

# Rapid Deployment of Internet-Connected Environmental Monitoring Devices

**Daniel K. Fisher**

USDA Agricultural Research Service, Crop Production Systems Research Unit, Stoneville, Mississippi, USA  
Email: [daniel.fisher@ars.usda.gov](mailto:daniel.fisher@ars.usda.gov)

Received 8 August 2014; revised 9 September 2014; accepted 8 October 2014

Copyright © 2014 by author and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Advances in electronic sensing and monitoring systems and the growth of the communications infrastructure have enabled users to gain immediate access to information and interaction with physical devices. To facilitate the uploading, viewing, and sharing of data via the internet, while avoiding the complexities and expense of creating personal web servers, a number of service providers have created websites offering free data hosting and viewing capabilities. Simple setup and configuration routines and available application programming interfaces allow users to quickly and easily interface sensing and monitoring devices to the internet. To demonstrate the ease and feasibility of deploying internet-connected devices, an urban landscape environmental monitoring system, consisting of two wireless field sensing systems and a wireless central receiver, was developed. The field-data sensing units consisted of Arduino microcontroller platforms, sensors, and Xbee radio modems, while the receiver consisted of an Arduino microcontroller, Xbee radio, and Ethernet module connected to an internet modem. The receiver collected and stored data from the wireless transmitters, and uploaded the data to the Xively Cloud Services data hosting and sharing website. Real-time and archived sensor data were then made available for public viewing via the internet and a web browser. Data-sharing services such as Xively provide rapid and convenient means of developing internet-accessible data-collection and viewing applications, enabling enhanced access to the Internet of Things.

## Keywords

Arduino, Xively, Wireless, Sensors, Microcontroller

---

## 1. Introduction

Ease of use, capability, and affordability of sensing and monitoring instrumentation have increased greatly due to rapid advances in electronics and communications technologies. Microcontrollers, microprocessors, and sol-

id-state sensors are commonplace, and are found in most modern household appliances (coffee makers, washing machines, ovens), entertainment devices (televisions, radios, game machines), computer equipment (desktop, notebook, tablet computers), and cellular devices (smartphones). These same electronic components are increasingly being incorporated into environmental sensing and monitoring applications in order to study physical processes, monitor activities, and improve production operations.

Advances in communications capabilities have been realized on all scales, from local (Bluetooth, radio modems), to regional (cellular), to global (internet). The growth of the communications infrastructure has resulted in the ability of people to be connected in real time over any distance at any time. This immediate access is now driving a desire for, and an expectation of, real-time access to information and interaction with physical devices, leading to the development of the so-called Internet of Things, in which sensing and monitoring devices are connected and accessible via the internet.

The technology for interfacing sensing and monitoring devices with the internet communications infrastructure is mature and available to all users. Providing a method for uploading and accessing information via the internet usually involves creating a web server, and connecting this server to the internet. Web server software must be installed and configured on a computer, a website and web pages must be created using HTML code, a router and its firewall must be configured to allow incoming requests to access the website, and a website domain and static IP address must be obtained to allow users to find and access the website. Data collection devices must then be programmed and interfaced to provide data to the web server. The lack of knowledge, computer skills, and ability among many users to realize these tasks, however, can be an obstacle.

To facilitate the uploading and sharing of data, while avoiding the complexities and expense of creating personal web servers, a number of service providers have created websites offering data hosting and viewing capabilities. While often intended for commercial use by industry for industrial monitoring and Machine to Machine (M2M) communications [1], several services offer both commercial (paid) and limited (free) access. Services such as Carriots [2], ThingSpeak [3], and Xively [4] offer free hosting of sensor data, and provide Application Programming Interfaces (APIs) to enable programming of devices and transfer of data to their respective websites. APIs, consisting of programming libraries and software routines, for common data-collection devices, such as Arduino microcontrollers, Raspberry Pi computers, and Android devices, are freely available and allow developers to rapidly configure and deploy internet-enabled sensing and monitoring systems. A number of developers have begun using these free services to simplify connection of devices to the internet and share data related to environmental monitoring [5] [6], parking lot management [7], electrical power and energy consumption [8] [9], education [10], and home/office automation [11].

