A recipe for standards-based data sharing using open source software and low-cost electronics

Jeffrey M. Sadler, Daniel P. Ames and Rohit Khattar

ABSTRACT

Environmental data are critical to understanding environmental phenomena, yet their consistent collection and curation can be cost-prohibitive. This paper describes a recipe for the design, development, and deployment of a low-cost environmental data logging and transmission system for environmental sensors and its connection to an open source data-sharing network. The hardware is built using several low-cost, open-source, mass-produced components. The system automatically ingests data into HydroServer, a standards-based server in the open source hydrologic information system (HIS) created by the Consortium of Universities for the Advancement of Hydrologic Sciences Inc. (CUAHSI). By publishing data in this way, they are discoverable through the geographic information system (GIS)-based CUAHSI tools, HydroDesktop and HydroShare. In addition, because they follow WaterML encoding, open hardware data stored in the HIS can be included in international catalog such as the global earth observation system of system catalog. A recipe for building the system is provided. Multiple deployments used to test proof-of-concept of the system are described and their results are given. Ease of deployment and reliability of the logging and transmission system is also addressed.

Key words | environmental monitoring, open-source hardware, open-source software, sensor networks

Jeffrey M. Sadler
Daniel P. Ames (corresponding author)
Rohit Khattar
Civil and Environmental Engineering,
Brigham Young University,
368 Clyde Building,
Provo,
UT 84602,
USA
E-mail: *dan.ames@byu.edu*

INTRODUCTION

It is generally accepted that to understand and address environmental phenomena, environmental observations are required (Baker 1936). While important, it is often very costly to install and maintain systems that collect environmental data at high spatial and temporal resolutions; thus cost is a common barrier to exhaustive environmental monitoring. This barrier affects many organizations including researchers, educators, and government agencies around the world. For example, more than 200 USGS gauging stations have been recently discontinued, with dozens more threatened with the same fate due to funding issues (US Geological Survey 2014).

High costs are especially problematic when sensors are deployed for a brief time period, in a high-risk environment, or in developing countries with limited resources. A short-term investigation of an area requiring several nodes may be needed for an environmental model calibration (Quinn *et al.* 2010). In these cases, an investment of several thousand dollars for equipment to be used only for short-term deployments may be infeasible. When several nodes are needed, costs, and thus infeasibility, rise quickly (Younis & Akkaya 2008). Another example where the expense of sensor systems may be an especially significant barrier is when the equipment is likely to be, or intended to be, destroyed (Oliveira & Rodrigues 2011; Kean *et al.* 2012). A final example is the collection of data in developing countries where the capital required to invest in expensive monitoring and telemetry equipment may be in short supply (Basha & Rus 2007). For example, since flooding is one of the costliest natural disasters, stream stage and rainfall data are invaluable to countries that are especially affected by these events, such as many of the developing countries in Latin America

doi: 10.2166/hydro.2015.092

(Baker 1936; Berz 2000; Charvériat 2000). The costs associated with purchasing, implementing, and maintaining equipment needed to collect stream level and rainfall data can be burdensome to developing economies.

In recent years, the advancement of open-source hardware has sparked significant reductions in the cost of scientific equipment in many technical areas. A key contributor to this movement is the Arduino, an open-source, programable microcontroller. This easy-to-use electronics prototyping platform is adaptable to both simple and complex tasks (Arduino 2014). For example, an Arduino has been coupled with 3D printers to manufacture custom equipment for experimental laboratories, significantly reducing costs (Pearce 2012; Anzlone *et al.* 2013). Similar technology has been used to address broad problems such as energy availability and efficiency (Pourmirza & Brooke 2013; Mousa & Claudel 2014). More specifically, researchers have used low-cost open-source hardware in environmental monitoring to measure parameters such as latrine usage, snow depth, soil moisture, air quality, marine environments, and even pathogen levels (Lundquist & Lott 2008; Clasen *et al.* 2012; Trevathan *et al.* 2012; Fedi *et al.* 2013; Mitchell *et al.* 2014).

In addition to hardware costs, difficulties with the interoperability can threaten the value of environmental data. As science becomes increasingly data-intensive and collaborative, the need for easily shareable data grows as well (Michener *et al.* 2012). The ability to share heterogeneous data from a variety of sources can be greatly enhanced through the use of data structure and sharing standard formats (Giuliani *et al.* 2011). While the importance of data sharing is generally accepted and can be facilitated through industry standards, in many cases the effort needed to make data shareable is a considerable hindrance (Borgman 2012).

Lower financial and technical barriers for data collection have increased the amount of data that can be affordably collected (Henson *et al.* 2013). Recently, plans for managing and sharing data collected for research have become heavily emphasized by funding institutions such as the US National Science Foundation (Tenopir *et al.* 2011).

The effective and creative use of newly developed and more affordable technologies to increase the spatial and temporal resolutions of environmental data is an active area of research (Newman *et al.* 2012; Hut 2013; Zaslavsky *et al.* 2013; van de Giesen *et al.* 2014). A number of low-cost hardware solutions to collect and transmit environmental data have been developed and described (Baker 2014; Wickert 2014).

An open hardware system developed by the Stroud Research Center is used to collect data in the Christina River Basin Critical Zone Observatory. With Arduino boards and other open-source electronics such as Zigbee radios, the system uses several custom sensor modules including a radio reporting stream gauge, soil moisture sensor systems, pressure transducer readout, and respirometer controllers. These data are made available in WaterML1.1 format via WaterOneFlow web services created by the Consortium of Universities for the Advancement of Hydrologic Sciences Inc. (CUAHSI) (Hicks *et al.* 2013).

Our goal is to improve upon the methods used in the cases above for managing and sharing collected data. For example, Trevathan *et al.* (2012) and Fedi *et al.* (2013) do not describe a specific plan to make the collected data publicly available, and the latrine usage data described by Clasen *et al.* (2012) were stored on an SD memory card but not made publicly available. The data storage and sharing schema of Hicks *et al.* (2013) are quite extensive, with streaming data viewable via a custom website and accessible through web services, but lacks implementation of Open Geospatial Consortium (OGC) standards.

To improve the management and shareability of the data collected by low-cost systems, our system automatically inserts observed data into a customized, OGC-compliant CUAHSI HydroServer (Kadlec & Ames 2012). This approach leverages the capabilities and standards of the relatively mature CUAHSI hydrologic information system (HIS) (CUAHSI 2014). Hence this solution provides a lightweight, low-cost connector that could be used to automatically ingest data collected by existing and future low-cost data collection systems into a federated, standards-based software system for sharing time series data.

