

# Jomitek I3 Power Sensor

- User manual -

- Updated August 29, 2016 -

## Contents

<b>1 Introduction to the I3 Power Sensor system</b>	<b>4</b>
1.1 Overview	4
1.2 First time quick start guide	4
1.3 Sensor variants	4
1.4 Interface options and requirements in brief	4
1.5 Starter kit option and contents	5
<b>2 System installation</b>	<b>6</b>
2.1 Voltage sensor interfacing	6
2.2 I3 sensor mounting	7
2.3 Installation considerations	10
2.4 Communication hardware	10
2.5 Network architecture	10
2.6 Power supply	10
<b>3 Measuring voltages</b>	<b>11</b>
3.1 Measurement configurations	11
3.2 Voltage signal interfacing	12
3.3 Voltage measurement accuracy	12
3.4 A note on spread capacitances	13
3.5 Guide for setting up voltage cross talk cancellation	15
3.6 How to set up VoltageSync	16
3.6.1 Network architecture	16
3.6.2 Principle of operation	16
3.6.3 Configuring the sensors	17
3.6.4 Data interpretation on the slave sensor(s)	17
<b>4 Software and firmware structure</b>	<b>18</b>
4.1 Microcontroller platform and RTOS	18
4.2 File system	19
4.3 Lua scripting	19
4.4 Live Measurements	20
4.4.1 Live data settings	20
4.4.2 RMS calculation	21
4.4.3 Measurements	22
4.5 Event types and handling	22
4.5.1 Event metadata structure	22
4.5.2 Data structure	24
4.5.3 Functionality	25
4.5.4 Generating an event (for super users)	25
4.5.5 Viewing event content	27
4.5.6 Event types	29
<b>5 Web interface</b>	<b>29</b>
5.1 General structure	29
5.2 Sub pages	30
5.2.1 Explorer view	30
5.2.2 Device location	33
5.2.3 Cable configuration	34
5.2.4 Phase identification	34
<b>6 IEC-60870-5-104 configuration</b>	<b>35</b>

<b>7</b>	<b>Signal analysis</b>	<b>37</b>
7.1	WAV files . . . . .	37
7.2	Frequency analysis . . . . .	37
7.2.1	Fast Fourier Transform . . . . .	37
7.2.2	Total Harmonic Distortion . . . . .	37
7.2.3	FFT architecture . . . . .	38
7.2.4	FFT settings . . . . .	41
7.2.5	Viewing an FFT . . . . .	42
7.2.6	Downloading an FFT . . . . .	43
<b>8</b>	<b>Firmware and software updates</b>	<b>43</b>
8.1	Boot loading sequence . . . . .	43
8.2	Firmware upgrade . . . . .	44
<b>9</b>	<b>Advanced user interfaces</b>	<b>44</b>
9.1	Telnet interface . . . . .	44
9.2	FTP interface . . . . .	45
<b>10</b>	<b>Advanced mass reconfiguration and service</b>	<b>46</b>
10.1	Mass reconfiguration . . . . .	46
10.2	Operational overview and service . . . . .	47

## 1 Introduction to the I3 Power Sensor system

### 1.1 Overview

The I3 Power Sensor system is designed to measure voltage, current and derived quantities (i.e. reactive power, THD, earth faults) for multiple conductors in low- or medium voltage installations. It is not intended for protection purposes, and typically not sufficiently accurate for metering purposes. It is designed to provide an operational overview of a power grid, both during normal operation, as well as for fault events and fault states. This functionality is a key component in establishing a smart grid operation, assisting in areas such as reduction of non-delivered energy, increased workforce efficiency, prolonged equipment lifetime as well as timely equipment replacement.

A unique and patented feature of the I3 sensor system is the ability to measure currents of individual conductors in a 3 phase screened cable. By utilizing a number of magnetic field sensors within the I3 sensor casing, the position of the conductors within a 3 phase cable is calculated on the fly, and the current running in each conductor is then determined sample-by-sample. This eliminates the need for calibration both during the installation, and as a maintenance activity. In addition, it makes installation of current measurements extremely simple and flexible compared to installation of current transformers, especially for retrofitting scenarios.

A basic system consists of one or more capacitive voltage interfaces, towards a low- or medium voltage capacitor, integrated magnetic current sensors, a power supply using either PoE or DC via an Ethernet cable (RJ45 connector), and data transmission either via the Ethernet cable or using an integrated cellular modem (modem pending further development).

In summary an I3 Power Sensor is a very compact self-contained voltage and current sensor and data processing platform, including high level post-processing of measurements, remotely configurable, easy to install, easy to use, and virtually maintenance free.

For the latest documentation and software, please visit the I3 support web site at <http://jomitek.dk/support/i3>

### 1.2 First time quick start guide

1. Attach the sensor to a PoE switch on your network (DHCP assumed).
2. Determine the IP address of the sensor by looking into the allocation table in your network switch/router/-firewall or by use of the Jomitek Device Locator software which can be found at <http://jomitek.dk/downloads>.
3. Open your browser and go to the IP address of the sensor.
4. Explore the graphical web interface.

### 1.3 Sensor variants

The basic I3 sensor allows for current measurement on up to four conductors (i.e. 3 phases and ground) with a single phase voltage measurement. The I3+ sensor expands the number of voltage inputs to four input channels.

Both the basic I3 and the I3+ sensor variants come in two physical sizes intended for cable diameters of up to 60 mm and up to 80 mm respectively, which covers most use cases for low- and medium voltage cables.

The roadmap for further I3 sensor development includes options for digital input and/or output interfacing, serial communication and an integrated cellular modem. Prioritization of these hardware features will depend on industry feedback.

### 1.4 Interface options and requirements in brief

The sensor uses a single RJ45 Ethernet connector for both communication and the power supply. The sensor may be powered by either a Power over Ethernet (PoE) class 2 device or through 24V or 48V DC power supplied via the Ethernet cable.

When the sensor is connected, it can be accessed through a graphical web interface, IEC 60870-5-104 protocol, FTP, or a Telnet or Lua command line interface.

## 1.5 Starter kit option and contents

As a new user of the I3 Power Sensor, it is recommended to start out with the I3 Power Sensor Starter Kit. It contains the essentials to help build a test setup for evaluating the functionality and performance of the I3 system. The kit contains the following components:

- 1 × 60mm I3+ sensor
- 1 × 80mm I3+ sensor
- 2 × Capacitive voltage measurement interface with CEE connector for low voltage measurements
- 2 × CEE adapter
- Cable ties with fastening tool
- Adhesive rubber patches
- Adhesive velcro patches
- 2 × Screened Ethernet cables
- 1 × User manual



Figure 1: The I3 Power sensor starter kit.

## 2 System installation

Installation of the I3 sensor system is simple and requires no calibration for current measurements, and optional calibration for voltage measurements, depending on accuracy needs. Simply attach the I3 current sensor to the cable to be measured. Connect the capacitive voltage measurement interface connector(s), a grounding wire to the ground bolt, and finally attach the Ethernet cable. The installation procedure is described in detail in the following sections.

### 2.1 Voltage sensor interfacing

Jomitek supplies CEE connector based capacitive voltage measurement interfaces for voltage sources below 500V nominal peak, which are thus well suited for 110-240VAC measurements. The measurement interface uses built-in capacitors in the CEE casing and can therefore be directly connected between a voltage source with a CEE plug and the BNC connectors on the I3 sensor. If your voltage source does not use a CEE plug, you may use a CEE adapter to convert to any other interface relevant. For interfacing options towards medium voltage inputs, and further detail on the voltage measurement setup, please see section 5.



Figure 2: Front and bottom view of the I3 sensor showing the PoE connector, grounding bolt and voltage measurement BNC connectors.

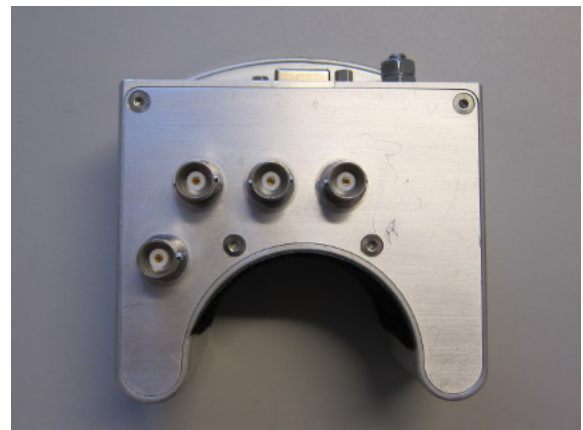


Figure 3: Voltage measurement BNC connectors. The default channel usage starting from the lower left going up to the right is: Null, L3, L2, L1.

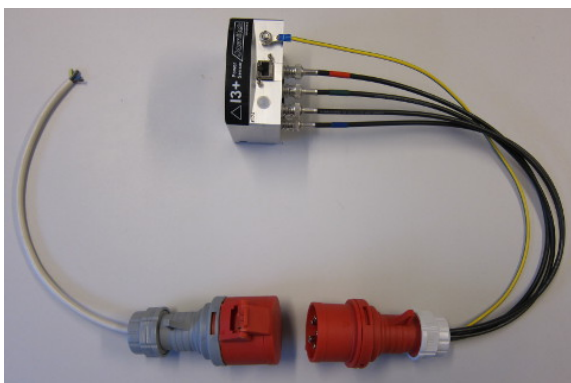


Figure 4: Capacitive voltage measurement interface attached to the I3 sensor. The CEE adapter is shown to the left.

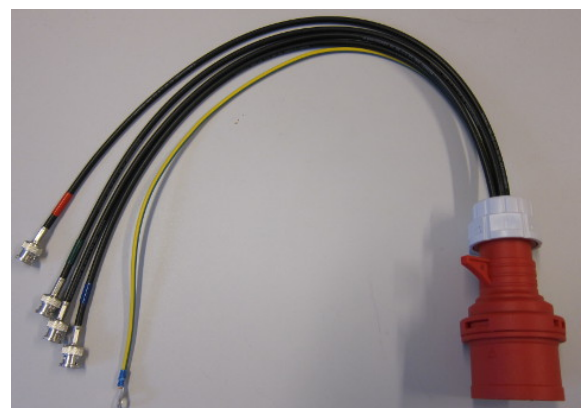


Figure 5: CEE connector for use with the capacitive voltage measurement interface. Note the integrated ground connector.

When using the CEE adapter for a custom interfacing, please refer to the voltage phase identification color codes listed in table 1 for correct wiring of your own connector.



Electrical potential	Capacitive voltage measurement interface (wire color)	CEE adapter (wire color)
Ground	Green/yellow	Green/yellow
Null	Blue	Blue
L1	Red	Brown
L2	Green	Black
L3	Black (no label)	Grey

Table 1: Electrical phase color coding for voltage measurement.

## 2.2 I3 sensor mounting

Place the current sensor on the cable and secure it with the provided cable ties as shown. The sensor must be mounted such that the voltage BNC connectors are facing down.

Note, that for an armored cable using steel that can be magnetized, the installation must be made at a point where such armor is removed. The I3 sensor cannot measure on multiple conductors through metals that can be magnetized. I.e. in switchgear installations this should be possible just before the cable enters an elbow connector, since the armoring is removed as part of the termination. Measuring through lead shielding or similar non-magnetizable metals is not a problem. A simple test is to use a permanent magnet, or magnetic screw driver to see if a particular shielding layer is magnetizable.



Figure 6: Place the sensor on the cable to be measured.



Figure 7: Make sure that the sensor is mounted such that the voltage BNC connectors are facing down.

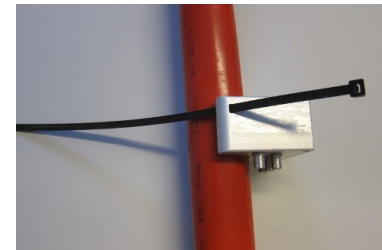


Figure 8: Insert the cable tie in the cutout on the sensor.

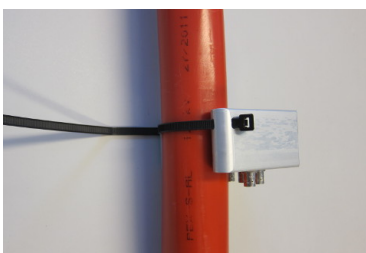


Figure 9: Wrap the cable tie around the cable and thread it through the cutout on the opposite side of the sensor.



Figure 10: Complete the cable tie connection.



Figure 11: Use the provided tool to tighten the cable tie.

The sensor may be lined with the included adhesive rubber or velcro patches if the cable diameter is significantly smaller than the sensor U-shape. It is recommended to always use a cable tie to complete the sensor installation. The rubber and velcro patches are optional.

Please see the following figures for an example on the use of the rubber and velcro patches.

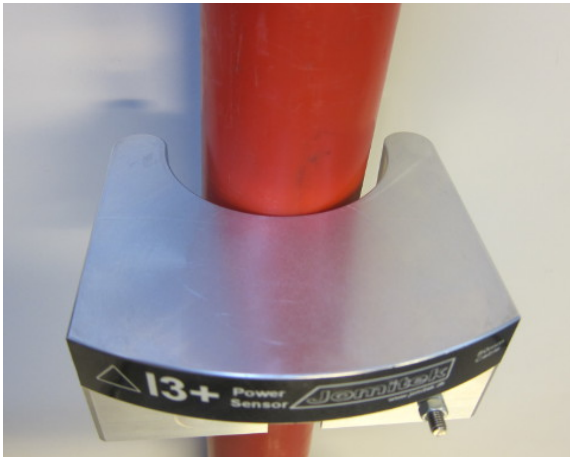


Figure 12: The cable diameter is too small for the sensor. A good fit is not possible.



Figure 13: Use the adhesive rubber patch to line the sensor.

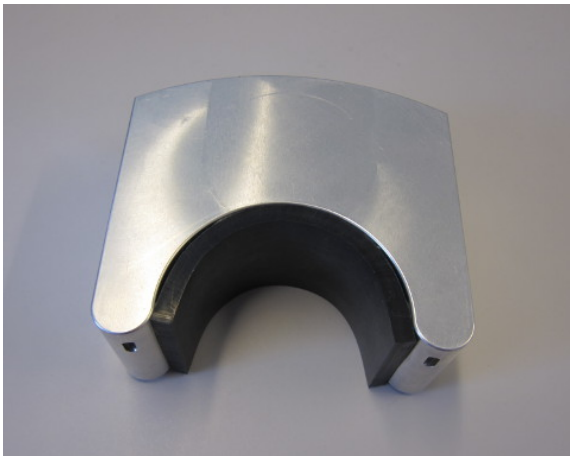


Figure 14: Adhesive rubber patch applied to the sensor in order to reduce the diameter.

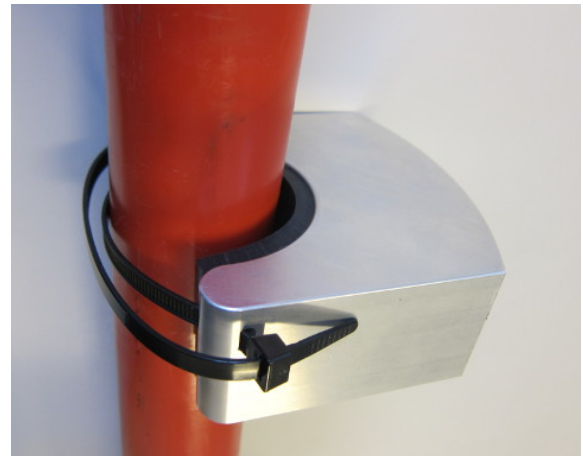


Figure 15: The cable and sensor diameters match, and a secure fit is now possible.

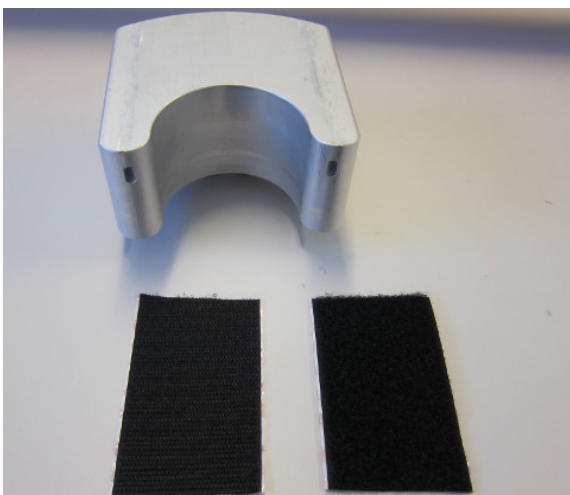


Figure 16: Adhesive velc patch applied to the sensor in order to reduce the diameter, and provide additional mounting stability.

