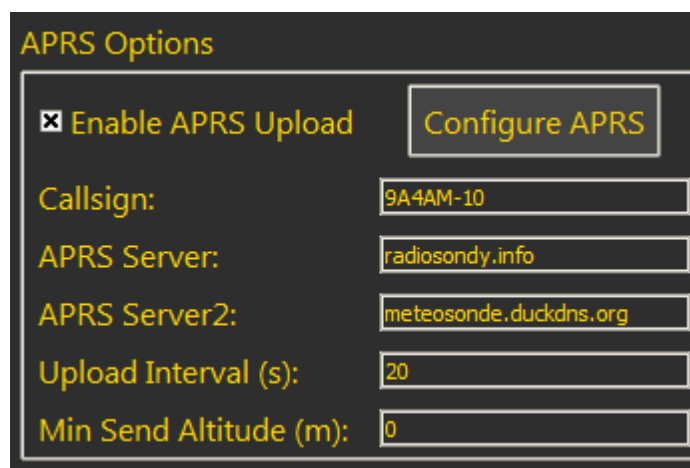**The Port2 field may remain filled.**

Sending to Radiosondy only:

Send to both Radiosondy and Meteosonde

Clicking **SAVE** writes the settings to the aprs_settings.ini file, so they are preserved the next time the decoder is started.

IMPORTANT: After any change in the APRS configuration, it is mandatory to restart the program in order to load the new settings from the configuration.
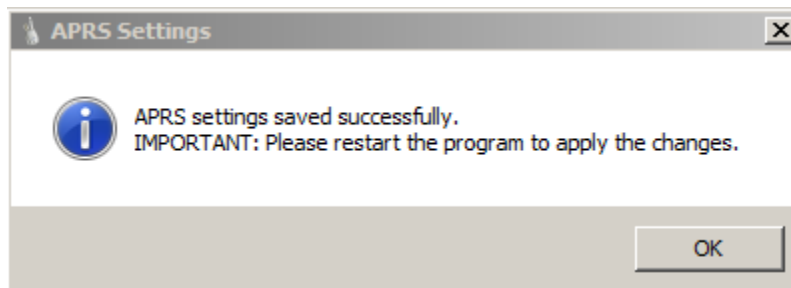
Clicking **CLOSE** closes the window and stores the settings into local variables for continued use without needing to restart the decoder.

At the bottom, there are two buttons: **Start Decoder** and **Stop Decoder**, used to start and stop the decoding process.

Additionally, the Start Decoder button also performs the function (as previously described) of saving the selected SDR settings and decoder type.

Once the decoder is started and no errors are detected, the status changes from:

Or when it is not running

And in window

Or when it is not running

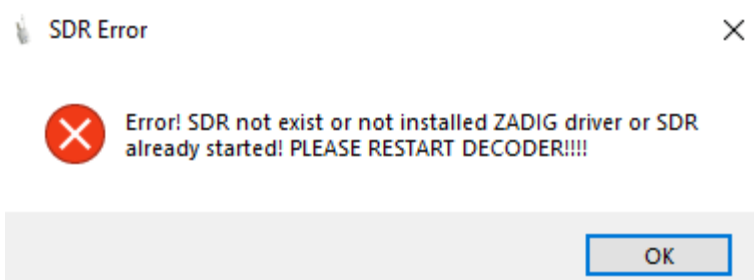If an error occurs, you will see something like this.


```
Starting with device: Microphone (Realtek HD Audio Mic input)
Decoder started.
[ERR] error: wav header
```

This means that the decoder was started successfully, but there is a problem with the audio input.

As mentioned earlier, a Virtual Audio source should be used — in this case, the error occurred because the Microphone input was selected.

At the bottom of the window, you can see decoded messages, DEBUG messages, and various other pieces of information that may be useful for identifying potential issues.

