

LC03



1. Introduction

1.1 What is LC03 LoRaWAN Socket

The Dragino LC03 is a **LoRaWAN Socket** for Internet of Things solution. It controls the device power switch via the IoT server downlink of the LoRaWAN wireless protocol.

It can be applied to **smart agricultural irrigation, smart home, industrial automation and other industries**.

The LoRa wireless technology used in the LC03 allows devices to send data at low rates and reach extremely long transmission distances. The technology provides ultra-long-range spread spectrum communication and high anti-interference capabilities while minimizing current consumption.

By default, LC03 has the function of detecting the access status of devices that need power. When the device that needs power is connected, it will automatically upload the data packet; when the device that needs power is disconnected, LC03 will also automatically upload the data packet.

LC03 **wireless OTA update** which make user easy to use.

Each LC03 is pre-load with a set of unique keys for LoRaWAN registrations, register these keys to local LoRaWAN server and it will auto connect after power on.

1.2 Features

- LoRaWAN Class C modes

- Optional Customized LoRa Protocol
- Frequency Bands: CN470/EU433/KR920/US915/EU868/AS923/AU915/RU864/IN865/MA869
- US standard socket
- Detect machine plug/unplug status
- Support Downlink and Timing control relay opening and closing
- AT Commands to change parameters
- Remotely configure parameters via LoRaWAN Downlink
- Firmware upgradable via program port
- Support wireless OTA update firmware
- Uplink on periodically
- Downlink to change configure

1.3 Specification

Common DC Characteristics:

- Supply Voltage: AC 125V ~60HZ
- Operating Temperature: -40~+70°C
- Current: 15A Resistive load

LoRa Spec:

- Frequency Range, Band 1(HF): 902.3~ 914.9 Mhz
- Max -1.51 dBm constant RF output vs.
- RX sensitivity: down to -139 dBm.
- Excellent blocking immunity

1.4 Applications

- Smart buildings & home automation
- Logistics and supply chain management
- Smart metering
- Smart agriculture
- Smart cities
- Smart factory

1.5 Working mode

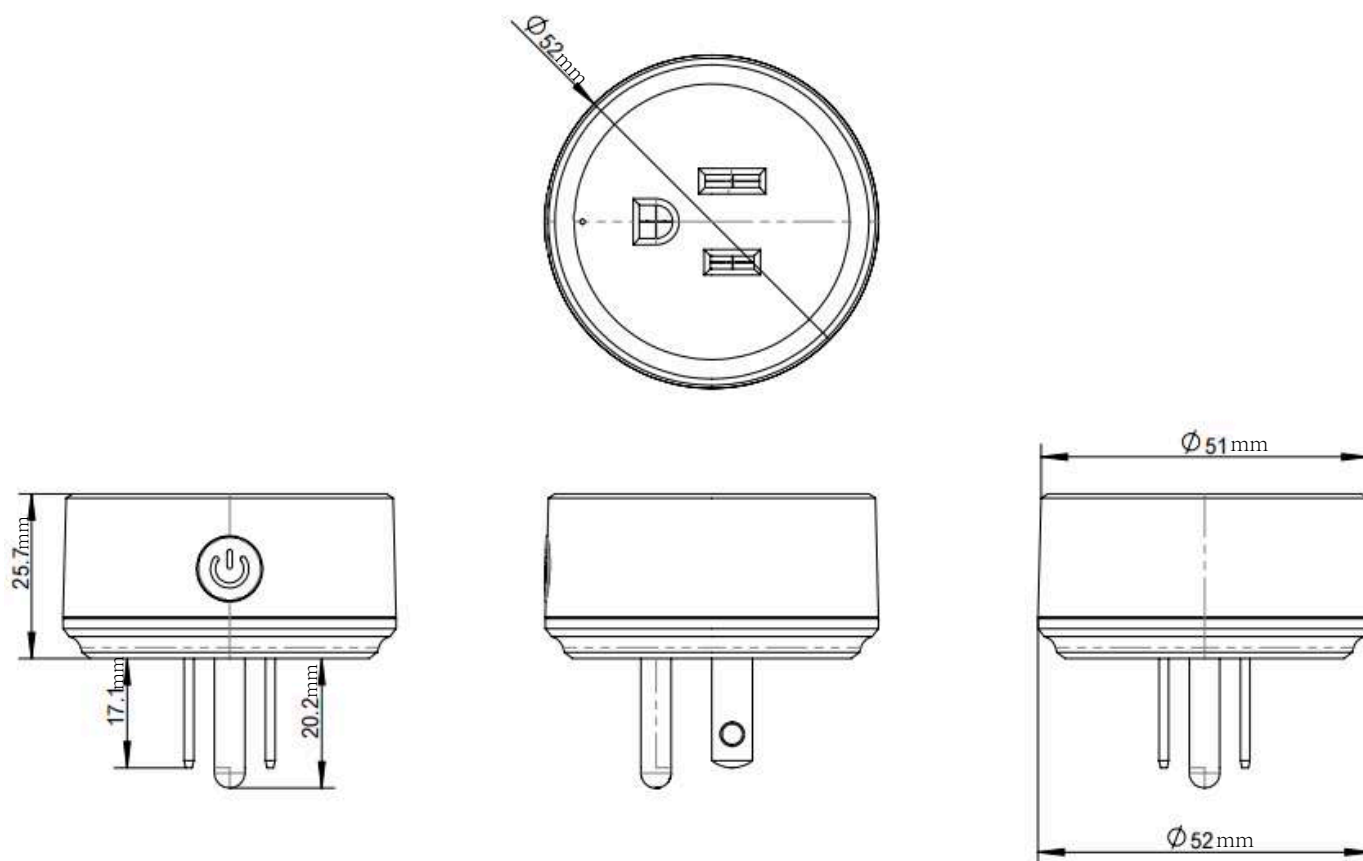
Working Mode: In this mode, Sensor will work as LoRaWAN Sensor to Join LoRaWAN network and send out sensor data to server. Between each sampling/tx/rx periodically, sensor will be in IDLE mode).

