# **RAK7431 Quick Start Guide**

### Prerequisite

### What do you need?

Before going through each and every step in the installation guide of the RAK7431 WisNode Bridge Serial, make sure to prepare the necessary items listed below:

### Hardware Tools

- 1. RAK7431 WisNode Bridge Serial
- 2. Micro USB Cable
- 3. Gateway in Range, for Testing
- 4. A Windows/Mac OS/Linux Computer

### Software Tools

- RAK Serial Port Tool

   Z
- MQTTfx Tool 
   <sup>™</sup>
   <sup>™</sup>

### **Product Configuration**

### **Typical Network Application**

RAK7431 converts data from the RS485 protocol into LPWAN wireless messages and delivers it to a cloud server through an LPWAN gateway. Cloud servers can also proactively send data to RAK7431 for two-way data transmission. Using the RAK7431, you can convert data from a conventional RS485 wired network to a wireless network.





## **Connect the RAK7431 to the Sensor Power Interface Configuration**

The RAK7431 device can be powered either by:

- DC (VIN/GND) terminals
- Micro USB.

The DC screw terminals are supporting 8 to 48 VDC. The Micro USB port can be used to power the RAK7431, up to 5 V / 500 mA DC. At the same time, the USB port is used as the configuration port for the device. Using the USB cable to connect the RAK7431 to a computer's USB port, you can import your configuration settings.

#### 📝 NOTE

The Micro USB port can be used only for powering the device. It cannot provide power to VOUT and power other devices in the RS485 network.

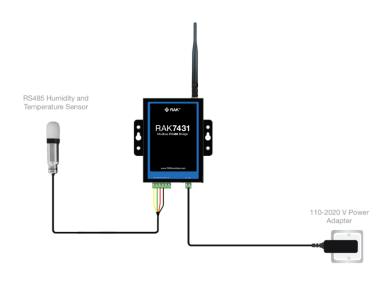


Figure 2: RAK7431 bridge with connected sensor and power supply

### **Data Interface Configuration**

The RAK7431 - RS485 serial interface can support up to **16 RS485 devices**. VOUT on the data interface can supply external power to the RS485 connected devices (only when the device is powered from the DC input). The VOUT output voltage is the same as the DC input voltage VIN.

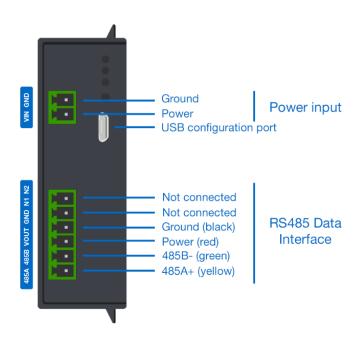


Figure 3: RAK7431 Interface pin definition

### **Gateway Connection Settings**

In this section, the **RAK7431 WisNode Bridge Serial** shall be connected into the **RAK**wireless Gateway. For this demonstration, a **RAK7249 WisGate Edge Max** shall be used. Listed below are the requisites for this section.

- RAK Serial Port Tool □ used to configure the RAK7431 WisNode Bridge Serial
- Web Management Platform Documentation guide on how to configure the RAK7249 WisGate Edge Max

### **Gateway Configuration**

### Set-up the Built-in Network Server

- 1. Sign in to the gateway by following the Accessing the Web Management section of the WEB Management Platform documentation.
- 2. Setup the RAK7249 WisGate Edge Max using its Built-in Network Server by following this guide.

### **Adding Application**

1. To enter the application configuration interface click: LoRaNetwork > Application. Enter a name for the application and click the Add button.

🏟 RAK						Logout
🏵 Status	LoRa Ne	twork Server Ap	plication Overview			
🐼 Network	Applicati	ons				
(¼) Channel Plan						
Ш LoRa Network	ID	Name	Devices	Creation Date	Description	
Network Settings Network Server Status	This sect	ion contains no value	es yet			
Gateway Application	RAK7431	_RS485		Type 1 : Unified Application Ke	ay ~ Add	
🗞 Services					Save & App	ly Reset
System						

Figure 4: Create Application in the Built-In Network Server

- 2. Turn on the Auto Add LoRa Device slider.
- 3. Generate Application EUI and Application Key by pressing the generate icon marked in the image below.



The description is optional.

<b>಄RAK</b>			Logout
🍘 Status	Application Edit - 4		
l Network	Application RAK7431_RS485		
<sup>(</sup> Å <sup>)</sup> Channel Plan			
네네 LoRa Network	Devices Application Configuration Payload F	ormats Integrations	
Network Settings	Device Authentication Mode	Unified Application Key	
Network Server Status		O Separate Application Key	
Gateway	Name	RAK7431_RS485	
Application	Auto Add LoRa Device	If enabled, LoRa Device will be added automatically after Application	on EUI and Application Key pass verification.
Global Integration	Application EUI	85143ffab9c79661	<b>2</b>
🗞 Services	Application Key	998162b7ef8bc3452b059eb9ba84191e	2
🗄 System	Description		
	Back to Overview		Save & Apply Reset

Figure 5: Registering an application

- 4. After which, press **Save & Apply**.
- 5. You will be returned to the Application page. Select **Edit** on the created application.

🏟 RAK					Logout
Status	LoRa Network Server Applic	ation Overvie	ew		
🔞 Network	Applications				
(ង) Channel Plan					
네네 LoRa Network	ID Name	Devices	Creation Date	Description	
Network Settings Network Server Status	4 RAK7431_RS485	0	Mon Jul 27 16:17:02 2020		Edit Delete
Gateway Application	Please input application name		Type 1 : Unified Application Key	,	~ Add
Global Integration					Save & Apply Reset
🗐 System					
		Fig	Jure 6: Application list		

6. Enter the **Device EUI** and press **Add**.

	NOTE
--	------

The RAK7431 Device EUI can be seen at the label on the back

🏟 RAK	Logout
🏵 Status	Application Edit - 4
🐼 Network	Application RAK7431 RS485
"Å" Channel Plan	
Ш LoRa Network	Devices Application Configuration Payload Formats Integrations
Network Settings	Last seen Device name Device EUI Class Activation mode Device Address Link margin Battery Packet Loss Description
Network Server Status Gateway	Select All     Remove     Rows per page     10     Page     1     In 0     Prev     Next       Device EUI     Q
Application	60C5A8FFFE75404B     Add     Batch Add     Import     Export
🗞 Services	Back to Overview Save & Apply Reset
Curtaria 🗐	Save & Apply Reset

- 7. On the next page, select the settings provided below:
- LoRaWAN Class: C
- Join Mode: OTAA
- Description: Optional

<b>@RAK</b>		Logout
🎯 Status	Application - Edit Device 60c5a8fffe75404b	
l Network	Device-60c5a8fffe75404b	
<sup>(</sup> Å <sup>)</sup> Channel Plan		
네네 LoRa Network	Overview Configuration Activation Downlink Live Device Data	
Network Settings	Device name RAK7431_1	
Network Server Status Gateway	Class C ~	
Application	Join mode OTAA ~	
Global Integration	Frame-counter Width 32 bit ~	
😣 Services		
🗐 System	Description RAK RS485 Bridge	
	Back to Overview Save & App	oly Reset

Figure 8: Adding the RAK7431 to the Built-In Server

### **RAK7431 Configuration**

### Connect the RAK7431 to your Network

- 1. Connect the RAK7431 to a computer using the Micro USB cable.
- 2. Open the RAK Serial Tool and select the correct COM port. The default baud rate is 115200.
- 3. After selecting, press **Open**.



Figure 9: RAK Serial Tool

• To set up the Device EUI, run the command:

AT+DEVEUI=<Device EUI>

• To check the Device EUI run:

AT+DEVEUI			

• To set up the Application EUI run the command:

AT+APPEUI=<application EUI>

• To set up the Application Key run the command:

AT+APPKEY=<application Key>

• To check the previously configured Application EUI and Key, run the commands:

AT+APPEUI	sh
ΑΤ+ΑΡΡΚΕΥ	sh

RAK	COM:	COM4 -	BaudRate: 15200	-	CLOSE
RECEIVING				CLEAR	R RECV
ок					
60C5A8FFFE754	04B				
ок					
ок					
85143FFAB9C79	9661				
ок					
ок					
998162B7EF8B0	C3452B0	59EB9BA84	4191E		
ок					
SENDING(With	\r\n)				
AT+APPKEY					
				2	SEND

Figure 10: Configuring the RAK7431

### Set the Frequency Region

The node supports the following Regional Frequencies:

- EU433
- CN470
- CN470ALI
- RU864
- IN865
- EU868
- US915
- AU915
- KR920
- AS923

For this demonstration, EU868 shall be used. To set the desired regional frequency band use the command:

AT+REGION=EU868	

#### 📝 NOTE

The regional frequency settings need to be consistent with the RAK commercial gateway supported band.

