



HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES

Hochschule Ruhr West
Gesundheit und Medizintechnologien

Bachelorarbeit zum Thema:

**Datenverarbeitung für ein Zustandsmonitoring von Medizingeräten
mittels energiesparender Funk-Sensoren**

Bachelor of Science (B.Sc.)

Vorgelegt von:

Tagne Reine Nahomie Sigha

Wüstenhöfer Str. 123, 45355 Essen

E-Mail: reine.tagne@stud.hs-ruhrwest.de

Matrikelnummer: 10015774

Fachsemester: 7

Studiengang: Gesundheit und Medizintechnologien

Abgabe: 12.10.2023

Erstgutachter: Prof. Dr. Ing Andreas Hennig


Zweitgutachter: Prof. Dr. Ing Dirk Rüter

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Essen, 14.09.2023

Ort, Datum

A handwritten signature in black ink, consisting of stylized, overlapping letters that appear to be 'R' and 'M'.

Unterschrift

Abstract

In der modernen Medizintechnik ist die zuverlässige Funktionsweise von Geräten von entscheidender Bedeutung. Diese Bachelorarbeit befasst sich mit dem Zustandsmonitoring von medizinischen Geräten, um die verschiedenen Zustände von medizinischen Geräte zu erfassen.

Das Monitoring hilft dabei, die tatsächliche Nutzungsfrequenz der Geräte zu ermitteln. Ein verbessertes Verständnis der Gerätenutzung kann dazu beitragen, Kosten zu senken und das Gerätemanagement in Krankenhäusern zu optimieren.

Das primäre Ziel dieser Arbeit war die Entwicklung einer Datenverarbeitungsmethode für das Zustandsmonitoring von Medizingeräten mittels energiesparender Funk-Sensoren. Oftmals werden in Krankenhäusern mehr Geräte angeschafft als benötigt, und dennoch müssen alle Geräte gewartet werden, selbst wenn einige selten in Betrieb sind. Dies resultiert in unnötigen Kosten und Ausfallzeiten.

Für die Untersuchung wurde ein energiesparender Funksensor eingesetzt, der mit einem Stromsensor ausgestattet ist. Dieser erfasst den Stromfluss der angeschlossenen Geräte und kommuniziert online über das IoT. Die erfassten Daten wurden komprimiert und an das TTN (The Things Network) übertragen. Ein KI-Modell wurde mit den gesammelten Daten trainiert, um die verschiedenen Zustände der Geräte zu erkennen.

Die wichtigsten Ergebnisse zeigen, dass die Zustände der Geräte variieren und typischerweise in "AN", "AUS", "Idle-Modus" und "Im Betrieb" kategorisiert werden können. Sobald der Stromfluss einen bestimmten Wert erreicht, kann die KI bestimmen, in welchem Zustand sich das Gerät befindet. Dies bietet eine wertvolle Grundlage für ein verbessertes Management und Wartungssystem in Krankenhäusern.

Inhaltsverzeichnis

| | |
|--|-----|
| Eidesstattliche Erklärung..... | I |
| Abstract | II |
| Inhaltsverzeichnis..... | III |
| Tabelle Verzeichnis..... | IV |
| Abbildungsverzeichnis | IV |
| 1 Einleitung und Motivation..... | 1 |
| 1.1 Motivation | 1 |
| 1.2 Ziel der Arbeit | 1 |
| 2 Grundlagen | 2 |
| 2.1 Gerätemanagement im Praxiseinrichtungen..... | 2 |
| 2.1.1 Was ist Zustandsmonitoring?..... | 3 |
| 2.2 Internet of Things (IOT)..... | 4 |
| 2.3 LoRaWAN..... | 4 |
| 2.4 LoRa | 7 |
| 2.5 The Things Network..... | 8 |
| 2.5.1 Encoder/Decoder in TTN | 8 |
| 2.6 Maschine Learning Model..... | 9 |
| 2.6.1 Random Forest Algorithmus..... | 9 |
| 3 Anforderungen für das Zustandsmonitoring von medizinischen Geräten | 10 |
| 3.1 Anwendungsszenario Anforderung | 10 |
| 3.2 Grundlegende Anforderungen an das Monitoring-System..... | 11 |
| 3.3 Anforderungen an den energiesparenden Funksensor:..... | 11 |
| 3.3.1 Anforderung der Funksensor | 12 |
| 3.3.2 Datenminimierung und Übertragung..... | 12 |
| 3.3.3 Leistungseffizienter Algorithmus | 12 |
| 4 Lösungskonzept..... | 13 |
| 4.1 Anwendungsfall des Gesamtprojekts | 13 |
| 4.1.1 Aufnahme der Daten von verschiedenen Geräten | 14 |
| 4.2 Benutzeroberfläche (User Interface) | 16 |
| 4.3 Effiziente Datenübertragung durch Reduzierung der Datenmenge..... | 17 |
| 4.4 Einsatz eines KI-Modells für Zustandsmonitoring..... | 18 |
| 4.5 Anlernphase der KI-Model..... | 18 |
| 4.6 Energieeffizienz der Funksensor | 18 |
| 5 Ausarbeitung und Realisierung der Lösung | 20 |
| 5.1 Verwendete Komponenten..... | 20 |
| 5.2 Aufbau der Funksensor | 22 |

| | | |
|-------|--|----|
| 5.3 | Umsetzung der Funck Sensor bzw. Projekt..... | 23 |
| 5.1 | Verbindung an The Things Network (TTN)..... | 24 |
| 5.1.1 | Verbindung der LoRa Modul mit der TTN..... | 24 |
| 5.1.2 | Verbindung des Gateways mit der TTN..... | 25 |
| 5.1 | Erstellung der Machine-Learning-Model (ML-Model)..... | 29 |
| 5.2 | Decodern der Hexadezimal Daten in TTN..... | 30 |
| 5.2.1 | Zustände von Geräten..... | 30 |
| 5.3 | WebApp..... | 31 |
| 5.4 | Ablauf der Übertragung, Verarbeitung und Darstellung der Stromdaten..... | 32 |
| 6 | Validierung und Test..... | 34 |
| 6.1.1 | Bewertung der ML-Model..... | 34 |
| 7 | Fazit und Ausblick..... | 35 |
| | Literaturverzeichnis..... | 36 |
| | Anhang..... | 38 |

Tabelle Verzeichnis

| | | |
|------------|--|----|
| Tabelle 1 | Verwendete Komponenten..... | 11 |
| Tabelle 2: | Zustand und Farbe Zuordnung..... | 16 |
| Tabelle 3: | Grove-Wio-E5 Spezifikation und Parameters..... | 21 |
| Tabelle 4: | Elektronische Eigenschaften der Stroms Sensor..... | 22 |
| Tabelle 5: | Verbindung der Strom Sensor mit der Mikrokontroller..... | 22 |
| Tabelle 6: | Verbindung der LoRa Modul zu dem Board..... | 22 |
| Tabelle 7 | Zustand und Status von Geräten Zuweisung..... | 30 |
| Tabelle 8: | Genauigkeit nach dem Training der ML-Model..... | 34 |
| Tabelle 9 | Entwicklungsumgebung..... | 38 |

Abbildungsverzeichnis

| | | |
|----------------|--|----|
| Abbildung 1: | Anzahl von Medizinische Geräte in Universitätsmedizinzentren in Deutschland..... | 3 |
| Abbildung 2: | Verschiedene drahtlosen Kommunikationstechnologien..... | 5 |
| Abbildung 3: | LoRaWAN-Netzwerkinfrastruktur..... | 6 |
| Abbildung 4: | Authentifizierung von Geräten OTAA..... | 7 |
| Abbildung 5: | LoRaWAN Kommunikation Ablauf Diagramm..... | 13 |
| Abbildung 6: | Workflow Datenaufnahme..... | 14 |
| Abbildung 7: | Strom Fluss Messung und Übertragung..... | 15 |
| Abbildung 8: | Webapp Zustand Darstellung..... | 16 |
| Abbildung 9: | Registrierung von Geräten..... | 17 |
| Abbildung 10: | Energieverbrauch eine LoRa..... | 19 |
| Abbildung 11: | Dragino LoRaWAN Gateway..... | 20 |
| Abbildung 12 : | Microcontroller ST - NUCLEO-F767ZI..... | 20 |
| Abbildung 13: | Grove-Wio-E5 (LoRa Module)..... | 21 |
| Abbildung 14: | Grove Elektrizitätssensormodul..... | 22 |
| Abbildung 15: | Aufbau der Funksensor..... | 23 |
| Abbildung 16: | Ablesen Workflow der maximale stromwert..... | 24 |
| Abbildung 17: | TTN Application..... | 25 |

| | |
|--|----|
| Abbildung 18: Verbindung des Gateways an das Internet..... | 25 |
| Abbildung 19: Internet-Router verbundene Geräte bzw. Dragino Gateway | 26 |
| Abbildung 20: Dragino Gateway Konfiguration..... | 27 |
| Abbildung 21: Erfolgreiche Verbindung an TTN..... | 27 |
| Abbildung 22: Verbundenen LoRa Modul an TTN..... | 28 |
| Abbildung 23: Verbundene Gateway ans Internet und an LoRaWAN | 28 |
| Abbildung 24: Verbundene Gateway an TTN | 29 |
| Abbildung 25: Auswahl der ML-Model | 29 |
| Abbildung 26: Payload Decodierte Beispiel | 30 |
| Abbildung 27: Lokale host im Wenhook..... | 31 |
| Abbildung 28: Webapp Zustand Darstellung | 31 |
| Abbildung 29: Workflow der Ausarbeitung | 32 |
| Abbildung 30: Reset Taste der Funksensor | 39 |

1 Einleitung und Motivation

1.1 Motivation

In der heutigen Zeit der medizinischen Innovation und Komplexität stehen Krankenhäuser vor der Herausforderung, eine wachsende Anzahl von Medizingeräten effizient zu verwalten. Die Überwachung und Wartung dieser Geräte sind entscheidend, wird jedoch oftmals durch unklare Informationen über den Standort und Zustand der Geräte behindert. Die Unsicherheit über die Nutzung und den Bedarf führt oft zu überflüssigen Anschaffungen und zu hohen Wartungskosten für selten benutzte Geräte.

Die Coronavirus-Pandemie hat diese Herausforderungen noch verschärft, indem sie zu einem plötzlichen Anstieg der Anschaffungen von Medizingeräten führte. Mit dem Ende der akuten Phase der Pandemie steht das Gesundheitswesen nun vor der Aufgabe, den tatsächlichen Bedarf an diesen Geräten zu bewerten und ihre Verwaltung zu optimieren.

In dieser Arbeit wird daher ein innovatives Zustandsmonitoring-System für Medizingeräte vorgestellt, das mittels energiesparender Funk-Sensoren den Zustand der Geräte erfasst. Die Motivation für diese Forschung liegt in der Notwendigkeit, die Effizienz in der Verwaltung von Medizingeräten zu steigern, die Kosten für unnötige Wartungen zu reduzieren und eine fundierte Entscheidungsfindung in Bezug auf Anschaffungen und Wartung zu ermöglichen. Ziel ist es, frühzeitige Fehlererkennung oder Störungen am Gerät zu ermöglichen und gleichzeitig die finanzielle und operative Effizienz von Gesundheitseinrichtungen zu erhöhen.

1.2 Ziel der Arbeit

Diese Arbeit zielt darauf ab, ein innovatives Überwachungssystem zu entwickeln, das den Zustand von Medizingeräten mithilfe energiesparender Funksensoren erfasst. Dieses System soll die Effizienz der Gerätenutzung steigern, die Wartungskosten senken und sicherzustellen des Gerät Funktion. Daher zielt die Arbeit darauf ab, die Qualität und Kosteneffizienz in medizinischen Einrichtungen deutlich zu steigern. Die Ziele könnten wie folgt formuliert werden.

- Entwicklung eines energiesparenden Funksensors.
- Entwurf eines Datenverarbeitungssystems, das den Zustand (wie AN, AUS, Idle und Im Betrieb) von Medizingeräten mithilfe energiesparender Funk-Sensoren erfassen kann.
- Darstellung der erkannten Zustände auf einer Weboberfläche.
- Minimierung der Datenmenge bei der Übertragung zwischen Sensor und The Things Network.

2 Grundlagen

Das nachfolgende Kapitel vermittelt die essenziellen Grundlagen, um die Konzeption und Funktionsweise des Funksensors besser zu verstehen. Auch um das Thema besser zu verstehen. Dabei wird zu Beginn ein Überblick über das Gerätemanagement in Praxiseinrichtungen eingeworfen und was genaue Zustandsmonitoring bedeutet. Es folgt die Erklärung der LoRaWAN und Maschine Learning.

2.1 Gerätemanagement im Praxiseinrichtungen

Gerätemanagement bezeichnet das systematische Organisieren, Überwachen und Pflegen eines Bestands an Geräten oder Ausrüstungen. Es umfasst Aktivitäten wie Anschaffung, Wartung, Reparatur, Inventarisierung und Entsorgung von Geräten.

Ein systematisches Gerätemanagement ist unerlässlich, um einen ständigen Überblick über den Gerätebestand zu behalten. Dies gilt insbesondere für größere Praxen mit vielen Geräten. Ein strukturiertes Management stellt sicher, dass bei Wartungsarbeiten keine Geräte übersehen werden. Ein Inventarverzeichnis dient hierbei als schnelle Überprüfungsmöglichkeit [1].

Darüber hinaus gewährleistet effektives Gerätemanagement, dass nur geschultes Personal Geräte bedient und pflegt, was die Nutzungsdauer verlängert und die Wirtschaftlichkeit erhöht. Fehlendes oder mangelhaftes Gerätemanagement kann zu vorzeitigem Geräteausfall, teuren Reparaturen und somit zu höheren Kosten führen. Darüber hinaus kann ein Geräteausfall die Patientenbehandlung beeinträchtigen, wenn zum Beispiel Therapien unterbrochen oder Behandlungstermine abgesagt werden müssen. [1]

Laut einer Studie beläuft sich die durchschnittliche Anzahl der Medizingeräte in Universitätsmedizinzentren (UMCs) in Deutschland auf 25.000 Geräte (siehe Abbildung 1) [2]. Diese Zahl verdeutlicht die Notwendigkeit eines effektiven Gerätemanagements, um den reibungslosen Betrieb einer medizinischen Einrichtung sicherzustellen und die Informationssicherheit sowie Behandlungsqualität aufrechtzuerhalten. Ineffiziente Managementpraktiken bei einer solchen Gerätemenge können erhebliche finanzielle und operative Risiken mit sich bringen.

