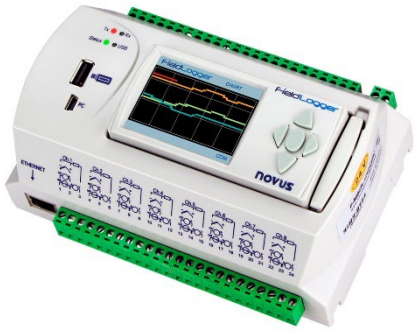
novus

We Measure, We Control, We Record

FieldLogger



USER GUIDE V1.9x D

⚠	**Compatible with devices with firmware version V1.9x or higher.**

CE

1 INTRODUCTION

The **FieldLogger** is a high-resolution and high-speed data acquisition and logging device for analog and digital signals. The result of an advanced technological development, the device is distinguished in diverse aspects, such as high-performance, high-connectivity and ease of configuration and operation. This technology is presented as the ideal solution for applications that require flexibility and functionality for diverse communication networks.

Its main features include:

- 8 analog inputs.
- 8 digital inputs and outputs.
- 2 relay outputs.
- 2 MB internal memory.
- SD Card interface[1]. Supports SD cards up to 32 GB with FAT32 format.
- RS485 interface:
  - Main:
    - Modbus RTU slave
    - Modbus RTU master[2]
  - Auxiliary:
    - Modbus RTU slave (DB9 connector)[3]
- Ethernet services[4]:
  - DHCP
  - HTTP (web page)
  - FTP (client and server)
  - SMTP (emails sending)
  - SNMP
  - Modbus TCP
  - MQTT
- USB:
  - Host
  - Device
- HMI (human-machine interface – optional)

There are 4 input channel types in **FieldLogger**: analog, digital, remote, and virtual. Analog and digital channels are those acquired directly by **FieldLogger** through their respective inputs. The remote channels are those acquired through Modbus RTU protocol, operating as master on its RS485 interface. Virtual channels are a special type of input channel where mathematical operations can be carried out, allowing the calculation of complex formulas from the measured information.

The analog input channels are configurable for reading voltage, current, thermocouples, Pt100 and Pt1000 signals. These inputs count on the precision of a 24-bit A/D converter with high acquisition speed, which can reach 1000 samples per second. The digital channels can be configured individually as inputs or outputs.

The Ethernet interface allows data download and data access of the inputs and outputs, through services that can be individually enabled and configured. Through a web browser (HTTP), one can visualize the data of the enabled channel, diagnostics, and general information of the **FieldLogger**. FTP client can be used for logging data downloads. **FieldLogger** can identify up to 32 distinct alarm conditions, allowing the triggering of outputs, sending emails or SNMP traps whenever an alarm condition is detected. All the information relative to the variables, status, and diagnostics of the **FieldLogger** are available in Modbus registers that can be accessed through the Modbus TCP interface or the Modbus RTU interface available either through USB interface (device) or RS485 (when operating as slave).

The USB Interface device is used for connection to a computer for configuration, monitoring or download. Whereas the USB interface host is used for connection of a USB flash drive, for data downloading from the logging memory.

The data logging memory can be transferred by any of the interfaces to the **NXperience** software, which allows the exportation of the most diverse data formats.

---

[1] Not available in some models.

[2] Available only for models with Ethernet interface.

[3] Not available in some models.

[4] Not available in some models.

2 CONNECTIONS AND INSTALLATION

2.1 MECHANICAL INSTALLATION

FieldLogger chassis can be installed on a 35 mm DIN rail.

For the installation on the rail, you must pull out the two clamps located right below the connections of the channels, taking care not to remove them, as per the figure below.

If necessary, the connector may be removed for installation of DIN rail.

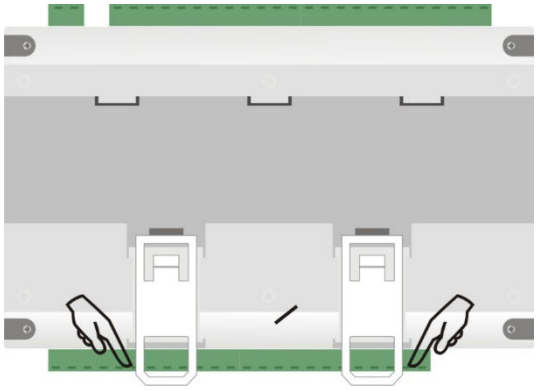


Figure 1

Afterwards, fit the **FieldLogger** to the rail:

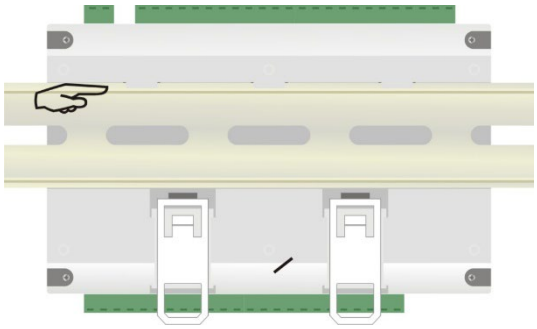


Figure 2

And finally, push the 2 clamps until you hear a double click for the complete installation:

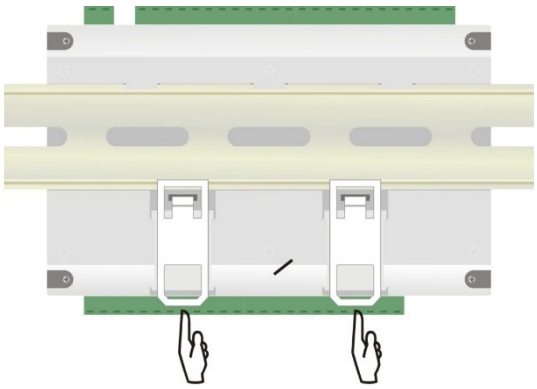


Figure 3

2.1.1 DIMENSIONS

WITHOUT HMI MODULE

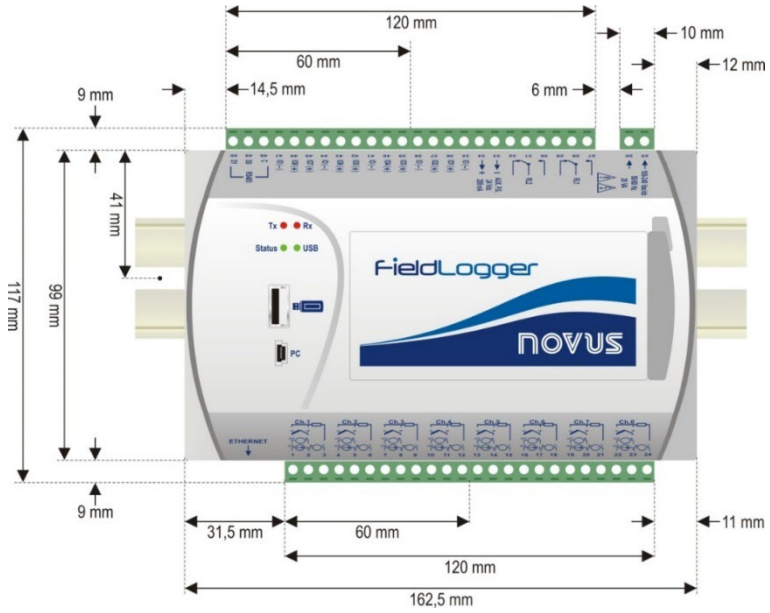


Figure 4

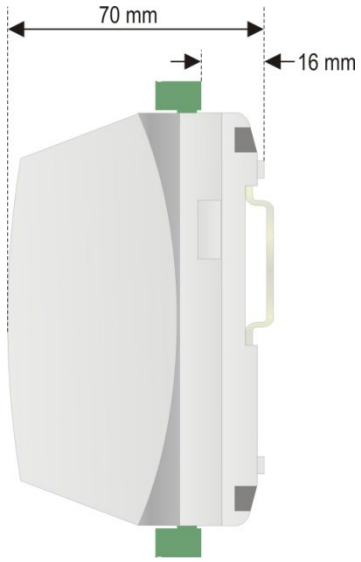


Figure 5

WITH HMI MODULE (OPTIONAL)

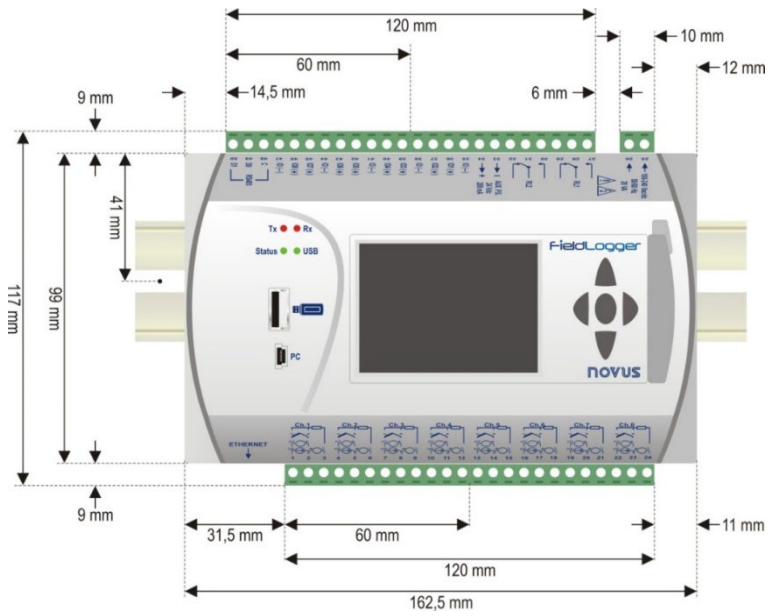


Figure 6

Figure 7

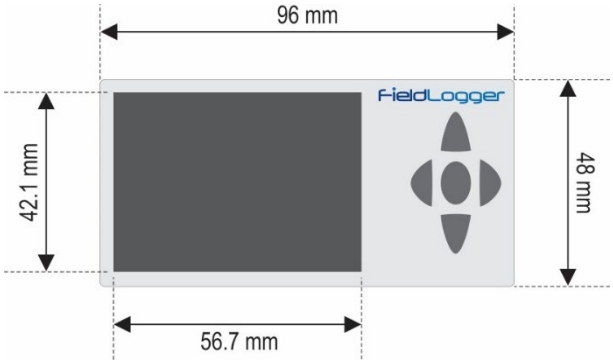
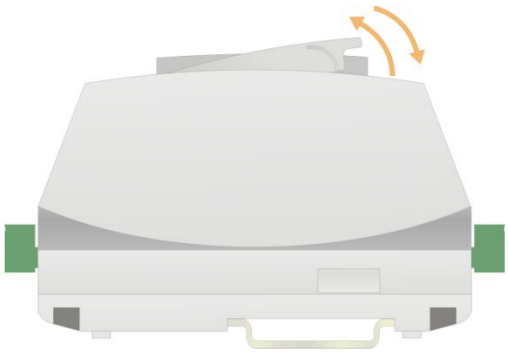


Figure 8

2.1.2 ATTACHING AND DETACHING THE FRONT COVER

When detaching the front cover, pull the lever at the right side of the **FieldLogger** up to the end.	When attaching it, insert the cover (left side first) and press its right side gently. After that, press the lever back into the chassis.
 Figure 9	 Figure 10

2.1.3 ATTACHING AND DETACHING THE HMI

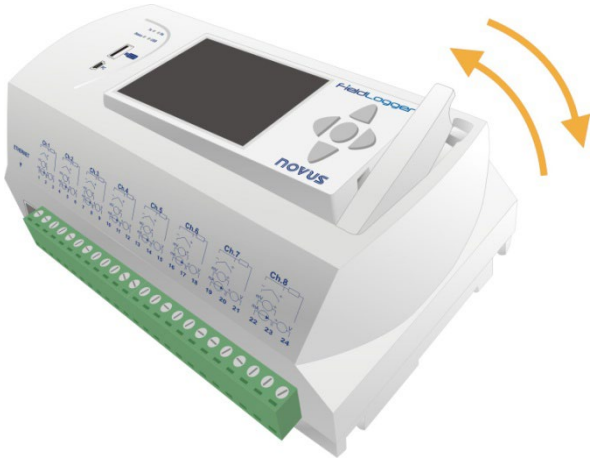
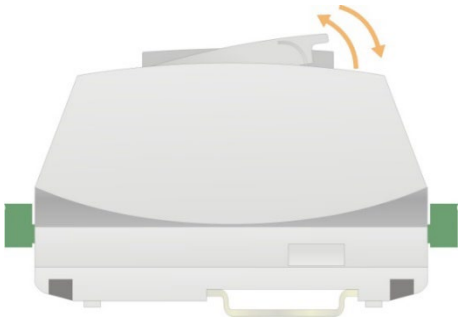
When detaching the HMI, pull up the lever at the right side of the **FieldLogger** up to the end.	When attaching it, insert the HMI (left side first) and press its right side until DB9 connectors are firmly connected. After that, press the lever down into the chassis.
 Figure 11	 Figure 12

Figure 13

2.2 ELECTRICAL CONNECTIONS

2.2.1 SAFETY SUMMARY

The symbols below are used on the device and throughout this document to draw the user's attention to important operational and safety information.

			
CAUTION or WARNING: Read complete instructions prior to installation and operation of the unit.	**CAUTION or WARNING:** Electrical Shock Hazard.	**DOUBLE INSULATION:** The **FieldLogger** power supply is double insulated, represented by the above symbol printed on the device's connection label.	**POWER INPUT:** **FieldLogger** may be powered from either AC or DC power source.

All safety-related instructions that appear in the manual must be observed to ensure personal safety and to prevent damage to either the instrument or the system. If the instrument is used in a manner not specified by the manufacturer, the protection provided by the device may be impaired.

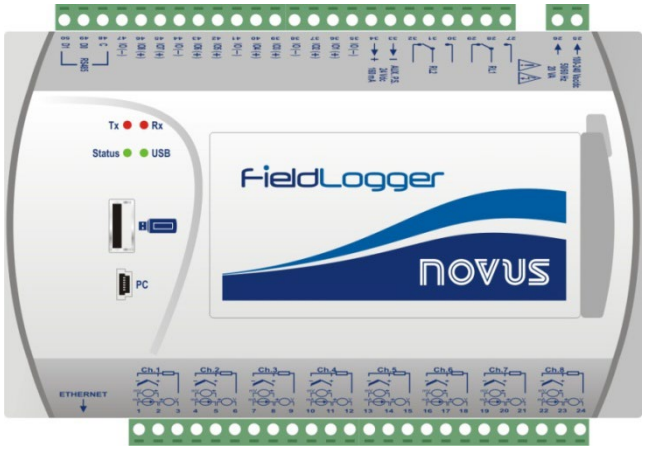


Figure 14

FieldLogger has 2 rows of terminals for diverse connections. This information is identified in the box of **FieldLogger**:

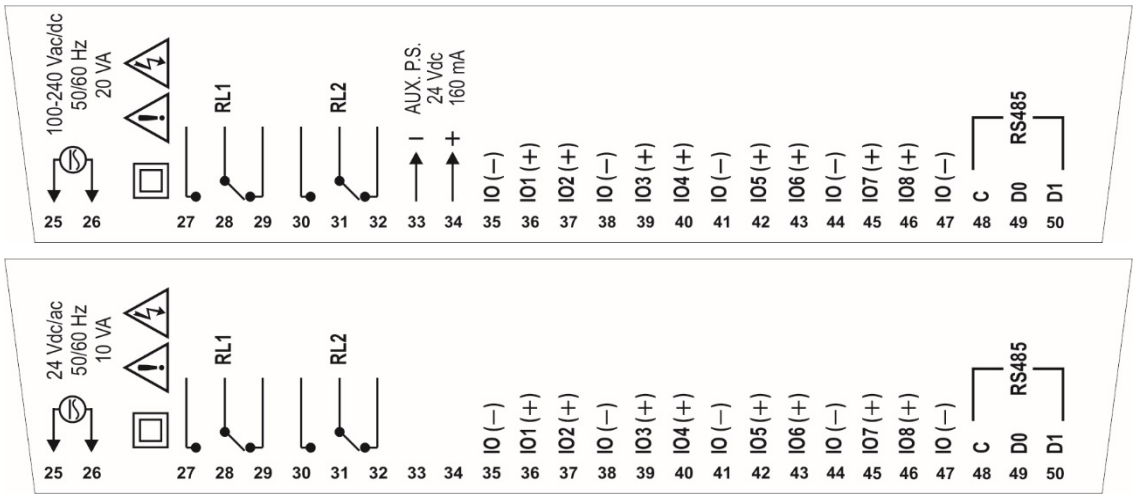


Figure 15

A switch or circuit-breaker placed near to **FieldLogger** shall be used as a disconnecting device.

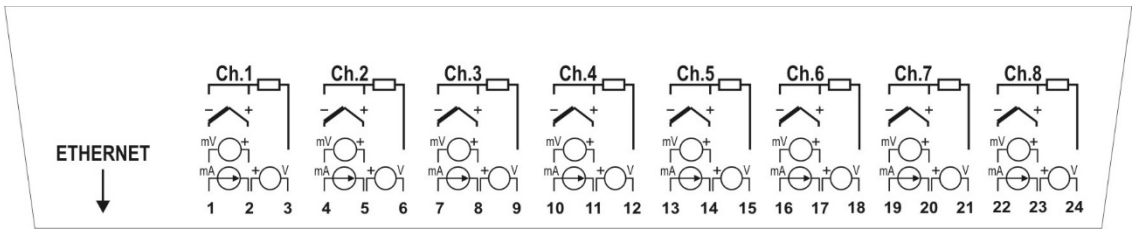


Figure 16

2.2.2 INSTALLATION RECOMMENDATIONS

- Input signal leads must be installed in grounded conduits and away from power or contactor wires.
- Instruments must be powered only by an exclusive power supply.
- Installing RC filters (47 Ω and 100 nF, serial) is strongly recommended at contactor coils or any other inductors.
- System failure should always be considered when designing a control panel to avoid irreversible damage to devices or injury to people. The RL1 and RL2 output relays, used as alarms, do not warrant total protection.
- Wires section: minimum gauge 0.14 mm².

2.2.3 POWER SUPPLY

The terminals 25 and 26 indicate the main power supply of **FieldLogger**.

2.2.4 DIGITAL I/O

There are 8 I/O that can be individually configured as inputs or outputs. There is a terminal for the positive signal of each I/O, but the negative terminal of all of them is common (there is no isolation among the channels).

2.2.4.1 INPUTS

When configured as inputs, they may be connected to voltage outputs (please check the acceptable levels in SPECIFICATIONS), dry-contact outputs and NPN outputs.

Care must be taken when connecting multiple outputs because of the lack of isolation between the **FieldLogger** inputs.

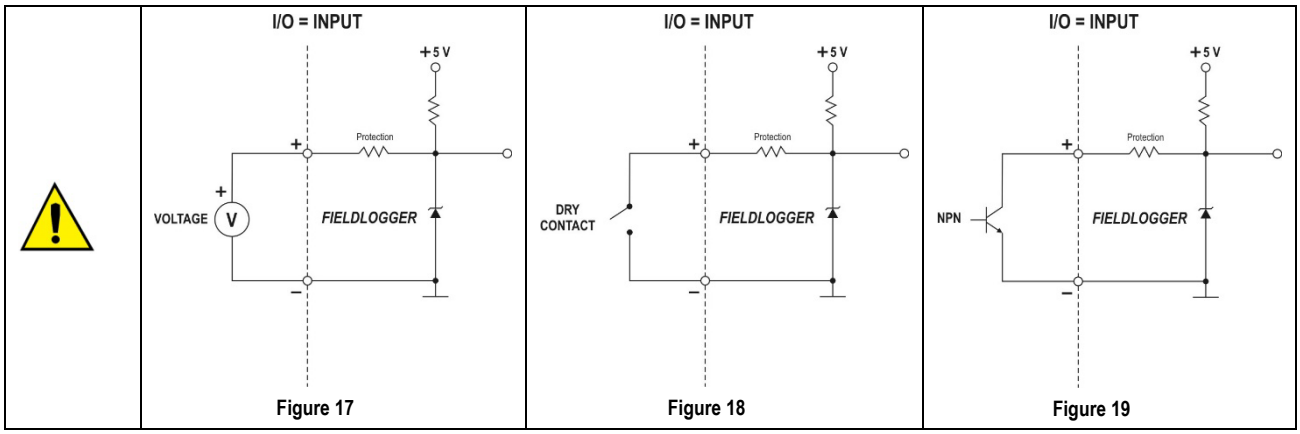


Figure 17 Figure 18 Figure 19

When we remove or disconnect the input signal, the corresponding value to the logic level "1" will be read.

2.2.4.2 OUTPUTS

When configured as outputs, they can activate limited power loads:

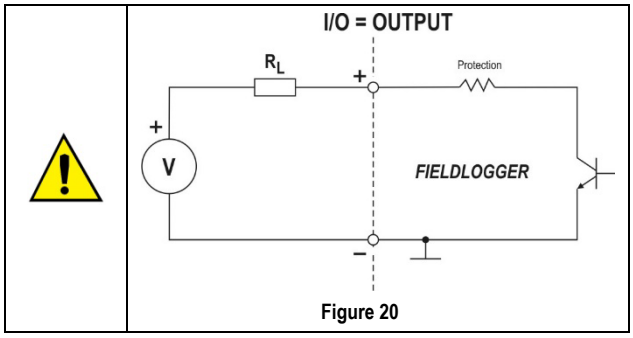


Figure 20

	The digital inputs/outputs terminals are not isolated from the analog inputs terminals! **Do not use analog and digital signals coming from the same voltage source. This will cause the device to malfunction.**

2.2.5 RELAYS

FieldLogger has 2 relays that can be used in the activation of electrical loads (check SPECIFICATIONS). For each relay, we have the common terminal, NC (normally closed) terminal and the NO (normally open) terminal. When deactivated, the relay common is in contact with the NC terminal. When activated, the common is in contact with the NO terminal.

2.2.6 RS485

The **FieldLogger** RS485 interface has terminals for 3-wire communication, including common. The connection in a Modbus network will depend on whether the device is configured to operate as a master or a slave.

	**The RS485 interface of the FieldLogger features a 1/4 UL (Unit Load) class transceiver. According to the TIA/EIA-485 standard, this allows the hardware to support addressing of up to 128 devices on the same physical bus without the need for repeaters. The actual maximum number of devices in a network also depends on the Unit Load class of the other devices connected to the same bus.**

More details about the implementation of a Modbus devices network via RS485 can be found in the document **Basic RS485 and RS422 Concepts**, available on the website www.novusautomation.com.

2.2.6.1 MASTER

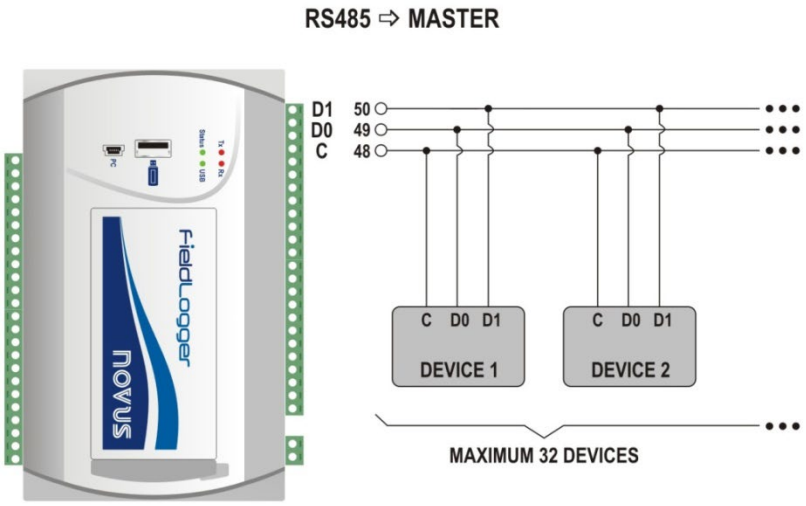


Figure 21

2.2.6.2 SLAVE

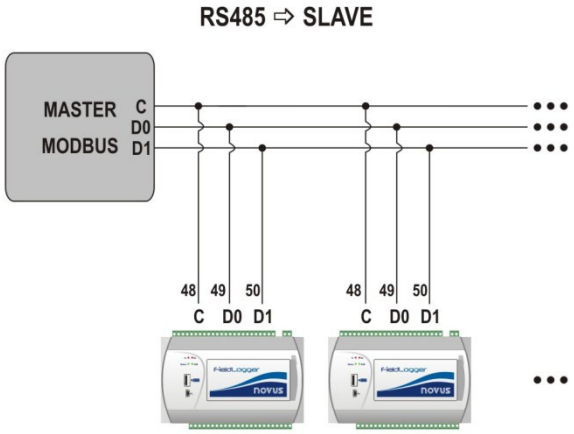


Figure 22

2.2.7 AUXILIARY POWER SUPPLY FOR POWERING TRANSMITTERS

For non 24 V models, there is a 24 Vdc power supply available in the **FieldLogger** for powering transmitters in the field. This auxiliary power supply is electrically isolated from the other **FieldLogger** terminals.

Below is the correct way to use the auxiliary power supply for powering 4-20 mA transmitters (2-wire).

