



SNAPPERGPS: Open Hardware for Energy-Efficient, Low-Cost Wildlife Location Tracking with Snapshot GNSS

HARDWARE
METAPAPER

JONAS BEUCHERT

AMANDA MATTHES

ALEX ROGERS

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

Location tracking with global navigation satellite systems (GNSS), such as GPS, is used in many applications, including the tracking of wild animals for research. Snapshot GNSS is a technique that only requires milliseconds of satellite signals to infer the position of a receiver. This is ideal for low-power applications such as animal tracking. However, there are few existing snapshot systems, none of which is open source.

To address this, we developed SNAPPERGPS, a fully open-source, low-cost, and low-power location tracking system designed for wildlife tracking. SNAPPERGPS comprises three parts, all of which are open-source: (i) a small, low-cost, and low-power receiver; (ii) a web application to configure the receiver via USB; and (iii) a cloud-based platform for processing recorded data. This paper presents the hardware side of this project.

The total component cost of the receiver is under \$30, making it feasible for field work with restricted budgets and low recovery rates. The receiver records very short and low-resolution samples resulting in particularly low power consumption, outperforming existing systems. It can run for more than a year on a 40 mAh battery.

We evaluated SNAPPERGPS in controlled static and dynamic tests in a semi-urban environment where it achieved median errors of 12 m. Additionally, SNAPPERGPS has already been deployed for two wildlife tracking studies on sea turtles and sea birds.

METADATA OVERVIEW

Main design files: <https://github.com/SnapperGPS/snappergps-pcb>

Target group: biologists tracking animal movement

Skills required: PCB manufacturing and assembly (can be outsourced) – advanced;

Replication: this hardware has been replicated by every author. See section “Build Details” for more detail.

CORRESPONDING AUTHOR:

Jonas Beuchert

University of Oxford, UK

beuchert@robots.ox.ac.uk

KEYWORDS:

conservation technology; wildlife tracking; satellite navigation; low power; low cost; open source

TO CITE THIS ARTICLE:

Beuchert, J, Matthes, A and Rogers, A. 2023. SNAPPERGPS: Open Hardware for Energy-Efficient, Low-Cost Wildlife Location Tracking with Snapshot GNSS. *Journal of Open Hardware*, 7(1): 2, pp. 1–13. DOI: <https://doi.org/10.5334/joh.48>

INTRODUCTION

Animal location trackers most commonly use global navigation satellite systems (GNSS) such as the Global Positioning System (GPS). These systems use constellations of satellites which continuously broadcast radio signals containing precise transmission timestamps and their ephemeris data describing their orbits. A GNSS tracking device on Earth captures these signals and infers its location from this information. A traditional GNSS receiver usually requires more than one minute of data for a first fix (for which it needs to decode both, ephemerides and timestamps), and at least several seconds for subsequent fixes (if recently decoded ephemerides are available) (van Diggelen 2009). However, powering the radio to capture these signals and then the processor to calculate the location is energy expensive, resulting in the need for large batteries with high capacity. This often makes traditional GNSS tags impractical for tracking small animals over long deployment periods (McMahon et al. 2017).

Assisted GNSS (A-GNSS) receivers address this issue by obtaining some of the satellite data in another way, which allows them to reduce their on-time, thereby saving energy. This is done either by pre-loading ephemerides before the deployment or by regularly downloading them via another connection (e.g. cellular). However, pre-loading data is only possible for short deployments and an additional download link requires more expensive hardware and is energy intensive.

Snapshot GNSS is another alternative GNSS concept, which, by design, has significantly lower energy needs, resulting in small, light-weight, energy-efficient, and low-cost receivers. Instead of capturing seconds or even minutes of the satellite signals for a fix, a snapshot receiver records just a few milliseconds. This reduces its power consumption by several orders of magnitude compared to a traditional GNSS approach. Additionally, a snapshot GNSS device does not need to calculate its position on-board, saving even more energy and lowering the requirements for its computing hardware. Instead, it can just locally store the raw signal snapshots until the deployment ends. Afterwards, the data processing can be off-loaded to the cloud.

Table 1 Existing snapshot GNSS systems.

^aData from commercial GPS front-end, not snapshot receiver.

^bRooftop with good sky visibility.

^cEvaluation by McMahon et al. (2017).

^dEvaluation board.

	CO-GPS	BASEBAND TECHNOLOGIES	ETH ZÜRICH	ATS G10 ULTRALITE GPS
Reference	Liu et al. (2012)	Baseband Technologies Inc. (n.d.)	Eichelberger et al. (2019)	Morrison (n.d.)
Memory	8 MBit ≤ 1,000 snapshots	4 GB ≤ 2,000,000 snapshots	2 GB 65,600 snapshots	4 GB ≤ 244,000 snapshots
Maximum deployment duration	1.5 years (2 AA batteries, 1 fix/s)	18 days–1 year (10 mAh) weeks (coin cell) years (phone battery)	683 days (coin cell, 235 mAh, 4 fixes/h)	80 min (19 mAh, 1 fix/s) –759 days (200 mAh, 1 fix/h)
Snapshot duration	5 chunks of 2 ms	2–20 ms	1–30 ms	?
Quantisation	2 × 2 bit	?	2 bit	?
Sampling frequency	16.368 MHz	?	16.368 MHz	?
Accuracy	< 35 m ^a	median < 9 m	< 25 m ^b	mean 15.5 m ^c
Weight	?	?	1.3 g	11 g
Size [mm]	70 × 52	22 × 27	23 × 14	32 × 23 × 12
Available (2022)	no	yes ^d	no	no
Price	N/A	189 USD	N/A	N/A
Open source	no	no	no	no

Table 1 details pure snapshot GNSS systems that have been developed in the past. Only BASEBAND TECHNOLOGIES currently offers its solution for purchase (Baseband Technologies Inc. n.d.). It is available as a proprietary development board and, therefore, not readily available for deployments. Only the ATS G10 module has been evaluated in a realistic scenario, but was found to be unreliable by McMahon et al. (2017) due to battery and software failures (Morrison n.d.). Furthermore, all existing solutions also use high sample rates and/or long

snapshots at multi-bit resolution. This improves the snapshot quality but also requires complex and expensive hardware. For example, the ETHZ receiver records two-bit signals sampled at 16 MHz, which limits their microcontroller choice to one with a parallel input capture interface (PARC) (Eichelberger et al. 2019). The CO-GPS receiver uses additional circuitry to convert its two-bit GPS input stream at 16 MHz into a 16-bit parallel signal at 2 MHz, which a microcontroller then captures using direct memory access (DMA) (Liu et al. 2012).

