International Conference on Hydroinformatics

2014

Hydrometeorological Data Collection, Publication And Analysis Using Open-Source Hardware And Software

Paul Celicourt

Michael Piasecki

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

HYDROMETEOROLOGICAL DATA COLLECTION, PUBLICATION AND ANALYSIS USING OPEN-SOURCE HARDWARE AND SOFTWARE

PAUL CELICOURT (1), MICHAEL PIASECKI (1)
(1): Department of Civil Engineering, The City College of the City University of New York, USA

ABSTRACT

Hydrometeorological data is exiguous in developing countries due to the exorbitant price of actual weather monitoring systems, data acquisition and communication infrastructure produced mostly in developed countries. Another impediment to large scale data collection in developing countries is the lack of or presence of limited local resources both pecuniary and human to ensure the maintenance of station installations. Despite significant advances in sensor network technologies which gave birth to open hardware and software, low-cost (less than $50) and low-power (in the order of a few microhorsepower) sensor platforms in the last two decades, sensors and sensor network deployment remains a labor-intensive, time consuming, cumbersome, and thus expensive task. Hence, there is the need for an affordable, simple to deploy, scalable and self-organizing end-to-end (from sensor to data publication) system suitable for deployment in such countries. In this paper, we present the design of an envisioned end-to-end system prototype which will consist of a very few sensor nodes controlled by the Arduino micro-controller with a complete set of low-cost sensors attached to measure variables relevant to hydrological processes, a central node which gathers data from each leaf node via radio frequency devices, data transmission from the central node to a webserver via a GSM network, and a back-end data management system hosted by the micro-computer named Raspberry Pi.

INTRODUCTION

Environmental monitoring remains the only path to detect trends in environmental parameters, derive knowledge from the monitoring process and build the basis for formulation of science-based decisions making regarding environmental management. Understanding the characteristics of our physical environment composed of the atmosphere, biosphere, cryosphere, lithosphere and hydrosphere is crucial in providing timely and effective information that allows individuals exposed to hazard to take action to avoid or reduce their risk (early warning) as well as implement mitigative effects should any unwanted environmental impacts or extreme events loom in the future and ensure long-term sustainability (Tegler et al., 2001[18]; Quansah et al., 2010[14]; UNISDR, 2009[19]).

Despite significant advances in sensor network technologies which that produced open hardware and software, low-cost (less than $50) and low-power (in the order of a few microhorsepower) sensor platforms in the last two decades, sensors and sensor network

deployment remains a labor-intensive, time consuming, thus cumbersome and expensive task The cost of one station, depending on the application and the precision of the sensors and specifications of the sensor platform, ranges from $1,000 up to $86,000 (Cox and Perez, 2006[7]; Rao et al., 2013[15]; Barrenetxea et al., 2008a[2]; 2008b[3]; 2008c[4]; Zennaro al., 2009[21]; Celebi, 2002[6]; Hirafuji et al., 2011[9]).
It is no surprise therefore that environmental data collection in developing countries where technical infrastructure, expertise, and resources are scant poses a challenge. They also highlight the need for an affordable, simple to deploy, scalable and self-organizing end-to-end (from sensor to data publication) system suitable for deployment in such countries. In addition, most sensor network deployment projects do not fully capture the full lifecycle of sensors data overlooking the data management side of the process which is a critical step. In this paper, we present the envisioned design of the HydroQEDS (**Q**uickly-and-**E**asily-**D**eployable **S**ystem to enable inexpensive **Hydro**meteorological Sensor Network and Cyberinfrastructure) prototype, an end-to-end (from sensor to data publication) system that is a) affordable, b) simple to deploy, c) scalable, d) self-organizing, and thus suitable for deployment in developing countries.

OBJECTIVE

Our objective is to develop a prototype of a nearly plug-and-play hydro-meteorological data collection system projected to cost less than 250 USD including all components. The system will be able to collect: solar radiation, air temperature, precipitation, wind direction and velocity, and we also hope to add a soil moisture sensor to the collection. The envisioned system assumes that users do not possess any computer programming knowledge except a few basic computer operating skills. Therefore, it must be smart enough to be able to operate as a nearly "plug-and-play everything" system where the base station (sink) will be capable of sensing and sending programs (according to the node programming requirement) to any new node that is located in its broadcast range in a dense deployment scheme. However, in a disperse deployment scheme, the end-nodes (leaves) will be able to contact the sink to obtain programming code or the other way around whichever is best. This latter design is relevant to support the Trans-African Hydro-Meteorological Observatory (TAHMO, see http://tahmo.info for more information) initiative aiming at bringing an affordable high density observation network (20,000 weather stations at a resolution of about 100,000 ft) to the African continent.