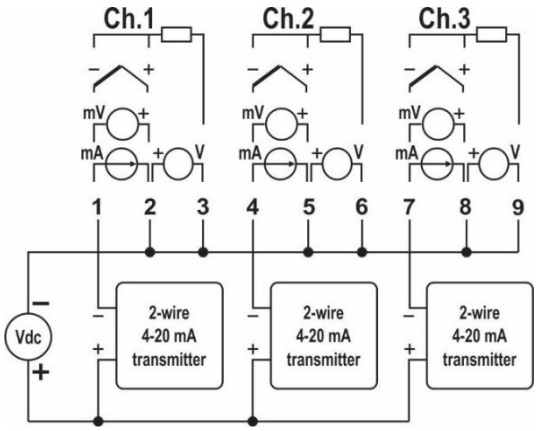


Figure 23

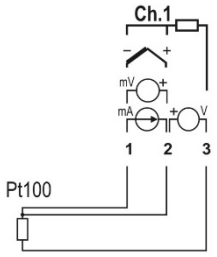
When an analog channel is configured for current input, it is necessary to make a jumper (short circuit) between the 2nd and 3rd terminals of the input. The **FieldLogger** hardware requires this assembly to reduce measurement interference between channels.

As the channels are not isolated from each other, the decision not to use this short circuit can cause interference between channels in certain situations.

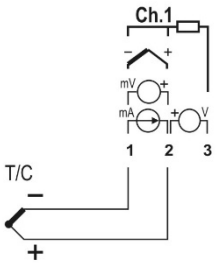
2.2.8 ANALOG INPUTS

	**The digital inputs/outputs terminals are not isolated from the analog inputs terminals!** **Do not use analog and digital signals coming from the same voltage source; this will cause the device to malfunction.**

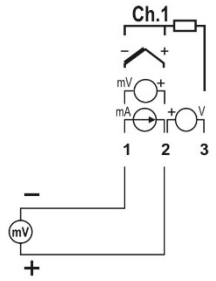
Pt100/Pt1000 CONNECTION

	The connection for the channels is made to the terminals in accordance with the figure on the left. The 3-wire connection from the Pt100 sensing element to the **FieldLogger** input guarantees the cancellation of the error caused by the resistance of the wires. All 3 wires must have the same gauge and length. For the 2-wire Pt100, interconnect terminals 1 and 2.

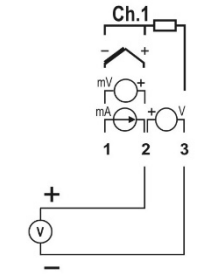
THERMOCOUPLE CONNECTION

	The connection for the channels is made in the terminals in accordance with the figure on the left. Please observe the correct connection polarity. Cables used for connecting thermocouples must have the same thermoelectric characteristics as the used thermocouple (compensation cable or extension cable) and must be connected with the correct polarity. The non-use of compensation cables or the use with incorrect polarity can cause significant measurement errors.

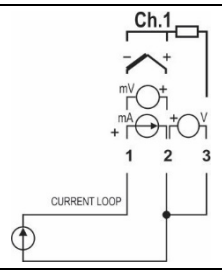
VOLTAGE (mV) CONNECTION

	The connection for the channels is made in the terminals in accordance with the figure on the left. Please observe the correct polarity of the connection.

VOLTAGE (V) CONNECTION

	The connection for the channels is made in the terminals in accordance with the figure on the left. Please observe the correct polarity of the connection.

CURRENT (mA) CONNECTION

	The connection for the channels is made in the terminals in accordance with the figure on the left. Please observe the correct polarity of the connection. For loop-powered current transmitters, you can use the same wiring scheme shown in **Figure 22**. It is important to note that since the digital inputs/outputs are not isolated from the analog inputs, you should not use the same source to power both circuits, otherwise the device may malfunction.

2.3 CONNECTIVITY RESOURCES

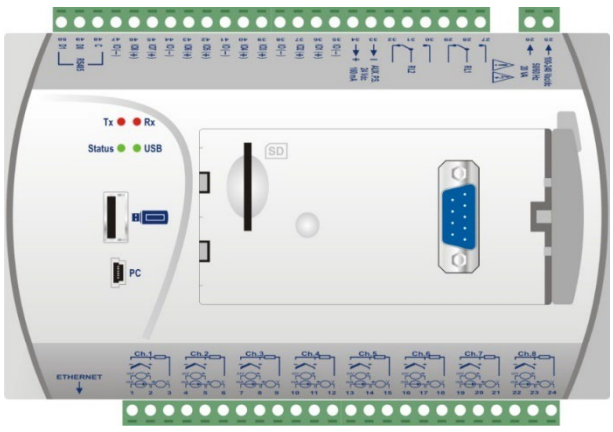
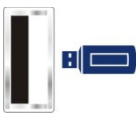


Figure 24

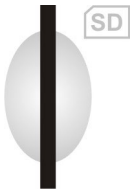
2.3.1 USB CONNECTION

	Interface used for a USB drive connection, for downloading data from the logging memory.

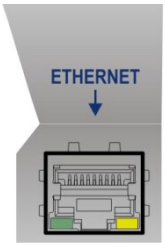
2.3.2 PC CONNECTION

	Interface used for connection to a computer for configuration, monitoring or data download.

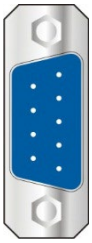
2.3.3 SD CARD CONNECTION

	Interface used for SD card expansion. The data from the logging memory can be transferred by any of the interfaces to the configuration software, which allows the exportation of the most diverse data formats.

2.3.4 ETHERNET

	Interface used for Ethernet 10/100 communication. It is recommended to use a category 5 (or better) cable in an RJ45 connector. The Ethernet connector of the **FieldLogger** has 2 LEDS for luminous indication: the green LED (on the left side) lights indicating the connection to the Ethernet network; the yellow LED (on the right side) flashes indicating that there is data traffic in the interface.

2.3.5 DB9 CONNECTION FOR HMI (OPTIONAL)

	Connection for installation of HMI (Human–Machine Interface) of the **FieldLogger** for indication of signals measured in the process. From firmware version 1.20 on, it can be used as a generic RS485/Modbus RTU (slave) port.

3 FLAGS (LEDS)

3.1 TX/RX FLAGS

	They signal transmission and reception of data by the RS485/Modbus interface, regardless of if it is configured to operate in master or slave mode.

3.2 STATUS / USB FLAGS

	When connecting the device, both LEDs flash two times and remain off until all the initialization has been completed. The **Status** flag remains ON in normal condition. When it is logging, flashes 2 times at 3 second intervals. In error cases, this LED will be flashing 3 times at 8 second intervals. In the error cases, please check if the clock of the **FieldLogger** has the correct date and time. If they are wrong, probably the battery of the clock has run low and needs to be replaced. If it is OK, try rebooting the machine by turning off its power supply and restarting it after 10 seconds. If the LED continues to indicate an error, there may be something wrong with your **FieldLogger**. The **USB** flag remains ON only while a cable is connected to the USB device or while the USB flash drive is plugged into the USB host interface. The following exceptions are: • Download errors via USB flash drive: flash drive with insufficient writing space, inability to write in the flash drive (write protected) or flash drive not compatible (sector different from 512 bytes, for example), the USB LED flashes while the error condition remains (typically, until the flash drive has been removed). Check USB INTERFACE. • At the end of the download, if everything is correct, the USB flag remains ON until the flash drive has been removed from the device.

4 NXPERIENCE SOFTWARE

The **NXperience** software is the main tool for configuring the device and downloading and analyzing data from **FieldLogger**. It allows you to explore all its features (for more information, see FIELDLOGGER OPERATION).

This chapter describes the features of **FieldLogger**. **FieldLogger** is not compatible with the monitoring functionality of the software. For instructions on other software features, see the **NXperience** operation manual. The software and its respective manual can be downloaded for free in the Download Area of our website: www.novusautomation.com.

4.1 CONFIGURATION

On the configuration screen, you can select the following options:

- **Create Configuration:** Allows you to create a configuration, which can be saved in a file for later use.
- **Read Device:** Allows you to read the current configuration from a **FieldLogger** connected to the USB interface of the computer being used.
- **Open Configuration:** Allows you to open previously saved configuration files.

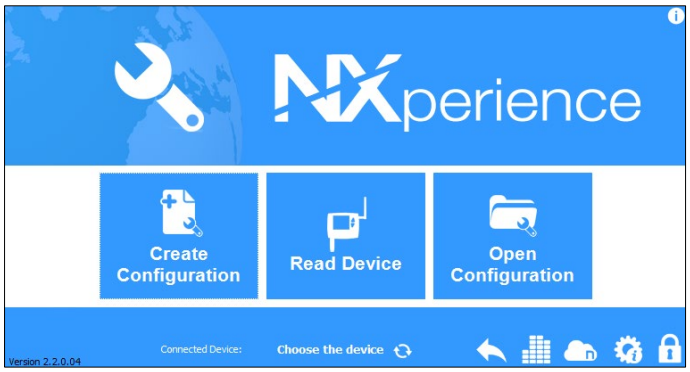


Figure 25

When you select the **Read Device** option, **NXperience** will open a new configuration window.

4.1.1 GENERAL

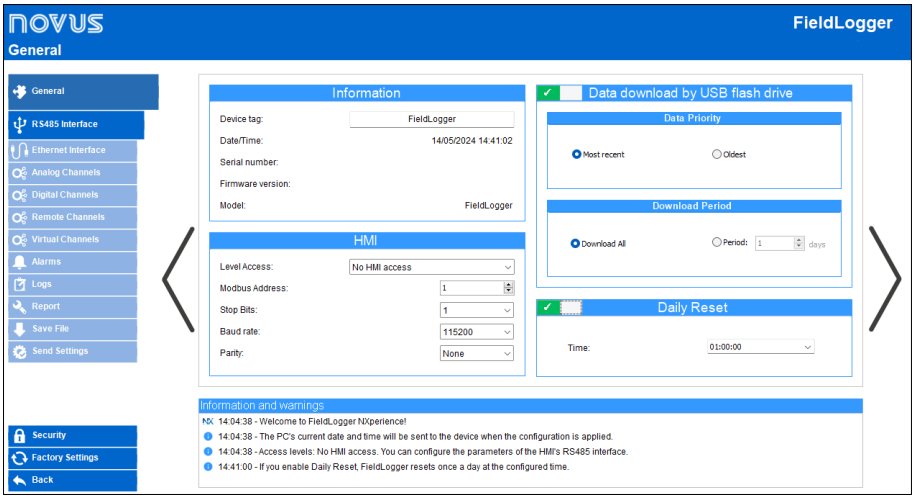


Figure 26

INFORMATION

- **Device tag:** Allows you to configure a name that will be used to identify the device. The field allows up to 20 characters.
- **Date/Time:** Displays the date/time that will be used to set the device's clock.
- **Serial number:** Displays the unique identification number of the device. The serial number is also used to register the device in **NOVUS Cloud**.
- **Firmware version:** Displays the firmware version of the device.
- **Model:** Displays the model of the device.

HMI

- **Level Access:** When using an HMI, you can identify the operator's access level through the screen:
  - **No HMI access:** Stops you from using the HMI. No parameters can be viewed by it.

    In this case, the software will display the parameters of the auxiliary RS485 interface, normally used by the HMI. These parameters can be configured to be used as a generic Modbus-RTU slave interface.
  - **Read parameters only:** The HMI can only be used to view the channels and status of **FieldLogger**. No configuration is allowed.

- **HMI configuration and general reading:** The HMI parameters can be configured. You can view the channels and status of **FieldLogger**.
- **Configuration and general reading:** HMI parameters and some **FieldLogger** parameters can be configured. You can view the channels and status of **FieldLogger**.

DATA DOWNLOAD BY USB FLASH DRIVE

- **Data priority:** When enabled, this allows you to define the priority with which data will be saved on a USB flash drive that does not have enough space to collect all the data.
- **Download period:** Allows you to define the number of days (from the day of download, if the priority is the most recent, or from the oldest data, if the priority is the oldest) that you want to download. Since it doesn't copy all the data from the device's memory (which, in the case of memory cards, can be a lot), this can speed up the download time.

DAILY RESET

- **Time:** From firmware version 1.64, you can reset the **FieldLogger** once a day, at a set time. This practice, common in routers, guarantees that if everything goes wrong and communication can no longer progress, restarting the device will solve the problem.

INFORMATION AND WARNINGS

The **Information and Warnings** screen remains at the bottom of all configuration sections, displaying important information about the configuration process.

4.1.2 RS485 INTERFACE

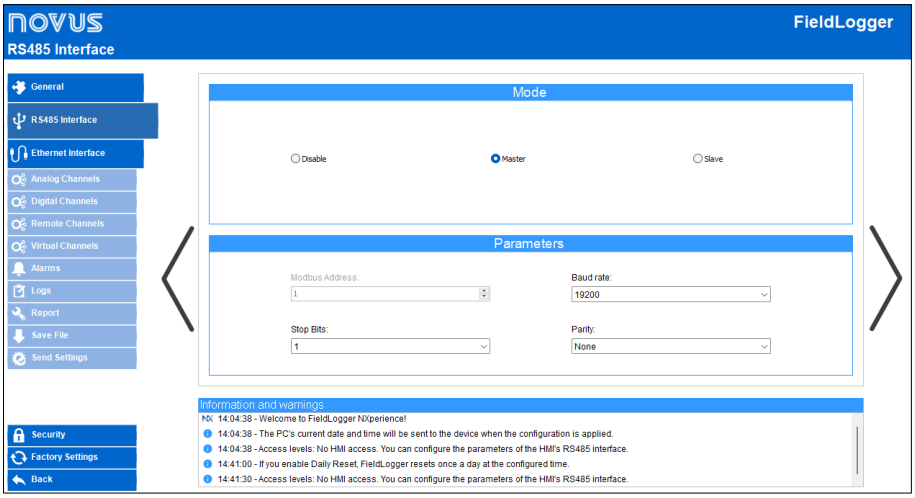


Figure 27

MODE

In this section, you can choose the operating mode of this interface: **1)** Disabled, **2)** Modbus-RTU Master, or **3)** Modbus-RTU Slave.

If this interface is not used, it is recommended to disable it.

PARAMETERS

When you select the **Slave** option, you must configure the Modbus address, Baud Rate, parity, and the number of Stop Bits.

When you select the **Master** option, you do not need to configure the Modbus address (only valid for the slaves). Furthermore, in this case, the configuration of the Modbus network, which allows you to define which registers are read from which slaves, will be done later, in the **Remote Channels** screen (see REMOTE CHANNELS).

4.1.3 ETHERNET INTERFACE

4.1.3.1 TCP/IP

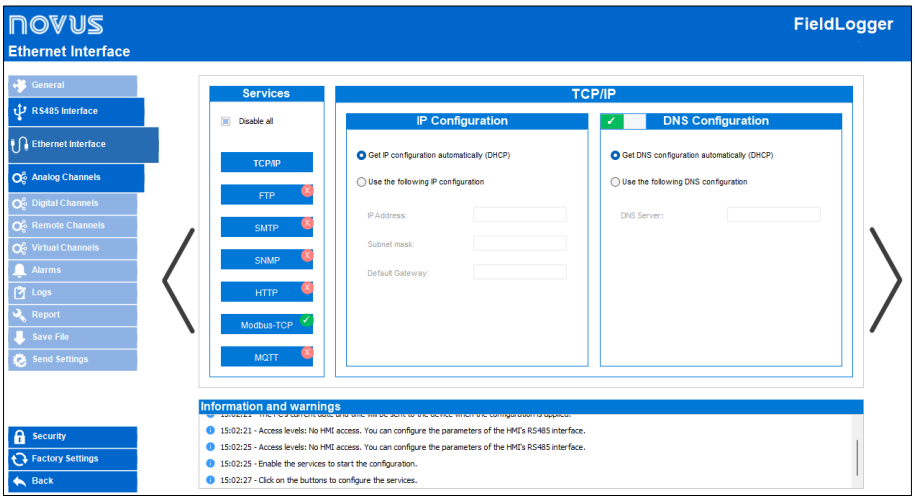


Figure 28

IP CONFIGURATION

In this section, you can define how to configure the IP: **1)** Static IP or **2)** DHCP. When you select the **Use the following IP configuration** option, you must fill in the **IP Address**, **Subnet Mask**, and **Default Gateway** parameters.

DNS CONFIGURATION

You can enable or disable the DNS function. When enabling the DNS, you must choose between using a fixed DNS or DHCP. The DNS allows you to connect to the email server or the FTP server (when **FieldLogger** is the FTP client for daily data download via this service).

When you select the **Use the following DNS configuration** option, you must configure the IP number of the DNS server. By selecting the **Get DNS configuration automatically (DHCP)** option, **FieldLogger** can search for the DNS server IP from the DHCP server.

4.1.3.2 FTP

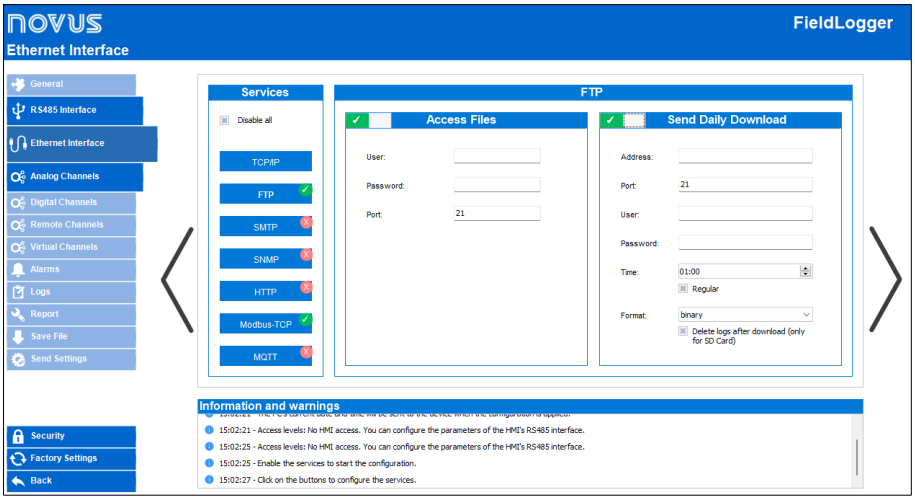


Figure 29

FieldLogger provides two types of FTP: **1)** Client and **2)** Server.

ACCESS FILES

As a server, **FieldLogger** allows an external client to connect to the device to download log data from both the SD card and the internal memory, by enabling the **Access Files** tab and filling in the parameters below:

- **User:** Allows you to configure connection and access data. The field allows up to 10 characters.
- **Password:** Allows you to enter the connection password. The field allows up to 10 characters.
- **Port:** Allows you to enter the connection port number.

SEND DAILY DOWNLOAD

As a client, **FieldLogger** can automatically connect to an FTP server to send data from its log memory (internal or memory card) by enabling the **Send Daily Download** tab and filling in the parameters below:

- **Address:** Allows you to enter the IP address or, if DNS is enabled, the server's name. For names, it allows up to 50 characters.
- **Port:** Allows you to define the server port to be used.
- **User:** Allows you to define the connection user. The field allows up to 50 characters.
- **Password:** Allows you to define the connection password. The field allows up to 10 characters.
- **Time:** Allows you to send the download once a day, at a set time, or several times a day (from firmware version 1.50), over a selectable time.
  - **Regular:** If selected, allows you to send the download several times a day, over a selectable time. In hours.
- **Format:** Allows you to define the format of the file generated by the download: **1)** Binary (default), **2)** CSV (Comma Separated Value) or **3)** TXT (text).
- **Delete after download:** If selected, allows you to delete the log files after the download. Only valid when registering on a memory card.

4.1.3.3 SMTP

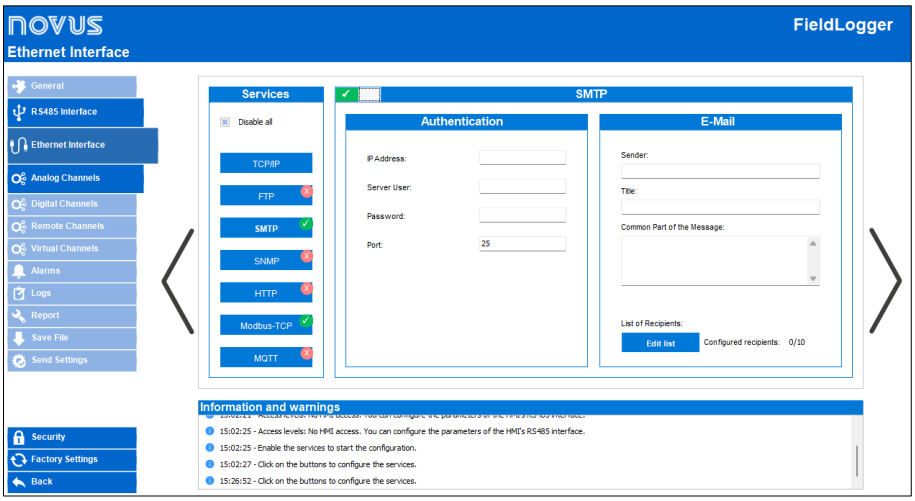


Figure 30

During alarm situations, **FieldLogger** allows you to send emails to multiple recipients.

AUTHENTICATION

In this section, you must configure the parameters related to the email server:

- **IP address:** Allows you to enter the IP address or, if the DNS is enabled, the server's name. For names, it allows up to 50 characters.
- **User:** Allows you to define the connection user. The field allows up to 50 characters.
- **Password:** Allows you to define the connection password. The field allows up to 12 characters.
- **Port:** Allows you to define the server port to be used.

EMAIL

In this section, you must configure the parameters for the email to be sent in the event of an alarm.

- **Sender:** Allows you to enter the email address of the sender of the message. The field allows up to 50 characters.
- **Title:** Allows you to set a title for the email to be sent. The field allows up to 32 characters.
- **Common part of the message:** Allows you to configure a message that will be identical for all alarms. The field allows up to 64 characters.
- **List of recipients:** By clicking on the **Edit list** button, you can add an email. **FieldLogger** allows up to 10 emails.

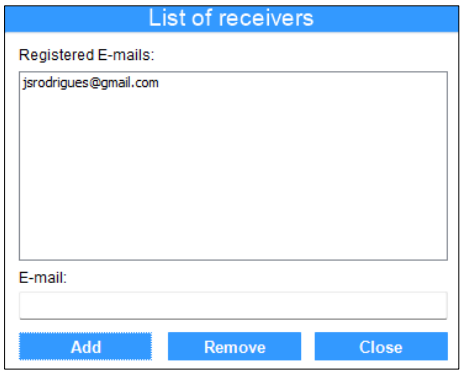


Figure 31

Using the **Add** and **Remove** buttons, you can add and remove emails from the recipient list. All possible recipients of alarm emails must be included in this window. On the **Alarms** screen, you can select which alarms should be sent to which recipients (see ALARMS).

4.1.3.4 SNMP

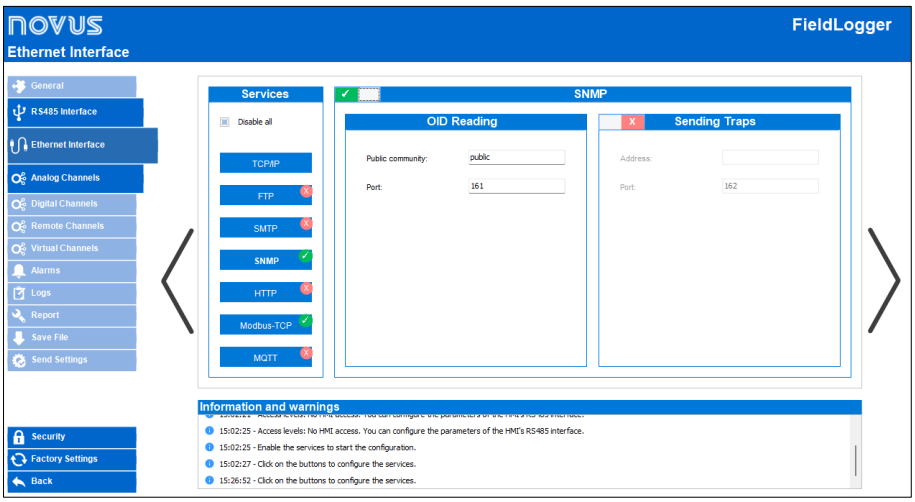


Figure 32

OID READING

The SNMP protocol only allows readings to be performed. To do this, you must configure the parameters below:

- **Public community:** Allows you to configure the community for access. The field allows up to 16 characters.
- **Port:** Allows you to define the server port to be used.

SENDING TRAPS

To enable the sending of traps, you must configure the parameters below:

- **Address:** Allows you to enter the IP address of the connection.
- **Port:** Allows you to enter the destination port for the traps.