### **Data Serial Port Rate Setting**

#### 📝 NOTE

The baud rate setting needs to be consistent with the baud rate of the sensor, which is 9600.

The AT command for execution is:

AT+BAUDRATE=9600

### **Operating and Activation Mode Settings**

1. Supported operating modes are two: **Class A** and **Class C**. To set the operating mode (Class C in this case), you need to execute the AT command:

AT+CLASS=C

#### 📝 NOTE

Changes will take effect as soon as they are made.

2. Activation mode supports the following two modes: **ABP** and **OTAA**. To set the activation mode (OTAA in this case), you need to execute the AT command:





4. If everything is configured right, after the execution of the restart command this output pops up in the RAK Serial Tool:



Figure 11: RAK7431 Successful Join

### **Configure RAK7431 Working Modes**

### **Data Transparent Mode**

When the RS485 data interface works in Modbus mode, the data encapsulation format can be divided into two types: **transparent mode** and **non-transparent mode**.

- In **transparent mode**, the Modbus execution instruction response data (data, received by the node) will be directly forwarded through the LoRaWAN network.
- In the non-transparent mode, the Modbus execution instruction response data (data, received by the node) will be encapsulated in the message header according to the Modbus protocol, and then transmitted to the server through LoRaWAN.

#### 📝 NOTE

The non-transparent mode is the default one.

Enter the following AT command in the RAK Serial Tool to change the mode:

AT+TRANSPARENT=n

n Condition

transparent mode is turned off

0

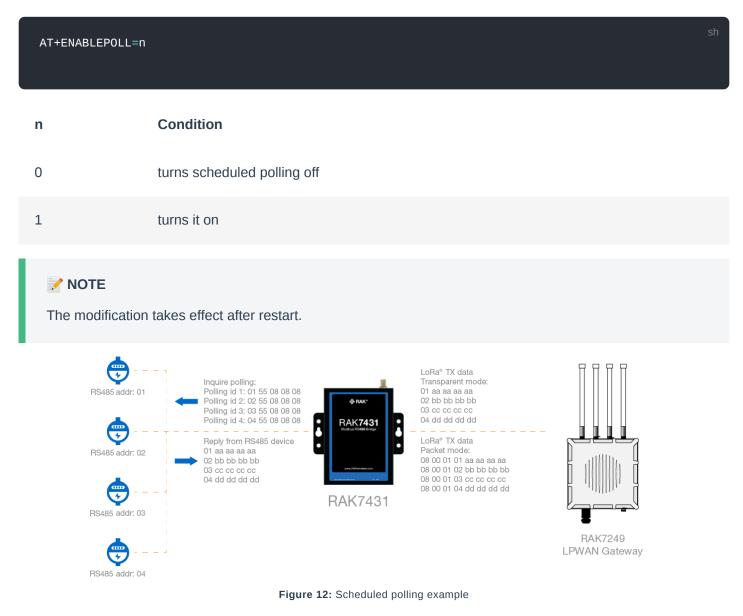
n	Condition
1	it is turned on
<b>NOTE</b> The change tal	kes effect immediately after modification.

### **Scheduled Polling Function**

When the device works in MODBUS mode, it supports the scheduled polling function.

This means that the device will perform a polling operation every given period (polling cycle). During polling, the device will send the pre-added MODBUS instructions in turn and forward the corresponding response data through the LoRaWAN network.

The device turns on the scheduled polling by default. The AT command for this is:



### **Scheduled Polling Cycle**

This command sets/reads the scheduled polling cycle. This command only works if scheduled polling is enabled. The modification takes effect after the next polling cycle or a restart.

Example: To set the polling cycle to 60 seconds, use this command:

AT+POLLPERIOD=60

RAK7431 supports polling mode, which stores up to 32 query instructions at a maximum length of 128 bytes per instruction. Polling intervals and wait times can be adjusted as needed. RAK7431 converts the data returned by the RS485 node into a LoRaWAN message, which can be sent to the LoRaWAN gateway as is or encapsulated. In transparent mode, the data for the RS485 is sent in the payload of the LoRa message as is, and in non-transparent mode, the data of RS485 is encapsulated in the LoRa message with a header and validation.

### **Add Polling Instructions**

To add polling instruction, execute the AT command:

AT+ADDPOLL= <n></n>	: <xxxx></xxxx>	sh
Parameter	Description	Value Range
n	polling instruction ID	1 to 127
хххх	polling instruction content; hexadecimal string	128 bytes max

According to the temperature and humidity register address of the temperature and humidity sensor in the example and the RS485 address, the polling instruction should be:

AT+ADDPOLL=1:010300000002C40B		

**Example**: If you have added multiple RS485 temperature and humidity sensors, continue to increase the polling instructions based on the RS485 address and register address, for example:

- RS485 Temperature and humidity sensor addr: 01, Polling 1: 01030000002C40B
- RS485 Temperature and humidity sensor addr: 04, Polling 2: 04030000002C45E
- RS485 Temperature and humidity sensor addr: 08, Polling 3: 08030000002C492
- RS485 Temperature and humidity sensor addr: 0F, Polling 4: 0F030000002C525

You will need to increase the polling instruction by the following AT commands:



sl

AT+ADDPOLL=4:0F0300000002C525

The RAK7431 sends an instruction to the sensor every 1 minute to obtain temperature and humidity data, and the following is the result of 3 consecutive scheduled polls:

- DTU Tx: The polling instruction sent to the Sensors over RS485 Data Interface
- DTU Rx: The sensor data received.
- LoRa Tx : Send the received data through a LoRaWAN network.

<u> </u>	KAK	SERI	AL P	ORI	IOOL	



Figure 13: Data in transparent mode

🚬 rak serial port tool

<b>RAK</b>	COM: COM4	BaudRate: 15200	CLOSE
RECEIVING			CLEAR RECV
EVENT:0:START SYSLOG:4:OTAA SYSLOG:4:OTAA EVENT:1:JOIN_	Join Request Join Success NETWORK		
SYSLOG:4:DTU	Tx : 0103000000 Rx : 01030402050 Tx : 810001000A		BE8
SENDING(With	(r\n)		
			SEND

Figure 14: Data in non-transparent mode

- Humidity calculation: hex is 0210, the decimal is 528, converted humidity is 52.8% RH.
- Temperature calculation: hex is 012F, the decimal is 303, converted temperature is 30.3 °C.

sh

### **MQTT Subscribe to Data Server**

To better demonstrate the functionality we will use the Application Server Integration feature to subscribe to the Built-In Network Server Topics, using the MQTT client, to obtain data and send instructions to the RAK7431.

<b>@RAK</b>		Logout
🏵 Status	Application Server Integration	
l Network	Application Server Integration	
<sup>(</sup> Å <sup>)</sup> Channel Plan		
네 LoRa Network	General Setup MQTT Topic template Setup	
Network Settings Network Server Status	All topics support one or more of the four variable fields {{application_ID}} {{application_name}} {{application_EUI}} {{device_EU} be in a separate domain (the content between the two '/'s is a separate domain). All topics must contain the {{device_EUI}} variab	
Gateway	Join Topic application/{{application_ID}}/device/{{device_EUI}}/join	
Application	Event published when a device joins the network.	
Global Integration	Uplink Topic application/{{application_ID}}/device/{{device_EUI}/rx	
👶 Services	Contains the data and meta-data for an uplink application payload.	
	Downlink Topic application/{{application_ID}}/device/{{device_EUI}}/tx	
System	Scheduling downlink data by application server	
	Ack Topic application/{{application_ID}}/device/{{device_EUI}}/ack	
	Event published on downlink frame acknowledgements.	
	Status Topic application/{{application_ID}}/device/{{device_EUI}}/status	
	Event for battery and margin status received from devices.	
		Save & Apply Reset

Figure 15: Gateway MQTT Topic Templates

To communicate with the MQTT bridge in the gateway we need to use MQTT Topic Templates.

