Hydrology and
Earth System
Sciences

Open-source Arduino-compatible data loggers designed for field research

Andrew D. Wickert[1,2,3], **Chad T. Sandell**[3], **Bobby Schulz**[1,3,4], **and Gene-Hua Crystal Ng**[1,2]

[1]Department of Earth Sciences, University of Minnesota, John T. Tate Hall, 116 Church St. SE,
Minneapolis, MN 55455, USA
[2]Saint Anthony Falls Laboratory, University of Minnesota, 2 3rd Ave. SE, Minneapolis, MN 55414, USA
[3]Northern Widget LLC, Saint Paul, MN, USA
[4]Department of Electrical and Computer Engineering, University of Minnesota, Kenneth H. Keller Hall,
200 Union St. SE, Minneapolis, MN 55455, USA

Correspondence: Andrew D. Wickert (awickert@umn.edu)

Abstract. Automated electronic data loggers revolutionized environmental monitoring by enabling reliable high-frequency measurements. However, the potential to monitor the complex environmental interactions involved in global change has not been fully realized due to the high cost and lack of modularity of commercially available data loggers. Responding to this need, we developed the ALog (Arduino logger) series of three open-source data loggers, based on the popular and easy-to-program Arduino microcontroller platform. ALog data loggers are low cost, lightweight, and low power; they function between -30 and $+60\,°\text{C}$, can be powered by readily available alkaline batteries, and can store up to 32 GB of data locally. They are compatible with standard environmental sensors, and the ALog firmware library may be expanded to add additional sensor support. The ALog has measured parameters linked to weather, streamflow, and glacier melt during deployments of days to years at field sites in the USA, Canada, Argentina, and Ecuador. The result of this work is a robust and field-tested open-source data logger that is the direct descendant of dozens of individuals' contributions to the growing open-source electronics movement.

1 Introduction

Studies of complex environmental systems require high-density and widespread environmental data (Lovett et al., 2007). Such information is necessary to establish baseline environmental conditions, to track global change, and to build theory that is consistent with observations. In spite of three decades of rapid advances in measurement technology (Hirschfeld, 1985; Martinez et al., 2004; Hart and Martinez, 2006; Ferdoush and Li, 2014), most of Earth's surface still needs higher-resolution monitoring in order to understand the consequences of global change and prepare for the future (Vitousek, 1994; Tauro et al., 2018). This shortfall results primarily from instrumentation cost (Oliveira and Rodrigues, 2011), hardware requirements to work in harsh environmental conditions, obstacles to advanced technology and repairs in less-developed countries (e.g., Reda et al., 2017), and power and data-retrieval limitations (Martinez et al., 2004; Padhy et al., 2005).

The growing open-source electronics movement has given scientists new tools to develop technologies for both lab and field research (Harnett, 2011; Pearce, 2012; Cressey, 2017), including automated data loggers (Fisher, 2012; Wickert, 2014; Hicks et al., 2015; Hund et al., 2016; Beddows and Mallon, 2018; Hicks et al., 2019). These innovations have led to significant advances in research and monitoring (Tauro et al., 2018). What the field-monitoring community requires from the open-source movement is a low-power, modular, single-board data logger that is easy to use and whose code and hardware designs are well documented and freely available.

We answered this need by developing the ALog (Arduino logger), a small, lightweight, and low-power data-logging system that is a fraction of the cost of conventional proprietary data-recording systems (Wickert and Sandell, 2017; Sandell et al., 2018). Hardware advances alone cannot produce an effective standalone measurement platform, so we paired our new designs with custom-built firmware libraries – built atop the popular and easy-to-use Arduino platform – and software to streamline data-logger programming. We iterated development and field testing from 2010 to present and deployed each round of prototypes across rugged environments, including glaciers, tundra, the high alpine, and wetlands (Wickert, 2014; Tauro et al., 2018; Saberi et al., 2019). Here we present a suite of modern open-source data loggers and the principles that guided their development.

2 Design

The ALog series of three data loggers (Fig. 1) were designed as an integrated set of hardware, software, and firmware (Fig. 2). Together, these layers of the embedded system and its interfaces enable low-power data collection. The hardware includes a set of subsystems to manage power, data storage, timekeeping, sensor interfacing, and connections to computers (for testing, clock setting, and programming). This hardware is tightly integrated with the ALog firmware library (Wickert et al., 2018a), which manages low-level utilities (power, boot sequence, fail-safes), on-board hardware (the clock and data storage systems, through their own libraries), and a library of sensor commands. The full system is programmable through the Arduino integrated development environment (IDE), which is designed for use by beginner programmers and therefore lowers the barrier to entry for environmental monitoring. All hardware designs and code for the fully open-source ALog system are available at https://github.com/NorthernWidget (last access: 18 April 2019). Core firmware and software are licensed under the GNU General Public License (GPL), which requires that all derivatives of the ALog remain open source, and hardware is released under the Creative Commons Attribution-ShareAlike license.

To design the ALog series of data loggers, we followed the approach taken by the popular Arduino project (Barragán, 2004; Banzi and Shiloh, 2014) in order to maintain compatibility with open-source standards. We designed the circuitry (hardware) using EAGLE (Cadsoft and Autodesk, 2019), an electronic schematic and board-layout program that is freely available for noncommercial use and is a de facto standard across the open-source community. We wrote the data logger software in the Wiring/Arduino variant of C++ (Barragán, 2004), using an object-oriented framework that abstracts the low-level core components of embedded hardware programming (e.g., writing bits to registers) into intuitive functions to read, write, and operate a microcontroller. The
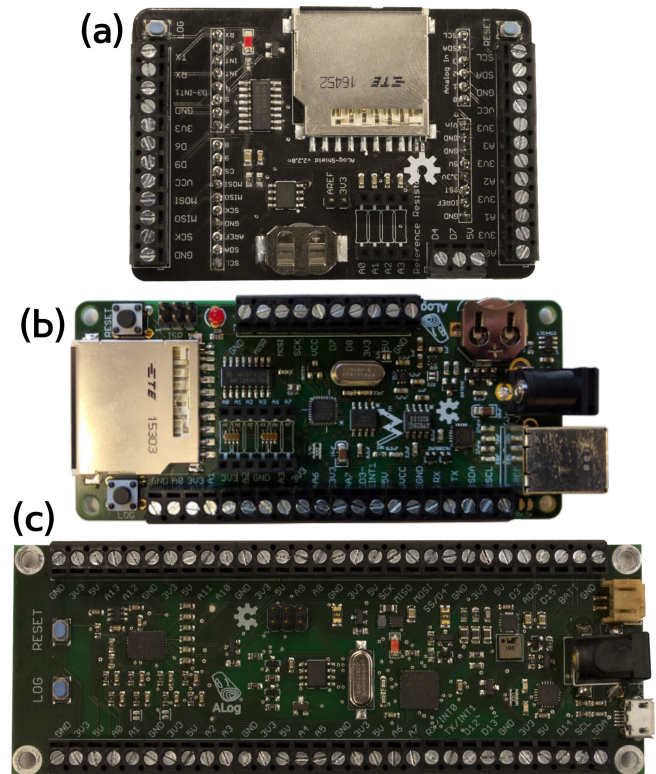
Figure 1. Photos of the ALog series of data loggers. **(a)** The ALog Shield 2.2 (Wickert et al., 2018b), which nests atop a standard Arduino board. **(b)** The ALog BottleLogger 2.2 (Wickert and Sandell, 2017). **(c)** The ALog BottleLogger 3.0 (Sandell et al., 2018); the battery and SD card holders are on the reverse side.

