

CONNECTING A SENQUIP DEVICE TO UBIDOTS

1. Introduction

Ubidots is an Internet of Things (IoT) application builder with industry leading data analytics and visualisation. This application notes describes how to connect a Senquip device to the Ubidots platform using a Ubidots plugin and without a plugin by using the Ubidots v1.6 API.

You will require:

- An active Ubidots account.
- An active Senquip Portal account.
- A Senquip device.

Your Senquip device should be updated to be running the latest firmware.

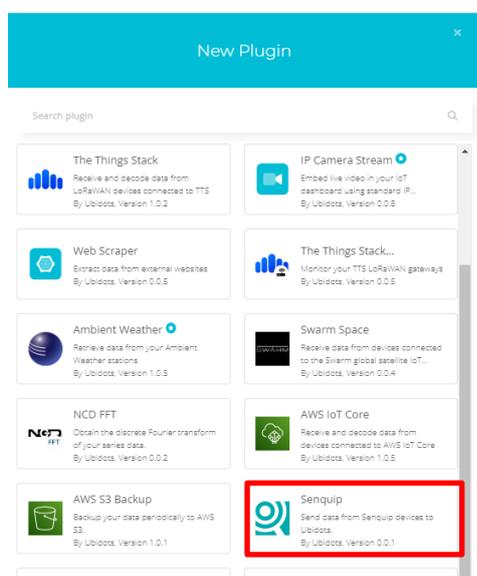
2. Using a Ubidots Plugin

The Ubidots Senquip Plugin is an integration that allows you to send data from your Senquip device to Ubidots with ease. The plugin will carry out all steps needed to set up devices and variables at Ubidots to simplify the integration process.

2.1. Create a Plugin

To configure a Senquip compatible Ubidots plugin:

1. Log in to your Ubidots account, click on the *Devices* tab, then *Plugins*, then click on the + icon or press *Create Plugin* to create a new data plugin.
2. Select the Senquip plugin.