Moreover, the hardware is only part of a full snapshot GNSS solution. A complete system also needs a signal processing chain to calculate positions from the raw snapshot data. However, this software is not openly available for any of these systems, which renders users dependent on the technology provider over the whole lifespan of their devices.

To address these issues, we developed SNAPPERGPS, a completely open-source, small, low-cost, low-power snapshot GNSS system designed for tracking wildlife. The algorithmic details of the SNAPPERGPS cloud-processing chain have already been described (Beuchert & Rogers 2021a). This paper presents the accompanying hardware. The SNAPPERGPS printed circuit board (PCB) measures 32.0 mm by 27.3 mm and weighs 3 g. The total weight, including an antenna and a 40 mAh lithium-ion polymer battery, is 9 g. With such a battery, SNAPPERGPS can run for more than a year. SNAPPERGPS works with short 12 ms snapshots sampled at only 4 MHz with 1-bit amplitude quantisation. This unmatched low resolution allows us to use low-cost, off-the-shelf components. The total component cost adds up to \$21 for each device if ordered in a batch of 100, excluding battery and antenna. SNAPPERGPS tags are used with an open-source web application. The app serves as a tool for configuring the tag and later for uploading the recorded snapshots to the cloud for processing. We evaluated SNAPPERGPS in controlled deployments, both moving and stationary. With good sky visibility, SNAPPERGPS achieves a median error of 10 m. Modified versions of SNAPPERGPS have additionally been deployed on free-ranging sea turtles and sea birds, demonstrating its usefulness for wildlife tracking.

The rest of the paper is structured as follows: Section (1) Overview continues with the Overall implementation and design of SNAPPERGPS. It includes sub-sections on the Electronics, the Firmware and the Web application. Section (2) Quality Control covers how to safely operate a SNAPPERGPS receiver and how to test that the system reliably provides accurate location estimates over long periods of time. Examples of use cases are presented in (3) Application and (4) Build Details provides the instructions to replicate SNAPPERGPS. Section (5) Discussion includes a conclusion and an outlook on future work.

OVERALL IMPLEMENTATION AND DESIGN

SNAPPERGPS consists of three components: (i) a purpose-built energy-efficient low-cost receiver, (ii) a web application for configuration of the receiver, and (iii) a cloud-based data processing platform.

Electronics

We place all electronic components on a single side of a two-layered PCB to enable low-cost assembly, as seen in Figure 1. The block diagram in Figure 2 shows all core components, which we describe in the following.

The antenna captures GNSS signals in the GPS L1 band, which has a centre frequency of 1.57542 GHz. There are more GNSS signal bands, but the cheapest ICs work with the L1 band, the oldest civilian band, and we can receive the modernised GPS L1C, the Galileo E1, the BeiDou B1C, and the potential future GLONASS L1OCM signal in this band, too. This allows us to make use of multiple satellite systems and, therefore, to increase localisation reliability by using more satellites.

Sampling and storing satellite signals at a rate of multiple gigahertz is not possible with a low-cost receiver. Therefore, we use the heterodyne method: by mixing the received signal with an unmodulated signal from a local oscillator, we shift the original signal down to a lower, so-called intermediate frequency.

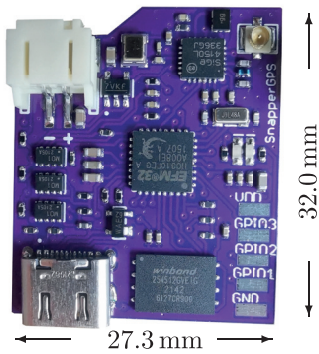


Figure 1 Assembled SNAPPERGPS receiver.

For this, we choose an integrated circuit that is designed for low-power applications, is available at very low costs (<\$1), and is designed for the GPS L1 band: the SE4150L-R from SKYWORKS SOLUTIONS INC. This *radio* includes an analogue-to-digital converter (ADC) and a voltage-controlled oscillator (VCO), which needs a reference frequency of 16.368 MHz from an external temperature-compensated crystal oscillator (TCXO).

To sample the radio's output signal, we connect it to a low-power microcontroller unit (MCU); a 32-bit SILICON LABS HAPPY GECKO with an ARM CORTEX-M0+ core. To reduce the overall number of components, we operate the MCU on the same clock as the radio when acquiring snapshots. Otherwise, we fall back to an internal 14 MHz RC oscillator.

The MCU stores the sampled signals on an external flash memory chip. Its size is the main limitation for the number of fixes. We use a 512 Mbit flash, which stores up to about 11,000 snapshots, 6 KB each.

To configure the board or download data, it must be connected to a host device. For this, we add a USB-C connector and use the internal USB hardware of the MCU.

The PCB also has two LEDs to provide information on the state of the device when it is not connected via USB.

The board draws its power from either USB or a rechargeable lithium-ion polymer (LiPo) battery. The latter's high power density allows to reduce weight. It provides 3.0 V–4.2 V, while the USB power line is at 5 V. Therefore, we need a power regulator and a MOSFET-based switch to provide a stable 3.3 V supply voltage. There are also MOSFETs to switch off the radio and the flash to minimise quiescent currents. Finally, we add a circuit to measure the battery voltage.

