
Inhaltsverzeichnis

DXL - APRStracker

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 14. September 2013, 21:46
Uhr (Quelltext anzeigen)
OE5HPM (Diskussion | Beiträge)
K (Downloads als ZIP anstatt TGZ)
← Zum vorherigen Versionsunterschied

Aktuelle Version vom 24. September
2023, 20:17 Uhr (Quelltext anzeigen)
OE2WAO (Diskussion | Beiträge)
K
Markierung: Visuelle Bearbeitung

(25 dazwischenliegende Versionen von 3 Benutzern werden nicht angezeigt)

Zeile 7:		Zeile 7:	
	Als Minimum an Hardware wird benötigt		Als Minimum an Hardware wird benötigt
-	* ein Mikrocontroller ATTiny13, ATTiny2313, Atmea88 oder auch andere Typen mit (Quarz)-Takt durch Anpassen der I/O Pins im Source	+	
-	* Quarz frei waehlbar ca. 6..20MHz (Im Source eingeben).	+	*ein Mikrocontroller ATTiny13, ATTiny2313, Atmea88 oder auch andere Typen mit (Quarz)-Takt durch Anpassen der I/O Pins im Source
-	* PTT-Transistor	+	*Quarz frei waehlbar ca. 6..20MHz (Im Source eingeben).
-	* RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt.	+	*PTT-Transistor
		+	*RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt.
	==Software==		==Software==
	Opensource Software von OE5DXL, in Assembler geschrieben, welche den Tracker zum Leben erweckt:		Opensource Software von OE5DXL, in Assembler geschrieben, welche den Tracker zum Leben erweckt:
-	[[Datei:aprsTracker.zip]]	+	[[Datei:AprsTracker.zip AprsTracker.zip]]

<p>Der Assemblercode wird im einfachsten Fall mit dem Compiler 'gavasm' [http://www.avr-asm-tutorial.net/gavasm/index_de.html Gerd's AVR Assembler], welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt. Es ist auch möglich dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.</p>	<p>Der Assemblercode wird im einfachsten Fall mit dem Compiler 'gavasm' [http://www.avr-asm-tutorial.net/gavasm/index_de.html Gerd's AVR Assembler], welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt (z.B.: mit AVRdude [http://www.nongnu.org/avrdude/] und dem USBasp Programmier [http://www.fischl.de/usbasp/]). Es ist auch möglich mit dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.</p>
	+ Fertige Kompilate:
{ class="wikitable sortable"	{ class="wikitable sortable"
- ! Prozessor	+ !Prozessor
- ! Systemtakt	+ !Systemtakt
- ! GPS-Baud	+ !GPS-Baud
- ! AFSK Baud	+ !AFSK Baud
- ! Download	+ !Download
-	-
- ATME1 Tiny13	+ ATME1 Tiny13
- 10 Mhz	+ 10 Mhz
- 4800	+ 4800
- 1200	+ 1200
- [[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]	+ [[Datei: Dxl-aprsTracker-tiny13-10MHz-4800Bd.zip verweis=Special:FilePath/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]
-	-
- ATME1 Tiny13	+ ATME1 Tiny13
- 10 Mhz	+ 10 Mhz
- 9600	+ 9600
- 1200	+ 1200

<div>-</div> <div> [[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]</div> <div> -</div> <div> }</div> <div>-</div> <div></div> <div></div> <div>Als Input dienen serielle GPS-Daten (GPRMC und GPGGA), der Tracker generiert daraus anhand der programmierten Konfiguration als Output (nebst PTT) die AFSK-Modulation im APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Hoehe, welche dem Funkgerät zugeführt wird.</div>	+	<div>[[Datei:Dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath/dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]</div> <div> -</div> <div> }</div> <div></div> <div></div> <div>Als Input dienen serielle GPS-Daten (GPRMC und GPGGA), der Tracker generiert daraus anhand der programmierten Konfiguration als Output (nebst PTT) die AFSK-Modulation im APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Hoehe, welche dem Funkgerät zugeführt wird.</div>
<div>Zeile 46:</div> <div></div> <div>==Hardware==</div> <div>Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:</div> <div></div> <div>-</div> <div>[[Datei:dxlTracker-schematic.png]]</div> <div></div> <div>Detaillierte Schaltungsbeschreibung folgt.</div>	+	<div>Zeile 47:</div> <div></div> <div>==Hardware==</div> <div>===Trackerschaltung===</div> <div>Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:</div> <div></div> <div>[[Datei:DxlTracker-schematic.png DxlTracker-schematic.png]]</div> <div></div> <div>Detaillierte Schaltungsbeschreibung folgt.</div> <div>+</div> <div></div> <div>===Geeignete GPS-Empfänger===</div> <div>{ class="wikitable sortable"</div> <div>!Hersteller</div> <div>!Type</div> <div>!Baudrate</div>

