

---

# **PB01 -- LoRaWAN Push Button User Manual**

last modified by Xiaoling

on 2024/07/05 09:53

---



**Table of** Contents:

- [1. Introduction](#)
  - [1.1 What is PB01 LoRaWAN Push Button](#)
  - [1.2 Features](#)
  - [1.3 Specification](#)
  - [1.4 Power Consumption](#)
  - [1.5 Storage & Operation Temperature](#)
  - [1.6 Applications](#)
- [2. Operation Mode](#)
  - [2.1 How it work?](#)
  - [2.2 How to Activate PB01?](#)
  - [2.3 Example to join LoRaWAN network](#)
  - [2.4 Uplink Payload](#)
    - [2.4.1 Uplink FPORT=5, Device Status](#)
    - [2.4.2 Uplink FPORT=2, Real time sensor value](#)
      - Battery:
      - Sound\_ACK & Sound\_key:
      - Alarm:
      - Temperature:
      - Humidity:
    - [2.4.3 Uplink FPORT=3, Datalog sensor value](#)
    - [2.4.4 Decoder in TTN V3](#)
  - [2.5 Show data on Datacake](#)
  - [2.6 Datalog Feature](#)
    - [2.6.1 UnixTimeStamp](#)
    - [2.6.2 Poll sensor value](#)
    - [2.6.3 Datalog Uplink payload](#)
  - [2.7 Button](#)
  - [2.8 LED Indicator](#)
  - [2.9 Buzzer](#)
- [3. Configure PB01 via AT command or LoRaWAN downlink](#)
  - [3.1 Downlink Command Set](#)
  - [3.2 Set Password](#)
  - [3.3 Set button sound and ACK sound](#)
  - [3.4 Set buzzer music type\(0~4\)](#)
  - [3.5 Set Valid Push Time](#)

- [4. Battery & How to replace](#)
  - [4.1 Battery Type and replace](#)
  - [4.2 Power Consumption Analyze](#)
- [5. Accessories](#)
- [6. FAQ](#)
  - [6.1 How to use AT Command to configure PB01](#)
  - [6.2 AT Command and Downlink](#)
  - [6.3 How to upgrade the firmware?](#)
    - [6.3.1 Update firmware \(Assume device have bootloader\)](#)
    - [6.3.2 Update firmware \(Assume device doesn't have bootloader\)](#)
  - [6.4 How to change the LoRa Frequency Bands/Region?](#)
  - [6.5 Why i see different working temperature for the device?](#)
- [7. Order Info](#)
  - [7.1 Main Device](#)
- [7. Packing Info](#)
- [8. Support](#)
- [9. Reference material](#)
- [10. FCC Warning](#)

# 1. Introduction

## 1.1 What is PB01 LoRaWAN Push Button

PB01 LoRaWAN Push Button is a LoRaWAN wireless device with one **push button**. Once user push the button, PB01 will transfer the signal to IoT server via Long Range LoRaWAN wireless protocol. PB01 also senses the **environment temperature & humidity** and will also uplink these data to IoT Server.

PB01 supports **2 x AAA batteries** and works for a long time up to several years\*. User can replace the batteries easily after they are finished.

PB01 has a built-in speaker, it can pronouns different sound when press button and get reply from server. The speaker can by disable if user want it.

PB01 is fully compatible with LoRaWAN v1.0.3 protocol, it can work with standard LoRaWAN gateway.

\*Battery life depends how often to send data, please see [battery analyzer](#).

## 1.2 Features

- Wall Attachable.
- LoRaWAN v1.0.3 Class A protocol.
- 1 x push button. Different Color available.
- Built-in Temperature & Humidity sensor
- Built-in speaker
- Frequency Bands: CN470/EU433/KR920/US915/EU868/AS923/AU915
- AT Commands to change parameters
- Remote configure parameters via LoRaWAN Downlink
- Firmware upgradable via program port
- Support 2 x AAA LR03 batteries.
- IP Rating: IP52

## 1.3 Specification

### Built-in Temperature Sensor:

- Resolution: 0.01 °C
- Accuracy Tolerance: Typ  $\pm 0.2$  °C
- Long Term Drift: < 0.03 °C/yr

- Operating Range: -10 ~ 50 °C or -40 ~ 60 °C (depends on battery type, see [FAQ](#))

#### **Built-in Humidity Sensor:**

- Resolution: 0.01 %RH
- Accuracy Tolerance: Typ  $\pm 1.8$  %RH
- Long Term Drift: < 0.2% RH/yr
- Operating Range: 0 ~ 99.0 %RH(no Dew)

## **1.4 Power Consumption**

PB01 : Idle: 5uA, Transmit: max 110mA

## **1.5 Storage & Operation Temperature**

-10 ~ 50 °C or -40 ~ 60 °C (depends on battery type, see [FAQ](#))

## **1.6 Applications**

- Smart Buildings & Home Automation
- Logistics and Supply Chain Management
- Smart Metering
- Smart Agriculture
- Smart Cities
- Smart Factory

# **2. Operation Mode**

## **2.1 How it work?**

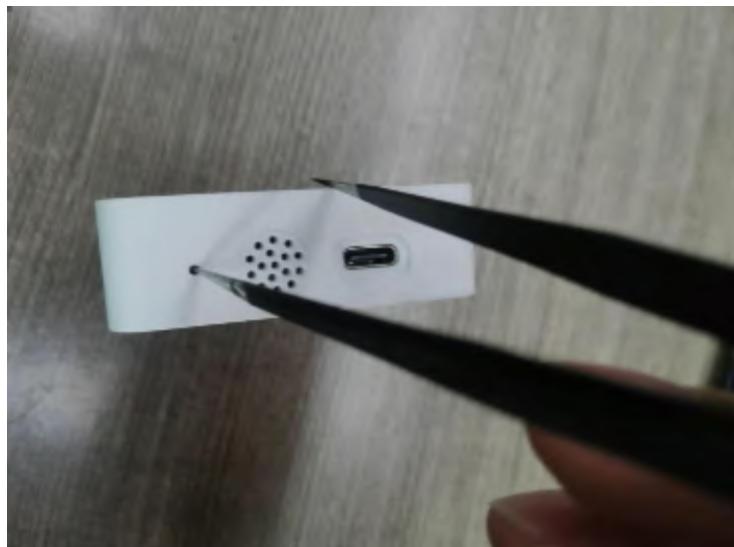
Each PB01 is shipped with a worldwide unique set of LoRaWAN OTAA keys. To use PB01 in a LoRaWAN network, user needs to input the OTAA keys in LoRaWAN network server. After this, if PB01 is under this LoRaWAN network coverage, PB01 can join the LoRaWAN network and start to transmit sensor data. The default period for each uplink is **20 minutes**.

## **2.2 How to Activate PB01?**

1. Open enclosure from below position.



- 2. Insert 2 x AAA LR03 batteries and the node is activated.**
- 3. Under the above conditions, users can also reactivate the node by long pressing the ACT button.**

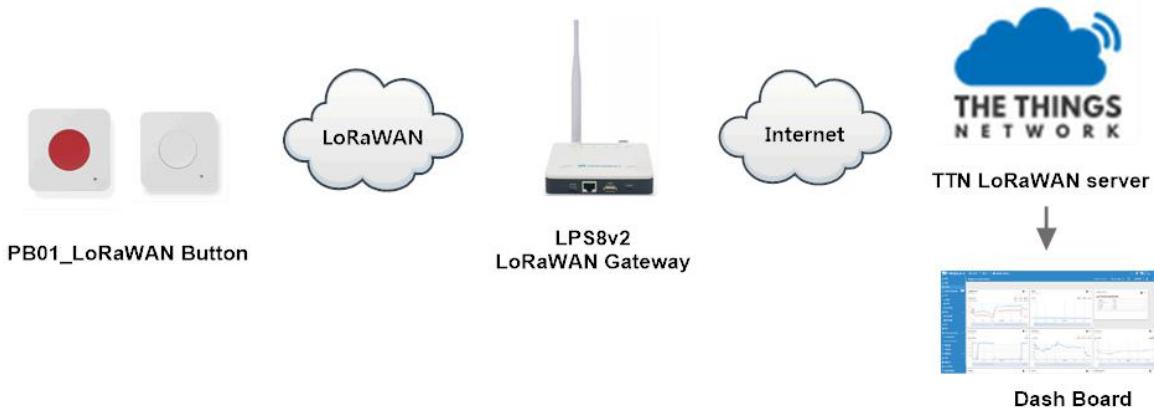


User can check [LED Status](#) to know the working state of PB01.

## 2.3 Example to join LoRaWAN network

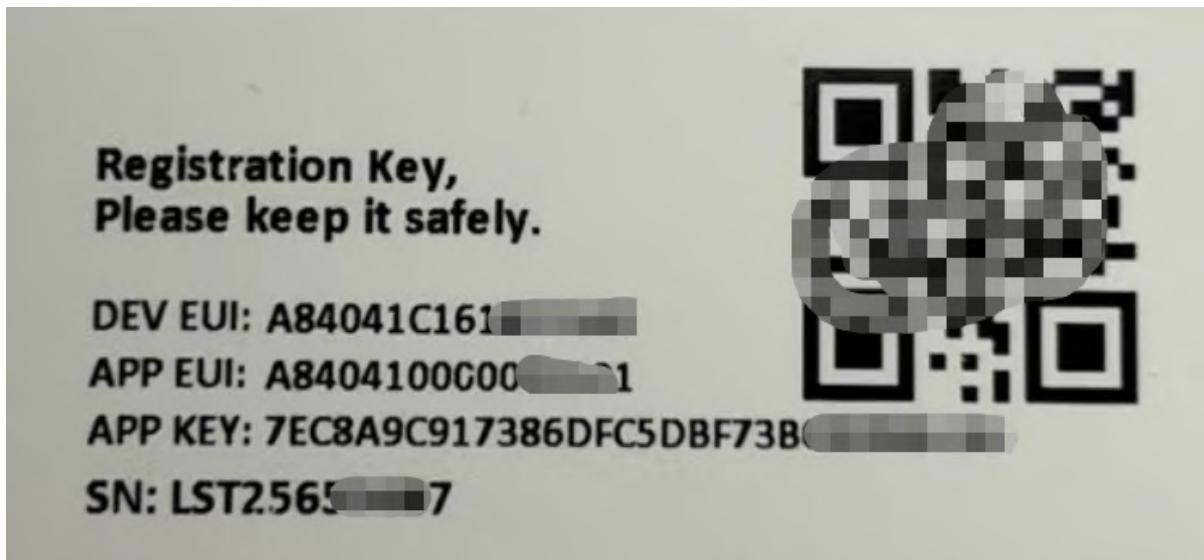
This section shows an example for how to join the [TheThingsNetwork](#) LoRaWAN IoT server. Usages with other LoRaWAN IoT servers are of similar procedure.

Assume the LPS8v2 is already set to connect to [TTN V3 network](#). We need to add the PB01 device in TTN V3 portal.



**Step 1:** Create a device in TTN V3 with the OTAA keys from PB01.

Each PB01 is shipped with a sticker with the default DEV EUI as below:



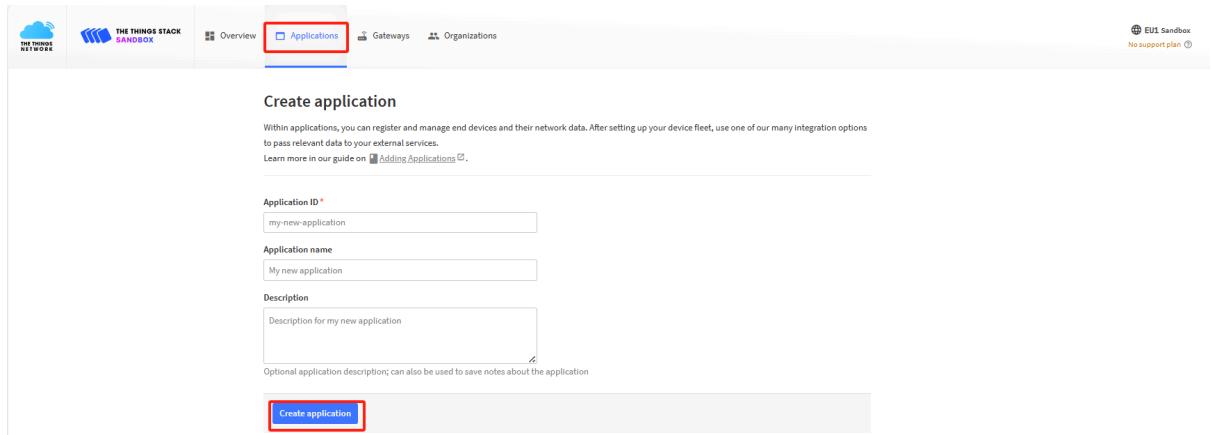
Enter these keys in the LoRaWAN Server portal. Below is TTN V3 screen shot:

Create application.

choose to create the device manually.

Add JoinEUI(AppEUI), DevEUI, AppKey.

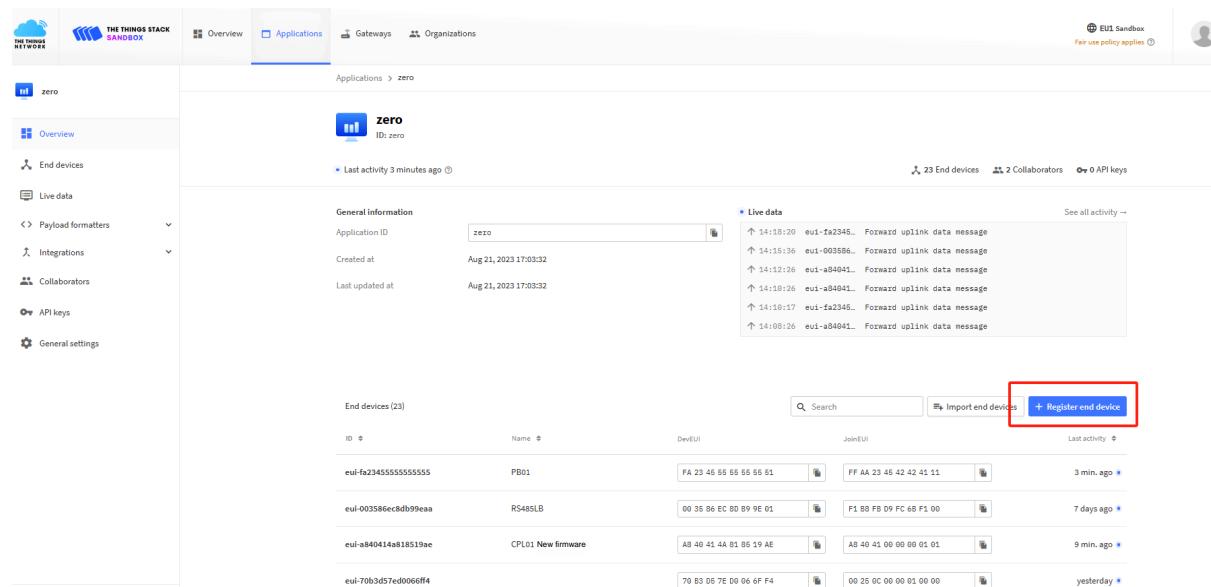
# User Manual for LoRaWAN /NB-IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual



The screenshot shows the 'Create application' form in The Things Stack Sandbox. The 'Applications' tab is selected and highlighted with a red box. The form fields are as follows:

- Application ID \***: my-new-application
- Application name**: My new application
- Description**: Description for my new application
- Optional application description; can also be used to save notes about the application**: (empty)

At the bottom is a blue 'Create application' button, which is also highlighted with a red box.