## Objectives

The objective of this work was to demonstrate the ease of development and deployment of an internet-connected monitoring and datalogging system using the Xively data-sharing service.

## 2. Materials and Methods

An internet-connected monitoring and datalogging system was developed to monitor environmental parameters in an urban landscape environment and enable internet-access of the data collected. The monitoring system consisted of two main parts: 1) the Xively-based internet-accessible component (software); and 2) the data collection and upload component (hardware).

### 2.1. Xively Internet Component (Software)

Xively Cloud Services [4] provide a platform for uploading, viewing, and sharing data via the internet. Developers are given free access to Xively services in order to connect devices and create datastreams and applications. APIs, programming libraries, and example applications are provided to simplify programming and enable rapid development and interfacing of devices with the Xively website. To enable user data to be uploaded and viewed, a page is created on the Xively website and the user enters information about the incoming datastream, or feed, and data values, or channels. The user then configures the data-collection device using the feed and channel information to send sensor data to the user's Xively page.

### 2.1.1. Creation and Configuration of a Xively Account

To use the Xively Cloud Services, the developer first accesses the Xively website (<http://www.xively.com>) using a web browser and creates a developer account by providing a unique account name and password. A new web page is generated, along with a unique identification number (feed ID) for accessing the web page. Another identification number, an API key, is also generated which identifies the incoming datastreams, and is used by the developer when programming the data-collection device to post data to the proper web page.

The developer then configures the web page to identify the incoming datastreams, or feeds. Feeds can consist of multiple parameters, or channels, and each individual channel is detailed in terms of channel name, physical description, and unit of measure. Additional details, including geographic location, data-collection system information, and developer contact information can also be included. The web page can be configured as publicly available, allowing public viewing of the data by others, or private, allowing access only by the developer. Data on the web page can then be viewed, if publicly available, via web browser by appending the feed ID to the Xivelyfeeds web address (<https://xively.com/feeds/1947445049>, for example). The Xively data feed web page created during this study is shown in **Figure 1**.

### 2.1.2. Integrating the Xively Service with the Monitoring Device

Integrating the monitoring device and the Xively service is accomplished by programming the developer's data-stream information into the monitoring device. Information including the feed ID, API key, and data stream channels is incorporated into the monitoring device programming. Programming libraries have been written for a variety of common programming languages, and can be downloaded and installed from the Xively website. The programming libraries offer convenient access to required communications and data-transfer operations, simplifying the programming for developers. With minimal additional coding, the developer can rapidly and easily upload data from the monitoring device to the Xively website.

## 2.2. Monitoring Device Hardware Components (Hardware)

The internet-connected data logging instrument system demonstrated in this study was developed and deployed to monitor environmental conditions in an urban landscape setting. The system consisted of two outdoor wireless field-data transmitters and sensors, and an indoor Ethernet-enabled wireless receiver. The field transmitters

