



## LTraX – ein „Maker-Projekt“

Ultra-Low-Power Datalogger mit GPS/LTE/2G und Bluetooth 5

### Motivation

Ein Verwandter von mir betreibt einen Anhängerverleih (kleine PKW-Anhänger als „Mobile Werbefläche“ [1]) und die Fahrzeuge werden oft für mehrere Jahre verliehen. Wäre also gut zu wissen, wo sie sich jeweils befinden. Das Problem an der Sache ist, dass Anhänger keine eigene Stromversorgung besitzen. Ein geeignetes Gerät sollte also für mehrere Jahre mit eigener Batterie die Position übermitteln können. Einige kommerziell angebotene Lösungen haben wir auch getestet, das Ergebnis war absolut ernüchternd: sämtliche Geräte waren unbrauchbar!

Da ich eh gerade dabei war, mich in Bluetooth-Low-Energy einzuarbeiten, entstand daraus „LTraX“ als Hobby-Projekt (und jemand, der seinem Hobby nachgeht, wird auf Denglisch neuerdings als „Maker“ [2] bezeichnet...).

### Ergebnis

Es zeigte sich schnell, dass – mit der richtigen Plattform und den richtigen Werkzeugen – sich problemlos wirklich beeindruckende und absolut professionelle Resultate erzielen lassen. Für Fragen, Anregung, etc., freue ich mich über Ihre Mail, meine Kontaktdaten finden sich am Ende des Artikels [3].

## Plattform mit BLE

Bluetooth galt ja über lange Zeit als wenig geeignet für Ultra-Low-Power-Anwendungen. Doch spätestens seit Bluetooth-Low-Energy (auch als „BLE“ bezeichnet) stimmt dies absolut nicht mehr! Und seit BLE 5 sind auch die erzielbaren Reichweiten hervorragend.

Inzwischen sind auch fast alle aktuellen Browser (Chrome, Opera, Edge, etc...) in der Lage, direkt per HTML/Javascript mit BLE-Geräten zu kommunizieren! Und das funktioniert selbst mit lokalen HTML-Seiten und auch egal, ob diese auf PC (ab Windows 10) oder auf einem Android-Gerät laufen! Das ist wirklich sehr bequem und einfacher geht es wirklich kaum noch!

Auch Javascript ist inzwischen eine sehr mächtige Sprache geworden und seit Google's V8-Engine oft schneller als vergleichbare C++-Programme. Javascript kann aber nicht nur für Browser verwendet werden, es läuft auch perfekt auf Servern, kann mit Datenbanken kommunizieren und Nachrichten verschicken.

Ich rechne fest damit, dass BLE, insbesondere in Kombination mit Javascript, eine DER Schlüsseltechnologien für lokale Kommunikation der nächsten Dekade werden dürfte!

## Hardware

Bei Geräten mit BLE empfiehlt es sich unbedingt, auf zertifizierte Module renommierter Anbieter zurückzugreifen! Ich habe eine Reihe von zertifizierten und unsertifizierten Modulen getestet und es gab einige Aha-Erlebnisse. Und selbst für ein Hobby-Projekt würde ich unbedingt zu einem hochwertigen Modul raten! In unserem Fall haben wir uns für das NINA-B3 (u-Blox) [4] entschieden: Es ist angenehm klein, verwendet als Kern einen extrem leistungsstarken Nordic nRF52840 mit 64MHz-Cortex-M4F-Core mit Floating-Point-Unit (FPU) und 1MByte Flash und 256kB RAM [8]. Fast sämtliche I/Os sind flexibel belegbar, z.B. als analoger Eingang zum Überwachen der Batteriespannung.

Die Reichweite des NINA-B3 ist in unserem Fall nicht so wichtig, aber die angegebenen 1400 mtr halte ich für durchaus realistisch.

Zur Entwicklung von Software stellt Nordic kostenlos das „Embedded Studio“ von Segger („SES“) [5] zu Verfügung. Auf dem EVAL-Kit zum NINA-B3 befindet sich gleich dazu der passende J-Link-Emulator. Die Software-Entwicklung damit ist sehr flott und macht wirklich Spaß!

Die Auswahl des Netzes für Mobiles Internet ist in Europa aktuell noch etwas problematisch, da in vielen Ländern 2G noch lange die beste Flächenabdeckung haben dürfte, in manchen Ländern aber auch schon (und teilweise sogar ausschließlich) LTE-M1 möglich ist.

Daher wurde als Modem eine SARA-R412 (u-Blox) [6] eingesetzt, Es ist angenehm klein und (fast) weltweit einsetzbar.

