



Jugend forscht
2024



Das Hochwasserwarnsystem

Geo- und Raumwissenschaften

Niklas R.

Jana S.

**Schülerforschungszentrum Südwestfalen
Standort Olschhausen**

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Kurzfassung	2
Einleitung	3
Motivation	4
Was ist ein Hochwasser?	4
Anforderungen an ein einsetzbares System	5
Funktionsweise	5
Diskussion der einzelnen Komponenten	6
Übertragungsprotokoll	6
Sensor zur Wasserhöhenmessung	6
Probleme des alten Designs	7
Neue Idee	7
Neue Designs	8
Alternativen	9
Datenspeicherung / Application-Server	9
Warnalgorithmus	11
Alarmfunktion	12
E-Mail	12
Existierende Warnapps	12
Eine eigene Warnapp	12
Standortauswahl	13
Betrachtung der Messdaten	15
Danksagung	16
Literatur- und Bildquellenverzeichnis	17

Kurzfassung

Die präzise Vorhersage lokaler Überschwemmungen wird aufgrund der Klimawandelauswirkungen zunehmend bedeutsam. Modelle prognostizieren verstärktes Auftreten von Starkregen, was zu dezentralen Überschwemmungen, auch an kleinen Gewässern, führt. In Ochsenhausen kam es wiederholt zu unvorhergesehenen Hochwasserereignissen. Wir entwickeln daher ein lokales, kostengünstiges und vielseitiges Hochwasserwarnsystem. Es nutzt kleine, fast autarke Messstationen, erhebt in Echtzeit Daten aus den Zuläufen der Rottum, analysiert diese und gibt präzise Warnungen aus. Das System ist problemlos an andere Einsatzgebiete anpassbar. Wasserstands-Daten werden über LoRaWAN zum Server gesendet und mit eigenen Warnalgorithmen verarbeitet. Letztes Jahr präsentierten wir einen Prototypen, nun haben wir das System in die Realität umgesetzt, weiterentwickelt und eine Handy-App für Warnungen erstellt. Unser Ziel ist es, bis zum Jugend-forscht-Wettbewerb ein funktionsfähiges Hochwasserwarnsystem vorzustellen.

Einleitung

Mit dem fortschreitenden Klimawandel prognostiziert der Weltklimarat, dass sowohl die Häufigkeit als auch die Intensität von Fluten und Überschwemmungen zunehmen werden [1]. Dies wurde vom Intergovernmental Panel on Climate Change (IPCC) in ihrem Bericht über den Klimawandel bestätigt:

„Der vom Menschen verursachte Klimawandel wirkt sich bereits auf viele Wetter- und Klimaextreme in allen Regionen der Welt aus. Seit dem Fünften Sachstandsbericht (AR5) gibt es stärkere Belege für beobachtete Veränderungen von Extremen wie Hitzewellen, Starkniederschlägen, Dürren und tropischen Wirbelstürmen sowie insbesondere für deren Zuordnung zum Einfluss des Menschen.“ [2]

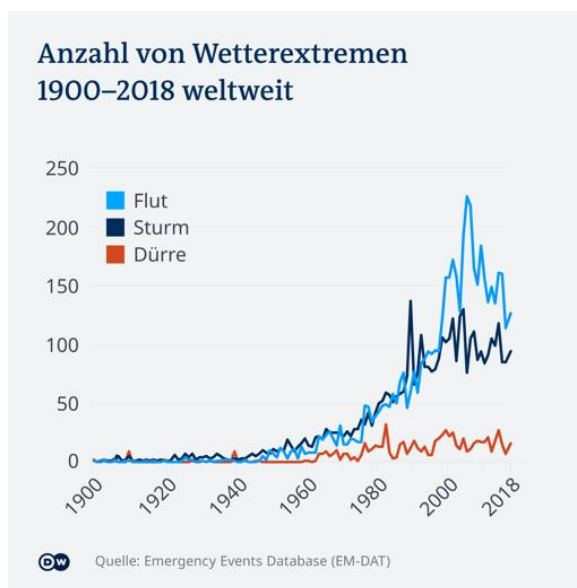


Abbildung 1: Wetterextreme 1900-2018

Die Anzahl von Wetterextremen hat im letzten Jahrhundert durch die Auswirkungen des Klimawandels weltweit stark zugenommen (Abb. 1). Auch in Deutschland sind diese Auswirkungen stark zu spüren. Die aktuelle Flut, ausgelöst durch anhaltende Starkregenereignisse entlang der Elbe, hat zu erheblichen Überschwemmungen und Infrastrukturschäden geführt. Zahlreiche Gebiete entlang des Flussverlaufs sind betroffen, wobei lokale Evakuierungen notwendig wurden [3]. Auch unsere Region sah sich in den vergangenen Jahren wiederholt mit unterschiedlich starken Überschwemmungen konfrontiert. Sogar unsere Schule erlitt 2016

beträchtliche Schäden durch Hochwasser, was zu erheblichem Sachverlust und dem Ausfall von Unterricht führte (Abb. 2) [4]. All dies hätte womöglich verhindert werden können, wenn alle Bewohner rechtzeitig einen Warnhinweis erhalten hätten. Trotz Bemühungen im Klimaschutz sowie dem Einsatz von Deichen und Rückhaltebecken ist davon auszugehen, dass ein effektives Frühwarnsystem in Zukunft nötig sein wird, um derartige Katastrophen zu vermeiden [5].

Motivation

Abgesehen von der Flutkatastrophe im Ahrtal 2021 kam es auch im letzten Jahr zu weiteren erheblichen Überschwemmungen. Beispielsweise das Weihnachtshochwasser 2023, wo es zu erhöhten Wasserständen, vor allem in Teilen Norddeutschlands wie Niedersachsen, Sachsen-Anhalt, Thüringen und Nordrhein-Westfalen kam. Deshalb haben wir uns das Ziel gesetzt, ein Hochwasserwarnsystem zu entwickeln, das bei Überschwemmungsgefahr einen Warnhinweis an Anlieger sendet. Um die Kosten niedrig zu halten, setzen wir auf die Anwendung eines LoRaWAN-Infrarotsensors zur präzisen Messung des Wasserstands in Flüssen. Die erfassten Daten werden effizient mithilfe des energiesparenden LoRaWAN-Protokolls an ein The-Things-Network-Gateway übermittelt. Von dort aus erfolgt die Weiterleitung über das MQ-Telemetry-Transport-Protokoll (MQTT) an einen Server, wo die Speicherung und Analyse stattfinden. Bei Überschreitung eines vorab festgelegten Grenzwerts ist die geplante Benachrichtigung über eine spezielle App vorgesehen, welche sämtliche gefährdete Einwohner in einem betroffenen Gebiet erreicht. Diese Maßnahme gewinnt an Bedeutung aufgrund der rechtlichen Verpflichtung gemäß Wasserhaushaltsgesetz (WHG, §78a) in der Kommune Ochsenhausen, in der Bewohner, die sich in einem Überschwemmungsgebiet befinden, dazu verpflichtet sind, sich und ihre Wertgegenstände im Falle einer drohenden Überschwemmung in Sicherheit zu bringen [6]. Durch frühzeitige Warnungen könnte im Ernstfall das Risiko von Verletzten oder Toten erheblich reduziert werden, was einen positiven Beitrag zur Bewältigung der potenziell dramatischen Auswirkungen des Klimawandels leisten würde.