The screenshot shows the 'zero' application details in The Things Stack Sandbox. The 'Applications' tab is selected. The application details are:

- Application ID**: zero
- Created at**: Aug 21, 2023 17:03:32
- Last updated at**: Aug 21, 2023 17:03:32

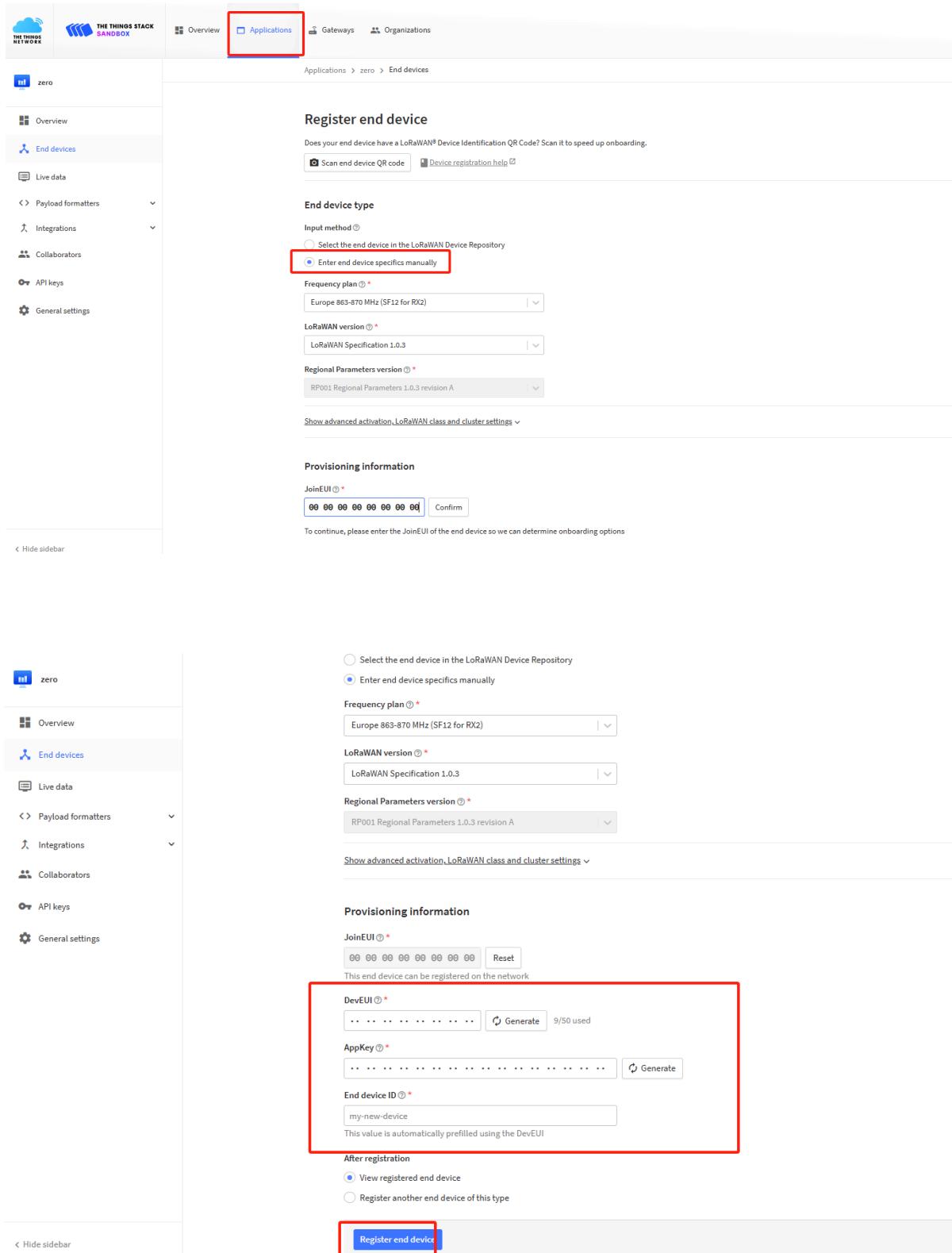
Activity log (Live data):

- ↑ 14:18:20 eui-fa2345... Forward uplink data message
- ↑ 14:18:26 eui-003586... Forward uplink data message
- ↑ 14:18:26 eui-a84041... Forward uplink data message
- ↑ 14:18:26 eui-a84041... Forward uplink data message
- ↑ 14:18:17 eui-fa2345... Forward uplink data message
- ↑ 14:08:26 eui-a84041... Forward uplink data message

End devices (23):

ID	Name	DevEUI	JoinEUI	Last activity
eui-fa23455555555555	PB01	FA 23 45 55 55 55 55 51	FF AA 23 45 42 42 41 11	3 min. ago *
eui-003586ec8db99ea	RS485LB	00 35 86 EC 8D B9 9E 01	F1 B8 FB D9 FC 6B F1 00	7 days ago *
eui-a840414a818519ae	CPL01 New firmware	A8 40 41 4A B1 B5 19 AE	A8 40 41 00 00 00 01 01	9 min. ago *
eui-70b3d57ed0066ff4		70 B3 D5 7E D0 06 6F F4	00 25 0C 00 00 02 00 00	yesterday *

At the top right of the end devices table is a red box around the '+ Register end device' button.



The screenshot shows the 'Register end device' form in The Things Stack Sandbox. The 'Enter end device specifics manually' radio button is selected. The 'JoinEUI' field contains '00 00 00 00 00 00 00 00'. The 'DevEUI' field is highlighted with a red box, containing '...'. The 'AppKey' field contains '...'. The 'End device ID' field contains 'my-new-device'. The 'Register end device' button is highlighted with a red box.

## Default mode OTAA

**Step 2:** Use ACT button to activate PB01 and it will auto join to the TTN V3 network. After join success, it will start to upload sensor data to TTN V3 and user can see in the panel.

## 2.4 Uplink Payload

Uplink payloads include two types: Valid Sensor Value and other status / control command.

- Valid Sensor Value: Use FPORT=2
- Other control command: Use FPORT other than 2.

### 2.4.1 Uplink FPORT=5, Device Status

Users can get the Device Status uplink through the downlink command:

**Downlink: 0x2601**

Uplink the device configures with FPORT=5.

Size(bytes)	1	2	1	1	2
Value	Sensor Model	Firmware Version	Frequency Band	Sub-band	BAT

The screenshot shows the The Things Stack Sandbox interface. At the top, there are navigation links: Overview, Applications (selected), Gateways, and Organizations. On the right, there are EU1 Sandbox and No support plan buttons, and a user profile for paopao.

The main area displays the device status for PB01. The device ID is eu868fa23455555555555. The status shows 2 uplinks and 0 downlinks, with the last activity 9 seconds ago. Below this, there are tabs for Overview, Live data (selected), Messaging, Location, Payload formatters, and General settings. The Live data tab shows a table of recent events:

Time	Type	Details
15:20:40	Schedule data downlink for transmission	DevAddr: 26 08 74 11   FPort: 5   Rx1 Delay: 5
15:20:40	Forward uplink data message	DevAddr: 26 08 74 11   FPort: 5   Payload: { BAT: 3.294, FIRMWARE_VERSION: "1.0.0", FREQUENCY_BAND: "EU868", SENSOR_MODEL: "PB01-L", SUB_BAND: "NULL" }   36 01 00 01 FF 0C 0E   FPort: 5   Data rate: SF7BW125 SNR: 13.8 RSSI: -
15:20:40	Successfully processed data message	DevAddr: 26 08 74 11
15:20:34	Schedule data downlink for transmission	DevAddr: 26 08 74 11   FPort: 1   MAC payload: 1A DC   Rx1 Delay: 5
15:20:34	Forward uplink data message	DevAddr: 26 08 74 11   FPort: 1   Payload: { Alarm: "TRUE", BatV: 3.306, Hum_SHT41: 68, Sound_ACK: "OPEN", Sound_Key: "OPEN", TempC_SHT41: 27.3 }   0C EA 03 01 01 11 02 A8   FPort: 2   Data rate: SF7BW125 SNR: 11.5 RSSI: -
15:20:34	Successfully processed data message	DevAddr: 26 08 74 11
15:20:29	Receive downlink data message	26 01   FPort: 1
15:20:10	Schedule data downlink for transmission	DevAddr: 26 08 74 11   Rx1 Delay: 5

Example Payload (FPort=5): **35 01 00 01 FF 0C DE**

**Sensor Model:** For PB01, this value is 0x35.

**Firmware Version:** 0x0100, Means: v1.0.0 version.

#### **Frequency Band:**

\*0x01: EU868

\*0x02: US915

\*0x03: IN865

\*0x04: AU915

\*0x05: KZ865

\*0x06: RU864

\*0x07: AS923

\*0x08: AS923-1

\*0x09: AS923-2

\*0x0a: AS923-3

**Sub-Band:** value 0x00 ~ 0x08(only for CN470, AU915,US915. Others are 0x00)

**BAT:** shows the battery voltage for PB01.

**Ex1:** 0x0C DE = 3294mV

## 2.4.2 Uplink FPORT=2, Real time sensor value

PB01 will send this uplink after Device Status uplink once join LoRaWAN network successfully. And it will periodically send this uplink. Default interval is 20 minutes and [can be changed](#).

Uplink uses FPORT=2 and every 20 minutes send one uplink by default.

Size(bytes)	2	1	1	2	2
Value	Battery	Sound_ACK	Alarm	Temperature	Humidity

&Sound\_key

Example in TTN.

Example Payload (FPort=2): **0C EA 03 01 01 11 02 A8**

### **Battery:**

Check the battery voltage.

- Ex1: 0x0CEA = 3306mV
- Ex2: 0x0D08 = 3336mV

### **Sound\_ACK & Sound\_key:**

Key sound and ACK sound are enabled by default.

- Example1: 0x03  
Sound\_ACK: (03>>1) & 0x01=1, OPEN.  
Sound\_key: 03 & 0x01=1, OPEN.
- Example2: 0x01  
Sound\_ACK: (01>>1) & 0x01=0, CLOSE.  
Sound\_key: 01 & 0x01=1, OPEN.

### **Alarm:**

Key alarm.

- Ex1: 0x01 & 0x01=1, TRUE.
- Ex2: 0x00 & 0x01=0, FALSE.

### **Temperature:**

- Example1: 0x0111/10=27.3°C
- Example2: (0xFF0D-65536)/10=-24.3°C

If payload is: FF0D : (FF0D & 8000 == 1) , temp = (FF0D - 65536)/100 =-24.3°C

(FF0D & 8000: Judge whether the highest bit is 1, when the highest bit is 1, it is negative)

### **Humidity:**

- Humidity: 0x02A8/10=68.0%

## **2.4.3 Uplink FPORT=3, Datalog sensor value**

PB01 stores sensor value and user can retrieve these history value via downlink command. The Datalog sensor value are sent via FPORT=3.

- Each data entry is 11 bytes, to save airtime and battery, PB01 will send max bytes according to the current DR and Frequency bands.

For example, in US915 band, the max payload for different DR is:

1. **DR0:** max is 11 bytes so one entry of data
2. **DR1:** max is 53 bytes so devices will upload 4 entries of data (total 44 bytes)
3. **DR2:** total payload includes 11 entries of data
4. **DR3:** total payload includes 22 entries of data.

**Notice: PB01 will save 178 set of history data, If device doesn't have any data in the polling time. Device will uplink 11 bytes of 0.**

See more info about the [Datalog feature](#).

#### 2.4.4 Decoder in TTN V3

In LoRaWAN protocol, the uplink payload is HEX format, user need to add a payload formatter/decoder in LoRaWAN Server to get human friendly string.

In TTN , add formatter as below:

# User Manual for LoRaWAN /NB -IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual

The screenshot shows the 'Payload formatters' tab selected in the top navigation bar. The left sidebar shows 'zero' as the application, with 'End devices' selected. The main content area shows a device named 'PB01' with an ID of 'eui-fa23455555555555'. The 'Payload formatters' tab is active. A red box highlights the 'Payload formatters' tab in the top navigation bar. Another red box highlights the 'Uplink' tab in the left sidebar. A red arrow points to the 'Custom Javascript formatter' dropdown in the 'Formatter type' section. A third red arrow points to the 'Save changes' button at the bottom.

**Formatter type\***  
Custom Javascript formatter

```

1  function datalog(i,bytes){
2    var aa= parseFloat(((bytes[0+i]<<24>>16 | bytes[1+i])/10).toFixed(1));
3    var bb= parseFloat(((bytes[2+i]<<24>>16 | bytes[3+i])/10).toFixed(1));
4    var cc= parseFloat(((bytes[4+i]<<24>>16 | bytes[5+i])/10).toFixed(1));
5    var dd= ((bytes[6+i]<<16)>>16) ? "True" : "False";
6    var ee= getMyDate((bytes[7+i]<<24 | bytes[8+i]<<16 | bytes[9+i]<<8 | bytes[10+i]).toString());
7    var string=[`'aa='+aa+',bb='+bb+',cc='+cc+',dd+',ee+'`];
8
9    return string;
10 }
11
12 function getzf(c_num){
13   if(parseInt(c_num) < 10)
14   | c_num = '0' + c_num;
15
16   return c_num;
17 }
18
19 function getMyDate(str){
20   var c_Date;
21   if(str > 9999999999)
22   | c_Date = new Date(parseInt(str));
23   else
24   | c_Date = new Date(parseInt(str) * 1000);
25

```

**Save changes**

Please check the decoder from this link: <https://github.com/dragino/dragino-end-node-decoder>

## 2.5 Show data on Datacake

Datacake IoT platform provides a human friendly interface to show the sensor data in charts, once we have sensor data in TTN V3, we can use Datacake to connect to TTN V3 and see the data in Datacake. Below are the steps:

**Step 1:** Be sure that your device is programmed and properly connected to the LoRaWAN network.

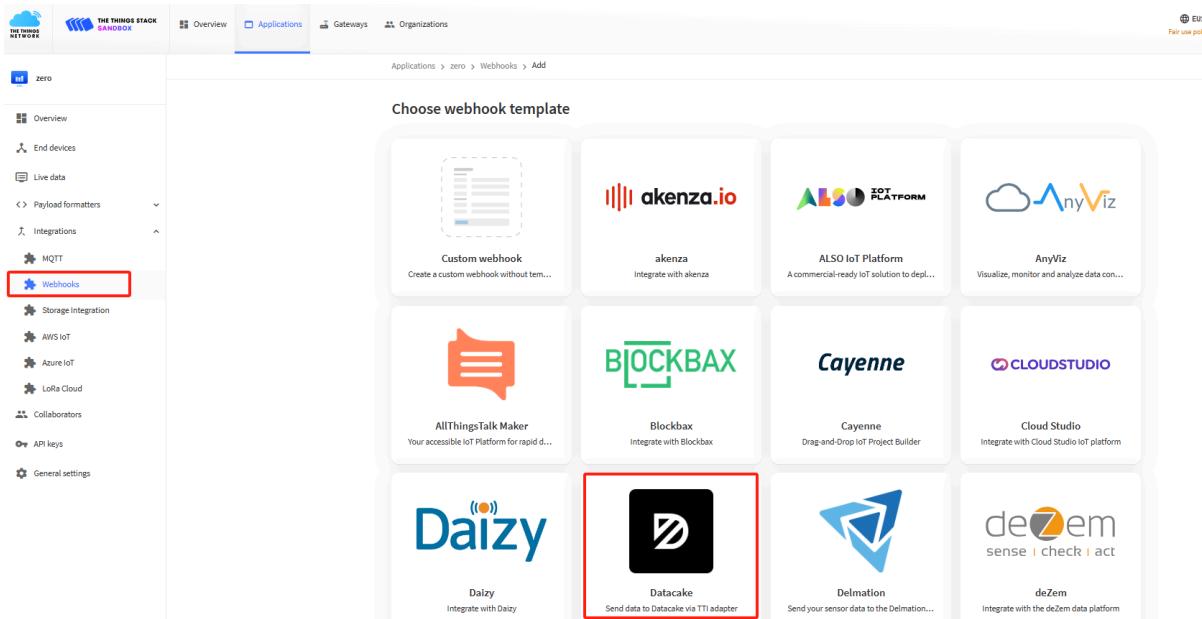
**Step 2:** Configure your Application to forward data to Datacake you will need to add integration. Go to TTN V3 Console --> Applications --> Integrations --> Add Integrations.