4.1.3.5 HTTP

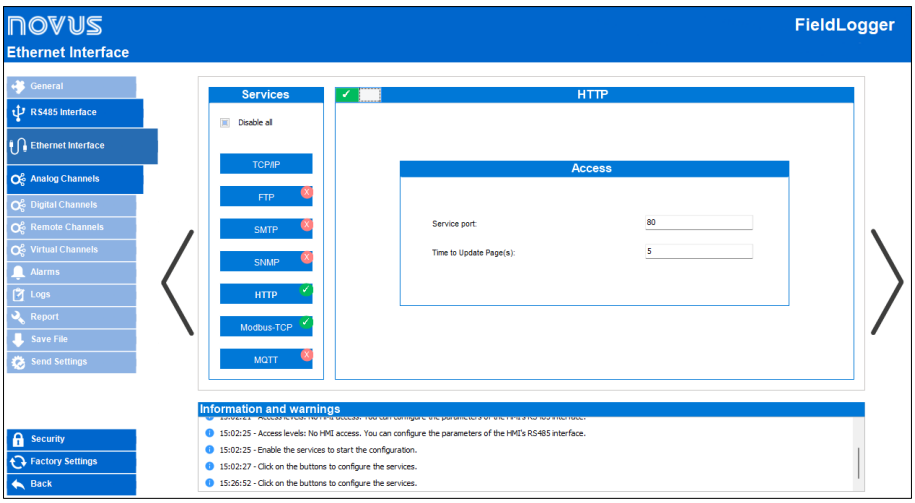


Figure 33

By enabling the HTTP interface, **FieldLogger** allows you to access a web page with data from the device. This page has a refresh parameter, which tells the browser how often the page should be reloaded with the updated data.

To do this, you must configure the **Service Port** parameter, which refers to the connection port, and the **Time to Update Page(s)** parameter, which refers to the time in seconds between page refreshes.

4.1.3.6 MODBUS-TCP

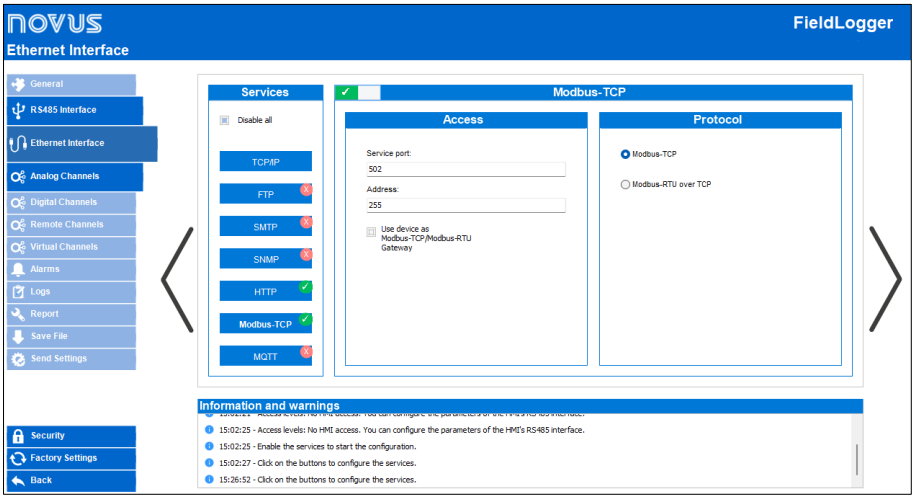


Figure 34

The Modbus-TCP communication protocol allows data to be read from and written to **FieldLogger**. Among other things, **NXperience** uses this protocol to communicate with **FieldLogger** to read and configure the device, perform diagnostics, and download data by selecting the Ethernet interface.

ACCESS

FieldLogger must be accessed via the identifier (ID) configured in the **Address** parameter. Any different identifier used when accessing **FieldLogger** via Modbus-TCP will be understood as being destined for a slave on the RS485 network to be accessed via the Gateway function. In this case, if this feature is enabled, the packet will be retransmitted to the RS485 bus.

When the RS485 interface is configured as **Master** (see RS485 INTERFACE) and the Modbus-TCP protocol is enabled, you can select the **Use the device as a Modbus-TCP/Modbus-RTU Gateway** option, which allows you to send Modbus commands (Ethernet Interface – Modbus-TCP) to the slaves on the Modbus-RTU network (RS485 interface) via **FieldLogger**.

⚠	**FieldLogger operation as a Gateway between a Modbus-TCP network and the Modbus-RTU network is only available from firmware version 1.10 onwards.**

PROTOCOL

From firmware version 1.40, you can select the communication protocol for this interface: **1)** Modbus-TCP or **2)** Modbus-RTU over TCP.

4.1.3.7 MQTT

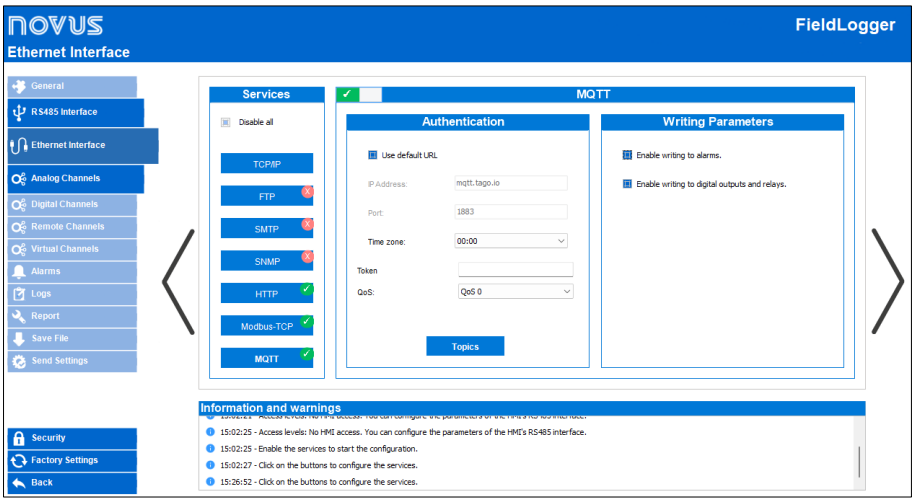


Figure 35

The MQTT protocol allows you to publish data to the cloud. **FieldLogger** is compatible with generic clouds and **NOVUS Cloud**. When using this protocol with **NOVUS Cloud**, however, it is necessary to select the **FieldLogger MQTT** option when configuring the device in the cloud.

The implementation of this protocol is compatible with versions 3.1 and 3.11, but it does not meet security requirements, i.e. it does not support TLS.

⚠	**1. To use the cloud data publishing service, you need an Internet connection.** **2. Data recorded on the SD card is NOT sent via MQTT.**

AUTHENTICATION THROUGH NOVUS CLOUD

To use **NOVUS Cloud**, simply check the **Use default URL** option. This will automatically fill in the **IP Address** and **Port** parameters. When configuring the device in **NOVUS Cloud**, you will need to add the token generated for it. For more information on how to do this, see the **NOVUS Cloud** manual, available on our website.

AUTHENTICATION WITH A GENERIC BROKER

To use a generic Broker, you must uncheck the **Use default URL** option and fill in the following parameters:

- **IP Address:** Allows you to define the Broker's address (IP Address or URL). The field allows up to 50 characters.
- **Port:** Allows you to define the service port number to be used to connect to the Broker.
- **User:** Allows you to enter the name of the user registered with the Broker. The field allows up to 36 characters.
- **Password:** Allows you to enter the password of the user registered with the Broker. The field allows up to 50 characters.
- **Time zone:** Allows you to configure the connection according to the local time zone.
- **QoS:** Allows you to select the quality-of-service level used when sending MQTT messages. Only levels 0 and 1 are available.

TOPICS

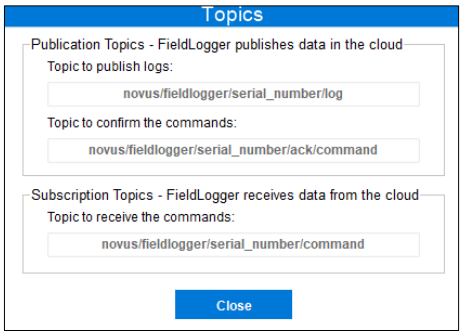


Figure 36

When you click on the **Topics** button, **NXperience** will display the typical **FieldLogger** publication and subscription topics.

These topics are not editable.

WRITING PARAMETERS

By selecting the **Enable writing to alarms** option or the **Enable writing to digital outputs and relays** option, it is possible, as the name suggests, to write to both the digital outputs and the alarm Setpoints of the device via **NOVUS Cloud**.

Writing to digital outputs and relays is only possible by configuring the desired channel as **Output** on the **Digital Channels** screen and then selecting the **Output controlled by Modbus commands** option (see DIGITAL CHANNELS section).

4.1.4 ANALOG CHANNELS

This screen allows you to select and configure the analog channels to be used:

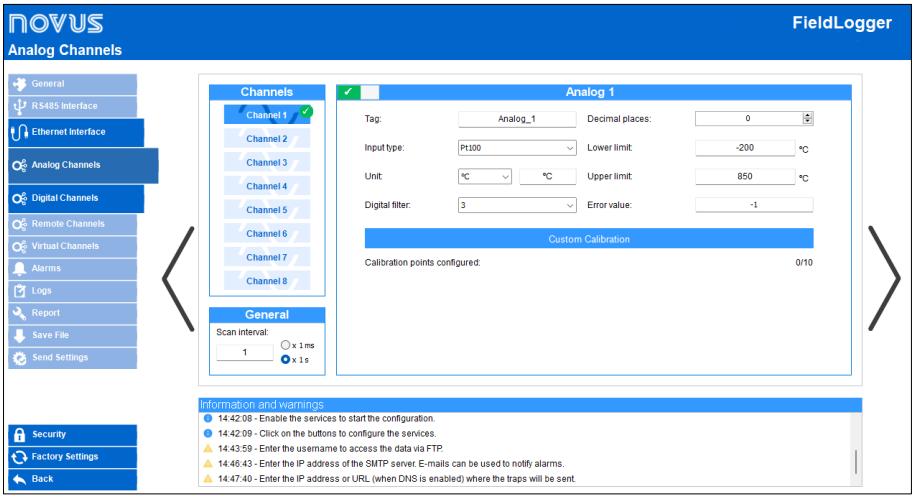


Figure 37

CHANNEL

- **Tag:** Allows you to configure a name that will be used to identify the channel. The field allows up to 16 characters.
- **Input Type:** Allows you to select the type of input to be used. Depending on the type of input configured, the display limits must also be configured (in others, this limit is fixed).

- **Unit:** When configuring the **Input Type** parameter with one of the temperature sensor options, you can set the display in Celsius or Fahrenheit degrees. When configuring the **Input Type** with the other types, you can set a string for the unit. In this case, the field allows up to 10 characters.
- **Digital Filter:** Allows you to set a digital filter for the input channel. The higher the value, the more filtered the channel indication becomes, making the response more immune to noise in the input signal, but also slower to variations. The maximum filter value is 20.
- **Decimal Places:** Allows you to set the number of decimal places to be displayed. This parameter refers to the following cases:
  - Reading the channel value via 16-bit Modbus registers (INT16 with signal).
  - Reading the channel value via HMI.
  - Reading the channel value via the HTML page generated by **FieldLogger** (HTTP service of the Ethernet interface. See HTTP).
  - Reading the channel value by reading the OID from the SNMP protocol (SNMP service of the Ethernet interface. See SNMP).
  - Reading of the alarm value related to the channel sending emails by **FieldLogger** (SMTP service of the Ethernet interface. See SMTP).
- **Lower Limit:** Allows you to set the lower limit value for the linear input type. This parameter is editable if the **Input Type** parameter is set to a linear input type.
- **Upper Limit:** Allows you to set the upper limit value for the linear input type. This parameter is editable if the **Input Type** parameter is set to a linear input type.
- **Error Value:** Allows you to set an error value to be displayed whenever the **FieldLogger** detects an error in the input signal, such as a broken 4-20 mA loop or a Pt100 with a broken cable.

SCAN INTERVAL

You can set the Scan Interval of the **FieldLogger's** analog channels between 1 millisecond and 5 seconds. However, it is important to remember that accuracy is only guaranteed when the scan interval is greater than 200 ms per channel (i.e. if 8 channels are enabled, 1.6 s). To obtain the best resolution and signal-to-noise ratio, several samples are needed for filtering.

By configuring the device with a scan interval of less than 50 ms per channel, the 50/60 Hz automatic filter is disabled and therefore the readings can vary greatly depending on the noise present in the application environment.

Thermocouple, mV, Pt100, and Pt1000 sensors, which have a low amplitude signal, are more susceptible to variations in readings. Meanwhile, 4-20 mA and 0-10V signals, which have a higher amplitude, have a better signal-to-noise ratio.

4.1.4.1 CUSTOM CALIBRATION

By clicking on the **Custom Calibration** button, you can define up to 10 custom calibration points for each analog channel:

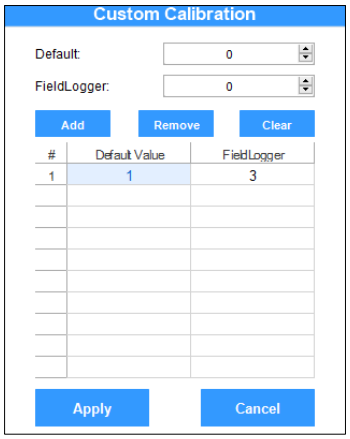


Figure 38

The 10 pairs of inserted points create line segments to adjust the indicated value (see FIELDLOGGER OPERATION).

To create a customized calibration point, you must:

- Apply the desired values to the **Default** and **FieldLogger** parameters.
- After entering the custom calibration point pairs, click on **Add**.
- To remove a custom calibration point, simply select it and click on **Remove**. To remove all points created, click on **Clear**.
- Once all the customized calibration points have been added, click on **Apply**.

	**Before measuring new points for another custom calibration, all custom calibration points must be deleted! The fact that there is already calibration points on the device will mask the measurements and the new calibration may contain errors.** **For the same reason, all custom calibration points must be entered into FieldLogger at the same time.**

4.1.5 DIGITAL CHANNELS

This screen allows you to select and configure the analog channels to be used:

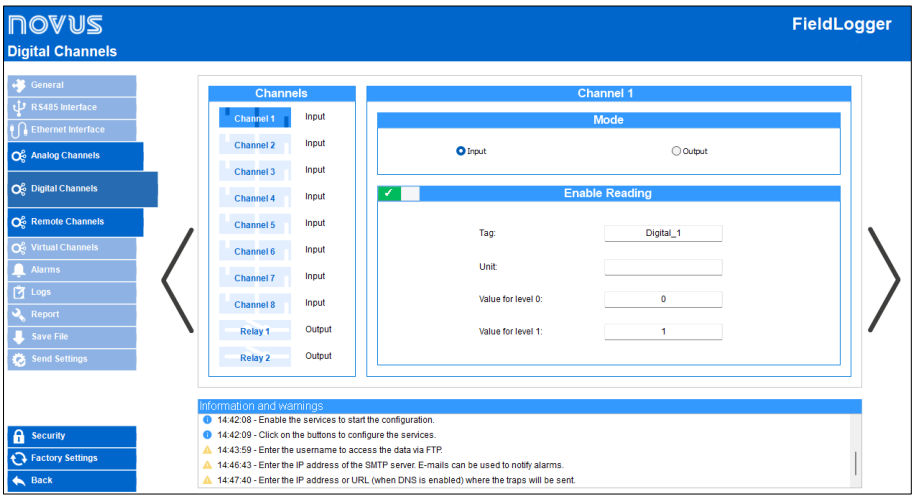


Figure 39

MODE

Each channel can be configured as an input or output. As an input, you can enable it or not. If it is disabled, it will not appear as an option in logs, alarms, virtual channels, or even the HMI.

ENABLE READING

Once the channel has been configured as an input and enabled, the parameters must be configured:

- **Tag:** Allows you to configure a name that will be used to identify the channel. The field allows up to 16 characters.
- **Unit:** Allows you to configure the input unit. The field allows up to 10 characters.
- **Value for level 0:** Allows you to set the value to be displayed when the input is at level 0, i.e. when the voltage level is low or when the contact is closed on the input.
- **Value for level 1:** Allows you to set the value to be displayed when the input is at level 1, i.e. when the voltage level is high or when the contact is open on the input.

From firmware version 1.10, you can perform counts on the **FieldLogger's** digital inputs. No special configuration is required. Simply leave the channels enabled as inputs. These counts can be accessed via Modbus registers (see "FieldLogger – Modbus" document) and can be copied to a virtual channel, from where they can be registered or used in alarms, for example.

When you apply a new configuration to the digital channels, the counts for all of them will be reset to zero.

OUTPUT

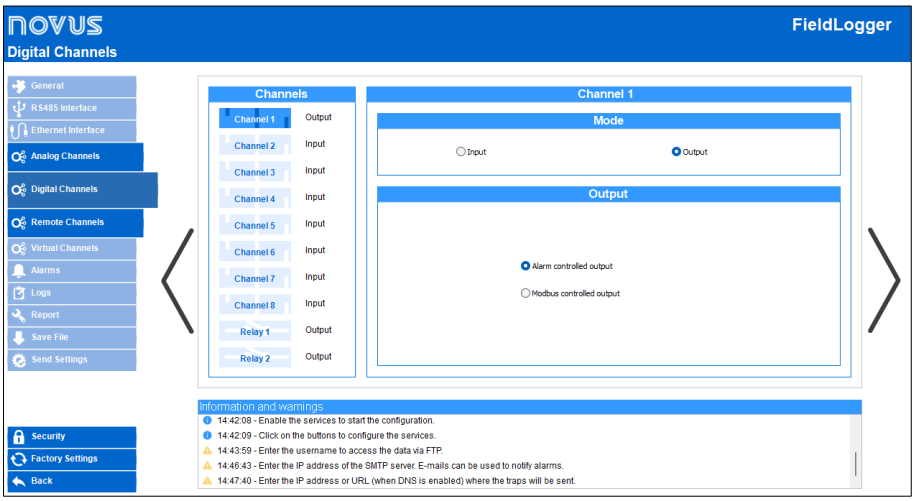


Figure 40

Once the channel has been configured as an output, it must be configured with one of the options below:

- **Alarm controlled output:** Allows this output to be activated only by the **FieldLogger's** alarms.
- **Modbus controlled output:** Allows this output to be activated only by external Modbus commands, coming from a PLC or supervisory software, for example.

  You must use this option to allow the outputs to be activated by MQTT.

The status set via the Modbus command is not retentive (i.e. it is lost after a reset, for example during a power failure). It will remain if the output is configured in this way.

If this option is checked, the status set by the Modbus command will be maintained even after a new configuration has been sent.

RELAY

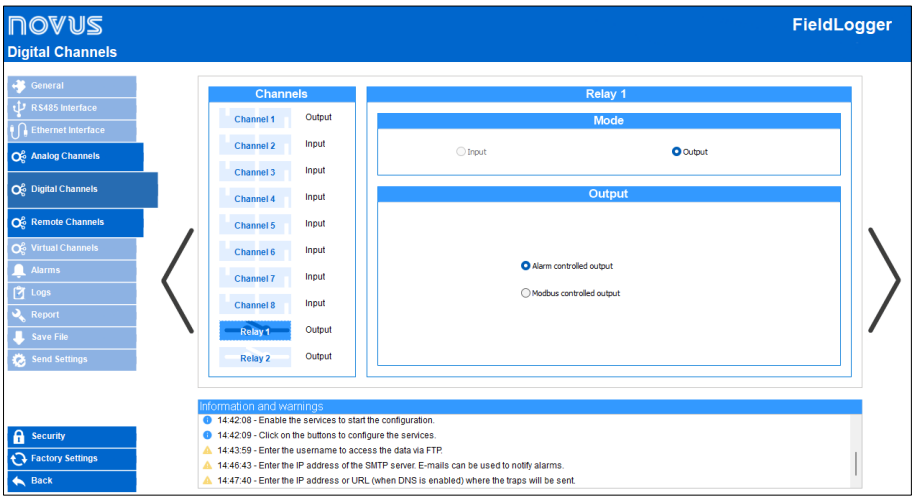


Figure 41

The configuration of the 2 relays is like that of the other digital outputs. Simply indicate whether they will be triggered by internal alarms or by external Modbus commands.

4.1.6 REMOTE CHANNELS

The configuration of remote channels is only available when the RS485 interface is configured as a **Modbus** master (see RS485 INTERFACE). If the RS485 interface is configured as a **Slave** or has been disabled, the **Remote Channels** screen will display a message preventing it from being configured.

With the RS485 interface configured as a Modbus master, the following screen will be displayed:

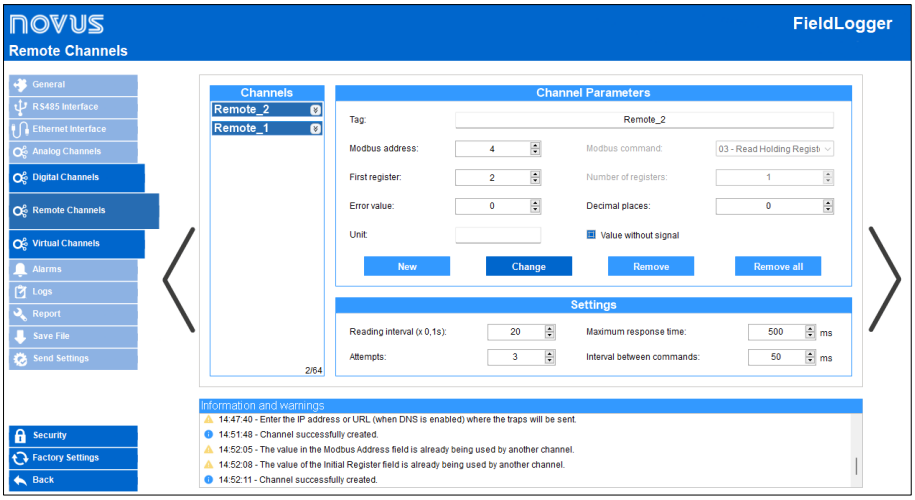


Figure 42

CHANNEL PARAMETERS

- **Tag:** Allows you to configure a name that will be used to identify the channel. The field allows up to 16 characters.
- **Modbus address:** Allows you to select the address of the Modbus slave to be read.
- **Modbus command:** Allows you to define the Modbus command to be used.
- **Initial register:** Allows you to set the initial register.
- **Number of registers:** Allows you to set the number of registers to be read by the slave. It is possible to read up to a maximum of 8 registers in a single Modbus command.

  When linked to virtual channels, this feature makes it possible to obtain values from 32-bit registers (see VIRTUAL CHANNELS for an example of how to configure virtual channels for 32-bit registers). When reading in blocks to obtain 32-bit registers, the **Value without signal** parameter does not affect the result.

  The block reading feature is available for firmware versions higher than 1.70 and configurator software versions higher than 1.6.9.00.
- **Error value:** Allows you to set an error value. This value will be displayed on the channel when there is a problem, such as when there is a communication error with the slave.

- **Decimal places:** From firmware version 1.40, allows you to set the number of decimal places to be displayed.
- **Value without signal:** Allows you to define whether the value being read will be an unsigned or signed value, which basically depends on how the slave makes the information available.
- **Unit:** Allows you to set a unit for the value read. The field allows up to 10 characters.

To configure a remote channel, click on the **New** button. To finish the configuration, click on the **Create channel** button. This way, the configured channel will appear in the channel list on the left of the screen.

To remove a channel from this list, simply select it in the list and then click on **Remove**. To remove all channels, click on **Remove all**. It is not possible to remove just one register from the channel.

SETTINGS

At the bottom of the screen, you must configure the general parameters of the Modbus master.

- **Reading interval (x 0.1 s):** Allows you to configure the reading interval (in tenths of a second) of all configured remote channels. The field allows up to 18,000 seconds, which is equivalent to 30 minutes.
- **Attempts:** Allows you to set the number of attempts to be made during each scan. If communication fails on all attempts, **FieldLogger** will display the error value configured for each channel. Range: 1~10.
- **Maximum response time:** Allows you to define how long after the master sends the command it will wait for the slave to respond. Range: 2~10000 ms.
- **Interval between commands:** Allows you to define how long **FieldLogger** should wait between receiving the slave's response and sending the next command. Range: 1~5000 ms.

4.1.7 VIRTUAL CHANNELS

This screen allows you to select and configure the virtual channels to be used:

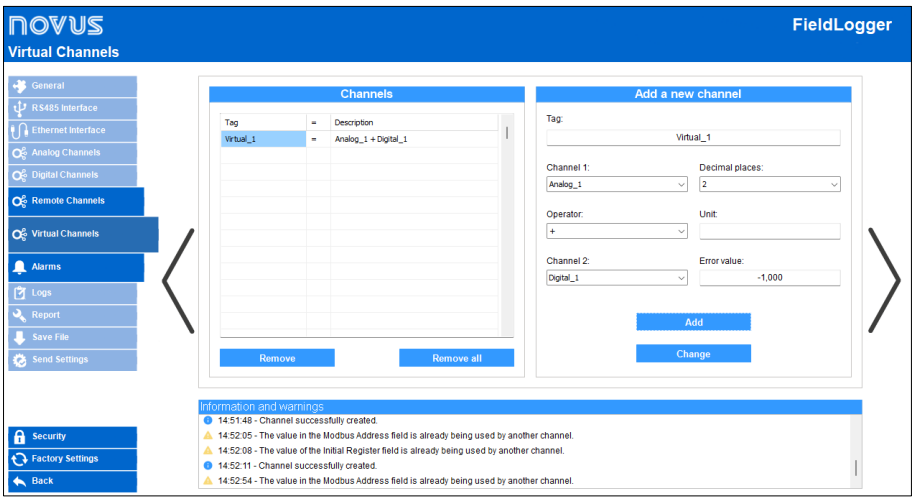


Figure 43

Each virtual channel is the result of a mathematical or logical operation. To configure it, you must:

1) Choose a Tag for the channel. The field allows up to 16 characters.
2) Select the first channel that will serve as the operand (depending on the operation chosen, it will be the only one).

