

Development of an Open-Source Cloud-Connected Sensor-Monitoring Platform

Daniel K. Fisher^{1*}, Reginald S. Fletcher¹, Saseendran S. Anapalli¹, H. C. Pringle III²

¹Agricultural Research Service, Crop Production Systems Research Unit, United States Department of Agriculture, Stoneville, USA

²Delta Research and Extension Center, Mississippi State University, Stoneville, USA

Email: *daniel.fisher@ars.usda.gov, reginald.fletcher@ars.usda.gov, saseendran.anapalli@ars.usda.gov, lpringle@drec.msstate.edu

How to cite this paper: Fisher, D.K., Fletcher, R.S., Anapalli, S.S. and Pringle III, H.C. (2018) Development of an Open-Source Cloud-Connected Sensor-Monitoring Platform. *Advances in Internet of Things*, 8, 1-11.

<https://doi.org/10.4236/ait.2018.81001>

Received: September 26, 2017

Accepted: November 28, 2017

Published: December 1, 2017

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Rapid advances in electronics and communications technologies offer continuously evolving options for sensing and awareness of the physical environment. Many of these advances are becoming increasingly available to “non-professionals,” that is, those without formal training or expertise in disciplines such as electronic engineering, computer programming, or physical sciences, via the open-source concept. The open-source concept of collaboration and sharing of ideas offers advantages including low cost, ease of use, extensive array of electronic technologies offered, and technical and programming support. Expansion of communications infrastructure, including wireless, cellular, and internet networks, continues to provide greater ability to be connected and share information over any distance in real time. A basic data-collection platform using open-source hardware and software and internet cloud components was developed and discussed. The simple and inexpensive platform was used to develop and implement an instrument system to remotely monitor soil-moisture status in agricultural fields. The monitoring system transferred data regularly from the field to an internet website via the cellular communications network. The system performed reliably over an entire growing season with no maintenance requirements. The basic platform can be modified to suit a user’s specific requirements, and offers options for automated collection, viewing, and sharing of remotely sensed data.

Keywords

Arduino, Microcontroller, Cellular, Internet, Soil Moisture, Agriculture

1. Introduction

Rapid advances in electronics and communications technologies offer continuously

evolving options for sensing and awareness of the physical environment. Many of these advances are becoming increasingly available to “non-professionals,” that is, those without formal training or expertise in disciplines such as electronic engineering, computer programming, or physical sciences. Open-source projects such as Arduino (<https://www.arduino.cc>) offer free, informative, and collaborative tools and communities to allow development of unique sensing systems and sharing of ideas and knowledge. Expansion of communications infrastructure, including wireless radio, cellular, and internet networks, continues to provide greater ability to be connected and share information over any distance in real time. A growing number of open-source cloud-based service providers, such as Thingspeak (<https://www.thingspeak.com>), Adafruit IO (<https://io.adafruit.com>), and Thingier.io (<https://thingier.io>) offer internet-based platforms for remote uploading and sharing of user-generated data.

The open-source concept of collaboration and sharing of ideas is increasingly being adopted by research and citizen-scientist communities [1] [2]. Advantages of open-source hardware, including low cost, ease of use, extensive array of electronic technologies offered, and technical and programming support, are available to users in all parts of the world. A number of researchers have begun taking advantage of open-source hardware to develop unique solutions to specific problems related to agriculture [2] [3] [4] [5], irrigation [6] [7], and environmental issues [8] [9] [10].

Remote data transfer and real-time data access are also being explored as means of improving timeliness and convenience of data access. Researchers are evaluating cellular and internet-based services to further expand open-source monitoring, especially in agricultural settings which are typically underserved by emerging technologies [2] [7] [11] [12] [13] [14].

The objective of this work was to develop a data-collection platform using open-source hardware and software and internet cloud components. The simple and inexpensive platform can be modified to suit a variety of applications, and offers options for collecting, viewing, and sharing data from remote locations via the cellular communications network and convenient internet access. An example application developed to monitor soil-moisture status for agricultural irrigation scheduling is presented and discussed.

