


# LORA-BASIERTE ÜBERWACHUNG DER BODENFEUCHTE UND TEMPERATUR

Berkan Kurt & Ugur Turhal

berkan.kurt@stud.unibas.ch  
u.turhal@stud.unibas.ch



# Inhaltsverzeichnis

<b>1.</b>	<b>EINLEITUNG.....</b>	<b>2</b>
1.1	PROJEKT BESCHREIBUNG.....	2
1.2	SOFTWARE-PROZESSMODELL UND TEAMORGANISATION.....	3
<b>2.</b>	<b>ANFORDERUNGEN.....</b>	<b>4</b>
2.1	WERKZEUGE UND TECHNOLOGIEN.....	4
2.2	FUNKTIONELLE ANFORDERUNGEN.....	4
2.2	NICHT-FUNKTIONELLE ANFORDERUNGEN.....	5
2.3	EINSCHRÄNKUNGEN.....	5
<b>3.</b>	<b>METHODOLOGIE.....</b>	<b>6</b>
3.1	SYSTEM-DESIGN.....	6
3.2	SYSTEMKOMMUNIKATION.....	6
<b>4.</b>	<b>ERGEBNIS.....</b>	<b>8</b>
4.1	SYSTEM-LEISTUNG.....	8
<b>5.</b>	<b>PROBLEME.....</b>	<b>10</b>
<b>6.</b>	<b>FAZIT.....</b>	<b>10</b>
<b>7.</b>	<b>ANHANG.....</b>	<b>11</b>
6.1	ENDPRODUKT.....	11
6.2	BIBLIOGRAFIE.....	11

## 1. Einleitung

Im Rahmen der Vorlesung “Internet and Security”, in der ein Teil des Leistungsnachweises in Form eines Projektes erbracht wird, haben wir uns entschieden ein System zur Messung und Überwachung eines städtischen Hausgartens mit Hilfe der Internet of Things (IoT)-Technologie zu entwickeln. Das System verfügt über drei Sensoren für Temperatur und Feuchtigkeit, die mit einem lokalen LoRa-basierten Netzwerk verbunden sind. Der Zweck unseres Projektes ist es, Hobbygärtner, wie unsere Eltern, eine Möglichkeit zu bieten autark, drahtlos und ferngesteuert ihre Gartenarbeit zu überwachen.

Stand 2018 leben etwa 54 % der Weltbevölkerung in städtischen Gebieten, was die Produktionsrate des Agrarsektors verringert hat.<sup>1</sup> Dies hat zu einer Zunahme der städtischen Landwirtschaft geführt<sup>2</sup>, die zusammen mit der Tendenz, vernetzte und verbundene Geräte zu haben, die Notwendigkeit hervorgerufen hat, städtische Gärten in intelligente, unabhängige und produktive Räume zu verwandeln. Allerdings gibt es derzeit kaum Anbieter von solchen Systemen. Startups wie PlantLink konnten Ihre versprochenen nach dem Launch nicht halten und wurden eingestellt.<sup>3</sup> Aufgrund der Probleme der oben genannten Systeme und der für ihre Vernetzung erforderlichen Infrastrukturen wird in diesem Beitrag ein wirtschaftliches und effizientes System zur Messung und Überwachung eines Obst- und Gemüsegartens eines Hauses in Basel-Stadt, durch den Einsatz von IoT-Technologie vorgeschlagen.

### 1.1 Projekt Beschreibung

Das Projekt wird ausschliesslich für den Benutzer implementiert. Im Einzelnen besteht die Hauptfunktion des Projektes darin, Daten zu messen und auf einem Server abzuspeichern. Die Daten werden anhand von drei Temperatur- und Bodenfeuchtigkeitssensoren gemessen und anschliessend an ein Gateway gesendet, welches mit dem Heimnetzwerk verbunden ist via WLAN. Die Sensoren, senden in 40, 60 und 80 Sekunden abständen die Daten an das Gateway. Die Möglichkeit der individuellen Regulation des Messintervalls hat den Vorteil, dass wir die Lebensdauer der Batterie der Sensoren schonen können. Je grösser das Zeitintervall ist, desto

---

<sup>1</sup> UN contributors: 68% of the world population projected to live in urban areas by 2050, un.org, 16.05.2018, < <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html>>, Stand: 12.07.2022.

