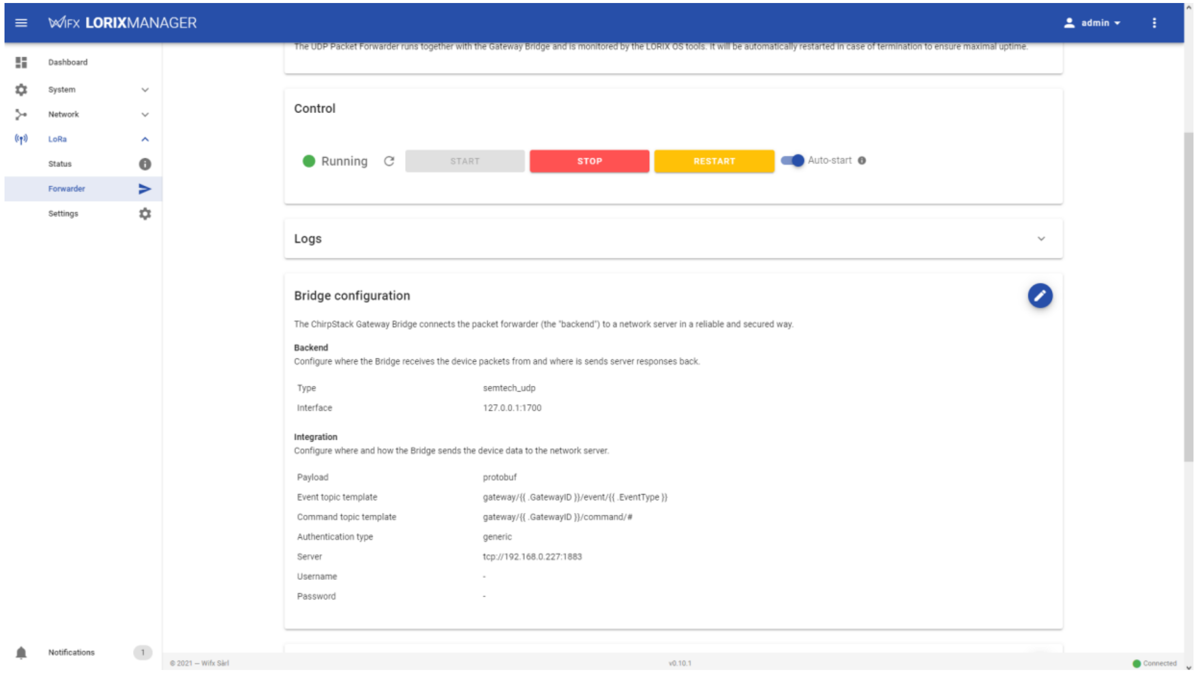
Mise en place de la maquette

Paramétrage de l’antenne

- Lancement du service LORIXMANAGER en accédant à &l’adresse IP de l’antenne

- Mise en place dans les paramètres de l’adresse du serveur



The UDP Packet Forwarder runs together with the Gateway Bridge and is monitored by the LORIX OS tools. It will be automatically restarted in case of termination to ensure maximal uptime.

Bridge configuration

The ChirpStack Gateway Bridge connects the packet forwarder (the "backend") to a network server in a reliable and secured way.

Backend
Configure where the Bridge receives the device packets from and where is sends server responses back.