   By selecting the **Constant value** option, you can link a numerical value with the virtual channel. This amount can be used as an operand on other virtual channels.
3) Choose the operation to be performed.
4) Choose the channel to be used as the second operand in the operation.

   Once the **Constant value** option has been chosen, you must enter the channel's numerical value.
5) Select a unit. The field allows up to 10 characters.
6) Define an error value.
7) Define the number of decimal places to be used.

By clicking on **Add**, the newly configured channel will be added to the list of virtual channels.

Whenever a virtual channel is configured, all enabled input channels can be used as operands, including virtual channels already on the list. This allows us to generate relatively complex expressions, using the result of one operation as an operand in another. For more details, see VIRTUAL CHANNELS.

When you click on any virtual channel in the list, its parameters are loaded at the top. To delete a channel, select it from the list and click on **Remove**. To delete all virtual channels, click on **Remove all**.

4.1.8 ALARMS

This screen allows you to create alarms to indicate error or exception situations:

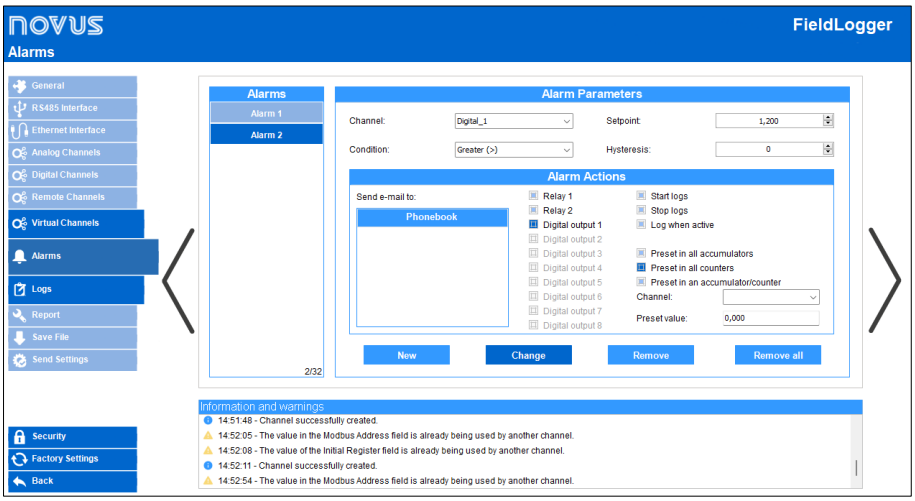


Figure 44

ALARM PARAMETERS

- **Channel:** Allows you to select the alarm to be used.
- **Condition:** Allows you to select the alarm condition.
- **Setpoint:** Allows you to define a comparison Setpoint.
- **Hysteresis:** Allows you to define a hysteresis value.

You can, for example, select an analog temperature channel to alarm when the temperature exceeds a critical limit or configure a digital input channel to alarm when its status differs from the expected value.

You can configure a hysteresis to prevent a marginal condition, such as an input that keeps oscillating around the Setpoint value, from generating multiple alarm events.

	**When an alarm uses a digital channel, hysteresis should not be used. The parameter must be set to "0"!**

ALARM ACTIONS

As a rule, each alarm can have one or more linked actions. The actions to be taken in the event of an alarm must then be chosen. After selecting the channel to be linked, the available actions are:

- Activating relays.
- Activating digital outputs (if they are configured to be activated by an alarm).
- Sending emails to one or more recipients (chosen from the emails configured on the Ethernet interface setting page).
- Sending a SNMP trap.
- Start and/or end of the logs.
- Forcing values into counter and/or accumulator channels.

SAFE FAILURE

Once all the settings have been made, the alarm will be created by clicking on the **Create alarm** button.

When you click on an alarm in the list on the left, its parameters will be loaded into the fields on the right. To modify an alarm parameter, update the desired parameter and click on **Change**.

To delete an alarm from the list, select it and click on **Remove**. To delete all alarms, click on **Remove all**.

4.1.9 LOGS

This screen allows you to define the settings for the logs:

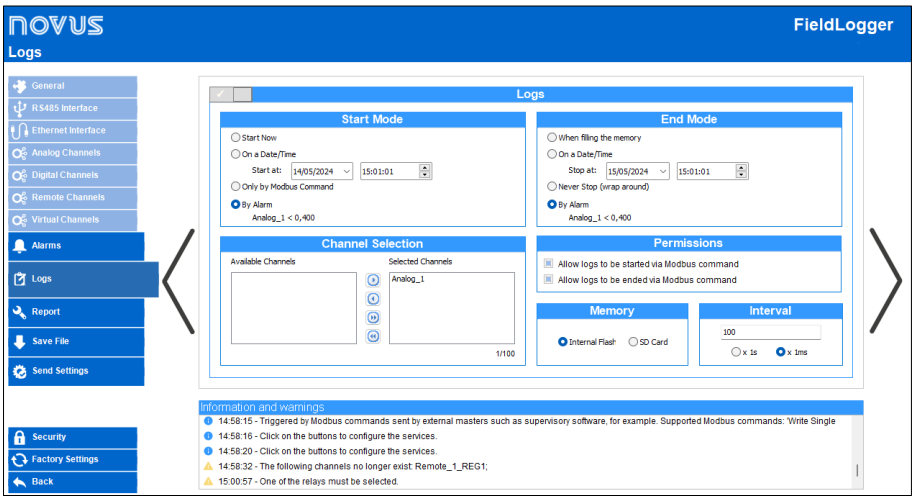


Figure 45

If the logs are not wanted, they should be disabled.

START MODE AND END MODE

You must define a start mode and an end mode. If, on the previous screen, the start and/or end mode of the logs has been assigned to one or two alarms, this option cannot be changed on the **Logs** screen.

If you do not want the **FieldLogger** to log, it must be disabled.

PERMISSIONS

By selecting the options **Allow start of logs via Modbus command** or **Allow end of logs via Modbus command**, it is possible to carry out this process via Modbus commands.

MEMORY

You must select the memory in which the logs will be saved: **1)** Internal flash memory (allows up to 512k logs) or **2)** Memory card inserted by the user.

CHANNEL SELECTION

You must select which channels you want to register and the desired recording rate. The channels to be recorded should be selected by moving from the left-hand column (available channels) to the right-hand column (channels to be registered).

A maximum of 100 channels can be registered, but remember that the more channels you register, the slower the recording rate you can use.

INTERVAL

The recording rate is common to all channels, i.e. a rate of 1 second means that all channels selected for recording will be registered once a second. Although the device allows a recording rate of up to 1000 per second, the following side effects should be considered:

- There's no point in having a recording rate that's faster than the reading rate of analog channels or the reading rate of remote channels. In these cases, the logs would have a lot of repeated data.
- The more logs in memory, the slower the download process will be, and the more data will have to be processed during the viewing and exporting process.

4.1.10 SEND SETTINGS

To send the configuration to **FieldLogger**, click on **Send configuration**.

4.1.11 FUNCTIONS MENU

On the left of the screen, below the configuration section, there is a tools menu:

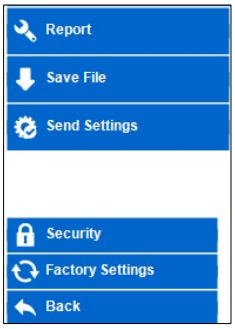


Figure 46

- **Report:** Allows you to create a report with a .fp3 extension, displaying the device's settings. You can also export this information to an Excel file.
- **Save file:** Allows you to save the configuration in a file with a .nxc extension, which you can later open in **NXperience**.
- **Send settings:** Allows you to send the settings to the device.
- **Security:** Allows you to password protect the **FieldLogger** configuration or download.
- **Factory setting:** Allows you to return the device to its factory settings.

4.2 DIAGNOSTICS

When you click on the **Diagnostics** button on the main screen, **NXperience** will establish a connection with **FieldLogger**, displaying the following screen:

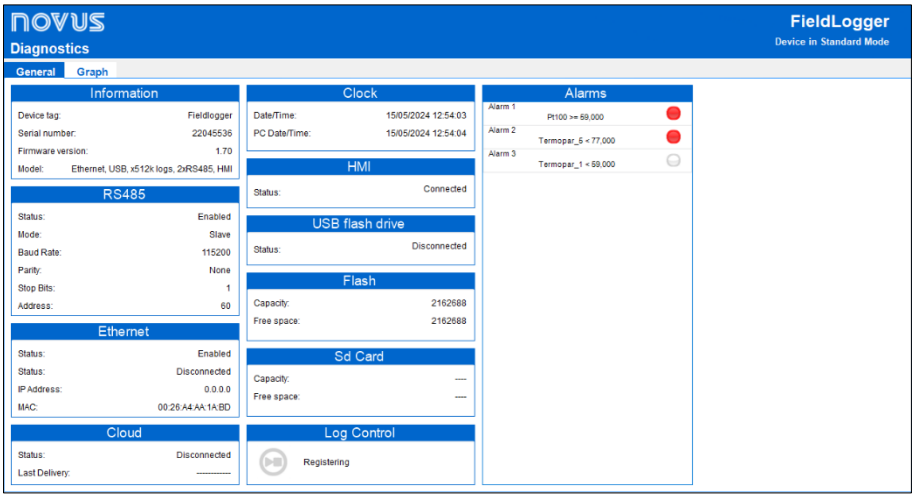


Figure 47

4.2.1 GENERAL

This screen has the following sections:

- **Information:** Displays the device tag, device number, firmware version, and model.
- **RS485:** Displays information on the status of the RS485 interface (whether it is enabled or disabled), the operating mode, Baud Rate, parity, Stop Bits, and configured address.
- **Ethernet:** Displays information about the status of the Ethernet interface (whether it is enabled or disabled), the status, the IP address, and the MAC.
- **Cloud:** Displays information about the status of the connection to the cloud, as well as the time since the last successful upload.
- **Clock:** Displays information about the current date/time and the date/time of the PC being used.
- **HMI:** Displays information on the status of the HMI.
- **USB flash drive:** Displays information about the status of the USB flash drive.
- **Flash:** Displays information about the capacity and free space of the internal Flash memory.
- **SD Card:** Displays information about the capacity and free space of the memory card (when connected).
- **Via Software:** Informs whether the device is registering.
- **Alarms:** Allows you to check the status of all configured alarms in real time. If the alarm is active, its setting will be displayed in red.

  For the **Safe Failure** alarm type, the causes of the alarm will be displayed in 2 ways: **1)** Single cause: Text; **2)** Multiple causes: Failure code. This code is in hexadecimal format and can be decoded according to the following table and example:

Bits 15...11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	SLAVE TCP	SLAVE RS485	REMOTE CHANNEL	ANALOG CHANNEL	FTP CLIENT	SDCARD	MAINFAIL	TASKFAIL	BROWNOUT	WATCHDOG	POWER ON

Table 1

Example:

Failure code: 0x0104

Decoding the above code into binary code results in the following value: 0000 0001 0000 0100

By relating the above values to the table, the example indicates the occurrence of the following failures: REMOTE CHANNEL and BROWNOUT.

4.2.2 GRAPH

This screen displays a graph, which allows you to select up to 6 channels:

Figure 48

At the bottom, you can select the channel to view and the color to link to it. When you select a channel, it will be read every 2 seconds (approximately), and its value will be displayed on the graph. If any channel is in an error status, the channel selection will be displayed in red.

4.3 DOWNLOAD

For information on the download process, see the **NXperience** manual.

5 COMMAND-LINE OPERATION

FieldLogger configurator has a new feature. It can now be called with command-line parameters, which means it can be used to download and export data by other applications.

It is advisable to create a batch file (.bat) to help organize the desired parameters. Once this batch file is done, it can be called (executed) from any other software to run the desired tasks, which include:

- Download data from the device
- Get data from a folder where data has already been downloaded to
- Export downloaded data to a fil
- Export data from the **FieldLogger** data base to a file
- Create a report with logged data

Some parameters are sent in the command-line, others are read by the INI files of the configurator. So, it is important that you perform the desired action with the configurator at least once before trying to make it by the command-line. It assures you that the method you are about to use works, and it saves the needed parameters in the INI file as well, so they can be used later.

The order of the parameters is important! Do not try to use them in any order but the one shown here. Parameters examples are shown inside quotation marks (" "), but there must be no quotation marks in the real file. In addition, each group must end with ";" and each group must be separated by a simple space.

5.1 BATCH FILE

The batch file is the same of those ancient DOS batch files.

In the file, we separate the parameters into 4 or 5 parts:

1. PART 1 – Implementation mode
2. PART 2 – Connection parameters
3. PART 3 – Download parameters
4. PART 4 – Export parameters
5. PART 5 – Special parameters for Mode 4

All parameters must be separated by ";" and each part must be ended with ";".

5.1.1 SYNTAX FOR CREATING THE BATCH FILE

- The file name cannot be longer than 8 characters, 1 period (.), and 3 other characters to define the file extension ("bat"). No accents or spaces should be used.

  Example: MyColect.bat
- The file is created in any text editor.
- There will be 4 (or 5, if you are using Mode 4) parameters separated by a space, as presented above. Each one means a step in the procedure. Within each of these parameters there are a group of sub-parameters, divided by a ";" and without spaces.

  Example: Lote1= 1;127.0.0.0;2;4;

Information taken from the configurator MUST BE previously configured before using the batch file (except for Mode 4).

5.1.2 PART 1: IMPLEMENTATION MODE

This mode contains only one parameter that defines what will be executed.

- Mode = 1 (Connect + Download + Export)
- Mode = 2 (Export)
- Mode = 3 (Download)
- Mode = 4 (Connect + Download + Create report + Save to PDF)

5.1.3 PART 2: CONNECTION PARAMETERS

This mode allows you to define the type of connection to be used and the parameters of each connection.

The connection mode can be:

- Communication mode = 1 (Ethernet)

  In this mode, the port and reconnection time will be read from the configurator "INI" file, which means that it should have been done before in the configurator "normal" operation. This option requires the IP address as an additional parameter. From software version 1.30 on, there must be also included port number and timeout in milliseconds. From software version 1.40 on, **FieldLogger** Modbus address (typically 255) must be included as well.

  Example:
  - IP (Example: 127.0.0.1)
  - Port (Example: 502)
  - Timeout (Example: 3000)
  - Address (Example: 255)

- Communication mode = 2 (USB)

  This mode requires the COM port as an additional parameter.

  Example:

  - Port (Example: 13)

- Communication mode = 3 (RS485)

  In this mode, the Baud Rate, parity, Stop Bits, and timeout will be read from the configurator "INI" file, which means that it should have been done before in the configurator "normal" operation. This option requires the COM port and the Modbus address as additional parameters.

  Example:

  - Port (Example: 13)
  - Address (Example: 1)

- Connection mode = 4 (Pen drive)

  This mode should be used when data was already downloaded from the **FieldLogger** through a USB flash drive, FTP or when reading directly from the device SD card (plugged in your computer). This option requires the folder path as an additional parameter (the folder with the device serial number must be included). Important: there must be no blank spaces in the path!

  Example:

  - Directory (Example: C:\Meudir\12341234)

When the chosen Action is "2" (export only, no download), the parameter "NULL" must be used to indicate that no connection is needed. **Example:** "NULL;"

5.1.4 PART 3: DOWNLOAD PARAMETERS

These parameters are directly related on the chosen implementation mode, as can be shown in Part 1 of this chapter.

- **Mode = 1 or Mode = 4**
  - Memory type: "0" for Flash memory, "1" for SD memory, and "2" for File.
  - Period: "0" for all and "X" for the number of days ago from the time of download.
  - Erase memory: "0" for no and "1" for yes.
  - Serial number.
  - Password: Download password. It the **FieldLogger** does not have a password, you must enter "NULL".
- **Mode = 2**
  - Serial number.
  - Period: "0" for all and "X" for the number of days ago from the time of download.
- **Mode = 3**
  - Memory type: "0" for Flash memory, "1" for SD memory, and "2" for File.
  - Period: "0" for all and "X" for the number of days ago from the time of download.
  - Erase memory: "0" for no and "1" for yes.
  - Password: Download password. It the **FieldLogger** does not have a password, you must enter "NULL".

5.1.5 PART 4: EXPORT PARAMETERS

These parameters are related to file export and reunite information about their format. The target folder for the exported files, as well as the "Author", "Company" and "Title" fields for the reports, are read by the configurator INI file, so be sure to set it in the software before trying to use it by the command-line.

- **File name:** File name with no extension or blank spaces.
- **Export type:**
  - PDF
  - CSV
  - DOC
  - XLS
  - FieldChart
  - Report
  - SQL
  - HTML
  - XLSX
  - DOCX
  - XML
- **Number of decimal places.** Valid values: 0 to 6.

When the chosen action is "3" (download only, no export), the parameter "NULL" must be used to indicate that no export is needed. **Example:** "NULL;"

5.1.6 PART 5: SPECIAL PARAMATERS FOR MODE 4

When Mode 4 is used, you must fill in the following parameters:

- **Number of channels:** From 1 to 8.
- **Name of channels:** The names must be identical to those configured in **FieldLogger**. They cannot have spaces, special characters, or accents. Each name must be separated by "/". Example: Boiler/Freezer/Charge1
- **Default download directory:** Path where the collected files should be saved. It cannot contain spaces. Example: C:\Downloads.
- **Path to preferences file of the report:** Path where the preferences file will be found. It cannot contain spaces. Example: C:\reports\MdRprt_1.rpf.

5.2 MODES

5.2.1 MODE 1

Automated download mode that connects to the device, perform data download and export data to a known format, such as .xls, .doc, .csv, etc. This mode requires that previous configuration be made in the configuration software.

5.2.2 MODE 2

Automated export mode. In this mode, the script does not connect or download from a device. It only exports the data, using an existing database, to a known format, such as .xls, .doc, .csv, etc. This mode requires that previous configuration be made in the configuration software.

5.2.3 MODE 3

Automated download mode that connects to the device and downloads its data for later use. This data is saved in a database previously configured in the software. This mode requires that previous configuration be made in the configuration software.

5.2.4 ITEMS THAT NEED TO BE PREVIOUSLY CONFIGURED FOR MODES 1, 2, AND 3

For these modes, the **FieldLogger** configurator must be preconfigured with the following parameters:

- Default download directory
- Author of the report
- Title of report
- Reporting company
- Default directory for burning exported files

5.2.5 MODE 4

This download mode by command-line performs the tasks of downloading, saving data, and generating PDF reports automatically. For the operation of this functionality, a further "batch" of data was added to the command line, besides the use of a preferences file for each report that one wishes to create. Here are instructions for completing this mode.

Some parameters are informed via the command line. Other parameters will be read from the new preferences file that must be created. Unlike others, this mode does not require a previously configuration of the configurator before to perform an automated download.

5.2.6 PREFERENCES FILE FOR MODE 4

This file is used to enter information about the report to be created. The image below shows what the body of this file should look like.

```
1  [Relatorio]
2  IdiomaAplicacao=PORTUGUESE
3  Destino=C:\Users\fsilva\Desktop\ReportTeste
4  ArquivoLogo=C:\Users\fsilva\Desktop\UsersKeys TP.png
5  Responsavel=Fabio Coelho
6  Titulo=Relatório Teste
7  Empresa=Novus
8  CasasDecimais=0.0
9  Description=Relatório teste Conaprole
10 Ylow=0
11 Yhigh=50
12 ReportType=2
13 ShowMKT=1
```

Figure 49

- **IdiomaAplicacao (Application language):** It allows you to enter the application language: PORTUGUESE, ESPANOL or ENGLISH.
- **Destino (Path):** It allows you to enter the file name and where it will be saved. No extension.
- **ArquivoLogo (Logotype file)**: It allows you to enter the path to the logotype image to be used in the report. Optional.
- **Responsavel (Author):** It allows you to inform the author's name.
- **Titulo (Title):** It allows you to inform a title for the report.
- **Empresa (Company):** It allows you to enter the company name.
- **CasasDecimais (Decimal Places):** It allows you to enter the number of decimal places to be used in the report.
- **Description:** It allows you to enter the text that will be shown in the report.
- **Ylow:** It allows you to enter the minimum Y-axis scaling of the graph. Optional.

- **YHigh:** It allows you to enter the maximum Y-axis scaling of the graph. Optional
- **ShowMKT:** It allows you to show the MKT calculation in the report. "1" to show and "0" to not show.
- **ReportType:** It allows you to enter the report type: "0" for one-page reports with minimum, average, and maximum values; "1" for averages reports; "2" for reports with chart and data table.

5.3 EXAMPLES

5.3.1 MODE 1: EXAMPLE

```
ECHO
SET Lote1=1;
SET Lote2=1;10.51.10.101;502;3000;255;
SET Lote3=0;0;0;12341234;NULL;
SET Lote4=FileName;2;3;

"C:\Program Files\FieldLoggerConfig\FLConfig.exe" %Lote1% %Lote2% %Lote3% %Lote4%
IF ERRORLEVEL 1 ECHO Invalid Parameters
exit
cls
```

5.3.2 MODE 2: EXAMPLE

```
ECHO
SET Lote1=2;
SET Lote2=NULL
SET Lote3=12341234;7;
SET Lote4=FileName;7;3;Fabio_Coelho;Report_Example;Novus;

"C:\Program Files\FieldLoggerConfig\FLConfig.exe" %Lote1% %Lote2% %Lote3% %Lote4%
IF ERRORLEVEL 1 ECHO Invalid Parameters
exit
cls
```

5.3.3 MODE 3: EXAMPLE

```
ECHO
SET Lote1=3;
SET Lote2=1;127.0.0.0;502;3000;255;
SET Lote3=0;0;0;111111;
SET Lote4=NULL

"C:\Program Files\FieldLoggerConfig\FLConfig.exe" %Lote1% %Lote2% %Lote3% %Lote4%
IF ERRORLEVEL 1 ECHO Invalid Parameters
exit
cls
```

5.3.4 MODE 4: EXAMPLE

```
ECHO
SET Lote1=4;
SET Lote2=2;13;
SET Lote3=0;2;0;11018291;NULL;
SET Lote4=FileName;7;3;
SET Lote5=5;teste/canal/caldeira/freezer/caminhão;C:\Users\fsilva\Desktop;c:\users\fsilva\desktop\MdReprt.rpf;

"C:\Program Files\FieldLoggerConfig\FLConfig.exe" %Lote1% %Lote2% %Lote3% %Lote4% %Lote5%
IF ERRORLEVEL 1 ECHO Invalid Parameters
exit
cls
```

For this mode, you must create a preferences file for the report.

6 FIELDLOGGER OPERATION

6.1 ANALOG INPUTS

FieldLogger has eight channels for reading analog variables. The types of accepted signals and sensors are the following: thermocouples J, K, T, E, N, R, S and B; RTDs Pt100 and Pt1000; 0 to 50 mV; 0 to 60 mV; 0 to 20 mV; -20 to 20 mV; 0 to 5 V; 0 to 10 V; 4 to 20 mA and 0 to 20 mA.

The accuracy of these types of signals is described in SPECIFICATIONS. The connection of these signals is described in CONNECTIONS AND INSTALLATION.

In these inputs, we use an analog/digital converter (A/D) with high resolution (24 bits) and accuracy. In the desired scan interval, all the analog channels enabled will be read. The ratio between the number of channels enabled and the scan time is limited up to 1000 readings per second. In other words, we can have one channel being read 1000 times per second, two channels being read 500 times per second and so on. So, the A/D converter will work faster to cope with the desired channels scan.

The A/D converter has the property of having a better signal-noise ratio when operating at low speeds (larger scan intervals), as well as better immunity to noise from the power grid and a higher effective resolution. In such way, to obtain better results in the analog inputs reading, it is strongly recommended to use the largest possible scan interval for the application. In the same thought, it is recommended to disable all the channels that are not necessary, because the increase in the number of enabled channels causes the A/D converter to work faster to cope with the scan rate configured by the user.

Each type of input signal has a valid range of measurement (detailed in SPECIFICATIONS). However, the device usually can measure signals that exceed the limits of this range. How far beyond the valid range will depend on the type of the configured input and may vary from device to device.

The following table outlines what to expect in the indication of **FieldLogger** depending on the signal applied at the input, for each type of configured input.

INPUT TYPE	CONDITION OF INPUT SIGNAL	INDICATION
Thermocouples: J, K, T, E, N, R, S and B	Within the range	Value read from input
	Open thermocouple	Value of configured error
	Just above the upper limit	Value read from input *
	A little below the lower limit	Value read from input *
	Far above the upper limit	Value of configured error
	Well below the lower limit	Value of configured error
Pt100 and Pt1000	Within the range	Value read from input
	Pt100/Pt1000 with one or more wires disconnected	Value of configured error
	Just above the upper limit	Value read from input *
	A little below the lower limit	Value read from input *
	Far above the upper limit	Value of configured error
	Well below the lower limit	Value of configured error
Voltage (mV): 0 to 50 mV, 0 to 60 mV, 0 to 20 mV and -20 to 20 mV	Within the range	Value read from input
	Disconnected signal	Value of configured error
	Just above the upper limit	Value read from input *
	A little below the lower limit	Value read from input *
	Far above the upper limit	Value of configured error
	Well below the lower limit	Value of configured error
Voltage (V): 0 to 5 V and 0 to 10 V	Within the range	Value read from input
	Disconnected signal	Value close to 1.8 V
	Just above the upper limit	Value read from input *
	A little below the lower limit	Value read from input *
	Far above the upper limit	Value of configured error
	Well below the lower limit	Value of configured error
Current (mA): 4 to 20 mA and 0 to 20 mA	Within the range	Value read from input
	Disconnected signal	4 to 20 mA: Value of configured error 0 to 20 mA: 0 mA
	Just above the upper limit	Value read from input *
	A little below the lower limit	4 to 20 mA: Value read from input *

INPUT TYPE	CONDITION OF INPUT SIGNAL	INDICATION
		0 to 20 mA: it is not possible to decrease besides the inferior limit
	Far above the upper limit	Value of configured error
	Far below the lower limit	4 to 20 mA: Value of configured error 0 to 20 mA: it is not possible to decrease besides the inferior limit

(*) Note: The indication of the analog channel is still a bit beyond the limits specified for the selected input type. However, in this condition, the accuracy is not guaranteed.

