Datalogger DatumX Operations Manual

xactus

Datalogger DatumX: Operations Manual

xactus
SMART CITIES

Website: www.xactus.io
Email: info@xactus.io
Calle 34a#36a-27, Bucaramanga, Santander, Colombia.
English — Version 2.0.0

January 27th of 2025

Abstract

This document describes the capabilities, features and technical specifications of the **DatumX** data-logger, as well as its correct usage, installation, and configuration. As our flagship device, the DatumX provides reliable measurements and operations control for a wide variety of applications, being an excellent choice specially for remote operations. In addition, both our hardware and software are constantly updated improving its characteristics and overall performance. We have prepared this document to explain in the most comprehensive way its whole functions and features.

The information contained in this manual is subject to change without notice. Effort has been made to make the information in this manual complete, accurate, and current. The manufacturer shall not be held responsible for errors or omissions in this manual. Consult www.xactus.io for the most up-to-date version of this manual.

Please, read the operations manual before using the Datum X datalogger for the first time and keep the manual for later reference once familiar with the equipment. Also, make sure to observe all of safety instructions and warnings given in the manual. Only operate and maintain the Datum X data-logger if you are qualified and trained for this purpose. Please, do not open the Datum X, as there are no user-serviceable parts inside of it. If the device is defective and needs to be repaired, please contact our support center for further assistance (contact information in section 5.3).

The manufacturer is not responsible for any damages due to misapplication or misuse of this product including, without limitation, direct, incidental, and consequential damages, and disclaims such damages to the full extent permitted under applicable law. The user is solely responsible to identify critical application risks and install appropriate mechanisms to protect processes during a possible equipment malfunction. Consequently, Xactus S.A.S. accepts no liability for damage resulting from the device not being used as intended. All warranty claims are nullified in this case. Unauthorized structural modifications, as well as additions or alterations to the device are prohibited. To ensure that the device is used as intended, only connect and use accessories/spare parts that have been approved by Xactus.

Operating instructions, manuals and software are subject of copyright. Copying, duplication, translation, conversion into any electronic medium or any machine readable form, as a whole or in parts, is not permitted, with the exception of making a back-up copy of the software for saving purposes, insofar as this is technically feasible and is recommended by our company. Contraventions will lead to compensation.

Thank you for purchasing a DatumX Datalogger. This manual is a brief overview of the DatumX and its basic features. Aditional communications options, including cellular modems and sensor integration are not included in this guide.

Contents

1 About the Device

The Data-logger DatumX, is a compact device which allows acquisition, storage, processing and transmission of data, previously measured from sensors, to the cloud . It provides many different sensor interfaces, from **analogical** and **digital** inputs, to relevant industrial protocols as **RS485**. Among the DatumX features, there's both local and remote data storage, having **3G/4G modem** compatibility for data transmission to the cloud and **SD card** backup features. In addition, it allows **FTP transmission**, **Socket**, **API REST**, **HTTP-JSON Transmission** and **Modbus TCP** slave configuration.

The DatumX has a **built-in web server** accessible via WiFi that allows sensor and device configuration, as well as data download and many visualization options. Its compact and robust design allows it to be the main core of monitoring stations, giving quick data insights for smart decision making.

1.1 General Information

→ Table 1.1

Each device comes with basic information that identifies it, table 1.1 shows each information parameter.

→ Section 3.2

To fully inspect this information, users can go to the built-in **web-server**. Please, review section 3.2. Instructions about how to change some of the parameters are in section 3.5.

Table 1.1: General Information Parameters

Parameter	Description
Serial Number	Unique ID number that identifies it. Generally, has the following structure: **CE0000**.
Alternative Serial	This parameter can be defined by the user. It is intended for client's own traceability.
Device Name	User defined. Allows to put a familiar name to the device.
Board	Consist in the motherboard model. Most of DatumX comes with the model: **Black**.
Firmware Date	This is closely related to the *Image Date* field, being the date that the firmware version was released.
Firmware Version	Version of the currently installed firmware. This manual was made over version: **1.3.1**
Configuration Date	Date of the last time the device was configured.
Image Date	Date when the installed software version was released.
API Version	Version of the currently installed API REST. This manual was made over version: **1.2.1**.
Lot	This field refers to the device's production year and period. The number digits refer to the production year and the letter corresponds to the production period.
Devices	This field shows configured sensors and other devices. This field is modified during device configuration at factory.
Date	System current date.

1.2 Specifications

Among the DatumX features, it is relevant to point out the following:

1. The DatumX has a built-in web server, which allows to see measurements in real time, configure the device and download data.
   For detailed web server features and instructions, please see Chapter 3.

   → Chapter 3

2. It is possible to connect the device to industrial modems (3G, 4G, or Satellite). This feature, permits data to be sent to the cloud or to a specific server. External modems are pre-configured in our factory; so, modems configuration and installation instructions are not shown in this manual. If you have any modem change or re-configuration needs, please contact us.

3. It can generate alerts triggered by either a value going outside predefined limits or an intrusion sensor sending a signal to the device. Alerts could be observed on the **website** and the **Intelity** platform. Additionally, alerts also include a relay alarm and the possibility to connect a proximity sensor for triggering purposes.
   For further information about alarms and alerts, please refer to 2.9.

   → Section 2.9

4. The device has several different **over-voltage** and **over-current** protections, which make it robust enough for a wide kind of industrial applications.

5. There are many possibilities of integration with already existing third-party systems. There are direct extraction methods, as a **Socket** or **Modbus Slave** configuration. In addition, an **API REST**, **FTP Transmission**, and **HTTP-JSON** methods, allow all sorts of data transmissions.

A full specification sheet can be found on Table 1.2 for detailed information.

→ Table 1.2

Table 1.2: DatumX Specification Sheet (Marked items indicates **additional hardware requirements**)

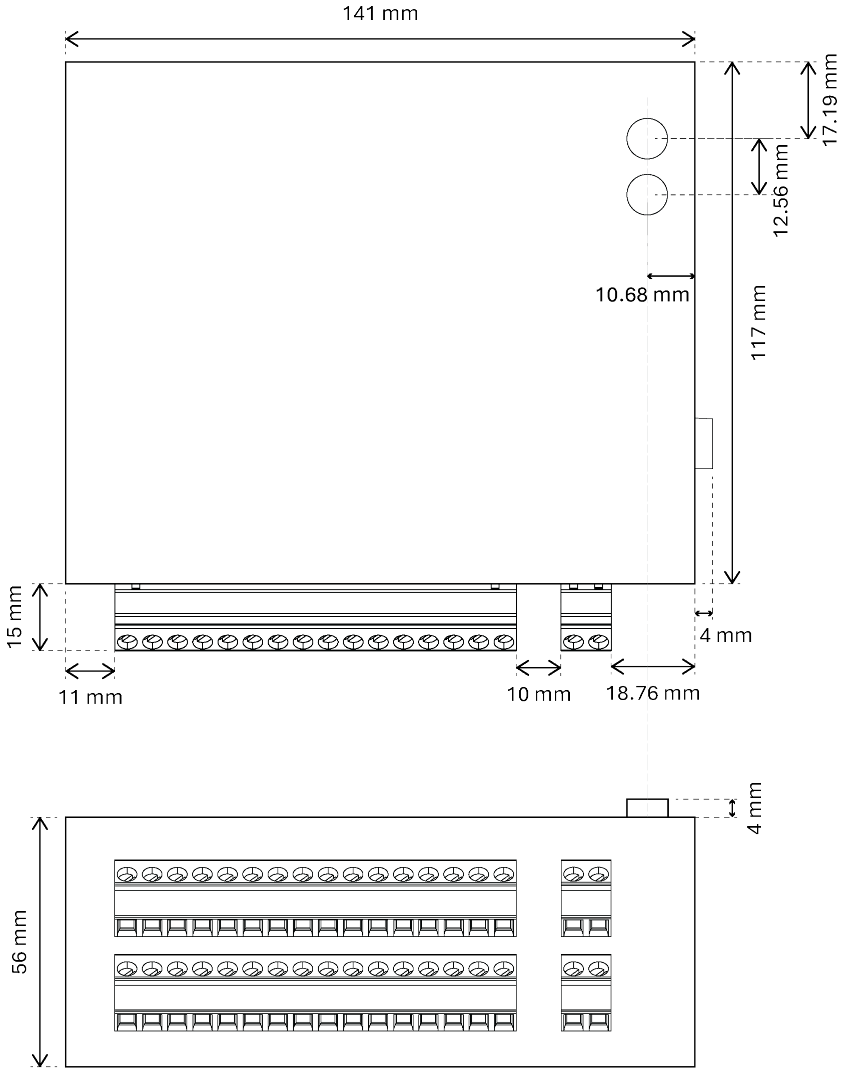
CPU	
Frequency	1.5 Ghz
Cores	4
Instructions Set	Quad-Core ARM Cortex A72 (64 bits)
RAM Memory	8 Gb
Operating System	Linux
Storage (O.S. + Data)	16 Gb; about 10 million values
System Characteristics	
Clock	DS3232; up to 5 years life-span
Indicators	LED and Buzzer
Operating Temperature	-35 C° to 50°C
Variables	Up to 600, with up to 27 sensors
Configuration	Web server
Data Extraction	Modbus TCP Slave, API REST, Socket, Local Download
Data Transmission	FTP Server, HTTP-JSON, LoRaWAN[1], TCP
Power Requirements	
Power Supply	11.5 Vdc to 16 Vdc
Power Consumption	5 Watts (full Operation)
Sensor Interfaces	
RS232	2 ports; up to 2 sensors
RS485	1 port; up to 10 sensors
SDI12[1]	1 port; up to 10 sensors
USB	4 ports; up to 10 sensors.
Communication Protocols	Modbus RTU, Mode4, Bayern-Hessen, NMEA
Baud rate	up to 115200 bauds
Meteorological Station	*Davis Vantage* port
Analogic Inputs	
Ports	2
ADC	Delta-Sigma
Resolution	16 bits
Voltage Range	0 - 2.5 Volts
Resistor	100 Ω±0.01%
Digital Pulse Inputs	
Ports	2
Events	7, refer to 2.7.4
Alarms	
Alerts	Relay, jack 3.5 mm audio output
Relay	Solid state
Voltage Peak	±40 Volts peak; 2 Amp RMS
WiFi	
Adapter	Built-in IEEE802,11b/g hotspot
Frequency	2.4 Ghz
Ethernet	
Protocols	IPv4, IPv6, TCP/UDP, HTTP(s), SMTP/TLS, POP3/TLS, FTP/SFTP, SSH, Telnet
Speed	10/100 Mbps
Ports	RJ-45
Miscellaneous	
Weight	Aprox. 780 gr

1.3 Hardware Overview

→ Figure 1.1

In general aspects, the DatumX datalogger is enclosed by a metallic case, which protects it from dust and other environmental facts that can harm the device. However, this is not considered a complete outdoor protection making necessary the use of an additional cabinet. DatumX is designed as a compact and versatile device usable in many different applications.
Figure 1.1 gives an overview of the device's dimensions.

Figure 1.1: DatumX Dimensions



→ Figure 1.2

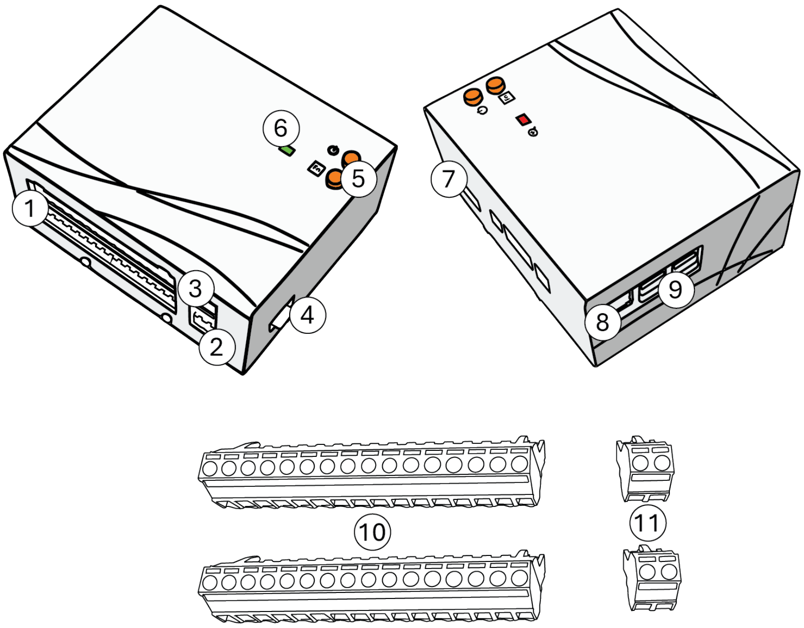
All general hardware features can be seen on Figure 1.2.

Wiring Panel: The DatumX wiring panel provides ports and **removable** terminals for connecting sensors, power, and communication devices. This feature allows user to make connections easily and efficiently; it is also necessary to **tighten** the corresponding screws for each slot used.

→ Table 1.3 & Figure 1.3

According to the **communication interface** or **sourcing option** used, DatumX wiring panel has a different classification for each slot. Table 1.3 and figure 1.3 describe all slot types functions and location within the wiring panel.

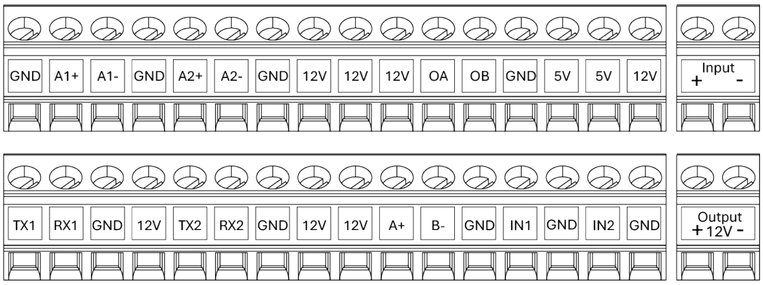
Figure 1.2: DatumX Hardware Overview



All parts shown in figure 1.2 are described as follows:

1. Main wiring panel
2. 12 Volts power output
3. Main power input
4. Main fuse port
5. Start button & auxiliary button
6. Status LED
7. Weather station port
8. Ethernet port
9. USB ports
10. Wiring panel’s removable terminal
11. Power input & 12 Volts power output removable terminals

Figure 1.3: DatumX Wiring Panel Slots



GND	A1+	A1-	GND	A2+	A2-	GND	12V	12V	12V	OA	OB	GND	5V	5V	12V	Input + -
TX1	RX1	GND	12V	TX2	RX2	GND	12V	12V	A+	B-	GND	IN1	GND	IN2	GND	Output + 12V -

Status LED and Buttons: Among the mentioned external features, there are also the status LED and action buttons, which are useful for basic troubleshooting and some other functions.

→ Sections 1.4, and 1.5

All the functions related to these components are discussed in sections 1.4, and 1.5.

Internal Hardware:

DatumX enclosure is easily removable, facilitating preventive maintenance duties. Thus, it is important to get familiar with its main internal components and structure. These internal characteristics are shown in further detail on figure 1.4.

WARNING! →

Of course, it is important to mention that enclosure removal is **only** allowed for this purpose, and that this procedure might be performed by a trained professional.

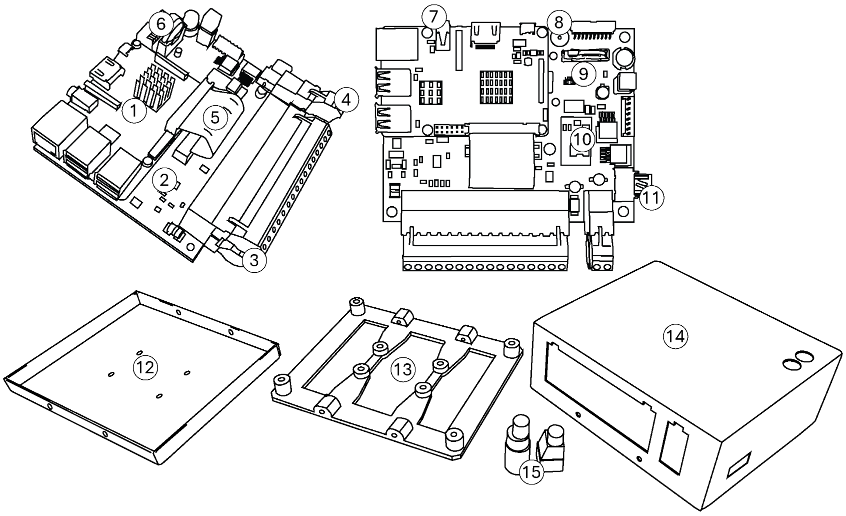
→ Section 5.1

Please, review section 5.1 for further information about this process.

Hardware Protection:

The device is protected against surge, over-voltage, and over-current in its power supply terminal. It is also dust protected through a metallic hardcase.