The HIS enables standards-based storage, sharing, publication, and discovery of time series data (Horsburgh *et al.* 2009). There are over 85 data sources currently cataloged in the CUAHSI Central Catalog, some of which use sensor data streams. Our work builds on and adds to such capabilities by enabling real-time data streaming from low-cost hardware through a custom HydroServer that supports the OGC Web Feature Service (WFS) and the newly

adopted OGC WaterML2.0 standard (Taylor *et al.* 2014). Our custom HydroServer, called 'HydroServer Lite', is built on the open source Linux-Apache-MySQL-PHP (LAMP) software stack, requiring only a simple computer and inexpensive, or free, webhosting. Because it does not use closed-source hardware or software, the system is extremely inexpensive and light-weight to deploy and hence can be highly suitable in rapid deployment, low-cost, short term, underfunded, or citizen science monitoring efforts (Buytaert *et al.* 2014). This paper is the first complete description of the integration of low-cost environmental data collection equipment and a free, open-source data management system that follows industry standards (i.e. WaterML1.0 and WaterML2.0).

The remainder of the paper describes the methods used to develop a low-cost data logging and transmission system configured to automatically input collected data into a HydroServer – including a recipe for others to follow to create a similar system. We also outline methods for testing the success of the hardware, the interoperability of the data, and, the ease-of-configuration of the hardware. The results of each of these assessments are given, followed by a brief conclusion.

METHODS

Hardware

In selecting hardware components for the environmental data logging and transmission system, our primary aims were to build a unit that was self-contained, low-cost, and relatively easy to build (e.g. with minimal soldering). The module also had to accommodate the automatic ingestion of collected data into the CUAHSI HIS. These criteria guided design decisions. The two main electronics components we used were an Arduino Uno and a SIMCOM SIM900 Quad-band GSM GPRS Shield. These are shown in Figure 1 along with the case, temperature and humidity sensor, and solar power system we used.

We chose the Arduino Uno (Arduino 2014) as an easy-to-use and inexpensive prototyping platform. We initially used a DHT22 temperature and humidity sensor, and later made the system compatible with SDI-12 (serial data interface at

Figure 1 | Arduino, solar panel, battery, sensor, and GPRS modem.

1,200 baud) (www.sdi-12.org/index.php) sensors to allow the use of any of the many SDI-12 sensors available, many of which are generally well-respected and trusted as manifested by their use in organizations including the USGS (Fitzpatrick 2010). The SDI-12 sensor used in this research was a KPSI pressure transducer water level sensor that meets USGS standards.

We chose to use the cell phone network to eliminate the need for expensive long-range radio telemetry, and to avoid the spatial limitations of short-range radios like Zigbees; the system need only be deployed in a location with cell phone coverage. Furthermore, cell phone coverage is spreading rapidly everywhere, including in developing countries which could greatly benefit from low-cost sensor and data-transmission systems (Thomson *et al.* 2012). For power, a 6 W solar panel with an accompanying 4,000 mAh lithium ion battery were purchased for a total of $88 USD. This constituted the largest portion of the total cost.

Software

To automatically insert the collected data into our custom HydroServer, an HTTP GET request is transmitted over

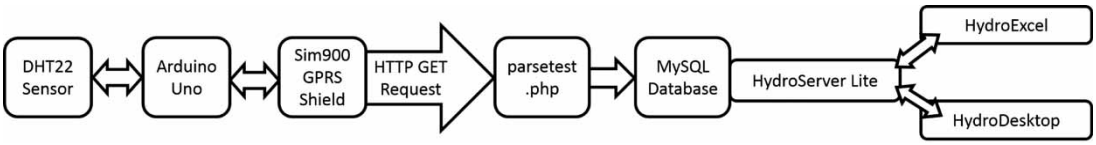

Figure 2 | Workflow of data from sensor to user.

the GPRS network. On the server, a PHP script (i.e. 'parsetest.php') parses the data into an SQL query which inserts the metadata and data into the MySQL version of the observations data model (ODM) (Horsburgh *et al.* 2009) with front-end access through HydroServer. This workflow is shown in Figure 2. HydroServer provides a simplistic, web-based user-interface for adding, editing, and deleting raw data and metadata in the ODM (Kadlec & Ames 2012; Conner *et al.* 2013). Notwithstanding its low cost and simplicity, HydroServer provides many useful features, including a web map interface showing the locations of data-collection sites, graphical and tabular representations of the collected data, and download functionality.

In addition to being free and simple to use, our custom HydroServer is an endpoint for web services including CUAHSI WaterOneFlow services. The HIS provides a well-defined system and several software tools to make data easily discoverable and downloadable in standards-compliant WaterML1.1 format through WaterOneFlow web services. Through these services the data on a HydroServer can be accessed through REST and SOAP services, making the data discoverable and downloadable by client tools such as HydroExcel and HydroDesktop (Ames *et al.* 2012). As an endpoint for WFS, HydroServer makes the data interoperable across platforms such as ArcGIS and Quantum GIS. In addition, with WaterML1.1 and WaterML2.0, data on a HydroServer can be included in international catalogs such as the global earth observation system of system (GEOSS) catalog.

Implementation recipe

This section is an implementation guide for the system describing the HydroServer installation and giving step-by-step guidance for the hardware configuration. To use the PHP based HydroServer, the primary requirement is basic web-hosting with access to a MySQL database and permissions to run PHP scripts. Once proper hosting is acquired, the HydroServer source code must be downloaded and copied onto the webserver domain. The source code and details about installation and configuration for use with the hardware in this paper are available at the project's CodePlex site (HydroServer Lite 2014).

Regarding hardware, we used two independent but similar configurations. Our first test case used the DHT22 sensor and secondly we used the SDI-12 pressure transducer. Assembling the system hardware was relatively simple, requiring minimal soldering and very few tools (i.e. drill, soldering iron, solder, pin-wire connector, male-to-male jumpers). All of the parts were purchased online and are listed, excluding sensors, along with their prices and online purchasing locations in Table 1. Table 2 lists the same information for the two sensors we used.

The hardware system can be assembled in six steps:

1. Attaching the GPRS shield to the Arduino.
2. Preparing the weatherproof box.

Table 1 | Individual item costs, purchasing locations, and total cost of hardware for logging, transmission, case, and solar power

Item (where to purchase)	Cost
Arduino Uno (www.amazon.com/Arduino-UNO-board-DIP-ATmega328P/dp/B006H06TVG)	$16
SIMCOM SIM900 Quad-band GSM GPRS Shield (www.ebay.com/itm/like/151121095914?lpid=82)	$35
Miniature breadboard (www.sparkfun.com/products/12045)	$4
SIM card and phone plan (www.ptel.com/phones)	$15
MicroSD Card (www.amazon.com/b?node=3015433011)	$10
Voltaic 6W Solar Panel and 4,000 mAh Battery (www.voltaicsystems.com/6-watt-kit)	$90
Pelican 1050 Clear Micro Case (www.bhphotovideo.com/bnh/controller/home?O=&sku=257883&gclid=CjwKEAiAj)	$15
Total cost	$181

Table 2 | Costs and online locations of sensors used

Item (where to purchase)	Cost
DHT22 Temperature and humidity sensor (www.sparkfun.com/products/10167)	$10
SDI-12 Pressure transducer (www.meas-spec.com/product/t_product.aspx?id=9023#)	~$700

3. Connecting the sensor to jumpers/wires.
4. Wiring the sensor jumpers to GPRS shield via the mini-breadboard.
5. Uploading sketch (custom source code) to Arduino.
6. Connecting the system to power source.

