



Hochschule für angewandte Wissenschaften München
Fakultät für Informatik und Mathematik

Masterarbeit zum Thema:

**Entwicklung und Analyse eines LoRaWAN-basierten
Tracking-Systems zur globalen Überwachung von Transportgütern**

English Title

Vorgelegt von: Christoph Schwarz
Matrikelnummer: 32004720
Studiengang: Informatik (Master Embedded Computing)
Betreuer: Prof. Dr. Lars Wischhof
Firma: MicroDoc GmbH
Firma Betreuer: Regine Rudeck
Abgabedatum: ...

1. Abstract

Inhaltsverzeichnis

1. Abstract	II
2. Einleitung	1
2.1. Motivation und Problemstellung	1
2.2. Zielsetzung der Arbeit	1
2.3. Methodisches Vorgehen	1
3. Grundlagen	3
3.1. Drahtlose Datenübertragung	3
3.2. LoRa und LoraWAN	5
3.3. Öffentliche LoraWAN Netzwerke	11
3.4. LoRaWAN Industrielle Anwendungsbereiche	13
3.5. Verwandte Arbeiten	14
4. Konzeptionierung und Entwicklung des Tracking-Systems	16
4.1. Systemanforderungen	16
4.2. Hardware-Design	16
4.3. Tooling Auswahl und Software Design	18
4.4. Implementierung	19
4.5. Anwendungslogik	20
4.6. Energiemanagement und Batterielaufzeit	21
5. Implementierung des Analysewerkzeuges	22
5.1. TTN Mapper	22
5.2. Einflussfaktoren auf die Signalausbreitung (Umwelt, Gebäude, menschliche Aktivität)	22
5.3. Implementierung des Analysewerkzeuges	22
5.4. Modelle zur Schätzung der Verbindungsqualität	22
6. Untersuchung und Auswertung	22
6.1. Überblick und Versuchsaufbau	22
6.2. LoraWAN Tracking-System	22
6.3. Analysewerkzeuge mit Signalausbreitungs-Modellen	22
7. Diskussion der Ergebnisse	22
7.1. Netzwerkabdeckung und Verbindungsqualität	22
7.2. Grenzen und Herausforderungen	22
7.3. Verbesserungspotentiale der Implementierungen	22
7.4. Bewertung der Systemleistung	22
7.5. Einordnung zu bestehenden Arbeiten	22
8. Zusammenfassung und Ausblick	22
8.1. Zusammenfassung der Ergebnisse	22

8.2. Ausblick auf zukünftige Entwicklungen	22
A. Appendix	23
Literatur	24
Abbildungsverzeichnis	28
Tabellenverzeichnis	29
Quellcodeverzeichnis	30

2. Einleitung

<Einleitung>

2.1. Motivation und Problemstellung

Durch die immer größer werdende Globalisierung werden immer mehr Güter verschickt. Zudem steigt ebenso die Nachfrage an transparenten und nachverfolgbaren Transportwegen um die Qualität und Sicherheit der Ware zu gewährleisten. Dies stellt Transportunternehmen vor eine große Herausforderung da die Überwachung zum einen oft über eine sehr große Strecke sowie über Ländergrenzen bewerkstelligt werden muss, zum anderen gibt es eine Vielzahl an zu überwachenden Sendungen und damit an benötigten Trackern.

Traditionelle Tracking-Systeme, die auf Mobilfunktechnologien oder Satellitenkommunikation basieren, sind oft mit hohen Betriebskosten, einem hohen Energieverbrauch oder einer eingeschränkten Reichweite konfrontiert. Dies führt zu Sichtbarkeitslücken in der Lieferkette, was wiederum Risiken wie Diebstahl, Verlust oder Beschädigung von Gütern erhöht.

Vor diesem Hintergrund bietet die Low-Power Wide-Area Network (LPWAN)-Technologie, insbesondere Long-Range Wide-Area-Network (LoRaWAN), vielversprechende Ansätze. LoRaWAN zeichnet sich durch seine Fähigkeit aus, Daten über große Distanzen bei geringem Energieverbrauch zu übertragen. Das macht diese Technologie ideal für Anwendungen im Bereich des Internet of Things (IoT) und der Logistik. Trotz des Potenzials von LoRaWAN für Tracking-Anwendungen, gibt es noch Forschungsbedarf hinsichtlich der praktischen Implementierung und der Analyse der Verbindungsqualität unter realen Bedingungen.

2.2. Zielsetzung der Arbeit

Ziel dieser Arbeit ist die Entwicklung eines modularen Tracking-Geräts, das die in Abschnitt 2.1 beschriebenen Herausforderungen adressiert. Das Gerät soll flexibel an unterschiedliche Anforderungen der Transportlogistik angepasst werden können, um in einer Vielzahl von Einsatzszenarien abzudecken. Des Weiteren soll ein Analysetool entwickelt werden, das mithilfe verschiedener Methoden zur Vorhersage der Verbindungsqualität und der damit verbundenen Netzwerkabdeckung den Transportweg möglichst transparent abbildet. Ziel ist es, potenzielle Signalverluste und Sendeverzögerungen frühzeitig vorherzusagen, um die Zuverlässigkeit der Datenübertragung entlang der Route besser bewerten zu können.

2.3. Methodisches Vorgehen

Das Vorgehen dieser Arbeit gliedert sich in drei Phasen:

Als erstes soll mit der Entwicklung des LoRaWAN Tracking-Gerät begonnen werden. Basierend auf den definierten Anforderungen an Reichweite, Energieverbrauch und Modularität wurde ein prototypisches Tracking-Gerät entwickelt. Dies umfasste die Auswahl geeigneter Hardware-Komponenten, die Entwicklung der Firmware sowie die Integration in öffentliche LoRaWAN-Netzwerke wie The Things Network (TTN) und Helium.

Zum zweiten soll ein Analyse-Werkzeug entwickelt werden das bei der Bewertung der Verbindungsqualität über eine vorgegebene Strecke hilft. Hierzu soll das Werkzeug die RSSI mithilfe von diversen Modellen vorhersagen. Diese Analyse soll dann aufschluss über die Erreichbarkeit des jeweiligen Trackers und mögliche Abdeckungsfreie bereiche geben, die dann für die Planung der jeweiligen Transportroute genutzt werden können.

Als letztes sollen die beiden Komponenten in realen Testszenarien validiert werden. Dazu werden beide Systeme in unterschiedlichen Szenarien getestet, um reale Umgebungsbedingungen und deren Einfluss zu untersuchen. Dabei wurden verschiedene Einflussfaktoren wie Bebauung, Umwelt und Bewegung berücksichtigt. Die gesammelten Daten dienen sowohl der Evaluierung der Systemleistung als auch der Validierung der im Analysewerkzeug implementierten Modelle.

Definition Metriken

Zur Bewertung der Qualität drahtloser Verbindungen werden verschiedene Metriken verwendet, die jeweils unterschiedliche Aspekte der Signalübertragung abbilden. Diese Metriken ermöglichen es, die Qualität eines Datenpfades zu quantifizieren und kritische Schwellenwerte zu definieren, ab denen beispielsweise eine Datenübertragung als instabil oder unzuverlässig gilt.

Im Rahmen dieser Arbeit dienen solche Metriken als Grundlage für die Analyse und Vorhersage der Verbindungsqualität. Dabei wird unter anderem betrachtet, wie stark ein Signal ist, wie hoch der Anteil fehlerfreier Übertragungen ist und welche Umgebungsbedingungen die Werte beeinflussen können.

Konkrete Schwellenwerte, ab denen eine Verbindung als „gut“ oder „kritisch“ eingestuft wird, hängt immer von der eingesetzten Technologie sowie der spezifischen Hardware ab. Die genaue Definition und Anwendung dieser Metriken erfolgt in den nachfolgenden Kapiteln.

3. Grundlagen

Im nachfolgenden Teil sollen die theoretische Grundlagen erläutert werden die für das Verständnis der nachfolgenden Arbeit wichtig sind. Dabei bilden die Prinzipien der drahtlosen Datenübertragung das Fundament, auf dem alle weiteren technischen Konzepte aufbauen.

3.1. Drahtlose Datenübertragung

Drahtlose Datenübertragungsverfahren ermöglichen die Übertragung von Informationen ohne elektrische Leiter. Sie nutzen elektromagnetische Wellen (Radiowellen), Magnetfelder und elektrische Felder als Übertragungsmedium und können somit eine Kommunikation über Entfernungen von mehreren Kilometern oder mehr ermöglichen.

Dieses Grundlegende Prinzip ermöglicht Anwendungen die deutlich portabler und flexibler sind als kabelgebunden Verbindungen. Der Kernmechanismus besteht darin die am Sender als elektrisches Signal Vorliegenden Daten in elektromagnetische Wellen umzuwandeln die sich dann durch die Umgebung ausbreiten können. Diese Signale können dann beim Empfänger wiederum in elektrische Signale umgewandelt und somit interpretiert werden.

Für die drahtlose Telekommunikation werden überwiegend elektromagnetische Wellen, insbesondere Funkwellen, eingesetzt. Drahtlose Kommunikationssysteme arbeiten in verschiedenen Frequenzbändern, die stark reguliert sind, um mögliche Interferenzen zu vermeiden. [19]

Bekannte Modulationsverfahren lassen sich in Analoge und Digitale Verfahren aufteilen. Bei der analogen Modulation werden die Parameter des Trägersignals (Amplitude Modulation (AM) oder Frequenz Modulation (FM)) kontinuierlich entsprechend dem analogen Eingangssignal verändert. Bei der digitalen Modulation hingegen wird zwischen diskreten, fest definierten Zuständen umgeschaltet, um digitale Daten zu übertragen. [53, S. 112 ff. und S. 156 ff.]

Ein einfaches Digitales Übertragungsprotokoll ist hier bei das Frequency Shift Keying (FSK), dabei werden digitale Informationen durch die Variation der Frequenzen eines Trägers kodiert. Im Wesentlichen wird die Trägerfrequenz periodisch zwischen mehreren Frequenzen verschoben, wobei jede Frequenz ein bestimmtes digitales Symbol darstellt.

