

Dreiländereck-Sysop-Treffen 2023
in Engen
- LoRa APRS -

(c) 2023-02-10 Thomas Osterried DL9SAU <dl9sau@darc.de>
Version 1.03
License: CC-BY-SA

1 Was ist LoRa APRS

1.1 Überblick

LoRa wurde für ISM Anwendungen entwickelt (868 MHz, 433 MHz, 920 MHz (USA)).

APRS kennen wir von 2m. APRS bedeutet Automatic Packet Reporting System.
LoRa APRS ist APRS, auf dem LoRa MAC Layer.

IARU Bandplan:

QRG: 433.775 MHz (Haupt-Frequenz) und 433.900 MHz.

LoRa Mode-Einstellungen, die beiden Frequenzen, sowie die Verwendung des TNC-Trace-Format, die den „Standard“ in der IARU-Region 1 setzten, entstanden in OE (die in Sachen LoRa-APRS Vorreiter waren).

1.1 Motivation

- Neue Technologie
- Spaß mit QRPP
- Geringe Stromaufnahme
- Neue Erkenntnisse über Ausbreitungsbedingungen auf 70cm
- Spannende Softwareprojekte mit dem ESP32 Microcontroller
- Regt zu Bastelprojekten an, man lernt viel (Platineneigenentwicklung, Mac-Layer Codierung, Programmierung), u.v.a.m.

2 Funklayer

Schmalbandiges Signal, verteilt über breites Spektrum:

CSS (Chirp Spread Spectrum).

Sehr starke FEC (Forward Error Correction).

Für embedded Systems konzipiert; sehr energiesparend.

Chiphersteller Semtec.

Patentbehaftet, nicht frei (vgl. Voice AMBE Chip bei DMR/D-STAR/YSF);
gut dokumentiert, aber in einigen Punkten nur ins notwendigste Detail.

LoRaWAN ist ein Wide-Area Netzwerkprotokoll.

Wir verwenden hingegen **unser eigenes**, zu APRS kompatibles Format zum Senden von Paketen.

Unsere LoRa APRS Pakete sind an alle gerichtet und werden vom Empfänger nicht bestätigt.

3 Nutzlast: APRS

TNC-Trace-Format:

Ein TNC2 für Packet-Radio stellt AX.25-Pakete beim trace-Modus in Textform dar.

Die APRS Spezifikation verwendet ebenfalls diese Notation.

Ins APRSIS Netzwerk im Internet wird der Datenverkehr über tcp oder udp geleitet und hat ebenfalls dieses Format.

Das Format eines Pakets auf Funk ist bei LORA das TNC-Trace-Format.

[Hingegen bei AX.25 (also 2m APRS) ist es das AX.25 Paket-Format.]

Diese Entscheidung ist m.E. schlecht (ineffizient, ASCII Parser-Horror mit vielen Fehlermöglichkeiten), aber jetzt nachträglich nicht änderbar.

Eines APRS Paket (im Beispiel "Status Paket") im TNC-Trace-Format sieht aus wie folgt:

```
DL9SAU>APRS:>test
```

```
^SRC    ^DST
```

```
    ^': Trenner zwischen Header und Inhalt
```

```
    ^ ,>' Trenner zwischen Source und Destination-Call
```

```
    ^ ,>' APRS Protokoll: Status-Paket.
```

```
    ^Direkt gefolgt vom eigentlichen Inhalt
```

Zielcall: ist frei verwendbar. Üblich sind „APRS“, oder codierte Software-Versionen wie „APDW15“ (hier: Direwolf Version 1 5). aprs.fi zeigt dies dann in der Stations-Info an. MIC-E compressed positions codieren Teile von Position (Lat/Lon) im Zielfeld. Ein Digi darf das deshalb niemals umschreiben (von einer optionalen SSID abgesehen; s.u.).

Da Freitext und nicht AX.25-Protokoll-Header, finden sich manche Blüten wie DL1AAA-1234 oder DL1AAA-L4.

Ich empfehle dringend, sich an die übliche Notation (Call Länge bis 6, SSID -1 bis -15) zu halten. Das schafft größtmögliche Kompatibilität, insbesondere, wenn wir auf HF 2m APRS und 70cm LoRa-APRS verbinden möchten, um z.B. den Austausch von Kurznachrichten zu ermöglichen.

