



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

CitizenScience – IoT für die Sinne
Dokumentation
Projekt Wetterstation für die Adler-Apotheke

Name des Studiengangs

Wirtschaftsinformatik

Fachbereich 4

vorgelegt von Gruppe 6:

Markus Flasche

Marco Lindermeier

Robin Streckies

Patrick Siegel

Datum:

Berlin, 27.01.2021

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Tabellen- und Abbildungsverzeichnis	3
Abkürzungsverzeichnis	4
1 Einleitung	1
1.1 <i>Ziel und Konzept.....</i>	<i>1</i>
1.2 <i>Beschreibung der Entscheidungslogik (Logikfluss).....</i>	<i>3</i>
2 Prototyp.....	6
2.1 <i>Aufbau der Hardware</i>	<i>6</i>
2.2 <i>Verwendete Softwaretechnologien</i>	<i>8</i>
2.3 <i>Softwarearchitektur.....</i>	<i>9</i>
2.4 <i>Frontend.....</i>	<i>9</i>
2.5 <i>Backend</i>	<i>12</i>
2.6 <i>Kommunikationswege</i>	<i>13</i>
3 Inbetriebnahme	18
4 Problemstellungen während der Realisierung.....	20
4.1 <i>Covid-19</i>	<i>20</i>
4.2 <i>Reichweite LoRaWAN Gateway</i>	<i>20</i>
4.3 <i>Deaktivierung der NeoPixel des IoT Octopus</i>	<i>20</i>
5 Ausblick	22
6 Fazit	22
7 Literaturverzeichnis	V

Tabellen- und Abbildungsverzeichnis

Abbildung 1: Entscheidungslogik der Krankheitsbilder:	4
Abbildung 2: Hardware Vorderseite	6
Abbildung 3: Hardware Rückseite	6
Abbildung 4: IoT Octopus Vorderseite	6
Abbildung 5: HT-Rohr Außenansicht	7
Abbildung 6: HT-Rohr Innenansicht	7
Abbildung 7: Grobkonzept Softwarearchitektur.....	9
Abbildung 8: Darstellung des Frontends	10
Abbildung 9: Kommunikationswege	12
Abbildung 10: Screenshot TTN Decoder	14
Abbildung 11: Überführung JSON-Objekt	14
Abbildung 12: Anbindung des The Things Networks mithilfe des MQTT-Protokolls	15
Abbildung 13: Erstellung eines Points zur Speicherung in der Datenbank.....	15
Abbildung 14: Anbindung der Influx Cloud Datenbank	16
Abbildung 15: Auswertung der maximale Schwankung des Druck mit Hilfe der InfluxDB.....	16
Abbildung 16: Auswahl des Krankheitsbilds.	17
Abbildung 17: Einstufung der Feinstaubbelastung.....	17
Abbildung 18: Vorderseite Octopus.....	21
 Tabelle 1: Krankheitsbilder	 2

Abkürzungsverzeichnis

HTTP	Hypertext Transfer Protocol
IoT	Internet of Things (Internet der Dinge)
LoRaWAN	Long Range Wide Area Network
MQTT	Message Queuing Telemetry Transport
PM2,5	particulate matter
QR-Code	Bundesdatenschutzgesetz
TTN	The Things Network
URL	Uniform Resource Locator
Vgl.	Vergleich

1 Einleitung

Die dargestellte IoT-Lösung ist dem Projekt: Citizen Science – IoT für die Sinne zuzuordnen. Es entstand im Rahmen des Masterstudiengangs Wirtschaftsinformatik und wurde in der Lehrveranstaltung Wissensmanagement von Frau Prof. Dr. Willner erarbeitet. Der angefertigte Prototyp dient der Automatisierung von Produktempfehlungen und soll die Interaktivität mit vorbeilaufenden Passanten*innen fördern.

1.1 Ziel und Konzept

Die sich stetig verändernden Wetterbedingungen stehen bei vielen Menschen im direkten Zusammenhang mit gesundheitlichen Beschwerden oder Symptomen. So geht aus einer Studie hervor, dass 50 Prozent der 1623 Probanden*innen Angaben an wetterbedingten Krankheiten zu leiden[1]. Dies kann sich in Kopfschmerzen, Müdigkeit oder Kreislaufproblemen äußern, die vorerst aus unerklärlichen Gründen bei den betroffenen Personen eintreten. Des Weiteren ist zu erkennen, dass bestimmte klimatische Situationen das Auftreten von ausgewählte Krankheiten begünstigen. So spricht man im Herbst und Winter von der Erkältungszeit [2], während die Gefahr an einer Grippe zu erkranken von Januar bis April erhöht ist [3]. Zusätzlich ist bereits bekannt, dass auch die UV-Strahlung gesundheitliche Probleme wie Sonnenbrand und Hautkrebs verursachen kann [4], wenn keine entsprechenden Maßnahmen, wie beispielsweise die Nutzung von Sonnencreme, vorgenommen werden.

Um dieses Problem aufzugreifen soll die vorliegende IoT-Lösung dazu beitragen, ausgewählte Krankheitsmuster durch die Messung des Wetterverlaufs als Jene zu identifizieren, um daraufhin automatisierte Produktempfehlung darzustellen.

Dies soll mittels der Erhebung und Analyse von Temperatur, Luftfeuchtigkeit, Luftdruck, UV-Strahlung und des Zeitraums bewerkstelligt werden. Zusätzlich soll der Prototyp den betroffenen Personen Auskunft über den Grund der vorerst unerklärlichen Beschwerden geben, da beispielsweise Schwankungen in der Temperatur oder des Luftdruck erschwert als solche wahrgenommen werden können.

Folgende Produkte wurden in Absprache mit der Adler-Apotheke in Adlershof für das jeweilige Krankheitsbild bestimmt und empfohlen (vgl. Tabelle 1).

Tabelle 1: Krankheitsbilder

Krankheitsbild	Medikament
Erkältung	Aspirin Complex
Kreislaufprobleme	Cataegutt
Müdigkeit	Vigantol
Kopfschmerzen	Thomapyrin
Sonnenbrand (Hautkrebs)	Roche Posay
Default	Aspirin Complex

Neben der Darstellung der automatisch generierten Produktempfehlung soll gleichzeitig der Feinstaub (Partikelgröße 2,5) gemessen und dargestellt werden, da ein erhöhter Feinstaubwert nachweislich gesundheitliche Beschwerden hervorrufen kann [5]. Die gemessenen Feinstaubwerte werden den vorbeilaufenden Passanten mittels einer Ampel veranschaulicht, die bei erhöhten Feinstaubwerten von grün auf gelb beziehungsweise rot wechselt.

Um die Interaktion zu fördern, wird sowohl ein Plakat erstellt als auch ein QR-Code verwendet, der vorbeilaufenden Passanten anregen soll, das Projekt selbst nachzubauen beziehungsweise Teilprozesse durchzuführen, um beispielsweise eine IoT-Wetterstation zu erstellen

1.2 Beschreibung der Entscheidungslogik (Logikfluss)

Die Parameter zur Entscheidungslogik stammen aus einer Literaturrecherche, die für die jeweiligen Krankheiten und Symptome durchgeführt wurde. Lediglich die Bestimmung der Schwankungen bezüglich der Temperatur sowie des Luftdrucks wurden aus historischen Datensätzen geschätzt und könnten bei weiterer Betrachtung aktueller Werte gegebenenfalls angepasst werden, um die Logik zu optimieren [6].