Table 2

There are 2 kinds of filtering in **FieldLogger**:

6.1.1 HARDWARE FILTER

This filter is usually "invisible" for the users. Analog to digital conversion always happens the faster it needs to satisfy all requirements for the number of enabled channels and the configured scan interval. If this conversion speed turns out to be 20 samples per second or slower, it will automatically filter the 50/60 Hz noise.

6.1.2 SOFTWARE FILTER

This is related to the "digital filter" configured on the ANALOG CHANNEL screen of the NXPERIENCE SOFTWARE. The digital filter implementation in this device makes that, on each new input signal sample (A/D conversion of a certain channel), the conversion outcome is updated with the difference between the previous result and the new sample value divided by the filter value.

The filter accepts values ranging from "0" to "20". The "0" value means that no filter must be used, and the conversion outcome is always the result from the last sample. From that filter value on, the bigger the value, the bigger the divider, which makes the new samples to have a smaller weigh in the outcome composition. Bigger filter values have the side-effect of slowing the response to actual changes in the input signal.

There is also the feature of inserting up to 10 points for each channel to correct distortions in the reading of these channels at these points. We call this feature **Custom Calibration** because it allows the user to adjust the indication at the desired points, resetting to zero the error on these points (see CUSTOM CALIBRATION). Between the inserted points, the adjustment is done linearly, depending on the entered values.

It is important to point out that the insertion of custom calibration points is optional, available only to those who want to adjust the indication with a local standard, because **FieldLogger** already comes fully calibrated from the factory.

Whenever you change the input type, make sure that the custom calibration points of the previous input are deleted!

For each channel, you must assign a unique name (tag), which will be used to reference the channel. You should also choose the type of input (sensor) that will be connected to that channel. In addition, you can assign the unit for the measured value: when temperature sensors (Pt100, Pt1000 or thermocouples) are involved you must choose between Centigrade (°C) and Fahrenheit degrees (°F); when linear sensors (voltage or current) are involved, you can type the desired unit.

In the case of linear input types, you should choose what the indication range of the channel is, in other words, what the channel must indicate when the input is at its minimum value and what it should indicate when at its maximum value (minimum and maximum values considering the working range of the **FieldLogger** for the selected input type). Example: selecting the 4-20 mA input type connected to a 0 to 2 bar pressure transmitter. In this case, we must choose the minimum value in the configuration of the input "0.0" and the maximum value "2.0". All the available resolution and accuracy will be contained in the selected range.

When using any device on analog inputs that are connected to the power network (for example: thermocouples or voltage simulator), it is recommended not to use the USB interface for communicating. In some cases, it was noticed the occurrence of noise and offsets in the readings due to the influence of the USB cable connection, probably due to ground loops.

6.2 DIGITAL INPUTS/OUTPUTS

FieldLogger has 8 digital channels that can be individually configured as input or output.

Channels configured as inputs, if enabled, may be logged in memory, used as alarm inputs and as operands of virtual channels. Channels configured as outputs are "open-drain" type and may be triggered by alarms or triggered via external Modbus commands.

The digital inputs have 2 (floating point) values associated with their logic states during the device configuration. Depending on the input's current logic state ("0" is equivalent to a low voltage or a closed contact in the input; "1" is equivalent to a high voltage or an open contact in the input), this corresponding value is used by the virtual channels, alarms, and logs.

The digital outputs configured to be controlled by the alarms cannot be triggered by external Modbus commands. Likewise, the outputs configured to be controlled by external Modbus commands cannot be used by the alarms.

The outputs controlled by external commands can be used, for example, as control outputs or alarm outputs of SCADA software or PLCs.

The digital inputs/outputs terminals are not isolated from the analog inputs' terminals!

Do not use analog and digital signals coming from the same voltage source; this will cause the device to malfunction.

6.2.1 COUNTINGS

From firmware version 1.10 on, it is possible to perform pulse counting at the digital inputs of the **FieldLogger**. Counting resolution is 32-bit, which means that each input can count from "0" up to (2^{32} – 1 =) "4294967295". When reaching the maximum value, next counted pulse will cause a roll-over and will reset the counting to 0.

The counting values can be accessed from Modbus registers (see the "FieldLogger – Modbus" document) and can also be copied to a virtual channel, where they can be logged to memory or used by the alarms, for example. When applying a new configuration in the digital channels, the counting's of all the digital channels are reset to 0.

Counting's can be zeroed by writing it in the related Modbus registers.

Counting values are persistent, which means they will remain even when **FieldLogger** is turned off. Whenever it is turned on again, counting is restarted from the values existent right before it was turned off.

6.3 OUTPUT RELAYS

FieldLogger has two relay type outputs (**RL1** and **RL2**) which can act as **Alarms** or **Digital Outputs**.

When configured as Alarms, they act according to the configuration adopted for the alarms. When configured as Digital Outputs, they are controlled remotely via Modbus commands (via RS485, USB or Ethernet).

6.4 RS485 INTERFACE

6.4.1 MAIN RS485

FieldLogger RS485 interface can be enabled or disabled. When disabled, it does not impact the traffic that may exist on the data bus. When enabled, it can be configured to operate as a Modbus RTU master or as a slave.

6.4.1.1 SLAVE

When operating as a Modbus RTU slave, the device makes the values of the channels available to be accessed by the master of the Modbus network, either being a PLC, a SCADA application, or any other device.

6.4.1.2 MASTER

When operating as a Modbus RTU master, it allows data from other devices on the bus to be read by **FieldLogger** and used in loggings, alarms, virtual channels or simply made available via another interface (Ethernet, for example).

From firmware version 1.10 on, it can work as a gateway between a Modbus TCP network and a Modbus RTU network. This way, all Modbus TCP requests received by **FieldLogger** with an identifier which is not "255" will be forwarded to the Modbus RTU network, assuming that this request was addressed to a slave located in this network. Modbus TCP requests received by **FieldLogger** with an identifier equals to "255" will be answered by **FieldLogger**.

The RS485 interface can be configured to operate in the following speeds (baud rates): 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. Moreover, it can be configured to operate with one or two stop bits, and in the parities: even, odd and none.

D1	**D**	**D+**	**B**	Bidirectional data line.	**Pin 50**
D0	**D**	**D-**	**A**	Inverted bidirectional data line.	**Pin 49**
C				Optional connection that improves communication performance.	**Pin 48**
GND					

Table 3

6.4.2 AUXILIARY RS485

FieldLogger auxiliary RS485 interface is in the DB9 connector which is located under **FieldLogger** cover. Its main function is to provide data to **FieldLogger** HMI, but from firmware version 1.20 on, it can be used as a generic interface, acting as a Modbus-RTU slave. Some further details can be seen in HMI (HUMAN-MACHINE INTERFACE).

As an interface to the HMI, it must be configured with baud rate 115200 bps, one stop bit and no parity.

As a generic interface, it can be configured to operate in the following baud rates: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. Besides that, it can be configured to operate with one or two stop bits and with even, odd and no parity.

D1	**D**	**D+**	**B**	Bidirectional data line.	**Pin 4**
D0	**D**	**D-**	**A**	Inverted bidirectional data line.	**Pin 8**
C				Optional connection that improves communication performance.	**Pin 7**
GND					

Table 4

6.5 REMOTE CHANNELS

Function not available in models:

1. **FieldLogger – USB, 512k logs, RS485 and**
2. **FieldLogger – USB, 512k logs, RS485, 24 V.**

FieldLogger can act as a master of a Modbus RTU network (please check the RS485 interface configuration), being able to read up to 64 registers from other devices (Modbus slaves) and using these registers as inputs in the virtual channels, alarms, and loggings. Each one of these registers read from other slaves is called a "remote channel".

The device starts reading the channels in the same order that they were created at configuration time. It goes on reading all channels, respecting the times between commands, until it has read them all. When the scan interval has elapsed, it restarts the readings of all remote channels. In case the channels scan takes longer than the settled scan interval, it restarts the channels reading immediately.

If the slave takes longer than the configured time out to respond, this will be considered a communication error. In the same way, response packets with invalid CRCs or with times between bytes greater than those specified in the Modbus standard will be considered communication errors. In the occurrence of a communication error in a remote channel reading, it tries again until the configured number of attempts has been reached. If the error persists, the error value configured for the channel is assumed.

The whole communication flow can be monitored by the **FieldLogger** Tx and Rx LEDs. Whenever a command is sent to a slave, the Tx led turns ON. When the slave responds to the command, the Rx led turns ON. This way, during a normal **FieldLogger** scan of some slaves, Tx and Rx LEDs should flash alternately as many times as the number of the configured remote channels.

From firmware version 1.50 on, it is possible to assign several decimal places directly to the remote channels. This way it is not necessary to use virtual channels to view a remote channel which has one or more decimal places. For example, reading a temperature from a Modbus slave that has a decimal place in integer format, in other words, it is multiplied by "10" (25.7 °C is read as 257), it is possible to read "25.7" directly. In older versions, you had to create a virtual channel that received this remote channel and divided its value by 10, restoring the original temperature value. This virtual channel could then be used to replace the remote channel for logging, alarms, or simple indication.

6.6 VIRTUAL CHANNELS

There are up to 128 virtual channels available in the **FieldLogger**. These channels are nothing more than channels whose values are the result of logical or mathematical operations. Depending on the selected operation, one or two operands will be necessary, defining that the operands are always other channels (analog, digital, remote, or even virtual). All operands are in the "floating point" format, which allows a greater precision in the calculations using several decimal places.

It is possible to "chain" several operations, causing the outcome of an operation to be the operand of another. If a channel used as operand is in an error condition, the resulting virtual channel will also assume its own error value. In other words, the error condition is passed on to the virtual channels that are dependent on a channel in error.

Constant values can be used through the "Constant" operation, where a value chosen by the user may be assigned to a virtual channel.

The following operations are available and can be used:

OPERATION	NUMBER OF OPERANDS	FUNCTION
Constant	1	Assigns a value to the virtual channel. **Example:** VC = 123.67
Addition	2	It receives the addition from 2 channels. **Example:** VC = C1 + C2
Subtraction	2	It receives the subtraction from 2 channels. **Example:** VC = C1 - C2
Multiplication	2	It receives the multiplication from 2 channels. **Example:** VC = C1 * C2
Division	2	It receives the division from 2 channels. **Example:** VC = C1 / C2
Logical "And"	2	It receives "0" if at least one of the 2 channels is 0. It receives "1" if both channels are different from 0.
Logical "OR"	2	It receives "0" if at least one of the 2 channels is not 0. It receives "0" if both channels are equal to 0.
Logical "Exclusive OR"	2	It receives "1" if only one of the 2 channels is not 0. It receives "0" if both channels are equal to 0 or if both are different from 0.
FloatToFloat	2	Transforms two 16-bit values in a "float". Typically used when reading 2 Modbus registers (remote channels) from another device that, when combined, represent a floating-point value.
Int32ToFloat	2	Transforms an integer 32-bit signaled value in a floating-point value. The allowable range for the integer value (operand) is from -16777215 to 16777215. Values outside this range will suffer truncation.
Square Root	1	Receives the square root from the operand (source channel).

OPERATION	NUMBER OF OPERANDS	FUNCTION
Exponentiation [4]	2	Receives the result from the first high channel to the second channel. **Example:** VC = $C1^{C2}$. The value of the exponent is limited to the range of -120 to 120.
Count [1]	1	Copies the current counting value of the selected digital channel (operand) to this virtual channel. It uses only 24 bits, which means it can count to "16777215" (most significant byte of the counting's ignored).
Variation [2]	1	At a configurable time rate (in seconds), shows the difference from the current channel value related to the previous time value. In other words, it has the value of the final time minus the value of the initial time. In case this operation is being performed on a digital channel, the variation of its counting will be computed.
Accumulation [2]	1	Receives the accumulated value of the selected channel. At a configurable time rate (in seconds), gets the current channel value and sums to the accumulated amount.
ByteInv FloatToFloat [3]	2	Transforms two 16-bit values into a "float". It works just like the **FloatToFloat** operation, but each register has its bytes swapped (byte high ↔ byte low) right before the operation. Useful when the register read from a slave has a byte order (endianness) which is opposite to the one expected by **FieldLogger**.
ByteInv Int32ToFloat [3]	2	Transforms an integer 32-bit signaled value into a floating-point value. It works just like the **Int32ToFloat** operation, but each register has its bytes swapped (byte high ↔ byte low) right before the operation. Useful when the register read from a slave has a byte order (endianness) which is opposite to the one expected by **FieldLogger**.
Sine[5]	1	Calculates the sine of an angle. Operating in degrees (Channel 1). **Example:** CV = Sen(C1)
Cosine[5]	1	Calculates the sine of an angle. Operating in degrees (Channel 1). **Example:** CV = Cos(C1)

Table 5

(1) Note 1: Available from firmware version 1.10 on.

(2) Note 2: Available from firmware version 1.20 on.

(3) Note 3: Available from firmware version 1.40 on.

(4) Note 4: Exponent of float type is only available from firmware version 1.72 onwards. For earlier versions, the exponent must necessarily be of integer type (the decimal part will be disregarded).

(5) Note 5: Available from firmware version 1.80 on.

When an error occurs on any channel (for example, disconnected sensor in an analog channel), its configured error value is set to it. If this channel is used as an operand of a virtual channel, its error value will be detected and the outcome of the virtual channel will be its own configured error value, leading to a propagation of the error values. One exception is the operation "Int32ToFloat", that does not propagate the error of the source channels (operands), because it is typically used in the conversion of two remote channels in a 32-bit value and, on this case, it would show an error every time one of the remote channels would have the error value (all values are valid – there is no error value that can be set outside of the valid range). The other exception is the "Accumulation" operation, that simply holds accumulation instead of displaying the error value.

Sample of chaining virtual channels to obtain more complex formulas:

As an example, we will be using the formula for calculating the flow measurement using an orifice plate, which is very popular in the industrial instrumentation environment. The formula is the following:

$$Q = K\sqrt{\frac{\Delta P}{\rho}}$$

Where Q = flow

ρ = flow density

ΔP = differential pressure

K = constant that makes the appropriateness of units and dimensions involved

In this case, we will be considering that the differential pressure (ΔP) will be read as an analog channel (ChAnalog_1 = ΔP), with the limits configured for measuring the correct unit.

In the **NXperience** software, we must enter the following virtual channels:

VC1 = K ("constant" operation with the numeric value of K)

VC2 = p ("constant" operation with the numeric value of p)

VC3 = ChAnalog_1 / VC2 ("Division" operation)

VC4 = √VC3 ("square root" operation)

VC5 = VC1 x VC4 ("multiplication" operation)

As a result, VC5 has the flow value of Q.

Virtual channel configuration examples for 32-bit registers:

Int32 type values:

Decimal Representation = 12000000

Hexadecimal representation = 0x00B71B00

DATA FORMAT IN THE SLAVE REGISTERS		VIRTUAL CHANNEL CONFIGURATION		
REG1	**REG2**	**Channel 1**	**Operator**	**Channel 2**
0x00B7	0x1B00	REG1	Int32ToFloat	REG2
0x1B00	0x00B7	REG2	Int32ToFloat	REG1
0xB700	0x001B	REG1	ByteInv Int32ToFloat	REG2
0x001B	0xB700	REG2	ByteInv Int32ToFloat	REG1

Table 6

Float type values:

Decimal Representation = 1234,12

Hexadecimal representation = 0x449A43D7

DATA FORMAT IN THE SLAVE REGISTERS		VIRTUAL CHANNEL CONFIGURATION		
REG1	REG2	**Channel 1**	**Operator**	**Channel 2**
0x449A	0x43D7	REG1	FloatToFloat	REG2
0x43D7	0x449A	REG2	FloatToFloat	REG1
0x9A44	0xD743	REG1	ByteInv FloatToFloat	REG2
0xD743	0x9A44	REG2	ByteInv FloatToFloat	REG1

Table 7

6.7 USB INTERFACE

FieldLogger has 2 USB interfaces: one USB device, used for the configuration, monitoring and data downloading, and one USB host, used for data download and firmware updates, when necessary.

6.7.1 USB DEVICE

The USB interface device is the preferential interface for the device configuration. It is the only interface that can never be disabled.

To access it, you should use the supplied USB cable. The "USB" led should light ON, indicating that the interface is ready for use (On first access, it may be necessary to install the USB drivers on your computer).

The communication on this interface is Modbus RTU, just like the slave interface RS485. The same commands and the same table of registers are available for this interface.

6.7.2 USB HOST

The USB host interface can be used to download the logged data from either the internal flash memory or the SD Card (it will download the memory currently configured), via a USB flash drive.

To perform the data download with a USB flash drive, this option must have been enabled on the **NXperience** software. Such being the case, when you connect the USB flash drive in the USB host interface, the "USB" led should light ON, indicating that it has been correctly recognized. From then on, the transference of data to the USB flash drive is started and the "USB" led starts to flash quickly. When all data have been transferred, the "USB" led stops flashing and stays lit on, indicating that the USB flash drive can now be removed.

In case data download with a USB flash drive has been disabled, nothing will happen when you plug in a USB flash drive in the USB interface, not even the LED will turn on!

There is an option to download all the memory and the option to download data of a limited number of days. In the second case, we can still choose if we want to download the earlier or later data. Example: when configuring the download of the most recent four-day data, the machine scans the memory and transfers to the USB flash drive all logging data found from the last four days that have some logged data.

Remarks:

- Use a USB flash drive with enough space for all data that must be downloaded (preferably an empty USB flash drive).

- Depending on the volume of data and the busy rate of the **FieldLogger** processor, the download may require a long period of time. To optimize data download via USB flash drive, try to download data more frequently and set the Download Period option for a few days, which will reduce the data volume in each download.

Throughout the **FieldLogger** operation, while not downloading data to the USB flash drive, all different types of errors generate 3 blinks on the status led. When the USB flash drive is connected for download and an error occurs, it is indicated as shown in the table below:

BLINK NUMBER	ERROR
1	Not enough space on the USB flash drive
2	File not found
3	USB flash drive disconnected before the end of the download
4	Writing error in USB flash drive
5	Error in creating the path in the USB flash drive
6	Error in the opening of the download session: there is already an open session (configuration or download)

Table 8

6.8 ETHERNET INTERFACE

Function not available in models:

1. **FieldLogger – USB, 512k logs, RS485 and**
2. **FieldLogger – USB, 512k logs, RS485, 24 V.**

The **FieldLogger** Ethernet interface allows the connection of the device in Ethernet 10/100 Mbps networks. There are several services available for this interface, all enabled and configured individually, which gives **FieldLogger** a high degree of versatility.

Note: It is always recommended that when not needing an interface or a service anymore, these should be disabled.

For connection to the TCP-IP network, we can set a fixed IP or use DHCP (dynamic host configuration protocol, the protocol that allows **FieldLogger** to have an IP number assigned by the network server). Furthermore, if desired, you can enable the DNS service, where in some services, instead of the IP number of the servers, you can configure your name (URL).

Note: FieldLogger does not have the processing capability of computers; therefore, the switch port where it is connected (especially in data center monitoring applications) configuration should be as basic as possible, avoiding exaggerated traffic in the network where it is. This will greatly increase its performance, avoiding potential packet losses.

The following services are available:

6.8.1 MODBUS-TCP

FieldLogger can be enabled to communicate through Modbus-TCP protocol, widely used in SCADA systems. By enabling that functionality, **FieldLogger** will act as a Modbus-TCP server, allowing channel values, as well as status and configuration information, to be read and written by external software applications. Enabling the gateway functionality (assuming the device is configured as a Modbus-RTU/RS485 master), FieldLogger can forward Modbus requests to slaves connected to the Modbus-RTU network.

FieldLogger must be accessed by the ID "255". Any other ID will be assumed to be targeting a Modbus-RTU/RS485 slave through **FieldLogger** by the gateway function. On this case, if this functionality is enabled, the data packet will be forwarded to the RS485 bus.

From firmware version 1.40 on, FieldLogger also supports the "Modbus RTU over TCP" protocol.

The gateway functionality works as a multiplexer, receiving all the simultaneous Modbus-TCP connections and putting them in a line to be sent to the Modbus-RTU network, one at a time. Due to this line, in a worst-case scenario, Modbus-TCP client's timeout should be set as big or bigger than Modbus-RTU master timeout (configured in REMOTE CHANNELS) multiplied by the number of the active Modbus-TCP connections. If the client Modbus-TCP timeout is smaller than that value, a timeout must occur instead of an exception code "0Bh" (Gateway Target Device Failed to Respond) in a missing slave situation.

6.8.2 SENDING EMAILS - SMTP

FieldLogger can be configured to send emails to multiple receivers upon the occurrence of alarms or events. The receivers must have been entered in the device through configuration. In alarms configuration, we must select which receivers should be notified via email on the occurrence of each alarm or event.

For this service, you must configure an email server that will be accessed at the time of the alarm occurrence. On this server, there obviously must be a valid email account for the **FieldLogger** to login.

Note: Secure connections (TLS/SSL) are not supported.

In the email message bodies, there are 2 parts: a constant part that is common to all messages, and a variable part that is dependent on the alarm or event that originated the message. The constant part is defined by the user, as well as the message subject. The variable part contains the **FieldLogger** tag, its serial number and a string describing the alarm condition or event that caused the message.

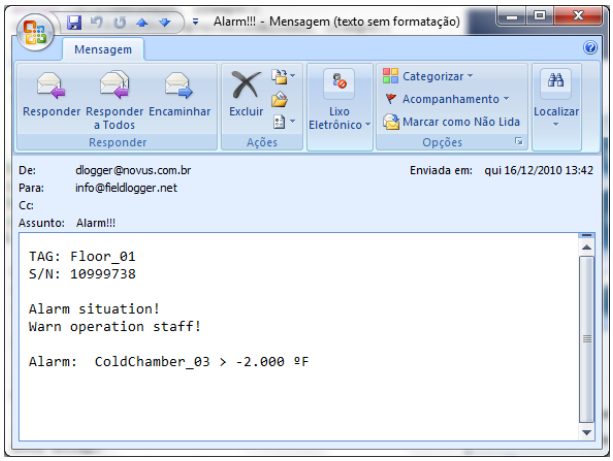


Figure 50

Emails do not have delivery or reading guarantees. Thus, in alarm cases, try to use some other ways to report alarm or events.

6.8.2.1 DEBUGGING

In case of any problems during the start-up of the device when sending emails, you can use a Telnet client to view messages that may help you to identify the problem. All you need do is point the Telnet client to the **FieldLogger** IP address (port 23) and observe the messages posted.

6.8.3 WEB PAGES - HTTP

FieldLogger has the capacity to serve web pages. It has three unchangeable and always available pages with basic information about channels, configuration, status, and alarms. Besides that, from firmware version 1.30 on, can serve custom web pages, which are hosted in the SD card.

⚠	**Due to a FieldLogger hardware limitation, there is no support to HTTPS (secure HTTP).**

6.8.3.1 STANDARD PAGES

FieldLogger has the capacity to serve three unchangeable web pages: one that has basic information about the enabled channels, one that has configuration information and device status and one with configured alarms information. There are two possible formats for these pages: HTML or XML. These pages include simple HTML code and can be viewed with most popular browsers. They are reloaded automatically at a configurable rate. Pages in XML format have the same information as that found in the related HTML pages, in a standardized format (format descriptions are available in separate documents), which allows the external usage of this data, increasing application flexibility (a typical application is the generation of a customized HTML page with data read from the XML page). There is no automatic reload of XML pages.

To ease information access, it is possible to use links to get directly into the desired webpage and to the most interesting information, as the following table shows.

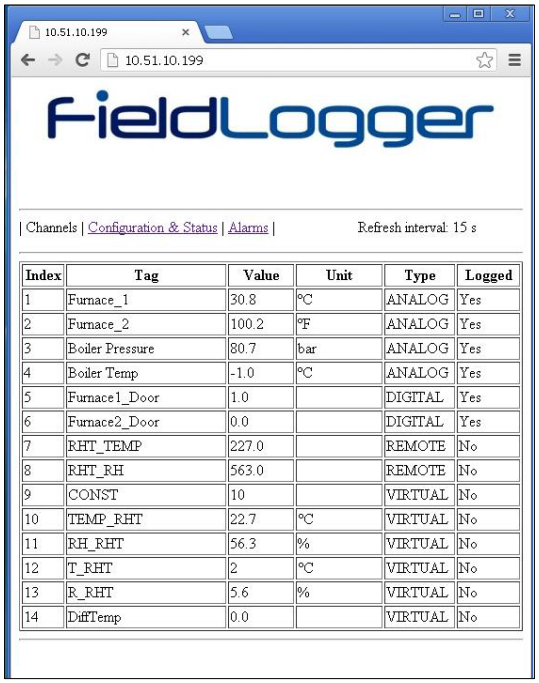
WEB PAGES	HTML	XML
Channels	FIELDLOGGER_IP/channels.html	FIELDLOGGER_IP/channels.xml
Configuration and Status	FIELDLOGGER_IP/status.html	FIELDLOGGER_IP/status.xml
Alarms	FIELDLOGGER_IP/alarms.html	FIELDLOGGER_IP/alarms.xml

Table 9

If only the IP address of the device is typed in the browser, the channels HTTP page will be loaded.

