

Connecting Dragino LPS8 indoor Gateway to Noria Chirpstack server

First log in the system

Since it has DHCP enabled by default, let's connect to our router thru a patchcord.

Look on the router ip map the corresponding IP adress

This time it is 192.168.1.125

Let's enter on the Gateway 192.168.1.125:8000

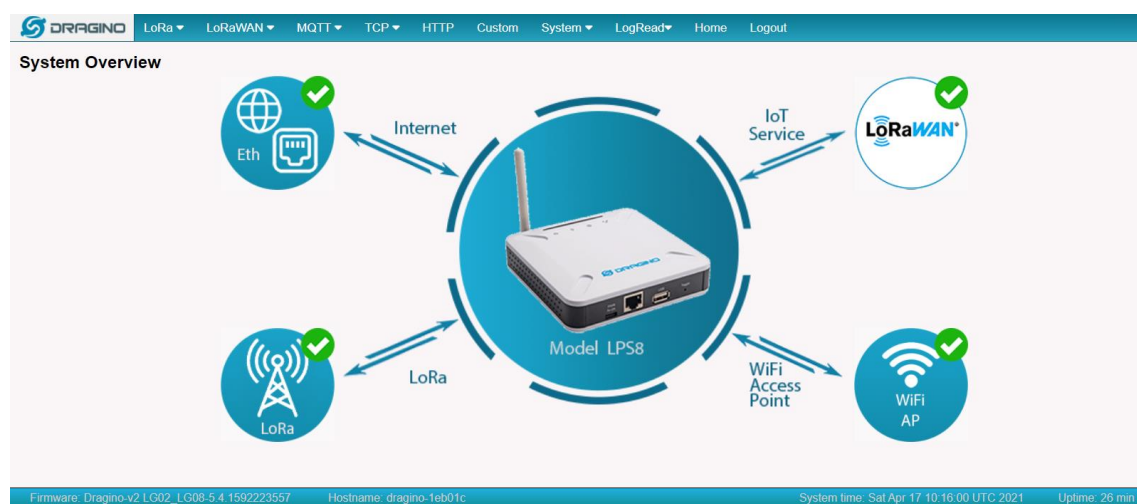
Root


Iniciar sesión

<http://192.168.1.125:8000>
Tu conexión con este sitio web no es privada

Nombre de usuario

Contraseña




LoRa ▼ LoRaWAN ▼ MQTT ▼ TCP ▼ HTTP Custom System ▼ LogRead▼ Home Logout

LoRaWAN Configuration

Server Settings

LoRaWAN Service Provider TTN-router-EU ▼
Gateway ID a840411eb01c4150
Server Port Upstream 1700
Server Port Downstream 1700
Latitude 22.705177
Longitude 114.243423
Email dragino-1eb01c@dragino.com

Packet Filter

Port Filter: 0


Save&Apply Cancel

These are the user setting to connect to TTN

But we want to connect to Noria chirpstack server

So let's change these settings

We have to select custom


LoRa ▼ LoRaWAN ▼ MQTT ▼ TCP ▼ HTTP Custom System ▼ LogRead▼ Home Logout

LoRaWAN Configuration

Server Settings


LoRaWAN Service Provider TTN-router-EU ▼
Gateway ID a840411eb01c4150
Server Port Upstream 1700
Server Port Downstream 1700
Latitude 22.705177
Longitude 114.243423
Email dragino-1eb01c@dragino.com