2. Materials and Methods

The data-collection platform integrates open-source hardware, open-source software, and internet cloud components to enable remote data transfer and access. The basic platform integrates a programmable microcontroller with built-in cellular communications hardware, rechargeable battery, and a low-power battery controller, enabling it to operate for extended periods of time and transmit sensor data to the internet. Software is written and uploaded to the microcontroller to perform data collection and transmission functions, and data are transferred to an internet cloud-based service for viewing and analysis.

2.1. Hardware Component

The data-collection platform consists of a microcontroller development board with integrated cellular modem, rechargeable battery, and low-power timer. Components are mounted on a prototyping board, which provides space for integrating external sensors and components. The hardware components are housed in a weatherproof enclosure for deployment in the field.

The microcontroller development board, Feather 32u4 FONA (Adafruit Industries, New York, NY USA; <https://www.adafruit.com>), consists of an 8-bit microcontroller, cellular modem, and auxiliary electronic circuitry mounted on a single circuit board. The microcontroller provides approximately 28 KB of memory space for program storage and 1 KB non-volatile data storage, and 22 digital and analog input/output (I/O) connections, six of which can be configured as 10-bit analog-to-digital (A/D) converters. Two hardware serial ports are available, one used for communicating with a computer via USB for programming the microcontroller, and a second port for interfacing with external devices. Additional hardware includes dedicated ports for Inter-Integrated Circuit (I2C, I²C) two-wire and Serial Peripheral Interface (SPI) three-wire protocols, standardized communications protocols for interfacing microcontrollers with digital sensors and other peripheral devices.

The microcontroller board provides cellular communications capability via a built-in cellular modem and SIM card holder. The cellular modem operates on the 2G cellular network, which is being phased out and replaced with more advanced networks in many regions. The 2G network is scheduled to remain operational in many parts of the world for several more years, however, until 2020 in the USA (<https://connectedworld.com/the-uncarrier-view-to-2g-sunset/>). In areas where 2G network coverage may be problematic, a similar microcontroller/cellular development board operating on the 3G network, such as the Particle Electron (<https://www.particle.io>) might be an option. The Feather 32u4 FONA modem communicates with the microcontroller via a software serial connection, and offers low-power sleep functions designed to enable long-term battery-powered operation. The microcontroller and cellular modem operate at a 3.3 V level, supplied from a rechargeable 3.7 V Lithium Ion Polymer Battery (Adafruit Industries) and onboard 3.3 V voltage regulator.

The cellular modem requires an external antenna and a micro-size SIM card and active cellular data plan on a 2G network to enable cellular communications. A SIM card and pre-paid data plan were purchased for use in the USA from Embedded Works Corporation (Santa Clara, CA USA; <http://www.embeddedworks.net>). The plan allows the transfer of up to 1 MB of data per month, for a period of 12 months.

To enable long-term remote operation, current consumption on the battery needs to be minimized. In many applications, data are collected at regular but infrequent intervals, with no activity occurring between measurement periods. Microcontroller-based measurement systems are often programmed to spend

time between measurements in a low-power sleep mode. A hardware option can also be used to turn off almost completely all power to the circuit. The TPL5110 Low Power Timer Breakout (Adafruit Industries) acts as an interface between the battery and the microcontroller and any additional circuitry, and can be configured to connect and disconnect battery power to the microcontroller circuit at regular user-defined intervals. The intervals are set by adjusting an on-board variable resistor (potentiometer), or can be set with the addition of an external resistor. After powering on and completing a measurement cycle, the low power timer disconnects the battery supply completely from the circuit. The circuit shuts down, and no current consumption occurs from the microcontroller circuit. Current consumption does occur from the low power timer, but on the order of approximately 20 μA rather than the several-milliamp rate if the circuit were in the more-common software sleep mode.

