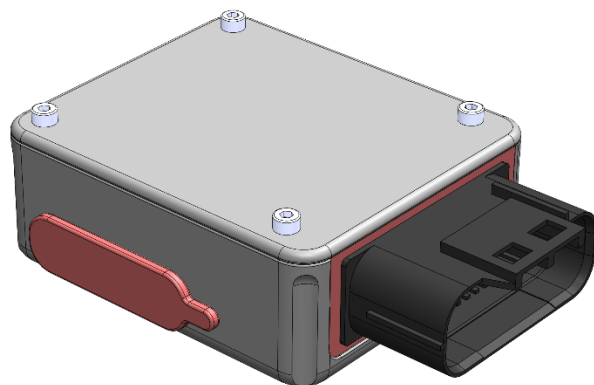




GUBELOG-01

GUBELOG-02

GUBELOG-03



Introduction

GUBELOG-01 (02 and 03) are multifunctional loggers, designed for use in various fields (such as laboratory analysis and monitoring, machinery monitoring, precision agriculture, industrial monitoring, process analysis, automotive, motorsport, etc.) thanks to their data logging capabilities but also thanks to the various additional functions available. The data logger is designed to acquire up to 38 inputs simultaneously, with frequencies ranging from 0.01Hz to 200Hz, making it an excellent data recording system. The hardware also features two digital outputs capable of generating programmable 0-5 volt logic levels or driving two relays for managing any external automations. The data logger also has the ability to connect to an external WiFi module (accessory) for remote data transmission, management of up to 4 alarms, and communication with a dashboard on the Cloud platform.

Integrated Sensors (GPS/GNSS and IMU)

The device is equipped with two integrated sensors that are essential for the dynamic analysis of the vehicle or machinery:

- **GPS/GNSS receiver:** Built-in module with a refresh rate of 1 Hz (for GUBELOG-01 and GUBELOG-02) and 10 Hz (for GUBELOG-03). The antenna is either integrated into the device or external (as in the case of GUBELOG-03).
- **Inertial Measurement Unit (IMU):** 6-axis sensor (3-axis accelerometers + 3-axis gyroscopes) rigidly mounted on the internal PCB. It measures longitudinal/lateral accelerations and roll, pitch, and yaw angles.
Accelerometers +/- 30g, internal sampling rate 1125Hz, low-pass filter 68.8Hz
Gyroscopes +/- 2000°/sec internal sampling rate 1125Hz, low-pass filter 73.3Hz

Analog Inputs (AN1 - AN8)

The datalogger features 8 analog inputs with a 12-bit ADC converter. The inputs are divided into two categories to ensure maximum industrial and motorsport compatibility:

- **Standard 0-5V Channels (AN1 - AN4):** High-impedance voltage inputs, ideal for reading potentiometers, pressure sensors, and pre-amplified thermocouples. Maximum tolerated voltage: -0.3V to 5.3V.
- **0-20mA / 4-20mA Current Channels (AN5 - AN8):** These 4 inputs are equipped internally with a precision 250 Ohm pull-down resistor to ground. They allow "Plug & Play" connection of industrial current sensors without additional hardware.

Digital Inputs / Frequency (DI1 - DI4)

The 4 digital inputs are designed for reading pulse signals (wheel speed sensors, engine RPM, flow meters, tone wheels).

- **Frequency and Precision:** The Hertz calculation is entrusted to a very high frequency hardware counter that guarantees almost zero latency for signals from 0.1Hz to 20KHz.
- **Logic Thresholds (Trigger):** The signal is considered "HIGH" above 2.5V and "LOW" below 1.0V. The maximum voltage tolerated on the pin is the battery voltage (VBatt).

CAN Bus Interface

The system integrates a CAN Bus capable of extracting up to 16 channels simultaneously.

- **Transceiver:** High-Speed CAN 2.0B.
- **Protocols:** Using the GUBELLINI DataStudio software it is possible to configure the CAN bus according to the OBD II, SAE J1939, ISOBUS (ISO 11783), OpenCAN (EN-50325-4) protocols.
- **Termination:** The 120 Ohm termination resistor is present internally in the datalogger.

Digital Outputs

The system integrates two digital outputs connected to two programmable software alarms (based on the input values) that allow the generation of a 0-5V signal. The same two outputs can drive two relays to activate external actuators (such as motors, solenoid valves, pumps, or other).

- **Voltage signal:** 0-5V.
- **Relay:** Relay drive via external power supply.
- **Maximum load:** Max. drain voltage 60V, Max. current 1.5A

CAN Bridge

This mode allows you to create a bridge between "physical" inputs (GPS, IMU, digital inputs, and analog inputs) and the CAN bus. You can transmit the values detected on the physical inputs directly to the CAN bus. In this mode, the CAN input channels will not be considered. In this mode, the input data will not be recorded to the SD card.

WiFi module

An additional connector allows the data logger to interface with the GP-DL-WF01 WiFi module (accessory). The WiFi module acts as a bridge to the local WiFi network and allows the signals detected on the data logger's inputs to be sent (in real time). The statuses of four programmable alarms are also transmitted.

- **Network Processor:** WIZnet WizFi360 (Industrial Grade).
- **Wireless Standard:** Full compatibility with IEEE 802.11 b/g/n networks.
- **Operating Frequency:** 2.4 GHz (Channels 1-13), which ensures excellent obstacle penetration and a long range.
- **Data Interface:** UART (Baud Rate 115200 bps).
- **Network Protocols:** Native TCP/IPv4 stack (used for real-time data streaming on TCP port 9000). The TCP port is programmable.
- **Security and Encryption:** Support for WPA / WPA2-PSK protected networks.
- **Operating Mode:** Station (STA). The module connects transparently and automatically to the router or corporate Wi-Fi network.
- **Power supply and consumption:** The module is powered directly by the Datalogger with a peak transmission (TX) consumption of approximately 200 mA.

Cloud service

Using the Wi-Fi module, if the local network to which the data logger is connected has internet access, you can activate our cloud service to remotely monitor sensors and alarms. The cloud service provides a monitoring dashboard and a messaging service for alarms.

Isolation and work environment

The datalogger guarantees an IPX7 insulation level, allowing use in harsh environments. The guaranteed operating temperature range is from -20°C to +85°C.

Internal memory

Logged data is saved to an unformatted micro SD card (RAW writing). Supported micro SD card formats are 16/32/64/128GB (larger sizes are supported but cannot be used to their full capacity). The micro SD card can be removed and connected directly to a PC to retrieve data (organized into datasets) using the dedicated GUBELLINI DataStudio function.

