

Table of Contents

Registering your Gateway	
Registering your Busylight	3
Testing the Busylight	9
Controlling the Busylight with http Requests	

ChirpStack allows the addressing of the Busylight LoRa devices with https – requests.

We will describe how to send commands to your LoRa Busylight in various ways.

Registering your Gateway

First log in into your Chipstack Admin GUI.

If you have a Gateway with built-in Network Server, the Gateway may already be registered.



Click on Gateways / "+ Create":



a	NigStack Application Server	x +	-		1			
	C A Not secure	192.168.8.20.8080/#/organizations/1/gateways/create	2 🛧	*	0			
E	ChirpStack	Q. Search organization, application, gateway or device	0	Θ	admi			
	Dashboard	Gateways / Create						
	Network-servers							
	Gateway-profiles	GENERAL TAOS METADATA						
	Organizations							
	All users	Oateway name *						
API keys								
rp	stack +	Gateway description *						
	Ora, dashboard							
	Ora. users	24mm 24						
	Ora API keya	Gatoway to *	,	MSB	G			
	Cog. No Congle	Nexuel-arve* Belect network-server						
	Dervice-promes	Select the neuronicenser to which the gateway will connect. When no network-servers are available in the displacen, make sure a service-profile exists for this organization.						
	Device-profiles	teenny porte Select gatemay-profile						
	Galenays	Optional. When satisfying a generary-profile to the generary OhipDrack Network Server will enterpt to update the generary according to the generary-profile. Note that this does require a generary with OhipDrack Concentrated						
	Apprications	Underway on scovery enabled						
	Multicast-groups	Datavary altitude (metara) * D						
		When the gateway has an or-board GPS, this value will be set automatically when the network has received attrictics from the gateway.						
		Gateway location (set to current location)						
		9						
		Diag the marker to the location of the gateway then the gateway has an an-based GPS, this value will be set automatically when the network reasons atteined a from the gateway.	OpenSteel	wap cores	Duffer			
		ADD BOARD CONFIGURATION	CREAT		WAY			

The Gateway Name and the description can be set as you would like defined.

The Gateway ID must match the ID of your gateway. In most cases, it is the MAC Address of the LoRa radio.

If everything is OK, you will see a recent 'Last seen' entry in the Gateway list.

æ	ChirpStack Application Serve	r X	+					~	-		×
←	→ C A Not secu	ire 192	.168.8.20:8080/#/	organizatior/	ns/1/gateways			Q	☆	* 🕑	:
	ChirpStack								?	e admi	n
^	Dashboard Network-servers	G	iateways							+ CREATE	E
_ ® ₽	Gateway-profiles Organizations	Γ	Last seen	Name	Gateway ID	Network server	Gateway activity (30d)				
• «	All users API keys	L	a few seconds ago	rak- gateway	ME140#Webschwi	build_in_ns				I	
chirp	pstack 👻						Rows per page: 10 👻	1-1 of 1		< >	
ħ	Org. dashboard										
*	Org. users										
٩	Org. API keys										
<u>.</u> ≡	Service-profiles										
클는	Device-profiles										
R	Gateways										
	Applications										
2	Multicast-groups	-									



Registering your Busylight

First you need to create a device profile. You need to create a device profile once and can use it for all of your Busylight devices.

In the ChirpStack Console, click on Device Profiles, then "+ Create".

You can copy the entries from these screen shots:

<i>a</i> 0	⊘ ChirpStack Application Server x +								
← →	C A Not secure 192.1	58.8.20:8080/#/organizations/1/device-profiles/d54b4aa4-6f3f-4102-b741-41be8d173d49 🛠 🗍 😢	1						
€	ChirpStack	Q Search organization, application, gateway or device ? e admin							
A	Dashboard	Device-profiles / device_profile_busylight							
	Network-servers								
R	Gateway-profiles	GENERAL JOIN (OTAA / ABP) CLASS-B CLASS-C							
Ð	Organizations	Device-profile name *							
*	All users	device_profile_busylight A name to identify the device-profile.							
٩	API keys	LoRaWAN MAC version *							
chirp	ostack 👻	1.0.3 The LoRaWAN MAC version supported by the device.							
A	Org. dashboard	LoRaWAN Regional Parameters revision *							
*	Org. users	Revision of the Regional Parameters specification supported by the device.							
٩	Org. API keys	Max EIRP * 4							
. ≡	Service-profiles	Maximum EIRP supported by the device.							
	Device-profiles	Uplink: interval (seconds) * 30							
Ŵ	Gateways	The expected interval in seconds in which the device sends uplink messages. This is used to determine if a device is active or inactive.							
	Applications								
٣	Multicast-groups	UPDATE DEVICE-PROFILE							