Schließlich ist gutes Gerätemanagement nicht nur aus Gründen der Patientensicherheit wichtig, sondern auch aus wirtschaftlicher Perspektive. Durchschnittlich können Reparaturkosten 7,4% der Anschaffungskosten eines Geräts betragen. Ein effektives Management kann diese Kosten reduzieren [3]

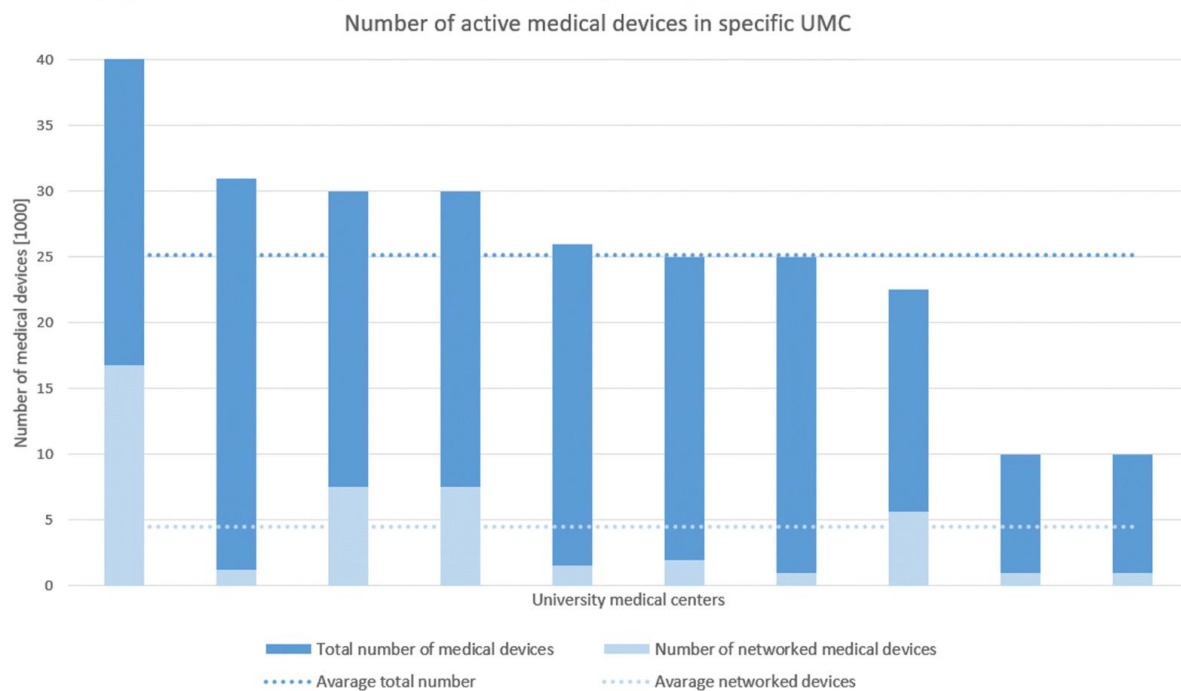


Abbildung 1: Anzahl von Medizinische Geräte in Universitätsmedizinzentren in Deutschland

(Quelle BioMed Central Ltd, 2023)

2.1.1 Was ist Zustandsmonitoring?

Das Zustandsmonitoring, oft als "Condition Monitoring" (CM) bezeichnet, ist eine Technik, die Maschinen- und Anlagenkomponenten systematisch während des Betriebs überwacht. Ziel ist es, Veränderungen frühzeitig zu erkennen, die auf mögliche Fehler oder Funktionsstörungen hinweisen. Ziel dieser Überwachung ist es, den genauen Zeitpunkt für Wartungs- oder Reparaturmaßnahmen zu bestimmen und somit ungeplante Stillstandzeiten und mögliche Schäden oder Ausfälle zu vermeiden.

2.1.1.1 Warum ist Zustandsüberwachung wichtig?

Die Überwachung von Medizingeräten hat sich von manuellen Inspektionen zu automatisierten Systemen entwickelt, die mithilfe von Sensoren Zustandsdaten in Echtzeit sammeln [4].

Im Bereich der Intensivmedizin und bei implantierbaren Medizingeräten wird das Zustandsmonitoring besonders intensiv genutzt, um das Wohlbefinden des Patienten kontinuierlich zu überwachen [5].

Vorteile von Zustandsmonitoring:

- **Frühzeitige Fehlererkennung:** Mithilfe von Zustandsüberwachung können potenzielle Probleme und Fehlfunktionen frühzeitig erkannt werden, bevor sie zu einem ernsthaften Ausfall führen [6].
- **Verlängerung der Lebensdauer von Anlagen:** Durch die rechtzeitige Identifikation und Behebung kleinerer Probleme kann die Lebensdauer von Maschinen und Anlagen verlängert werden [6].
- **Verringerung ungeplanter Ausfallzeiten:** Durch die Vorhersage von Maschinenproblemen können ungeplante Stillstandzeiten und die damit verbundenen Produktionsverluste minimiert werden [7] [8].

- **Kostensparnis:** Ungeplante Stillstandzeiten und Notfallreparaturen können sehr teuer sein. Zustandsüberwachung kann dazu beitragen, diese Kosten zu senken, indem potenzielle Probleme frühzeitig erkannt und behoben werden [6].
- **Optimierung von Wartungsplänen:** Anstatt Maschinen basierend auf einem festen Zeitplan zu warten, ermöglicht die Zustandsüberwachung die Umsetzung von zustandsabhängigen Wartungsstrategien [7]. Dies bedeutet, dass die Wartung nur durchgeführt wird, wenn sie tatsächlich erforderlich ist.

2.2 Internet of Things (IOT)

Das Internet of Things (IoT) bezieht sich auf die Vernetzung physischer Geräte – von Alltagsobjekten über Fahrzeuge bis hin zu Gebäudeinfrastrukturen –, die Daten sammeln und austauschen können [9]. Über diese Verbindungen können Geräte ohne menschlichen Eingriff miteinander kommunizieren und intelligente Entscheidungen treffen [10].

Das IoT ist in vielen Sektoren wie Gesundheitswesen, Landwirtschaft, Verkehr und Energie integriert, [11] wodurch sowohl Unternehmen als auch Verbraucher von Daten profitieren. Trotz seiner Vorteile gibt es bei der Implementierung des IoT Herausforderungen in puncto Sicherheit, Datenschutz und technische Aspekte wie Interoperabilität [12]. Dennoch wird angesichts technologischer Innovationen ein stetiges Wachstum des IoT prognostiziert.

2.3 LoRaWAN

LoRaWAN, das für "Long Range Wide Area Network" steht, ist ein Netzwerkprotokoll, das Netzwerke mit geringem Stromverbrauch und großer Reichweite ermöglicht. Es wurde speziell für das Internet der Dinge (IoT) entwickelt und verwendet die LoRa-Modulationstechnik zur Kommunikation zwischen IoT-Geräten (Endgeräten) und Netzwerk-Gateways [13].

Vergleich von LoRaWAN mit anderen drahtlosen Technologien

Es gibt viele drahtlose Kommunikationsoptionen wie LTE.M, WIFI, Zigbee, Bluetooth, 5G und natürlich LoRa bzw. LoRaWAN. (Abbildung 2)

Jede dieser Technologien hat ihre eigenen Vor- und Nachteile:

- **Bluetooth** verbraucht im Vergleich zu Wi-Fi und LTE zwar weniger Energie, benötigt jedoch immer noch mehr als LoRa. Zudem ist seine Reichweite kürzer, wodurch es sich eher für Geräte eignet, die sich in unmittelbarer Nähe zueinander befinden. [14]
- **5G** bietet höhere Geschwindigkeiten als LoRaWAN, ist jedoch in seiner Ausdehnung noch begrenzt. Obwohl 5G in der Lage ist, mehr Daten schneller zu übertragen, erfordert der Aufbau der dafür notwendigen Infrastruktur Zeit und erhebliche Investitionen. [14]

Dagegen hat sich **LoRaWAN**, insbesondere in industriellen Umgebungen, als bevorzugtes Netzwerk für IoT-Geräte herausgestellt. Diese Geräte übertragen zuverlässig kleine Datenpakete – von Temperatur bis Luftfeuchtigkeit.

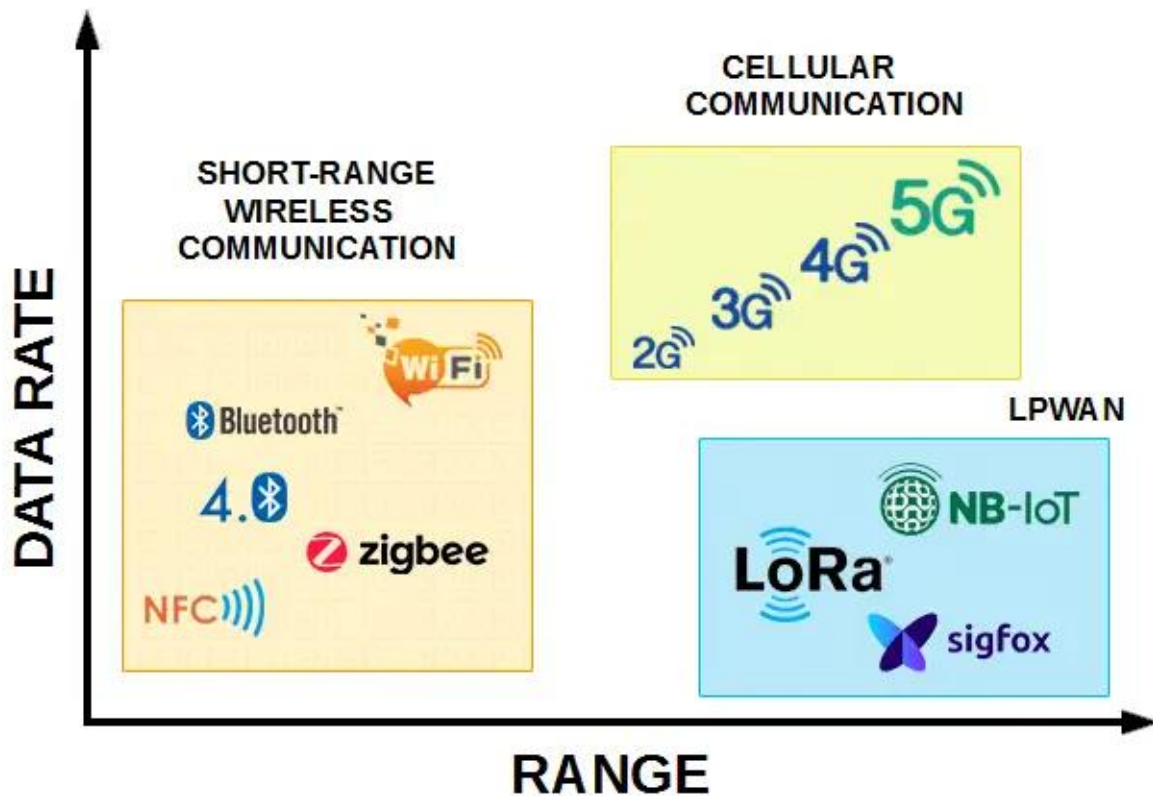


Abbildung 2: Verschiedene drahtlosen Kommunikationstechnologien

(Quelle mokolora.com, 2022)

LoRaWAN Architektur

Das LoRaWAN-IoT-Netzwerk ist in einer Stern-Stern-Topologie implementiert. Hierbei übertragen Gateways Nachrichten zwischen den Endgeräten und einem zentralen Netzwerkprozessor. Auf der LoRa-Physikschicht wird drahtlose Technologie eingesetzt, um die Vorteile der Long-Range-Eigenschaft zu nutzen. Dies ermöglicht eine Einzelpunkt-zu-Mehrpunkt-Kommunikation zwischen einem Endgerät und einem oder mehreren Gateways. Beidseitige Konnektivität ist in beiden Typen möglich, und es werden auch Multicast-Gruppen unterstützt, was eine effiziente Nutzung des Spektrums während Aufgaben wie Firmware Over The Air (FOTA) Updates oder anderen Massenslieferungsnachrichten ermöglicht [14].

Abbildung 3 zeigt die LoRaWAN Netzwerkinfrastruktur

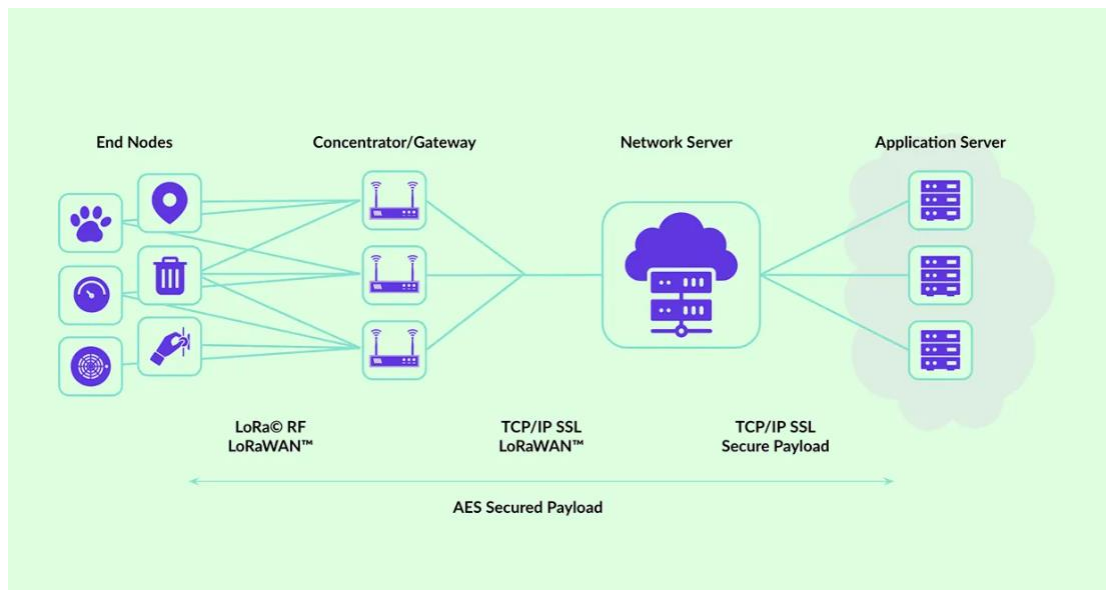


Abbildung 3: LoRaWAN-Netzwerkinfrastruktur

(Quelle mokolora.com, 2022)

Endgeräte -Sensoren oder Aktuatoren senden drahtlos LoRa-modulierte Nachrichten an die Gateways oder empfangen drahtlos Nachrichten von den Gateways zurück. [13]

Gateways -empfangen Nachrichten von Endgeräten und leiten sie an den Netzwerkservers weiter. [13]

Netzwerkservers -eine Software, die auf einem Server läuft und das gesamte Netzwerk verwaltet. [13]

Einige Hauptmerkmale von LoRaWAN sind:

- **Adaptive Datenrate (ADR):** Das Protokoll passt die Datenrate je nach Entfernung und Signalstärke zwischen dem Endgerät und dem Gateway an [13].
- **Verschlüsselung:** LoRaWAN gewährleistet die Sicherheit der übertragenen Daten durch Ende-zu-Ende-Verschlüsselung [15].
- **Skalierbarkeit:** Es unterstützt tausende von gleichzeitig verbundenen Geräten pro Gateway [15].
- **Geolocation:** LoRaWAN kann auch für die ungefähre Standortbestimmung von Endgeräten genutzt werden, ohne dass GPS erforderlich ist. [15]

LoRa unterstützt zwei Verifikationstechniken: die personalisierte Aktivierung ABP und die Over-the-Air Activation (OTAA).

Bei der **personalisierten Aktivierung ABP** werden die Device-Adresse (DevAddr) und Verschlüsselungsschlüssel frühzeitig in das Gerät geschrieben (Gerätepersonalisierung). Geräte werden mit Netzwerk- und Anwendungssitzungsschlüsseln sowie einer vorab zugewiesenen 32-Bit-Netzwerkadresse konfiguriert, ähnlich der Zuweisung einer statischen IP-Adresse.

Für diese Arbeit wurde die **Over-the-Air -Activation (OTAA)** als Aktivierungsmethode verwendet. Dies ist ein zentrales Sicherheitsmerkmal von LoRaWAN. Bei der OTAA-Methode wird ein gemeinsamer Prozess benötigt, um die Geräteadresse (DevAddr) und die

Sitzungsverschlüsselungsschlüssel zu erstellen. Mit OTAA können Geräte eine Kommunikationsanfrage an einen Netzwerkserver senden. Dieser Server authentifiziert dann das Gerät und weist ihm eine Adresse und einen Token zur Beschaffung von Sitzungsschlüsseln zu. Die Netzwerk- und Anwendungssitzungsschlüssel werden während des Verbindungsverfahrens von dem zuvor auf dem Gerät bereitgestellten öffentlichen Anwendungsschlüssel abgeleitet [14].

