

Projektdokumentation P7 Data Acquisition for Urban Biodiversity Monitoring



Autor	Timeo Wullschleger timeo.wullschleger@fhnw.ch
Betreuung	Prof. Thomas Amberg
Ort, Datum	Windisch, 29.01.2022

Abstract

Biodiversity is important and must be preserved because, not least, human existence depends on it. If it were to disappear, there would be no food, no clean water and the entire human ecosystem would collapse.

Humans influence biodiversity through their actions, such as the construction of buildings or roads. For this reason, it is important to record the biodiversity in order to detect changes and to take measures.

The starting point of this project is a proof of concept, which is used to collect ecologically relevant data in urban areas. Within the scope of this project, it will be analyzed how a practical, robust system can be developed on the basis of the proof of concept. It should be usable for laymen and easy to rebuild. As a result of the work, an easy-to-use, robust and expandable system was developed and tested.

Inhaltsverzeichnis

1	Einleitung	6
1.1	Ausgangslage	6
1.2	Aufgabenstellung	6
1.3	Strukturierung der Dokumentation	6
2	Theoretische Grundlagen	7
2.1	Biodiversität	7
2.1.1	Genetische Diversität	7
2.1.2	Artendiversität	8
2.1.3	Beschreibung der Artendiversität	8
2.1.4	Diversität der Lebensgemeinschaften und Lebensräume	10
2.1.5	Monitoring der Biodiversität	11
2.1.6	Zukunft der Biodiversitätsmonitoring	16
2.2	IoT Systemarchitektur	19
2.2.1	Referenzmodell	19
2.2.2	Device	19
2.2.3	Gateway	20
2.2.4	Backend	21
2.2.5	Client	24
2.2.6	Transportprotokolle	24
2.2.7	Datenformate	27
2.3	IoT Hardware & Sensorik	29
2.3.1	Mikrocontroller	29
2.3.2	Single Board Computer (SBC)	34
2.3.3	Connectivity	34
2.3.4	Sensorik	42
3	Analyse Ist-Zustand	50
3.1	Kamerasystem	50
3.1.1	PoE Kameras	50
3.1.2	AP Gateway	51
3.1.3	Aufbau und Inbetriebnahme	54
3.2	LoRa Sensoren	54
3.2.1	Umweltsensoren	55
3.2.2	PAX Counter	55
3.2.3	Hardware	56
3.2.4	Aufbau und Inbetriebnahme	57
3.3	Backend	57

3.3.1	TTN Monitoring VM	58
3.3.2	Deployment	58
4	Optimierungsmöglichkeiten	59
4.1	Kamerasystem	59
4.1.1	Hardware PoE Kameras	59
4.1.2	Unterstützung von USB Kameras	59
4.1.3	Robustheit der Capture Applikation	60
4.1.4	Benutzerfreundliche Konfiguration	60
4.1.5	Erweiterte Konfigurationsmöglichkeiten	60
4.1.6	Automatisches Erkennen von angeschlossenen Kameras	60
4.1.7	Deployment	61
4.2	Sensorsysteme	61
4.2.1	Hardware	61
4.2.2	Energiemanagement	61
4.2.3	Sensorik	61
4.2.4	PAX Counter	62
4.2.5	Deployment	62
4.2.6	Connectivity	63
4.3	Backend	63
5	Implementierung	65
5.1	PoE Kamera	65
5.1.1	Hardware	65
5.1.2	Unterstützung von USB Kameras	67
5.1.3	Deployment	67
5.2	AP Gateway	67
5.2.1	Konfigurationsverwaltung	67
5.2.2	Capture	68
5.2.3	Hinzufügen von Kameras	68
5.2.4	Deployment	68
5.2.5	Betrieb	68
5.3	Sensor Nodes	71
5.3.1	Hardware	71
5.3.2	Energiemanagement	76
5.3.3	Unterstützte Sensoren	77
5.3.4	Connectivity	77
5.3.5	Applikation	78
5.3.6	Deployment	79

5.3.7	Konfiguration	80
5.3.8	Payload Format	82
5.4	PAX Counter	83
5.4.1	Hardware	83
5.4.2	Connectivity	84
5.4.3	Messvorgang	84
5.4.4	Applikation	85
5.4.5	Deployment	86
5.4.6	Konfiguration	87
5.4.7	Payload Format	88
5.5	Backend	89
5.5.1	Applikationen	89
5.5.2	Deployment	91
5.5.3	Konfiguration	91
5.5.4	Visualisierung	91
5.5.5	Exportieren von Messwerten	94
6	Fazit	95
7	Literaturverzeichnis	96
8	Abbildungsverzeichnis	101
9	Tabellenverzeichnis	103
10	Formelverzeichnis	103
11	Ehrlichkeitserklärung	104
12	Anhang	104

1 Einleitung

1.1 Ausgangslage

Das interdisziplinäre SNF Forschungsprojekt Mitwelten befasst sich mit dem Monitoring der Biodiversität auf urbanem Terrain. Im Rahmen des Projekts wurde ein Proof of Concept erarbeitet, womit ökologisch relevante Daten erfasst werden.

Konkret wurde ein Kamerasystem zum Fotografieren von Pflanzen entwickelt, womit das Ziel verfolgt wird, Bestäuber auf den Blüten zu erkennen und somit eine Aussage über die Biodiversität zu machen. Die Biodiversität ist stark abhängig von den Umweltfaktoren. Zur Erfassung von Messgrößen der Umwelt wurden Low-Power Sensorsysteme mit LoRaWAN Connectivity entwickelt. Das Kamerasystem und auch das Sensorsystem sind im aktuellen Zustand nicht für einen länger andauernden Einsatz tauglich und haben Verbesserungspotential in der Software und Hardware. Der Aufbau der Systeme kann von Laien nicht ohne fachliche Unterstützung erfolgen.

Im Rahmen der Projektarbeit soll aus dem Proof of Concept ein praxistaugliches System entwickelt werden, das von Laien nachgebaut und bedient werden kann.

1.2 Aufgabenstellung

Im Rahmen der Projektarbeit soll analysiert werden, wie auf Basis des bestehenden Proof of Concept ein einfach bedienbares, robustes und erweiterbares System zur Datenerfassung entwickelt werden kann. Das System soll von Laien bedienbar und einfach nachbaubar sein. Gemäss den Ergebnissen sollen die Änderungen und Optimierungen implementiert werden.

1.3 Strukturierung der Dokumentation

Die Arbeit ist in vier Teile gegliedert. Die Theoretischen Grundlagen dienen als Basis der Arbeit und werden in Form einer Literaturrecherche erarbeitet. Sie umschliesst die Grundlagen der Biodiversität, der IoT Systemarchitektur, Hardware und Sensorik.

Die Analyse des Ist-Zustandes bildet den zweiten Teil. Stärken und Schwächen des Proof of Concept werden analysiert und dokumentiert. Mögliche Optimierungen und Erweiterungen des Systems werden im dritten Teil bearbeitet. Im letzten Teil werden die Implementierungen von ausgewählten Optimierungen behandelt.

2 Theoretische Grundlagen

2.1 Biodiversität

Der Begriff Biodiversität steht kurz für «biologische Diversität» und wurde seit einer Tagung im September 1986, wo er als Thema im Programm stand, immer populärer. Dem Begriff werden zwei Bedeutungen zugeschrieben, zum einen die Umschreibung der Vielfalt des Lebens auf der Erde, zum anderen die Beschreibung von schützenswertem Gut, welches vom Zerfall bedroht ist [1]. Die Convention on Biological Diversity beschreibt den Begriff wie folgt [2]:

"Biological diversity" means the variability among living organisms from all sources including, inter alia, terrestrial, marine and other aquatic ecosystems and the ecological complexes of which they are part; this includes diversity within species, between species and of ecosystems.

Biodiversität umfasst folgende Ebenen:

- Genetische Diversität
- Artendiversität oder interspezifische Diversität
- Diversität der Lebensgemeinschaften und Lebensräumen

Die Bestandteile der Ebenen sind in Abbildung 1 ersichtlich. Die genannten Bestandteile *Arten*, *Unterarten* und *Populationen* sind in allen drei Ebenen enthalten [3].

genetische Ebene	organismische Ebene	ökologische Ebene
		Biome Landschaften Prozesse, <i>Kreisläufe</i> Ökosysteme <i>Interaktionen</i> Biozönosen Habitate Nischen
	Reiche Stämme Klassen <i>Ordnungen</i> Familien Genera	
<i>Arten</i> Unterarten Populationen Individuen	Arten Unterarten Populationen Individuen	<i>Arten</i> <i>Unterarten</i> Populationen
Chromosomen Gene Nucleotide		
* nach Heywood & Baste (1995), ergänzt (<i>kursive Schrift</i> : Ergänzung)		

Abbildung 1 Ebenen und Bestandteile der Biodiversität [3]

2.1.1 Genetische Diversität

Das Aussehen und die biologischen Merkmale von Lebewesen sind durch ihre Gene bestimmt. Die genetische Differenz zwischen zwei Lebewesen nimmt zu, je weiter sie stammesgeschichtlich voneinander entfernt sind (Phylogenie). Neben der genetischen Diversität zwischen verschiedenen Arten (interspezifische Diversität) hat sie auch innerhalb der Arten (intraspezifische Diversität) eine grosse Bedeutung. Wie divers die Gene innerhalb einer Art sind, hängt unter anderem von dem Fortpflanzungsverhalten und der Mobilität der Individuen zusammen. Ermittelt wird die genetische Diversität mittels DNA-Analysen. Die gängigsten vier Verfahren sind [3]:

- restriction fragment length polymorphism (RFLP)
- random amplification of polymorphic DNA (RAPD)
- microsatellite or simple sequence repeat polymorphism (SSR)

- amplified fragment length polymorphism (AFLP) DNA sequencing

In der Naturschutzgenetik wird davon ausgegangen, dass eine hohe genetische Vielfalt für die langfristige Existenz einer Art besser ist als eine kleine. Die Gründe dafür sind [4]:

- Inzucht wird vermieden, womit auch das Auftreten von Erbkrankheiten verringert wird.
- Die Anpassungsfähigkeit einer Population ist nur mit einer Vielzahl verschiedener Genvarianten möglich.
- Populationen mit hoher genetischer Vielfalt sind resistenter gegenüber Krankheitskeimen, da die Wahrscheinlichkeit höher ist, dass resistente Individuen vorhanden sind.
- Kurzfristige Veränderungen der Lebensbedingungen werden von genetisch vielfältigen Populationen leichter abgefangen.

2.1.2 Artendiversität

Die Artendiversität beschreibt die Anzahl verschiedener Arten, welche auf der gesamten Erde oder in einzelnen Gebieten vorkommen. Arten wurden herkömmlich ausschliesslich morphologisch definiert, wobei die Einteilung nach Gestalt, Form und Färbung geschah. Im Laufe der Zeit setzte sich das Konzept der Biospezies durch, worin eine Art eine Gruppe von Individuen umfasst, welche sich miteinander paaren können. Eine Paarung mit anderen Arten ist nicht möglich, womit eine reproduktive Isolation zwischen den Arten besteht [5].

Weltweit gesehen wird das Ziel verfolgt, möglichst alle Arten zu erhalten. Im konkreten Fall spielt nicht nur die Anzahl (Quantität) der Arten, sondern auch die Qualität, also die Verteilung und Eigenschaften von Arten, eine wichtige Rolle [3]. Wenn man die in Abbildung 2 ersichtlichen Ökosysteme vergleicht, ist das System *a* trotz einer kleineren Artenanzahl diverser als das System *b*, da im letzteren zwei der drei vorkommenden Arten nur in einem kleinen Teilbereich des Ökosystems vorhanden sind. Im Gegensatz dazu sind die vorkommenden Arten im System *a* in jedem Bereich vorhanden.

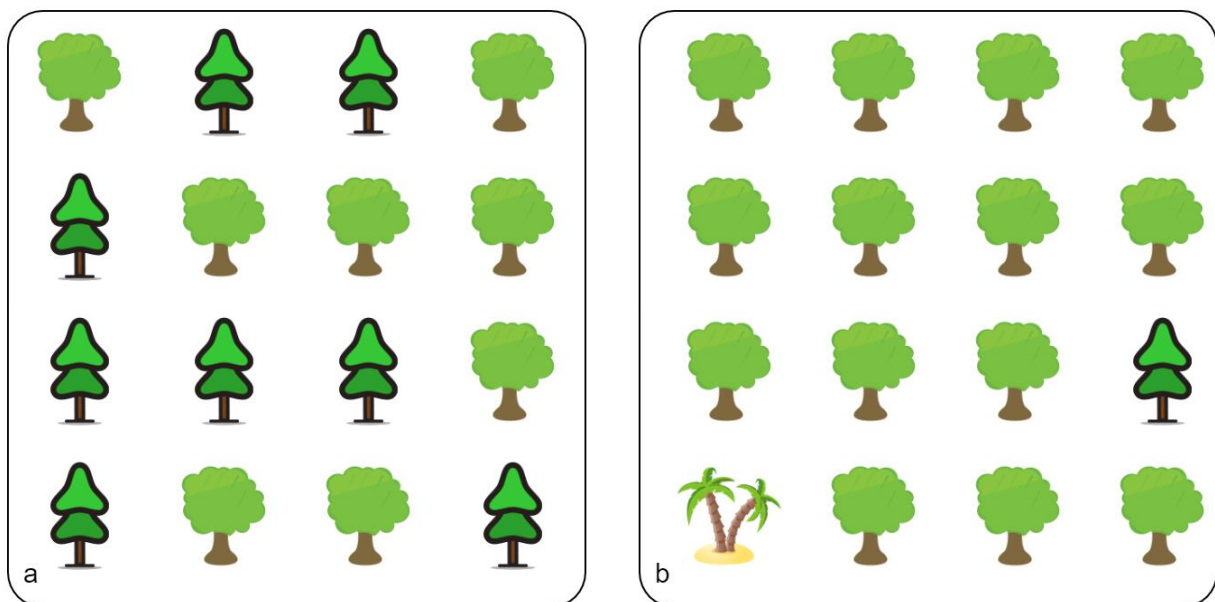


Abbildung 2 Biodiversität bestimmt durch Anzahl Arten, Individuen und Verteilung der Individuen

2.1.3 Beschreibung der Artendiversität

In der Literatur werden verschiedene Typen der Artendiversität, welche jeweils mit griechischen Buchstaben bezeichnet werden, unterschieden.

Die **α -Diversität** bezeichnet das Artenreichtum, also die Artenzahl pro Flächeneinheit in einer Lebensgemeinschaft. Sie wird durch einfaches Zählen oder Schätzen ermittelt.

Mit der **β-Diversität** werden Unterschiede zwischen verschiedenen Biozönosen entlang eines örtlichen Gradienten bezeichnet. Im Gegensatz zu der α-Diversität kann sie nicht durch Zählen ermittelt werden, sondern muss aus vorhandenen Daten berechnet werden.

Um die Artenvielfalt einer grossräumigeren Fläche zu beschreiben, wird die Summe aller Arten, welche in verschiedenen Ökosystemen innerhalb eines Vegetationskomplexes oder einer Landschaft vorkommen, berechnet. Diese Summe wird als **γ-Diversität** bezeichnet. Eine Berechnung der γ-Diversität aus den jeweiligen α-Diversitäten der eingeschlossenen Ökosystemen ist schwierig, wenn die α-Diversitäten geschätzt wurden.

In Tabelle 1 ist ein stark vereinfachtes Beispiel abgebildet, welches die Berechnung der α-, β- und γ-Diversität aufzeigt.

Art	Ökosysteme		
	X	Y	Z
A	✓		
B	✓		
C	✓	✓	
D		✓	✓
E		✓	✓
F		✓	✓
G		✓	✓
H		✓	✓
I			✓
K			✓
α-Div.	3	6	7
β-Div.	X→Y: 7	Y→Z: 3	X→Z: 10
γ-Div.	10		

Tabelle 1 Rechenbeispiel α-, β- und γ-Diversität

Im Ökosystem X kommen drei Arten vor, die α-Diversität ist 3. Im Ökosystem Y kommen insgesamt sechs Arten vor, wobei eine Art davon auch im Ökosystem X vorkommt. Die β-Diversität, bezogen auf die Ökosysteme X und Y beschreibt die Anzahl Arten, welche nur in einem der beiden Systeme vorkommt, also 7. Die Gesamtanzahl der Arten in allen Ökosystemen beträgt 10, woraus die γ-Diversität über alle Ökosysteme 10 beträgt.

2.1.3.1 Metriken zur Beschreibung der Artenvielfalt

Um die Biodiversität auf der Ebene der Arten zu beschreiben, gibt es verschiedene Möglichkeiten. Folgend werden die gebräuchlichsten Formeln beschrieben [3].

Um den Artenreichtum zu berechnen, wird die Artenzahl n durch die Fläche F geteilt.

$$R = \frac{n}{F}$$

Formel 1 Artenreichtum R

Um Ähnlichkeiten oder Differenzen verschiedener Systeme zu beschreiben, werden häufig die Koeffizienten von Jaccard oder Sørensen verwendet. Der Jaccard-Koeffizient S_j wird wie folgt

berechnet, wobei a die Gesamtartenzahl beider Systeme repräsentiert, b und c die Arten beschreibt, die nur in einem der Systeme vorkommen:

$$S_J = \frac{a}{a + b + c}$$

Formel 2 Jaccard-Koeffizient

Sehr ähnlich sieht die Berechnung des Sørensen-Koeffizienten S_S aus. Auch hier beschreibt a die Gesamtartenzahl der beiden Systeme, b und c die Arten, welche nur in einem der Systeme vorkommen:

$$S_S = \frac{2 * a}{2a + b + c}$$

Formel 3 Sørensen-Koeffizient

Weiter können Zusammenhänge und Differenzen mit der Euklidischen Distanz ED berechnet werden. Dabei werden mehrere Bedeutungswerte $x_1 \dots x_n$ definiert, wie zum Beispiel die Individuenzahl oder Deckung. Berechnet werden die Bedeutungswerte in Verbindung mit den Bezugsflächen i und j der beiden Systeme:

$$ED_{ij} = \sqrt{\sum (x_{i,n} - x_{j,n})^2}$$

Formel 4 Euklidische Distanz

Neben dem darstellen von Unterschieden zweier Ökosysteme kann die β -Diversität auch eingesetzt werden, um die zeitliche Entwicklung in einem System aufzuzeigen. Der resultierende Wert repräsentiert den Artenumsatz in der Zeit (species turn-over), wobei a die Anzahl Arten bezeichnet, welche bei beiden Zeitpunkten vorhanden waren, b und c die nur bei einem Zeitpunkt erfassten Arten darstellt:

$$S = \frac{(b + c)}{(a + b + c)}$$

Formel 5 Species turn-over

Für die Diversität ist neben der Artenzahl auch die Individuenzahl massgebend, welche beim Shannon-Weaver-Index H_S berücksichtigt wird. Berechnet wird er aus der Gesamt Individuenzahl N und den Individuenzahlen pro Art n_i :

$$H_S = \sum \left(\frac{n_i}{N} * \ln \left(\frac{N}{n_i} \right) \right)$$

Formel 6 Shannon-Weaver-Index

Um Bestände mit unterschiedlichen Artenzahlen zu vergleichen, wird der Shannon-Weaver-Index H_S durch den Logarithmus Naturalis der Artenzahl S geteilt:

$$E = \frac{H_S}{\ln(S)}$$

Formel 7 Evenness

Neben den beschriebenen Metriken gibt es weitere Formeln, um die Biodiversität zu beschreiben.

2.1.4 Diversität der Lebensgemeinschaften und Lebensräume

Die dritte Ebene der Biodiversität umfasst die Ökosysteme. Ein Ökosystem besteht aus einem charakteristischen Lebensraum (Biotop) und Lebensgemeinschaften (Biozönosen), wobei die Existenz der Biozönosen von ihren Biotopen abhängt. Innerhalb des Ökosystems befindet sich ein bis zu einem gewissen Grad selbstregulierendes Wirkungsgefüge aus biotischen und abiotischen Komponenten, welches auch Stoffkreisläufe, Prozesse und die Zeit einschliesst.

Aufgrund der Abhängigkeit zwischen verschiedenen Ökosystemen werden Lebensräume oftmals auf ganze Landschaften, welche mehrere Biotope umschliessen, erweitert. Im Gegensatz zu den Biodiversitätsstufen der Gene und Arten, wo die Individuen sehr genau eingeteilt und abgegrenzt werden können, ist dies bei Lebensräumen kaum möglich.

Die Lebensbedingungen in einem Biotop werden neben den biotischen Bedingungen durch abiotische Eigenschaften festgelegt. Dazu zählen Umweltbedingungen wie das lokale Klima, die Temperatur und Qualität von Gewässern, Säuregrad und Mineralstoffgehalt des Bodens und Gefälle des Geländes [6].

In jeder Lebensgemeinschaft gibt es Funktionen, für welche die Lebewesen zuständig sind. Die Artenzahl, welche für die Funktion notwendig ist, variiert je nach Art und Funktion. Die Zusammenhänge sind in Abbildung 3 visualisiert. Neben dem linearen Zusammenhang (*species equally important*) kann auch sehr schnell eine Sättigung erreicht werden (*species redundancy*), wobei die Funktion bereits bei einer geringen Artenzahl erfüllt werden kann. Für gewisse Funktionen braucht es eine Schlüsselart (*keystone species*), welche erforderlich ist, um sie zu erfüllen, bei anderen besteht kein genereller Zusammenhang mit der Artenzahl (*idiosyncratic*) [7].

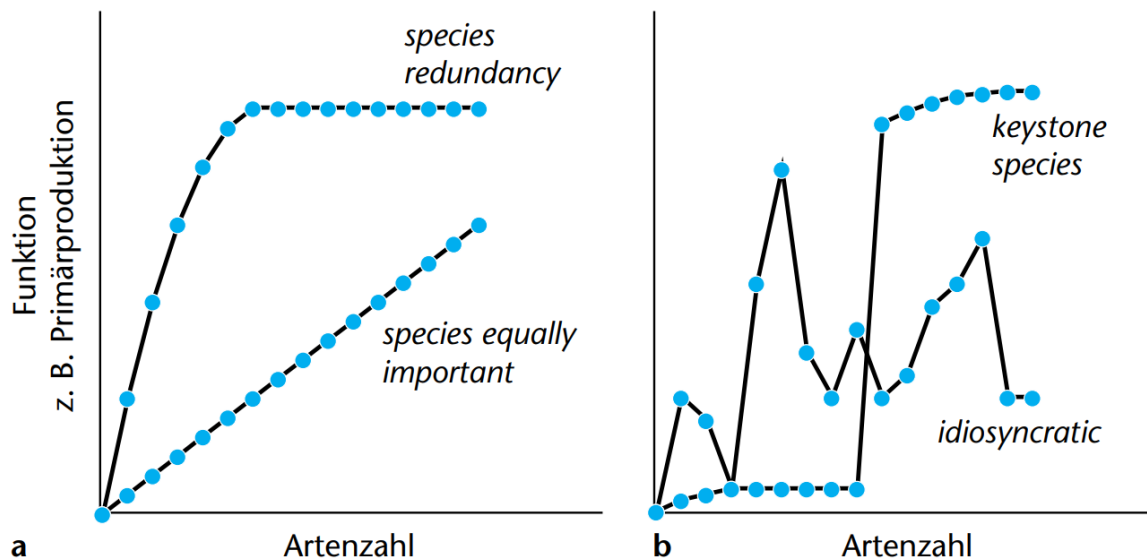


Abbildung 3 Funktion in einer Lebensgemeinschaft in Anhängigkeit der Artenzahl [7]

Nicht zuletzt wird die Vielfalt in einem Ökosystem von der Art und Intensität der menschlichen Eingriffe auf die Landschaft bestimmt.

2.1.5 Monitoring der Biodiversität

Unter Biodiversitätsmonitoring versteht man das Beobachten der Entwicklung der Biodiversität in ihren verschiedenen Stufen. Die Ziele, welche durch das Monitoring verfolgt werden, sind:

- das Erkennen von langfristigen Trends
- das Überwachen der Artenvielfalt in den verschiedenen Ebenen
- repräsentative Aussagen über eine grössere Landschaft oder ein Land zu machen
- Aussagen über Normallandschaften, also nicht primär Naturschutzflächen, zu machen
- Einwirkung von Einflüssen und Massnahmen aufzuzeigen

In der Schweiz wird die Biodiversität regelmässig durch Feldaufnahmen von Biologinnen und Biologen bestimmt. Diese Aufnahmen werden im Rahmen des Projekts Biodiversitätsmonitoring Schweiz (BDM) des Bundesamts für Umwelt BAFU.

2.1.5.1 Indikatoren des BDM

Zur Beschreibung der Biodiversität wurden vom BDM verschiedene Indikatoren eingeführt. Für die Gliederung der Indikatoren wurde das PSR (Pressure, State, Response) Modell verwendet. Die zwölf Indikatoren, welche die Biodiversität beschreiben, sind die Zustände im PSR Modell. Sie sind in Tabelle 2 aufgelistet.

Indikator	Definition
Z1	Anzahl Nutzrassen und -sorten
Z2	Anteil der Nutzrassen und -sorten
Z3	Artenvielfalt der Schweiz und in den Regionen
Z4	Weltweit bedrohte Arten in der Schweiz
Z5	Gefährdungsbilanzen
Z6	Bestand bedrohter Arten
Z7	Artenvielfalt in Landschaften
Z8	Bestand häufiger Arten
Z9	Artenvielfalt in Lebensräumen
Z10	Fläche der wertvollen Biotope
Z11	Qualität der wertvollen Biotope
Z12	Vielfalt von Artengemeinschaften

Tabelle 2 Zustandsindikatoren BDM zur Beschreibung der Biodiversität [8]

Genetische Vielfalt wird aufgrund des hohen Aufwandes der Bestimmung nur von Nutzpflanzen und Nutztieren erhoben. Auf der Ebene der Vielfalt der Lebensräume beschränkt sich das BDM auf einen qualitativen (Z10) und einen quantitativen (Z11) Indikator zu den wertvollen Biotopen.

Die Indikatoren Z3, Z7, Z9 und Z12, welche sich auf die **Diversität der Arten** und Artengemeinschaften beziehen, bilden den Kern der BDM-Erhebungen. Der Indikator Z4 dokumentiert die Wahrnehmung der internationalen Verantwortung der Schweiz. Die Diversität wird in verschiedenen Skalen erfasst, welche in Abbildung 4 ersichtlich sind.

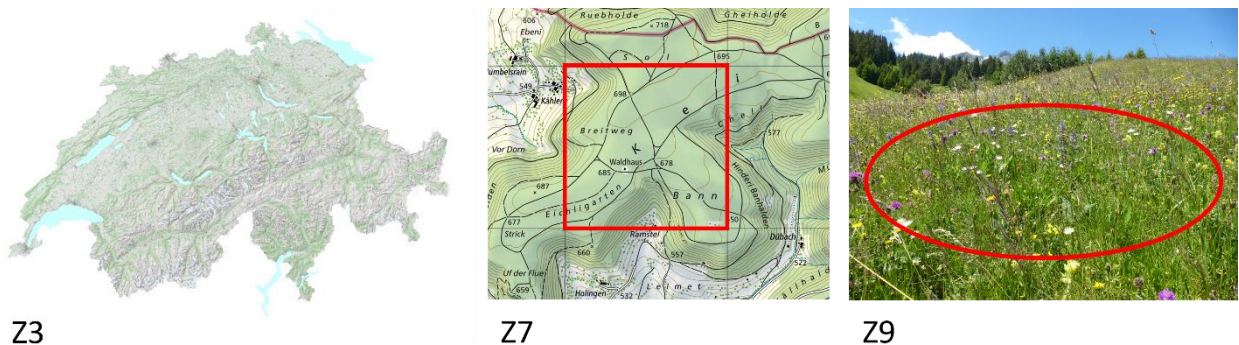


Abbildung 4 Skala der Indikatoren Z3, Z7 und Z9 des BDM [9] [10]

Für die **Einflüsse** auf die Biodiversität wurden 14 Indikatoren festgelegt. Ein ausgewählter Ausschnitt ist in Tabelle 3 ersichtlich. Der Zustandsindikator Z10, welcher die Fläche wertvoller Biotope beschreibt, hat auch einen Einfluss auf die Biodiversität und wird auch bei den Einflüssen als Indikator (E1) verwendet.

Indikator	Definition
E1	Fläche der wertvollen Biotope
E2	Flächennutzung
E3	Fläche der naturüberlassenen Gebiete
E5	Nutzungs- und Bedeckungsvielfalt des Bodens
E6	Nährstoffangebot im Boden
E10	Totholz
E13	Wasserqualität
E15	Landschaftszerschneidung

Tabelle 3 Einflussindikatoren BDM [8]

Die Vielfalt auf verschiedenen räumlichen Ebenen ist von unterschiedlichen Einflüssen gesteuert. In einem Lebensraum wird die Diversität vor allem vom Nährstoffangebot, der Struktur der Fläche und der Nutzungstechnik bestimmt. Neben den vom BDM definierten Indikatoren für Einflüsse gibt es weitere bedeutsame Indikatoren, welche teilweise aufgrund fehlender Datengrundlagen nicht berücksichtigt werden.

Die **Massnahmen** werden in sieben Indikatoren definiert, darunter die Fläche der Schutzgebiete (M1), Ökologische Ausgleichsflächen (M4) und Finanzen für Natur- und Landschaftsschutz (M7). Die Massnahmenindikatoren wurden nach Verfügbarkeit der Daten gewählt. Wie bei den Indikatoren der Artenvielfalt gibt es Massnahmen, welche sich auf die verschiedenen örtlichen Skalen wie Land, Region, Landschaft und Lebensraum auswirken.

2.1.5.2 Artenvielfalt in Landschaften

Da eine flächendeckende Erhebung in einer gesamten Landschaft in der Praxis fast unmöglich ist, wird mit Stichproben gearbeitet. Um die Artenvielfalt in Landschaften aufzunehmen, werden sogenannte Transekte definiert. Eine Transekt-Route ist ein definierter Weg mit vorgeschriebener Länge durch ein Stichprobengebiet. Abbildung 5 zeigt eine Transekt-Route für das Bestimmen der Artenvielfalt von Gefässpflanzen. Das gesamte Gebiet hat eine Fläche von einem Quadratkilometer, die Route ist genau 2.5 Kilometer lang. Bei der Aufnahme werden nur Arten notiert, welche innerhalb eines 2.5 Meter breiten Streifens zu beiden Seiten des Weges wachsen [11].



Abbildung 5 Beispiel Transekt Z7 Gefässpflanzen [8]

2.1.5.3 Artenvielfalt in Lebensräumen

Die Artenvielfalt in Lebensräumen wird auf Stichprobenflächen erfasst. Die Aufnahmeflächen, in welchen die Artenvielfalt Z9 von Gefässpflanzen bestimmt werden, haben eine Fläche von exakt 10 Quadratmetern. In Abbildung 6 ist eine beispielhafte Stichprobenfläche eingezeichnet. Im Zentrum der Flächen sind Magnete vergraben, welche mit einem Magnetsuchgerät lokalisiert werden. Eine 1.78 Meter lange Schnur, welche oberhalb des Magneten befestigt wird, hilft, den Radius einzuhalten.



Abbildung 6 Beispiel Erhebungsfläche Z9 [8]

2.1.5.4 Räumliche Auflösung der Erhebungen

Um Veränderungen der Artenzahl aussagekräftig zu bestimmen, müssen die Standorte der jeweiligen Erhebung immer die selben sein. Alle Standorte zusammen ergeben ein Messnetz. In Abbildung 7 ist das Messnetz für die Erfassung der Artenvielfalt in Landschaften abgebildet. Es umfasst rund 450 Probeflächen von je einem Quadratkilometer Ausdehnung. Die Standorte sind im Jura und auf der Alpensüdseite verdichtet.

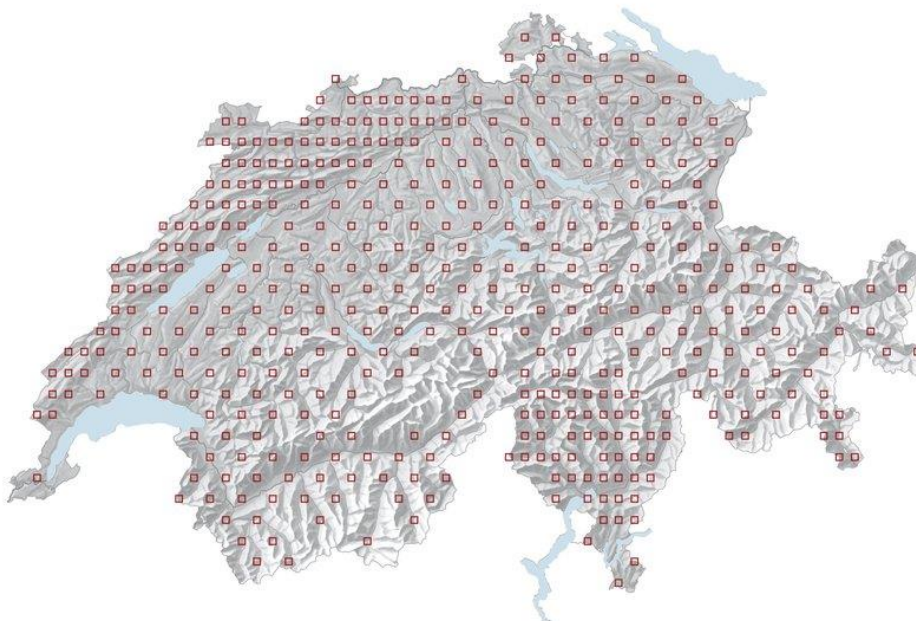


Abbildung 7 Messnetz BDM Artenvielfalt Z7 [12]

Für die Erhebung der Artenvielfalt in Lebensräumen gibt es rund 1450 Messpunkte mit einer Fläche von jeweils zehn Quadratmetern. Es werden Wald, Wiesen und Weiden, Siedlungen, Äcker Alpweiden und Gebirgsflächen unterschieden. Das Messnetz ist in Abbildung 8 abgebildet.

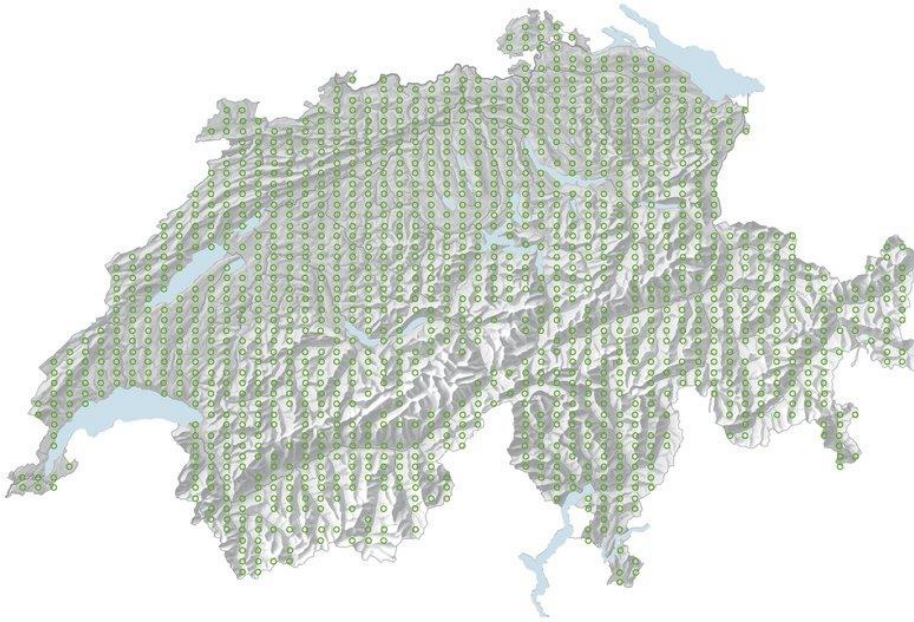


Abbildung 8 Messnetz BDM Artenvielfalt Z9 [12]

Für die Erhebung der Artenvielfalt in Fließgewässern wurde ein Messnetz definiert, welches rund 500 Abschnitte mit einer Länge von 5 bis 100 Meter umfasst. Die Standorte sind in Abbildung 9 ersichtlich.

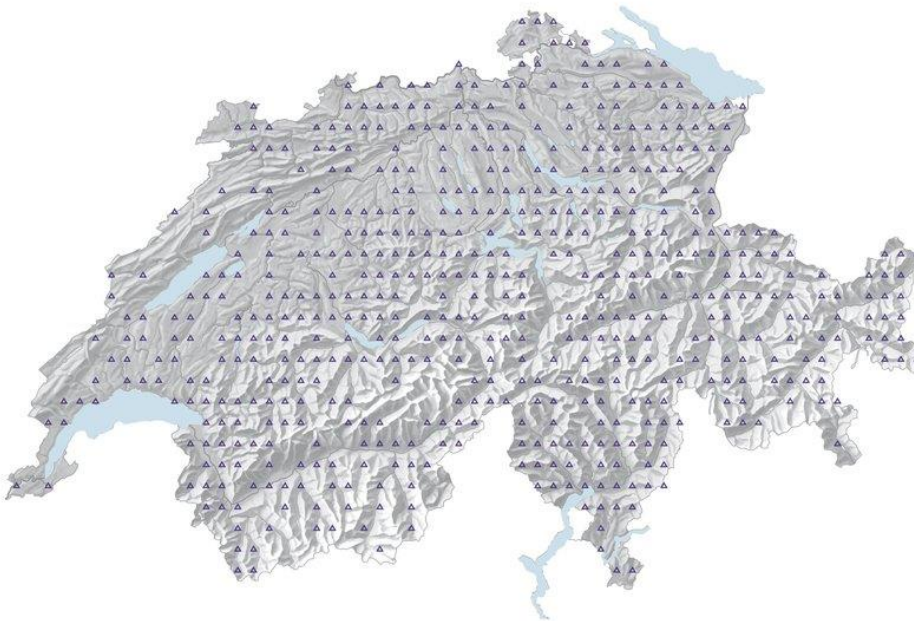


Abbildung 9 Messnetz BDM Artenvielfalt Z9 in Fließgewässern [12]

2.1.5.5 Zeitliche Auflösung der Erhebungen

Die Schweiz hat weltweit das dichteste Netz von Dauerflächen, auf welchen regelmässige Erhebungen durchgeführt werden [13]. Aufgrund der vielen Flächen ist es finanziell nicht möglich, in jedem Aufnahmejahr Erhebungen aller Flächen durchzuführen. Um trotzdem auf allen Flächen eine gleichmässige Datenerhebung durchzuführen, wurde eine zeitliche Staffelung eingeführt,

welche in Abbildung 10 visualisiert ist. Jedes Jahr wird nur ein Fünftel der Gesamtfläche erhoben, was bedeutet, dass man starke Veränderungen einer Teilstichprobe für zurückliegende 5 Jahre nachweisen kann. Schwächere Veränderungen der Gesamtstichprobe sind nach zehn Jahren nachweisbar [8].

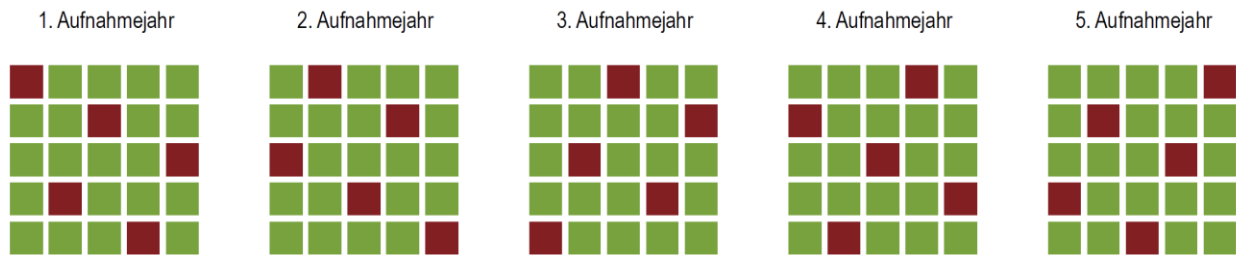


Abbildung 10 Zeitliche Staffelung der Erhebungen des BDM [8]

2.1.6 Zukunft der Biodiversitätsmonitoring

Bei dem herkömmlichen Ansatz des Überwachens der Artenvielfalt werden die Arten durch Menschen identifiziert. Diese Aufgabe erfordert neben den finanziellen Mitteln auch das Fachpersonal in den entsprechenden Bereichen. Um diesen Vorgang zu vereinfachen, wird in verschiedenen Projekten nach einer Automatisierungslösung für die Datenerfassung und das Bestimmen der Spezies geforscht. Die Vision ist es, analog zu einer Wetterstation, eine Station zum Monitoring der Biodiversität zu entwickeln. Ein grosses Forschungsprojekt in Deutschland, AMMOD, verfolgt dieses Ziel [14]. AMMOD steht für *Automated Multisensor Station for Monitoring of Species Diversity* und wird vom Bundesministerium für Bildung und Forschung gefördert. Das Forschungsprojekt beschäftigt sich mit folgenden Technologien:

Mit **Kamerafallen** werden Bilder von Nachtfaltern und Wildtieren aufgenommen. Es werden Algorithmen entwickelt, mit welchen sich die Falter oder Wildtiere lokalisieren und bestimmen lassen. Um Zusatzinformationen für die Erkennung und das Zählen von Tieren zu erlangen, werden Tiefendaten von Stereokameras genutzt. Abbildung 11 zeigt auf der linken Seite ein Graustufenbild, auf der rechten Seite ein Tiefenbild. Die Informationen des Tiefenbilds erleichtern das Erkennen des Wildtieres.

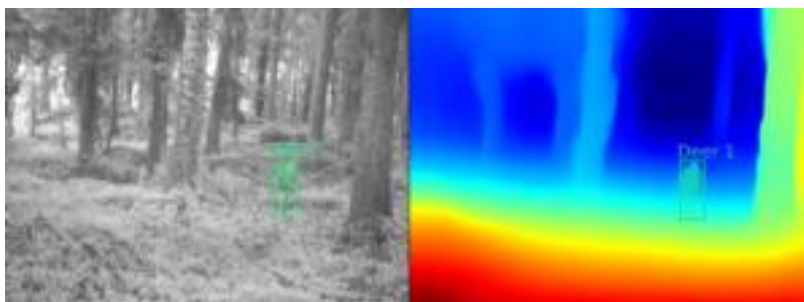


Abbildung 11 AMMOD: Tiefendaten als Zusatzinformation [15]

Mit **Akustiksensoren** werden Tierarten anhand ihrer Rufe oder verursachten Geräuschen erkannt und identifiziert. Erfasst werden die Daten sowohl im hörbaren als auch im Ultraschallbereich. Die Art wird durch Mustererkennung in den Aufnahmen bestimmt. Um die Anzahl der Individuen zu bestimmen, werden jeweils mehrere Mikrofone in verschiedene Richtungen positioniert. Somit können Tiere, deren Geräusche unterschiedlichen Richtungen kommen, voneinander unterschieden werden. Abbildung 12 visualisiert diesen Vorgang mit vier Mikrofonen.

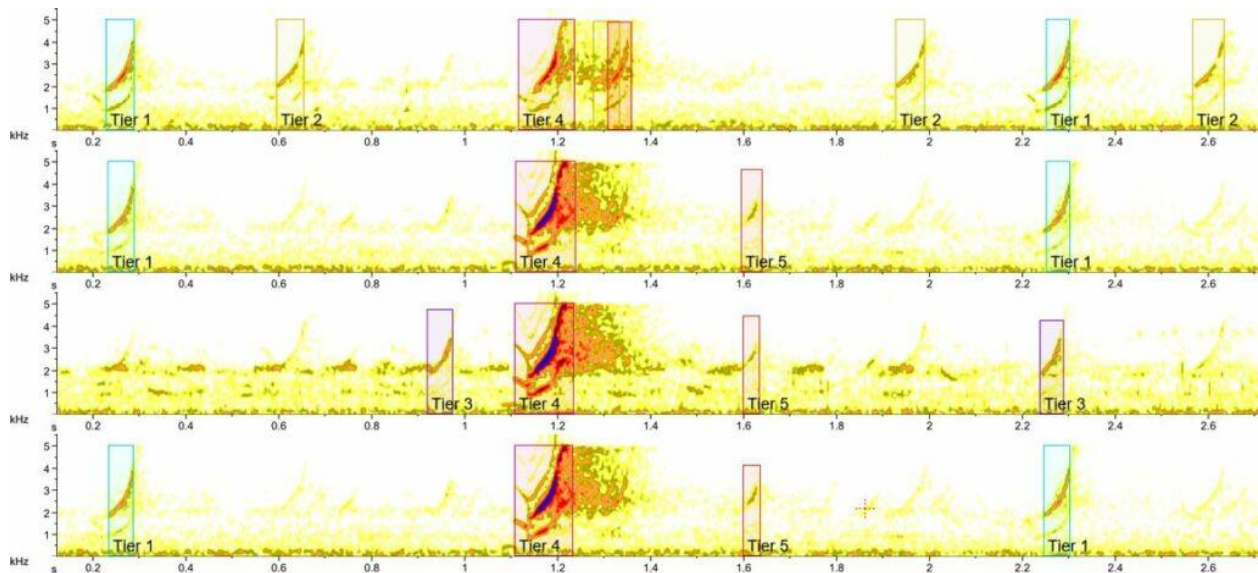


Abbildung 12 AMMOD: Erkennung der Anzahl Individuen mit gerichteten Mikrofonen [16]

Pflanzen sondern **Duftstoffe**, sogenannte pflanzlich volatile organische Verbindungen (pVOCs) ab, aus welchen bedeutsame Informationen über das Wohlergehen der Pflanze und auch der Umwelt entnommen werden können. Die Duftstoffe werden zum einen bei Stress, wie Trockenstress, aber auch um Bestäuber anzulocken, ausgesendet. Auch Tiere, Mikroorganismen oder chemische Prozesse können VOC's aussenden [17]. Mit der Ionenmobilitätsspektrometrie (IMS) können Pflanzen anhand spezifischer Emissionsmuster identifiziert werden. Das Ziel von AMMOD ist, Smellscapes, also verschiedene Duftstoffe an einem bestimmten Bereich, mittels IMS automatisch zu erfassen [18].

