**DATASHEET**

**ECO Sensor Kit**

**Release 2, Feb 2020**

**Copyright 2020 LLC VidaLink. All rights reserved.**

Table 1: Release History

**Release Date Description**

1 02/07/2019 Second release

2 07/02/2020 Second release

The latest release of this document can be found at [airtist.me](http://airtist.me/).

**Contents**

[1 Introduction 3](#_Toc31974805)

[2 Features 4](#_Toc31974806)

[2.1 Hardware 4](#_Toc31974807)

[2.2 Peripherals 4](#_Toc31974808)

[2.3 Software 4](#_Toc31974809)

[3 Block Diagram 5](#_Toc31974810)

[4 Mechanical Specification 6](#_Toc31974811)

[5 Electrical Specification 9](#_Toc31974812)

[6 Peripherals 9](#_Toc31974813)

[6.1 Sensors 9](#_Toc31974814)

[6.2 Screen 9](#_Toc31974815)

[7 Availability 10](#_Toc31974816)

[8 Eco Sensor kit integration with other platforms and devices 10](#_Toc31974817)

[8.1 Introduction 10](#_Toc31974820)

[8.2 Get data from eco sensor kit to personal computer. 12](#_Toc31974821)

[8.3 Get data from eco sensor kit to Android device. 15](#_Toc31974822)

[Through OTG cable and standard com port terminal 15](#_Toc31974823)

[Through BLE 15](#_Toc31974824)

[8.4 Get data from the Eco sensor kit platform to your server. 16](#_Toc31974825)

[9 Support 17](#_Toc31974826)

# Introduction

The ECO sensor kit Module is a range -mechanically-compatible System on Modules (SoMs) containing processor, peripherals, set of sensors and supporting power circuitry. These modules allow a designer to leverage the ECO sensor kit hardware and software stack in their own custom systems and form factors. In addition, these modules have extra IO interfaces over and above, opening up more options for the designer.

# Features

## Hardware

* + - ESP 32 based
    - 3G, NB-IoT, LoRa, BLE, Wi-Fi as communications methods
    - Set of important sensors
    - Low cost
    - Low power
    - High availability
    - High reliability

## Peripherals

* + - 4x GPIO
    - 1x I2C
    - 1x SPI

## Software

* + - ThingSpeak open IoT Planform

# Block Diagram

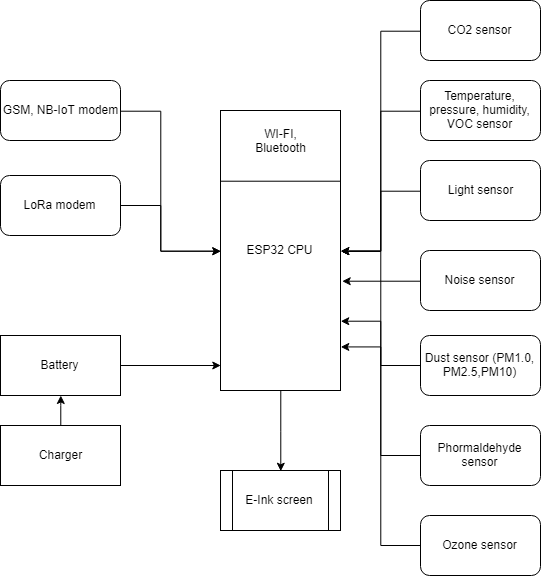


Figure 1: ECO Sensor kit block diagram

# Mechanical Specification

The ECO sensor kit modules conform

The maximum component height on the underside of the box is 1.2mm.

The maximum component height on the top side of the box is 2.5mm.

The Module PCB thickness is 1.6mm +/- 0.1mm.

Note that the location and arrangement of components on the Compute Module may change slightly over time due to revisions for cost and manufacturing considerations; however, maximum component heights and PCB thickness will be kept as specified.

Figure [2](#_bookmark9) gives the ECO sensor kit mechanical dimensions.

