Open-source Internet of Things remote aquatic environmental sensing

Jarrod Trevathan a,⇑, Simon Schmidtke b

⇑Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Queensland Australia
bSubstation33, Logan, Queensland, Australia

A R T I C L E   I N F O

Article history:
Received 1 February 2022
Received in revised form 7 July 2022
Accepted 7 July 2022

Keywords:
Remote environmental monitoring
Arduino microcontroller
I2C sensors
3D printing
LX
Temperature
Turbidity
Environmental data
Internet of Things
Wireless sensor network

A B S T R A C T

This paper details the physical and hardware design of a flexible open-source IoT (Internet of Things) platform for environmental sensing. The application is a remote water quality monitoring buoy that can be deployed in calm, shallow near-shore aquatic environments with fresh or brackish water. The system’s development has been informed by experience through conducting multiple actual water quality studies over a prolonged period. The system runs an Arduino Mega 2560 microcontroller using off-the-shelf Adafruit LX and temperature sensors. A light attenuation turbidity sensor is adapted and integrated into the design. A TinySine 3G GSM module transmits data to a server that is displayed via a ThingsBoard IoT dashboard. The system is stable over time, provides reliable remote sensor readings, has low energy consumption, and is powered by renewable energy (up-cycled batteries). The hardware aspires to be general-purpose so that future environmental monitoring applications can repurpose the electronics by adding new compatible sensors and modifying the physical design to match the requirements.

© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Specifications table

| Hardware name                                      | General-Purpose Sensor Board                                                                 |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Subject area                                      | • Environmental, planetary and agricultural sciences                                         |
| Hardware type                                     | • Field measurements and sensors                                                             |
| Closest commercial analog                         | No commercial analog is available.                                                            |
| Open-source license                               | GNU Lesser General Public License (LGPL) 3.0                                                  |
| Cost of hardware                                  | $155 (USD)                                                                                    |
| Source file repository                            | https://osf.io/nhj4z                                                                         |
| OSF registration DOI                              | https://doi.org/10.17605/OSF.IO/9SWE3                                                        |
| OSHWA certification UID                           | AU000010                                                                                     |

* Corresponding author.
E-mail address: j.trevathan@griffith.edu.au (J. Trevathan).

HardwareX 12 (2022) e00336

© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Hardware in context

Remote aquatic ecological data acquisition reduces the amount of time humans must spend in the field, thereby limiting the physical hazards and logistical expenses [1–14]. The ability to observe the data in near real-time (i.e., every 5–15 min) allows stakeholders to monitor an event as it unfolds (e.g., a fish kill) and take appropriate curative actions. This paper presents the physical and hardware design of an aquatic environmental monitoring buoy [15]. The hardware is referred to as the General-Purpose Sensor Board (GPSB) in that it can incorporate almost any type of sensor to facilitate remote environmental monitoring (i.e., the design is not just constrained to aquatic applications). The architecture aims towards minimal system complexity, low energy consumption, renewable power supply (up-cycled from e-waste), plug and play operation and stability/reliability over time using commercial-grade sensors. The system is informed by practical requirements based on actual deployment logistics and constraints specific to aquatic environments. Developed in conjunction with a social enterprise [16–18], the IoT platform has been used in multiple environmental studies involving numerous types of water bodies [9,15,19,20]. To our knowledge this is the first documented open-source design for a non-commercial system (see Tables 1–4).

Hardware description

Fig. 1A illustrates the IoT environmental monitoring application. Stationary aquatic monitoring buoys are deployed in a water body. The buoys are moored in locations of scientific interest to the end user. Each buoy is independent and has a series of sensors (in this case lux, temperature, and turbidity) that take periodic environmental readings. The environmental data (in addition to system status information) is sent back to the end user via GSM. An IoT web dashboard displays the data (see sb.substation33.com for a demo) (Fig. 1 B).

Fig. 2 depicts the main components of the physical buoy and shows one deployed in water. The electronic components (i.e., PCB, batteries, telemetry, wiring) are contained in a waterproof canister that is partially submerged. A shaft extends from the base of the canister to approximately 0.5 m underwater. The shaft terminates with a sensor head containing the environmental sensors. Ballast is attached to the shaft to ensure the buoy is appropriately weighted and balanced in the water. The buoy is moored to an anchor (via a tie-off point). The buoy lid contains a surface lux sensor and solar panels to charge the batteries.

Fig. 3 presents the GPSB functional block diagram. At the core of the GPSB is an Arduino Mega 2560. The GPSB uses a hardware-controlled 555 timer to regulate duty cycling – that is, the ability for the buoy to power down between sampling and transmission to conserve energy (refer [15] for the rationale). The system uses off-the-shelf Adafruit sensors for lux (surface and 0.5 m underwater) and temperature (also 0.5 m underwater). A custom-designed light attenuation turbidity sensor is incorporated into the platform (refer to [20]).

Fig. 3 also illustrates support for additional sensor parameters such as dissolved oxygen, potential of hydrogen (pH) and electrical conductivity (salinity). These are not specifically addressed in this paper. [19] describe how the GPSB was expanded to support Atlas Scientific1 circuitry for these additional sensor types.

A design goal for our proposal is simplicity whereby the buoy is easily deployed without the need for end users to engage in complex configuration or establish a land-based telemetry station to relay information. We elected for a TinySine 3G GSM module to place the network infrastructure duty with the telecommunications providers.

Fig. 4 A shows the assembled GPSB with the mounted batteries and 3D printed components to attach the PCB within the canister. Fig. 4 B illustrates the sensor head electronics with the lux, temperature and turbidity sensors installed. This paper will focus on the physical buoy build and the electronics design and assembly.

The principal requirement for this system is to remotely view the data in near real-time (i.e., samples are taken and transmitted every 15-minutes). The duty cycling period can be adjusted according to the end user’s needs. Over time, we have developed custom web user interfaces and trialed generic IoT dashboards (e.g., Thingsboard). While third party IoT dashboards lack flexibility compared to custom designs, they are ultimately less expensive and less complicated to manage over the longer term. As this paper addresses the buoy hardware design, we will not discuss the server software and data management functionality.

Design files summary

Bill of materials summary

Required materials for the physical assembly/buoy enclosure, 3D printed components and electronics hardware.

