

DXL - APRStracker

Versionsgeschichte interaktiv durchsuchen
VisuellWikitext

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

Zeile 2:

[[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 **erheblich e 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).

Zeile 15:

[[Datei:aprsTracker.zip]]

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

Version vom 28. August 2013, 21:56 Uhr
(**Quelltext anzeigen**)
OE5HPM ([Diskussion](#) | [Beiträge](#))
(Compiler für ASM angeführt)
Zum nächsten Versionsunterschied →

Zeile 2:

[[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 (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 wählbar ca. 6..20MHz (Im Source eingeben).

Zeile 15:

[[Datei:aprsTracker.zip]]

Der Assemblercode wird im einfachsten Fall **mit dem Compiler 'gavasm'** ([\http://www.avr-asm-tutorial.net

| | |
|--|--|
| - | + /qavrasm/index de.html], welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt. Es ist auch möglich dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen. |
| | |
| In Kürze werden hier auch noch fertig kompilierte Varianten für unterschiedliche Baudraten und Prozessoren hochgeladen. | In Kürze werden hier auch noch fertig kompilierte Varianten für unterschiedliche Baudraten und Prozessoren hochgeladen. |
| Zeile 21: | Zeile 21: |
| ---- | ---- |
| | |
| - Als Input dienen serielle GPS-Daten (GPRMC und GPGBGA). | + Als Input dienen serielle GPS-Daten (GPRMC und GPGBGA), der Tracker generiert daraus anhand der programmierten Konfiguration als Output (nebst PTT) die AFSK-Modulation im APRS-Mic-e Format mit Position, Geschwindigkeit, Fahrtrichtung und Hoehe, welche dem Funkgerät zugeführt wird. |
| - | + |
| - 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. | Dabei sind (pro Profil) 2 Bakenzeiten und eine Geschwindigkeit einstellbar unter /ueber der langsam/schnell gebakt wird. |

Version vom 28. August 2013, 21:56 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 (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.

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' ([\[1\]](#)), welcher sowohl für Linux als auch für Windows verfügbar ist, kompiliert und anschließend in den µC gebrannt. Es ist auch möglich dem von ATMEL angebotenen AVR-Studio den Code zu übersetzen.

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

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

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 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.

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

Generate Config and Programm

[Datei:DXLtrackerConfig.zip](#)