programmable core of the ALog is compatible with Arduino, enabling the use of its extensive firmware libraries and IDE. Multiple examples are bundled with the ALog library, which is fully documented using doxygen (van Heesch, 2008), a program that automatically converts in-code documentation into a users' manual. We customized doxygen to include the README Markdown file, which includes instructional text and images, at the beginning of the automatically generated manual. The customized doxygen configuration is available at https://github.com/NorthernWidget/ALog/tree/master/doc (Wickert et al., 2018a), and the generated reference manual is included in the Supplement.

2.1 Hardware

The ALog series comprises three main data loggers (Fig. 1): (a) the ALog Shield 2.2 (Wickert et al., 2018b), (b) the ALog BottleLogger 2.2 (Wickert and Sandell, 2017), and (c) ALog BottleLogger 3.0 (Sandell et al., 2018) (Table 1). The ALog BottleLogger data loggers are standalone units (i.e., no built-in telemetry) that we designed for field research; version 3.0 has a more powerful microcontroller core and a dedicated 16-bit analog-to-digital converter (ADC), whereas version 2.2
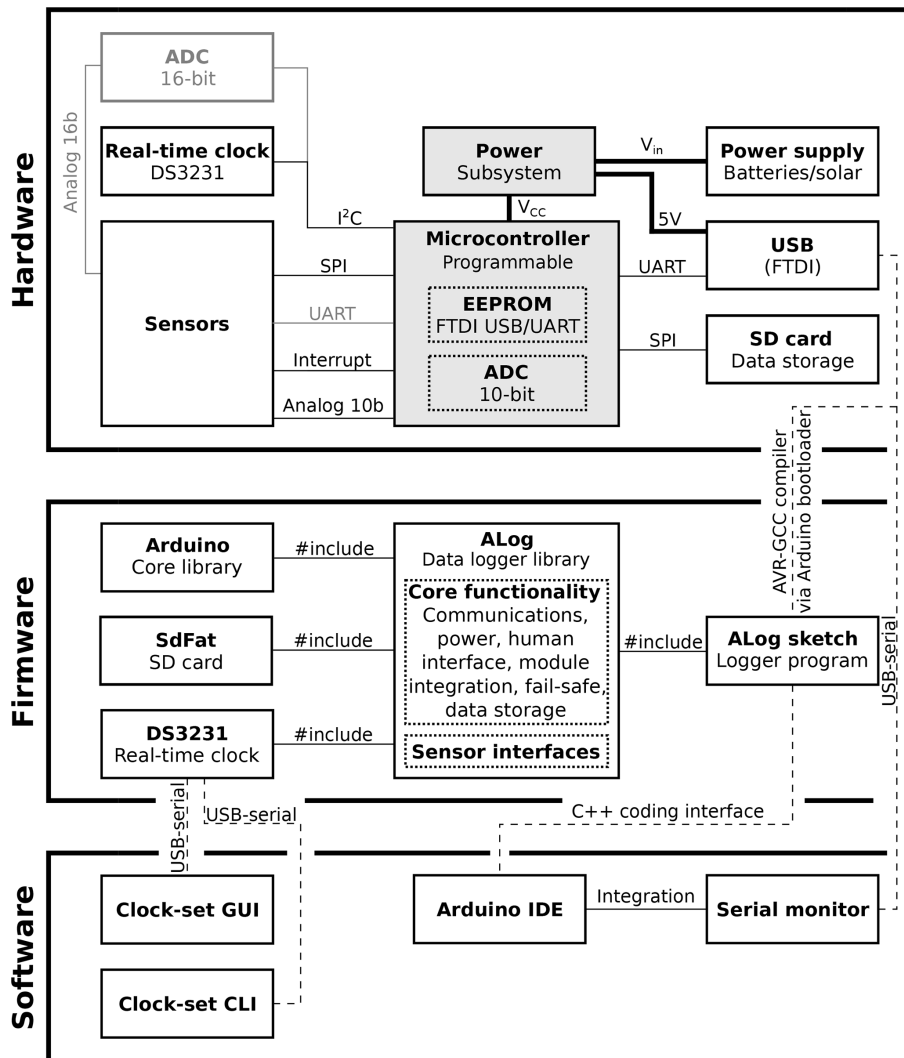
Figure 2. Flowchart of ALog hardware, firmware, and software. "Hardware" includes the generalized subsystems of the physical ALog. "Firmware" includes all of the code that runs on the AVR microcontroller. "Software" runs on the user's computer. Dashed lines indicate temporary connections (e.g., during programming), whereas solid lines indicate permanent connections and dependencies. Dotted lines within boxes contain information on components of larger systems. Lines in gray indicate features that are included only on the BottleLogger 3.0; gray-shaded boxes denote features that are incorporated into the BottleLogger designs but for the ALog Shield are supplied by a standard Arduino. ADC: analog–digital converter; EEPROM: nonvolatile variable memory that holds values after a power reset; I²C, SPI, UART, interrupt, analog: communications protocols; GUI: graphical user interface; CLI: command-line interface.

uses less power and fewer components. Both are described in more detail in this section. The ALog Shield 2.2 functions as an entry point for the global community of Arduino users (Buechley and Eisenberg, 2008; Cressey, 2017) to develop their own scientific-data-logging capabilities. It nests atop a standard Arduino board (Barragán, 2004; Banzi and Shiloh, 2014), such as the Arduino Uno. The power consumption of this pair is too high for field deployment, but together these form a benchtop prototyping system that is compatible with the ALog firmware (Wickert et al., 2018a). The Supplement contains the electrical schematics and circuit board layouts for all three data loggers.

Each data logger of the ALog series contains six critical subsystems (Fig. 2): power, timekeeping, data storage, sensor interfaces, input/output (I/O), and the microcontroller core. The high-efficiency power system permits multi-year deployments on a single set of primary alkaline batteries. These battery lifetime measurements are based both on extrapolation from laboratory power-consumption measurements (Table 1) and field deployments that remained unvisited for ≥ 1 year (Armstrong et al., 2016). A high-accuracy real-time clock (RTC) keeps time, regulates logging intervals, and is temperature-compensated to reduce drift to $\pm 0.432\,\mathrm{s\,d^{-1}}$ (firmware implementation by Ayars and Wickert, 2018). Data

Table 1. ALog attributes.

