
Anisca Bird

Release 1.0

Sam Sulaimanov, Raffael Tschui

Feb 13, 2023

CONTENTS

1	Contents	3
1.1	Weight Measurement	3
1.2	Device Status	5
1.3	Webhooks	6
	HTTP Routing Table	11

Anisca Bird replaces the usual roosting place with a weighing and tracking device and can be used with owls, eagles, pigeons, chickens and any other bird that can be ringed.

CONTENTS

1.1 Weight Measurement

Any bird weighing event is recorded and stored to memory as a new log entry. In addition, continuous offset measurements and calibration data are logged. The most important logs are also transmitted wirelessly via SIGFOX (if coverage is available).

1.1.1 Bird Weight

Anisca Bird continuously measures the weight present on the wooden perch (“Detection”). It does so in a low-power fashion until the measured weight is above WAKE_UP_THRESHOLD. The device then attempts to read the RFID tag of a potential bird present. If successful, the weight is measured continuously (“Acquisition”) for as long as it stays above WAKE_UP_THRESHOLD.

Detection

During detection, the following hard-coded constants are used:

- Low-power detection interval: 1 second
- WAKE_UP_THRESHOLD: 50 grams¹
- RFID read attempts: max. 10x in 3 second intervals. If unsuccessful after this, system goes back to sleep.

Acquisition

After a bird was detected and a valid RFID was read out, its weight is sampled continuously. Multiple samples are taken into account because the bird may be moving and hence an instantaneous measurement would not provide a reliable weight value.

The final **raw weight** value is obtained as a result of 3 acquisition steps:

1. **Sub-samples:** The load cell ADC is sampling the weight at 10 Hz.
2. **Samples:** Always 10 sub-samples are averaged to create one sample. A standard deviation on the 10 sub-samples is calculated.
 - a. If the sub-sample standard deviation for a given group of 10 exceeds MAX_SUB_STDEV, the sample gets discarded. This ensures only samples where the bird was sitting still will be considered.

¹ Note: the values in grams are internally converted into raw ADC values by using the last available calibration slope.

- b. If a sample is MAX_SAMPLE_DIFF larger than the previous sample, then the previous sample gets discarded. This ensures all samples to be on a relatively flat line and removes outliers.
3. **Median:** A maximum of N_SAMPLES is collected as described in points 1 & 2. If the bird leaves before reaching N_SAMPLES, a smaller number of samples will be available.
 - a. The median value over all collected samples is calculated. This is the retained **raw weight** value.
 - b. The standard deviation over all collected samples is calculated. This is the retained **raw stddev** value.

Hard-coded parameters:

- MAX_SUB_STDEV: 5 grams^{Page 3, 1}
- MAX_SAMPLE_DIFF: 5 grams^{Page 3, 1}
- N_SAMPLES: 19

Processing

The **raw weight** and **raw stddev** values obtained above are unitless ADC values. They need to be converted to grams using calibration data in order to present meaningful weight data. This is done as follows:

1. The last recorded **raw offset** value is subtracted from the **raw weight**, in order to receive the **net raw weight**
2. **net raw weight** and **raw stddev** are divided by the most recently measured **calibration slope** to obtain weight and standard deviation of the sample in grams.

Logging

For more details on what data is logged or transmitted, see: [Weight](#).

1.1.2 Offset measurement

The **raw offset** is measured in the exact same way as the bird weight, except that it happens in a periodic manner whenever the weight detection stays below the WAKE_UP_THRESHOLD.

The following parameters are hard-coded:

- N_OFFSET_SAMPLES: 11
- Offset acquisition & logging period: 3 minutes
- Offset uplink period (transmitted via SIGFOX): 1-2 hours

1.1.3 Calibration

The calibration is a linear regression between raw ADC values and the weight placed on the perch in grams. We denote the following terms:

- **Slope (a):** the multiplication factor between grams and raw ADC value
- **Intercept (b)** the calculated raw ADC value when zero weight is placed on the perch. Theoretically, this is the offset value during calibration.
- **R^2** the R-square of the linear regression.

When performing a calibration routine, both a and b are calculated automatically by the device and stored for internal use. All subsequent weight measurements are processed using these most up-to-date calibration values.

It is known that the calibration values can change over time as the physical state of the system changes. Therefore, frequent re-calibrations of the system are advised.