Die Daten, welche von den Systemen erfasst wurden, werden anschliessend fusioniert. Bei der **Sensordatenfusion** müssen die Eigenschaften der einzelnen Messungen miteinbezogen werden. Bei der Erfassung könnte eine falsche Art, eine falsche Anzahl oder Tiere, wo gar keine sind, erkannt werden. Diese Unsicherheiten sind auch bei der Fusion der Daten zu beachten. Zusatzinformationen wie Hintergrund- und Kontextwissen oder Daten aus Geoinformationssystemen werden in das Modell miteinbezogen [19].

2.1.6.1 Metabarcoding

Mithilfe der DNA lassen sich Arten eindeutig bestimmen und zuordnen. Das Forschungsprojekt German Barcode of Life (GBOL) hat im Jahr 2009 begonnen, Fauna und Flora in Deutschland genetisch zu charakterisieren und Genabschnitte in einer Referenzdatenbank abzulegen [20]. Die in der Datenbank abgelegten Sequenzen werden auch als DNA Barcodes oder Metabarcodes bezeichnet.

Lebewesen hinterlassen ihre DNA in vielen Formen. Beispiele dafür sind Hautpartikel, Haare, Fischschuppen, Speichel, Urin, abgestorbene Lebewesen, Pollen, Zellen oder Blutspuren. Somit ist es möglich, das Vorhandensein einer Art nachzuweisen, ohne sie jemals zu Gesicht zu bekommen. Mittels Pollenproben werden sehr genaue Aussagen darüber gemacht, wie viele Arten vorhanden sind und wie sich dieser Bestand verändert.

Da jeder Organismus beständig DNA-Haltiges Material an die Umwelt abgibt, sind auch Wasserproben sehr interessant. Um dies zu erforschen, wurden in einem Fluss bei Zürich Proben entnommen und analysiert. In den Proben wurden 255 verschiedene Arten aus bis zu 120 verschiedenen Familien nachgewiesen. Neben Kleinlebewesen aus Gewässern wurden auch Schnecken und Gliedertiere erkannt, deren DNA Bruchstücke durch Regen in den Fluss gelangten [13].

Auch das Forschungsprojekt AMMOD arbeitet am Metabarcoding. Eine automatisierte Windpollenfalle sammelt die Pollen, welche sich in der Luft befinden. Sie ist so konstruiert, dass sie über längere Zeit autonom eingesetzt werden kann und sammelt in bestimmten Zeitintervallen.

Mit einer Malasie-Falle (Abbildung 13) werden fliegende Insekten gesammelt, um deren Arten und die Arten der Pflanzenpartikeln, welche sich an ihnen befinden, zu bestimmen. Die Falle wechselt die Probegefäße zu bestimmten Zeiten automatisch aus, um die Tag-Nacht Rhythmen oder eine kontinuierliche Überwachung von Veränderungen der Insektengemeinschaften zu ermöglichen. Durch das simultane Erfassen der Insekten und Pollen ist es möglich, komplexe Wechselbeziehungen zwischen Fauna und Flora zu erforschen.



Abbildung 13 AMMOD Malaisefalle

Die DNA-Sequenzierung der Analysen geschieht im Labor. Die Entwicklung der Analysegeräte geht rasch voran. Heutzutage erhält man unter Verwendung der Nanopore-Sequenzierungstechnologie die Ergebnisse nahezu in Echtzeit. Die Firma Oxford Nanopore entwickelt portable, leistungsstarke Sequenziergeräte für den Einsatz im Feld. Abbildung 14 zeigt das Modell MinION, welches mit einer Länge von etwas über 10 Zentimetern und einem Gewicht von knapp 90 Gramm für den portablen Einsatz entwickelt wurde [21].



Abbildung 14 Oxford Nanopore MinION [21]

2.2 IoT Systemarchitektur

Unter der Bezeichnung Internet of Things (IoT) versteht man das Vernetzen von Objekten über das Internet, sodass ein globales Netzwerk aus Computern, Sensoren und Aktoren entsteht. Eine einheitliche Definition für den Begriff IoT gibt es nicht. Der Einsatzbereich reicht von industriellen Anwendungen über Medizintechnik bis zu Anwendungen in der Umweltforschung und sind somit sehr divers.

2.2.1 Referenzmodell

Die Grundlegende Basisarchitektur eines IoT Systems ist in Abbildung 15 visualisiert.

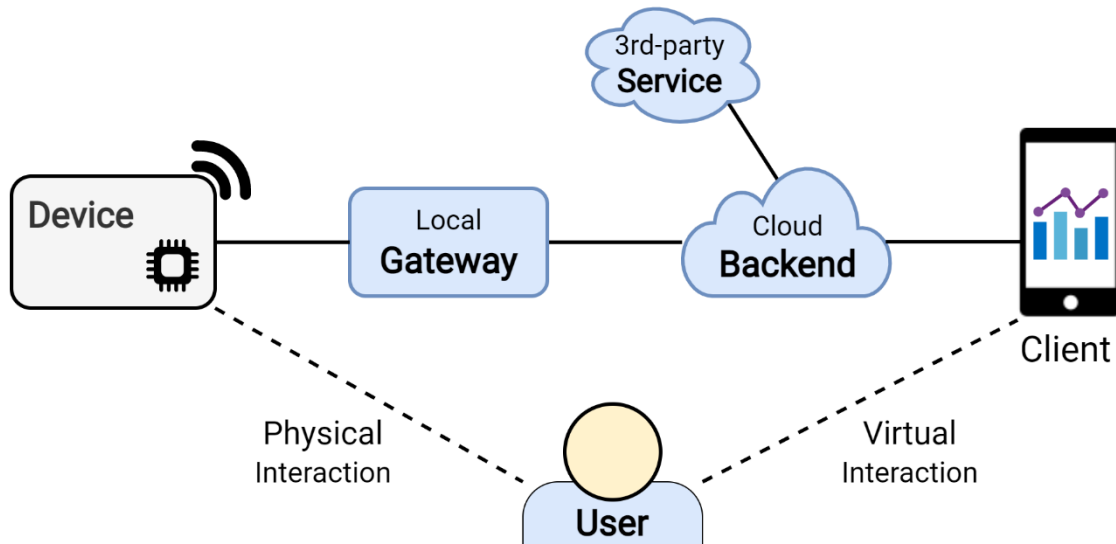


Abbildung 15 IoT Referenzmodell [22]

Das Device kann die Funktion eines Sensors, eines Aktors oder beides zugleich sein. Der Benutzer des Systems kann physikalisch damit interagieren. Es ist mit einem lokalen Gateway verbunden, von welchem aus eine Verbindung zum Backend besteht. Der lokale Gateway kann in gewissen Fällen auch im Device selbst implementiert sein. Im (Cloud) Backend werden Daten gespeichert und es laufen Services, wie zum Beispiel die Auswertung von Sensordaten. Services von Drittanbietern werden vom Backend erreicht. Über den Client kann der Benutzer virtuell mit dem System interagieren. Er kann zum Beispiel ein Dashboard von den erfassten Daten aufrufen. In den folgenden Abschnitten werden die Systemkomponenten genauer definiert.

2.2.2 Device

IoT Devices bestehen aus programmierten Controllern, Kommunikationsverbindungen und je nach Funktion Sensoren, Aktoren oder eine Kombination davon. Sie verbinden physikalische Objekte oder Messgrößen mit der digitalen Welt, um eine grössere Aufgabe zu erfüllen. Oft werden die Geräte nach ihrer Funktion klassifiziert.

Sensing Devices erfassen Messwerte von Sensoren und übermitteln diese. Das können rohe Daten, zum Beispiel die gemessene aktuelle CO2 Belastung, oder aggregierte Daten, wie erkannte Anomalien von Vibrationen in einem Motor, sein.

Acting Devices beinhalten Aktoren und führen gemäss den erhaltenen Kommandos eine Aktion aus. Als Aktoren werden Indikatoren, zum Beispiel Displays, Statusleuchten oder Lautsprecher, oder mechanische Systeme wie Motoren oder Hydraulikventile eingesetzt. Die Kommandos werden in vielen Fällen von Sensing Devices getriggert. Ein Lüftungssystem entscheidet aufgrund der von den Sensing Devices, welche den CO2 Wert messen, ob das Fenster, welches vom Acting Device bedient wird, geöffnet wird.

Es gibt auch Devices, die sowohl Daten erfassen als auch Aktionen ausführen. Die Steuer- oder Regelung kann in dem Device selbst oder auf dem Backend implementiert sein.

Abbildung 16 zeigt ein LHT65 Sensor, klassisches IoT Sensing Device. Er überträgt Messwerte von Temperatur und Luftfeuchtigkeit an ein LoRa Netzwerk.



Abbildung 16 IoT Device Beispiel: LHT65 [23]

Im Kapitel IoT Hardware & Sensorik wird der Aufbau der Hardware genauer beschrieben.

IoT Devices können mit anderen Geräten kommunizieren. Die Kommunikation kann kabelgebunden oder drahtlos, über das Internet oder eine lokale Verbindung implementiert werden. Abhängig von der Übertragungstechnologie wird ein Gateway eingesetzt, um die Daten in ein anderes Protokoll zu konvertieren.

2.2.3 Gateway

Ein Gateway ist ein aktiver Netzwerkknoten und hat die primäre Funktion, eine logische Verbindung zwischen einer Quelle und einem Ziel herzustellen und Daten zu übermitteln bzw. weiterzuleiten. Gateways können alle Schichten des OSI Schichtenmodells berücksichtigen, in der Praxis wird von drei Ansätzen ausgegangen [24]:

- Gateways auf den OSI Schichten 1 und 2 verbinden Netze oder Systeme mit unterschiedlichen Übertragungsmedien und Sicherungsschichten. Sie werden auch als Medienkonverter bezeichnet.
- Protokoll-konvertierende Gateways, welche Netze mit unterschiedlichen Protokollen auf den Schichten 3 und 4 verbinden.
- Gateways zur Anwendungs-konvertierung auf Schicht 7 übersetzen Datenformate, Datenströme und Adressformate.

IoT Gateways haben weitaus mehr Funktionen als nur das weiterleiten von Datenverkehr. Sie umfassen:

- Vorverarbeitung von Daten (Aggregation, Komprimierung, Deduplizierung)
- Speichern von Daten über kurze oder längerfristige Zeitperiode
- Ausführen von Code, beispielsweise für das Beschaffen von Daten durch regelmäßige HTTP Requests an Devices
- Offlinefunktionalität, um bei einem Ausfall der Internetverbindung trotzdem Daten zu erfassen
- Sicherheitsvorkehrungen und Verwaltung für die Devices

Die Verwendung eines IoT Gateways kann dem Gesamtsystem zusätzliche Sicherheit bieten. Im Gegensatz zu IoT Devices, welche direkt mit dem Internet verbunden sind existiert beim Einsatz nur eine Eintrittsstelle, welche über das Internet erreichbar ist. Durch die Rechenleistung, welche in der Regel höher ist als die eines IoT Devices sind bessere Sicherheitsvorkehrungen wie berechnungsintensive Verschlüsselungen möglich.

Ein weiterer Einsatzbereich von IoT Gateways ist Edge Computing, wo die Datenverarbeitung direkt auf dem Gateway durchgeführt wird. Edge Computing reduziert die benötigte Bandbreite und auch das Datenvolumen, da die Rohdaten auf dem Gateway bleiben. Eine Überwachungssystem mit Kameras ist ein Anwendungsbeispiel. Anstatt die Videodaten in Echtzeit hochzuladen analysiert man die Bilder auf Bewegungen und lädt nur die Sequenzen, auf welchen Aktivitäten ersichtlich sind, hoch. Das Gateway wird in diesem Fall auch als Edge Gateway bezeichnet [25].

Wenn ein System wächst und eine hohe Skalierbarkeit gefordert ist, können mehrere IoT Gateways eingesetzt werden, die zusammen kommunizieren. Die verbundenen Gateways befinden sich in einem eigenen Netzwerk, welches von aussen nur über einen Edge Router erreichbar ist. Das Netzwerk wird Fog genannt und die Gateways werden als Fog Server bezeichnet. Die zu erfüllende Funktion ist identisch mit denen eines IoT Edge Gateways. Im Einsatzbereich von Fog Computing liegen Systeme mit hoher Ausfallsicherheit, wie Fabriken oder Gebäudeautomations-systeme.

Abbildung 17 stellt eine Übersicht der drei Topologien dar. Auf der linken Seite ist die Topologie ohne Gateway dargestellt, welche als Cloud Computing bezeichnet wird. In der Mitte ist die Topologie mit einem Edge Gateway dargestellt. Rechts davon ist ein Beispiel einer Fog Computing Topologie. Es besteht aus zwei Servern und einem Edge Router, welcher für die Verbindung mit dem Cloud Backend verantwortlich ist.

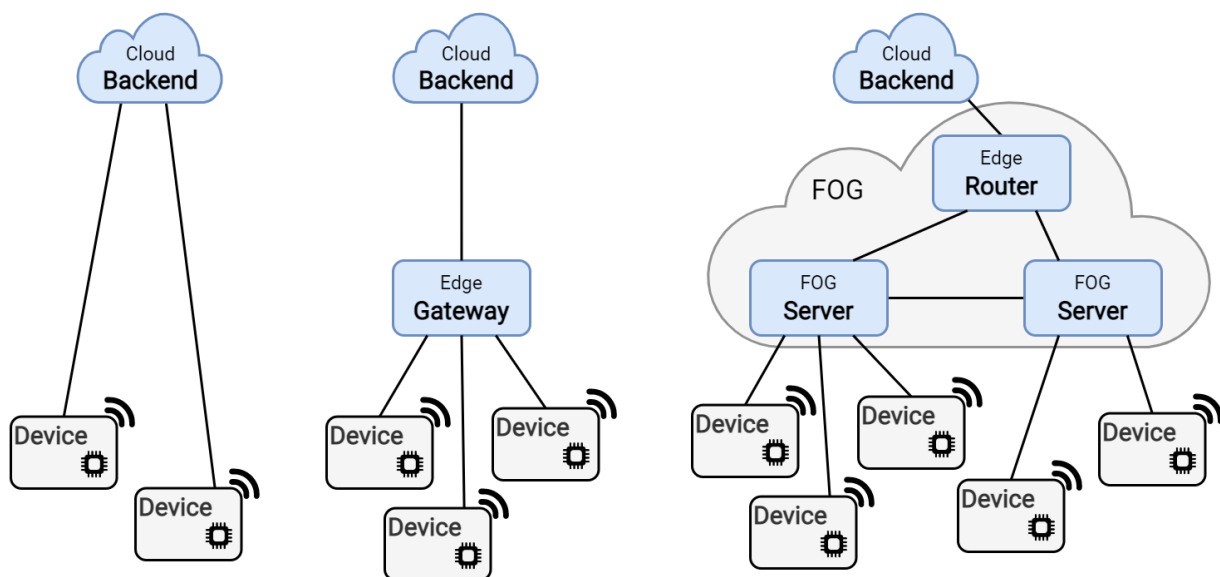


Abbildung 17 IoT Topologien Cloud, Edge und Fog

2.2.4 Backend

IoT Sensing Devices erfassen Daten und übermitteln diese entweder direkt oder über ein Gateway an das Backend. Im Backend läuft ein Stack aus Software, welcher für die Funktionalität des IoT Systems zuständig ist. Es kann auf einem lokalen Server oder auf einer virtuellen Maschine (VM) implementiert sein, wobei man bei letzterem von einem Cloud Backend spricht.

Ein Backend muss Daten empfangen und versenden können. Abbildung 18 zeigt die Bestandteile eines sehr einfachen Backends mit monolithischer Architektur. Der Reverse Proxy leitet die eingehenden Verbindungen an den entsprechenden Endpunkt, in Abbildung 18 an die Application,

weiter. Die Application verarbeitet die Daten, schreibt sie in die Datenbank oder liest Daten aus der Datenbank und antwortet auf Anfragen.

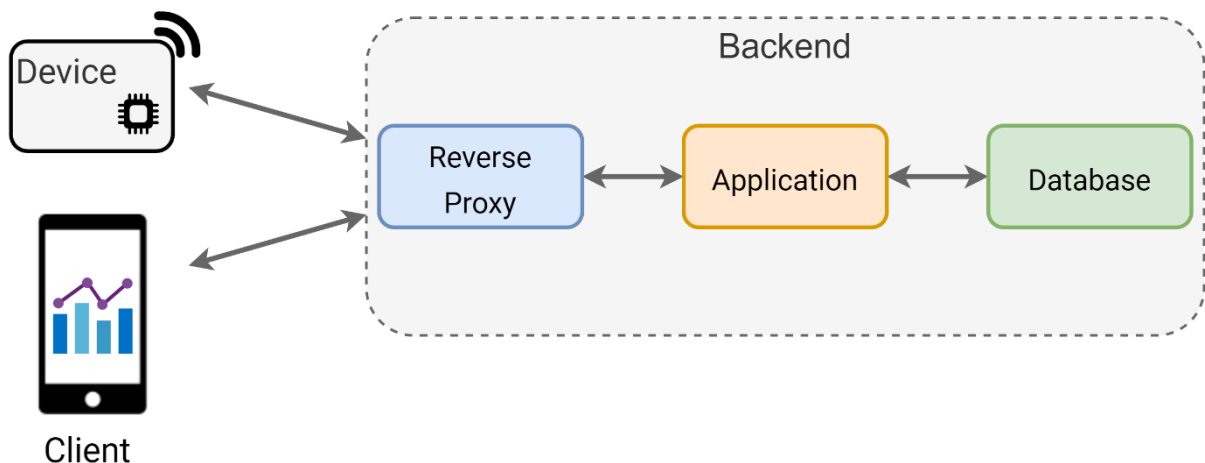


Abbildung 18 Bestandteile einfaches Backend

Folgend werden die Komponenten, welche in IoT Backends eingesetzt werden, beschrieben.

2.2.4.1 Reverse Proxy

Ein Reverse Proxy stellt Ressourcen aus einem oder mehreren internen Servern oder Services für externe Clients zur Verfügung. Er übersetzt die eingehende Anfrage und leitet sie im Backend Netzwerk an den entsprechenden Service weiter. Die internen Adressen der Infrastruktur bleiben dem Client verborgen. Abbildung 19 zeigt das Funktionsprinzip. Die Anfragen für App 1 und App 2 an die selbe Domain werden vom Reverse Proxy an die entsprechenden Services weitergeleitet.

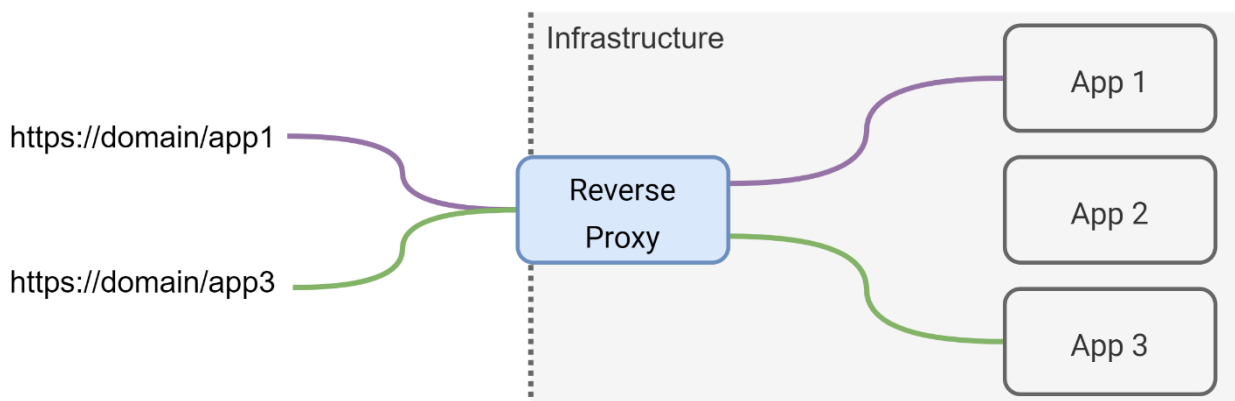


Abbildung 19 Visualisierung Weiterleitung Reverse Proxy

Beim Einsatz von TLS Verschlüsselungen können die Zertifikate direkt auf dem Proxy eingesetzt und verwaltet werden, wobei der Aufwand der Implementierungen von TLS für die einzelnen Applikationen entfällt. Auch die Benutzerauthentifizierung kann auf dem Reverse Proxy für einzelne Server oder für die gesamte Infrastruktur implementiert werden.

Wenn mehrere identische Applikationen auf dem Backend laufen, können einige Reverse Proxys die Aufgabe der Lastverteilung (Loadbalancing) übernehmen. Beim Loadbalancing werden die Anfragen gemäss definierten Regeln an die Server weitergeleitet, damit eine möglichst gleichmässige Auslastung besteht. Beim Ausfall eines Servers werden die Anfragen abgefangen. Damit wiederholte identische Anfragen nicht immer an die Applikationen weitergeleitet werden müssen, unterstützen manche Reverse Proxys eine Caching Funktion von gewissen Inhalten wie zum Beispiel Bildern.

In der Regel werden Reverse Proxys schichtweise konfiguriert. Abbildung 20 zeigt die Module, welche eine Anfrage durchläuft, bis sie zum Zielserver gelangt [26]. Im Modul Entrypoints werden die Adressen und Ports, welche gegen Aussen erreichbar sind, und die erlaubten Protokolle, festgelegt. In den Routern werden Routing Regeln festgelegt, welche entscheiden, wohin die Anfrage weitergeleitet wird. Die Regeln beinhalten Subdomains, Path Präfixe und Headers, für die Grundlage der Entscheidung. Die Anfrage wird dann, wenn vorhanden, an Middlewares weitergeleitet, wo die Anfrage modifiziert oder auf Inhalte geprüft wird. Beispiele einer Middleware sind die Authentifizierung durch Überprüfung des Basic Auth Headers oder das Anfügen von Headers. Die Anfrage wird nach den Middlewares an einen Service weitergeleitet, sofern einer implementiert wurde. Zu den Services gehören Loadbalancing und Health Checks. Der Service entscheidet dann, an welchen Server die Anfrage geleitet wird. Bei einer Implementierung ohne Middlewares und Services wird die Anfrage direkt nach dem Routing an den entsprechenden Server weitergeleitet.

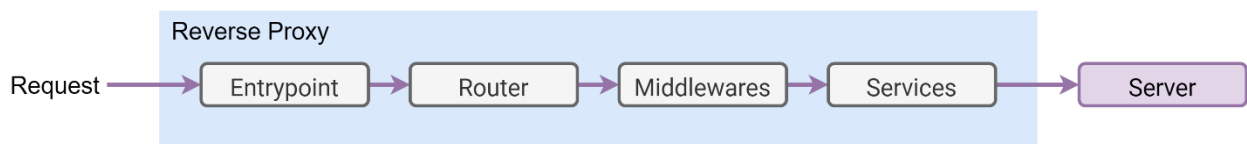


Abbildung 20 Bestandteile Reverse Proxy

2.2.4.2 Datenbanksysteme

Um Messwerte, Metadaten oder Verwaltungsdaten zu speichern, werden Datenbanksysteme eingesetzt. Die Aufgaben eines Datenbanksystems sind [27]:

- Das speichern von Daten
- Die physikalische Verwaltung der gespeicherten Daten
- Das bereitstellen von Daten durch Abfragen
- Gewährleistung Datensicherheit und Datenschutz
- Zugriffsregelung

Grundsätzlich kann man die Datenbanksysteme in SQL und NoSQL (Not only SQL) Datenbanksysteme unterteilen. Zu den SQL Datenbanken gehören die relationalen Datenbanksysteme, worin die Daten in Tabellen gespeichert werden. Die Struktur, in welcher die Daten gespeichert und organisiert sind, muss im voraus festgelegt werden. Auch die Abhängigkeiten unter den Daten sind beim Entwurf festzulegen. Zu den NoSQL Datenbanksystemen gehören Key-Value Store, Dokumentorientierte und Zeitreihenbasierte Systeme. Die Strukturen der Daten für die wichtigsten Systeme sind in Abbildung 21 visualisiert.

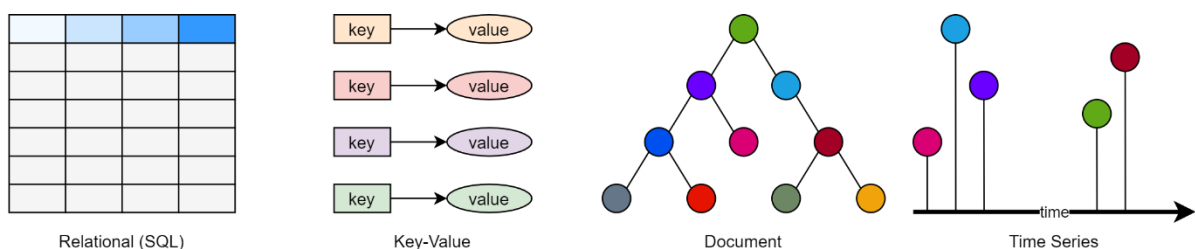


Abbildung 21 Datenstruktur verschiedener Datenbanktypen

Relationale Datenbanksysteme werden eingesetzt, wenn die Struktur und die Abhängigkeiten der Daten bekannt sind. Ein Grund zum Einsatz von SQL Datenbanksystemen liegt in der Möglichkeit, effiziente mengenorientierte Abfragen und Manipulationen zu tätigen. Die Informationen in der Datenbank sind bei einem festgelegten Schema zudem einfach zu verwalten. Die Skalierung geschieht vertikal, was bei hohen Datenmengen zu Effizienzverlusten führt. Ein Einsatzbeispiel ist die Konfiguration eines industriellen IoT Systems mit im Voraus bekannten Geräten und Abhängigkeiten.

NoSQL Datenbanksysteme kommen zum Einsatz, wenn die Daten keine vordefinierte Struktur haben, die Abhängigkeiten nicht vorgegeben sind oder eine hohe Skalierbarkeit gefordert ist. Durch die nicht vorgegebene Struktur erübrigt sich die Entwicklung eines Schemas, erschwert sich aber die Verwaltung der Daten.

Bei vielen Monitoring Systemen haben die Messwerte die Zeitachse als hauptsächliche Abhängigkeit. In diesen Fällen werden Time-Series Datenbanken eingesetzt. Sie sind optimiert für zeitbasierte Werte und unterstützen zeitbasierte Abfragen und Aggregationen. Auf eine Definierung von Abhängigkeiten zwischen Messwerten auf Datenbankebene wird in der Regel verzichtet.

2.2.4.3 Applikationen

Die Applikation ist das Herzstück des Backends, denn sie beinhaltet die Logik des Systems. Im Normalfall beinhaltet ein IoT Backend mehrere Applikationen mit verschiedenen Funktionen. Beispiele für Applikationen sind:

- Webserver mit REST API für die Bereitstellung von Daten
- Kontinuierliche Analyse von Realtime Messwerten
- Analyse von Daten mit Machine Learning Algorithmen
- Management Tool für IoT Devices
- Monitoring Applikation für Systemzustände

Für eine Vielzahl von Funktionen gibt es Open-Source Applikationen, die eingesetzt werden können. Als Beispiel können Messages, welche an ein MQTT Broker gesendet werden von Telegraf [28] in eine Influx Datenbank geschrieben und von Grafana [29] visualisiert werden. Wenn die Payload der Message aber eine spezielle Struktur, welche nicht von Telegraf unterstützt wird, vorweist, ist eine eigene Applikation für das Schreiben der Informationen in die Datenbank erforderlich.

2.2.5 Client

Alles, was auf das Backend zugreift, um Daten abzufragen oder zu ändern, wird als Client bezeichnet. Der Zugriff des Clients geschieht oftmals über eine REST API.

2.2.6 Transportprotokolle

Es existiert eine Vielzahl von Protokollen, welche für die Datenübertragung über eine Netzwerkverbindung eingesetzt werden. Die wichtigsten Protokolle, die in IoT Systemen eingesetzt werden, sind folgend beschrieben.

2.2.6.1 HTTP

HTTP (Hypertext Transfer Protocol) ist ein Client-Server Protokoll, welches seit 1996 eingesetzt wird und in RFC2616 [30] definiert ist. Der Client sendet dem Server einen Request, der Server antwortet mit der Response. Ein Beispiel ist in Abbildung 22 visualisiert.

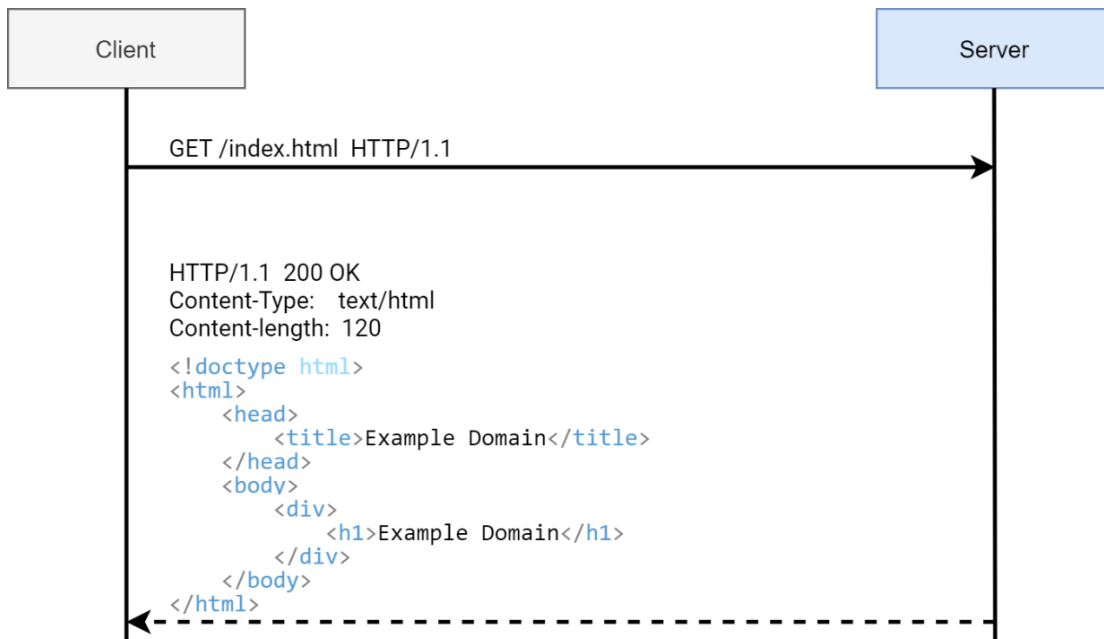


Abbildung 22 HTTP Request Beispiel

Jeder Request hat eine Methode angehängt, welche die mit der Anfrage zu erfüllenden Funktion beschreibt. Im Beispiel in Abbildung 22 wurde eine GET-Anfrage gesendet. Die wichtigsten Methoden sind:

- GET: Anfrage für Daten auf dem Server
- POST: Senden von Daten an den Server
- PUT: Aktualisieren von Daten, welche sich bereits auf dem Server befinden
- DELETE: Löschen von Daten auf dem Server

Jede Antwort, welche vom Server an den Client zurückgeschickt wird, beinhaltet einen Statuscode, welcher Informationen über den Zustand des Requests preisgibt. Die Codes sind gruppiert nach Anfangsziffern. Die wichtigsten Kategorien sind:

- 1xx: Der Request wurde empfangen und wird verarbeitet.
- 2xx: Der Request wurde erfolgreich empfangen, verstanden und akzeptiert.
- 3xx: Der Request wurde weitergeleitet, Umleitung zu neuer URL.
- 4xx: Clientseitiger Error, der Request hat nicht alles, was gebraucht wird oder die Resource existiert nicht.
- 5xx: Serverseitiger Error, der Server konnte den Request nicht beantworten.

Der Statuscode wird im Header einer Antwort mitgeschickt. Headers können sowohl bei Requests oder Responses beinhaltet sein. Headers werden verwendet, um Zusatzinformationen zu einem Request zu übermitteln. Sie dienen zum Beschreiben des Content Types (Content-Type), dessen Länge (Content-Length), der Authorisierung (Basic Auth Header) und weiteren Informationen.

Standardmässig werden die Ports 80 für HTTP und 443 für HTTPS (HTTP mit TLS Verschlüsselung) verwendet.

2.2.6.2 MQTT

MQTT steht für Message Queue Telemetry Transport und ist ein Open Source Broker / Client Protokoll. Clients können Topics abonnieren oder Daten auf ein Topic publizieren. Abbildung 23 zeigt ein Beispiel mit zwei Clients. Der Client «Device» ist ein IoT Device, welches Temperaturwerte misst und diese an den Broker unter dem Topic «measurement/temperature» sendet. Der Client «Backend» befindet sich im Backend und abonniert die Nachrichten aller Topics, welche

mit «measurement/» beginnen. Den Messwert, welcher das Device publiziert hat, wird nach Eintreffen im Broker direkt an den Client im Backend publiziert.

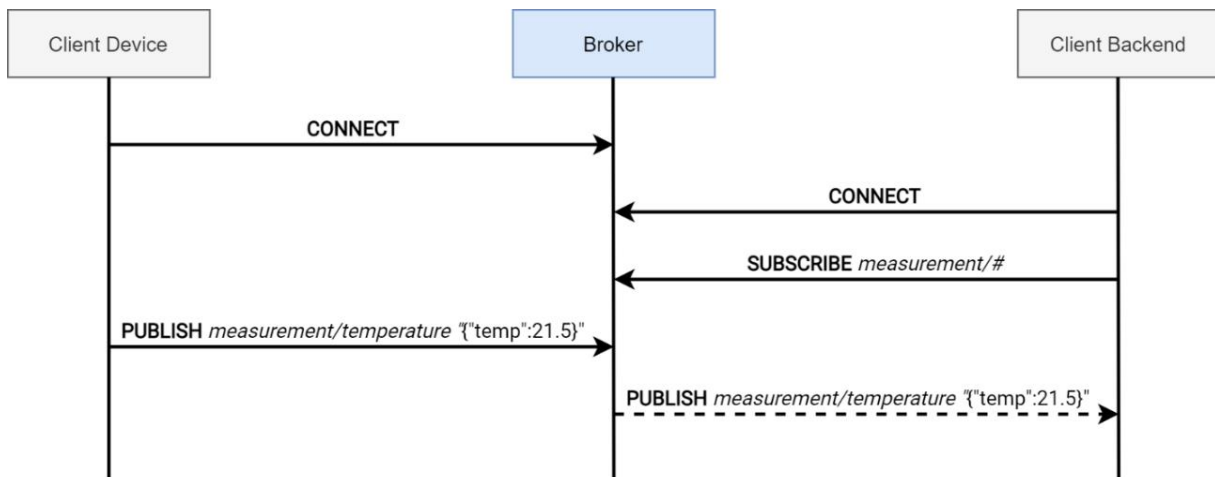


Abbildung 23 MQTT Kommunikation Beispiel

Die Topics sind hierarchisch strukturiert. Um nur ein Subtopic zu abonnieren können Wildcards eingesetzt werden. Beim Beispiel in Abbildung 23 abonniert der Client im Backend mit dem Subscribe Topic «measurement/#» alle Subtopics, welche unter dem Haupttopic «measurement» publiziert werden. Die Hierarchiestufen werden mit Schrägstrich Symbolen unterteilt. Es besteht auch die Möglichkeit, mit dem Plus Symbol eine Wildcard für eine spezifische Hierarchiestufe zu abonnieren. Wenn eine Device ID im Topic mitgesendet wird (*measurement/<device_id>/temperature*) und im Backend die Temperaturwerte von allen Devices empfangen werden sollen, so kann das Topic *measurement/+/temperature* abonniert werden.

Für die Payload der Messages gibt es kein standardisiertes Format, sie werden vom Broker als Byte Array angesehen. Sie maximale Grösse, die im MQTT Standard festgelegt wurde, beträgt knappe 270 MB.

MQTT implementiert drei QoS (Quality of Service) Levels, welche eine Garantie für den Erhalt einer gesendeten Message repräsentieren. Die Abfolgen der zwischen Client und Broker ausgetauschten Pakete für die verschiedenen QoS Levels sind in Abbildung 24 visualisiert.

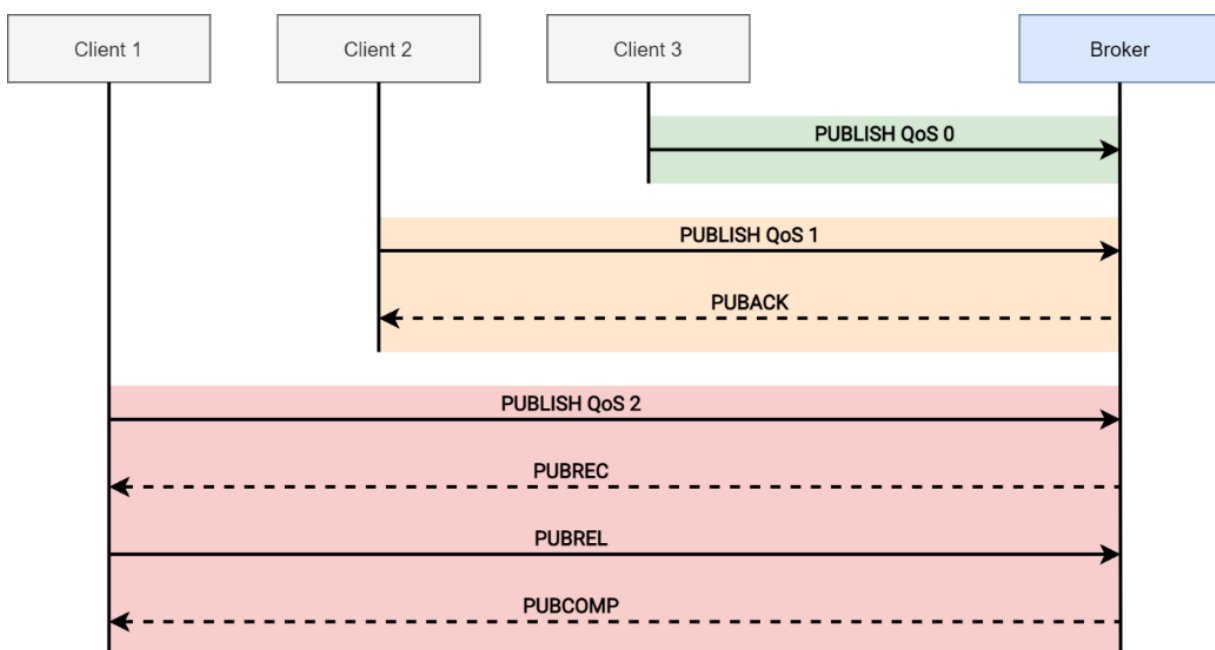


Abbildung 24 MQTT QoS

Bei QoS 0 trifft die Nachricht maximal einmal beim Empfänger ein, bei QoS 1 mindestens ein-mal und bei QoS 2: genau einmal. QoS 0 wird dann eingesetzt, wenn die Nachrichten möglichst schnell zu übermitteln sind und ein Verlust einer einzelnen Message hinnehmbar ist. Ein Einsatz ist das Übermitteln von Messwerten in kurzen Intervallen über eine stabile Verbindung. Wenn keine Nachricht verloren gehen darf, muss mindestens das Level 1 eingesetzt werden. Bei QoS 1 können Nachrichten doppelt beim Broker eintreffen, was bei QoS 2 nicht der Fall ist. Je höher die QoS gewählt wird, desto länger dauert die Übermittlung. Einige Geräte oder Integrationen unterstützen nicht alle Stufen [31].

Um den Zugriff auf den Broker zu regulieren, verfügen die meisten Broker über ACL (Access Control Lists), in welchen die Berechtigungen für das Publizieren und das Abonnieren pro Client oder für eine Gruppe von Clients definiert werden. Die Verwaltung der ACL geschieht statisch in einem File oder dynamisch via WebHooks.

Mit der im Januar 2018 veröffentlichten Version MQTT 5 können zusätzliche benutzerdefinierte Key-Value Paare mit den Nachrichten mitgeschickt werden. In den sogenannten User-Property Feldern können Metainformationen wie Content-Type, Geräte-ID oder Schemabezeichnungen definiert werden. In der Funktion sind die Felder sehr ähnlich wie die HTTP Headers [32].

MQTT basiert auf TCP Verbindungen. Für unverschlüsselte Verbindungen wird standardmässig der Port 1883 eingesetzt. Sobald TLS implementiert ist, wird der Port 8883 als Standard Port verwendet.

2.2.6.3 Weitere Protokolle

Zusätzlich zu den dokumentierten Protokollen existieren weitere Transportprotokolle, welche ihren Einsatz im IoT Bereich finden. Unter anderem die Protokolle [33]:

- AMQP, welches in der Funktionalität grosse Ähnlichkeiten zu MQTT aufweist.
- CoAP, ein Protokoll für Geräte mit geringer Leistung und Netzwerke mit geringer Bandbreite.
- OPC-UA, welches vor allem im IIoT Bereich eingesetzt wird.
- XMPP, ein sicheres Protokoll zur Nachrichtenübermittlung in dezentralen Architekturen.
- SIP, ein auf UDP basierendes, HTTP ähnliches Client/Server Protokoll.
- RTP, ein Protokoll zur Echtzeit End-to-End Übertragung von Audio und Video Sessions.

2.2.7 Datenformate

Von IoT Systemen erfasste Daten können sehr heterogen sein. Ein einfacher Temperatursensor generiert einen einzelnen Wert, ein Drei Achsen Beschleunigungssensor verfügt über Drei. Wenn Bildmaterial oder Audioaufnahmen erstellt werden, sind es schnell sehr grosse binäre Informationen. Es gibt eine Vielzahl von Formaten zur Datenübertragung. Die wichtigsten werden folgend beschrieben.

2.2.7.1 JSON

JSON heisst ausgeschrieben JavaScript Object Notation und ist ein textbasiertes, kompaktes Datenformat für den Datenaustausch zwischen Anwendungen. Die Daten werden in hierarchischer Struktur definiert und in UTF-8 kodiert. Es ist definiert in ECMA-404 [34], sehr verbreitet und ist unabhängig von Programmiersprachen oder Plattformen. Die Typen, die in JSON unterstützt werden, sind Objekte, Arrays, Nummern, Strings, True / False und Null.

Objekte beginnen und enden mit einer geschweiften Klammer ({}). In einem Objekt befinden sich Key-Value Paare, wobei der Schlüssel mit einem Doppelpunkt vom Wert getrennt ist. Der Key ist ein String, die Werte können alle unterstützen Typen sein. Mehrere Key-Value Paare werden mit Kommas getrennt, leere Objekte sind zulässig. Codeausschnitt 1 zeigt ein Beispiel einer

Nachricht in JSON Format. Sie beinhaltet ein Objekt, welches eine Device ID, ein Timestamp und eine Liste von Messwert-Objekten beinhaltet.

```
{
  "device_id": "1234-5678",
  "timestamp": "2021-11-23T18:25:43.511Z",
  "measurements": [
    {
      "type": "temperature",
      "unit": "C",
      "value": 22.34
    },
    {
      "type": "humidity",
      "unit": "RH",
      "value": 52.4
    }
  ]
}
```

Codeausschnitt 1 Nachricht im JSON Format

Leerzeichen und Zeilenumbrüche ausserhalb der Anführungszeichen werden beim Parsen ignoriert und dienen lediglich der Lesbarkeit. Um binäre Daten in einer JSON Nachricht zu übermitteln, müssen sie Base64 codiert werden, was ein Overhead von 33 bis 36 Prozent mit sich bringt.