Das einfachste FSK-Verfahren ist die binäre FSK (Binary FSK, BFSK oder 2-FSK), bei der zwei unterschiedliche Frequenzen verwendet werden, um die Binärziffern '0' und '1' zu repräsentieren. Wie in Beispielsweise Abbildung 1 kann eine höhere Frequenz eine binäre '1' darstellen, während eine niedrigere Frequenz eine binäre '0' repräsentiert. Wenn die zu übertragenden Daten eine '0' enthalten, wird die Trägerfrequenz f_1 verwendet um dieses Bit zu übertragen, wenn die Daten eine '1' sind, wird die Trägerfrequenz f_2 verwendet.[18]

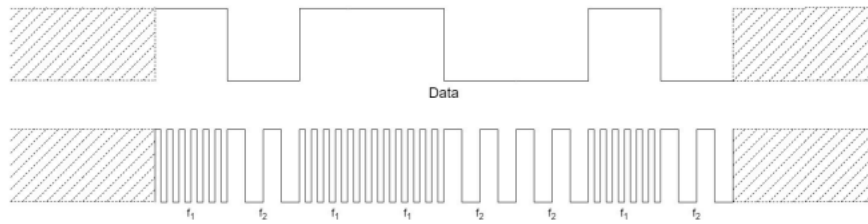


Abbildung 1: Frequency Shift Keying | Quelle: [18]

Frequenz und Bandbreite

Die **Frequenz** beschreibt, wie oft sich ein periodisches Signal pro Sekunde wiederholt. Sie wird in Hertz (Hz) angegeben und ist eng mit der Wellenlänge λ verknüpft. Diese lässt sich aus der Ausbreitungsgeschwindigkeit c des Signals und der Frequenz f über $\lambda = c/f$ berechnen. Höhere Frequenzen entsprechen kürzeren Wellenlängen und erfordern in der Regel kleinere Antennenabmessungen.

Die **Bandbreite** eines Kommunikationskanals bezeichnet den Frequenzbereich, innerhalb dessen ein Signal mit akzeptabler Qualität übertragen werden kann. Sie wird in Hertz angegeben und ist sowohl durch physikalische Eigenschaften des Übertragungsmediums als auch durch elektronische Komponenten begrenzt. Die Bandbreite bestimmt zusammen mit der Signalqualität die maximal mögliche Datenrate. In vielen Kanälen beträgt sie nur einen Bruchteil der Mittenfrequenz, typischerweise zwischen 1% und 10% des Trägerfrequenzbereichs. [33, S. 4]

Schlüsselmetriken der Verbindungsqualität: RSSI, SNR und PDR

Zur Bewertung der Verbindungsqualität werden primär drei Schlüsselmetriken herangezogen: der Received Signal Strength Indicator (RSSI), das Signal-to-Noise Ratio (SNR) und die Packet Delivery Rate (PDR).

Der **RSSI** ist wie im Namen schon enthalten ein Indikator für das empfangene Signal. Er dient als relatives Maß und gibt die Stärke des Funksignals an. Er wird typischerweise in dBm ausgedrückt, wobei Werte näher an Null ein stärkeres empfangenes Signal bedeuten. Faktoren, die den RSSI hauptsächlich beeinflussen, sind der Pfadverlust oder der Antennengewinn. Zudem kommen auch noch weitere Verluste durch Kabel und Steckverbinder, für die Antenne, an Sender und Empfänger hinzu. Darüber hinaus kann der RSSI aufgrund von Signalkollisionen, externen Interferenzen und dem Nah-Fern-Effekt abnehmen. Als direktes Maß für die empfangene Leistung ist der RSSI eine entscheidende Eingangsgröße für viele Ausbreitungsmodelle.

Oft als S/N bezeichnet, stellt das **SNR** das Verhältnis der empfangenen Signalleistung zum Umgebungsrauschpegel dar, üblicherweise in Dezibel (dB) ausgedrückt. Es ist eine kritische Metrik zur Bestimmung der Gesamtqualität des empfangenen Signals. [39] Das SNR wird mit der Formel 1 berechnet.

$$SNR(dB) = P_{receivedSignal}(dBm) - P_{noise}(dBm) \quad (1)$$

Auch als Packet Reception Rate (PRR) bezeichnet, quantifiziert die **PDR** den Prozentsatz der Datenpakete, die erfolgreich über eine Kommunikationsverbindung gesendet und empfangen werden. Sie dient als Maß für die Netzwerkkonsistenz und -leistung. Die PDR wird durch eine Kombination von Faktoren beeinflusst, darunter die Bandbreite, die Übertragungsdistanz und die zugrunde liegende Signalqualität (RSSI, SNR). Insbesondere niedrige RSSI-Werte, insbesondere unter einem bestimmten Schwellenwert (z. B. -120 dBm), können zu Paketkorruption oder vollständigem Verlust führen.

Während der RSSI eine Angabe der Rohsignalstärke liefert, bietet das SNR ein nuancierteres Maß für die Signalqualität relativ zum Rauschen. Letztendlich stellt die PDR die Ergebnisgröße dar, die die kombinierten Effekte von Signalstärke, Signalqualität und vorherrschenden Netzwerkbedingungen zusammenfasst. [36]

3.2. LoRa und LoraWAN

Im Nachfolgenden soll die Funktechnologie auf der diese Arbeit aufbaut beschrieben werden. LoRa (Long Range) bildet dabei die Physikalische Schicht während LoraWAN (Long Range Wide Area Network) die Netzwerkschicht übernimmt. Die genaue aufteilung kann der Abbildung 2 entnommen werden.

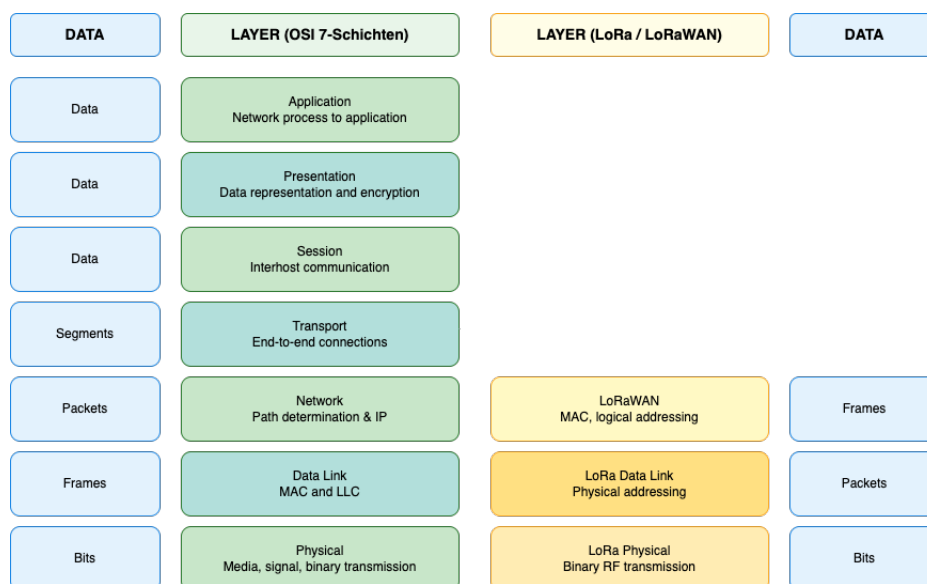


Abbildung 2: LoRa und LoRaWAN im OSI Schichtenmodell | Quelle: ??

LoRa

Einordnung und Überblick LoRa ist ein proprietäres und patentiertes drahtloses Übertragungsverfahren, das von der Semtech Corporation entwickelt wurde. Die Technologie arbeitet auf der physikalischen Schicht (Bitübertragungsschicht) und basiert auf der Spread-Spectrum-Modulationstechnik *Chirp Spread Spectrum* (CSS). Ziel von LoRa ist es, mit sehr

geringem Energieverbrauch eine hohe Reichweite zu ermöglichen, auch unter schwierigen Ausbreitungsbedingungen.

Modulationstechnik chirp Spread Spectrum (CSS) ist ein Modulationsverfahren, bei dem die Frequenz des Signals während der Übertragungszeit kontinuierlich ansteigt oder abfällt. Ein „Upchirp“ (Abbildung 3 rechts) ist eine Erhöhung der Frequenz von niedrig nach hoch, während ein „Downchirp“ (Abbildung 3 links) eine Absenkung der Frequenz von hoch nach niedrig darstellt. Durch diese Methode wird das Datensignal über ein breiteres Frequenzband verteilt, wodurch es robust gegenüber Rauschen und Interferenzen wird. Ein wesentlicher Vorteil von CSS gegenüber anderen Modulationstechniken, wie z. B. der in Abschnitt 3.1 genannten FSK, ist die Fähigkeit, Signale selbst dann zu dekodieren, wenn sie unterhalb des Rauschpegels liegen.

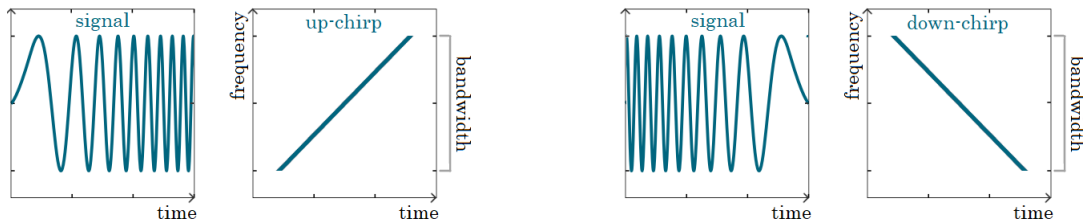


Abbildung 3: LoRa Chirps | Quelle: [45]

Symbolstruktur und Datenübertragung Die Datenübertragung in LoRa erfolgt in Symbolen, die durch eine Folge von Chirps realisiert werden. Ein Symbolindex $s(nT_s)$ wird aus SF (Spreading Factor) Bits gebildet (siehe Formel 2).

$$s(nT_s) = \sum_{h=0}^{SF-1} w_h(nT_s) \cdot 2^h, \quad s \in \{0, 1, \dots, 2^{SF} - 1\} \quad (2)$$

Jedes Symbol hat eine Dauer $T_s = 2^{SF} \cdot T$, wobei $T = \frac{1}{B}$ die Abtastperiode ist und B die genutzte Bandbreite.

