

Inhaltsverzeichnis

1. DXL - APRStracker	34
2. Benutzer:OE2WAO	10
3. Benutzer:Oe1mcu	18
4. DXL - APRSmap	26
5. Datei:DXLtrackerConfig.zip	42
6. TCE Tinycore Linux Projekt	51

DXL - APRStracker

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

Oe1mcu ([Diskussion](#) | [Beiträge](#))

Markierung: Visuelle Bearbeitung

← Zum vorherigen Versionsunterschied

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

OE2WAO ([Diskussion](#) | [Beiträge](#))

K

Markierung: Visuelle Bearbeitung

Zum nächsten Versionsunterschied →

Zeile 16:

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 '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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

[[Datei:**aprsTracker.zip**|**verweis=Special:FilePath**/aprsTracker.zip]]

+

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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

-

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

[[[Datei:**dxl-aprsTracker-tiny13-10MHz-4800Bd.zip**|**verweis=Special:FilePath**/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

+

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[[Datei:dxlTracker-schematic.png]]	+ [[[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis	
1 Einleitung	38
2 Software	38
3 Hardware	39
3.1 Trackerschaltung	39
3.2 Geeignete GPS-Empfänger	39

4 Konfiguration	39
5 Referenzprojekte	40
5.1 OE5EEP, Heinz	40
6 Prämiert	41
6.1 LoRa Variante	41
7 Weitere Projekte	41

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

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

Konfiguration

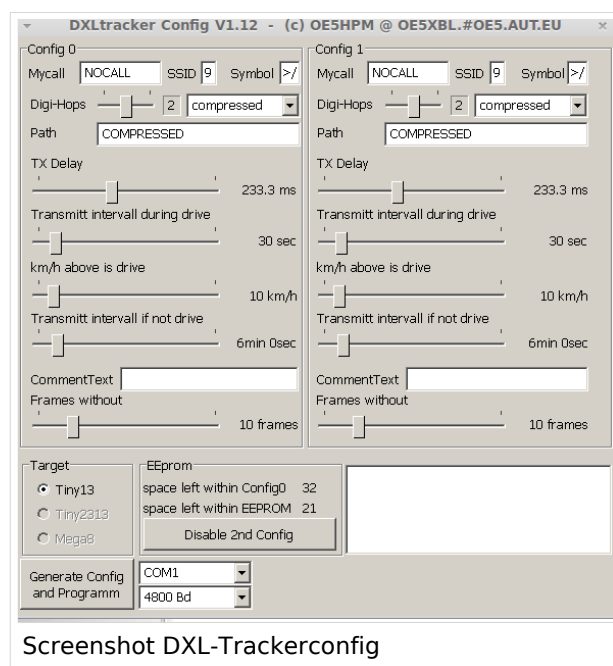
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

DXL - APRTracker: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen

VisuellWikitext

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

Oe1mcu ([Diskussion](#) | [Beiträge](#))

Markierung: [Visuelle Bearbeitung](#)

[← Zum vorherigen Versionsunterschied](#)

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

OE2WAO ([Diskussion](#) | [Beiträge](#))

K

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 16:

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 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

[[Datei:**aprsTracker.zip**|**verweis=Special:FilePath**/aprsTracker.zip]]

+

Der Assemblercode wird im einfachsten
Fall mit dem Compiler 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

-

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

[[[Datei:**dxl-aprsTracker-tiny13-10MHz-4800Bd.zip**|**verweis=Special:FilePath**/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

+

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[[Datei:dxlTracker-schematic.png]]	+ [[[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis	
1 Einleitung	14
2 Software	14
3 Hardware	15
3.1 Trackerschaltung	15
3.2 Geeignete GPS-Empfänger	15