Internas:

Der LoRa MAC Header kennt auch Quell- und Ziel-MAC (in sehr kleinem Wertebereich 0-255). Unsere LoRa Zieladresse für LoRa APRS ist '<', mit der ID 0x1. LoRa ist paketorientiert, d.h. Stückelungen in zwei Pakete wie „...SAU>“ „APRS:...“ ist nicht zulässig.

=> Wir senden an den Chip o.g. Lora-Header bestehend aus < und 0x01, direkt vom payload „DL9SAU>APRS:>test“

4 LoRa Einstellungen

Bandwidth (BW)	125 kHz, 250 kHz, 500 kHz
Coding Rate (CR)	Stärke der FEC. 4/5, 4/6, 4/7, 4/8.
Spreading Factor (SF)	Werte (6) 7 - 12
CRC:	Prüfsumme wird angehängt.

Je höher die Bandbreite, desto weniger spielen Störungen mit ein paar kHz eine Rolle; ein Empfänger muss ein breiteres Spektrum beobachten → mehr Rauschen → geringerer SNR.

SF: je höher der SF, desto mehr steigt letztlich der SNR.

Lese-Tipp zu SF/BW/CR:

<https://blog.ttulka.com/lora-spreading-factor-explained/>

Forward-Error-Correction:

Bei CR 4/8 werden 4 bit in 8 bit codiert (und über mehrere bit gestreckt). Die Übertragung dauert damit doppelt so lang.

Defekte Bits können aber bei wesentlich schlechterem SNR noch rekonstruiert werden, und es ergibt sich eine Toleranz gegenüber burst-Störungen auf der Frequenz.

Der Sender gibt im Header an, mit welcher CR er die Nutzlast codiert.

Der Paket Header wird immer im robustesten und langsamsten CR 4/8 gesendet.

Nicht-behebbarer Fehler: werden an falscher CRC (Prüfsumme) erkannt. CRC muss beim Sender eingeschaltet sein und vom Empfänger berücksichtigt werden, sonst gibt es Datensalat.

Ein Empfänger kann stets alle CR-Varianten dekodieren; die CR-Einstellung des Geräts betrifft nur den Sendevorgang.

Der Sender wählt eine höhere FEC, (also z.B. CR 4/8) um besser dekodiert werden zu können; sein Sendevorgang wird länger dauern (-> weniger Airtime für andere verfügbar) und mehr Energie verbrauchen.

[Für meine Firmware habe ich eine *automatic CodeRate-Adaption* entwickelt: wird auf dem Kanal in für einen längeren Zeitraum niemand anderes gehört, wird in 4/8 gesendet. Das braucht bei portablem Tracker mehr Energie, da die Aussendung doppelt so lang ist. Gibt es mehr Nutzer, wird die CR reduziert, damit mehr Airtime für alle zur Verfügung steht]

Wer aber unterschiedliche SF oder BW dekodieren will, oder gar mehrere "Kanäle" gleichzeitig überwachen will, müßte auf teure kommerzielle Lösungen zurückgreifen.

In EU (bis auf SP) verwenden wir **BW125, SF12**. → Sensitivity -137 dBm.

Datenrate: Je nachdem ob CR 4/5 oder CR 4/8 eingestellt wird:

CR 4/5: 293 bps.

CR 4/8: 183 bps.

Vgl. Packet-Radio 1200 baud.

Die effektive Datenrate setzt sich aus o.G. Parametern zusammen.
Folgende Formel zeigt das Zusammenspiel von Datenrate, BW, CR, SF:

$$DR = SF \cdot BW \cdot \frac{CR}{2^{SF}}$$

DR = Data rate; SF = Spreading factor; BW = Bandwidth
CR = Coding rate

Es ist offensichtlich, dass sich das nicht einfach auf unsere althergebrachten Bitratendefinitionen wie „Baud“ oder „kbit“ übertragen lässt. Deshalb vermeidet man diesen Begriff.

Das Ergebnis (die Datenrate) kann aber mit Baud- / kbit-Werten verglichen werden.

Zum Experimentieren: online-Bitraten-Berechnungen der verschiedenen Parameter-Variationen

<https://unsigned.io/understanding-lora-parameters/>

4.1 Bewertung der Parameter

1200 Baud AFSK benötigt grundsätzlich Signalpegel über dem Rauschen. LoRa dekodiert unter dem Rauschteppich.

In Polen wird BW125 SF9 CR4/7 (etwa 1200 bps) verwendet und ist mit 2m APRS vergleichbar. SF9 benötigt aber etwa 8dB mehr SNR als die von uns verwendete SF12.