	Shield 2.2 + Uno	BottleLogger 2.2	BottleLogger 3.0
Width (mm)	78.7	44.1	44.4
Length[a] (mm)	85.7	113	120.8
Mass[a] (g)	63.49	42.66	53.84
Input voltage (V)	3.3–12	3.5–5.0	2.5–12
Power (sleep)[b] (μA)	34 000	12	80
Power (awake)[b] (mA)	54	7.5	11.9
MCU	ATMega328p	ATMega328p	ATMega644p[c]
Clock speed (MHz)	16	8	8
Program memory (KB)	32	32	64
Variable memory (KB)	2	2	4
EEPROM (KB)	1	1	2
External interrupts	1	1	2
Analog I/O	10-bit (4)	10-bit (6)	16-bit (16)
Digital I/O[d]	4	2	6
I^{2}C	1	1	2
SPI	1	1	1
Dedicated UART	0	0	1
Data storage (GB)	32	32	32
RTC drift (s d^{-1})	±0.432	±0.432	±0.432

[a] Including the SD card and backup battery. [b] At 4.5 V input; "awake" state is not including additional power draw from sensors. [c] Also compatible with the ATMega1284p, with 128 KB program memory, 8 KB variable memory, and 4 KB EEPROM. [d] 10-bit analog I/O also functions as digital I/O.

are written as ASCII comma-separated text files to Secure Digital (SD) cards for low-cost, high-volume storage. Screw terminals connect sensors and other peripherals to the ALog, where they link to the appropriate interfaces on the microcontroller. The ALog communicates with and is programmed by a computer via a USB–serial converter that links the computer's USB interface with the universal asynchronous receiver–transmitter (UART) of the microcontroller. Each ALog is built around a reprogrammable 8-bit AVR microcontroller (Wollan et al., 1998; Bogen and Wollan, 1999).

While it is a simple design decision, using an SD card instead of internal memory has multiple advantages. The opensource SdFat interface library written by Greiman (2016) allows up to 32 GB of data to be written in human-readable comma-separated ASCII format. By recording data in text files on SD cards, we also simplify data download and visualization, making it easier for field staff and citizen scientists to work with the ALog. Removable storage limits the time that the box must be opened and exposed to the elements and also allows SD cards to be swapped in the field. We chose standard large-format SD cards because they include space to write physical notes and because smaller cards are more easily lost in the field.

We greatly reduced power consumption – a key system feature – while simplifying power supply options. To reduce power consumption, we implemented a "sleep" cycle to shut down all nonessential subsystems while not logging, utilized a lower-speed (8 MHz) crystal to set the processor clock speed, and powered the ALog using either an unregulated power supply (BottleLogger 2.2) or a step-up–stepdown (buck–boost) converter with a high-efficiency switching architecture (BottleLogger 3.0). Power may be supplied by primary alkaline cells – commonly available across the globe – or through a solar panel, rechargeable battery (typically lithium ion), and charge controller. When powered by three in-series AA alkaline cells ($\sim$ 2600 mAh for these calculations) and awake (logging) for 1 s per minute, the ALog BottleLogger 2.2 can run for $\sim$ 2 years and the BottleLogger 3.0 can for $\sim$ 1 year, based on our laboratory measurements of power consumption by the data logger alone. This time may decrease if sensors that require significant power compared to the "awake" state of the logger are attached (Table 1). If this is the case, D cells ($>$ 10 000 mAh) may be a suitable alternative. In our field deployments (Sect. 3), we typically recorded data once every 10 min, further increasing battery life. As a result of this low power consumption, we ran our ALog field deployments exclusively with primary alkaline cells. In addition, any ALog may be powered over USB, and a diode array prevents short circuits between USB and external power supplies.

In order to reduce power consumption, which is especially important for remote field deployments, we decided

Table 2. ALog labor and pricing. All costs are given in US dollars. Component costs were found at Digi-Key (https://www.digikey.com/). Board prices were sourced from OSH Park (https://oshpark.com/). Build and testing times are based on our work with the data loggers, and the labor costs are estimated based on quotes from Caltronics Design & Assembly (https://caltronicsdesign.com/). All values are per board, and all prices were determined on 12 January 2019.

	Shield 2.2	BottleLogger 2.2	BottleLogger 3.0
Components (QTY 1)	USD 22.24	USD 43.38	USD 59.46
Components (QTY 100)	USD 14.45	USD 28.30	USD 39.43
Board (QTY 3)	USD 10.87	USD 11.28	USD 13.85
Board (QTY 100)	USD 6.52	USD 6.77	USD 8.31
Build and test time (QTY 1)	20 min	75 min	120 min
Build and test labor estimate (QTY 100)	USD 20	USD 35	USD 45

not to include on-board telemetry. Off-board radio (e.g., RFM95, XBee), mobile phone (e.g., Particle Electron, Particle Boron), or satellite (SPOT, Iridium) telemetry packages could be added through the exposed digital interfaces on the ALog data logger. However, such additions would require their own significant power paths, including rechargeable batteries, charge controllers, and solar panels, thus negating much of the low-power benefit of the ALog BottleLogger design. In contrast to the ALog BottleLogger, the open-source Mayfly data logger (Hicks et al., 2015, 2019) includes an XBee radio header (Hicks et al., 2019) and firmware support for radio telemetry (Aufdenkampe et al., 2017; Damiano et al., 2019). In addition, Adafruit (Feather) or particle internet-of-things (IoT) boards can serve as low-cost platforms for telemetry; these do not necessarily include an on-board RTC or data storage, reducing cost but making consistent telemetry critical. These alternatives to the ALog BottleLogger are effective options for deployments in which data return and not power consumption is the variable to optimize.

ALog data loggers are designed for versatility in the field. Each ALog is only $\sim$ 50 g due to the low mass of solid-state electronic components mounted on a circuit board (Table 1). Their low mass and associated small size, along with the stability of solid-state electronics, help users to carry ALog data loggers into the field and reduce the chance of damage if they are dropped. All on-board electronics are rated to function between -40 and $+85\,°C$ (standard "industrial" components); commonly available batteries can power the ALog between temperatures of -30 and $+60\,°C$. ALog data loggers are inexpensive and accessible because they are built from standard off-the-shelf components that are available in most of the world. Further reducing total system cost and increasing versatility, the ALog's generalized set of sensor interfaces allows it to read data from common and inexpensive commercially available sensors. This sets it apart from closed-source data loggers, which are designed to interface with specific proprietary sensors. This combination of affordability and accessibility can help to expand the reach of automated environmental observations and reduce the financial risk associated with recording data in hazardous or un-

secured locations. Finally, the open-source schematics and circuit board layout assist users in diagnosing and repairing their own ALog data-logger systems.

