YOSENSI.IO

YO Analog

User guide v1.0

Release notes

Released	Version	Key changes
11.06.2021	1.0	Initial release.

Content

Product description

Overview

YO Analog is used for measuring analogue signals. The device has six configurable measurement inputs, each of which can be used in one of two modes: voltage input (0–10 V) or current input (4–20 mA).

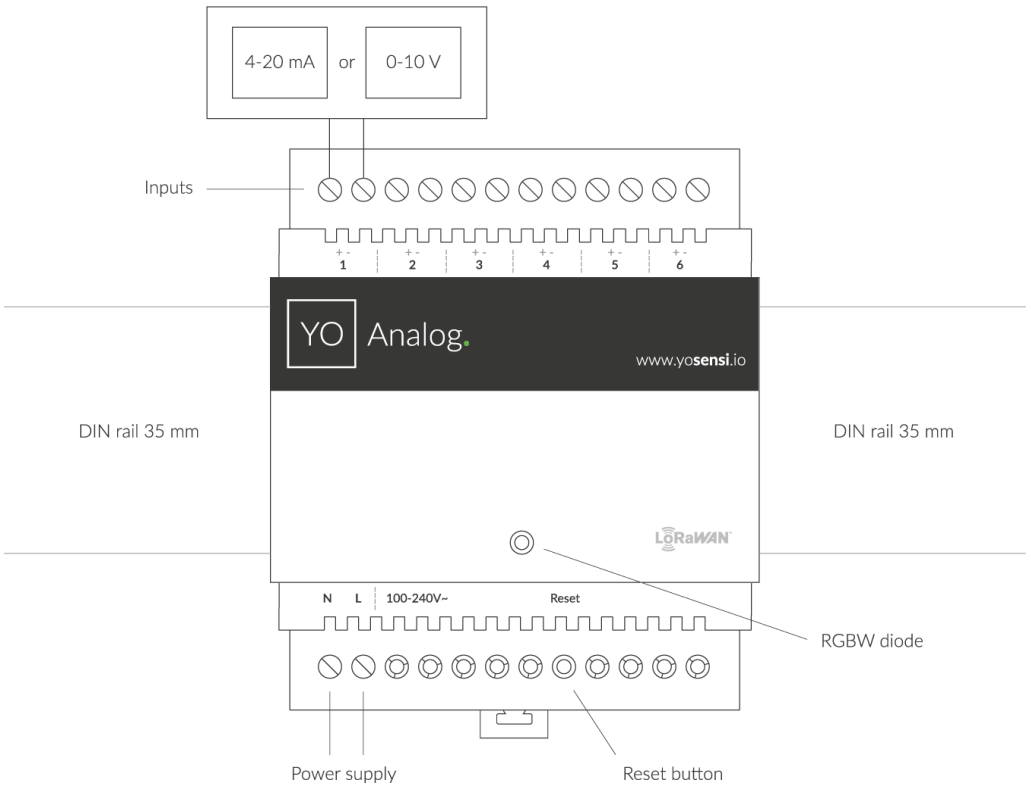

Figure 1 Device top view.

Visible device elements:

- analog inputs (voltage or current),
- power supply connectors,
- reset button,
- RGBW diode.

Applications

Based on the data collected by YO Analog, it is possible to monitor measurement values of devices and processes in automation.

IoT system components

Typical IoT systems consist of 3 main elements (Figure 2), brief described below. In order to set communication, each element must be properly configured.

1. Node – device with sensors and a wireless communication module that gathers data, forms the payload and sends it to the gateway.
2. Gateway – device similar to routers, equipped with a LoRa concentrator, that receives LoRa packets and push them to the Internet-connected server.
3. Server – in most cases, a cloud-based service where data are processed, stored, analysed, and presented in user-friendly ways (via a user interface); Yosensi default and recommended tools are YO Suite (for IoT structure management) and Grafana (for data presentation).

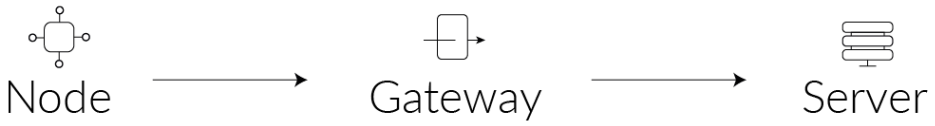

Figure 2 IoT system components.