Packet Filter

Port Filter: 0

Save&Apply Cancel

And introduce the Noria Chirpstack server address


LoRa ▼ LoRaWAN ▼ MQTT ▼ TCP ▼ HTTP Custom System ▼ LogRead▼ Home Logout

LoRaWAN Configuration

Server Settings

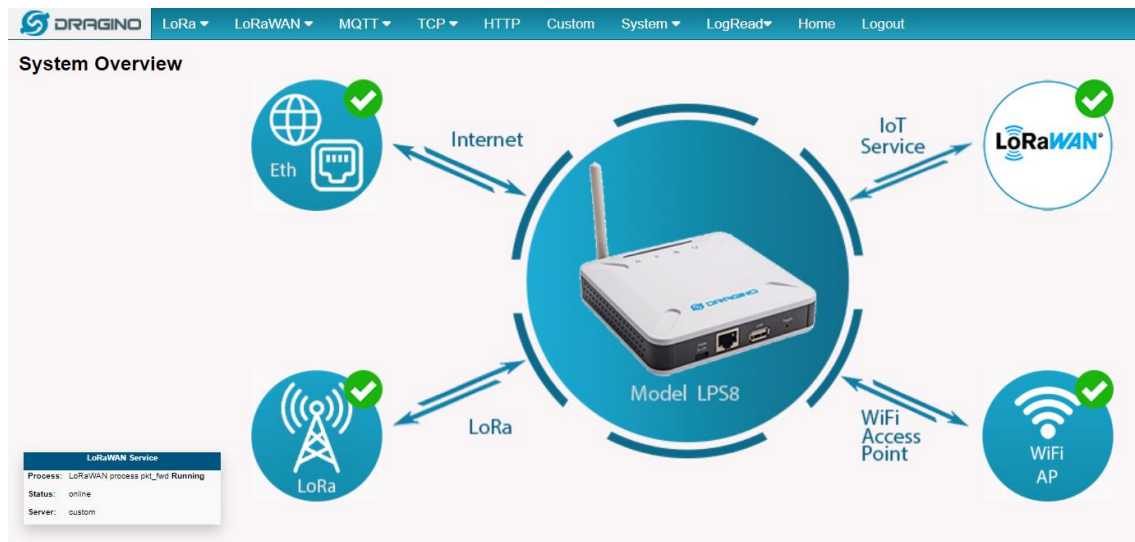
LoRaWAN Service Provider Custom ▼
Gateway ID a840411eb01c4150
Server Port Upstream 1700
Server Port Downstream 1700
Latitude 22.705177
Longitude 114.243423
Email xavier.florensa@gruponoria.com

Packet Filter

Port Filter: 0

Save&Apply Cancel

Save and Apply



Now Let's jump to The Noria Network server (Chirpstack)

Serverip address:8080

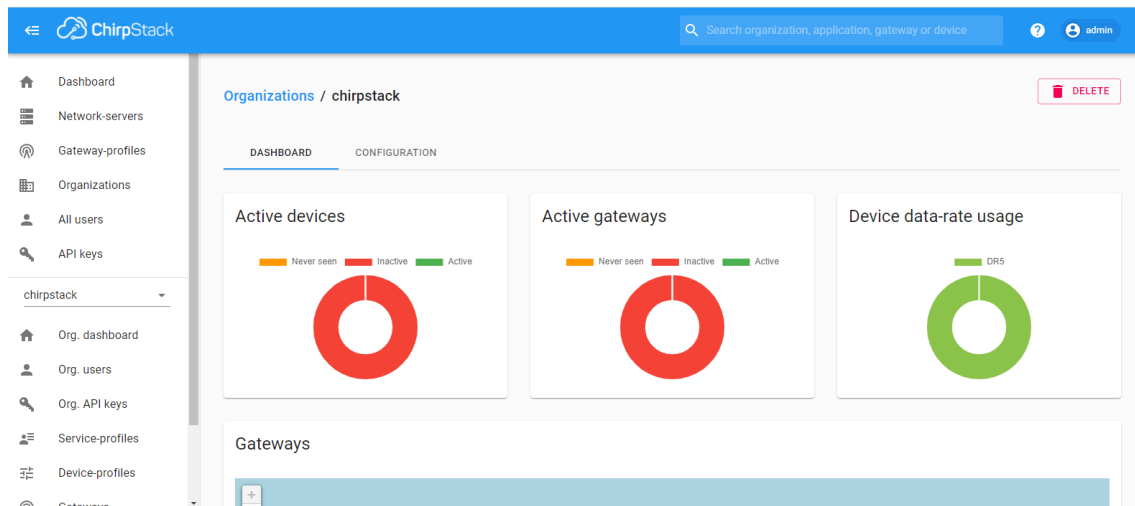
ChirpStack Login

Username / email *

Password *

LOGIN

Here we are



Now let's settle the new Gateway

Create

The 'Gateways' table contains the following data:

Last seen	Name	Gateway ID	Network server	Gateway activity (30d)
a month ago	RAKGATEWAY_xavier	b827ebfffe1bd4e1	localhost	

Rows per page: 10 | 1-1 of 1

The 'GENERAL' tab of the gateway creation form contains the following fields:

- Gateway name ***: Gateway Carles
The name may only contain words, numbers and dashes.
- Gateway description ***: Gateway Carles
- Gateway ID ***: a8 40 41 1e b0 1c 41 50

ChirpStack
Search organization, application, gateway or device
admin

Gateway-profiles
Organizations
All users
API keys
chirpstack
Org. dashboard
Org. users
Org. API keys
Service-profiles
Device-profiles
Gateways
Applications
Multicast-groups

