MeshCore

Complete Manual

Off-grid mesh communication

Text-based communication via LoRa radio

Version 2.0 - January 2026

By PE1HVH / ChatGPT / Claude.ai

Table of Contents

Foreword

My introduction to MeshCore coincided with my retirement a month ago. So I had all the time to dive into this. Which was also necessary, as I knew nothing about this at all.

Using the Internet and AI tools such as ChatGPT (OpenAI), Claude (Anthropic) and Perplexity.ai, I delved into the subject matter. While these tools helped me structure information, translate technical concepts into understandable language, and consistently format the document, the content choices and practical experience are my own.

I have started by purchasing the SenseCAP Solar Node, SenseCAP T1000-E, and the LilyGO T-Deck.

In my search for the information - which can mostly be found on the internet, but is very fragmented - it has been brought together here. Together with the experiences of installing and the problems I encountered, it has become this document.

This document has been translated dutch into English by Claude.ai.

Not all text within the images has been translated, but the assumption is that the images are reasonably self-explanatory."

Disclaimer

The AI tools used can hallucinate: this means they present information with great confidence that is incorrect. This is a well-known phenomenon in AI, caused by limitations in training data and statistical patterns that do not represent factual knowledge, or are simply wrong.

Although checking is done, errors may still occur. When in doubt, always consult the official MeshCore documentation. MeshCore is still actively in development. Information in this manual may therefore be outdated. If you have found errors, suggestions for improvements, or want to contribute, send an email.

73 de PE1HVH

Januari 2026

meshcore@pe1hvh.nl

Reading Guide

This manual is divided into four parts, each with its own focus:

Part A: User Manual

In addition to background, it provides information on how to use MeshCore. From initial installation to practical applications, hardware choices and group communication. Start here if you want to get started right away.

Part B: Technology Explained

For those who want to understand how LoRa and MeshCore work under the hood. From bits to chirps, dechirp analysis, modulation, demodulation, packet structures and advanced features such as remote control. Technical but accessibly written.

Part C: Abbreviations and Terminology

Alphabetical reference of all technical terms and abbreviations used in this manual.

Part D: References and Sources

Comprehensive source list for further study: datasheets, academic papers, official documentation and educational resources.

You can read the manual from front to back, or jump directly to the relevant section. The table of contents and header on each page help you navigate.

PART A

User Manual

1. Introduction: What is MeshCore?

MeshCore - a combination of Mesh (network of interconnected devices) and Core - is a wireless communication system that operates completely independently of the internet, mobile networks and other central infrastructure.

It combines LoRa radio technology with advanced mesh networks to enable reliable communication over distances of tens of kilometers.

Important: MeshCore is a text-based communication system. It is specifically designed for sending text messages, GPS locations, telemetry data and control commands. Voice calls or video are not possible due to the limited bandwidth of LoRa radio.

1.1 What makes MeshCore unique?

Unlike traditional communication systems, MeshCore does not need a central server or access point. Each device in the network is simultaneously:

- A transmitter and receiver - for direct communication
- A router - that forwards messages for others
- An autonomous computer - that makes decisions without human intervention

This architecture makes the network robust: if one device fails, the network automatically finds an alternative route.

1.2 Who is MeshCore for?

- Families - who want to communicate with each other during travel or emergencies
- Clubs and associations - such as morse clubs or hiking groups
- Amateur radio operators - for digital experiments and emergency communication
- Outdoor enthusiasts - hikers, campers, adventurers

2. Origin and History

The technological evolution from radar technology to off-grid mesh communication

2.1 Chirp Spread Spectrum (1940s)

The roots of MeshCore lie in Chirp Spread Spectrum (CSS), a modulation technique developed in the 1940s for radar applications. CSS spreads a signal over a wide bandwidth by linearly increasing or decreasing the frequency (chirps). This makes the signal robust against multipath fading, interference and jamming.

Multipath fading A radio signal can reach the receiver via multiple paths: directly, but also via reflections from buildings, mountains or other objects. These signals arrive at slightly different times and can reinforce or cancel each other out (fading). CSS is robust against this because the frequency changes continuously - a reflection that arrives a bit later has a different frequency and therefore does not interfere destructively with the direct signal.

Interferentie Other radio sources on the same or nearby frequencies can interfere with your signal. Think of WiFi, other LoRa transmitters, or industrial equipment. CSS spreads the signal over a wide bandwidth and a long time. A short interfering transmitter only affects a small part of your chirp, and the FFT can still reconstruct the symbol from the remaining samples.

Jamming Intentional interference by a transmitter continuously emitting noise or a strong signal on your frequency. CSS is difficult to jam because:

- You must jam the entire 64 kHz (in NL) bandwidth simultaneously (not just one frequency)
- The processing gain ensures that your signal is still detectable as long as the jammer is not extremely strong
- Different spreading factors are orthogonal, so a jammer on SF7 does not disturb SF12

Orthogonal in this context means "independent" or "non-interfering
For a more detailed explanation see section 14.4

2.2 LoRa: from idea to chip (2009-2012)

In 2009, two French engineers, Nicolas Sornin and Olivier Seller, began developing a long-range, low-power modulation technique based on CSS. In 2010, François Sforza joined them and together they founded Cycleo in France. In May 2012, Cycleo was acquired by Semtech Corporation, which commercialized the technology under the brand name LoRa (Long Range).

2.3 Accessible hardware (2016-2018)

The real breakthrough for hobbyists came with the combination of Semtech's LoRa chips (SX1276/SX1262) and Espressif's ESP32 microcontroller. Manufacturers like Heltec and LILYGO brought integrated development boards to market for ~$20-30.

2.4 Meshtastic: the first mesh wave (2019-2020)

In 2019, American software enginee Kevin Hester (GitHub: geeksville) started the Meshtastic project. Meshtastic uses a flooding mesh protocol where each device forwards messages. In large networks, this led to congestion.

2.5 MeshCore: intelligent routing (2024-2025)

At the end of 2024, Australian developer Scott Powell (Ripple Radios) started a new protocol. In early 2025, he launched the MeshCore project together with Andy Kirby (UK) and Liam Cottle (NZ) with:

- Hybrid routing: first contact via flood, then learned routes for efficiency
- Role separation: Companion Radios, Repeaters and Room Servers as separate functions
- Scalability: up to 64 hops, state-aware network, AES-128 encryption
- Lightweight C++: no dynamic memory allocation, embedded-first design

2.6 Timeline

Year	Milestone	Meaning
1940s	CSS developed for radar	Basic modulation technique
2009	Sornin & Seller start LoRa development	CSS for data communication
2010	Cycleo opgericht (Frankrijk)	Commercialization begins
2012	Semtech acquires Cycleo	LoRa brand born
2015	LoRa Alliance opgericht	LoRaWAN standard
2016+	ESP32 + LoRa boards (Heltec e.a.)	Hardware accessible
2019	Kevin Hester start Meshtastic	First off-grid mesh
2024	Scott Powell starts MeshCore	Intelligent routing
2025	MeshCore publicly launched	Scalable architecture

3. Node Types (Firmware Profiles)

MeshCore distinguishes different node types based on their function in the network.
These types are determined by the firmware profile you flash.
These are described in the following paragraphs.

3.1 Companion Radio

This is the most commonly used type for end users. The node acts as a radio interface for a smartphone via Bluetooth (BLE), USB or WiFi. The MeshCore Companion App controls the node.

3.2 Repeater

A node configured as a repeater has one primary task: forwarding messages to extend the network range. Typically placed at strategic locations with good antenna position such as roof edges or tall buildings. Works fully autonomously without smartphone.

3.3 Room Server

A Room Server manages one or more Rooms and provides extensive store-and-forward functionality. Stores messages for offline recipients. Requires Ultra license for remote management.

3.4 Standalone Device

Some hardware such as the T-Deck Plus can function completely independently without an external smartphone. With built-in screen and keyboard you can type and read messages directly.

3.5 Telemetry Node

A node specifically configured for transmitting sensor data such as temperature, humidity, battery voltage. Expandable via GPIO, I²C or SPI interfaces.

3.6 Typical network

In practice, a network can consist of a combination of these node types. A typical family network could consist of:

- 2-4 Companion Radios for family members
- 1 Repeater at a high point for better coverage
- 1 Room Server at home for store-and-forward

4. Getting Started: First Installation

4.1 Requirements

- MeshCore-compatible hardware (e.g. Heltec V3, T-Deck Plus)
- Antenna for 868 MHz
- Android or iOS phone with MeshCore Companion App
- USB cable for initial configuration

4.2 Step 1: Flash firmware

Go to flasher.meshcore.co.uk in Chrome or Edge. Select your device, choose 'Companion Radio BLE' firmware, and click Flash.

4.3 Step 2: Connect with the app

Start the MeshCore Companion App, choose Scan, select your node and tap Connect.

4.4 Step 3: Configure

Set: frequency 868.300 MHz, bandwidth 125 kHz, spreading factor SF11, power 14 dBm (EU limit). Encryption is enabled by default.

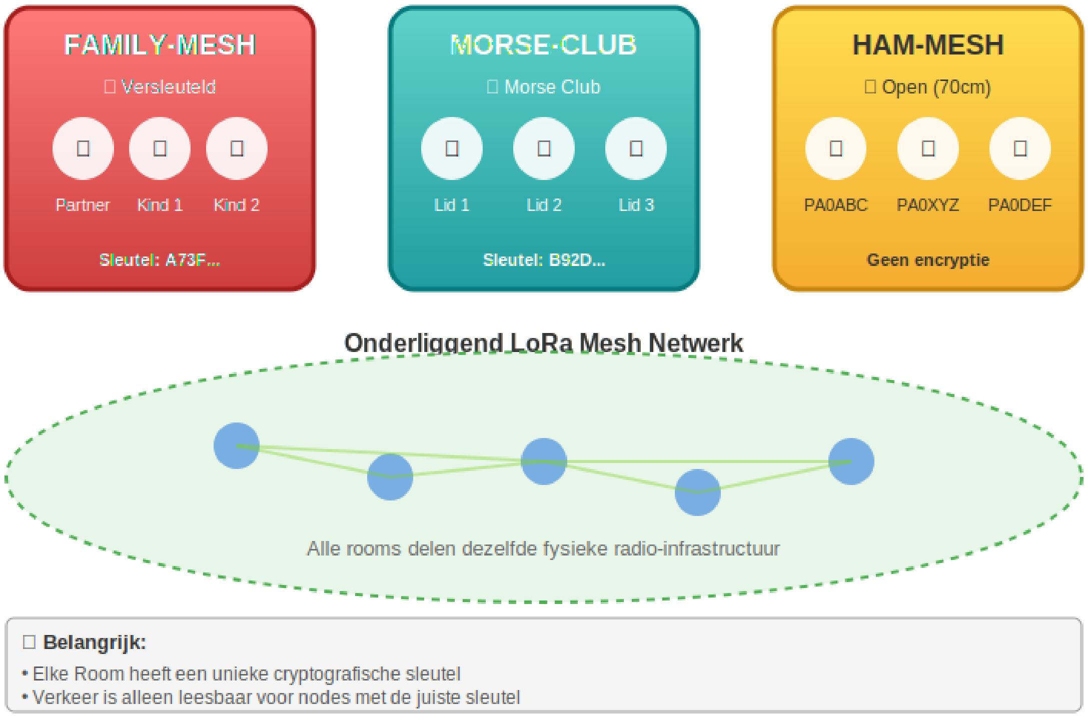
5. Group Communication: Channels and Room Servers

MeshCore has two ways to communicate in groups: Channels and Room Servers. These are fundamentally different concepts.