Document Number
APN0026

Revision
1.0

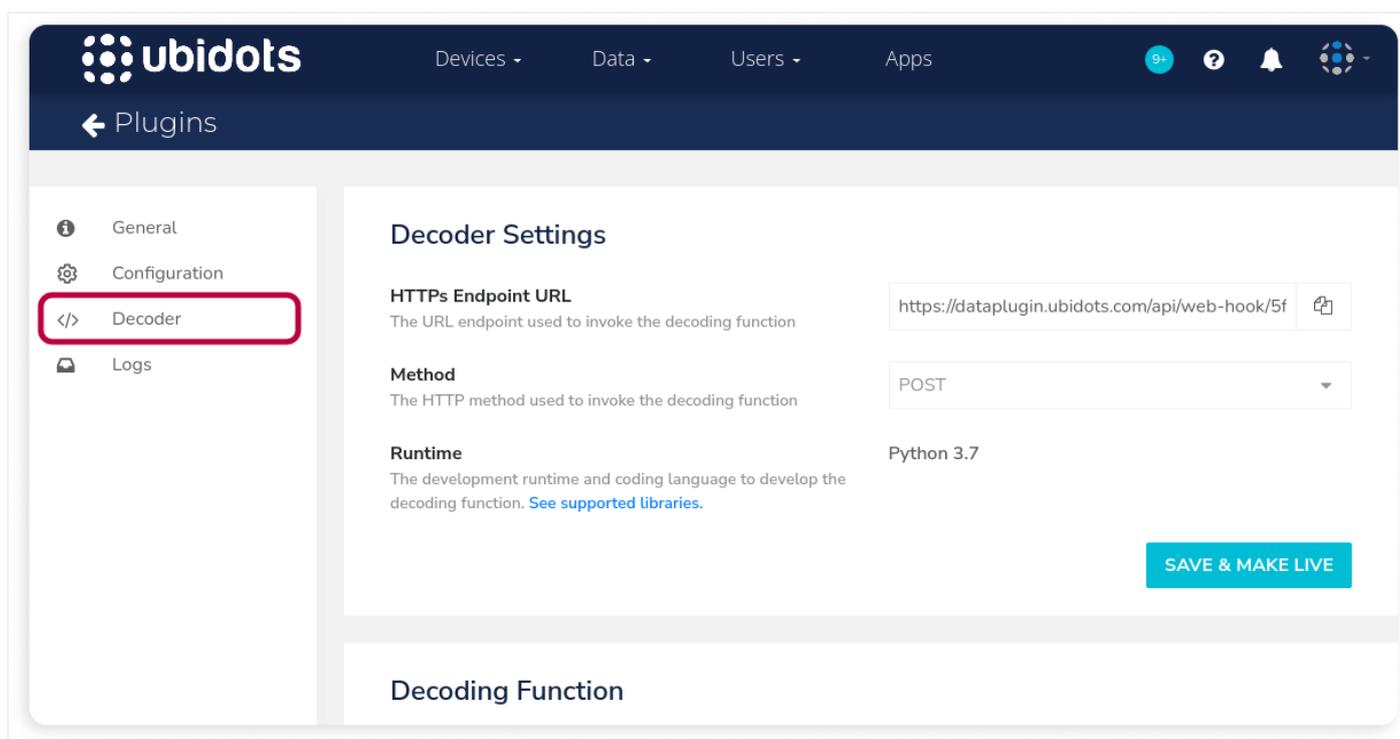
Prepared By
NGB

Approved By
NB

Title
Connecting a Senquip Device to Ubidots

Page
2 of 7

3. The plugin setup wizard will appear, click *Next*.
4. Select the *Default token* or create a new token dedicated to this plugin only. Click *Next*.
5. Give the plugin a name and description and click *Next* to finish the process. You will see the new plugin listed on the Plugins list.
6. Click on the newly created plugin and navigate to the Decoder section on the left menu.
7. Copy the *HTTPs Endpoint URL* field. This is the URL you will be using to set up the HTTP endpoint on the Senquip device.



The screenshot shows the Ubidots web interface. At the top, there's a navigation bar with 'ubidots' logo and menu items: Devices, Data, Users, Apps. On the right, there are notification icons. Below the navigation bar, there's a 'Plugins' section with a left-hand menu containing 'General', 'Configuration', 'Decoder' (highlighted with a red box), and 'Logs'. The main content area is titled 'Decoder Settings' and contains three fields: 'HTTPs Endpoint URL' with the value 'https://dataplugin.ubidots.com/api/web-hook/5f', 'Method' set to 'POST', and 'Runtime' set to 'Python 3.7'. A 'SAVE & MAKE LIVE' button is located at the bottom right of the settings area.

2.2. Configuring the Senquip Device

Senquip devices can send data directly to third-party endpoints via UDP, MQTT, and HTTP. Secure connections are enabled by uploading certificates to the Senquip device.

Senquip devices starting from firmware version SFW002 are equipped with Transport Layer Security (TLS) that aligns with Ubidots' specifications. These devices also come preloaded with a CA Certificate in their trust store for Ubidots. If an alternative certificate is needed, you can obtain it by downloading it from [here](#).

We will now configure the Senquip device endpoint to send data securely to Ubidots using HTTPS.

1. Log in to your [Senquip Portal account](#), click on the device that you want to use, open the Settings page and select *Endpoint* settings.

2. Tick the *HTTP Post* option.
3. In *HTTP Address*, enter the plugin URL from Section 2.1.
4. If you need an alternate certificate, use the *Configure HTTPS* option to upload your certificate.
5. Press *Save Settings* and wait for the changes to be applied to the device.

You should now be receiving data on Ubidots.

3. Connecting Without a Plugin

Senquip devices can send data directly to Ubidots using the [Ubidots V1.6 API](#). First, a simple script will be written for the Senquip device to post a standard Senquip JSON format data packet to Ubidots. The script is then enhanced to create a custom data packet containing only the measurements (dots) that need to be sent. By limiting the number of measurements sent, the cost of a Ubidots subscription can be managed.

