Oyster 3 LoRaWAN Integration

Revision 1.3 - 13 September 2023

1. INTRODUCTION

This document is a specification for the Digital Matter Oyster 3 LoRaWAN tracker.
Contact info@digitalmatter.com for more information.

1.1. Revision History

Date	Version	Changes
2022-04-08	1.0	Initial release
2022-04-12	1.1	Added downlinks 30-35
2022-04-13	1.2	Corrected ADR default in downlink 9
2023-09-13	1.3	Corrected stats uplink number in downlink 1

2. DEVICE BEHAVIOUR

The Oyster 3's behaviour can be configured by setting a variety of parameters using a USB programming adapter, or by sending a subset of those parameters during a downlink message. The details of these parameters can be found on the product support website.

2.1. Uplink

The Oyster 3 uses an accelerometer to detect movement, allowing it to decide when an asset is in-trip, and when it is stationary. This allows it to schedule the battery hungry GNSS fixes and transmissions as infrequently as possible, to maximize battery life. Each time a status update is scheduled the Oyster 3 will attempt a GNSS fix, then transmit the results (whether the fix succeeded or not). You can configure the Oyster 3 to transmit a status update:

- Periodically (i.e. 24-hour heartbeat)
- At configured times of the day
- At the start of a regular or after-hours trip
- During the trip
- At the end of the trip
- When an inactivity timeout has expired

2.1.1. Device Statistics

The Oyster 3 transmits device statistics messages once every 4 days, to allow monitoring of device activity and associated battery usage. The transmit frequency can be changed to prevent tracking disruption in regions where the transmit duty cycle restrictions are a bottleneck (868 MHz regions can limit transmissions to 1 every 3 minutes).

2.2. Downlink

LoRaWAN potentially allows for a downlink with every regular data uplink, but most networks will only allow a few per day. When a parameter update is sent down to the Oyster 3, it responds with an application layer acknowledgement uplink as soon as possible. In some regions, this could take several minutes. If the next uplink received is not the expected acknowledgement, the downlink should be resent.

2.3. Oyster LoRaWAN Compatibility

The Oyster 3 LoRaWAN is backwards compatible with the Oyster LoRaWAN in its default configuration. If you don't change any of the parameters, the only differences visible at the network layer are:

- The Oyster 3 sends an Hello uplink (Uplink 30) at startup
- The Oyster 3 supports 3.6 V Lithium Thionyl Chloride batteries
  - If you use 1.5 V batteries, all messages are exactly backwards compatible
  - But if you use 3.6 V batteries, voltages in Uplinks 1 and 3 will show their maximum readings. Voltages in Uplink 4 will show readings in a new format which might confuse an integration expecting the older format.
- The Oyster 3 sends an additional battery statistics uplink (Uplink 31)
- The Oyster 3 sends additional bytes in the downlink acknowledgement (Uplink 2)

To make use of new features in the Oyster 3, you must configure it to send one of the new position uplinks (for instance, Uplink 33).

The Oyster 3 supports all of the same downlinks as the Oyster 1, as well as some new ones numbered from 30 onwards. The intention is to allow configuring a mixed deployment of Oysters and Oyster 3s using a single set of downlinks, without having to know which device is which. The original Oysters will simply reject any downlinks that don't apply to them, while the Oyster 3s will accept them all.

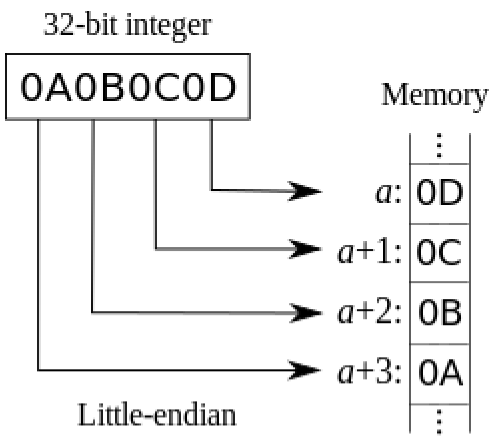
3. MESSAGES

3.1. Number Formats

3.1.1. Little Endian

Except where noted, all data in the payloads is LITTLE ENDIAN. Be aware of this when converting data fields that consist of more than one byte from the data payload.

https://en.wikipedia.org/wiki/Endianness#Little



3.1.2. Signed (Negative) Numbers

When a field is specified as *signed*, it is represented in 'two's complement' form. Be aware of this when converting signed fields from the data payload. Where unspecified, assume that fields are unsigned.

https://en.wikipedia.org/wiki/Two's_complement

3.2. **Oyster 1 Compatible Uplinks**

LoRaWAN uplink payloads can be as small as 11 bytes in some regions (for the longest-range transmissions). The packet headers already include the device serial number, and a 'port number' from 1 to 223, which we will use as a message type.

3.2.1. Uplink Port 1: GNSS Data Record

Offset	Description
0 - 3	32 bit latitude, **signed**, LSb = 0.000'000'1°. To convert, first calculate the signed integer value, then divide by 10 million to get a floating-point value.
4 - 7	32 bit longitude, **signed**, see above
8.0	0: Out of trip, 1: In-trip
8.1	Last fix failed
8.2 - 8.7	Heading, LSb = 5.625°
9	Speed, LSb = 1 km/h
10	Battery voltage, LSb = 25 mV

This is the default status message, sent if the *Inactivity Indicator* feature is not enabled, and the uplink type is left on the default setting of Automatic. When the *Inactivity Indicator* feature is enabled, the Oyster 3 will send Uplink 4 instead.

Note that when an Oyster 3 is fitted with Lithium Thionyl Chloride batteries, Uplink 1 is not capable of representing the full battery voltage, and will read its maximum value of 6.375 V. It is therefore advisable to configure the Oyster 3 for a different uplink type if you wish to use LTC batteries. However, Uplink 1 is the default for backward compatibility with the Oyster 1, which does **not** support high voltage LTC batteries.

Example: 53AB783C0421F98E940AB3

- 53AB783C little endian
  - 3C78AB53 in hex
  - 1014541139 in decimal
  - 101.4541139°
- 0421F98E little endian
  - 8EF92104 in hex
  - 2398691588 in decimal (unsigned), >= 2^{31}
  - 2398691588 - 2^{32} = -1896275708 in decimal (signed)
  - -189.6275708°
- 94
  - 10010100 in binary

    - 0 - not in trip
    - 0 - last fix did not fail
    - 100101 - 37 x 5.625 = 208.12°
- 0A
    - 10 km/h
- B3
    - 179 x 25 = 4475 mV

3.2.2. Uplink Port 2: Downlink Ack

Offset	Description
0.0 - 0.6	Sequence number (identifies downlink to server)
0.7	0: Downlink rejected, 1: Downlink accepted
1	Firmware major version
2	Firmware minor version
3	Product Id (98)
4	Hardware revision
5	Downlink port number

Note that the last 3 bytes of this uplink are newly introduced in the Oyster 3. They were not present in the original Oyster 1.