METHODOLOGY

A survey of popular platforms and programming language used in Wireless Sensor Network (WSN) with various applications including environmental monitoring demonstrated that Arduino is the best candidate to develop a low-cost sensor node. Therefore, it will constitute the backbone of the sensor nodes in addition to Xbee modules (node communication), Raspberry Pi (for compute power, the "smart side") and sets of ready-made sensors to collect variables relevant to study hydro-meteorological processes, all of which will be mounted on a lightweight and simple to deploy platform.

Sensor Node Physical Design

Key to the development is the acquisition of cheap sensors that are commercially available, i.e. are SOTS and the supporting API libraries to allow the Arduino to pair with these devices. We foresee that a new board and adaptors specific to this application might be needed in order to achieve simplicity on the plug and play objective. We intend also to develop cheap sensors and the deployment platform using a 3D printer to pair with the customized Arduino-based sensor for our prototype development. The three different sensor node designs/architectures envisioned are:

a. A stand-alone Arduino, an Xbee or GSM module and a sensor board,
b. An Arduino controlled by Raspberry Pi through serial communication to make use of the full capability of the Python programming language, an Xbee or GSM module and a sensor board,
c. A Raspberry Pi as a sensor platform to make use of the full capability of Python, an Xbee or GSM Module and sensor board such as PiFace.

The sensor node is expected to have a very small footprint as all the equipment are expected to be mounted on a small optimized platform attached to a pole with a small solar panel for easy deployment.

Sensor Node Programming via the Base Station

In order to provide a customizable compute platform we will use the Python lightweight interpreter PyMite designed specifically to run Python codes on microcontrollers. Python is a high-level and powerful open source programming language with extensive capabilities such as scientific computation, image processing, XML processing and serial communication. In addition, also it has a large community of developers behind it that is continuously adding more capabilities to the language either through the development of new libraries or by the improvement of existing ones. Python offers, through supporting libraries, the capability to fully access the General Purpose Input Output (GPIO) features by communicating with the GPIO pins of a device and can connect to serial devices as well. Interestingly, despite its utility and power Python has been used scarcely in sensor network programming where C remains the de facto programming language. Due to the scarce memory resources of the Arduino, there is also the possibility of installing the PyXis Operating System (OS) on the Arduino to allow for the development of more robust programming software to smarten the nodes as this OS offers the opportunity to run a program from a USB drive or SD card. In the case where an acceptable degree of smartness cannot be achieved with PyMite due to its reduced scope when compared to the full Python package, an alternative means would be to use a Raspberry Pi that can accommodate the full Python language. This becomes a resources issue however, in which a balance must be struck between performance and price.

Moreover, with the intention of smartening the nodes and depending of the final system architecture adopted, we can make use of a variety of Python libraries such as pySerial to enable Python codes to communicate over serial ports, platforms to access information to identify underlying platform, PySensors to get information about sensors connected to the platform, Xbee to easily access advanced features of XBee devices, Xebec to communicating with a GSM Modem or mobile phone, etc. These libraries will hopefully facilitate the modification of radio frequency devices settings to enable the establishment of the network and the transmission of programming codes generated to the desired node from a Graphical User Interface (GUI) or a phone app. As the system will be completely abstract to users, we intend to make the application automatically create and feed a database with the characteristics of each node.

Sensor Node Programming and Data Retrieval through Phone Application

Android, the dominant smartphone operating system developed by google, is the primary target for a phone app that we will seek to develop to both remotely program the nodes and retrieve the sensors data collected and stored in the database system. Here too, we will make use of

Python to develop this app. Two of the best candidates to develop the Android application in Python are: PySide-based QML GUIs using the Necessitas Qt port and the Python for Android/ Scripting Layer for Android (Py4A/SL4A)-based approach. We are favoring the SL4A-based approach because it is open source based and offers the most flexible and features rich technology. In addition, it allows access to the Android OS Application Programming Interface (API) which makes it possible to use many features that allow developers to add capabilities to their application and also develop sophisticated application. For example, privileged resources such as hardware devices (CPU, memory), kernel data, or data in another process space can be accessed.