These basic hardware components are mounted on a prototype board using female headers (Adafruit Industries) soldered to the prototype board. Mating male header pins soldered to the microcontroller board and other components are inserted into the female headers. Additional components needed for a particular sensing or monitoring application can then be added to the prototype board to complete a measurement circuit. The basic hardware components are shown in **Figure 1**.

2.2. Software Component

The microcontroller is programmed using the open-source Arduino Integrated Development Environment (Arduino IDE; <https://www.arduino.cc>), which is downloaded and installed on a personal computer. The IDE is based on the C++ programming language, and is used to write software code, compile the code and check for errors, and upload the compiled program to the microcontroller. The IDE's serial monitor allows output from the microcontroller to be viewed on the connected computer, and can also be used to send instructions from the computer to the microcontroller.

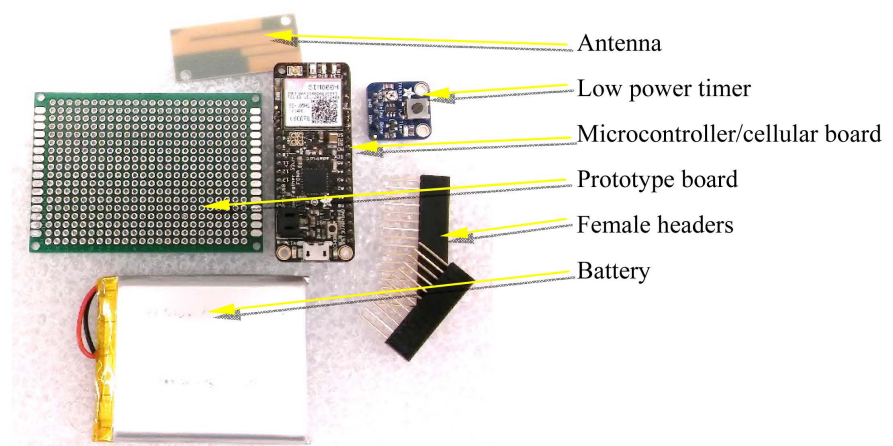


Figure 1. Basic hardware components.

An important feature of the IDE and the programming language is the capability of including external libraries in a microcontroller program. Many users in the open-source Arduino community have written libraries, which are software code routines that provide certain features or control specific sensors or other external peripherals, such as real-time clocks, data storage devices, or cellular modems. The libraries contain the instructions for interfacing with the devices and accessing the devices' features, allowing the use of these devices without having to learn the internal workings and complexities of the devices, which often require significant knowledge and expertise. Use of these libraries allows the user to more easily incorporate the peripheral devices into a circuit for rapid development of a unique measurement or control system.

2.3. Internet Cloud-Based Component

Thingspeak (<https://thingspeak.com>) is an internet cloud service designed to allow sensor data streams to be posted to the internet to be viewed in real-time. Configuration of data streams, called channels, is simple and straightforward, and multiple options are available to customize the appearance and display of the data streams. The ability to integrate and execute code written for the MATLAB technical and mathematics language (<http://www.mathworks.com>) allows for online analysis and processing of incoming data immediately as the data are posted.

Using the Thingspeak cloud-based service begins by creating a free user account. After logging into the account, the user creates a new channel to accept a data stream, and a webpage is displayed with configuration options for the channel. The user lists the data fields that will be uploaded, and can give the channel a name and add information and keywords to describe the data. A channel number is assigned to identify the data stream, and a unique key code, called the API key, is generated for this channel that allows the user to upload data. Tabs on the webpage link to additional configuration options, allowing the user to customize the appearance of the data displayed. An example channel configuration for a soil-moisture monitoring system discussed below is shown in **Figure 2**.