Example: D3010262010A

- D3
    - 11010011 in binary
    - 1010011 - sequence number 83
    - 1 - downlink accepted
- 0102
    - Firmware 1.2
- 6201
    - Product 98 (Oyster 3), hardware revision 1
- 0A
    - Acknowledging a downlink on port 10

3.2.3. Uplink Port 3: Device Statistics

Offset	Description
0.0 - 0.3	Initial battery voltage, LSb = 100 mV, 4V offset
0.4 - 1.6	Tx count, LSb = 32 Tx attempts
1.7 - 3.3	Trip count, LSb = 32 trips
3.4 - 4.5	GNSS Successes, LSb = 32 fixes
4.6 - 5.5	GNSS Failures, LSb = 32 failed fixes
5.6 - 6.6	Average GNSS fix time, calculated over successful fixes only, seconds
6.7 - 7.7	Average GNSS fail time, calculated over failed fixes only, seconds

8.0 - 8.7	Average GNSS freshen time per successful fix, seconds
9.0 - 9.6	Wakeups per trip
9.7 - 10.7	Uptime, LSb = 1 week

Note that when an Oyster 3 is fitted with Lithium Thionyl Chloride batteries, Uplink 3 is not capable of representing the full battery voltage, and will read its maximum value of 5.5 V. This is sufficient to identify the batteries as LTC, since 1.5 V lithium batteries will read 5.4 V or less. The Oyster 3 also sends an updated statistics uplink (Uplink 31) which includes accurate voltages. If your integration supports the new uplink, it is advisable to disable Uplink 3 to save battery.

Example: 8BF3DC7B9438984278B85E

```
↓10.7                                                                                  0.0↓
      5E       B8       78       42       98       38       94       7B       DC       F3       8B
0101111010111000011110000100001010011000001110001001010001111011110111001111001110001011
0101111010111000011110000100001010011000001110001001010001111011110111001111001110001011
    189     56     120      133      96      226      327        6073         1848      11
```

- 11 x 0.1 + 4.0 = 5.1 V when battery was inserted
- 1848 x 32 = 59136 transmissions
- 6073 x 32 = 194336 trips
- 327 x 32 = 10464 GNSS fixes
- 226 x 32 = 7232 GNSS failures
- 96 s spent per successful GNSS fix, not including freshen time
- 133 s wasted per failed GNSS fix attempt
- 120 s spent freshening GNSS ephemerides per successful GNSS fix
- 56 wakeups per trip
- 189 weeks uptime

3.2.4. Uplink Port 4: Extended GNSS Data Record

Offset	Description
0.0 - 2.7	24 bit latitude, **signed**, LSb = 0.000’025’6°. To convert, first calculate the signed integer value, then multiply by 256E-7 to get a floating-point value.
3.0 - 5.7	24 bit longitude, **signed**, see above
6.0 - 6.2	Heading, LSb = 45°, rounded to nearest
6.3 - 6.7	Speed, LSb = 5 km/h, rounded down

7	Battery voltage, rounded down
	LSb = 25 mV, 0 V offset, if *Battery scale* is 0
	LSb = 32 mV, 3.5 V offset, if *Battery scale* is 1

8.0	0: Out of trip, 1: In-trip
8.1	Last fix failed
8.2	Inactivity indicator alarm
8.3	Battery scale

8.4 - 8.5	Battery critical (brownouts detected), 0: null (unknown), 1: ok, 2-3: critical
8.6 - 8.7	Reserved
9	Reserved, optional
10	Reserved, optional

This is the extended status message, sent if the *Inactivity Indicator* feature is enabled, and the uplink type is left on the default setting of Automatic. When *Inactivity Indicator* is disabled (default), the Oyster 3 will send Uplink 1 instead. The base message is 9 bytes long, but 10 or 11 bytes may be sent in future firmware versions.

Note that fields from *Battery scale* onwards are newly introduced in the Oyster 3 and will be sent as zero in early firmware versions of the Oyster 1.

Example: 53AB783C04A1F98E06

- 53AB78 little endian
  - 78AB53 in hex
  - 7908179 in decimal
  - 7908179 x 256E-7 = 202.4493824°
- 3C04A1 little endian
  - A1043C in hex
  - 10552380 in decimal (unsigned), >= 2^{23}
  - 10552380 - 2^{24} = -6224836 in decimal (signed)
  - -6224836 x 256E-7 = -159.3558016°
- F9
  - 11111001 in binary
  - 001 - 1 x 45 = 45°
  - 11111 - 31 x 5 = 155 km/h
- 8E
  - 142 x 25 = 3550 mV, if battery scale is 0
  - 142 x 32 + 3500 = 8044 mV, if battery scale is 1
- 06
  - 00000110 in binary
  - 0 - not in trip
  - 1 - last fix failed
  - 1 - inactivity indicator alarm
  - 0 - battery scale is 25 mV, 0 V offset
  - 00 - battery critical is null (unknown)

3.3. Oyster 3 Uplinks

3.3.1. Uplink Port 30: Hello

Offset	Description
0	Firmware major version
1	Firmware minor version
2	Product Id (98)
3	Hardware revision
4.0	Power on reset
4.1	Watchdog reset
4.2	External reset
4.3	Software reset
4.4 - 3.7	Reserved
5 – 6	Watchdog reset code
7	Battery voltage, LSb = 32 mV, 3.5 V offset

Please note that when the device resets to apply new parameters after a downlink, the watchdog reset flag will be set and the watchdog reset code will be 1. This indicates a normal reboot under software control.

Example: 010A62010203017A

- 010A
  - Version 1.10
- 6201
  - Product Id 98, hardware revision 1
- 02
  - 00000010 in binary
  - 1 indicates watchdog reset
- 0301
  - 0x0103 in hex
  - Watchdog reset reason 259
- 7A
  - 122 in decimal
  - 3500 + 32 x 122 = 7404 mV

3.3.2. Uplink Port 31: Device Statistics V3

Offset	Description
0	Average GNSS Time To First Fix, LSb = 1 s
1	Average wakeups per trip
2	Initial battery voltage, LSb = 32 mV, 3.5 V offset, rounded down
3	Current battery voltage, LSb = 32 mV, 3.5 V offset, rounded down

4.0	Battery critical (brownouts detected)
4.1	Battery low
4.2 - 5.7	Trip count, LSb = 32 trips
6.0 - 7.1	Uptime, in weeks
7.2 - 8.3	Energy used, LSb = 10 mWh
8.4 - 9.0	Percentage used on LoRaWAN, LSb = 3.125%
9.1 - 9.5	Percentage used on successful GNSS, LSb = 3.125%
9.6 - 10.2	Percentage used on unsuccessful GNSS, LSb = 3.125%
10.3 - 10.7	Percentage used on sleeping and battery self-discharge, LSb = 3.125%

The *Battery low* indicator is currently based on voltage for alkaline and $LiFeS_2$ batteries and based on estimated capacity usage for LTC batteries. It will also be forced high if the *Battery critical* indictor is high. The *Battery critical* indicator goes high and stays high when a brownout is detected. The Oyster 3 will sleep for 3 hours after every brownout, so there will usually be a cycle of brownouts and activity (following the daily temperature variations) before the batteries are completely dead.