Für die Berechnung der Entscheidungslogik (vgl. Abbildung 1) wurden folgende Quellen genutzt:

- Quellen zur Entscheidung der Parameter - Erkältung: [7]
- Quellen zur Entscheidung der Parameter - Kreislaufprobleme: [8–12]
- Quellen zur Entscheidung der Parameter - Müdigkeit: [13, 14]
- Quellen zur Entscheidung der Parameter - Sonnenschutz: [15, 16]
- Quellen zur Entscheidung der Parameter - Kopfschmerzen: [8, 10, 11, 17, 18]
- Quellen zur Entscheidung der Parameter - Feinstaub: [19, 20]
- Allgemeine Quellen: [21–27]

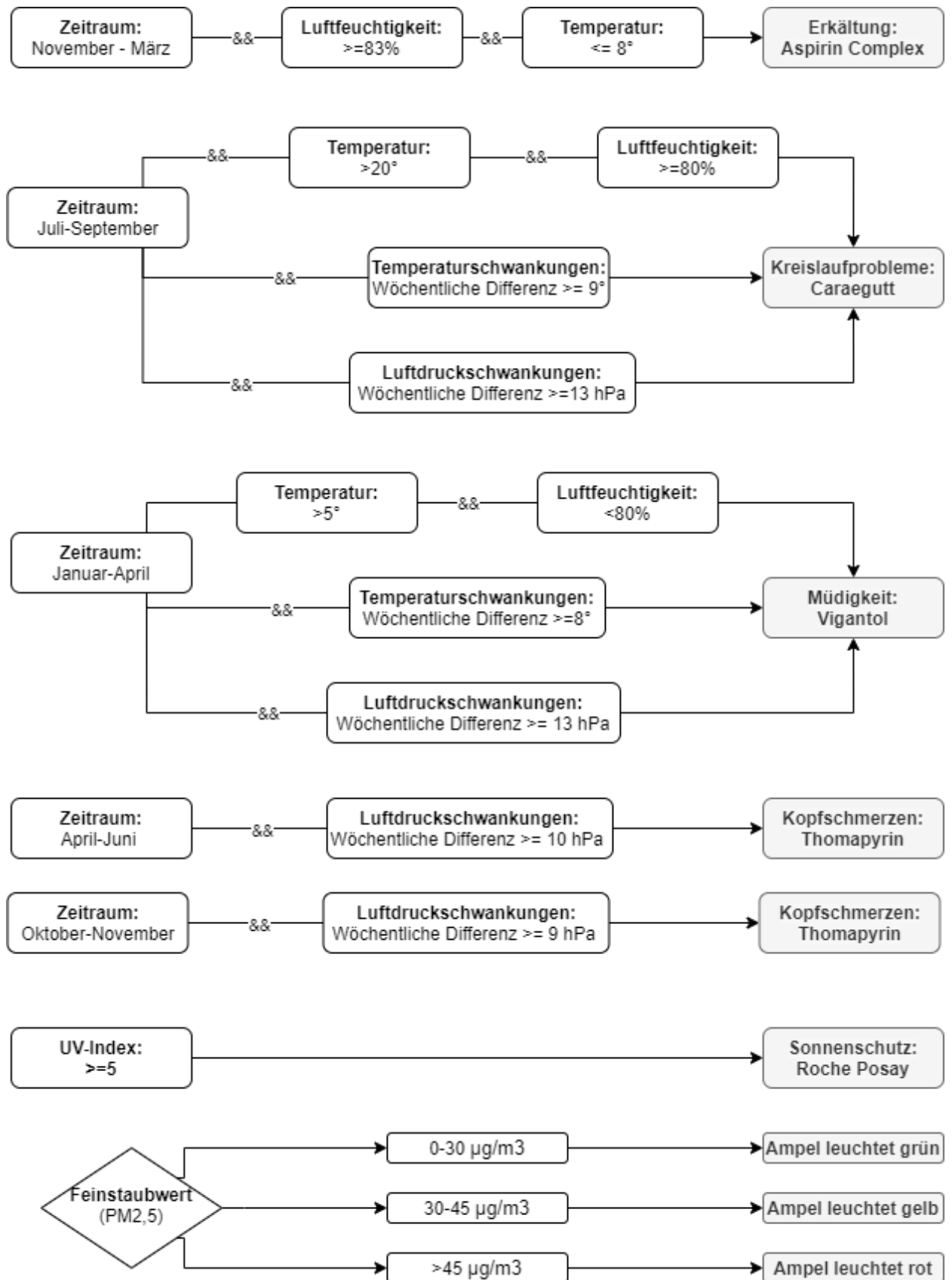


Abbildung 1: Entscheidungslogik der Krankheitsbilder:

Zur Berechnung der wöchentlichen Differenz werden die gemessenen Daten einer Kalenderwoche betrachtet. Hierbei wird der niedrigste gemessene Wert von dem Maximalwert subtrahiert. Bei den Temperaturschwankungen konzentriert sich die Berechnung der wöchentlichen Differenz auf unterschiedliche Zeiträume wodurch die Genauigkeit gewährleistet werden kann. Somit ergaben sich folgende Bereiche: 12:00 Uhr bis 24:00 Uhr (Tag) und 00:01 Uhr bis 08:00 Uhr (Nacht) und 08:00 Uhr bis 12:00 (Vormittag) gemessen.

2 Prototyp

Im folgenden Kapitel werden der Aufbau und die Herangehensweise für den Prototypen geschildert.

2.1 Aufbau der Hardware

Grundlegend besteht der Prototyp aus einem Microcontroller, dem IoT Octopus Board (Siehe Hardware Vorderseite Abbildung 2 und Abbildung 2 Abbildung 3). Auf diesem befinden sich ein BME680 Sensor, welcher für uns notwendige Temperatur-, Luftfeuchtigkeits- und Luftdruckdaten zur Verfügung stellt. Wie in Abbildung 4 zu sehen ist am Digital-Pin I2C ein "Grove - Laser PM2.5 Sensor (HM3301)" angeschlossen. Dieser Feinstaubsensor stellt die Daten für die Luftverschmutzung bereit.

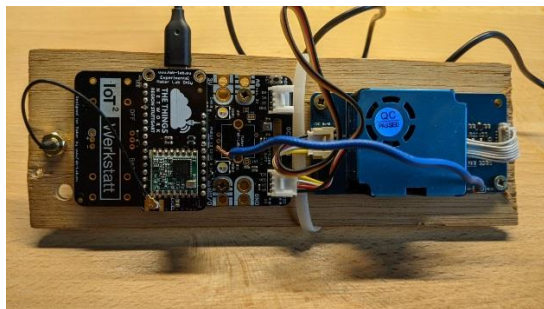


Abbildung 2: Hardware Vorderseite



Abbildung 3: Hardware Rückseite

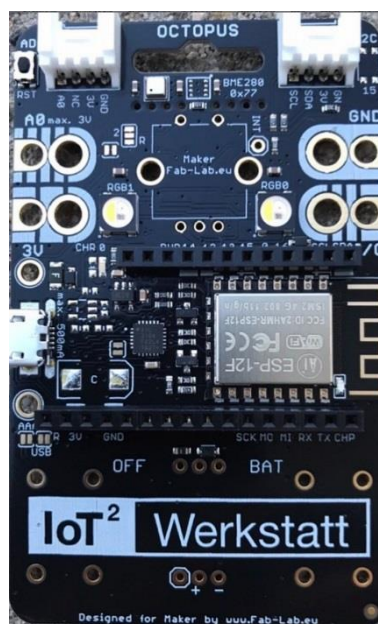


Abbildung 4: IoT Octopus Vorderseite

Auf dem Octopus Board ist mittig ein Feather Wing aufgesetzt, auf dem eine LoRaWan-Antenne angeschlossen ist. Die Stromversorgung erfolgt mittels Powerbank.