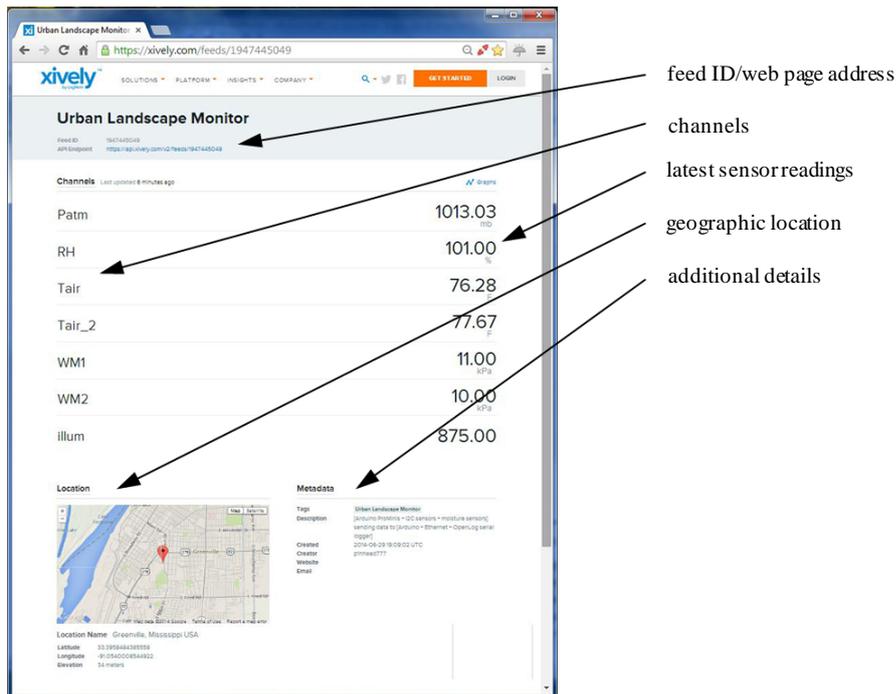


Figure 1. Xively data feed web page.

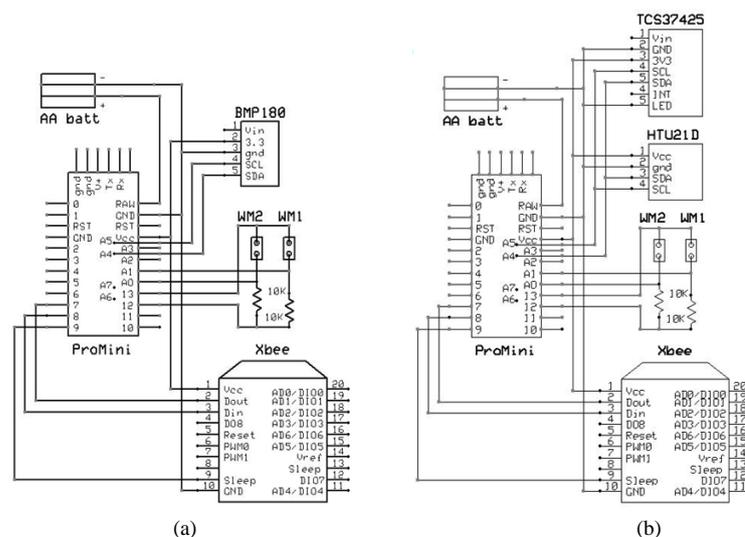
monitored various sensors, and transmitted sensor data to the receiver, which stored sensor readings locally and uploaded data to the Xively website. Design and fabrication of the monitoring system are detailed in the following sections, with a summary of the components and their costs presented in **Table 1**.

### 2.2.1. Field Transmitters

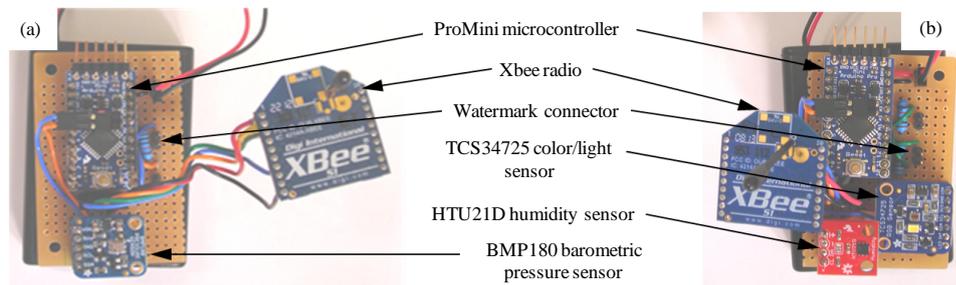
The main components of the two field transmitter units consisted of Arduino-compatible microcontroller development boards, soil-moisture sensors and measurement circuitry, and radio modems. Three digital sensors were also connected, with one transmitter unit containing a barometric pressure/air temperature sensor, and the second transmitter containing a relative humidity/air temperature sensor and a light sensor. The schematics for the two units are shown in **Figure 2**, with a photo of the devices and components shown in **Figure 3**.

