

版本 1.1 | 2020 年 8 月



请访问网站文档下载中心以获取本手册的最新副本。



#### 目录

1. 目的	3
2. LoRaWAN 及 MQTT 简介	3
2.1. LoRaWAN 简述	4
2.2. MQTT 简述	5
2.3. MQTT 在 LoRaWAN 网络中的作用	6
3. 订阅节点数据	7
3.1. 商业网关使用内置 MQTT 服务器	7
3.1.1. 商业网关配置内置 MQTT 服务器	7
3.1.2. 通过 mqtt.fx 订阅节点信息	11
3.1.3. 通过 mqtt.fx 向节点发送信息	17
3.2. 商业网关使用私有 MQTT 服务器	20
3.2.1. 商业网关配置私有的 MQTT 服务器	20
3.2.2. 通过 mqtt.fx 连接私有的 MQTT 服务器	21
4. 通过 MQTT 获取的节点数据格式定义	24
4.1. Uplink	24
4.2. Downlink	25
4.3. Join	26
4.4. Ack	26



	4.5. Device Status	.26
5.	程序示例	.27
6.	修订历史	.31



# 1.目的

本文旨在针对购买 RAK 商业网关产品,并且使用内置 LoRa Server 功能的用户,通过 本文可以帮助用户了解如何通过 MQTT 订阅网关的内置 LoRa Server 得到的数据,使用户 了解商业网关的工作原理,方便用户在自己的应用服务器获取节点数据,达到便捷使用应用 服务器完成数据展示和数据分析的目的。

本文将以 RAK 公司的商业网关产品(室内网关 RAK7258 或者室外网关 RAK7249) 为例,介绍如何使用网关内置 MQTT 服务器和私有(外置) MQTT 服务器;介绍如何订阅 节点数据以及节点数据格式的解析。

# 2.LoRaWAN 及 MQTT 简介

在本章节内容中,我们将介绍 LoRaWAN 网络和 MQTT 网络,使大家对 LoRaWAN 和 MQTT 的工作原理有一个简单直观的了解。如果您已经熟悉了 LoRaWAN 以及 MQTT 的工 作原理,可跳过此章节内容。



## 2.1. LoRaWAN 简述



图 1

如图1所是, LoRaWAN 网络角色分为:

- 终端设备(End Nodes)
- 网关 ( Concentrator/Gateway )
- NS 服务器(Network Server)

LoRaWAN 网络中的角色作用:

终端设备即节点设备,负责数据采集,并将数据加密后以无线信号的形式传递给网关;

网关将终端发送的数据透传给 NS 服务器;

NS 服务器根据数据身份信息、密钥对网关转发的数据进行解密、处理。

RAK 简化 LoraWAN 实际部署条件,商业网关集合了 NS 服务器。配合 RAK 的节点可以更轻松搭建起 LoraWAN 网络。

## 2.2. MQTT 简述

RAK

MQTT 代表 MQ 遥测传输。是一种发布/订阅,极其简单和轻量级的消息传递协议,旨在用于受限设备和低带宽、高延迟或不可靠的网络。设计原则是使网络带宽和设备资源要求最小化,同时还要确保可靠性。这些原则也使该协议成为物联网世界的理想选择。



图 2

如图 2 所是, MQTT 网络角色分为:

- 发布者 (Publisher)
- 订阅者 (Subscriber)
- MQTT Broker

发布者 (Publisher) 发布信息;

订阅者(Subscriber)收集发布者发布的信息;



MQTT Broker 接收发布信息并将信息向订阅者进行展示。

MQTT Broker 同比是新闻发布网站,发布者是新闻发布成员,订阅者是浏览、查看新 闻的用户。

## 2.3. MQTT 在 LoRaWAN 网络中的作用



图 3

实际应用1:

使用 RAK 商业网关内置 MQTT Broker 功能。

RAK 商业网关获得数据并发送给内置 NS,内置 NS 将数据发布至内置 MQTT Broker,

用户通过第三方程序订阅数据。

RAK 商业网关即为发布者,又为 MQTT Broker。

注意:1.使用网关内置的 MQTT Broker,无法通过公网订阅或者发布数据;

2.网关内置的 MQTT Broker 仅适合于项目研发、测试阶段使用,请勿做生产部署使用。



实际应用 2:

不使用 RAK 商业网关内置 MQTT Broker 功能。

RAK 商业网关获得数据并发送给内置 NS,内置 NS 将数据发布至第三方 MQTT Broker。用户通过第三方程序订阅数据。

RAK 商业网关仅为发布者。

## 3.订阅节点数据

本章以 RAK 的商业网关为基础,学习如何通过 mqtt.fx 工具订阅节点上报的数据。

所需工具:mqtt.fx工具(MQTT客户端)。

下载地址: <u>https://mqttfX.jensd.de/indeX.php/download</u>。

### 3.1. 商业网关使用内置 MQTT 服务器

### 3.1.1. 商业网关配置内置 MQTT 服务器

将商业网关设置为 Network Server 模式(在该模式下,商业网关既是网关角色,又

是 NS 服务器角色),并在商业网关中添加一个节点。

在浏览器中输入商业网关的 IP 地址,进入商业网关的 web 界面:

Copyright© Shenzhen RAKwireless Technology Co., Ltd.





默认用户名和密码都为 root。输入密码,点击 Login,进入如下界面:

🏟 RAK						UNSAVED CHAI	IGES: 1 AUTO REFRESH O	N Logout
🎯 Status	Status							
Overview LoRa Packet Logger System Log Firewall	Received	9236	Transmitted	5	Active Nodes	0	Busy Nodes	0
🐼 Network	Duty Cycle	Of the LoRa Char	nnel					
🕅 LoRa Gateway	18:36							100
LoRa Network Server	20:06							
🗞 Services	22:21 23:06							
📰 System	23:51 00:36 01:21 02:06 02:51 03:36 04:21 05:06 05:51 06:36							0

图 5

参考图 6,依次选择 LoRa Gateway,点击 LoRa Packet Forwarder,在 Protocol 中选择 Build-inLoRa Server 协议



🏟 RAK				UNSAVED CHANGES: 1	Logout
🎯 Status	LoRa Packet For	warder LoraWAN Packet Forward	er Configuration		
🐼 Network	Gateway Config	uration			
(ሧ) LoRa Gateway					
LoRa Packet Forwarder	General Setup	Packet Filter GPS Information	1		
LoRa Gateway MQTT Bridge		LoRa Packet forwarder is Enabled	Disable		
LoRa Basic Station		Gateway EUI	60C5A8FFFE74D2FC		
네 LoRa Network Server		Protocol	Build-in LoRa Server		
👶 Services		Push Timeout (ms)	Semtech UDP GWMP Protocol LoRa Gateway MQTT Bridge		
📰 System		Statistic Interval (s)	Build-in LoRa Server		
		Keepalive Interval (s)	5		
		Automatic data recovery	messages are automatically stored when the connection to the server is	lost	
		Auto-restart Threshold	30		
			Packet forwarder will automatically restart when the keepalive timeout excee	ds this threshold. Set '0' to	disable

参考图 7,点击 Save&Apply 保存设置

🏟 RAK		10						UNSAVED	CHANGES: 1 LC	gout	
③ Status		Advance	d Frquency Setup	Mode Switch t	o Standard Mode					^	
🐼 Network											
<sup>(</sup> ል) LoRa Gateway	LoRa Conc	.oRa Concentrator 0									
LoRa Packet Forwarder	Radio Conf	iguration									
Bridge			Radio 0 0	Center Frequency	867500000						
Lull La Da Naturali Camura	Basic Station			Center Frequency	868500000			۲			
Loka Network Server			Minimum Tx Frequency 863000000			•					
💑 Services		Maximum Tx Frequency			87000000						
System			1	LoRaWAN Public							
	Chan. ID	MultiSF 0	MultiSF 1	MultiSF 2	MultiSF 3	MultiSF 4	MultiSF 5	MultiSF 6	MultiSF 7	LoRa std	
	Enable										
	Radio	Radio 1 🗸	Radio 1 🗸	Radio 1 🗸	Radio 0 🗸	Radio 0 🗸	Radio 0 🗸	Radio 0 🗸	Radio 0 🗸	Radio 1 🗸	
	If	-400000	-200000	0	-400000	-200000	0	200000	400000 🖨	-200000	
	Freq.	868.1MHz	868.3MHz	868.5MHz	867.1MHz	867.3MHz	867.5MHz	867.7MHz	867.9MHz	868.3MHz	
	Bandwidth	125 KHz	125 KHz	125 KHz	125 KHz	125 KHz	125 KHz	125 KHz	125 KHz	250 KHz 🔽	
	DataRate	All SF	All SF	All SF	All SF	All SF	All SF	All SF	All SF	SF7 🗸	
								Sav	e & Apply R	eset	

