



# Easy to build low-power GPS drifters with local storage and a cellular modem made from off-the-shelf components

Rolf Hut<sup>1</sup>, Thanda Thatoe Nwe Win<sup>2</sup>, and Thom Bogaard<sup>2</sup>

<sup>1</sup>Chair of Water Resources Engineering, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CH, Delft, the Netherlands

<sup>2</sup>Chair of Hydrology, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CH, Delft, the Netherlands

**Correspondence:** Rolf Hut (r.w.hut@tudelft.nl)

Received: 14 August 2019 – Discussion started: 7 October 2019

Revised: 8 September 2020 – Accepted: 28 September 2020 – Published: 5 November 2020

**Abstract.** Drifters that track their position are important tools in studying the hydrodynamic behavior of rivers. Drifters that can be tracked in real time have so far been rather expensive. Recently, due to the rise of the open-hardware revolution and the associated Arduino ecosystem, both GPS receivers and cellular modems have become available at lower prices to “tinkering scientists”, i.e., scientists that like to build their own measurement devices as much as is possible. This article serves two goals. Firstly, we provide detailed instructions on how to build a low-power GPS drifter with local storage and cellular model that we tested in a fieldwork on the confluence of the Chindwin and Ayeyarwady rivers in Myanmar. The device was designed from easily connected off-the-shelf components, allowing construction without a background in electrical engineering. The instructions allow fellow geoscientists to recreate the device. Secondly, we set the following question: has the open-hardware revolution progressed to the point that a low-power GPS drifter that wirelessly transmits its position can be made from open-hardware components by most geoscientists? We feel this question is relevant and timely as more low-cost open-hardware devices are promoted, but in practice applicability is often restricted to the “tinkering engineer”. We argue that because of the plug-and-play nature of the components geoscientists should be able to construct these type of devices. However, to get such devices to operate at low power levels that fieldwork often demands requires detailed (micro)electrical expertise.

## 1 Introduction

Drifters, devices that naturally float and follow the surface flow lines in a moving water body, are a method to measure important parameters, such as surface water velocity distributions, breakthrough curves, and mixing parameters. In drifter research it is important to have as many drifters as possible to optimally measure flow patterns (Tinka et al., 2010). Drifters measure their location almost exclusively using GPS technology and store it on the device or transmit it using a network connection. Currently, drifters are made for specific research questions and are either expensive to purchase or are built in-house, requiring detailed knowledge of electrical engineering and embedded software (Network, 2012; Tinka et al., 2013). In general, drifters consist of a location device (often GPS), a logger to store the data, and/or a communication unit. Recent developments, most notably the rise of the “maker movement” (Chris, 2012) and its ecosystem based on Arduino, Raspberry Pi, and other open-hardware development boards (Banzi and Shiloh, 2014; Foundation, 2018), have made the individual components of a GPS drifter commercially available as plug-and-play components for hobbyists and developers. This has led scientists to build GPS loggers geared to their own research need at lower costs. Daniel and Peter (2012) gave an early overview of the possibilities of the Arduino ecosystem as a low-cost alternative for expensive scientific instrumentation. Wickert (2014) developed the ALOG, a low-power Arduino-inspired logger system specifically designed with geoscientific fieldwork in mind.

Relating to using open hardware to track GPS positions, Cain and Cross (2018) made a low-power GPS logger to track eastern box turtles. Their device stores the data in the local memory of the Arduino microcontroller, requiring retrieval after the experiment and providing no real-time view on where the turtles are. Pham et al. (2013) and Costanzo (2013) both connected a GPS receiver and a cellular modem to an Arduino to track locations of vehicles. In vehicle tracking sufficient power is usually available through the battery of the car. GPS drifters are often used in near-shore applications. Drifters are often designed to store information locally, either by using a completely off-the-shelf GPS logger such as that of Sabet and Barani (2011) or by building one from open-hardware components such as those of Lockridge et al. (2016), and Johnson et al. (2003). For applications where drifters can not be tracked visually for pick-up, wireless solutions are sometimes used. Austin and Atkinson (2004) added the ability to communicate with the drifters over local radio to keep track of their location for easy pick-up. Trevathan et al. (2012) custom-built a suite of near-shore buoys that together act as a wireless sensor network (WSN). Tinka et al. (2013) gives an overview of work on drifters (not focusing on low-cost examples) in their introduction and subsequently present their high-end, custom-designed floating sensor network. For open-ocean tracking Cadena et al. (2018) adds an iridium satellite modem to their drifter.

