AQUAMETRIC DISTRIBUTED AND LOW COST STREAM MONITORING

Rohan Menon and Ian Vernooy

ABSTRACT

Metrics such as stage height, conductivity, and temperature can provide valuable insight into ecological health and can also serve as predictors for events such as flooding. Traditionally, these metrics are measured either manually by a scientist owr by large and expensive permanently installed stations. Unfortunately, human measurement reduces the possible frequency of data collection and traditional sensor stations require lots of infrastructure. We aimed to address both of these problems by creating a low-cost, distributed IOT sensor network. The stream monitoring system consists of one or more sensing units that communicate over cellular data to a central server. A cellular enabled microcontroller coordinates input from the various sensors. A laser time of flight (ToF) sensor is mounted towards the surface of the water to measure changes in gauge. These components are mounted in a 3D printed, weather-proof case about 30 cm above the highest expected height of the stream. Conductivity and temperature sensors are mounted below the water's surface. The sensing unit is designed to use as little power as possible in order to maximize their lifetime. They enter a sleep state for the time between sensor readings and wake once an hour. Then, they connect to a central server via an LTE data connection. The server stores the sensor data in a database and hosts a website to access and analyze historical data.

BACKGROUND

Streams and rivers are crucial components of many ecosystems and human activities. Today, the USGS maintains the largest collection of stream sensing devices. They have about 10,000 sensing devices across the nation. Unfortunately, this does not cover millions of American homes susceptible to flooding from nearby streams and rivers. Additionally, these sensing stations are designed for accuracy. This means that installing one has a high cost of infrastructure and can cost thousands of dollars.

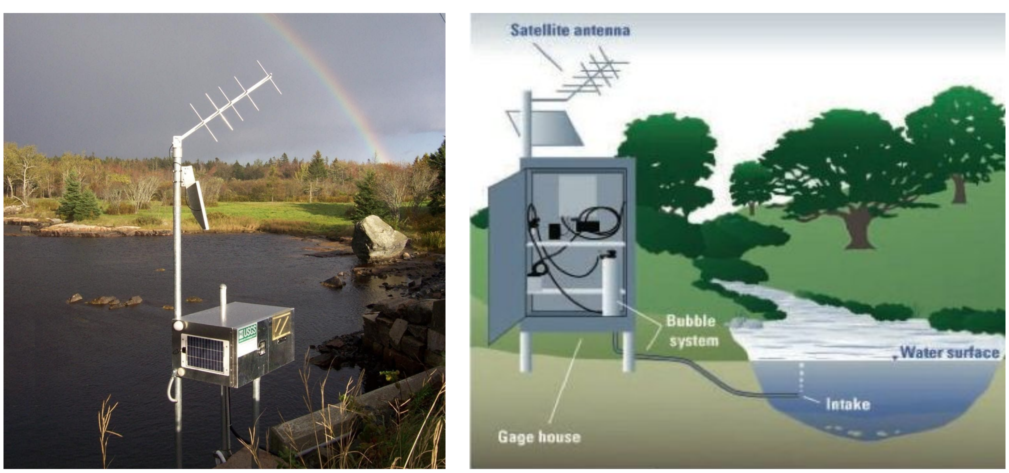


Figure 1: A typical USGS stream gage

Stream data does not have to be limited to gauge and flow rates. Many residential drinking supplies come from water that originates as runoff. Impurities like road salt and fertilizer can sometimes find their way into these drinking supplies. By monitoring temperature and conductivity in these streams, you can help predict ecological events like algal blooms. And, if the network of sensors is large enough, you can pinpoint the origin of the impurities.

By creating a low-cost, distributed IOT sensor network, we hope to address both of these problems. The availability of large volumes of high-quality sensor data has the potential to improve applications such as weather forecasting and ecosystem health monitoring.

DEVELOPMENT

The sensor went through many design iterations before we arrived at the current version.