2.2 Firmware

We built a firmware library (see design files in the Supplement) that streamlines ALog programming through a modular two-component architecture (Wickert et al., 2018a). The first is a set of utilities that manage logger core functionality. The second is a library of functions that communicate with and record data from sensors (Table 3). This separation prevents users from altering the code that manages core logger functions when adding or editing sensor functions. We classify this code as "firmware" rather than "software" because it is uploaded to the microcontroller as a semipermanent set of instructions that exists in program memory until it is externally wiped and replaced. The ALog library is written in the Wiring/Arduino variant of C++, which is the standard for open-source microcontrollers (Barragán, 2004; Banzi and Shiloh, 2014). It maintains compatibility with any standard Arduino-compatible device in order to ensure that the ALog firmware can support the open-source community even in the absence of ALog hardware. This library is then imported, and its core ALog class is instantiated, within an ALog program (Arduino "sketch") that is uploaded to the data logger.

The core-utilities portion of the library manages its power and logging cycle and interfaces with the user, SD card, and RTC. The firmware reduces power consumption by a factor of 150–625 from an "always-on" state (Table 1) by instructing the system to spend most of its time in a low-power sleep mode in which all nonessential systems are either shut down or put into low-power modes themselves. Errors during the logging cycle are caught by the watchdog timer, which reboots the ALog if it hangs and writes a timestamp to a file to help in uncovering the source of the error. The user interface comprises status and sensor messages passed to the serial monitor as well as a set of status messages flashed by the LED.

Table 3. Measurements and sensors currently supported by the ALog. Support for additional sensors may be added to the open-source ALog firmware library (Wickert et al., 2018a), following the _sensor_function_template example, which may be found in the design files and reference manual in the Supplement.

Property	Sensor	Communications
Temperature	Thermistor	Analog R
Temperature	BMP280	Analog R
Soil moisture	Dielectric probe	Analog V, UART
Rainfall	Tipping-bucket rain gauge	Interrupt
Wind speed	Cup anemometer	Interrupt
Wind direction	Magnetic wind vane	Analog R
Distance	Ultrasonic rangefinder	Analog V, UART
Distance	Linear potentiometer	Analog R
Absolute pressure (atmos.)	Digital barometer (BMP280)	I^2C
Absolute pressure (water)	Sealed pressure transducer	Analog V, I^2C
Solar radiation	Pyranometer with amplifier	Analog V
Relative humidity	Humidity probe	Analog V
Image or video	Camera trigger	Digital I/O
Groundwater temperature and flux	Thermal profiler	Analog R ($\times 6$)
Overland flow status	Binary conductivity sensor	Analog R
Angle or tilt	Two-axis inclinometer	Analog V ($\times 2$)
Force	Force-sensitive resistor	Analog R

The sensors component includes both private utility functions for standard operations (e.g., calculating standard deviations, solving the voltage-divider equation) and public functions that link each analog or digital sensor interface to the ALog and record data. Sensor functions are modular and written following a standard inputs–processing–outputs template. Outputs are written to the SD card as plain text ASCII data and header files and are printed to a serial monitor if the data logger is connected to a computer. Current support exists for a broad range of off-the-shelf sensors (Table 3), many of which are inexpensive. Users may add additional sensor support to the ALog library with help from a function template (design files in the Supplement) and the documentation (reference manual in the Supplement). Users may then contribute their code for additional sensor support to the main ALog repository, thereby increasing the reach of open-source instrumentation.

Sensors may be read on a standard interval or in response to an event. When reading measurements at a standard interval (typically 1–10 min), the RTC wakes the ALog using an interrupt. Once awake, the ALog retrieves data from all sensors, which usually measure environmental states. Reading and recording data from these sensors typically takes 1–3 s, during which the ALog is operating in its high-power "awake" state (Table 1). An event-based impulse, such as that from a tipping-bucket rain gauge, instantaneously wakes the ALog, which then records a timestamp to a different data file from that which is used for regular RTC-driven measurements. Reading and recording this timestamp typically requires < 0.3 s of awake-state power consumption. If the ALog is already awake (e.g., during RTC-driven data log-

ging) when an event occurs, the ALog firmware records the time of the event to its file and then continues the remainder of its ongoing task.

To program ALog data loggers, users import the ALog library into an Arduino sketch and instantiate the ALog class. Using examples included with the ALog library as a guide, users write a set of instructions that prescribe which sensors should be read and how often data should be recorded (Fig. 3). This sketch is then compiled and uploaded to the ALog as firmware.

2.3 Software

Arduino sketches to program the ALog may be written and uploaded using the Arduino IDE (Banzi and Shiloh, 2014), which evolved from Processing (Reas and Fry, 2007) and Wiring (Barragán, 2004). The Arduino IDE contains an interface to automatically download and install the custom ALog hardware definitions files (including the appropriate bootloaders) and code libraries. It also includes a serial monitor to view communications between the ALog and the computer.