Was ist ein Hochwasser?

Der Begriff "Hochwasser" bezieht sich auf einen Zustand, in dem der Pegelstand eines Gewässers erheblich über seinem normalen, durchschnittlichen Niveau liegt. Diese Definition erfolgt präzise anhand des HQ-Werts eines Gewässers, wobei "HQ" für die Abflussmenge bei Hochwasser steht. Die Kennzahl setzt sich aus "H" für "Hochwasser" und "Q" für die Abfluss-Kennzahl zusammen. Der HQ-Wert wird im Verhältnis zur statistischen Häufigkeit eines beobachteten Pegelstands angegeben, was die Anzahl der Jahre angibt, in denen ein bestimmter Pegelstand statistisch betrachtet auftreten kann (Jährlichkeit). Zum Beispiel wird ein Hochwasser mit HQ100 statistisch gesehen einmal alle 100 Jahre erwartet und wird als 100-jährliches Ereignis bezeichnet, auch bekannt als "Jahrhundertflut". [7]



Abbildung 2: Schäden durch das Hochwasser 2016 in Ochsenhausen. Links: Straßenschäden durch das Hochwasser, Rechts: vom Hochwasser betroffener Flur am Gymnasium Ochsenhausen

Anforderungen an ein einsetzbares System

Ein System, das die Situation von Gewässern regional überwacht und bei Starkwetterereignissen sowie drohenden Überflutungen Alarm auslöst, sollte nach unserer Auffassung folgende Anforderungen erfüllen:

- Es muss **kostengünstig** sein, um die Komponenten in größeren Mengen herstellen zu können. Angesichts der aktuellen Bedrohung großer Flächen durch Überflutungen, die voraussichtlich in Zukunft noch zunehmen werden, ermöglicht eine kosteneffiziente Lösung eine rasche Ausdehnung des Anwendungsbereichs. Bisherige kommerzielle Systeme zur lokalen Überwachung von Hochwasser sind aufgrund ihrer hohen Kosten nur an wenigen Standorten im Einsatz.
- Es muss möglichst **transportabel** sein, da das Messsystem auch an schwer zugänglichen Orten installiert werden muss. Eine einfache und schnelle Montage ist daher essenziell.
- Es muss **autonom** funktionieren, ohne auf eine Internetverbindung angewiesen zu sein, da nicht an allen vorgesehenen Standorten eine entsprechende Netzabdeckung für WLAN oder Mobilfunk besteht.
- Es muss **wartungsarm** sein, da regelmäßige Wartungsarbeiten mit erheblichem zeitlichem und materiellem Aufwand verbunden sind.
- Es muss sehr **stabil** sein, da es dauerhaft der Witterung ausgesetzt sein wird.

Funktionsweise

Diesen Anforderungen entsprechend haben wir das System wie folgt geplant (Abb. 3):

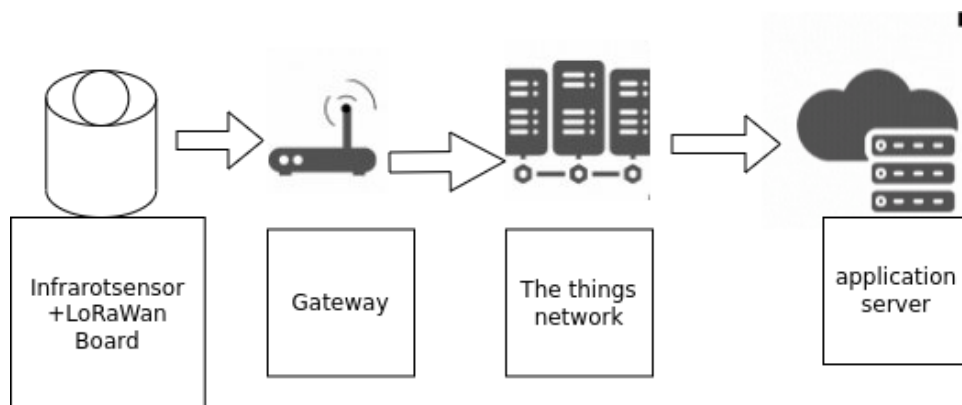


Abbildung 3: Schematische Darstellung der Einzelkomponenten des Hochwasserwarnsystems

LoRaWAN-Protokoll: Die Bezeichnung LoRaWAN steht für Long Range Wide Area Network und wurde speziell für das Internet der Dinge (IoT) entwickelt. Diese Technologie ermöglicht eine energieeffiziente Übertragung von Daten über große Entfernungen, insbesondere in Gebieten, in denen Mobilfunknetze keine ausreichende Abdeckung bieten [8]. Für unser Projekt ist diese Technologie optimal, weil es uns erlaubt, kostengünstig, energiesparend und über große Distanzen Daten zu übertragen.

Gateway: Die Bezeichnung Gateway bezieht sich auf eine Hardware- oder Softwarekomponente, die eine Verbindung zwischen zwei verschiedenen Systemen herstellt. Gateways fungieren als Umwandler oder Vermittler und ermöglichen die Kommunikation zwischen nicht direkt kompatiblen Systemen [9]. In unserem Fall nutzen wir ein Gateway als Vermittler zwischen dem LoRaWAN-Board und dem The

Things Network, also dem Internet. Die Daten können dabei von jedem beliebigen Gateway empfangen werden, das im The Things Network registriert ist. Solche Gateways werden beispielsweise von Unternehmen, Hochschulen oder Privatpersonen aufgestellt und stehen für eine freie Nutzung zur Verfügung.

The Things Network (TTN): Das TTN (The Things Network) ist ein Netzwerk, das Daten von Sensoren, die über eine LoRaWAN-Gateway-Verbindung übertragen werden, aufnimmt und diese über das Internet zugänglich macht. Die Besonderheit liegt in seinem Open-Source-Konzept und der kostenfreien Nutzung [10]. Diese Technologie erlaubt es uns, ortsunabhängig auf unsere Daten zuzugreifen.

Application-Server: Da 'The Things Network' die Daten nur für 24 Stunden speichert und nicht auswerten kann, leiten wir sie über die TTN API zu einem Third Party Server weiter, auf dem die Datenanalyse durchgeführt wird [11].

Diskussion der einzelnen Komponenten

Übertragungsprotokoll

Wir haben uns für das Long Range Wide Area Network (LoRaWAN) entschieden, da es stromsparend ist, nicht auf eine Internetverbindung angewiesen ist und über eine Reichweite von ca. 10 km verfügt. Zusätzlich sind die Boards (also die Sender), in unserem Fall das ELV-LW-Base, sehr kostengünstig.