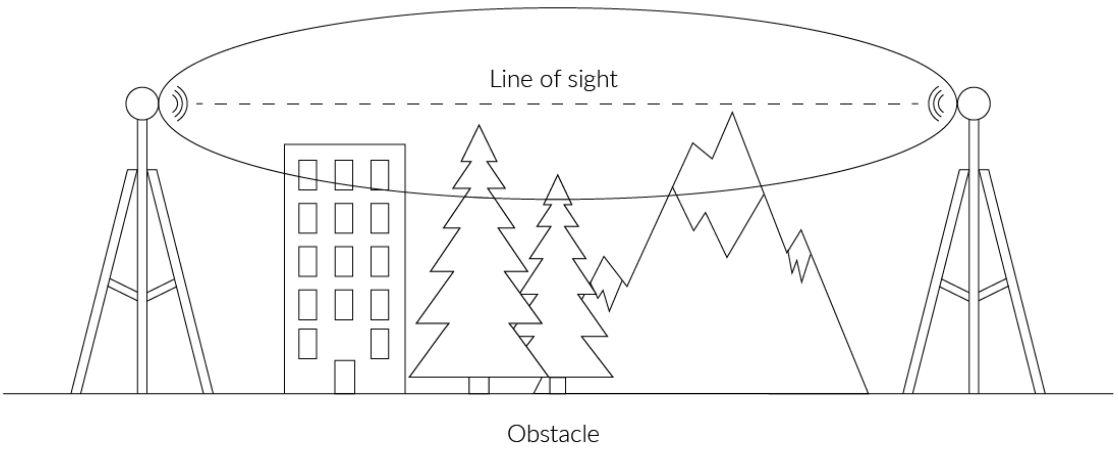

Figure 3 Fresnel zone where communication between two antennas can occur.

Device configuration

Configurable parameters

A few parameters must be set in order to send data to the gateway. The default firmware is configured in ABP connection type with predefined *deveui*, *appkeyota*, *keyotaa*, *nwkskey* by MCU.

CONFIGURABLE PARAMETERS

NAME	DESCRIPTION	POSSIBLE VALUES	DEFAULT VALUE	READ/ WRITE
devname	Device name	LNAN	LNAN	R
conntype	Connection type	0 – OTAA 1 – ABP	1	R/W
deveui	Device address EUI	8 B (HEX)	predefined	R/W
appkeyota	OTAA application EUI	8 B (HEX)	predefined	R/W
keyotaa	OTAA key	16 B (HEX)	predefined	R/W
addrabp	ABP device address	4 B (HEX)	predefined	R/W
nwkskey	Network Session Key	16 B (HEX)	predefined	R/W
appskey	Application Session Key	16 B (HEX)	predefined	R/W
advble	Interval advertising Bluetooth [s]	MS_INPUT[1]	1600	R/W
measinter	Measuring and/or sending interval LoRa [s]	1–999999	300	R/W
measlevel	Measuring threshold [%][2]	1–9999	20	R/W
input1	Analog input type 1	1 – 4–20 [mA] 2 – 0–10 [V]	1	R/W
input2	Analog input type 2		1	R/W
input3	Analog input type 3		1	R/W
input4	Analog input type 4		1	R/W
input5	Analog input type 5		1	R/W
input6	Analog input type 6		1	R/W

[1] Calculation formula: MS_INPUT = INTERVAL_MS × 1.6.
[2] Sets the threshold that is used to save the peak timestamp and value that is between regular.

Parameters description:

1. Connection type (*conntype*)

   Used for setting the device in ABP – value "1" or OTAA – value "0" mode.

2. Predefined addresses and keys (*deveui*, *appkeyota*, *keyotaa*, *addrabp*, *nwkskey*, *appskey*)

   These parameters are generated using multiple IDs specific to the particular MCU and are unique for each device. They can be changed if needed.

3. Interval advertising Bluetooth (*advble*)

   Determines the interval of sending broadcast packets, used to connect to every BLE receiver around the device.

