

# DOKUMENTATION

LoRaWAN-Starterkit Innotas

**INNOTAS ELEKTRONIK GMBH**



5 März 2019

Verfasst von: H. Töpfer/S. Sieber

# 1 Inhaltsverzeichnis

2	REVISIONSVERZEICHNIS .....	1
3	VERWENDUNGSZWECK .....	2
4	AUFBAU.....	3
4.1	HARDWARE .....	4
4.2	. SOFTWARE .....	4
4.3	INBETRIEBNAHME DES LORAWAN-STARTERKITS .....	5
4.3.1	INSTALLATION DER KOMPONENTEN UND SCHNELLSTART .....	5
4.4	BETREIBEN DES LORAWAN-SERVERS KIT .....	7
4.4.1	DASHBOARD.....	7
4.4.2	MENÜ / NAVIGATION.....	8
4.4.3	GATEWAYS .....	9
4.4.4	PROFILE .....	11
4.4.5	DEVICE .....	12
4.4.6	HANDLER .....	13
4.4.7	CONNECTORS.....	14
5	ANBINDUNG VON EXTERNEN GATEWAYS .....	15
6	WEITERE LINKS UND QUELLEN .....	16
7	LORAAPPLICATIONS TOOL.....	17
7.1	FUNKTION UND AUTOMATISCHER BETRIEB.....	18

## 2 Revisionsverzeichnis

REVISION	DATUM	ÄNDERUNG
1.0	5 März 2019	Erstausgabe
1.1	04.04.2019	Schnellstartanleitung zugefügt und ergänzt

# DOKUMENTATION

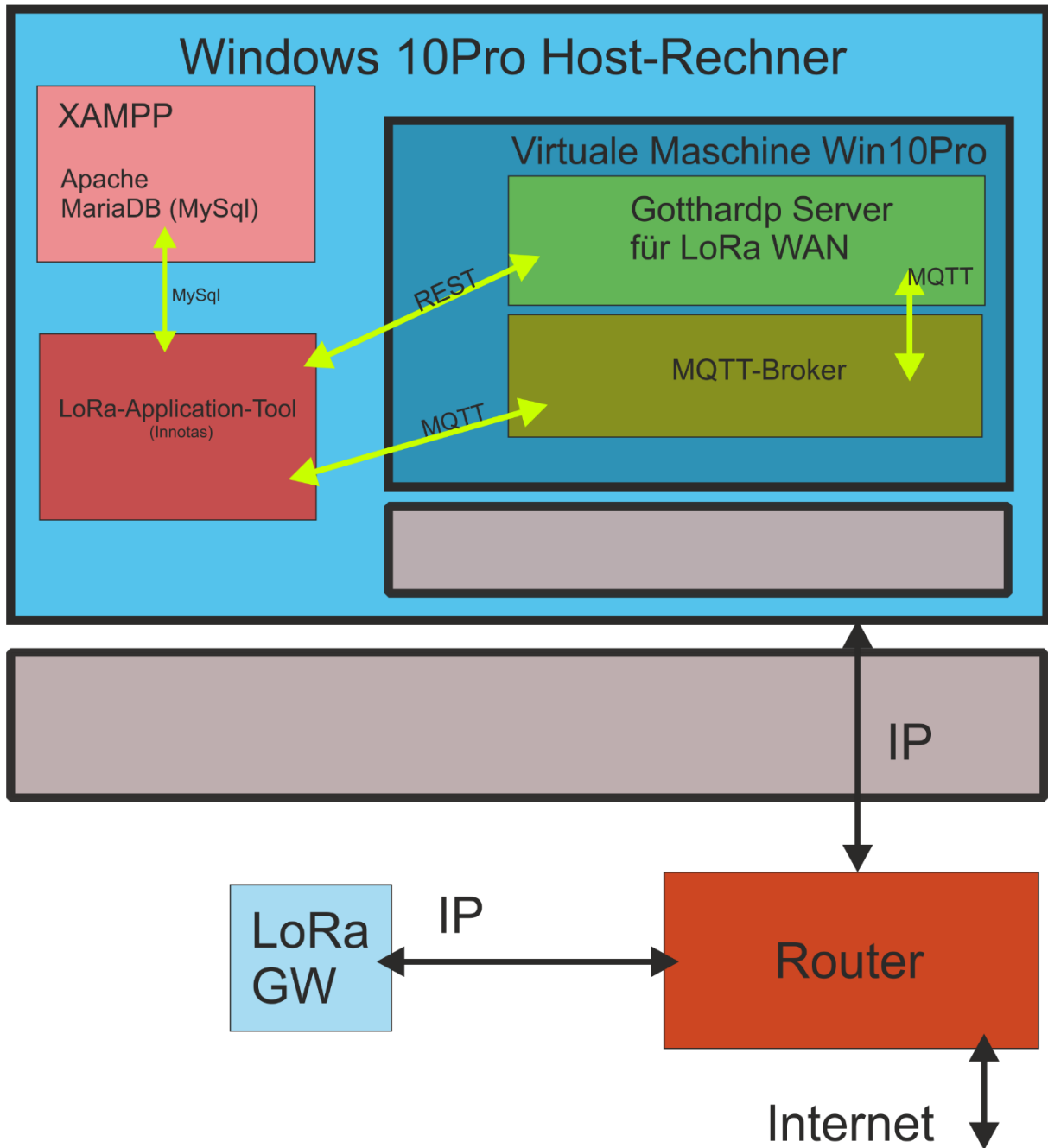
---

## LoRaWAN-Starterkit Innotas

### 3 Verwendungszweck

Das LoRaWAN-Starterkit von Innotas erlaubt den einfachen und schnellen Aufbau eines LoRa-WAN Netzwerkes. Dabei wird dem Kunden die gesamte, notwendige Hard- und Software zum Betreiben des Servers zur Verfügung gestellt. Das Starterkit wird mit mindestens einem LoRa-Gateway ausgeliefert. Er kann einfach um weitere Gateways erweitert werden, wobei Gateways verschiedener Anbieter angeschlossen werden können

## 4 Aufbau



## 4.1 Hardware

Hauptbestandteil der Hardware sind:

Windows-Notebook (Windows 10 basierend)

LoRa-Gateway mit POE-Stromversorgung

Ethernet-Router

Das System ist vorkonfiguriert, die IP-Adressen werden über DHCP von einem Router festgelegt. LoRa-Host 192.168.88.254 und Lora-Server 192.168.88.21 sind fest eingestellt. Es wird ein kleines, lokales Netz gebildet, welches ohne externen Internetanschluss in Betrieb genommen werden kann. Der Nutzer muss keine Konfigurationen vornehmen, das System arbeitet automatisch.