We expose three general-purpose input/output pins (GPIOs) of the MCU at an edge of the PCB to allow for the connection of external modules to extend the functionality of the device. We also expose the reset pin and the Serial Wire Debug (SWD) pins.

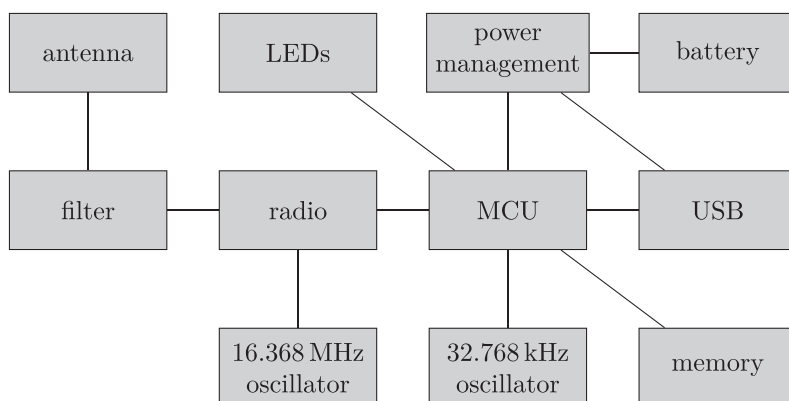
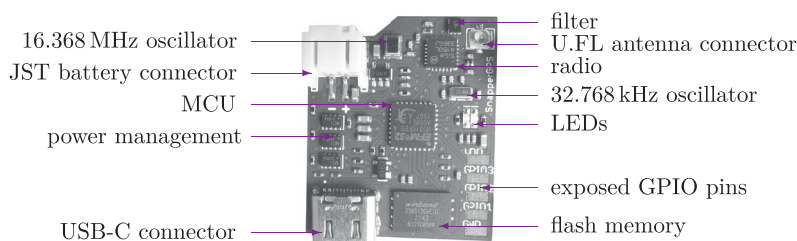


Figure 2 The main electronic components of a SNAPPERGPS receiver.

The HAPPY GECKO MCU runs custom firmware written in the C programming language. The state machine in Figure 3 visualises its core functionalities. To minimise the overall energy consumption, we carefully select for each state one of the five energy modes (EMs) that the HAPPY GECKO offers (Silicon Laboratories Inc. 2015). Almost all the time during a deployment, the MCU sleeps in stop mode (EM3), which consumes only 0.5 μ A. A real-time counter (RTC) triggers wake-ups in regular intervals to record a satellite signal snapshot, respectively, and the MCU switches to run mode (EM0) for this. The flow chart in Figure 4 shows the steps that it completes in this mode before it goes to sleep again.

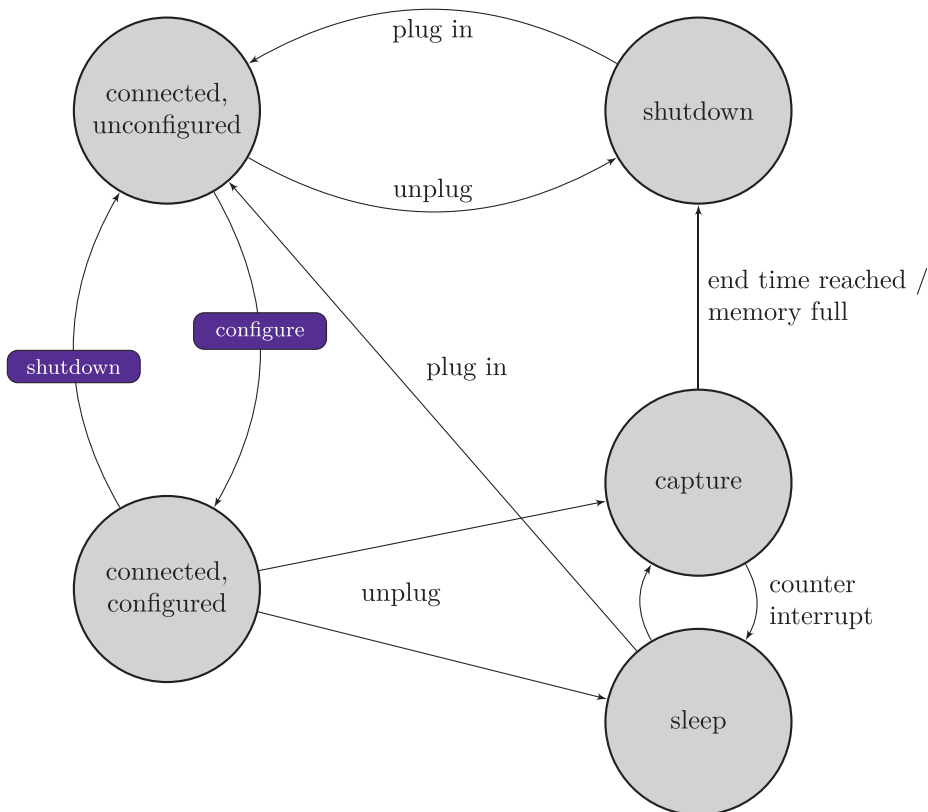


Figure 3 State machine of a SNAPPERGPS receiver.

The radio outputs satellite signal samples at 16.368 MHz with two-bit amplitude quantisation. However, we sample the signal at a low 4.092 MHz rate, a quarter of the clock frequency. Additionally, we use only one bit for the amplitude quantisation. Both together reduce the amount of required memory by a factor of eight. However, the main advantage is that we can capture the one-bit receiver output using the existing device USART configured as an SPI master. Our choice gives us 32 clock cycles to acquire an eight-bit sample and write it to RAM. The small RAM size of our low-cost MCU limits the snapshot length to 6 KB, i.e., 12 ms. Afterwards, we write the snapshot together with a timestamp and measurements of the battery voltage and the temperature via SPI to the external flash memory. Subsequently, the MCU goes to sleep in EM3 again or switches to shutoff mode (EM4) if all snapshots have been captured and the deployment is complete. In shutoff mode, the RTC is also powered down and the MCU consumes just 20 nA.