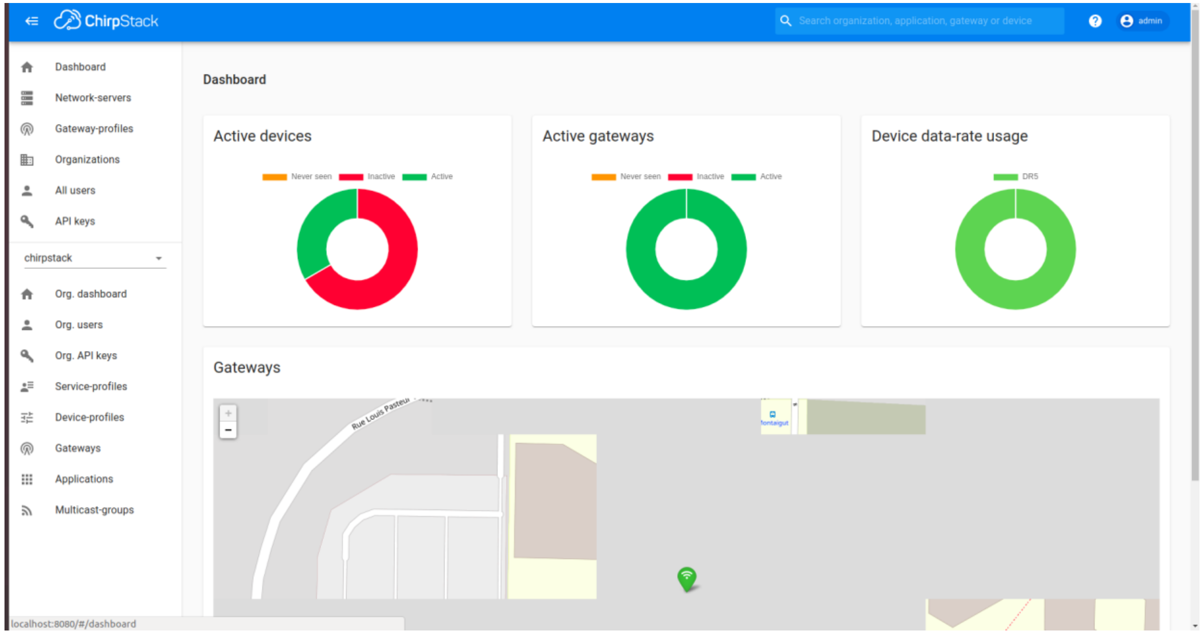
Type	semtech_udp
Interface	127.0.0.1:1700

Integration
Configure where and how the Bridge sends the device data to the network server.

Payload	protobuf
Event topic template	gateway/{{ .GatewayID }}/event/{{ .EventType }}
Command topic template	gateway/{{ .GatewayID }}/command/#
Authentication type	generic
Server	tcp://192.168.0.227:1883
Username	-
Password	-