Das modulierte Signal für ein Symbol $s(nT_s)$ ist in Formel 3 dargestellt.

$$c(nT_s + kT) = \frac{1}{\sqrt{2^{SF}}} e^{j2\pi \frac{[(s(nT_s) + k) \bmod 2^{SF}] \cdot k}{2^{SF}}}, \quad k = 0, \dots, 2^{SF} - 1 \quad (3)$$

Dieses Signal stellt einen linearen Frequenzanstieg (Upchirp) dar, der um eine frequenzabhängige Startposition (abhängig vom Symbolwert s) verschoben ist. Ein Beispiel für diese Symbole ist in Abbildung 4 dargestellt.

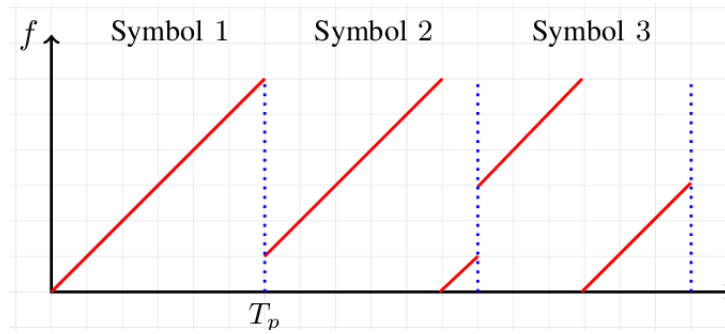


Abbildung 4: LoRa Symbole | Quelle: [27]

Spreading Factor Ein wesentlicher Parameter für LoRa ist der *Spreading Factor* (SF). Er legt fest, wie lange ein Symbol dauert und wie viele verschiedene Symbolwerte pro T_s unterschieden werden können. Technisch gesehen beschreibt der SF das Verhältnis zwischen Chiprate und Symbolrate. Wie aus Gleichung 4 [35, S.6] hervorgeht, wirkt sich der SF direkt auf die erreichbare Datenrate R_b aus:

$$R_b = \frac{SF \cdot B}{2^{SF}} * \frac{4}{4 + CR}, \quad (4)$$

Mit steigendem SF verlängert sich die Dauer eines Symbols, da der zugrunde liegende Chirp über eine größere Zeitspanne ausgesendet wird. Dadurch wird die Übertragung robuster gegenüber Rauschen und Störungen und erlaubt größere Reichweiten. Gleichzeitig sinkt jedoch die Datenrate, da weniger Symbole pro Zeiteinheit übertragen werden können. In der Praxis bedeutet das: Ein niedriger SF (z. B. SF7) ermöglicht eine hohe Datenrate, eignet sich aber nur für kurze Distanzen mit gutem Empfang, während ein hoher SF (z. B. SF12) sehr große Reichweiten erlaubt, jedoch nur eine geringe Datenrate bereitstellt. Der Spreading Factor stellt somit einen zentralen Kompromiss zwischen Reichweite und Datenrate dar.

Signalverarbeitung im Empfänger Die Fähigkeit, Signale unterhalb des Rauschpegels zu erkennen, basiert auf der *Korrelation* zwischen empfangenem Signal und den Referenz-Chirps. Da LoRa-Symbole die gesamte Bandbreite B abtasten, erfolgt eine Art *Spektral-spreizung*, die eine Energieintegration über die gesamte Symbolzeit ermöglicht. Der optimale Empfänger multipliziert das Empfangssignal mit einem Downchirp und führt anschließend eine diskrete Fourier-Transformation (DFT) durch. Das Maximum im Frequenzspektrum gibt den übertragenen Symbolindex an (siehe Formel 5).

$$\hat{s}(nT_s) = \arg \max_p \left| \sum_{k=0}^{2^{SF}-1} r(nT_s + kT) \cdot e^{-j2\pi \frac{k^2}{2^{SF}}} \cdot e^{-j2\pi \frac{pk}{2^{SF}}} \right| \quad (5)$$

Durch die Integration über T_s kann das Signal-Rausch-Verhältnis effektiv erhöht werden, wodurch eine Dekodierung auch unterhalb des Rauschpegels möglich ist. [45, 48]

Regulatorische Rahmenbedingungen (EU) In Europa wird das EU863–870-Band für LoRa verwendet mit *Duty-Cycle*-Grenzen je Subband und Sendeleistungsgrenzen. Gerätekonfiguration und Kanalpläne folgen den *LoRa Alliance Regional Parameters* (RP002), während die verbindlichen Funkparameter in *ETSI EN 300 220* festgelegt sind. [1, 3]

Zusammenfassung LoRa nutzt CSS als Modulationstechnik, wodurch eine robuste Datenübertragung selbst unter dem Rauschpegel möglich ist. Der Spreading Factor ist der zentrale Parameter, der Reichweite und Datenrate bestimmt. Die Datenübertragung erfolgt über symmetrische Chirp-Symbole, die durch Korrelation und Fourier-Analyse effizient erkannt werden können. Damit eignet sich LoRa besonders für Anwendungen mit geringer Datenrate, aber hoher Reichweitenanforderung.

LoRaWAN

LPWANs (Low Power Wide Area Networks) ermöglichen energieeffiziente Kommunikation über große Distanzen und gelten daher als eine Schlüsseltechnologie für das Internet der Dinge (IoT). LoRaWAN zählt zu den vielversprechendsten LPWAN-Technologien. Es bietet geringe Leistungsaufnahme, niedrige Kosten und große Reichweite, geht jedoch mit einer geringen Datenrate einher. Während LoRa lediglich die physikalische Modulation beschreibt, definiert LoRaWAN (Long Range Wide Area Network) das darüberliegende Netzwerkprotokoll und die Architektur für die Kommunikation zwischen Endgeräten, Gateways und Netzwerkservers.

LoRaWAN Netzwerkarchitektur Ein typisches LoRaWAN-Netzwerk besteht aus drei zentralen Komponenten. An erster Stelle stehen die Endgeräte, auch *Nodes* genannt (in Abbildung 5 unter *End Nodes*). Dabei handelt es sich um Sensoren oder Aktoren, die Daten erfassen oder Befehle ausführen können. Diese Geräte besitzen keine direkte Internetanbindung, sondern lediglich einen LoRa-Chip sowie den LoRaWAN-Stack in ihrer Firmware. Alle Daten werden also über diese Schnittstelle empfangen oder gesendet.

Die zweite Komponente bilden die Gateways (in Abbildung 5 unter *Concentrator / Gateway*). Sie sind mit einem LoRa-Chip und einer aktiven Internetverbindung ausgestattet und fungieren als Schnittstelle zwischen der drahtlosen LoRa-Kommunikation und dem IP-Netzwerk. Sobald ein Gateway ein gültiges LoRaWAN-Paket empfängt, leitet es dieses an den Netzwerkservers weiter.

Der Netzwerkservers stellt die zentrale Instanz des gesamten LoRaWAN-Systems dar (in Abbildung 5 unter *Network Server*). Er filtert redundante Nachrichten, übernimmt die Authentifizierung und Sicherheit, steuert die Weiterleitung an die jeweiligen Anwendungsservers und verwaltet gleichzeitig die angebundenen Endgeräte [25].

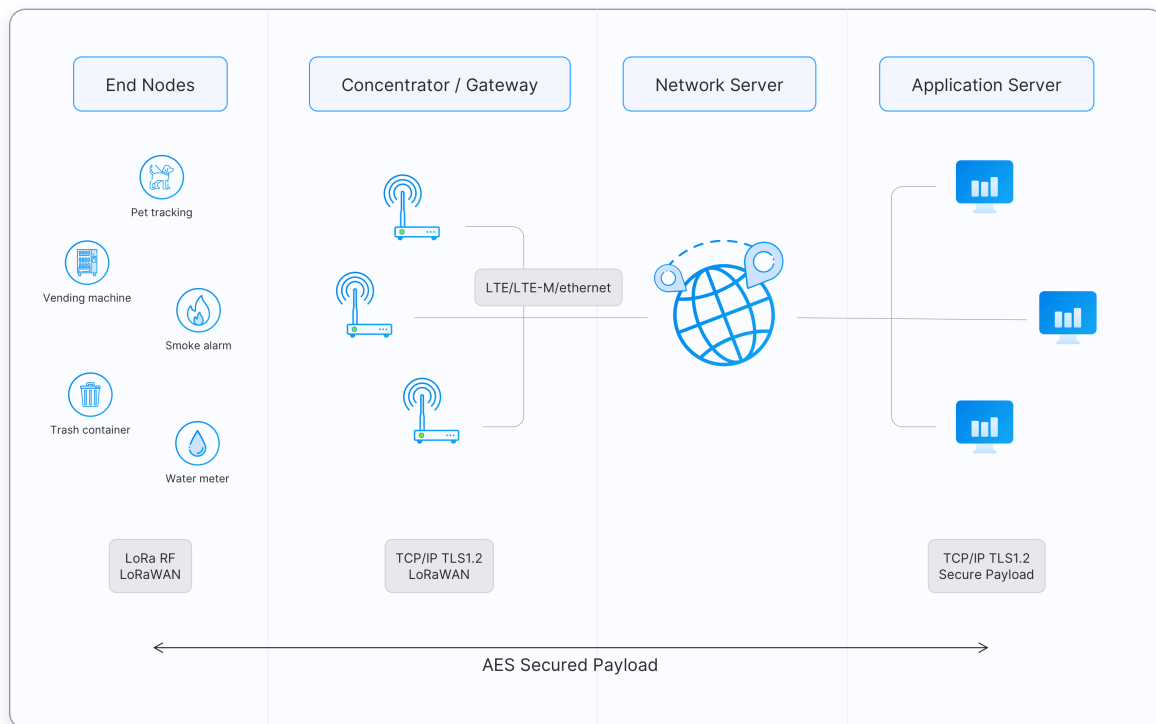


Abbildung 5: LoRaWAN Architektur | Quelle: [25]

LoRaWAN Geräteklassen LoRaWAN unterscheidet drei Gerätekategorien, die sich vor allem in ihren Energiesparmechanismen und Kommunikationsmöglichkeiten unterscheiden. Die energieeffizienteste Variante und zugleich die Mindestanforderung an ein LoRaWAN-Endgerät ist die Geräteklasse **A**. Sie eignet sich besonders für batteriebetriebene Systeme. Wie alle Klassen unterstützt auch Klasse A eine bidirektionale Kommunikation. Um den Energieverbrauch jedoch so gering wie möglich zu halten, können Downlink-Nachrichten vom Netzwerkserver nur kurz nach einem Uplink empfangen werden. Zu diesem Zweck öffnet das Endgerät nach jeder eigenen Übertragung zwei kurze Empfangsfenster (RX1 und RX2).