1.6 Button & LEDs



Behavior on ACT	Function	Action
 x1	Active Device	Red led will fast blink 5 times, device will enter Bootloader mode for 5 seconds. And then start to JOIN LoRaWAN network. Red led will solidly turn on for 5 seconds after joined in network.

1.7 Mechanical



2. Configure LC03 to connect to LoRaWAN network

2.1 How it works

The LC03 is configured as **LoRaWAN OTAA Class C** mode by default. It has OTAA keys to join LoRaWAN network. To connect a local LoRaWAN network, you need to input the OTAA keys in the LoRaWAN IoT server and press the button to activate the LC03. It will automatically join the network via OTAA and start to send the sensor value. The default uplink interval is 1 hour.

2.2 Quick guide to connect to LoRaWAN server (OTAA)

Following is an example for how to join the [TTN v3 LoRaWAN Network](#). Below is the network structure; we use the [LPS8v2](#) as a LoRaWAN gateway in this example.

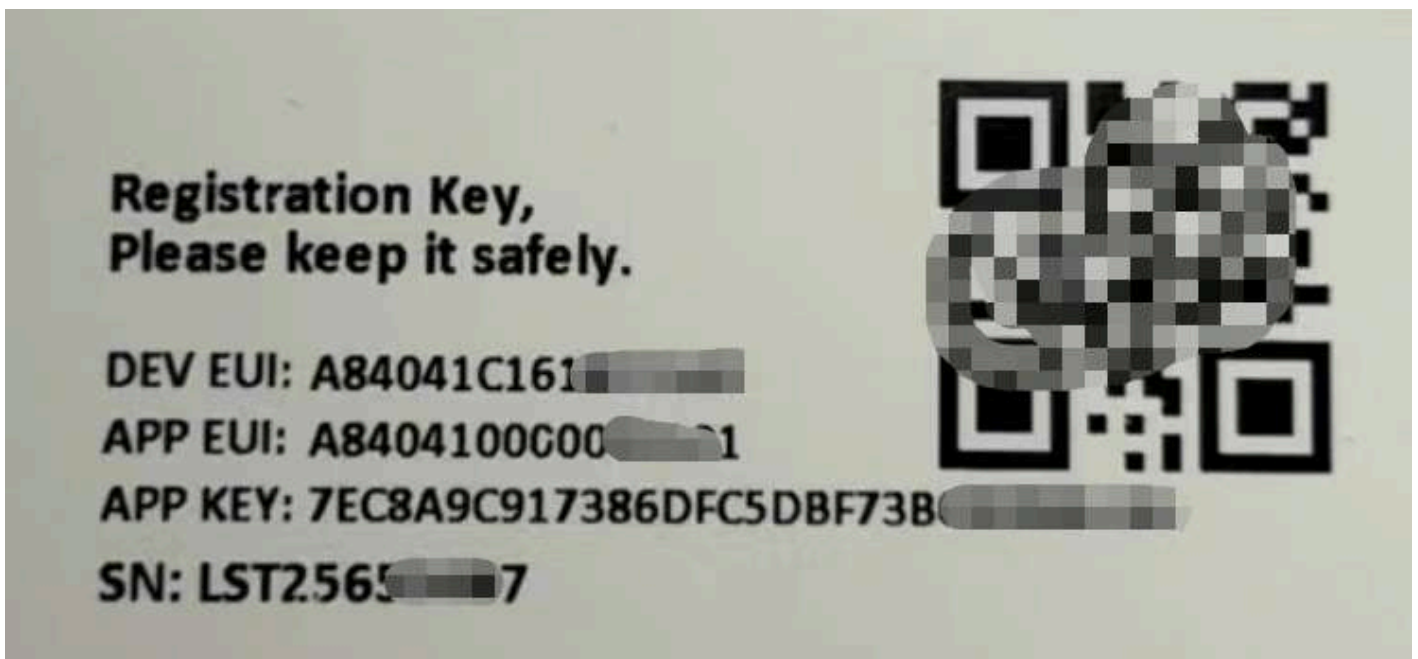
The LPS8V2 is already set to connected to [TTN network](#), so what we need to now is configure the TTN server.