For fieldwork on the Ayeyarwady and Chindwin rivers in Myanmar, we were in need of a low-cost, low-power GPS drifter that stores its data locally and at the same time transmits its location in real time for retrieval of the device. Given the length scales involved (tens of kilometers of sampling along the river per day) drifters would float out of range of base stations. However, since we are conducting our fieldwork on a stretch of river with ample cellular coverage, using a satellite modem would be costly overkill. In this article we present, to the best of our knowledge, the first complete open-hardware GPS drifter design that adds cellular communication to relay its location in real time. It features an (open-source) online back-end for both real-time tracking and data collection.

All of the projects cited above, while built using low-cost open-hardware components, were either too expensive to purchase for our project or required extensive engineering. In this article we furthermore set out to test the following question: has the open-hardware revolution progressed to the point that a low-power GPS drifter that wirelessly transmits its position can be made from open-hardware components by most geoscientists? The drifter presented in this paper was deliberately designed from easily connected off the shelf components, allowing for easy construction.

This article will explain how we build such a device, detail the software designed, and present the raw measurement results from the fieldwork on the Ayeyarwady river. The detailed analyses of the data falls beyond the scope of this article, which focuses on the development of the data collection

devices, but is presented in the MSc thesis of Bakker (2017). We conclude by reflecting on the primary research question about the "build-ability" of our device by geoscientists in the discussion and conclusions.

## 2 Materials

The drifter design can be separated in two parts. Firstly the drifter itself, i.e. the material that makes the drifter float optimally given the local flow conditions of the field site. Secondly the electronics that do the GPS localization and communication. We describe the electronics first and the drifter design afterwards.

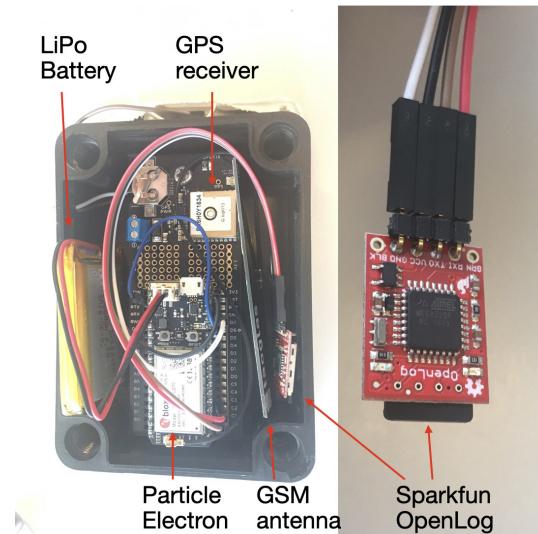
### 2.1 Electronic hardware

The design criteria for our device were to develop a system that could perform the following tasks:

- track its own location using GPS,
- upload its location to the internet to both save the measurement data and to be able to locate the device in real time during the measurement campaign,
- store the locations locally in case of no connection to the internet,
- operate continuously for at least 2 d without losing power.

We decided to buy a ready-made solution for each design criteria and connect those together. The electronics that matched our design criteria that we chose are as follows.

- *A particle electron.* A development board with a built-in cellular modem. The electron comes with a data subscription with near-global coverage and an online cloud back-end. Users can program the electron with their own code using an online development environment (Particle, 2018) in the C++ dialect that is also used for the Arduino ecosystem (Arduino, 2018). The particle electron is open-hardware Particle-Iot/Electron (2020) and the operating system running on the device is open-source software Particle-Iot/Device-Os (2020).
- *A particle asset tracker shield.* An extension to the particle electron that includes a GPS receiver and an accelerometer. The particle electron connects to this shield using a set of headers on the shield. Communication between the electron and the GPS receiver is done over UART serial communication using the Tx and Rx pins of the electron.
- *A Sparkfun OpenLog.* A board that records any data sent to it using UART serial communication onto an SD card. While the particle electron does have 80 kb of



**Figure 1.** Photo of the electronics components. The left photo identifies the Asset Tracker, Electron and OpenLog. The right photo shows the OpenLog in detail.

