

Tracker Firmware 1.67 lora-aprs.at (OE1ACM & OE1CGC)

Achtung – unser WebHosting Provider hat leider stillschweigend die RSA Certificates von **Let's Encrypt Authority X3** auf **R3** umgestellt und die Validierung für den FW-Download funktioniert nun nicht mehr, was zu einem ‚Connection Refused‘ Error führt und den FW-Download abbricht.

Ich habe eine **neue FW-Version 1.67** mit neuem Certificate gebaut, die allerdings **über OTA** oder USB installiert werden muss. Weitere Firmware-Downloads können dann wieder automatisch vom Server durchgeführt werden.

Features im Überblick:

- LoRa-APRS Tracker
 - Parameter und Steuerung über Mini-Menü am Tracker oder
 - Mini-Web-Interface
- iGate Funktion using Peter's APRS-IS Lib (nach OE5BPA)
 - Konfiguration über Web-Interface
 - über mobilen Hotspot
 - Alternate Server Umschaltung bei Verbindungsproblem
 - Aktuelle Position kann von GPS übernommen werden
- Online Konfiguration über Web-Interface mit WiFi Anbindung
- Mini-Menü am Tracker
- Remote-Steuerung oder Abfragen (Remote Relais- Steuerung)
- I/O-Expander & BME 280 Sensor Betrieb am verlängerten I2C Bus
- Mini-Messaging über Tracker & Web-Interface
- Sleep-Modus zwischen Aussendungen im Tracker-Betrieb zur Minimierung des Stromverbrauchs
- Automatisches Firmware Update direkt vom Tracker (!!) (kein Build-Environment mehr nötig für Updates) ** (**ab 2. Dezember ab FW-v.1.67**)
- GPS Anzeige, Kompass, MaidenHead Locator, etc

- Telnet Server zur Anzeige der empfangenen und gesendeten Daten mit SNR/RSSI
- KISS-Modem Betrieb über USB/Serial (derzeit Testbetrieb)

Wir möchten vorausschicken, dass wir keine professionellen Entwickler sind, dieses Projekt aus Spaß am Hobby Amateurfunk neben dem QRL betreiben und vor einigen Jahren als pure Anfänger im MicroController Bereich gestartet haben.

Wir haben versucht, den Schwerpunkt auf möglichst einfache Bedienung zu legen und auf die komplexe Installation von diversen Build-Environments zur Installation von Updates oder dem Ändern von einfachen Parametern, zu verzichten.

Die Tracker sind mit einem Mini-Menü, bzw. Web-Interface zur Konfiguration ausgestattet.

Des Weiteren können Updates relativ einfach von unserem Server über WiFi geladen und automatisch installiert werden. Änderungen / Updates können somit einfach und schnell ohne Programmierkenntnisse installiert werden.

Das Projekt ist momentan noch nicht als ‚Open-Source‘ verfügbar, ist aber langfristig geplant, da wir auch ‚Open-Source‘ Teile verwenden. – u.A. Peter’s (OE5BPA) APRS-IS lib.

In Punkto Hardware haben wir ursprünglich auf fertige LoRa Boards ‚Made in China‘ gesetzt, haben aber dann begonnen, eigene Hardware zu entwickeln, nachdem wir mit ständigen HW-Änderungen in Pin-Belegung gekämpft haben. Mit der Eigenentwicklung der Boards sind wir davon jetzt unabhängig, da die Belegung auf allen Tracker-Typen gleich bleibt.

Bedient werden die Tracker über 3 Tasten (<- Enter/Menü ->). Die Links/Rechts Tasten schalten durch die diversen Bildschirme, mit der Enter/Menü-Taste gelangt man ins ‚Main Menu‘ wo über die Tasten diverse Funktionen oder Sub-Menüs aufgerufen werden können. Hintergrundprozesse laufen während des Scrollens durch die Anzeige-Bildschirme, werden aber nur beschränkt ausgeführt, sobald sich der Tracker im ‚Main Menu‘ oder Sub-Menues befindet. Als Zeichen dafür leuchtet die gelbe LED konstant und erlischt beim Verlassen der Menüs, als Zeichen dafür dass Hintergrundprozesse wieder ausgeführt werden.

Die Hauptaufgaben entweder Tracker-Mode oder iGate-Mode werden als Task ausgeführt. Nachdem der ESP32 zwei Cores für die Verarbeitung hat, steht ein Core für die Anzeige/Bedienung zur Verfügung. Es kann somit fast überall das Menü aufgerufen werden und Konfigs geändert werden, bzw. zwischen den Betriebsmodi umgeschaltet werden. Zum Umschalten zwischen den Modi wird meistens neu initialisiert.

Die Konfiguration wird auf Wunsch im NVRAM abgelegt und beim Restart wieder eingelesen.

Eine weitere Funktion ist der ‚Sleep-Mode‘, der den Stromverbrauch minimiert. Es werden dafür ein Großteil der Komponenten abgeschaltet und nur der RTC (3. Admin-Core am ESP32), mit minimalstem Stromverbrauch, versorgt. Der Stromverbrauch sinkt dafür auf von ca 200 mA im Vollbetrieb RX&GPS&WiFi auf **ca 4mA** am 1W Tracker der letzten Generation.

Aufgeweckt aus dem Sleep wird das System wieder über Tastendruck.

Setup

Zur Konfiguration des Trackers kann das Menü am Tracker oder der kleine Webserver verwendet werden.

