

# Proteus - A Platform for Monitoring and Predicting Water Quality and Availability

Mika Engels, Ludwig Müller, Patrick Preuß, and Paul Schütz

February 2020

## Abstract

We propose a hardware and software artifact which aims at helping to tackle Chile's ongoing agricultural water crisis. The artifacts combined allow for monitoring and predicting water quality and quantity.

**Keywords**— water, monitoring, prediction, Chile, Arduino

**Affiliations**— University of Cologne

**Contact email**— mengels3@mail.uni-koeln.de, lmuell88@mail.uni-koeln.de, ppreuss1@mail.uni-koeln.de, pschuet5@mail.uni-koeln.de

## 1. Hardware in Context

### 1.1 Motivation

The access to drinking water is essential for human life as well as for life on earth in general. This and the mega drought Chile is facing in the near future requires the establishment of sustainable water usage in the entire country. Water resources, its distribution, and its usage in Chile is neither owned nor regulated nor controlled by a governmental institution. Rather there are several individual owners or communities in charge. Therefore, the installation of an appropriate monitoring system has the ability to improve the usage of water resources and gives all stakeholders of the drinking water industry an overall view over the resources.

Any kind of such a system should be installed on a mandatory basis. The system has to be offered at a reasonable price especially for smaller, rural communities. The system has to measure some basic sensor data about wells. At least it measured the water level inside a well as well as its temperature and the pH value as a measure for quality. If required, the system could be extended with even more sensors, e.g. turbidity, but water level, temperature and pH value are always required. For transmission of sensor data, free radio networks like LoRaWAN is prefered. Such a radio network has not only the big advantage of being accessible freely but it also is easy to install and has a great coverage going along with low energy consumption. A single gateway can cover an area up to 10km<sup>1</sup>. These gateways are configured to transmit the data received by the system installed at a well to a cloud instance where the data will be stored in an appropriate database. A platform will make this data then accessible for the well-owners as well as for the communities and population.

---

<sup>1</sup>[https://www.tuv.com/media/corporate/products\\_1/electronic\\_components\\_and\\_lasers/TUEV\\_Rheinland\\_Overview\\_LoRa\\_and\\_LoRaWANtmp.pdf](https://www.tuv.com/media/corporate/products_1/electronic_components_and_lasers/TUEV_Rheinland_Overview_LoRa_and_LoRaWANtmp.pdf)

This gives the owners the possibility to have an overview over the changes in both quantity and quality of water resources. As already mentioned, a basic version of the system has to provide information about water level in a well and also the waters' temperature and pH value. It could also be useful to track incoming and outgoing water flows of the wells. Meanwhile, the population's advantage is basically that they can inform themselves about the waters availability and quality they are using every day. In addition, these information help to protect the population from diseases caused by contaminated water.

## 1.2 Related Work

Environmental monitoring solutions have been designed and implemented before. Geetha and Gouthami categorize existing solutions along the following dimensions: whether the system is used to monitor running or stored water, whether it is deployed in- or outside (e.g. in a river or ocean), whether it monitors fresh water meant for drinking, and whether it allows to measure the quality of other environmental elements (e.g. air) as well (Geetha & Gouthami, 2016). Monitoring systems whose sensors are located outside of buildings are most relevant to our design solution, as our hardware artifact should monitor the water in wells outside of buildings. For systems like this it has been shown that using multiple distributed wireless sensor networks based on low-cost internet of things (IoT) sensors yield the desired results regarding wide-area communication (Ingelrest et al., 2010; Geetha & Gouthami, 2016; Peng, Hongbo, Zhiye, & Zheming, 2009; Faustine et al., 2014; Li, Wang, & Liu, 2013).

The usage of sensors for measuring water-related quality data has been described for different use cases. The most commonly used sensors are sensors which measure the water's pH-value, its temperature, and its turbidity (Cloete, Malekian, & Nair, 2016; Peng et al., 2009; Vijayakumar & Ramya, 2015; Wiranto, Maulana, Hermida, Syamsu, & Mahmudin, 2015). Measuring water levels, which is highly relevant to our design solution (as it aims towards helping to tackle drought-related water shortages), has been implemented for dams (Siddula, Babu, & Jain, 2018; Kavitha, Kavitha, Jayalakshmi, & Senthil, 2018), rivers (Amirruddin, Mukhtar, Halim, & Noorpi, 2012), and on agricultural land (Deekshath, Dharanya, Kabadia, Dinakaran, & Shanthini, 2018) before.

In the past, platforms that collect environmental data in large quantities have been created by governmental agencies and non-governmental initiatives alike (McKinley et al., 2017). Most European and American countries enforce the collection of environmental data through their environmental protection agencies (e.g. Sweden<sup>2</sup> or the United States<sup>3</sup>). As open as this data might be, it is not suitable for ad hoc visualization or real-time predictions, as most datasets published are snapshots at a single point in time. For communities that thrive to closely monitor their water resources, real-time data is needed. Non-governmental initiatives, especially so-called *citizen science* projects, are an emerging source for different (including environmental) sorts of data (Kosmala, Wiggins, Swanson, & Simmons, 2016).

The sensor-based systems described above usually include some means of alerting authorized personnel via cellular short message service or incorporate an online dashboard or some similar kind of visualization for the gathered data (Shah & Mishra, 2016). Frameworks and algorithms for predicting data from wireless sensor networks have been developed and implemented as well, utilizing

---

<sup>2</sup><http://www.swedishepa.se/State-of-the-environment/Data-databases-and-applications/Environmental-monitoring-data/>

<sup>3</sup><https://edg.epa.gov/data/>

a variety of different statistical prediction methods (Mou, Liansheng, & Naixue, 2016; McLean, Cullen, Davis, Dempsey, & Adams, 2005; Lynch, Schwartz, Yang, & Freeman, 2008).