#### **MQTT Topic Configuration**:



- 1. Download and install MQTTfx tool 🖾 to read the topics and send data to the gateway and node.
- 2. After installation, the MQTT Client must be configured. Select **local mosquitto** from the drop-down list and click the **edit connection profiles** icon marked in the image below to open the settings page.

😔 MQIT.fx - 1.7.1 —		$\times$
File Extras Help		
local mosquitto Connect Disconnect		•
Publish Subscribe Scripts Broker Status Log		
application/4/device/60c5a8fffe75404b/rx Cos0 Qos1 Qos2 Au	toscroll	0;•

Figure 16: MQTT.fx Client

- 3. On the next window, input the **Broker Address** and **Broker Port**. If the Client ID is empty press **Generate**. Then click **OK**.
- Broker Address: Address of MQTT server the gateway IP.
- Broker Port: Consistent with MQTT Broker Port set by the gateway by default 1883.

Profile Nam	local mosquitto	
Profile Typ	MQTT Broker	
MQTT Broker Profile Setting	s	
Broker Addre	ss 192.168.0.120	
Broker Po	rt 1883	
Client I		Conorate
Client	D 7b6ebcf661ea476d8569fc403e3a046c	Generate
General User Credentials	SSL/TLS Proxy LWT	
Connection Timeo	ut 30	
Keep Alive Interva	al 60	
Clean Sessio		
Auto Reconne	ct	
Max Inflig	ht 10	
MQTT Versio	on 🗸 Use Default	
	3.1.1	
	Clear Publish History	
	Clear Subscription History	
	clear subscription history	
Revert		Cancel OK Apply

Figure 17: MQTT.fx settings

4. Click on the **Connect** button. The green dot indicates that the connection is successfully subscribed to the MQTT Broker.

😔 MQIT.fx - 1.7.1	- 🗆 ×
File Extras Help	
local mosquitto Connect Disconnect	-
Publish Subscribe Scripts Broker Status Log	
Figure 10: MOTT for connected successfully	

Figure 18: MQTT.fx connected successfully

- If we want to receive all data from the MQTT Bridge, we can use the wildcard character #.
- 5. Choose the Subscribe tab, enter the wildcard and press Subscribe.

🌚 MQTT.fx - 1.7.1		- 🗆 ×
File Extras Help		
local mosquitto	Connect     Disconnect	<b>₽</b> ●
Publish Subscribe Scripts	Broker Status Log	
#	Subscribe	Qo5 0 Qo5 1 Qo5 2 Autoscroll 05v
#	Dump Messages Mute Unnultwortbe	

Figure 19: Subscribing to MQTT Broker with wildcard

• If the node sends data, the MQTT client will display it as it is subscribed to the topic.

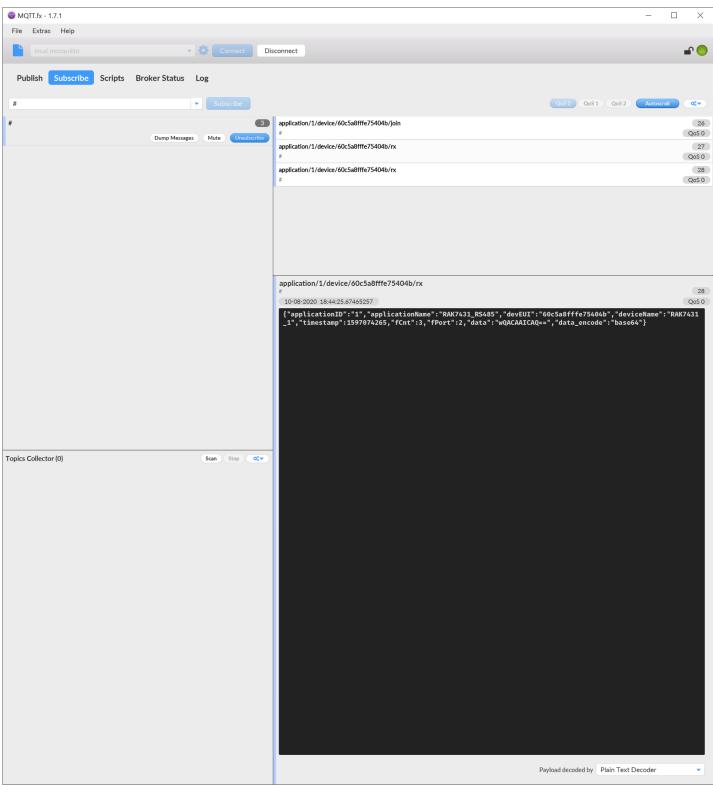


Figure 20: Subscribed topic data

- Notice that the data field is in **base64** format, which has to be converted to hex string to be useful. We can change the data format from the built-in server settings.
- 6. This is done by going to Gateway>Application>Integrations>Data Encode/Decode Type and chose HEX String form the drop-down menu. Press Save & Apply.

🏟 RAK		Logout
🍘 Status	Application Edit - 4	
l Network	Application RAK7431_RS485	
(🎢) Channel Plan		
네네 LoRa Network	Devices Application Configuration Payload Formats Integrations	
Network Settings Network Server Status Gateway Application Global Integration	Data Encode/Decode Type HEX String ~ Report LoRa Radio Infomation Enable HTTP/HTTPS Integration HTTP/HTTPS Headers Header Name :	
😣 Services	Header Value	
III System	Uplink data URL	
	Join notification URL       Ack notification URL	
	Device-status notification URL	
	Maximum number of concurrent connections 16	
	Maximum length of queue 64	
	Back to Overview	Save & Apply Reset

Figure 21: Change the Data Encode/Decode Type

• Now, all received data will be in HEX String.

### **AK** Documentation Center

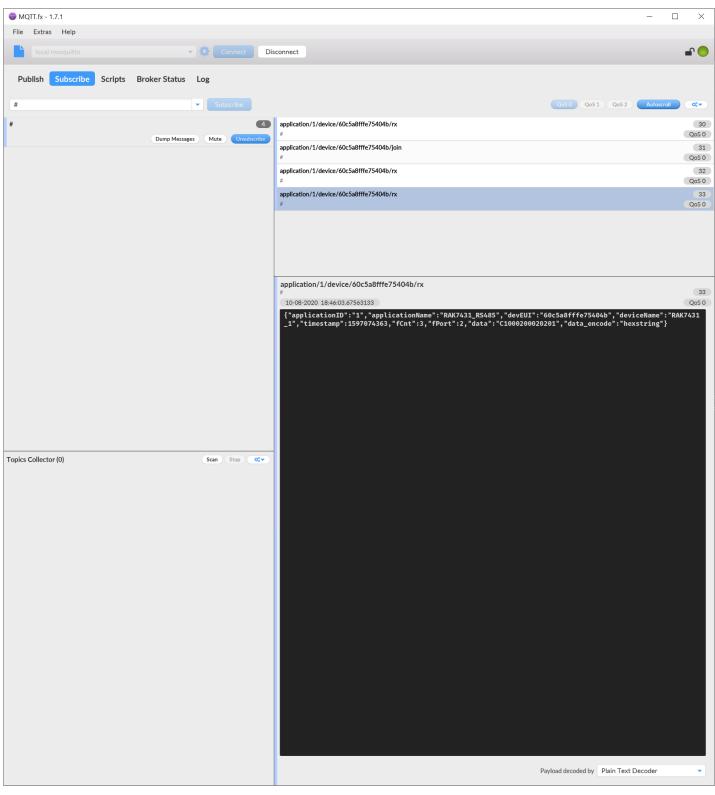


Figure 22: Received data field in HEX format

# RAK7431 Remote Control and Configuration via MQTT.fx

To remotely control the RAK7431 you need to publish messages to the **Gateway's Network Server MQTT "TX" topic**.

### Add a Scheduled Polling Task List

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA	
0x03	2Byte	2Byte	TASK_ID	DATA
0.03	ZDyle	ZDyte	1Byte	nByte

#### **NOTE**

The message length does not contain the header

**Example**: We will add a polling instruction.

```
Publish topic:
```

application/1/device/60c5a8fffe75404b/tx

#### 📝 NOTE

Application ID and Device EUI should be consistent with the settings within the gateway.

• To successfully complete this, the JSON data format must be followed.