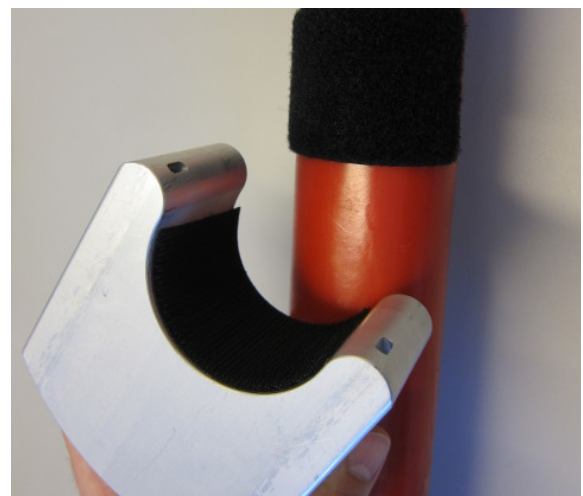


Figure 17: The cable and sensor diameters match, and a secure fit is now possible.



Attach the grounding cable on the front of the sensor, then plug in the voltage cables to the BNC connectors at the bottom of the sensor as shown in figure 18, to secure proper grounding and voltage sensor interfacing.

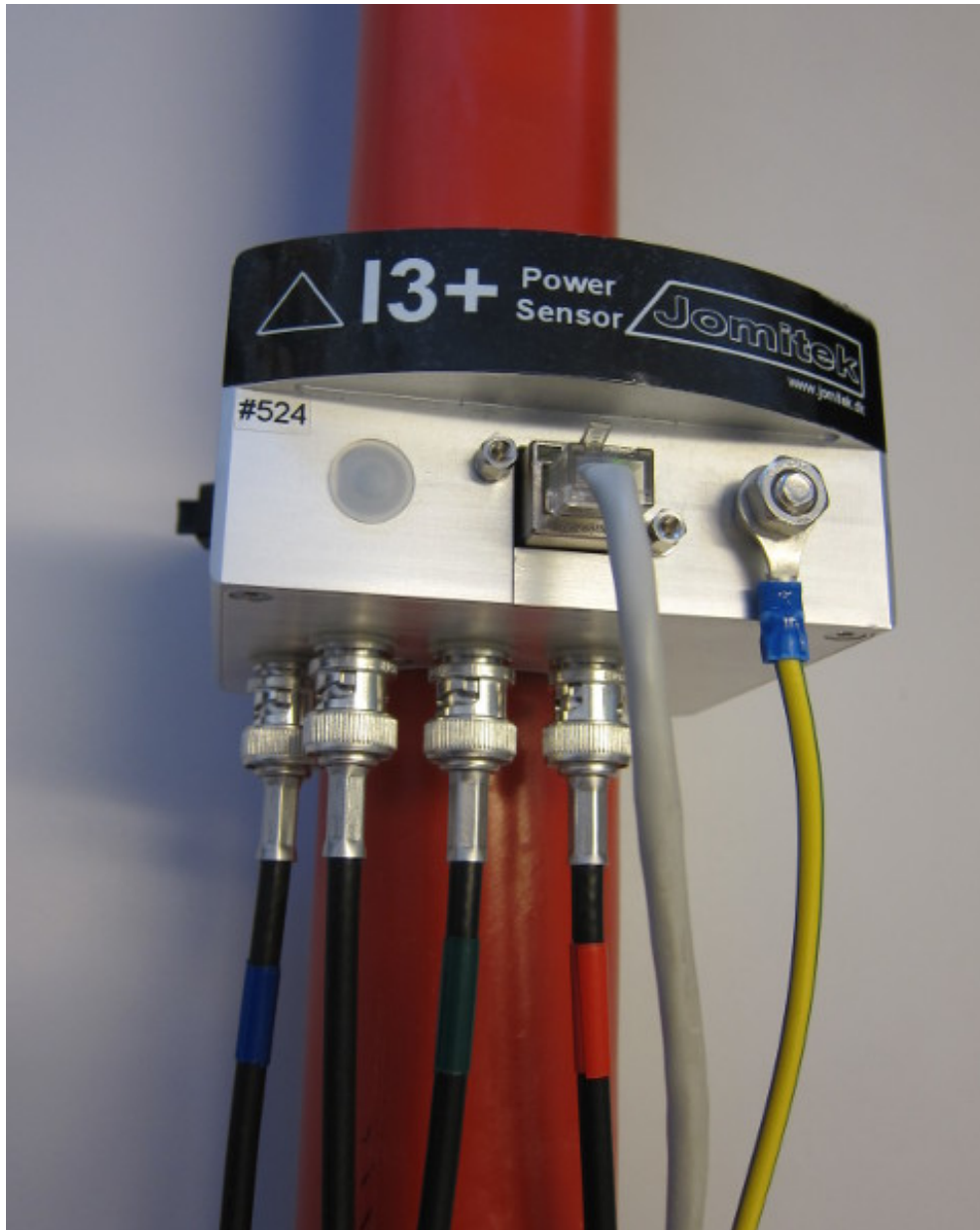


Figure 18: Attach the grounding cable and the voltage BNC connectors.

## 2.3 Installation considerations

The I3 current measurements work by measuring the magnetic field generated by the currents running in the cable. For optimum sensor performance, it is important to minimize magnetic disturbances close to the sensor. In general, this means that iron and other ferromagnetic materials should be avoided within 5-10 cm of the sensor box.

If building a laboratory setup for testing and evaluation, please note that the return conductor in a current loop should be kept as far away from the sensor as possible. This requirement may be suppressed if the sensor is configured to cancel out disturbances from a neighbor conductor.

The sensor is designed to be particularly weather resistant when mounted with the bottom part facing down.

If connection of BNC interfacing is limited by the physical space at the installation point, please note the option of using right angle connectors, rather than straight connectors. Please contact Jomitek, if this need should arise.

## 2.4 Communication hardware

Communication with the I3 sensor is via a standard Ethernet RJ45 cable. The Ethernet interface supports 10/100Mbps. A screened Ethernet cable must be used. The cable screen is connected to the I3 sensor box casing. The box must be earthed/grounded.

In at typical installation an industrial PoE Switch is used. High quality UV resistant Ethernet cables with robust connectors are recommended to ensure a long lasting installation.

During installation the technician must ensure that the LED's integrated in the Ethernet connector in the sensor starts blinking, when the PoE source is turned on and the cable is connected. When confirmed, the Ethernet communication can be checked i.e. using the 'Jomitek Device Locator'<sup>1</sup>. Note that the computer running the locator software must be connected to the same network, in the same subnet, as the I3 sensor, in order to be discoverable.

## 2.5 Network architecture

It is possible to realize the network architecture of a collection of I3 sensors in numerous ways, supported by the adaptability and wide usage of Ethernet. A sensor deployment for a single utility may have a volume ranging from hundreds to tens of thousands of sensor units, and many different communication backhaul media used, e.g. a closed radio network operated by the utility, use of the network of commercial cellular networks, via microwave links, fixed connections via fiber or copper, or even satellite based backhaul. With the diversity in sensor volume and backhaul used, it is good practice to establish a routing, and possibly NAT, architecture which supports placing all the sensors within the same subnet, seen from the SCADA point of view.

Having the sensor base placed within the same subnet ensures that a single server may easily reach the entire sensor base, and through this server performance and configuration management data may be distributed as needed within the network operations center and beyond.

Please contact Jomitek for verification or suggestions regarding your communication network architecture towards the I3 Power Sensors.

## 2.6 Power supply

The I3 sensor uses Power over Ethernet (PoE) as the power source. PoE is specified in the standard IEEE 802.3. Power is transferred in the 8-conductor Ethernet wire. The standard specifies 2 options for selection of wire-pairs, signal pair or separate pair, named Alternative A or B. The I3 sensor supports both options. PoE devices informs the Power Supply of the required power class at power up. The I3 sensors are characterized as Class 2 devices,

<sup>1</sup><http://jomitek.dk/downloads/tools>

drawing power in the range 3.84W-6.49W.

If the installation does not offer a PoE power supply, an external 24/48VDC supply may be used. This supply has to enter through the Ethernet cable. A special splitter cable can be ordered at Jomitek, which makes use of the mode B pins 4/5 (DC+) and pins 7/8 (DC-).

The power supply circuit is electrically isolated sensor electronics by a 1500V isolating transformer.

802.3af Standards A and B from the power sourcing equipment perspective

















Pins at switch	T568A color	T568B color	10/100 mode B, DC on spares		10/100 mode A, mixed DC & data		1000 (1 gigabit) mode B, DC & bi-data		1000 (1 gigabit) mode A, DC & bi-data	
Pin 1	 White/green stripe	 White/orange stripe	Rx +		Rx +	DC +	TxRx A +		TxRx A +	DC +
Pin 2	 Green solid	 Orange solid	Rx -		Rx -	DC +	TxRx A -		TxRx A -	DC +
Pin 3	 White/orange stripe	 White/green stripe	Tx +		Tx +	DC -	TxRx B +		TxRx B +	DC -
Pin 4	 Blue solid	 Blue solid		DC +		Unused	TxRx C +	DC +	TxRx C +	
Pin 5	 White/blue stripe	 White/blue stripe		DC +		Unused	TxRx C -	DC +	TxRx C -	
Pin 6	 Orange solid	 Green solid	Tx -		Tx -	DC -	TxRx B -		TxRx B -	DC -
Pin 7	 White/brown stripe	 White/brown stripe		DC -		Unused	TxRx D +	DC -	TxRx D +	
Pin 8	 Brown solid	 Brown solid		DC -		Unused	TxRx D -	DC -	TxRx D -	

Figure 19: Image source: [https://en.wikipedia.org/wiki/Power\\_over\\_Ethernet](https://en.wikipedia.org/wiki/Power_over_Ethernet)

### 3 Measuring voltages

#### 3.1 Measurement configurations

All voltage measurements use the sensor box grounding bolt as the reference level. It's therefore important to ensure proper grounding of the sensor.

Starting with the most complete voltage measurement setup, an I3+ is used to measure 3 individual phase voltages, as well as e.g. a transformer star point voltage, or any auxillary voltage of interest in the measurement setup. In combination with current measurements, such a setup ensures a full overview on the power load, quality, and any fault situation on the individual phases measured.

A basic I3 sensor (or an I3+ with only 1 input in use), allows for 1 voltage measurement point. The I3 sensor will automatically generate an internal copy of the remaining 2 phases, shifted 120 degrees to both sides. This allows for a more cost efficient implementation, since the voltage measurement points can be reduced to a minimum. This option allows for what would in many cases be a reasonable estimate of the reactive and active power load of the 2 phases with no direct voltage measurements. It is, however, not at setup which allows for full transparency in fault situations, e.g. earth faults. The measurement limitations of such a setup should be weighed against the added installation costs of a full voltage measurement.

The I3 sensor can also be mounted on a current conducting cable with no voltage input. With no other I3 sensors locally, this will limit the measurement capabilities to be based on currents alone. This means that reactive and active power cannot be measured, as well as any type of directional measurement (e.g. direction of energy flow).

A powerful option for saving on equipment and installation costs for substations with several I3 sensors installed, is to utilize the Jomitek VoltageSync capability which can distribute the voltage measurements from a single I3+ to a number of locally installed I3 sensors via Ethernet, see section 3.6 for more information.

### 3.2 Voltage signal interfacing

The I3 sensors employ the same signal conditioning for all voltage inputs. The inputs are based on measurement via capacitive voltage division, and are protected up to 2kV. The high voltage capacity for a given nominal voltage should be selected such that:

$$C_{hv} = 5 \times 10^4 \text{ [pF} \times \text{V]} / U_{\text{nomRMS}}$$

This relation is exemplified in table 2:

$V_{\text{nomRMS}}$	$C_{hv}$	Comment
57.7 V (100 V/ $\sqrt{3}$ )	$\leq 820$ pF	Standard output from voltage transformer
110 V	$\leq 470$ pF	110 V RMS, as used in the US, low voltage
230 V	$\leq 220$ pF	230 V RMS, as used in Europe, low voltage
5.77 kV (10 kV/ $\sqrt{3}$ )	$\leq 8.6$ pF	10 kV RMS phase-phase, medium voltage level
6.92 kV (12 kV/ $\sqrt{3}$ )	$\leq 7.2$ pF	12 kV RMS phase-phase, medium voltage level
9.23 kV (16 kV/ $\sqrt{3}$ )	$\leq 5.4$ pF	16 kV RMS phase-phase, medium voltage level
11.5 kV (20 kV/ $\sqrt{3}$ )	$\leq 4.3$ pF	20 kV RMS phase-phase, medium voltage level
13.8 kV (24 kV/ $\sqrt{3}$ )	$\leq 3.6$ pF	24 kV RMS phase-phase, medium voltage level
18.5 kV (32 kV/ $\sqrt{3}$ )	$\leq 2.7$ pF	32 kV RMS phase-phase, medium voltage level
28.9 kV (50 kV/ $\sqrt{3}$ )	$\leq 1.7$ pF	50 kV RMS phase-phase, medium voltage level

Table 2

Note that you may never make a direct galvanic connection to the voltage to be measured. You must always use a capacitor in between, appropriately selected based on above formula and table. Selecting a capacitor of much lower value than the maximum recommended, will reduce the dynamic range, and thus the measurement resolution of the nominal voltage. It is generally recommended to target a capacitor value no less than 25% of the maximum allowed value for a given voltage.

The maximum allowed capacitance value for the medium voltage levels fits well with industry standard silicone based capacitive test points, e.g. those found in elbow connectors. Note that ceramic based medium voltage capacitors should be avoided, as their capacity is both too high, and exhibits unlinear behavior. Note, that in order to obtain high accuracy voltage measurements, the capacitive test point must be screened, such that spread capacitances are avoided. Jomitek produces screening cups for elbow connectors. Please contact Jomitek for more information on medium voltage capacitive screening options.

Configuration of the capacity used, and further voltage scaling options are available in the '*settings.conf*' file accessible via FTP or the sensor web interface.

### 3.3 Voltage measurement accuracy

The capacitive measurement principle used for the voltage signal interfacing provides a wide range of options for the physical measurement setup. In order of measurement accuracy the main options are listed in table 3. The indicated accuracies are indicative based on normally seen implementations. Note that for most O&M purposes, a precision of 5% should be satisfactory, to determine the load situation, direction of energy flow, earth faults, and similar key parameters.

Note that while a low voltage (LV) measurement may provide a relatively high accuracy of the measured voltage, and a fairly cheap interfacing solution, some major drawbacks exists for this approach, when the current is measured on the medium voltage side of a transformer.

Measurement interface type	Accuracy uncalibrated	Accuracy calibrated
Voltage transformer (VT) via 820pF capacity	3%	0.5%
LV via 470pF capacity	3%	0.5%
JCap, a high precision capacitive insulator	N/A	1%
Elbow connector with integrated capacity, shielded	5%	2%
MV stand-alone capacitor (e.g. insulator)	10%	5%
MV attachable capacitor plate	20%	10%

Table 3

- The transformation ratio may be variable, so the actual voltage level on the medium voltage side is not possible to describe with just a constant conversion ratio.
- An earth fault on the MV side will typically not be possible to detect on the LV side of a transformer.
- High frequency content may be distorted by the transformer, making it very difficult to correctly assess MV THD, and the MV waveform during transient events.

The mentioned drawbacks must be compared to the implementation cost, and the target for the monitoring solution. If only basic measurement parameters (such as voltage, current, power flow) is of interest, then low voltage measurement interfacing may be adequate for the purpose.

### 3.4 A note on spread capacitances

In particular for the capacitors suitable for the medium voltage level, the physical distances involved between capacitor plates can lead to significant spread capacitances being introduced. Ideal solutions for such measurement systems involve well screened setups, such as using an elbow connector with integrated capacitive end plug, and placing a conducting screening cup as a termination for any parasitic/spread capacitances. This ensures that any capacitive coupling between the measurement plate and the surrounding material is constant, and preferably low compared to the capacitive coupling towards the medium voltage conductor.

In some cases it's not possible to achieve a well screened solution at a reasonable cost. Non-ideal solutions may then be considered, at the cost of the usefulness of certain measurement parameters. Spread capacitances may lead to two main measurement deviation effects:

- Varying spread capacitance towards ground potential
- Cross talk between voltage phases

The first effect will change the capacitive voltage division ratio, leading to incorrect scaling of the voltage amplitude, and thus all measurement parameters related to the voltage amplitude. A typical example is with a setup where the unscreened medium voltage capacitor is near the enclosure door/lid of the installation. When opening the door, the spread capacitance to ground changes significantly, and thus the voltage division ratio. This effect may indicatively lead to 0-15% sensitivity variation.

The second effect is almost always experienced in combination with the first. It involves a capacitive coupling from one or more neighboring medium voltage conductors towards the measurement plate of the primary phase being measured. Effectively this results in the raw measurements of a given input being a mix of signals, with the major component being the primary phase measured, but with neighboring phases typically adding their signals at a 0-25% sensitivity level compared to the primary.