## 2. Solution Description

The Proteus platform consists of three individual components. Namely those are the Proteusmeter, the Grafana dashboard, and the admin interface such as illustrated in Figure 1. The Proteusmeter is an Arduino device with sensors which collects data about the well such as water level and water temperature and then forwards this data to our cloud server via LoRa network which then persists the sensor data. The Grafana dashboard provides an aggregated view of the collected sensor data and also provides basic forecasting functionality. This is the main point of contact for the communities and water agencies as it monitors the water levels and the water quality of all tracked wells. Furthermore there is an admin interface which allows the water agencies to add new wells or update existing wells. In the following chapters each component will be described in more detail.

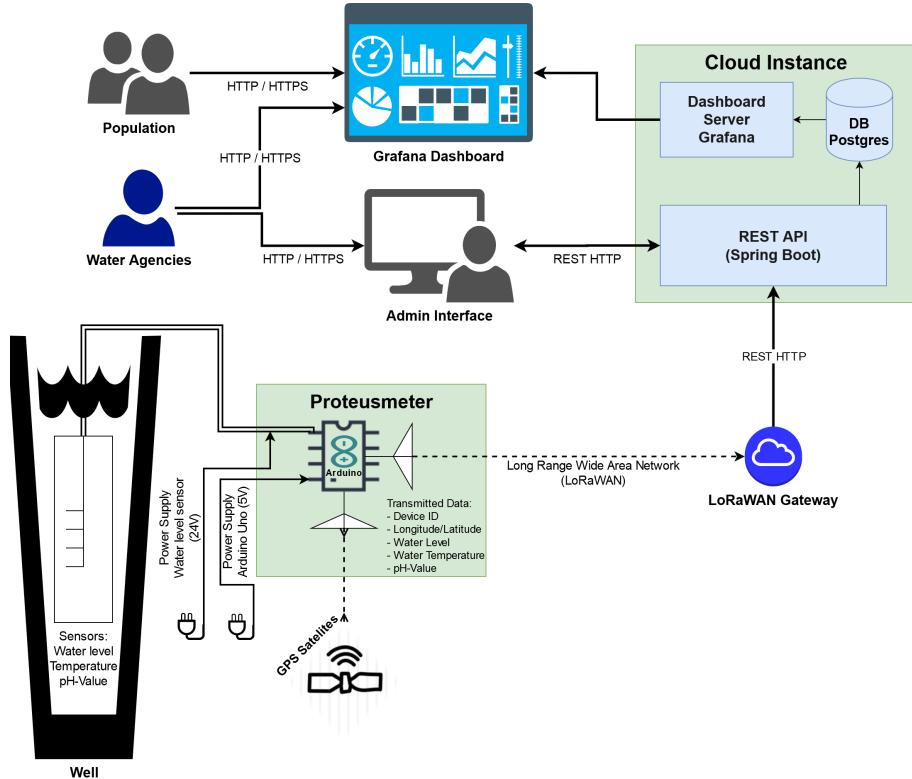


Figure 1: Proteus Platform Architecture Overview

### 2.1 Hardware Artifact

The actual hardware device named Proteusmeter is basically an Arduino UNO REV 3 connected with several sensors and a combined LoRa/GPS shield onttop of the microcontroller. The Arduino and all sensor cablings are placed inside of a closed and waterproofed box with an incoming wiring harness for power supply and an outgoing wiring harness for sensors. The code running on the Arduino is a small programm written in an Arduino-specific language based on C and C++ and

can be accessed via the GitHub repository<sup>4</sup>. It reads the data from the sensors and transmits its via the LoRa-Shield.

### 2.1.1 Hardware

The basic version of a Proteusmeter contains three sensors: A water level sensor, a temperature sensor and a pH value sensor. The temperature and the pH value sensor have upstream controllers to convert the measured data to digital signals and therefore are connected to digital input ports at the Arduino. Their power supply is secured by the 5V and the 3.3V power supply ports of the Arduino UNO. Meanwhile, the water level sensor has its own power supply, its own circuit and is connected to an analog port of the Arduino. A basic Proteusmeter looks as shown in Figure 2 in the real world. A schematic representation is shown in Figure 3. In the following all three sensors, their functionality and their wiring are described seperately in detail.