(28 CF (← →	hirpStack Application Server ×	+ 8.8.20:8080/#/organizations/1/device-profiles/d54b4aa4-6f3f-4102-b741-41be8d173d49	✓ - □ × ★ ★ ₽ :
€	ChirpStack	Q Search organization, application, gateway or device	? 👌 admin
^	Dashboard Network-servers	Device-profiles / device_profile_busylight	DELETE
© #	Gateway-profiles Organizations All users	GENERAL JOIN (OTAA / ABP) CLASS-B	CLASS-C
٩	API keys	UPD	ATE DEVICE-PROFILE
chirp	ostack -		
A	Org. dashboard		
*	Org. users		
٩	Org. API keys		
≜ ≡	Service-profiles		
	Device-profiles		
R	Gateways		
	Applications		
2	Multicast-groups		
			✓ - □ ×
Ch Ch	irpStack Application Server 🗙 -	+	

← -	C A Not secure 192.10	58.8.20:8080/#/organizations/1/device-profiles/d54b4aa4-6f3f-4102-b741-41be8d173d49 🖈 🕑
€	ChirpStack	Q Search organization, application, gateway or device ? e admin
A	Dashboard	Device-profiles / device_profile_busylight
R	Network-servers Gateway-profiles	
	Organizations	GENERAL JUIN (UIAA / ADP) CLASS-D CLASS-C
•	All users	Device supports Class-C Select this option when the device will operate as Class-C device immediately after activation. In case it sends a
٩	API keys	DeviceModelnd mac-command when it changes to Class-C, do not select this option. Class-C confirmed downlink timeout *
chirp	ostack 👻	3 There & Known & Concern Difference & constitute to provide inter-
f	Org. dashboard	Class-U timeout (in seconds) for confirmed downlink transmissions.
*	Org. users	UPDATE DEVICE-PROFILE
٩	Org. API keys	
.≞≡	Service-profiles	
	Device-profiles	
R	Gateways	
	Applications	
2	Multicast-groups	





You need to enter a Payload Encoder to let ChirpStack code the hardware payload bytes.

Here is the Encoder to copy into your own ChirpStack instance:

```
// Encode encodes the given object into an array of bytes.
// - fPort contains the LoRaWAN fPort number
// - obj is an object, e.g. {"temperature": 22.5}
// - variables contains the device variables e.g. {"calibration": "3.5"} (both the
key / value are of type string)
// The function must return an array of bytes, e.g. [225, 230, 255, 0]
function Encode(fPort, obj, variables) {
  return [ (obj.red & 0x00FF), (obj.blue & 0x00FF), (obj.green & 0x00FF),
  (obj.ontime & 0x00FF), (obj.offtime & 0x00FF) ];
}
```

Next, you need to create a ChirpStack Application. Please Click on Applications and then to "+ Create" in the ChirpStack Console.

You can choose Application Name and Application Description as you would like it defined.



<i>a</i> 0	hirpStack Application Server 🗙 -	+ ~ - □ ×						
÷ -	C A Not secure 192,16	8.8.20:8080/#/organizations/1/applications/create						
€	ChirpStack	Q Search organization, application, gateway or device 🤪 😝 admin						
÷	Dashboard	Applications / Create						
	Network-servers							
R	Gateway-profiles							
▦	Organizations	Application name *						
•	All users	ine name may only contain words, numbers and dashes.						
٩	API keys Application description *							
chir	ostack 👻	Service-profile* service-profile-build-in						
f	Org. dashboard	The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.						
*	Org. users	CREATE APPLICATION						
٩	Org. API keys							
.≞≡	Service-profiles							
	Device-profiles							
R	Gateways							
	Applications							
2	Multicast-groups							

After Creation, please enter the new Application by clicking on your Application name.

You will see it in the device list.

<i>a</i> 0	nirpStack Application Server 🗙 🕂	F						\sim	-		×
\leftarrow	C A Not secure 192.16	8.8.20:80	080/#/organiz	ations/1/a	pplications/2				☆	* 6	
€	ChirpStack		Q Se	arch orga	anization, applicati	ion, gateway o	or device	?		e admi	n
A	Dashboard	A	oplication	s / ap	D				Î	DELETI	=
	Network-servers										
R	Gateway-profiles		DEVICE	s	APPLICATION CO	NFIGURATION	INTEGRATI	ONS		FUO	ТА
	Organizations										
•	All users								+	CREAT	:
٩	API keys		Last	Device					Link		
chir	ostack 👻		seen	name	Device EUI	D	evice profile		mar	gin	3attery
A	Org. dashboard		a few seconds	busylig	ht1 302030410	#256700 d	evice_profile_busy	rlight	n/a	1	n/a
*	Org. users		ago								
٩	Org. API keys					Rows per page	n: 10 v 1-1	of 1	<	>	
<u>_</u> ≡	Service-profiles										
	Device-profiles										
R	Gateways										
	Applications										
2	Multicast-groups										
4									_		

To Add a Busylight Device, click on "+ Create".

