

TEKTELIC Communications Inc. 7657 10th Street NE Calgary, Alberta Canada, T2E 8X2

# Embedded LNS Gateway Instruction

# Contents

1.	Intro	oduction	3
2.	Con	figure Embedded LNS	4
3.	Арр	lication Specific Configurations	9
	3.1	Connecting to Modbus TCP/IP	9
	3.2	Send Decoded Uplink to External MQTT Broker	. 11
	3.3	Connecting to BACnet IP	. 12
	3.4	Get Decoded Uplinks from Chirpstack LNS to Node Red	. 14
4.	Trou	ubleshooting	. 17
	4.1	Sending Downlinks to Embedded LNS	. 17
	4.2	Configure a Modbus Client	. 19
	4.3	View Uplink Messages	. 21

## **1. Introduction**

Tektelic Embedded LNS Gateways simplify the integration of LoRaWAN devices with industrial protocols such as MQTT, Modbus TCP/IP, and BACnet IP.

Depending on the intended use case, the configuration steps for these gateways may vary. To start, ensure that all components are correctly set up as follows:

- 1. Initial Setup (<u>Section 2</u>):
  - Add Gateways, Devices, and Codecs in the Embedded LNS platform.
  - Enable specific protocol using Kona Link.
- 2. Protocol-Specific Configuration: Use Node-RED to configure and manage the supported protocols:
  - Modbus TCP/IP: Refer to <u>Section 3.1</u> for detailed steps.
  - MQTT: Refer to <u>Section 3.2</u> for configuration instructions.
  - BACnet/IP: Refer to <u>Section 3.3</u> for setup guidelines.

You can see some troubleshooting tips specific for the use cases in Section 4

## 2. Configure Embedded LNS

- 1. Login to Kona Link web page using "Host Name" or "IP Address":
  - Using "Host Name" Host Name URL: https://kona-<GW variant>-<last 6 digit GW ID>.local/ Eg: https://kona-micro-0011ab.local/
  - Using "IP Address" IP Address URL: https://<GW IP Address>/ Eg: https://192.0.2.111/
- 2. Log into Embedded LNS Network Server by **Embedded LNS** link in Kona Link Embedded LNS link in Kona Link.

Embedded LNS
Use the links below to access the tools and configure your Industrial Gateway for your specific needs.    Embedded LNS
Manage LoRaWAN devices and connect them to the BACnet or Modbus applications.
Node-RED Web UI
Build and manage automation flows for BACnet and Modbus using visual programming in Node- RED.

3. Navigate to Gateway section and click Add Gateway.

<ul> <li>Network Server</li> </ul>	Gateways					Add gateway Selected gateways
② Dashboard		Last seen	Gateway ID	Name	Region ID	Region common-name
습 Tenants 유 Users	• Online	2025-01-30 11:45:41	647fdafffe005841	Micro-647FDAFFE005841	us915_0	US915
🔎 API Keys						< 1 > 10/page >
Device Profile Templates						
Regions						
🟠 Tenant						
② Dashboard						
A Users						
P API Keys						
Device Profiles						
👳 Gateways	-					
🍘 Gateway Mesh						
Applications						

4. Provide the **name** and **Gateway ID**(can be found on the KONA Link home page) and click submit.

General Tags Metadata	
* Name	
Description	
* Gsteway ID (EUI64)	* Stats interval (secs) (0)
Location	-
* -	I Leafet   @ OpenStreetMap contributors
Submit	

5. Navigate to **Device Profile** section.

් Dashboard බ Tenants	Dashboard Configuration		
A Users A API Keys	Active devices	Active gateways	Device data-rate usage
Device Profile Templates	Never seen Inactive Active	Never seen Offline Offline Online	DR0 DR2 DR3 DR5
Regions			
🙆 Tenant			
② Dashboard			
A Users			
Device Profiles			
🗢 Gateways			

#### 6. Press Add Device Profile.

Tenants / ChirpStack / Device profiles Device profiles					-	Add device profile
Name	Region	MAC version	Revision	Supports OTAA	Supports Class-B	Supports Class-C
EU868 Test Profile Comfort	EU868	LoRaWAN 1.0.2	RP002-1.0.2	yes	no	no
US915	US915	LoRaWAN 1.0.2	RP002-1.0.2	yes	no	no
load test	EU868	LoRaWAN 1.0.2	RP002-1.0.2	no	no	no

