



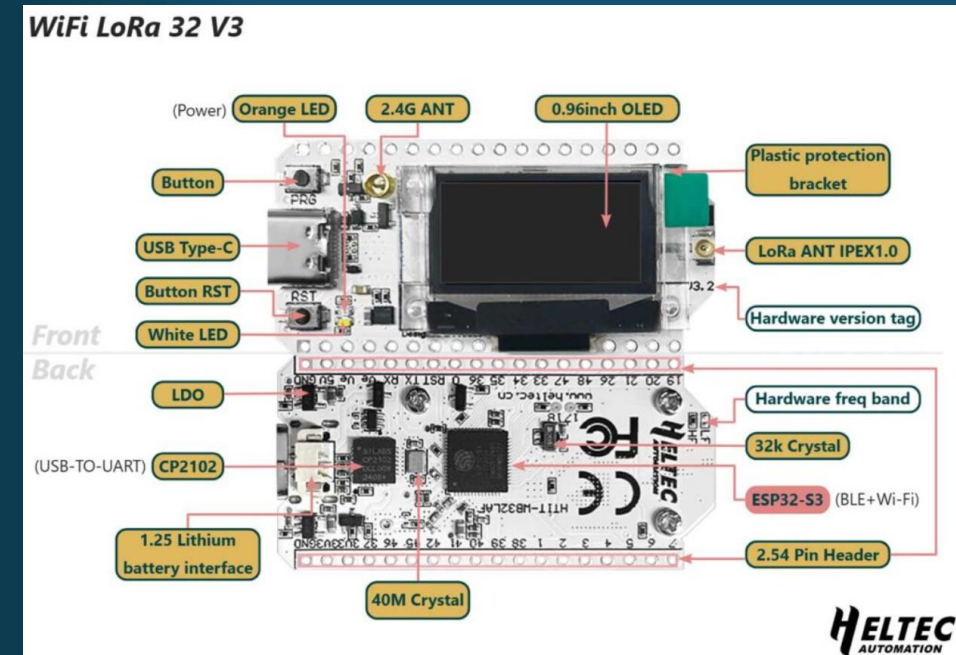
LoRa

MeshCore

Einführung und erste Schritte

Übersicht der Themen

- Grundlagen – Was ist LoRa und woher kommt Meshcore?
- Wie wird eine Nachricht zwischen Companion und Repeater Nodes technisch versendet
- Frequenzen und Ausbreitung - Physik ist bei LoRa alles
- Presets – Wann welches nutzen?
- Verbindung und Einstellungen via Bluetooth (BT) App
- Datenschutz und #Kanäle beitreten/einrichten
- Hardware und die richtigen Einstellungen für den Start
- Erste eigene Analysen
- Stromverbrauch



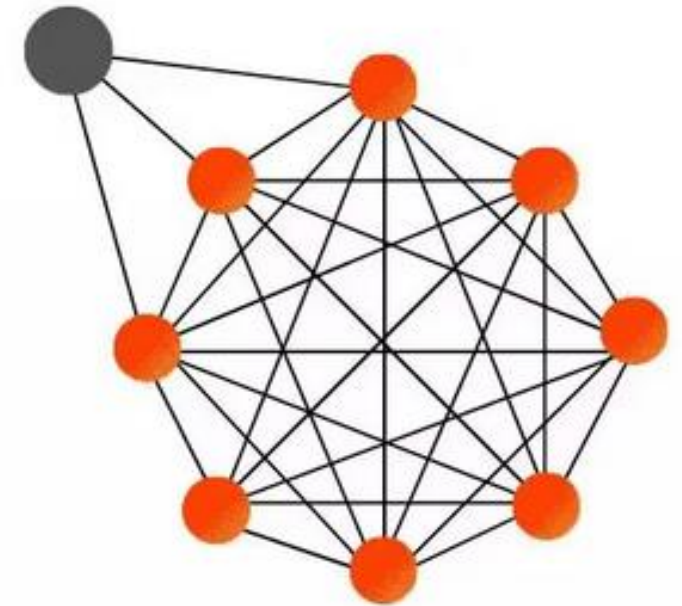
Die Geschichte von Meshcore ist eng mit dem Meshtastic-Projekt verknüpft

- Die Geschichte von **Meshcore** ist eng mit dem **Meshtastic-Projekt** und der Open-Source-Bewegung verknüpft, die etwa um **2019** Fahrt aufnahm.
- **Ursprung:** Das Projekt entstand aus dem Wunsch der Community (allen voran Entwickler wie **Kevin Hester**), eine völlig unabhängige, lizenzfreie Kommunikation für Outdoor-Aktivitäten wie Wandern, Gleitschirmfliegen oder Skifahren zu schaffen.
- **Die Vision:** Ziel war es, die damals neue LoRa-Technologie nicht nur für Sensoren (wie im industriellen LoRaWAN), sondern für die **direkte Textkommunikation zwischen Menschen** nutzbar zu machen – ohne auf zentrale Infrastrukturen oder Mobilfunkmasten angewiesen zu sein.
- **Entwicklung:** Was als kleines Bastelprojekt für Funkamateure und Technik-Begeisterte begann, entwickelte sich schnell zu einer globalen Plattform für **Krisenvorsorge (Prepping)** und zivilgesellschaftliche Kommunikation, da die Hardware immer günstiger und die Software (der "Core") immer leistungsfähiger wurde.

Grundlagen – Was ist LoRa und woher kommt Meshcore?

Bevor man Mesh versteht, muss man das Fundament kennen.

- **LoRa (Long Range):** Ein proprietäres Funkprotokoll für die Low-Power-Kommunikation.
- **Chirp Spread Spectrum (CSS):** Informationen werden auf "Chirps" (frequenzmodulierte Impulse) moduliert. Dies macht das Signal extrem widerstandsfähig gegen Rauschen und Störungen.
- **Eigenschaften:**
 - Sehr hohe Reichweite (bis zu 20+ km bei Sichtverbindung).
 - Extrem geringer Energieverbrauch (Batteriebetrieb über Wochen/Monate).
 - Geringe Datenrate (ideal für Text, nicht für Bilder/Video).



Mesh Topology

In der Netzwerktechnik unterscheidet man meist zwischen der Stern-Topologie (zentral) und der Mesh-Topologie (dezentral).

Das zentrale Netzwerk (Stern-Topologie)

Stell dir ein WLAN zu Hause oder das Mobilfunknetz vor. Alle Geräte (Handys, Laptops) sprechen mit einem zentralen Punkt: dem Router oder dem Funkmast.

Abhängigkeit: Wenn der Router ausfällt oder der Funkmast keinen Strom hat, bricht die gesamte Kommunikation für alle Teilnehmer zusammen.

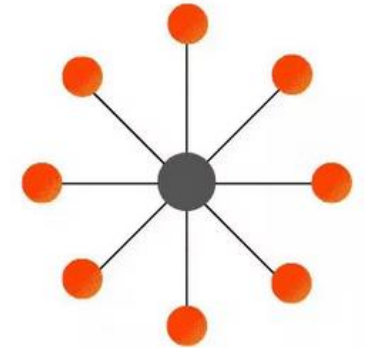
Hierarchie: Es gibt eine klare Trennung zwischen "Server" (Verteiler) und "Client" (Nutzer).

Das dezentrale Mesh-Netzwerk

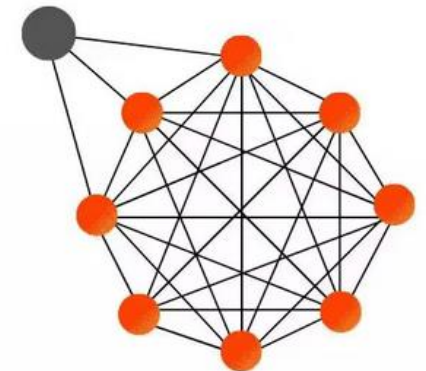
Bei Meshcore/Meshtastic gibt es diesen "Chef" nicht.

Gleichberechtigung: Jeder Node (Knoten) ist gleichzeitig Sender, Empfänger und Vermittler.

Struktur: Die Intelligenz des Netzwerks ist auf alle Teilnehmer verteilt.



Star Topology



Mesh Topology

Die Vorteile eines dezentralen Mesh-Netzwerks

Merkmal	Zentrales Netzwerk (WLAN/LTE)	Dezentrales Mesh (Meshcore)
Ausfallrisiko	Hoch (Basisstation ist kritisch)	Sehr gering (Netz heilt sich selbst)
Infrastruktur	Erfordert Masten/Kabel/Internet	Autark (Off-Grid)
Kosten	Monatliche Gebühren	Einmalige Hardware-Anschaffung
Einrichtung	Durch Profis/Provider	Durch die Community / Ad-hoc
Datenschutz	Provider kann mitlesen/tracken	Ende-zu-Ende Verschlüsselung möglich

Vertiefung: Wie funktioniert Chirp Spread Spectrum (CSS) genau?

Das Herzstück von LoRa ist nicht einfach nur Funk, sondern eine mathematisch elegante Art, Daten in Frequenzänderungen zu verstecken.

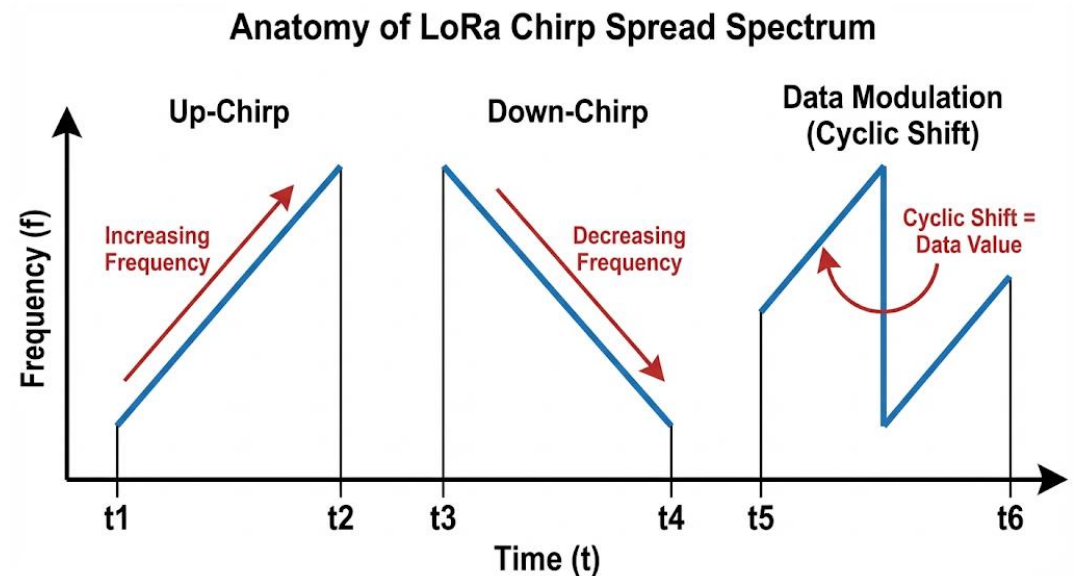
Ein "Chirp" ist ein Signal, dessen Frequenz sich über die Zeit kontinuierlich ändert.

In der LoRa-Welt nutzen wir zwei Hauptformen:

Up-Chirp: Die Frequenz steigt über die Zeit linear an (von der unteren zur oberen Grenze der Bandbreite).

Down-Chirp: Die Frequenz sinkt linear ab.

Daten-Modulation: Die Information wird nicht durch die Frequenz selbst übertragen, sondern durch den **Zeitpunkt**, an dem ein Chirp innerhalb seines Frequenzsprungs "neu ansetzt" (Zyklische Verschiebung).



Warum ist CSS so extrem robust?

Die Magie von CSS liegt in der sogenannten **Störfestigkeit** und dem **Verarbeitungsgewinn (Process Gain)**:

Signale unter dem Rauschen: CSS ermöglicht es, Signale noch zu dekodieren, wenn sie bis zu **20 dB unter dem thermischen Rauschen** liegen. Für herkömmliche Funkverfahren (wie FSK oder AM) wäre dieses Signal unlesbar.

Doppler-Resistenz: Da CSS auf der *Änderung* der Frequenz über die Zeit basiert und nicht auf einer exakten festen Frequenz, ist es sehr unempfindlich gegenüber Frequenzverschiebungen, die durch Bewegung (Doppler-Effekt) entstehen.