Der Webserver ist über WLAN erreichbar. Im Initialzustand ist das WiFi Interface aktiv und der Tracker befindet sich im Accesspoint-Mode.

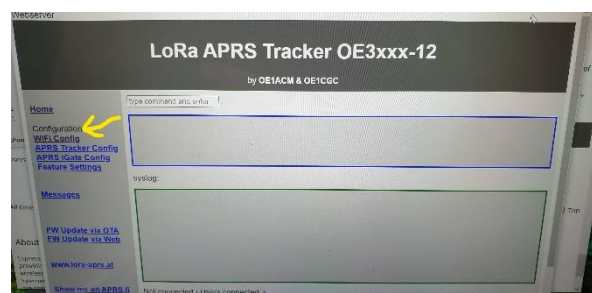
Das ‚W‘ im Anzeigefeld (grün) gibt an, dass WiFi aktiv ist. Der Tracker kennt noch keine WLANs und startet deshalb im Accesspoint-Mode mit der SSID: **APLoRA<mac_addr>** und IP: **192.168.4.1**.

Über diese SSID ist der Tracker über WiFi zu erreichen. – Am PC erscheint unter Netzwerk/WLAN Konfig jetzt das neue WLAN und wir können uns damit connecten.

Das Passwort für das Accesspoint-WLAN (SSID LoRa....) ist „**lora12345**“.

Bestätigen und wir sind im WLAN des Accesspoints und können nun den WebServer am Tracker unter der IP@ **192.168.4.1** erreichen.

Der Tracker meldet ihn unserm lokalen bekanntmachen...

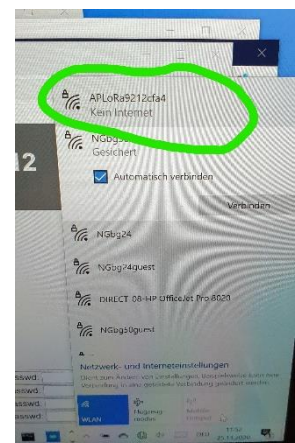
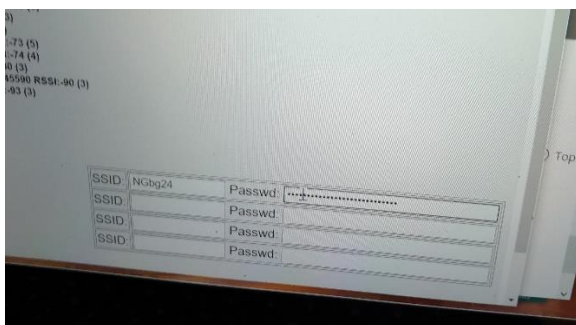


sich und wir können WLAN

WiFi-Config anklicken und wir können die SSID und Password unseres lokalen WLANs eingeben.



Der Tracker merkt sich bis zu vier WLAN/SSIDs und wird sich zu dem ersten sichtbar und in der Tabelle bekannten WLAN connecten. Es können somit mehrere WLANS (Home, mobiler Hotspot, Auto,etc...) konfiguriert werden.



Lokale WLAN SSID / Passwd eingeben ...

Bitte mit **'SAVE'** oder Enter zum Abspeichern bestätigen.

Nach einem Reboot meldet sich der Tracker dann, wenn SSID und Passwd richtig gesetzt wurden, im lokalen WLAN an.

Reboot, und der Tracker booted neu und connected sich jetzt im lokalen WLAN und bezieht seine neue Adresse über DHCP. – Diese wird angezeigt:



Der Tracker ist nun unter der angezeigten IP@ - als Beispiel hier 192.168.2.156. Wir können diese Adresse im Browser verwenden, um den Webserver des Trackers für weitere Konfig oder Update der Firmware zu verwenden.

Wir haben bereits die letzte FW installiert und können mit der Konfiguration unserer Parameter beginnen.

Webseite des Trackers.

APRS-Konfig

Webserver

LoRa APRS Tracker OE1ACM-19

by OE1ACM & OE1CGC

[Home](#)

Configuration
[Config WiFi](#)
[Config APRS Tracker](#)
[Config APRS IGate](#)
[Config HW & Params](#)

Messages
[APRS Messages](#)

Firmware
[FW Update via OTA](#)
[FW Update via Web](#)

[Restart Tracker](#)

Links
www.lora-aprs.at
[Show me on APRS.fi](#)

Radio APRS Config

Callsign:	OE1ACM-19
Comment:	LoRa 17 dBmW
Name:	Bernd
APRS Symbol:	? Table: \

TX Interval (s):	60
TX power (5-23):	
TX Freq (MHz):	433.775
RX Freq (MHz):	433.900
Override default QRG:	no

Firmware: FW 1.66 OE1ACM

iGate Konfig

LoRa APRS Tracker OE1ACM-19

by OE1ACM & OE1CGC

[Home](#)

Configuration
[Config WiFi](#)
[Config APRS Tracker](#)
[Config APRS iGate](#)
[Config HW & Params](#)

Messages
[APRS Messages](#)

Firmware
[FW Update via OTA](#)
[FW Update via Web](#)

[Restart Tracker](#)

Links
www.lora-aprs.at
[Show me on APRS.fi](#)