**Table 1.** Summary and approximate costs of hardware components.

Assembly	Component	Source	Cost (US\$)
Transmitters	ProMini microcontroller board	Sparkfun electronics	10
	BMP180 barometric pressure breakout board	Adafruit industries	10
	HTU21D relative humidity breakout board	Sparkfun electronics	15
	TCS34725 color-to-light breakout board	Adafruit industries	8
	Watermark 200SS soil-moisture sensor	Irrometer company	30
	miscellaneous		8
Receiver	Diavolino microcontroller board	Evil mad science	15
	Ethernet shield	ITEAD studios	20
	OpenLog serial datalogger	Sparkfun electronics	25
	DS1307 real-time clock	Maxim integrated	2
	Prototype shield	Sparkfun electronics	10
	Xbee adapter	Adafruit industries	10
	miscellaneous		7
	Xbee series 1 radio modems	Digi international	20
Total cost	Transmitter #1		108
	Transmitter #2		121
	Receiver		109



**Figure 2.** Schematics for (a) Field transmitter #1, and (b) Field transmitter #2.



**Figure 3.** Components of (a) Field transmitter #1 and (b) Field transmitter #2.

The ProMini microcontroller development board (Sparkfun Electronics, Niwot, Colorado USA) was selected, and consisted of an ATmega328 8-bit microcontroller operating at 3.3 V and a clock speed of 8 MHz. The ProMini microcontroller board featured 14 digital input/output pins and 8 digital/analog pins with 10-bit resolution analog-to-digital converter (ADC) capability. The board contained an on-board voltage regulator to convert the nominal 4.5-V supply from 3 AA-size batteries to the 3.3-V level required by the microcontroller.

Two soil-moisture sensors, Watermark model 200SS (Irrrometer Company, Inc., Riverside, California USA) were read via an alternating-current half-bridge circuit to prevent polarization and damage to the sensors. The circuit, shown in the schematics in [Figure 2](#) and described in more detail [12], consisted of a half-bridge (voltage divider) for each sensor, with a fixed, 10-k $\Omega$  (kiloOhm) resistor as one leg and a Watermark sensor as the other leg. The circuit was energized at one polarity by setting two digital pins, 3.3 V on pin 13 and 0 V (ground) on pin 12. The voltage at the center of the half-bridge was measured with one ADC, and the resistance of the Watermark sensor was calculated using the standard voltage-divider equation. The polarity of the circuit was then reversed, setting 0 V on pin 13 and 3.3 V on pin 12, the center voltage was measured, and the Watermark resistance calculated again. The two resistances were then averaged, and the average resistance value was input to a sensor calibration equation to determine the soil-water potential, in units of kiloPascals (kPa).

The three digital sensors operated at 3.3 V and communicated with the microcontroller via the I2C two-wire protocol ([www.i2c-bus.org](http://www.i2c-bus.org)), a standard method designed to allow multiple digital devices to communicate using only two microcontroller pins. Because the digital sensing elements were very small in physical size, with the largest measuring less than 4 mm  $\times$  4 mm, sensors installed on breakout boards were used. The breakout boards included the sensing elements and supporting circuitry, such as voltage regulators and pull-up resistors, and enabled convenient access to all necessary sensor pins for rapid prototyping and ease of use.