Figure 1.4: DatumX Internal Hardware Overview



All parts shown in figure 1.4 are described as follows:

1. Main CPU
2. Motherboard
3. Wiring panel (removable ports on)
4. Power supply input
5. Main data connector (CPU-Motherboard)
6. RTC battery
7. Jack 3.5 mm audio input
8. Buzzer
9. Status LED
10. 12 Volts & 5 Volts regulator
11. Main Fuse
12. Case lower part
13. Board-Case attachment piece
14. Case top part
15. Start & Auxiliary buttons extensions

1.4 Buttons

→ Figure 1.2

As shown in figure 1.2, there are two buttons called **Start Button** and **Auxiliary Button**.

Start Button: This Button allows to manually turn off the DatumX without accessing the web serve nor unplugging the device. It also allows the device to turn on again without unplugging and plugging the power supply.

Auxiliary Button:
→ Figure 1.3 & Table 1.3

The Auxiliary button function is to deactivate the sensor power supply (12 Volts output, see figure 1.3 & table 1.3). Thus, allowing to modify the wiring of any connected sensors without turning off the DatumX and affecting the data-logging function. When the power supply output is stopped, the buzzer will emit a signal every 5 seconds in order to aviod levaing this output turned off. By default, the DatumX turns on this power supply output on startup.

WARNING! → It is relevant to mention that this function **only** deactivates the sensor power supply output, **NOT** the modem power supply output, 5 Volts power supply, Ethernet/USB, nor weather station input. So, making any changes to sensors connected to these power supply outputs may cause damage to the sensors, DatumX, and personal involved.

1.5 Status Beeps and Blinks

→ Figure 1.4

The Datalogger DatumX possesses several status blinks and beeps, which indicate its status during the startup and data-logging state. This feature is helpful for constantly monitoring the current state of the device; and in case there is a problem, it is useful for an initial troubleshooting approach. As shown in figure 1.4, There is only one **LED** and **Buzzer** in charge of status indication. There are several situations recognizable via these components as follows:

Startup Sequence:

1. Once the DatumX is turned on, there will be **two beeps** (in a span not longer than 20 seconds). This will mean that the device has begun to initialize all processes. The status LED must be **yellow** before the first beep, then, it will turn to **green** after the beep, finally, it will **turn off** after the second beep. After the next three minutes, the device will **beep once** and generate a **WiFi network**, which indicates the start of the measurement function.

Error Codes:

2. There are different error situations that could be recognized by the amount of times that both the buzzer and the status LED blink and beep, respectively. The error code will repeat every few seconds.
   For reference of the error code, go tho the Table 1.4.

→ Table 1.4

Status LED: There are some situations that are **only** expressed by the status LED:

1. Whenever a sensor is configured in the device, the DatumX will look for its measurements since the startup sequence. If a configured sensor is not found, the status LED will turn **red**.
2. After the device startup, it should start measuring the sensors connected (known as the *logger* process). If there is an error in the measurement process, the status LED will turn **red** before the startup sequence ends.

Buzzer: After all the mentioned situations, there is one pointed out by the buzzer **alone**. As the section 1.4 mentioned, the buzzer will start to beep **once** every **five** seconds whenever the auxiliary button triggers the function to stop the 12 volts sensor supply output.

→ Section 1.4

Table 1.3: Wiring Panel Classification

Slot Name	Description
GND	Electric Reference (0 Volts). Every GND port is equivalent, as they are parallel connected
A1+	Positive terminal for first Analog channel
A1-	Negative terminal for first Analog channel
A2+	Positive terminal for Second Analog channel
A2-	Negative terminal for Second Analog channel
12V	12 Volts output for sensor sourcing. Every 12V port is equivalent, as they are parallel connected
OA	Relay output ”A” Terminal
OB	Relay output ”B” Terminal
5V	5 Volts output for sensor sourcing. Every 5V port is equivalent, as they are parallel connected
TX1	First RS-232 channel transmission terminal, this is from DatumX perspective
RX1	First RS-232 channel reception terminal, this is from DatumX perspective
TX2	Second RS-232 channel transmission terminal, this is from DatumX perspective
RX2	Second RS-232 channel reception terminal, this is from DatumX perspective
A+	RS-485 interface ”A+” Terminal
B-	RS-485 interface ”B-” Terminal
IN1	Positive terminal for the first digital pulse channel
IN2	Positive terminal for the second digital pulse channel

Table 1.4: Error Codes

State	Alert	Description
Error 1	2 blinks and beeps	12 Volts Sensor Output Error
Error 2	3 blinks and beeps	5 Volts Sensor Output Error
Error 3	4 blinks and beeps	Power Supply Error (out of range)
Error 4	5 blinks and beeps	Davis Power Supply Output Error

2 DatumX Functions

This chapter shows in detail how the DatumX measures, process, stores, and send data. Also, date and time is explained here. Finally, all sensor or communication interfaces available in the DatumX are described in this chapter.

WARNING! → All specifications shown in this chapter are **only** applicable to DatumX devices with the firmware version **1.3.1** or higher.

→ Sections 5.3 & 5.3.1

If your device needs a firmware update, please contact us through our support (see section 5.3) or general contact channels (see section 5.3.1).

2.1 Sampling

There are three essential parameters involved in a sensor measurement: **Sample Time**, **Sub-Samples Quantity**, and **Sampling Function**. Every measurement (or sample) registered in the DatumX is the **result** of the sampling function application to a set of sub-samples.

Sampling Parameters: All three mentioned parameters are chosen by the user when configuring a sensor. The DatumX defines how often to ask the sensor for a sub-sample, this is done by dividing the **sample time** with the **sub-samples quantity** parameters. Each time that the DatumX takes a sub-sample from the sensor, it fills a **sub-samples vector**. Once a full sample time cycle has passed, the now complete sub-samples vector has to go through a **Sampling Function**, turning into a single value corresponding to the final **Sample**.

→ Figure 2.1

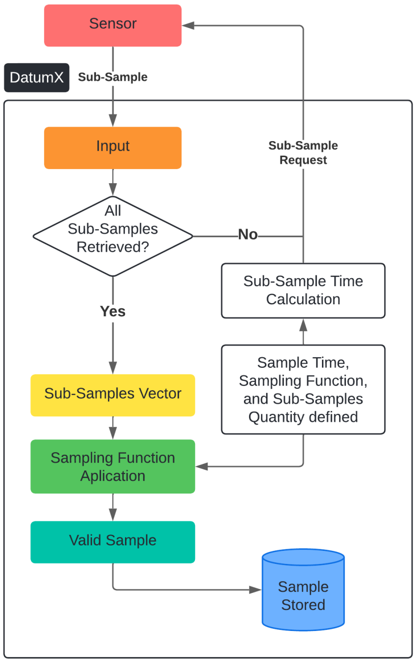
This process is better shown in the Figure 2.1.

Sampling Limitations: It is also important to mention that there are certain **limitations** about the sub-sample time, being limited to a minimum of **one second**. So, this condition limits the maximum amount of sub-samples according to every situation.

Sampling Functions: There are certain kind of variables that might need different sampling functions. Thus, DatumX has a wide set of sampling functions as follows:

1. Arithmetic Mean
2. Arithmetic Geometry
3. Arithmetic Quadratic
4. Statistical Mode
5. Median
6. Maximum
7. Minimum
8. Sum
9. Vector Addition
10. Last Sub-Sample
11. First Sub-Sample
12. Logarithmic Mean

Figure 2.1: Sampling Process


Sensor
DatumX
Sub-Sample
Input
Sub-Sample Request
All Sub-Samples Retrieved?
No
Yes
Sub-Sample Time Calculation
Sample Time, Sampling Function, and Sub-Samples Quantity defined
Sub-Samples Vector
Sampling Function Aplication
Valid Sample
Sample Stored

2.1.1 Sampling Synchronicity

Sampling Synchronicity: **WARNING!** → The DatumX is able to sample in two different ways, **synchronous** or **asynchronous**. However, this parameter affects **all** measurements being taken. In other words, all variables must be either one or the other.

Synchronous Sampling: This means that the samples taken will **synchronize** with the time of the day. For instance, if a measurement began at 16:34:23, and the sample time is 10 minutes, the first sample will be calculated and stored at 16:40:00. There are different sample time possible values for synchronous sampling.

→ Table 2.1 Please, refer to Table 2.1 for all possible sample time values.

Asynchronous Sampling: In case of asynchronous sampling, the DatumX will store measurements **without synchronizing** them with the time of the day. In other words, if a measurement startet at 16:34:23 (with sample time equal to 10 minutes) it will be stored at 16:44:23. It is important to note that in this mode the **lowest** sample time is **1** second, and **87000** seconds, the **highest**.

2.1.2 Sample Quality

Sample Quality Parameters: In order to guarantee different sample quality standards. There are two parameters that ensure sample quality by different approaches. First, the user can define a parameter called **valid subsamples percentage**, which defines what proportion (percentage) of **sub-samples** must be **valid** to consider the whole sample as valid. A valid sub-sample is just a valid response from the sensor (non-null). Lastly, to avoid erratic and noisy sets of sub-samples, the user can add a restriction for valid sub-samples being a **standard deviation threshold**. This parameter, called **std dev treshold**, will discard any sub-sample whose standard deviation is **past** from the threshold defined. By default, both of these parameters are set to not make any quality verification.

Table 2.1: Sample Time Values

Sample Time
30 seconds
1 minute
2 minutes
3 minutes
4 minutes
5 minutes
6 minutes
10 minutes
12 minutes
15 minutes
20 minutes
30 minutes
1 hour
2 hours
3 hours
4 hours
6 hours
8 hours
12 hours
24 hours

→ Section 3.3.1

For reference about how to configure these parameters, please refer to section 3.3.1.

2.2 Logging

The datalogger DatumX creates files with *Logs*, which allows the user to monitor its activity with useful messages during its functioning. As there are two main processes in the device, **Logger** and **Transmission**, two log files will be created.
Log files have a defined structure, showing one piece of information per line. Each line contains: Date, category, and description.

2.2.1 Logger Process

This might be called as the **system logs** as well, and show all kind of information related with the device operation and measurements made. Generally speaking, it will commonly show information about the created **variables**, if an **alert** is triggered, updates about the **filesystem**, and so on.
Ahead, there is an example of a typical system log content.

```
2024-08-23 08:12:57,078 - MAIN: DATUMX Datalogger V1.2.1
2024-08-23 08:12:57,082 - MAIN: Serial Number = CE0000
2024-08-23 08:12:57,350 - MAIN: I2C lock was deleted.
2024-08-23 08:12:57,004 - RTC: System clock updated and verified.
2024-08-23 08:12:57,469 - MAIN: Mainboard detected. Model = 2 (DatumX)
2024-08-23 08:12:57,480 - FILESYSTEM: core files checked.
2024-08-23 08:12:57,499 - FILESYSTEM: added extension: FPI_TRANSMITTER_lib.pyc
2024-08-23 08:12:57,502 - FILESYSTEM: added extension: PLANTOWER_lib.pyc
...
2024-08-23 08:12:58,474 - FILESYSTEM: added protocol: modbus_tcp_client.pyc
2024-08-23 08:12:58,602 - DATABASE: main thread started.
2024-08-23 08:12:58,606 - DATABASE: monitor thread started. Sampletime = 900
2024-08-23 08:12:58,607 - DATABASE: cleaner mode: instantaneous. Period = 30.0 days
2024-08-23 08:12:59,121 - INTRUSION ALARM: disabled.
```

```
2024-08-23 08:12:59,122 - VAR_SYNC: time-synchronized mode active.
2024-08-23 08:12:59,136 - DATABASE: logging sampletime = 1
2024-08-23 08:12:59,305 - MONITOR: measure thread started.
2024-08-23 08:12:59,369 - PONSEL: 30: read process started. Sampletime = 32.0 [s]
2024-08-23 08:12:59,768 - POWER: 12V line power ON.
```

2.2.2 Transmission Process

As well as the Logger Process, the log file corresponding to the transmission process shows updates related to all **communications** and **data transmissions** made by the device. For instance, a typical transmission log file would look the following way:

```
2024-08-26 11:52:46,261 - XACTUS_PROTO: DATA: data sent, and flags updated.
2024-08-26 11:52:46,245 - XACTUS_PROTO: DATA: processing IDs...
2024-08-26 11:52:46,169 - XACTUS_PROTO: DATA: sending success!
2024-08-26 11:52:46,014 - XACTUS_PROTO: DATA: sending...
2024-08-26 11:52:46,013 - XACTUS_PROTO: CFG_UPDATE: no changes were made.
2024-08-26 11:52:46,012 - XACTUS_PROTO: HEADER: sent.
```

2.2.3 Errors

Even though the already described files will cover the majority of errors during operations, there might be situations of programming conflict (as a corrupt **configuration file**) that will be shown in another couple of log files, called **Logger Error** and **Transmission Error** respectively.

In this case, the previously described log file structure does not apply, and the file will only have the error message.

2.2.4 VPN Process

There is a third process constantly running in the background in charge of managing remote connections to the device. This remote connection is made through a ***Virtual Private Network*** (VPN).

WARNING! → This section is intended for information purposes and VPN processes **should not** be disabled in any way.

2.3 Monitor

By default, the DatumX monitors its main **internal variables**. These values are available with the instant variables for user monitoring, but also, the device would trigger an alert if any of the values are outside recommended boundaries.

→ Section 2.9

For further information about the monitor alerts, please refer to 2.9.

The monitor process will check main system variables once every 10 minutes. This value can be changed according to the user needs in the device configuration, however, it is recommended **not** to change it.

→ Table 2.2

The complete list of system variables monitored by the system can be found on Table 2.2.

It is important to note that depending on the modem used with the DatumX, the Signal Strength variable might not show any value. However, this does not reflect a bad signal reception or any related signal problems.

Network Monitor

The network monitor works alongside the overall monitor and allows the device to constantly verify the **network status** (as shown in table 2.2). However, as there is no internal modem module, some parameters have to be defined in order to

Table 2.2: System Variables Monitored

Variable	Sample Time (min.)	Sub-Samples
Main Power Input	10	10
3.3 Volts Power Output	10	10
5 Volts Power Output	10	10
12 Volts Power Output	10	10
RTC Temperature	10	10
Processor Temperature	10	10
Processor Usage	10	10
Signal Strength	10	10
Network Status	10	10
Detected Ping	10	10
Data Usage	10	10
RAM Usage	10	10
Disk Data Base Size	10	10

properly monitor the network status.

→ Table 2.3

Table 2.3 explain these parameters. In any case, modems are factory configured and further configuration is not necessary in the vast majority of cases. Thus, explanations made are just for information purposes.

Table 2.3: General Transmission Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Network Monitor	Enables or disables network monitoring	**Disabled**; Enabled
Ping Keepalive	Enables or disables ping keepalive	Disabled; **Enabled**
Ping Keepalive IP address	Defines the IP address for ping keepalive	**8.8.8.8**
Ping Keepalive retry	Defines the amount of retries for ping keepalive	0 - 20; **5**
Network Monitor Sample Time	sampling time for network monitor in seconds	10 - 87.000; **600**

→ Sections 2.8.1, 3.5.4 and 4.3.4

For further information about internet connection, please visit sections 2.8.1, 3.5.4 and 4.3.4.

2.4 Instant Variables

The **last** measured values of each variable are known as **instant values**, and, depending on configuration, they could be either the **last sample** or the **last sub-sample**. These values could be retrieved by **Modbus TCP**, **Socket**, **API**, or consulted on the **web server**.

→ Section 2.8

For further information about data retrieval through all described options, please go to section 2.8.

2.5 Data Storage

All samples taken are **locally** stored, at least initially, as the DatumX has approximately **10 million sample** capacity. However, by default, the DatumX is configured to **only** stores data that **has not** been sent to the cloud server. The device will execute a **cleaning routine** (every 30 days by default) that will check stored data for already sent data deletion. This parameter can be changed by user request.

If the device has **no internet access** or data deletion is not active, all samples taken will be stored **locally**. The maximum time that the DatumX can store data could be estimated by dividing the maximum samples that can be stored (10 million) by the amount of data gotten in a specific time span. For this, the number of variables and sample time must be taken into account.

→ Equation 2.1

Now, this calculation is shown in equation 2.1, where T_{max} represents the maximum storage time, $Data_{max}$ the maximum samples that can be stored, and $Data_t$ the data stored in a specific time span.

$$T_{max} = Data_{max}/Data_t \tag{2.1}$$

WARNING! → If the device will be used without any internet connection, it is recommended to estimate the maximum storage time and download data periodically.

WARNING! → If data is programmed to be sent to the cloud through both of the available methods, the data will only be erased after being successfully sent on both of them.

→ Section 2.8

For additional information about the available data availability methods, please refer to section 2.8.

2.6 Date & Time

The DatumX has an internal **RTC circuit** (*Real Time Clock*) dedicated to date and time conservation. This circuit, allows the device to **keep** the actual time even when is **turned off**. In addition, the DatumX has **automatic time request** features through a NTP server (*Network Time Protocol*), so it is able to keep date and time up to date unattended.