4. Measuring threshold (*measlevel*)

   YO Power is checking for changes in signal **every 0,5 s**. If consecutive measurement readings differ by more than specified by this parameter (see equation below) the present value and timestamp will be saved and prepared for transmission to the server (see figure 4).

$$|a - b| > b \cdot th$$

where:
a – current measurement
b – immediately preceding measurement
th – threshold value [%]

5. Measuring and/or sending interval in seconds (*measinter*)

   Interval of sending each LoRa payload (t_{li} on figure 4). It can be dynamically changed due to the amount of peaks in buffer (threshold exceed) in order to keep server updated with the latest data. For example if payload can hold 4 points and we have 6 points to send the algorithm will send these 4 points first and reduce the time interval of future payloads (see figure 4).

6. Analog inputs (*input1*, …, *input6*)

   You can set the device inputs to work in 2 modes:
   - value "1" – 4–20 mA current standard
   - value "2" – 0–10 V voltage standard

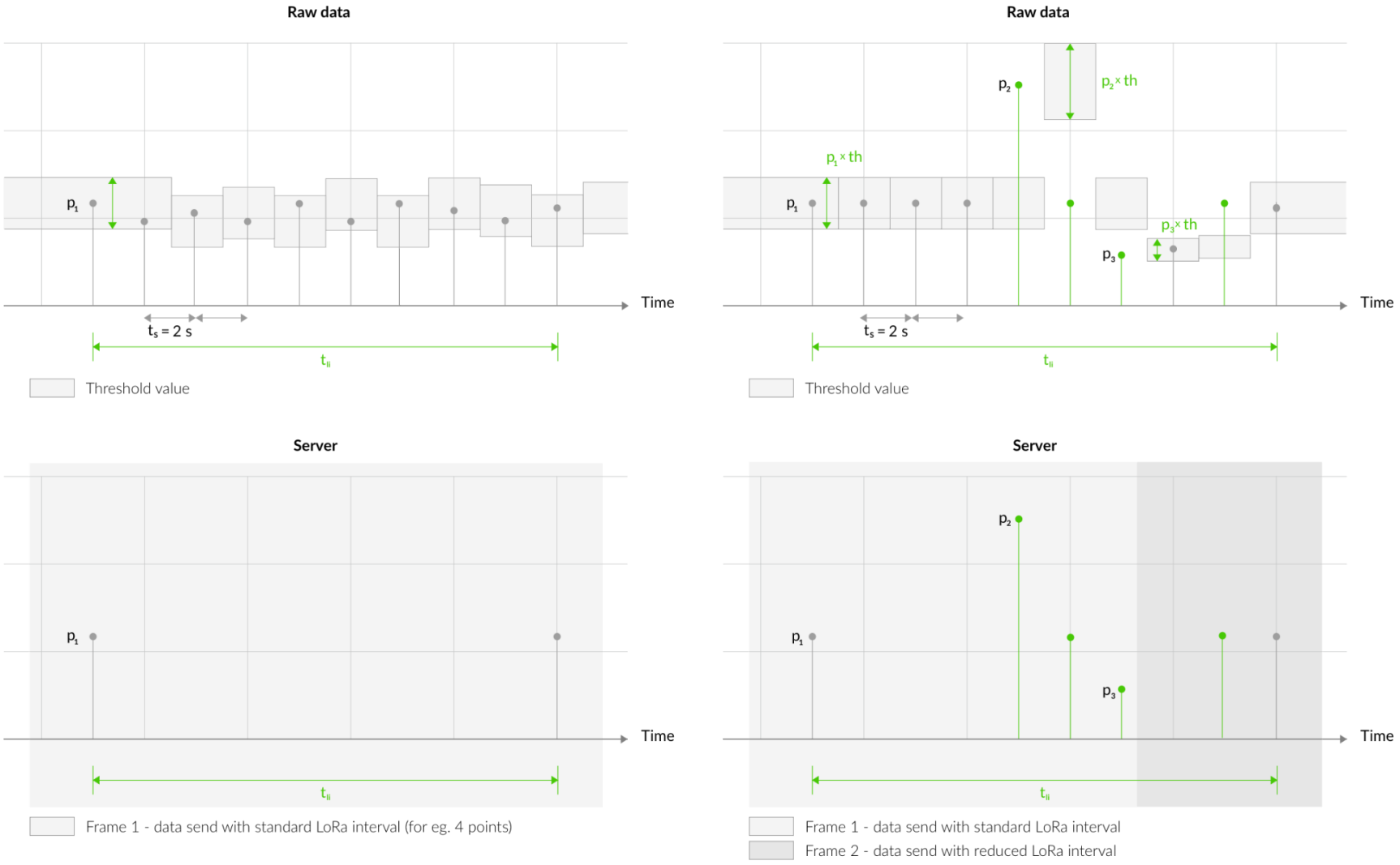

Figure 4 Threshold algorithm visualisation.

***t_{li}** - measinter, **t_s** - time between consecutive measurements, **p_1 ... p_3** - measurements values, **th** - measlevel [%]*

Quick start guide

Required equipment/software:

- PC with Bluetooth adapter (USB dongle or built-in),
- power supply 100–240 V AC,
- Yosensi CLI tool.

NOTE **Yosensi CLI tool works only with YO Analog default Yosensi firmware. For official firmware updates, contact us on support@yosensi.io.**

Node configuration

Connect to the device following these instructions:

1. Read the macBLE address from the sticker on the device. Alternatively, you can identify the device by RSSI parameter (which correlates with distance between transmitter and receiver).

Figure 5 Device information sticker/nameplate example.

2. Make sure you have your Bluetooth LE adapter turned on and working properly.
3. Download and run the CLI tool in the terminal/console application.
4. Connect power supply pins (input - N, L) of the YO Analog device. See Quick Installation Guide for step by step instructions.
5. Run `yosensi-cli-tool_vX.X.X_WIN.exe list` to scan for Bluetooth devices. You can see all commands by typing `yosensi-cli-tool_vX.X.X_WIN.exe --help` or add `-h` to your current command to see all needed parameters.