---

1 atlas-scientific.com.
Build instructions

Physical buoy build

The main electronics (i.e., the GPSB) is contained in the buoy’s canister (see Fig. 5). The GPSB provides system control, power management, timing, data logging and telemetry. The sensor head is submerged 0.5 m underwater. The sensor head consists of a light (lux) sensor, temperature sensor, and turbidity sensor.

| Design file name       | File type | Description                      | Open-source license | Location of the file   |
|------------------------|-----------|----------------------------------|---------------------|------------------------|
| PCB.brd                | Eagle     | Eagle board file for the main PCB| GNU GPL v3          | https://osf.io/n29sw/   |
| PCB.sch                | Eagle     | Eagle schematic file for the main PCB | GNU GPL v3     | https://osf.io/jkwbq/   |
| Turbidity.brd          | Eagle     | Eagle board file for the sensor head PCB | GNU GPL v3 | https://osf.io/kux3j/    |
| Turbidity.sch          | Eagle     | Eagle schematic file for the sensor head PCB | GNU GPL v3 | https://osf.io/kux3j/    |
| Lid.stl                | STL       | Design file for the buoy lid 3D print | GNU GPL v3 | https://osf.io/8ySau/    |
| Sensor Head.stl        | STL       | Design file for the sensor head 3D print | GNU GPL v3 | https://osf.io/2g8C7/    |
| Sensor Head Cover.stl  | STL       | Design file for the sensor head cover 3D print | GNU GPL v3 | https://osf.io/7fr46/    |
| Ballast.stl            | STL       | Design file for the ballast 3D print | GNU GPL v3 | https://osf.io/wpfc/     |
| Tie-Off Point.stl      | STL       | Design file for the tie-off point 3D print | GNU GPL v3 | https://osf.io/k7c64/    |
| Battery Holder.stl     | STL       | Design file for the battery holder 3D print | GNU GPL v3 | https://osf.io/3b68/     |
| PCB Frame.stl          | STL       | Design file for the PCB frame 3D print | GNU GPL v3 | https://osf.io/un2mw/    |
| gpsb_firmware_v1.0.zip | Arduino Sketch | Arduino source code to run the GPSB | GNU GPL v3 | https://osf.io/h3arz/   |

Table 1
Design file summary.

Table 2
Purchased materials for physical buoy components.

Consumables items such as hot melt glue and sandpaper may be required but are not included in the bill of materials. Required tools are also not included.

* Component can be used for multiple builds. As such, the price listed can be averaged out over the number of builds.

Build instructions

Physical buoy build

The main electronics (i.e., the GPSB) is contained in the buoy’s canister (see Fig. 5). The GPSB provides system control, power management, timing, data logging and telemetry. The sensor head is submerged 0.5 m underwater. The sensor head consists of a light (lux) sensor, temperature sensor, and turbidity sensor.
Main buoy canister

Fig. 6 depicts the main components of the physical buoy build. The body must be constructed to the appropriate dimensions (size and weight) to ensure that it sits at the correct height in the water.

The steps for the construction of the physical buoy canister and shaft are as follows:

1. Cut PVC pipe to 20 cm length.
2. Using a tapered reamer drill bit create a 25 mm diameter hole in centre of the end cap.
3. Sand the outer PVC pipe surfaces, inner end cap and inner coupling with 80 grit sandpaper.
4. Apply PVC priming fluid to the outer PVC pipe surface, inner end cap and inner coupling.
5. Install the tank outlet in end cap. Apply high vacuum grease to the sides of the rubber seal that contact with the end cap. Tighten as appropriate.
6. Apply PVC cement to the outer PVC pipe surface and inner end cap (using dabbing motion with applicator rather than consistent brush strokes).
7. Insert the PVC pipe into end cap. Twist back and forth momentarily before moving into final setting position.
8. Repeat the same process to attach the threaded access coupling to the top of the PVC pipe.
9. Paint the outer canister with yellow paint.
10. Apply Teflon tape to the poly riser thread and insert into the tank outlet.

Lid

The buoy lid consists of the PVC threaded cap bonded to a 3D print that houses several electrical components (Fig. 7). The 3D print contains slots to mount 2 solar panels, the surface lux sensor, lux sensor diffuser and the Status LED. Care must be taken when sealing the lid as it is a critical point of failure/water ingress. Penetrations are made in the lid (or ideally only one penetration) that corresponds to points on the 3D print where wires to the respective electronics components must be fed through into the canister.

The threaded access cap contains an O-ring which compresses as the access cap is tightened to form a watertight seal. A liberal amount of high vacuum grease (or similar compound) should be applied to the O-ring prior to deployment. This will provide an extra degree of silicone-based sealing and protection from ingress. Ensure that the O-ring is free from dirt/debris and is not pinched by the access cap when turning. Any minor breach or deformity in the O-ring may result in the seal failing.

The lux sensor can saturate under direct or intense sunlight. Two mechanisms are needed on the lid to prevent this from occurring. The first is aperture, where a gap in the lid controls the amount of light entering the sensor. The second mechanism is a light diffuser. A light diffuser is opaque white acrylic placed over the lux sensor. The diffuser scatters incoming light to evenly distribute the photonic energy reaching the sensor. In addition to preventing light saturation, the diffuser can also alleviate problems due to shadowing. Fig. 7 shows dimensions of light diffuser and its positioning on the lid over the aperture.

The steps for the construction of the physical buoy lid are as follows:

1. Drill appropriate penetrations through the threaded access cap.
2. Sand the top surface of the threaded access cap and apply PVC priming fluid.
3. Apply acetone and liquid ABS coating to the lid 3D print as appropriate to create a waterproof seal.
4. Paint the lid 3D print.
5. Install solar panels by feeding the wires through holes. Run a bead of silicon sealant along the center of the solar panel depressions on the lid. The silicone should sandwich outwards as pressure is applied to the panels. It is critical to ensure that the silicone covers all of the surface area between the solar panels and the 3D print.
6. Install the surface lux light diffuser by running a bead of silicon around the aperture depression.
7. Mount the Status LED in place, seal from underneath with Blu-Tack™ and backfill the cavity with epoxy.
8. Glue the lid 3D print to the threaded access cap using PVC cement. Run a bead of silicone around the outside surface between the 3D print and threaded access cap.

### Table 3

| Designator | Component | Number | Total cost - currency | Source of materials | Material type |
|------------|-----------|--------|-----------------------|---------------------|--------------|
| Lid        | Lid       | 1      | Aliexpress            | Acrylonitrile Butadiene Styrene |
| PCB Frame  | PCB Frame | 1      | Aliexpress            | Acrylonitrile Butadiene Styrene |
| Battery Holder | Battery Holder | 1 | Aliexpress | Acrylonitrile Butadiene Styrene |
| Tie-Off Point | Tie-Off Point | 1 | Aliexpress | Acrylonitrile Butadiene Styrene |
| Ballast    | Ballast   | 1      | Aliexpress            | Acrylonitrile Butadiene Styrene |
| Sensor Head | Sensor Head | 1 | Aliexpress | Acrylonitrile Butadiene Styrene |
| **Total**  |           |        | **$6.56 USD**         |                     |              |

The total cost is estimated based on ~ 350 g of 3D print filament required from a 1Kg roll ($18.75).
9. Mount the surface lux sensor in place from the underside. Ensure that it is correctly positioned as any minute discrepancies will influence the readings.