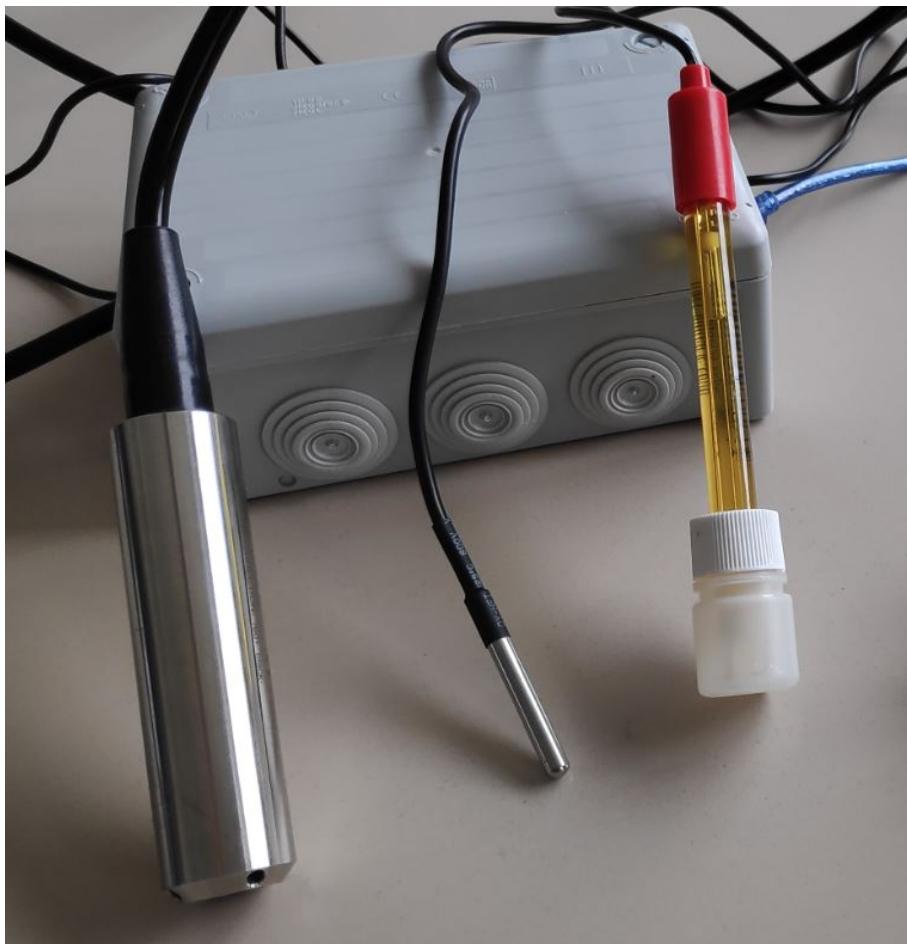


Figure 2: An Proteusmeter and the three basic sensors

---

<sup>4</sup><https://github.com/mengels3/proteus-platform/>

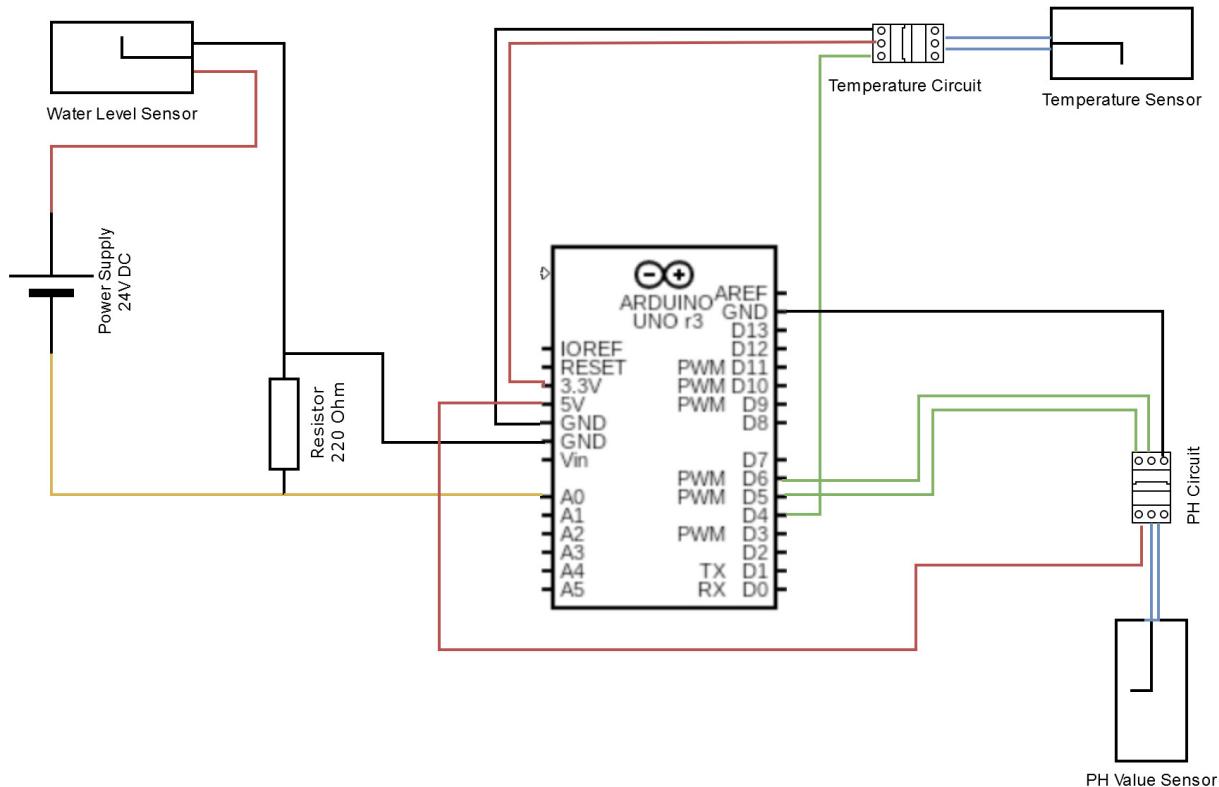


Figure 3: Schematic representation of a basic Proteusmeter

The first and most important sensor is the water level sensor. It is a heavy metallic pole operating with 24 Volts power supply. Therefore the installation of an second power supply in addition to the supply of the Arduino is necessary. A transformator from 24V to 5V for the Arduino would be another option but is not that easy to realize and also more expensive. The Arduino measures the voltage at analog port A0 while a  $220\Omega$  resistor is connected in parallel to the Arduino. The high weight of the sensor presses the sensor to the ground of the well. The sensor measures the pressure difference between this point and the surface. The current surface pressure is measured via a small tube included in the cabling. The higher the pressure difference is, the higher is the voltage present at the analog port of the Arduino. The `analogRead()`-function reads a raw value from 0 to 1023. Out of this raw value the actual voltage, the current, and finally the depth of the well can be calculated. The current is 4mA if the sensor is outside of the well. Every meter of depth increases the current by 3.2mA. The resulting formulas are:

$$Voltage[V] = \frac{raw\_value * 5[V]}{1023} \quad (1)$$

$$Current[mA] = \frac{Voltage[V] * 1000}{220[\Omega]} \quad (2)$$

$$Waterlevel[m] = \frac{Current[mA] - 4[mA]}{3.2[\frac{mA}{m}]} \quad (3)$$



Figure 4: Water level sensor

The temperature sensor looks optically quite similar to the water level sensor even if it is much smaller and lighter. The placement depends on what will be most interesting for water agencies and could be at the ground of the well together with the level sensor or just below the water surface. In contrast to the level sensor this sensor can be supplied with the 3.3V port of the Arduino and does not need an own wiring circuit. It has its own upstream controller to transform the measured value into digital data. Therefore the sensor is connected to an digital port. With the `digitalRead()`-function the Arduino directly reads the current temperature in degree Celsius ( $^{\circ}\text{C}$ )



Figure 5: Water Temperature Sensor

In contrast to both sensors described above the pH value sensor look and work different. Optically, the pH-probe is not a metallic but a plastic pole with a sensing area at the bottom. This sensor works with 5V power supply and has its own upstream controller to convert the measured values in

digital data readable by the Arduino. But this sensor requires a connection to two digital ports of the Arduino. The Arduino reads a pH value as a string from the upstream controller. The sensor has to be calibrated occasionally. This can be done with three different liquids with specific pH values. The calibration source code can be downloaded from the manufacturer's website<sup>5</sup>.



Figure 6: Atlas Scientific pH Probe

### 2.1.2 Software

The source code running on the Arduino microcontroller consists of four parts:

- Reading pH value
- Reading and calculating values from water level and temperature sensor
- Querying GPS coordinates
- Sending all values as LoRa-Message

In the first part only the value of the pH probe is queried due to the sensor's special characteristics. The full value is a string compounded from single characters received loopwise. The last character received is a newline (\r). After this character a full pH value is received and returned by the respective function.

After the pH value string reached this point the code continues with the next step. To determine the water level the formulas mentioned in the water level sensor's section are used: The raw value from analog port A0 is used in the first formula. From this point the level is calculated by using the result of formula (1) in formula (2) and the result of formula (2) in formula (3). For querying the temperature the package of the manufacturer has to be imported. Then the value in °C can be read from a digital port.

The next step is the determination of the geographical position of the Arduino respectively the well. The GPS shield ontop of the Arduino delivers the current longitude and latitude.

The last step is transmitting all values received from the sensors via LoRaWAN. The LoRa-Shield ontop of the Arduino is used for this purpose. A LoRa message (or LoRa-Package) consists out of a bunch of "Key=Value"-Pairs separated by semicolons. In addition to the sensor values a device id

<sup>5</sup>[https://www.atlas-scientific.com/product\\_pages/kits/ph-kit.html](https://www.atlas-scientific.com/product_pages/kits/ph-kit.html)

is transmitted to assign the messages from different Arduinos. For transmitting the Message the package “LoRa” has to be included in the source code. After a settled delay the whole program starts again, queries the sensors again and transmits a new message.

## 2.2 Monitoring Platform

The Proteusmeter monitoring platform is designed as a cloud-ready package of multiple small subsystems. It is made of four distinct parts: a database for storing measurement and master data, a dashboard that provides insight into the data, an API that is used to publish data, and an administrative interface. The monitoring platform can be deployed to a properly configured Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) cloud environment as well as an on-premise solution, as it is distributed in pre-configured Docker<sup>6</sup> container images.

### 2.2.1 Data Publishing Interface

To enable communities to gather data from a large fleet of Proteusmeter devices which may be spread across a long spatial distance, a wireless communication technology must be used. LoRaWAN<sup>7</sup> is used for Proteusmeters, because it is the most adopted one (Adelantado et al., 2017). A special LoRaWAN gateway device is used for (passively) collecting the sensor values from all devices in its range. The gateway has to be connected to the internet or a local area network, allowing it to transform the LoRa packages into “ordinary” transmission control protocol (TCP) packets. For this task, a small piece of software is running on the LoRaWAN gateway device itself, executing hypertext transfer protocol (HTTP) requests via a widely adopted network transfer library (cURL<sup>8</sup>), which allows the use of common security measures like transport layer security (TLS) encryption or certificate-based authentication methods.<sup>9</sup>

In previous work on similar projects, researchers sent data packets like these to IoT data collection platforms like ThingSpeak<sup>10</sup> or The Things Network<sup>11</sup>, from where they could be downloaded for analysis or visualization (Deekshath et al., 2018; Sivaiah, Sowmya, Susmitha, Sai, & Suma, 2018). But the LoRaWAN gateway used for Proteusmeters rather sends all the data it receives to the application programming interface (API) of a custom web service. As a custom solution it (obviously) lacks the standardized APIs from well established platforms like ThingSpeak, but it allows for a maximum of flexibility regarding the data’s structure and the volume of data transmitted.<sup>12</sup> The API can be exposed publicly or could be hidden behind commonly used security means like a virtual private network (VPN). It allows for (almost) arbitrary environmental data to be published (see below), which greatly improves the future possibilities for Proteus as an environmental platform.

---

<sup>6</sup><https://www.docker.com/products/container-runtime>

<sup>7</sup><https://lora-alliance.org/about-lorawan>

<sup>8</sup><https://curl.haxx.se/>

<sup>9</sup>There are many possible ways of authenticating a LoRaWAN gateway device: for instance see RFC5246 for details on client certificate authentication (<https://tools.ietf.org/html/rfc5246#section-7.4.4>) or libssh’s documentation on public-key-based challenge response authentication ([http://api.libssh.org/master/libssh\\_tutor\\_authentication.html](http://api.libssh.org/master/libssh_tutor_authentication.html)).