6. If you find your MAC address in the `list` command results, you can connect and reconfigure the device by using one of the available commands. You should use *change_dev_params* subcommand to change device settings. Additional information, including the commands supported by the CLI, are available at https://yosensi.io/support.

Command help view:

```
.\yosensi-cli-tool_vX.X.X_WIN.exe change_dev_params --help
usage: yosensi-cli-tool change_dev_params [-h] mac dev_param [dev_param ...]

positional arguments:
  mac         Bluetooth public MAC address
  dev_param   device parameter(s) name and value, in format 'name=value'
```

Example of use:

```
.\yosensi-cli-tool_vX.X.X_WIN.exe change_dev_param 80:e1:26:1d:2a:33 conntype=0 measinter=1000
It will take up to 180s to find and write new device param(s) into memory, please wait ...
Searching for a device ...
Trying to connect to the device ...
Updated 'conntype' device parameter: SUCCESS
Updated 'measinter' device parameter: SUCCESS
```

After node reconfiguration, you need to have access to the gateway and server. Remember that LoRaWAN network can be set in either of **public** or **private** (default) mode. If you have your gateway prepared and configured, you can start to customize the server side. We'll go through an example in our recommended YO Suite software.

NOTE **The firmware update process is described in CLI tool manual page 13. Visit https://yosensi.io/support to see all available documentation.**

YO Suite configuration

Before you can make the node visible, you'll need an **organization** and an **application**. The organization is your own space, at the highest level of IoT systems management (like the root directory in operating systems). It can be created only by Yosensi staff, and all clients using YO Suite have it created for them by default (in case of any problems you can find us on support@yosensi.io). The application is a representation of each system and, together with nodes definitions, are created by customers. Basic integration of a Node and YO Suite is described below.

NOTE **A subscription is needed to use YO Suite. Visit https://yosensi.io/ for more information and pricing.**

YO Suite integration instructions:

1. Go to app.yosensi.io and log in.
2. You'll see the default organization view. To switch to another organization, click on the user avatar in the right top corner and select 'Switch Organization'.
3. To create an application, press the bottom right '+' button. Fill in the 'Name' and 'Description' fields and 'Select Application Profile' which is the region definition.