- 7. Configure device profile parameters:
  - Name after the device type
  - Enter Region and Regional configuration based on gateway
  - Enter MAC version and regional parameters revision appropriate for your device. Default MAC version is LoRaWAN 1.0.2 and Regional parameters revision as RP002-1.0.2. For Tektelic devices on LoRaMAC version 1.0.4 such as Breeze and eDoctor, the MAC version should be set to LoRaWAN 1.0.4 and Regional parameters revision as RP002-1.0.4.
  - Set ADR algorithm as Default ADR
  - Leave other fields as default

General	Join (OTAA / ABP)	Class-B	Class-C	Codec	Relay	Tags	Measurements		Select device-profile template
* Name									
		_ <							
Descriptio	n								
* Region								Region configuration @	6
EU868							~		~ )
* MAC ver	sion ⊘							* Regional parameters revision ⑦	
LoRaWA	N 1.0.3							A	~ )
* ADR algo	orithm ②								
Default /	ADR algorithm (LoRa or	nly)							~ )

8. In Codec tab choose JavaScript functions in Payload codec.

EU868 Test Profile Comfort device profile id: 82e332e4-6aed-4168-92e2-a6d1fc6100f0	Delete device profile
General Join (OTAA / ABP) Class-B Class-C Codec Reterments	Select device-profile template
Payload codec 💿	
JavaScript functions	× ]
Codec functions	
Insert Decoder here	

9. Contact your device vendor to obtain the decoder script. For Tektelic devices, go to LoRa Alliance Payload Codec 1.0.0 folder in Tektelic GitHub (LoRa Alliance format).

무 TektelicCommunications / data-converters (Public)										
<> Code	⊙ Issues 3	រ៉ា Pull requests		🛈 Security 🗠 Insights						
			🐉 master 👻 🐉 2 Branches	🛇 0 Tags	Q Go to file	<> Code -				
			* kschluff-tektelic Merge pull	request #5 from TektelicCommunications/seal-v1.0	7c7ef2d · 8 months ago	🕚 39 Commits				
			Encoders	add seal 1.0 converters		8 months ago				
			🖿 LoRa Alliance Payload Codec	1.0.0 Seal 1.0 converters		8 months ago				
			TTN v2	add seal 1.0 converters		8 months ago				

• Choose the folder named according to your device type

Name	Last commit message	Last commit date
🖿		
agro 🖿	Replaced US916 with US915	last year
aura-flux	Added separate folder for Tektelic v2	last year
breeze	Replaced US916 with US915	last year
comfort-vivid	Replaced US916 with US915	last year

• Copy the code and insert it into Embedded LNS Codec functions window



#### 10. Navigate to the Application section.

<ul> <li>Dashboard</li> <li>Tenants</li> </ul>	Dashboard Configuration		
A Users API Keys	Active devices	Active gateways	Device data-rate usage
Device Profile Templates	Never seen Inactive Active	Never seen Offline Offline	DR3
Regions     Tenant			
② Dashboard			
R Users			
Device Profiles			
🗢 Gateways 🎯 Gateway Mesh			
Applications			

11. Create a new application or pick an existing one. You need application for each type of device so it's recommended to name application after a device type.

Applications	Add appli	cation
Name	Description	
Your Application Name	-	

12. Press Add Device.

Comfort Sensor Test application id:	cf636d8a-2785-4a4e-a10e-e6c91436996	d		Delete application
Devices Multicast groups Relays	Application configuration Integr	ations		
				Add device Selected devices
Last seen	DevEUI	Name	Device profile	Battery
Never	8718a2d7f84a5c62	Comfort EU868 Device Test	EU868 Test Profile Comfort	

- 13. Configure device parameters:
  - Name
  - DevEUI (Device Unique Identifier)
  - JoinEUI (Join Server Identifier)
  - Device profile (chose existing from the dropdown list)
  - Application Key (on submit)

Device Tags Variables				
* Name				
Comfort EU868 Device Test				
Description				
				6
* Device EUI (EUI64)	Join EUI (EUI64) 💿			
8718a2d7f84a5c62	8718a2d7f84a5c62	MSB $\vee$	С	٥
* Device profile				
EU868 Test Profile Comfort				V
Device is disabled ⑦	Disable frame-counter validation ⑦			
Submit				

14. Press **Submit**. Before moving on to Protocol Configurations in Node Red, Gateways, Devices and Codec should be registered and configured correctly. And Device should join and start sending uplinks successfully.