Cross talk between voltage phases can often be considered fairly static, as they are primarily dependent on the physical shape of a given type of breaker equipment. If the cross talk sensitivities have been mapped (calibrated) for a particular physical configuration, it can be reused on similar equipment models. It must however be confirmed that the enclosure is also fairly static, such that the first effect does not affect the second. One example of a measurement setup involving both spread capacitance measurement effects is on a Magnefix switchgear using a RetroCap measurement interface, as illustrated, see figure 36. In this case, the capacitive measurement point



is glued to the surface of the fuses of a Magnefix switchgear. These are very easy to install, but unfortunately also very sensitive to cross talk and spread capacitances to ground.



Figure 20: Horstmann RetroCaps mounted on Magnefix switchgear.

#### **Accuracy of measurement parameter vs. type of voltage measurement setup**

If the cross talk ratios have been determined, the I3 sensor platform supports on-the-fly de-mixing of the voltage measurement signals, effectively cancelling the cross talk phenomena. As suggested by the above table, it is strongly recommended to enable de-mixing of voltage cross talk, on installations with unscreened high voltage capacitors. Installation specific calibration will likely enable performance equal or close to a well screened system, but the cost of performing installation specific calibration will likely discourage such use cases. As such, installations will generally either be well screened, where needed, and otherwise non-screened, but calibrated according to the type (model) of the installation.



Measurement parameter	Well screened	Non-screened, specifically calibrated	Non-screened, calibrated by model	Non-screened, no de-mixing
phase_delta	OK	(OK)	Not OK	Not OK
phase_voltage	OK	(OK)	Not OK	Not OK
power_factor	OK	(OK)	Not OK	Not OK
rms_apparent_power	OK	OK	(OK)	Not OK
rms_reactive_power	OK	(OK)	Not OK	Not OK
rms_real_power	OK	(OK)	Not OK	Not OK
rms_voltage	OK	(OK)	(OK)	Not OK
thd_voltage	OK	OK	(OK)	(OK)

Table 4: **OK** = Measurement will perform according to I3 data sheet specification, **(OK)** = Measurement will likely perform according to I3 data sheet specification, but with no commitment from Jomitek, **Not OK** = Measurement should be considered indicative at best, but with a high risk of being misleading

### 3.5 Guide for setting up voltage cross talk cancellation

This guide is only relevant for poorly screened voltage measurement configurations.

#### Getting started

In order to enable the default cross talk cancellation settings, the following parameter should be set, either directly in the *'settings.conf'* file or via the web interface: *i3.calibration.voltage\_cross\_talk = true*. **IMPORTANT:** Note that the interfacing to an I3+ must be symmetrical, i.e. of the following order: L1 I3 interface connected to either the top or bottom phase of the physical measurement setup, L2 to the center, and L3 opposite to the connection point of L1.

#### Advanced configuration and calibration

The cross talk figures provided per default are based on the behavior of a Magnefix switchgear. For other measurement setups, it is recommended to verify that the same cross talk relation applies, and if not, the cross talk parameters must be updated to reflect the behaviour of the relevant measurement setup. These parameters can be found in the *i3.calibration.voltage\_cross\_talk\_coeff* folder. Using the web interface, the subfolder L1 reflects the relative signal strengths received on L1 from the phase connected to L1 (denoted phase01), L2 (phase02), L3 (phase03) and the auxillary input (phase04) respectively. The same logic applies for the remaining subfolders. It is the same reference input level used across all the calibration measurements. It is recommended to use the L2 input, when applying a reference voltage signal to the phase of L2 capacitor, as the mentioned reference.

As an example, figure 21 illustrates a case where:

- A reference voltage was applied to the phase being interfaced to L2 on the I3, while the other phases were kept grounded, and the resulting output noted for the L2 input. This is the reference input level the measurements was normalized with, and it is with respect to this level, that the *i3.voltage\_setup.phase02\_hv\_capacitor* value was aligned, such that the absolute voltage measured by the I3 matches that of the known voltage level of the reference voltage.
- The same reference signal was then applied to the phase being interfaced to L1, L2 and L3, one at a time, and with the other phases grounded for each measurement. The input on L1 was normalized with respect to the first measurement, and a signal strength of 127.0%, 11.2% and 3.0% was noted.
- The auxillary input was not connected, so the cross talk for AUX is set to 0%.
- The AUX input, is assumed free of cross talk and therefore indicated with 0% for cross talk from signals relating to L1-L3.

Following the guideline above, a capacitance value of **3.2 pF** was found for the Horstmann RetroCap sensors mounted on a Magnefix switchgear. This value must be specified in *i3.voltage\_setup.phase0[x]\_hv\_capacitor*, with [x] being 1-3.

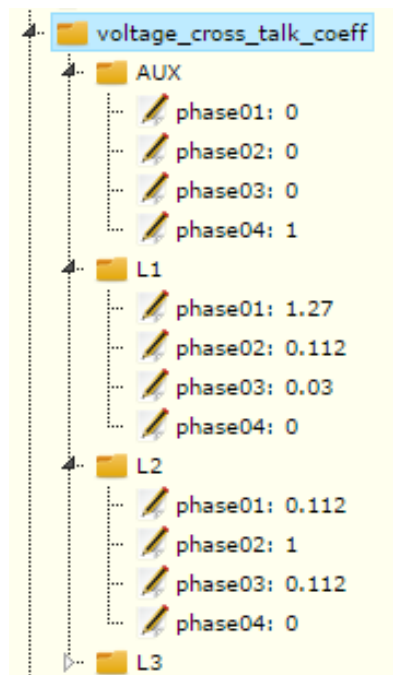


Figure 21

### 3.6 How to set up VoltageSync

This section describes the VoltageSync system, which can be used to synchronize basic voltage information between several locally installed I3 sensors, with one I3+ acting as a voltage master.

#### 3.6.1 Network architecture

The voltage synchronisation system requires three essential components:

1. A Jomitek configured MikroTik Ethernet switch.
2. An I3+ sensor configured as master, measuring both currents and voltages.
3. One or more I3 sensors configured as slave, measuring only currents.

Figure 22 shows how to connect these components. Please note that all I3 sensors in a voltage synchronization network must be connected directly to the same Jomitek configured MikroTik Ethernet switch.

#### 3.6.2 Principle of operation

The Jomitek configured MikroTik Ethernet switch periodically broadcasts a data package used as a recurring time synchronization reference for the master as well as the slave sensors.

When the master and slave sensors receive the synchronization package they set an internal time stamp, and the master sensor then calculates and broadcasts its voltage and phase information to all the slave sensors in the network.

Using the information received from the master sensor combined with the synchronization reference time stamp, the slave sensors calculate the phase difference between their locally measured current and the remotely measured voltage. This phase difference is used to calculate RMS power values. Since all voltage data recreated on the slave sensors are based on RMS values, it is not possible to inspect the voltage time series on the slave sensors, nor to see voltage FFT data.

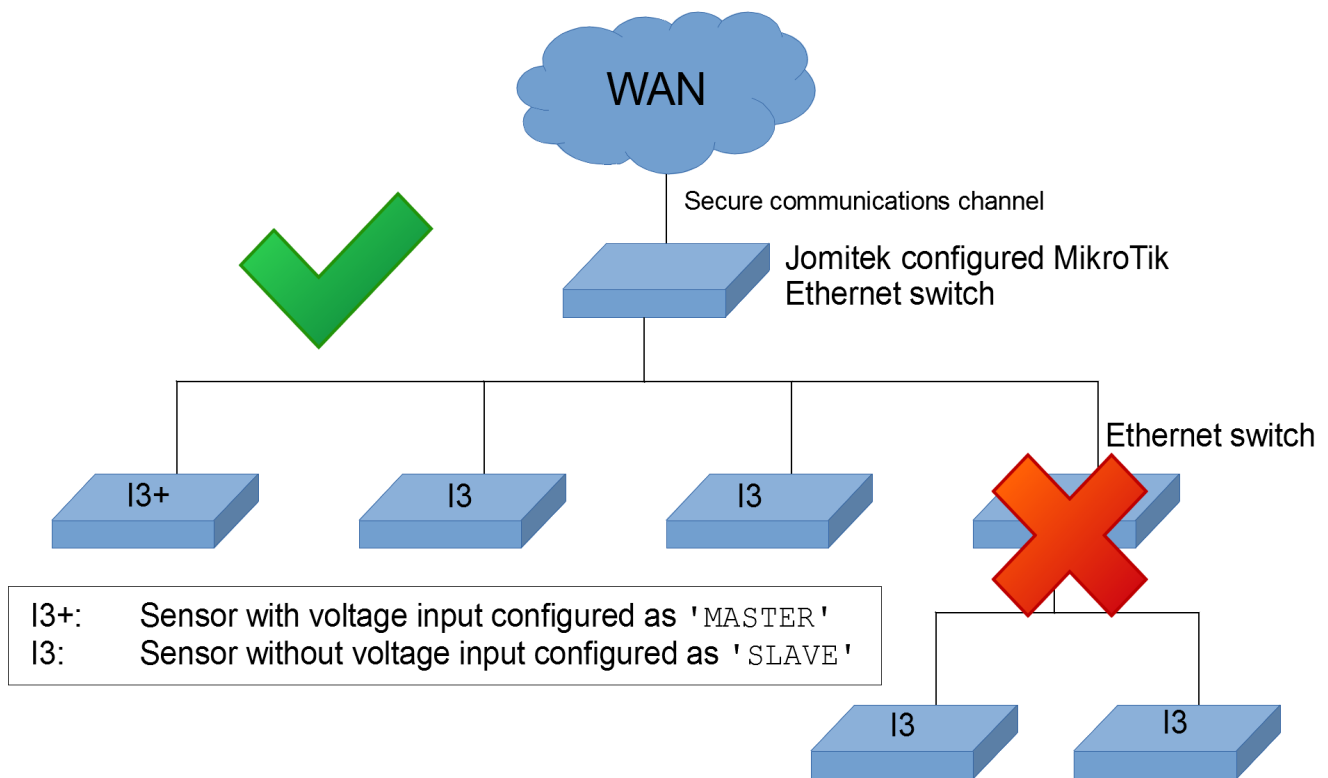


Figure 22: Network diagram showing how to connect master and slave configured sensors. Sensors should only be connected directly through the Jomitek MikroTik switch with no other network switches layered in between. While only one master sensor may be configured for any given installation, the only practical limit to the number of slave sensors is the number of ports available on the router.

### 3.6.3 Configuring the sensors

Voltage synchronization requires one sensor, preferably an I3+, to be configured as master and one or more I3 to be configured as slaves.

To configure a sensor as master, the `voltage_sync` flag must be set to 'MASTER' via the web interface, see figure 23. Make sure the setting is saved, and restart the sensor for the change to take effect. After the restart the sensor will broadcast its voltage and phase information to the slave sensors.

To configure a sensor as slave, the `voltage_sync` flag must be set to 'SLAVE' via the web interface, see figure 23. Make sure the setting is saved, and restart the sensor for the change to take effect. After the restart the sensor will receive voltage and phase information from the master sensor.

To disable voltage synchronization, the `voltage_sync` flag must be set to 'DISABLE' via the web interface, see figure 23. Make sure the setting is saved, and restart the sensor for the change to take effect. Voltage synchronization will now be disabled on the device.

### 3.6.4 Data interpretation on the slave sensor(s)

Since only voltage RMS levels rather than full time series measurements are shared by the master sensor, the voltage measurement time series and FFT data on the slave sensors will show zero.

The computed phase difference between remotely measured voltages and locally measured currents on the slave sensor is noise sensitive. It is therefore recommended to use long averaging times for the SCADA readout (`average_time > 20`) for RMS values. RMS averaging times can be configured under: `settings` → `i3` → `post_processing`.

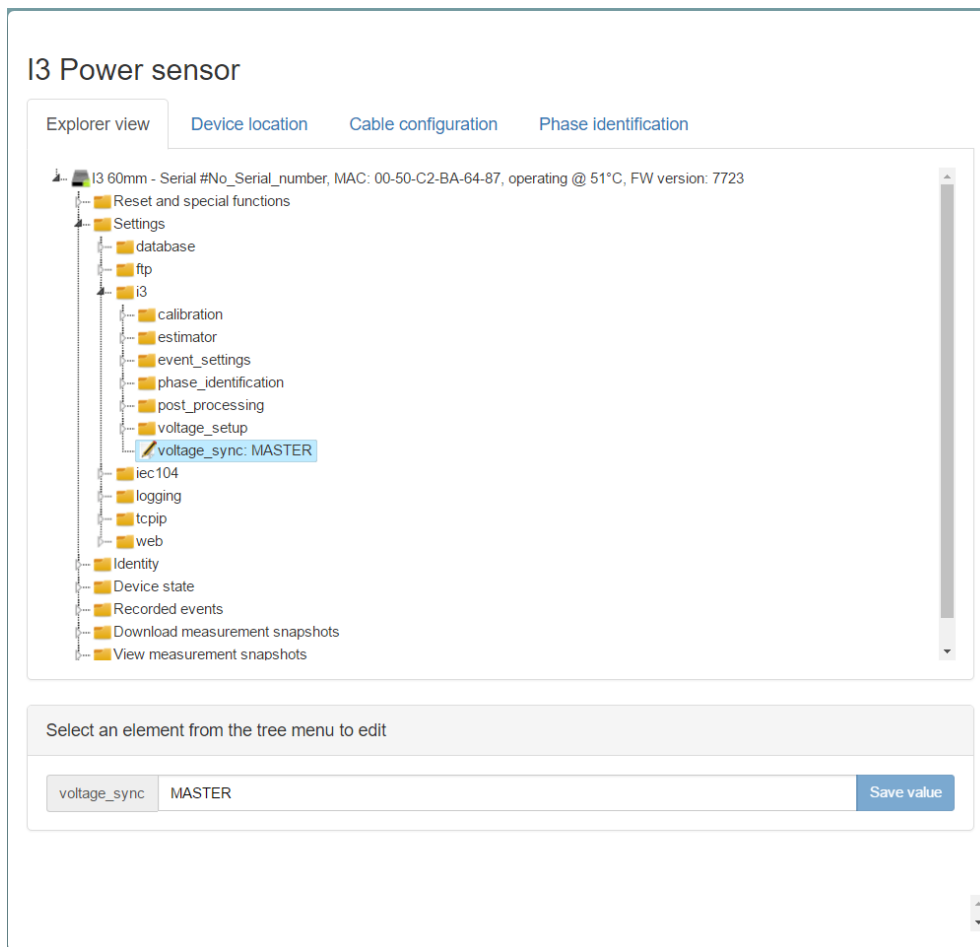


Figure 23: The voltage\_sync flag can be set to either 'MASTER', 'SLAVE' or 'DISABLE' in the web interface. In this figure the flag is set to 'MASTER'.

## 4 Software and firmware structure

### 4.1 Microcontroller platform and RTOS

The I3 sensor features an ARM Cortex M4 microprocessor operating at a clock frequency of 96Mhz. A real-time operating system (RTOS) manages processor resource allocation for measurement channel sampling, low- and high level processing as well as communication tasks.

The platform supports Timing over Packet (ToP) functionality, which depending on the communication infrastructure may supply time synchronization on the microsecond level locally between several I3 sensors, and on the millisecond level via i.e. a cellular network. Development is in progress to utilize this support, which will enable synchronization of voltage information across several I3 sensors within the same LAN, and thus lower the overall installation cost. Please contact Jomitek for more information.

The firmware being used in the microprocessor can be securely updated over the air using fully encrypted firmware images, which enables addition of further functionalities after installation, i.e. ToP support.

The I3 software includes a Lua scripting engine which allows for execution of custom code and dynamical re-configuration of the sensor. This allows for easy integration and customization of live measurements, reports, event summaries, etc. beyond the default capabilities supplied by Jomitek. Requests for assistance on any such customization are welcome.

## 4.2 File system

The I3 sensor platform features a FAT formatted 24MB file system used to store non-volatile data. These data include software in the form of Lua scripts and web server content, as well as configuration files, log files, measurement data (typically .wav files), etc.. The drive structure is described in the following.

- 0:/root - The root folder containing all other folders. This folder can be accessed through the FTP interface when logged in with the root user (super users only).
- 0:/root/debug - Jomitek development folder containing debug information.
- 0:/root/service - The main user facing folder containing all user relevant sub folders. This folder is accessible through the FTP interface when logged in with the user customizable FTP account (default account name: service).
- 0:/root/service/conf - System configuration files.
- 0:/root/service/data - Measurement data such as .wav file time series and event logs.
- 0:/root/service/log - System related log files.
- 0:/root/service/software\_update - Folder for uploading firmware and software update images.
- 0:/root/lua - Lua scripts and Lua plugin modules.
- 0:/root/web - Contains all files needed to serve the web interface. This folder is copied to RAM on system boot (to increase web server speed), which means that any update in this folder requires a system reboot to be reflected on the web server.