LC03 in a LoRaWAN Network



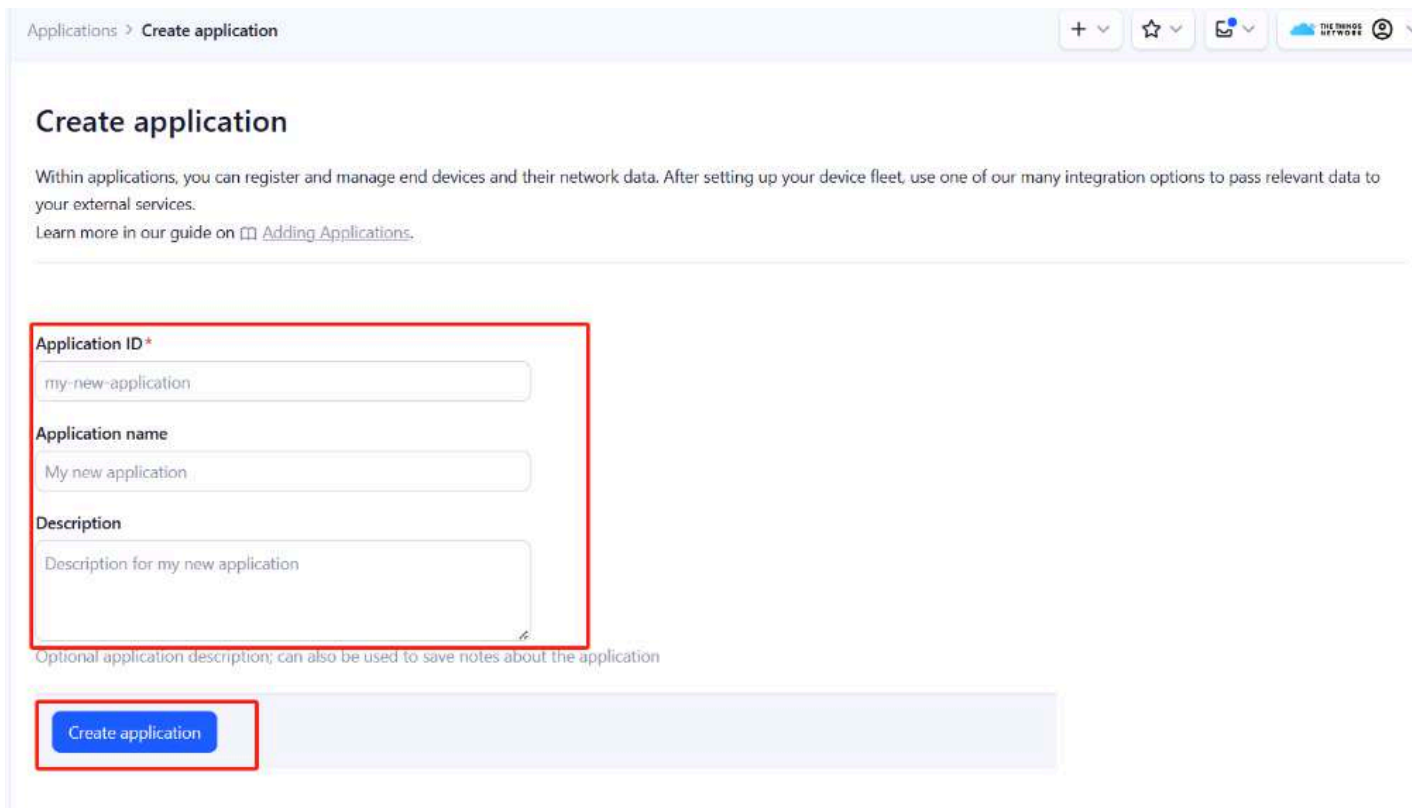
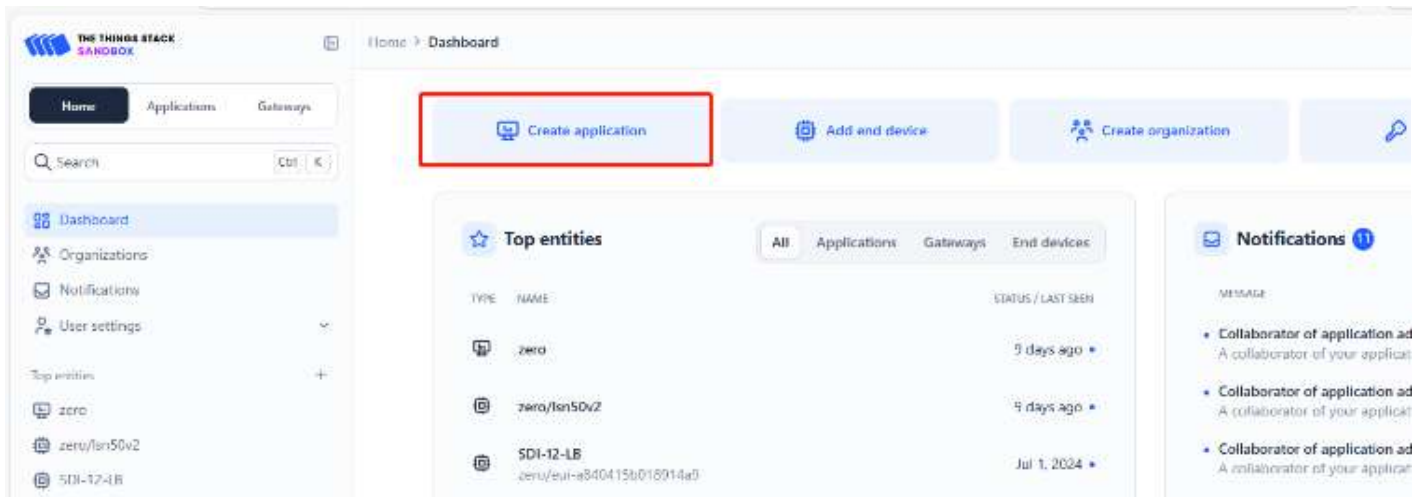
Step 1: Create a device in TTN with the OTAA keys from LC03.