1. Add Datacake:

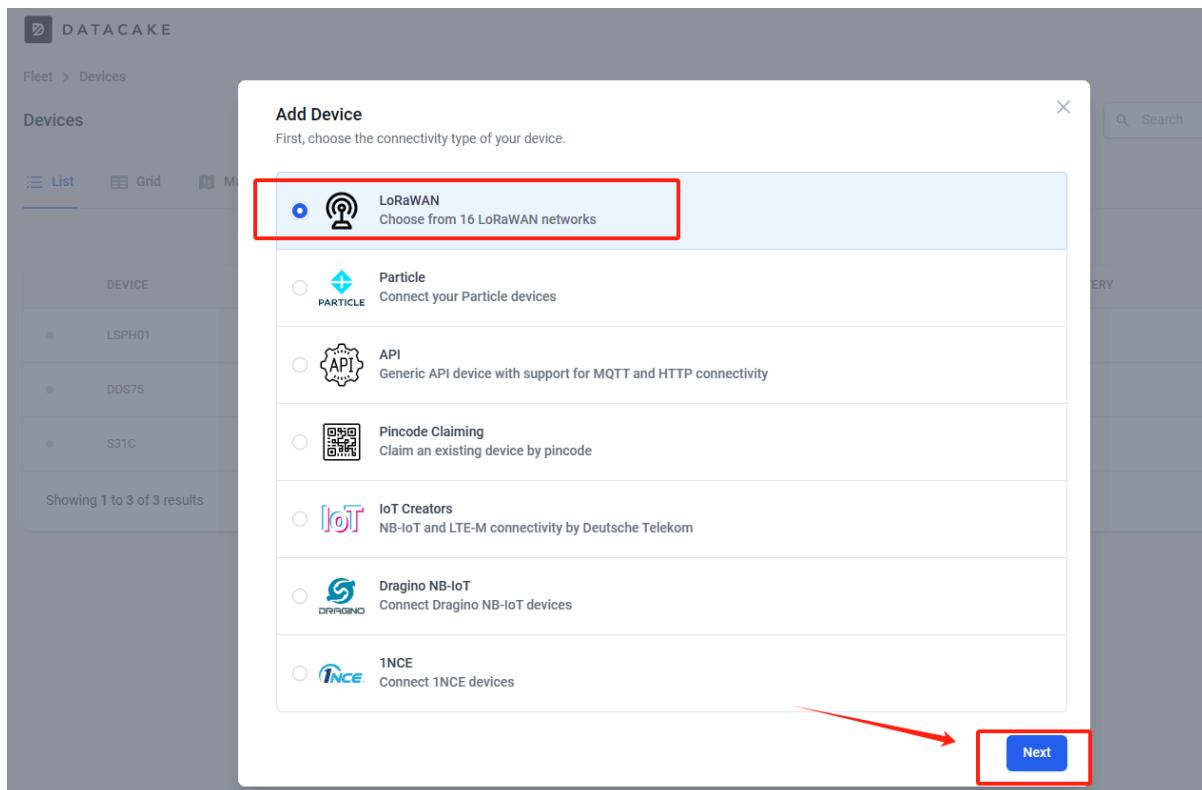
2. Select default key as Access Key:

3. In Datacake console (<https://datacake.co/>) , add PB01:

Please refer to the figure below.



Log in to DATACAKE, copy the API under the account.



### Add LoRaWAN Device

STEP 1 Product    STEP 2 Network Server    STEP 3 Devices    STEP 4 Plan

#### Datacake Product

You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

**New Product from template**  
 Create new product from a template

**Existing Product**  
 Add devices to an existing product

**New Product**  
 Create new empty product

**New Product**

If your device is not available as a template, you can start with an empty device. You will have to create the device definition (fields, dashboard) and provide the payload decoder in the device's configuration.

**Product Name**

pb01-l

**Back** **Next**

## Add LoRaWAN Device

STEP 1  
Product

STEP 2  
Network Server

STEP 3  
Devices

STEP 4  
Plan

### Network Server

Please choose the LoRaWAN Network Server that your devices are connected to.

The screenshot shows a list of network servers for LoRaWAN devices. Each entry includes a checkbox, the server logo, the name, a description, and two buttons for 'Uplinks' and 'Downlinks'.

- Datacake LNS** (AUTOMATIC SETUP): Start and scale easily with a managed LNS. 
- The Things Stack V3** / TTN V3 / Things Industries
- helium**: Use your own console
- LORIOT**
- ChirpStack**
- Actility**

At the bottom, it says "Showing 1 to 5 of 15 results" and has "Previous" and "Next" buttons. A red arrow points to the "Next" button, which is highlighted in blue.

STEP 1
STEP 2
STEP 3
STEP 4

Product
Network Server
Devices
Plan

## Add Devices

[Manual](#)   [Import from The Things Stack](#)

Please provide one or multiple LoRaWAN device EUIs along with the corresponding names they should have on Datacake.

Alternatively, you can choose to upload a CSV file that contains the DevEUI, device Name, location, and a set of tags. For more information on how to format the file, please refer to [our documentation](#).

Drag and drop a .csv file here or click to choose one

DEVEUI	NAME	LOCATION	TAGS
FA 23 45 55 55 55 55 51 <span style="color: green;">8 bytes</span>	PB01	<input type="text" value="Location"/>	<input type="button" value="Add tag"/>

+ Add another device
➡

Back
Next

## Payload Decoder

When your devices sends data, the payload will be passed to the payload decoder, alongside the event's name. The payload decoder then transforms it to measurements

### Product-wide setting

```

1 < var function datalog1(bytes){ 
2   var aee = parseFloat(((bytes[0]+1)<<24)>>16) | bytes[1+1]>>10).toFixed(1));
3   var bbe = parseFloat(((bytes[2]+1)<<24)>>16) | bytes[3+1]>>10).toFixed(1));
4   var cee = parseFloat(((bytes[4]+1)<<24)>>16) | bytes[5+1]>>10).toFixed(1));
5   var dde = parseFloat(((bytes[6]+1)<<24)>>16) | bytes[7+1]>>10).toFixed(1));
6   var eee = parseFloat(((bytes[8]+1)<<24)>>16) | bytes[9+1]>>10).toFixed(1));
7   var string = "["+aee+","+bbe+","+cee+","+dde+","+eee+"]"+',';
8 
9   return string;
10 }
11 
12 < function getT(c_num){ 
13   if(parseInt(c_num) < 10)
14     c_num = '0' + c_num;
15 
16   return c_num;
17 }
18 
19 < function getMyDate(str){ 
20   var c_Date;
21   if(str == "00000000000000000000000000000000")
22     c_Date = new Date(parseInt(str));
23   else
24     c_Date = new Date(parseInt(str + ' 1000'));
25 
26   var c_Year = c_Date.getFullYear(),
27   c_Month = c_Date.getMonth() + 1,
28   c_Day = c_Date.getDate(),
29   c_Hour = c_Date.getHours(),
30   c_Min = c_Date.getMinutes(),
31   c_Sec = c_Date.getSeconds();
32   var c_Time = c_Year + ":" + getzf(c_Month) + ":" + getzf(c_Day) + ":" + getzf(c_Hour) + ":" + getzf(c_Min) + ":" + getzf(c_Sec);
33 
34   return c_Time;
35 }
36 
37

```

## Upload

## Port

Visual widgets please read the DATACAKE documentation.

## 2.6 Datalog Feature

When user want to retrieve sensor value, he can send a poll command from the IoT platform to ask sensor to send value in the required time slot.

### 2.6.1 UnixTimeStamp

UnixTimeStamp shows the sampling time of uplink payload. format base on

Size (bytes)	4	1
DeviceTimeAns Payload	32-bit unsigned integer : Seconds since epoch*	8bits unsigned integer: fractional- second in $1/2^8$ second steps

Figure 10 : DeviceTimeAns payload format

User can get this time from link: <https://www.epochconverter.com/> :

For example: if the Unix Timestamp we got is hex 0x60137afd, we can convert it to Decimal: 1611889405. and then convert to the time: 2021 – Jan -- 29 Friday 03:03:25 (GMT)

The screenshot shows two websites side-by-side. On the left, EpochConverter.com displays the current Unix epoch time as 1611889405. On the right, Code Beautify's Decimal to Hex converter shows the same decimal value being converted to the hex value 60137afd. A red arrow points from the decimal value on EpochConverter.com to the input field on the Code Beautify converter, indicating the flow of data between the two tools.

## 2.6.2 Poll sensor value

User can poll sensor value based on timestamps from the server. Below is the downlink command.

Timestamp start and Timestamp end use Unix TimeStamp format as mentioned above. Devices will reply with all data log during this time period, use the uplink interval.

For example, downlink command 31 5FC5F350 5FC6 0160 05:  
31 5FC5F350 5FC6 0160 05:

Is to check 2020/12/1 07:40:00 to 2020/12/1 08:40:00's data

Uplink Internal =5s, means PB01 will send one packet every 5s. range 5~255s.

## 2.6.3 Datalog Uplink payload

See [Uplink FPORT=3, Datalog sensor value](#)

## 2.7 Button

- ACT button

Long press this button PB01 will reset and join network again.



- Alarm button

Press the button PB01 will immediately uplink data, and alarm is "TRUE".



## 2.8 LED Indicator

The PB01 has a triple color LED which for easy showing different stage.

Hold the ACT green light to rest, then the green flashing node restarts, the blue flashing once upon request for network access, and the green constant light for 5 seconds after successful network access

### In a normal working state:

- When the node is restarted, hold the ACT **GREEN** lights up , then the **GREEN** flashing node restarts.The **BLUE** flashing once upon request for network access, and the **GREEN** constant light for 5 seconds after successful network access.
- During OTAA Join:
  - For each Join Request uplink:** the **GREEN LED** will blink once.
  - Once Join Successful:** the **GREEN LED** will be solid on for 5 seconds.
- After joined, for each uplink, the **BLUE LED** or **GREEN LED** will blink once.
- Press the alarm button, The **RED** flashes until the node receives the ACK from the platform and the **BLUE** light stays 5s.

## 2.9 Buzzer

The PB01 has **button sound** and **ACK sound** and users can turn on or off both sounds by using [AT+SOUND](#).

- Button sound** is the music produced by the node after the alarm button is pressed.

Users can use[AT+OPTION](#) to set different button sounds.

- ACK sound** is the notification tone that the node receives ACK.

## 3. Configure PB01 via AT command or LoRaWAN downlink

Users can configure PB01 via AT Command or LoRaWAN Downlink.

- AT Command Connection: See [FAQ](#).
- LoRaWAN Downlink instruction for different platforms: [IoT LoRaWAN Server](#)

There are two kinds of commands to configure PB01, they are:

- General Commands:**

These commands are to configure:

- General system settings like: uplink interval.

- LoRaWAN protocol & radio-related commands.

They are the same for all Dragino Devices which supports DLWS-005 LoRaWAN Stack(Note\*\*). These commands can be found on the wiki: [End Device Downlink Command](#)

- **Commands special design for PB01**

These commands are only valid for PB01, as below:

### 3.1 Downlink Command Set

Command Example	Function	Response	Downlink
AT+TDC=?		1200000 OK	Default 1200000(ms)
	View current TDC time		
AT+TDC=300000	Set TDC time	OK	0X0100012C: 01: fixed command 00012C: 0X00012C=
			300(seconds)
ATZ	Reset node		0x04FF
AT+FDR	Restore factory settings		0X04FE
AT+CFM=?	View the current confirmation mode status	0,7,0 OK	Default 0,7,0
AT+CFM=1,7,1	Confirmed uplink mode, the maximum number of retries is seven, and uplink fcnt increase by 1 for each retry	OK	05010701 05: fixed command 01:confirmed uplink 07: retry 7 times 01: fcnt count plus 1
AT+NJM=?	Check the current network connection method	1 OK	Default 1
AT+NJM=0	Change the network connection method to ABP	Attention:Take effect after ATZ OK	0X2000: ABP 0x2001: OTAA 20: fixed command
AT+RPL=?	View current RPL settings	0 OK	Default 0
AT+RPL=1	set RPL=1	OK	0x2101: 21: fixed command 01: for details, check wiki
AT+ADR=?	View current ADR status	1 OK	Default 0
AT+ADR=0	Set the ADR state to off	OK	0x2200: close 0x2201: open 22: fixed command
AT+DR=?	View the current DR settings	OK	
AT+DR=1	set DR to 1 It takes effect only when ADR=0	OK	0X22000101: 00: ADR=0 01: DR=1 01: TXP=1 22: fixed command
AT+TXP=?	View the current TXP	OK	

AT+TXP=1	set TXP to 1 It takes effect only when ADR=0	OK	0X22000101: 00: ADR=0 01: DR=1 01: TXP=1 22: fixed command
AT+RJTDC=10	Set RJTDC time interval	OK	0X26000A: 26: fixed command 000A: 0X000A=10(min) for details, check wiki
	Retrieve stored data for a specified period of time		0X3161DE7C7061DE8A800A: 31: fixed command 61DE7C70:0X61DE7C70=2022/1/12 15:00 61DE8A80:0X61DE8A80=2022/1/12 16:00 0A: 0X0A=10(second) View details 2.6.2
AT+DDETECT=?	View the current DDETECT setting status and time	1,1440,2880 OK	Default 1,1440,2880(min)
AT+DDETECT=1,1440,2880	Set DDETECT setting status and time <i>(When the node does not receive the downlink packet within the set time, it will re-enter the network)</i>	OK	0X320005A0: close 0X320105A0: open 32: fixed command 05A0: 0X05A0=1440(min)

## 3.2 Set Password

Feature: Set device password, max 9 digits.

### AT Command: AT+PWDWORD

Command Example	Function	Response
AT+PWDWORD=?	Show password	123456 OK
AT+PWDWORD=999999	Set password	OK

### Downlink Command:

No downlink command for this feature.

## 3.3 Set button sound and ACK sound

Feature: Turn on/off button sound and ACK alarm.

### AT Command: AT+SOUND

Command Example	Function	Response
AT+SOUND=?	Get the current status of button sound and ACK sound	1,1 OK
AT+SOUND=0,1	Turn off the button sound and turn on ACK sound	OK

### Downlink Command: 0xA1

Format: Command Code (0xA1) followed by 2 bytes mode value.

The first byte after 0XA1 sets the button sound, and the second byte after 0XA1 sets the ACK sound. (0: off, 1: on)

- **Example:** Downlink Payload: A10001 // Set AT+SOUND=0,1 Turn off the button sound and turn on ACK sound.

### 3.4 Set buzzer music type(0~4)

Feature: Set different alarm key response sounds. There are five different types of button music.

#### AT Command: AT+OPTION

Command Example	Function	Response
AT+OPTION=?	Get the buzzer music type	3 OK
AT+OPTION=1	Set the buzzer music to type 1	OK

#### Downlink Command: 0xA3

Format: Command Code (0xA3) followed by 1 byte mode value.

- **Example:** Downlink Payload: A300 // Set AT+OPTION=0 Set the buzzer music to type 0.

### 3.5 Set Valid Push Time

Feature: Set the holding time for pressing the alarm button to avoid miscontact. Values range from **0 ~1000ms**.

#### AT Command: AT+STIME

Command Example	Function	Response
AT+STIME=?	Get the button sound time	0 OK
AT+STIME=1000	Set the button sound time to 1000ms	OK

#### Downlink Command: 0xA2

Format: Command Code (0xA2) followed by 2 bytes mode value.

- **Example:** Downlink Payload: A203E8 // Set AT+STIME=1000

**Explain:** Hold the alarm button for 10 seconds before the node will send the alarm packet.

## 4. Battery & How to replace

### 4.1 Battery Type and replace

PB01 uses 2 x AAA LR03(1.5v) batteries. If the batteries running low (shows 2.1v in the platform). Users can buy generic AAA battery and replace it.

#### Note:

1. The PB01 doesn't have any screw, users can use nail to open it by the middle.



2. Make sure the direction is correct when install the AAA batteries.

## 4.2 Power Consumption Analyze

Dragino battery powered product are all runs in Low Power mode. We have an update battery calculator which base on the measurement of the real device. User can use this calculator to check the battery life and calculate the battery life if want to use different transmit interval.