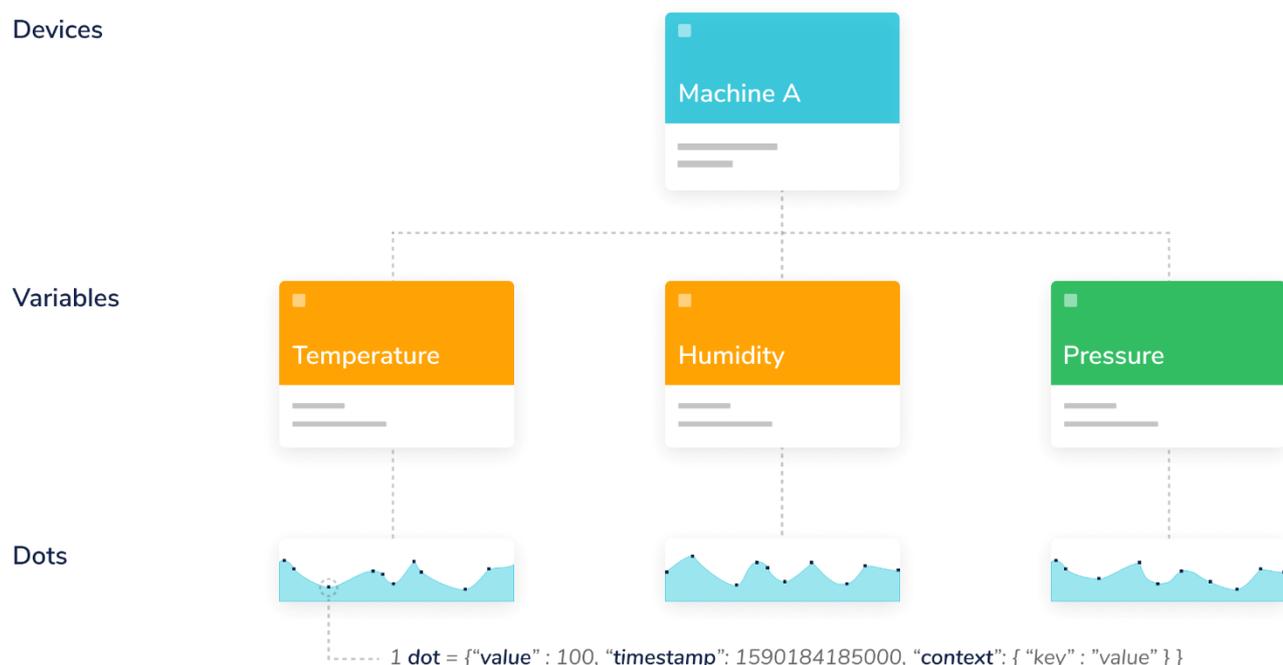


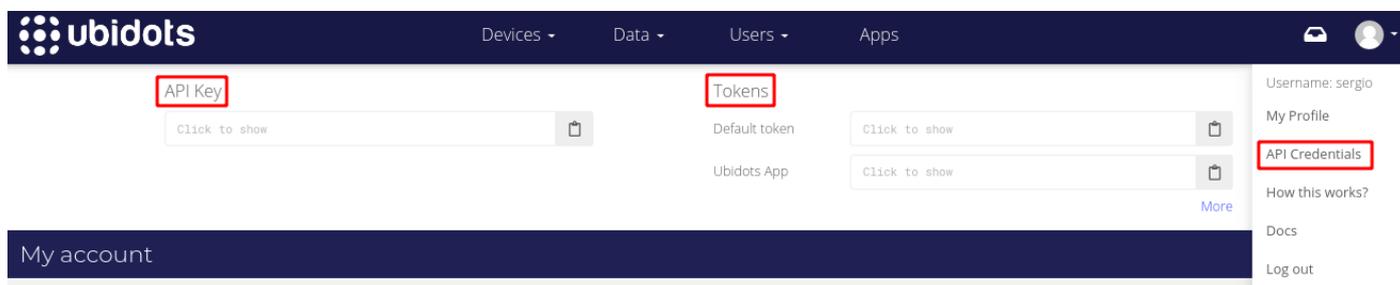
Figure 1 - Description of Ubidots Dot

3.1. Finding Your Token from Ubidots Account

Every request sent to the Ubidots API requires a token. A token is a unique key that authorizes requests sent to Ubidots. Your user token can be found by clicking on *API Credentials* from the *User* dropdown.

1. Sign into your Ubidots account.

2. Go to your *User* dropdown and click on *API credentials*.
3. Copy your user token.



The Token will be used in a script on your Senquip device when communicating with Ubidots.

You do not need to create a device in Ubidots, the device will be created automatically when the Senquip device makes contact.

3.1. Configuring the Senquip Device

Senquip devices can send data directly to third-party endpoints via UDP, MQTT, and HTTP. Secure connections are enabled by uploading certificates to the Senquip device.

Senquip devices starting from firmware version SFW002 are equipped with Transport Layer Security (TLS) that aligns with Ubidots' specifications. These devices also come preloaded with a CA Certificate in their trust store for Ubidots. Nonetheless, if an alternative certificate is needed, you can obtain it by downloading it from [here](#).