To preserve energy, we switch off the high-frequency 16.368 MHz oscillator during sleeping because we only need a high frequency clock while capturing a snapshot and the low-frequency 32.768 kHz oscillator is sufficient to keep track of time between consecutive snapshots and the RTC running.

We configure a GPIO with a pull-up resistor to be pulled low when a USB connector is plugged in. Then, the MCU leaves EM3 or EM4, stops the RTC and, therefore, the recording of further snapshots, and switches to EM0. A SNAPPERGPS receiver provides USB bulk endpoints in both directions to facilitate fast transfers of larger amounts of data. These enable it to be configured as a WebUSB device, for example, together with the SNAPPERGPS web application, which is linked in the USB descriptor. The documentation in the snappergps-firmware repository defines the

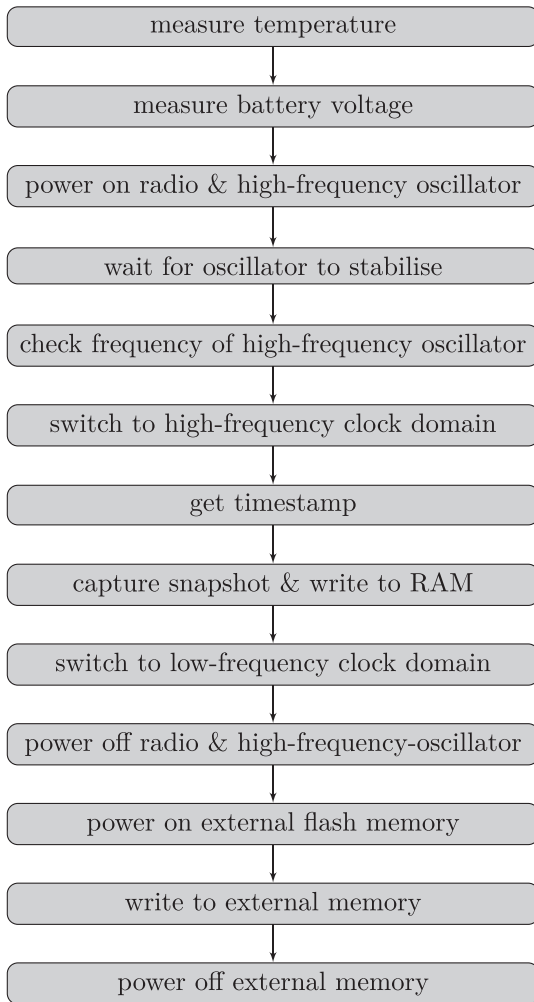


Figure 4 Flow chart for capturing a snapshot.

messages to communicate with a SNAPPERGPS receiver via USB. The messages can be used to get status information from the receiver, configure a receiver for a deployment, and transfer snapshots and timestamps to a host computer. We also use USB to send a firmware update via the MCU to the external flash memory and to request a cyclic redundancy check (CRC) afterwards to detect transmission errors. Next, a custom bootloader function that is held in SRAM copies the firmware update from the external flash memory to the flash memory of the MCU before the MCU reboots to complete the update.

Cloud-processing back-end

The actual processing of the GNSS signals snapshots, i.e., the location estimation, is done in the cloud rather than on the device itself (see Figure 5). For this, we wrap the open-source snapshot GNSS algorithms developed by Beuchert & Rogers (2021a) into an open-source Python back-end that runs on a server. The back-end pulls snapshots that have been uploaded from a SNAPPERGPS receiver from a relational database, processes them, and puts back location estimates.

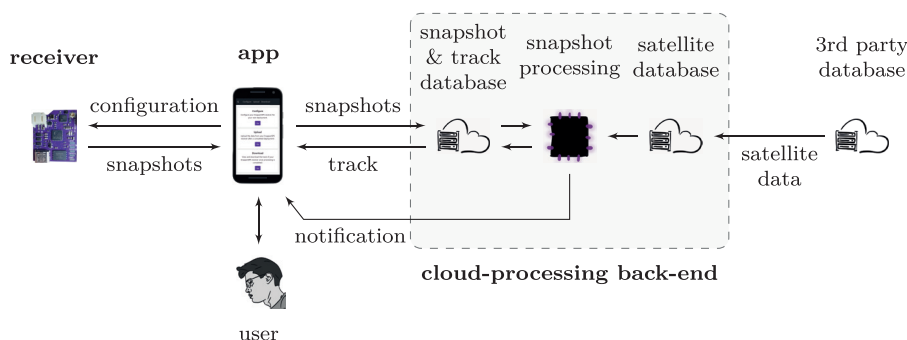


Figure 5 The SNAPPERGPS system comprising receiver, web application, and cloud-processing back-end.

Web application

A front-end written in JavaScript functions as an interface between SNAPPERGPS receiver, user, and back-end. It is a progressive web app (PWA), which means that it can run in a browser window (see Figure 6), but also as native app. Additionally, PWAs load quickly, require little memory, are secure to use, and a service worker automatically keeps the app up-to-date.