Wie ebenfalls auf Abbildung 2 zu sehen ist der HM3301-Sensor mittels blauen Überbrückungskabel am Set-Pin und am 14-Pin mit dem Octopus verbunden. Dies hat den Grund, dass so der Feinstaubsensor, im genauen der Lüfter, nicht durchweg läuft. So ist es möglich weniger Strom der Powerbank zu verbrauchen und die Ladezyklen zu verringern.

Alle genannten Teile sind auf einer dünnen Holzplatte verschraubt und wurden dann in einem U-förmiges HT-Rohr DN 75 mit Belüftungsventil geschoben (Siehe Abbildung 5 und Abbildung 6). So sind eine Luftzirkulation und wiederum der Schutz der Hardware gewährleistet.



Abbildung 5: HT-Rohr Außenansicht



Abbildung 6: HT-Rohr Innenansicht

Zur endgültigen Darstellung des Dashboards auf dem Monitor wird eine Raspberry PI 4 genutzt. Dieser wird eingesetzt, um die eingehenden Daten zu routen und zu verwalten. Die Inbetriebnahme wird in Punkt X erläutert.

2.2 Verwendete Softwaretechnologien

Die im Folgenden genannten Softwaretechnologien wurden für die Umsetzung der Wetterstation genutzt.

2.2.1 Arduino IDE

Die Arduino IDE ist eine Open-Source Anwendung, welche Code für die jeweiligen genutzten Sensoren kompiliert und anschließend auf das Octopus Board via USB laden kann.

2.2.2 jQuery

JavaScript-Bibliothek die zur Anzeige der Werbung und für die Abfrage der Daten im Backen genutzt wird.

2.2.3 NodeJS

JavaScript-Laufzeitumgebung für serverseitige JavaScript-Anwendungen. NodeJs wird in diesem Projekt genutzt um eine Schnittstelle zwischen TTN, der Datenbank und dem Frontend herzustellen.

2.2.4 InfluxDB-Cloud

Zeitreihen Datenbank die zur Speicherung der Sensordaten genutzt wird.

2.2.5 The Things Network

Dezentralisiertes LoRaWan-Netzwerk das zur Kommunikation zwischen Entwicklerboard und NodeJs-App genutzt wird.

2.3 Softwarearchitektur

In diesem Abschnitt wird genauer auf die softwareseitige Umsetzung eingegangen. In der folgenden Abbildung 7 ist das Gesamtkonzept zu sehen.¹⁻⁴ Die weiteren Punkten des Kapitels befassen sich genauer mit der Architektur.

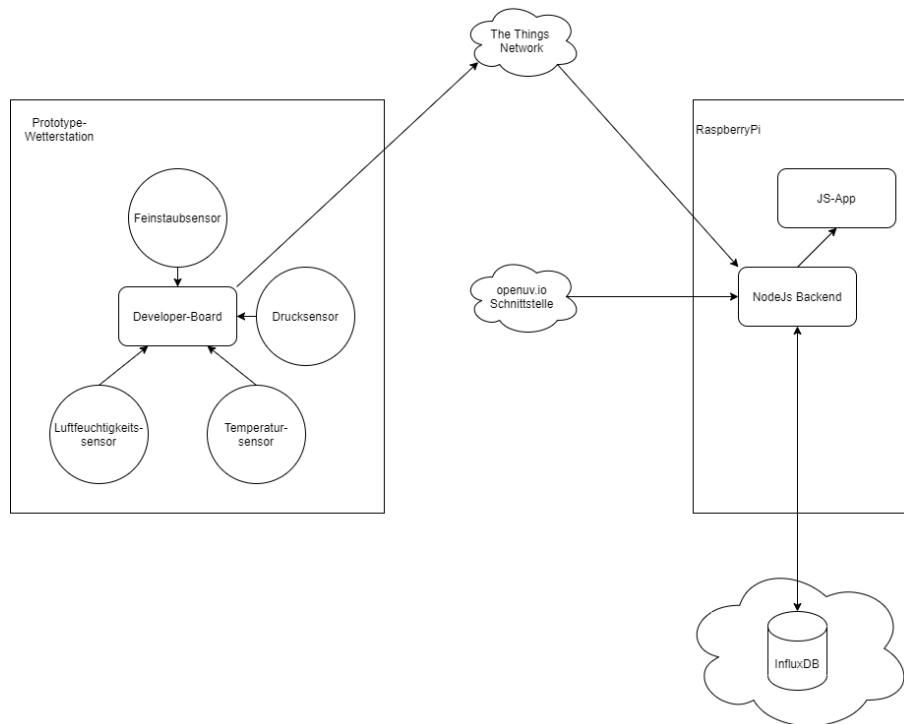


Abbildung 7: Grobkonzept Softwarearchitektur

2.4 Frontend

Das in Abbildung 8 dargestellte Frontend wird sich auf einem Monitor im Schaufenster der Adler-Apotheke in Adlershof befinden. Die Visualisierung lässt sich in drei Teilbereiche aufteilen. Während der obere Abschnitt der Abbildung eine Informationszeile, mit den Angaben aller gemessenen Werte des Prototypen darstellt, wird den vorbeilaufenden Passanten*innen im mittleren Teil die Produktempfehlung veranschaulicht. Im unteren Teil des Frontend haben Kunden*in die Möglichkeit mittels QR-Code auf den genutzten Source Code sowie einer Anleitung zur Inbetriebnahme zuzugreifen. Somit sollen die Interaktivität und Anregung zum selbst bauen gefördert werden.

¹ <https://nodejs.org/en/>

² <https://jquery.com/>

³ <https://www.arduino.cc/en/software>

⁴ <https://www.influxdata.com/products/influxdb-cloud/>



Abbildung 8: Darstellung des Frontends

Im Folgenden werden die einzelnen Darstellungen (in der Abbildung nummeriert) detaillierter erläutert.

1. Das Logo der Hochschule für Technik und Wirtschaft Berlin dient zur Veranschaulichung des Projektursprungs.
2. Die Temperatur wird mithilfe eines Thermometers sowie der letzten gemessenen Temperatur dargestellt. Alle aufgeführten Temperaturen werden in Grad Celsius angegeben.
3. Der zuletzt gemessene Luftdruck wird durch das Barometer gekennzeichnet. Die aufgeführten Luftdruckwerte werden in Hektopascal (hPa) angegeben.
4. Die Luftfeuchtigkeit wird mithilfe des Regentropfens widerspiegelt. Bei der Darstellung handelt es sich um die relative Luftfeuchtigkeit, die stets in Prozent angegeben wird.