<sup>2</sup> Unter städtischer Landwirtschaft sind hier Lebensräume zu verstehen, die die Natur mit dem städtischen Umfeld verbinden und eine soziale und kulturelle Entwicklung ermöglichen.

<sup>3</sup> Burwood-Taylor, Louisa: Scotts Miracle-Gro Acquires Oso Tech After Rollercoaster Ride for Startup, agfundernews.com, 14.12.2016, < <https://agfundernews.com/scotts-miracle-gro-acquires-plant-link-after-rollercoaster-ride-for-startup>>, Stand: 12.07.2022.

länger ist auch die Lebensdauer der Batterie. Das Gateway hat ein Delay von 5 Sekunden, um sich mit dem Server zu verbinden. Der Benutzer kann dann die Werte auf einem Server, welcher auf einem Raspberry PI läuft, abrufen. Die Daten werden in drei CSV-Datei abgespeichert und kategorisiert.

## 1.2 Software-Prozessmodell und Teamorganisation

Wir haben uns für ein agiles Modell entschieden.<sup>4</sup> Dadurch können wir die Transparenz und die Änderungsgeschwindigkeit in dieser kurzen Zeitspanne erhöhen, was hoffentlich zu einer schnelleren Bereitstellung des entwickelten Systems führt. Nach dem Motto: Je mehr man nach Plan arbeitet, desto mehr bekommt man das, was geplant war, aber nicht das, was gebraucht wird. Da Software-Prozesse aber in einem Team von zwei Personen nur schwer umzusetzen sind haben wir uns auf bestimmte Grundsätze beschränkt, wie z. B., dass die Entwickler während des Projekts täglich zusammenarbeiten müssen und dass die funktionierende Soft- wie auch Hardware das wichtigste Mass für den Fortschritt ist.

Kontinuierliches Mentoring und geteilte Kompetenzen sind Grundsätze, die wir auch in dieser Phase beibehalten wollen. Einer der grossen Vorteile der Teamarbeit besteht darin, dass wir voneinander lernen und jeder den anderen anleitet. Da jeder im Team etwas vom anderen lernt, ist die Wirkung des Teams grösser als die Summe der Wirkung des Einzelnen. Die Paarprogrammierung war eine sehr hilfreiche Methode, um dies aufrechtzuerhalten. Die meisten Vorarbeiten und die Umsetzung wurden von Ugur geleistet, während Berkant sich auf die Optimierung, Qualität und Verbesserungen konzentrierte. Der Grund dafür ist, dass die vergangenen Projekte gezeigt haben, dass wir beide unterschiedliche Stärken bei der Umsetzung haben. Ugurs Stärke lag eher auf der Implementierungsseite.

Was den Bericht angeht, so haben wir beschlossen, dass Berkant den grössten Teil des Textes schreiben wird. Der Grund dafür ist, dass Berkant aufgrund seines Hauptfachs Geschichte mehr Erfahrung im Schreiben von Texten hat. Ugur wird seinen Teil in Form von Diagrammen beisteuern.

---

<sup>4</sup> Beck, K., et al.: Manifesto for Agile Software Development, [agilemanifesto.org](https://agilemanifesto.org/), 2001, <  
<https://agilemanifesto.org/>>, State: 12.06.2022.

## 2. Anforderungen

### 2.1 Werkzeuge und Technologien

Wir haben folgende Werkzeuge und Technologien verwendet;