In einem typischen LoRaWAN-Setup senden Endgeräte drahtlos Daten zu den Gateways, welche diese Nachrichten an einen zentralen Netzwerkserver weiterleiten. Dieser Server dekodiert dann die Daten und leitet sie an entsprechende Anwendungs-Server weiter [13].

Die LoRa Alliance, eine Allianz von Unternehmen, die sich für die Verbreitung von LoRaWAN engagieren, pflegt und entwickelt den LoRaWAN-Standard weiter. Dieser offene Standard sorgt für Interoperabilität zwischen Geräten und Netzwerken verschiedener Hersteller [15].

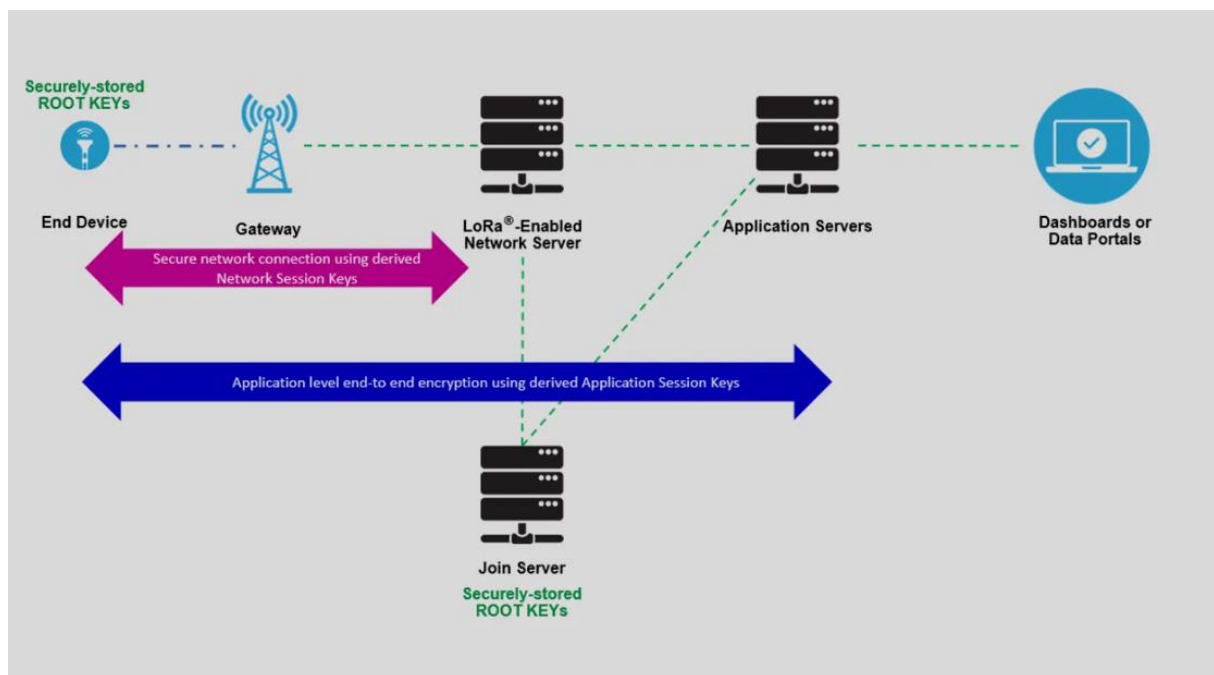


Abbildung 4: Authentifizierung von Geräten OTAA

(Quelle mokolora.com, 2022)

2.4 LoRa

LoRa (Long Range), ist eine von Semtech patentierte Modulationstechnik, die in IoT-Netzwerken zur drahtlosen Kommunikation eingesetzt wird [16]. LoRa ist eine drahtlose Modulationstechnik, die von der Chirp Spread Spectrum (CSS) Technologie abgeleitet ist. Sie kodiert Informationen auf Funkwellen mit Hilfe von Chirp-Impulsen [17]. Aufgrund ihrer Fähigkeit, Daten über weite Entfernungen mit geringem Energieverbrauch zu senden, eignet sich LoRa besonders für IoT-Anwendungen, die eine weite Netzabdeckung und Langlebigkeit erfordern. LoRa bietet verschiedene konfigurierbare Parameter, die es erlauben, die Übertragung an spezifische Anforderungen wie Reichweite, Datenrate und Energieverbrauch anzupassen.

Die wichtigsten Parameter von LoRa sind:

Bandbreite (BW): Die Bandbreite beschreibt die Breite des Spektrums, das von einem LoRa-Signal belegt wird. Typische Werte sind 915 MHz, 868 MHz und 433 MHz. [17]

Reichweite: LoRa hat eine hohe Sensibilität von -137 dBm, was die Verfügbarkeit des Netzes erhöht. Die Signale durchdringen Gebäudemauern problemlos und erreichen so zum Beispiel auch Kellerräume oder andere, sogenannte Deep-Indoor-Standorte. Die Verfügbarkeit kann also auch dort gegeben sein, wo Mobilfunk typischerweise nicht hinkommt. [18]

Codierungsfaktor (SF, Spreading Factor): Der SF reicht von SF7 bis SF12. Ein höherer SF erhöht die Zeit, die für die Übertragung einer Nachricht benötigt wird, verbessert aber die Reichweite und die Empfindlichkeit des Empfängers. Ein niedriger SF führt zu einer höheren Datenrate, reduziert aber die Reichweite. [15]

Fehlertoleranz (CR, Coding Rate): Die Kodierungsrate kann von 4/5 bis 4/8 variieren. Eine höhere CR erhöht die Redundanz der Übertragung, was die Fehlerresistenz verbessert, aber auch die benötigte Übertragungszeit verlängert. [17]

2.5 The Things Network

The Things Network (TTN) stellt eine global vernetzte und offene Infrastruktur für das Internet der Dinge (IoT) bereit, die hauptsächlich auf LoRaWAN-Technologie basiert (Abbildung 3). Es ermöglicht Endbenutzern und Entwicklern den Aufbau und Betrieb von IoT-Anwendungen, ohne dabei auf proprietäre Netzwerke oder teure Datenpläne angewiesen zu sein [13].

TTN wurde mit der Vision ins Leben gerufen, eine demokratische und dezentralisierte Netzwerkinfrastruktur zu schaffen, die durch Gemeinschaften und Einzelpersonen weltweit betrieben wird. Dieses dezentrale Modell stellt sicher, dass das Netzwerk selbst in Bereichen verfügbar ist, in denen kommerzielle IoT-Netzwerke nicht präsent sind [9].

Durch seine Flexibilität und Offenheit eignet sich TTN für eine Vielzahl von Anwendungen, von städtischer Infrastruktur über Landwirtschaft bis hin zu Smart Home-Lösungen [19].

2.5.1 Encoder/Decoder in TTN

Die Verwendung von The Things Network (TTN) für LoRaWAN-basierte Anwendungen erfordert oft die Kodierung und Dekodierung von Daten, um sicherzustellen, dass die übertragenen Nachrichten sowohl effizient als auch in einem verständlichen Format sind. Diese Kodierung und Dekodierung geschieht in der Regel auf Anwendungsebene im TTN-Stack. Hier ist eine Übersicht über den Prozess:

Warum Kodieren?

Daten, die von IoT-Geräten gesendet werden, können aus verschiedenen Sensormessungen bestehen, z. B. Temperatur, Feuchtigkeit, Position usw. Direktes Senden dieser Daten im Klartext oder im JSON-Format wäre ineffizient und würde mehr Energie und Bandbreite verbrauchen. Daher werden die Daten vor dem Senden kodiert [15].

Payload Formats in TTN

TTN bietet in seiner Konsole die Möglichkeit, benutzerdefinierte Payload-Formate (wie Encoder- und Decoder-Funktionen) in **JavaScript** zu definieren (siehe Anhang) [20]:

Encoder: Diese Funktion wird verwendet, um die Daten vor dem Senden vom Gerät zu kodieren. Zum Beispiel kann ein Temperaturwert von 25,5 Grad Celsius in einen 2-Byte-Wert umgewandelt werden.

Decoder: Nachdem die Daten von einem Gerät empfangen wurden, wird diese Funktion im TTN-Stack verwendet, um die Rohdaten in ein verständliches Format zurückzukonvertieren, z.B. von einem 2-Byte-Wert zurück zu einem Temperaturwert.

Vorteile der Kodierung und Dekodierung

- **Effizienz:** Reduziert die Größe der übertragenen Daten, was zu einer längeren Batterielebensdauer und geringeren Übertragungskosten führt [21].
- **Flexibilität:** Benutzer können Datenformate je nach Anforderung anpassen.
- **Sicherheit:** Obwohl dies nicht als Hauptverschlüsselungsmechanismus dient, bietet die Kodierung eine zusätzliche Ebene der Datenunschärfe.

2.6 Maschine Learning Model

Maschinelles Lernen (ML) ist ein Zweig der künstlichen Intelligenz (KI) und der Informatik, der sich auf die Verwendung von Daten und Algorithmen konzentriert, um die Art und Weise, wie Menschen lernen, nachzuahmen und die Genauigkeit schrittweise zu verbessern. [22] Es gibt 3 Haupt Maschine Learning Modelle:

überwachtes maschinelles Lernen, wird durch die Verwendung gekennzeichnete Datensätze definiert, um Algorithmen zu trainieren, um Daten zu klassifizieren oder Ergebnisse genau vorherzusagen. Einige beim überwachten Lernen verwendete Methoden umfassen neuronale Netze, Naive Bayes, lineare Regression, logistische Regression, Random Forest und Support Vector Machine (SVM). [22]

unüberwachtes maschinelles Lernen, verwendet Algorithmen für maschinelles Lernen, um unbeschriftete Datensätze zu analysieren und zu gruppieren. Diese Algorithmen entdecken verborgene Muster oder Datengruppierungen, ohne dass ein menschliches Eingreifen erforderlich ist. Hauptkomponentenanalyse (PCA) und Singular Value Decomposition (SVD) sind hierfür zwei gängige Ansätze. [22]

Halbüberwachtes Lernen bietet einen guten Mittelweg zwischen überwachtem und unbeaufsichtigtem Lernen. Während des Trainings wird ein kleinerer beschrifteter Datensatz verwendet, um die Klassifizierung und Merkmalsextraktion aus einem größeren, unbeschrifteten Datensatz zu steuern. Halbüberwachtes Lernen kann das Problem lösen, dass nicht genügend gekennzeichnete Daten für einen überwachten Lernalgorithmus zur Verfügung stehen. [22]

2.6.1 Random Forest Algorithmus

Random Forest ist ein Überwachter Algorithmus für Maschine Learning, der häufig bei der Klassifizierung und Regressionsproblemen eingesetzt wird. Es baut Entscheidungsbäume auf verschiedenen Stichproben auf und nutzt deren Mehrheitsvotum für die Klassifizierung und den Durchschnitt im Falle einer Regression. Eines der wichtigsten Merkmale des Random-Forest-Algorithmus besteht darin, dass er Datensätze verarbeiten kann, die kontinuierliche Variablen und kategoriale Variablen enthalten. [23]

3 Anforderungen für das Zustandsmonitoring von medizinischen Geräten

3.1 Anwendungsszenario Anforderung

In einem modernen Krankenhaus ist es entscheidend, dass alle medizinischen Geräte ordnungsgemäß funktionieren und verfügbar sind, wenn sie benötigt werden. Hier wird das Zustandsmonitoring eines Ultraschallgerätes mit energiesparenden Funk-Sensoren beschrieben.

Geräteidentifikation:

Jedes Ultraschallgerät sollte eindeutig identifizierbar sein, sei es durch eine Gerätenummer, einen QR-Code oder eine andere Methode.

Zustandserkennung:

Der Funk Sensor sollte in der Lage sein, alle vier Zustände des Ultraschallgerätes zu erkennen: AN, AUS, im Betrieb und Idle. Bei einem Wechsel des Gerätezustandes sollte dies innerhalb von Sekunden im System erkannt und dokumentiert werden.

Zeitliche Auflösung:

Die Zustandsinformationen sollten in ungefähr 5 Sekunden kurzen, regelmäßigen Intervallen aktualisiert werden.

Ereignismeldung:

Im Falle einer Anomalie (Zum Beispiel wenn ein Gerät länger als eine festgelegte Zeit im "Im Betrieb"-Zustand bleibt) sollte das System eine Meldung anzeigen (auch eine Nachricht an das Personal senden).

Bei niedrigem Batteriestand des Funksensors sollte ebenfalls eine Benachrichtigung versendet werden.

Anomalie Erkennung:

Das System sollte in der Lage sein, unerwartete Zustandsänderungen oder Abweichungen von einem typischen Betriebsmuster zu erkennen. Beispielsweise, wenn ein Ultraschallgerät ungewöhnlich oft zwischen den Zuständen "Im Betrieb" und "Idle" wechselt.

Datenspeicherung:

Alle Zustandsinformationen, einschließlich des Zeitstempels, sollten in einer Datenbank gespeichert werden, um eine historische Analyse und Trenderkennung zu ermöglichen.

Schnittstelle zur Datenabfrage:

Das System sollte eine Benutzeroberfläche (zum Beispiel eine Webapp) bieten, über die das Krankenhauspersonal den aktuellen Zustand sowie die Historie der Zustände eines bestimmten Ultraschallgerätes abrufen kann.

Für das zuverlässige Monitoring des Zustands medizinischer Geräte sind bestimmte Anforderungen an den Funksensor zu stellen. Ein solcher Sensor muss über die Fähigkeit verfügen, den aktuellen Stromverbrauch des jeweiligen medizinischen Geräts mit Hilfe eines analogen Stromsensors genau zu erfassen.

Die Aufgabe der Datenerfassung und -verarbeitung wird von einem leistungsfähigen Mikrocontroller übernommen. Konkret handelt es sich hierbei um das Modell STM32F767ZI von ST. Nachdem die Daten erfasst wurden, kommen Maschinenlern-Algorithmen zum Einsatz, um die Informationen zu analysieren und zu kategorisieren.

Die erfassten Stromsignale werden über eine LoRa-Schnittstelle übertragen. Im nächsten Schritt werden diese Daten an einen Anwendungsserver gesendet, in diesem Fall an The Things Network (TTN).

Die dafür gebrauchten Komponenten sind:

Tabelle 1 Verwendete Komponenten

| | |
|-------------------------|--|
| Basis/Entwicklungsboard | ST-NUCLEO-F767ZI |
| LoRa-Modul | Seeed Studio – Grove LoRa-E5 |
| LoRaWAN Gateway | Dragino LPS8v2 |
| Stromsensor | Seeed studio – Grove electricity sensor (TA12-200) |

3.2 Grundlegende Anforderungen an das Monitoring-System

Für ein wirkungsvolles Zustandsmonitoring von Medizingeräten muss das System kontinuierlich den Gerätezustand überwachen können. Es sollte frühzeitig jegliche Abweichungen oder Anomalien erkennen. Eine intuitive und benutzerfreundliche Bedienung ist entscheidend, um eine nahtlose Integration in den Krankenhausalltag zu ermöglichen. Bei festgestellten Anomalien oder drohenden Ausfällen sollte das System rechtzeitig warnen.

3.3 Anforderungen an den energiesparenden Funksensor:

Bei der Konzeption eines energiesparenden Funksensors für das Zustandsmonitoring von Medizingeräten sind folgende Kriterien von zentraler Bedeutung:

- **Lange Übertragungreichweite:** Der Sensor sollte Daten über große Distanzen senden können, um eine durchgehende Überwachung in weitläufigen Krankenhauseinrichtungen sicherzustellen.
- **Hohe Präzision:** Der Sensor muss genau messen, um verlässliche Daten bereitzustellen.
- **Gateway-Integration:** Ein integriertes Gateway sollte die Daten des Sensors effizient an das IoT bzw. TTN weitergeben. Ein Gateway könnte für mehrere Funksensoren benutzt werden um weniger Kosten zu haben
- **Zuverlässigkeit:** In einem medizinischen Kontext ist es unerlässlich, dass der Sensor beständig und ohne Fehler operiert.
- **Optimale Übertragungsfrequenz:** Die Frequenz sollte so gewählt werden, dass sie eine klare Kommunikation gewährleistet und Interferenzen mit anderen Geräten vermeidet.
- **Dauerbetrieb:** Der Mikrocontroller des Sensors muss fähig sein, ununterbrochen zu arbeiten.
- **Energieeffizienz:** Trotz Dauerbetrieb sollte der Sensor energieeffizient sein, um Wartungen und Batteriewechsel selten zu machen.