Channel - A shared cryptographic key (PSK for AES-256 encryption). Nodes with the same key can read each other's messages. There is no central server or member list. Messages are real-time: miss it, and it's gone.

Room Server - A physical node with server firmware that works as a BBS (Bulletin Board System). Users log in with a password. Messages are stored (store-and-forward) and you can come back later to retrieve missed messages (up to 32 messages).

MeshCore Rooms - Versleutelde Groepen



5.1 Channel types

Public Channel (#public)

The default channel that is automatically added with every MeshCore installation. This is the "marketplace" channel where everyone listens. Convenient for making first contact, but no privacy.

Hashtag Channel (#naam)

Community channels for specific topics or regions, such as #switzerland, #berlin, or #morsecode. The key is calculated from the name, so anyone who knows the name can listen in. Convenient for open communities.

Private Channel (own key)

A channel with a self-chosen, random key that you only share with the intended participants. This provides true privacy - only those who have the key can listen.

Type	Key	Privacy	Use
#public	Universally known	Geen	General communication
#hashtag	Calculated from name	None (derivable)	Community, region, topic
Private	Self-chosen, secret	Full	Familie, werk, gevoelig

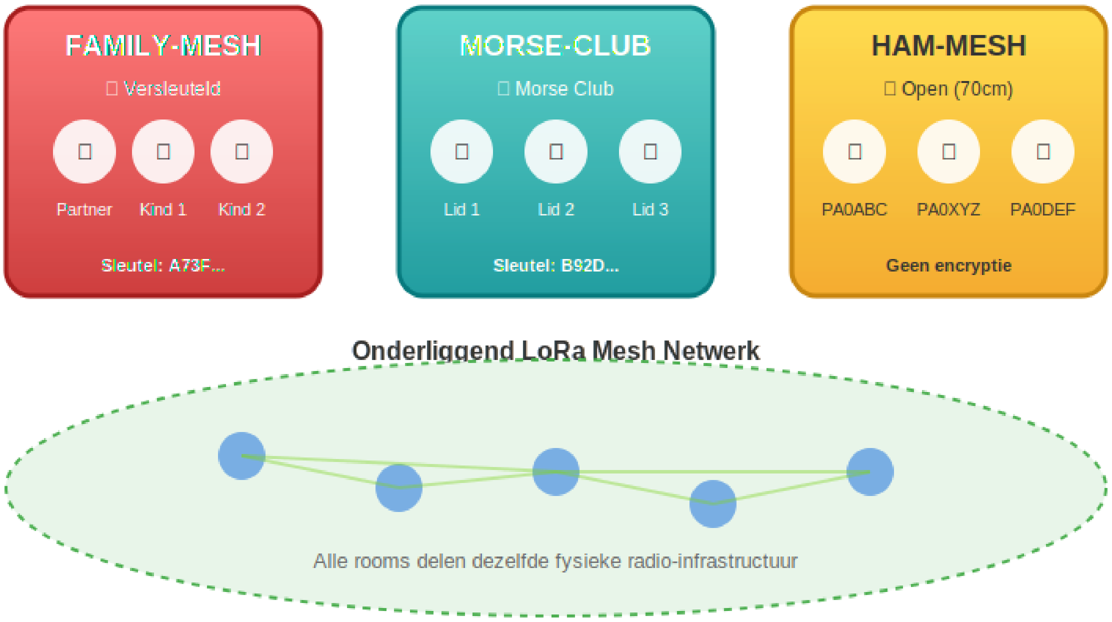
5.2 Room Servers

A Room Server is a node that is always online and offers additional features for reliable communication:

- Store-and-forward: messages are stored until the recipient comes online
- Member list: you see who is in the Room
- Management: moderators can add/remove members
- Persistence: up to 32 messages are stored

Functie	Channel	Room Server
Send messages	Only to online nodes	Also store for offline nodes
Store-and-forward	Limited (only in transit)	Full (long-term storage)
Bereikbaarheid	Depends on participants	24/7 via fixed node
Remote management	No	Yes (with Ultra license)

MeshCore Rooms - Versleutelde Groepen



Belangrijk:

- Elke Room heeft een unieke cryptografische sleutel
- Verkeer is alleen leesbaar voor nodes met de juiste sleutel

6. Direct Messages (DM)

In addition to group communication via Rooms, MeshCore also supports Direct Messages (DMs): private messages between two specific nodes. DMs are end-to-end encrypted and can only be read by the sender and receiver.

6.1 Difference between Rooms and DMs

Aspect	Room (Public)	Direct Message
Requirement	Member of the same Room	Receiver's public key
Encryption	Shared Room key	End-to-end (public key)
Key-uitwisseling	Via QR code or Room Server	Via advertisements (beacons)

6.2 How DMs work

- Node A broadcasts an advert with its public key
- Node B receives the advert and stores the public key
- Node B can now send an encrypted DM to Node A
- For two-way traffic, Node B must also broadcast an advertisement

Important: Adverts are NOT forwarded by repeaters. Both nodes must be able to "hear" each other directly for the key exchange.

6.3 Troubleshooting DMs

Problem	Solution
DM fails after 3 retries	Check if both nodes have received each other’s advert. Enable "Auto-add nodes to contacts".
Node does not appear in "Heard" list	Adverts are not forwarded. Ensure direct range between the nodes.
Contact is in list but DM fails	Contact may have been added via public message. Delete and have re-added via advert.
Public messages work, DM does not	Public goes via Room, DM requires public key. Send adverts and wait 15 seconds.

7. Privacy en Security

7.1 What is always visible?

Each node must transmit beacons for routing. Other nodes see that a node exists and is active.

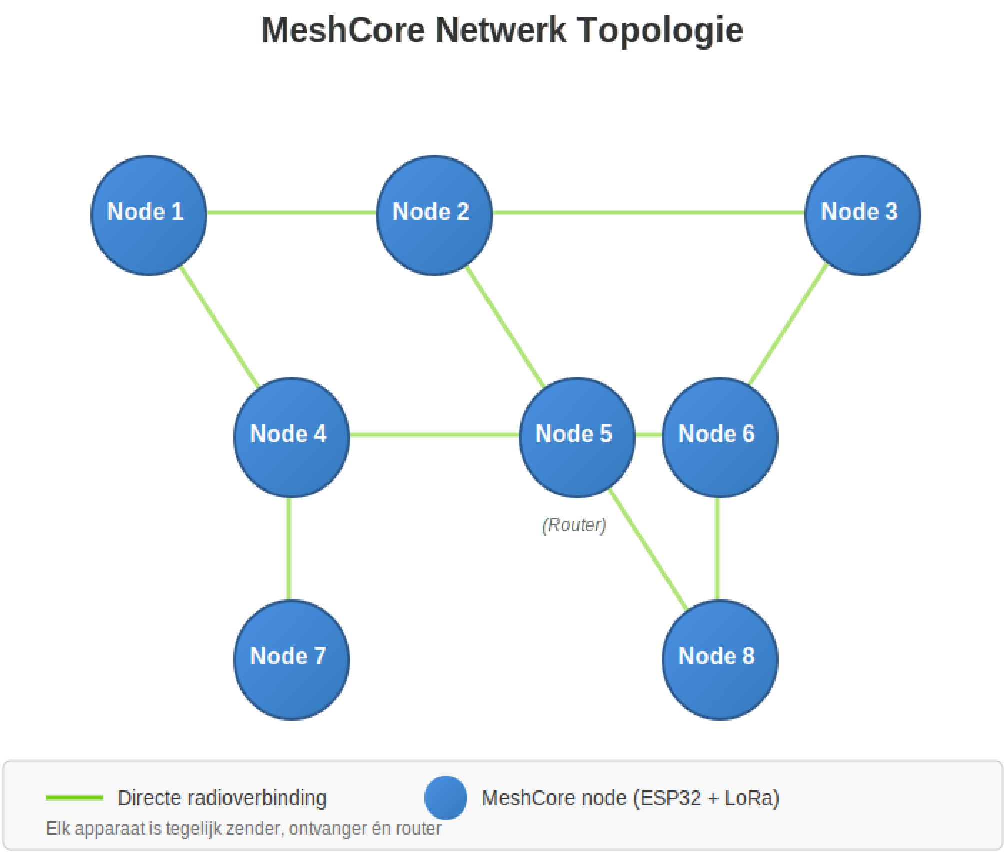
7.2 What is NEVER visible (ISM mode)?

- Who you communicate with
- Which Rooms you are in
- The content of your messages
- Even the existence of your Rooms

7.3 HAM vs ISM

Property	HAM-modus	ISM-modus
Frequency	70 cm (430-440 MHz)	868 MHz
License	Required	Not required
Encryption	Not allowed	Required
Identification	Callsign	Anonymous possible

8. Practical Applications



8.1 Familie-mesh

All family members have a MeshCore device in Room 'FAMILY-MESH'. Use: short messages, GPS tracking, appointments. Works without mobile network on vacations, festivals or in remote areas.

8.2 Morse-club

Club members coordinate CW activities, field days and spot reports via a shared Room. Members see each other on the map and can share real-time QSO information.

8.3 Amateur Radio Mesh

On the 70 cm band with callsigns. Digital QSOs, emergency communication, self-routing repeaters. Each node identifies itself with callsign according to amateur regulations.

8.4 Remote Station

Node on roof terrace with good antenna, node inside with BLE to phone. Control your radio infrastructure via your own mesh. No VPN or internet connection needed.