- Espressif ESP32: Ein Arduino kompatibles Board mit Wifi und Bluetooth on board.
- LoRa Temperatur- und Bodenfeuchtigkeitssensor: Werden zur Überwachung der relativen Bodenfeuchtigkeit und der Temperatur eingesetzt. I<sup>2</sup>C kompatibel.<sup>5</sup>
- LoRa ESP32 Gateway Shield: Ein Gateway ist erforderlich, um alle Elemente in den LoRa-Netzen für das Internet auswertbar zu machen.
- Arduino IDE: Sie ermöglicht es uns, eine Verbindung zur Hardware herzustellen, um Programme hochzuladen und mit der Hardware zu kommunizieren.
- Micropython: MicroPython ist eine schlanke und effiziente Implementierung der Programmiersprache Python 3, die eine kleine Teilmenge der Python-Standardbibliothek enthält und für die Ausführung auf Mikrocontrollern und in eingeschränkten Umgebungen optimiert ist.<sup>6</sup>
- Raspberry Pi Model 4: Einplatinencomputer, welches in unserem Fall als Server eingesetzt wird, um die erhaltenen Messwerte abzuspeichern.
- Code with me (JetBrains): Plugin, mit dem Sie Ihr lokal laufendes Projekt mit anderen teilen können. Sie können Code with me für die Zusammenarbeit in Echtzeit, die Bearbeitung von Code, die gemeinsame Nutzung eines Servers und die Durchführung einer Debugging-Sitzung verwenden.
- Lötstation: Um das LoRa Gateway funktionstüchtig zu löten.
- Fusion360, Ender-3 v2: Werkzeuge und Technologien um ein Gehäuse für die Sensoren herzustellen.

### 2.2 Funktionelle Anforderungen

- Das Produkt muss die Temperatur messen können.
- Das Produkt muss die Bodenfeuchtigkeit messen können.
- Das Produkt muss die gemessenen Daten in einem Server abspeichern.

---

<sup>5</sup> Wikipedia contributors: I<sup>2</sup>C, wikipedia.org, < <https://de.wikipedia.org/wiki/I%C2%B2C> >, Stand: 17.07.2022.

<sup>6</sup> Micropython contributors: Micropython, micropython.org, < <https://micropython.org/> >, Stand: 12.07.2022.

- Die Sensoren müssen auf ein Datenverlust reagieren können.
- Die Sensoren müssen ein eindeutiger Knoten im Netzwerk sein.
- Der Server sollte die abgespeicherten Daten als CSV Format auf einer Website aufzeigen.
- Das Endprodukt sollte so viel Pakete erhalten und eintragen wie gesendet.

## 2.2 Nicht-funktionelle Anforderungen

- Jeder Knoten muss durch eine Identifikationsnummer gekennzeichnet sein.
- Die Sensoren sollen in verschiedenen Zeitintervallen die Messung durchführen.
- Das Intervall, in dem die Daten gemessen werden sollte 40 Sekunden für den ersten Sensor sein, 60 Sekunden für den zweiten und 80 Sekunden für den Dritten Sensor.
- Die Website, in welcher die Daten aufgezeigt werden, soll von verschiedenen Betriebssystemen und Geräten erreichbar sein.
- Dem Produkt sollte es möglich sein, Daten auch über eine grössere Entfernung als 1 km zu senden.
- Das System sollte 48 Stunden am Stück funktionieren.

## 2.3 Einschränkungen

Wir haben festgestellt, dass LoRa-Sensoren ein grosses Potenzial in Bezug auf ihre Funktionalitäten haben, aber sie nicht zuverlässig sind. Ein Aspekt, der Anlass zur Sorge gibt, ist die Verfügbarkeit auf freien Funkfrequenzen, so dass das System anfällig für Angriffe ist. Ausserdem ist es durch langsame Übertragungsraten gekennzeichnet und könnte zu Interferenzproblemen führen. Weiter erlaubt das Protokoll aufgrund der Vorschriften für das verwendete Frequenzband kein kontinuierliches Senden.

### 3. Methodologie

#### 3.1 System-Design

Es wurde ein eingebettetes System entwickelt, das mit Batterien betrieben wird und auf drei Komponenten basiert, um Klimavariablen in Echtzeit zu überwachen. Das System führt Überwachungs- und Übertragungsprozesse durch.

##### **I) LoRa Bodenfeuchtigkeit- und Temperatursensor**

Die LoRa-Bodenfeuchtesensoren fungieren als Sensorknoten. Sie messen den Bodenfeuchtigkeitswert in analoger Form nach einem Intervall von 40, 60 und 80 Sekunden. Das Intervall ist hierbei frei im Codeteil einstellbar. Der Sensorknoten sendet die Daten über den LoRa-Funk an das Gateway.

##### **II) ESP32 Gateway**

Das LoRa-Gateway wird für die Langstreckenkommunikation in der komplexen Umgebung der Stadt eingesetzt. Die tatsächliche Testdistanz erreicht laut Hersteller 2,8 Kilometer. Wenn die Störungen in den Vororten gering sind, ist die Entfernung noch grösser.