Auto-Start Conditions

The data logger can manage the start of recording in two modes: "Always On" and "Auto-start Condition." In "Always On" mode, the data logger generates a new dataset number and begins recording data from active inputs immediately. The dataset closes when the device is turned off. In "Auto-Start Condition" mode, the data logger generates a new dataset number and begins recording data only after a condition (related to the value of one of the active inputs) is met. The "Auto-Start" condition is software-programmable. The dataset closes and recording stops when the device is turned off.

Artificial intelligence

The GUBELLINI DataStudio data logger management software is equipped with an AI module called **AI Analyst**. **AI Analyst** is an innovative way to analyze data recorded by the data logger. **AI Analyst** isn't just a simple AI "chatbot" you can ask generic questions to, but a true **multi-agent artificial intelligence system** integrated into the software. It's designed to assist you in analyzing the acquired data, automating the search for anomalies, the creation of complex graphs, and the preparation of professional reports.

Diet

The device can be powered by a DC voltage from 7 to 24V. The main connector provides a 5V supply voltage for the connected sensors.

USB-C connection

You can connect to the data logger via a USB-C connector located on the side of the enclosure and protected by a rubber gasket. The USB connection allows you to send and receive configuration data, monitor the value of active inputs in real time, and download recorded data organized into datasets. Transmission limit: 1 Mbps.

Main Connector: JAE MX23A26NF1 (Automotive Grade)

To ensure maximum reliability in signal transmission and maintain the IPX7 waterproof certification, the physical interfacing of all channels (analog, digital, CAN Bus and power) is entrusted to a single automotive grade connector: the **26-pin JAE Electronics MX23A26NF1**.

The MX23A series was specifically designed for the automotive and motorcycle industries for use in exposed areas (engine compartments, external frames, agricultural machinery). It features the highest level of technical features:

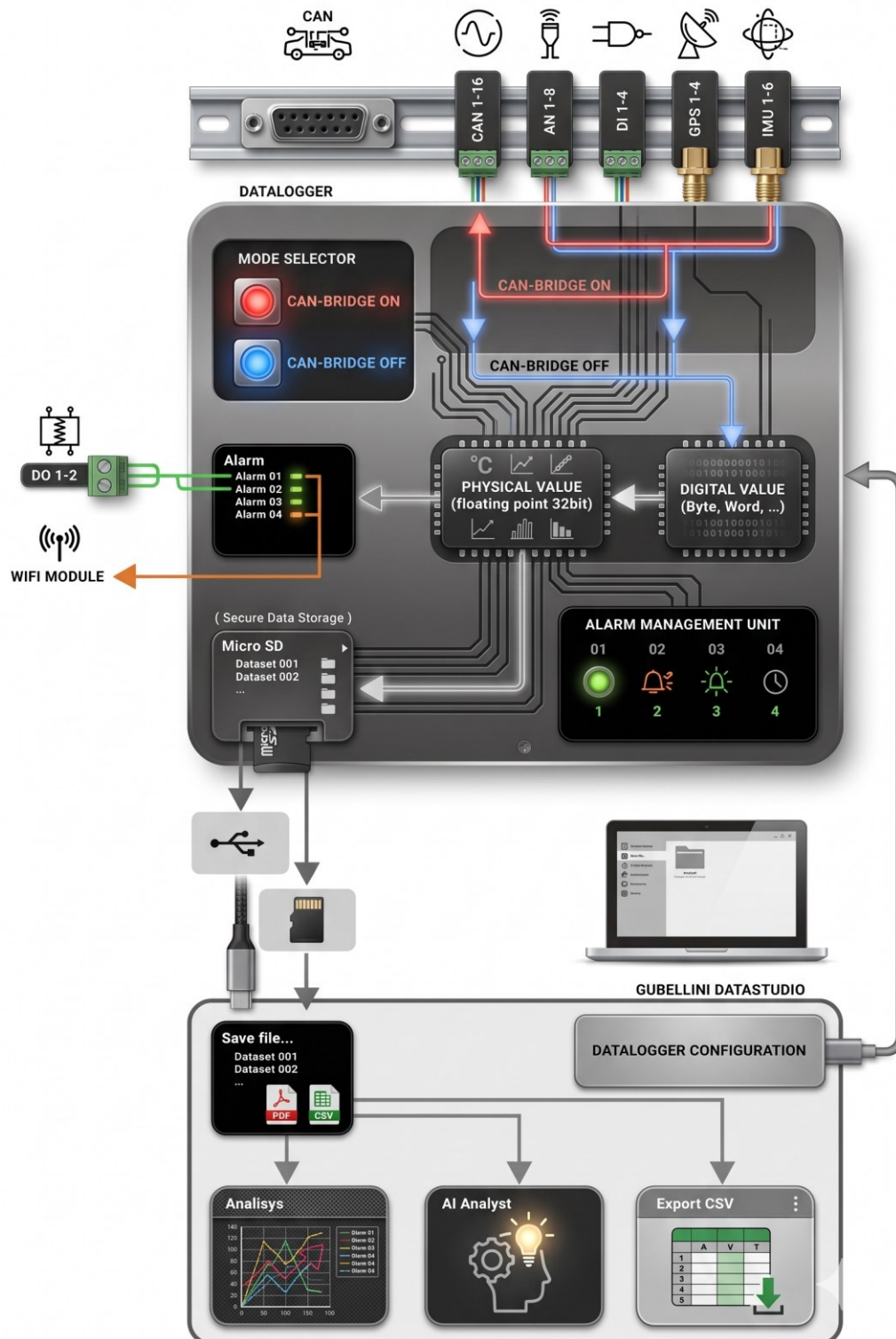
- **Waterproof:** The connector is equipped with silicone gaskets (sealing rings) on the coupling and individual rubber grommets for each inserted cable, preventing the entry of water, oils and dust.

- **Vibration Resistance:** The mechanical click-lock system ensures that the connector will not accidentally come loose, even when subjected to the extreme stresses of a single-cylinder engine or the curbs of a racetrack.
- **Electrical Protection:** Low-profile housing physically separates pins to prevent short circuits due to moisture.