Die Geräteklasse **B** erweitert dieses Konzept um zusätzliche, periodische Empfangsfenster. Dadurch lassen sich Downlink-Nachrichten planbar zustellen. Jedes Gerät startet zunächst in Klasse A und kann vom Netzwerkserver in Klasse B versetzt werden, sofern es den entsprechenden Standard unterstützt.

Die dritte Kategorie ist die Geräteklasse **C**. Sie ist vor allem für netzbetriebene Geräte gedacht, da ihr Empfangsfenster (RX2) permanent geöffnet ist. Damit wird eine verzögerungsfreie Kommunikation zwischen Endgerät und Netzwerkserver möglich, was allerdings mit einem deutlich höheren Energieverbrauch verbunden ist [30].

LoRaWAN Join-Mechanismen Damit ein Endgerät Nachrichten an ein LoRaWAN-Netzwerk senden kann, muss es sich zunächst anmelden. Hierfür existieren zwei grundlegende Verfahren: *Over-The-Air Activation* (OTAA) und *Activation by Personalization* (ABP).

Bei OTAA, dem empfohlenen und sichersten Mechanismus, authentifiziert sich das Endgerät über einen Join-Vorgang beim Netzwerkservers. Das Gerät sendet dazu eine Join-Request-Nachricht, die unter anderem die eindeutige Geräteerkennung *DevEUI* sowie eine eindeutige Anwendungskennung enthält. In LoRaWAN v1.0.x wird diese als *AppEUI* bezeichnet, während ab v1.1 der Begriff *JoinEUI* verwendet wird. Zusätzlich wird ein *DevNonce* übertragen, ein einmaliger Wert, der sicherstellt, dass Join-Nachrichten nicht wiederverwendet werden können, um Replay-Angriffe zu verhindern. Die Join-Nachricht ist zwar unverschlüsselt, jedoch mit einem MIC (Message Integrity Code) abgesichert.

Nach erfolgreicher Prüfung sendet der Netzwerkservers eine Join-Accept-Nachricht zurück, die weitere Parameter wie *NetID*, *DevAddr* sowie Nonces für die Schlüsselerzeugung enthält. Anschließend werden die Sitzungsschlüssel abgeleitet. In LoRaWAN v1.0.x entstehen aus dem geheimen *AppKey* zwei Session Keys, nämlich der *NwkSKey* für die Netzwerkschicht und der *AppSKey* für die Anwendungsschicht. Ab Version 1.1 unterscheidet LoRaWAN stärker zwischen Netzwerk- und Anwendungsebene. Hier wird der *AppSKey* weiterhin aus dem *AppKey* erzeugt, während drei verschiedene Netzwerkschlüssel (*FNwkSIntKey*, *SNwkSIntKey* und *NwkSEncKey*) aus einem separaten *NwkKey* abgeleitet werden. Durch diese Trennung wird die Sicherheit und Flexibilität bei der Schlüsselverwaltung erhöht.

Im Gegensatz dazu verzichtet ABP vollständig auf einen Join-Prozess. Die für die Kommunikation erforderlichen Parameter, also die Geräteadresse *DevAddr* sowie die Session Keys, werden hierbei direkt im Gerät vorkonfiguriert und auch im Netzwerkservers hinterlegt. In LoRaWAN v1.0.x handelt es sich dabei um den *NwkSKey* und den *AppSKey*, während ab Version 1.1 zusätzlich die drei Netzwerkschlüssel *FNwkSIntKey*, *SNwkSIntKey* und *NwkSEncKey* berücksichtigt werden müssen. ABP ermöglicht somit eine sofortige Nutzung des Netzwerks, bietet jedoch geringere Sicherheit, da die Schlüssel statisch sind und nicht regelmäßig erneuert werden.

Zusammenfassend bietet OTAA durch die dynamische Erzeugung von Sitzungsschlüsseln bei jedem Join deutlich mehr Sicherheit und Flexibilität, während ABP vor allem für Testaufbauten oder spezielle Szenarien mit festen Parametern genutzt wird.

Leistungsmerkmale und Herausforderungen LoRaWAN kann durch Parameteroptimierung an unterschiedliche Anwendungen angepasst werden. Wichtige Designaspekte sind Skalierbarkeit, Durchsatz, Abdeckung, Energieeffizienz und geringe Kosten [9]. Herausforderungen bestehen insbesondere in:

- **Skalierbarkeit:** Beeinflusst durch Faktoren wie verfügbare Kanäle, Spreizfaktor, Bandbreite und regulatorische Einschränkungen. In Europa steht für LoRa das Band von 863 MHz bis 870 MHz zur Verfügung, in Amerika von 902 MHz bis 928 MHz [17].
- **Energieeinsparung:** Durch die Adaptive Data Rate (ADR) oder eine optimierte Parameterwahl kann der Energieverbrauch zusätzlich reduziert werden [23].
- **Sicherheit:** LoRaWAN setzt auf Ende-zu-Ende-Verschlüsselung mit einer klaren Aufgabentrennung zwischen Join Server, Network Server und Application Server. Damit werden Schlüssel sicher verwaltet, Betreiber erhalten nur die für sie notwendigen Informationen, und Multi-Tenant-Szenarien werden unterstützt. Neuere Spezifikationen (ab

v1.1) verbessern zudem die Schlüsselverwaltung durch die Trennung von Anwendungs- und Netzwerkschlüsseln [10, 2].

Anwendungsgebiete LoRaWAN eignet sich für zahlreiche IoT-Szenarien, darunter smarte digitale Städte, smarte Messungen (z. B. für Pflanzen), intelligentes Parkraummanagement oder adaptive Straßenbeleuchtung [8]. Aufgrund der niedrigen Bitrate ist der Einsatz jedoch auf Anwendungen mit geringer Datenübertragungsrate beschränkt.

Roaming, Peering und Interoperabilität LoRaWAN-Roaming erlaubt es, dass Endgeräte auch außerhalb des eigenen Netzwerks funktionieren können. Es gibt zwei Arten von Roaming:

- **Passives Roaming:** Das Endgerät bleibt unter der Kontrolle seines Heimat-Netzservers (Serving NS), die Funkverbindung läuft aber über fremde Gateways und Netzserver (Forwarding NS). Diese leiten die Daten nur weiter, ohne die eigentliche Gerätesteuerung zu übernehmen.
- **Handover Roaming:** Hier übergibt der Heimat-Netzserver die Kontrolle an einen besuchten Netzserver (Serving NS). Dieser steuert dann das Endgerät aktiv, während der Heimat-NS im Hintergrund für die Schlüsselaushandlung und Verwaltung eingebunden bleibt.

Damit Roaming überhaupt funktioniert, müssen Netzbetreiber Peering- oder Roamingabkommen schließen. *Packet Broker* bietet dazu ein globales Vermittlungsnetz, das den Austausch von Paketen zwischen verschiedenen Betreibern erleichtert. Ein Beispiel ist das offene Netz The Things Network (TTN) mit der NetID 000013.

Kommerzielle Anbieter wie Senet setzen zusätzlich auf bilaterale Roaming-Vereinbarungen mit Partnernetzen, um ihre Abdeckung zu erweitern. [2, 32, 31, 38]

3.3. Öffentliche LoraWAN Netzwerke

Öffentliche LoRaWAN-Netze stellen Konnektivität als gemeinschaftliche oder kommerzielle Infrastruktur bereit. Im Gegensatz zu privaten Netzen (unter eigener Kontrolle) unterscheiden sie sich durch *Zugang* (offen vs. vertraglich), *Kostenmodell* (kostenfrei/Fair-Use vs. verbrauchs- oder Service-Level-Agreement (SLA)-basiert ¹), *Peering/Roaming* sowie *Betriebs- und Sicherheitsprozesse*. [2, 1, 3]

Abgrenzung und Kategorien

In der Praxis zeigen sich drei Netzwerktypen:

¹Ein SLA ist eine vertraglich festgelegte Leistungszusage zwischen Anbieter und Nutzer eines Dienstes (z. B. garantierte Verfügbarkeit, Reaktionszeit oder Durchsatz). *SLA-basiert* bedeutet, dass das Netzwerk diese zugesicherten Leistungsmerkmale vertraglich garantiert.

1. **Community-Netze** (z. B. The Things Network, TTN): offen, best-effort, Fair-Use, Enterprise-Plan.
2. **Dezentrale Crowd-Netze** (z. B. Helium IoT): von vielen Hotspot-Betreibern getragen, nutzungsbasierte Abrechnung.
3. **Carrier/Neutral-Host** (z. B. Senet, Everynet): vertraglich, SLA/Peering, nur nationale Abdeckung.

The Things Network (TTN)

TTN ist ein globales Community-Netz auf Basis von *The Things Stack*. Es bietet einen kostenfreien Zugang unter einer Fair-Use-Policy. Die Fair-Use-Policy limitiert in der kostenfreien Version u. a. die Uplink-Airtime pro Gerät und Tag. Genauer sind es „30 Sekunden Uplink-Airtime pro Tag (24 Stunden) und Gerät sowie maximal 10 Downlink-Nachrichten pro Tag (24 Stunden) und Gerät“ [13]. SLAs bestehen im Community-Netz nicht, jedoch bietet die Enterprise-Variante kommerzielle Zusagen. Das Netzwerk nutzt *Packet Broker* als globales Rückrad zur sicheren Weiterleitung von Verkehr zwischen Netzen. Das TTN ist über die Community-Projekte TTN Mapper & Packet-Broker-Mapper kartiert [14, 32, 44].

Die Motivation zur Aufstellung eigener Gateways liegt im gemeinschaftlichen Charakter des Netzes: Durch zusätzliche Gateways wird die Netzabdeckung insgesamt verbessert, wovon wiederum alle Teilnehmer profitieren. Auch der Betreiber selbst erhält dadurch typischerweise eine stabilere Verbindung und bessere Abdeckung für seine eigenen Geräte. Jedoch ohne dass er dadurch mehr Nutzungsrechte erhält. TTN versteht sich damit als offenes, gemeinschaftsgetriebenes Infrastrukturprojekt, das vor allem durch den Nutzen der gemeinsamen Netzressourcen getragen wird.