9. Hardware Overview

MeshCore supports a wide range of hardware devices. The data below is taken from various internet sources. In my limited knowledge built up in a short time, I can say that the Standalone T-Deck is not very suitable for family use in our case.
The combination of companion app with the SensCap T1000e was experienced as much more user-friendly.

9.1 Hardware Categories

- Companion Radios - require smartphone for control via Bluetooth/USB/WiFi
- Standalone Devices - complete communication devices with keyboard and screen
- Repeaters and Servers - nodes that extend the network

9.2 LilyGO T-Deck Plus (€70-80)

Best for: Standalone use without smartphone

Complete standalone oplossing met ESP32-S3, 2.8 inch LCD kleurenscherm, fysiek QWERTY toetsenbord, ingebouwde GPS, trackball navigatie en SMA antenneconnector.

Advantages: Fully independent, physical keyboard, robust housing, GPS built-in, external antenna possible.

Disadvantages: Trackball sometimes sensitive, reset button can be accidentally pressed.

9.3 Heltec WiFi LoRa 32 V3/V4 (€20-40)

Best for: Budget entry with smartphone

Cheapest way to get started with MeshCore. ESP32-S3 with SX1262 LoRa, small 0.96 inch OLED display. V4 has higher transmit power (28dBm).

Advantages: Very affordable, compact, widely supported, suitable as companion and repeater.

Disadvantages: Requires smartphone, small screen, no housing standard.

9.4 RAK WisBlock RAK4631 (€40-60)

Best for: Energy-efficient and modular solutions

Based on nRF52840 with extremely low power consumption - weeks to months on battery. Modular system with expansion modules for GPS, sensors, etc.

Advantages: Extremely low consumption, modular, perfect for solar, professional quality.

Disadvantages: More expensive, no WiFi (Bluetooth only), more complex setup.

9.5 Seeed Studio T1000-E (€30-40)

Best for: Compact companion radio with GPS

Extremely compact credit card format with built-in GPS. nRF52840 based for low power consumption.

Advantages: Very compact, built-in GPS, low consumption, robust.

Disadvantages: No display or buttons, requires smartphone for all interaction.

9.6 Hardware Comparison Table

Specificatie	T-Deck Plus	Heltec V3/V4	RAK4631	T1000-E
Price	€70-80	€20-40	€40-60	€30-40
Standalone	Yes	No	No	No
Display	Yes	Yes	No	No
GPS ingebouwd	Yes	No	Optional	Yes
Battery life	2-3 days	1-2 days	Weeks	Weeks
Externe antenne	Ja (SMA)	Ja (IPEX)	Ja (U.FL)	No
Best voor	Standalone	Budget	Solar/IoT	Compact/GPS

Note: Always check that the device has the correct frequency for Europe (868 MHz, not 915 MHz).

9.7 Offline Maps on T-Deck

The T-Deck Plus can display offline maps via a microSD card. This is useful for navigation without internet connection. On the internet I found the following information.

9.7.1 Requirements

- microSD card (max 32 GB, formatted as FAT32)
- Computer with Python 3 installed
- Internetverbinding voor het downloaden van kaarten

9.7.2 The different methods

Method 1: Map Tiles Downloader (recommended)
The simplest method is the Map Tiles Downloader from OM7TEK:

Windows: Open PowerShell and run:

- pip install pipx
- python -m pipx ensurepath
- pipx install mt-downloader
- mt-downloader

macOS: Open Terminal and run:

- brew install pipx
- pipx ensurepath
- pipx install mt-downloader
- mt-downloader

The program displays a text interface where you can select continents, countries and regions. The tiles are downloaded to a 'tiles' folder.

Method 2: tdeck-maps script
Alternatively you can use the tdeck-maps Python script:

- Installeer dependencies: pip install pillow requests
- Clone repository: git clone https://github.com/JustDr00py/tdeck-maps.git
- Download tiles for a city:

python3 meshtastic_tiles.py --city "Amsterdam, Netherlands" --min-zoom 8 --max-zoom 12

Method 3: Buy ready-made maps
Pre-downloaded map packages are available at https://buymeacoffee.com/ripplebiz for Europe and other regions. This supports the developer and saves time.

Method 4: Download ready-made maps with higher zoom level.
For this method you need a lot of patience.
Via the URL below you can download all of Europe with a very high zoom level, note that copying to SD card can take more than a day.

https://www.reddit.com/r/meshtastic/comments/1j1chem/meshtastic_26_map_tiles_with_higher_zoom_levels/.

9.7.3 Installation on SD card

- Formatteer de microSD-kaart als FAT32 (niet exFAT)
- Copy the ‘tiles’ folder to the root of the SD card
- The folder structure must be: /tiles/[zoomlevel]/[x]/[y].png
- Safely eject the SD card
- Insert the SD card into the T-Deck (on the left side)
- Restart the T-Deck

9.7.4 Using the map

- Navigate to the Maps screen via the map icon
- Use the trackball or touchscreen to pan
- Zoom in/out with the 'W' and 'S' keys or the buttons on the screen
- Tap the cross to return to your current location

Note: The SD card must be inserted before powering on the T-Deck.

9.7.5 Zoom levels

Zoom level	Detail	Use	Storage per region
8-10	Lowe	Large region overview	Small (~50 MB)
11-12	Avarage	Cities and roads	Avarage (~200 MB)
13-14	High	Street level	Big (~1 GB+)

Tip: Start with zoom level 8-12 for a good balance between detail and storage space. Deeper zoom (13+) requires an Ultra license.

10. Useful Links and Resources

10.1 Official Websites

- https://meshcore.co.uk/ - Officiële website
- https://github.com/meshcore-dev/MeshCore - GitHub broncode

10.2 Web Tools

- https://flasher.meshcore.co.uk/ - Web Flasher for firmware
- https://analyzer.letsmesh.net/ - Packet Analyzer
- https://meshcore.nz/ - Web Companion Client
- https://map.meshcore.dev/ - Officiële MeshCore kaart

10.3 Apps

- https://meshcore.co.uk/apps.html - Overzicht alle apps
- MeshCore Companion App - Android en iOS

10.4 Community

- https://discord.gg/ZVH2ujy9ex - MeshCore Discord server
- https://forum.meshcore-net.nl/ - Nederlandstalig forum

11. Open Source and Paid Licenses

MeshCore is an open-source project. The core software is freely available under the MIT license, but some advanced features require a paid license.

11.1 What is open-source?

The open-source core of MeshCore includes: mesh routing, end-to-end encryption, room systems, basic mesh functionality. Everyone can view the source code, modify it and create their own builds.

11.2 MeshCore Ultra Software License (£8)

The MeshCore store sells unlock licenses for extra features on certain devices. The Ultra license costs £8 (one-time) and unlocks:

- Higher zoom levels on maps
- Remote Repeater profile (node as fixed repeater)
- Room server admin (manage rooms remotely)
- Telemetry node (send/receive sensor data)
- Store-and-forward (store messages and deliver later)

11.3 What always remains free?

- Full mesh routing
- Encrypted chat
- Rooms and group communication
- Basic GPS
- LoRa networks
- Custom firmware builds

PART B

Technology Explained

12. From Text to Chirp

How LoRa encodes data - a step-by-step explanation from bits to radio signal

12.1 You want to send "Test"

Let's start with something concrete: you want to send the word "Test" via LoRa. How does that become a radio signal that can travel kilometers?

Letters become bits

Each letter has an ASCII code, and that code is a number we can write as bits:

Letter	ASCII	Binair
T	84	01010100
e	101	01100101
s	115	01110011
t	116	01110100

Samen is "Test" dus 32 bits.

12.2 Bits become symbols

Now the Spreading Factor (SF) comes into play. At SF12 we group bits in groups of 12.

Why 12 bits?

SF12 means: each symbol carries 12 bits of information. Those 12 bits together form a number from 0 to 4095 (because 2^{12} = 4096 possible values).

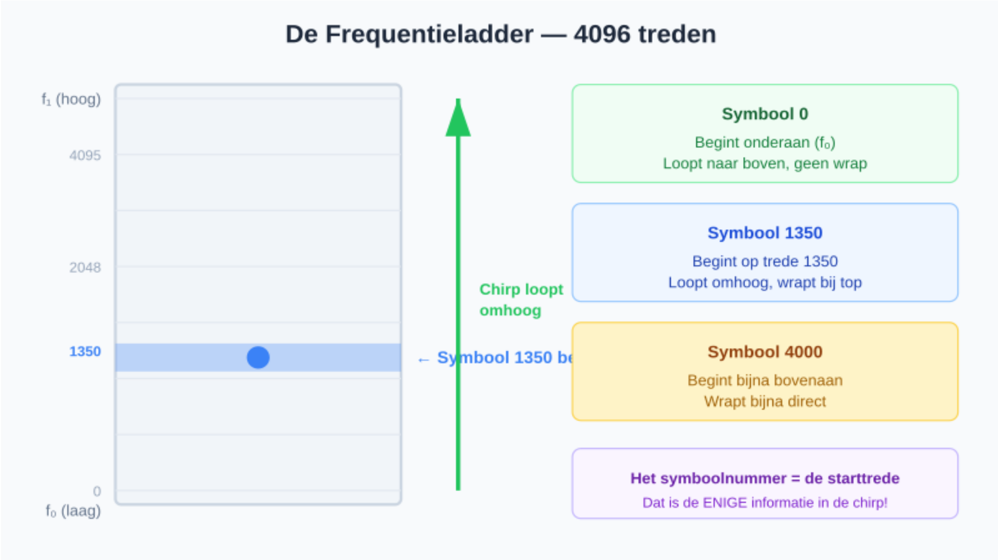
We take our 32 bits and divide them into groups of 12. The word "Test" becomes three symbols: 1350, 1395 and 1860.

12.3 Symbols become chirps

Now the crucial step: how does a number (for example 1350) become a radio signal?

The frequency ladder

Imagine the 125 kHz bandwidth as a ladder with 4096 rungs. Each symbol number corresponds to a starting position on that ladder.