If the SDR (RTL-SDR) is not connected, the ZADIG driver is not installed, or another program is already using the SDR (for example, if the Radiosonde Decoder is accidentally launched more than once), the following message appears:

SDR Error   ✕

❌ Error! SDR not exist or not installed ZADIG driver or SDR
     already started! PLEASE RESTART DECODER!!!!

                                              OK

If this message appears, you need to identify and resolve the issue with the RTL-SDR, and then it is MANDATORY to restart the program.

The HELP button opens this document.

A PDF viewer must be installed on the PC in order for the application to display it.

HELP
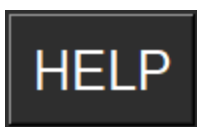
The second window displays the data from the last received sonde packet if the decoder is running in LIVE mode.If a log file has been selected instead, it shows the last entry from that log.

The Payload ID is the serial number of the sonde:

For RS-41, the original serial is used.

For M10 and M20, an APRS-compatible ID is used (due to the 9-character limit on APRS servers).

For DFM sondes, the serial number is prefixed with the letter D.

This is the standard format for uploading to APRS servers used for radiosonde tracking.

Other parameters shown in the window include:

Current location

Direction

Speeds (horizontal and vertical)

Date and time of the last decoded packet

Clicking the **Open Archive** button opens the contents of the LOG directory. If there are saved log files, they are displayed. You can select a file (logs follow the Payload_ID.log format, e.g. X12345678.log, ME1234567.log, or D12345678.log) and click **Open** to load the data from the last packet into the Last Received Data window.
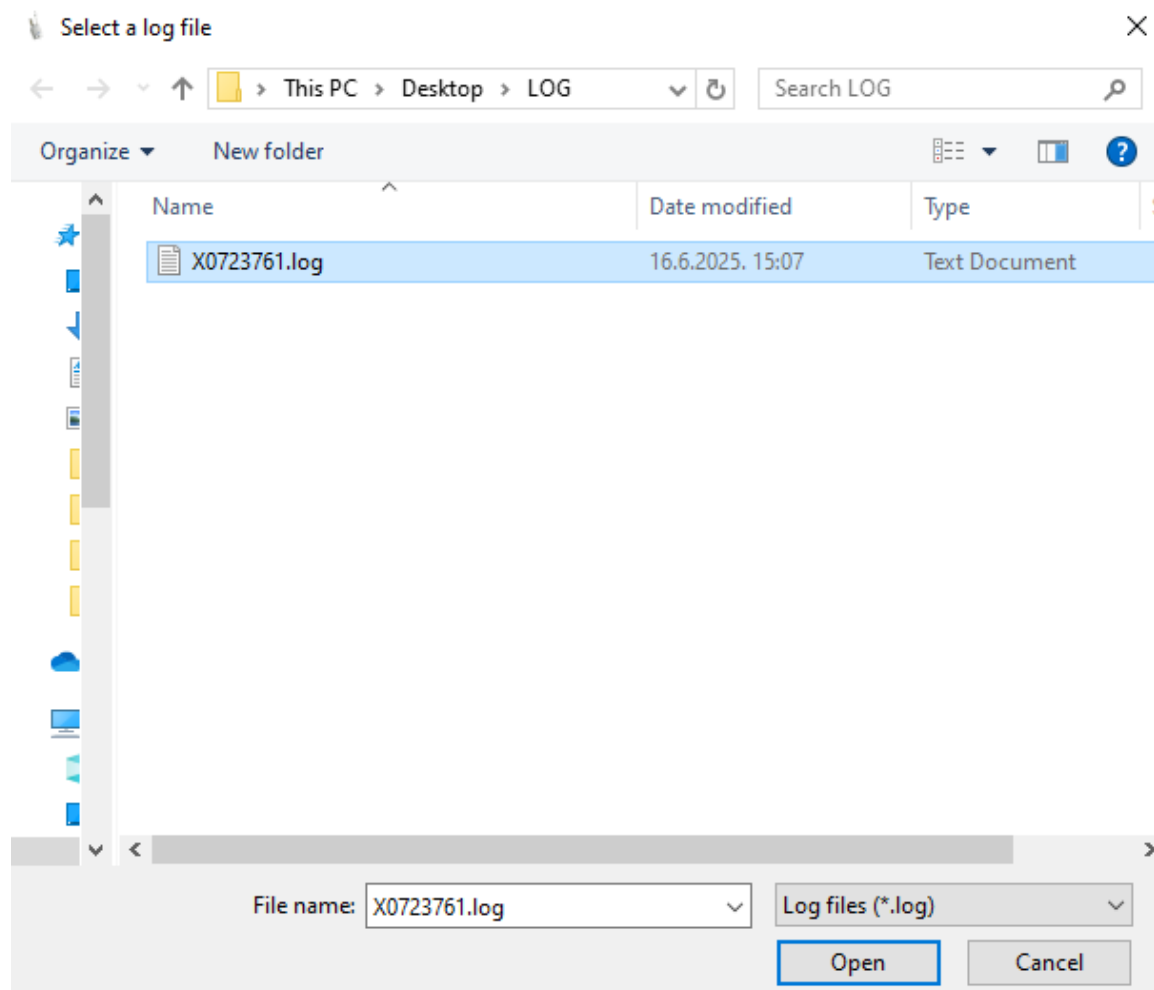
And show in window:
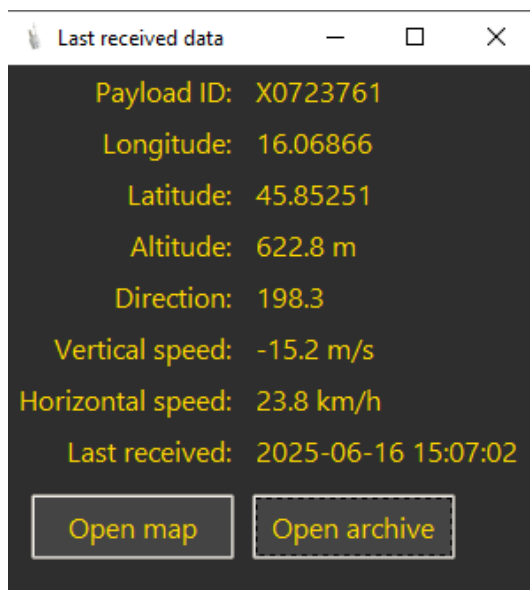
When there is data present in the Last Received Data window — whether from LIVE decoding or loaded from an archive (LOG file) — clicking Open Map will launch the map view. The map (based on Leaflet) is displayed as a web view in the system's default internet browser.