图 7

商业网关内置服务器开启成功之后,我们需要在商业网关中内置服务中增加 Application 和节点。



#### 参考图 8 获取订阅的 topic。

ର୍ତ୍ତ୍ତ RAK			UNSAVED CHANGES: 1	Logout
Status	Application Server Integration			
🐼 Network	Application Server Integration			
(¥) LoRa Gateway				
네 LoRa Network Server	General Setup MQTT Topic template Setup			
Status General	All topics support one or more of the four variable fields a separate domain (the content between the two ), vis a	{{application_ID}} {{application_name}} {{application_EUI}} {{devi separate domain). All topics must contain the {{device_EUI}} variab	ce_EUI}}. The variable fie le field.	ld must be in
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		
		Contains the data and meta-data for an uplink application payload.		
Services	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

图 8

使用 Uplink Topic 订阅节点上报的数据,要求如下:

"application/{{application\_ID}}/device/{{device\_EUI}}/rx",其中,

{{application\_ID}}需要替换成我们实际的 application ID , application ID 参考图 9

获取。{{device\_EUI}}需要替换成我们节点的 device\_EUI, device\_EUI参考图 10 获

取。

ର୍ଭ RAK					UNSAVED CHANGES: 1 LOGOUT
③ Status	LoRa Network Server Ap	plication Overv	iew		
🔞 Network	Applications				
(X) LoRa Gateway					
네 LoRa Network Server	ID Name	Devices	Creation Date	Description	
Status General	2 RAK_mqtt_test	1	Wed Jul 1 01:49:03 2020		Edit Delete
Gateway			Type 1 : Unified A	oplication Key	Add
Global Integration					0.000
i Services					Save & Apply Reset
E System					



Applicatio	on RAK_mqtt_test						
Devices	Application Configur	ation Payload Form	nats	Integrations			
La	st Device name en	Device EUI	Class	Activation mode	Device Address	Link margin	Batt
🗌 1 ne	ever <u>RAK_node_test</u>	00d939567c667e1e	A	otaa	-	-dB	-
Select All	Remove				Rows	per page	10 *
Nevice FHI			0				

替换之后的 topic 如下:

application/2/device/00d939567c667e1e/rx

如果我们要订阅一个应用下的所有节点数据,比如应用2下的所有节点数据,可以使

用以下 topic:

application/2/device/+/rx

如果我们要订阅所有应用下的节点数据,可以使用以下 topic:

application/+/device/+/rx

## 3.1.2. 通过 mqtt.fx 订阅节点信息

打开 mqtt.fx 工具, 主界面如图 11 所示:



				MQTT.fx - 1	.7.1				8
File	Extras He	lp							
Ľ	local mosquit	to	•	Connect	Disconnect				•
P	ublish Subse	cribe Scripts	Broker Status	Log					
2	home/garder	/fountain		Publish		Q050	QoS 1 QoS	2 Retained	

#### 点击图 12 左上角的新建按钮:

		MQTT.I	īx - 1.7.1		- 🛛 🔕
File Extras	Help				
local mos	quitto	- 🔅 Connect	Disconnect		•
Publish Su	bscribe Scripts Broker S	tatus Log			
Switch Fountain	» home/garden/fountain		Y Publish	QoS 0 QoS 1 QoS 2	Retained (97)
Switch Fountain					



在图 13 中红框处输入网关 IP 地址:

			MQTT.fx -	1.7.1			📀
File Extras Help							
172.31.204.37	)	1883	Connect	Disconnect			•
Publish Subscrib	e Scripts	Broker Status	Log				
»				Publish	QoS 0 QoS 1	QoS 2 Re	tained

