



HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT  
DRESDEN

ÜBERTRAGUNG VON SENSORDATEN MITTELS LoRa  
PROJEKTSEMINAR

# Katastrophennetz mithilfe von Meshtastic

## Dokumentation

*Christian Grieb*  
*Göran Heinemann*  
*Julian Meinking*

unter Aufsicht von  
Prof. Dr.-Ing. Jörg VOGT

14. März 2024

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>2</b>
<b>2</b>	<b>Fragestellung</b>	<b>2</b>
<b>3</b>	<b>Technologie</b>	<b>2</b>
3.1	Was ist LoRa? . . . . .	2
3.2	Warum LoRa? . . . . .	3
3.3	LoRa und LoRaWAN . . . . .	4
3.4	Meshtastic . . . . .	4
<b>4</b>	<b>Geräte</b>	<b>5</b>
4.1	Heltec LoRa32 v3 . . . . .	5
4.2	LILYGO T-Echo . . . . .	5
4.3	LILYGO T-Deck . . . . .	5
<b>5</b>	<b>Firmware flashen</b>	<b>6</b>
5.1	Hardware identifizieren oder auswählen . . . . .	6
5.1.1	Serial Treiber für ESP32 . . . . .	6
5.1.2	Serial Treiber für nRF52 . . . . .	6
5.2	Firmware für ESP32 flashen . . . . .	6
5.3	Firmware für ESP32 nRF52 . . . . .	7
<b>6</b>	<b>Software</b>	<b>7</b>
6.1	UI . . . . .	8
6.1.1	iOS . . . . .	8
6.1.2	Web . . . . .	9
<b>7</b>	<b>Häufige Probleme - Kommunikation mit Gerät über USB</b>	<b>10</b>
<b>8</b>	<b>Simulation</b>	<b>12</b>
<b>9</b>	<b>Praxistest</b>	<b>13</b>
9.1	Logfiles . . . . .	16
9.2	Fazit . . . . .	16
<b>10</b>	<b>Theorie</b>	<b>17</b>
10.1	Link-Budget . . . . .	17
10.2	Kenngrößen . . . . .	17
10.3	dB . . . . .	17
10.4	Channelsettings . . . . .	18
10.5	Transceiverwerte . . . . .	18
10.6	Maximal mögliche Übertragungsstärke . . . . .	18
10.7	Überlegungen . . . . .	19
10.8	Fazit . . . . .	23
10.9	Notizen . . . . .	24
10.9.1	Link-Budget Formel . . . . .	24
10.9.2	Line of Sight tool . . . . .	24
10.9.3	Beispielrechnung . . . . .	25
<b>11</b>	<b>Fazit</b>	<b>26</b>

# 1 Aufgabenstellung

Ziel dieses Projekts war das Experimentieren mit Meshtastic auf LoRa-fähigen Geräten. Meshtastic ist ein Open-Source-Projekt, das es ermöglicht, ein Mesh-Netzwerk aufzubauen, das auf der LoRa-Technologie basiert. Es ist eine kostengünstige und energieeffiziente Möglichkeit, ein Netzwerk aufzubauen, das unabhängig von Internet und Mobilfunknetzen funktionieren kann.

## 2 Fragestellung

Ist Meshtastic als unabhängiges Kommunikations-Netzwerk für den Krisenfall im Raum Dresden geeignet?

## 3 Technologie

### 3.1 Was ist LoRa?

LoRa (von Long Range) ist eine proprietäre Funktechnologie im Besitz von Semtech. Sie ist für die Langstreckenübertragung (z.B. 10 km), schmalbandige Übertragung (gemessen in Kbps) und energiesparende Kommunikation konzipiert, hauptsächlich für Internet of Things (IoT)-Netzwerke. Dafür wird eine drahtlose Modulationstechnik, die aus der Chirp Spread Spectrum (CSS)-Technologie abgeleitet ist, verwendet. Sie codiert Informationen auf Radiowellen mithilfe von Chirp-Impulsen. Die modulierte Übertragung von LoRa ist robust gegen Störungen und kann über große Entfernungen empfangen werden.

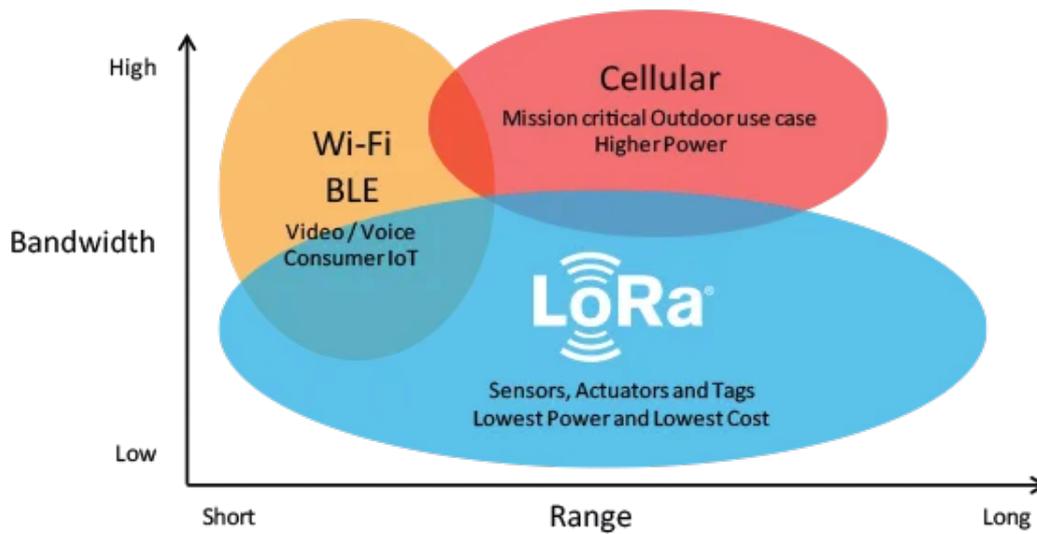
Es eignet sich ideal für Anwendungen, die kleine Datenmengen mit niedrigen Bitraten übertragen. Daten können über eine längere Reichweite übertragen werden im Vergleich zu Technologien wie WLAN, Bluetooth oder ZigBee. Diese Eigenschaften machen LoRa besonders geeignet für Sensoren und Aktoren, die im Niedrigenergiemodus arbeiten.

Außerdem arbeitet LoRa in einem lizenzfreien Sub-Gigahertz-Frequenzband (d.h. unter 1 GHz), aber die zu verwendenden Frequenzen variieren von Region zu Region aufgrund regulatorischer Anforderungen. Wenn Sie ein LoRa-Gerät kaufen, muss sichergestellt sein, dass das richtige Frequenzband unterstützt wird.

In Europa - 863–870MHz (normalerweise 868MHz).

## 3.2 Warum LoRa?

LoRa versucht die Lücke zwischen zwischen Kommunikationstechnologien wie WiFi, Bluetooth und LTE zu schließen.



[https://www.semtech.com/uploads/images/LoRa\\_Why\\_Range.png](https://www.semtech.com/uploads/images/LoRa_Why_Range.png)

Es ist für große Reichweite, kleine Bandbreite und Niedrigenergiekommunikation gemacht. Alles in allem also extrem nützlich für IoT Geräte. Einige Beispiele sind:

- Wassersensoren in einer entfernten Umgebung (Grundwasser)
- Rauchwarnmelder
- Tierbeobachtung
- Verbrauchsmessungen bei Endkunden (Gas, Strom)
- Wetterstationen die nur ab und zu Informationen übertragen

### 3.3 LoRa und LoRaWAN

LoRaWAN ist über LoRa angesiedelt und definiert das Kommunikationsprotokoll und die Systemarchitektur.

