
Diplomarbeit

Herr
Ronald Creutz

**Entwurf und Implementierung
eines IoT Systems zur Infor-
mationsgewinnung in der
Agrarwirtschaft (hier: Imkerei)
unter Nutzung von LoRaWan.**

Mittweida, 2019

Diplomarbeit

Entwurf und Implementierung eines IoT Systems zur Informationsgewinnung in der Agrarwirtschaft (hier: Imkerei) unter Nutzung von LoRaWan.

Autor:
Herr

Ronald Creutz

Studiengang:
Technische Informatik

Seminargruppe:
TI15w1-f

Erstprüfer:
Prof. Dr. Ing. Olaf Hagenbruch

Zweitprüfer:
Dr. rer. pol. Uwe Wieland

Einreichung:
Mittweida, 20.10.2019

Verteidigung/Bewertung:
Mittweida, 2019

Faculty of applied bio- and computersciences

Diploma THESIS

**Design and implementation of an IoT
system for information gathering in
agricultural economics
using LoRaWan.**

author:

Mr.

Ronald Creutz

course of studies:

Technische Informatik

seminar group:

TI15w1-f

first examiner:

Prof. Dr. Ing. Olaf Hagenbruch

second examiner:

Dr. rer. pol. Uwe Wieland

submission:

Mittweida, 20.10.2019

defence/ evaluation:

Mittweida, 2019

Bibliografische Beschreibung:

Creutz, Ronald:

Entwurf und Implementierung eines IoT Systems zur Informationsgewinnung in der Agrarwirtschaft (hier: Imkerei) unter Nutzung von LoRaWan. - 2019. - XI, 89, XII S.

Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Diplomarbeit, 2019

Referat:

In der vorliegenden Arbeit werden geeignete Kriterien zur Beurteilung von Bienenvölkern in der Bienenhaltung (Imkerei) ermittelt. Für die datentechnische Erfassung dieser Kriterien wird prototypisch ein Internet of Things (kurz: IoT) System konzeptioniert und implementiert. Dieses System nutzt zur drahtlosen Datenübertragung die Technologie LoRa/LoRaWan. Eine Darstellung der Informationen erfolgt abschließend in einer IoT Plattform. Für diese Gesamtlösung werden dazu geeignete Hard- und Softwarelösungen verglichen und prototypisch implementiert. Eine Inbetriebnahme erfolgt an einem Referenzbienenstock.

Danksagung

Mein Dank geht an Herrn Prof. Dr. Ing. Olaf Hagenbruch für die Betreuung seitens der Hochschule Mittweida während des Studiums und der Diplomarbeit.

Weiterhin möchte ich mich bei Herrn Dr. Uwe Wieland für die fachliche Betreuung während der Erstellung der Diplomarbeit bedanken. Er hat eigentlich keine freie Kapazität und war trotzdem immer erreichbar.

Für die Unterstützung bei der Auswahl einer geeigneten Außenantenne bedanke ich mich bei Herrn Hennig von der Firma FTS Hennig GmbH.

Abschließend möchte ich mich bei meiner Familie bedanken. Sie hat mir den zeitlichen Freiraum geschaffen, um das Diplomthema zu bearbeiten.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	IX
1 Übersicht.....	1
1.1 <i>Motivation.....</i>	1
1.2 <i>Zielsetzung.....</i>	1
1.3 <i>Kapitelübersicht.....</i>	2
2 Grundlagen	3
2.1 <i>Bienen.....</i>	3
2.1.1 <i>Bienenvolk.....</i>	3
2.1.2 <i>Bedeutung der Bienen.....</i>	4
2.1.3 <i>Bienenhaltung</i>	5
2.2 <i>LoRa</i>	8
2.2.1 <i>Einführung in Modulationsverfahren</i>	8
2.2.2 <i>LoRa Modulationsverfahren</i>	10
2.2.3 <i>LoRa Packet Struktur</i>	15
2.2.4 <i>LoRa Modems.....</i>	18
2.3 <i>LoRaWan</i>	18
2.3.1 <i>Netzwerk Architektur - Überblick</i>	19
2.3.2 <i>Netzwerk Architektur – Erweitert</i>	20
2.3.3 <i>Roaming.....</i>	23
2.3.4 <i>Profile.....</i>	26
2.3.5 <i>End-Device Class A, B, C.....</i>	27
2.3.6 <i>Übertragung und Sicherheit von Schlüsseln</i>	28
2.3.7 <i>Nachrichten und Nutzdaten</i>	28
2.3.8 <i>Regionale Parameter</i>	31
3 Präzisierung der Aufgabenstellung	32
3.1 <i>Voruntersuchung.....</i>	32

3.2	<i>Hauptuntersuchung</i>	33
3.2.1	Ist-Erhebung	34
3.2.2	Ist-Analyse	36
3.2.2.1	Standort Bienen	36
3.2.2.2	Standort des LoRa Gateways	37
3.2.2.3	Bienenbeute.....	37
3.2.2.4	Chemikalien in der Beute	38
3.2.2.5	Ergebnis der Ist-Analyse	38
3.2.3	Soll-Konzeption / Grobentwurf	38
3.2.3.1	Hardware LoRaWan Node	39
3.2.3.2	Hardware LoRaWan Gateway.....	40
3.2.3.3	Sensoren zur Temperatur und Luftfeuchtemessung.....	42
3.2.3.4	Waage zur Ermittlung der Gesamtmasse einer Bienenbeute	44
3.2.3.5	Konkretisierung Sendeintervall / Visualisierung der Messergebnisse	47
3.2.3.6	Ergebnis der Sollkonzeption / Grobentwurf	48
4	Umsetzung	50
4.1	<i>Plattformwaage</i>	50
4.1.1	Versuchsaufbau	50
4.1.2	Mechanische Umsetzung	52
4.1.3	Kalibrierung.....	54
4.2	<i>Sensoreinheit</i>	56
4.2.1	Versuchsaufbau	56
4.2.2	Mechanische Umsetzung	57
4.2.3	Auszug Softwareimplementierung.....	58
4.3	<i>LoRaWan Node</i>	60
4.3.1	Versuchsaufbau	60
4.3.2	Mechanische Umsetzung	61
4.3.3	Auszug Softwareimplementierung.....	63
4.3.4	Programmablauf	64
4.4	<i>LoRaWan Gateway</i>	66
4.4.1	Versuchsaufbau	66
4.4.2	Mechanische Umsetzung	68
4.4.3	Konfiguration am LoRaWan Gateway	70
4.5	<i>IoT Plattform</i>	76
4.5.1	Überblick zu ThingsBoard	76
4.5.2	Möglichkeiten zum Betrieb	77
4.5.3	Auszug Konfiguration	77
4.6	<i>Inbetriebnahme Gesamtsystem</i>	82
4.6.1	Testaufbau.....	82
4.6.2	Referenzbienenstock	84

4.7	<i>Software-Werkzeuge in der Übersicht</i>	86
5	Zusammenfassung	88
5.1	<i>Erkenntnisse</i>	88
5.2	<i>Ausblick</i>	89
Literatur		90
Anlagen		97
Anlagen, Teil 1		I
Selbstständigkeitserklärung		13

Abbildungsverzeichnis

Abbildung 1 - Aufbau Magazinbeute	6
Abbildung 2 - Magazin mit Rähmchen (Warm- und Kaltbau).....	7
Abbildung 3 - Amplitudenmodulation.....	8
Abbildung 4 - Frequenzmodulation	9
Abbildung 5 - Frequency Shift Keying (FSK)	10
Abbildung 6 - Chirp Impuls (Spread Spectrum)	11
Abbildung 7 - LoRa Modulation	11
Abbildung 8 - signal-ratio-to-noise (SNR).....	14
Abbildung 9 - LoRa Paket Strukturen	15
Abbildung 10 - CRC im explicit header.....	16
Abbildung 11 - CRC im implicit header.....	16
Abbildung 12 - LoRa Modem von Semtech (SX1276/77/78/79)	18
Abbildung 13 - LoRaWan Netzwerk Architektur	19
Abbildung 14 - LoRaWan Netzwerk Architektur (erweitert).....	20
Abbildung 15 - Typen End-Devices (ABP, OTA)	21
Abbildung 16 - Passive Roaming	24
Abbildung 17 - Handover Roaming	25
Abbildung 18 – OTA Roaming Activation Procedure (allgemein).....	26
Abbildung 19 - Join Request über http	28
Abbildung 20 - Nachrichten und deren Inhalt	30

Abbildungsverzeichnis	V
Abbildung 21 - Bienenstandort	36
Abbildung 22 - Schaltung Wheatstone-Messbrücke	46
Abbildung 23 - Schema Plattformwaage für ein Bienenvolk.....	47
Abbildung 24 - Wireframe Dashboard Bienenvölker - aktueller Überblick	48
Abbildung 25 - Grobentwurf des Gesamtsystems.....	49
Abbildung 26 - Zeichnung Bosche Wägezelle H30A.....	50
Abbildung 27 - Versuchsaufbau Steckbrett Waage	51
Abbildung 28 - Unter- und Oberteil Plattformwaage.....	53
Abbildung 29 - Ablauf Ermitteln Kalibrierkoeffizient	55
Abbildung 30 - Kalibrierkoeffizient ermitteln.....	55
Abbildung 31 - Versuchsaufbau Bosch BME280	57
Abbildung 32 - Sensor im Gehäuse.....	58
Abbildung 33 - Ausgabe der Messwerte	60
Abbildung 34 - Versuchsaufbau LoRaWan Node	61
Abbildung 35 - LoRaWan Node mit Peripherie	62
Abbildung 36 - Node sendet Daten.....	64
Abbildung 37 - Programmablauf LoRaWan Node.....	65
Abbildung 38 - Versuchsaufbau Raspberry Pi und iC880A.....	68
Abbildung 39 - LoRaWan Gateway	68
Abbildung 40 - Installation Außenantenne	69
Abbildung 41 - Etcher erfolgreiche Installation.....	71
Abbildung 42 - erfolgreicher Login am Raspberry.....	72
Abbildung 43 - LoRa Gateway Konfiguration.....	73

Abbildung 44 - Konfiguration Packet-Forwarder.....	74
Abbildung 45 - Auszug global_conf.json	74
Abbildung 46 - LoRaServer Administrationswebsite.....	75
Abbildung 47 - Empfangene LoRaWan Live Daten vom LoRaWan Node	76
Abbildung 48 - ThingsBoard Geräteübersicht.....	78
Abbildung 49 - ThingBoard Gerät anlegen	78
Abbildung 50 - Konfiguration ThingsBoard Integrator im LoRaServer	79
Abbildung 51 - ThingsBoard Konfiguration Access Token im LoRaServer	79
Abbildung 52 - custom codec function LoRaServer.....	80
Abbildung 53 - Verarbeitung der Nutzdaten (LoRaServer)	81
Abbildung 54 - Beispiel Telemetrie Daten in ThingsBoard.....	81
Abbildung 55 – erstes Dashboard in ThingsBoard	82
Abbildung 56 - Gesamtsystem mit Kontrollpunkten	83
Abbildung 57 - Positionierung Innenraumsensor	84
Abbildung 58 - Waage mit Magazinbeute (Vorderansicht).....	85
Abbildung 59 - Waage mit Magazinbeute (Seitenansicht)	85
Abbildung 60 - Bienenvölker Überblick.....	86

Tabellenverzeichnis

Tabelle 1 - Formelzeichen Dauer Präambel	17
Tabelle 2 – Formelzeichen Payload Symbole.....	18
Tabelle 3 - Rollen Networkserver	22
Tabelle 4 - Schnittstellen LoRa Backend	23
Tabelle 5 - Eigenschaften Device Profile	27
Tabelle 6 - Nachrichtentypen.....	29
Tabelle 7 - Fragenkatalog.....	36
Tabelle 8 - Auswahl Boards.....	40
Tabelle 9 - Auswahl LoRaWan Gateways	42
Tabelle 10 - Auswahl Sensoren.....	44
Tabelle 11 - Auswahl Waagen.....	45
Tabelle 12 - Verbindung HX-711 und Half-Bridge-Load-Cells.....	52
Tabelle 13 - Verbindung HX-711 und Arduino MKR Wan 1300	52
Tabelle 14 - Stückliste Waage.....	54
Tabelle 15 - Stückliste Sensor	58
Tabelle 16 - Stückliste LoRaWan Node	63
Tabelle 17 - anwendungsspezifische Daten nach Cayenne Lpp.....	66
Tabelle 18 - Verbindung ic880A und Raspberry Pi3	67
Tabelle 19 - Stückliste LoRaWan Gateway	70
Tabelle 20 - Softwarewerkzeuge zur Konzeption / Implementierung	87

Abkürzungsverzeichnis

ABP	Activation-by-Personalization
ADC	Analog-Digital-Converter (Analog-Digital-Wandler)
ADR	Adaptive Data Rate
AM	Amplituden Modulation (Modulationsverfahren)
AWS	Amazon Webservices
CRC	cyclic redundancy check
CSS	chirp spread spectrum (Modulationsverfahren)
D.I.B.	Deutscher Imkerbund e.V
DHCP	Dynamic Host Configuration Protocol
DN	Deutsch Normalmaß (Rähmchenmaß)
FM	Frequenz Modulation (Modulationsverfahren)
FSK	Frequency Shift Keying (Modulationsverfahren)
HTTP	Hypertext Transfer Protocol (Hypertext Übertragungsprotokoll)
IC	Integrated Circuit
I²C	Inter-Integrated Circuit
IoT	Internet Of Things
IP	Adresse für das Internet Protokoll
IPSO	Internet Protocol for Smart Objects
JSON	JavaScript Object Notation

LPWan	low power wide area network
LoRa	long range
LoRaWan	long range wide area network
MAC	media access control layer
OPC-UA	open platform communication unified architecture
OTA	Over-the-air
OTAA	Over-the-air-activation
Rb	modulation bit rate bzw. Nominal modulation bit rate (Parameter für CSS bzw. LoRa Modulationsverfahren)
Rc	chip rate (Parameter für CSS Modulationsverfahren)
Rest	Representational State Transfer
Rs	symbol rate (Parameter für CSS Modulationsverfahren)
RS232	Recommended Standard 232 (für serielle Schnittstelle)
RSSI	Received Signal Strenght Indication
SF	Spreading Factor (Parameter für CSS Modulationsverfahren)
SNR	Signal-to-Noise Ratio (Rauschabstand)
SPI	Serial Peripheral Interface
SSH	Secure Shell
ToA	Time on Air
Ts	symbol periode (Parameter für CSS Modulationsverfahren)
TTN	The Things Network
UART	Universal Asynchronounous Receiver Transmitter

URL

Uniform Resource Locator

1 Übersicht

Dieses Kapitel beschreibt die Motivation für die Wahl des Themas und die Ziele. Weiterhin gibt es eine Kapitelübersicht der vorliegenden Diplomarbeit.

1.1 Motivation

„... der Imker muss seine Bienen beobachtend und betreuend durch den Jahresablauf führen, nur dann wird er Freude an ihnen haben.“¹

„Ihr Lebensrhythmus ist eng mit der Natur verbunden, und der Imker muss sie mit leichter Hand lenken, um zum Erfolg zu kommen. Es kann nicht heißen „der Bienen muss“ sondern man sollte stets fragen „Was braucht der Bienen?““²

Die zwei Zitate beschreiben sehr gut, dass Bienenhaltung auf Grund der Nähe zur Natur komplex ist, weil die Natur sich ständig verändert und kaum vorhersehbar ist. Um die richtigen Entscheidungen zur Führung eines Bienenvolks zu treffen, sind deshalb detaillierte Informationen sehr wichtig. Anhand dieser kann ein Imker bspw. Auskunft über die Vitalität eines Bienenvolkes, Schwarmneigung oder Eignung zur Honigernte ableiten. Konventionell werden diese Informationen durch aufwändige „Arbeit am Volk“ erhalten. Neben dem hohen Aufwand besitzt diese Vorgehensweise auch den Nachteil einer geringeren Informationsdichte der manuell aufgenommenen Daten sowie Störung des Bienenvolkes während dieser Tätigkeiten. Es besteht damit ein Risiko der Fehlinterpretation der vorliegenden Informationen und damit eine unzureichende bzw. falsche Handlungsempfehlung für den Imker.

1.2 Zielsetzung

In der vorliegenden Arbeit sollen geeignete Kriterien gefunden werden, welche zur Beurteilung von Bienenvölkern relevant sind. Grundlegende Vorgaben an ein praktisch umsetzbares und kostengünstiges Zielsystem zur Informationsgewinnung, Verarbeitung und Visualisierung sollen ermittelt werden.

¹ Gettert, Lieselotte (1998): Mein Bienenjahr – ein Arbeitskalender für den Imker, 2. Auflage, Ulmer, Stuttgart, Seite 11

² Gettert, Lieselotte (1998): Mein Bienenjahr – ein Arbeitskalender für den Imker, 2. Auflage, Ulmer, Stuttgart, Seite 12 f.

Long range (kurz: LoRa) bzw. long range wide area network (kurz: LoRaWan), als eine mögliche low power wide area network (kurz: LPWan) Technologie, bietet offenbar eine geeignete Grundlage für autarke Systeme zur Datenkommunikation ohne Nutzung externer Infrastruktur, wie Stromversorgungsnetz oder kosten- bzw. lizenzpflichtige Funknetze. Durch ein geeignetes Vorgehen sollen mögliche Soft- und Hardwaretechnologien, wie Bibliotheken, Mikrocontroller, Sensoren für einen Prototyp ermittelt werden. Der Fokus für die Auswahl der Komponenten soll neben Energieeffizienz vor allem bei geringen Kosten in der Umsetzung und im Betrieb liegen.

Das Gesamtsystem soll als Werkzeug für Imker bzw. Imkereibetriebe prototypisch implementiert werden. Dabei soll die Informationsgewinnung, Datenübertragung unter Anwendung der LoRa bzw. LoRaWan Technologie und die Visualisierung in einer Internet of Things (kurz: IoT) Plattform gezeigt werden.

Die technische Machbarkeit sowie die Bedeutung für die Bienenhaltung und ähnliche Anwendungsfälle der Agrarwirtschaft sollen abschließend kurz bewertet werden.

1.3 Kapitelübersicht

Die vorliegende Arbeit gliedert sich in fünf Kapitel.

Im **Kapitel 2** erfolgt eine kompakte Einleitung zum Thema Bienen. Die Themen LoRa und LoRaWan werden ausführlicher eingeführt. Damit werden die nötigen Grundlagen geschaffen, um im **Kapitel 3** eine Präzisierung der Aufgabenstellung vorzunehmen. Über Vor- und Hauptuntersuchung wird dieses Kapitel mit dem Grobentwurf der Ziellösung abgeschlossen. Das **Kapitel 4** stellt die Umsetzung der Ziellösung vor. Dabei werden ausgewählte Auszüge aller Teilsysteme beschrieben. Die Inbetriebnahme des Gesamtsystems bildet den Abschluss des Kapitels. Eine Zusammenfassung mit einer kompakten Darstellung der Erkenntnisse sowie einem Ausblick auf die weitere Verwendung ist Bestandteil des abschließenden **Kapitel 5**.

2 Grundlagen

Dieses Kapitel dient als Einführung zu Themen, welche in den nachfolgenden Kapiteln näher bzw. in Bezug zur Aufgabenstellung betrachtet werden. Dazu erfolgt eine kompakte Einleitung in das Thema Bienen und eine etwas ausführlichere Vorstellung der LPWAN Technologie LoRa und LoRaWan.

2.1 Bienen

In diesem Kapitel werden kompakt und etwas vereinfacht ausgewählte Themen zu dem Lebewesen Biene, deren Bedeutung und Haltung dargestellt.

2.1.1 Bienenvolk

Ein Bienenvolk besteht aus einer Vielzahl von Einzellebewesen und bewohnt eine Behausung mit Waben. Es wird gesamtheitlich als Bien bezeichnet und betrachtet. Im Bien existieren drei Bienenwesen - die Königin, die Drohnen und die Arbeitsbienen. Dabei hat jedes Bienenwesen bestimmte Aufgaben. Die Königin (i.d.R. eine je Bien) legt nach ihrer Begattung durch die Drohnen ihr ganzes Leben lang (ca. 3 bis 4 Jahre) Eier und sorgt damit für Nachkommen. Die Drohnen (einige 100 je Bien) haben keine weiteren Aufgaben und werden durch die Arbeitsbienen mitversorgt und sterben entweder nach dem Begattungsakt oder am Ende des Sommers. Die Aufgaben der Arbeitsbienen (einige Zehntausende je Bien) ändern sich im Laufe ihrer Lebenszeit. Sie führen die Brutnestpflege durch, füttern Larven (erstes Entwicklungsstadium nach der Eiablage) sowie jüngere und ältere Bienen, beteiligen sich am Ausbau der Waben, sammeln Futter und vieles mehr. Die Lebenszeit einer Arbeitsbiene beträgt zur Trachtzeit im Frühjahr und Sommer ca. 4 Wochen und im Winter einige Monate (Bezeichnung als Winterbiene).

Die Entscheidung, welches Bienenwesen aus der Larve entsteht, liegt bei den Arbeitsbienen. Beispielsweise entsteht eine Königin, wenn die Larve ausschließlich mit Gelée Royal gefüttert wird.

Ein Bien hat einen Vermehrungstrieb. Dieser wird ausgelebt, indem durch die Arbeiterbienen neue Königinnen gezogen werden. Mögliche Gründe dafür sind beispielsweise, dass das der Bien zu groß geworden ist (Platzmangel) oder die Legeleistung der Königin abnimmt. Es entsteht ein Schwarm bestehend aus einer Königin und einer Vielzahl von Arbeitsbienen bzw. mehrere Nachschwärme.

Die Anzahl der Einzellebewesen variiert im Lauf eines Jahres. So ist sie im Sommer am höchsten, nimmt im Spätsommer (Trachtende) ab, bleibt im Winter konstant und steigt im Frühjahr wieder (vgl. GETTERT 1998, S. 13 ff.).

2.1.2 Bedeutung der Bienen

„Ohne Bienen sähe die Welt heute mit Sicherheit anders aus.“³

„Stirbt die Biene, hat der Mensch noch vier Jahre zu leben. Keine Bienen mehr, keine Bestäubung mehr, keine Pflanzen mehr, keine Tiere mehr, kein Mensch mehr.“ (Albert Einstein zugeschrieben)⁴

Diese Zitate zeigen, wie wichtig die Bienen für unsere Erde sind. Das älteste Bienenfossil ist ca. 90 Millionen Jahre alt. In dieser Zeit entwickelten Bäume und Sträucher, später Kräuter und Blumen. Die Vermehrung war damals unsicher, weil sie abhängig vom Wind war. Durch die Insekten wurde dieser Prozess wesentlich sicherer. Insbesondere die Bienen waren und sind ein gern gesehener Gast für die Pflanzen, weil sie, ohne die Blüte zu zerstören, die Vermehrungschancen der Pflanzen erhöhen. Deshalb haben sich Pflanzen und Bienen im Laufe der Zeit perfekt aufeinander abgestimmt. Neben der heute sehr bekannten Honigbiene, welche durch den Imker betreut wird, gibt es ca. 1500 Arten von Wildbienen (weltweit) mit zum Teil völlig anderem Aussehen. Sie gelten in unserer Zeit als sehr gefährdet, weil sie nicht auf die Monokulturen abgestimmt sind (vgl. ARMBRUSTER 2014, S. 11 ff.). Insgesamt wird davon ausgegangen, dass ca. zwei Drittel aller Tiere auf der Welt zu den Insekten gezählt werden. Davon werden in den nächsten Jahrzehnten über 40 Prozent aussterben (vgl. KAUPERT 2019, S. 22 f.).

Neben dieser fundamentalen Bedeutung für unsere Umwelt, gibt es auch eine wirtschaftliche Bedeutung. Beispielsweise werden die landschaftlichen Erträge im Pflanzen-, Garten-, Gemüse- und Obstbau gesteigert. Deutschland liegt mit einem Pro-Kopf-Verbrauch von ca. 1 kg Honig im Jahr weltweit an der Spitze. Das entspricht etwa 78.000 Tonnen Honig pro Jahr. In Deutschland werden mehr als ca. 700.000 Bienenvölkern gehalten, welche ca. 20 – 30 Prozent des Gesamtbedarfs decken. Der verbleibende Anteil wird importiert (vgl. BMEL 2019, S. 6 ff.).

³ Armbruster, Sabine (2014): Das Bienen-Praxisbuch, 1. Auflage, BLV Buchverlag, München, Seite 11

⁴ Bundesministerium für Ernährung und Landwirtschaft (BMEL): Bienen, Unverzichtbar für Natur und Erzeugung, (https://www.bmel.de/SharedDocs/Downloads/Broschueren/Bienen.pdf?__blob=publicationFile, verfügbar am 6.8.2019)

2.1.3 Bienenhaltung

Bis zum heutigen Überfluss an süßen Lebensmitteln, galt in der Menschheitsgeschichte der Verzerr von Süßem als Überlebensvorteil. Wissenschaftler führen dies darauf zurück, dass es in der Natur praktisch nichts Giftiges gibt, was süß schmeckt. Honig ist bekanntlich sehr süß und deshalb geht man davon aus, dass Menschen bereits in der Steinzeit den Honig für sich entdeckten. Seit etwa 12.000 Jahren vor Christus wurde gezielt Honig von wildlebenden Bienen gewonnen. Mit dem Sesshaft werden des Menschen, vor ca. 10.000 Jahren, war es nur ein logischer Schritt, die Bienen näher an der Wohnstätte zu halten. Seit ca. 7.000 Jahren vor Christus ist die systematische Bienenhaltung belegt. In der Hochkultur Ägyptens wurde die Biene nahezu Gottgleich verehrt. Das Produkt Honig galt neben Wasser, Feuer, Eisen usw. als elementares Bedürfnis des Menschen. Bei den Griechen, etwa 600 Jahre vor Christus, wurden Gesetze geschaffen, welche den räumlichen Abstand von Imkern auf mind. 300 Fuß festlegten. Auch bei anderen Völkern, z.B. den Germanen, den Slaven und den Kelten, konnte eine Bienenhaltung nachgewiesen werden. Zunächst wurden Bienen im Wald gehalten. In der Regel bewohnen Bienen Hohlräume in hohlen Baumstämmen oder Felsen. Der Mensch schaffte künstliche Höhlen, indem er entsprechende Hohlräume an starken Bäumen ausschlug. Die Honigernte in großen Höhen war anstrengend und gefährlich und die Bäume wurde ebenfalls stark geschädigt. Deshalb folgten Holzklotzbeuten, danach Bienenkörbe geflochten aus Ruten und Reisig. Diese künstlichen Behausungen, bezeichnet als Beute, wurde auf dem Boden gestellt, so dass die Ernte des Honigs und die Pflege des Bienenvolkes stark vereinfacht wurde. Die Bienen bauen naturgemäß ihre Waben fest an ihre Behausung. Damit ist die Entnahme der Waben zur Honiggewinnung sehr schwierig. Nach dem Dreißigjährigen Krieg entwickelte sich deshalb der Bau von mobilen Waben. Damit wurde ein Entfernen der Waben möglich, ohne sie zerstören. In der konventionellen Imkerei werden heutzutage gedrahtete Rähmchen eingesetzt. Auf diesen Rähmchen wird eine Mittelwand aus Wachs durch Erhitzen der Drähte eingebracht. Ein Ausbau der Mittelwand zur Wabe erfolgt durch die Bienen (vgl. ARMBRUSTER 2014, S. 11 ff.).

Als Behausung für die Bienen werden heute oft sog. Hinterbehandlungs- oder Magazinbeuten eingesetzt. Beide haben spezielle Vor- und Nachteile, welche an dieser Stelle nicht betrachtet werden. Im Kontext der Aufgabenstellung handelt sich um Bienenvölker, welche in Magazinbeuten gehalten werden. Deshalb wird diese Variante etwas näher beschrieben (vgl. GETTERT 1998, S. 30 f.).

Eine Magazinbeute besteht in der konventionellen Imkerei aus einem Boden mit Fluglog und einem Deckel. Dazwischen befinden sich Magazine (auch als Zargen bezeichnet). Dabei gibt es ein oder mehrere Magazine, welche zusammengefasst als Brutraum bezeichnet werden. Zur Trachtzeit gibt es zusätzlich ein oder mehrere Magazine, welche als Honigraum bezeichnet werden. Zwischen dem Brut- und dem Honigraum wird ein Absperrgitter eingesetzt. Bedingt durch die Größe der Arbeitsbienen (sind kleiner), gelangen nur sie durch das Absperrgitter und lagern im Honigraum den Honig ab. Die Königin passt nicht durch das Absperrgitter und legt ihre Eier somit nur im Brutraum.

Folgende Abbildung zeigt den Aufbau zur Trachtzeit (links) und außerhalb der Trachtzeit (rechts).

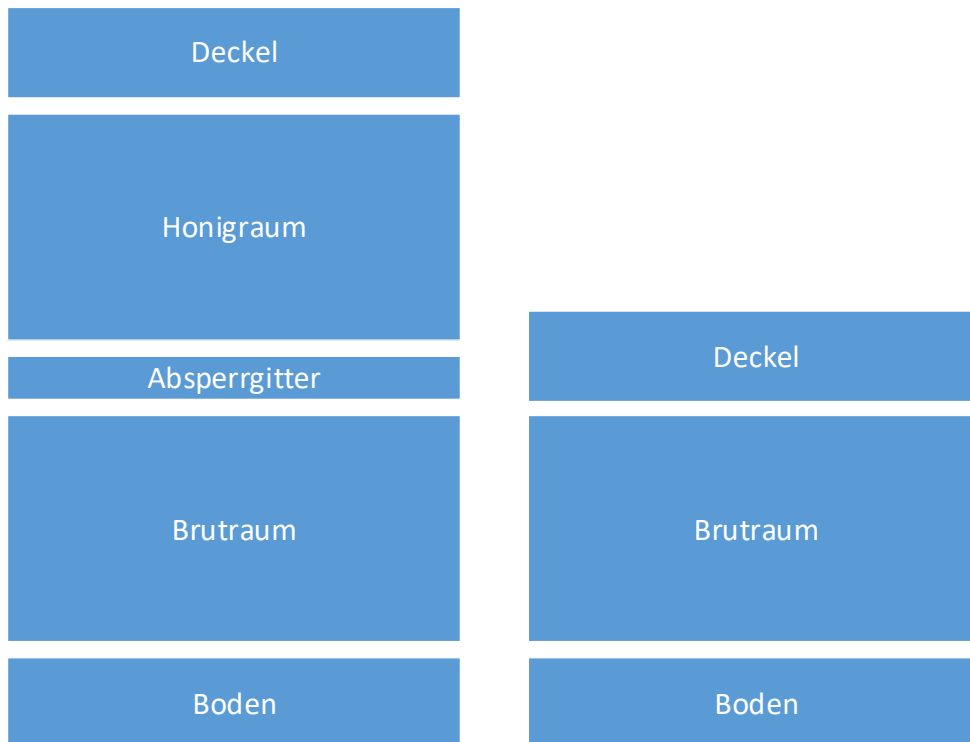


Abbildung 1 - Aufbau Magazinbeute

Es gibt eine Vielzahl von Varianten an Magazinbeuten. Sie lassen sich u.a. anhand ihrer Abmessungen unterscheiden. Dabei gibt es Standardformate in Abhängigkeit zum Format der verwendeten Rähmchen. In Deutschland sind beispielweise Deutsch Normal (DN), Zander oder Dadant gebräuchlich. Anhand der Anordnung der Waben zum Flugloch wird zwischen Warm- und Kaltbau unterschieden. Der Abstand zwischen den Waben wird als Wabengasse bezeichnet. Damit haben die Bienen genügend Freiraum. Folgende Abbildung zeigt ein Magazin, befüllt mit elf Waben im Warmbau (links) und Kaltbau (rechts) in der Draufsicht.

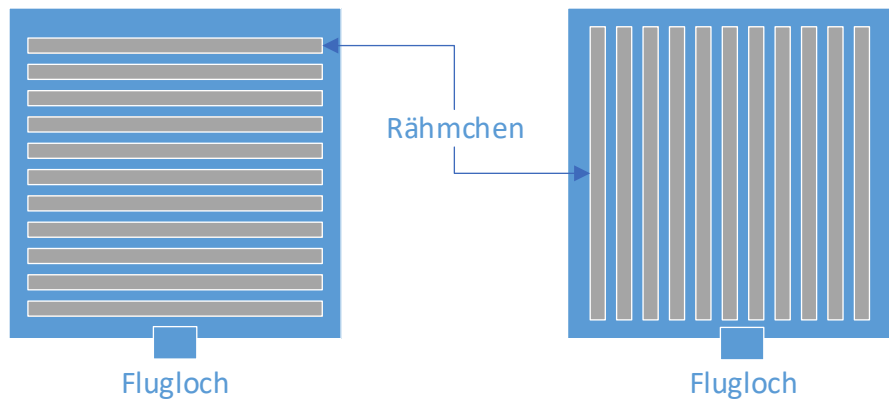


Abbildung 2 - Magazin mit Rähmchen (Warm- und Kaltbau)

Eine weitere Unterscheidung ist anhand des Werkstoffes möglich, aus denen die Beute gefertigt wurde. Im Handel sind Beuten aus Holz oder Styropor erhältlich.

Neben der Aufzucht von Königinnen, Bilden von Ablegern (Schaffen neuer Bienenvölker), Erweitern, Ernten, Einfüttern von Bienenvölkern gehört in Europa auch die Behandlung gegen die Varroa Milbe zur Aufgabe eines Imkers. Diese Milbe stammt aus Asien und hat sich innerhalb weniger Jahre zum gefährlichsten Parasiten unserer Honigbiene entwickelt. Sie schwächt das Bienenvolk, indem sie die Brut und erwachsene Bienen gleichermaßen befällt. Da die europäische Honigbiene an diesen Parasiten nicht angepasst ist, würde ein Bienenvolk ohne die Unterstützung des Imkers verenden. Es existiert eine überschaubare Anzahl von Behandlungsmethoden gegen die Varroa Milbe. Eine Möglichkeit ist die Behandlung mit Ameisensäure. Diese wird beispielsweise über einen Verdunster, z.B. Nasenheider Verdunster, in das Bienenvolk nach der Tracht eingebracht. Weitere Behandlungsmethoden verwenden Thymovol oder Bayvarol. Diese Behandlung wirkt für erwachsene Bienen und die Brut. Zusätzlich wird bei brutfreien Bienenvölkern eine Behandlung mit Milchsäure oder Oxalsäure durchgeführt. Dabei werden die Waben entnommen und mit Milchsäure besprüht bzw. wird Oxalsäure auf die Bienen, sitzend auf den Wabengassen, geträufelt. Eine Verbreitung erfolgt durch die Bienen in der Beute (vgl. ARMBRUSTER 2014, S. 98 ff.).