Data are uploaded to the channel by assembling and posting a unique website address (URL). The URL contains the Thingspeak website domain name, API key, and data field values. An example URL for the channel shown in **Figure 2** would take the form

http://api.thingspeak.com/update?key=_API_key_&field1=-25&field2=-55&field3=-101&field4=-55&field6=-65&field6=4100. This URL would then be sent via cellular modem to the Thingspeak website, and each data display would be updated immediately.

An additional setting configures the channel's webpage as either private or public. If private, the webpage can only be viewed by the user after logging into the user's account. If public, the channel's webpage can be viewed by anyone via

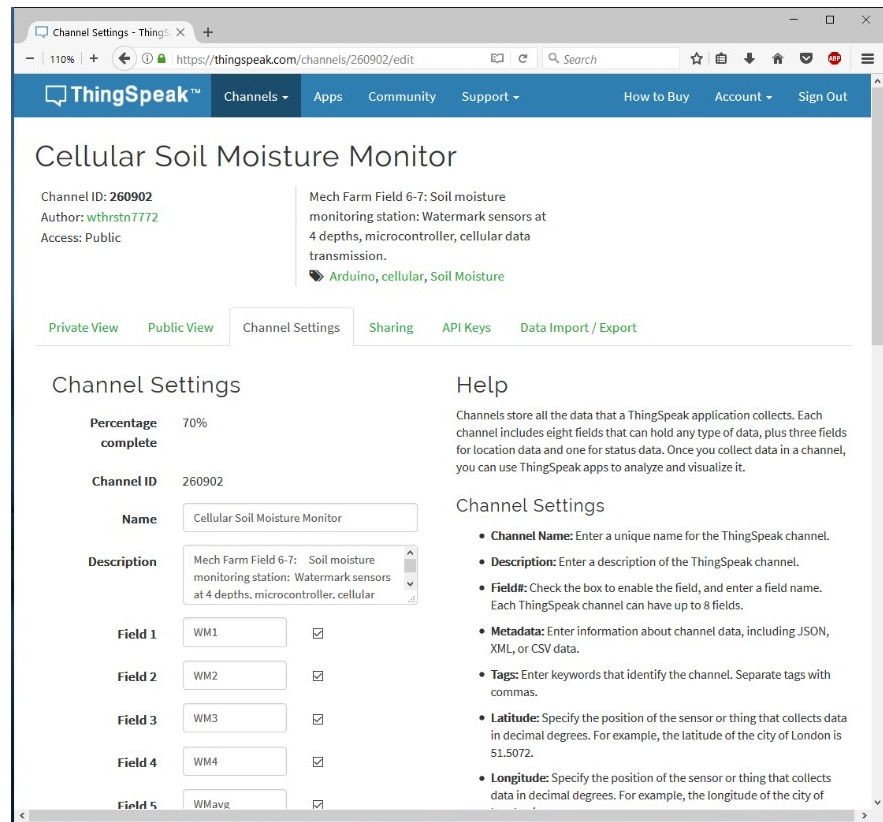


Figure 2. Example of data stream configuration for cloud-based data-hosting service.

the data stream's channel ID number. The webpage could, therefore, be viewed at any time from any place via a web browser on a computer or smartphone.

3. Soil-Moisture Monitoring System

Agricultural fields are often remotely located in rural areas, and require a producer's time and labor to travel to and visit fields to assess field conditions. A monitoring system was developed to measure soil-moisture status for use in scheduling the appropriate timing of irrigation water applications. The automated monitoring system measures soil-moisture conditions at regular intervals and posts the data to the Thingspeak website, allowing the producer to view field conditions remotely and determine when an irrigation event is needed.

3.1. Hardware

The soil-moisture monitoring system was developed by incorporating the basic hardware components discussed previously, shown in **Figure 1**, and additional circuitry for reading soil-moisture sensors. Four moisture sensors, Watermark model 200SS granular matrix sensors (Irrrometer Company, Inc, Riverside, CA USA; <https://www.irrometer.com>), connect to the circuit via spring-loaded terminal blocks (Adafruit Industries). Sensor measurement circuitry consists of resistor half-bridge (voltage divider) circuits, one for each of the four soil-moisture sensors. The moisture sensors perform electrically as variable resistors, with

sensor resistance varying in proportion to soil-water content. Each sensor acts as one resistor in the half-bridge circuit, with the second resistor being a fixed-value (10 k-ohm) resistor.