Es ist wichtig zu verstehen, dass es möglich ist LoRa ohne LoRaWAN zu benutzen. Andere LoRa-basierte Netzwerke sind Helium, The Things Network, Disaster.radio und was wir weiter betrachten werden, Meshtastic.

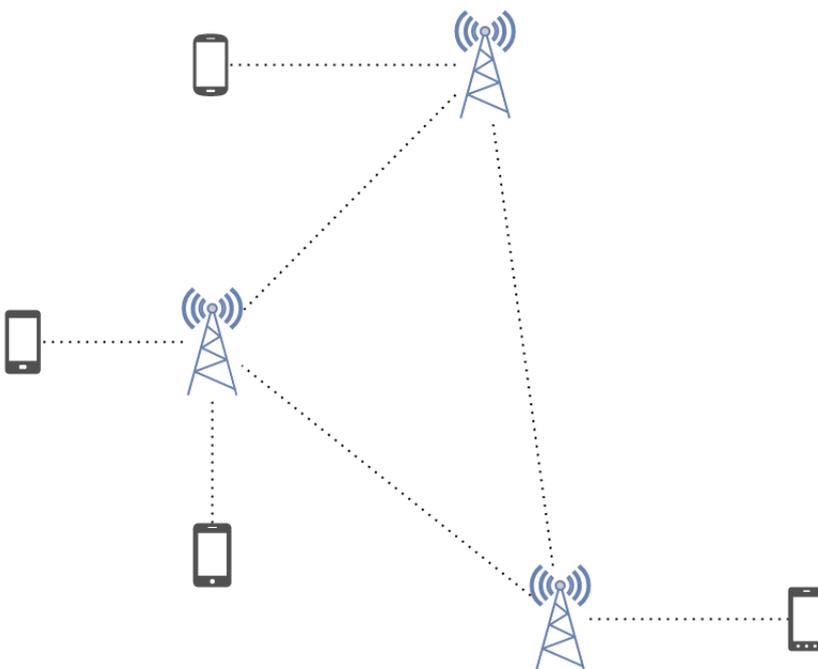
### 3.4 Meshtastic

Wie im vorherigen Absatz erwähnt, baut Meshtastic auf LoRa auf und schafft ein dezentrales Mesh-Netzwerk.

Es bringt folgende Eigenschaften mit sich:

- Verschlüsselte und Textbasierte Kommunikation
- Plattformunabhängig
  - Computer (unabhängig vom Betriebssystem)
  - Android (native App)
  - iOS (native App)
- Dezentralisiert
- Geringer Stromverbrauch
- Optionales Standort teilen
- Open-Source

Anders als traditionelle Mobilfunknetzwerke, verbindet sich jedes Endnutzegerät mit einem LoRa Radio und alle LoRa Radios, welche Meshtastic nutzen, können Nachrichten, selbst wenn die Radios nicht im gleichen Mesh sind, weiterleiten. Das passiert so lange, bis die Nachricht Ihr Ziel erreicht oder die voreingestellten “Hops” ausgeschöpft werden.



[https://loganmarchione.com/2023/05/lora-and-meshtastic/20230510\\_003.png](https://loganmarchione.com/2023/05/lora-and-meshtastic/20230510_003.png)

## 4 Geräte

Folgende Geräte haben wir für das Projekt genutzt:

### 4.1 Heltec LoRa32 v3

max TX power: +21dBm



<https://heltec.org/project/wifi-lora-32-v3>

### 4.2 LILYGO T-Echo

max TX power: +21dBm



<https://www.lilygo.cc/products/t-echo>

### 4.3 LILYGO T-Deck

max TX power: +21dBm



<https://www.lilygo.cc/products/t-deck>

## 5 Firmware flashen

Meshtastic bietet eine Firmware für verschiedene Geräte an. Je nach Gerät gibt es kleine Unterschiede in Funktionalität (Display-Support, Speicher, etc.) und im Flashvorgang.

### 5.1 Hardware identifizieren oder auswählen



Achtung! Vorweg, Geräte nur mit angeschlossener Antenne einschalten! Anderenfalls kann das Gerät sich im schlimmsten Fall selber zerstören.

Meshtastic wird offiziell nur von bestimmten Geräten unterstützt, welche ein LoRa Modul besitzen.

Es ist darauf zu achten, dass jedes Gerät welches im Meshnetz betrieben werden soll, auf der gleichen Frequenz arbeitet. Hier gibt es Unterschiede!

In Deutschland können die freien Frequenzbänder 433 MHz und 868MHz, auf welchen Lora operiert, ohne Lizenzkosten oder Amateurfunklizenz genutzt werden.

Solange eine WLAN-Verbindung zu einem Device nicht notwendig ist und Bluetooth ausreicht, sollte ein nRF52 Chip gewählt werden, da diese energieeffizienter als ESP32 Chips arbeiten und einfacher zu flashen sind. Es gibt auch noch Geräte auf Basis des RP2040, diese haben wir allerdings nicht getestet.

Eine Liste mit unterstützter Hardware findet sich hier:  
<https://meshtastic.org/docs/hardware/devices/>

#### 5.1.1 Serial Treiber für ESP32

Einen für das eigene Betriebssystem passenden Treiber auf folgender Seite identifizieren, herunterladen und Installieren:

<https://meshtastic.org/docs/getting-started/serial-drivers>

#### 5.1.2 Serial Treiber für nRF52

nRF52 Chips benötigen normalerweise keinen Serial Treiber. Sie benutzen einen UF2 bootloader, welche das Gerät als USB-Stick vom Betriebssystem erkennen lassen.

Auf keinen Fall folgenden USB geräte treiber herunterladen, es sei denn es wird UF2 support benötigt

<https://meshtastic.org/docs/getting-started/serial-drivers/nrf52>

### 5.2 Firmware für ESP32 flashen

<https://meshtastic.org/docs/getting-started/flashing-firmware/esp32/>

Da es bei uns auf verschiedenen PCs Probleme gab haben wir zum Flashen unter Linux eine Nix-Flake erstellt, die Python mit den richtigen Paketen installiert und eine kleine Anleitung (auch zum selber Compilieren der Firmware) für ESP32 und nRF52 Geräte enthält.

### 5.3 Firmware für ESP32 nRF52

<https://meshtastic.org/docs/getting-started/flashing-firmware/nrf52/>

Beim diesen Geräten ist es bei uns manchmal vorgekommen, dass das Flashen von Firmware zwar bis zu dem “Drag und Drop”-Schritt funktioniert und dann aber nicht wirklich mit der neuen Version neu startet. Falls das passiert muss man sich mit einer seriellen Konsole mit dem Gerät verbinden und einfach nur einmal Enter drücken, besonders nachdem Factory-Erase. Das steht unter dem Punkt Factory-Erase auch dokumentiert, aber man benötigt nicht zwingend die Meshtastic CLI, sondern lediglich ein Programm wie z. B. minicom unter Linux.

## 6 Software

Zum aktuellen Zeitpunkt gibts es jeweils eine App für iOS und Android. Außerdem gibt es einen Web Client, der sich per Bluetooth, Wi-Fi und Serial verbinden kann.