2.2.7.2 CSV

Im Comma-Separated-Values (CSV) Format werden Werte mit Kommas getrennt. Die erste Zeile wird als Header verwendet und repräsentiert die Bezeichnung der Werte in den Spalten. Das folgende Beispiel in Codeausschnitt 2 zeigt eine CSV Nachricht, welche eine Device ID, einen Timestamp, Temperatur und Luftfeuchtigkeitswerte beinhaltet. Es ist erkennbar, dass die Device ID und der Timestamp aufgrund des Formats doppelt notiert werden.

```
"device_id","timestamp","type","unit","value"
"1234-5678","2021-11-23T18:25:43.511Z","temperature","C",22.34
"1234-5678","2021-11-23T18:25:43.511Z","humidity","RH",52.4
```

Codeausschnitt 2 Nachricht im CSV Format

Der Inhalt der Felder in einer CSV Nachricht kann ein Text oder eine Nummer sein. Leere Felder werden unterstützt. Die Verwendung von CSV ist dann sinnvoll, wenn eine grosse Anzahl von Werten in der selben Struktur vorkommen. Es können auch andere Trennzeichen als Kommas eingesetzt werden. Um binäre Daten in einer CSV Nachricht zu übermitteln, müssen sie Base64 codiert werden.

2.2.7.3 Protocol Buffers

Protocol Buffers, auch als Protobuf bekannt, ist ein von Google entwickelter Mechanismus zur Serialisierung von strukturierten Daten. Das Format der serialisierten Daten wird, identisch zum Mechanismus, als Protobuf Format bezeichnet. Im Gegensatz zu JSON, XML oder CSV, wo die Daten im Textformat übertragen werden, setzt Protobuf auf binäre Daten, was die Grösse der Nachrichten minimiert.

Die Struktur der Nachricht wird als Message Type definiert, welches in einem .proto File gespeichert wird. Ein Beispiel ist in Codeausschnitt 3 zu sehen. In der ersten Zeile wird die Version, im Beispiel «proto3», definiert. Die Nachricht besteht aus einem Objekt «Measurements», welches die Device ID, einen Timestamp und mehrere «Measurement» Objekte beinhalten kann. Die «Measurement» Objekte bestehen aus je einem Feld für Typ, Einheit und dem Wert der Messung. Jedem Feld einer Nachricht wird eine eindeutige Nummer zugewiesen. Bei verschachtelten Nachrichten müssen die Nummern in der entsprechenden Hierarchieebene eindeutig sein.

```
syntax = "proto3";

message Measurements {

    message Measurement {
        string type = 1;
        string unit = 2;
        double value = 3;
    }

    string device_id = 1;
    string timestamp = 2;
    repeated Measurement measurement = 3;
}
```

Codeausschnitt 3 Definition einer Protobuf Nachricht

Neben Skalaren Datentypen wie Integer, Double, Float, String, Bool und Bytes können verschachtelte Typen verwendet werden. Ein Beispiel ist der Typ «Measurement» in Codeausschnitt 3. Bei Verwendung der Version proto2 können den Feldern Regeln zugewiesen werden. Folgende Regeln können angewendet werden:

- Required: Die Nachricht muss dieses Feld beinhalten.
- Optional: Die Nachricht kann das Feld beinhalten, muss aber nicht.
- Repeated: Die Nachricht kann ein Feld beliebig oft beinhalten. Sie muss es aber nicht beinhalten.

Aus der definierten Struktur kann mit dem Protobuf Compiler [35] Code generiert werden. Unterstützt werden neben anderen Sprachen Java, Python, C++, C#, und Go. Mit dem generierten Code können Nachrichten serialisiert und deserialisiert werden.

Neben der geringen Nachrichtengröße liegt ein weiterer Vorteil darin, dass binäre Daten ohne eine Codierung in Base64 in eine Nachricht geschrieben werden können. Der Einsatz von Protobuf setzt jedoch voraus, dass die Struktur der Nachricht fest definiert ist und sowohl vom Sender als auch vom Empfänger behandelt werden kann.

2.3 IoT Hardware & Sensorik

Die «Things» in einem IoT System, also die IoT Devices, bestehen aus programmierbaren Controllern, Connectivity, Sensoren und / oder Aktoren. Die Wahl der Bestandteile hängt vom Zweck und den Anforderungen des jeweiligen Gerätes und der zu erfüllenden Funktion ab. Die häufigsten Anforderungen liegen in den Bereichen Energieverbrauch, Rechenaufwand und Übertragungsdistanz.

2.3.1 Mikrocontroller

Mikrocontroller, auch unter der Benennung uC, μ C oder MCU (Micro Controller Unit) bekannt, sind Halbleiterchips, welche neben einem Prozessor auch entsprechende Peripheriefunktionen und Speicher enthalten. Praktisch jedes elektronische Gerät verfügt heutzutage über eingebaute Mikrocontroller. Durch die hohe Verbreitung resultieren hohe Verfügbarkeiten und tiefe

Anschaffungspreise. Ein 8-Bit Mikrocontroller ist in Abbildung 25 abgebildet. Die Grösse des Controllers beträgt 10 x 10 mm.

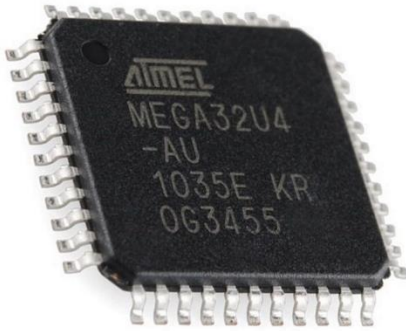


Abbildung 25 ATMEGA 8 Bit Mikrocontroller [36]

Mikrocontroller können anhand der Wortbreite der Instruktionen, welche vom Prozessor in einem Schritt verarbeitbar sind, in 4-, 8-, 16- und 32-Bit Controller unterteilt werden.

4-Bit Mikrocontroller sind sehr preiswert und eignen sich für einfache Aufgaben mit wenig Rechenaufwand. Beispiele sind die Ansteuerung eines Backofens, Uhren oder für Fernbedienungen. Für komplexere Aufgaben sind sie aufgrund ihrer geringen Leistung nicht geeignet.

8- und 16-Bit Mikrocontroller sind einfach implementierbar und aus diesem Grund sehr weit vertreten. Sie werden in einem breiten Anwendungsspektrum für wenig- bis mittelkomplexe Anwendungen eingesetzt. Die typische Arbeitsfrequenz liegt bei 8 MHz. Bei den gängigsten Modellen sind rund 30 frei belegbare Ein- oder Ausgangspin (GPIO) verfügbar. Anwendungsbeispiele sind Smart Home Geräte mit Touchscreens, Herzfrequenzmessinstrumente in der Medizintechnik oder Steuerungen in Autos. Das Arduino Uno, ein weit verbreitetes Entwicklungsboard, basiert auf einem 8-Bit Mikrocontroller.

32-Bit Mikrocontroller sind sehr leistungsstark und werden immer wie populärer. Sie verfügen über Arbeitsfrequenzen grösser als 100 MHz und genug Speicher womit sie sich zur Datenverarbeitung, beispielsweise eine FFT, eignen. Auch für eine Netzwerkkommunikation sind sie schnell genug, was sie für den Einsatz im Internet der Dinge beliebt macht. Die Anzahl der GPIO kann, je nach Modell, über 100 liegen.

2.3.1.1 Peripherie

Neben dem Prozessor besteht ein Mikrocontroller aus integrierten Peripheriemodulen. Sie umschliessen [37]:

- Interrupt Controller
- Zähler und Timer Module
- Digitale Inputs und Outputs
- Analoge Inputs, implementiert mit ADCs
- Analoge Outputs, implementiert durch PWM
- Systemintegrationsmodule
- Serielle Schnittstellen
- Netzschnittstellen
- Busschnittstellen

Je nach Anwendung, für welche der Controller entwickelt wurde, enthält er zudem weitere spezifische Schnittstellen wie anwendungsspezifische parallele Schnittstellen um Displays anzusteuern, Infrarot Schnittstellen oder USB.

2.3.1.2 Speicher

Mikrocontroller implementieren verschiedene Arten von Speicher. Der Speicher wird in Programmspeicher und Arbeits- / Datenspeicher aufgeteilt. Im Programmspeicher müssen die Informationen auch ohne Stromversorgung persistent bleiben, es wird also nicht-flüchtiger Speicher eingesetzt. Wenn das Programm mehrfach änderbar sein soll, wird EPROM oder EEPROM (Electrically Erasable Programmable Read-Only Memory) verwendet. Wenn der Controller einmalig programmiert wird, kann dafür OTP-ROM (One Time Programmable - Read-Only Memory) eingesetzt werden. Es besteht mit der Maskenprogrammierung die Möglichkeit, das Programm direkt bei der Herstellung des Controllers unveränderlich einzuarbeiten.

Der Datenspeicher liegt auf dem SRAM (Static Random Access Memory) Speicher, in welchem Daten nur bei vorhandener Versorgungsspannung speicherbar sind. Fällt diese weg, wird der Inhalt gelöscht. Alle in einem Programm generierten dynamischen Variablen werden im Arbeitsspeicher gespeichert. Es gibt Anwendungen, bei welchen gewisse dynamische Daten nicht-volatil zu speichern sind. Ein Beispiel ist eine Beleuchtungsanlage, welche über einen Mikrocontroller gesteuert wird. Wenn die Spannung kurzzeitig ausfällt soll sie, die Lampen, welche zuvor leuchteten, automatisch wieder einschalten. Dafür können gewisse Variablen im EEPROM Speicher abgelegt werden. Die Verwendung des EEPROM Speichers für dynamische Daten soll möglichst klein gehalten werden, denn die Schreibzyklen sind begrenzt [38].

2.3.1.3 Programmierung

Bei Mikrocontrollern wird die Firmware programmiert. Anders als bei Anwendungen auf einem Computer mit Betriebssystem läuft auf dem Mikrocontroller nur ein Programm. Die ersten Mikrocontroller wurden mit der sehr hardwarenahen Assemblersprache (engl. Assembly) programmiert. Heutzutage wird für die Programmierung von embedded Devices hauptsächlich C oder C++ verwendet. Ein aufkommender Trend ist der Einsatz von MicroPython, um 32-Bit Mikrocontroller zu programmieren. MicroPython ist eine in C geschriebene Implementation der Programmiersprache Python, welche weitgehend mit Python3 kompatibel ist. Die hohe Benutzerfreundlichkeit von Python geht jedoch auf Kosten der Effizienz, des erforderlichen Speicherplatzes und des Stromverbrauchs.

Im Gegensatz zu Programmen auf Betriebssystembasierten Plattformen gibt es bei Mikrocontrollern keine Scheduler, die verschiedene Prozesse oder Tasks verwalten. Die Ausführung der Prozesse muss im Programm implementiert werden. Die einfachste Struktur besteht aus einer Setup Funktion, in welcher die Hardware initialisiert wird und einer Endlosschleife. Somit besteht das Programm aus einem Task, der durchgehend läuft.

2.3.1.3.1 State Machine

Mit State Machines können Zustände in embedded Programmen verwaltet werden. Das System ist zu jedem Zeitpunkt in genau einem Zustand. Transitionen zwischen den Zuständen werden durch Inputs, Events oder Timer ausgelöst. Abbildung 26 zeigt die Implementierung einer State Machine mit der Funktionalität einer Ampel. Sie befindet sich immer in einem der Zustände Rot, Gelb oder Grün. Der Initialzustand ist Rot. Von Rot auf Grün wird beim Command Go gewechselt, von Grün auf Gelb bei Stop. Von Gelb auf Rot wird nach einem definierten Timeout gewechselt.

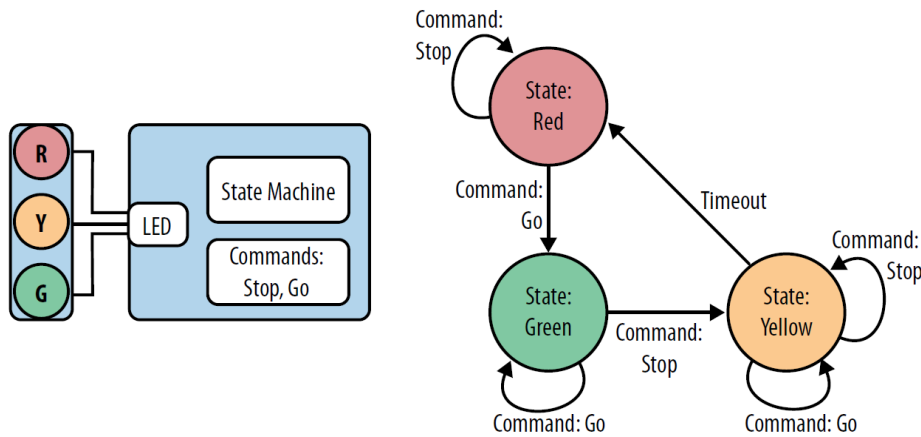


Abbildung 26 State Machine [39]

State Machines können grundsätzlich auf zwei Arten, State-Centric und Event-Centric, implementiert werden. Bei der State-Centric Implementierung wird basierend auf dem aktiven Zustand überprüft, ob ein Event vorliegt, welcher die Übergangsbedingung für den nächsten Zustand erfüllt. Bei der Event-Centric Implementierung wird beim Eintreffen eines Events geprüft, ob er einen Einfluss auf den aktiven State hat. Die Wahl der Implementierungsart ist abhängig von der zu erfüllenden Logik [39].

2.3.1.3.2 Interrupts

Eine State Machine wechselt den Zustand beim Auftreten von Events. Ein einfaches Programm kann die Inputs in einer Endlosschleife auf Events überprüfen (polling) und entsprechend behandeln. Mit Interrupts besteht die Möglichkeit, Events parallel zum laufenden Hauptprogramm zu registrieren. Sie werden benutzt, um bestimmte Tasks, wie die Erfassung von Benutzereingaben, automatisch ablaufen zu lassen oder um zeitkritische Aufgaben zu lösen. Die Funktionsweise von Interrupts ist wie folgt:

1. Ein Interrupt Request (IRQ) wird durch eine konfigurierte Soft- oder Hardware Trigger-source ausgelöst. Zu den Sources gehören Zustände oder Zustandsänderungen von digitalen Eingangspins (LOW, HIGH, CHANGE, RISING, FALLING), interne und externe Timer und der Empfang von Daten über serielle Schnittstellen. Neben den konfigurierten Sources lösen auch Systemfehler Interrupts aus.
2. Der Prozessor speichert den Kontext des Interrupts.
3. In der Interrupt Vector Table wird die dem Interrupt zugewiesene Callback Funktion gesucht.
4. Die Callback Funktion (Interrupt Service Routine, ISR) wird ausgelöst.

ISR sind spezielle, eingeschränkte Funktionen, die weder Parameter noch Rückgabewerte aufweisen. Sie müssen so schnell wie möglich sein, da der Aufruf das Hauptprogramm während der Ausführung unterbricht. Typischerweise ändern ISR globale Variablen, um Informationen mit dem Hauptprogramm auszutauschen. Bei Low-Power Mikrocontrollern, welche über Deep-Sleep Optionen verfügen, wird das Wakeup oftmals durch Interrupts mit externen oder zeitbasierten Trigger ausgelöst [39]. Codeausschnitt 4 zeigt ein Arduino Programm, welches Spannungspegeländerungen auf dem Pin 2 mit einem Interrupt registriert. Sobald der Interrupt ausgelöst wird, wird die Variable «state» von der ISR negiert. In der Endlosschleife wird ein für eine LED vorgesehener Ausgangspin gemäss der Variable ein- oder ausgeschaltet.

```

const byte ledPin = 13;
const byte interruptPin = 2;
volatile byte state = LOW;

void blink_isr() {
    state = !state;
}

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(interruptPin, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(interruptPin), blink_isr, CHANGE);
}

void loop() {
    digitalWrite(ledPin, state);
}

```

Codeausschnitt 4 Beispiel externer Interrupt [40]

2.3.1.4 System on a Chip (SoC)

Wenn neben den Funktionen des Mikrocontrollers alle für ein bestimmtes System erforderlichen Funktionen in einen Chip integriert werden, spricht man von einem System on a Chip (SoC). Additional Bestandteile umfassen Grafikprozessoren, Implementierungen von Industriebus-schnittstellen, Speichereinheiten, Kryptografie Module, Schnittstellen für Peripheriegeräte wie beispielsweise Tastaturen, Wireless Controller und mehr. Im Gegensatz zu Mikrocontrollern kann auf einem SoC ein vollumfängliches Betriebssystem laufen. Abbildung 27 zeigt ein Blockdiagramm der Funktionalitäten, welche in einem ESP32 SoC der Firma Espressif implementiert sind.

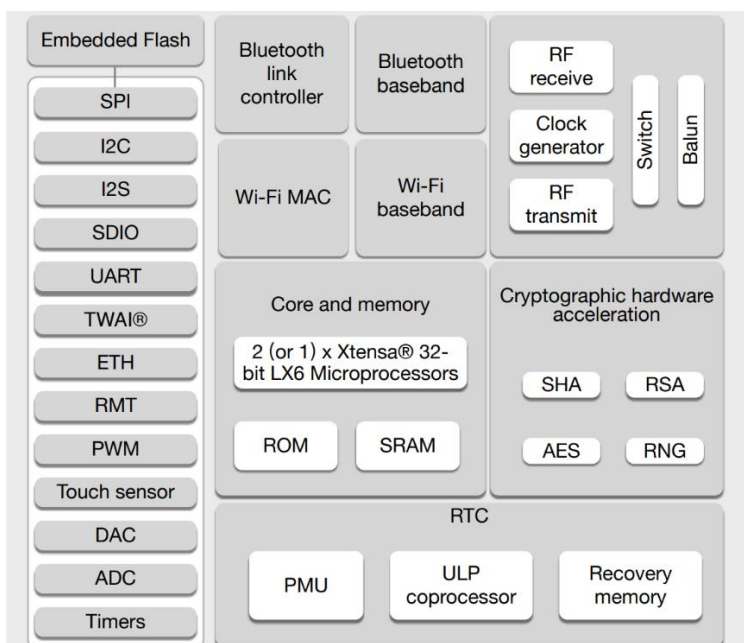


Abbildung 27 ESP32 Functional Block Diagram [41]

Auf dem Chip befinden sich neben dem Mikroprozessor Controller für WiFi, Bluetooth und Ethernet, eine Real-Time Clock, diverse Peripherieschnittstellen, embedded Flash Speicher und Kryptografische Hardwarebeschleunigungseinheiten [41].

In der Praxis werden einige Bestandteile, welche auf einem anderen Spannungslevel sind oder eine erhöhte Wärmeentwicklung verursachen oftmals bewusst ausserhalb des Chips implementiert [37].

2.3.2 Single Board Computer (SBC)

Als Single Board Computer werden ein lauffähige, auf einer Platine basierende, Computer bezeichnet. Auf der Platine befindet sich ein oder mehrere SoC, Speichermodule und Anschlussbuchsen für Peripheriegeräte. Der wohl bekannteste SBC ist der Raspberry Pi, welcher in Abbildung 28 ersichtlich ist. Das Modell 4 basiert auf einem Broadcom Quad Core 64-Bit ARM SoC, in welchem auch die GPU implementiert ist. Er ist in rot markiert. Der Arbeitsspeicher befindet sich ausserhalb des SoC (blau markiert) und umfasst 2, 4 oder 8 GB. Der gelb markierte Wireless Controller unterstützt WiFi AC und Bluetooth 5.0. Zudem umfasst das Board einen Gigabit Ethernet Controller (violett markiert) und 40 GPIO. Die USB-C Buchse dient zur Spannungsversorgung. Alternativ kann der SBC auch über den GPIO Header oder mit einem PoE Modul mit Spannung versorgt werden. Die Anschlüsse für Peripheriegeräte umfassen HDMI, USB 3, Camera- und Display Schnittstellen, RJ45 und eine 3.5 mm Audiobuchse.

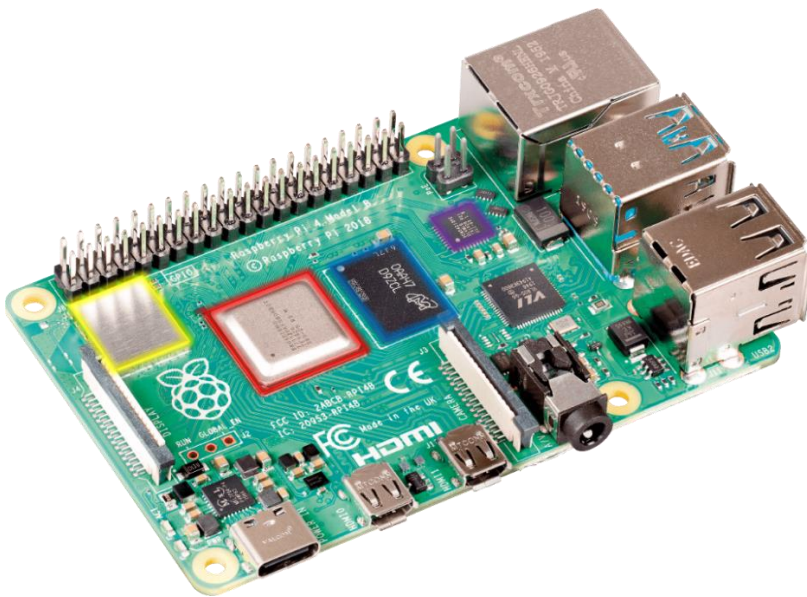


Abbildung 28 Raspberry Pi 4 [42]

SBC sind für den Einsatz eines Betriebssystems prädestiniert. Oftmals werden Linux-Basierte Betriebssysteme eingesetzt, die für SBC optimiert sind. Für die Raspberry Pi wurde das auf Debian basierende Open Source Betriebssystem Raspberry Pi OS entwickelt. Raspberry Pi OS ist mit der Hardware vollumfänglich kompatibel und gilt als benutzerfreundlich [43].

2.3.3 Connectivity

Eine essenzielle Fähigkeit für IoT Geräte ist die Connectivity, um Daten zu übermitteln und zu empfangen. Die Kommunikation kann kabelgebunden oder drahtlos erstellt werden. Da IoT Devices oftmals mobil sind, ist der Einsatz von Wireless Verbindungen in vielen Fällen unerlässlich. Es existieren verschiedene Technologien mit unterschiedlichen Eigenschaften wie Energieverbrauch, Durchsatz und Reichweite. Abbildung 29 zeigt die Eigenschaften der wichtigsten Technologien, welche folgend beschrieben werden.

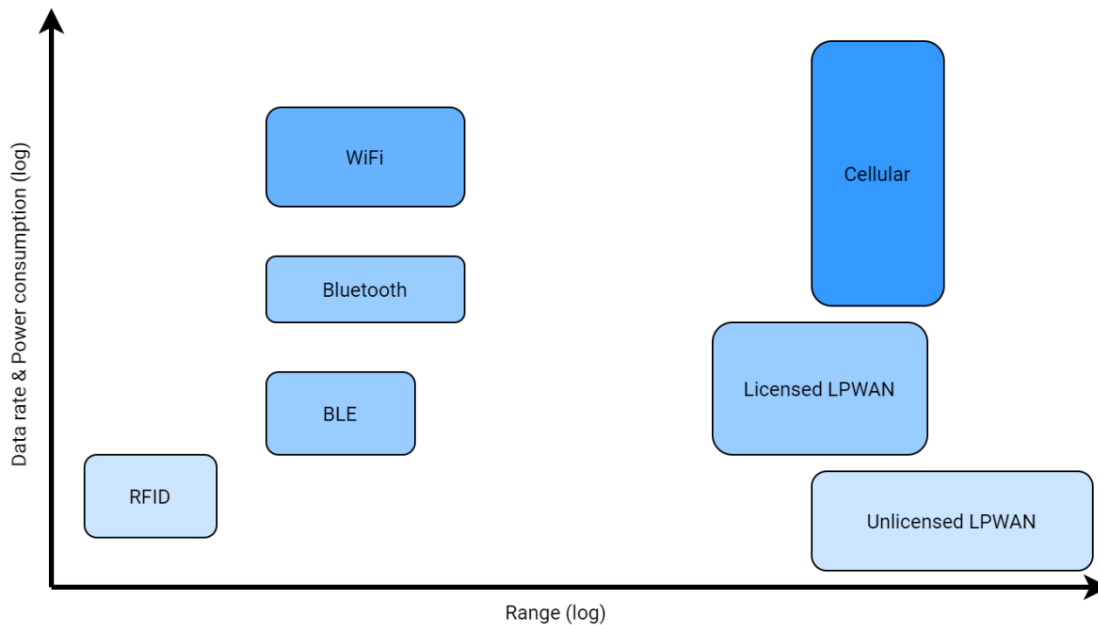


Abbildung 29 Wireless Technologien im Vergleich

2.3.3.1 WLAN

Die wohl bekannteste Drahtlostechnologie im Netzwerkbereich ist WLAN (Wireless Local Area Network). Die erste Version wurde im Jahre 1997 entwickelt und hatte einen Maximaldurchsatz von 2 Mbit/s. Die neueste Version, 802.11ax (Wi-Fi 6) verfügt über einen theoretischen Maximaldurchsatz von 9.6 Gbit/s. Die Übertragungsfrequenz beträgt 2.4 oder 5 GHz. 5 GHz Verbindungen sind schneller, haben aber Limitierungen in der Reichweite und werden durch Hindernisse stärker abgeschwächt. Tabelle 4 zeigt die Eigenschaften der aktuellen WLAN Standards.

Generation	Wi-Fi 4	Wi-Fi 5	Wi-Fi 6
IEEE-Standard	IEEE 802.11n	IEEE 802.11ac	IEEE 802.11ax
Realistische Übertragungsrate	300 MBit/s	867 MBit/s	1200 MBit/s
Max. Reichweite	100 m	50 m	50 m
Max. Bandbreite	40 MHz	160 MHz	160 MHz
Modulationsverfahren	64QAM	256QAM	1024QAM

Tabelle 4 Vergleich aktueller WLAN Standards [44]

Die maximale Reichweite von Wi-Fi ist abhängig von der Sendeleistung und erreicht rund 100 Meter. Wie der Durchsatz und die Reichweite ist auch der Energieverbrauch abhängig von der Sendeleistung. Er kann relativ hoch werden, womit der Einsatz von Wi-Fi bei Low-Power Anwendungen suboptimal ist.

Clients können sich bei der Anmeldung durch einen Pre-shared Key (PSK) authentifizieren. Die PSK Authentifizierung vor Allem in simplen Netzwerken eingesetzt. Mit der Verwendung von IEEE 802.11x können Clientprofile zentral verwaltet werden. Die Authentifizierung geschieht durch die Eingabe eines Benutzernamens und Passworts.

2.3.3.2 Bluetooth

Bluetooth ist eine Datenübertragungstechnologie für kurze Distanzen und wurde in den 1990er Jahren mit dem Ziel, Kabel zu ersetzen, entwickelt. Die Übertragungsfrequenz liegt mit 2.4 GHz im lizenzfreien Spektrum. Bluetooth gilt als robust, energieeffizient und preiswert. Bezüglich des Energieverbrauchs wurden drei Klassen definiert, die in Tabelle 5 ersichtlich sind.

Energieklasse	Max. Sendeleistung	Reichweite
1	100 mW	100 m
2	2.5 mW	10 m
3	1 mW	1 m

Tabelle 5 Bluetooth Energieklassen

Für Smartphones, Notebooks und Wearables wird typischerweise Klasse 2 eingesetzt. Klasse 1 wird hauptsächlich in industriellen Anwendungen implementiert. Die maximale Datenrate wird mit der 8DPSK Modulation erreicht und beträgt 3 Mbit/s. Im Frequenzband sind 79 Kanäle mit jeweils 1 MHz Bandbreite verfügbar. Für die Nutzung der Kanäle wird Frequency-Hopping Spread Spectrum (FHSS) verwendet [45]. Unterstützte Topologien sind Punkt-zu-Punkt Verbindungen und Piconets.

2.3.3.3 BLE

Bluetooth Low Energy (BLE) ist ein Zusatz zu der Bluetooth Spezifikation, welcher für den Einsatz im Low-Power IoT Bereich taugt. Durch die Unterstützung von kleineren Datenraten wird der Energieverbrauch im Vergleich zu der ersten Bluetooth Spezifikation (Bluetooth classic) erheblich minimiert. Die Datenrate liegt zwischen 125 Kbit/s und 2 Mbit/s. Für BLE stehen 40 Kanäle mit jeweils 2 MHz Bandbreite zur Verfügung, wovon drei davon für das Advertising genutzt werden. Neben Punkt-zu-Punkt werden Broadcast und Mesh Topologien unterstützt.

Ein BLE Device hat die Rolle des Masters (Central) oder eines Slaves (Peripheral). Ein Peripheral sendet periodisch Advertising Packets, welche bereitgestellte Daten beinhalten. Eine Central führt periodische Scans durch, um Peripherals zu entdecken. Sobald ein Peripheral entdeckt wurde, kann sich die Central darauf verbinden und auf die Daten zugreifen. Die Daten sind in Services und Characteristics strukturiert. Daten auf einem Peripheral können von der Central gelesen oder geschrieben werden. Abbildung 30 zeigt den Ablauf eines Lese- und Schreibvorgangs zwischen der Central und Peripheral 2. Durch die Aktivierung von Notifications werden bestimmte Werte eines Peripheral abonniert. Der Wert wird danach aktiv vom Peripheral zu der Central gesendet. Der Ablauf der Aktivierung ist in Abbildung 30 zwischen der Central und Peripheral 1 ersichtlich.

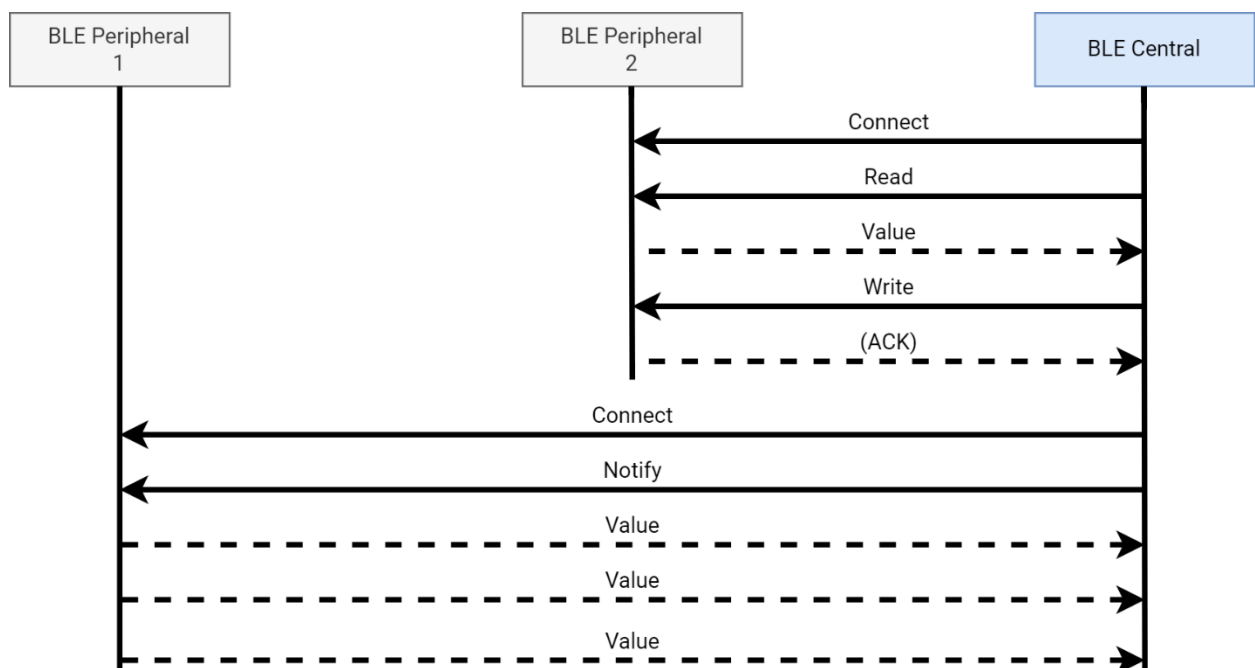


Abbildung 30 BLE Read / Write / Notify

2.3.3.4 LPWAN

LPWAN (Low Power Wide Area Network) beschreibt eine Gruppe von Übertragungsprotokollen für stromsparende Anwendungen. In der Regel arbeiten sie im Sub-Gigahertz Bereich mit kleinen Bandbreiten, wodurch grosse Reichweiten erreicht werden. Die Datenrate ist gering und wird oftmals auf wenige Bytes pro Nachricht begrenzt. Es existieren lizenzierte und unlizenzierte LPWAN. Bei lizenzierten Übertragungsfrequenzen und können die Netzwerke kostenpflichtig genutzt werden, die Infrastruktur wird typischerweise durch Mobilfunkanbieter bereitgestellt. Die bekanntesten Typen von lizenzierten LPWAN sind Narrowband IoT (NB-IoT) und LTE-M. Die populärsten Technologien im unlizenzierten Bereich sind LoRaWAN und Sigfox.

LoRaWAN

Die LoRa (Long Range) Funktechnologie wurde von Semtech entwickelt und ist eine der am häufigsten eingesetzten Connectivity Technologie im IoT Bereich. Die Arbeitsfrequenz liegt bei 433 oder 868 MHz. Die Übertragungsrate liegt unter 50 Kbit/s. LoRaWAN ist ein auf LoRa basierendes Netzwerkprotokoll, welches den Zugriff auf das Medium regelt (MAC). Die verwendete Topologie ist sternförmig. LoRa Geräte (Nodes) werden in drei Klassen aufgeteilt:

- Klasse A (All): Eingesetzt für batteriebetriebene Sensoren und Aktoren. Sie wird von allen LoRa Nodes unterstützt. Das Senden der Daten hat Priorität, Empfangen ist möglich. Hohe Latenzzeiten werden in Kauf genommen.
- Klasse B (Beacon): Eingesetzt für batteriebetriebene Aktoren, welche Daten Nachrichten empfangen. Die Kommunikation ist zeitgesteuert, der Energieverbrauch ist höher als bei Klasse A.
- Klasse C (Continuous): Wird eingesetzt für Netzbetriebene Aktoren, welche jederzeit Daten empfangen können. Die Latenzzeiten beim Downlink sind minimal, der Energieverbrauch am höchsten.

LoRa verwendet auf dem Physikalischen Layer die Chirp-Spread-Spectrum (CSS) Modulationstechnik. Bei CSS wird die Frequenz beim Übertragen inkrementiert oder dekrementiert. Abbildung 31 zeigt die Übertragung eines Bits mit dem Wert «1». Die ansteigende Frequenz ist in der FFT, welche auf der rechten Seite abgebildet ist, gut zu erkennen.

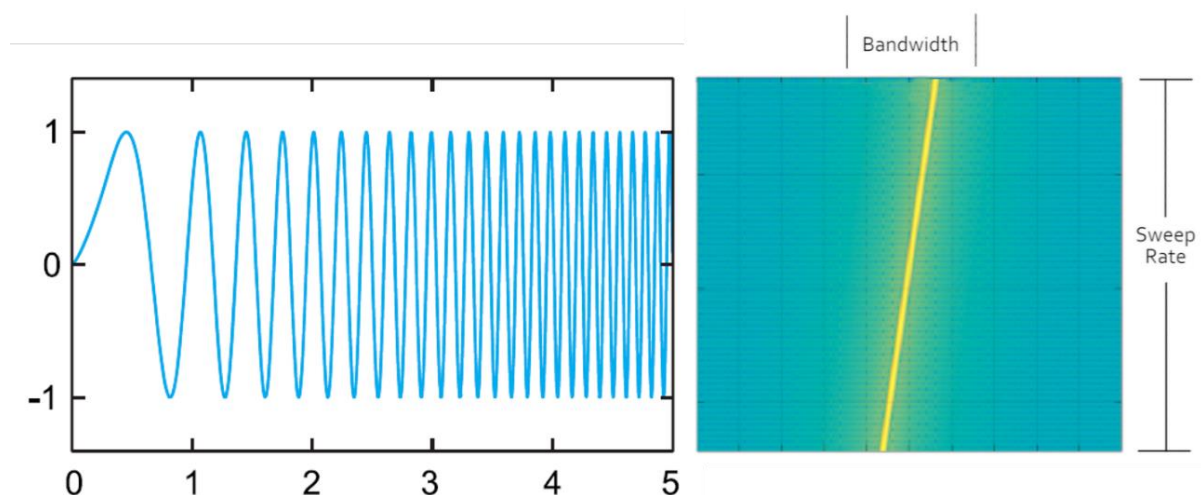


Abbildung 31 CSS Modulation [46] [47]

Die Sweep Rate bezeichnet die Zeit, welche für das In- oder Dekrementieren der Frequenz gebraucht wird. Sie ist proportional zu der Übertragungsrate und bestimmt die Reichweite. Üblicherweise wird die Bezeichnung Spreading Factor verwendet, um die Sweep Rate zu beschreiben. Im europäischen Raum sind sechs verschiedene Spreading Factors definiert, welche die Bezeichnungen SF7 bis SF12 tragen.

Abbildung 32 zeigt die maximale Bit Rate und die Empfangsempfindlichkeit der Kategorien im 125 KHz Bandbreite. Die zum Übermitteln benötigte Energie ist bei SF7 am kleinsten, da die Sendezeit am kürzesten ist.

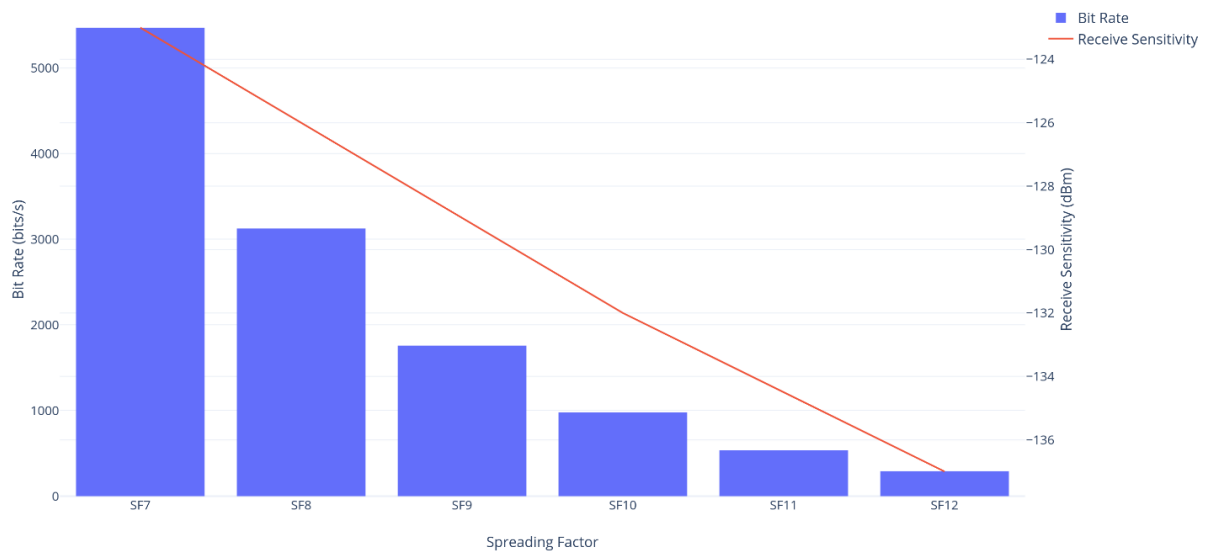


Abbildung 32 LoRa Spreading Factors [48]

Das grösste Community LoRaWAN ist The Things Network (TTN). Das Projekt wurde im Jahr 2015 gestartet und hat eine Grösse von über 150'000 Mitglieder in 151 Ländern erreicht. Die Infrastruktur besteht aus rund 22'000 aktiven Gateways, über welche Daten empfangen und gesendet werden. Die Nutzung des Netzwerks ist kostenlos. Die Abdeckung wird durch das Hinzufügen von Gateways durch die Community ständig erhöht. Die Schweiz hat eine der höchsten Netzwerkabdeckungen bezogen auf die Gesamtfläche. Zürich bildet mit 231 Gateways die weltweit grösste Subcommunity.

Die Architektur des TTN ist in Abbildung 33 schematisch dargestellt. Endgeräte (Nodes) senden Nachrichten an die durch die Community bereitgestellten Gateways. Über den Network Server werden die Nachrichten an den Application Server weitergeleitet und sind dort durch Integrationen für die weitere Verwendung verfügbar. Die Integrationsmöglichkeiten umfassen MQTT, Webhook, AWS IoT und Azure IoT Hub. Über eine Application können auch Nachrichten an die Geräte übermittelt werden. Die Nachrichten sind auf der Strecke zwischen Endgerät und Application Server mit dem AES-128 Algorithmus verschlüsselt. Danach wird eine TLS Verschlüsselung eingesetzt.

Endgeräte werden durch Over-The-Air-Activation (OTAA) oder durch Personalisierung (ABP) aktiviert. Bei OTAA wird dem Device dynamisch eine Adresse zugewiesen, was die Methode sicherer aber auch komplizierter macht. ABP nutzt statisch definierte Adressen und Keys, welche auf dem Endgerät gespeichert werden. Eine Aktivierung durch ABP ist schneller aber weniger sicher als OTAA. TTN definiert im Rahmen einer Fair Use Policy eine Limitierung von 30 Sekunden Sendezeit pro Tag und Endgerät im Community Netzwerk. Zudem wird ein Maximum von 10 Downlink Nachrichten pro Tag festgelegt [49].

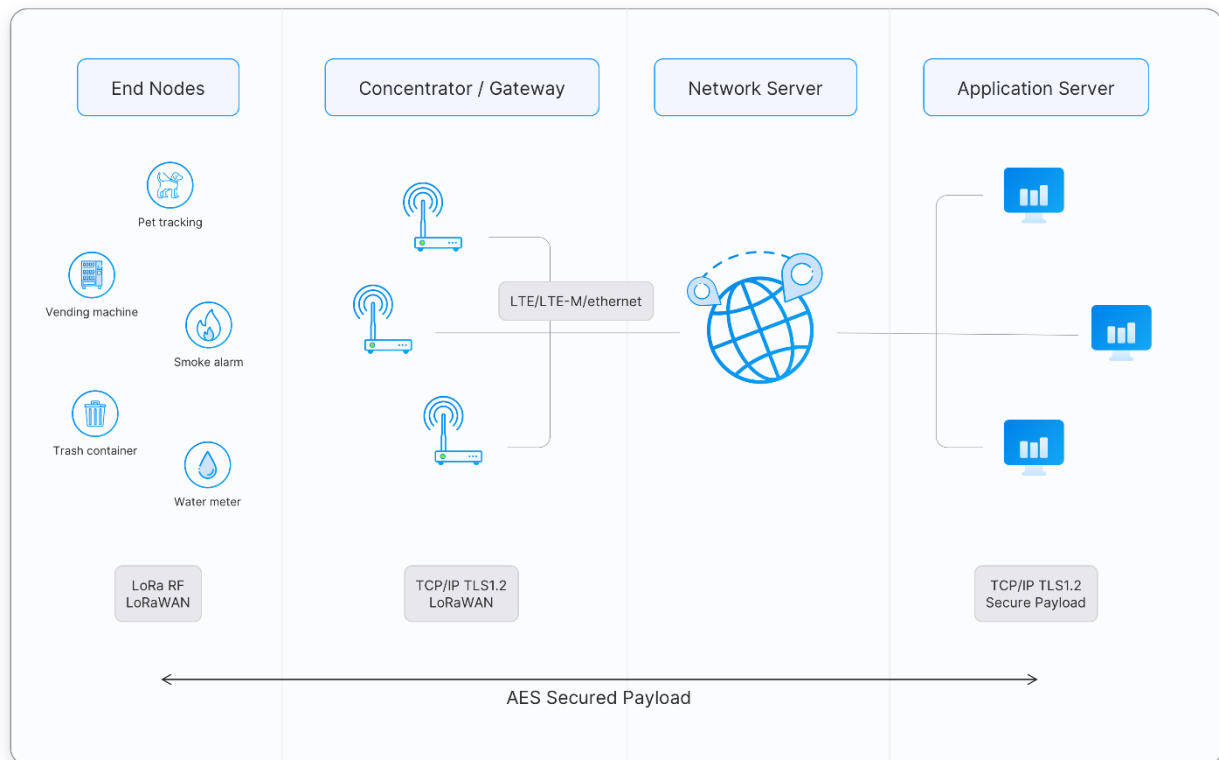


Abbildung 33 TTN LoRaWAN Architecture [50]

Neben LoRa gibt es im unlizenziierten Spektrum das Sigfox LPWAN, welches von der gleichnamigen Firma mit Hauptsitz in Frankreich entwickelt wurde. Die Infrastruktur wird von der Firma bereitgestellt, die Nutzung ist kostenpflichtig. In den letzten Jahren nahm die Popularität von Sigfox ab. Gründe dafür sind zum einen Markteintritte neuer Technologien wie NB-IoT, zum anderen die wesentlich schlechtere Performance der Sigfox Technologie in den USA. Der Unterschied liegt im höheren Frequenzband, welches in den USA genutzt wird [51].

NB-IoT

Narrowband IoT (NB-IoT) ist ein Add-on zu bestehenden GSM/LTE Netzwerken und arbeitet im lizenzierten Frequenzspektrum. Wie auch bei LoRa werden mit geringem Energieverbrauch kurze Nachrichten versendet. Als Übertragungsfrequenz wird typischerweise 800 oder 900 MHz eingesetzt, was eine hohe Abdeckung ermöglicht. Theoretisch können aber alle Mobilfunkfrequenzen dafür eingesetzt werden. Die Nutzung von NB-IoT ist kostenpflichtig und setzt eine SIM Karte voraus. Die maximale Bitrate ist grösser als bei LoRa, was einen erhöhten Energieverbrauch verursacht.