The TPL5110 Low Power Timer was adjusted to power the microcontroller-based circuit at regular intervals by setting an internal resistance via a built-in potentiometer. Since soil-moisture conditions usually change very slowly, the potentiometer was set to its maximum resistance (approximately 100k ohm), resulting in a measurement interval of approximately 2.5 hrs.

The electrical circuit schematic for the moisture-sensor monitor is shown in **Figure 3(a)**. The circuit is laid out on a prototype board in **Figure 3(b)**, showing the locations of female headers, spring terminals, and jumper wires. A complete, assembled monitoring system with one soil-moisture sensor connected is shown in **Figure 4**. Hardware components, sources, and approximate costs are listed in **Table 1**. Fabrication of the monitoring system required approximately one hour to complete using basic soldering materials. Final cost of the system totaled approximately US\$85 for the data-collection platform hardware, and US\$30 each for the soil-moisture sensors. A SIM card and cellular data plan cost an additional US\$30 per year.

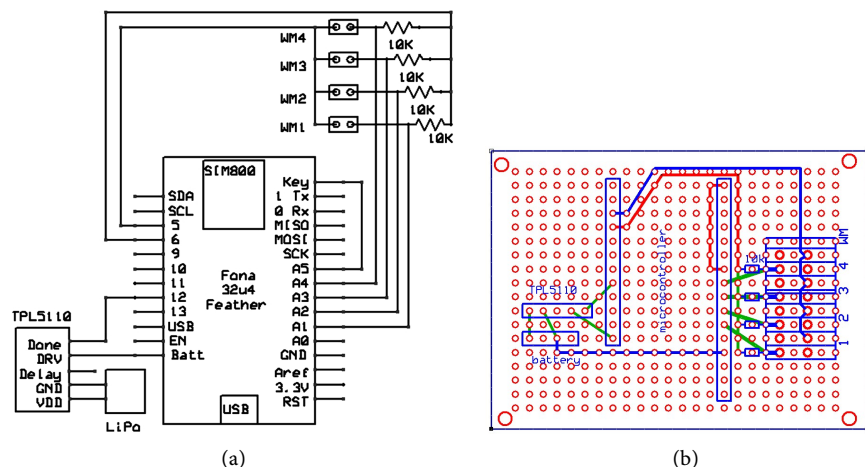


Figure 3. Electrical circuit schematic (a) and prototype board (b) for the soil-moisture monitoring system.

Table 1. List of materials and approximate costs for soil-moisture monitoring system.

Main components	Part number	Source	Cost (US\$)
Microcontroller/cellular board	Feather 32u4 FONA	Adafruit Industries	45
Cellular antenna	Sticker-type GSM	Adafruit Industries	3
Rechargeable battery	3.7 V LiPoly, 2500 mAh	Adafruit Industries	15
Terminal block	Spring terminals	Adafruit Industries	3
Low-power timer	TPL5110	Adafruit Industries	5
Weatherproof enclosure	NBF-32002	Mouser Electronics	9
Miscellaneous (protoboard, headers, resistors, connector)			5
Total			85
Soil-moisture sensors (4)	Watermark 200SS	Irrrometer Company	120
SIM card, cellular data plan	1 MB/month, 1 year	Embedded Works	30