+	!Versorgungsspannung	
+	!Stromaufnahme	
+	!Preis	
+	!Datenblatt	
+	-	
+	Fastrax	
+	UP501	
+	9600	
+	3.3V	
+	~30mA	
+	ca. 22€ (inkl.Versand)	
+	http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/UP501.pdf http://www.adafruit.com/datasheets/UP501_brochure_rev_1_2.pdf	
+	-	
+	Ublox	
+	NEO-6M	
+	9600	
+	3.3V	
+	~22mA	
+	ca. 10€ (inkl.Versand)	
+	https://www.u-blox.com/images/downloads/Product_Docs/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf Ublox NEO-6M Datenblatt]	
+	-	
+	Globaltop	
+	GTPA013	
+	9600	
+	3.3V	

+ |~20mA

+ |ca. 19€

+ |<http://www.adafruit.com/datasheets/GlobalTop-FGPMOPA6H-Datasheet-V0A.pdf>

+ |}

==Konfiguration==

==Konfiguration==

+ |[[Datei:DXLtrackerConfig.png|rechts|mini|300px|Screenshot DXL-Trackerconfig]]

GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

Zeile 58:

AFSK-Baud und Tonfrequenzen (Shift) sind ebenso frei konfigurierbar, gebräuchlich sind:

- * 300 auf Kurzwelle

- * 1200 auf UKW

Zeile 96:

AFSK-Baud und Tonfrequenzen (Shift) sind ebenso frei konfigurierbar, gebräuchlich sind:

+ *300 auf Kurzwelle

+ *1200 auf UKW

Zur Einstellung von HUB bzw. Mikrofonpegel eignet sich am besten ein Poti.

Zur Einstellung von HUB bzw. Mikrofonpegel eignet sich am besten ein Poti.

Zeile 73:

Zur Konfiguration kommt ein kleines Tool von OE5HPM - DXLtrackerConfig zum Einsatz, dies generiert einen Konfigurationsstring für den DXLtracker und schickt selbigen über die serielle Schnittstelle raus.

Zeile 112:

Zur Konfiguration kommt ein kleines Tool von OE5HPM - DXLtrackerConfig zum Einsatz, dies generiert einen Konfigurationsstring für den DXLtracker und schickt selbigen über die serielle Schnittstelle raus.

– [[Datei:DXLtrackerConfig.png]]

+ [[Datei:DXLtrackerConfig.zip]]

+

+ ==Referenzprojekte==

+ ===OE5EEP, Heinz===

+ [[Datei:Oe5eep 2. jpg|rechts|mini|hochkant|200px|Gesamtansicht|verweis=Special:FilePath/oe5eep_2.jpg]]

+

+ [[Datei:Oe5eep 1. jpg|rechts|mini|200px|Gesamtansicht|verweis=Special:FilePath/oe5eep_1.jpg]]

+

+ [[Datei:Oe5eep 3. jpg|rechts|mini|200px|Innenleben|verweis=Special:FilePath/oe5eep_3.jpg]]

+

+ [[Datei:Oe5eep 4h. jpg|rechts|mini|200px|Innenleben-Detail-Tracker+GPS-RX|verweis=Special:FilePath/oe5eep_4h.jpg]]

+

+

+ **Mein Aufbau eines DXL Modems mit einem Fastrax UP501 GPS Empfänger, den ich für Bergtouren im Rahmen des SOTA-Programms einsetzen möchte: Das DXL Modem wurde mir freundlicherweise vom ADL501 zur Verfügung gestellt, der UP501 ist über e-Bay zu beziehen. Als Spannungsquelle für beide Baugruppen gemeinsam dienen 4 NiMH AAA-Zellen von Conrad mit einer nominalen Kapazität von 1100mAh, was für einen ganzen**

Bergtaq leicht ausreicht. Die stabilisierte Spannung von 3,3V für des GPS Modul kann nach Brückung einer Diode an Pin 7 des Sub-D Anschlusses des DXL Trackers abgegriffen werden.