Barometric pressure and air temperature were measured with a BMP180 Digital Pressure Sensor (Bosch Sensortec GmbH, Kusterdingen, Germany) installed on a BMP180 Breakout Board (Adafruit Industries, New York, New York USA). Specifications of the BMP180 sensor included an accuracy of pressure measurements of  $\pm 0.12$  hPa over a range of 900 - 1050 hPa (1 hPa = 1 mb (millibar)), and a measurement resolution of 0.01 hPa. Accuracy of temperature measurements was  $\pm 0.5^\circ\text{C}$  at  $25^\circ\text{C}$ , with a measurement resolution of  $0.1^\circ\text{C}$ .

Relative humidity and air temperature were measured with an HTU21D Digital Relative Humidity Sensor (Measurement Specialties, Inc., Toulouse, France) mounted on an HTU21D Breakout Board (Sparkfun Electronics). Specifications of the HTU21D sensor included an accuracy of relative humidity measurements of  $\pm 2\%$  at  $25^\circ\text{C}$ , and a measurement resolution of 0.04%. Accuracy of temperature measurements was  $\pm 0.3^\circ\text{C}$  at  $25^\circ\text{C}$ , with a measurement resolution of  $0.01^\circ\text{C}$ .

Light levels were measured with a TCS34725 Color-to-Light Digital Converter (ams AG, Unterpremstaetten, Austria) mounted on a TCS34725 Breakout Board (Adafruit Industries). The TCS34725 color sensor contained an array of photodiodes filtered to measure light levels in the red, green, and blue (RGB) wavelengths, and unfiltered to measure ambient (clear) light levels. A calibration equation was provided by the sensor manufacturer to convert RGB levels to luminance estimates.

Wireless data transmission was accomplished via an Xbee Series 1 radio modem (Digi International, Inc., Minnetonka, Minnesota USA). The radio operated at a 3.3 V level and a frequency of 2.4 GHz.

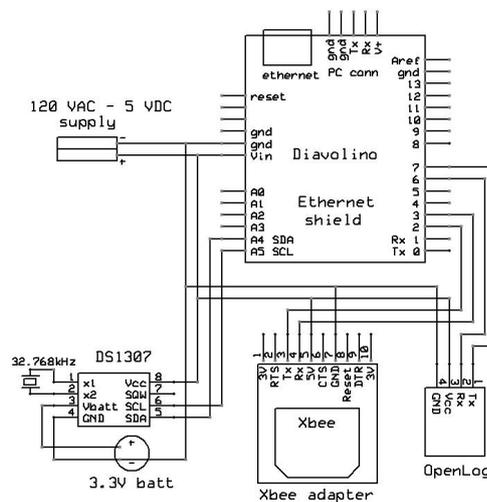
### 2.2.2. Ethernet-Connected Receiver

The receiver consisted of a microcontroller board, Ethernet module, real-time clock/calendar, Xbee radio modem, and serial datalogger board installed on a prototyping circuit board. The schematic for the receiver is shown

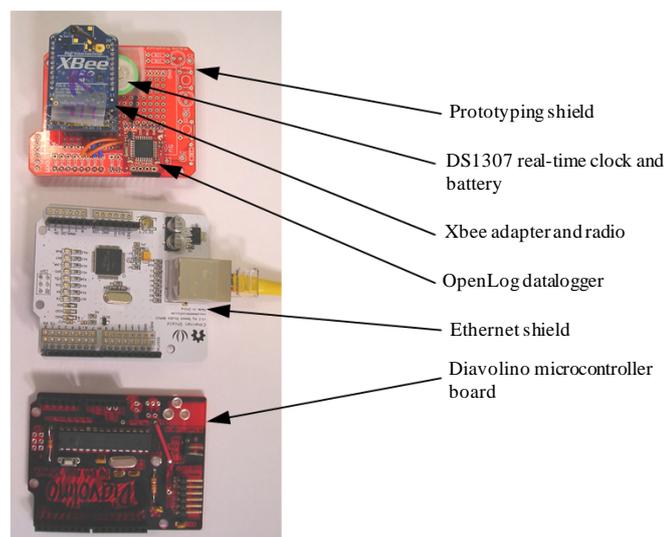
in **Figure 4**, with a photo of the components shown in **Figure 5**.