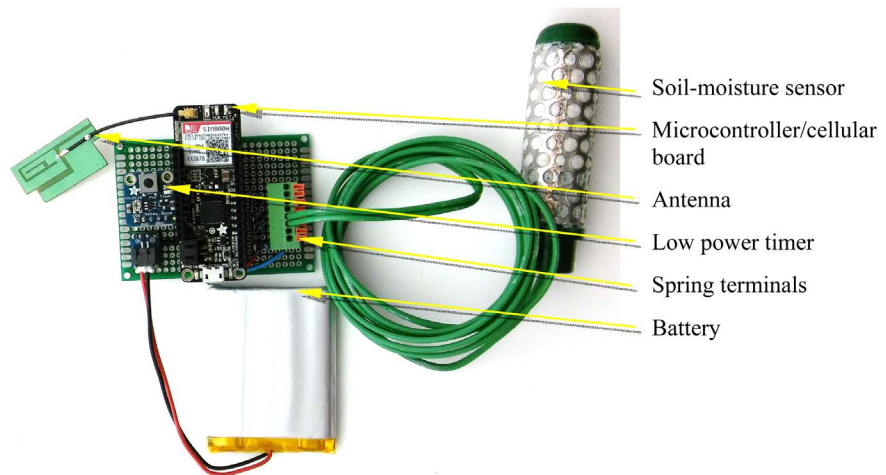


Figure 4. Monitoring system components.

3.2. Software

The program to control the monitoring system was written in the Arduino IDE, compiled, and uploaded to the microcontroller. The program includes functions for reading the four soil-moisture sensors, assembling the Thingspeak URL, and transmitting the sensor data to the Thingspeak website via the cellular communications network. The program is open-source and freely available by contacting the corresponding author.

At each measurement interval, the low power timer activates the microcontroller-based circuit by supplying battery power to the circuit, and the microcontroller program begins execution. Each soil-moisture sensor is read sequentially by applying an alternating-polarity excitation voltage to the sensor half-bridge circuit. An excitation voltage with one polarity is applied and the center voltage of the half-bridge, between the sensor and fixed resistor, is measured with the microcontroller's built-in A/D converter. The polarity is reversed, and the center voltage is measured again. This process is repeated five times, and the average voltage at each polarity is determined. Sensor resistance is then calculated using the basic voltage-divider equation, and soil-water potential (also referred to as tension), in units of kPa, is calculated by applying a calibration equation that relates resistance to potential. The circuit and calibration are described in detail by Fisher and Gould [2].

The cellular modem is then turned on, and serial communications are established with the microcontroller. The modem registers on the cellular network to establish cellular and data services. The Thingspeak website URL is assembled, and data are sent to the data-hosting website. If the cellular modem cannot register on the cellular network, or the data-upload process is not successful, the modem is turned off and the process is repeated, up to a maximum of three times. Following successful data transmission, or three unsuccessful attempts, the microcontroller sends a signal to the low power timer, which completely disconnects battery power from the microcontroller and circuit.

3.3. Field Deployment of the Monitoring System

Soil-moisture monitoring systems were deployed in several agricultural fields at the United States Department of Agriculture, Agricultural Research Service's Crop Production Systems Research Unit farm at Stoneville, MS USA during the cropping season of 2017. At each site, soil-moisture sensors were installed in the soil profile at depths of 30, 60, 90, and 120 cm below the soil surface. The electronic monitoring hardware was placed inside a weatherproof enclosure mounted on a wooden stake that was driven into the soil near the sensors. Sensor wires were inserted into the terminal blocks on the system circuit board, the battery was connected, and the monitoring system began processing and transmitting soil-moisture measurements.

Throughout the cropping season, six data values, consisting of four sensor measurements, an average of the four measurements, and battery voltage, were transmitted at 2.5-hr intervals. The website's data display was configured to show the latest 30-days of data. For the soil-moisture measurements, the y-axis of each data graph was set to the limits of the sensor measurements, from 0 to -200 kPa. Data are displayed for one monitoring site in **Figure 5**.