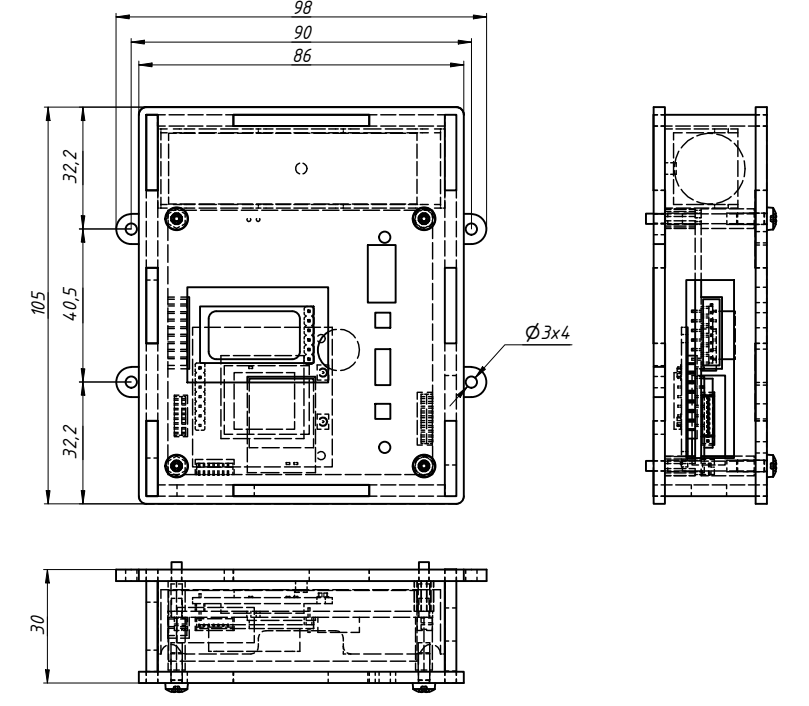


Figure 2: ECO sensor kit Mechanical Dimensions

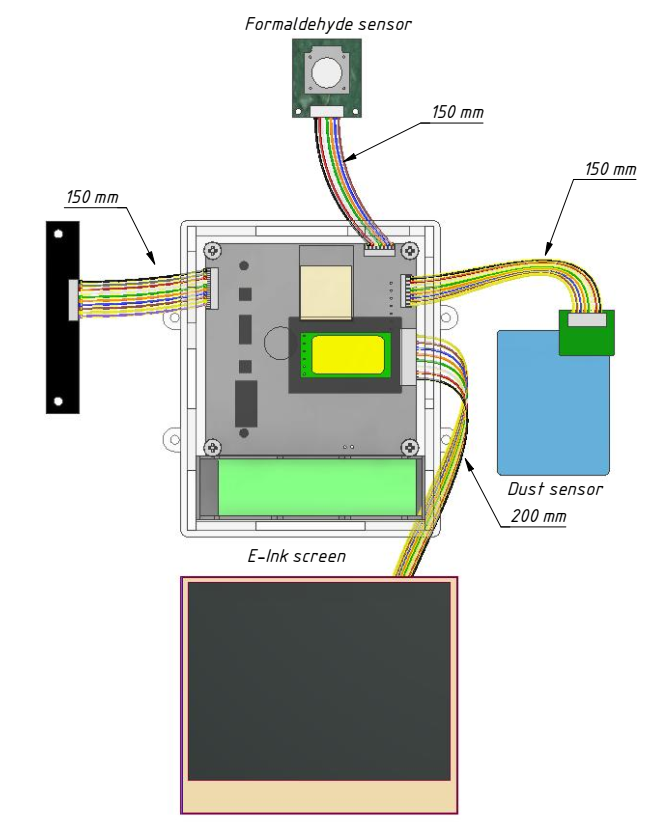


Figure 3: ECO sensor kit Mechanical Dimensions of full set of sensors

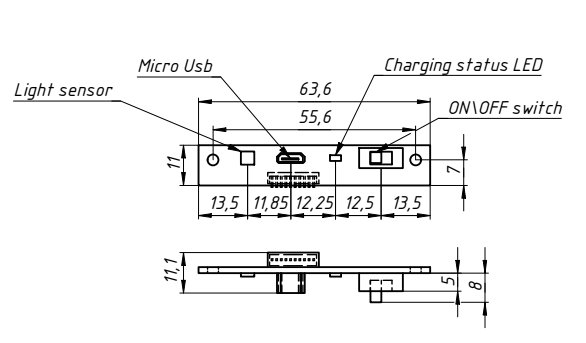


Figure 4: ECO sensor kit front board

# Electrical Specification

**Caution!** Stresses above those listed in Table [4](#_bookmark14) may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Minimum** | **Maximum** | **Unit** |
| VBAT | Li-Ion battery voltage | 2.8 | 4.3 | V |
| 3V3 | 3V3 Supply Voltage | 3 | 3.5 | V |
| 5V | 5V Supply Voltage | 4.5 | 6 | V |