The microcontroller board consisted of a Diavolino Arduino-compatible microcontroller platform (Evil Mad Science LLC, Sunnyvale, California USA) with an ATmega328 microcontroller operating at a 5-V level and a speed of 16 MHz. An Ethernet Shield (ITEAD Studio, Shenzhen, China) add-on board was connected to the Diavolino via mating header pins on each board. A Prototyping Shield (Sparkfun Electronics) was used to support and connect the remaining circuit components to the microcontroller board. A DS1307 Real-Time Clock/Calendar (Maxim Integrated, San Jose, California USA), oscillator, and coin-cell backup battery provided time and date information. An OpenLog Serial Datalogger (Sparkfun Electronics) and microSD card provided backup data-storage capability. An Xbee radio was interfaced with the microcontroller via an Xbee Adapter Board (Adafruit Industries). The adapter board was required to connect the Xbee radio, which operated at 3.3 V, to the Diavolino microcontroller board, which operated at 5 V. The adapter board contained a 3.3-V voltage regulator and voltage-level circuitry to enable bidirectional logic-signal conversions between 5 V and 3.3 V for safe operation with a 5-V supply.

The Ethernet shield was connected to an Ethernet router, which was connected to an internet modem, via an Ethernet cable. The receiver was powered by an AC-to-DC adapter, which converted the 120-VAC mains voltage to a regulated 5-VDC supply.



**Figure 4.** Ethernet-enabled receiver schematic.



**Figure 5.** Ethernet-enabled receiver components.

### 2.2.3. Microcontroller Programming

The transmitter and receiver microcontrollers were programmed using the open-source Arduino IDE, version 1.0.5, downloaded from the Arduino project website ([www.arduino.cc](http://www.arduino.cc)) and installed on a Windows-based computer. The IDE consisted of a programming environment for developing microcontroller programs in a language based on C++, a compiler with built-in error checking, and an uploader for writing compiled programs to the microcontrollers. Microcontroller programs written for the receiver and transmitters in this study are freely available by contacting the author.

The receiver was programmed to collect, store, and upload sensor measurements to the Xively website at 15-minute intervals. The receiver was programmed to periodically read the current time from the real-time clock, and, if the time indicated the beginning of a 15-minute measurement period, turn on the Xbee radio. The radio remained on for a period of 60 seconds, and any radio transmissions received were buffered and stored temporarily by the microcontroller. After 60 seconds, the radio was turned off, the microcontroller read in all the data from the buffer, and stored the date, time, and sensor readings to the Open Logdatalogger's micro SD card. The sensor readings were then uploaded to the proper feed web page on the Xively website via the Ethernet Shield.

Each transmitter microcontroller was programmed to periodically make and transmit sensor measurements, then go into a low-power sleep mode to minimize current consumption and prolong battery life. The microcontroller woke at intervals of approximately 56 seconds and read each sensor. The Xbee radio was then turned on, data were transmitted, and the radio was turned off. Since the receiver radio was programmed to be active for only a 60-second interval each 15 minutes, sending data at sub-60-second intervals ensured that at least one transmission occurred during the receiver's active period. Following data transmission, the microcontroller returned the circuit to the low-power sleep mode.

The Xbee radios consisted of Series 1 radios which operated in a point-to-point configuration. While the firmware incorporated into the radios provided a variety of parameters for adjusting radio performance and operation, only three network and identification parameters needed to be configured for basic operation. Network parameters included the personal area network identifier (PANID), which gave a unique identifier to the network, and the channel (CHAN), or frequency, at which the radios operated. The radio on the receiver was given a unique name, or MY address, and each transmitter was then configured to transmit to the receiver's MY address. The radios were programmed using the radio manufacturer's computer software, X-CTU ([www.digi.com](http://www.digi.com)), and an Xbee Explorer board (Sparkfun Electronics), which allowed connection of the Xbee radios to a computer via USB port.