First, the GPRS shield was attached to the Arduino board. This was done by simply inserting the shield male pins into the Arduino female pins so that the shield sits on top of the Arduino board. Second, since the sensors have to be outside of the weatherproof box, we drilled a hole large enough for the wires to fit through, about 1 cm wide. Thereafter, the wiring was done through this hole. Third, to connect the sensors to the GPRS shield we used solderless breadboards to minimize soldering. For the DHT22, we needed to use a connector to connect the pins of the sensor to wires. The SDI-12 pressure transducer came prewired, but we needed to solder male-to-male jumpers with a gauge small enough to fit into the breadboard. With the connector from the DHT22 pins and jumpers soldered onto the SDI-12 pressure transducer wires we were able to make the connections into the breadboards. Fourth, we used male-to-male jumpers between the breadboards, connected to the sensors and the GPRS shield. The connections to the sensors are shown in Figures 3 and 4.

Fifth, the sketch (source code) was uploaded using an A/B USB cable. A sketch is the program that the Arduino runs. Sketches are written in a programing language derived from C++ and are uploaded using the free Arduino IDE (Arduino 2014). We used existing Arduino libraries to interface with the GPRS shield and the two sensors. For interaction between with the GPRS shield, the built-in Software Serial library was used. A third party open source library was used for the DHT22 sensor (Adams 2012). The compatibility with SDI-12 sensors was provided by a library written by the Stroud Research Center (Hicks *et al.* 2013; Stroud Research Center 2014). To make the Software Serial library and the SDI-12 libraries run at the same time, a small modification had to be made to both codes. This change, more detailed instructions, pictures, and all the sketches that we used can be accessed on GitHub (Sadler 2014).

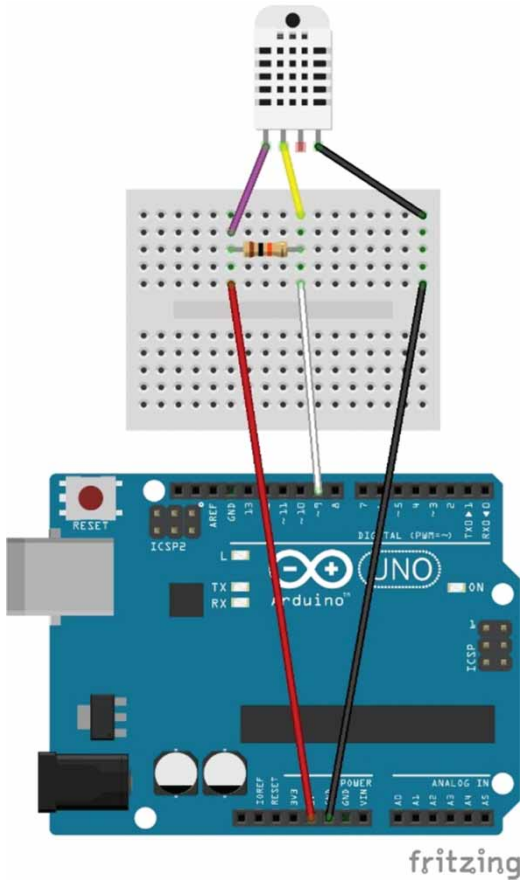

Figure 3 | Wiring diagram for DHT22 and Arduino.

To adapt the Arduino code for a specific use, several variables in the sketch must be modified. These variables are all included in the HTTP GET sent by the GPRS shield. A proper request includes the URL corresponding to the desired HydroServer, the source id, the site id, and the variable id. It is noted that, although all the correct parameters (URL, source id, etc.) would be required for a malicious or negligent data entry into the system, there is no explicit authentication for the submission of data with the current configuration.

The final step is to connect the device to a power source. With the DHT22, we used the solar power device from

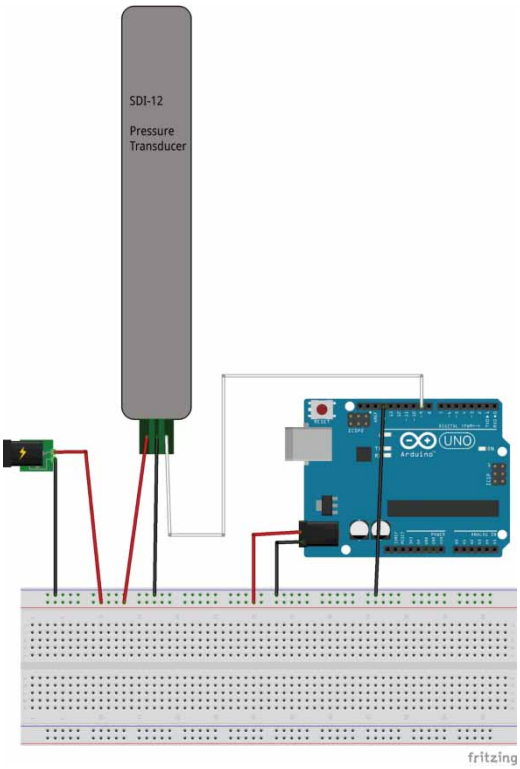

Figure 4 | Wiring diagram for SDI-12 pressure transducer and Arduino.

voltaic systems (see Table 1). The panel and battery are easily connected with the adapters included with the shipment. The Arduino connects to the battery using the same A/B USB cord that connects the Arduino to the computer. To set the battery to 'always-on' mode, the proper setting for continuous monitoring, we pressed and held the power button for 6 s. With the pressure transducer we used a wired power source. While the DHT22 sensor was powered directly from the Arduino board, the SDI-12 required an external power supply. In this case, we used a wall-wart to power the rails on the breadboard. The rails then powered jumpers from the sensor and wires connected to a 2.1 mm center-positive plug inserted into the Arduinos power jack (see Figure 4).

Hardware testing

Three deployments of the system were conducted to demonstrate proof-of-concept of the low-cost environmental sensor, the link to HydroServer, and the interoperability of collected data made possible by web services. Each deployment took place in Provo, Utah, USA and in each deployment, data were collected every 15 min. In the first two deployments, the data were transmitted every hour but in the third deployment the data were sent every 15 min. After each deployment, the transmission success rate was recorded and reported. In the first deployment, the sensor system was deployed on the side of a building where temperature and relative humidity data were collected using the DHT22 sensor. The deployment was powered using the solar panel and battery described above. In this deployment, the collected data were received by the webserver and simply saved to a text file.