Instruction to use as below:

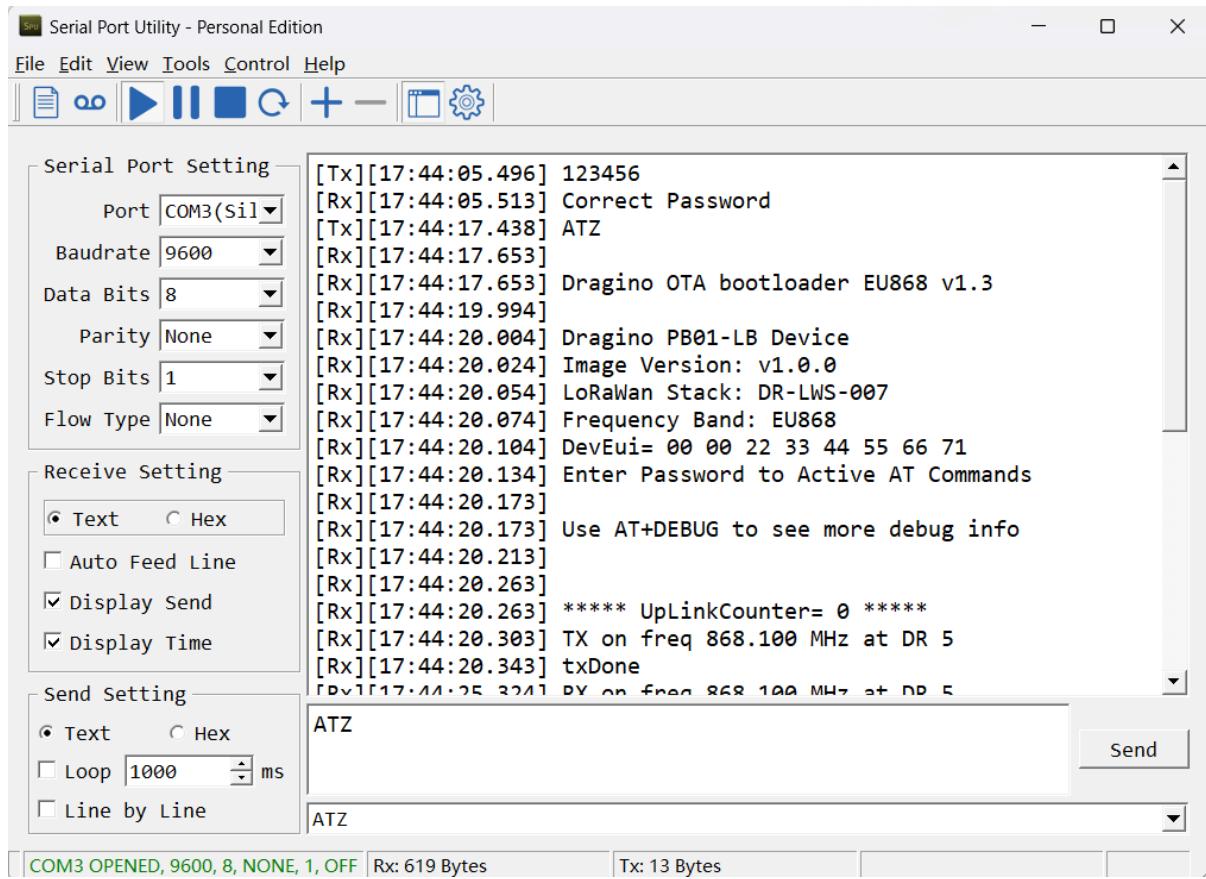
**Step 1:** Downlink the up-to-date DRAGINO\_Battery\_Life\_Prediction\_Table.xlsx from:

[battery calculator](#)

**Step 2:** Open it and choose

- Product Model
- Uplink Interval
- Working Mode

And the Life expectation in difference case will be shown on the right.



## 6.2 AT Command and Downlink

Sending ATZ will reboot the node

Sending AT+FDR will restore the node to factory settings

Get the node's AT command setting by sending AT+CFG

Example:

AT+DEUI=FA 23 45 55 55 55 55 51

AT+APPEUI=FF AA 23 45 42 42 41 11

AT+APPKEY=AC D7 35 81 63 3C B6 05 F5 69 44 99 C1 12 BA 95

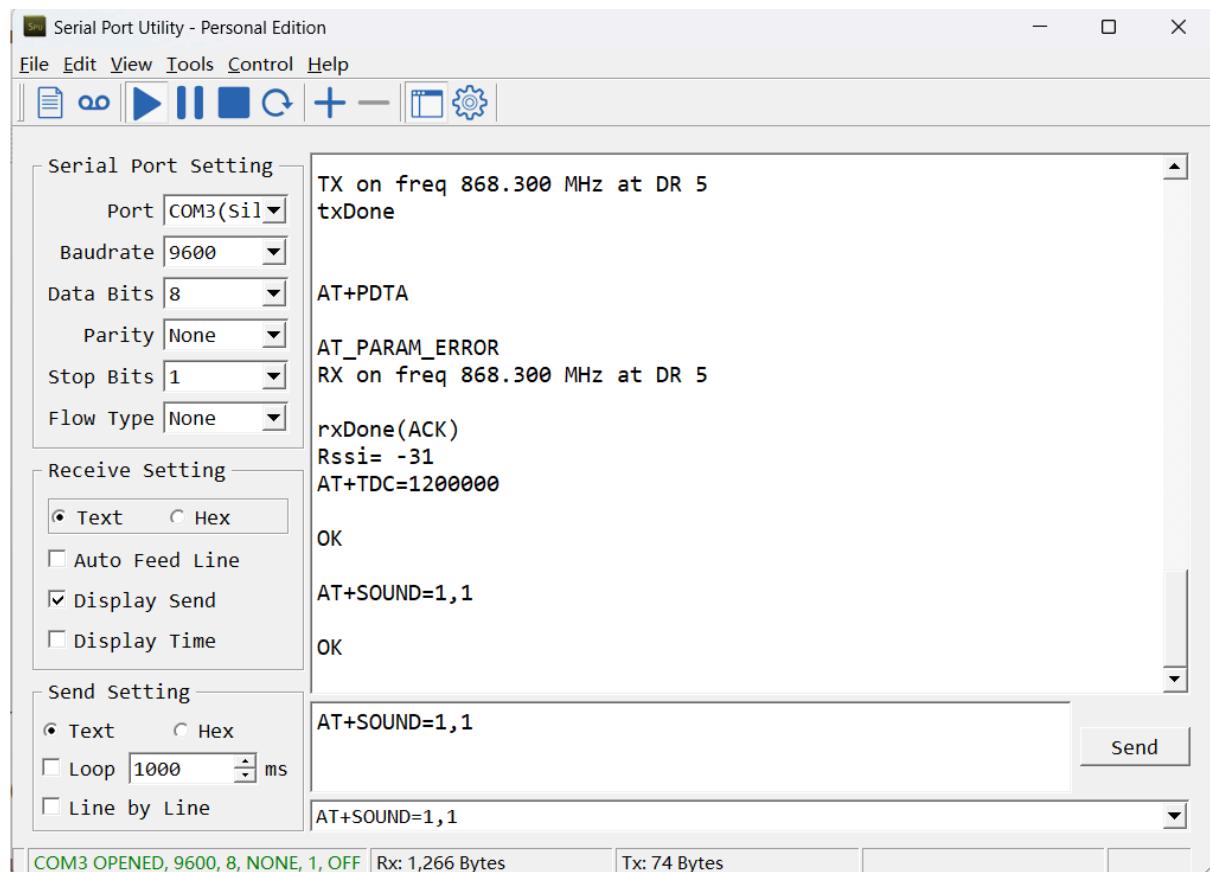
AT+DADDR=FFFFFF

AT+APPSKEY=FF FF  
AT+NWKSKEY=FF FF  
AT+ADR=1  
AT+TXP=7  
AT+DR=5  
AT+DCS=0  
AT+PNM=1  
AT+RX2FQ=869525000  
AT+RX2DR=0  
AT+RX1DL=5000  
AT+RX2DL=6000  
AT+JN1DL=5000  
AT+JN2DL=6000  
AT+NJM=1  
AT+NWKID=00 00 00 13  
AT+FCU=61  
AT+FCD=11  
AT+CLASS=A  
AT+NJS=1  
AT+RECVB=0:  
AT+RECV=  
AT+VER=EU868 v1.0.0  
AT+CFM=0,7,0  
AT+SNR=0  
AT+RSSI=0  
AT+TDC=1200000  
AT+PORT=2  
AT+PWD=123456  
AT+CHS=0  
AT+RX1WTO=24  
AT+RX2WTO=6  
AT+DECRYPT=0  
AT+RJTDC=20  
AT+RPL=0  
AT+TIMESTAMP=systime= 2024/5/11 01:10:58 (1715389858)  
AT+LEAPSEC=18

---

```
AT+SYNCFMOD=1
AT+SYNCTDC=10
AT+SLEEP=0
AT+ATDC=1
AT+UUID=003C0C53013259E0
AT+DDETECT=1,1440,2880
AT+SETMAXNBTRANS=1,0
AT+DISFCNTCHECK=0
AT+DISMACANS=0
AT+PNACKMD=0
AT+SOUND=0,0
AT+STIME=0
AT+OPTION=3
```

Example:



## 6.3 How to upgrade the firmware?

PB01 requires a program converter to upload images to PB01, which is used to upload image to PB01 for:

- Support new features
- For bug fix
- Change LoRaWAN bands.

PB01 internal program is divided into bootloader and work program, shipping is included bootloader, the user can choose to directly update the work program.

If the bootloader is erased for some reason, users will need to download the boot program and the work program.

### 6.3.1 Update firmware (Assume device have bootloader)

**Step 1:** Connect UART as per FAQ 6.1

**Step 2:** Update follow [Instruction for update via DraginoSensorManagerUtility.exe](#).

### 6.3.2 Update firmware (Assume device doesn't have bootloader)

Download both the boot program and the worker program . After update , device will have bootloader so can use above 6.3.1 method to update woke program.

**Step 1:** Install [TremoProgrammer](#) first.

**Step 2:** Hardware Connection

Connect PC and PB01 via USB-TTL adapter .

**Note:** To download firmware in this way, you need to pull the boot pin(Program Converter D- pin) high to enter the burn mode. After burning, disconnect the boot pin of the node and the 3V3 pin of the USB-TTL adapter, and reset the node to exit the burning mode.

**Connection:**

- USB-TTL GND <-> Program Converter GND pin
- USB-TTL RXD <-> Program Converter D+ pin
- USB-TTL TXD <-> Program Converter A11 pin
- USB-TTL 3V3 <-> Program Converter D- pin

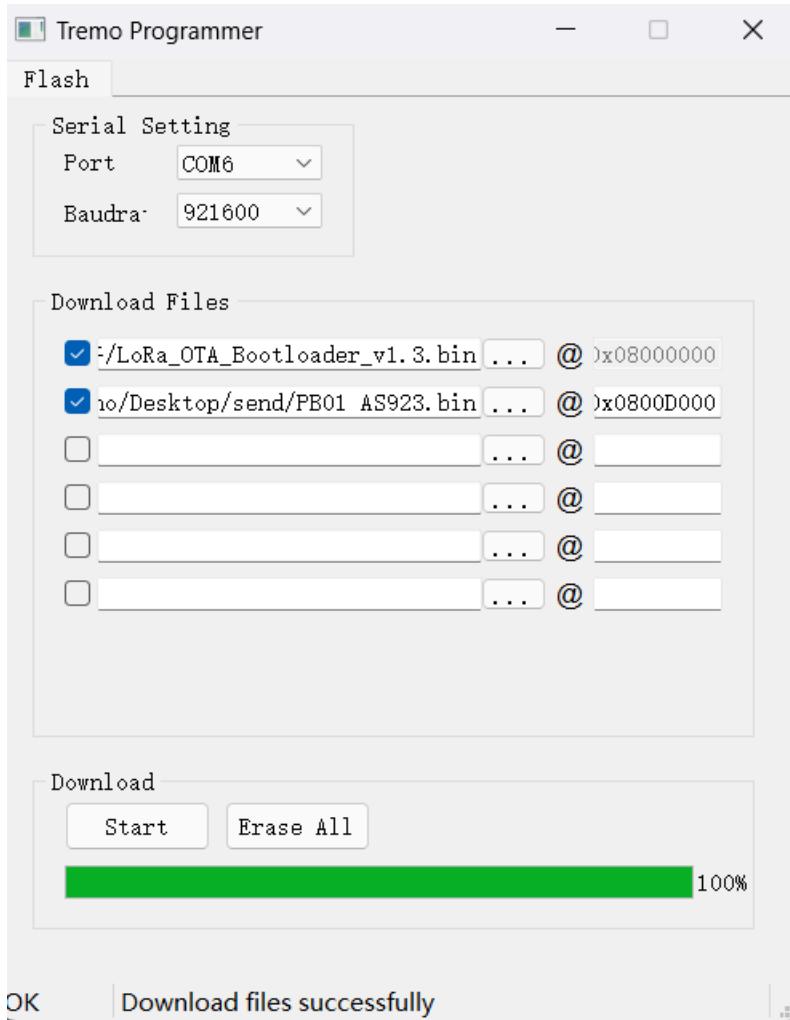
**Step 3:** Select the device port to be connected, baud rate and bin file to be downloaded.



Users need to reset the node to start downloading the program.

1. Reinstall the battery to reset the node
2. Hold down the ACT button to reset the node (see [2.7](#) ).

When this interface appears, it indicates that the download has been completed.



Finally, Disconnect Program Converter D- pin, reset the node again , and the node exits burning mode.

## 6.4 How to change the LoRa Frequency Bands/Region?

User can follow the introduction for [how to upgrade image](#). When download the images, choose the required image file for download.

## 6.5 Why i see different working temperature for the device?

The working temperature range of device depends on the battery user choose.

- Normal AAA Battery can support -10 ~ 50°C working range.
- Special AAA battery can support -40 ~ 60 °C working range. For example: [Energizer L92](#)

## 7. Order Info

### 7.1 Main Device

Part Number: [PB01-LW-XX](#) (white button) / [PB01-LR-XX](#)(Red Button)

**XX** : The default frequency band

- **AS923**: LoRaWAN AS923 band
- **AU915**: LoRaWAN AU915 band
- **EU433**: LoRaWAN EU433 band
- **EU868**: LoRaWAN EU868 band
- **KR920**: LoRaWAN KR920 band
- **US915**: LoRaWAN US915 band
- **IN865**: LoRaWAN IN865 band
- **CN470**: LoRaWAN CN470 band

## 7. Packing Info

**Package Includes:**

- PB01 LoRaWAN Push Button x 1

## 8. Support

- Support is provided Monday to Friday, from 09:00 to 18:00 GMT+8. Due to different timezones we cannot offer live support. However, your questions will be answered as soon as possible in the before-mentioned schedule.
- Provide as much information as possible regarding your enquiry (product models, accurately describe your problem and steps to replicate it etc) and send a mail to [support@dragino.com](mailto:support@dragino.com).

## 9. Reference material

- [Datasheet, photos, decoder, firmware](#)

## 10. FCC Warning

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference;
- (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

---

▲ 14:03:47	1	2	payload: CB0B 0B 64 02 72 06 0B 06 7F FF
			Ext: 0x06, ADC Sensor and cable connects ok. The most left bit stands for Cable Connnection: 0: Connects ok. 1: Cable doesn't connect
			Voltage Value : 0x0B06 = 2.822v

▲ 11:36:51	0	2	payload: CBD8 0B 68 02 D7 86 FF FF 7F FF
------------	---	---	------------------------------------------

Applications > lgt92test > Webhooks > Add > Datacake

## Add custom webhook

### Template information



#### Datacake

Send data to Datacake via TTI adapter

[About Datacake](#) | [Documentation](#)

### Template settings

Webhook ID \*

my-new-datacake-webhook

Token \*

Datacake API Token

**Create datacake webhook**

Complex configuration and setup.

