

Inhaltsverzeichnis

1. DXL - APRStracker .....	6
2. Benutzer:Oe5dxl .....	4

## DXL - APRStracker

[Versionsgeschichte interaktiv durchsuchen](#)  
[VisuellWikitext](#)

**Version vom 28. August 2013, 15:34 Uhr**  
([Quelltext anzeigen](#))

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))  
(APRS Tracker, Schaltung, Sourcecode)  
[← Zum vorherigen Versionsunterschied](#)

**Zeile 3:**

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

Da an einer "richtigen" Doku noch  
gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine  
Schaltungsvariante, die man vereinfachen  
kann

**Zeile 8:**

[[Media:tracker.zip|Source,Config,  
Schaltungsvariante]]

Als Hardware wird ein ATTiny13,  
ATTiny2313, Atmea88 oder mit IO-Pins

**Version vom 28. August 2013, 15:50 Uhr**  
([Quelltext anzeigen](#))

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))  
(APRS Tracker, Schaltung, Sourcecode)  
[Zum nächsten Versionsunterschied →](#)

**Zeile 3:**

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

Da an einer "richtigen" Doku noch  
gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine  
Schaltungsvariante, die man vereinfachen  
kann

**Zeile 9:**

[[Media:tracker.zip|Source,Config,  
Schaltungsvariante]]

**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, Atmea88 oder mit IO-Pins

<b>Zeile 14:</b>	<b>Zeile 21:</b>
Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).	Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).
– GPS und Config-Programm koennen die seriellen Daten in TTL oder RS232-	+ GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232-
Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden	Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden
im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach	im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach
– Takt) 300..200000.	+ Takt) 300..200000, <b>ueblich 4800, 9600.</b>
AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),	AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),

## Version vom 28. August 2013, 15:50 Uhr

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode, Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann durch einen passiven Tiefpass statt dem OP.

### Source,Config,Schaltungvariante

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

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins anpassen auch andere Typen mit (Quarz)-Takt, PTT-Transistor und einem RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt. Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).

GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232- 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, ueblich 4800, 9600.

AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw), 1200(ukw) Zum Mic-Pegel anpassen eignet sich ein Poti, fuer Handfunk-PTT Koppel-C und 4k7 zum PTT-Fet/Transistor.

Als Option kann mit Jumper/Schalter zwischen 2 (Tiny13) oder 4 User-Profilen ausgewaehlt werden, zB. Fahrrad/Auto. An einer Blink-Led fuer korrektem GPS-Empfang am Prozessor Pin fuer GPS-Rx (ohne diese) wird gearbeitet.

## DXL - APRStracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

**Version vom 28. August 2013, 15:34 Uhr**

**(Quelltext anzeigen)**

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))

(APRS Tracker, Schaltung, Sourcecode)

[← Zum vorherigen Versionsunterschied](#)

**Version vom 28. August 2013, 15:50 Uhr**

**(Quelltext anzeigen)**

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))

(APRS Tracker, Schaltung, Sourcecode)

[Zum nächsten Versionsunterschied →](#)

**Zeile 3:**

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann

**Zeile 8:**

[[Media:tracker.zip|Source,Config, Schaltungsvariante]]

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins

**Zeile 3:**

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann

**Zeile 9:**

[[Media:tracker.zip|Source,Config, Schaltungsvariante]]

**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.**

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins

<b>Zeile 14:</b>	<b>Zeile 21:</b>
Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).	Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).
– GPS und Config-Programm koennen die seriellen Daten in TTL oder RS232-	+ GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232-
Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden	Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden
im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach	im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach
– Takt) 300..200000.	+ Takt) 300..200000, <b>ueblich 4800, 9600.</b>
AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),	AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),

## Version vom 28. August 2013, 15:50 Uhr

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode, Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann durch einen passiven Tiefpass statt dem OP.

### Source,Config,Schaltungvariante

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

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins anpassen auch andere Typen mit (Quarz)-Takt, PTT-Transistor und einem RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt. Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).

GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232- 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, ueblich 4800, 9600.

AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw), 1200(ukw) Zum Mic-Pegel anpassen eignet sich ein Poti, fuer Handfunk-PTT Koppel-C und 4k7 zum PTT-Fet/Transistor.

Als Option kann mit Jumper/Schalter zwischen 2 (Tiny13) oder 4 User-Profilen ausgewaehlt werden, zB. Fahrrad/Auto. An einer Blink-Led fuer korrektem GPS-Empfang am Prozessor Pin fuer GPS-Rx (ohne diese) wird gearbeitet.

## DXL - APRStracker: Unterschied zwischen den Versionen

[Versionsgeschichte interaktiv durchsuchen](#)

[VisuellWikitext](#)

**Version vom 28. August 2013, 15:34 Uhr**

**(Quelltext anzeigen)**

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))

(APRS Tracker, Schaltung, Sourcecode)

[← Zum vorherigen Versionsunterschied](#)

**Version vom 28. August 2013, 15:50 Uhr**

**(Quelltext anzeigen)**

[Oe5dxl](#) ([Diskussion](#) | [Beiträge](#))

(APRS Tracker, Schaltung, Sourcecode)

[Zum nächsten Versionsunterschied →](#)

**Zeile 3:**

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann

**Zeile 8:**

[[Media:tracker.zip|Source,Config, Schaltungsvariante]]

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins

**Zeile 3:**

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode,

Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann

**Zeile 9:**

[[Media:tracker.zip|Source,Config, Schaltungsvariante]]

**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.**

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins

<b>Zeile 14:</b>	<b>Zeile 21:</b>
Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).	Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).
– GPS und Config-Programm koennen die seriellen Daten in TTL oder RS232-	+ GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232-
Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden	Pegel liefern. Der Pegel wird mit einem (10k) Widerstand und den Schutzdioden
im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach	im Prozessor angepasst, die Polarisation erkennt die Software. Baud (je nach
– Takt) 300..200000.	+ Takt) 300..200000, <b>ueblich 4800, 9600.</b>
AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),	AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw),

## Version vom 28. August 2013, 15:50 Uhr

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

Da an einer "richtigen" Doku noch gearbeitet wird, hier mal Sourcecode, Konfigurations-Tool und eine Schaltungsvariante, die man vereinfachen kann durch einen passiven Tiefpass statt dem OP.

### Source,Config,Schaltungvariante

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

Als Hardware wird ein ATTiny13, ATTiny2313, Atmea88 oder mit IO-Pins anpassen auch andere Typen mit (Quarz)-Takt, PTT-Transistor und einem RC-Tiefpass zum wegfiltern der PWM-Frequenz benoetigt. Quarz frei wählbar ca. 6..20MHz (Im Source eingeben).

GPS und Config-Programm koennen die seriellen Daten in TTL- oder RS232- 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, ueblich 4800, 9600.

AFSK-Baud und Tonfrequenzen (Shift) frei Konfigurierbar, ueblich: 300(kw), 1200(ukw) Zum Mic-Pegel anpassen eignet sich ein Poti, fuer Handfunk-PTT Koppel-C und 4k7 zum PTT-Fet/Transistor.

Als Option kann mit Jumper/Schalter zwischen 2 (Tiny13) oder 4 User-Profilen ausgewaehlt werden, zB. Fahrrad/Auto. An einer Blink-Led fuer korrektem GPS-Empfang am Prozessor Pin fuer GPS-Rx (ohne diese) wird gearbeitet.