## 3. Field Trial

The two field transmitters with sensors were deployed in an urban residential landscape environment in Mississippi USA in 2014. Field transmitter #1 was located at the front of the residence, and two Watermark 200SS sensors were installed at a depth of 10 cm, one sensor under grass and one under ornamental plants. The electronic components (microcontroller, digital sensor, and Xbee radio) were housed inside a weatherproof enclosure installed amongst the ornamental plants. Field transmitter #2 was similarly installed at the back of the residence. The Ethernet-enabled receiver was located inside the residence, approximately 8 m from transmitter #1 and 15 m from transmitter #2, and interfaced with the residence's internet connection.

Samples of data collected by the transmitters and uploaded to the Xively website are shown in [Figure 6](#). Upon accessing a data feed's web page, the latest numerical data values are displayed for each channel. Archived data values during various time periods can also be displayed in graphical form by selecting the channel name. The default time period for each graph is 6 hours, and the time period can be changed to display data for periods ranging from 5 minutes to 3 months. Relative humidity (RH) data for a 1-day period, air temperature (Tair) data from transmitter #1 for a 6-hr period, and air temperature from transmitter #2 (Tair\_2) for a 7-day period are shown in [Figure 6](#).

## 4. Summary

The growth of the communications infrastructure and advances in electronic sensing and monitoring systems have enabled immediate access to information and interaction with physical devices via the internet. Technical complexities and expense of creating personal web servers to enable internet access, however, can limit the deployment of internet-connected devices. To facilitate the transmission, viewing, and sharing of data, a number

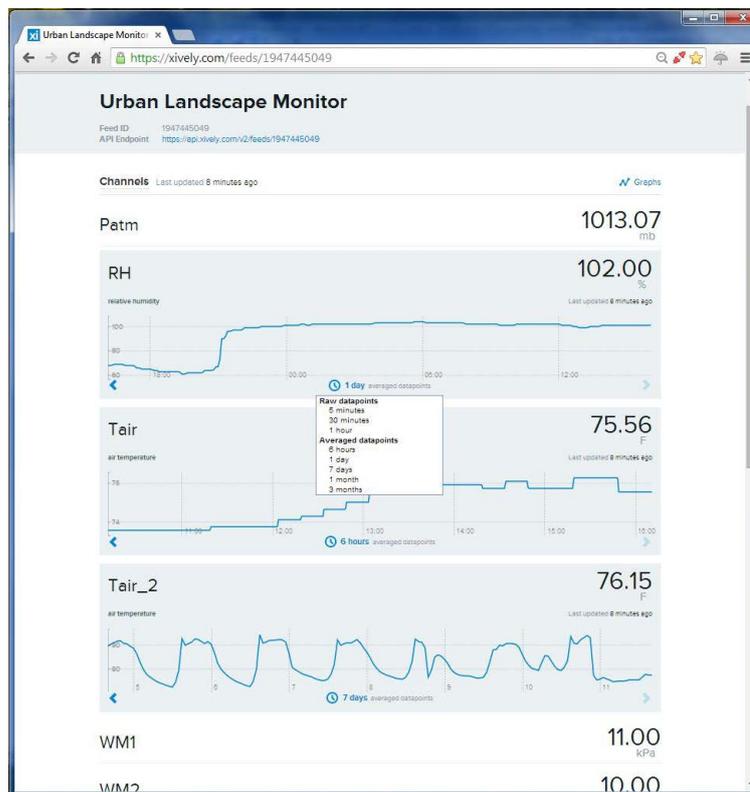


Figure 6. Time-series representations of sensor data.

of service providers have created websites offering free data hosting and viewing capabilities. Simple setup and configuration routines and available application programming interfaces allow users to quickly and easily interface sensing and monitoring devices to the internet.