5. Die Feinstaubampel veranschaulicht den zuletzt gemessenen Feinstaubwert. Sie kann drei verschiedene Darstellungen annehmen (grün, gelb, rot). Somit soll den vorbeilaufenden Passanten*innen auf den ersten Blick Auskunft darüber gegeben werden, inwiefern der angegebene Wert, gesundheitliche Probleme verursachen kann. Die verschiedenen Ausprägungen der Ampel werden gemäß Abbildung 1 berechnet. Es wird sich auf die Visualisierung der Ultrafeinen-Partikel (PM_{2,5}) konzentriert, die jeweils in Mikrogramm pro Kubikmeter angegeben werden.
6. Hier befindet sich das beworbene Medikament, welches die Symptome der ermittelten Erkrankung lindern soll. Alle Produktbilder, die in diesem Abschnitt dargestellt werden, stammen aus der Datenbank der Adler-Apotheke und wurden nach Absprache mit dem Geschäftsführer für das jeweilige Krankheitsbild verwendet. Eine Auflistung der Produkte lässt sich in Tabelle 1 betrachten.
7. Der Textabschnitt dient zur Erläuterung und Rechtfertigung warum das jeweilige Medikament empfohlen wird. Demnach verhält sich der hier angegebene Text, dynamisch gegenüber dem Produktbild. Zusätzlich soll den vorbeilaufenden Kunden*in deutlich dargestellt werden, warum sie möglicherweise an dem ausgewählten Beschwerden leiden.
8. Hier befindet sich der QR-Code der auf das Git-Repository verweist und die Möglichkeit darstellt, das Projekt selber nachzubauen. Passanten*innen die den QR-Code scannen werden zu folgender URL weitergeleitet: <https://github.com/mlinderm1993/weatherAd/blob/master/Documentation>

2.5 Backend

Die NodeJS App dient als Verbindungsstück zwischen Frontend, Datenbank und dem TTN Netzwerk. Zudem bindet die NodeJs App die öffentliche Schnittstelle `openuv.io` ein. Für die jeweiligen eingebundenen Schnittstellen sind Clients in Form von Services zuständig. Diese nehmen die Kommunikation und die Datenaufbereitung vor. So bindet beispielsweise der `InfluxClient` die Datenbank an und bereitet die aus der Datenbank ausgelesenen Daten zur Weiterverarbeitung auf. Instanzen der Services werden als Module exportiert und stehen dadurch innerhalb der Anwendung als Singleton zur Verfügung. Die Weiterverarbeitung aller Daten findet im `DataService` statt. Dort wird die Entscheidung getroffen welche Werbung im Frontend angezeigt wird. Lediglich das Frontend wird nicht über einen Client, sondern über den Router und eine einfachen HTTP Schnittstelle angebunden. Folgende Abbildung 9 soll diesen Kommunikationsweg besser darstellen,

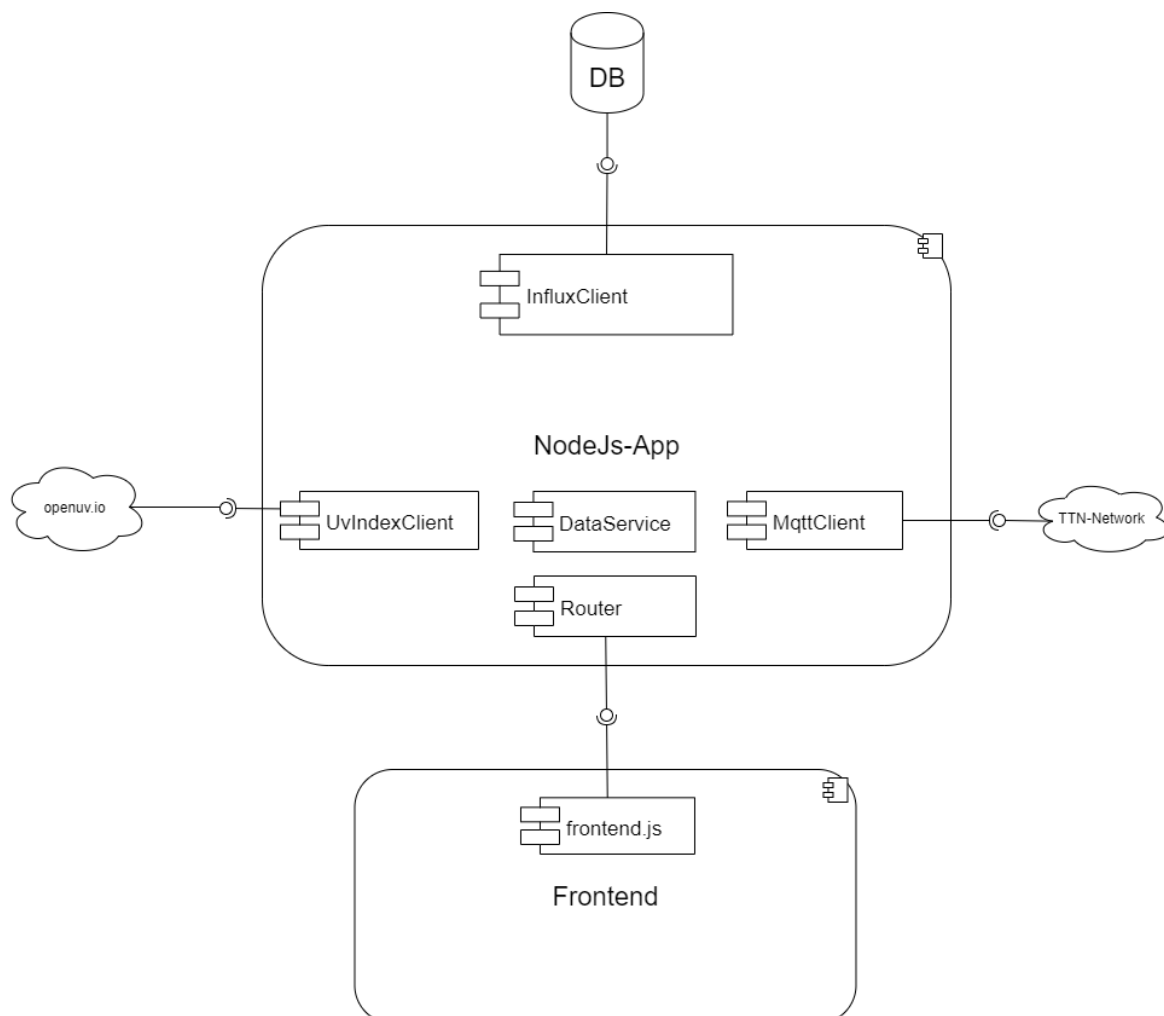


Abbildung 9: Kommunikationswege

Da ein Feinstaubsensor verbaut ist wurde auch Luftdaten.info an NodeJs App angebunden (Vgl. Abbildung 7 und 8). Auch dies geschieht über einen Client Service. Diese Architektur ist leicht erweiterbar und so können auch in Zukunft weitere Schnittstellen schnell eingebunden werden.

2.6 Kommunikationswege

Das Folgende Kapitel soll zeigen, wie die Kommunikation zwischen der Hardware und Software stattfindet. Dabei wird beim Hardwareseitigen Octopus begonnen und endet beim Frontend.