Gateway ID: a8 40 41 1e b0 1c 41 50 MSB
Network-server *: localhost
Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.
Service-profile: IOTSERV
Select the service-profile under which the gateway must be added. The available service-profiles depend on the selected network-server, which must be selected first.
Gateway-profile: IOTGATEWAY
Optional. When assigning a gateway-profile to the gateway, ChirpStack Network Server will attempt to update the gateway according to the gateway-profile. Note that this does require a gateway with ChirpStack Concentratord.
Gateway discovery enabled
When enabled (and ChirpStack Network Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.
Gateway altitude (meters) *: 0
When the gateway has an on-board GPS, this value will be set automatically when the network has received statistics from the gateway.
Gateway location (set to current location)

ChirpStack
Search organization, application, gateway or device
admin

Gateway-profiles
Organizations
All users
API keys
chirpstack
Org. dashboard
Org. users
Org. API keys
Service-profiles
Device-profiles
Gateways
Applications
Multicast-groups

Select the service-profile under which the gateway must be added. The available service-profiles depend on the selected network-server, which must be selected first.
Gateway-profile: IOTGATEWAY
Optional. When assigning a gateway-profile to the gateway, ChirpStack Network Server will attempt to update the gateway according to the gateway-profile. Note that this does require a gateway with ChirpStack Concentratord.
Gateway discovery enabled
When enabled (and ChirpStack Network Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.
Gateway altitude (meters) *: 0
When the gateway has an on-board GPS, this value will be set automatically when the network has received statistics from the gateway.
Gateway location (set to current location)

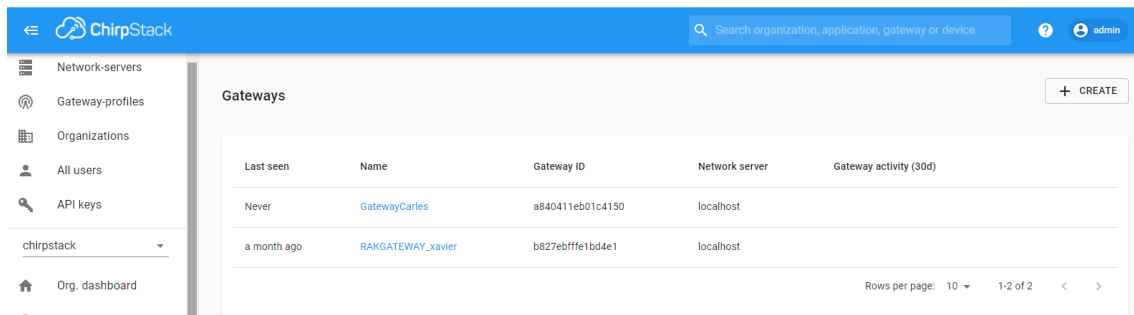
Gateway altitude (meters) *: 0
When the gateway has an on-board GPS, this value will be set automatically when the network has received statistics from the gateway.
Gateway location (set to current location)

Drag the marker to the location of the gateway. When the gateway has an on-board GPS, this value will be set automatically when the network receives statistics from the gateway.

ADD BOARD CONFIGURATION CREATE GATEWAY

And create Gateway

The Gateway is there



The screenshot shows the ChirpStack web interface. On the left is a sidebar with navigation links: Network-servers, Gateway-profiles, Organizations, All users, API keys, and a dropdown for 'chirpstack'. The main content area is titled 'Gateways' and includes a '+ CREATE' button. It contains a table with the following data:

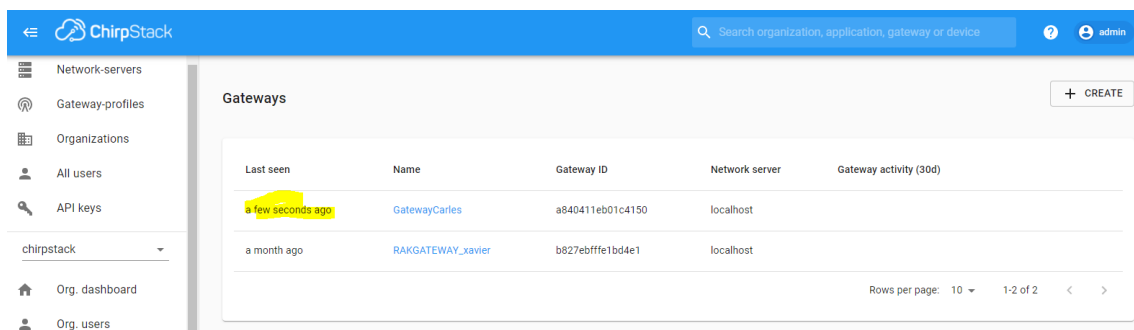
Last seen	Name	Gateway ID	Network server	Gateway activity (30d)
Never	GatewayCarles	a840411eb01c4150	localhost	
a month ago	RAKGATEWAY_xavier	b827ebfffe1bd4e1	localhost	

At the bottom right of the table, it says 'Rows per page: 10' and '1-2 of 2'.