⚠	**Pages in XML format and the links with direct access to FieldLogger pages are available only from firmware version 1.10 on.**

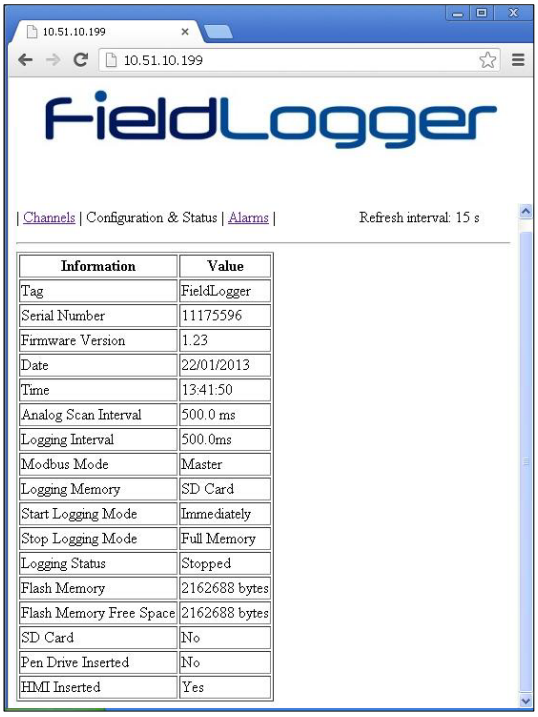
The channels information page reports, for each enabled channel in the device, its tag, its current value, its unit, its type (analog, digital, remote, or virtual) and if this channel is enabled for logging.



Index	Tag	Value	Unit	Type	Logged
1	Furnace_1	30.8	°C	ANALOG	Yes
2	Furnace_2	100.2	°F	ANALOG	Yes
3	Boiler Pressure	80.7	bar	ANALOG	Yes
4	Boiler Temp	-1.0	°C	ANALOG	Yes
5	Furnace1_Door	1.0		DIGITAL	Yes
6	Furnace2_Door	0.0		DIGITAL	Yes
7	RHT_TEMP	227.0		REMOTE	No
8	RHT_RH	563.0		REMOTE	No
9	CONST	10		VIRTUAL	No
10	TEMP_RHT	22.7	°C	VIRTUAL	No
11	RH_RHT	56.3	%	VIRTUAL	No
12	T_RHT	2	°C	VIRTUAL	No
13	R_RHT	5.6	%	VIRTUAL	No
14	DiffTemp	0.0		VIRTUAL	No

Figure 51

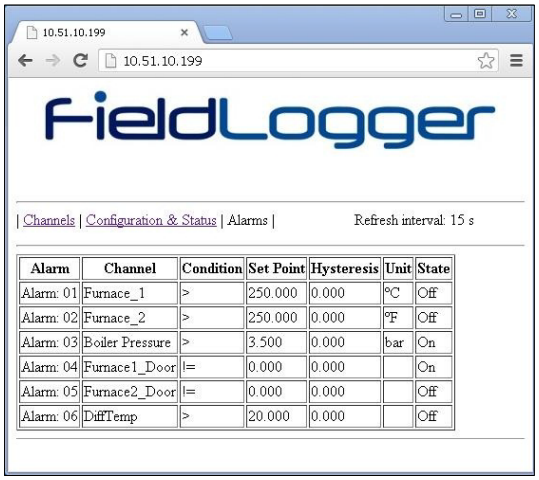
The configuration and status information page has the status of some parameters, as well as serial number, firmware version and other information from the device.



Information	Value
Tag	FieldLogger
Serial Number	11175596
Firmware Version	1.23
Date	22/01/2013
Time	13:41:50
Analog Scan Interval	500.0 ms
Logging Interval	500.0ms
Modbus Mode	Master
Logging Memory	SD Card
Start Logging Mode	Immediately
Stop Logging Mode	Full Memory
Logging Status	Stopped
Flash Memory	2162688 bytes
Flash Memory Free Space	2162688 bytes
SD Card	No
Pen Drive Inserted	No
HMI Inserted	Yes

Figure 52

Alarm information page shows, for each enabled alarm, its index, the related channel, the condition, the setpoint, hysteresis, the unit (if configured) and if the alarm is activated (ON) or not (OFF).



Alarm	Channel	Condition	Set Point	Hysteresis	Unit	State
Alarm: 01	Furnace_1	>	250.000	0.000	°C	Off
Alarm: 02	Furnace_2	>	250.000	0.000	°F	Off
Alarm: 03	Boiler Pressure	>	3.500	0.000	bar	On
Alarm: 04	Furnace1_Door	!=	0.000	0.000		On
Alarm: 05	Furnace2_Door	!=	0.000	0.000		Off
Alarm: 06	DiffTemp	>	20.000	0.000		Off

Figure 53

6.8.3.2 CUSTOM PAGES

Pages in XML format and the links with direct access to FieldLogger pages are available only from firmware version 1.10 on.

FieldLogger can serve web pages that are hosted in its SD card, since they are in the "webserv" folder. These pages can use much information directly from **FieldLogger**, like channels values and units, clock time and serial number.

WORKING CONDITIONS

- All needed files must be placed inside the "webserv" folder in the SD card.
- **FieldLogger** works in the "8.3" file name format, so no file should have more than 8 characters (plus the 3 of the extension).
- Files with markers that should be replaced by **FieldLogger** information need their names to start with the "_" character. Otherwise, these markers will not be replaced. Files which are not started with this "_" will not be parsed and will be served faster.
- Subfolders are accepted, though the maximum path length must be limited to 60 characters top (including the "webserv" folder and the "/" characters).
- If a valid marker is used with a channel or alarm that does not exist (example: analog channel 130), an "ERROR" message is returned.
- If a valid marker is used with a channel or alarm that is disabled, a "DISABLED" message is returned.
- Whenever you want to display the character "%", it is advisable to use it doubled "%%". It will assure that it will not be misinterpreted with a marker start.

MARKERS

A lot of information from **FieldLogger** is available to be inserted in the custom web pages. It is done using alphanumeric markers that, when found in the file, are replaced by their related value. Again, this will only happen in the files whose names start with a "_" (example: "_data.htm").

These are all available markers:

MARKER	REPLACEMENT INFORMATION
%ANALOG___001.T%	Analog channel 1 tag.
%ANALOG___001.V%	Analog channel 1 current value.
%ANALOG___001.U%	Analog channel 1 unit.
%ANALOG___001.D%	Analog channel 1 diagnostics information value: • Bit 0: Open sensor. • Bit 1: Shorted Pt100/Pt1000. • Bit 2: Out of limits. • Bit 3: Pt100/Pt1000 open cable.
...	...
%ANALOG___008.T%	Analog channel 8 tag.
%ANALOG___008.V%	Analog channel 8 current value.
%ANALOG___008.U%	Analog channel 8 unit.

MARKER	REPLACEMENT INFORMATION
%ANALOG___008.D%	Analog channel 8 diagnostics information value: • Bit 0: Open sensor. • Bit 1: Shorted Pt100/Pt1000. • Bit 2: Out of limits. • Bit 3: Pt100/Pt1000 open cable.
%DIGITAL__001.T%	Digital channel 1 tag.
%DIGITAL__001.V%	Digital channel 1 current value.
%DIGITAL__001.U%	Digital channel 1 unit.
...	...
%DIGITAL__008.T%	Digital channel 8 tag.
%DIGITAL__008.V%	Digital channel 8 current value.
%DIGITAL__008.U%	Digital channel 8 unit.
%REMOTE___001.T%	Remote channel 1 tag.
%REMOTE___001.V%	Remote channel 1 current value.
%REMOTE___001.U%	Remote channel 1 unit.
...	...
%REMOTE___064.T%	Remote channel 64 tag.
%REMOTE___064.V%	Remote channel 64 current value.
%REMOTE___064.U%	Remote channel 64 unit.
%VIRTUAL__001.T%	Virtual channel 1 tag.
%VIRTUAL__001.V%	Virtual channel 1 current value.
%VIRTUAL__001.U%	Virtual channel 1 unit.
...	...
%VIRTUAL__128.T%	Virtual channel 128 tag.
%VIRTUAL__128.V%	Virtual channel 128 current value.
%VIRTUAL__128.U%	Virtual channel 128 unit.
%ALARM__001.STS%	Alarm 1 tag.
%ALARM__001.SPT%	Alarm 1 status: "ON" (active alarm) or "OFF" (not active).
%ALARM__001.CND%	Alarm 1 configured Setpoint.
%ALARM__001.UNI%	Alarm 1 configured condition: ">", ">=", "<", "<=", "==" or "!=".
%ALARM__001.HYS%	Alarm 1 configured unit.
...	...
%ALARM__032.TAG%	Alarm 32 tag.
%ALARM__032.STS%	Alarm 32 status: "ON" (active alarm) or "OFF" (not active).
%ALARM__032.SPT%	Alarm 32 configured Setpoint.
%ALARM__032.CND%	Alarm 32 configured condition: ">", ">=", "<", "<=", "==" or "!=".
%ALARM__032.UNI%	Alarm 32 configured unit.
%ALARM__032.HYS%	Alarm 32 configured hysteresis.
%INFO.IN.FLTAG_%	FieldLogger tag.
%INFO.IN.SERIAL%	Serial Number.
%INFO.IN.FWVER_%	Firmware version.
%INFO.IN.MBMODE%	Main RS485 interface Modbus mode: "Disabled", "Master" or "Slave".
%INFO.IN.MBADDR%	Main RS485 interface Modbus address.
%INFO.IN.D_TYPE%	Selected logging memory: "Flash" or "SD Card".

MARKER	REPLACEMENT INFORMATION
%INFO.IN.START_%	Start logging mode selected: "Immediately", "Date/Time", "Alarm" or "Via Modbus Only".
%INFO.IN.STOP__%	Stop logging mode selected: "Full Memory", "Circular Memory", "Date/Time" or "Alarm".
%INFO.IN.FDRVIN%	USB flash drive connection status: "Yes" (connected) or "No" (disconnected).
%INFO.IN.SDSIZE%	SD card memory capacity, in kbytes.
%INFO.IN.SDFREE%	SD card available free memory, in kbytes.
%INFO.IN.INSIZE%	Internal flash memory capacity, in bytes.
%INFO.IN.INFREE%	Internal flash available free memory, in bytes.
%INFO.IN.ANLSCN%	Analog channels scan interval, in ms.
%INFO.IN.LOGPER%	Logging interval, in ms.
%INFO.IN.LOGSTS%	Current logging status: "Logging" or "Stopped".
%INFO.IN.HMICON%	HMI connection status: "0" (no HMI) or "1" (HMI connected).
%INFO.RTC.YEAR_%	Internal calendar year.
%INFO.RTC.MONTH%	Internal calendar month.
%INFO.RTC.DAY__%	Internal calendar day.
%INFO.RTC.HOUR_%	Internal clock hour.
%INFO.RTC.MIN__%	Internal clock minute.
%INFO.RTC.SEC__%	Internal clock second.

Table 10

There are examples available for download on our website.

6.8.4 FILE TRANSFER - FTP

The **FieldLogger** provides both FTP client and server for the transference of logged data files.

6.8.4.1 CLIENT

FTP client is used to make programmed downloads of logging data. It can be done once a day, in a determined time, or many times a day, in a determined time interval (this option requires firmware version 1.50 or higher). To use, simply configure the username and password to be used in server login and set the daily download start time or the interval between them, depending on if the "periodic" download (more than once a day) option is enabled. In case of downloading at periodic intervals, start download moments are always referred to midnight (example: 4-hour interval will turn out to downloads at 00:00, 04:00, 08:00, 12:00, 16:00 and 20:00). If, at the time configured for starting the download, a problem with the connection arises, **FieldLogger** will retry for 30 minutes.

From firmware version 1.40 on, there is the possibility to download data directly in the CSV (comma-separated values) format. When selecting this option for the client FTP download, the files created in the download folder will be in the CSV format instead of the binary (standard) format.

The created CSV file will have the device tag in its first row. In the second row, there will be the column names. From the third row on, there are the dates, times, and channel values for each log. If the logging rate was configured in milliseconds, this extra column will be placed before the values columns. All fields are between quotation marks (") and the field separator used is the semi-colon (;).

Due to the extra processing to transform the files to the CSV format, this option may cause a delay in the download.

Note: Due to a limited size of **FieldLogger** internal buffer, it is recommended that FTP server welcome message should have sentences no longer than 60 characters (each sentence must necessarily be finished with "car return" and "new line" characters, or "0Dh" and "0Ah" in hexadecimal).

6.8.4.2 SERVER

To use FTP server on **FieldLogger**, simply enable this feature in the **NXperience** software and set up a username and password. In this way, the user will have access to the data files for read-only purposes.

The standard used for the FTP is the "Unix" type.

Note: There is no support for simultaneous connections. Therefore, the used client will have to be configured to just use one connection. In addition, the client must be configured to use the passive mode.

Due to a FieldLogger hardware limitation, there is no support to FTPS (secure FTP).

6.8.5 NETWORK MANAGEMENT - SNMP

All the enabled channels for reading, as well as several status information, have their values available via SNMP protocol (read-only, there is no writing implemented on the device via SNMP). **FieldLogger** implements SNMPv1 protocol version.

Standard MIBs are not available in FieldLogger. Available SNMP information is listed below, besides the following OIDs of the "System" group:

- **sysDescr**
- **sysObjectID**
- **sysUpTime**

The available MIB is described below. A MIB file is available in our website. The branch "Enterprise" of MIB is used, where the "Enterprise number" is **34590**. The sub-branch assigned to **FieldLogger** is **1**. Thus, all information will have the OID starting in **1.3.6.1.4.1.34590.1**.

Here follows the available OIDs:

- 1.3.6.1 .4.1.34590.1.0 = Device tag. [OCTET STRING]
- 1.3.6.1 .4.1.34590.1.1 = Device serial number. [OCTET STRING]
- 1.3.6.1.4.1.34590.1.2 = Firmware version. [OCTET STRING]
- 1.3.6.1 .4.1.34590.1.3 = Device date year. [INTEGER]
- 1.3.6.1.4.1.34590.1.4 = Device date month. [INTEGER]
- 1.3.6.1.4.1.34590.1.5 = Device date day. [INTEGER]
- 1.3.6.1.4.1.34590.1.6 = Device time hour. [INTEGER]
- 1.3.6.1.4.1.34590.1.7 = Device time minute. [INTEGER]
- 1.3.6.1.4.1.34590.1.8 = Device time second. [INTEGER]
- 1.3.6.1.4.1.34590.1.9 = RS485 interface enabling and configuration (0 = disabled; 1 = master; 2 = slave). [INTEGER]
- 1.3.6.1.4.1.34590.1.10 = Selected logging memory (0 = internal memory; 1 = SD card). [INTEGER]
- 1.3.6.1.4.1.34590.1.11 = Logging status (0 = logging stopped; 1 = logging in progress). [INTEGER]
- 1.3.6.1.4.1.34590.1.12 = Available memory for logging in internal memory, in bytes. [INTEGER]
- 1.3.6.1.4.1.34590.1.13 = Presence of SD card (0 = no card; 1 = card present). [INTEGER]
- 1.3.6.1.4.1.34590.1.14 = Available memory for logging in the SD card, in kilobytes. [INTEGER]
- 1.3.6.1.4.1.34590.1.15 = Presence of USB flash drive (0 = no USB flash drive; 1 = USB flash drive present). [INTEGER]
- 1.3.6.1.4.1.34590.1.16 = Presence of HMI (0 = no HMI; 1 = HMI present). [INTEGER]
- 1.3.6.1.4.1.34590.1.17 = Total number of enabled channels. [INTEGER]
- 1.3.6.1.4.1.34590.1.18.X.Y
  - X = Sequential channel number, starting with analog channels and followed by the digital, remote, and virtual ones:
    - X = 1: First channel enabled.
    - X = 2: Second channel enabled.
    - etc.
  - Y = Channel information:
    - Y = 1: Channel tag. [OCTET STRING]
    - Y = 2: Channel read value. Wherever appropriate (virtual and analog channels, for example), it will be multiplied by the number of decimal places configured by the user. [INTEGER]
    - Y = 3: Channel unit. [OCTET STRING]
    - Y = 4: Channel type indication (analog, digital, etc.), and the channel number related to this type. Examples: ANALOG_001, DIGITAL_005, REMOTE_014, VIRTUAL_103. [OCTET STRING]
    - Y = 5: Channel error indication (0 = channel ok; 1 = channel in error state). [INTEGER]
    - Y = 6: Indicates that the channel is enabled (value = 1) or disabled (value = 0) for logging. [INTEGER]

6.8.5.1 TRAPS

Traps will be generated if enabled and if selected in the alarms to warn about their occurrence. The address and port of destination are configurable.

They have the following fields:

- Community: "**FieldLogger**".
- OID: .1.3.6.1.4.1.34590.1.100 (where **34590** is the "enterprise number" used and the **1** following is the branch of **FieldLogger**).
- General number: 6 (indicating that the trap is "enterprise-specific").
- Specific number: Index for the occurred alarm (zero - based).
- Trap value
  - OID: .1.3.6.1.4.1.34590.1.100.XX, where XX is the index of the occurred alarm.
  - Type: OCTET STRING.

- Value: String indicating the occurred alarm in the format: TAG + CONDITION + VALUE. Example: "Channel_1 > 129.43"

6.8.6 CLOUD VIA MQTT

From firmware version 1.90, **FieldLogger** is compatible with the MQTT protocol (versions 3.1 and 3.1.1), but without secure connection support and thus without TLS. It is therefore possible to use it to connect the device to the cloud.

Although **FieldLogger** is compatible with different types of clouds, it uses the default URL of NOVUS Cloud, NOVUS' cloud storage service. The type of cloud can be changed according to the user's needs.

	**1. To use the cloud data publishing service, you need an Internet connection.** **2. NOVUS Cloud is a paid and optional service. You can find more information on the NOVUS website.**

6.8.6.1 PREREQUISITES

To send data to the cloud, **FieldLogger** must meet the following conditions:

1. You must have an Internet connection with rules that allow the device to access the Broker. In this case, it may be necessary to check firewalls and network policies.
2. In the **NXperience** software, enable the MQTT protocol and configure all the connection parameters appropriately (see the MQTT section of the NXPERIENCE SOFTWARE chapter). The DNS service must also be enabled.
3. Log data to the internal memory. Data recorded on the SD card is **NOT** sent via MQTT.
4. Set the interval between logs to be greater than or equal to 1 second (regardless of whether the device is configured with the **Circular Memory** or **Full Memory** option). Logging must be slower than the system's capacity (**FieldLogger** + network) to send data.
5. When using **NOVUS Cloud**, enable the cloud service in the portal and, during the first connection, configure a **FieldLogger MQTT** device with the serial number of the **FieldLogger** used. For more information on how to do this, see the **NOVUS Cloud** manual, available on our website.
6. When using other MQTT-compatible cloud services or generic Brokers, configure the subscription and publication topics on the platform appropriately.

	**NOVUS Cloud activation expires in 24 hours. If FieldLogger has made its first connection within this period, it will continue to work. A new activation will not be necessary.** **If it fails to make its first connection within this period, a new activation will be required.**

6.8.6.2 OPERATION

Once enabled, the connection to the Broker starts from **FieldLogger**. Once the device has connected to the cloud and has data in its memory, it will start sending this data to the cloud.

NOVUS Cloud has a template that displays the **FieldLogger's** declared variables. These variables can be adjusted. In addition, the cloud will show a sample dashboard, displaying variables that allow you to check whether the device was successful in sending data.

All channels recorded in the internal memory will be sent to the cloud via MQTT. The internal memory serves as a buffer to hold the data and can be read locally using any other download method. In the event of a possible disconnection for an indefinite period, the data from that period will be stored in the internal memory and sent to the cloud as soon as the connection returns.

Periodically, **FieldLogger** will check for new data to send to the cloud. Typically, this period is equal to the logging interval. If there is data to transmit, all available data will be sent, and the device will return to wait for a new transmission time.

Data will be transmitted if there is data to transmit. Therefore, even if **FieldLogger** is stopped logging when memory is full (if configured in **Full Memory** logging mode), it will be able to transmit data that has not yet been sent.

The data must be sent to the cloud with the times normalized to UTC (Universal Time Coordinated). It is therefore necessary to configure the device's time zone. This information must be provided at the time of configuration.

6.8.6.3 PUBLICATION AND SUBSCRIPTION TOPICS

FieldLogger uses 3 predefined, non-editable topics to identify messages:

- **Log:** Used to publish the data generated by the device.
- **Command:** The device receives (subscribes to) commands via this topic. The result of executing the command is published in the **Command Ack** topic.
- **Command Ack**: The device publishes the result of the commands executed in this topic.

PUBLICATION

The publication topic will be fixed:	**novus/fieldlogger/<SN>/log**

Channel data is published periodically in JSON format and has the following key/value sets:

```
{"<tag>": {
   "pck": "1/1",
   "timestamp": value,
   "channel": {
         "channel_name_xx":  value,
              "channel_name_xx ": value
        }
  }
}
```

Where:

- **channel_name:** Fixed+numeral values can be displayed as they were enabled.
- **ANL:** Analog channels.
- **REM:** Remote channels.
- **DIG:** Digital channels.
- **VRT:** Virtual channels.

Division of inbound packages:

- **Pck:** Current package value / total packages. Each packet contains a maximum of 8 channels. A log structure with 32 channels will have a final pck value of 4.
  Example: pck 4/4

EXAMPLE OF A 34-CHANNEL PUBLICATION

The publication is divided into 5 independent packages.

As the example below shows, the timestamp is the same for all parts of the package, as it is the same publication.

➢ **Package 1**

```
{
    "tag_fieldLogger_test": {
        "pck": "1/5",
        "timestamp": 1719583317,
        "channel": {
            "ANL1": 18.202476,
            "ANL2": -1.0,
            "ANL3": -1.0,
            "ANL4": 18.726448,
            "ANL5": -1.0,
            "ANL6": -1.0,
            "ANL7": -1.0,
            "ANL8": -1.0
        }
    }
}
```

- **Package 2**


```
{
    "tag_fieldLogger_test": {
        "pck": "2/5",
        "timestamp": 1719583317,
        "channel": {
            "DIG1": 1.0,
            "DIG2": 1.0,
            "DIG3": 1.0,
            "DIG4": 1.0,
            "DIG5": 1.0,
            "DIG6": 1.0,
            "DIG7": 1.0,
            "DIG8": 1.0
        }
    }
}
```


- **Package 3**


```
{
    "tag_fieldLogger_test": {
        "pck": "3/5",
        "timestamp": 1719583317,
        "channel": {
            "VRT001": 10,
            "VRT002": 28,
            "REM01": 65535.0,
            "REM02": 0.0,
            "REM03": 0.0,
            "REM04": 0.0,
            "REM05": 0.0,
            "REM06": 0.0
        }
    }
}
```


- **Package 4**


```
{
    "tag_fieldLogger_test": {
        "pck": "4/5",
        "timestamp": 1719583317,
        "channel": {
            "REM07": 0.0,
            "REM08": 0.0,
            "REM09": 0.0,
            "REM10": 0.0,
            "REM11": 0.0,
            "REM12": 0.0,
            "REM13": 0.0,
            "REM14": 0.0
        }
    }
}
```

➢ **Package 5**

```
{
    "tag_fieldLogger_test": {
        "pck": "5/5",
        "timestamp": 1719583317,
        "channel": {
            "REM15": 0.0,
            "REM16": 0.0
        }
    }
}
```

6.8.6.4 COMMANDS

FieldLogger can receive commands to configure the setpoints of the 32 possible alarms. Boolean values can be set for the digital outputs and relays 1 and 2 can be activated, according to the device model.

Once **FieldLogger** has connected to the Broker, it must subscribe to the topics below to receive commands:

✓ **novus/fieldlogger/<SN>/command**
✓ **novus/fieldlogger/<SN>/ack/command:** Displays the "echo" responses of the command received.

This action is executed without user intervention.

JSON object to send commands:

```
{
    "timestamp": 1585819567,
    "desired": {
        Output object {
        < Key: value >
                    }
    }
}
```

Where the output object refers to:

- **alarm**: Alarm Setpoints.
- **output**: Digital outputs.
- **relay**: Relay outputs.

COMMAND KEYS

COMMAND	DESCRIPTION
Alarm Setpoints	The key is composed of **setpXX**, where **XX** is a numeric value from 1 to 32, corresponding to the alarm Setpoint. **Example:** "setp1".
Digital outputs	The switch is composed of **outXX**, where **XX** is a numeric value from 1 to 8, corresponding to the digital output. **Example:** "out7".
Relay outputs	The key is composed of rlyXX, where XX is the numeric value from 1 to 2. **Example:** "rly2".