EEPROM memory that could be used for storing measurements, adding a removable SD card makes it easy for any team member to offload the measurement data during fieldwork.

- An 8 cm × 12 cm solar panel.

The physical layout of the components in the enclosure can be seen in Fig. 1. Table 1 list online locations where the components were purchased and prices at the time of writing. All data sheets with detailed technical specifications of the components are available in the Supplement. Connection of the components to each other is straightforward and given in Table 2.

## 2.2 Software and online back-end

The software required to operate the GPS drifters is divided into two parts: the software that runs on the particle electron and the software that runs online. All code used is available on GitHub and has been archived on Zenodo (Hut, 2018). This separation between online and firmware was made to facilitate reuse by researchers that want to build on one but not both of these parts.

Just as with its spiritual predecessor the Arduino, the particle electron stores a single program in its program memory, and this program is executed continuously as soon as the particle electron is powered. The particle electron is programmed in the same C++ dialect as is used for the Arduino. Ready-made examples exist that let the particle electron and the asset tracker shield upload GPS locations on a regular basis to the particle cloud service. However, these programs keep the cellular connection open continuously, which drains too much power for our application. The code

**Table 1.** Price and online location of information on off-the-shelf electronic components used to build the low-power GPS drifters with local storage and a cellular modem.

Name	Price	Online location
Particle asset tracker	USD 109	<a href="https://www.particle.io/products/hardware/asset-tracker">https://www.particle.io/products/hardware/asset-tracker</a> (last access: 8 September 2020)
Particle electron	USD 14.95	<a href="https://www.particle.io/products/electron-cellular-dev-kit">https://www.particle.io/products/electron-cellular-dev-kit</a> (last access: 8 September 2020)
SparkFun OpenLog	USD 14.95	<a href="https://www.sparkfun.com/products/13712">https://www.sparkfun.com/products/13712</a> (last access: 8 September 2020)
Conrad solar panel	EUR 14.95	<a href="https://www.conrad.nl/ml/polykristallijn-zonnepaneel-09-wp-6-v-110-54.html">https://www.conrad.nl/ml/polykristallijn-zonnepaneel-09-wp-6-v-110-54.html</a> (last access: 8 September 2020)

**Table 2.** Electrical connections between the particle asset tracker and the other components. Note that the connection between ground (GND) and OpenLog Black (BLK) can be made by connecting the OpenLog pins BLK and GND, saving the need for one wire. The connection between pin D6 and GND is to force the GPS unit on. See Fig. 1 for an image of the physical setup.

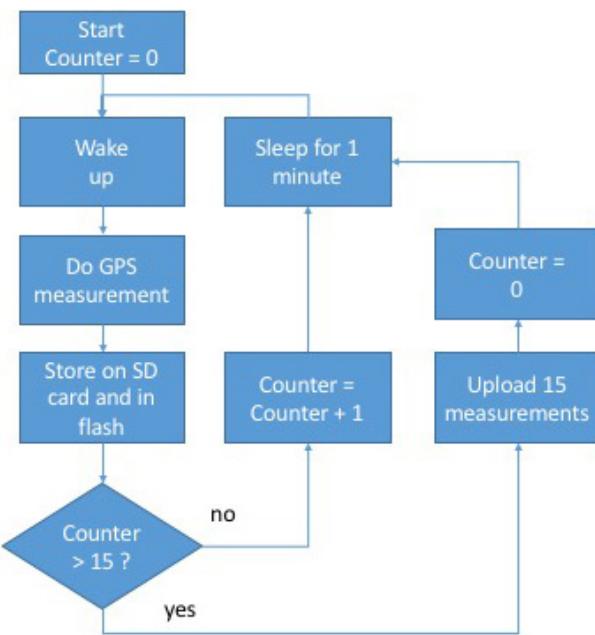
Pin on asset tracker	Connected to
Power terminal –	Solar panel –
power terminal +	Solar panel +
C2	OpenLog TX0
C3	OpenLog RX1
GND	OpenLog GND
GND	OpenLog BLK
3V3	OpenLog VCC
D6	Asset tracker GND