The *Percentage used* values are the percentage of the *Energy used* value attributed to each activity. The sum of the values will usually be less than 100%. The *Percentage used on wakeups* is calculated as 100 minus the sum of the transmitted percentages.

Example: 8BF3DC7B9438984278084 3

```
↓10.7                                                                                0.0↓
      43      08      78      42      98      38      94      7B      DC      F3      8B
0100001100001000011110000100001010011000001110001001010001111011110111001111001110001011
0100001100001000011110000100001010011000001110001001010001111011110111001111001110001011
  8    12    4    7      528       664        3621      00  123     220     243     139
```

- 139 second Time To First Fix
- 243 wakeups per trip
- 3500 + 32 x 220 = 10540 mV initial battery Voltage
- 3500 + 32 x 123 = 7436 mV current battery Voltage
- 0 - battery not critical
- 0 - battery not low
- 32 x 3621 = 115872 trips
- 664 weeks of uptime
- 10 x 528 = 5280 mWh energy used
- 3.125 x 7 = 21.875 % of energy used on LoRaWAN
- 3.125 x 4 = 12.5 % of energy used on successful GNSS
- 3.125 x 12 = 37.5 % of energy used on unsuccessful GNSS
- 3.125 x 8 = 25 % of energy used on sleeping and battery self-discharge
- 100 - 21.875 - 12.5 - 37.5 - 25 = 3.125 % of energy used on device wakeups

3.3.3. Uplink Port 33: Position w/ Inactivity Timer

Offset	Description

0.0	Last fix failed
0.1 - 2.7	23 bit latitude, **signed**, LSb = 180° / 2^{23}. To convert, first calculate the signed integer value, then multiply by 180 / 2^{23} to get a floating-point value.
3.0 - 5.7	24 bit longitude, **signed**, LSb = 360° / 2^{24}. Convert as above.
6.0	0: Out of trip, 1: In-trip
6.1 - 6.7	Timestamp, round(UnixTime / 15) modulo 127, or 127 for invalid
7.0	Battery critical (brownouts detected), 0: battery ok, 1: battery critical
7.1	Inactivity indicator alarm
7.2 - 8.7	Inactivity timer, lower bound on time since last movement, LSb = 2 min
9	Battery voltage, LSb = 32 mV, 3.5 V offset, rounded down
10.0 - 10.2	Heading, LSb = 45°, rounded to nearest
10.3 - 10.7	Speed, LSb = 5 km/h, rounded down

This message is newly introduced in the Oyster 3. Compared to Uplink 4 from the Oyster 1, it adds a *Timestamp*, and an *Inactivity timer* field. The scaling of the GNSS coordinates has been changed to improve accuracy, and the battery voltage is always sent with high voltage support.

The *Timestamp* field allows you to accurately determine the time when the uplink was sent, ±8 seconds, if you have a coarse estimate accurate to within ±15 minutes. Some networks don't have accurate timestamps on their gateways and suffer from unpredictable network lag between the gateway and network server. This produces errors in the network timestamp that approach the size of the TCP disconnection timeout. The *Timestamp* field allows you to reduce the timestamp error to less than 8 seconds (plus any inaccuracy on the Oyster 3's clock). See the example Javascript decoder for a decoding algorithm.

The *Inactivity timer* field gives a lower bound on the time since the last accelerometer movement was detected. Its reported precision is 2 minutes. Due to rounding and polling latency, it can under report by up to 3 minutes. The timer can be used to group GNSS coordinates into clusters that were measured at the same location. They also provide a well-defined measurement of asset stop time, that isn't affected by packet loss.

The *Inactivity indicator alarm* is an indicator that goes high when there has been no detected movement for a configured time. The threshold can be set to a larger range of values than the *Inactivity timer* can represent - up to 2 months on the alarm, versus 3 weeks on the timer. The Oyster 3 can also be configured to uplink when the alarm changes, and to heartbeat more frequently while the alarm is set.

Example: E825F49B9E872B6A992AAB

- E825F4
  - 0xF425E8 in hex, 16000488 in unsigned decimal
  - Bottom bit is clear, so last fix was successful
  - Top 23 bits = 16000488 / 2 = 8000244
  - 8000244 is >= 2^{22}, so it becomes 8000244 - 2^{23} = -388364 in signed decimal
  - Latitude = 180° / 2^{23} x -388364 = -8.3333874°
- 9B9E87

  - 0x879E9B in hex, 8887963 in unsigned decimal
  - 8887963 is >= 2^{23}, so it becomes 8887963 - 2^{24} = -7889253
  - Latitude = 360° / 2^{24} x -7889253 = -169.2850041°
- 2B
  - 43 in decimal
  - Bottom bit is set, so in-trip
  - Timestamp = 43 / 2 = 21
  - The equation for the transmit time is:
    - round(TxUnixTime / 15) modulo 127 = 21
  - If reception time is 04 Mar 2022 14:03:00 GMT, then
    - RxUnixTime = 1646402580
  - The closest solution for TxUnixTime is 1646402280
    - 04 Mar 2022 13:58:00 GMT
- 6A99
  - 0x996A in hex, 1001100101101010 in binary
  - Bottom bit is clear, so battery is not critical
  - Next bit is set, so inactivity indicator alarm is active
  - Top 14 bits = 9818 decimal
  - So inactivity timer = 2 x 9818 = 19636 minutes
  - Or 13 days, 15 hours and 16 minutes
- 2A
  - 42 in decimal
  - Battery voltage = 3500 + 32 x 42 = 4844 mV
- AB
  - 10101011 in binary
  - Bottom 3 bits = 011b = 3
    - Heading = 45 x 3 = 135°
  - Top 5 bits = 10101b = 21
  - Speed = 5 x 21 = 105 km/h