Um Bienenvölker zu halten, ist die Wahl des Standortes wichtig. Dabei sind Kriterien wie gutes Trachtangebot, gute Pollenversorgung, Wasser in erreichbarer Nähe, Schutz vor Kälte und Wind sowie vor extremer Sonneneinstrahlung, die Entfernung zu anderen Bienenvölkern benachbarter Imker oder auch die Zugänglichkeit zum Bienenstandort zu beachten. Da Bienen wilde Lebewesen sind, sollten sie nicht in unmittelbarer Nähe zu Spielplätzen, Wanderwegen, Eisenbahnlinien oder ähnlichem aufgestellt werden. Der Standort kann fix oder auch variable (Wanderimker) sein. Wanderimker ziehen mit ihren Bienenvölkern in Abhängigkeit zur jeweiligen Blüte an den entsprechenden Standort.

In Deutschland war die Bienenhaltung um 1914 auf Ihren Höhepunkt mit etwa 158.000 registrierten Imkern im Deutschen Imkerbund. Heute beträgt die Zahl der Mitglieder ca. 120.700⁵.

2.2 LoRa

LoRa steht für long range (lange Distanz) und nutzt ein proprietäres Protokoll von dem Unternehmen Semtech (<https://semtech.com>). Es dient der drahtlosen, bidirektionalen Datenübermittlung zwischen Sender und Empfänger. Die Ziele von LoRa sind eine hohe Entfernung zwischen Sender und Empfänger bei niedrigem Energiebedarf zu erreichen (low power applications). LoRa besitzt eine geringe Bandbreite und ist lizenzfrei zu verwenden. Zusammengefasst bildet LoRa die physikalische Schicht zur Datenübertragung.

2.2.1 Einführung in Modulationsverfahren

Zur Übertragung von Informationen von Sender zu Empfänger wird ein Trägersignal benötigt. Das Trägersignal wird mit einem Informationssignal transformiert. Die Transformation ist je nach Modulationsverfahren unterschiedlich. Das transformierte Signal wird ausgehend vom Sender über ein Übertragungsmedium zum Empfänger übertragen und demoduliert (Rücktransformation). Es gibt eine Vielzahl von Modulationsverfahren.

Die folgende Abbildung zeigt die Amplitudenmodulation (kurz: AM). Dabei bleibt die generierte Frequenz des Trägers (hier Sinusfunktion) durch den Oszillator konstant und die Amplitude des AM Signals ändert sich in Abhängigkeit zum Informationssignal.

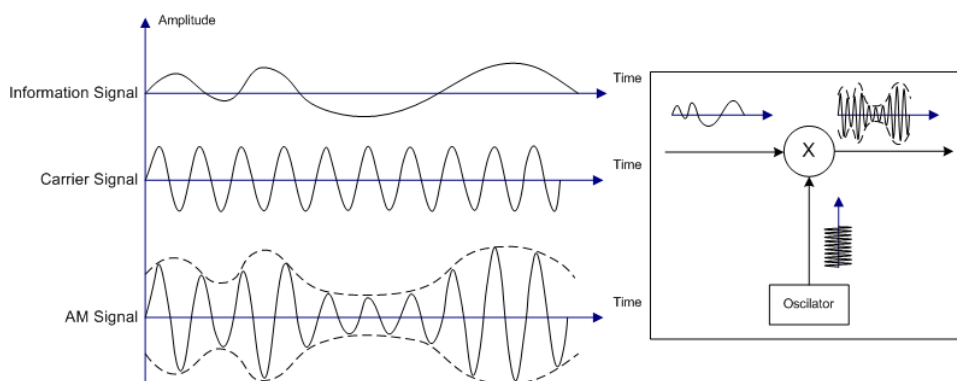


Abbildung 3 - Amplitudenmodulation

Unveränderte Übernahme: Abbildung 3: Illustration of amplitude modulation (https://en.wikipedia.org/wiki/Amplitude_modulation, verfügbar am 11.8.2019)

⁵ Deutscher Imkerbund e.V., Mitgliedsverbände des D.I.B. (https://deutscherimkerbund.de/171-Die_Imker_Landesverbaende, verfügbar am 9.8.2019)

Mit diesem Modulationsverfahren sind hohe Reichweiten möglich, aber die Störanfälligkeit (Rauschen) ist höher im Vergleich zu Frequenzmodulierten Signalen (kurz: FK). Letztere haben den Nachteil einer geringeren Reichweite. Folgende Abbildung zeigt das Trägersignal (Carrier Signal), das Informationssignal (Modulating Signal) und das frequenzmodulierte Signal (Frequency Modulation). Mit steigender Amplitude des Informationssignals steigt die Frequenz des modulierten Signals, bei sinkender Amplitude sinkt die Frequenz, bei einer Amplitude von 0 entspricht die Frequenz der des Trägersignals.

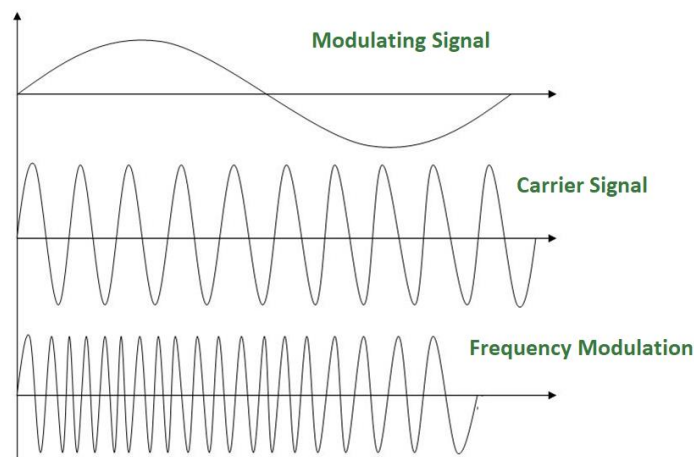


Abbildung 4 - Frequenzmodulation

Unveränderte Übernahme: Abbildung 4: frequency modulation

(<https://www.geeksforgeeks.org/digital-electronics-analog-analog-conversion-modulation/>, verfügbar am 11.8.2019)

Ein Modulationsverfahren für digitale Informationssignale, welches die Frequenzmodulation nutzt, ist das Frequency Shift Keying (kurz: FSK) Modulationsverfahren. Dabei wird der Zustand 0 und 1 jeweils einer Frequenz zugeordnet und mit dem Trägersignal (einer dritten Frequenz) gemischt.

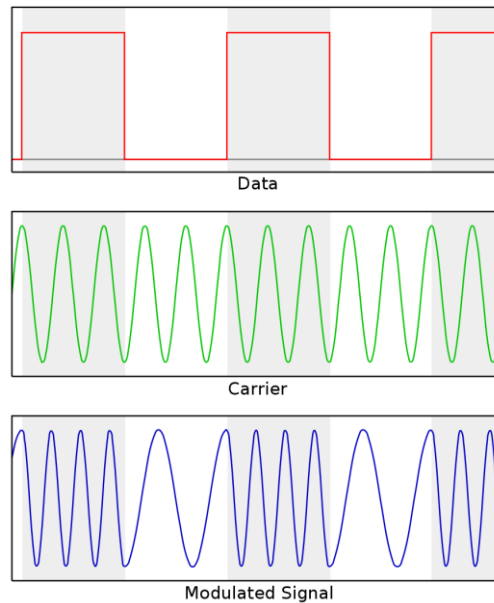


Abbildung 5 - Frequency Shift Keying (FSK)

Unveränderte Übernahme: Abbildung 5: Bildung eines binären FSK-Signals
(<https://de.wikipedia.org/wiki/Frequenzumtastung>, verfügbar am 11.8.2019)

Die Eigenschaften sind ein geringer Energiebedarf zur Datenübertragung sowie eine geringe Störanfälligkeit.

2.2.2 LoRa Modulationsverfahren

LoRa nutzt das Chirp Spread Spectrum (kurz: CSS) Modulationsverfahren, welches für Radaranwendungen in den 1940´er Jahren entwickelt wurde. Es bedarf einen geringen Energiebedarf und ist robust gegen Störeinflüsse. CSS wird den Frequenzspreizenden Modulationsverfahren zugeordnet (vgl. SENEVIRATNE 2019, S. 6 f.)). Dabei ändert der sogenannte Chirp Impuls seine Frequenz innerhalb einer festgelegten Zeit und eines festgelegten Frequenzbandes. Dieses Modulationsband (auch Bandbreite, kurz: BW) entspricht der zulässigen Bandbreite im regionalen Einsatzgebiet (z.B. für Europa 125 kHz oder 250 kHz). Ändert sich die Frequenz von unten nach oben wird dieser Prozess als up-chirp bezeichnet und umgekehrt als down-chirp. Die folgende Abbildung zeigt links die zunehmende Frequenz für einen up-chirp.

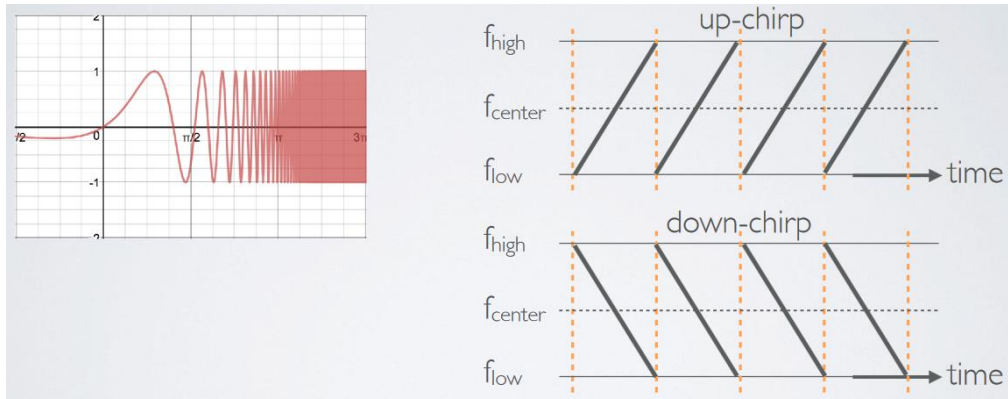


Abbildung 6 - Chirp Impuls (Spread Spectrum)

Unveränderte Übernahme: Abbildung 6: Spread Spectrum

(<https://lora.readthedocs.io/en/latest/#id27>, verfügbar am 11.8.2019)

In Abhängigkeit zum Spreading Factor (SF) wird die Information möglicherweise nicht vollständig in einem Schritt moduliert, sondern vereinzelt. Der zu modulierende Teil der Information wird im Folgenden als Symbol bezeichnet. Der SF beschreibt die Anzahl der Bits, um ein Symbol zu kodieren. Er kann einen ganzzahligen Wert zwischen 7 und 12 annehmen (vgl. SEMTECH-1 2015, S. 9 ff.).

Über den SF kann mit der folgenden Formel die Anzahl chips berechnet werden.

$$chips = 2^{SF}$$

Formel 1 - Berechnung chips

Als Beispiel wird ein SF 9 verwendet. Die Anzahl der chips ergibt sich damit aus Formel 1 mit 512. Damit sind für ein Symbol die Werte 0 bis 512 möglich.

Das Symbol wird dem Träger, dem Chirp Impuls, auf moduliert. Folgende Abbildung zeigt links den Chirp Impuls als Sequenz von up-chirp Impulsen und rechts beispielhaft ein moduliertes Signal. Die Symbolperiode (T_s) ist der jeweilige Abstand zwischen den orange Strichlinien.

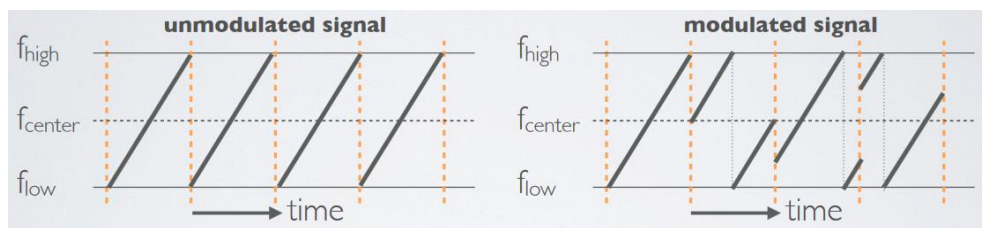


Abbildung 7 - LoRa Modulation

Unveränderte Übernahme: Abbildung 7: Spreading Factor (SF)

(<https://lora.readthedocs.io/en/latest/#id27>, verfügbar am 11.8.2019)

Beträgt die Anzahl der chips in dem vereinfachten Beispiel 512, werden die folgenden vier Symbole modulierte:

- Symbol 1: Startet bei 0 % von 512, somit ist Wert des Symbols 0
- Symbol 2: Startet bei 50 % von 512, somit ist Wert des Symbols 256
- Symbol 3: Startet bei 25 % von 512, somit ist Wert des Symbols 128
- Symbol 4: Startet bei 75 % von 512, somit ist Wert des Symbols 384

Ein Vorteil der Modulation durch den chirp Impuls ist, dass das Zeitverhalten und der Offset der Frequenz zwischen Sender und Empfänger gleich sind. Damit entspricht die Bandbreite des chirp Impuls der Spektral Bandbreite des Signals und reduziert die Komplexität zur Verarbeitung beim Empfänger. Folgend wird dieser Zusammenhang betrachtet (vgl. SEMTECH-1 2015, S. 8).

Es wird eine modulation bit rate (R_b) eingeführt, welche sich nach der folgenden Formel berechnet. Die Einheit wird in Bits je Sekunde angegeben (vgl. SEMTECH-1 2015, S. 9).

$$R_b = SF \cdot \frac{1}{\left(\frac{2^{SF}}{BW}\right)}$$

Formel 2 - modulation bit rate (allgemein)

Die Symbolperiode (symbol period, T_s) wurde in der **Error! Reference source not found.** dargestellt. Mit folgender Formel lässt sie berechnen. Die Einheit ist Sekunde (vgl. SEMTECH-1 2015, S. 10).

$$T_s = \frac{2^{SF}}{BW}$$

Formel 3 - symbol period

Demnach ist die symbol rate (R_s) das Reziproke der symbol period (T_s) und wird mit der Einheit Symbole je Sekunde angegeben.

$$R_s = \frac{1}{T_s} = \frac{BW}{2^{SF}}$$

Formel 4 - symbol rate

Als weitere Eigenschaft wird mit der chip rate (R_c) definiert, welche in der Einheit chips je Sekunde angegeben wird (vgl. SEMTECH-1 2015, S. 10).

$$R_c = R_s \cdot 2^{SF}$$

Formel 5 - chip rate (Allgemein)

Bedingt der LoRa Definition, dass ein chip je Sekunde per Hz Modulationsbreite gesendet wird, kann die Formel 5 - chip rate vereinfacht werden, indem die chips nicht berücksichtigt werden müssen (vgl. SEMTECH-1 2015, S. 10).

$$R_C = BW$$

Formel 6 - chip rate (LoRa)

Für ein Wertebeispiel werden die obigen Formeln angewandt. Dazu wird ein SF gleich 9 und eine Bandbreite (BW) für Europa mit 125 kHz vorgegeben.

Berechnung der modulation bit rate mit Formel 2:

$$R_b = 9 \cdot \frac{1}{\left(\frac{2^9}{125 \text{ kHz}}\right)} = 2197,27 \text{ bit/s}$$

Es folgt die Berechnung der symbol periode mit Formel 3.

$$T_s = \frac{2^9}{125 \text{ kHz}} = 0,004096 \text{ s}$$

Das entspricht nach der Formel 4.

$$R_s = \frac{1}{0,004096 \text{ s}} = 244,141 \text{ symbols/s}$$

Man erkennt die Auswirkung einer Änderung von SF. Verringert man den Parameter SF um 1, so verdoppelt sich die symbol rate.

Mit der Formel 5 berechnet sich die chip rate.

$$R_C = R_s \cdot 2^9 = 125.000 \text{ chips/s}$$

Bedingt durch die LoRa Definition liefert die Anwendung der Formel 6, dasselbe Ergebnis.

$$R_C = BW = 125.000 \text{ Hz} = 125.000 \text{ chips/s}$$

Weiterhin wird bei LoRa eine nominal modulation bit rate definiert, welche ein Fehlerkorrektur Schema enthält. Die code rate (CR) nimmt dabei einen ganzzahligen Wert von 1 bis 4 ein. Und gibt an, wieviel Bit zur Fehlerkorrektur übertragen werden. Damit lässt sich der rate code berechnen (vgl. SEMTECH-1 2015, S. 10).

$$\text{rate code} = \frac{4}{4 + CR}$$

Formel 7 - rate code

Die unter Formel 2 wird nun für LoRa angepasst (vgl. SEMTECH-1 2015, S. 10).

$$R_b = SF \cdot \frac{\text{rate code}}{\left(\frac{2^{SF}}{BW}\right)}$$

Formel 8 – nominal modulation bit rate (LoRa)

Als Wertebeispiel werden erneut die Parameter $BW = 125 \text{ kHz}$ sowie der $SF = 9$ verwendet. Zusätzlich wird der Parameter $CR = 2$ festgelegt.

Unter Benutzung der Formel 7 wird der rate code berechnet.

$$\text{rate code} = \frac{4}{4 + 2} = \frac{2}{3}$$

Der rate code wird mit den Parametern BW und SF in die Formel 8 eingesetzt.

$$R_b = 9 \cdot \frac{\frac{2}{3}}{\left(\frac{2^9}{125.000}\right)} = 1464,84 \text{ bit/s}$$

Im Vergleich zur modulation bit rate aus dem Beispiel ohne Fehlerkorrektur wurde diese mit 2197,27 Bits je Sekunde berechnet. Aus der Formel erkennt man weiterhin, dass ein steigender rate code, eine steigende Bandbreite oder ein steigender SF die nominal modulation bit rate reduziert.

Allerdings steigt mit abnehmender Datenrate auch die mögliche Time on Air (kurz: ToA), auch bezeichnet als message transmission time, und damit die Entfernung zwischen Sender und Empfänger (vgl. SEMTECH-2 2019, S. 3 ff.).

Ein Kriterium zur Beurteilung der Signalqualität ist der signal-ratio-to-noise (kurz: SNR, auch Rauschabstand). Es wird aus dem Verhältnis empfangener Signalpegel zu Störpegel angegeben. Ist das Verhältnis in einem kritischen Bereich, kann das Signal gestört werden und wird unbrauchbar. Folgende Abbildung zeigt den $SNR > 0$ (links) und $SNR < 0$ (rechts).

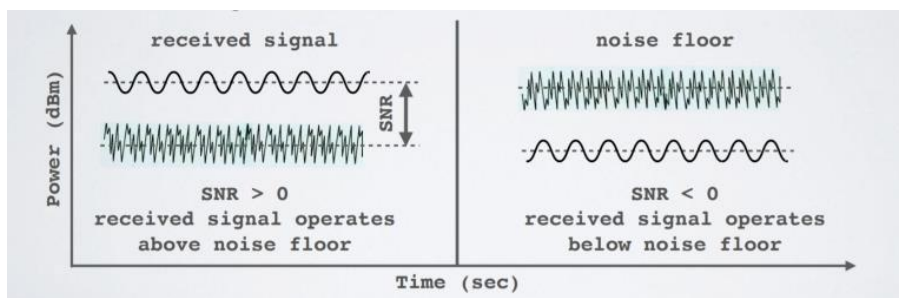


Abbildung 8 - signal-ratio-to-noise (SNR)

Unveränderte Übernahme: Abbildung 8: SNR (<https://www.eforengineer.com/rssi-snr-in-lora-wan/>, verfügbar am 11.8.2019)

Die Signalqualität wird u.a. durch die Parameter (siehe oben) beeinflusst. Somit entsteht für einen SF12 ein typischer SNR von -20 dB oder unter Verwendung eines SF10 ein typischer SNR von -15 dB (vgl. SEMTECH-2 2019, S. 3).

2.2.3 LoRa Paket Struktur

Im vorangegangenen Kapitel wurde das eingesetzte Modulationsverfahren bei LoRa beschrieben. Damit die eigentliche Nutzinformation vom Sender zum Empfänger zu übertragen werden, sind weitere Festlegungen notwendig. Beispielweise muss sichergestellt werden, dass die Information unversehrt übertragen werden (Manipulation durch Dritte, Fehler bei der Übertragung etc.). Beschrieben wird dies in einem Regelwerk oder Struktur, welche dem Sender und dem Empfänger bekannt sein müssen. Im Kontext von LoRa wird dies als LoRa Paket Struktur (LoRa packet structure, LoRa packet format) bezeichnet. Folgende Abbildung zeigt die zwei LoRa Paket Strukturen – als implizite und als explizit Variante (vgl. SEMTECH-3 2019, S. 29).

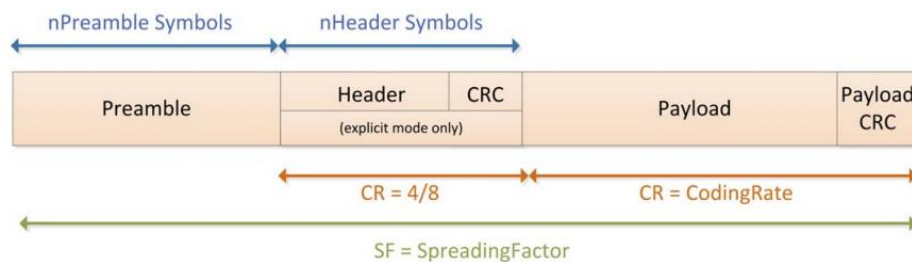


Abbildung 9 - LoRa Paket Strukturen

Unveränderte Übernahme: Abbildung 9: LoRa Packet Structure, S. 29

(https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, verfügbar am 11.8.2019)

Die **Präambel (preamble)** wird zur Synchronisierung der empfangenen Daten beim Empfänger benötigt. Dazu wird eine Anzahl von Symbolen beim Sender und Empfänger mit dem gleichen Wert definiert (vgl. SEMTECH-3 2019, S. 29). Beispielsweise wird in Europa (EU863-870) eine Präambel von 8 Symbolen verwendet.

Der **Header**, welcher explizit (explicit mode) angegeben wird, dient dazu unterschiedliche Konfigurationen für die Länge der Nutzdaten, CRC aktiviert bzw. deaktiviert sowie Fehlerkorrektur zu verwenden. Folgende Abbildung zeigt die Auswirkung, der Konfiguration für die cyclic redundancy check (kurz: CRC) im explicit mode (vgl. SEMTECH-3 2019, S. 29 f.).

Explicit Header	Transmitter	Receiver	CRC Status
Value of the bit RxPayloadCrcOn	0	0	CRC is not checked
	0	1	CRC is not checked
	1	0	CRC is checked
	1	1	CRC is checked

Abbildung 10 - CRC im explicit header

Unveränderte Übernahme: Abbildung 10: explicit header mode, S. 30 (https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, verfügbar am 11.8.2019)

Dagegen muss die CRC im implicit header mode vom Sender immer aktiviert werden, weil ansonsten die CRC als fehlerhaft interpretiert wird.

Implicit Header	Transmitter	Receiver	CRC Status
Value of the bit RxPayloadCrcOn	0	0	CRC is not checked
	0	1	CRC is always wrong
	1	0	CRC is not checked
	1	1	CRC is checked

Abbildung 11 - CRC im implicit header

Unveränderte Übernahme: **Error! Reference source not found.**: explicit header mode, S. 30 (https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, verfügbar am 11.8.2019)

Zusätzlich gibt es die Möglichkeit die Optimierung für die langsame Datenrate zu aktivieren (low data rate optimization). Diese erhöht die Robustheit der Information und wird bei einer Symboldauer (T_s) größer 16ms verwendet (vgl. SEMTECH-3 2019, S. 31).

Im **Payload** befinden sich die eigentlichen Informationen zzgl. der optional angehängenen **Payload CRC** (vgl. SEMTECH-3 2019, S. 31). Die Dauer zur Übertragung eines LoRa Pakets kann mit den folgenden Formeln berechnet werden (vgl. SEMTECH-3 2019, S. 31). Dabei wird ein bekannter SF vorausgesetzt (siehe Kapitel 2.2.2).

Die Gesamtdauer zur Übertragung eines LoRa Pakets ergibt sich aus der Addition der Dauer zur Übertragung der Präambel und des Payloads. Die Dauer wird in Sekunden angegeben.

$$T_{packet} = T_{preamble} + T_{payload}$$

Formel 9 – Paketdauer (time on air)

Zur weiteren Berechnung wird die Symbolzeit (T_S) aus dem Kapitel 2.2.2 benötigt.

$$T_{preamble} = (n_{preamble} + 4,25)T_S$$

Formel 10 - Dauer Präambel

Formelzeichen	Beschreibung
$n_{preamble}$	Anzahl der vereinbarten Symbole für die Präambel.
T_S	Symbolzeit (siehe Kapitel 2.2.2)

Tabelle 1 - Formelzeichen Dauer Präambel

$$n_{payload} = 8 + \max \left(\text{ceil} \left[\frac{(8PL - 4SF + 28 + 16CRC - 20IH)}{4(SF - 2DE)} \right] \cdot (CR + 4), 0 \right)$$

Formel 11 - Anzahl Payload Symbole

Folgende Tabelle beschreibt die verwendeten Formelzeichen.

Formelzeichen	Beschreibung
PL	Die verwendete Anzahl in Bytes des Payloads.
CRC	Ist die CRC Berechnung aktiviert, ist der Wert 1, ansonsten 0.
IH	Ist der implicit Header aktiviert, dann ist der Wert 1, ansonsten 0 (explicit header).
DE	Ist die Optimierung (low data rate optimization) aktiviert, dann ist der Wert 1, ansonsten 0.

CR	Coding rate (bits zur Fehlerkorrektur, siehe Kapitel 2.2.2)
----	---

Tabelle 2 – Formelzeichen Payload Symbole

Die Dauer zur Übertragung des Payloads kann nun ermittelt werden, indem Anzahl der Payload Symbole mit der Symbolzeit (T_s) multipliziert wird.

$$T_{payload} = n_{payload} \cdot T_s$$

Formel 12 - Dauer Payload

Die erforderliche Paketdauer (T_{packet}) wird beispielsweise in den USA reguliert. Damit kann die Paketdauer die maximal zulässige Kanalverweilzeit übersteigen. Deshalb wurde das sog. frequency hopping spread spectrum (FHSS) implementiert. Dabei wird das Gesamtpaket zerlegt und jeder Teil einem Sprungkanal zugeordnet. Zur Übertragung wird periodisch zwischen den Sprungkanälen umgeschaltet, so dass der entsprechende Teil des Gesamtpaketes übertragen wird (vgl. SEMTECH-3 2019, S. 32 f.).

2.2.4 LoRa Modems

Das proprietäre LoRa Protokoll ist am Markt in unterschiedlichen Varianten als integrated circuits (kurz: ICs) erhältlich. Diese unterscheiden sich u.a. im Frequenzbereich. Im Einsatzfall für das für das jeweilige Land beachtet werden. Folgende Abbildung zeigt die Chiptypen SX1276/77/78/79 mit ausgewählten Eigenschaften.

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

Abbildung 12 - LoRa Modem von Semtech (SX1276/77/78/79)

nveränderte Übernahme: Abbildung 12: SX1276/77/78/79 Device Variantes and Key Parameters, S. 10 (https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, verfügbar am 11.8.2019)

2.3 LoRaWan

LoRaWan ist eine offene Spezifikation der LoRaWan Alliance. Sie ist entstanden, um low power Geräte mit Netzwerken, z.B. dem Internet oder Unternehmensnetzwerke, zu verbinden. Dabei sind Features speziell für die Anforderungen von Anwendungen im IoT Kontext berücksichtigt, wie beispielsweise Mobilität, Sicherheit oder auch ein günstiger

Preis beim Betrieb. LoRaWan nutzt LoRa als physikalische Bit-Übertragungsschicht (vgl. LORAALLIANCE-1 2014, S. 3 ff.).

2.3.1 Netzwerk Architektur - Überblick

Die folgende Abbildung zeigt die LoRaWan Netzwerk Architektur beispielhaft.

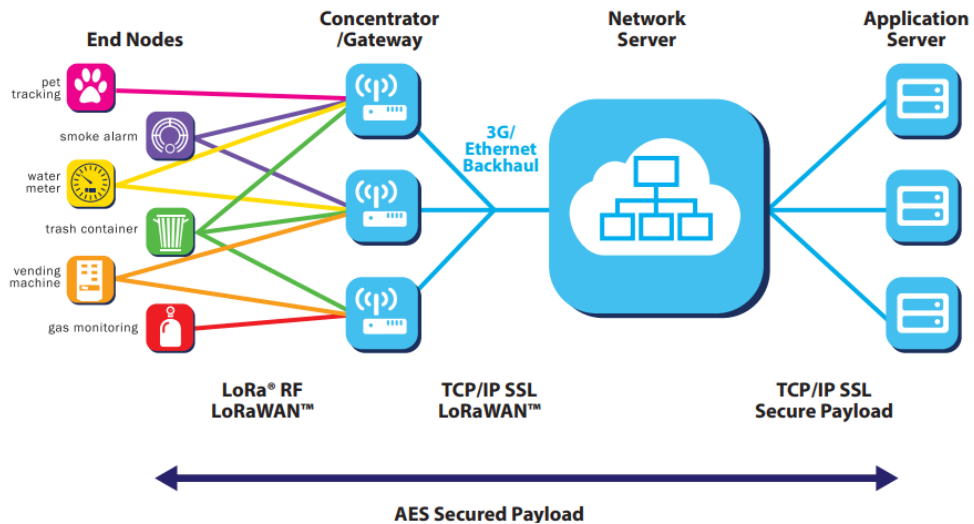


Abbildung 13 - LoRaWan Netzwerk Architektur

Unveränderte Übernahme: Abbildung 13: network Architecture, S. 8 (<https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>, verfügbar am 11.8.2019)

Die **End Nodes**, auch bezeichnet als End Devices, sind Sensoren oder Aktoren. Sie kommunizieren drahtlos mit einem oder mehreren Gateways, auch bezeichnet als Concentrators. Es besteht eine logische Verbindung zu einem oder mehreren Application Servern (vgl. LORAALLIANCE-2 2017, S. 8).

Das **Gateway** ist der physikalischen Schicht zugeordnet. Es verbindet die End Nodes und den Network Server. Dabei dekodiert es empfangene Nachrichten vom End Node und leitet diese an den Network Server weiter bzw. kodiert die Nachrichten vom Network Server und sendet diese zum End Node. Es erfolgt keine Interpretation der Pakete (vgl. LORAALLIANCE-2 2017, S. 9).

Der **Network Server** bildet das Zentrum in der LoRaWan Netzwerk Architektur. Die Aufgaben des Network Servers sind beispielsweise die Prüfung der Adresse des End Nodes, Frame Authentifizierung, Anpassung der Datenrate (Adaptive Data Rate), Verarbeitung der join-requests von End Nodes, Weiterleitung der empfangenen Pakete zu einem oder mehreren Application Servern (vgl. LORAALLIANCE-2 2017, S. 9).

Im **Application Server** werden die empfangenen Nutzdaten der Anwendungsschicht verarbeitet (vgl. LORAALLIANCE-2 2017, S. 11).

Weiterhin existieren ein oder mehrere **Join Server**. Diese sind mit einem oder mehreren Network Servern verbunden. Sie sind Teil des Aktivierungsprozesses für End Nodes.

2.3.2 Netzwerk Architektur – Erweitert

Einen Überblick zur Netzwerkarchitektur wurde im Kapitel 2.3.1 gegeben. In diesem Kapitel erfolgt aus Gründen der besseren Übersicht die Ergänzung ausgewählte Informationen.

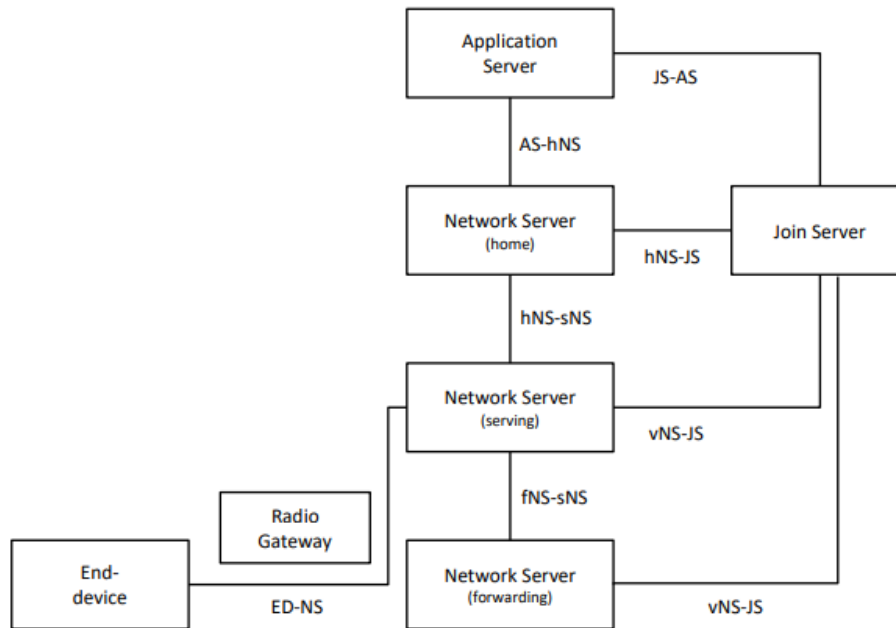


Abbildung 14 - LoRaWAN Netzwerk Architektur (erweitert)

Unveränderte Übernahme: Abbildung 14: LoRaWAN Network Reference Model (NRM), roaming End-Device, S. 8 (<https://lora-alliance.org/sites/default/files/2018-04/lorawantm-backend-interfaces-v1.0.pdf>, verfügbar am 31.8.2019)

End-Device (Nodes) können anhand der Integration im LoRaWAN Netzwerk in zwei Typen eingeteilt werden – Activation-by-Personalization (kurz: ABP) und Over-the-Air (kurz: OTA). Ein ABP Device wird direkt, d.h. ohne Anmeldeprozedur, an ein Netzwerk angebunden. Dazu müssen folgende Informationen bekannt sein: DevAddr, AppSKey, Network Session Keys (SNwkSIntKey, FNwkSIntKey, NwkSEncKey für R1.1 bzw. NwkSKey für R1.0/1.0.2 Endgeräte). Diese Informationen müssen im Endgerät gespeichert sein. Dem Network Server müssen die DevAddr und die Network Session Keys bekannt sein. Dem Application Server muss die DevAddr und AppSKey vorliegen. Ein OTA Device muss sich am Netzwerk anmelden, bevor als Teilnehmer registriert ist. Dazu müssen folgende Informationen bekannt sein: DevEUI, NwkKey (nur bei R1.1), AppKey, JoinEUI. Diese Informationen sind im End-Device gespeichert. Der Join Server besitzt ebenfalls diese Informationen des End-Device. Kein Network Server oder Application Server besitzt diese Informationen vor einer Registrierung des End-Device. Bei der Kommissionierung des

End-Device erfolgt eine Verknüpfung mit dem Home Network Server und einem ausgewählten Application Server. Der Join Server besitzt die Informationen des Home Network Server. Der Application Server hat Kenntnis über die DevEUI. Unter Umständen (abhängig von der Implementierung der Spezifikation) besitzt der Home Network Server zusätzliche Informationen. Nachdem die Registratur und die Aktivierung (Beitritt) erfolgreich waren, hat das End-Device Zugang zu den Informationen: DevAddr, Network Session Keys, AppSKey. Diese Informationen sind identisch zu einem ABP registrierten End-Device. Der Join Server kennt die DevEUI, DevAddr, Network Session Keys, AppSKey, DevNonce. Der Join Server übermittelt DevEUI und AppSKey zum Application Server sowie die Network Session Keys und evtl. AppSKey zum Network Server (vgl. LORAALLIANCE-2 2017, S. 12 f.).