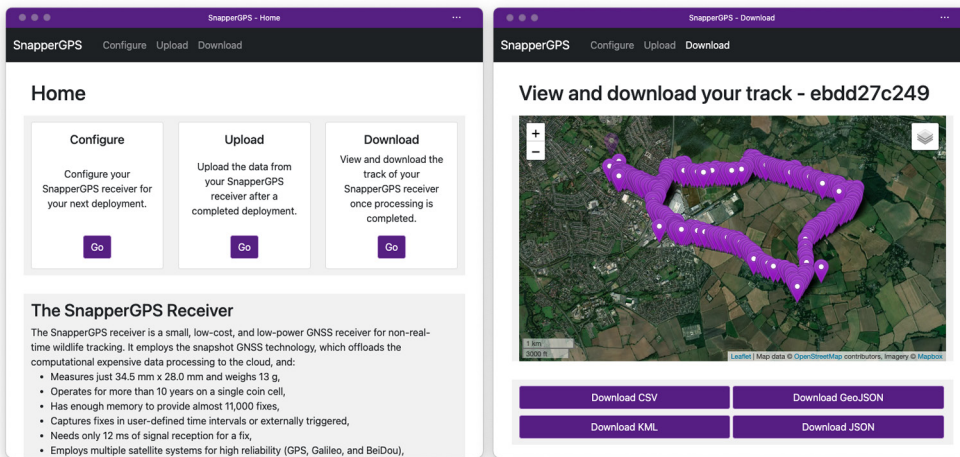


Figure 6 Example views of the SNAPPERGPS app (<https://snappergps.info>).

The SNAPPERGPS app communicates with a SNAPPERGPS board via WebUSB, thus not requiring any driver installation.

A configuration view allows to read out various information, including the battery voltage and the number of signal snapshots on the receiver. Furthermore, it is used to set parameters such as the device clock/time, the time interval between two consecutive snapshots, and the start and end date and time of the next deployment. It can also be used to update the firmware via USB (see the firmware description presented earlier).

After a deployment, an upload view transfers snapshots and timestamps from the device to a server.

Finally, a download view shows the track on an interactive map once the back-end has calculated it and the user can download it in four different file formats.

(2) QUALITY CONTROL

SAFETY

A lithium-ion polymer battery can overheat or catch fire when short-circuited, charged too quickly, overcharged, or damaged. Always use a battery with in-built circuit protection and follow the safety instructions of the supplier.

CALIBRATION

A SNAPPERGPS receiver does not require calibration despite the fact that it is made of low-cost components with relatively wide tolerances. This is because the software automatically estimates and corrects errors.

The main components of concern are the two oscillators. The SNAPPERGPS receiver uses the 32.768 kHz oscillator to time stamp snapshots. The snapshot GNSS method requires timestamps with an accuracy of at least 1 min (van Diggelen 2009). However, the 32.768 kHz oscillators that we use have a frequency tolerance of at least ± 10 ppm, which can cause an error of over 5 min during a year-long deployment. Our snapshot GNSS back-end mitigates this by estimating the time error for every single snapshot and using the previous estimate as initialisation for the next one. Since the on-board time is synchronised with the browser time during configuration, this propagation is sufficient to keep track of the time error during the location estimation process.

The synthesizer of the radio turns the 16.368 MHz reference frequency of the external TCXO into a frequency close to the GPS L1 frequency of 1575.42 MHz. Therefore, frequency errors of the TCXO are amplified by almost a factor 100 and a frequency tolerance of ± 500 ppb can turn into a frequency offset of the front-end of more than 800 Hz, which shifts the intermediate frequency of the captured snapshots by the same amount. However, for successful satellite acquisition, we need an accuracy that is a magnitude better than that, especially for Galileo and BeiDou satellites. To avoid searching a large frequency space for the satellite signals, the back-end automatically estimates the offset of the receiver front-end for each uploaded dataset using a couple of snapshots from the beginning of the deployment (Beuchert & Rogers 2021b). Afterwards, it corrects the offset estimate over time based on the recorded temperatures and a linear model that relates temperature and frequency offset and which we fitted to a training dataset.

Both steps together allow us to obtain reliable localisation accuracy without manual calibration.

GENERAL TESTING

Localisation accuracy

To test the accuracy of the location estimates of a SNAPPERGPS receiver, we advise users to just record some data outdoors at a location with known coordinates and then upload it via the SNAPPERGPS app. Then they can calculate the horizontal error between the estimated locations and the ground truth. For comparison, we obtained median tracking errors of 11–12 m on an open test data collection that we established by ourselves using three SNAPPERGPS receivers. It consists of four static and seven dynamic cycling tests in urban and rural environments with 3700 snapshots in total (Beuchert & Rogers 2021a,b). In addition, a receiver that we mounted on a roof top with good sky visibility collected about 8000 snapshots over more than five months and achieved a median error of 10 m (see Figure 7).

