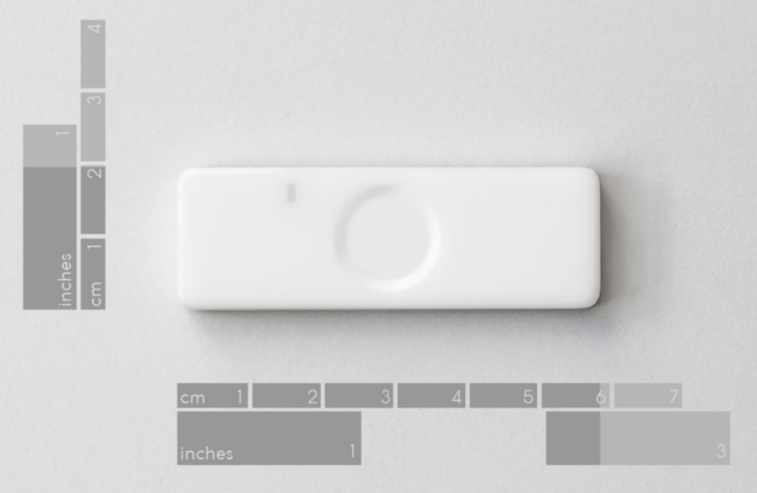
TouchTAG Product Brief

FEATURE SET

- Very low cost and disposable
- LoRaWAN™ compliant
- Up to 1-year battery life*
- LoRa™ geolocation
- Motion and fall detection
- Temperature sensor
- Compact and lightweight
- White Label and reference design options



DESCRIPTION

The TouchTAG product is a disposable action driven IoT device. It is a very low cost, compact and lightweight geotag with additional sensor functionality. It is fully compatiable with the LoRaWAN™ LPWA protocol stack and is fully programmable,

With an integrated temperature sensor and an accelerometer TouchTAG can be deployed in many IoT verticals. The very small size, low cost and weight allow for it to be used for locaating different shipments from containers and pallets to boxes and letters. The temperature sensor and accelerometer allow condition of carriage for perishable or fragile cargos to also be monitored. Programmable instant alerts allow notifications when the sensors have been triggered. TouchTAG can also be used in many other IoT application verticals.

TECHNICAL SPECIFICATIONS

MECHANICAL:

DIMENSIONS: **19.4 x 58 x 6.6 mm**
WEIGHT: **5 grams**

FUNCTIONAL:

BUIT-IN SENSORS: **temperature, orientation, button, fall detector, movement detector**

TEMPERATURE RANGE: **-20°C to 60° C**
ACCELEROMETER RANGE: **up to 16g**
LOCATION: **LoRa™ Geolocation**

WAKE-UP ON: **button, fall detector, movement detector, periodic timer**

COMMUNICATIONS: **LoRaWAN™ 868/915**
BATTERY: **3,6V 150mAh**
DEVICE UI: **red and green LEDs, button**

BATTERY LIFE PROFILE
(With all triggers on)

Reporting interval	Lifetime
1 minute	4 days
5 minute	20 days
15 minute	50 days
30 minute	95 days
1 hour	160 days
2 hours	250 days
4 hours	340 days
8 hours	420 days
12 hours	450 days

USE CASE EXAMPLES

Below are some indicative examples of applications and verticals where TouchTAG can be used to add tangible value. For more details of these examples and to find out how TouchTAG is being used today or how it may be used to help solve your IoT challenge please contact sales@everynet.com or your local Everynet Representative.

DEMAND BUTTON

Like the Amazon DASH button, TouchTAG can be used as a 'demand button', this can be used for automating non-periodic demands in a multitude of scenarios: it can be pressed when the printer needs paper or ink, when a meeting room user has finished early, when a production line needs stock unexpectedly, when staff are entering or leaving a restricted area, when a 'clean up' or IT team is needed, when you need a certain piece of machinery or tool that is a shared asset. All of the simultaneous multiple use case demands can be centralized, prioritized and actioned within your facility: office, factory, port, rail yard fulfilment center, hospital or airport. The applications of TouchTAG as a 'demand button' is only limited by your imagination!

RETAIL FRIDGE MONITORING

Retail Fridge Monitoring is an area where LoRaWAN networks can easily demonstrate its value, TouchTAG can relay the temperate of the fridge on a periodic basis to ensure that optimum temperature is always maintained. The historic data can show adherence to Health and Safety regulations. In the case of a 'deviation in temperature' event, TouchTAG can send an ALERT message which can be translated to a text message or email notification to the maintenance engineer minimizing the delay to check and rectify the problem.