WARNING! → If time in the **RTC** gets wrong, the DatumX will instantly detect the error and **will stop** any measurement. The device itself is able to detect any error on the RTC date and time. Whenever this error happens, it is necessary to update the date and time through the web server.

2.6.1 Time Zone

The time zone Linux package (*tzdata*) is used for all time zone managing in the device. In this package, time is specified by different time zones. By default, DatumX factory timezone corresponds to the region the data-logger is acquired. Unfortunately, this package does not perform automatic updates, so any change in time regulations will not be reflected on it.

→ Section 3.5.1

Time is easily configured through the web server, please refer to section 3.5.1 for instructions.

2.7 Communication Interfaces

There are some different communication interfaces that allow all kind of sensors to be measured. The available interfaces are the following:

1. RS485.
2. RS232.
3. Digital Pulse Input.
4. USB.

5. Ethernet.
6. Analog Input.
7. Meteorological Station Input (*Davis Vantage*).
8. SDI-12[1].

→ Figure 1.3

As seen in Chapter 1, Section 1.3, all the DatumX input ports are shown in Figure 1.3.
It is possible to connect up to **27 sensors simultaneously** using all available communication interfaces. The connection present between a sensor and the DatumX is called *Communication Channel*, thus, each sensor correponds to a *Channel*. A physic input port could have several *Channels* according to the maximum amount of sensors it can communicate with. For instance, RS485 communication interface has 10 available *Channels* associated to just one input port, hence, allowing to connect up to 10 sensors to it.

→ Table 2.4

For detailed information about DatumX *Communication Channels*, please refer to Table 2.4.

Table 2.4: Sample Time Values

Communication Interface	Channels Quantity	Channel Number
Analog	2	1,2
Meteorological Station	1	17
RS232	2	18,19
RS485	10	22-31
Digital	2	34, 35
USB/Ethernet/SDI-12	10	37-46

There are a few **accepted communication protocols** for RS232, RS485, and Ethernet/USB communication interfaces: Modbus RTU, Modbus TCP, Mode4, Bayern-Hessen, NMEA, TOPAS. In addition, all sensors using any of these communication interfaces must have been integrated on the DatumX firmware.

→ Appendix A

For a detailed list of integrated sensors, please refer to appendix A.

WARNING! →

It is important to make sure that each sensor's power requirements meets (and not exceeds) the power supplied by the DatumX. If not met, both devices could be **harmed**.

→ Figure 2.2

The physical location of all mentioned communication interfaces is shown in figure 2.2.

2.7.1 Analog

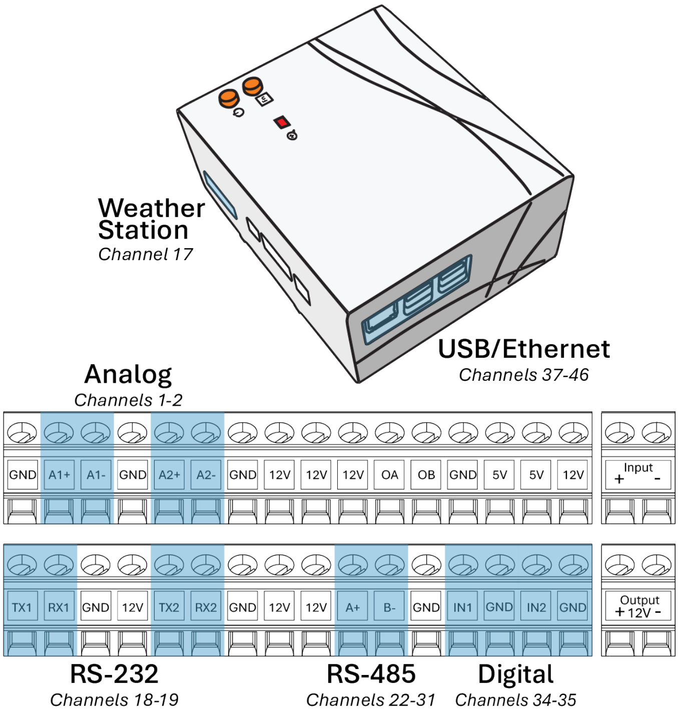
There are 2 independent channels (1 and 2) compatible with current based analog sensors (4-20 mA). Both channels have an internal 100 $\Omega \pm 0.01\%$ resistor. The ADC (*Analog to Digital Converter*) has a 16 bit resolution.

WARNING! →

The analog input ports measure the current through the positive and negative terminals (either A1+ to A1- or A2+ to A2-), however, this current must **NOT** be higher than 25 mA nor negative.
As the reading made is in mili Amperes, it has to be converted to specific dimensions. This, is made by setting the parameters **Span** and **Offset** (usually provided by sensor manufacturer). If the channel contains several variables, these parameters can be set for each variable individually. Also, **adcslope** and **adcoffset** parameters affect the whole channel if needed.

[1] With an additional USB adapter. Contact us for further information.

Figure 2.2: DatumX Channels



2.7.2 Meteorological Station

This communication interface is reserved **only** for either a ***Davis Vantage Pro2*** or ***Davis Vantage Vue*** meteorological station. The assigned channel to this input is the channel 17.

2.7.3 RS232/RS485

There are two physically separated channels dedicated to RS232 sensors (channels 18 and 19). In the other hand, the RS485 communication interface has 10 channels (22-31) included in just one physical input. Baudrate values could be between 1200 and 115200 bauds.

WARNING! → It is important to mention that each sensor connected must have a different **address**. Therefore, all addresses must be checked and configured on the sensors **prior** to configure them on the DatumX.

2.7.4 Digital Pulse

Both channels 34 and 35 were designed as digital pulse inputs. There are many different applications in which these inputs may be used; where **rain** and **intrusion** sensors are some of the most common. Hence, there is a wide variety of configurable ***Events*** in each input that might adjust to each purpose.

→ Table 2.5

As seen in the Table 2.5, there are 7 different *Events* explained.
The logic levels used in the digital inputs are 0 volts for logic 0, and 5 volts for logic 1. Each input has a *pull-up* resistor (2KΩ), which allows to read both logical values when there is a high impedance. This, permits to connect sensors with 0 volts input through a relay.

Every sensor connected to this interface only needs to meet the specified **signal ranges** for logic 1 and 0, without the necessity of integration.
Lastly, as well as the other mentioned communication interfaces, parameters as **Span** and **Offset** are also available for sensors connected to any of the digital pulse inputs.

Table 2.5: Digital Input Configuration Options

Event	Input Signal Mode	Description	Common Use
0	General Purpose Input	Determines if the input value is either 1 or 0 at sub-sampling moment.	General Purpose
1	Rising Edge Event	Triggers an interruption when a rising edge is detected	Intrusion Alerts.
2	Falling Edge Event	Triggers an interruption when a falling edge is detected	Intrusion Alerts.
3	Both Edges Event	Triggers an interruption when either a rising edge or a falling edge is detected.	Intrusion Alerts
4	Rising Edge Counter	Counts all the rising edges detected until the next sub-sample.	Rain Sensors
5	Filling Edge Counter	Counts all the falling edges detected until the next sub-sample.	Rain Sensors
6	Both Edges Counter	Counts all edges detected until the next sub-sample.	Rain Sensors

2.7.5 USB/Ethernet

It is possible to connect up to 10 sensors through all the 4 USB ports and the Ethernet port, of course, knowing that the sensors must be integrated into the DatumX firmware. Hence, the assigned channels for both Ethernet and USB are from 37 to 46.
It is also possible to connect a SDI-12 adapter allowing to use the defined channels with this interface. The SDI-12 module is an additional module that has to be requested and is subject to availability.

WARNING! → It is **ONLY** allowed to connect USB expansion modules with an external power source.

2.8 Communications and Transmissions

This section describes all DatumX functions related to data retrieval or data transmissions. It is important to mention that this section does not cover any information related to cellular or satellite modems configuration, as this type of transmissions are **ONLY** factory configured.

→ Section 5.3

If your device presents a problem or needs a re-configuration regarding communications, please contact us (information in section 5.3).
Even though all the transmission options allow to define exclusive parameters for

each of them, there are a couple of parameters that would override them and affect all transmission means globally. The Table 2.6, shows each one of this parameters and its description.

Table 2.6: General Transmission Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Automathic Data Transmission	Enables data transmission as soon as a new value is written in the database	**Disabled**; Enabled
Max Rows per Package	Defines the maximum amount of rows per package in a transmission	1 - 10'000.000; **3.000**

Taking a closer look to the Table 2.6, the **Automathic Data Transmission** parameter can indirectly **change** the transmission time in most transmissions configured. If **enabled**, the device will automatically transmit all new data written into the data base, making the transmission time equal to the **minimum** sampling time configured.

In the other hand, the parameter **Max Rows per Package** modifies the amount of packages transmitted, limiting each package's size.

→ Sections 2.8.7, 2.8.8, 2.8.9, 2.8.10

Both of the parameters described affects the following transmission options:

1. FTP Transmission, see Section 2.8.7.
2. HTTP-JSON Post, see Section 2.8.5.
3. Xactus Transmission, see Section 2.8.9.
4. LoRaWAN Transmission, see Section 2.8.10.

2.8.1 Internet Connection

Some of the communications described in this section depend on the device's internet connection and its quality (i. e.,**FTP Transmission**, **HTTP-JSON Post**, **API REST**, and **Xactus TCP**). Hence, the DatumX can obtain internet connection through a modem. Usually, DatumX are delivered wit cellular, satellite, or radio modems among different options.

→ Table 2.7

Even though modems are factory configured to work alongside DatumX and no further configuration is necessary, there are a couple of configurable parameters according to network requirements in table 2.7.

Table 2.7: Network Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Enable DHCP Client	Enables or disables DHCP client	**Disabled**; Enabled
Gateway	Defines gateway's IP address	**192.168.10.1**
IP Address	Defines device's IP address	**192.168.10.3**
Modem Username	Defines modem's username	Non-applicable
Modem Password	Defines modem's password	Non-applicable

2.8.2 CSV File Generator

The DatumX provides a tool for CSV files generation which allows highly custom files according to each user needs.

→ Sections 2.8.7 & 2.8.3

All CSV files generated are used in two different data availability options, being **FTP transmission** and **Data Download** (Sections 2.8.7 & 2.8.3).

→ Table 2.8

There are four different parameters that can customize CSV files generated, each parameter can be found in Table 2.8.

As shown in the table, the user can choose both the desired columns and order of them in the CSV file.

Header: In addition, it is possible to add a header in the first row of the file containing the following values: **Serial Number, Date of Generation, Alternative Serial Number**.The **Serial Number** value referes to the factory serial number of the device. In the other hand, the **Date of Generation** is an automatically generated timestamp of the file (either the moment when the file was downloaded or sent). Finally, the **Alternative Serial Number** refers to a user identification parameter.

→ Table 2.9

In complement, Table 2.9 explains each column in the parameter **Custom Columns**.

Date Format: The parameter of **Date Format** provides flexibility by receiving any valid argument documented by the **1989 C Standard** (sometimes known as **ANSI C** or **strftime**) and the **ISO 8601**. Please, refer to ANSI C documentation for reference about how to correctly format date.

Table 2.8: CSV Generator Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
CSV Separator	Chooses between colon and semicolon csv separator	**Colon (,)**; Semicolon (;)
CSV Header	Enables or disables an additional header into the csv file. See 2.8.2	**Disabled**; Enabled
Date Format	Defines the date format into the csv file	**YYYY-MM-DD hh:mm:ss**
Custom Columns	Defines which columns to include into the csv file, order can be picked as well	**Timestamp, Variable ID, Variable, Units, Value**, See Table 2.9

Table 2.9: Custom Columns

Column	Description
Timestamp	Date and time of the measurement, format is defined by **Date Format**
Variable ID	Unique identification number of the measured variable
Units	Measurement units set for the measured variable
Variable	Name of the measured variable
Value	Value of the measured variable

→ Figure 2.3

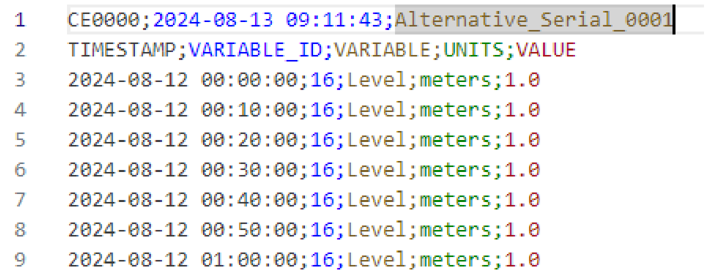
In Figure 2.3, there's an example of a generated CSV file through the CSV generation tool. The parameters values are the following:

1. **CSV Separator:** Semicolon (;).
2. **CSV Header:** Enabled.
3. **Date Format:** Default value (YYYY-MM-DD hh:mm:ss).

4. **Custom Columns:** Default value (Timestamp, Variable ID, Variable, Units, Value).

Looking at the header in the example, there's an **Alternative Serial**, which can have both numeric and alphabetic characters. Also, for illustration purposes, the example contains a configured **level sensor** with **VARIABLE ID: 16**, **VARIABLE: Level**, and **UNITS: meters**.

Figure 2.3: CSV file generated



```
CE0000;2024-08-13 09:11:43;Alternative_Serial_0001
TIMESTAMP;VARIABLE_ID;VARIABLE;UNITS;VALUE
2024-08-12 00:00:00;16;Level;meters;1.0
2024-08-12 00:10:00;16;Level;meters;1.0
2024-08-12 00:20:00;16;Level;meters;1.0
2024-08-12 00:30:00;16;Level;meters;1.0
2024-08-12 00:40:00;16;Level;meters;1.0
2024-08-12 00:50:00;16;Level;meters;1.0
2024-08-12 01:00:00;16;Level;meters;1.0
```

2.8.3 Data Download

Files generated by the CSV Generator can be downloaded through a built-in function for this purpose. For accessing this function, the user can go to the **web server** and insert the required **start** and **end** dates for the data to be included in the file.

→ Section 3.4.3

For further information related to this function in the web server, please refer to 3.4.3.

2.8.4 Modbus TCP Slave

→ Section 2.4

The DatumX can be configured as a ModbusTCP Slave providing an easy integration with existing data retrieval systems. It is important to mention that this method would only retrieve **instant variables**, see section 2.4.
In order to use this function, the master should be in the same LAN network as the device (either WiFi or wired).
There are some parameters that the Modbus master needs for data retrieval:

1. Function Code: 3 or 4
2. Ipv4 address: 192.168.10.3 (Ethernet) or 192.168.100.1 (WiFi)
3. Address: 0

There are some registries associated to each variable extracted, which retrieve all information related to it. Variables retrieved by modbus tcp are structured in blocks of **6 registries**, where the first 2 refer to its value, the next 2 show its current state, the following registry retrieves any error state, and the last one refers to any alert status (see Table 2.20).

→ Figure 2.4, Table 2.20

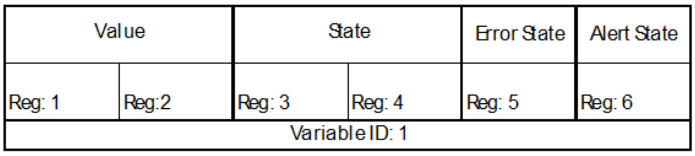
This is better seen in Figure 2.4.

By applying Equation 2.2, any desired registry can be obtained for every **Variable ID**. The present a variable can have different values according to the desired registry (see Table 2.10). Also, consider that $VariableID$ refers to the unique identifier of each configured variable in the device.

→ Table 2.10

Lastly, it is also necessary to consider each registry endianness, which can be

Figure 2.4: Modbus Registries Structure



Value		State		Error State	Alert State
Reg: 1	Reg:2	Reg: 3	Reg: 4	Reg: 5	Reg: 6
Variable ID: 1					

found in Table 2.10 with all relevant information about all available registries.

$$VariableRegistry = (VariableID * 6) - a \tag{2.2}$$

Table 2.10: Registry Details

“a” Value	Registry	Endianness	Data Type
5	Value	little endian	float
3	Status	little endian	unsigned long int
1	Error Status	big endian	unsigned int
0	Alert Status	big endian	unsigned int

→ Table 2.11

In case of this data retrieval option, there are just a couple parameters for users to modify conveniently. In Table 2.11, there are all configurable Modbus TCP parameters.

Table 2.11: Modbus TCP Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Port	Port for Modbus TCP communication	1 - 65532; **502**
Close Port Automatically	Enables or disables Modbus communication automatically	**Disabled**; Enabled

2.8.5 Socket

This open socket allows another data retrieval possibility for third-party applications through a TCP Connection.

→ Sections 2.4 & 2.3

It is only necessary to request connection in the port **35.000** for the device to respond with both the Instant Values and Monitor Variables (See Sections 2.4 and 2.3) and then close the connection.