Zu Testzwecken können an den Router weitere LoRa-Gateways angeschlossen werden, was aber auf Grund der dann räumlichen Nähe der Gateways kaum praktischen Nutzen bringt.

Es ist auch möglich, weitere LoRa-Gateways von extern an das „Inselnetz“ anzubinden. Dazu ist das „Inselnetz“ an das Internet anzuschließen und das externe LoRa-Gateway kann das Internet zur Verbindung mit dem LoRa-Server nutzen. Es werden Konfigurationsschritte notwendig, welche im Punkt „Anbindung von externen Gateways“ beschrieben sind.

Das von Innotas mitgelieferte Gateway besitzt als Stromversorgung „Power over Ethernet“, d.h. zum Gateway muss nur das Netzkabel gezogen werden, eine Stromversorgung in unmittelbarer Nähe wird nicht benötigt. Dadurch kann das Gateway funktechnisch günstig platziert werden, in der Regel gilt, je höher der Anbauort, umso besser die Empfangsbedingungen. Das Gateway ist für den Outdoor-Einsatz konzipiert, kann also auch auf dem Dach angebaut werden. Dabei sind die gesetzlichen Vorschriften, insbesondere der Blitzschutz einzuhalten. Der Router dient als DHCP Server und möglichen Verbindungspunkt zum Internet.

## 4.2 Software

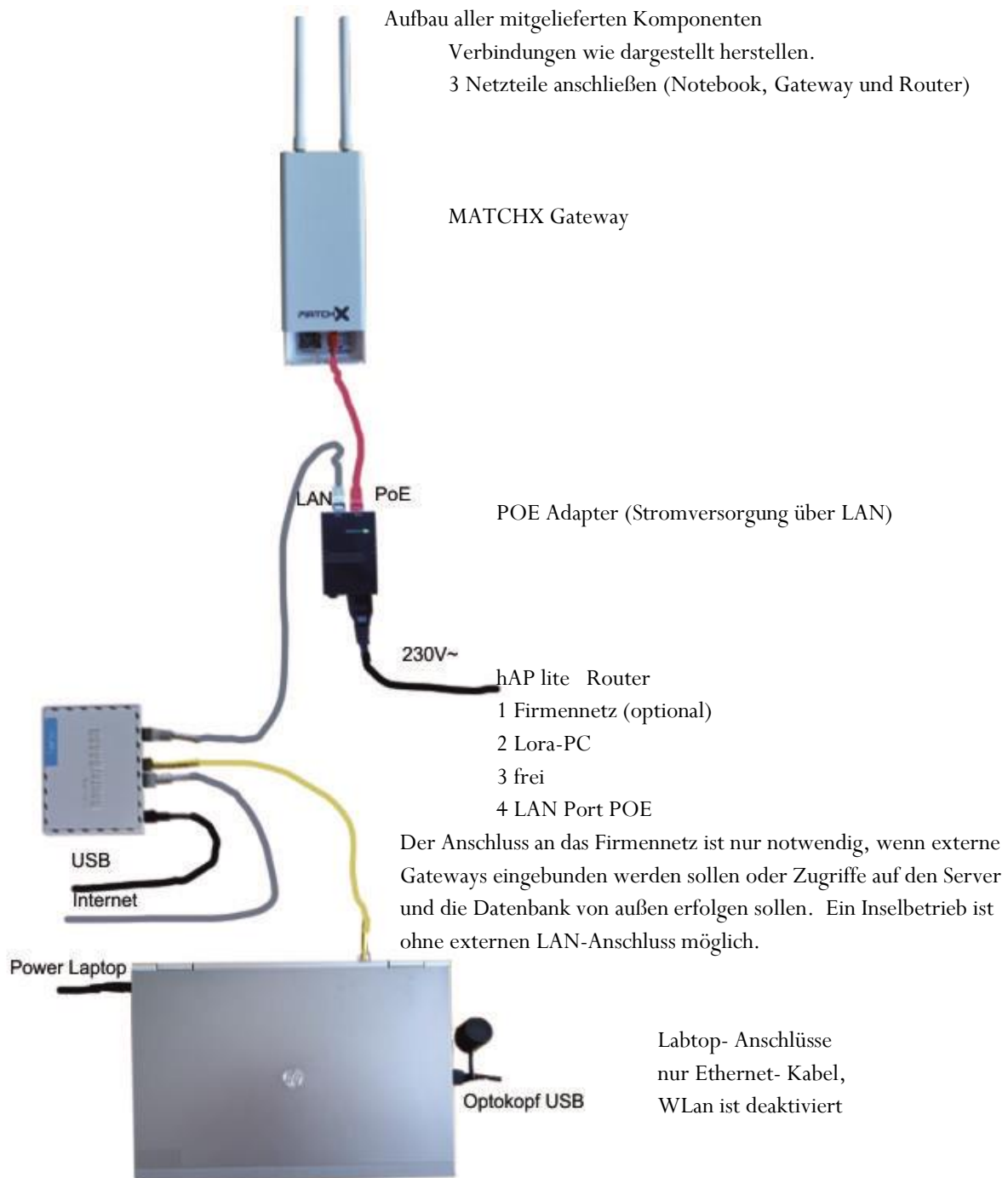
Alle zum Betreiben des Starterkits notwendigen Programme sind auf dem Notebook installiert. Es handelt sich um kostenfrei nutzbare Software (open Source). Die Software wird von Innotas ebenfalls kostenfrei bereitgestellt, Eine Garantie auf das Fehlen von Ansprüchen Dritter wird nicht gegeben. Insbesondere bei Änderungen der Lizenzbedingungen ist der Nutzer für die Lizenzierung der Software selbst verantwortlich.