# **3. Application Specific Configurations**

## 3.1 Connecting to Modbus TCP/IP

**NOTE:** Before moving on to Modbus Configurations in Node Red, Gateways, Devices and Codec should be registered and configured correctly. And Device should join and send uplinks successfully.

- 1. Login to Kona Link web page using "Host Name" or "IP Address":
  - Using "Host Name" Host Name URL: https://kona-<GW variant>-<last 6 digit GW ID>.local/ Eg: https://kona-micro-0011ab.local/
  - Using "IP Address" IP Address URL: https://<GW IP Address>/ Eg: https://192.0.2.111/
- 2. In SET UP WIZARDS section use Modbus Set Up to enable Modbus



3. Follow the Node-RED Web UI link in Embedded LNS subsection in Kona Link



4. In a new flow go to menu and select Import

Node-RED					
Q filter nodes	Flow 1	+ -	i inf 4	Edit	
~ common		Î	4	View Arrange	
⇒ inject •				Import Export	ctrl-i ctrl-e

5. Go to section Local and select Modbus Server flow:

Import nodes	
Clipboard	✓
	✓ ♥ flows
Local	BACnetServer.json
Examp	🗅 GetUplinks.json
	ModbusClient.json
	C ModbusServer.json
_	🗅 MQTT.json
	C SendDownlinks.json
Import to current	t flow new flow
	Cancel Import

6. In the Modbus Server flow and in the **Get Uplinks from Embedded LNS** node, configure the following. Set the subscription topic to match the one being used by your LoRaWAN device (e.g., device uplinks) or leave default.

Flow 1	Send Downlinks	Modbus Server	Edit mqtt in no	de	
For more infor	mation on this sample flow, click on the t	pelow comment then the Info icon on	Delete		Cancel Done
			Properties		•
			@ Server	localhost:1883	✓
Modbus Serv			Action	Subscribe to single topic	~
) Get Uplinks fr	om Embedded LNS	ss Device Event Payload [up]	- 🔁 🖀 Topic	application/+/device/+/event/+	
			@ QoS	0 ~	

7. In Mapping Uplink Fields to Modbus Register Addresses node click on the On Start tab.

Moddbus Server       Cancel       Done         O Properties       Image: Concernent of the second of	Flow 1	Modbus Server	Ser	Edit function no	de						
O Properties Image: Spit by Uplink Field Check if Object is not empty Modbus Field Modbus Field O Properties Image: Spit by Uplink Field Image: Spit	Modbus Seru	•		Delete				Cancel		Don	e
	Modolas Sci vi			© Properties					¢		121
O Setup       On Start         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field         Image: Spirit by Uplink Field       Image: Spirit by Uplink Field <t< td=""><td>Get Uplinks fr</td><td>rom Chirpstack LNS</td><td>ocess Devio</td><td>Name</td><td>Mapping</td><td>Uplink Fields</td><td>to Modbus Register Addre</td><td>esses</td><td></td><td><i>R</i> •</td><td>•</td></t<>	Get Uplinks fr	rom Chirpstack LNS	ocess Devio	Name	Mapping	Uplink Fields	to Modbus Register Addre	esses		<i>R</i> •	•
<pre>// Only edit the the below list of objects/mapping variable // An example object is provided below const mapping = [ // An example object is provided below const mapping = [ // An example object is provided below const mapping = [ // An example object is provided below const mapping = [ // An example object is provided below const mapping = [ // An example object is provided below const mapping = [ // Deteul': "647fda00001eca0", // Deteul': "647fda000001eca0", // Tetribute": "extconnector_count", // Deteul': "647fda000001eca0", // Deteul': "647fda000001eca0", //</pre>				Ø Setup		On Start		On Stop			
Mapping Uplink Fields to Modbus Register Addresses	Cneck if O	Dink Field Check if Uss	er Mapping s Ibus Flex Wr	1 // ( 2 // / 4 3 con: 4 con: 5 6 7 7 8 9 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 25 26	<pre>inly edit in example t mapping { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "DevE "Attr "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "Data "Data "Data "Data "addr }, { "Data "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "Data "Data "Data "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "addr }, { "Data "Data "addr }, { "Data "Addr }, { "Data</pre>	the the bell object is ; = [ UUT: "647fd ibute": "ha JType": "int "ess": 40001 UUT: "647fd ibute": "ee 'ibute": "ee 'ibute": "ree 'ibute": "ree 'ibute": "ree 'ibute": "ree 'ibute": "an UUT: "647fd ibute": "an 'ibute": an 'ibute": an 'ibute	ow list of objects/ma provided below a000001eca0", ll_effect_count", , , a0000001eca0", lative_humidity", at", a000001eca0", bient_temperature", at",	pping variable	A		