#### Content of the uplink:

{		
"C	confirmed":true,	
"f	fPort":129,	
"0	data":"030001000901010300000002C40B"	
}		

Parameter	Description
"confirmed":true	This indicates that the downlink to the RAK7431 will be confirmed for successful receiving.
"fPort":129	Defines the port that we want to send the command. (For more information on the fPort see the AT Command Manual for RAK7431)
"data":"03000100090101030000002C40B"	The data of the task in hexadecimal format.

The content of the data that we will send is:

03 0001 0009 01 01030000002C40B



Figure 23: Data arrangement

- 1. DTU command word
- 2. The message number
- 3. Message length (excluding header)
- 4. The task ID
- 5. The content of the task

🤓 MQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>-</b>
Publish         Subscribe         Scripts         Broker Status           >         application/1/device/60c5a8fffe75404b/tx	Log Publish	Qo50 Qo51 Qo52 Retained Cr
{ "confirmed":true, "fPort":129, "data":"03000100090101030000002C40B" } }		

Figure 24: Publishing data to RX topic

• After publishing the data, we can see the downlink instruction and uplink answer from the RAK Serial Tool:



Figure 25: Received data and sent an answer

#### Message format when execution is successful:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x83	2Byte		TASK_ID
0,03	ZDyte	2Byte	1Byte

• The MQTT subscription bar can see the upstream message "83000100010101" for successful execution.

😇 MQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>e</b> O
Publish Subscribe Scripts Broker St	atus Log	
#	Subscribe Qos C	QoS 1 QoS 2 Autoscroll
# 3 Dump Messages Mute Unsubscribe	application/1/device/60c5a8fffe75404b/tx #	16 QoS 0
Dunip Messages Mute Onsuscince	application/1/device/60c5a8fffe75404b/rx #	17 QoS 0
	application/1/device/60c5a8fffe75404b/rx #	18 QoS 0
Topics Collector (0) Scan Stop 😋	application/1/device/60c5a8fffe75404b/rx # 03-08-2020 18:52:15.67935882	17 QoS 0
	<pre>{"applicationID":"1","applicationName":"RAK7431 b","deviceName":"dev-60c5a8fffe75404b","timestar 29,"data":"830001000101","data_encode":"hexstrin</pre>	_RS485","devEUI":"60c5a8fffe75404 mp":1596469935."fCnt":4."fPort":1

Figure 26: Received confirmation of the task

### **Remove the Scheduled Polling Task List**

Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x04	2Byte		TASK_ID
0.04	ZDyte	2Byte	1Byte

**Example**: Removal of timed polling temperature and humidity sensor task order on a node:

#### Publish the topic:

Application/1/device/60c5a8fffe75404b/tx

#### Content:

{ "confirmed": <b>true,</b> "fPort": <b>129,</b> "data":"040001000101" }		sh
MQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>■</b> ●
Publish         Subscribe         Scripts         Broker State           >         application/1/device/60c5a8fffe75404b/tx	atus Log Publish	QoS0 QoS1 QoS2 Retained OST
{ "confirmed":true, "fPort":129, "data":"040001000101" }		

Figure 27: Remove poll downlink message

#### Message format when execution is successful:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x84	28,46	2Byte 2Byte	TASK_ID
0,04	ZDyle		1Byte

WQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>-</b>
Publish Subscribe Scripts Broker Sta	tus Log	
#	Subscribe     QoS 0 Qos	S1 QoS2 Autoscroll
# 2 Dump Messages Mute Unsubscribe	application/1/device/60c5a8fffe75404b/tx #	22 QoS 0
Dump Messages Muce Onsubscribe	application/1/device/60c5a8fffe75404b/rx #	23 QoS 0
Topics Collector (0) Scan Stop 😪 🗸	<pre>application/1/device/60c5a8fffe75404b/rx # 03-08-2020 18:59:00.68340364 {"applicationID":"1", "applicationName":"RAK7431_RS485 b","deviceName":"dev-60c5a8fffe75404b","timestamp":15 29,"data":"840001000101","data_encode":"hexstring"}</pre>	23 QoS 0 ","devEUI":"60c5a8fffe75404 96470340,"fCnt":9,"fPort":1
	Payload decoded by	Plain Text Decoder

Figure 28: Poll removed successfully message

• The MQTT subscription bar sees the upstream message "84000100010101", which means the task was successfully removed.

### **Read the Scheduled Polling Task List**

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x05	2Byte 2Byte		TASK_ID
0x05	ZDyle	ZDyte	1Byte
Publish topic:			

application/1/device/60c5a8fffe75404b/tx

#### Content:

# Section Center

{ "confirmed": <b>true,</b> "fPort": <b>129,</b> "data":"050001000101" }	sh
WQIT.fx - 1.7.1	- 🗆 X
File Extras Help	
local mosquitto	<b>-</b>
Publish       Subscribe       Scripts       Broker Status       Log         > application/1/device/60c5a8fffe75404b/tx       Publish       Qot         ["confirmed":true, "fPort":129, "data":"050001000101"       Qot	50 QoS1 QoS2 Retained Cr

Figure 29: Publishing the read poll task message

#### Perform successful upstream message format:

DTU_CMD	MSER	MDATA_LEN	MDAT	A
0×85	2Bvto	2Duto	TASK_ID	DATA
0,05	0x85 2Byte 2B	2Byte	1Byte	nByte

Open the MQTT subscription column that is to see to the performance of the above line:
 "850001000901010300000002C40B" is the query to the task, the order ID is 1, the task order content is
 01030000002C40B (example registers).

# **BAK**° Documentation Center

🜚 MQTT.fx - 1.7.1		_	
File Extras Help			
local mosquitto	Connect Disconnect		<b>-</b>
Publish Subscribe Scripts Broker Sta	tus Log		
#	Subscribe	Qo5 0 Qo5 1 Qo5 2 Au	toscroll 🕵 🗸
# 2 Dump Messages Mute Unsubscribe	application/1/device/60c5a8fffe75404b/tx #		28 QoS 0
Sumpressiges (Hace Shoulderhoe	application/1/device/60c5a8fffe75404b/rx #		29 QoS 0
Topics Collector (0) Scan Stop Cor	<pre>application/1/device/60c5a8fffe75404b/rx # 03-08-2020 19:07:21.68841182 {"applicationID":"1","applicationName": b","deviceName":"dev-60c5a8fffe75404b", 129,"data":"850001000901010300000002C40</pre>	"timestamp":1596470840,"fCnt"	:12,"fPort":
		Payload decoded by Plain Text Decode	ar 🔹

Figure 30: Received message from the node

### **Read the LoRa Configuration**

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x06	2Byte	2Byte	0Byte

#### Publish topic:

Application/1/device/60c5a8fffe75404b/tx	sh

#### Content:

{	
"confirmed":true,	
"fPort":129,	
"data":"0600010000"	
}	

🥹 MQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>-</b>
Publish         Subscribe         Scripts         Broker Status           >>         application/1/device/60c5a8fffe75404b/tx	Log Publish	QoS 0 QoS 1 QoS 2 Retained 🕸
{ "confirmed":true, "fPort":129, "data":"0600010000" } 		



#### Perform successful upstream message format:

DTU_CMD	MSER	MDATA_LEN	MDATA					
0x86	2Byte	2Byte	DATA RATE	TXPWR	CONFIRM	RETRY	ADR	DUTY CYCLE
	5	2	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

- DATARATE: Speed rate (0 5)
- **TXPOWER**: The transmit power level (0 20)
- CONFIRM: Whether to turn on ACK 0 off, 1 on
- **RETRY**: Maximum re-transmission times when ACK is on (0 ~ 15)
- ADR: Whether to turn on the dynamic rate adjustment 0 off, 1 on
- **DUTY CYCLE**: Whether to turn on duty cycle limit 0 off, 1 on

application/1/device/60c5a8fffe75404b/rx #	3
10-08-2020 17:42:42.63762943	QoS 0
{"applicationID":"1","applicationName":"RAK7431_RS 5404b","deviceName":"RAK7431_1","timestamp":159707 "data":"8600010006000001030100","data_encode":"hex	6485","devEUI":"60c5a8fffe7 70562,"fCnt":2,"fPort":129,
Payload decoded by	Plain Text Decoder 🔹

Figure 32: Received message with LoRa configuration

• Open the MQTT subscription bar to see the upstream message "860001000006000010301000000" to read the LoRa configuration based on the upstream message format for the successful execution above.