Multipath-Resistenz: In Städten werden Funksignale an Hauswänden reflektiert und kommen zeitversetzt an (Echo). **CSS kann diese Echos sehr gut ignorieren, da die zeitliche Verschiebung des Chirps mathematisch klar vom Original unterscheidbar ist.**

Der Spreading Factor (SF) – Das "Tuning-Rad von Chirp"

Der Spreading Factor (SF7 bis SF12) bestimmt, wie "breit" und "langsam" ein Chirp ist.



Höherer SF (z.B. SF12): Ein einzelnes Daten-Bit wird über einen sehr langen Zeitraum und viele Chirps verteilt.

- **Vorteil:** Enorme Reichweite und Empfindlichkeit.
- **Nachteil:** Die Nachricht braucht sehr lange zum Senden ("Airtime"), was den Akku belastet und die Datenrate minimiert.

Niedrigerer SF (z.B. SF7): Die Chirps sind kurz und schnell.

- **Vorteil:** Hohe Datenrate, geringer Stromverbrauch.
- **Nachteil:** Geringere Reichweite.

Der Verbindungsaufbau zwischen 2 Nodes: Preamble und Sync

Damit ein Empfänger weiß, dass eine Nachricht kommt, beginnt jedes LoRa-Paket mit einer festen Struktur:

1.Preamble: Eine Folge von mehreren Up-Chirps (meist 8). Der Empfänger erkennt daran: "Achtung, hier spricht jemand in meiner Sprache!".

2.Sync-Word: Ein spezieller Down-Chirp signalisiert den exakten Startzeitpunkt der eigentlichen Daten. Das ist wie ein Startschuss für die Stoppuhr des Empfängers.

Im Kontext von Funktechnik und Datenübertragung (wie bei LoRa oder Meshcore) ist die **Preamble** (deutsch: Präambel oder Einleitung) der allererste Teil eines gesendeten Datenpakets. Man kann sie sich wie ein "**Hallo, Achtung!**" vorstellen, das der Sender ruft, bevor er mit der eigentlichen Nachricht beginnt.

Meshcore in der Praxis. Die Kommunikation erfolgt in der Regel zwischen Companion Nodes und Repeatern

Der Companion Node (Der persönliche Begleiter)

Dies ist das Gerät, das du in der Tasche hast oder das auf deinem Schreibtisch steht.

Zusammensetzung: Er besteht meist aus der **LoRa-Hardware** (z. B. Heltec V3 oder T-Beam) und einem **Endgerät** (Smartphone, Tablet oder PC).

Verbindung: Die Verbindung zwischen Mensch und Node erfolgt meist über **Bluetooth (BT)** oder **WLAN**.

Aufgabe: Er dient als Ein- und Ausgabegerät für Nachrichten.

- Er speichert die Nachrichten lokal in der App.
- Er sendet deine Position (GPS), wenn gewünscht.

Energie: Oft auf Mobilität ausgelegt (Akku), wird bei Nichtbenutzung oft in den Schlafmodus versetzt.

Der Repeater Node (Die Infrastruktur)

Ein Repeater (oft auch als "Router" konfiguriert) ist ein unbemannter Knotenpunkt, der strategisch platziert wird.

Zusammensetzung: Nur die **LoRa-Hardware**, oft wetterfest verbaut und an einem hohen Punkt (Dach, Berg, Mast).

Verbindung: Er hat meist **keine** Verbindung zu einem Smartphone. Er arbeitet völlig autark.

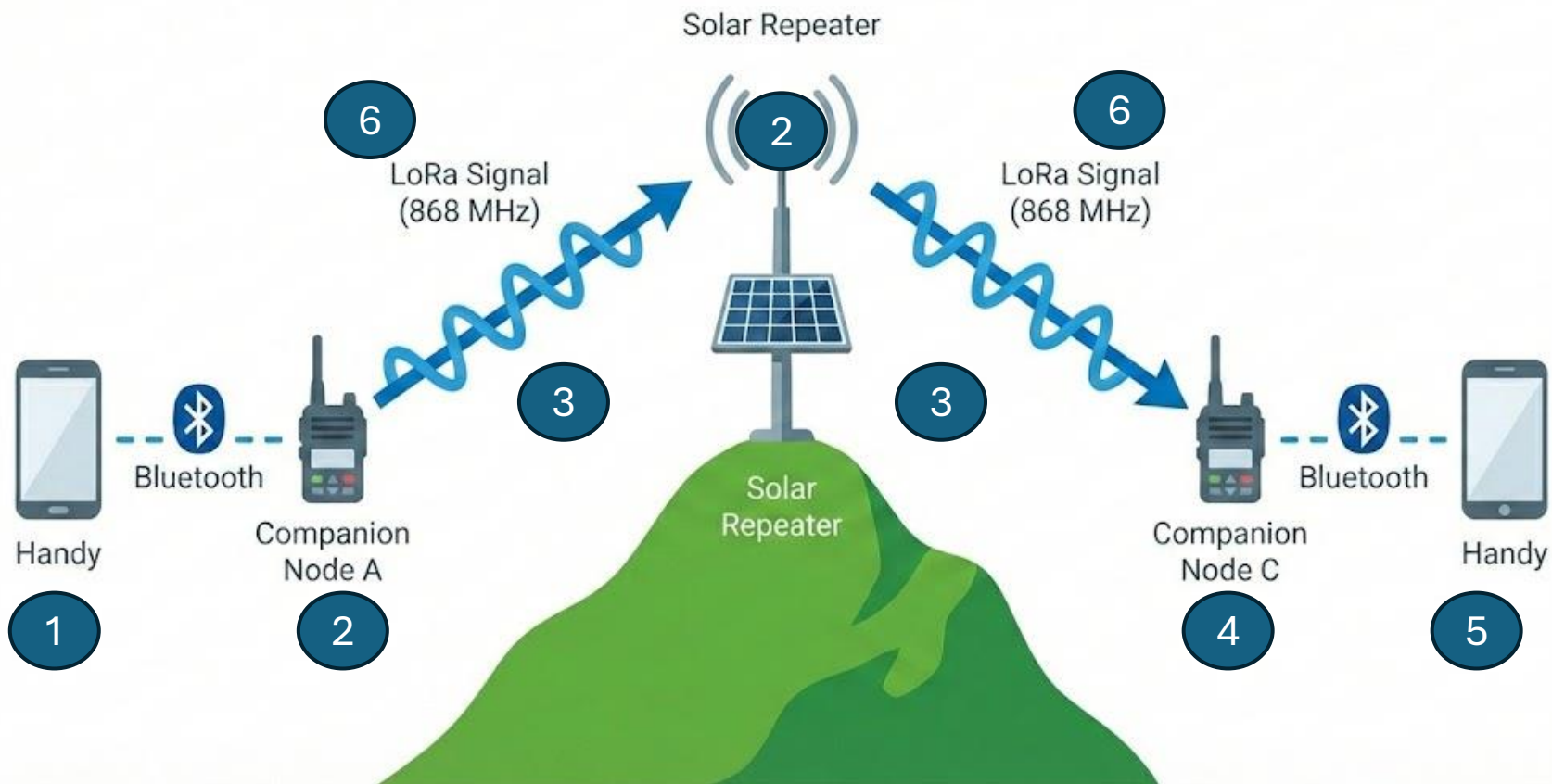
Aufgabe: Reichweitenvergrößerung: Er empfängt Pakete von schwächeren Companion Nodes und sendet sie mit voller Kraft erneut aus.

- **Netzstabilität:** Er sorgt dafür, dass Nachrichten "um Hindernisse herum" geleitet werden.

Energie: Meist auf Dauerbetrieb ausgelegt (Solarpanel + großer Pufferakku).

Wie wandert die Nachricht technisch durch das Netz?

Vereinfachte Darstellung



1. Die Paketierung (Encapsulation)
2. Das "Listen Before Talk" (Kollisionsvermeidung)
3. Die Ausbreitung via "Flood Routing",
4. Zielerreichung und De-Duplizierung
5. Dekapsulierung
6. Die Bestätigung (Acknowledgment - ACK)

1

Um zu verstehen, wie die Nachricht "wandert", müssen wir uns ansehen, was im Inneren der Hardware passiert, während die Funkwellen fliegen.

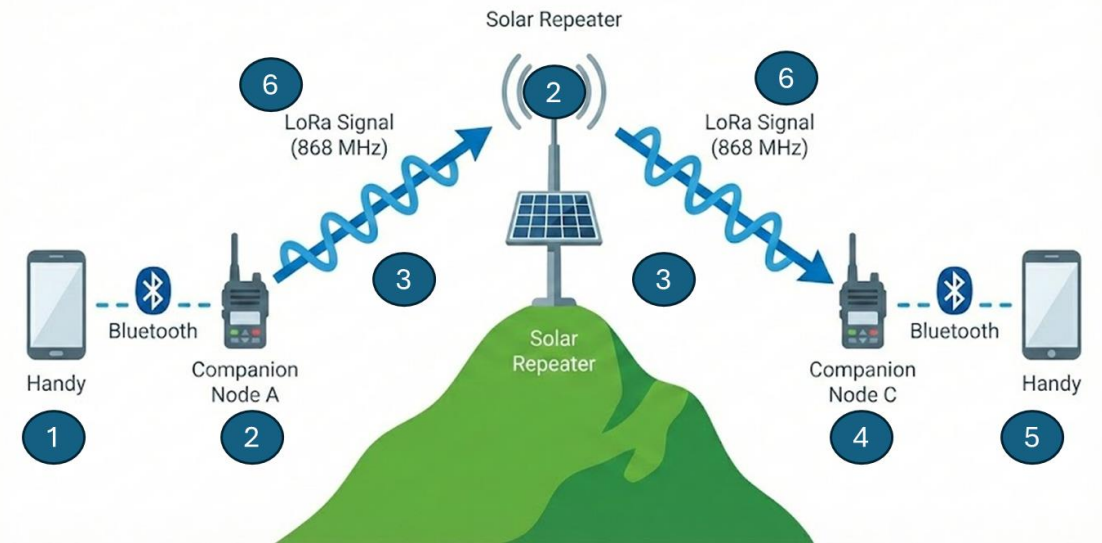
1. Die Paketierung (Encapsulation)

Bevor die Nachricht gesendet wird, muss sie "verpackt" werden. Das Smartphone schickt den Text via Bluetooth an den Node. Dieser erstellt ein Datenpaket:

Header: Enthält Metadaten (Wer sendet? Wer soll empfangen? Paket-ID).

Payload: Der eigentliche Inhalt (deine Nachricht), meist mit AES-256 verschlüsselt.

CRC (Checksum): Eine Prüfsumme, damit der Empfänger später feststellen kann, ob das Paket beim Transport beschädigt wurde.



2

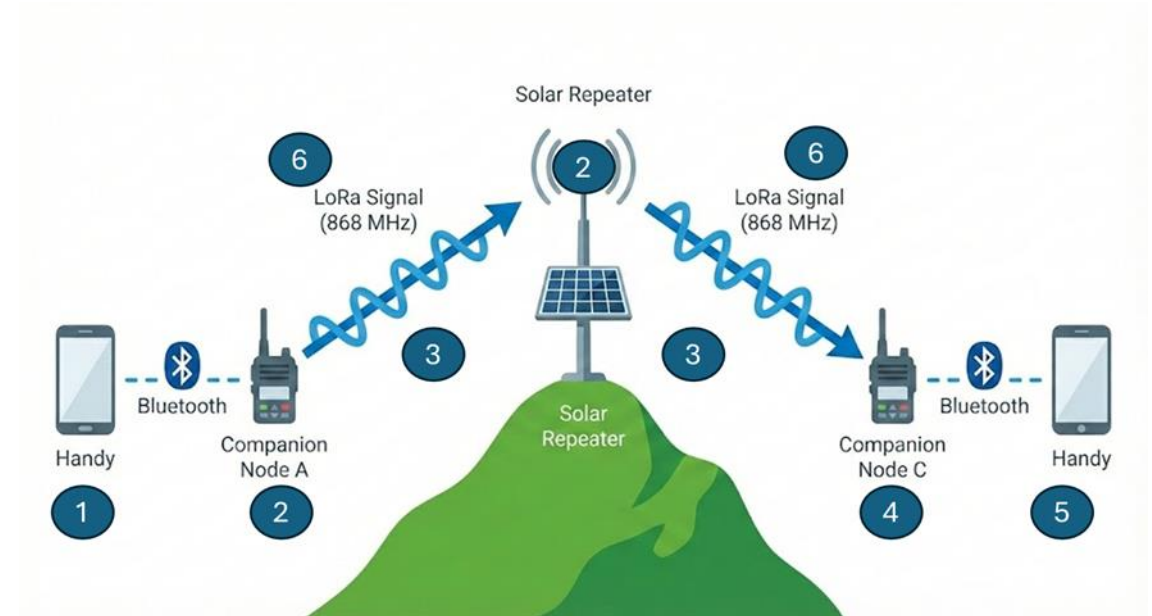
Das "Listen Before Talk" (Kollisionsvermeidung)

Bevor der Node sendet, hört er kurz in das Frequenzband hinein (CAD - Channel Activity Detection).