We will now configure the Senquip device endpoint to send data securely to Ubidots using HTTPS.

1. Log in to your [Senquip Portal account](#), click on the device that you want to use, open the Settings page and select *Endpoint settings*.
2. Untick the *Use Senquip Data Format* option. This prevents standard Senquip data messages from being sent to the HTTP endpoint.
3. Tick the *HTTP Post* option.
4. In *HTTP Address*, enter the *Server Address*: https://industrial.api.ubidots.com/api/v1.6/devices/DEVICE_NAME where *DEVICE_NAME* is a unique name by which your Senquip device will be known in Ubidots. We will use the Senquip device ID.
5. If you need an alternate certificate, use the *Configure HTTPS* option to upload your certificate.
6. Press *Save Settings* and wait for the changes to be applied to the device.

You will not receive data on Ubidots until a suitable script is written for the Senquip device.

3.2. Writing the Senquip Device Script

A script will now be written on the Senquip device that posts the standard Senquip JSON format data packet to Ubidots. The script will then be updated to create a custom packet that only sends a few variables. Further information on the Senquip scripting language is available [here](#).

Raw Data	272 bytes
<pre> { "ts": 1693198615.8, "cp1": 6.8, "ambient": 22.5, "vbat": 0, "aws_ts": 1693198615, "wifi_rssi": -60, "light": 1, "state": 0, "wifi_ip": "192.168.1.130", "deviceid": "EU8D75173", "events": [{ "msg": "Low", "topic": "vin", "lv": 20 }, { "msg": "Hibernate Mode", "topic": "sys", "lv": 20 }], "vin": 0, "vsys": 3.67 } </pre>	
28-Aug-23 14:56:55	272 bytes

Figure 2 - Typical Default Senquip JSON Data Packet

The basic script that sends the default Senquip data packet is very simple. The *Senquip* and *api_endpoint* libraries are included. The *api_endpoint* contains functions for sending data to custom endpoints.

By default, the standard Senquip data packet is passed to the *data_handler* that runs after each measurement cycle.

Ubidots specifies header information that will be used in the HTTP post. The header information includes a reference to the token that we copied earlier and specified the data type as JSON. Note: The format of the header information is critical and must be copied exactly.

The post function sends the data to the endpoint specified in the HTTP settings specified earlier.

```

load('senquip.js');
load('api_endpoint.js');

SQ.set_data_handler(function (data) {

  let header = "Content-Type: application/json\r\nX-Auth-Token: XXX-xxxxxxxxxxx\r\n";
  HTTP.post(data, header);

}, null);

```

Once the script is downloaded to the Senquip device, data will start arriving on Ubidots.