To demonstrate the ease of deploying internet-enabled devices, an environmental monitoring system was developed which consisted of two field sensing and data-transmission instruments and a central receiver. The field-data sensing units consisted of Arduino microcontroller platforms, soil-moisture and environmental sensors, and Xbee radio modems. Each field-data unit made periodic sensor measurements, transmitted data via the radio modem, and went into a low-power sleep mode to conserve battery power. The receiver consisted of an Arduino microcontroller, serial datalogging module, Xbee radio, and Ethernet module connected to an internet modem. The receiver collected and stored data from the wireless transmitters, and uploaded the data to the Xively Cloud Services data hosting and sharing website. Real-time and archived sensor data were then made available for public viewing via the internet and a web browser pointed to the Xively website.

Data-sharing services such as Xively provide rapid and convenient means of developing internet-accessible data-collection and viewing applications. These services allow essentially any type of data-collection device to provide immediate access to the data via the internet with minimal additional hardware and programming requirements. Data can then be shared and viewed by all, enabling enhanced access to the Internet of Things.

## References

- [1] Balamuralidhara, P., Misra, P. and Pal, A. (2013) Software Platforms for Internet of Things and M2M. *Journal of the Indian Institute of Science*, **93**, 487-498.
- [2] Carriots (2014) Carriots—Internet of Things Platform. <http://www.carriots.com>
- [3] ThingSpeak (2014) Internet of Things—ThingSpeak. <http://www.thingspeak.com>
- [4] Xively (2014) Xively by LogMein—Business Solutions for the Internet of Things. <http://www.xively.com>
- [5] Snyder, E.G., Watkins, T.H., Solomon, P.A., Thoma, E.D., Williams, R.W., Hagler, G.S.W., Shelow, D., Hindin, D.A., Kilaru, V.J. and Preuss, P.W. (2013) The Changing Paradigm of Air Pollution Monitoring. *Environmental Science and*

- Technology*, **47**, 11369-11377. <http://dx.doi.org/10.1021/es4022602>
- [6] Peres, A., Miletto, E.M., Kapusta, S., Ojeda, T., Lacasse, A. and Gagnon, J. (2013) WAITS—An IT Structure for Environmental Information via Open Knowledge, Dynamic Dashboards and Social Web of Things. *Proceedings of IADIS International Conference WWW/Internet*, Fort Worth, 22-25 October 2013, 67-74.
  - [7] Yang, K.-P., Alkadi, G., Gautam, B., Sharma, A., Amatya, D., Charchut, S. and Jones, M. (2013) Park-A-Lot: An Automated Parking Management System. *Computer Science and Information Technology*, **1**, 276-279.
  - [8] Miquel, A., Belda, R., de Fez, I., Arce, P., Fraile, F., Guerri, J.C., Martínez, F. and Gallardo, S. (2013) A Power Consumption Monitoring, Displaying and Evaluation System for Home Devices. *Waves*, **5**, 5-13.
  - [9] Craig, T., Polhill, J.G., Dent, I., Galan-Diaz, C. and Heslop, S. (2014) The North East Scotland Energy Monitoring Project: Exploring Relationships between Household Occupants and Energy Usage. *Energy and Buildings*, **75**, 493-503. <http://dx.doi.org/10.1016/j.enbuild.2014.02.038>
  - [10] Palma, D., Agudo, J.E., Sánchez, H. and Macías, M.M. (2014) An Internet of Things Example: Classrooms Access Control over Near Field Communication. *Sensors*, **14**, 6998-7012. <http://dx.doi.org/10.3390/s140406998>
  - [11] Kulkarni, S.S. and Padgilwar, R.S. (2014) Automation over Internet & Android App Using RTOS. *International Journal of Computer and Electronics Research*, **3**, 43-48.
  - [12] Fisher, D.K. and Gould, P.J. (2012) Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research. *Modern Instrumentation*, **1**, 8-20. <http://dx.doi.org/10.4236/mi.2012.12002>

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either [submit@scirp.org](mailto:submit@scirp.org) or [Online Submission Portal](#).