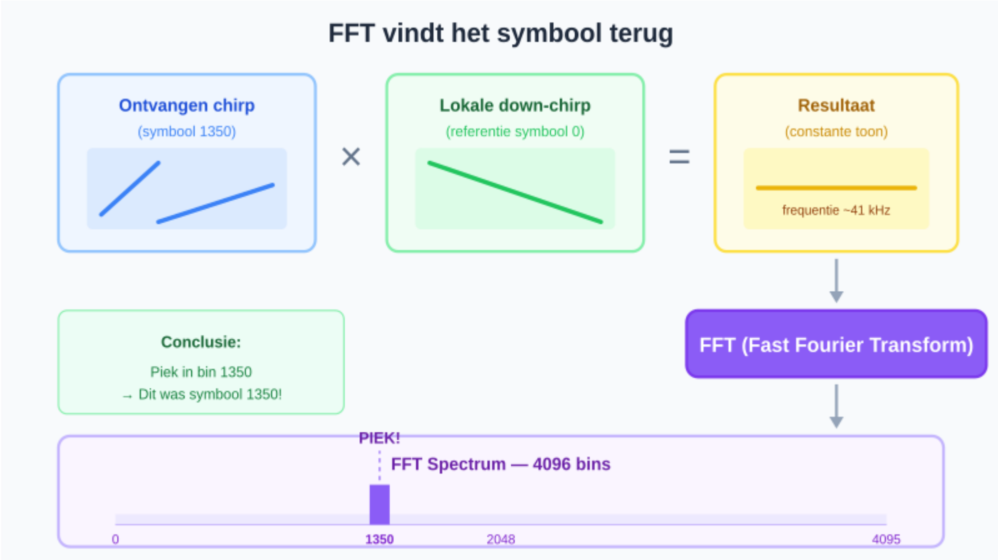
The chirp walks up the ladder

A chirp starts at its starting position and then walks through all steps, upward. At the top it wraps to the bottom and continues until it returns to its starting point.

12.4 How does the receiver know which symbol it was?

The receiver performs a clever mathematical trick: it multiplies the received chirp with a locally generated down-chirp (descending frequency).

Rising frequency × falling frequency = constant tone. The pitch of that tone depends on where the original chirp started.



The FFT (Fast Fourier Transform) analyzes the tone and produces a spectrum with 4096 bins. The bin where the energy is concentrated = the symbol number.

Error tolarance: Processing Gain

The power of LoRa lies in redundancy. At SF12, 12 bits are spread over 4096 frequency steps. This is 341× more bandwidth than strictly necessary. This "processing gain" (~36 dB) enables detection below the noise floor.

12.5 Summary

The complete chain from text to radio signal:

Written;

- Text → ASCII → Bits
- Bits → Groups of SF bits → Symbols (0-4095)
- Symbol → Start position on frequency ladder
- Chirp traverses all steps, wrapping at the top
- Receiver multiplies with down-chirp → constant tone
- FFT finds the peak → symbol number recovered

13. Dechirp Analysis

Why the simple table fails and how FFT peak detection works

13.1 Introduction

This analysis is part of an attempt to better understand why LoRa works so well. The goal is to make clear in a simple way what the dechirp process does and how the FFT extracts the symbol from it.

To keep things clear, a highly simplified example is used: a "bandwidth" of 10 frequency steps (0-9), where each sample takes 1 second. In practice, the times are much shorter (milliseconds) and the number of samples per symbol much larger (128-4096), but the principle remains the same.

13.2 The problem with the simple table

In my first attempt to demonstrate LoRa dechirp with a simple sum table, things go wrong as soon as a frequency wrap occurs. First the case without shift.

Symbol 0 (no shift) - Works

Here the TX (up-chirp) and RX (down-chirp) run neatly opposite each other, and the sum remains constant.

TX (up)	RX (down)	Sum
0	10	10
1	9	10
2	8	10
3	7	10
4	6	10
5	5	10
6	4	10
7	3	10
8	2	10
9	1	10

✓ Constant sum = 10. Here the idea "constant sum = good" works.

Symbol 3 (shifted) - Fails after wrap

Then the same table for the shifted version (symbol 3):

TX (start=3)	RX (down)	Sum - 10	Status
3	10	3	✓
4	9	3	✓
5	8	3	✓
6	7	3	✓
7	6	3	✓
8	5	3	✓
9	4	3	✓
0 (wrap)	3	-7	✗ FOUT
1	2	-7	✗ FOUT
2	1	-7	✗ FOUT

After the wrap, the calculation no longer works in this simple approach.

13.3 Why wrapping is necessary

The frequency band is physically limited. In this example we use 10 steps (0-9), in practice for example 64 kHz bandwidth with a fixed channel width.

The frequency must not go outside the band because:

- It would fall outside the allocated spectrum (illegal)
- It would interfere with other services
- The receiver would not be able to follow the signal

Therefore, time/position runs neatly (0-9), but the frequency jumps back to 0 when it would go past 9: that is the wrap.

Important: the wrap is a practical limitation of the frequency band, not a real "reset" of the underlying phase development of the signal.

13.4 The correct interpretation

The solution is to look at the difference instead of the sum, and to let the TX frequency continue mathematically after the wrap. Then the difference remains constant.

Position	TX freq	TX (math.)	RX ref	Difference
0	3	3	0	3
1	4	4	1	3
2	5	5	2	3
3	6	6	3	3
4	7	7	4	3
5	8	8	5	3
6	9	9	6	3
7	0 (wrap)	10	7	3
8	1	11	8	3
9	2	12	9	3

The difference remains constant = 3, regardless of the wrap!

By "counting through" the TX frequency (10, 11, 12, ...) you see that the difference with the RX reference remains 3 everywhere, even after the wrap. That constant difference is exactly what the FFT finds as a peak.

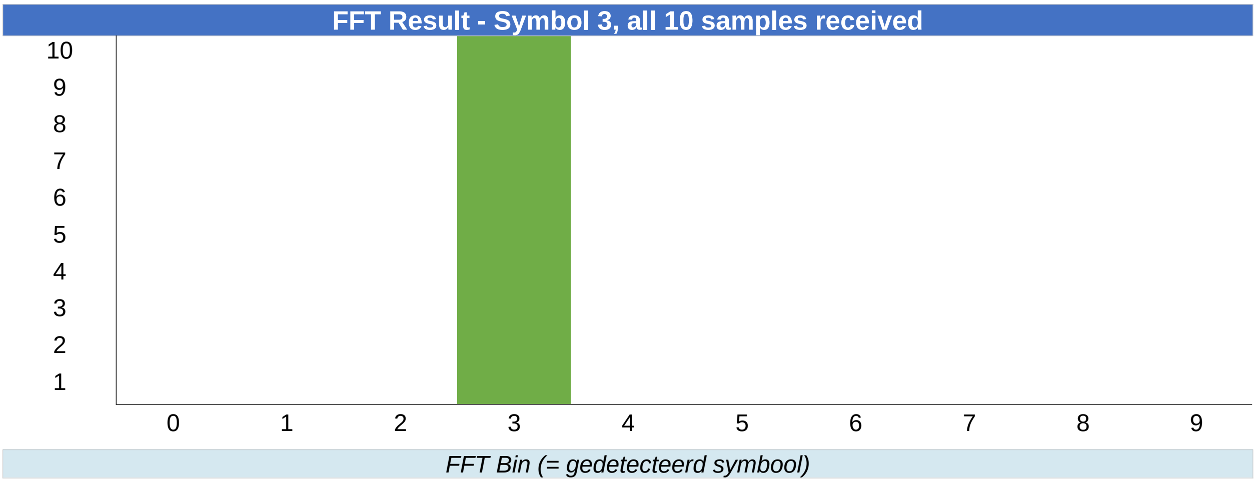
Note: this is a simplified representation intended to make the working principle of dechirp and FFT peak detection intuitive; the actual LoRa implementation uses a more complex, but mathematically equivalent description.

13.5 FFT peak detection - Graphical representation

The FFT essentially counts how often each possible difference occurs; each difference ends up in its own bin. For symbol 3, all 10 samples yield difference 3, so all energy accumulates in bin 3.

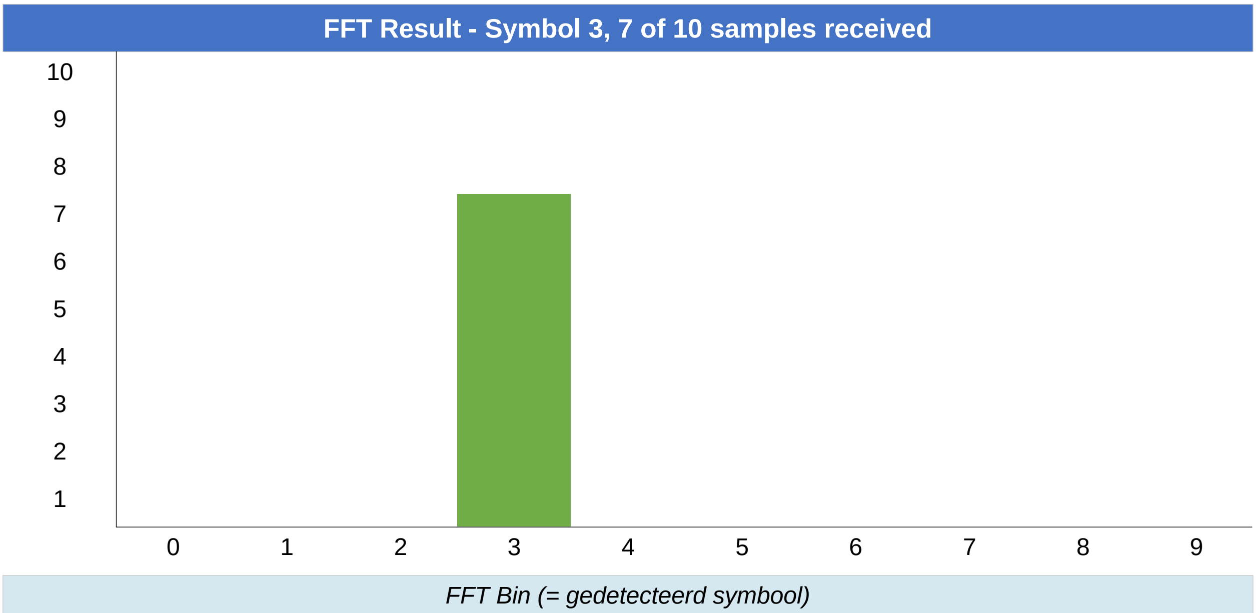
A bin is literally a "bucket" in which the energy for that specific frequency difference is accumulated.