Back-end Data Management

For our data management task we will make use of open source software that has been specifically developed to handle sensor data, i.e. point oriented time series streams. For HydroQEDS we will make use of the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)'s Hydrologic Information System (HIS) which is gaining a wide national and international recognition as a suitable collection of software installations that handle storing, discovering, publishing and analyzing water-related data. The HIS project (Taborton et al., 2009[17]; Taborton et al., 2011[16]) aims at developing standard infrastructures to support the sharing of distributed hydrologic data through Web services and tools for data discovery, access, publication and analysis. The central piece of the HIS is the HydroServer (Horsburgh et al., 2010[12]), which is a server hosting an array of databases, Web services, and software that allows data owners to store data in the Observations Data Model (Horsburgh et al., 2008[11]); publish data through HydroServer; access data through the internet-based WaterOneFlow (Valentine et al., 2007[20]; Beran et al; 2009[5]) web services using the XML-based language WaterML; discover data through a metadata catalog; and analyze space-time hydrologic datasets and other environmental observation data through Hydrodesktop (Ames et al., 2012[1]).

One of the drawbacks of this system is that it requires the installation of several Microsoft products, such as Windows 2008 Server, SQL Server, and ASP.NET. Our computer resources, however, are Linux-based and also are too small to meet the minimum hardware requirements to run HydroServer. Therefore, as an alternative we will use the MySQL or PostgreSQL version of the ODM and the WOFpy (WaterOneFlow Web Services in Python) developed by the Texas Water Development Board for 'Water Data for Texas' which can produce WaterML web services from a variety of back-end database installations such as SQLite, MySQL, and PostgreSQL. As these code stacks are open source, we can also modify the codes of other components (such as the CUAHSI Time Series Analyst) to use them in this setup if necessary.

Using the PostgreSQL version of the ODM would be beneficial for the following reasons: PostgreSQL is more powerful than MySQL in the sense that it is more reliable, faster than MySQL when performing complex operations due to its performance features including cache management and indexing. We can combine the PostgreSQL spatial capability (e.g. load the spatial table behind a watershed shapefile into the database and display it online as a basemap using OGC Web Feature Service) with Python statistics and analytics tools (Scipy, Numpy, Pandas) to produce an appealing online visualization tool. This tool will be to show the spatial distribution of variables across an entire watershed as a time variant layer on top of a basemap in contrast to SensorMap (Nath et al., 2007[13], Grosky et al., 2007[8]) which shows the spatial distribution over a rectangular area only and also cannot handle time variant information.

.

Evaluation of the Data Acquisition System and Real-World Deployment

To evaluate the performance and effectiveness of the system, we will seek to deploy this system in Haiti where we plan to partner with two governmental agencies (ONEV and the Direction of Water Resources of the Environment Ministry of Haiti) and a regional organization called DATIP who have both declared their willingness to provide access to test plots and sites for an initial installation of several networks. These will be in support and in conjunction with an existing hydrometeorological sensor network that we have been operating since March 2012. Requirements will be: not too far apart as the regular XBee has a max outdoor range of about 120m (the Pro module can reach up to 1500m however), reasonable accessibility, safety against vandalism, theft, and accidental damage (animals), and line of sight. To this end we will work with DATIP and ONEV personal to also provide feedback on utility and ease of installation. They will also provide mid- to long term feedback once the units have been installed so we can collect data on reliability (thus maintenance interval requirements) and longevity especially in view of extreme weather such as hurricanes.