was heavily modified to optimize for minimal power consumption. Figure 2 shows the architecture of the code that runs on the particle electron. Every minute the device wakes from a low-power deep sleep mode and records the GPS coordinates from the asset tracker shield. This is stored both in the volatile memory of the device as well as on the SD card in the OpenLog. Once every 15 min the device will try to connect to the cellular network. If successful, it will upload the 15 latest GPS measurements to the particle cloud. The code is provided on GitHub and archived on Zenodo (Hut, 2018).

Figure 3 shows how the data moves from the particle electron via the particle cloud service onto a personal web server. A webhook running on the particle cloud service parses the messages from the particle electron and posts it to the “receiveData.php” file on the server. This file first stores the raw data in a database as a backup measure. Following this, it parses the message into the individual GPS measurements and stores these in the database labeled “Processed Data”. Users can look at this data in two ways. By opening the file dataDump.php in a browser, a .csv dump from the database is generated. This allows researchers to perform analyses on the measurements after the field campaign. When real-TimeMap.html is opened, a single device can be selected from a drop-down box. Once selected, a map with the latest measurements from that device is shown. This allows researchers to track the devices in real time. All source code that runs on the personal server, including instructions how to install the particle webhook, is provided on GitHub and archived on Zenodo (Hut, 2018).

### 2.3 Power consumption

A critical design criteria for our devices is that they should be able to run continuously on the combination of solar panel and battery. This is not an uncommon constraint for equipment in geoscientific applications. The logging of battery



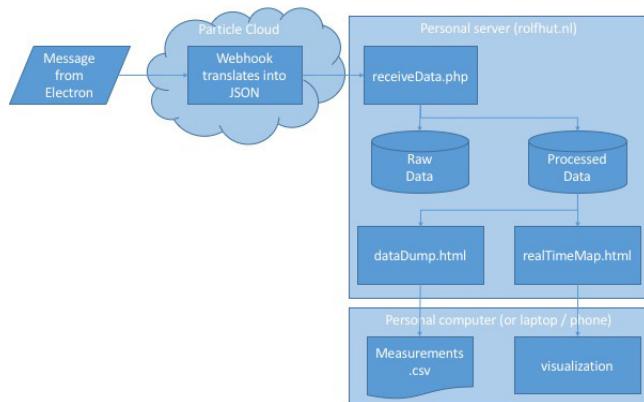
**Figure 2.** A flowchart of the software that runs on the particle electron. The full code is available on GitHub (Hut, 2018). In situations where limited power is not an issue, this software can be greatly simplified by removing the part where the electron goes into power-saving deep sleep mode. For our case study, however, low power usage was essential to a successful field campaign.

level is an additional feature built into the software running on the particle electron. We implemented a deep sleep cycle, where the microcontroller unit (MCU) goes into a deep sleep mode, which uses less than 200  $\mu$ A between taking measurements from the GPS module. In deep sleep, the GPS unit remains active because it can take up to 20 min to achieve a fix on the GPS satellites, a process that takes considerably more power than maintaining this fix.

We based our decisions relating to power consumption on the specified data sheets of the hardware involved, others, such as Kumar (2016), have verified that the actual power consumption is close enough to data sheet specifications for design decisions. Testing in the field showed that the device functions for at least 2 d on a 1200 mAh battery. The addition of the chosen solar panel extends this to about 4 d, depending on local solar conditions. For applications where longer periods of measurement are needed, adding batteries to the particle electron is straightforward. Given the low power consumption, simple batteries can easily extend the measurement period by several days.