Ich meine, der langsame BW125 SF12 CR4/(5bis8) in Verbindung mit QRPP ist sehr spannend zum Beobachten von Ausbreitungsphänomenen auf 70cm.

Der Langsamste LoRa Mode kann Daten 20 dBm unter dem Rauschen dekodieren. Das sind Werte wie wir sie von WSPR auf KW kennen!

Deshalb brauchen wir keine PA. 60km Distanz mit z.B. 100mW sind möglich.

Auch faszinierend: iGate in 4km Entfernung. Tracker liegt im Haus im Erdgeschoß auf der entgegengesetzten Seite (Teile des Hauses dazwischen) auf dem Tisch, mit Antenne von einem Handfunkgerät. Signal wird stets zuverlässig, vollständig und ohne Paketverlust vom iGate empfangen.

Die max. Paketgröße ist kleiner als bei PR (240 vs 256).

APRS nutzt aber gem. Spezifikation kleinere Paketlängen als 256; so sollte es zu keinen Inkompatibilitäten kommen, selbst beim Gaten zwischen 2m APRS und Lora-APRS.

Allerdings ist bei AX.25 die MTU auf die Nutzlast bezogen; bei LoRa APRS ist der TNC-trace-Format-Header hinzuzurechnen.

4.2 Einstellungen zum Sendeverhalten und Inhalten

Ein großes APRS Paket benötigt im langsamsten Mode eine Übertragungszeit von ca. 4-5 Sekunden, je nach gewählter CR.

Genauer zur Airtime-Berechnung siehe

<https://www.rfwireless-world.com/calculators/LoRaWAN-Airtime-calculator.html>

→ Kollisionsfrei können in einer Minute max 12 Stationen senden (60s / 5s).
Deshalb bitte nicht öfter als 1x die Minute senden (außer es ist ein „wichtiger Informationsgewinn“, z.B. "Kursänderung"), eher alle 1,5min als untere Marke von Smart-Beaconing.

Feste Baken besser im 30-, 20, oder 15-Minuten-Abstand aussenden.

Lange beschreibende Texte vermeiden. Idealerweise wird der Bakentext nicht bei jeder Aussendung mitgeschickt. Das verringert die Airtime deutlich.
Eigener Name und DOK sind in der Regel auch noch in 60 Minuten die selben.
Und Dienste wie aprs.fi merken sich ohnehin den letzten nicht-leeren beschreibenden Text.

Position im „Compressed“ Format, gemäß APRS-Standard wird das Paket nennenswert kürzer: 10 Byte, statt 17 Byte.

Doch nicht nur das: es enthält gleichzeitig auch Geschwindigkeit und Kurs bzw. Höhe, für die bei einem unkomprimierten Positions-Paket zusätzlich noch 7 Byte (CSE/SPD) bzw. 9 Byte für die Höhe (/A=001234) nötig sind.

Telemetrie-Frames empfehle ich ausschließlich auf der Zweit-Frequenz 433.900 MHz auszusenden.

Negativbeispiel HAMRADIO 2022 Campingplatz:

ein OM sendete Telemetrie-Paket plus die beschreibenden EQNS, PARM, UNIT, BITS hintereinander, und nach 10s Pause kam das nächste Telemetrie Paket, wieder zusammen mit den vier, den aktuellen Messwert, beschreibenden.

Freilich hatte sich die gemessene Temperatur in den letzten 30s nicht geändert.

→ **SCHLECHT!**

5. iGates / eher kein Digipeating

Bei Übertragungszeiten von 5s pro Paket ist Digipeating, also das Wiederaussenden von gehörten Paketen, störend.

Deshalb stützte sich von Anfang an unser LoRa-APRS-Netzwerk auf iGates (Gateways RF->Internet ins APRSIS-Netz).

Vom HAMNET aus kann in der HAMCLOUD aprs.r1.hc.ampr.org verwendet werden; das System ist redundant ausgelegt.

→ Wer HAMNET, aber kein Internet hat, kann trotzdem APRS-Daten von 2m APRS oder 70cm LoRa APRS ins APRSIS-Netz leiten.