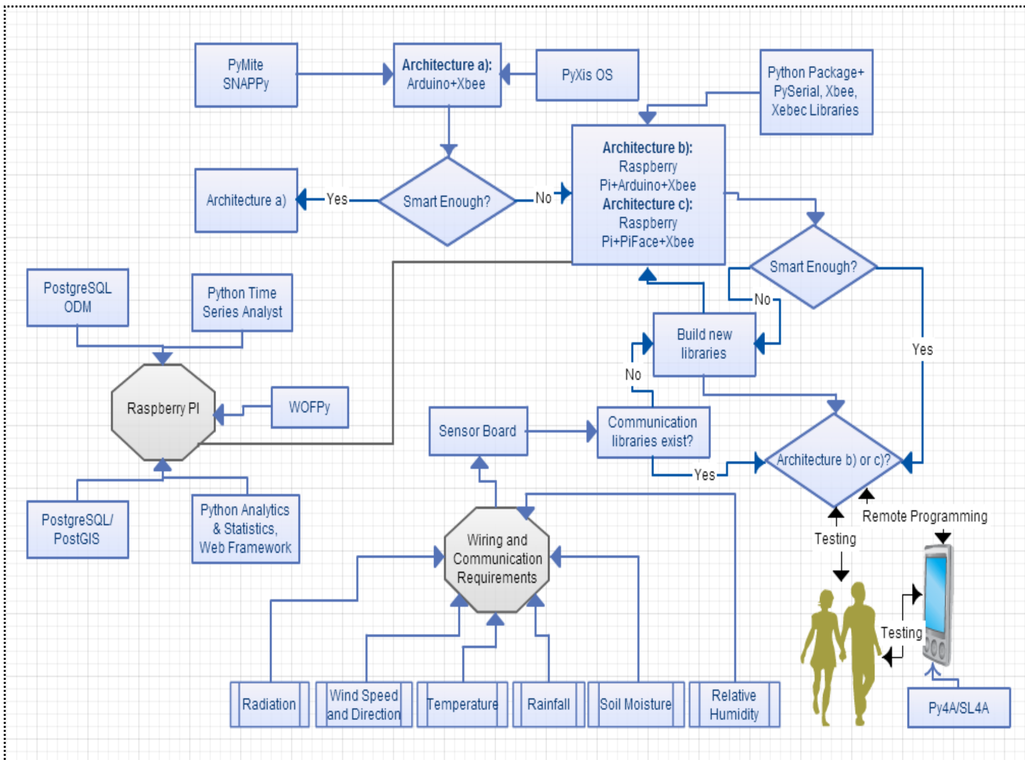

Figure 1: Summary of the Development Strategy

Summary

Developing a low cost, programming-free, smart, and easy-to-deploy hydroclimatological sensor node requires the inclusion of a number of software and hardware aspects. We propose to use an Arduino and Raspberry Pi unit, the former for data logging the latter equipped with on-board capabilities to handle some of the data management on the probe, as well as managing the sensor behavior within a network of nodes. We want to make this node as easy to deploy as possible so that potential users and technicians in the field require little knowledge and training on the inner workings of the node and thus making maintenance a low level effort. The node will contain cheap of-the-shelf sensors in addition to cost effective housings with little material used to construct the backbone of the node, and also be based on very compact electronic components such as the Arduino and the Raspberry Pi units. All programming and software stacks required for logging, data transfer, data management, and data retrieval are either open source or are based on internal acknowledged standards for example ISO, OGC, and W3C.

We believe that this system will have a general appeal to a larger variety of sensing applications, and in particular, will lower the entry barrier for hydrometeorological data acquisition both in developed and developing countries as users do not need any background in computer science or electrical engineering. As future work, we intend to overcome some of the limitations by expanding this effort to other well accepted sensor platforms such as the Mica family among others, work on the security issue that WSNs might face using Python, and scale up and extend this setup to a multi-hop configuration.

REFERENCES

[1] *Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T. & Valentine, D., "Hydrodesktop: web services-based software for hydrologic data discovery, download, visualization, and analysis.", Environmental Modelling & Software, 37(2012), pp. 146--156.*

[2] *Barrenetxea, G., Ingelrest, F., Schaefer, G. & Vetterli, M., "Wireless sensor networks for environmental monitoring: the sensorscope experience.", Proc. 2008 International Zurich Seminar on Communications, Zurich, Switzerland, (2008a), pp. 98-101.*

[3] *Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O. & Parlange, M. "Sensorscope: out-of-the-box environmental monitoring.", Proceedings 2008 International Conference on Information Processing in Sensor Networks, St Louis, Missouri, USA, (2008b), pp. 332--343.*

[4] *Barrenetxea, G., Ingelrest, F., Lu, Y. M. & Vetterli, M., "Assessing the challenges of environmental signal processing through the sensorscope project.", Proceedings 2008 IEEE international conference on acoustics, speech, and signal processing, Las Vegas, Nevada, USA, (2008c), pp. 5149--5152.*

[5] *Beran, B., Goodall, J., Valentine, D., Zaslavsky, I. & Piasecki, M., "Standardizing access to hydrologic data repositories through web services." Proceedings 2009 International Conference on Advanced Geographic Information Systems & Web Services, Washington, DC, USA, (2009), pp. 64--67.*