- 8. Add a mapping object for all possible uplink fields for each and every device you would like. The four attributes required for each mapping object are as follows:
  - DEVEUI: The device EUI for a given device
  - Attribute: The uplink field name exactly as it is written in the decoded uplink object.
  - DataType: Choose between float (decimal number), int (16-bit unsigned integer) and int32 (32-bit signed integer). Please note that floats and int32 require 2 contiguous registers. So, for example, if you have an uplink field that is a float assigned to register 40001, then both 40001 and 40002 register addresses are unavailable.
  - address: The Modbus register address that you would like the above specified attribute to write to.
- 9. Deploy the changes. The data from the uplink should now be available in Modbus format.

## 3.2 Send Decoded Uplink to External MQTT Broker

- 1. Login to Kona Link web page using "Host Name" or "IP Address":
  - Using "Host Name" Host Name URL: https://kona-<GW variant>-<last 6 digit GW ID>.local/ Eg: https://kona-micro-0011ab.local/
  - Using "IP Address" IP Address URL: https://<GW IP Address>/ Eg: https://192.0.2.111/
- 2. Follow the Node-RED Web UI link in Embedded LNS subsection in Kona Link



3. In a new flow go to menu and select Import

Node-RED		
Q filter nodes	Flow 1	+ - i Inf • Edit
common		View     Arrange
⇒ inject		Flo import ctrl
o debug		Export ctri-

4. Go to section Local and select **MQTT.json** flow for your application:

Im	port nodes	
	Clipboard	✓
		✓ ♥ flows
	Local	BACnetServer.json
	Examp	🗅 GetUplinks.json
		ModbusClient.json
		D ModbusServer.json
	_	🗅 MQTT.json
		C SendDownlinks.json
	Import to current	flow new flow
		Cancel Import

5. In **Send External MQTT Broker** node in Server field enter the address of the MQTT broker you wish to use

Modbus Server	Modbus Client	Ø Get Uplinks	Ø Send Downlinks	Ø MQTT	Ø BAC	Edit mqtt out r	node			
						Delete			Cancel	Done
, click on the below commer	it then the into icon on the	top right and view the expl	anation on the bottom right	of the flow and what	needs to be do	© Properties	•		<	
				-		Server	broker.hivem	q.com:1883	~	
				+		📰 Торіс	application/ex	_app/device/ex_dev	//event/ex_eve	ent
Process Device Eve	nt Payload (up)	Get Decoded Uplink	Send to Ex	ternal MQTT Broker		🛞 QoS	0 ~	Retain	false	~
						Name Name	Send to Exten	nal MQTT Broker		
	4	Uplink with Decoded U	plink Object 🖌 📃 🔲			Tip: Leave t properties.	opic, qos or retain	blank if you want to	set them via r	nsg

6. Deploy the flow.

#### 3.3 Connecting to BACnet IP

**NOTE:** Before moving on to BACnet Configurations in Node Red, Gateways, Devices and Codec should be registered and configured correctly. And Device should join and start sending uplinks successfully.

- 1. Login to Kona Link web page using "Host Name" or "IP Address":
  - Using "Host Name" Host Name URL: https://kona-<GW variant>-<last 6 digit GW ID>.local/ Eg: https://kona-micro-0011ab.local/
  - Using "IP Address" IP Address URL: https://<GW IP Address>/ Eg: https://192.0.2.111/

2. In SET UP WIZARDS section use BACnet Set Up to enable the BACnet server



3. Follow the Node-RED Web UI link in Embedded LNS subsection in Kona Link

Y KONA Link	Онома Диоме Биакол
SET UP WIZARDS LoRa Basics Sation Set Up Chirpstack Set Up Gennic UDP Set Up BAChert Set Up	Embedded LNS
Motious Set Up  NTWORK Cellular Settings OpenVPN  IPace	Use the links below to access the tools and configure your Industrial Gateway for your specific needs.  Embedded LNS Manage LoRaWAN devices and connect them to the BACnet or Modbus applications.
PYSTEM Change Patsword SHMP Settings Upgrade Firmware Embrodded UKS	Build and manage automation flows for BACnet and Modbus using visual programming in Node- RED.