Table 11

COMMAND EXAMPLES

Example of a command sent to alarm Setpoints 1 to 5:

```
{
    "timestamp": 1585819567,
    "desired": {
        "alarm": {
            "setp1": 20.67,
            "setp2": 33.05,
            "setp3": 15.08,
            "setp4": 16.01,
            "setp5": 17.02
        }
    }
}
```

Example of writing to digital outputs:

```
{
    "timestamp": 1585819567,
    "desired": {
        "output": {
            "out1": 1,
            "out2": 0,
            "out3": 1,
            "out4": 1,
            "out5": 0,
            "out6": 1,
            "out7": 1,
            "out8": 1
        }
    }
}
```

Example of writing to relays:

```
{
    "timestamp": 1585819567,
    "desired": {
        "relay": {
            "rly1": 0,
            "rly2": 0
        }
    }
}
```

Example of writing to digital outputs and relays:

```
{
    "timestamp": 1585819567,
    "desired": {
        "output": {
            "out1": 1,
            "out2": 0,
            "out3": 1,
            "out4": 1,
            "out5": 0,
            "out6": 1,
            "out7": 1,
            "out8": 1
```

```
        },
        "relay": {
            "rly1": 0,
            "rly2": 0
        }
    }
}
```

NOTE ON COMMAND FORMATTING

The order of the keys does not influence how they are carried out. It is possible, for example, to send commands to the Setpoint of the digital outputs as follows:

```
{
    "timestamp": 1585819567,
    "desired": {
        "output": {
            "out6": 1,
            "out1": 1,
                                "out4": 1,
            "out2": 0,
            "out3": 1,
                                "out8": 1,
            "out5": 0,
            "out7": 1,
        }
    }
}
```

INTERPRETING THE RESPONSE TO THE COMMAND RECEIVED

The responses to the commands will be displayed in the topic: **novus/fieldlogger/<SN>/ack/command**, where **SN** is the serial number.

```
{
    "device_id": <FieldLogger tag>,
    "timestamp": <timestamp returned from command sent>,
    "reported": {
        " error": <status code>
    }
}
```

Possible status values:

STATUS CODE (ERROR)	VALUE
0	Command successfully carried out.
1	Error when recording the alarm set.
2	Inconsistent value or invalid field format.
4	Error when recording digital outputs.
8	Inconsistent value or invalid field format.
16	Digital output in use or inactive.
32	Error when recording the relay.
64	Inconsistent value or invalid field format.
128	Relay in use.
256	Generic error in JSON format.

Table 12

Example:

command *= {"device_id":"teste","timestamp":1585819567, "reported":{" error":0}}*

If you send a command to several outputs and an error occurs in only one of them or if one of them is in use, the configuration of all the other outputs will be carried out even though the return (error) indicates a problem.

6.8.6.5 DEBUGGING

In case of any problems during the start-up of the device in the cloud, you can use a Telnet client to view messages that may help identify the problem. All you must do is point the Telnet client to the **FieldLogger** IP address (port 23) and observe the messages posted.

6.9 DATA LOGGING AND DOWNLOAD

Data logging can be done in the internal memory of **FieldLogger** or, optionally in a SD card (not included), which must be inserted into the proper compartment behind the cover (or HMI), as shown in **Figure 20**. The internal memory capacity is up to 532,480 loggings, while SD card logging capability will depend on its storage capacity (size), assuming that the card is empty.

Poor quality of the card can compromise data logging, causing missed logging periods at higher logging rates or being more susceptible to data write corruption. Therefore, it is always recommended the usage of high-speed (class 4 or higher) SD cards of established brands.

SD Card logging not available in models:

1. **FieldLogger – USB, 512k logs, RS485 and**
2. **FieldLogger – USB, 512k logs, RS485, 24 V.**

Any kind of channels (analog, digital, remote, and virtual ones) can be written into the memory. Logging is periodic and has its period (rate) set through the **NXperience** software. After each time interval, the current values of the configured channels are logged in the selected memory.

There are several ways to start and stop the loggings, and many of them can be freely combined. During the logging, all the selected channels will be recorded in the requested memory (internal flash or SD card) based on the interval between loggings configured.

Depending on the start and stop modes selected, there may be "snippets" of logs in memory, and therefore, periods without any data recorded. This is fully compatible with the device and means no problem at all.

The start modes available are the following:

- **Immediate start:** Loggings will be started right after the device reconfiguration.
- **By date/time:** Loggings will be started on the configured date/time.
- **By alarm:** When the alarm condition is met (alarm active), loggings will be started.
- **Only by Modbus commands:** Loggings will be started only when a specific command is sent to the Modbus device.

The stop modes available are the following:

- **Full memory:** The loggings will be stopped only when the selected memory (internal flash or SD card) has no more available space.
- **Never stop (circular memory):** Loggings will be continually done in the selected memory. When filling up the memory, the oldest data are erased so that the most recent data may be saved.
- **By date/time:** Loggings will be started on the configured date/time.
- **By alarm:** If loggings were started by the same alarm, when the alarm condition is no longer met (alarm inactive), loggings are stopped. If loggings were started for some other reason, as soon as the alarm condition takes place (alarm active), loggings will be stopped.
- **By Modbus command:** A specific Modbus command can be sent to stop loggings. This command is independent of the selected stop mode and has priority over it.

The following possibilities to start and stop loggings are available:

AVAILABLE COMBINATIONS		LOGGING START			
		IMMEDIATE	DATE/TIME	ALARM	ONLY MODBUS COMMAND
Logging stop	Full memory	Yes	Yes	Yes	Yes
	Circular memory	Yes	Yes	No	Yes
	Date/Time	Yes	Yes	No	No
	Alarm	Yes	Yes	Yes	No

Table 13

Here are operation details for each of these modes:

- **Immediate start and stop when memory get full:** As soon as the new configuration is applied, loggings are started. The **FieldLogger** follows on logging until filling up the memory (internal or SD card). In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Immediate start and never stops (circular memory):** As soon as the new configuration is applied, loggings are started. The **FieldLogger** goes on logging forever, overwriting the oldest data as soon as the memory has no more room for new logs. In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Immediate start and stop by date/time:** As soon as the new configuration is applied, loggings are started. The **FieldLogger** goes on logging until the arriving of the date and time set. In the case of a power failure, logging is interrupted, returning as soon as power returns. If the memory fills up before the date/time set, **FieldLogger** will stop logging.
- **Immediate start and stop by alarm:** As soon as the new configuration is applied, loggings are started. The **FieldLogger** goes on logging until an alarm condition takes place (configured alarm for loggings stop). When leaving alarm condition, loggings return to be done. In the case of a power failure, logging is interrupted, returning as soon as power returns. When memory fills up, **FieldLogger** will stop logging.

- **Start by date/time and stop when memory gets full:** Once the date/time set for loggings start has arrived, loggings will start. **FieldLogger** goes on logging until filling up the memory (internal or SD card). In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Start by date/time and never stops (circular memory):** Once the date/time set for loggings start has arrived, loggings will start. **FieldLogger** goes on logging forever, overwriting oldest data as soon as the memory has no more room for new logs. In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Start by date/time and stop by date/time:** Once the date/time set for loggings start has arrived, loggings will start. **FieldLogger** goes on logging until the arriving of the date and time set. In the case of a power failure, logging is interrupted, returning as soon as power returns (in case the stop date/time has not yet elapsed). If the memory fills up before the set-up date / time, **FieldLogger** will stop logging.
- **Start by date/time and stop by alarm:** Once that the date/time set for the loggings start has arrived, loggings will start. **FieldLogger** goes on logging until an alarm condition takes place (configured alarm for loggings stop). When leaving alarm condition, loggings return to be done. In the case of a power failure, logging is interrupted, returning as soon as power returns. When memory fills up, **FieldLogger** will stop registering.
- **Start by alarm and stop when memory gets full:** When an alarm condition takes place (configured alarm for logging start), logging is started (it will not start if, upon applying the configuration, the alarm condition is already satisfied: it must exit the alarm condition and get into it again.) **FieldLogger** goes on logging until filling up the memory (internal or SD card). In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Start by alarm and stop by alarm:** When an alarm condition takes place (configured alarm for logging start), logging is started (it will not start if, upon applying the configuration, the alarm condition is already satisfied: it must exit the alarm condition and get into it again). **FieldLogger** goes on logging while this alarm condition is met. When leaving the stop alarm condition, logging is stopped. In the case of a power failure, logging is interrupted, returning as soon as power returns. When memory fills up, **FieldLogger** will stop logging.
- **Start only by Modbus command and stop when memory gets full:** When the Modbus command for the logging start is sent to the device (more details about this command in the document "FieldLogger – Modbus" and in the chapter NXPERIENCE SOFTWARE), logging will start. **FieldLogger** goes on logging until filling up the memory (internal or SD card). In the case of a power failure, logging is interrupted, returning as soon as power returns.
- **Start only by Modbus command and never stops (circular memory):** When the Modbus command for the logging start is sent to the device (more details about this command in the document "FieldLogger – Modbus" and in the chapter NXPERIENCE SOFTWARE), logging will start. **FieldLogger** goes on logging forever, overwriting the oldest data as soon as the memory has no more room for new logs. In the case of a power failure, logging is interrupted, returning as soon as power returns.

	**Modbus commands to start and stop loggings, when enabled, can be sent at any time, and have precedence over the configured start and stop modes. In such a way, if a Modbus command to start loggings is sent to the device, they will start up, no matter the state of the device in relation to the configured modes (the only exception is a configured mode different from "circular memory" with the memory already full).** **Similarly, when sending a Modbus command to stop loggings, they will stop regardless of the mode configured for start and stop. If a stop command has been sent and then there is a start situation (as per the configured mode), loggings will not occur, because the command has higher priority than the modes. It is worth to point that the sent command persists even after a power outage. When sending a new logging configuration, however, the previously sent commands are "forgotten".**

When logging is done in the SD card, it is generated a folder structure whose purpose is to organize the logging data and to ease access for the data download software to such data. The following figure shows an example of this structure:

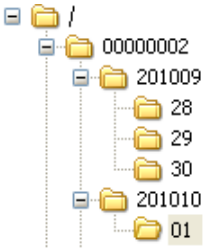


Figure 54

In the root folder it is created a folder whose name is the serial number of the device. Inside this folder, in its turn, are generated other folders whose name are the year and month in which the logging began. And inside of the year-months folders, other folders are generated whose names are the days in which the loggings start up. The data files are stored within these last folders and their names are made up by the information related to hour, minute, second and hundredths of second of the first log in the file, followed by the extension "fl". For example, the file /00000002/201010/01/18243516.fl was generated by **FieldLogger** with the serial number 00000002 on October 01, 2010, and the first log contained in this file was held on this day at 18 hours, 24 minutes, 35 seconds and 16 hundredths.

The files generated by the registration process are composed of a header followed by periodic data blocks and *timestamps* (time markers). The header carries information relevant to the logs from that file, essential for the correct interpretation of its data. In the data block are stored values for each channel that is being logged, in floating point format (IEEE 754) with simple precision, where each log occupies 4 bytes. The timestamps are used as time reference to identify the moment in which the channels were logged. Periodic timestamps serve to ensure the integrity of the loggings in time, even under power failure or fault in the SD card.

When the logging is done in internal memory, the file format is the same as when it is written on the SD card. However, by the size of the available memory, all data will be contained in one sole file.

In its internal flash memory, **FieldLogger** has something about 2M bytes that turns into an approximated value of 512k (512000) loggings. The exact loggings number depends a lot on the enabled channels number for loggings and on the configured logging interval. Logging intervals equal or bigger than 30 seconds record, on each logging instant, extra data to indicate the time when the log happened (timestamp), and that increases

memory consumption makes the total logging capacity decrease. Therefore, speaking of logging memory usage, the worst possible scenario is logging only one channel on slow (bigger than 30 s) intervals.

Data download consists of copying the data from the internal memory of **FieldLogger** or from the card to a computer. This process, when using the configuration software, is guided, and assisted. When done via USB flash drive, it is also simple and automatic. However, when performed manually, via an FTP client, for example, one should take care that the directory structure is kept, otherwise an error may come to occur in the interpretation of the data download.

It is not possible to download data through different interfaces at the same time. If a user starts a download through the Ethernet interface (FTP, for example) and another user tries to download via USB flash drive, this one will not be able to accomplish the download and must wait until the first download has been terminated.

Depending on the amount of data on this device, mainly when using high-capacity SD cards, the download process can take hours. In these cases, prepare to make the download through faster interfaces and, perhaps, in more likely times of the day.

The simultaneous data download while in logging process assumes that the download rate is faster than the logging rate. This is particularly important when using circular memory, because, otherwise, we could never download all the data (the logging, being faster, might eventually turn around, reaching and surpassing the download, which would lead to data inconsistencies). Such being the case, whenever the circular memory is used with high registration rates, you should try to use a faster interface for download.

6.10 ALARMS

32 alarms are available in each **FieldLogger**. Each alarm requires you to choose a channel, a condition, a set point, and hysteresis. When the alarm condition is met (example: Channel_1 > 45.0 °C), an event is generated for which different actions can be associated. For each selected channel, its current value is used in the comparison. In case of digital channels, we will use the values associated with 2 logical states. When one channel is in error, the configured error value will be used in the alarm. The value of the channel to be used will always be the floating-point value.

The available conditions are:

- **Greater than (>):** Alarm condition is met when the selected channel is greater than the Setpoint.
- **Greater than or equal (>=):** Alarm condition is met when the selected channel is greater or equal to the Setpoint.
- **Minor (<):** Alarm condition is met when the selected channel is minor than the Setpoint.
- **Minor than or equal (<=):** Alarm condition is met when the selected channel is minor or equal to the Setpoint.
- **Equal (==):** Alarm condition is met when the selected channel is equal to the set point. With this condition, hysteresis does not make much sense and should be maintained at "0.0". Analog channels can hardly have a value exactly equal to the set point value, so you should avoid this condition with analog channels, because it will be an alarm condition that might never be satisfied.
- **Different (!=):** Alarm condition is met when the selected channel is different from the set point. With this condition, hysteresis does not make much sense and should be maintained at "0.0". Analog channels can hardly have a value exactly equal to the set point value, so you should avoid this condition with analog channels, because it will be an alarm condition that might always be satisfied.
- **Safe Failure:** This special alarm condition is not linked to a specific channel. It serves to report internal failures, analog and digital circuit failures, and **FieldLogger** Modbus communication failures, such as:
  - SLAVE TCP: **FieldLogger** is configured as a slave in a TCP network and has not received any requests from the master in the time interval determined in the **Timeout for Slave TCP** parameter.
  - SLAVE 485: **FieldLogger** is configured as a slave on a 485 network and has not received any requests from the master in the time interval determined in the **Timeout for Slave RTU** parameter.
  - REMOTE CHANNEL: **FieldLogger** is configured as a master in a 485 network (i.e., remote channels are configured) and a communication timeout has occurred with one of the devices.
  - ANALOG CHANNEL: Sensor failure.
  - FTP CLIENT: Service failure.
  - SDCARD: SD card failure.
  - MAINFAIL: Internal failure.
  - TASKFAIL: Program execution failure.
  - BROWNOUT: Power failure. Power supply momentarily dropped to critical levels.
  - WATCHDOG: Failure during program execution. **FieldLogger** was automatically restarted to recover.
  - POWER ON: Power failure. The power supply has been interrupted.

The **Safe Failure** alarm type must be linked to a relay output. It cannot produce other events or enable another type of digital output.

In this case, the relay output will operate with inverted logic, that is, the relay will be enabled in normal operation and will be disabled during the incidence of an alarm. If there is a power failure, for example, the relay remains disabled and will be recognized as a failure by the rest of the system.

Available actions are:

- Activating the relays (they must have been configured to "activation by alarm" in the software).
- Activating the digital outputs (they must have been configured to "activation by alarm" in the software).
- Loggings start: When reaching the alarm condition, starts loggings if they are not already logging.
- Loggings stops: When reaching the alarm condition, stops loggings if they are not already stopped.
- Loggings start and stop: Logs while the alarm condition is met.
- Sending emails: Allows sending an email reporting the alarm condition to up to 10 receivers (they must have been enclosed in the receivers list of the device).
- Sending SNMP traps: sends a trap to the configured server with a number regarding the index of the alarm.

Sending emails depends on the availability of TCP connections on the part of the **FieldLogger** (it has a limit of simultaneous connections, see SPECIFICATIONS). Sending traps depends on the availability of UDP connections on the part of the **FieldLogger** (it has a limit of simultaneous connections, see SPECIFICATIONS).

Just one alarm can start loggings, and just one alarm can stop them. In case you configure the start and/or stop of the logging for alarms, there is no option of "circular memory", that is, loggings will terminate as soon as the loggings available memory gets full.

The alarm condition must stay at least 250 ms to ensure that it is detected. In the same way, the output of the alarm condition must remain the least 250 ms to guarantee that it is detected. Besides, you should remember that the analog channels can have a delay in the measurement of the true input value, depending on the sensor type, on the interval between readings and the configured filter, among others. The remote channels, in turn, depend directly on the set scan interval. Thus, these types of channels can offer an additional latency in the detection of the alarm condition.

When powering **FieldLogger** up, there are no reliable values in the channels. In case of the analog and remote channels, you should wait for the first scan to be done for valid values to be available. So, alarms that use these kinds of channels may take a while until they are ready to compare with the configured set point. Particularly when using remote channels, a complete scan may take several seconds.

7 DATA COMMUNICATION

FieldLogger has several communication interfaces. Among them, we may highlight some that can be used as Modbus slaves:

- RS485, acting as a slave on the Modbus RTU protocol.
- USB device, acting as a slave on the Modbus RTU protocol.
- Ethernet, acting as a server in the Modbus TCP protocol.

All the configuration of the device, as well as the reading of the data of the inputs, are done via Modbus protocol.

In the document "**FieldLogger** - Modbus" (available on our website) all the information needed to read data from the input channels is found without using the **NXperience** software.

To ensure proper configuration and data download of **FieldLogger**, always use the software. After configuring the device, data of its inputs and/or outputs can be accessed by any other software capable of Modbus RTU or Modbus TCP communication.

8 HMI (HUMAN-MACHINE INTERFACE)

The HMI (Human-Machine Interface) is available as an accessory to the **FieldLogger**. Several features are implemented in this device, such as enabled channels monitoring, viewing these channels in chart mode, alarms monitoring, status checking and configuration of some basic **FieldLogger** operation parameters.

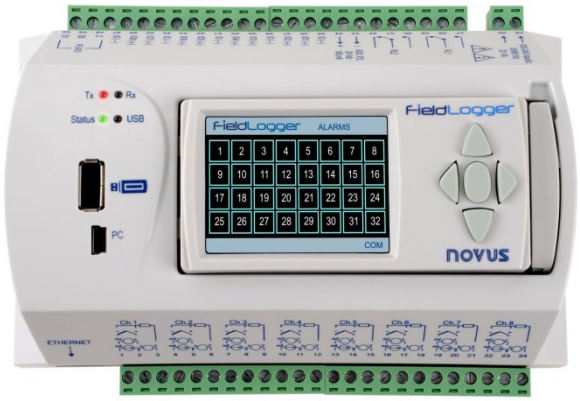


Figure 55

The HMI is attached to the **FieldLogger** through a DB9 connector located under its cover. The **Figures 10** and **11** demonstrate how the IHM is connected to **FieldLogger**.

Both power and serial communication go through the DB9 connector, so it is the only necessary connection. There is also the possibility of using the IHM remotely, attached to the **FieldLogger** through an extension cable. As the communication between IHM and **FieldLogger** is done through RS485 using a baud rate of 115200 bps what puts a limit to the HMI operation distance is the power, what means that the voltage drop in the power cables (both positive and negative cables summed) must be 0.4 V top. Considering a current consumption of 80 mA, it makes the maximum cable resistance to be 5 ohms. As cable resistance per meter depends mainly on its section, we recommend checking a wire resistance table. For some common cables, HMI's maximum operation distance is shown in the table below:

GAUGE	SECTION	MAXIMUM LENGTH
24 AWG	0.21 mm^2	29 m
22 AWG	0.33 mm^2	47 m
20 AWG	0.52 mm^2	75 m

Table 14

The figure below shows the pin-out needed to build an extension cable to the HMI:

Figure 56

PIN	SIGNAL	DESCRIPTION
1	-	-
2	+5 V	HMI power input: +5 Vdc
3	-	-
4	B / D1 / D+ / D	RS485 Tx/Rx positive data
5	-	-
6	-	-
7	GND	HMI power input: ground
8	A / D0 / D- / D\	RS485 Tx/Rx negative data
9	-	-

Table 15

HMI screen is divided into three sections: top bar, main area, and bottom bar. Top bar (**A**) shows the **FieldLogger** logo and the name of the current screen. Main area (**B**) shows the information regarded to that screen. Bottom bar (**C**) shows alarms; download progress and serial communication indication whenever necessary, as shown in the following image.

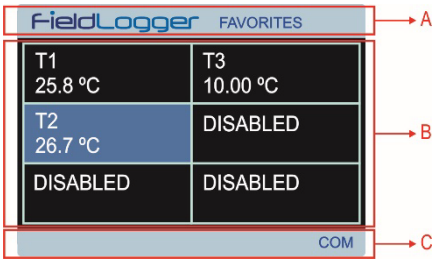
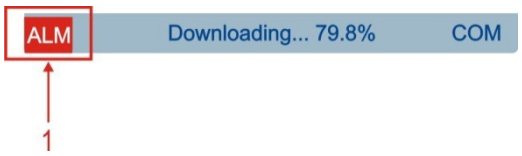
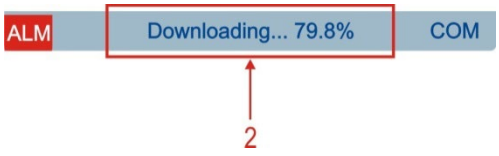
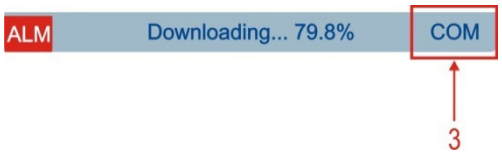


Figure 57

Bottom bar has the following information, as shown in the picture below:

	**Alarm events (1):** Flags that one or more alarms are active. To check which alarms are active, please go to the Alarms screen.
	**Download progress (2):** When plugging a USB flash drive to download **FieldLogger** data, there is a percentage indication of the download already done. If you download data from **FieldLogger** internal flash memory, it shows the percentage of the whole memory already downloaded. If downloading from the SD card, it shows the percentage of the current file being downloaded.
	**Communication in progress (3):** Informs that there is communication going on between the HMI and the **FieldLogger**. Typically, this flag must be on most of the time, varying from screen to screen. If this flag stays off and the HMI stops updating information, there probably is a communication problem between the HMI and the **FieldLogger**

Interfacing with this HMI is done through some buttons, which allow you to navigate between all the available screens, select and edit parameters. The following picture shows the keys and their functions.

Not all characters can be visualized on the HMI. Occidental characters for tags and units are recommended.

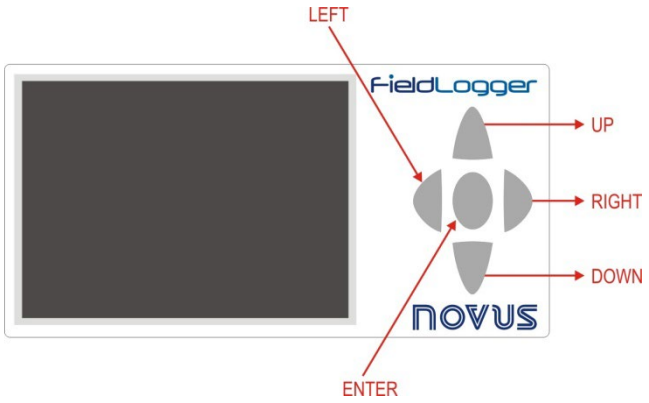


Figure 58

Navigation between the screens is accomplished through keys RIGHT and LEFT. There are six different screens, each one with a specific functionality, as follows:

8.1 "FAVORITES" SCREEN

This screen shows a six-position grid where you can assign a channel to be displayed on each position. Every enabled channel in the **FieldLogger** is available for assignment, that is done using UP and DOWN keys to select the desired position in the grid and ENTER to open a list of the available channels. After that, just use the UP and DOWN keys again to find the channel you want to display and ENTER to select it.

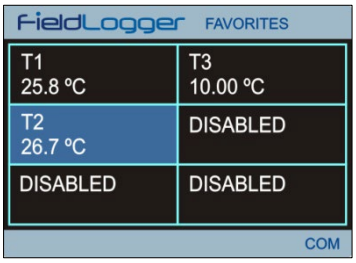


Figure 59

8.2 "CHART" SCREEN

This screen saves a log of the favorite channels' values. The period of this log, the limits of the channels displayed (vertical limits of the chart) and some other parameters can be configured in the "Configuration" screen. Pressing the ENTER key will bring a legend screen with the tags of the favorite channels that is on the chart and their associated colors. Pressing the ENTER key again will show the parameters configured for the chart. Pressing the ENTER key one more time will lead you back to the Chart screen. The red color is reserved to inform that one or more channels have values over the configured high limit or under the configured low limit for the chart.