### 2.4 Physical construction

The particle asset tracker comes delivered with a waterproof enclosure. All electrical components except the solar panel fit inside this enclosure. A small hole was drilled into the enclosure to feed the connections from the solar panel to the par-



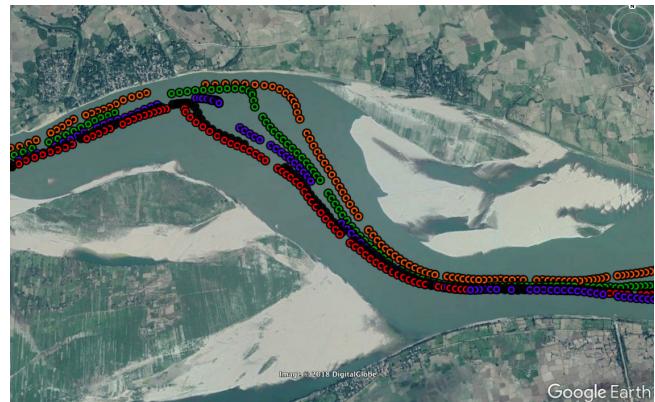
**Figure 3.** The flow of data once it leaves the particle electron. The electron sends the measurements as a comma-separated message to the particle cloud. A webhook running on the particle cloud parses the raw message into a JSON message that is posted to a .php file on the personal server of the main author. This file adds the data to two databases: one to store all raw messages, and one that parses the message into the individual measurements. Two .html files can be called upon by users. DataDump.html transforms the database into a comma-separated file for downloading, and realTimeMap.html shows the last known locations of the drifters for real-time localization during the field campaign. All software is available on GitHub (Hut, 2018).

ticle electron. The hole and the connectors on the solar panel were sealed with a two-component epoxy to make them waterproof. This was chosen over more conventional closures, such as hot-glue, because of its excellent thermal properties: once hardened it has negligible expansion when exposed to temperature differences.

We designed, based on local available material, a wooden plate with Styrofoam and five anchors to make sure the drifter followed the flow of the river. In our conditions, i.e., low wind conditions, this design worked adequately for our experiment, but of course the design of the floaters needs to be tailored based on local conditions and scientific questions.

### 3 Fieldwork: test in Ayeyarwady

Experiments were conducted both upstream of the confluence in the Ayeyarwady River and at the Chindwin–Ayeyarwady confluence. The Ayeyarwady river system is a very dynamic and naturally unregulated river, one of the largest rivers in the world, and a vital vein for Myanmar (SOBA, 2018). It is not only one of the main transport corridors but is also a source of fish, irrigation, and drinking water and an important aquatic ecosystem (SOBA, 2018). It is a shallow, dynamic river system, with a typical depth range of only 1–10 m, whereas it has a width of several 100 m. The migrating sandbanks, especially upstream and around the confluence with the Chindwin, are a very pressing problem for river management. Insight into hydraulic behavior of



**Figure 4.** The location measurements for the different floaters (different colors) in the Ayeyarwady river near Ngazun. The river flows west to east on this stretch, with downstream being the left side of the figure (© Google Earth, 2019).

the river is therefore a continuous demand. First of all, groups of five drifters were released at the same time in a line across the river. The purpose of this experiment was to validate a numerical study on the surface water flow velocity of the river. These analyses fall outside of the scope of this article and are presented in Bakker (2017), who conclude that “the floater paths around the confluence have been useful for validating the Delft3D model, as the floater speeds and paths observed in the field were compared to those of the Delft3D model”.

The drifters were followed both online and with a trailing boat. For logistical reasons, when all drifters became too dispersed along the river length (not width), they were taken out of the water and brought together to restart the experiment. If a single drifter started significantly lagging behind with respect to the rest of the group, it was retrieved.

The upstream Ayeyarwady experiment was executed with a single group of five drifters. The confluence experiment was executed with two groups of five drifters upstream of the Chindwin–Ayeyarwady confluence, with one group in each river. The drifters were tracked along the confluence. At the end of the experiment, drifters were collected and data were read out from both the SD cards of the OpenLog and from the website to which the data had been uploaded.