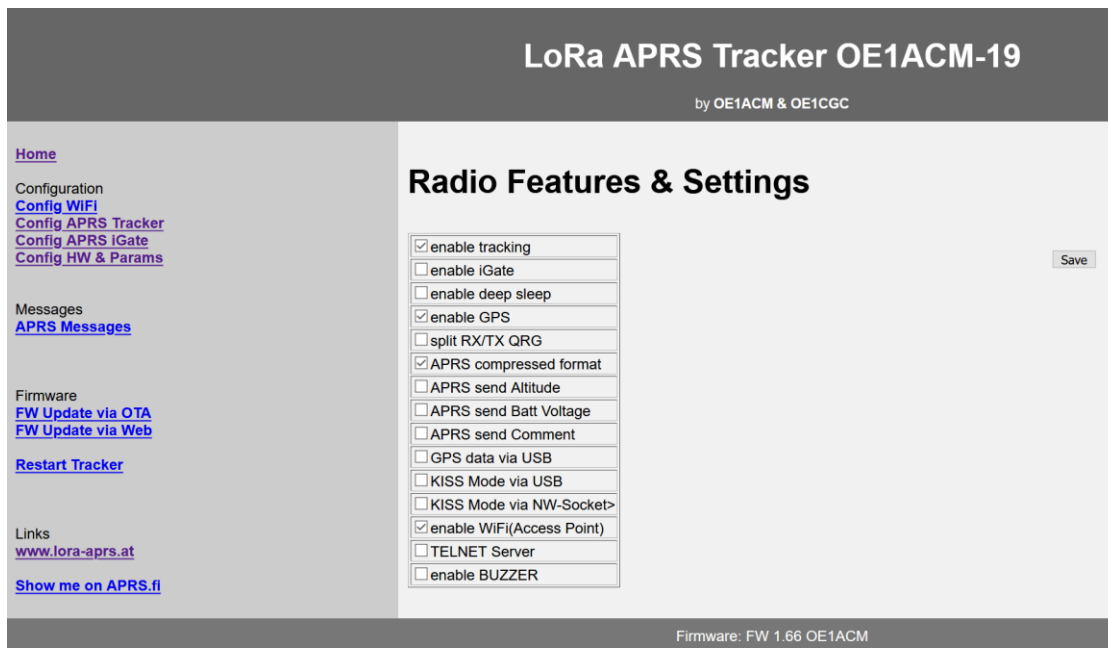
APRS iGate Config

using APRS-IS Lib by OE5BPA

iGate Callsign:	OE1ACM-19	Save
APRS-IS Passwd:	20139	
VERS:	0.1	
TOOL:	ESP32-APRS-IS	
APRS Symbol:	g Table: /	
APRS-IS-Server1:	austria.aprs2.net	
APRS-IS-Server2:	euro.aprs2.net	
Port:	14580	
Beacon Time:	15	
Latitude:	4812.95N	
Longitude:	01627.E26	
use Location from GPS:	<input checked="" type="checkbox"/>	
Beacon-Text:	www.lora-aprs.at Tracker running iGate APRS-IS Lib by OE5BPA	
TX power (5-23):	<input type="checkbox"/>	
TX Freq (MHz):	433.775	
RX Freq (MHz):	433.775	

Firmware: FW 1.66 OE1ACM

Parameter zur Steuerung



In den Features & Settings Menue werden über Flags bestimmte Programmabläufe zur Steuerung definiert.

Die Parameter werden auf Wunsch im NVRAM abgespeichert und dann beim Boot wieder geladen. Zweck ist die Steuerung interner Programmabläufe, bzw Aktivierung von Features. Bitte Abspeichern nicht vergessen!

Flags:

enable tracking

enabled Tracker-Betrieb

enable iGate

enabled auf iGate-Betrieb

enable deep sleep

aktiviert DeepSleep Mode – Tracker geht zwischen Aussendungen in Tiefschlaf-Modus und wird vor nächstem Aussendungszeitpunkt geweckt, die GPS Position überprüft, sendet das APRS Paket und legt sich bis zum nächsten TX wieder Schlafen. Der Stromverbrauch wird somit wesentlich reduziert.

enable GPS

GPS-Empfänger kann zur Reduktion des Stromverbrauchs abgedreht werden, wenn nicht benötigt.

split RX/TX QRG

erlaubt die Trennung von Empfangs-/Sendefrequenz

APRS compressed format

Aktiviert das compressed Format bei APRS-Positionsdaten (Base 91)

APRS send Altitude

schickt Höhe mit

APRS send Batt Voltage

schickt Batteriespannung im APRS-Kommentar

APRS send Comment

schickt Kommentar mit APRS Aussendungen

GPS data via USB

ohne Funktion / Diagnostic

KISS Mode via USB

noch ungetestet, wahrscheinlich in späterer Version supported

KISS Mode via NW-Socket

noch ohne Funktion

enable WiFi

aktiviert das WiFi Interface

TELNET Server

startet Telnet-Server am Tracker zur Ausgabe von empfangenen/gesendeten Paketen kann über ,telnet ip@_tracker 23' verwendet werden

enable BUZZER

erlaubt den Lautsprecher an- oder abzudrehen. Der Tracker schickt Benachrichtigungen, Infos oder Alarme per Lautsprecher/Buzzer.