Eine besondere Herausforderung war die Embedd Antenne: Aufgrund der unterschiedlichen Frequenzbänder von LTE-M1 (700 MHz) und 2G (in Europa GSM 900/1800) und der angestrebten schmalen Bauform, gibt es hier nur wenig Möglichkeiten. Die eingesetzte Antenne von Fractus RUN mXTEND™ [7] ist recht klein und kann mit einem Anpassungsnetzwerk exakt auf die gewünschten Bereiche gezogen werden.

LTraX ist mit zwei internen Sensoren ausgerüstet: ein Temperatur/Feuchte-Sensor (Sensirion SHT30-DIS [9]) kann feststellen, ob in das Gehäuse Wasser eingedrungen ist: steigt die Innenfeuchte über einen kritischen Wert (z.B. 80 % rH), wird der Anwender benachrichtigt und hat dann noch ausreichend Zeit, sich um das Problem zu kümmern, bevor Schäden durch Kondenswasser entstehen können. Anhand des Temperatur-Wertes kann das System beispielsweise entscheiden, ob Internet-Übertragungen verzögert werden, bis es wieder wärmer ist (Standard SIM-Karten sind oft nur bis -25 °C zugelassen).

Der zweite interne Sensor ist ein kombinierter Beschleunigungs- und Magnetfeld-Sensor LSM303AGR von ST [10]: Er kann sehr stromsparend die 3-dimensionale Lage des Gerätes im Raum messen, dazu Beschleunigungen (inklusive einer Bewegungs- und Freifall-Detektion) und auch ein externes Magnetfeld. Das Magnetfeld kann das Erdmagnetfeld sein, oder aber auch ein in der Nähe angebrachter Magnet, der damit z.B. Sabotagesensor-Funktion hat.

Als Schnittstelle zu externen Sensoren ist LTraX mit einem universell programmierbaren, 4-poligen Interface ausgerüstet. Treiber für I<sup>2</sup>C, 1-Draht-Schnittstellen (diverse Temperatsensoren), Zähler, etc., sind vorhanden (rein technologisch sind bis zu 4 dieser Schnittstellen möglich und in den Treibern vorgesehen und bis zu 100 Sensorkanälen sind möglich).

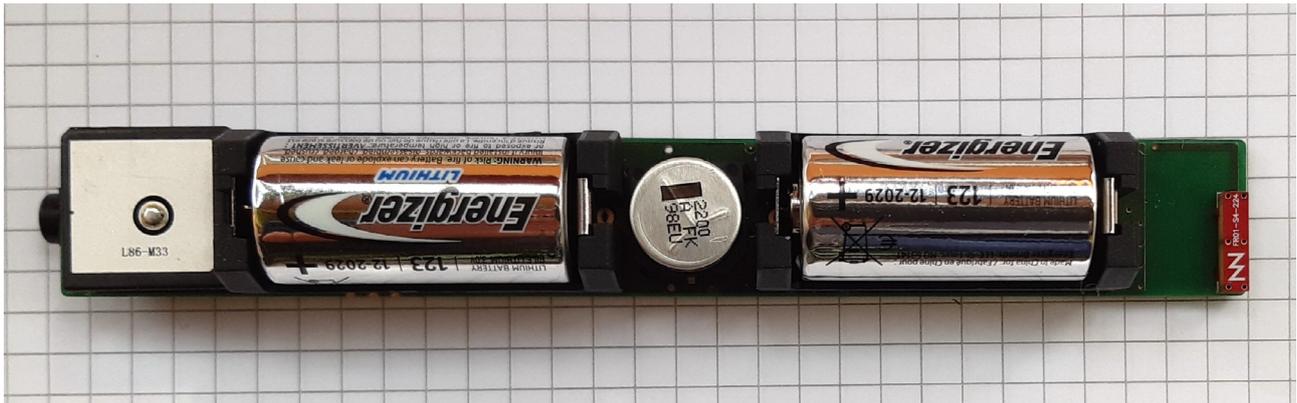
Als letzter Sensor ist ein GPS/GLONASS-Modul verbaut (GPS wird von den USA betrieben, GLONASS von der Russischen Föderation), hier nur als „GPS“ bezeichnet.