<sup>10</sup><https://thingspeak.com/>

<sup>11</sup><https://www.thethingsnetwork.org/>

<sup>12</sup>While prototyping our artifacts, we found especially the latter one, volume of data transmitted, to be severely limited with the existing (openly accessible) IoT cloud platforms.

## 2.2.2 Database

Proteus' cloud system by default comes equipped with a single PostgreSQL<sup>13</sup> database that stores all relevant data. The data can be categorized as (i) master data on the deployed wells and their location and sensors, as well as (ii) measurement data. The latter being the whole set of sensor values. These measurement data points are stored as key-value pairs, allowing for flexibility regarding the sensors canonical data type (floating point values vs. integer values vs. GPS coordinates etc.) as well as the actual database used. The default PostgreSQL instance could be swapped out for an non-relational database at any point without losing information.

## 2.2.3 Dashboard

The main purpose for building the well sensor box is to collect data and provide a better overview on the current water situation in Chile. The sensors provide us with the ability to collect water from the wells, but in order to provide a monitoring for water shortages to the Chilean officials, a dashboard has to be developed. We persist the sensor data in a PostgreSQL database and use the open-source dashboarding platform Grafana to visualize the data. This dashboard will inform the official about water shortages or water contamination. Grafana is an open-source dashboarding platform that offers a wide set of graphs and other panels one can place on their dashboard to visualize data. A plugin structure allows it to install even more third-party panels. Grafana is compatible to a large amount of data sources, which made it a perfect fit for the task at hand. To aggregate the data of several wells, the main dashboard page, contains a map showing all the wells connected to the platform and their water level. Additional panels show average information. A list of all the wells, can be used to jump to a more detailed dashboard for a dedicated well. On this detail view, the officials are able to analysis the development of water level, temperature, and pH value of a selected well. From this information you can derive whether the water is drinkable or not. The possibility to see a large time window, can help to better understand water shortages in certain areas of Chile.

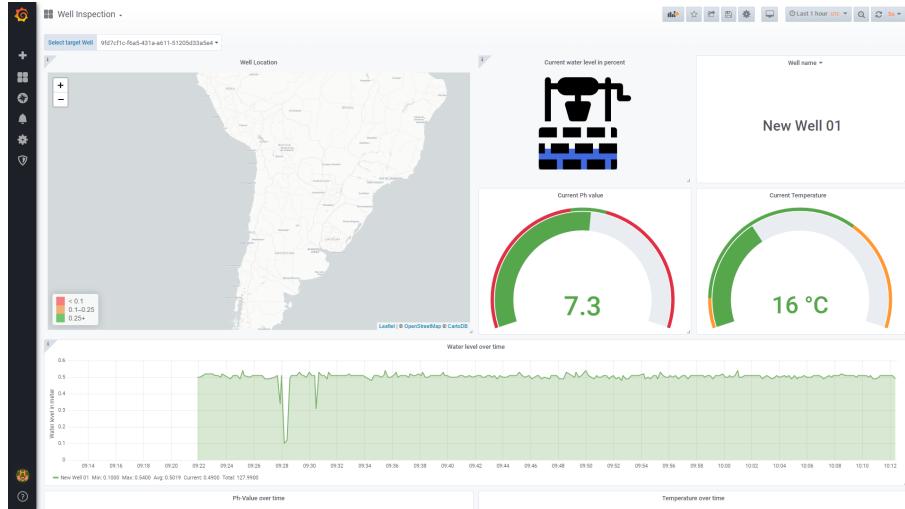


Figure 7: Proteus Well Dashboard

To support the officials even more, a basic time-series prediction is included in the dashboard. On a

<sup>13</sup><https://www.postgresql.org/>

separate dashboard page, one is able to see a prediction of our measured values. These predictions are based on the historic data from our database. A separate webservice endpoint offers this prediction using an AutoRegressive-Moving Average (ARIMA) model(Ho & Xie, 1998). ARIMA is stochastic model that can be used for a static time-series analysis. The ARIMA model looks for patterns inside the time-series data and tries to predict the next development of the curve. Using extensive weather data and by utilizing the grid structure of the wells one could build an even more advanced prediction.

#### 2.2.4 Well Management Admin Interface

The platform should enable the water agencies to add new wells and edit or delete existing wells dynamically. Therefore we provide an administrator interface where administrators can manage all wells. For example it could be the case that an existing well which currently only measures the water level gets an additional sensor for metering also the water temperature which can be considered a measure for water quality. In order for the system to know what type of parameter is actually measured and how it should display it we have to modify the existing well in the admin interface and add the temperature sensor to our well. This changes are directly stored to the database (see section 2.2.2). Figure 8 shows a screenshot of the main interface where all wells are listed. When this figure was made apparently there were only two wells in our system. The first measured pH, temperature, and water level while the second measured only pH and temperature. So this interface provides basic CRUD<sup>14</sup> functionality for the well management.