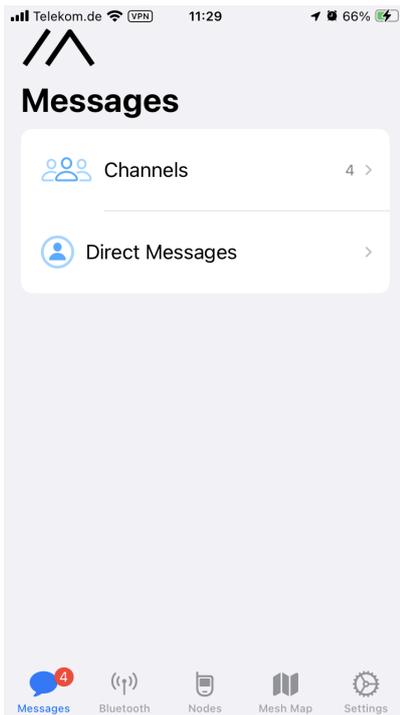
App	Link	Quellcode
Android	Playstore	Github
iOS	App Store	Github
Web	Web App	Github

Die Apps werden regelmäßig geupdated und sind modern aufgebaut.

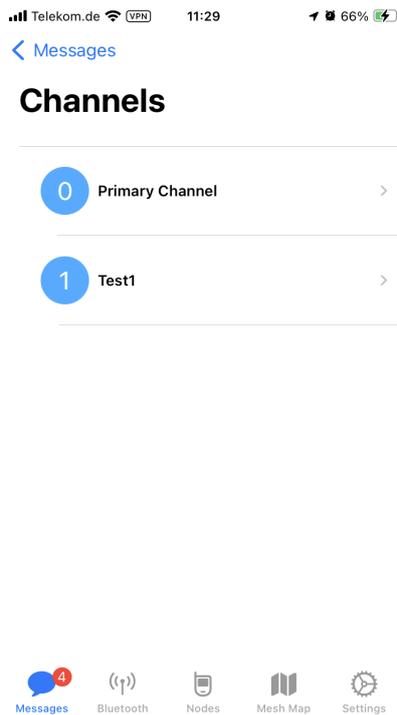
Ursprünglich sollte hier ein Vergleich der Funktionalitäten erfolgen, es gibt aber keine erkennbaren Unterschiede, außer kleinere in der Bennung und Bedienung.