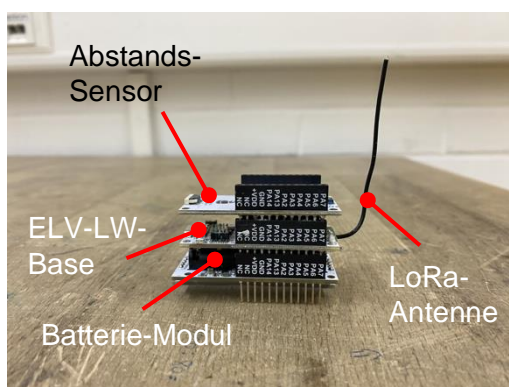


Abbildung 4: Aufbau des Sensor-Moduls

Sensor zur Wasserhöhenmessung

Für die Messung des Wasserpegels verwenden wir einen Infrarot-Sensor (ELV Distance 1 [12]). Dieser kann zusammen mit einem Batteriemodul einfach auf die ELV-LW-Base aufgesteckt werden (Abb. 4). Zu Beginn haben wir den Sensor mit Hilfe einer Plattform auf einem Rohr platziert und dieses dann so im Wasser befestigt, dass das Wasser ungehindert von unten in das Rohr fließen kann, wodurch der Wasserstand im Rohr dem Pegel des Gewässers entspricht (Abb. 5, 6).

Der Infrarot-Sensor misst den Abstand zwischen Sensor und der Wasseroberfläche. Diese Messung ermöglicht Rückschlüsse auf die Höhe des Wasserspiegels im Gewässer (Abb. 5). Um auf die absolute Höhe des Wasserspiegels zu kommen, müsste man natürlich zusätzlich die Höhe des Sensors kennen. Diese Messung ist allerdings für eine Hochwasser-Warnung nicht notwendig, da unser Warnalgorithmus lediglich die relative Veränderung des Wasserstands betrachtet. Dies bringt uns zusätzlich den Vorteil, dass wir die Breite des Flusses an dieser Stelle nicht berücksichtigen müssen. Zunächst hatten wir Bedenken, dass die Reflexionseigenschaften des Wassers die Genauigkeit des Infrarot-Sensors beeinträchtigen könnten. Theoretisch sollte dies bei klarem Wasser kein Problem sein (Abb. 7), in einem Bach ist das Wasser aber nicht immer so klar. Glücklicherweise erwiesen sich diese Bedenken nach unseren Versuchen als unbegründet.

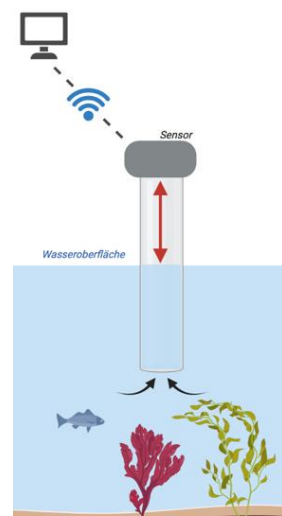


Abbildung 5: Aufbau mit Rohr

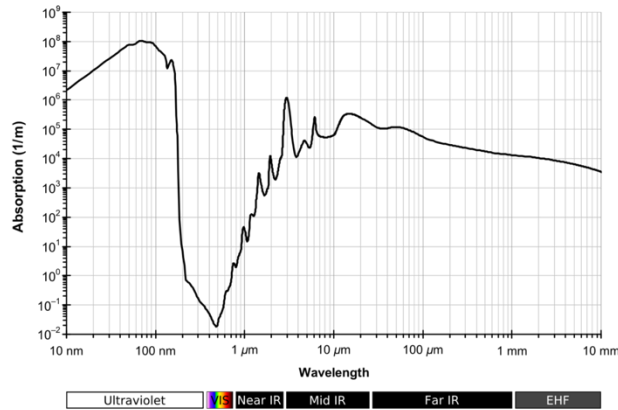


Abbildung 6: Test-Aufbau mit Rohr Abbildung 7: Absorptions-Spektrum von Wasser

Probleme des alten Designs

Das zuvor beschriebene Design ließ uns allerdings auf verschiedene Probleme stoßen: Letztes Jahr war der Plan, die Rohre an Bäumen, Pfählen oder etwas anderem Vertikalem zu befestigen. Nun zeigte sich allerdings, dass dies zwar bei einer Demo an einem kleinen Bach neben unserem Wasserrad funktioniert (Abb. 6), im echten Leben aber absolut unpraktisch ist. Die Probleme mit unserem Design waren:

1. Durch das Rohr ist unsere Messtation unnötig groß und schwer.
2. Durch Tiere, die sich im Rohr einnisten, können die Messergebnisse verfälscht werden.
3. In der Stadt oder an anderen Punkten, wo man den Sensor schlecht verstecken kann und viele Menschen am Bach entlanglaufen beeinträchtigen sie das Landschaftsbild und sind Vandalismus ausgesetzt.
4. Es gibt sehr wenige Bäume, an denen wir unsere alte Messtation mit dem Rohr sicher festmachen können, da die Bäume selten direkt im Wasser stehen oder weit genug überhängen.
5. Um die Messstation aufzustellen, war die alternative Idee, Pfähle zu verwenden. Allerdings wird aufgrund von Erosion der Boden unter dem Pfahl durch die ständige Strömung des Wassers abgetragen, wodurch der Pfahl nicht mehr fest im Boden verankert ist.

Neue Idee

Aufgrund der fehlenden Befestigungsmöglichkeiten und der anderen genannten Nachteile mussten also ein neues Design und eine neue Idee her. Als wir durch den Bach liefen und überlegten, wo man die Messstationen sonst noch befestigen konnte, kam uns die Idee:

Brücken bieten eine perfekte Befestigungsmöglichkeit, da unter den meisten Brücken Metallstangen und Rohre verlaufen, an denen man die Sensoren befestigen kann. Zusätzlich sind Brücken meist höher als Pfähle, wodurch der Sensor besser vor Wasser geschützt ist und offensichtlich deutlich langlebiger. Dies brachte uns dazu, unser Sensor-Design mehrmals zu überarbeiten. (Abb. 8)

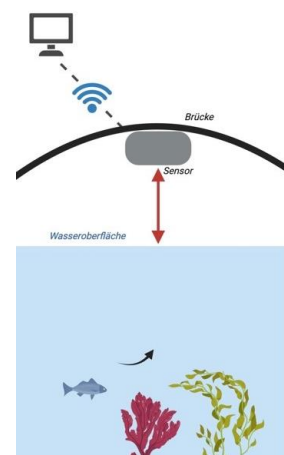


Abbildung 8: Neues Design ohne Rohr

Neue Designs



Abbildung 9: Befestigung des alten Designs (ohne Rohr)

Beim zweiten Design haben wir den oberen Teil der alten Messstation angepasst und mit Löchern für die Kabelbinder versehen. Somit kann das neue Design schnell und einfach vertikal befestigt zu werden (Abb. 10).