4. In a new flow go to menu and select Import

Node-RED		
Q filter nodes	Flow 1	+ - i Inf Edit
common		View     Arrange
🗧 debug		Flo Import ctrl-i b ctrl-e

5. Go to section Local and select **BACnetServer** flow for your application and click **Import** 

Clipboard	<ul> <li>✓ If flows</li> </ul>
Local	BACnetServer.json
	GetUplinks.json
	ModbusClient.json
	🗋 ModbusServer.json
	MQTT.json
	C SendDownlinks.json
Import to curre	nt flow new flow



6. In BACnet Server flow press the BACnet Server information node to get extra instruction

7. After implementing needed changes deploy flow

#### 3.4 Get Decoded Uplinks from Chirpstack LNS to Node Red

- 1. Login to Kona Link web page using "Host Name" or "IP Address":
  - Using "Host Name"

Host Name URL: https://kona-<GW variant>-<last 6 digit GW ID>.local/ Eg: https://kona-micro-0011ab.local/

- Using "IP Address" IP Address URL: https://<GW IP Address>/ Eg: https://192.0.2.111/
- 2. Follow the Node-RED Web UI link in Embedded LNS subsection in Kona Link



3. In a new flow go to menu and select Import

Node-RED						-			
Q filter nodes	Flow 1	4		•	i	inf 4	Edit		
~ common				î		•	View Arrange		
🗧 inject			•	h	Y F	0	Import	ctr]	I-1
debug					>	Ð	Expon	ctrl	-e

4. Go to section Local and select **GetUplinks** flow for your application:

Im	port nodes	
	Clipboard	✓ ⊖ Local
		✓ ₽ flows
	Local	BACnetServer.json
	Examp	🗅 GetUplinks.json
		D ModbusClient.json
		ModbusServer.json
	-	🗅 MQTT.json
		C SendDownlinks.json
	Import to current	flow new flow
		Cancel

5. On the incoming warning message, click View nodes.



6. On the Import nodes window, ensure that only Get Uplinks is checked, then click **Import selected**.

Import nodes	
Some of the nodes you are importing already exist in y Select which nodes to import and whether to replace th or to import a copy of them.	our workspace. ne existing nodes,
Flows	
Configuration nodes	
localhost:1883	Peplace
Cancel	Import selected

- 7. In GetUplinks flow configure Get Uplinks from Embedded LNS MQTT Server input node:
  - 1.1. Server: localhost:1883 for Embedded LNS
  - 1.2. Topic: Set the subscription topic to match the topic of your LoRaWAN device (e.g., device uplink).



- 8. Once you receive the uplink data, use a function node to process it. You may need to decode the payload using a data converter if Embedded LNS is set up with the correct profiles.
- 9. After decoding the data, you can use it to trigger other actions, such as sending data to a database, displaying it on a dashboard, or forwarding it to other systems.
- 10. Deploy the flow.

# 4. Troubleshooting

#### 4.1 Sending Downlinks to Embedded LNS

1. In a new flow go to menu and select Import

Node-RED			-	
Q filter nodes	+ •	i in	edit	
~ common	Î		<ul> <li>View</li> <li>Arrange</li> </ul>	
🔹 inject ,		Flo	Import	ctrl-i
debug		>	Export	ctrl-e
		> [		

2. Go to section Local and select **SendDownlinks** flow for your application:

Import nodes		
Clipboard	~ 🖻 Local	
	∽ € flows	
Local	BACnetServer.json	
Examp	🗋 GetUplinks.json	
	ModbusClient.json	
	ModbusServer.json	
	🗋 MQTT.json	
	🗅 SendDownlinks.json	
Import to current	flow new flow	•
		Cancel

3. Click twice on **Send Downlink** and configure deveui, fPort and payload for your device according to the device documentation. For Tektelic devices, you can use the KONA Atlas tool to generate the payload.