4 Konfiguration	15
5 Referenzprojekte	16
5.1 OE5EEP, Heinz	16
6 Prämiert	17
6.1 LoRa Variante	17
7 Weitere Projekte	17

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

Hersteller	Type	Baudrate	Versorgungsspannung	Stromaufnahme	Preis	
Fastrax	UP501	9600	3.3V	~30mA	ca. 22€ (inkl. Versand)	http://cloi/Se/htt/da/UPpdf
Ublox	NEO-6M	9600	3.3V	~22mA	ca. 10€ (inkl. Versand)	Ubl
Globaltop	GTPA013	9600	3.3V	~20mA	ca. 19€	http://da/FGI/V0/

Konfiguration

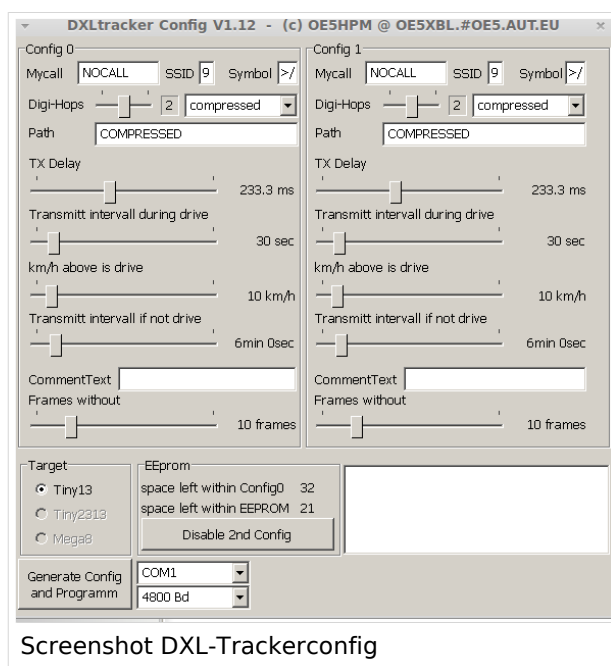
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

DXL - APRTracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

[Oe1mcu](#) ([Diskussion](#) | [Beiträge](#))

Markierung: [Visuelle Bearbeitung](#)

[← Zum vorherigen Versionsunterschied](#)

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

[OE2WAO](#) ([Diskussion](#) | [Beiträge](#))

[K](#)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 16:

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 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

[[Datei:**aprsTracker.zip**|**verweis=Special:FilePath**/aprsTracker.zip]]

+

Der Assemblercode wird im einfachsten
Fall mit dem Compiler 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

-

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

[[[Datei:**dxl-aprsTracker-tiny13-10MHz-4800Bd.zip**|**verweis=Special:FilePath**/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

+

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[Datei:dxlTracker-schematic.png]]	+ [[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis

1	Einleitung	22
2	Software	22
3	Hardware	23
3.1	Trackerschaltung	23
3.2	Geeignete GPS-Empfänger	23

4 Konfiguration	23
5 Referenzprojekte	24
5.1 OE5EEP, Heinz	24
6 Prämiert	25
6.1 LoRa Variante	25
7 Weitere Projekte	25

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

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

Konfiguration

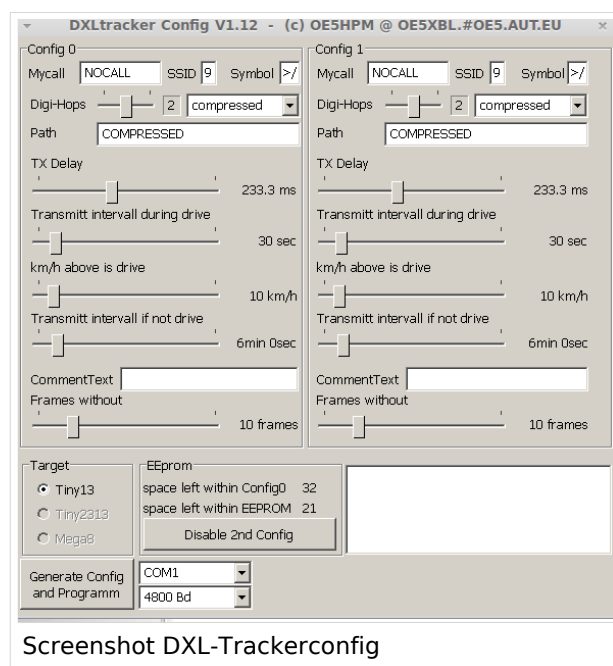
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