In the second deployment, the sensor system was placed on the roof of a three-storey university building on the Brigham Young University campus. As in the first deployment, temperature and relative humidity values were collected by the DHT22 sensor and the unit was powered by the solar panel and battery. Unlike the first deployment, however, the data in this deployment were inserted automatically into the ODM and thus were made accessible in HydroServer.

The third deployment was conducted on the Provo River. There, the water level in a stilling basin was recorded by the SDI-12 pressure transducer. The location was the Harbor Drive gauging station in Provo run by the Central Utah Water Conservancy District (CUWCD). The sensor used by the CUWCD is a guided-wave radar level sensor, and their data is transmitted via radio. In this deployment, the low-cost sensor system was powered using the CUWCD power source. The values collected and transmitted by the CUWCD were compared with the values collected by the system presented in this paper as a method of proving validity.

Data interoperability testing

To validate the open hardware/open software interconnectivity via custom PHP scripts and the HydroServer ODM database schema used in the second and third deployments, data search and discovery tasks were performed using two distinct data access systems: HydroDesktop and the open source geographic information system, Quantum GIS (QGIS 2014).

HydroDesktop was used to test WaterOneFlow and WaterML 1.1 web services access to the collected data.

With the WSDL URL of the HydroServer, HydroDesktop was used to gather the metadata on the server including the Harbor Drive metadata (site geolocation, variables collected at the site, etc.). Once the metadata were gathered, the sites were discoverable data sources within the program. The Harbor Drive site was discovered and represented as a point on an online basemap using the HydroDesktop search function based on spatial extents, variable, and time frame (i.e. 'Utah County', 'Stream Level', and '7/1/14–8/31/14' in this case). HydroDesktop provides several data viewing and processing tools including statistical tools such as an R portal, and a plotting tool. After the Harbor Drive site was discovered on the HydroDesktop map, the data were downloaded into the program by clicking a link in the Harbor Drive site popup window. The data were then plotted using the HydroDesktop plotting function.

Within Quantum GIS, our objective was to view the Harbor Drive site as a geographic information system (GIS) feature through WFS and to access the WaterML2.0 document of the time series data at the site. To use WFS in Quantum GIS, the WFS URL provided by HydroServer was input into the 'Add WFS Service' function. The service added features according to collected variable. We added all the sites where water level was collected. After the service call was complete, these points were added to the map as a new feature set. The attribute table of the new feature set, including the Harbor Drive site, contained URLs for each site corresponding with WaterML2.0 time series documents. Results of the HydroDesktop and Quantum GIS testing are provided in the Results and discussion section below.

Usability testing

To test the usability of the system, data from a user survey were gathered from a K-12 teacher's workshop. In the 105 min workshop, 16 teachers were organized into four groups and were given instructions and materials to assemble a temperature/humidity sensor system. At the end of the workshop the teachers were given surveys. Fourteen of the 16 teachers returned the survey. Eight of the teachers taught high school classes (grades 9–12) and six taught in middle schools (grades 6–8). All but one teacher indicated that they regularly taught sciences and one answer was left blank. The survey questions were focused on competence, difficulty, perceived educational value of the technology, interest, barriers, and applications in the classroom.

RESULTS AND DISCUSSION

Results of the hardware, data interoperability, and usability tests described above are provided in this section.

Hardware testing results

First deployment

The first deployment shown in Figure 5 was conducted over a 10-day period during which 1,696 data points were collected and transmitted via HTTP request to the webserver. The data, including the timestamp, were stored in a text file on the webserver. With ample sunlight on these days, the sensor was able to collect and transmit data without losing power throughout the test period. A graphical representation of the temperature and humidity data is shown in Figure 6. After retrieving the data, we noticed that the temperature values were invalid when the temperature was below zero, due to a problem in the library we used. That said, we were not overly concerned with this because the focus of this deployment was not on the actual environmental data collected by the sensors. Rather, we focused on the success of the data logging and transmission and the interoperability of the data once received on the server.

Figure 5 | Sensor in first deployment location. The temperature sensor, not visible here, is external to the case.

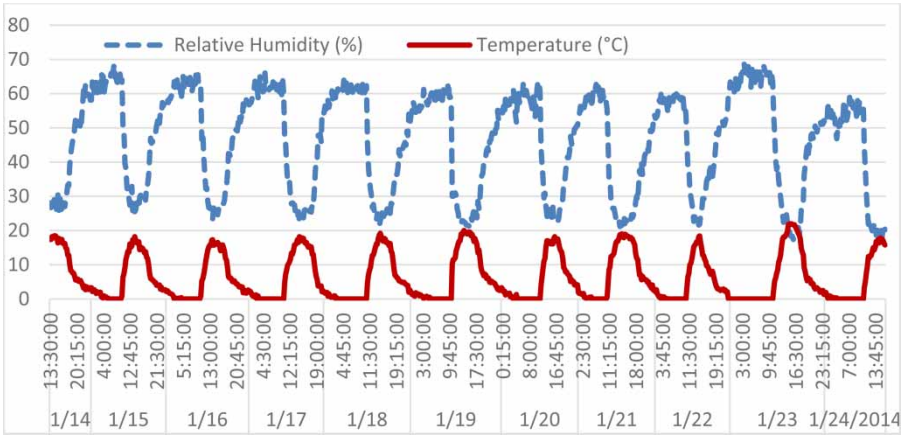

Figure 6 | Temperature and humidity values collected by sensor in first deployment.