Ideal signal (no loss)



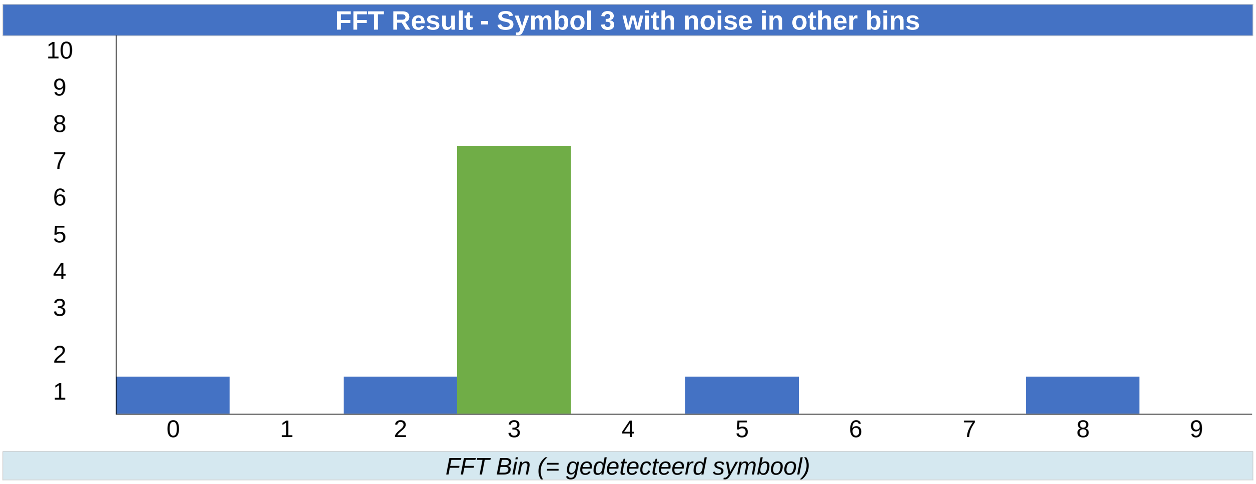
- All 10 samples contribute to bin 3
- Bin 3 therefore gets a peak height of 10, the other bins remain low

With 30% sample loss



Even with 3 missed samples, the peak at bin 3 remains dominant. The symbol is correctly detected.

With noise/interference



Noise and errors land in random bins. They are spread out and cannot exceed the signal peak.

13.6 The core principle

SIGNAAL	RUIS
All samples → same bin	Fouten → willekeurige bins
= CONCENTRATED ENERGY	= DISPERSED ENERGY

Therefore, incidental contamination can never exceed the signal peak.

13.7 How many samples can you miss?

The error tolerance depends on the Spreading Factor:

Spreading Factor	Samples per symbol	~30% loss tolerable
SF7	128	~38 samples
SF10	1024	~307 samples
SF12	4096	~1229 samples

Additionally, the Coding Rate (CR) adds extra error correction:

Coding Rate	Overhead	Fouttolerantie
CR 4/5	25%	Basic
CR 4/6	50%	Moderate
CR 4/7	75%	Good
CR 4/8	100%	Maximum

13.8 Processing gain - The power of LoRa

The "magic" of LoRa lies in processing gain: by spreading the signal over many samples, you can detect signals below the noise floor.

In our example with 10 samples:

- Signal energy concentrates in 1 bin
- Noise distributes over 10 bins
- Processing gain ≈ 10× (10 dB)

At SF12 with 4096 samples: processing gain ≈ 4096× (36 dB)!

This explains why LoRa can establish connections that would be impossible with conventional radio.

13.9 Conclusion

The original table approach failed because:

- A sum was used instead of a difference
- The wrap was seen as a real reset, instead of a practical limitation of the frequency band
- Instantaneous frequencies were compared, while LoRa works with phase differences that become visible as frequency difference via the FFT

The FFT sees the constant difference as a clear peak:

- All correct samples accumulate in the same bin (signal)
- Noise and errors are spread across other bins
- The peak of the signal therefore remains dominant, even with significant sample loss

In MeshCore, the FFT is used as a decision maker: the bin with the highest energy is the decoded symbol. This keeps the implementation relatively simple, scalable and robust against errors.

14. LoRa Modulation

From OSI model to radio signal - CSS, Spreading Factors and Demodulation

14.1 LoRa in the network layers

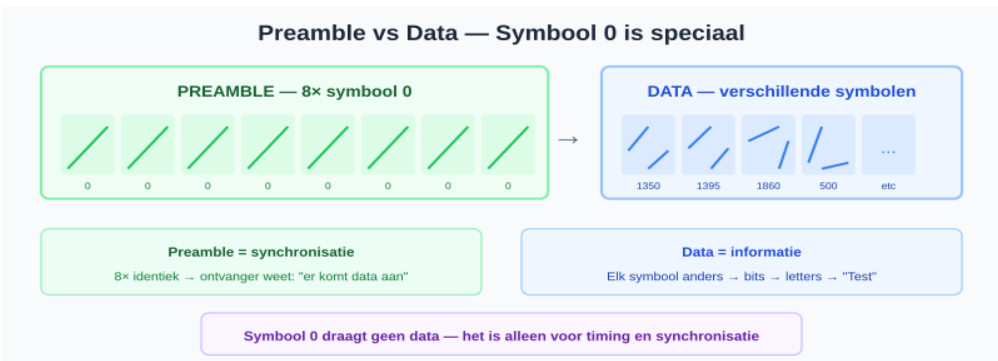
LoRa and MeshCore can be fitted into the well-known OSI model:

OSI layer	LoRa/MeshCore equivalent
7. Applicatie	MeshCore Companion App, chat, GPS
6. Presentation	— (not present)
5. Sessie	— (no persistent sessions)
4. Transport	MeshCore ACKs, retries, fragmentation
3. Network	MeshCore routing, hops, Room addressing
2. Datalink	LoRa packet: preamble, sync word, header, CRC
1. Physical	LoRa PHY: chirps, SF, BW, the radio itself

14.2 Synchronization without handshake

There is no handshaking with LoRa. The transmitter transmits, and whoever listens, listens. Synchronization happens via the preamble - identical chirps at the beginning of each packet.

Symbol 0 is the reference - a chirp that starts at the lowest frequency and runs neatly upward. The preamble consists of 8× symbol 0, followed by the data.



What the receiver learns from the preamble

Informatie	Hoe
Timing	Chirps arrive at regular intervals
Frequentie-offset	Preamble chirps are symbol 0, deviation = drift
SF confirmation	Chirp length matches expected SF
Netwerk-ID	Sync word must match

14.3 Demodulation: down-chirp mixing

The receiver multiplies the received signal with a locally generated down-chirp (descending frequency). This process is called dechirping.

Demodulation steps

1. Receive up-chirp with unknown start frequency
2. Multiply with local down-chirp
3. Result: constant tone (frequency depends on start position)
4. Apply FFT to the dechirped signal
5. Find the peak in the spectrum
6. Peak position = symbol value

14.4 SF Orthogonality

Different spreading factors are (almost) orthogonal: an SF10 receiver ignores SF12 signals, and vice versa. This is because the chirp duration differs per SF:

SF	Symbols	Chirp duration (BW=125kHz)	Bit/symbols
SF7	128	1.02 ms	7
SF8	256	2.05 ms	8
SF9	512	4.10 ms	9
SF10	1024	8.19 ms	10
SF11	2048	16.38 ms	11
SF12	4096	32.77 ms	12

14.5 Chirp Spread Spectrum (CSS)

LoRa uses Chirp Spread Spectrum (CSS): a modulation technique where the frequency runs linearly through the bandwidth. This makes the signal very robust.

Why CSS is robust

- Multipath: reflections are shifted in time, not in frequency evolution
- Interference: other signals do not correlate with the chirp
- Doppler: frequency shift shifts the entire chirp, not the shape

14.6 The Encoding Pipeline

LoRa has a multiple encoding pipeline that converts data to chirps:

1. Data → Whitening (pseudo-random XOR to prevent DC bias)
2. Whitened data → Hamming FEC (add error correction)
3. FEC codewords → Interleaving (spread bits over symbols)
4. Interleaved data → Gray coding (minimize bit errors)
5. Gray-coded symbols → Chirp modulation
6. Preamble + Sync + Header + Payload + CRC → Packet
7. Packet → Radiosignaal

14.7 Bits, Symbols and Chips

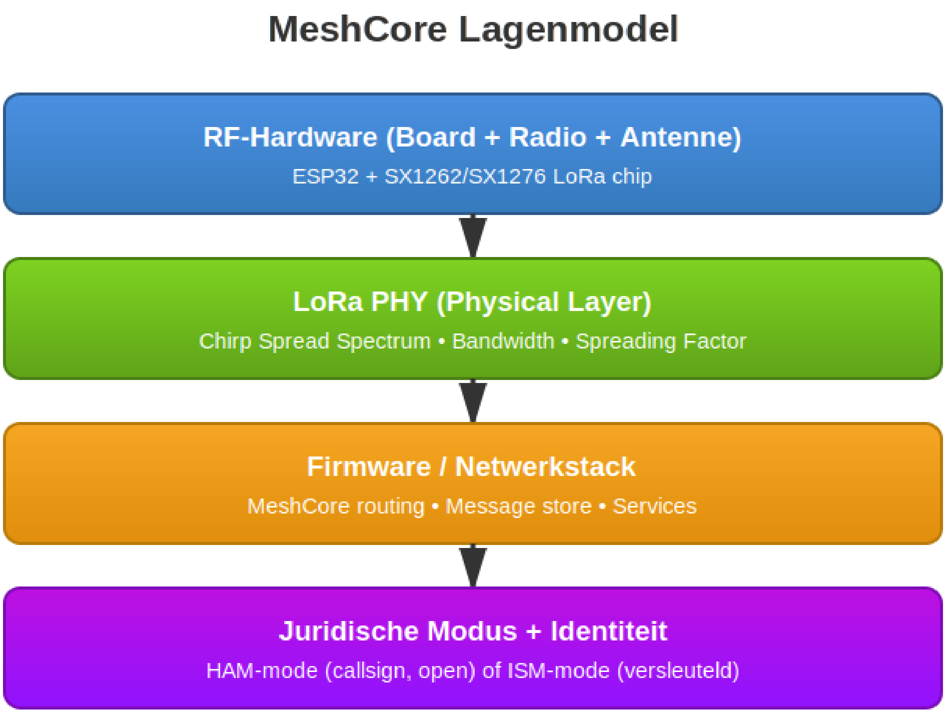
In spread spectrum terminology:

- Bit: the smallest unit of information
- Symbol: a group of SF bits (e.g. 12 bits at SF12)
- Chip: one frequency step in the chirp (2^SF chips per symbol)

At SF12, 12 bits are spread over 4096 chips. This provides a processing gain of ~21 dB, allowing signals below the noise floor to still be decodable.