3.3.4. Example JavaScript Decoder

```
function MakeBitParser(bytes, offset, length) {
  return {
    bits: bytes.slice(offset, offset + length),
    offset: 0,
    bitLength: length * 8,
    U32LE: function U32LE(bits) {
      if (bits > 32)
        throw ("Invalid argument!");
      if (this.offset + bits > this.bitLength)
        throw ("Read past end of data!");

      var out = 0;
      var total = 0;
      while (bits > 0) {
        var byteNum = Math.floor(this.offset / 8);
        var discardLSbs = this.offset & 7;
        var avail = Math.min(8 - discardLSbs, bits);
        var extracted = (this.bits[byteNum] >>> discardLSbs);
        var masked = (extracted << (32 - avail)) >>> (32 - avail);

        out |= ((masked << total) >>> 0);
        total += avail;
        bits -= avail;
        this.offset += avail;
      }

      return out;
    },
    S32LE: function S32LE(bits) {
      return (this.U32LE(bits) << (32 - bits)) >> (32 - bits);
    }
  };
}

function ResolveTime(timestamp15, approxReceptionTime) {
  if (timestamp15 === 127)
    return null;

  var approxUnixTime = Math.round(approxReceptionTime.getTime() / 1000);

  // Device supplies: round(unix time / 15) modulo 127.
  // We're assuming that the uplink was sent some time BEFORE refTime,
  // and got delayed by network lag. We'll resolve the timestamp
  // in the window [approxReceptionTime - 21m, approxReceptionTime + 10m],
  // to allow for 10m of error in approxReceptionTime, and 10m of network lag.
  // So refTime = approxReceptionTime + 10m.

  var refTime = approxUnixTime + 600;
  timestamp = timestamp15 * 15;

  //                          refTime
  //                             v
  // [              |              |              |              ]
  //           ^              ^              ^              ^
  //       timestamp      timestamp      timestamp      timestamp

  //                             refTime
  //                                v
  // [              |              |              |              ]
  //           ^              ^              ^              ^
  //       timestamp      timestamp      timestamp      timestamp

  // We want the timestamp option immediately to the left of refTime.
  var refTimeMultiple = Math.floor(refTime / (127 * 15));
  var refTimeModulo = refTime % (127 * 15);
  var closestUnixTime = 0;

  if (refTimeModulo > timestamp)
    closestUnixTime = refTimeMultiple * (127 * 15) + timestamp;
  else
    closestUnixTime = (refTimeMultiple - 1) * (127 * 15) + timestamp;

  return new Date(closestUnixTime * 1000).toISOString();
}

function decodeUplink(input) {
  var p = input.fPort;
  var b = MakeBitParser(input.bytes, 0, input.bytes.length);
  var d = {};
  var w = [];

  if (p === 1) {
    d._type = "position";
    var l = {};
    l.latitudeDeg = Number((b.S32LE(32) / 1e7).toFixed(7)); // decimal scaling
    l.longitudeDeg = Number((b.S32LE(32) / 1e7).toFixed(7));
    d.inTrip = (b.U32LE(1) !== 0);
    d.fixFailed = (b.U32LE(1) !== 0);
    l.headingDeg = Number((b.U32LE(6) * 5.625).toFixed(2));
    l.speedKmph = b.U32LE(8);
    d.batV = Number((b.U32LE(8) * 0.025).toFixed(3));
    d.inactivityAlarm = null;
    d.batCritical = null;

    if (d.fixFailed) {
      d.cached = l;
      //w.push("fix failed");
    } else {
      d = Object.assign(d, l);
    }
  } else if (p === 2) {
    d._type = "downlink ack";
    d.sequence = b.U32LE(7);
    d.accepted = (b.U32LE(1) !== 0);
    d.fwMaj = b.U32LE(8);
    d.fwMin = b.U32LE(8);
    if (input.bytes.length < 6) {
      d.prodId = null;
      d.hwRev = null;
      d.port = null;
    } else {
      d.prodId = b.U32LE(8);
      d.hwRev = b.U32LE(8);
      d.port = b.U32LE(8);
    }
  } else if (p === 3) {
    d._type = "stats";
    d.initialBatV = Number((4.0 + 0.1 * b.U32LE(4)).toFixed(2));
    d.txCount = 32 * b.U32LE(11);
    d.tripCount = 32 * b.U32LE(13);
    d.gnssSuccesses = 32 * b.U32LE(10);
    d.gnssFails = 32 * b.U32LE(8);
    d.aveGnssFixS = b.U32LE(9);
    d.aveGnssFailS = b.U32LE(9);
    d.aveGnssFreshenS = b.U32LE(8);
    d.wakeupsPerTrip = b.U32LE(7);
    d.uptimeWeeks = b.U32LE(9);
  } else if (p === 4) {
    d._type = "position";
    var l = {};
    // decimal scaling, truncated integer
    l.latitudeDeg = Number((256 * b.S32LE(24) / 1e7).toFixed(7));
    l.longitudeDeg = Number((256 * b.S32LE(24) / 1e7).toFixed(7));
    l.headingDeg = 45 * b.U32LE(3);
    l.speedKmph = 5 * b.U32LE(5);
    d.batV = b.U32LE(8);
    d.inTrip = (b.U32LE(1) !== 0);
    d.fixFailed = (b.U32LE(1) !== 0);
    d.inactivityAlarm = (b.U32LE(1) !== 0);
    if (b.U32LE(1) === 0)
      d.batV = Number((0.025 * d.batV).toFixed(3));
    else
      d.batV = Number((3.5 + 0.032 * d.batV).toFixed(3));
    crit = b.U32LE(2);
    if (crit === 0)
      d.batCritical = null;
    else if (crit === 1)
      d.batCritical = false;
    else
      d.batCritical = true;

    if (d.fixFailed) {
      d.cached = l;
      //w.push("fix failed");
    } else {
      d = Object.assign(d, l);
    }
  } else if (p === 30) {
    d._type = "hello";
    d.fwMaj = b.U32LE(8);
    d.fwMin = b.U32LE(8);
    d.prodId = b.U32LE(8);
    d.hwRev = b.U32LE(8);
    d.resetPowerOn = (b.U32LE(1) !== 0);
    d.resetWatchdog = (b.U32LE(1) !== 0);
    d.resetExternal = (b.U32LE(1) !== 0);
    d.resetSoftware = (b.U32LE(1) !== 0);
    b.U32LE(4);
    d.watchdogReason = b.U32LE(16);
    d.initialBatV = Number((3.5 + 0.032 * b.U32LE(8)).toFixed(2));
  } else if (p === 31) {
    d._type = "stats v3";
    d.ttff = b.U32LE(8);
    d.wakeupsPerTrip = b.U32LE(8);
    d.initialBatV = Number((3.5 + 0.032 * b.U32LE(8)).toFixed(3));
```

```
    d.currentBatV = Number((3.5 + 0.032 * b.U32LE(8)).toFixed(3));
    d.batCritical = (b.U32LE(1) !== 0);
    d.batLow = (b.U32LE(1) !== 0);
    d.tripCount = 32 * b.U32LE(14);
    d.uptimeWeeks = b.U32LE(10);
    d.mWhUsed = 10 * b.U32LE(10);
    d.percentLora = 100 / 32 * b.U32LE(5);
    d.percentGnssSucc = 100 / 32 * b.U32LE(5);
    d.percentGnssFail = 100 / 32 * b.U32LE(5);
    d.percentSleepDis = 100 / 32 * b.U32LE(5);
    d.percentOther = 100 - d.percentLora - d.percentGnssSucc -
                       d.percentGnssFail - d.percentSleepDis;
  } else if (p === 33) {
    d._type = "position";
    var l = {};
    d.fixFailed = (b.U32LE(1) !== 0);
    l.latitudeDeg = Number((180 * b.S32LE(23) / (1 << 23)).toFixed(7)); // binary scaling
    l.longitudeDeg = Number((360 * b.S32LE(24) / (1 << 24)).toFixed(7));
    d.inTrip = (b.U32LE(1) !== 0);
    d.timestamp = b.U32LE(7);
    d.time = ResolveTime(d.timestamp, new Date());
    d.batCritical = (b.U32LE(1) !== 0);
    d.inactivityAlarm = (b.U32LE(1) !== 0);
    mins = 2 * b.U32LE(14); // lower bound
    d.inactiveDuration = Math.floor(mins / 1440) + 'd' +
                         Math.floor((mins % 1440) / 60) + 'h' + (mins % 60) + 'm';
    d.batV = Number((3.5 + 0.032 * b.U32LE(8)).toFixed(3));
    l.headingDeg = 45 * b.U32LE(3);
    l.speedKmph = 5 * b.U32LE(5);

    if (d.fixFailed) {
      d.cached = l;
      //w.push("fix failed");
    } else {
      d = Object.assign(d, l);
    }
  } else {
    return {
      warnings: ['unknown FPort'],
    };
  }

  return {
    data: d,
    warnings: w,
  };
}
```