2.6.1 Kommunikation zwischen Octopus und TTN

Für die Kommunikation des IoT Octopus mit dem TTN⁵ wird die im Aufbau der Hardware genannte LoRaWAN-Antenne verwendet. Unter LoRaWAN versteht man Long Range Wide Area Network und ist ein Low-Power-Wireless-Netzprotokoll, welches speziell für das Internet der Dinge entwickelt wurde. Es ermöglicht kostengünstig batteriebetriebene Geräte mit dem Internet zu verbinden. Das Konzept des LoRaWAN Netzwerk ist darauf ausgelegt neben einem niedrigen Energieverbrauch auch noch eine Skalierbarkeit von Gateway-Installation bis zum großen globalen Netzwerk zu gewährleisten. [vgl.28]

Die Sensoren bzw. das Gerät wurde im nächsten Schritt im TTN registriert. So können die von den Sensoren erfassten Wetterdaten mittels des LoRaWAN an das TTN weitergeleitet werden.

2.6.2 Kommunikation zwischen TTN und NodeJS

Die dargestellte Kommunikation zwischen TTN und dem NodeJS Backend geschieht über das MQTT Protokoll. Hierzu registriert sich das NodeJS Backend bei der eigenen Initialisierung über einen Connect am TTN für das Topic "mwvt6/devices/iotoctopus2/up". Fortan werden die oben beschriebenen Messages vom IoT Octopus an das TTN durch das TTN an das NodeJS Backend weitergereicht. Damit das NodeJS die Daten weiterverarbeiten kann müssen diese aus der Payload noch extrahiert werden. Dies übernimmt ein Payload Decoder auf der TTN Plattform. Hier wird das Datenpaket vom IoT Octopus in seine Bestandteile zerlegt und zu menschenlesbaren Werten zusammengesetzt. Vgl. Abbildung 10.

⁵ <https://www.thethingsnetwork.org/>

```

1 function Decoder(bytes, port) {
2   var temperature = ((bytes[0] << 16) + (bytes[1] << 8) + (bytes[2]))/1000;
3   var humidity = ((bytes[3] << 16) + (bytes[4] << 8) + (bytes[5]) + 30000)/1000;
4   var pressure = ((bytes[6] << 16) + (bytes[7] << 8) + (bytes[8]))/1000;
5   var fineParts = ((bytes[9] << 16) + (bytes[10] << 8) + (bytes[11]))/1000;
6
7   return {
8     data: {
9       temperature: temperature,
10      pressure: pressure,
11      humidity: humidity,
12      fineParts : fineParts
13    }
14  };
15 }

```

Abbildung 10: Screenshot TTN Decoder

Um die Werte durch das NodeJS Backend einfacher auswerten zu lassen, werden die Werte im TTN nicht nur dekodiert sondern auch in ein JSON Format überführt. Somit erhält das NodeJS Backend ein definiertes JSON Objekt als Message (Siehe Abbildung 11).

The screenshot shows the TTN Uplink interface. Under the 'Uplink' section, the 'Payload' is displayed as a hexadecimal string: 00 12 E3 00 A2 22 11 22 AE 00 0F 2A. Below this, the 'Fields' section shows the decoded JSON object:

```

{
  "data": {
    "fineParts": 3.882,
    "humidity": 71.506,
    "pressure": 1122.99,
    "temperature": 4.835
  }
}

```

At the bottom, the 'Metadata' section shows the following JSON object:

```

{
  "time": "2021-01-24T08:02:23.306203747Z",
  "latitude": 52.436058,
  "longitude": 13.543183
}

```

Abbildung 11: Überführung JSON-Objekt

2.6.3 Kommunikation zwischen NodeJS und Datenbank sowie Frontend

Serverseitig wird das MQTT Protokoll mithilfe einer Opensource-Bibliothek implementiert (Vgl. Abbildung 12).

```
function MqttClient() {
  const mqtt = require('mqtt');
  const influxClient = require('./influxClient');
  const x = 0;

  const options = {
    password: '<PASSWORD>',
    username: '<USERNAME>',
    port: '<PORT>'
  };

  MqttClient.prototype.connectToTTN = () => {
    const client = mqtt.connect('<ADRESSE>', options);
    client.on('connect', function () {
      console.log('mqtt connect');
      client.subscribe('<TOPIC>', function (err) {
        if (err) {
          console.log('mqtt connect fehlgeschlagen');
          return;
        }
        console.log('mqtt connect erfolgreich');
      });
    });
  };

  client.on('message', (topic, message) => {
    const messageJson = JSON.parse(message.toString());
    writeStoredSensorData(messageJson.payload_fields, new Date(messageJson.metadata.time));
  });
}
```

Abbildung 12: Anbindung des The Things Networks mithilfe des MQTT-Protokolls

Nachdem die Daten im Backend angekommen sind werden diese sofort in die Datenbank geschrieben. Wie in Abbildung 13 zu sehen muss lediglich das Datum geparkt und die Daten in ein Pointformat gebracht werden.

```
function createPointFromData(data) {
  const point = new Point(measurement);
  point.floatField("temperature", data.temperature);
  point.floatField("fineParts", data.fineParts);
  point.floatField("humidity", data.humidity);
  point.floatField("pressure", data.pressure);
  point.timestamp(data.date);
  point.tag("weatherApi");
  console.log(point);
  return point;
}
```

Abbildung 13: Erstellung eines Points zur Speicherung in der Datenbank

Alle Sensordaten werden gebündelt in einem Point gespeichert und gleichzeitig an die Datenbank übertragen. Die Kommunikation findet im InfluxClient statt und es wird das offizielle node-influx Package genutzt.

```
You, seconds ago | 2 authors (you and others)
const { Point } = require("@influxdata/influxdb-client");
const { InfluxDB } = require("@influxdata/influxdb-client");

const token =
  "<token>";
const org = "<organisation>";
const bucket = "weatherAd";
const measurement = "weather";
You, 10 days ago • connected to db
function InfluxClient() {
  const client = new InfluxDB({
    url: "https://eu-central-1-1.aws.cloud2.influxdata.com",
    token: token,
  });
```

Abbildung 14: Anbindung der Influx Cloud Datenbank

Das Frontend fragt die zur Anzeige benötigten Daten alle 10 Sekunden ab dies geschieht über eine Standard HTTP GET Anfrage. Diese Anfrage wird durch den Router verarbeitet. Da das Frontend so einfach wie möglich gehalten ist, werden im Backend alle Entscheidungen getroffen. Dies bedeutet, dass das Backend die Datenanalyse vornimmt. Ein Teil der Analyse der Daten findet über die Influx-Datenbank statt. So wird die Varianz des Druckes schon mit der Datenbankabfrage berechnet (Vgl. Abbildung 15).

```
InfluxClient.prototype.getVariationOfPressure = () => {
  const query = `from(bucket:"${bucket}") |> range(start: -14d)
|> filter(fn: (r) => r._measurement == "${measurement}" and r._field == "pressure") |> spread`;
  return queryApi
    .collectRows(query)
    .then((row) => {
      return row._value;
    })
    .catch((err) => console.log(err));
};
```

Abbildung 15: Auswertung der maximale Schwankung des Druck mit Hilfe der InfluxDB