+

+

Da ich der Verlässlichkeit von freien Verkabelungen nicht traue, hab ich das GPS Modul mittels der mitgelieferten Steckerleiste auf einer Lochrasterplatte aufgelötet und die Verkabelungen zum DXL Modem auf beiden Seiten der Lochrasterplatte auf eine 9-polige Sub-D Buchse geführt. Dazu kommt noch ein Aus /Ein Schalter und ein Halter für die NiMH Zellen. Eingebaut hab ich das Ganze in eine ehemalige Präsentationsverpackung von 2 Kugelschreibern. Dieses transparente Kunststoffgehäuse erlaubt es, das Aufleuchten der LED im Sendefall von außen zu erkennen und passt von der Größe her gut zum verwendeten Handfunkgerät (siehe Fotos). Der Aufbau wird am Handfunkgerät einfach mit Gummibändern (Stücke von einem Fahrradschlauch) befestigt. Die einzige elektrische Verbindung zum Handfunkgerät ist eine Audioleitung zur Mikrofonbuchse, in meinem Fall mit einem 2,5mm Mono-Klinkenstecker.

+

+

Bisher hab ich den Tracker auf mehreren Bergtouren eingesetzt und positive Erfahrungen gemacht. Der GPS Empfänger findet binnen weniger Minuten eine Position. Die Empfindlichkeit ist ausgezeichnet.

+ Teilweise kann ich im Inneren von Gebäuden eine Position bekommen, allerdings dann mit größerem Fehler. Jedenfalls ist der GPS-Empfänger deutlich empfindlicher als der in meinem LG Smartphone. Im Zusammenspiel mit einem schon etwas älteren IC-2e mit etwa 3W Sendeleistung und Gummiantenne ist es mir auf Touren in den Kalkalpen immer gelungen APRS-Meldungen im Gipfelbereich abzusetzen. Im Funkschatten von Bergen verlief das nicht so verlässlich. Das ist aber kein Problem der Kombination Tracker und Handfunkgerät sondern ist auf den dünnen Ausbau der sonstige APRS Infrastruktur zurückzuführen.

+

+

+ Dieses Selbstbauprojekt ist sowohl technisch als auch finanziell überschaubar. Ich hoffe, dass genaue Positionsmeldungen während meiner Bergtouren die SOTA Jäger unterstützen und gegebenenfalls zu meiner Sicherheit in den Bergen beitragen! Danke an OE5DXL für die Entwicklung der Basis für dieses nette Selbstbauprojekt!

+

+

+ 73 Heinz, OE5EEP

+

+

+ ==Prämiert==

+ Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!

	+	
	+	
	+	
	+	== LoRa Variante ==
	+	Von OE5DXL gibt es auch eine LoRa Variante für Hardware-Tracker des Typs TTGO bzw. LILYGO inkl. Unterstützung für komprimierte Aussendung und Verwendung von MIC-E.
	+	
	+	Zu finden ist die fertige INO-Datei in verschiedenen Ausführung, neben anderer Software, wie immer im [http://oe5dxl.ampr.org/aprs/tracker/HAMNET Indexserver von OE5DXL].
	+	
	+	==Weitere Projekte==
	+	Weiter passende Projekte von OE5DXL aus dieser Serie sind
-		[[Datei:DXLtrackerConfig.zip]]
	+	*[[DXL - APRSmap]] - Der neue APRS Client für Windows, Linux und ARM auf OSM Basis
	+	*[[TCE Tinycore Linux Projekt]] - Der mächtige und innovative Digipeater für APRS, Packet Radio, Echolink, u.v. m.

