

WELCOME TO

**NORD > < LINK**

**.ORG**

Verein zur Förderung der digitalen  
Betriebsarten im Amateurfunk  
<http://www.nordlink.org>

## LoRa APRS

Lilygo T-ECHO als Tracker und mobile Wetterstation

Karsten Heddenhausen DC7OS  
[dc7os@nordlink.org](mailto:dc7os@nordlink.org)

Stand: 05.04.2023

## Der Lilygo T-ECHO

Der T-ECHO ist ein relativ neues Gerät. Man kann ihn quasi als Weiterentwicklung des schon vorgestellten T-Beam vom gleichen Hersteller ansehen. Gegenüber diesem unterscheidet er sich nicht nur auf Grund des höheren Preises.

- Der T-ECHO hat gleich einen Lithium-Polymer Akku mit verbaut.
- Er wird mit Gehäuse geliefert. Dieses ist auch noch deutlich kleiner.
- Die mitgelieferte Antenne passt sehr viel besser in das 70cm Band.
- Es ist ein stromsparendes E-Ink Display verbaut.
- Zum Anschluss an ein Netzteil oder zum Programmieren ist ein USB-C Anschluss verbaut.
- Er ist optional, und dann natürlich gegen Aufpreis, mit einem BME280 Sensor zur Messung von Temperatur, Luftfeuchtigkeit und Luftdruck erhältlich.



## Der Wettersensor BM280



Wie schon zuvor geschrieben, ist der T-ECHO mit optionalem BME280 erhältlich. Dieser ist dann fest auf der Platine verlötet und damit auch im Gehäuse eingebaut. Damit hat man den Nachteil, dass der Sensor nicht wirklich die Umgebungswerte misst, sondern die Werte innerhalb des Gehäuses. Das ist nicht unbedingt gewollt. Allerdings kann man auch etwas basten. Es gibt eine steckbare Platine, die es einem ermöglicht, diesen Sensor extern nachzurüsten.

Diese Nachrüstung hat den Vorteil, dass man den Sensor, oder einen Anschluss-Stecker dafür nach außen führen kann, um den Sensor dann mit einem Kabel anzubinden.

## Die benötigte Software

Das Angebot an Software ist längst nicht so breit gefächert, wie das für den T-Beam. Ich verwende die Software von Thierry F4EWI, die auf einem Projekt von Fabrice F4AVI basiert.

Zunächst benötigt man die Arduino IDE<sup>1</sup> und so einige zusätzliche Bibliotheken<sup>2</sup>. Sehr gut wurde das anhand des Programmes von F4AVI in einem Video<sup>3</sup> von Manuel DO3MLA beschrieben.

Nach der Installation der Arduino IE ist als erstes in der Board-Verwaltung „Adafruit nRF52“ auszuwählen, welches den nRF52840 Chipsatz unterstützt.



Das Programm von Thierry F4EWI ist auf GITHUB<sup>4</sup> zu finden. Wenn man das Programm in der IE öffnet, sollten alle anderen Dateien auch angezeigt werden, insbesondere die config.h. Die notwendigen Änderungen sind recht schnell zu erkennen. Vor dem Kompilieren und Hochladen auf das Gerät ist auf jeden Fall das Board „Nordic nRF52840 DK“ und die entsprechende Schnittstelle auszuwählen.

Etwas anders sieht es bei der Software von Thomas DL5TKL aus. Man kann sie auch bei GITHUB downloaden<sup>5</sup>. Da ein Kompilieren nur mit dem gcc möglich sein soll, habe ich gar nicht erst probiert, dieses mit irgendwelchen der anderen Tools zu erledigen. Stattdessen sollte man eines der fertigen Abbilder<sup>6</sup> installieren.

Aber Vorsicht: Es wird dabei auch ein neuer Bootloader installiert. Man sollte also vorher das vorhandene Abbild sichern. Dazu verbindet man das Gerät über ein USB-Kabel mit dem Computer und drückt man zweimal kurz hintereinander auf den Knopf oben links. Dann sollte ein weiteres Laufwerk mit dem Namen TECHOBOOT eingebunden werden. Dort gibt es eine Datei Current.uf2, diese ist zu sichern und durch eines der Abbilder der Software zu ersetzen. Möchte man späte eine andere Software installieren, ist auf gleichem Wege wieder die eben erstellte Sicherung zurück zu kopieren.

Die Konfiguration des Programmes kann zum Teil in der Software selbst gemacht werden. Aber für viele Sachen wird ein externes Programm benötigt. Dieses findet man, wie so vieles, im Internet<sup>7</sup>. Wie und wo man was eintragen muss, ist in der ReadMe.md\_ des Programmes beschrieben. Allerdings müssen alle Zeichen zuvor in ihre hexadezimalen Werte umgewandelt werden.

```

5/6/7 TX RX
<<< Exit
Receiver off
Tracker on
GNSS Warmup on
TX Power > +10 dBm
Symbol > /b
Info >

```

Wenn man dieses geschafft hat, kann man in der Software den Rest einstellen, wie man es benötigt. In den Einstellungen kann man insbesondere den Empfänger aus und den Sender einschalten. Letzteres geht nur, wenn man zuvor ein Rufzeichen eingegeben hat. Die Einstellung „GNSS Warmup“ sollte eingeschaltet sein, es speichert die Satellitendaten zum schnelleren Fix. Die Einstellung „TX Power“ sollte eigentlich selbsterklärend sein. Hier muss jeder für sich eine Einstellung finden.

Im Programm selbst gibt es verschiedene Ansichten, um sich seine Position anzeigen zu lassen. Hier kann sich jeder selbst ansehen, welche Ansicht ihm am meisten zu sagt.



GNSS-Status:

Lat: 42.345001 Alt:

Lon: 11.123000 100

GPS: 3D [auto] Sats: 4

GLO: 2D [auto] Sats: 1

DOP H: 2.0 V: 3.0 P: 1.0

Trk: GP: 4/4, GL: 2/3



DL5TKL-4

Lat: 43.20999

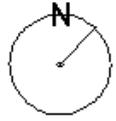
Lon: 12.34000

Alt: 100.0 m

Hello World!

Distance: 138.26 km

Direction: 45.5 deg



<sup>1</sup> <https://www.arduino.cc/en/software>

<sup>2</sup> <https://github.com/Xinyuan-LilyGO/T-Echo>

<sup>3</sup> <https://www.youtube.com/watch?v=O4ZPJyB1o4>

<sup>4</sup> <https://github.com/F4EWI/LoRa-TEcho-APRSTracker>

<sup>5</sup> <https://github.com/cfr34k/t-echo-lora-aprs>

<sup>6</sup> <https://github.com/cfr34k/t-echo-lora-aprs/releases>

<sup>7</sup> nRF Connect for Mobile - nordicsemi.com