Der Empfänger wird ebenfalls mit ESP32 und LoRa-Modul hergestellt. Der Empfänger sammelt die Daten von mehreren Sensorknoten. Dann erstellt der Empfänger einen lokalen Server mit Hilfe der in ESP32 eingebauten Bibliotheksfunktion und einem Raspberry Pi.

##### **III) Server**

Bei dem Server handelt es sich um einen Raspberry Pi Server mit einer statischen IP-Adresse. Auf diesem Server werden die Daten gespeichert und es wird ein CSV Dokument erstellt, welches die Daten automatisch kategorisiert.

#### 3.2 Systemkommunikation

Wie erwähnt besteht unser System aus einem LoRa-Sender und einem LoRa-Empfänger (Siehe Abbildung 1). Der LoRa-Sender sendet die Messwerte der Bodenfeuchtigkeit und Temperatur im Aussenbereich an einen LoRa-Empfänger im Innenbereich. Der Vorteil der Verwendung

von LoRa ist, dass sie leicht eine drahtlose Verbindung zwischen zwei ESP32-Boards herstellen können, die mehr als 100 Meter voneinander entfernt sind. Im Gegensatz zu Wifi oder Bluetooth, die nur die Kommunikation über kurze Entfernungen unterstützen. Der Empfänger speichert die Messwerte auf einer Raspberry Pi und speichert die neuesten Sensormesswerte auf einem Webserver. Ein Gateway wird hierbei benötigt, um alle Elemente in LoRa-Netzwerken mit dem Internet verbinden zu können. Dieses LoRa-Gateway basiert auf dem MakePython ESP32 und dem 2-Kanal LoRa Expansion Board. Die Makepython LoRa-Transceiver verfügen über das LoRaTM Langstreckenmodem, das eine Spreizspektralkommunikation mit grosser Reichweite und hoher Störfestigkeit bei minimalem Stromverbrauch bietet. Das kombinierte Board enthält das LoRa-Modul RFM95W, das das 868MHz-Band bedient, während der ESP32 für die Bluetooth- oder WiFi-Fähigkeiten zuständig ist. Unser LoRa-Gateway bietet eine Bandbreite von 250kHz mit einem Spreizfaktor von 7. Wenn Sie die Datenrate erhöhen (die Bandbreite erhöhen oder den Spreizfaktor verringern), können Sie diese Bytes in kürzerer Zeit übertragen. In diesen Fällen sieht die Berechnung etwa so aus: Wenn Sie die Bandbreite um das Doppelte erhöhen (von BW125 auf BW250), können Sie in der gleichen Zeit 2 Bytes mehr übertragen. Verringert man den Spreizungsfaktor um eine Stufe (von SF10 auf SF9), kann man in der gleichen Zeit 2x mehr Bytes übertragen. Durch die Verringerung des Spreizungsfaktors wird es für das Gateway schwieriger, eine Übertragung zu empfangen, da es empfindlicher gegenüber Rauschen ist. Sie können dies mit zwei Personen vergleichen, die sich an einem lauten Ort (z. B. einer Bar) treffen. Wenn sie weit voneinander entfernt sind, müssen sie langsam sprechen (SF10), aber wenn sie nahe beieinander sind, können sie schneller sprechen (SF7).