Folgende Abbildung zeigt links ein ABP End-Device und rechts das OTA End-Device. Man erkennt leicht, dass bei dem ABP End-Device nur die Kommissionierung und beim OTA die Kommissionierung und Aktivierung (Join) erforderlich sind.

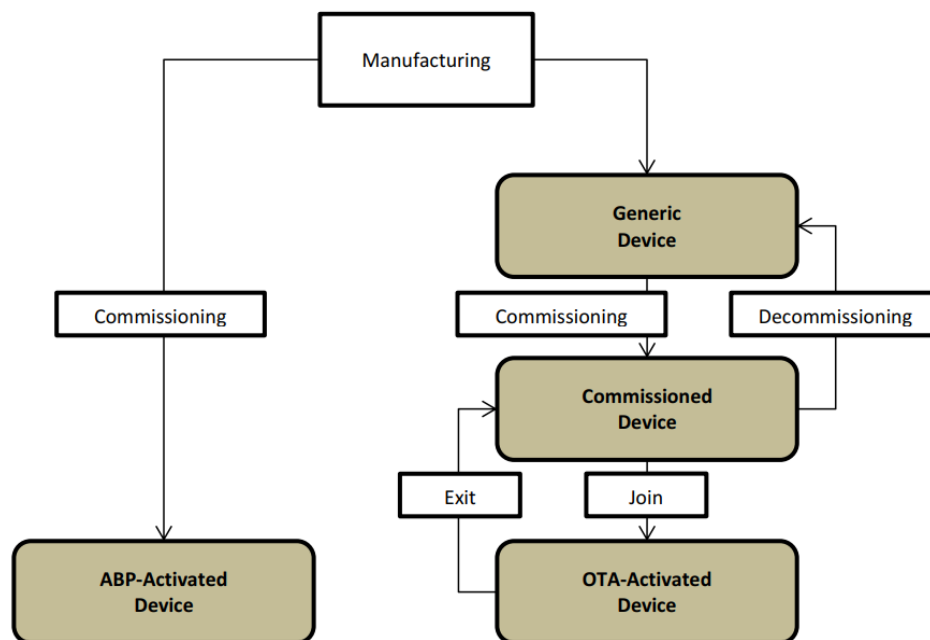


Abbildung 15 - Typen End-Devices (ABP, OTA)

Unveränderte Übernahme: Abbildung 15: End-Device types and states, S. 12 (<https://loralliance.org/sites/default/files/2018-04/lorawantm-backend-interfaces-v1.0.pdf>, verfügbar am 31.8.2019)

Eine De-Kommissionierung des End-Device erfolgt durch das Löschen der bereitgestellten Verbindungsinformationen zwischen End-Device, Home Network Server und Application Server sowie Join Server.

Die Abbildung 14 zeigt, dass ein **Network Server** in drei Rollen unterteilt werden, wobei ein Network Server alle oder auch nur eine Rolle im Gesamtsystem einnehmen kann.

Folgende Tabelle listet eine Kurzbeschreibung der Rollen auf. Die Kürzel an den Verbindungswegen in der **Error! Reference source not found.** zeigen, welche Schnittstellen der beteiligten Systeme genutzt werden (z.B. vNS-JS) (vgl. LORAALLIANCE-2 2017, S. 9 ff.).

Rolle	Beschreibung
Serving Network Server (sNS)	Steuert den Zugriff auf die MAC Schicht (die MAC Schicht befindet sich über der physikalischen Schicht)
Home Network Server (hNS)	In diesem Bereich wird das Device Profile, Service Profile, Routing Profile sowie die DevEUI vom End Node gespeichert. Es besteht eine direkte Verbindung zum Join Network Server und zum Application Server.
Forwarding Network Server (fNS)	Die Verwaltung der Gateways erfolgt in diesem Bereich. Es besteht eine Verbindung zum Serving Network Server.

Tabelle 3 - Rollen Networkserver

Der **Application Server** behandelt die Nutzdaten der End-Devices (bi-direktional). Zwischen Application Server und Network Server kann eine 1 zu m oder eine n zu 1 Verbindung existieren. Weiterhin kann ein Application Servern mit einem oder mehreren Join Servern verbunden sein. Der Home Network Server leitet eine Anfrage anhand der DevEUI zum zugeordneten Application Server (vgl. LORAALLIANCE-2 2017, S. 11).

Der **Join Server** dient zur Umsetzung des Aktivierungsprozess von OTA End-Device. Zwischen Join Server und Networkserver kann eine 1 zu m oder n zu 1 Verbindung bestehen. Der End Node schickt in seinem Join Request eine JoinEUI (früher als AppEUI bezeichnet). Anhand dieser JoinEUI findet der Join Server einen richtigen Network Server für den End Node. Damit wird der entsprechende Uplink (End Node sendet Daten) und Downlink (End Node empfängt Daten) bereitgestellt. Weiterhin wird dem End Node der Network Server Session Key bekanntgegeben. Der richtige Application Server erhält den Application Session Key. Zusammenfassend benötigt der Join Server Zugriff auf die folgenden Daten des End Nodes: DevEUI, AppKey, NwkKey (Network Key, ab LoRaWan 1.1 End Node), Home Network Server Identifier, Application Server Identifier, LoRaWan Version vom End Node, eine Logik um ein bevorzugtes Netzwerk auszuwählen (wenn das End Node mit mehreren Network- bzw. Application Server kommunizieren könnte). Dabei werden AppKey sowie der NwkKey nur auf dem Join Server gespeichert und gelangen niemals zum End Node oder zum Application Server. Die Verbindung zwischen den Join

Server und Network Server sollte durch Authentifizierung und Schutz vor Datenmanipulation Dritter abgesichert werden (vgl. LORAALLIANCE-2 2017, S. 9 ff.).

Zusammenfassend werden die Schnittstellen der Abbildung 14 beschrieben.

Schnittstelle	Beschreibung
hNS-JS	Dient zur Bearbeitung der Aktivierung der OTA End-Devices (Join Requests).
vNS-JS	Wird erforderlich, um die NetID des Home Network Servers zu ermitteln, welcher mit dem End-Device verbunden ist.
ED-NS	Dient der bidirektionalen Datenübertragung auf der MAC Schicht zwischen End-Device und Network Server.
AS-hNS	Wird genutzt vom Application Server und Network Server genutzt, um Nutzdaten zu übermitteln.
hNS-sNS / sNS-fNS	Dient dem Datenaustausch von Nutzinformationen zwischen Home Network Server und Network Server (Roaming Szenario).
AS-JS	Darüber wird Application Session Key von Join Server an den Application Server übertragen.

Tabelle 4 - Schnittstellen LoRa Backend

2.3.3 Roaming

Roaming ermöglicht die Nutzung von Diensten, welche von einem anderen Netzwerk bereitgestellt werden. Bzw. ermöglicht den Wechsel in ein anderes Netzwerk. Dabei sind im LoRaWan Standard zwei Typen spezifiziert – Passive Roaming und Handover Roaming.

Beim Passive Roaming empfängt ein Network Server Daten von einem End-Device. Dieser Network Server hat das End-Device aber nicht aktiviert, damit fehlt der Kontext zu diesem End-Device. Der Network Server versucht über die im Paket enthaltene NwkID einen zugeordneten Network Server zu finden (Mapping NwkID zu NetID). Nachdem der Server gefunden wurde, fungiert der erste Network Server als Forwarding Network Server, welcher die Anfragen an den zweiten Network Server weiterleitet. Vorausgesetzt wird dabei, dass die beteiligten Network Server und das End-Device passive Roaming unterstützen. Ist es an mindestens einer Stelle nicht aktiviert, wird das Paket verworfen. Passive Roaming wird für die Versionen R1.0 sowie R1.1 unterstützt (vgl. LORAALLIANCE-2 2017, S. 24 ff.). Folgende Abbildung zeigt den Ablauf vereinfacht.

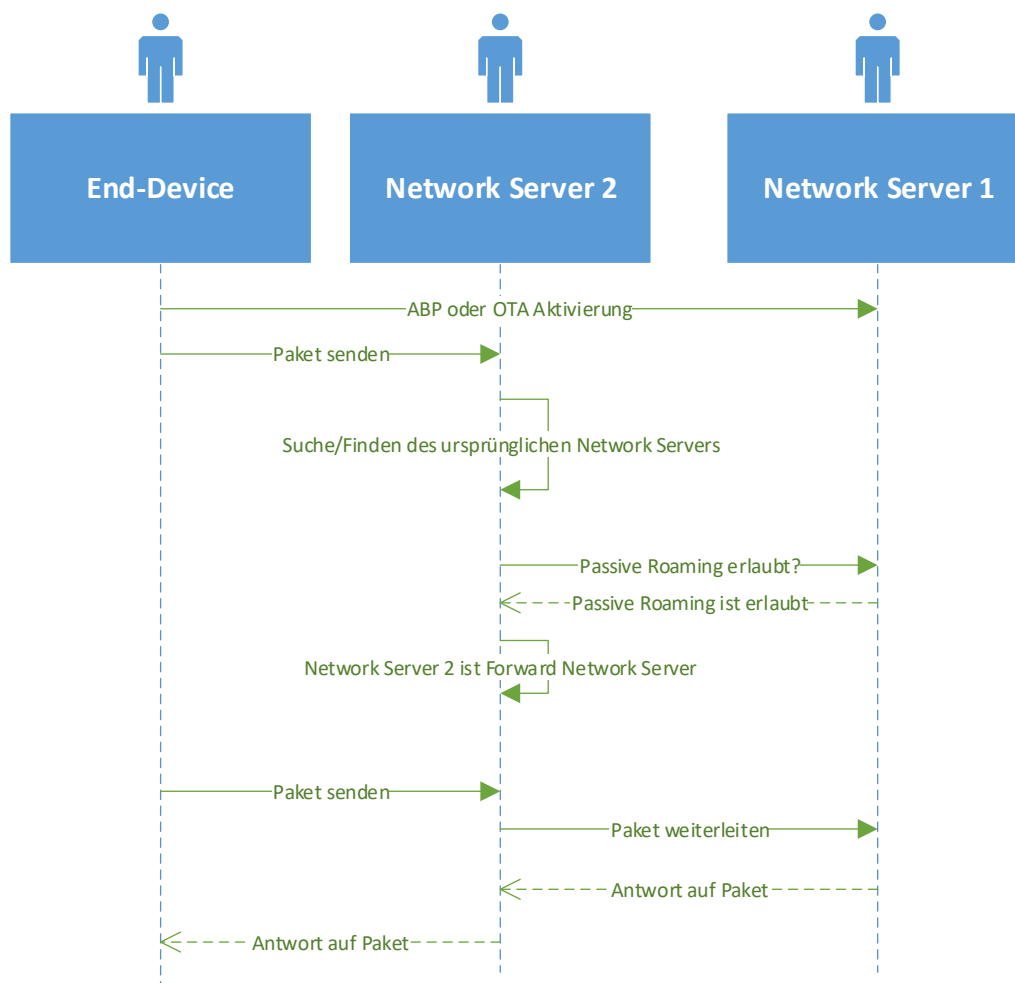


Abbildung 16 - Passive Roaming

Ein End-Device wurde zuvor von einem Network Server aktiviert. Das End-Device sendet einen Rejoin Request, welcher von einem anderen Network Server empfangen wird. Dieser Network Server versucht anhand der NetID (über Rejoin Request ermittelt) den Network Server in seiner Roaming policy Nachschlagetabelle zu finden und aufzulösen. Danach erfolgt der Datenaustausch (z.B. evtl. Device Profile und weitere Informationen) zwischen den Network Servern. Danach sendet das End-Device Pakete zu dem neuen Network Server. Dabei wird das erste Paket vom neuen Network Server an den alten Net-

work Server weitergeleitet, welcher daraufhin seinen Service für dieses End-Device stoppt. Vorausgesetzt wird dabei das Handover für die beteiligten Systeme aktiviert und richtig konfiguriert ist. Handover Roaming wird ab der Version R1.1 unterstützt (vgl. LO-RAALLIANCE-2 2017, S. 30 ff.). Folgende Abbildung zeigt den beschriebenen Ablauf.

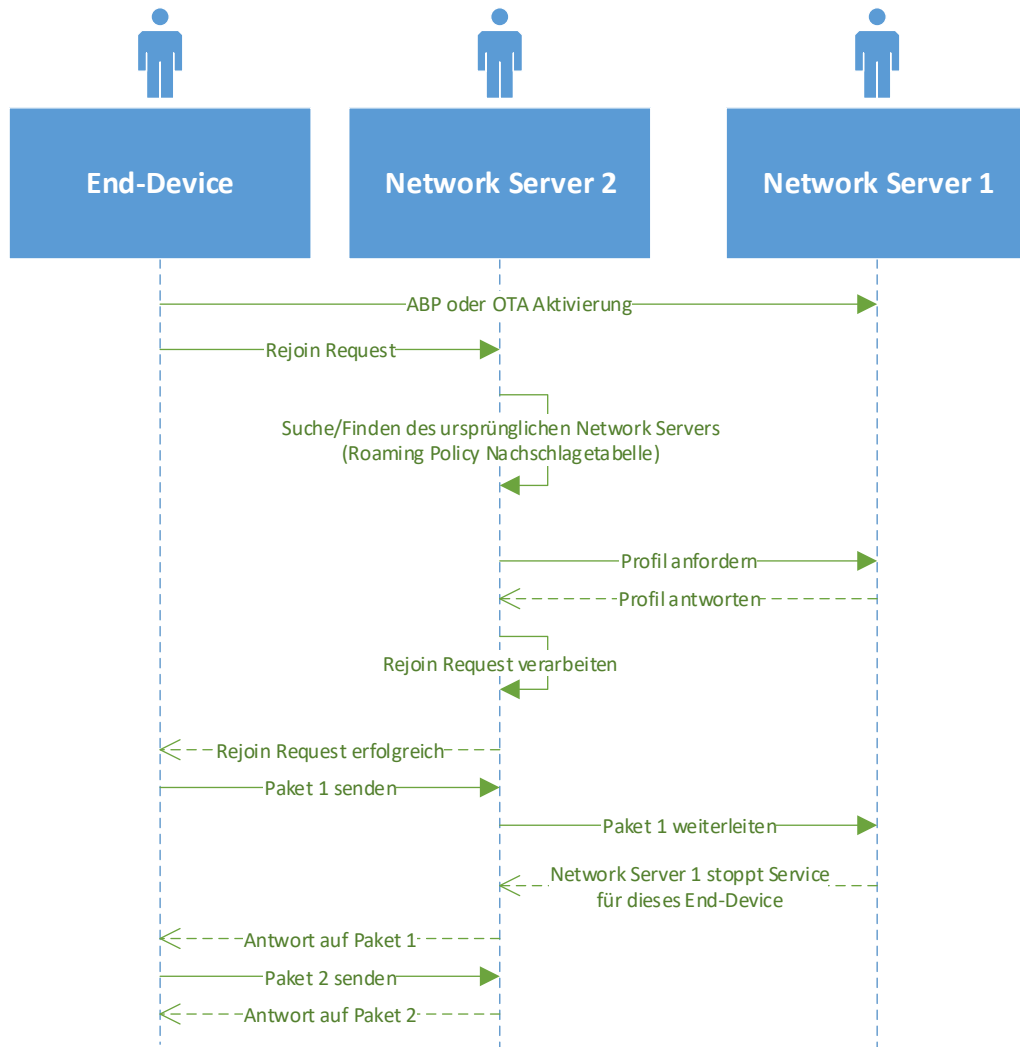


Abbildung 17 - Handover Roaming

In den zuvor beschriebenen Szenarien wurde davon ausgegangen, dass zwischen den beteiligten Network Servern eine Verbindung besteht. Es gibt Szenarien bei denen der ursprüngliche Home Network Server nicht mehr verfügbar ist, aber ein zuvor besuchter Network Server. Dafür wurde das OTA Roaming Activation Procedure spezifiziert. Die Aktivierung eines End-Devices erfolgt über zwei mögliche Varianten – Handover Roaming Activation oder Passive Roaming Activation. Es wird dabei davon ausgegangen, dass der Join Server der gleiche ist bzw. dem Join Server die Daten des ursprünglichen Home Network Server (über den zuletzt besuchten Network Server) zur Verfügung stehen. Ein Network Server empfängt einen Join Request von einem ihm unbekanntem End-Device. Über den Join Server wird über eine DNS Nachschlagetabelle der ursprüngliche Home Network Server gefunden. Es erfolgt ein Datenaustausch zwischen Network Server und

Join Server (u.a. Device Profile und weitere Informationen). Wird an einer Stelle festgestellt, dass der Network Server oder das End-Device unbekannt sind oder es kein business agreement zwischen dem neuen und dem ursprünglichen Network Server gibt, wird die Verbindung abgelehnt. Andernfalls sendet das End-Device Informationen zum neuen Network Server (vgl. LORAALLIANCE-2 2017, S. 38 ff.). Folgende Abbildung zeigt den Ablauf schematisch für beide Varianten der Aktivierung.

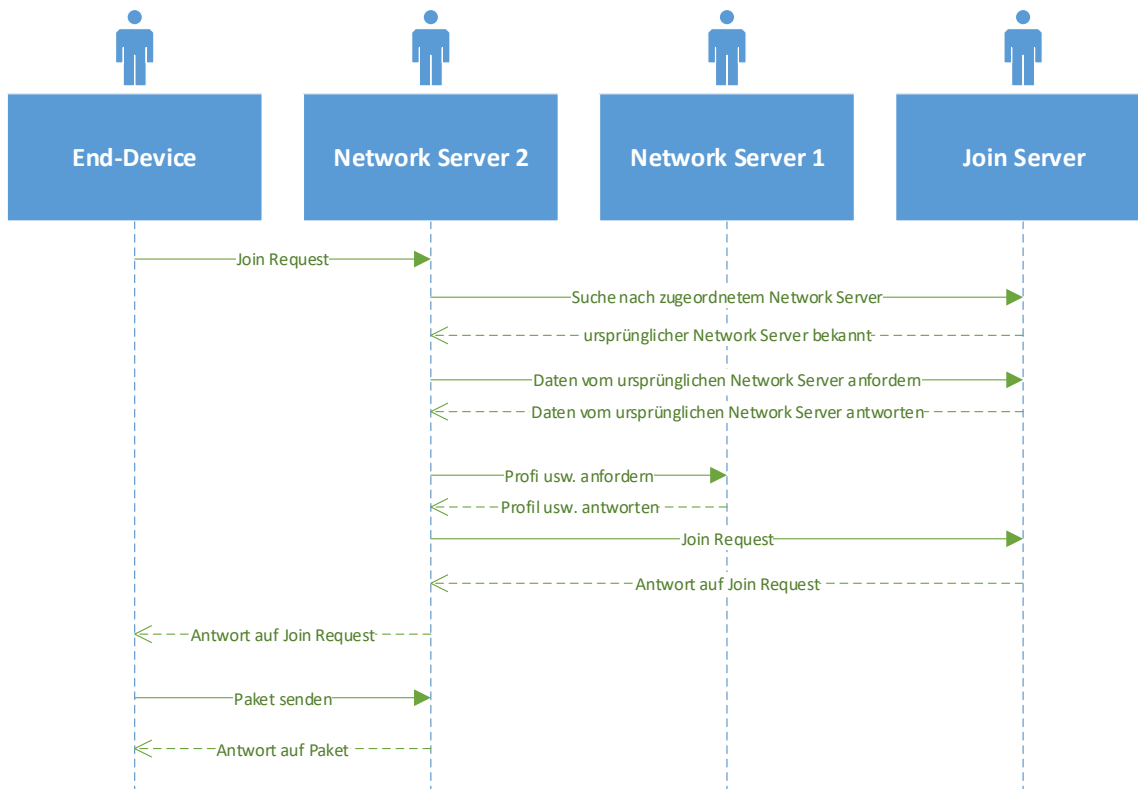


Abbildung 18 – OTA Roaming Activation Procedure (allgemein)

2.3.4 Profile

Profile dienen dem Network Server um spezifizierte Informationen zu erhalten. Es gibt Device Profile, Service Profile und Routing Profile.

Das **Device Profile** enthält Eigenschaften des End-Devices. Folgende Tabelle listet die Eigenschaften auf, welche für jedes Device Profile angegeben werden müssen. Zusätzlich gibt es weitere optionale Eigenschaften (vgl. LORAALLIANCE-2 2017, S. 53).

Eigenschaft	Beschreibung
DeviceProfileID	Eine ID des Profils.

SupportsClassB	Flag für die Unterstützung der Klasse B.
SupportsClassC	Flag für die Unterstützung der Klasse C.
MACVersion	LoRaWan Version, welche vom End-Device unterstützt wird.
RegParamsRevision	Revision der regionalen Parameter, welche vom End-Device unterstützt wird.
SupportsJoin	Gibt Auskunft, ob das End-Device die Aktivierung über OTA unterstützt. Falls nicht, dann ABP.
MaxEIRP	Maximum des EIRP Werts, welcher durch das End-Device unterstützt wird.
RFRegion	Name der RF Region.

Tabelle 5 - Eigenschaften Device Profile

Im **Service Profile** sind Informationen zur LoRa bzw. LoRaWan Kommunikation enthalten. Beispielsweise wird die Eigenschaft „PRAllowed“ gefordert, welche das passive Roaming erlaubt bzw. nicht erlaubt (vgl. LORAALLIANCE-2 2017, S. 53 f.).

Das **Routing Profile** enthält Informationen zur Einrichtung einer Datenebene zwischen Application Server und Network Server (vgl. LORAALLIANCE-2 2017, S. 54).

2.3.5 End-Device Class A, B, C

Ein End-Device der **Klasse A** arbeitet nach dem Aloha Prinzip, d.h. das End-Device sendet eine Information zum Server und erhält eine Antwort. Jedes End-Device arbeitet mindestens nach diesem Prinzip.

Wird aber eine Möglichkeit gefordert, um ein End-Device von der Serverseite anzusprechen, wird ein **Klasse B** End-Devices benötigt. Dabei horcht das End-Device zu definierten Zeitfenstern auf Anfragen aus dem LoRaWan Netzwerk (vgl. LORAALLIANCE-3 2018, S. 40).

Ein End-Device wird der **Klasse C** zugeordnet, wenn es permanent auf Anfragen horcht, wenn es nicht gerade sendet (Funktion der Klasse A). Diese Option ist nicht für low power Anwendungen geeignet (vgl. LORAALLIANCE-3 2018, S. 59).

2.3.6 Übertragung und Sicherheit von Schlüsseln

Für die Umsetzung der zuvor beschriebenen Funktionalitäten, müssen Schlüssel (Keys) zwischen den Teilsystemen (Application Server, Network Server, Join Server) ausgetauscht werden. Diese Informationen sollen im JSON Format über den http 1.1 (RFC2616) Standard übertragen werden. Um diese vertraulichen Informationen zu schützen, können sie mit Key Encryption Keys (kurz: KEK) verschlüsselt werden. Weiterhin kann jeder Key Encryption Key mit einem Key Encryption Key Label (kurz: KEKLabel) versehen werden. Die verwendeten Verfahren zur Verschlüsselung sind im RFC3394 spezifiziert (vgl. LORAALLIANCE-2 2017, S. 60 ff.).

2.3.7 Nachrichten und Nutzdaten

Die Nachrichten werden im JSON Format über http übertragen (Anfragen und Antworten). Anfragen generieren dabei eine http Post Nachricht an die Ziel URL. Das angefragte Ziel antwortet mit einer Nachricht und JSON formatierte Informationen. Jede Anfrage und jede Antwort enthält dabei spezifizierte Objekte (z.B. Protokollversion).

Die folgende Abbildung zeigt den Ablauf für eines Join Requests als Beispiel für einen Nachrichtentyp.

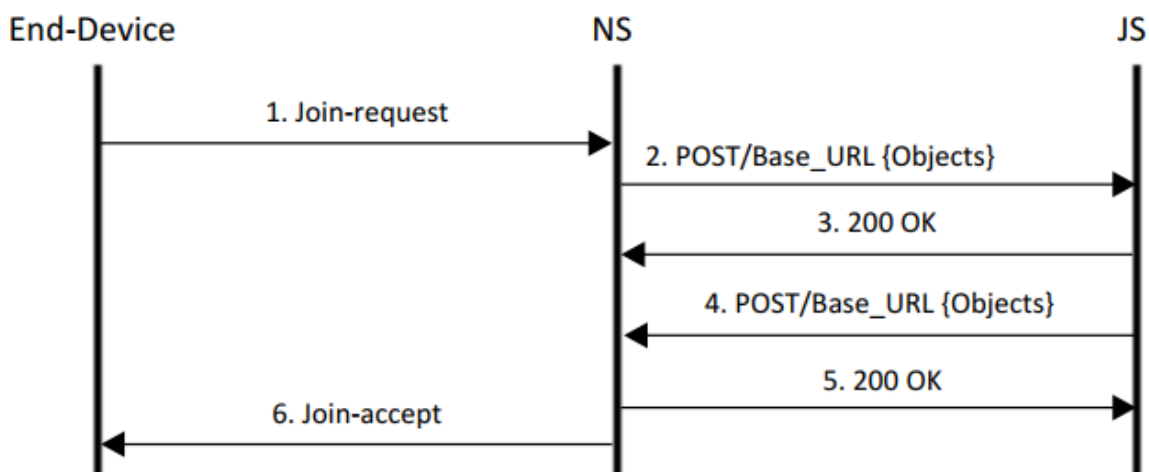


Abbildung 19 - Join Request über http

Unveränderte Übernahme: Abbildung 19: Backend messages carried over http Requests, S. 63 (<https://lora-alliance.org/sites/default/files/2018-04/lorawantm-backend-interfaces-v1.0.pdf>, verfügbar am 31.8.2019)

Weitere Typen von Nachrichten sind in der folgenden Tabelle aufgelistet.

Anfrage	Antwort
JoinReq	JoinAns
RejoinReq	RejoinAns
AppSKeyReq	AppSKeyAns
PRStartReq	PRStartAns
PRStopReq	PRStopAns
HRStartReq	HRStartAns
HRStopReq	HRStopAns
HomeNSReq	HomeNSAns
ProfileReq	ProfileAns
XmitDataReq	XmitDataAns

Tabelle 6 - Nachrichtentypen

Zu jedem Nachrichtentyp gibt es geforderte und optionale Informationen (Objekte, Argumente). Diese sind in der folgenden Abbildung kompakt dargestellt (vgl. LORAALLIANCE-2 2017, S. 62 ff.). Der jeweilige Datentyp (z.B. string) und die Interpretation (z.B. 1.0) dieser Information ist ebenfalls spezifiziert, aber nicht in der vorliegenden Arbeit dargestellt. Als Datentyp sind auch komplexe Datentypen (z.B. Device Profile Object) spezifiziert (vgl. LORAALLIANCE-2 2017, S. 66 ff.).

	JoinReq	JoinAns	RejoinReq	RejoinAns	AppSKeyReq	AppSKeyAns	PRStartReq	PRStartAns	PRStopReq	PRStopAns	HRStartReq	HRStartAns	HRStopReq	HRStopAns	HomeNSReq	HomeNSAns	ProfileReq	ProfileAns	XmitDataReq	XmitDataAns
ProtocolVersion	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
SenderID	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
ReceiverID	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
TransactionID	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
MessageType	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
SenderToken	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
ReceiverToken	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
MACVersion	M		M								M									
PHYPayload	M	Ms	M	Ms		M				M	Ms								M ¹	
FRMPayload																			M ¹	
Result		M		M		M		M		M		M		M		M		M		M
DevEUI	M		M		M	M	Os	M				M		M		M				
Lifetime		Ms		Ms				Ms	O			Ms								
SNwkSIntKey		Ms _{1a}		Ms _{1a}								Ms _{1a}								
FNwkSIntKey		Ms _{1a}		Ms _{1a}			Os _t					Ms _{1a}								
NwkSEncKey		Ms _{1a}		Ms _{1a}								Ms _{1a}								
NwkSKey		Ms _{1b}		Ms _{1b}			Os _t					Ms _{1b}								
FCntUp							Os													
DevAddr	M		M								M									
DeviceProfile										M	Of						Ms			
ServiceProfile							Os				Ms									
ULMetaData						M				M									M ²	
DLMetaData											Ms								M ²	
DLSettings	M		M							M										
RxDelay	M		M							M										
CFList	O		O							O										
AppSKey		Ms ₁		Ms ₁		Ms														
SessionKeyID		Ms ₁		Ms ₁	M	M														
DeviceProfileTimestamp										M	Of							Ms		
HNetID															Ms					
FCntDown																				
RoamingActivationType																		Ms		
DLFreq1																				Os
DLFreq2																				Os
VSExtension	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

Abbildung 20 - Nachrichten und deren Inhalt

2.3.8 Regionale Parameter

LoRaWan ist für den weltweiten Einsatz ausgelegt. In jedem Staat existieren unterschiedliche Regularien. Deshalb gibt es eine Vielzahl von Parametern, um LoRaWan im jeweiligen Gebiet einzusetzen. Beispielweise existiert für die Bundesrepublik Deutschland eine Spezifikation mit dem Namen EU863-870. Dabei wird ein frei verwendbares Frequenzband von 863 bis 870 MHz definiert, das Präambel Format vorgeschrieben (z.B. 8 Symbole) und vieles mehr (vgl. LORAALLIANCE-4 2017, S. 15 ff.).

3 Präzisierung der Aufgabenstellung

In diesem Kapitel erfolgt die Präzisierung der Aufgabenstellung. Die strukturierte Vorgehensweise orientiert sich an dem Organisationhandbuch des Bundesministeriums des Innern (vgl. BMI 2018, S. 59 ff.).

3.1 Voruntersuchung

Ein Teil der Aufgabenstellung ist es, geeignete Kriterien zur Beurteilung von Bienenvölkern zu finden und mögliche Lösungsansätze aufzuzeigen. Deshalb wurde eine Voruntersuchung durchgeführt. Als Ergebnis der Voruntersuchung werden Aussagen über die Machbarkeit oder Prognosen der zu erwartenden Kosten erwartet. Die erforderlichen Methoden zur Datenerhebung sind Dokumentenanalyse, Fragebogen oder Interviews (vgl. BMI 2018, S.81). In diesem Fall erfolgt die Datenerhebung zum Aufstellen geeigneter Kriterien zur Beurteilung von Bienenvölkern durch Literaturrecherche.

Die **Temperatur** ist ein wichtiges Kriterium. Anhand der Temperatur lässt sich feststellen, ob ein Volk über Brut verfügt oder nicht. Brütende Völker haben eine Temperatur von 33 °C, während nicht brütende Völker Kerntemperaturen zwischen 21 und 23 °C aufweisen. Diese Erkenntnis gibt Auskunft, ob beispielsweise die Königin während der Ameisensäurebehandlung (siehe dazu Kapitel 2.1.3) verstorben ist. In diesem Fall sinkt die Temperatur um ca. 10°C, weil keine Brut vorhanden ist. Auch in weiteren Fällen, z.B. Völkerentwicklung im Frühjahr oder von Ablegern, kann durch eine steigende Temperatur eine positive Entwicklung diagnostiziert werden. Für die Behandlung mit Oxalsäure im Dezember ist sinkende Temperatur als positiv zu bewerten, weil diese Behandlung eine Brutfreiheit voraussetzt. Der Sitz der Bienenraube in der Beute gibt Auskunft, ob ausreichend Futter im Winter vorhanden ist. Dies trifft zu, wenn die Raube im unteren Teil der Beute sitzt. Sitzt die Raube im Februar (oder früher) direkt unter dem Deckel, so muss nachgefüttert werden, damit das Volk nicht verhungert. Auch ein Schwarmvorgang lässt sich anhand der Temperatur erkennen. Dabei steigt die Kerntemperatur auf 38°C, kurz bevor es zum Schwärmen kommt (vgl. PERDELWITZ 2018, S. 15 f.).

Mögliche technische Umsetzungen wurden mit Wärmebildkameras erprobt. Dies erfordert keinen Eingriff in das Bienenvolk, stellt aber Bedingung an den verwendeten Werkstoff bzw. Oberfläche der Bienenbeute. Dabei wurden gute Erfahrungen mit unbehandelten Holzbeuten mit einer Wandstärke zwischen 18 bis 20 mm gemacht. Aber Isolation, dunkel-matt lackierte Oberflächen oder metallische Gegenstände beeinflussen das Ergebnis durch Reflexion negativ (vgl. PERDELWITZ 2018, S. 15 f.). Ein Nachteil dieser Methode ist auch die geringe Informationsdichte, weil die Aufnahmen in der Regel nicht automatisch durchgeführt werden können. Eine andere Herangehensweise erfolgte mit Feuchte-

sensoren, Temperatursensoren und Messrechner. Die Temperatursensoren wurden in der Bodenzarge und in der Brutzarge positioniert. Ein Feuchtesensor für den Beuteninnenraum sowie die äußere Umgebung wurden installiert. Die erfassten Messwerte werden mit einem Messrechner erfasst und über eine WLAN Verbindung an einen Webserver gesendet und dort mit einer Website dargestellt (vgl. BRELL 2019, S. 18 f.).