Each LC03 is shipped with a sticker with the default device EUI as below:

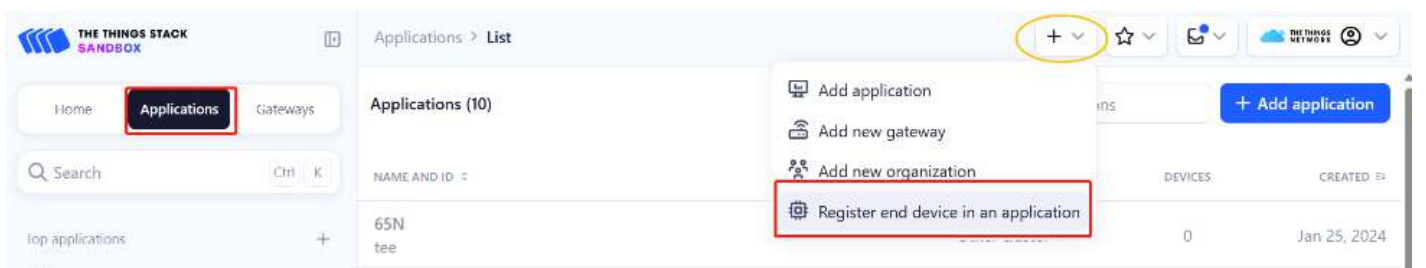


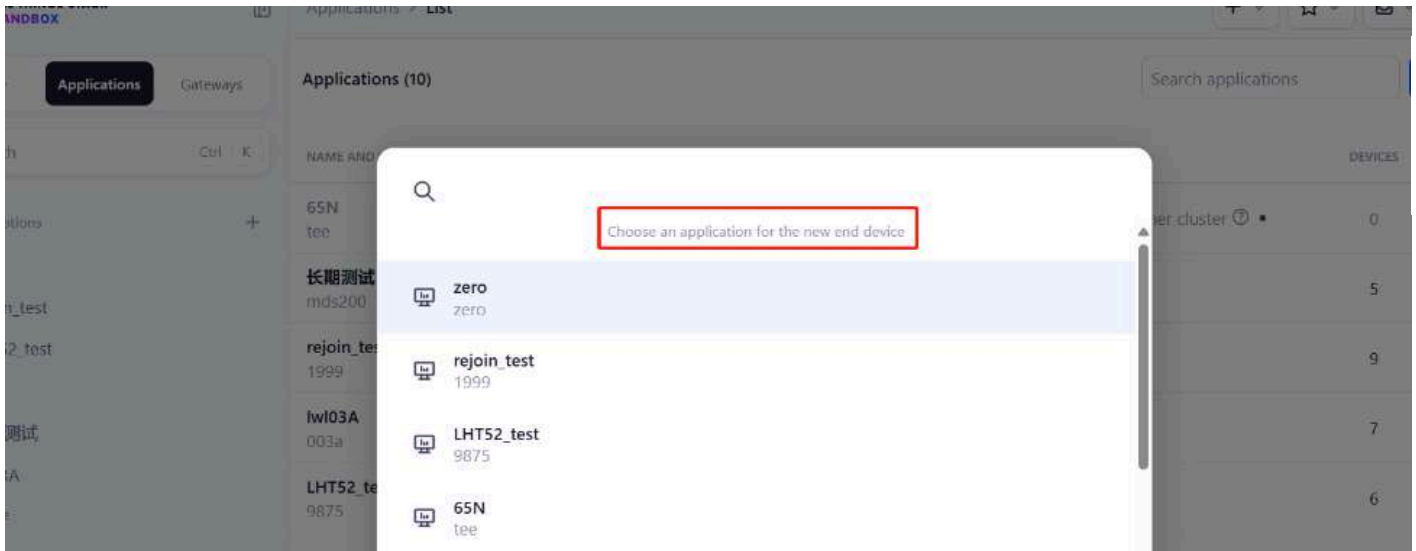
You can enter this key in the LoRaWAN Server portal. Below is TTN screen shot:

Create the application.



Add devices to the created Application.





Enter end device specifics manually.

[Applications](#) > [zero](#) > [End devices](#) > **Register end device**

Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

[📄 Scan end device QR code](#)[📄 Device registration help](#)

End device type

Input method ⓘ

☐ Select the end device in the LoRaWAN Device Repository

☒ Enter end device specifics manually

Frequency plan ⓘ *

Europe 863-870 MHz (SF12 for RX2)

Select the frequency corresponding to the node

LoRaWAN version ⓘ *

LoRaWAN Specification 1.0.3

Regional Parameters version ⓘ *

RP001 Regional Parameters 1.0.3 revision A

[Show advanced activation, LoRaWAN class and cluster settings](#)

Provisioning information

JoinEUI ⓘ *

00 00 00 00 00 00 00 00

Confirm

AppEUI

To continue, please enter the JoinEUI of the end device so we can determine onboarding options

Add DevEUI and AppKey.

Customize a platform ID for the device.

Provisioning information

JoinEUI ⓘ *

00 00 00 00 00 00 00 00

Reset

This end device can be registered on the network

DevEUI ⓘ *

.. .. .

Generate

10/50 used

AppKey ⓘ *

.. .. .

Generate

End device ID ⓘ *

my-new-device

After registration

- ☒ View registered end device
- ☐ Register another end device of this type

Register end device

Step 2: Add decoder.

In TTN, user can add a custom payload so it shows friendly reading.

Click this link to get the decoder: [Dragino end node decoder](#)

Below is TTN screen shot:

lc03
ID: lc03

Last activity 12 minutes ago • n/a up / n/a down

Device overview Live data Messaging Location **Payload formatters** Settings

Uplink Downlink

Setup

Formatter type^{*}
Custom Javascript formatter

Formatter code^{*}

```

1 function Datalog(bytes, num)
2 {
3     var datalog_timestamp = "";
4     var datalog_event = "";
5     var datalog_plugstatus = "";
6     var datalog_relaystatus = "";
7     var datalog_reply = "";
8
9     datalog_timestamp |= bytes[num] << 24;
10    datalog_timestamp |= bytes[1 + num] << 16;
11    datalog_timestamp |= bytes[2 + num] << 8;
12    datalog_timestamp |= bytes[3 + num] << 8;
13
14    var eventType = bytes[4 + num];
15    switch (eventType) {
16        case 0x01:
17            datalog_event = "PLUG_INSERT_EVENT";
18            break;
19        case 0x02:
20            datalog_event = "PLUG_PULL_EVNET";
21            break;
22        case 0x03:
23            datalog_event = "HEARTBEAT_EVNET";
24            break;
25        case 0x04:

```

Replace the TTN original decoding with our decoding

Test

Byte payload 68 18 20 B4 03 00 00 FPort 2 Test decoder

Decoded test payload

```

{
  "Event": "HEARTBEAT_EVNET",
  "PlugStatus": "PULL_OUT",
  "RelayStatus": "CLOSE",
  "Timestamp": 1746986308
}

```

Users can enter the original payload test decoder here.

Complete uplink data

```

{
  "f_port": 2,
  "frm_payload": "a$agtAMAAA==",
  "decoded_payload": {
    "Event": "HEARTBEAT_EVNET",
    "PlugStatus": "PULL_OUT",
    "RelayStatus": "CLOSE",
    "Timestamp": 1746986308
  },
  "rx_metadata": [
  ]
}

```

✓ Payload is valid

Learn more about payload formatters

Save changes

Step 3: Activate on LC03

Press the reset button once to activate the LC03.