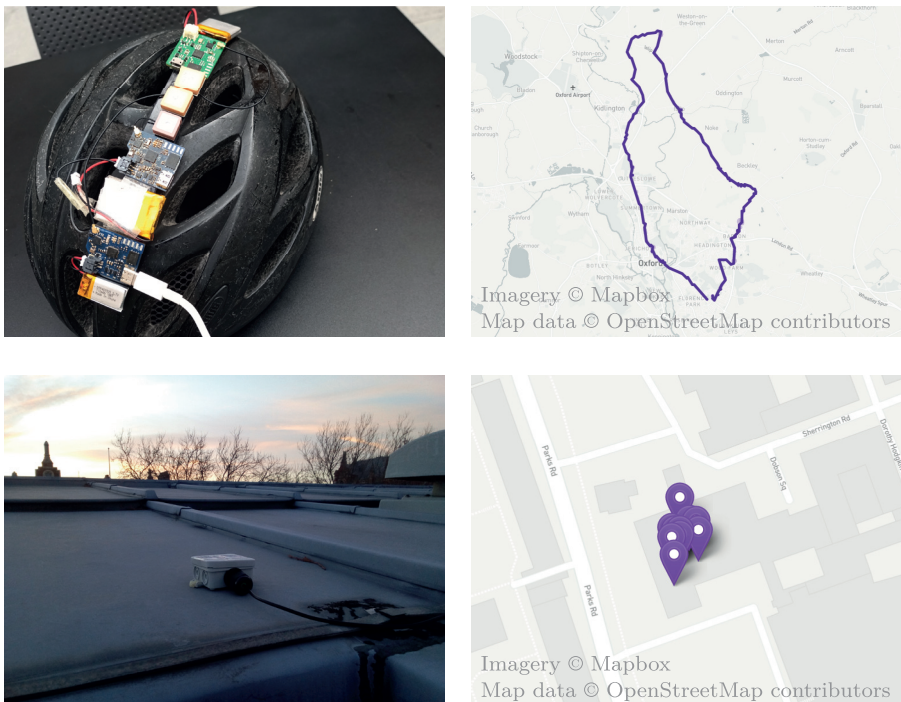


Figure 7 Three SNAPPERGPS receivers on a cycling helmet (top left), example track captured while cycling and walking in and around Oxford, UK (top right, from <https://snappergps.info/view?uploadid=8821aa36c3>), roof-mounted SNAPPERGPS receiver for stationary data collection over six months (bottom left), and example set of ten fixes (bottom right).

Power consumption

Measuring the power/current consumption of a SNAPPERGPS receiver is critical for long-term deployments to ensure that it will work for the desired operation time, although none of our test boards had problems in this regard. The current should be $1 \mu\text{A}$ – $2 \mu\text{A}$ if the board is sleeping and the maximum current should be around 25 mA when capturing a snapshot. The charge consumption for a single snapshot should be $<0.3 \mu\text{Ah}$.

(3) APPLICATION

USE CASES

SNAPPERGPS could be used for any location data logging application, however, the snapshot GNSS approach has advantages over traditional GNSS that make it particularly well suited for wildlife tracking. Most notable is the significantly reduced power consumption which means that smaller batteries can be used. Minimising size and weight of a tag is crucial when tagging wildlife to avoid unnecessary stress to the animal. Furthermore, a snapshot receiver is capable of taking a snapshot at any time and only requires milliseconds of clear signal (as opposed to multiple seconds with a traditional receiver). These attributes make the snapshot approach particularly well suited for marine settings, where animals may only surface sparsely and briefly.

Marine deployment

In summer 2021, SNAPPERGPS tags were deployed on loggerhead sea turtles (*Caretta caretta*) on the island of Maio, Cape Verde (Matthes et al. 2022). The receivers were placed in enclosures milled out of plastic and aluminium, which we tested to be waterproof to at least 100 m. We open-sourced an updated version of the enclosure design.

The team tagged 20 females with SNAPPERGPS tags while they were nesting on the beach. Nine tags were recovered when the individuals returned to lay another clutch of eggs. Whenever a snapshot was taken above the surface, SNAPPERGPS reliably calculated a position. GNSS signals are easily stopped by water and so any snapshots taken underwater do not contain any usable satellite information. However, the SNAPPERGPS signal processing chain can filter out these poor fixes, see Figure 8.

The obtained tracks are potentially helpful for identifying important habitats that may benefit from further protection.

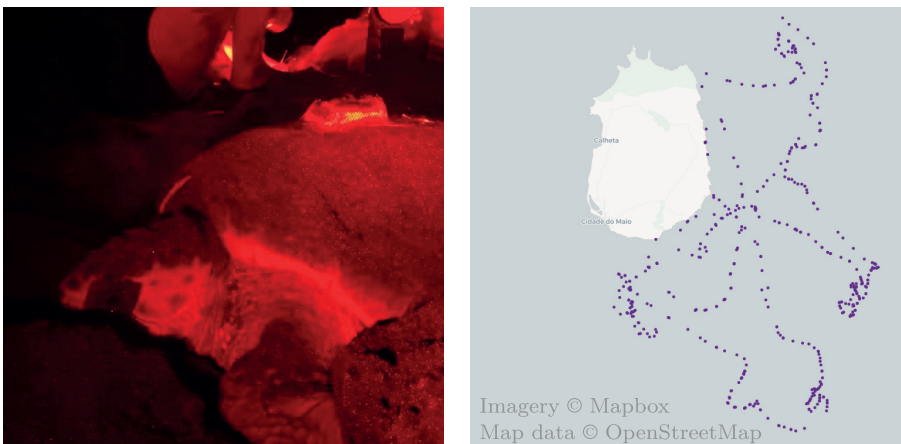


Figure 8 A SNAPPERGPS tag on the back of a loggerhead sea turtle and a track captured over two weeks.

Avian deployment

In May 2022, a custom variant of SNAPPERGPS was deployed on Manx shearwaters (*Puffinus puffinus*) nesting on Skomer, Wales and the Copeland Islands, Northern Ireland. These birds weigh between 400 g and 450 g (Brooke 2013), so minimising the weight of the tag is particularly important. To achieve this, we reduced the size of the PCB by removing the GPIO pads and the JST connector for the lithium-ion polymer battery. The battery is soldered on instead. To avoid having to remove the battery for charging, we added a charging circuit to the PCB. This allows the tag to be charged using the USB port. The altered tag with a 40 mAh lithium-ion polymer battery and an APAM1368YB13V3.0 antenna from Abracon LLC weighs 8.3 g.

The receivers were waterproofed using heat-shrink tubing. The tubing was shrunk with a heat gun and then the ends were sealed with flat needle-nose pliers while the plastic was still warm. In total, 21 tags were deployed and 19 were recovered over the incubation and chick rearing period. The recorded data has high enough temporal resolution to identify behaviour. In particular, foraging behaviour shows up in the data as erratic movement which clearly distinguishes it from sustained flight or rafting on the surface. This makes it possible to monitor behaviour and locate major foraging sites of the species during the nesting season without having to co-deploy other sensors such as accelerometers.

See Figure 9 for a photograph of a SNAPPERGPS receiver on a bird and a plot of one recorded track.



Figure 9 A custom SNAPPERGPS tag on the back of a Manx shearwater and a recorded track.

REUSE POTENTIAL AND ADAPTABILITY

SNAPPERGPS, as it is presented here, is designed to be general-purpose and low-cost. However, as demonstrated in the previous section, the SNAPPERGPS hardware can be customised for specific applications. This is a key advantage of a fully open-source solution.