- Dragino LSE01**  
Dragino
- Dragino LT-22222-L**  
Dragino
- Dragino LWL01**  
Dragino
- ESP32-Paxcounter**  
cyberman54
- Elsys ELT-2**  
Elsys.se

Showing 26 to 30 of 79 results

DeviceTimeAns Payload	Size (bytes)	4	1
		32-bit unsigned integer : Seconds since epoch*	8bits unsigned integer: fractional- second in $1/2^8$ second steps

Figure 10 : DeviceTimeAns payload format

**EpochConverter**

Epoch & Unix Timestamp Conversion To

The current Unix epoch time is **1611889418**

Convert epoch to human-readable date and vice versa

1611889090 | **Timestamp to Human date** | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:

**GMT:** 2021-10-29 Friday 02:58:10

**Your time zone:** 2021-10-29 Friday 10:58:10 GMT+08:00

**Relative:** 3 minutes ago

Mon Day Yr Hr Min Sec | **Human date to Time**

▲ 09:57:27 102 2 payload: 7F FF 08 98 01 46 41 60 06 5F 97 7F FF 08 8E 01 4B 41 60 06 60 09 7F FF 08 75 01 51 41 60 06

**Uplink**

**Payload**

7F FF 08 98 01 46 41 60 06 5F 97 7F FF 08 8E 01 4B 41 60 06 60 09 7F FF 08 85 01 4E 41 60 06 60 66 7F FF 08 75 01 51 41 60 06

**Fields**

no fields

**Metadata**

```
{
  "time": "2021-01-20T01:57:27.690185935Z",
  "frequency": 904.5, ←
  "modulation": "LORA",
  "data_rate": "SF7BW125", ←
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-a840411cfe60415c",
      "timestamp": 3270993355,
      "time": "2021-01-20T01:57:27.544057Z",
      "channel": 3,
      "rssi": -55,
      "snr": 10
    }
  ]
}
```

**Code Beautify**

All Numbers Converter

Numbers to Words Converter

Decimal to Binary Converter

**Decimal to Hex Converter**

Decimal to Octal Converter

Binary to Decimal Converter

Binary to Hex Converter

Binary to Octal Converter

Binary to Text Converter

Text to Binary Converter

Hex to Decimal Converter

Hex to Binary Converter

Hex to Octal Converter

Octal to Decimal Converter

JSON Formatter

Decimal to Hex

Enter the Decimal number to decode

1611889405

Sample

Auto | **Convert** | [File...](#) | [Go](#)

The number in hex (base 16) representation:

60137afd

Applications > lht111 > End devices > eui-a84041ffff1234dd

---



**eui-a84041ffff1234dd**

ID: eui-a84041ffff1234dd

↑ 156 ↓ 156 • Last activity 13 days ago ?

Overview

Live data

**Messaging**

Location

Payload formatters

General settings

Uplink

**Downlink**

---

### Schedule downlink

#### Insert Mode

Replace downlink queue  
 Push to downlink queue (append)

#### FPort \*

1

#### Payload type

Bytes  JSON

#### Payload

The desired payload bytes of the downlink message

Confirmed downlink

**Schedule downlink**

**My Devices**

**lht65**

**DEVICE DETAILS**

Name	lht65
ID	155a3ad7-62be-4ced-85a8-a2852c299000
Device EUI	AB4041000181A756
App EUI	25B975245F871C56
App Key	*****
Activation Method	OTAA
Profile	None

**PACKETS TRANSFERRED** DC USED

All Time	345
Last 7 Days	0
Last 30 Days	262
Last 24 Hours	0

**Delete Device**

**Unlink** **Link** **+**

**Add Label**

**0 LABELS ATTACHED**

**FLOWs**

No flows exist for this device

**aws** **seve** **Search for services, features, blogs, documents, and more** [Alt+S]

**Device traffic**

The gateway to which it was last connected DevEUI RSSI (dBm) SNR (dB) frequency Data rate

a840411e96744159	003586ec8db99eb	-79	13.25	916800000	3
------------------	-----------------	-----	-------	-----------	---

**Configuration file**

Device profiles Service configuration file

8287665f-9338-415c-ad8c-57f3f2d71ee2	fdb90f1-f524-4f14-ac20-2a22a17e0933
--------------------------------------	-------------------------------------

**Downlink message queue (0) [information](#)**

**label**

**key** **value**

No labels

You don't have any tags attached to this resource.

[Applications / kazk / Devices / wsc1](#)

DETAILS      CONFIGURATION      KEYS

---

Details

[Applications / kazk / Devices / wsc1](#)

DETAILS      CONFIGURATION      KEYS (OTAA)

---

Details

Mouse drop  
down

Name

Enqueue downlink payload

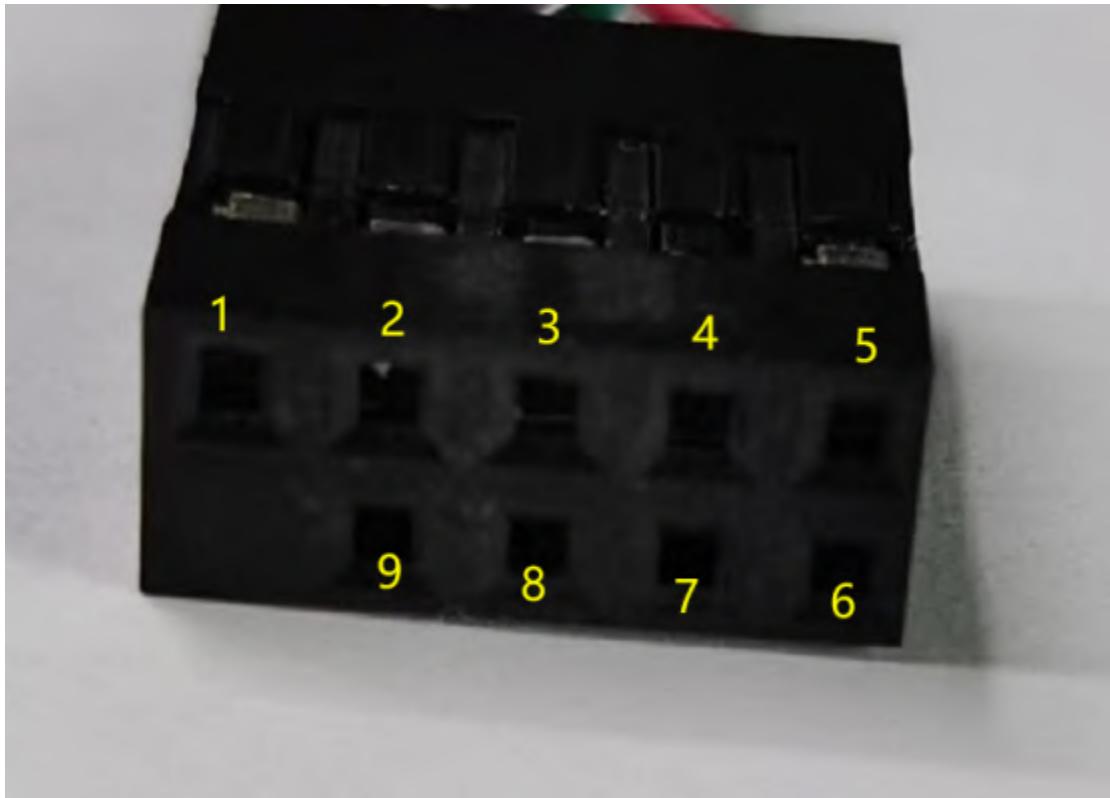
Port  Please note that the fPort value must be > 0.

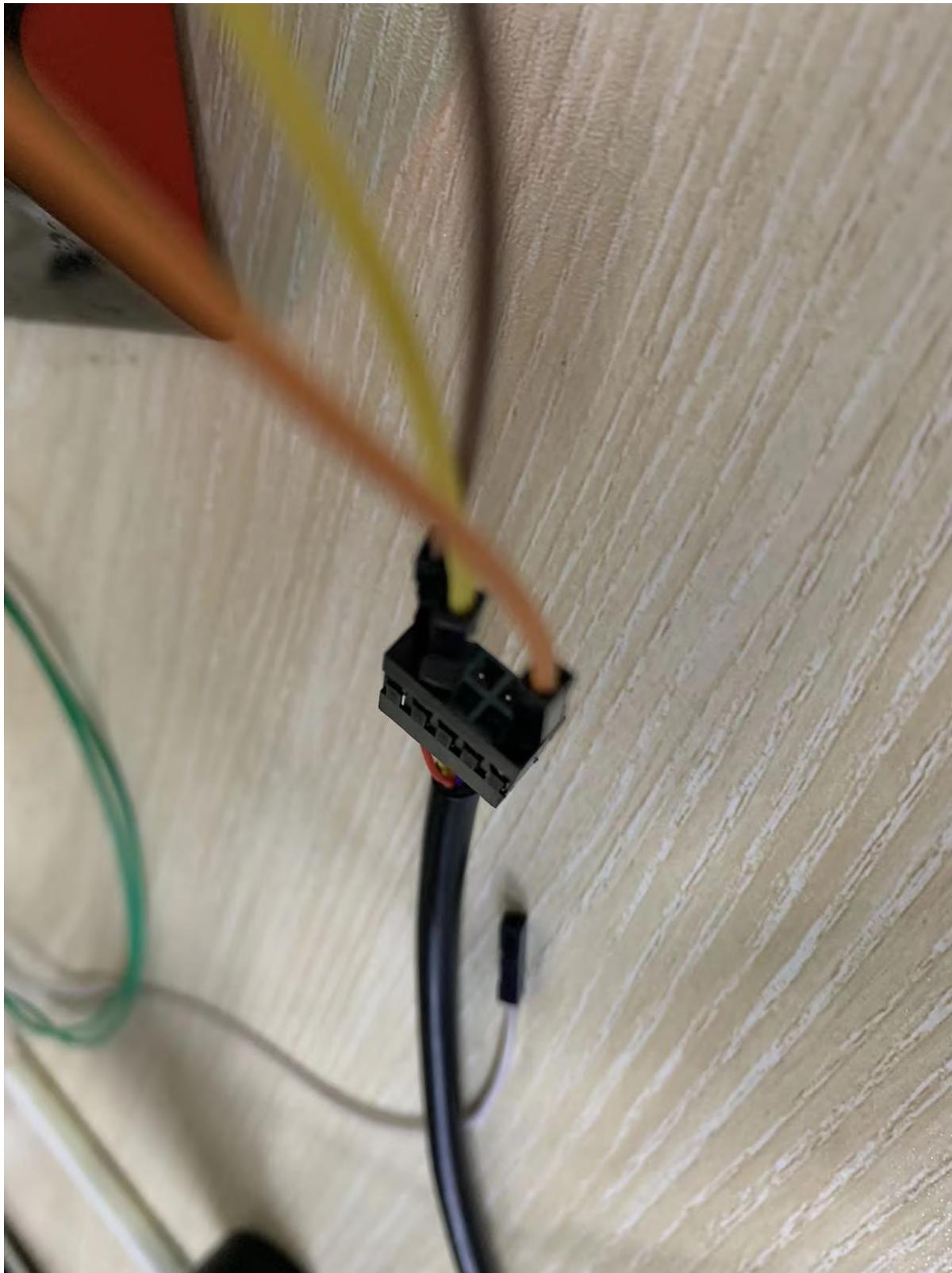
Confirmed downlink

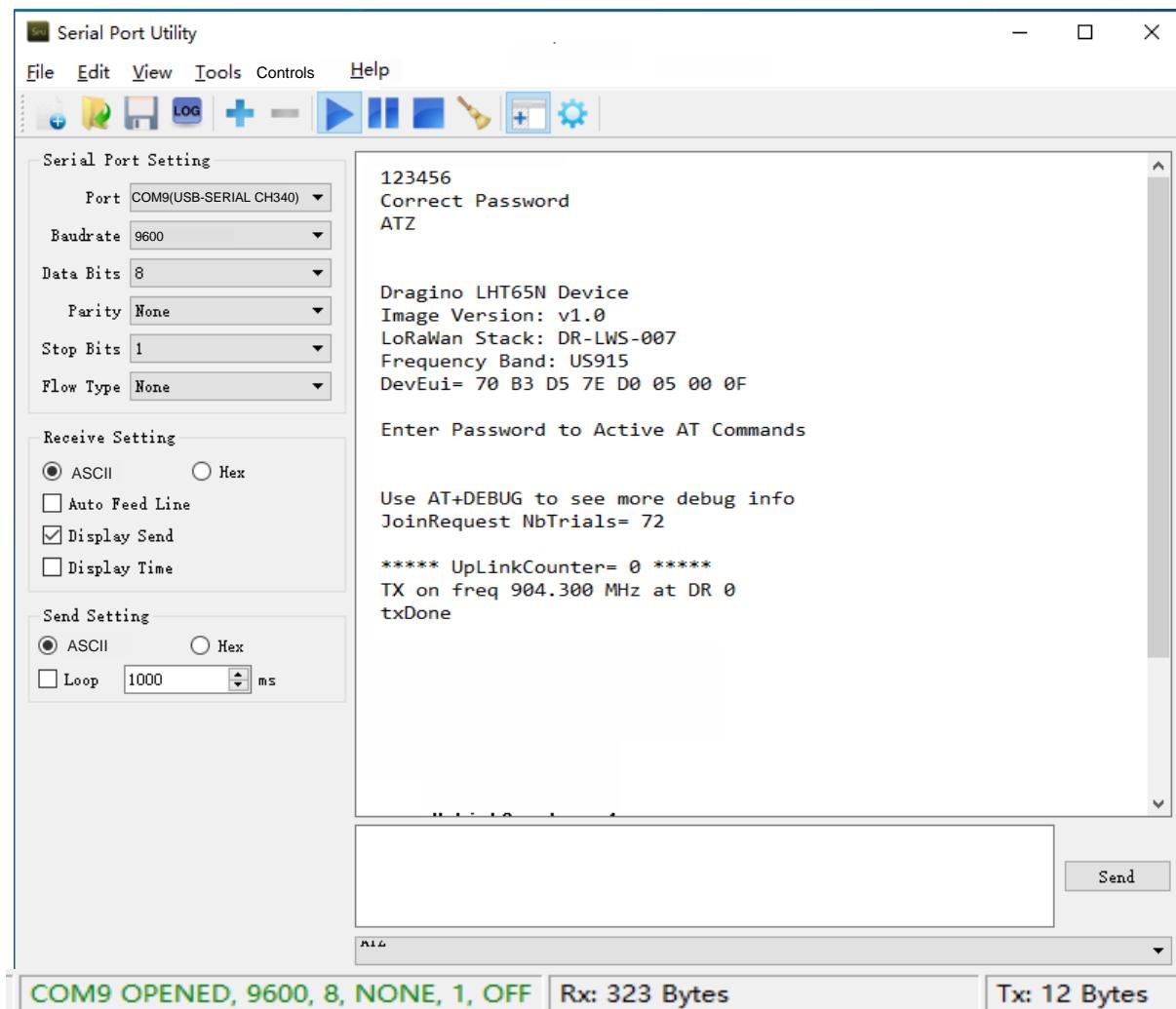
BASE64 ENCODED  JSON OBJECT

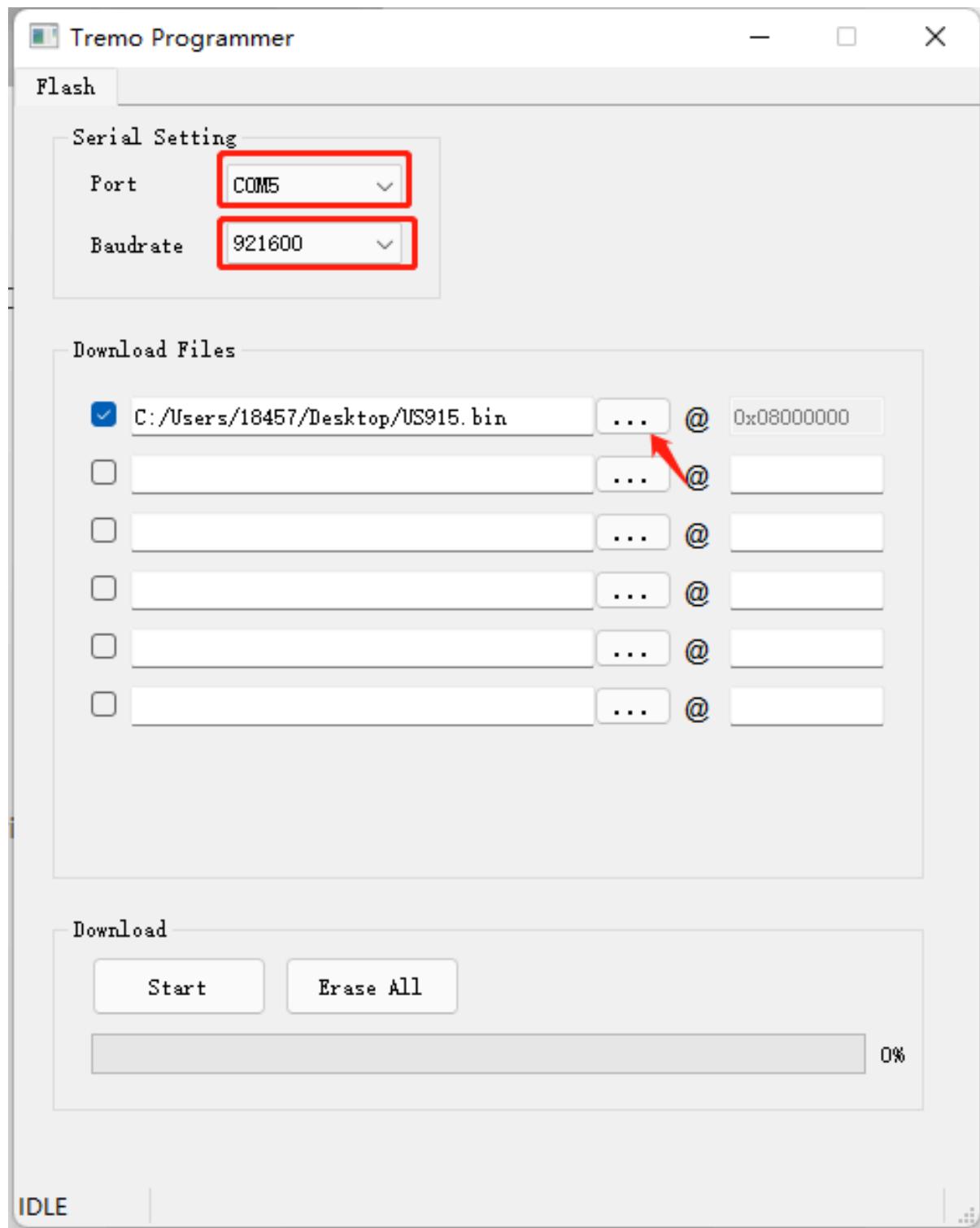
Base64 encoded string  \*

[ENQUEUE PAYLOAD](#)