Aktuelle Version vom 24. September 2023, 20:17 Uhr

Inhaltsverzeichnis

1	Einleitung	12
2	Software	12
3	Hardware	13
3.1	Trackerschaltung	13
3.2	Geeignete GPS-Empfänger	13

4 Konfiguration	13
5 Referenzprojekte	14
5.1 OE5EEP, Heinz	14
6 Prämiert	15
7 LoRa Variante	16
8 Weitere Projekte	16

Einleitung

Der APRStracker von OE5DXL erlaubt es mit minimalstem Hardwareaufwand in der Betriebsart APRS QRV zu werden. Außerdem wurde bei der Entwicklung der Software darauf geachtet, die derzeitigen Möglichkeiten des APRS-Protokolls in Form von Mic-e optimalst auszunutzen. In der Praxis ist dies durch extrem kurze Frames erkennbar, was im Mobilbetrieb (QSB) erhebliche Vorteile bringt. Zu dem wird das schon etwas in die Jahre gekommene aber durchaus am effektivsten SSID-Pathrouting (im Configtool 'COMPRESSED' genannt) unterstützt.

Als Minimum an Hardware wird benötigt

- ein Mikrocontroller ATtiny13, ATtiny2313, Atmega88 oder auch andere Typen mit (Quarz)-Takt durch Anpassen der I/O Pins im Source
- Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).
- PTT-Transistor
- RC-Tiefpass zum wegfiltern der PWM-Frequenz benötigt.

Software

Opensource Software von OE5DXL, in Assembler geschrieben, welche den Tracker zum Leben erweckt:

Datei:[AprsTracker.zip](#)

Der Assemblercode wird im einfachsten Fall mit dem Compiler 'gavrasm' [Gerd's AVR Assembler](#), welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt (z.B.: mit AVRdude [\[1\]](#) und dem USBasp Programmer [\[2\]](#)). Es ist auch möglich mit dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.

Fertige Kompilate:

Prozessor	Systemtakt	GPS-Baud	AFSK Baud	Download
ATMEL Tiny13	10 Mhz	4800	1200	Datei:Dxl-aprsTracker-tiny13-10MHz-4800Bd.zip
ATMEL Tiny13	10 Mhz	9600	1200	Datei:Dxl-aprsTracker-tiny13-10MHz-9600Bd.zip

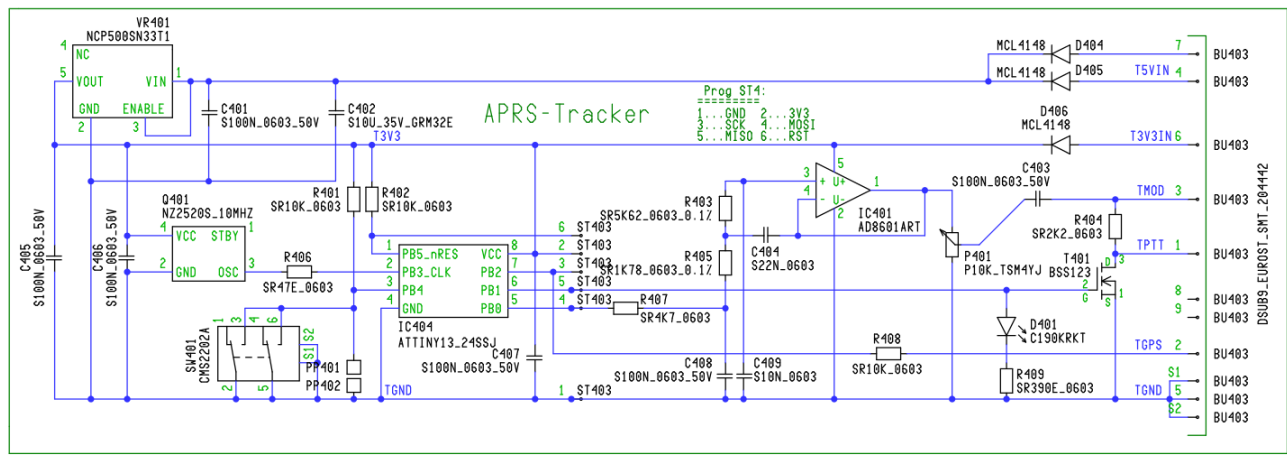
Als Input dienen serielle GPS-Daten (GPRMC und GPGLL), der Tracker generiert daraus anhand der programmierten Konfiguration als Output (nebst PTT) die AFSK-Modulation im APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Höhe, welche dem Funkgerät zugeführt wird.

Dabei sind (pro Profil) 2 Bakenzeiten und eine Geschwindigkeit einstellbar unter/ueber der langsam/schnell gebakt wird.