Abbildung 10: Zweites Design mit neuer Befestigung

Dieses Design funktioniert sehr gut, schnell und hat auch noch die Möglichkeit, auf einem Rohr festgemacht zu werden, sollte man diese Funktionalität doch brauchen, z. B., um den Sensor in bestimmten Einsatzgebieten zusätzlich zu schützen. Dieser Fall passiert allerdings sehr selten und wenn man es nicht auf ein Rohr setzt, ist dieses Design unnötig groß und verbraucht viel 3D-Druck-Filament.

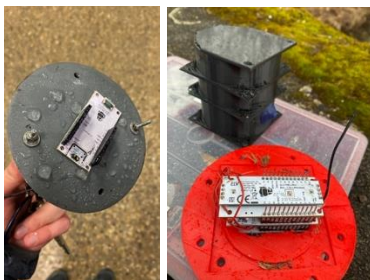


Abbildung 11: Sensoren, in die Wasser, Insekten und Schmutz eingedrungen sind

Zusätzlich haben wir bemerkt, dass sich in drei von fünf Messtationen Insekten eingenistet hatten und bei einer Station, die sich zu nah am Wasser befand, die Platine leicht oxidiert ist und offensichtlich in Kontakt mit Wasser kam (Abb. 11, in allen Fällen funktionierte der Sensor zum Glück noch).

Infolgedessen haben wir ein neues Design entwickelt (Abb. 12), das nicht nur deutlich kompakter ist, sondern auch den Infrarotsensor hinter einer Plexiglas-Scheibe platziert hat. Diese beeinträchtigt die Messung nicht, da unser Sensor bei 940nm arbeitet [13] und Plexiglas bei 940nm eine Durchlässigkeit von 94% Prozent besitzt [14].

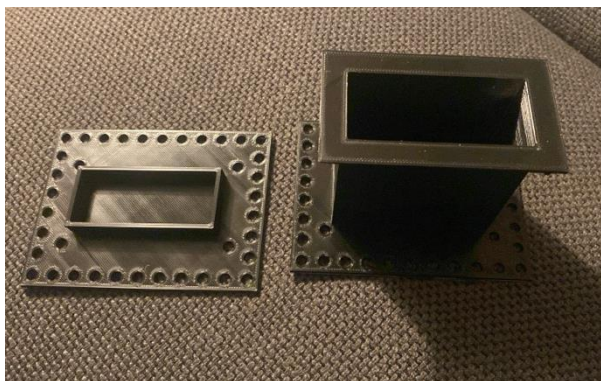


Abbildung 12: Neues Gehäuse

Alternativen

Alternativ zu einem Infrarotsensor könnten wir einen Schwimmkörper benutzen. Dieser hat im Vergleich allerdings nur Nachteile, er ist komplizierter, teurer und anfälliger für Fehler. Wir haben auch eine Ultraschall-Messstation aus einem ATmega328P-Mikrocontroller, einem Dragino und einem Ultraschallsensor gebaut, uns allerdings dann für die Infrarot-Messung entscheiden, da es dafür fertige Boards gibt. Diese können deutlich einfacher in Betrieb genommen werden und sind bereits auf einen energieeffizienten Betrieb optimiert.

Ein herausragender Vorteil unseres Messsystems besteht in seinem niedrigen Preis von etwa 50 € pro Messstation. Im Gegensatz dazu sind vorhandene kommerzielle Hochwasserwarnsysteme aufgrund ihrer hohen Kosten nur an wenigen Standorten im Einsatz. Unser Ansatz bietet somit die Möglichkeit, den Anwendungsbereich nahezu beliebig zu erweitern, ohne dabei hohe Kosten zu verursachen. Darüber hinaus ist unser Messsystem flexibel einsetzbar und kann ohne großen Aufwand an neue Einsatzorte verlagert werden.

Datenspeicherung / Application-Server

Da The Things Network die Daten nicht dauerhaft speichert, müssen die Daten für die Auswertung an einen eigenen Server weitergeleitet und dort gespeichert werden. Für diesen Fall bietet TTN sogenannte "Integrations" an. Hier können wir zwischen der TTN-eigenen API, WebHooks, dem MQTT-Protokoll und vielen weiteren Optionen wählen.

Zu Beginn haben wir WebHooks mit dem kostenfreien Storage-Anbieter "bplaced" verwendet, um die Messwerte zu speichern. Aufgrund verschiedener Einschränkungen und der fehlenden Möglichkeit, unseren Auswertungs-Algorithmus auf dem Server laufen lassen zu können, war dies jedoch keine passende Option.

Im letzten Jahr haben wir einen Raspberry Pi 4 als Server verwendet. Zwar ist ein solcher Einplatinencomputer sehr preisgünstig, bietet jedoch für kritische Infrastruktur wie ein Hochwasserwarnsystem nicht die nötige Zuverlässigkeit. Aus diesem Grund haben wir für unser System einen Server beim Anbieter Hetzner gemietet, auf dem unsere Anwendung in einem Docker-Container läuft.

Auf dem Raspberry Pi und anfangs auf dem Server haben wir ein Bash-Skript verwendet, das über die Storage API minütlich die letzte Änderung von TTN herunterlädt und in eine große JSON-Datei schreibt. Im letzten Jahr war dies noch kein Problem, da wir nur eine Messstation hatten, die zusätzlich sehr nah an einem Gateway war und es dort selten zu fehlerhaften Datenübertragungen kam.

Mittlerweile sind jedoch fünf Sensoren im Einsatz, wobei es vorkommen kann, dass einzelne Nachrichten nicht ankommen. Dadurch erweist sich unser früherer Ansatz, die Daten minütlich abzufragen, als nicht sinnvoll.

Um nun diese provisorische Lösung zu ersetzen, haben wir uns die anderen verfügbaren Integrations angeschaut und uns dann für MQTT, ein einfaches Übertragungsprotokoll basierend auf Websockets, entschieden. Dieses bietet die Möglichkeit, einzelne Sensoren zu abonnieren und so sofort die neuen Daten der abonnierten Sensoren geschickt zu bekommen, sobald sie bei TTN eintreffen. Dies ist eine viel elegantere, skalierbarere und stabilere Lösung, war dafür jedoch mit einem höheren Programmieraufwand verbunden. Um nun auf unserem Sever die Daten über MQTT herunterzuladen, haben wir uns nach dem Ausprobieren einiger vorhandener Tools dazu entschieden, ein eigenes Python-Programm zu schreiben, was genau dies tut.

Nun hatten wir einen Weg, die Daten auf den Server zu bekommen, allerdings wurden diese immer noch in einer einzelnen sehr großen JSON-Datei gespeichert. Dies war für Demo-Zwecke bei ein paar Tagen Daten und einem einzelnen Sensor noch akzeptabel, es wurde aber schnell klar, dass diese Lösung nur temporär sein kann. Wenn wir fünf Sensoren haben, die alle minütlich Daten schicken, haben wir in einem Jahr 2.628.000 "Payloads" (alle Daten, die der LoRaWAN-Sensor schickt), die ja nicht nur den Distanz-Wert selbst beinhalten, sondern zusätzlich viele andere Metadaten wie z. B. die Zeit oder die Batteriespannung und dadurch ca. 1,4 kB groß sind.