Table 1: Absolute Maximum Ratings

# Peripherals

## Sensors

**MEASUREMENTS RANGE AND TOLERANCE**

|  |  |  |
| --- | --- | --- |
| Sensor | Measurements range | Tolerance |
| CO2 (Telaire T6703)  TVOC (BOSCH BME 680)  O3\* (Winsen ZE25-O3)  HCHO\* (Winsen ZE08-CH2O)  PM1.0 (Plantover PMS7003/G7)  PM2.5 (Plantover PMS7003/G7)  PM5.0 (Plantover PMS7003/G7)  PM10 (Plantover PMS7003/G7)  Temperature (BOSCH BME680)  Pressure (BOSCH BME680)  Humidity (BOSCH BME680)  Light (TI OPT3001)  Noise (AD DMP401) | 400ppm – 64000ppm  0ppb – 64000ppb  0-100ppm  0-1000ppm  0-10000 µg/m3  0-1000 µg/m3  0-200 µg/m3  0-100 µg/m3  -20 ºC ... 80 °C  80-130 kPa  10–90%(non condensing)  0.53 – 2220000 Lux  25– 110dB | ± 50 ppm  ± 10 %  ± 10 %  ± 8 %  ± 10 %  ± 10 %  ± 10 %  ± 10 %  ± 3 %  ± 2 %  ± 3 %  ± 3 %  ± 10 % |

***\* Optional***

Table 2: Measurements range and tolerance of sensors

## Screen

Overview

This is an E-Ink display module, 4.2inch, 400x300 resolution, with embedded controller, communicating via SPI interface.

Due to the advantages like ultra low power consumption, wide viewing angle, clear display without electricity, it is an ideal choice for applications such as shelf label, industrial instrument, and so on.

Features:

No backlight, keeps displaying last content for a long time even when power down;

Ultra low power consumption, basically power is only required for refreshing;

SPI interface, for connecting with controller boards like Raspberry Pi/Arduino/Nucleo, etc.;

Onboard voltage translator, compatible with 3.3V/5V MCUs;

Nano/Arduino/STM32);

Specifications;

Operating voltage: 3.3V~5V;

Interface: 3-wire SPI, 4-wire SPI;

Outline dimension: 103.0mm × 78.5mm;

Display size: 84.8mm × 63.6mm;

Dot pitch: 0.212 × 0.212;

Resolution: 400 × 300;

Display color: black, white;

Grey level: 2;

Full refresh time: 4s;

Refresh power: 26.4mW(typ.);

Standby power: <0.017mW;

Viewing angle: >170°.

# Availability

ECO sensor kit is available since August 2019.

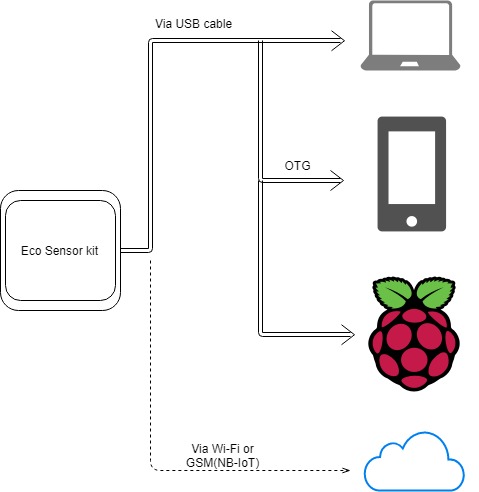
# Eco Sensor kit integration with other platforms and devices



## Introduction

There are many ways to connect the Eco sensor kit with various devices for transmitting and using data from sensors.

Consider the following connection options:

1. Personal Computer,
2. Raspberry PI,
3. Android device,
4. Cloud connected device.

In all cases, the information will be stored and either transmitted in JSON format. JSON (JavaScript Object Notation) is a common format for representing values ​​and objects. Its description is documented in RFC 4627. It was originally created for JavaScript, but many other languages ​​also have libraries that can work with it. Thus, JSON is easy to use for exchanging data when the client uses JavaScript and the server is written on Ruby / PHP / Java / any other language.

Example data line:

{"Voltage":3.8,"CO2":858,"formaldehyde":25,"PM1\_0":4,"PM2\_5":4,"PM10":6,"Light":35,"Temperature":25,"Pressure":742,"Humidity":48,"AQI":391,"Noise":72},

Where:

**"Voltage" - Voltage on the battery, V;**

**"CO2" - Carbon dioxide, ppm;**

**"Formaldehyde" - Formaldehyde, ppm; \*optional or "O3" - Ozone, ppm; \*optional**

**"PM1\_0" – 1.0 μm dust particles, μg / m3**

**"PM2\_5" - 2.5 μm dust particles, μg / m3**

**"PM10" – 10 μm dust particles, μg / m3**

**"Light" - Illumination, Lux;**

**"Temperature" - Temperature, ⁰C;**

**"Pressure" - Atmospheric pressure, mmHg;**

**"Humidity" - Humidity, %;**

**"AQI" – Air quality index, ppb;**

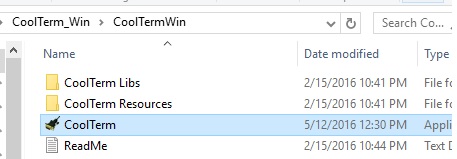
**"Noise" - Noise, dB;**

## Get data from eco sensor kit to personal computer.

There are many software options available, but typically you can use **CoolTerm** which is a serial monitor which will also capture transmitted data to a **text file** and automatically add **time stamps** to each line of data which are essential for a good datalogger.