Auf dem Notebook sind 2 lizenzierte Windows 10 Betriebssysteme installiert. Das zweite System ist für die virtuelle Maschine zuständig. Darin läuft der LoRa Server (Gotthardp LoRaWAN-Server) und der MQTT-Server (Broker). Durch die Kapselung der beiden Programme in der virtuellen Box ist eine Portierung der Server auf eine andere Maschine zu einem späteren Zeitpunkt einfacher möglich.

Auf dem Notebook-Host-System ist das Programmpaket XAMPP installiert. Davon wird die Datenbank MariaDB genutzt. Als Bindeglied zwischen den Servern und MariaDB fungiert das Programm „LoRa Applikations Tool“, kurz LAT von Innotas. Die Funktion des Programmes ist weiter unten beschrieben.

## 4.3 Inbetriebnahme des LoRaWAN-Starterkits

### 4.3.1 Installation der Komponenten und Schnellstart



Eine Anleitung kann auch unter youtube eingesehen werden: <https://youtu.be/MIBFAArJSNY>

- Notebook einschalten  
Anmeldung am Notebook-Windows (LoRa-Host System)  
Passwort „lora“ eingeben
- Virtueller Box startet automatisch. Wenn das Windows-Startfenster erscheint, melden Sie sich mit Passwort lora an, Gotthardp-Server und MQTT-Broker starten in der virtuellen Box.  
Nach Erscheinen der DOS-Fenster minimieren Sie die virtuelle Box und warten 2 Minuten, damit der Server komplett hochfährt.

- Danach Google Chrome auf der Host-Maschine starten (nicht Internet Explorer) und 192.168.88.251:8080 eingeben, falls kein Anmeldefenster erscheint, Vorgang wiederholen
- Im Anmeldefenster Passwort „admin“ eingeben  
Es erscheint das Dashboard vom Gotthardp-Server zur Kontrolle.
- LAT-Programm starten (LoRa Applikations Tool)
- links unten muss stehen: MQTT connect    DB connect  
falls nicht, links oben auf connect gehen

#### Metering-Geräte einpflegen über Programm LAT:

- Fenster Kommissionierung auswählen
- -1- DEVEUI-Datei auswählen: z.B. devlist1.csv  
wird mit den Geräten mitgeliefert, ist hier auf dem Desktop vorinstalliert
- -2- Profil auswählen (z.B. Wasserzähler)
- -3- Start betätigen, warten und Fertigmeldung quittieren
- Bei Bedarf den Vorgang mit einer anderen Liste wiederholen (z.B. für EHKV)
- Alternativ können die Geräte auch einzeln im Gotthardp unter Devices/commissioned angelegt werden

#### Metering-Geräte aktivieren:

- Optokopf an Notebook anschließen,
- Programm LST (LoRa Setup Tool) starten
- unter „Einstellungen“ passende COM auswählen
- Gerätefenster wählen, z.B. „Wasserzähler“
- Optokopf aufsetzen und „Aktivierung“ betätigen
- Die Geräte beginnen mit der Verbindung zum Gotthardp und können im Dashboard eingesehen werden
- Für alternative Methoden zur Aktivierung der Geräte z.B. über eine definierte Durchflussmenge, einen Aktivierungskopf oder über ein Android-App von Innotas lesen Sie bitte die entsprechenden Dokumente.

#### IP's (zur Information)

192.168.88.1	hAP lite Router
192.168.88.251	LORASERVER1 (Virtual Box)
129.168.88.254	LORAHOST1 (Laptop)
Über mac angebunden	MATCHX Gateway

#### User / Passwörter

	User	Passwd
<a href="https://192.168.88.251:8080">https://192.168.88.251:8080</a> DASHBOARD LoRa-Server	admin	admin
LORASERVER1 (Virtual Box)	Lora	lora
LORAHOST1 (Laptop)	Lora	lora
mySQL (auf Laptop)	alles	Beton#

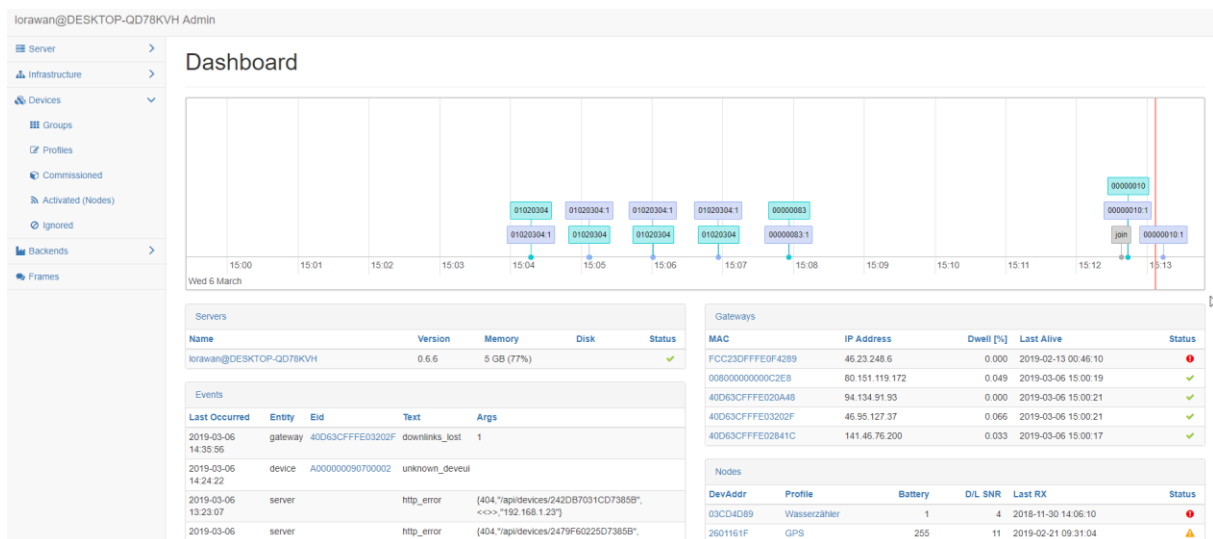
## 4.4 Betreiben des LoRaWAN-Servers Kit