Die Entwicklung von NB-IoT verfolgt das Ziel, Connectivity Möglichkeiten für Milliarden von Geräten bereitzustellen. Da die Technologie als Zusatz zu der bereits bestehenden Mobilfunkinfrastruktur genutzt wird, beträgt die Netzabdeckung nahezu 100% der bevölkerten Fläche. Das hat den Vorteil, dass vernetzte Produkte, beispielsweise eine Waschmaschinen mit Anomalie Erkennung, mit einem Sendemodul ausgestattet werden können und davon ausgehen kann, dass sie Empfang haben. Die Zuverlässigkeit des Netzwerks ist sehr hoch und wird in Verträgen festgelegt.

Die NB-IoT Frequenzbänder mit einer Carrier Bandbreite von 200 kHz können vom Anbieter auf verschiedene Arten implementiert werden. Bei der Standalone Implementierung wird ein GSM Band durch NB-IoT ersetzt. Da die Nutzung der Frequenzen innerhalb eines LTE Bands flexibel ist, kann mit sich die Frequenz bei der In-Band Implementierung innerhalb eines LTE Bands befinden. Bei der Wahl einer Frequenz zwischen zwei LTE Bändern spricht man von der Guard-Band Implementierung. Eine Übersicht der verschiedenen Arten ist in Abbildung 34 ersichtlich.

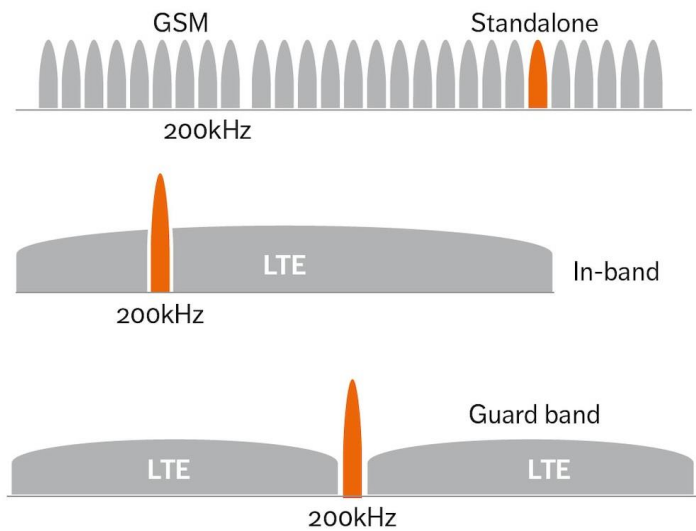


Abbildung 34 NB-IoT spectrum options [52]

2.3.3.5 Cellular

Sobald die Datenmengen ansteigen und eine hohe Mobilität erforderlich ist, können LPWAN Technologien oder WiFi nicht mehr eingesetzt werden. Ein Mobilfunknetz ist «always on» und besteht aus mehreren Antennen, welche jeweils für eine Zelle abdecken. Abbildung 35 visualisiert den Aufbau. Durch die Zellenförmige Anordnung können Frequenzen F1 – F4 im Netzwerk mehrfach verwendet werden.

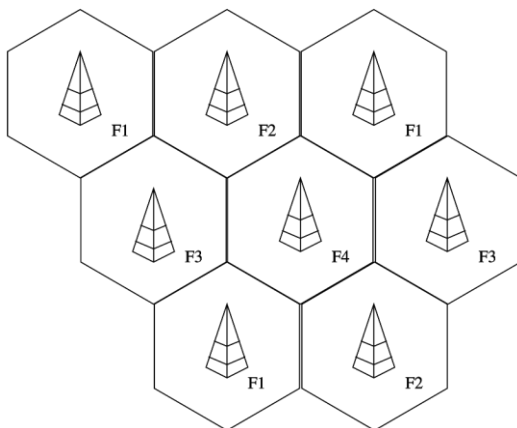


Abbildung 35 Cellular network cells [53]

Wie bei Smartphones kann für IoT Devices oder Gateways über 2G, 3G, 4G oder 5G eine Internetverbindung erstellt werden. Für die Verbindung fallen Kosten an. Tabelle 6 zeigt die Eigenschaften der aktuell eingesetzten Standards 3G, 4G und 5G.

Generation	Release	Maximale Datenrate (Theoretisch)	Latenzzeit
3G	2002	42 Mbit /s	>100 ms
4G	2009	1000 Mbit /s	20-30 ms
5G	2019	10 Gbit/s	<10 ms

Tabelle 6 Vergleich 3G, 4G und 5G [54]

In der Schweiz sind je nach Netzbetreiber bis zu 99 % der Fläche mit LTE (4G) abgedeckt [55]. Mit 5G sind sehr hohe Durchsätze möglich, was die Technologie vor Allem für die Übertragung von Bild- und Audiomaterial interessant macht. Die Abdeckung in der Schweiz beschränkt sich aktuell auf stärker besiedelte Gebiete. Abbildung 36 zeigt die 5G Abdeckung in der Schweiz. Die

gelb hinterlegten Flächen verfügen über eine Abdeckung von mindestens zwei Anbietern, die grünen von mindestens drei Anbietern.

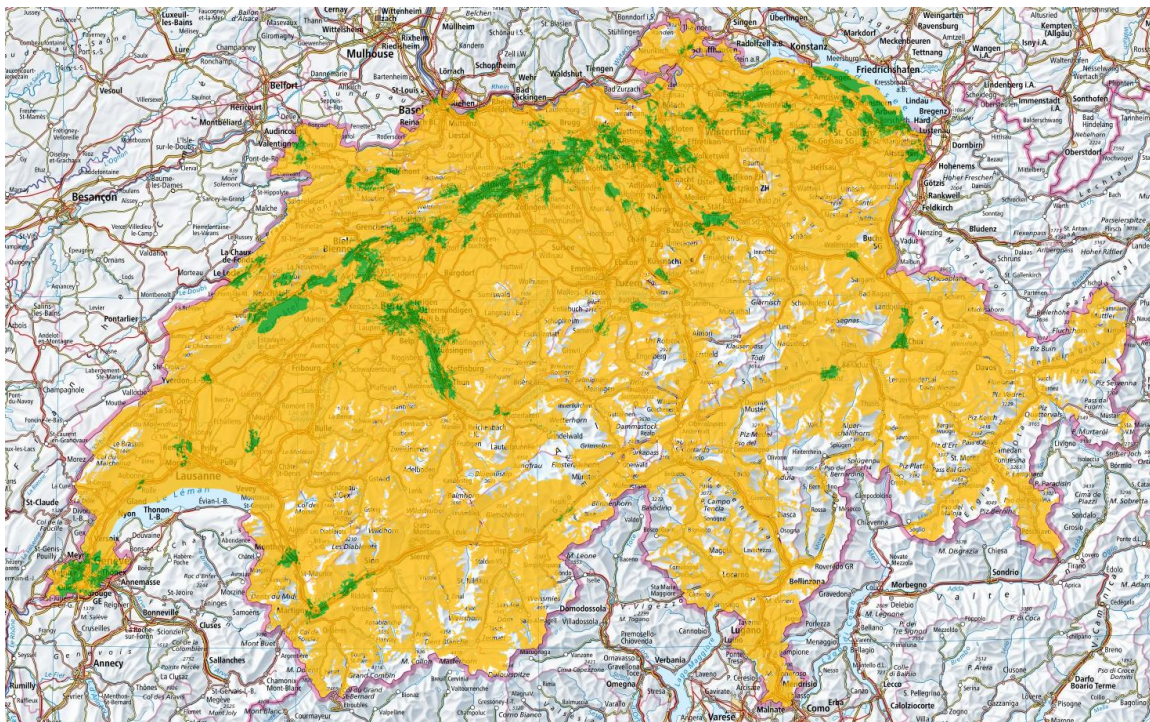


Abbildung 36 Abdeckung 5G in der Schweiz [56]

Der Energieverbrauch ist verglichen mit anderen Technologien hoch. Eingesetzt werden Mobilfunkverbindungen bei mobilen Systemen mit grösseren zu übertragenden Datenmengen [57].

2.3.3.6 Wired

Kabelgebundene Verbindungen werden bei stationären Systemen eingesetzt. Sie sind zuverlässiger als Drahtlosverbindungen, da sie weniger anfällig auf Störungen sind. Zudem sind sie schneller und einfacher zu implementieren. Die populärste Technologie ist Ethernet. Ethernet befindet sich auf OSI Layer 1 und 2. Als Medium werden Kupferkabel oder Lichtwellenleiter eingesetzt. Die Übertragungsgeschwindigkeiten liegen zwischen 100 Mbit/s und mehreren Gbit/s. Im Gegensatz zu Drahtlossystemen fallen Kosten für das Übertragungsmedium an.

Mit Power over Ethernet (PoE) ist es möglich, die Energiezufuhr und die Kommunikationsverbindung über ein einzelnes Kabel zu erstellen. Die bereitgestellte Energie ist limitiert durch die PoE Klasse. In der höchsten Klasse sind bis zu 25.5 Watt möglich. Die Ausgangsspannung liegt zwischen 36 und 57 V DC. Es muss Gleichstrom verwendet werden, da anderenfalls die Kommunikationsverbindung gestört würde. Typische PoE Endgeräte sind Wireless Access Points, IP Kameras oder Systeme zur Zugangskontrolle wie Batch Leser. Die Energieversorgung eines Kabels wird durch einen PoE Injektor erstellt. Bei mehreren Kabeln kann ein PoE Switch eingesetzt werden.

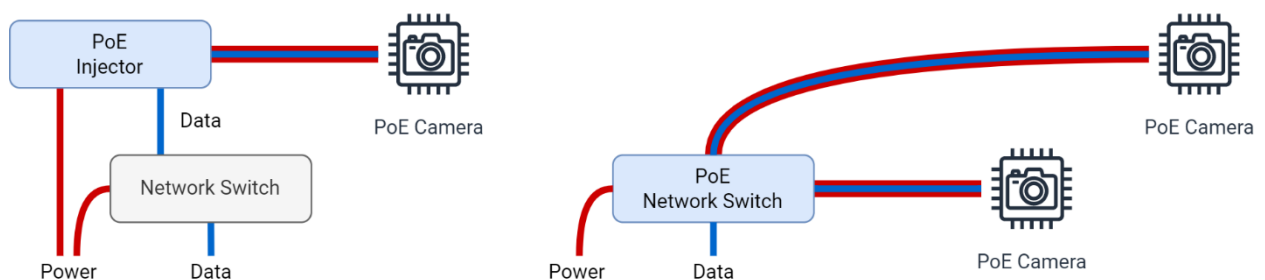


Abbildung 37 PoE Systemaufbau

Abbildung 37 zeigt auf der linken Seite ein System mit einem Endgerät, welches über einen PoE Injektor mit Energie versorgt wird. Auf der rechten Seite wird ein PoE Switch eingesetzt, welcher zwei Endgeräte versorgt.

2.3.4 Sensorik

Um Daten zu erfassen, wird in IoT Systemen eine Vielzahl von Sensortypen eingesetzt. Sensoren übersetzen eine reale physikalische Grösse in einen Wert. So divers wie die Anwendungsbereiche der IoT sind auch die Sensoren. Sensoren werden gemäss dem Ausgangssignal klassifiziert.

2.3.4.1 Temperatur

Temperatursensoren sind weit verbreitet, da die Temperatur einen grossen Einfluss auf viele Systeme hat. Es existieren vier verbreitete Technologien, um Temperatur zu messen.

PTC (Positive Temperature Coefficient) Temperatursensoren, auch als Kaltleiter bezeichnet, erhöhen ihren Widerstand bei Erhöhung der Temperatur. Ein verbreiteter Werkstoff ist Platin, der eine zur Temperatur nahezu lineare Widerstandsänderung aufweist. Platin Messwiderstände haben typischerweise 100 Ω oder 1000 Ω Referenzwiderstand (Bei Temperatur von 0 $^{\circ}\text{C}$). Sie tragen die Bezeichnung PT100 und PT1000. Die maximal messbaren Temperaturen liegen, je nach Implementierung, zwischen -200 $^{\circ}\text{C}$ und 850 $^{\circ}\text{C}$. Für die Beschreibung der Genauigkeit wurden die Klassen AA bis C definiert, wobei sich die Messabweichung von ± 0.1 $^{\circ}\text{C}$ bis ± 0.6 $^{\circ}\text{C}$ erstreckt. Durch die hochwertigen Materialien liegt der Preis im mittleren Bereich.

NTC (Negative Temperature Coefficient) Thermistoren sind Heissleiter. Der Widerstand nimmt bei Temperaturerhöhung ab. Im Vergleich zu PTC haben NTC eine höhere Nichtlinearität, was den maximalen Messbereich bei ca. 150 $^{\circ}\text{C}$ festlegt. Die Messabweichung liegt, je nach Messbereich, zwischen ± 1.5 $^{\circ}\text{C}$ und ± 0.05 $^{\circ}\text{C}$. NTC Temperatursensoren gelten als die preiswertesten Sensoren, um Temperatur zu messen.

Semiconductor Temperatursensoren sind auch unter der Bezeichnung IC-Temperatursensor bekannt. Sie bestehen aus einem Festkörperschaltkreis, welcher zwei verschiedene Bandabstandreferenzen (bandgap-generated voltage sources) beinhaltet. Zwischen den Bandabstandsreferenzen entsteht eine Spannungsdifferenz, welche direkt proportional zur anliegenden Temperatur ist [58]. IC-Temperatursensoren gibt es mit analogen und digitalen Schnittstellen. Digitale Schnittstellen sind typischerweise I2C oder 1-Wire. Abbildung 38 zeigt einen 1-Wire Temperatursensor des Herstellers Dallas. Auf der rechten Seite befindet sich der Sensor in einem wasserdichten Gehäuse mit Anschlusskabel.

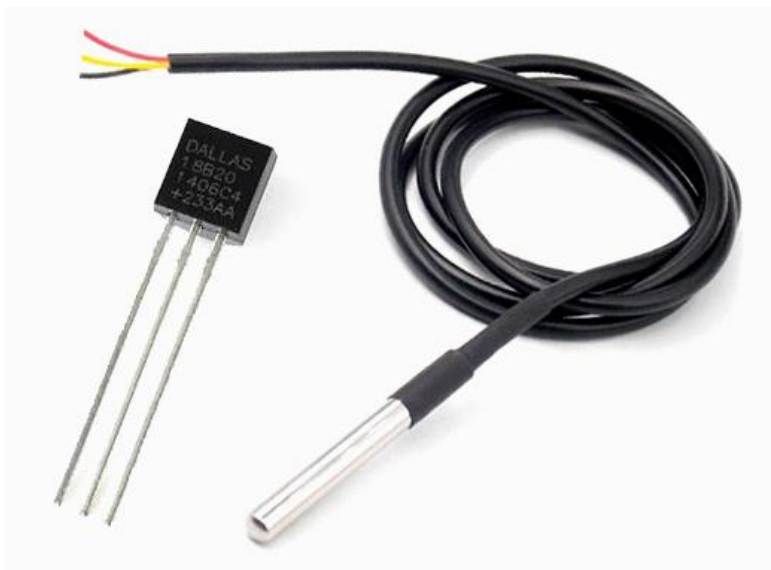


Abbildung 38 DS18B20 1-Wire Temperature Sensor [59]

Der Messbereich des Sensors in Abbildung 38 erstreckt sich von -55 °C bis 125 °C , die Genauigkeit im Messbereich von -10 °C bis 85 °C beträgt $\pm 0.5\text{ °C}$ [60]. Die Anschaffungskosten sind niedrig.

Eine weitere Möglichkeit zur Erfassung der Temperatur sind Infrarotthermometer, welche jedoch nur für spezifische Anwendungen eingesetzt werden. Zur Messung von sehr hohen Temperaturen werden Thermoelemente eingesetzt [61].

2.3.4.2 Luftfeuchtigkeit

Als Luftfeuchtigkeit wird der Anteil von Wasserdampf am Gasgemisch der Luft bezeichnet. Wasser im flüssig- oder festem Zustand wird nicht einberechnet. Die Sättigung ist abhängig von der Temperatur. Wenn die Sättigung maximal ist, beträgt die relative Luftfeuchtigkeit (RH) 100 %. Über der maximalen Sättigung befindet sich der Taupunkt. Ein beliebter Luftfeuchtigkeitssensor ist das Modell DHT11, welches in Abbildung 39 abgebildet ist. Neben der relativen Luftfeuchtigkeit misst er die Temperatur. Die maximale Messabweichung der Luftfeuchtigkeit liegt bei $\pm 5\%RH$, der Messbereich liegt zwischen 5 und 95 %RH [62]. Er verfügt über eine digitale Schnittstelle und ist preiswert.

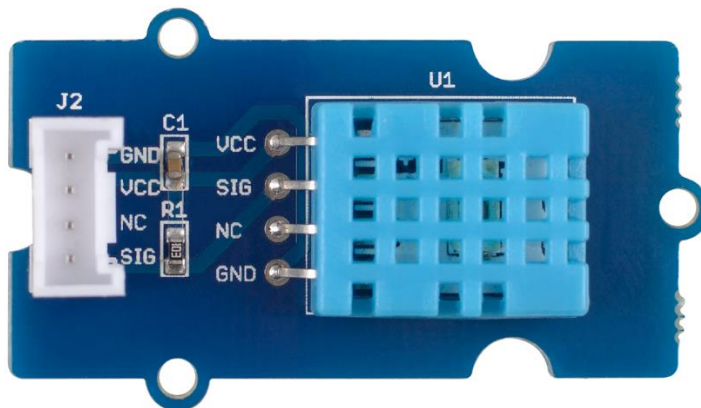


Abbildung 39 DHT11 Temperature- and Humidity Sensor [62]

2.3.4.3 Bodenfeuchtigkeit

Die Feuchtigkeit des Bodens hat eine direkte Auswirkung auf das Wohlergehen von Pflanzen. Es gibt zwei einfache Möglichkeiten, um sie zu bestimmen. Mit der resistiven Methode wird die Leitfähigkeit zwischen zwei in den Boden gesteckten Leitern gemessen. Im Wasser befinden sich elektrisch leitbare Partikel, welche den Widerstand beeinflussen. Abbildung 40 zeigt ein Sensormodul zur resistiven Feuchtigkeitsmessung. Durch das Exponieren der Kontakte ist es korrosionsanfällig.

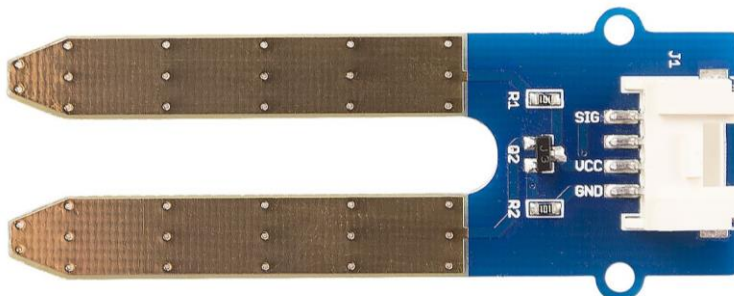


Abbildung 40 Resistive Soil Moisture Sensor [63]

Kapazitive Sensoren messen die elektrische Kapazität in der Erde. Die Feuchtigkeit hat einen direkten Einfluss auf die Kapazität und kann somit bestimmt werden. Abbildung 41 zeigt einen kapazitiven Sensor für die Messung der Bodenfeuchtigkeit. Er hat gegenüber dem resistiven Sensor den Vorteil, dass er mit der feuchten Erde nicht direkt in Kontakt sein muss.



Abbildung 41 Capacitive Soil Moisture Sensor [64]

Bei beiden Messmethoden ist es schwierig, den gemessenen Wert in eine physikalische Größe umzuwandeln. Typischerweise wird mit Referenzwerten gearbeitet und nur die relative Veränderung betrachtet.

2.3.4.4 Luftqualität

Als Luftqualität wird die Beschaffenheit der Luft mit Einbezug der darin enthaltenen Verunreinigungen bezeichnet. Zu den Verunreinigungen gehören Ozon (O_3), Feinstaub (PM2.5, PM10), Stickstoffdioxid (NO_2), Schwefeldioxid (SO_2) und Kohlenmonoxid (CO).

Für die Erfassung der Konzentration bestimmter Gase in der Luft existiert eine Vielzahl an Sensoren. Sie bestehen aus einem Heizelement und einem Chemiresistor, der Strom leitet. Der Chemiresistor ist ein Halbleiter, welcher freie Elektronen hat. Sauerstoff zieht die freien Elektronen an, wodurch sie an die Oberfläche des Chemiresistors gedrückt werden und somit die Leitfähigkeit erhöhen. Das zu messende Gas reagiert mit den absorbierten Sauerstoffpartikeln und löst die Elektronen von der Oberfläche. Die Leitfähigkeit nimmt mit zunehmender Konzentration des zu messenden Gases ab [65]. Abbildung 42 zeigt einen Sensor für die Erfassung der Konzentration von LPG und Butan.



Abbildung 42 MQ-6 Gas Sensor [65]

Mit der Sensorreihe MQ sind viele Gase oder Gruppen von Gasen messbar. Tabelle 7 gibt einen Überblick über die verschiedenen MQ Gassensoren. Die Preise variieren je nach Typ. Die Schnittstelle ist analog.

Modell	Messbare Gase
MQ-2	Methane, Butane, LPG, smoke
MQ-3	Alcohol, Ethanol, smoke
MQ-4	Methane, CNG Gas
MQ-5	Natural gas, LPG
MQ-6	LPG, butane gas
MQ-7	Carbon Monoxide
MQ-8	Hydrogen Gas
MQ-9	Carbon Monoxide, flammable gasses.
MQ131	Ozone
MQ135	Air Quality (Benzene, Alcohol, smoke)
MQ136	Hydrogen Sulfide gas
MQ137	Ammonia
MQ138	Benzene, Toluene, Alcohol, Acetone, Propane, Formaldehyde gas, Hydrogen
MQ214	Methane, Natural gas
MQ216	Natural gas, Coal gas
MQ303A	Alcohol, Ethanol, smoke
MQ306A	LPG, butane gas
MQ307A	Carbon Monoxide
MQ309A	Carbon Monoxide, flammable gasses
MG811	Carbon Dioxide (CO ₂)
AQ-104	Air quality

Tabelle 7 List of MQ Gas Sensors [65]

Bei der Messung von Feinstaub werden Partikel in der Luft erfasst. Feinstaub wird in zwei normierten Grössen, PM2.5 und PM10 gemessen, wobei die Bezeichnung die Grösse der Partikel, 2.5 µm resp. 10 µm beschreibt.

Bei Laser-Scattering Messmethode besteht das Messgerät aus einer Laserdiode und einer Photodiode. Die von den Partikeln reflektierten Laserstrahlen werden von der Photodiode gemessen. Mit einem Ventilator wird die Luft zirkuliert, was zu einer gleichmässigen Verteilung der Partikel führt. Das Signal wird typischerweise von einer eingebauten Signalverarbeitungseinheit umgerechnet und in der Einheit µg/m³ bereitgestellt. Abbildung 43 zeigt einen Feinstaubsensor für die Erfassung von PM2.5 und PM10. Die relative Messabweichung beträgt ±15%. Als Schnittstelle wird UART verwendet, die Sampling Rate beträgt 1 Hz. Für den Einlass ist ein Stecksystem für einen Schlauch vorgesehen [66]. Der Preis liegt bei gut 30 CHF.

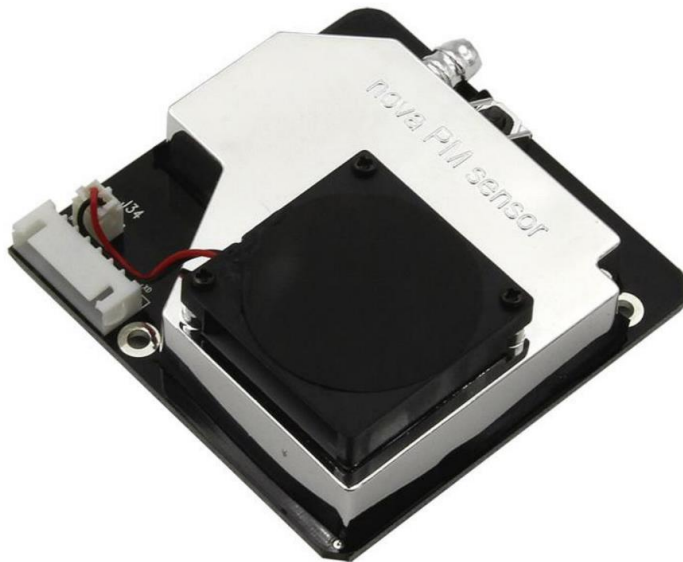


Abbildung 43 SDS011 PM Sensor [66]

2.3.4.5 Wasserqualität

Die Wasserqualität kann auf verschiedenen Ebenen gemessen werden. Der Anteil von organischem Kohlenstoff im Wasser, TOC (Total Organic Carbon), wird als allgemeines Mass zur Beschreibung der Wasserqualität eingesetzt. TOC Sensoren erkennen schädliche Verunreinigungen wie Benzole, Pestizide, Lösungsmittel oder Antibiotika. Für die Bestimmung des TOC Levels wird das bei der Oxidation des organischen Kohlenstoffs entstandene CO₂ gemessen [67]. Durch den komplexen Messvorgang sind die Sensoren sehr teuer.

Gelöste Feststoffe im Wasser sind als TDS (Total Dissolved Solids) messbar. Das Messprinzip ist ähnlich wie bei resistiven Bodenfeuchtigkeitssensoren. Zwischen zwei Leiter, welche sich im Wasser befinden, wird eine Spannung angelegt. Je mehr gelöste Feststoffe sich im Wasser befinden, desto grösser wird der Strom, der fließt [68]. Durch das Messprinzip wird der Einsatzbereich auf stehende Gewässer beschränkt, deren Temperatur unter 70 °C liegt. Die relative Messabweichung von preiswerten TDS Sensoren beträgt $\pm 10\%$ [69].

Ein weiteres Mass zur Bestimmung der Wasserqualität ist die Trübung. Trübungssensoren bestehen aus einer Lichtquelle und einem optischen Sensor. Dazwischen befindet sich die zu messende Flüssigkeit. Abbildung 44 zeigt ein Trübungssensor mit einer Datenverarbeitungseinheit. Der Preis des abgebildeten Sensormoduls liegt bei knapp 25 CHF.



Abbildung 44 Turbidity Sensor [70]

PH Sensoren messen den Säuregrad von Flüssigkeiten, ein weiterer Indikator der Wasserqualität. Zur Bestimmung des Säuregrades wird die Wasserstoffionenkonzentration gemessen und anschliessend in den pH-Wert umgerechnet. Der Aufbau besteht aus einer elektrochemischen Zelle. Eine Hälfte der Zelle besteht aus einer Messelektrode, deren Potenzial direkt proportional zum pH-Wert der Flüssigkeit ist. Auf der anderen Seite befindet sich eine Referenzelektrode mit unabhängigem Potential. Über den Vergleich des Potentials lässt sich der Säuregrad bestimmen.

Abbildung 45 zeigt eine Messsonde und eine Auswertungseinheit. Der Sensor muss vor der Messung mit einer Referenzflüssigkeit kalibriert werden. Bei länger andauerndem Gebrauch entsteht ein kleiner Drift. Die Genauigkeit liegt bei ± 0.2 PH. Die Messspanne deckt den gesamten pH Bereich (0 – 14) ab, wobei die Messtemperatur höchstens 60 °C betragen darf. Der abgebildete Sensor wird analog ausgelesen. Der Preis liegt unter 20 CHF [71].



Abbildung 45 PH Sensor Kit [71]

2.3.4.6 Lichtsensoren

Für die Messung der Intensität von Licht existieren verschiedene Sensoren für verschiedene Arten von Strahlung. LDR (light dependent resistance), auch Photowiderstand genannt, ändern ihren Widerstand bei Lichteinfall. Da es sich um simple passive Bauteile handelt sind die Anschaffungskosten sehr gering.

Für die Messung spezifischer Wellenlängen, beispielsweise UV oder Infrarot, existieren spezielle Sensoren mit entsprechenden Empfindlichkeiten. Sonnenlicht besteht aus sichtbarer und unsichtbarer Strahlung. Die Wellenlängen der sichtbaren Strahlung liegen zwischen 400 und 700 nm.

Abbildung 46 zeigt einen Sonnenlichtsensor mit integriertem Näherungssensor. Er kann Wellenlängen zwischen 280 und 950 nm erkennen und wird über I2C angesteuert. Es sind drei Messwerte (Sichtbar, UV, IR) verfügbar. Die Anschaffungskosten des abgebildeten Sensors betragen rund 10 CHF [73].

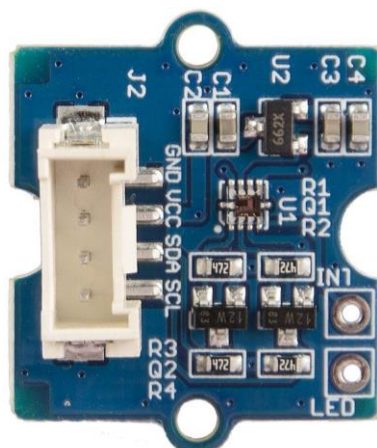


Abbildung 46 Sunlight Sensor [73]

2.3.4.7 GPS

Mit Global Positioning System (GPS) Modulen wird die aktuelle Position des Geräts erfasst. Sie empfangen von Satelliten ausgesendete Signale, decodieren diese und stellen sie in lesbarem Format zur Verfügung. Zusätzlich zu den Koordinaten beinhalten die Datensätze eine sehr genaue Zeitangabe. Die Geschwindigkeit, mit welcher sich das Gerät bewegt, wird je nach Modul auch als Zusatzinformation bereitgestellt. Abbildung 47 zeigt ein GPS Modul mit Antenne. Der Anschluss erfolgt via UART.



Abbildung 47 Grove GPS Modul [74]

2.3.4.8 Kameras

Kameras gehören zu der Kategorie der optischen Sensoren. Es gibt eine Vielzahl von Kameras für den Einsatz im IoT Bereich, welche sich hauptsächlich durch Auflösung und Brennweite unterscheiden. Ein preiswertes Kameramodul, welches durch die Kompatibilität mit Raspberry Pi SBC eine hohe Popularität erlangte, ist das Raspberry Pi Camera Module v2, welches in Abbildung 48 ersichtlich ist. Es verfügt über eine maximale Auflösung von 3280 x 2464 Pixel und kann mittels Flachbandkabel mit dem SBC verbunden werden. Der Fokus kann mit Spezialwerkzeug eingestellt werden. Die minimale Distanz beträgt 80 mm. Die Anschaffungskosten liegen unter 30 CHF [75].

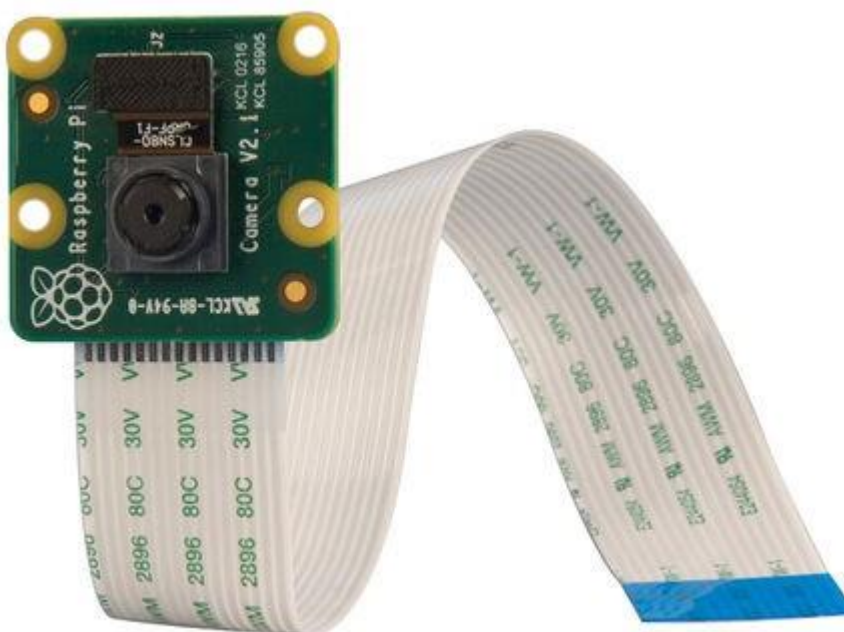


Abbildung 48 Raspberry Pi Camera Module v2 [75]

2.3.4.9 Mikrofone

Für die Erfassung von Tönen werden Mikrofone eingesetzt. Sie verfügen über verschiedene Eigenschaften wie Frequenzbereich, Sampling Rate und die Form des Erfassungsbereichs.

2.3.4.10 Digitale Broadcast Nachrichten

Smartphones senden in definierten Intervallen BLE Advertising Packets aus, um von Geräten erkannt zu werden. Auch Wi-Fi Interfaces von Smartphones senden regelmässig kleine Datenpakete, welche die MAC Adresse beinhalten, als Broadcast Nachricht aus. Diese Datenpakete können verwendet werden, um eingeschaltete Smartphones in der Umgebung zu zählen.

Aus Datenschutzgründen müssen die MAC Adressen anonymisiert und so wenig wie möglich gespeichert werden. MAC Adressen werden von neueren Smartphones zwar zufällig generiert, jedoch ist ein personenbezogenes Tracking über kürzere Zeiten trotzdem möglich.

Wie Smartphones senden auch Flugzeuge regelmässig Signale aus. Die Signale beinhalten Informationen über die Position, Höhe, Flugrichtung und Geschwindigkeit. Mit ADS-B Empfänger können diese Signale empfangen werden. Der Kostenpunkt eines preiswerten Empfängers liegt bei knapp 40 CHF [76].

2.3.4.11 Wahl eines Sensors

Oft gibt es mehrere Möglichkeiten, eine Grösse zu messen. Um die bestmögliche Wahl zu treffen, sollten die Punkte in Tabelle 8 beachtet werden.

Kriterium	Zu beachten
Einsatzbereich	<ul style="list-style-type: none"> • Temperaturbedingungen • Schutz gegen Feuchtigkeit und Berührung • Staubschutz • Zulässiges Funkspektrum
Auflösung	<ul style="list-style-type: none"> • Messabweichung im akzeptablen Bereich?
Kosten	<ul style="list-style-type: none"> • Sensor • Zubehör wie Kabel, Schutzhülle
Schnittstelle	<ul style="list-style-type: none"> • Besteht Anschlussmöglichkeit auf MCU? • Bei Analog: Spannungsbereich
Datenrate	<ul style="list-style-type: none"> • Wird sie von der MCU unterstützt? • Anpassbar?
Dokumentation	<ul style="list-style-type: none"> • Datenblatt verfügbar? • Messprinzip klar definiert? • Umrechnung in physikalische Einheiten

Tabelle 8 Kriterien bei der Auswahl eines Sensors [77]

3 Analyse Ist-Zustand

Dieses Kapitel befasst sich mit der Analyse des Systems. Sie umschliesst die Erfassung des Ist-Zustands, die angestrebten Ziele und dafür nötige Optimierungen. Der aktuelle Zustand beschreibt den Zustand bei Projektbeginn.

Im Rahmen des SNF Forschungsprojekts Mitwelten wurden für zwei Systeme Proof of Concepts entwickelt. Das erste System ist ein aus mehreren Kameras und einem Access Point bestehendes Bilderfassungssystem. Als zweites Proof of Concept wurden Low-Power Sensorsysteme mit LoRa Connectivity entwickelt. Das Backend besteht aus zwei VM's.

3.1 Kamerasystem

Das Kamerasystem besteht aus mehreren über PoE an einem AP Gateway angeschlossenen Kameras. Abbildung 49 zeigt die Bestandteile. Im AP Gateway befindet sich eine Harddisk, in welcher die Aufnahmen gespeichert werden. Über eine Mobilfunkverbindung kommuniziert das AP Gateway mit dem Backend.

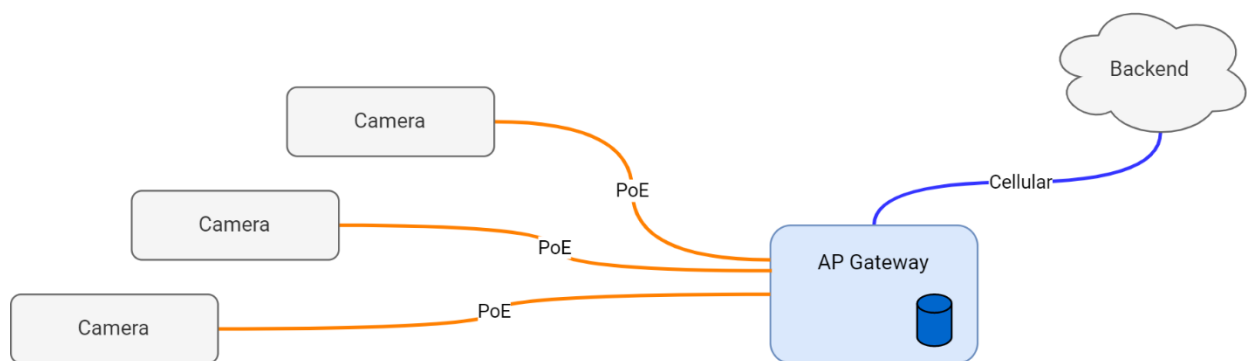


Abbildung 49 Übersicht Kamerasystem

Abbildung 50 zeigt eine Kamera (links) und ein AP Gateway, der in einem Fass verbaut ist (rechts).



Abbildung 50 PoE Cam (l) and AP Gateway (r) [78]

3.1.1 PoE Kameras

Eine Kamera besteht aus einem Raspberry Pi, einem PoE Hat und einem Camera Module v2. Die Bauteile sind in Abbildung 51 zu erkennen. Die Module sind in einer AP Abzweigdose montiert, um sie vor äusseren Einflüssen zu schützen. Für die Montage wurde ein Stecksystem aus

Kunststoffplatten entwickelt. Die Linse wird durch eine Plexiglasplatte geschützt. Die schwarze Membrane, welche auf der rechten Seite erkennbar ist, dient zum Ablass der Feuchtigkeit im inneren des Gehäuses. Für die Abzweigdose wurde eine Halterung entworfen, mit welcher sie an ein M20 Alu Installationsrohr geklemmt wird. Die Montage der Kamera wird in Abbildung 50 gezeigt.



Abbildung 51 PoE Camera [78]

Das Kameramodul wird mit einem Spezialwerkzeug fokussiert. Diese Einstellung erfordert einen Ausbau von allen Modulen aus der Abzweigdose und ist im Nachhinein nur mit grossem Aufwand anzupassen. Erste Tests haben ergeben, dass Wasser entlang des Kabels in das Gehäuse dringen kann. Durch die engen Platzverhältnisse im Gehäuse kann es bei Belastung des Kabels zu einem Bruch der SD Karte führen. Die Halterung für die Abzweigdose wurde durch Umwelteinflüsse wie Sonneneinstrahlung und Temperatur stark verbogen.

Auf dem Raspberry Pi läuft Raspberry Pi OS Lite. Für die Aufnahmen wird mjpg-streamer [79] eingesetzt. Die Software wird beim Startup automatisch gestartet und stellt über HTTP Endpunkte statische Bilder und einen Video stream zur Verfügung. Die Auflösung der Bilder wird im Service Skript definiert. Die Identifizierung der Kameras geschieht über den Hostnamen, welcher die ID beinhaltet. In den ersten Tests entstanden keine Probleme mit der Software der Kamera.

3.1.2 AP Gateway

Das Herzstück des AP Gateways besteht aus einem Raspberry Pi. Neben PoE Access verfügt das AP Gateway auch über einen WiFi Access Point. Das Raspberry Pi befindet sich in einem Gehäuse mit eingebautem Ventilator, um die Wärme abzuführen. Im Blockdiagramm (Abbildung 52) sind alle Bestandteile ersichtlich. Adapter für die Energieversorgung sind nicht eingezeichnet.

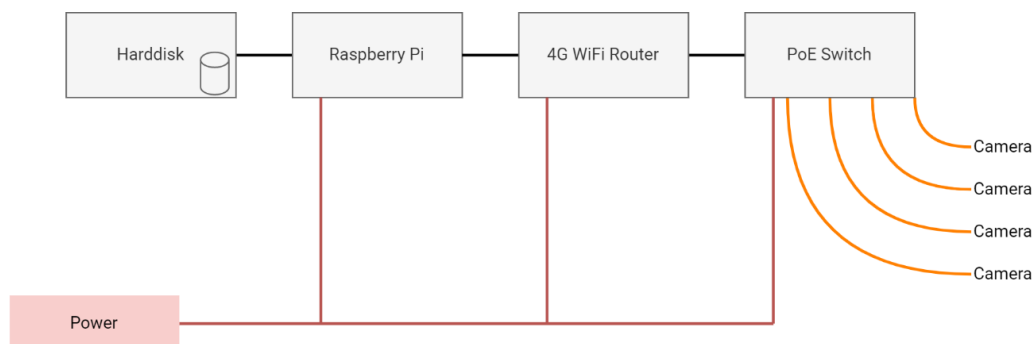


Abbildung 52 Blockdiagramm AP Gateway

Die Bauteile des AP Gateways sind durch ein Fass von äusseren Einflüssen geschützt. Für die Befestigung wurde ein steckbares System aus Holzplatten entworfen. Abbildung 53 zeigt links das Innenleben und rechts das geschlossene Fass. Die entwickelte Standvorrichtung besteht aus zugeschnittenen Schalungsplatten. Durch den Deckel des Fasses werden die Kabel für die Kameras und Stromversorgung nach aussen geführt. Wie bei den Kameras befindet sich eine Membrane zur Ausscheidung von Feuchtigkeit an der Aussenwand des Fasses.



Abbildung 53 AP Gateway offen und geschlossen [80] [81]

Der AP Gateway hat mehrere Aufgaben. Die Hauptaufgabe besteht darin, das System mit Internetzugang zu versorgen. Mit dem 4G Router wird die Verbindung über Ethernet oder WLAN bereitgestellt. Weiter stellt er über PoE sowohl Energie als auch Netzwerkverbindungen für die Kameras bereit. Über den verbauten 5-Port PoE Switch können maximal vier Kameras angeschlossen werden. Das PoE Budget beträgt 65 Watt, womit bei gleichmässiger Verteilung maximal 16.25 Watt pro Verbraucher konsumiert werden dürfen.

Die Aufnahmen der Kameras werden auf der am Raspberry Pi angeschlossenen 2 TB Harddisk gespeichert. Auf den Kameras befindet sich ein HTTP Endpunkt, über welchen die Bilder aufgenommen und heruntergeladen werden können. Ein Python Skript [82] auf dem AP Gateway löst in einem konfigurierbaren Intervall Aufnahmen der Kameras aus und speichert das Bildmaterial in einer generierten Ordnerstruktur auf der Harddisk.

Über ein auf Python basierendes Web Gui [83] können die Bilder angesehen und heruntergeladen werden. Ein Screenshot ist in Abbildung 54 ersichtlich. Der Zugriff geschieht über einen verschlüsselten Tunnel. Für die Zugriffskontrolle wird Basic Auth eingesetzt.

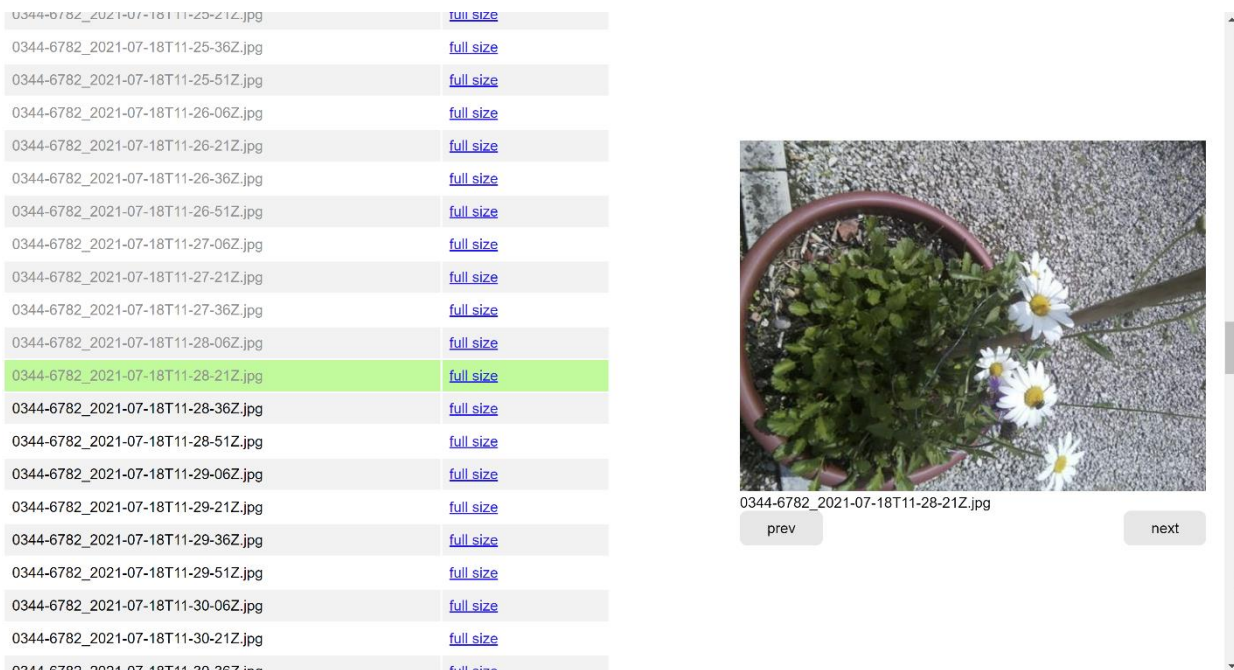


Abbildung 54 Screenshot ImagePreview Gui

Durch die lokale Speicherung der Aufnahmen kann das System offline funktionieren. Das Risiko von Datenverlust ist jedoch relativ hoch. Bei einem Hardwaredefekt oder Diebstahl sind die Daten verloren.