The screenshot shows the configuration of a custom webhook in The Things Stack Community Edition. The left sidebar shows the navigation menu with 'Webhooks' selected. The main panel shows the 'Add custom webhook' configuration for the 'Datacake' adapter. The 'Webhook ID' field contains 'lht52testlu01' and the 'Token' field contains a masked value. A red arrow points to the 'Create datacake webhook' button. Below this, a screenshot of the Datacake interface shows the 'Devices' list with one device entry and a red box around the '+ Add Device' button.

### Add Device

 LoRaWAN  PARTICLE  API  D Zero  D Zero LTE  PINCODE

**STEP 1**  
Product      **STEP 2**  
Network Server      **STEP 3**  
Devices      **STEP 4**  
Plan

#### Datacake Product

You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

**New Product from template**  
Create new product from a template

**Existing Product**  
Add devices to an existing product

**New Product**  
Create new empty product

**Product Name**  
Iht52test01

**Next**

## Add Device



LoRaWAN



PARTICLE



API



D Zero



D Zero LTE



PINCODE

STEP 1

Product

STEP 2

Network Server

STEP 3

Devices

STEP 4

Plan

## Network Server

Please choose the LoRaWAN Network Server that your devices are connected to.

	<p>The Things Stack V3 TTN V3 / Things Industries</p>	<a href="#">Uplinks</a>	<a href="#">Downlinks</a>
<input type="radio"/>	 The Things Network V2 The old Things Network	<a href="#">Uplinks</a>	<a href="#">Downlinks</a>
<input type="radio"/>	 helium Helium	<a href="#">Uplinks</a>	<a href="#">Downlinks</a>
<input type="radio"/>	 LORIOT	<a href="#">Uplinks</a>	<a href="#">Downlinks</a>
<input type="radio"/>	 kerlink Kerlink Wanesy	<a href="#">Uplinks</a>	

Showing 1 to 5 of 8 results

[Previous](#) [Next](#)

Back

Next



## Add Device



LoRaWAN



PARTICLE



API



D Zero



D Zero LTE



PINCODE

**STEP 1**  
Product

**STEP 2**  
Network Server

**STEP 3**  
Devices

**STEP 4**  
Plan

## Add Devices

Enter one or more LoRaWAN Device EUIs and the names they will have on Datacake.

**New:** You can now upload a CSV file with either one column (just the device's DevEUI) or two columns (DevEUI and Name), which will populate the form below.

ⓘ Drag and drop a .csv file here or click to choose one

DEVEUI

NAME

ⓘ 25 32 12 45 65 26 12 38 8 bytes

ⓘ LHT52

+ Add another device

Back

Next

**k kazike lu 459550896@qq.com**

**DATA CAKE**

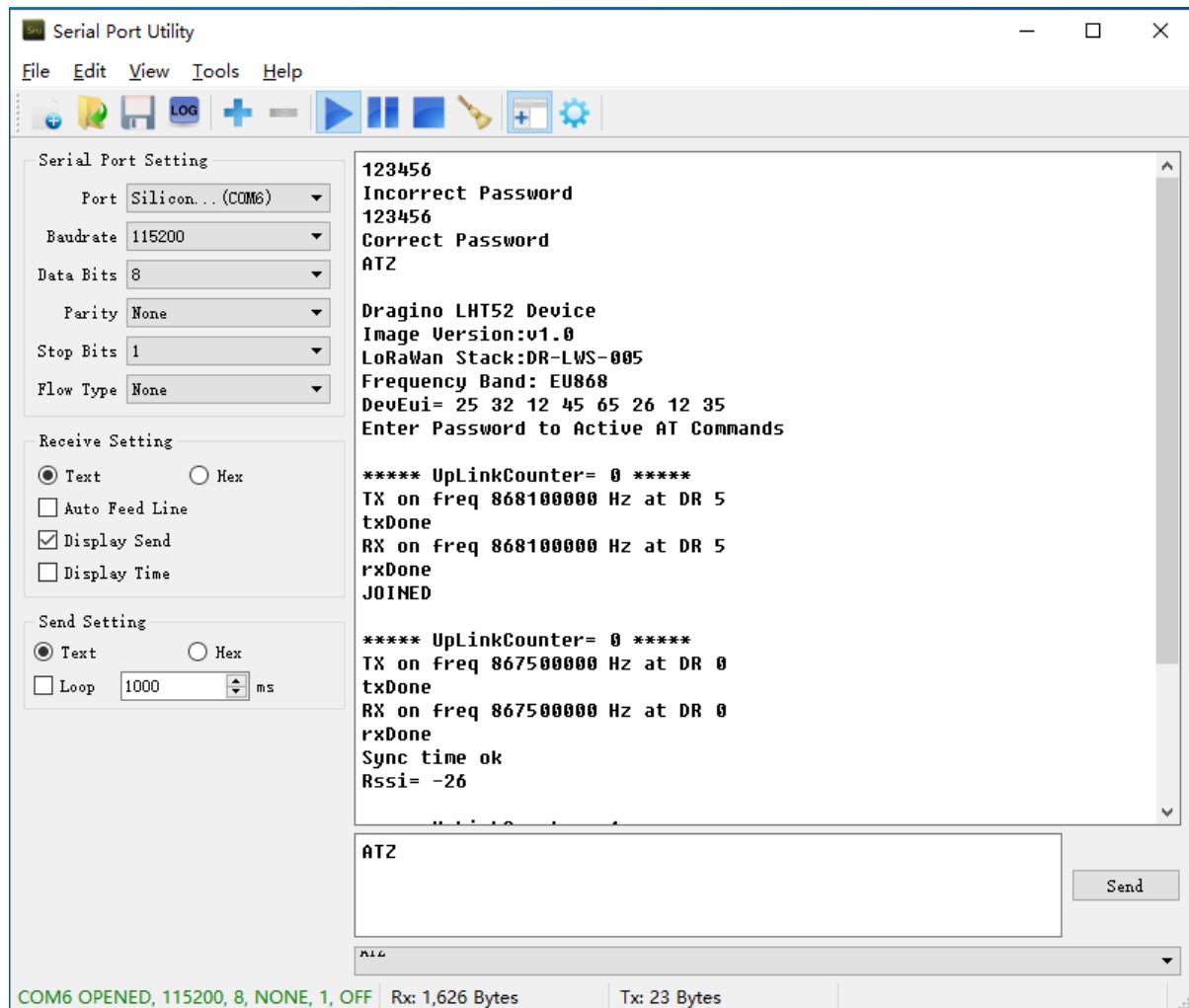
Fleet > Devices

**Devices**

DEVICE	LOCATION
LDS02WAB65N-LU	
LHT52	

Showing 1 to 2 of 2 results

50 per page ▾ Previous Next



Fleet > LHT52

## LHT52

Serial Number: 2532124565261235 Last update: Mon Jan 03 2022 16:50:26 GMT+0800

Dashboard History Downlinks Configuration Debug Rules Permissions

**TempC\_SHT**  
4 minutes ago



2022年1月3日 GMT+8下午4:30  
TempC\_SHT : 25.06

**TempC\_DS**  
4 minutes ago



2022年1月3日 GMT+8下午4:30  
TempC\_DS

**Hum\_SHT**  
4 minutes ago



2022年1月3日 GMT+8下午4:30  
Hum\_SHT

**Registration Key, Please keep it safely.**

DEV EUI: A84041C161 [REDACTED]

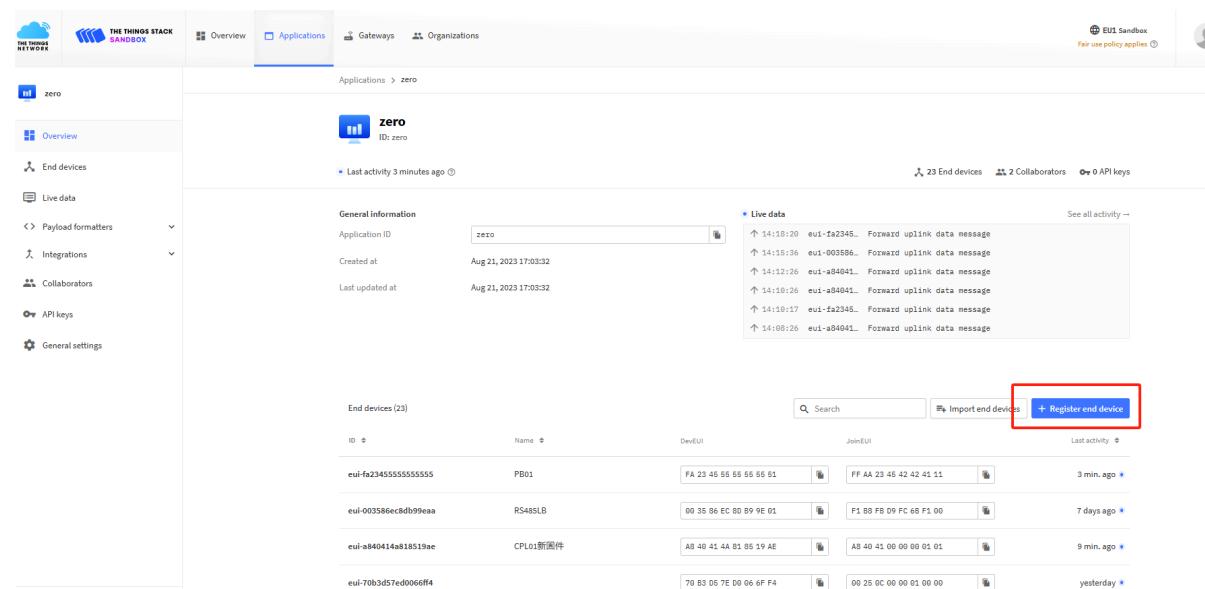
APP EUI: A8404100C00 [REDACTED]1

APP KEY: 7EC8A9C917386DFC5DBF73B [REDACTED]

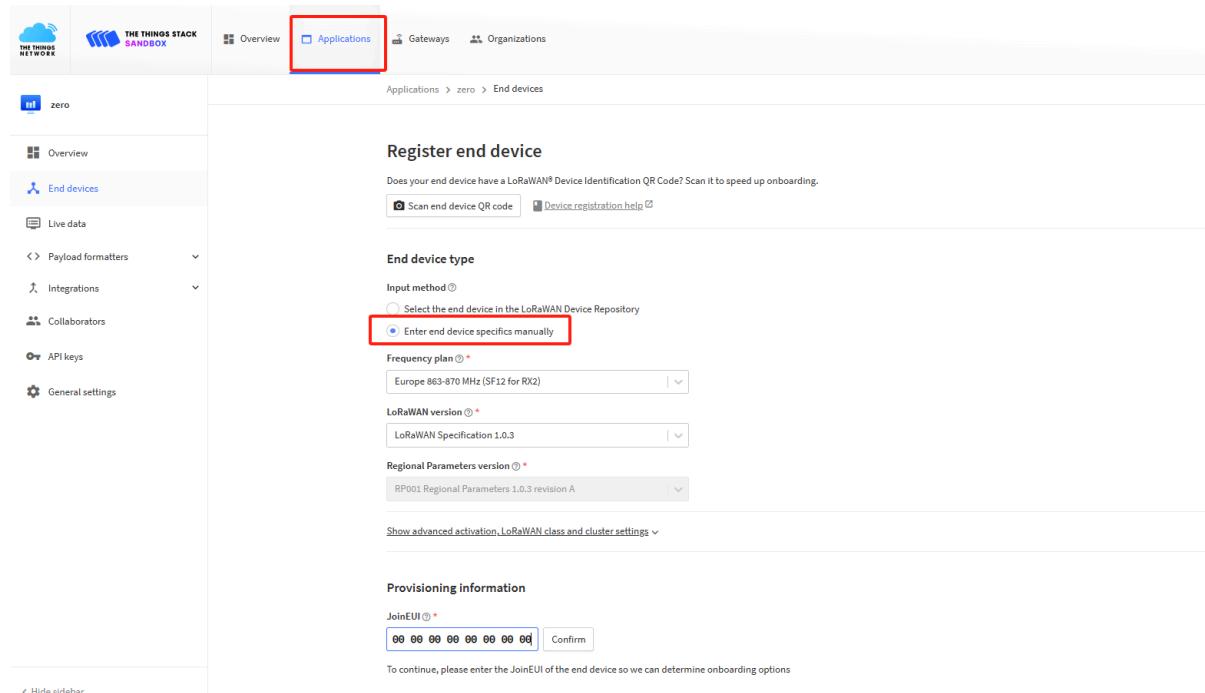
SN: LST2565 [REDACTED]7



# User Manual for LoRaWAN /NB-IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual



The screenshot shows the 'Applications' section of The Things Stack. The 'zero' application is selected. The 'General information' table shows Application ID: zero, Created at: Aug 21, 2023 17:03:32, and Last updated at: Aug 21, 2023 17:03:32. The 'Live data' section shows a list of uplink messages. Below this is a table of 'End devices (23)' with columns for ID, Name, DevEUI, JoinEUI, and Last activity. The 'Register end device' button is highlighted with a red box.



The screenshot shows the 'End devices' section for the 'zero' application. The 'Register end device' form is displayed. The 'End device type' section includes 'Input method' (radio buttons for 'Select the end device in the LoRaWAN Device Repository' and 'Enter end device specifics manually', with the latter highlighted by a red box). Other fields include 'Frequency plan' (Europe 863-870 MHz (SF12 for RX2)), 'LoRaWAN version' (LoRaWAN Specification 1.0.3), and 'Regional Parameters version' (RP001 Regional Parameters 1.0.3 revision A). The 'Provisioning information' section shows a 'JoinEUI' field with the value '00 00 00 00 00 00 00 00' and a 'Confirm' button. A note at the bottom says 'To continue, please enter the JoinEUI of the end device so we can determine onboarding options'.