- **Robustheit:** Der Sensor sollte über längere Zeiträume verlässlich funktionieren, ohne dass häufige Reparaturen oder ein Austausch notwendig sind.

Die Beachtung dieser Kriterien garantiert nicht nur eine effektive Überwachung des Zustands von Medizingeräten, sondern unterstützt auch die Erhöhung der Patientensicherheit und die Reduzierung von Betriebskosten.

3.3.1 Anforderung der Funksensor

Der Funksensor sollte extern an der Medizingerät angebracht werden. Dies verhindert nicht nur potenzielle Schäden am zu überwachenden Medizingerät, sondern erhöht auch die Flexibilität des Systems, da es bei unterschiedlichen Gerätetypen und -modellen eingesetzt werden kann.

Für den Funksensor sollte ein Microcontroller die UART (Universal Asynchronous Receiver / Transmitter) besitzen, für den Austausch von seriellen Daten (ZB. Strom Daten) zwischen der LoRa Modul und der Microcontroller selbst.

3.3.2 Datenminimierung und Übertragung

Die Menge der Stromwerten für ein Gerät ist oft riesig, da die Geräte fast immer angeschlossen ist, auch wenn es nicht betrieben wird. Daher ist es von entscheidender Bedeutung, die Datenmenge so weit wie möglich zu minimieren, ohne die Qualität und Genauigkeit der Information zu beeinträchtigen. Effiziente Algorithmen zur Datenkomprimierung und selektiven Übertragung von nur kritischen oder relevanten Daten können dabei helfen, die Belastung der Netzwerke zu verringern und gleichzeitig eine schnelle Datenübertragung zu gewährleisten.

3.3.3 Leistungseffizienter Algorithmus

Energieeffizienz ist ein Schlüsselaspekt, wenn es um die Überwachung von Medizingeräten geht. Der eingesetzte Algorithmus sollte nicht nur präzise und verlässlich in der Datenerfassung und -analyse sein, sondern auch so gestaltet sein, dass er den Stromverbrauch minimiert. Dies kann durch Optimierung des Codes, Verringerung redundanter Operationen und Einsatz von stromsparenden Techniken und Architekturen erreicht werden. Ein leistungseffizienter Algorithmus sorgt dafür, dass die Überwachungssensoren länger laufen, bevor ein Batteriewechsel oder eine erneute Aufladung notwendig wird

4 Lösungskonzept

Das Lösungskonzept, das in diesem Kapitel entwickelt wird, basiert auf den Anforderungen, wie die genannten Anforderungen am besten erfüllt werden können. Jede gestellte Anforderung stellt eine spezifische Herausforderung dar, die mit einem bestimmten Design- oder Technologieelement gelöst werden muss.

4.1 Anwendungsfall des Gesamtprojekts

Die Abbildung 5 gibt einen detaillierten Überblick über die einzelnen Komponenten sowie über den Prozess der Datenübertragung zwischen dem LoRa-Modul, dem LoRa-Gateway und TTN.

Der Datenübertragungsprozess von Stromwerten kann wie folgt zusammengefasst werden:

- **Datenerfassung durch den Sensor:** Der Stromsensor würde die maximale Stromwert von der angeschlossenen Medizin Gerät messen. Da der Stromsensor direkt mit der LoRa Modul kommuniziert, erhält das loRa Modul die gemessene Stromwerten.
- **Datenverarbeitung und -übertragung durch den Mikrocontroller:** Da viele Stromwerten ausgelesen werden, ist es sinnvoll die Werten erst zu komprimieren, bevor es an TTN übertragen werden. Der Mikrocontroller nimmt eine Verarbeitung der erfassten Daten vor, kodiert sie entsprechend und überträgt sie in Form von HEX-Daten an TTN mittels des LoRa-Moduls und des LoRa-Gateways.
- **Datenempfang und -dekodierung durch TTN:** TTN empfängt die übertragenen HEX-Daten, dekodiert diese und konvertiert sie in ein ASCII-Format für eine weiterführende Analyse oder Verarbeitung.

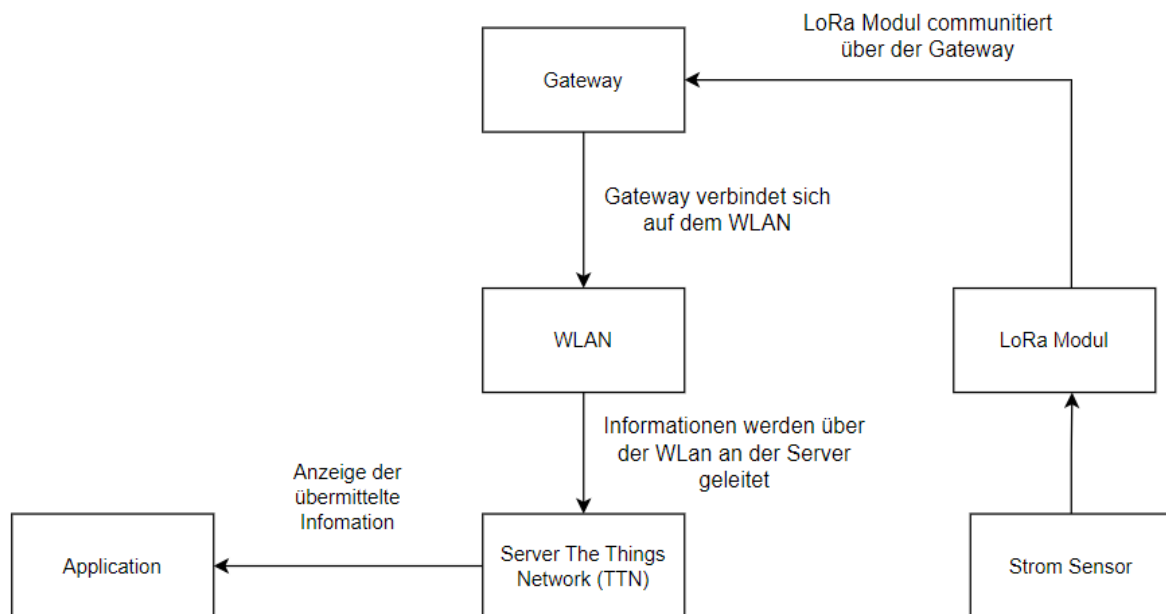


Abbildung 5: LoRaWAN Kommunikation Ablauf Diagramm

4.1.1 Aufnahme der Daten von verschiedenen Geräten

Für die Überwachung und Zustandsbewertung von medizinischen Geräten wird ein System entwickelt, das die Stromwerte von verschiedenen Geräten erfassen, verarbeiten und an das TTN übertragen kann.

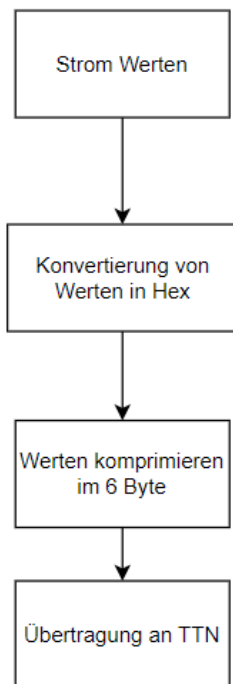


Abbildung 6: Workflow Datenaufnahme

Um dies zu erreichen, werden die folgenden Schritte durchgeführt:

- **Stromwert-Erfassung:**

Ein Funksensor wird verwendet, um den aktuellen Stromverbrauch von medizinischen Geräten zu messen. Diese Sensoren ist flexibel und kann ohne physische Eingriffe in das medizinische Gerät angebracht werden.

- **Konvertierung von Werten:**

Die erfassten momentan analogen Stromwerten werden in digitalem integrem Werten umgewandelt. Eine Funktion in dem Code vom Mikrokontroller ist dafür zuständig, der maximale Stromwert von jeder Oszillation zu ermitteln. Dieser Maximale Stromwert wird dann in eine hexadezimale Form (Hex) konvertiert, um die Datenübertragung und -verarbeitung zu vereinfachen.

- **Wertekomprimierung:**

Die hexadezimalen Daten werden komprimiert, um die Menge der zu übertragenden Daten zu minimieren und die Übertragungsgeschwindigkeit zu optimieren. In diesem Fall werden die Daten in einem 6-Byte-Format komprimiert, um eine effiziente und schnelle Übertragung zu gewährleisten.

Zum Beispiel:

Dezimal Stromwert :000209

Hexadezimal Stromwert: D1

- **Übertragung an TTN:**

Die komprimierten Daten werden dann drahtlos über das LoRa-Protokoll an TTN übertragen. TTN dient als zentraler Datenhub, der die Daten von verschiedenen Sensoren sammelt, speichert und für weitere Analysen und Aktionen bereitstellt.

Abbildung 7 ist eine Darstellung von der gesamte Lösungskonzept.

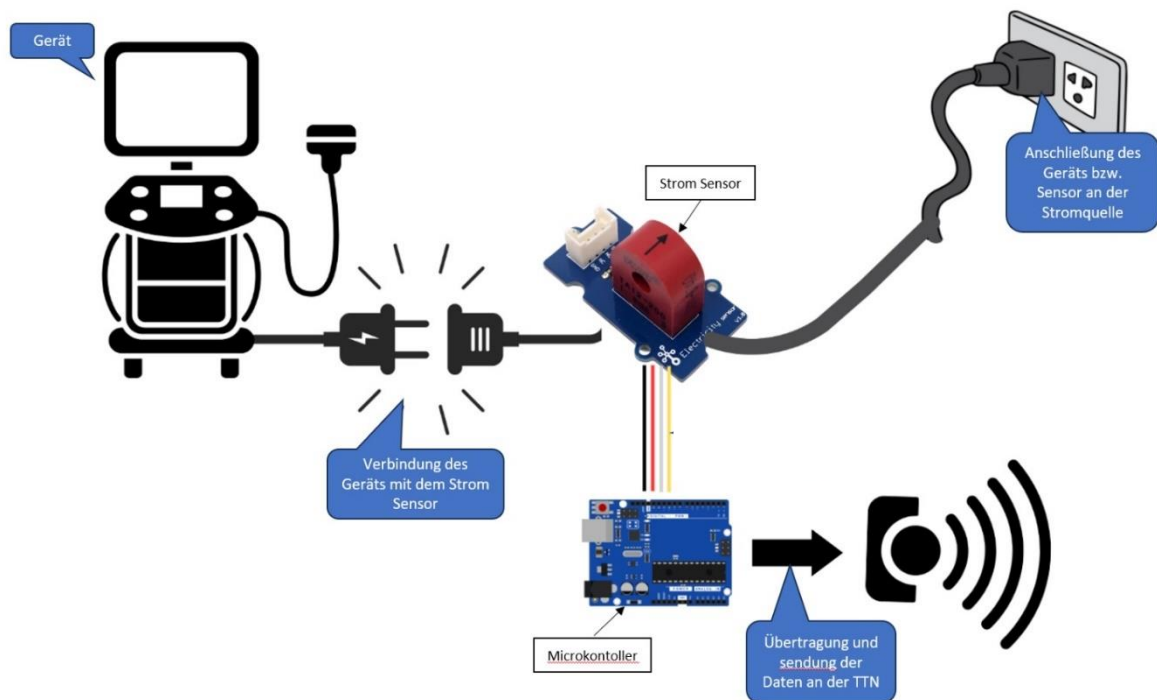


Abbildung 7: Strom Fluss Messung und Übertragung

Auswertung der Maximale Stromwerte für die Zustandserkennung bei jeder Oszillation

Zunächst sollte zur Erkennung der Gerätezustände ein Schwellwert für unterschiedliche Gerätezustände festgelegt werden. Nach der Etablierung dieser Methode ermöglicht die Einbindung des gemessenen Stroms dem Programm, den Zustand des gerade angeschlossenen Geräts zu identifizieren.

Um die Genauigkeit dieser Zustandserkennung zu steigern, empfiehlt es sich, ein maschinelles Lernmodell bzw. KI-Modell mit den Stromwerten verschiedener Geräte zu trainieren. Dies ermöglicht dem System, basierend auf den aktuellen Messungen automatisch Vorhersagen über den Gerätezustand zu treffen.

Für die fortlaufende Aktualisierung der Datenbasis und die Integration neuer Geräte, sollte eine WebApp entwickelt werden. Mit dieser WebApp können neue Geräte und deren spezifische Stromwerte erfasst werden. Dies gewährleistet, dass das ML-Modell bei einer erneuten Verbindung in der Lage ist, auch diese neu hinzugefügten Geräte korrekt zu identifizieren und deren Zustand zu bestimmen.

Nach der Implementierung dieser Maßnahmen steht ein System zur Verfügung, das effizient und zuverlässig den Zustand von Geräten anhand ihrer Stromwerte erkennt.

4.2 Benutzeroberfläche (User Interface)

Die Visualisierung und Interaktion mit den Daten des Funksensors sind für den Endbenutzer entscheidend. Daher wurde eine WebApp entwickelt, die darauf abzielt, diese Daten auf eine nutzerfreundliche und intuitive Weise darzustellen.

Die WebApp teilt sich in zwei Hauptbereiche, oder "Tabs":

View Device:

Dieser Bereich ermöglicht dem Nutzer, die aktuellen Daten des Funksensors in Echtzeit zu betrachten. Die dargestellten Informationen umfassen:

- Den aktuellen Stromwert des Sensors, der alle 5 Sekunden von TTN aktualisiert wird.
- Den aktuellen Status des Geräts, der durch eine bestimmte Farbe visualisiert wird.
- Den Namen des an den Sensor angeschlossenen Geräts.

Zur intuitiven Erkennung des Gerätezustands wird jedem Zustand eine spezifische Farbe zugewiesen. Die folgende Tabelle veranschaulicht die Zuordnung von Zustand und Farbe:

Tabelle 2: Zustand und Farbe Zuordnung

| Zustand | Farben |
|---------|--------|
| ON | Grün |
| OFF | Rot |
| Betrieb | Blue |
| Idle | Gelb |

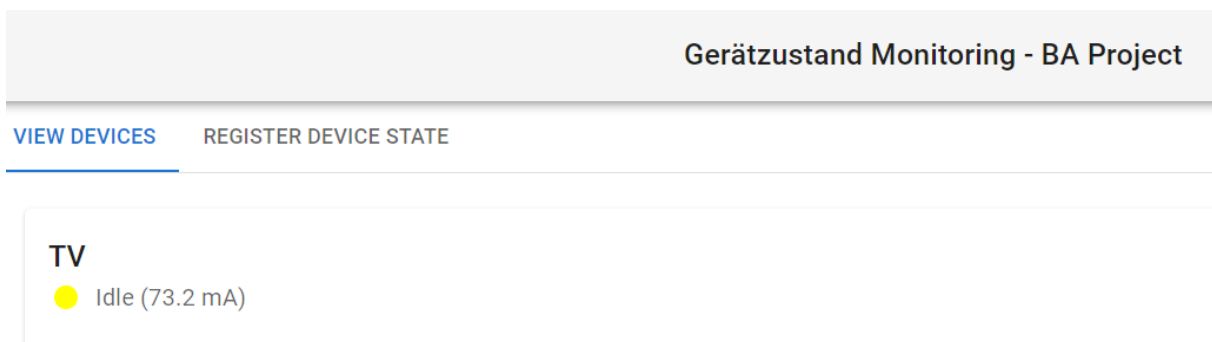


Abbildung 8: Webapp Zustand Darstellung

Register Device:

In diesem Bereich können neue Geräte ins System eingeführt und registriert werden. Hier müssen folgende Felder ausgefüllt werden:

- Name des Gerätes
- Aktueller Zustand des Geräts
- Typ des Geräts (Siehe Tabelle 7)

Diese Informationen tragen dazu bei, das KI-Modell effizienter zu trainieren und eine genauere Interpretation des Gerätezustands zu gewährleisten.