CARGO TEMPERATURE MONITORING

When transporting cargo in by air, road or sea TouchTAG can be used to give indication of the storage temperature of the respective boxes, pallets or containers at various stages of the transit, enabling provenance on storage conditions for goods that require temperature consistency Indication at which stage of the transit the storage temperature was compromised can be easily monitored or ALERT notifications can be sent. A history of the temperature of the goods at intervals during the transit provides assurance and provenance of the transportation conditions.

ORIENTATION MONITORING OF FRAGILE GOODS

During the transportation of fragile goods having access to information that indicates how the goods have been handled during transit offers security and provenance that appropriate care has been taken. TouchTAG's orientation feature allows periodic monitoring of the boxes pallets or containers position, if this is compromised by mishandling, accidental dropping or impact damage an ALERT notification will register the event allowing for an analysis of the transit history ensuring that quality standards are met.

ORIENTATION MONITORING OF INFRASTRUCTURE: LIGHT POLES, ROAD SIGNS

Orientation of permanent infrastructure installations such as lampposts, traffic signs and telegraph poles is made simple with TouchTAG and allows simple remote monitoring of these fixed assets. The low power and low cost allows for large scale deployments that simply send an ALERT when a change in the orientation of the fixed infrastructure asset occurs. This change may be caused by accidental damage, land subsidence or just generic movement overtime. Periodic monitoring of the assets will provide information that can indicate potential risks, by monitoring any very small changes in orientation ensuring that the maintenance of the assets can be preventative not reactionary.

CUSTOMER SERVICE – FACILITIES MANAGEMENT

Offer your customers or staff an ability to let you know when something is needed in your washrooms or public areas, attach TouchTAG to your paper towel holders, soap dispensers, waste bins and seating areas to provide a simple low cost way to receive feedback of the status of these amenities. TouchTAG can simply be attached to each item and your customers or staff simply press the button when the soap dispenser is empty, the waste bin full or the seating area dirty. Customer and staff engagement can be utilized in ensuring a better service and removes the requirement and expense to upgrade the respective dispensers with integrated technology. A visualization of your public areas can give an overview of status showing each time a button has been pressed, this ALERT can simultaneous be sent as a text message or email to the appropriate person responsible for ensuring that the requirement is actioned. The improved efficiency of the entire process can be used as a Customer Service marker and drive engagement.

HGH VALUE ASSET LOCATION AND CONDITION MONITORING

Asset management or stock control of high value items is made simple with TouchTAG, each asset is equipped with a TouchTAG device that utilizes the LoRaWAN geolocation feature. This feature ensures that the respective assets can be located in a specific location. With assets that can be at one of several locations or that are continually moved between locations stock control is as simple as looking at a screen. The assets can be associated with a driver or a specific job and removes the need for an annual stock take across the multiple locations. Some examples are cable drums, portable toilets, temporary traffic signals or road signs, railway carriages or specialized machinery. Condition monitoring as detailed in the use case above can also be utilized in this use case example.

FUNCTIONAL DESCRIPTION

TouchTAG is a disposable action driven IoT device: it reacts to external or internal events that activates triggers inside device. Currently the following triggers are supported: periodic timer, button, movement detector, free-fall detector, device has a user interface with a button and two LEDs. When a trigger is activated the device wakes up and flashes the LED and transmits data to the server about the trigger event and the time that the event occurred.

Device is completely switched off during transportation and storage and is activated by the button.

Device operates in the following manner:

1. After first power up the device goes to sleep. All triggers except the button are completely deactivated. No trigger can wake device up, except a button press.
2. The button is pressed 3 times within 2 seconds to activate the device and to enable all other triggers.
3. The activated device wakes up via any of the following triggers:

Trigger	Indication
Timer	no indication
Button pressed	🟩
Movement start	🟩🟩🟩🟩🟩
Movement stop	🟩🟩🟩🟩🟩
Free-fall	🟥🟥🟥🟥🟥
Activation	no indication
Deactivation	🟥🟥