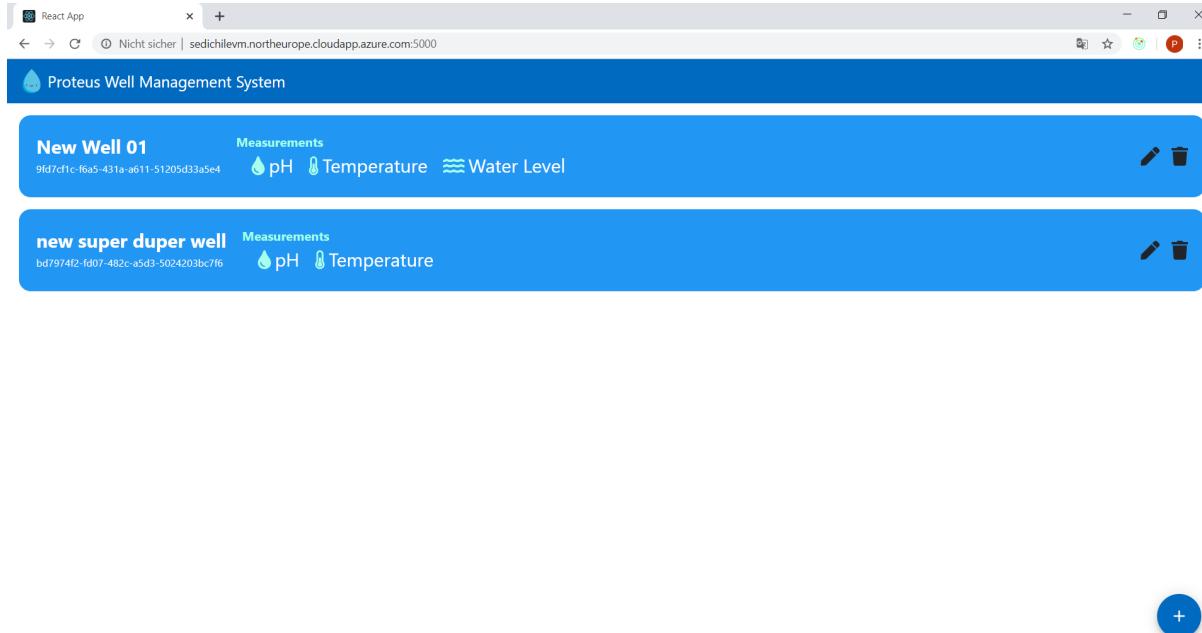


Figure 8: Proteus Well Management Admin Interface

---

<sup>14</sup>Create,Read,Update,Delete

### 3. Bill of Materials

Designator	Component	Number	Cost per unit currency	Total cost	Source of materials
LoRa IoT Development Kit	Dragino LoRa IoT Development Kit v2 868 MHz	1	124,95 €	124,95 €	EXP Tech
Water Level Sensor	TL231 DC24V 4-20mA Water Level Sensor	1	48,49 €	48,49 €	Amazon
Water Level Sensor	Aistuo 24V Power Supply	1	11,58 €	11,58 €	Amazon
pH Probe	Atlas Scientific Lab Grade pH Probe (Gen 2)	1	85,68 €	85,68 €	EXP Tech
pH Probe	EZO-pH Embedded pH Circuit	1	42,84 €	42,84 €	EXP Tech
Water Temperature Sensor	Velleman VMA324	1	13,99 €	13,99 €	Conrad

### 4. Build Instructions

The building of the Proteusmeter is done by the following steps:

- Build the circuit required for the water level sensor and connect it with the Arduino
- Connect the upstream controllers of the temperature and the pH value sensors with digital Arduino ports
- Connect the pins of the GPS shield with digital Arduino ports
- Place the Arduino and the whole cabling inside an appropriate box

The basic instruction for all cablings were taken from the respective manuals. The whole sensor cabling (without the GPS cabling) is shown in Figure 3 and has to be respected at all times.

The most difficult step is the circuit for the water level sensor. It is important to draw the circuit and think about voltages and currents before connecting the sensor with the power supply and especially the Arduino, because a higher voltage than 5V would destroy parts of the Arduino or

more. This together with the information of an increasing current of 3.2mA for every meter of water depth the formulas provided in section 2.1.1 the required amount of resistance can be calculated and hence the whole circuit can be realized.

Since the other two sensors have upstream controllers for converting their signals into digital values the cabling is quite easy. For both sensors there are manuals how to use them correctly. An Arduino wiring manual for the temperature sensor can be found at the Arduino project hub<sup>15</sup>. For the pH probe the manufacturer's website is quite helpful<sup>16</sup>. The same manuals also help to answer the question how to receive the sensor values at the software site. For the usage of the GPS shield it also has to get connected with digital Arduino ports. Again the manufacturer's website helps for cabling and at the software side of things<sup>17</sup>.

The last step is the placement of the Arduino and the whole cabling inside of a closed and waterproofed box. This box needs two openings: At one side for both power supplies and at the other side for outgoing sensor cablings. In the closed state this box looks as shown in Figure 2. Uncovered it looks as in Figure 9. As long as all intrcutions made in this section and by the manufacturer of the sensor are respected and the cabling are isolated appropiate there are no other electric security constrains.