Ist der Kanal frei? Dann wird gesendet.

Ist der Kanal belegt? Der Node wartet eine zufällige Zeitspanne (Random Backoff) und versucht es erneut.

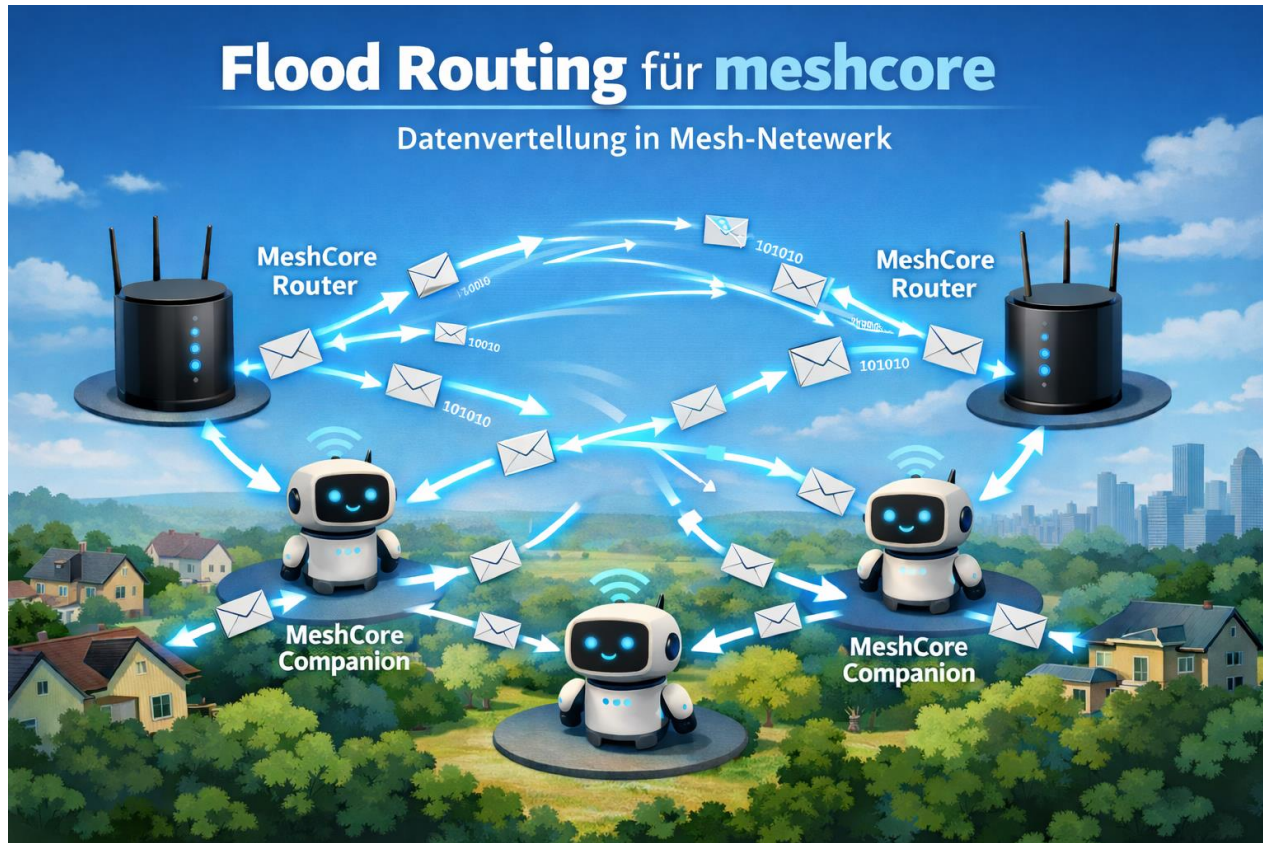
Das verhindert, dass sich zwei Nodes gegenseitig "überschreien".



3

Die Ausbreitung via "Flood Routing,,

Aussendung: Node A sendet das Paket als "Broadcast" in alle Richtungen



Empfang & Prüfung:

Jeder Node in Reichweite empfängt das Paket und prüft: Habe ich dieses Paket (anhand der Paket-ID) schon einmal gesehen? **(Wenn ja: Ignorieren).**

Ist das Hop-Limit größer als 0? (Wenn ja: Weiter zum nächsten Schritt).

Weiterleitung: Der Node dekrementiert (verringert) das Hop-Limit um 1 und sendet das Paket nach einer kurzen, zufälligen Verzögerung erneut aus.

Warum die Verzögerung? Wenn alle Repeater exakt gleichzeitig antworten würden, käme es zu Funkstörungen. Durch Millisekunden-Verzögerungen "staffeln" sich die Nachrichten.

4

5

Zielerreichung, De-Duplizierung, und Dekapsulierung

Da die Nachricht wie eine Welle durch das Netz "flutet", erreicht sie den Ziel-Node oft über mehrere Wege gleichzeitig oder kurz hintereinander.

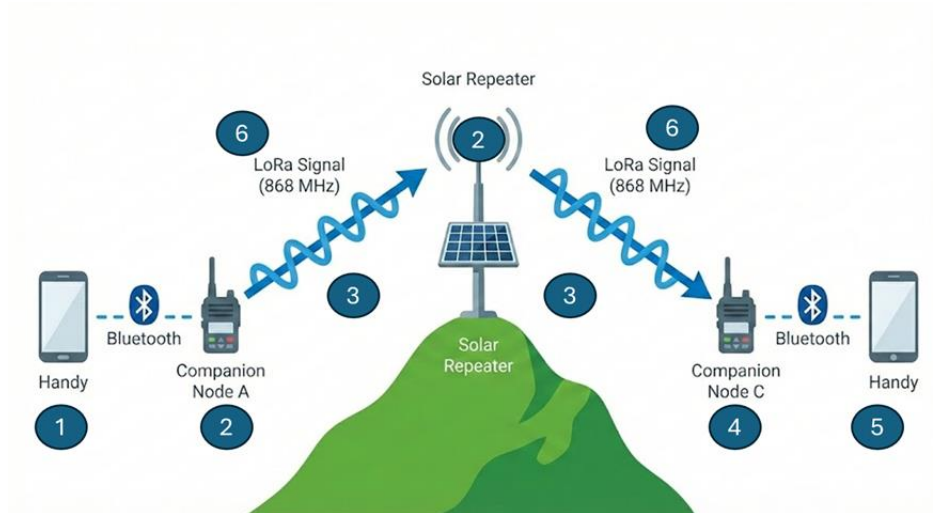
De-Duplizierung:

Der Ziel-Node erkennt an der ID, dass es sich um dieselbe Nachricht handelt. Er verarbeitet nur das erste fehlerfreie Paket und verwirft die Kopien, die Bruchteile von Sekunden später eintreffen.

Dekapsulierung: Der Node entschlüsselt die Payload und schickt sie via Bluetooth an das Handy des Empfängers.

Beispiel-Payload:

2C20D7D26A2EEC409896B062DFBD8CB39A725CEA



6

Bestätigung - (Acknowledgment - ACK)

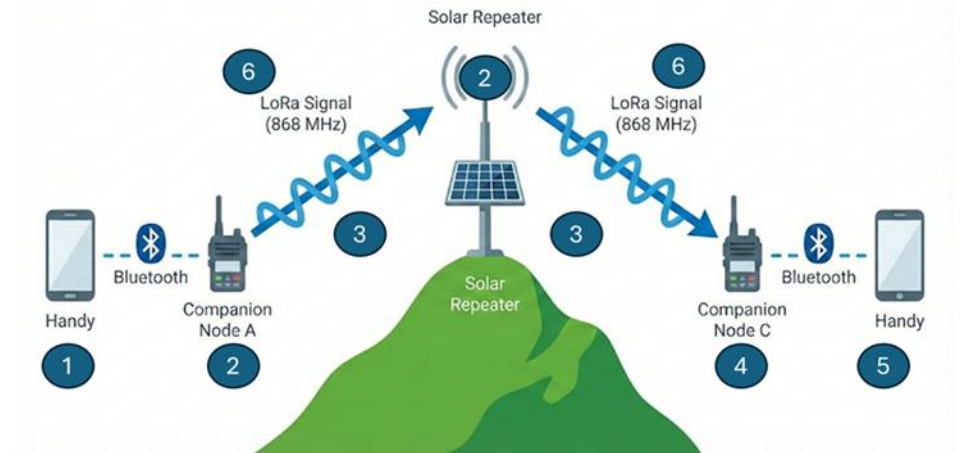
Die Bestätigung (Acknowledgment - ACK)

Damit du sicher sein kannst, dass die Nachricht angekommen ist, sendet der Ziel-Node ein winziges **ACK-Paket** zurück.

Dieses wandert auf demselben Weg (via Mesh-Hops) zurück zum Absender.

In deiner App erscheint dann Delivered oder Heard x Repeats neben der Nachricht.

Meshcore ist "Air-Only", d. h. es erfolgt keine Anbindung an MQTT. Deshalb findest du in der grundlegenden Konfiguration von Meshcore keine Felder für "WLAN-Passwort" oder "Gateway-IP". Es ist technisch blind für das Internet.



Freuenzen und Ausbreitung - Physik ist bei LoRa alles

Frequenz: In Europa nutzen wir das lizenzfreie **868 MHz Band**.

Die Ausbreitung: Quasi-optisch. Das bedeutet, das Signal verhält sich fast wie Licht.

Sichtverbindung (LoS - Line of Sight): Das Wichtigste für hohe Reichweiten.

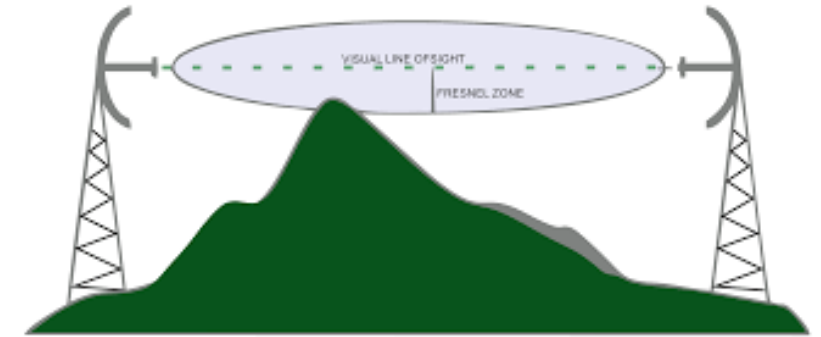
Hinweis: In der Bluetooth-App gibt es ein Tool, um sich mögliche Hindernisse bei der Funkstrecke anzeigen zu lassen.

Fresnel-Zone: Es reicht nicht, dass man das Ziel "sieht"; ein ellipsenförmiger Bereich um die Sichtlinie muss frei von Hindernissen sein, um Signalverluste durch Beugung zu vermeiden.

Die Fresnel-Zone ist einer der am häufigsten unterschätzten Faktoren bei der Planung von LoRa-Links. Viele Anfänger denken: "*Ich kann den anderen Node sehen, also habe ich Empfang.*" Das ist ein Trugschluss.

Was ist die Fresnel-Zone?

Stell dir die Funkverbindung zwischen zwei Antennen nicht als dünnen Laserstrahl vor, sondern als ein ellipsenförmiges (zigarrenförmiges) Kraftfeld. Obwohl die direkte Sichtlinie (Line of Sight - LoS) frei sein kann, wird das Signal abgeschwächt oder sogar ausgelöscht, wenn Hindernisse in diesen ellipsenförmigen Bereich ragen.



Das Funksignal breitet sich als Welle aus. Wenn ein Teil der Welle auf ein Hindernis (z. B. eine Hauskante oder den Boden) innerhalb der Fresnel-Zone trifft:

- Dann wird dieser Teil der Welle reflektiert. Er legt einen etwas längeren Weg zurück als der direkte Strahl.
- Er kommt am Empfänger leicht zeitversetzt an.

Praxisbeispiel Fresnel-Zone

Nehmen wir eine Strecke von **5 km** bei **868 MHz**:
Die Fresnel-Zone hat in der Mitte einen Radius von ca. **20 Metern**.