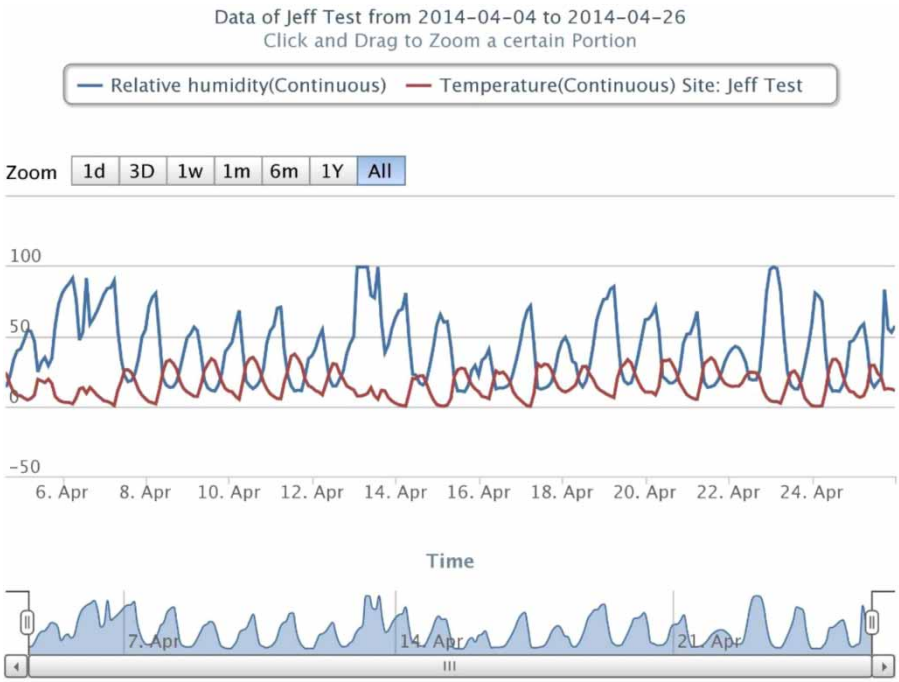

Figure 7 | Plot of data collected in second deployment produced by HydroServer.

In the 10-day period, there were 30 instances in which the data were not successfully transmitted to the server. This is a transmission success rate of 96%. Of the 30 instances, a failure in consecutive hours only occurred three times. As an attempt to address this issue, a redundancy was written into the Arduino code. With the redundancy, at each hour the GPRS modem transmitted the data for the past 2 h. In this way, each set of data was transmitted twice in case of a problem with one of the connections.

Second deployment

The second deployment was conducted over a 22-day period in which 4,248 data points were collected for both relative humidity and temperature. Only eight transmission failures

occurred out of the 2,132 attempts. This corresponds to a 99.6% transmission success rate. This increase in success rate, 96–99.6%, suggests that the redundancy was effective. The data in this deployment were successfully inserted into the MySQL ODM and were thus accessible by HydroServer and the CUAHSI HIS. A graphical representation, produced by HydroServer, of the data collected in the second deployment is shown in Figure 7.

Third deployment

The third deployment was conducted over a 40-day period in which 3,380 data points of stream level were collected. The overall transmission success rate for this deployment was 88.16%. In the 40 days, the transmission stopped twice and needed to be manually reset. These two periods of no data lasted for 48 h and 45 h, respectively. The reason the transmission failed in these periods is uncertain. Several factors may have contributed, including an error in the cell phone network, loss of power, or some malfunction in the GPRS board. Not taking into account the two extended periods of transmission failure, the transmission success rate was 97.58%. The data collected by the low-cost sensor and the data collected by the CUWCD sensor are shown in Figure 8. As seen in the figure, the values are very similar. When plotted against each other, Figure 9 results. The r^2 value of the two data series is very high, 0.9963.

Data interoperability testing results

Data interoperability testing using HydroDesktop and Quantum GIS yielded positive results indicating that the custom OGC standards-based HydroServer was successfully coupled with custom Ardiuno and PHP scripts. In both cases (HydroDesktop and Quantum GIS), all Harbor Drive data stored within the ODM database were retrieved from the custom HydroServer with 0% loss.

Figure 10 shows a screen capture of the HydroDesktop software having completed a data search in the area of Harbor Drive. Successful execution of the search is illustrated by the appearance of an icon at the proper latitude and longitude of the Harbor Drive site. An informative 'pop up' bubble with the site name and a data download link appears when hovering the mouse cursor over the icon. By clicking the download link, we were able to successfully retrieve the WaterML 1.1 document representing the data logged at this site. A plot of these data, as generated by HydroDesktop, is shown in Figure 11. Retrieval and visualization of

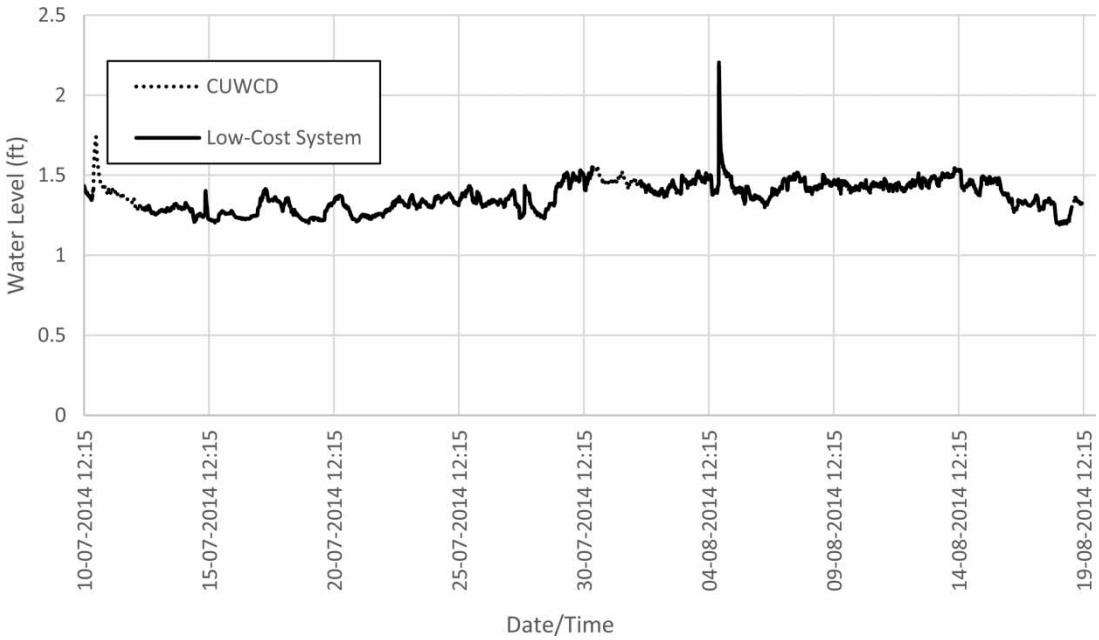

Figure 8 | Stream level as reported by low-cost system and CUWCD system.

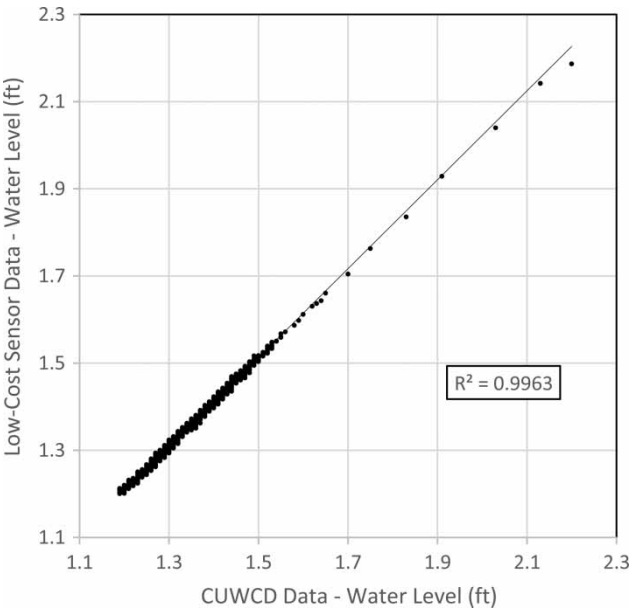

Figure 9 | Low-cost system and CUWCD system values plotted against each other.

the Harbor Drive dataset via HydroDesktop indicates successful 'closing the loop' from data collection, logging, transmittal, server storage, and web-services based access.