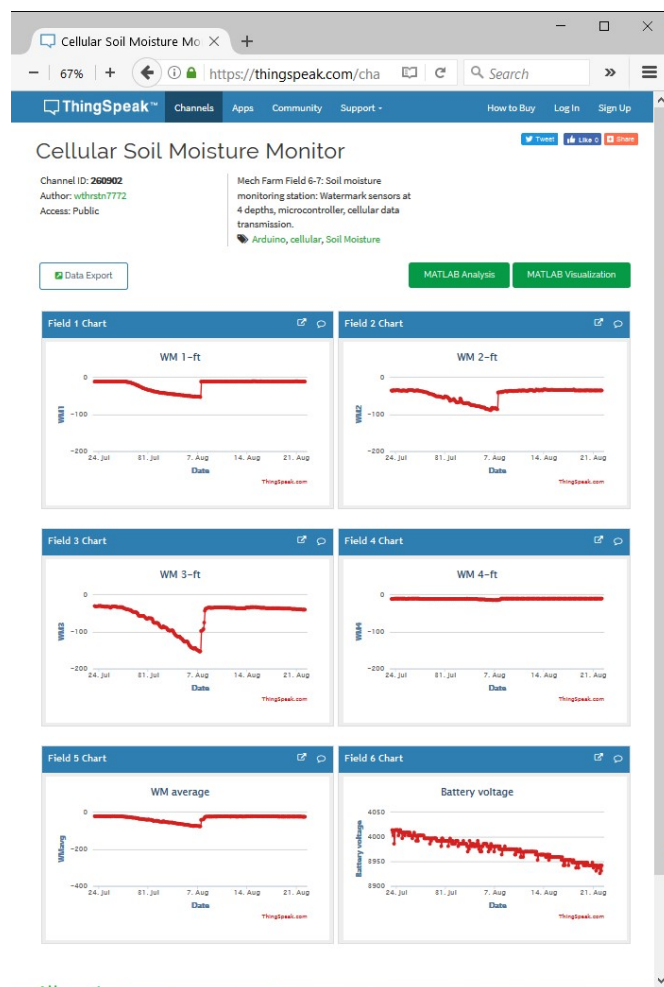


Figure 5. Soil-moisture monitor data stream webpage.

Data were transmitted with a 98% success rate, with almost no missing measurement intervals. The monitoring systems were installed in the field in mid-May 2017 and operated uninterrupted until harvest in September 2017, a period of almost four months (116 days). The systems required no maintenance or battery replacement during this period, which included almost the entire growing season. During development of the monitoring system, current consumption during active measurement and cellular data-transmission events and low-power periods was measured to estimate battery life. On average, active periods lasted 41 seconds with an average current consumption of 53 mA, while current consumption dropped to approximately 20 μ A during low-power periods. With a battery capacity of 2500 mAh and sending data at 2.5-hour intervals, battery life was estimated, as (active current * active time period + sleep current * sleep period)/measurement interval, to be 390 days.

4. Conclusions

Advances in electronic technologies and communications infrastructures offer many options for development and implementation of sensing and monitoring systems. Open-source concepts of inexpensive hardware and open exchange of ideas and collaboration offer tools and support for development of unique sensing and monitoring systems. Cloud-based services offer convenient and real-time access to and sharing of remotely sensed data.

A basic data-collection platform using open-source hardware and software and internet cloud components was developed and discussed. The simple and inexpensive platform was used to develop and implement an instrument system to remotely monitor soil-moisture status in agricultural fields. The monitoring system transferred data regularly from the field to an internet website via the cellular communications network. The system performed reliably over an entire growing season with no maintenance requirements. The basic platform can be modified to suit a user's specific requirements, and offers options for automated collection, viewing, and sharing of remotely sensed data.

Acknowledgements

The authors wish to acknowledge the efforts of Ms. Keysha Hamilton, Biological Science Technician, and her expertise in the fabrication of the prototype circuit boards deployed in this project.

Disclaimer

Mention of a trade name, proprietary product, or specific equipment does not constitute a guarantee or warranty by the United States Department of Agriculture, and does not imply approval of the product to the exclusion of others that may be available.