The open-source design has multiple GPIO pins to extend functionality of the tag without altering the PCB layout. These can be used to connect a sensor such as an accelerometer, a calibrated thermometer or a light sensor. The additional sensor data can then be logged alongside the GNSS data. Such additional sensors can also be used to trigger snapshots.

Other potentially useful changes to the PCB design may include switching out the flash storage for a larger alternative (e.g. a microSD card), and swapping the power source from a LiPo battery to something that is more readily available (e.g. a coin cell).

(4) BUILD DETAILS

AVAILABILITY OF MATERIALS AND METHODS

Table 2 presents the bill of materials (BOM) for the electronic components of a SNAPPERGPS receiver. The price of \$21 for a single device in a batch of 100 is significantly cheaper than both existing commercial and open-source GNSS receivers for wildlife tracking, which usually cost a few hundred USD.

Table 2 Bill of materials for one, 100, and 1000 SNAPPERGPS receivers.

COMPONENT	VALUE	PART	QTY	PRICE [USD]		
				ONE	100	1000
Ceramic capacitor	100 nF	GRM155R60J104KA01D	10	0.33	14.60	81.70
Ceramic capacitor	10 nF	GRM155R60J103KA01D	5	0.50	4.43	29.10
Ceramic capacitor	100 pF	GRM1555C1H101JA01J	1	0.10	0.81	4.54
Ceramic capacitor	22 pF	GRM1555C1H220JA01D	1	0.10	0.81	4.57
Ceramic capacitor	18 pF	GRM1555C1H180JA01D	2	0.20	2.32	13.00
Multi-layer ceramic capacitor	1 uF	GRM188R60J105KA01D	4	0.40	11.00	61.84
Multi-layer ceramic capacitor	4.7 uF	GRM185R60J475ME15D	1	0.30	11.67	72.02
Multi-layer ceramic capacitor	10 uF	GRM188R60J106KE47D	1	0.15	5.07	29.44
Fixed inductor	39 nH	0402HS-390EKTS	1	0.26	15.68	109.72
Ferrite bead		BLM15HB221SH1D	1	0.10	4.36	25.08
Resistor	15 Ω	RC0402FR-0715RL	2	0.20	1.32	5.98
Resistor	1 kΩ	RC0402FR-071KL	2	0.20	1.32	5.98
Resistor	4.7 kΩ	RC0402FR-074K7L	3	0.30	1.98	7.77

(Contd.)

COMPONENT	VALUE	PART	QTY	PRICE [USD]		
				ONE	100	1000
Resistor	5.1 kΩ	RC0402FR-075K1L	2	0.20	1.32	5.98
Resistor	6.8 kΩ	RC0402FR-076K8L	1	0.10	0.66	2.99
Resistor	10 kΩ	RC0402FR-0710KL	2	0.20	1.32	5.98
Resistor	100 kΩ	RC0402FR-07100KL	7	0.70	4.62	14.98
Crystal	32.768 kHz	Q 0,032768-JTX310-12,5-10-T1-HMR-50K-LF	1	0.86	62.70	495.00
Yellow-green LED	572 nm	SML-D12M1WT86	1	0.22	5.51	39.86
Red LED	620 nm	SML-D12U1WT86	1	0.22	5.51	39.86
Temp. compensated oscillator	16.368 MHz	D32G-016.368M	1	8.00	700.00	7000.00
GPS receiver		SE4150L-R	1	0.86	86.40	864.00
SAW filter	1.57542 GHz	AFS20A42-1575.42-T3	1	1.05	71.37	543.74
Microcontroller		EFM32HG310F64G-C-QFN32R	1	5.91	435.04	3703.46
TVS diode	5.5 V	PRTR5V0U2AX,235	1	0.66	42.47	257.87
NAND flash memory	512 Mbit	W25N512GVEIG	1	2.90	228.68	2051.92
U.FL connector jack	50 Ω	1909763-1	1	0.51	36.55	595.90
MOSFET array N/P-channel		TT8M1TR	3	1.56	86.58	487.08
Linear voltage regulator		S-1318D33-M5T1U4	1	1.58	109.94	908.20
USB-C receptacle connector		DX07S016JA1R1500	1	1.67	129.65	1080.44
JST connector header		S2B-PH-SM4-TB	1	0.54	39.08	339.84
Total				31.55	2124.54	18900.69

EASE OF BUILD

We intend to support making the SNAPPERGPS hardware available through group purchasing websites. This will be more cost- and time-effective for most potential users.

OPERATING SOFTWARE AND PERIPHERALS

See sections Firmware and Web application.

The SNAPPERGPS app works in the MICROSOFT EDGE and GOOGLE CHROME browsers. Optionally, the user can also install it as a native app, either through the website <https://snappergps.info>, the MICROSOFT STORE (<https://apps.microsoft.com/store/detail/snappergps/9P9RPRS6LSMM>), or GOOGLE PLAY (https://play.google.com/store/apps/details?id=com.herokuapp.snapper_gps.twa).

DEPENDENCIES

The SNAPPERGPS hardware has no dependencies. We provide all firmware dependencies together with the firmware. We list all dependencies of the web app in the respective repositories and they can be installed via pip (Python) or npm (JavaScript). No dependencies are closed-source and all are free software.