Version 1

For the first version of our sensor, we focused on developing a device that could take accurate measurements of the water's stage height. We initially chose an ultrasonic range finder for this purpose. It was oriented towards the water's surface. By subtracting the measured value from the height at which the sensor was mounted, we could calculate the height of water flowing through the stream.

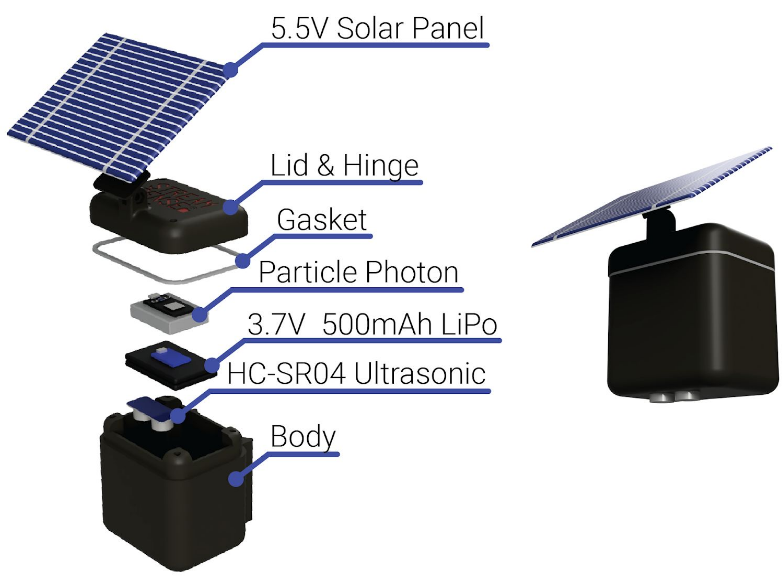


Figure 2: A mockup of design iteration 1

We decided to power this device with a solar panel and lithium polymer battery. This would theoretically allow the device to be deployed indefinitely. For communication, a WiFi-based Particle Photon microcontroller was used. WiFi limits the possible locations of the device's deployment, but we were less focused with this issue with this iteration. We were more focused on developing a device that could stay powered while providing accurate measurements.

Figure 3: Design iteration 1

While this iteration showed promise, it had issues revolving around waterproofing, battery life and connectivity.

While ultrasonic seemed like the clear choice for the sensor, it made the device almost impossible to waterproof. Unfortunately, the ultrasonic sounds from the HC-SR04 can not permeate almost any waterproof materials. This means that the receiving and transmitting elements were directly exposed to weather. They developed rust and failed within a week.

Both the battery and microcontroller operated on 3.3 volts. The ultrasonic sensor required 5 volts to function properly, so a voltage boost circuit had to be added. This added complexity and reduced battery life.

The WiFi based microcontroller that was used prevented the device from being deployed anywhere without an access point and permission to use it. This was the main barrier that we sought to overcome in our next iteration.

Version 2

The second design iteration was driven by some of the issues we discovered in preliminary testing of the first version.

This version began with a choice in communication method. LoRa (Long Range) is a communication protocol for low bandwidth and low power communication. There is a large community of developers using LoRa for sensors similar to ours. Some were achieving distances of upwards of 20 miles from transmitter to base station. We determined that most streams of interest would be within 20 miles of a WiFi and power source where a base station could be installed.

After beginning to test the possible range of the sensor, we quickly started realizing that it was going to pose an issue. Simply getting one mile to the nearest stream was impossible. We knew that tree cover was going to reduce the signal strength, but we were not expecting the loss of signal that we observed.

Our signal strength was also significantly decreased by the fact that rivers tend to be located in valleys, which meant that our sensor would not have a direct line of sight to the base station. To attempt to fix these issues we first switched from standard linear antennas to two directional antennas.

Figure 4: A directional Yagi antenna mounted in the highest place we could get it: the attic.

This improvement in range allowed us to develop a sensor for a neighborhood stream about .2 miles from the base station.

Figure 5: This sensor was deployed in a neighborhood stream and used 815 MHz LoRa for communication.