Es ist aber nicht immer nötig, GPS zu verwenden, (und manchmal auch gar nicht möglich, da GPS einigermassen freie Sicht zum Himmel erfordert und auch kostbare Energie verbraucht). In einfachen Fällen reicht es oft auch schon aus, die Position des Funkmastens der Mobilfunkzelle zu kennen. Dieser wird von LTraX automatisch an den Server mitgeschickt. In urbanem Gebiet ist der Radius einer Funkzelle meist kleiner als 1 km, auf dem Land selten größer als 5-10 km und bei Bedarf kann der Server immer noch GPS zusätzlich aktivieren).

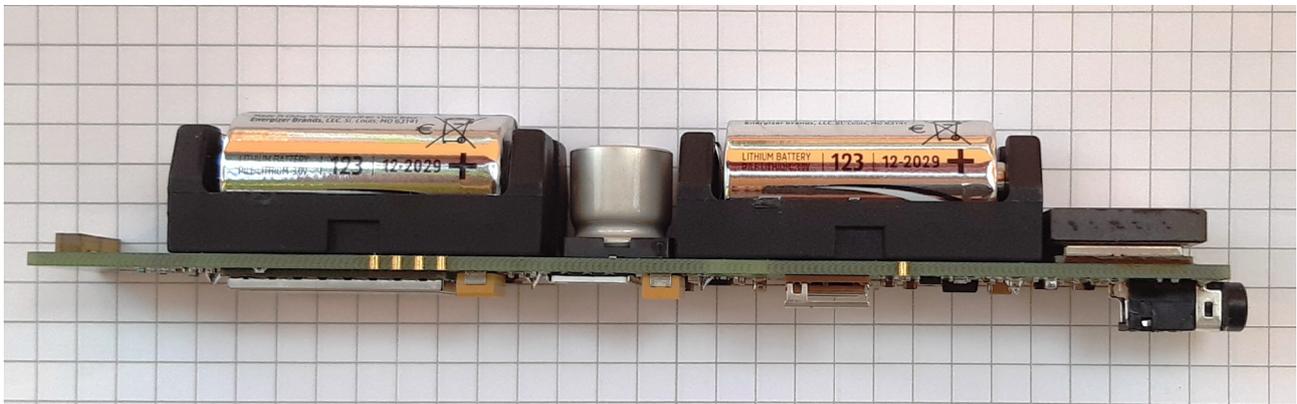
Zur Batterieversorgung sind 2 Stück Standard CR123a-Batterien (3V, ca. 1600 mAh Kapazität, Betrieb von -40 °C bis +70 °C ) vorgesehen, die sich preisgünstig überall beschaffen lassen.



*LTraX – Bestückungsseite, der 4 polige Klinkenstecker ist optional für externe Sensoren*



LTraX - Oberseite mit GPS-Modul, rechts die Embedded Antenne mit Feed



LTraX - Seitenansicht

## Das 3D-gedruckte Gehäuse

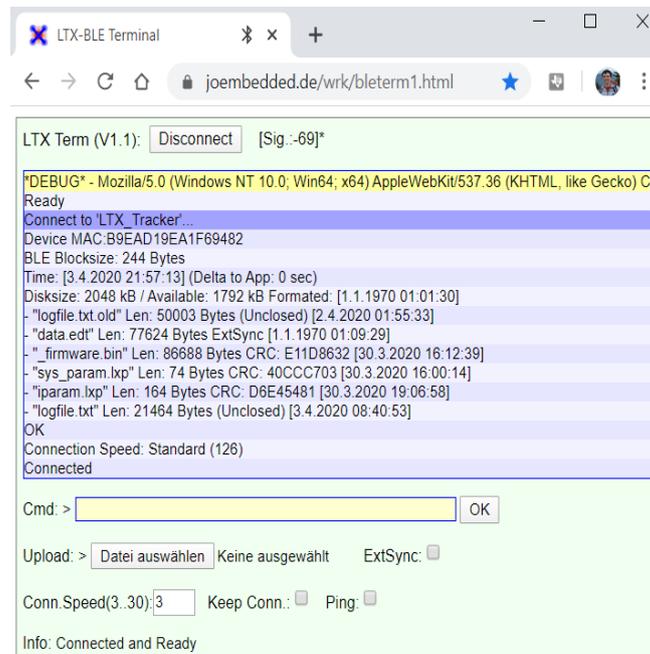
Die ersten Gehäuse habe ich auf dem kleinen 3D-Drucker meines Sohnes aus einem einfach zu verarbeitenden Material hergestellt. Für den späteren Einsatz wurden die Gehäuse aus robustem „ABS“ von einem Online-Anbieter für kleines Geld gespritzt. Das Gehäuse ist wasserdicht (IP68) und auch für den Außeneinsatz bestens geeignet: ein kleines Päckchen Trockenmittel im Gehäuse moderiert die eindringende (und ausdiffundierende) Feuchtigkeit und bei unzulässig hohen Feuchte-Werten wird der Anwender vom Server informiert. Ganz wichtig ist es, dass sich keine Über- oder Unter-Drücke im Gehäuse bilden können, denn das führt zu sogenannten Pump-Effekten, wo sich Kondenswasser im Gehäuse sammelt.