For registering your Busylight device, you need this information:

DeviceEUI	(8 Byte Hex)
AppEUI	(8 Byte Hex)
АррКеу	(16 Byte Hex)



First, you need to create the device itself:

<i>2</i> 8 CH	nirpStack Application Server 🗙	+ · · · · ×
$\leftarrow \frac{1}{2}$	C A Not secure 192.1	68.8.20:8080/#/organizations/1/applications/2/devices/create 🖈 🖈 🕑 🗄
∉	ChirpStack	Q Search organization, application, gateway or device 🕜 😝 admin
•	Dashboard	Applications / app / Devices / Create
®	Gateway-profiles	GENERAL VARIABLES TAGS
•	Organizations All users	Device name *
٩	API keys	The name may only contain words, numbers and dashes.
chirp	ostack 👻	Device description *
A	Org. dashboard	Device EUI * MSB C
•	Org. users	Device_profile * device_profile_busylight
s. ∡≡	Org. API keys Service-profiles	Disable frame-counter validation Note that disabling the frame-counter validation will compromise security as it enables people to perform
11	Device-profiles	replay-attacks.
Ŵ	Gateways	Device is disabled ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.
<i>۳</i>	Applications Multicast-groups	CREATE DEVICE

You can use appropriate entries for Device name and Device description here.

Please enter the DeviceEUI of your Busylight and select the device profile you have created earlier.

After creation, enter the device details by clicking on the device name:



Ø Ch ← →	irpStack Application Server × -	+ 8.8.20:8080/#/organizations/1/applications/2/devices/2010101	04250702	✓ - □ ★ ★ ₽
€	ChirpStack	Q Search organization, application, ga	teway or device	? 😫 admin
↑	Dashboard Network-servers	Applications / app / Devices / busylig	ht1	DELETE
\bigcirc	Gateway-profiles	DETAILS CONFIGURATION	KEYS (OTAA)	ACTIVATION >
₽	Organizations			
•	All users	Details	Status	
٩	API keys	Name busylight1	Last seen at	Oct 8, 2021 1:50 PM
chirp	stack 👻	Description Busylight	State	enabled
ħ	Org. dashboard	Device-		
•	Org. users	profile device_profile_busylight		
٩	Org. API keys			
<u>≜</u> ≡	Service-profiles	Enqueue downlink payload		
井	Device-profiles			
R	Gateways	Port *		
	Applications	Please note that the fPort value must be > 0.		
9	Multicast-groups	Confirmed downlink		
		BASE64 ENCODED JSON OBJECT		
		Base64 encoded string *		
				ENQUEUE PAYLOAD
		Downlink queue		C 🚺
		FCnt FPort Confirmed	Base64 encoded	payload

Please click on KEYS (OTAA).



Please click on "SET DEVICE KEYS" and enter your devices' APPKey.



You can now power up your Busylight, it will go to green light and you can see the status in the ChirpStack Console as well:

Ø Ch	⊘ ChirpStack Application Server x + × × × ×								
← →	C A Not secure 192.168.8.	20:8080/#/organiz	ations/1/applica	ations/2		0- 7 ☆	* 🕑 E		
∉	ChirpStack	۹	Search organ	ization, application, gat	eway or device	0	admin		
ŧ	Dashboard	Application	s / app			Ĩ	DELETE		
-	Network-servers								
\mathbb{R}	Gateway-profiles	DEVICE	S APF	LICATION CONFIGURATI	ON INTEGRATIONS		FUOTA		
	Organizations						005475		
•	All users					+	CREATE		
٩	API keys	Leat	Davias			Link			
chirp	ostack 👻	seen	name	Device EUI	Device profile	margin	Battery		
ŧ	Org. dashboard	18 minutes	busylight1	202020410a250702	device_profile_busylight	n/a	n/a		
*	Org. users	ago							
٩	Org. API keys	a few seconds	busylight2	2020204137160602	device_profile_busylight	n/a	n/a		
<u>*</u> ≡	Service-profiles	ago							
	Device-profiles			Rows p	er page: 10 👻 1-2 of 2	2 <	>		
\bigcirc	Gateways								
	Applications								
2	Multicast-groups								

Testing the Busylight

You can test your connected Busylight by sending the downlink payload from the ChirpStack console.

Please open the details page for your device.