Die Varianz der Temperatur wird jedoch Tageszeit bezogen in der NodeJs-App berechnet. Nachdem die Daten so aufbereitet wurden, wird im DataService nach

den oben beschriebenen Kriterien das Werbebild und die Beschreibung ausgewählt (Vgl. Abbildung 16).

```
// Erkältung
if (
  coldTimeRange.indexOf(month) != -1 &&
  latestData.humidity >= 83 &&
  latestData.temperature <= 8
) {
  return { src: "img/prod_muedigkeit.jpg",
    desc: "Das aktuelle Wetter weist ein erhöhtes Risiko auf an Erkältungen zu leiden. Wir empfehlen:" };
}
// Kreislauf
if (
  isCirculatoryProblemPossible(latestData, pressureVariation, temperatureVariation, month) &&
  summerTimeRange.indexOf(month) != -1
) {
  return { src: "img/prod_kreislauf.jpg",
    desc: "Das aktuelle Wetter und die Schwankungen des Luftdrucks und der Temperatur erhöhen das Risiko für Kreislaufprobleme. Wir empfehlen:" };
}
// Grippe
if (
  isGrippe(latestData, pressureVariation, temperatureVariation, month) &&
  grippeTimeRange.indexOf(month) != -1
) {
  return { src: "img/prod_muedigkeit.jpg",
    desc: "Das aktuelle Wetter und die Schwankungen des Luftdrucks und der Temperatur erhöhen das Risiko für Müdigkeit. Wir empfehlen:" };
}
// Kopfschmerzen
if (headacheTimeRange.indexOf(month) != -1) {
  return { src: "img/prod_kopfschmerz.jpg",
    desc: "Aufgrund der aktuellen Luftdruckschwankungen entsteht ein erhöhtes Risiko an Kopfschmerzen zu leiden. Wir empfehlen:" };
}
if (uvIndexData >= 5) {
  return { src: "img/prod_uv.jpg",
    desc: "Es lässt sich ein erhöhter UV-Index feststellen. Achten Sie auf genügend Sonnenschutz. Wir empfehlen:" };
}
return { src: "img/prod_kopfschmerz.jpg",
  desc: "Derzeitig lassen sich keine wetterbedingten Krankheitsmuster erkennen." +
  "Bei der Darstellung handelt es sich um eine generelle Werbung" };

```

Abbildung 16: Auswahl des Krankheitsbilds.

Zudem wird dort die Feinstaubbelastung eingestuft (Vgl. Abbildung 17).

```
function getFinePartIndicator(fineParts) {
  // grün
  if(fineParts < 30 ) {
    return "img/traffic-light-green.png"
  }
  // gelb
  if(fineParts >= 30 && fineParts <= 45) {
    return "img/traffic-light-yellow.png"
  }
  // rot
  if(fineParts > 45) {
    return "img/traffic-light-red.png"
  }
}

```

Abbildung 17: Einstufung der Feinstaubbelastung

Nun werden die zuletzt gespeicherten Daten das Werbebild-Pfad und die Beschreibung sowie die Feinstaubampel-Pfad an das Frontend geschickt. Diese werden dort mit Hilfe von jQuery angezeigt.

3 Inbetriebnahme



Um den vollständigen Piloten in seinen Betrieb zu überführen müssen wenige Konfigurationen durchgeführt werden, sowie die Geräte an einem geeigneten Ort aufgestellt werden. Nachfolgend werden die dafür notwendigen Tätigkeiten dargestellt. Für die Durchführung ist zusätzlich zu den bereitgestellten Geräten und Kabeln folgendes Zubehör von Nöten:

- Tastatur
- Maus
- Befestigungsmaterial
- Sensorik

Für die Ermittlung sinnvoller Werte durch die Sensorik des Piloten ist der Standort von entscheidender Bedeutung. Dieser sollte außerhalb von geschlossenen Räumen gewählt werden. Wichtig ist hier vor allem ein Ort, der vor Sonne geschützt ist, da einfallende UV-Strahlung das Gehäuse der Sensorik erwärmt und somit die gemessenen Werte verfälscht. Ein Schutz gegen Wettereinflüsse wie Regen, Schnee und Wind ist durch das verwendete Gehäuse grundsätzlich gegeben, ein übermäßiger Einfluss sollte aber vermieden werden. Wichtig ist außerdem ein Standort, der eine gute Konnektivität zu einem LoRaWAN Gateway bietet.

Um die Sensorik in Betrieb zu nehmen, muss die Powerbank mit dem IoT Octopus verbunden werden, damit dieser hochfährt und die durch uns hinterlegte Firmware ausführt. Hierzu wird das USB -> MicroUSB Kabel einfach in den IoT Octopus gesteckt, dies geht am Einfachsten durch das Entfernen des oberen Entlüftungsstutzens. Hier kann bei Bedarf die Powerbank auch nachgeladen werden. Nach erfolgreichem Start des IoT Octopus muss der Entlüftungsstutzen wieder befestigt werden und das Gehäuse kann samt Sensorik am gewählten Ort befestigt werden. Dazu eignen sich beispielsweise Kabelbinder oder Lochband, um das Gehäuse an bestehenden Objekten wie Rohren oder Laternenpfosten zu befestigen. Alternativ kann das Gehäuse auch in Mauerwerk verankert werden, hier eignen sich Rohrschellen (Größe DN 75) mit entsprechend gewählten Stockschrauben.

Um Kundinnen und Kunden das bestmögliche Erlebnis zu bieten sollte das Display in einem Schaufenster aufgestellt werden, welches entsprechend sichtbar und frequentiert ist. Bei der Wahl des Standplatzes sollte auch berücksichtigt werden, wie der Sonneneinfall auf das Display im Laufe des Tages ist und ob dieses dadurch auch durchgängig erkennbar bleibt. Bevor das Display samt Raspberry PI fest installiert werden ist jedoch noch die Anbindung an das Vorherrschende WLAN durchzuführen. Hierzu wird das Display an Strom angeschlossen, sowie das HDMI Kabel an das Display und den Raspberry PI angeschlossen. Hierzu liegt ein Entsprechendes Kabel bei, da der Raspberry Pi mit einem HDMI Mini Anschluss versehen ist. Für notwendige Benutzereingaben werden außerdem eine Tastatur und eine Maus angeschlossen. Nun muss der Raspberry PI mit Strom versorgt werden, damit dieser startet. Hierzu das beiliegende Netzteil mit dem Gerät über USB-C verbinden. Der Raspberry PI startet nun, auf dem Display sollte der Bootvorgang dokumentiert sein. Das Gerät ist so eingestellt, dass es automatisch nach dem Start des Betriebssystems einen Browser startet und die in diesem Projekt hergestellte Webseite aufruft. Dieser Modus kann über "ALT + F11" verlassen werden und der Browser geschlossen werden.

Die Einrichtung der WLAN-Verbindung ist mittels Klick auf das Symbol  in der oberen rechten Ecke möglich. Nach Auswahl des WLANs und Eingabe des Kennwortes sollte sich das Symbol  geändert haben. Die WLAN-Konfiguration ist damit abgeschlossen.