## 4.3 Lua scripting

Lua is a lightweight scripting language which allows for dynamic reconfiguration of the sensor, and is used to run high level non-time critical tasks on the I3 sensor.

A Lua command line terminal interface can be accessed through a raw TCP connection with Putty by connecting to the sensors IP address on port 2840.

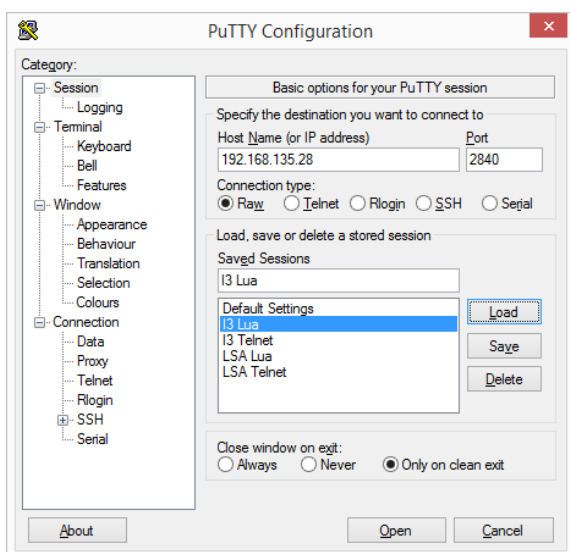


Figure 24: Putty Lua Session configuration.

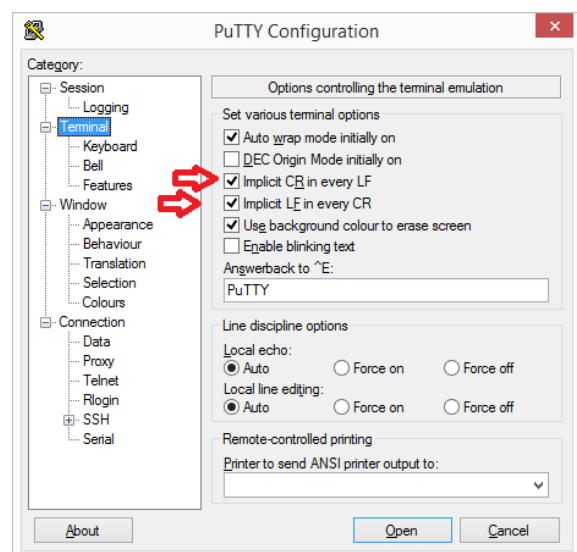


Figure 25: Putty Lua Terminal configuration.

The Lua interface is intended for advanced configuration and debugging and should only be used by experienced users.

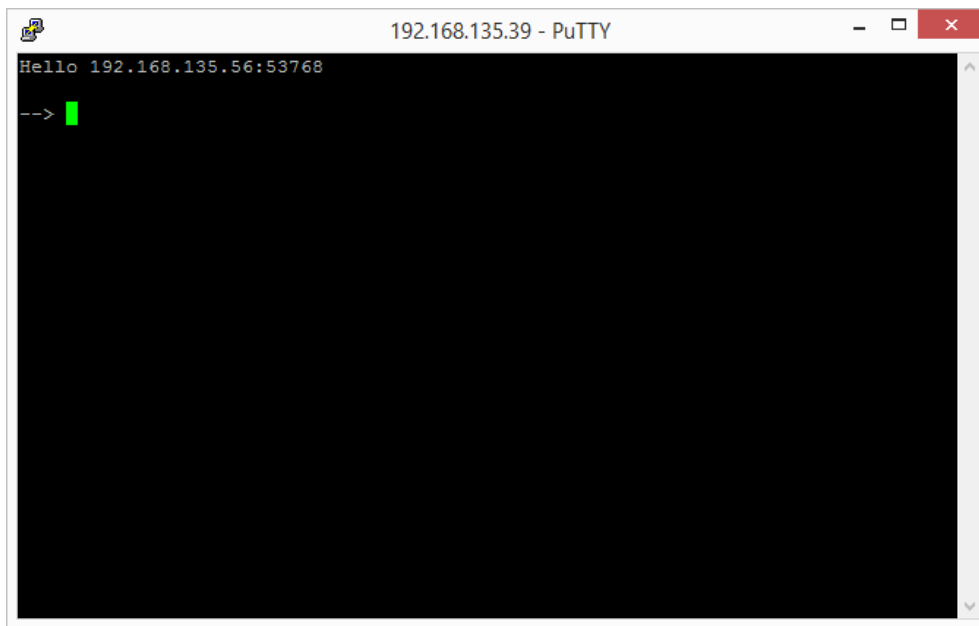


Figure 26: The Lua command prompt.

More information on the Lua scripting language can be found at <http://www.lua.org/about.html> and a full reference is available at <http://www.lua.org/manual/5.2/>.

## 4.4 Live Measurements

### 4.4.1 Live data settings

There are three parameters to consider when configuring the live data calculations. There are two RMS time constants and one boolean flag to signal a 180° phase change on all estimated current samples. These parameters may be modified in *settings.conf* which requires a restart of the sensor to apply the change, the relevant part of *settings.conf* is given below, or via the web-interface.

#### Listing 1: From settings.conf

```
i3 = {  
  post_processing = {  
    average_time = 60,           -- The averaging time used for SCADA measurements  
    flip_direction_of_current = false, -- Flips direction of the measured current  
    quick_time = 1,             -- The averaging time used for measurements shown  
                                -- in the 'Live measurements' view in the web interface  
  },  
}
```

To configure the same parameters from the sensor web interface, go to the *post\_processing* menu as seen in figure 27. To change the value of the settings you select the element you want to change and type in the desired value in the input box followed by a click on the 'save value' button. An explanation of the three different settings are given below. For further details on how to configure the sensor see section 5. Note that updating parameters via the web interface will usually not require a reboot of the sensor.



DEVICE MONITORING AND CONFIGURATION OVERVIEW

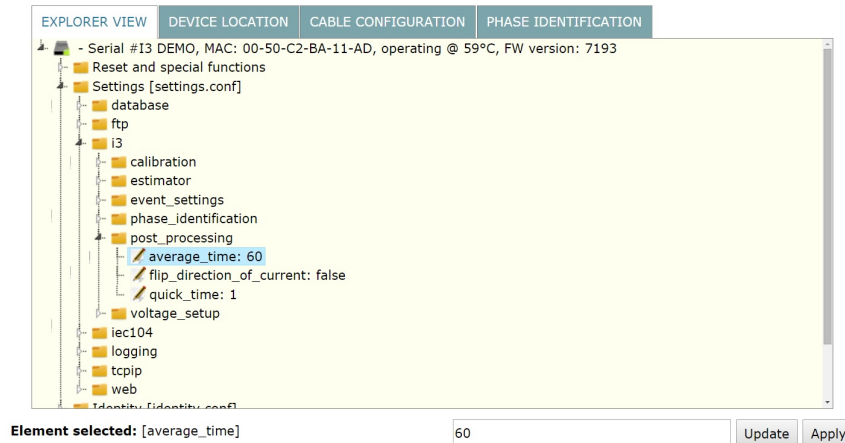


Figure 27: View of the settings menu.

**average\_time**

Is the number of seconds over which the RMS values used by the IEC60870-5-104 is calculated.

**quick\_time**

Is the number of seconds over which the RMS values used by the 'Live measurements' plots are calculated.

**flip\_direction\_of\_current**

Imposes a sign change on all the measured currents thus changing their phase 180°.

**4.4.2 RMS calculation**

All RMS values are true RMS values, this means that there is no assumption made regarding signal shape or frequency content.

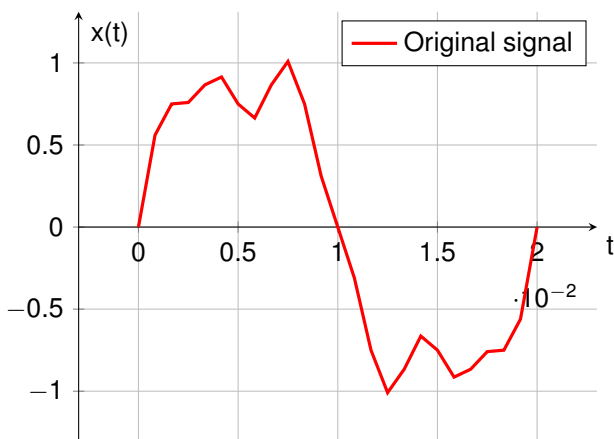


Figure 28: Signal with a main frequency of 50 Hz with harmonic distortion.

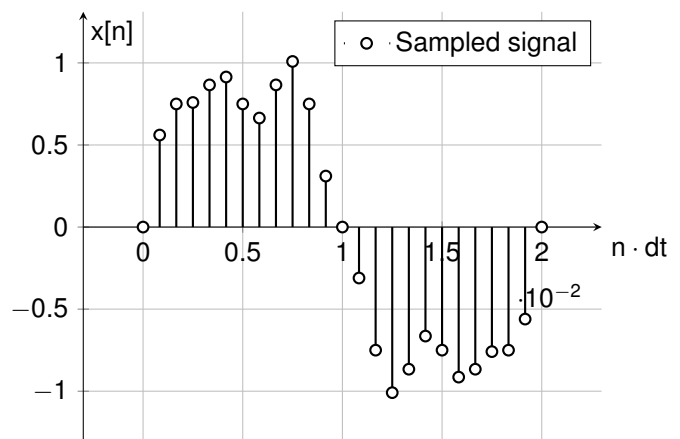


Figure 29: Signal with a main frequency of 50 Hz with harmonic distortion sampled at 1250Hz.

Considering the sampled signal in figure 29 the RMS value is calculated using equation 1. A special case is used when calculating the RMS of phase values, here the calculations are done using imaginary numbers to prevent wrapping issues around 0 and  $2\pi$ . The RMS values of the voltages and currents are calculated per phase based on a downsampled 1250Hz signal. The RMS values are refreshed towards Lua (which reports to the SCADA and

the web interface) with a fixed frequency of 1s. The RMS time constants themselves may be set to any value between 0.1 and 1000.0 seconds, with 100ms quantization steps.

$$X_{RMS} = \sqrt{\frac{1}{n} \sum_{n=0}^{n-1} x[n]^2} \quad (1)$$

#### 4.4.3 Measurements

The sensor measure both voltages and currents with a sampling frequency of 31250Hz as well as downsampled to 1250Hz. Both versions are kept in ringbuffers which for the 31250Hz samples cover at least 100ms and 1250Hz at least 2500ms. These measurements make it possible to perform high level processing of all voltages and currents for the individual phases, such as THD, frequency analysis, extracting time series snapshots of short circuit or earth fault events, as well as the following data points:

##### Phase\_voltage

The phase is measured using a zero-crossing method with interpolation. The first active voltage channel is automatically assigned to be the phase reference and is therefore always zero.

##### Phase\_current

The phase is measured using a zero-crossing method with interpolation. If there are no active voltages, the first active current is chosen as a phase reference instead, the reference is always zero.

##### Phase\_delta

Phase\_delta is defined as : **phase\_delta = phase\_voltag-phase\_current**

##### Power\_factor

Power\_factor is defined as : **power\_factor = cos(phase\_delta)**

##### Apparent\_power

Apparent\_power is defined as : **apparent\_power = current<sub>rms</sub> · voltage<sub>rms</sub>**

##### Real\_power

Real\_power is defined as : **real\_power = apparent\_power · power\_factor**

##### Reactive\_power

Reactive\_power is defined as : **reactive\_power = apparent\_power · sin(Phase\_delta)**

##### Main\_harmonic\_frequency

The main\_harmonic\_frequency is measured using a zero-crossing method with interpolation.

The sensor does not measure the current directly, instead it builds a model of the magnetic environment and uses this model to infer the currents from the magnetic field projected on to the magnetic field sensors inside the I3 sensor. In other words, it can scan the magnetic fields, in order to sense the position of the conductors, and the absolute value of the current running in each of them.

## 4.5 Event types and handling

### 4.5.1 Event metadata structure

The format for representing a collection of different events in a common structure is based on Lua tables. An example of the structure is given below:

Listing 2: Basic structure of event metadata.

```
{
  eventIdentifier = {
    eventType,
    eventTimeStamp = {
      sec,
      usec,
      iso,
    },
    eventFiles = {
      exampleFileName1,
      exampleFileName2,
    },
    eventParameters = {
      exampleParameter1,
      exampleParameter2,
    },
    plotOptions = {
      '{"t':'Example file 1 Title','u':'Example measurement unit','s':100,
      'x':'Example X label [ms]','y':'Example Y label [kA]',
      'l':['Example Channel 1 legend','Example Channel 2 legend']}',
      '{"t':'Example file 2 Title','u':'Example measurement unit','s':10,
      'x':'Example X label [ms]','y':'Example Y label [kA]',
      'l':['Example Channel 1 legend','Example Channel 2 legend']}',
    },
  },
}
```

The format is simply a table of parameters characterizing each event, uniquely defined by "eventIdentifier". The table contains all recorded events and can contain different event types. All events share the following entries (there are no limits on the number of additional entries):

- **eventType** - A string containing the type of event. Possible "eventType"s include be "lightningStrike", "earthFault", "shortCircuit", etc.
- **eventTimeStamp**
  - **sec** - An integer representation of the event timestamp in POSIX time format. For example, 22 Sep 2015 15:04:01 GMT would correspond to 1442934241000.
  - **usec** - An integer representing the microsecond part of the timestamp.
  - **iso** - A string containing the timestamp in text for example, "2015-12-02!T10:39:05.227Z"
- **eventFiles** - A lua list of strings containing the filename(s) to the file(s) related to the event. This table can be empty if no files are attached.
- **eventParameters** - A Lua table containing "eventType" specific parameters. Different "eventType"s can have different event parameters, but they must be contained in the "eventParameters" table.
- **plotOption** - A lua list of JSON formatted strings detailing the options needed for plotting and displaying the respective event files.

Example:

Listing 3: Example of event metadata.

```
{
  e00001 = {
    eventType = "earthFault",
    eventTimeStamp = {
      sec = 1449052745,
      usec = 227000,
      iso = "2015-12-02T10:39:05.227Z",
    },
    eventFiles = {
      "e00002_file_A.wav",
      "e00002_file_B.wav",
    },
    eventParameters = {
      faultPhase = 3,
    },
    plotOptions = {
      "{t:'Example Title','u':'Example measurement unit','s':100,
      'x':'Example X label [ms]','y':'Example Y label [kA]'",
      'l':['Example Channel 1 legend','Example Channel 2 legend']}",
      "{t:'Example Title','u':'Example measurement unit','s':10,
      'x':'Example X label [ms]','y':'Example Y label [kA]'",
      'l':['Example Channel 1 legend','Example Channel 2 legend']}",
    },
  },
}
```

#### 4.5.2 Data structure

The data files related to an event are zipped using gzip with file extension ".gz" while metadata is in plain text files with the extension ".log". The following diagram shows how data is structured.

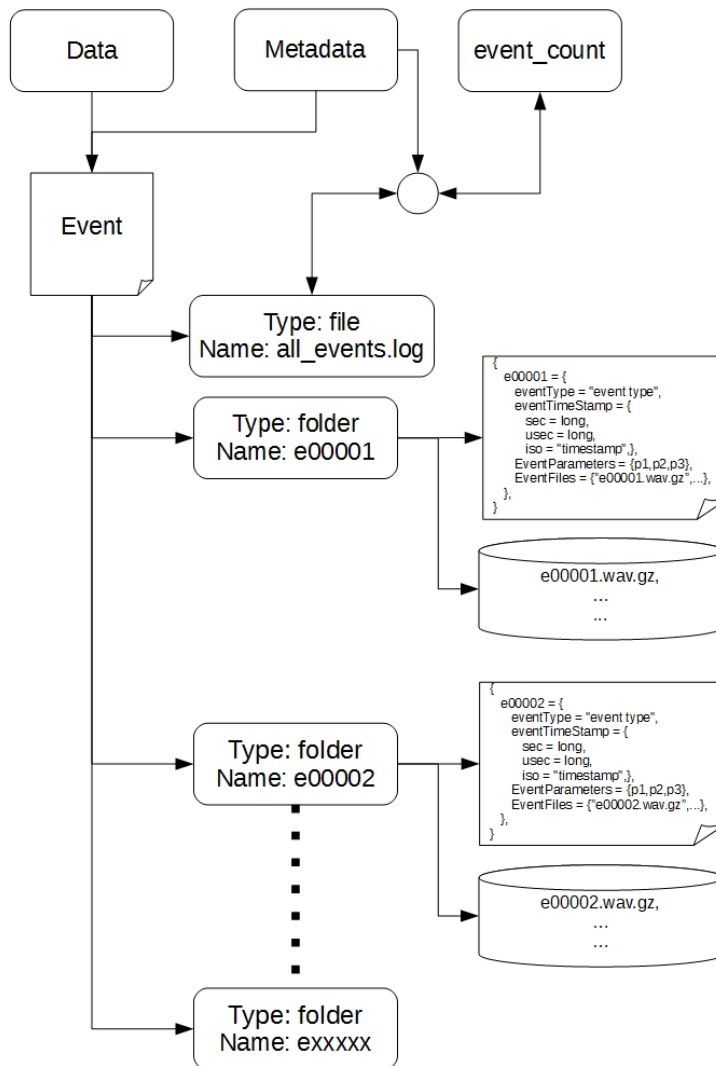


Figure 30: View of the event structure.