Successful data discovery using Quantum GIS is illustrated via the software screen capture shown in Figure 12. Here, we registered the WFS feature data end point within Quantum GIS and successfully retrieved a single point feature that is displayed in the map at the correct Harbor Drive location. Next, we open the Quantum GIS attribute table viewer for this point feature and observe the site metadata (stored as feature attributes) including a direct link to the OGC compliant WaterML 2.0 document containing the Harbor Drive data. The link is provided as a REST URL endpoint. This URL was then used to successfully retrieve the WaterML 2.0 document from the custom HydroServer.

Usability workshop results

The usability workshop gave some insight into the difficulties in assembling a low-cost hardware system for environmental data collection. In the usability workshop survey both quantitative and qualitative questions were used. The quantitative questions were based on a six number scale (1–6). On the scale, one corresponded to 'less confident', 'less difficult', 'less interested', and 'less valuable', and six to 'more confident', etc. Because of the small sample size we cannot draw any quantitative conclusions; however, some qualitative results were interesting. Perhaps

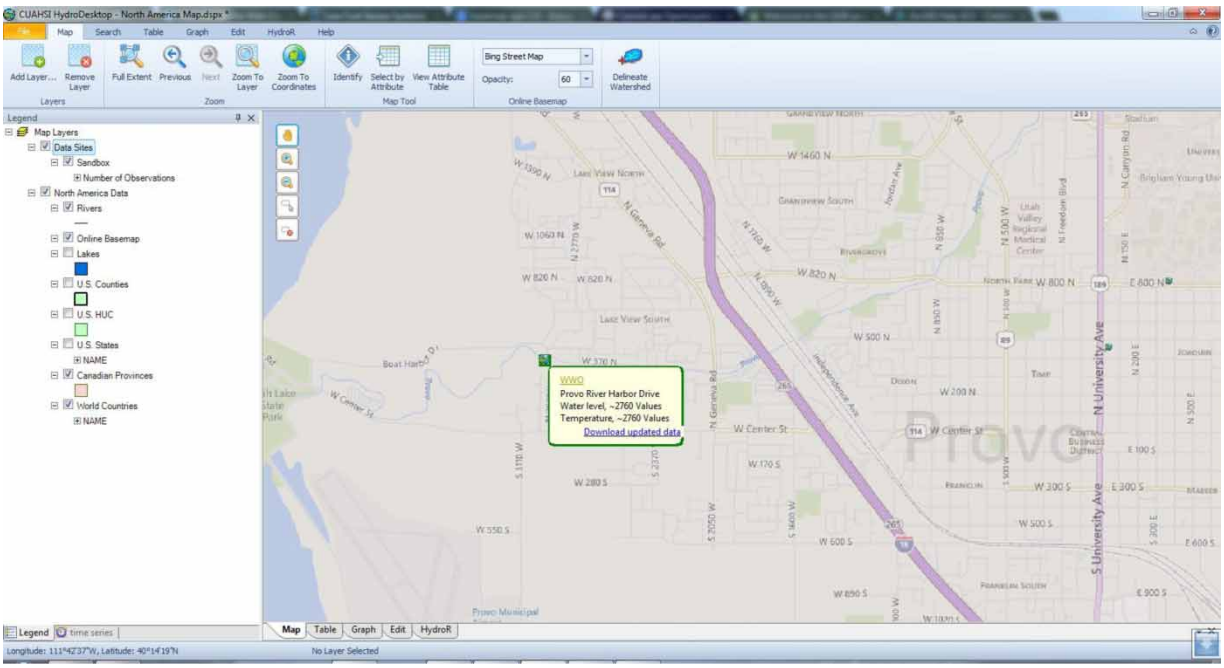

Figure 10 | Harbor Drive site in HydroDesktop.

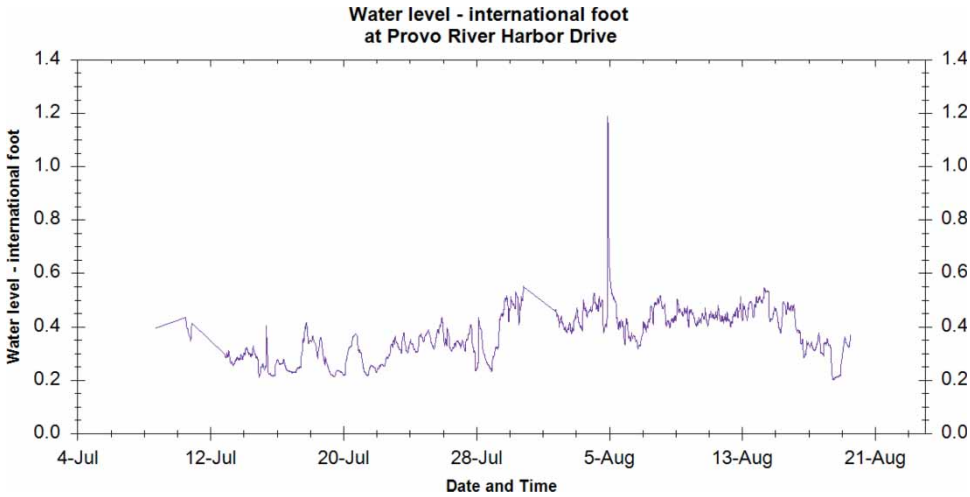

Figure 11 | HydroDesktop plot of data at Harbor Drive site.