Ein weiteres wichtiges Kriterium stellt die **Masse** der besetzten Bienenbeute dar. Beispielweise wird einem Bienenvolk Futter verabreicht (siehe Kapitel 2.1.3). Die Menge wird dabei von der Stärke des Volkes bestimmt. So errechnet sich der Winterbedarf des Futters als Faustformel mit etwa 1 kg Futter (Sirup, Zuckerlösung o.ä.) je Wabe im Deutsch Normalmaß zzgl. etwas Überschuss. Somit etwa 12 kg für 9 bis 10 Waben (vgl. GETTERT 1998, S. 120). Insbesondere zur Honigernte ist die Masse ebenfalls ein Indikator inwieweit eine Ernte sinnvoll ist. So wurde eine beidseitig gefüllte Honigwabe im DN mit 2,54 kg⁶ ermittelt.

Weiterhin wurde ein Zusammenhang zwischen Temperatur und **Lufffeuchte** in Bezug auf die Bienenraube festgestellt. Die Honigbienen sorgen für eine Entfeuchtung der Bienenraube durch Austausch warmer und feuchter Luft in der Bienenraube mit zugeführter kalter und trockener Luft. Es ist den Bienen damit möglich innerhalb weniger Minuten die relative Lufffeuchte um bis zu 20 Prozent zu senken (vgl. HOBOS 2016).

Das Fazit der Voruntersuchung ergibt, dass die Bewertung der Temperatur und die Luftfeuchte innerhalb des Bienenstocks ein wichtiges Kriterium darstellt. Die Masse des gesamten Bienenstocks stellt ein weiteres Kriterium dar. Für die Auswertung dieser Kriterien ist ein zeitlicher Verlauf, z.B. für die Masse $m(t)$, interessant. Um einen Bezug zur Wetterlage herzustellen zu können, ist die Temperatur, die Luftfeuchte und möglicherweise auch der Luftdruck (außerhalb der Bienenbeute) ebenfalls interessant.

3.2 Hauptuntersuchung

Die Hauptuntersuchung nutzt die Erkenntnisse der Voruntersuchung und stellt den Kern der Vorarbeiten für die nächsten Phasen dar. Es werden unabhängig vom inhaltlichen Schwerpunkt immer die Phasen Ist-Erhebung, Ist-Analyse und Soll-Konzeption durchlaufen. Eine Unterteilung in inhaltlich zusammenhängende und abgrenzbare Einheiten bietet sich an (vgl. BMI 2018, S.81 f.).

⁶ Gewichtsvergleich Honigwaben (http://www.imkerverein-ravensburg.de/fileadmin/daten_ravensburg/Daten/Dateien/Kurs08Gewicht-Honigwabe.pdf, verfügbar 11.9.2019)

3.2.1 Ist-Erhebung

In der Ist-Erhebung erfolgt die Erhebung und Sammlung von Informationen, um die Ist-Situation in Bezug zur Aufgabenstellung möglichst konkret darzustellen. Die Ist-Erhebung nutzt Erhebungstechniken, wie Interviews, Dokumentenanalyse und viele mehr (vgl. BMI 2018, S.82 f.). In diesem Fall existierte keine Historie oder keine weiteren Informationsträger. Deshalb wurden Kriterien gelistet, welche Einfluss auf die Ziellösung sowie die Systemimplementierung vor Ort haben könnten. Diese Kriterien wurden mit Hilfe eines Fragebogens erfasst. Zur Beantwortung erfolgte eine Ortsbegehung der Standorte der Bienenbeuten sowie des LoRa Gateways.

Nummer	Kriterium	Antwort
1	Standort Bienen	
1.1	Besteht eine Stromversorgung (230V Wechselspannung, 50 Hz)?	Nein
1.2	Besteht eine Verbindung zu einem zugänglichen kostenlosen LAN/WLAN Netz?	Nein
1.3	Befinden sich die Bienenbeuten innerhalb eines Gebäudes? (bei ja: Typ des Gebäudes [offene Umgebung, geschlossene Umgebung])	Ja, offene Umgebung
1.4	Werkstoff des evtl. vorhandenem Gebäude?	Holz, Dachblech
1.5	Ist der Standort grundlegend vor Witterungseinflüssen wie starkem Regen, Schnee oder extremer Sonneneinstrahlung geschützt?	Ja
2	Standort LoRa Gateway	
2.1	Standort LoRa Gateway ist identisch zum Standort der Bienen?	Nein
2.2	Besteht eine Stromversorgung (230V Wechselspannung, 50 Hz)?	Ja

	nung, 50 Hz)?	
2.3	Besteht eine Verbindung zu einem zugänglichen kostenlosen LAN/WLAN Netz?	Ja
2.4	Entfernung Standort LoRa Gateway und Standort Bienen (Luftlinie)?	Ca. 800 m
2.5	Montageort der Antenne (außen/innen, Höhe)?	Außenantenne, Hausdach Montagehöhe ca. 12 m
3	Bienenbeute	
3.1	Welcher Typ von Bienenbehausung wird genutzt?	Magazinbeute, Typ: Segeberger, DN (je Magazin 11 Vollwaben)
3.2	Aus welchem Werkstoff besteht die Bienenbehausung?	Styropor
3.3	Wie ist die Bienenbeute (z.B. Palette) aufgestellt?	Kantholzbank 10 cm (parallele Balken), Höhe ca. 40 cm vom Boden
3.4	Wie schwer ist eine Bienenbeute maximal?	Annahme 20 kg je Magazin, Trachtzeit bis zu 4 Magazine = Gesamtgewicht ca. 80 kg + Puffer = 120 kg
3.5	Grundfläche der Bienenbeute?	Ca. 50 x 50 cm
4	Chemikalien in der Beute	

4.1	Mit welchen Chemikalien erfolgt die Varroa-Behandlung?	Ameisensäure (60 Prozent) über Zerstäuber
-----	--	---

Tabelle 7 - Fragenkatalog

Folgende Abbildung zeigt den Bienenstandort mit sechs Bienenvölkern.



Abbildung 21 - Bienenstandort

3.2.2 Ist-Analyse

Die Ist-Analyse verarbeitet die gewonnenen Erkenntnisse der Ist-Erhebung. Sie ermittelt durch eine detaillierte Analyse Soll-/Ist-Abweichung. Dabei wird der Untersuchungsbe-
reich hinterfragt und Hinweise oder Anregungen von Informationsträgern ausgewertet (vgl. BMI 2018, S.84 f.). An dieser Stelle fließen praktische Erfahrung in der Imkerei des Autors (als Informationsträger) ein. Anhand der gewonnenen Erkenntnisse werden Ent-
scheidungen bezüglich der Machbarkeit oder Lösungsvorschlägen getroffen.

3.2.2.1 Standort Bienen

Am Standort der Bienen ist keine eigene Stromversorgung vorhanden. Somit sind ener-
giesparende Systeme mit eigener Energiequelle, z.B. Batterie, erforderlich. Eine Verbind-
ung zu einem LAN bzw. WLAN Netz ist nicht vorhanden, somit eignet sich der Standort
nicht für die Installation eines Gateways. Zudem macht es den Einsatz einer drahtlosen
Datenübertragung erforderlich. Das Gebäude ist aus Holz gefertigt, nicht geschlossen und
besitzt ein Dach. Somit bietet es Schutz vor größeren Witterungseinflüssen. Auf Grund

der verwendeten Versorgungsspannung, voraussichtlich kleiner gleich 9V Gleichspannung, ist das Gerät der Schutzklasse III zuzuordnen (vgl. WIKIPEDIA-1 2019). Um die Bauteile (Elektronik) zu schützen, sollte ein geschlossenes Gehäuse verwendet werden. Dabei sollte mindestens die Schutzart IP52 von dem eingesetzten Gehäuse erfüllt sein (vgl. WIKIPEDIA-2 2019). Als möglicher Störfaktor ist das Dachblech zu notieren (Reflexion von Funksignalen). Falls in der Implementierungsphase Verbindungsstörungen auftreten, könnten die Antennen beispielsweise seitlich am Unterstand montiert werden.

3.2.2.2 Standort des LoRa Gateways

Das Gateway wird an ca. 800 m Entfernung des Bienenstandortes installiert. Diese Entfernung ist für eine LoRa basierte Funkverbindung unkritisch. Infrastruktur zur Energieversorgung sowie LAN bzw. WLAN sind am Standort vorhanden. Auf dem Hausdach wird in ca. 12 m Höhe eine, für LoRa Frequenzen geeignete, Außenantenne montiert. Die Außenantenne wird mit einem Antennen-Koaxialkabel mit dem Gateway verbunden. Das Gateway wird voraussichtlich mit einer Gleichspannung kleiner 10 V, gespeist aus einem externen Netzteil, betrieben. Es ist somit der Schutzklasse III zuzuordnen (vgl. WIKIPEDIA-1 2019). Es besteht für das Gateway kein Einfluss durch Witterung. Die Bauteile sollten in einem Gehäuse verbaut werden.

3.2.2.3 Bienenbeute

Die eingesetzte Beute besteht aus dem Werkstoff Styropor (Expandiertes Polystyrol, EPS). Damit ist eine Messung der Temperatur mit Hilfe von Wärmebildkamera ausgeschlossen, weil die Messergebnisse durch die gute isolierende Wirkung des Werkstoffs verfälscht werden würden. Vergleicht man beispielsweise die spezifische Wärmekapazität (auch Dämmwert) von Holz⁷ und Styropor⁸, so stellt man fest, dass Styropor die Wärme um den Faktor ca. sechs schlechter leitet. Weiterhin ist die Weiterverarbeitung von Bildern auf Grund der Datenmenge im Kontext von LoRa eher ungeeignet. Die Kosten eine Wärmebildkamera übersteigen um ein Vielfaches die Kosten einer Implementierung mit gewöhnlichen Sensoren. Die Messung über Sensoren, z.B. Temperatursensoren, scheint aus oben genannten Gründen geeignet zu sein. Die Magazinbeute ist stapelbar. Damit kann versucht werden, die Kabel der Sensoren im Innerraum nach außen zu führen, ohne die Beute mechanisch zu verändern (Bohrungen etc.). Das sollte ein Vorteil sein, wenn zur Bestimmung der optimalen Lage der Sensoren, eine Neupositionierung erfolgen muss. Die Gesamtmasse einer Bienenbeute kann bis zu 120 kg betragen, somit ein Messsystem (Waage) für dieses Gewicht entsprechend ausgelegt sein.

⁷ Wärmeleitfähigkeit λ von Holz (Eiche) 0,2 W / (mK)

⁸ Wärmeleitfähigkeit λ von Styropor 0.035 W / (mK)

3.2.2.4 Chemikalien in der Beute

Zur Varroa-Behandlung wird Ameisensäure eingesetzt. Die Reaktion der im Innenraum eingesetzten Sensoren während der Behandlung muss beobachtet werden. Im schlechtesten Fall können die Sensoren zerstört werden. Im Rahmen der Diplomarbeit wird dieser Aspekt nicht betrachtet.

3.2.2.5 Ergebnis der Ist-Analyse

Die Aufgabe kann mit der LoRa bzw. LoRaWan Technologie technisch umgesetzt werden. Vor der konkreten Implementierung ist ein Auswahlverfahren für die am Markt erhältlichen LoRa bzw. LoRaWan Komponenten erforderlich.

Die zu verwendeten Schutzklassen und Schutzarten sind für die LoRa Nodes sowie den LoRa Gateway unkritisch.

Zur Ermittlung der Gesamtmasse einer Bienenbeute muss eine Waage ausgewählt bzw. entworfen werden, welche in die Gesamtlösung integrierbar ist.

Zur Messung der Temperatur sowie Luftfeuchte sind eine Vielzahl von Sensoren am Markt erhältlich. Auch diese sollten im Kontext des eingesetzten LoRa Node (z.B. Schnittstellen) betrachtet werden.

Zusammenfassend sind Problembereiche aufgelistet, welche in der Soll-Konzeption gelöst werden müssen.

- Auswahl Hardware LoRaWan Node
- Auswahl Hardware LoRaWan Gateway
- Auswahl Sensoren zur Messung der Temperatur und Luftfeuchte (Innenraum der Beute und Außenbereich)
- Auswahl Lösung für Waage zur Ermittlung der Gesamtmasse einer Bienenbeute
- Konkretisierung Sendeintervall / Visualisierung der Messergebnisse

3.2.3 Soll-Konzeption / Grobentwurf

Die Soll-Konzeption hat das Ziel Lösungsansätze für ausgewählte Bereiche, i.d.R. Schwachstellen oder Problembereiche aus der Ist-Analyse, zu entwickeln. Häufig werden dabei Kreativitätstechniken (z.B. Brainstorming) verwendet, um mit Stakeholdern gemeinsam Lösungen zu erarbeiten bzw. Lösungen auf Grund geltender Rahmenbedingungen zu verwerfen. Die Bewertung ist dann unkompliziert, wenn eine Messung monetärer Unterschiede möglich ist. (vgl. BMI 2018, S.86 f.).

In den folgenden Unterkapiteln werden Lösungen für die ermittelten Problembereiche dargestellt.

3.2.3.1 Hardware LoRaWan Node

Bei der Auswahl der Hardware ist zu berücksichtigen, dass es sich um einen Prototyp handelt. Es sollte auf fertige Entwicklungsboards zurückgegriffen werden, um entsprechend Zeit und Kosten zu sparen.

Folgende Minimalanforderungen wurden festgelegt:

- Programmierung mit einer freien Entwicklungsumgebung muss möglich sein
- Schnittstellen (z.B. I²C, SPI) für den Betrieb von Sensoren oder Erweiterungsplatinen müssen vorhanden sein
- LoRa Kommunikation muss möglich sein (LoRa Chip SX1276 – geeignet für low power applications und einsetzbar im Frequenzband für Deutschland)
- Batteriebetrieb muss möglich sein

Die Marktrecherche konzentrierte sich auf fertige Entwicklungsboards der Marken Arduino und Adafruit. Für diese Boards existieren eine große Community, eine Menge an Beispielprojekten sowie eine Vielzahl an nutzbaren Bibliotheken. Damit wird der Ansatz zur schnellen Bereitstellung von fertigen Lösungen unterstützt.

Folgende Tabelle zeigt die Auswahlkriterien der berücksichtigten Boards.

Option / Kriterium	Option 1	Option 2	Option 3
	Adafruit Metro 328 mit Header (vgl. ADAFRUIT-1 2019) und Adafruit RFM 95 Lora Radio Transceiver (vgl. ADAFRUIT-2 2019)	Arduino MKR WAN 1300 (vgl. ARDUINO-1 2019)	Arduino Uno Rev 3 (vgl. ARDUINO-2 2019) mit Dragino LoRa Shield (vgl. DRAGINO-1 2019)
Abmessung (Breite x Länge [mm])	53 x 71 und 29 x 25	25 x 67,64	53,4 x 68,60 und 62 x 43
Preis (inkl. 19 % MwSt.), Versand aus Deutschland	21 Euro + 24 Euro	40 Euro	20 Euro + 35 Euro

Spannungsversorgung	7 - 9 V DC	5 V (z.B. 2 x AA oder AAA Batterien)	7 - 12 V DC
Schnittstellen	1x, UART, 1 x SPI, 1 x I ² C, analoge In-/Output Pins, Pins, GPIO Pins, PWM	1 x UART, 1 x SPI, 1 x I ² C, analoge In-Output Pins, GPIO Pins, PWM	1 x SPI, 1 x I ² C, analoge In- Output Pins, GPIO Pins, PWM

Tabelle 8 - Auswahl Boards

Alle aufgeführten Boards werden von der freien Arduino Entwicklungsumgebung unterstützt. Option 1 und 3 haben eine SMA Antennenbuchse zum Anschluss der Antenne auf der Platine integriert, welche in der Option 2 durch ein Adapterkabel gelöst werden könnte. Die Standardschnittstellen wie I²C, SPI und diverse GPIOs werden von allen Optionen bereitgestellt. Die Option 3 bietet keine UART Schnittstelle. Alle aufgezeigten Optionen verwenden das Semtech Modem SX1276, aber nur die Option 2 bietet eine kompakte und platzsparende OnBoard Umsetzung. Die Option 2 bietet den günstigsten Einkaufspreis. Als Ergebnis kann festgehalten werden, dass Option 2 gewählt wird. Die Gründe dafür sind die kompakte Abmessung ohne Verzicht auf Schnittstellen, Eignung zum Batteriebetrieb sowie der günstigste Preis im Vergleich.

3.2.3.2 Hardware LoRaWan Gateway

Zur Auswahl des Geräts für das LoRaWan Gateway ist neben der Hardware auch die entsprechende Software zu berücksichtigen. Im Rahmen der Diplomarbeit soll die Software nicht implementiert werden, sondern eine existierende Implementierung eingesetzt werden. Mittlerweile gibt es eine Vielzahl an verfügbaren Produkten in diversen Preiskategorien. Die Recherche beschränkte sich auf kostengünstige Produkte, welche nicht unbedingt einen industriellen Einsatz ermöglichen. Die Bezeichnung „fertiges“ Produkt bezeichnet hier ein Produkt mit Gehäuse und Netzteil, welches als solches angeboten wird. Eine kompakte Darstellung der Rechercheergebnisse zeigt die folgende Tabelle.

Option / Kriterium	Option 1	Option 2	Option 3
	Dragino LoRaWan Gateway LG308 (vgl. DRAGINO-2 2019)	RAK7243C Pilot Gateway (vgl. RAK-1 2019, Seite 4 ff.)	Lite Gateway for LoRa Technology (vgl. IMST-1 2019, Seite 6 ff.)

„fertiges“ Produkt	Ja	Ja	Ja
Teilkomponenten erhältlich	Nein	Ja (vgl. RAK-2 2019, Seite 3 ff.)	Ja (vgl. IMST-2 2018, Seite 2 ff.)
LoRa Zertifiziert	Ja	Ja	Ja
Hardware	Keine Angabe	Raspberry Pi 3B+, RAK2245, RAK2013 Erweiterungen	Raspberry Pi, iC880A-SPI
Open Source Software	Ja	Ja	Ja
Individualisierbar	Ja mit Linux und C Kenntnissen	Ja mit Linux und C Kenntnissen	Ja
Build-in LoRaWan App/Network Server	Evtl. Nur mit TTN Network	TTN, LoRaServer.io	Auslieferungszustand nur LoRa Concentrator installiert
vollständig Cloud oder OnPremise umsetzbar	Cloud ja, OnPremise nicht sicher, da evtl. immer TTN erforderlich	Cloud (TTN) und On-Premise (eigene LoRaServer.io Instanz) möglich	Cloud und On-Premise können durch Installation von Software erreicht werden.
Anzahl Kanäle	10	8	10
einsetzbar für Deutschland (EU868)	Ja	Ja	Ja
LoRaWan Version	Keine Angabe	1.0.2	1.0.2
Empfehlung für	Ca. 1000	Keine Angabe	Keine Angabe

Anzahl LoRa Nodes			
Dokumentation	Umfangreiche Dokumentation in Englisch auf der Website des Herstellers abrufbar	Umfangreiche Dokumentation in Englisch auf der Website des Herstellers abrufbar	Umfangreiche Dokumentation in Englisch auf der Website des Herstellers abrufbar
Preis	Ca. 200 Euro (ohne LTE Option)	Ca. 200 US Dollar (ohne LTE Option)	Ca. 200 Euro
Anbieter	Draguino, erhältlich z.B. über: http://Antraktek.de	RAKwireless, erhältlich z.B. über: https://rakwireless.com	IMST GmbH, erhältlich z.B. über: https://wireless-solutions.de

Tabelle 9 - Auswahl LoRaWan Gateways

Alle aufgeführten Optionen bieten den Anschluss einer externen Antenne sowie einen Port für LAN. Sie können mit Linux Kenntnissen angepasst bzw. erweitert werden. Alle Optionen enthalten den LoRa Concentrator von Semtech. Für ein LoRaWan Netzwerk werden zusätzliche Komponenten benötigt (siehe Kapitel 2.3.2). Dazu kann die Option 1 und 2 in das TTN (cloud) integriert werden. Die Optionen 2 und 3 bieten die Möglichkeit mit Loraserver.io ein eigenes LoRaWan Netzwerk (onpremise) zu erstellen. Im Auslieferungszustand ist bei der Option 3 nur der LoRa Concentrator installiert. Weitere Softwarekomponenten müssen nachträglich eingerichtet werden. Die Anbieter der Option 2 und 3 bieten die LoRa Concentrators auch als einzeln erhältliches Produkt (LoRa Shield) für den Raspberry PI an. Die Kosten sind dabei bei beiden Optionen mit ca. 120 Euro anzusetzen.

Für die Bearbeitung der Diplomarbeit wurde sich auf Grund des Supports aus Deutschland und der besseren Verfügbarkeit (Lieferzeit 2 Tage) für das einzeln erhältliche LoRa Shield der Option 3 entschieden. Der Concentrator IMST iC880A wird von LoRaServer unterstützt (vgl. LORASERVER-1 2019).

3.2.3.3 Sensoren zur Temperatur und Luftfeuchtemessung

Zur Auswahl geeigneter Sensoren wurden folgende Minimalanforderungen festgelegt:

- Geringer Stromverbrauch
- Schnittstelle über I²C oder SPI (da diese vom Board unterstützt werden)

- Verfügbar als einsatzfähige Leiterplatte mit entsprechende Vorbeschaltung (bezeichnet als Breakout / Shield)
- Zu erfassender Temperaturbereich -20 °C bis 50°C
- Verfügbarkeit von freien Bibliotheken für die Arduino Entwicklungsumgebung

Durch Recherche im Internet wurden folgende Sensoren ermittelt und für einen Vergleich ausgewählt. Alle Sensoren werden oft für Do-It-Yourself Projekten im IoT Kontext verwendet und Bibliotheken für die Arduino Entwicklungsumgebung sind für jede Option vorhanden.

Option / Merkmal	Option 1	Option 2	Option 3
	DHT-11 (vgl. AOSONG, Seite 2 ff.)	DHT-22 / AM2302 (vgl. LIU 2019, Seite 1)	Bosch BME280 (vgl. Bosch 2018, Seite 2 ff.)
Datenschnittstellen	Serieller Single-Wire-Bus (bidirektional)	Serieller Single-Wire-Bus (bidirektional)	SPI, I ² C
Betriebsspannung	3,5 V bis 5 V DC	3,3 V bis 5 V DC	1,71 V bis 3,6 V DC
Stromaufnahme	Messung: 0,3 mA Standby: 60 µA	Messung: 1,5 mA Standby: 50 µA	Messung: je nach Typ der Messung zwischen 0,34 mA und 0,7 mA Standby: 0,2 µA bis 0,5 µA Sleep: 0,1 µA bis 0,3 µA
Messfunktionen	Temperatur, relative Luftfeuchte	Temperatur, relative Luftfeuchte	Temperatur, relative Luftfeuchte, relative Luftdruck
Temperaturbereich / Genauigkeit	Keine Angabe	-40 °C bis 80 °C	-40 °C bis 85 °C

	Keine Angabe	+/- 0,5 °C	+/- 1 bis 1,5 °C
Rel. Luftfeuchte Bereich / Genauigkeit	Keine Angabe Bei 25 °C +/- 5%	0 % bis 100 % +/- 2 % bis +/- 5 %	0 % bis 100 % +/- 3 %
Einsetzbar In- nen/Außen	Gehäuse erforder- lich	Gehäuse erforder- lich	Gehäuse erforder- lich

Tabelle 10 - Auswahl Sensoren

Alle Optionen bieten die erforderlichen Messfunktionen für Temperatur und relative Luftfeuchte sowie deren geforderte Messbereiche (für Option 2, 3 dokumentiert). Sie verfügen über entsprechende Standard-Datenschnittstellen zur Anbindung zum Einbinden in Entwicklerboards. Alle Sensoren sind als Leiterplatte mit Vorbeschaltung erhältlich. Unterschiede zeigen sich in der Spannungsversorgung, beim Stromverbrauch sowie Mechanismen zur Stromersparnis oder bei der Genauigkeit bei der Messwertaufnahme. Im Hinblick auf eine Batterieversorgung mit 3 V (siehe Kapitel 3.2.3.1) ist nur die Option 3 geeignet, weil die Optionen 1 und 2 eine Betriebsspannung von größer 3 V fordern. Weiterhin bietet die Option 3 auch die Möglichkeit einer Messung des relativen Luftdrucks, erzielt den geringsten Stromverbrauch im Standby Modus und bietet zusätzlich einen Sleep Modus. Dadurch wird der geringste Stromverbrauch im Vergleich erzielt.

3.2.3.4 Waage zur Ermittlung der Gesamtmasse einer Bienenbeute

Bei der Auswahl der Waage ist zu berücksichtigen, dass diese nicht geeicht sein muss und es keine Vorgabe hinsichtlich einer Genauigkeitsklasse gibt. Folgende Minimalanforderungen werden an die Waage gestellt:

- Integrierbar in das eingesetzte LoRaWAN Node (Signaleingang oder Datenschnittstelle)
- Geringer Energiebedarf
- Messung einer Masse von bis 120 kg muss möglich sein
- Geringer Preis

Option / Kriterium	Option 1 GRAM K2-F (vgl. WAAGENET 2019)	Option 2 XLOG Bienen- stockwaage SMS / GPRS (vgl. XLOG 2019), Seite 7 ff.)	Option 3 Individualentwicklung
-----------------------	---	--	--

Übertragung Messwert	Via Datenschnittstelle RS232	SMS / GPRS	
Geeicht	Ja	Keine Angabe	
Typ	Plattform Waage	Plattform Waage	
Kapazität	150 kg	200 kg	
Preis	Ca. 630 Euro	Ca. 890 Euro	
Anbieter	GRAMM, erhältlich z.B. über: https://Waagenet.de	XLOG, erhältlich z.B. über: https://www.bienen-ruck.de	

Tabelle 11 - Auswahl Waagen

Eine Nutzung des Messwerts im LoRa Node ist bei der Option 1 unter Umständen mit dem Einsatz eines zusätzlichen RS232 Konverters möglich. Die Option 2 bietet keine Möglichkeit dazu an. Beide Optionen bieten umfangreiche Features, welche in diesem Rahmen nicht benötigt werden. Auf Grund des hohen Preises kann keine der Optionen verwendet werden. Einzig das Konzept einer Plattformwaage scheint geeignet zu sein, da diese mechanisch einfacher umzusetzen ist, als beispielsweise die Ermittlung der Masse auf Zug mit einem Kran o.ä.

An dieser Stelle wird das zu Grunde liegende Prinzip einer Plattformwaage näher beschrieben, weil es die Basis zur Umsetzung der Option 3 darstellt. Bei Plattformwaagen befindet sich eine Wägezelle unter einer Plattform, welche von oben belastet wird. Die Wägezelle ist dabei ein Federkörper, auf dem Dehnungsmessstreifen angebracht sind. Oft besteht dieser Federkörper aus Stahl oder Aluminium. Der Federkörper muss robust, aber auch elastisch sein, damit eine elastische Verformung des Federkörpers bei Belastung festgestellt werden kann. Die angebrachten Dehnungsmessstreifen sind elektrische Leiter, welche bei Änderung ihrer Länge, ihren elektrischen Widerstand ändern. Bei Dehnung wird dieser Widerstand größer und bei einer Stauchung kleiner. Bei den Plattformwägezellen sind vier Dehnungsmessstreifen angebracht (vgl. SCHMIDT 2019). Diese Dehnungsmessstreifen sind als Wheatstone-Messbrücke aufgebaut. Nach Anlegen der Eingangsspannung (U_E) tritt eine Brückenspannung (Ausgangsspannung, U_A) auf, deren Größe und Polarität abhängig von den vier Widerständen ist. Diese Schaltung kann auch

zur Ermittlung von einem unbekanntem Widerstand genutzt werden. Mit folgender Formel wird die Brückenspannung (U_A) berechnet (vgl. FELDERHOFF 1982, S. 82 f.).

$$U_A = U_E \left(\frac{R_2}{R_2 + R_1} - \frac{R_4}{R_4 + R_3} \right)$$

Formel 13 - Ausgangsspannung Wheatstone-Messbrücke

Die Schaltung der Wheatstone-Messbrücke ist in der folgenden Abbildung dargestellt.

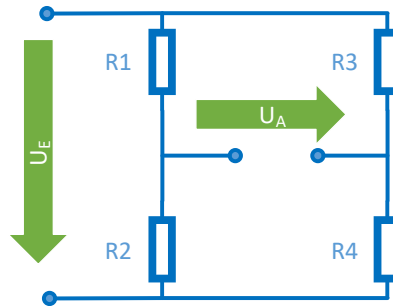


Abbildung 22 - Schaltung Wheatstone-Messbrücke

Die gemessene Brückenspannung wird im nächsten Schritt verstärkt. Dieses zeit- und wertkontinuierliche Signal wird mit einem Analog-Digital-Wandler (ADC) in ein zeit- und wertdiskretes Signal überführt (digitalisiert). Letztlich erfolgt die Interpretation in ein darstellbares Messergebnis für die Masse mit der Einheit Kilogramm.

Eine Wägezelle unterliegt äußeren Einflussfaktoren. Jedes Material reagiert auf Temperaturveränderungen und somit dehnt bzw. staucht sich der Dehnungsmessstreifen und verursacht eine temperaturabhängige Störung. Weiterhin wird bei der Ermittlung der Masse von einer senkrecht auftretenden Kraft (Gewichtskraft) ausgegangen. Es können auch unerwünschte Kräfte in andere Richtungen auftreten, welche das Messergebnis ebenfalls beeinflussen. Durch Justage, Kalibrierung sowie Eichung müssen diese Einflussfaktoren korrigiert werden (vgl. SCHMIDT 2019).

Eine schematische Darstellung einer Plattformwaage für ein Bienenvolk zeigt die folgende Abbildung.

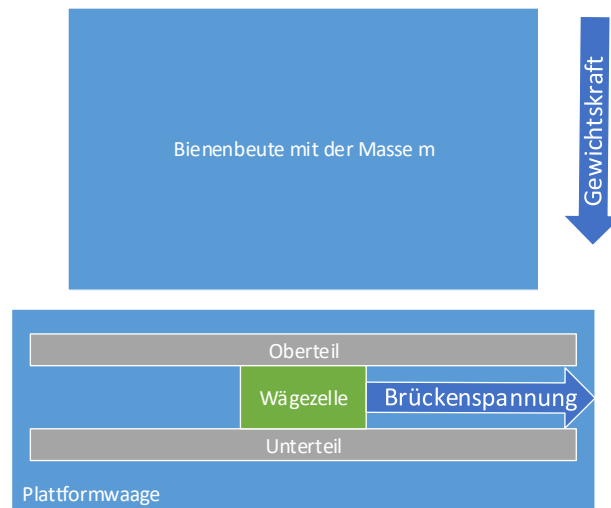


Abbildung 23 - Schema Plattformwaage für ein Bienenvolk

3.2.3.5 Konkretisierung Sendeintervall / Visualisierung der Messergebnisse

Im Kapitel 3.1 wurden die zu erfassenden Kriterien zur Beurteilung von Bienenvölkern ermittelt. Dabei kann festgestellt werden, dass sich die Werte nicht sprunghaft ändern sollten. Die Außentemperatur fällt beispielsweise nicht innerhalb weniger Sekunden um 10 °C. Auch die Schwarmerkennung anhand der Masse sollte in einem größeren Zeitabstand möglich sein. Jede Datensendung verbraucht Energie am LoRaWan Node. Falls Daten in der Cloud verarbeitet werden, fallen möglicherweise auch hier höhere Kosten bei einer höheren Datenmenge an. Deshalb wird der **Sendeintervall auf 15 Minuten** festgelegt und später im Einsatz möglicherweise korrigiert. Es wird eine Schnittstelle zwischen der IoT Plattform und LoRaWan Application Server gefordert, damit die Darstellung der Daten erfolgen kann. Bei der Umsetzung des Formats zur Übertragung der Nutzinformation muss ein Format implementiert werden, welches geringen Overhead erzeugt.

Die unter Kapitel 3.2.3.2 festgelegte Lösung für den LoRaWan Gateway ermöglicht die Nutzung der Software LoRaServer.io. Diese Lösung stellt konfigurierbare Konnektoren für die Google Cloud, Microsoft Azure, AWS von Amazon sowie die kostenlose IoT Plattform ThingsBoard bereit. Zusätzlich können diverse Events (z.B. Join Notifications, Status Notifications) genutzt werden, um eine Web API via http Rest aufzurufen (vgl. LORASERVER-1 2019).

Eine detaillierte Untersuchung dieser IoT Plattformen wird im Rahmen dieser vorliegenden Arbeit nicht durchgeführt, weil sie im Rahmen der vorliegenden Arbeit zu aufwendig ist. Es wird deshalb die **IoT Plattform auf die ThingsBoard Lösung** festgelegt. Damit ist ein OnPremise Installation sowie Cloud Variante möglich. Für die community Lizenz fallen keine Kosten an. Die Lösung bietet eine moderne Website zur Administration, Konfiguration von Regeln und Geräten sowie zur Visualisierung der Daten (vgl. THINGSBOARD-1 2019).

Die erfassten Daten sollen in Diagrammen zusammengefasst in einem Dashboard dargestellt werden. Die Abbildung zeigt das Dashboard „Bienenvölker – aktueller Überblick“ als Wireframe.