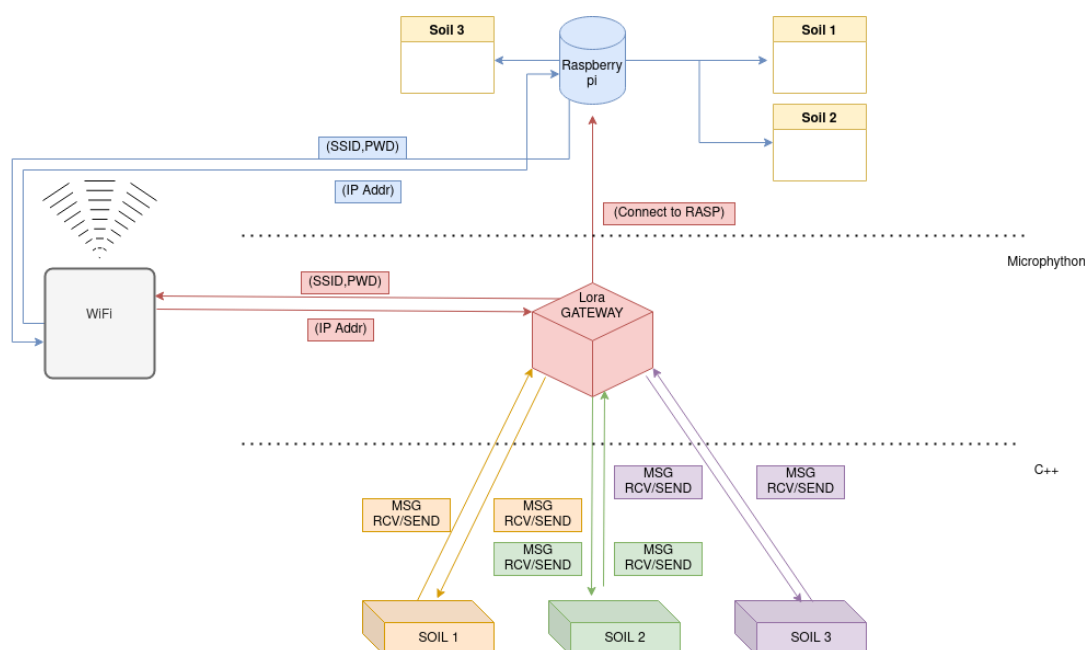


Abbildung 1 – Architektur für die Kommunikation



## 4. Ergebnis

### 4.1 System-Leistung

Wenn das eingebettete System betrieben wird, werden die beobachteten Ergebnisse für die Messungen in den nachfolgenden Abbildungen dargestellt. Der Zustand des Bodens wird auf die gleiche Weise mit den entsprechenden Informationen dargestellt. Die Abbildungen zeigen den jeweiligen Median der Messungen über jeweils 24 Stunden auf.