Eine solche Menge an Daten lässt sich in Form einer JSON-Datei nur sehr schwer auswerten, da stets die gesamte Datei gelesen werden muss und die Distanz-Werte aus den Payloads extrahiert werden müssen. Dies benötigt zu viel Rechenzeit, Rechenleistung und ist technisch sehr aufwändig sowie zeitintensiv.

Um unsere Daten effizient auswerten zu können, benötigen wir also eine Datenbank. Zu dieser Zeit war in der App-Development-Community gerade die Firebase/Firestore-Datenbank sehr beliebt. Wir haben zuerst diese Art von Datenbank verwendet, es wurde allerdings schnell klar, dass die unstrukturierte "document-based" Art der Firestore Datenbank nicht für unseren Anwendungsfall geeignet ist. Wir haben uns deshalb für eine herkömmliche SQL-Datenbank entschieden.

Jetzt hatten wir nur noch ein letztes Problem: Das Python-Programm muss dauerhaft auf dem Server laufen, es wird jedoch über eine SSH-Verbindung aus der Ferne gestartet. Wird die SSH-Verbindung geschlossen, wird das Command Line Interface und alle darin gestarteten Prozesse ebenfalls geschlossen. Um zu verhindern, dass unser Python-Programm dadurch geschlossen wird, haben wir verschiedene Ansätze ausprobiert. Zuerst haben wir den "nohup"-Befehl verwendet. Dies hat jedoch nicht zuverlässig funktioniert und war sehr kompliziert zu verwalten. Danach haben wir versucht, das Python-Programm zu einem sogenannten Daemon zu machen, was jedoch ebenfalls unnötig komplex war. Schlussendlich fanden wir die perfekte Lösung mit Tmux, welcher im Kern ein Terminal-Multiplexer ist. Das bedeutet, er ermöglicht mehrere sogenannte "Panels", also virtuelle "Terminal-Fenster". Tmux ist für unseren Fall nützlich, da er zusätzlich die Fähigkeit hat, Sessions im Hintergrund weiterlaufen zu lassen, wenn sie verlassen werden. Später können wir mit dem einfachen Tmux-Attach-Befehl wieder darauf zugreifen, um zu überprüfen, ob zum Beispiel Fehler aufgetreten sind.

Unser Setup sieht also jetzt wie folgt aus: Es gibt einen Linux-Container auf dem Hetzner-Server, auf den wir mithilfe von SSH zugreifen. Dort starten wir Tmux, in dem

wir das Python-Programm ausführen, welches die Daten über MQTT herunterlädt und dann mithilfe der Python SQLite DB-API 2.0 in die SQLite Datenbank schreibt. Der gesamte Workflow ist in Abbildung 13 dargestellt.

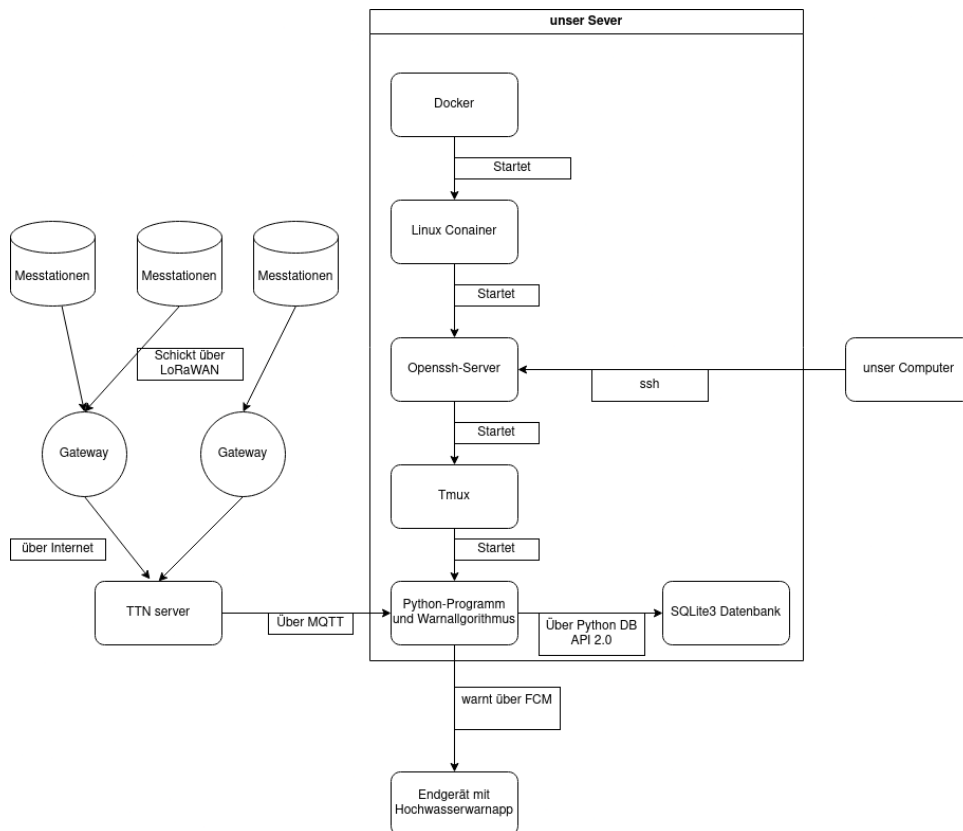


Abbildung 13: Datenübertragung vom Sensor zum Server und Abruf durch die Warnapp

Warnalgorithmus

Wir haben nun dank der Datenbank die Möglichkeit, die Daten zu filtern, mit Daten von anderen Sensoren zu kombinieren und in verschiedenen passenden Formaten auszugeben. Der Warnalgorithmus ist noch nicht final, aber bis jetzt haben wir folgende Parameter als Indikatoren für eine Hochwassersituation gewählt:

1. Überschreiten eines absoluten, festgesetzten Wertes für den Wasserpegel
2. Schneller relativer Anstieg des Wasserpegels

Aktuell experimentieren wir noch mit verschiedenen Parametergrößen für den Algorithmus.

Die grundlegende Funktionsweise des Algorithmus ist in Abbildung 14 dargestellt.