PART 1: Software

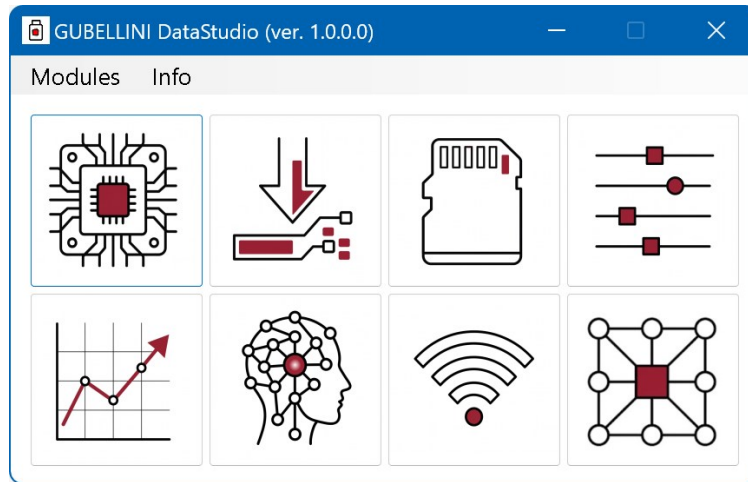
Chapter 1: Introduction and Main Interface

The GUBELOG-01 data logger is managed entirely through the GUBELLINI DataStudio software. Using GUBELLINI DataStudio, you can configure the logger (such as inputs, CAN bus, sampling rates, and more), download recorded data (saved in sessions called "datasets"), extract data directly from the micro SD card, open datasets to view graphs and statistics, and manage remote functions (such as external relay management, Wi-Fi module management, and cloud connectivity). Furthermore, the software features an "agentic" AI system that allows you to leverage the power of artificial intelligence to perform analyses and compile reports.



When you first launch the program, the software automatically generates some basic configuration files (channels_setting.xml, AI_contest.xml, logger_channel_setting.xml, logger_configuration_setting.xml), setting the default parameters for all 38 inputs supported by the system (grouped into categories such as GPS, IMU, DIGITAL, ANALOG and CAN). These files are saved in the software's execution folder.

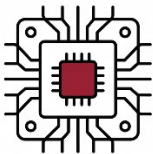
1.1 The Home Screen (Hub)



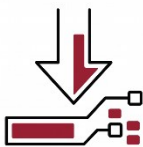
When you launch the software, you'll be presented with the main screen, which acts as a true "command center" (Hub). From here, you can quickly access all the program's modules via 8 large buttons or the top drop-down menu.

Here's an overview of the tools at your disposal:

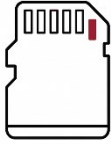
First Row of Buttons: Management and Datalogger



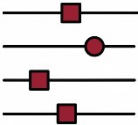
- **Logger:** Clicking this button takes you to the main Datalogger management panel. From here, you can configure input parameters (activation, sampling rates, conversion formulas, etc.), configure the CAN bus, define when and how to start recording, monitor connection status, and view data in real time.



- **Download :** Opens the interface dedicated to downloading recorded data from the device to the computer.



- **Micro SD:** Opens the section dedicated to direct management of the Micro SD memory card, useful for scanning, reading and downloading datasets saved in RAW format on the device.



- **Channels:** Opens the channel configuration form for analysis. Here you can customize the names, units of measurement, graph display colors, maximum/minimum values, and line thicknesses.

Second Row of Buttons: Analysis and Advanced Functions



- **Analysis :** Opens the main data analysis suite (Time Charts) to visualize the progress of your recordings (datasets) and examine the data through functions such as Scatter charts, histograms, FFTs and more.



- **AI Analyst:** Launch the built-in virtual AI assistant powered by Artificial Intelligence. This advanced tool will help you interpret the data it collects, providing insights and suggestions. Note: It requires a personal Gemini or OpenAI API key to function.



- **WiFi Monitor:** Opens the panel for managing and monitoring the Wi-Fi connection of the external module connected to your datalogger.



- **CAN Bridge:** Access the CAN bus network configuration interface in "CAN-BRIDGE" mode. This mode allows you to create a bridge between "physical" inputs (GPS, IMU digital inputs, and analog inputs) and the CAN bus. You can transmit the values detected on the physical inputs to the CAN bus. NOTE: In this mode, the CAN inputs (set via the DATALOGGER interface) will not be considered.