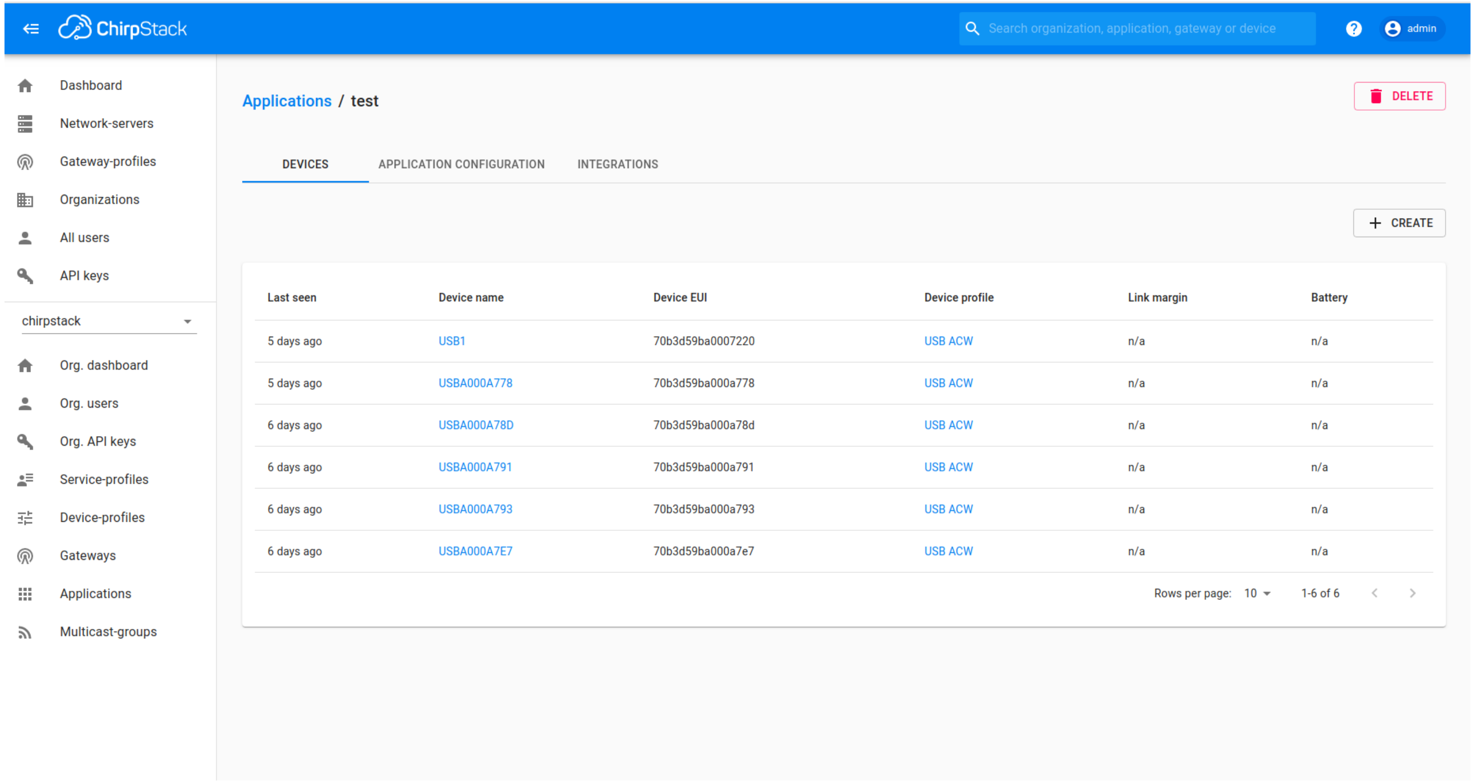
Paramétrage du server

- Mise en place des bases de données pour l’application et le réseau
- Lien avec l’antenne grâce a l’outil « ChirpStack »



Mise en place de l’application

- Enregistrement des cartes sur le serveur (device EUI)



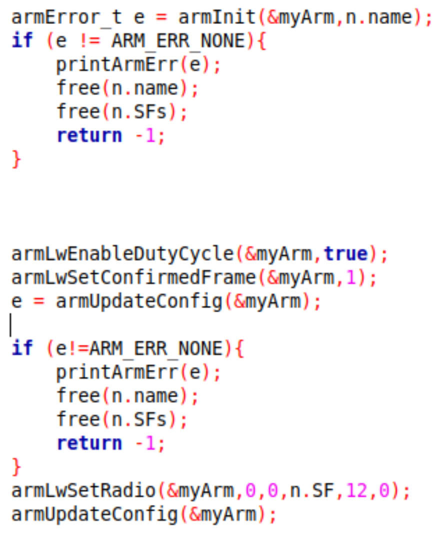
Applications / test

Last seen	Device name	Device EUI	Device profile	Link margin	Battery
5 days ago	USB1	70b3d59ba0007220	USB ACW	n/a	n/a
5 days ago	USBA000A778	70b3d59ba000a778	USB ACW	n/a	n/a
6 days ago	USBA000A78D	70b3d59ba000a78d	USB ACW	n/a	n/a
6 days ago	USBA000A791	70b3d59ba000a791	USB ACW	n/a	n/a
6 days ago	USBA000A793	70b3d59ba000a793	USB ACW	n/a	n/a
6 days ago	USBA000A7E7	70b3d59ba000a7e7	USB ACW	n/a	n/a