Im ersten Feldversuch hat das System grundsätzlich gut funktioniert. Komplikationen sind bei langen PoE Kabeln oder defekten Kameras aufgetreten. Dies hatte zur Folge, dass die Aufnahme Skript auf eine Antwort gewartet und somit andere Aufnahmen blockiert hat. Durch ein temporäres Entfernen der defekten Kameras wurde der Betrieb trotzdem aufrecht erhalten. Auf Seite der Hardware wurde eine grosse Wärmeentwicklung im Inneren des Fasses festgestellt. Der Grund war die ausgeschaltete Lüftung des Raspberry Pi Cases. Als zusätzliche Massnahme wurden die Fässer mit einer reflektierenden Folie eingekleidet, wie in Abbildung 55 ersichtlich. Der Schutz der Hardware stellte sich als sehr robust heraus. Auch ein umgekipptes Gateway hat weiter funktioniert. Es hat sich nur minimal Feuchtigkeit angesammelt.



Abbildung 55 AP Gateway eingekleidet [84]

3.1.3 Aufbau und Inbetriebnahme

Für den Aufbau der Hardware ist ein Lasercutter, ein 3D Drucker und eine Säge zur Fertigung des Standfusses für die Fässer erforderlich. Die Bauteile sind in einem Baumarkt erhältlich. Erfahrung im Umgang mit den Werkzeugen ist erforderlich.

Damit der AP Gateway eine Internetverbindung bekommt, muss ein Daten Abonnement abgeschlossen werden. Die softwareseitige Inbetriebnahme ist in einem GitHub Repository [78] dokumentiert und ist für Fachpersonen relativ simpel.

3.2 LoRa Sensoren

Als zweites Proof of Concept wurden solarbetriebene Low-Power Sensorsysteme mit LoRa Connectivity entwickelt. Die Entwicklung umfasst ein System zur Erfassung von Umweltdaten, wie Temperatur und Luftfeuchtigkeit, und ein PAX Counter, mit welchem Bluetooth Broadcast Signale der umliegenden Mobiltelefonen gezählt werden. Abbildung 56 zeigt die Bestandteile eines Nodes. Bei den PAX Counter sind keine zusätzlichen Sensoren angeschlossen.

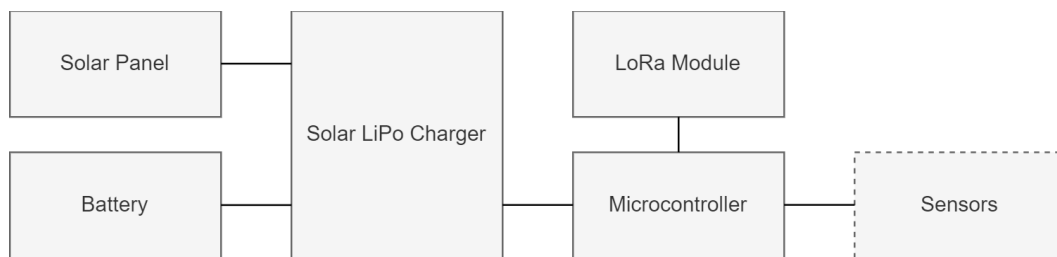


Abbildung 56 Blockdiagramm LoRa Nodes

Um die Elektronik von Umwelteinflüssen zu schützen, wurden AP Abzweigdosen eingesetzt. Abbildung 57 zeigt eine Dose mit eingebautem Feather M4 Entwicklungsboard, LoRa Modul und LiPo Akku. Das Entwicklungsboard wurde auf eine Grove Shield gesteckt, um Sensoren einfach anzuschliessen.

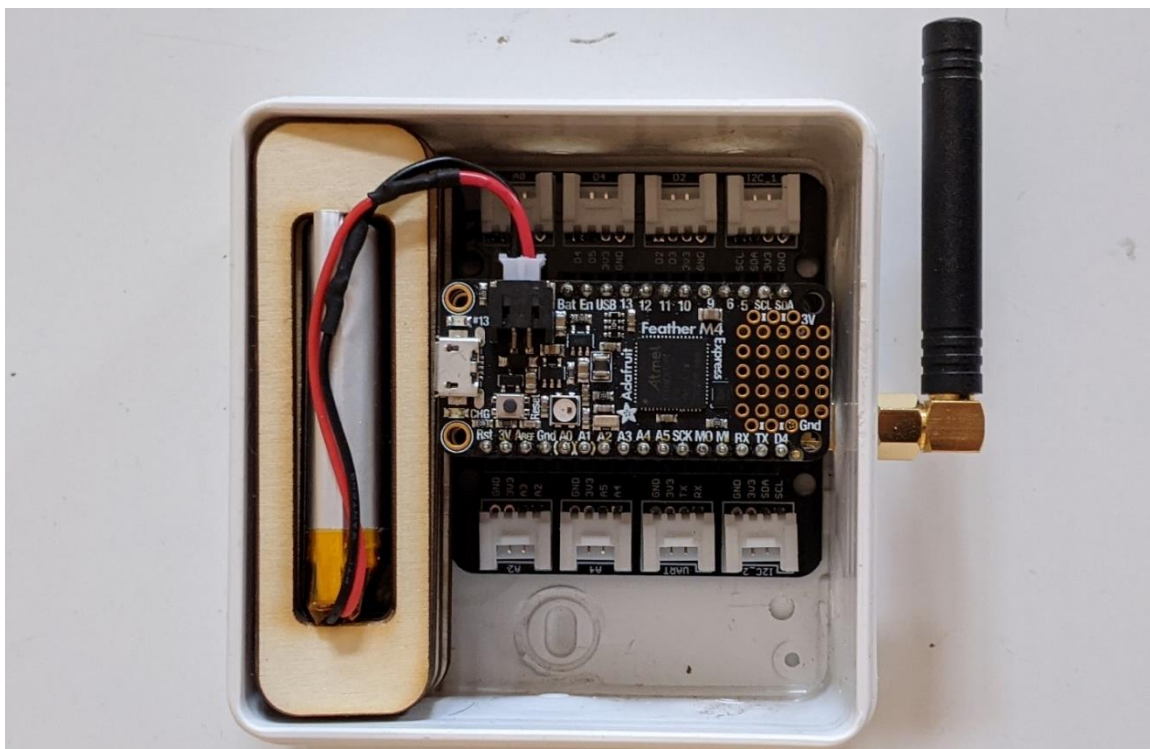


Abbildung 57 Case for LoRa Nodes [85]

3.2.1 Umweltsensoren

Die Sensoren dienen zur Erfassung physikalischer Messgrößen der Umwelt. Das Proof of Concept beinhaltet einen kombinierten Temperatur- und Luftfeuchtigkeitssensor und einen Kapazitiven Sensor zur Bestimmung der Bodenfeuchtigkeit. Abbildung 58 zeigt ein Sensor Node für Umweltmessungen. Im unteren Bereich ist der Bodenfeuchtigkeitssensor mit angeschlossenem Kabel ersichtlich. Der Temperatur- und Luftfeuchtigkeit Sensor befindet sich im aufgeschnittenen Pingpong Ball, der als Schutz dient.



Abbildung 58 LoRa Umweltsensor [86]

Die Nodes waren rund drei Monate im Feld. Der Akku konnte sich an sonnigen Tagen in kurzer Zeit auf die Maximalspannung aufladen. Bei mehreren aufeinanderfolgenden Tagen ohne direkte Sonneneinstrahlung haben sich einige Akkus komplett entleert und mussten manuell geladen werden.

Bei dem Feldversuch ist es zu defekten bei den Temperatur- und Luftfeuchtigkeitssensoren gekommen. Es ist möglich, dass Komponenten durch die hohe Luftfeuchtigkeit oder durch Wasser beschädigt wurden.

Die Messwerte für Temperatur und Luftfeuchtigkeit waren durch die Montage im Pingpongball sehr ungenau. Bei Sonneneinstrahlung erwärmt sich die Luft im Inneren des Balles und die Feuchtigkeit konnte sich anstauen. Zudem ist die Distanz zum Boden sehr gering, womit die Temperaturwerte nicht mit denen von standardisierten Messmethoden verglichen werden kann.

3.2.2 PAX Counter

Die PAX Counter basieren auf einem ESP32 Entwicklungsboard, da für die Erfassung der umliegenden Smartphones eine Bluetooth Schnittstelle erforderlich ist. In einem ersten Testlauf wurde die Open-Source Software ESP32-Paxcounter [87] eingesetzt, die auf PlatformIO basiert. ESP32-Paxcounter ist modular aufgebaut und verfügt neben dem Zählen von Smartphones über eine Vielzahl weiterer Integrationen wie Umweltsensoren. Es ist jedoch nicht möglich, mehrere

Messungen zu machen und anschliessend die Summe der erkannten Geräte zu übermitteln. Auch mit eingeschaltetem Deep Sleep Mode wurde relativ viel Energie verbraucht.

Um den Energieverbrauch zu minimieren wurde die Software des PAX Counter von Grund auf neu geschrieben. In einem periodischen Intervall wird ein BLE Scan ausgeführt. Aus den Adressen wird ein 32 Bit CRC berechnet und im EEPROM gespeichert. Beim nächsten Scan wird anhand des CRC verglichen, ob die Geräte schon im letzten Scan erfasst wurden und entsprechend zwischengespeichert. Bei Erreichen des Sendezeitpunktes wird die Summe der Geräte via TTN übermittelt und die zwischengespeicherten CRC's gelöscht.

Trotz der Optimierung des Energieverbrauchs entluden sich einige Akkus beim zweiten Testlauf komplett. Im Gegensatz zum ersten Testlauf, wo nur vereinzelt Mobiltelefone erkannt wurden, wurden mit der neuen Version wesentlich realistischere Werte erfasst. Eine Visualisierung von Messwerten einer gesamten Woche ist in Abbildung 59 ersichtlich. Es sind klare Unterschiede zwischen den Wochentagen erkennbar.

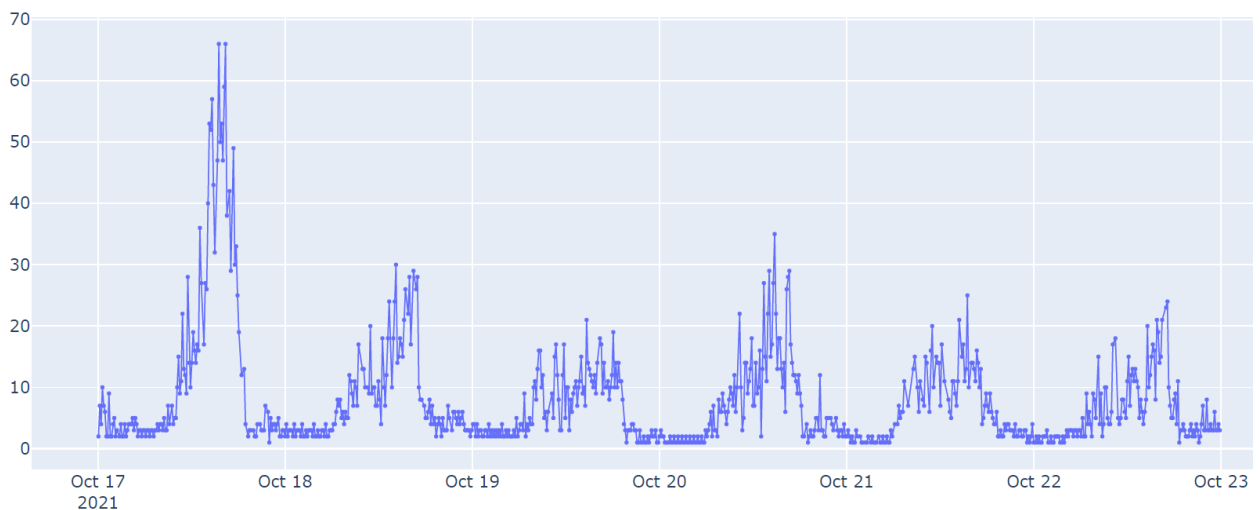


Abbildung 59 PAX Counter Measurements 7 days

3.2.3 Hardware

Nach knapp 5 Monaten Einsatzzeit wurden die Sensoren eingesammelt und auf Beschädigungen kontrolliert. Bei verschiedenen Nodes wurden korrodierte Stellen festgestellt. Abbildung 60 zeigt ein PAX Counter mit Schäden am Stecker des Solar Panels, am LiPo Charger und an den GPIO des ESP32 Entwicklungskits. Die Ursache liegt mit hoher Wahrscheinlichkeit an den Einführstellen der Kabel und Stecker in die Abzweigdose. Zudem kann Kondenswasser nicht abgeführt werden und staut sich an.

Durch die korrodierten Kontakte kann es zu Kurzschlüssen oder überbrückten Kontakten kommen, womit der Energieverbrauch erhöht wird.



Abbildung 60 Korrodierte Elektronik eines PAX Counter

3.2.4 Aufbau und Inbetriebnahme

Für die Fertigung des Innenausbaus in der Abzweigdose und der Halterung für das Solar Panel ist ein Lasercutter erforderlich. Ein 3D Drucker wird benötigt, um die Halterung der Abzweigdose zu fertigen. Die Kommunikation über das TTN erfordert ein kostenloses Konto. Wenn keine LoRa Abdeckung vorhanden ist, wird zusätzlich ein LoRa Gateway benötigt.

Um die Entwicklungsboards zu programmieren wird eine Arduino IDE benötigt. Der Vorgang ist durch gute Dokumentationen auch von interessierten Laien durchführbar.

3.3 Backend

Das Backend läuft auf einer bei SwitchEngines gehosteten VM. Zum aktuellen Zeitpunkt dient es nur zur Erfassung von Messwerten der LoRa Nodes. Zur persistenten Speicherung wird eine PostgreSQL Datenbank mit PostGIS Erweiterung verwendet. Die Daten werden via MQTT von den TTN Applications bereitgestellt und mittels Worker in die Datenbank geschrieben. Die Datenpakete von TTN beinhalten eine proprietäre Identifikationsnummer (end-device identifier, EUI) der Source Nodes. Um die IDs der Geräte zu vereinheitlichen, beinhaltet die Datenbank eine Mapping-Table mit Übersetzungen der EUI in eine interne ID. Das Mapping wird durch den MQTT2PostgreSQL Worker durchgeführt.

Ein Client kann über eine mit PostgREST implementierte Rest Schnittstelle, die hinter einem Nginx Reverse Proxy liegt, auf die Daten zugreifen. Geoserver stellt den Zugriff auf räumliche Daten bereit. Eine Übersicht ist in Abbildung 61 ersichtlich.

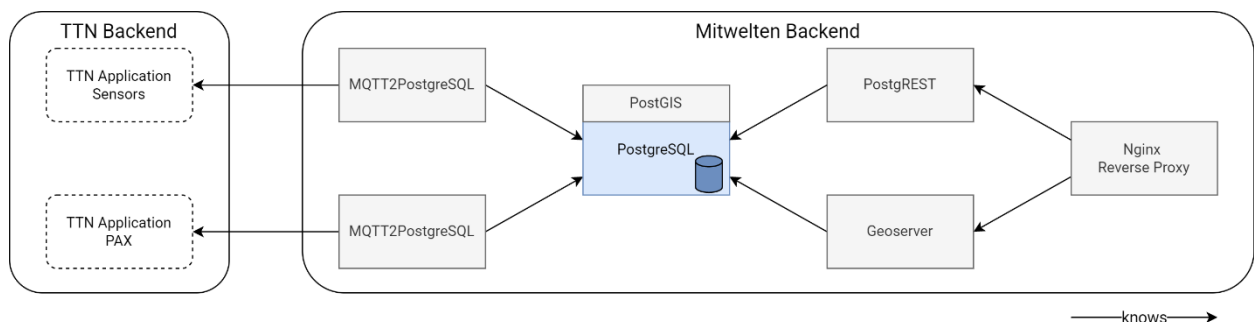


Abbildung 61 Overview Mitwelten Backend

Die PostGIS Erweiterung ist für räumliche Berechnungen der erfassten Messwerte vorgesehen. Im aktuellen Stand werden keine positionsabhängige Berechnungen mit PostGIS durchgeführt.

3.3.1 TTN Monitoring VM

Für die Überwachung der LoRa Sensoren wird eine weitere VM eingesetzt, welche auf AWS läuft. Auf ihr läuft eine Docker Instanz mit einem TIG (Telegraf, InfluxDB, Grafana) Stack in Containern. Eine Übersicht befindet sich in Abbildung 62. Die über MQTT bereitgestellten Messwerte werden durch Telegraf in die InfluxDB Zeitreihendatenbank geschrieben. Grafana greift auf die Datenbank zu und stellt die Messwerte und Metadaten der Übertragung in Form von Dashboards bereit.

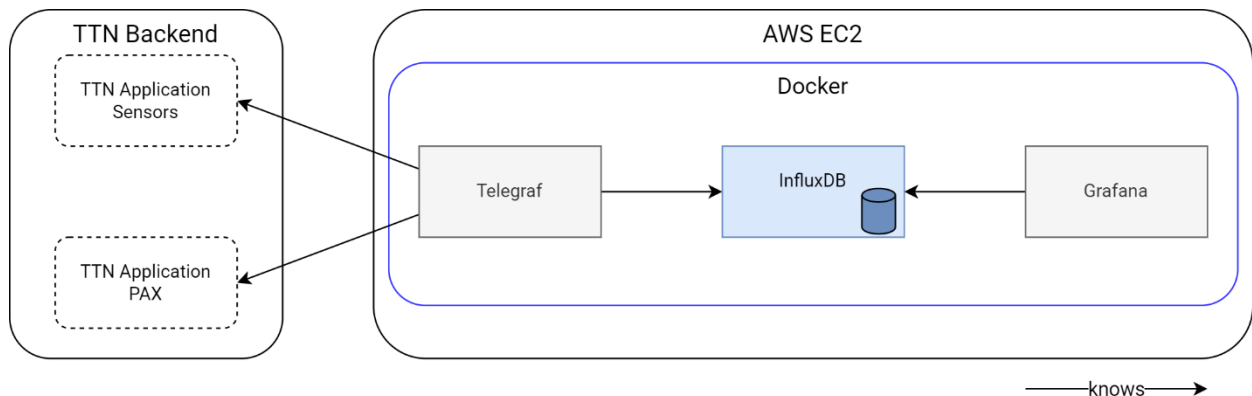


Abbildung 62 TTN Monitoring VM Overview

Der Zugriff auf die Grafana Dashboards geschieht durch einen Yaler Tunnel [88] mit TLS Verschlüsselung.

3.3.2 Deployment

Um das Backend zu realisieren wird eine VM benötigt, was in den meisten Fällen mit Kosten verbunden ist. Da die Software komplett Open-Source ist, fallen dafür keine Kosten an. Das Installieren und konfigurieren der Applikationen erfordert Fachkenntnis im Umgang mit Linux und SQL.

Die TTN Monitoring VM wird durch den Einsatz von Docker Compose mit wenigen Befehlen aufgesetzt. Alle Applikationen laufen als Docker Container und werden automatisch heruntergeladen und gestartet. Die Zustände der Applikationen werden überwacht und bei Fehlern werden sie neu gestartet.

4 Optimierungsmöglichkeiten

Das Hauptziel besteht darin, auf Basis der bestehenden Systemteile ein robustes, von Laien einsetzbares System zu entwickeln, um Daten von ökologischer Relevanz zu erfassen. Folgend werden mögliche Optimierungen pro Subsystem dokumentiert.

4.1 Kamerasystem

Bei dem Kamerasystem liegen die wichtigsten Optimierungen bei der Hardware und der Robustheit der Software, um die Funktionalität zu gewährleisten.

4.1.1 Hardware PoE Kameras

Bei den PoE Kameras sind Optimierungen des Gehäuses unerlässlich, um die Elektronik vor Feuchtigkeit und mechanischen Einwirkungen zu schützen. Um den Schutz gegen Feuchtigkeit zu gewährleisten, eignen sich Kabelverschraubungen. Eine Kabelverschraubung, wie in Abbildung 63 zu sehen, presst beim Zusammenschrauben eine Gummidichtung an das Kabel und erreicht somit eine hohe Dichtheit. Zusätzlich werden dadurch mechanische Belastungen durch Zug am Kabel abgedämpft.



Abbildung 63 Kabelverschraubung [89]

Bei der Verwendung von Kabelverschraubungen muss das Gehäuse deutlich grösser gewählt werden. Die Einführung benötigt mehr Platz als die bisherigen Würfenippel und der Ethernet Stecker darf nicht direkt hinter der Verschraubung platziert werden.

Um Beschädigungen aufgrund mechanischer Einwirkungen vorzubeugen müssen alle Bauteile fest im inneren des Gehäuses fixiert werden. Somit können abgebrochene SD Karten beim Transport, wie es im ersten Testlauf vorgekommen ist, verhindert werden. Um die Fokussierung der Kameras möglichst benutzerfreundlich zu halten, ist die Montage auf einer aus dem Gehäuse herausnehmbaren Befestigungsplatte von Vorteil. Der Deckel soll nur mit entsprechendem Werkzeug entfernbar sein, um zufälliges Öffnen zu verhindern.

Die Befestigung des Gehäuses an dem Aluminiumrohr kann durch eine mechanisch stabilere Konstruktion ersetzt werden. Die bisherigen Halterungen aus PLA haben sich aufgrund der werkstoffbedingten Eigenschaften durch Sonneneinstrahlung plastisch verformt. Eine Konstruktion aus Metallwinkeln würde diese Verformungen ausschliessen. Die Winkel können mit Rohrschellen an dem Aluminiumrohr befestigt werden.

4.1.2 Unterstützung von USB Kameras

Die PoE Kamera setzt die Verwendung einer über die Kameraschnittstelle angeschlossene Raspberry Pi Kamera voraus. Als Optimierung kann die Unterstützung von generischen, über USB angeschlossenen Kameras entwickelt werden. Durch die grössere Auswahl an Kameras wird das Einsatzgebiet der PoE Kameras vergrössert.

4.1.3 Robustheit der Capture Applikation

Die Capture Applikation fordert in einem definierten Intervall Snapshots von den angeschlossenen Kameras an und speichert die Bilder auf einer angeschlossenen Harddisk. Falls eine Kamera nicht erreichbar ist, wartet die Applikation auf die entsprechende Kamera und ist blockiert. Um dies zu verhindern, kann ein Timeout definiert werden, nach welchem eine Abfrage abgebrochen wird, damit von den funktionierenden Kameras trotzdem regelmässig Bilder heruntergeladen werden.

4.1.4 Benutzerfreundliche Konfiguration

Um den AP Gateway zu konfigurieren, müssen die URL der angeschlossenen Kameras und Parameter wie Aufnahmeintervall, Aufnahme Start- und Endzeitpunkt und Credentials für den Zugriff auf die Kameras definiert werden. Im bisherigen System geschieht diese Konfiguration durch manuelles Bearbeiten einer JSON Datei. Um diese Eingabe benutzerfreundlicher zu gestalten, kann eine Benutzeroberfläche entwickelt werden, deren Benutzung keinen Zugriff auf die Shell voraussetzt.

Die Benutzeroberfläche kann in Form eines Webservers entwickelt werden, welcher durch einen sicheren Tunnel oder aus dem lokalen Netzwerk erreichbar ist. Als zusätzliche Funktion können die Zustände der einzelnen Kameras visualisiert werden.

4.1.5 Erweiterte Konfigurationsmöglichkeiten

Die Konfigurationsmöglichkeiten der Capture Applikation beschränken sich im bestehenden System auf den Aufnahmeintervall, Startstunde und Endstunde der Aufnahmezeit, den Speicherort der Aufnahmen und Credentials für die Kameras. Die Parameter werden auf alle Kameras angewendet.

Als Optimierung kann eine Konfiguration pro Kamera ermöglicht werden, um sie unabhängig voneinander zu betreiben. Mit kameraspezifischen Konfigurationen besteht auch die Möglichkeit, Basic Auth Anmeldedaten pro Gerät zu definieren.

4.1.6 Automatisches Erkennen von angeschlossenen Kameras

Um die Konfiguration eines AP Gateways zu erleichtern kann ein Mechanismus entwickelt werden, mit welchem die Kameras automatisch erkannt und registriert werden. Für das Hinzufügen einer Kamera muss der AP Gateway die ID und eine URL, wo die Bilder bereitgestellt werden, kennen.

4.1.6.1 Erkennen durch Port Scanning

Ein einfaches Prinzip für das Erkennen von angeschlossenen Kameras ist das periodische Scannen eines Ports auf den Netzwerkschnittstellen. Auf den Kameras würde auf einem vordefinierten Port ein zu erkennendes Service laufen, der zur Identifikation dient. Somit wären die Abhängigkeiten zwischen dem AP Gateway und den Kameras minimal. Weder der Hostname noch die IP von den Kameras oder von dem AP Gateway muss fest definiert werden. Der Nachteil dieses Prinzips liegt jedoch darin, dass das Scannen von gewissen Netzwerken untersagt ist und als Angriff auf die Infrastruktur klassifiziert wird. Manche Netzwerke verfügen über Mechanismen, die das Scannen unterbinden, was das Erkennen von Kameras unmöglich machen würde. Bei dem bestehenden Kamerasystem wird das Netzwerk vom Eigentümer betrieben, was ihm auch die Freiheit des Scannen erlaubt, solange es alleinstehend betrieben wird.

4.1.6.2 Erkennen durch Anmeldeserver

Eine weitere Möglichkeit der Erkennung angeschlossener Kameras besteht darin, dass sich die Kamera bei dem AP Gateway anmeldet. Das setzt voraus, dass die Kamera weiss, über welchen Hostnamen oder über welche IP sie den AP Gateway erreichen kann. Wenn die IP für die AP

Gateways bei allen Standorten statisch definiert wird, kann diese für alle Kameras als Anmeldeendpunkt definiert werden.

Auf den AP Gateway muss ein REST Server laufen, der Anmeldeanfragen der Kameras entgegennimmt. Die Anmeldeanfragen der Kameras beinhalten die ID als Information. Wenn eine Anfrage einer noch nicht bekannten Kamera eintrifft, wird sie im System eingetragen.

4.1.7 Deployment

Das Aufsetzen der Kameras besteht aus mehreren Schritten, die nacheinander ausgeführt werden. Als Optimierung können die Schritte von einem Shell Skript ausgeführt werden, was den Setup Vorgang auf eine Eingabe minimieren würde. Der Hostname, welcher die ID der Kamera beinhaltet, kann mit dem Raspberry Pi Imager [90] direkt beim Schreiben des OS auf die SD Karte gesetzt werden.

Das Setup des AP Gateway kann durch die Verwendung eines Shell Skripts auch vereinfacht werden.

4.2 Sensorsysteme

4.2.1 Hardware

Die grössten Komplikationen entstanden durch das Eindringen von Wasser in das Gehäuse. Die verwendete IP65 Abzweigdose von ABB schützt vor Strahlwasser aus allen Winkeln. Die Einführungen der Kabel stellen die grösste Schwachstelle dar. Bei Bewegungen der Kabel oder Antennen dichten die Würgenippel nicht mehr komplett ab. Mit Kabelverschraubungen sind dichtere Verbindungen möglich. Sie dienen neben dem Schutz vor Berührung und Feuchtigkeit auch als mechanische Zugentlastung. Durch die Bauform wird mehr Platz benötigt und die Abzweigdose muss grösser gewählt werden.

4.2.2 Energiemanagement

Die Testläufe haben gezeigt, dass das eingesetzte 6V 1W Solar Panel mit dem 2000 mAh LiPo Akku nur bei optimalen Wetterbedingungen funktioniert. An sonnigen Tagen wurden die Akkus bereits am Vormittag vollgeladen, jedoch entluden sie sich innerhalb von vier Tagen ohne Sonneneinstrahlung komplett. Mit einem grösseren Solar Panel würden die Akkus auch an bewölkten Tagen bis zu einem bestimmten Level geladen werden. Bei Verwendung eines grösseren Akkus kann die Entladedauer hinausgezögert werden, eine komplette Entladung wäre aber trotzdem möglich.

Um den Energieverbrauch der Entwicklungskits zu verringern, können die an der direkten Stromquelle angeschlossenen Sensoren während des Deep Sleep Betriebs ausgeschaltet werden.

4.2.3 Sensorik

Die Umweltsensor Nodes verfügen im aktuellen System über fest definierte Sensormodelle und sind nicht erweiterbar. Um das Anwendungsspektrum zu erweitern, kann eine Unterstützung mehrerer Sensormodelle entwickelt werden.

4.2.3.1 Temperaturmessung

Die Temperatur der Luft wird standardmässig in einer Höhe von zwei Metern ab Boden und im Schatten gemessen. Der Boden erwärmt sich durch die Sonneneinstrahlung und gibt die Wärme dann an die Luft ab. Wenn der Abstand zum Boden geringer gewählt wird, sind tendenziell höhere Messwerte zu erwarten [91]. Mit einem Wasserdichten Sensor für Temperatur- und Luftfeuchtigkeit ohne zusätzlichem Schutz, der in der normierten Höhe platziert wird, kann die Qualität der Messwerte erhöht werden. Ein wetterfester Temperatur- und Luftfeuchtigkeitssensor von DF Robot ist in Abbildung 64 ersichtlich. Er verfügt über eine Messgenauigkeit von ± 0.2 °C [92].



Abbildung 64 SHT31 Weatherproof Temperature & Humidity Sensor [92]

Zusätzlich zu der Lufttemperatur ist die Bodentemperatur interessant für die Ökologischen Zusammenhänge. Als zusätzlicher Sensor kann ein wasserdichter IC-Temperatursensor in einer definierten Tiefe in den Boden gesteckt werden.

4.2.3.2 Bodenfeuchtigkeit

Beim ersten Testlauf wurde die Bodenfeuchtigkeit der obersten fünf Zentimetern gemessen. Gemäss den zuständigen Fachpersonen in den Merian Gärten sagt dieser Wert aber nicht viel aus, da die oberste Schicht sehr schnell austrockne. Messwerte von tieferen Erdschichten wären wertvollere Daten. Die Sensoren müssen kapazitiv und komplett wasserdicht sein, um Korrosionen vorzubeugen. Für verschiedene Messnetze in der Schweiz wird die Feuchtigkeit jeweils in einer Tiefe von 20 cm und 35 cm gemessen [93].

4.2.4 PAX Counter

Die aktuelle Version der PAX Counter führt alle 30 Sekunden einen Scan mit einer Dauer von drei Sekunden durch. Alle Geräte im Umkreis von ca. 10 Meter werden gezählt. Die Wahl des Intervalls ist abhängig von der verfügbaren Energie. Die qualitativ hochwertigsten Messwerte werden bei kontinuierlichem Scannen erreicht. Wenn eine Person mit einer Geschwindigkeit von 4 km/h einen PAX Counter passiert, kann sie während knapp 20 Sekunden erfasst werden. Mit einer doppelten Abtastrate gemäss Nyquist-Shannon darf der Abstand zwischen den Erfassungen maximal 10 Sekunden betragen.

4.2.5 Deployment

Für das Deployment der Nodes wird eine installierte Arduino IDE vorausgesetzt. Die entsprechenden Boards müssen hinzugefügt werden und die Installation einer Vielzahl von Libraries ist erforderlich. Der Source Code muss für jedes Gerät abgeändert und kompiliert werden, da die Credentials hartkodiert sind.

Durch eine Trennung von Code und Konfiguration muss der Code nur einmal kompiliert werden und kann für alle Nodes verwendet werden. Die Konfiguration kann im EEPROM oder im Flash Speicher der Boards liegen und dynamisch angepasst werden.

4.2.5.1 Feather M4 Boards

Eine weitere Optimierung besteht darin, den bereits kompilierten Code bereitzustellen und somit keine installierte Arduino IDE vorauszusetzen. Die ARM Cortex M4- basierten Feather M4 Boards verfügen über einen UF2 Bootloader [94], welcher das Board vom PC als USB Speichergerät erkennen lässt. Mit Drag and Drop kann ein neues Programm auf den Controller geladen werden. Das kompilierte Programm muss vor dem Hochladen vom binären in das UF2 Format konvertiert werden.

Für die Konfigurierung der Nodes kann mit der TinyUSB Library [95] von Adafruit eine Konfigurationsdatei über USB auf das Board geladen werden, welche beim Setup ausgelesen wird.

4.2.5.2 ESP32 Boards

Die PAX Counter basieren auf dem Espressif ESP32 SoC und verfügen in der Basisversion über keinen UF2 Bootloader. Der Upload der kompilierten Programme wird von der Arduino IDE durch das Python Tool Esptool [96] durchgeführt. Sofern eine Python Installation auf dem PC vorhanden ist, kann der Upload einer bereits kompilierten Applikation ohne Arduino IDE durchgeführt werden. Bei Verwendung des Modells ESP32-S2 für die PAX Counter wird der UF2 Bootloader unterstützt und das Programm kann mittels Drag and Drop hochgeladen werden.

Für die Konfigurierung der PAX Counter wird ein direkter Zugriff auf den EEPROM Speicher nicht unterstützt. Da die ESP32 Boards über eine WiFi Schnittstelle verfügen, kann die Konfiguration über einen lokalen Webserver geschehen.

4.2.6 Connectivity

Die bisherigen Connectivity Optionen der LoRa Nodes beschränken sich auf LoRa mit ABP Aktivierung, was das System auf das Things Network beschränkt. Als Optimierung kann die Unterstützung der OTAA Aktivierung ermöglicht werden.

Die Adaptive Datenrate (ADR) passt den LoRa Spreading Factor, mit welchem das Node sendet, dynamisch an. Wenn das nächstgelegene LoRa Gateway, auf welches die Datenrate abgestimmt wurde, offline ist, kann dies zu Verlusten von Nachrichten führen. Als Optimierung kann die Option implementiert werden, ADR zu deaktivieren. Bei deaktiviertem ADR werden Nachrichten mit einem festgelegten SF gesendet.

Als Optimierung der Connectivity für die PAX Counter kann MQTT über WiFi als zusätzliche Connectivity Option implementiert werden. Ein WiFi Access Point in der Umgebung wird vorausgesetzt, um die Daten zu übertragen. Da WiFi wesentlich mehr Energie verbraucht als LoRa, ist es von Vorteil, den PAX Counter nicht im Akkubetrieb zu realisieren.

4.3 Backend

Das Setup des Backend ist mit einem sehr hohen Aufwand verbunden und ist für Laien nicht ohne fachliche Unterstützung möglich. Es setzt eine VM voraus, welche im Normalfall mit Kosten verbunden ist.

Als Optimierung kann eine einfache Version des Backends entwickelt werden, deren Aufgabe darin besteht, Sensordaten zu speichern und zu visualisieren. Es soll plattformunabhängig sein und auch auf einem lokalen Rechner oder Single Board Computer ausgeführt werden können. Um das Deployment möglichst simpel zu gestalten, kann Docker mit Docker Compose verwendet werden.

Für die Speicherung der Messwerte eignet sich InfluxDB, da es sich um Zeitreihendaten handelt und die eingesetzten Sensoren nicht fest definiert sind. Die über LoRa gesendeten Daten werden über MQTT von der entsprechenden Integration im Things Network Backend bereitgestellt. Mit Telegraf können die Messwerte aus den MQTT Nachrichten extrahiert und in die InfluxDB

geschrieben werden. Um die Messwerte zu visualisieren, eignet sich Grafana. Neben der hohen Benutzerfreundlichkeit bietet Grafana die Möglichkeit, Daten im Tabellenformat herunterzuladen.

Bei einem Einsatz von MQTT bei den PAX Countern kann im Backend ein MQTT Broker betrieben werden, der aus dem Internet über einen Tunnel oder einen freigeschalteten Port erreichbar ist.

Das Deployment des Backends kann mit einem Shell Skript einfach gestaltet werden, um die Komplexität zu verringern.

5 Implementierung

Nachfolgend werden die implementierten Optimierungen, Erweiterungen und Anpassungen der Systeme dokumentiert.

5.1 PoE Kamera

Für die PoE Kameras wurden Optimierungen von Software und Hardware implementiert.

5.1.1 Hardware

In einem Autodesk Inventor wurden die Bestandteile der PoE Kamera modelliert und möglichst platzsparend angeordnet, um die minimalen Abmessungen für ein Gehäuse zu ermitteln. Da der Stecker des Ethernet Kabels vollständig im Gehäuse sein soll, muss in diesem Bereich genügend Platz eingeplant werden. Die ermittelten minimalen Innenabmessungen eines Gehäuse betragen 130 mm auf 65 mm.

Anschliessend wurde nach einem passenden Gehäuse für den Schutz vor Wasser und Berührungen gesucht. Die Hauptkriterien lagen beim Preis und der mechanischen Festigkeit, die minimale IP Schutzklasse wurde bei IP65 angesetzt. Ein Kunststoffgehäuse des Herstellers RND Components entsprach den Anforderungen am besten. Die Aussenabmessungen betragen 160 x 80 x 55 mm und die IP Schutzklasse liegt bei IP65. Im Inneren des Gehäuse befinden sich Innengewinde für die Befestigung von Bauteilen. Der Preis beträgt 12.11 CHF bei der Bestellung eines Einzelstücks.

Für die mechanische Befestigung der Bauteile im Inneren des Gehäuse wurde eine Befestigungsplatte konstruiert, auf welche das Raspberry Pi und die Cam mit Schrauben montiert werden. Die Befestigungsplatte kann mittels 3D Druckverfahren in rund zwei Stunden gefertigt werden. Die Platte mit den montierten Komponenten wird mit vier Schrauben an den Innengewinden des Gehäuse befestigt. Die Einführung des Ethernet Kabels erfolgt durch eine Bohrung mit 20 mm Durchmesser, durch welche eine M20 Kabelverschraubung montiert wird. Auf der Unterseite des Gehäuse muss eine Öffnung für die Kamera erstellt werden. Abbildung 65 zeigt die Montage der Bestandteile als Explosionsansicht.

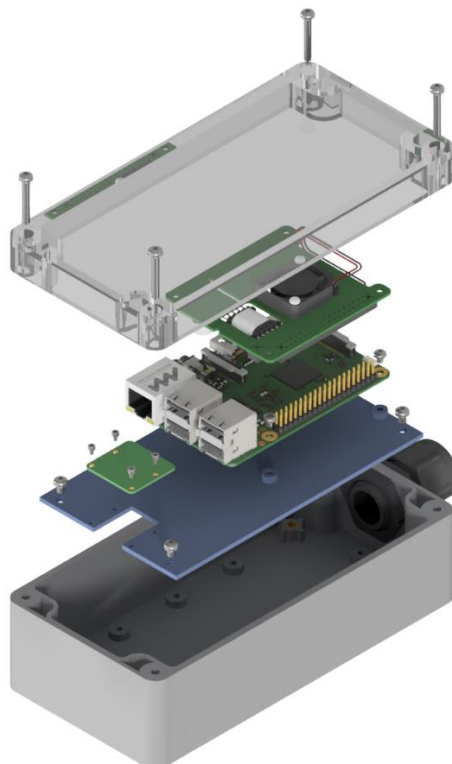


Abbildung 65 PoE Kamera Explosionsansicht

Abbildung 66 zeigt eine zusammengesetzte PoE Kamera mit geöffnetem Deckel. Die abgebildete Version des PoE Adapters verfügt über keine Durchführungsmöglichkeit für den Kameraanschluss. Durch das Entfernen einer Schraube des Lüfters kann das Kabel zwischen dem Raspberry Pi und dem PoE Adapter durchgeführt werden. Für die Kamera wurde auf der Unterseite des Gehäuses eine Öffnung gebohrt. Zum Schutz vor Feuchtigkeit wird sie mit einer PMMA (Plexiglas) Platte verschlossen. Als Alternative dazu kann ein transparenter Deckel eingesetzt und die Kamera gedreht werden.



Abbildung 66 PoE Kamera im offenen Gehäuse

Die Kamera wird mit Metallwinkeln an dem Aluminiumrohr befestigt. Eine mögliche Anordnung ist in Abbildung 67 ersichtlich. Bei der Wahl der Winkel ist darauf zu achten, dass der Werkstoff gegen Korrosion geschützt ist.

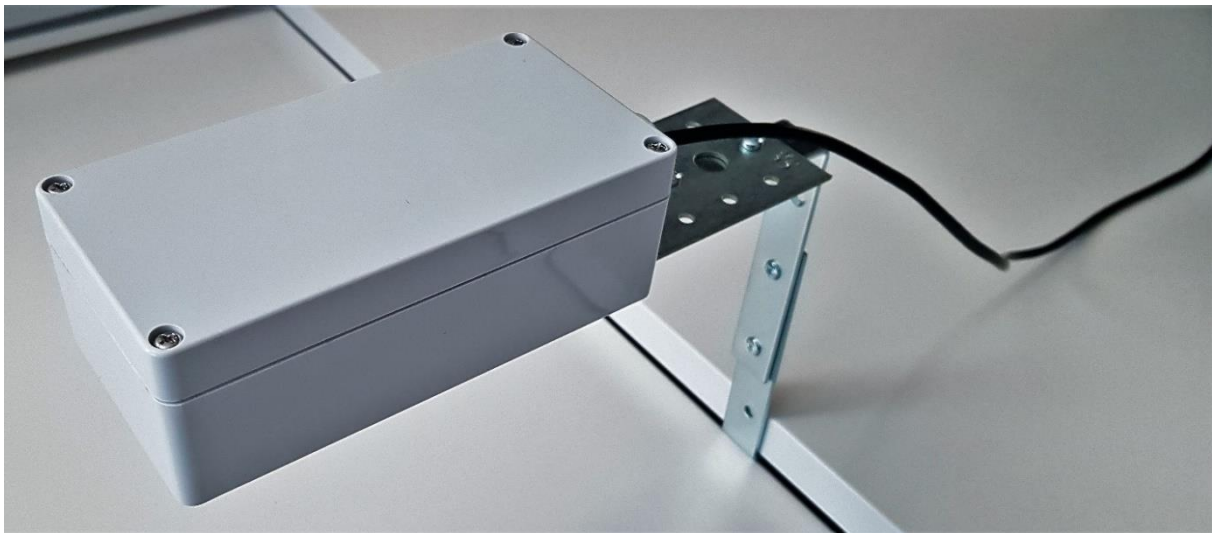


Abbildung 67 PoE Kamera Halterung mit Metallwinkeln

Für PoE Kameras mit einer externen USB Kamera wurde ein IP 65 Gehäuse des Herstellers BOX4U eingesetzt. Die Eigenschaften des Gehäuse sind im Kapitel *Version BOX4U Gehäuse* genauer beschrieben, da es auch für die Sensor Nodes eingesetzt wird. Für die Fixierung des Raspberry Pi wurde eine Montageplatte konstruiert und mittels 3D Druckverfahren gefertigt. Abbildung 68 zeigt die Kamera mit einem angeschlossenen Endoskop. Die Montageplatte wurde aus PLA gefertigt.