Although the use of directional antennas did result in an increase in usable range, this increase was not significant enough to make this a practical solution. We briefly considered creating a system of several repeater stations to re-broadcast signals until the data arrived at our base station, but we decided that a simpler solution would be to move to a communication protocol for which a significant infrastructure of “repeater nodes” already exists - cellular.

Version 3

For the third version of the sensor, we again desired to improve on some of the issues that we discovered with our second design iteration.

Firstly, we switched from LoRa to cellular as the method we would use to transmit data from the sensor. Cellular was initially dismissed as a possible communication method due to its relatively high power consumption. We knew that there would be new challenges to using cellular connectivity, but its benefits would outweigh the costs. By using cellular, we were longer limited by our distance to the sensor - as long as we were within the range of a cell tower, we would (theoretically) be able to transmit stream data.

Another change that was made in the third design iteration was to switch to AA batteries instead of LiPo (Lithium Polymer) battery cells for our power source. Although alkaline batteries are less energy dense than lithium based batteries, switching came with a variety of advantages. AA batteries are less dangerous, as they don't have a tendency to ignite when over-discharged. They also have a much longer shelf life, and are less susceptible to extreme temperature and water damage. We found that they performed at temperatures near freezing, which was an important consideration for us.

To reduce power consumption in this version, we added a secondary microcontroller to control power to the main cellular microcontroller. While on the surface, this may seem like simply doubling the required power, it proved highly effective to extend the battery life of the device. The power switching microcontroller is outfitted with a real time clock (RTC). It allows it to enter a very deep sleep state in between sensor readings. Each hour, the RTC wakes up the power switching microcontroller which subsequently uses a MOSFET to provide power to the cellular Particle Boron. This reduced the on-time of the device from 60/60 minutes to about 7/60 minutes.

For the first installation of our final sensor version, we used a 1.5 m aluminum extrusion that was driven about 50 cm into the ground. Finding a location in the riverbed that was soft enough to drive in the extrusion but firm enough to support the sensor was difficult, but we settled on a location under a bridge. Unfortunately, when we came to check on this sensor after a few days, we found the pole had been bent significantly from what we imagine were large ice sheets being carried down by the water. Additionally, a large amount of debris collected on the sensor pole which obstructed our ToF distance sensor.

Figure 6: Sensor version 3 at time of installation and 4 days after

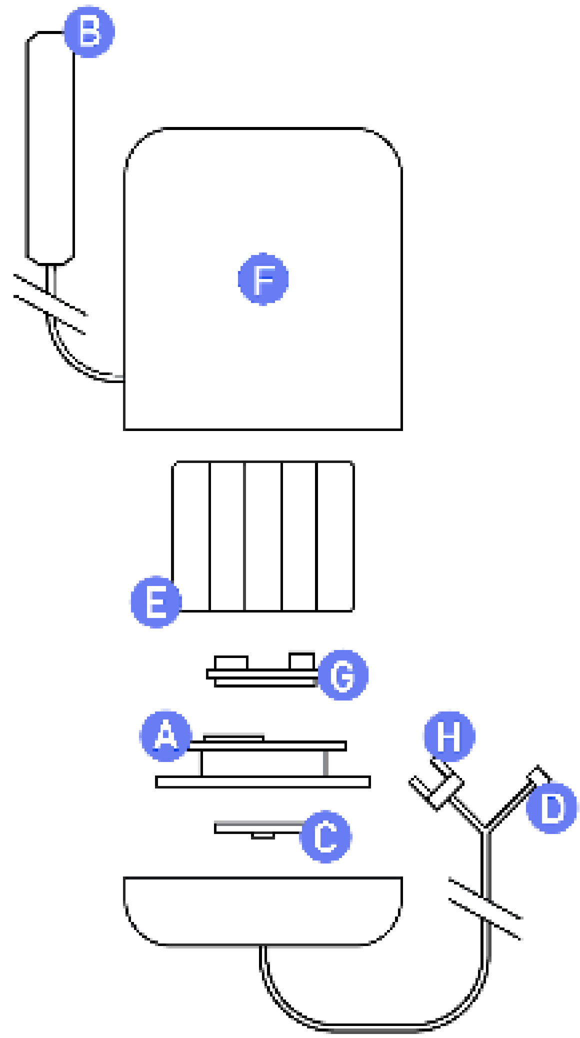
The next deployment of our third iteration drove the development of new mounting systems. We located an alternative spot to mount the sensor, somewhat downstream from our first location, in shallower and faster-moving water. The streambed here was much rockier than our original location, making it nearly impossible to get the metal pole deep enough to provide the necessary stability. To solve this issue, a system was developed in which the aluminum extrusion was fastened to a steel plate through the use of L-brackets. This was placed onto the streambed, and covered with rocks for stability. This new system gave us much greater versatility in the selection of deployment locations. The faster-moving and shallower water made it much more difficult for debris to build up after the sensor, and after a few days there was essentially no debris built up on the sensor at all. Additionally, by moving to a location in the stream somewhat away from the bridge, the strength of the sensor's cellular connection was also improved.