Hardware

Trackerschaltung

Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:



Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

Hersteller	Type	Baudrate	Versorgungsspannung	Stromaufnahme	Preis	
Fastrax	UP501	9600	3.3V	~30mA	ca. 22€ (inkl. Versand)	http://clo.../Se...http://da.../UP...pdf
Ublox	NEO-6M	9600	3.3V	~22mA	ca. 10€ (inkl. Versand)	Ubl...
Globaltop	GTPA013	9600	3.3V	~20mA	ca. 19€	http://da...FGI...VO...

Konfiguration

GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

Baudraten vom GPS bzw. PC zur Konfiguration sind je nach Prozessortakt im Bereich 300..200000, ueblich 4800, 9600.

AFSK-Baud und Tonfrequenzen (Shift) sind ebenso frei Konfigurierbar, gebräuchlich sind:

- 300 auf Kurzwelle
- 1200 auf UKW

Zur Einstellung von HUB bzw. Mikrofonpegel eignet sich am besten ein Poti.

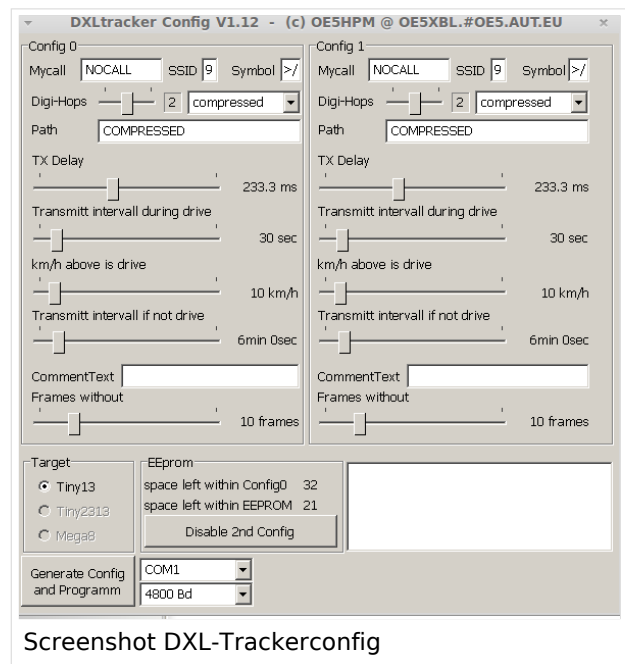
Um die PTT von Handfunkgeräten aufzutasten muss der Mikrofoneingang mit dem PTT Transistor kombiniert werden. Einfachste Variante ist im obigen Schaltungsbild sichtbar, der FET T402 zieht über einen 2k2 Widerstand den MIC-Eingang vom Handfunk gegen Masse und aktiviert dadurch die PTT, über C403 (100nF) wird die Modulation "eingekoppelt".

Als Option kann mit einem Jumper/Schalter (im Schaltbild SW401) zwischen 2 (am Tiny13) oder 4 (auf größeren Prozessoren) User-Profilen ausgewählt werden, zB. Fahrrad/Auto.

An einer Blink-Led für korrekten GPS-Empfang am Prozessor Pin wird gearbeitet.

Zur Konfiguration kommt ein kleines Tool von OE5HPM - DXLtrackerConfig zum Einsatz, dies generiert einen Konfigurationsstring für den DXLtracker und schickt selbigen über die serielle Schnittstelle raus.

[Datei:DXLtrackerConfig.zip](#)



Screenshot DXL-Trackerconfig

Referenzprojekte

OE5EEP, Heinz

Mein Aufbau eines DXL Modems mit einem Fastrax UP501 GPS Empfänger, den ich für Bergtouren im Rahmen des SOTA-Programms einsetzen möchte: Das DXL Modem wurde mir freundlicherweise vom ADL501 zur Verfügung gestellt, der UP501 ist über e-Bay zu beziehen. Als Spannungsquelle für beide Baugruppen gemeinsam dienen 4 NiMH AAA-Zellen von Conrad mit einer nominalen Kapazität von 1100mAh, was für einen ganzen Bergtag leicht ausreicht. Die stabilisierte Spannung von 3,3V für des GPS Modul kann nach Brückung einer Diode an Pin 7 des Sub-D Anschlusses des DXL Trackers abgegriffen werden.