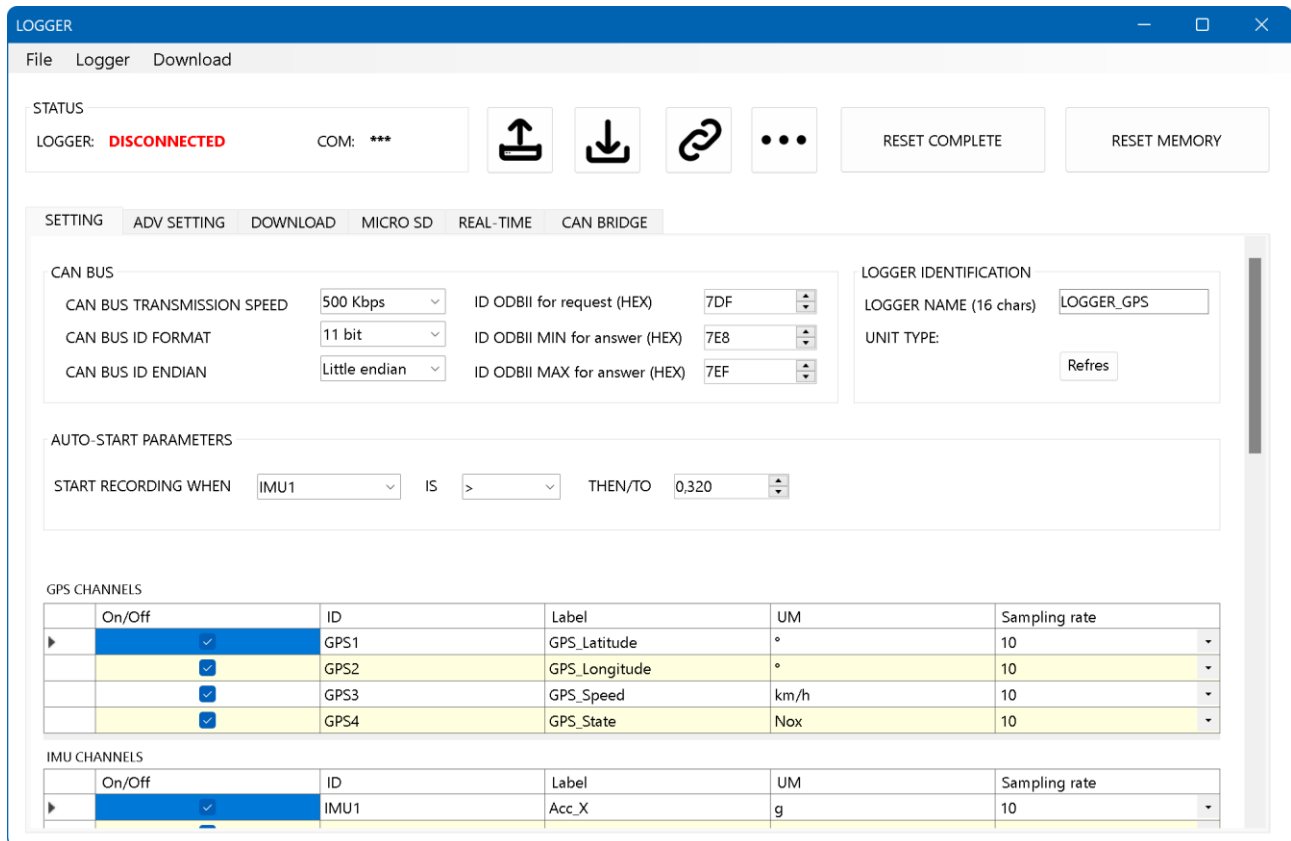
1.2 The Top Menu Bar

In addition to the buttons, there is a classic menu bar at the top of the window:

- **"Modules" menu** : Contains a drop-down list that exactly replicates the functions of the main buttons (Logger, Download, Micro SD, Channels, Analysis, AI Analyst, WiFi Monitor, CAN Bridge), giving you an alternative way to navigate between windows.
- **"Info" menu -> "Software version"** : Clicking this item will display a summary window containing program information. You can view the software name, current version, author, copyright information, and the version of the operating system currently running on your PC.

Chapter 2: Logger (Datalogger Configuration)

This section, accessible by clicking on **Logger** from the main screen, is the heart of the software. From here, you can configure all the data logger parameters, download the data stored in the data logger (grouped into datasets), download data directly from the micro-SD, monitor sensors in real time, and configure external modules. The Logger screen is divided into two areas: **STATUS** and **TABS** (which includes **SETTING**, **ADV SETTING**, **DOWNLOAD**, **MICRO SD**, **REAL-TIME**, and **CAN BRIDGE**).



2.1 STATUS - Connection and Status

At the top of the window, you'll find the **STATUS** panel. To connect your PC to the device, use the top menu: go to **Logger -> Connect to logger** (shortcut **Ctrl+C**). If the USB connection is successful, the status will change from **DISCONNECTED** (red) to **CONNECTED** (green) and the COM port used will be shown.

2.2 CAN BUS Settings

CAN BUS box you can define the general communication parameters of the vehicle line:

- **CAN BUS TRANSMISSION SPEED:** Choose the network speed (e.g. 1 Mbps, 500 Kbps or 250 Kbps).
- **CAN BUS ID FORMAT:** Select standard (11-bit) or extended (29-bit) identifier. In the CAN channel-specific settings below, you can also set the protocol type (e.g., *CUSTOM*, *OBD II - RPM*, *OBD II - Speed*, etc.), ID, message length, and byte order (Little/Big Endian).
- **CAN BUS ENDIANESS:** Select whether to use “little endian” or “big endian” endianness.
- **OBD-II Parameters:** The default parameters are those normally used in the automotive industry. Do not change these values unless specifically needed.

2.3 LEGGER IDENTIFICATION

You can assign a name to each logger so that when you download the dataset, the file will automatically be named with that specific name. The data logger name will also remain saved in the data file.

2.4 Auto-Start Parameters

To avoid starting recording simply by turning it on, you can instruct the data logger to begin recording data only after a specific physical event occurs. In the dedicated box, you can set a logic such as:

- *START RECORDING WHEN [Channel] IS [Condition: >, <, =] THEN/TO [Numeric value] (For example: Start when GPS speed or engine RPM exceeds a certain threshold.)*

2.5 Channels - Channel Configuration

The Logger's main screen features several tables divided by channel type, allowing you to configure the behavior and acquisition logic of each channel in detail.

GPS Channels and IMU (Integrated Sensors)

The GPS module and the inertial measurement unit (IMU) are physically integrated into the Datalogger system. To ensure the correct functioning of the dynamic analysis, these channels are already mapped and **preconfigured at the factory** (you will find fixed and unchangeable ID, Name, and Unit of Measurement). For the GPS and IMU channels, you can only configure:

- **On/Off:** A checkbox to enable or disable recording for the individual channel (e.g. disable GPS_State or Gyro_X).
- **Sampling rate:** A drop-down menu that allows you to set the specific sampling rate for that channel's acquisition.

DIGITAL and ANALOG channels

The digital and analog channels represent the device's physical hardware inputs, to which you can connect external sensors (e.g., potentiometers, pressure sensors, switches). For each of these channels, you can freely customize the following items:

- **On/Off:** Check the box to enable or disable acquisition for that specific input.
- **Label:** By clicking on the cell you can write a custom name for the sensor (e.g. "Front Suspension", "Engine Temp"), which will then be displayed in the graphs.
- **UM (Unit of Measurement):** Allows you to enter the text of the physical unit of measurement (e.g. "mm", "bar", "°C").
- **Sampling rate:** Allows you to establish the sensor sampling rate by choosing preset values from the drop-down menu (ranging from 0.001 Hz up to peaks of 400 Hz).
- **Multiplier:** The mathematical multiplication factor to apply to the raw value read from the input.

- **Offset:** The deviation (tare) to add or subtract from the calculated value. Together with the *Multiplier*, it allows you to convert the raw electrical value into the desired real physical quantity.

Technical Note: Calculating Values for Analog and Digital Channels

To convert the "raw" electrical signal coming from the sensors into a readable physical quantity (e.g. bar, degrees, km/h), the Datalogger uses a linear formula based on two fundamental parameters that you must enter in the table: the **Multiplier** and the **Offset**.

Here's how the system handles signals and how to calculate these two parameters depending on the type of input.

1. Analog Inputs (0-5V Voltage Sensors)

The analog channels read an input voltage between 0 and 5 volts. The datalogger uses a 12-bit analog-to-digital converter (ADC), which means that this voltage is divided and mapped onto a numerical scale (raw value) ranging from 0 to 4095.

- 0V = 0 Raw
- 5V = 4095 Raw

The final value displayed on the chart is calculated with this formula: **Value = (Raw * Multiplier) + Offset**

How to calculate multiplier and offset: Take your sensor's datasheet and identify two reference points. *Practical example: a linear pressure sensor where 0.5V = 0 bar and 4.5V = 10 bar.*

1. Convert voltages to raw values:

- $Raw_1 = 0.5 * (4095 / 5) = 409.5$
- $Raw_2 = 4.5 * (4095 / 5) = 3685.5$

2. Calculate the Multiplier (M): It is the ratio between the physical variation and the Raw variation.

- $M = (10 \text{ bar} - 0 \text{ bar}) / (3685.5 - 409.5) = 10 / 3276 = 0.00305$

3. Calculate the Offset (O): This is the value to add/subtract to square zero.

- $O = 0 \text{ bar} - (409.5 * 0.00305) = -1.25$

By entering Multiplier = 0.00305 and Offset = -1.25, the logger will directly show you the pressure in bar!