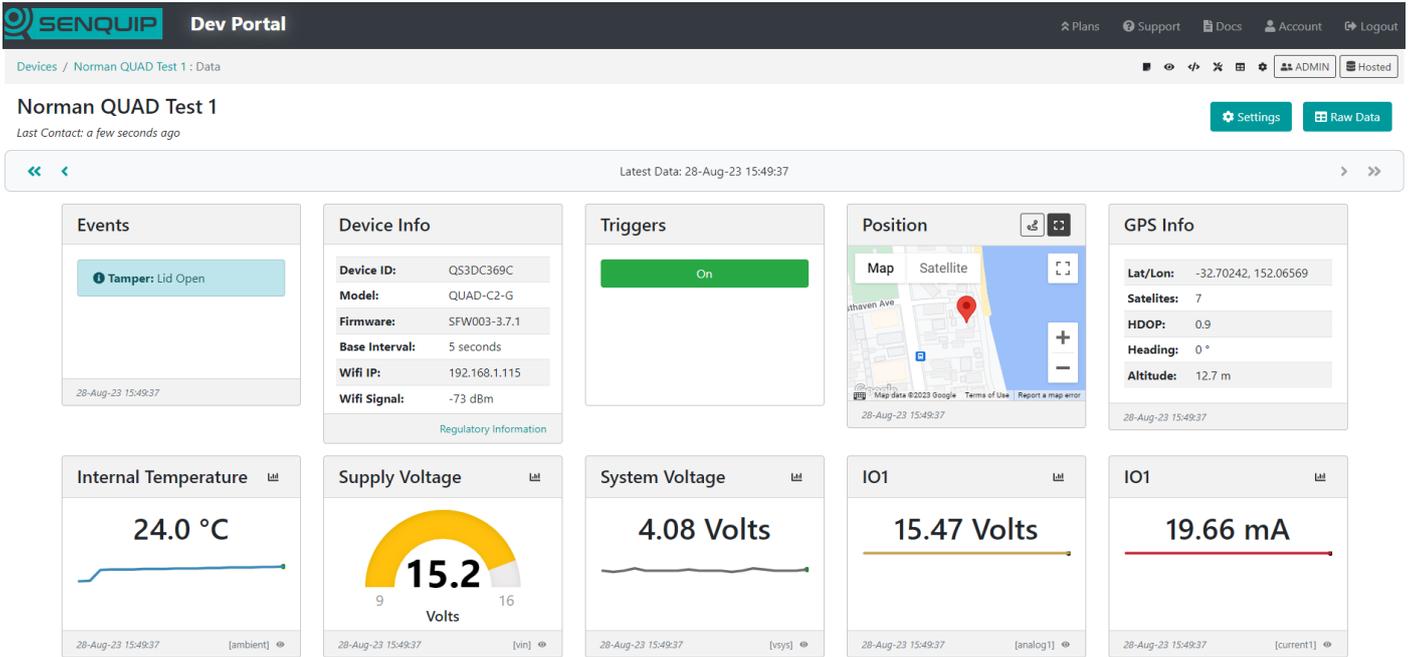


Figure 3 - Data Shown on Senquip Portal

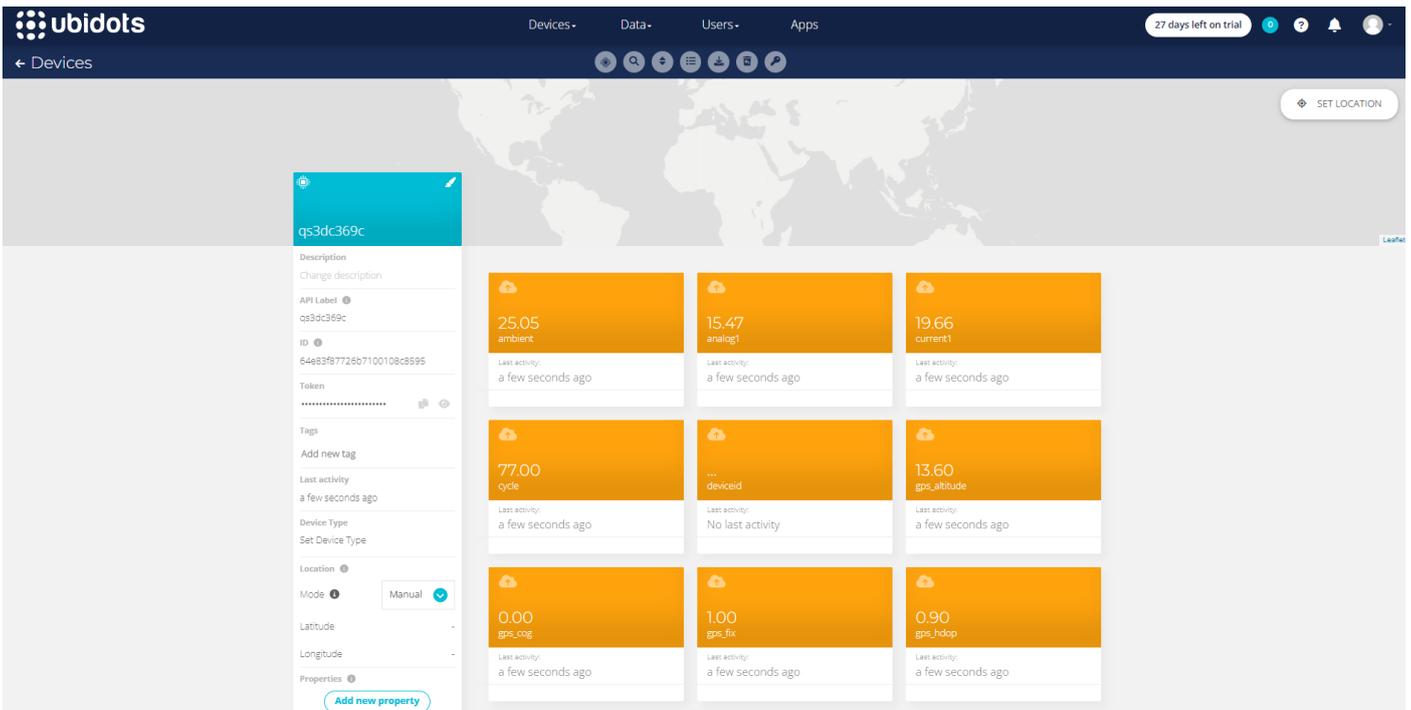


Figure 4 - Data Shown on Ubidots

The script will now be enhanced to create a custom JSON data packet that only sends the device temperature and GPS position.