# 6.1 UI

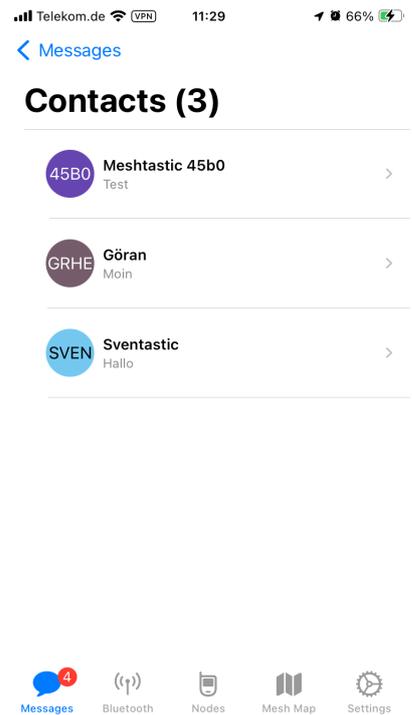
## 6.1.1 iOS



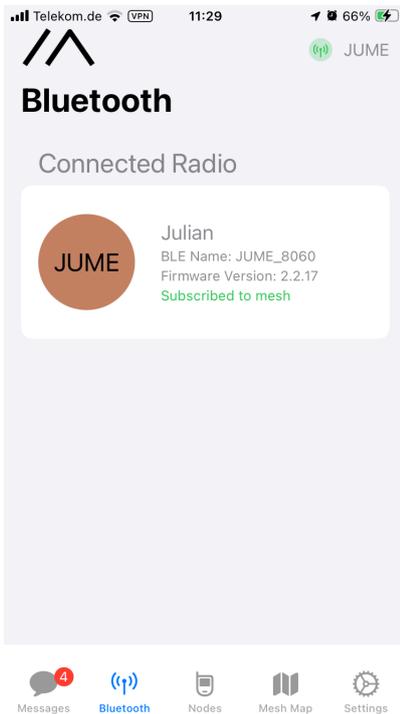
Nachrichten Tab



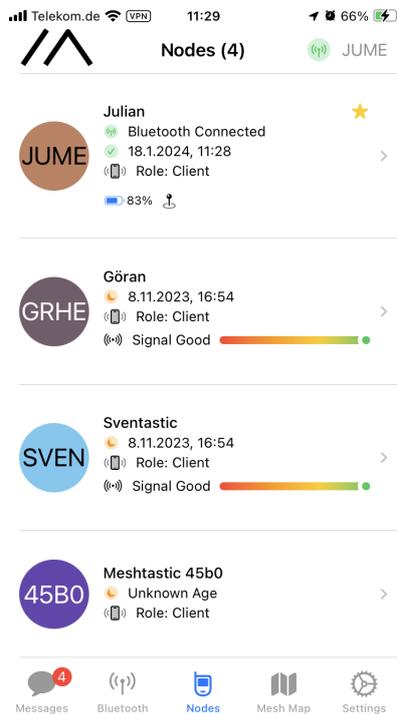
Nachrichten in Channels



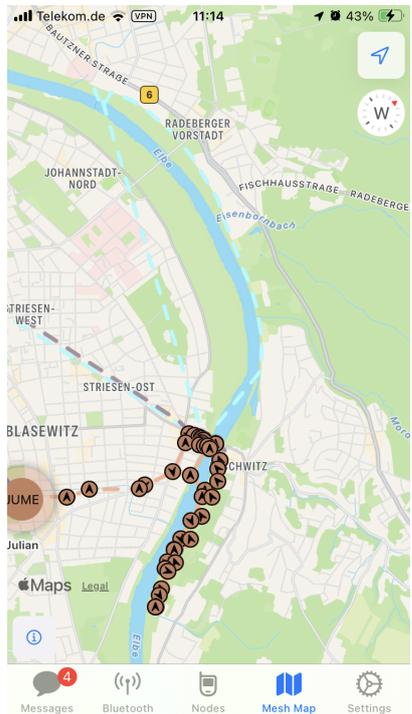
Direktnachrichten mit anderen Nodes



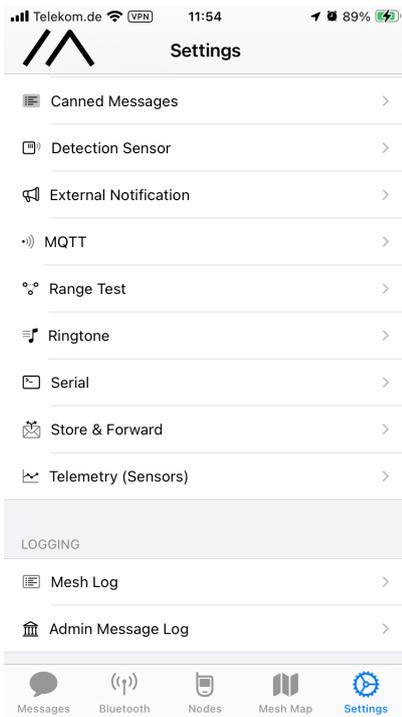
Bluetooth-Verbindung zum eigenen Node



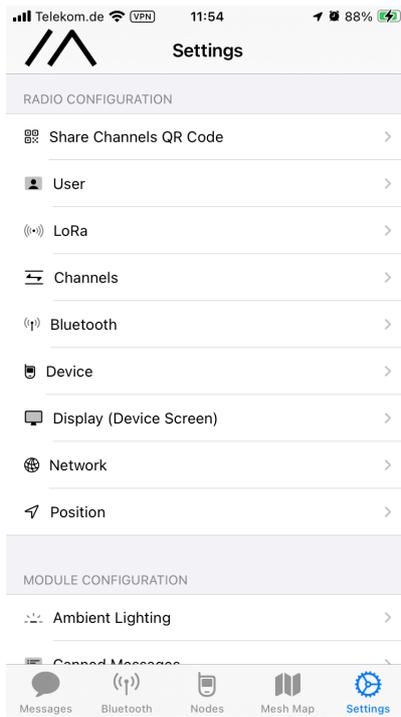
Übersicht der verbundenen Nodes



Kartenansicht

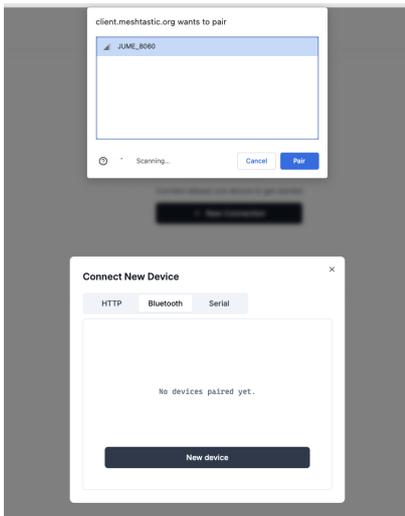


Modul-Konfiguration

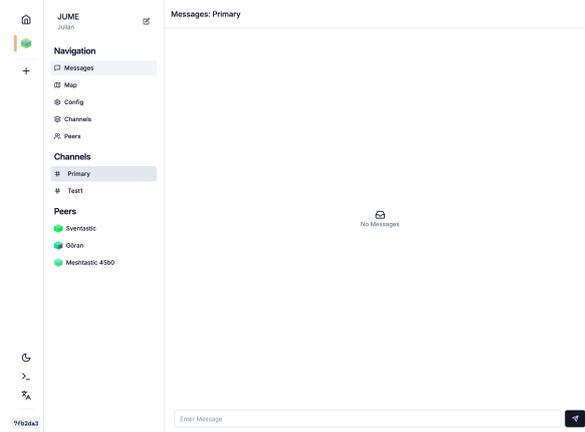
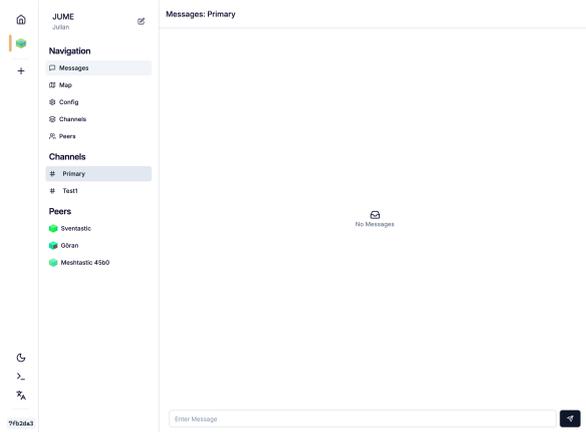


Radio-Konfiguration

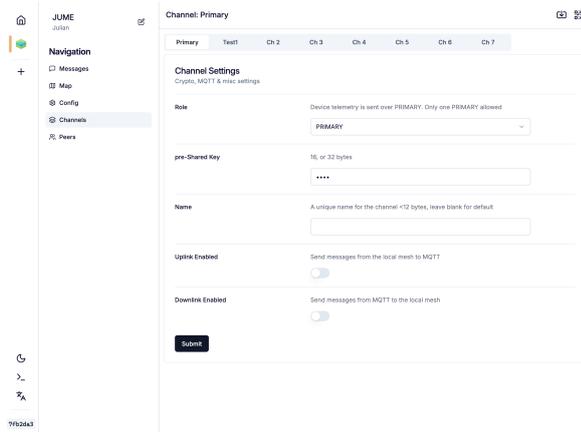
## 6.1.2 Web



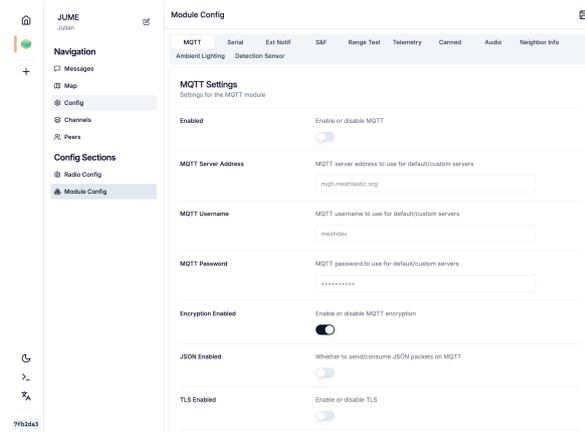
Bluetooth-Verbindung mit Node



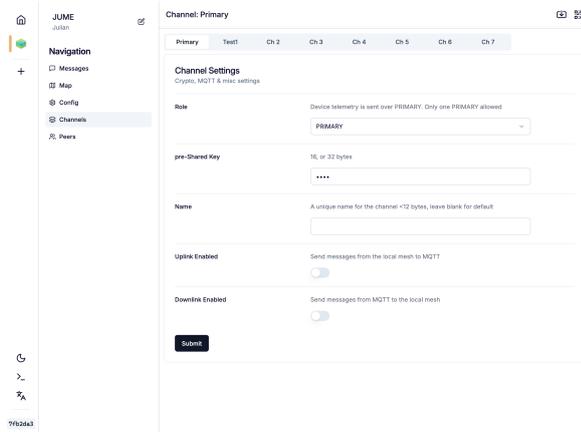
## Nachrichten



## Übersicht der Verbundenen Nodes



## Channel-Konfiguration



## Modul-Konfiguration

## Radio-Konfiguration

# 7 Häufige Probleme - Kommunikation mit Gerät über USB

- Die Gerät-Datei `/dev/ttyUSB0` gehört der Nutzergruppe `dialout`. Damit der Nutzer Schreibrechte erhalten kann, muss er zur Gruppe hinzugefügt werden:

```
sudo usermod -aG dialout <user-name>
logout
```

Der Nutzer muss sich ausloggen und wieder einloggen, damit er in der Gruppe enthalten ist.

- Prüfen, ob das Kabel zwischen Computer und Gerät auch wirklich Daten übertragen kann.
- USB-C → USB-C funktioniert manchmal nicht. Dies könnte an einem Fehler bei USB-C Power Delivery liegen. Adapter USB-C → USB-A (findet man meist als OTG-Adapter) schafft Abhilfe.

## 8 Simulation

Die Wartung vieler Meshtastic Nodes für den Zweck des Experimentierens mit der Software bzw. dem Protokoll kommt mit einem größeren Aufwand einher. Dafür kann sich eine Simulation besser eignen, um Resultate ohne Aufwand direkt beobachten zu können.

Mit der Software Meshtasticator ist es möglich, ein Meshtastic Netzwerk zu simulieren. Diese basiert auf zwei vorherigen Simulatoren (`lora-network-simulator` und `LoRaSim`) und nutzt unterliegend die Meshtastic Linux Anwendung, weswegen die Verwendung eines Linux Betriebssystems Voraussetzung ist.

Wir haben den Simulator in einer Ubuntu-VM installiert. Einige Pakete mussten installiert werden.

```
sudo apt install git python3-tk python3-pip \
python3-venv gpio libgpio gpio libgpio-dev \
libgpio2 libyaml-cpp-dev libbluetooth-dev
```

Danach muss das Meshtasticator und Meshtastic-Firmware Repository herunter geladen werden.

Um die Firmware zu compilieren, benötigt man PlatformIO. Das Tool lässt sich mit dem System Paketmanager wie z.B. `apt` installieren, häufig bekommt man eine veraltete Version, da die Paketquellen der meisten Betriebssysteme nur langsam aktualisiert werden.