15. The Layer Model of MeshCore

MeshCore is built from four strictly separated layers, each with a specific function:



15.1 Layer 1: RF hardware

Microcontroller (ESP32/nRF52), LoRa chip (SX1262), antenna, power supply.

15.2 Layer 2: LoRa PHY

Modem layer: bits → radio signals via CSS. BW, SF and CR determine range vs speed.

15.3 Layer 3: Firmware and Network Stack

MeshCore firmware handles routing, message handling and network management. Each node has its own identity (4-byte Node-ID) and can learn routes to other nodes. Intelligent routing instead of pure flooding.

15.4 Layer 4: Legal mode

Determines whether the device runs in HAM mode (70 cm band, no encryption, callsign required) or ISM mode (868 MHz, encryption, anonymous possible).

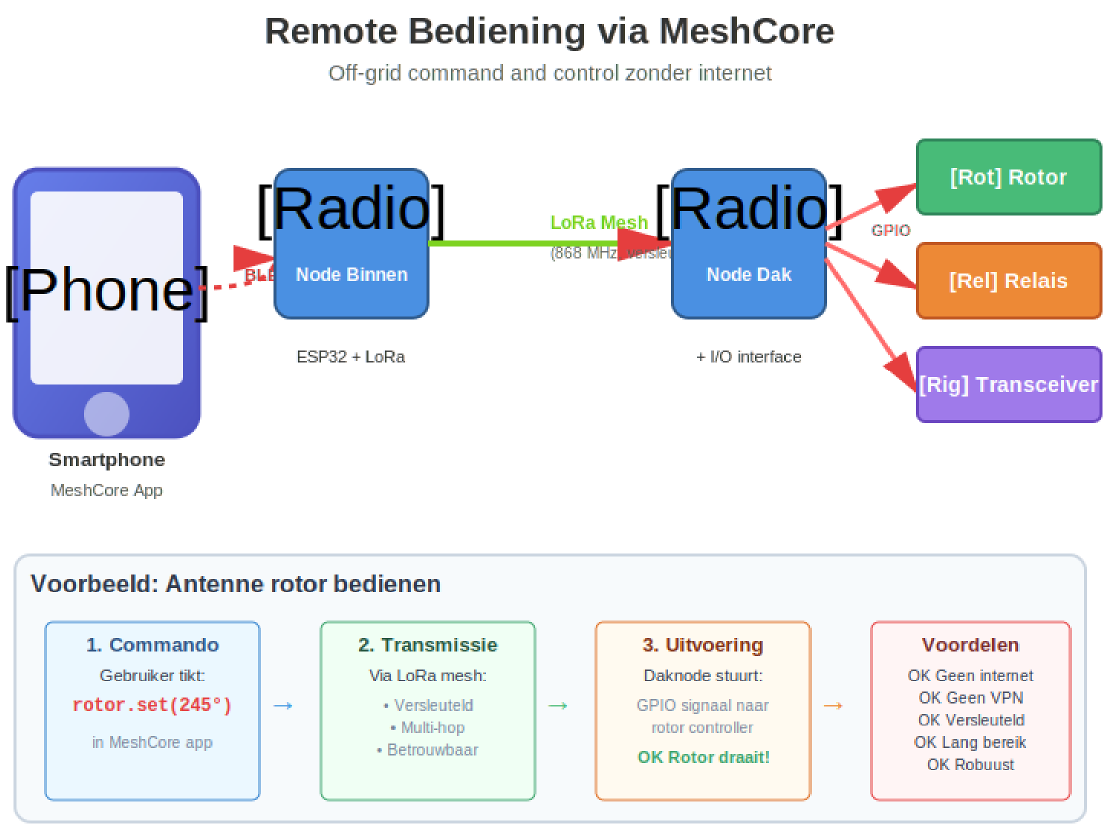
15.5 Link Budget - Why LoRa reaches so far

Link budget is the total signal loss a connection can endure and still be decodable. LoRa achieves 150+ dB link budget, enabling:

- Long distances possible with low power
- Connections below the noise floor can still be decodable
- Meshes with few nodes can already scale regionally (via hops)

16. Advanced Features: Remote Control

MeshCore is not just a chat system. It can be used as a fully off-grid, encrypted command-and-control network.



16.1 Technically speaking

Elke MeshCore-node is tegelijkertijd:

- A LoRa radio - for wireless communication
- A router - for mesh network functions
- A computer (ESP32) - with processing power
- With I/O interfaces - UART, GPIO, I2C, SPI, USB

This means you can literally connect devices:

- Relays - for switchable power
- Arduinos and Raspberry Pis - for more complex tasks
- Transceivers - via CAT interface
- Sensors - temperature, humidity, air pressure
- Motors and rotors - for antenna direction
- Power switches - for remote on/off

16.2 How does it work?

MeshCore has a service layer on top of the mesh. A node can offer services such as gpio.toggle, uart.send or i2c.read. Other nodes can call these services via encrypted messages.

16.3 Concrete example: Roof terrace

You place a node at your antennas on the roof. That node is in Room ‘REMOTE-MESH’ and has GPIO connections to your antenna rotator and transceiver. Inside you have a node with BLE connected to your phone.

Send command: Phone → BLE → indoor node → mesh → roof node → GPIO → rotor turns. No wifi needed, no internet provider, no cloud.

16.4 Why is this so powerful?

- Cannot be jammed with a single transmitter
- No central point that can fail
- No cloud needed
- Fully encrypted
- Tens of kilometers range

17. MeshCore Packet Structure

17.1 General Packet Structure

This document describes two layers working together:

LoRa Fysieke Laag	Radio hardware (SX1262/SX1276)	Preamble, Sync Word, MIC, CRC
MeshCore Protocol	MeshCore firmware (ESP32/nRF52)	Header + Payload (alle andere velden)

The LoRa chip automatically adds Preamble/Sync Word/CRC. MeshCore software only processes Header and Payload.

Each MeshCore packet follows a fixed structure consisting of the following parts:

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Synchronization for radio receiver
Sync Word	2	`12 34`	Network identifier (configurable)
Header	12-16	`(zie details)`	Routing and type information
Payload	0-200	`(variabel)`	Message content (type-dependent)
MIC	4	`XX XX XX XX`	Message Integrity Code (auth)
CRC	2	`XX XX`	Errordetection(CRC-16)

Total maximum packet size: 232 bytes

17.2 Header structure

The header contains essential routing and type information for each packet:

Field	Bytes	Hex Example	Description
Version	1	`01`	Protocol version (0x01)
Flags	1	`00-FF`	Packet flags (encrypt, ack, etc.)
Type	1	`01-06`	Message type identifier
Hop Count	1	`00-0F`	Number of hops (max 15)
Packet ID	4	`4F 3A 2B 1C`	Unique packet identifier
Source ID	4	`DE AD BE EF`	Sender Node-ID
Dest ID	4	`CA FE BA BE`	Receiver Node-ID (or FF FF FF FF)

Flags byte breakdown:

```
Bit 0: Encrypted (1=yes)
Bit 1: ACK Required (1=yes)
Bit 2: Is ACK (1=this is an ACK)
Bit 3: Is Broadcast (1=broadcast)
Bit 4-7: Reserved
```

Type values:

```
0x01 = TEXT_MESSAGE (text/chat)
0x02 = POSITION (GPS location)
0x03 = TELEMETRY (sensor data)
0x04 = NODEINFO (node identification)
0x05 = ROUTING (route discovery)
0x06 = ACK (acknowledgment)
0x07 = BEACON (periodic announcement)
0x10 = COMMAND (remote control)
```

17.3 Message types

Code	Type	Encryption	Routing
0x01	TEXT_MESSAGE	E2E / Room	Unicast / Room
0x02	POSITION	Optional	Broadcast
0x03	TELEMETRY	Optional	Broadcast
0x04	NODEINFO	No	Broadcast
0x05	ROUTING	No	Broadcast
0x06	ACK	E2E	Unicast
0x07	BEACON	No	Broadcast
0x10	COMMAND	E2E (verplicht)	Unicast

17.4. The text messages explained

17.4.1 TEXT_MESSAGE(0x01)

Text messages for user communication. Can be private (encrypted) or public.

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Radio synchronization
Sync Word	2	`12 34`	MeshCore network ID
Version	1	`01`	Protocol v1
Flags	1	`03`	Encrypted + ACK required
Type	1	`01`	TEXT_MESSAGE
Hop Count	1	`00`	First hop
Packet ID	4	`4F 3A 2B 1C`	Unique ID
Source ID	4	`DE AD BE EF`	Sender node
Dest ID	4	`CA FE BA BE`	Receiver node
Room ID	4	`00 00 00 00`	Room (0=DM)
Timestamp	4	`65 A1 B2 C3`	Unix timestamp
Text Length	1	`10`	Payload length (16 bytes)
Text Data	var	`44 69 74 20 69 73 ...`	"This is a test!"
MIC	4	`AB CD EF 12`	Authentication
CRC	2	`34 56`	Checksum

Complete Hex Dump:

```
AA AA AA AA AA AA AA AA 12 34 01 03 01 00 4F 3A 2B 1C DE AD BE EF CA FE BA BE 00 00
00 00 65 A1 B2 C3 10 44 69 74 20 69 73 20 65 65 6E 20 74 65 73 74 21 AB CD EF 12 34
56
```

17.4.2 POSITION(0x02)

GPS coordinates of a node. Automatically broadcast at configured interval.

Field	Bytes	Hex Example	Description
Preamble	8	AA AA AA AA AA AA AA AA	Radio synchronization
Sync Word	2	12 34	MeshCore network ID
Version	1	01	Protocol v1
Flags	1	08	Broadcast, no ACK
Type	1	02	POSITION
Hop Count	1	00	First hop
Packet ID	4	12 34 56 78	Unique ID
Source ID	4	DE AD BE EF	Sender node
Dest ID	4	FF FF FF FF	Broadcast
Latitude	4	42 50 24 DD	52.0705° (float)
Longitude	4	40 89 9A 44	4.3007° (float)
Altitude	2	00 05	5 meter
Precision	1	0A	GPS HDOP ×10
Timestamp	4	65 A1 B2 C3	Unix timestamp
Speed	2	00 00	Speed km/h
Heading	2	00 B4	Direction 180°
MIC	4	78 9A BC DE	Authentication
CRC	2	F0 12	Checksum

Complete Hex Dump:

```
AA AA AA AA AA AA AA AA 12 34 01 08 02 00 12 34 56 78 DE AD BE EF FF FF FF FF 42 50
24 DD 40 89 9A 44 00 05 0A 65 A1 B2 C3 00 00 00 B4 78 9A BC DE F0 12
```

17.4.3 TEXT_MESSAGE(0x03)

Sensor data such as battery percentage, temperature and voltage.