Wenn deine Antennen auf 2 Meter hohen Masten im flachen Feld stehen, ragt der Erdboden massiv in die Fresnel-Zone hinein.

Ergebnis: Obwohl du den anderen Node theoretisch sehen könntest, wird die Reichweite drastisch sinken.

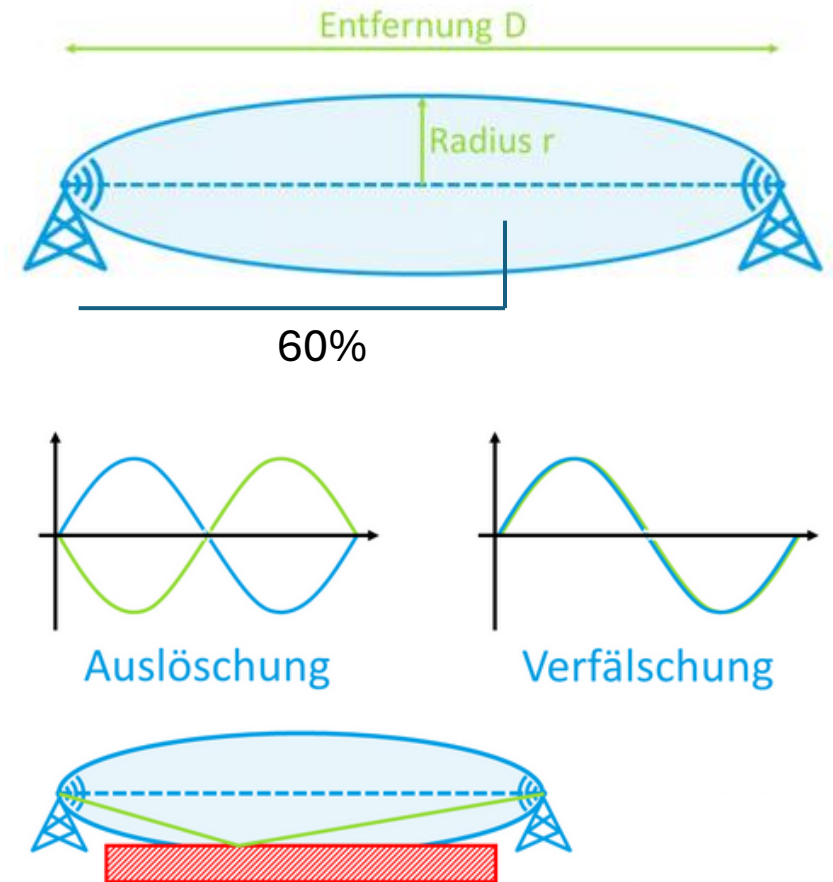
Lösung: Antennen so hoch wie möglich platzieren, um die Fresnel-Zone vom Boden und von Hindernissen "wegzuheben".

Das Problem: Wenn das reflektierte Signal "phasenverschoben" eintrifft, kann es das Hauptsignal auslöschen (Interferenz). Das ist so, als würde man gleichzeitig "Ja" und "Nein" rufen – beim Empfänger kommt nur Rauschen an.

Die 60%-Regel

Für eine stabile LoRa-Verbindung muss nicht die gesamte Ellipse frei sein, aber mindestens 60 % der ersten Fresnel-Zone müssen völlig frei von Hindernissen sein.

Sobald mehr als 20 % blockiert sind, treten signifikante Signalverluste auf.



Dämpfung, Antennenhöhe und Überreichweiten

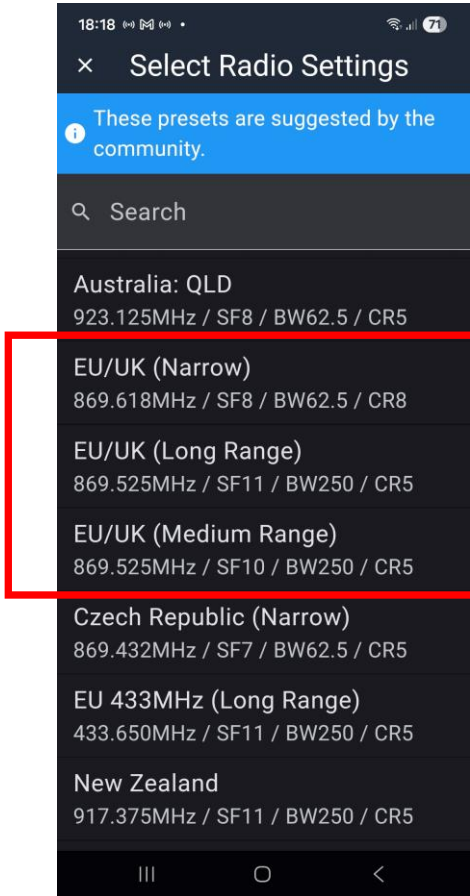
Dämpfung (Signal-Killer):

- **Feuchtigkeit:** Nasses Laub (Wälder im Regen) dämpft 868 MHz stark.
- **Beton/Stahl:** Gebäude blockieren das Signal fast vollständig.
- **Erde:** Hügel zwischen zwei Nodes sind unüberwindbar.

Antennenhöhe: Höhe ist durch nichts zu ersetzen außer durch noch mehr Höhe.

Überreichweiten (Troposphärische Überreichweiten): Bei bestimmten Wetterlagen (Inversionswetterlagen) können Signale an Luftschichten reflektiert werden und hunderte Kilometer weit reisen. Das ist bei LoRa selten, aber möglich.

Presets – Wann welches nutzen?



Meshcore bietet verschiedene Konfigurationen, die die Balance zwischen Reichweite und Geschwindigkeit halten.

Einstellung	Frequenz	BW (kHz)	SF	CR	Optimiert für
EU/UK (Narrow)	869.618 MHz	62.5	8	8	Störfestigkeit & stabile Verbindungen
EU/UK (Medium Range)	869.525 MHz	250	10	5	Balance zwischen Reichweite und Datenrate
EU/UK (Long Range)	869.525 MHz	250	11	5	Maximale Reichweite & Reichweite durch Hindernisse

LoRa-Einstellungsempfehlungen nach Umgebung

Umgebungstyp	Typische Reichweite	Empfohlene MeshCore-Einstellung	Optimale Parameter
Offenes Land (Sichtverbindung)	Bis zu 15+ km	EU/UK (Long Range)	Hoher SF (11), Große BW (250kHz)
Vorstädtisch/Normal	5 – 10 km	EU/UK (Medium Range)	Mittlerer SF (10), Große BW (250kHz)
Dicht bebaut/Städtisch	2 – 5 km	EU/UK (Narrow)	Niedriger SF (8), Schmale BW (62.5kHz)
Innerhalb von Gebäuden	0.5 – 2 km	EU/UK (Narrow)	Hohe Störfestigkeit ist hier entscheidend

Reichweite hängt stark vom Spreizfaktor (SF) ab: Je höher der SF, desto weiter die Reichweite, aber desto langsamer die Datenrate. Störfestigkeit hängt von der Bandbreite (BW) ab: Eine schmale Bandbreite (62.5 kHz) ist robuster in "verrauschten" oder hindernisreichen Gebieten, während eine große Bandbreite (250 kHz) höhere Datenraten in klaren Sichtverbindungen ermöglicht.

Erklärung der Parameter im Detail

Frequenz (MHz):

Die Frequenz von 869,618 MHz im "Narrow"-Modus liegt in einem spezifischen, regulatorisch festgelegten Bereich für Low-Power-ISM-Anwendungen, was Störungen minimiert. Die anderen Modi nutzen 869,525 MHz.

Bandbreite (BW): Die Bandbreite bestimmt, wie viel Frequenzspektrum genutzt wird.

62.5 kHz (Narrow) bietet eine hohe Störfestigkeit und ist gut für bewaldete oder hügelige Gebiete geeignet.

250 kHz (Medium/Long Range) ermöglicht eine höhere Datenrate, ist aber anfälliger für Störungen in dicht besiedelten Gebieten.

Spreizfaktor (SF): Der SF bestimmt, wie viele Symbole zur Codierung der Daten verwendet werden.

Ein höherer SF (z. B. 11) verlängert die Reichweite und die Fähigkeit, Signale durch Rauschen zu filtern, reduziert aber die Datenrate und erhöht die Übertragungszeit. Ein niedrigerer SF (z. B. 8) bietet eine höhere Datenrate, aber geringere Reichweite.

Codierungsrate (CR): Die CR fügt Redundanz hinzu, um Rauschen zu widerstehen.

CR8 (entspricht 4/8 oder 1/2) bietet eine verbesserte Fehlerkorrektur und erhöht die Zuverlässigkeit der Verbindung bei Störungen, was nützlich für den stabilen "Narrow"-Betrieb ist.

CR5 (entspricht 4/5) bietet eine weniger robuste Fehlerkorrektur, was in den Modi mit größerer Bandbreite und Reichweite akzeptabel ist.

Kein Datenschutz in #Kanälen

In Bezug auf MeshCore ist beim Datenschutz in den Kanälen `#public` bzw. allen `#xxxx`-Kanälen höchste Vorsicht geboten. Es handelt sich um Öffentliche Kanäle, somit besteht keine Privatsphäre.

In den Kanälen `#public` und `#Kanälen` gibt es keinen Datenschutz im Sinne von Vertraulichkeit. Gemeinsamer Schlüssel: Diese Kanäle nutzen einen bekannten, öffentlichen Verschlüsselungsschlüssel, den jeder MeshCore-Teilnehmer besitzt.

Mitlesen für alle: Jede Nachricht, die du in diesen Kanälen sendest, kann von jedem Gerät in Funkreichweite (oder über Repeater im gesamten Mesh) empfangen und im Klartext gelesen werden.

Kanal-Hashtag	Secret Key (HEX)	Zweck
<code>#public</code>	8b3387e9c5cdea6ac9e5edbaa115cd72	Der Standard-Hauptkanal für alle.
<code>#test</code>	9cd8fcf22a47333b591d96a2b848b73f	Für Reichweitentests und Experimente.
<code>#pingpong</code>	237e090aff0dbb589ecb38cac65602ba	Spezieller Testkanal für Ping-Antworten.
<code>#switzerland</code>	8ad1ce57ad257627090ed28413c1f0b7	Nationaler offener Kanal der Schweiz.
<code>#wetter</code>	afc5276e0d18bdeddbda96dcd722b60e	Allgemeine Wetterinformationen.

Zweck der Kanäle am Beispiel `#public`:
Dient als allgemeiner Chatraum für alle Teilnehmer, um Kontakt aufzunehmen oder die Netzabdeckung zu prüfen.

Datenschutz-Risiken in #Kanälen

Metadaten & Standort: Wenn du Nachrichten sendest, können andere Teilnehmer oft sehen, über welchen Pfad (Repeater-Kette) die Nachricht kam. Je nach Konfiguration könnten auch Standortdaten übertragen werden, was Rückschlüsse auf deinen Aufenthaltsort zulässt.

Dauerhafte Speicherung: Da MeshCore dezentral arbeitet, gibt es zwar keinen zentralen Server, der alles mitloggt. Aber: Jeder einzelne Teilnehmer kann die empfangenen Nachrichten lokal auf seinem Handy oder Node speichern.

Empfehlung für vertrauliche Daten

Für private oder sensible Informationen solltest du private Kanäle erstellen. Diese nutzen einen individuellen, geheimen Schlüssel, den nur du und deine Kommunikationspartner kennen. Nur so ist eine echte Ende-zu-Ende-Verschlüsselung gewährleistet, bei der Außenstehende nur Zeichensalat sehen.

#Kanäle einrichten und/oder privaten Kanälen beitreten

Einem bestehenden Kanal beitreten:

Um einem existierenden privaten oder Hashtag-Kanal beizutreten, musst du den Secret Key von der Person erhalten, die den Kanal erstellt hat.

Private Kanäle: Der Ersteller kann dir den Key entweder als Textfolge schicken oder du scannst einen generierten QR-Code direkt von dessen Handy ab.