Red led will fast blink 5 times, device will enter **bootloader mode** for 5 seconds. And then start to JOIN LoRaWAN network. Red led will solidly turn on for 5 seconds after joined in network.

After join success, it will start to upload messages to TTN and you can see the messages in the panel.

2.3 Uplink Payload

2.3.1 Device Status, FPORT=5

Users can use the downlink command(0x26 01) to ask LC03 to send device configure detail, include device configure status. LC03 will uplink a payload via FPort=5 to server.

The Payload format is as below.

Device Status (FPORT=5)

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

Example parse in TTNv3

The screenshot shows the TTNv3 interface for device LC03. The 'Live data' tab is selected, showing a list of messages. A red box highlights the payload of a 'Forward uplink data message' at 17:15:14, which contains device status information: FIRMWARE_VERSION: '1.1.0', FREQUENCY_BAND: 'EU868', SENSOR_MODEL: 'LC03', SUB_BAND: 'NULL'. Another red box highlights the 'FPort: 5' field in the same message.

Sensor Model: For LC03, this value is 0x46

Firmware Version: 0x0110, Means: v1.1.0 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

0x0b: CN470

0x0c: EU433

0x0d: KR920

0x0e: MA869

Sub-Band:

AU915 and US915: value 0x00 ~ 0x08

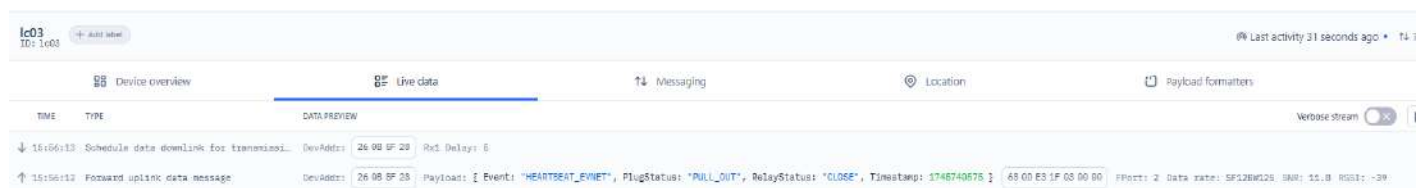
CN470: value 0x0B ~ 0x0C

Other Bands: Always 0x00

2.3.2 Sensor Data. FPORT=2

Sensor Data is uplink via FPORT=2

Size(bytes)	4	1	1	1
Value	Unix TimeStamp	EventType	Plugstatus	Relaystatus



Unit timestamp

Unit TimeStamp Example: 680DE31F(H) = 1745740575(D)

Put the decimal value into this link([Epoch Converter](#))to get the time.

Plugstatus

Example:

If payload is: 01H: Electrical INSERT

If payload is: 00H: Electrical PULL_OUT

Event

Example:

If payload is: 01H: PLUG_INSERT_EVENT

If payload is: 02H : PLUG_PULL_EVNET

If payload is: 03H :HEARTBEAT_EVNET

If payload is: 04H :RELAY_ACK_EVENT

Relaystatus

Example:

If payload is: 01H: Relay open

If payload is: 00H : Relay closed

2.4 Payload Decoder file

In TTN, use can add a custom payload so it shows friendly reading

In the page **Applications → Payload Formats → Custom → decoder** to add the decoder from:

[LC03 decoder](#)

2.5 Datalog Feature

Datalog Feature is to ensure IoT Server can get all sampling data from Sensor even if the LoRaWAN network is down. For each sampling, LC03 will store the reading for future retrieving purposes.

2.5.1 Ways to get datalog via LoRaWAN

With the platform downlink **07 01**, turn on the device to send None-ACK message function automatically, LC03 will wait for every upstream ACK, when there is no LoRaWAN network, LC03 will mark these records as non ack message and store the sensor data, and wait for the network to be restored (10s interval) to send all the messages.

- a) LC03 will do an ACK check for data records sending to make sure every data arrive server.
- b) When automatic sending of None-ACK messages is enabled, the LC03 will send data in CONFIRMED mode, but if an ACK is not received, the LC03 will not resend the packet, it will only mark it as a NONE-ACK message. On subsequent uplinks, if the LC03 receives an ACK, the LC03 will consider the network connected and resend all NONE-ACK messages.

2.5.2 Unix TimeStamp

LC03 uses Unix TimeStamp format based on

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

Figure 10 : DeviceTimeAns payload format

User can get this time from link: [Epoch Converter](#) :

Below is the converter example

The screenshot shows two web interfaces. On the left, EpochConverter displays the current Unix epoch time as 1611889418. On the right, Code Beautify's 'Decimal to Hex' converter shows the decimal number 1611889405 being converted to the hexadecimal value 60137afd. A red arrow points from the epoch time on the left to the decimal input on the right.