Field	Bytes	Hex Example	Description
Preamble	8	AA AA AA AA AA AA AA AA	Radio synchronization
Sync Word	2	12 34	MeshCore network ID
Version	1	01	Protocol v1
Flags	1	08	Broadcast
Type	1	03	TELEMETRY
Hop Count	1	00	First hop
Packet ID	4	AB CD EF 01	Unique ID
Source ID	4	DE AD BE EF	Sender node
Dest ID	4	FF FF FF FF	Broadcast
Battery %	1	57	87% batterij
Voltage	2	0E D8	3.80V (×1000)
Temperature	2	00 DC	22.0°C (×10)
Humidity	1	41	65% RH
Pressure	2	27 D6	1018.2 hPa (×10)
Uptime	4	00 01 51 80	86400 sec (1 dag)
TX Count	2	00 C8	200 transmissions
RX Count	2	01 2C	300 received
MIC	4	11 22 33 44	Authentication
CRC	2	55 66	Checksum

Complete Hex Dump:

```
AA AA AA AA AA AA AA AA 12 34 01 08 03 00 AB CD EF 01 DE AD BE EF FF FF FF FF 57 0E
D8 00 DC 41 27 D6 00 01 51 80 00 C8 01 2C 11 22 33 44 55 66
```

17.4.4 NODEINFO(0x04)

Information about the node: name, hardware model and firmware version.

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Radio synchronization
Sync Word	2	`12 34`	MeshCore network ID
Version	1	`01`	Protocol v1
Flags	1	`08`	Broadcast
Type	1	`04`	NODEINFO
Hop Count	1	`00`	First hop
Packet ID	4	`FE DC BA 98`	Unique ID
Source ID	4	`DE AD BE EF`	Sender node
Dest ID	4	`FF FF FF FF`	Broadcast
Name Len	1	`08`	Name length (8)
Node Name	var	`4E 6F 64 65 2D 41 42 43`	"Node-ABC"
Model ID	2	`00 03`	Heltec V3 (model 3)
FW Major	1	`01`	Firmware v1
FW Minor	1	`02`	.2
FW Patch	1	`03`	.3 → v1.2.3
Capabilities	2	`00 1F`	Feature flags
Public Key	32	`04 AB CD ... (32 bytes)`	Ed25519 pubkey
MIC	4	`AA BB CC DD`	Authentication
CRC	2	`EE FF`	Checksum

Model ID values:

```
0x0001 = T-Deck Plus
0x0002 = Heltec V2
0x0003 = Heltec V3
0x0004 = RAK4631
0x0005 = T1000-E
0x0006 = T-Beam
```

17.4.5 ROUTING(0x05)

Routing messages for network discovery and route table updates.

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Radio synchronization
Sync Word	2	`12 34`	MeshCore network ID
Version	1	`01`	Protocol v1
Flags	1	`08`	Broadcast
Type	1	`05`	ROUTING
Hop Count	1	`00`	First hop
Packet ID	4	`11 22 33 44`	Unique ID
Source ID	4	`DE AD BE EF`	Sender node
Dest ID	4	`FF FF FF FF`	Broadcast
Route Type	1	`01`	Route announce
Neighbor Cnt	1	`03`	3 neighbors
Neighbor 1 ID	4	`11 11 11 11`	Neighbor node 1
Neighbor 1 RSSI	1	`C8`	-56 dBm (signed)
Neighbor 1 SNR	1	`0A`	10 dB SNR
Neighbor 2 ID	4	`22 22 22 22`	Neighbor node 2
Neighbor 2 RSSI	1	`B0`	-80 dBm
Neighbor 2 SNR	1	`05`	5 dB SNR
Neighbor 3 ID	4	`33 33 33 33`	Neighbor node 3
Neighbor 3 RSSI	1	`9C`	-100 dBm
Neighbor 3 SNR	1	`02`	2 dB SNR
MIC	4	`12 34 56 78`	Authentication
CRC	2	`9A BC`	Checksum

Route Type values:

```
0x01 = Route Announce (periodic)        0x03 = Route Reply (response)
0x02 = Route Request (find route)       0x04 = Route Error (route broken)
```

17.4.6 ACK(0x06)

Acknowledgement: confirmation that a message was received.

Field	Bytes	Hex Example	Description
Preamble	8	AA AA AA AA AA AA AA AA	Radio synchronization
Sync Word	2	12 34	MeshCore network ID
Version	1	01	Protocol v1
Flags	1	04	Is ACK flag
Type	1	06	ACK
Hop Count	1	00	First hop
Packet ID	4	99 88 77 66	ACK packet ID
Source ID	4	CA FE BA BE	ACK sender (orig receiver)
Dest ID	4	DE AD BE EF	ACK receiver (orig sender)
ACK'd Pkt ID	4	4F 3A 2B 1C	Original packet ID
Status	1	00	0=OK, 1=Error
MIC	4	DE AD C0 DE	Authentication
CRC	2	FA CE	Checksum

Complete Hex Dump:

```
AA AA AA AA AA AA AA AA 12 34 01 04 06 00 99 88 77 66 CA FE BA BE DE AD BE EF 4F 3A
2B 1C 00 DE AD C0 DE FA CE
```

17.4.7 ROOM MESSAGE

Encrypted group message to all Room members via Room Key.

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Radio synchronization
Sync Word	2	`12 34`	MeshCore network ID
Version	1	`01`	Protocol v1
Flags	1	`09`	Encrypted + Broadcast
Type	1	`01`	TEXT_MESSAGE
Hop Count	1	`00`	First hop
Packet ID	4	`AA BB CC DD`	Unique ID
Source ID	4	`DE AD BE EF`	Sender node
Dest ID	4	`FF FF FF FF`	Broadcast (Room filtered)
Room ID	4	`D0 CW A0 01`	DOCWA Room ID
Timestamp	4	`65 A1 B2 C3`	Unix timestamp
Nonce	12	`XX XX ... (12 bytes)`	Encryptie nonce
Enc Payload	var	`(encrypted)`	Encrypted text
Auth Tag	16	`XX XX ... (16 bytes)`	AES-GCM auth tag
MIC	4	`12 34 56 78`	Packet authentication
CRC	2	`9A BC`	Checksum

Encryption: AES-256-GCM with Room Key

17.4.8 COMMAND MESSAGE

Command messages for remote control of devices via GPIO/UART/I2C.

Field	Bytes	Hex Example	Description
Preamble	8	AA AA AA AA AA AA AA AA	Radio synchronization
Sync Word	2	12 34	MeshCore network ID
Version	1	01	Protocol v1
Flags	1	03	Encrypted + ACK required
Type	1	10	COMMAND (0x10)
Hop Count	1	00	First hop
Packet ID	4	C0 MM AA ND	Unique ID
Source ID	4	DE AD BE EF	Command sender
Dest ID	4	DA K0 DE 01	Target node (roof)
Service ID	2	00 01	GPIO service
Command	2	00 02	SET command
Target	1	05	GPIO pin 5
Value Len	1	02	Value length
Value	var	00 F5	245° (rotor position)
Nonce	12	XX XX ...	Encryptie nonce
Auth Tag	16	XX XX ...	AES-GCM auth
MIC	4	CO DE CA FE	Packet auth
CRC	2	12 34	Checksum

Service IDs:

```
0x0001 = GPIO (pin control)
0x0002 = UART (serial passthrough)
0x0003 = I2C (sensor/device bus)
0x0004 = CAT (transceiver control)
0x0005 = POWER (relay/switch)
0x0006 = ROTOR (antenna rotation)
```

Command codes:

```
0x0001 = GET (read value)
0x0002 = SET (write value)
0x0003 = TOGGLE (flip state)
0x0004 = PULSE (momentary)
```

17.4.9 BEACON MESSAGE

Periodic announcement messages for mesh routing and node discovery.

Field	Bytes	Hex Example	Description
Preamble	8	`AA AA AA AA AA AA AA AA`	Radio synchronization
Sync Word	2	`12 34`	MeshCore network ID
Version	1	`01`	Protocol v1
Flags	1	`08`	Broadcast, no encrypt
Type	1	`07`	BEACON (0x07)
Hop Count	1	`00`	Original beacon
Packet ID	4	`BE AC 0N 01`	Beacon sequence
Source ID	4	`DE AD BE EF`	Beacon node
Dest ID	4	`FF FF FF FF`	Broadcast
Interval	2	`01 2C`	300 sec (5 min)
TX Power	1	`0E`	14 dBm
Capabilities	2	`00 1F`	Feature flags
Uptime	4	`00 01 51 80`	Node uptime
Load	1	`20`	CPU load 32%
Queue	1	`02`	2 packets queued
MIC	4	`BE AC 0N ED`	Authentication
CRC	2	`FA ST`	Checksum

Capability flags:

```
Bit 0: GPS present
Bit 1: Telemetry sensors
Bit 2: Repeater mode
Bit 3: Store & Forward
Bit 4: GPIO services
Bit 5-15: Reserved
```

PART C

Abbreviations and Terminology

18. Abbreviations and Terminology

Alphabetical overview of all technical terms and abbreviations.