2. Analog Inputs (0-20mA or 4-20mA Current Sensors)

In the world of industrial and automotive data acquisition, many professional sensors output current (mA) rather than voltage (V). To natively manage this type of sensor, analog inputs 5 to 8 (inclusive) are equipped with an internal interface that transforms the 0-20mA signal (current signal) into a 0-5 Volt signal.

The current signal is a perfectly readable voltage from the 0-5V range of the system:

- If the sensor supplies **20mA** (0.02A), the voltage across the channel will be: $0.02A * 250 \text{ Ohm} = 5V$ (i.e. 4095 Raw).
- If the sensor is a classic 4-20mA, at rest it will deliver **4mA** : $0.004A * 250 \text{ Ohm} = 1V$ (i.e. 819 Raw).

Once you know the Raw value at the extremes of the current sensor (e.g. 819 and 4095), you just need to apply the same formulas explained above to obtain the Multiplier and Offset.

3. Digital Inputs (Frequency)

Unlike analog channels, digital inputs do not read a DC voltage, but detect pulses (e.g., a pitch wheel sensor, a tachometer, or a flow meter). The Datalogger uses a highly precise internal hardware counter operating at 7.5 MHz to calculate the exact frequency of the input signal in Hertz (Hz).

The formula applied in this case is: **Value = (Frequency [Hz] * Multiplier) + Offset**

How to calculate Multiplier and Offset: Imagine you've attached a speed sensor to a wheel with 4 teeth (or 4 magnets). For each complete revolution of the wheel, the sensor generates 4 pulses. If the wheel rotates once per second, the data logger will read 4 Hz. If the wheel's circumference is 2 meters, one revolution per second equals 2 m/s.

- We know that 4 Hz = 2 m/s.
- The Multiplier will simply be: $M = 2 / 4 = 0.5$.
- The Offset in these cases is usually 0 (if the wheel is stationary, at 0 Hz the speed is 0 m/s).

CAN channels

The CAN BUS section allows you to "sniff," extract, and translate digital messages from the vehicle's data network (e.g., the ECU) and transform them into actual datalogger inputs. The entries are extremely detailed to accommodate any data string:

- **On/Off, Label, UM, Sampling rate, Multiplier, Offset:** These adjust the activation, name, unit of measurement, sampling rate and mathematical conversion of the extracted data, exactly as described for the Analog and Digital channels.
- **Protocol:** Allows you to choose the data extraction logic from the CAN bus. By selecting *CUSTOM*, you will have to manually fill in the following fields (to intercept the OPENCan protocol); alternatively, you can choose pre-filled OBD-II channels (such as *OBD II - Speed*, *OBD II - RPM*, *OBD II - TPS*, etc.) or J1939 channels (used in agricultural machinery, heavy vehicles, and industry).
By selecting one of the OBDII options, the datalogger will send a specific request on the CAN bus to obtain the relevant information in response (usually from the vehicle control unit). By selecting a J1939 channel, the logger will "sniff" all messages on the CAN bus and filter the messages using the PNG included in the 29-bit ID.
- **CAN ID:** The unique identifier (in decimal format) of the CAN message from which you want to extract information.
- **Data offset:** Sets the starting point (byte number) within the CAN packet from which the system should start reading your data.

- **Data type:** Allows you to define the computer format of the read number, being able to choose from the drop-down menu between: *unsigned byte* , *unsigned word* , *signed word* , *unsigned long word* , *signed long word* or *float* .

2.6 Save or Send Configuration

Once you have set the datalogger parameters according to your needs, you can save the configuration for subsequent software launches and send the configuration to the datalogger:

- **File -> Save configuration:** Save an .xml file to your PC.
- **Logger -> Send configuration to logger:** Physically transfers the newly set parameters to the internal memory of the connected data logger.