References

- [1] Pearce, J.M. (2012) The Case for Open Source Appropriate Technology. *Environ-*

- ment, *Development and Sustainability*, **14**, 425-431.
<https://doi.org/10.1007/s10668-012-9337-9>
- [2] 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.
<https://doi.org/10.4236/mi.2012.12002>
- [3] Lozoya, C., Aguilar, A. and Mendoza, C. (2016) Service Oriented Design Approach for a Precision Agriculture Datalogger. *IEEE Latin America Transactions*, **14**, 1683-1688. <https://doi.org/10.1109/TLA.2016.7483501>
- [4] Di Prima, S., Lassabatere, L., Bagarello, V., Iovino, M. and Angulo-Jaramillo, R. (2016) Testing a New Automated Single Ring Infiltrometer for Beerkan Infiltration Experiments. *Geoderma*, **262**, 20-34.
<https://doi.org/10.1016/j.geoderma.2015.08.006>
- [5] Thalheimer, M. (2016) A New Optoelectronic Sensor for Monitoring Fruit or Stem Radial Growth. *Computers and Electronics in Agriculture*, **123**, 149-153.
<https://doi.org/10.1016/j.compag.2016.02.028>
- [6] Masseroni, D., Facchi, A., Vannutelli Depoli, E., Renga, F.M. and Gandolfi, C. (2016) Irrig-OH: An Open-Hardware Device for Soil Water Potential Monitoring and Irrigation Management. *Irrigation and Drainage*, **65**, 750-761.
<https://doi.org/10.1002/ird.1989>
- [7] Payero, J.O., Mirzakhani-Nafchi, A., Khalilian, A., Qiao, X. and Davis, R. (2017) Development of a Low-Cost Internet-of-Things (IoT) System for Monitoring Soil Water Potential Using Watermark 200SS Sensors. *Advances in Internet of Things*, **7**, 71-86. <https://doi.org/10.4236/ait.2017.73005>
- [8] Fisher, D.K. (2014) Rapid Deployment of Internet-Connected Environmental Monitoring Devices. *Advances in Internet of Things*, **4**, 46-54.
<https://doi.org/10.4236/ait.2014.44007>
- [9] Lockridge, G., Dzwonkowski, B., Nelson, R. and Powers, S. (2016) Development of a Low-Cost Arduino-Based Sonde for Coastal Applications. *Sensors*, **16**, 528.
<https://doi.org/10.3390/s16040528>
- [10] Mesas-Carrascosa, F.J., Verdú Santano, D., Meroño, J.E., Sánchez de la Orden, M. and García-Ferrer, A. (2015) Open Source Hardware to Monitor Environmental Parameters in Precision Agriculture. *Biosystems Engineering*, **137**, 73-83.
<https://doi.org/10.1016/j.biosystemseng.2015.07.005>
- [11] Ma, J., Zhou, X. and Li, S. (2011) Connecting Agriculture to the Internet of Things THROUGH Sensor Networks. *International Conference on Internet of Things (iThings/CPSCoM), and 4th International Conference on Cyber, Physical and Social Computing*, Dalian, 19-22 October 2011.
<https://doi.org/10.1109/iThings/CPSCoM.2011.32>
- [12] Dlodlo, N. and Kalezhi, J. (2015) The Internet of Things in Agriculture for Sustainable Rural Development. *International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Windhoek, 17-20 May 2015.
- [13] Malche, T. and Maheshwary, P. (2017) Internet of Things (IoT) Based Water Level Monitoring System for Smart Village. *Proceedings of International Conference on Communication and Networks*, Springer, Singapore, 305-312.
https://doi.org/10.1007/978-981-10-2750-5_32
- [14] Ray, P.P. (2017) Internet of Things for Smart Agriculture: Technologies, Practices and Future Direction. *Journal of Ambient Intelligence and Smart Environments*, **9**, 395-420. <https://doi.org/10.3233/AIS-170440>