Bob Bruninga hatte noch für iGates das Icon Viereck mit "L" auf schwarzem Grund zugewiesen, so dass diese auf der Karte leicht als solche zu erkennen sind.

Einige Firmwares können nicht digipeaten. Andere Firmwares (auch meine) können digipeaten, schränken das aber ein auf
WIDE1-1 oder WIDE2-1 oder WIDE1-1,WIDE2-1 (Pfadreduktion)

Der APRS Standard kennt eine interessante, wenn auch nicht-mehr-empfohlene, Notation: Ziel-Rufzeichen mit SSID -1 entspricht einmal digipeaten, SSID -2 entspricht zwei mal digipeaten.

Da die Notation eingeführt wurde, bevor auf das WIDE1/WIDE2-Konzept eronnen wurde, ist nicht eindeutig, was mit Zielcall APRS-1 gemeint ist:

Beispiele:

APRS-1: Fill-In-Digi WIDE1-1 nutzen, oder nur WIDE2-Digi WIDE2-1 ?

APRS-2: Fill-In-Digi WIDE1-1 nutzen und einen WIDE2-Digi, oder zwei WIDE2-Digis? Oder gar: keinen Fill-In-Digi nutzen?!

Und was tun nach dem Digipeaten? Zielcall APRS-1 bzw APRS-2 umschreiben auf

APRS-1: SRCCALL>APRS,GATECALL,WIDE1* , oder
SRCCALL>APRS,GATECALL,WIDE2* ?

oder

APRS-2: SRCCALL>APRS,WIDE1,GATECALL*,WIDE2-1 , oder
SRCCALL>APRS,GATECALL*,WIDE2-1 ?

Die Spezifikation ist also relativ unscharf.

Vorteil der Notation: spart kostbare Bytes, also Airtime. „-1“ ist kürzer als „WIDE1-1“.

5.1 Die zweite Frequenz: für automatische Stationen

Man muss aber nicht auf der selben Frequenz digipeaten.

Automatische Stationen, die selbst auf RF senden, werden nur auf der Zweitfrequenz 433.900 MHz genehmigt.

Meine Empfehlung ist, den polnischen Mode BW125 SF9 CR4/7 (der 1200 bps entspricht) auf der Zweitfrequenz zu fahren.

5.1.2 Anwendungsbeispiel

So läßt sich auch ein, die-Hauptfrequenz-schonender, WIDE1-Fill-In-Digi auf einen Berg stellen, der auf Grund seines guten Standortes das iGate auch im schnellen Mode gut erreicht (und dank kürzerer Airtime zudem noch stromsparender ist), und deshalb eine größere SF nicht erforderlich ist.

In diesem Szenario muss das (den Fill-In-Digi empfangende) iGate mit zwei LoRa-Geräten und Antennen ausgestattet sein: das eine hört auf der Haupt-Frequenz, das andere hört und sendet auf der Sekundärfrequenz.

Nutzer hören entweder überhaupt nicht (spart Strom), und wenn, dann (derzeit noch) auf der Hauptfrequenz.

Auf der Hauptfrequenz sieht man die direkten Pakete anderer Nutzer.

Wenn automatische Stationen auf der Sekundärfrequenz 433.900 MHz senden, und dort z.B. auch sog. *3rd-Party-Traffic* aus dem Internet aussenden, könnte es für einen Nutzer auch interessant sein, auf dort zu hören.

Meine Firmware kann zeitschlitzbasiert auf beiden Frequenzen hören und verpaßt so "nur" statistisch etwa 50% der Pakete ;)

Da aprs-Messaging eine Bestätigung vom Empfänger einfordert und bis zum Eintreffen (oder timeout) die Nachricht wiederholt, kann man mit dem sich ergebenden delay m.E. gut leben.

6. Hardware

Die kleinen Geräte gibt es mit und ohne Display, mit oder ohne GPS; alles aus CN. Früher sehr preiswert; mittlerweile wird das Doppelte bis Dreifache abgerufen. Mein Lilygo TTGO kostete ca 24 Euro. Heute liegt er bei 60.

- Embedded standalone:

mit CPU ESP32

Hersteller firmieren unter Namen wie *Lilygo* oder *DollaTec*.

Es gibt einen kleinen, mit oder ohne Display. Er hat keinen GPS-Empfänger.

Der "große" TTGO hat einen GPS-Empfänger und ist in Versionen mit oder ohne Display erhältlich.