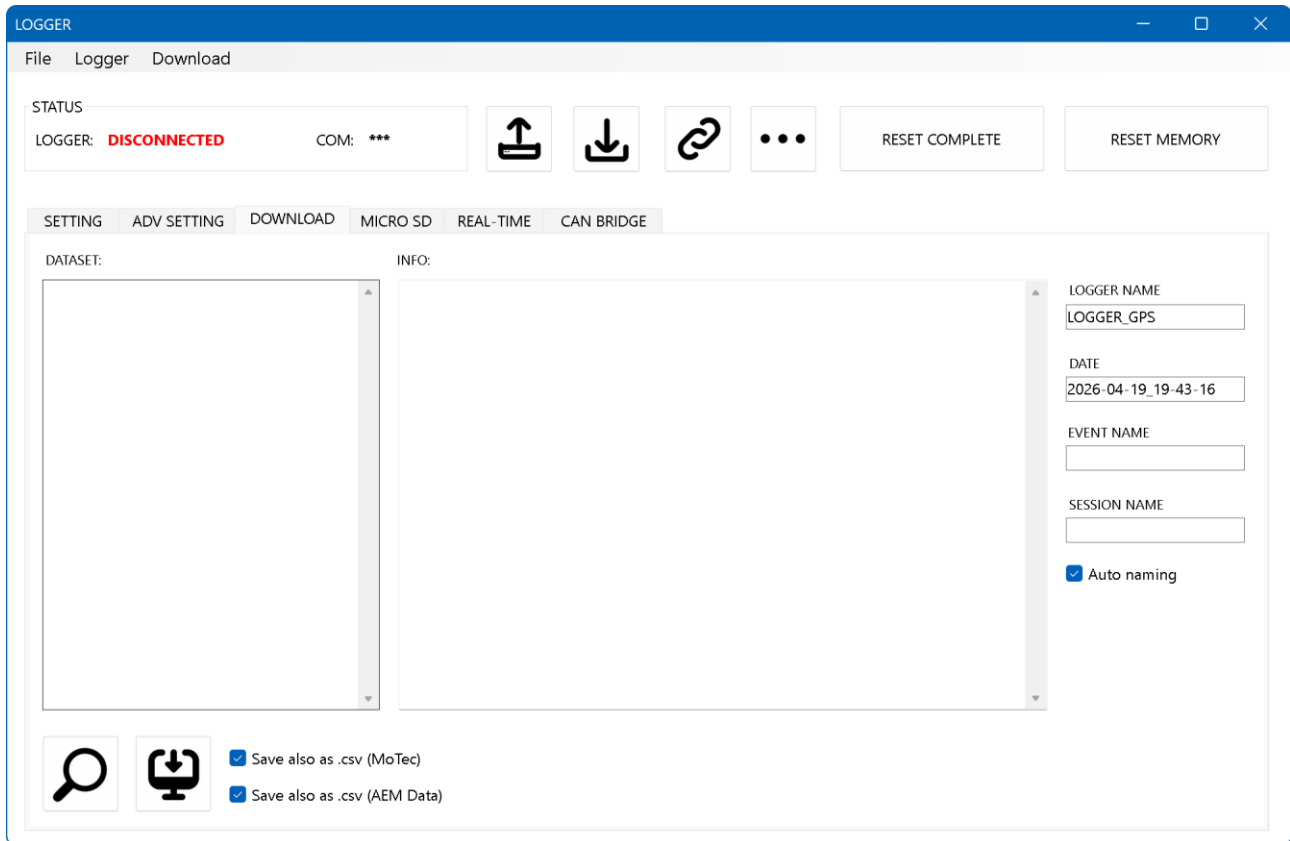
2.6 Datalogger operating logic

Each time the data logger is turned on, it reads the configuration metadata from the microSD memory (relating to the last configuration sent via USB) and decides whether to start storing the input values from the first available moment or wait for the event defined by the **AUTO-START parameters** . When recording starts, the data logger's LED will change its flashing pattern. At the start of recording, a new dataset is defined (by assigning a unique serial number). The dataset will be closed when the data logger is turned off.

If the AUTO-START condition is set to “Always-on”, a new dataset identified by a sequential number is generated each time the data logger is turned on. If the AUTO-START condition is set to detect a physical event (e.g., analog input 1 pressure greater than 4 bar), the data logger will wait to generate a new dataset. When the AUTO-START condition is met, a new dataset identified by a sequential number is generated and the values begin to be stored in the micro-SD memory. All datasets remain saved on the micro-SD card in the data logger, even after the data logger is turned off.

Chapter 3: Download - Downloading data via USB

By selecting the **Download folder (in the Logger window or by clicking the Download icon from the main window)**, you enter the interface for downloading the data stored in the device's internal memory using the USB cable.



3.1 Scanning and Viewing Datasets

- From the top menu select **Download -> Scan for dataset** (🔍).
- The software will query the datalogger and populate the list on the left (**DATASET:**) with the recorded sessions, indicating their size and space occupied. Select the Dataset you wish to download and click the **Download -> Download dataset menu** (📁).

3.2 Nomenclature and Export

Below, you can decide how to name the exported files:

- **Auto naming:** If enabled, the program will name the file by automatically combining Logger Name, Date, Event and Session.
- **Advanced Saving (CSV):** In addition to saving data in raw format (.dat file), you can check the **Save also as .csv (MoTec)** or **.csv (AEM Data) options** to directly export formats compatible with third-party analysis software.

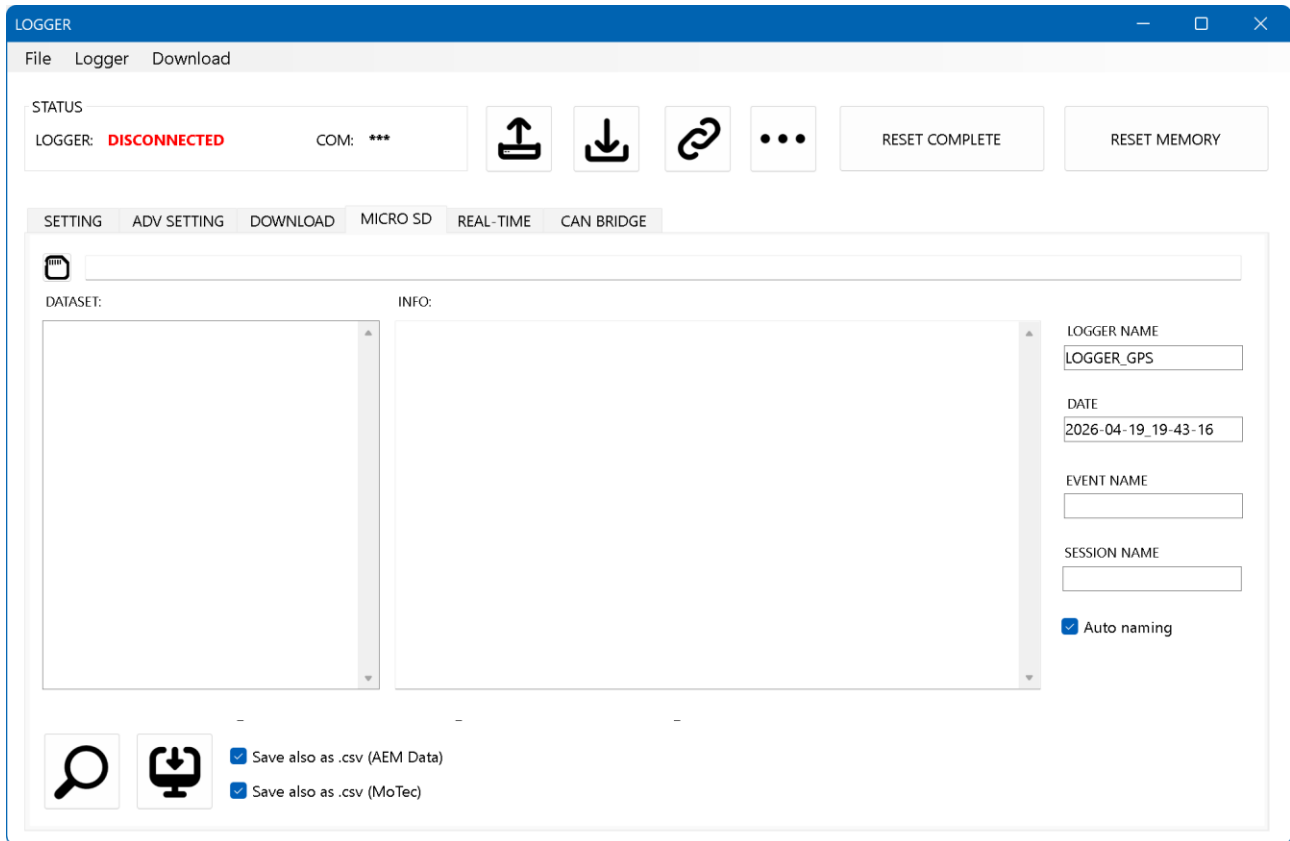
3.3 Start Download and Cleanup

- Press the download button to start the transfer (you will see the scroll bar indicating the percentage).