÷ →	C A Not secure 192.16	8.8.20:8080/#/organizations/1/applications/2/devices/2	104250702	x 🛪 🖻
∉	ChirpStack	Q Search organization, application, ga	teway or device	? 🕒 admir
ħ	Dashboard	Applications / app / Devices / busylig	ht1	DELETE
	Network-servers			
R	Gateway-profiles	DETAILS CONFIGURATION	KEYS (OTAA)	ACTIVATION >
	Organizations			
•	All users	Details	Status	
٩,	API keys	Name busylight1	Last seen at	Oct 8, 2021 1:50 PM
chirp	stack 👻	Description Busylight	State	enabled
ħ	Org. dashboard	Device-		
•	Org. users	profile		
٩	Org. API keys			
<u>e</u> ≡	Service-profiles	Enqueue downlink payload		
	Device-profiles			
R	Gateways	Port *		
	Applications	Please note that the fPort value must be > 0.		
2	Multicast-groups	Confirmed downlink		
		BASE64 ENCODED JSON OBJECT		
		Base64 encoded string *		
				ENQUEUE PAYLOAD
		Downlink queue		C 🔋
		FCnt FPort Confirmed	Base64 encoded p	payload

The Port needs to be set to 15. They Payload needs to be specified in Base64 or JSON notation.

For a first test, enter this string into the Base64 payload:

AP8A/wA=

In this case, the Busylight will be solid blue.

Please have a look into the chapter about the hardware payload format.

You can use this JSON object for testing:

```
{ "red":255, "blue":255, "green":0, "ontime":255, "offtime":0 }
```

In this case, the Busylght will be solid purple.



Controlling the Busylight with http Requests

If you plan to control the Busylight with http/https requests, you need to create an API Key.

To create a key, open the application and click on "Org. API Keys".

2 Chirp	🖉 ChirpStack Application Server 🗙 🕂							
← → C ▲ Not secure 192.168.8.20:8080/#/organizations/1/api-keys/create								
∉	ChirpStack	Q Search organization, application, gateway or device	?		3 admi	n		
^	Dashboard Network-servers	Organization API keys / Create						
$\widehat{\mathbb{N}}$	Gateway-profiles	API key ID						
	Organizations	APT Ney 10 682377fd-45d4-4274-b8b2-3004d0f1c95b APT Ney name XXX Token						
•	All users							
٩	API keys							
chirpstack -		eyJhbGci0iJlUz11NilsInR5cCl6lkpXVCJ9.eyJhcGlfa2V5X2lkljoiNjgyMzc3ZmQtNDVkNC00M jc0LWl4YjltMzAwNGQwZjFj0TViliwiYXVkljoiYXMiLCJpc3Mi0iJhcyIsIm5iZil6MTYzMzY5NT						
A	Org. dashboard	MxNiwic3ViljoiYXBpX2tleSJ9.5CL9Bn1Bbry-8PL3zhDTSg7itG3V-y4JzXicznAlS44						
•	Org. users							
٩	Org. API keys							
* =	Service-profiles							
	Device-profiles							
\bigotimes	Gateways							
	Applications							
2	Multicast-groups							

Please copy the token now - you will need it for your http requests.

You are now ready to write control your Busylight with a HTML request. Please change the yellow highlighted text parts to your to match your setup.

You need to send a POST request to this URI:

```
https://your_chirpstack_server:8080/api/devices/your_device_eui/queue
You send this Body:
{
    "deviceQueueItem": {
        "confirmed": true,
        "jsonObject": "{ \"red\":255,
        "blue\":255,\"green\":0,\"ontime\":255,\"offtime\":0 }",
        "fPort": 15
    }
}
```

```
And you need to send these Headers:
```



Grpc-Metadata-Authorization: Bearer your_api_key Content-Type: application/json User-Agent: busylight/v1

Here is a PowerShell example that switches a Busylight to solid blue:

```
$body2 = '{
    "deviceQueuEItem": {
        "confirmed": true,
        "jsonObject": "{ \"red\":255, \"blue\":255, \"green\":0, \"ontime\":255, \"offtime\":0 }",
        "fPort": 15
    }
}'

$headers = @{`
        "Content-Type" = "application/json";
        "User-Agent" = "busylight/1";`
        "Grpc-Metadata-Authorization" = "Bearer eromodecontinent and action action and action a
```

Busylight LoRa Hardware Payload format

The Busylight expects a 5-byte binary payload for switching the colors.

Byte 0: Red Color intensity (0..255) Byte 1: Blue Color intensity (0..255) Byte 2: Green Color intensity (0..255) Byte 3: On Steps (0..255) Byte 4: Off Steps (0..255) Example for blue static light:

Byte[0]=0 Byte[1]=255 Byte[2]=0 Byte[3]=255 Byte[4]=0

The base64 form will be: AP8A/wA=

For ChirpStack https operating, if using the data property to specify the payload for the end device, the byte array needs to be send as a base64 encoded string.

When using the payload formatter, you can specify the values using a json string to put into the jsonObject property:

```
{
    "deviceQueueItem": {
        "confirmed": true,
        "jsonObject": "{ \"red\":255,
        "blue\":255,\"green\":0,\"ontime\":255,\"offtime\":0 }",
        "fPort": 15
    }
}
```