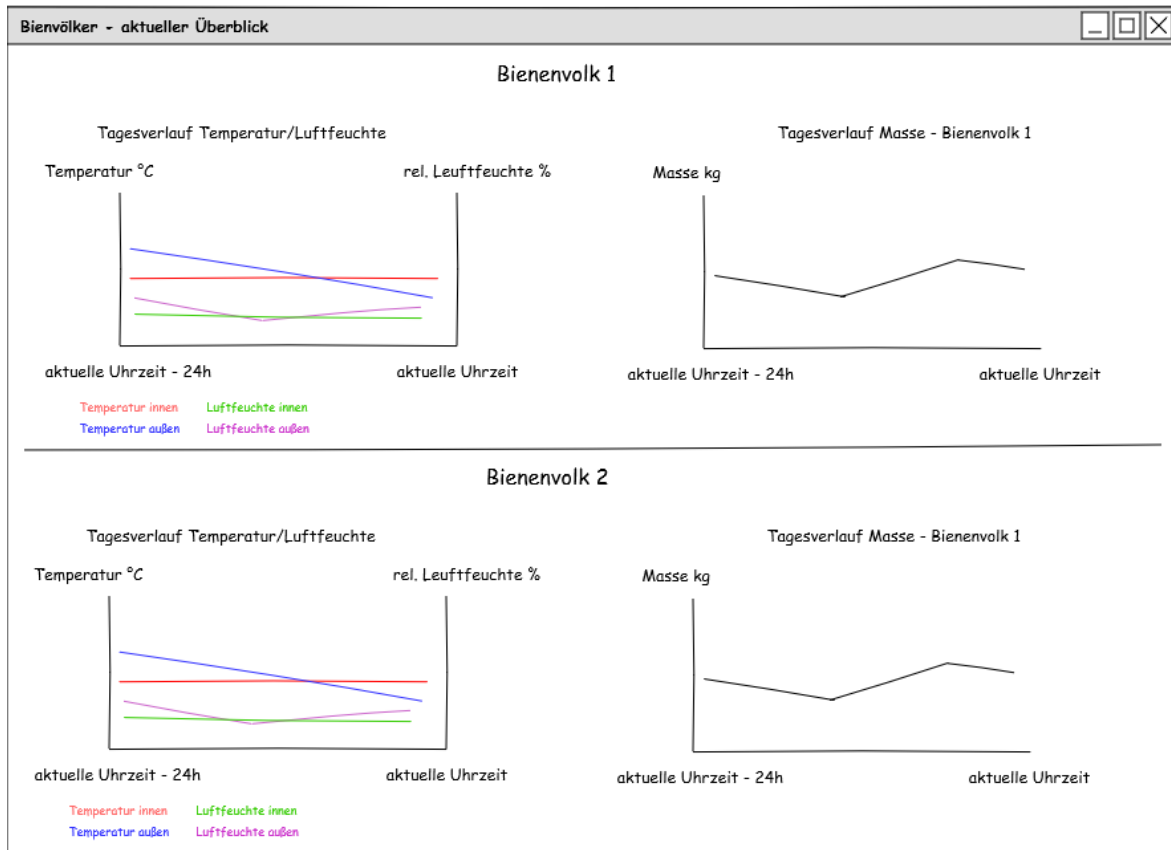
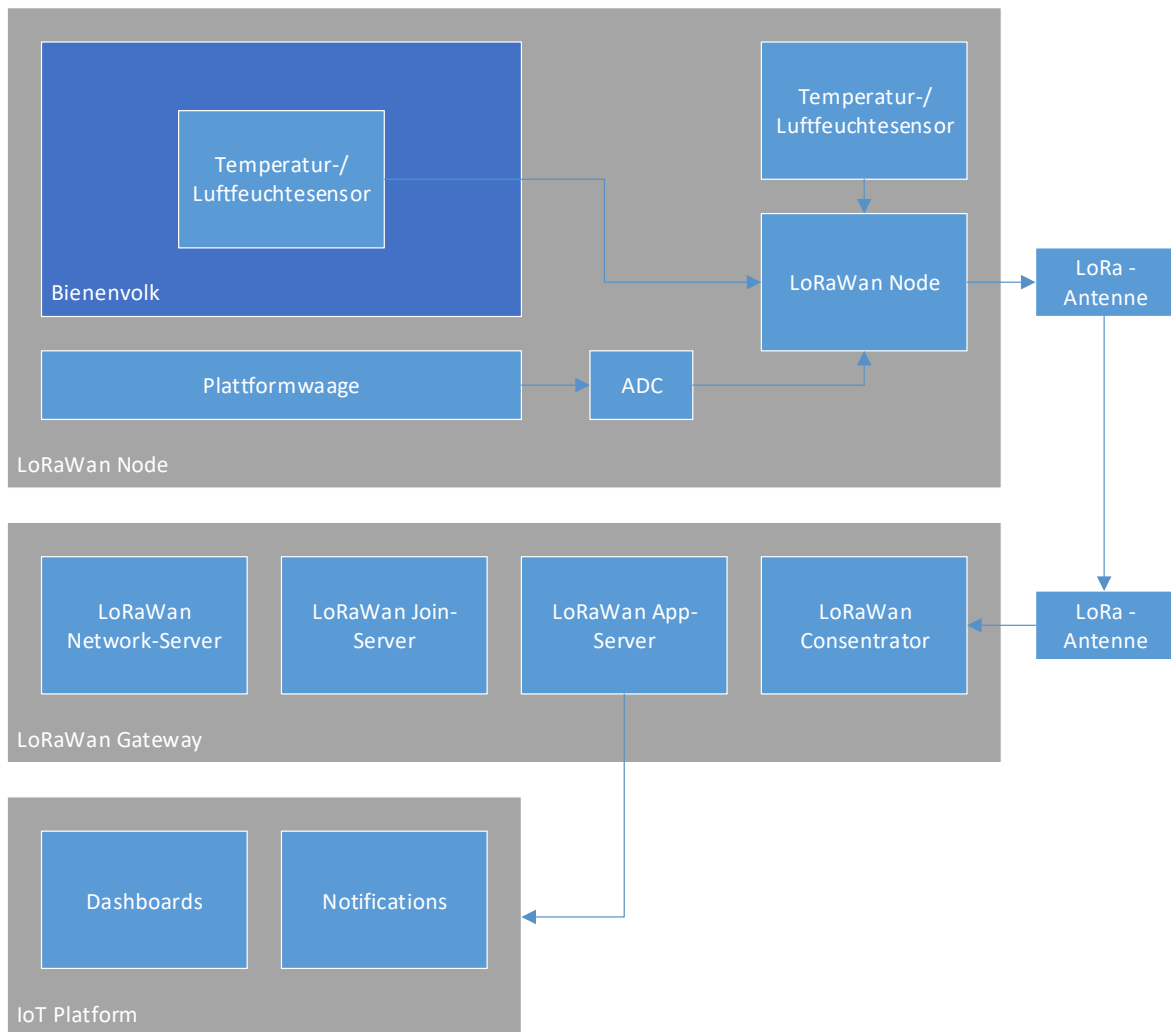


Abbildung 24 - Wireframe Dashboard Bienenvölker - aktueller Überblick

3.2.3.6 Ergebnis der Sollkonzeption / Grobentwurf

Die Gesamtlösung besteht aus mehreren Subsystemen. Das Subsystem LoRaWan Node erfasst die Temperatur in Grad Celsius und Luftfeuchte in Prozent der Umgebung und im Innenraum der Bienenbeute sowie die analoge Ausgangsspannung der Wägezelle in der Plattformwaage. Letztere wird in der Einheit Kilogramm dargestellt. Die Energieversorgung erfolgt über Batterie. Die elektronischen Bauteile, außer die Sensoren, sind in einem Gehäuse untergebracht. Diese erfassten Informationen werden über LoRaWan mit Hilfe einer Antenne an das LoRaWan Gateway in einem festgelegten Intervall gesendet. Die Elektronik Komponenten für das Gateway befinden sich in einem Gehäuse. Es wird über ein externes Netzteil betrieben. Im LoRaWan App-Server werden die Daten aufbereitet und schließlich an die IoT Plattform ThingsBoard übertragen. Für diesen Prototyp sollten alle erforderlichen LoRaWan Backend-Komponenten auf einem System bereitgestellt werden. Folgende Abbildung zeigt das Gesamtsystem.

**Abbildung 25 - Grobentwurf des Gesamtsystems**

4 Umsetzung

4.1 Plattformwaage

Die Plattformwaage hat die Aufgabe die Masse der senkrecht aufgestellten Bienenbeute zu ermitteln. Die folgenden Kapitel beschreiben Realisierung der Plattformwaage.

4.1.1 Versuchsaufbau

Im Kapitel 3.2.3.4 wurde der Einsatz und die Funktionsweise der Wägezelle beschrieben. Am Markt wird die Plattform Wägezelle H30A von dem Anbieter Bosche Wägetechnik (<https://bosche.eu>) angeboten. Sie bietet u.a. Nennlast bis max. 200 kg, kompensierte Eckenlastfehler (bei ungleicher Belastung der Plattform), Genauigkeitsklasse C3 Y=7500, Eichfähigkeit, max. Plattformgröße 400 x 400 mm. Der Preis beträgt ca. 60 Euro. Die Abbildung zeigt die Wägezelle mit Maßen und Anschlussleitungen.

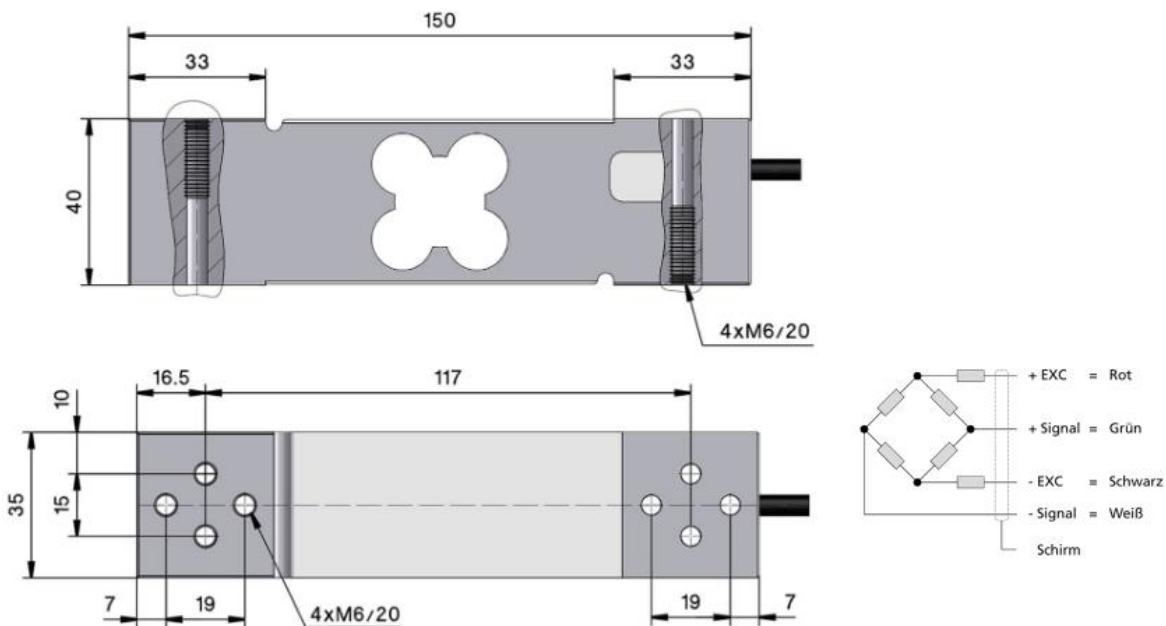


Abbildung 26 - Zeichnung Bosche Wägezelle H30A

Unveränderte Übernahme: Abbildung 26: Zeichnung (<https://www.bosche.eu/waagenkomponenten/waagezellen/plattform-waagezellen/plattform-waagezellen-h30a>, verfügbar am 23.9.2019)

Eine preisgünstige Alternative zur Bosche Wägezelle ist die Realisierung mit vier Half-Bridge-Load-Cells (Typ YZC-161E). Dabei sind vier Stück für ca. 10 Euro erhältlich. Je Zelle ist eine Nennlast von 50 kg angegeben. Beim Einsatz von vier Zellen, welche als Wheatstone Messbrücke verschaltet werden müssen, beträgt die Nennlast damit 200 kg. Die Montage dieser Zellen ist etwas aufwendiger, aber auf Grund des geringen Preises wird dieser Nachteil akzeptiert. Zur Verstärkung und Wandlung der Brückenspannung wird ein ADC des Typs HX-711 eingesetzt. Dieser bietet u.a. 24 bit Wandlung, einstellbare Verstärkung des Eingangssignals, einstellbare Ausgangsdatenrate. Die erforderliche Betriebsspannung beträgt zwischen 2,6 bis 5,5 V (vgl. AVIA 2019, Seite 1 ff.).

Ein erster Aufbau auf dem Steckbrett zeigt die Verschaltung der vier Half-Bridge-Load-Cells, dem HX-711 und dem Arduino MKR WAN 1300.

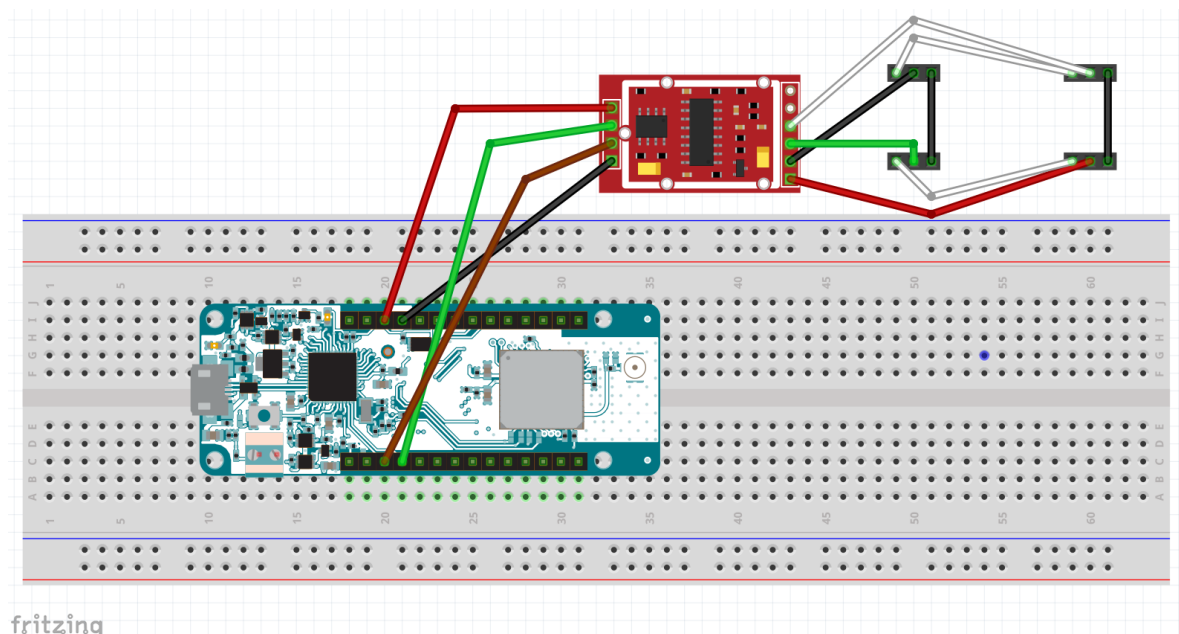


Abbildung 27 - Versuchsaufbau Steckbrett Waage

Die Tabelle zeigt die Verbindung zwischen HX-711 und den nach außen geführten Leitungen der Half-Bridge-Load-Cells.

HX-711	Half-Bridge-Load-Cells
E+	EP
E-	EM
A+	AP

A-	AM
----	----

Tabelle 12 - Verbindung HX-711 und Half-Bridge-Load-Cells

Die Verbindung zwischen Arduino Board und HX-711 nach der folgenden Tabelle herzustellen.

HX-711	Arduino MKR WAN 1300
GND	GND
DT	A2
SCK	A1
VCC	VCC

Tabelle 13 - Verbindung HX-711 und Arduino MKR Wan 1300

Die softwaretechnische Anbindung des HX-711 am Arduino erfolgt mit der Bibliothek HX711. Die Bibliothek ist über die Arduino Entwicklungsumgebung installierbar. Weiterhin ist sie über Github (URL: <https://github.com/bogde/HX711>) abrufbar. Sie bietet u.a. eine leicht verständliche API, Power On/Off Funktionen, Kalibrierfunktionen sowie eine gute Dokumentation. Die Bibliothek findet eine weite Verbreitung in der Community und wird aktuell weiterentwickelt. Die Funktionsweise kann mit einem Programmbeispiel von der Github-Seite überprüft werden. Im Quellcode müssen die verwendeten Pins für DT und SCK codiert werden:

```
// HX711 circuit wiring to arduino mkr 1300
const int LOADCELL_DOUT_PIN = A1;
const int LOADCELL_SCK_PIN = A2;
```

4.1.2 Mechanische Umsetzung

Die Waage besteht aus einem Unter- und Oberteil. Es wurden 15 mm dicke Siebdruckplatte in den Maßen 470 x 470 mm verwendet. Auf dem Unterteil wurden vier Half-Bridge-Load-Cells angeschraubt, dabei wurde mit Unterlegscheiben eine Distanz zur Oberfläche des Unterteils hergestellt, damit eine Belastung (elastische Verformung) möglich ist. Die Anschlussleitungen wurden über den Mittelpunkt der Platte zur Seite verlegt und mit Heißkleber befestigt. Am Oberteil wurden an den Punkten, an denen die Half-Bridge-

Load-Cells auf das Oberteil drücken würden, Holzschrauben mit Distanzmutter angeschraubt. Damit soll ein Verrutschen und damit mögliche Reibungseffekte vermindert werden. Zusätzlich wurden Leisten als Führung am Oberteil angebracht. Damit wird ein vollständiges Verrutschen des Oberteils verhindert. An der Seite wurde eine nach unten geöffnetes Kabeldurchführung vorgesehen, um später an dieser Stelle die Leitungen durchzuführen und das Gehäuse mit weiterer Elektronik zu montieren.

Folgende Abbildung zeigt die Innenseite des Unter- und Oberteils.

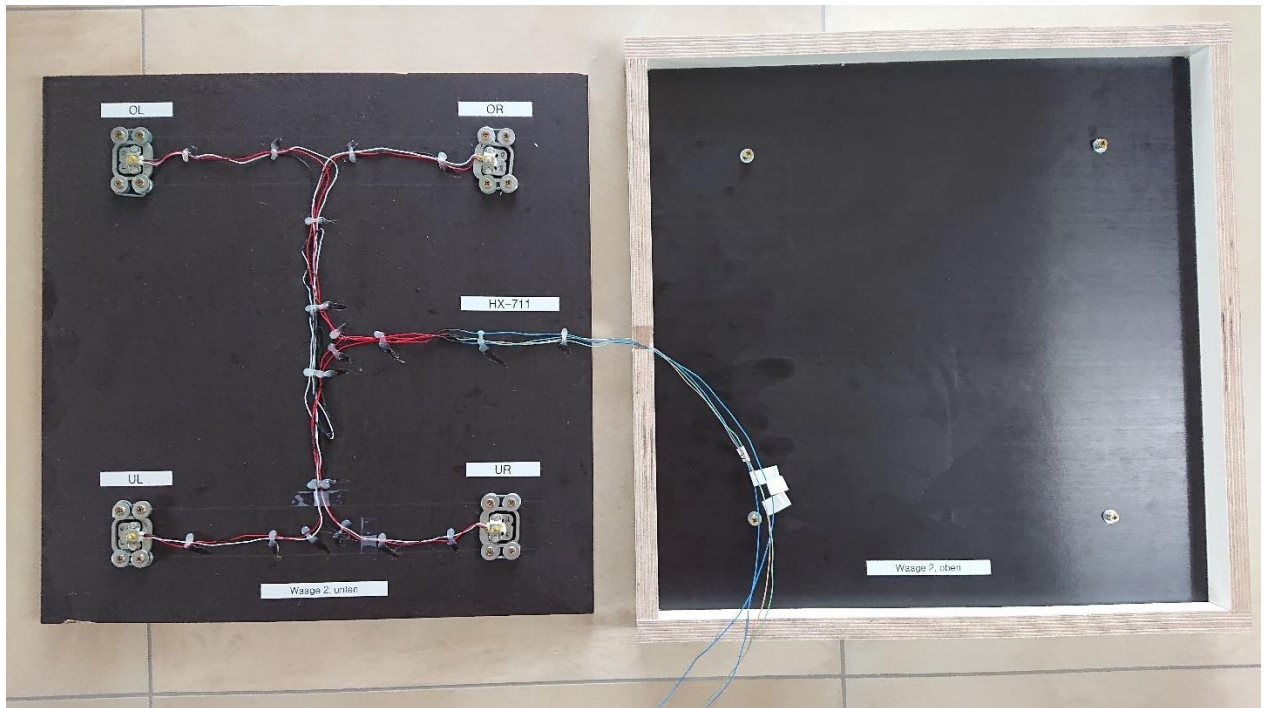


Abbildung 28 - Unter- und Oberteil Plattformwaage

Folgende Tabelle zeigt die Stückliste für eine Plattformwaage.

Position	Beschreibung	Anzahl
1	Siebdruckplatte, 15 mm, 470 mm x 470 mm (Breite x Länge)	2
2	Leiste, 15mm, 500 mm x 40 mm (Länge x Höhe)	2
3	Leiste, 15mm, 470 mm x 40 mm (Länge x Höhe), 1 mit Bohrung für Leitungsöffnung	2
4	Half-Bridge-Load-Cells, je 50 kg	4

5	HX711 Platine mit Steckerstiften zum Auflöten	1
6	Unterlegscheiben je Half-Bridge-Load-Cell 8 Stück unten, 4 Stück oben am Unterteil und 1 am Oberteil	52
7	Holzschrauben zur Befestigung der Half-Bridge-Load-Cells und Auflagen Oberteil, 4 x 40mm	52
8	Holzschrauben zur Befestigung der Leisten, 3,5 x 35mm	12
9	Heißkleber	k.A.

Tabelle 14 - Stückliste Waage

4.1.3 Kalibrierung

Nachdem die Waage vollständig aufgebaut und die Leitungen entsprechend Versuchsaufbau (siehe Kapitel 4.1.1) angeschlossen sind, erfolgt eine initiale Kalibrierung. Mit dem Kalibrierprogramm wird der Kalibrierkoeffizient ermittelt. Das Programm wird auf den Arduino übertragen, dabei darf die Plattformwaage kein zusätzliches Gewicht enthalten. Mit dem Serial Monitor in der Arduino Entwicklungsumgebung wird der ermittelte Wert in Kilogramm sowie der Kalibrierkoeffizient ausgegeben. Die Vorgehensweise zeigt das folgende Ablaufdiagramm.

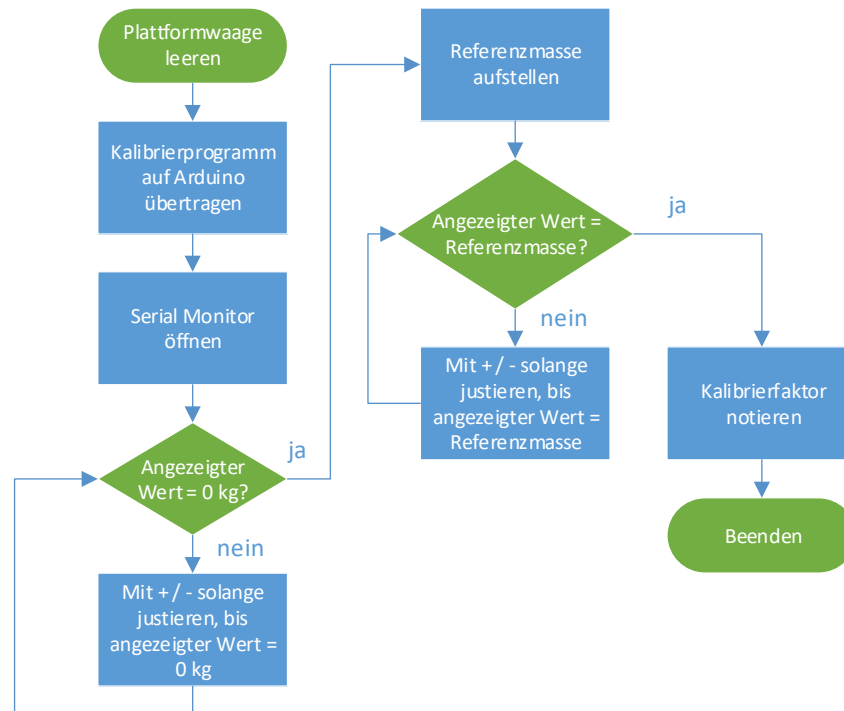


Abbildung 29 - Ablauf Ermitteln Kalibrierkoeffizient

Folgende Abbildung zeigt diesen Vorgang mit einer Referenzmasse von 10 Kilogramm. Dabei wurde der ursprüngliche verwendete Koeffizient von 20050 auf 19980 verändert.

```

// original code taken from here: https://github.com/sparkfun/HX711-Load-Cell-Amplifier/blob/master/firmware/SparkFun_HX711
#include "HX711.h"

// HX711 circuit wiring to arduino mkr 1300
const int LOADCELL_DOUT_PIN = A1;
const int LOADCELL_SCK_PIN = A2;

// calibration settings
float calibration_factor = 20050;

HX711 scale;

void setup() {
  Serial.begin(9600);

  // set hardware pins
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

  // initial set to 0 (without extra weight!)
  scale.set_scale();
  scale.tare();

  // get zero factor
  long zero_factor = scale.read_average();
  Serial.print("Zero factor:");
  Serial.println(zero_factor);
}
  
```

```

COM6
Calibration factor:19970.00
10.10kg
Calibration factor:19980.00
10.09kg
Calibration factor:19980.00
10.09kg
Calibration factor:19990.00
10.09kg
Calibration factor:19990.00
10.09kg
Calibration factor:20000.00
10.09kg
Calibration factor:20000.00
10.09kg
Calibration factor:20010.00
10.09kg
  
```

Abbildung 30 - Kalibrierkoeffizient ermitteln

Für Plattformwaage 1 wurde der Koeffizient **19980** und für die Plattformwaage 2 der Koeffizient **20190** notiert.

Im Kalibrierprogramm sind die Codeausschnitte enthalten, welche im eigentlichen Anwendungsprogramm wiederverwendet werden können, um die Masse des aufgestellten Bienenstocks zu ermitteln. Dabei sind die Kalibrierkoeffizienten als Konstanten in das Anwendungsprogramm zu übernehmen.

4.2 Sensoreinheit

Die Sensoreinheit besteht in einem Fall aus einem Sensor, welcher im Innenbereich der Bienenbeute installiert ist. Im anderen Fall ist ein zusätzlicher Sensor im Außenbereich montiert. Beide Sensoren haben die Aufgabe die Temperatur, die relative Luftfeuchte und den relativen Luftdruck zu erfassen. Die folgenden Kapitel zeigen Auszüge der Umsetzung.

4.2.1 Versuchsaufbau

Der einzusetzende Sensor wurde im Grobentwurf (siehe Kapitel 3.2.3.3) auf den Bosch BME280 festgelegt. Dieser Sensor kann über die SPI oder I²C eingebunden werden. Unter Verwendung von I²C benötigt der Sensor eine eindeutige Slave Adresse. Der Sensor verwendet dabei die Adresse 0x76, wenn das SDO Pin mit dem GND Pin verbunden wird. Wird das SDO Pin mit dem V_{DDIO} Pin verbunden, verwendet der Sensor die Slave Adresse 0x77. Damit können zwei Sensoren gleichzeitig am I²C Bus verwendet werden (vgl. BOSCH 2018, Seite 32).

Im Versuchsaufbau wird der Einsatz des Arduino MKR WAN 1300 und nur einem Sensor aufgezeigt. Dabei wird ein Shield, d.h. Sensor mit Vorbeschaltung und Pins, verwendet. Die folgende Abbildung zeigt die schematische Verbindung zwischen den Komponenten.

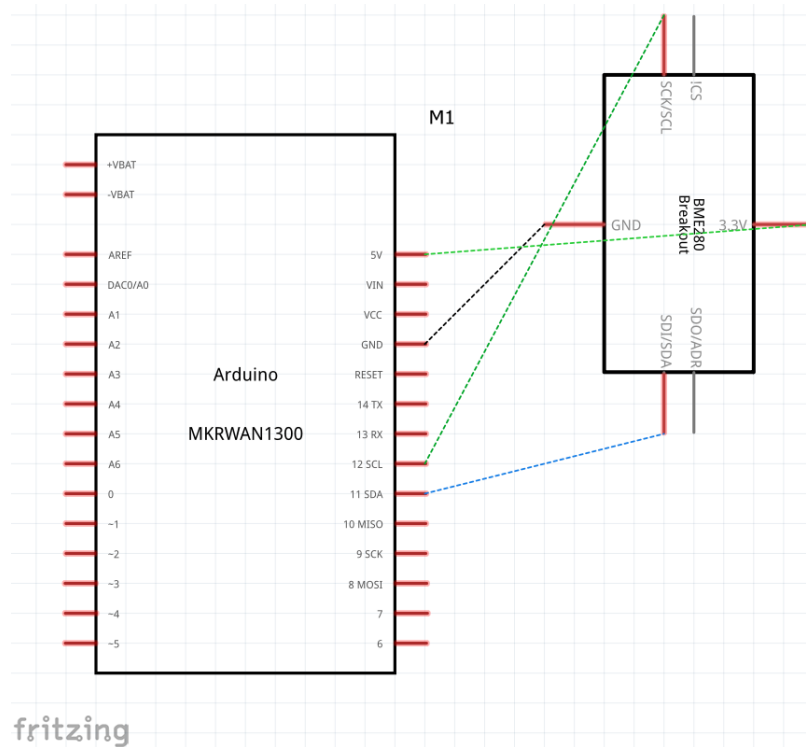


Abbildung 31 - Versuchsaufbau Bosch BME280

4.2.2 Mechanische Umsetzung

Der Sensor wurde in einem Gehäuse montiert. Das Gehäuse wurde mit Öffnungen für die Anschlussleitungen sowie Lufteinlässen versehen. Diese Öffnungen sind so ausgelegt, dass keine Bienen in den Sensor gelangen sollten. Es wurden 2 m lange Flachband-Leitungen verwendet. Diese Baugruppe als Innen- und Außensensor verwendet werden und bietet Schutz gegen Berührung.

Folgende Abbildung zeigt das geschlossene und geöffnete Gehäuse mit Sensor und Verbindungsleitung.

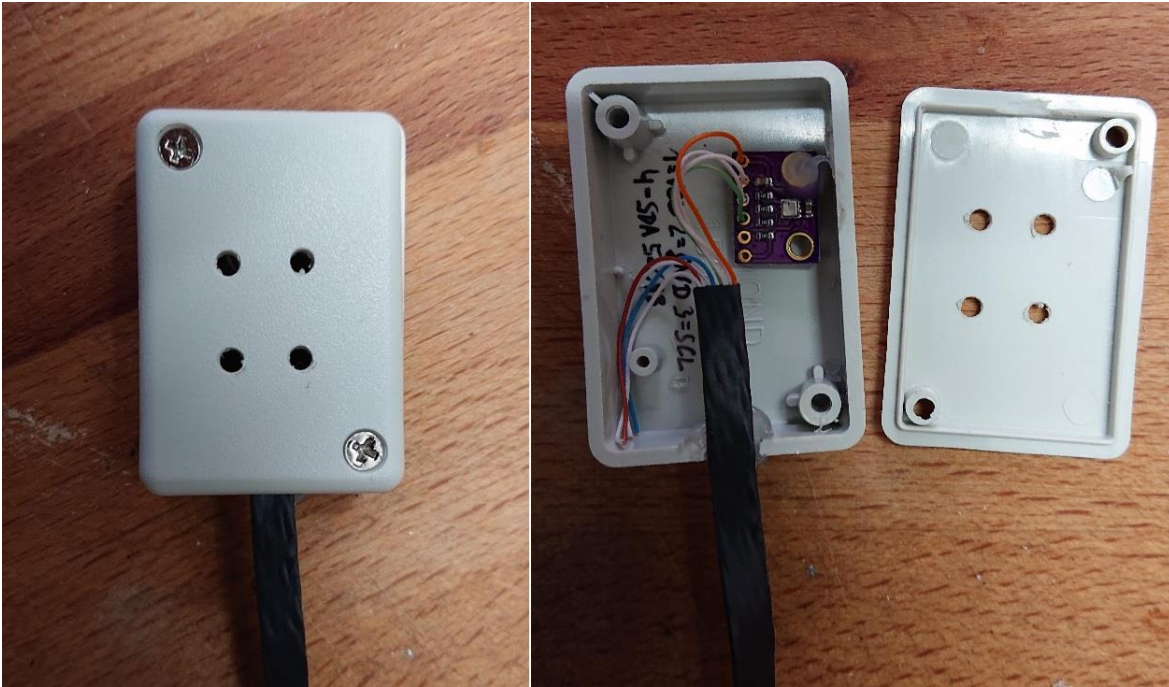


Abbildung 32 - Sensor im Gehäuse

Die folgende Tabelle zeigt die Stückliste für einen Sensor.

Position	Beschreibung	Anzahl
1	Sensor, Bosch BME280 (Breakout), z.B. Sparkfun oder Adafruit	1 bzw. 2
2	Gehäuse, Schalengehäuse Serie 1551, 35 x 50 x 15 mm	1
3	Anschlussleitung, Flachbandleitung	Je 2 m
4	Heißkleber	

Tabelle 15 - Stückliste Sensor

4.2.3 Auszug Softwareimplementierung

Zur softwaretechnischen Anbindung des Sensors gibt es eine Vielzahl von Bibliotheken. Es wurde die „SparkFun BME 280“ Bibliothek über die Arduino Entwicklungsumgebung installiert. Der Quellcode ist auf Github veröffentlicht (URL: https://github.com/sparkfun/SparkFun_BME280_Arduino_Library). Diese Bibliothek bietet u.a. die folgenden Funktionen an:

- Setzen der I²C Slave Adresse für den angeschlossenen Sensor
- Ermitteln der Temperatur
- Ermitteln des Luftdrucks
- Ermitteln der Luftfeuchte
- Ermitteln der ungefähren Höhe über dem Meeresspiegel (wird intern in der Bibliothek über den Luftdruck berechnet)
- Bereitstellen der Energiesparoptionen des Sensors

Relevante Codeausschnitte sind folgend dargestellt.

```
#include <Wire.h>
#include <SparkFunBME280.h>
BME280 bme; // I2C
...
// I2C Adresse ändern
Wire.begin();
bme.setI2CAddress(0x76); // default I2C address
...
// Sensor in Schlafmodus setzen
bme.setMode(MODE_SLEEP);
...
// Sensor aufwecken
bme.setMode(MODE_FORCED)
...
// Messwerte ermitteln
bme.readTempC(); // Temperatur in °C
bme.readFloatPressure(); // Luftdruck in Pa (-> Pa/100 = mbar)
bme.readFloatAltitudeMeters(); // Höhenmeter in m
bme.readFloatHumidity(); // rel. Luftfeuchte in %
```

Die folgende Abbildung zeigt die Messwerte im Serial Monitor der Arduino Entwicklungsumgebung.