Rows per page: 10 1-6 of 6

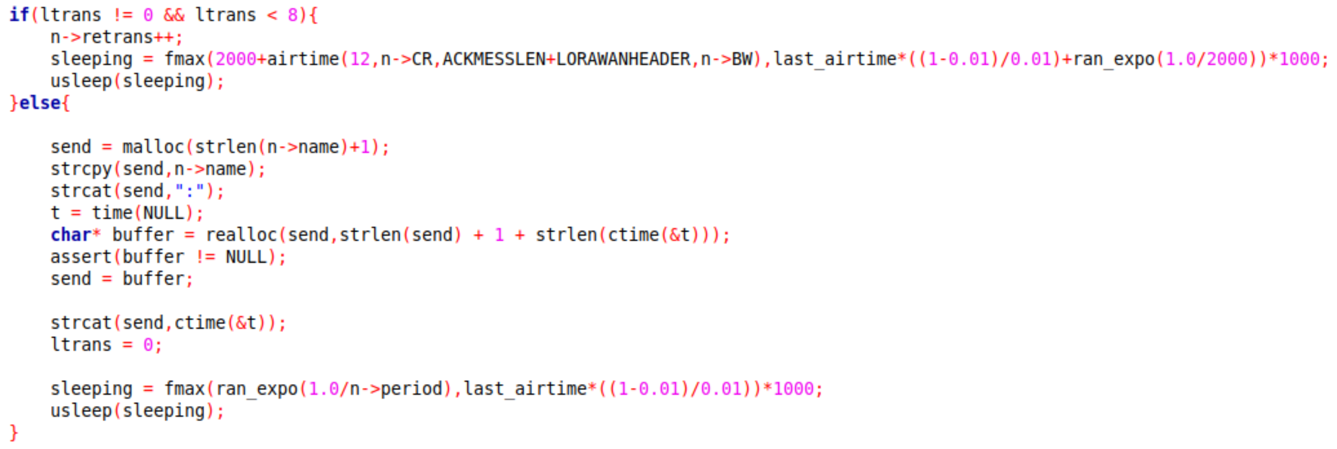
Code d’envoi

- Librairie C fourni par ATIM utilisant des commandes AT
- 2 arguments : nombre de nœuds et paramètres des nœuds (période, minSF, durée de vie)
- Initialisation des cartes avec demande d’acquittement



```
armError_t e = armInit(&myArm,n.name);
if (e != ARM_ERR_NONE){
    printArmErr(e);
    free(n.name);
    free(n.SFs);
    return -1;
}


armLwEnableDutyCycle(&myArm,true);
armLwSetConfirmedFrame(&myArm,1);
e = armUpdateConfig(&myArm);

if (e!=ARM_ERR_NONE){
    printArmErr(e);
    free(n.name);
    free(n.SFs);
    return -1;
}
armLwSetRadio(&myArm,0,0,n.SF,12,0);
armUpdateConfig(&myArm);
```

Boucle d’envoi respectant les 8 retransmissions maximums et les temps d’attente.

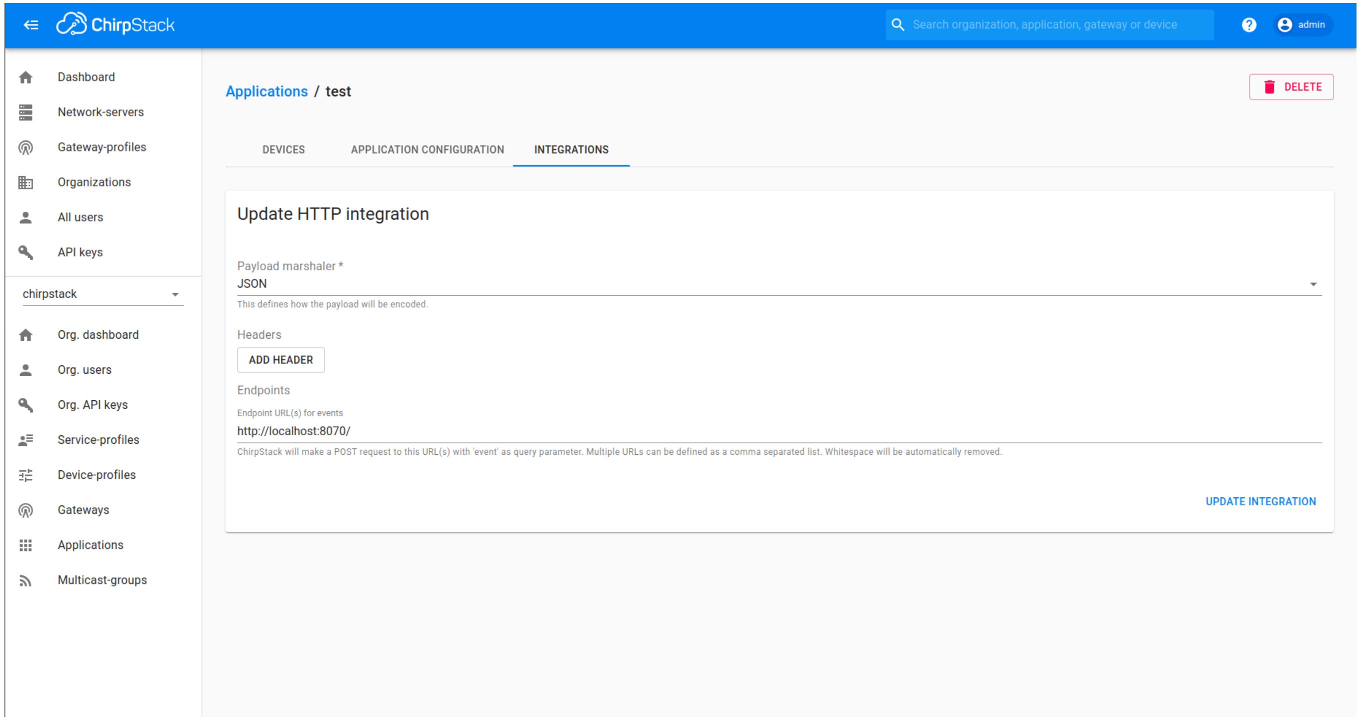


```c
if(ltrans != 0 && ltrans < 8){
    n->retrans++;
    sleeping = fmax(2000+airtime(12,n->CR,ACKMESSLEN+LORAWANHEADER,n->BW),last_airtime*((1-0.01)/0.01)+ran_expo(1.0/2000))*1000;
    usleep(sleeping);
}else{

    send = malloc(strlen(n->name)+1);
    strcpy(send,n->name);
    strcat(send,":");
    t = time(NULL);
    char* buffer = realloc(send,strlen(send) + 1 + strlen(ctime(&t)));
    assert(buffer != NULL);
    send = buffer;

    strcat(send,ctime(&t));
    ltrans = 0;

    sleeping = fmax(ran_expo(1.0/n->period),last_airtime*((1-0.01)/0.01))*1000;
    usleep(sleeping);
}
```