# User Manual for LoRaWAN /NB-IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual

zero

- Overview
- End devices**
- Live data
- Payload formatters
- Integrations
- Collaborators
- API keys
- General settings

Select the end device in the LoRaWAN Device Repository  
 Enter end device specifics manually

Frequency plan ④ \*

LoRaWAN version ④ \*

Regional Parameters version ④ \*

Show advanced activation, LoRaWAN class and cluster settings

**Provisioning information**

JoinEUI ④ \*

This end device can be registered on the network

DevEUI ④ \*  
  9/50 used

AppKey ④ \*

End device ID ④ \*  
  
This value is automatically prefilled using the DevEUI

After registration

View registered end device  
 Register another end device of this type

**Register end device**

THE THINGS STACK SANDBOX Overview Applications Gateways Organizations EU1 Sandbox No SLA applicable paopao

Applications > zero > End devices > PB01 > Live data

**PB01** ID: eui-4fa234555555555555555555555555555

105 n/a Last activity 1 minute ago

Overview Live data Messaging Location Payload formatters General settings

Data preview

14:29:31 Schedule data downlink for transmission DevAddr: 26 08 FB E3 Rx1 Delay: 6

14:29:31 Forward uplink data message DevAddr: 26 08 FB E3 Payload: { Alarm: "FALSE", BatV: 3.366, Hum\_SHT41: 67.6, Sound\_ACK: "OPEN", Sound\_Key: "OPEN", TempC\_SHT41: 26.8 } 60 26 03 00 01 0C 02 A3 FPort: 2 Data rate: SF128W125 SNR: 9.2 RSSI

14:29:31 Successfully processed data message DevAddr: 26 08 FB E3

14:29:26 Forward join accept message DevAddr: 26 08 FB E3 JoinEUI: FF AA 23 45 42 42 41 11 DevEUI: FA 23 45 65 65 65 65 65

14:29:24 Successfully processed join-request DevAddr: 26 08 B1 92 JoinEUI: FF AA 23 45 42 42 41 11 DevEUI: FA 23 45 65 65 65 65 65

14:29:23 Accept join-request DevAddr: 26 08 FB E3 JoinEUI: FF AA 23 45 42 42 41 11 DevEUI: FA 23 45 65 65 65 65 65

14:29:20 Schedule data downlink for transmission DevAddr: 26 08 B1 92 Rx1 Delay: 6

Applications > zero > End devices > PB01 > Live data

**PB01** ID: eui-4fa234555555555555555555555555555

2 n/a Last activity 4 minutes ago

Overview Live data Messaging Location Payload formatters General settings

Data preview

14:35:08 Schedule data downlink for transmission DevAddr: 26 08 FB E3 Rx1 Delay: 6

14:35:03 Forward uplink data message DevAddr: 26 08 FB E3 Payload: { BAT: 3.33, FIRMWARE\_VERSION: "10.0.0", FREQUENCY\_BAND: "EU868", SENSOR\_MODEL: "PB01-L", SUB\_BAND: "NULL" } 35 0A 00 01 FF 00 02 FPort: 5 Data rate: SF78W125 SNR: 13.8 RSSI

14:35:02 Successfully processed data message DevAddr: 26 08 FB E3

14:34:57 Schedule data downlink for transmission DevAddr: 26 08 FB E3 FPort: 1 MAC payload: 80 80 Rx1 Delay: 6

14:34:47 Forward uplink data message DevAddr: 26 08 FB E3 Payload: { Alarm: "TRUE", BatV: 3.342, Hum\_SHT41: 65.7, Sound\_ACK: "OPEN", Sound\_Key: "OPEN", TempC\_SHT41: 26.5 } 00 0E 03 01 01 09 02 91 FPort: 2 Data rate: SF88W125 SNR: 16 RSSI

14:34:47 Successfully processed data message DevAddr: 26 08 FB E3

14:34:02 Receive downlink data message DevAddr: 26 01 FPort: 1

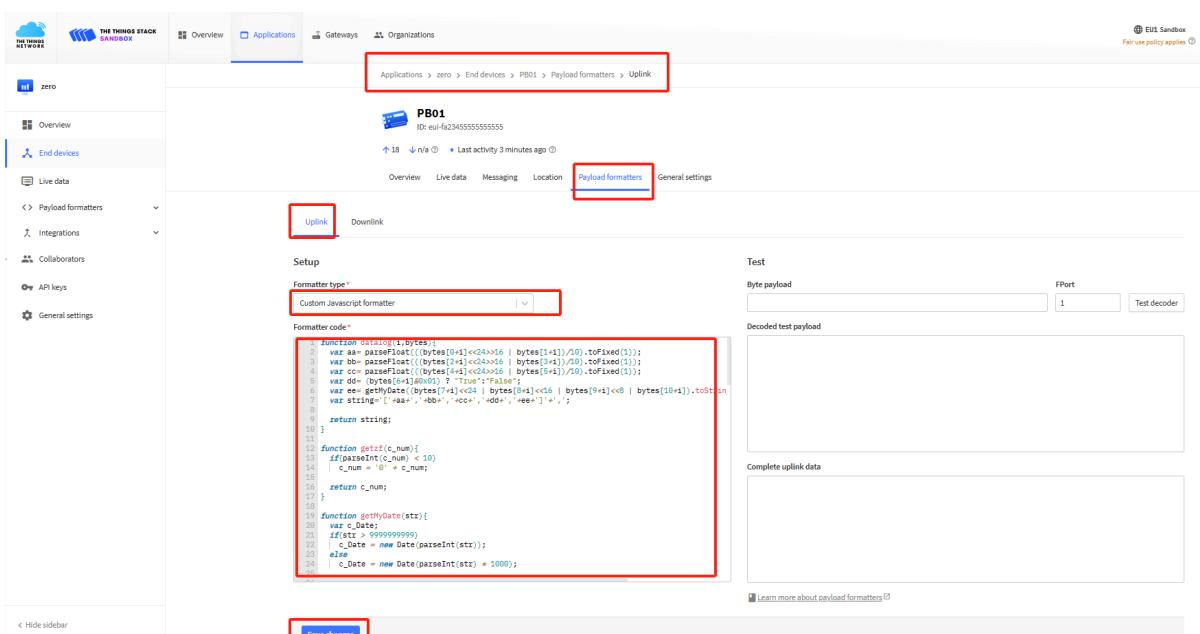
14:29:31 Schedule data downlink for transmission DevAddr: 26 08 FB E3 Rx1 Delay: 6

# User Manual for LoRaWAN /NB -IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual

35 0A 00 01 FF 0C AE

35 0A 00 01 FF 0C AE

35 01 00 01 FF 0C DE



```

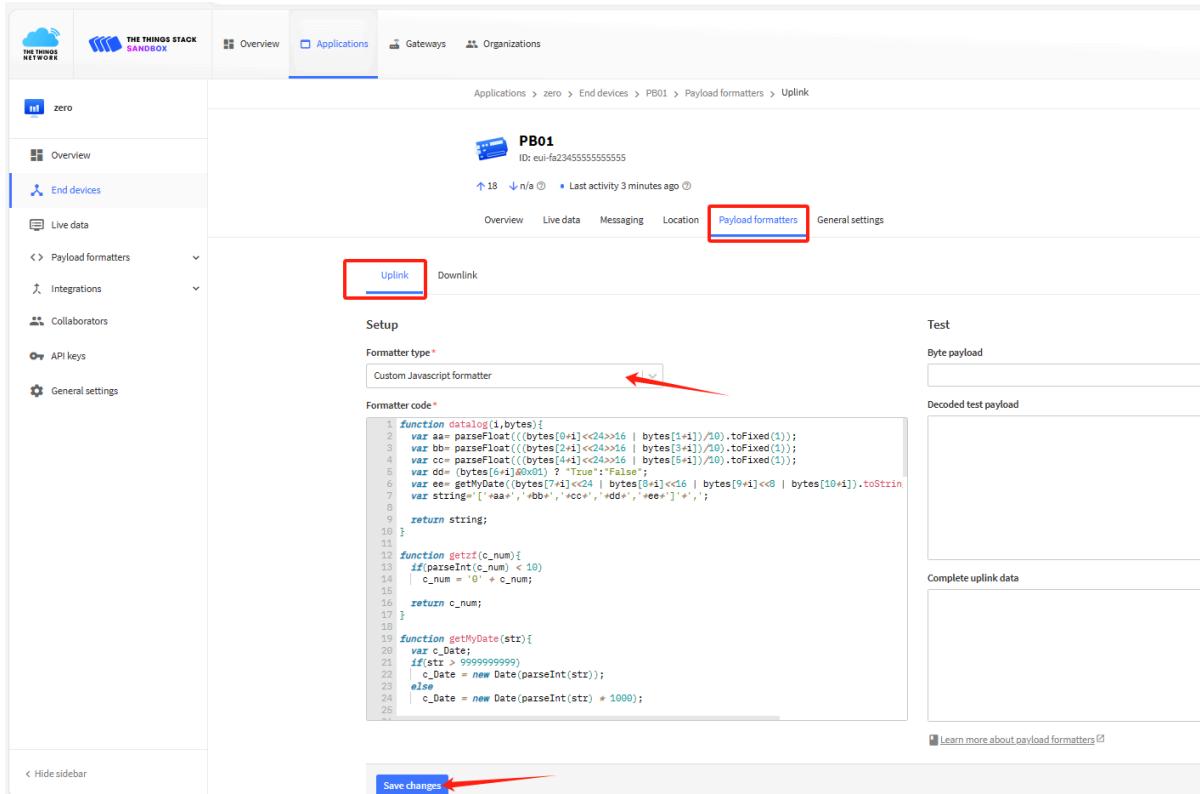
function details(i,bytes){
  var aa= parseFloat(((bytes[0]+1)<>4>>16 | bytes[1+1]>>10).toFixed(1));
  var bb= parseFloat(((bytes[2]+1)<>4>>16 | bytes[3+1]>>10).toFixed(1));
  var cc= parseFloat(((bytes[4]+1)<>4>>16 | bytes[5+1]>>10).toFixed(1));
  var dd= (bytes[6+1]&0x01) ? "True" : "False";
  var ee= getMyDate((bytes[7+1]<>24 | bytes[8+1]<>16 | bytes[9+1]<>0 | bytes[10+1]).toString());
  var string= ["aa", "bb", "cc", "dd", "ee"];
  return string;
}

function getzf(c_num){
  if(parseInt(c_num) < 10)
    | c_num = '0' + c_num;
  return c_num;
}

function getMyDate(str){
  var c_Date;
  if(str > 9999999999)
    | c_Date = new Date(parseInt(str));
  else
    | c_Date = new Date(parseInt(str) * 1000);
}

```

Save changes



```

function details(i,bytes){
  var aa= parseFloat(((bytes[0]+1)<>4>>16 | bytes[1+1]>>10).toFixed(1));
  var bb= parseFloat(((bytes[2]+1)<>4>>16 | bytes[3+1]>>10).toFixed(1));
  var cc= parseFloat(((bytes[4]+1)<>4>>16 | bytes[5+1]>>10).toFixed(1));
  var dd= (bytes[6+1]&0x01) ? "True" : "False";
  var ee= getMyDate((bytes[7+1]<>24 | bytes[8+1]<>16 | bytes[9+1]<>0 | bytes[10+1]).toString());
  var string= ["aa", "bb", "cc", "dd", "ee"];
  return string;
}

function getzf(c_num){
  if(parseInt(c_num) < 10)
    | c_num = '0' + c_num;
  return c_num;
}

function getMyDate(str){
  var c_Date;
  if(str > 9999999999)
    | c_Date = new Date(parseInt(str));
  else
    | c_Date = new Date(parseInt(str) * 1000);
}

```

Save changes

# User Manual for LoRaWAN /NB -IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual

The screenshot shows the configuration of a LoRaWAN end node (PB01) in The Things Stack. The interface is divided into two main sections: the left sidebar and the main content area.

**Left Sidebar:**

- THE THINGS NETWORK** and **THE THINGS STACK BANDBOX** icons.
- Applications** tab is selected.
- zero** project is selected.
- End devices** section is expanded, showing **PB01** (ID: eui-fa23455555555555).
- Payload formatters** section is expanded, showing the **Uplink** tab selected.
- General settings** section is expanded.

**Main Content Area:**

**Uplink Tab:**

- Payload formatters** tab is selected.
- Formatter type**: Custom Javascript formatter.
- Formatter code** (highlighted with a red box and arrow):

```

1 function datalog1(i,bytes){
2     var aa= parseFloat(((bytes[0+1]<>24)<>16 | bytes[1+1]<>10).toFixed(1));
3     var bb= parseFloat(((bytes[2+1]<>24)<>16 | bytes[3+1]<>10).toFixed(1));
4     var cc= parseFloat(((bytes[4+1]<>24)<>16 | bytes[5+1]<>10).toFixed(1));
5     var dd= (bytes[6+1]&0x01) ? "True" : "False";
6     var ee= getMyDate((bytes[7+1]<>24 | bytes[8+1]<>16 | bytes[9+1]<>8 | bytes[10+1]).toString());
7     var string= ["aa=", +aa+, "+bb=", +bb+, "+cc=", +cc+, "+dd=", +dd+, "+ee=" +ee+];
8
9     return string;
10 }
11
12 function getMyDate(c_num){
13     if(parseInt(c_num) < 10)
14         | c_num = '0' + c_num;
15
16     return c_num;
17 }
18
19 function getMyDate(str){
20     var c_Date;
21     if(str > 9999999999)
22         | c_Date = new Date(parseInt(str));
23     else
24         | c_Date = new Date(parseInt(str) * 1000);
25

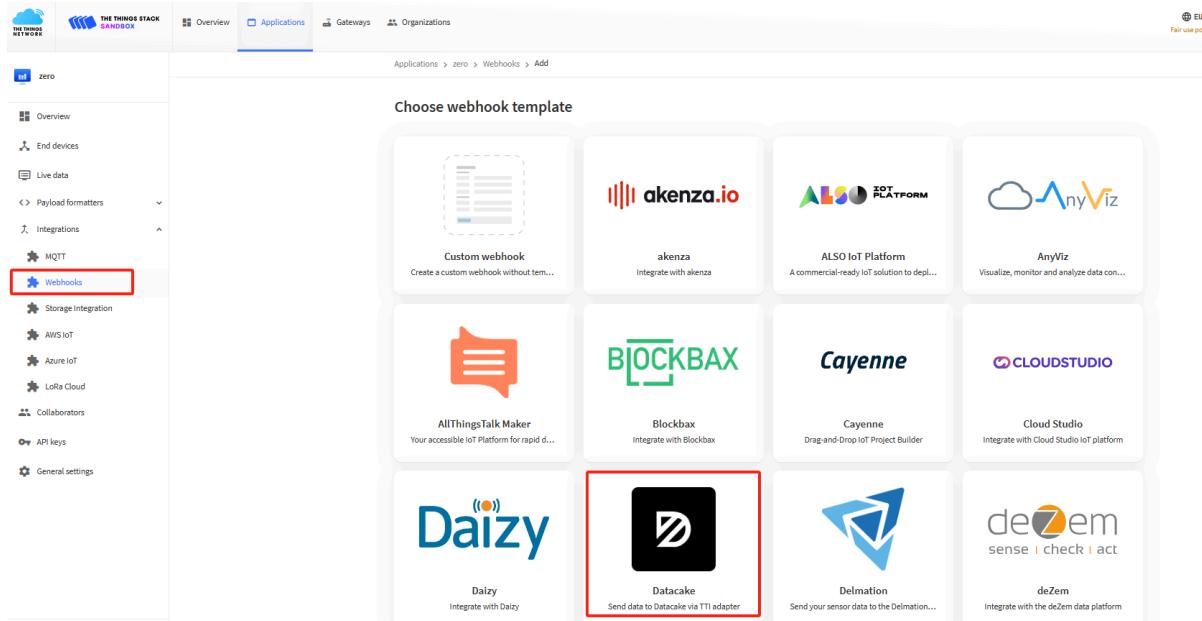
```
- Save changes** button (highlighted with a red box and arrow).