### **Change the LoRa Configuration**

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA					
0x07	2Byte	2Byte	DATA RATE	TXPWR	CONFIRM	RETRY	ADR	DUTY CYCLE
	,	,	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

#### Publish topic:

Application/1/device/60c5a8fffe75404b/tx	

#### Content:

5	sh
נ "confirmed" <b>:true,</b>	
"fPort":129,	
"data":"070001000601050103010"	
}	

• The above command changes the **data rate to "1"** and the **transmit power to "5"**.

MQTT.	.fx - 1.7.1									-	$\times$
File E	xtras Help										
				• • (	Connect	Disconnect					<b>•</b> O
Publis	sh Subscribe	Scripts	Broker Status	Log							
» app	blication/1/device/60	0c5a8fffe75	404b/tx	•	Publish		QoS 0	QoS 1	QoS 2	Retained	<b>Q</b> <sub>0</sub> <sup>0</sup> ▼
"fPort":1	ned":true, 129, 0700010006010501	.03010"									

Figure 33: Publish change LoRa configuration data

#### Perform successful upstream message format:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x87	2Byte	2Byte	0Byte

• Open the MQTT subscription bar to see the upstream message for successful execution: "8700010000".



Figure 34: Received confirmation message

### **Reset the default LoRa Configuration**

Publish topic:

Application/1/device/60c5a8fffe75404b/tx	sh
Content:	
{ "confirmed":true, "fPort":129, "data":"1D00010000" }	sh
@ MQIT.fx - 1.7.1	- 🗆 X
File Extras Help	
local mosquitto  Connect Disconnect	<b>₽</b> ●
Publish Subscribe Scripts Broker Status Log	
» application/1/device/60c5a8fffe75404b/tx	QoS 0 QoS 1 QoS 2 Retained 😋
{ "confirmed":true, "fPort":129, "data":"1D00010000" }	

Figure 35: Publish reset the default LoRa configuration

• Open the MQTT subscription bar to see the upstream message for successful execution: "9D00010000".



Figure 36: Received Data

LORA configuration default values:

DATARATE	TXPOWER	CONFIRM	RETRY	ADR_ENABIE	DUTYCYCLE_ENABLE
0 – DR_0	19 -19dBm	1 – open	3 times	1 – open	0 – close

### **Read the DTU Configuration**

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x08	2Byte	2Byte	0Byte

#### Publish topic:

Application/1/device/60c5a8fffe75404b/tx	

#### Content:

		sh
{		
"C	onfirmed":true,	
"f	Port":129,	
"d	ata":"0800010000"	
}		

🌚 MQTT.fx - 1.7.1		- 🗆 X
File Extras Help		
local mosquitto	Connect Disconnect	<b>-</b>
Publish Subscribe Scripts Brok		
<pre>&gt;&gt; application/1/device/60c5a8fffe75404b/t {     "confirmed":true,     "fPort":129,     "data":"0800010000" }</pre>	x Publish	QoS0 QoS1 QoS2 Retained Co

Figure 37: Publish message for reading the DTU configuration

#### Uplink data message format when execution successful:

DTU_CMD	MSER	MDATA_LEN			MDATA		
0x88	2Byte	2Byte	POLL ENABLE	POLL PERIOD	BUS TIMEOUT	RETRY	RS485
	5	5	1Byte	4Byte	1Byte	1Byte	1Byte

- POLL ENABLE: Enables scheduled polling, 0 off, 1 on
- POLL PERIOD: Polling period, in seconds
- BUS TIMEOUT: Bus timeout. The unit is seconds
- RETRY: Number of retries after bus timeout. 0 turn off retry function
- **RS485**: 485 bus parameters

Open the MQTT subscription bar to see the upstream message "**880001000080000003C010050**" to read the DTU configuration according to the successful upstream message format above.

application/1/device/60c5a8fffe75404b/rx #	9
10-08-2020 17:44:30.63870805	QoS 0
{"applicationID":"1","applicationName":"RAK7431_RS 5404b","deviceName":"RAK7431_1","timestamp":159707 "data":"8800010008010000064010050","data_encode":	0670, "fCnt":4, "fPort":129,
Payload decoded by	Plain Text Decoder

Figure 38: Received message with current DTU configuration

### **Change the DTU POLL configuration**

#### Downlink instruction message format:

DTU_CMD	MSER	MDATA_LEN			MDATA		
0x09	2Byte	2Byte	POLL ENABLE	POLL PERIOD	BUS TIMEOUT	RETRY	RS485
	,	,	1Byte	4Byte	1Byte	1Byte	1Byte

#### Publish topic:

Application/1/device/60c5a8fffe75404b/tx	

#### Content:

	sh
"confirmed":true,	
"fPort":129,	
"data":"09000100080100000E10010050"	
}	

• The above command changes the polling period to only 1 hour.

ITIQM 🐨	.fx - 1.7.1									_	×
File E	Extras Help										
				• • (		Disconnect					<b>P</b> 🔴
Publi	sh Subscribe	Scripts	Broker Status	Log							
» app	plication/1/device/6	0c5a8fffe75	404b/tx	•	Publish		Qo5 0	Qo51	Qo5 2	Retained	Q <sub>6</sub> °▼
"fPort"::	ned":true, 129, 0900010008010000	00E1001005	0"								

Figure 39: Publish message for change the DTU configuration

#### Uplink data message format when execution successful:

DTU_CMD	MSER	MDATA_LEN	MDATA
0x89	2Byte	2Byte	0Byte

• Open the MQTT subscription bar to see the upstream message for successful execution: "8900010000".



Figure 40: Received confirmation message

### **Reset the default DTU Configuration**

Publish topic:

Application/1/device/60c5a8fffe75404b/tx	sh
Content:	
{ "confirmed": <b>true,</b> "fPort": <b>129,</b> "data":"1E00010000" }	sh
WQTT.fx - 1.7.1	- 🗆 ×
File Extras Help	<b>-</b>
Publish       Subscribe       Scripts       Broker Status       Log         > application/1/device/60c5a8fffe75404b/tx <ul> <li>Publish</li> <li>forfirmed":true, "fPort":129, "data":"1E00010000"</li> <li>j</li> </ul>	QoS 0 QoS 1 QoS 2 Retained 05 v

Figure 41: Publish reset the default DTU configuration

• Open the MQTT subscription bar to see the upstream message for successful execution: "9E00010000".

### **AK** Documentation Center



Figure 42: Received Data

DTU Configure the initial value:

POLL_ENABLE	POLL_PERIOD	BUS_TIMEOUT	RS485
1 - on	3600 seconds	1 second	0xE0

### **Connecting to the Helium Network**

Helium has quickly become the most widespread LPWAN communal network with more than 27,000 devices deployed globally. All the RAKwireless node products are compatible with it and the process of adding a device to the network is intuitive and straightforward.

This section will focus on giving a brief guide on how to connect the RAK7431 to the network console, assuming that there is a Helium Hotspot within range.

Log in or create your account in the Helium console page  $\ensuremath{\mathbb{Z}}$  .

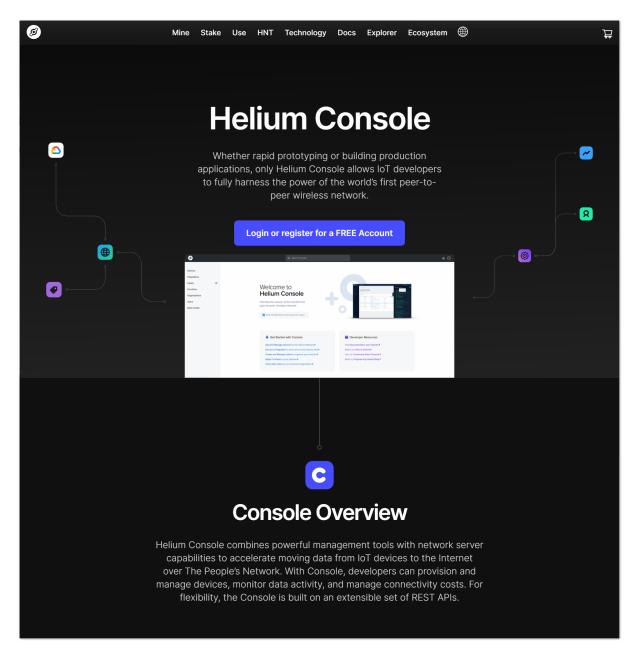


Figure 43: Helium Console

Once registered/logged in, you will end up at the home page where you can see your function tree on the left and your DC balance at the top, as well as several useful links.