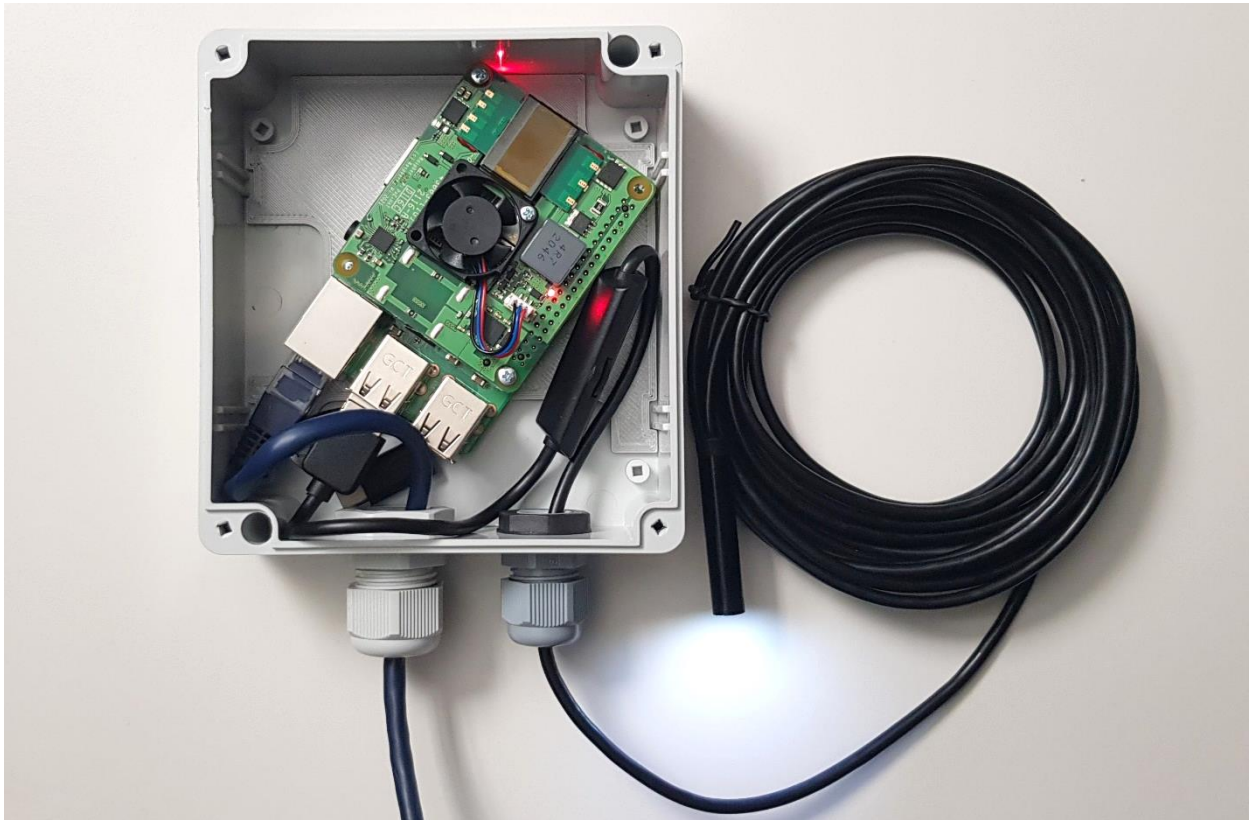


Abbildung 68 PoE Kamera mit über USB angeschlossener Endoskop Kamera

5.1.2 Unterstützung von USB Kameras

Neben der Unterstützung der Raspberry Pi Kameras wurde die Möglichkeit, eine generische USB Kamera einzusetzen, implementiert. Somit erweitert sich das Einsatzspektrum der PoE Kameras.

5.1.3 Deployment

Im Anhang und auf GitHub [97] befindet sich eine detaillierte Anleitung zum Deployment der PoE Kameras. Der Hostname, welcher die ID der Kamera beinhaltet, wird vor dem Schreiben des Betriebssystems auf die SD Karte definiert. Sobald ein Konsolenzugriff auf das Raspberry Pi besteht, können alle Applikationen mit der Ausführung eines Shell Skripts mit einem Befehl installiert werden. Das Skript beinhaltet neben dem Installieren der Programme die Aktivierung der Kamerarschnittstelle und das Erweitern des GPU Speichers auf 256 MB. Für den Einsatz von generischen, über USB angeschlossenen Kameras wurde ein separates Skript erstellt.

5.2 AP Gateway

Für den AP Gateway wurden ausschliesslich Optimierungen der Software implementiert. Die Hardware hat sich bei dem ersten Testlauf als zuverlässig und robust herausgestellt.

5.2.1 Konfigurationsverwaltung

Anstelle der JSON Datei, welche die Konfiguration für das gesamte System beinhaltet wurde eine Verwaltung der Konfiguration in einer SQLite Datenbank entwickelt. Sie beinhaltet pro Kamera Informationen über die Erreichbarkeit, Credentials, Abfrageintervall und Aufnahmezeitraum. Für den Zugriff auf die Datenbank wurde ein Adapter geschrieben, der von allen anderen Applikationen genutzt wird.

5.2.2 Capture

Die Capture Applikation wird zum Herunterladen von Aufnahmen von allen aktiven Kameras verwendet. Zur Erhöhung der Robustheit wurden für die Anfragen Verbindungs- und Lesetimeouts definiert. Bei Überschreitung der Zeitparameter wird die Anfrage abgebrochen.

Um einen individuellen Aufnahmeintervall zu gewährleisten, wird für alle vorhandenen Kameras periodisch überprüft, wann die letzte Aufnahme heruntergeladen wurde und in welchem Zeitraum aufgenommen wird. Entsprechend wird ein Capture ausgelöst oder nicht. Bei jeder Aufnahme wird der Zeitpunkt und der HTTP Status Code in der Datenbank aktualisiert. Der Intervall der periodischen Überprüfung legt den minimalen Aufnahmeintervall für das gesamte System fest.

5.2.3 Hinzufügen von Kameras

Kameras können manuell oder automatisch hinzugefügt werden. Der manuelle Vorgang ist im Kapitel *Betrieb* beschrieben.

Für die automatische Registrierung von neuen Kameras wurde ein HTTP Server entwickelt, der auf Port 8888 eingehende POST Requests auf der Route «/register» entgegennimmt. Die PoE Kameras senden regelmässig Anfragen an den Endpunkt, um sich anzumelden. Aus den Anfragen wird der Hostname und die ID der Kamera extrahiert und, falls nicht bereits vorhanden, in die Datenbank eingetragen. Die Registrierung setzt voraus, dass die Kameras Kenntnis über die Erreichbarkeit des AP Gateway haben. Um auf kameraspezifische Konfigurationen zu verzichten wird die IP des AP Gateway statisch konfiguriert.

Als weitere Option wurde eine Applikation entwickelt, welche die angeschlossenen Netzwerke periodisch auf einem festgelegten Port scannt. Auf den Kameras läuft auf dem entsprechenden Port ein HTTP Server, um erkannt zu werden. Nach dem Scan wird an alle erkannten Geräte eine Anfrage geschickt, worauf die Kameras mit ihrer ID antworten. Der Vorteil dieses Prinzips liegt darin, dass keine direkte Abhängigkeit zwischen AP Gateway und den Kameras vorliegt und somit beidseitig keine Kenntnis der Erreichbarkeit vorausgesetzt wird. Eine Gefahr entsteht, wenn der AP Gateway mit einem externen Netzwerk verbunden wird, in welchem das Scanning verboten ist. Um das Risiko auszuschliessen wird dieses Prinzip nicht eingesetzt.

5.2.4 Deployment

Eine detaillierte Beschreibung des Deployments ist im Anhang und auf GitHub [98] verfügbar. Alle Schritte sind in einem Shell Skript zusammengefasst. Für das Deployment wird somit nur ein Befehl erfordert, der das Skript herunterlädt und ausführt.

5.2.5 Betrieb

Für die Überwachung und Konfiguration des Systems wurde ein Webserver entwickelt, um die Benutzerfreundlichkeit zu erhöhen. Er ist erreichbar über den Port 8080, der Zugriff ist durch Basic Auth geschützt.

Auf der Startseite, ersichtlich in Abbildung 69, wird eine Liste der Kameras mit den wichtigsten Parametern, den Status Codes der letzten Requests und einem Button zum Ändern der Konfiguration angezeigt. Im oberen Bereich befinden sich Indikatoren der Servicezustände und des Speichers. Bei gestoppten Services oder knappem Speicherplatz werden sie rot hinterlegt. Unterhalb der Liste befinden sich Schaltflächen zum Hinzufügen von Kameras und zum Anpassen der globalen Konfiguration.

Capture Service running
Registrationsserver Service running
Disk Usage 34.73 % out of 28.91 GB

Cameras

ID	Capture Interval	Capture Start (utc)	Capture Stop (utc)	State	Last Capture (utc)	HTTP Code	
0000-0000	60 s	07:00	21:00	Enabled	28.01.2022 09:04:22	200	Edit
0000-0001	20 s	07:00	21:00	Enabled	28.01.2022 09:05:12	200	Edit
0000-0002	20 s	07:00	21:00	Enabled	28.01.2022 09:05:12	200	Edit
0000-0003	20 s	07:00	21:00	Enabled	28.01.2022 09:05:12	408	Edit

[+ Add Camera](#)
[Edit Global Settings](#)

Abbildung 69 AP Gateway User Interface Übersicht

Auf der Seite für die Einstellungen pro Kamera können URL, Zeitparameter und Credentials modifiziert werden. Über eine Checkbox kann die Kamera aktiviert oder deaktiviert werden. Abbildung 70 zeigt einen Screenshot der Eingabeform.

Edit Camera 0000-0000

Camera ID
0000-0000

Capture URL
http://cam-0000-0000.local:8080/?action=snapshot

Capture Interval [s]
60

Record Start time [utc]
07:00

Record Stop time [utc]
21:00

Username
MY_USER

Password
MY_PASSWORD

Camera Enabled

[Apply Changes](#)
[Delete Camera 0000-0000](#)

Abbildung 70 AP Gateway User Interface Kameraeinstellungen

Für das manuelle Hinzufügen einer Kamera müssen ID, URL und Credentials eingegeben werden. Die restlichen Parameter werden auf die Standardwerte gesetzt und können nach dem Vorgang abgeändert werden. Ein Screenshot der Seite ist in Abbildung 71 ersichtlich.

Add Camera

Enter the camera ID
0000-0000

Enter the capture URL
http://cam-0000-0000.local:8080/?action=snapshot

Username
MY_USER

Password
MY_PASSWORD

You can modify all parameters after the registration.

[Add Camera](#)

Abbildung 71 AP Gateway User Interface Hinzufügen von Kameras

In den globalen Einstellungen wird der Pfad für die Speicherung der Bilder und die Standardkonfiguration für die Kameras definiert. Durch eine Checkbox wird festgelegt, ob eine neu hinzugefügte Kamera automatisch aktiviert wird oder nicht. Abbildung 72 zeigt die Seite für die Anpassungen der globalen Einstellungen.

Data Directory

Directory to store data
/mnt/usb/captures

Default Camera Configuration

Camera Port
8080

URL Action
/?action=snapshot

Capture Interval [s]
20

Record Start time [utc]
07:00

Record Stop time [utc]
21:00

Username
MY_USER

Password
MY_PASSWORD

Enabled by default

[Apply](#)

Abbildung 72 AP Gateway User Interface Globale Einstellungen

5.3 Sensor Nodes

Für die Sensor Nodes wurden mehrere Optimierungen der Software implementiert. Die Implementierungen umschliessen eine benutzerfreundliche Konfiguration, ein Deployment ohne einer IDE und die Unterstützung von mehreren Sensoren. Auf Seite der Hardware wurden Optimierungen zum Schutz der Elektronik vor äusseren Einflüssen implementiert.

5.3.1 Hardware

Um die Hardware vor Feuchtigkeit zu schützen, müssen Kabelverschraubungen eingesetzt werden, was ein grösseres Gehäuse erfordert. Die minimale Anforderung an das Gehäuse ist eine IP Schutzklasse von IP65. Um die Nodes auch bei trübem Wetter zu betreiben, sollen die Abmessungen gross genug sein, um einen grösseren Akku einzubauen. Um Beschädigungen beim Transport zu vermeiden, muss eine Möglichkeit bestehen, alle Komponenten im Inneren des Gehäuse fest zu montieren. Der Preis des Gehäuse muss im unteren Segment liegen. Es wurden mehrere Gehäuse bestellt und verschiedene Anordnungen der Bauteile in Autodesk Inventor analysiert. Zwei verschiedene Versionen, eine mit einem kleineren Gehäuse und eine mit einem grösseren Gehäuse wurden implementiert.

5.3.1.1 Version BOX4U Gehäuse

Für die kleine Version wird ein ABS Gehäuse des Herstellers BOX4U verwendet. Die Aussenmasse des IP65 Gehäuse betragen 115.4 x 125.3 x 58.05mm. Die Anschaffungskosten bei Bezug eines Einzelstücks liegen bei 5.85 CHF [99]. Um den Schutz gegen Feuchtigkeit zu gewährleisten, muss eine Gummidichtung in die dafür vorgesehenen Rillen im Deckel eingesetzt werden. Der Deckel wird durch Schraubverbindungen befestigt. Öffnungen für Kabeleinführungen sind nicht vorhanden. Das Gehäuse ist in Abbildung 73 zu sehen.



Abbildung 73 ABS Gehäuse BOX4U [99]

Für die Befestigung der Elektronikbauteile wurde eine Halterung konstruiert, die in das Gehäuse geschraubt werden kann. Die Halterung enthält Bohrungen zur Montage des Grove Shields und eine Trennwand zum Schutz des Akkus. Für die Kabeleinführungen des Solar Panels und der Sensoren wurden auf der Unterseite des Gehäuse zwei Bohrungen mit Durchmesser 16 mm und 20 mm erstellt. Um das Risiko von eindringender Feuchtigkeit zu minimieren befindet sich die gesamte Antenne im inneren des Gehäuse. Abbildung 74 zeigt eine Aufsicht der Konstruktion mit den entsprechenden Bauteilen. Der Akku wird mit selbstklebenden Klettstreifen in dem Gehäuse befestigt.

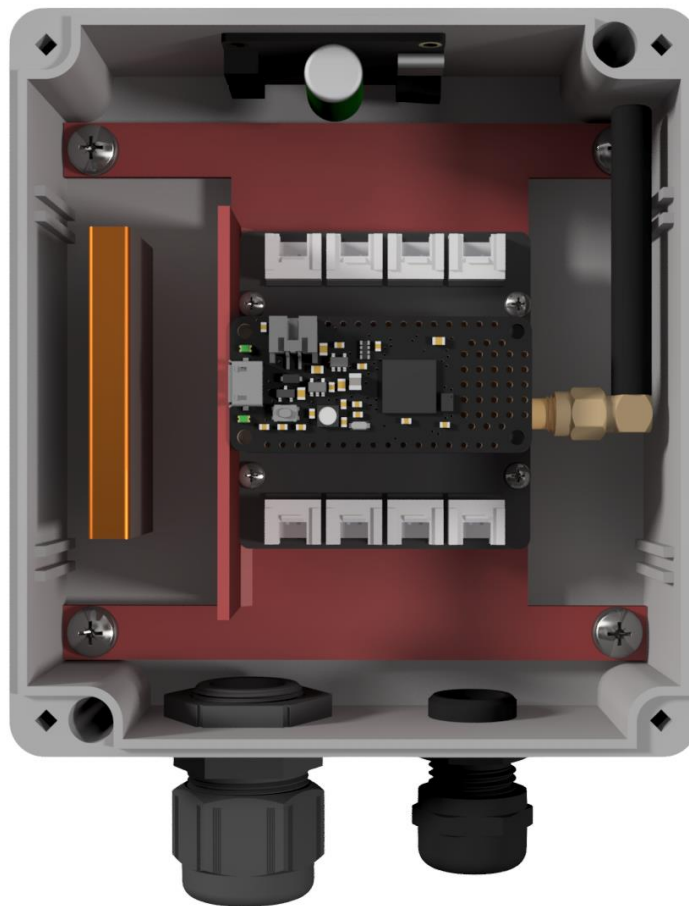


Abbildung 74 Sensor Node Anordnung der Bauteile in BOX4U Gehäuse

Die Kabel der Sensoren, die auf den oberen Steckern angeschlossen werden, können unterhalb von dem Grove Shield durchgeführt werden, da ein Abstand eingeplant wurde. Durchführungsmöglichkeiten für die Kabel sind in Abbildung 75 erkennbar.

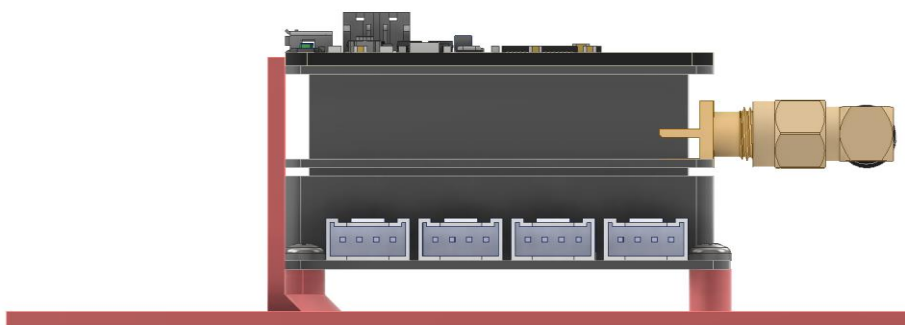


Abbildung 75 Sensor Node Seitenansicht Montageplatte für BOX4U Gehäuse

Die konstruierte Halterung wurde aus dem Kunststoff PLA gefertigt und die Bauteile darauf befestigt. Um die Platzverhältnisse zu prüfen, wurde ein wasserdichter Distanzsensor und ein Bodentempersensoren angeschlossen, wie in Abbildung 76 ersichtlich. Das LiPo Charging Modul und das Solar Panel wurden noch nicht angeschlossen. Die Platzverhältnisse für die Anschlussstecker sind im Vergleich zu der ursprünglichen Version sehr grosszügig berechnet. Trotzdem könnte der Platz bei mehr als vier angeschlossenen Sensoren, die als Kabel und nicht als einzelne Drähte in das Gehäuse geführt werden, knapp werden. Die maximalen Abmessungen des Akkus betragen 95 x 25 x 30 mm. Dieses Gehäuse eignet sich für einen Einsatz mit einer geringen Anzahl angeschlossener Sensoren. Das Gehäuse bietet einen wesentlich besseren Schutz vor äusseren Einflüssen als die ursprüngliche Variante, hat als Trade Off aber grössere Abmessungen.

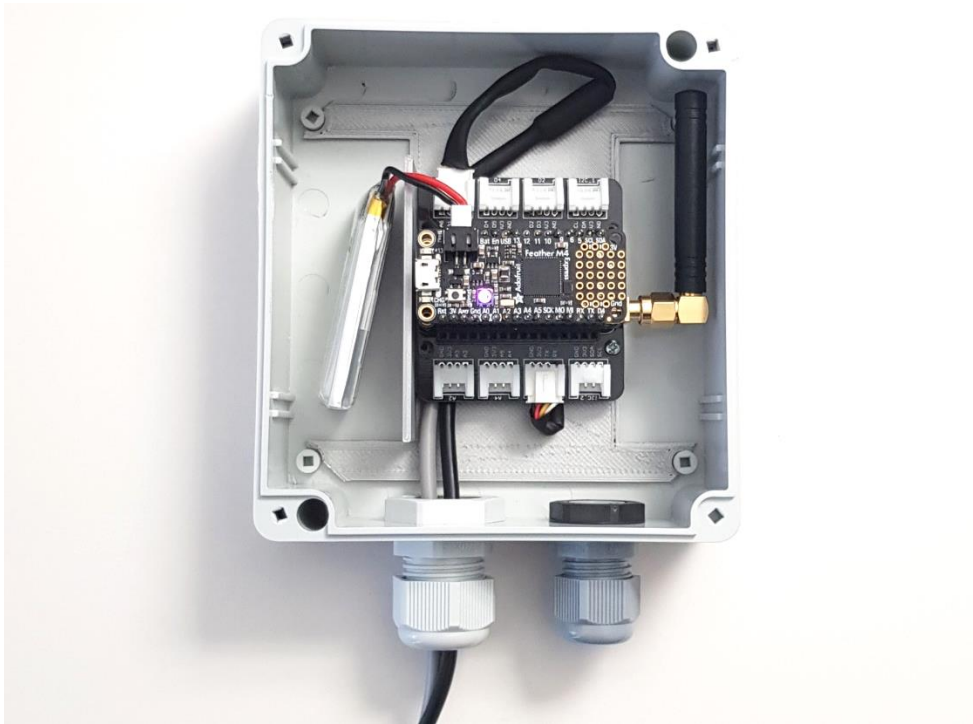


Abbildung 76 Sensor Node in BOX4U Gehäuse

5.3.1.2 Version mit WISKA Gehäuse

Für die zweite Version wurde ein Gehäuse des Herstellers WISKA verwendet. Das Thermoplast Gehäuse hat Aussenabmessungen von 120 x 160 x 170 mm und bietet einen Berührungs- und Wasserschutz der Klasse IP65. Der Deckel wird durch ein Schnellschraubensystem verschlossen. Die Anschaffungskosten bei Bezug eines Einzelstücks liegen bei knapp 6 CHF [100]. Eine Version mit transparentem Deckel kostet 8.30 CHF. Auf der inneren Rückwand des Gehäuse befinden sich Bohrungen, welche für die Befestigung von Bauteilen mittels M3 Schrauben vorgesehen sind. Das Produkt ist in Abbildung 77 ersichtlich.



Abbildung 77 WISKA Gehäuse [100]

Wie für die Version mit dem BOX4U Gehäuse wurden die Bauteile im CAD Programm angeordnet und eine Halterung für die mechanische Fixierung der Module entworfen. Mehr als ein Drittel des vorhandenen Platzes kann für einen grösseren Akku verwendet werden. Abbildung 78 zeigt die angeordneten Bauteile in dem WISKA Gehäuse. Als Akku wurde ein Modell aus sechs 18650 LiPo Zellen, was einer Kapazität von über 20'000 mAh bei 3.7 V entspricht, eingeplant. Das LiPo Charging Modul wird in der konstruierten Halterung durch eine Schraube blockiert, um

Kurzschlüsse zu verhindern. Die Höhe der Trennwand entspricht der Innenhöhe des Gehäuses. Bohrungen für die Kabeleinführungen müssen erstellt werden.

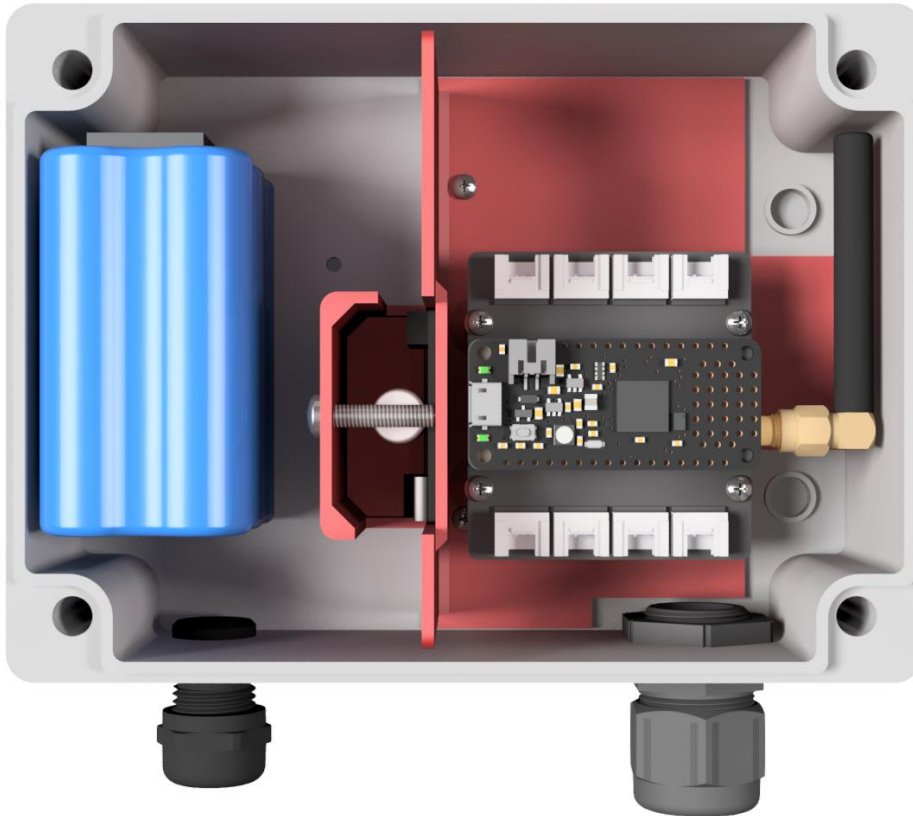


Abbildung 78 Sensor Node Anordnung der Bauteile in WISKA Gehäuse

In der Höhe bietet das WISKA Gehäuse 12 mm mehr Platz als das von BOX4U. Somit war es möglich, mehr Platz für die Kabeldurchführungen unterhalb des Grove Shields einzuplanen. Eine Seitenansicht ist in Abbildung 79 zu sehen.

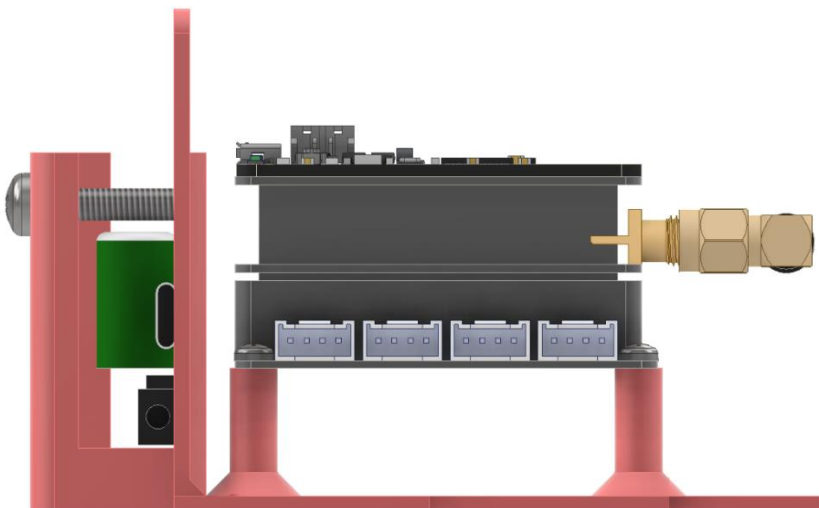


Abbildung 79 Sensor Node Seitenansicht Montageplatte für WISKA Gehäuse

Die Halterung wurde aus dem Werkstoff PLA gefertigt und die Module darauf befestigt. Im Vergleich zu der BOX4U treten keine Platzprobleme bei den Anschlüssen der Sensoren auf. Abbildung 80 zeigt ein Sensor Node mit drei angeschlossenen Sensoren in dem Wiska Gehäuse mit transparentem Deckel. Es ist erkennbar, dass im Gehäuse genügend Platz für die Anschlüsse und für Klemmverbindungen von Sensorkabeln ohne Grove Stecker vorhanden ist.



Abbildung 80 Sensor Node in Wiska Gehäuse

Das Sensor Node hat in dem Gehäuse wesentlich grössere Abmessungen als die ursprüngliche Version. Der Einsatz lohnt sich, falls viele Sensoren angeschlossen werden oder aufgrund des Einsatzortes ein grösserer Akku erforderlich ist.

5.3.1.3 Elektroschema

Abbildung 81 zeigt den Verdrahtungsplan des Sensor Nodes. Die Sensoren sind vereinfacht als Block visualisiert.

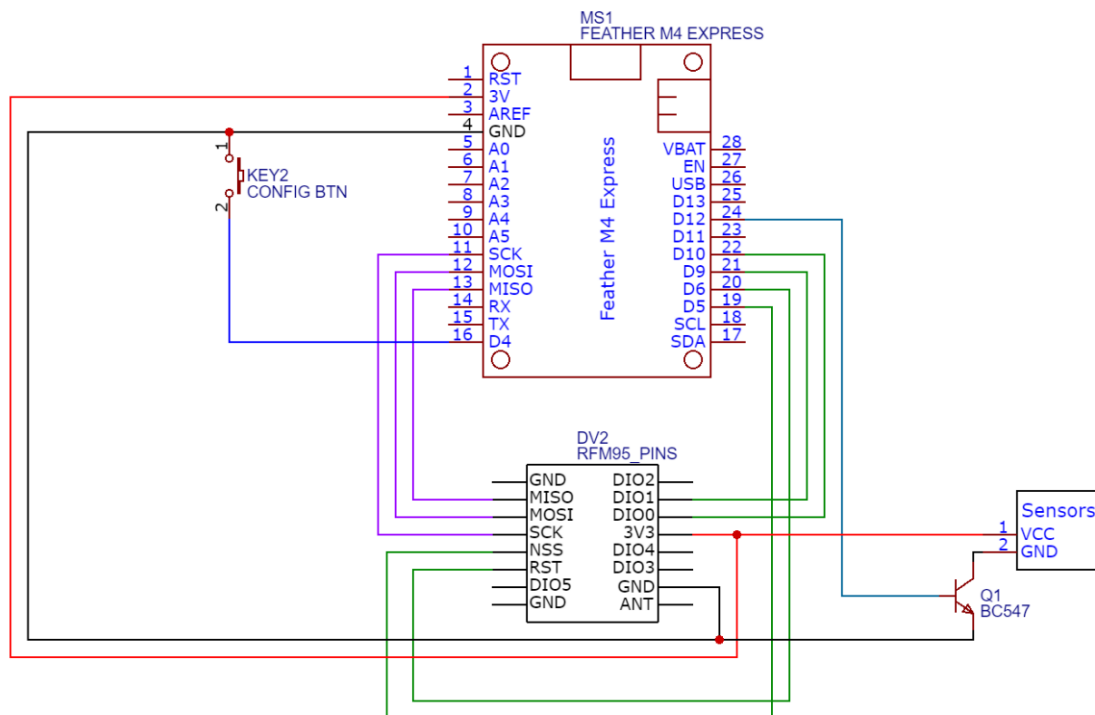


Abbildung 81 Elektroschema Sensor Node

5.3.2 Energiemanagement

Um die Nodes im Akkubetrieb zu betreiben, muss der Energieverbrauch möglichst gering gehalten werden. Wie in der bereits bestehenden Version wird der Prozessor zwischen den Mess- und Sendevorgängen in den Standby Sleep Mode versetzt, wobei die CPU und die Main Clock komplett ausgeschaltet werden. Der typische Stromverbrauch in Standby Mode liegt gemäss Datenblatt bei 85 μ A [101].

5.3.2.1 Ausschalten von Sensoren

Die Sensoren verbrauchen je nach Typ mehrere mA im Standby Betrieb. Um den Verbrauch zu minimieren, wurde eine Unterbrechung der Spannungsversorgung für die Sensoren zwischen den Messvorgängen implementiert. Über einen digitalen Ausgang wird ein NPN BC547 Transistor angesteuert, welcher das Ground Potential der Sensoren von dem des Boards trennt. Abbildung 82 zeigt das Schema der Schaltung. Alle Sensoren sind vereinfacht als ein Block gezeichnet.

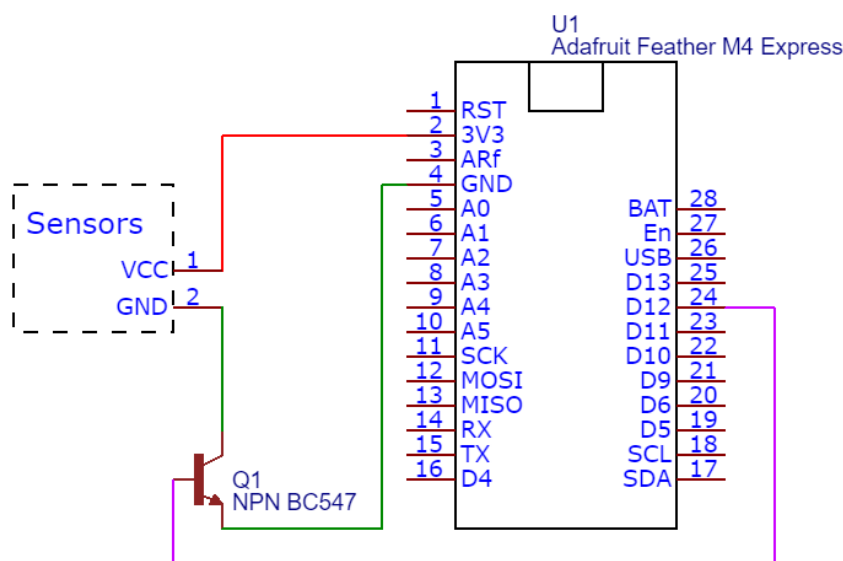


Abbildung 82 Schema Energiemanagement Sensoren Feather M4

Es gibt Sensoren, zum Beispiel Gassensoren, die eine gewisse Zeit zum Initialisieren brauchen, um ein Heizelement auf eine bestimmte Temperatur aufzuwärmen. Falls entsprechende Sensoren eingesetzt werden, muss auf die Schaltung verzichtet werden.

5.3.2.2 Solar Panel

Die Energie des bisher eingesetzten 1 W Solar Panels genügt nicht, um den Akku bei bewölktem Wetter aufzuladen. Die eingesetzte Ladeschaltung ist für Eingangsspannungen von bis zu 10 V dimensioniert. Um den Unterschied zu überprüfen wurde ein 6 W Solar Panel mit eingebautem 5 V Regulator eingesetzt, welches in Abbildung 80 gezeigt wird.

Der Anschaffungspreis liegt bei rund 12 CHF [102] bei einer Bestellung aus China. Gemäss Herstellerangaben ist das Panel für Aussenanwendungen geeignet. Für den Anschluss wurde die USB Buchse des Panels demontiert und ein entsprechenden Verbindungsstecker angeschlossen. Schon bei mässiger Umgebungsbeleuchtung in Innenräumen hat die Ladeschaltung angezeigt, dass der Akku geladen wird.

Um die Solar Panels zu vergleichen, wurden zwei Nodes an einem hellen aber schattigen Ort platziert. Die Umgebungstemperatur lag zwischen 0 und 4 °C. Das 6 Watt Solar Panel hat den Akku nach einigen Stunden vollständig geladen, während beim 1 W Panel lediglich zwei Peaks

zu erkennen sind, die zeigen, dass sich der Akku zu diesen Zeitpunkten im Ladezustand befand. Die Entwicklung der Spannungen sind in Abbildung 83 zu sehen.

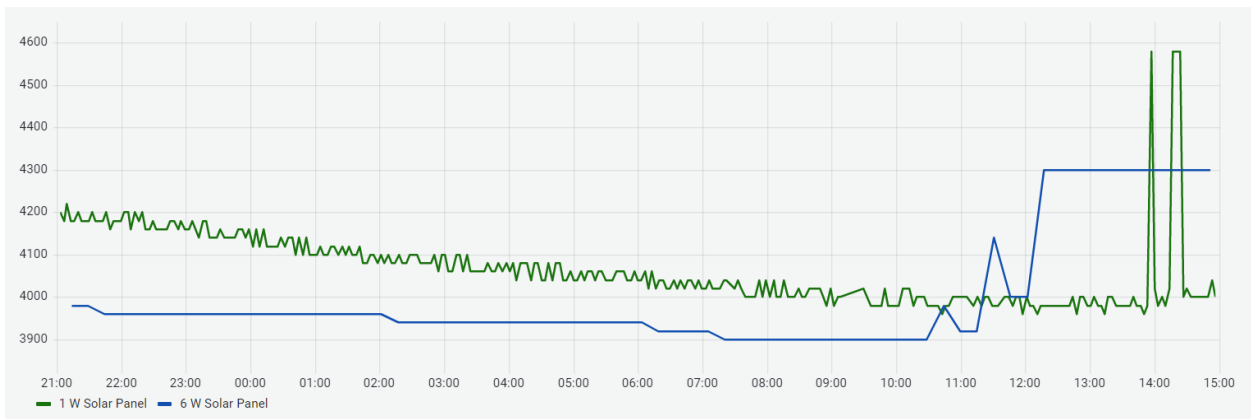


Abbildung 83 Solar Panel Vergleich Akkuladung

Der längere Einsatz von LiPo Akkus bei Temperaturen unter 0 °C verkürzt die Lebensdauer. Insbesondere beim Ladevorgang können bei Temperaturen unter -7 °C einzelne Zellen beschädigt werden. Eine kältebeständige Alternative zu den LiPo Akkus sind Bleiakkus. Sie verfügen über eine wesentlich geringere Energiedichte und die Bestandteile sind umweltschädlich. Der Einsatz von Bleiakkus setzt eine andere Ladeschaltung und ein grösser dimensioniertes Gehäuse voraus. Die Sensor Nodes in der Projektarbeit beschränken sich auf LiPo Akkus, womit sie für den Einsatz bei Temperaturen über der kritischen Temperatur ausgelegt sind.

5.3.3 Unterstützte Sensoren

Die Sensoren der Sensor Nodes beschränkten sich in der ersten Version auf die Messung von Lufttemperatur, Luftfeuchtigkeit und Bodenfeuchtigkeit. Um mehr Messgrößen zu unterstützen, wurden acht Sensoren implementiert, die in Tabelle 9 aufgelistet sind.

Messgrösse	Sensortyp	Anschluss
Lufttemperatur und Relative Luftfeuchtigkeit	SHT31	I2C
Bodenfeuchtigkeit	Kapazitiver Feuchtigkeits-sensor	ADC
Bodentemperatur	DS18B20	OneWire
UV Index	GUVA-S12D	ADC
Distanz	A02YYUW Waterproof Ultra-sonic distance sensor	Serial
Wassertrübung	Grove Turbidity Sensor V1.0	ADC
Gaskonzentration (CO, NO2, VOC, C2H5OH) in der Luft	Grove Multichannel Gas Sensor	I2C
Umgebungslautstärke	Grove Sound Sensor (mit L358 Verstärker)	ADC

Tabelle 9 Unterstützte Sensoren für Sensor Node

5.3.4 Connectivity

Als zusätzliche Optionen für die LoRaWAN Connectivity wurde eine Geräteaktivierung über OTAA implementiert. Bei Testläufen mit einem LoRaWAN Gateway in der näheren Umgebung konnte sich das Node innerhalb der ersten 15 Minuten im Netzwerk anmelden. An Standorten,

an denen die Netzwerkabdeckung zwar vorhanden aber sehr schwach ist, dauerte der Anmeldevorgang mehrere Stunden. Der Einsatz lohnt sich nur, wenn eine gute Netzwerkabdeckung garantiert ist.

Eine weitere Option, die implementiert wurde, ist das Senden mit einem konstanten Spreading Factor. Dabei wird die Adaptive Datenrate deaktiviert und der Spreading Factor nicht durch die Gateways in der Umgebung beeinflusst. Mit dieser Option ist die Wahrscheinlichkeit, dass eine gesendete Nachricht von einem Gateway empfangen wird, bei einem hohen Spreading Factor höher. Die Nachteile der Option liegen bei dem höheren Konsum von Airtime und Energie zum Senden, wenn auch ein tieferer Spreading Factor ausreichen würde.

5.3.5 Applikation

Nachfolgend werden die Abläufe der Applikation im Überblick beschrieben. Der Source Code befindet sich im Anhang und auf GitHub [103].

5.3.5.1 Setup

Beim Startvorgang der Applikation wird geprüft, ob der Taster zum Aktivieren des Konfigurationsmodus betätigt wird. Im Falle einer Betätigung bleibt die Applikation im Konfigurationsmodus bis der Reset Button betätigt wird. Der Modus wird im Unterkapitel *Konfiguration* beschrieben. Im normalen Modus wird die Konfigurationsdatei aus dem Flash Speicher gelesen und die Parameter extrahiert. Falls die LoRa Credentials nicht oder in ungültigem Format vorliegen, stoppt die Applikation und die NeoPixel LED blinkt in roter Farbe. Gemäss der Aktivierungsmethode wird anschliessend das LoRa Modul initialisiert und in den Main Loop gewechselt. Ein Flussdiagramm des Setup Vorgangs ist in Abbildung 84 ersichtlich.

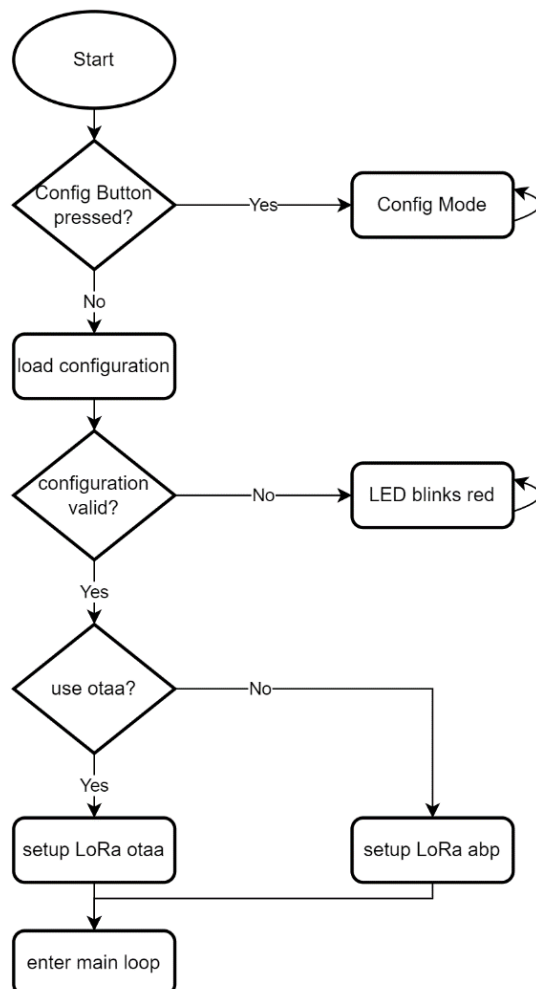


Abbildung 84 Sensor Node Flowchart Setup

5.3.5.2 Main Loop

Wenn OTAA als Aktivierungsmethode eingesetzt wird, muss sich das Node im Netzwerk anmelden, bevor es Messwerte übermitteln kann. Bei fehlgeschlagenen Anmeldeversuchen wird der Vorgang wiederholt, bis sie erfolgreich ist. Nach erfolgreicher Anmeldung oder bei der Aktivierung mittels ABP werden die Messwerte der Sensoren ausgelesen und die Payload zusammengestellt. Um die Nachrichten in einem konstanten Intervall zu übermitteln, wird zu Beginn des Messvorgangs die Systemzeit gespeichert. Anschliessend wird die Übertragung der Payload gestartet und das LoRa Modul auf Events überprüft. Nach Eintreffen des Event TX_COMPLETE wird der Zeitpunkt des nächsten Messvorgangs berechnet. Anschliessend versetzt sich das Node in den Deep Sleep Mode. Abbildung 85 zeigt ein Flussdiagramm des Ablaufs.

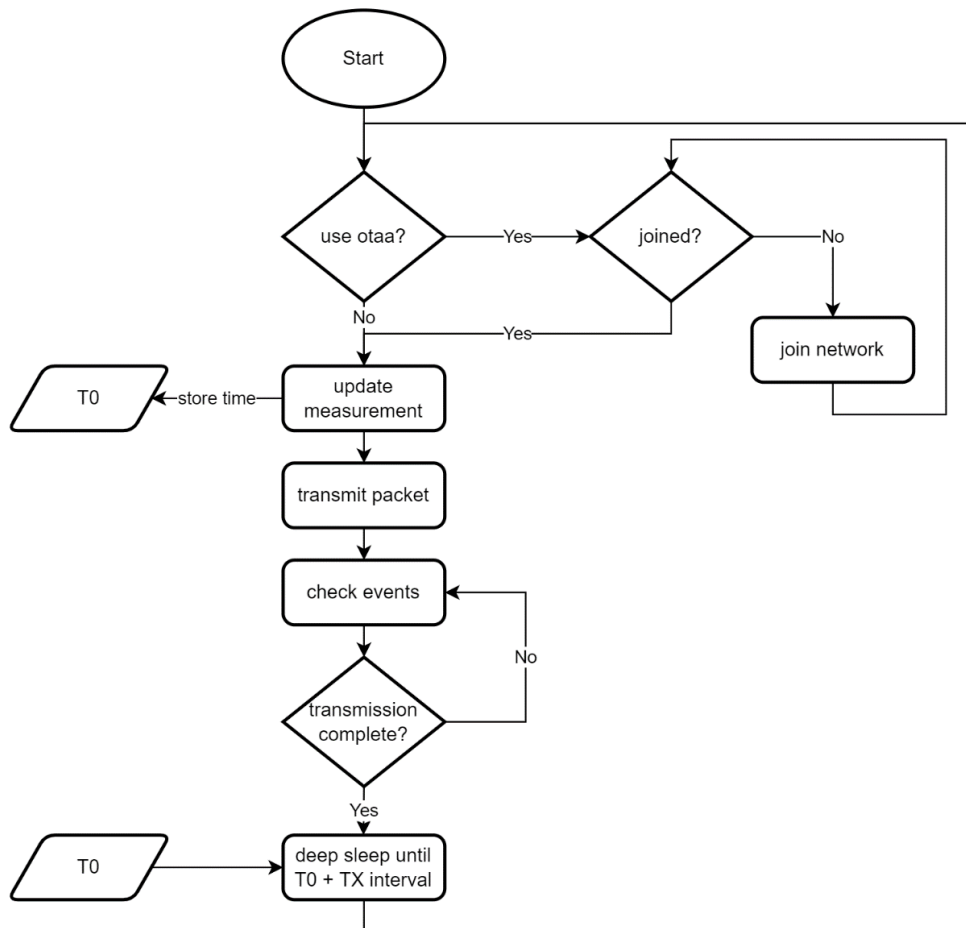


Abbildung 85 Sensor Node Flowchart Main

5.3.6 Deployment

Der grösste Aufwand für das Deployment liegt bei der Installation der Arduino IDE und den erforderlichen Libraries. Als Alternative wurde die Möglichkeit implementiert, das Board mit der vorkompilierten Software im UF2 Format mittels Drag and Drop zu programmieren.

Dafür wurde das Programm kompiliert und in binärem Format gespeichert. Für die Umwandlung der binären Datei in das UF2 Format stellt Microsoft ein entsprechendes Tool [104] zur Verfügung. Im Anhang und auf GitHub [103] befinden sich jeweils eine UF2 Version mit seriellem Debug Output und ohne Output.