Term	Meaning
70 cm-band	Amateur radio band 430-440 MHz, license required
868 MHz	ISM band frequency for Europe
ACK	Acknowledgement - confirmation that a message was received
Advert/Beacon	Periodic signal for presence announcement and public key exchange
AES	Advanced Encryption Standard - versleutelingsalgoritme (128/256-bit)
BLE	Bluetooth Low Energy - energy-efficient connection between node and smartphone
BW	Bandwidth - bandwidth in kHz (125/250/500), smaller = more robust
Callsign	Callsign - unique identification for amateur radio operators (e.g. PE1HVH)
Channel	Shared cryptographic key (PSK) for group communication
Chirp	Frequency sweep from low to high (up-chirp) or high to low (down-chirp)
Companion App	Smartphone application to control the MeshCore node
CR	Coding Rate - error correction level (4/5 to 4/8), more = more reliable
CSS	Chirp Spread Spectrum - the modulation method that LoRa uses
dBm	Decibel-milliwatt - unit for transmit power (14 dBm = 25 mW)
Dechirp	Demodulation by multiplying received chirp with local down-chirp
DFU	Device Firmware Update - updating firmware via Bluetooth
DM	Direct Message - private message between two nodes, end-to-end encrypted
Duty Cycle	Percentage of transmission time allowed (EU: max 1% at 868 MHz)
E2E	End-to-End encryption - only sender and receiver can read messages
EIRP	Effective Isotropic Radiated Power - effectively radiated power incl. antenne
ESP32	Popular microcontroller from Espressif with WiFi and Bluetooth

FFT	Fast Fourier Transform - algorithm for frequency analysis of dechirped signal
Firmware	Software that permanently runs on a node's microcontroller
Flashen	Installing or updating firmware on a device
GPIO	General Purpose Input/Output - connection pins for external devices
GPS/GNSS	Global Navigation Satellite System - satellite navigation for positioning
HAM	Amateur radio mode - license required, no encryption allowed
Hop	One hop between two nodes in the mesh network
I^2C	Inter-Integrated Circuit - bus for connecting sensors and displays
IPEX/U.FL	Small click-on antenna connector for internal antennas
ISM-band	Industrial, Scientific, Medical - free frequency band, 868 MHz in Europe
Key Rotation	Periodically replacing cryptographic keys for extra security
Link Budget	Total signal loss that a connection can tolerate and still be decodable
LoRa	Long Range - patented modulation technique for long-range communication
MCU	Microcontroller Unit - the central processor of a node
Mesh	Network where each device can forward messages to others
Meshtastic	Alternative LoRa mesh firmware - not compatible with MeshCore
MIC	Message Integrity Code - packet authentication (4 bytes)
MIT License	Open-source license for free use, modification and distribution
Node	A device/node in the MeshCore network
Node-ID	Unique identification of a node in the network (4 bytes)
nRF52840	Nordic microcontroller with ultra-low power consumption and Bluetooth
OTA	Over-The-Air - wireless firmware update
PHY	Physical Layer - the physical radio layer that converts bits to radio signals
Preamble	Series of identical chirps at the beginning of each packet for synchronization
Private Key	Private key - kept secret, for decrypting messages
Processing Gain	Signal amplification through spreading over many samples (SF12: ~36 dB)

Public Key	Public key - freely shareable, for encrypting messages
Repeater	Node that forwards messages to extend network range
Room Server	Physical node with BBS function for store-and-forward (up to 32 messages)
Routing	Determining the best route for a message through the network
SF	Spreading Factor - determines range vs speed (SF7-SF12), higher = further
SMA	SubMiniature version A - screw-on antenna connector
SNR	Signal-to-Noise Ratio - ratio between signal and noise in dB
SPI	Serial Peripheral Interface - fast bus for LoRa chip and SD card
Store-and-forward	Store messages until recipient is reachable
SX1262	Semtech LoRa radio chip - newer, more efficient version
SX1276	Semtech LoRa radio chip - older but still used version
Sync Word	Network identifier (2 bytes) to separate different networks
Telemetry	Measurement data from sensors sent through the network
UART	Universal Asynchronous Receiver-Transmitter - serial communication interface
Web Flasher	Browser tool to install firmware without special software
Wrap	Frequency jumps back to beginning of the band (0) after reaching maximum

PART D

References and Sources

19. References and Sources

Comprehensive source list organized by category.

19.1 Datasheets

Titel:	**SX1276/77/78/79 Datasheet**
URL:	https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf
Bron:	Semtech
Description:	Official datasheet for legacy SX127x LoRa transceivers with spreading factors and modulation specs

Titel:	**SX1262 Productpagina**
URL:	https://www.semtech.com/products/wireless-rf/lora-connect/sx1262
Bron:	Semtech
Description:	Product information for the newer SX1262 LoRa transceiver

19.2 Application Notes

Titel:	**AN1200.22 LoRa Modulation Basics**
URL:	https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf
Bron:	Semtech
Description:	Fundamental application note on Chirp Spread Spectrum modulation and processing gain

Titel:	**AN1200.13 LoRa Modem Designer's Guide**
URL:	https://www.openhacks.com/uploadsproductos/loradesignguide_std.pdf
Bron:	Semtech
Description:	Technical manual on SNR requirements and spread spectrum processing gain calculations

Titel:	**RSSI and SNR for LoRa Modulation**
URL:	https://www.st.com/resource/en/application_note/an5664-rssi-and-snr-for-lora-modulation-on-stm32wl-series-stmicroelectronics.pdf
Bron:	STMicroelectronics
Description:	Application note on processing gain calculation and detection of signals below the noise floor

19.3 Academic Papers

Titel:	From Demodulation to Decoding: Complete LoRa PHY
URL:	https://dl.acm.org/doi/10.1145/3546869
Bron:	ACM TOSN
Description:	Deep analysis of LoRa encoding pipeline, dechirping and FFT peak detection

Titel:	Complete Reverse Engineering of LoRa PHY
URL:	https://www.epfl.ch/labs/tcl/wp-content/uploads/2020/02/Reverse_Eng_Report.pdf
Bron:	EPFL
Description:	Detailed reverse engineering report of the LoRa physical layer

Titel:	A Tutorial on CSS for LoRaWAN
URL:	https://arxiv.org/abs/2310.10503
Bron:	arXiv
Description:	Comprehensive academic tutorial on CSS modulation in LoRaWAN systems

Titel:	LoRaWAN Mesh Networks: A Review
URL:	https://pmc.ncbi.nlm.nih.gov/articles/PMC7435450/
Bron:	PMC/MDPI
Description:	Comprehensive overview of multihop mechanisms and mesh topologies for LoRaWAN

Titel:	NELoRa: Ultra-low SNR LoRa Communication
URL:	https://cse.msu.edu/~caozc/papers/sensys21-li.pdf
Bron:	SenSys'21
Description:	Research on dechirp signal concentration and SNR threshold analysis

Titel:	FTrack: Parallel Decoding for LoRa
URL:	https://www4.comp.polyu.edu.hk/~csyqzheng/papers/FTrack_Sensys19.pdf
Bron:	SenSys'19
Description:	Paper on FFT peak detection in dechirped signals

Titel:	Design of a Baseband LoRa Demodulator
URL:	https://ieeexplore.ieee.org/document/8836176/
Bron:	IEEE
Description:	Paper on digital LoRa demodulator design with de-chirp method and FFT

19.4 Educational Resources

Titel:	LoRa/CSS: Overview and Decoding
URL:	https://gyulab.github.io/lora/
Bron:	Gyujun Jeong
Description:	Comprehensive tutorial on CSS modulation, dechirping and FFT-based symbol detection

Titel:	Understanding LoRa PHY
URL:	https://wirelesspi.com/understanding-lora-phy-long-range-physical-layer/
Bron:	Wireless Pi
Description:	Accessible explanation of LoRa physical layer concepts and de-chirping

Titel:	LoRa modem with LimeSDR
URL:	https://myriadrf.org/news/lora-modem-limesdr/
Bron:	MyriadRF
Description:	Practical guide on dechirping through conjugate chirp multiplication

Titel:	The Hidden Side of LoRa
URL:	https://www.disk91.com/2024/technology/lora/the-hidden-side-of-lora/
Bron:	disk91
Description:	Explanation of the internal LoRa packet structure

Titel:	LoRa Link Budget Calculations
URL:	https://www.techplayon.com/lora-link-budget-sensitivity-calculations-example-explained/
Bron:	Techplayon
Description:	Practical explanation of spread spectrum processing gain and SNR requirements

19.5 Standards and Specifications

Titel:	Spreading Factors
URL:	https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/
Bron:	TTN
Description:	Official documentation on spreading factors and receiver sensitivity

Titel:	ETSI EN 300 220
URL:	https://www.etsi.org/
Bron:	ETSI
Description:	European standard for short-range devices in the 868 MHz ISM band

Titel:	LoRa Alliance Regional Parameters
URL:	https://www.lora-alliance.org/
Bron:	LoRa Alliance
Description:	Official specifications for regional frequency plans (EU868, US915)

Titel:	LoRa (Wikipedia)
URL:	https://en.wikipedia.org/wiki/LoRa
Bron:	Wikipedia
Description:	Overview of cyclically shifted chirps, spreading factor selection (SF5-SF12)

19.6 MeshCore Documentation

Titel:	MeshCore Officiële Website
URL:	https://meshcore.co.uk/
Bron:	MeshCore
Description:	Official website with product information, downloads and documentation

Titel:	MeshCore GitHub Repository
URL:	https://github.com/meshcore-dev/MeshCore
Bron:	GitHub
Description:	Open-source repository with MeshCore firmware and example code

Titel:	MeshCore FAQ
URL:	https://github.com/meshcore-dev/MeshCore/wiki/FAQ
Bron:	GitHub Wiki
Description:	Frequently asked questions about advertising, frequencies, firmware and licenses

Titel:	MeshCore Web Flasher
URL:	https://flasher.meshcore.co.uk/
Bron:	MeshCore
Description:	Web-based firmware flash tool for supported devices

Titel:	MeshCore Companion Apps
URL:	https://meshcore.co.uk/apps.html
Bron:	MeshCore
Description:	Overview of available client applications for Android, iOS and web

Titel:	MeshCore Discord Server
URL:	https://discord.gg/ZVH2ujy9ex
Bron:	Discord
Description:	Official community server for support and development discussion

Titel:	MeshCore Kaart
URL:	https://map.meshcore.dev/
Bron:	MeshCore
Description:	Live map with active MeshCore nodes, repeaters and room servers

Titel:	MeshCore Nederlandstalig Forum
URL:	https://forum.meshcore-net.nl/
Bron:	Community
Description:	Dutch-language forum for MeshCore users

19.7 Hardware Documentation

Titel:	Heltec WiFi LoRa 32 V3
URL:	https://heltec.org/project/wifi-lora-32-v3/
Bron:	Heltec
Description:	Product page for Heltec V3 development board with ESP32-S3 and SX1262

Titel:	Heltec WiFi LoRa 32 V4
URL:	https://heltec.org/project/wifi-lora-32-v4/
Bron:	Heltec
Description:	Product page for Heltec V4 with 28dBm transmit power

Titel:	**RAKwireless WisBlock**
URL:	https://store.rakwireless.com/
Bron:	RAKwireless
Description:	Modular IoT hardware system with RAK4631 (nRF52840 + SX1262)

Titel:	**LilyGO T-Deck**
URL:	https://www.lilygo.cc/
Bron:	LilyGO
Description:	Standalone MeshCore device with screen and keyboard

Titel:	**Seeed Studio T1000-E**
URL:	https://www.seeedstudio.com/
Bron:	Seeed Studio
Description:	Compact GPS tracker with nRF52840 and LoRa

— End of Document —