The output is given in **ASCII** format, and each line has the following structure:

```
Monitor value 1, Monitor Value 2, ..., Monitor Value 13
ID Variable 1; Value; measurement status; error; alert status; Name, Units
ID Variable 2; Value; measurement status; error; alert status; Name, Units
...
ID Variable N; Value; measurement status; error; alert status; Name, Units
```

→ Tables 2.2 & 2.20

In the previous example, the first row corresponds to all system monitored variables (see Table 2.2). Then, the next rows show **All** ”N” variables configured with its name, units, value and different status codes (known as instant variables, See section 2.4). If a variable alert is configured, the consequent status codes are found in Table 2.20.

The only parameter changeable by the user is the selected **port** for the connection, being the port 35.000 the **default value**.

2.8.6 API REST

The datalogger DatumX allows third-party applications to request outputs and information through its API REST. The message bodies use the **JSON syntax** and every HTTP request must include the **Content-type: application/json** header.

→ Table 2.14

Of course, several relevant HTTP status codes exist in the response's body, see the error reference for details (Table 2.14).
There are values that must be replaced by valid values. For instance, `token` is a parameter that you need to replace with the actual token retrieved with the correct method.
The methods described in this documentation may be expanded in the future. This documentation can also be consulted on the following link: **documenter.getpostman.com**

POST: Request Bearer Token

```
http://192.168.100.1/api/users/auth
```

Datum X API REST implements bearer token authorization, each token lasts one hour before **expiring**. Every other method found here needs this token in order to respond correctly. Otherwise, an error message will be returned. Credentials shown in the request body are set by default in every device:

```
{
    ”username”: ”datalogger”,
    ”password”: ”xactus.mediciones”
}
```

The example below shows a typical response:

```
{
    ”access_token”: ”eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJl
    OWViM2EzZi0yNWUyLTRkZGItYmY1ZS1lYWZmMTdlMGU0N2EiLCJleHAiOjE2ODYxNj
    c4NzAsImZyZXNoIjpmYWxzZSwiaWF0IjoxNjg2MTY0MjcwLCJuYmYiOjE2ODYxNjQy
    NzAsImlkZW50aXR5IjoiZGF0YWxvZ2dlciIsInR5cGUiOiJhY2Nlc3MifQ.25pL7o-
    WAnj8xsWecbEo64ZDutB-T52SGBCxhAEUncI”
}
```

GET: Information

```
http://192.168.100.1/api/information
```

Basic information about the device would be obtained through this method. Here, you can double check seial number, board type, image date, among other important features.

→ Section 2.8.6

The authorization must include the **bearer token** obtained in Section 2.8.6. Most of other methods work the same way.
A typical response would be similar to:

```
{
    ”Api_version”: ”V 1.0.0”,
    ”Board”: ”DatumX (Black)”,
    ”Firmware_date”: ”230525_0927”,
    ”Firmware_version”: ”V 1.1.1”,
    ”Image_date”: ”230525_0927”,
    ”Lot”: ”23A”,
    ”Serial”: ”CE0000”
}
```

GET: Instant Data

```
http://192.168.100.1/api/instant_data
```

→ Section 2.4 & Table 2.20

Instant values can be retrieved through this method. Variables are ordered by its assigned **Variable ID**. Besides value, instant data reports important features as well, as units, variable name, state, among others. If an alert is configured in any variable, the corresponding alert status will appear as an **”Alert”** value. The different alert states are described in Table 2.20.
As seen before, this method just require **bearer token** authorization. The example below shows a typical response:

```
{
    ”vars”: {
        ”1”: {
            ”Alert”: 0,
            ”Error”: 0,
            ”Name”: ”Main Power Supply”,
            ”State”: 0,
            ”Unit”: ”V”,
            ”Value”: 12.254687
        },
        ”2”: {
            ”Alert”: 0,
            ”Error”: 0,
            ”Name”: ”Voltaje línea 3.3V”,
            ”State”: 0,
            ”Unit”: ”V”,
            ”Value”: 3.312891
        },
        ”3”: {
            ”Alert”: 0,
            ”Error”: 0,
            ”Name”: ”Voltaje línea 5V”,
            ”State”: 0,
            ”Unit”: ”V”,
            ”Value”: 4.995117
        }
    }
}
```

POST: Historic Data

```
http://192.168.100.1/api/historic_data
```

Historic data stored in the device’s database can be retrieved using this method. It is necessary to include **start** and **end** dates to filter output as well as the bearer token. This is a typical body:

```
{
    ”start_time” : ”2023-06-07 14:10:00”,
    ”end_time” : ”2023-06-07 14:20:00”
}
```

In the example below, it is possible to identify variables by their configured **Variable ID**, followed by the data ordered by date. It also returns variable names.

```
{
    ”Historic_data”: {
      ”1”: [
            [
                ”2023-06-07 14:10:00”,
                12.2547
            ],
            [
                ”2023-06-07 14:20:00”,
                12.2891
            ]
        ],
      ”2”: [
            [
                ”2023-06-07 14:10:00”,
                3.31289
            ],
            [
                ”2023-06-07 14:20:00”,
                3.31934
            ]
            ]
        },
    ”Names”: {
      ”1”: [
                ”Main Power Supply”,
                ”V”
           ],
      ”2”: [
                ”Line Voltage 3.3V”,
                ”V”
           ]
    }
}
```

GET: Monitor

```
http://192.168.100.1/api/monitor
```

→ Section 2.3

All internal variables monitor by the monitor feature discussed in Section 2.3 can be retrieved by this method. As previous methods, monitor separates its output into two separate groups, **Network** which shows information about network status, and **System**, that measures different voltages, temperatures, and other vital variables for the device.
As usual, this method requires the bearer token authorization.
A typical response might look like the example below:

```
{
    ”network”: {
        ”Data usage”: ”0”,
        ”Data usage alert”: ”False”,
        ”Keepalive ping”: ”0”,
        ”LAN IP”: ”NULL”,
        ”Last data sending”: ”NULL”,
        ”Network status”: ”4”,
        ”Signal strength”: ”0”,
        ”WAN IP”: ”NULL”
    },
    ”system”: {
        ”Board temp”: 31.75,
        ”CPU temp”: 41.8,
        ”CPU usage”: 3.6,
        ”DB disk usage”: 9.3,
        ”RAM usage”: -9999,
        ”V12”: 12.1354,
        ”V3.3”: 3.3129,
        ”V5”: 4.9951,
        ”Vin”: 12.2891
    }
}
```

GET: Logs

```
http://192.168.100.1/api/logs?from=true&lines=10
```

→ Section 2.2

As described in Section 2.2, this method allows to retrieve both **transmission** and **system** logs.
Besides the required authorization bearer token, there are two important parameters to define. The parameter **from** determines if the logs shown will start from the beginning or the end of the list, and the default value **true** means that the log list will start from the most recent logs. Also, the parameter **lines** specifies the number of desired rows, by default 10.
Thus, a typical output of this method would be the following:

```
{
    ”logger”: [
      ”2023-06-07 13:57:13,917 - POWER: 12V line power ON.”,
      ”2023-06-07 13:57:13,917 - MAIN: STOP_FLAG: measurement STARTED.”,
      ”2023-06-07 13:57:14,841 - SDI-12: no devices were detected.”,
      ”2023-06-07 13:57:18,630 - MONITOR: checking thread started.”,
],
    ”transmisor”: [
      ”2023-06-07 13:57:14,350 - MAIN: automatic trigger active.”,
      ”2023-06-07 13:57:14,352 - WiFi MONITOR: active. ”,
      ”2023-06-07 13:57:14,764 - FILESYSTEM: protocol loaded: ftp.comms.pyc”,
      ”2023-06-07 13:57:14,776 - FILESYSTEM: protocol loaded: lora.comms.pyc”,
      ”2023-06-07 13:57:14,777 - NET_MONITOR: MODEM: powering up. ”,
    ]
}
```

GET: Restart

`http://192.168.100.1/api/restart`

Some troubleshooting procedures might need to restart the device. In order to allow third-party applications to apply these kind of procedures, the restart feature is intended to perform a device restart remotely. There will be a response message indicating the system reboot in the next 5 seconds. Of course, the authorization bearer token is needed.

GET: Shutdown

`http://192.168.100.1/api/shutdown`

WARNING! →

This method performs a **system shutdown** remotely. It should be used carefully, as the device **WON'T** turn on without physical assistance plugging the power cable again.
As the previous method, Shutdown will show a message before the system turning off and the authorization bearer token must have been given.

Complementary Information

→ Tables 2.12 and 2.13

DatumX API REST has different response times according to the feature used. The Table 2.12, shows typical response times for every method described.
It is possible to use the API REST with different users (tokens) at once. However, more users making requests will **increase** response times. The Table 2.13, shows response times for different number of users in most used methods.

→ Table 2.14

There are several **error codes** that might appear in response. These error will show a little further information related with the error faced. In Table 2.14, there are all available error codes.

Table 2.12: Method Average Response Times

Method	Response Time [ms]
Request Bearer Token	1000
Historic Data	1100
Logs	40
Instant Data	1100
Monitor	1100
Information	30
Restart	40
Shutdown	40

Table 2.13: Method Average Response Times by users

Method	Response Time [ms]	Users
Instant Data	1100	1
Instant Data	2500	5
Instant Data	3000	10
Monitor	1100	1
Monitor	2500	5
Monitor	3000	10
Information	30	1
Information	60	5
Information	100	10

Table 2.14: API REST Error Codes

Error	Description
200	The request was made succesfully
400	Invalid Entry
401	Invalid User/Password
422	Bearer token introduced badly

2.8.7 FTP Transmission

→ Table 2.15

It is possible to send data to a FTP server through this function. It is only necessary to set the parameters present in Table 2.15. The files sent have the same structure as the generated by the **CSV File Generator**, see section 2.8.2.

→ Section 3.5.2

For more information about FTP Transmission configuration, please refer to 3.5.2.

Table 2.15: FTP Transmission Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Enable FTP Transmission	Enables or disables FTP Transmission	Enable, **Disable**
Server IP	FTP server's IP address	Any valid IP address
Server Port	FTP server's port	1 - 65.535; **21**
Server Username	FTP server's username	Non-Applicable
Server Password	FTP server's password	Non-Applicable
Enable TLS	Enables the transport layer security (TLS)	Enable, **Disable**
Transmission Rate	Defines the transmission period in seconds	30 - 87000; **1800**
Transmission Rate Alert	Defines the transmission period during an alert state in seconds	0 - 3600; **60**
Server Path	Desired path for the file to be transfered	**"/"**
File Name	Desired name for the file to be transfered	**serial**, alt_serial, string
Transfer Timeout	Waiting timeout for FTP connection closure	1 - 60; **10**

Particularly, the **File Name** parameter allows to define the name of the CSV files sent through this transmission, having three possible options: **"serial"** (Device's serial number), **"alt_serial"** (User's alternative serial), or any **string** defined by the user. However, even though the CSV file contents are changeable, every CSV file's name will have the same structure as follows:

```
yyyyMMddhhmmss_File Name.csv
```

Alert Messages: During alert status, every alert status change generate **new files** that are immediately transmitted. These alert FTP Transmission files differ in both content and name, as they have a different purpose. A typical alert file name would have the following structure:

```
ALERT_yyMMddhhmmss_Serial Number.csv
```

→ Section 2.9 & Table 2.20

Thus, as described in Section 2.9 and Table 2.20, alert status codes are included in the CSV file with the following structure:

```
Serial Number, yyyy-MM-dd hh:mm:ss
Variable ID, Value, Alert Status Code
```

→ Section 2.9, Table 2.22

In case of a **Monitor Alert** (see Section 2.9), the status codes will correspond to the ones found in Table 2.22 and the **Variable ID** will be always be **0**.

2.8.8 HTTP-JSON Post

DatumX has the capability of making transmissions to a third party API REST by doing a HTTP post. In general, the output format is JSON and has the possibility to use a bearer token as a security authentication method.
The third party API just has to fulfill some requirements. First of all, the method requires the necessary **endpoint** to send the data (it could send alerts data to a different one if required). Next, if token authentication is used, it is necessary to provide both user and password or token retrieval.

→ Table 2.16

Table 2.16, shows in further detail all parameters related with this feature.

Table 2.16: HTTP-JSON Post Parameters (Default value is **highlighted**)

Parameter	**Description**	**Possible Values**
Enable HTTP-JSON	Enables or disables HTTP-JSON Transmission	Enable, **Disable**
Service URL	Service's endpoint for data transfer	Any valid URL address
Alerts Service URL	Service's endpoint for alert data transfer	Any valid URL address
Authentication Service URL	Service's enfpoint for token request	Any valid URL address
Clients ID	Username for bearer token request	Any valid username
Client Secret	Password for bearer token request	Any valid password
Transmission Rate	Defines the transmission period in seconds	30 - 87000; **1800**
Transmission Rate Alert	Defines the transmission period during an alert state in seconds	0 - 3600; **60**
Certificates Verification	Enables or disables certificates verification	**Enable**, Disable
Transfer Timeout	Waiting timeout for FTP connection closure	1 - 60; **20**

If token authentication is used, the request will be made with the following structure:

```
{
    ”grant_type” : ”client_credentials”,
    ”client_id” : ”username”,
    ”client_secret” : ”password”
}
```

It is important to point out that this transmission method has a fixed output structure, being necessary for the receiving API to adapt to it. Both parameters of username and password have the respective keys of **client id** and **client secret**. Hence, the service must identify them as the credentials for token retrieval.
Once a token is retrieved correctly, each transmission will have the bearer token in the header.

→ Section 2.8.6

In the case of the measured data, the output structure is quite similar to the method presented in the device's API REST section 2.8.6, being **3** important keys (**Serial**, **Names**, **Data**):

```
{
    ”serial”:”CE0000”,
    ”Names”:{
        ”14”: [”CO2”, ”ppm”],
        ”13”: [”Humidity”, ”%”],
        ”5”: [”Main Board Temperature”, ”°C”],
        ”12”: [”Ambient Temperature”, ”°C”]
    },
    ”data”:{
        ”14”: [[”2024-10-10 09:20:00”, 1351.0]],
        ”13”: [[”2024-10-10 09:20:00”, 80]],
        ”5”: [[”2024-10-10 09:20:00”, 38.0]],
        ”12”: [[”2024-10-10 09:20:00”, 23.45]]
    }
}
```

As described, the **Serial** key will contain the serial number of the device, **Names** key will enumerate all configured variables by **Variable ID** showing both each variable name and units, and lastly, the **Data** key will return each variable id measured data with its timestamp.

→ Table 2.20

In case of alert status, the device will transmit variable alert status codes(see table 2.20) with the following structure:

```
{
    ”serial”:”CE0000”,
    ”datetime”:”2024-10-10 10:08:29”,
    ”variable”:1,
    ”value”:12.220312,
    ”alert_status”:6
}
```

2.8.9 Xactus TCP Transmission

By default, DatumX sends its collected data to a cloud server, which allows our proprietary platform **Intelity** to offer further analysis and services.
There are a few configurable parameters for the user to change conveniently, however, it is recommended to change them **only** if necessary.

→ Table 2.17

Please, refer to Table 2.17 for further detail.
All factory configured parameters are recommended and personalized for each requirement. For instance, **Remote Configuration** parameter allows Xactus to perform remote updates into the device's software without interrupting operations. Thus, it is important to avoid any changes without consulting our support team.

Table 2.17: Xactus TCP Transmission Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Enable TCP/IP Transmission	Enables or disables Xactus TCP transmission	**Enabled**; Disabled
Transmission Rate	Transmission rate in seconds	30 - 87.000 sec. ; **1.800**
Tranmission Rate Alert	Transmission rate during an alert in seconds	0 - 3.600 sec. ; **60**
Remote Configuration	Enables or disables remote configuration (do **NOT** change)	**Enabled**, Disabled

2.8.10 LoRaWAN Transmission

According to the environmental conditions, it might be necessary to use alternative transmission means different from cellular network. Thus, it is possible to connect the device through **LoRaWAN**.

WARNING! → Even though DatumX has programmed all necessary functionalities related to LoRaWAN transmission, it needs **additional hardware** in order to operate correctly.

→ Section 5.3 Please, contact us to request the LoRaWAN antenna and gateway for your device. See section 5.3.

WARNING! → It is also important to note that LoRaWAN Transmission functionality **only** transmits to Xactus cloud server, and that transmission to any other server it is not supported.

Nonetheless, there are a few configuration parameters for the user to change according to the present needs.

→ Table 2.18 Please, refer to Table 2.18 for further detail.