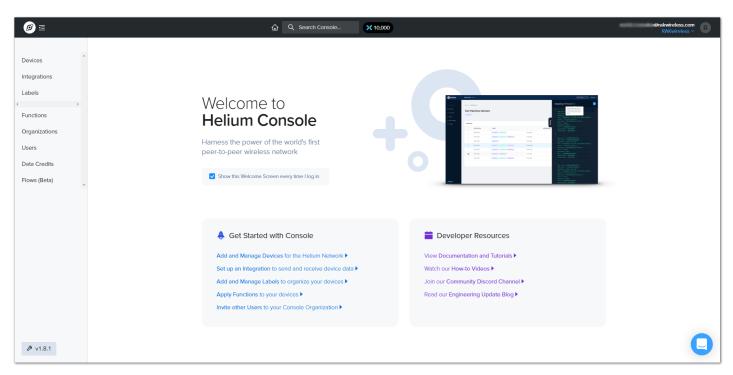


Figure 44: Helium console home screen

Go to the **Devices** section in the function tree. If this is your first time doing this, there will be no devices registered. Click the **+ Add Device** button to get started.

<b>ø</b> =	☆ Q. Search Console X 10,000
Devices	Devices
Integrations	
Labels	
Functions	
Organizations	
Users	
Data Credits	No Devices
Flows (Beta)	NO Devices
	You haven't added any devices yet.
	+ Import Devices + Add Device
	Devices can be added to the Helium network.
	More details about adding devices can be found here.

Figure 45: Devices section

A window will pop up with a set of a field containing the device parameters required for its registration.

Name	RAK7431	
Dev EUI	66E4769D55	8/8 Bytes
App EUI	0D19EB695D	8 / 8 Bytes
App Key	© 37D206C31DF3A4A016EF28AA3B	16 / 16 Bytes
ttach a L	abel (Optional)	

Figure 46: Adding a new device

Fill in a name of your choosing. The **Dev EUI**, **App EUI**, and **App Key** will have random values generated for you by default. Press the eye icon to reveal the values. You can manually replace them with values of your own. For this tutorial, use the default values. Press the **Submit** button, and you are done.

ø =		@rakwireless.com RAKwireless ∨
	Devices	+ Import Devices + Add Device
Devices		
Integrations	Devices can be added to the Helium network. Tell me more about adding devices.	
Labels	1 Devices Edit Columns 10 r	results $\lor$ Quick Action $\lor$
Functions	Device Frame Frame	e Packets DC Date
Organizations	Name Device EUI Cabels Integrations Up Down	
Users		Apr 20,
Data Credits	RAK7431 66E4769D55 None	0 0 2021 11:48 AM
Flows (Beta)		

Now, your RAK7431 is registered and is awaiting activation. For this to happen, you need to import the Dev EUI, App EUI, and App Key in the RAK7431 using the RAK Serial Port Tool 2.

Open the tool, select the desired port (default baud rate) and open it. Then start importing your settings.

Configure your LoRa band and activation mode. This tutorial will be using the EU868 band and OTAA (the only option available for now with Helium) with device class A (default one, does not need configuring).

· Regional band, device class, and activation mode setting



• Enter the Dev UI

Use the command below by replacing the XXXX with your Device EUI from the Helium console:

at+deveui=XXXX

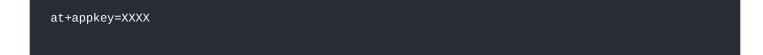
• Enter the App EUI

The same as with the Device EUI, replace the XXXX with your value:

at+appeui=XXXX

• Enter App Key

Finally, fill in the App key with the following command:



• Join Network

Run the following AT command in order for the node to join the network.

Once the procedure is initiated and successfully complete, you will have a notification in the serial console.

at+restart

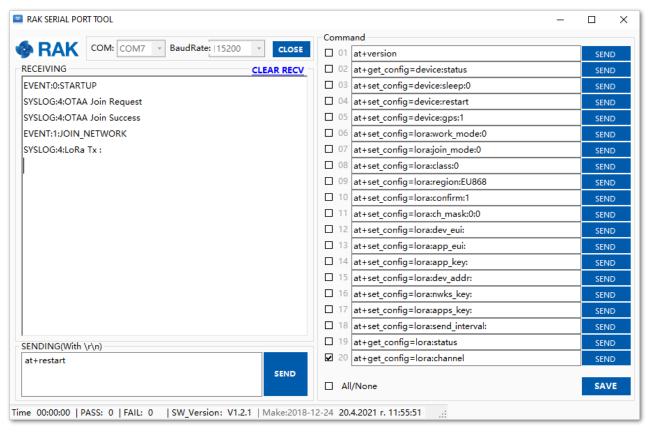


Figure 48: RAK7431 EUIs and key

If you take a look at the Helium console, you will also see the join request packets both in the graph and event log. Your node is now a part of the Helium Network.

ø =			습 Q Search Conso	ole )( 9,9	997				erak R	wireless.com AKwireless >
Devices	Real Time Packets									
Integrations	Live Data									
Labels			_							
Functions	0									
Organizations	-20									
Users	-40									
Data Credits	-60 02									Û
Flows (Beta)	-80									$\bigcirc$
	-100									
	-120									
	0s	-30s -60s	-90s	-120s	-150s Time Past in Seconds	-180s	-210s	-240s	-270s	-300s
	Event Log Expand All	Show Dropped Uplinks:	ite Inactive Device						Ex	port JSON
	Frame Count		Туре		Time					
	+ 🗸 O		Join Accept		Apr 20, 2021	11:37:15.157 AM				
	+ •		Join Request		Apr 20, 2021	11:37:13.157 AM				Q

Figure 49: Helium console live device data

### Connecting to The Things Network V3 (TTNv3)

At The Things Conference 2021, it was announced that The Things Network is upgrading to The Things Stack v3. In this section, it will be shown how to connect RAK7431 WisNode Bridge Serial to The Things Stack. To login into the TTNv3, head on here 🖸 . If you already have a TTN account, you can use your The Things ID credentials to log in.

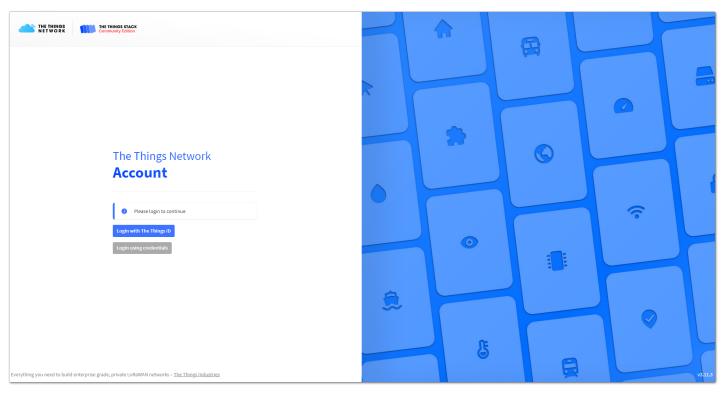


Figure 50: The Things Stack Home Page

THE THINGS STACK	Gateways 🕰 Organizations	🕀 eu1 Community RAKwireless 👻
	<b>Welcome to the Console!</b> Get started right away by creating an application or registering a gateway. Need help? Have a look at our Pocumentation of or <u>Get Support</u> .	
	Original     Original       Original     O	
	Version info Component status	

Figure 51: Console Page after successful login

#### 📝 NOTE

To be able to connect RAK7431 WisNode Bridge Serial to TTNv3 you should already have connected a gateway in range to TTNv2 or TTNv3, or you have to be sure that you are in the range of a public gateway.

### Adding an application

#### 📝 NOTE

This tutorial is for EU868 Frequency band.

1. To create an application, choose **Create an application** (for new users that do not already have created applications) or **Go to applications** > **+ Add application** (for users that have created applications before).