Es besteht auch die Möglichkeit, am Gehäuse absichtlich eine Öffnung anzubringen, damit der interne Temperatur/Feuchtesensor beispielsweise zur Feuchteüberwachung eines Lagerplatzes (als eine der vielen alternativen Anwendungsszenarien des LTraX) genutzt werden kann.

## Embedded-Software mit JesFs

Viele „Maker-Projekte“ scheitern daran, dass zwar eine bestimmte Aufgabe gelöst wurde, aber die Flexibilität für einen professionellen Einsatz fehlt.

Im ersten Moment mag der Einsatz eines echten Dateisystems auf einem „soo kleinen Gerät“ zwar übertrieben scheinen, doch genau aus obigen Grund habe ich für LTraX das (freie) JesFs-Dateisystem [11] (mit dem dazugehörigen Bootloader JesFsBoot) eingesetzt und für die nRF52-CPU portiert. JesFs ermöglicht es beispielsweise Funktionalität und Parameter komplett zu trennen oder auch sehr einfach per BLE auf die Dateien der abgespeicherten Daten, Logdateien oder Einstellungen zuzugreifen.



*Das BLE-Terminal, in HTML/Javascript für  
bequemen Dateitansfer und Firmware-Upload*

JesFs wurde so konzipiert, dass es sich selbständig mit minimalem Kommunikations-Overhead mit einem virtuellen Dateisystem eines Servers synchronisieren kann. Bei jeder Internet-Übertragung findet dieser Prozess automatisch statt. Sehr bequem!

Parameter- oder Firmware-Updates werden so automatisch auf den LTraX übertragen, während die neuesten Daten automatisch vom LTraX auf den Server übertragen werden.

Da auf dem LTraX nur ein maximaler Speicher von 2 MB verbaut ist, arbeiten die Dateien im Ringspeicher-Modus und der Speicher kann nie überlaufen, aber man hat – falls kein Internet zur Verfügung stehen sollte – per BLE immer noch die Historie der Daten. Und 2 MB sind da schon sehr viel.

LTraX ist darauf ausgelegt, möglichst wenig Energie zu verbrauchen. Als „Lebenszeichen“ meldet er sich regelmäßig (beispielsweise täglich) beim Server und optionale Sensoren werden mit einer einstellbaren Rate (1min – 24h) erfasst. Das GPS wird aber nur aktiviert, wenn etwa der Beschleunigungssensor eine Bewegung feststellt (oder der Magnetfeldsensor eine signifikante Lageänderung). Das GPS wird dann zusätzlich beispielsweise stündlich aufgezeichnet.

## Energie-Haushalt

Das NINA-B3 benötigt mit aktiviertem BLE ca. 5-10  $\mu\text{A}$ , dazu kommen ca. 10  $\mu\text{A}$  an Ruhestromen. Mit einem Grundverbrauch von  $< 20 \mu\text{A}$  halten die 1600 mAh der Batterien so für bis zu 10 Jahren und der LTrax ist trotzdem jederzeit per Bluetooth ansprechbar!

Eine Übertragung des SARA-R412 via 2G und Mobilem Internet dauert (bei gutem Netz und wenig Daten) ca. 15 sec und benötigt im Schnitt ca. 50 mA (LTE-M1 ist deutlich sparsamer). Es sind also im Idealfall 8000 Übertragungen mit einem einzigen Batteriesatz möglich!

Bei täglicher Übertragung ergeben sich so (rechnerisch und ohne Selbstentladung) 6.8 Jahre!

Der Bewegungssensor verbraucht aktiv ca. 5  $\mu\text{A}$ .

Das verwendete GPS Module benötigt ca. 30 mA und findet bei freier Sicht zum Himmel nach ca. 10-60 sec seine Position (je länger es inaktiv war, umso länger dauert es). Das GPS Modul benötigt für seine RTC ca. 7  $\mu\text{A}$ , kann dann aber schneller ein Signal finden. Bei sehr seltener Bewegung und langer gewünschter Batterielaufzeit kann es also sinnvoll sein, die RTC schlafend zu deaktivieren. Auch die gewünschte Genauigkeit der GPS Position beeinflusst den Stromverbrauch. Alle diese Parameter sind aber – dank JESFs – leicht einstellbar. Überschlagsmäßig kann man aber mit ca. 0.4 mAh Verbrauch bei täglicher Positionsbestimmung rechnen.