10. Backfill all lid penetrations with epoxy resin to ensure that there is no possibility of water ingress.

11. Apply high vacuum grease to the O-ring (ensure the O-ring is clean and dry first).

12. Once cured, consider pressure testing the lid underwater over a period to check for leaks.

### Table 4

| Designator | Component | Number | Cost per unit - currency | Total cost - currency | Source of materials |
|------------|-----------|--------|--------------------------|----------------------|--------------------|
| Temp | MCP9808 Temperature Sensor (1782) | 1 | $8.85 | $8.85 | Core Electronics |
| Lux | TSL2561 Lux Sensor (435) | 2 | $5.95 | $11.90 | Adafruit |
| Turbidity | Gravity Analog Turbidity Sensor (SKU: SEN0189) | 1 | $9.90 | $9.90 | DF Robot |
| Temp-Humidity | DHT22 Temp-Humidity Sensor | 1 | $2.40 | $2.40 | Aliexpress |
| Cat5 Cable | Cat5 Cable | 1 | $2.00 | $2.00 | Umarpt |
| LED1 | Green LED (C2895478) | 1 | $0.444 | $0.444 | lccom.com |
| I2C-1, I2C-2, TEMP1-TURBIDITY, SV1 | Plug, AMP Connector (C2913381) | 4 | $0.9212 | $3.6848 | lccom.com |
| I2C-1, I2C-2, TEMP1-TURBIDITY, SV1 | Socket, AMP Connector (C2913359) | 4 | $0.073 | $0.292 | lccom.com |
| R1, R2, R3, R5, R14, R52, R13, R15, R16 | 10 K Resistor (C2903266) | 9 | $0.0064 | $0.0576 | lccom.com |
| R4 | 100 K Resistor (C2843019) | 1 | $0.0069 | $0.0069 | lccom.com |
| R6, R7, R8, R12, R17 | 4 K7 Resistor (C274465) | 5 | $0.0106 | $0.0530 | lccom.com |
| R9 | 100 K Potentiometer (C118963) | 1 | $0.2025 | $0.2025 | lccom.com |
| R10 | 1 K Resistor (C38590) | 1 | $0.0093 | $0.0093 | lccom.com |
| R11 | 15 K Resistor (C119351) | 1 | $0.0088 | $0.0088 | lccom.com |
| R26 | 33 K Resistor (C119358) | 1 | $0.0088 | $0.0088 | lccom.com |
| R39 | 2 K7 Resistor (C119333) | 1 | $0.0088 | $0.0088 | lccom.com |
| C1 | 10n Capacitor (C914518) | 1 | $0.0229 | $0.0229 | lccom.com |
| C2 | 2.2uF Capacitor (C914524) | 1 | $0.0969 | $0.0969 | lccom.com |
| C3, C5, C8 | 100n Capacitor (C914519) | 3 | $0.0229 | $0.0687 | lccom.com |
| C6, C7 | 22uf Polarized Capacitor (C2831715) | 8 | $0.0529 | $0.4232 | lccom.com |
| D1 | 4V7 Zener Diode (C261192) | 1 | $0.0379 | $0.0379 | lccom.com |
| D4 | 1 N5400 Diode (C698905) | 1 | $0.1561 | $0.1561 | lccom.com |
| Q1, Q4 | P-Channel Power MOSFET (C701297) | 2 | $0.615 | $1.230 | lccom.com |
| Q2, Q3, Q6 | 2N7000, 2N7002, NDS7002A (C9114) | 3 | $0.0472 | $0.1416 | lccom.com |
| Q7 | BS250 P-Channel MOSFET (C151455) | 1 | $0.2549 | $0.2549 | lccom.com |
| F1 | 3A TE5 Fuse (C369106) | 1 | $0.0812 | $0.0812 | lccom.com |
| IC1 | 78M05 Voltage Regulator (C347289) | 1 | $0.2467 | $0.2467 | lccom.com |
| IC2 | 12-Stage Binary/Ripple Counter (C73220) | 1 | $0.4941 | $0.4941 | lccom.com |
| IC3 | TLE4431 Shunt Voltage Reference (C184253) | 1 | $1.0946 | $1.0946 | lccom.com |
| IC4 | NE555N, DIL08 Timer (C398063) | 1 | $0.1262 | $0.1262 | lccom.com |
| US3 | XL6009 DC-DC Boost Converter | 1 | $0.85 | $0.85 | Aliexpress |
| US4 | TinySine 3G GSM SIM5320A (G5763B41F1A276) | 1 | $59.95 | $59.95 | TinySine |
| US4 | TINY_SINE_3G, 2.54 mm 40 Socket Header, AMP Connector (C47940) | 2 | $0.2624 | $0.5248 | lccom.com |
| SV1 | RGB LED, AMP Connector (C2837020) | 1 | $0.139 | $0.139 | lccom.com |
| GPIO01 | Plug MTA06-156 AMP Connector (C577781) | 1 | $0.2347 | $0.2347 | lccom.com |
| GPIO1 | Sockk MTA06-156 AMP Connector (C577792) | 1 | $0.1124 | $0.1124 | lccom.com |
| PCB | 2 Layer PCB | 1 | $8.00 | $8.00 | lccom.com |
| J2, PCB1 | Wake, Arduino MTaxx-100 2.54 mm 40 Pin Header, AMP Connector (C113597) | 2 | $0.2078 | $0.4156 | lccom.com |
| PCB1 | Arduino Mega 2650 | 1 | $9.92 | $9.92 | Aliexpress |
| PCB17 | SHARP OPTO COUPLER (C390734) | 1 | $0.1019 | $0.1019 | lccom.com |
| S1 | TINY SWITCH ON – MOM (C2837020) | 1 | $0.7502 | $0.7502 | lccom.com |
| SOLAR-IN, BATTERY-IN | XT30_CONN, 2 × Plug (C9101) | 2 | $0.5032 | $1.0064 | lccom.com |
| SOLAR-IN, BATTERY-IN | XT30_CONN, 2 × Socket (C9102) | 2 | $0.6618 | $1.3236 | lccom.com |
| 6 V 1 W | 6 V 1 W Solar Panel | 2 | $1.55 | $3.10 | Aliexpress |
| Antenna | GSM Quad-Band Antenna SMA Plug (1859) | 1 | $5.95 | $5.95 | Adafruit |
| 18,650 cell | NCR18650B 3400mAh | 8 | $2.02 | $16.16 | Aliexpress |

Total **$154.84 USD**

Consumables items such as solder, isopropyl alcohol and hook-up wire are required but are not included in the bill of materials. Required tools are also not included. Some vendors have minimum quantity limits on purchases, which will increase the prices slightly on what is listed. A SIM card and data plan is necessary for the GSM communications.
Fig. 1. Remote aquatic environmental monitoring.

Fig. 2. The IoT remote aquatic environmental monitoring buoy.

Fig. 3. General-Purpose Sensor Board block diagram.
Mooring system

The mooring system ensures that the buoy remains in its intended location, stays upright and has some in-built resiliency to water height fluctuations (i.e., it does not get submerged). Fig. 8 illustrates a suggested mooring system. The complete buoy system has several components: 1) Ballast; 2) Tie-off point; 3) Float; 4) Mooring (and tethering line); and 5) Anchor.