### **AK** Documentation Center

THE THINGS THE THINGS STACK Community Edition	Overview  Applications  Gateways  Corganizations	eu1 Community	RAKW
	Add application		
	Owner* RAKwireless		
	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		
	Create application		
you need to build enterprise grade, private LoRaWAN	networks - <u>The Things Industries</u>	v3.11.3 Documen	tation 🗹 🛛 🖓 G

Figure 52: Create an application page

- 2. Fill in the needed information:
- Owner Automatically filled by The Things Stack, based on your account or created Organization.
- Application ID This will be the unique ID of your application in the Network. Note that the ID must contain only lowercase letters, numbers, and dashes (-).
- Application name (optional) This is the name of your application.
- Description (optional) Description of your application. Optional application description; can also be used to save notes about the application.
- 3. After you fill in the information, click **Create application**. If everything is filled in correctly, you will see the same page, as shown in Figure 53.

	THE THINGS STACK Community Edition	🗖 Applications 🛛 🚔 Gate	ways 🎎 Organizations			tommunity eu1 Community	RAKwireless •
Rakwireless Test Applicati		Applications > Rakwireless					
Overview		ID: raktest-applica	ss Test Application				
🙏 End devices		🙏 0 End devices 🛛 🏦 1 0	Collaborator 🛛 🗣 0 API keys			Created 2 seconds ago	
<ul> <li>□ Live data</li> <li>◇ Payload formatters</li> <li>✓ Arlikeys</li> <li>◇ API keys</li> <li>♦ General settings</li> </ul>		General Information Application ID Created at Last updated at	raktest-application Mar 8, 2021 12:22:21 Mar 8, 2021 12:22:21	5	Live data     Walting for events from rektest     Walting for events from rektest		
		End devices (0)	Name O	DevEUI No items	JoinEUI	Created \$	
< Hide sidebar							
Everything you need to build enterprise	e grade, private LoRaWAN networks – <u>The Thin</u>	gs Industries				v3.11.1 <u>Do</u>	cumentation 🛛 🔇 Get Support

Figure 53: Application Overview

### Registering and Configuring the Device In OTAA Mode Registering the Device in OTAA Mode

1. From the Application Overview page, click on + Add end device.

	THE THINGS STACK Community Edition	🗖 Applications 📑 Gateways 🎿 Organizations	eul Community RAKwireless •
Rakwireless Test Applicati		Applications > Rakwireless Test Application > End devices > Register from The LoRaWAN Device Repository	
Cverview		Register end device	
🙏 End devices		From The LoRaWAN Device Repository Manually	
<ul> <li>□ Live data</li> <li>&gt; Payload formatters</li> <li>↓ Integrations</li> <li>↓ Collaborators</li> <li>→ API keys</li> <li>♦ General settings</li> </ul>		1. Select the end device         Brand*         Type to search         V         Cannot find your exact end device? Get help here and try manual device registration.         2. Enter registration data         Please choose an end device first to proceed with entering registration data         Register end device	
< Hide sidebar			
Everything you need to build enterprise	grade, private LoRaWAN networks - The Thin	gs Industries	v3.11.2 Documentation 🗹 🕜 Get Support

Figure 54: Adding a device in OTAA mode

Below the Register end device heading you can find two options for registering a device. Since RAK7431
WisNode Bridge Serial is part of The LoRaWAN Device Repository, you can register it From The LoRaWAN
Repository option. In the Brand dropdown menu find and select RAKwireless Technology Co. and a Model
field will pop up next to it. In it choose RAK7431 WisNode Bridge Serial.

	THE THINGS STACK Community Edition	Applications 🝶 Gateways	S 🔐 Organizations					eu1 Community	RAKwireless •
Rakwireless Test Application		Applications > Rakwireless Test	Application > End devices > Re	gister from The LoRaW	AN Device Reposito	ry			
Overview		Register end devi	ce						
🙏 End devices		From The LoRaWAN Device Rep	ository Manually						
Live data     Payload formatters		1. Select the end device							
大 Integrations v		Brand *	Model*	Hardware Ver.*	Firmware Ver.*	Profile (Region)*			
Collaborators		RAKwireless Technology   >>	RAK7431 WisNode Bridg   ~	1.0 v	1.2.0 🗸 🗸	Select			
Ov API keys									
General settings		2. Enter registration data Please choose an end device first Register end device	to proceed with entering registratio	n data					
< Hide sidebar	stade, orivate LoRaWAN networks – The Things								ntation 2 Q Get Support

Figure 55: Choosing the device

- 3. After choosing the device, three more fields will pop up.
- Hardware Ver. Version of the hardware. This is the only option, so leave it as default.
- Firmware Ver. Version of the firmware. This is the only option, so leave it as default.
- Profile (Region) Here the region is chosen.

#### **NOTE**

For this example, the EU\_863\_870 is chosen.

4. Next, an **Enter registration data** heading will pop up below. Scroll down to enter the required data for the device.

	Brand *	Model*	Hardware Ver.*	Firmware Ver.*	Profile (Region)*
plication	RAKwireless Technology 🗸	RAK7431 WisNode Bridg 🗸	1.0 🗸 🗸	1.2.0 🗸 🗸 🗸	EU_863_870 🗸 🗸
•	MAC V. RAK74 applic wireles	1431 WisNode Bridge Serial 1.0.3, PHY V1.0.3 REV A, Over the air a 31 WisNode Bridge Serial is an R5485 ations. The device relays ModBUS dat saly transmitting to and from the end ct website [2]	to LoRaWAN convert a using the LoRaWAN	er designed for indust	
~					
	2. Enter registration data	3			
	DevEUI © * 	99     the owner of the end device. If no App     the owner of the end device     ther for this end device     the secure communication between	EUI is provided by th		r (usually for development), it can
	Register end device				

Figure 56: Registration Data

- 5. Here you must enter the following information:
- Frequency plan Note: For this example, we will choose Europe 863-870 MHz (SF9 for RX2 recommended).
- AppEUI The AppEUI uniquely identifies the owner of the end device. It is provided by the device manufacturer. To get the AppEUI, connect your device via USB cable to your computer. Open RAK Serial Port Tool, choose the correct COM port and BaudRate and run the following command:

RAK SERIAL PORT TOOL		– 🗆 X
	Command	
RAK COM: COM27 Y BaudRate: 115200 Y CLOSE	01 at+version	SEND
RECEIVING CLEAR RECV	_ 02 at+get_config=device:status	SEND
0DEE6D28B846EB17	03 at+set_config=device:sleep:0	SEND
ок	04 at+set_config=device:restart	SEND
	05 at+set_config=device:gps:1	SEND
	06 at+set_config=lora:work_mode:0	SEND
	07 at+set_config=lora:join_mode:0	SEND
	08 at+set_config=lora:class:0	SEND
	09 at+set_config=lora:region:EU868	SEND
	10 at+set_config=lora:confirm:1	SEND
	11 at+set_config=lora:ch_mask:0:0	SEND
	12 at+set_config=lora:dev_eui:	SEND
	13 at+set_config=lora:app_eui:	SEND
	14 at+set_config=lora:app_key:	SEND
	15 at+set_config=lora:dev_addr:	SEND
	16 at+set_config=lora:nwks_key:	SEND
	17 at+set_config=lora:apps_key:	SEND
	18 at+set_config=lora:send_interval:	SEND
SENDING(With \r\n)	□ 19 at+get_config=lora:status	SEND
AT+APPEUI	☑ 20 at+get_config=lora:channel	SEND
SEND		

Figure 57: AppEUI of the device

- **DevEUI** The DevEUI is the unique identifier for this end device. It is provided by the manufacturer and is printed on the label on the back of the device.
- AppKey The root key to deriving session keys to secure communication between the end device and the

application. AppKey can be generated by clicking the Generate button  ${}^{arphi}$  .

• End device ID – The End device ID is automatically filled based on the DevEUI. It can be changed. Note that the end device ID must contain only lowercase letters, numbers, and dashes (-).

#### 📝 NOTE

If you are going to register more than one device of this type, you can choose the option **Register another end device of this type** and be transferred to the same page to register the next device.

6. After filling in the registration information, click Register end device.

### Configuring the Device in OTAA Mode

 For configuring the node you will need the following three parameters: Device EUI, Application EUI, and Application Key. You can see them all in the Device Overview page, but since the two EUI's come with the device, you only need the Application Key from there.