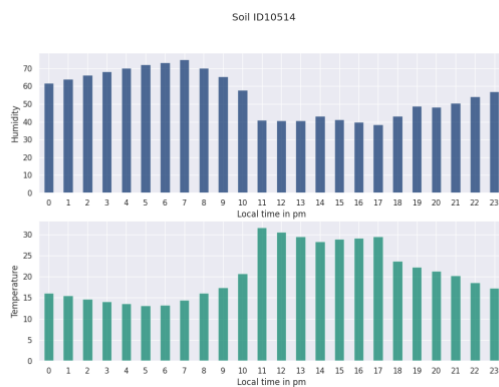


Abbildung 2 – Sensor 1 vor Optimierung

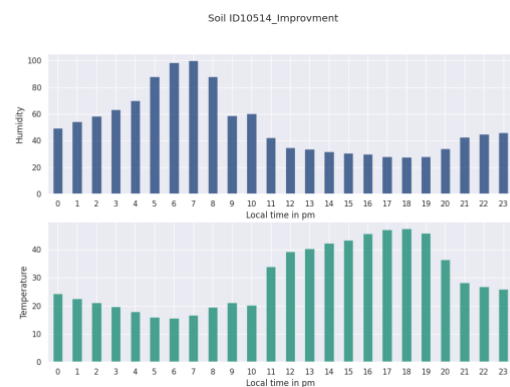


Abbildung 3 – Sensor 1 nach Optimierung

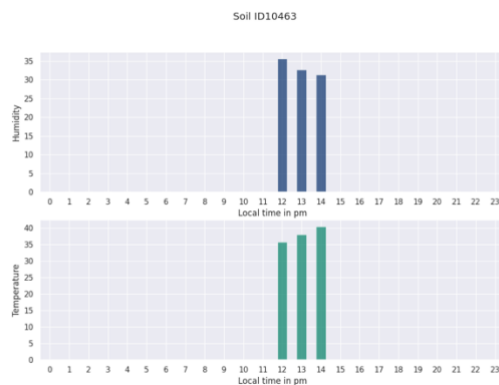


Abbildung 4 – Sensor 2 vor Optimierung

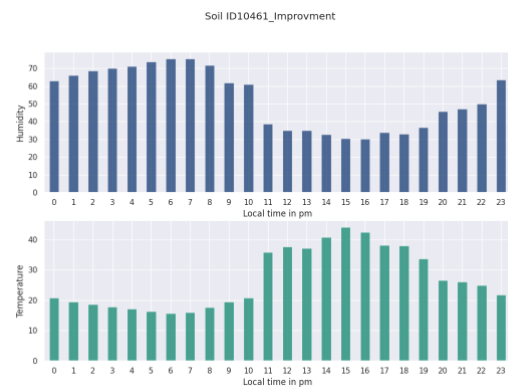


Abbildung 5 – Sensor 2 nach Optimierung

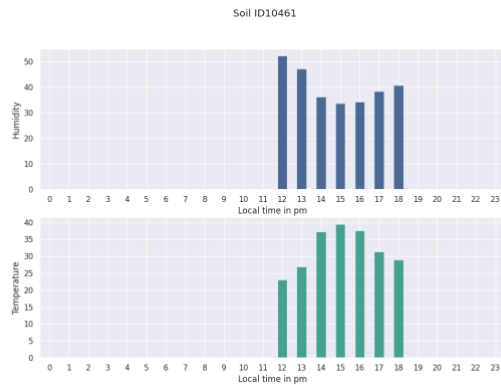


Abbildung 6 – Sensor 3 vor Optimierung

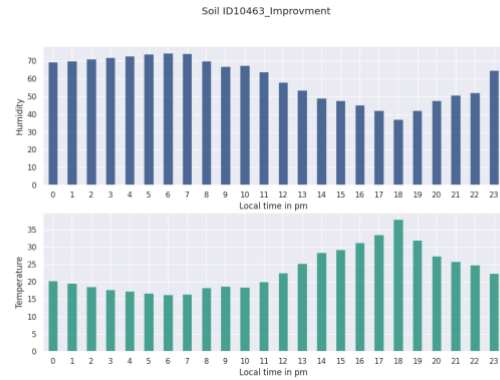


Abbildung 7 – Sensor 3 nach Optimierung

Die Abbildungen auf der linken Seite stellen die drei LoRa-Sensoren vor der Optimierung dar. Anhand von Abbildung 4 und 6 ist zu sehen, dass wir mit einem massiven Datenverlust konfrontiert waren. Dieser Datenverlust ist schwachen Batterien und fehlerhaftem Code zu verschulden. Dieser Datenverlust konnte aber komplett neutralisiert werden. Durch bidirektionalität konnte erreicht werden, dass wir kontinuierliche Messungen ohne Datenverlust haben. Ausserdem haben wir ein Array implementiert, welches den Datenverlust überprüft und somit einen zusätzlichen Schutz bietet.

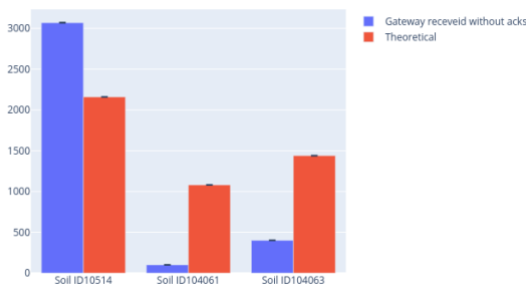


Abbildung 8 – Datentransfer vor Optimierung

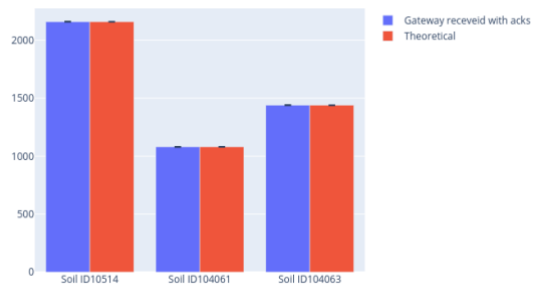


Abbildung 9 – Datentransfer nach Optimierung

Um die Diskrepanz zwischen theoretisch zu erhaltenen Daten und realen erhaltenen Daten aufzuzeigen haben wir sie in einem Säulendiagramm dargestellt.