Deshalb empfehlen wir, den Paketmanager von Python zu nutzen:

```
pip install platformio
```

Der Installationspfad liegt in `./local/bin`, welcher nicht standardmäßig in der Systemvariable `PATH` enthalten sein kann. Falls der Befehl `pio` beim Aufruf nicht gefunden wird, reicht es aus, den Installationspfad zum System-`PATH` hinzuzufügen:

```
export PATH="$HOME/.local/bin:$PATH"
```

Die Meshtastic Firmware, welche normalerweise für IoT Hardware kompiliert wird, kann auch nativ auf einem Linux Betriebssystem mithilfe von Portduino ausgeführt werden.

Mit Environment-Parameter `native` baut PlatformIO die Firmware für ein Linux Betriebssystem.

```
pio run --environment native
```



Die PlatformIO Extension für VSCode bietet eine praktische GUI, mit der die Befehle ausgeführt werden können.

Das kompilierte Binary befindet sich dann im Ordner `firmware/.pio/build/native`. Damit der Meshtasticator die gewünschten Anordnungen simulieren kann, muss die Binary in den Ordner von Meshtasticator kopiert werden. Bevor die Abhängigkeiten von Meshtasticator installiert werden können, sollte in dem Ordner eine Python-Umgebung aktiviert werden:

```
python3 -m venv venv
source venv/bin/activate
```

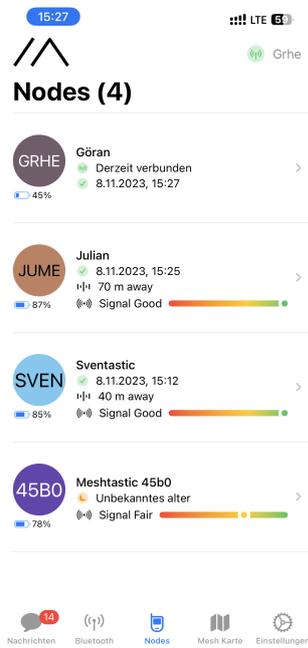
Die Abhängigkeiten können mit `pip3 install -r requirements.txt` installiert werden.

## 9 Praxistest

In unseren Praxistests am Elbufer in Dresden haben wir eine Reichweite von 4km Luftlinie erreicht.

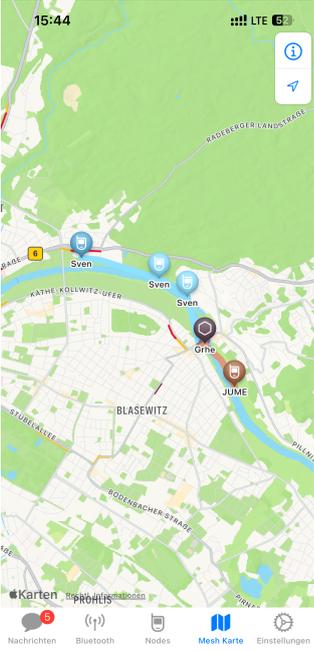
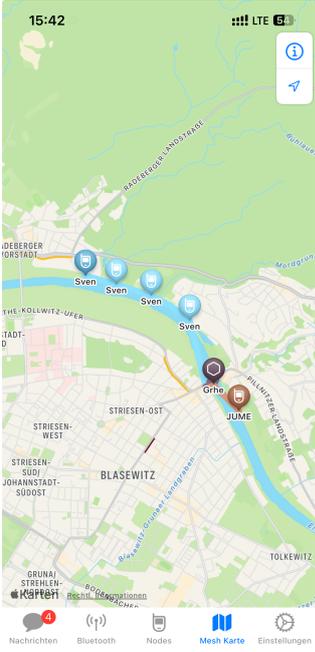
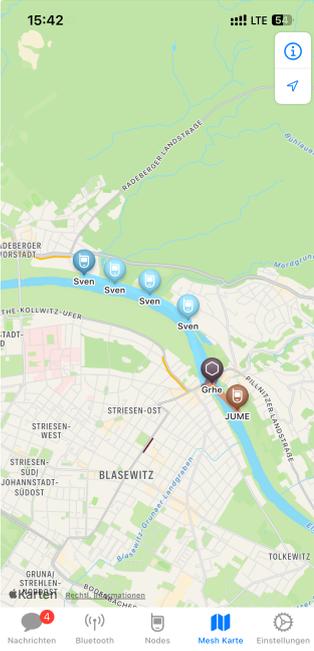
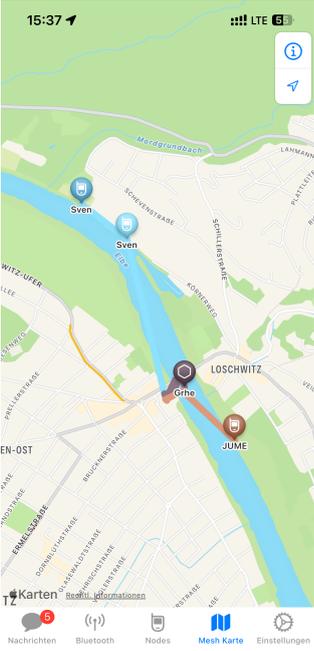
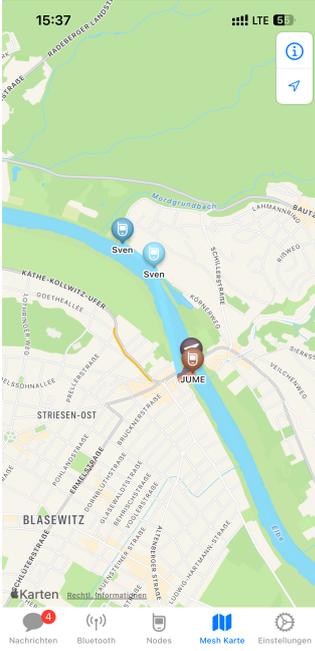
Wir sind mit dem Fahrrad jeweils vom Blauen Wunder in beide Richtungen gefahren und haben unsere Meshtastic Geräte im Rucksack mitgenommen.

Für den kontinuierlichen Sendetest, haben wir in den Moduleinstellungen den Range Test aktiviert und auf ein Intervall von 1min gestellt.

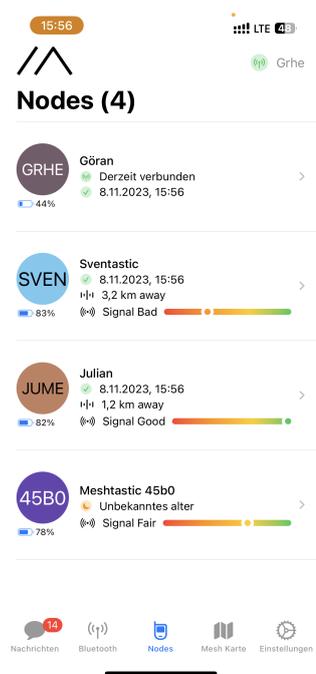
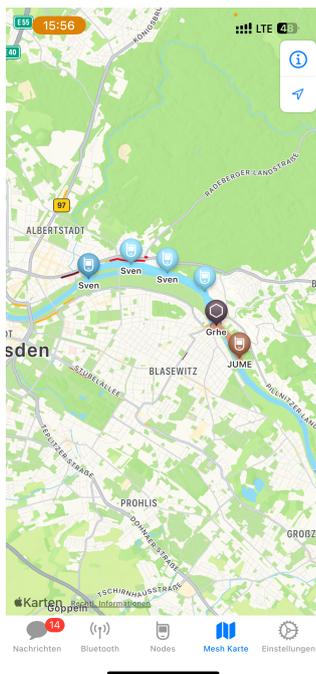


Mithilfe eines Setup QR Codes, konnten wir unser Mesh aufbauen. Node "Meshtastic 45b0" war noch von einem vorherigem Versuch im Mesh enthalten, aber nicht am Experiment beteiligt.