Flow 1 Send Downli	nks Edit fu	tion node	
For more information on this sample flow	, click on the below comment		Cancel Done
	¢ Pr	erties	
	Ni	e Create Downlink	<i></i>
Trigger Downlink	•	etup On Start On Mess	age On Stop
Click to Trigger Downlink	Create Downlink	<pre>return {     devEui: "034050787f3a0930",     fPort: 1,     confirmed: true,     payload: "170000" };</pre>	

To get the DevEUI go to Application section in Embedded LNS and pick the application.
 In Devices section pick the DevEUI of the Device.

<ul> <li>Network Server</li> </ul>	test application application id: f392a5	f0-6289-4291-92e9-06201a3d5193	1	Delete application
② Dashboard				
습 Tenants	Devices Multicast groups Relays	Application configuration	Integrations	
A Users				Add device Selected devices
🔎 API Keys				
Device Profile Templates	Last seen	DevEUI	Name	Device profile Battery
Regions	2024-11-15 13:28:52	034050787f3a1870	sensor 4	U\$915
습 Tenant	2024-11-15 13:20:36	034050787f3a1871	-	U\$915
② Dashboard	2024-11-15 14:43:53	034050787f3a1872	sensor 6	U5915
Ø API Keys	2024-11-15 12:59:19	034050787f3a1874	sensor 8	U\$915
Device Profiles				< 1 > 10/page ~
< Gateways				
🍘 Gateway Mesh				
Applications				

5. In Embedded LNS go to API Keys section and press Add API Key

Network Server	API keys		Add APi key
② Dashboard	ID	Name	Action
<ul> <li>☆ Tenants</li> <li>A Users</li> </ul>	6bb52190-4895-419f-95e0-87195b84b6e3	Comfort API Key Test	٥
P API Keys	5a174b9c-bf4c-446d-99a9-a0dc74d60be5	load test	0
I Device Profile Templates			

6. Record and copy the API key then press Back. It is critical that this key is recorded as it cannot be viewed again in the future. If the key is lost, you will need to generate a new one.

ChirpStack V Network Server	Network-server / API keys / Add Add API key
<ul> <li>Dashboard</li> <li>Tenants</li> </ul>	Use the following API token when making API requests. This token can be revoked at any time by deleting it. Please note that this token can only be retrieved once:
A Users	ey/J0XA/OUKV1QLC/hbGcOUIUz11NJ9.ey/hdWQ/OUjaGycHN0YWMrlwiaXNzjorY2hpcn8zdG5jay/sinN1YII6JJM5ZmUyOTILTBIMTAtNGE3OC1hZyVLTU2NmUwNDimNTYxhilsinR5cCl6lmtle5J8.ErN1r1nmh9Ou61H9DyuKu51u8ozbKj2zavKA2Fhe Q58
🔎 API Keys	
Device Profile Templates	Back

- 7. Back in NodeRED:
- double click on Send Downlink to Embedded LNS
- make sure the Server is localhost:8080
- insert API Token (Key) you got from Embedded LNS into API Token line
- press Done and then after window closes press Deploy

Flow 1	Send Downlinks		Edit device dov	vnlink node					
Eor more info	rmation on this sample flow, click on the below comm	ment then the Info icon on the top right and view t	Delete			Cancel		Done	
			© Properties				٥		191
		>	Server	localhost:80	080			1	
Trigger Downl	link		Use TLS						
			🔩 API Token					-	
Click to Trigge	er Downlink — + Create Downlink	Send Downlink to Embedded LNS	Payload Encoding	HEX	4	•			
		Downlink Message 🖌 📕	Name	Send Down	link Embedded L	NS			

#### 8. Click Deploy

Node-RED					■/ Deploy ▼	Ξ
Q filter nodes	Flow 1	Send Downlinks	-	i info	i 🍠 🕸	۰.
common	For more information or	this sample flow, click on the bel	low comment then the Info icon on the top right and view t		Q Search flows	•
⇒ inject			or conner, act to mo tool of the top right and view in	✓ Flows → E Flow 1		

## 9. Trigger downlink by pushing the activating button

Flow 1	Send Downlinks
Click to Trigger Downlink	Create Downlink
1	Downlink Message 🖌 🗐 🔲

# 4.2 Configure a Modbus Client

1. In a new flow go to menu and select Import



#### 2. Go to section Local and select Modbus Client flow:

Import nodes	
Clipboard	✓
	✓ ♥ flows
Local	BACnetServer.json
	C GetUplinks.json
	ModbusClient.json
	ModbusServer.json
_	C MQTT.json
	C SendDownlinks.json
Import to current	flow new flow
	Cancel Import