Zusammengefasst bietet diese Webapp eine intuitive Benutzeroberfläche, die dem Endbenutzer erlaubt, die Zustände und Daten der angeschlossenen medizinischen Geräte leicht zu interpretieren und zu verwalten. Abbildung 9 zeigt ein Beispiel der Registrierung eines Ultraschallgeräts.

The screenshot shows a web application interface for 'Gerätzustand Monitoring - BA Project'. It features two tabs: 'VIEW DEVICES' and 'REGISTER DEVICE', with the latter being the active tab. Below the tabs is a form with four input fields, each with a label and a value: 'Device Name' (Ultraschall), 'Device State' (ON), 'Current' (2563), and 'Type' (3). A blue 'REGISTER' button is positioned at the bottom of the form.

Abbildung 9: Registrierung von Geräten

4.3 Effiziente Datenübertragung durch Reduzierung der Datenmenge

Ein entscheidender Aspekt zur Energieeinsparung im Monitoring-System ist die Minimierung der Datenmenge, die durch den Funksensor übertragen wird. Durch eine effiziente Datenreduktion kann nicht nur Energie gespart werden, sondern auch die Übertragungsgeschwindigkeit und Zuverlässigkeit des Systems gesteigert werden.

Einer der Ansätze zur Datenreduktion besteht darin, die gemessenen Stromwerte im Mikrocontroller von einem Dezimal- zu einem Hexadezimalformat zu kodieren. Dieser Vorgang erlaubt es, zwei Zahlen innerhalb eines Bytes zu speichern. Das bedeutet, dass für die Übertragung der Stromwerte insgesamt nur 3 Byte benötigt werden, was eine signifikante Reduzierung im Vergleich zu nicht-kodierten Daten darstellt. Wenn die Konvertierung der 3 Byte nicht erfüllt ist, werden die fällende Byte mit null ausgefüllt.

Der Hintergrund dieser Kodierungsentscheidung liegt in der Beschaffenheit der gemessenen Stromwerte. Es wurde angenommen, dass der maximale Stromwert nicht mehr als sechs Ziffern umfasst. Unter dieser Annahme ergibt die Verwendung von 3 Byte für die Kodierung der Stromwerte Sinn.

Zusätzlich zu dieser Methode wurde eine zeitliche Auflösung von 5 Sekunden gewählt, anstelle einer kontinuierlichen Messung im Sekundentakt. Dies trägt ebenfalls dazu bei, die Datenmenge zu reduzieren und das System energieeffizienter zu gestalten, da nicht jede Sekunde Daten übertragen werden müssen. Somit werden nur relevante Daten in sinnvollen Intervallen erfasst und übermittelt, ohne Kompromisse bei der Genauigkeit oder Zuverlässigkeit des Monitoring-Systems einzugehen.

4.4 Einsatz eines KI-Modells für Zustandsmonitoring

Ein KI-Modell ist im Bereich des Zustandsmonitorings von medizinischen Geräten von entscheidender Bedeutung. Seine Hauptfunktion ist die effiziente und präzise Erkennung von Statusänderungen sowie das Aufspüren von Anomalien. Ein solches Modell erkennt automatisch verschiedene Zustände von unterschiedlichen Geräten, selbst wenn diese Geräte noch nicht zuvor getestet wurden. Die Funktionsweise des KI-Modells basiert auf neuronalen Netzwerken, welche in der Lage sind, die verschiedenen Betriebszustände der Geräte zu identifizieren.

4.5 Anlernphase der KI-Model

Ein KI-Modell erfordert für seine optimale Funktion eine Trainingsphase, in der es mit ausreichend Daten versorgt wird. Für diese Arbeit wurden unterschiedliche Stromwerte von verschiedenen medizinischen Geräten in Bezug zu deren Zuständen erfasst nämlich: Ultraschall, Laufband, Monitor, Cardio Maschine, Bett Lampe und Tisch Lampe.

Ein Beispiel: Während der Datenaufnahme im Labo bei einem Ultraschallgerät im Idle-Modus könnten bis zu neun verschiedene Stromwerte erfasst werden, um dem KI-Modell eine präzise Zustandserkennung zu ermöglichen. Basierend auf diesen Daten kann das KI-Modell genau bestimmen, in welchem Zustand sich ein bestimmtes Gerät befindet. Um eine umfassende Erkennungsfähigkeit für verschiedene Gerätezustände zu gewährleisten, ist es unerlässlich, eine Vielzahl von Geräten zu testen und eine umfangreiche Datensammlung anzulegen. Das trainierte Modell kann so nicht nur Zustände von bekannten, sondern auch von bisher nicht registrierten Geräten erkennen und eventuell auftretende Anomalien identifizieren.

4.6 Energieeffizienz der Funksensor

Ein zentrales Element in dem Funksensor-Design ist das verwendete LoRa-Modul, konkret das Grove-Wio-E5. LoRa-Technologie ist bekannt für seine beeindruckenden Energieeffizienzmerkmale, die es ideal für Anwendungen machen, die eine langanhaltende Batterielebensdauer erfordern, wie es in einem Krankenhausumfeld der Fall ist [14].

Die bemerkenswerte Energieeffizienz von LoRa zeigt sich insbesondere in seinem Ruhezustand, bei dem der Stromverbrauch lediglich zwischen $10mA$ und $100na$ (*Nanoamper*) liegt [18]. Dies ermöglicht es dem Modul, den Großteil seiner Betriebszeit im Ruhezustand zu verbringen, wodurch der Batterieverbrauch drastisch reduziert wird. Selbst wenn das Modul aktiv ist und Signale empfängt, beträgt der Stromverbrauch nur $10mA$, was im Vergleich zu vielen anderen Kommunikationstechnologien immer noch sehr niedrig ist [14]. Die Kombination dieser Faktoren führt dazu, dass die allgemeine Lebensdauer einer LoRa-Batterie bis zu 5 Jahre beträgt. Dies bedeutet, dass die hier verwendete Funksensor, der das Grove-Wio-E5 LoRa-Modul verwendet, über lange Zeiträume hinweg betrieben werden kann, ohne dass die Batterie gewechselt werden muss.

Abbildung 10 zeigt der niedrige Energieverbrauch von LoRa im Vergleich zu anderen drahtlosen Technologien.

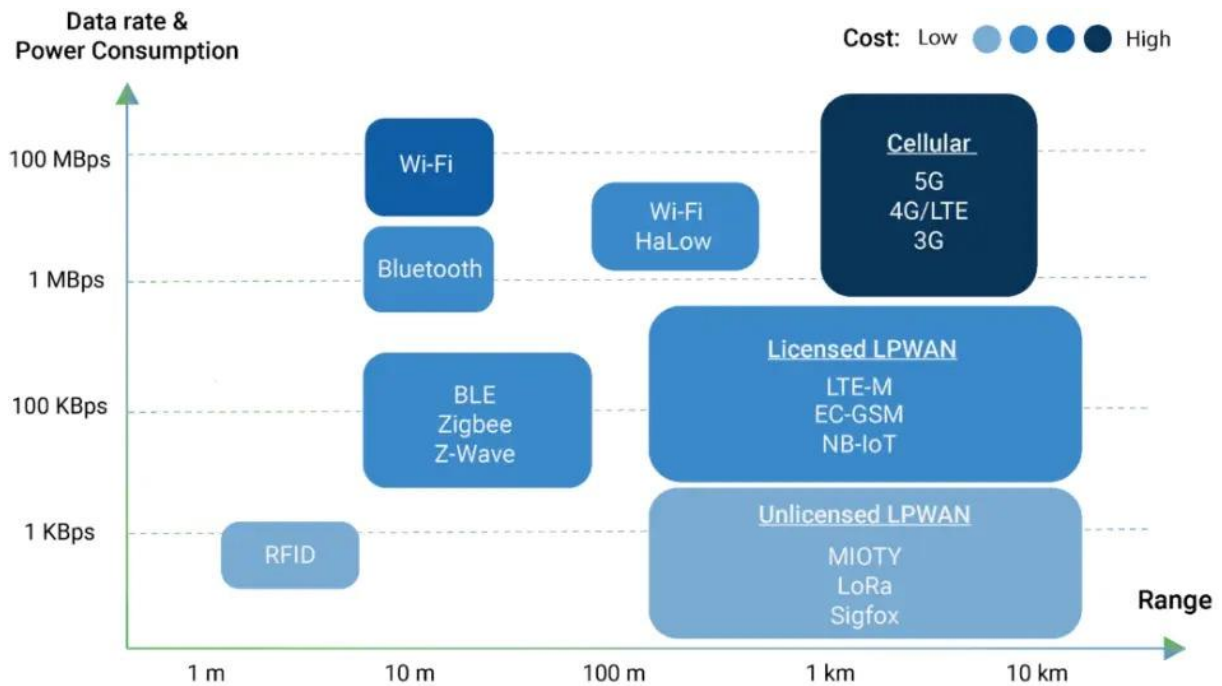


Abbildung 10: Energieverbrauch eine LoRa

(Quelle Moko Technology Limited 2022)

Das Hervorheben dieser Energieeffizienz ist von besonderer Bedeutung im Kontext eines modernen Krankenhauses, wo eine unterbrechungsfreie Überwachung und Minimierung von Wartungsintervallen von entscheidender Bedeutung ist. Durch die Auswahl von LoRa und insbesondere des Grove-Wio-E5-Moduls für unseren Funksensor wurde sichergestellt, dass das System nicht nur zuverlässig und präzise, sondern auch außerordentlich energieeffizient ist.

5 Ausarbeitung und Realisierung der Lösung

Für eine erfolgreiche Umsetzung des Lösungskonzepts sind eine strukturierte Vorgehensweise und spezifische Schritte erforderlich, um die vorgeschlagenen Anforderungen zu erfüllen.

5.1 Verwendete Komponenten

LoRaWAN Gateway

Ein LoRaWAN-Gateway spielt eine zentrale Rolle in LoRaWAN Architektur. Es dient als Brücke zwischen den Endgeräten (z. B. Sensoren) und dem Netzwerkservers:

In diese Arbeit wurde das DraginoLPS8-V2 Gateway verwendet und besetzt diese technischen Merkmale. (siehe Anhang)



Abbildung 11: Dragino LoRaWAN Gateway

(Quelle: Dragino Technology Co., LTD. 2010-2023)

Microcontroller ST - NUCLEO-F767ZI

Der ST NUCLEO-F767ZI ist ein Entwicklungsboard von STMicroelectronics, basierend auf dem STM32F767ZI Mikrocontroller mit einer ARM Cortex-M7 CPU.

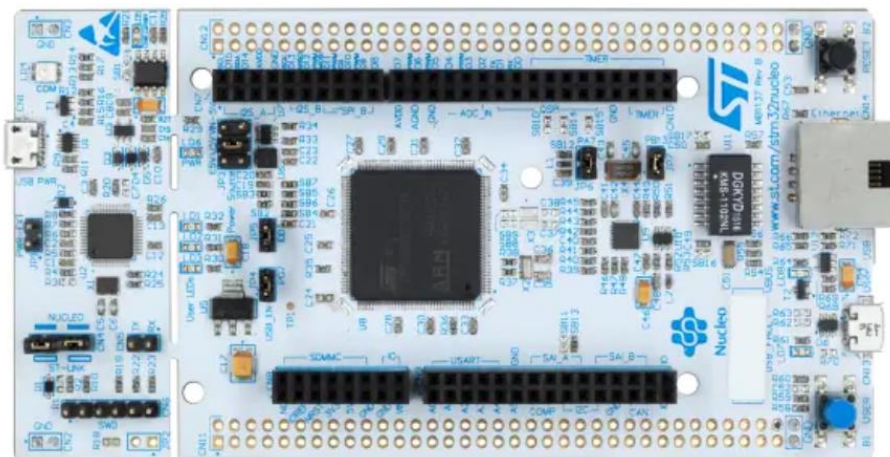


Abbildung 12 : Microcontroller ST - NUCLEO-F767ZI

(Quelle: STMicroelectronics 2023)

LoRa Module (Grove-Wio-E5):

Das Grove-Wio-E5 ist ein LoRa-Modul, das speziell für drahtlose Kommunikationsanwendungen entwickelt wurde. Das Modul bietet nicht nur die Fähigkeit, Daten über große Entfernungen zu senden und zu empfangen, sondern bietet auch die Vorteile eines geringen Energieverbrauchs [24]. Das Grove-Wio-E5 eignet sich insbesondere für Projekte, bei denen eine zuverlässige Datenübertragung über größere Distanzen und unter schwierigen Bedingungen erforderlich ist, wie sie beispielsweise in vielen IoT-Anwendungen auftreten. Was der Fall für diese Arbeit ist. [24]

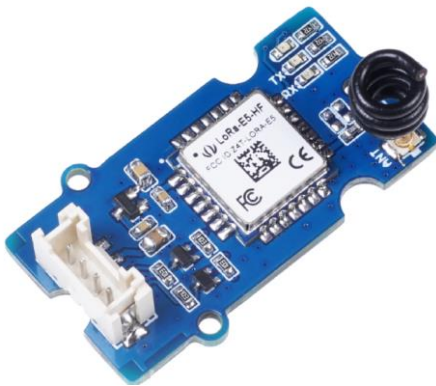


Abbildung 13: Grove-Wio-E5 (LoRa Module)

(Quelle: Seeed Studio 2023)

Tabelle 3: Grove-Wio-E5 Spezifikation und Parameters

| | |
|--------------------|-------------------------------|
| Spannung | 3,3 V – 5 V |
| Ausgangsleistung | Bis zu +20dBm bei 3,3V |
| Arbeitsfrequenz | 868/915MHz |
| Protokoll | LoRaWAN |
| Empfindlichkeit | -116,5 dBm~-136dBm |
| Modulation | LoRa, (G)FSK, (G)MSK und BPSK |
| Strömung | Nur 60uA im Ruhemodus |
| Größe | 20*40mm |
| Betriebstemperatur | -40°C ~ 85°C |

Stromsensor-Modul

Der Stromsensor-Modul gehört zur Grove-Familie und basiert auf dem TA12-200 Stromwandler, der in der Lage ist, großen Wechselstrom in kleine Amplituden umzuwandeln. Der Sensor ist ein Transformator und kann den gemessenen Strom mithilfe des Übersetzungsverhältnisses und des angeschlossenen Lastwiderstands berechnen. Es könnte verwendet, um Wechselströme von bis zu 5A zu testen. Das Modul wird in diese Arbeit verwendet für Medizinische Gerätezustandsüberwachung. Die folgende Tabelle 4 zeigt die elektrische Eigenschaft der Sensor [25].