Da ich der Verlässlichkeit von freien Verkabelungen nicht traue, hab ich das GPS Modul mittels der mitgelieferten Steckerleiste auf einer Lochrasterplatte aufgelötet und die Verkabelungen zum DXL Modem auf beiden Seiten der Lochrasterplatte auf eine 9-polige Sub-D Buchse geführt. Dazu kommt noch ein Aus/Ein Schalter und ein Halter für die NiMH Zellen. Eingebaut hab ich das Ganze in eine ehemalige Präsentationsverpackung von 2 Kugelschreibern. Dieses transparente Kunststoffgehäuse erlaubt es, das Aufleuchten der LED im Sendefall von außen zu erkennen und

passt von der Größe her gut zum verwendeten Handfunkgerät (siehe Fotos). Der Aufbau wird am Handfunkgerät einfach mit Gummibändern (Stücke von einem Fahrradschlauch) befestigt. Die einzige elektrische Verbindung zum Handfunkgerät ist eine Audioleitung zur Mikrofonbuchse, in meinem Fall mit einem 2,5mm Mono-Klinkenstecker.

Bisher hab ich den Tracker auf mehreren Bergtouren eingesetzt und positive Erfahrungen gemacht. Der GPS Empfänger findet binnen weniger Minuten eine Position. Die Empfindlichkeit ist ausgezeichnet. Teilweise kann ich im Inneren von Gebäuden eine Position bekommen, allerdings dann mit größerem Fehler. Jedenfalls ist der GPS-Empfänger deutlich empfindlicher als der in meinem LG Smartphone. Im Zusammenspiel mit einem schon etwas älteren IC-2e mit etwa 3W Sendeleistung und Gummiantenne ist es mir auf Touren in den Kalkalpen immer gelungen APRS-Meldungen im Gipfelbereich abzusetzen. Im Funkschatten von Bergen verlief das nicht so verlässlich. Das ist aber kein Problem der Kombination Tracker und Handfunkgerät sondern ist auf den dünnen Ausbau der sonstige APRS Infrastruktur zurückzuführen.

Dieses Selbstbauprojekt ist sowohl technisch als auch finanziell überschaubar. Ich hoffe, dass genaue Positionsmeldungen während meiner Bergtouren die SOTA Jäger unterstützen und gegebenenfalls zu meiner Sicherheit in den Bergen beitragen! Danke an OE5DXL für die Entwicklung der Basis für dieses nette Selbstbauprojekt!

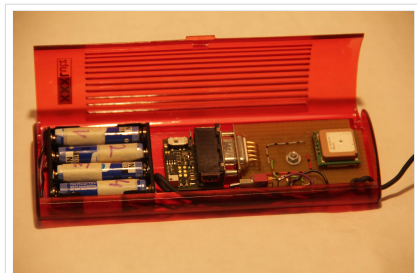
73 Heinz, OE5EEP



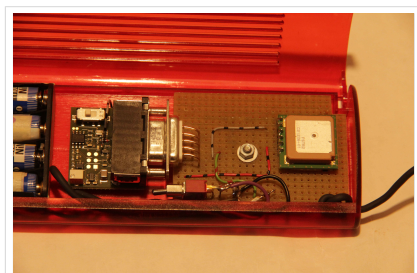
Gesamtansicht



Gesamtansicht



Innenleben



Innenleben-Detail-Tracker+GPS-RX

Prämiert

Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert !
GRATULATION !!

LoRa Variante

Von OE5DXL gibt es auch eine LoRa Variante für Hardware-Tracker des Typs TTGO bzw. LILYGO inkl. Unterstützung für komprimierte Aussendung und Verwendung von MIC-E.

Zu finden ist die fertige INO-Datei in verschiedenen Ausführung, neben anderer Software, wie immer im [HAMNET Indexserver von OE5DXL](#).

Weitere Projekte

Weiter passende Projekte von OE5DXL aus dieser Serie sind

- [DXL - APRSmap](#) - Der neue APRS Client für Windows, Linux und ARM auf OSM Basis
- [TCE Tinycore Linux Projekt](#) - Der mächtige und innovative Digipeater für APRS, Packet Radio, Echolink, u.v.m.