Table 2.18: LoRaWAN Transmission Parameters (Default value is **highlighted**)

Parameter	**Description**	**Possible Values**
Enable LoRaWAN	Enables or disables LoRaWAN transmission	**Disabled**; Enabled
Transmission Rate	Transmission rate in seconds	30 - 87.000 sec. ; **600**
Tranmission Rate Alert	Transmission rate during an alert in seconds	0 - 3.600 sec. ; **200**
Baudrate	Defines LoRaWAN transmission baudrate	1.200 - 1'000.000 bauds; **115.200**
Timeout	Defines LoRaWAN transmission time out in seconds	0 - 3.600 sec.; **1**

2.9 Alerts

There are two specific situations in which an alert state can be triggered. The first one, occurs when a **variable exceeds** predefined boundaries. In the other hand, a connected and configured **intrusion sensor** would trigger the alert. Consequently, any triggered alert state will activate several things. First, it will activate a **relay** and **Audio** outputs. Also, it will shows messages on both **Intelity** (if the DatumX is connected to the cloud) and its built-in **Web Server**. Lastly, it will send **alert messages** through all configured transmissions (as FTP Transmissions, HTTP JSON Post, or the proprietary transmission).

Variable Alerts

It is possible to generate alerts for one or more variables. Hence, each variable has to be configured to trigger an alert.

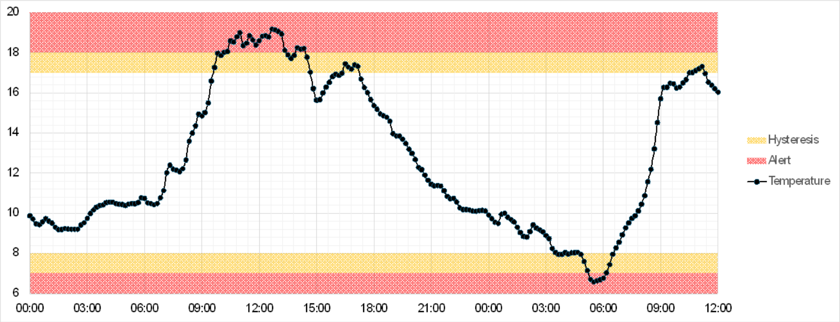
Variable alerts can be **separated** into levels, which were thought to be used as **alert priorities** or similar purposes, useful for many applications. DatumX provides the possibility to choose if a defined boundary will trigger a medium or high priority alert. Of course, there's also the option to work with just one priority level alert. Both of these priority levels are named **RED** as high priority alert and **ORANGE** as medium priority alert.

Parameters: There are many parameters to adjust alerts to each individual requirement. Hence, parameters as **Persistence** and **Hysteresis** help avoiding multiple alert triggering due to noisy readings. The **Alert Origin** is useful depending on the configured sample time and the fastness needed for an alert activation in response to a variable behavior. All configuration parameters affect **both** RED and ORANGE alerts.

→ Table 2.19 & Figure 2.5

Every alert parameter is found in Table 2.19.
In order to completely understand variable alerts functioning, in Figure 2.5, there is an example of an alarm configured with both upper limit and lower limit **hysteresis**, allowing the alert to stay triggered even though the variable value went down the upper limit but is still higher than the upper limit hysteresis. Thus, Hysteresis is expressed in a percentage value, being proportional to the defined limit (either upper or lower). Then, hysteresis is calculated with the Equation 2.3.

Figure 2.5: Example of Hysteresis Configuration. The alert should trigger at the moment the temperature pass 18°C. However, alert state stops as soon as temperature levels are lower than 17°C (Around 15:00 hours). This same behavior occurs when temperature is lower than 7°C, but hysteresis keep alert status until temperature reaches above 8°C.



$$H = L * \frac{1 - H_p}{100} \tag{2.3}$$

Where H is either upper or lower limit hysteresis, L is either upper or lower limit, and H_p is hysteresis percentage.
The example also contemplates default values for some of the parameters, as persistence and alert origin.

Alert Messages:

Every time that an alert is **triggered** or its **status** changes, a particular message per alert configured is sent through all configured transmissions. Hence, DatumX possess different variable alert status codes, so users can easily interpret the current state of a variable alert throughout the different communication options available. Of course, depending on the transmission configured, the output messages will differ slightly, however the status codes will remain the same.

→ Table 2.20

Variable alert status codes are described in Table 2.20.

Intrusion Alerts

In order to start generating intrusion alerts with the DatumX, it is necessary to connect and configure an intrusion sensor.

WARNING! →

It is important to point out that the sensor **MUST** use digital pulse interface (any of the 2 existent channels) and suit to its electrical levels.
After the external intrusion sensor is properly configured, there are some important things to consider about intrusion alerts. The DatumX divides intrusion alerts into **three** different states. The first state contemplates the **alarm armed** and is listening to the intrusion sensor. The next state is triggered once a detection is made from the sensor. The device will **wait** a defined time span ($T_{Waiting}$) for the actual alert to trigger, this time is intended for alert deactivation for the user. Once the

Table 2.19: Alert Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Enable Outputs	Enables both audio and relay outputs	**Disabled**; Enabled
Activation Period	duration of relay and audio outputs during alert state	0 - 20.000 sec. **(10)**
Alert Origin	Defines if alert state is triggered by the sample or the sub-sample	**Samples**; Sub-Samples
Sample time	Defines a new sample time in alert state	defined by Synchronous or Asynchronous sample time rules, see Section 2.1.1
Upper Limit: RED	Superior alert activation value	-99.999 - 99.999; **(10.000)**
Lower Limit: RED	Inferior alert activation value	-99.999 - 99.999; **(-10.000)**
Upper Limit: ORANGE	Superior alert activation value	-99.999 - 99.999; **(10.000)**
Lower Limit: ORANGE	Inferior alert activation value	-99.999 - 99.999; **(-10.000)**
Upper Hysteresis	Upper limit reduction by percentage. It turns off the alert state once crossed. See Equation 2.3	**0** - 100
Lower Hysteresis	Lower limit increase by percentage. It turns off the alert state once crossed. See Equation 2.3	**0** - 100
Persistence	Samples measured inside/outhside limits to turn on/off alert state (considering **hysteresis**)	**0** - 1800, Depending on **Alert Origin**

Table 2.20: Variable Alert Status Codes

Status	Description
0	Normal State
1	Upper Limit: Orange persistence
2	Upper Limit: Orange persistence return
3	Upper Limit: Red persistence
4	Upper Limit: Red persistence return
5	Upper Limit: Orange
6	Upper Limit: Red
7	Lower Limit: Orange persistence
8	Lower Limit: Orange persistence return
9	Lower Limit: Red persistence
10	Lower Limit: Red persistence return
11	Lower Limit: Orange
12	Lower Limit: Red

time span has passed, the device will be in **Alert State**, which will send alert messages through its configured communications, also, the audio and relay outputs will be **activated**. Then, the device will wait for either the alarm to finish (T_{Alert}) or another sensor detection which will restart that timer. It is also possible to configure the repetitiveness and frequency of alert messages during each separate alert event, please see Table 2.21.

→ Figure 2.6, Table 2.21

The whole intrusion alarm process is better explained in figure 2.6. Also, intrusion alerts parameters are found in Table 2.21.

Intrusion Alerts Deactivation: As mentioned, the intrusion alert cycle allows a time span for alert deactivation before it gets triggered. This deactivation is performed by disarming the intrusion alert on the web server. For instruction about this process, please go to section **??**.

Intrusion Alerts Status Code: It is important to point out that alert events are reported in various forms across all different communication options available. For instance, both **FTP Transmission** (Section 2.8.7) and **HTTP-JSON** (section 2.8.8) shows alert events with **status code: 50** allowing to differentiate intrusion alerts events from other alerts events, as monitor alerts (see section 2.9).

Figure 2.6: Intrusion Alert Cycle

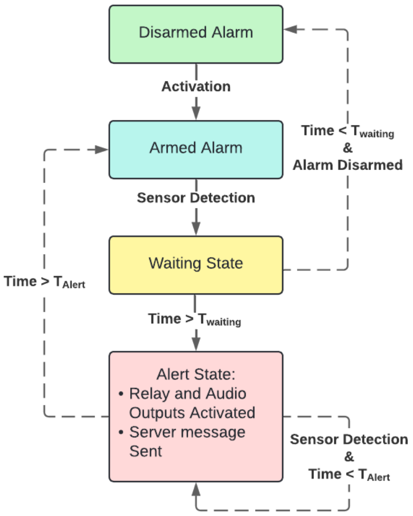


Table 2.21: Intrusion Alert Parameters (Default value is **highlighted**)

Parameter	**Description**	**Possible Values**
Waiting Time	Deactivation period, starts with sensor detection	0 - 20.000 sec. **(20)**
Alert Time	Alert period	0 - 20.000 sec. **(20)**
Activation Period	Audio and relay outputs activation period	0 - 20.000 sec. **(10)**
Repeat Messages	Enables to repeat alert messages during an alert state.	**Disabled**; Enabled
Messages Rate	Defines messages rate during a unique alert state	**1** - 60 minutes

Monitor Alerts

As mentioned in Section 2.3, The main internal variables are self-monitored by the system itself. Any detected **outlier** in some of the variables will trigger an alert and (in some cases) a consequent action. Monitor alerts are enabled by default and is not possible to disable this function.

→ Table 2.22

The Table 2.22 shows alert conditions for the internal variables that have alerts enabled. All additional actions triggered are described as well. The Table also shows **alert state codes**, which are useful for **identifying** the variable that presents a problem when using other communication means (for instance, FTP Transmission or HTTP-JSON Post).

→ Sections 2.8.7 & 2.8.8

For reference about monitor alert states in other communications, please refer to 2.8.7 and 2.8.8.

According to the Table 2.22, the alert condition corresponding to a extremely low (or high) input voltage (below 11.3 Volts, or above 15.5 Volts) has a built-in **hysteresis** function, which maintains the alert state over the defined voltage boundaries. Thus, the inpult voltage has to get **over** 11.9 Volts or **below** 15.5 Volts in order to **return** to a normal state. This is mainly for avoiding to stop all measurements repetitively.

Table 2.22: Monitor Alerts

Variable	Alert State Condition	State	Action
Power Input	<11.5 Volts, or >15.5 Volts	61	Alert state triggered
Power Input	<11.3 Volts, or >16 Volts. **Hysteresis** up to >11.9 Volts, or <15.5 Volts	61	**ALL** measurements are **stopped**. CPU is **shut down**. See Table 1.4 for error reference
5 Volts Power Output	<4.7 Volts, or >5.2 Volts	63	Alert state triggered
12 Volts Power Output	<11 Volts, or >12.4 Volts	64	Alert state triggered
CPU Temperature	>75°C	66	Alert state triggered
Data Usage	Consumption Limit	74	Alert state triggered
RAM Usage	>80%	72	Alert state triggered
Disk Data Base Usage	>80%	73	Alert state triggered
Disk Data Base Usage	>90%	73	Compress data base to a zip file, stores it into the Backups folder and creates a new data base

The monitor alerts function allows the user to set a cellular **data consumption limit**, which will trigger an alert state if trespassed. There are a few parameters that have to be set before using this function. Please, refer to Table 2.23 for details.

Additional Alert Parameters

Each transmission method has an **exclusive** parameter for changing the **transmission period** during an alert state. However, the user **MUST** be aware of its value and its implications if an alert is configured.

Table 2.23: Data Consumption Limit Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
Data Usage Limit	Maximum allowed data usage limit	1 - 999999 Mbytes; **(5)**
Enable Data Usage Monitor	Enables cellular data usage monitoring	**Disabled**; Enabled
Data plan cap	Day of the month that resets data usage	**1** - 30

Depending on the particular application, it is possible to use low transmission periods, such as 1 hour or higher time frames, being **24 hours** the limit of the device's transmission period in **normal state**. In contrast, the transmission period during an **alert state** could be up to **1 hour**, which would increase transmission traffic and carry extra transmission costs. Nonetheless, in most of the cases, it is possible to match transmission period in both states in case a transmission traffic increase is not an option.

→ Table 2.24

All exclusive transmission period parameters and its default values are in Table 2.24.

Table 2.24: Additional Transmission Period Parameters (Default value is **highlighted**)

Parameter	Description	Possible Values
FTP Transmission Rate Alert	Transmission rate during an alert state in the FTP transmission mode	1 - 3600 Seconds; **(60)**
LoRaWAN Transmission Rate Alert	Transmission rate during an alert state in the LoRaWAN transmission mode	1 - 3600 Seconds; **(60)**
HTTP-JSON Transmission Rate Alert	Transmission rate during an alert state in the HTTP-JSON transmission mode	1 - 3600 Seconds; **(60)**
Xactus Transmission Rate Alert	Transmission rate during an alert state in the main transmission mode	1 - 3600 Seconds; **(60)**

3 Configuration

There are **several** ways of configuring the DatumX, however, this chapter will **only** cover configuration through its built-in **web server**, which centralizes all functions and configuration tools in a user oriented interface.

3.1 Getting Started

Before engaging all the features and options that the DatumX web server offers, it is important to know how to have access to it and familiarize with its main characteristics.

At first, you **must** connect with the device through either WiFi or Ethernet port. Depending on that, you just need to have your computer in the same network and enter the appropriate **IP address** into a web browser.

→ Section 4.3.4

For more information about internet connection into the device, please refer to section 4.3.4.

For instance, the easiest and fastest way to connect with the DatumX is through WiFi. DatumX is always transmitting a WiFi network, so there are no extra steps needed for activating this function. Hence, please identify a network with this name **structure** among all available networks:

```
DatumX CE0000
```

The **CE0000** refers to the unique serial number of each device. The default WiFi password is the following:

```
d4t4l0gg3r
```

Once connection to WiFi is established, it is necessary to type the appropriate IP address in a web browser. In this case, it is **192.168.100.1**. If Ethernet connection is used, the default address would be **192.168.10.3**.

A login screen will appear, which requires **user name** and **password** authentication. Usually, DatumX devices are factory configured with the required users by client's request, however, there is a default user whose credentials are the following:

```
User: admin@xactus.io
Password: datalogger
```

→ Sections 3.2.4

User creation, edition, and deletion is further explored in section 3.2.4.

WARNING! →

Consider default user edition or deletion for security purposes.

After user authentication, the home screen will load. Here, there are shortcuts to the main functions of the device. For example, the **left hand panel** allows access to the different main paths, the **top right corner banner** shows important information about the current errors and alert status, date & time, system and transmission processes, and user information. There are different panels in the **home** page, showing general information about the device as *Serial Number*, API and Firmware *Versions*, etc.

→ Section 2.3

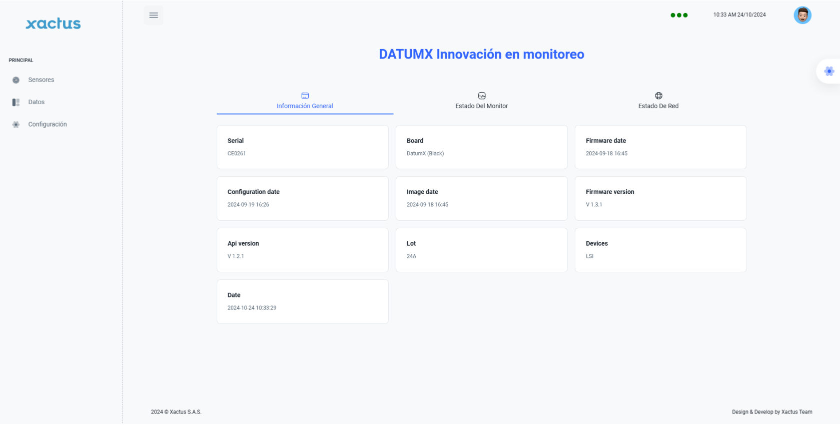
Other panel will show you the *monitor status* (see section 2.3), with important system variables.

Lastly, the third panel has information related to the network status.

→ Figure 3.1

This information is better illustrated in figure 3.1.

Figure 3.1: Home Screen



Serial
CE0261

Board
DatumX (Black)

Firmware date
2024-09-18 16:45

Configuration date
2024-09-19 16:26

Image date
2024-09-18 16:45

Firmware version
V 1.3.1

Api version
V 1.2.1

Lot
24A

Devices
LSI

Date
2024-10-24 10:33:29

3.2 Home Screen

This section covers in further detail the functions and shortcuts included in the main/home screen of the web server.

Even though the main screen is very simple and clean, there are some relevant points to explain.

The **General Information** panel, will show the basic but relevant information about the device:

1. **Serial Number**: Unique ID number of the device, allowing to quick identify the device connected.
2. **Board**: This field indicates the motherboard model.
3. **Firmware Date**: This is closely related to the *Image Date* field, being the date that the firmware version was released.
4. **Configuration Date**: Date of the last time the device was configured.
5. **Image Date**: Date when the installed software version was released.
6. **Firmware Version**: Version of the currently installed firmware.
7. **API Version**: Version of the currently installed API REST.
8. **Lot**: This field refers to the device's production year and period.
9. **Devices**: This field shows configured sensors and other devices. This field is modified during device configuration.
10. **Date**: System current date.

→ Section 1.1

→ Section 2.3

General information parameters are also reviewed in section 1.1.