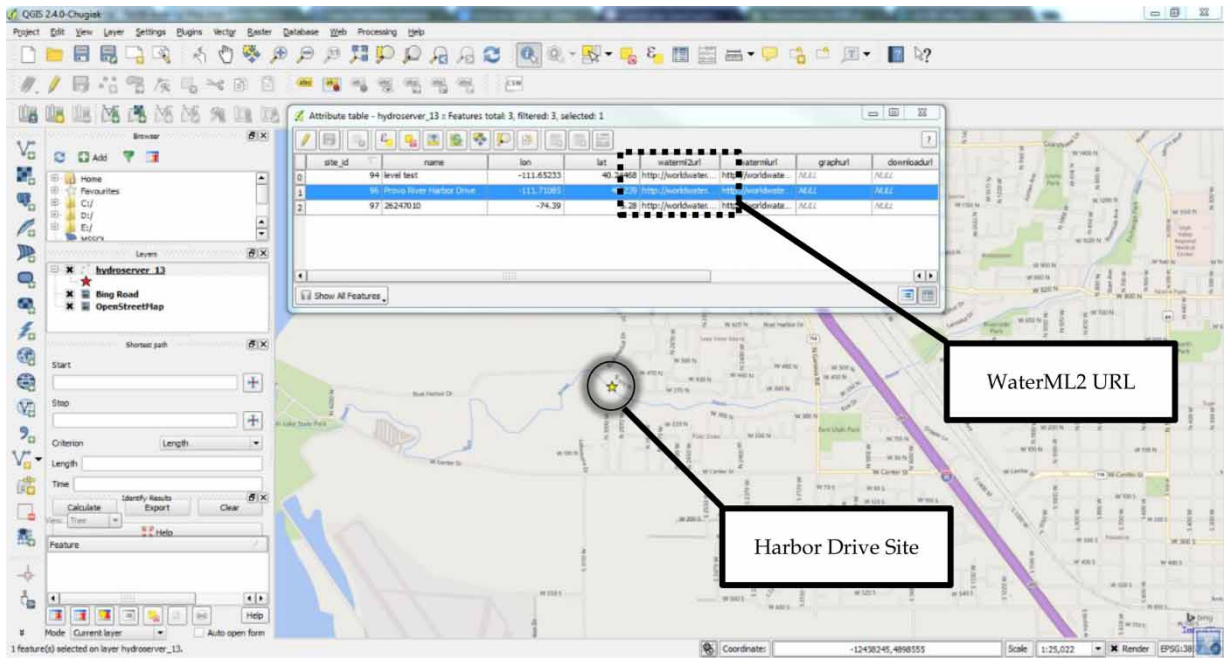

Figure 12 | Harbor Drive site and WaterML2 link in QGIS.

most notable were the teachers' responses when asked about the most difficult part of the exercise. The majority (11 of 14) reported that getting the software working was the most difficult part. This part of the exercise included downloading the libraries needed for the DHT22 sensor and the real-time clock. To use these libraries, they had to be unzipped, renamed, and placed in the libraries folder of the Arduino IDE. This process was difficult for the teachers.

CONCLUSIONS

Hydrologic data logging and transmission hardware as well as software to curate and make collected data shareable can be prohibitively expensive. Often data interoperability is a problem. This paper presents a solution including a recipe and test cases using open hardware for data logging and transmission and the automatic ingestion of its data into

the free and open source CUAHSI HIS. Excluding sensor, the total cost of the data logging and transmission hardware, including solar powering system, was $181. The system was successful as a low-cost solution for gathering environmental data, viewing that data in near-real-time, and storing the data in a standards-based system allowing for easy access and sharing through the widely used WaterOneFlow web services and WFS services.

Our results did reveal some shortcomings. The failure of data transmission, likely caused by either loss of power or loss of cell phone reception, was the primary area that could be improved in the system. Despite some transmission failure, the lowest transmission success rate of the system was over 88%. Programing Arduino to react according to the HTTP Response or using SMS for data transmission are possible solutions to this problem. The autonomy of the GPRS shield was also a problem in the third deployment. To make the hardware deployment more usable, the workshop results suggest focusing on the software needed to run the Arduino. Providing a downloadable version of the software, sketch, and any other needed external libraries would be a way to simplify this process in the future.

Our results serve as a proof-of-concept for a low-cost hardware system that automatically inserts data into a standards based HIS for data distribution. Future work in this area should include additional testing and refinement of the approach presented. Long-term testing would need to be performed to assess long-term deployment reliability. The use of many nodes on the same system is another area that should be tested. Finally, the integration of collected data into CUAHSI's newest software, HydroShare, as a Referenced Time Series resource would be a beneficial use of this technology.

ACKNOWLEDGEMENTS

We thank Measurement Specialties in Virginia for the generous donation of the KPSI 500 Pressure Transducer and Central Utah Water Conservancy District for their sharing of their stilling well, data, and time.

REFERENCES

Adams, B. 2012 nethoncho/Arduino-DHT22. In: GitHub. Available from: https://github.com/nethoncho/Arduino-DHT22 (accessed February 2014).

Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T. & Valentine, D. 2012 Hydrodesktop: web services-based software for hydrologic data discovery, download, visualization, and analysis. *Environ. Model. Softw.* **37**, 146–156.

Anzlone, G. C., Glover, A. G. & Pearce, J. M. 2013 Open-source colorimeter. *Sensors* **13** (4), 5338–5346.

Arduino 2014 Arduino – Home. www.arduino.cc/ (accessed 15 August 2014).

Baker, D. M. 1936 Basic hydrologic data. *Trans. Am. Geophys. Union* **17** (2), 481–486.

Baker, M. E. 2014 Open source data logger for low-cost environmental monitoring. *Biodiversity Data J.* (2), doi: 10.3897/BDJ.2.e1059

Basha, E. & Rus, D. 2007 Design of early warning flood detection systems for developing countries. In: *Information and Communication Technologies and Development, 2007 ICTD 2007 International Conference IEEE*, pp. 1–10.

Berz, G. 2000 Flood disasters: lessons from the past – worries for the future. *Proc. ICE-Water Maritime Eng.* **142** (1), 3–8.

Borgman, C. L. 2012 The conundrum of sharing research data. *J. Am. Soc. Inf. Sci. Technol.* **63** (6), 1059–1078.

Buytaert, W., Zulkafli, Z., Grainger, S., Acosta, L., Alemie, T. C., Bastiaensen, J., De Bièvre, B., Bhusal, J., Clark, J. & Dewulf, A. 2014 Citizen science in hydrology and water resources: opportunities for knowledge generation, ecosystem service management, and sustainable development. *Hydrosphere* **2**, 26.

Charvériat, C. 2000 Natural disasters in Latin America and the Caribbean: An overview of risk. Working Paper, Inter-American Development Bank, Research Department.

Clasen, T., Fabini, D., Boisson, S., Taneja, J., Song, J., Aichinger, E., Bui, A., Dadashi, S., Schmidt, W.-P., Burt, Z. & Nelson, K. L. 2012 Making sanitation count: developing and testing a device for assessing latrine use in low-income settings. *Environ. Sci. Technol.* **46** (6), 3295–3303.

Conner, L. G., Ames, D. P. & Gill, R. A. 2013 Hydroserver Lite as an open source solution for archiving and sharing environmental data for independent university labs. *Ecol. Inform.* **18**, 171–177.

CUAHSI 2014 CUAHSI Hydrologic Information System (CUAHSI-HIS). http://his.cuahsi.org/ (accessed 1 August 2014).