The ALog clock is set via the USB–serial connection using a digital handshake programmed within the ALog library. Two options to set the clock are available: a command-line serial interface program written in Python (Wickert, 2017) and a graphical program written in Processing (Schulz, 2018) (see design files in the Supplement). Both methods interact with the ALog immediately upon boot-up (see Fig. 3).
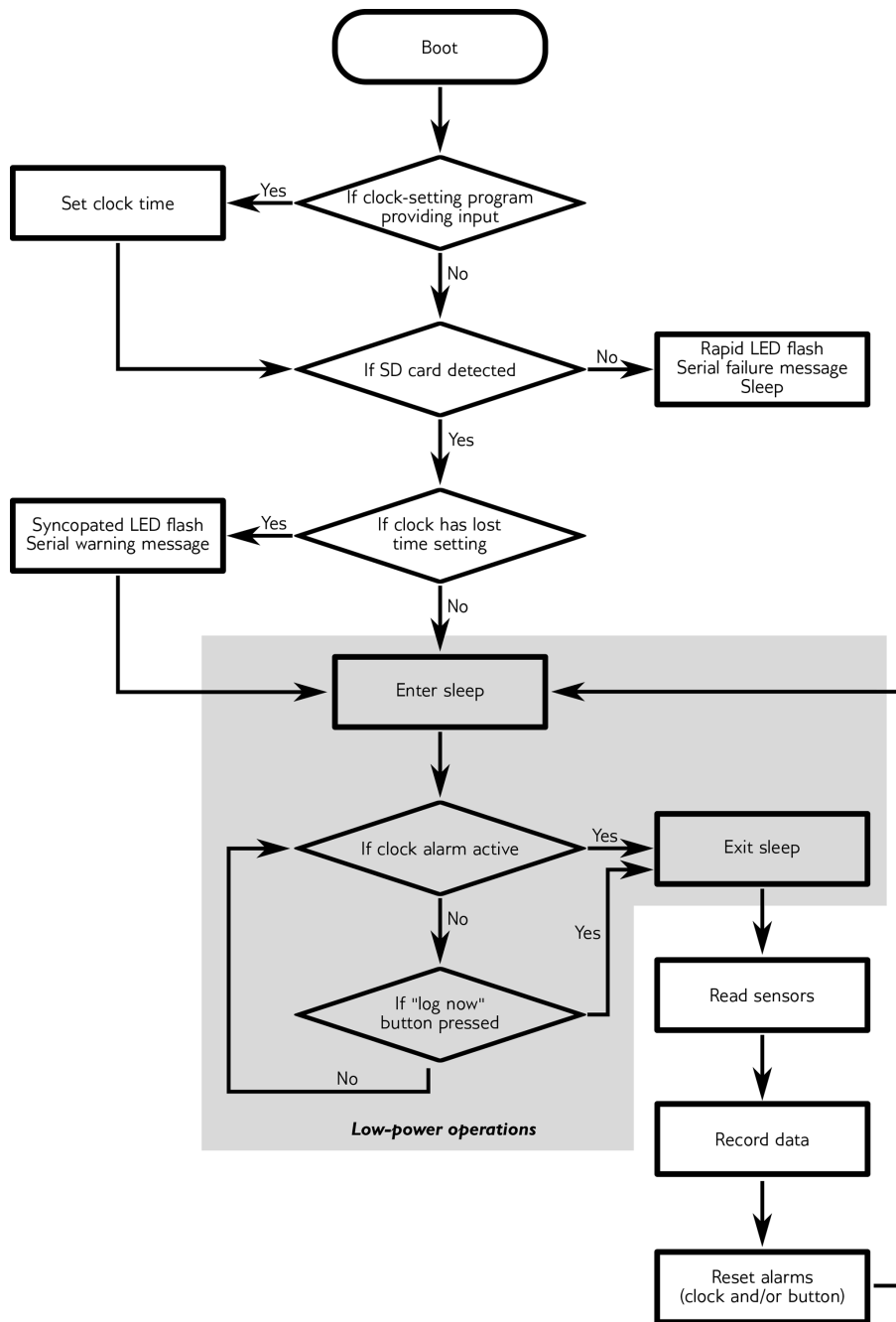
Hydrol. Earth Syst. Sci., 23, 2065–2076, 2019

www.hydrol-earth-syst-sci.net/23/2065/2019/

Figure 3. Flowchart of ALog operations. This set of steps is prescribed by the firmware. Not pictured is the watchdog timer, which resets the data logger, returning it to the "boot" step, if it hangs for more than 8 s.

3 Field deployment

ALog development evolved iteratively over more than 8 years of lab design and field deployments (Fig. 4). ALog data loggers have been deployed in the high alpine (Niwot Ridge, Colorado, USA), the high desert (Quebrada del Toro, Salta, Argentina), coastal wetlands (Wax Lake Delta, Louisiana, USA), subalpine valleys (Gordon Gulch, Col-

orado, USA), tropical mountains (Chimborazo volcano, also known as Volcán Chimborazo, Ecuador), continental lacustrine regions (Minnesota, USA; and Ontario, Canada), and on a large valley glacier (Kennicott Glacier, Alaska, USA) (Wickert, 2014; Armstrong et al., 2016; Tauro et al., 2018; Saberi et al., 2019). Field deployments ranged from a few days to 3 years. During these deployments, the ALog recorded data from weather stations, glacier-ablation mon-

Figure 4. Example ALog deployments. **(a)** Weather station with anemometer, pyranometer, wind vane, thermistor, and tipping-bucket rain gage, to measure high-desert climate parameters. **(b)** Glacier monitoring station with look-down ultrasonic sensor, thermistor, and relative humidity sensor, to monitor ablation and its drivers. **(c)** Downloading field data by copying and pasting an ASCII text file from the SD card. **(d)** Look-down ultrasonic sensor as a simple stream gauge; despite the destruction of the monitoring system in a historic flood event, the data file remained saved and uncorrupted on the SD card. **(e)** Solar radiation and wind speed covary at the station in **(a)**; wind speed lags radiation, indicative of surface-heating-driven convective winds in this high-altitude arid environment.

itoring stations, thermistors, stream gauges, soil moisture probes, pressure transducers for water levels in wells, subsurface temperature profilers, and frost-heave gauges; Table 3 contains a full list of sensors for which firmware has been developed and included in the ALog library (Wickert et al., 2018a). These field deployments tested the ALog data logger in humid and dry environments, onshore and offshore, and in temperatures that ranged from −30 to +35 °C.

The final ALog designs were guided as much by failure as by success. When the ALog failed in the field, it was typ-ically due to (1) moisture intrusion, (2) failure to properly seat the SD card, (3) loss of power to the real-time clock that caused it to reset its date and time to midnight on the morning of 1 January 2000, or (4) poorly written firmware. Moisture intrusion was managed by improving the enclosures (see below). To ensure that the SD card was seated properly, we established a protocol of pressing the reset button upon reinsertion and waiting for a "long–short–short" flash of the indicator LED (see the reference manual in the Supplement). Real-time clock failures are denoted by a syncopated flash on the indicator LED. This flash pattern notifies the user to set the RTC, and the ALogTalk software (Wickert, 2017) records a set of five measurements of the computer time and logger time before setting the RTC. These linked timestamps help the user to manually correct the timestamps on data that were recorded after the clock reset to its factory-default time of midnight on 1 January 2000. To recover from firmware errors that could cause failures over times long enough that we would not observe them in the lab, we enabled a watchdog timer to reset the ALog if it hung for 8 s. On each watchdog-timer reset, the logger writes a timestamp to a file in order to help with data QA/QC and to assist in future debugging.

3.1 Enclosures

Choosing an appropriate enclosure is a key decision for equipment survival in the field. The "BottleLogger" moniker comes from its designed form factor that allows it to fit inside a wide-mouth Nalgene bottle. Such bottles seal well and are commonly available from suppliers of both laboratory and outdoor equipment. This design feature was created as an option for occasions when easier-to-use but harder-to-souce enclosures were not available. In the majority of our deployments, we have used acrylonitrile butadiene styrene (ABS) plastic enclosures (models NBF-32104 and NBF-32108). These boxes are gasketed, include lever-style clips for easy opening and closing in the field, are large enough for either 3 × D or 3 × AA cells, and may be easily drilled or machined to accommodate cable glands for connections to sensors. The shorter enclosure (NBF-32104) requires a right-angle barrel jack plug in order to fit the length of the logger, SD card, and power connector.