The raw data collected from the sensors are plotted on a map of the experiment area; see Fig. 5. No post-processing has been done on these results: they show the raw GPS locations as recorded by the sensors and downloaded from the online database. The figure has been made by loading these measurements in Google Earth. The different flow paths that the sensors followed can be clearly identified. These data being directly available online allowed the measurement team to identify trackers that had beached or were stuck in near-shore eddies. The full raw dataset is provided in the Supplement. These raw data will be used in follow-up research. Figure 4 furthermore shows a closeup of the



**Figure 5.** The location measurements for the different floaters (different colors) at the confluence of the Chindwin and Ayeyarwady rivers. Both rivers stream north to south on this stretch with downstream being the lower part of the figure (© Google Earth, 2019).

upstream Ayeyarwady experimental results. In this experiment the floaters were in the water for 16 h and 49 min and traveled 50 km in that time, with a distance of 4.6 km between the fastest and slowest drifter. Mean floater velocity over this stretch was  $0.8 \text{ m s}^{-1}$  (Bakker, 2017). The location of the drifters clearly shows the different flow paths in the river. Unfortunately, we do spot some gaps in the data. These are caused by lack of cell coverage. In these instances, the particle electron spent a long time attempting to connect, before deciding the switch to offline, leading to a loss of data. Based on the time between signals that were received, we estimate that around 10 % of the data were lost in this manner, although differences between different trackers were significant. Future versions of the software code running on the particle electron should switch to offline sooner when cell coverage drops to prevent loss of data. Finally, although clearly labeled as experimental equipment, some sensors were picked out of the water by citizens. This was easily identified in the online visualization (i.e., sensors moving upstream) and allowed the research team to chase and recover those sensors. These instances were manually removed from the dataset.

#### 4 Discussion

The intention of this research was to test if currently available plug-and-play sensors and communication solutions are such that most members of the geoscientific community can use them to build their own measurement equipment suited to the specifics of the measurement campaign. Our experience shows that many geoscientifically trained persons should be able to connect the different pieces together and build the basic drifters. However, to have the entire setup operate at

the required low power levels requires changing the standard programs that run on the device. The level of understanding of the electronic functioning of the device is not something that in general can be expected from a typical geoscientist, even one with extensive experience in using various types of electronic measurement equipment. Furthermore, developing the online back-end that visualizes the location of the GPS drifters in real time required back up from colleagues (see the Acknowledgements) with web development expertise. Aside from the design of the drifter, the construction of the trackers in Myanmar also required the creativity to use locally available materials as well as the capacity for the design to fine-tune and debug the tracker in the field. In our case, the code was changed such that if no cell signal was found, the logger would check again in 5 min instead of 1 min to minimize the data gaps (see above). Also, revitalizing a drowned tracker (due to collision with a commercial shipping vessel) needed engineering expertise. These type of experiences taught us that the real-world application of self-designed open-hardware equipment is not as straightforward as is sometimes envisioned. Here we establish that multi-disciplinary collaboration is still required when using low-cost self-designed equipment. Making the design more robust will lead to higher expenses.

Being able to trace the low-cost GPS drifters in real time proved invaluable during the fieldwork, for example in cases when curious locals picked up our drifters, as explained above. The real-time visualization allowed us to recover equipment.

#### 5 Conclusions

We presented a low-cost GPS drifter that communicates its data in real time over the cellular network. We presented our design such that fellow geoscientists can (re)build or modify our GPS drifters using this article. Our device was tested in extensive fieldwork in the Chindwin and Ayeyarwady rivers in Myanmar. During development we tested if such a device can be completely developed by a geoscientist from available plug-and-play components. Although creating an online-logging GPS drifter turned out to be doable without an electrical engineering background, it required considerable understanding of embedded programming to develop a version that was low enough power to be useful in our fieldwork environment. It would be extremely useful for the geoscientific community if low-power versions of the electronics used in this research were available “out of the box”. Several companies are building lower-power logging equipment for geoscientists Wickert (2014), but these are usually not yet usable without having to program them first.

The online environment that visualizes the fieldwork data in real time as they come in was developed by the main author and hosted on his personal website. It would be a great asset to geoscientists doing fieldwork if there was a platform

that both stores and visualizes real-time data coming in. Part of this functionality is built into the CUAHSI Hydroshare project Tarboton et al. (2014), although this focuses more on storing data for posterity than on real-time checking of data as they come in.