APRS Messages

The screenshot shows the 'Messages' section of the LoRa APRS Tracker OE1ACM-19 web interface. The page title is 'LoRa APRS Tracker OE1ACM-19' by OE1ACM & OE1CGC. The left sidebar contains navigation links for Home, Configuration (Config WIFI, Config APRS Tracker, Config APRS iGate, Config HW & Params), Messages (APRS Messages), Firmware (FW Update via OTA, FW Update via Web, Restart Tracker), and Links (www.lora-aprs.at, Show me on APRS.fi). The main content area displays a table of messages:

Sender	Target	Time	ack	Text
OE1ACM-11	OE1ACM-29	<116>	(190)	: No GPS-Fix Batt=0.67V (190
OE1ACM-11	OE1ACM-19	<273>	(0)	:QRV?

Below the table, there is a 'Send Message to:' field with 'OE1ACM-11' entered, a 'Text' field with 'ab 19:00h - 73!' entered, and a 'Senden' button. The footer indicates 'Firmware: FW 1.66 OE1ACM'.

Vorhanden, aber muss in den nächsten Versionen noch ausgebaut werden.

FW Update via OTA

Startet den OTA Server am Tracker bei Bedarf für Over-The-Air Updates falls Code geflucht oder Daten aus Entwicklungsumgebung transferiert werden sollen.

FW Update via Web

The screenshot shows the 'Download Firmware from lora-aprs.at' section of the LoRa APRS Tracker OE1ACM-19 web interface. The page title is 'LoRa APRS Tracker OE1ACM-19' by OE1ACM & OE1CGC. The left sidebar contains navigation links for Home, Configuration (Config WIFI, Config APRS Tracker, Config APRS iGate, Config HW & Params), Messages (APRS Messages), Firmware (FW Update via OTA, FW Update via Web, Restart Tracker), and Links (www.lora-aprs.at, Show me on APRS.fi). The main content area displays the 'Download Firmware from lora-aprs.at' section with the text 'For available Firmware check: lora-aprs.at/Software' and a 'Version Number:' dropdown menu. The footer indicates 'Firmware: FW 1.66 OE1ACM'.

Es kann eine andere Firmware Version von unseren Server geladen werden. i.E Versionsnummer 166 entspricht FW v 1.66. Die geladene FW wird nach Überprüfung

installiert. Der Tracker muss dafür natürlich unseren Server www.lora-aprs.at erreichen können – d.h. WiFi muss im Wlan konfiguriert und aktiv sein ...

Ein FW-Update auf die letzte Version kann auch aus dem Menue ‚Main -> Setup/Config/FW-Update‘ direkt am Tracker durchgeführt werden. Der Tracker wird dann auf die ‚letztverfügbare und freigegebene FW Version‘ automatisch upgedated.

Restart Tracker

Restartet den Tracker

Das Menü am Tracker & Bedienung

Der Tracker bietet verschiedene Funktionen, die im Wesentlichen durch die Feature-Settings gesteuert werden. So werden gewisse Subsysteme (z.B WiFi, Telnet-Access, WebServer, etc ...) beim Start automatisch aktiviert, bzw, der Tracker im gewünschten Mode betrieben (Tracker-, iGate-Mode). Die wichtigsten Einstellungen/Features werden in der Statuszeile in den Bildschirmen angezeigt.

Bedient wird der Tracker über die Weboberfläche oder das kleine Menü-System am Tracker über die drei Tasten ‚Links‘ ‚Enter‘ ‚Rechts‘.

Die beiden äußeren Tasten werden meistens zum Blättern verwendet, die mittlere Taste (Enter) dient zum Aufruf von Menüs, zur Umschaltung oder zur Bestätigung. Die Wichtigsten Informationen werden in der Statuszeile angezeigt (soweit Platz vorhanden).

Statuszeile:

Rechts: Anzahl der gesehenen GPS-Satelliten

Mitte: Bildschirmname

Links: gesetzte Features/Funktionen:

G - GPS enabled

T – Tracking Mode

S – SleepMode

W- WiFi enabled



Message erhalten



Des Weiteren werden meist in der untersten Zeile das verwendete Call+SID, der geschätzte Akku-Stand und die Bildschirmseite angezeigt.

Die Bildschirmseiten können über die beiden äußeren Tasten ‚horizontal‘ durchgeblättert werden.



Kurzbeschreibung der Bildschirmseiten:

- 0 Geschwindigkeit GPS, Altitude
- 1 System: FW-Version, BattVoltage, Sekunden bis zur nächsten Aussendung, free Mem
- 2 APRSSettings: Call+SID, Kommentar, APRS-Symbol/Table
- 3 TRX Settings: Intervall Secs, TX-PWR, RX/TX QRG
- 4 GPS I: Position, Altitude, Direction, Speed
- 5 GPSII: Zeit seit letztem GPS-Fix, Sats seen, GPS-Time
- 6 MH Locator: Maidenhead Locator (+ Fix Status)
- 7 Telnet: IP@ & Portnummer bei gestartetem Telnet Server (wird zur Ausgabe der empfangenen/gesendeten APRS Daten verwendet)
- 8 Received: Anzeige empfangener Pakete
- 9 RSSI: Tabelle der RSSI der letzten fünf empfangenen Pakete
- 10 Transmitted: letztes gesendete Paket
- 11 Message: empfangene Message
- 12 WiFi AP: bei aktiven WiFi SSID des verbundenen WLANS und eigene IP Adresse, unter der der WebServer zur Konfig läuft
- 13 BME 280: Sensordaten wenn BME280 Sensor connected
- 14 Kompass aus GPS-Daten Richtung, Winkel, Speed
- 15 – Platzhalter für Erweiterungen ---
- 16 – Platzhalter für Erweiterungen ---
- 17 Messages: Tabelle empfangene Messages
- 18 – Platzhalter für Erweiterungen ---
- 19 – Platzhalter für Erweiterungen ---
- 20 KISS TNC – Anzeige KISS-Mode (grundsätzlich implementiert aber noch nicht getestet!)
- 21 Message: Anzeige der gewählten Message mit Aktion (Enter-Taste bringt Sub_Menue: List Messages, Send quick Message aus Tabelle, Query my Report, (Quick QSO mit der Station – noch nicht implementiert) , exit
- 22 Time: Zeit in UTC entweder vom GPS oder NTP-Timeserver bei verbundenem WLAN.
- 23 iGate: Status iGatebei aktivem iGate
- 24 – Platzhalter für Erweiterungen ---
- 25 – Platzhalter für Erweiterungen ---