图 13

输入完网关的 IP 之后,点击图 14 中的 Connect 按钮,最右侧的黑色圆形变成绿色,

说明我们已经成功连接到网关内置的 MQTT 服务器:

www.RAKwireless.com



				MQTT.fx - 1	.7.1					0 🔕
File Extra	as Help									
7 172.3	3 <mark>1.204.37</mark>		1883	Connect	Disconnect					-0
Publish	Subscribe	Scripts	Broker Status	Log						
	»			*		Qo5 0	QoS 1	QoS 2	Retained	() () ()

点击图 15 的 Subscribe 订阅按钮, 在输入框中输入订阅的 topic, 再点击输入框右侧

Subscribe 按钮,开始订阅节点上报的数据:



MQTT.fx - 1.7.1	2 <del></del>	o 📀
File Extras Help		
7 172.31.204.37 1883 Connect Disconnect		<b>-</b>
Publish Subscribe Scripts Broker Status Log		
2 application/2/device/00d939567c667e1e/rx Subscribe 3 QoS 0 QoS 1 QoS 2	Autoscroll	<b>0</b> ;•
application/2/device/00d939567c667e1e/rx Dump Messages Mute Unsubscribe		
Topics Collector S St OS	DUB Retai	ned

使用节点成功 join 之后发送一条数据。我们在节点测发送一条"HelloRakwireless"。

节点需要接收 16 进制的数据,我们将"Hello Rakwireless"转换为 16 进制就是"48656c

6c6f52616b776972656c657373"。



	<b>命令</b>	
RAK 端口: COM4 * 波特率: 115200 * 关係	at+version	发送
会收窗口 <u>清空接收</u>	□ 02 at+get_config=device:status	发送
DTAA Join Start	↑ □ 03 at+set_config=device:sleep:0	发送
[LoRa]:Join Success	04 at+set_config=device:restart	发送
ок	□ 05 at+set_config=device:gps:1	发送
[LoRa]: RUI_MCPS_UNCONFIRMED send success	□ 06 at+set_config=lora:work_mode:0	发送
ок	07 at+set_config=lora:join_mode:0	发送
at+recv=0,-28,7,0	08 at+set_config=lora:class:0	发送
>>at+set_config=lora:confirm:1	09 at+set_config=lora:region:EU868	发送
oRa configure confirm success	□ 10 at+set_config=lora:confirm:1	发送
ок	□ 11 at+set_config=lora:ch_mask:0:0	发送
[LoRa]: RUI_MCPS_CONFIRMED send success	□ 12 at+set_config=lora:dev_eui:00d939567c667e1e	发送
ок	□ 13 at+set_config=lora:app_eui:00d939567c667e1e	发送
at+recv=0,-26,7,0	14 g=lora:app_key:4444e6d479d2b6cba70c62220b6db3e3	发送
[LoRa]: RUI_MCPS_CONFIRMED send success	□ <sup>15</sup> at+set_config=lora:dev_addr:	发送
рк	□ 16 at+set_config=lora:nwks_key:	发送
at+recv=0,-43,7,0	□ 17 at+set_config=lora:apps_key:	发送
	v □ 18 at+join	发送
送窗口(默认发送回车)	□ 19 at+get_config=lora:status	发送
at+send=lora:1:48656c6c6f52616b776972656c657373	20 at+get_config=lora:channel	发送
发送	□ 全选/全不选	保存

图 16

在 mqtt.fx 界面可以看到我们订阅到的节点数据,如图 17 所示:

	MQTT.fx - 1.7.1		0 🔕
File Extras Help			
<b>5</b> 172.31.204.37	1883 Connect Disconnect		<b>•</b>
Publish Subscribe Scripts Broker S	tatus Log		
application/2/device/00d939567c667e1e/rx	Subscribe	QoS 0 QoS 1 QoS 2 Autoscroll	() () ()
application/2/device/00d939567c667e1e/rx	application/2/device/00d939567c667e1e/rx		1 Qo50
Topics Collector (0) Scan Stop Co	pplication/2/device/00d939567c667e1e/rx 20-05-2020 10:12:07.36727867		1 Qo5 0
	{"applicationID":"2","applicationName" c667e1e","deviceName":"RAK_node_test", rt":1,"data":"SGVsbG9SYWt3aXJlbGVzcw==	:"RAK_mqtt_test","devEUI":"00d9 "timestamp":1589940727,"fCnt":2 ","data_encode":"base64"}	39567 ,"fPo