Bei guten Bedingungen, täglich 6 Stunden Bewegung, GPS Aufzeichnung alle 2h und Übertragung zum Server alle 6h werden ca. 2 mAh/Tag verbraucht, die Batterien halten dann ca. 2 Jahre, der Anwender wird aber vom Server rechtzeitig benachrichtigt, wenn sie erneuert werden sollen.

## SIM-Karten für „Regalware“

Es gibt inzwischen eine Vielzahl an Anbietern für M2M-SIM-Karten, aber die meisten davon arbeiten auf Vertragsbasis. Meines Wissens nach gibt es aktuell lediglich 2 Anbieter in Europa von Prepaid M2M-SIM-Karten. Diese haben den Vorteil, dass sich damit „Regalware“ produzieren lässt und gleichzeitig das Risiko für Missbrauch komplett wegfällt.

Bei 1NCE-SIMs [12] sind 500 MB mit einer Laufzeit von 10 Jahren enthalten, die 1NCE-SIMs sind in über 100 Ländern gültig und die SIM-Karte arbeitet leider nur im Temperaturbereich  $-25 \text{ }^{\circ}\text{C}$  bis  $+85 \text{ }^{\circ}\text{C}$  (die bis  $-40 \text{ }^{\circ}\text{C}$  einsetzbare Variante ist ein eSIM, die aufgelötet werden muss).

Bei Things-Mobile [13] sind zwar nur 100 MB enthalten, die Laufzeit ist aber unbeschränkt und Things-Mobile bietet SIMs für den erweiterten Temperaturbereich von  $-40^{\circ}\text{C}$  bis  $+105^{\circ}\text{C}$  als SIM an.

Beide SIMs sind standardmäßig mit 10 EUR aufgeladen.

Einmal davon abgesehen, dass bei selbst bei sehr intensiver Nutzung mit LTrax kaum mehr als 5 MB pro Jahr zu erwarten sind, bieten beide Anbieter die Möglichkeit, die Karten online nachzuladen.

Aus meiner Sicht sind damit erstmalig die Voraussetzungen geschaffen, um die Cloud auch auf „Regalware“ für den freien Handel auszudehnen!

## Fazit und Ausblick

Die Entwicklung des LTraX hat sehr viel Spaß gemacht und mein Verwandter kann nun seine Anhänger überwachen.

Es gibt eine Unmenge an weiteren Anwendungsmöglichkeiten des LTraX, beispielsweise bei der Diebstahlüberwachung von anderen Gütern, Agrar-Sensorik, Internet-of-Animals (Lokalisierung von frei weidenden Schafen, Ziegen oder Kühen), ...

Für Fragen, Anregung, etc., freue ich mich über Ihre Mail [3]!

## Links/Quellen

- [1] Die-mobile-Werbefläche: <https://www.die-mobile-werbeflaeche.de>
- [2] Maker – Wikipedia: <https://de.wikipedia.org/wiki/Maker>
- [3] Meine Kontaktdaten: [joembedded@gmail.com](mailto:joembedded@gmail.com) und <https://joembedded.de>
- [4] BLE-Modul NINA-B3 Open CPU:  
<http://spezial.com/de/u-blox-kurzstrecken-funkmodule-bluetooth-module-nina-b3-series>
- [5] Segger Embedded Studio: <https://www.segger.com>
- [6] LTE/2G-Modem SARA-R412: <https://www.spezial.com/de/u-blox-sara-r4>
- [7] Embedded Antenne Fractus RUN mXTEND™: <https://fractusantennas.com/mobile-antenna>
- [8] BLE-Soc Nordic nRF52840:  
<https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840>
- [9] Temperatur/Feuchte Sensor Sensirion SHT30-DIS:  
<https://www.sensirion.com/de/umweltsensoren/feuchtesensoren>
- [10] Motion Sensor ST LSM303AGR: <https://www.st.com/en/mems-and-sensors/lsm303agr.html>
- [11] JesFs – Jo's Embedded Serial File System: <https://github.com/joembedded/JesFs>
- [12] 1NCE Prepaid SIMs: <https://1nce.com>
- [13] Things-Mobile Prepaid SIMs: <https://www.thingsmobile.com/de>

\*\*\*