The log starts when the HMI is powered and it is updated constantly, no matter which HMI screen is being watched. It is important to notice that this log is limited to the data that can be seen in the screen (100 points) and it is not possible to retrieve old (out-of-the-screen) data. It is also important to notice that this log has nothing to do with the one done by the **FieldLogger**.

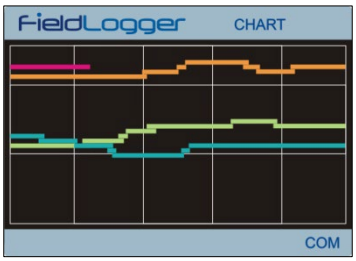


Figure 60

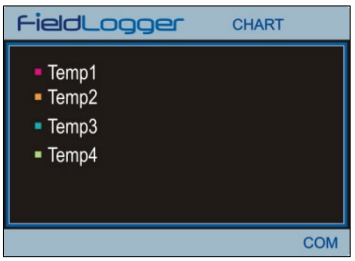


Figure 61

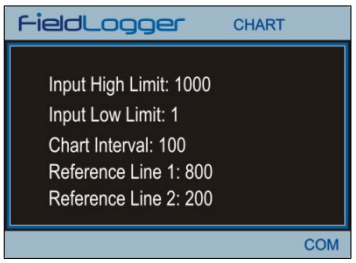


Figure 62

8.3 "CHANNEL LIST" SCREEN

This screen shows a list with all **FieldLogger** enabled channels (analog, digital, remote, and virtual). UP and DOWN keys are used to navigate between the channels. The ENTER key has no function on this screen.

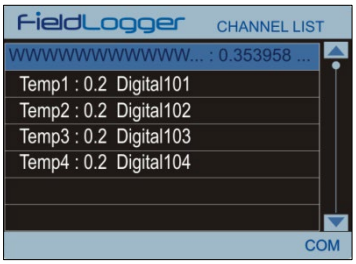


Figure 63

8.4 "ALARMS" SCREEN

The Alarms screen is 32-position grid where each numerated position is related to its equivalent alarm or event. Whenever an alarm is active, its number will be displayed in red on this screen.

In the left side of the bottom bar an alarm flag will be on as long as any of the **FieldLogger** enabled alarms are active. This flag can be seen in all screens. Keys UP, DOWN and ENTER have no function on this screen.

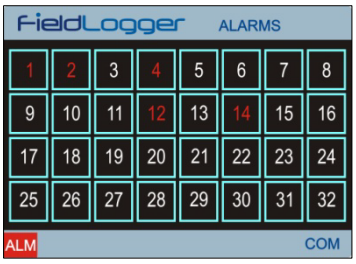


Figure 64

8.5 "STATUS" SCREEN

On this screen, a list is shown with information about **FieldLogger** and the HMI itself. Navigation is accomplished by the UP and DOWN keys and there is no function for the ENTER key.

Information is organized in a hierarchical way, where hierarchical levels are displayed by indentation on the left side. Parameters that are more on the right "belong" to the previous parameter (more on the left). For example, information about the Ethernet interface is placed in the "Ethernet" level, and that, by its turn, is placed in the "**FieldLogger**" level.

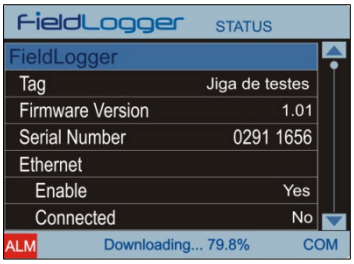


Figure 65

The following parameters can be monitored on this screen:

FieldLogger: ..... **FieldLogger** information
- Tag ..... Configured tag for the **FieldLogger**
- Firmware Version ..... **FieldLogger** firmware version
- Serial Number ..... **FieldLogger** serial number
- Ethernet ..... Ethernet interface information
  - Enabled ..... Yes/No if interface enabled/disabled
  - Connected ..... Yes/No if interface cable connected/disconnected
  - Cloud Status ..... Disconnected/Connected
- Data Logger: ..... Information about data logging
  - Scan Interval ..... Analog scan interval configured
  - Logging Rate ..... Logging rate configured
  - Logging Enable ..... Yes/No if logging is enabled/disabled
  - Storage Memory ..... Shows which memory is being used for data logging
  - SD Card Connected ..... Yes/No if SD card connected/disconnected
  - Logging Status ..... Indicates if logging is running or if it is stopped
  - Logging Mode ..... Logging mode information
    - Start ..... Indicates how logging is supposed to start
    - Stop ..... Indicates how logging is supposed to stop
- SD Free Memory ..... Shows the available memory for logging in the SD card
- Int. Free Memory ..... Shows the available memory for logging in the internal memory

HMI ..... HMI information
- Firmware Version ..... HMI firmware version
- Serial Number ..... HMI serial number

8.6 "CONFIGURATION" SCREEN

Some **FieldLogger** and HMI parameters can be changed on this screen. To edit a parameter, you should first select it by pressing the UP and DOWN keys. Once it is selected, press the ENTER key to put it into edit mode. When in edit mode, the UP and DOWN keys can be used to edit the parameter value (increment and decrement). When editing a parameter composed by more than one field (like IP addresses) you should select the correct field using the RIGHT and LEFT keys. Single field parameters that have a wide range have a customized edition way, where you can change the increment/decrement factor using the RIGHT and LEFT keys during edition: it starts with a factor of "1"; every time you press the LEFT key, it multiplies this factor for 10 (tens, hundreds, thousands, …); every time you press the RIGHT key, it divides this factor for 10 (thousands, hundreds, tens, …). Once you have the desired value, just press ENTER to get it out of the edit mode.

Just like the "Status" screen, information is organized in a hierarchical way. As an example, information about the Ethernet interface is placed in the "Ethernet" level, and that, by its turn, is placed in the "**FieldLogger**" level.

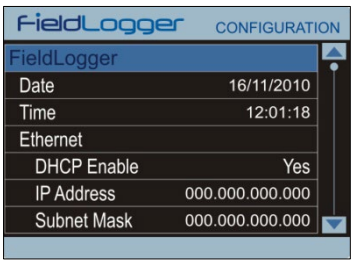


Figure 66

The available parameters of this screen are:

FieldLogger: ..... **FieldLogger** configuration
- Date ..... Date configuration
- Time ..... Time configuration
- Ethernet ..... Ethernet interface configuration
  - DHCP Enable ..... DHCP enabling
  - IP Address ..... IP address configuration
  - Subnet Mask ..... Subnet mask configuration
  - Gateway ..... Gateway address configuration
- Modbus RTU ..... Modbus RTU interface configuration
  - Mode ..... Operation mode: master/slave
  - Slave Address ..... Modbus slave address
  - Baud rate ..... Communication baud rate
  - Parity ..... Communication parity

HMI ..... HMI configuration
- Backlight Timeout ..... Time to turn the backlight off (0 = always on)*
- Chart ..... "Chart" screen configuration
  - Color Background ..... Background color: black/white
  - Interval ..... Period of time displayed on the Chart
  - Reference Value 1 ..... First reference line value**
  - Reference Value 2 ..... Second reference line value**
  - Input High Limit ..... Chart maximum displayed value***
  - Input Low Limit ..... Chart minimum displayed value***

* The maximum Backlight timeout is 48 seconds.

** For disabling the reference lines, just configure them to a value out of the input configured range.

*** If the Low Limit is configured to a value greater than the High Limit, the Chart will not display the values of the channels in a correct way!

9 DEVICE SOFTWARE (FIRMWARE) UPDATE

The **FieldLogger** allows the update of its embedded software (firmware) in the field, through a USB flash drive. The versions of this software for this device are available on the device web site. Following is the procedure for updating the software.

	**Before performing the update, proceed with the data download and, if desired, also the reading of the device's configuration and saving it to a file. Both the configuration and the logged data in flash will be lost in the upgrade process.**

1. Turn off and unplug all **FieldLogger** connectors attached to it.

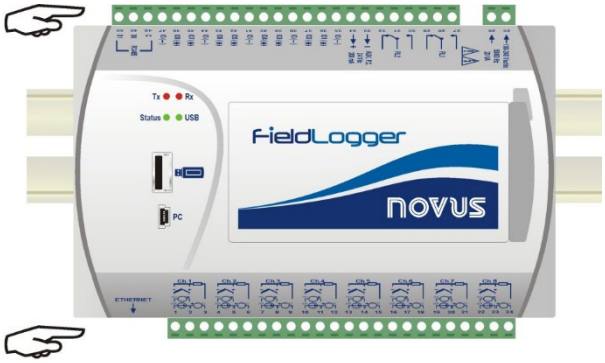


Figure 67

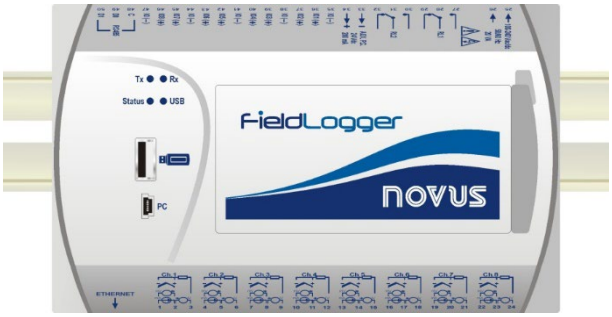


Figure 68

2. Copy the *flogger.flb* file, which is the binary firmware file of the **FieldLogger**, into USB flash drive root folder. This file can be downloaded from the product's website.
3. Insert the USB flash drive into the USB host of the **FieldLogger** (that must remain off).

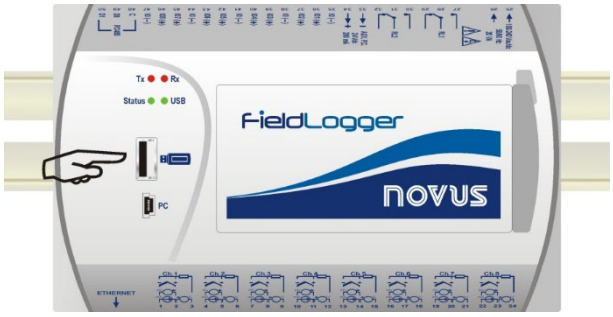


Figure 69

4. Replace the plug and powering cord of the device, then turn **FieldLogger** ON again. The status and USB LEDs should flash together, indicating that the boot loader is running.

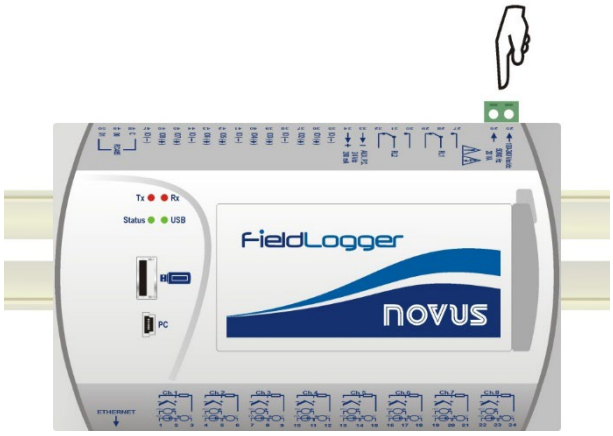


Figure 70

5. The status led should still be flashing in the same pattern. The USB led should flash quickly while the *flogger.flb* file copy is being performed. Once the copy has been done, the USB led should stay lit, indicating that the process has come to an end.

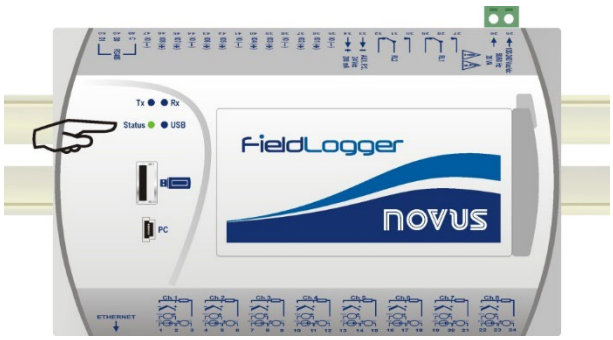


Figure 71

6. If during the copy (writing) of the firmware of **FieldLogger** there is an error, it will be indicated by the status led. The status led will stop flashing periodically and repeatedly, showing a pattern of blinking as per the occurred error:

BLINKS	MEANING
1	An error occurred while reading *flogger.flb* file.
2	Error when initializing the **FieldLogger** USB host (when inserting the USB flash drive)
3	Error when erasing the **FieldLogger** program memory.
4	Error when writing the **FieldLogger** program memory.
5	The written firmware is larger than the program memory available in **FieldLogger**. The writing was partial.
6	Error when closing *flogger.flb* file.
7	Invalid *flogger.flb* file.

Table 16

7. Remove the USB flash drive from the USB host of the **FieldLogger**.

   FieldLogger should reset itself automatically and may be used normally, now with the new updated firmware.

   We recommend deleting from the USB flash drive the firmware file (*flogger.flb*) so that no undesired updates in the future occur.

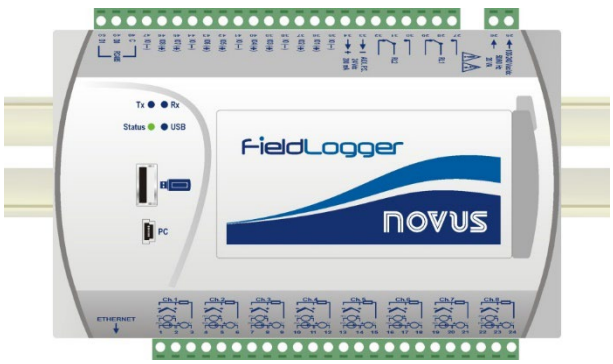


Figure 72

10 CLOCK BATTERY REPLACEMENT

FieldLogger clock is kept by an internal battery if the device is not powered by the AC mains power. In case of discharge of this battery, logged dates and times may not represent real values. **FieldLogger**, whenever detecting such a situation, informs it by continuously flashing the Status led three times in a row (check FLAGS (LEDS)).

⚠	**Replace battery with Panasonic lithium battery Part No. CR 2032 only.** **Usage of another battery may present a risk of fire or explosion.**

Replacement procedure is described as follows:

- Detach **FieldLogger** from DIN rail.

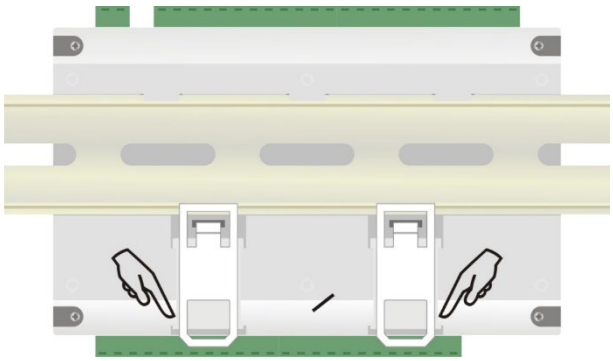


Figure 73

- Disconnect all terminals carefully.

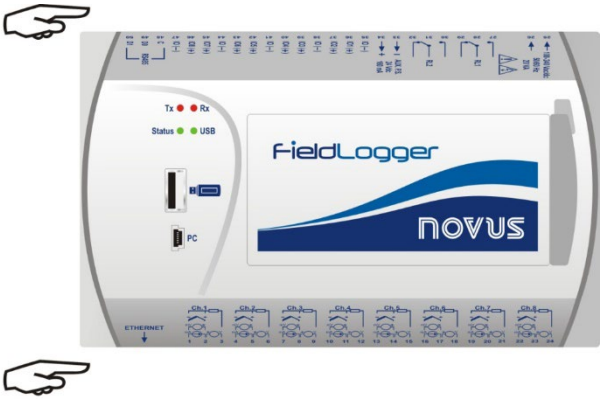


Figure 74

- Unscrew the four screws located under the enclosure. Remove the back cover, taking care not to remove the circuit board.

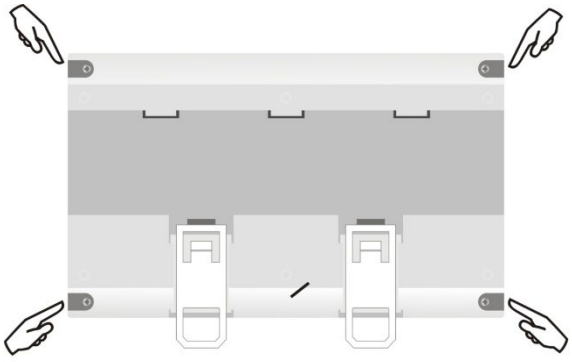


Figure 75

- After removing the back cover, remove the battery. Avoid touching the circuit board!

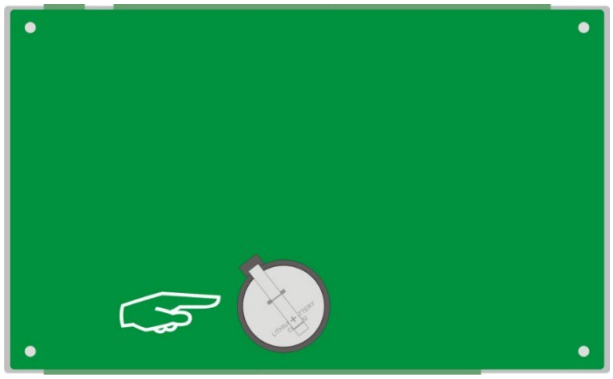


Figure 76

- Insert the new battery and put back the back cover and its 4 screws.

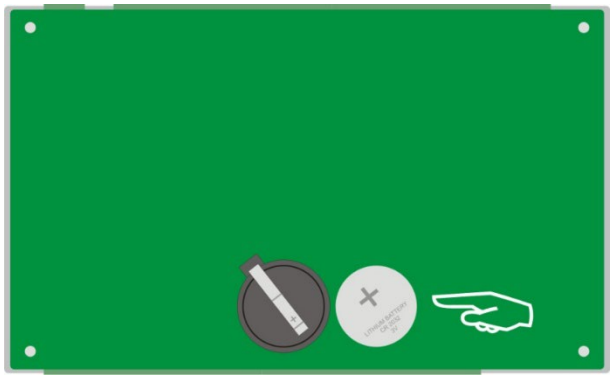


Figure 77

10.1 WARNINGS & RECOMMENDATIONS

For devices that have batteries (rechargeable or no):

Dispose of spent batteries in accordance with the valid legal specifications.

At the end of its useful life, send the batteries to the separate collection for electric and electronic devices (observe local regulations).

11 SPECIFICATIONS

Power:

- **Standard AC Model**: 100 to 240 Vac/dc ± 10 %, 50/60 Hz. Maximum consumption: 20 VA.
- **24 V Model**: 24 Vac/dc ±10%, 50/60 Hz. Maximum consumption: 10 VA.

*24 V model units from serial number **15247775** on can operate from 12 to 30 Vdc/ac ±10%, 50/60 Hz. Maximum consumption: 10 VA.

Environmental Conditions: Operating temperature: 0 to 50 °C. Relative Humidity: 80 % to 30 °C. For temperatures above 30 °C, decrease 3 % per °C.

For internal use.

Installation category II.

Pollution degree II.

Altitude < 2000 m.

Dimensions: 164 x 117 x 70 mm

Weight: 400 g

Housing: ABS+PC

Protection: IP20

Analog Inputs:

The types of input signals accepted by **FieldLogger** and their maximum ranges of measurement are selected in the **NXperience** software and are listed in the following table.

INPUT TYPE	MEASURING RANGE	ACCURACY
Thermocouple J	-120 to 1000 °C (-184 °F to 1832 °F)	± 0.2 % (F.R.) ± 1 °C
Thermocouple K	-130 to 1372 °C (-202 °F to 2501,6 °F)	± 0.2 % (F.R.) ± 1 °C
Thermocouple T	-130 to 400 °C (-202 °F to 752 °F)	± 0.2 % (F.R.) ± 1 °C
Thermocouple E	-130 a 780 °C (-202 °F to 1436 °F)	± 0.2 % (F.R.) ± 1 °C
Thermocouple N	-130 to 1300 °C (-202 °F to 2372 °F)	± 0.2 % (F.R.) ± 1 °C
Thermocouple R	20 to 1768 °C (68 °F to 3214,4 °F)	± 0.2 % (F.R.) ± 3 °C
Thermocouple S	20 to 1768 °C (68 °F to 3214,4 °F)	± 0.2 % (F.R.) ± 3 °C
Thermocouple B	100 to 1820 °C (212 °F to 3308 °F)	± 0.2 % (F.R.) ± 3 °C
Pt100	-200 to 850 °C (-328 °F to 1562 °F)	± 0.15 % (F.R.)
Pt1000	-200 to 850 °C (-328 °F to 1562 °F)	± 0.15 % (F.R.)
Linear 0 to 20 mA	Configurable	± 0.15 % (F.R.) *
Linear 4 to 20 mA	Configurable	± 0.15 % (F.R.) *
Linear 0 to 20 mV	Configurable	± 0.15 % (F.R.) *
Linear 0 to 50 mV	Configurable	± 0.15 % (F.R.) *
Linear 0 to 60 mV	Configurable	± 0.15 % (F.R.) *
Linear -20 to 20 mV	Configurable	± 0.15 % (F.R.) *
Linear 0 to 5 V	Configurable	± 0.15 % (F.R.) *
Linear 0 to 10 V	Configurable	± 0.15 % (F.R.) *

F.R. = Full Range = Span

(*) Note: The full scale refers to the input of the sensor signal and not the range of configured indication.

Note: Accuracy is only guaranteed if the equipment is installed as recommended in the manual, with all terminals connected to the equipment and with the equipment operating in a stable temperature environment for at least 1 hour.

Table 17

Accuracy is guaranteed at intervals of reading greater than 0.2 seconds per channel. For faster read rates (intervals of less than 0.2 seconds), there will be a loss of accuracy that will be as increased as the rate of reading. Likewise, although most of the channels can indicate a little beyond the range limits set, the specification is not guaranteed outside the range.

Input impedance of the analog channels:

- Thermocouples / Pt100 / Pt1000 / mV: > 2 MΩ
- mA: 15 Ω + 1.5 V
- V: 1.1 MΩ

Maximum compensated Pt100/Pt1000 cable resistance: 40 ohms

Excitation current:

- Pt100s: 360 µA
- Pt1000s: 320 µA

Pt100/Pt1000 used curves: Alfa = 0.00385

Digital Inputs:

- Logical levels:
  Logical level "0": from 0 to 0.8 Vdc
  Logical level "1": from 3 to 30 Vdc
- Maximum input voltage: 30 Vdc
- Input Current @ 30 Vdc (typical): 3 mA
- Counting's:
  - Maximum pulse frequency for counting (square wave): 250 Hz
  - Minimum pulse time in logic level "0": 2 ms
  - Minimum pulse time in logic level "1": 2 ms
  - Number of bits for counting: 32

Digital Outputs

- Maximum voltage that can switch at the output: 30 Vdc
- Maximum current that can switch at the output: 200 mA

Relays:

- Maximum Current: 3 A @ 250 Vac; 3 A @ 30 Vdc

Loggings:

- Can log up to 1000 channels per second. Limits: 1 channel at 1000 loggings per second or 100 channels at 10 loggings per second.
- Can log either in the internal flash memory (2162688 bytes → approximately 512 k loggings) or in the SD card (*).
- Allows the option of circular logging. After filling up the memory, the oldest data will be overwritten by more recent data.
- Allows data to be downloaded during logging.

(*) IMPORTANT: The effective logging rate depends heavily on the quality and speed of the SD card used. Always prefer to use cards of well-known brands (*SanDisk* and *Kingston*, for example). If the wanted logging rate is high, prefer to use Class 10 or above cards.

Maximum number of channels that can be logged: 100

Supported file systems: FAT32 e FAT16, both in USB flash drives and SD cards

24 V output: 24 Vdc ± 20 % with a minimum load current of 4 mA. Maximum load current: 160 mA. **This output is not provided in the 24 V model!**

Clock battery: Panasonic lithium battery 3 V (Part No. CR 2032).

Clock accuracy: ±3 ppm (typ); ±5 ppm (max)

Supported Modbus commands:

- Read Coil Status (01h)
- Read Holding Registers (03h)
- Write Single Coil (05h)
- Write Single Register (06h)
- Write Multiple Registers (0Fh)

Number of simultaneous TCP connections: 10

Number of simultaneous UDP connections: 10

FTP (FieldLogger as a server):

Supported mode: passive.

Standard: UNIX.

Number of simultaneous connections: 1.

SMTP (email):

Supported authentication mode: AUTH LOGIN.

Certifications: CE, UL.

11.1 CERTIFICATIONS

CE Mark

This is a Class A device. In a domestic environment, this device may cause radio interference in which case the user may be required to take adequate measures.

12 SAFETY INFORMATION

Any control system design should consider that any part of the system has the potential to fail. This device is not a protection or safety device, and its alarms are not intended to protect against device failures. Independent safety devices should be always provided if personnel or property are at risk.

Device performance and specifications may be affected by its environment and installation. It is user's responsibility to assure proper grounding, shielding, cable routing and electrical noise filtering, in accordance with local regulations, EMC standards and good installation practices.

13 WARRANTY

Warranty conditions are available on our website www.novusautomation.com/warranty.