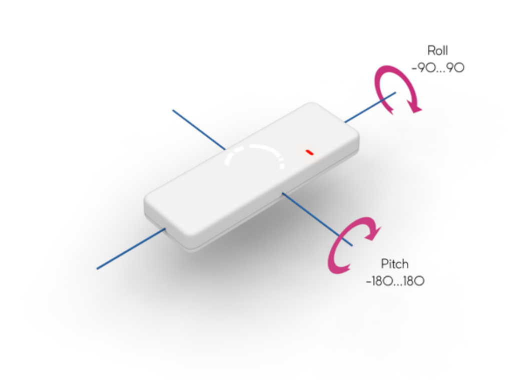
Each time the device is triggered it transmits the following data: time, temperature, orientation and trigger type

4. To deactivate device press and hold the button for 5 seconds. Current consumption dramatically decreases in this mode. When device in not being used it should be set in this mode to ensure maximum battery life.

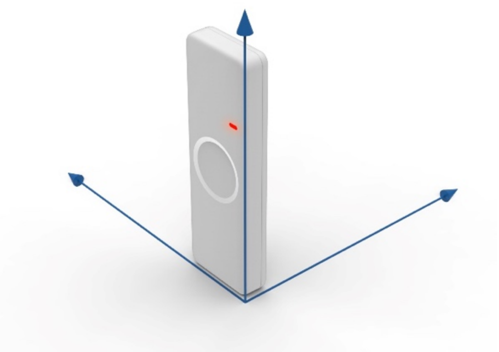
ADVANCED TECHNICAL INFORMATION

Since device has an intergated accelerometer it sends its spatial orientation. Orientation is determined by the Euler's angles (https://en.wikipedia.org/wiki/Euler_angles):

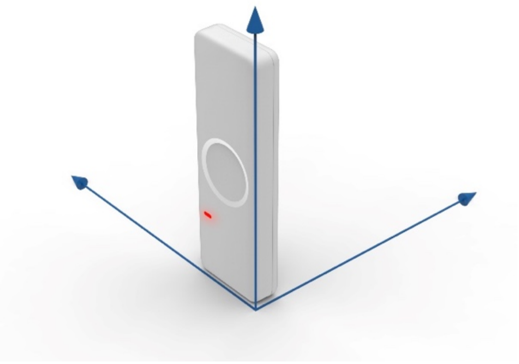
Euler angles



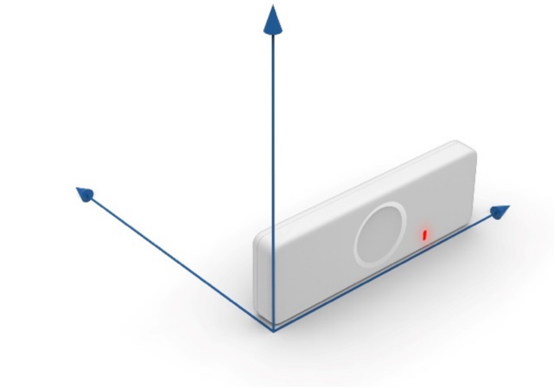
Pitch = 90; Roll = 0



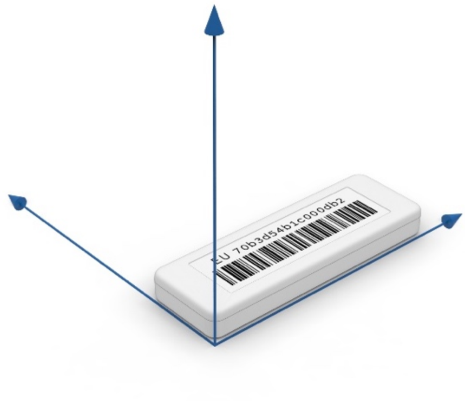
Pitch = -90; Roll = 0;



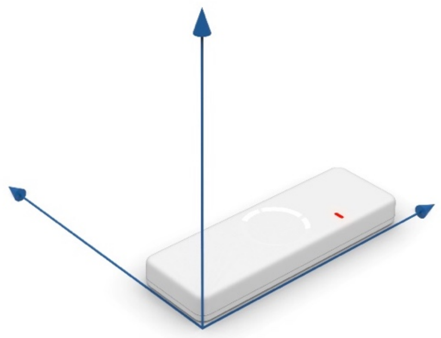
Pitch = 0; Roll = 90;



Pitch = 180; Roll = 0;



Pitch = 0; Roll = 0



DEVICE CONFIGURATION

TouchTAG is provisioned through ABP and has immutable keys flashed into device.

Messages from device could be received using device-specific API provided by Everynet. Device settings can be updathe through API as well.