Récupération du nombre d’envoi, de retransmissions et d’acquittements

Intégration du serveur HTTP

- Récupération des événements au niveau de l'antenne concernant les cartes enregistrées.



Applications / test

Update HTTP integration

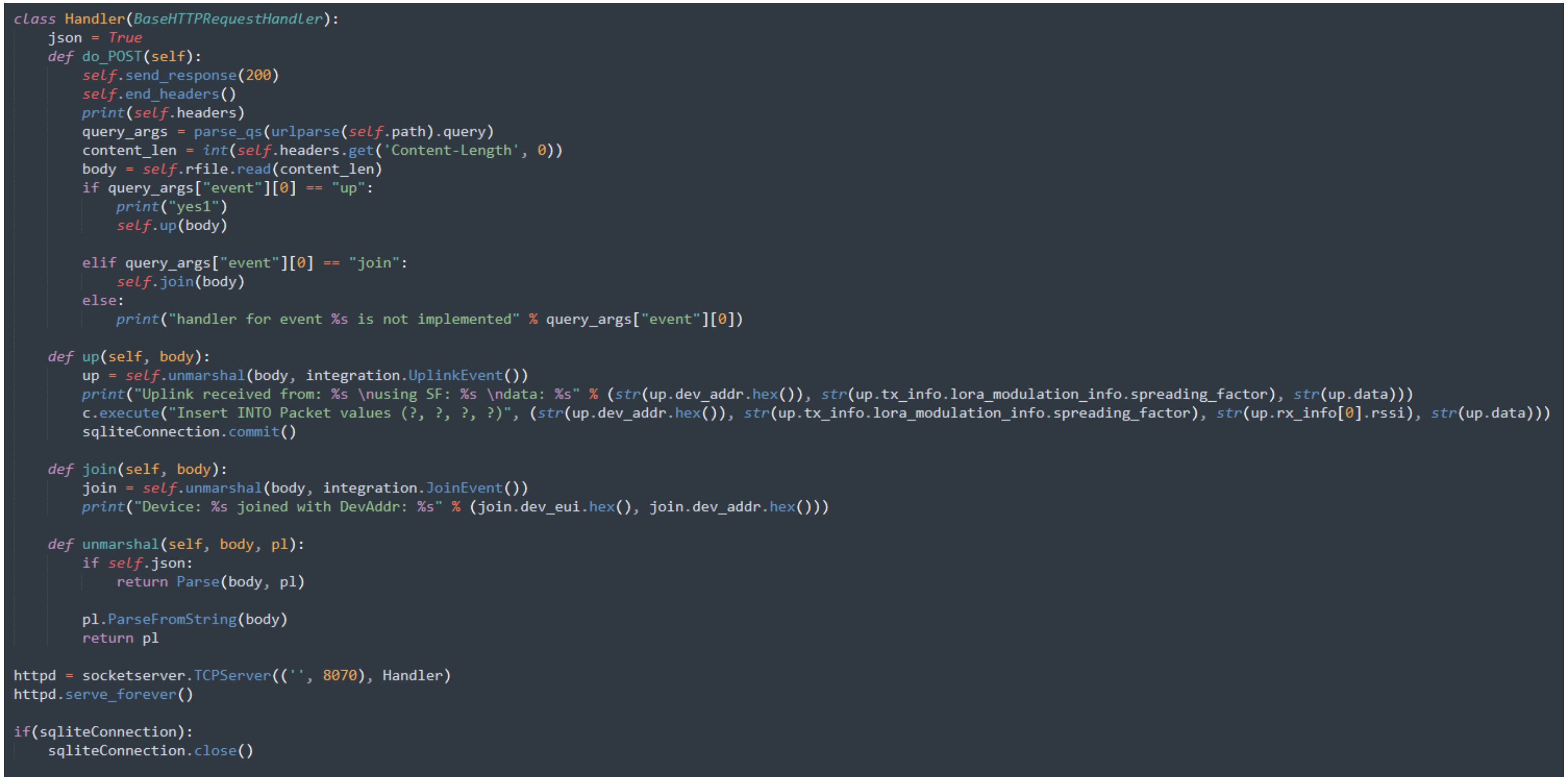
Payload marshaler *
JSON
This defines how the payload will be encoded.