We typically attach the loggers to the lid of the enclosure and the battery pack to the bottom of the enclosure using self-adhesive hook-and-loop fasteners. This holds both in place but allows either to be easily removed for wiring. The longer boxes (NBF-32108) permit cable glands to be drilled in the lid next to the logger, reducing the need for cable strain relief. The shorter boxes (NBF-32104) may include cable glands in one or more sides, and this is easier if they are fitted with a 3 × AA cell pack (as opposed to a 3 × D cell pack).
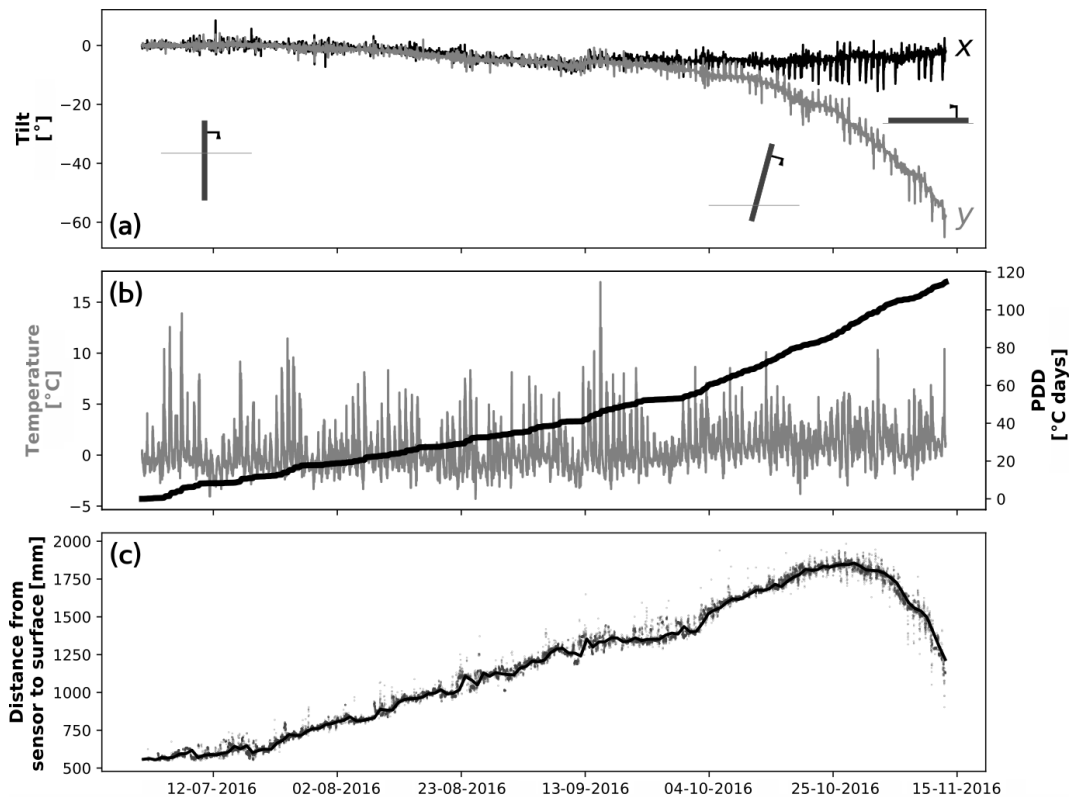
Figure 5. Automated ablation stake data. Figure 4b shows an example of an identical ablation stake deployed elsewhere on Volcán Chimborazo (Saberi et al., 2019). **(a)** Inclinometer output as the snow and ice melt, eventually causing the ablation stake to tilt and fall over as pictured in the cartoon drawings. **(b)** Temperature (gray) and cumulative positive-degree days (black). **(c)** Vertical distance from the ultrasonic rangefinder to the surface as single-time measurements (gray semitransparent points) and a daily moving average (black line). The rollover indicates the point at which the ablation stake tilt begins to dominate the signal, and until this point ablation (i.e., increasing distance) generally tracks the positive-degree-day line.

3.2 Examples

We highlight two hydrologically relevant example deployments performed at Volcán Chimborazo, Ecuador, following on the work of La Frenierre (2014) and La Frenierre and Mark (2017). In the first deployment, we measure weather conditions; in the second, we measure glacier ablation and its drivers. In both cases, the ability of the ALog to communicate with multiple sensors from different manufacturers allows these stations to record relevant data to better understand mountain hydrology in the glacierized Andes.

On the arid eastern side of Volcán Chimborazo, we installed an ALog BottleLogger connected to sensors for wind speed (Inspeed Vortex anemometer), wind direction (Inspeed e-Vane), and solar radiation (Kipp and Zonen CMP3 pyranometer linked with our in-house-designed instrumentation amplifier). We affixed these sensors and the ALog BottleLogger to an existing structure that was used to measure rainfall and temperature using a proprietary data-logging system (Fig. 4a) (La Frenierre, 2014; La Frenierre and Mark, 2017).

Figure 4e contains the first 5 d of data from this deployment, which lasted 1 year.

We installed a prototype automated ablation stake on Reschreiter Glacier on the more humid eastern flank of Volcán Chimborazo (Fig. 4b). We designed this automated stake to measure both the atmospheric factors that drive snow and ice ablation and the amount of snow and/or ice melt that occurs. Atmospheric variables include temperature, measured using a CanTherm epoxy-embedded thermistor paired with a reference resistor, and humidity, measured with a TE Connectivity HM1500LF sensor; both of these sit within a solar radiation shield (Fig. 5b). Distance to the snow and ice surface is measured using a MaxBotix ultrasonic rangefinder (Fig. 5a) (see Wickert, 2014), which is paired with a digital inclinometer to check and correct for station tilt as the ablation stake gradually melts out of the snow and/or ice (Fig. 5a). This station, here programmed to record data every 5 min, dramatically increases ablation data density beyond traditional methods, which incorporate daily to weekly field surveys of snow and/or ice surface elevation change around ablation stakes. Furthermore, by including on-stake temper-

ature measurements, we are able to compute at-stake melt factors for degree-day melt models, which are significant for both glaciological and water-resources research (Saberi et al., 2019).

4 Discussion

The paradigm of global change research has been one of scientists studying, reporting, predicting, and communicating how human activities impact the environment (Syvitski et al., 2009; Foley et al., 2011; Pelletier et al., 2015; Tauro et al., 2018), ideally followed by the broader public responding with plans to better manage Earth's environment and natural resources. In order to develop the ALog, we reversed this flow of information by drawing on open-source hardware, firmware, and software designs from the public to develop a scientific tool (Cressey, 2017). The open-source electronics movement has grown rapidly as part of the "maker revolution", in which individuals develop new technology and share their designs (Buechley et al., 2008; Anderson, 2012; Libow Martinez and Stager, 2013; Hut et al., 2016). We predict that building atop a broad and popular base platform will increase public accessibility to and interest in scientific measurements, as well as improve the support for and longevity of the data logger technology.