In der Abbildung 8 ist zu sehen, dass der erste Sensor zu viele Daten erhalten hat, hingegen der zweite und dritte Sensor zu wenig. Der theoretische Wert ist in der Grafik als rote Säule repräsentiert. Der theoretische Wert wurde hierbei berechnet, indem man Faktoren wie Dauer der Messung und das Intervall der Messung berücksichtigt. Bei einer Messung über 24 Stunden mit einem Messungsintervall von 60 Sekunden (Sensor ID104063) kommen wir somit auf einen theoretischen Wert von 1440 Daten, die gesendet werden müssen. In der Realität aber haben

wir von diesen 1440 Daten lediglich einen Bruchteil erhalten. Wie in der Abbildung 9 zu sehen ist konnte dieses Problem durch bidirektionalität behoben werden.

## **5. Probleme**

Während des Projektes sahen wir uns mit verschiedenen Problemen konfrontiert. Technische Probleme wie eine Lötstation und passende Batterien zu organisieren, erwiesen sich schwieriger als erwartet. Bei der Lötstation mussten wir auf eine sehr kleine Lötspitze zurückgreifen, da die Arbeit mit einer Standardgrösse nicht machbar gewesen wäre. Bei den Batterien für die Sensoren hatten wir Probleme mit den richtigen Spezifikationen und es kam zu einer kleinen Explosion, welches unser Sensoren aber überstanden hat.

Programmierspezifische Probleme waren der Umgang mit Micropython mit welchem wir bis jetzt noch nicht gearbeitet haben. Dies war aber unumgänglich, um das Gateway zu konfigurieren. Ausserdem sahen wir uns konfrontiert mit Scheduling-problemen(fairness) bei den Sensoren, das Problem konnte mithilfe von threads gelöst werden. Wie schon erwähnt war das grösste Problem aber der Datenverlust, welches wir schlussendlich durch bidirektionalität lösen konnten.

## **6. Fazit**

Ein kostengünstiges und verbrauchsarmes System zur Messung und Überwachung wurde unter Verwendung von Open-Source- und IoT-Technologie entwickelt und im städtischen Garten in Basel verwendet. Das System besteht aus mehreren Komponenten: LoRa-Temperatur- und Bodenfeuchtigkeitssensoren, LoRa-Gateway und einem RaspberryPi. Der Kommunikationsprozess wurde in mehreren Phasen durchgeführt, wobei die erste auf den Sensor-Knoten wirkt, dieser wiederum kommuniziert mit dem Gateway Prozessor welches mit einem WLAN verbunden ist, welches dann wiederum mit unserem Webserver (Raspberry Pi) kommuniziert. Jede Messung wurde in 40, 60 und 80 Sekunden in den Abständen durchgeführt und dann wurde ein Durchschnittswert jeder Variable ermittelt, um die Genauigkeit und Präzision der Werte in den erfassten Daten zu gewährleisten. Alle Daten der einzelnen Messgrößen werden im Server in Form einer CSV Datei gespeichert. Im Falle eines Netzwerkproblems über mehrere Stunden, kann man auch auf den Bildschirm des Gateways zurückgreifen, welcher ebenfalls in der Lage ist die Temperatur und die Bodenfeuchte anzuzeigen.

## 7. Anhang

### 6.1 Endprodukt



Abbildung 10 – Bodenfeuchtigkeits- und Temperatursensor



Abbildung 11 - Gateway



Abbildung 12 - Gateway

### 6.2 Bibliografie

- Beck, K., et al.: Manifesto for Agile Software Development, agilemanifesto.org, 2001, <https://agilemanifesto.org/> , Stand: 12.07.2022
- Burwood-Taylor, Louisa: Scotts Miracle-Gro Acquires Oso Tech After Rollercoaster Ride for Startup, agfundernews.com, 14.12.2016, < <https://agfundernews.com/scotts-miracle-gro-acquires-plant-link-after-rollercoaster-ride-for-startup>>, Stand: 12.07.2022.
- Micropython contributors: Micropython, micropython.org, < <https://micropython.org/>>, Stand: 12.07.2022.
- UN contributors: 68% of the world population projected to live in urban areas by 2050, un.org, 16.05.2018, < <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html>>, Stand: 12.07.2022.
- Wikipedia contributors: I<sup>2</sup>C, wikipedia.org, < <https://de.wikipedia.org/wiki/I%C2%B2C>>, Stand: 17.07.2022.