Figure 7: Sensor version 3 alternative mounting solution

SOLUTION

Hardware

The device is based around a Particle Boron LTE. This microcontroller is responsible for coordinating all the sensors present on the device and using a cellular data connection to transmit this to the central server. The device consists of many off-the-shelf components in order to reduce cost.



(A) A Particle Boron LTE microcontroller is responsible for coordinating all the sensors present on the device and using a cellular data connection to transmit this to the central server.
(B) A 4G antenna is mounted externally with a 1.5 m cable to improve connection reliability.
(C) A laser time of flight (ToF) distance sensor points towards the ground measuring fluctuations in the height of water flowing through the stream or river. The raw sensor data can then be used as a measure of relative flow rate.
(D) A thermistor mounted under the surface of the water measures the temperature of water flowing through the stream or river.
(E) AA batteries were chosen as the device's power source for their high energy capacity, low leakage current, and resilience to extreme temperature. Cells are wired in series in sets of 3. This comes out to 4.5V with 18 total cells.

Several components on the device were designed custom for this application.

(F) A 3D printed enclosure is waterproof and designed to be easy to install at the field.
(G) An Arduino microcontroller uses a MOSFET to switch power to the main Boron microcontroller. This reduces the device's power consumption and extends battery life significantly.
(H) Conductivity is measured by passing a small current through two aluminum plates. When placed under the water, it can monitor relative conductivity.

Bill of Materials

Item	Price
3D printed body	$4.50
Particle Boron	$50.00
Cellular Antenna	$7.00
ToF Distance Sensor	$10.00
AA Batteries x 18	$11.70
Arduino Pro Mini	$5.00
Thermistor	$2.00
MOSFET	$3.00
Passives, wire, hardware	$10.00
Total	**$103.20**

Software

The software powering the project consists of two main components: the embedded software running on the actual sensor, and the software powering the web server which receives and stores the data from the sensor and provides historical data to users.

There are two microcontrollers on each device. One is responsible for waking up the sensor each hour. The other, cellular connected, microcontroller is responsible for communicating with all of the peripherals and transmitting the data to the main server. Both have software designed to limit power consumption. The wake microcontroller creates an alarm on the attached RTC. This causes it to exit its deep sleep state each hour and send power to the cellular microcontroller. When powered, the cellular microcontroller begins collecting data and attempts to send it. If it is unable to in 10 minutes, it shuts itself off to prevent power wastage. There are actually a total of 3 failsafe shutoffs present on the device. If a subsystem fails or is stuck in code execution, these failsafes trigger, preventing the sensor from draining its battery prematurely. Additionally, we are able to monitor when these failures occur to better troubleshoot possible issues.