The ALog is part of a community of open-source tools for scientific research (Harnett, 2011; Pearce, 2012; Cressey, 2017) that includes both sensors (Keeler and Brugger, 2012; Barnard et al., 2014; Fatehnia et al., 2016; Hut et al., 2016) and automated data loggers (Fisher, 2012; Hund et al., 2016; Beddows and Mallon, 2018). Of these, the ALog BottleLogger designs have the lowest power consumption (Table 1). Their screw terminals easily communicate with any sensor via multiple methods of analog and digital communication, and their fully integrated and documented firmware (Fig. 2) reduces end-user coding to a few lines. These features have been included individually in other firmware and hardware designs (Hund et al., 2016; Aufdenkampe et al., 2017; Damiano et al., 2019), but the ALog incorporates all of these into a single streamlined system. All hardware schematics and software are regularly updated and available from GitHub in formats that can be read by free software. These design decisions are essential to building a broad and active user base that can work collectively to increase environmental monitoring across the globe using high-quality open-source technologies.

5 Conclusions

We developed the ALog system of open-source data-logging hardware, firmware, and software; built compatibility for a wide range of sensors; and tested the designs extensively in the field. Multiple design iterations over more than 8 years of development decreased power consumption, improved field

usability, and led to a simple library-based system to streamline programming into one-line function calls. We created the ALog with the help of community-developed open-source code and designs, and we hope that the ALog in turn can be a stepping stone to even more advanced, usable, and powerful open-source technology. By expanding the reach of open-source field instrumentation, we hope to help scientists and members of the broader community measure and understand our changing world.

References

Anderson, C. R.: Makers: The New Industrial Revolution, Crown Business, New York, New York, USA, 2012.

Armstrong, W. H., Anderson, R. S., Allen, J., and Rajaram, H.: Modeling the WorldView-derived seasonal velocity evolution of Kennicott Glacier, Alaska, J. Glaciol., 62, 763–777, https://doi.org/10.1017/jog.2016.66, 2016.

Aufdenkampe, A. K., Damiano, S. G., Hicks, S., and Horsburgh, J. S.: EnviroDIY ModularSensors: A Library to give Environmental Sensors a Common Interface of Functions for use with Arduino-Compatible Dataloggers, in: American Geophysical Union, Fall Meeting, H41J–1579, American Geophysical Union, New Orleans, 14 December 2017, LA, USA, 2017.

Ayars, E. and Wickert, A. D.: DS3231 v1.0.3, https://doi.org/10.5281/zenodo.2008622, 2018.

Banzi, M. and Shiloh, M.: Getting Started with Arduino: The Open Source Electronics Prototyping Platform, Maker Media, Inc., San Francisco, California, USA, https://doi.org/10.1017/CBO9781107415324.004, 2014.

Barnard, H. R., Findley, M. C., and Csavina, J.: PARduino: a simple and inexpensive device for logging photosynthetically active radiation, Tree Physiol., 34, 640–645, https://doi.org/10.1093/treephys/tpu044, 2014.

Barragán, H.: Wiring: Prototyping Physical Interaction Design, PhD thesis, Interaction Design Institute, Ivrea, available at: http://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf (last access: 24 April 2019), 2004.

Beddows, P. A. and Mallon, E. K.: Cave Pearl Data Logger: A Flexible Arduino-Based Logging Platform for Long-Term Monitoring in Harsh Environments, Sensors, 18, 530, https://doi.org/10.3390/s18020530, 2018.

Bogen, A.-E. and Wollan, V.: AVR Enhanced RISC Microcontrollers, Tech. rep., ATMEL Development Center, Trondheim, Norway, 1999.

Buechley, L. and Eisenberg, M.: The LilyPad Arduino: Toward Wearable Engineering for Everyone, IEEE Pervas. Comput., 7, 12–15, https://doi.org/10.1109/MPRV.2008.38, 2008.

Buechley, L., Eisenberg, M., Catchen, J., and Crockett, A.: The LilyPad Arduino: Using Computational Textiles to Investigate Engagement, Aesthetics, and Diversity in Computer Science Education, in: Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems – CHI '08, p. 423, Florence, Italy, https://doi.org/10.1145/1357054.1357123, 2008.

Cadsoft and Autodesk: EAGLE, available at: https://www.autodesk.com/products/eagle, last access: 18 April 2019.

Cressey, D.: The DIY electronics transforming research, Nature, 544, 125–126, https://doi.org/10.1038/544125a, 2017.

Damiano, S., Aufdenkampe, A., and Arscott, D.: EnviroDIY/ModularSensors: v0.21.0: Support for all Atlas Scientific I2C sensors, compiler-safe begin functions, https://doi.org/10.5281/zenodo.2586200, 2019.

Fatehnia, M., Paran, S., Kish, S., and Tawfiq, K.: Automating double ring infiltrometer with an Arduino microcontroller, Geoderma, 262, 133–139, https://doi.org/10.1016/j.geoderma.2015.08.022, 2016.

Ferdoush, S. and Li, X.: Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications, Procedia Comput. Sci., 34, 103–110, https://doi.org/10.1016/j.procs.2014.07.059, 2014.

Fisher, D. K.: 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, 2012.

Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., Balzer, C., Bennett, E. M., Carpenter, S. R., Hill, J., Monfreda, C., Polasky, S., Rockström, J., Sheehan, J., Siebert, S., Tilman, D., and Zaks, D. P. M.: Solutions for a cultivated planet, Nature, 478, 337–342, https://doi.org/10.1038/nature10452, 2011.

Greiman, W.: sdfatlib: A FAT16/FAT32 Arduino library for SD/SDHC cards, available at: https://github.com/greiman/SdFat (last access: 18 April 2019), 2016.

Harnett, C.: Open source hardware for instrumentation and measurement, IEEE Instru. Meas. Mag., 14, 34–38, https://doi.org/10.1109/MIM.2011.5773535, 2011.

Hart, J. K. and Martinez, K.: Environmental Sensor Networks: A revolution in the earth system science?, Earth-Sci. Rev., 78, 177–191, https://doi.org/10.1016/j.earscirev.2006.05.001, 2006.

Hicks, S., Aufdenkampe, A. K., Damiano, S., and Arscott, D.: EnviroDIY Mayfly Logger: v0.5b (Version v0.5b), https://doi.org/10.5281/zenodo.2572006, 2019.

Hicks, S. D., Aufdenkampe, A. K., Montgomery, D. S., Damiano, S., and Brooks, H.: A New Arduino Datalogger Board for Simple, Low Cost Environmental Monitoring and the EnviroDIY Web Community, in: American Geophysical Union Fall Meeting, H23G–1658, 15 December 2015, San Francisco, CA, USA, 2015.

Hirschfeld, T.: Instrumentation in the Next Decade, Science, 230, 286–291, https://doi.org/10.1126/science.230.4723.286, 1985.

Hund, S. V., Johnson, M. S., and Keddie, T.: Developing a Hydrologic Monitoring Network in Data-Scarce Regions Using Open-Source Arduino Dataloggers, Ael, 1, 160011, https://doi.org/10.2134/ael2016.02.0011, 2016.