\*Notice that each event can have several wavefiles attached to it.

### 4.5.3 Functionality

The event table is kept in Lua memory (volatile) and can be accessed and modified by functions in the "levents" module. The table is committed to and read from a file in the flash memory (non-volatile). When an event is detected, the most recent event is updated into the event table via the metadata file related to the event. This is done in order to ensure that both files exists in non-volatile memory (ie. a file on the flash drive) if the cause of the event is a power outage that affects the sensor before a full table commit can be completed.

### 4.5.4 Generating an event (for super users)

Advanced users can make scripts that generate custom events.

To generate an event, a valid event file using the format defined above should be written into an event folder with the name "exxxxx" in a file sharing the same name e.g. "exxxxx.log" in a structure according to figure 30 and located in the directory "0:root/service/data/event". A Lua script will periodically check this folder for new

events and load new events into memory and add event metadata to the event summary file "../service/data/event/events.index" if the field "exxxxx" does not already exist.

The below example shows how to generate an event if the THD of one of the voltage channels rises above a specified level:

First a handler function is defined. This function describes what happens when the event is triggered. In our example it creates the event folder and copies the frequency spectrum for all the voltage channels to the event folder and generates an event log with the actual values of the THD for all voltage channels.

#### Listing 4: Example: data.conf with event generation (Handler function).

```
--Define what the event should do when triggered.
function voltage_thd_handler()

    -- Get the event number.
    event_number = EVENTS.event_counter + 1
    event_name = "e" .. string.format('%05d',event_number)

    -- Make the output directory
    lfs.mkdir(dirlist.flashevent.."/"..event_name)

    --Make the event table
    e = {}
    -- Fill in the event metadata
    e[event_name] =
    {
        eventType = "THD event" ,
        eventTimeStamp = {
            sec = string.format('%s',os.time()),
            usec = "000",
            iso = os.date("%Y-%m-%dT%X.000Z ",os.time()),
        },
        eventFiles = {
            "v_freq_L1.wav" ,
            "v_freq_L2.wav" ,
            "v_freq_L3.wav" ,
        },
        eventParameters = {
            phase01_thd = data.phase01.thd.voltage.value ,
            phase02_thd = data.phase02.thd.voltage.value ,
            phase03_thd = data.phase03.thd.voltage.value ,
        },
        plotOptions = { "{t':'THD of Phase01','u':'Volts','s':100,
'x':'Frequency [Hz]','y':'Volts [V]','l':['Phase01 frequency spectrum']}",
"{t':'THD of Phase02','u':'Volts','s':100,
'x':'Frequency [Hz]','y':'Volts [V]','l':['Phase02 frequency spectrum']}",
"{t':'THD of Phase03','u':'Volts','s':100,
'x':'Frequency [Hz]','y':'Volts [V]','l':['Phase03 frequency spectrum']}",
        },
    }

    -- Write the event metadata to a file
    dump(e,dirlist.flashevent.."/"..event_name.."/"..event_name..".log")

    --The copy the voltage frequency spectrums to the event folder
    cpfile(dirlist.ramfrequency.."v_freq_L1.wav",
dirlist.flashevent.."/"..event_name.."v_freq_L1.wav")
    cpfile(dirlist.ramfrequency.."v_freq_L2.wav",
dirlist.flashevent.."/"..event_name.."v_freq_L2.wav")
    cpfile(dirlist.ramfrequency.."v_freq_L3.wav",
dirlist.flashevent.."/"..event_name.."v_freq_L3.wav")

end
```



Next, set up a cyclic timer to evaluate if an event should be triggered. In this example we set up the cyclic timer to run every 10 minutes and define the threshold parameters. The action of the timer is defined in the update function.

Listing 5: Example data.conf with event generation (Timer and update function).

```
-- Definition of the cyclic timer that handles the event:
--THD_event_timer timer triggers every 10 minutes (CTT uses the time in ms as input)
THD_event_timer = CTT(10*60*1000)

-- Make the THD event settings
THD = {threshold = {voltage = 0.08} }

-- You can implement your timer in the update function. It will be called periodically
-- with an interval defined by database.update_interval defined in the
-- main configuration file.
-- You should never change the name of the update function or delete it.
-- You can always add more timers to the update function.
function update()

    -- Here we describe what the timer should do
    if THD_event_timer(true) then

        -- Check the THD level on the first three voltage and current channels
        for i=1,3 do

            -- Make a "pointer" to the channel we are analysing ( data.phase0i ).
            local phase_data = data['phase' .. string.format('%02d',i)]

            -- Check the voltage THD
            if phase_data.thd.voltage.value >= THD.threshold.voltage then
                voltage_thd_handler()
            end
        end
    end
end
```

The "THD\_event\_timer" will check the level of THD on all voltage channels every 10 minutes and generate an alarm if one of the channels gets above 0.08. What should further be considered in an actual implementation of above example is if hysteresis, or similar logic, should be included in the update function, in order to avoid repeated triggering of events every 10 minutes, in the case where the THD exceeds the threshold for a long period of time.

#### 4.5.5 Viewing event content

The metadata for all registered events are stored in the event summary file "../service/data/event/events.index" this file is automatically updated and can be viewed via the web interface.

### I3 Power sensor

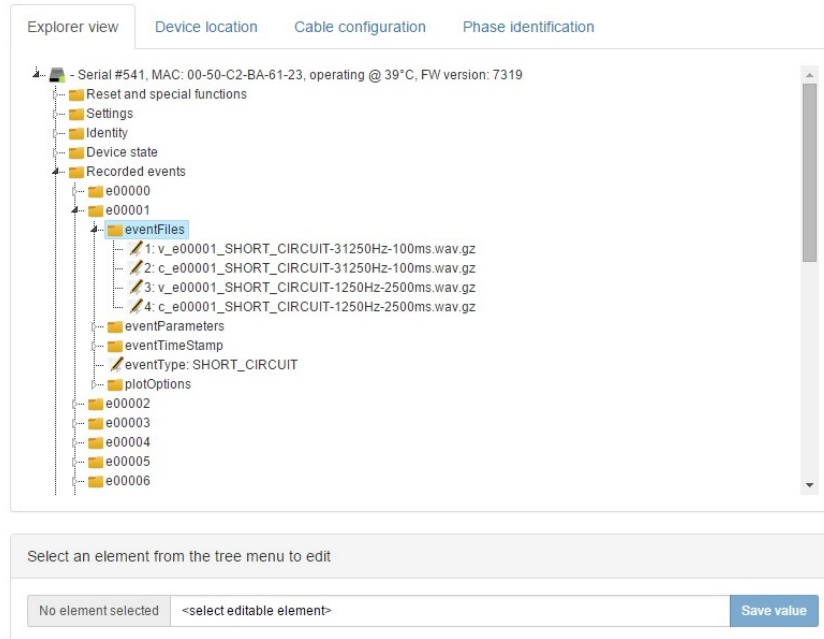


Figure 31: View of the event summary file.

When an event has files attached to it, the file content can be displayed by simply clicking on them.



Figure 32: View of one of the data files of a short circuit event, tested with a short circuit threshold of 5Ampere.

#### 4.5.6 Event types

The I3 sensor automatically detects and generates the following event types.

<b>CURRENT SHORT CIRCUIT</b>	This alarm is triggered when the current in one conductor is above the threshold defined by <b>settings/i3/event_settings/short_circuit_current_threshold</b>
<b>VOLTAGE HIGH</b>	This alarm is triggered when the voltage in one conductor is above its nominal value by the amount defined by <b>settings/i3/event_settings/high_voltage_trigger</b>
<b>VOLTAGE LOW</b>	This alarm is triggered when the voltage in one conductor is below its nominal value by the amount defined by <b>settings/i3/event_settings/low_voltage_trigger</b>
<b>VOLTAGE STAR HIGH</b>	This alarm is triggered when the voltage in the null/star point is above mean of the nominal voltage set for all other phases by the amount defined by <b>settings/i3/event_settings/null_high_voltage_trigger</b>

## 5 Web interface

### 5.1 General structure

The web interface is structured around four main components as shown in figure 33, with the following description.

- The *tab menu* for navigating between sub pages.
- The *main content area* for displaying sub page content.
- The *page context menu* for interacting with the currently selected sub page. This menu may not be available on all sub pages.
- The *notification area* for displaying messages as feedback to user actions and input. Notifications will be queued, with the newest notification appearing at the top of the notification area. Notifications will automatically disappear after 20 seconds, or can be manually dismissed.

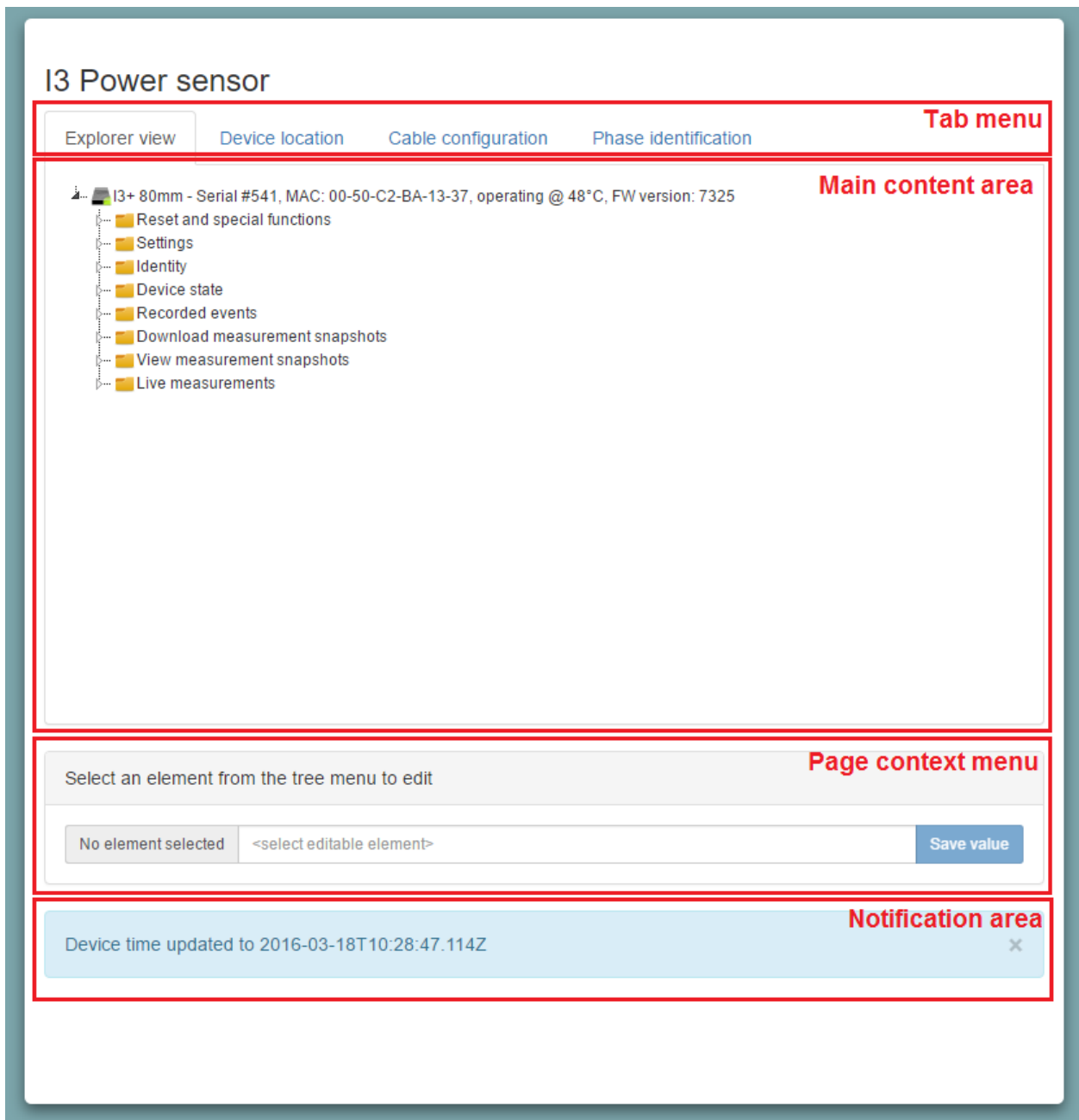


Figure 33: Overview of the web interface showing the layout of the four basic components.

## 5.2 Sub pages

### 5.2.1 Explorer view

The *Explorer view* page provides a quick and direct access to most I3 settings, functions and measurements. All items are arranged in a hierarchical folder structure that can be expanded when clicked.

When a node in the tree is clicked, the *page context menu* will show the node key and value, and if the value is editable, let you enter a new value which can be saved by clicking the *Save value* button.

Depending on the information type, the lowest level expansion will either act as a button executing the relevant command, a link for file download, a link which expands the bottom part of the main page with the requested content, or in case of specific parameter values, the value is inserted in an editable text box.

The **main folder** (top level icon) represents the I3 sensor device itself, including a few main status and identification parameters. The device text is assembled using the following parameters: The 'type' and 'serial\_number' from '*conf/identity.conf*', the MAC address using a special system call and the processor core temperature using another system call. When the mouse cursor is positioned on the folder, additional information presenting the server side time (GMT) of the web page load will occur.

The **Reset options** features a **Reset main device button**. A reset command is sent to the sensor, and the web client is set to automatically reload the page when the sensor should be up and running again. If this is not the case, please reload the page again, and remember to check that the IP address targeted is still correct after the reset.

The **Settings** information folder presents a database table as defined by the '*conf/settings.conf*' file. Any web based modification of the parameters will be directly updated in the '*.conf*' file itself. Note that some parameter changes will require a reset to take effect. This folder, along with the **Identity** and **Device state** folders, is a direct mapping of a data table in Lua (in this case 'CONFIG'). This is the reason why there are no 'space' characters, or other special characters, in the subfolder and parameter names. It provides the added benefit, that any modification in the Lua data tables will be directly reflected on the web page at the next reload.

Note that the majority of the parameters in the **Settings** folder are described by placing the cursor above the parameter in question. The same is the case for many of the parameters included in the other folders. The descriptions, as well as basic data format validation during web based parameter updates, are per default defined via the '*conf/description.csv*' file, or in the iec104 data point mapping '*.csv*' files, e.g. '*conf/iec104\_analogue.csv*'.

The **database** subfolder defines a few high level behavior settings for the main Lua data table.

The **i3** subfolder contains a number of I3 sensor specific configuration options:

- **calibration** is offering a simple linear scaling option for current and voltages. It is only expected to be needed for fine tuning of voltages. An option for a general adjustment of the voltage to current phase is also offered, but should only be used if trustworthy reference measurements have shown it to be necessary. This could be the case, if the currents are measured at the MV side of a transformer, and the voltages are from the LV side.
- **estimator** defines the high level settings for the position estimation algorithm. By clicking on the folder icon, a special view is triggered. This expands the bottom of the main page to include a simple schematic of the I3 sensor (the small version per default), including position estimates for the first 4 conductors in the model. This view can be used as a basic sanity check, to confirm that the estimator has arrived at a reasonable result. The *absolute\_threshold*, *relative\_threshold*, and the *timer* fields define the criteria for starting a new position estimation cycle. The two first acts on the estimator error (Device state  $\Rightarrow$  state  $\Rightarrow$  estimator\_error), while the last is a cyclic timer. The *estimator\_mode* defines the geometry type, that the estimator is using. Clicking on this parameter triggers a special view which presents a table with schematics of all the applicable modes. Clicking on a schematic will automatically change the model used. The results will then be visible after the following position estimation cycle.
- **event\_settings** defines trigger points for logging of measurement events.
- **phase\_identification** provides a settings list for naming and linking of the currents measured.
- **post\_processing** defines processing steps related to the RMS calculations performed on the raw measurement data. *flip\_direction\_of\_current* (default: false) can be used to reverse the sign of the current, and hereby flip the sign of the active power measurement (direction of energy flow). Note that with the default setting (false) a positive active power flow is defined as being fed from the bottom (where the 4 BNC connectors are mounted in an I3+ configuration), to the topside (the blank surface). The RMS read-out of measurement values are split up into two variants. The first, defined by *quick\_time*, is intended to provide near-instantaneous true RMS measurements for use in the web interface. The default setting is 1 second. The second, defined by *average\_time* is intended to provide true RMS averaging windows for SCADA/automated readouts, with a default setting of 60 seconds.