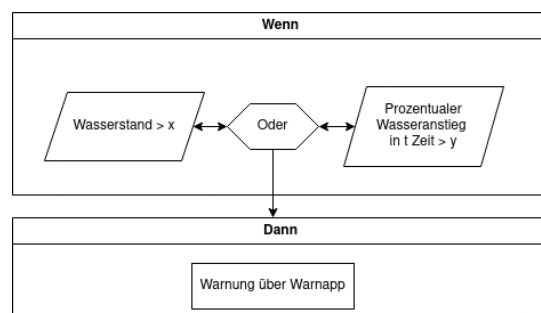


Abbildung 14: Funktionsweise des Algorithmus

Alarmfunktion

Sobald unser Algorithmus basierend auf den Messwerten eine Hochwassergefahr erkennt, muss nun eine Warnmeldung an die Anwohner verschickt werden. Dafür haben wir verschiedene Optionen getestet:

E-Mail

Zuerst war unsere Idee, einen eigenen SMTP-Mail-Server aufzusetzen. Jedoch gab es dabei bei den meisten großen E-Mail-Anbietern Probleme, da die Nachrichten durch den Spam-Filter abgefangen wurden. Um dieses Problem zu lösen, könnten wir „Resend“ verwenden, ein Startup, dessen Werbe Slogan lautet: “E-Mail for developers, reach people not spam folders”. Allerdings erschien uns eine Warnung per E-Mail generell als ungeeignet, da man die meisten Menschen damit nicht in Echtzeit erreichen kann, dies jedoch im Katastrophenfall sehr wichtig ist.

Existierende Warnapps

Wir haben vor kurzem versucht, Kontakt mit den Entwicklern von Katwarn sowie der Nina-Warnapp aufzunehmen, sind aber nicht sehr zuversichtlich auf eine positive Antwort, da wir momentan doch noch sehr klein und lokal sind.

Eine eigene Warnapp

Die Entwicklung einer eigenen Warnapp ist aus unserer Sicht die beste Option, da die meisten Menschen ihr Smartphone immer dabei haben oder zumindest immer in der Nähe sind und somit mit einer Benachrichtigung und/oder einem Klingeln schnell erreicht werden können.

Im letzten Jahr haben wir eine Live-Demo mit einem Bash-Skript gezeigt, das sich über die TTN API die letzten Daten heruntergeladen hat und verglichen hat, ob die Distanzwerte einen bestimmten Grenzwert unterschreiten. Warum diese Möglichkeit, an die Daten zu kommen, suboptimal ist, haben wir oben bereits ausführlich erklärt und unser Algorithmus ist seit letztem Jahr ebenso ausgereifter geworden.

Ein Bash-Skript lässt sich zwar mit Programmen wie WSL auf Windows und Termux auf Android auch auf anderen Betriebssystemen als Linux ausführen, ist jedoch nichts, was der Durchschnitts-Benutzer tun möchte oder kann.

Aufgrund dessen haben wir uns dazu entschieden, die Programmiersprache Dart und das darauf aufbauende Flutter-Framework zu verwenden. Flutter und Dart haben gegenüber plattformspezifischen Alternativen wie Java oder Swift den Vorteil, dass sie auf verschiedenen Betriebssystemen ausgeführt werden können. Das bedeutet, dass man eine Code-Base hat, man also die App nur einmal programmieren muss, und sie dann zu einer Website, zu einem Windows-, Linux- oder macOS-Programm und – am wichtigsten – zu einer Android- oder iOS-App kompilieren kann.

Um nun die Benachrichtigungen zu versenden, haben wir zuerst das Flutter-Package Notify verwendet. Es hat funktioniert, hat aber den sehr großen Nachteil, dass man nur Benachrichtigungen bekommt, wenn die App im Hintergrund läuft. Dies bedeutet einerseits, dass die Benutzer die App dazu autorisieren müssen, im Hintergrund ausgeführt werden zu können, was bei vielen Nutzern zu Problemen führen könnte. Außerdem bedeutet dies, dass die App immer im Hintergrund Strom verbraucht, was wir unseren Benutzern nicht zumuten wollen.

Nach einiger Recherche sind wir auf Firebase Cloud Messaging (kurz FCM) gestoßen, was unseren Kriterien entspricht. FCM ist sehr groß und wird von vielen großen und

einflussreichen Firmen verwendet, was bedeutet, dass in Android und iOS die Fähigkeit eingebaut ist, direkt Nachrichten von FCM zu bekommen, ohne dass die App laufen muss (in Android heißt der Service dafür „Google Push Notifications“).

Standortauswahl

Nachdem wir nun ein funktionierendes Warnsystem haben, mussten wir noch die passenden Standorte für unsere Messstationen identifizieren. Ochsenhausen verfügt über insgesamt vier Zuläufe, die für die folgenden Tests von Relevanz sind (Abb. 15). Diese Zuläufe münden alle am Marktplatz der Stadt in die Rottum, durch die es im Jahr 2016 zu einer erheblichen Überschwemmung kam (Abb. 2 weiter oben). Aufgrund intensiver Niederschläge stieg der Wasserpegel rapide an, wodurch die Rottum über die Ufer trat und die Innenstadt von Ochsenhausen überflutet wurde.

Für die genau Identifizierung der Standorte müssen allerdings noch weitere Faktoren beachtet werden. Wie bereits oben erwähnt, machen wir uns Brücken zunutze, indem wir alle bisher installierten Messstationen darunter befestigten. Unser Plan ist es, dies auch in der Zukunft weiterzuführen. Außerdem arbeiten wir an einer Alternative dafür, da es natürlich nicht an allen relevanten Standorten Brücken gibt.

Ein zusätzlicher wichtiger Aspekt ist der LoRaWAN-Empfang. Natürlich muss dieser vor jeder Installation geprüft werden, denn ohne Empfang können keine Daten gesendet werden. Bisher hatten wir damit aber noch keine Probleme. Prinzipiell ist die Reichweite von LoRaWAN sehr hoch (ca. 10km), jedoch können Bäume oder Gebäude die Signalstärke beeinträchtigen.

Die Analyse der Starkregengefahrenkarten verdeutlicht, dass die besonders gefährdeten Gebiete vor allem der Marktplatz und der Bereich um die beiden Schulen sind (Abb. 16). Aufgrund dieser Erkenntnisse haben wir beschlossen, die Messstationen in diesen Zuläufen zu positionieren. Aktuell haben wir sechs Messstationen in und um Ochsenhausen platziert.

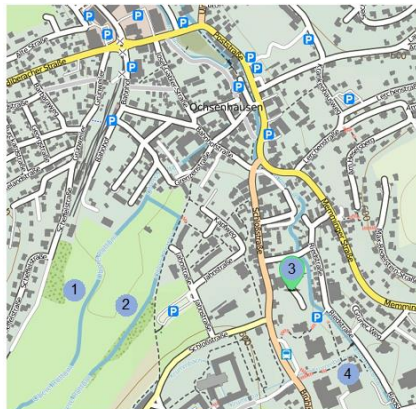


Abbildung 15: Geplante Standorte vom letzten Jahr (1,2 = Obere Rottum, 3 = Untere Rottum, 4 = Krumbach)

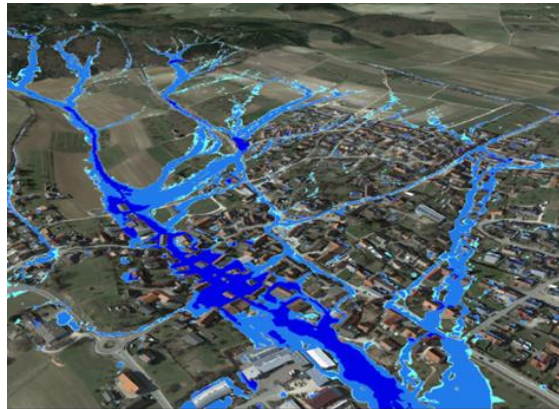


Abbildung 16: Starkregengefahrenkarte der Stadt Ochsenhausen. Zu sehen sind die von Starkregen am meisten betroffenen Bereiche, dazu gehören der Marktplatz sowie die Umgebung der beiden Schulen.