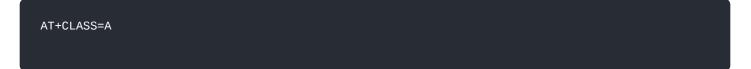
	THE THINOS STACK Community Edition	plications 🔒 Gateway:	s 🔐 Organizations			eu1 Community	RAKwireless •
Rakwireless Test Application		Applications > Rakwireless	Test Application > End devices > rak7431				
Uverview		Fak7431 ID: rak7431					
Live data		Last seen info unavailable     Overview Live data	↑ n/a ↓ n/a Messaging Location Payload formatters Claiming	General settings	Created 2 minutes ago		
<> Payload formatters ✓ ↑ Integrations ✓		General information		Live data	See all activity $\rightarrow$		
Collaborators		End device ID Description	rak7431 🐐				
o⊷ API keys ✿ General settings		Created at	Mar 11, 2021 14:53:38	Waiting for ev	vents from rak7431		
- •		Hardware Brand	rakwireless				
		Model Hardware version	rak7431 1.0	Location	Change location settings →		
		Firmware version	1.2.0				
		Activation information AppEUI	70 83 D5 7E D0 03 C8 C5 🛛 🗘 🐐	No location	information available		
		DevEUI Root key ID	00 10 EE 24 C5 E1 88 E0 🔗 🐐				
		AppKey NwkKey	[ ■ ⊗] n/a				
< Hide sidebar		Session information	11/0				

Figure 58: OTAA device parameters

- 2. Using the RAK Serial Port Tool, set the join mode, device class, and your LoRaWAN region to your correct frequency band, with the following set of AT commands:
- For the join mode (OTAA)



• For the class (Supported classes are: Class A, Class B and Class C. Remember for different classes to change the command with the correct letter, for example for Class B it will be AT+CLASS=B, in this case it is Class A.)



For the region (Remember to replace the frequency band with the one for your LoRaWAN region. Check here
 for your frequency plan.)

AT+REGION=EU868

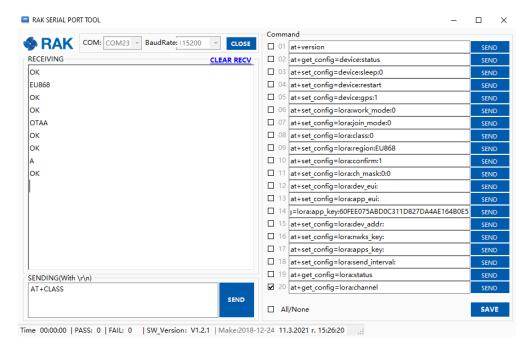


Figure 59: Setting up the RAK7431 WisNode Bridge Serial operation modes

#### 📝 NOTE

The following tutorial is based on using the EU868 frequency band.

3. Now that those parameters are set, enter the **App Key**, using the command below. Remember to replace the **"XXXX"** with the corresponding parameter value for your particular case.

RAK SERIAL PORT TOOL	
KAK SERIAL PORT TOOL	- 🗆 X
RAK COM: COM23 - BaudRate: 115200 - CLOSE	
RECEIVING CLEAR RECV	
ок	□ 03 at+set config=device:sleep:0 SEND
	□ 04 at+set_config=device:restart SEND
	05 at+set_config=device:gps:1 SEND
	06 at+set_config=lora:work_mode:0 SEND
	07 at+set_config=lora:join_mode:0 SEND
	08 at+set_config=lora:class:0 SEND
	09 at+set_config=lora:region:EU868 SEND
-	10 at+set_config=lora:confirm:1 SEND
	11 at+set_config=lora:ch_mask:0:0 SEND
	12   at+set_config=lora:dev_eui:   SEND
	13   at+set_config=lora:app_eui:   SEND
	14 J=lora:app_key:60FEE075ABD0C311D827DA4AE164B0E5 SEND
	15   at+set_config=lora:dev_addr:   SEND
	16 at+set_config=lora:nwks_key:     SEND
	17 at+set_config=lora:apps_key: SEND
	18   at+set_config=lora:send_interval:   SEND
SENDING(With \r\n)	□ 19 at+get_config=lora:status SEND

Figure 60: Setting up the RAK7431 WisNode Bridge Serial OTAA parameters

4. To connect to the LoRaWAN Network after configuration, the device must be restarted. Restart it with the command:

AT+RESTART

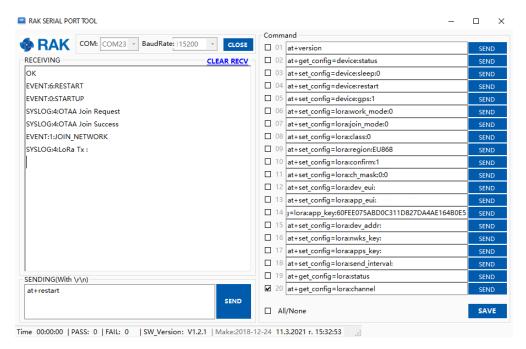


Figure 61: Joining the network confirmation

You can see in the Live data feed that the RAK7431 WisNode Bridge Serial is successfully joined.

	THE THINGS STACK Community Edition	: 📸 Gateways 🎎 Organizations	🜐 eu1 Community	RAKwireless
Rakwireless Test Application				
Overview		<b>rak7431</b> D: nak7431		
Lend devices		■ Last seen 1 minute ago 1 ↓ n/a Crea	ted 41 minutes ago	
Live data		Overview Live data Messaging Location Payload formatters Claiming General settings		
Payload formatters ~	Time Type	Data preview		🔢 Pause 🛛 📋 Cle
t Integrations ∽	$\leftrightarrow$ 15:32:51 Link ADR request enqueued	DevAddr: 26 08 4E 45		
Collaborators	$ \longleftrightarrow \texttt{15:32:51}  \texttt{Device status request enqueued} $	DevAddr: 26 08 4E 45		
🖛 API keys	$\psi$ 15:32:51 Successfully scheduled data downlink .	. DevAddr: 26 08 4E 45		
General settings	$\psi$ 15:32:51 Schedule data downlink for transmissi	DevAddr: 26 08 4E 45 Rxl Delay: 5		
	$\uparrow$ 15:32:51 Forward data message to Application S	DevAddr: 26 08 4E 45 MAC payload: 17 SNR: 8.8 RSSI: -37 Bandwidth: 125000		
	$\uparrow$ 15:32:51 Forward uplink data message	DevAddr: 26 08 4E 45 MAC payload: 1C SNR: 8.8 RSSI: -37 Bandwidth: 125000		
	$\uparrow$ 15:32:51 Receive uplink data message	Dev/Addr: 26 08 4E 45		
	$\uparrow$ 15:32:51 Successfully processed data message	DevAddr: 26 08 4E 45 FCnt: 1 MAC payload: 17 Bandwidth: 125000 SNR: 8.8 RSSI: -37 Raw payload: 40 45 4E 08 26 80 01 00 00 17 66 F3 80	FE	
	$\leftrightarrow 15{:}32{:}51$ . Device time answer enqueued	DevAddr: 26 08 4E 45		
	<⇒15:32:51 Device time request received	DevAddr: 26 08 4E 45		
	↑ 15:32:51 Drop data message	Uplink is a duplicate		
	↑ 15:32:51 Receive data message	DevAddr: 26 08 4E 45 FCnt: 1 MAC payload: 17 Bandwidth: 125000 SNR: 10.2 RSSI: -45 Raw payload: 40 45 4E 08 26 80 01 00 00 17 66 F3 8	9 FE	
	↑ 15:32:51 Drop data message	Uplink is a duplicate		
	↑ 15:32:51 Receive data message	DevAddr: 26 08 4E 45 FCnt: 1 MAC payload: 17 Bandwidth: 125000 SNR: 9.8 RSSI: -42 Raw payload: 40 45 4E 08 26 80 01 00 00 17 66 F3 80	FE	
	↑ 15:32:51 Drop data message	Uplink is a duplicate		
< Hide sidebar	↑ 15:32:51 Receive data message	DevAddr: 26 08 4E 45 FCnt: 1 MAC payload: 17 Bandwidth: 125000 SNR: -9.8 RSSI: -121 Raw payload: 40 45 4E 08 26 80 01 00 00 17 66 F3	30 FE	
	↑ 15:32:51 Receive data message	DevAddx: 26 08 4E 45 FCnt: 1 MAC payload: 17 Bandwidth: 125000 SNR: 8.8 RSSI: -37 Raw payload: 40 45 4E 08 26 80 01 00 00 17 66 F3 80	FE	

Figure 62: Receiving data in the Live data feed