Helium IoT

Helium IoT ist ein globales LoRaWAN-Netzwerk, das von vielen unabhängigen Teilnehmern betrieben wird. Im Unterschied zu TTN ist die Nutzung nicht kostenlos, sondern erfolgt über sogenannte *Data Credits (DC)*. Diese sind an den US-Dollar gebunden (fester Wert: 1 DC = 0,00001 USD) und können nur durch das *Burning* von HNT-Token erzeugt werden. Dadurch wird ein direkter ökonomischer Anreiz geschaffen, der die Nachfrage nach Netzwerkressourcen mit dem Helium-Krypto-Token verbindet [11].

Ein Uplink von 24 Byte entspricht einem Abrechnungsblock. Downlinks sind in der Regel kostenfrei, können jedoch durch zusätzliche Konfiguration zuverlässiger gestaltet werden. Dafür können Betreiber von LoRaWAN-Network-Servern (LNS) mehrere Gateway-Duplikate pro Uplink „einkaufen“ (*multibuy*), um die Wahrscheinlichkeit einer erfolgreichen Zustellung zu erhöhen [5].

Neben der Nutzung führt das Modell von Helium auch zu einem finanziellen Anreiz für die Aufstellung von Gateways: Ähnlich wie bei anderen Kryptowährungen wird einem Betreiber eines Miners ein wirtschaftlicher Nutzen in Form von Token-Anteilen zugesichert. Betreiber erhalten durch die Bereitstellung eines Gateways und der damit verbundenen Netzwerkabdeckung Vergütungen in Form von HNT-Tokens. Die Höhe dieser Vergütung hängt dabei

maßgeblich von der Erreichbarkeit benachbarter Gateways sowie von der Gateway-Dichte in der jeweiligen Region ab. Dieses Belohnungsmodell führte in der Vergangenheit zu einem schnellen und globalen Ausbau der Netzabdeckung, jedoch mit einer stark variierenden Qualität von Hardware und Installationsbedingungen.

Während das Helium-Netz dadurch eine breite Abdeckung erreichen konnte, ist die resultierende Infrastruktur weniger einheitlich als bei klassischen Community-Netzen. Professionelle Gateways mit optimaler Standortwahl stehen oftmals neben kostengünstigen Geräten, die unter suboptimalen Bedingungen betrieben werden. Dies wirkt sich unmittelbar auf die Netzqualität und Zuverlässigkeit aus, verdeutlicht jedoch zugleich die Stärke des ökonomischen Anreizes als Treiber für den Netzausbau.

Die Netzabdeckung wird u. a. durch *CoverageMap.net* vermessen, ein Service, der ursprünglich aus dem TTN Mapper-Projekt hervorgegangen ist [44].

Carrier-/Neutral-Host-Netze (Beispiele: Senet, Everynet, Actility/ThingPark)

Carrier- bzw. Neutral-Host-Netze bieten Konnektivität auf Basis von vertraglich zugesicherten Service-Level-Agreements (SLAs) an und greifen dabei teilweise auf überregionale Roaming-Abkommen zurück. Ein Beispiel ist *Senet* in den USA, das durch Roaming-Abkommen eine erweiterte Netzabdeckung („Extended Coverage“) bereitstellt, unter anderem auch durch eine Integration mit Helium [38]. Das Unternehmen *Everynet* betreibt Neutral-Host-Netze in mehreren Ländern und bietet zusätzlich Integrationen mit *AWS IoT Core for LoRaWAN* an [22]. *Actility* mit seiner Plattform *ThingPark* gilt als einer der verbreitetsten Anbieter in diesem Bereich und stellt sowohl Roaming-Funktionalitäten als auch hybride Szenarien für verschiedene Betreiber bereit [3].

Praktische Einordnung für diese Arbeit

Für prototypische *Tracking*-Workloads sind TTN (kostenfrei, Fair-Use) und Helium (breite Crowd-Abdeckung, DC-basiert) aufgrund ihrer großen Abdeckung und einfachen Anwendung naheliegend. Während TTN durch seine Community-getragene Struktur häufig qualitativ hochwertige Gateways bietet, die von technisch versierten Betreibern (z. B. Universitäten oder Kommunen) installiert werden, zeichnet sich Helium durch eine sehr große Anzahl an privat betriebenen Hotspots aus. Dies führt zu einer breiten globalen Abdeckung, jedoch erreichen die im Helium-Netz eingesetzten Gateways häufig nicht die Qualitätsstandards, wie sie im TTN-Umfeld üblich sind. Gründe hierfür sind unter anderem weniger optimale Installationsorte sowie der Einsatz kostengünstigerer Hardware, beispielsweise bei Antennen und Kabeln. TTN eignet sich somit besonders für stabile Prototyping-Szenarien mit gut dokumentierten Tools und Community-Support, während Helium Vorteile bei einer schnellen globalen Skalierbarkeit bietet.

3.4. LoRaWAN Industrielle Anwendungsbereiche

LoRaWAN (Long Range Wide Area Network) bietet durch seine Charakteristiken die in Abschnitt 3.2 beschrieben viele Vorteile für industrielle Anwendungen.

Vorausschauende Instandhaltung LoRaWAN ermöglicht eine zuverlässige und energiearme Übertragung von Zustandsdaten (z. B. Vibration, Temperatur) die für frühzeitige Wartungserkennung genutzt werden können. Besonders im Industrie-4.0-Umfeld lassen sich Sensoren schnell installieren, mit Legacy-Geräten koppeln und in Cloud-basierte Analyseplattformen integrieren, um Effizienz, Sicherheit und Zuverlässigkeit zu steigern [26].

Telemetrie und Alarmüberwachung LoRaWAN eignet sich für die gleichzeitige Übertragung von periodischen Telemetriedaten und seltenen Alarmmeldungen. Durch unterschiedliche Spreading-Factor-Zuweisung und gezielte Retransmission kann hohe Zuverlässigkeit für Alarmer bei minimaler Verzögerung erreicht werden—ohne die reguläre Kommunikation zu beeinträchtigen [37].

Umwelt- und Gefahrenüberwachung In industriellen Umgebungen mit Unfall- oder Störfallrisiko (z. B. toxische Gase) erlaubt ein LoRaWAN-System mit optimiertem Downlink-Control-Packet-Mechanismus zuverlässige, latenzarme Umweltdatenübermittlung zur schnellen Gefahrenabwehr [42].

3.5. Verwandte Arbeiten

In diesem Abschnitt wird der aktuelle Forschungs- und Entwicklungsstand zu LoRaWAN im Kontext von Tracking-Systemen dargestellt. Ziel ist es, existierende Studien und Projekte zu analysieren, die sich mit der Leistungsfähigkeit, Energieeffizienz und Praxistauglichkeit von LoRaWAN im Bereich der globalen Güterüberwachung beschäftigen.

LoRaWAN im Vergleich zu anderen LPWAN-Technologien Mehrere Arbeiten vergleichen LoRaWAN mit alternativen LPWAN-Technologien wie Sigfox, NB-IoT und LTE-M. Studien zeigen, dass LoRaWAN insbesondere durch seine Flexibilität und die Verfügbarkeit von Community-Netzen (z. B. The Things Network genauer in Abschnitt 3.3 beschrieben) oft einfacher zu verwenden ist, während NB-IoT Vorteile in Netzstabilität und QoS bietet dafür aber auf lizenzierten Frequenzbändern arbeiten [28, 49, 4].

Leistungsfähigkeit und Reichweite Analysen der Reichweite von LoRaWAN zeigen, dass in urbanen Umgebungen typische Reichweiten von 2–5 km und in ländlichen Gebieten bis zu 15 km erreicht werden können [7, 21]. Faktoren wie Bebauung, Antennenhöhe und Spreading Factor haben signifikanten Einfluss auf die Verbindungsqualität.

Energieeffizienz Für batteriebetriebene Endgeräte ist die Energiebilanz entscheidend. Arbeiten wie [16, 34, 20] zeigen, dass LoRaWAN durch die Class-A-Architektur eine hohe Energieeffizienz erreicht, die Laufzeiten von mehreren Jahren bei typischen Sensordatenraten ermöglicht. Allerdings erfordert die Konfiguration des Duty-Cycles und der Sendeintervallen eine Balance zwischen Datenaktualität und Batterielaufzeit.

Anwendungsgebiete in der Logistik LoRaWAN wird in zahlreichen Szenarien des IOTs eingesetzt, insbesondere in der Logistik für Güterverfolgung und Anwendungen in intelligenten Städten. Charakteristisch für die Anwendungsszenarien sind niedrige Datenraten, extrem geringer Energieverbrauch und lange Batterielaufzeiten, was für großflächige, wartungsarme Sensornetze wichtig ist. Im Bereich der Logistik werden vor allem Anwendungen wie die Überwachung von Transportgütern, Temperatur- oder Positionsdaten und allgemeine Güterverfolgung adressiert [20]. Darüber hinaus betonen beide Studien die Bedeutung von Erweiterbarkeit und Zusammenarbeit zwischen verschiedenen Netzwerken, was durch Mechanismen wie Netzwechsel unterstützt werden kann [34].

4. Konzeptionierung und Entwicklung des Tracking-Systems

Aufbauend auf den in Kapitel 3 dargestellten theoretischen Grundlagen wird in diesem Kapitel die konkrete Umsetzung des LoRaWAN-basierten Tracking-Systems beschrieben. Der Schwerpunkt liegt auf der Konzeptionierung der Systemarchitektur, der Auswahl geeigneter Hardwarekomponenten sowie der Entwicklung der Software. Dabei wird insbesondere auf die Anforderungen an Modularität, Energieeffizienz und Erweiterbarkeit eingegangen. Ziel war es, ein prototypisches System zu realisieren, das als Grundlage für die spätere Analyse der Verbindungsqualität und der praktischen Einsatzmöglichkeiten dient.

4.1. Systemanforderungen

Die Entwicklung des Tracking-Systems erfolgte unter der Prämisse, zunächst einen funktionsfähigen Prototypen als Proof-of-Concept zu realisieren. Dabei standen Aspekte wie Kostenoptimierung oder Robustheit im industriellen Einsatz nicht im Vordergrund. Stattdessen lag der Schwerpunkt auf Modularität und Erweiterbarkeit, um eine Plattform zu schaffen, die flexibel an unterschiedliche Szenarien angepasst werden kann.