Einen eigenen Kanal erstellen (Key generieren)

Wenn du einen neuen Kanal für dich oder deine Gruppe anlegen willst, generiert die App den Secret Key automatisch für dich:

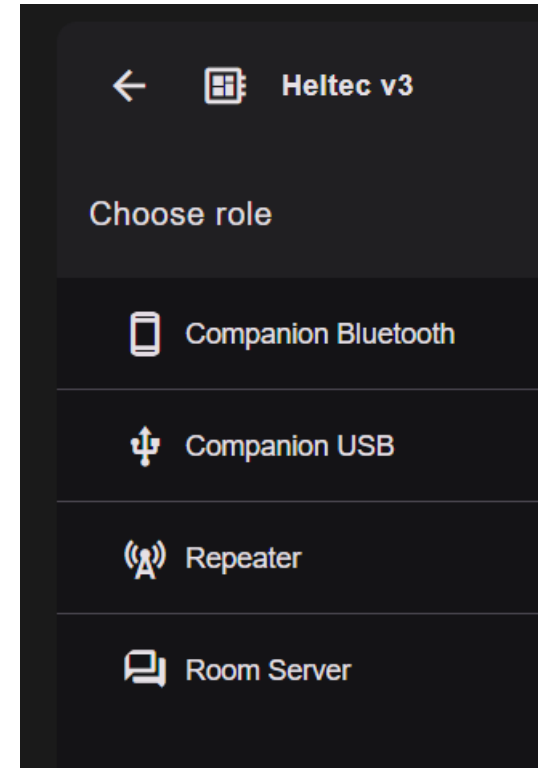
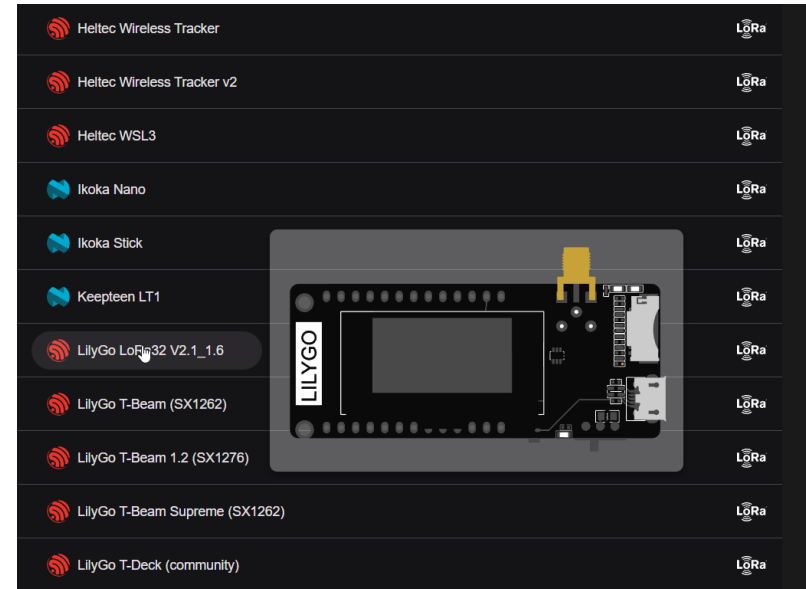
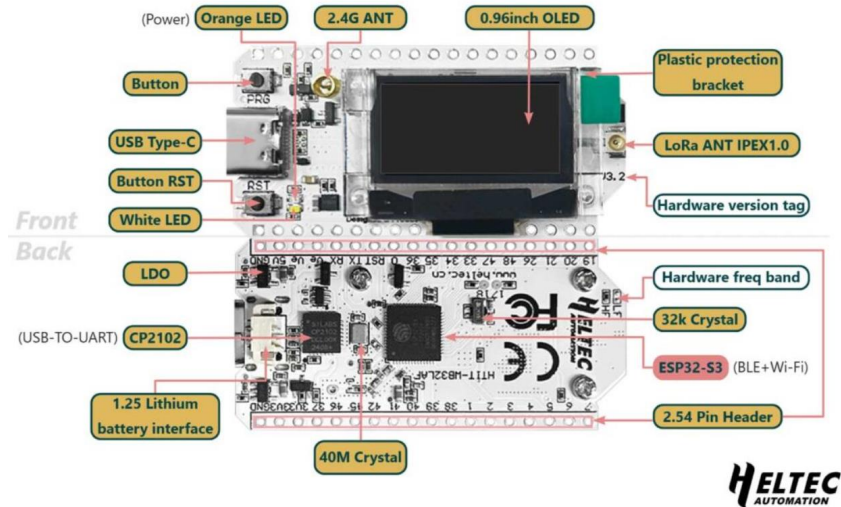
Öffne die MeshCore App und gehe zum Bereich Channels.

- Tippe auf das Plus-Icon oder "Add Channel".
- Wähle "Create Private Channel" (oder ähnlich, je nach App-Version).
- Gib dem Kanal einen Namen.
- Die App generiert nun automatisch einen 256-Bit Secret Key.
- Nach dem Speichern wird dir oft ein QR-Code angezeigt. Diesen können deine Freunde scannen, um automatisch beizutreten, ohne den langen Key manuell abtippen zu müssen.

Hardware und die richtigen Einstellungen für den Start

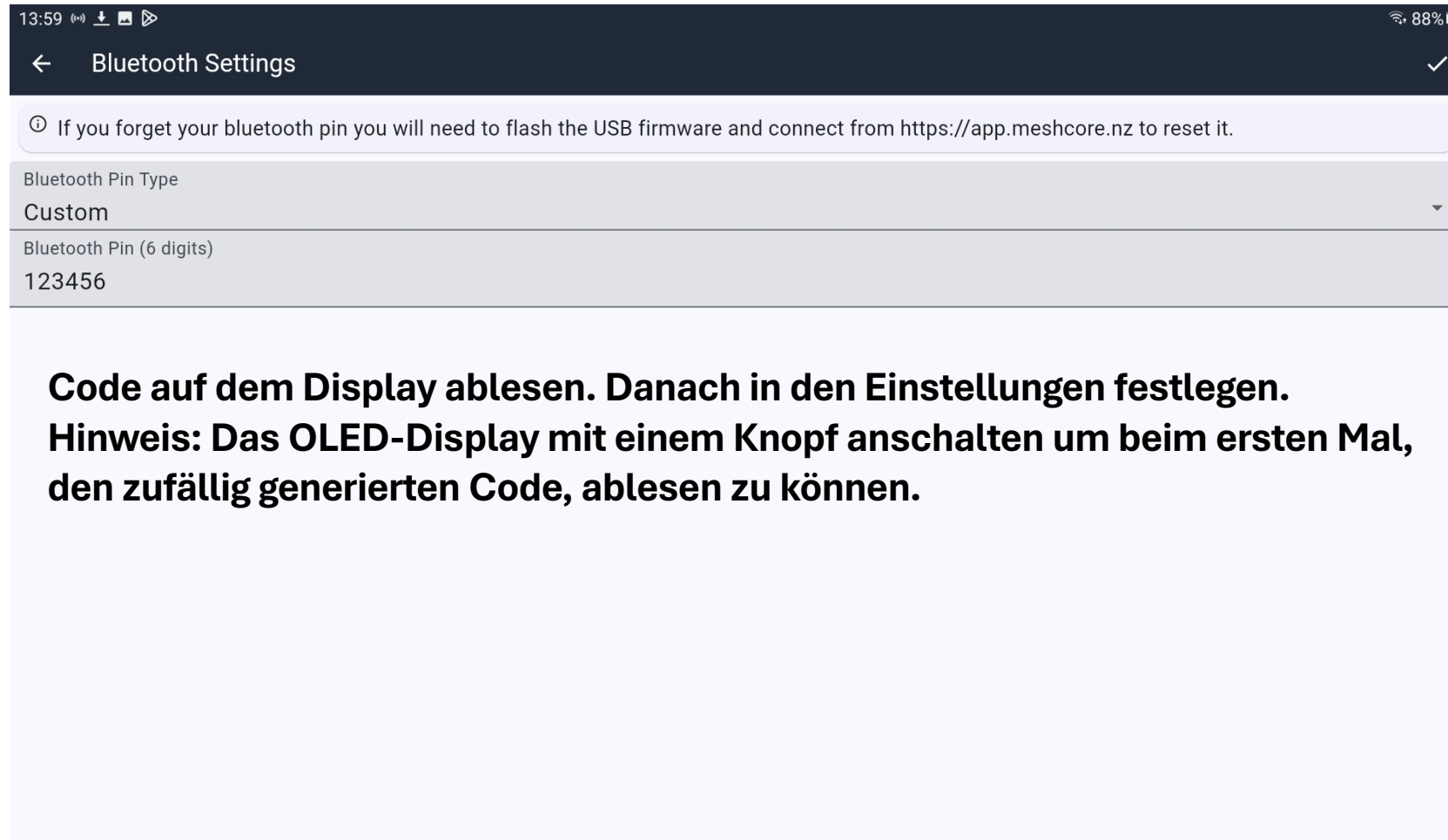
<https://flasher.meshcore.co.uk/>

WiFi LoRa 32 V3

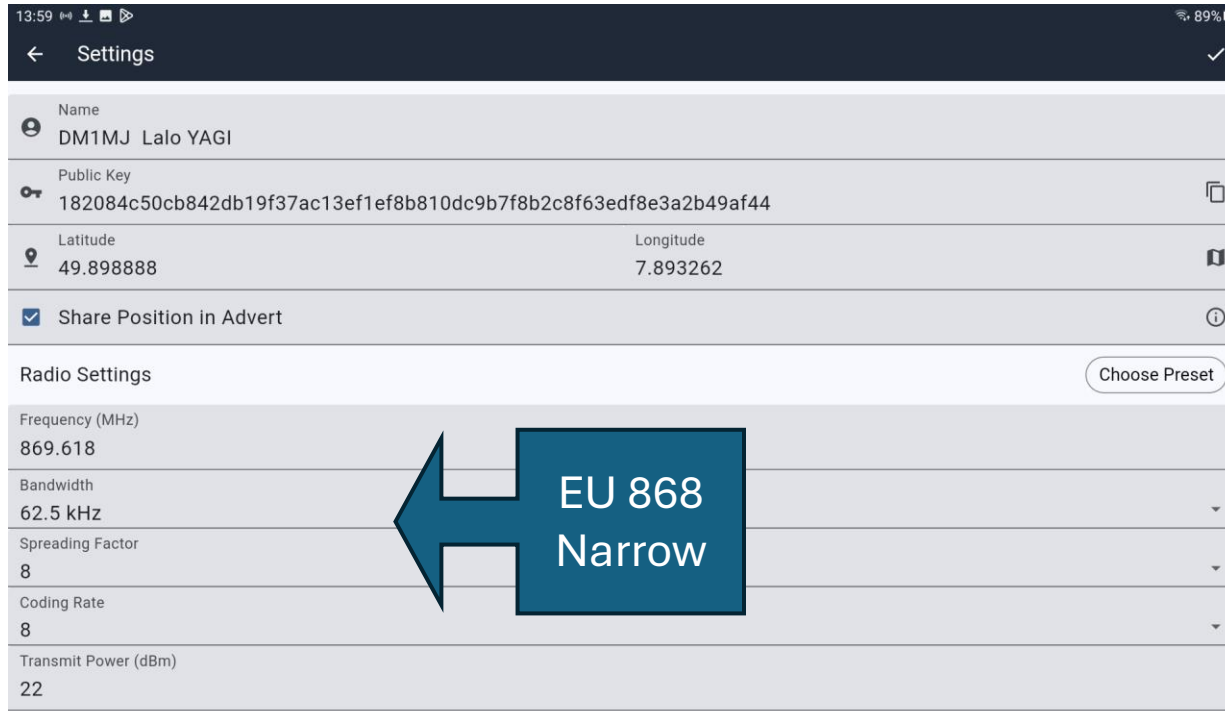


Beim Flashen weist man dem Node eine Rolle zu, die unterschiedliche Funktionen beinhaltet.
Router, Companion, Room Server.