Für die Programmierung des Nodes muss es über USB mit einem PC verbunden werden. Nach einem Doppelklick des Reset Buttons erkennt der PC ein USB Speichergerät mit dem Namen *FEATHERBOOT*. Zum Schreiben des Programms auf das Board wird die UF2 Datei auf das USB Speichergerät kopiert. Nach abgeschlossenem Kopiervorgang startet das Board die Applikation.

5.3.7 Konfiguration

Die Konfiguration der Nodes geschieht durch eine Konfigurationsdatei, welche über USB auf das Feather M4 Board geladen wird. Bei dem Startvorgang des Boards wird überprüft, ob eine Verbindung zwischen GPIO D4 und GND vorliegt. Die Verbindung wird durch das Bedienen des Config Tasters erstellt. Wird der Taster betätigt, wird der Konfigurationsmodus aktiviert und der Flash Speicher auf dem Board als USB-Massenspeichergerät initialisiert. Der PC, an welchem das Board angeschlossen ist, erkennt es als USB-Speichergerät. Auf dem Flash Speicher liegt die Konfigurationsdatei mit dem Namen *configuration.conf*. Die Datei kann in einem Text Editor bearbeitet werden. Bei jedem Speichervorgang wird auf dem Board überprüft, ob die LoRa Credentials im richtigen Format vorliegen. Im Erfolgsfall leuchtet die NeoPixel LED auf dem Board grün, bei ungültigen Formaten rot. Die Konfigurationsdatei beinhaltet Informationen über angeschlossene Sensoren, die LoRa Aktivierungsmethode, den Sendeintervall und den Spreading Factor.

5.3.7.1 Generieren der Konfigurationsdatei

Für das benutzerfreundliche Generieren der Konfigurationsdatei wurde ein HTML Dokument geschrieben. Mittels JavaScript Funktionen werden die nicht erforderlichen Eingabefelder ausgeblendet und die Eingabe validiert. Der Generator ist unter <https://wullt.github.io/MultisensorLoRaNode/> erreichbar. In der ersten Sektion werden Connectivity Parameter eingegeben. Die Felder der erforderlichen LoRa Credentials werden gemäss ausgewählter Aktivierungsmethode aktiviert oder deaktiviert. Abbildung 86 zeigt einen Screenshot der ersten Sektion.

The screenshot shows a web form titled "Connectivity". It contains several sections:

- Spreading Factor:** A dropdown menu with "SF 8" selected.
- Adaptive Data Rate:** A checkbox labeled "Use ADR" which is checked.
- Activation Method:** A dropdown menu with "ABP" selected.
- Paste TTN Credentials in MSB Format:** A heading for the ABP section.
- ABP:**
 - Device address:** Text input field containing "01234567".
 - NwkSKey:** Text input field containing "0123456789ABCDEF0123456789ABCDEF".
 - AppSKey:** Text input field containing "0123456789ABCDEF0123456789ABCDEF".
- OTAA:**
 - AppEUI:** Text input field containing "0123456789ABCDEF".
 - DevEUI:** Text input field containing "0123456789ABCDEF".
 - AppKey:** Text input field containing "0123456789ABCDEF0123456789ABCDEF".
- Transmit Interval (s):** Text input field containing "900".

Abbildung 86 Sensor Node Konfigurationsgenerator Connectivity

In der zweiten Sektion werden angeschlossene Sensoren ausgewählt, indem die Checkbox betätigt wird. Für die ausgewählten Sensoren können die Anschlusspin oder I2C Adressen geändert

werden. Für alle Sensoren ist ein Link aufgeführt, welcher mehr Informationen über das Produkt enthält. Im unteren Bereich der Sektion befindet sich ein Button mit der Bezeichnung Generate, welcher die Konfigurationsdatei generiert. Wenn ein Pin doppelt definiert ist oder Credentials in falschem Format vorliegen, wird es dem Benutzer mit einer Meldung angezeigt und das File wird nicht generiert. Abbildung 87 zeigt einen Screenshot der zweiten Sektion.

Sensors

Air Temperature & Humidity		
<input checked="" type="checkbox"/> SHT31 Temp and Humi Sensor	DFRobot Wiki Grove Wiki	I2C Address 0x44
Soil Moisture		
<input checked="" type="checkbox"/> Capacitive Soil Moisture Sensor	DFRobot Wiki Grove Wiki	Analog Pin A0
Soil Temperature		
<input checked="" type="checkbox"/> DS18B20 Temperature Sensor	Grove Wiki	Analog/Digital Pin A2
Multichannel Gas Sensor		
<input type="checkbox"/> Gas Sensor V2(Multichannel)	Grove Wiki	I2C Address 0x08
UV Index		
<input checked="" type="checkbox"/> UV Sensor	Grove Wiki	Analog Pin A4
Water Turbidity		
<input type="checkbox"/> Turbidity Sensor Meter v1.0	Grove Wiki	Analog Pin A0
Sound Sensor		
<input type="checkbox"/> Sound Sensor	Grove Wiki	Analog Pin A2
Distance		
<input type="checkbox"/> A02YYUW Ultrasonic Distance Sensor	DFRobot Wiki	Serial Port Serial1

Generate
Click to generate the config file

Abbildung 87 Sensor Node Konfigurationsgenerator Sensorauswahl

Am Ende des Dokuments befindet sich ein Textfeld mit dem Inhalt der Konfigurationsdatei. In den ersten Zeilen der Datei wird das Datum und die Uhrzeit der Generierung als Kommentar eingefügt. Durch Betätigung des Download Buttons kann die Konfigurationsdatei heruntergeladen werden. Abbildung 88 zeigt ein Screenshot der dritten Sektion.

Config File

```
# Sensors
has_sht31=true
sht31_i2c_addr=0x44
has_moisture_sensor=true
cap_moist_pin=A0
has_ds18b20=true
ds18b20_pin=A2
has_uv_sensor=true
uv_sensor_pin=A4
```

download

Abbildung 88 Sensor Node Konfigurationsgenerator Ausgabe

5.3.7.2 Statische Metadaten

Als statische Metadaten gelten die Node ID und die Koordinaten des Messstandorts. Sie werden in der Things Network Console konfiguriert und bei Bedarf modifiziert. Der Standort des Nodes wird mit der MQTT Integration standardmässig mitgeschickt. Die Node ID muss im Payload Decoder pro Node definiert werden.

5.3.8 Payload Format

Die Kombination der angeschlossenen Sensoren kann variieren. Es wurden zwei verschiedene Payload Formate entworfen.

Die erste Variante besteht aus einem Byte am Anfang der Nachricht, welches in Form einer Bit Maske die Information der angeschlossenen Sensoren beinhaltet. Für jeder der acht Sensoren existiert ein Bit, welches bei Vorhandensein auf 1 gesetzt wird. Die zusätzliche Datenmenge zur Beschreibung der Nachricht ist mit einem Byte minimal. Der Nachteil liegt bei der Beschränkung auf acht vordefinierte Sensoren, was die Erweiterbarkeit des Systems einschränkt.

In der zweite Variante, die schlussendlich implementiert wurde, wird vor jedem Messwert ein Byte gesetzt, welches die folgenden Bytes beschreibt. Somit sind bis zu 255 Sensoren möglich und die Erweiterbarkeit ist gewährleistet. Die zusätzliche Nachrichtengrösse steigt linear zu der Anzahl angeschlossener Sensoren an. Abbildung 89 zeigt den Aufbau der Payload mit der Bit Mask (oben) und mit der ID pro Feld (unten).

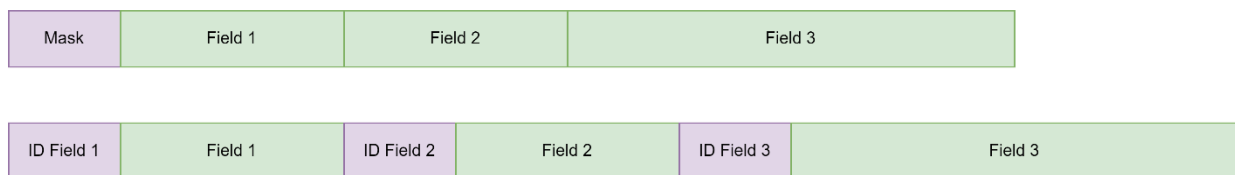


Abbildung 89 Varianten der Payload Strukturierung

Der Grund der Entscheidung für die zweite Variante liegt bei der Erweiterbarkeit des Systems, welche nicht eingeschränkt werden darf. Für die Dekodierung der Payload wurde eine JavaScript Funktion geschrieben, welche die binären Nachrichten im Thing Network Backend in eine JSON Nachricht umwandelt. Zusätzlich zu den Messwerten wird die in der Funktion definierte Node ID mitgesendet.

5.4 PAX Counter

Die implementierten Optimierungen für die PAX Counter umschliessen eine benutzerfreundliche Konfiguration, zusätzliche Connectivity Optionen ein einfaches Deployment.

5.4.1 Hardware

Um den Schutz der Hardware vor Feuchtigkeit zu gewährleisten, wurde ein Gehäuse des Herstellers RND eingesetzt. Das Gehäuse mit transparentem Deckel stammt aus der selben Modellreihe wie das Gehäuse für die PoE Kameras und weist die selben Eigenschaften auf. Die Beschaffungskosten für ein Einzelstück betragen 2.93 CHF. Durch die Dimensionen ist eine Fixierung der Komponenten ohne zusätzliche Montageplatte möglich. Abbildung 90 zeigt einen PAX Counter mit angeschlossenem Solar Panel.

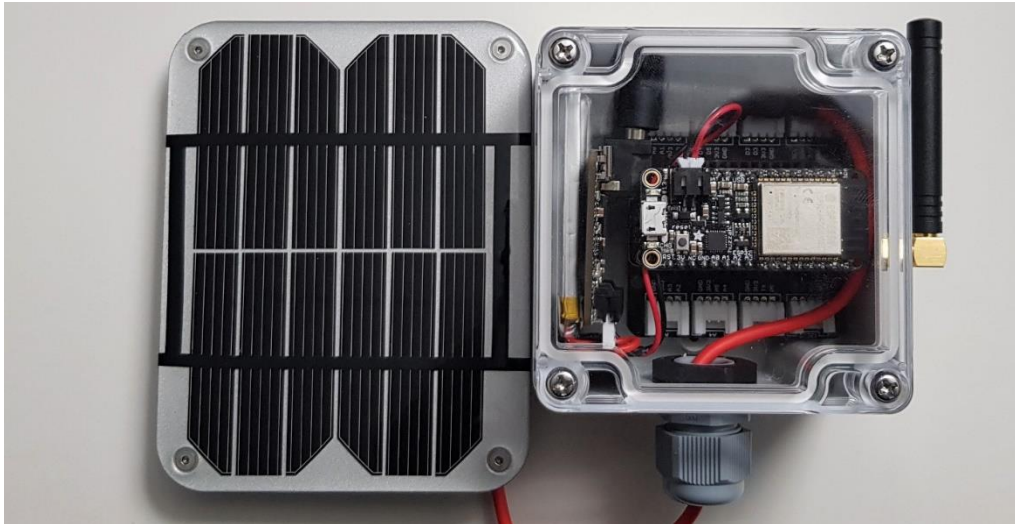


Abbildung 90 PAX Counter in RND Gehäuse mit angeschlossenem Solar Panel

Um den PAX Counter ohne regelmässig vorhandene Sonneneinstrahlung einzusetzen, kann, wie im Kapitel *Solar Panel* beschrieben, ein grösseres Solar Panel verbaut werden.

5.4.1.1 Elektroschema

Die Verdrahtung der Bestandteile ist im Schema in Abbildung 91 ersichtlich. Bei Verwendung eines Feather LoRa Moduls mit RFM95 Chip werden die entsprechenden Verbindungen durch das Aufstecken auf das Feather M4 Board erstellt. Für den Config Button kann ein Grove Button Modul [105] auf den Port A4 des Grove Shields gesteckt werden.

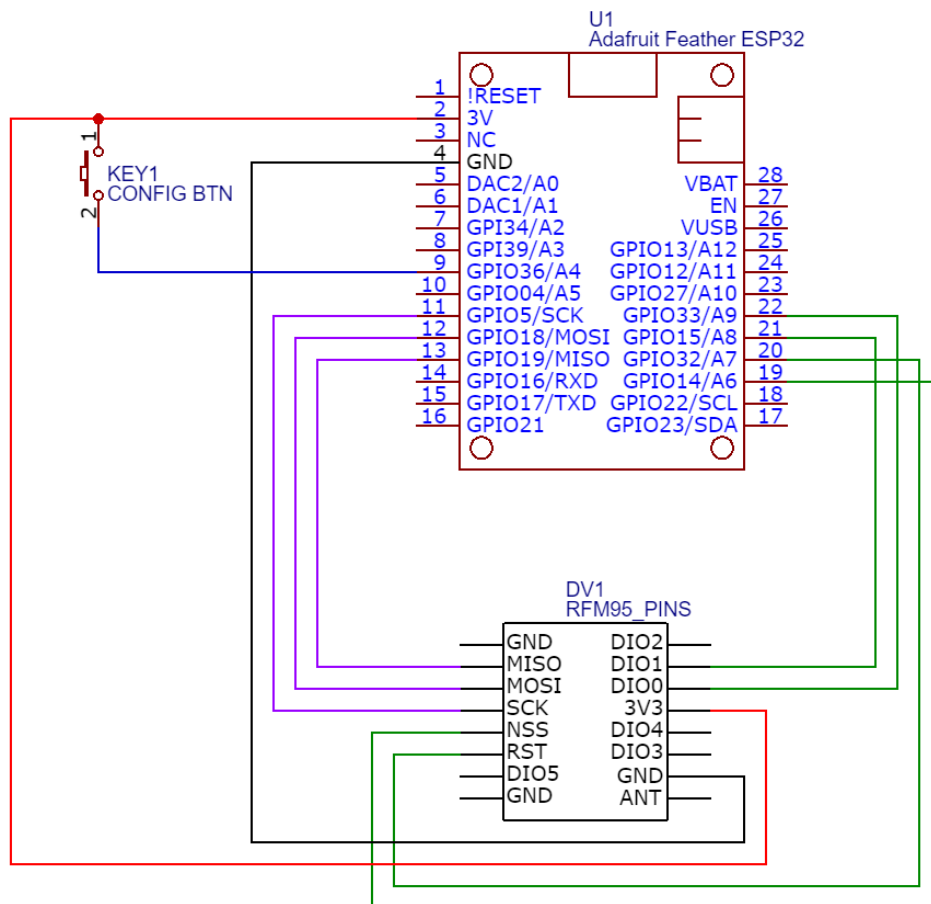


Abbildung 91 Elektroschema PAX Counter

5.4.2 Connectivity

Wie bei den Sensor Nodes wurde für die PAX Counter die Option einer OTAA Aktivierung und die Möglichkeit, ADR auszuschalten, implementiert. Zusätzlich wurde die Datenübertragung über MQTT via WiFi entwickelt, um Nachrichten in kürzeren Intervallen zu übermitteln. Die Übertragung kann mit oder ohne TLS Verschlüsselung durchgeführt werden.

5.4.3 Messvorgang

Als Messvorgang wird das Erfassen der BLE Broadcast Paketen bezeichnet, die Smartphones oder andere Geräte mit BLE Schnittstelle periodisch aussenden. Die Datenpakete enthalten die MAC Adresse des sendenden Geräts. Die meisten Smartphones verwenden aus Datenschutzgründen regelmässig neue, zufällige MAC Adressen, welche durch ihr Format von statischen Adressen unterscheidbar sind. Durch diese Eigenschaft ist es möglich, nur Geräte mit zufällig generierten Adressen zu erfassen. Die Option kann in der Konfiguration aktiviert oder deaktiviert werden.

Der Ablauf des Scanning Vorgangs ist in Abbildung 92 visualisiert. Der Scan Intervall bezeichnet den Intervall, in welchem ein Scan gestartet wird. Die Scan Duration bezeichnet die Dauer, während der BLE Pakete empfangen werden. In der Zeit, die bis zum nächsten Scan übrig bleibt, wird das Node in den Sleep Mode versetzt.

Zwischen dem Startzeitpunkt der Messung und der Übertragung der Daten an das Backend können maximal 512 Geräte gezählt werden.

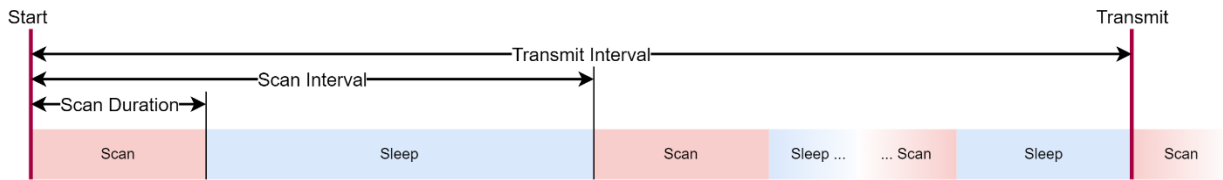


Abbildung 92 PAX Counter Zeitparameter Messvorgang

5.4.4 Applikation

Nachfolgend werden die Abläufe der Applikation im Überblick beschrieben. Der Source Code befindet sich im Anhang und auf GitHub [106].

5.4.4.1 Setup

Beim Starten des Programms liest es als erstes die Konfiguration aus dem EEPROM Speicher. Anschliessend überprüft das Programm den Zustand des Buttons zum Aktivieren des Konfigurationsmodus. Bei Betätigung wird der Konfigurationsmodus, der folgend dokumentiert ist, aktiviert. Im normalen Modus wird die BLE Schnittstelle initialisiert. Gemäss der konfigurierten Connectivity Option wird die entsprechende Schnittstelle initialisiert. Abbildung 93 zeigt ein Flussdiagramm des Ablaufs.

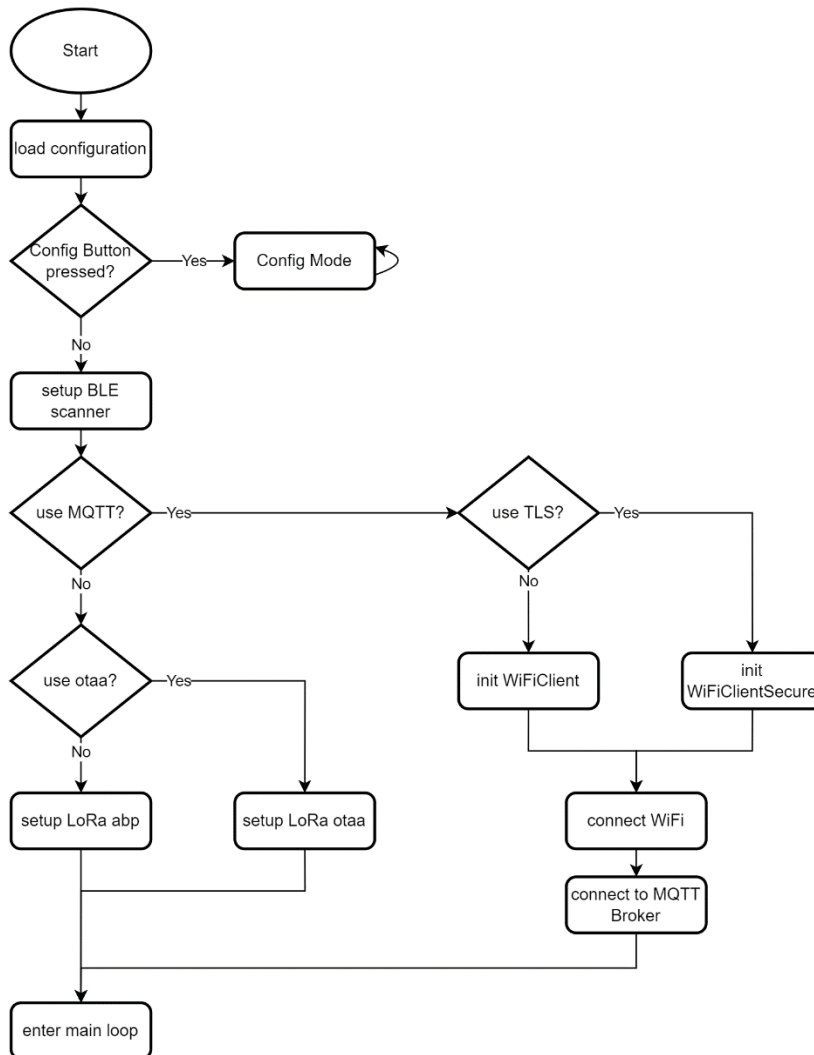


Abbildung 93 PAX Counter Flowchart Setup

5.4.4.2 Main Loop

Wenn eine Datenübertragung via MQTT und WiFi konfiguriert ist, findet eine Überprüfung des Verbindungszustandes statt. Wenn eine der Verbindungen unterbrochen ist, wird versucht, sie neu zu erstellen. Nach einer erfolgreichen Anmeldung bei dem MQTT Broker werden in dem konfigurierten Intervall BLE Scans durchgeführt. Zwischen den Scans wird die MQTT Verbindung aufrecht erhalten. Bei Erreichen des Sendezeitpunkts wird die Payload generiert und übertragen.

Die Verbindungsprüfung wird zeitlich vor dem Scannen durchgeführt, um Fehlerhafte Credentials möglichst früh zu erkennen und keine Daten zu erfassen, die nicht übertragen werden können.

Bei der Datenübertragung via LoRa wird, falls OTAA als Aktivierung verwendet wird, als erstes die Anmeldung in Netzwerk durchgeführt. Sobald das Node im Netzwerk angemeldet ist, oder wenn ABP verwendet wird, startet das periodische Scannen. Zwischen den Scans begibt sich das Node in den Light Sleep Modus. Bei Erreichen der Sendezeit wird die Payload generiert und übertragen. Nach dem abgeschlossenen Sendevorgang wird der periodische Scanning Vorgang erneut begonnen. Abbildung 94 zeigt den Ablauf des Main Loops als Flussdiagramm.

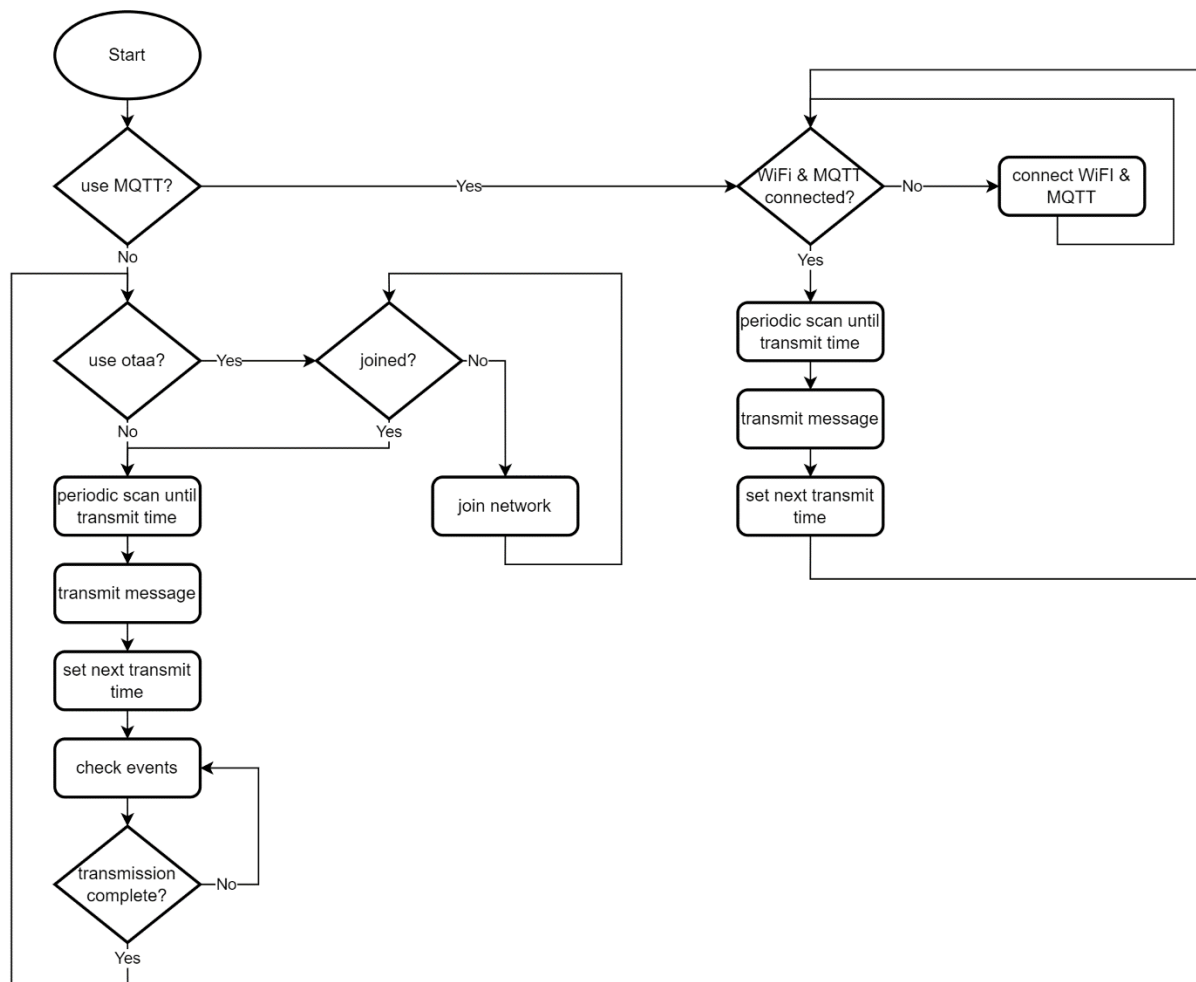


Abbildung 94 PAX Counter Flowchart Main Loop

5.4.5 Deployment

Das Kompilieren der Applikation erfordert eine installierte Arduino IDE. Zusätzlich müssen mehrere Libraries heruntergeladen werden, die auf GitHub [106] aufgelistet sind.

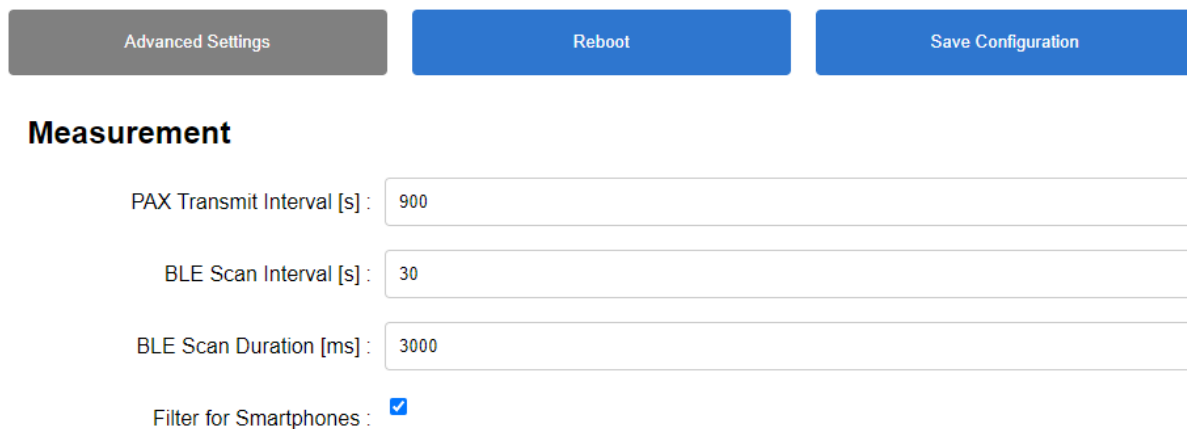
Für die Vereinfachung des Deployments wird die kompilierte Applikation in binärem Format bereitgestellt. Für den Upload der Applikation auf das ESP32 wurde ein Python Skript entwickelt. Neben einer Installation von Python erfordert die Ausführung des Skripts die Python Library

esptool [96], welche auch bei einem Upload mittels Arduino IDE genutzt wird. Das Upload Skript fordert den Benutzer nach dem Start auf, das ESP32 mit dem PC zu verbinden. Sobald die Verbindung besteht, beginnt der Upload und das Deployment ist vollendet. Das Skript funktioniert auf Windows und auf Linux Betriebssystemen.

5.4.6 Konfiguration

Für die Konfiguration der PAX Counter wurde der Konfigurationsmodus entwickelt. Um den Modus zu aktivieren, muss der Config Button beim Starten des Boards betätigt werden. Der aktivierte Config Mode ist durch die leuchtende LED auf dem Board erkennbar. Um die Geräteparameter zu modifizieren, wird auf dem ESP32 ein WiFi Access Point mit der standardmässigen SSID «PaxConfig» gestartet. Der Zugriff ist mit einem Passwort, standardmässig «password», geschützt. Nach der Aktivierung des Access Points wird auf dem Node ein asynchroner Webserver und ein Captive Portal gestartet. Das Captive Portal leitet alle Anfragen im Netzwerk auf den Webserver um. Sobald sich ein Gerät mit dem Access Point verbindet, wird es aufgefordert, sich im Netzwerk anzumelden. Als Anmeldeseite wird die URL des Webserver aufgerufen. Über den Webserver kann das Gerät konfiguriert werden. Die Konfiguration wird im EEPROM des ESP32 gespeichert.

Im Oberen Bereich der Benutzeroberfläche befinden sich die drei Buttons, die zur Navigation zu den erweiterten Einstellungen, zum Neustarten des Nodes und zum Speichern der Konfiguration dienen. Ein Screenshot der Sektion für die Konfiguration des Messvorgangs ist in Abbildung 95 ersichtlich. Sie beinhaltet Eingabefelder zum Festlegen des Sendeintervalls, des Scan Intervalls und der Scandauer. Durch eine Checkbox kann gewählt werden, ob nur Geräte mit zufälligen Adressen (hauptsächlich Smartphones) oder alle BLE Geräte gezählt werden sollen. Die Konfiguration kann nur gespeichert werden, wenn die Felder zulässige Werte enthalten.



The screenshot displays a configuration interface for the PAX Counter. At the top, there are three buttons: 'Advanced Settings' (grey), 'Reboot' (blue), and 'Save Configuration' (blue). Below the buttons, the section is titled 'Measurement'. It contains three input fields: 'PAX Transmit Interval [s]' with the value '900', 'BLE Scan Interval [s]' with the value '30', and 'BLE Scan Duration [ms]' with the value '3000'. Below these fields is a checkbox labeled 'Filter for Smartphones' which is checked.

Abbildung 95 PAX Counter Konfiguration Messung

Die Sektion zur Konfigurierung der Connectivity Parameter ist in Abbildung 96 ersichtlich. Im ersten Feld wird die Art der Übertragung, LoRa oder MQTT, ausgewählt. Gemäss der Auswahl werden die nicht benötigten Eingabefelder deaktiviert. Bei LoRa als Connectivity Option können alle Credentials im MSB Format von der Things Network Console kopiert und eingefügt werden.

Beim Einsatz von MQTT sind Anmeldedaten für ein WLAN Netzwerk, in welchem sich der PAX Counter anmelden kann, erforderlich. Anschliessend werden MQTT Broker URL und Port eingegeben. Die Verwendung von TLS kann durch eine Checkbox ein- oder ausgeschaltet werden. Für die Anmeldung beim Broker können Benutzername und Passwort eingegeben werden. Im untersten Feld wird das Topic, unter welchem die Nachrichten publiziert werden, festgelegt.

Anders als bei LoRa werden die statischen Metadaten bei MQTT direkt auf dem Gerät konfiguriert. Die Node ID und die Koordinaten werden in den entsprechenden Feldern eingetragen. Als MQTT Client ID wird die eingetragene Node ID verwendet.

Connectivity

Choose a connectivity option :

LoRa Credentials

Activation method :

Use ADR :

Spreading Factor :

ABP

Device Address :

NwkSKey :

AppSKey :

OTAA

AppEUI :

DevEUI :

AppKey :

WiFi Credentials

SSID :

WiFi Password :

MQTT Credentials

MQTT Broker URL :

MQTT Port :

Use TLS :

MQTT Username :

MQTT Password :

Node ID :

Latitude :

Longitude :

MQTT Topic :

Abbildung 96 PAX Counter Konfiguration Connectivity

Die erweiterten Einstellungen sind über die Schaltfläche «Advanced Settings» erreichbar. Sie bieten die Möglichkeit, die SSID und das Passwort des PAX Counter Access Points zu ändern. Da bei einem Zugriff auf den Access Point alle Credentials einsehbar sind, soll das Passwort geändert werden. Durch Betätigung der Schaltfläche «Save Configuration and restart» speichert der PAX Counter die neuen Credentials und führt einen Reboot durch.

Über die Schaltfläche «Reset Device» kann die gesamte Konfiguration vom Gerät gelöscht werden. Die Löschung erfordert eine Bestätigung vom Benutzer, um unbeabsichtigten Datenverlust zu verhindern. Abbildung 97 zeigt einen Screenshot der erweiterten Einstellungen.

Advanced Configuration

Configuration Portal WiFi Credentials

Make sure you have written down your credentials!

Config AP SSID :

Config AP Password :

Reset Device

Return

Save Configuration and restart

Abbildung 97 PAX Counter Konfiguration Advanced

5.4.7 Payload Format

Die Payload beinhaltet den PAX Wert, welcher die Anzahl der aktiven BLE Geräte im Messzeitraum repräsentiert, die Akkuspannung und den Status des MAC Filters. Bei aktiviertem MAC

Filter werden nur die BLE Geräte mit zufälligen Adressen gezählt. Falls er ausgeschaltet ist, werden die Adressen nicht gefiltert und jedes Gerät wird gezählt.

5.4.7.1 LoRa

Die über LoRa gesendete Payload besteht aus vier Bytes. Byte 0 und 1 beinhalten den PAX Wert, Byte 2 die Akkuspannung und Byte 3 den Status des MAC Filters. Die Payload wird im TTN Backend von einem in JavaScript geschriebenen Decoder in eine JSON Nachricht umgewandelt. Die Node ID wird im Decoder festgelegt. Die Koordinaten des Messstandortes werden in der Things Network Console definiert.

5.4.7.2 MQTT

Bei der Verwendung von MQTT als Connectivity Option wird die Payload im JSON Format übermittelt. Neben den Feldern für den PAX Wert, MAC Filter und Akkuspannung werden die statischen Metadaten mitgesendet.

5.5 Backend

Für die Speicherung, Analyse und Visualisierung der Sensordaten wurde eine vereinfachte Version des Backends entwickelt. Die darauf laufenden Applikationen werden virtualisiert und in Docker Containern ausgeführt. Für das Starten, die Zustandsüberwachung und eventuelle Restart Vorgänge ist Docker zuständig. Durch die Verwendung von Docker Compose wird das Deployment und die Konfiguration vereinfacht.

Das Backend kann auf einer virtuellen Maschine, auf einem lokalen Rechner oder Single Board Computer betrieben werden.

5.5.1 Applikationen

Die im Backend betriebenen Applikationen werden in folgenden Unterkapiteln beschrieben. Alle Docker Images werden von Docker Hub bezogen. Die Container wurden so konfiguriert, dass sie im Fehlerfall neu starten. Bei einem Reboot der Host Maschine werden die Container anschließend automatisch gestartet.

5.5.1.1 InfluxDB

InfluxDB [107] wird eingesetzt, um die Messwerte persistent zu speichern. Für die Zugriffsregelung werden drei Benutzer erstellt:

- Ein Admin Benutzer mit allen Berechtigungen
- Ein Benutzer mit der Berechtigung, Daten in die Datenbank zu schreiben
- Ein Benutzer mit der Berechtigung, Daten aus der Datenbank zu lesen

Die Datenbank wird, falls sie noch nicht besteht, bei der ersten Ausführung erstellt. Standardmäßig werden die Daten in einem Docker Volume gespeichert. Bei Bedarf kann ein lokaler Ordner als Speicherort gewählt werden.

Die Daten werden in InfluxDB nach Messwerttyp, der Zeit und den Metadaten strukturiert. Die vorhandenen Messwerttypen sind «pax» für die PAX Counter und «env» für die Umweltsensoren.

Als Docker Image wird *influxdb:1.8.10* eingesetzt, das auf ARM und AMD Architekturen ausgeführt werden kann. Aus anderen Containern wird über den Port 8086 auf die Datenbank zugegriffen.

5.5.1.2 Telegraf

Die Messwerte der Sensoren werden von der Things Network Applikation mit der MQTT Integration via MQTT bereitgestellt. Telegraf [108] ist ein Open Source Server Agent, der zum Sammeln von Sensordaten oder Systemdaten eingesetzt wird. Daten werden von Input Plugins erfasst und an die Output Plugins weitergeleitet. Die Konfiguration wird in einer Datei definiert.

Für das Empfangen der MQTT Nachrichten vom Things Network Backend wird das MQTT Input Plugin eingesetzt. Das Plugin meldet sich als Client beim Broker an und abonniert das festgelegte Topic. Die Nachrichten von der Things Network Applikation werden im JSON Format übertragen. Neben den Messwerten und den statischen Metadaten sind in den Nachrichten weitere Metadaten enthalten.

Um den Einsatz von PAX Countern mit MQTT Connectivity zu ermöglichen, wird eine zweite Instanz des MQTT Input Plugins erstellt, welches sich auf den lokalen MQTT Broker verbindet.

Für das Extrahieren der relevanten Daten aus den MQTT Nachrichten wird ein integrierter Parser genutzt. Mit GJSON Queries [109] werden die Datenfelder und die Felder der Metadaten aus der JSON Nachricht extrahiert. Pro Input Plugin wird ein Parser verwendet, um unterschiedliche Nachrichtenformate entsprechend zu verarbeiten.

Die Messdaten werden von dem InfluxDB Output Plugin in die Datenbank geschrieben. Um die Datenbank zu erreichen, müssen URL, Datenbankname und Credentials des InfluxDB Benutzers mit Schreibberechtigung konfiguriert werden.

Als Docker Image wird *telegraf:1.21.2* eingesetzt, womit ARM und AMD Architekturen unterstützt werden.

5.5.1.3 Grafana

Für die Analyse und die Visualisierung der Messwerte wird Grafana [29] eingesetzt. Grafana ermöglicht das benutzerfreundliche Erstellen von Dashboards und Charts. Es können mehrere Benutzer mit unterschiedlichen Rollen und Berechtigungen erstellt werden, um den Zugriff zu regeln. Standardmässig wird ein Admin Benutzer konfiguriert, der weitere Benutzer hinzufügen kann.

Der Grafana Container ist durch Port Mapping über die Host IP auf Port 3000 erreichbar. Um die Applikation über das Internet zu erreichen, eignet sich ein verschlüsselter Tunnel.

Als Docker Image wird *grafana/grafana:8.3.4* eingesetzt.

5.5.1.4 Mosquitto

Mosquitto ist ein Open Source MQTT Broker, der lokal betrieben wird. Er wird für die Übertragung von Daten der PAX Counter mit MQTT Connectivity eingesetzt. Die Berechtigungen zum Publizieren und Abonnieren von Topics werden pro User festgelegt.

Der Port 1883 des Containers ist durch Port Mapping über die Host IP erreichbar. Für den Zugriff über das Internet eignet sich ein verschlüsselter Tunnel.

Als Docker Image wird *eclipse-mosquitto:2.0.14* eingesetzt. Die Wahl von Mosquitto als MQTT Broker liegt bei der Kompatibilität des Docker Image mit ARM Architekturen und der grossen Popularität.

Als Alternative zu dem lokalen MQTT Broker kann ein Cloudbasierter Broker eingesetzt werden, was den Zugriff erleichtert und die Sicherheit erhöht.

5.5.2 Deployment

Alle erforderlichen Bestandteile für die Backend Software sind im Anhang und auf GitHub [110] verfügbar. Für das Deployment auf Linux Systemen wurden Shell Skripte geschrieben, die den Vorgang vereinfachen. Die Installation von Docker und Docker Compose und die Aktivierung des entsprechenden Systemd Service werden durch das Skript «install_docker_and_dc.sh» ausgeführt. Alle Konfigurationsdateien sind im Repository als Vorlage enthalten. Durch die Ausführung des Skripts «prepare_docker.sh» werden sie an die erforderlichen Orte kopiert.

Nach der Anpassung der Konfiguration, was im folgenden Unterkapitel beschrieben ist, können alle Container mit dem Befehl «docker-compose up -d» gestartet werden. Bei der ersten Ausführung werden die Images automatisch vom Docker Hub heruntergeladen.

5.5.3 Konfiguration

Eine detaillierte Anleitung der Konfiguration befindet sich im Anhang und auf GitHub [110]. Um die Konfiguration möglichst einfach zu gestalten, sind alle änderbaren Parameter der Docker Compose Datei als Umgebungsvariablen in einer Datei gespeichert. In der Datei werden die Credentials für InfluxDB und Grafana festgelegt. Die Anpassung erfolgt mit einem Texteditor.

Die Konfiguration für Telegraf wird in der Datei «telegraf.conf» angepasst. Erforderlich sind Anmeldedaten für den Things Network MQTT Broker, den lokalen MQTT Broker und für InfluxDB.

Für den lokalen MQTT Broker sind in der ACL drei Benutzer definiert. Der Admin User darf auf alle Topics publizieren und von allen Topics lesen. Der Read-User darf alle Topics abonnieren aber keine Nachrichten publizieren. Der Write-User darf auf alle Subtopics mit dem Präfix «data/» publizieren. Anpassungen der Credentials für die Benutzer werden nach dem Starten des Containers vorgenommen.

Sobald alle Container gestartet sind, kann mit einem Web Browser auf Grafana zugegriffen werden, um InfluxDB als Datenquelle hinzuzufügen. Grafana erreicht die Datenbank unter der URL «http://influxdb:8086». Da Grafana nur Daten aus der Datenbank liest, werden für den Zugriff die Credentials des InfluxDB Benutzers mit Leseberechtigungen verwendet. Beim Speichern der hinzugefügten Datenquelle wird die Verbindung von Grafana auf die Funktionalität getestet.

5.5.4 Visualisierung

In Grafana wurden mehrere Dashboards erstellt und als Vorlagen exportiert. Durch die Import Funktion können die Vorlagen in einer neuen Installation genutzt werden. Standardmässig werden die Dashboards im dunklen Anzeigemodus geladen. Für die folgenden Screenshots wurde der helle Anzeigemodus gewählt.

Eine Übersicht über die vorhandenen Sensor Nodes und PAX Counter und deren Standorte sind im Dashboard in Abbildung 98 ersichtlich. Der Base Layer der Karte kann beliebig abgeändert werden. Im rechten Bereich wird für alle vorhandenen Geräte automatisch eine Schaltfläche erzeugt, über die man auf ein Dashboard der Messwerte des entsprechenden Nodes gelangt. Die Schriftfarbe dient als Indikator der Akkuspannung. Falls kein Akku verwendet wird, wird violett verwendet.

Im unteren Bereich werden die eingehenden Datenpakete pro Node dargestellt, um Ausfälle früh zu erkennen.



Abbildung 98 Grafana Dashboard Übersicht

Die Detailansicht eines Sensor Nodes ist in Abbildung 99 ersichtlich. Über eine Variable, die oben links erkennbar ist, kann die Node ID des entsprechenden Gerätes angewählt werden. Auf einer Karte ist der genaue Standort erkennbar. Im rechten Bereich werden die im ausgewählten Zeitraum verfügbaren Messwerttypen in blauen Rechtecken angezeigt. Unterhalb befinden sich Anzeigeelemente für die aktuelle Akkuspannung, den Verlauf der Akkuspannung und den Empfangszeitpunkt der letzten Nachricht.

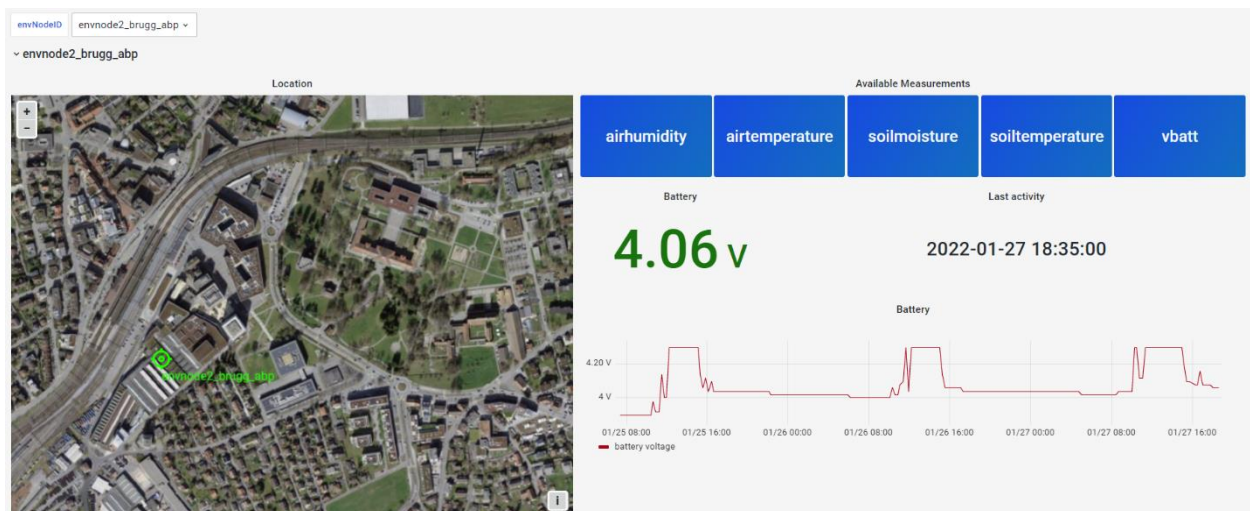


Abbildung 99 Grafana Dashboard Sensor Node

Im unteren Bereich des Dashboards befindet sich pro Messgröße ein Tab, in welchem die entsprechenden Messwerte visualisiert werden. Die Visualisierung von Luft- und Bodentemperatur und Bodenfeuchtigkeit sind in Abbildung 100 ersichtlich. Die Charts sind zoombar und einzelne Traces können ein- oder ausgeblendet werden.