- **voltage\_setup** defines the high voltage capacity, in pF, used for the voltage measurement. Lua background processing translates the value into an appropriate scaling factor used internally in the sensor system.

**iec104** lists all main parameters used for the IEC 60870-5-104 protocol configuration, including file links used for address mapping and type definition of parameters.

**logging** provides parameterization options for what event or error types should be put into the system log. See also /log via FTP. In general, it should not be needed to change these settings.

**tcPIP** provides the option whether to use dhcp or not. Per default dhcp is set to true. The IP and gateway addresses and the subnet mask is also defined here (only used if dhcp=false). Note that if the device experiences severe errors during startup, it will revert to the IP configuration defined in the '*conf/panic.ip*' file, and if this also fail to load, the following hardcoded settings will be used: dhcp=false, gateway=192.168.135.100, ipv4=192.168.135.18, subnet=255.255.255.0. In DHCP mode, in the case where no DHCP server can be reached, the sensor will automatically make an allocation within the IP range 169.254.1.0 to 169.254.254.255 in compliance with RFC 3927 Section 2.1.

The parameters in the **web** section defines which port is used for the web service (default is '80'), and the file path for the description file used to link in (a) basic parameter validation via the web interface ('boolean', 'integer', 'float', 'string', 'ipv4'), and (b) description text when mouse cursor is above the parameter in question.

The **Identity** folder contains all the location and naming related information for a specific device. It should always be edited, or generated, on an individual basis. The parameters include address, city, zip, country\_state, location\_name and location\_description as identifying text for the location. The location\_name could e.g. be the name or reference for a substation, and the location\_description could detail the cable section and/or phases being measured on. Furthermore the location may be specified using WGS84 coordinates in a '<latitude>,<longitude>' decimal degrees format. The serial\_number may be set to match the Jomitek device serial number, or a custom format defined by the operator. In this context, note that the only hardcoded identifier of a device is the uniquely assigned MAC address. The type parameter will, if left blank, be automatically set during bootup based on a device internal check of the accessible peripherals. Custom naming may be used as an alternative.

The **Device state** reflects the 'data' Lua table. It contains a measurement data hierarchy starting with measurement nodes; 1 for each of the 4 phases possible to measure as main sources, and a post-processing summary on the section level (average and aggregation, where relevant, of measurement values for phases 1-3). Below each measurement node the following list of measurement parameters are reflected:

- **phase\_current** is the phase, in degrees, of the current with respect to the sensor master reference, which is typically the voltage input belonging to phase01.
- **phase\_delta** is the phase difference, in degrees, between the paired voltage and current of a given phase. A leading voltage is defined as positive.
- **phase\_voltage** is the phase, in degrees, of the voltage with respect to the sensor master reference, which is typically the voltage input belonging to phase01.
- **power\_factor** is the power factor, defined as the active power, P, divided by the apparent power, S.
- **rms\_apparent\_power** is the apparent power (current RMS multiplied with voltage RMS), in kVA.
- **rms\_current** is the RMS current, in A.
- **rms\_reactive\_power** is the reactive power, in kVAr.
- **rms\_real\_power** is the real (active) power, in kW.
- **rms\_voltage** is the RMS voltage, in kV.
- **thd\_current** is the Total Harmonic Distortion of the current.
- **thd\_voltage** is the Total Harmonic Distortion of the voltage.



Except for the THD measurements all of the above values are based on the 'CONFIG.i3.post\_processing.average\_time' RMS averaging window. The THD measurements are based on the latest frequency analysis (hardcoded to run for all active voltage and current inputs, once every 10 minutes).

For the section measurement node, phase measurements are not applicable, but otherwise the same parameters as mentioned above can be found. In addition the 'main\_harmonic\_frequency' of the master measurement channel (default is voltage of phase01).

In addition to the phase and section measurement nodes, a 'state' node displays the status for the sensor unit.

- **estimator\_error** is an error figure result from the last completed physical position estimation of the phases
- **mac\_address** may be used to provide unique identification of the sensor unit
- **resource\_alert** is 1, when either the RAM or flash dips below 1MB of free memory or the processor load is 100% for 10 consecutive cycles.
- **processor\_temperature** is the processor temperature in degrees Celcius. The temperature should remain below 75 degrees in normal operation. This measure may be used to indicate the temperature of the conductor as well, depending on the mounting conditions.

### 5.2.2 Device location

The *Device location* page visualizes the geographical sensor installation location on an interactive map. The location coordinates can be edited by clicking the map to place a new marker. Once the new mark has been placed at the desired position, the location coordinates can be saved by clicking the *Save coordinates* button in the *page context menu*.

The device location coordinates can also be set through the *Identity/wgs84\_coordinate* node in the *Explorer view* sub page.

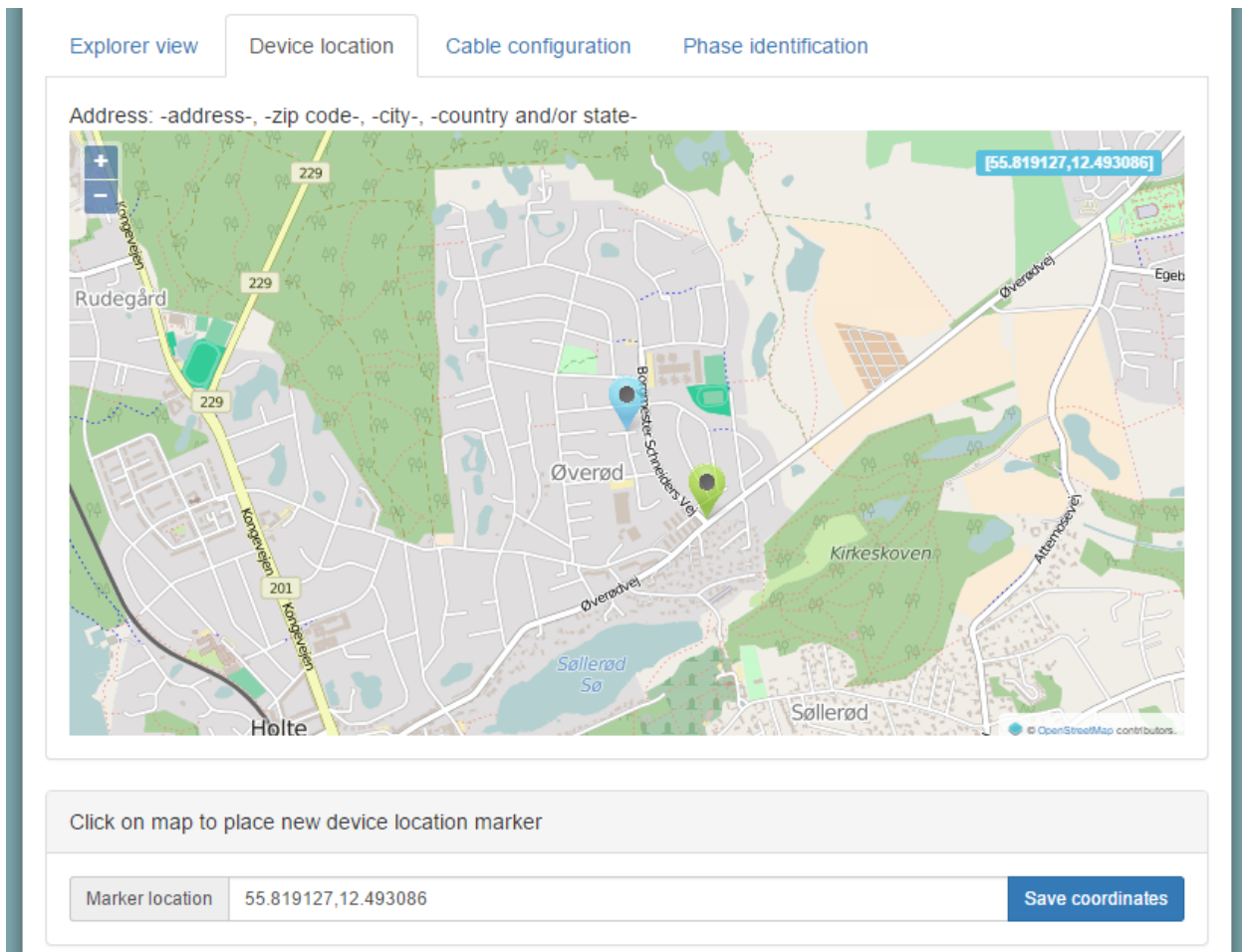


Figure 34: Screen shot from the *Device location* sub page showing the placement of a new marker location ready to be saved.

### 5.2.3 Cable configuration

The *Cable configuration* sub page provides a graphical interface for selecting the best suited cable conductor model for the I3 current estimator algorithm. To set the estimator mode, simply click the desired model and save your selection by clicking the *Save estimator model* button in the *page context menu*.

The estimator model can also be set through the *Settings/i3/estimator/estimator\_mode* node in the *Explorer view* sub page.

### 5.2.4 Phase identification

The *Phase identification* sub page provides a method for correctly pairing the detected current phase indexes with the proper current phase names when not using automatic pairing.

Either drag and drop the rows in the current table, or click the *Get suggested pairing* button to arrange the current phases. Save your pairing by clicking the *Save settings* button when done.

The voltage phase and current phase plots provided can be used as a visual sanity check reference for the phase identification process.

The phase pairing and current direction parameters can also be set through the *Settings/i3/phase\_identification/link* and *Settings/i3/post\_processing/flip\_direction\_of\_current* nodes in the *Explorer view* sub page.

## 6 IEC-60870-5-104 configuration

The IEC 60870-5-104 (abbreviated "IEC 104" or "104" in the following) application layer protocol can be used for transmitting sensor data from the sensor to a SCADA system or another data-aggregation device implementing the IEC 104 protocol. The TCP/IP protocol suite is used to ensure reliable data transfer. Additionally, the 104 protocol supports message sequencing on the application layer. A 104 client connects to the IANA assigned TCP port 2404.

The sensor allows flexible configuration of the protocol settings, and of the data points. The data points can be transmitted based on a variety of triggers, including on-change, periodic, on manual request, or using interrogation commands requesting a larger set of data. The sensor can be configured to use all features defined in the IEC 104 standard, or the sensor can use a small subset for simpler applications.

The 104 protocol settings are found in the settings.conf file under the "iec104" section. These shall match the protocol settings defined in the SCADA client, and must be in compliance with the IEC standard. Interoperability requirements can be documented by a 104 client by filling out Chapter 9 in the IEC 104 standard. An overview of the parameters defined in the sensor are listed in table 5.

Setting	Meaning	Default value
t1	Time-out of send or test APDUs.	16
t2	Time-out for acknowledges in case of no data messages, t2<t1.	11
t3	Time-out for sending test frames in case of a long idle state.	21
k	Maximum difference receive sequence number to send state variable.	8
w	Latest acknowledge after receive w l format APDUs.	12
common_address	IEC 104 address of sensor.	1
background_scan	Interval in seconds between transmission of digital data points.	1800
cyclic	Interval in seconds between transmission of analogue data points.	600
scan	Data change scan interval in milliseconds for sending event-triggered data.	500
clock_validity	Time-tagged client commands validity period in seconds.	600
redundancy	Max number of redundant connections as defined by IEC 104.	2

Table 5: IEC-104 parameters.

Three .csv files are used to map internal sensor measurements to data points accessible by an IEC 104 client. A macro enabled Excel file is used to generate these .csv files, and all changes to the data mapping should be applied in the Excel file. There are three sheets in the Excel file, and each generate one .csv file. The sheets are called "iec104\_analogue", "iec104\_digital" and "iec104\_commands", which define analogue and digital measurement mapping to the IEC 104 protocol. An example snapshot of the "iec104\_analogue" sheet is shown in figure 35.

	A	B	C	D	E	F	G	H	I	
1	address	type	data	interrogation	spontaneous	cyclic	absmin	absmax	rel	desc
2	1000	M_ME_TF_1	section.rms_current	global	TRUE	TRUE	10	100	10	RMS current [A]
3	1001	M_ME_TF_1	section.rms_voltage	global	TRUE	TRUE	10	100	10	RMS voltage [kV]
4	1002	M_ME_TF_1	section.power_factor	global	TRUE	TRUE	10	100	10	Power factor [-1;1]
5	1003	M_ME_TF_1	section.rms_apparent_power	global	TRUE	TRUE	10	100	10	Apparent power  S  [kVA]
6	1004	M_ME_TF_1	section.rms_reactive_power	global	TRUE	TRUE	10	100	10	Reactive power Q [kVar]
7	1005	M_ME_TF_1	section.rms_real_power	global	TRUE	TRUE	10	100	10	Real power P [kW]

Figure 35: iec104\_analogue.

Description of columns:

- **address** The address of the data point, [1-65535].
- **type** The IEC 104 type (see interoperability list in IEC 60870-5-104 chapter 9).
- **data** Internal sensor variable name.

- **interrogation** The interrogation group this data point belongs to. Can be "global" or "groupX" where X is in the range [1-16].
- **spontaneous** Set to true if data point should be transmitted on change.
- **cyclic** Set to true if data point should be sent periodically.
- **absmin** Minimum absolute change in data value to trigger spontaneous transmission. This is useful for avoiding sending a data point too often when the value is low.
- **absmax** Maximum absolute change in data value to trigger spontaneous transmission.
- **rel** Relative change in percent of the data value to trigger spontaneous transmission. For example, setting  $rel = 10$  for a data point with value 500 triggers transmission at 450 or 550.
- **desc** A short description of the data point and unit. The description will be used as a tooltip in the web interface.

Refer to figure 36 for a graphical example of the transmission thresholds **absmin**, **absmax** and **rel**.

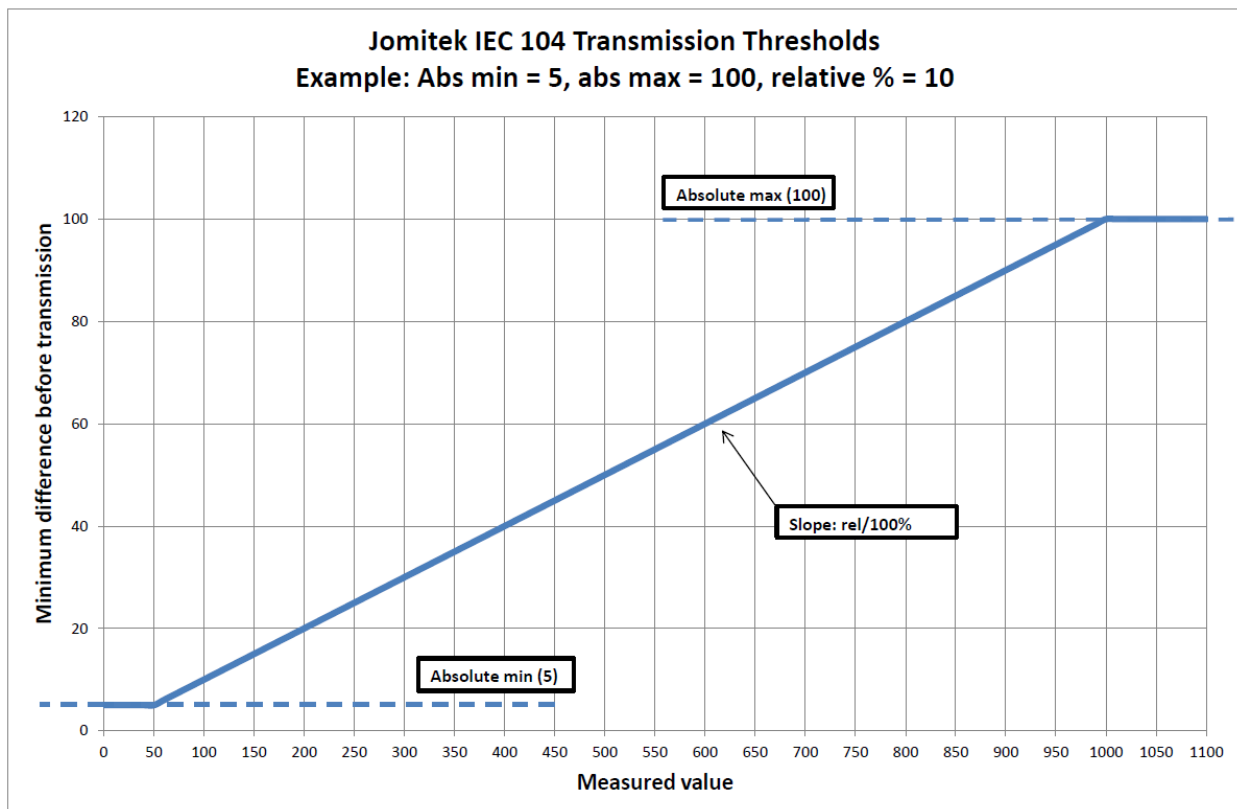


Figure 36: **absmin**, **absmax** and **rel**.