Nach erfolgreicher Konfiguration des WLANs kann der Raspberry PI heruntergefahren werden. Ein Abziehen und erneutes Einstecken des Stromkabels im Raspberry PI führt zum Start des Raspberry PIs. Dieser sollte nun wieder den Browser starten und diesmal die Webseite inkl. aktueller Wetterdaten und einer Produktempfehlung zeigen. Alternativ bitte die WLAN Konfiguration erneut prüfen. Ist dieser Test erfolgreich beendet, kann das Display und der Raspberry Pi am gewählten Standort aufgebaut werden. Eine Verbindung von Tastatur und Maus ist nun nicht mehr von Nöten.

4 Problemstellungen während der Realisierung

In den folgenden Punkten wird auf die Probleme, die während der Realisierung des Prototypens eingetreten sind, eingegangen.

4.1 Covid-19

Die Grundlegenden Covid-19 Beschränkungen hat diese Gruppenarbeit eingeschränkt. Zum einen war es nur möglich die Hardware von einem Gruppenmitglied zu bearbeiten und weiterhin war der Bau des Prototypen so natürlich auch nur eingeschränkt möglich. Jedoch konnte dies mittels virtuellen Session zum Großteil gut bewerkstelligt werden.

4.2 Reichweite LoRaWAN Gateway

Bei der Abholung der Hardware zum Start des Projektes wurde das Gruppenmitglied ausgewählt, welches am nächsten an einem LoRaWAN Gateway wohnt. Leider war es bei einer Entfernung von 2,38 km Luftlinie nicht möglich Tests direkt vom Wohnort zu realisieren. Auf Grund dessen waren wir gezwungen immer wieder mittels öffentlichen Nahverkehr in die Nähe des Gateways zu fahren. Dies erschwerte den Bau des Prototyps erheblich.

4.3 Deaktivierung der NeoPixel des IoT Octopus

Für folgenden Problemstellungen wurde aktuell noch keine Lösung implementiert.

Im Betrieb des IoT Octopus mit LoRaWAN-Antenne auf der Feather Wing schalten sich die NeoPixel automatisch ein. Verschiedene Ansätze wie Veränderung der PIN-Konfiguration von NeoPixel und LoRaWAN, sowie mehrfaches Setzen einer Deaktivierung der NeoPixel brachten keinen Erfolg. Ein offener Lösungsansatz ist das generelle Deaktivieren der NeoPixel durch die Unterbrechung der Lötbrücke SJ9 auf dem IoT Octopus (Siehe Abbildung 18).⁶

⁶ Vgl. <https://www.umwelt-campus.de/iot-werkstatt/community>

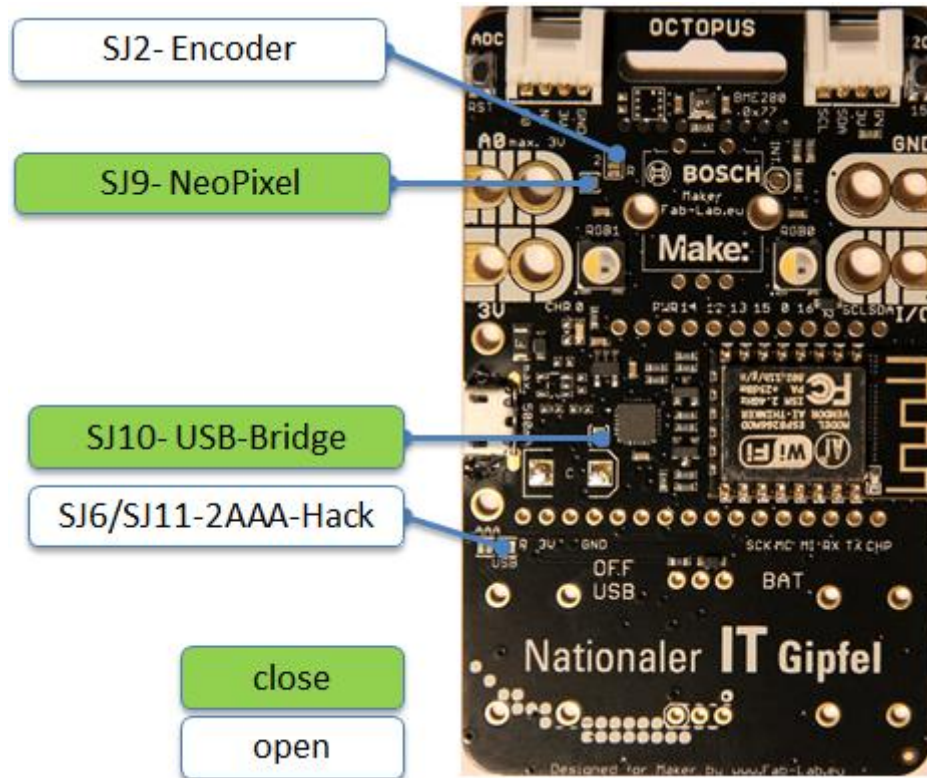


Abbildung 18: Vorderseite Octopus

Dies würde für nachfolgende Gruppen bedeuten, dass die Lötbrücke wiederhergestellt werden muss, wenn im Projekt die NeoPixel verwendet werden sollen.

5 **Ausblick**

Inzwischen ist mittlerweile vielen klar, dass der von dem in unserem Projekt eingesetzte Raspberry Pi viele Vorteile mit sich bringt. Eine preiswerte und kompakte Hardware, welche die Möglichkeit bietet, zahlreiche Schnittstellen anzubinden. So kann es z. B. in unserem Projekt möglich sein auch andere Daten (beispielsweise Werte des Robert-Koch-Instituts) anzubinden umso weitere Krankheitsbilder abzubilden und Medikamente bzw. auch Verhaltensempfehlungen vorzuschlagen. Die Entwickler der Pi-Reihe bringen mit jeder Generation mehr Leistung bei geringerem Stromverbrauch für die Lösung bereit. So bietet er hier auch die Möglichkeit auf mehrere Bildschirme zu erweitern. Der Erweiterbarkeit ist so kaum eine Grenze gesetzt und das bei einem Produkt, dass ins Leben gerufen wurde, um der Bildung neue Möglichkeiten in der Informatik und Elektronik zu fördern.

6 **Fazit**

Das Projekt und der Prototyp für die Wetterstation hatte zum Ziel die Sensordaten Temperatur, Luftfeuchtigkeit, Luftdruck und Feinstaub bereitzustellen, auszuwerten, darzustellen und schlussendlich passende Medikamente herzuleiten. Als Übertragungsmedium wurde das beschriebenen LoRaWAN genutzt. Das ist dann von Vorteil, wenn kleine Datenmengen drahtlos mit geringem Stromverbrauch übertragen werden sollen. In unserem Fall sind also die perfekten Voraussetzungen um die Wetterstation als Prototypen umzusetzen.

Die Frage, die man sich stellen muss, ist, ob das Bereitstellen der Daten selbst mittels Sensorik notwendig ist. Da es für dieses Projekt genug Quellen gibt, um diese anderweitig bereitzustellen kann man hier überlegen. Jedoch sind Daten die direkt Ortsbezogen sind vielleicht auch passender.

Abschließend ist zu sagen, dass Projekte wie diese, der Erweiterbarkeit und Kreativität in Bezug auf das Internet der Dinge viel Spielraum lassen. Demnach entstand in der Projektarbeit und in unserem Team ein positiver Eindruck in Bezug auf die Thematik rund um das IoT.