# Node Übersicht



Bei 3km Gesamt-Distanz (Luftlinie) herrschte immer noch eine gute Verbindung von der Brücke zu beiden anderen Nodes.



Bei 3.2km Distanz (Luftlinie), begann die Verbindung von Görän zu Christian schlechter zu werden.



Blick vom Blauen Wunder Richtung Pillnitz

## 9.1 Logfiles

Node	Typ	Link
Göran	Metrics	Download
	Position	Download
Julian	Metrics	Download
	Position	Download
Sventastic	Metrics	Download
	Position	Download

## 9.2 Fazit

Dieser Test konnte zeigen, dass Meshtastic mit batteriebetriebenen Geräten ohne besondere Ausrichtung der Antennen eine gute Distanz auf Luftlinie zurücklegen kann. Wegen der Weiterleitung von Nachrichten konnte die Verbindung zwischen Christian und Julian aufrechterhalten werden.

Die Werte ließen sich wahrscheinlich auch noch verbessern, wenn bessere Antennen verwendet würden und eine genauere Ausrichtung der Antennen erfolgen würde.

Innerhalb der Stadt, mit vielen Gebäuden und anderen Hindernissen, ist die Reichweite wesentlich geringer. Das Meshnetzwerk müsste dann enger aufgebaut werden, um eine gute Verbindung zu gewährleisten.

# 10 Theorie

## 10.1 Link-Budget

Die Reichweite einer Funkverbindung lässt sich mittels des Link-Budgets (Leistungs-übertragungsbilanz) darstellen und gibt die Qualität eines Funk-Übertragungskanals an. Eines der einfachsten Modelle um ein Linkbudget zu errechnen ist mittels Addition der Sendeleistung (Transmitter Power, Tx), der Empfängerempfindlichkeit (Receiver Power, Rx), des Antennengewinns und der Freiraumdämpfung (Free Space Path Loss, FSPL).

## 10.2 Kenngrößen

Der Spreading Faktor und somit die Reichweite eines Senders sind von den Ausbreitungsbedingungen abhängig. Die Empfängerempfindlichkeit hängt von Signal-Rausch-Verhältnis (SNR), Rauschfaktor (NF) und Bandbreite (BW) ab.

Die Freiraumdämpfung beeinträchtigt die Reichweite. Durch die Verdopplung der Entfernung nimmt die Freiraumdämpfung um 6 dB zu. Reflektionen und Brechungen der Funkwellen an Hindernissen und Boden beeinflussen Signalpegel und Reichweite. Im LoRaWAN-Netzwerk befindet sich eine Seite der Funkverbindung in der Regel in Bodennähe. Hindernisse in der ersten Fresnelzone beeinflussen den Signalpegel auf der Rx-Seite und verkürzen die Reichweite. SF-Werte und somit die Reichweite eines Senders hängen von den Ausbreitungsbedingungen ab. LoRaWAN erlaubt mittels ADR ein automatisches Netzmanagement und regelt damit die Reichweiten der Sender.

## 10.3 dB

Die Einheit dB (Dezibel) wird im Zusammenhang mit Funkverbindungen verwendet, um die Signalstärke, Dämpfung oder Verstärkung von elektromagnetischen Signalen zu messen. dB ist eine logarithmische Einheit, die das Verhältnis zwischen zwei Größen ausdrückt. In Bezug auf Funkverbindungen sind die beiden häufigsten Anwendungen die Messung der Signalstärke und die Angabe von Dämpfung oder Verstärkung.

### 1. Signalstärke in dBm (Dezibel Milliwatt):

- dBm misst die absolute Leistung eines Signals im Vergleich zu einem Referenzwert von 1 Milliwatt.
- Ein positives dBm-Wert zeigt an, dass das Signal stärker ist als 1 Milliwatt, während ein negativer Wert darauf hinweist, dass es schwächer ist.
- Beispiel: Ein Signal mit -50 dBm ist stärker als ein Signal mit -60 dBm.

### 2. Dämpfung und Verstärkung in dB:

- dB wird auch verwendet, um die Dämpfung oder Verstärkung von Signalen in einer Leitung oder einem System auszudrücken.
- Eine positive dB-Angabe deutet auf Verstärkung hin, während eine negative dB-Angabe auf Dämpfung hinweist.
- Beispiel: Ein Verstärker, der das Signal um 20 dB verstärkt, erhöht die Signalstärke um das 100-fache.

Bei Funkverbindungen wird die Signalstärke oft in dBm gemessen, während Dämpfung oder Verstärkung von Antennen, Kabeln oder Verstärkern in einfachen dB-Angaben ausgedrückt werden. Dies ermöglicht eine präzise und effektive Kommunikation über die Leistung von Funksignalen und die Leistung von Komponenten in drahtlosen Netzwerken.

## 10.4 Channelsettings

Channelsetting	Long Range / Fast
Alt Channelname	Long Fast
Data Rate	1.07 kbps (default)
Spreading Factor/ Symbols	11 / 2048
Coding Rate	4/5
Bandwith	250

## 10.5 Transceiverwerte

transmit Power	21dBm
Antenna gain	0dBi
RX sensitivity	-131dBm
RX antenna	0dBi
Linkbudget	152dB

WiFi LoRa 32 v3 (SX1262 Lora Chip)

$P(\text{dBm}) = 21\text{dBm}$

Max Receiving sensitivity = -136dBm@SF12 BW=125KHz

(1)<https://www.semtech.com/design-support/lora-calculator>

## 10.6 Maximal mögliche Übertragungsstärke

Maximal 500mW ERP (2)

Maximal 2.15dBi Antenne (3)

$$\begin{aligned}P(\text{dBm}) &= 10 \cdot \log_{10}(P(500 \text{ mW}) \div 1 \text{ mW}) \\ &= 26.9897000434 \\ &= 27\text{dBm}\end{aligned}$$

$$P(\text{max}) = P(\text{dBm}) + 2.15\text{dBi}$$

$$P(\text{max}) = 29.15\text{dBm}$$

(2) <https://www.bundesnetzagentur.de/DE/Fachthemen/Telekommunikation/Frequenzen/Grundlagen/Frequenzplan/frequenzplan-node.html>

(Stand März 2022) Eintrag 251004 (Frequenznutzungsbedingungen)

(3) <https://www.thethingsnetwork.org/forum/t/max-allowed-antenna-gain-for-lora/37182>

## 10.7 Überlegungen

Mit einem Link Budget von 152dBm (Einstellung = long fast, mit 21dBm transmit Power und 0dBi Antenne) lässt sich unter optimalen Bedingungen(nur Freiraumdämpfung(Vakuum)) eine Distanz von mehr als 800km Überwinden. Da dies nur unter optimalen Bedingungen möglich ist werden solche Werte natürlich nicht erreicht.

Im Weltraum können Funksignale tatsächlich viel weiter reisen. Unter Normalbedingungen, auf unserem Planeten spielt die Sichtlinie eine bedeutende Rolle. Aufgrund der Tatsache, dass die Welt eine Kugel ist, ist es möglich mit 1,7m Höhe 4,7km weit zu sehen. Da im sub-GHz Bereich eine quasioptische Ausbreitung für Radiowellen gilt, ist der Horizont auch gleichzeitig eine Begrenzung für Funksignale.

Im nächsten Schritt betrachten wir die Wellenausbreitung mit dem Hata-Okumura-Modell (Hata-Modell).