Monitor Status panel gives a quick look at the system status (see section 2.3). This panel will exclude all network parameters and focus just in the internal variables. The subsequent panel, **Network Status**, will show the remaining network monitoring parameters as well as other important network information:

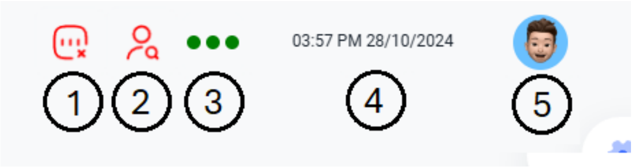
1. **IP LAN**: Local network IP address (default: **192.168.10.3**)
2. **IP WAN**: Public IP address
3. **IP WLAN**: Wireless local network IP address (default: **192.168.100.1**)

Other elements present in the main/home screen are available at any time for the user. For instance, the left side navigation panel will provide access to any section of the website in every moment. In addition, the top right corner elements are also available in every path.

→ Figure 3.2

Figure 3.2 closes up the top right corner of the home page showing the **Errors and Alerts**, **Intrusion Alerts**, **Status**, **System Date & Time**, and **User Settings** icons, the subsequent sections will further explain about these elements.

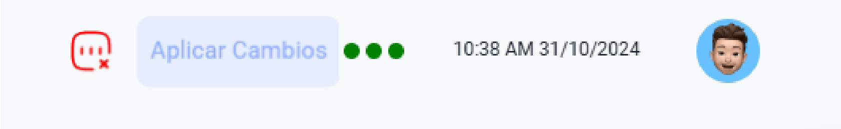
Figure 3.2: Top Right Corner Elements. 1. Errors, 2. Intrusion Alerts, 3. Status, 4. System Date and time, and 5. User



WARNING! →

From now on, all configuration changes made in the web server need at least one of the main processes **reboot** in order to be fully applied to the device. To avoid multiple reboots, any configuration changes will only apply if the user authorizes it by clicking the button present in figure 3.3.

Figure 3.3: Apply Changes Button: By clicking it, the device will apply all configuration changes made accross the web server.



3.2.1 Error and Alert States

→ Figure 3.2

Whenever an error occurs (i. e.,a sensor is not found, or a variable alert has been triggered), the error notification icon will appear in the top right corner as shown in figure 3.2, element 1. By clicking this icon, all current errors and variable alerts notifications will appear. In case that a particular error is solved, the corresponding notification will stop appearing in this section.

3.2.2 Intrusion Alerts

→ Section 2.9 and Figure 3.2

As described in section 2.9, the DatumX has an intrusion alert feature. If triggered, a notification will appear as an icon in the top right corner of the home screen, see figure 3.2, element 2.

3.2.3 Processes and Logs

In the top right corner of the screen there is a quick process status indicator, which allows to acknowledge main processes status through a simple color code.

→ Figure 3.2

As shown in figure 3.2, element 3, the status icon is formed out of three indicators that reflect current processes status. From left to right, each circle corresponds to a system process: **Logger**, **Transmission**, and **VPN**.

→ Table 3.1

Hence, table 3.1 shows the color code.

Table 3.1: Process status indicator color code

Color	Description
Green	The process is running successfully
Yellow	The process is booting
Red	The process is stopped
Orange	The process has an error
White	Waiting for current status response

→ Section 2.2

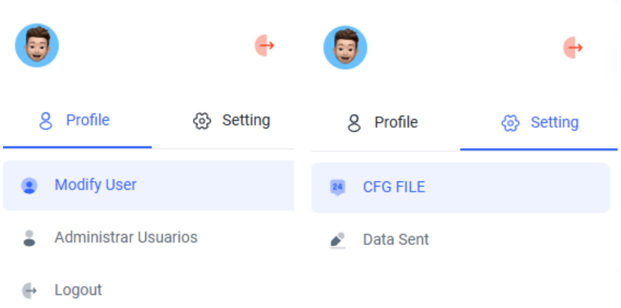
Section 2.2 has further information related to all system processes.

3.2.4 User Settings

→ Figures 3.2 & 3.4

As seen in figure 3.2, the user settings icon can be easily identified by the presence of an avatar. Also, figure 3.4 represents both sections of the user menu.

Figure 3.4: User Menu Sections: At the left, there is the profile section, having all user configuration tools. Thus, settings section in the right, provides other important device configuration tools.



This menu has some important features split into two different sections. The first one corresponds to **profile**, which provides user privileges and management tools as well as the logout button. Then, the settings section provides some other advanced device's configuration tools.

Profile Section

In the case of the **Modify User** option, the user can change its **Username**, **Email**, and **Password**.
In the other hand, the **User management** option will redirect to the users management table, where all configured users basic information and their respective roles are visible.

WARNING! →

It is important to mention that this option is **only** available for **administrator** users.

→ Table 3.2

There are a total of three role categories with different privileges each. It is possible for a user to have more than one or even all privileges. Table 3.2 has detailed information about each role.

Table 3.2: User Role Categories

Role	Description
Administrator	**Technician** and **Visualizer** privileges with user management
Technician	Complete device configuration, plus **Visualizer** privileges
Visualizer	Historic and Instant data visualization and download

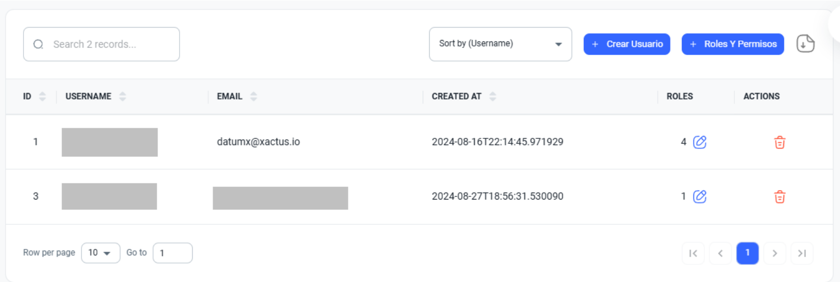
User Management: **Administrator** users have all privileges, being able to completely change device, sensors, transmission, and every other configuration parameter. Administrators are the **only** ones who can create, delete, and modify every other user including administrators.

→ Figure 3.5

Figure 3.5 shows the user management interface. Basic user information is easily identifiable in the table, as user **ID**, **Username**, **Email**, and **Date of creation**. Furthermore, the **Role** column shows how many roles a user has. Lastly, the **Actions** column allows to quick delete a user.

Apart from the columns in the table, there are other useful features in the interface as the **search bar** and the **sorting list** located at the top of the table. Finally, the **Roles and permissions** and **Create User** buttons add extra customization.

Figure 3.5: User Management Interface



ID	USERNAME	EMAIL	CREATED AT	ROLES	ACTIONS
1	[redacted]	datumx@xactus.io	2024-08-16T22:14:45.971929	4	
3	[redacted]	[redacted]	2024-08-27T18:56:31.530090	1	

Roles and Permissions: This button allows to finely customize privileges of each existing role (see table 3.2) by selecting permissions from the table 3.3.

→ Table 3.3

Thus, administrators could prevent access to certain configuration parameters to other users. This would add an extra security layer to the overall operation.

WARNING! →

Blocking access to the whole path will block access to its subsequent features even though their permission were granted. For instance, blocking access to the configuration path will block all Device, Data, Alert, Extraction, Network, and Init configuration features even if their permission is granted.

In case that modifying existing roles is not an option, administrators can easily create a new custom role according to their needs. There is a **Create Role** button inside of the roles and permissions section that will allow a new role creation.

Create User: Finally, the **Create User** button will allow to create new users from scratch by inserting basic information: **Username**, **Email**, and **Password**. Once the new user is created, it is necessary to assign roles to it by the edit roles button on the user management table.

Table 3.3: Role Permissions. Administrator users have all described permissions. Technician users have both **green** and **blue** highlighted permissions. Lastly, visualizer users have only **blue** highlighted permissions

Permissions	**Description**
User Administration	Access to the user management section
Sensors	Create and configure sensors
Configuration	Access to the configuration path
Device Configuration	General device, NTP, and maintenance configuration
Data Configuration	Access to transmission configuration (FTP, HTTP-JSON, etc.)
Alert Configuration	Variable and intrusion alerts configuration
Extraction Configuration	Extraction methods configuration (Modbus TCP, Socket, etc.)
Network Configuration	Overall network configuration
Init Configuration	Modify the configuration file
Logs	Read and download log files
System Information	Reserved for future developments
Data	Access to the data path
Data Dashboard	Reserved for future developments
Data Map	Reserved for future developments
Data Download	Download CSV data files

Settings Section

The settings section provides a couple of miscellaneous tools:

1. CFG File: This option allows to see and edit the plain text configuration file. This function is intended for advanced users that are quite familiar with the device development and it is not recommended to change this file. All other website functions are designed to modify this file automatically.
2. Data Sent: This shortcut will lead to the data sent history. This option is helpful to keep track of all values transmitted by the device.

3.3 Sensors

This is one of the main three sections of the configuration website. All main sections are accessible through the left hand banner.

→ Chapter 2 and Figure 3.6

The sensors section is intended to cover all sensor configuration described throughout the chapter 2. Figure 3.6, shows the main components of this page. Firstly, the **Configured Devices** table allows to quickly acknowledge all configured sensors while giving a shortcut to configure a new sensor from scratch and edit any of the current sensor's configuration. In addition, there is the **Connection Terminal** section, providing a general i/o overview of the device. Lastly, there's the **Channels** table, which shows all channels in the device and their current occupation.

→ Section 2.7

For more information about channels, please refer to section 2.7.

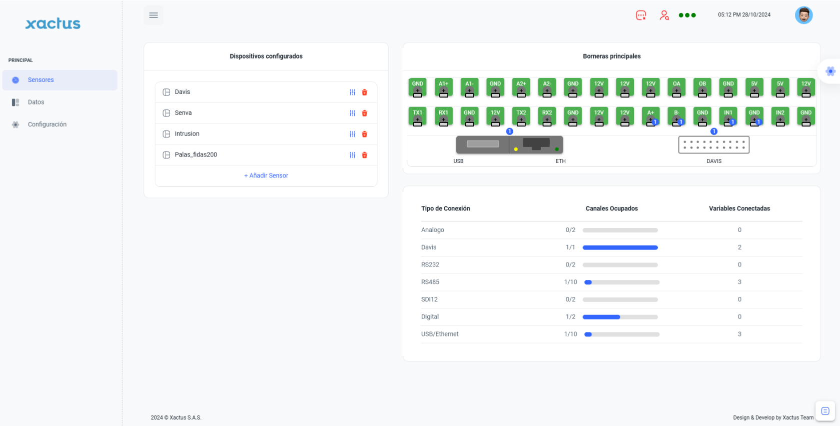
3.3.1 Configured Devices

→ Figure 3.6

As shown in figure 3.6, the **Configured Devices** table lists all devices being measured and stored by the DatumX. Additionally, device's configuration can be quickly modified through the **edit** button accessing to the **Configuration Wizard**. Lastly, there is a **delete** button in case a sensor is no longer required for monitoring.

WARNING! → Sensor deletion will **delete** all measurement data related to it in the device.

Figure 3.6: Sensors main page


xactus
PRINCIPAL
Sensores
Datos
Configuración
05:12 PM 28/10/2024
Borneras principales
GND A1+ A1- GND A2+ A2- GND 12V 12V 12V OA OB GND 5V 5V 12V
TX1 RX1 GND 12V TX2 RX2 GND 12V 12V A+ B- GND IN1 GND IN2 GND
USB
ETH
DAVIS
2024 © Xactus S.A.S.
Design & Develop by Xactus Team

Dispositivos configurados

- Davis
- Senva
- Intrusion
- Palas_fidas200

+ Añadir Sensor

Tipo de Conexión	Canales Ocupados	Variables Conectadas
Analogo	0/2	0
Davis	1/1	2
RS232	0/2	0
RS485	1/10	3
SDI12	0/2	0
Digital	1/2	0
USB/Ethernet	1/10	3

Add a Sensor

→ Figure 3.6

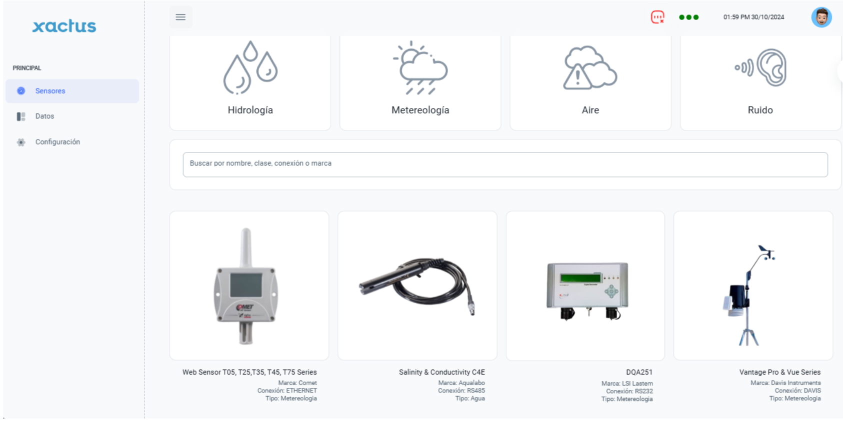
→ Section 2.7

Down in the **Configured Devices** section of the sensors page (see figure 3.6), there is an **Add Sensor** button. By clicking it, the page will turn into the **sensor catalog**. The sensor catalog screen combines both software **integrated sensors** and all **communication interfaces** (see section 2.7). Thus, users could filter items either by its purpose, such as **Hydrology**, **Meteorology**, **Air**, and **Noise**, or by the search bar.

→ Figure 3.7

Figure 3.7, shows better the sensor catalog section.

Figure 3.7: Sensor Catalog



Web Sensor T05, T25,T35, T45, T75 Series
Marca: Comet
Conexión: ETHERNET
Tipo: Metereologia

Salinity & Conductivity C4E
Marca: Aqualabo
Conexión: RS485
Tipo: Agua

DQA251
Marca: LSI Lastem
Conexión: RS232
Tipo: Metereologia

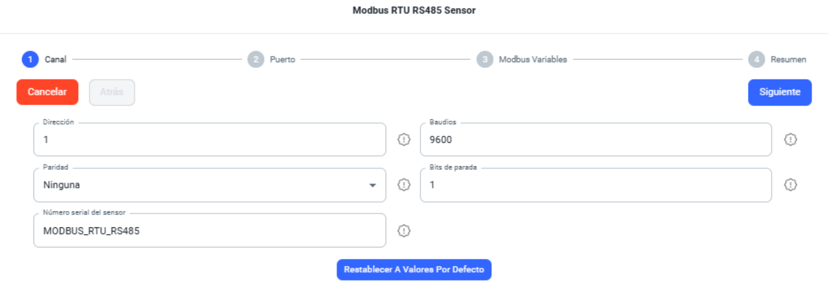
Vantage Pro & Vue Series
Marca: Davis Instruments
Conexión: DAVIS
Tipo: Metereologia

By clicking any sensor or interface present in the sensor catalog, the **configuration wizard** will open. This tool will help the user in every configuration step depending on the option selected at first instance. The steps present in the wizard may vary depending on the sensors or communication interface selected. However, it will show all relevant configuration parameters required for a proper operation. In addition, it is possible to get back to previous steps in order to correct or change parameters, the wizard will put a default value (if possible) in any unfilled field.

→ Figure 3.8

Figure 3.8 shows an overview of the configuration wizard.

Figure 3.8: Configuration Wizard: This example shows the configuration steps of a **Modbus RTU** sensor, being channel, port, and variables configuration.



Dirección: 1

Baudios: 9600

Paridad: Ninguna

Bits de parada: 1

Número serial del sensor: MODBUS_RTU_RS485

→ Table 3.4

→ Section 2.9

In case of an integrated sensor, the wizard will allow to select desired variables for the DatumX to store and monitor. On the other hand, if the wizard is configuring a communication interface, there will be an option to create a variable from scratch. Nonetheless, all variables will have the parameters present in table 3.4.
Apart from the described parameters, there might be additional parameters according to the communication interface used (i. e.,data endianness in modbus communication). The configuration wizard will also provide all parameters and configuration related to **variable alerts** (see section 2.9).

Table 3.4: Variable Configuration Parameters (Default value is **highlighted**)

Parameter	**Description**	**Possible Values**
Variable Name	Descriptive name of a variable	Non-Applicable
Variable Units	Measurement units of the variable	Non-Applicable
Sampling Time	Rate at which a variable will be measured	See table 2.1; **10 minutes**
Sub-samples	Amount of sub-samples for each sample	See section 2.1; **10**
Function	Function applied to all sub-samples	See section 2.1; **Arithmetic Mean**
Span	Part of the span and offset calibration parameters	**1**
Offset	Part of the span and offset calibration parameters	**0**
Ignore Errors	Enables error ignoring features	Enabled; **Disabled**
Ignore Instant Values	Ignore any sub-sample below the value	**-99999999**
Ignore Values	Ignore any sample below the value	**-99999999**
Valid Sub samples percentage	Percentage of valid sub-samples; see section 2.1.2	**0** - 100 %
Std Dev Threshold	Standard deviation threshold for valid samples; see section 2.1.2	**9999999**

→ Appendix A

For further information about the software integrated sensors, please refer to appendix A.