Als Basiskomponenten wurden eine GPS-Einheit zur Positionsbestimmung sowie eine SD-Karte zur lokalen Zwischenspeicherung vorgesehen. Externe Sensoren, beispielsweise zur Temperaturerfassung oder zur Anbindung von Sicherheitssiegeln (elektronische Plomben), sollten über modulare Schnittstellen integriert werden können. Dadurch ergibt sich eine universelle Plattform, die sich für unterschiedliche Anforderungen in der Transportlogistik anpassen lässt.

Für die Kommunikation wurde der Einsatz von LoRaWAN spezifiziert. Aufgrund der Fokussierung auf energieeffiziente Uplink-Übertragungen ohne Bedarf an kontinuierlichen Downlinks wurde die Geräteklasse A gewählt. Diese stellt die grundlegende und am weitesten verbreitete Gerätekategorie dar, die insbesondere für batteriebetriebene Endgeräte geeignet ist. Eine Bindung an ein spezifisches öffentliches oder privates Netzwerk wurde im Rahmen der Konzeption nicht vorgenommen, um die Flexibilität in der prototypischen Umsetzung zu wahren.

Neben der Modularität wurde auch der Energieverbrauch als zentrale Anforderung berücksichtigt. Zwar war die Laufzeitoptimierung nicht vorrangiges Ziel der ersten Implementierung, dennoch sollte die Architektur grundsätzlich energieeffiziente Betriebsmodi unterstützen. Ergänzend dazu bietet LoRaWAN inhärente Sicherheitsmechanismen wie Ende-zu-Ende-Verschlüsselung und Schlüsselverwaltung, die für die angestrebten Einsatzszenarien als ausreichend betrachtet wurden.

4.2. Hardware-Design

Für die prototypische Umsetzung des Tracking-Systems wurden zwei unterschiedliche Entwicklungsboards verwendet: das **LoRa-E5 Mini** von Seeed Studio sowie das **Heltec LoRa V2**. Beide Plattformen bieten eine gute Ausgangsbasis für eine schnelle Implementierung,

unterscheiden sich jedoch in wesentlichen Aspekten, die im Rahmen der Konzeption berücksichtigt wurden.

Wie in Abbildung 6 zu erkennen basiert das LoRa-E5 Mini auf einem STM32WLE5-Chip, der sowohl einen Mikrocontroller als auch ein integriertes LoRaWAN-Modul bereitstellt. Durch diese enge Integration reduziert sich die Komplexität der Schaltung, da keine externe Anbindung eines Funkmoduls erforderlich ist. Darüber hinaus verfügt das Board über einen SMA-Anschluss für externe Antennen, was die Flexibilität im Test erhöht.

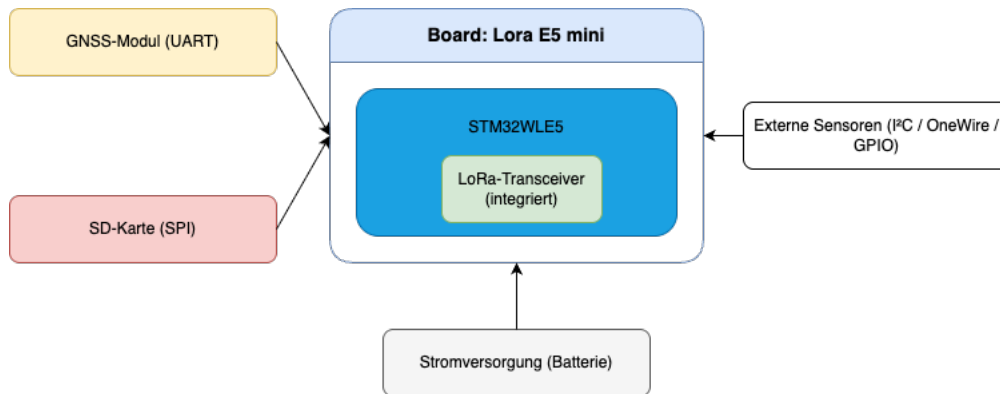


Abbildung 6: Systemarchitektur Lora E5 mini

Das Heltec LoRa V2 hingegen basiert auf einem ESP32-Mikrocontroller, wie in Abbildung 7 zu sehen, ist bei diesem Board der LoRa-Chip separat angebunden. Der ESP32 bietet zudem umfangreiche Energiesparmodi, insbesondere den Deep-Sleep-Modus, wodurch sich verschiedene Ansätze zur Laufzeitoptimierung evaluieren lassen. Zusätzlich verfügt das Board über integrierte Peripheriekomponenten wie ein OLED-Display, das für Debugging und Testzwecke nützlich war.

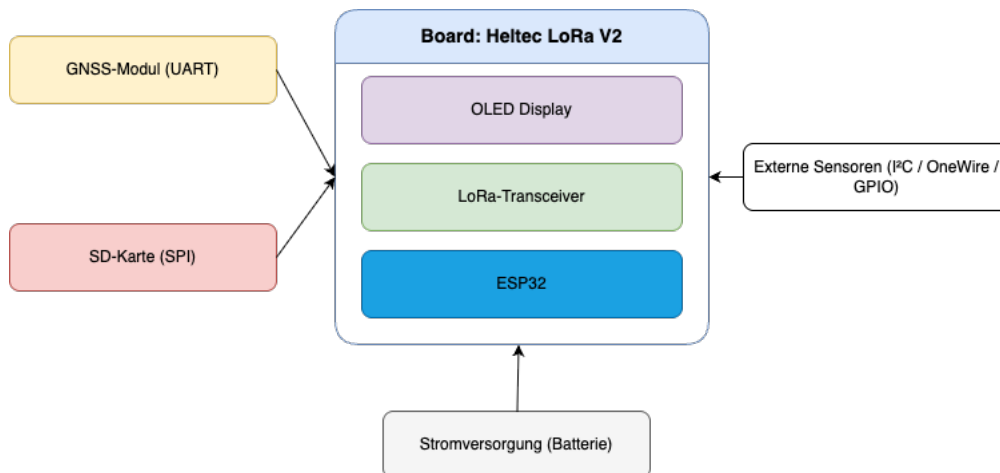


Abbildung 7: Systemarchitektur Heltec LoRa V2

Neben den Entwicklungsboards bildeten zwei weitere Komponenten zentrale Elemente der

Hardwarearchitektur:

- **GPS-Modul:** Das GNSS-Modul dient primär der Positionsbestimmung und stellt die wesentlichen Geodaten für das Tracking bereit. Darüber hinaus wird die vom Modul gelieferte Zeitinformation als Zeitbasis für das Gesamtsystem genutzt, wodurch das Gerät ohne separate Echtzeituhr über eine zuverlässige Zeitreferenz verfügt.
- **SD-Karte:** Die SD-Karte wird einerseits für das lokale Logging der Messdaten eingesetzt, sodass auch bei temporären Übertragungsproblemen eine vollständige Aufzeichnung gewährleistet bleibt. Andererseits speichert sie persistente Systemparameter, wie das aktuell konfigurierte Sendeintervall sowie den LoRaWAN-Nonce-Wert, der für die sichere Teilnahme am Netzwerk erforderlich ist. Dadurch lassen sich sowohl Sicherheit als auch Reproduzierbarkeit der Übertragungen erhöhen.

Die Grundarchitektur der Prototypen sah damit eine zentrale Mikrocontroller-Einheit mit angebundener Peripherie vor. Das GPS-Modul wurde über eine UART-Schnittstelle angeschlossen, während die SD-Karte über SPI verbunden war. Weitere Sensoren, wie beispielsweise Temperatursensoren, konnten über gängige Schnittstellen wie I²C oder OneWire ergänzt werden.

Die Entwicklung erfolgte auf Basis von Entwicklungsboards in Verbindung mit Breadboards, wodurch eine schnelle Iteration und flexible Erweiterbarkeit möglich war. Auf die Anfertigung eigener Leiterplatten wurde im Rahmen des Proof-of-Concepts bewusst verzichtet, da der Fokus auf Funktionalität und Modularität lag und nicht auf der finalen Produktoptimierung.

4.3. Tooling Auswahl und Software Design

Für die Entwicklung der Firmware wurde das Zephyr Real-Time Operating System (RTOS) eingesetzt. Zephyr bietet eine modulare Architektur auf Basis von Device Trees. Dies ermöglicht eine gemeinsame Codebasis für alle Microcontroller und Peripheriegeräten. Darüber hinaus stellt das Framework eine Vielzahl vorgefertigter Treiber und Bibliotheken bereit, die für die vorliegende Arbeit unmittelbar nutzbar waren. Dazu zählen unter anderem Implementierungen für LoRaWAN, UART-gebundene GPS-Module, SD-Karten mit Dateisystemunterstützung, Displays (z. B. OLED beim ESP32-basierten Board) sowie Schnittstellen wie OneWire und I²C für externe Sensoren.

Die Implementierung der Firmware erfolgte in C++, da diese Sprache im Embedded-Bereich weit verbreitet ist und ebenso einer der 3 Hauptsprache, neben C und Rust, für Zephyr RTOS ist [24]. Auf zusätzliche externe Libraries wurde verzichtet, da die Funktionalität durch die Zephyr-Basis bereits vollständig abgedeckt war.

Das Software-Design folgt einem modularen Ansatz: Für jede Komponente existiert ein eigenständiges Modul, das während des Systemstarts initialisiert wird. Hierzu gehören insbesondere die SD-Karte, das GPS-Modul sowie beliebige Sensorschnittstellen. Alle implementierten externen Sensoren besitzen eine einheitliche `read()`-Methode, über die Messwerte ausgelesen werden können. Die Auswahl und Aktivierung einzelner Module erfolgt über die

Projektkonfiguration in der Datei `prj.conf`. Auf diese Weise kann beispielsweise über den Eintrag:

```
lora_wan_tracker_temp_sensor=y
```

die Einbindung eines Temperatursensors aktiviert werden. Dieser Ansatz ermöglicht eine flexible Anpassung der Firmware an unterschiedliche Anwendungsfälle, ohne tiefgreifende Änderungen am Quellcode vorzunehmen.