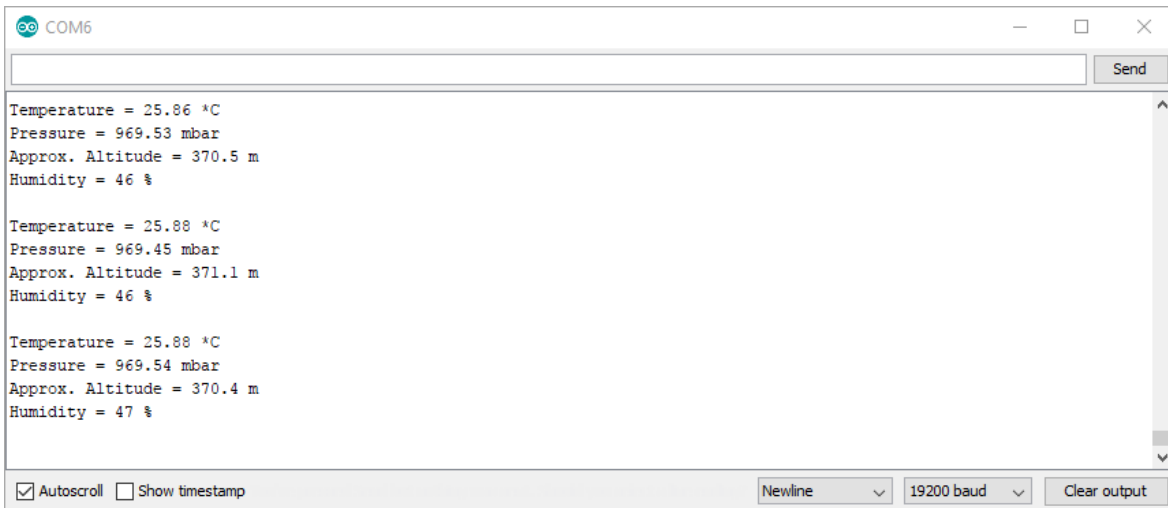


Abbildung 33 - Ausgabe der Messwerte

4.3 LoRaWan Node

Der LoRaWan Node übernimmt u.a. das „Einsammeln“ und Aufbereiten der Messwerte. Die Umsetzung wurde bereits in den Kapiteln 4.1 und 4.2 vorgestellt. Eine weitere Aufgabe des Teilsystems ist das Senden dieser Informationen über LoRa bzw. LoRaWan. In den folgenden Kapiteln wird die Umsetzung auszugweise dargestellt.

4.3.1 Versuchsaufbau

Als Hardware wurde das im Kapitel 3.2.3.1 das Arduino MKR WAN 1300 festgelegt. Es wird eine externe Antenne mit entsprechendem Anschlussadapter benötigt. Diese Antenne wird später am Gehäuse montiert. Die Abbildung zeigt den sehr einfachen Versuchsaufbau auf dem Steckbrett.

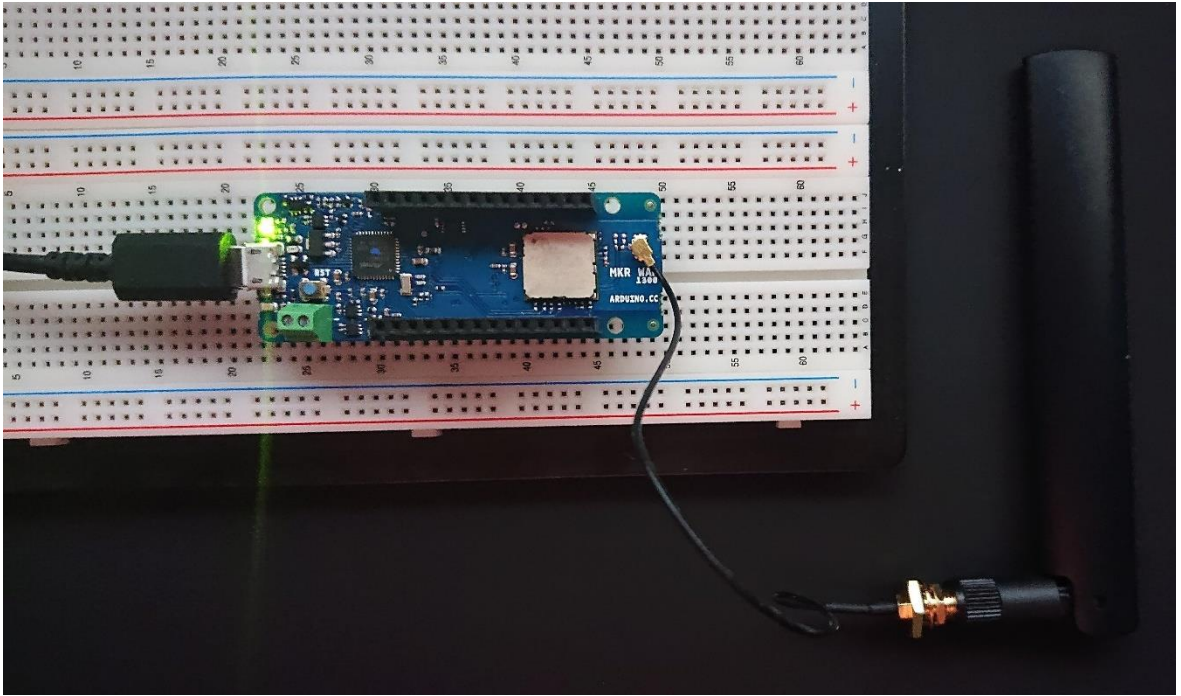


Abbildung 34 - Versuchsaufbau LoRaWan Node

4.3.2 Mechanische Umsetzung

Alle Elektronikkomponenten, außer die Sensoren, wurden in einem Gehäuse montiert. Die Anschlussleitungen der Temperatursensoren und der Plattformwaage wurden in das Gehäuse geführt. Weiterhin erfolgte der Einbau des für die Waage erforderlichen ADC. Die Antenne wurde über eine Buchse am Gehäuse verschraubt. Die Verbindungen zwischen den Komponenten wurden als Löt-, Steck- oder Schraubverbindungen realisiert.

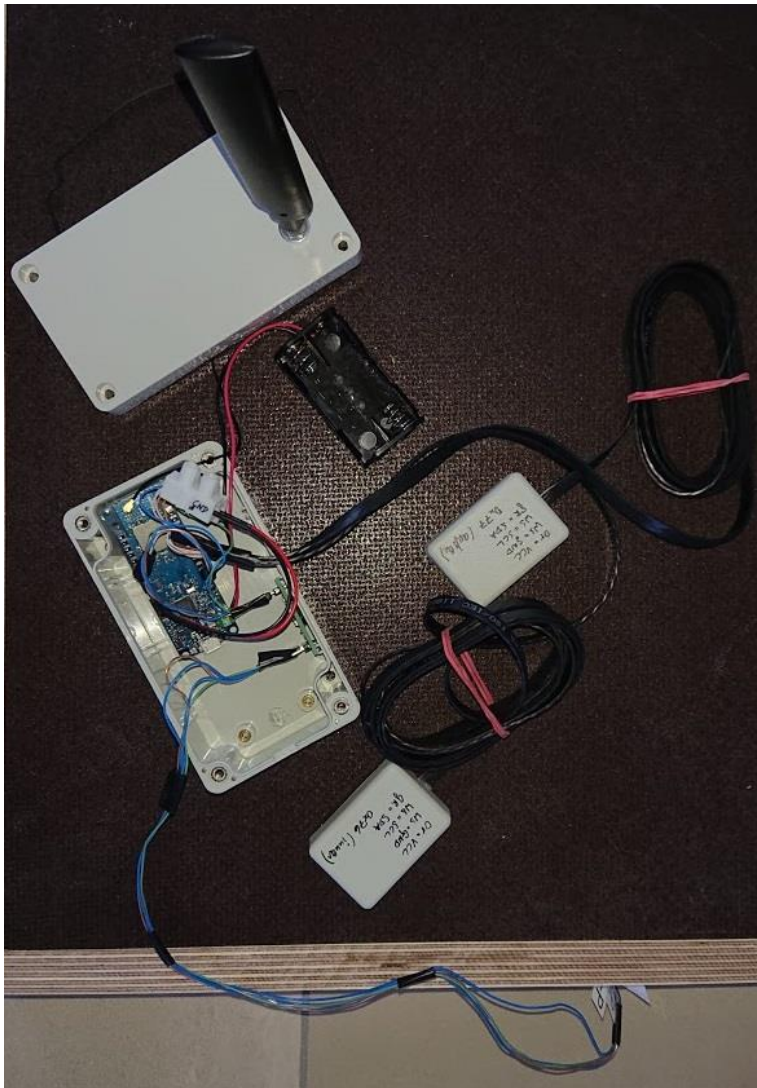


Abbildung 35 - LoRaWAN Node mit Peripherie

Zusammengefasst als Stückliste werden folgende Bauteile für das LoRaWAN Node benötigt.

Position	Beschreibung	Anzahl
1	Arduino MKR WAN 1300	1
2	Batterie AA, 1,5 V	2
3	Batteriehalter für 2 Mignonzellen (AA) mit Anschlusskabel	1
4	SMA Einbaubuchse (für Antenne), Delock, MHF/U.FL	1

5	Antenne, Delock LoRa 868 MHz Antenne SMA Stecker 3 dBi omnidirektional mit Kippgelenk schwarz	1
6	Gehäuse, Kunststoff Serie 1551, 35 x 50 x 15 mm	1

Tabelle 16 - Stückliste LoRaWan Node

4.3.3 Auszug Softwareimplementierung

Für das Arduino MKR1300 WAN Board wird eine Bibliothek von Arduino zur Verfügung gestellt, welche die Funktionen für das LoRa Modem als einfache API bereitstellt. Sie kann über die Arduino Entwicklungsumgebung installiert werden. Der Quellcode ist auf Github veröffentlicht (URL: <https://github.com/arduino-libraries/MKRWAN>). Diese Bibliothek installiert u.a. ein Beispiel „FirstConfiguration“, welches für diesen Versuch etwas umgeschrieben wurde. Damit das Beispiel funktioniert, ist ein LoRaWan Netzwerk (siehe Kapitel 2.3.1) erforderlich.

Folgende Schritte sind nacheinander durchzuführen:

- Programm auf Arduino kopieren und den SerialMonitor in der Arduino Entwicklungsumgebung starten
- Die angezeigte device EUI notieren (wird zur Einrichtung im LoRa App Server benötigt wird)
- Den AppKey vom LoRa App Server in die gleichnamige Variable kopieren
- Die AppEUI Variable wird mit „0000000000000000“ gesetzt
- Im LoRa App Server muss das Gerät als LoRaWan mit der MAC Version 1.0.0 angelegt sein, der Beitritt (Join) erfolgt über OTAA

Folgend sind relevante Codeausschnitte für die Anwendung dargestellt (vgl. ARDUINO-3 2017).

```
// Anlage globaler Variablen
String appEui = "0000000000000000"; // immer dieser Wert
String appKey = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"; // als MSB kopieren

LoRaModem modem; // Zugriff auf die LoRa Bibliothek
modem.begin(EU868); // Regionalparameter konfigurieren und starten (Auswertung über Rückgabewert erforderlich)

modem.deviceEUI(); // deviceEUI ermitteln

modem.joinOTAA(appEui, appKey); // Join über OTAA
```

```

modem.beginPacket(); // Sendevorgang aktivieren

modem.print("Hello World."); // Zeichenkette senden

modem.write("Hello World."); // als Bytes versenden, diverse
Überladungen

modem.endPacket(true); // Sendevorgang beenden, Rückgabe (int) >
0, dann Fehler aufgetreten

```

Die Abbildung zeigt die Ausführung des Programms im SerialMonitor der Arduino Entwicklungsumgebung. Dabei ist das Node dem Netzwerk erfolgreich über OTAA beigetreten (Join) und konnte Daten senden.

```

COM6
19:54:17.836 -> Welcome to MKRWAN1300 first configuration sketch
19:54:17.836 -> Register to your favourite LoRa network and we are ready to go!
19:54:18.681 -> Your module version is: ARD-078 1.1.5
19:54:18.715 -> Your device EUI is: [redacted]
19:54:18.715 -> Try to connect to LoRa gateway via OTAA with key:
19:54:18.715 -> Used APP EUI is:
19:54:18.715 -> 0000000000000000
19:54:18.715 -> Used APP KEY is: [redacted]
19:54:18.715 -> Do you want to go on? Type 'y' or 'Y' (with
19:54:18.715 -> ).
19:54:36.732 -> Starting modem.
19:54:37.568 -> Starting modem fine.
19:54:37.568 -> Connect to gateway.
19:54:44.818 -> Connection fine.
19:54:44.818 -> Send message.
19:54:44.818 -> t0:22.5;p0:0.8
19:54:48.847 -> message sent fine.
19:54:48.847 -> Do you want to go on? Type 'y' or 'Y' (with
19:54:48.847 -> ).

```

Annotations in the image:

- wird aus library ausgelesen
- konstant diesen Wert benutzen
- kopieren aus LoRa AppServer
- Beispieldaten wurden erfolgreich gesendet

Abbildung 36 - Node sendet Daten

4.3.4 Programmablauf

Der umzusetzende Programmablauf für die Software auf dem LoRaWAN Node ist in der Abbildung dargestellt.

3	1	Humidity Sensor	3304 / 104	Relative Luftfeuchte [%]
4	1	GPS Location	3336 / 136	GPS Koordinate (Latitude, Longitude, Altitude); hier: umgenutzt zur Angabe der Höhe von Innen- und Außensensor Altitude = Altitude innen; Longitude = Altitude außen
5	1	Analogue Input	3202 / 2	Einheitenlose Gleitkommazahl, hier genutzt als Masse [kg]
6	2	Analogue Input	3202 / 2	Einheitenlose Gleitkommazahl, hier genutzt als Batteriespannung [V]

Tabelle 17 - anwendungsspezifische Daten nach Cayenne Lpp

Wenn ein Außensensor vorhanden ist, wird die Struktur bestehend aus Position 1 bis 3 wiederholt und der Channel 2 zur Identifikation als Außensensor genutzt. Der vollständige Quellcode ist im Anhang enthalten.

4.4 LoRaWan Gateway

Der LoRaWan Gateway implementiert bei dieser Umsetzung alle erforderlichen LoRa bzw. LoRaWan Komponenten, d.h. LoRa Concentrator, Application Server, Networkserver sowie den Join Server in einem Gerät. Die Umsetzung wird in den folgenden Kapiteln vorgestellt.

4.4.1 Versuchsaufbau

Im Versuchsaufbau wird der Raspberry Pi3 mit dem IMST iC880A LoRa Concentrator über SPI verbunden. Beide Geräte werden über ein Netzteil gespeist, welches einen Strom von 3 A liefern kann. Für eine ordnungsgemäße Funktionsweise wurde der Raspberry Pi3 über Pin 2 und Pin 6 mit Spannung versorgt. Folgende Tabelle zeigt die Verschaltung.

iC880A	Beschreibung	Raspberry Pi3
21	Betriebsspannung 5 V DC (Supply)	2
22	Masse (GND)	6
13	Reset	22
14	SPI CLK	23
15	MISO	21
16	MOSI	19
17	NSS	24
12	GND	9

Tabelle 18 - Verbindung ic880A und Raspberry Pi3

Bei oben beschriebener Verschaltung ist die Verbindung zwischen Pin 12 (iC880A) und 9 (Raspberry Pi3) nicht erforderlich und nur der Vollständigkeit halber aufgeführt (vgl. IMST-1 2018).

Folgende Abbildung zeigt den Versuchsaufbau.

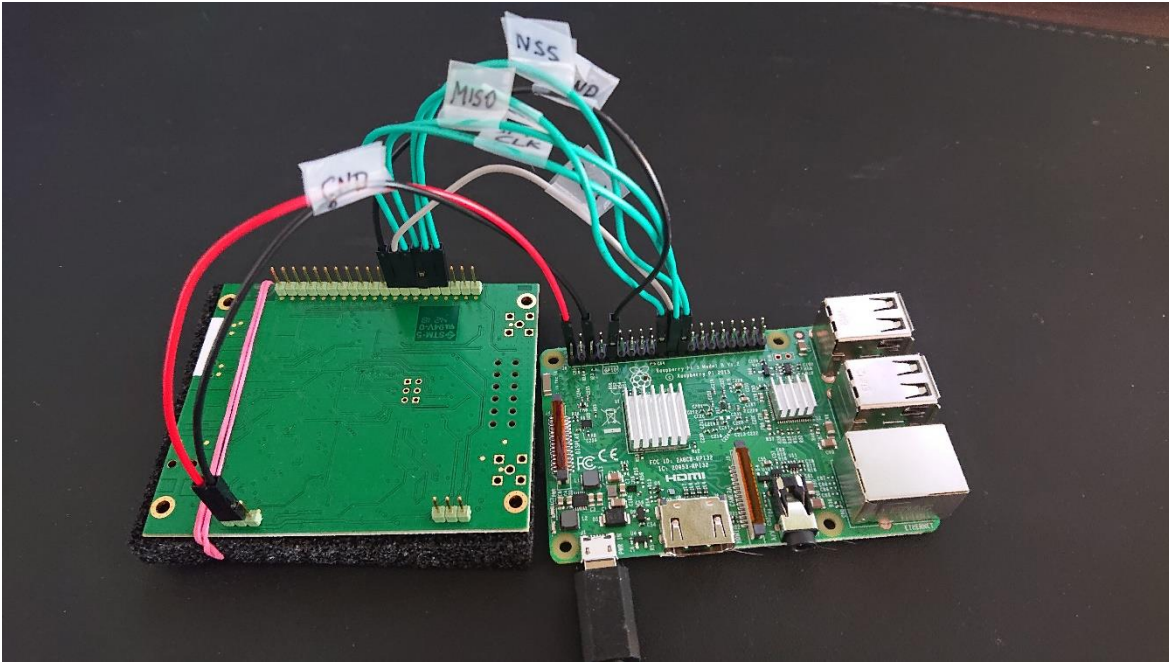


Abbildung 38 - Versuchsaufbau Raspberry Pi und iC880A

4.4.2 Mechanische Umsetzung

Der Raspberry Pi3 und der LoRa Concentrator IMST iC880A wurden in ein Universalgehäuse eingebaut. Im Gehäuse wurden dabei Öffnungen für den Ethernet Port sowie Anschlussbuchse für Spannungsversorgung und Antenne berücksichtigt. Die Abbildung zeigt das geöffnete Gehäuse mit verbauten Komponenten und nach außen geführten Anschlüssen. Die kleine Antenne wurde später durch eine Außenantenne ersetzt.

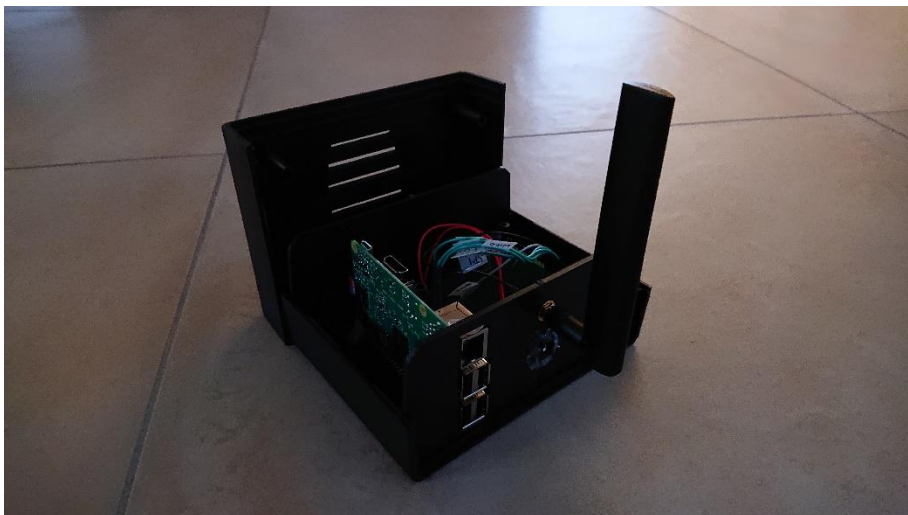


Abbildung 39 - LoRaWan Gateway

Die Installation der Außenantenne ist in der folgenden Abbildung dargestellt. Zur Befestigung wurde ein vorhandener Mast genutzt.



Abbildung 40 - Installation Außenantenne

Folgend ist die Stückliste zur Erstellung eines Gateways mit Außenantenne aufgeführt.

Position	Beschreibung	Anzahl
1	Raspberry PI 3+	1
2	16 GB SD Karte	1
3	LoRa Concentrator Shield, IMST iC880A	1
4	Netzteil, Mean Well, Eingang 240V AC 50 Hz, Ausgang 5V / 3A DC	1
5	Gehäuse, Kunststoff 110 x 150 x 70 mm, schwarz	1
6	SMA Einbaubuchse (für Antenne), Delock, MHF/U.FL	1
7	Anschluss für Stromversorgung, Delock Kupplung, außen 5,5	1

	mm, innen 2,1 mm	
8	Hohlstecker für Stromversorgung, Delock, außen 5,5 mm, innen 2,1 mm	1
9	Außenantenne, Delock LTE UMTS LORA GSM, N Buchse 7-10, Adapter N (male) auf BNC (female) mit Masthalterung	1
10	Anschlussleitungen mit Steckerleisten	
11	BNC Anschlussleitung für Außenantenne	20 m
12	Heißkleber	

Tabelle 19 - Stückliste LoRaWan Gateway

4.4.3 Konfiguration am LoRaWan Gateway

Zur Einrichtung wurden die Softwarekomponenten des LoRaServer Projekts verwendet. Mit dieser open-source Software lassen sich eigene LoRaWan Netzwerk erstellen oder Teile davon in bestehende Infrastrukturen integrieren. LoRaServer kann über verschiedene Varianten genutzt bzw. installiert werden, z.B. als docker image, auf Linux Debian und Ubuntu Distributionen. Für den Raspberry gibt es LoRa Gateway Linux basiertes Betriebssystem. Dabei wird eine Variante als reines Gateway. Dabei ist der Semtech package und LoRa Gateway Bridge vorinstalliert. Eine zweite Variante enthält zusätzlich alle weiteren LoRaWan Komponenten (vgl. LORASERVER-1 2019). Für die Umsetzung wurde die zweite Variante gewählt.

Folgend ist die schrittweise Installation beschrieben.

Das Image (vgl. LORASERVER-1 2019) wurde heruntergeladen und mit dem Werkzeug Etcher von Balena (vgl. ETCHER 2019) auf eine SD Karte installiert. Folgende Abbildung zeigt den erfolgreichen Vorgang.

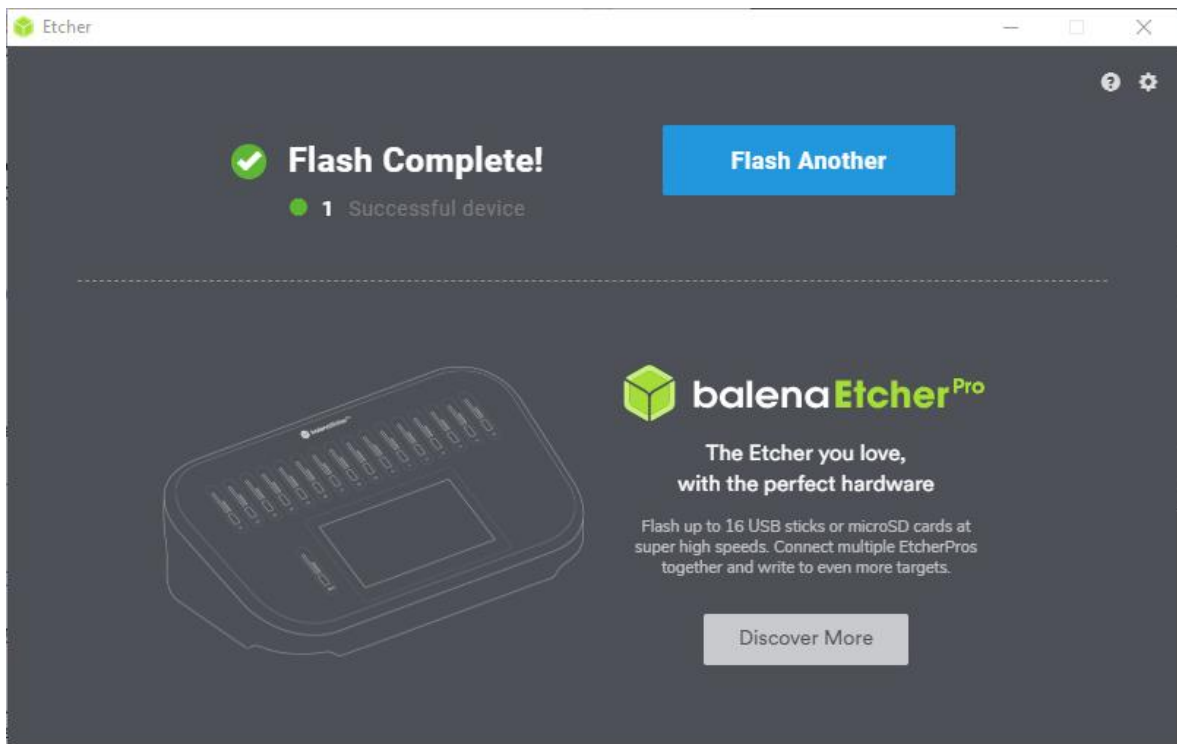
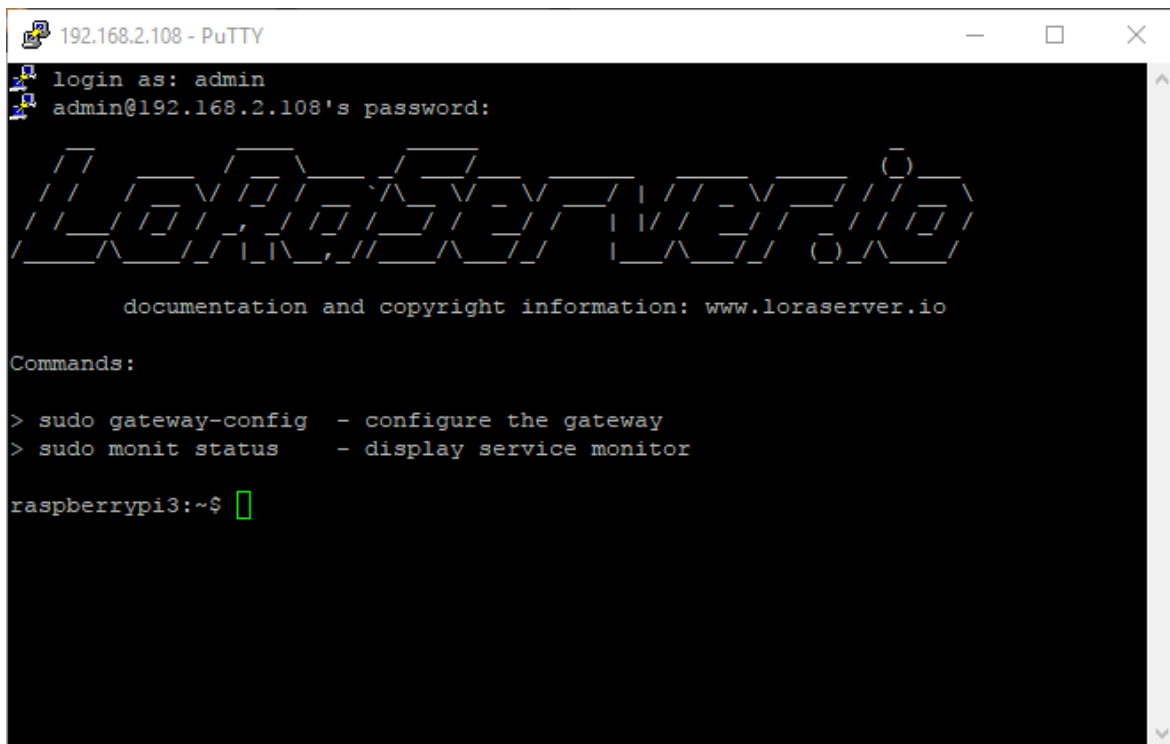


Abbildung 41 - Etcher erfolgreiche Installation

Nach Installation der SD Karte im Raspberry Pi und Bereitstellen der Betriebsspannung bootet der Raspberry Pi und bezieht eine IP Adresse über dynamic host configuration protocol (kurz: DHCP). Über eine SSH Verbindung zum Raspberry, z.B. mit dem Werkzeug Putty, erfolgte die weitere Konfiguration. Dazu sind Kenntnis der zugewiesenen IP Adresse erforderlich. Der Login erfolgt mit Benutzernamen admin und Passwort admin. Folgende Abbildung zeigt den erfolgreichen Login über das Werkzeug Putty.



```
192.168.2.108 - PuTTY
login as: admin
admin@192.168.2.108's password:

LORASERVER

documentation and copyright information: www.loraserver.io

Commands:
> sudo gateway-config - configure the gateway
> sudo monit status - display service monitor

raspberrypi3:~$
```

Abbildung 42 - erfolgreicher Login am Raspberry

Im nächsten Schritt erfolgte der Aufruf des Konfigurationstools durch den Befehl:

```
sudo gateway-config
```

Die folgende Abbildung zeigt gestarteten Konfigurationsdialog.

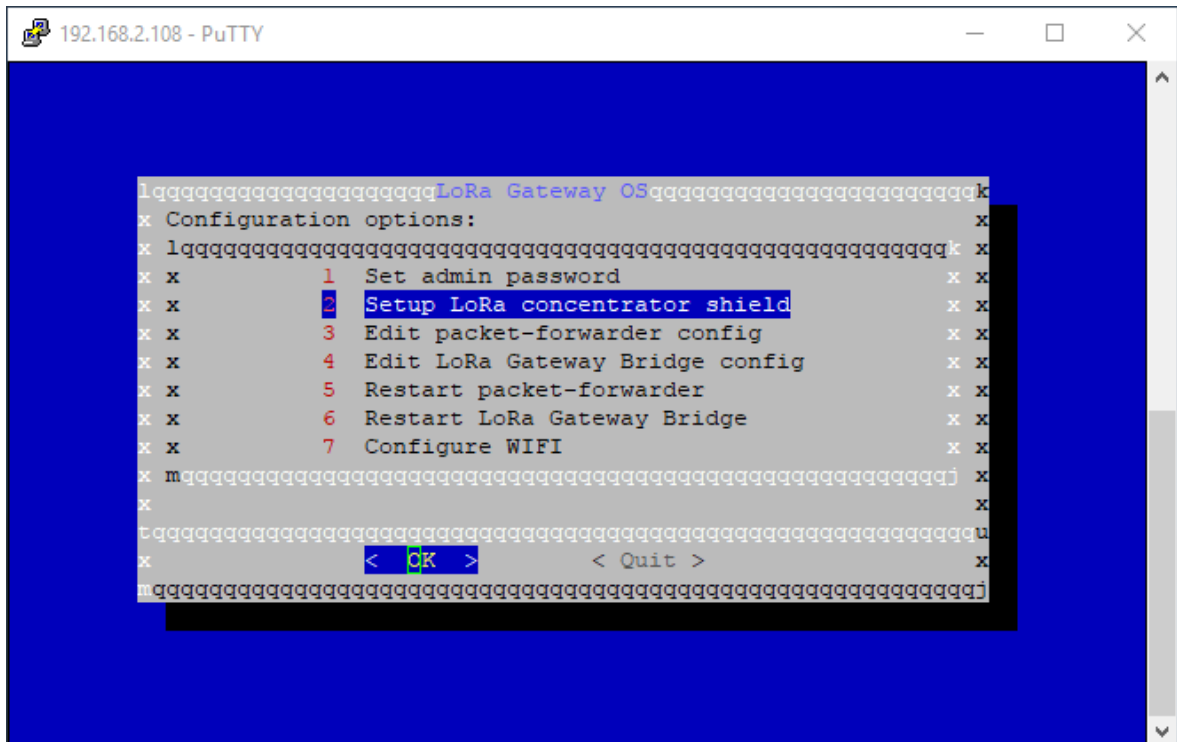


Abbildung 43 - LoRa Gateway Konfiguration

Es erfolgt dabei die Auswahl auf die Option „2“, um den LoRa concentrator einzurichten. In einem Folgefenster wurde der verwendete LoRa concentrator „IMST iC880A“ mit der Option „1“ ausgewählt. Laut der Verschaltung (siehe Kapitel 4.4.1) wird das Reset auf das Pin 22 gesetzt und zum Schluss die Regional EU868 bestätigt. Es folgen einige Erfolgsmeldungen. Danach wird der Konfigurationsdialog verlassen. Mit dem folgenden Befehl erhält man einen detaillierten Bericht der gestarteten Dienste.

```
sudo monit status
```

Während der Umsetzung startete der Prozess „lora-packet-forwarder“ oft nicht. Dies konnte behoben werden indem man den Pin 13 des iC880A mit Masse, Betriebsspannung und wieder auf Masse in Verbindung brachte. Auch die beschriebene Bash Skriptlösung funktioniert nicht (vgl. IMST-3 (2018), Seite 10). Nach Aktualisierung des Images auf die hier beschriebene Version trat dieser Fehler nicht mehr auf.

Vorbereitend für die Konfiguration der LoRaWan Komponenten ist die Gateway-Id zu beziehen. Dazu wird erneut der Konfigurationsdialog gestartet und die die Option „3“ gewählt.

Die weiteren Schritte wurden in der LoRaServer Administrationswebsite durchgeführt, welche im über die IP Adresse und Port 8080 erreichbar ist. Auch hierfür wird der Benutzernamen und Passwort admin/admin verwendet. Folgende Abbildung zeigt die Administrationswebsite.