Der LoRaWAN-Server GotthardP ist auf einem virtuellen Rechner in einer Virtual-Box installiert. Die Virtual-Box läuft wiederum auf dem Notebook Windows10 System. Wenn das Notebook gestartet wird startet auch die Virtual-Box automatisch. In der Virtual-Box starten dann der GotthardP Server und der MQTT-Broker. Der LoRa-Server verbindet sich mit dem Broker wie voreingestellt und sendet fortan alle empfangenen Daten an den MQTT-Broker. Auf dem Host-Rechner startet das xampp und fährt den MariaDB-Datenbank-Server hoch. Weiterhin wird das LAT-Tool-Programm automatisch gestartet. Dieses verbindet sich dann zum einen mit der Datenbank und zum anderen mit dem MQTT-Broker in der Virtual-Box. Das Gateway hat einen PacketForwarder welcher die empfangenen LoRa-Telegramme zum Server sendet. Ist der Server gestartet, so verbindet sich das Gateway allein mit dem Server.

### 4.4.1 Dashboard

Das Dashboard zeigt dem Betreiber eine Übersicht.

Im Browser kann das Dashboard des GotthardP Servers über die Eingabe <http://RechnerIP:8080> erfolgen. Es erfolgt eine Nutzerabfrage. Als Default ist der Nutzer „admin“ mit Password „admin“ eingerichtet. Es erscheint das Dashboard.





## 4.4.2 Menü / Navigation

☰ Server	▼
👤 Users	
🖥 Servers	
⚙ Configuration	
⚠ Events	
👥 Infrastructure	▼
📍 Areas	
📶 Gateways	
☁ Networks	
📢 Multicast Channels	
🔌 Devices	▼
🗃 Groups	
📄 Profiles	
📦 Commissioned	
📡 Activated (Nodes)	
🚫 Ignored	
🏠 Backends	▼
⚙ Handlers	
⚡ Connectors	
🗨 Frames	

-Nutzer

-Informationen zum Server

-ZusatzEinstellungen wie Google API / Anzeige usw.

-die letzten Meldungen vom System

-Bereiche (optional)

-Liste der Gateways

-Netzwerke, es können mehrere log.Netzw.exist.

-wird nicht genutzt

-Gruppierungen von Profilen

-Profile für Geräte mit Appl. und Netzwerkeinst.

-registrieren von Geräten vor der OTAA Anmeldung

-schon aktivierte Geräte

-Liste zu ignorierender Geräte

-Applikationen also Gerätetypen

-Daten-Verbinder zur Außenwelt

-empfangene Rohdaten

### 4.4.3 Gateways

Die Gateways werden nur durch ihre MAC beschrieben. Zusätzlich kann eine Beschreibung vermerkt werden. Hat das Gateway keinen GPS-Empfänger so kann händisch die Anbauposition vermerkt werden. Bei verbautem GPS-Empfänger wird die Position automatisch vom Gateway übermittelt. Die Location Darstellung funktioniert jedoch nur mit einem Google API Key. Der kann unter Server/Configuration eingetragen werden.

MAC	Area	Description	IP Address	Dwell [%]	Last Alive	Status
FCC23D...		LoriX Schlechteberg		0.000	2019-02-13 00:46:10	❌
008000...	Innotas internArea	erstet Test GW		0.124	2019-03-06 15:29:29	✅
40D63...	Innotas intern	MatchX Th.Korselt Str.19		0.000	2019-03-06 15:29:29	✅
40D63...	Innotas intern	Horst zu Hause		0.066	2019-03-06 15:29:27	✅
40D63...		MATCHX Hochschule		0.033	2019-03-06 15:29:24	✅

Der Lora-Server ist schon fertig eingerichtet. Es gibt ein logisches Netzwerk und die eingetragenen Gateways.

General Status

MAC \* FCC23DFFFE0F4289

Area Filter values

TX Chain \* 0

Antenna Gain (dBi) e.g. 6

Description LoriX Schlechteberg

Location \*

Altitude

Save changes

- Server >
- Infrastructure >
- Areas
- Gateways**
- Networks
- Multicast Channels
- Devices >
- Backends >
- Frames

## Edit gateway #40D63CFFFE03202F

General Status

### Alerts

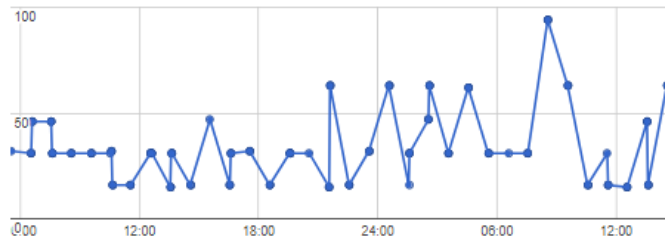
**IP Address** 46.95.127.37

**Last Alive** 2019-03-06 15:33:58

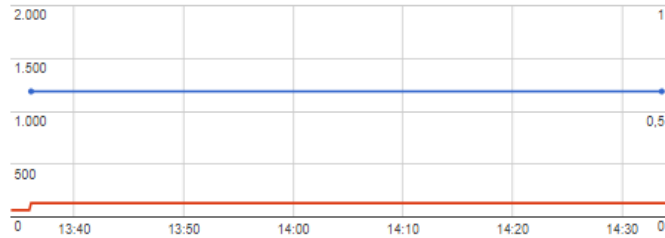
**Last GPS** 2019-03-06 15:33:28

**Last Report** 2019-03-06 15:33:50

### Network Delay



### Transmissions



Save changes

Für jedes Gateway kann der aktuelle Status eingesehen werden. Hier wird unter anderem die Signallaufzeit angezeigt. Das ist bei Gateways, die über LTE/UMTS angebunden sind, sehr wichtig.