Das Haupt-Menü (Main Menu)

Durch Drücken der mittleren Taste (Enter) gelangt man ins ‚Main Menue‘ aus dem gewisse weitere Funktionen aufgerufen werden. Gescrollt wird mit den äußeren Tasten, bestätigt mit (Enter).



Achtung während man sich im Menüsystem befindet, sind die meisten Hintergrundaufgaben gestoppt. Diese werden weitergeführt, sobald das Menüsystem wider per ‚Exit‘ verlassen wird. Als Warnung leuchtet die gelbe LED permanent, bis das Menüsystem wieder verlassen wird und der Tracker wieder seine Hintergrundaufgaben ausführt.

Die wichtigsten Punkte im Main-Menu:

- **TX Position now** : sendet ein APRS-Paket mit momentaner GPS-Position
- **Compass**: Abkürzung zu Bildschirm 14
- **Stations**: Liste empangener Stationen mit Aktionsmenü List Messages, Send quick Message aus Tabelle, Query my Report, (Quick QSO mit der Station – noch nicht implementiert)
- **Messages**: List Messages, Send quick Message aus Tabelle, Query my Report, (Quick QSO mit der Station – noch nicht implementiert)
- **RX/TX Packets**: Abkürzung zu Bs 8
- **Info**: Submenü
- **Setup/Config**:
 - Untermenü Einstellungen:
 - **Start Wifi** Startet WiFi interface (setzt enable_WiFi nicht permanent – nur bis zum nächsten Boot)
 - **Config APRS Params**: Call-SID, Symbol, Table, Kommentar, TX Power 5 min – 23 Max, QRG (nur Anzeige) , TX-Intervall. Das Untermenü wird per ‚exit‘ verlassen. Dabei Parameter im NVRAM abspeichern, um diese nach Restarts zu erhalten.
 - **Config HW & Params** Feature Settings
 - **Messages** – Platzhalter – für spätere Konfig der Kurztexte.
 - **IO Expander** – Platzhalter – für weitere Konfig des IO-Expander Moduls
 - **Scan I2C Bus**: scannt den I2C Bus nach Devices. 0x3c Bildschirm,....
 - **KISS TNC** - startet Kiss TNC auf Seriell – zwar implementiert, aber noch nicht ausreichend getestet.
 - **Calib Sensor**: Kalibration des BME 280 Sensors nach momentaner Höhe (GPS-Höhe wird vorgeschlagen)

- **FW Update**: prüft Verbindung zu unserem Server, installiert die Letzversion der freigegebenen Firmware automatisch und bootet mit der neuen Firmware.
 - **Restart** restarted den Tracker
 - **ESP32 OS Info** zeigt infos über OS und Prozessoren
 - **RESET ALL** setzt Alle Einstellungen zurück
 - **Set Debug LVL** setzt Debug Mode
 - **Reset WiFi-Conf** setzt WiFi Einstellungen Zurück und löscht eingegebene SSID/Password Kombinationen
 - **Exit** Ausstieg aus dem Menü-System
-
- **Set Mode**: bestimmt den Betriebsmodus des Trackers:
 - Run as:
 - **Tracker**: Tracker-Modus
 - **APRS-IS Gateway**: iGate Mode using Peter's (OE5BPA) APRS-IS lib.
 - **KISS-TNC**: Startet KISS-Mode (noch nicht getestet).
 - **Exit**
 - **Remote IO-Expander** steuert remote Tracker mit IO-Expander Modul.
 - **Statusabfrage**
 - **Schalten des IO-Expanders / Remot Relais Karte.**
 - ...
 - **Power-SAVE** versetzt den Tracker in den Tief-Schlaf Modus, der Stromverbrauch sinkt auf ca 3-4 mA (1W Tracker). Aufgeweckt wird der Tracker durch Tastendruck (Enter). Sollte der Tracker längere Zeit nicht verwendet werden, bitte über den Schalter abdrehen (0mA Verbrauch).
 - **Exit** Ausstieg aus dem Menü

Texteingabe/Editieren in **Config APRS Params.**

Die Texteingabe über drei Tasten ist sehr beschränkt und wir haben auf die Konfiguration der iGate Parameter über das Tracker-Menü verzichtet. – Bitte WebServer zur Konfig verwenden, WiFi muss für iGate Betrie ohnehin aktiv sein, somit kann der WebServer auch zur Konfig verwendet werden.

Für die Tracker Konfig kann das Tracker-Menü verwendet werden.