Ballast enables the buoy to sit in the water at the desired level and keeps the buoy in an upright position in the presence of wind, waves/ripples and other actions that may cause the buoy to move around. Ballast is simply an appropriate amount of weight added at a low centre of gravity. While dive weights are suitable, the design files include a 3D print for a ballast mount. The ballast dimensions fit the outer diameter of the poly riser shaft. The print comes in two parts that are attached either side of the shaft via screws. We use recycled e-waste screws to weight the ballast and seal this with epoxy resin. This ballast system means that the buoy is appropriately weighted prior to deployment, thereby alleviating the end user of these concerns when out in the field.

The tie-off point allows the end user to attach the mooring to the buoy. This is a 3D printed component that is physically bonded via PVC cement to the buoy canister just below the water line. A snap-lock or any simple mechanism can be used to attach the mooring to the tie-off point.
The anchor is a submerged weight used to keep the buoy in place. In this paper we used an 8 kg plastic coated kettlebell. The kettlebell has a handle which makes it convenient to maneuver or tie things to it when out in the field. A brick or any other weighted item can also be used as an anchor.

The mooring system uses a rope from the anchor to a surface float (i.e., polystyrene crab float) which is then tethered to the buoy tie-off point. This allows the buoy to freely move around the float (to the extent of the tethering line). In extreme wind or current situations, the buoy will not be angled as would be the case if it were directly attached to the mooring from the anchor. Furthermore, if the water level rises, there is some resiliency in that the buoy will rise/fall with the water level by the degree of the tethering line preventing it from becoming submerged.
PCB mount and battery holder

There are two internal 3D printed components that are attached to the PCB (Fig. 9). The first is the PCB frame. This is connected to the upper section of the PCB via 2 M3 mounting screws through the board. The frame is then bonded to the underside of the buoy lid. This approach is used to position the electronics inside the canister and prevent unnecessary shocks. Additionally, the PCB is manipulated as part of the lid when opening or sealing the buoy. The battery holder is attached to the lower portion of the PCB via 2x M3 mounting screws through the board. The battery holder contains the lithium-ion battery bank and the associated wiring.

Sensor head

Fig. 10 shows the sensor head. There are three sensors contained in the sensor head – lux, temperature, and turbidity. The sensor head is connected to the main electronics in the canister via a Cat5 cable running through the shaft. The external shell of the sensor head is a 3D printed component that houses the circuit board and sensors. The turbidity sensor uses a High-Density Polyethylene (HDPE) low surface energy plastic cover to transmit through. A light shield is added to the top of the
turbidity sensor to reduce interference from ambient light. The sensor head uses mechanical and chemical bonding for the water seal.

The steps to assemble the sensor head are as follows:

1. Solder the Cat5 cable wires to the respective terminals on the sensor head.
2. Drill a 5 mm hole in the centre of the plug and feed through the Cat5 cable. Ensure that the threaded end of the plug is facing away from the sensor head circuit.
3. Apply acetone and liquid ABS coating to the sensor head 3D print as appropriate to create a waterproof seal.
4. Use hot melt glue to plug the turbidity sensor LED/ IR phototransistor into a cavity on the turbidity sensor HDPE cover. (This will prevent resin from step 8 entering the cavity.)
5. Use Scotch-Weld™ 8005 to attach the HDPE cover to the sensor head. (Apply mechanical strength during the curing process.)
6. Install the light diffuser on the sensor head using a fine silicone bead.
7. Insert the sensor head electronics into the 3D print.
8. Backfill with epoxy to create a waterproof seal around the electronics.
9. Insert the plug while the resin is still workable.
10. Attach the light shield to the sensor head and fasten via a 316 stainless steel screw.
11. Apply Teflon tape to the elbow joint thread and attach to the shaft.
12. Feed Cat5 cable through the buoy shaft and attach connectors at the canister end.
13. Apply Teflon tape to the plug thread and attach to the elbow joint. (Ensure that the sensor head is oriented with the light diffuser facing upwards.)

Hardware build

A systems approach has been taken in the development of the buoy electronics. This allows the complexity to be managed and related functionality grouped according to a particular subsystem. Table 5 shows the buoy subsystems. The following sections outline the functional requirements for each of these subsystems, the current technologies employed, and the rationale for the design approach (see Table 6).

Microcontroller subsystem

Fig. 11 shows the electronics schematic for the Arduino Mega 2560. The Arduino is placed in the context of the greater circuit by showing the respective components attached to the various pins. The design files contain a version of the Arduino sketch for the buoy which adheres to this pin configuration.
The voltage is supplied from the power subsystem to the GND/5V pins. The battery and solar reference voltages are connected to analog pins A6 and A7 respectively. Resistors R1, R2 and R3 function as voltage dividers. These reference voltages allow an estimate of the current battery voltage levels and the solar charging status.

An RGB Status LED is externally visible on the lid. This provides the end user an indication of how the buoy is functioning (refer to the Software operation section for details). This LED requires analog pins A1 and A2 for the blue and red colours respectively. Green is powered via the duty cycling hardware. The LED has a common cathode.

The lux and temperature sensors use the I2C bus to interface with the system. These require common pins for power (GND/5V), a data signal (SDA) and clock signal (SCK). An internal temperature and humidity sensor is connected to digital pin D7 to monitor the buoy for leaks. The turbidity sensor requires power rails and two analog pins. The first pin provides a PWM power source to the IR transmitter. The second records the response from the IR receiver. The TinySine GSM module requires a power rail and receive (RX) and transmit (TX) pins.

Power and solar charging subsystem

Fig. 12 shows the assembled battery pack. 8 18,650 Lithium-ion batteries are placed in a series to deliver 3.6 V (nominal) (27,200mAh). The battery pack is assembled by soldering a metal strip across the respective common terminals of the batteries. Hot melt glue bonds the batteries in place and positive/negative wires are soldered to the ends. The battery pack is inserted into the battery holder as outlined earlier. We used repurposed 18,650 batteries salvaged from e-waste components for the buoy builds outlined in this paper.

Fig. 13 shows the electronics schematic for power and solar charging. Using the aforementioned battery pack as the power source, the DC-DC step up convertor increases the battery voltage to 7 V before dropping it to 5 V via the regulator. This design is used to protect against an undervoltage condition that has the potential to damage the circuitry (potential to spike at 35 V when the battery is near 3 V). The battery switch controls the power to the circuit. The timing circuitry can be bypassed by J2.

The two 6 V solar cells on the lid supply power to a solar charge regulator circuit. This connects the 6 V panel to the 18,650 lithium-ion cells when charging is enabled. D4 forward biases the current to protect against polarity issues and discharging back into the panels overnight. LED1 provides a visual cue to the user for diagnostic purposes to indicate when sunlight is present on the solar panels and charging is occurring.

The solar charge circuit contains a voltage divider and a TLV431 (voltage adjustable shunt regulator) which compares the output of a voltage divider across the battery to the TLV431’s internal reference voltage of 1.24 V. If voltage is higher than internal reference, the TLV431 turns on and activates the optocoupler. The optocoupler then turns off the solar charge P-Channel MOSFET. If the voltage is less than input reference, the TLV431 is off and the optocoupler is off. This enables the P-Channel MOSFET to operate in its default state via a pulldown resistor on its gate (i.e., solar charge takes place). The voltage divider can be set to any trigger voltage – in this case 3.97 V which is the optimum charge for the batteries.