But appears as not connected

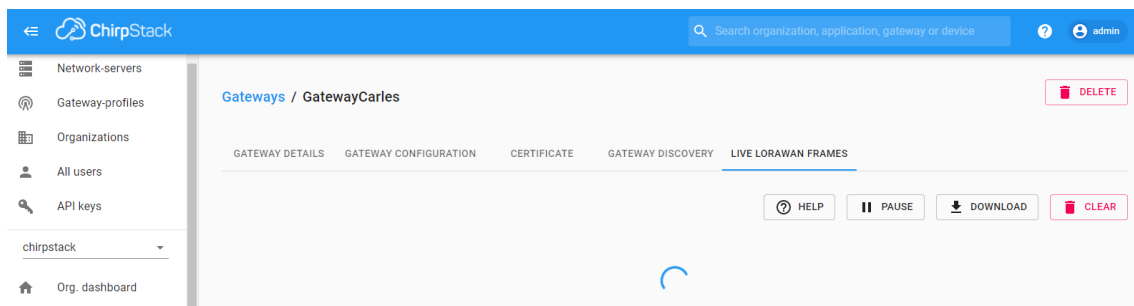
Maybe it takes some minutes

Let's connect a sensor



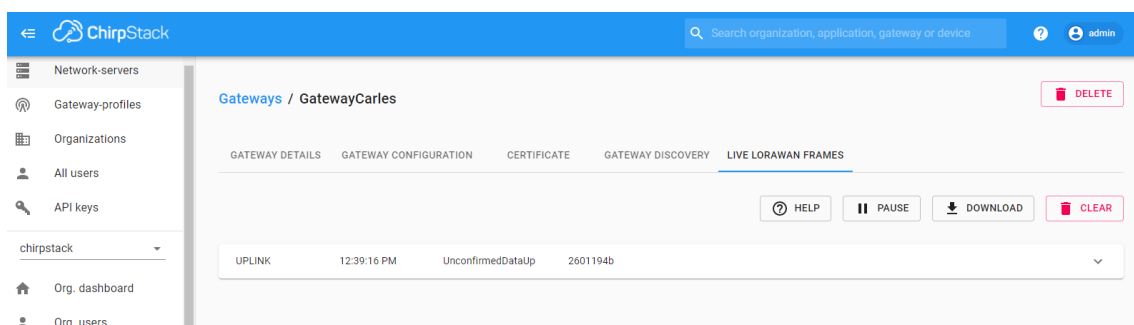
This screenshot is similar to the first one, but the 'Last seen' value for 'GatewayCarles' has been updated to 'a few seconds ago', which is highlighted with a yellow box.

Last seen	Name	Gateway ID	Network server	Gateway activity (30d)
a few seconds ago	GatewayCarles	a840411eb01c4150	localhost	
a month ago	RAKGATEWAY_xavier	b827ebfffe1bd4e1	localhost	



The screenshot shows the 'GatewayCarles' details page. The breadcrumb is 'Gateways / GatewayCarles'. There is a 'DELETE' button in the top right. Below the breadcrumb are tabs: GATEWAY DETAILS, GATEWAY CONFIGURATION, CERTIFICATE, GATEWAY DISCOVERY, and LIVE LORAWAN FRAMES (which is selected). Below the tabs are buttons: HELP, PAUSE, DOWNLOAD, and CLEAR. A blue circular loading spinner is visible at the bottom of the page.

After 40 seconds we have the first uplink



This screenshot shows the 'LIVE LORAWAN FRAMES' tab for 'GatewayCarles'. It displays a single uplink frame in a table:

UPLINK	Time	Data	Frequency
12:39:16 PM	UnconfirmedDataUp	2601194b	

The table has a dropdown arrow on the right side.