,Links' ,Rechts' zum Scrollen und ,Enter' zum Auswählen



Buchstaben-Modus ,A-Z' es wird in der Bustaben-Tabelle gescrollt, Buchstaben/Zahlen können selektiert werden.(Abb 1)

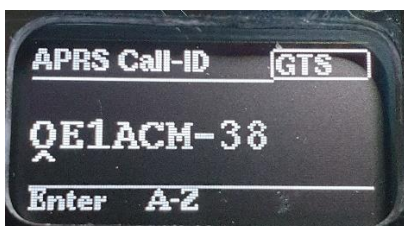
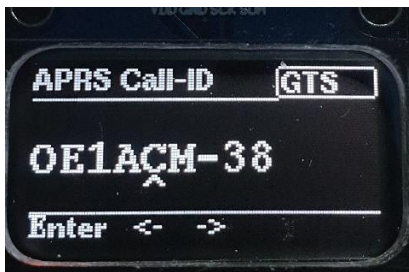
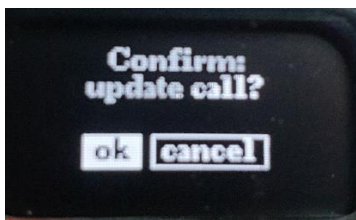


Abbildung 1

Mit ,Enter' wird zwischen Buchstaben (A-Z) und Cursor-Position (<- ->) umgeschaltet.



Zum Verlassen der Editierfunktion den Cursor entweder rechts oder Links (<- ->) aus dem Feld schieben. Es erscheint dann die Bestätigung.

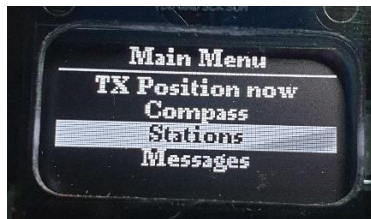


Sollten die Änderungen permanent gespeichert werden, bitte im NVRAM abspeichern.

Stations / Messages

Es gibt ein einfaches Menü zum Anzeigen der empfangenen Stationen und Messages.

(Die Daten werden noch nicht dauerhaft gespeichert und es findet noch kein Bestätigungsverkehr statt– dies ist für zukünftige Versionen vorgesehen, momentan noch nicht implementiert.)



Auswahl Stations oder Messages...

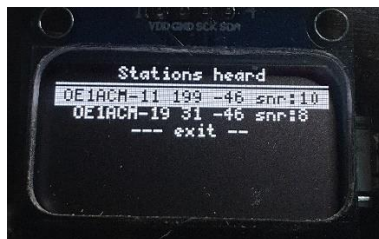
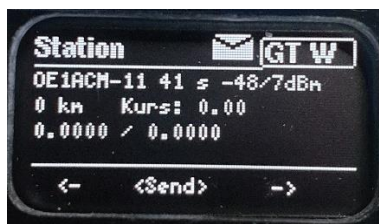


Tabelle mit empfangenen Stationen



Zum Verlassen der Seite bitte ,links' oder ,rechts' Tasten drücken, über (Enter) wird ein weiteres Untermenü gestartet



List messages: zeigt Messages für die ausgewählte Station

Send Messages: sendet Kurznachrichten aus vordefinierter Tabelle an Station.

Query Report: Anfrage über Remote-Command den Rapport zu schicken

Quick QSO: noch nicht implementiert – geplant Austausch von Position, Name, Rapport Entfernung,etc ... und Eintrag ins Log am Tracker.

Stromverbrauch und Sleep-Mode

Der Stromverbrauch ist abhängig von aktiven Subsystemen. Im Normalbetrieb Tracker/GPS/Display beläuft sich der Verbrauch auf ca. 90-140 mA (je nach GPS-Fix) . Das WiFi Modul benötigt weitere 70mA bei Betrieb.

Am 1W Tracker wird bei TX mit max Leistung bis zu 1A vom Akku gezogen.

Unter normalen Umständen reicht die Leistung eines 900-2000 mAH Akkus für mehrere Stunden Betrieb, wobei das WiFi Modul nicht dauerhaft in Betrieb sein sollte.

Um die Akku-Laufzeit jedoch wesentlich zu erhöhen, haben wir einen **Deep-Sleep Mode (S)** im Tracker implementiert. Der Tracker schläft zwischen den Aussendungen mit minimalem Verbrauch, benötigte Subsysteme (CPU, GPS, LoRa Modul) werden kurz vor nächster Aussendung geweckt und initialisiert, das GPS Signal geprüft, es erfolgt die Aussendung. Nach erfolgter Aussendung wird der Schlaf-Modus wieder aktiviert bis zur nächsten, geplanten Aussendung.

Wird innerhalb von 60 Sekunden kein gültiger GPS-Fix gesehen, werden die Intervalle auf 20 Minuten im Schlafmodus erweitert, bis wieder ein gültiges GPS Signal/Fix erreicht wird.

Der Stromverbrauch wird somit wesentlich reduziert, die mögliche Betriebsdauer pro Akku-Ladung erweitert sich (auf bis zu mehrere Tage).

APRS-Aussendung - Intervall

Der Aussendungs-Intervall ist geschwindigkeits-abhängig.

< 5 km/h : alle 10 Intervalle

< 10 km/h: alle 3 Intervalle

< 30 km/h: alle 2 Intervalle

< 60 km/h: 1 Intervall

IGate – Modus

Wie bereits erwähnt haben wir Peter's (OE5BPA) APRS-IS Library mit eingebaut, um iGate-Betrieb zu ermöglichen. Die Tracker können zwischen den verschiedenen Modi umgeschaltet werden. Für den IGate-Betrieb ist WiFi/WLAN notwendig, um die APRS-IS Server erreichen zu können. Beim Umschalten auf iGate-Mode wird WiFi automatisch enabled und im NVRAM abgespeichert, da der Tracker möglicherweise mit neuer Konfiguration gestartet wird.