3.4. Oyster 1 Compatible Downlinks

LoRaWAN downlink payloads can be as small as 11 bytes in some regions (for the longest-range transmissions). The packet headers already include a 'port number' from 1 to 223, which we will use as a message type. The Oyster 3 sends an explicit acknowledgement uplink (Uplink 2) on reception of a downlink. It sends the acknowledgement only once, on its next uplink attempt. The uplink includes a sequence number to help identify the specific downlink being acknowledged, despite any queuing / buffering in the network. 'Confirmed' and 'unconfirmed' downlinks are handled in the same way.

3.4.1. Downlink Port 1: Set Trip Parameters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Ignore V1 parameters on Oyster 3, default 0 (apply closest match)

1	Max time between fix attempts out of trip (‘heartbeat’), 1-127: 1-127 mins, 129-255: 1-127 hours, default 24 hours
2	Trip end timeout, LSb = 10s without accelerometer detecting movement, 0 disables trip tracking entirely (heartbeats only), default 5 mins
3	Time between fix attempts in-trip during work hours, 1-127: 1-127 seconds, 129-255: 1-127 minutes, 0 or 128 disables, default 10 mins
4	Time between fix attempts in-trip after hours, 1-127: 1-127 seconds, 129-255: 1-127 minutes, 0 or 128 disables, default 10 mins
5.0	Fix on start of trips during work hours, default true
5.1	Fix on end of trips during work hours, default true
5.2	Fix on start of trips after hours, default true
5.3	Fix on end of trips after hours, default true
5.4	Optimise GNSS for trip tracking (download all ephemerides), default true
5.5	V1 Disable stats messages (Uplinks 3 and 31), default 0 (send both)
5.6	Disable wakeup filtering during work hours, 1: only apply accelerometer wakeup threshold and count, 0: apply threshold, count, and filter, default 0
5.7	Disable wakeup filtering after hours, 1: only apply accelerometer wakeup threshold and count, 0: apply threshold, count, and filter, default 0
6	V1 Accelerometer wakeup threshold, 1-8: 63-504 mG, default 126 mG
7	V1 Accelerometer wakeup count, 1-12: 80-960 ms, default 80 ms

Note that the stats messages and accelerometer parameters on the Oyster 3 differ slightly to those on the Oyster 1. The Oyster 3 supports an additional stats uplink (Uplink 31), that you may wish to disable separately from Uplink 3. And the accelerometer on the Oyster 3 has slightly different threshold settings.

These Oyster 3 specific settings can be set using Downlinks 30 and 32 respectively. If you wish to do that, you can set the *Ignore V1 parameters on Oyster 3* bit to one, and send those downlinks separately. Otherwise, if you send a zero in bit 0.7, the Oyster 3 will apply the *V1 Disable stats messages* parameter to both stats uplinks, and will translate the accelerometer settings to their closest equivalents.

3.4.2. Downlink Port 2: Set After-Hours 1

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1	Start of after-hours period on Monday, LSb = 7m30s since 00:00 local time, setting to end time indicates no after-hours period, default 0
2	End of after-hours period on Monday, LSb = 7m30s since 00:00 local time, setting to start time + 1 indicates a 7m30s after-hours period, set start time to 00:00 and end time to 24:00 to indicate the whole day is after-hours, default 0

3	Start of after-hours period on Tuesday, see above
4	End of after-hours period on Tuesday, see above
5	Start of after-hours period on Wednesday, see above
6	End of after-hours period on Wednesday, see above
7	Start of after-hours period on Thursday, see above
8	End of after-hours period on Thursday, see above

3.4.3. Downlink Port 3: Set After-Hours 2

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1	Start of after-hours period on Friday, LSb = 7m30s since 00:00 local time, setting to end time indicates no after-hours period, default 0
2	End of after-hours period on Friday, LSb = 7m30s since 00:00 local time, setting to start time + 1 indicates a 7m30s after-hours period, set start time to 00:00 and end time to 24:00 to indicate the whole day is after-hours, default 0
3	Start of after-hours period on Saturday, see above
4	End of after-hours period on Saturday, see above
5	Start of after-hours period on Sunday, see above
6	End of after-hours period on Sunday, see above

3.4.4. Downlink Port 4: Set Time Zone

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7 - 1.1	Reserved, set to zero
1.2 - 2.1	Offset of normal time zone from UTC, LSb = 15 minutes, **signed**, default 0
2.2 - 2.5	Daylight saving time shift, LSb = 15 minutes, 0 disables, default 0
2.6 - 3.0	Start of DST Nth day of month, 1: First day, …, 5: Fifth day, 6: Last day, 0: Use absolute date instead of relative '2nd Sunday of October' style
3.1 - 3.5	1-7: Monday-Sunday, or 1-31: Day of month (absolute date), local time
3.6 - 4.1	1-12: January-December
4.2 - 5.2	Offset from 00:00 local time, LSb = 15 mins, **signed**, can select hours in previous days like '02:00 on Friday before last Sunday of October local time' would need Offset = (2hr - 48hr) * 4 = -184
5.3 - 5.5	End of DST Nth day of month, 1: First day, …, 5: Fifth day, 6: Last day, 0: Use absolute date instead of relative '2nd Sunday of October' style
5.6 - 6.2	1-7: Monday-Sunday, or 1-31: Day of month (absolute date), local time

6.3 - 6.6	1-12: January-December
6.7 - 7.7	Offset from 00:00 local DST, LSb = 15 mins, **signed**, can select hours in previous days like '02:00 on Friday before last Sunday of April local DST' would need Offset = (2hr - 48hr) * 4 = -184

3.4.5. Downlink Port 5: Set GNSS Parameters 1

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Require 3D GNSS fixes, default false
1	Max time to wait for initial coarse GNSS fix, 45-127: 45-127 seconds, 129-255: 1-127 minutes, default 3 mins
2	Max time to wait for fine GNSS fix, 1-127: 1-127 seconds, 129-255: 1-127 minutes, 0 or 128 disables, default 5 s
3	Target accuracy for the fine GNSS fix wait, LSb = 1m, default 20 m
4	Required PDOP for valid GNSS, 25-100: 2.5 to 10.0, default 10.0
5	Required position accuracy for valid GNSS, 5-100: 5-100 m, default 75 m
6	Required speed accuracy for valid GNSS, 8-55: 2.88-19.8 km/h, default 10 km/h
7	Discard first N GNSS points to allow solution to settle, 0-32: 0-32 positions, default 3