The screenshot shows the ChirpStack web interface. The left sidebar contains navigation links: Network-servers, Gateway-profiles, Organizations, All users, API keys, and a dropdown menu for 'chirpstack' with options: Org. dashboard, Org. users, Org. API keys, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area displays a LoRaWAN frame for an application named 'chirpstack'. The frame is an UPLINK received at 12:39:48 PM. The payload is 'UnconfirmedDataUp' with a device address of '2601194b'. The frame details are as follows:

```

rxInfo: { 1 item
  0: { 14 keys
    gatewayID: "a840411eb01c4150"
    time: "2021-04-17T10:39:48.025852Z"
    timeSinceGPSEPOCH: null
    rssi: -6
    loRaSNR: 8.5
    channel: 0
    rfChain: 1
    board: 0
    antenna: 0
  }
  location: { 5 keys
    latitude: 41.46742831254425
    longitude: 2.2412109375000004
    altitude: 0
    source: "UNKNOWN"
    accuracy: 0
    fineTimestampType: "NONE"
    context: "P2GqUw=="
    uplinkID: "c461ceb9-ba7e-4c98-b9a3-7fc65d3d3b14"
    crcStatus: "CRC_OK"
  }
  txInfo: { 3 keys
    frequency: 868100000
    modulation: "LORA"
    ioRaModulationInfo: { 4 keys
      bandwidth: 125
    }
  }
}
phyPayload: { 3 keys
  mhdr: { 2 keys
    mType: "UnconfirmedDataUp"
    major: "LoRaWANR1"
  }
  macPayload: { 3 keys
    fhdr: { 4 keys
      devAddr: "2601194b"
      fCtrl: { 5 keys
        adr: true
        adrAckReq: false
        ack: false
        fPending: false
        classB: false
      }
      fCnt: 4
      fOpts: null
      fPort: 1
    }
    frmPayload: { 1 item
      0: { 1 key
        bytes: "argkPQ=="
      }
    }
    mic: "12f9ab55"
  }
}

```

The application is not correctly set for ABP since we do not see any traffic on the application

Let's try to setup a OTAA application with a Dragino LSN50 v2

The screenshot shows the ChirpStack web interface for an application named 'dragino_lsn50v2'. The left sidebar is the same as the previous screenshot. The main content area shows the application configuration for 'dragino_lsn50v2'. The 'LORAWAN FRAMES' tab is selected, showing a table of frames. The table has columns: DOWNLINK, UPLINK, Time, Payload, and Device Address. The frames are as follows:

Direction	Time	Payload	Device Address
DOWNLINK	1:12:16 PM	UnconfirmedDataDown	00bade59
UPLINK	1:12:16 PM	UnconfirmedDataUp	00bade59
DOWNLINK	1:12:04 PM	UnconfirmedDataDown	00bade59
UPLINK	1:12:04 PM	UnconfirmedDataUp	00bade59

We have only entered the App key

After first join request, the device is already on the system.

Applications / dragino_Isn50v2 / Devices / dragino

DETAILS

CONFIGURATION

KEYS (OTAA)

ACTIVATION

DEVICE DATA

LORAWAN FRAMES

Device address *

01 a1 e4 44

While any device address can be entered, please note that a LoRaWAN compliant device address consists of an AddrPrefix (derived from the NetID) + NwkAddr.

Network session key (LoRaWAN 1.0) *

ab 17 bb 3e 21 89 ad 3b 03 9a 54 e4 2f 61 db 57

Application session key (LoRaWAN 1.0) *

b5 cf 50 49 ec 87 8b 3e f9 8e 5f a1 35 db 2d 45

Uplink frame-counter *

39

Downlink frame-counter (network) *

39

The device address changes on each join request

UPLINK

1:15:16 PM

UnconfirmedDataUp

00bade59

▼ rxInfo: {} 1 item

▼ 0: {} 14 keys

gatewayID: "a840411eb01c4150"

time: "2021-04-17T11:15:16.268450Z"

timeSinceGPSEPOCH: null

rssI: -48

loRaSNR: 9.5

channel: 1

rfChain: 1

board: 0

antenna: 0

▼ location: {} 5 keys

latitude: 41.46742831254425

longitude: 2.2412109375000004

altitude: 0

source: "UNKNOWN"

accuracy: 0

fineTimestampType: "NONE"

context: "vjuV9A=="

uplinkID: "3a5733bb-5633-4ff8-9171-d5bc5fc9c6ac"

crcStatus: "CRC_OK"

▼ txInfo: {} 3 keys

frequency: 868300000

modulation: "LORA"

▼ loRaModulationInfo: {} 4 keys

bandwidth: 125

spreadingFactor: 12

codeRate: "4/5"

▼ phyPayload: {} 3 keys

▼ mhdr: {} 2 keys

mType: "UnconfirmedDataUp"

major: "LoRaWANR1"

▼ macPayload: {} 3 keys

▼ fhdr: {} 4 keys

devAddr: "00bade59"

▼ fCtrl: {} 5 keys

adr: true

adrAckReq: false

ack: false

fPending: false

classB: false

fCnt: 23

fOpts: null

fPort: 2

▼ frmPayload: {} 1 item

▼ 0: {} 1 key

bytes: "OWFptwgrwHpOIUg="

mic: "b65ad679"

There is no data

Since we have to look at the data tab

Applications / dragino_Isn50v2 / Devices / dragino

DETAILS


CONFIGURATION

KEYS (OTAA)