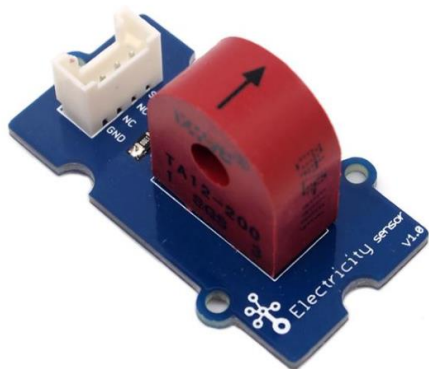


Abbildung 14: Grove Elektrizitätssensormodul

(Quelle: Seeed Studio 2023)

Tabelle 4: Elektronische Eigenschaften der Stroms Sensor

| Artikel | Min | Norm | Max | Einheit |
|---------------------------|-----|--------|------|--------------------|
| Transformation Verhältnis | - | 2000:1 | - | - |
| Eingangsstrom | 0 | - | 5 | Ein |
| Ausgangsstrom | 0 | - | 2,5 | Mutti |
| Abtastwiderstand | - | 800 | - | Ω |
| Abtastspannung | 0 | - | 2 | V |
| Arbeitsfrequenz | 20 | - | 20k | HZ |
| Nichtlineare Skala | - | - | 0,2% | - |
| Phasenverschiebung | - | - | 5 | - |
| Betriebstemperatur | -55 | - | 85 | $^{\circ}\text{C}$ |
| Durchschlagsfestigkeit | - | 6 | - | KVAC/1min |

5.2 Aufbau der Funksensor

Der Funksensor wurde mit Hilfe der Tabelle 1 Komponenten aufgebaut. Diese wurde wie folg verbündet.

Tabelle 5: Verbindung der Strom Sensor mit der Mikrokontroller

| Strom Sensor | Mikrocontroller | Farben |
|-----------------|-----------------|---------|
| Nicht verbunden | 5V | Rot |
| GND | GND | Schwarz |
| Nicht verbunden | - | Wieß |
| Signal | PA3 (bzw. A0) | Gelb |

Tabelle 6: Verbindung der LoRa Modul zu dem Board

| Board | LoRa Modul |
|-----------|------------|
| PD 6 (RX) | TX |
| PD 5(TX) | RX |
| GND | GND |
| VCC(3V) | VCC |

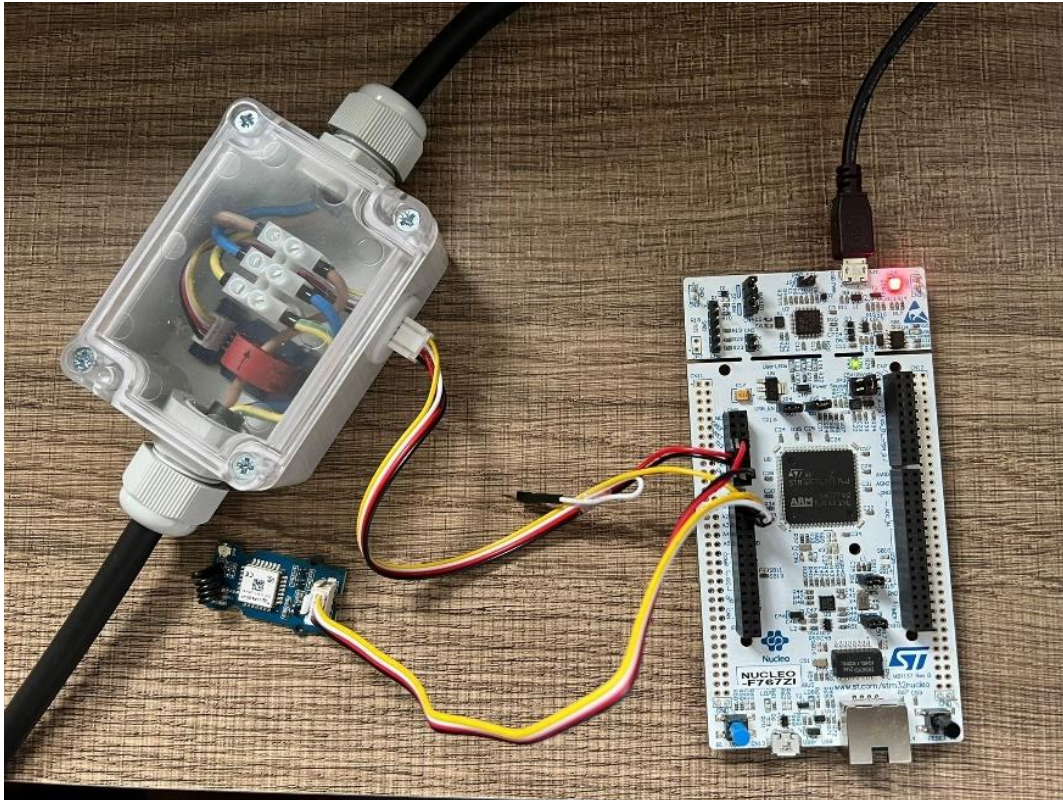


Abbildung 15: Aufbau der Funksensor

5.3 Umsetzung der Funck Sensor bzw. Projekt

Für die Funktionsweise des aufgebauten Funksensors waren verschiedene Vorbereitungsschritte sowie eine passende Programmierung erforderlich.

Zu Beginn wurden die Pin-Belegungen des LoRa-Moduls (PD5 und PD6) sowie des Stromsensors (PA3) initialisiert. Dies ermöglicht eine Kommunikation zwischen dem Mikrocontroller und den beiden Modulen.

Alle notwendigen Bibliotheken wurden im Code importiert (siehe Code im Anhang).

Eine Funktion namens „loraConnect“ wurde implementiert, um sämtliche relevante Informationen bezüglich des LoRa-Moduls zu erhalten und zu konfigurieren. Hierzu zählen Daten wie AppEUI, AppKey und NWKSKEY. Nachdem diese Informationen abgerufen wurden, wurden sie im Code auskommentiert, da sie nur einmalig benötigt werden. Diese Funktion dient ebenfalls dazu, das LoRa-Modul mit dem TTN-Server zu verbinden, was mit dem AT+ JOIN-Befehl realisiert wird, der nur einmalig ausgeführt werden kann.

Für die Erfassung des Stromwertes mit dem Stromsensor wurde die Funktion „analogRead()“ verwendet. Eine Funktion „getMaxValue()“ wird implementiert, um die maximalen Stromwerte der fließende Strom innerhalb von 1000 ms zu nehmen.

Um den genauen maximalen Stromwert zu erfassen, wird innerhalb einer Schleife in der Funktion „getMaxValue()“ kontinuierlich überprüft, ob der aktuell gelesene Sensorwert den bisher festgehaltenen Maximalwert übersteigt. Ist dies der Fall, wird der neue Wert als Maximalwert gespeichert.

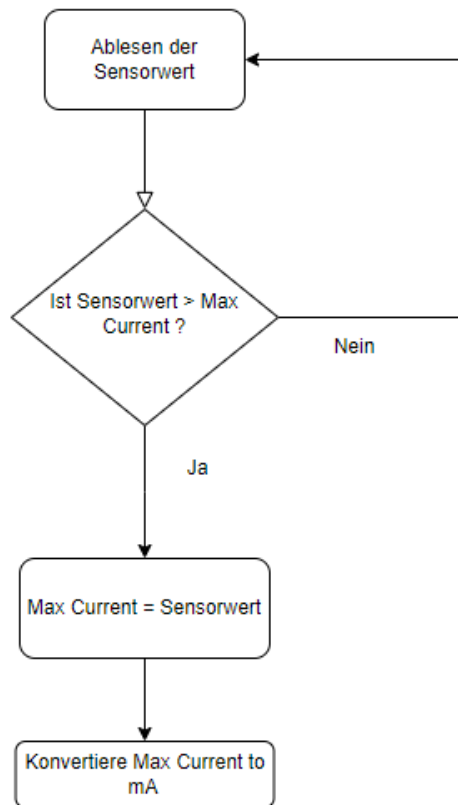


Abbildung 16: Ablesen Workflow der maximale stromwert

Der erfasste Höchstwert wird dann mittels folgender Formel in Milliampere (*mA*) umgewandelt:

$$\text{Max Current} = \text{sensor value} / 1024 * 5 / 800 * 2000000 \quad [25]$$

Nach der Erfassung des Stromwerts werden diese Daten über die „loraSend“-Funktion an den TTN gesendet.

Aufschlüsselung der einzelnen Bytes

Der maximale Stromwert wird in Dezimalform mit Nachkommastellen dargestellt. Um diesen Wert in eine hexadezimale Darstellung zu konvertieren, muss er zuerst in einen Ganzzahlwert umgewandelt werden. Da der maximale Stromwert auf zwei Dezimalstellen begrenzt ist, wird er mal 100 multipliziert. Diese Multiplication entfernt die Dezimalstellen und konvertiert den Wert in einen Integer. Anschließend kann dieser Integer-Wert problemlos in ein Hexadezimalformat umgewandelt werden.

5.1 Verbindung an The Things Network (TTN)

Um sich an der TTN zu verbinden und mit der Lora Modul damit zu kommunizieren, soll folgende Schritte gemacht werden.

5.1.1 Verbindung der LoRa Modul mit der TTN

Zunächst soll auf der Webseite von The Things Network ein Account über die TTN Community Edition erstellt werden. Nach erfolgreicher Registrierung navigiert man zum Bereich „Console“ → „Applications“ und richtet die Cluster sowie den Standort des LoRa-Moduls ein.

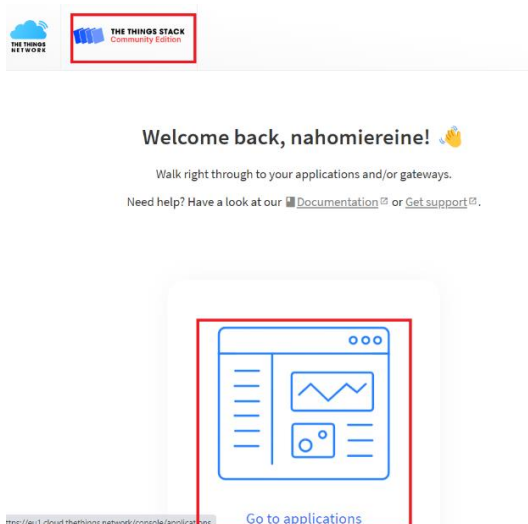


Abbildung 17: TTN Application

In einem nächsten Schritt navigiert man zum Tab "Devices" und wählt die Option "Device hinzufügen", um das LoRa-Modul zu registrieren. Dafür sollen die benötigten Informationen wie „Join EUI, DevEUI und AppKey“ in die vorgesehenen Felder eingetragen werden. Der Device EUI kann über den Konsolenbefehl „println“ ermittelt werden:

AT+ID=AppEui.

AT+ID=DevEui

AT+KEY=AppKey

Nach dem erfolgreichen Hinzufügen des Geräts sollen die von TTN bereitgestellten Schlüssel und Einstellungen, einschließlich „Application EUI, Device EUI und App Key“, zur Konfiguration des LoRa-Moduls genutzt werden.

Auch wenn das LoRa-Modul mit den angegebenen Einstellungen erfolgreich bei The Things Network registriert wird, soll es für den aktiven Betrieb und zur Datenübermittlung noch mit einem Gateway verbunden werden.

5.1.2 Verbindung des Gateways mit der TTN

Zuerst soll das Gateway mittels eines Ethernet-Kabels mit dem Internet verbunden werden. Nachdem das Gateway online ist, soll es eingeschaltet und in den Anmelde-Modus versetzt werden.

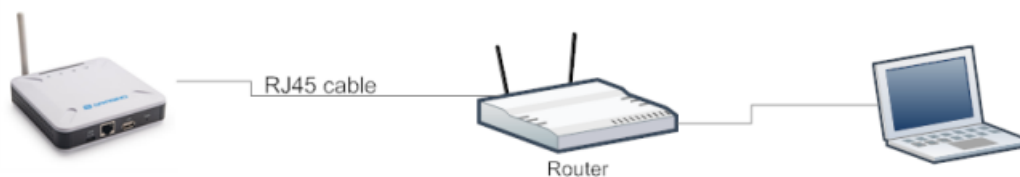


Abbildung 18: Verbindung des Gateways an das Internet

(Quelle Dragino Technology Co., LTD, 2010-2023)

Da das Gateway direkt mit dem Internet-Router verbunden ist, soll die Gateway-URL über den Internet-Router genommen werden. Dies erfolgt, indem man die mit dem Router verbundenen Geräte durchsucht und den Dragino Gateway identifiziert (siehe Abbildung 19). Durch einen Klick auf den Gateway-Namen soll die Gateway-URL aufgerufen werden.

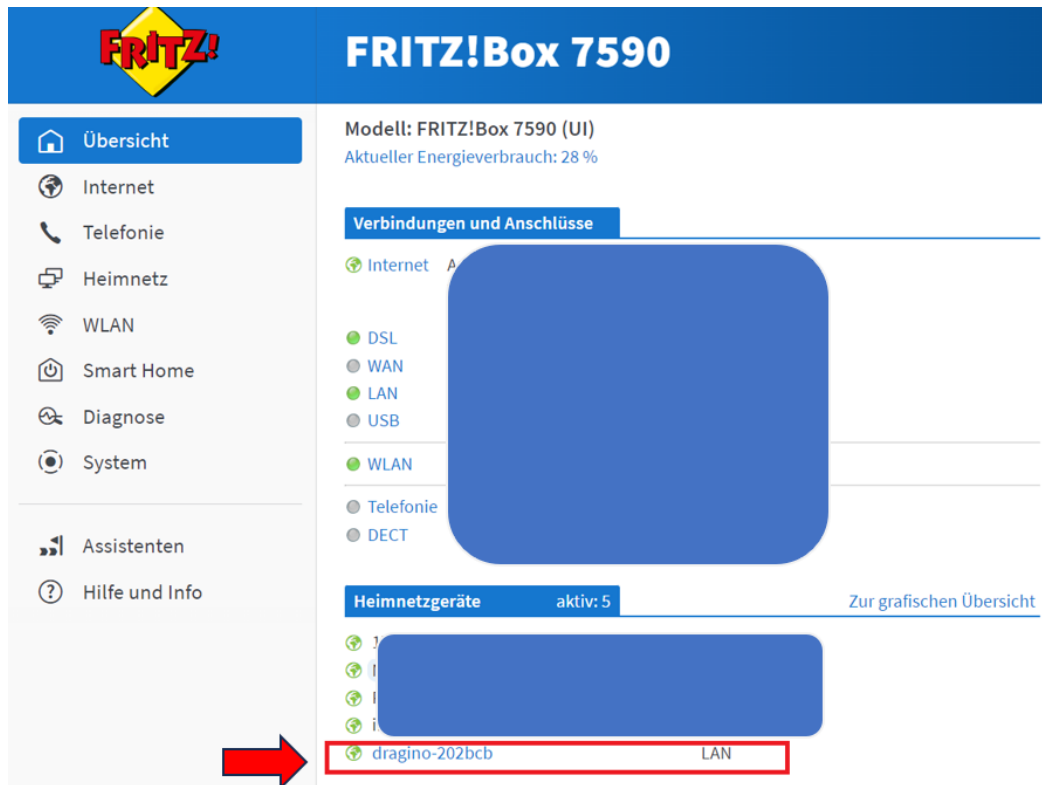


Abbildung 19: Internet-Router verbundene Geräte bzw. Dragino Gateway

Um das Gateway EUI zu bekommen, navigiert man in das Gateway URL über „LoRaWAN-→LoRAWAN Semtech UDP“.

LoRaWAN Configuratio

LoRaWAN Semtech UDP
LoRaWAN Basic Station

General Settings

Email: dragino-202bcb@dragino.com

Gateway EUI: 02819bdf202bcb

Primary LoRaWAN Server

Service Provider: The Things Network V3

Server Address: eu1.cloud.thethings.network

Uplink Port: 1700

Downlink Port: 1700

Secondary LoRaWAN Server

Service Provider: Disable

Abbildung 20: Dragino Gateway Konfiguration

Um das Gateway mit The Things Network (TTN) zu verknüpfen, soll auf der TTN-Webseite zum Bereich "Gateways" navigiert werden. Hier soll die Option "Gateway hinzufügen" ausgewählt werden. Die erforderlichen Informationen, hauptsächlich die Gateway-ID und gegebenenfalls andere spezifische Daten, sollen in die vorgesehenen Felder eingetragen werden. Nach der Konfiguration und Registrierung des Gateways sollte es erfolgreich mit TTN verbunden werden.