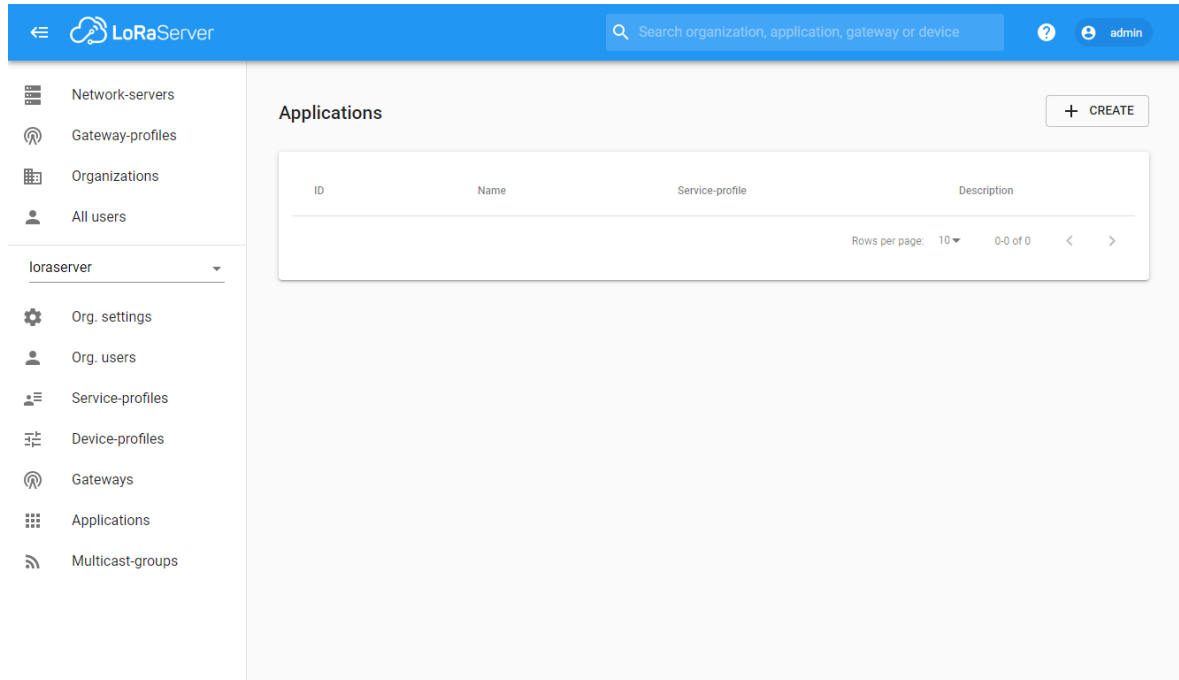


Abbildung 46 - LoRaServer Administrationswebsite

In der folgenden Kurzbeschreibung sind die Konfigurationsschritte zusammengefasst.

- Organisation anlegen und festlegen, dass die Organisation Gateways benutzen darf
- Network Server einrichten, Erreichbarkeit mit localhost und Port 8000, mögliche Änderungen in der Datei „lora-app-server.toml“
- Service-Profil einrichten, dient der Definition von Features für die Organisation und Verknüpfung Organisation mit Network Server
- Application einrichten, Verknüpfung zum Service Profile, Anlage LoRa Devices (Nodes), Anlage von Integratoren (z.B. ThingsBoard)
- Anlage des Gateways unter Verwendung der zuvor notierten Gateway-Id, Angabe des Standorts möglich
- Anlage Device Profile, damit gleichartige Geräte schneller administrierbar (z.B. LoRaWan Version, Join über OTAA oder ABP, Class A, B, C device), wichtig ist hier die Codec Funktion für eigene Datenstrukturen
- Anlage LoRa Devices (Nodes) unter Angabe der zuvor notierten Device EUI und des verwendeten Application Key (siehe Kapitel 0)

Die Konfiguration wurde mit dem LoRa Node und dem Gateway getestet. Dabei versendete der LoRa Node die Testnachricht „HeLoRA world!“ an das Gateway. Folgende Abbil-

ung zeigt die empfangenden LoRa Live Daten Base64 kodiert. Diese können durch einen Base64-Dekoder, z.B. <https://www.base64decode.org/>, in lesbaren Text entschlüsselt werden.

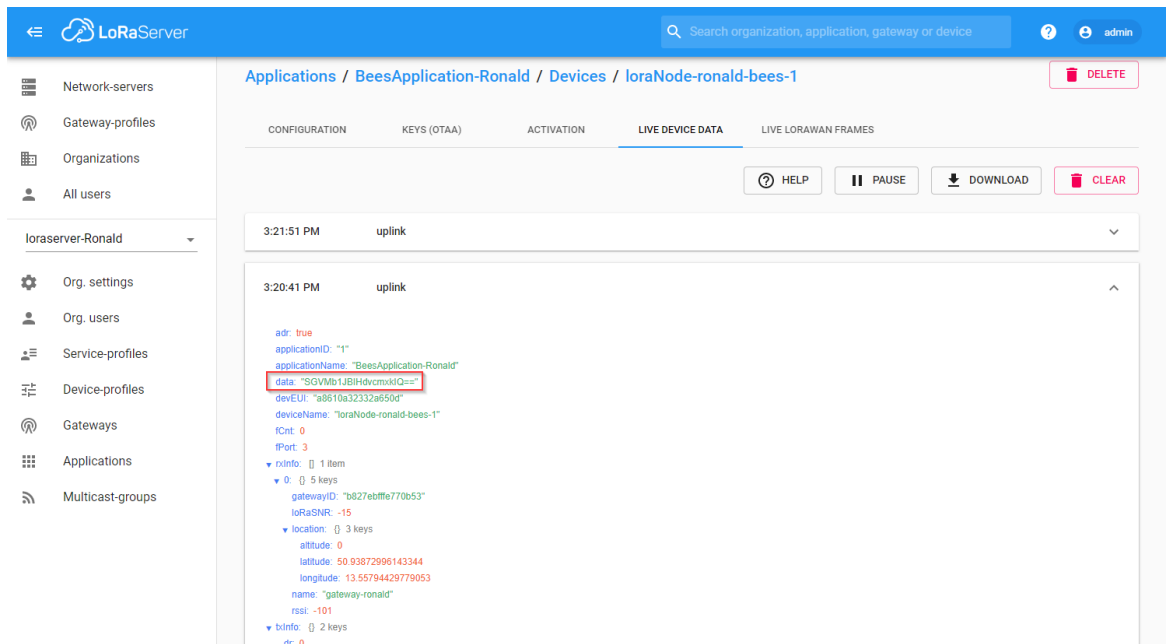


Abbildung 47 - Empfangene LoRaWAN Live Daten vom LoRaWAN Node

4.5 IoT Plattform

In den vorangegangenen Kapiteln wurden die Systeme vorgestellt, welche Informationen aufnehmen, verarbeiten und übertragen. Im letzten Schritt in der Kette der Informationsverarbeitung folgt die Festlegung des Formats der Nutzdaten sowie die Visualisierung in einer IoT Plattform. Im Kapitel 3.2.3.5 wurde ThingsBoard als IoT Plattform festgelegt. Die folgenden Kapitel vermitteln einen kompakten Überblick über ThingsBoard und zeigen Möglichkeiten zum Betrieb der Lösung auf. Weiterhin wird eine beispielhafte Umsetzung für den Prototyp beschrieben und mögliche Formate zur Verarbeitung der Nutzdaten vorgestellt.

4.5.1 Überblick zu ThingsBoard

ThingsBoard ist eine open-source IoT Plattform. Sie bietet die Möglichkeit Kunden und Geräte zu verwalten. Die empfangenden Telemetriedaten der Geräte, können gesammelt, analysiert und visualisiert werden. Weiterhin können Ereignisverarbeitung, z.B. Alarmer, in individuellen Regelketten konfiguriert werden. Durch eine bidirektionale Kommunikation können Geräte gesteuert werden. Die eingehenden Daten können reaktionsschnell und dynamisch in Dashboards dargestellt werden (vgl. THINGSBOARD-2 2019). Um Daten zu persistieren werden Plattform Integratoren ab der ThingsBoard Professional Edition zur Verfügung gestellt. Dabei werden diverse Plattformen wie Amazon Webservices (kurz:

AWS), Azure Event Hub, OPC-UA Server, http und einige unterstützt (vgl. THINGSBOARD-3 2019). ThingsBoard ist als kostenlose Community Edition oder als kostenpflichtige Professional Edition erhältlich. Dabei ist die Community Edition vom Funktionsumfang beschränkt (vgl. THINGSBOARD-1 2019).

4.5.2 Möglichkeiten zum Betrieb

Die Architektur von ThingsBoard ermöglicht einen Betrieb in der Cloud ebenso wie on-premise. Darüber hinaus ist ein Clusterbetrieb möglich. Dabei existiert eine Vielzahl von identischen ThingsBoard Servern. Wenn ein ThingsBoard Server ausfällt, wird die Verarbeitung sofort auf einen anderen Server übertragen. Es Koordinator, welcher im Ausfall die Gesamtverfügbarkeit beeinflusst. Die ThingsBoard Plattform wurde als Microservice Architektur umgesetzt. Dabei ist es möglich alle erforderlichen Komponenten auf einem Server bereitzustellen (Monolith) oder die eine Verteilung vorzunehmen (Microservices) (vgl. THINGSBOARD-4 2019).

Zur Installation wird eine Vielzahl von Möglichkeiten unterstützt. So ist eine Installation auf Linux Derivaten (z.B. Ubuntu, CentOS, redhat), Windows basierten Systemen sowie dem Raspberry Pi3 möglich. Weiterhin wird eine containerbasierte Installation mit Docker und eine große Auswahl an Cloud Anbietern unterstützt. Für erste Schritte kann ein kostenloser Zugang zur Live Demo nach Registrierung genutzt werden (vgl. THINGSBOARD-5 2019).

4.5.3 Auszug Konfiguration

Für den Prototyp wurde auf eine individuelle Installation verzichtet und die Option der Live Demo für die Community Edition gewählt. Alle Konfigurationsschritte für ThingsBoard wurden in der bereitgestellten, modernen Administrationswebsite durchgeführt. Um Daten im ThingsBoard zu verarbeiten ist zuerst ein Gerät erforderlich. Die folgende Abbildung zeigt einen Ausschnitt bereits erstellter ThingsBoard Geräte.

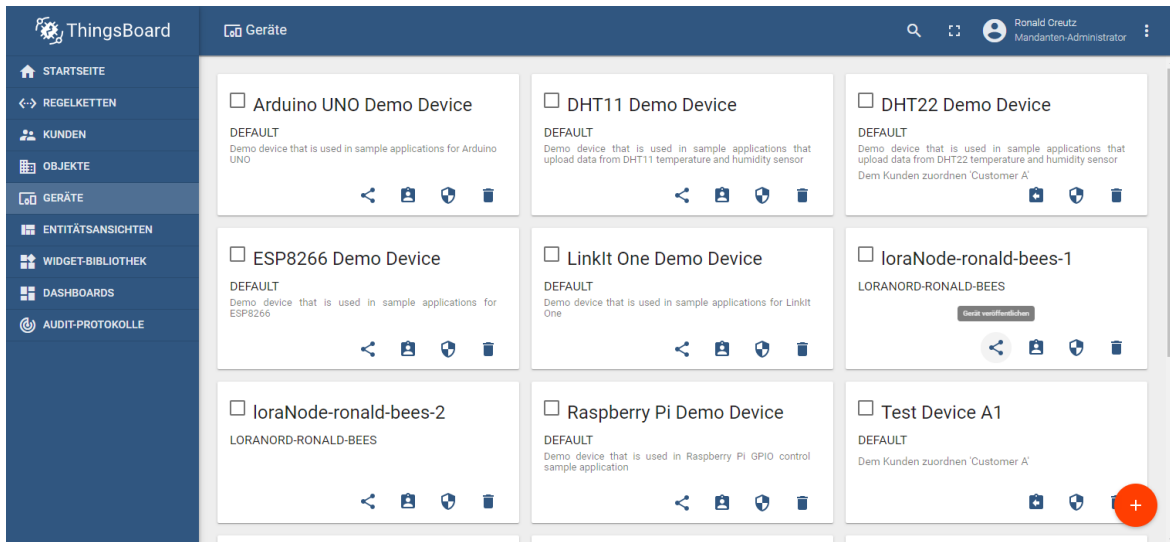


Abbildung 48 - ThingsBoard Geräteübersicht

Für die Verbindung zwischen LoRa Wan Application Server und dem ThingsBoard wird ein Access Token benötigt. Dieser wurde in der ThingsBoard Website generiert und notiert.

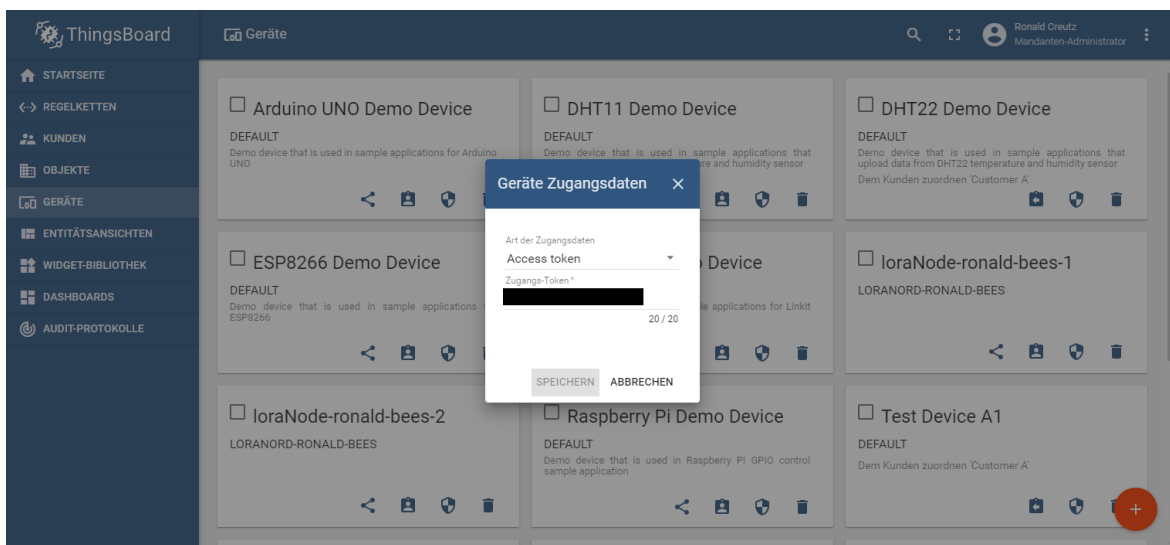


Abbildung 49 - ThingBoard Gerät anlegen

Damit der LoRa Wan Application Server Daten an das ThingsBoard senden kann, wurden die folgenden Konfigurationsschritte in der LoRaServer Administrationswebsite (siehe Kapitel 0) durchgeführt. Da ThingsBoard seit der Version 3.0 des LoRaServers unterstützt wird, gibt es dazu den Typ ThingsBoard.io Integrator. Dieser wurde der zuvor erstellten Application hinzugefügt. Wie folgende Abbildung zeigt, wurde die Url für die Live Demo der ThingsBoard IoT Plattform eingetragen.

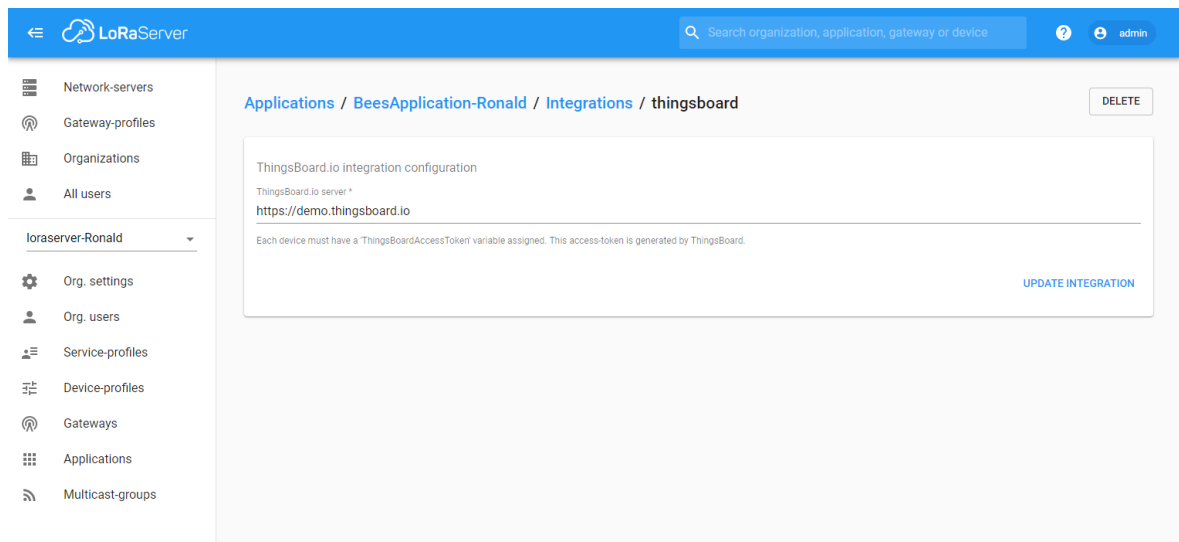


Abbildung 50 - Konfiguration ThingsBoard Integrator im LoRaServer

Im nächsten Schritt wurde der zuvor generierte Access Token für das jeweilige Device eingetragen. Dazu erfolgte die Anlage einer Variablen mit dem Namen „ThingsBoardAccessToken“ mit dem Wert des kopierten Access Tokens.

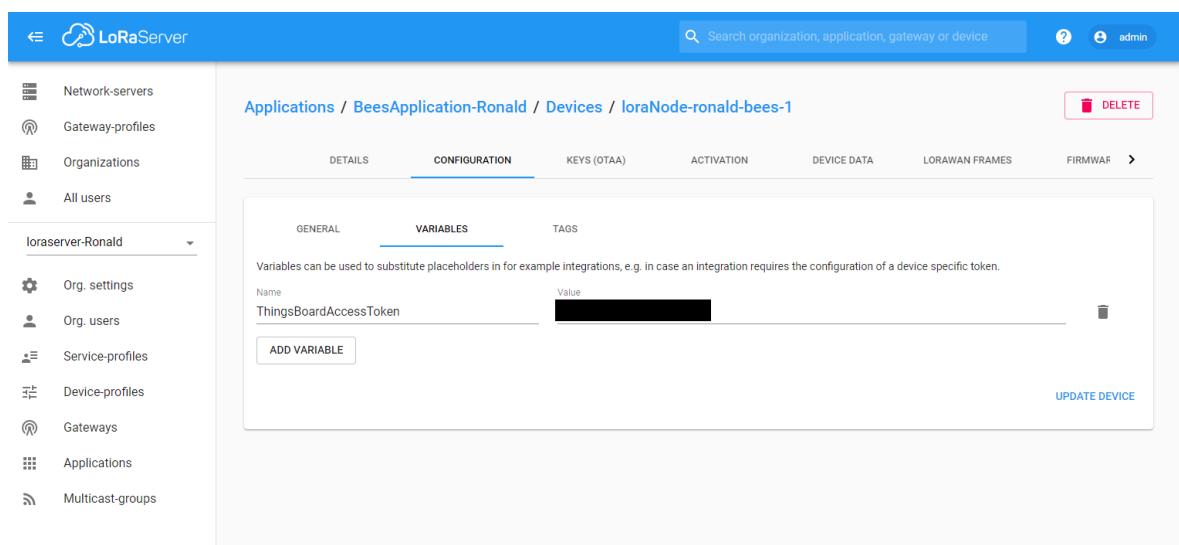


Abbildung 51 - ThingsBoard Konfiguration Access Token im LoRaServer

In der Informationsverarbeitungskette sind an dieser Stelle die Nutzdaten (Payload) extrahiert und können nun in ein Format überführt werden, damit eine Vorverarbeitung schon auf dem LoRa Server möglich ist. Im LoRaServer erfolgt dazu die Einstellung im Device-profile mit drei möglichen Optionen. Mit None erfolgt eine reine Weiterleitung ohne Vorverarbeitung. Mit der Wahl auf das Cayenne Low Power Payload (kurz: Cayenne LPP) Protokoll werden die Nutzdaten auf dem LoRaServer in das bzw. aus diesem Format überführt.

Das Cayenne LPP Protokoll ist eine Datenstruktur für low power applications im IoT Bereich mit dem Ziel der Reduzierung der zu übertragenden Nutzdaten. Dies erfolgt durch

Standardisierung von Sensor- und Datentypen. Als kurzer Überblick dient das folgende Beispiel zur Übertragung eines Messergebnisses von einem Temperatursensor. Folgende Byte-Sequenz ist dabei in den Nutzdaten enthalten: 0x03 0x67 0x01 0x10. 0x03 ist der Kanal, welcher als Identifizierungsmerkmal genutzt werden kann, z.B. Temperatursensor an Beute Nummer 3. 0x67 beschreibt den Datentyp für eine Temperatur mit der Einheit °C. 0x01 und 0x10 entspricht in der dezimalen Darstellung dem Wert 272 und somit lautet das Messergebnis 27,2 °C (vgl. CAYENNE-LPP 2019).

Weiterhin können individuelle Funktionen mit JavaScript erstellt werden. Folgende Abbildung zeigt die Möglichkeit der Dekodierung in ein JavaScript Object für ein individuelles Format.

Device-profiles / deviceprofile-ronald-bees

GENERAL JOIN (OTAA / ABP) CLASS-B CLASS-C **CODEC**

Payload codec

Custom JavaScript codec functions

By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you.

```

1 // Decode decodes an array of bytes into an object.
2 // - fPort contains the LoRaWAN fPort number
3 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
4 // The function must return an object, e.g. {"temperature": 22.5}
5 function Decode(fPort, bytes) {
6   var contentString = "";
7
8   // convert array to string (here I am expect a json formatted string)
9   for (var i = 0; i < bytes.length; ++i) {
10    contentString += String.fromCharCode(bytes[i]);
11  }
12
13  var obj = {};
14  // now contentString should be equal to: key1:value1;key2:value2
15  var keyValueArray = contentString.split(";") //[key1:value1][key2:value2]
16  //for (const pairItem of keyValueArray) {
17  for (var i = 0; i < keyValueArray.length; ++i) {
18    var ar = keyValueArray[i].split(":"); // [key1][value1][key2][value2]
19
20    for (var n = 0; n < ar.length; n = n + 2) {

```

The function must have the signature function Decode(fPort, bytes) and must return an object. LoRa App Server will convert this object to JSON.

Abbildung 52 - custom codec function LoRaServer

Das vereinfachte Schema der Verarbeitung von Nutzdaten zeigt folgende Abbildung. Dabei erfolgt die Aufbereitung der Nutzdaten für die ThingsBoard IoT Plattform auf dem LoRaServer.

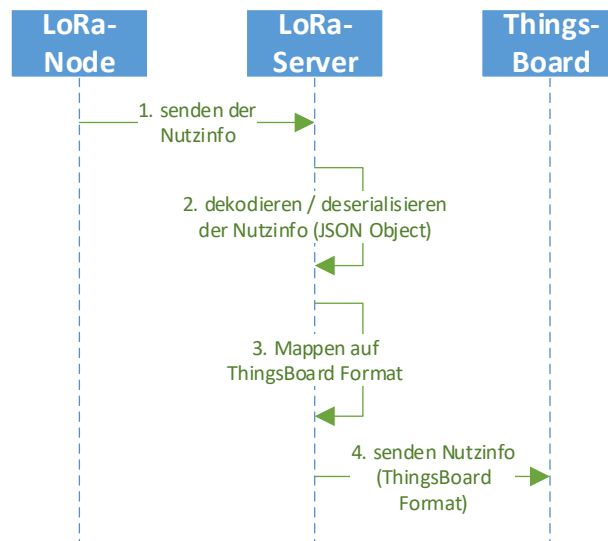


Abbildung 53 - Verarbeitung der Nutzdaten (LoRaServer)

Für eine Validierung der Konfigurationsschritte wurden Daten vom LoRa Node gesendet. Diese Daten wurde zunächst auf dem LoRaServer aufgezeichnet. Im zweiten Schritt wurden die Daten im ThingsBoard kontrolliert. Folgende Abbildung zeigt empfangene Beispieldaten mit einem individuellen Format (Datum 25.9.2019) sowie mit dem Cayenne LPP Protokoll (Datum 26.9.2019).

Das Screenshot zeigt die ThingsBoard IoT Plattform. Die linke Seite zeigt die Navigationsleiste mit den Optionen: STARTSEITE, REGELKETTEN, KUNDEN, OBJEKTE, GERÄTE, ENTITÄTSANSICHTEN, WIDGET-BIBLIOTHEK, DASHBOARDS und AUDIT-PROTOKOLLE. Die rechte Seite zeigt die Detailsansicht für das Gerät 'LORANODE-RONALD-BEES-2'. Die 'Neueste Telemetrie' sind wie folgt dargestellt:

Datum der letzten Aktualisierung	Schlüssel	Wert
2019-09-26 23:18:28	data_barometer_2	1073.2
2019-09-25 21:16:18	data_p0	0.8
2019-09-25 21:16:18	data_t0	22.5
2019-09-26 23:18:28	data_temperature_sensor_1	22.5
2019-09-26 23:18:28	data_temperature_sensor_2	23.5

Abbildung 54 - Beispiel Telemetrie Daten in ThingsBoard

Die Konvertierung der Nutzdaten in Schlüssel-Wert-Paar hat dabei der ThingsBoard Integrator auf dem LoRaServer vorgenommen. In der ThingsBoard IoT Plattform ist damit keine Konvertierung der Datenstruktur erforderlich. Die hier angezeigten Daten können nun in einem Dashboard verwendet, um sie entsprechend zu visualisieren. Dazu steht

eine Vielzahl von Graphen mit umfangreichen Konfigurationsmöglichkeiten zur Verfügung. Folgende Abbildung zeigt empfangene Informationen aufbereitet in drei Diagrammen sowie die Anzeige der aktuellen Spannung, welche von der Batterie bereitgestellt wird.

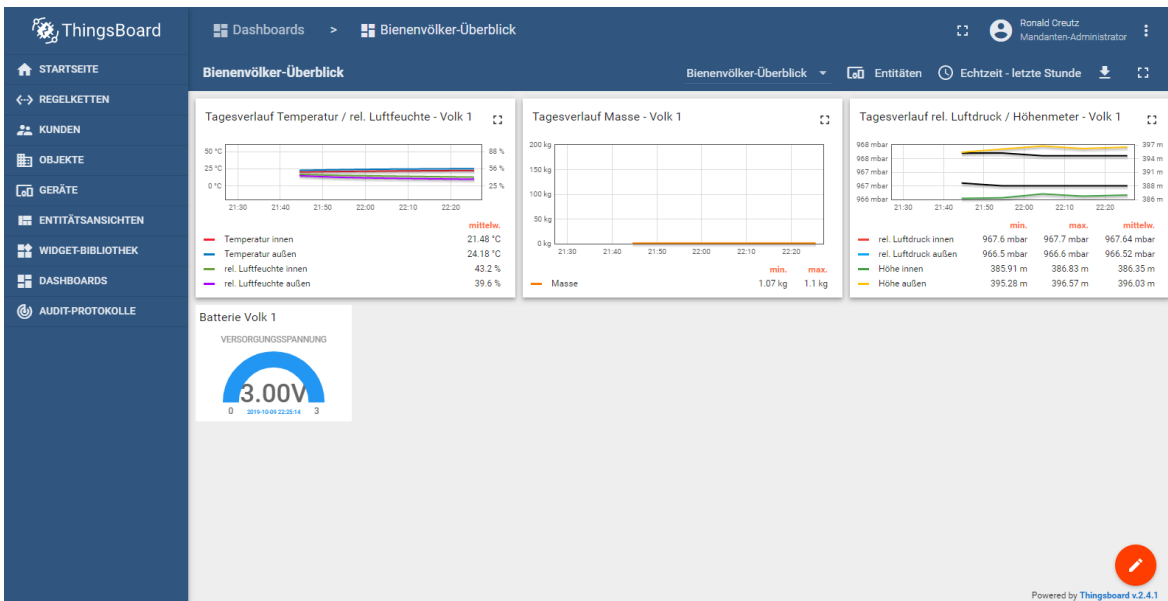


Abbildung 55 – erstes Dashboard in ThingsBoard

4.6 Inbetriebnahme Gesamtsystem

Bei der Inbetriebnahme werden alle Teilsystem zusammengesetzt. Die Inbetriebnahme erfolgte dabei in zwei Stufen. Als Testaufbau und als Aufbau am Referenzbienenstock.

4.6.1 Testaufbau

Der Testaufbau erfolgt in unmittelbare Nähe zum LoRaWan Gateway. Es erfolgte eine Datensendung im verkürzten Sendeintervall von 30 Sekunden und im Debugging Modus im Arduino Studio am LoRaWan Node. Ein Test der Funktionsweise erfolgte durch Definition geeigneter Kontrollpunkte im Gesamtsystem. Folgende Abbildung zeigt das Gesamtsystem mit Kontrollpunkten.

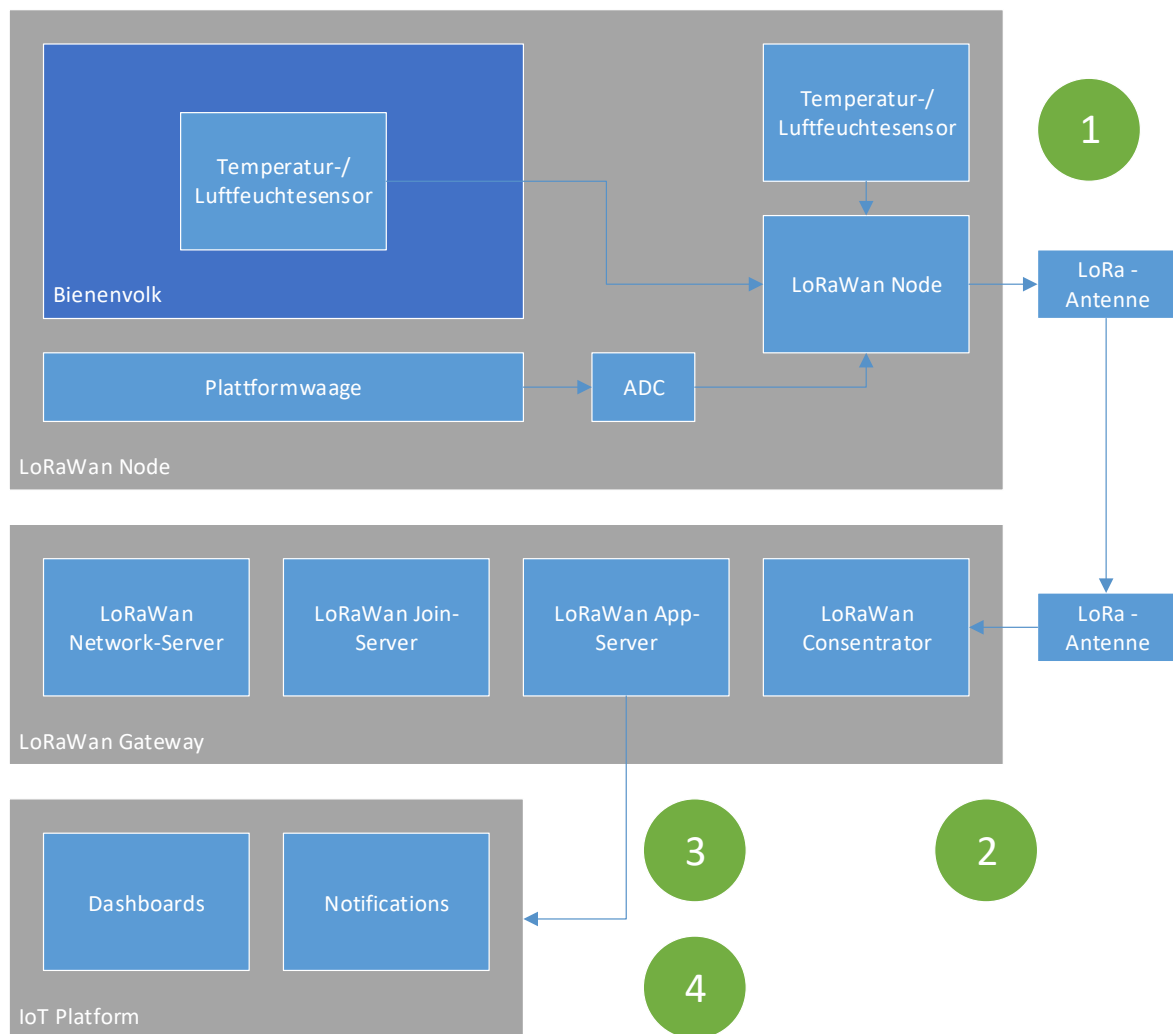


Abbildung 56 - Gesamtsystem mit Kontrollpunkten

Kontrollpunkt 1 ist das Arduino Studio im Serial Monitor. Alle eingehenden Messwerte der Sensoren sowie der Waage sowie weitere Debug-Informationen werden im Serial Monitor ausgegeben und können überprüft werden. In der Administrationswebsite des LoRaServer befindet sich der **Kontrollpunkt 2**. Dabei können eingehende Join Requests des LoRaWAN Nodes sowie der Datensendungen kontrolliert werden. Darüber erhält man Informationen über die Signalstärke oder gesendete bzw. empfangene Daten (kodiert). Im **Kontrollpunkt 3** wurde ein zusätzlicher http Integrator in der LoRaServer Administrationswebsite konfiguriert (vgl. LORASERVER-1 2019). Als Endpunkt wurde ein temporär verfügbarer Testendpunkt bei <https://hookbin.com> erstellt. Damit besteht die Möglichkeit die ausgehenden Daten vom LoRaWAN AppServer zu kontrollieren (siehe Kapitel 4.5.3). Darüber kann beispielsweise die De- und Kodierungsfunktion geprüft werden. **Kontrollpunkt 4** können unter Gerätedetails in der ThingsBoard Administrationswebsite die neusten Telemetriedaten und Client-Eigenschaften angezeigt werden.

4.6.2 Referenzbienenstock

Zur Inbetriebnahme wurde ein kleines Bienenvolk mit einer Zarge ausgewählt, welches im Mai gebildet wurde (Ableger). Zum Zeitpunkt der Inbetriebnahme nach dem Sommer sind für kleine Völker die in Kapitel 3.1 ermittelten Informationen besonders wichtig, um bei Bedarf das Volk vor dem Verenden zu retten.

Die folgende Abbildung zeigt die Installation des Innenraumsensors. Dabei wurde der Sensor mittig auf die Rähmchen positioniert. Die Gehäuseöffnung für den Lufteinlass zeigt nach unten. Dort befindet sich im Winter die Bienentraube.



Abbildung 57 - Positionierung Innenraumsensor

Die Magazinbeute wurde auf die Plattformwaage gesetzt. Die Leisten wurden mit einem Lack gestrichen, um sie vor eindringendem Wasser zu schützen.



Abbildung 58 - Waage mit Magazinbeute (Vorderansicht)

An der Seite ist das Gehäuse mit den übrigen Komponenten sowie der Außensensor montiert.