Hut, R., Tyler, S., and van Emmerik, T.: Proof of concept: temperature-sensing waders for environmental sciences, Geosci. Instrum. Method. Data Syst., 5, 45–51, https://doi.org/10.5194/gi-5-45-2016, 2016.

Keeler, M. L. and Brugger, K. A.: A method for recording ice ablation using a low-cost ultrasonic rangefinder, J. Glaciol., 58, 565–568, https://doi.org/10.3189/2012JoG11J153, 2012.

La Frenierre, J. and Mark, B. G.: Detecting Patterns of Climate Change at Volcán Chimborazo, Ecuador, by Integrating Instrumental Data, Public Observations, and Glacier

www.hydrol-earth-syst-sci.net/23/2065/2019/

Hydrol. Earth Syst. Sci., 23, 2065–2076, 2019

Change Analysis, Ann. Am. Assoc. Geogr., 4452, 1–19, https://doi.org/10.1080/24694452.2016.1270185, 2017.

La Frenierre, J. D.: Assessing the Hydrologic Implications of Glacier Recession and the Potential for Water Resources Vulnerability at Volcán Chimborazo, Ecuador, PhD thesis, The Ohio State University, Columbus, Ohio, USA, 2014.

Libow Martinez, S. and Stager, G. S.: Invent to learn: Making, tinkering, and engineering in the classroom, Constructing Modern Knowledge Press, Torrance, CA, USA, 2013.

Lovett, G. M., Burns, D. A., Driscoll, C. T., Jenkins, J. C., Mitchell, M. J., Rustad, L., Shanley, J. B., Likens, G. E., and Haeuber, R.: Who needs environmental monitoring?, The Ecological Society of America, 5, 253–260, https://doi.org/10.1890/1540-9295(2007)5[253:WNEM]2.0.CO;2, 2007.

Martinez, K., Hart, J., and Ong, R.: Environmental sensor networks, Computer, 37, 50–56, https://doi.org/10.1109/MC.2004.91, 2004.

Oliveira, L. M. and Rodrigues, J. J.: Wireless Sensor Networks: a Survey on Environmental Monitoring, J. Commun., 6, 143–151, https://doi.org/10.4304/jcm.6.2.143-151, 2011.

Padhy, P., Martinez, K., Riddoch, A., Ong, H. L. R., and Hart, J. K.: Glacial Environment Monitoring using Sensor Networks, in RealWSN'05, 10–14, Stockholm, Sweden, 2005.

Pearce, J. M.: Building Research Equipment with Free, Open-Source Hardware, Science, 337, 1303–1304, https://doi.org/10.1126/science.1228183, 2012.

Pelletier, J. D., Brad Murray, A., Pierce, J. L., Bierman, P. R., Breshears, D. D., Crosby, B. T., Ellis, M., Foufoula-Georgiou, E., Heimsath, A. M., Houser, C., Lancaster, N., Marani, M., Merritts, D. J., Moore, L. J., Pederson, J. L., Poulos, M. J., Rittenour, T. M., Rowland, J. C., Ruggiero, P., Ward, D. J., Wickert, A. D., and Yager, E. M.: Forecasting the response of Earth's surface to future climatic and land-use changes: A review of methods and research needs, Earth's Future, 3, 220–251, https://doi.org/10.1002/2014EF000290, 2015.

Reas, C. and Fry, B.: Processing: a programming handbook for visual designers and artists, MIT Press, Cambridge, MA, USA, 2007.

Reda, H. T., Daely, P. T., Kharel, J., and Shin, S. Y.: On the Application of IoT: Meteorological Information Display System Based on LoRa Wireless Communication, IETE Tech. Rev., 4602, 1–10, https://doi.org/10.1080/02564602.2017.1279988, 2017.

Saberi, L., McLaughlin, R. T., Ng, G.-H. C., La Frenierre, J., Wickert, A. D., Baraer, M., Zhi, W., Li, L., and Mark, B. G.: Multiscale temporal variability in meltwater contributions in a tropical glacierized watershed, Hydrol. Earth Syst. Sci., 23, 405–425, https://doi.org/10.5194/hess-23-405-2019, 2019.

Sandell, C. T., Schulz, B., and Wickert, A. D.: ALog BottleLogger v3.0.0, https://doi.org/10.5281/zenodo.2008637, 2018.

Schulz, B.: SetTime GUI v1.0.0, https://doi.org/10.5281/zenodo.2008618, 2018.

Syvitski, J. P. M., Kettner, A. J., Overeem, I., Hutton, E. W. H., Hannon, M. T., Brakenridge, G. R., Day, J., Vörösmarty, C., Saito, Y., Giosan, L., and Nicholls, R. J.: Sinking deltas due to human activities, Nat. Geosci., 2, 681–686, https://doi.org/10.1038/ngeo629, 2009.

Tauro, F., Selker, J., van de Giesen, N., Abrate, T., Uijlenhoet, R., Porfiri, M., Manfreda, S., Caylor, K., Moramarco, T., Benveniste, J., Ciraolo, G., Estes, L., Domeneghetti, A., Perks, M. T., Corbari, C., Rabiei, E., Ravazzani, G., Bogena, H., Harfouche, A., Brocca, L., Maltese, A., Wickert, A., Tarpanelli, A., Good, S., Lopez Alcala, J. M., Petroselli, A., Cudennec, C., Blume, T., Hut, R., and Grimaldi, S.: Measurements and Observations in the XXI century (MOXXI): innovation and multi-disciplinarity to sense the hydrological cycle, Hydrolog. Sci. J., 63, 169–196, https://doi.org/10.1080/02626667.2017.1420191, 2018.

van Heesch, D.: Doxygen: Source code documentation generator tool, available at: http://www.doxygen.nl/ (last access: 18 April 2019), 2008.

Vitousek, P. M.: Beyond Global Warming: Ecology and Global Change, Ecology, 75, 1861, https://doi.org/10.2307/1941591, 1994.

Wickert, A. D.: The ALog: Inexpensive, Open-Source, Automated Data Collection in the Field, Bulletin of the Ecological Society of America, 95, 68–78, https://doi.org/10.1890/0012-9623-95.2.68, 2014.

Wickert, A. D.: ALogTalk v1.0.0, https://doi.org/10.5281/zenodo.2008620, 2017.

Wickert, A. D. and Sandell, C. T.: ALog BottleLogger v2.2.0, https://doi.org/10.5281/zenodo.2008633, 2017.

Wickert, A. D., Sandell, C. T., Barnhart, K. R., and Schulz, B.: ALog v1.0.0, https://doi.org/10.5281/zenodo.2008610, 2018a.

Wickert, A. D., Sandell, C. T., and Schulz, B.: ALog-Shield v2.2.0, https://doi.org/10.5281/zenodo.2008624, 2018b.

Wollan, V., Bogen, A.-E., Myklebust, G., and Bryant, J. D.: Eight-bit microcontroller having a RISC architecture, 1998.