Das Hata-Modell verwendet verschiedene Begriffe in der Dämpfungsformel, um zwischen verschiedenen Typen von Landoberflächen zu unterscheiden. Es werden ländliche bis zu dicht besiedelte Regionen unterschieden.

Hata-Ausbreitungsformel:

$$\begin{aligned}Loss(dB) &= A + B \cdot \log_{10}(d) - K + C \\A &= 69.55 + 26.16 \cdot \log_{10}(f) - 13.82 \cdot \log_{10}(h_s) \\B &= 44.9 - 6.55 \log_{10}(h_s)\end{aligned}$$

$d$  = Distanz zwischen Sender und Empfänger (km)

$h_s$  = Höhe der Senderantenne (m)

$h_r$  = Empfängerantennenhöhe (m)

$K$  = Korrekturfaktor

$C$  = Korrekturfaktor

$f$  = Übertragungsfrequenz [MHz]

**Urban:** Große Stadt mit großen Gebäuden welche zwei oder mehreren Stockwerken besitzen oder größere Dörfer mit sehr nahe stehenden Häusern

**Medium:** Mittlere Stadt

**Suburban:** Vorort

**Freifläche:** Keine großen Bäume oder Gebäude in Sichtlinie

Freifläche:

$$K = (1.1 \cdot \log_{10}(f) - 0.7) \cdot h_r - (1.56 \cdot \log_{10}(f) - 0.8)$$

$$C = -4.78 \cdot (\log_{10}(f))^2 + 18.33 \cdot \log_{10}(f) - 40.94$$

Suburban:

$$K = (1.1 \cdot \log_{10}(f) - 0.7) \cdot h_r - (1.56 \cdot \log_{10}(f) - 0.8)$$

$$C = -2 \cdot [\log_{10}(f/28)]^2 - 5.4$$

Medium:

$$K = (1.1 \cdot \log_{10}(f) - 0.7) \cdot h_r - (1.56 \cdot \log_{10}(f) - 0.8)$$

$$C = 0$$

Urban,  $150 \leq f \leq 200$  MHz:

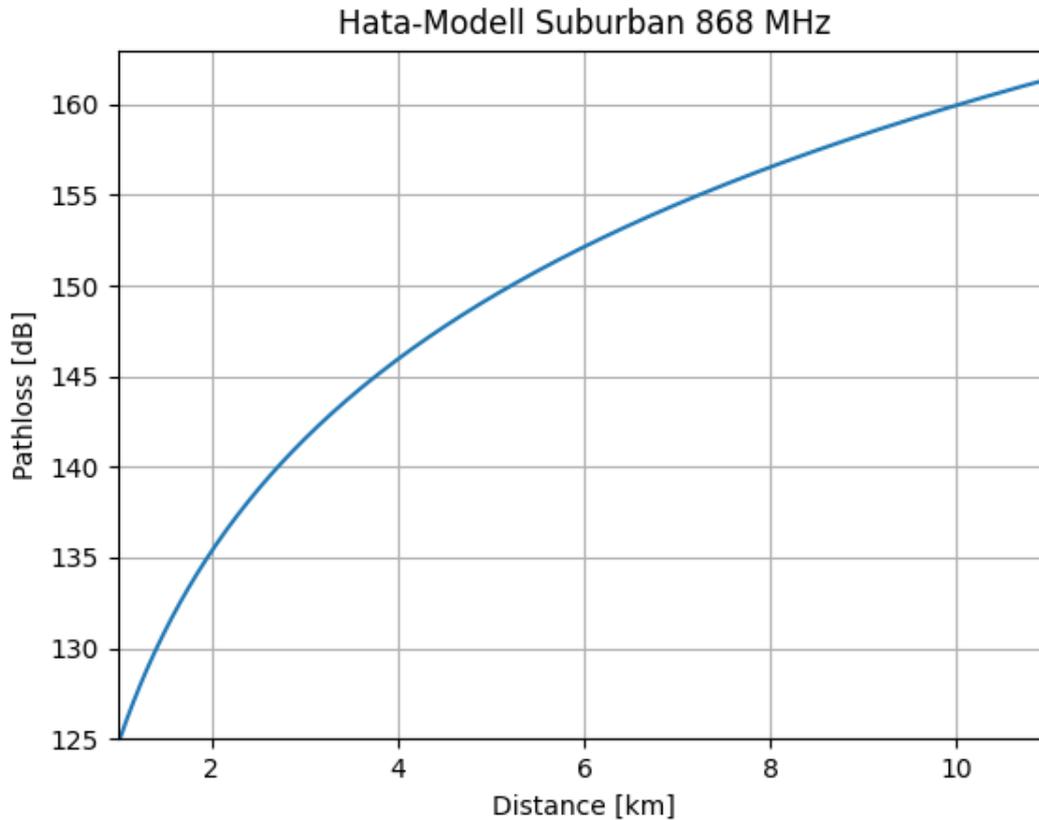
$$K = 8.29 \cdot (\log_{10}(1.54 \cdot h_r))^2 - 1.1$$

$$C = 0$$

Urban,  $200 < f \leq 1500$  MHz:

$$K = 3.2 \cdot (\log_{10}(11.75 \cdot h_r))^2 - 4.97$$

$$C = 0$$



Die Grafik zeigt die Reichweite nach dem Okumura-Hata-Modell für ein Suburbanes Gebiet bei Nutzung einer 868MHz frequenz. Wie der Grafik zu entnehmen, wird bei unserem Linkbudget von 152dBm maximal eine Reichweite von ungefähr 6km möglich

Die bereitgestellten Formeln decken nicht alle im Okumura-Modell vorgeschlagenen Bedingungen ab. Hatas Ansatz gilt nur mit folgende Einschränkungen:

Frequenz	150-1500 MHz
Distanz	1-20km
Höhe Senderantenne	30-200m
Höhe Empfängerantenne	1-10m

Mit folgenden Werten wurde im suburbanen gearbeitet:

$$h_s = 30\text{m}$$

$$h_r = 2\text{m}$$

$$f = 869\text{MHz}$$

$$\begin{aligned} A &= 69.55 + 26.16 \cdot \log_{10}(f) - 13.82 \cdot \log_{10}(h_s) \\ &= 69.55 + 26.16 \cdot \log_{10}(868) - 13.82 \cdot \log_{10}(30) \\ &\approx 126.008 \end{aligned}$$

$$\begin{aligned} B &= 44.9 - 6.55 \log_{10}(h_s) \\ &= 44.9 - 6.55 \log_{10}(30) \\ &\approx 35.225 \end{aligned}$$

$$\begin{aligned} K &= (1.1 \cdot \log_{10}(f) - 0.7) \cdot h_r - (1.56 \cdot \log_{10}(f) - 0.8) \\ &= (1.1 \cdot \log_{10}(868) - 0.7) \cdot 2 - (1.56 \cdot \log_{10}(868) - 0.8) \\ &\approx 1.281 \end{aligned}$$

$$C = 0$$

$$Loss(dB) = A + B \cdot \log_{10}(d) - K + C$$

$$\log_{10}(d) = \frac{Loss(dB) - A + K - C}{B}$$

$$d = 10^{\frac{Loss(dB) - A + K - C}{B}}$$

$$= 10^{\frac{Loss(dB) - 126.008 + 1.281 - 0}{35.225}}$$

$$= 10^{\frac{Loss(dB) - 124.727}{35.225}}$$

Freifläche(Max)	64.640km (Begrenzung! Sichtlinie 24.6km)
Suburban(Max)	10.130km
Urban(Max)	9.975km
Freifläche(152dB)	37.943km (begrenzt auf Sichtlinie 24.6km)
Suburban(152dB)	5.946km
Urban(152dB)	15.855km
Freifläche(Gebäude)	12.612km
Suburban(Gebäude)	1.220km
Urban(Gebäude)	1.018km
Sichtlinie	24.6km

Es darf nicht vergessen werden, dass das Hata-Modell nur eine annäherung im Bereich bei Entfernungen um die 1-20km ist. Alle Werte die darüber berechnet werden sollten mit anderen Werkzeugen überprüft werden. Wie in unserem Fall Freifläche(Max) und Freifläche(Gebäude).