### Table 5

Buoy subsystems.

| Subsystem                  | Description                                                                 |
|----------------------------|-----------------------------------------------------------------------------|
| Microcontroller            | System control                                                              |
| Power and Solar Charging   | Batteries, power management logic, solar panels and solar charge controller |
| Timing                     | Duty cycling control                                                        |
| Sensors                    | Sensor head, connections, sampling, calibration                             |
| Communications             | Transmissions of sensor samples and system status to the server             |

### Table 6

Remote environmental monitoring buoy deployment outcomes over a 12-month development period.

| Buoy Id | Location       | Deployment Duration | Failed / Challenged | Reason                      |
|---------|----------------|---------------------|---------------------|-----------------------------|
| Buoy 1  | Slacks Creek   | 56 days             | ✓                   | Vandalised                  |
| Buoy 2  | Slacks Creek   | 25 days             | ✓                   | Lack of sufficient solar power |
| Buoy 3  | Slacks Creek   | 56 days             | ✓                   | Stolen                      |
| Buoy 4  | Wyaralong Dam  | 70 days             | ✓                   | Solar panel water ingress   |
| Buoy 5  | Wyaralong Dam  | 270 days            | ✓                   |                             |
| Buoy 6  | Keyatta Lake   | 120 days            | ✓                   |                             |
| Buoy 7  | Fairfield Waters | 120 days         | ✓                   |                             |
| Buoy 8  | Ross River     | 130 days            | ✓                   |                             |
| Buoy 9  | Ross River     | 130 days            | ✓                   |                             |
| Buoy 10 | Burdekin River | 200 days            | ✓                   |                             |
| Buoy 11 | Hinze Dam      | 168 days            | ✓                   | Submerged due to water level rise |
Timing subsystem

System timing is critical for determining when to take samples and correctly time stamping the samples. There are three sources of timing for the system:

1. A Real Time Clock (RTC) located on the buoy;
2. Obtaining the time from the GSM network; and
3. The time on the backend server database when a transmission is received.
The RTC provides a way to keep track of the time (either 24-hour or 12-hour format) so that sensor readings can be time stamped. A prior version of the buoy [9] employs the DS3231 RTC by Texas Instruments. The RTC contains a CR2032 coin-cell battery to allow it to keep track of the current time. The RTC is independent of the Arduino. The battery cell can last for up to 5 years. The time set in the RTC is synchronised with the computer that loads the script onto the Arduino. However, over time we observed that there is quite significant time drift and variation in alarm timing between different DS3231 RTCs.

If a buoy is relied on to timestamp its own sensor samples, then there is the definite likelihood that all deployed buoys will not be sampling at a common time in synchrony. As such, apart from setting an alarm to control duty cycling, the RTC should not be relied upon for time stamping sensor samples.

A common time can be obtained via the GSM network. Here the time of transmission can be logged, but the RTC cannot be easily updated to resynchronise all buoys. Ultimately, the easiest approach is to have the backend server log the time a sample is received from each buoy and use the database’s common timestamp. This removes the responsibility of time stamping from the buoys (and the need for the CR2032 battery) but does not fix issues of buoy synchronisation. Experience indicates
that on a 15-minute sample interval, all buoys will transmit within 7½ minutes of each other. Generalised, this means that for an \( n \)-minute sample interval, all buoys are synchronised to transmit within \( n/2 \) min of each other.

**Fig. 14** shows the electronics schematic for the timing subsystem. Duty cycling is controlled by a 555 timer (NE55N) and a 12-bit binary counter (4040 N) set up to cycle the most significant bit every 15 min. When the most significant bit is reached the binary counter outputs a signal on pin Q12. This triggers a MOSFET (Q2) to change state which then allows current to follow from the battery and power up the circuit.

**Fig. 15** shows the sleep and failsafe timing approach. The most significant bit triggers the wake-up signal that wakes the buoy up. When the buoy finishes executing its code, it sends a sleep command to re-initiate the timer (via the Counter_Reset pin). If this signal is missed, a software watchdog timer will attempt to send the signal again. If the signal is again missed, the buoy will stay awake for at most 15 min before a secondary hardware timer kicks in and forces the buoy back into a sleep cycle.

**Sensors subsystem**

**Fig. 16** shows the environmental sensors used in the buoy and their associated schematic as it relates to the PCB interface. These consist of the MCP9808 temperature sensor, 2x TSL2561 luminosity sensors, a custom-built light attenuation turbidity sensor, and an internal DHT22 temperature and humidity sensor.

MCP9808 is a low-current I^2C digital temperature sensor that operates between 2.7 V and 5 V. The sensor has an operating range of \(-40 \, ^\circ C\) to \(+125 \, ^\circ C\) with \( \pm 0.25 \, ^\circ C\). The sensor requires the Adafruit MCP9808 library to be installed.
The TSL2561 is a low-current I²C digital light sensor that operates between 2.7 V and 3.6 V. The sensor can measure 0.1–40,000 + Lux and contains infrared and full spectrum diodes enabling separate infrared, full-spectrum or human-visible light readings. The sensor has a spectral responsivity between 300 and 1100 nm wavelengths. The TSL2561 requires the corresponding Adafruit library. While this sensor now discontinued, variants still exist. (We moved to the TSL2591 in new iterations of the buoy.).

Fig. 17 shows the electronics schematic for the turbidity sensor. The sensor was developed from parts salvaged from a DF Robot Gravity Analog Turbidity sensor (refer to [15,20] for the rationale). The circuit retained the original LED and IR phototransistor from the Gravity turbidity sensor. However, all the remaining electronics were reduced to a bare minimum. The

---

**Fig. 16.** Sensors.

**Fig. 17.** Turbidity sensor electronics schematic.
Arduino has direct control over the voltage supplied to the LED. Pulse Width Modulation (PWM) is used to control the power supplied to the IR LED. PWM operates by providing either zero volts or five volts in a cycle: by increasing or decreasing the duration of the five volts over time, the power delivered to the IR can be increased or decreased.

The left side of Fig. 17 depicts the LED transmitter. The Arduino Mega 2560 contains fifteen PWM output and runs six internal clocks. The default frequency of the Mega is changed from 490.20 Hz to 31,372.55 Hz to maximise the PWM cycling and provide a stable voltage. The 100uF capacitor smooths out the voltage provided to the LED. The capacitor maintains a consistent amount of power in the periods of zero volts during the PWM duty cycling. This forms an aggressive low-pass filter to turn a digital PWM signal into an ongoing stable voltage for the sensor.

The second part of the schematic is the IR phototransistor. A 4.7 K resistor biases the IR phototransistor. An increase of IR light on the base of the IR phototransistor increases the current flowing through it, increasing the output voltage. This is further filtered by a low pass filter (a 10 K resistor and 100nf capacitor) that blocks high frequency signals from being output.