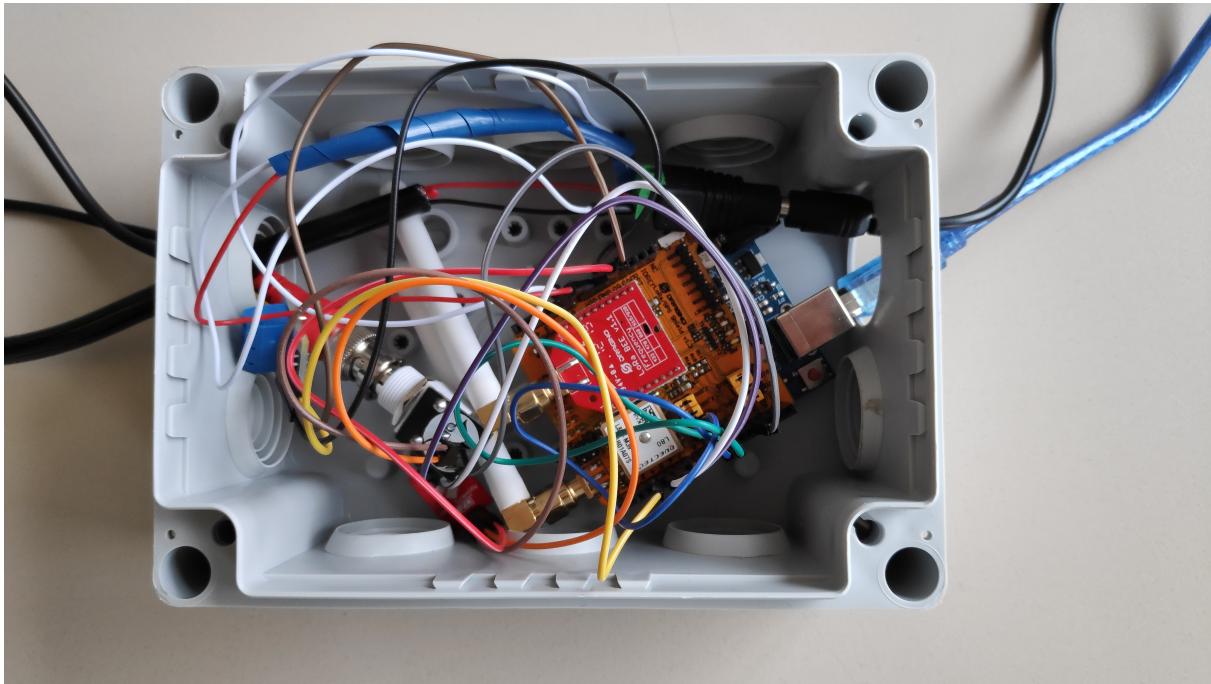


Figure 9: Inside of a Proteusmeter

## 5. Operation Instructions

In order to operate the Proteus platform and a set of Proteusmeters, the following steps must be taken. One has to

<sup>15</sup><https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806>

<sup>16</sup>[https://www.atlas-scientific.com/product\\_pages/kits/ph-kit.html](https://www.atlas-scientific.com/product_pages/kits/ph-kit.html)

<sup>17</sup>[https://wiki.dragino.com/index.php?title=Lora/GPS\\_Shield](https://wiki.dragino.com/index.php?title=Lora/GPS_Shield)

- deploy the pre-configured Docker containers to a cloud system of their choice
- create a new entry for each well in the admin interface, providing the appropriate data on its location, sensors etc.
- set up a LoRaWAN gateway device in a suitable spot where it covers the target area
- install the Proteusmeter boxes in all wells

The Docker images can be deployed to a cloud system or a single data center machine. If needed, the database implementation could be swapped out for a different one. Once the the images are deployed and all security measures are implemented (e.g. a VPN or a reverse proxy checking client certificates), a new well entity has to be created for each well that will be monitored by a Proteusmeter. It's not necessary to know all the wells and their values beforehand, these information can be fed or updated in multiple steps as well.

To be able to gather data from all Proteusmeters, one must find a suitable spot for a LoRaWAN gateway device, which not only allows for a good coverage of the targeted area, but also allows for a stable internet (or LAN) connection, as by default, data packets are only sent once. No retries are performed if data packets are lost. Once the gateway device is set in place, one has to adjust the data-transfer script (i.e. add a proper domain name or internet protocol address and configure any security measures like client certificate) and copy it to the gateway device. If necessary (e.g. due to the spatial distance that should be covered), more than one gateway could be deployed.

Last, the Proteusmeter boxes must be installed in the appropriate wells. The one who installs the boxes must ensure that the cable lengths matches the well's depth. For instance, the water level sensor described above must come with a cable long enough to reach the well's ground. Certain sensors (e.g. the pH value sensor described above) must be calibrated according to the manufacturers' specifications.

## 6. Validation

A first feasibility test was conducted during the initial development of the prototype. For this, DIN1401-compliant pipes were filled with water to simulate rising water levels in a well. The sensor data was then sent over a LoRa compatible radio frequency (868 MHz) to the LoRa gateway and from there it's send to the platform. Tests with the pH value sensor were done using citric acid. Adding the acid caused a significant drop of the reported pH value. The water level and temperature sensor were tested by pouring in hot water. This led to an increase of the measured temperature and the water level. These tests can ensure that all the sensor of the Proteus Box work properly and that the LoRa connection to the cloud works fine.



Figure 10: Well Setup

## 7. Limitations and Future Work

When prototyping the Proteusmeter device our laboratory setup was somewhat limited due to its laboratory nature itself. While aiming towards both a device and a platform that are as flexible as possible regarding the collection of arbitrary data, only so many different sensors could be used. We believe that the three sensors (i.e. for temperature, pH value, and water level) and the GPS coordinates combined are suitable as a proof of concept. Yet, expanding the Proteusmeter to use more (and different types of) sensors would be a reasonable direction for future work.

The Proteusmeter was also limited by its power supply and cable length, with the former being bound to an ordinary power outlet and the latter being rather short (less than 5m). Therefore, another direction could be to upgrade the Proteusmeter with a solar power panel, hydro generator or other kind of renewable energy source; or to test the device in a deep well, which would require longer cables and proper cable management (to avoid any debris, obstacles and the like).

A third possible direction for future work would revolve around the prediction of environmental phenomena, as our prototypical device only uses a single kind of prediction technique. Extending the predictive capabilities by more, different and more sophisticated prediction algorithms, the Proteus software artifact could be enhanced into a full-fledged environmental monitoring and prediction platform.