ACTIVATION

DEVICE DATA

L

 HELP

1:19:16 PM

up

1:19:04 PM

up

```
applicationID: "3"
applicationName: "dragino_Isn50v2"
deviceName: "dragino"
devEUI: "a84041135182a3b1"
▼ rxInfo: [] 1 item
▼ 0: {} 14 keys
  gatewayID: "a840411eb01c4150"
  time: "2021-04-17T11:19:04.271566Z"
  timeSinceGPSEPOCH: null
  rssi: -48
  loRaSNR: 8.8
  channel: 5
```

```

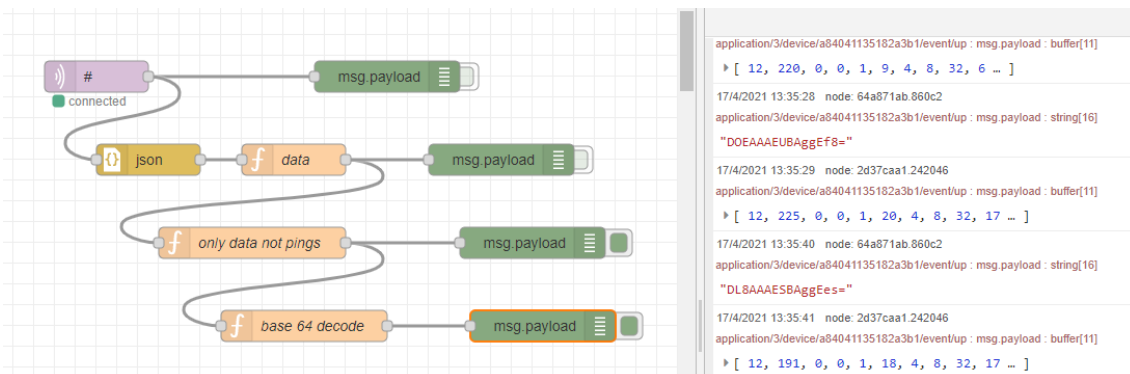
source: "UNKNOWN"
accuracy: 0
fineTimestampType: "NONE"
context: "y9KqPA=="
uplinkID: "7d126556-3634-44ea-9041-b288a0aa4c1b"
crcStatus: "CRC_OK"
▼ txInfo: {} 3 keys
  frequency: 867500000
  modulation: "LORA"
  ▼ loRaModulationInfo: {} 4 keys
    bandwidth: 125
    spreadingFactor: 12
    codeRate: "4/5"
    polarizationInversion: false
adr: true
dr: 0
fCnt: 42
fPort: 2
data: "DNwAAAERBATECFo="
objectJSON: ""
tags: {} 0 keys
confirmedUplink: false
devAddr: "00bade59"

```

This data is Base64 encoded

DNwAAAERBATECFo=

But we can decode the payload



"DM4AAAERBALkCgI="

17/4/2021 13:45:05 node: 2d37caa1.242046

application/3/device/a84041135182a3b1/event/up : msg.payload : buffer[11]

▼ *buffer[11]* raw

▼ [0 ... 9]

0: 0xc
1: 0xce
2: 0x0
3: 0x0
4: 0x1
5: 0x11
6: 0x4
7: 0x2
8: 0xe4
9: 0xa
▼ [10 ... 10]
10: 0x2

Edit mqtt in node

Delete

Car

⚙ Properties

🌐 Server	localhost:1883
📋 Topic	#
⚙ QoS	2 ▼
➡ Output	auto-detect (string or buffer)
🏷 Name	Name

Edit json node

Delete

Cancel

Properties

 Action Always convert to JavaScript Object

 Property msg. payload

 Name Name

Edit function node

Delete

Properties

 Name data

Setup

Function

```
1 var datastring = msg.payload.data
2 msg.payload = datastring
3 return msg;
```

Edit function node

Delete

⚙️ Properties

🔑 Name

only data not pings

Setup

Function

```

1 if(typeof msg.payload !== 'undefined') {
2     return msg;
3 }
4 else
5 {}

```

Edit function node

Delete

⚙️ Properties

🔑 Name

base 64 decode

Setup

Function

```

1 var b = new Buffer (msg.payload,'base64');
2 msg.payload = b;
3 return msg;

```

The Device Address changes on each join request to Chirpstack

So if we have a device sending to a chirpstack Gateway and then we also plug a TTN Gateway, the device will no longer be visible from the TTN application since it has changed the device address.