The web server serves two main functions: to receive data from the individual sensor units, and to make this data available to users. To submit data to the server, a sensor unit submits an HTTP POST request consisting of a JSON string which contains the current sensor readings. The server parses this string and adds the data to a logfile for that specific sensor unit. This data is then accessible from the interactive website. The website was designed to provide a streamlined user experience and easy access to historical data from all sensor units. All sensor locations are displayed on a map, allowing the user to select the desired sensor. Users can then view graphs of each of the data fields being collected by the sensor, and download JSON or CSV files containing historical data to perform further analysis.

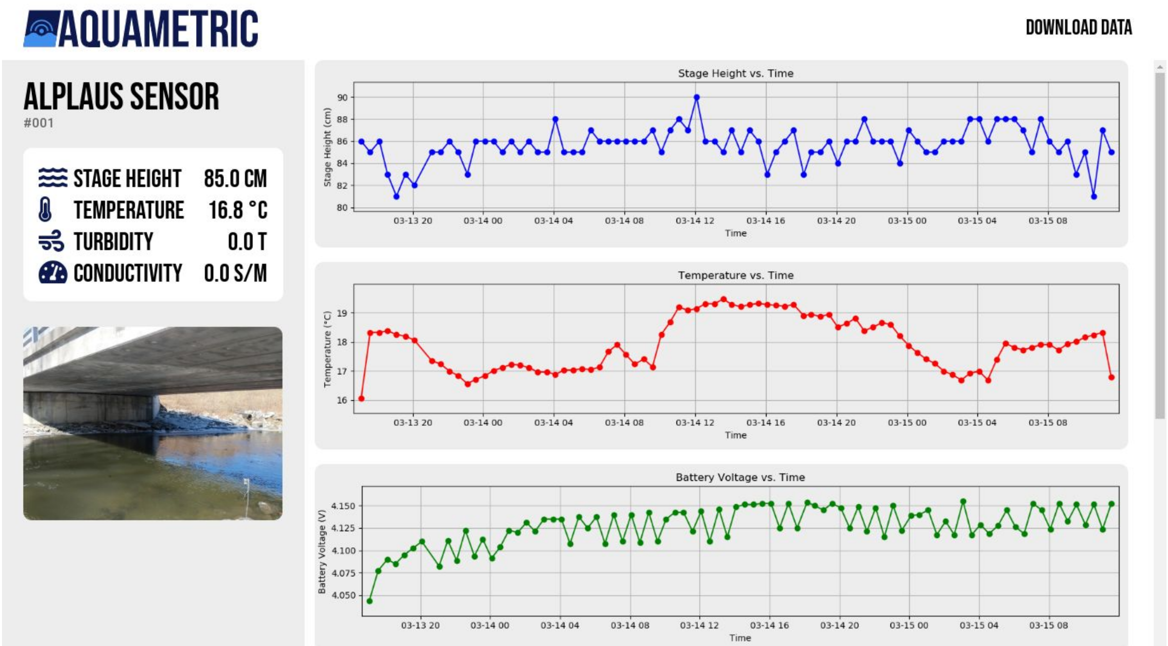


Figure 8: Sensor dashboard displaying current values and plots of values over time

All the code for the sensors and website is open-source, and is available online at https://github.com/ver09934/aquametric.

RESULTS

To determine the effectiveness of the sensors used in the device, we tested them against standard test equipment. The measured values shown below are the averages of 1 reading every 3 seconds for approximately 1 minute, and the standard deviation represents the standard deviation of the data set of measured values from the sensors. The temperature and stage sensors were both found to be accurate enough for our application.

	Measured	Accepted	Standard Deviation
Temperature	15.36°C	15.9°C	0.0452
	1.57°C	1.9°C	0.0405
Stage	446.6 mm	452.12 mm	1.928
	257.5 mm	241.3 mm	1.457

Figure 9: Temperature and stage sensors measured vs. accepted values

The ToF stage sensor showed preliminary issues with measuring distance to the surface of water. Some investigation showed that the waterproofing of the ToF sensor seemed to contribute to measurement errors, but when we replaced the plastic cover on the sensor and expanded the viewport, accurate measurements could be taken.