Temperature Dependency

It is known that notably the calibration slope is temperature dependent. This dependency is not compensated for by the device. It is advised to perform a temperature compensation of the logged weight values once the relationship between temperature and slope has been established in a post-processing step.

1.2 Device Status

While the anisca device is running, system status data is continuously logged and stored on the onboard memory (“offline”). Some logs are also transmitted wirelessly via SIGFOX (“uplink” - if coverage is available).

Logging frequency:

- offline: every 3 minutes
- uplink: 3x per day

For more details on the logged data fields see: *Status*.

1.2.1 Battery Voltage

The battery voltage is measured directly on the mainboard using a resistive divider and an onboard ADC. The reference voltage is VCC, generated from the 3.3V buck converter.

The ADC value is scaled in order to indicate the battery voltage in mV.

Limits and Thresholds

Rechargeable NiMH batteries should not be discharged below 0.9V per cell. Depending on the measured battery voltage, the device may decide to go to deep-sleep mode in order to protect the batteries.

Below are the hard-coded voltage values that are used to characterize the batteries’ charging state. Note that 4x NiMH are powering the system (nominal 4 x 1.2 V)

- VBAT_FULL: 5350 mV. Batteries are considered to be 100% charged
- VBAT_EMPTY: 3600 mV. Batteries are considered to be 0% charged
- VBAT_SHUTDOWN: 3800 mV. When below this voltage, the system will go to deep sleep and weight logging and message transmission is disabled.
- VBAT_STARTUP_MIN: 4200 mV. At boot, the battery voltage must be at least this value in order for the device to complete start-up, else it will go directly to deep sleep. This means, the inserted batteries must be charged to at least this value.

1.2.2 Uptime

Number of days since last reboot. A reboot happens when the batteries get replaced and upon an unexpected error. If the number of days displayed in the uptime field is smaller than the number of days since the last battery replacement, an unexpected error may have occurred. Please service the device if this is observed multiple times.

1.2.3 Memory usage

The onboard-memory can store up to 3 million log messages. The memory usage is reported as percentage (100 = memory full). The memory can be erased using the anisca App.

1.2.4 Clock difference

The on-board clock may drift by several seconds per day. If the device has SIGFOX connectivity and the clock difference is too high, a clock update is performed once per day. The clock can also be updated by connecting to the device via USB and using the anisca App.

1.3 Webhooks

The primary method that Anisca Bird uses to deliver data is through the use of webhooks. A webhook URL will be called via a POST request with the message encoded as a JSON body.

1.3.1 Messages

Your webhook URL should be prepared to receive the following messages:

weight: Bird detected and weighed

This message is sent when a ringed bird is detected on the perch. A detection is valid if the RFID tag was successful read.

POST **example.com/webhook**

Example request body:

```
{
  "timestamp": 1648642396,
  "serial": "CODEABFFEE",
  "device_id": "secret-red-wake",
  "type": "weight",

  "activation": {
    "latlng": "46.535,6.569",
    "site_id": "rdc20",
    "activation_date": 1648641394
  },

  "body": {
```

(continues on next page)

(continued from previous page)

```

    "duration": 12,
    "flags": "ab",
    "weight": 120.5,
    "uid": "0600c9a463",
    "raw_adc_pos": 56020,
    "stddev": 1,
    "temperature": -10
  }
}

```

Request JSON Object

- **timestamp** (*integer*) – Epoch timestamp (UTC+0): seconds since January 1, 1970 12:00:00 AM
- **serial** (*string*) – device serial number
- **device_id** (*string*) – human readable hash of serial number that is also printed on device label
- **type** (*string*) – message type (one of `weight`, `status`, `offset`, or `calibration`)
- **activation** (*string*) – when and where this device was last activated
- **duration** (*integer*) – number of seconds that owl is on perch (max. 240)
- **flags** (*integer*) – reserved
- **weight** (*integer*) – weight estimate in 0.01 grams (unsigned value)
- **uid** (*integer*) – RFID UID (hexadecimal). Attention: the second byte of the RFID tag is not transmitted in order to save data. Example: The UID 050031b049 will become 0531b049.
- **raw_adc_pos** (*integer*) – raw ADC value from “positive excitation” (incl. offset)
- **stddev** (*integer*) – weight measurement series standard deviation in 0.1 grams
- **temperature** (*integer*) – temperature in degrees Celsius (signed value)

status: Device status

Device status is sent twice a day.