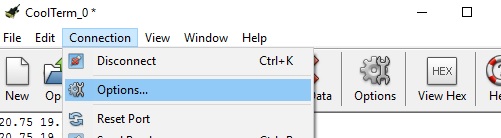
So that every time data is taken from the all sensors, those measurements and also the system status (battery voltage) are output through the serial port to a connected PC. (Full details on generating Serial output from an Arduino are available here: [**Arduino Serial**](https://www.arduino.cc/en/Reference/Serial) from the official Arduino Reference site.)

When that is done, go into the folder created, and double click on the CoolTerm application.

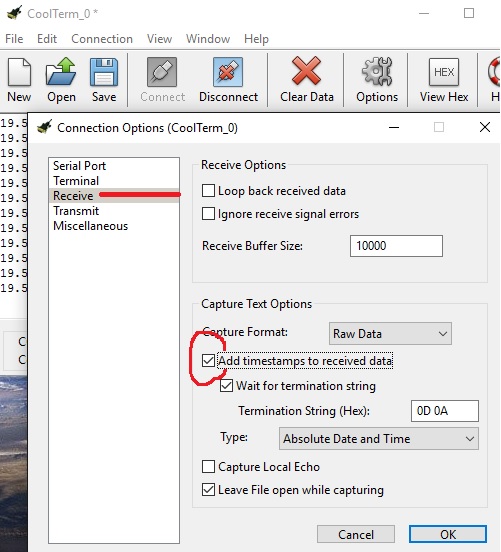
[](http://www.reuk.co.uk/wordpress/wp-content/uploads/2016/05/launch-coolterm.jpg)

Click on Connection > Options and then in Serial Port Options select the Port you would like to use.

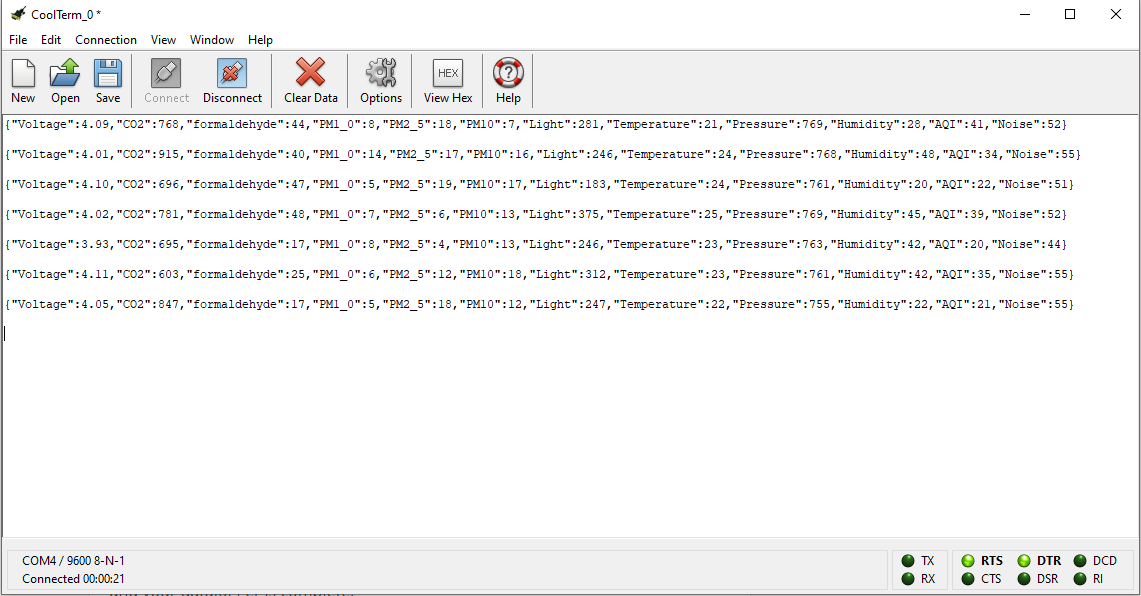
At the same time set the **baudrate to 9600**.

[](http://www.reuk.co.uk/wordpress/wp-content/uploads/2016/05/coolTerm-connection-options.jpg)

Assuming that you would like all data to be **timestamped** (adding the date and time to every line of data sent), do Connection > Options > Receive and check the ‘Add timestamps to received data’ box.