API documentation is available here: http://posttracker.app.eu-west-1.everynet.io/api/1.0/

RESTful API endpoints could be found using on this URL:

http://posttracker.app.eu-west-1.everynet.io/api/1.0/

All requests must be authorized using the Authorization header with the Bearer authorization scheme and api_key token provided by Everynet. Please, contact Everynet support support@eveynet.com to receive.

API allow user to:

- Get battery level;
- Get device orientation;
- Get device temperature;
- Get message radio parameters and receive time;
- Get gateway info, that receive message;
- Set device timer update period;
- Set trasmittion speed;
- Check device configuration history;
- Add/remove device from application;

All messages stores in database ,so, user can retrieve last message either messages history.

Full API reference can be found here: http://posttracker.app.eu-west-1.everynet.io/api/1.0/

Examples are presented as 'curl' commands. For more information visit official page.

Example 1. How to get last message.

For this purposes special method was implemented: GET /messages/last

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization key}' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/70b3d54b1c000de4/messages/last'
```

Place your authorization key instead of curly braces and you will receive full last message for device '70b3d54b1c000de4', that contains packet radio parameters, temperature, orientation etc. Message fields description can be found in API reference.

Example 2. How to get massage for last 24 hours.

Imagine now is 30 May 2017 12:00:00, and you want to get all messages for last day. So you need add *select* parameter to 'GET /messages' request:

Select=rx_time> 1496059200 – timestamp of the 29 May 2017 12:00:00

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization key}' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/70b3d54b1c000de4/messages?select=rx_time>1496059200'
```

Example 3. How to get messages with temperature highet than 20 °C.

As *select* parameter you can choose almost any field from message (except radio and debug). Here is filtering by temperature:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-
1.everynet.io/api/1.0/devices/70b3d54b1c000de4/messages?select=temperature>20'
```

Example 4. Combining filters.

Also you can combine any number of available filters you want. Ex.2 and ex.3 can be done by one request:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-
1.everynet.io/api/1.0/devices/70b3d54b1c000de4/messages?select=temperature>20&select=rx_time
>1496059200'
```

There are another parameters, that also can be applied to request – limit and order.

limit – allow user to limit how many messages will be displayed. If nothing is specified, default value will be 20.

order – applying this parameter to request will cause sorting messages by given field name (except radio and debug), e.g. order=temperature.

Note. Some browsers represent '>' symbol as '%3E', so 'select=rx_time>1496059200' can be shown as 'select=rx_time%3E1496059200'. This format is also acceptible.

Example 5. Set device configuration.

This method should always contain two mandatory parameters – *update_period* and *datarate*.

update_period – timer event timeout in seconds. Has value in range [0, 86400].

datarate – transmittion speed, has value in range [0, 6].

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header
'Authorization: Bearer {your authorization key}' -d '{ \
   "update_period": 300, \
   "datarate": 2 \
 }' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/70b3d54b1c000de4/config'
```

This command will increase 'config_version' number by 1 and create new database entry.

Example 6. Get device configuration.

As messages interface, user can get last config or history. Filters and parameters has the same mechanism.

This request will return last created device configuration.

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-
1.everynet.io/api/1.0/devices/70b3d54b1c000de4/config/current'
```

This requesrt returns all configurations created after 24 May 2017 13:25:41 and sort list by 'proposed_at' field (that mean, when configuration was sent to the device).

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-
1.everynet.io/api/1.0/devices/70b3d54b1c000de4/config?order=proposed_at&select=created_at>2
017-05-24T13:25:41'
```

Description of the config entries fields can be found on the API reference page.

Example 7. How to manage devices.

Via API user can add/remove devices and retreive devices list. Get request has same selecting/filtering mechanism as ‘get message history’ and ‘get config history’ methods (select, order, limit).

This request will return list of 20 last added devices.

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/'
```

This request will *add* device with DevEUI ‘70b3d54b1c000de4’ to the application.

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header
'Authorization: Bearer {your authorization key}' -d '{ \
   "dev_eui": "70b3d54b1c000de4" \
 }' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/'
```

This request will *delete* device with DevEUI ‘70b3d54b1c000de4’ from the application.

```
curl -X DELETE --header 'Accept: application/json' --header 'Authorization: Bearer {your authorization
key}' 'http://posttracker.app.eu-west-1.everynet.io/api/1.0/devices/70b3d54b1c000de4'
```