Wichtig ist auch die korrekte Konfiguration der iGate Parameter und des gültigen APRS-IS Passwords.

i.e. Password-Generator: <http://n5dux.com/ham/aprs-passcode/>

Es gibt die Möglichkeit mehrere APRS-IS Server zu definieren. Die Erreichbarkeit wird mittels ‚Ping‘ überprüft, bzw wird bei Ausfall auf Alternativ-Server umgeschaltet. Sind beide Server nicht erreichbar, wird nach zehn erfolglosen Versuchen wieder auf Tracker-Mode zurückgeschaltet. Es wird ein Alarm über den eingebauten Lautsprecher ausgegeben – falls das zu nervig ist, kann der Buzzer über die Feature-Settings disabled werden.



Es gibt weiterhin die Möglichkeit, die Position für Baken-Aussendungen vom GPS zu übernehmen. Dies ist allerdings mit Vorsicht zu verwenden! Ich weiß nicht, ob auch mobile Gateways im APRS-IS unterstützt werden, falls ihr mal vergesst, das iGATE im Auto abzdrehen...

Btw, ihr könnt euer Gateway mit Statistiken sehen auf Port 14501 am APRS-IS server ...

http://<aprs-is_server>:14501 <http://euro.aprs2.net:14501/>

Clients

Port	Username	Address	Verified	Up	Last in	Software	Packets Tx	Packets Rx	Bytes Tx	Bytes Rx	Tx/Rx bytes/s	OutQ	MsgRepts	Filter
14580	S54LD-KM	193.77.58.121:61847	Yes	1s	1s	ESP8266_S54LD 1.00	0	3/0/1	71	227	7.1 / 23	0	2	m/1
14580	DB0LC-1	93.218.137.186:5004	Yes	1m	4s		623	17/16/0	58899	1634	1090 / 10	0	4	p/D/OE/LX/HB/P/WX
14580	OE3PQA-13	192.164.132.235:56090	Yes	3m	3m	ESP8266_SM 0.1	0	2/0/0	774	232	7.8 / 0	0	1	m/1
14580	NOCALL	193.33.111.22:41424	No	6m	6m	APRS2RRD 1.3.2-23112020	6	0/0/0	2129	81	0 / 0	0	0	b/SR9NSK-1/SR9NSK-2
14580	OE1ACM-38	80.108.164.199:57078	No	10m	10m	ESP32-APRS-IS 0.1	0	1/0/1	2492	151	0 / 0	0	0	

Zusätzliche Features / Paket-Typen

LoRa bietet eine gesicherte Datenübertragung über weite Entfernungen - einfach viel zu schade um damit nur APRS-Betrieb zu machen...

Wir haben begonnen, die Liste der Features zu erweitern, um weitere Möglichkeiten aufzuzeigen. Hier kommt auch der vieldiskutierte, ominöse 3-Byte Packet-Header zum Tragen, der uns erlaubt, unterschiedliche Protokoll-Typen zu unterscheiden. Der 3-Byte Header ist im folgenden Format „Source_Destination_Packet-Type“ aufgebaut, wobei Packet-Type **,>** für APRS Pakete vorgesehen ist.

Wir verwenden andere Paket-Typen zusätzlich, die nicht ins APRS-Netzwerk weitergeleitet werden sollen und von iGates ausgefiltert werden.

Remote-Steuerung / Abfrage

Als Demo haben wir einige Funktionen der Remote Steuerung bereits implementiert – Beispiel **I/O Expander** Module, über die remote-Relais-Steuerungen oder Statusabfragen durchgeführt werden können. Bestätigung über APRS-Message...

Ein weiteres Beispiel ist die **Rapport-Abfrage** des eigenen Signals von der Gegenstation. Der Rapport wird auf Anfrage direkt zurückgesendet und per APRS-Message übertragen.

Ein **Quick-QSO-Mode** mit Austausch von Namen, Position, Rapport, Entfernung und Eintrag ins Log-File lässt sich somit auch einfach durchführen ...

Die Liste an Remote-Befehlen lässt sich bei Bedarf erweitern.

Firmware-Update:

Automatisch von unserem Server (**https**) über den Tracker oder WebServer.

Die rootCACertificates wurden leider stillschweigend am 2. Dezember 2020 geändert und wir brauchen ein neues Certificate, das erst wieder mit FW v 1.67 inkludiert ist.

Für ältere Versionen (< 1.67) bitte OTA oder Seriell/USB verwenden.

Um in dieses Zertifikats-Änderungs Problem nicht nochmals reinzulaufen, hab ich die FW erweitert. Es können jetzt beliebige Zertifikate abgespeichert werden.