- *Warning:* Downloading via USB via the serial port (COM) may take time for very large files. In this case, it is recommended to download the datasets by removing the micro SD card from the data logger and connecting it directly to the PC.
- After using the data logger for a long time, you can decide to clear the device by going to the **Download -> Clear logger memory menu** . This will reset the serial number of the datasets to 1.

Chapter 4: Micro SD (Download data directly from PC)




For larger datasets, downloading via USB can be slow. The software features a raw (low-level) **Micro SD card reading function** that allows you to download gigabytes of data in seconds! Simply remove the micro SD card from the datalogger (unpowered) and connect it to your PC via a card reader.



4.1 The "Micro SD" Procedure

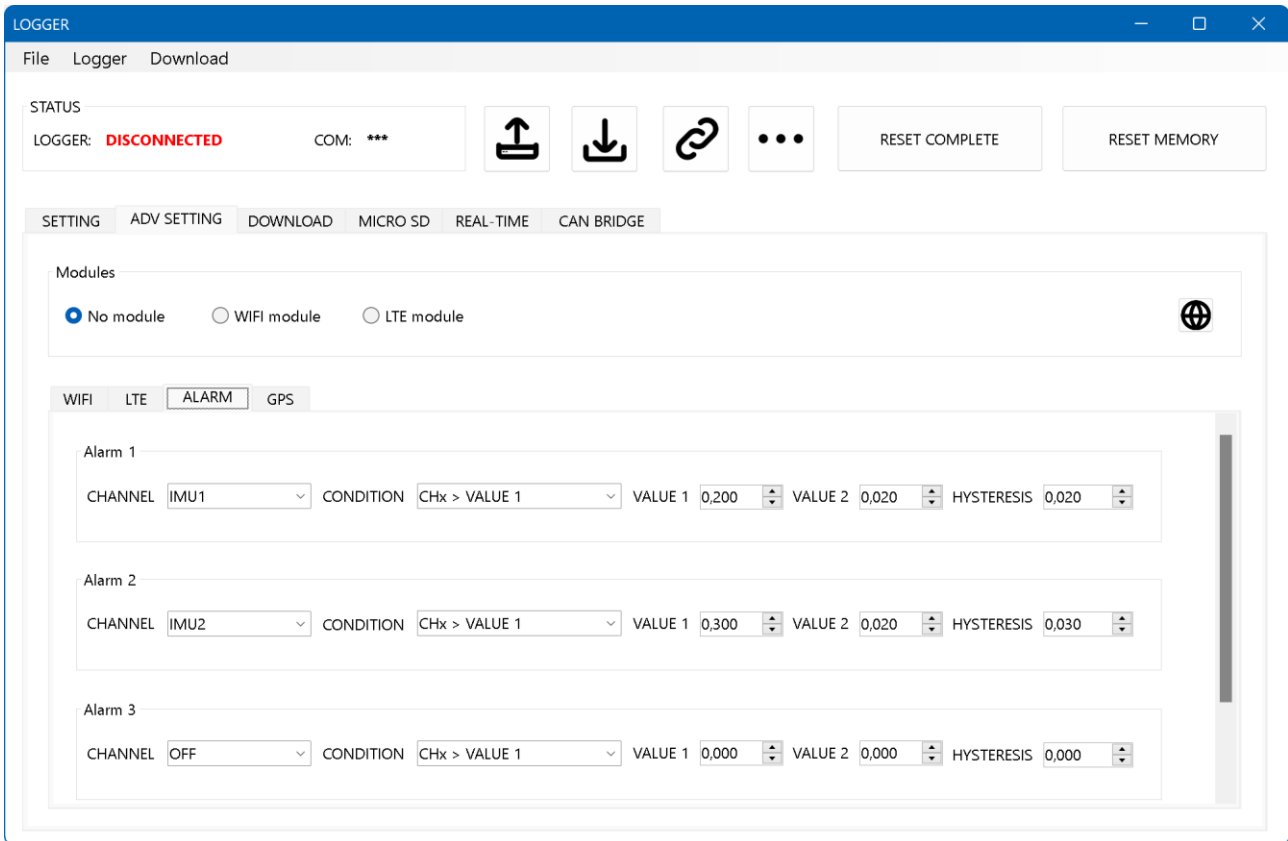
This feature (accessible by clicking the **Micro SD icon** in the main Hub) bypasses the data logger.

Important: You must disconnect the Micro SD from the data logger and insert it directly into your PC's card reader.

1. **Disk Selection:** Click the button next to the *SD PATH* line (). A window will open asking you to select the physical drive to which the MicroSD memory card is connected. The microSD card is usually the last line listed.
2. **Scan:** Using the dedicated button (), the software will read the Micro-SD index at block level, instantly populating the Dataset list.
3. **Download:** As with the traditional **Download folder** , fill in the "Event Name" fields or select the .csv extraction option. By clicking the extraction button (), the blocks will be reassembled on your hard drive at the maximum speed supported by the SD card.

Chapter 5: ADV SETTING - Advanced Configurations of External Modules

ADV SETTING tab of the **Logger** window , you'll find the **Modules pane** . From here, you can instruct the data logger to determine which additional hardware is currently connected to the system, choosing between: **LTE module** , **WiFi module** , or **No module** .



In addition to the external modules, here you can also find the folder dedicated to the module that manages the alarms (and the digital outputs used to drive relays or to generate 0-5V signals) and the internal GPS module.

5.1 ALARM Module (Physical Actuation)

GUBELOG-01 is able to manage and transmit up to **4 independent alarms** . The logical setting of these alarms is repeated for each of the 4 boxes. Follow the steps below to configure the alarm:

- 1) select the **CHANNEL** to be analyzed,
- 2) select the **CONDITION** (Greater than, Less than, Included, External) and the two numerical reference values **VALUE 1** and **VALUE 2**.
- 3) set the thresholds using the values **VALUE 1** and **VALUE 2**.
- 4) set the **HYSTERESIS** value to avoid the alarm state “oscillating” between on and off (when the channel value is close to the thresholds).

When the set condition is met, the alarm will switch from OFF to ON. The alarm will be sent via Wi-Fi or LTE.

The first two alarms (ALARM1 and ALARM2) are linked to the behavior of the digital outputs (DO1 and DO2). The digital outputs included in the data logger can drive two power relays. When you connect the digital outputs to relays, the relay will be energized when the associated condition is met. The relay will remain off when the condition is not met.