DXL - APRTracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

[Oe1mcu](#) ([Diskussion](#) | [Beiträge](#))

Markierung: [Visuelle Bearbeitung](#)

[← Zum vorherigen Versionsunterschied](#)

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

[OE2WAO](#) ([Diskussion](#) | [Beiträge](#))

[K](#)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 16:

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 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

[[Datei:**aprsTracker.zip**|**verweis=Special:FilePath**/aprsTracker.zip]]

+

Der Assemblercode wird im einfachsten
Fall mit dem Compiler 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

-

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

[[[Datei:**dxl-aprsTracker-tiny13-10MHz-4800Bd.zip**|**verweis=Special:FilePath**/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

+

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[[Datei:dxlTracker-schematic.png]]	+ [[[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis	
1 Einleitung	30
2 Software	30
3 Hardware	31
3.1 Trackerschaltung	31
3.2 Geeignete GPS-Empfänger	31

4 Konfiguration	31
5 Referenzprojekte	32
5.1 OE5EEP, Heinz	32
6 Prämiert	33
6.1 LoRa Variante	33
7 Weitere Projekte	33

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

Hersteller	Type	Baudrate	Versorgungsspannung	Stromaufnahme	Preis	
Fastrax	UP501	9600	3.3V	~30mA	ca. 22€ (inkl. Versand)	http://cloi/Se/htt/da/UPpdf
Ublox	NEO-6M	9600	3.3V	~22mA	ca. 10€ (inkl. Versand)	Ubl
Globaltop	GTPA013	9600	3.3V	~20mA	ca. 19€	http://da/FGI/V0/

Konfiguration

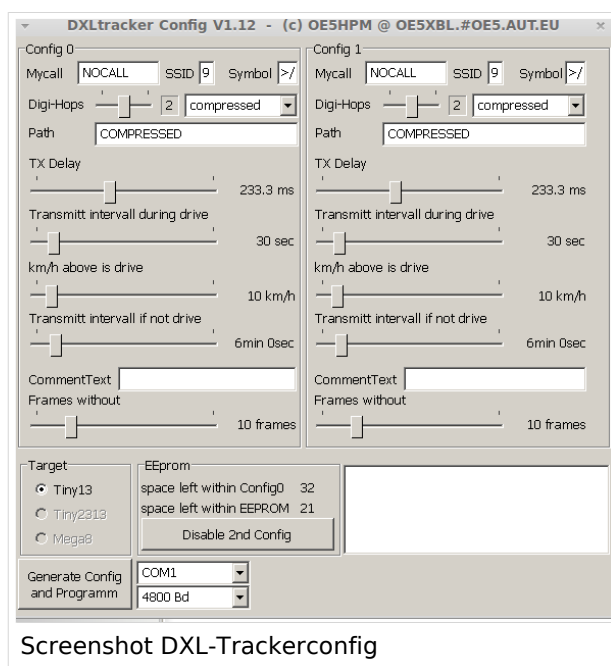
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

DXL - APRTracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

[Oe1mcu](#) ([Diskussion](#) | [Beiträge](#))

Markierung: [Visuelle Bearbeitung](#)

[← Zum vorherigen Versionsunterschied](#)

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

[OE2WAO](#) ([Diskussion](#) | [Beiträge](#))

[K](#)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 16:

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 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrdupe/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

+ `[[Datei:aprsTracker.zip|verweis=Special:FilePath/aprsTracker.zip]]`

Der Assemblercode wird im einfachsten
Fall mit dem Compiler 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrdupe/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

– `[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]`

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

+ `[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip|verweis=Special:FilePath/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]`

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[Datei:dxlTracker-schematic.png]]	+ [[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis

1 Einleitung	38
2 Software	38
3 Hardware	39
3.1 Trackerschaltung	39
3.2 Geeignete GPS-Empfänger	39

4 Konfiguration	39
5 Referenzprojekte	40
5.1 OE5EEP, Heinz	40
6 Prämiert	41
6.1 LoRa Variante	41
7 Weitere Projekte	41

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

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

Konfiguration

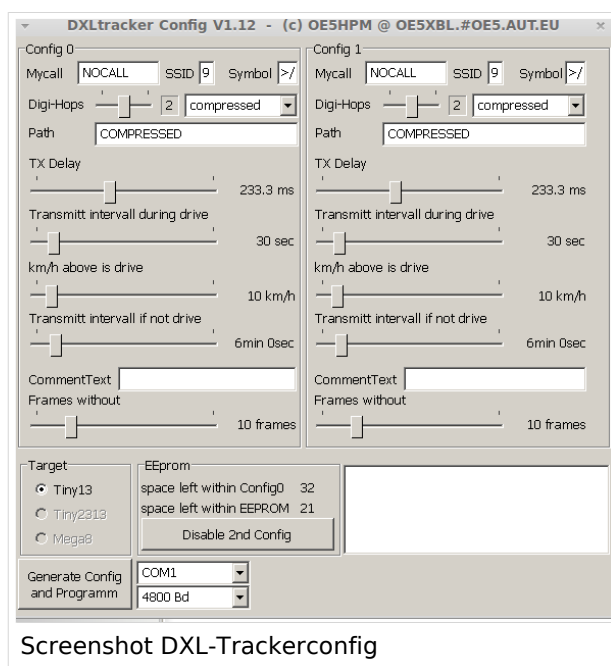
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

DXL - APRTracker: Unterschied zwischen den Versionen

Versionsgeschichte interaktiv durchsuchen

VisuellWikitext

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

Oe1mcu ([Diskussion](#) | [Beiträge](#))

Markierung: Visuelle Bearbeitung

← Zum vorherigen Versionsunterschied

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

OE2WAO ([Diskussion](#) | [Beiträge](#))

K

Markierung: Visuelle Bearbeitung

Zum nächsten Versionsunterschied →

Zeile 16:

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 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

[[Datei:**aprsTracker.zip**|verweis=Special:FilePath/aprsTracker.zip]]

+

Der Assemblercode wird im einfachsten
Fall mit dem Compiler 'gavrasn'
[http://www.avr-asm-tutorial.net/gavrasn/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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

-

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

[[[Datei:**dxl-aprsTracker-tiny13-10MHz-4800Bd.zip**|verweis=Special:FilePath/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]

+

|-

|ATMEL Tiny13

Zeile 38:

|9600

|1200

- [[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]

|-

|}

Zeile 38:

|9600

|1200

+

[[Datei:**dxl-aprsTracker-tiny13-10MHz-9600Bd.zip**|**verweis=Special:FilePath**/dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]

|-

|}

Zeile 50:

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

- [[Datei:dxlTracker-schematic.png]]

Detaillierte Schaltungsbeschreibung folgt.

Zeile 50:

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

+

[[Datei:**dxlTracker-schematic.png**|**verweis=Special:FilePath**/dxlTracker-schematic.png]]

Detaillierte Schaltungsbeschreibung folgt.

Zeile 120:

==Referenzprojekte==

===OE5EEP, Heinz===

- [[Datei:oe5eep_2.jpg|rechts|mini|hochkant|200px|Gesamtansicht]]

- [[Datei:oe5eep_1.jpg|rechts|mini|200px|Gesamtansicht]]

- [[Datei:oe5eep_3.jpg|rechts|mini|200px|Innenleben]]

Zeile 120:

==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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis	
1 Einleitung	46
2 Software	46
3 Hardware	47
3.1 Trackerschaltung	47
3.2 Geeignete GPS-Empfänger	47

4 Konfiguration	47
5 Referenzprojekte	48
5.1 OE5EEP, Heinz	48
6 Prämiert	49
6.1 LoRa Variante	49
7 Weitere Projekte	49

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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