Einstellungen am Beispiel Companion. Alles fängt mit der BT-Verbindung an. Code auf dem Display ablesen



Die Grundeinstellungen am Companion Node

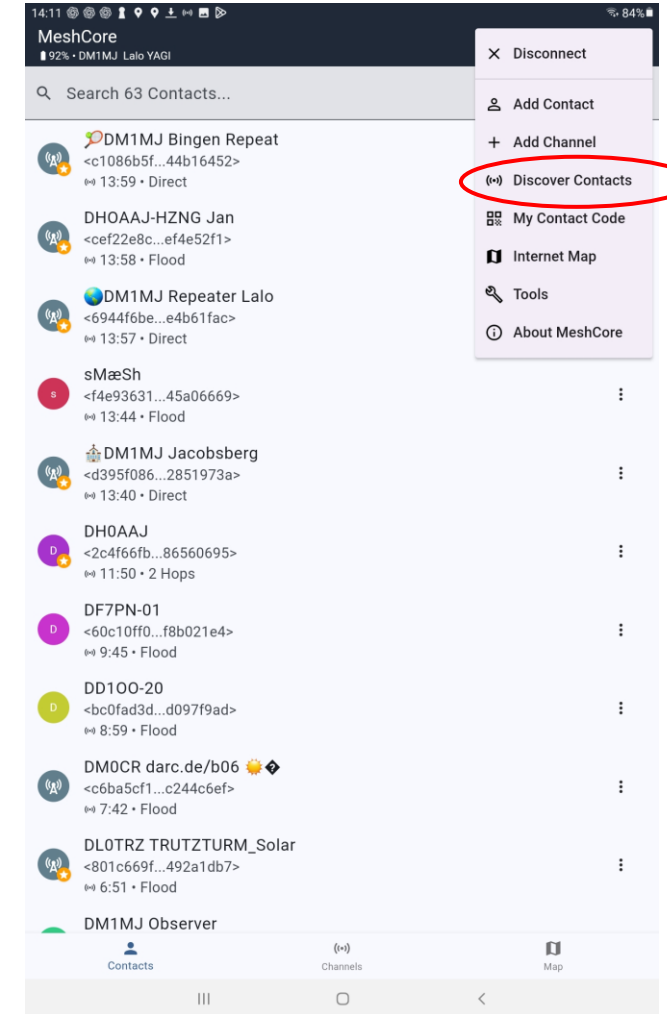
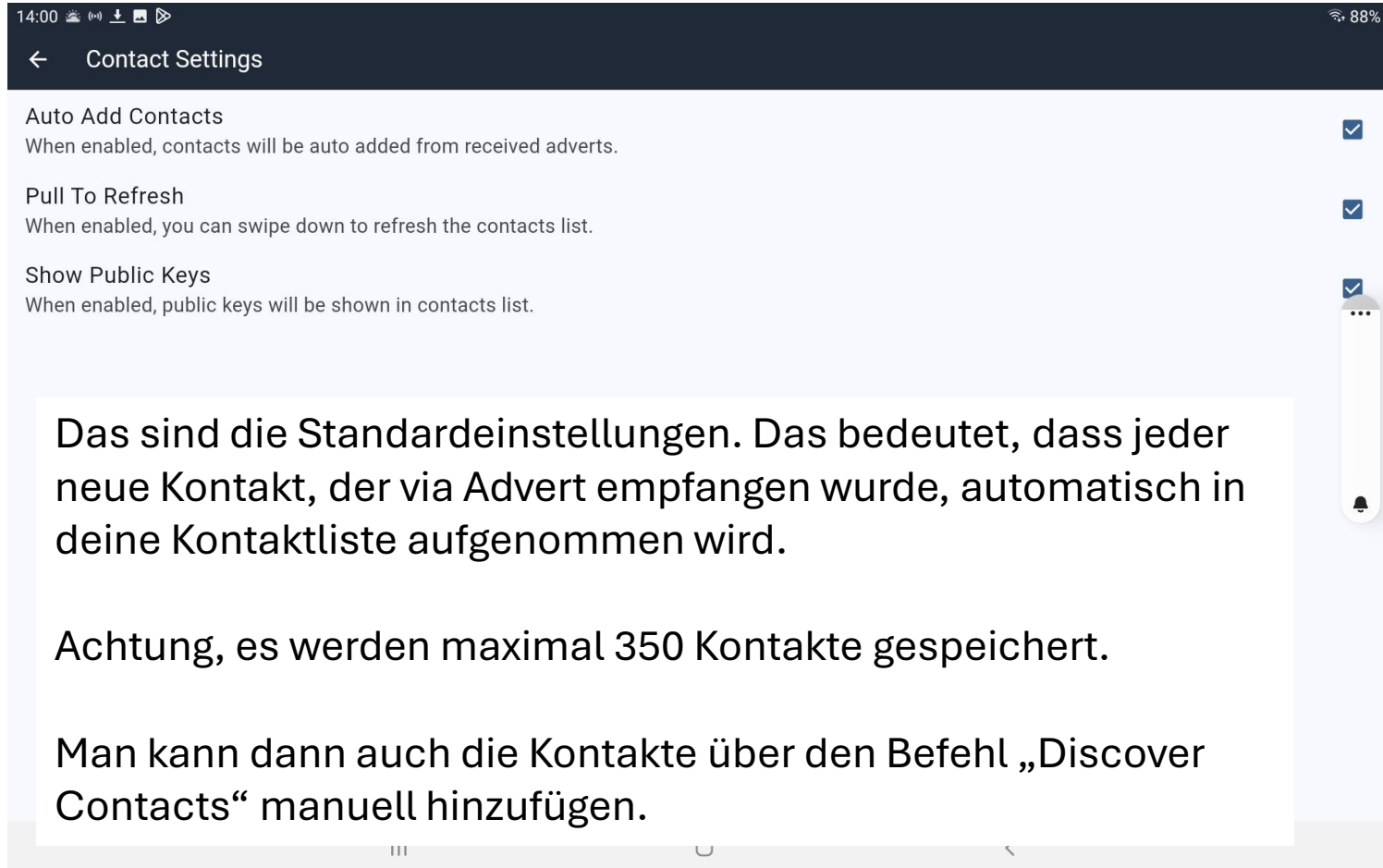


Der Public Key erfüllt bei der Ersteinrichtung deines MeshCore Companion Nodes vor allem drei Funktionen:

- Verschlüsselung von Direktnachrichten: Wenn dir jemand eine private Nachricht senden möchte, nutzt sein Gerät deinen Public Key, um die Daten zu verschlüsseln. Nur dein passender Private Key (der sicher auf deinem Gerät bleibt) kann diese Nachricht wieder lesbar machen.

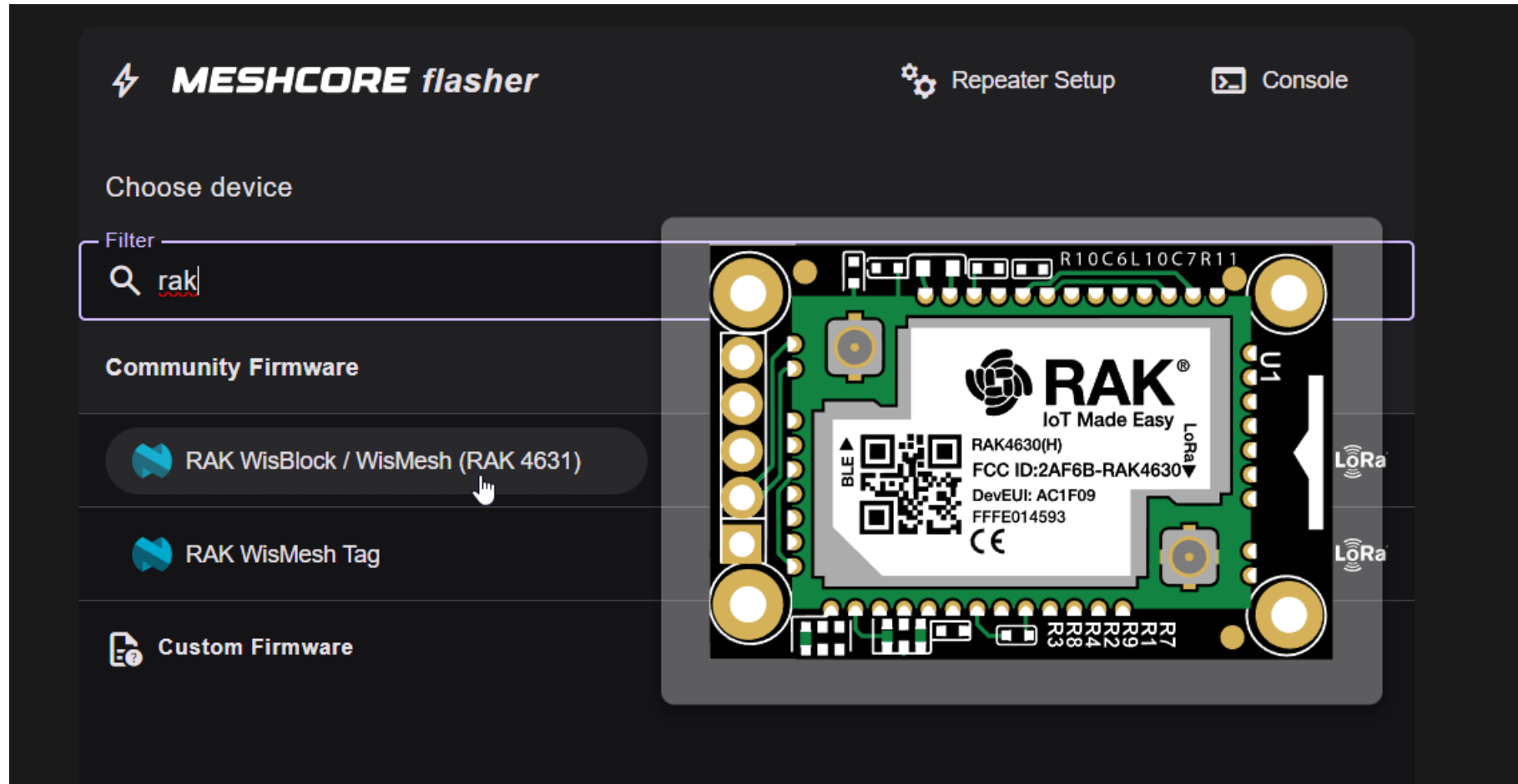
- Identifikation im Netzwerk: Durch das Senden eines sogenannten Adverts (einer "Bekanntmachung") teilt dein Node seinen Namen und seinen Public Key anderen Teilnehmern mit. So wissen andere Nodes, dass du online bist und wie sie dich sicher erreichen können.
- Authentifizierung & Admin-Rechte: Der Public Key wird genutzt, um sicherzustellen, dass Nachrichten wirklich von dir kommen (digitale Signatur). Er wird auch benötigt, wenn du einen anderen Node aus der Ferne administrieren möchtest – dort muss dein Public Key hinterlegt sein, damit der Remote-Node dich als Administrator akzeptiert

Kontakte managen, maximal 350 Kontakte möglich

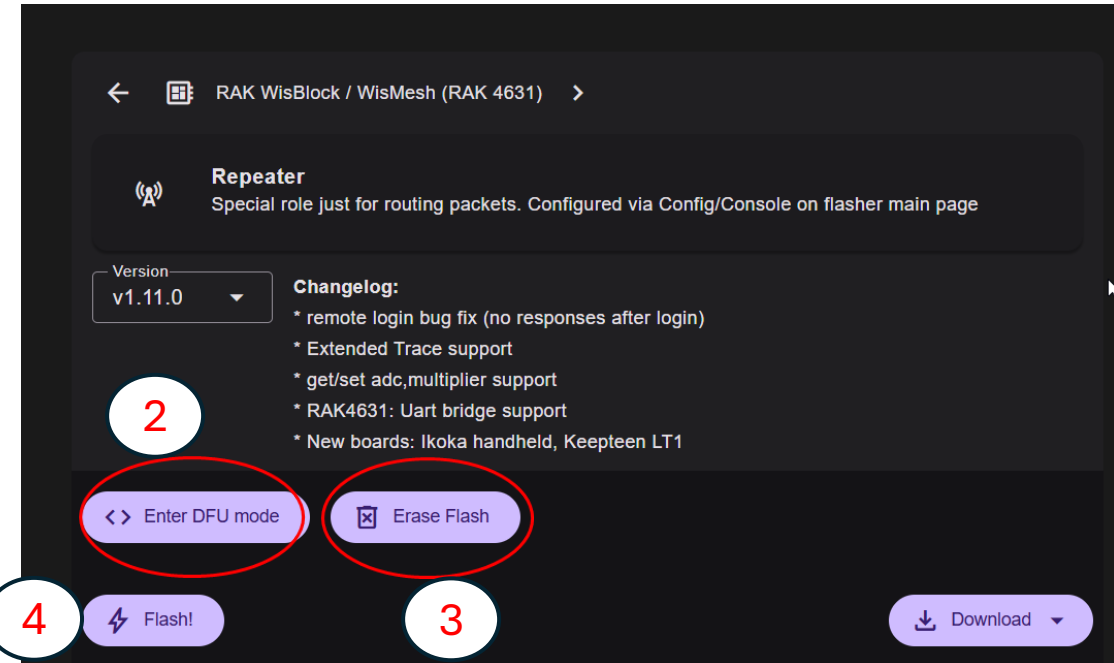
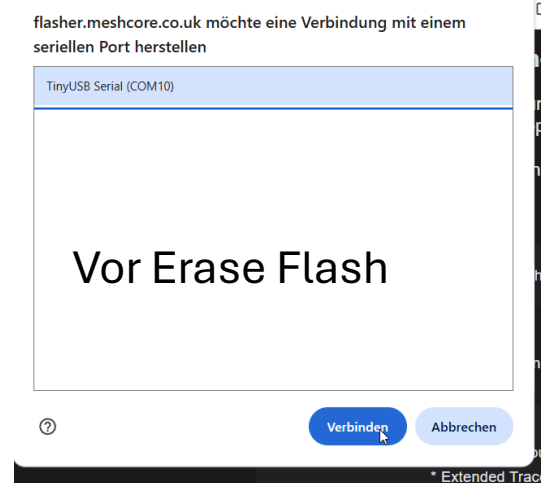
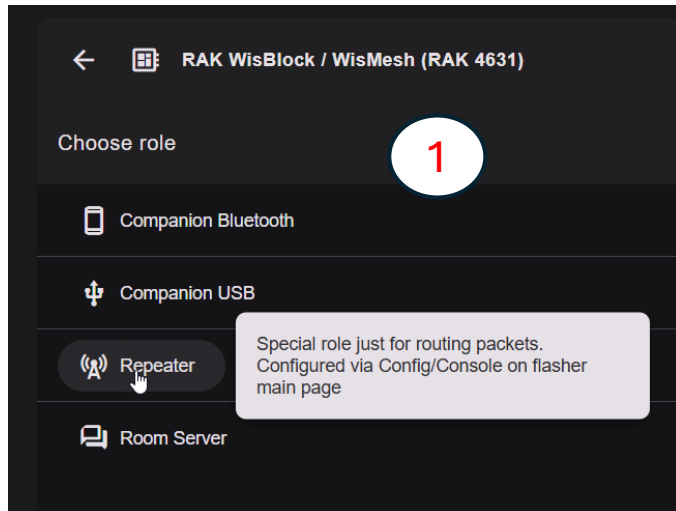