So, we can use downlink 3060137afd00 to set the current time 2021 - Jan - 29 Friday 03:03:25

2.5.3 Set Device Time

Users need to run downlink command **28 01** to enable the synchronization time.

Once LC03 Joined LoRaWAN network, it will send the MAC command (DeviceTimeReq) and the server will reply with (DeviceTimeAns) to send the current time to LC03. If LC03 fails to get the time from the server, LC03 will use the internal time and wait for next time request (The default time is once every 10 days.).

Note LoRaWAN Server need to support LoRaWAN v1.0.3(MAC v1.0.3) or higher to support this MAC command feature, Chirpstack,TTN V3 v3 and loriot support but TTN V3 v2 doesn't support. If server doesn't support this command, it will through away uplink packet with this command, so user will lose the packet with time request for TTN V3 v2 if when the automatic time synchronization function is turned on.

Downlink Command: 0x28

- Example: 0x28 01 // Automatic time synchronization Enabled
- Example: 0x28 00 // Automatic time synchronization Disable.

2.5.4 Datalog Uplink payload (FPORT=3)

The Datalog uplinks will use below payload format.

Retrieval data payload:

Size(bytes)	4	1	1	1	1
Value	Unix Time Stamp	Event	Plugstatus	Relaystatus	DatalogReply

No ACK Message: 1: This message means this payload is fromn Uplink Message which doesn't get ACK from the server before (for **Automatically send None-ACK** feature)

DatalogReply: 1: This message is a poll message reply.

Poll Message Flag is set to 1.

Each data entry is 8 bytes, to save airtime and battery, devices will send max bytes according to the current DR and Frequency bands.

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

a) DR0: max is 11 bytes so one entry of data

b) DR1: max is 53 bytes so devices will upload 4 entries of data (total 44 bytes)

c) DR2: total payload includes 11 entries of data

d) DR3: total payload includes 22 entries of data.

If devise doesn't have any data in the polling time. Device will uplink 8 bytes of 0

Example:

If user sends below downlink command: 31681D4580681D6FB005

Where : Start time: 681D4580 = time 25/5/9 08:00:00

Stop time: 681D6FB0 = time 25/5/9 11:00:00

LC03 will uplink this payload.

The screenshot displays the Dragino IoT platform interface. The top navigation bar shows 'Applications > LC系列 > End devices > LC03 > Live data'. The main content area is divided into two panels: 'Live data' and 'Event details'.

The 'Live data' panel shows a list of messages with columns for 'TIME', 'TYPE', and 'DATA PREVIEW'. The messages include 'Schedule data downlink for transmissi...', 'Forward uplink data message', and 'Successfully processed data message'. The 'DATA PREVIEW' column shows the payload data, such as 'Event: "HEARTBEAT_EVENT", PlugStatus: "PULL_OUT", RelayStatus: "CLOSE", Timestamp: 1746754043'.

The 'Event details' panel shows the full JSON payload of the selected event. The payload is a JSON object with the following structure:

```

{
  "join_eui": "AB4041090000000000",
  "dev_eui": "2608AEFF",
  "correlation_ids": {
    "gsluplink": "613750E04170NKN3F5A6123Y0"
  },
  "received_at": "2025-05-09T03:11:51.131827780Z",
  "uplink_message": {
    "session_key_id": "A2ayjPeKXJlDNI9Q8x0A==",
    "f_port": 3,
    "event": 3,
    "firm_payload": "mB1cUQ0AEB0mPhAWAQA==",
    "decoded_payload": {
      "Datalog": [
        {
          "DatalogReply": "POLL_REPLY",
          "Event": "RELAY_ACK_EVENT",
          "PlugStatus": "PULL_OUT",
          "RelayStatus": "CLOSE",
          "Timestamp": 1746754043
        }
      ],
      "LoraNode": "LC03"
    }
  },
  "zx_metadata": {
    "gateway_id": {
      "gateway_id": "m40411d1e064150",
      "eui": "AB40411D1E064150"
    },
    "time": "2025-05-09T03:11:50.768074Z",
    "ts1": "4104900307",
    "ts1": "-43",
    "channel": "vcc4"
  }
}

```

68 1D 5C 51 04 00 00 40 68 1D 6A 61 03 00 00 40

Where the first 8 bytes is for the first entry:

68 1D 5C 51 04 00 00 40

Unix time is 0x68 1D 5C 51=1746754641s=25/5/9 09:37:00

Event =0x04 = RELAY_ACK_EVENT

Plugstatus =0x00 = PULL_OUT

Relaystatus = 0x00 = Relay closed

DatalogReply = 0x40 = Represents the current data as polled data

2.6 Frequency Plans

The LC03 uses OTAA mode and below frequency plans by default. Each frequency band use different firmware, user update the firmware to the corresponding band for their country. See: [Frequency Band](#)

2.7 Firmware Change Log

Firmware download link: [Dropbox - firmware](#)

3. Configure LC03

3.1 Configure Methods

LC03 supports below configure method:

- LoRaWAN Downlink. Instruction for different platforms: See [IoT LoRaWAN Server](#) section.