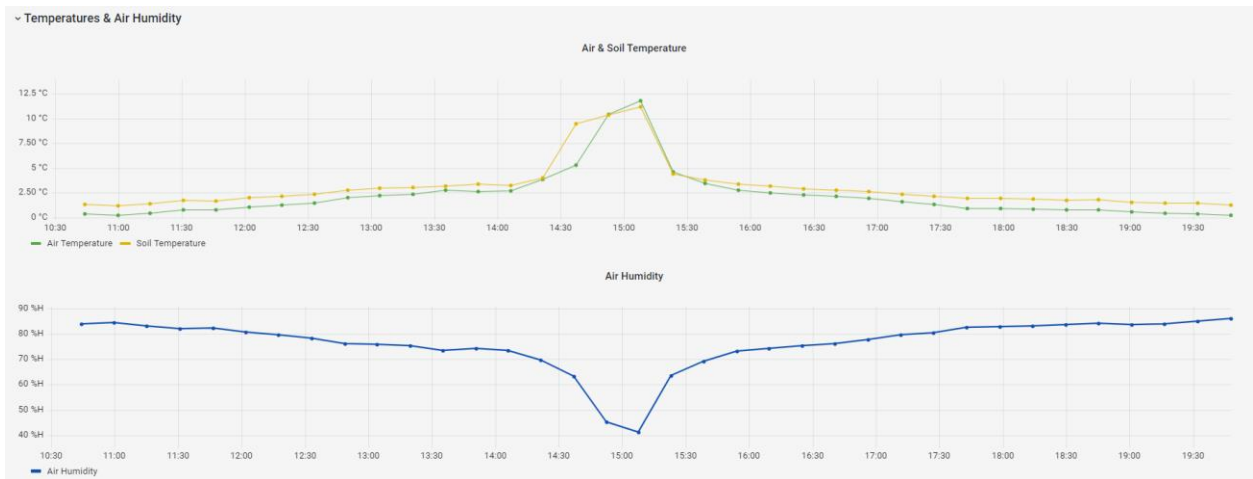


Abbildung 100 Grafana Visualisierung Sensordaten

Für die PAX Counter wurde ein separates Dashboard entwickelt. Es wird in Abbildung 101 gezeigt. Auch in diesem Dashboard kann die Variable der Node ID gewählt werden.

Der Standort wird in einer Karte angezeigt. Der aktuellste Messwert und die Summe der gezählten Geräte sind rechts davon ersichtlich. Der Empfangszeitpunkt der letzten Nachricht und der Zustand des MAC Filters werden daneben angezeigt. Über die Schaltfläche «Compare measurements» können Messwerte von zwei PAX Countern verglichen werden. Unterhalb ist der Verlauf der Akkuspannung ersichtlich.

Die PAX Werte sind in Form eines zoombaren Bar Charts visualisiert. Weiter unten in Dashboard befinden sich experimentelle Charts, mit denen die gemessenen Werte als Heatmap oder als Candle Chart visualisiert werden.

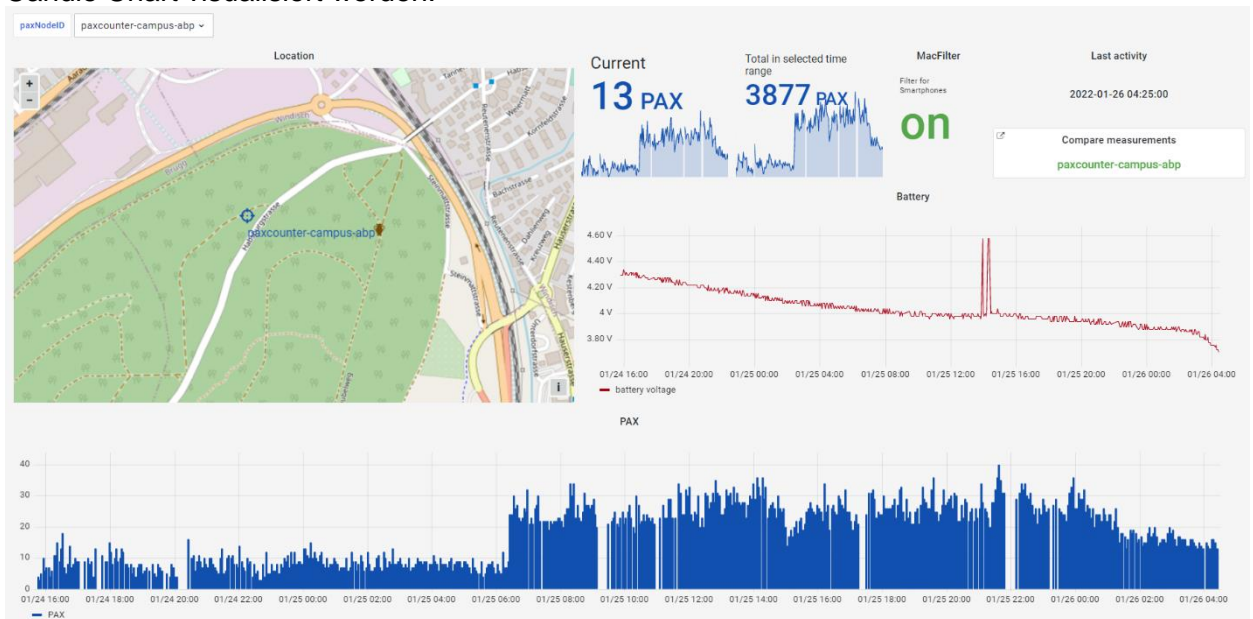


Abbildung 101 Grafana PAX Counter

Zum Vergleichen der Messwerte von zwei PAX Countern wurde ein entsprechendes Dashboard entwickelt. Abbildung 102 zeigt ein Screenshot. Über Variablen können zwei PAX Counter zum Vergleichen gewählt werden. In der Karte wird ein Kreis, dessen Grösse proportional zu den durchschnittlichen PAX Werten ist, angezeigt. Die Durchschnittswerte werden rechts davon angezeigt. Im Chart unter den Durchschnittswerten befindet sich ein Line Chart der PAX Messwerten.

Der Bar Chart im unteren Bereich visualisiert die Summe der erkannten Geräte pro PAX Counter in einem bestimmten Zeitraum. Der Time Bucket kann über eine Variable gewählt werden.

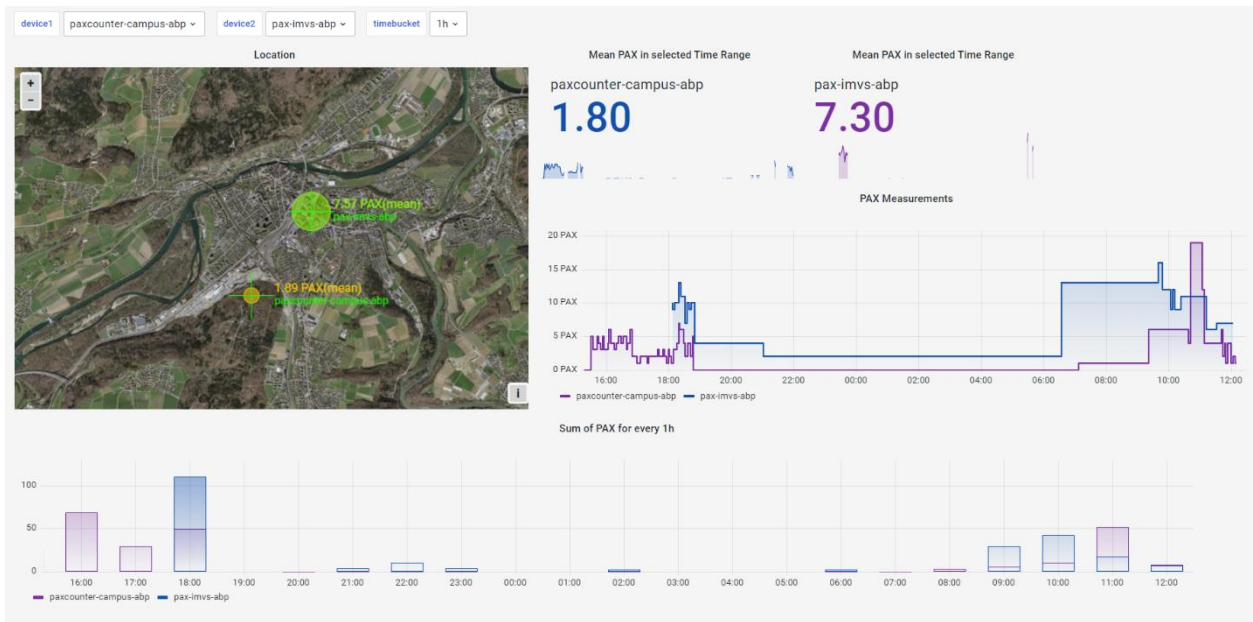


Abbildung 102 Grafana Vergleich von PAX Werten

5.5.5 Exportieren von Messwerten

Messwerte können aus Grafana im CSV Format heruntergeladen werden. In dem Dashboard mit der Bezeichnung Raw Data können Zeitbereich und Node ID's der zu exportierenden Messwerte ausgewählt werden. Durch Klicken auf die Schaltfläche «Inspect → Data» öffnet sich ein Menü für den Download der Messwerte in verschiedenen Formaten.

6 Fazit

Um auf Basis des Proof of Concept ein praxistaugliches System zu entwickeln, sind Optimierungen von Hardware und Software erforderlich. Im Bezug auf die Hardware muss verhindert werden, dass die Bestandteile durch Feuchtigkeit oder andere äussere Einwirkungen beschädigt werden. Um einen Einsatz durch Laien zu ermöglichen, muss der Betrieb und die Konfiguration des Systems ohne IT Fachwissen ermöglicht werden. Das Deployment der Software ist teilweise sehr aufwändig und muss vereinfacht werden.

Im Rahmen der Arbeit wurde die Hardware für die PoE Kameras und die Sensor Nodes neu entwickelt, um die Robustheit zu erhöhen und Beschädigungen zu verhindern. Die gewählten Komponente sind witterungsbeständig und schützen vor mechanischen Belastungen. Die Fertigung der Montageplatten sind mit dem 3D Druckverfahren möglich, womit ein Nachbau mit Zugang zu einem 3D Drucker sehr einfach ist.

Für die Konfiguration des Kamerasystems, der Sensor Nodes und der PAX Counter wurde jeweils eine benutzerfreundliche Lösung entwickelt, die ohne zusätzliche Software oder Konsolenzugriff und auch im Feld genutzt werden kann.

Das Deployment des Kamerasystems wurde soweit vereinfacht, dass pro Gerät nur noch eine Befehlseingabe nötig ist. Für das Deployment der Sensor Nodes wurde eine Lösung entwickelt, die keine zusätzliche Software erfordert. Das Hochladen der Software auf die PAX Counter ist nicht mehr abhängig von einer installierten Arduino IDE und den entsprechenden Libraries.

Zur Erweiterbarkeit des Kamerasystems wurde die Unterstützung von generischen USB Kameras implementiert. Der Einsatzbereich der Sensor Nodes wurde durch eine grössere Sensorauswahl erweitert. Für die Speicherung, Analyse und Visualisierung von Sensor- und PAX Daten wurde ein plattformunabhängiges Backend entwickelt. Die Backend Software kann mit wenigen Befehlen auf einem lokalen Rechner oder auf einer virtuellen Maschine installiert werden und ist einfach zu bedienen.

Insgesamt wurden die in der Aufgabenstellung festgelegten Ziele erreicht und das Projekt wurde erfolgreich abgeschlossen.

7 Literaturverzeichnis

- [1] E. Weber, "Was ist Biodiversität?," in *Biodiversität - Warum wir ohne Vielfalt nicht leben können*, E. Weber, Ed. Berlin, Heidelberg: Springer, 2018, pp. 1–16. doi: 10.1007/978-3-662-55624-5_1.
- [2] B. Unit, "Convention Text," Nov. 02, 2006. <https://www.cbd.int/convention/articles/?a=cbd-02> (accessed Oct. 11, 2021).
- [3] R. Wittig and M. Niekisch, "Was ist Biodiversität?," in *Biodiversität: Grundlagen, Gefährdung, Schutz*, R. Wittig and M. Niekisch, Eds. Berlin, Heidelberg: Springer, 2014, pp. 3–23. doi: 10.1007/978-3-642-54694-5_1.
- [4] E. Weber, "Persönlichkeiten bei Tier und Pflanze," in *Biodiversität - Warum wir ohne Vielfalt nicht leben können*, E. Weber, Ed. Berlin, Heidelberg: Springer, 2018, pp. 83–97. doi: 10.1007/978-3-662-55624-5_6.
- [5] E. Weber, "Von neu entdeckten Arten," in *Biodiversität - Warum wir ohne Vielfalt nicht leben können*, E. Weber, Ed. Berlin, Heidelberg: Springer, 2018, pp. 19–34. doi: 10.1007/978-3-662-55624-5_2.
- [6] E. Weber, "Lebensräume und Schicksalsgemeinschaften," in *Biodiversität - Warum wir ohne Vielfalt nicht leben können*, E. Weber, Ed. Berlin, Heidelberg: Springer, 2018, pp. 99–109. doi: 10.1007/978-3-662-55624-5_7.
- [7] W. Nentwig, S. Bacher, and R. Brandl, "Lebensgemeinschaften," in *Ökologie kompakt*, W. Nentwig, S. Bacher, and R. Brandl, Eds. Berlin, Heidelberg: Springer, 2017, pp. 173–225. doi: 10.1007/978-3-662-54352-8_4.
- [8] B. für U. B. | O. fédéral de l'environnement O. | U. federale dell'ambiente UFAM, "Biodiversitätsmonitoring Schweiz BDM." 2014. Accessed: Oct. 11, 2021. [Online]. Available: <https://www.bafu.admin.ch/bafu/de/home/themen/thema-biodiversitaet/biodiversitaet--publikationen/publikationen-biodiversitaet/biodiversitaetsmonitoring-schweiz-bdm.html>
- [9] "Verbreitungskarten Info Flora." <https://obs.infoflora.ch/app/atlasses/de/index.html> (accessed Oct. 15, 2021).
- [10] "Modul Vegetation - WSL." <https://biotopschutz.wsl.ch/de/modul-vegetation.html> (accessed Oct. 15, 2021).
- [11] Auftragnehmer Biodiversitäts-Monitoring Schweiz, "Anleitung für die Feldarbeit zum Indikator «Z7-Gefässpflanzen»." Bundesamt für Umwelt, Dec. 2020.
- [12] biodiversitymonitoring, "Programm." <https://www.biodiversitymonitoring.ch/index.php/de/programm> (accessed Oct. 15, 2021).
- [13] E. Weber, "Biomonitoring – klassisch und modern," in *Biodiversität - Warum wir ohne Vielfalt nicht leben können*, E. Weber, Ed. Berlin, Heidelberg: Springer, 2018, pp. 63–79. doi: 10.1007/978-3-662-55624-5_5.
- [14] "AMMOD Portal – Automated Multisensor Stations for Monitoring of BioDiversity." <https://ammod.de/> (accessed Oct. 15, 2021).
- [15] "Kamerafallen – AMMOD Portal." <https://ammod.de/kamerafallen/> (accessed Oct. 18, 2021).
- [16] "Akustiksensoren – AMMOD Portal." <https://ammod.de/akustiksensoren/> (accessed Oct. 18, 2021).
- [17] "Flüchtige organische Verbindungen (VOCs)," *Pflanzenforschung.de*. <https://www.pflanzenforschung.de/de/pflanzenwissen/lexikon-a-z/fluechtige-organische-verbindungen-vocs-10064> (accessed Oct. 18, 2021).
- [18] "Smellscapes – AMMOD Portal." <https://ammod.de/smellscapes/> (accessed Oct. 18, 2021).
- [19] "Sensordatenfusion – AMMOD Portal." <https://ammod.de/sensordatenfusion/> (accessed Oct. 18, 2021).
- [20] "German Barcode of Life – Reference Library of German Plants, Animals and Fungi." <https://bolgermany.de/home/> (accessed Oct. 18, 2021).
- [21] "MinION," *Oxford Nanopore Technologies*. <http://nanoporetech.com/products/minion> (accessed Oct. 18, 2021).
- [22] T. Amberg, *IoT Reference Model*. 2019. Accessed: Oct. 18, 2021. [Photo]. Available: <https://www.flickr.com/photos/tamberg/46404459874/>
- [23] A. @ AEQ-WEB, "LHT65 mit LoRaWAN & TTN verbinden," *AEQ-WEB*, Nov. 14, 2020. <https://www.aeq-web.com/> (accessed Oct. 19, 2021).

- [24]Elektronik-Kompendium, "Gateway." <https://www.elektronik-kompendium.de/sites/net/0901111.htm> (accessed Oct. 19, 2021).
- [25]Tréa Lavery, "DEFINITION IoT-Gateway," *ComputerWeekly.de*, Sep. 2021. <https://www.computerweekly.com/de/definition/IoT-Gateway> (accessed Oct. 19, 2021).
- [26]Traefik Labs, "Traefik," Sep. 23, 2020. <https://doc.traefik.io/traefik/> (accessed Oct. 28, 2021).
- [27]P. Bühler, P. Schlaich, and D. Sinner, "Datenbanken," in *Datenmanagement: Daten – Datenbanken – Datensicherheit*, P. Bühler, P. Schlaich, and D. Sinner, Eds. Berlin, Heidelberg: Springer, 2019, pp. 50–79. doi: 10.1007/978-3-662-55507-1_3.
- [28]Telegraf. InfluxData, 2021. Accessed: Nov. 01, 2021. [Online]. Available: <https://github.com/influxdata/telegraf>
- [29]"Grafana: The open observability platform," *Grafana Labs*. <https://grafana.com/> (accessed Sep. 21, 2021).
- [30]H. Nielsen *et al.*, "Hypertext Transfer Protocol – HTTP/1.1," Internet Engineering Task Force, Request for Comments RFC 2616, Jun. 1999. doi: 10.17487/RFC2616.
- [31]T. H. Team, "Quality of Service 0,1 & 2 - MQTT Essentials: Part 6." <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/> (accessed Nov. 01, 2021).
- [32]F. Raschbichler, "MQTT 5: How the new User Properties feature works and why we love it." <https://www.hivemq.com/blog/mqtt5-essentials-part6-user-properties/> (accessed Nov. 01, 2021).
- [33]R. Herrero, "Application Layer," in *Fundamentals of IoT Communication Technologies*, R. Herrero, Ed. Cham: Springer International Publishing, 2022, pp. 111–149. doi: 10.1007/978-3-030-70080-5_5.
- [34]"ECMA-404," *Ecma International*. <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/> (accessed Nov. 02, 2021).
- [35]*Protocol Buffers - Google's data interchange format*. Protocol Buffers, 2021. Accessed: Nov. 08, 2021. [Online]. Available: <https://github.com/protocolbuffers/protobuf>
- [36]Sharon Lin, "FPGAs, SoCs, Microcontrollers— A Quick Rundown of IoT Devices | Hacker Noon," Dec. 26, 2018. <https://hackernoon.com/fpgas-socs-microcontrollers-a-quick-rundown-of-iot-devices-c5a25c7290c6> (accessed Nov. 09, 2021).
- [37]H. Bähring, "Mikrocontroller," in *Anwendungsorientierte Mikroprozessoren: Mikrocontroller und Digitale Signalprozessoren*, H. Bähring, Ed. Berlin, Heidelberg: Springer, 2010, pp. 11–45. doi: 10.1007/978-3-642-12292-7_2.
- [38]K. Wüst, "Mikrocontroller," in *Mikroprozessortechnik: Grundlagen, Architekturen, Schaltungstechnik und Betrieb von Mikroprozessoren und Mikrocontrollern*, K. Wüst, Ed. Wiesbaden: Vieweg+Teubner, 2011, pp. 258–301. doi: 10.1007/978-3-8348-9881-4_14.
- [39]E. White, *Making Embedded Systems: Design Patterns for Great Software*, 1st ed. Beijing Köln: O'Reilly and Associates, 2011.
- [40]Arduino, "attachInterrupt() - Arduino Reference," *attachInterrupt() - Arduino Reference*, 2021. <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/> (accessed Nov. 12, 2021).
- [41]Espressif, "ESP32 Datasheet." Espressif, Oct. 15, 2021. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [42]reichelt elektronik GmbH, "RASP PI 4 B 8GB - Raspberry Pi 4 B, 4x 1,5 GHz, 8 GB RAM, WLAN, BT," *Elektronik und Technik bei reichelt elektronik günstig bestellen*. <https://www.reichelt.com/dk/de/raspberry-pi-4-b-4x-1-5-ghz-8-gb-ram-wlan-bt-rasp-pi-4-b-8gb-p276923.html> (accessed Nov. 12, 2021).
- [43]Raspberry Pi, "Raspberry Pi Documentation - Raspberry Pi OS," 2021. <https://www.raspberrypi.com/documentation/computers/os.html> (accessed Nov. 12, 2021).
- [44]Elektronik-Kompendium, "IEEE 802.11 / WLAN-Grundlagen," *IEEE 802.11 / WLAN-Grundlagen*, 2021. <https://www.elektronik-kompendium.de/sites/net/0610051.htm> (accessed Nov. 17, 2021).
- [45]"Bluetooth Technologieübersicht | Bluetooth® Technologie Website," *Website zur Bluetooth®-Technologie*. <https://www.bluetooth.com/de/learn-about-bluetooth/tech-overview/> (accessed Nov. 22, 2021).

- [46] Semtech Developer Portal, "LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL," *LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL*. <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/> (accessed Nov. 18, 2021).
- [47] Richard Wenner, *LoRa CHIRP*, (Nov. 17, 2017). Accessed: Nov. 18, 2021. [Online]. Available: <https://www.youtube.com/watch?v=dxYY097QNs0>
- [48] The Things Network, "Spreading Factors," *The Things Network*. <https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/> (accessed Nov. 19, 2021).
- [49] "Duty Cycle," *The Things Network*. <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/> (accessed Nov. 19, 2021).
- [50] "LoRaWAN Architecture," *The Things Network*. <https://www.thethingsnetwork.org/docs/lorawan/architecture/> (accessed Nov. 19, 2021).
- [51] B. Ray, "SigFox Vs. LoRa: A Comparison Between Technologies & Business Models | Blog | Link Labs," 2018. <https://www.link-labs.com/blog/sigfox-vs-lora> (accessed Nov. 19, 2021).
- [52] Sara Landström, "NB-IoT: a sustainable technology for connecting billions of devices - Ericsson Technology Review," Apr. 25, 2016. <https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/nb-iot-a-sustainable-technology-for-connecting-billions-of-devices> (accessed Nov. 19, 2021).
- [53] A. pmk, *Frequency reuse*. 2007. Accessed: Nov. 19, 2021. [Online]. Available: https://commons.wikimedia.org/wiki/File:Frequency_reuse.svg
- [54] "What are the differences between 1G, 2G, 3G, 4G, and 5G? Is 5G that innovative? | Widgeti," Sep. 30, 2020. <https://widgeti.tech/5g/what-are-the-differences-between-1g-2g-3g-4g-and-5g-is-5g-that-innovative/> (accessed Nov. 22, 2021).
- [55] E. K. ComCom, "Mobilfunkabdeckung," 2020. <https://www.comcom.admin.ch/comcom/de/home/dokumentation/zahlen-und-fakten/mobilfunkmarkt/mobilfunk-abdeckung.html> (accessed Nov. 19, 2021).
- [56] B. für K. BAKOM, "Breitbandatlas," 2021. <https://www.bakom.admin.ch/bakom/de/home/telekommunikation/atlas.html> (accessed Nov. 22, 2021).
- [57] S. Cheruvu, A. Kumar, N. Smith, and D. M. Wheeler, "Connectivity Technologies for IoT," in *Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment*, S. Cheruvu, A. Kumar, N. Smith, and D. M. Wheeler, Eds. Berkeley, CA: Apress, 2020, pp. 347–411. doi: 10.1007/978-1-4842-2896-8_5.
- [58] Maxim Integrated, "1-Wire Protocol PDF of DS18S20 vs DS18B20 Digital Thermometers | Maxim Integrated," 2009. <https://www.maximintegrated.com/en/design/technical-documents/app-notes/4/4377.html> (accessed Nov. 22, 2021).
- [59] Components101, "DS18B20 Temperature Sensor," *Components101*, May 07, 2018. <https://components101.com/sensors/ds18b20-temperature-sensor> (accessed Nov. 22, 2021).
- [60] Maxim Integrated, "DS18B20 - Programmable Resolution 1-Wire Digital Thermometer." Jul. 2019.
- [61] Omega, "Working principle of thermocouples," <https://www.omega.com/en-us/>, Mar. 05, 2019. <https://www.omega.com/en-us/resources/how-thermocouples-work> (accessed Nov. 22, 2021).
- [62] Seeed Studio, "Grove - Temperature & Humidity Sensor (DHT11)," 2021. <https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DHT11.html> (accessed Nov. 23, 2021).
- [63] Seeed Studio, "Grove - Soil Moisture Sensor," 2021. <https://www.seeedstudio.com/Grove-Moisture-Sensor.html> (accessed Nov. 23, 2021).
- [64] Seeed Studio, "Grove - Capacitive Soil Moisture Sensor (Corrosion Resistant)," 2021. <https://www.seeedstudio.com/Grove-Capacitive-Moisture-Sensor-Corrosion-Resistant.html> (accessed Nov. 23, 2021).
- [65] Components101, "Introduction to Gas Sensors: Construction Types and Working," *Components101*, Jun. 04, 2020. <https://components101.com/articles/introduction-to-gas-sensors-types-working-and-applications> (accessed Nov. 23, 2021).
- [66] Nova Fitness Co, "Laser PM2.5 Sensor specification." Oct. 09, 2015.

- [67]Edinburgh Sensors, "TOC Analyser | Total Organic Carbon Analysis | Gas Detection / Monitoring," *Edinburgh Sensors*, May 14, 2018. <https://edinburghsensors.com/industries/total-organic-carbon-toc/> (accessed Nov. 23, 2021).
- [68]Admin, "TDS Sensor & Arduino Interfacing for Water Quality Monitoring," *How To Electronics*, Mar. 04, 2020. <https://how2electronics.com/tds-sensor-arduino-interfacing-water-quality-monitoring/> (accessed Nov. 23, 2021).
- [69]Seeed Studio, "Grove - TDS Sensor/Meter For Water Quality (Total Dissolved Solids)," 2019. <https://www.seeedstudio.com/Grove-TDS-Sensor-p-4400.html> (accessed Nov. 23, 2021).
- [70]Seeed Studio, "Grove - Turbidity Sensor (Meter) for Arduino V1.0." <https://www.seeedstudio.com/Grove-Turbidity-Sensor-p-4399.html> (accessed Nov. 23, 2021).
- [71]"Grove - PH Sensor Kit (E-201C-Blue)." <https://www.seeedstudio.com/Grove-PH-Sensor-Kit-E-201C-Blue-p-4577.html> (accessed Nov. 23, 2021).
- [72]Adafruit, "Photocells," *Adafruit Learning System*, Nov. 23, 2021. <https://learn.adafruit.com/photocells/using-a-photocell> (accessed Nov. 24, 2021).
- [73]Seeed Studio, "Grove - Sunlight sensor (UV-light, visible light and infrared light) - SI1145." <https://www.seeedstudio.com/Grove-Sunlight-Sensor.html> (accessed Nov. 24, 2021).
- [74]Seeed Studio, "Seeed Documentation - Grove-GPS," Nov. 16, 2021. <https://github.com/SeeedDocument/Seeed-WiKi/blob/0f8c056c408a84c6423d77fef229d6efdd6c9994/docs/Grove-GPS.md> (accessed Nov. 22, 2021).
- [75]Pi-Shop, "Original Raspberry Pi Kamera Module V2," *Pi-Shop.ch*. <https://www.pi-shop.ch/raspberry-pi-kamera-module-v2> (accessed Nov. 24, 2021).
- [76]Pi-Shop, "FlightAware Pro Stick Plus (USB SDR ADS-B Receiver)," *Pi-Shop.ch*. <https://www.pi-shop.ch/flightaware-pro-stick-usb-sdr-ads-b-receiver> (accessed Nov. 24, 2021).
- [77]K. Magdy, "What Are Different Types Of Sensors, Classification, Their Applications?," *Deep-Blue*, Aug. 01, 2020. <https://deepbluembedded.com/different-types-sensors-applications/> (accessed Nov. 22, 2021).
- [78]*Mitwelten.org IoT Hardware Proof of Concept*. Mitwelten, 2021. Accessed: Sep. 21, 2021. [Online]. Available: <https://github.com/mitwelten/mitwelten-iot-hardware-poc>
- [79]jacksonliam, *mjpg-streamer*. 2021. Accessed: Nov. 29, 2021. [Online]. Available: <https://github.com/jacksonliam/mjpg-streamer>
- [80]T. Amberg, *PXL_20210621_130506099*. 2021. Accessed: Nov. 29, 2021. [Photo]. Available: <https://www.flickr.com/photos/tamberg/51262937777/>
- [81]T. Amberg, *PXL_20210425_193553976*. 2021. Accessed: Nov. 29, 2021. [Photo]. Available: <https://www.flickr.com/photos/tamberg/51139361764/>
- [82]wullt, *capture.py*. Mitwelten, 2021. Accessed: Nov. 29, 2021. [Online]. Available: <https://github.com/mitwelten/mitwelten-iot-hardware-poc/blob/e072957e8e4a11e0041430d6f3eec4204181b0d1/RaspberryPi/APGateway/Capture/capture.py>
- [83]wullt, *ImagePreview*. Mitwelten, 2021. Accessed: Nov. 29, 2021. [Online]. Available: <https://github.com/mitwelten/mitwelten-iot-hardware-poc>
- [84]T. Amberg, *PXL_20210621_125940152*. 2021. Accessed: Nov. 29, 2021. [Photo]. Available: <https://www.flickr.com/photos/tamberg/51264412624/>
- [85]Thomas Amberg, "PXL_20210210_112232092.PORTRAIT," *Flickr*. <https://www.flickr.com/photos/tamberg/50928448188/> (accessed Nov. 30, 2021).
- [86]T. Amberg, *PXL_20211102_124713638*. 2021. Accessed: Nov. 29, 2021. [Photo]. Available: <https://www.flickr.com/photos/tamberg/51650385070/>
- [87]"ESP32-Paxcounter/src at master · cyberman54/ESP32-Paxcounter," *GitHub*. <https://github.com/cyberman54/ESP32-Paxcounter> (accessed Sep. 21, 2021).
- [88]Yaler, "Yaler.net - access devices from the Web," *Yaler.net Access devices from the Web*. <https://yaler.net/> (accessed Dec. 07, 2021).
- [89]Distrelec, "LCM 12 G | Hummel Kabelverschraubung, 3 ... 6.5mm, M12, Polyamid, Grau," *Distrelec Schweiz*. <https://www.distrelec.ch/de/kabelverschraubung-5mm-m12-polyamid-grau-hummel-lcm-12/p/15501218> (accessed Dec. 06, 2021).

- [90] R. P. Ltd, "Raspberry Pi OS," *Raspberry Pi*, Jan. 2022. <https://www.raspberrypi.com/software/> (accessed Jan. 24, 2022).
- [91] Sandra Boner, "Lufttemperatur richtig messen - An der Hauswand wärmer als in der freien Luft," *Schweizer Radio und Fernsehen (SRF)*, Mar. 28, 2021. <https://www.srf.ch/meteo/meteo-stories/lufttemperatur-richtig-messen-an-der-hauswand-waermer-als-in-der-freien-luft> (accessed Nov. 30, 2021).
- [92] DF Robot, "SHT31 Weather-proof Temperature & Humidity Sensor - DFRobot." <https://www.dfrobot.com/product-2160.html> (accessed Dec. 06, 2021).
- [93] Meteotest, "Bodenmessnetz," *Bodenmessnetz*. <https://bodenmessnetz.ch/hintergrund/technik> (accessed Dec. 06, 2021).
- [94] *USB Flashing Format (UF2)*. Microsoft, 2022. Accessed: Jan. 24, 2022. [Online]. Available: <https://github.com/microsoft/uf2>
- [95] *Adafruit TinyUSB Library for Arduino*. Adafruit Industries, 2022. Accessed: Jan. 24, 2022. [Online]. Available: https://github.com/adafruit/Adafruit_TinyUSB_Arduino
- [96] *esptool.py*. Espressif Systems, 2022. Accessed: Jan. 24, 2022. [Online]. Available: <https://github.com/espressif/esptool>
- [97] WullIT, *PoECam*. 2022. Accessed: Jan. 28, 2022. [Online]. Available: <https://github.com/WullIT/PoECam>
- [98] WullIT, *APGateway*. 2022. Accessed: Jan. 28, 2022. [Online]. Available: <https://github.com/WullIT/APGateway>
- [99] Distrelec, "4U63131206017 | BOX4U Kunststoffgehäuse 115.4x125.3x58.05mm Grau ABS IP65," *Distrelec Schweiz*, 2022. <https://www.distrelec.ch/de/kunststoffgehaeuse-115-4x125-3x58-05mm-grau-abs-ip65-box4u-4u63131206017/p/30219152> (accessed Jan. 25, 2022).
- [100] Distrelec, "WIB 2 | WISKA LTD Abzweigdose, 120x160x70mm, Thermoplast," *Distrelec Schweiz*, 2022. <https://www.distrelec.ch/de/abzweigdose-120x160x70mm-thermoplast-wiska-ltd-wib/p/30158504> (accessed Jan. 25, 2022).
- [101] Microchip Technology, "SAM D5x/E5x Family Data Sheet." Jan. 2021.
- [102] Aliexpress, "5V 1200mA 6Watt Battery Charger USB port DC 5.5*2.1 Charge Regulators Solar Panel 6W Outdoor Power Li ion Batteries Portable|Solar Cells| - AliExpress," *aliexpress.com*, Jan. 2022. https://www.aliexpress.com/item/4000969476579.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp= (accessed Jan. 25, 2022).
- [103] WullIT, *MultisensorLoRaNode*. 2022. Accessed: Jan. 26, 2022. [Online]. Available: <https://github.com/WullIT/MultisensorLoRaNode>
- [104] *uf2conv.py*. Microsoft, 2022. Accessed: Jan. 26, 2022. [Online]. Available: <https://github.com/microsoft/uf2/blob/f0506988eaf24cc12646ed1a39799fa29b204eed/utils/uf2conv.py>
- [105] Seeed Studio, "Grove - Button," 2021. <https://www.seeedstudio.com/Grove-Button.html> (accessed Jan. 27, 2022).
- [106] WullIT, *ESP32LoRaMqttPaxCounter*. 2022. Accessed: Jan. 26, 2022. [Online]. Available: <https://github.com/WullIT/ESP32LoRaMqttPaxCounter>
- [107] Influxdata, "InfluxDB," *InfluxData*, 2022. <https://www.influxdata.com/products/influxdb/> (accessed Jan. 27, 2022).
- [108] Influxdata, "Telegraf Open Source Server Agent | InfluxDB," *InfluxData*, 2022. <https://www.influxdata.com/time-series-platform/telegraf/> (accessed Jan. 27, 2022).
- [109] J. Baker, *GJSON Path Syntax*. 2022. Accessed: Jan. 27, 2022. [Online]. Available: <https://github.com/tidwall/gjson/blob/master/SYNTAX.md>
- [110] WullIT, *TGIBackend*. 2022. Accessed: Jan. 27, 2022. [Online]. Available: <https://github.com/WullIT/TGIBackend>

8 Abbildungsverzeichnis

Abbildung 1 Ebenen und Bestandteile der Biodiversität [3]	7
Abbildung 2 Biodiversität bestimmt durch Anzahl Arten, Individuen und Verteilung der Individuen	8
Abbildung 3 Funktion in einer Lebensgemeinschaft in Anhängigkeit der Artenzahl [7]	11
Abbildung 4 Skala der Indikatoren Z3, Z7 und Z9 des BDM [9] [10]	12
Abbildung 5 Beispiel Transekt Z7 Gefässpflanzen [8]	13
Abbildung 6 Beispiel Erhebungsfläche Z9 [8]	14
Abbildung 7 Messnetz BDM Artenvielfalt Z7 [12]	14
Abbildung 8 Messnetz BDM Artenvielfalt Z9 [12]	15
Abbildung 9 Messnetz BDM Artenvielfalt Z9 in Fließgewässern [12]	15
Abbildung 10 Zeitliche Staffelung der Erhebungen des BDM [8]	16
Abbildung 11 AMMOD: Tiefendaten als Zusatzinformation [15]	16
Abbildung 12 AMMOD: Erkennung der Anzahl Individuen mit gerichteten Mikrofonen [16]	17
Abbildung 13 AMMOD Malaisefalle	18
Abbildung 14 Oxford Nanopore MinION [21]	18
Abbildung 15 IoT Referenzmodell [22]	19
Abbildung 16 IoT Device Beispiel: LHT65 [23]	20
Abbildung 17 IoT Topologien Cloud, Edge und Fog	21
Abbildung 18 Bestandteile einfaches Backend	22
Abbildung 19 Visualisierung Weiterleitung Reverse Proxy	22
Abbildung 20 Bestandteile Reverse Proxy	23
Abbildung 21 Datenstruktur verschiedener Datenbanktypen	23
Abbildung 22 HTTP Request Beispiel	25
Abbildung 23 MQTT Kommunikation Beispiel	26
Abbildung 24 MQTT QoS	26
Abbildung 25 ATMEGA 8 Bit Mikrocontroller [36]	30
Abbildung 26 State Machine [39]	32
Abbildung 27 ESP32 Functional Block Diagram [41]	33
Abbildung 28 Raspberry Pi 4 [42]	34
Abbildung 29 Wireless Technologien im Vergleich	35
Abbildung 30 BLE Read / Write / Notify	36
Abbildung 31 CSS Modulation [46] [47]	37
Abbildung 32 LoRa Spreading Factors [48]	38
Abbildung 33 TTN LoRaWAN Architecture [50]	39
Abbildung 34 NB-IoT spectrum options [52]	40
Abbildung 35 Cellular network cells [53]	40
Abbildung 36 Abdeckung 5G in der Schweiz [56]	41
Abbildung 37 PoE Systemaufbau	41
Abbildung 38 DS18B20 1-Wire Temperature Sensor [59]	42
Abbildung 39 DHT11 Temperature- and Humidity Sensor [62]	43
Abbildung 40 Resistive Soil Moisture Sensor [63]	43
Abbildung 41 Capacitive Soil Moisture Sensor [64]	44
Abbildung 42 MQ-6 Gas Sensor [65]	44
Abbildung 43 SDS011 PM Sensor [66]	46
Abbildung 44 Turbidity Sensor [70]	46
Abbildung 45 PH Sensor Kit [71]	47
Abbildung 46 Sunlight Sensor [73]	47
Abbildung 47 Grove GPS Modul [74]	48
Abbildung 48 Raspberry Pi Camera Module v2 [75]	48
Abbildung 49 Übersicht Kamerasystem	50
Abbildung 50 PoE Cam (l) and AP Gateway (r) [78]	50
Abbildung 51 PoE Camera [78]	51
Abbildung 52 Blockdiagramm AP Gateway	51
Abbildung 53 AP Gateway offen und geschlossen [80] [81]	52
Abbildung 54 Screenshot ImagePreview Gui	53

Abbildung 55 AP Gateway eingekleidet [84]	53
Abbildung 56 Blockdiagram LoRa Nodes	54
Abbildung 57 Case for LoRa Nodes [85]	54
Abbildung 58 LoRa Umweltsensor [86]	55
Abbildung 59 PAX Counter Measurements 7 days	56
Abbildung 60 Korrodierte Elektronik eines PAX Counter	57
Abbildung 61 Overview Mitwelten Backend	57
Abbildung 62 TTN Monitoring VM Overview	58
Abbildung 63 Kabelverschraubung [89]	59
Abbildung 64 SHT31 Weatherproof Temperature & Humidity Sensor [92]	62
Abbildung 65 PoE Kamera Explosionsansicht	65
Abbildung 66 PoE Kamera im offenen Gehäuse	66
Abbildung 67 PoE Kamera Halterung mit Metallwinkeln	66
Abbildung 68 PoE Kamera mit über USB angeschlossener Endoskop Kamera	67
Abbildung 69 AP Gateway User Interface Übersicht	69
Abbildung 70 AP Gateway User Interface Kameraeinstellungen	69
Abbildung 71 AP Gateway User Interface Hinzufügen von Kameras	70
Abbildung 72 AP Gateway User Interface Globale Einstellungen	70
Abbildung 73 ABS Gehäuse BOX4U [99]	71
Abbildung 74 Sensor Node Anordnung der Bauteile in BOX4U Gehäuse	72
Abbildung 75 Sensor Node Seitenansicht Montageplatte für BOX4U Gehäuse	72
Abbildung 76 Sensor Node in BOX4U Gehäuse	73
Abbildung 77 WISKA Gehäuse [100]	73
Abbildung 78 Sensor Node Anordnung der Bauteile in WISKA Gehäuse	74
Abbildung 79 Sensor Node Seitenansicht Montageplatte für WISKA Gehäuse	74
Abbildung 80 Sensor Node in Wiska Gehäuse	75
Abbildung 81 Elektroschema Sensor Node	75
Abbildung 82 Schema Energiemanagement Sensoren Feather M4	76
Abbildung 83 Solar Panel Vergleich Akkuladung	77
Abbildung 84 Sensor Node Flowchart Setup	78
Abbildung 85 Sensor Node Flowchart Main	79
Abbildung 86 Sensor Node Konfigurationsgenerator Connectivity	80
Abbildung 87 Sensor Node Konfigurationsgenerator Sensorauswahl	81
Abbildung 88 Sensor Node Konfigurationsgenerator Ausgabe	81
Abbildung 89 Varianten der Payload Strukturierung	82
Abbildung 90 PAX Counter in RND Gehäuse mit angeschlossenen Solar Panel	83
Abbildung 91 Elektroschema PAX Counter	84
Abbildung 92 PAX Counter Zeitparameter Messvorgang	85
Abbildung 93 PAX Counter Flowchart Setup	85
Abbildung 94 PAX Counter Flowchart Main Loop	86
Abbildung 95 PAX Counter Konfiguration Messung	87
Abbildung 96 PAX Counter Konfiguration Connectivity	88
Abbildung 97 PAX Counter Konfiguration Advanced	88
Abbildung 98 Grafana Dashboard Übersicht	92
Abbildung 99 Grafana Dashboard Sensor Node	92
Abbildung 100 Grafana Visualisierung Sensordaten	93
Abbildung 101 Grafana PAX Counter	93
Abbildung 102 Grafana Vergleich von PAX Werten	94

9 Tabellenverzeichnis

Tabelle 1 Rechenbeispiel α -, β - und γ -Diversität	9
Tabelle 2 Zustandsindikatoren BDM zur Beschreibung der Biodiversität [8]	12
Tabelle 3 Einflussindikatoren BDM [8]	13
Tabelle 4 Vergleich aktueller WLAN Standards [44]	35
Tabelle 5 Bluetooth Energieklassen	36
Tabelle 6 Vergleich 3G, 4G und 5G [54]	40
Tabelle 7 List of MQ Gas Sensors [65]	45
Tabelle 8 Kriterien bei der Auswahl eines Sensors [77]	49
Tabelle 9 Unterstützte Sensoren für Sensor Node	77

10 Formelverzeichnis

Formel 1 Artenreichtum R	9
Formel 2 Jaccard-Koeffizient	10
Formel 3 Sørensen-Koeffizient	10
Formel 4 Euklidische Distanz	10
Formel 5 Species turn-over	10
Formel 6 Shannon-Weaver-Index	10
Formel 7 Evenness	10

11 Ehrlichkeitserklärung

Hiermit bestätige ich, dass die eingereichte Projektarbeit mit dem Titel «Data Acquisition for Urban Biodiversity Monitoring» das Resultat meiner persönlichen Erarbeitung der Themen ist. Ich habe keine anderen als die angegebenen Quellen benutzt.

Ort, Datum Windisch, 29.01.2022

Unterschrift



12 Anhang

Der Anhang beinhaltet

- Den entwickelten Code
- Anleitungen zum Deployment in Form von Readme Dateien
- 3D Daten
- Elektroschemas