Any change in a digital value triggers transmission given **spontaneous** is set true for this data point in the "iec104\_digital" sheet. For this reason the "iec104\_digital" sheet does not define **absmin**, **absmax** and **rel** columns. Also, the "iec104\_digital" sheet replaces the "cyclic" column found in the analogue sheet with "background". The use of the "cyclic" column for analogue values is similar to the use of "background" column because they are both used to trigger transmission at fixed intervals.

Additionally, all IEC 104 commands are supported by the sensor, and they can be used to invoke any function or define any setting in the sensor as desired. However, depending on the application and due to the variety of protocols supported by the sensor, IEC 104 commands may not be needed. The "iec104\_commands" sheet in

the Excel file map a data point address to an internal variable.

Since the IEC 104 protocol uses TCP/IP, it is in some cases useful to debug the communication stream using Wireshark<sup>2</sup>.

## 7 Signal analysis

### 7.1 WAV files

All time series data files recorded on the I3 sensor are stored as WAV files (file extension .wav). The WAV file format is a lossless and uncompressed format that holds time series meta data such as sampling frequency and sample bit depth as well as multiple time series measurements stored as separate channels in a single file.

Additional meta data describing the type of measurement (voltage, current, etc.), scaling, time stamp and similar is contained in the event meta data for events.

The format structure itself is simple and widely compatible with many software tools. Further information on the format can be found at <https://en.wikipedia.org/wiki/WAV>.

### 7.2 Frequency analysis

The I3 sensor has a build-in spectrum analyser that offers a per phase high resolution FFT of both voltages and currents. An FFT is a valuable tool in diagnostics when assessing the health state of a transformer or when looking for harmonic distortion and noise.

#### 7.2.1 Fast Fourier Transform

An FFT (Fast Fourier Transform) computes the DFT (Discrete Fourier Transform) and produces the same result as evaluating the DFT directly but uses significantly less computations for large size sample sets. For the FFT application in the I3 sensor, the reduction in processing time is a factor of 388.

Let  $x_0, \dots, x_{N-1}$  be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad k = 0, \dots, N-1, \quad n \in \mathbb{Z}$$

Calculating the DFT directly requires  $O(N^2)$  operations whereas an FFT algorithm achieves the same result in  $O(N \log N)$  operations.

The result of the transform is a complex spectrum representation from which the magnitude and phase of each of the frequency components of the input signal is calculated. The I3 sensor presents the user to the magnitude of the transformed signal divided into 1562 bins each covering a span of 10Hz, giving a total span of 15620Hz, and starting from 10Hz. The I3 does not measure DC signals, so the DC component is omitted in the output. The THD described in the next section is calculated based on the magnitude of the FFT bins.

#### 7.2.2 Total Harmonic Distortion

One of the basic measures of power quality is the THD (Total Harmonic Distortion), which is used to describe the ratio of (typically unwanted) energy in a signal outside of the main frequency component. The Jomitek I3 sensor provides three different measures of the THD, their definitions are given below.

---

<sup>2</sup><https://www.wireshark.org/>

**THD** according to EN50160 measures the harmonic distortion up to the 40<sup>th</sup> harmonic or up to 15620Hz (whichever is the lowest) according to the following equation

$$THD = \frac{\sqrt{\sum_{n=2}^{40} X_n^2}}{X_1}$$

where  $X_n$  is the magnitude of the  $n$ th harmonic and  $X_1$  is the magnitude of the main harmonic frequency, which is automatically determined.

**THD Extended** measures the harmonic distortion up to the maximum harmonic available according to the following equation

$$THD_{Extended} = \frac{\sqrt{\sum_{n=2}^{H_{max}} X_n^2}}{X_1}$$

where  $X_n$  is the magnitude of the  $n$ th harmonic and  $X_1$  is the magnitude of the main harmonic frequency.  $H_{max}$  depends on the main harmonic frequency and is determined with the following equation

$$H_{max} = \lfloor \frac{15620}{f_{main}} \rfloor$$

for a main harmonic frequency  $f_{main} = 50\text{Hz}$  the equation evaluates to

$$H_{max} = \lfloor \frac{15620\text{Hz}}{50\text{Hz}} \rfloor = \lfloor 312.4 \rfloor = 312$$

**THD+N** measures the harmonic distortion between the amplitude of the main harmonic frequency and all the other frequency bins produced by the FFT according to the following equation

$$THD_{+N} = \frac{\sqrt{(\sum_{n=1}^{N_{bin-max}} X_n^2) - X_m^2}}{X_m}$$

where  $X_n$  is the magnitude of the  $n$ 'th frequency bin and  $X_m$  is the magnitude of the main harmonic frequency.

### 7.2.3 FFT architecture

The I3 Power Sensor offer two processing variants of the FFT of the signals measured by the sensor. These modes are called **SINGLE** and **CONTINUOUS** each of which are explained below.

#### Single Shot mode

In single shot mode, called **SINGLE** in the I3 settings menu, the FFT of a signal is a spectrum of a time signal with length 100ms. The spectrum is produced simultaneously for all active voltage and current channels which allow comparison of the frequency spectrum of the same period in time between the individual channels. In this mode the sensor ignores the settings in **settings** → **I3** → **post\_processing** → **fft** → **voltage\_channels** & **current\_channels**

#### Continuous mode

In continuous mode, called **CONTINUOUS** in the I3 settings menu, the FFT spectrum is an average of a series of short-time Fourier transforms(STFT) each with a length of 100ms. The average is calculated over a user defined interval using incoherent FFT averaging.

Note that for continuous mode **ONLY ONE** channel is processed at a time. This means that an averaging length of 10 seconds will produce one updated FFT every 10 seconds for a particular channel. If for example the 4 voltage channels on an I3+ is selected, it would take 40 seconds to generate an updated FFT for all channels with 10 seconds allocated for each channel. When the last FFT is generated the FFT processing restarts with the first channel in the defined sequence, in an infinite loop.



### FFT averaging

When the I3 sensor operates in continuous mode, the resulting FFT is based on an average of multiple short-time Fourier transforms(STFT). The resulting FFT can be described by the following equation

$$F_{average}(M) = \frac{1}{K} \sum_{k=1}^K |F_k(M)|$$

where the  $F_{average}(M)$  is an average of the M'th frequency bin of K STFT's. The process is illustrated in figure 37.

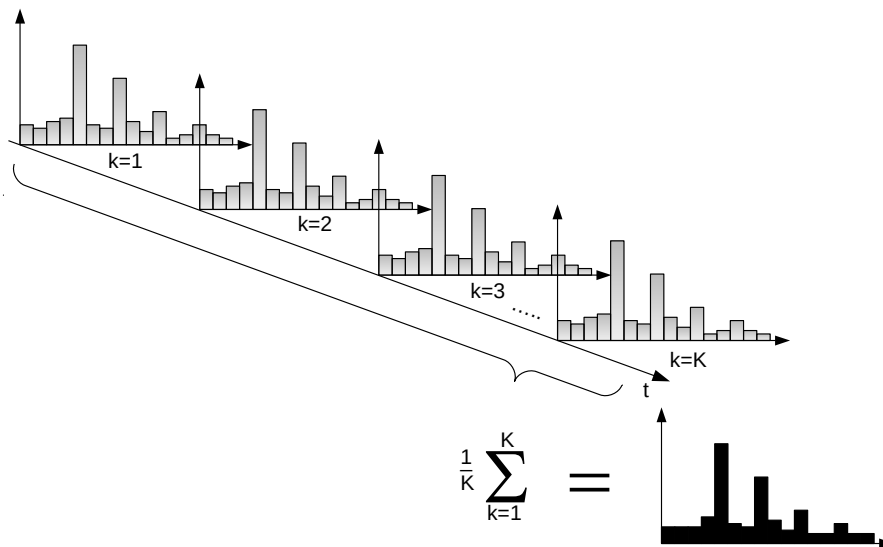


Figure 37: Illustration of the incoherent average of the FFT's.

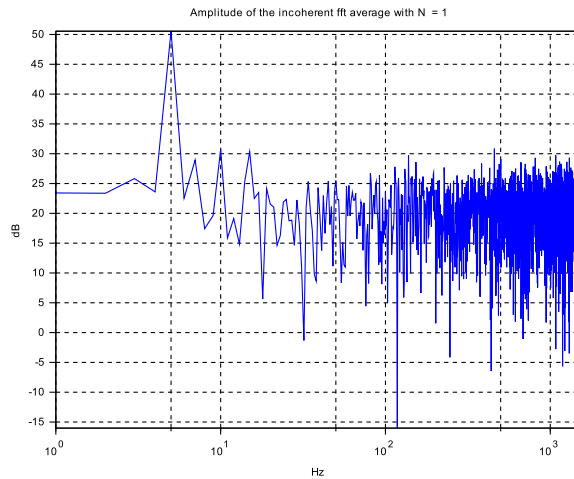
The process of averaging helps improve the signal to noise ratio(SNR) of the individual FFT bin. The SNR is improved according to the following equation.

$$SNR_{incoh}gain(dB) = 10 \log_{10}(\sqrt{N})$$

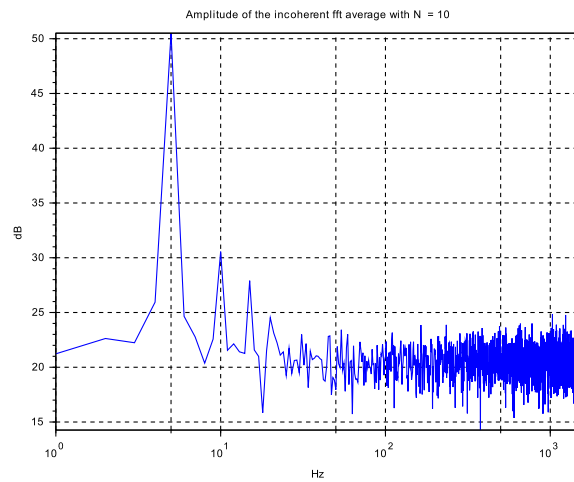
The simulated incoherent FFT average on figure 38 contains the following graphs.

- Figure 38a shows an FFT of a 100ms simulation of a 50Hz sine wave with 3 harmonic frequencies and added zero mean Gaussian white noise.
- Figure 38b shows an incoherent FFT average over 10s of the same signal as in figure 38a.
- Figure 38c shows an incoherent FFT average over 60s of the same signal as in figure 38a.

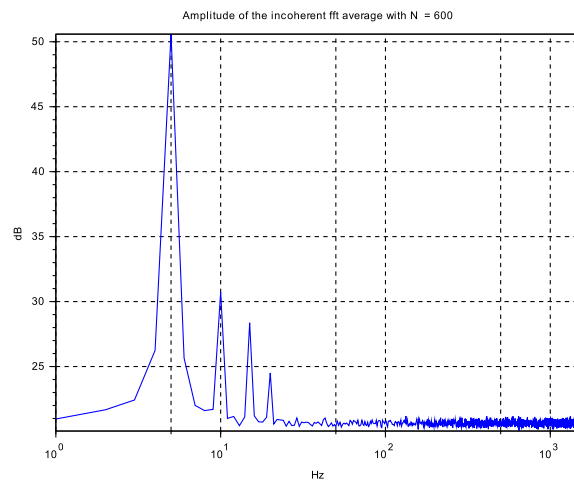
It can easily be seen from figure 38 how the fluctuation of the added random noise is reduced when the average time is increased.



(a) Simulated spectrum without average



(b) Simulated spectrum averaged over 1 second



(c) Simulated spectrum averaged over 60 seconds

Figure 38: Illustration of the signal to noise ratio for different averaging intervals.

## 7.2.4 FFT settings

This section describes how to set up the FFT processing using the I3 sensor web interface. Under the settings menu **settings** → **I3** → **post\_processing** → **fft**, see figure 39, the following options can be found.

- **mode** "CONTINUOUS" or "SINGLE"
- **update\_interval** 10s to 3600s
- **log\_fft** true or false
- **voltage\_channels** – array with index 1-4, true or false
- **current\_channels** – array with index 1-5, true or false

### In single shot mode

If **settings** → **I3** → **post\_processing** → **fft** → **mode** is set to **SINGLE** the only further configuration need is to adjust **update\_interval** and **log\_fft**. The other FFT settings are not used in this mode.

- **update\_interval** sets the time interval with which the FFT's and THD's of all active channels are recalculated simultaneously, based on a 100ms snapshot.
- **log\_fft** activates logging of FFT data to the FTP accessible folder "/data/fft".

### In continuous mode

If **settings** → **I3** → **post\_processing** → **fft** → **mode** is set to **CONTINUOUS** all of the FFT settings are relevant to consider adjusting to the need.

- **update\_interval** sets the averaging time interval for the FFT's and THD's of each of each channel.
- **log\_fft** activates logging of FFT data to the FTP accessible folder "/data/fft".
- **voltage\_channels** Toggles the voltage channels, *active = true*
- **current\_channels** Toggles the current channels, *active = true*

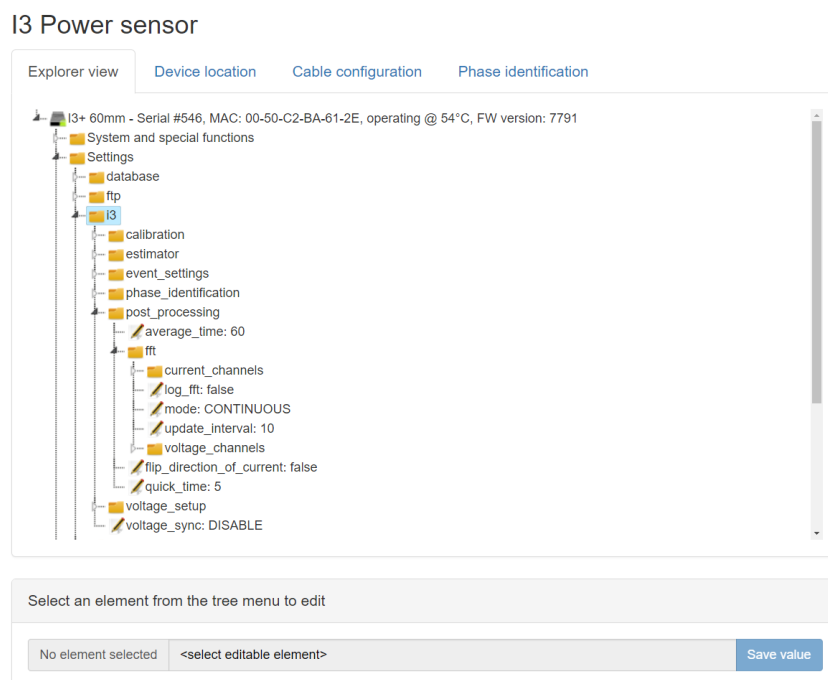


Figure 39: Overview of the FFT settings found in the web interface.

### 7.2.5 Viewing an FFT

To view the most recent FFT, go to the "Explorer view" tab in the I3 web-interface, then find and click the "View measurement snapshots" menu and select "Frequency analysis". You will then see a view similar to figure 40.

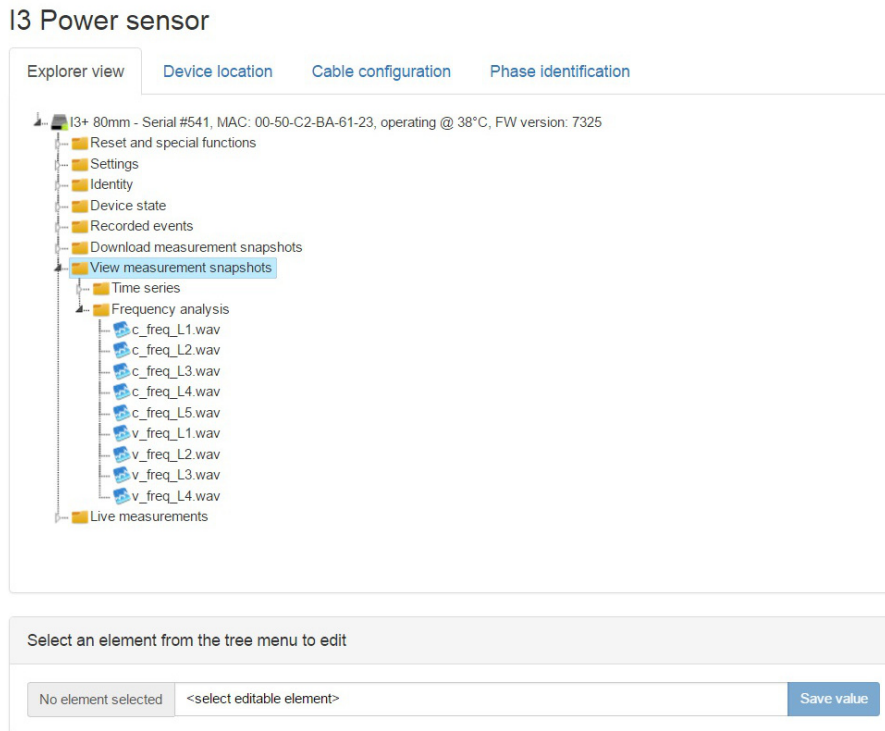


Figure 40: "View measurement snapshots" menu.