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

Konfiguration

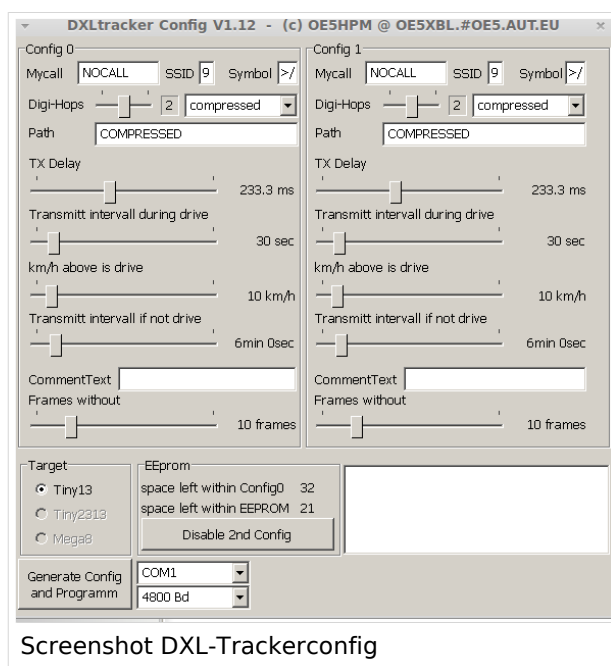
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.

Dateiversionen

Klicken Sie auf einen Zeitpunkt, um diese Version zu laden.

	Version vom	Maße	Benutzer	Kommentar
aktuell	20:44, 29. Sep. 2013	(48 KB)	OE5HPM (Diskussion) (Beiträge)	(V1.15 - SSID with more than 2 digits didn't work properly)
	21:21, 3. Sep. 2013	(48 KB)	OE5HPM (Diskussion) (Beiträge)	V1.14 - calls with len < 6 digits didn't work properly
	21:00, 28. Aug. 2013	(48 KB)	OE5HPM (Diskussion) (Beiträge)	V1.12 - letzte Releaseversion

Sie können diese Datei nicht überschreiben.

Dateiverwendung

Die folgende Seite verwendet diese Datei:

- [DXL - APRTracker](#)

DXL - APRTracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

Version vom 16. März 2021, 13:12 Uhr (Quelltext anzeigen)

[Oe1mcu](#) ([Diskussion](#) | [Beiträge](#))

Markierung: [Visuelle Bearbeitung](#)

[← Zum vorherigen Versionsunterschied](#)

Version vom 27. August 2023, 12:59 Uhr (Quelltext anzeigen)

[OE2WAO](#) ([Diskussion](#) | [Beiträge](#))

[K](#)

Markierung: [Visuelle Bearbeitung](#)

[Zum nächsten Versionsunterschied →](#)

Zeile 16:

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 '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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 16:

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

+ `[[Datei:aprsTracker.zip|verweis=Special:FilePath/aprsTracker.zip]]`

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/avrduide/>]) und dem USBasp Programmer
[<http://www.fischl.de/usbasp/>]). Es ist auch
möglich mit dem von ATMEL angebotenen
AVR-Studio den Code zu übersetzen.

Zeile 32:

|4800

|1200

– `[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]`

|-

|ATMEL Tiny13

Zeile 32:

|4800

|1200

+ `[[[Datei:dxl-aprsTracker-tiny13-10MHz-4800Bd.zip|verweis=Special:FilePath/dxl-aprsTracker-tiny13-10MHz-4800Bd.zip]]`