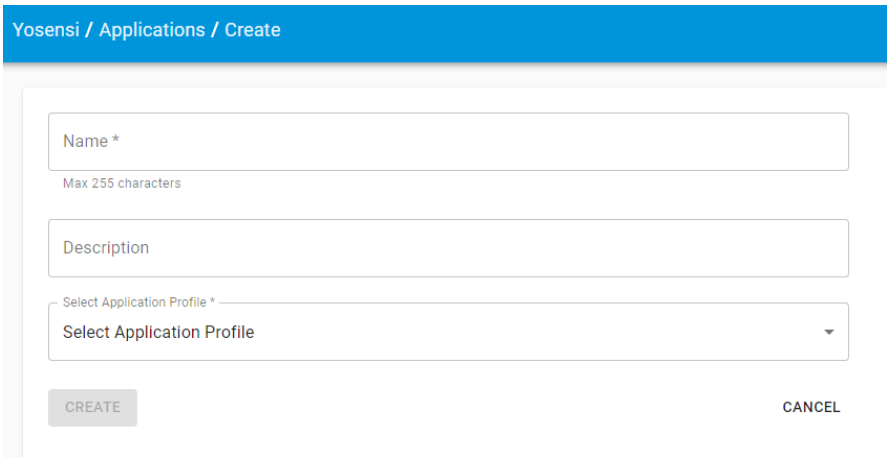

Figure 6 Application creation form.

4. Proceed to the application by clicking its name on the list, and press the '+' button to add a node. Select 'Lora' in 'Node Type' field. Set the 'Name', 'Description' fields and fill in 'Node ID' (*deveui*) and 'OTAA Key' (*keyotaa*) **which can be found using the CLI tool (*list_dev_params*).** Select a model that is compatible with your device—it affects the number of charts, its placement and data source (YO Analog model). You can also set the node's 'Location', if locations have been pre-defined. If you haven't defined a suitable location, leave this field set at <None>.

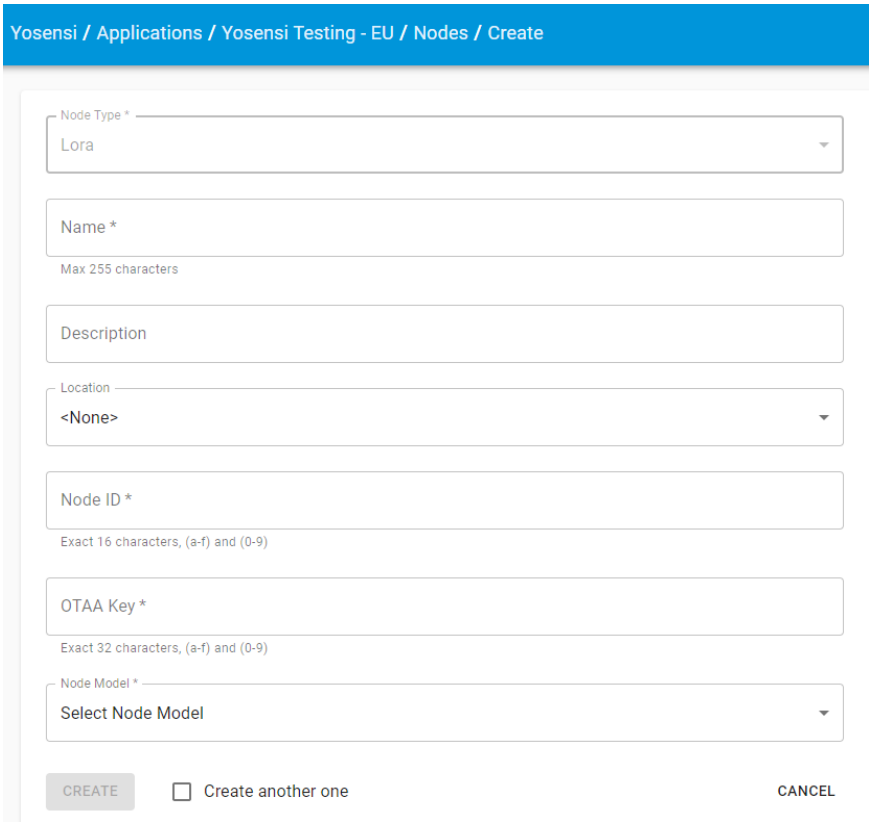

Figure 7 Node creation form.