图 17



data 部分就是节点发送的数据。Data 部分的内容是对节点发送的数据进行了 base6

4 编码,我们只需要对 data 部分进行 base64 解码,即可看到原始数据。

查看商业网关 web 管理页面,看到的数据与 mqtt.fx 一致。

evice-00d	939567c667e1	e		
Overview	Configuration	Activation	Downlink	Live Device Data
2020/05/20	0 10:12:10 Uplink	48 65 6c	6c 6f 52 61 6b	) 77 69 72 65 6c 65 73 73
{	antionID", "2"			
( "appli "appli	.cationID": "2", .cationName": "RAM	K_mqtt_test",		
{ "appli "appli "devEU	.cationID": "2", .cationName": "RAM II": "00d939567c6	K_mqtt_test", 67ele",	ŝ	
<pre>{     "appli     "appli     "devEU     "devic     "evice</pre>	.cationID": "2", .cationName": "RAJ II": "00d939567c6 :eName": "RAK_node	K_mqtt_test", 67ele", e_test",	ŝ	
<pre>{     "appli     "appli     "devEU     "devic     "times     "fCnt"</pre>	.cationID": "2", .cationName": "RAJ UI": "00d939567c6 weName": "RAK_node utamp": 158994072 ; 2.	K_mqtt_test", 67ele", e_test", 7,	: :	
( "appli "devEU "devic "times "fCnt" "fPort	.cationID": "2", .cationName": "RA II": "00d93956766 eName": "RAK_nod ttamp": 158994072' ': 2, ": 1,	K_mqtt_test", 67ele", e_test", 7,		
( "appli "devEU "devic "times "fCnt" "fPort "data"	.cationID": "2", .cationName": "RAI II": "00d939567c6/ reName": "RAK_nod tamp": 158994072' ': 2, ': 1, : "1, : " <u>SGV=bG35VWt3a</u> ]	K_mqtt_test", 67ele", e_test", 7, XJ1bGVzcw==",		
<pre>{     "appli     "appli     "devEU     "devic     "times     "fCnt"     "fPort     "data"     "data_</pre>	.cationID": "2", .cationName": "RAI II": "00d939567c6/ reName": "RAK_nod ttamp": 158994072' 2: 2, 2: 1, 2: "5GV=bG45YBt3a] encode": "base64'	K_mqtt_test", 67ele", e_test", 7, XJ1bGVzcw==",		

图 18

## 3.1.3. 通过 mqtt.fx 向节点发送信息

通过 mqtt.fx 给节点发送数据,需要使用 Downlink Topic。

🏟 RAK			URSAVED CHANGES: 1	Logout
🕑 Status	Application Server Integration			
🐼 Network	Application Server Integration			
(ሧ) LoRa Gateway				
네 LoRa Network Server	General Setup MQTT Topic template Setup			
Status General	All topics support one or more of the four variable fields ( a separate domain (the content between the two ), it is a	{{application_ID}} {{application_name}} {{application_EUI}} { separate domain). All topics must contain the {{device_EUI}} variab	ce_EUI}}. The variable fiel le field.	d must be in
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		
		Contains the data and meta-data for an uplink application payload.		
🤣 Services	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.		
			1	
			Save & Apply	Reset

图 19



我们需要将 Downlink Topic 中的{{application\_ID}}和{{device\_EUI}}更换为节

点对应的 Appplication ID 和节点的 Device EUI。Appplication ID 的获取参考图 20,

Device EUI 的获取参考图 21。

ର୍ତ୍ତ୍ତ RAK					UNSAVED	CHANGES 1 LC	ogout
Status	LoRa Network Server Appli	cation Overview					
🐼 Network	Applications						
<sup>(</sup> ጆ <sup>)</sup> LoRa Gateway							
네 LoRa Network Server	ID Name	Devices	Creation Date	Description			
Status General	2 RAK_mqtt_test	1	Wed Jul 1 01:49:03 2020			Edit D	elete
Gateway			Type 1 : Unified Appl	ication Key		Add	
😸 Services					5	Save & Apply	Reset
🔳 System							
Application	Edit - 2	E	ন্থ 20				
Devices	Application Configur	ration P	ayload Formats	Integrations			
Last seer	Device name	Device EU	II Clas	Activation mode	Device Address	Link margin	Batter
🗌 1 neve	er <u>RAK_node_test</u>	00d93956	7c667ele A	otaa	÷	-dB	-
Select All	Remove		0		Rows	per page	10 *