Nachdem das Gateway eingerichtet und mit TTN verbunden ist, soll das LoRa-Modul aktiviert werden.

Damit das LoRa Modul mit dem TTN kommunizieren kann, muss der Befehl „**AT + JOIN**“ in das C++ Programm geschrieben werden, was dann nur einmalig aufgerufen werden soll.

Das LoRa Module wird nach verfügbaren LoRa-Gateways in seiner Reichweite suchen und sich automatisch mit dem zuvor eingerichteten Gateway verbinden. Nach erfolgreicher Verbindung wird sowohl im Terminal als auch im TTN gezeigt, dass das Modul verbunden und aktive ist.

```
+JOIN: Start
+JOIN: NORMAL
```

Abbildung 21: Erfolgreiche Verbindung an TTN

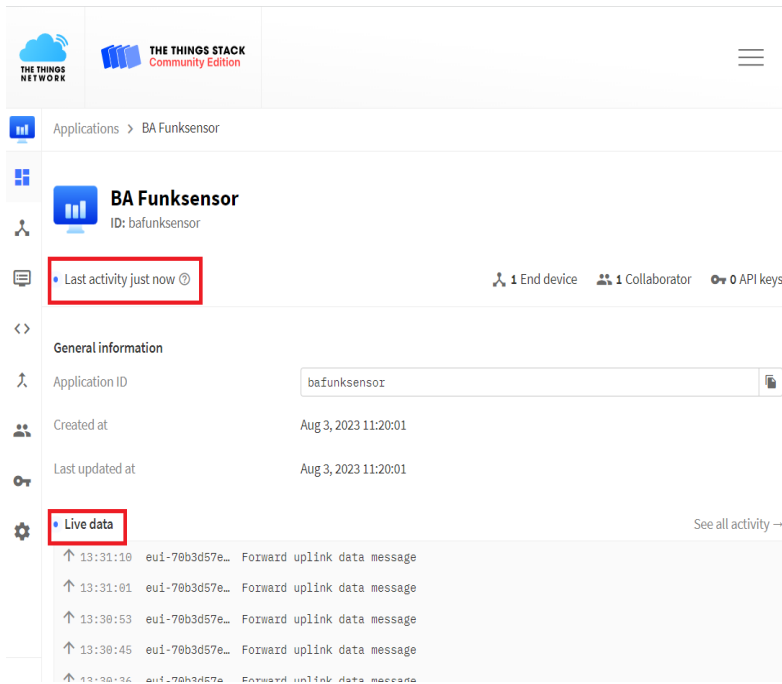


Abbildung 22: Verbundenen LoRa Modul an TTN

Sobald die Verbindung hergestellt ist, kann das LoRa-Modul Daten über das Gateway an TTN senden und von dort empfangen.

Durch diese Schritte sollen sowohl das Gateway als auch das LoRa-Modul erfolgreich mit The Things Network verbunden werden, und es wird ermöglicht, Daten über das LoRaWAN-Netzwerk zu senden und zu empfangen.



Abbildung 23: Verbundene Gateway ans Internet und an LoRaWAN

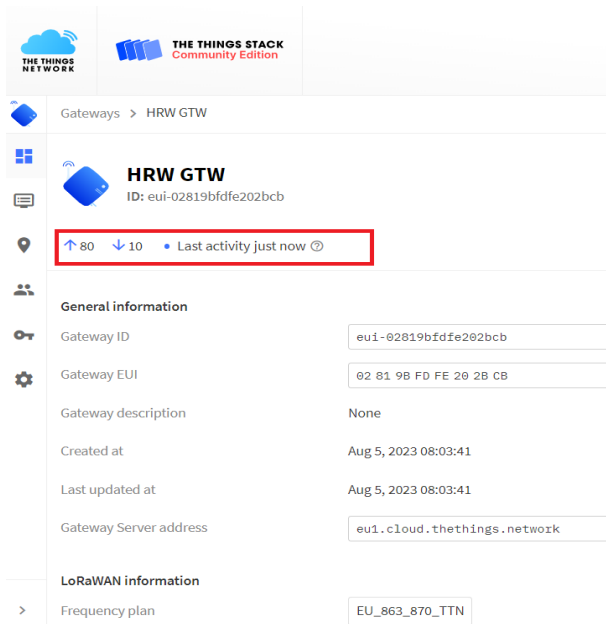


Abbildung 24: Verbundene Gateway an TTN

5.1 Erstellung der Machine-Learning-Model (ML-Model)

Damit man ein geeignetes ML-Modell auswählt, müssen einige Kriterien erfüllt werden. Mit Hilfe der Abbildung 25 sieht man, wie das Modell genau ausgewählt wurde.

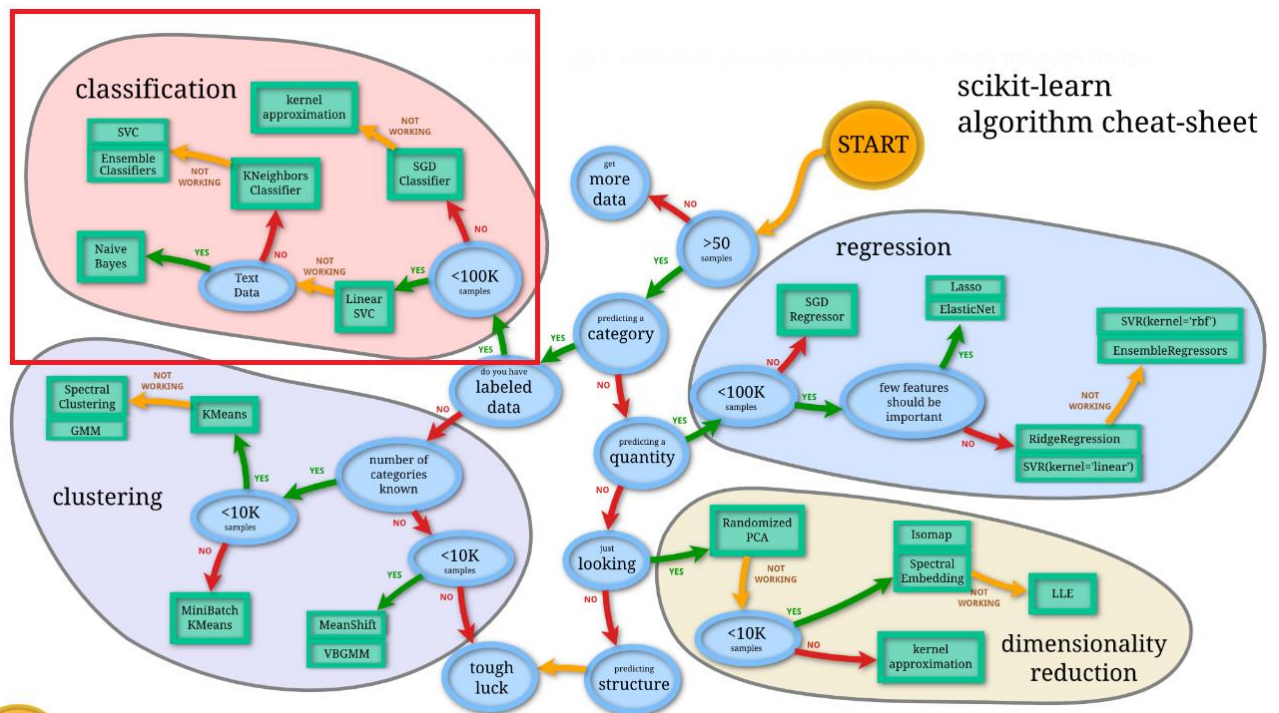


Abbildung 25: Auswahl der ML-Model

(Quelle: scikit-learn.org 2023)

Bevor man ein ML-Modell erstellen kann, ist es notwendig, ausreichend Daten zu sammeln. In dieser Arbeit wurden diverse Stromwerte von verschiedenen Geräten (Ultraschall, Laufband, Monitor, Cardio Messgerät, Lampen, Ventilator) in unterschiedlichen Zuständen gesammelt und in einer CSV-Datei gespeichert.

Zu Beginn wurden wesentliche Bibliotheken und Abhängigkeiten in **Python** importiert (siehe Anhang), darunter Pandas, Sklearn und Pickle. Anschließend wurde die CSV-Datei mit einem Semikolon als Trennzeichen in das Programm importiert, um die Daten strukturiert und korrekt darzustellen (in Tabellenform).

Die Daten wurden anschließend so aufgeteilt, dass 20% für Tests und 80% für das Training des ML-Modells verwendet wurden. Als ML-Modell wurde der "**RandomForestClassifier**" aus der Scikit-Learn-Bibliothek gewählt.

Die Daten wurden mit diesem Modell verknüpft, wobei das Modell die X- und Y-Werte übernahm. Die X-Werte repräsentieren die Merkmale (Features), die dem Modell helfen, Vorhersagen zu treffen, während Y die Auszeichnungen (Labels) repräsentiert, die die Ergebnisse darstellen. In unserem Fall repräsentiert X den Gerätetyp und den maximalen Strom des Geräts, während Y den Status darstellt (.Tabelle 7)

Schließlich wurde das Modell mit den gesammelten Daten trainiert und danach in unser Backend exportiert.

5.2 Decodern der Hexadezimal Daten in TTN

Um die empfangenen Daten über TTN interpretieren zu können, war es notwendig, einen Decoder Code im Bereich "Payload Formatters" bzw. "Uplink Formatter" in JavaScript zu schreiben. Der codierte maximale Stromwert wurde dabei von 3 Byte Hex wieder auf Dezimalwert decodiert.

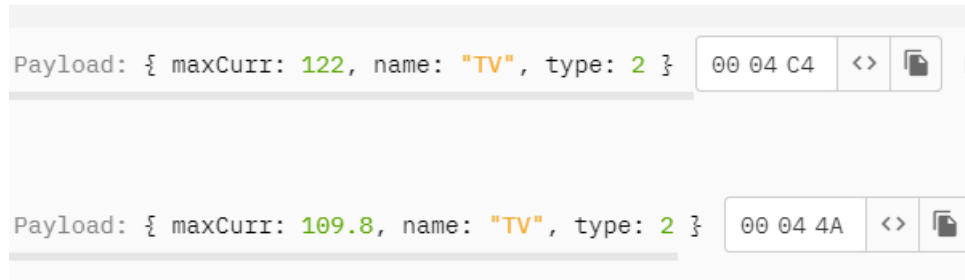


Abbildung 26: Payload Decodierte Beispiel

5.2.1 Zustände von Geräten

Um dem KI-Modell zu ermöglichen, die verschiedenen Zustände zu interpretieren, wurden unterschiedliche Typen für die verschiedenen Zustände der Geräte zugewiesen (siehe Tabelle 7). Dabei ist zu beachten, dass nicht alle Geräte dieselben Zustände aufweisen können. Einige Geräte haben lediglich die Zustände AN oder AUS, während andere spezielle Zustände wie "Idle" besitzen.

Tabelle 7 Zustand und Status von Geräten Zuweisung

| Type des Geräts | Status |
|-----------------|---------------------------|
| 1 | ON/OFF |
| 2 | ON/OFF/Idle |
| 3 | ON/OFF/Idle/Betrieb |
| 4 | ON/OFF/Kein Idle/ Betrieb |

ON: Der Zustand "ON" bedeutet, dass das Gerät an die Steckdose angeschlossen ist und Strom fließt.

OFF: Der Zustand "OFF" tritt auf, wenn das Gerät entweder komplett ausgeschaltet ist oder über den Ausschaltknopf des Geräts deaktiviert wurde.

Idle: Dieser Zustand zeigt an, dass das Gerät eingeschaltet ist, aber derzeit nicht aktiv genutzt wird. Einige Geräte haben Anzeigen, die signalisieren, dass sie eingeschaltet sind, selbst wenn sie über ihren Ausschaltknopf deaktiviert wurden. Ein Beispiel hierfür ist ein Monitor: Er ist im "ON"-Zustand, wenn die On/OFF-Anzeige blau leuchtet und im "Idle"-Zustand, wenn sie orange leuchtet.

Betrieb: Der "Betrieb"-Zustand zeigt an, dass das Gerät aktiv in Gebrauch ist oder eine spezifische Aktion durchführt. Zum Beispiel würde ein Ultraschallgerät, das gerade an einem Patienten verwendet wird, in den "Betrieb"-Modus wechseln. Während dieses Zustandes verbraucht das Gerät in der Regel mehr Strom als im "Idle"-Modus.

5.3 WebApp

Um die Ergebnisse übersichtlich zu präsentieren und die Registrierung neuer Geräte zu ermöglichen, wurde eine WebApp entwickelt. Sie teilt sich in ein Frontend und ein Backend auf, welche mit den Technologien React und Python umgesetzt wurden. (siehe Anhang)

Das Frontend und Backend werden lokal auf dem Computer mit Hilfe des Software „Ngrok“ gehostet.

Applications > BA Funksensor > Webhooks

| ID | Base URL |
|--------|---|
| webapp | https://e155-2003-ca-2f0c-2a00-156d-f4f1-266... |

Abbildung 27: Lokale host im Webhook

Das Frontend ist in zwei Tabs unterteilt: „View Device“ (Abbildung 8) und „Register Device State“ (Abbildung 9). Während „View Device“ den aktuellen Stromwert und den Zustand des an den Funksensor angeschlossenen Geräts anzeigt, dient „Register Device State“ zur Eingabe und Registrierung neuer Geräteinformationen.

| Gerätzustand Monitoring - BA Project | |
|--------------------------------------|-----------------------|
| VIEW DEVICES | REGISTER DEVICE STATE |
| TV ● Idle (73.2 mA) | |

Abbildung 28: Webapp Zustand Darstellung

Um eine bessere Übersicht zu gewährleisten, wurden den verschiedenen Zuständen jeweils eine Farbe zugeordnet. (Sehe Tabelle 2)

Die WebApp stellt nicht nur die Gerätezustände visuell dar, sondern visualisiert ebenso den aktuellen Stromwert des an den Funksensor angeschlossenen Geräts.

Das Backend dient als Schnittstelle zwischen TTN und dem Frontend. Es empfängt die Stromwerte von TTN und leitet diese, zusammen mit den vom ML-Modell vorhergesagten Zuständen, an das Frontend weiter. Die Kommunikation zwischen Frontend und Backend erfolgt über das HTTP-Protokoll. Wenn das Frontend eine Anfrage stellt, bezieht das Backend die benötigten Daten von TTN und sendet sie im JSON-Format an das Frontend zurück. Dort werden sie dann entsprechend aufbereitet und dargestellt.

Der Tab „Register Device State“ bietet verschiedene Eingabefelder an, in die Informationen über neue Geräte eingetragen werden können. Dazu gehören der Name des Geräts, sein Zustand und sein Typ. Diese Informationen sind essenziell, um die CSV-Datei mit dem der ML-Model arbeitet zu aktualisieren und das ML-Modell kontinuierlich zu verbessern.