A DHT22 temperature and humidity sensor is included to provide information about internal humidity build-up and to detect leaks. Additional status information regarding the buoy’s operation (uptime, transmit time, network signal strength, solar charge and battery voltage) is transmitted with the sensor readings.

Communications subsystem

Fig. 18 shows the communications subsystem for the buoy. We are using a TinySine 3G/GPRS/GSM Shield for Arduino with a SIM5320A. The TinySine is a less expensive variant of the Adafruit Fona Cellular breakout (note that we are not using the GPS functionality in this version of the buoy). The TinySine enables connection to the GSM network provided you have a SIM with a data provider. There are two version of the TinySine (i.e., American and European) which differ based on the frequencies they use. You need to ensure you have the correct version that is supported by your telecommunications provider.

TinySine requires the Fona Library to be installed. We initially experienced some issues with lack of data support for the SIM5320A in the Fona Library. This required modifications to the Fona Library to support HTTP posts. The modified Fona Library is included as part of the design files.

The buoy's transmission is encoded as a JSON string. The first parameter is a unique identifier for the buoy. The subsequent parameters are the data for the sensor readings. The version of code provided allows for three connection and send attempts before failing (indicated by the Status LED flashing red). The code does not support logging or bulk data transfers. The Arduino’s EEPROM or the TinySine’s Flash memory can be used to buffer sensor data. The TinySine transmission is the most power-intensive component when a buoy is awake. We have experimented with using logging techniques to provide transmission back-offs during low battery voltage conditions.

The buoy does not timestamp data at the source. This occurs when the server enters sensor data into the database. Alternatively, a timestamp can be obtained from the network when the TinySine establishes a connection.

The antenna shipped with the TinySine needs to be replaced with a vertical quad band antenna to be in the appropriate position within the buoy canister. We have attempted to situate the antenna at the highest point inside the canister clear of electrical interference.

We recommend placing a cable tie around the TinySine or adding hot melt glue to mechanically bond the shield to the PCB. Over time wave action and/or vibrations against the buoy can potentially cause the shield to become partially dislodged from its headers.

The TinySine module supports GPS and can be used to provide a fix on the buoy’s location. This is helpful to locate a wayward buoy or to track theft. However, the system outlined in the paper did not use the GPS functionality mainly due to power and complexity reasons. The GPS requires significant time to get a signal lock (e.g., 2 – 5 min of transmission). This load on the duty cycle would place strain on the battery and the number of GPS location checks would have to be significantly reduced.

Fig. 18. TinySine 3G/GPRS/GSM Shield for Arduino with GPS - American version SIM5320A.
reduced (e.g., once a day or once a week). An active antenna could be used to marginally reduce the signal lock establishment
time. But this also adds another power-intensive component to the system.

Operation instructions

Software operation

Configure the appropriate parameters in the Arduino sketch (i.e., the buoy's unique identifier and the location of your
server to transmit the data to). Also, set the PWM value specific to your turbidity sensor. Load the sketch onto the Arduino
via a USB cable. Provided that the USB cable is connected, all power management circuitry will be bypassed, and the sketch
will immediately commence execution. Use the Arduino IDE's serial monitor to observe the output during testing. If satisfied
with the operation of the code, disconnect the USB cable.

The circuit can be activated by toggling the power switch to the on position. The Status LED will illuminate to green indi-
cating that the buoy is in a sleep state (the internal counter will immediately commence). If you need to force the buoy
awake you can bypass the switch and timing circuitry by placing a jumper on J2 (refer to Fig. 13). Ensure that you remove
the jumper when finished so that the buoy reverts back to its normal operation.

The server software setup is outside the scope of this paper. However, we have designed the firmware (i.e., the Arduino
sketch) to be minimal. Most of the functionality has been moved to the server software where possible. This allows the end
user greater control and system resources. Apart from the configuration information outlined above, once the server knows
the buoy's identifier it should be able to handle all incoming requests from the buoy and manage the database accordingly.

In addition to the sensor data, the buoy will transmit system status information. This includes parameters such as solar
charge, battery level, previous uptime, network signal status, internal temperature and humidity, and error codes. This infor-
mation makes it easier to monitor the buoy's operation and diagnose problems during deployment.

Sensor calibration

Sensor calibration is a process to ensure the accuracy of the sensor readings. All sensors will read differently based on
their individual mechanical and electrical properties. Furthermore, some sensor readings will drift over time due to errors
or environmental conditions. Calibration is a procedure that inspects a sensor's behaviour and readings against a known
source prior to deployment. The calibration process should also be repeated periodically so that the reading accuracy can
be relied on over time.

Calibration is a scientifically rigorous process – the importance of which should not be overlooked. Do not deploy the
buoy until you are happy with the sensor readings and that they are valuable for your application.

The MCP9808 temperature sensor's datasheet outlines the sensor's range, accuracy, and thermal response properties.
Bear in mind that this sensor is not in direct contact with the water. As it is housed inside the sensor head, it usually takes
up to 15 min before the sensor's readings equalise with the actual water. This rate-of-change differential should be consid-
ered if your application relies on accurate and timely readings. An appropriate calibration procedure should enable you to
compensate.

The TSL2561 lux sensor's datasheet outlines the sensor's range, accuracy, and spectral response curves. In terms of our
specific application, we are interested in measuring the intensity of the light within the photosynthetically active radiation
wavelengths – which the TSL2561 can do. Note that the inclusion of the light diffuser and aperture change the sensor's
response properties and this should be investigated if it affects your application. We experimented with comparing the sen-
 sor readings against a Li-Cor2 light meter.

We undertook extensive calibration of the turbidity sensor against a Hach turbidimeter based on the ISO 7027 nephelo-
metric standard for measuring turbidity. Formazin (industry turbidity standard) calibration samples were used that approx-
imated several known NTU values (i.e., <0.2, 20, 200, 1,000, 4,000, and 7,500).

Fig. 19 A illustrates a typical response curve for our turbidity sensors. Here, we uniformly input all the pulse width mod-
ulation (PWM) values between 0 and 255 to observe the voltage received (referred to as a PWM sweep). In this instance, we
used tap water (NTU < 1).

All turbidity sensors have a curve similar to the one shown in Fig. 19 A. However, each sensor varies in its power and
response. In this example, at PWM values > 150, the sensor tended to saturate. PWM values < 60 were difficult for the sensor
to get any sort of reading. Therefore, the measurable range for the sensor appeared to be PWM values between 60 and 150.
The optimal PWM value to use should be near the top of the curve before the saturation effect occurs. However, the exact
PWM value for each sensor is different depending on the sensor's response to differing turbidity values.

To establish the optimal PWM value for a sensor we tested 10 sensors on Formazin samples in controlled lighting/tem-
perature conditions. Once the PWM value was chosen, the calibration curve was determined. Fig. 19 B shows the calibration
curve for a typical sensor. The fit for NTU values > 100 is good, indicating that the sensor was extremely reliable and accurate
for measuring higher levels of turbidity (100NTU to 4,000NTU). However, the reliability and accuracy decreased somewhat

\(^2\) licor.com.
when the samples had turbidity below 100NTU. This is below the sensitivity range of the sensor. (See [20] for a complete discussion on the sensor’s calibration.).