|-

|ATMEL Tiny13

Zeile 38:	
9600	9600
1200	1200
- [[[Datei:dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]	+ [[[Datei: dxl-aprsTracker-tiny13-10MHz-9600Bd.zip verweis=Special:FilePath /dxl-aprsTracker-tiny13-10MHz-9600Bd.zip]]
-	-
}	}
Zeile 50:	
Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:	Schaltungsvorschlag von OE5HPM mit einem Tiny13 (kleinster Prozessor), wie er bereits mehrfach im Einsatz ist:
- [[[Datei:dxlTracker-schematic.png]]	+ [[[Datei: dxlTracker-schematic.png verweis=Special:FilePath /dxlTracker-schematic.png]]
Detaillierte Schaltungsbeschreibung folgt.	Detaillierte Schaltungsbeschreibung folgt.
Zeile 120:	
==Referenzprojekte==	==Referenzprojekte==
===OE5EEP, Heinz===	===OE5EEP, Heinz===
- [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht]]	+ [[[Datei:oe5eep_2.jpg rechts mini hochkant 200px Gesamtansicht verweis=Special:FilePath /oe5eep_2.jpg]]
- [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht]]	+ [[[Datei:oe5eep_1.jpg rechts mini 200px Gesamtansicht verweis=Special:FilePath /oe5eep_1.jpg]]
- [[[Datei:oe5eep_3.jpg rechts mini 200px Innenleben]]	+ [[[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]]	+	[[Datei:oe5eep_4h.jpg rechts mini 200px Innenleben-Detail-Tracker+GPS-RX verweis=Special:FilePath/oe5eep_4h.jpg]]
Zeile 147:	Der DXL APRStracker wurde mit dem € 2000,- dotierten ÖVSV Innovationspreis 2013 prämiert ! GRATULATION !!	Zeile 147:	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==		==Weitere Projekte==

Version vom 27. August 2023, 12:59 Uhr

Inhaltsverzeichnis

1	Einleitung	55
2	Software	55
3	Hardware	56
3.1	Trackerschaltung	56
3.2	Geeignete GPS-Empfänger	56

4 Konfiguration	56
5 Referenzprojekte	57
5.1 OE5EEP, Heinz	57
6 Prämiert	58
6.1 LoRa Variante	58
7 Weitere Projekte	58

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 GPGLA), 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:

Datei: [dxlTracker-schematic.png](#)

Detaillierte Schaltungsbeschreibung folgt.

Geeignete GPS-Empfänger

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

Konfiguration

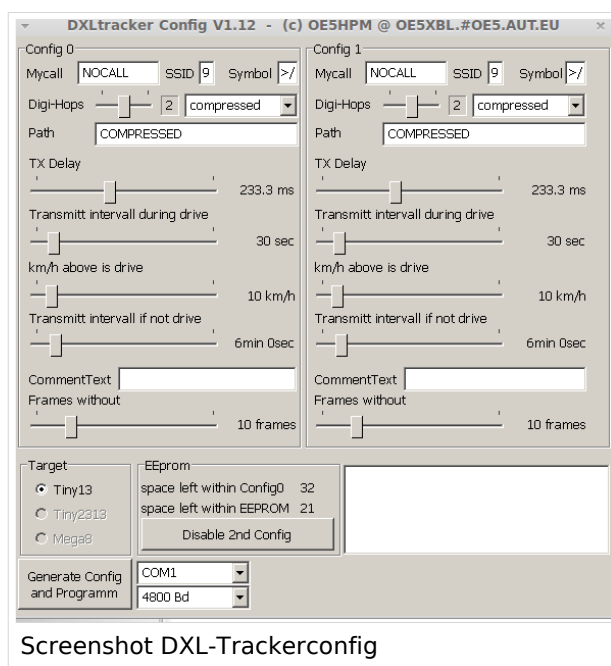
GPS und Config-Programm liefern die seriellen Daten in TTL- oder RS232-Pegel. Der Pegel wird mittels einem (10k) Widerstand und der im Prozessor eingebauten Schutzdiode angepasst, die Polarisation erkennt die Software automatisch.

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.



Screenshot DXL-Trackerconfig

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](#)

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.

Datei:oe5eep 2.jpg
Gesamtansicht

Datei:oe5eep 1.jpg
Gesamtansicht

Datei:oe5eep 3.jpg
Innenleben

Datei:oe5eep 4h.jpg
Innenleben-Detail-
Tracker+GPS-RX

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 [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.