5. **Every new node must be added in OTAA mode**. You must then switch its type to ABP after activation in YO Suite by changing the Node configuration. Click on the link in the 'Node Name' column. Go to the 'KEYS' tab and switch 'Lora Type' from OTAA to ABP, and fill in the 'Device Address' (*addrabp*), 'Application Session Key' (*appskey*) and 'Network Session Key' (*nwkskey*) with **values from the CLI tool,** and press UPDATE.

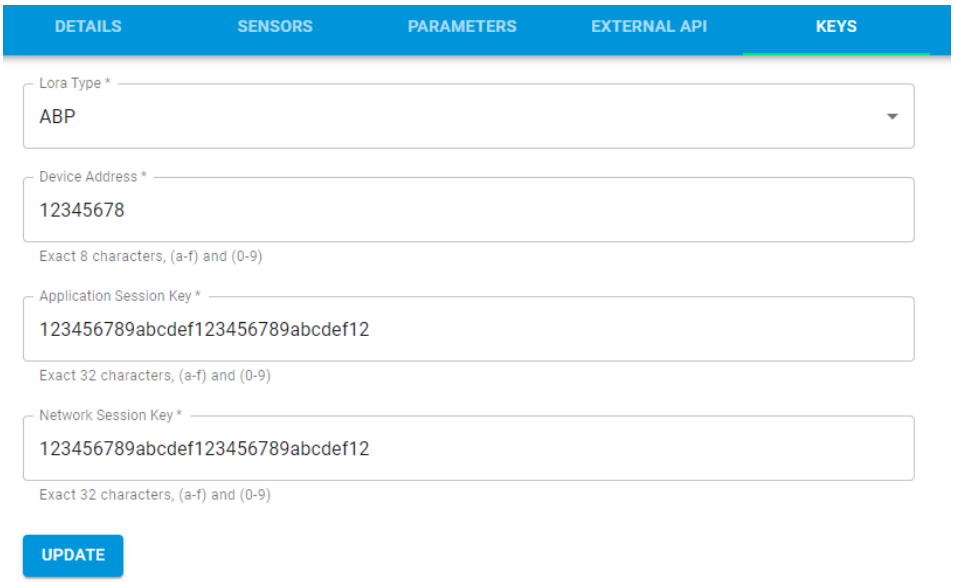

Figure 8 Node LoRa type configuration form.

6. When data are received by the server, you'll notice that the 'Last Seen' column ('NODES LIST' tab) status changes from 'never' to 'a few seconds ago'.
7. Open charts by clicking on the 'OPEN' button in Dashboard columns or by entering the node's 'DETAILS' tab ('Node Name' column link) and clicking 'CHARTS'.

Payload description

If you want to connect to your own server you need the payload decoder which can be found at https://yosensi.io/support.

Example of YO Analog payload with description:

02:00:00:09:2e:00:11:01:00:00:2e:00:11:02:01:90:2e:00:11:03:04:1a:32:00:11:04:00:00:32:00:11:05:01:e6:32:00:11:06:00:f1

Payload header				First measurement					
0x02	0x00	0x00	0x09	0x2E	0x00	0x11	0x01	0x00	0x00
ver = 2	cnt = 0	pct [s] = 9		type = 11, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 1	val = 0 (0,0 [mA])	

Second measurement					
0x2E	0x00	0x11	0x02	0x01	0x90
type = 11, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 2	val = 400 (4,0 [mA])	

Third measurement					
0x2E	0x00	0x11	0x03	0x04	0x1A
type = 11, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 3	val = 1050 (10,5 [mA])	

Fourth measurement					
0x32	0x00	0x11	0x04	0x00	0x00
type = 12, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 4	val = 0 (0,0 [V])	

Fifth measurement					
0x32	0x00	0x11	0x05	0x01	0xE6
type = 12, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 5	val = 486 (4,86 [V])	

Sixth measurement					
0x32	0x00	0x11	0x06	0x00	0xF1
type = 12, prec = 2	md [s] = 0	addr_len = 1, meas_len = 2	addr = 6	val = 241 (2,41 [V])	

Revision history

Date	Version	Changes
11.06.2021	1.0	Initial release