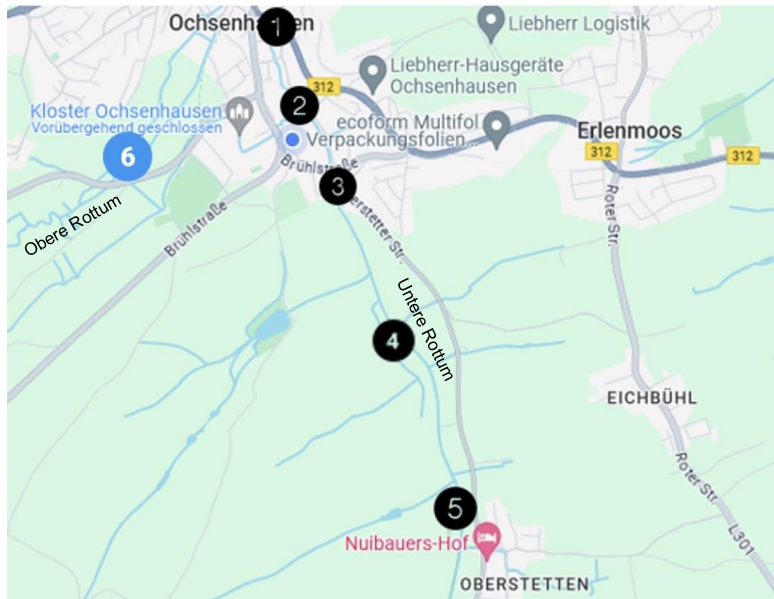


Abbildung 17: Momentane Standortauswahl (1 = Untere Rottum (Adler-Brücke), 2 = Krumbach, 3 = Untere Rottum (Brühlstraße), 4 = Untere Rottum (Oberstetter Straße), 5 = Untere Rottum (Steinhauser Straße), 6 = Obere Rottum (noch nicht installiert))

Im Vergleich zum letzten Jahr haben wir unsere Standortauswahl nochmals überarbeitet, denn nachdem wir vor Ort an den potenziellen Standorten waren, mussten wir feststellen, dass diese nicht alle für unsere Messungen geeignet waren.

Zunächst wollten wir in der Oberen Rottum zwei Messstationen platzieren. Jedoch haben wir uns später dazu entschieden, dort nur eine aufzustellen, da der Zulauf 2 (Abb. 15) aufgrund seiner kleinen Größe nicht für unsere Messungen relevant ist.

Nach der Installation der Messstationen 1, 2 und 6 (Abb. 17) haben wir unsere Ideen aus dem letzten Jahr verwirklicht und zusätzlich Sensoren außerhalb von Ochsenhausen aufgestellt.

Um den Wasserpegel in der Unteren Rottum besser untersuchen zu können, haben wir noch drei weitere Messstationen aufgestellt (Abb. 17). Dies ist von Relevanz, da einige Bäche im Verlauf nach Ochsenhausen in der Unteren Rottum münden und sich so der Wasserstand leicht verändern kann.

Allerdings standen wir nach der Befestigung der Messstation am Standort 4 vor einem Problem: Die Brücke, an der wir sie befestigt hatten, war zu niedrig, sodass der Sensor bei einem Anstieg des Wasserstandes überschwemmt wurde. Unsere Befürchtung, der Sensor sei kaputt, bestätigte sich ich glücklicherweise nicht, da er ohne Probleme weiterhin Daten schickt. Aber trotzdem wissen wir seit diesem Vorfall, dass wir beim Installieren der Messstationen auch auf die Höhe der Brücke achten müssen.

Betrachtung der Messdaten

Die bisherigen Messwerte der bereits installierten Sensoren sind in Abbildung 18 dargestellt. Da wir Sensor 2 (am Krumbach) bereits zu einem früheren Zeitpunkt als Testsensor installiert hatten, war dort zum Beginn der Datenaufzeichnung durch die anderen Sensoren die Batterie leer, sodass er in diesem Zeitraum leider keine Messwerte geliefert hat.

Betrachtet man jedoch die Werte der restlichen Sensoren, die entlang der unteren Rottum aufgestellt sind, wird über alle diese Messstellen hinweg ein gleichmäßiger Verlauf des Wasserstands sichtbar, beispielsweise die Schwankung des Wasserstandes am 13. Januar. Die Diagramme zeigen den Abstand zur Wasseroberfläche, nicht den absoluten Pegelstand. Dies bestätigt die Richtigkeit unserer Messwerte: Sensor 3 hängt an einer höher gelegenen Brücke, während die anderen Sensoren sich näher an der Wasseroberfläche befinden.

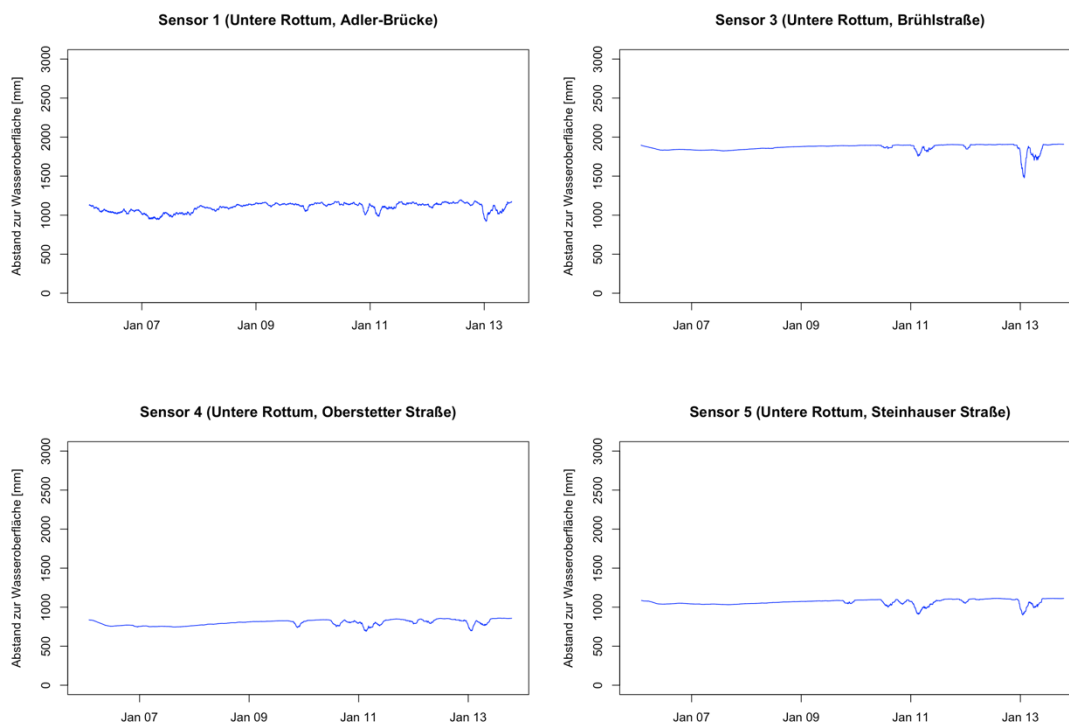


Abbildung 18: Bisherige Messwerte der aufgestellten Sensoren an der unteren Rottum. Alle dargestellten Auswertungen wurden durch die Bildung eines gleitenden Durchschnitts über je 100 im 1-Minuten-Intervall aufgezeichnete Messwerte erstellt.