Einen RAK4631 Node als Router flashen und einstellen

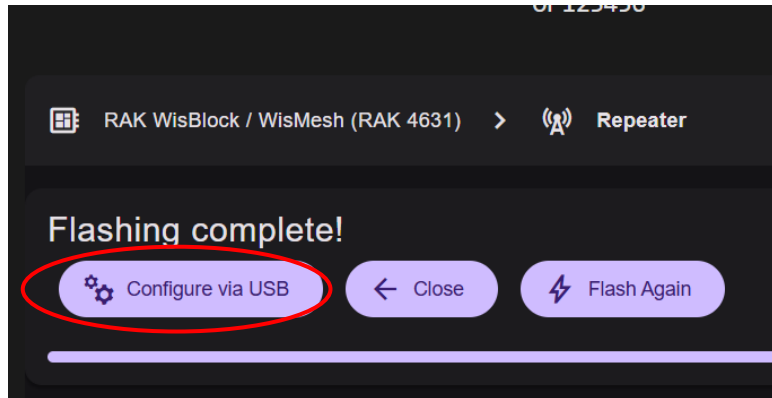
<https://flasher.meshcore.co.uk/>



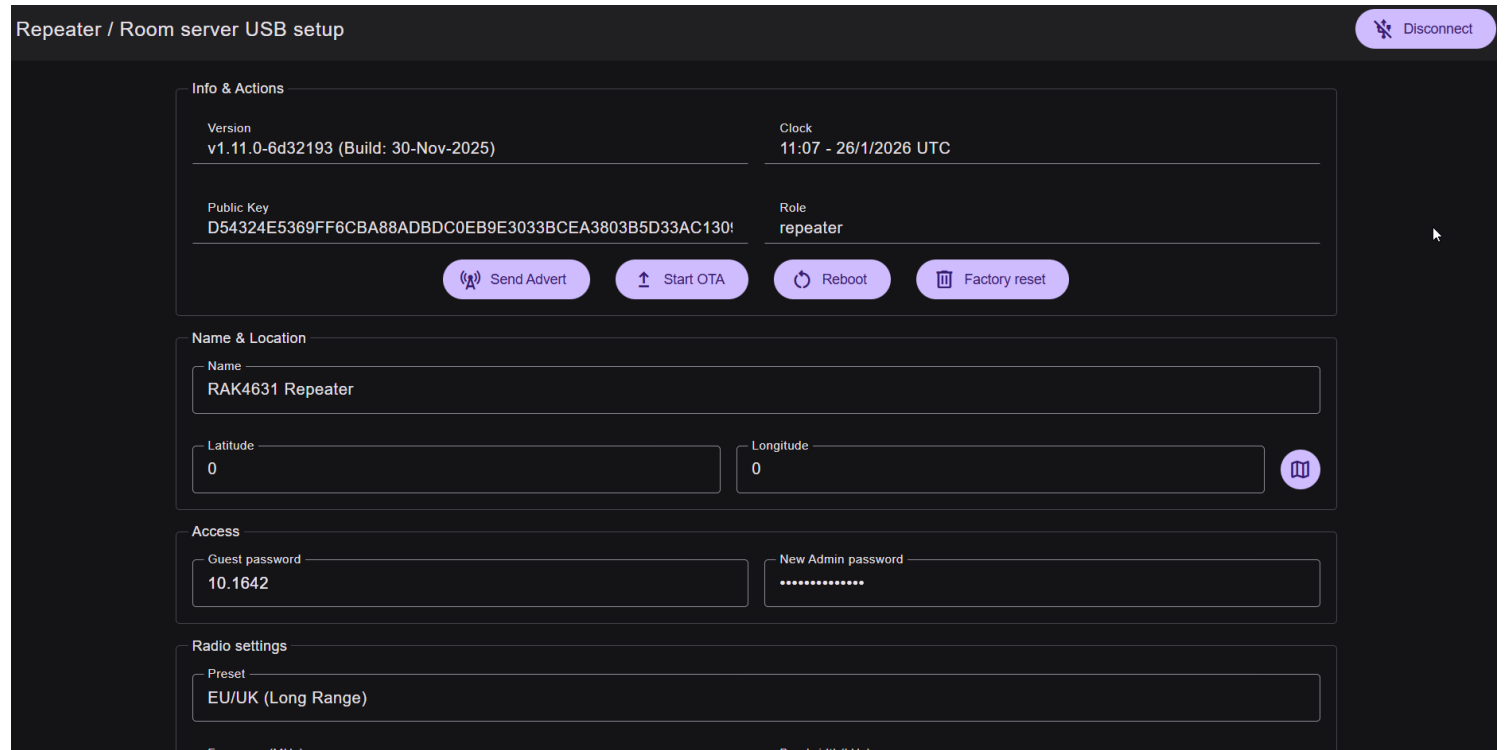
Die Rolle Repeater auswählen, Enter DFU Mode, Erase flash und zum Schluss Flash



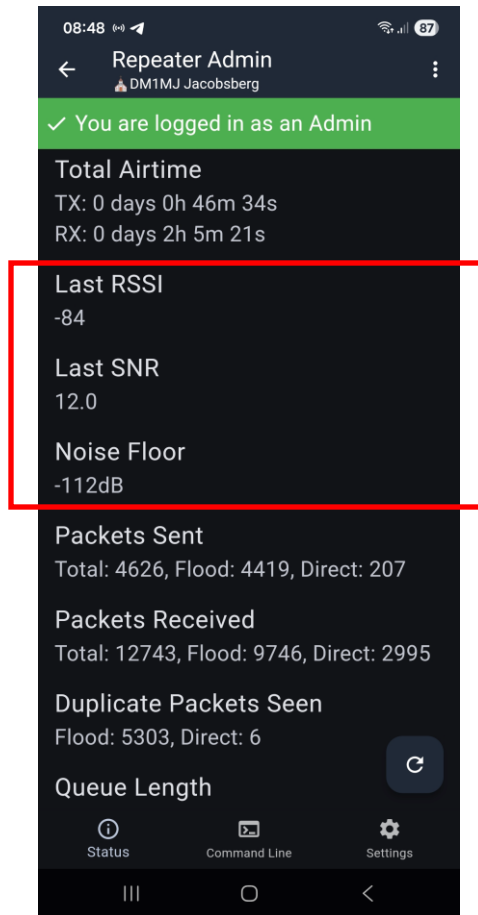
Konfiguration des Routers nur per USB am PC



Hier können dann die Einstellungen für den Router, ähnlich wie in der BT APP vorgenommen werden.



Erste Analysen aus der BT App heraus. RSSI, SNR, Noise Floor



Diese Werte sind das "Herzstück" der Verbindung:

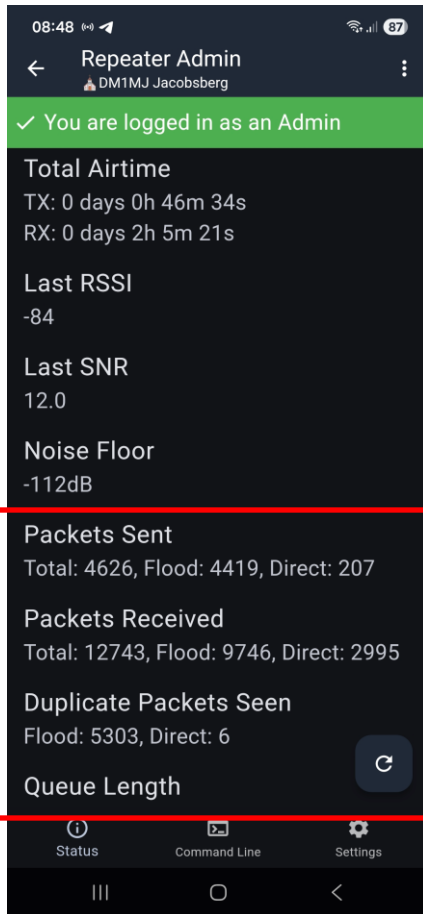
Last RSSI (-84 dBm): Dies ist ein guter Wert für eine LoRa-Verbindung. Er liegt weit über der Empfindlichkeitsgrenze (die oft bis -130 dBm oder tiefer geht). Das Signal ist stark genug für eine stabile Kommunikation.

Last SNR (12.0 dB): Ein Signal-Rausch-Verhältnis von 12 dB ist exzellent. LoRa kann Signale sogar noch dekodieren, wenn sie unter dem Rauschen liegen (negatives SNR). 12 dB zeigen eine sehr saubere, störungsfreie Verbindung zum letzten Absender.

Noise Floor (-112 dB): Ein typischer und guter Wert für eine ländliche oder halbstädtische Umgebung. Es gibt keine massiven lokalen Störquellen, die das Band "zumüllen".

Erste Analysen aus der App heraus.

Sende- und Empfangsstatistik (Traffic) TX vs. RX Airtime



Der Node hat etwa 46 Min. gesendet, aber über 2 Std. empfangen. Das ist typisch für einen Repeater. Er "hört" viel mehr, als er selbst aktiv weitergeben muss.

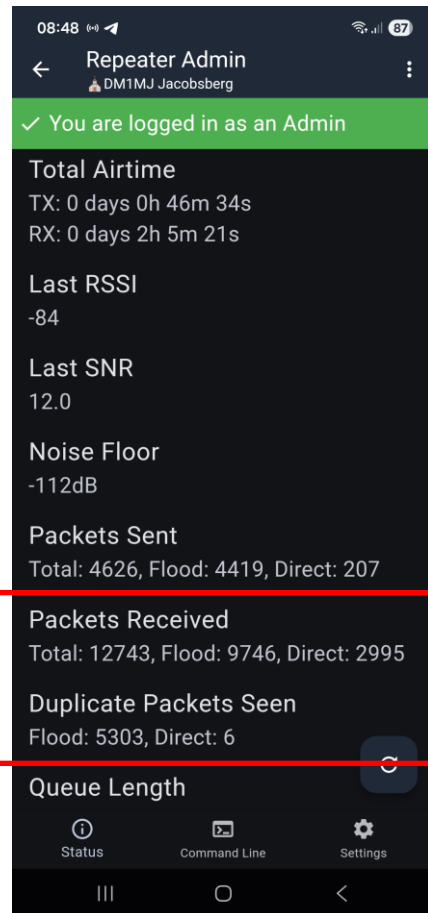
Packets Received (12.743): Der Node ist sehr aktiv und befindet sich offensichtlich in einem gut ausgebauten Mesh-Netzwerk.

Flood vs. Direct: Über 90% des Traffics sind "Flood"-Pakete. Das ist normal für Meshcore, da Nachrichten meist an alle (Broadcast) verteilt werden, um den Weg durch das Mesh zu finden.