To be visible from the TTN application we have to stop the Chirpstack Gateway

Let's compare to a typical payload on TTN

▲ 13:58:33

1

2

payload: 0CBB 00 00 01 14 04 02 26 0E 1E

ADC_CH0V: 0.276

BatV: 3.259

Digital_IStatus: "L"

Distance_cn

"DM4AAAERBALkCgI="

17/4/2021 13:45:05 node: 2d37caa1.242046

application/3/device/a84041135182a3b1/event/up : msg.payload : buffer[11]

▼ *buffer[11]* raw

▼ [0 ... 9]

0: 0xc
1: 0xce
2: 0x0
3: 0x0
4: 0x1
5: 0x11
6: 0x4
7: 0x2
8: 0xe4
9: 0xa
▼ [10 ... 10]
10: 0x2

Payload

0C C3 00 00 01 0E 04 04 1A 0A 3E



Fields

```
{  
  "ADC_CH0V": 0.27,  
  "BatV": 3.267,  
  "Digital_IStatus": "L",  
  "Distance_cm": 105,  
  "Distance_signal_strength": 2622,  
  "Door_status": "OPEN",  
  "EXTI_Trigger": "FALSE",  
  "TempC1": 0,  
  "Work_mode": "Distance"
```

So payload are 11 bytes

And same we have received on Chirpstack

Starting with 0C

Now let's try to decode the payload with node-red

We try to copy the payload decoder used in TTN

```
function Decoder(bytes, port) {
```

```

var mode=(bytes[6] & 0x7C)>>2;
var decode = {};
if(mode!=2)
{
    decode.BatV=(bytes[0]<<8 | bytes[1])/1000;
    decode.TempC1= parseFloat((((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));
    decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;
    decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
    if(mode!=6)
    {
        decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
        decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
    }
}

if(mode=='0')
{
    decode.Work_mode="IIC";
    if((bytes[9]<<8 | bytes[10])===0)
    {
        decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
    }
    else
    {
        decode.TempC_SHT=parseFloat((((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
        decode.Hum_SHT=parseFloat((((bytes[9]<<8 | bytes[10])/10).toFixed(1));
    }
}
else if(mode=='1')
{
    decode.Work_mode=" Distance";
}

```

```

decode.Distance_cm=parseFloat((((bytes[7]<<8 | bytes[8])/10) .toFixed(1));
if((bytes[9]<<8 | bytes[10])!=65535)
{
    decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]) .toFixed(0));
}
}
else if(mode=='2')
{
    decode.Work_mode=" 3ADC";
    decode.BatV=bytes[11]/10;
    decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
    decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
    decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
    decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
    decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
    decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
    if((bytes[9]<<8 | bytes[10])===0)
    {
        decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
    }
    else
    {
        decode.TempC_SHT=parseFloat((((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
        decode.Hum_SHT=parseFloat((((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
    }
}
else if(mode=='3')
{
    decode.Work_mode="3DS18B20";
    decode.TempC2=parseFloat((((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
    decode.TempC3=parseFloat((((bytes[9]<<8 | bytes[10])/10) .toFixed(1));

```



```

}

else if(mode=='4')
{
    decode.Work_mode="Weight";
    decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
}

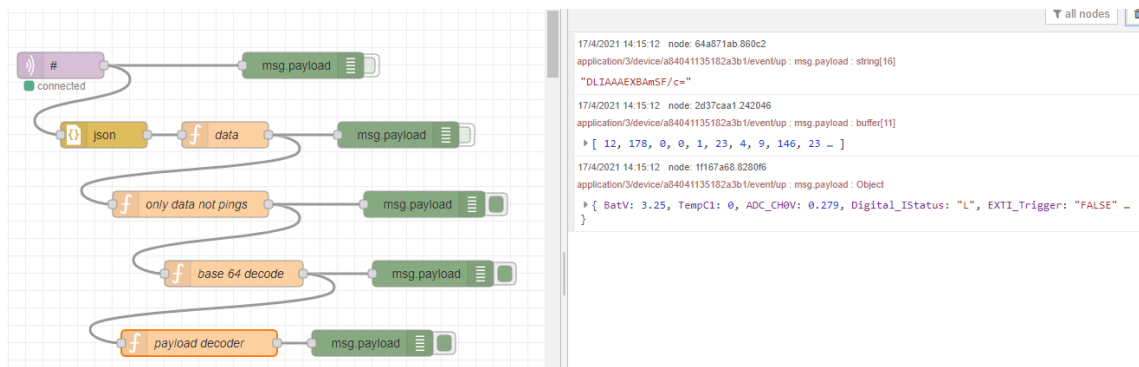
else if(mode=='5')
{
    decode.Work_mode="Count";
    decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
}

}

if((bytes.length==11) || (bytes.length==12))
{
    return decode;
}

}

```



You can find the code here

<https://github.com/xavierflorensa/Chirpstack-Gateway-to-Node-red-local/blob/master/node-red%20chirpstack%20payload%20decoder%20dragino%20lsn50v2.txt>