⚠ It is important that no other program is using port 6001, as the application launches a background server on this port. The map is then accessible at: http://localhost:6001

If LIVE decoding is active, the sonde is shown on the map at its current position:

As a balloon icon if it is ascending

As a parachute icon if it is descending

Clicking the marker opens a popup window with the latest telemetry data.

The map refreshes every 2 seconds.

The map view automatically pans to keep the sonde within the visible area of the browser window — even as it moves.

💡 Note:

When using the map display, the decoder creates an HTML file in the LOG directory (e.g. X0723761_map.html).

These files can be deleted manually if desired — but this is not necessary.

In a future version, automatic cleanup might be added, but for now, I've chosen not to implement file deletion to avoid unnecessary write operations to the user's disk.

Also, keep in mind that Windows OS does not have a built-in RAM-based temp directory like Linux, which is why these files are stored in the LOG directory.

Show on map.
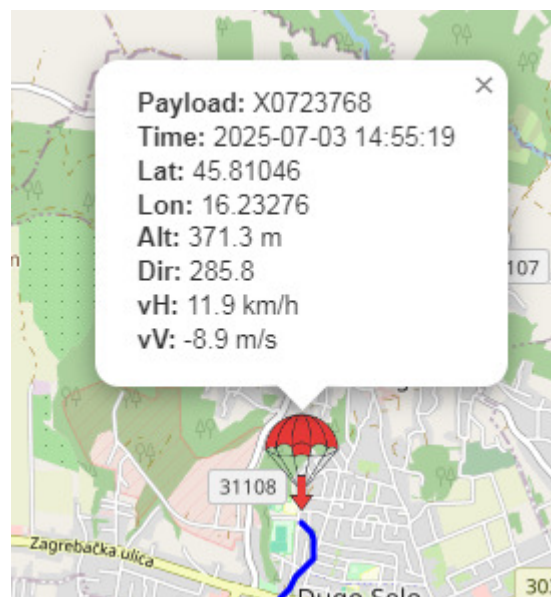
Sources Used in the Implementation of This Project:

https://github.com/rs1729/RS – RS decoder for radiosonde

https://github.com/rtlsdrblog/rtl-sdr-blog – RTL_FM program for SDR

https://github.com/chirlu/sox – audio signal processing program

https://github.com/projecthorus/radiosonde_auto_rx - help with implementation, this was the inspiration for the project along with RS1729, which is also used by AutoRX itself.

https://github.com/dl9rdz/rdz_ttgo_sonde – help with implementation

http://oe5dxl.hamspirit.at:8025/aprs/c/ - dxlAPRS, help with implementation

https://www.aprs.org/doc/APRS101.PDF – APRS protocol

https://apps.magicbug.co.uk/passcode - APRS passcode generator Online

https://radiosondy.info/ - APRS server and database radiosondy, help with testing and implementation

http://meteosonde.duckdns.org:1184 – APRS server and database by 9A4AM, help with testing

https://www.google.com/ - research and assistance in implementation

https://chatgpt.com/ - research and assistance in implementation

Special thanks to the team from the WX Sonde Viber group for their help with testing, support, and valuable advice — without them, this project would have never been completed, nor would the server exist.

LICENECE

External Tools Licensing Notice This project utilizes external command-line tools that are licensed under the GNU General Public License (GPL), including but not limited to:

SoX – Licensed under GPLv2

rtl_fm (from the rtl-sdr package) – Licensed under GPLv2

rs (radiosonde decoders) – Licensed under GPLv3

These tools are used as standalone programs executed via system calls or subprocesses. No GPL-licensed code is included directly in this project's source code, nor are any GPL libraries linked or modified.

As such, this project is not a derivative work of those GPL programs and is not subject to GPL licensing terms. However, users are responsible for complying with the licenses of the external tools they install and use.

Included External Tools (GPL Notice) This application uses several external open-source command-line tools, which are bundled in the executable distribution and invoked only via subprocess (i.e., as separate processes). These tools are licensed under the GNU General Public License (GPL), and are not integrated directly into the application's codebase.

The included tools are:

Radiosonde decoders (rs1729)

Files:`rs41mod.exe`, `m10mod.exe`, `dfm09mod.exe`, `mXXmod.exe`

License*: GNU General Public License v3.0

Source: https://github.com/rs1729/RS

Usage: These tools are used as external decoders and are invoked via `subprocess`.

Location in application: `AppDir\rs\decoders\`

Replacement: Users may replace these `.exe` files with compatible versions from the original repository, as required by the GPLv3 license.

RTL-SDR Tool rtl_fm.exe

License: GNU GPL v2

Source: https://github.com/rtlsdrblog/rtl-sdr-blog

Audio Processor sox.exe (SoX - Sound eXchange)

License: GNU GPL v2

Source: https://sourceforge.net/projects/sox/

These tools are invoked as external processes and are not linked or imported into the main application code. Their inclusion is solely for convenience and proper operation of the system. Each tool remains under its original license, and full source code or links to the official repositories are provided above in accordance with their respective GPL licenses.

Licensing Clarification The main application does not include or derive from GPL-licensed code. Therefore, the GPL requirements apply only to the tools listed above, not to the rest of this application.

This application uses map data from OpenStreetMap, licensed under the Open Database License.