**Deployment and maintenance**

We are assuming that the Arduino sketch has been loaded onto the buoy and all appropriate configuration has occurred. When the power switch is turned on, the buoy will activate and instantly go into sleep mode (Status LED will be green). In 15 min, the buoy will wake up (Status LED will be blue). If an error occurs during operation, the Status LED will flash a red error code and the buoy will go back into a sleep cycle.

If the buoy is operating correctly and data is being received at the server, the buoy canister can be sealed. Refer to all previous instructions on the physical construction – especially ensuring Teflon tape on the threaded joins and silicone resin seals are intact. Grease the O-ring in the threaded access cap. Insert a fresh silica gel pack into canister with the electronics and seal the lid. Ensure the lid is firmly sealed, but do not overtighten.

We recommend placing the buoy in direct sunlight and/or in a tank of water over several days prior to deployment. Use this period to allow sensor readings to stabilise and to check the operation of all systems – particularly the solar charge cycle. Do not open the buoy while out in the field. The buoy should remain sealed until it is ready for maintenance in a controlled environment.

If there is an issue with communication, check for any interference with the 3G signal such as physical obstructions. Ensure that your SIM plan is activated and compatible with the TinySine version you are using. It would be advantageous to know the GSM signal strength in the target location prior to deploying the buoy.

Once testing is complete, the buoy can be deployed using the mooring approach described earlier. The exact deployment logistics and approach will be dependent on the application and the type of water body. We recommend that this version of the buoy be used in calm, shallow near-shore conditions in fresh or slightly brackish water. The buoy is unsuitable for open oceanic conditions.

The buoy should be periodically inspected for visible external damage. The sensor head and solar panels should be cleaned often. Biofouling naturally occurs in water and will cause a build-up on submerged components. Furthermore, birds and other wildlife can leave excrement on above water components (or even attack them). The amount and rate of fouling will vary according to the environment. The sensor readings and diagnostic data transmitted with the buoy usually give the first indication when fouling is becoming an issue. The exact frequency of cleaning will be dependent on your circumstances. Antifoul paint may prolong the time between cleaning.

We suggest retrieving the buoy every six months and undertake thorough maintenance by opening the buoy and inspecting its internal componentry for leaks or damage (do this in a controlled environment). Use this as an opportunity to perform a firmware update, refresh the silica gel pack, check the O-ring integrity, and re-calibrate the sensors. The version of the buoy outlined in this paper will function for approximately 18–24 months under ideal conditions (based on our experience). In newer versions of the buoy (to be outlined in future papers) we have addressed some of the identified shortcomings in the physical robustness to extend the buoy’s lifespan.

Working with water is dangerous, so plan your work carefully. Prior to undertaking any field work, ensure that you have appropriate permits to enter the water body and to deploy equipment. You must also complete relevant occupational health and safety training and fill out field work risk assessment documentation as per your organisation’s requirements.
Work with experienced and qualified field technicians. The buoys have been designed to mitigate some of the risks and challenges when out in the field.

Validation and characterization

Trial deployments

Ten buoys were constructed during the development period and were deployed on an ongoing basis throughout 2019/2020 in Wyaralong Dam, Hinze Dam, Ross River, Slacks Creek and the Burdekin River. Table 5 shows the deployments outcomes in terms of the duration and whether any of the buoys became challenged or failed.

Newer iterations of the buoys were constructed throughout 2020/2021 to address these shortcomings. These later buoy versions have achieved commercial success and are used in applications for scientific research, council waterway monitoring and prawn farming [19].

Energy consumption

Fig. 20 A displays the battery voltages from the Wyaralong Dam deployment from mid-winter to the end of spring. The figure shows the (averaged) fluctuations in the voltages every fifteen minutes with the (averaged) daily voltage trend. There are three basic features in this graph:

1. The first five days, where the batteries are increasing in voltage from their pre-deployment charge up to full charge;
2. The middle period up to ninety days where the batteries are maintaining a steady voltage, corresponding to the end of winter; and
3. The last stage from ninety days on where the battery voltage is on a slightly increasing trend, corresponding to the spring period.

Fig. 20 B displays the voltages over two five-day periods: the July readings in the middle of winter and the November readings in the middle of the last month of spring. The battery voltage is higher during spring. This is due to the increasing amount of sunlight available to the batteries in spring. The batteries are not stressed and would have continued operating indefinitely under this duty cycle. These results indicate that with recycled Li-ion batteries charged to within an optimal capacity 50%-75%, the system can essentially last an indefinite period out in the field. That is, the batteries will last for potentially 20 years, which is longer than the actual life of the buoy. Furthermore, the batteries from retrieved buoys can also be recycled and repurposed for other applications. No other commercial systems or academic proposals can make this claim.

The system (with the given battery pack) can operate for 5–6 weeks in the absence of solar charge. This circumstance typically would not occur unless the buoy is deployed in a shaded area or there is a problem with the solar charging subsystem. Increasing the battery pack size would prolong the amount of time the buoy could function without charge. Likewise, lengthening the duty cycle (i.e., number of power ups for a given period) would also preserve battery life.
Calibrated data validation

The deployment results demonstrated that the turbidity sensor was reactive to its environment and indicative of change. Fig. 21 presents calibrated data from Wyaralong Dam. The plot shows 48 h of turbidity readings taken over four consecutive night-time periods of a twelve-hour duration. The blue line displays the raw turbidity, while the red line is a running average of the five previous and post estimates (i.e., the smoothed turbidity estimates averaged over the five previous and post estimates). In conditions where there were high amounts of suspended sediment, the NTU value for the sensor would read much higher than water bodies with less sediment. The sensor’s NTU value would rise significantly during heavy rainfall due to an influx of sediment being washed into the system, which would gradually fall again as the sediment settled.

There was a degree of ambient IR interference during daytime readings. Even though the sensor was shielded from the direct sun, the sensor picked up refracted light through the water. The nighttime readings proved to be more stable when this interference was not present. Future work would be to take a sensor reading without the transmitter turned on to capture the amount of ambient IR and then subtract this from the official reading to compensate for the interference.

Different water bodies foul at different rates. Some of the deployments were maintained regularly (i.e., the sensor was cleaned), whereas other locations did not have any maintenance. The impacts of fouling can be seen as the gradual rise in NTU value over time as it became more difficult to transmit signals through the lenses. This evidenced the need for regular maintenance to ensure optimal readings.

Conclusion

This paper presented a design for a flexible open-source environmental sensing platform for monitoring aquatic environments. The buoys are intended for calm, shallow near-shore fresh or brackish water environments. The system uses an Arduino Mega 2560 microcontroller with Adafruit TSL2561 lux and MCP9808 temperature sensors and a custom-built light attenuation turbidity sensor. A TinySine 3G GSM module provides telemetry and data is displayed via a ThingsBoard IoT dashboard. The system is powered by solar charged up-cycled lithium-ion batteries. On-board timing circuitry completely powers the system down between duty cycles resulting in a system with low energy consumption that can be deployed up to 18 months before requiring servicing.

