

DXL - APRStracker

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 28. August 2013, 15:50 Uhr
([Quelltext anzeigen](#))
[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))
(APRS Tracker, Schaltung, Sourcecode)
[← Zum vorherigen Versionsunterschied](#)

Version vom 28. August 2013, 21:17 Uhr
([Quelltext anzeigen](#))
[OE5HPM](#) ([Diskussion](#) | [Beiträge](#))
[Zum nächsten Versionsunterschied →](#)

Zeile 1:

[[Kategorie:APRS]]

[[Kategorie:Selbstbau]]

Zeile 1:

[[Kategorie:APRS]]

[[Kategorie:Selbstbau]]

+

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 erhebliche Vorteile (QSB) bringt.

+

Zu dem wird das schon etwas in die Jahre gekommene aber durchaus am effektivsten SSID-Pathrouting (im Configtool 'COMPRESSED' genannt) unterstützt.

-

Hier entsteht die Seite zum Hard- bzw. Softwareprojekt um APRS von OE5DXL

+

Als Minimum 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 wählbar ca. 6..20MHz (Im Source eingeben).

+

* PTT-Transistor

	+	* RC-Tiefpass zum Wegfiltern der PWM-Frequenz benötigt.
- Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode,	+	Opensource Software von OE5DXL, in Assembler geschrieben, welche den Tracker zum Leben erweckt:
- Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann		
- durch einen passiven Tiefpass statt dem OP.		
- [[Media:tracker.zip Source,Config, Schaltungsvariante]]	+	[[Datei:aprsTracker.zip]]
	+	
	+	Der Assemblercode wird im einfachsten Fall mittels AVR-Studio kompiliert und anschließend in den µC gebrannt.
	+	
	+	In Kürze werden hier auch noch fertig kompilierte Varianten für unterschiedliche Baudraten und Prozessoren hochgeladen.
	+	
	+	----
Als Input dienen serielle GPS-Daten (GPRMC und GPGGA).		Als Input dienen serielle GPS-Daten (GPRMC und GPGGA).
- Output ist (nebst PTT) AFSK APRS-Mic-e Format mit Position,		
- Geschwindigkeit, Fahrtrichtung und Höhe. Dabei sind (pro Profil)		
- 2 Bakenzeiten und eine Geschwindigkeit einstellbar unter /ueber der		

– langsam/schnell gebakt wird.

Als Hardware wird ein ATTiny13, ATTiny2313, Atmega88 oder mit IO-Pins

anpassen auch andere Typen mit (Quarz)-Takt, PTT-Transistor und einem

RC-Tiefpass zum wegfiltern der PWM-Frequenz benötigt.

Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).

Output ist (nebst PTT) AFSK APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Höhe.

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

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

[[Datei:dxlTracker-schematic.png]]

Detaillierte Schaltungsbeschreibung folgt.

GPS und Config-Programm **können** die seriellen Daten in TTL- oder RS232-

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**.

Pegel **liefern**. Der Pegel wird **mit** einem (10k) Widerstand und **den Schutzdioden**

im Prozessor angepasst, die Polarisation erkennt die Software. **Baud (je nach**

Takt) 300..200000, üblich 4800, 9600

-	AFSK-Baud und Tonfrequenzen (Shift) frei konfigurierbar, ueblich : 300(kw),	+	Baudraten vom GPS bzw. PC zur Konfiguration sind je nach Prozessortakt im Bereich 300.. 200000, ueblich 4800, 9600.
-	1200(ukw) Zum Mic-Pegel anpassen eignet sich ein Poti, fuer Handfunk-PTT	+	
-	Koppel-C und 4k7 zum PTT-Fet /Transistor.	+	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 Jumper/Schalter zwischen 2 (Tiny13) oder 4 User-Profilen	+	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.
-	ausgewählt werden, zB. Fahrrad/Auto.	+	
-	An einer Blink-Led fuer korrektem GPS-Empfang am Prozessor Pin fuer	+	
-	GPS-Rx (ohne diese) wird gearbeitet.	+	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.png]]

[[Datei:DXLtrackerConfig.zip]]

Version vom 28. August 2013, 21:17 Uhr

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 erhebliche Vorteile (QSB) bringt. Zu dem wird das schon etwas in die Jahre gekommene aber durchaus am effektivsten SSID-Pathrouting (im Configtool 'COMPRESSED' genannt) unterstützt.

Als Minimum 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 wählbar ca. 6..20MHz (Im Source eingeben).
- PTT-Transistor
- RC-Tiefpass zum wegfiltern der PWM-Frequenz benötigt.

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

Datei:aprsTracker.zip

Der Assemblercode wird im einfachsten Fall mittels AVR-Studio kompiliert und anschließend in den µC gebrannt.

In Kürze werden hier auch noch fertig kompilierte Varianten für unterschiedliche Baudraten und Prozessoren hochgeladen.

Als Input dienen serielle GPS-Daten (GPRMC und GPGGA).

Output ist (nebst PTT) AFSK APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Hoehe.

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

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

Datei:dxlTracker-schematic.png

Detaillierte Schaltungsbeschreibung folgt.

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 is 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 ausgewaehlt 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.

DXLtracker Config V1.12 - (c) OE5HPM @ OE5XBL.#OE5.AUT.EU

Config 0

Mycall SSID Symbol

Digi-Hops

Path

TX Delay

Transmitt intervall during drive

km/h above is drive

Transmitt intervall if not drive

CommentText

Frames without

Config 1

Mycall SSID Symbol

Digi-Hops

Path

TX Delay

Transmitt intervall during drive

km/h above is drive

Transmitt intervall if not drive

CommentText

Frames without

Target

☒ Tiny13
☐ Tiny2313
☐ Mega8

EEprom

space left within Config0 32
space left within EEPROM 21

[Datei:DXLtrackerConfig.zip](#)