Duplicate Packets (5.303): Fast die Hälfte der empfangenen Flood-Pakete sind Duplikate. Das Netzwerk um den Jacobsberg herum ist sehr dicht. Pakete erreichen den Node über mehrere Pfade gleichzeitig. Das ist einerseits gut für die Zuverlässigkeit, zeigt aber auch, dass das Netz "gut ausgelastet" ist.

Queue Length (Warteschlange): Alles bestens. Der Node ist so schnell, dass keine Nachrichten warten müssen. Sobald ein Paket reinkommt, wird es sofort verarbeitet oder weitergeleitet.

Erste Analysen aus der App heraus. Wie gesund ist das Netz?



Das Duplikat-Verhältnis: Dies ist der wichtigste Indikator für die Mesh-Gesundheit in deinem System.

Daten: **5.303 Duplikate** bei 9.746 Flood-Paketen.

Das ist für MeshCore ein sehr hoher Wert.

Analyse: Das bedeutet, dass ca. 54 % aller empfangenen Pakete diesen Node bereits über einen anderen Weg erreicht haben oder er sie schon kennt.

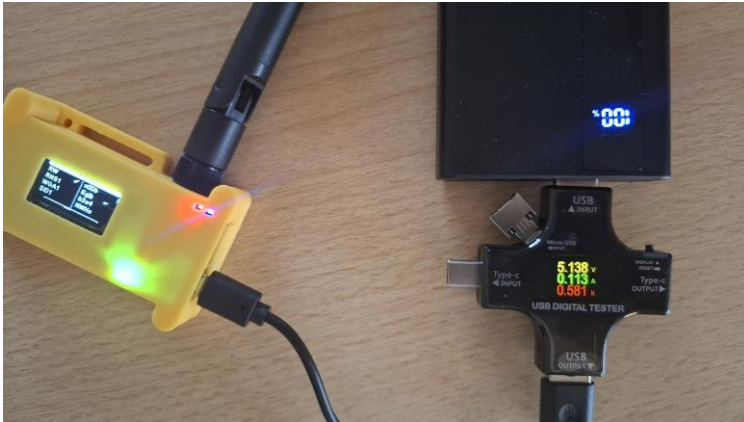
MeshCore-Status: Dein Node am Standort „Jacobsberg“ ist extrem gut vernetzt. In einem MeshCore-Algorithmus ist ein hohes Duplikat-Aufkommen ein Zeichen für eine hohe Redundanz. Sollte ein anderer Node in der Umgebung ausfallen, ist das Netz so stabil, dass die Pakete problemlos über alternative Pfade zu dir finden.

Er deutet darauf hin, dass der Node am Jacobsberg so zentral liegt, dass er "Flood"-Anfragen von vielen verschiedenen Seiten gleichzeitig hört.

Direct Packets (RX 2995): MeshCore ist exzellent darin, "Direct Paths" zu lernen. Wenn zwei Nodes sich einmal gefunden haben, schaltet der Core vom Fluten (Flood) auf direktes Routing um. Dass du so viele Direct Packets empfangst, zeigt, dass der MeshCore deines Nodes bereits viele feste Routen in seiner Routing-Tabelle etabliert hat.

Stromverbrauch bei verschiedenen Nodes

Beispiel: Lilygo T3 V1.6.1 FW Meshtastic – sehr stromhungrig



Beispielrechnung für den Anschluss an eine Powerbank mit 20.000 mAh:

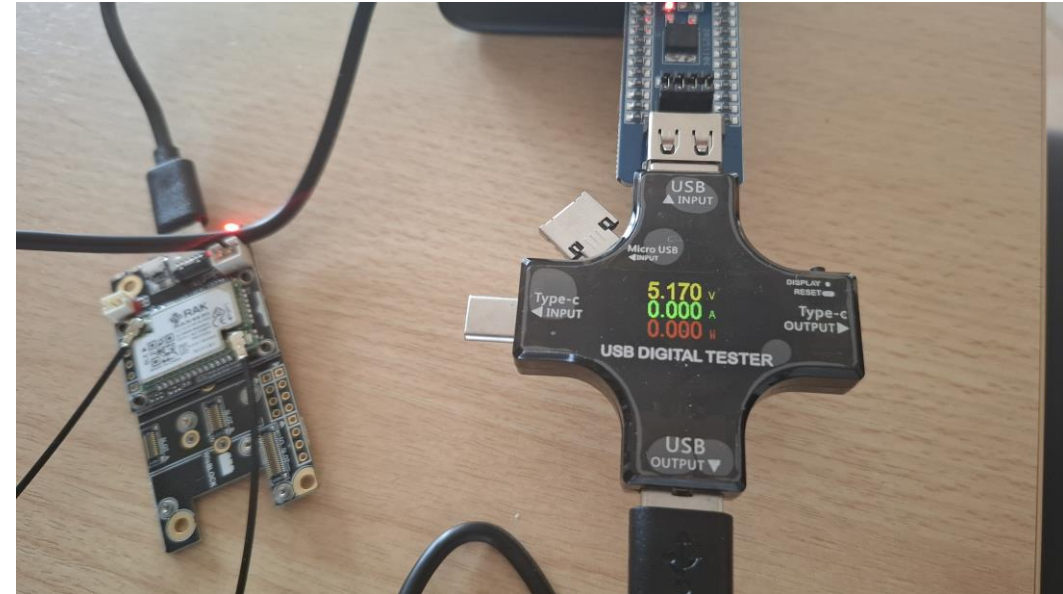
Wir nehmen einen realistischen Durchschnittsverbrauch von 90 mA an (Mischung aus RX/TX und eingeschaltetem Bluetooth). **Der Node hält etwa 6,5 Tage am Stück durch.** Dabei sind Verluste an der Powerbank eingerechnet.

$$\text{Laufzeit (Stunden)} = \frac{\text{Kapazität (mAh)}}{\text{Verbrauch (mA)}} = \frac{14.000 \text{ mAh}}{90 \text{ mA}} \approx 155 \text{ Stunden}$$

Beispielrechnung Stromverbrauch RAK4631 Repeater Node an einer Powerbank mit 20.000 mAh

Grundverbrauch (Idle/Listen): ca. 10 mA bis 12 mA.
Verbrauch bei hoher Last (RX/TX Mix): ca. 15 mA bis 25 mA
(da er ständig Pakete verarbeitet).
Powerbank-Netto-Kapazität mit einem Verlust durch 5V
Wandlung: 14.000 mAh effektive Kapazität.

Selbst wenn der Standort sehr aktiv ist, bleibt der
Durchschnittsverbrauch beim RAK extrem niedrig. Wir
nehmen einen „Stress-Durchschnitt“ von **20 mA** an:



$$\text{Laufzeit (Stunden)} = \frac{14.000 \text{ mAh}}{20 \text{ mA}} = 700 \text{ Stunden}$$

Hardware	Durchschnittsverbrauch	Laufzeit (20k mAh Powerbank)
LilyGO T3 (ESP32)	~90 mA	ca. 6,5 Tage
RAK4631 (nRF52)	~20 mA	ca. 29 Tage

Ergebnis: Der RAK-Node hält etwa 29 Tage durch.

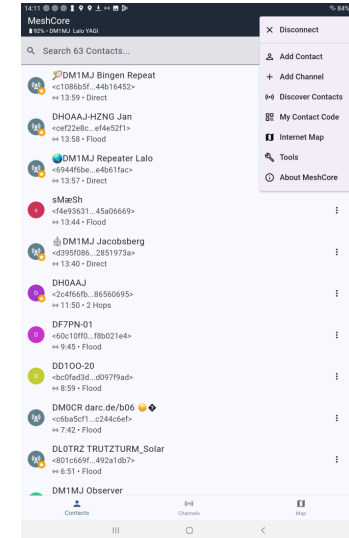
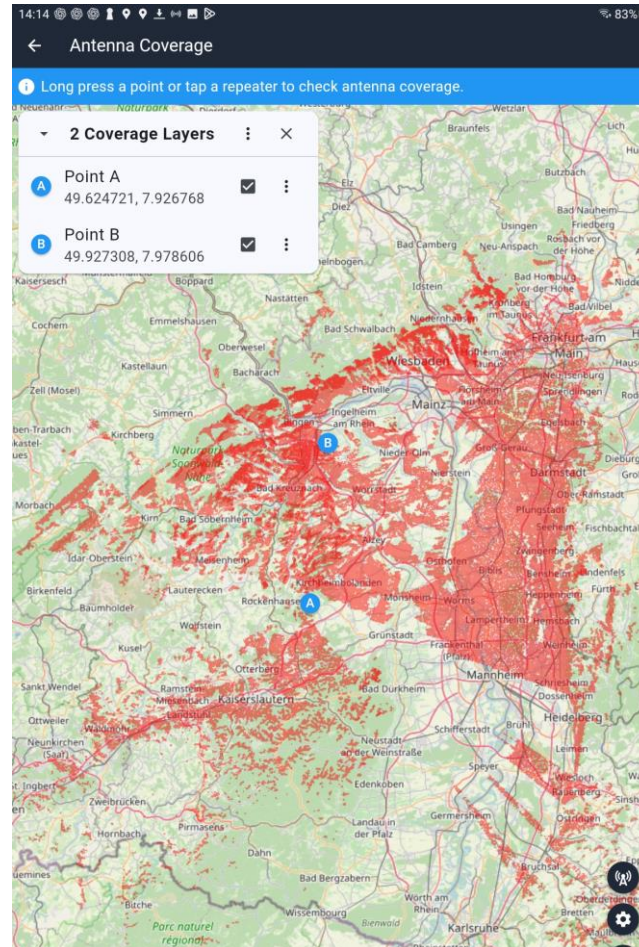
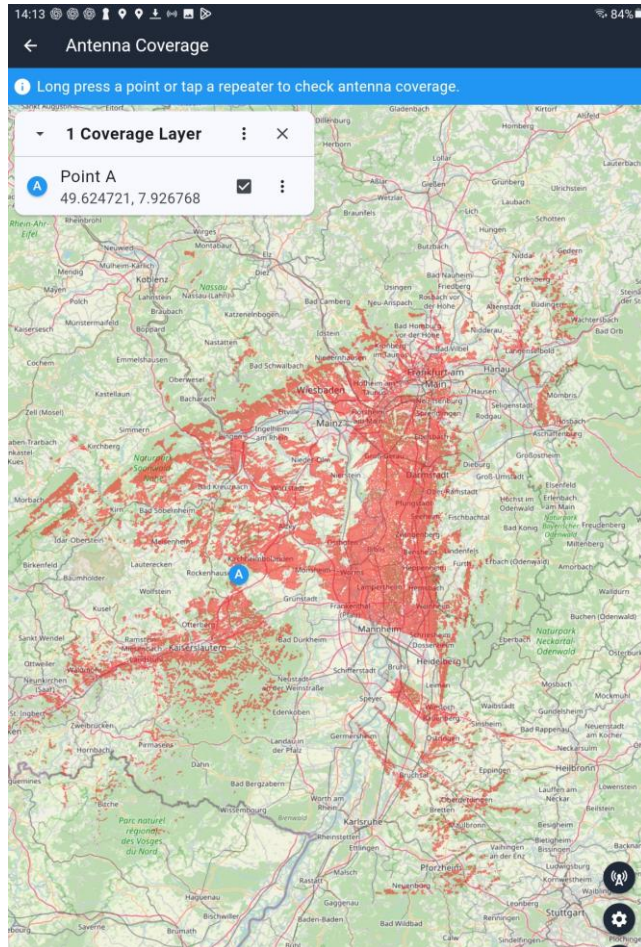
Tipps, um einen RAK 4631 Node mit einer Powerbank mit Strom dauerhaft zu versorgen



Virtuelles Last Anti-Abschaltmodul

Dieser USB-Stick hilft, dass die Powerbank eingeschaltet bleibt. Es kann auch als Last verwendet werden, um Strom zu ziehen und so zu verhindern, dass die Powerbank aufgrund unzureichender Stromaufnahme abgeschaltet wird.

Tools innerhalb der BLUE-App, für die Analyse, wie mein Node funktioniert : Tools – Antenna Coverage.



Mit dem Tool kann die Antennenreichweite simuliert werden.
Hier am Beispiel Donnersberg.

Kombiniert man das mit einem zweiten Standort, ist eine große Reichweite gezielt möglich.

Tools innerhalb der BLUE-App, für die Analyse, wie mein Node funktioniert : Tools –Showing Neighbors, Trace Path, Line of Sight.