HARDWARE DOCUMENTATION AND FILES LOCATION

Name: snappergps-pcb

Persistent identifier: <https://github.com/SnapperGPS/snappergps-pcb>

Licence: Solderpad Hardware License v2.1

Publishers: Jonas Beuchert, Amanda Matthes, Alex Rogers

Date published: 10/06/22

SOFTWARE CODE REPOSITORIES

Names: snappergps-firmware, snappergps-app, and snappergps-backend

Identifiers: <https://github.com/SnapperGPS/snappergps-firmware>,
<https://github.com/SnapperGPS/snappergps-app>, and
<https://github.com/SnapperGPS/snappergps-backend>

Licence: MIT

Date published: 10/06/22

(5) DISCUSSION

CONCLUSIONS

This paper presents SNAPPERGPS, a location data logging system designed for wildlife tracking. It consists of three parts: (i) a GNSS receiver, (ii) a web application that is used to configure it, and (iii) a cloud-based data processing platform. The receiver is small, low-cost and energy-efficient enough to run on small batteries for multiple years. SNAPPERGPS has been tested in multiple controlled static and dynamic tests in urban and rural environments. Additionally, SNAPPERGPS has already been used for tracking sea turtles and sea birds in the wild. These deployments have demonstrated that SNAPPERGPS is a viable wildlife tracking system.

FUTURE WORK

Future work could include: altering the PCB design to further reduce the size and weight of the receiver; switching to a power source that is more readily available; increasing the on-board storage to allow for more snapshots per deployment; and adding commonly used sensors to log additional data (e.g. sound, light, acceleration).

ACKNOWLEDGEMENTS

The development of SNAPPERGPS was supported by Andy Hill and Peter Prince from Open Acoustic Devices. The SNAPPERGPS deployment on loggerhead sea turtles was conducted in cooperation with the Arribada Initiative and the Maio Biodiversity Foundation. Special thanks to Alasdair Davies, Juan Patino-Martinez and Rocio Moreno. We also thank Joe Morford, Patrick Lewin, Katrina Davies, Lewis Fisher-Reeves and Tim Guilford of the Oxford Navigation Group, which conducted the SNAPPERGPS deployment on Manx shearwaters.

FUNDING INFORMATION

SNAPPERGPS was funded by an EPSRC IAA Technology Fund (D4D00010-BL14). Jonas Beuchert and Amanda Matthes are funded by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (DFT00350-DF03.01, DFT00350-DF03.05).

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR CONTRIBUTIONS

JB: Firmware development, web application development, hardware design, prototype assembly, testing, deployments, documentation, paper writing

AM: Prototype assembly, hardware design, testing, deployments, deployment organisation, documentation, paper writing

AR: Concept, hardware design, prototype assembly, firmware development, web application development, testing, project supervision

Jonas Beuchert  orcid.org/0000-0003-1463-2853
University of Oxford, UK

Amanda Matthes  orcid.org/0000-0002-3646-3299
University of Oxford, UK

Alex Rogers  orcid.org/0000-0001-6324-0536
University of Oxford, UK

REFERENCES

- Baseband Technologies Inc.** (n.d.). 'Snapshot positioning – next generation GNSS receiver for low power applications'. URL: <http://www.basebandtech.com/wp-content/uploads/Snapshot-Receiver.pdf> (accessed 24 March 2020).
- Beuchert, J. & Rogers, A.** (2021a). SNAPPERGPS: Algorithms for energy-efficient low-cost location estimation using GNSS signal snapshots. In 'Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems', *SenSys '21*, Association for Computing Machinery, New York, NY, USA, p. 165–177. DOI: <https://doi.org/10.1145/3485730.3485931>
- Beuchert, J. & Rogers, A.** (2021b). 'SNAPPERGPS: Collection of GNSS signal snapshots'. Oxford, UK: University of Oxford. DOI: <https://doi.org/10.1145/3485730.3485931>
- Brooke, M.** (2013). *The Manx Shearwater*. A&C Black.
- Eichelberger, M., von Hagen, F. & Wattenhofer, R.** (2019). Multi-year GPS tracking using a coin cell. In 'Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)', ACM, New York, NY, USA, p. 141–146. DOI: <https://doi.org/10.1145/3301293.3302367>
- Liu, J., Priyantha, B., Hart, T., Ramos, H. S., Loureiro, A. A. F. & Wang, Q.** (2012). Energy efficient GPS sensing with cloud offloading. In 'Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys)', ACM Press, New York, NY, USA, p. 85–98. DOI: <https://doi.org/10.1145/2426656.2426666>
- Matthes, A., Beuchert, J., Davies, A., Patino-Martinez, J. & Rogers, A.** (2022). SNAPPERGPS: Deployment of a low-cost snapshot GNSS receiver to track loggerhead sea turtles. In 'Proceedings of the 40th International Sea Turtle Symposium (ISTS40)', International Sea Turtle Society. URL: <https://ora.ox.ac.uk/objects/uuid:c9acf083-d5e5-4265-8425-67509c5e3b9b>
- McMahon, L. A., Rachlow, J. L., Shipley, L. A., Forbey, J. S., Johnson, T. R. & Olsoy, P. J.** (2017). 'Evaluation of micro-GPS receivers for tracking small-bodied mammals'. *PLOS ONE* 12(3). DOI: <https://doi.org/10.1371/journal.pone.0173185>
- Morrison, C.** (n.d.). 'ATS G10 Ultralite GPS'. URL: <https://nebula.wsimg.com/2c21477b3a3e90c8279c4b5979312195?AccessKeyId=CEA7BFCD16F82D670903> (accessed 24 March 2020).
- Silicon Laboratories Inc.** (2015). 'EFM32HG reference manual'. URL: <https://www.silabs.com/documents/public/reference-manuals/efm32hg-rm.pdf> (accessed 16 March 2020).
- van Diggelen, F.** (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, Norwood, MA, USA.

TO CITE THIS ARTICLE:

Beuchert, J, Matthes, A and Rogers, A. 2023. SNAPPERGPS: Open Hardware for Energy-Efficient, Low-Cost Wildlife Location Tracking with Snapshot GNSS. *Journal of Open Hardware*, 7(1): 2, pp. 1–13. DOI: <https://doi.org/10.5334/joh.48>

Published: 10 February 2023

COPYRIGHT:

© 2023 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Hardware is a peer-reviewed open access journal published by Ubiquity Press.