A structure called *message* is created that includes temperature and position. The format of the position data is found in the Ubidots [API description](#).

The structure is converted into a JSON object using the `JSON.stringify` function and is sent using the `HTTP.post` function.

```
load('senquip.js');
load('api_endpoint.js');
SQ.set_data_handler(function (data) {

    let obj = JSON.parse(data);

    let header = "Content-Type: application/json\r\nX-Auth-Token: XXXX-xxxxxxxxxxxx\r\n";
    let message = {
        temperature: {value: obj.ambient, timestamp: obj.ts},
        position: {latitude: obj.gps_lat, longitude: obj.gps_lon}
    };

    HTTP.post(JSON.stringify(message), header);

}, null);
```

Far fewer variables (dots) are now received by Ubidots. The device location has automatically been recognised by Ubidots and displayed on a map.

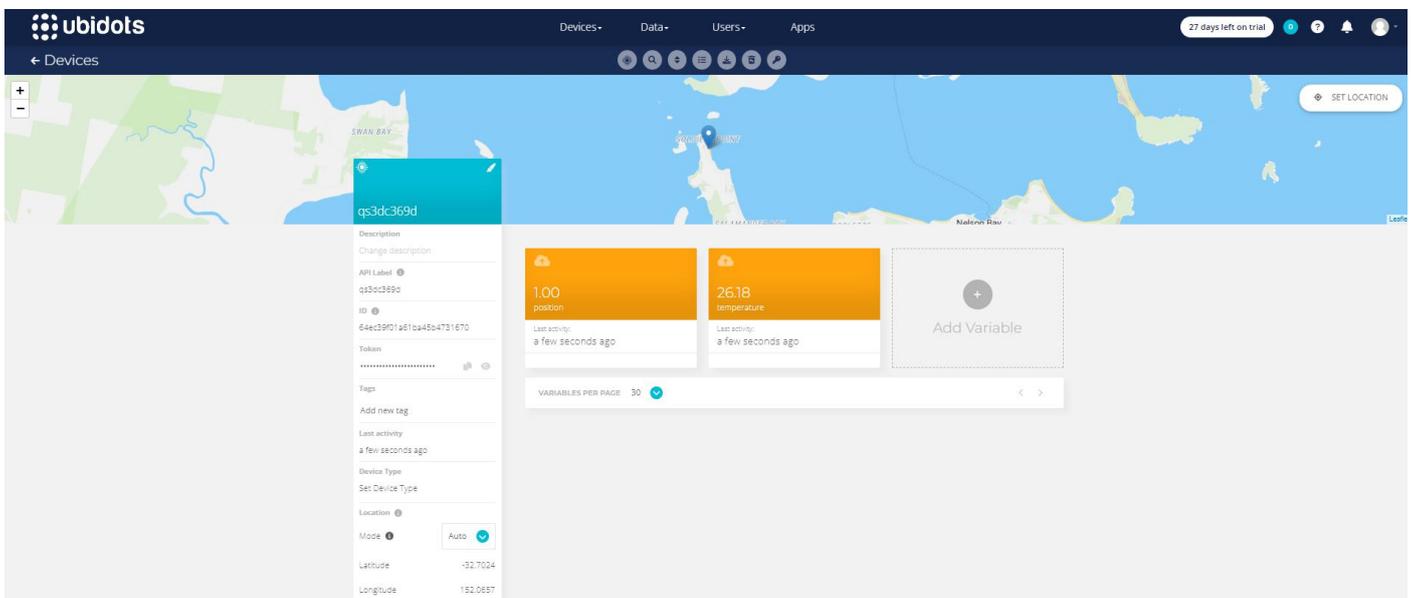


Figure 5 - Ubidots Display of Minimised Data Packet