图 21

如图 22 所示,点击 mqtt.fx 左上角的 Publish 标签,在输入框 2 的位置输入 Downlink Topic , 在 输 入 框 3 的 位 置 输 入 {"confirmed": true,"data": "SGVsbG8=","fPort":10},点击按钮 4 的 Publish 按钮,即可将数据发送到节点(注: class c 模式节点会立刻收到 mqtt.fx 下发的数据; class a 模式节点会在下一次上发数据 之后收到 mqtt.fx 下发的数据)。



MQTT.fx - 1.7.1	🛛
File Extras Help	
7         172.31.204.37         1883         Connect         Disconnect	🖬 🔴
Publish <sup>1</sup> Subscribe Scripts Broker Status Log	
» application/2/device/00d939567c667e1e/tx 2 v Publish 4 Qos0 Qos1 Qos2       ["confirmed": true,"data": "SGVsbG8=","fPort": 10]   3	Retained Cor

'{"confirmed": true,"data": "SGVsbG8=","fPort": 10}'格式说明

- a. Confirmed 可选值为 true 或者 false。
- b. data 的内容就是我们要发送的数据,需要对数据进行 base64 编码。
- c. fPort 是要发送的端口号,有效端口号为 1-255。

如图 23 所示,可以在节点端看到接收到的数据:



	- <u> </u>	-
RAK 端口: COM4 → 波特率: 115200 → <del>X</del> 闭	01 at+version	发送
收窗口 清空接收	02 at+get_config=device:status	发送
t+recv=155,-26,8,5:48656c6c6f	□ 03 at+set_config=device:sleep:0	发送
and a second	04 at+set_config=device:restart	发送
	05 at+set_config=device:gps:1	发送
	06 at+set_config=lora:work_mode:0	发送
	07 at+set_config=lora:join_mode:0	发送
	08 at+set_config=lora:class:2	发送
	09 at+set_config=lora:region:EU868	发送
	10 at+set_config=lora:confirm:1	发送
	11 at+set_config=lora:ch_mask:0:0	发送
	12 at+set_config=lora:dev_eui:00d939567c667e1e	发送
	13 at+set_config=lora:app_eui:00d939567c667e1e	发送
	☐ 14 g=lora:app_key:4444e6d479d2b6cba70c62220b6db3e3	发送
	15 at+set_config=lora:dev_addr:	发送
	□ 16 at+set_config=lora:nwks_key:	发送
	17 at+set_config=lora:apps_key:	发送
	☐ 18 at+join	发送
送窗口(默认发送回车)	19 at+get_config=lora:status	发送
	☑ 20 at+get_config=lora:channel	发送
した 一般	□ 全选/全不选	保存

## 3.2. 商业网关使用私有 MQTT 服务器

本节将说明如何在网关配置私有的 MQTT 服务器,如何通过 mqtt.fx 从私有 MQTT 服

务器订阅信息。

本节讲述以用户名/密码方式访问私有 MQTT。

3.2.1. 商业网关配置私有的 MQTT 服务器

在浏览器打开商业网关的 web 管理页面,参考图 24 配置:





- MQTT Broker Address 处填写用户 MQTT 服务器的 IP 地址
- MQTT Broker Port 处填写 MQTT 服务的端口号,该端口号如果用户没有更改的 话默认为 1883
- 打开 Enable User Authentication 开关
- 输入访问 MQTT 服务的用户名和密码

配置完成之后,点击右下角的 Save&Apply 按钮保存更改。

## 3.2.2. 通过 mqtt.fx 连接私有的 MQTT 服务器

打开 mqtt.fx,点击图 25 所示设置图标:



### 瑞科慧联科技有限公司



图 25

点击图 26 左下角的加号,新建一个 Profile,输入 Profile Name,配置 MQTT 服务器的 IP 地址和端口,在 User Credentials 标签下输入用户名和密码。配置完成之后点击 右下角的 OK 按钮:



M2M Eclipse New Profile local mosquitto	Profile Name Custom_MQTT Profile Type MQTT Broker
	VIQTT Broker Profile Settings
	Broker Port 1883 Client ID MQTT_FX_Client
	General User Credentials SSL/TLS Proxy 🝷
	User Name rakwireless Password
	Burst Of Anto
	Kevert OK Apply

图 26

如图 27 所示,选择我们刚刚创建的 Profile,点击 Connect 按钮,即可成功连接到私

File Extras Help			~
Custom_MQTT	Connect Disconnect		ſ
Publish Subscribe Scripts Bro	ker Status Log		Ŭ
application/2/device/00d939567c667e1e/rx	Subscribe	QoS 0 QoS 1 QoS 2 Autoscroll	0° •
Topics Collector S S S			
	-		
	图 27		
	<u>isi</u> Z /		

有的 MQTT 服务器上。订阅与发布可参考上一小节 3.1 商业网关内置 MQTT 服务器



# 4. 通过 MQTT 获取的节点数据格式定义

在前面章节我们了解了通过 MQTT 如何订阅节点的上行数据 Uplink 以及如何向节点下 发数据 Downlink ,节点还有其他三种类型的数据 ,分别是 Join、Ack 和 Status。其中 Join 就是节点入网时的信息 , Ack 是向节点下发数据之后节点回复的确认信息 , Status 是节点 的电池电量信息。

下面我们将分别介绍这五种数据的具体格式以及含义。

## 4.1. Uplink

RAK

#### {

"applicationID": "1", // 节点所属应用的 id

"applicationName": "test-app", // 节点所属应用的名称

"devEUI": "3637343457387e11", // 节点的 devEUI

"deviceName": "dev-5205", // 节点名称

"timestamp": 1592730721, // 接收到节点数据的 Unix 时间戳

"fCnt": 6,

"fPort": 2,

"data": "AQIDBA==", // base64 编码后的数据, 解码之后就是节点实际上发的数据

"data\_encode": "base64", // 数据的编码类型

"adr": true, // 节点是否开启了 adr

"rxInfo":[ // 所有接收到节点数据的网关信息

RAK 瑞科慧联科技有限公司

```
🧐 RAK
```

{

```
"gatewayID": "d896e0fff010611e", // 网关的 gateway_id
       "loRaSNR": 13.3, // 当前网关的信噪比
       "rssi": -71, // 当前网关的 RSSI
       "location": { // 对应网关的经纬度以及海拔信息
          "latitude": 0,
          "longitude": 0,
          "altitude": 0
       }
   }
],
"txInfo": {
   "frequency": 486300000, // 节点发送数据使用的的频率
   "dr": 2 // 节点当前的 data rate
}
```

## 4.2. Downlink

```
{
```

}

"confirmed": true, // This dl pkt need confirm or not.

"fPort": 2, // The port will be used for sending this packet

"data": "AgAAAA==" // 发送给节点的数据 , 经过 base64 编码

}



```
4.3. Join
```

{

```
"applicationID": "1",
```

"applicationName": "test-app",

"deviceName": "dev-5205",

```
"devEUI": "3637343457387e11",
```

"devAddr": "02000001" // Join 成功之后分配给节点的短地址

}

## 4.4. Ack

注意:服务器只有给节点下发了 confirmed 类型的数据后,节点才会回复 Ack(注意:

节点不一定立即回复 Ack, Ack 可能会在节点下一次发送上行数据是携带。)。

```
{
```

```
"applicationID": "1",
```

"applicationName": "test-app",

```
"deviceName": "dev-5205",
```

```
"devEUI": "3637343457387e11",
```

"acknowledged": true,

"fCnt": 7

}

## 4.5. Device Status

```
{
```

"applicationID": "1",

"applicationName": "test-app",

"deviceName": "dev-5205",



"devEUI": "3637343457387e11",

"battery": 254, // 电池剩于电量的分级。254 表示电源满电状态 ,1 表示电池电 量即将耗尽。

"margin": 8, // 是最近一次成功接收 DevStatusReq 命令的解调信噪比

"externalPowerSource": false, // 是否使用了额外的电源

"batteryLevelUnavailable": false, // 节点的电量级别是否有效

"batteryLevel": 100 // batteryLevelUnavailabl 为 true 的情况下, batteryLevel 表示电量百分比

#### }

## 5.程序示例

以下是使用 python 代码调用 mqtt 订阅节点的上发数据并将对应的内容打印出来,每 收到一条上行数据会,程序会主动向节点发送一个下行数据,内容是"Hello rak"。使用代 码前请仔细阅读代码注释。