Headers

Endpoints

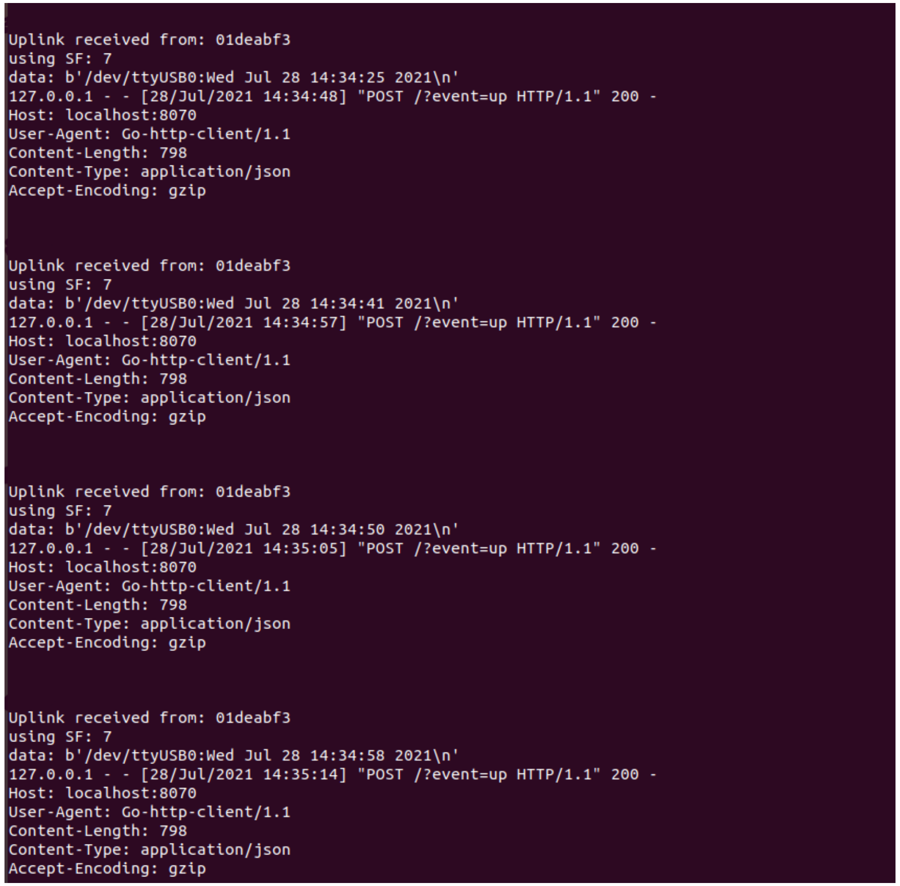
Endpoint URL(s) for events
http://localhost:8070/
ChirpStack will make a POST request to this URL(s) with 'event' as query parameter. Multiple URLs can be defined as a comma separated list. Whitespace will be automatically removed.