3.4.6. Downlink Port 6: Set GNSS Parameters 2

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1	V1 Daily GNSS on-time budget, 45-127: 45-127 seconds, 129-255: 1-127 minutes, 0 or 128 for unlimited on-time budget, default unlimited, supported on Oyster 1, ignored on Oyster 3
2	Max time to spend looking for 1st satellite, 1-255: 5-1275s, 0 disables signal validator, default 1 min
3	Max time to spend looking for 2nd satellite, 0-255: 0-1275s, default 1 min
4	Max time to spend looking for 3rd satellite, 0-255: 0-1275s, default 1 min
5	Max time to spend looking for 4th satellite, 0-255: 0-1275s, default 1 min
6.0 - 6.4	Satellite detection margin adjustment, **signed**, -16-15: -16-15 dB, >=0 requires stronger signal, <0 allows weaker signal, 0 is default
6.5	V1 Enable Autonomous Aiding, default off, ignored on Oyster 3
6.6 - 6.7	Reserved, set to zero

7	V1 GNSS statistical model, 0: Portable, 2: Stationary,3: Pedestrian, 4: Automotive, 5: Sea, 6: Air 1G, 7: Air 2G, 8: Air 4G, default 4, ignored on Oyster 3
8 - 9	V1 Approximate maximum error from Autonomous Aiding calculations, 5-1000: 5-1000 m, 0: automatic, default: 100, ignored on Oyster 3

Note that the *V1 Daily GNSS on-time budget*, *V1 Enable Autonomous Aiding*, *V1 GNSS statistical model*, and *V1 Approximate maximum error from Autonomous Aiding calculations* parameters are supported by the Oyster 1 and ignored on the Oyster 3. However, all ten bytes must be supplied in the downlink.

3.4.7. Downlink Port 7: Set LoRaWAN Channels

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1.0 - 1.3	Minimum data rate to use when ADR is disabled, default is 0 (DR0)
1.4 - 1.7	Maximum data rate to use when ADR is disabled, default is 2 (DR2)
2 - 10	Uplink channel mask, set bits are enabled channels, LSb of the 1st byte is channel 0, MSb of the 9th byte is channel 71, set all zeros (default) for the region-specific defaults

The Oyster 3 will spread its transmissions out over the allowed data rates in such a way as to equalize the time spent on-air at each data rate. For the default setting of DR0-DR2, this gives a 16 / 30 / 54% split between the three data rates, and maximizes the gateway's capacity. However, the relative range of the three data rates are 100, 75, and 50% respectively. When ADR is enabled, the network server controls the data rate instead.

The uplink channel mask should be left 0 (default) in regions where the network join channels are fixed. In these regions, the gateway will tell the Oyster 3 which channels to use, during the Join procedure.

In regions where the join channels are not specified (US902-928, AU915-928), you should set the channel mask to avoid continued transmission on unused channels. In these regions some networks might not tell the Oyster 3 which channels to use, resulting in significant packet loss if the mask hasn't been programmed.

3.4.8. Downlink Port 8: Set LoRaWAN Join / App EUI

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Set JoinEUI, 0: Use default JoinEUI, 1: Use supplied JoinEUI, default 0
1 - 8	JoinEUI, non-zero, **big endian**, i.e. default JoinEUI 70-B3-D5-70-50-00-00-00 is encoded with first byte as 0x70 and the second byte 0xB3

The acknowledgement will be transmitted **once** on the existing JoinEUI, and then the Oyster 3 will switch to the new JoinEUI. It continues to use the already provisioned NwkKey and AppKey, which cannot be programmed over the air.

Note that:

- In LoRaWAN 1.0, the JoinEUI is known as the AppEUI
- In LoRaWAN 1.1, changing the JoinEUI resets three cryptographic counters:
    - RJCount1
    - DevNonce
    - JoinNonce
- So after changing the JoinEUI in 1.1, you must reset the counters on the join server

3.4.9. Downlink Port 9: Set Advanced LoRaWAN Options

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1	Days between network joins, 0 disables, default 7
2.0 - 2.1	ADR support, 0: Never, 1: When out-of-trip, 2: Always, default 1
2.2 - 2.3	Reserved, set to zero
2.4 - 2.7	Initial frame repetitions (NbTrans/Reps), 1-15, default 1
3.0 - 3.3	Initial MaxCount0, sets uplinks between Rejoin0 attempts in LoRaWAN 1.1 OTAA, uplink interval equals 2^(4+MaxCount0), default 15
3.4 - 3.7	Initial MaxTime0, sets approx. time between Rejoin0 attempts in LoRaWAN 1.1 OTAA, uplink period equals 2^(10+MaxTime0) s, default 15
4.0 - 4.3	Initial AdrAckLimitExp, sets uplinks between ADR confirmation requests, uplink interval equals 2^AdrAckLimitExp, default 6 (limit is 64)
4.4 - 4.7	Initial AdrAckDelayExp, sets uplinks between ADR backoff steps, uplink interval equals 2^AdrAckDelayExp, default 5 (delay is 32)
5	Maximum Tx power limit, **signed**, -128-127: -128 to 127 dBm EIRP, default 127 (no limit)
6	Random Tx delay, 0: disabled, 1-127: 1-127 seconds, 129-255: 1-127 minutes, default disabled

By default, the Oyster 3 will rejoin the network if there have been no downlinks received in the last 7 days. This insures against database failure or accidental device deletion on the network server. The exact rejoin behavior depends on the Activation and LoRaWAN version.

- In ABP, the rejoin is equivalent to removing and reinserting the batteries. All protocol state is reset, with the exception of the uplink and downlink counters (unless you have chosen to reset them as well, in LoRaWAN 1.0)
- In OTAA 1.0, the rejoin is also equivalent to removing and reinserting the batteries. A regular Join Request will be sent.
- In OTAA 1.1, a Rejoin1 Request is used to rejoin without resetting any protocol state. Please note that older network servers might not support this mechanism.

You can lengthen the rejoin period, or disable it, using the *Days between network joins* parameter.

Initial MaxCount0 and *Initial MaxTime0* control a separate rejoin mechanism called Rejoin0, which exists only in LoRaWAN 1.1, to facilitate roaming. Usually, the network will set these values automatically if required, so they can be left at the default maximum values.

Initial frame repetitions sets the default number of frame repetitions (called NbTrans in the standard). Usually this value is one, until the server requests a higher number. You shouldn't change this value unless you have special requirements (for instance, a transmit-only ABP configuration).

ADR is a server-side mechanism for controlling the data rate, to optimize network capacity and battery life. By default, the Oyster only requests ADR when out-of-trip. When in-trip, it controls its data rate according to the configured range. Once the trip ends, it switches to its lowest configured data rate and maximum power and begins requesting ADR from the server. Thereafter, the network server will configure it to a lower power mode if appropriate. You may wish to instead force ADR on at all times, in order to control it using a policy on the network server. It is also possible to force it off at all times, but this can lead to compatibility issues on older network servers.

Initial AdrAckLimitExp and *Initial AdrAckDelayExp* control the number of uplinks between connectivity checks when ADR is active. The device starts asking for a response once the Limit is reached with no downlinks, and if no response is heard after Delay attempts, it switches to a lower data rate. The default values (64 and 32) are very slow. Please consult your network support before choosing more aggressive settings, as they may have policies forbidding lower values. In LoRaWAN 1.1, these values can also be set dynamically by the network server.