3.4 Data

This is the second main section accessible through the left hand banner. This section is oriented to visualize and download all the data gathered by the DatumX.

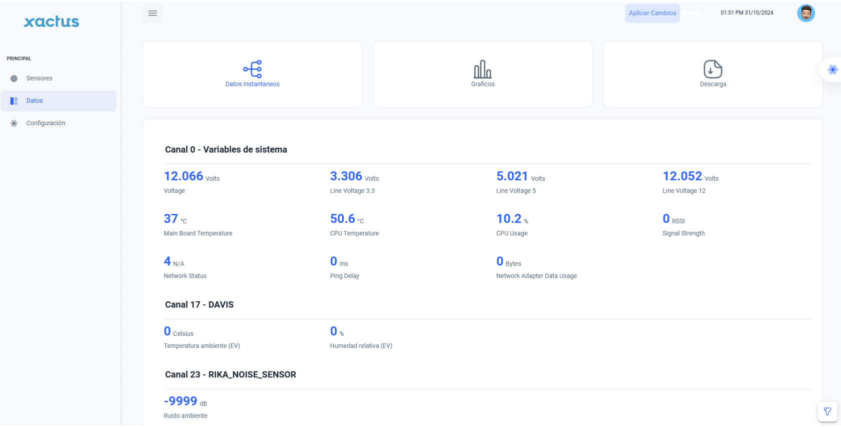
→ Sections 2.4 2.5

There are three main sub-sections covering **Instant Data** (see section 2.4), **Graphics** (see section 2.5), and **Downloads**.

→ Figure 3.9

Figure 3.9 shows the data page, which displays **Instant Data** as its default option. However, there are three buttons that toggle between the different sections across the data page.

Figure 3.9: Instant Data



Canal 0 - Variables de sistema

12.066 Volts
Voltage

3.306 Volts
Line Voltage 3.3

5.021 Volts
Line Voltage 5

12.052 Volts
Line Voltage 12

37 °C
Main Board Temperature

50.6 °C
CPU Temperature

10.2 %
CPU Usage

0 RSSI
Signal Strength

4 N/A
Network Status

0 ms
Ping Delay

0 Bytes
Network Adapter Data Usage

Canal 17 - DAVIS

0 Celsius
Temperatura ambiente (EV)

0 %
Humedad relativa (EV)

Canal 23 - RIKA_NOISE_SENSOR

-9999 dB
Ruido ambiente

3.4.1 Instant Data

→ Section 2.1 Figure 3.9

This section provides a visual approach of all data being measured in **real-time** for the DatumX. Hence, all data visualized in this section corresponds and changes according to each **sub-sample** time (see section 2.1). The main instant data page is shown in figure 3.9.

→ Section 2.7

The measured variables are clustered according to its corresponding *channel* (see section 2.7). Thus, the first channel shown would be the *Channel 0* (System Variables). Nonetheless, there's a filter button at the low right corner that allows to select which channels and variables to display for a better experience.

3.4.2 Graphics

This section provides a graphic observation of all historic values measured. As described in previous section (3.4.1), variables are ordered by **channel** and **variable ID**. Hence, the first variables displayed will be the **system variables**. In addition, this section provides a **channel**, **variable**, and **date time** filters, allowing to analyze a specific time span. This filter is accessible through the button located in the bottom right corner of the screen.

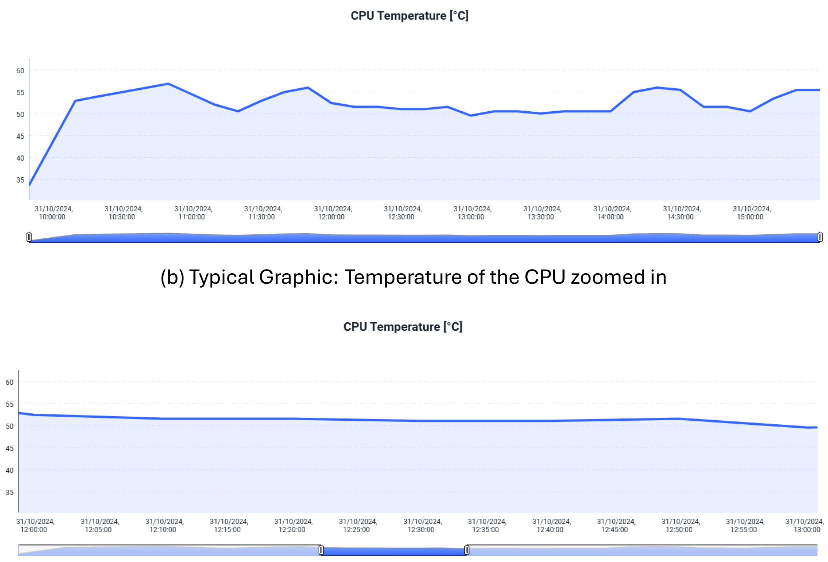
→ Figure 3.10

Figure 3.10 shows a typical graphic from this section. This example pictures one of the system variables during some hours of monitoring every 5 minutes. In addition, the slide bar below the graphic functions as a zoom for better inspection.

3.4.3 Download

This is the last section present in the data path, which is reserved for historic data downloading. Hence, the user interface shows two elements, being the **date filter**

Figure 3.10: Typical Graphic: Temperature of the CPU



(a) Typical Graphic: Temperature of the CPU zoomed out

(b) Typical Graphic: Temperature of the CPU zoomed in

and the **Download button**. Once the date filters are defined and the download button clicked, a third element will appear indicating the **status** of file preparation prior to file download. The status bar will display time left for the file to be fully prepared and the download button will be available for file downloading.

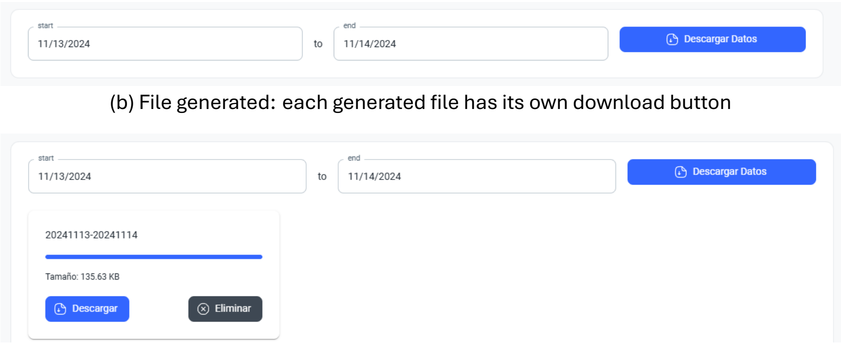
→ Figure 3.11

Figure 3.11, explains better this information.

→ Sections 2.8.2 & 3.5.2

It is important to mention that all generated file's structure is defined according to the **CSV file generator** configuration (see section 2.8.2), which is accessible through **Settings** path (section 3.5.2).

Figure 3.11: CSV File Download



(a) No files generated

start: 11/13/2024 to end: 11/14/2024

(b) File generated: each generated file has its own download button

start: 11/13/2024 to end: 11/14/2024

20241113-20241114

Tamaño: 135.63 KB

3.5 Settings

→ Chapter 2

This is the last of the three main sections accessible from the left hand banner. The settings section covers a wide variety of device's configuration parameters divided into five sub-sections that correspond to some of the features explained in chapter 2.

By default, the main screen of the settings section will show the device's **general**

configuration sub-section. However, at the top of the screen, there will be five buttons to swap between all configuration sub-sections.
From now on, all other data settings sub-sections have almost the same layout. With sub-sections divided in buttons at the top part of the screen and specific functionalities or parameters gathered in categories at the center part of the page.

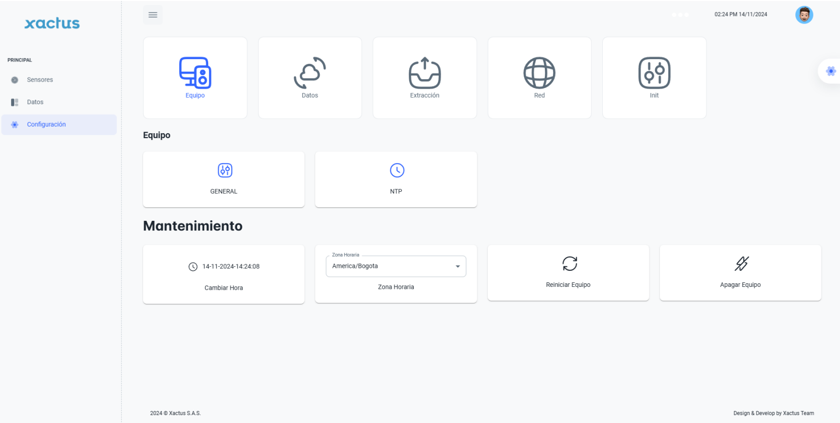
3.5.1 Device Settings

As mentioned, this is the default view of the settings path. This sub-section has some basic parameters an general functions of the device, starting with two shortcuts to **NTP** (see section 2.6) and **General Configuration** (section 3.5.1). In addition, there are **maintenance functions** handy for the user to quickly use.

→ Sections 2.6, 3.5.1, and Figure 3.12

Figure 3.12 displays the device settings page.

Figure 3.12: Settings: Device Settings



14-11-2024-14:24:08

Zona Horaria: America/Bogota

General

General configuration allows to define the following parameters: **Alternative Serial**, **Device's Name**, **Latitude** and **Longitude** coordinates.

NTP

This section allows to configure the NTP system of the DatumX. Hence, it is possible to define the following parameters: **NTP Server**, **NTP Synchronization Rate** and **NTP Protocol Version**.

WARNING! → Usually, these parameters are configured by default and there is no need of changing them. However, some requirements may need to.

Maintenance

This section covers relevant device's actions or features, these functions are easily triggered by a button and are described as follows:

1. **Change date and time**: It changes and updates date and time of the device. See section 2.6
2. **Timezone**: Allows to quickly change the timezone of the device. See section 2.6

→ Section 2.6

3. **Restart**: Reset the device

4. **Turn off**: Turns off the device. After using this function it will be necessary to physically plug the device to the power network.

3.5.2 Data Settings

→ Sections 3.5.1 2.8.2 2.8 and 2.9

Similarly to the previous sub-section (3.5.1), the data sub-section focus its configuration parameters and features in **data preparation** and **transmission**, including **intrusion alerts** as well. See sections 2.8.2, 2.8, and 2.9.

General

→ Section 2.8 and Table 2.6

General data settings is the first section available among all of the different features present. This section allows to configure a couple of important parameters: **Automathic Data Transmission** and **Max Rows per Package**. See section 2.8 and table 2.6.

CSV File Generator

→ Section 2.8.2 and Table 2.8

Next, there is the button for accessing all configuration related to the **CSV File Generator** (see section 2.8.2). Here, there is a dedicated user interface for selecting all parameters present in table 2.8.

Data Transmission

→ Sections 2.8.7 2.8.8 2.8.9 and 2.8.10

This section covers four different transmissions divided into buttons. Specifically, each button corresponds to: **FTP**, **HTTP-JSON Post**, **Xactus TCP**, and **LoRaWAN** transmissions. See sections 2.8.7, 2.8.8, 2.8.9, and 2.8.10.

→ Section 2.8

Hence, the functions present in this sub-section are grouped by its capability of transmit data remotely. For further information about the specific parameters of each transmission method, please refer to section 2.8.

Alerts

Lastly, there are a couple of important buttons in this section. The first button corresponds to the **Intrusion Alert** configuration shortcut. Once clicked, a configuration wizard will help out in every configuration step of the intrusion alert.
Here, the intrusion alert can be deactivated during the **wait time** of the intrusion alert cycle. For information about this, please follow section 2.9.

→ Section 2.9

See section 2.9 for more information about the relevant parameters and concepts of the intrusion alert.
Additionally, the second button is intended for alert transmission to **Intelity cloud server**.

3.5.3 Extraction Settings

→ Section 3.5.2

The extraction sub-section plays a role similar to the data transmission settings (see 3.5.2), which gathers data transmission methods settings. However, this sub-section only collects transmission methods that require a physical connection.

→ Sections 2.8.4 and 2.8.5

The available methods that meet this condition are: **Socket** and **Modbus Slave TCP/IP**. For further information about their functioning principle and configuration parameters, please refer to sections 2.8.4 and 2.8.5.

3.5.4 Network Settings

→ Sections 2.8.1 and 2.3

This sub-section gathers all configuration parameters related to basic **network** functions as well as the **data consumption monitoring** features. All parameters found here are further explained in sections 2.8.1 and 2.3.

3.5.5 Init Settings

→ Section 3.2.4

This is the last sub-section present in the settings path and corresponds to the **configuration file edition** (CFG file). As mentioned in section 3.2.4, the CFG file is a plain text configuration file and is **also accessible** through the shortcut mentioned in that same section.
This function is intended for **advanced users** that are quite familiar with the device development and it is not recommended to change this file. All other website functions are designed to modify this file automatically.

4 Installation

This section covers all physical installation instructions and considerations.

4.1 Previous Considerations

First, please be sure to have a **reliable** and **stable** power supply of at least 30 Watts, which its voltage output is within 11.5 and 16 VDC.

In case that the DatumX is not intended to be powered by the electric grid but a **solar panel**, it is important to consider its power requirements and the quality of the solar circuit elements.

→ Section 5.3.1

If you need assistance to calculate, acquire, connect, or anything related to the mentioned elements, please contact us (see section 5.3.1).

Please, follow the next installation suggestions in order guarantee an optimal operation of the DatumX:

1. Avoid to place the equipment in **crowded** locations and/or with easy access for the general public. This is solely for security and equipment integrity purposes, as **electrical risk** and other factors may present.

2. In case that a cellular modem provides Internet connection to the device, please be sure to consider a physical spot with the **highest** signal quality to optimize data uploading the cloud and other desired transmissions. If any other data transmission method is considered instead, as LoRaWAN, it is important to acknowledge all signal and relevant requirements for the specific transmission method.

3. The device's case specification **does not** guarantee full protection against ambient weather factors, as high humidity, heavy rain, or high temperatures, etc. In addition, other factors as **dust accumulation** and **insect nests** in the connection slots, might represent a detriment or failure of the operation in the mid/long term. Thus, placing the device with an additional cabinet in is recommended for ambient conditions operations.

   → Section 5.3.1

   A suitable cabinet for each specific requirement, contemplating other devices as peripherals and similar, can be requested alongside with the DatumX purchase. Please, see section 5.3.1.

4. Be sure to have all necessary **electric protections** according to the operation location. For instance, lightning and/or surge protection might be necessary as well as a ground rod.

5. Device's configuration is usually made **prior** to field installation. Please, be sure to have tested **all** relevant factors of the device's required operation before installment. Important elements, such as data recollection, sensors response, and proper data formatting, are common factors that might have to be tested out before hand.

   → Section 5.3

   DatumX devices are configured according to each requirement. However, if further assistance is needed, please refer to section 5.3.

6. DatumX's manufacturer, Xactus, offers **sensor integration** services for advanced data acquisition. All integrations made are available for every user to use if needed. See Appendix A

→ Section 5.3.1, Appendix A

If sensor integration service is required, please contact us. See section 5.3.1.

4.2 Physical Installation

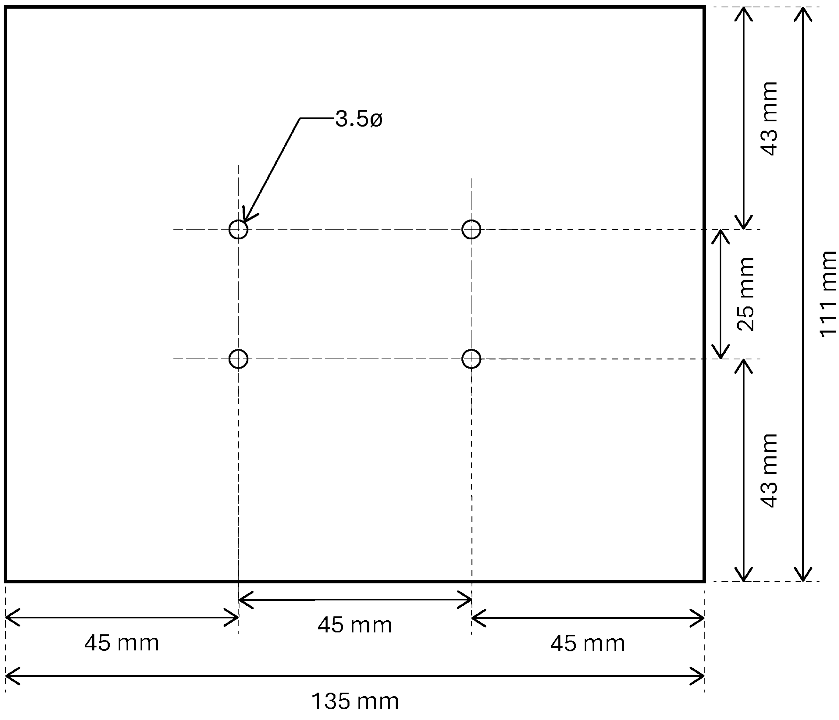
→ Figure 4.1

In order to attach the DatumX to other cabinet, surface, platform, or similar, the device has four screw holes located in the bottom face of it. All clearance holes are **medium fit** for **M3** screws. Figure 4.1 shows all clearance holes location in the device.