- Un code python qui permet de lancer un serveur http sur le port souhaitée.



```python
class Handler(BaseHTTPRequestHandler):
    json = True
    def do_POST(self):
        self.send_response(200)
        self.end_headers()
        print(self.headers)
        query_args = parse_qs(urlparse(self.path).query)
        content_len = int(self.headers.get('Content-Length', 0))
        body = self.rfile.read(content_len)
        if query_args["event"][0] == "up":
            print("yes1")
            self.up(body)

        elif query_args["event"][0] == "join":
            self.join(body)
        else:
            print("handler for event %s is not implemented" % query_args["event"][0])

    def up(self, body):
        up = self.unmarshal(body, integration.UplinkEvent())
        print("Uplink received from: %s \nusing SF: %s \ndata: %s" % (str(up.dev_addr.hex()), str(up.tx_info.lora_modulation_info.spreading_factor), str(up.data)))
        c.execute("Insert INTO Packet values (?, ?, ?, ?)", (str(up.dev_addr.hex()), str(up.tx_info.lora_modulation_info.spreading_factor), str(up.rx_info[0].rssi), str(up.data)))
        sqliteConnection.commit()

    def join(self, body):
        join = self.unmarshal(body, integration.JoinEvent())
        print("Device: %s joined with DevAddr: %s" % (join.dev_eui.hex(), join.dev_addr.hex()))

    def unmarshal(self, body, pl):
        if self.json:
            return Parse(body, pl)

        pl.ParseFromString(body)
        return pl

httpd = socketserver.TCPServer(('', 8070), Handler)
httpd.serve_forever()

if(sqliteConnection):
    sqliteConnection.close()
```