Maximum Tx power limit restricts the transmit power. This allows you to save a small amount of battery on the transmission and limit the peak current draw from the batteries. Limiting the current may be useful in extreme cold weather (less than -20°), where lithium batteries begin to struggle. Your network server may require knowledge of the maximum power to correctly operate its ADR algorithm. In this case, be sure to configure the device accordingly on the server.

Random Tx delay allows you to insert a random delay before transmission. This may be useful in applications involving a large number of assets on the same vehicle. The random delay spreads the transmissions out, so they don’t interfere with each other. Note that the GNSS timing already provides some measure of built-in randomness, so this isn't essential if less than 10 units are involved.

The amount of delay required depends on the data rates in use. For a single spreading factor, you need approximately (Number of units) / 1500 * 1.9^{SF}). For instance, for 100 units at SF10, you'd need 100 / 1500 * 1.9^{10}) = 613 / 15 = 41 seconds.

For multiple spreading factors, the formula is $(1/DelayA + 1/DelayB + 1/DelayC + ...)^{-1}$, where DelayX is the delay required for one of the spreading factors. So for 100 units at both SF10 (41 seconds) and SF9 (22 seconds), you'd need $(1/41 + 1/22)^{-1}$ = 14 seconds.

3.4.10. Downlink Port 10: Set Inactivity Parameters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero

1.0 - 1.1	Inactivity Indicator Alarm fix on set, 0: no fix, 1: single fix attempt, 2: retry fix attempts if network not ready (i.e. join failed - uplinks are still unconfirmed), default 0
1.2 - 1.3	Inactivity Indicator Alarm fix on clear, see above
1.4 - 1.7	Reserved, set to zero
2	Inactivity Indicator Alarm timeout, 1-127: 10-1270 min, 129-255: 12-1524 hours, others disable, default 0
3	Inactivity Alarm Heartbeat Period, 1-127: 1-127 mins, 129-255: 1-127 hours, default 0 (disabled)
4	V3 Position Uplink, when in-trip, 0: Auto, others: uplink number
5	V3 Position Uplink, when not in-trip, 0: Auto, others: uplink number
6	GNSS fix every nth heartbeat, >= 1, default 1 (fix every heartbeat)
7.0 - 7.4	Extra end-of-trip heartbeats, default 0
7.5 - 8.7	Time between extra end-of-trip heartbeats, 0-2047: 0-20470 s, default 60 s

Note that parameters from *Inactivity Alarm Heartbeat Period* onwards are newly introduced in the Oyster 3, and will be ignored by older versions of the Oyster 1.

The *Inactivity Alarm Heartbeat Period* is applied when the *Inactivity Indicator Alarm timeout* has expired, if it is shorter than the regular heartbeat period.

The *V3 Position Uplink* settings are applied only to the Oyster 3, even though this downlink is understood by both the Oyster 1 and the Oyster 3. Their default is 0: Auto, which gives the same automatic Uplink 1 or Uplink 4 behaviour as the Oyster 1. To take advantage of the timestamping features on the Oyster 3, we recommend a value of 33, to select Uplink 33. The in-trip and out-of-trip behaviour is independently configurable, since heading and speed are generally not required out-of-trip, and inactivity timers are generally not required in-trip. Future firmware releases may make use of this to optimize available uplink space.

The *GNSS fix every nth heartbeat* parameter allows you to skip the power-hungry GNSS fix for most heartbeats, to save battery while still providing downlink opportunities. Starting a trip resets the counter for fixes, so there will always be a fix in the first uplink after a trip end.

Extra end-of-trip heartbeats allows you to fire off additional end-of-trip position uplinks, with a short delay between them. The extra fixes can be averaged to give higher accuracy, or they can simply be used to guard against packet loss. All of the extra heartbeats will perform a fix - this feature does not interact with the *GNSS fix every nth heartbeat* parameter.

3.4.11. Downlink Port 11: Set Scheduled Upload Parameters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	0: Single fix attempt, 1: Retry fix attempts if network not ready (i.e. join failed - uplinks are still unconfirmed), supported from firmware version 2.0 onwards, unconfigured default is no attempts
1	Local time* of scheduled upload time 1, 1-192: 00:07:30-24:00:00 (7.5 min intervals), +- 5 minutes randomization, default 0 (disabled)

2	Local time* of scheduled upload time 2, see above
3	Local time* of scheduled upload time 3, see above
4	Local time* of scheduled upload time 4, see above
5	Local time* of scheduled upload time 5, see above
6	Local time* of scheduled upload time 6, see above
7	Local time* of scheduled upload time 7, see above
8	Local time* of scheduled upload time 8, see above
9	Local time* of scheduled upload time 9, see above
10	Local time* of scheduled upload time 10, see above

*Local time takes time zone and daylight savings offset into account (configurable in Downlink 4).

The scheduled upload feature allows for heartbeats to be scheduled, rather than setting an upload interval. Each upload will occur within a 5 minute offset (i.e. scheduled upload at 13:00 could occur anywhere between 12:55 and 13:05). Up to 12 upload times may be configured per day. The first 10 upload times are configured with this downlink.

3.4.12. Downlink Port 12: Set Scheduled Upload Parameters (continued)

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1	Local time* of scheduled upload time 11, 1-192: 00:07:30-24:00:00 (7.5 min intervals), +- 5 minutes randomization, default 0 (disabled)
2	Local time* of scheduled upload time 12, see above

*Local time takes time-zone and daylight savings offset into account (configurable in Downlink 4).

The scheduled upload feature allows for heartbeats to be scheduled, rather than setting an upload interval. Each upload will occur within a 5 minute offset (i.e. scheduled upload at 13:00 could occur anywhere between 12:55 and 13:05). Up to 12 upload times may be configured per day. The 11th and 12th upload times are configured with this downlink.

3.5. **Oyster 3 Downlinks**

3.5.1. Downlink Port 30: Set Device Statistics Parameters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Battery type, 0: lithium, 1: alkaline, default 0
1 - 2	Time between statistics uplinks, 0 - 65535 hours, +-12% variance, 0 disables, default 96 (4 days)
3 - 4	Per-cell battery capacity, 1-65535 mWh, 0 disables, default 5000

5	Battery percentage discharge per year, 0-255: 0-25.5 %, default 3 %
6.0	Disable Uplink 3, default 0 (send)
6.1	Disable Uplink 31, default 0 (send)
6.2 - 6.7	Reserved, send as zero

The *Battery type* field allows you to specify alkaline batteries, which are not officially supported. Alkaline batteries don’t have sufficient voltage to deliver their full potential in the Oyster 3 and are notorious for failing early due to low manufacturing standards in the alkaline battery industry. However, they can be useful for demo purposes, or even be economical if you can replace them easily. In real world deployments, we recommend Energizer Ultimate Lithium wherever possible, as the increased reliability is generally worth the additional cost.