**Live Data Tab:**

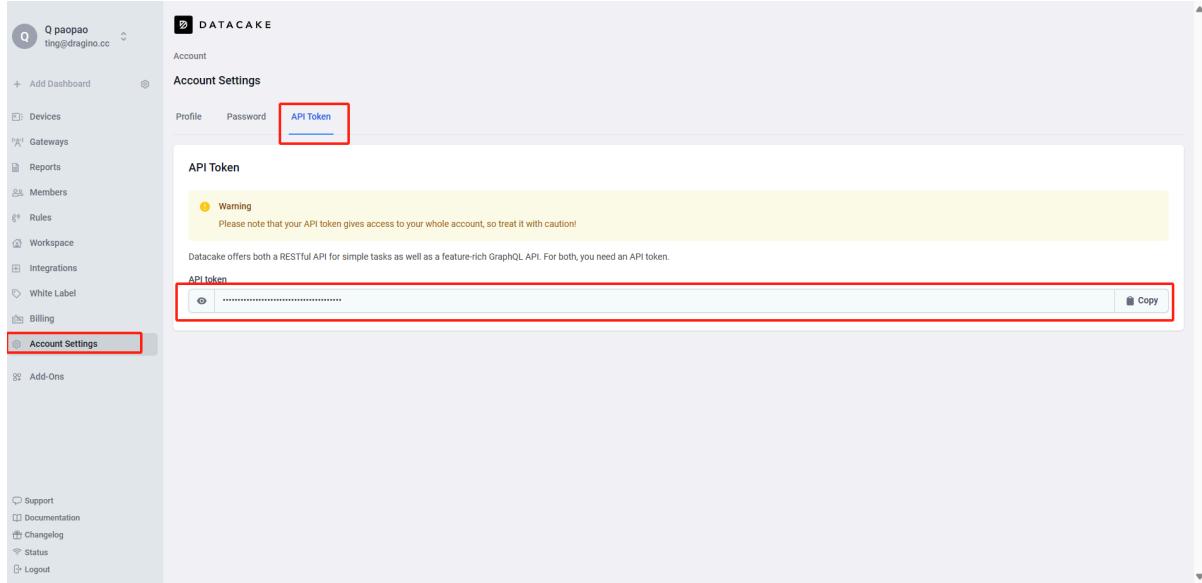
- PB01** device is selected.
- Time** and **Type** columns are shown.
- Data preview** table (highlighted with a red box and arrow):

Time	Type	DevAddr	FPort	Payload
14:45:39	Forward uplink data message	26 00 66 02	1	[ DATALOG [ 0.02.0.27.0.False,2024-09-19 04:44:13], [ 0.63.0.27.0.False,2024-09-19 00:04:13], [ 0.64.0.27.0.False,2024-09-19 00:24:13], [ 0.64.0.27.0.False,2024-09-19 00:44:13], [
14:45:39	Successfully processed data message	26 00 66 02	1	
14:45:39	Schedule data downlink for transmiss...	26 00 66 02	1	
14:45:39	Forward uplink data message	26 00 66 02	1	[ Alarm: "TRUE", BatV: 3.439, Hum_SHT41: 97.7, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.3 ]
14:45:39	Successfully processed data message	26 00 66 02	1	
14:45:41	Receive downlink data message	26 00 66 02	1	[ 1.66 20 AB CF 66 3D C1 EF 05 ]
14:45:49	Forward uplink data message	26 00 66 02	1	[ Alarm: "FALSE", BatV: 3.439, Hum_SHT41: 97.9, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.3 ]
14:45:49	Successfully processed data message	26 00 66 02	1	
14:46:00	Forward uplink data message	26 00 66 02	1	[ Alarm: "FALSE", BatV: 3.439, Hum_SHT41: 97.6, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.8 ]
14:46:00	Successfully processed data message	26 00 66 02	1	
14:46:00	Schedule data downlink for transmiss...	26 00 66 02	1	
14:49:59	Forward uplink data message	26 00 66 02	1	[ Alarm: "TRUE", BatV: 3.439, Hum_SHT41: 97.7, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.9 ]
14:49:59	Successfully processed data message	26 00 66 02	1	

# User Manual for LoRaWAN /NB -IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual



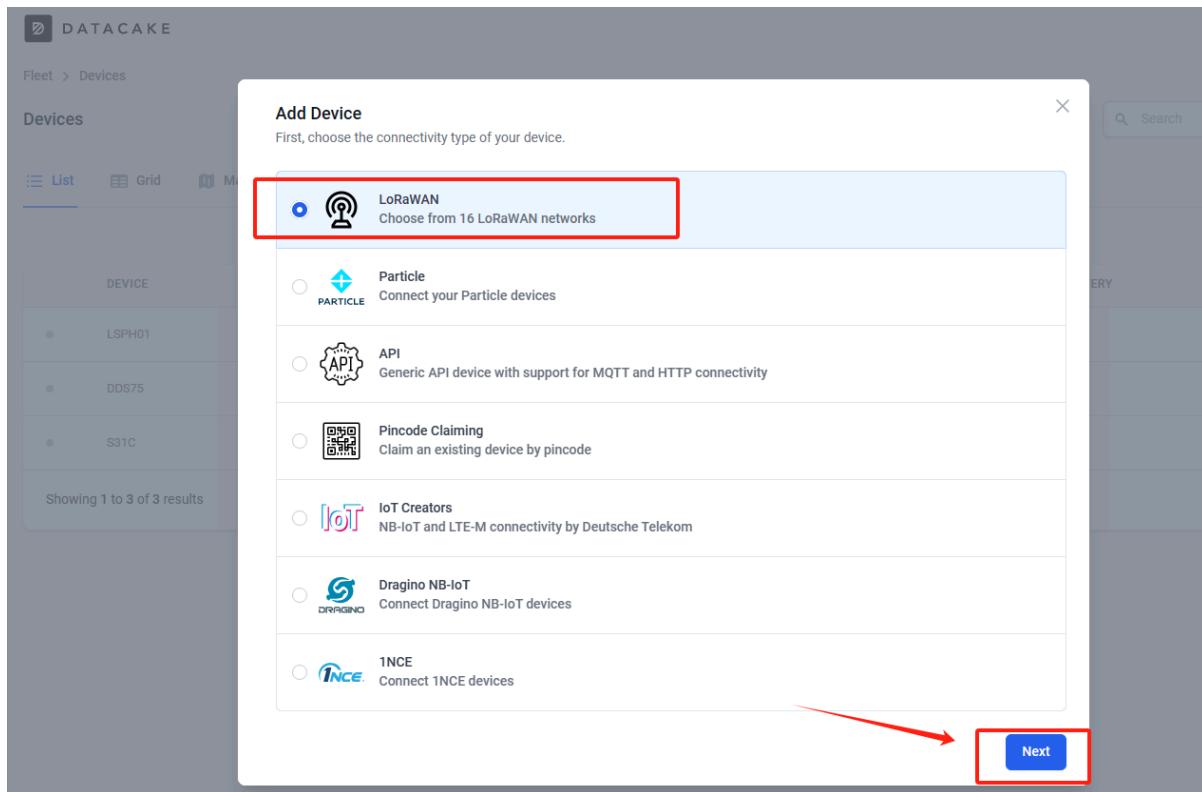
The screenshot shows the 'Webhooks' section of The Things Stack interface. The sidebar on the left is expanded, showing various integration options like MQTT, Webhooks, Storage Integration, AWS IoT, Azure IoT, LoRa Cloud, Collaborators, API keys, and General settings. The 'Webhooks' tab is selected and highlighted with a red box. The main area displays a grid of webhook templates. One template, 'Datacake', is selected and highlighted with a red box. The Datacake card shows its logo, name, and description: 'Integrate with Datacake'.

The screenshot shows the 'Account Settings' page for Datacake. The sidebar on the left includes options like Add Dashboard, Devices, Gateways, Reports, Members, Rules, Workspace, Integrations, White Label, Billing, and Account Settings. The 'Account Settings' tab is selected and highlighted with a red box. The main content area is the 'API Token' section. It includes a warning message: 'Please note that your API token gives access to your whole account, so treat it with caution!'. Below this, it says 'Datacake offers both a RESTful API for simple tasks as well as a feature-rich GraphQL API. For both, you need an API token.' An 'API token' input field is shown, with a red box highlighting the entire input area. There is also a 'Copy' button next to the input field.

The Things Stack Sandbox interface showing the setup of a webhook for Datacake. The webhook ID is pb01-l and the token is masked. A red arrow points to the 'Create Datacake webhook' button with the text 'Paste the API here'.

The Datacake interface showing a list of devices. The 'Devices' tab is selected, and a red box highlights the '+ Add Device' button.



**Add LoRaWAN Device**

STEP 1 Product    STEP 2 Network Server    STEP 3 Devices    STEP 4 Plan

**Datacake Product**  
You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

**New Product from template**  
Create new product from a template

**Existing Product**  
Add devices to an existing product

**New Product**  
Create new empty product

**New Product**  
If your device is not available as a template, you can start with an empty device. You will have to create the device definition (fields, dashboard) and provide the payload decoder in the device's configuration.

**Product Name**  
pb01-l

**Back** **Next**

### Add LoRaWAN Device

**STEP 1**  
Product

**STEP 2**  
Network Server

**STEP 3**  
Devices

**STEP 4**  
Plan

**Datacake Product**  
You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

**New Product from template**  
Create new product from a template

**Existing Product**  
Add devices to an existing product

**New Product**  
Create new empty product

**New Product**  
If your device is not available as a template, you can start with an empty device. You will have to create the device definition (fields, dashboard) and provide the payload decoder in the device's configuration.

**Product Name**  
pb01-l

**Back** **Next**

## Add LoRaWAN Device

STEP 1  
Product

STEP 2  
Network Server

STEP 3  
Devices

STEP 4  
Plan

### Network Server

Please choose the LoRaWAN Network Server that your devices are connected to.

 Datacake LNS AUTOMATIC SETUP  
Start and scale easily with a managed LNS

 The Things Stack V3  
TTN V3 / Things Industries

 Helium  
Use your own console

 LORIOT

 ChirpStack

 Actility

Showing 1 to 5 of 15 results

Previous Next

Next Back

**Add LoRaWAN Device**

STEP 1 Product    STEP 2 Network Server    STEP 3 Devices    STEP 4 Plan

**Add Devices**

**Manual**    Import from The Things Stack

Please provide one or multiple LoRaWAN device EUIs along with the corresponding names they should have on Datacake.

Alternatively, you can choose to upload a CSV file that contains the DevEUI, device Name, location, and a set of tags. For more information on how to format the file, please refer to [our documentation](#).

Drag and drop a .csv file here or click to choose one

DEVEUI	NAME	LOCATION	TAGS
FA 23 45 55 55 55 55 51 8 bytes	PB01	Location	Add tag

+ Add another device

**Next**

**Payload Decoder**

When your devices sends data, the payload will be passed to the payload decoder, alongside the event's name. The payload decoder then transforms it to measurements.

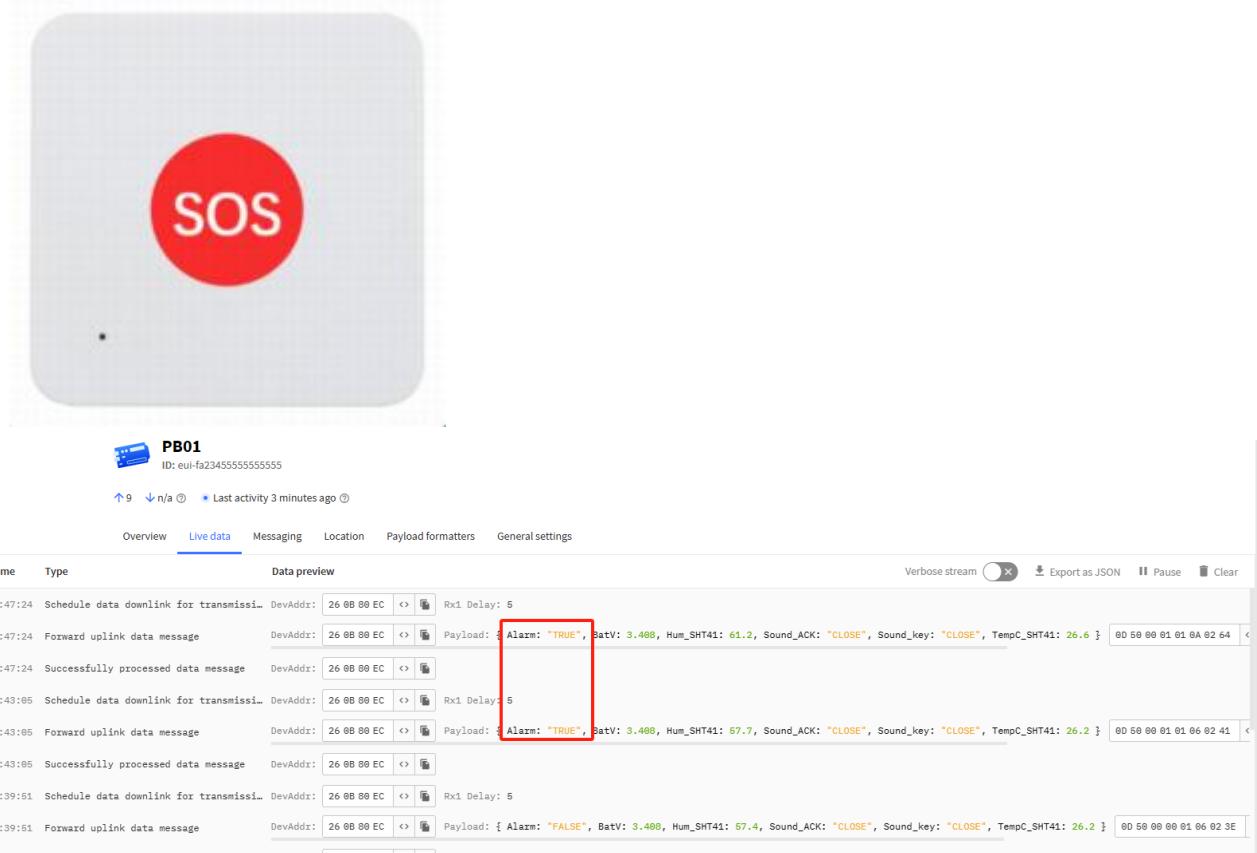
```

1< function Decoder(payload, port) {
2<   /*
3<   return [
4<     {
5<       field: "TEST",
6<       value: 123
7<     }
8<   ];
9< }
10>

```

**Payload** **Port** **1** **Try Decoder**

User Manual for LoRaWAN /NB -IoT End Nodes - PB01 -- LoRaWAN Push Button User Manual



The screenshot shows a LoRaWAN device interface. At the top, there is a large red button with the word "SOS" in white. Below the button, the device is identified as "PB01" with the ID "eui-fa23455555555555". The interface includes tabs for "Overview", "Live data" (which is selected), "Messaging", "Location", "Payload formatters", and "General settings". The "Live data" tab displays a table of data messages. The table has columns for "Time", "Type", and "Data preview". The data preview shows the following messages:

Time	Type	Data preview
↓ 15:47:24	Schedule data downlink for transmission	DevAddr: 26 0B 00 EC   Rx1 Delay: 5
↑ 15:47:24	Forward uplink data message	DevAddr: 26 0B 00 EC   Payload: { Alarm: "TRUE", BatV: 3.408, Hum_SHT41: 61.2, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.6 }   0D 50 00 01 01 0A 02 64
↑ 15:43:05	Successfully processed data message	DevAddr: 26 0B 00 EC
↓ 15:43:05	Schedule data downlink for transmission	DevAddr: 26 0B 00 EC   Rx1 Delay: 5
↑ 15:43:05	Forward uplink data message	DevAddr: 26 0B 00 EC   Payload: { Alarm: "TRUE", BatV: 3.408, Hum_SHT41: 57.7, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.2 }   0D 50 00 01 01 06 02 41
↑ 15:43:05	Successfully processed data message	DevAddr: 26 0B 00 EC
↓ 15:39:51	Schedule data downlink for transmission	DevAddr: 26 0B 00 EC   Rx1 Delay: 5
↑ 15:39:51	Forward uplink data message	DevAddr: 26 0B 00 EC   Payload: { Alarm: "FALSE", BatV: 3.408, Hum_SHT41: 57.4, Sound_ACK: "CLOSE", Sound_key: "CLOSE", TempC_SHT41: 26.2 }   0D 50 00 00 01 06 02 3E
↑ 15:39:51	Successfully processed data message	DevAddr: 26 0B 00 EC