5.4 Ablauf der Übertragung, Verarbeitung und Darstellung der Stromdaten

Im Abbildung 29 steht ein Workflow-Diagramm, um einen großen Überblick zu verschaffen, wie die Stromdaten übertragen, verarbeitet und dargestellt werden

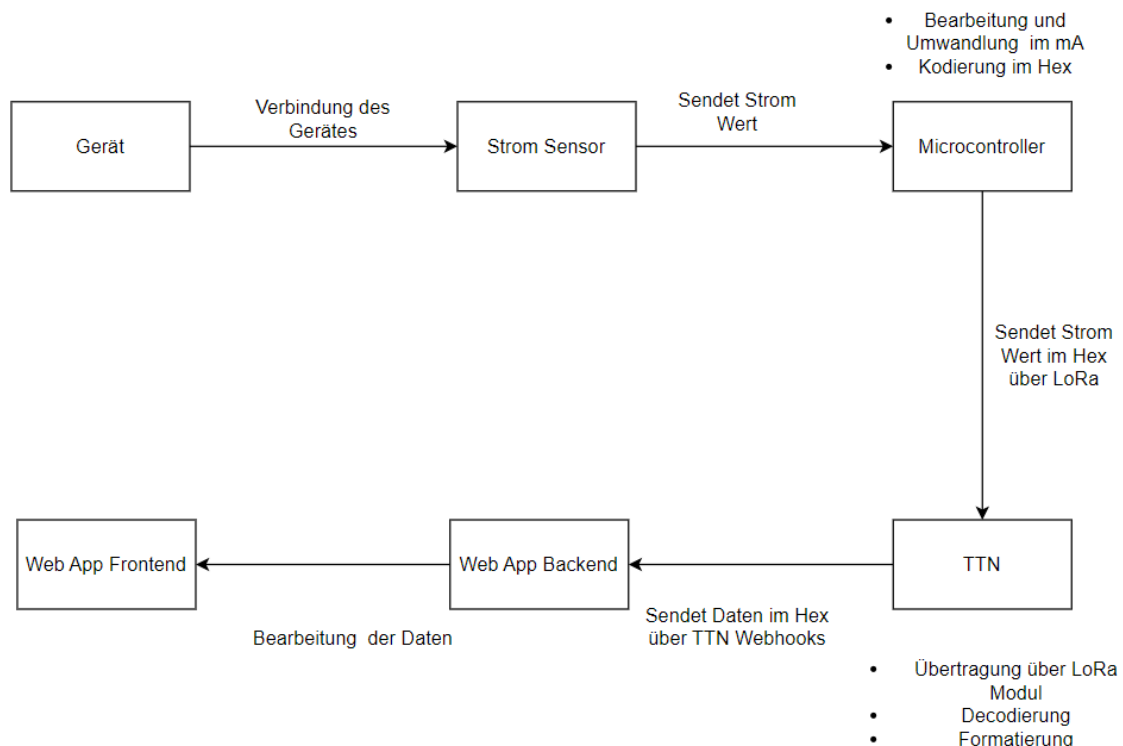


Abbildung 29: Workflow der Ausarbeitung

Übertragung und Darstellung der Stromdaten:

Der im Mikrocontroller programmierte Code wandelt den Strom von Dezimalwert in einen Hexadezimalwert um, sodass die Daten für eine Übertragung über TTN geeignet sind. Daraufhin sendet der Mikrocontroller die Daten über das LoRa-Modul und LoRaWAN Gateway an TTN.

Im TTN werden die Daten entschlüsselt, formatiert und anschließend mittels HTTP (WebHooks) an das entwickelte Webapp-Backend weitergeleitet.

Das Backend verarbeitet die empfangenen Daten und bestimmt den Zustand des Geräts mit Hilfe eines trainierten Machine-Learning-Modells, genannt „RandomForestClassifier Sklearn“. Nach der Verarbeitung speichert das Backend die Daten und sendet sie an das Frontend weiter.

Im Frontend werden die Daten visualisiert, wobei angezeigt wird, welches Gerät verbunden ist, was sein aktueller Stromwert ist und in welchem Zustand es sich befindet.

6 Validierung und Test

In diesem Kapitel wird beschrieben, wie die verschiedenen Gerätezustände mithilfe des Funksensors und des KI-Modells erfasst wurden und welche Ziele dadurch erreicht wurden.

Der Entwurf und Programmieren des Funksensors mit den verwendeten Komponenten war erfolgreich.

Mithilfe des Funksensors war es möglich, den fließenden Strom der verbundenen Geräte erfolgreich abzulesen, indem unterschiedliche Stromwerte für die verschiedenen Gerätezustände gemessen wurden.

Die Übertragung der gemessenen maximalen Stromwerte erfolgte nach jeder 5 Sekunden, somit trägt es bei der Datenminimierung. Es wird weiter mit LoRa nur einen Wert gesendet, nämlich der maximale Wert. Andere Werte werden mit Hilfe dieses maximalen Stromwertes ermittelt.

Anomalieerkennung werden nicht erkannt, das könnte in der Zukunft implementiert werden.

Daten werden aktuell in einer Variable gespeichert und könnten durch eine optimierte Speicherlösung wie SQL gespeichert werden.

6.1.1 Bewertung der ML-Model

Nach dem Training wurde Informationen über die Genauigkeit des Modells gegeben, dabei werden drei Haupt Informationen gegeben:

- Die Präzision (Precision - **p**). Diese Zahl gibt an, welcher Anteil der vom Modell als Entitäten identifizierten Wörter tatsächlich Entitäten sind.
- Die Sensitivität (Recall - **R**). Diese Zahl gibt an, welcher Anteil der tatsächlichen Entitäten vom Modell erkannt wurde
- Der F1-Score (**F**). Dieser Wert ist das harmonische Mittel aus Präzision und Sensitivität und gibt eine einzelne Metrik an, die sowohl die Präzision als auch die Sensitivität berücksichtigt.

Nachdem Training, hatte das Model eine Genauigkeit von **96%** und eine Fehlerquote von **4%**.

Nachfolgend sind die spezifischen Metriken nach dem Training des ML-Modells dargestellt:

Tabelle 8: Genauigkeit nach dem Training der ML-Model

| Status | P (in %) | R (in %) | F(in%) |
|---------|----------|----------|--------|
| Betrieb | 91 | 100 | 95 |
| Idle | 100 | 100 | 100 |
| Off | 100 | 100 | 100 |
| On | 100 | 89 | 94 |

Dank des KI-Modells konnten die verschiedenen Zustände der verbundenen Geräte sowohl für schon bekannte Geräten als auch für unbekanntes Geräte automatisch erkannt werden. Die Erkennung der Zustände waren nicht immer richtig, denn manchmal konnte das Geräte im Idle Mode sein aber der ML Model zeigt, dass es im Betrieb Mode ist. Diese Ergebnisse wurden in der Webapp dargestellt, zusammen mit den gemessenen Stromwerten in mA.

Bei der visuellen Fehlererkennung während des Gerätetests konnte überprüft werden, ob der angezeigte Zustand in der Webapp mit dem tatsächlichen Zustand des Geräts übereinstimmt.

Die Validierung des KI-Modells stand dabei im Fokus. Zunächst wurden Geräte für Tests herangezogen, die bereits für das Training des KI-Modells verwendet wurden. Dadurch sollte sichergestellt werden, dass das Modell die Zustände korrekt erkennt. Anschließend wurden Geräte getestet, die nicht im Trainingsdatensatz enthalten waren, um die Genauigkeit und Präzision des Modells zu beurteilen.

7 Fazit und Ausblick

Das Projekt konnte in vielen Bereichen erfolgreich umgesetzt werden. Insbesondere die Programmierung des Funksensors sowie die Verbindung des LoRa-Moduls zu TTN verliefen problemlos. Auch die Darstellung der Gerätezustände dank des ML-Modells und der Webapp wurde erfolgreich realisiert. Zudem leistet dieses Projekt einen wichtigen Beitrag im Bereich Gerätemanagement im Krankenhaus.

Allerdings gab es auch Herausforderungen während der Projektumsetzung. Bei der Verbindung des LoRa-Moduls mit dem LoRa-Gateway und TTN traten ein paar Schwierigkeiten auf, die dann gelöst wurden.

In der Zukunft bieten sich verschiedene Möglichkeiten für Verbesserungen und Erweiterungen:

- **Optimierung des KI-Modells:** Durch das Testen einer größeren Anzahl von Geräten und das Sammeln von Daten kann die Effizienz und Genauigkeit des KI-Modells weiter gesteigert werden.
- **Einrichtung einer Datenbank:** Durch die Erstellung einer Datenbank für die Speicherung aller neuen Daten kann das KI-Modell kontinuierlich trainiert und verbessert werden.
- **Integration der Geolokalisierung:** Die Implementierung der Geolokalisierung im Funksensor ermöglicht es, den genauen Standort eines Geräts zu bestimmen. Dies könnte dazu beitragen, die Zeit, die benötigt wird, um ein Gerät zu finden, erheblich zu reduzieren.
- **Bewertungsfunktionalität in der Webapp:** Durch das Einbinden einer Bewertungsfunktion für das KI-Modell in der Webapp kann kontinuierliches Feedback gesammelt werden. Dies würde dem KI-Modell ermöglichen, sich stetig weiterzuentwickeln und zu verbessern.

Literaturverzeichnis

- [1] L. Walter, „Umgang mit Praxisgeräten nach der neuen Rechtslage. BVOU Netzwerk.“ 2022. [Online]. Available: <https://www.bvou.net/umgang-mit-praxisgeraeten-nach-der-neuen-rechtslage/>. [Zugriff am 19 08 2023].
- [2] M. D. C. H. U. e. a. Willing, „Analyzing medical device connectivity and its effect on cyber security in german hospitals,“ 2020. [Online]. Available: <https://doi.org/10.1186/s12911-020-01259-y>.
- [3] M. N. A. M. K. M. Bektemur G, „Cost analysis of medical device spare parts,“ 2018. [Online]. Available: doi: 10.12669/pjms.342.14245..
- [4] K. Lee, „Evolution of device monitoring,“ *Journal of Health Tech*, 2019.
- [5] S. & B. R. White, „Monitoring in intensive care,“ *ICU Management Journal*, 2019.
- [6] R. K. Mobley, *An introduction to predictive maintenance*, Elsevier, 2002.
- [7] B. K. N. Rao, *Handbook of Condition Monitoring*, Elsevier, 2016.
- [8] A. C. C. Tan, *Condition monitoring and assessment of power transformers using computational intelligence*, Springer, 2016.
- [9] J. Smith, „The democratization of IoT: A look into The Things Network,“ *IoT For All*, 2019.
- [10] L. Martinez, „How IoT is Transforming Businesses,“ *Innovation Digest*, 2019.
- [11] H. & L. J. Kim, „Applications of IoT across Industries,“ *ournal of Advanced Technology*, pp. 10-17, 2021.
- [12] R. Gupta, „Challenges Faced in IoT Implementation,“ *Global Tech Review*, pp. 200-205, 2019.
- [13] T. T. Network, „What is LoRaWAN,“ 2020. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/>. [Zugriff am 16 08 2023].
- [14] Mokolora, „Moko Lora,“ 2022. [Online]. Available: <https://www.mokolora.com/>. [Zugriff am 05 09 2023].
- [15] L. Alliance, „LoRaWAN Specification,“ LoRa Alliance,“ 2015. [Online]. Available: <https://www.lora-alliance.org/lorawan-for-developers>. [Zugriff am 06 08 2023].
- [16] Semtech, „official LoRa page,“ 2023. [Online]. Available: <https://www.semtech.com/>. [Zugriff am 16 08 2023].
- [17] L. S. d. L. Alliance, „semtech.com,“ 2023. [Online]. Available: <https://info.semtech.com/find-documentation>. [Zugriff am 10 09 2023].
- [18] Elektronik-Kompendium.de, „LoRa / LoRaWAN - Long Range Wide Area Network,“ 2023. [Online]. Available: <https://www.elektronik-kompendium.de/sites/kom/2804211.htm>. [Zugriff am 12 09 2023].
- [19] K. Davis, „Use cases of The Things Network in modern cities,“ *City Tech*, 2021.

- [20] T. T. Network, „Payload Functions Documentation,“ 2018. [Online].
- [21] U. K. P. & S. M. Raza, „Low Power Wide Area Networks: An overview,“ 2017.
- [22] IBM, „Was ist maschinelles Lernen?,“ IBM, 2023. [Online]. Available: <https://www.ibm.com/topics/machine-learning>. [Zugriff am 13 09 2023].
- [23] A. Vidhya., „Understand Random Forest Algorithms With Examples (Updated 2023),“ 2013-2023. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>. [Zugriff am 13 09 2023].
- [24] S. Studio, „Grove LoRa E5 New Version,“ Seeed Studio Wiki, 2023. [Online]. Available: https://wiki.seeedstudio.com/Grove_LoRa_E5_New_Version/. [Zugriff am 16 08 2023].
- [25] S. Studio, „Grove-Electricity_sensor,“ 2023. [Online]. Available: https://wiki.seeedstudio.com/Grove-Electricity_Sensor/. [Zugriff am 10 08 2023].

Anhang

Installation der Programmierumgebung und Entwicklungsumgebung

Zunächst soll die Programmierumgebung "Visual Studio Code" heruntergeladen werden. Um das Nucleo-F767ZI-Board programmieren zu können, fügt man die Plattform "PlatformIO" als Erweiterung (Extension) zu Visual Studio Code hinzu. Des Weiteren erstellt man ein neues Projekt mit dem Framework "Arduino" und dem Board "nucleo_f767zi".

Nach dieser Installation sollte das LoRa-Modul mit dem Mikrocontroller-Board verbunden werden, wie in der beigefügten **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt ist. Anschließend verbindet man das Board mit dem Laptop.

Um einen ersten Test durchzuführen, sollte man mittels der "AT"-Befehle das LoRa-Modul mit der Konsole kommunizieren lassen.

Tabelle 9 Entwicklungsumgebung

| | |
|------------------|-------------------------------------|
| Quelltext-Editor | Visual Studio Code (VSC) |
| VSC Add-on | Platform IO IDE (Arduino framework) |

Test verfahren der Funk Sensor und Webapp

Um den Zustand eines Geräts zu testen, sollte man folgende Schritte durchführen:

WebApp starten:

Frontend:

- Zuerst sollte man mithilfe des „cd“-Befehls in das Verzeichnis des Frontends navigieren.
- Anschließend sollte man das Frontend mit dem Befehl „npm start“ starten.

Backend:

- Als Erstes sollte man die virtuelle Umgebung mit dem entsprechenden Befehl aktivieren.
- Danach sollte man ins Backend-Verzeichnis wechseln.
- Zum Schluss sollte man das Backend mithilfe des Befehls „python App.py“ starten.

Gerät an den Funksensor anschließen:

- Zuerst sollte man die „Reset“-Taste am Funksensor drücken, um den Stromfluss zu gewährleisten.
- Danach kann man das Gerät einstecken und zwischen seinen verschiedenen Zuständen wechseln.

In der WebApp überprüfen:

- Hier sollte man in der WebApp prüfen, ob der aktuell fließende Stromwert sowie die vom ML-Modell erkannten Gerätezustände in Echtzeit angezeigt werden.
- Schließlich kann man vergleichen, ob der erfasste Zustand des Geräts mit den Vorhersagen des ML-Modells übereinstimmt.



Abbildung 30: Reset Taste der Funksensor

Nachdem man die WebApp erfolgreich gestartet und den Funksensor aktiviert hat, kann man die Zustände des Geräts überprüfen (Abbildung 8). Die Echtzeitdarstellung in der WebApp ermöglicht es, direkt zu sehen, ob die erkannten Zustände des ML-Modells mit dem tatsächlichen Zustand des Geräts übereinstimmen (siehe Abbildung 30)

Decode JavaScript code

```
function Decoder(bytes, port) {
  var decoded = {};

  // Decode maxCurr from the first 3 bytes divide by 10 for 1 decimal place
  decoded.maxCurr = (bytes[0] << 16 | bytes[1] << 8 | bytes[2]) / 10;

  // Decode effCurr from the next 3 bytes divide by 10 for 1 decimal place
  //decoded.effCurr = (bytes[3] << 16 | bytes[4] << 8 | bytes[5]) / 10;

  // Decode status from the next byte
  //decoded.status = decoded.maxCurr > 40 ? "ON" : "OFF";

  return decoded;
}
```