以下源码基于 python3 运行环境,在运行代码前,需要使用命令 pip3 install paho-mqtt 安装依赖库。

#!/usr/bin/env python

import json import base64 import paho.mqtt.client as mqtt from datetime import datetime

# mqtt 服务器 IP



```
mqtt_ip = '111.230.247.253'
```

RAK

# mqtt 服务器端口

mqtt\_port = 1883

# mqtt 用户名

mqtt\_username = 'rakwireless'

# mqtt 密码

mqtt\_password = 'rakwireless.com'

# mqtt 订阅 topic。该 topic 可以订阅所有节点信息

mqtt\_rx\_topic = 'application/+/device/+/rx'

#### # 将字符串转换为 16 进制

```
def str_to_hex(s):
```

```
return r"\x"+r'\x'.join([hex(ord(c)).replace('0x', '') for c in s])
```

#### # 一旦订阅到消息,回调此方法

#### # 打印订阅到的节点信息

```
def on_print_node_rx_info(json_rx):
   try:
       devEUI
                       = json rx['devEUI']
                           = json rx['applicationID']
       applicationID
       applicationName = json_rx['applicationName']
                           = json rx['deviceName']
       deviceName
       timestamp
                       = json rx['timestamp']
       fCnt
                       = json_rx['fCnt']
       fPort
                       = json rx['fPort']
       data
                       = json_rx['data']
       data hex
                       =
str_to_hex(base64.b64decode(data).decode("utf-8"))
```

```
# 将时间戳转换为本地时间
```

str\_local\_time = datetime.fromtimestamp(timestamp)

print('-----' % devEUI:[%s] rxpk info ------' % devEUI)
print('+\t applicationName:\t%s' % applicationName)



```
print('+\t applicationID:\t\t%s' % applicationID)
       print('+\t deviceName:\t\t%s' % deviceName)
       print('+\t datetime:\t\t%s' % str local time)
       print('+\t fCnt:\t\t\%d' % fCnt)
       print('+\t fPort:\t\t\%d' % fPort)
       print('+\t data:\t\t\t%s' % data)
       print('+\t data_hex:\t\t%s' % data_hex)
       print('-----')
   except Exception as e:
       print(e)
   finally:
       pass
# 订阅到节点的数据之后,向节点发送"Hello rak"字符串
def on print rak node info(payload):
   json str = payload.decode()
   try:
       ison rx = ison.loads(ison str)
       on_print_node_rx_info(json_rx)
       dev eui = json rx['devEUI']
       app id = json rx['applicationID']
       # 商业网关默认的 tx topic
       tx topic = 'application/%s/device/%s/tx' % (app_id, dev_eui)
       str hello = "Hello Rak"
                        '{"confirmed":true,"fPort":10,"data":"%s"
       tx msq
                  =
                                                                  - }'
                                                                         %
str(base64.b64encode(str hello.encode("utf-8")), "utf-8")
       # 发布消息
       mgttc.publish(tx topic, tx msg, gos=0, retain=False)
       print('Send \'Hello rak\' to node %s' % dev_eui)
```

```
except Exception as e:
raise e
finally:
pass
```



mqttc = mqtt.Client()
mqttc.on\_message = on\_message

# 如果没有用户名和密码,请注释改行 mqttc.username\_pw\_set(mqtt\_username, password=mqtt\_password)

# 连接 mqtt broker , 心跳时间为 60s mqttc.connect(mqtt\_ip, mqtt\_port, 60)

mqttc.subscribe(mqtt\_rx\_topic, 0)

mqttc.loop\_forever()

www.RAKwireless.com



# 6.修订历史

版本	描述	日期
1.0	创建文档	2020-07-03
1.1	修改文档	2020-08-17



#### 关于瑞科慧联:

RAK 是一家专注于 IoT 领域以产品为驱动型的公司,凭借团队深厚的无线通讯技术领域的积累,采 用创新的商业模式高效地为全球中小型的网络运营商(Network Operator) , 系统集成商(System Integrator)和服务提供商(Service Provider)等提供高性能的 IoT 产品与应用方案。