Für (Gebäude) wird eine Dämpfung von 17dB für das Eindringen in Gebäude und ein Verlust von 8dB für das Fading angenommen. Das bedeutet, dass die Analyse an die Bedingungen für Sender innerhalb von Gebäuden angepasst wurde. Das Link-Budget wird daher um 25 dB auf 128dB reduziert.

Für (Max) wird das maximal mögliche legale Limit für Sendeleistung und Antennenverstärkung ausgereizt und liegt bei einem Linkbudget, wie anfänglich dargelegt, von 160.15dB.

(152dB) beschreibt die maximal mögliche ausbreitung des Signals bei der Sendeleistung der von uns verwendeten Geräte.

Die Höhe der Antenne des Senders wurde auf 30Meter festgelegt, da dass Hata Modell nur mit dieser minimalen Höhe arbeiten kann.

Die Empfängerantenne liegt auf 2Meter Höhe.

## 10.8 Fazit

Das Hata-Modell kennt keine Erdkrümmung und ist nur eine erste Näherung. Als Kontrolle sollten bei Berechnungen immer die Sichtlinie beachtet werden und damit die plausibilität der Berechnung kontrolliert und ggf. eingeschränkt werden. Die minimale Höhe der Senderantenne macht die Anwendung des Hata-Modells für unser Projekt weniger praktikabel gibt jedoch einen ersten Anhaltspunkt um mit unseren Messwerten zu vergleichen.

## 10.9 Notizen

### 10.9.1 Link-Budget Formel

$$PRX = PTX + GTX + GRX - LTX - LFS - LP - LRX$$

$PRX$  = received power (dBm)

$PTX$  = transmitter output power (dBm)

$GTX$  = transmitter antenna gain (dBi)

$GRX$  = receiver antenna gain (dBi)

$LTX$  = transmit feeder and associated losses (feeder, connectors, etc.) (dB)

$LFS$  = free space loss or path loss (dB)

$LP$  = miscellaneous signal propagation losses

(these include fading margin, polarization mismatch, losses associated with medium through which signal is travelling, other losses...) (dB)

$LRX$  = receiver feeder and associated losses (feeder, connectors, etc.) (dB)

$$FSPL(\text{dB}) = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55$$

$$FSPL = (4\pi d/\lambda) \cdot 2 = (4\pi df/c) \cdot 2$$

$FSPL$  = Free Space Path Loss (Freiraumdämpfung)

$d$  = Abstand zwischen Tx und Rx in Metern

$f$  = Frequenz in Hertz

$$\text{Rx-Empf.} = -174 + 10 \log_{10}(BW) + NF + SNR$$

$BW$  = Bandbreite in Hz

$NF$  = Rauschfaktor in dB

$SNR$  = Signal-Rausch-Verhältnis (signal to noise ratio).

Es gibt an, wie weit das Signal über dem Rauschen liegen muss.

### 10.9.2 Line of Sight tool

<https://www.heywhatsthat.com>

### 10.9.3 Beispielrechnung

Lassen Sie uns ein vereinfachtes Beispiel für ein LoRa Link-Budget für einen Punkt-zu-Punkt-Kommunikationslink betrachten. Bitte beachten Sie, dass reale Szenarien komplexer sind und zusätzliche Faktoren berücksichtigen, aber dieses Beispiel sollte ein grundlegendes Verständnis bieten.

1. **Sendeleistung (Tx Power):** Wir gehen von einer Sendeleistung von 20 dBm aus.
2. **Frequenz (f):** Es wird mit einer Frequenz von 868 MHz gerechnet.
3. **Distanz (d):** Als Distanz werden 2 Kilometer verwendet.
4. **Antennen Gains (Gt and Gr):** Für die Sende- und Empfangsantenne wird ein Gain von 2 dBi verwendet.
5. **Freiraumdämpfung (Lp):** unter verwendung der Freiraumdämpfungsformel:

$$Lp = 20 \log_{10} d + 20 \log_{10} (f) + L_{FS}$$

$$Lp = 20 \log_{10} 2 + 20 \log_{10} (868 \cdot 10^6) + 20 \cdot \log_{10} \frac{4\pi}{c}$$

In diesem Fall ist  $c$  die Lichtgeschwindigkeit. Das Ergebniss ist der Path Loss in dB.

6. **Empfangsleistung (Rx Power):** Die werte werden in die Formel eingesetzt:

$$\text{Rx Power} = \text{Tx Power} + \text{Tx Antenna Gain} - \text{Path Loss} + \text{Rx Antenna Gain}$$

Ersetze die Werte und berechne die Empfangsleistung.

Die berechnete Empfangsleistung sollte mit der Empfindlichkeitsspezifikation des Empfängers verglichen werden. Wenn die empfangene Leistung höher ist als der Empfindlichkeitsschwellenwert, wird erwartet, dass die Verbindung unter den spezifizierten Bedingungen funktioniert.

Bitte beachten Sie, dass dies ein vereinfachtes Beispiel ist, in einem realen Szenario sollten zusätzliche Faktoren wie atmosphärische Bedingungen, Störungen und Ausbleibungseffekte berücksichtigt werden, um eine genauere Analyse des Link-Budgets durchzuführen.

## 11 Fazit

Meshtastic ist ein gut voranschreitendes Projekt, das eine solide Software-Plattform für die Kommunikation zwischen LoRa-fähigen Geräten ermöglicht.

Damit aber Meshtastic als Katastrophennetz eingesetzt werden könnte, sind aktuell noch einige Hürden im Weg.

Für den Einsatz als Katastrophennetz müssten erstmal genug solcher Meshtastic- und LoRa-fähigen Geräte angeschafft werden, sodass ein zuverlässiges Mesh-Netzwerk innerhalb der Stadt aufgebaut werden könnte. Dies müsste am Besten mit einer Großbestellung geschehen, da sonst der Markt überlastet werden und die Preise steigen könnten.

Es müssten genug Geräte innerhalb der Stadt hoch positioniert werden, da die Reichweite von LoRa durch Gebäude wesentlich begrenzt ist. Nur bei freier Luftlinie sind Verbindungen über mehrere Kilometer möglich. Andere Funktechnologien mit mehr Sendeleistung, sind durch Gebäude weniger beeinträchtigt. Aber im Katastrophenfall mit zusammengebrochener Infrastruktur (Black-Out), ist LoRa mit geringem Energieverbrauch besser geeignet.

Aktuell ist Meshtastic noch softwareseitig auf Netzwerke mit bis zu 80 Nodes limitiert. Das reicht bei Weitem nicht, um den Raum Dresden vollständig abzudecken. Das Protokoll scheint auch nicht für solche Mega-Meshes ausgelegt zu sein, was man am maximalen Hop Count von 7 erkennen kann. Ein limitierter Hop Count könnte eine weitreichende Zustellung verhindern.

Die Software ist für ein Open Source Projekt ohne große Finanzierung auf einem guten Niveau, aber noch nicht für allgemeine Nutzung ausgereift. Dafür würde es einen Feinschliff in User Experience und Stabilität benötigen.