Ein Bestandteil des Software-Designs ist die Datenpaketstruktur, mit der die gesammelten Messwerte über LoRaWAN übertragen werden. Jedes Paket enthält zunächst die GPS-Daten und den Zeitstempel des Geräts in komprimierter Form. Hierzu werden Breiten- und Längengrad, Höhe, Genauigkeit sowie die Anzahl empfangener Satelliten als Big-Endian-Werte in ein Byte-Array codiert.

Im Anschluss an die GPS-Daten und Zeitdaten wird ein 4-Bit-Feld angefügt, das als Bitmap fungiert. Jedes Bit steht für einen spezifischen Sensor und zeigt an, ob für diesen Sensor Messwerte im aktuellen Paket enthalten sind. Die eigentlichen Sensordaten folgen in einer festgelegten Reihenfolge. Diese Struktur erlaubt eine flexible Erweiterung, da neue Sensortypen durch die Definition zusätzlicher Bits in der Bitmap ergänzt werden können, ohne die bestehende Protokollstruktur zu verändern. Eine Vollständige Ansicht des Paketes kann Abbildung 8 entnommen werden.

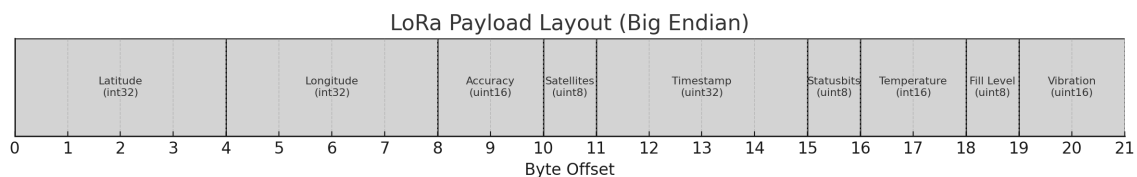


Abbildung 8: Datensegment des LoraWAN Pakets

Auf der Seite des LoRaWAN Network Servers (LNS) wird das Datenpaket mittels JavaScript entschlüsselt. Die dort implementierte Routine interpretiert die Bitfolge und überführt die enthaltenen Werte in ein JSON-Format, welches dann für nachgelagerte Anwendungen, wie Visualisierung oder Analyse, weiterverarbeitet werden kann.

Die gewählte Architektur vereint damit eine modulare Firmwarestruktur, die sich flexibel anpassen lässt, mit einer kompakten und erweiterbaren Datenrepräsentation, die für den Einsatz in ressourcenbeschränkten IoT-Geräten optimiert ist.

4.4. Implementierung

Im nachfolgenden soll die Implementierung des LoraWAN-Trackers beschrieben werden. Dabei wird der Systemstart und die Initialisierung der einzelnen Komponenten sowie der Datenpfad und das Paketformat beschreiben. Des weiteren wird der Netzbeitritt so wie das Senden der Pakete erklärt. Auch wird beschreiben wie das Gerät Konfiguriert werden kann sowie die Interaktion mit der Shell beschrieben. Zuletzt soll noch drauf eingegangen werden wie die Fehlerbehandlung sowie das Logging funktioniert.

Systemstart und Initialisierung Beim Start initialisiert die Firmware die Subsysteme in fester Reihenfolge: Shell, LED/GPIO, SD-Karte (FATFS, Mount-Punkt /SD:), GNSS-UART, LoRaWAN-Stack. Das LoRaWAN-Subsystem wird über die Zephyr-API gestartet und anschließend für EU868 konfiguriert. Ebenso wird ADR ist aktiviert. Die SD-Karte wird über das Disk-Access-API angebunden und als FATFS eingehängt. GNSS läuft als NMEA-Quelle am konfigurierten UART mit Callback-Registrierung.

Datenpfad und Paketformat GNSS-Daten werden über einen Callback gelesen, aufbereitet und in eine kompakte Payload kodiert der genaue Aufbau des einzelnen Pakets wurde in Abschnitt 4.3 erklärt. Breite und Länge als `int64` big-endian, Höhe `int32`, Genauigkeit `uint16`, Satelliten `uint8`. Die Kodierung erfolgt in ein Byte-Array fester Länge. Die Position wird nur bei gültigem Fix versendet.

Netzbeitritt und Senden Der Netzbeitritt nutzt OTAA (wie in Abschnitt ?? Join-Mechanismus (OTAA) erklärt). Die Join-Parameter (`dev_eui`, `join_eui`, `app_key`) werden je nach Zielnetz, TTN oder Helium, gesetzt. Der `dev_nonce` wird persistent verwaltet und vor jedem Join erhöht. Nach erfolgreichem Join sendet das Gerät bestätigte Uplinks (CONFIRMED) auf Port 2.

Persistenz und Konfiguration Sendeintervall (`/SD:/int.txt`) und LoRaWAN-Nonce (`/SD:/nonce.txt`) sind persistent gespeichert. Beide werden beim Start gelesen. Das Intervall kann zur Laufzeit über die Shell gesetzt werden und wird sofort wirksam. Dateioperationen nutzen Zephyrs VFS und FATFS-Implementierung.

Shell-Befehle Die Shell implementiert Befehle zum Setzen des Sendeintervalls sowie zum Abfragen von Positiondaten über das GPS modul. Ebenso kann ein Zähler der erfolgreicher Übertragungen ausgelesen werden. Die Implementierung verwendet das Zephyr-Shell-Subsystem mit statischen Kommandoregistrierungen.

Fehlerbehandlung und Logging Wenn der Tracker in einen Fehlerzustand läuft wird das durch eine blinkende LED-Schleife auf dem jeweiligen Dev-Board signalisiert. Jede Sendeoperation wird mit Positionsdaten, RSSI/SNR (falls vorhanden) und Erfolgsflag in `/SD:/log.txt` protokolliert, um spätere Auswertungen zu ermöglichen. Downlink-Metadaten werden über einen registrierten Downlink-Callback erfasst.

4.5. Anwendungslogik

Das Tracking Gerät hat zwei grundlegende Betriebsmodi die im Nachfolgenden kurz erläutert werden sollen:

- **Periodisches Senden:** Daten werden in festen Zeit Intervallen übermittelt (z. B. alle 5, 15 oder 60 Minuten). Dieser Ansatz ist einfach und sorgt für eine kontinuierliche Nachverfolgbarkeit über die Transportstrecke. Allerdings steigt der Energieverbrauch deutlich mit kürzer werdenden Intervallen [50, 47].

- **Ereignisgesteuertes Senden:** Eine Übertragung erfolgt nur bei bestimmten Ereignissen, etwa beim Öffnen einer Plombe, Überschreiten eines Temperaturgrenzwertes oder bei Erschütterung des Transportguts. Dadurch lässt sich die Batterielaufzeit zwar signifikant verlängern, jedoch sind die Positionsdaten zwischen den Ereignissen lückenhaft [43, 6].

Am besten geeignet ist deswegen ein Hybrider Ansatz. Dabei sollen regelmäßige Intervall-Updates mit großen Intervallen mit zusätzlichen Ereignisübertragungen kombiniert werden. Dadurch kann eine Balance zwischen Energieeffizienz und Transparenz der Transportüberwachung erreicht werden.

4.6. Energiemanagement und Batterielaufzeit

Die Herstellerangaben der wesentlichen Komponenten sind in Tabelle 1 zusammengefasst. Für jede Betriebsart sind typische Ströme aufgeführt. Ebenso wurden eigene Messungen angestellt um die bestehenden Werte zu validieren.

Komponente	Modus	Strom Datenblatt	Messung
LoRa-E5 Mini	DS	2,1 μ A [40]	...
	TX 22 dBm	122 mA [52]	...
	RX	5,5 mA [52]	...
Heltec ESP32 V2	DS	10 μ A [15]	...
	MCU aktiv	20–80 mA [15]	...
	LoRa TX 20 dBm	120 mA [41]	...
GPS (NEO-6M/ATGM336H)	Acquisition	45 mA [46]	...
	Tracking	25 mA [46]	...
SD-Karte	Idle	0,2–0,4 mA [sandisk_sd]	...
	Write	30–100 mA [sandisk_sd]	...
DS18B20	Conversion	1,5 mA [12]	...
	Standby	<1 μ A [12]	...
MPU-6050	Aktiv	3,9 mA [29]	...
	Sleep	8 μ A [29]	...
OLED (Heltec)	On	20–30 mA [51]	...
Plombe (GPIO)	Idle	<1 μ A	...

Tabelle 1: Stromaufnahme der Komponenten.

5. Implementierung des Analysewerkzeuges

5.1. TTN Mapper

5.2. Einflussfaktoren auf die Signalausbreitung (Umwelt, Gebäude, menschliche Aktivität)

5.3. Implementierung des Analysewerkzeuges

5.4. Modelle zur Schätzung der Verbindungsqualität

Log-Distanz-Modell (Log-Distance Path Loss Model)

Noch eins

Machine Learning Modell (DeepLora)