7 Literaturverzeichnis

- [1] Koppe, C., S. Zacharias & D. Bernhard, "Repräsentativbefragung zur Wetterfähigkeit in Deutschland. Bericht zum Forschungsvorhaben des Umweltbundesamtes UFOPLAN 3711 61 238.,"
- [2] "Wann ist die Erkältungssaison vorbei?," *Medlanes*, 16 Jan., 2018.
<https://medlanes.com/magazin/erkaeltung/wann-ist-die-erkaeltungssaison-vorbei/> (accessed: Jan. 24 2021).
- [3] *RKI - Navigation - Wann spricht man von Grippesaison, wann von Grippewelle?* [Online]. Available: <https://www.rki.de/SharedDocs/FAQ/Influenza/FAQ26.html> (accessed: Jan. 24 2021).
- [4] Bundesamt für Strahlenschutz, *Wie wirkt UV-Strahlung?* [Online]. Available: <https://www.bfs.de/DE/themen/opt/uv/wirkung/einfuehrung/einfuehrung.html> (accessed: Jan. 24 2021).
- [5] H. Li *et al.*, "Particulate Matter Exposure and Stress Hormone Levels: A Randomized, Double-Blind, Crossover Trial of Air Purification," *Circulation*, vol. 136, no. 7, pp. 618–627, 2017, doi: 10.1161/CIRCULATIONAHA.116.026796.
- [6] Daswetter.com | Meteored, *Wetterrückblick Berlin - daswetter.com*. [Online]. Available: https://www.daswetter.com/wetter_Berlin-Europa-Deutschland-Berlin--sactual-26301.html (accessed: Jan. 24 2021).
- [7] "Wetterlexikon: Erkältungswetter," *wetter.de*, 27 Jan., 2020. <https://www.wetter.de/cms/wetterlexikon-erkaeltungswetter-4476486.html> (accessed: Jan. 26 2021).
- [8] NDR, "Wenn das Wetter auf die Gesundheit schlägt," *NDR*, 03 May., 2016.
<https://www.ndr.de/ratgeber/gesundheit/Gesundheitliche-Probleme-durch-Wetterwechsel,wetterwechsel101.html> (accessed: Jan. 26 2021).
- [9] M. Riederer, "Wetterfähigkeit: Wenn der Kreislauf verrückt spielt," *BR24*, 27 Apr., 2020. <https://www.br.de/nachrichten/wissen/wetterfuehligkeit-kreislaufprobleme-und-kopfweh-durch-foehn,RL43YH7> (accessed: Jan. 26 2021).
- [10] K. Klein, "Wetterfähigkeit: Das hilft, wenn das Wetter krank macht," *FIT FOR FUN*, 09 Apr., 2018. <https://www.fitforfun.de/beauty/wellness/wetterfuehligkeit-wie-schafft-man-abhilfe-158992.html> (accessed: Jan. 26 2021).
- [11] *Wenn der Wetterumschwung krank macht*. [Online]. Available: <https://www.hkk.de/themen/behandeln/wenn-der-wetterumschwung-krank-macht> (accessed: Jan. 26 2021).
- [12] Apotheken Umschau, *Wetterfähigkeit – was steckt dahinter?* [Online]. Available: <https://www.apotheken-umschau.de/Wetterfuehligkeit> (accessed: Jan. 26 2021).
- [13] zm-online, *Das Wetter und das Virus*. [Online]. Available: https://www.zm-online.de/archiv/2020/23_24/medizin/das-wetter-und-das-virus/ (accessed: Jan. 26 2021).
- [14] *Können Wetterschwankungen eine Grippewelle verschlimmern?*: www.kinderaerzte-im-netz.de. [Online]. Available: <https://www.kinderaerzte-im-netz.de/news-archiv/meldung/article/koennen-wetterschwankungen-eine-grippewelle-verschlimmern/> (accessed: Jan. 26 2021).

- [15] Apotheken Umschau, *Sonnenbrand-Risiko nach Hauttyp*. [Online]. Available: https://www.apotheken-umschau.de/Haut/Sonnenbrand-Risiko-nach-Hauttyp--69645_3.html (accessed: Jan. 26 2021).
- [16] *Wetter und Klima - Deutscher Wetterdienst - Leistungen - UV-Gefahrenindex*. [Online]. Available: <https://www.dwd.de/DE/leistungen/gefahrenindizesuvi/gefahrenindexuvi.html> (accessed: Jan. 26 2021).
- [17] Kopfschmerzen und Migräne - Ursachen und Behandlung, *Wetter-Kopfschmerzen? Werden Sie wetterfit!* [Online]. Available: <https://www.kopfschmerzen.de/kopfschmerzen/kopfschmerz-ursachen/wetter> (accessed: Jan. 26 2021).
- [18] Mdr.de, *Tiefdruck-Wetter ist Kopfschmerz-Zeit - Was hilft? | Das Erste*. [Online]. Available: <https://www.mdr.de/brisantratgeber/wetter-sturm-kopfschmerzen-100.html> (accessed: Jan. 26 2021).
- [19] *Air Quality Now - Über uns - Definition der Indizes*. [Online]. Available: https://www.airqualitynow.eu/de/about_indices_definition.php (accessed: Jan. 26 2021).
- [20] Kachelmannwetter - Jetzt Lesezeichen setzen, *PM2.5 - Feinstaub, Messwerte Berlin (Stadt) vom 14.01.2021, 16:00 Uhr | Messstationen Luftqualität*. [Online]. Available: <https://kachelmannwetter.com/de/luftqualitaet/stadt-berlin/pm25-feinstaub/20210114-1500z.html> (accessed: Jan. 26 2021).
- [21] Statista, *Durchschnittstemperatur in Deutschland pro Monat 2020 | Statista*. [Online]. Available: <https://de.statista.com/statistik/daten/studie/5564/umfrage/monatliche-durchschnittstemperatur-in-deutschland/> (accessed: Jan. 26 2021).
- [22] WetterOnline, *UV-Index Deutschland - Sonnenbrandgefahr aktuell - WetterOnline*. [Online]. Available: <https://www.wetteronline.de/uv-index> (accessed: Jan. 26 2021).
- [23] WetterOnline, *Wetter aktuell, Wettervorhersage und Wetterbericht - WetterOnline*. [Online]. Available: <https://www.wetteronline.de/> (accessed: Jan. 26 2021).
- [24] Laenderdaten.info, *Länderdaten - Die Welt in Zahlen*. [Online]. Available: <https://www.laenderdaten.info/> (accessed: Jan. 26 2021).
- [25] Aero-Check, *Die relative Luftfeuchtigkeit in Deutschland (mit Tabelle)*. [Online]. Available: <https://aero-check.de/luftfeuchtigkeit/deutschland/> (accessed: Jan. 26 2021).
- [26] Daswetter.com | Meteored, *Wetterrückblick Berlin - daswetter.com*. [Online]. Available: https://www.daswetter.com/wetter_Berlin-Europa-Deutschland-Berlin--sactual-26301.html (accessed: Jan. 26 2021).
- [27] Laenderdaten.info, *Klima und Temperaturentwicklung in Deutschland*. [Online]. Available: <https://www.laenderdaten.info/Europa/Deutschland/Klima.php> (accessed: Jan. 26 2021).
- [28] LoRa-WAN Technology, *Was ist LoRaWAN?* [Online]. Available: <https://www.lora-wan.de/> (accessed: Jan. 24 2021).