## 8. Conclusion

In this work we developed two artifacts that aim at helping communities in drought-stricken regions like rural parts of Chile to monitor their precious water resources. The piece of hardware we describe here is assembled from low-cost IoT-based sensors and a widely used microcontroller board. The low price of the parts used allows communities in rural areas to purchase them in large quantities and to deploy a large fleet of independent measurement devices.

The software artifact is an easy-to-use system for visualization and real-time predictions that is solely based on open-source technology. This and the pre-configuration of software container images,

which allows for cost-efficient deployment in modern cloud environment, will help communities to keep the costs down as well. These two artifacts combined could help communities to not only gather knowledge on the status quo of their water resources. But also to better predict water shortages and act upon these information. Those communities willing to take one further step towards a federated environmental data platform could use the artifacts and the flexibility they provide to gain insight into even more environmental data.

## 9. Acknowledgments

The authors would like to thank the staff at the Cologne Institute for Information Systems for providing us with the project opportunity itself, and for their help and feedback during the project. The authors also want to express their very special thanks to Juan Ramón Candia of The Green Factory<sup>18</sup> and Anna Mehrländer and Franz von Weizsäcker of Deutsche Gesellschaft für Internationale Zusammenarbeit’s Blockchain Lab<sup>19</sup> for their inspiring input and helpful feedback throughout the project.

## References

- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the limits of lorawan. *IEEE Communications magazine*, 55(9), 34–40.
- Amirruddin, M., Mukhtar, N. M., Halim, H. A., & Noorpi, N. S. (2012). Microcontroller based water level indicator using gsm modem: design and application. In *1st international conference on future trends in computing and communication technologies* (pp. 79–83).
- Cloete, N. A., Malekian, R., & Nair, L. (2016). Design of smart sensors for real-time water quality monitoring. *IEEE access*, 4, 3975–3990.
- Deekshath, R., Dharanya, P., Kabadia, K., Dinakaran, G., & Shanthini, S. (2018). Iot based environmental monitoring system using arduino uno and thingspeak. *IJSTE-International Journal of Science Technology & Engineering*, 4(9).
- Faustine, A., Mvuma, A. N., Mongi, H. J., Gabriel, M. C., Tenge, A. J., & Kucel, S. B. (2014). Wireless sensor networks for water quality monitoring and control within lake victoria basin.
- Geetha, S., & Gouthami, S. (2016). Internet of things enabled real time water quality monitoring system. *Smart Water*, 2(1), 1.
- Ho, S., & Xie, M. (1998). The use of arima models for reliability forecasting and analysis. *Computers & Industrial Engineering*, 35(1), 213 - 216.
- Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., & Parlange, M. (2010). Sensorscope: Application-specific sensor network for environmental monitoring. *ACM Transactions on Sensor Networks (TOSN)*, 6(2), 1–32.
- Kavitha, R., Kavitha, R., Jayalakshmi, C., & Senthil, K. (2018). Dam water level monitoring and alerting system using iot. *International Journal of Electronics and Communication Engineering*, 5.6, 19–22.
- Kosmala, M., Wiggins, A., Swanson, A., & Simmons, B. (2016). Assessing data quality in citizen science. *Frontiers in Ecology and the Environment*, 14(10), 551–560.

---

<sup>18</sup><https://www.thegreenfactory.cl>

<sup>19</sup><https://www.giz.de/en/worldwide/67045.html>

- Li, Z., Wang, K., & Liu, B. (2013). Sensor-network based intelligent water quality monitoring and control. *International Journal of Advanced Research in Computer Engineering Technology*, 2(4).
- Lynch, K. M., Schwartz, I. B., Yang, P., & Freeman, R. A. (2008). Decentralized environmental modeling by mobile sensor networks. *IEEE transactions on robotics*, 24(3), 710–724.
- McKinley, D. C., Miller-Rushing, A. J., Ballard, H. L., Bonney, R., Brown, H., Cook-Patton, S. C., ... others (2017). Citizen science can improve conservation science, natural resource management, and environmental protection. *Biological Conservation*, 208, 15–28.
- McLean, S., Cullen, J., Davis, R., Dempsey, C., & Adams, R. (2005). The marine environmental prediction system (meps)-a new generation of moored ocean observing systems. In *Proceedings of oceans 2005 mts/ieee* (pp. 1016–1022).
- Mou, W., Liansheng, T., & Naixue, X. (2016). Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Information Sciences*, 329, 800–818.
- Peng, J., Hongbo, X., Zhiye, H., & Zheming, W. (2009). Design of a water environment monitoring system based on wireless sensor networks. *Sensors*, 9(8), 6411–6434.
- Shah, J., & Mishra, B. (2016). Customized iot enabled wireless sensing and monitoring platform for smart buildings. *Procedia Technology*, 23, 256–263.
- Siddula, S., Babu, P., & Jain, P. (2018, 02). Water level monitoring and management of dams using iot. In (p. 1-5). doi: 10.1109/IoT-SIU.2018.8519843
- Sivaiah, N., Sowmya, K. P. S., Susmitha, K., Sai, N. A., & Suma, N. (2018). Internet of things (iot) enabled water monitoring system. *IRE Journal*, 1(8).
- Vijayakumar, N., & Ramya, R. (2015). The real time monitoring of water quality in iot environment. In *2015 international conference on innovations in information, embedded and communication systems (iciiecs)* (pp. 1–5).
- Wiranto, G., Maulana, Y. Y., Hermida, I. D. P., Syamsu, I., & Mahmudin, D. (2015). Integrated online water quality monitoring. In *2015 international conference on smart sensors and application (icssa)* (pp. 111–115).

## A. Group Assignment Overview Sheet