Stats4Lox::lox2telegraf

my ($status, @queue) = lox2telegraf(\@data, $nosend);

Creates an array with InfluxDB line format protocol lines of all submitted values. Data can be send to telegraf directly.
The **first** parameter is the **Data** as an Array of Hashes (see below). This is mandatory.

The **second** parameter can be set to 1 if you do not want to send the data to telegraf and just return the prepared array. If set to 0, data will be send to telegraf (Default). This is optional.

Parameter

Parameter	Required	Beschreibung
\@data		Array of Hashes with data (see below)
$nosend		Toogle to suppress sending data to telegraf (1). Default is 0 and sending data to telegraf.

Return value

Parameter	Description
1. return	Status
2. retutrn	Array with text in Influx line protocol

The **first** return value is the status of the function:

- (0) Data was prepared and send
- (1) Data was prepared but not send
- (2) an error occurred.

The **second** return value is an array with the text in Influx line protocol. It can be send to influx using the HTTP API or the command line interface:
https://docs.influxdata.com/influxdb/v1.8/write_protocols/line_protocol_tutorial/#getting-data-in-the-database

A function call will looks like this:

```
my ( $status, @queue ) = lox2telegraf( \@data, 0 );
```

The response looks like this (as an array):

```
my_measurement,tag1=This\ is\ tag\ 1,tag2=This\ is\ tag\ 2 field1="A\
String",field2=20i,field3=10.2 1616935446000000
```

Data structure

Datastructure @data must be as follows:

```
$VAR1 = {
          'timestamp' => '1467360000000000000',
          'msno' => '1',
          'uuid' => '0ad03c43-0102-d583-ffffb70a5529d684',
          'room' => 'UG Hauswirtschaft',
          'type' => 'StateV',
          'category' => "L\x{c3}\x{bc}ftung",
          'source' => 'import',
          'values' => [
                        {
                          'key' => 'Default',
                          'value' => '24.184'
                        }
                      ],
          'description' => "Temperatur Au\x{c3}\x{9f}enluft",
          'measurementname' => "Temperatur Au\x{c3}\x{9f}enluft",
          'name' => "Au\x{c3}\x{9f}entemperatur"
        };
$VAR2 = {
          'name' => "Au\x{c3}\x{9f}entemperatur",
          'description' => "Temperatur Au\x{c3}\x{9f}enluft",
          'values' => [
                        {
                          'key' => 'Default',
                          'value' => '23.450'
                        }
                      ],
          'source' => 'import',
          'measurementname' => "Temperatur Au\x{c3}\x{9f}enluft",
          'room' => 'UG Hauswirtschaft',
          'category' => "L\x{c3}\x{bc}ftung",
          'type' => 'StateV',
          'uuid' => '0ad03c43-0102-d583-ffffb70a5529d684',
          'msno' => '1',
          'timestamp' => '1467361800000000000'
        };
```

Usage

```
#!/usr/bin/perl

use LoxBerry::System;
require "../Stats4Lox.pm";
```

```
# Debug
$Stats4Lox::DEBUG = 1;
$Stats4Lox::DUMP = 1;

my @data;
my %influxrecord = (
                timestamp => '',
                msno => '1',
                uuid => '11111-11111-11111-1111',
                name => 'Das ist mein Name',
                category => 'Heizung',
                room => 'Keller',
                type => 'Output',
        );

my @values;
push @values, { key => 'output1', value => 1 };
push @values, { key => 'output2', value => 2 };
push @values, { key => 'output3', value => 3 };
push @values, { key => 'output4', value => 4 };
push @values, { key => 'output5', value => 5 };

$influxrecord{values} = \@values;

push @data, \%influxrecord;

my $nosend = "0";
my (@response) = Stats4Lox::lox2telegraf( \@data, $nosend );
```