## 4.4.4 Profile

Die verschiedenen Gerätetypen die mit dem Server empfangen werden sollen, teilt man in verschiedene Applicationen ein, weil die Nutzdaten in der Regel unterschiedlich kodiert sind. Im Gotthard-Server wird organisatorisch aber noch ein Profil darüber gesetzt. Hier können für die jeweiligen Profile dann noch andere spezifischen Einstellungen vorgenommen werden.

Server >	Profiles List				
Infrastructure >	<input type="checkbox"/>	<b>Name</b>	<b>Group</b>	<b>Application</b>	<b>App Identifier</b>
Devices v	<input type="checkbox"/>	EHKV	Innotas intern	App2	InnotasEHKVs
Groups	<input type="checkbox"/>	Sowag	Innotas Eilbau	App1	Feldtest SOWAG
Profiles	<input type="checkbox"/>	Wasserzähler	Innotas intern	App1	InnotasWZ
Commissioned	<input type="checkbox"/>	GPS	Innotas intern	App3	GPS-Testsender
Activated (Nodes)					

### Edit profile #EHKV

General ADR

Name \* EHKV

Group \* Innotas intern

Application \* App2

App Identifier InnotasEHKVs

Join Filter values

FCnt Check Disabled

TX Window Auto

Save changes

Hier erfolgt die Zuordnung von Applikationen (Gerätetypen) mit Gruppen und einer Beschriftung.

### Edit profile #EHKV

General ADR

ADR Mode Auto-Adjust

Set Power Filter values

Set Data Rate Filter values

Max Data Rate SF7 125 kHz (5470 bit/s)

Set Channels e.g. 0-2

Set Duty Cycle e.g. 100%

Set RX1 DR Offset e.g. 0

Set RX2 DR SF9 125 kHz (1760 bit/s)

Set RX2 Freq (MHz) 869,525

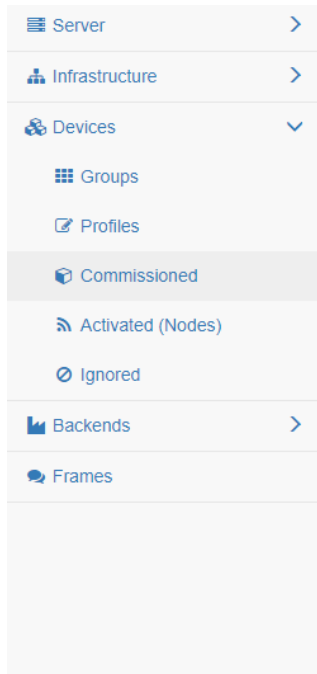
Request Status? true

Save changes

Hier können für ein jedes Profil LoRa spezifische Einstellungen vorgenommen werden, wenn sie von den Normeinstellungen des Netzwerkes abweichen.

## 4.4.5 Device

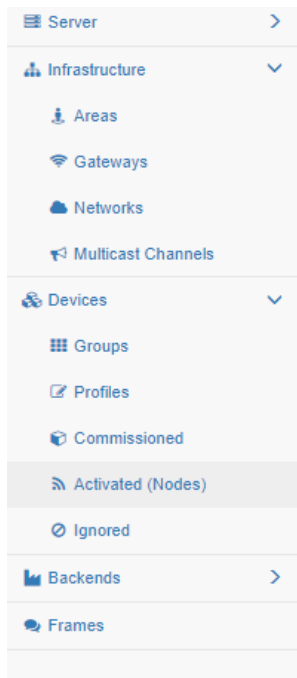
Die einzelnen LoRa-Geräte müssen vor ihrer Anbindung an dem Server dem Server bekannt gemacht werden. Geräte die sich über das OTAA Verfahren „join“ (verbinden) dürfen werden „kommissioniert“. Dazu werden die öffentlichen Schlüssel DEVEUI, APPEUI, und APPKEY eingetragen. Zusätzlich ist die Zuordnung zu welchem Profil das Gerät gehört notwendig. Im Profil wird unter anderem der Gerätetype festgelegt, also die Zuordnung, wie die Nutzdaten ausgewertet werden sollen.



### Create new device

Form fields for creating a new device:

- DevEUI \*: e.g. 0123456789ABCDEF
- Profile \*: Filter values
- AppArguments
- AppEUI: e.g. 0123456789ABCDEF
- AppKey \*: e.g. FEDCBA9876543210FEDCBA9876543210
- Description
- Last Joins
- Node: e.g. ABC12345
- Submit button



### Edit node #00000010

General ADR Status

Alerts

Status Time: 2019-03-06 15:13:13

Status FCnt: 1

Device Status

Save changes

Wenn ein Gerät im Server angelegt wurde und das Gerät sich über OTAA mit dem Join-Prozess verbunden hat wird es im Server als aktiviert angezeigt. Bei Join-Prozess teilt der Server dem Gerät private Schlüssel zu und gibt dem Gerät eine serverinterne Nummer, die DevAddr. Im Statusfenster können nun nach einiger Betriebszeit einige statistische Werte wie Empfangsparameter und der vom Gerät gesendete Batteriezustand abgelesen werden.

## 4.4.6 Handler

Die Handler sind die logischen Datenschnittstellen. Hier werden die einzelnen Gerätetypen und die einzelnen Daten definiert, die über eine physische Schnittstelle nach Außen gegeben werden sollen. Es besteht aber auch die Möglichkeit schon auf spezielle Empfangsdaten zu reagieren.