WARNING! →

The DatumX can be suited either horizontally (with the holes faced down) or vertically. However, it is important to consider that there cannot be **any weight** on top of it. Also, make sure that any connection cable is not applying torque or similar forces to the attachment screws.

Figure 4.1: DatumX Srew Holes



4.3 Electrical Connections

4.3.1 Power Supply

The DatumX datalogger supports power supply within the range from **11.5 VDC** to **16 VDC** with an available current of **2 Amperes**, allowing users to implement different sourcing options.

WARNING! →

A charge controller is not incorporated in the device. However, it is possible to supply energy by solar panel with an external charge controller.

→ Figure 1.3

The designed **connection port** for power sourcing is shown in figure 1.3. Note that the positive terminal must correspond to the **left connection hole**, and the negative terminal should go into the remaining connection hole.

WARNING! →

It is important to mention that powering the device should be the last connection step made, all other connections, such as sensors, peripherals, and others, must be performed prior to this.

4.3.2 Wiring Panel Connections

→ Section 1.3 Figure 1.3

As described in section 1.3 and figure 1.3, DatumX provides a wiring panel with removable ports, allowing to make all corresponding connections to each detached port. Once the connection process is finished, connection ports can be re-attached to the device. Each connection port has individual screws for every slot, it is important to properly tighten the screws in each used slot for a safe connection.

→ Section 2.7

Each slot has its own electrical specifications according to its function. Hence, section 2.7 covers specification of all communication interfaces present in the DatumX.

WARNING! → The sum of all output currents in the sourcing ports (as 12V or 5V) **should not** exceed **1.5 Amperes**.

WARNING! → Regardless if connections were made with the removable ports attached to the DatumX or not, it is recommended to make this process with the device unplugged from any source of power.

Be sure to connect each sensor or peripheral according to its manufacturer's indications. Even though DatumX wiring panel was designed for a variety of devices through common communication interfaces, connections may present some particularities of each sensor.

4.3.3 Relay Alarm Connection

→ Sections 1.3, 2.9, & 4.3.2

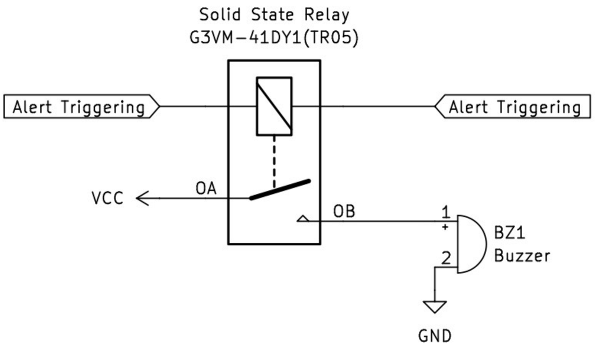
As described in sections 1.3, 2.9, and 4.3.2, DatumX offers a solid state relay output for users to have an audible alert, it could be used for either a variable alert, or an intrusion alert.

The solid state relay has the following electric limits: ±40 VDC, 28 VAC RMS, and 2 Amperes RMS.

→ Figure 4.2

In addition, relay output terminals A & B shall be connected as shown in figure 4.2. The figure suggests a basic alert system circuit by connecting the **OA** terminal to a continuous power supply and the **OB** terminal to the positive input of a common buzzer. Then, the negative terminal of the buzzer goes to ground closing the circuit. The alert triggering process is done internally according configuration mande.

Figure 4.2: Relay Alarm Connection



4.3.4 Internet Connection

→ Section 1.3

As seen in section 1.3, datalogger DatumX has a RJ45 connector for internet connection. Any modem or gateway that provides internet service to the DatumX **must** have the following IPv4 address: **192.168.10.1**.

If the application involves sensor reading through the RJ45 port, the use of a **LAN Switch** can allow both internet access and sensor readings without any problem.

By default, the device has **static** ip address for both wired and wireless networks. However, it is important to know that the wireless network function (WiFi) is just for user configuration and visualization rather than device's internet access. In any case, the IP Addresses are the following:

```
WiFi: 192.168.100.1 / 255.255.255.0
Wired: 192.168.10.3 / 255.255.255.0
```

→ Chapter 3 Section 3.5.4

Nonetheless, default configuration can be **overwritten** by user (see chapter 3, section 3.5.4) in any time, including wired, wireless, and gateway network addresses. It is possible to select DHCP or other desired static IP address according to application requirements.

4.3.5 Verification of Installment

Once all connections are finished and verified, DatumX, sensors, and other peripherals shall be energized and their functioning corroborated.

→ Section 1.5

One easy way of verifying DatumX correct functioning is to double-check its startup sequence, which will help in case of a common error. The startup sequence is explained in detail in section 1.5.

Keep in mind that some sensors connected to the DatumX may take longer to fully initialize, please wait until all of the system elements have started operations before any verification.

→ Section 5.3

Lastly, if problem persists, please contact our support team, see section 5.3.

5 Maintenance

The DatumX datalogger has been designed to operate long periods of time without human supervision. However, it is necessary to perform periodic **preventive maintenance** actions. This chapter, covers both preventive maintenance and some aspects of corrective maintenance.

WARNING! → It is important to note that the described actions **must** be performed by qualified personnel.

5.1 Preventive Maintenance

In a time span no longer than 45 days, it is recommended to perform the following actions with the device **turned off**:

1. Check if there is any wear damage, corrosion, stress cracks, etc.
2. Clean the device's case with a dry cloth. Please, do not let any dust or waste on it.
3. If there is any external element in the device's terminals (as insect nests, leaves, etc.) please, remove it.
4. Clean all peripheral devices (as sensors) according to factory recommendations.
5. Ensure that all external connection wires are in good condition.

→ Section 5.1.1

The internal battery life-span is approximately of **5 years**, however, it is recommended to replace it before this time. In order to change the battery, it is necessary to have access to the internal parts of the device. Even though it is recommended to request factory assistance, this process is described in Section 5.1.1.

WARNING! → Consider that device's manipulation may lead to a warranty loss.

WARNING! → Whenever a disconnection of the internal battery happens (even for a moment), the device's current date and time will loose configuration and will cause DatumX to **stop** all measurements. Date and time will **need** to be re-configured in order to resume measurements.

→ Section 3.5.1

For reference about date and time configuration, please see Section 3.5.1.

→ Table 5.1

Also, the internal battery specifications are shown in Table 5.1.

5.1.1 Internal Access

An **specialized technician** may have access to the device by removing all **4 screws** attaching the case's top part and **lifting** it. Please, be aware that both **fuse** and **removable** terminals must have been removed prior to this. Both of these components are easily removable by pulling them.

→ Figure 5.1

Figure 5.1, shows the process step by step of removing the top part of the device's case.

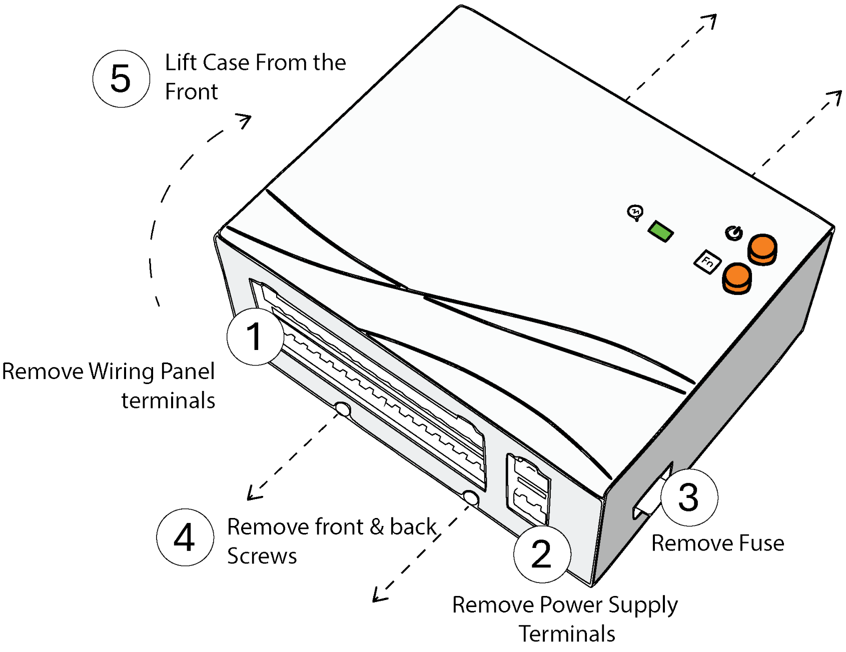
WARNING! → As described in figure 1.4, the internal structure of the device, allows for a quick change of RTC battery. Apart from the status and auxiliary buttons, there are no

Table 5.1: CR2032 Battery Specifications

Parameter	Description
Type Designation	IEC/JIS CR2032
Chemical System	Lithium / Manganese Dioxide (Li/MnO2)
Nominal Voltage	3.0 Volts
Weight	2.9 grams
Dimensions	Outer Diameter: 19.7 ∼20.0; Total Height: 2.9 ∼3.2
Nominal Capacity	220mAh
Heavy Metal Contents	Hg ≲5 ppm; Cd ≲20 ppm; Pb ≲40 ppm
Operating Temperature	-18°C ∼50°C
Recommended Storage	0 - 30°C

other user serviceable parts. Thus, it is recommended to have access to the internal parts of the device **ONLY** when a battery replacement and/or a hardware cleaning is needed.

Figure 5.1: Internal Access to the DatumX


5
Lift Case From the Front
1
Remove Wiring Panel terminals
4
Remove front & back Screws
2
Remove Power Supply Terminals
3
Remove Fuse
Fn

5.2 Corrective Maintenance

→ Section 5.3

If there is any problem with the device's functioning, please check your support terms and request technical support from our team, see Section 5.3.

WARNING! →

Consider that device's manipulation may lead to a warranty loss.

Also, beeps and blinks described in section 1.5 and table 1.4 may help to diagnose common errors in the device.

→ Table 5.2, Figure 5.2

Finally, if the device is not turning on, a fuse change may be needed. The main fuse replacement process it is quite simple, consisting only in pulling the damaged fuse and putting a spare one. Of course, the device **MUST** be shut off. The DatumX case has a hole exposing the main fuse (as shown in section 1.3). Please, **only** use **MINI Blade Fuses** with the characteristics described in table 5.2. A visual and size

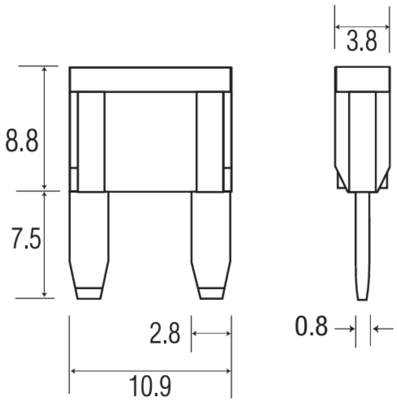
→ Section 5.3.1

reference of the fuse is presented in Figure 5.2.
Contact us in order to purchase fuse replacements. See section 5.3.1.

Table 5.2: Blade Fuse Specifications

Parameter	Description
Voltage Rating	32 VDC
Interrupting Rating	1000A ∼32 VDC
Recommended Temperature	-40°C to 125°C
Terminals Material	Silver plated / Tin plated zinc alloy
Housing Material	PA66 (UL 94 Flammability Rating - V2)
Net Weight	0.57 ±5%

Figure 5.2: Fuse dimensions



5.3 Support

Xactus provides a ticket-based support service for its clients. Thus, users must access and create an account in the following URL: **xactus-support.ngdesk.com**. Once an account is registered into the system, users can create tickets related with any support request, which will be answered in **3 business days** following the ticket creation.
It is important to mention that even though all ticket responses are automatically sent to the registered email, all user replies **have** to be entered through the platform, otherwise will be **unattended**.
It is preferred that each ticket has the following information:

1. Project attached to the device.
2. Device's or devices serial number.
3. Description of all the connections present in the device, as sensors and other peripherals.
4. If any changes in the device, its environment, or programming have been performed prior to the problem.
5. Detailed description of the problem.

6. Relevant pictures, graphs, video, or other material that may help to correctly diagnose the problem.
7. If the problem occurs frequently or not.
8. If a previous verification process or troubleshooting has been made.

WARNING! → Keep in mind that support coverage is subject to the agreed terms during device's purchase.

5.3.1 Contact

In cases that users need to contact Xactus for other services different than support, as for instance, replacement parts, additional modules, new sensor integration, and related, they can refer to the following options:

1. Phone number: +57 3107618987
2. Email: info@xactus.io
3. Website: www.xactus.io

An specialized department will help according to the request.

5.4 Disposal

At the end of this product's life it should not be put in commercial or domestic refuse but sent for recycling. Any batteries contained within the product or used during the products life should be removed from the product and also be sent to an appropriate recycling facility, per The Waste Electrical and Electronic Equipment (WEEE) Regulations 2012/19/EU.

Appendix A Software Integrated Sensors

→ Table A.1

In table A.1, there are all software integrated sensors by our development team. This list will be updated over time.

Table A.1: Software Integrated Sensors

Brand	Sensor	Interface
ABB	Optima AO2000	Ethernet
Acoem	Orion	Ethernet
Acoem	01dB	Ethernet
Advantech	ADAM-4017	RS485
Alphasense	OPC-N3[1]	RS485
Alphasense	A4, A43F, A431 Sensor Modules[1]	RS485
Ambilabs	Airpointer	Ethernet
Aqualabo	Salinity & Conductivity C4E, CTZ	RS485
Aqualabo	Turbidity NTU	RS485
Aqualabo	Dissolved Oxygen OPTOD	RS485
Aqualabo	pH PHEHT	RS485
Aqualabo	Total Suspended Solids MES5	RS485
Autonics	Total Suspended Solids MES5	RS485
Comet	Web Sensor T05, T25, T35, T45, T75 Series	Ethernet
Davis Instruments	Vantage Pro & Vue Series	Weather
Data from Sky	Traffic Camera	Ethernet
Delta Ohm	HD52 Series	RS485
Dilus	AirQrate	Ethernet
Ecowitt	FG-GW1000	Ethernet
EC-Sense	NO2, CO, SO2, O3 Gas sensor modules[1]	RS485
Endress + Hausser	Proline Promag W400	RS485
Envea	PM Cairsens Series	RS485
Envea	IQ-Link PM Cairsens Series	RS485
Envea	IQ-Link Cairsens Series	RS485
Envea	Cairsens Series	RS485
Envea	E-Series Gas Analyzer	Ethernet
EnviDAQ	E-8017B	RS485
Eureka	Manta Series	RS232
Focused Photonics	DO200, PH200, TUR200, EC200 Transmitters	RS485
Focused Photonics	BPM200 Transmitters	USB
Geolux	LX-80	RS485
Geolux	RSS-2-300WL	RS485
Geonica	44 Series weather station	RS485 / Ethernet
LSI Lastem	DQA251	RS232
LSI Lastem	DMA975	RS485
LSI Lastem	DNB105.2	RS485
LSI Lastem	DNB202	RS485

[1] Requires additional hardware.

Table A.1: Software Integrated Sensors (Continuation)

Brand	Sensor	Interface
Lufft	WS-UMB Series	RS485
Palas	Fidas 200	Ethernet
PCE Instruments	428	USB
Plantower	PMSA003[1]	RS485
Pulsar Measurement	Microwave	RS485
Renke	RS-RA-AL	RS485
Renke	RS-ZS-FL	RS485
Rika	RK520-02	RS485
Rika	RK300-06 & 06B Series	RS485
Rika	RK120-07	RS485
Rika	RK900-11	RS485
Rika	RK400-09	RS485
Sensirion	SHT30	RS485
Sensirion	SPS30[1]	RS485
SenTec	SEM1000	RS485
Senva	TG-UL Series	RS485
Senva	Total Sense Series	RS485
Siap + Micros	t027 TP200	RS485
Siemens	MAG 6000	RS485
Siemens	MAG 8000	RS485
SonTek	SL Series	RS232
Stalker	DSR Series	RS232
Tsingsense	TH1	RS485
Turnkey Instruments	Topas, Osiris, & DustMate	USB
Vaisala	WXT500 Series	RS485
Yosemite Technologies	Y610-B	RS485
YSI	WaterLOG H-3401	SDI12

[1] Requires additional hardware.

Appendix B Version Control

Table B.1: Version Control

Release Date	Description	Version
26/07/2022	Initial Version	V 1.0.0
27/01/2025	Overal updates in features and format change. First English translation	V 2.0.0