POST `example.com/webhook`

Example request body:

```

{
  "timestamp": 1648642396,
  "serial": "CODEABFFEE",
  "device_id": "secret-red-wake",
  "type": "status",

  "activation": {
    "latlng": "46.535,6.569",
    "site_id": "rdc20",
    "activation_date": 1648641394
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "body": {
      "header": "ff",
      "events_with_id": 29,
      "events_unknown_id": 16,
      "flags": "ab",
      "vbat_mv": 4620,
      "memory_used": 56,
      "days_since_boot": 120
    }
  }
}

```

Request JSON Object

- **timestamp** (*integer*) – Epoch timestamp (UTC+0): seconds since January 1, 1970 12:00:00 AM
- **serial** (*string*) – device serial number
- **device_id** (*string*) – human readable hash of serial number that is also printed on device label
- **type** (*string*) – message type (one of weight, status, offset, or calibration)
- **activation** (*string*) – when and where this device was last activated
- **header** (*integer*) – internal device status header (0xFF)
- **events_with_id** (*integer*) – number of events with successful RFID detection since last status update [0-255]. The value of 255 means ≥ 255 .
- **events_unknown_id** (*integer*) – number of events with UN-successful RFID detection since last status update [0-255]. The value of 255 means ≥ 255 .
- **flags** (*integer*) – reserved
- **vbat_mv** (*integer*) – Battery voltage in mV
- **memory_used** (*integer*) – Percentage of Flash memory used [0-100]
- **days_since_boot** (*integer*) – Number of days since last reboot [0-255]. The value of 255 means ≥ 255 .

offset: Tare weight

This is the tare weight (unladen weight) and is sent every 2 hours.

POST example.com/webhook

Example request body:

```

{
  "timestamp": 1648642396,
  "serial": "CODEABFFEE",
  "device_id": "secret-red-wake",
  "type": "offset",

```

(continues on next page)

(continued from previous page)

```

"activation": {
  "latlng": "46.535,6.569",
  "site_id": "rdc20",
  "activation_date": 1648641376
},
"body": {
  "header": "fa",
  "flags": "ac",
  "weight": 31660,
  "raw_adc_pos": 4999,
  "raw_adc_neg": 4986,
  "stddev": 0,
  "temperature": 20
}
}

```

Request JSON Object

- **timestamp** (*integer*) – Epoch timestamp (UTC+0): seconds since January 1, 1970 12:00:00 AM
- **serial** (*string*) – device serial number
- **device_id** (*string*) – human readable hash of serial number that is also printed on device label
- **type** (*string*) – message type (one of weight, status, offset, or calibration)
- **activation** (*string*) – when and where this device was last activated
- **header** (*integer*) – internal device status header (0xFA)
- **flags** (*integer*) – reserved
- **raw_adc_pos** (*integer*) – raw tare ADC value from “positive excitation”
- **raw_adc_neg** (*integer*) – raw tare ADC value from “negative excitation”
- **stddev** (*integer*) – tare measurement series standard deviation in 0.1 grams
- **temperature** (*integer*) – temperature in degrees Celsius (signed value)

calibration: Calibration result

This message is sent after the user has performed a calibration.

POST example.com/webhook

Example request body:

```

{
  "timestamp": 1648642396,
  "serial": "CODEABFFEE",
  "device_id": "secret-red-wake",

```

(continues on next page)

(continued from previous page)

```
"type": "calibration",
"activation": {
  "latlng": "46.535,6.569",
  "site_id": "rdc20",
  "activation_date": 1648641376
},
"body": {
  "slope": 29,
  "intercept": 25,
  "temperature": 2,
  "r2": 1.0
}
```

Request JSON Object

- **timestamp** (*integer*) – Epoch timestamp (UTC+0): seconds since January 1, 1970 12:00:00 AM
- **serial** (*string*) – device serial number
- **device_id** (*string*) – human readable hash of serial number that is also printed on device label
- **type** (*string*) – message type (one of weight, status, offset, or calibration)
- **activation** (*string*) – when and where this device was last activated
- **slope** (*integer*) – linear regression slope: (raw ADC counts per 10mg)
- **intercept** (*integer*) – linear regression intercept: (raw ADC value)
- **r2** (*float*) – linear regression R^2
- **temperature** (*integer*) – temperature in degrees Celsius (signed value)

HTTP ROUTING TABLE

/example.com

POST example.com/webhook, 9