The advance of plug-and-play electronics has opened the opportunity for geoscientists to patch together the functionality they need in a sensor–logger communication device. We showed that for a GPS drifter with cellular connectivity and local storage this is indeed possible. However, getting plug-and-play devices to be low enough power for fieldwork conditions still requires considerable electrical engineering knowledge. We hope that this research provides a step in the right direction to help geoscientists use plug-and-play devices in general and our GPS drifters in particular, and we hope to encourage the companies producing these devices to bring low-power versions to market that work out of the box.

**Data availability.** All research data used in this publication are available in the Supplement.

**Supplement.** The supplement related to this article is available online at: <https://doi.org/10.5194/gi-9-435-2020-supplement>.

**Author contributions.** Author contributions are given using the CASRAI Credit system Allen et al. (2014). RH contributed to conceptualization, methodology, investigation, resources, software, validation, visualization, writing of the original draft, and reviewing and editing subsequent versions. TTNW contributed to methodology, investigation, project administration, and finally writing of the original draft (section fieldwork: test in Ayeyarwady). TB contributed to conceptualization, investigation, resources, supervision, and reviewing and editing of the manuscript.

**Competing interests.** The authors declare that they have no conflict of interest.

**Acknowledgements.** The authors would like to thank MSc and PhD students from YTU and MMU (Yangon, Myanmar), MTU (Mandalay, Myanmar), and TU Delft (the Netherlands) for their assistance in the field. Furthermore, we would like to thank MSc student Rick Hagenars for helping to develop the online back-end, proving the value of having students with a diverse backgrounds and skill sets in your team or community.

Finally, the authors would like to thank VPDelta for their support, particularly the Partner voor Water project: “Leapfrogging Delta Management in Myanmar – Showcase smart information solutions in the Ayeyarwady Delta” and NICHE-NMR-250 project “Capacity Development on Integrated Water Resources Management in Myanmar”.

**Financial support.** This research has been supported by VPDelta and NICHE-NMR-250 projects.

**Review statement.** This paper was edited by Flavia Tauro and reviewed by two anonymous referees.

## References

- Arduino: Arduino – Home, available at: <https://www.arduino.cc/> (last access: 8 September 2020), 2018.
- Austin, J. and Atkinson, S.: The Design and Testing of Small, Low-Cost GPS-Tracked Surface Drifters, *Estuaries*, 27, 1026–1029, <https://doi.org/10.1007/BF02803428>, 2004.
- Bakker, T.: Dispersion in the Ayeyarwady: A Description of the Mixing of Tracers in the Area of the Ayeyarwady River–Chindwin River Confluence, available at: <http://resolver.tudelft.nl/uuid:07a91b3a-7068-48fd-a74f-1f7906602832> (last access: 2 November 2020), 2017.
- Banzi, M. and Shiloh, M.: Getting Started with Arduino: The Open Source Electronics Prototyping Platform, Maker Media, Inc., Make Community, LLC, ISBN-13: 978-1449363338, 2014.
- Cadena, A., Vera, S., and Moreira, M.: A Low-Cost Lagrangian Drifter Based on Open-Source Hardware and Software Platform, in: 2018 4th International Conference on Control, Automation and Robotics (ICCAR), 218–221, <https://doi.org/10.1109/ICCAR.2018.8384673>, 2018.
- Cain, P. W. and Cross, M. D.: An Open-Source Hardware GPS Data Logger for Wildlife Radio-Telemetry Studies: A Case Study Using Eastern Box Turtles, *HardwareX*, 3, 82–90, <https://doi.org/10.1016/j.hwx.2018.02.002>, 2018.
- Chris, A.: Makers: The New Industrial Revolution, New York, Crown Business, Currency, ISBN-13: 978-0307720962, 2012.
- Costanzo, A.: An Arduino Based System Provided with GPS/GPRS Shield for Real Time Monitoring of Traffic Flows, in: 2013 7th International Conference on Application of Information and Communication Technologies, Baku, 1–5, <https://doi.org/10.1109/ICAICT.2013.6722710>, 2013.
- Daniel K., F. and Peter J., G.: 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.
- Foundation, R. P.: Raspberry Pi, available at: <https://www.raspberrypi.org> (last access: 8 September 2020), 2018.
- Hut, R.: RolfHut/Low-Power-GPS-Trackers-with-GSM-Modems-Firmware-and-Online-Backend v1.0, Zenodo, <https://doi.org/10.5281/zenodo.1485899>, 2018.
- Johnson, D., Stocker, R., Head, R., Imberger, J., and Patiaratchi, C.: A Compact, Low-Cost GPS Drifter for Use in the Oceanic Nearshore Zone, Lakes, and Estuaries, *J. Atmos. Ocean. Technol.*, 20, 1880–1884, [https://doi.org/10.1175/1520-0426\(2003\)020<1880:ACLGDF>2.0.CO;2](https://doi.org/10.1175/1520-0426(2003)020<1880:ACLGDF>2.0.CO;2), 2003.
- Lockridge, G., Dzwonkowski, B., Nelson, R., and Powers, S.: Development of a Low-Cost Arduino-Based Sonde for Coastal Applications, *Sensors*, 16, 528, <https://doi.org/10.3390/s16040528>, 2016.
- Network, F. S.: Floating Sensor Network, available at: <http://float.berkeley.edu/> (last access: 8 September 2020), 2012.