6. Untersuchung und Auswertung

6.1. Überblick und Versuchsaufbau

6.2. LoraWAN Tracking-System

6.3. Analysewerkzeuge mit Signalausbreitungs-Modellen

7. Diskussion der Ergebnisse

7.1. Netzwerkabdeckung und Verbindungsqualität

7.2. Grenzen und Herausforderungen

7.3. Verbesserungspotentiale der Implementierungen

7.4. Bewertung der Systemleistung

7.5. Einordnung zu bestehenden Arbeiten

8. Zusammenfassung und Ausblick

8.1. Zusammenfassung der Ergebnisse

8.2. Ausblick auf zukünftige Entwicklungen

A. Appendix

Literatur

- [1] A. YEGIN (Actility), O. SELLER (Semtech), D. KJENDAL (Senet), D. KJENDAL (Senet). *LoRaWAN® Regional Parameters RP002-1.0.4*. Sep. 2022. (Besucht am 27.08.2025).
- [2] A.YEGIN (Actility), O.SELLER (Semtech), A.YEGIN (Actility). *LoRaWAN Backend Interfaces Specification*. (Besucht am 27.08.2025).
- [3] *Actility: IoT Solutions for a Smarter World*. (Besucht am 27.08.2025).
- [4] Ferran Adelantado u. a. „Understanding the Limits of LoRaWAN“. In: *IEEE Communications Magazine* 55.9 (Sep. 2017), S. 34–40. ISSN: 1558-1896. DOI: 10.1109/MCOM.2017.1600613. (Besucht am 31.08.2025).
- [5] *Advanced Configuration | Helium Documentation*. <https://docs.helium.com/iot/run-an-ns/advanced-configuration/>. (Besucht am 27.08.2025).
- [6] Vasileios Asteriou u. a. „LoRaWAN-based Adaptive MACs for Event Response Applications“. In: *IEEE Access* PP (Jan. 2022), S. 1–1. DOI: 10.1109/ACCESS.2022.3204654.
- [7] Aloÿs Augustin u. a. „A Study of LoRa: Long Range & Low Power Networks for the Internet of Things“. In: *Sensors* 16 (Sep. 2016), S. 1466. DOI: 10.3390/s16091466.
- [8] *Baden-Württemberg fördert Long Range Wide Area Networks in Städten und Gemeinden*. <https://im.baden-wuerttemberg.de/de/service/presse-und-oeffentlichkeitsarbeit/pressemittteilung/pwuerttemberg-foerdert-long-range-wide-area-networks-in-staedten-und-gemeinden>. Juni 2024. (Besucht am 28.08.2025).
- [9] Martin Bor und Utz Roedig. „LoRa Transmission Parameter Selection“. In: *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2017, S. 27–34. DOI: 10.1109/DCOSS.2017.10.
- [10] Ismail Butun, Nuno Pereira und Mikael Gidlund. „Security Risk Analysis of LoRaWAN and Future Directions“. In: *Future Internet* 11.3 (2019). ISSN: 1999-5903. DOI: 10.3390/fi11010003.
- [11] *Data Credit | Helium Documentation*. <https://docs.helium.com/tokens/data-credit/>. (Besucht am 27.08.2025).
- [12] „DS18B20 - Programmable Resolution 1-Wire Digital Thermometer“. In: ().
- [13] *Duty Cycle*. <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/>. (Besucht am 01.09.2025).
- [14] *Duty Cycle*. <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/>. (Besucht am 27.08.2025).
- [15] *ESP32 Series Datasheet*. (Besucht am 31.08.2025).
- [16] Vitor Fialho und Fernando Fortes. „Battery Lifetime Estimation for LoRaWAN“. In: *International Journal of Innovative Technology and Exploring Engineering* 9 (Sep. 2020), S. 306–310. DOI: 10.35940/ijitee.K7824.0991120.

- [17] *Frequency Plans*. <https://www.thethingsindustries.com/docs/concepts/features/lorawan/frequency-plans/>. (Besucht am 27. 08. 2025).
- [18] „Frequency-Shift Keying Modulation and Demodulation Using the Timer/Counter Peripherals on the AVR® EB“. In: (2024).
- [19] *Grundkenntnisse der drahtlosen Kommunikation: Drahtloser Mechanismus (1) | Murata Manufacturing Articles*. <https://article.murata.com/de/article/basics-of-wireless-communication-1>. Dez. 2023. (Besucht am 06. 08. 2025).
- [20] Jetmir Haxhibeqiri u. a. „A Survey of LoRaWAN for IoT: From Technology to Application“. In: *Sensors* 18 (Nov. 2018), S. 3995. DOI: 10.3390/s18113995.
- [21] Jetmir Haxhibeqiri u. a. *LoRa Indoor Coverage and Performance in an Industrial Environment: Case Study*. Sep. 2017, S. 8. DOI: 10.1109/ETFA.2017.8247601.
- [22] *Introducing Support for Public Networks with AWS IoT Core for LoRaWAN | The Internet of Things on AWS – Official Blog*. <https://aws.amazon.com/blogs/iot/introducing-support-for-public-networks-with-aws-iot-core-for-lorawan/>. Juni 2023. (Besucht am 27. 08. 2025).
- [23] Rachel Kufakunesu, Gerhard P. Hancke und Adnan M. Abu-Mahfouz. „A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges“. In: *Sensors* 20.18 (Sep. 2020), S. 5044. ISSN: 1424-8220. DOI: 10.3390/s20185044. (Besucht am 27. 08. 2025).
- [24] *Language Support — Zephyr Project Documentation*. <https://docs.zephyrproject.org/latest/develop/lang> (Besucht am 29. 08. 2025).
- [25] *LoRaWAN Architecture*. <https://www.thethingsnetwork.org/docs/lorawan/architecture/>. (Besucht am 01. 09. 2025).
- [26] *LoRaWAN for Smart Industry*. (Besucht am 28. 08. 2025).
- [27] Amro Lulu und Bijan Mobasser. „Dual-Use Chirp Spread Spectrum Waveform with Ranging Capability“. In: Apr. 2019, S. 1–6. DOI: 10.1109/RADAR.2019.8835752.
- [28] Kais Mekki u. a. „Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT“. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. März 2018, S. 197–202. DOI: 10.1109/PERCOMW.2018.8480255. (Besucht am 31. 08. 2025).
- [29] *MPU6000 Datasheet*. (Besucht am 31. 08. 2025).
- [30] N. Sornin (Semtech), M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM) und 7 O.Hersent (Actility). *LoRa Specification*. Jan. 2015. (Besucht am 27. 08. 2025).
- [31] *NetID and DevAddr Prefix Assignments*. <https://www.thethingsnetwork.org/docs/lorawan/prefix-assignments/>. (Besucht am 27. 08. 2025).
- [32] *Packet Broker*. <https://www.thethingsindustries.com/docs/concepts/packet-broker/>. (Besucht am 27. 08. 2025).
- [33] John G. Proakis und Masoud Salehi. *Digital Communications*. 5th ed. Boston: McGraw-Hill, 2008. ISBN: 978-0-07-295716-7.

- [34] Usman Raza, Parag Kulkarni und Mahesh Sooriyabandara. „Low Power Wide Area Networks: An Overview“. In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), S. 855–873. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2652320. (Besucht am 31.08.2025).
- [35] Rhode & Schwarz. *Characterization of LoRa Devices*.
- [36] *RSSI and SNR*. <https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/>. (Besucht am 06.08.2025).
- [37] Francisco Helder C. dos Santos Filho u. a. „Performance of LoRaWAN for Handling Telemetry and Alarm Messages in Industrial Applications“. In: *Sensors* 20.11 (Jan. 2020), S. 3061. ISSN: 1424-8220. DOI: 10.3390/s20113061. (Besucht am 28.08.2025).
- [38] *Senet Portal Docs*. <https://docs.senetco.io/ap/extCoverageServices/>. (Besucht am 27.08.2025).
- [39] *Signal-to-Noise Ratio (SNR) and Wireless Signal Strength*. [https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_\(SNR\)_and_Wireless_Signal_Strength](https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_(SNR)_and_Wireless_Signal_Strength). Okt. 2020. (Besucht am 07.08.2025).
- [40] *STM32WL Series - STMicroelectronics*. <https://www.st.com/en/microcontrollers-microprocessors/stm32wl-series.html>. (Besucht am 31.08.2025).
- [41] *SX1276 Datasheet*. <https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZU>. (Besucht am 31.08.2025).
- [42] Dinesh Tamang u. a. *Designing a Reliable and Low-Latency LoRaWAN Solution for Environmental Monitoring in Factories at Major Accident Risk*. Feb. 2022. DOI: 10.48550/arXiv.2202.08742. (Besucht am 28.08.2025).
- [43] TEKTELIC. *LoRaWAN Asset Tracking Explained with Cases*. Feb. 2023. (Besucht am 31.08.2025).
- [44] *TTN Mapper Documentation*. <https://docs.ttnmapper.org/>. (Besucht am 27.08.2025).
- [45] Tomas Tulka. *LoRa Spreading Factor Explained by Tomas Tulka*. <https://blog.ttulka.com/loras-spreading-factor-explained>. (Besucht am 07.08.2025).
- [46] *Ublox NEO6 Datasheet*. (Besucht am 31.08.2025).
- [47] *Ultimate Guide to LoRaWAN: How It Works, Benefits, and Real-World Applications | Cloud Studio IoT*. Apr. 2025. (Besucht am 31.08.2025).
- [48] Lorenzo Vangelista. „Frequency Shift Chirp Modulation: The LoRa Modulation“. In: *IEEE Signal Processing Letters* 24.12 (2017), S. 1818–1821. DOI: 10.1109/LSP.2017.2762960.
- [49] Lorenzo Vangelista, Andrea Zanella und Michele Zorzi. „Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios“. In: *IEEE Wireless Communications* 23 (Okt. 2015). DOI: 10.1109/MWC.2016.7721743.
- [50] *Why LoRaWAN Is the Best Choice for Long-Range, Low-Power IoT Tracking - Lansitec*. <https://www.lansitec.com/blogs/lorawan-the-superior-connectivity-option-for-long-range-low-power-tracking/>. Okt. 2024. (Besucht am 31.08.2025).

- [51] *Wifi Lora32 Datasheet*.
- [52] *Wio-E5 Mini | Seeed Studio Wiki*. https://wiki.seeedstudio.com/LoRa_E5_mini/. (Besucht am 31.08.2025).
- [53] Rodger E. Ziemer und William H. Tranter. *Principles of Communications: Systems, Modulation, and Noise*. Seventh edition. Danvers, MA: Wiley, 2015. ISBN: 978-1-118-07891-4.

Abbildungsverzeichnis

1.	Frequency Shift Keying Quelle: [18]	4
2.	LoRa und LoRaWAN im OSI Schichtenmodell Quelle: ??	5
3.	LoRa Chirps Quelle: [45]	6
4.	LoRa Symbole Quelle: [27]	7
5.	LoRaWAN Architektur Quelle: [25]	9
6.	Systemarchitektur Lora E5 mini	17
7.	Systemarchitektur Heltec LoRa V2	17
8.	Datensegment des LoraWAN Pakets	19

Tabellenverzeichnis

1.	Stromaufnahme der Komponenten.	21
----	--	----

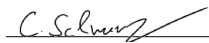
Quellcodeverzeichnis

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe.

Für die verbesserung von Gramatik und Wortwahl wurde ChatGPT verwendet.

München, den 2. September 2025

A handwritten signature in black ink, appearing to read 'C. Schwarz', is written over a horizontal line.

Christoph Schwarz