Application	Uplink Fields	Payload	D/L Expires
App2	deveui, devaddr, app, battery, port, datetime, rssi, lsnr, data, datr, fcnt, freq, all_gw, mac	custom	never
App1	deveui, devaddr, app, battery, port, datetime, rssi, lsnr, data, datr, fcnt, freq, all_gw, mac	custom	never
App3	deveui, devaddr, app, battery, port, datetime, rssi, lsnr, data, datr, fcnt, freq, all_gw, mac	custom	never

Name	Format	URI	Enabl
MQTT-Client3	JSON	mqtt://localhost:1883	✓

## 4.4.7 Connectors

Die Connectoren sind die physischen Schnittstellen nach außen. Wir nutzen hier nur die MQTT-Schnittstelle.

**Connectors List** Export Create

Name	Application	URI	Publish Uplinks	Received Topic	Enabled	Status
MQTT-Client	App1	mqtt://localhost:1883	out/{devaddr}	in/{devaddr}	✓	✓
MQTT-Client2	App2	mqtt://localhost:1883	out2/{devaddr}	in2/{devaddr}	✓	✓
MQTT-Client3	App3	mqtt://localhost:1883	out3/{devaddr}	in3/{devaddr}	✓	✓

1 - 3 of 3

### Edit connector #MQTT-Client

General Authentication Status

Connector Name \*

Application

Format \*

URI \*

Publish QoS

Publish Uplinks

Publish Events

Subscribe QoS

Subscribe

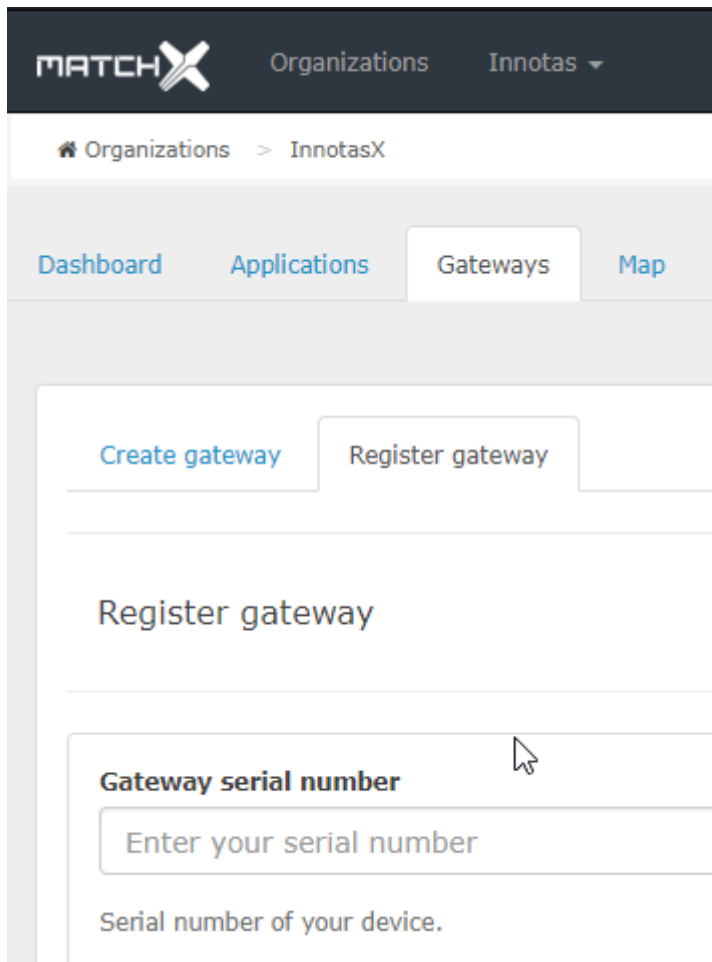
Received Topic

Enabled \*

Failed

## 5 Anbindung von externen Gateways

In der Grundversion des Kit ist ein Gateway installiert und schon mit dem LoRa-Server verbunden. Das Gateway der Firma MatchX ist ein Multikanal Outdoor Gateway. Es besitzt ein GPS-Empfänger und sendet selbständig seine Position an den LoRa Server. Zum Inbetriebnehmen eines weiteren LoRa-Gateway von MachX gehen sie auf die Webseite der Firma: <http://matchx.io>. Dort rechts oben auf den Menüpunkt „Cloud“ . scrollen Sie nach unten und klicken Sie auf die Gebiets-Flagge also die EU. Sie gelangen auf ein Login. Zum Anlegen eines neuen Gateways gehen Sie auf Gateways und rechts auf „creategateway“. Dort gehen Sie auf die Registerkarte „Register gateway“ und tragen ihre Seriennummer des Matchx Gateways ein. Das mitgelieferte Gateway haben wir schon angelegt.



Wenn ihr Gateway in Betrieb ist und Internetverbindung hat so wird es „nach Hause“ telefonieren also sich beim Matchx-Server melden und sie sollten unter „Gateways details“ Einzelheiten zu ihrem Gateway sehen. Das Gateway sendet laut Werkseinstellung die empfangenen LoRa Daten alle zum MatchX-Server. Sie können dort auch Geräte anlegen und weiter verarbeiten, jedoch sind die Möglichkeiten begrenzt und die Daten liegen irgendwo in der Cloud. Wir müssen nun dem Gateway mitteilen, wohin es die Daten in Zukunft senden soll. Dazu scrollen Sie in „Gateway details“ nach unten und klicken auf „Edit Config“. Sie sehen ungefähr folgendes.



```

{
  "SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1, /* radio_1 provides clock to concentrator */
    "lbt_cfg": {
      "enable": true,
    }
  }
  ....
  ....
  ....

  "gateway_conf": {
    /* change with default server address / ports, or overwrite in local_conf.json */
    /* "server_address": "eux.matchx.io", */
    "server_address": "innotas.selfhost.eu",
    /* "serv_port_up": 1700, */
    /* "serv_port_down": 1700, */
    "serv_port_up": 1680,
    "serv_port_down": 1680,
    /* adjust the following parameters for your network */
    "keepalive_interval": 10,
    "stat_interval": 30,
    "push_timeout_ms": 100,
    /* forward only valid packets */
    "forward_crc_valid": true,
    "forward_crc_error": false,
    "forward_crc_disabled": false,
    /* GPS configuration */
    "gps_tty_path": "/dev/ttyS1",
    /* GPS reference coordinates */
    "ref_latitude": 0.0,
    "ref_longitude": 0.0,
    "ref_altitude": 0
  }
}