Zusammenfassung & Ausblick

Im Rahmen unseres Projekts haben wir Zielsetzung, Planung, Bau und Installation der Messstationen bereits abgeschlossen. Zusätzlich existiert eine funktionierende Warnapp, die im Extremfall Warnungen verschickt. Nach der Überarbeitung der geplanten Standorte konnten wir mit der Installation unserer Messstationen beginnen, woraufhin wir fünf von ihnen aufstellten. Nachdem nun auch die Warnfunktion und die Datenauswertung problemlos funktionieren, sehen wir außerdem Potenzial zur Erweiterung unseres Messradius über das Stadtgebiet von Ochsenhausen hinaus.

Für die Zukunft unseres Projekts erhoffen wir uns außerdem, einzelnen Gemeinden sogenannten „Kits“ zum Aufbau von Messstationen zur Verfügung zu stellen. Diese sollen ohne viel Aufwand angebracht werden können. Dabei würden wir die Software hosten und alles vorbereiten, damit die Gemeinden die fertigen Messstationen nur noch aufstellen müssen, in einer von uns bereitgestellten Software angeben, wo welche Messstation steht und ihre Bewohner auf die Warnapp verweisen. Alles andere würde dann automatisiert ausgeführt werden.

Danksagung

Benno Hölz

Ganz besonders möchten wir uns bei unserem Betreuer Benno Hölz für die tolle Unterstützung über diese lange Zeit bedanken. Er hat uns beim ersten LoRaWAN-Workshop, bei der Bereitstellung des Servers, bei der DIY-Ultraschall Messstation, bei der Datenauswertung und allem zwischendrin mit seinem immensen Wissen unterstützt. Seine Hilfe bei Fragen, auch während der Ferien oder an Wochenenden, hat uns zu dem Punkt gebracht, wo wir jetzt sind.

Tobias Beck

Tobias Beck war ausschlaggebend dafür, dass wir vor ca. drei Jahren beim SFZ Ochsenhausen angefangen haben. Seine gute Laune und seine Art zu motivieren sind auch nach z. B. mehreren Stunden Python-Troubleshooting immer ansteckend. Zusätzlich war er die Person, die dieses Projekt ins Rollen gebracht hat. Außerdem haben wir schon öfters von seinem organisatorischen Talent profitiert, wie z. B. als er uns für den Landeswettbewerb 2023 vom Dublin Austausch einfliegen ließ.

Literatur- und Bildquellenverzeichnis

Literaturquellen:

- [1] <https://www.umweltbundesamt.de/themen/ipcc-bericht-klimawandel-verlaeuft-schneller>, 30.12.2023
- [2] https://www.de-ipcc.de/media/content/Hauptaussagen_AR6-WGI.pdf, 30.12.2023
- [3] <https://www.tagesschau.de/inland/gesellschaft/hochwasser-donnerstagsmorgen-100.html>, 30.12.2023
- [4] <https://www.schwaebische.de/regional/biberach/ochsenhausen/schaeden-in-millionenhoehe-505096?lid=true>, 30.12.2023
- [5] <https://www.ingenieur.de/technik/fachbereiche/umwelt/fruehwarnsysteme-in-echtzeit-haetten-tragoedien-vermeiden-koennen/>, 30.12.2023
- [6] https://www.gesetze-im-internet.de/whg_2009/_78a.html, 30.12.2023
- [7] <https://de.wikipedia.org/wiki/Hochwasser>, 30.12.2023
- [8] <https://www.deos-ag.com/de/blog/lorawan-einfach-erklaert/>, 30.12.2023
- [9] <https://www.placetel.de/ratgeber/gateway#>, 30.12.2023
- [10] <https://ecbm.me/2020/03/25/the-things-network-ein-lorawan-netzwerk-server-fuer-alle/>, 30.12.2023
- [11] <https://www.datacenter-insider.de/was-ist-ein-application-server-a-890641/>, 30.12.2023
- [12] <https://de.elv.com/elv-distance1-applikationsmodul-abstandsueberwachung-elv-am-dis1-158060>, 30.12.2023
- [13] Datenblatt des ELV-Sensors „ELV-AM-DIS1“ (<https://de.elv.com/elv-distance1-applikationsmodul-abstandsueberwachung-elv-am-dis1-158060?fs=3793357845>, 14.01.2024)
- [14] Datenblatt des ELV-Sensors „ELV-AM-DIS1“ (<https://de.elv.com/elv-distance1-applikationsmodul-abstandsueberwachung-elv-am-dis1-158060?fs=3793357845>, 14.01.2024)

Bildquellen:

Abbildung 1: <https://www.dw.com/de/t%C3%B6dliches-klima-weltweit-durch-hitze-sturm-und-flut-klima-risiko-index/a-51506072>, 10.01.2024

Abbildung 2: Privataarchiv

Abbildung 3: Erstellt mit Icons aus Icons8 Database, <https://icons8.de/icons/set/database>, 10.01.2024

Abbildung 4: Privataarchiv

Abbildung 5: Erstellt mit BioRender.com, <https://app.biorender.com/>, 30.12.2023

Abbildung 6: Privataarchiv

Abbildung 7: https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/Absorption_spectrum_of_liquid_water.png/1024px-Absorption_spectrum_of_liquid_water.png, 30.12.2023

Abbildung 8: Erstellt mit BioRender.com, <https://app.biorender.com/>, 30.12.2023

Abbildung 9: Privataarchiv

Abbildung 10: Privataarchiv

Abbildung 11: Privataarchiv

Abbildung 12: Privataarchiv

Abbildung 13: Mit Draw.io von uns erstelltes Diagramm

Abbildung 14: Mit Draw.io von uns erstelltes Diagramm

Abbildung 15: Google, Ochsenhausen. Aufrufbar unter: <https://www.google.de/maps/>, 10.01.2024

Abbildung 16: <https://images.app.goo.gl/FWbKx77ekuED4LvN6>, 10.01.2024

Abbildung 17: Google, Ochsenhausen. Aufrufbar unter: <https://www.google.de/maps/>, 12.01.2024

Abbildung 18: Mit RStudio basierend auf Messwerten von uns erstellte Diagramme