Yes, with this decoder

▼ *object*

BatV: 3.246
TempC1: 0
ADC_CH0V: 0.269
Digital_IStatus: "L"
EXTI_Trigger: "FALSE"
Door_status: "OPEN"
Work_mode: " Distance"
Distance_cm: 245
Distance_signal_strength: 6146

Edit function node

Delete

⚙ Properties

📁 Name payload decoder

Setup

Function

Close

```
1 var bytes =msg.payload;
2
3 //function Decoder(bytes, port) {
4 var mode=(bytes[6] & 0x7C)>>2;
5 var decode = {};
6 if(mode!=2)
7 {
8   decode.BatV=(bytes[0]<<8 | bytes[1])/1000;
9   decode.TempC1= parseFloat(((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));
10  decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;
11  decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
12  if(mode!=6)
13  {
14    decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
15    decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
16  }
17 }
18
19 if(mode=='0')
20 {
21   decode.Work_mode="IIC";
22   //if(mode=='0') decode.Distance_cm=0;
23   //if(mode=='0') decode.Distance_signal_strength=0;
24 }
```

```

21 decode.Work_mode=" 110 ";
22 if((bytes[9]<<8 | bytes[10])===0)
23 {
24     decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
25 }
26 else
27 {
28     decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
29     decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
30 }
31 }
32 else if(mode=='1')
33 {
34     decode.Work_mode=" Distance";
35     decode.Distance_cm=parseFloat(((bytes[7]<<8 | bytes[8])/10).toFixed(1));
36     if((bytes[9]<<8 | bytes[10])!=65535)
37     {
38         decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]).toFixed(0));
39     }
40 }
41 else if(mode=='2')
42 {
43     decode.Work_mode=" 3ADC";
44     decode.BatV=bytes[11]/10;
45     decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
46     decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
47     decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
48     decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
49     decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
50     decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
51     if((bytes[9]<<8 | bytes[10])===0)
52     {
53         decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
54     }
55     else
56     {
57         decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
58         decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
59     }
60 }
61 else if(mode=='3')
62 {
63     decode.Work_mode="3DS18B20";
64     decode.TempC2=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
65     decode.TempC3=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
66 }
67 }
68 else if(mode=='4')
69 {
70     decode.Work_mode="Weight";
71     decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
72 }
73 else if(mode=='5')
74 {
75     decode.Work_mode="Count";
76     decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);

```

```

76 decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
77 ^ }
78
79 if((bytes.length==11) || (bytes.length==12))
80 ^ {
81 msg.payload=decode;
82 return msg;
83 //return decode;
84
85 ^ }
86 ^ //}
87

```

var bytes =msg.payload;

//function Decoder(bytes, port) {

var mode=(bytes[6] & 0x7C)>>2;

var decode = {};

if(mode!=2)

{

decode.BatV=(bytes[0]<<8 | bytes[1])/1000;

decode.TempC1= parseFloat((((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));

decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;

decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";

if(mode!=6)

{

decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";

decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";

}

}

if(mode=='0')

{

decode.Work_mode="IIC";

if((bytes[9]<<8 | bytes[10])===0)

{

decode.Illum=(bytes[7]<<24>>16 | bytes[8]);

}

else

```

{
    decode.TempC_SHT=parseFloat((((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
    decode.Hum_SHT=parseFloat((((bytes[9]<<8 | bytes[10])/10).toFixed(1));
}
}
else if(mode=='1')
{
    decode.Work_mode=" Distance";
    decode.Distance_cm=parseFloat((((bytes[7]<<8 | bytes[8])/10) .toFixed(1));
    if((bytes[9]<<8 | bytes[10])!=65535)
    {
        decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]) .toFixed(0));
    }
}
else if(mode=='2')
{
    decode.Work_mode=" 3ADC";
    decode.BatV=bytes[11]/10;
    decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
    decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
    decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
    decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
    decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
    decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
    if((bytes[9]<<8 | bytes[10])===0)
    {
        decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
    }
    else
    {
        decode.TempC_SHT=parseFloat((((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
    }
}

```

```

    decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
  }
}
else if(mode=='3')
{
    decode.Work_mode="3DS18B20";
    decode.TempC2=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
    decode.TempC3=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));

}
else if(mode=='4')
{
    decode.Work_mode="Weight";
    decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
}
else if(mode=='5')
{
    decode.Work_mode="Count";
    decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
}

if((bytes.length==11) || (bytes.length==12))
{
    msg.payload=decode;
    return msg;
    //return decode;

}
//}

```