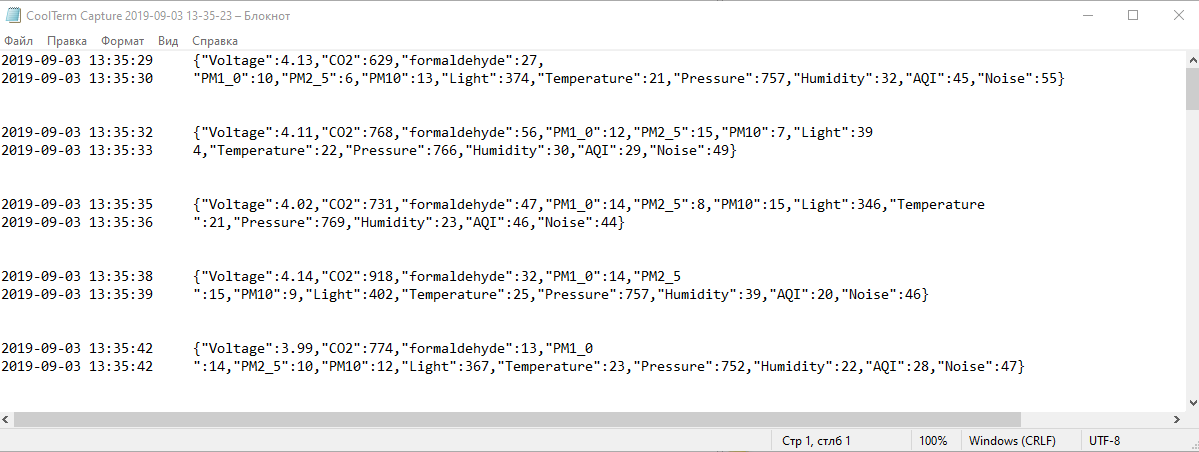
[](http://www.reuk.co.uk/wordpress/wp-content/uploads/2016/05/add-time-stamps-to-arduino-serial-data.jpg)

Then to have any serial data from your Eco sensor kit automatically stored in a text file on the PC, do Connection > Capture to Text File and then click on Start. You then just have to set the name for the file that you would like your data to be stored in, and your datalogger is complete.



To stop collecting data, you can either click on the large Disconnect icon, or if you want to stay connected to the Eco sensor kit board, do Connection > Capture to Text File > Stop.

Once you have either disconnected the Eco sensor kit board or Stopped the capture, you cannot then restart and append data to the same file – you can only overwrite the original file or start a new one. If you want to pause capture and then restart it to append to the same file, do Connection > Capture to Text File > Pause to pause, and then Connection > Capture to Text File > Resume to resume it at a later time.



When you have finished capturing data, you will end up with a text file of everything captured which can be processed and visualized using Excel or a similar application.

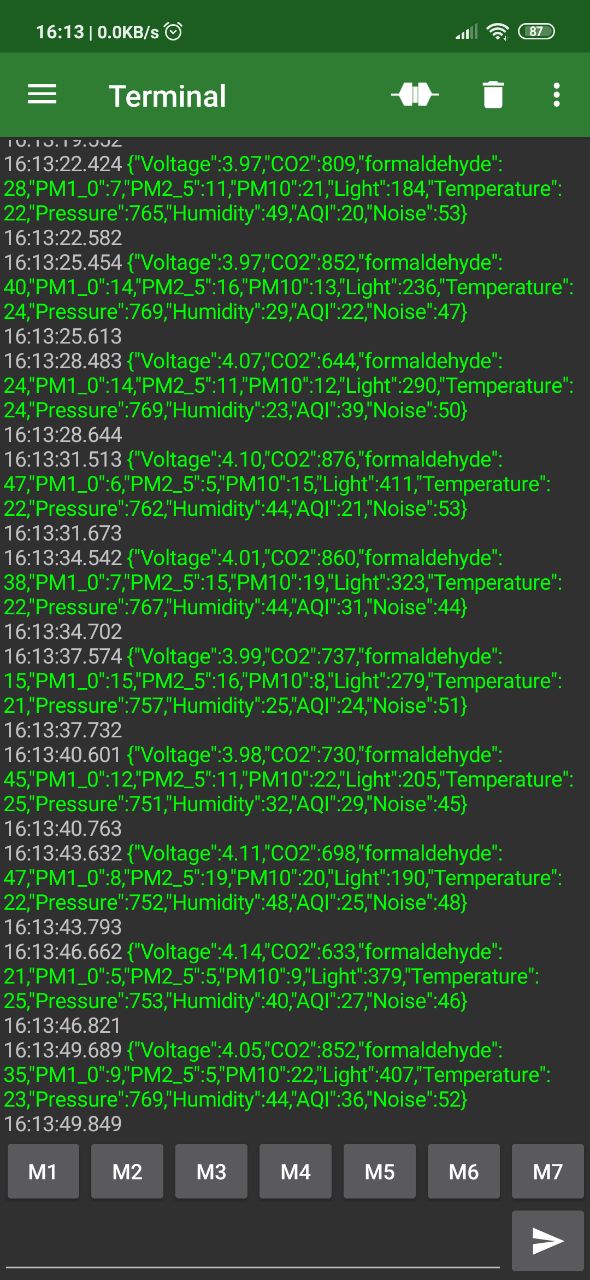
## Get data from eco sensor kit to Android device.