Selecting alkaline batteries changes the behaviour of the LoRaWAN DevStatusReq (network level battery reporting) to suite alkaline voltages, and allows alkaline batteries to work down to 3.5 V instead of cutting out at 4.0 V. But it also **disables detection of single lithium cell reverse polarity**, which cannot be disambiguated from normal alkaline battery operation. So please use caution if you select alkaline batteries. If you subsequently use 3.6 V lithium batteries and reverse a single cell, the Oyster will continue to function normally, and the reversed battery may leak or explode.

The *Per-cell battery capacity* parameter gives the total energy, in mWh, expected to be available from a single AA cell. The default value of 5000 is approximately correct for both 1.5 and 3.6 V lithium chemistries. Alkaline batteries can be expected to produce roughly half of this - if they don’t fail prematurely. The capacity is used to calculate the remaining battery life in DevStatusReq messages, to calculate self-discharge, and to trigger the Battery Low flag for 3.6 V lithium cells. For other chemistries, the Battery Low flag is based on voltage.

This downlink allows the statistics uplinks to be enabled and disabled individually, which is not possible using Downlink 1. Both Uplink 3 and Uplink 31 are sent by default, for backwards compatibility with the Oyster 1, but it is recommended to disable Uplink 3 and use Uplink 31 exclusively where compatibility is not required.

3.5.2. Downlink Port 31: Set GNSS Parameters 3

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1.0	Use GPS constellation, default 1 (use)
1.1	Use SBAS constellation, default 1 (use)
1.2	Use QZSS / CA constellation, default 1 (use)
1.3	Use QZSS / S constellation, default 1 (use)
1.4	Use GLONASS constellation, default 1 (use)
1.5	Use Galileo constellation, default 1 (use)

1.6 - 1.7	Reserved, set to zero

This downlink carries GNSS parameters specific to the Sony CXD5605 on the Oyster 3. It currently contains only constellation enable flags, all of which are on by default. There is a small battery life benefit to disabling constellations that aren't available in your area (ie. disabling QZSS in the Western hemisphere).

3.5.3. Downlink Port 32: Set Accelerometer Parameters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Enable higher precision thresholds, and thresholds below 0.078 G
1	Accelerometer wakeup threshold, 2-32: 32-500 mG, default 94 mG
2	Accelerometer wakeup count, 1-100: 10-1000 ms, default 100 ms
3.0 - 3.3	Accelerometer sampling rate (Hz), 0-4: {1, 10, 25, 50, 100}, default 1 (10 Hz)
3.4 - 3.7	Reserved, set to zero

This downlink carries accelerometer parameters specific to the LIS2DH12 accelerometer on the Oyster 3. The parameters in Downlink 1 are tailored to the MMA8451Q on the Oyster 1 and are mapped to the Oyster 3 for compatibility. You can use this downlink instead if you require tighter control.

Increasing the accelerometer sampling rate will increase its sensitivity to brief shocks but will increase the sleep current by about 0.08 µA/Hz using 1.5 V batteries.

The *Enable higher precision thresholds* parameter increases the accelerometer's power consumption, but lowers its noise floor, allowing thresholds below 0.078 G to be used. At 10 Hz, it increases the sleep current by about 0.9 µA using 1.5 V batteries.

3.5.4. Downlink Port 33: Reset ABP Session Counters

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reset FCnts every power cycle when using ABP on LoRaWAN 1.0, default 0 (don't reset)

This parameter controls a work-around for an issue encountered in LoRaWAN 1.0, when using Activation By Personalization. For security reasons, LoRaWAN end nodes are required to keep a monotonically increasing frame counter (FCnt) across reboots and power cycles. However, LoRaWAN 1.0 contains no mechanism for communicating that an end node has been reset. This makes it impossible for end nodes with limited memory to function in ABP mode, as they would be obligated to save the entire protocol state every time they received an update from the network.

As a work-around, some networks support signalling a device reset by resetting the frame counters. This defeats security, but allows ABP devices to operate even when there is no downlink connectivity, without requiring extra memory and complex manual reset procedures. On many networks, ABP is essentially unusable on the Oyster without this setting being active. But using this setting without explicit network support will cause uplinks to go missing when the device resets.

ABP on LoRaWAN 1.0 suffers from this state resetting issue, and ABP on LoRaWAN 1.1 requires a handshake at startup, just like OTAA. So unless you have a very firm technical grasp of the consequences, **we recommend not using ABP whatsoever**. OTAA is simpler and more flexible.

3.5.5. Downlink Port 34: Reset Statistics Command

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Reserved, set to zero
1.0	Reset all statistics in Uplink 3 and Uplink 31
1.1	Reset just the Battery Critical indicator
1.2 - 1.7	Reserved, set to zero

The Oyster 3 keeps battery use statistics from the moment the batteries are inserted, up until the next full Power-On-Reset cycle. When you swap new batteries into the device, it must be totally unpowered for several seconds (tens of seconds for higher battery voltages) before it realizes a battery change has taken place. If you've waited long enough, the LED will start flashing, and continue to flash throughout the first uplink cycle.

If you haven't waited long enough, or haven't inserted the batteries cleanly, the LED will either remain unlit or stay lit for ten seconds to indicate a brownout. If the installer doesn't realize this, the device may go out into the field without properly resetting the battery statistics, or with a false battery critical indication. This downlink allows you to correct such a mistake after deployment.

3.5.6. Downlink Port 35: Poll Command

Offset	Description
0.0 - 0.6	Downlink sequence number (reported in acknowledgement)
0.7	Send Downlink Ack (Uplink 2)
1.0	Send Device Statistics (Uplink 3)
1.1	Send Device Statistics V3 (Uplink 31)
1.2	Apply *Poll Position Count* and *Poll Position Period*
1.3 - 1.7	Reserved, set to zero
2 - 3	Poll Position Count, 0: cancel, 1-65535: send 1-65535 positions
4 - 5	Poll Position Period, 1-65535: 10 to 655350 seconds between uplinks

This downlink allows you to solicit statistics messages and GNSS positions. The statistics messages are by default sent only once every four days and could be missed if the Oyster is frequently out of coverage. By queuing a downlink requesting stats, you can ensure that they will be transmitted when the Oyster moves into coverage.

The GNSS position poll can be used together with the *GNSS fix every nth heartbeat* parameter in Downlink 10 to perform the power hungry GNSS fixes on demand, rather than on every heartbeat.

It can also be used to place the device into a temporary recovery mode, for instance by ordering a heartbeat every 5 minutes for the next 24 hours after a theft has been discovered. The advantage of using the position poll for this purpose rather than simply changing the tracking parameters is that there is no risk of draining the battery if the device can’t be reached to revert the parameters to normal.

The *Send Downlink Ack* option controls whether the normal Uplink 2 acknowledgement is sent before the polled uplinks are actioned. If you’re sending the Poll command from an automated system (for instance, a closed loop downlink queue that repeats downlinks until a device is configured) you may wish to receive the acknowledgement to close the loop. But if you’re polling manually on your network server’s web console, it is enough to receive the polled statistics or position directly.

4. CONTACT INFORMATION

For the latest version of this document plus other product information please visit our website at www.digitalmatter.com/support, or contact DM at info@digitalmatter.com.