The conductivity sensor was treated as a relative measurement, meaning that it was used as an indicator of changing conditions rather than as an absolute measurement of dissolved ions. Placing the sensor in 0.5 g of salt per liter of water solution resulted in a significant change in conductivity from distilled water. There was a significant change in conductivity between distilled and tap water.

We found that deploying in shallower, faster-moving water was better to avoid buildup of debris on the sensor. However, we also found that shallower stream beds are more rocky and hard to drive a pole into. A large plate weighted down by rocks was a far more effective mounting solution.

LTE cellular data was found to be less reliable than we predicted. Even with a large external antenna, connectivity is highly location dependent and must be planned for in advance. We found the consistency of sensor connection to be hard to predict, even with locations of nearby cell towers.

Our battery life was found to be sufficient for the device. The device would last about 8 weeks under normal conditions. While this is less than ideal, the benefits of having real time data outweigh the low cost of battery replacement. The battery life could be extended by reducing the frequency of data transmission (eg. every 3 hours rather than hourly).

Protecting an electronic device from the elements proved to be a difficult, but solvable challenge. The 3D printed body had to be coated with an epoxy layer on the inside to ensure no water leakage. A rubber gasket between the lid and main body of the sensor was used to prevent leakage at the seam. We tested various types of plastics to cover the ToF distance sensor to prevent water damage. A 1 cm square of polycarbonate (.5 mm thick) directly in contact with the sensing IC proved to be the most effective to block the elements while passing IR signals.

The web interface for displaying sensor data proved to be effective in viewing data being transmitted real-time from the sensor. It was able to successfully receive data from a deployed sensor and dynamically generate graphs for user analysis. The website was easy to use and aesthetically pleasing.

The device proved to be a cost-effective alternative to more traditional stream monitoring solutions. At around $100, it has the potential to improve the granularity of stream metric monitoring and allow for more widespread adoption. We think further development of this technology could enable organizations such as the USGS to vastly expand their network of stream monitoring devices. This increase in available data could help make ecological changes more predictable.

FUTURE WORK

Connectivity proved to be the most difficult part of deploying this sensor. LTE (4G) data does have some benefits over 3G data, such as bandwidth. However, 4G cell coverage is significantly worse than 3G coverage, especially in the regions we were hoping to measure. In future versions, we would likely switch to using a 3G based microcontroller. It would likely improve the rate at which data is successfully collected.

There are also improvements to be made to the quality of our measurements.

The ToF distance sensor performs best when measuring against matte white surfaces. To improve its reliability, adding a floating white disk around the sensor's mounting poll would likely increase the possible range of stream heights we could monitor. This disk would move freely up and down so its height would vary with the stream. Care would have to be taken to develop this disk to prevent it from being caught in debris.

We would also like to calibrate the conductivity sensor to allow us to read an absolute conductivity instead of a relative value against some known concentration of dissolved ions. To do this, we would calculate the resistance between the two plates of the conductivity sensor, using the fact that the sensor is one of two resistors in a voltage divider. We would then use the formula for resistivity, $\rho = RA/l$, where R is the resistance between the plates, A is the area of one of the plates, and l is the distance between the plates. We would then use the fact that conductivity (σ) is simply the inverse of resistivity. Doing so would give us an absolute measurement for conductivity, and allow us to compare our values against other known measurements.

If we were to scale up the system to support a greater number of sensors and/or run the sensor for significantly longer periods of time, it would be prudent to implement a SQL database system instead of the current system of plaintext files. This would allow for greater versatility in selecting values from specific time slots without having to parse the entire log file for that sensor.

An additional feature that we would like to implement is the short-term caching of sensor data on the unit in the case of a connection failure. In the current design, if an attempt to upload data to the server fails, the data is lost. In our next design iteration, we plan to store this data in the flash memory of the microcontroller. In this manner, when a successful connection is made all the data from previous failed attempts is uploaded in addition to the current data. This way, no data is lost, even if an upload attempt fails.