3.2 General Commands

These commands are to configure:

- General system settings like: uplink interval.
- LoRaWAN protocol & radio related command.

They are same for all Dragino Devices which support DLWS-005 LoRaWAN Stack. These commands can be found on the wiki: [AT Commands](#)

3.3 Commands special design for LC03

These commands only valid for LC03, as below:

3.3.1 Set Transmit Interval Time

AT Command:

There is no AT command to set TDC time.

Feature: Change LoRaWAN End Node Transmit Interval.

Downlink Command: 0x01

Format: Command Code (0x01) followed by 3 bytes time value.

If the downlink payload=**0100003C**, it means set the END Node's Transmit Interval to 0x00003C=60(S), while type code is 01.

- Example 1: Downlink Payload: 0100001E // Set Transmit Interval (TDC)= 30 seconds
- Example 2: Downlink Payload: 0100003C // Set Transmit Interval (TDC)= 60 seconds

3.3.2 Get Device Status

Send a LoRaWAN downlink to ask device send Alarm settings.

Downlink Payload: 0x26 01

Sensor will upload Device Status via FPORT=5. See payload section for detail.

3.3.3 Clear Flash Record

AT Command:

No AT command to control relay enable or disable

Feature: Clear flash storage for data log feature.

Downlink Command: 0x08

- Example: 0x0801 // Clear all saved data in flash.

3.3.4 Relay enable or disable

AT Command:

There is no AT Command to control the Relay Output

Feature: Controls the turning on and off of appliances plugged into the socket

Downlink Command: 0x06

- Example: 0x06 01 // Relay Enabled.
- Example: 0x06 00 // Relay Disable.

3.3.5 Relay – Control Relay Output time

Feature: Controls the relay output time.

AT Command:

There is no AT Command to control the Relay Output

Downlink Payload (prefix 0x09):

0x09 aa bb cc dd ee ff // Sets relays with time control

This controls the relay output time and includes 4 bytes:

First byte : Type code (0x09)

Second byte (aa) : Inverter Mode

01: Relays will change back to their original state after a timeout.

00: Relays will change to the inverter state after a timeout.

Third byte (bb) : Control Method and Ports status:

Second Byte	Third Byte	Before Timeout Status	After Time Status
00(inverter state)	0x01	Relay NO	Relay NC
	0x00	Relay NC	Relay NO
01(original state)	0x01	Relay NO	original state
	0x00	Relay NC	

Fourth/Fifth/Sixth/Seventh bytes (cc dd ee ff) : Latching time. (Min:6 seconds ,Unit: seconds)

The device will uplink a packet if the downlink code executes successfully.

Example payload:

1. 09 01 01 00 00 00 06

Relay will be set to NC, lasting 6 seconds, then revert to their original state

2. 09 01 00 00 00 00 06

Relay will change to NC, lasting 6 seconds, and then will will change to NO.

3. 09 00 01 00 00 00 06

Relay1 will change to NO, lasting 6 seconds, and then Relay will change to NC.

4. 09 01 01 00 00 00 06

Relay will change to NO, lasting 6 seconds, and then will revert to their original state.

3.3.6 Confirmed Mode

AT Command:

There is no AT command to control whether Confirmed mode is enabled or disabled.

Feature: Mode for sending data for which acknowledgment was not received

Downlink Command: 0x06

- Example: 0x07 01 // Confirmed Mode enabled.
- Example: 0x07 00 // Confirmed Mode Disable.

3.3.7 Set the time synchronization interval

Feature: Set how often to perform time synchronization.(default is 10 days, Unit: days)

Downlink Command: 0x28

- Example: 0x28 01 // Synchronize once a day
- Example: 0x28 03 // Synchronize once every three days

4. OTA Firmware update

User can change firmware LC03 to:

- Change Frequency band/ region.
- Update with new features.
- Fix bugs.

Firmware and changelog can be downloaded from : [Firmware download link](#)

Methods to Update Firmware:

- (Recommended way) OTA firmware update via wireless: [Firmware OTA Update for LoRaWAN End Node](#)

5. FAQ

6. Order Info

Part Number: **LC03-XX**

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:

- LC03 LoRaWAN Socket

Dimension and weight:

- Device Size: cm
- Device Weight: g
- Package Size / pcs : cm
- Weight / pcs : g

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.cc.

9. FCC

FCC Caution:

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, and (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 an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator& your body.