Kumar, S.: Particle: Electron Power Consumption Analysis, available at: <https://blog.voltaicsystems.com/particle-electron-power-consumption-analysis/> (last access: 8 September 2020), 2016.

Particle: Particle Build, available at: <https://build.particle.io/build-new> (last access: 8 September 2020), 2018.

Particle-Iot/Device-Os: Particle, available at: <https://github.com/particle-iot/device-os> (last access: 8 September 2020), 2020.

Particle-Iot/Electron: Particle, available at: <https://github.com/particle-iot/electron> (last access: 8 September 2020), 2020.

Pham, H. D., Drieberg, M., and Nguyen, C. C.: Development of Vehicle Tracking System Using GPS and GSM Modem, in: 2013 IEEE Conference on Open Systems (ICOS), Kuching, 89–94, <https://doi.org/10.1109/ICOS.2013.6735054>, 2013.

Sabet, B. S. and Barani, G. A.: Design of Small GPS Drifters for Current Measurements in the Coastal Zone, Ocean Coast. Manag., 54, 158–163, <https://doi.org/10.1016/j.ocecoaman.2010.10.029>, 2011.

SOBA: The Ayeyarwady State of the Basin Assessment (SOBA), available at: <https://www.myanmarwaterportal.com/news/663-the-ayeyarwady-state-of-the-basin-assessment-soba.html> (last access: 8 September 2020), 2018.

Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., Band, L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., and Maidment, D.: HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing, <https://doi.org/10.13140/2.1.4431.6801>, 2014.

Tinka, A., Strub, I., Wu, Q., and Bayen, A. M.: Quadratic Programming Based Data Assimilation with Passive Drifting Sensors for Shallow Water Flows, *Int. J. Cont.*, 83, 1686–1700, <https://doi.org/10.1080/00207179.2010.489621>, 2010.

Tinka, A., Rafiee, M., and Bayen, A. M.: Floating Sensor Networks for River Studies, *IEEE Syst. J.*, 7, 36–49, <https://doi.org/10.1109/JSYST.2012.2204914>, 2013.

Trevathan, J., Johnstone, R., Chiffings, T., Atkinson, I., Bergmann, N., Read, W., Theiss, S., Myers, T., and Stevens, T.: SEMAT – The Next Generation of Inexpensive Marine Environmental Monitoring and Measurement Systems, *Sensors*, 12, 9711–9748, <https://doi.org/10.3390/s120709711>, 2012.

Wickert, A. D.: The ALog: Inexpensive, Open-Source, Automated Data Collection in the Field, *Bull. Ecol. Soc. Am.*, 95, 166–176, <https://doi.org/10.1890/0012-9623-95.2.68>, 2014.