There are two ways how to connect Eco sensor kit with Android device:

### Through OTG cable and standard com port terminal

**USB On-The-Go** (**USB OTG** or just **OTG**) is a specification first used in late 2001 that allows USB devices, such as tablets or smartphones, to act as a host, allowing other USB devices, such as USB flash drives, digital cameras, mice or keyboards, to be attached to them. Use of USB OTG allows those devices to switch back and forth between the roles of host and device. A mobile phone may read from removable media as the host device, but present itself as a USB Mass Storage Device when connected to a host computer.

Just connect Eco Sensor kit trough the OTG cable and download Serial USB Terminal app from Play market. When connect you can see next picture.



### Through BLE

**Bluetooth Low Energy** (**Bluetooth LE**, colloquially **BLE**, formerly marketed as **Bluetooth Smart**[[1]](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy#cite_note-1)) is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group (Bluetooth SIG) aimed at novel applications in the healthcare, fitness, beacons,[[2]](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy#cite_note-2) security, and home entertainment industries.[[3]](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy#cite_note-3) Compared to Classic Bluetooth, Bluetooth Low Energy is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range. Mobile operating systems including iOS, Android, Windows Phone and BlackBerry, as well as macOS, Linux, Windows 8 and Windows 10, natively support Bluetooth Low Energy.

Just connect Eco Sensor kit through the Bluetooth and download Serial USB Terminal app from Play market.

## Get data from the Eco sensor kit platform to your server.

To get data from the platform you need to do the following:

**1.** You need to make a POST request to **http://dashboards.djinnsensor.com/api/auth/login**

Header:

**Content-Type: application/json**

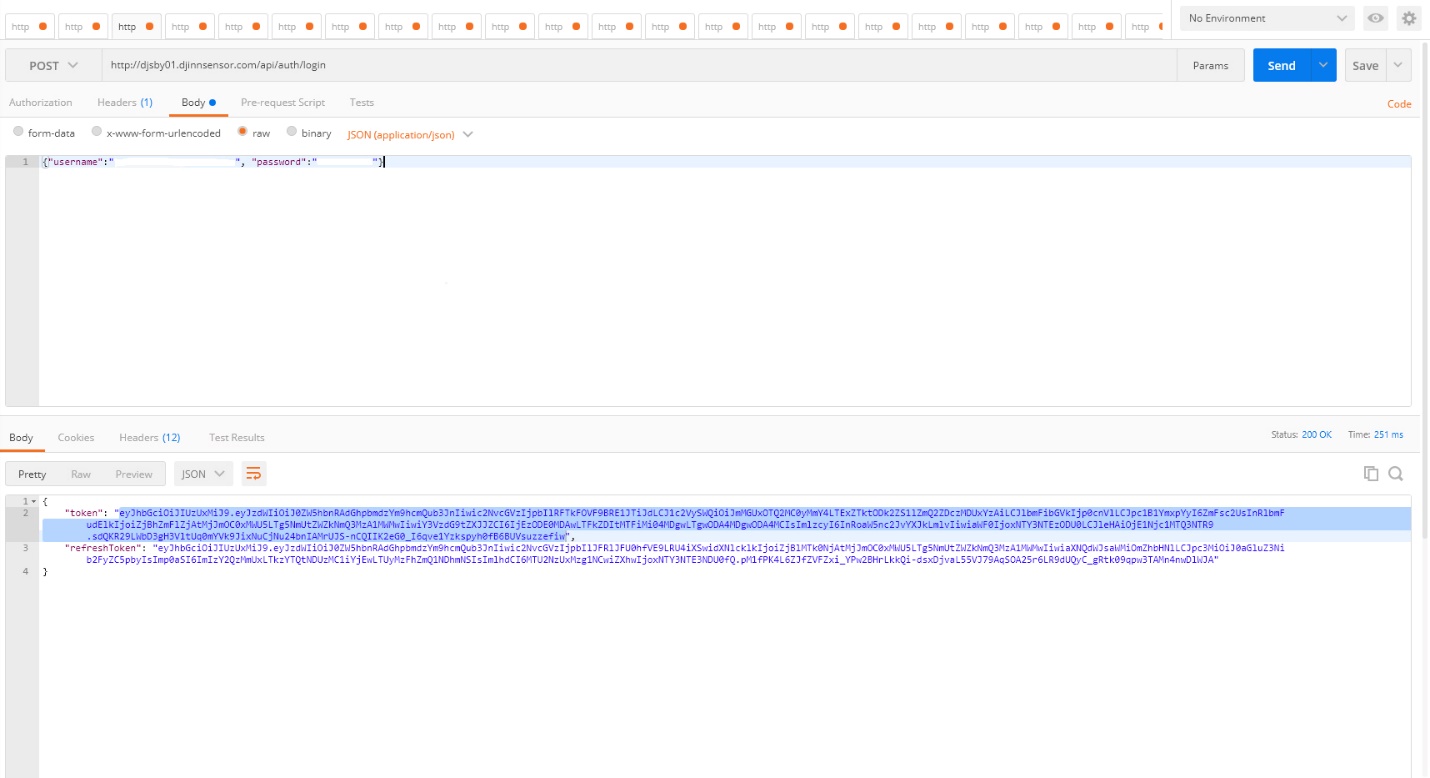
Body:

**{"username":"{YOUR\_USERNAME}", "password":"{YOUR\_PASSWORD}"}**

where you need to replace **{YOUR\_USERNAME}** and **{YOUR\_PASSWORD}** with your username and password from your personal account at **dashboards.djinnsensor.com**

example: **{"username":"admin@djinnsensor.com", "password":"12345678"}**

If you did everything correctly you should get the following answer:

[](https://github.com/destr1ke/REQUEST/blob/master/img/auth.jpg)

You need to copy this "token" and replace below in "X-Authorization".

**2.** Now you need to make a GET request to

**http://dashboards.djinnsensor.com/api/plugins/telemetry/DEVICE/{DEVICEID}/values/timeseries?keys=Temperature,Humidity&startTs=0000000000000&endTs=9999999999999&limit=500**

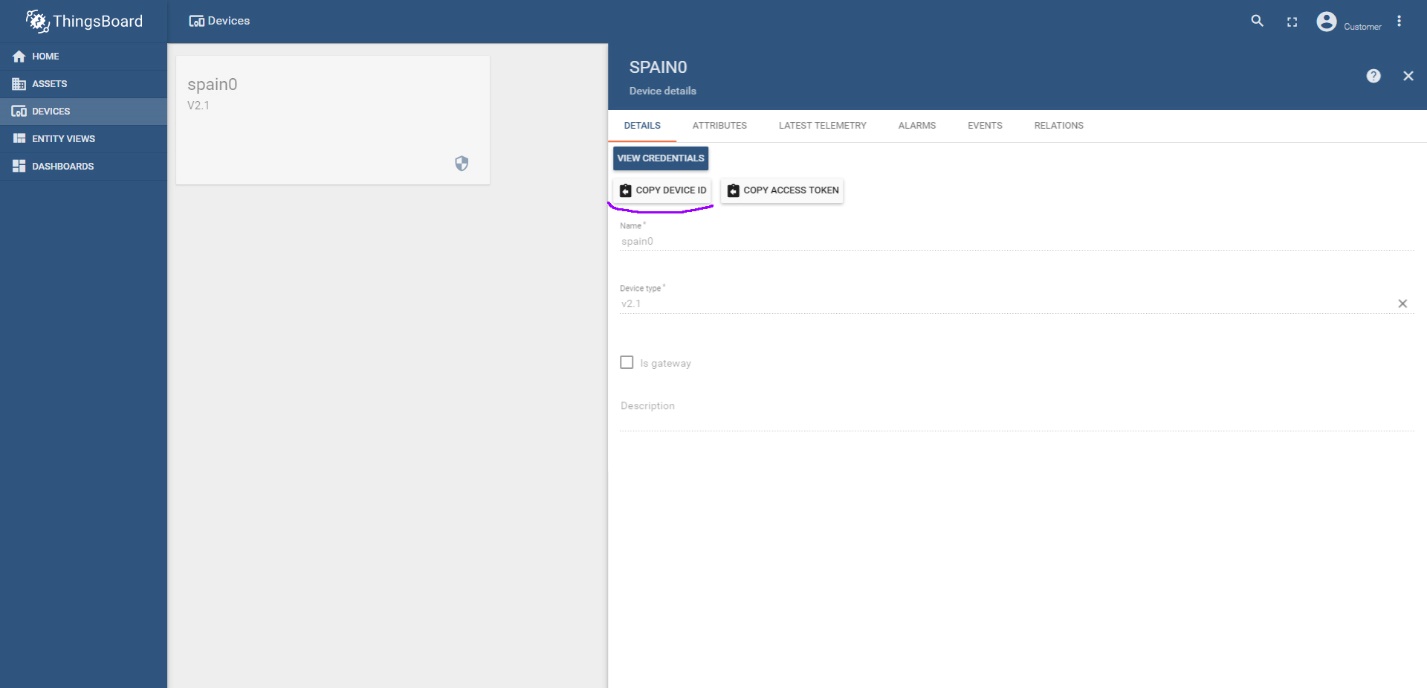
where

**keys** - list the parameters that interest us

**startTs** and **endTs** - time period in timestamp format

**limit** - number of measurements

you need to replace {DEVICEID} with your device id in your personal account, screen below:

[](https://github.com/destr1ke/REQUEST/blob/master/img/screen.jpg)

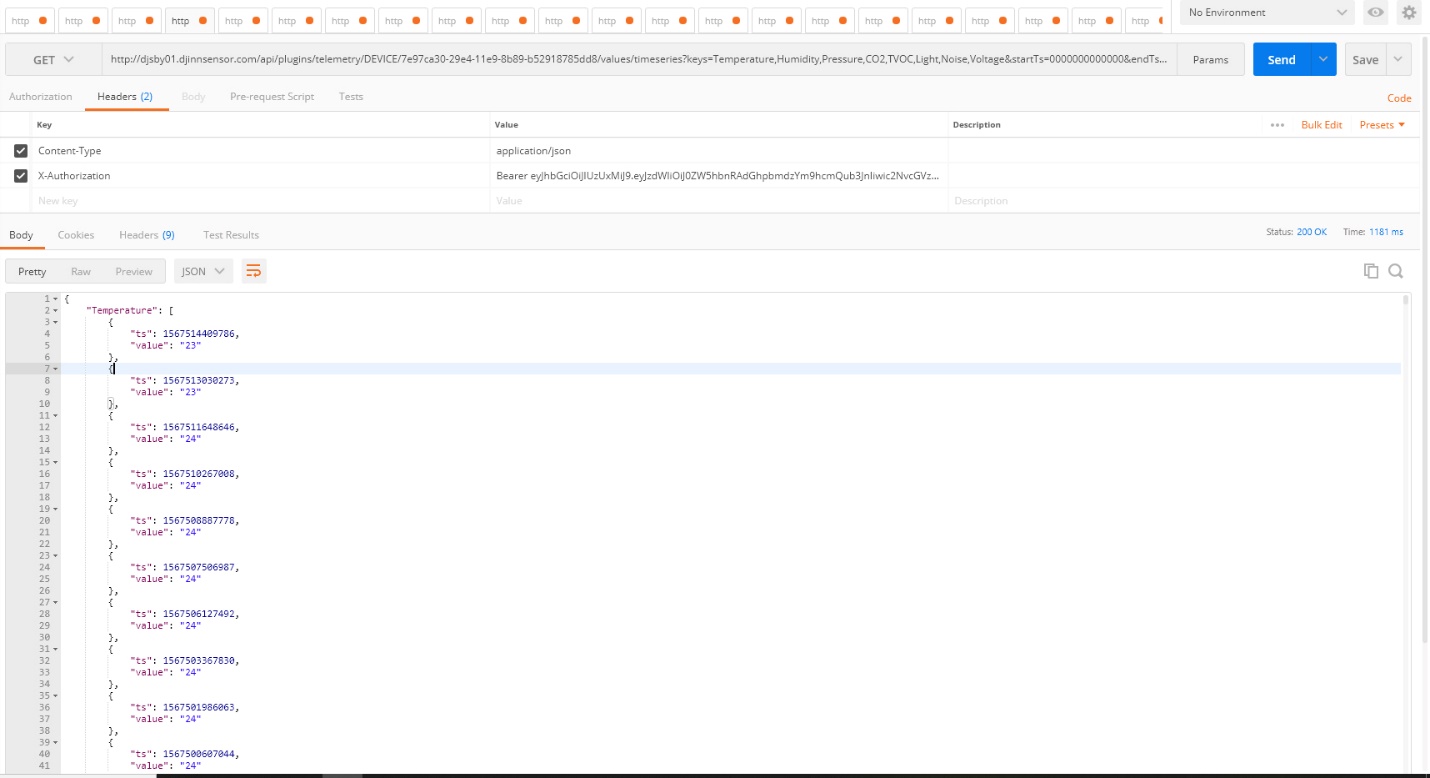
Header:

**"Content-Type": "application/json"**

**"X-Authorization": "Bearer {token}"**

where you need to replace **{token}** from your previous answer to post request

result screen:

[](https://github.com/destr1ke/REQUEST/blob/master/img/result.jpg)

# Support

For support please see the hardware documentation section of the [Raspberry Pi website](https://www.raspberrypi.org/documentation/hardware/) and post ques- tions to the [Raspberry Pi forum.](https://www.raspberrypi.org/forums/)