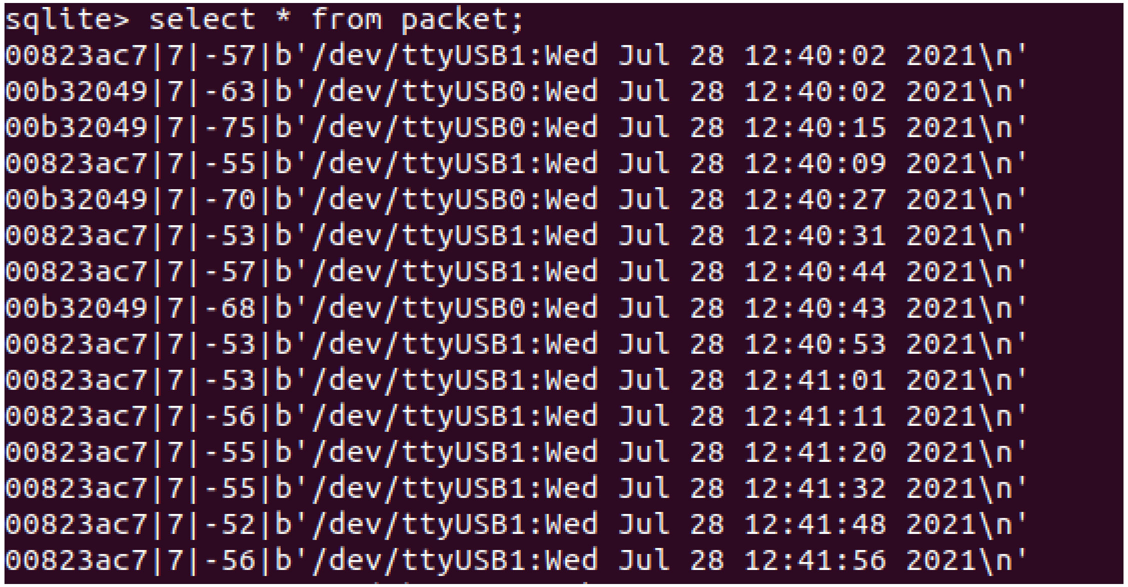
- Chaque paquets reçus par l’antenne est afficher sur le serveur http grâce à l’intégration avec l’antenne



```
Uplink received from: 01deabf3
using SF: 7
data: b'/dev/ttyUSB0:Wed Jul 28 14:34:25 2021\n'
127.0.0.1 - - [28/Jul/2021 14:34:48] "POST /?event=up HTTP/1.1" 200 -
Host: localhost:8070
User-Agent: Go-http-client/1.1
Content-Length: 798
Content-Type: application/json
Accept-Encoding: gzip



Uplink received from: 01deabf3
using SF: 7
data: b'/dev/ttyUSB0:Wed Jul 28 14:34:41 2021\n'
127.0.0.1 - - [28/Jul/2021 14:34:57] "POST /?event=up HTTP/1.1" 200 -
Host: localhost:8070
User-Agent: Go-http-client/1.1
Content-Length: 798
Content-Type: application/json
Accept-Encoding: gzip



Uplink received from: 01deabf3
using SF: 7
data: b'/dev/ttyUSB0:Wed Jul 28 14:34:50 2021\n'
127.0.0.1 - - [28/Jul/2021 14:35:05] "POST /?event=up HTTP/1.1" 200 -
Host: localhost:8070
User-Agent: Go-http-client/1.1
Content-Length: 798
Content-Type: application/json
Accept-Encoding: gzip



Uplink received from: 01deabf3
using SF: 7
data: b'/dev/ttyUSB0:Wed Jul 28 14:34:58 2021\n'
127.0.0.1 - - [28/Jul/2021 14:35:14] "POST /?event=up HTTP/1.1" 200 -
Host: localhost:8070
User-Agent: Go-http-client/1.1
Content-Length: 798
Content-Type: application/json
Accept-Encoding: gzip
```

- - Chaque paquet reçus par le server http est stocker dans la base de données.

- - La base de données contient une table Paquet(devAddr, SF, RSSI, data).



```
sqlite> select * from packet;
00823ac7|7|-57|b'/dev/ttyUSB1:Wed Jul 28 12:40:02 2021\n'
00b32049|7|-63|b'/dev/ttyUSB0:Wed Jul 28 12:40:02 2021\n'
00b32049|7|-75|b'/dev/ttyUSB0:Wed Jul 28 12:40:15 2021\n'
00823ac7|7|-55|b'/dev/ttyUSB1:Wed Jul 28 12:40:09 2021\n'
00b32049|7|-70|b'/dev/ttyUSB0:Wed Jul 28 12:40:27 2021\n'
00823ac7|7|-53|b'/dev/ttyUSB1:Wed Jul 28 12:40:31 2021\n'
00823ac7|7|-57|b'/dev/ttyUSB1:Wed Jul 28 12:40:44 2021\n'
00b32049|7|-68|b'/dev/ttyUSB0:Wed Jul 28 12:40:43 2021\n'
00823ac7|7|-53|b'/dev/ttyUSB1:Wed Jul 28 12:40:53 2021\n'
00823ac7|7|-53|b'/dev/ttyUSB1:Wed Jul 28 12:41:01 2021\n'
00823ac7|7|-56|b'/dev/ttyUSB1:Wed Jul 28 12:41:11 2021\n'
00823ac7|7|-55|b'/dev/ttyUSB1:Wed Jul 28 12:41:20 2021\n'
00823ac7|7|-55|b'/dev/ttyUSB1:Wed Jul 28 12:41:32 2021\n'
00823ac7|7|-52|b'/dev/ttyUSB1:Wed Jul 28 12:41:48 2021\n'
00823ac7|7|-56|b'/dev/ttyUSB1:Wed Jul 28 12:41:56 2021\n'
```

Conclusion

- Réalisation d’une maquettes avec du matériel

- Réalisation d’une plus grande maquette avec un plus grand nombre de cartes

- Faire des tests de performances on utilisant la maquette