Abbildung 59 - Waage mit Magazinbeute (Seitenansicht)

Bei der Inbetriebnahme wurden folgende LoRa Parameter aus der LoRaServer Administrationswebsite für empfangene Daten (Uplink Data) vom LoRaWan Node ausgelesen:

SF = 12; rate Code = 4/5; Bandwidth = 125 kHz; SNR 7,2 db; RSSI -106 dbm

Es wird der SF 12 verwendet und damit entsteht die maximale ToA. Der verwendete rate code von 4 bit ist die maximal mögliche Fehlerkorrektur. Das empfangene Signal liegt mit einem SNR von 7,2 db über dem Störsignal (siehe auch Kapitel 2.2.2). Ist der Wert für SNR größer +5 db, so wird das Signal nahezu fehlerfrei interpretiert (vgl. SEMTECH-4 2019).

Der Received Signal Strength Indication Wert (kurz: RSSI) beträgt für LoRa gewöhnlich bis zu -120 dBm. Damit ist auch der erreichte RSSI Wert im geforderten Bereich (vgl. SEMTECH-4 2019). Er gibt ebenfalls Auskunft über die Signalqualität.

Das Dashboard in der Thingsboard IoT Plattform zeigt die empfangenen Daten von dem einzargigen Bienenvolk.

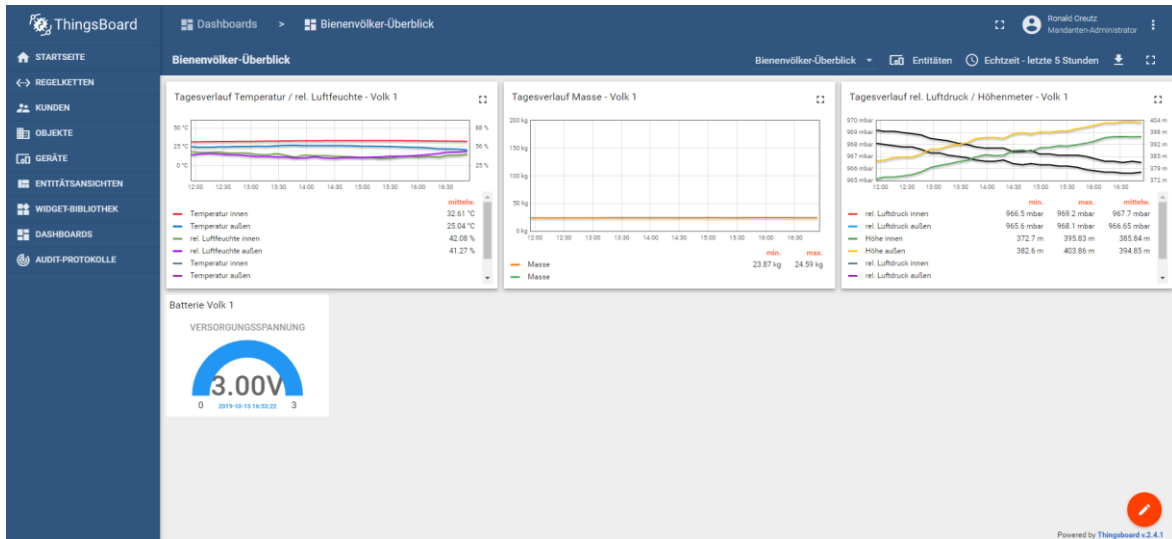


Abbildung 60 - Bienenvölker Überblick

4.7 Software-Werkzeuge in der Übersicht

Zur softwaretechnischen Konzeption und Implementierung wurden folgende Werkzeuge eingesetzt.

Name	Beschreibung	Hersteller / Bezugsquelle
Microsoft Visio 2016	Umfangreiche Software, um bspw. UML Diagramme oder Architekturskizzen zu erstellen.	Microsoft

Arduino Software	Entwicklungsumgebung für eine Vielzahl von Arduino Entwicklungsboards.	Arduino, https://www.arduino.cc/en/main/software
Fritzing	Anwendung zur Erstellung von Hardwareaufbauten (virtuell) mit Steckboards und nachträglicher Erstellung von Leiterplattenlayouts.	https://fritzing.org/download/
Putty	SSH Client Werkzeug.	https://www.putty.org/
Notepad++	Editor zur Bearbeitung von diversen Dateiformaten (z.B. XML).	https://notepad-plus-plus.org/
Hookbin.com	Freier Service, um http Requests anzuzeigen	https://hookbin.com/

Tabelle 20 - Softwarewerkzeuge zur Konzeption / Implementierung

5 Zusammenfassung

Als Zusammenfassung der Arbeit werden in den folgenden Kapiteln die Erkenntnisse zusammengefasst und ein Ausblick auf eine Weiterentwicklung gegeben.

5.1 Erkenntnisse

In der Arbeit wurden Kriterien ermittelt, um Bienenvölker zu beurteilen. Diese liefern wichtige Erkenntnisse, um die Bienenhaltung zum einen effizient und zum anderen schonender für die Bienen zu machen. Eine kosteneffiziente Lösung als Werkzeug wurde im Rahmen eines Prototyps konzeptioniert und realisiert.

Zur drahtlosen Datenübertragung wurde die LoRa/LoRaWan Technologie genutzt. Die dazu erforderlichen Hard- und Softwarekomponenten wurden ausgewählt und erfolgreich erprobt. Alle eingesetzten Softwarepakete für LoRaWan sind dabei open-source. Die Installation eines autarken Netzwerkes wurde aufgezeigt. Zusammenfassend kann festgestellt werden, dass der Einstieg in die LoRa und LoRaWan Technologie durch erhältliche LoRa Boards sowie freier Software bzw. Bibliotheken relativ einfach ist. Nur die LoRa Spezifikation, welches als physikalische Schicht genutzt wird, ist dabei proprietär. Gegenüber anderen, vollständig geschlossenen LPWan Technologien, wie SixFox, ist die LoRaWan Spezifikation offen. Dadurch entstehen möglicherweise Vorteile bei der Verbreitung dieser Technologie.

Die eingesetzte IoT Plattform, ThingsBoard in der kostenfreien Community Edition, bietet neben umfangreichen Möglichkeiten für den professionellen Betrieb auch eine Menge Funktionen. So können aussagefähige Dashboards oder Workflows intuitiv und sehr schnell erstellt werden.

Allerdings konnten im Rahmen dieser Arbeit keine Aussagen auf den Energiebedarf des LoRaWan Nodes getroffen werden. Die Entfernung zwischen Node und Gateway lag bei ca. 800 m und ist damit im LoRa Kontext relativ gering. Deshalb konnten keine Erkenntnisse in Bezug auf praktisch umsetzbare Reichweiten gewonnen werden.

Bei der Inbetriebnahme wurde weiterhin festgestellt, dass die verwendeten Antennen sowie Antennenleitungen maßgeblichen Einfluss auf die Signalqualität haben können. Dies wurde in der vorliegenden Arbeit nicht betrachtet.

5.2 Ausblick

Der realisierte Prototyp bzw. Teile davon können zur Verifizierung weiterer Anwendungsfälle branchenübergreifend genutzt werden. Es ist auch mögliche ähnliche oder neue Anwendungsfälle zu entwickeln und zu erproben. Die LoRa/LoRaWan Technologie findet Einsatzbereiche beispielsweise in der Agrarwirtschaft, Flotten- oder Gebäudemanagement. Insbesondere das lizenzfreie Frequenzband, die hohe Reichweite und der Einsatz in batteriebetriebene Anwendungen lassen das Einsatzgebiet nahezu grenzenlos erscheinen.

Die Erkenntnisse während der Umsetzung und des Betriebs können zur Spezifikation für die Entwicklung eines Produktes dienen. Auf Grund der Kosteneffizienz wäre eine Beschreibung als Do-it-yourself Projekt ebenso möglich. Die verwendeten Baugruppen, insbesondere Wägezellen, sind hinsichtlich Genauigkeit und Langlebigkeit zu untersuchen. Die Auswahl der Antennentechnik ist nicht zu vernachlässigen und sollte den Standort Bedingungen entsprechend konzipiert werden.

Literatur

- ADAFRUIT-1 Adfruit Industries, LLC (2019): Adafruit Metro 328 (<https://www.adafruit.com/product/2488>, verfügbar am 23.9.2019)
- ADAFRUIT-2 Adfruit Industries, LLC (2019): Adafruit RFM 95 Lora Radio Transceiver (<https://www.adafruit.com/product/3078>, verfügbar am 23.9.2019)
- AOSONG Aosong Guangzhou Electronics Co. Ltd (2019): Temperature and Humidity Module DHT11 Product Manual (http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001405544-da-01-en-TEMP_UND_FEUCHTESENSOR_DHT11.pdf, verfügbar am 23.9.2019)
- ARDUINO-1 Arduino (2019): Arduino MKR WAN 1300 (LoRa Connectivity) (<https://store.arduino.cc/mkr-wan-1300>, verfügbar am 23.9.2019)
- ARDUINO-2 Arduino (2019): Arduino Uno Rev3 (<https://store.arduino.cc/arduino-uno-rev3>, verfügbar am 23.9.2019)
- ARDUINO-3 Arduino (2017): MKR WAN library (<https://www.arduino.cc/en/Reference/MKRWAN>, verfügbar am 25.9.2019)
- ARMBRUSTER Armbruster, Sabine (2014): Das Bienen-Praxisbuch, 1. Auflage, BLV Buchverlag, München

- AVIA AVIA Semiconductor (2019): 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales (https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf, verfügbar am 23.9.2019)
- BMEL Bundesministerium für Ernährung und Landwirtschaft (BMEL): Bienen, Unverzichtbar für Natur und Erzeugung (https://www.bmel.de/SharedDocs/Downloads/Broschueren/Biene_n.pdf?__blob=publicationFile, verfügbar am 6.8.2019)
- BMI Bundesministerium des Innern Referat O1: Handbuch für Organisationsuntersuchungen und Personalbedarfsermittlung Bearbeitungsstand Februar 2018 (https://www.orghandbuch.de/OHB/DE/ohb_pdf.pdf?__blob=publicationFile&v=28, verfügbar am 8.9.2019)
- BOSCH Bosch Sensortec (2018): BME280 – Data sheet (https://ae-bst.resource.bosch.com/media_tech/media/datasheets/BST-BME280-DS002.pdf, verfügbar am 23.9.2019)
- BRELL Brell, Claus Prof. Dr. (2019): Der Körperwärme des Biens auf der Spur In: bienen & natur. Jg. 3 (Heft 2/2019), S. 18 f.
- CAYENNE-LPP Cayenne (2019): Cayenne Low Power Payload (<https://developers.mydevices.com/cayenne/docs/lora/>, verfügbar 28.9.2019)
- DIB Deutscher Imkerbund e.V., Mitgliedsverbände des D.I.B. (<https://deutscherimkerbund.de/171-Die-Imker-Landesverbaende>, verfügbar am 9.8.2019)
- DRAGINO-1 Dragino (2019): LoRa Shield for Arduino (<https://www.dragino.com/products/module/item/102-lora-shield.html>, verfügbar am 23.9.2019)

- DRAGINO-2 Dragino (2019): LoRa Shield for Arduino
(<https://www.dragino.com/products/lora-lorawan-gateway/item/140-lq308.html>), verfügbar am 23.9.2019)
- ETCHER Balena (2019): Flash OS images to SD cards & USB drives, safely and easily (<https://www.balena.io/etcher/>), verfügbar am 31.7.2019)
- FELDERHOFF Felderhoff, Rainer (1982): Elektrische Messtechnik, 4. Auflage, Carl Hanser Verlag, München
- GETTERT Gettert, Lieselotte (1998): Mein Bienenjahr – ein Arbeitskalender für den Imker, 2. Auflage, Ulmer, Stuttgart
- IMST-1 IMST GmbH (2018): WiMOD Lite Gateway QuickStartGuide Version 1.5 (https://www.wireless-solutions.de/downloads/Evaluation-Tools/LoRa_Lite_Gateway/WiMOD_LiteGateway_QuickStartGuide_V1_5.pdf), verfügbar am 23.9.2019)
- IMST-2 IMST GmbH (2018): WiMOD iC880A Datasheet Version 1.1 (https://www.wireless-solutions.de/downloads/Radio-Modules/iC880A/iC880A_Datasheet_V1_1.pdf), verfügbar 23.9.2019)
- IMST-3 IMST GmbH (2018): iC880A-SPI Quickstart Guide (https://wireless-solutions.de/downloads/Radio-Modules/iC880A/iC880A-SPI_QuickStartGuide.pdf), verfügbar 31.7.2019)
- KAUPERT Kaupert, Ursula (2019): Kein Platz für kleine Tiere? In: bienen & natur. Jg. 3 (Heft 6/2019), S. 22 f.
- LIU Liu, Thomas (2019): Digital relativ humidity & temperature sensor AM2303/DTH22 (<https://cdn->

- shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf, verfügbar am 23.9.2019)
- LORAALLI- ANCE-1 LoRaWAN™ Alliance (2015): LoRaWAN™ What is it? A technical overview of LoRa and LoRaWAN Version 1.0 (<https://loro-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>), verfügbar am 25.8.2019)
- LORAALLI- ANCE-2 LoRaWAN™ Alliance (2017): LoRaWAN™ Backend Interfaces 1.0 Specification (<https://loro-alliance.org/sites/default/files/2018-04/lorawantm-backend-interfaces-v1.0.pdf>), verfügbar am 25.8.2019)
- LORAALLI- ANCE-3 LoRaWAN™ Alliance (2018): LoRaWAN™ 1.0.3 Specification (<https://loro-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf>), verfügbar am 25.8.2019)
- LORAALLI- ANCE-4 LoRaWAN™ Alliance (2017): LoRaWAN™ 1.1 Regional Parameters (https://loro-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf), verfügbar am 8.9.2019)
- LORASERVER- 1 LoRa Server (2019): LoRa Server - Getting started (<https://www.loraserver.io/overview/>), verfügbar am 23.9.2019)
- LORASERVER- 2 LoRa Server (2019): Downloads (<https://artifacts.loraserver.io/downloads/loro-gateway-os/raspberrypi/raspberrypi3/3.0.0test1/loro-gateway-os-full-raspberrypi3--20190714105631.sdimg.gz>), verfügbar am 31.7.2019)
- LWGB Bayerische Landesanstalt für Weinbau und Gartbau – Fachzentrum Bienen (Hg.): Standort / Platzwahl für Bienenvölker (https://www.lwg.bayern.de/mam/cms06/bienen/dateien/wo_stelle)

[_ich_meine_bienen_hin.pdf](#), verfügbar am 3.8.2019)

- OMA OpenMobileAlliance (2019): OMA LightweightM2M (LwM2M) Object and Resource Registry (<http://openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html>), verfügbar am 29.9.2019)
- PERDELWITZ Perdelwitz, Robert (2018): Wärmebildkamera in der Imkerei, Wärmebildkamera am Bienenstand. In: *bienen & natur*. Jg. 2 (Heft 12/2018), S. 14 ff.
- RAK-1 RAK wireless TechnologyCo., Ltd. (2019): Pilot Gateway Pro RAK7243 Version 1.1 (https://downloads.rakwireless.com/en/LoRa/Pilot-Gateway-Pro-RAK7243/Hardware-Specification/Pilot_Gateway_Pro_RAK7243_User_Manual_V1.1.pdf), verfügbar am 23.9.2019)
- RAK-2 RAK wireless TechnologyCo., Ltd. (2019): Get Start with RAK2245 & RAK831 RPi LoRa Gateway Revision 1.0 (https://downloads.rakwireless.com/en/LoRa/RAK2245-Pi-HAT/Application-Notes/Get_Start_with_RAK2245%26RAK831_RPi_LoRa_Gateway.pdf), verfügbar am 23.9.2019)
- SACHS Sachs, Roland; Tautz, Jürgen Prof. Dr. (2016): Reduzierung der Feuchtigkeit im Bienenstock durch aktives Belüften bei der Biene (*Apis mellifera*) (https://www.hobos.de/media/user_upload/Images_pdfs_etc/publikationen_ebooks/sachstautzdt.pdf), verfügbar am 13.9.2019)
- SCHMIDT Schmidt, Stefan (keine Angabe): Wie funktioniert eigentlich eine Wägezelle? (<https://www.hbm.com/de/6768/wie-funktioniert-eigentlich-eine-waegezelle/>), verfügbar am 22.9.2019)

- SEMTECH-1 Semtech Corporation (2015), AN1200.22 LoRa Modulation Basics (<https://www.semtech.com/uploads/documents/an1200.22.pdf>, verfügbar am 10.08.2019)
- SEMTECH -2 Semtech Corporation (2019), SX1272/3/6/7/8: LoRa Modem Designer's Guide, AN1200.13 (<https://www.semtech.com/uploads/documents/LoraDesignGuideSTD.pdf>, verfügbar 24.8.2019)
- SEMTECH -3 Semtech Corporation (2019), Datasheet SX1276/77/78/79 (https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, verfügbar am 24.8.2019)
- SEMTECH-4 Semtech Corporation (2019), Frequently Asked Questions, What are typical ranges of good and poor values for RSSI and SNR one should expect for an uplink? (<https://lora-developers.semtech.com/knowledge-base/faq-single/189>, verfügbar 11.10.2019)
- SENEVIRATNE Seneviratne, Pradeeka (2019): Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks, 1. Auflage, Springer Science+Business Media, New York
- THINGS-BOARD-1 ThingsBoard (2019): ThingsBoard Documentation (<https://thingsboard.io/docs/>, verfügbar am 23.9.2019)
- THINGS-BOARD-2 ThingsBoard (2019): What is ThingsBoard? (<https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>, verfügbar am 28.9.2019)
- THINGS-BOARD-3 ThingsBoard (2019): Platform Integration (<https://thingsboard.io/docs/user-guide/integrations/>, verfügbar am 28.9.2019)

- THINGS-BOARD-4 ThingsBoard (2019): ThingsBoard architecture overview (<https://thingsboard.io/docs/reference/>, verfügbar am 28.9.2019)
- THINGS-BOARD-5 ThingsBoard (2019): ThingsBoard installation (<https://thingsboard.io/docs/installation/>, verfügbar am 28.9.2019)
- WAAGENET GRAM (2019): GRAM K2-F (<https://waagenet.de/gram-k2-f>, verfügbar am 23.9.2019)
- WIKIPEDIA-1 Wikipedia.org (2019): Schutzklasse (Elektrotechnik) – Schutzklasse III / Schutz durch Kleinspannung ([https://de.wikipedia.org/wiki/Schutzklasse_\(Elektrotechnik\)#Schutzklasse_III/_Schutz_durch_Kleinspannung](https://de.wikipedia.org/wiki/Schutzklasse_(Elektrotechnik)#Schutzklasse_III/_Schutz_durch_Kleinspannung), verfügbar am 13.9.2019)
- WIKIPEDIA-2 Wikipedia.org (2019): Schutzart (<https://de.wikipedia.org/wiki/Schutzart>, verfügbar am 13.9.2019)
- XLOG Micro EL, SMS/GPRS Bienenstockwaage XLOG bee (https://www.bienen-ruck.de/media/pdf/beschreibung_bienenstockwaage_sms_gps.pdf, verfügbar am 23.9.2019)

Anlagen

Teil 1 A-I

Anlagen, Teil 1

Arduino Quellcode für das LoRaWan Node.

```
//
*****
*****
// Bees Monitoring System - Node
// This file do following:
// - read temperature, pressure, humidity (and altitude) of one or
two connected BME280 sensors
// - read weight of connected Hx711 ADC
// - transmit data via LoRaWan connectivity as Cayenne LPP format-
ted data
// - all periphery and MKR1300 board will periodically sleep and
wake (to safe battery power)
//
// Thanks for libraries which were used in the project.
// This project is a part of my thesis for technical computer
science at the university of
// applicane sciences in Mittweida (Germany). The project is a
show case for LoRaWan technology.
//
// Thanks for your interest.
// Ronald Creutz, 06.10.2019
//
*****
*****

//
*****
*****

// insert needed libraries
//
*****
*****

// ... powersafer for akr boards
#include "ArduinoLowPower.h" // note: the library RTCZero is also
needed

// ... Hx711
#include "HX711.h"

// ... Bosch Bme280 sensor
#include <Wire.h>
#include <SparkFunBME280.h>
```

```
// ... LoRaWan
#include <MKRWAN.h>

// ... Cayenne LPP
#include <CayenneLPP.h>

//
*****
*****
// settings and datastructure
//
*****
*****
// ... global stuff
const int SLEEP_TIME = 600000; // ms 10min.
const int MAX_TRY_AFTER_ERROR = 3;
const int WAITING_TIME_AFTER_ERROR = 5000; // ms
int _cntError = 0;

// ... for Battery Power measurement
const int BATTERY_PIN = A3;
float _batteryVoltage = 0.0;

// ... for HX711
const int LOADCELL_DOUT_PIN = A1;
const int LOADCELL_SCK_PIN = A2;
float _calibration_factor = 0.0;
HX711 _scale;
float _weight = 0.0;

// ... for BME280 sensor(s) - I2C connection
const int _indoorSensorAddress = 0x76;
const int _outdoorSensorAddress = 0x77;
bool _outdoorSensorConnected = false;
BME280 _indoorSensor;
BME280 _outdoorSensor;
float _indoorTemperature = 0.0;
float _indoorPressure = 0.0;
float _indoorAltitude = 0.0;
float _indoorHumidity = 0.0;

float _outdoorTemperature = 0.0;
float _outdoorPressure = 0.0;
float _outdoorAltitude = 0.0;
float _outdoorHumidity = 0.0;

// ... for LoRaWan connectivity
const String APPEUI = "0000000000000000"; // always this value
```

```
const String APPKEY = "TODO: value have to match to you settings";
// use MSB and copy it of gateway website
LoRaModem _modem;

// ... for CayenneLPP
CayenneLPP _lpp(51);

//
*****
*****
// methods
//
*****
*****
// The setup which executes when board starts after power on.
void setup() {
  bool status;
  Serial.begin(9600);
  // todo: only debugging issues!
  // while (!Serial);

  // config for battery voltage measurement
  pinMode(BATTERY_PIN, INPUT);

  // LoRaWan configuration
  Serial.println("Configure LoRaWan.");

  _cntError = 0;
  do {
    if (!_modem.begin(EU868)) {
      _cntError = _cntError + 1;
      Serial.println("Failed to start LoRa modem.");
      delay(WAITING_TIME_AFTER_ERROR);
    };
  } while ((_cntError != 0) & (_cntError < MAX_TRY_AFTER_ERROR));

  if (_cntError == MAX_TRY_AFTER_ERROR) {
    Serial.print("Error ocured during LoRa modem configuration");
    return;
  }

  Serial.print("Your module version is: ");
  Serial.println(_modem.version()); // ARD-078 1.1.5
  Serial.print("Your device EUI is: ");
  Serial.println(_modem.deviceEUI()); // needed for gateway-device
  configuration

  // based on read deviceEUI -> configure HW
```

```
if (_modem.deviceEUI() == "Todo: this have to be create in Lo-
RaServer")
{
    configureHardwareForScale1();
}

//if (_modem.deviceEUI() == " Todo: this have to be create in
LoRaServer ")
//{
//configureHardwareForScale2();
//}

ShowConfiguration();

// Hx711 configuration
Serial.println("Hx711 configuration.");
_scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
_scale.set_scale(_calibration_factor);

// indoor - bosch bme280 sensor configuration
Wire.begin();

Serial.println("Indoor bosch bme280 configuration.");
_indoorSensor.setI2CAddress(_indoorSensorAddress);

_cntError = 0;
do {
    if (!_indoorSensor.beginI2C()) {
        _cntError = _cntError + 1;
        Serial.println("Could not find a valid indoor sensor, check
wiring!");
        delay(WAITING_TIME_AFTER_ERROR);
    }
} while ((_cntError != 0) & (_cntError < MAX_TRY_AFTER_ERROR));

if (_cntError == MAX_TRY_AFTER_ERROR) {
    Serial.print("Error occured during indoor sensor configura-
tion. Cancel process.");
    return;
}

if (_outdoorSensorConnected) {
    Serial.println("Outdoor bosch bme280 configuration.");
    _outdoorSensor.setI2CAddress(_outdoorSensorAddress);

    _cntError = 0;
    do {
        if (!_outdoorSensor.beginI2C()) {
```



```
        _cntError = _cntError + 1;
        Serial.println("Could not find a valid outdoor sensor,
check wiring!");
        delay(WAITING_TIME_AFTER_ERROR);
    }
    } while ((_cntError != 0) & (_cntError <
MAX_TRY_AFTER_ERROR));

    if (_cntError == MAX_TRY_AFTER_ERROR) {
        Serial.print("Error occurred during outdoor sensor configura-
tion. Cancel process.");
        return;
    }
}

// Main method which runs in a loop.
void loop() {
    Serial.println("loop - entry");

    WakeUpPeriphery();

    GetBatteryVoltage();

    GetWeight();

    GetValuesBoschBmeSensors();

    SendLoraWanData();

    SleepPeriphery();

    Serial.println("Enter sleep mode for complete arduino board");
    LowPower.sleep(SLEEP_TIME);
    Serial.println("Arduino board waked up (time period).");

    Serial.println("loop - exit");
}

// Configure some values based on hardware environment.
void configureHardwareForScale1()
{
    Serial.println("configureHardwareForScale1 - entry");

    _calibration_factor = 19980;
    _outdoorSensorConnected = true;

    Serial.println("configureHardwareForScale1 - exit");
}
```

```
}

// Configure some values based on hardware environment.
void configureHardwareForScale2()
{
  Serial.println("configureHardwareForScale2 - entry");

  _calibration_factor = 20190;
  _outdoorSensorConnected = false;

  Serial.println("configureHardwareForScale2 - exit");
}

// Show configuration for debugging.
void ShowConfiguration()
{
  Serial.println("ShowConfiguration - entry");

  Serial.print("Configured calibration factor (HX711):");
  Serial.println(_calibration_factor);
  Serial.println("Outddor sensor is connected:");
  Serial.println(_outdoorSensorConnected);
  Serial.println("I2C address for indoor sensor:");
  Serial.println(_indoorSensorAddress);
  Serial.println("I2C address for outdoor sensor (if connect-
ed):");
  Serial.println(_outdoorSensorAddress);

  Serial.println("ShowConfiguration - exit");
}

// Wake up all connected periphery (exclusive arduino mkr board).
void WakeUpPeriphery()
{
  Serial.println("WakeUpPeriphery - entry");

  _indoorSensor.setMode(MODE_FORCED);

  Serial.println("WakeUp - indoor sensor.");
  while
  (
    //Wait for sensor to start measurment
    _indoorSensor.isMeasuring() == false
  );

  while
  (
    //Hang out while sensor completes the reading
```

```
    _indoorSensor.isMeasuring() == true
  );

  Serial.println("WakeUp - outdoor sensor.");
  if (_outdoorSensorConnected) {
    _outdoorSensor.setMode(MODE_FORCED);

    while
    (
      //Wait for sensor to start measurement
      _outdoorSensor.isMeasuring() == false
    );

    while
    (
      //Hang out while sensor completes the reading
      _outdoorSensor.isMeasuring() == true
    );
  }

  Serial.println("WakeUp - HX711 adc.");
  _scale.power_up();

  Serial.println("WakeUpPeriphery - exit");
}

// Send all connected periphery (exclusive arduino mkr board) in
sleep mode.
void SleepPeriphery()
{
  Serial.println("SleepPeriphery - entry");

  Serial.println("Sleep - indoor sensor.");
  _indoorSensor.setMode(MODE_SLEEP);

  if (_outdoorSensorConnected) {
    Serial.println("Sleep - outdoor sensor.");
    _outdoorSensor.setMode(MODE_SLEEP);
  }

  Serial.println("Sleep - HX711 adc.");
  _scale.power_down();

  Serial.println("SleepPeriphery - exit");
}

// Get voltage (analogue input) of connected battery
void GetBatteryVoltage()
```

```
{
  Serial.println("GetBatteryVoltage - entry");

  // ADC value = 1001 value with full charged battery
  // 3.15 full charged battery (measured with dmm)
  // 1001 / 3.15 = 317
  int adcValue = analogRead(BATTERY_PIN);
  float analogueValue = (adcValue / 317);

  Serial.println("Battery voltage: ");
  Serial.println(adcValue);
  Serial.println(analogueValue);

  _batteryVoltage = analogueValue;

  Serial.println("GetBatteryVoltage - exit");
}

// get weight of HX711
void GetWeight(){
  Serial.println("GetWeight - entry");

  if (_scale.wait_ready_retry(MAX_TRY_AFTER_ERROR, WAIT-
    ING_TIME_AFTER_ERROR)) {
    _weight = _scale.get_units();
    Serial.println("measured weight:");
    Serial.println(_weight);
  }
  else {
    _weight = 0.0;
    Serial.println("Scale (HX711) not found. Cancel process.");
    return;
  }

  Serial.println("GetWeight - exit");
}

// Get all values of connected indoorsensor and outdoorsensor if
it is connected.
void GetValuesBoschBmeSensors()
{
  Serial.println("GetValuesBoschBmeSensors - entry");
  // no error handling possible

  //Start with temperature, as that data is needed for accurate
  compensation.
  //Reading the temperature updates the compensators of the other
  functions.
```

```
//
https://github.com/sparkfun/SparkFun_BME280_Arduino_Library/blob/m
as-
ter/examples/More_Advanced/I2C_and_SPI_Multisensor/I2C_and_SPI_Mul
tisensor.ino
  _indoorTemperature = _indoorSensor.readTempC(); // °C
  _indoorPressure = _indoorSensor.readFloatPressure(); // return
value in unit Pa -> mbar factor 100
  _indoorAltitude = _indoorSensor.readFloatAltitudeMeters(); // m
  _indoorHumidity = _indoorSensor.readFloatHumidity(); // %

  Serial.println("measured values of indoor sensor:");
  Serial.println(_indoorTemperature);
  Serial.println(_indoorPressure);
  Serial.println(_indoorAltitude);
  Serial.println(_indoorHumidity);

  if (_outdoorSensorConnected) {
    _outdoorTemperature = _outdoorSensor.readTempC(); // °C
    _outdoorPressure = _outdoorSensor.readFloatPressure(); // //
return value in unit Pa -> mbar factor 100
    _outdoorAltitude = _outdoorSensor.readFloatAltitudeMeters();
// m
    _outdoorHumidity = _outdoorSensor.readFloatHumidity(); // %

    Serial.println("measured values of outdoor sensor:");
    Serial.println(_outdoorTemperature);
    Serial.println(_outdoorPressure);
    Serial.println(_outdoorAltitude);
    Serial.println(_outdoorHumidity);
  }

  Serial.println("GetValuesBoschBmeSensors - exit");
}

// Sends data formatted as CayenneLpp with LoRaWan modem.
void SendLoraWanData()
{
  Serial.println("SendLoraWanData - entry");

  Serial.println("Starting modem.");
  _cntError = 0;
  do {
    if (!_modem.begin(EU868)) {
      _cntError = _cntError + 1;
      Serial.println("Error occured during LoRa modem configura-
tion (loop).");
      delay(WAITING_TIME_AFTER_ERROR);
    }
  }
```

```
} while ((_cntError != 0) & (_cntError < MAX_TRY_AFTER_ERROR));

if (_cntError == MAX_TRY_AFTER_ERROR) {
    Serial.print("Cancel process.");
    return;
}

delay(500); // important to wait a while?!

Serial.println("Connect to gateway (join request).");
_cntError = 0;
do {
    if (!_modem.joinOTAA(APPEUI, APPKEY)) {
        _cntError = _cntError + 1;
        Serial.println("Error occurred during join to LoRaWan network.");
        delay(WAITING_TIME_AFTER_ERROR);
    }
} while ((_cntError != 0) & (_cntError < MAX_TRY_AFTER_ERROR));

if (_cntError == MAX_TRY_AFTER_ERROR) {
    Serial.print("Cancel process.");
    return;
}

delay(500); // important to wait a while?!

_modem.setPort(3);
_modem.beginPacket();

Serial.println("Build cayenne lpp message.");
BuildLppMessage();

Serial.println("Send message.");
_modem.write(_lpp.getBuffer(), _lpp.getSize());

_cntError = 0;
do {
    if (_modem.endPacket(true) < 0) {
        _cntError = _cntError + 1;
        Serial.println("Error occurred during send LoRa message.");
        delay(WAITING_TIME_AFTER_ERROR);
    }
} while ((_cntError != 0) & (_cntError < MAX_TRY_AFTER_ERROR));

if (_cntError == MAX_TRY_AFTER_ERROR) {
    Serial.print("Cancel process.");
    return;
}
```

```
}

Serial.println("SendLoraWanData - exit");
}

// Build the message for cayenne lpp format.
// a workaround with one gps sensor is used to spend bytes.
// otherwise an overflow occurred with two connected sensor.
void BuildLppMessage()
{
    Serial.println("BuildLppMessage - entry");

    // build message
    _lpp.reset();

    _lpp.addTemperature(1, _indoorTemperature);
    _lpp.addBarometricPressure(1, _indoorPressure / 100); // Pa ->
hPa (expected unit!)
    _lpp.addRelativeHumidity(1, _indoorHumidity);
    // _lpp.addAltitude(1, _indoorAltitude); // not implemented on
LoRaServer decoder
    // _lpp.addGPS(1, 0, 0, _indoorAltitude); // needs 9 bytes -> to
big
    // _lpp.addGenericSensor(5, _weight); // not implemented on Lo-
RaServer decoder
    _lpp.addAnalogInput(1, _weight);
    _lpp.addAnalogInput(2, _batteryVoltage);

    if(_outdoorSensorConnected) {
        _lpp.addTemperature(2, _outdoorTemperature);
        _lpp.addBarometricPressure(2, _outdoorPressure / 100); // Pa -
> hPa (expected unit)
        _lpp.addRelativeHumidity(2, _outdoorHumidity);
        // _lpp.addGPS(2, 0, 0, _outdoorAltitude); // needs 9 bytes ->
to big
        _lpp.addGPS(1, 0, _outdoorAltitude, _indoorAltitude); // ->
to spend some bytes -> altitude: sensor1 alt; longitude: sensor2
alt
    }
    else {
        _lpp.addGPS(1, 0, 0, _indoorAltitude);
    }

    Serial.println("size of LPP message [Bytes]:");
    Serial.println(_lpp.getSize());

    Serial.println("cayenne LPP error:");
    Serial.println(_lpp.getError());
}
```

```
Serial.println("BuildLppMessage - exit");  
}
```


Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 16.10.2019

Ronald Creutz