The hardware (circuitry) design and basic software logic for the system is demonstrated (all files are available in the online repository). The hardware aspires to be general-purpose so that future environmental monitoring applications can repurpose the electronics by adding new compatible sensors and modifying the physical design to match the requirements. Energy consumption data and calibrated turbidity sensor data is presented from an actual deployment to validate the system’s operation over time. To our knowledge this is the first documented open-source design for a remote aquatic environmental monitoring system (i.e., a non-commercial, research-based product).

Future work involves studying the fouling rates of differing water bodies and determining a dynamic calibration adjustment to offset the fouling error. This will allow the sensors to be deployed for longer periods between maintenance cycles. We will also investigate power efficient methods of using the GPS features of the TinySine to provide periodic georeferencing of a buoy’s location. Furthermore, we will demonstrate techniques in future work for establishing two-way communications with the buoy to enable system configuration and firmware updates. [19] outlines additional work done to this platform to incorporate pH, dissolved oxygen, and electrical conductivity sensors.
Design files

Design files can be found here: https://osf.io/nhj4z.

Ethical statement

This work did not involve human subjects or animal experiments.

CRediT authorship contribution statement

Jarrod Trevathan: Conceptualization, Methodology, Software, Data curation, Software, Writing – original draft. Simon Schmidtke: Conceptualization, Investigation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank James Taylor for his work on software development.

Funding

This work was supported by the Australian Research Council Linkage Project [LP190101083], Logan City Council Enviro-Grants scheme [EG2018014] and Queensland Engaging Science [ESG0642021].

References:

[1] T.R. Consi, G. Anderson, G. Barske, H. Bootsma, T. Hansen, J. Janssen, V. Klump, R. Paddock, D. Szmania, K. Verhein, J.T. Waples, 2008. Real time observation of the thermal bar and spring stratification of Lake Michigan with the GLUCOS coastal observatory. IEEE Conference OCEANS, Quebec, Canada, 1-9.
[2] Z. Guo, F. Hong, H. Feng, P. Chen, X. Yang, M. Jiang, (2008). OceanSense: Sensor Network of Realtime Ocean Environmental Data Observation and Its Development Platform. 3rd ACM International Workshop on UnderWater Networks, 101-105.
[3] C. Albala-dejo, P. Sánchez, A. Iborra, F. Soto, J.A. López, R. Torres, Wireless sensor networks for oceanographic monitoring: a systematic review, Sensors 10 (7) (2010) 6948–6968.
[4] J. Matos, O. Postolache, IoT enabled aquatic drone for environmental monitoring, in: In 2016 IEEE International Conference and Exposition on Electrical and Power Engineering, 2016, pp. 598–603.
[5] G. Lockridge, B. Dzwonkowski, R. Nelson, S. Powers, Development of a low-cost Arduino-based sonde for coastal applications, Sensors 16 (4) (2016) 528.
[6] C.A. Pérez, F.S. Valles, R.T. Sánchez, M.J. Buendía, F. López-Castejón, J.G. Cervera, Design and deployment of a wireless sensor network for the minor coastal observation system, IEEE J. Oceanic Eng. 42 (4) (2017) 966–976.
[7] F. Wu, C. Rudiger, M.R. Yuce, Design and field test of an autonomous IoT WSN platform for environmental monitoring, in: 27th IEEE International Telecommunication Networks and Applications Conference (ITNAC), 2017, pp. 1–6.
[8] P.A. Beddows, E.K. Mallon, Cave pearl data logger: a flexible Arduino-based logging platform for long-Term monitoring in harsh environments, Sensors 18 (2) (2018) 530.
[9] J. Trevathan, R. Johnstone, Smart Environmental Monitoring and Assessment Technologies (SEMAT)—A New Paradigm for Low-Cost, Remote Aquatic Environmental Monitoring, Sensors 18 (7) (2018) 2248.
[10] A.D. Wickert, C.T. Sandell, B. Schulz, G.-H. Ng, Open-source Arduino-compatible data loggers designed for field research, Hydrol. Earth Syst. Sci. 23 (4) (2019) 2065–2076.
[11] J.S. Horsburgh, J. Caraballo, M. Ramírez, A.K. Aufdenkampe, D.B. Arscott, S.G. Damiano, Low-cost, open-source, and low-power: But what to do with the data?, Front Earth Sci. 7 (67) (2019) 1.
[12] G. Xu, Y. Shi, X. Sun, W. Shen, Internet of things in marine environment monitoring: a review, Sensors 19 (7) (2019) 1711.
[13] D. Menaka, S. Gauni, Ocean of things: Marine environment monitoring using discriminatory model, J. Phys. Conf. Ser. 1964 (7) (2021) 072015.
[14] M.M. Islam, M.A. Kashem, J. Uddin, An internet of things framework for real-time aquatic environment monitoring using an Arduino and sensors, Int. J. Electr. Comput. Eng. 12 (1) (2022) 826.
[15] J. Trevathan, S. Schmidtke, W. Read, T. Sharp, A. Sattar, 2021. An IoT General-Purpose Sensor Board for Enabling Remote Aquatic Environmental Monitoring. Internet of Things, 16, 100429.
[16] S. Adam, The Emergence of Social Enterprise, Vol. 4, Psychology Press, 2004.
[17] J. Defourny, M. Nyssens, 2006. Defining social enterprise. Social enterprise: At the crossroads of market, public policies and civil society, 7, 3-27.
[18] J. Trevathan, T. Sharp, 2020. Up-Cycling E-Waste into Innovative Products Through Social Enterprise. 9th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), 185–193.
[19] J. Trevathan, D. Nguyen, 2021. Adaptive IoT Technology for Measuring Salinity, Dissolved Oxygen and pH in Aquatic Environments. International Journal of Hyperconnectivity and the Internet of Things. 6(1), 1-20.
[20] J. Trevathan, W. Read, S. Schmidtke, Towards the development of an affordable and practical light attenuation turbidity sensor for remote near real-time aquatic monitoring, Sensors 20 (7) (2020) 1993.
Dr Jarrod Trevathan is an expert in affordable environmental monitoring technologies and ecommerce security and fraud algorithms. His current particular area is cost-effective technologies for water quality assessment and flood monitoring. Dr Trevathan is leading industry and community collaborative projects that seek to integrate technology, social enterprise and community engagement to improve environmental outcomes. This is particularly the case with water resource monitoring and disaster management (for which he has won significant community awards and prestigious Australian Research Council (ARC) funding).

Mr Simon Schmidtke is the led project officer and innovator at Substation33. He has a strong background in electrical engineering, power management and 3D printing technology. Mr Schmidtke has managed multiple large-scale commercial IoT project in sustainable technology, road safety and e-waste up-cycling.