3. On the incoming warning message, click **View nodes**.

Some of the nodes you are im workspace.	porting already ex	ist in your
	View nodes	Import copy

4. On the **Import nodes** window, ensure that only **Modbus Client** is checked, then click Import selected.

Import nodes
Some of the nodes you are importing already exist in your workspace.
Select which nodes to import and whether to replace the existing nodes, or to import a copy of them.
Flows
🗹 > 🔄 Modbus Client
Configuration nodes
modbus-tcp@127.0.0.1:502
Cancel Import selected

- 5. In Modbus Client flow in Read Register Values node:
  - Address: 40001
  - Quantity: 20 (means read 20 registers starting at the 40001 Modbus register address)
  - Poll Rate: 60 seconds (value update time period)

• Update the server address

Get Uplinks	Send Downlinks	Modbus Client	Modbus Server	BACNet	Edit Modbus-R	lead node			
					Delete			Cancel	Done
For more info	ormation on this sample	flow, click on the below	comment then the Info	icon on the top right and vie	© Properties				• • •
					Settings		Optionals		
Modbus Cile	nt				Name	Read Register	Values		
					Topic	Торіс			
Read	Register Values		Aodbus Response 🙀		Unit-Id				
			data: [ 0, 0, 0, 0, 0, 0, 0, 0, 0,	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], 6	FC	FC 3: Read Ho	olding Registers 🖌		
					Address	40001			
					Quantity	20			
					Poll Rate	60	second(s)	-	
					C Delay to act	tivate input			
					Server	modbus-tcp@1	127.0.0.1:502	~ /	

6. Deploy changes. After that you can activate push button to read the Modbus register values.

						Deploy 👻
Get Uplinks	Send Downlinks	Modbus Client	Modbus Server	BACNet	▶ +	• i info i
						* Q Search
For more in Modbus Cil	formation on this sample	flow, click on the below	comment then the Info	icon on the top rig	ht a	<ul> <li>Flows</li> <li>Get Uplinks</li> <li>Send Downlinks</li> <li>Modbus Client</li> <li>Modbus Server</li> <li>Modbus Server</li> <li>BACNet</li> <li>BACnet Room</li> <li>BACnet Tests</li> </ul>
Read	Register Values		Modbus Response 🔅			> 🛃 Modbus Server
active ( 6	50 sec. )	• {	data: [ 0, 0, 0, 0, 0, 0, 0, 0, 0,	0, 0, 0, 0, 0, 0, 0, 0, 0	, 0, 0 00 00 00 0	Modbus Client

## 4.3 View Uplink Messages

1. In a right panel select the debug icon represented by a bug.

Node-RED						🚽 Deploy 👻 🔒 🔳
Q filter nodes	Modbus Server	Modbus Client	Trigger Downlink		+ *	i info i 🖉 🔅 🌣 👻
v common					Â	Q Set or flows
⇒ inject						<ul> <li>✓ Flows</li> <li>☑ Modbus Server</li> </ul>
debug						> S Modbus Client
complete						> Subflows
catch						<ul> <li>Global Configuration Nodes</li> </ul>
🔸 status 🖻						
🔶 link in 💡				Modbus TCP Server		
ink call				active		

2. In the nodes filter menu click on selected nodes



3. Select the flow you want to view uplinks for

ℜ debug i  ∅
▼ selected nodes ▼ 🛍 all ▼
select all select none
>  Send Downlinks
>  Modbus Server
∼ 🗹 Get Uplinks
☑ Uplink with Decoded Uplink Object ☑

4. You will be able to see decoded Uplink messages.

i				
▼ selected nodes ▼ 📋 all ▼				
1/22/2025, 2:00:54 PM node: Uplink with Decoded Uplink Obj v msg.payload : Object				
▼object				
<pre>deduplicationId: "9e7a9c12-859c-4b1c-901c- c622f9913a2d"</pre>				
time: "2025-01-22T22:00:53.600405858+00:00"				
<pre>&gt; deviceInfo: object</pre>				
devAddr: "00fbd51a"				
adr: true				
dr: 0				
fCnt: 9257				
fPort: 10				
confirmed: true				
data: "A2cAwwD/AS4="				
▼object: object				
<pre>battery_voltage: "3.02"</pre>				
fPort: 10				
ambient_temperature: "19.5"				
raw: "03 67 00 C3 00 FF 01 2E"				