```

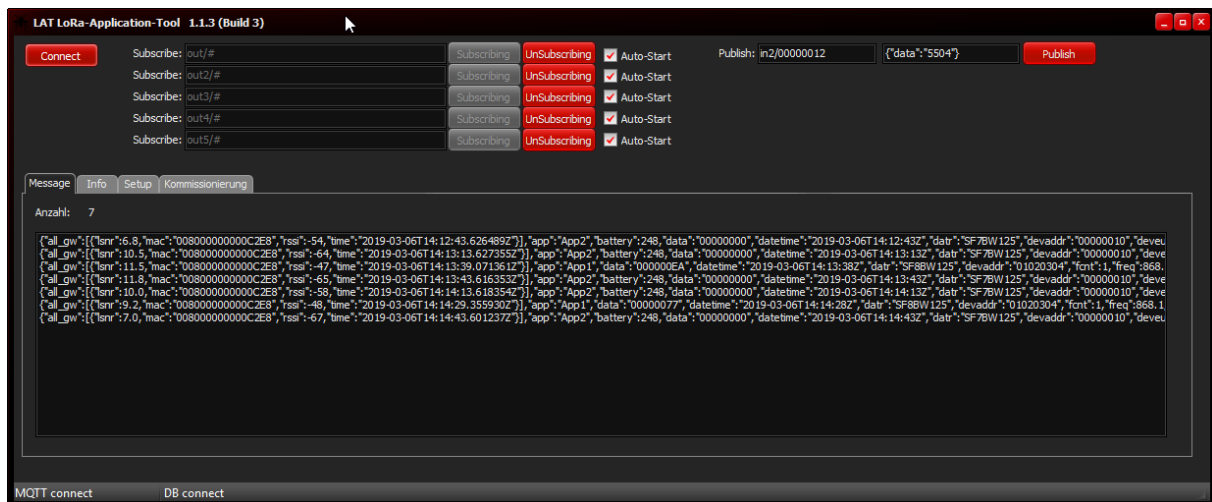
Bitte ändern Sie die farbig markierten Stellen. Für „innotas.selfhost.eu“ tragen Sie ihre Domain ein. Ihr Netzwerk muss dann so eingerichtet werden das Anfragen an Port 1680 auf ihren LoRa-Server weitergeleitet werden.

## 6 Weitere Links und Quellen

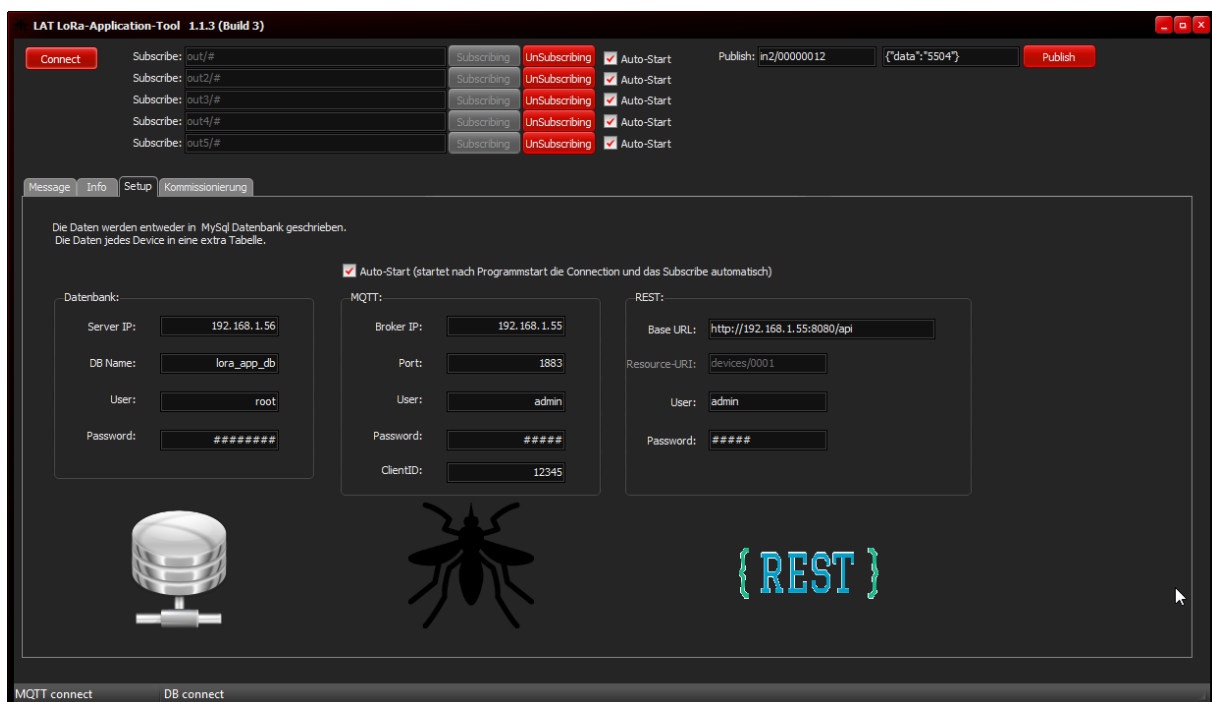
Für weitere Informationen zum GotthardP-LoRa-Server schauen Sie hier:

<https://github.com/gotthardp/lorawan-server/tree/master/doc>

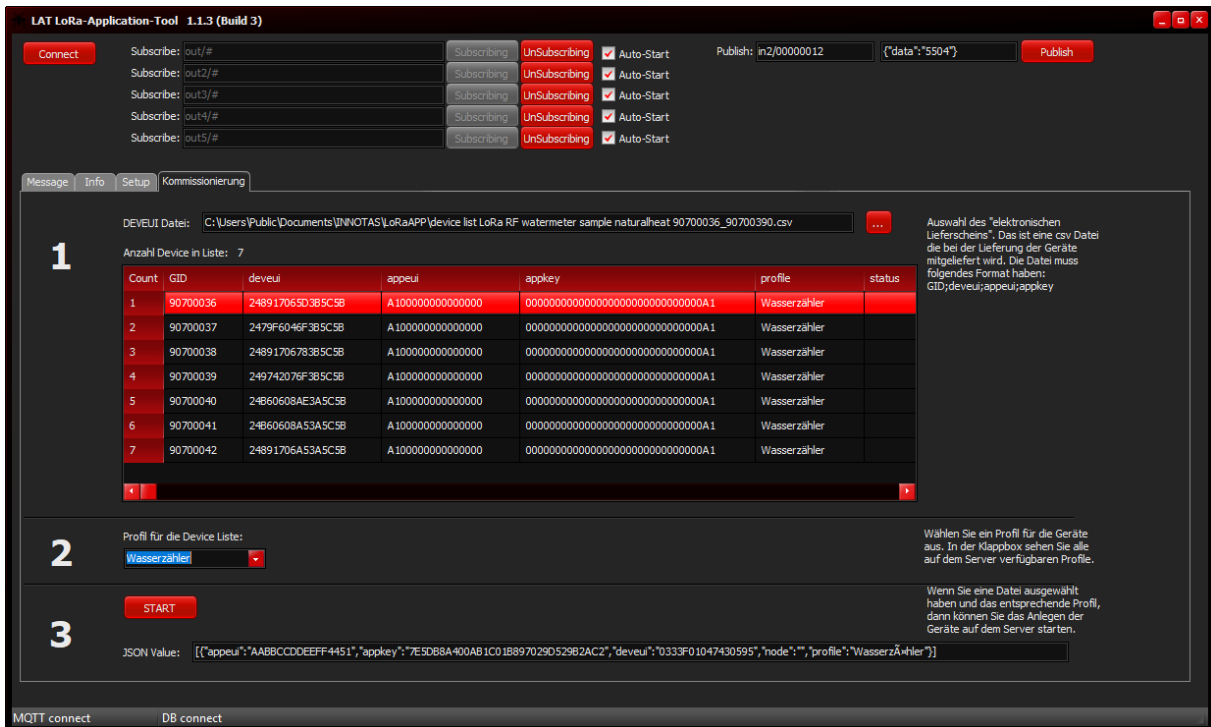
## 7 LoRaApplications Tool



Der LoRa-Server hat 3 Schnittstellen für Ein- und Ausgaben. Zum einen HTML für das WEB-Interface, MQTT um Daten von den Geräten und zu den Geräten zu senden und noch eine REST Schnittstelle zum Abfragen von Listen und das Anlegen von Geräten. Der LoRa-Server empfängt die LoRa-Protokolle und entschlüsselt sie. Was er nicht kann ist die Interpretation der Nutzdaten in einzelne Messwerte oder Verbrauchswerte (Payload). Der Payload wird herstellenspezifisch kodiert und ist bei jedem Gerätetyp unterschiedlich. Die Aufgabe der Dateninterpretation und Speicherung der einzelnen Daten in eine Datenbank übernimmt das LoRa-Application-Tool. Weiterhin besteht die Möglichkeit über die REST-Kommunikation mit dem Server eine automatische Kommissionierung der zu verbauenden Geräte auf dem Server vorzunehmen.



Die einzelnen Schnittstellen wie Datenbank, Mosquitto oder REST benötigen Verbindungsparameter. Beim DB-Server ist das die Server-IP, der Datenbank-Name, der Datenbank-Nutzer mit seinem Passwort. Es wird hier immer Port 3306 verwendet. Die Datenbank ist vorher auf dem Server anzulegen. Alle Tabellen werden automatisch erstellt. Die Schnittstellen MQTT und REST benötigen auch die Zugangspunkte und die Nutzerdaten. Wenn mehrere Instanzen des LAT Programms genutzt werden so ist wichtig zu beachten, das sich die ClientID im MQTT unterscheiden! Das wurde in dem Fall durch uns alles schon erledigt.



Mit dem Programm können auch an Hand des elektronischen Lieferscheines (csv Datei) die Geräte automatisch auf dem Server angelegt werden (Kommissionierung). Es ist als nicht notwendig jedes Gerät einzeln über das WEB-Interface auf dem LoRa-Server anzulegen. Weiterhin werden diese Geräte in die externe Datenbank übernommen. Somit ist es möglich eine Verbindung von der Geräte-ID (welche als einzige auf dem Gerät steht) und den LoRa spezifischen Nummern herzustellen. Somit kann recht einfach über SQL-Abfragen ein Bezug hergestellt werden zur Geräte-ID und den gewünschten Daten des jeweiligen Gerätes. Auf der Basis der Datenbank können in externen Anwendungen Verbindungen zu Daten der Immobilienwirtschaft zum Zwecke einer Abrechnungserstellung erstellt werden.

Geräte, welche zusammen mit dem LoRa-Kit erworben wurden, werden von uns schon kommissioniert.

## 7.1 Funktion und automatischer Betrieb

Das Programm kann für einen automatischen Start eingestellt werden. Wenn das Programm durch einen Autostart-Mechanismus gestartet wird so kann das Programm nach seinem Start automatisch versuchen den Kontakt zum MQTT-Broker und zum DB-Server herzustellen. Ist dieses erfolgt kann das Programm automatisch die einzelnen Topic's auf dem MQTT-Broker abonnieren. Das bedeutet das die Daten der verschiedenen Applikationen die vom LoRa-Server zum Broker gesendet werden bei Broker „bestellt“ werden.