Fedi, A., Ferrari, D., Lima, M., Pintus, F. & Versace, C. 2013 Acronet paradigm: an open hardware project. In: *2nd Open Water Symposium*, Brussels.

Fitzpatrick, C. 2010 New KPSI high accuracy SDI-12 transducer meets monitoring specifications. *Water* **21**. http://www.iwapublishing.co.uk/template.cfm?name=w21prodnews300310e.

Giuliani, G., Ray, N., Schwarzer, S., De Bono, A., Peduzzi, P., Dao, Q.-H., Van Woerden, J., Witt, R., Beniston, M. &

Lehmann, A. 2011 Sharing environmental data through GEOSS. *Int. J. Appl. Geospat. Res.* **2** (1), 1–17.

Henson, C., Barnaghi, P. & Sheth, A. 2013 From data to actionable knowledge: big data challenges in the web of things. *IEEE Intell. Syst.* **28** (6), 6–11.

Hicks, S., Damiano, S., Smith, K., Olexy, J., Horsburgh, J., Mayorga, E. & Aufdenkampe, A. 2013 An open-source wireless sensor stack: from Arduino to SDI-12 to water one flow. *In: AGU Fall Meeting Abstracts* **1**, 1572.

Horsburgh, J. S., Tarboton, D. G., Piasecki, M., Maidment, D. R., Zaslavsky, I., Valentine, D. & Whitenack, T. 2009 An integrated system for publishing environmental observations data. *Environ. Model. Softw.* **24** (8), 879–888.

Hut, R. 2013 New Observational Tools and Datasources for Hydrology: Hydrological data Unlocked by Tinkering. Thesis.

HydroServer Lite 2014 HydroServer Lite – Home. http://hydroserverlite.codeplex.com (accessed 10 August 2014).

Kadlec, J. & Ames, D. P. 2012 Development of a lightweight hydroserver and hydrologic data hosting website. In: *Proceedings of the AWRA Spring Specialty Conference on GIS and Water Resources, New Orleans*.

Kean, J. W., Staley, D. M., Leeper, R. J., Schmidt, K. M. & Gartner, J. E. 2012 A low-cost method to measure the timing of postfire flash floods and debris flows relative to rainfall. *Water Resour. Res.* **48** (5), doi: 10.1029/2011WR011460.

Lundquist, J. D. & Lott, F. 2008 Using inexpensive temperature sensors to monitor the duration and heterogeneity of snow-covered areas. *Water Resour. Res.* **44** (4), doi: 10.1029/2008WR007035.

Michener, W. K., Allard, S., Budden, A., Cook, R. B., Douglass, K., Frame, M., Kelling, S., Koskela, R., Tenopir, C. & Vieglais, D. A. 2012 Participatory design of DataONE – enabling cyberinfrastructure for the biological and environmental sciences. *Ecol. Inform.* **11**, 5–15.

Mitchell, K. A., Chua, B. & Son, A. 2014 Development of first generation in-situ pathogen detection system (Gen1-IPDS) based on NanoGene assay for near real time *E. coli* O157: H7 detection. *Biosens. Bioelectron.* **54**, 229–236.

Mousa, M. & Claudel, C. 2014 *Energy Parameter Estimation in Solar Powered Wireless Sensor Networks Real-World Wireless Sensor Networks*. Springer, New York, pp. 217–229.

Newman, G., Wiggins, A., Crall, A., Graham, E., Newman, S. & Crowston, K. 2012 The future of citizen science: emerging technologies and shifting paradigms. *Front. Ecol. Environ.* **10** (6), 298–304.

Oliveira, L. M. & Rodrigues, J. J. 2011 Wireless sensor networks: a survey on environmental monitoring. *J. Commun.* **6** (2), 143–151.

Pearce, J. M. 2012 Building research equipment with free, open-source hardware. *Science* **337** (6100), 1303–1304.

Pourmirza, Z. & Brooke, J. M. 2013 The wireless sensor network and local computational unit in the neighbourhood area network of the smart grid. In: *Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS 2013)*, SCITEPRESS Science and Technology Publications, pp. 84–88.

QGIS 2014 Welcome to the QGIS project! www.qgis.org/ (accessed 15 August 2014).

Quinn, N. W., Ortega, R., Rahilly, P. J. & Royer, C. W. 2010 Use of environmental sensors and sensor networks to develop water and salinity budgets for seasonal wetland real-time water quality management. *Environ. Model. Softw.* **25** (9), 1045–1058.

Sadler, J. M. 2014 jsadler2/arduino_to_hsl. https://github.com/jsadler2/arduino_to_hsl/releases/1.0 (accessed 19 November 2014).

Stroud Research Center 2014 StroudCenter/Arduino-SDI-12. https://github.com/StroudCenter/Arduino-SDI-12 (accessed 15 June 15 2014).

Taylor, P., Cox, S., Walker, D., Valentine, D. & SheAhan, P. 2014 WaterML 2.0: development of an open standard for hydrological time-series data exchange. *J. Hydroinform.* **16** (2), 425–446.

Tenopir, C., Allard, S., Douglass, K. L., Aydinoglu, A. U., Wu, L., Read, E., Manoff, M. & Frame, M. 2011 Data sharing by scientists: practices and perceptions. *PLoS One* **6** (6).

Thomson, P., Hope, R. & Foster, T. 2012 GSM-enabled remote monitoring of rural handpumps: a proof-of-concept study. *J. Hydroinform.* **14** (4), 829–839.

Trevathan, J., Johnstone, R., Chiffings, T., Atkinson, I., Bergmann, N., Read, W., Theiss, S., Myers, T. & Stevens, T. 2012 SEMAT – the next generation of inexpensive marine environmental monitoring and measurement systems. *Sensors* **12** (7), 9711–9748.

US Geological Survey 2014 USGS Surface Water Information. In: USGS. http://streamstatsags.cr.usgs.gov/ThreatenedGages/ThreatenedGages.html (accessed 1 August 2014).

van de Giesen, N., Hut, R. & Selker, J. 2014 The Trans-African hydro-meteorological observatory (TAHMO). Wiley interdisciplinary reviews. *Water* **1** (4), 341–348.

Wickert, A. D. 2014 The ALog: inexpensive, open-source, automated data collection in the field. *Bull. Ecol. Soc. Am.* **95** (2), 166–176.

Younis, M. & Akkaya, K. 2008 Strategies and techniques for node placement in wireless sensor networks: a survey. *Ad Hoc Netw.* **6** (4), 621–655.

Zaslavsky, A., Perera, C. & Georgakopoulos, D. 2013 Sensing as a service and big data. arXiv preprint arXiv:13010159.

First received 27 August 2014; accepted in revised form 5 February 2015. Available online 27 February 2015