To inspect the content of an FFT, click on the wav file. "c\_" is an FFT of a current channel, "v\_" is an FFT of a voltage channel, "\_L1" tells you that it is channel 1.

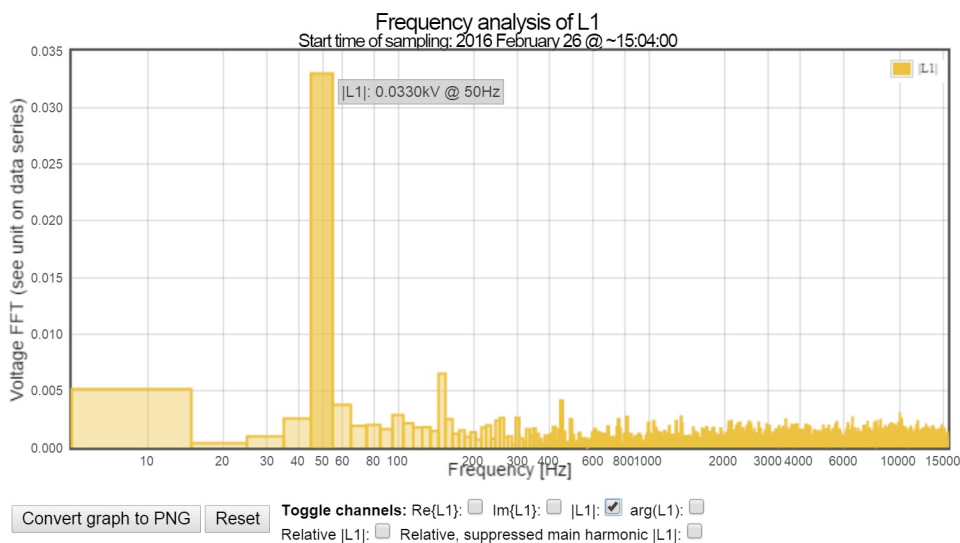


Figure 41: Example of an FFT of a live 50Hz voltage.

## 7.2.6 Downloading an FFT

To download a .wav file containing the most recent FFT, go to the "Explorer view" tab in the web-interface, click the "Download measurement snapshots" menu and select "Frequency analysis". See figure 42.

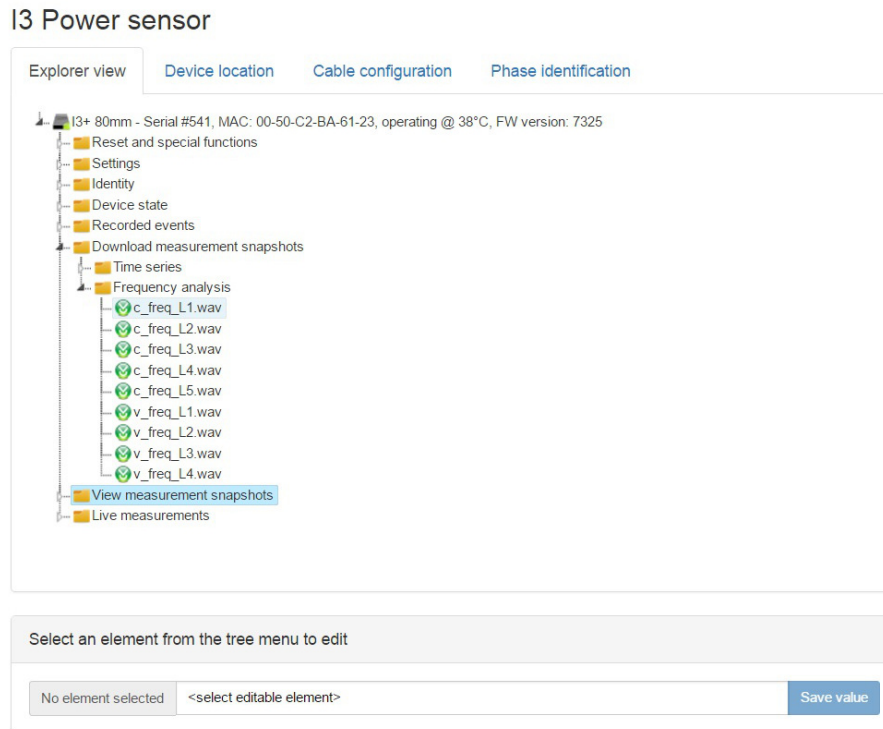


Figure 42: Example of the FFT file download menu.

## 8 Firmware and software updates

### 8.1 Boot loading sequence

During bootup the I3 checks the `0:/root/service/software_update` folder for the presence of a **software update file** or **firmware image file**.

- A **software update file** (`software.tar.gz`) contains a folder structure image that, when unpacked, overwrites the in-place file-system. The unpacked file-system may include a **firmware image file** in the software update folder.
- A **firmware image file** (`image.bin`) is a binary file which contains the most basic low-level software needed to run the I3 sensor.

If a **software update file** is found, it is unpacked and its folder structure contents copied to the `0:/root` folder, thus updating the software/file system on the sensor.

If a **firmware image file** is found, the boot loader will search for the presence of a **checksum file** (`checksum.txt`) file which is used to verify the integrity of the firmware image. If the calculated checksums match, the firmware is decrypted and its binary instructions copied to the microprocessor which will reboot upon completion, thus finalizing the firmware update process.

After boot loading completes, the system will proceed to initiate further low-level processes after which the Lua scripting engine and web server will be started. At this point the system is in a fully operational mode.

Depending on the size of the upgrade it will normally take 2-5 minutes, with the high end of this interval being relevant when both a firmware and software upgrade is included in a `software.tar.gz` file.

## 8.2 Firmware upgrade

The software or firmware on the I3 sensor can be updated in four simple steps:

1. Download the newest update file from [http://jomitek.dk/downloads/device\\_software\\_update](http://jomitek.dk/downloads/device_software_update). The page is password protected, use username: `jomitek` and password: `jomitek`.
2. Connect to the I3 sensor through the FTP interface with username: `service` and password: `service1234`.
3. Transfer the update file, `software.tar.gz`, to the `/service/software_update` directory.
4. Reboot the I3 sensor through the web interface and wait for the boot loading sequence to automatically apply the update.

The I3 sensor will now restart and apply the software update.

## 9 Advanced user interfaces

### 9.1 Telnet interface

For advanced maintenance and debugging tasks, the I3 sensor can be accessed and configured via a limited command line interface using Telnet. This interface is accessed by connecting to the sensor IP address on port 23 using a standard Telnet client. It is recommended to use PuTTY<sup>3</sup> which is open source and free to use.

Be sure to enable the 'Implicit CR in every LF' and 'Implicit LF in every CR' options when configuring PuTTY.

A list of applicable Telnet commands for the I3 sensor is listed by typing 'help' in a Telnet terminal, when connected to a sensor.

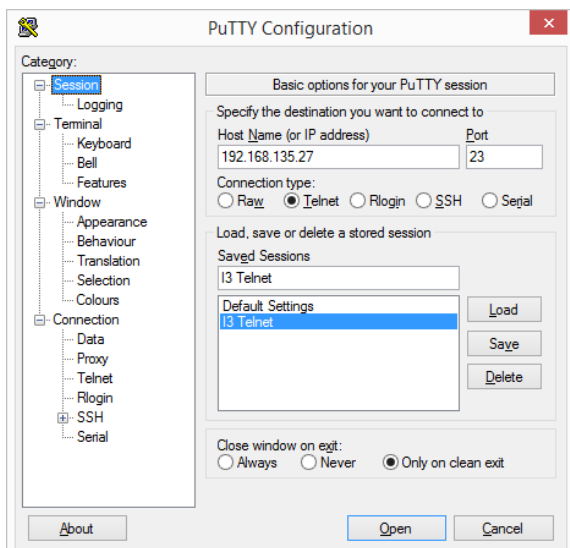


Figure 43: Putty Telnet Session configuration.

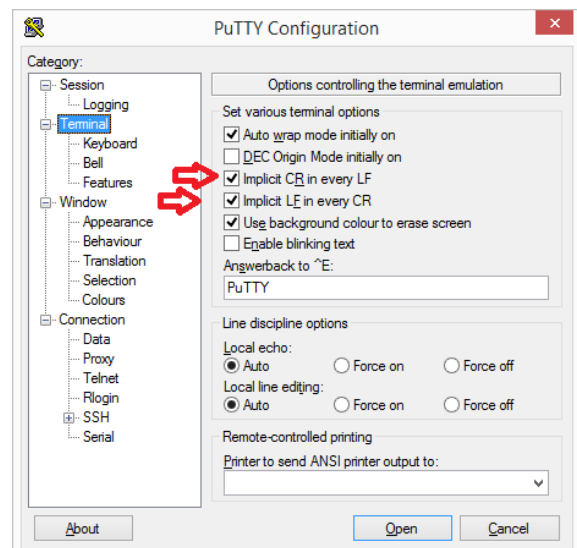


Figure 44: Putty Telnet Terminal configuration.

<sup>3</sup><http://www.putty.org/>





Figure 45: The Telnet command line interface.

## 9.2 FTP interface

The internal memory on the I3 sensor can be accessed via the FTP protocol on port 21. Use an FTP client program (e.g. FileZilla<sup>4</sup>) and sign in with default username: *service* and password: *service1234*.

When connected you will have access to three folders.

- The `conf` folder where configuration files and scripts are stored.
- The `software_update` folder where software update files can be uploaded.
- The `log` folder where various log files are placed by the sensor system.

<sup>4</sup><https://filezilla-project.org/>

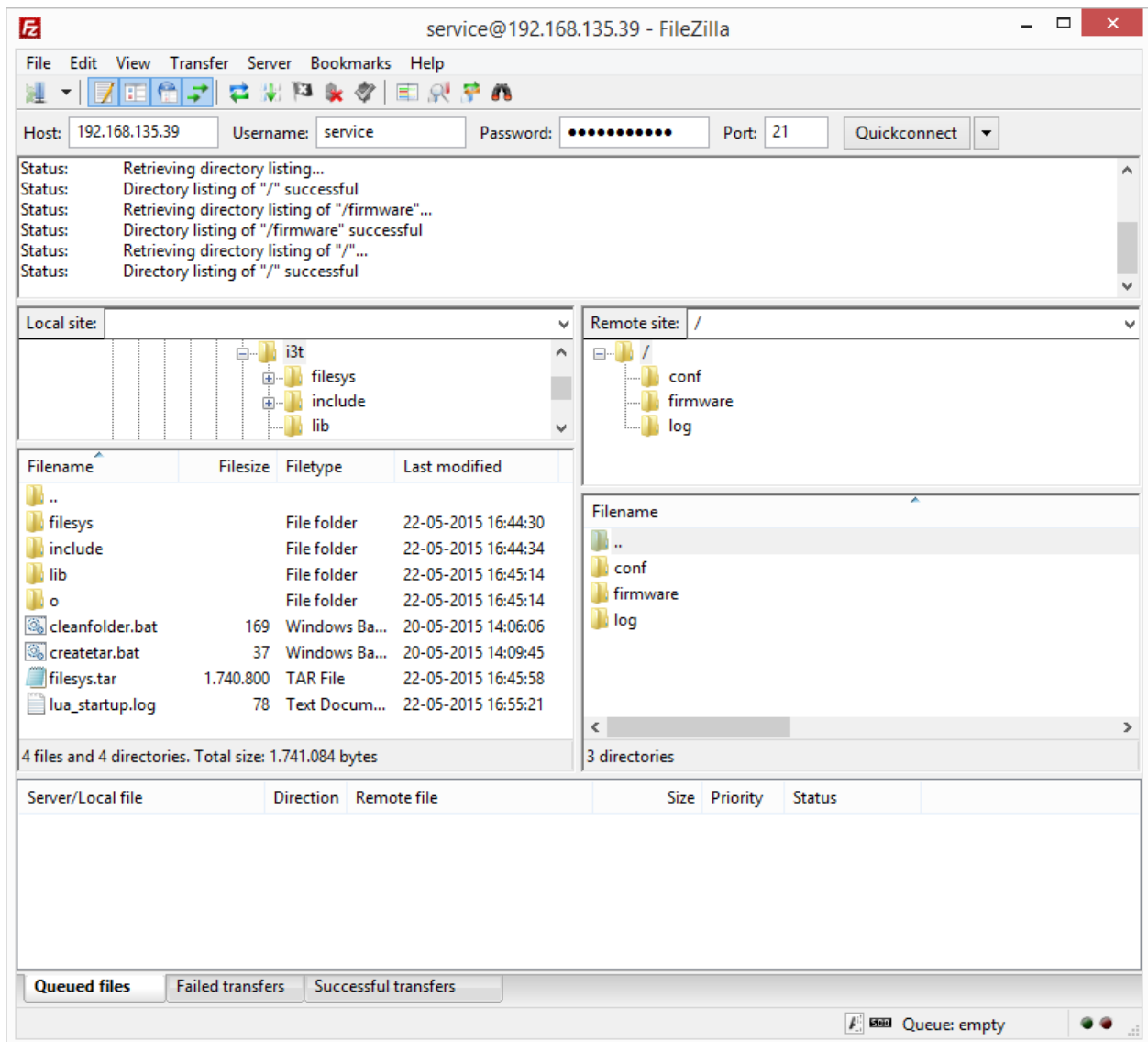


Figure 46: FTP connection to the I3 sensor shown with the FileZilla FTP client.

## 10 Advanced mass reconfiguration and service

### 10.1 Mass reconfiguration

The `'conf/identity.conf'` and `'conf/settings.conf'` files play a central role in the configuration and customization of a specific I3 sensor. To enable efficient mass reconfiguration of I3 sensors, it is vital to be able to update information for a select group of the sensor base, or update a specific setting across all sensors without the need to retrieve, edit, and upload configuration files on a case by case basis.

These needs can be efficiently addressed using the Jomitek Mass Reconf Tool and underlying functionalities. In addition to the above mentioned `.conf` files, the I3 sensor continually checks for updates to these files in the form of a `.reconf` variant (`'identity.reconf'` and `'settings.reconf'`), which allows incremental update of specific parameters in the `.conf` files, without affecting the remaining parameters.

The end user may choose to use their own tools to generate Lua formatted `'reconf'` files for large scale sensor management, or as an alternative use the functionality provided in the Macro enabled Excel sheets, which can be downloaded from <http://jomitek.dk/downloads/tools>.

Please read the User Manual-sheet in the Excel document carefully, and note the need for the WinSCP<sup>5</sup> tool when distributing '.reconf' file updates automatically.

## 10.2 Operational overview and service

Using the 'Jomitek Device Locator'<sup>6</sup>, an overview of all the I3 sensors within a given subnet can be created, using the 'Overview' tab. Using a UDP broadcast, the tool force all sensors to reply with a few basic pieces of information, such as the MAC address of the device, as well as the physical configuration, serial number and firmware version.

Note, depending on the network architecture used, that large subnets may result in failed discovery of part of the sensor base, due to data packages being discarded. Please contact Jomitek with information on the number of sensors, and network architecture used, should this be experienced. Then assistance will be provided.

The Jomitek Device Location incorporates a few simple management tools for the individual sensor in the form of the 'Web' tab and the 'Communication' tab. The 'Web' tab automatically loads the web page of a sensor selected (double-clicked) in the 'Overview' tab. The 'Communication' tab features a split screen command line interface to the selected sensor, where Telnet or Lua terminal commands may be entered.

It is a development ambition to expand the 'Overview' tab with a number of additional features and device information. The current development concepts are:

- Simple .csv-based database of all devices that should be available in the network. This database will populate the 'Overview' tab with sensors which have not yet been discovered, marking them as offline. The .csv based database will enable a simple import option into any database or management tool used by the operator.
- Sensor alarms and events.
- Operational state, e.g. total active power, THD, and power factor.
- Grouping of sensors, e.g. for geographical filtering

Prioritization of these management elements will rely on customer feedback. Please contact Jomitek regarding any such development requests.

---

<sup>5</sup><http://winscp.net/>

<sup>6</sup><http://jomitek.dk/downloads/tools>