DXL - APRStracker

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

Version vom 14. September 2013, 21:50 Uhr (Quelltext anzeigen) OE5HPM (Diskussion | Beiträge) K (Link zu Programmer hinzu) ← Zum vorherigen Versionsunterschied	**Version vom 14. September 2013, 21:51 Uhr (Quelltext anzeigen)** OE5HPM (Diskussion | Beiträge) K Zum nächsten Versionsunterschied →
Zeile 17:	**Zeile 17:**
[[Datei:aprsTracker.zip]]	[[Datei:aprsTracker.zip]]
− Der Assemblercode wird im einfachsten Fall mit dem Compiler 'gavrasm' [http://www.avr-asm-tutorial.net/gavrasm/index_de.html Gerd's AVR Assembler], welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt (z.B.: mit AVRdude [http://www.nongnu.org/avrdude/] und dem USBasp Programmer [http://www.fischl.de/usbasp/]. Es ist auch möglich mit dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.	+ Der Assemblercode wird im einfachsten Fall mit dem Compiler 'gavrasm' [http://www.avr-asm-tutorial.net/gavrasm/index_de.html Gerd's AVR Assembler], welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt (z.B.: mit AVRdude [http://www.nongnu.org/avrdude/] und dem USBasp Programmer [http://www.fischl.de/usbasp/]). Es ist auch möglich mit dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.
{| class="wikitable sortable"	{| class="wikitable sortable"

Version vom 14. September 2013, 21:51 Uhr

Inhaltsverzeichnis

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, Atmea88 oder auch andere Typen mit (Quarz)-Takt durch Anpassen der I/O Pins im Source
- Quarz frei waehlbar ca. 6..20MHz (Im Source eingeben).
- PTT-Transistor
- RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt.

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.

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 GPGGA), der Tracker generiert daraus anhand der programmierten Konfiguration als Output (nebst PTT) die AFSK-Moduluation im APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Hoehe, welche dem Funkgerät zugeführt wird.

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

Hardware

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

Datei:dxlTracker-schematic.png

Detailierte Schaltungsbeschreibung folgt.

Konfiguration

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

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

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

- 300 auf Kurzwelle
- 1200 auf UKW

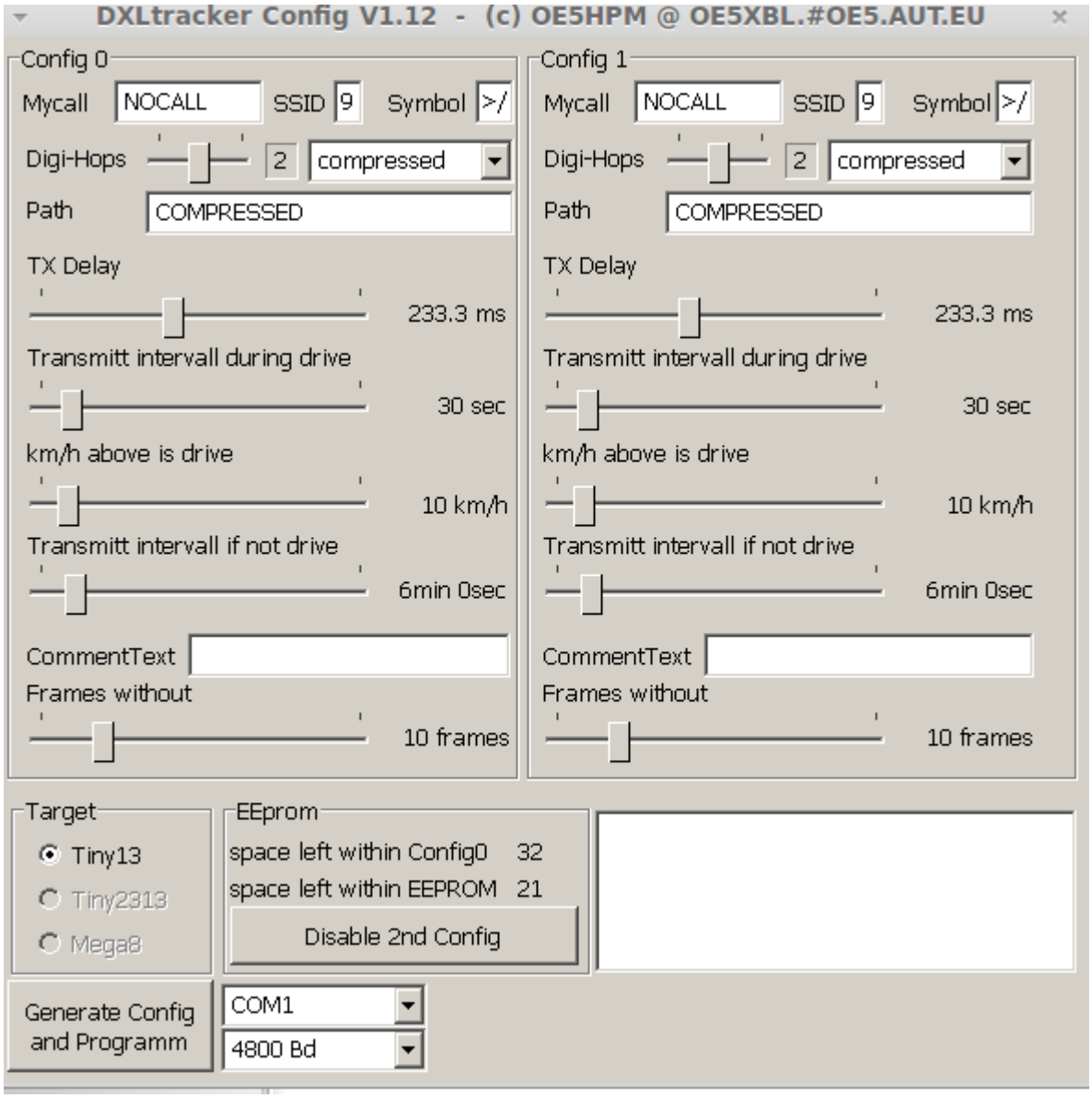
Zur Einstellung von HUB bzw. Mikrofonpegel eignet sich am besten ein Poti.

Um die PTT von Handfunkgeräten aufzutasten muss der Mikrofoneingang mit dem PTT Transistor kombiniert werden. Einfachste Variante 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.



Datei:DXLtrackerConfig.zip