Bluetooth und WLAN haben alle mit an Bord.

Dank Bluetooth lässt sich auch die Android-Software *aprsdroid* koppeln. So sieht man Aussendungen anderer Nutzer (auch auf der Karte!), und kann Nachrichten austauschen.

Ethernet haben diese Chips nicht; der ESP32 kennt aber Ethernet-Layer (der ja auch für WLAN und Bluetooth verwendet wird).

→ Es gibt zusätzlich Ethernet-POE-Boards.

Das ist interessant für den Betrieb als iGate am Relaisstandort, wenn WLAN nicht gewünscht, oder kein anderer Rechner mit USB-Seriell vorhanden ist bzw. kein zusätzlicher aufgebaut werden soll.

Lilygo T-Echo (mit Baro-/Temperatur-/Feuchtigkeits-Sensor und e-Paper Display). Die Firmware von OE5BPA oder mein SQ9MDD-Derivat laufen hierauf aktuell noch nicht.

- RA01 und RA02 heißen LoRa-Chips, die sich z.B. an die SPI Schnittstellen eines Raspberry PI anschließen (das vielleicht ohnehin schon vorhanden ist für 2m APRS oder EchoLink / svxlink). Gute Integration am Relais-Standort:
Bei schon vorhandenem Raspberry-PI ist somit keine zusätzliche Hardware am Relais-Standort nötig.

Link: <https://www.dl1nux.de/lora-aprs-igate-mit-dxlaprs-und-ra02-auf-einem-raspberrypi/>

- APRSCube (gibt es auch in einer 1 W Version; relativ großes Display)

Es gibt LoRa-Module (oder PAs) mit 1 W PEP → 10 dB mehr gegenüber den üblichen 100 mW.

Dies nur der Vollständigkeit halber. Das ist m.E. nicht nötig.

Software:

RA01/RA02 LoRa Chips:

Wie beschrieben, sind die RA01/RA02 Chips über die SPI Schnittstelle eines Raspberry PI ansprechbar. Standard-APRS-Software lässt sich direkt verwenden und mit APRS-Digi-Features (z.B. Filter) feintunen, ohne die Firmware eines embedded Systems ändern zu müssen. So kann 70cm LoRa APRS beispielsweise direkt mit 2m APRS und vernetzt werden.

LoRa auf embeded ESP32-Controller:

Hier gibt es einige Softwareprojekte.

Nennen möchte ich

1. die weit verbreitete OE5BPA's Software

es gibt zwei unterschiedliche Versionen:

- eine für den Einsatz als transportable Tracker
- eine für den Betrieb eines iGate.

<https://github.com/lora-aprs/>

Diese läuft auf den o.g. ESP32 Boards

2. SQ9MDD bzw. meine Erweiterung

SQ9MDD entwickelte eine Version mit zahlreichen Features, wie Web-Interface und KISS-Unterstützung.

Sein Projekt: <https://github.com/SQ9MDD/TTGO-T-Beam-LoRa-APRS/>

Ich habe die Software nochmals um viele Features erweitert; da er selbst kaum Zeit hatte, konnte bisher nichts in sein Hauptprojekt zurück fließen. Schade, denn ich bin kein Freund von Forken und 1001 Derivaten.

Meine Version steht hier:

<https://github.com/DL9SAU/TTGO-T-Beam-LoRa-APRS/>

Meine Version kann u.a. Digipeating (WIDE1, WIDE2; Pfadverkürzung; Cross-Frequenz-Digipeating (auf Sekundärfrequenz)).

Es ist eine einzige Software, die alle Einsatzgebiete abdeckt: für unterwegs genauso wie als iGate am Relais-Standort.

Über WLAN oder USB-Seriell kann entweder über KISS oder im TNC-Trace-Format das Gerät mit bestehender Digipeater-Software am Standort verbunden werden.

Mittlerweile entwickeln Thomas DL3EL und ich diese Software gemeinsam weiter.

In den letzten Wochen kamen etliche neue Features hinzu. So wurde aus ein geplantes Weihnachts-Release verschoben – und ein „Engen“-Release steht kurz bevor ;)

Weitere Informationen:

Die deutschsprachige Telegram Gruppe namens „LoRa-APRS“, mit knapp 750 Mitgliedern.

Hier werden spontan viele Fragen beantwortet oder auf passende Infoquellen hingewiesen.