Versteckt, Aufruf über den Browser des Trackers **http://192.168.2.160/certificate**

Webserver

LoRa APRS Tracker OE1ACM-38

by OE1ACM & OE1CGC

- Home
- Configuration
 - Config WIFI
 - Config APRS Tracker
 - Config APRS iGate
 - Config HW & Params
- Messages
 - APRS Messages
- Firmware
 - FW Update via OTA
 - FW Update via Web
- Restart Tracker
- Links
 - www.lora-aprs.at
 - Show me on APRS.fi

Add rootCAcertificate for SSL encryption

```
-----BEGIN CERTIFICATE-----
MIIFPjCCAv6gAwIBAgIRAJErcEz-PDBInU/bWLiWnXlowDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBHMCMVxKTnBgnVBAOTIEudGvYmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYVZDQGEwJULjE1IFJvby3QgWDEwHhcNMjAwOTA0MDAw
MDAwWhcNMjUwOTE1MTYwMDAwWjAQMjcwQVZDQGEwJUVzENMBQGA1UEChMNTGV0
J3MgRW5jcmlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQC7AhUozPag1NMPeuyNVZLD+ILxmaZ6QoinXsaqt.Su5xUyxr45r+XX
Io9cPR5QUVTVXjJ6oojk29YI8QqlObvU7wy7bjcCwXPNZ00ftz2nwWgsbvsCU
JWCWH+jdxsxpHkzhm+/b5DtFUkwwgcFTzjTiu61ru2P3mBw4qVug7Ztdpe1QRr
K908ZutmNhZ6a4uFVymZ+DAXXbpyb/ubka3Shlg9F8EnCbvxK/eG3MHacV3URu
PMrSXBILxgZ3Vms/E19Jc51P/Oo12R6X/ExjgmA13P51T+c8B5fWecb0Uz20k/
5mzk53cU6cG/k1FHaFpriVluxFMUgP17Vgh19eYAgMBAAgJggEIMIIBDDAObgNV
H08BAf8EBAMCAAYwHQYDVR0I1BBYwFAYIKwYBBQUHAWIGCCsGAQQUFBwMEMB
BIGA1UdEwEwB/wQIMAYBAf8CAQAwHQYDVR0OBQYEFBQusxe3WfBLr1AJQYfr52
LFMLGMB8GA1UdIwQYMBAAFHm0WeZ7tuXkAXOACijIG1j26ZtuMDIGCCsGA
QQUFBwEBBCYwJDA1BgggrBgEFBQcwAoYwHR0cDovL3gxlmkubGVuY3Iub3Jn
LzAnBgNVHR8EIDAEeMBYgGgAYhhZodHRwOi8veDEuYy5eZW5jcjE5cmcvMCI
GA1UdIARbMBkwCAQYz4EMAQIBMA0GCysGAQQBggt8TAQEBMA0GCsGSIb3DQ
EBwUAA4ICAQCFyk5HPqP3hUSFvNveLKYy611TR6WFTNlc1QtgaDpw+34IL9fz
LwLwLdu0/Zeln7kiJ+m74uyA+e1Rt8kc607TKC53w1ikfmZM4/RvTz3M6UR
+SUzhK8jCdLwM5YL6kvZKGRSgij3yLgjeWqCpRiVz6DQOZCkcheAmCJ8MgyJ
u5z1zyZMjAvnnAT45tRAXekrsu94eQ4egdRCnbWSDty7kh+BIm
```

Save

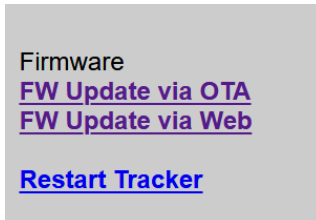
Firmware: FW 1.67 OE1ACM

Es wird das neue, Default Certificate für ‚www.lora-aprs.at‘ Domain übernommen, es kann aber überschrieben und im Filesystem des Trackers abgespeichert werden.

Über **OTA**:

Tracker muss im WLAN hängen.

Über den Webserver des Trackers den OTA_Server starten „FW_Update via OTA“



Sobald der OTA-Server am Tracker läuft, wird OTA_upd am WiFi Bildschirm angezeigt



Es kann jetzt das ,espota , Tool unter Angabe der IP@ des Trackers zum Updaten der Firmware verwendet werden. Bitte den Port 3232 in der Firewall freigeben.

hardware\espressif\esp32/tools/

espota.exe -i 192.168.2.156 -p 3232 --auth=ota -f ESP32_TTGO_v1_67.ino.esp32.bin

Sending invitation to 192.168.2.156

Authenticating...OK

Uploading.....
.....
.....
.....
.....
.....

Der update kann natürlich auch über Seriell / USB erfolgen:

Die Syntax:

```
hardware\espressif\esp32/tools/esptool/esptool.exe --chip esp32 --port COM12 --baud \
921600 --before default_reset --after hard_reset write_flash -z --flash_mode dio --\
flash_freq 80m --flash_size detect 0xe000 \
hardware\espressif\esp32/tools/partitions/boot_app0.bin 0x1000 \
hardware\espressif\esp32/tools/sdk/bin/bootloader_dio_80m.bin 0x10000 \
ESP32_TTGO_TrackUARTv1_67.ino.bin 0x8000 \
ESP32_TTGO_TrackUARTv1_67.ino.partitions.bin
```

In einigen Fällen kann ein Checksum Error beim Download auftreten. Der GPIO12 Pin wird auch zur Umschaltung der Flash-Voltage verwendet und kann auf 1.8V umschalten. Wir verwenden die EFUSE `set_flash_voltage 3.3V` um diesen fix auf 3.3V zu setzen.

espfuse.py -p COMxx summary

die Flash-Voltage steht in der letzten Zeile der Summary

espfuse.py -p COMxx set_flash_voltage 3.3V

er fragt nach Bestätigung: BURN eingeben.