[6] *Celebi, M., "Seismic instrumentation of buildings (with emphasis on federal buildings).", (2002), Technical Report No. 0-7460-68170, United States Geological Survey, Menlo Park, CA.*

[7] *Cox, G. & Perez C., "Weather Station Review." (2006). [online] Retrieved from: http://www.not.iac.es/instruments/development/weather_station.pdf. [Accessed: 12 Feb 2014].*

[8] *Grosky, W. I., Kansal, A., Nath, S., Liu, J. & Zhao, F., "Senseweb: an infrastructure for shared sensing.", IEEE Multimedia, Vol. 14 (2007), pp. 8--13.*

[9] *Hirafuji, M., Yoichi, H., Kiura, T., Matsumoto, K., Fukatsu, T., Tanaka, K., Shibuya, Y., Itoh, A., Nesumi, H., Hoshi, N. & Others., "Creating high-performance/low-cost ambient sensor cloud system using openfs (open field server) for high-throughput phenotyping. ",Proceedings of the SICE Annual Conference, Tokyo, (2011), pp. 2090--2092.*

[10] *Horsburgh, J. S., Stevens, D. K. and Goodall, J., "Time Series Analyst – An Internet Based Application for Analyzing Environmental Time Series.", Presented at the AWRA Spring Specialty Conference on GIS and Water Resources IV, Houston, Texas, USA, (2006).*

[11] *Horsburgh, J. S., Tarboton, D. G., Maidment, D. R. & Zaslavsky, I., "A relational model for environmental and water resources data.", Water Resources Research, Vol. 44 (2008).*

[12] *Horsburgh, J. S., Tarboton, D. G., Schreuders, K. A., Maidment, D. R., Zaslavsky, I. & Valentine, D., "Hydroserver: a platform for publishing space-time hydrologic datasets.", Proceedings 2010 American Water Resources Association Spring Specialty Conference Geographic Information Systems (GIS) and Water Resources VI, Orlando, Florida, USA, (2010).*

[13] *Nath, S., Liu, J. & Zhao, F.. "Sensormap for wide-area sensor webs.", Computer, 40 (2007), pp. 90--93.*

[14] *Quansah, J. E., Engel, B. & Rochon, G. L., "Early warning systems: a review.", Journal Of Terrestrial Observation, 2 (2010), p. 5.*

[15] *Rao, A. S., Marshall, S., Gubbi, J., Palaniswami, M., Sinnott, R. & Pettigrovet, V., "Design of low-cost autonomous water quality monitoring system.", Proceedings 2013 International Conference on Advances in Computing, Communications and Informatics, Mysore, India, (2013), pp. 14--19.*

[16] *Tarboton, D. G., Maidment, D., Zaslavsky, I., Ames, D., Goodall, J., Hooper, R. P., Horsburgh, J., Valentine, D., Whiteaker, T., and Schreuders, K., "Data interoperability in the hydrologic sciences, the CUAHSI hydrologic information system.", Proceedings of the Environmental Information Management Conference, Santa Barbara, CA, USA, (2011). pages 132–137.*

[17] *Tarboton, D., Horsburgh, J., Maidment, D., Whiteaker, T., Zaslavsky, I., Piasecki, M., Goodall, J., Valentine, D. & Whitenack, T., "Development of a community hydrologic information system.", 18th World IMACS / MODSIM Congress, Cairns, Australia, (2009). pp. 988--994.*

[18] *Tegler, B., Sharp, M. & Johnson, M. A., "Ecological monitoring and assessment network's proposed core monitoring variables: an early warning of environmental change.", Environmental Monitoring And Assessment, 67 (2001), pp. 29--55.*

[19] *UNISDR, "Terminology on disaster risk reduction.", United Nations, Geneva, (2009).*

[20] *Valentine, D., Zaslavsky, I., Whitenack, T. & Maidment, D. R., "Design and implementation of cuahsi waterml and wateroneflow web services.", American Geophysical Union, Fall Meeting, (2007), pp. 5--3.*

[21] *Zennaro, M., Floros, A., Dogan, G., Sun, T., Cao, Z., Huang, C., Bahader, M., Ntareme, H. & Bagula, A., "On the design of a water quality wireless sensor network (wqwsn): an application to water quality monitoring in malawi.", Proceedings International Conference on Parallel Processing Workshops, (2009), pp. 330--336.*