This configuration is essential if you want the Datalogger to physically operate switches (e.g. turn on a fan, a light, or cut power) in response to certain telemetry events.

NOTE : The digital outputs can also be used to generate a 0-5V signal. The signal will be fixed at 5V when the associated condition is **NOT satisfied**. The signal will be fixed at 0V when the associated condition is satisfied.

5.1.1 Configuration examples

Each digital output can be configured with a condition. Each condition can be associated with the value of one of the inputs. Both digital outputs can be associated with the same input.

Example : I can configure digital output 1 to activate a solenoid valve when the level in a tank (measured by analog input 4) exceeds a certain value. Similarly, I want to activate a pump to fill the same tank when the level (measured by analog input 4) drops below a certain value.

For each alarm, you have the following parameters available:

- **CHANNEL:** From the drop-down menu, you can choose the specific channel to monitor. You can select any sensor mapped in the system (GPS, IMU, DIGITAL, ANALOG, CAN) or select "OFF" to deactivate the alarm. Note: The channel must be active.
- **CONDITION:** Defines the logical triggering rule for the relay. The available options are:
 - CHx > VALUE 1: The relay trips if the channel value exceeds Value 1.
 - CHx < VALUE 1: The relay trips if the channel value drops below Value 1.
 - CHx > VALUE 1 and CHx < VALUE 2: The relay only trips if the value is within the range between Value 1 and Value 2.
 - CHx < VALUE 1 and CHx > VALUE 2: The relay trips if the value is outside the range delimited by the two thresholds (Value 1 and Value 2).
- **VALUE 1 / VALUE 2:** The numeric fields in which to physically enter the threshold values that determine the triggering of the logical conditions just described.
- **HYSTERESIS:** Used to define the hysteresis for the transition between the active and passive state of the alarms. It is recommended to use a numerical value between 1/50 and 1/100 of the delta between VALUE 1 and VALUE 2.

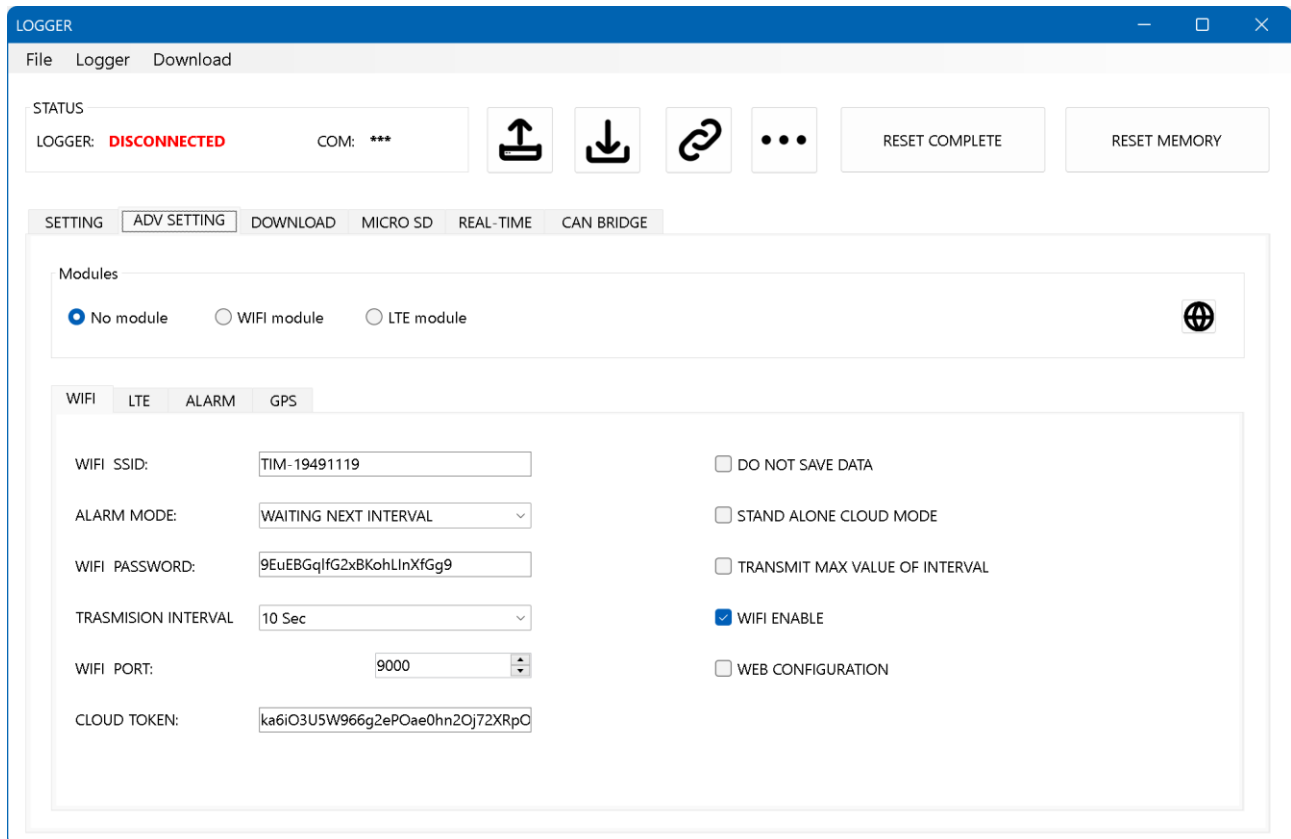
5.2 WIFI

GUBELOG-01 has a built-in Wi-Fi module. You can use the data logger's built-in Wi-Fi module to remotely send the values of all active inputs and alarms. By setting the parameters of the **Wi-Fi card** , GUBELOG-01 can operate in two distinct modes:

“STAND ALONE CLOUD” MODE: In this mode, the data logger starts up, connects automatically to the local Wi-Fi network and begins sending data to our cloud (GUBECLLOUD Service). Data transfer from the data logger to the cloud service occurs completely autonomously, without any external intervention. In this mode, the logger does not record data locally; data recording takes place in the cloud.

“PC AS GETWAY” MODE: In this mode, the data logger starts up, connects automatically to the local Wi-Fi network, and begins sending data to the TCP port (specified in the settings). The **GUBELLINI DataStudio suite** (specifically, the WIFI LINK window) will send the data to the cloud service (GUBECLLOUD Service). In this mode, the data logger will be able to record data locally on the micro SD card. The GUBELLINI DataStudio software must remain active on a PC connected to the local Wi-Fi network (and will act as a “bridge” between the data logger and the cloud service).

Both modes allow your data logger to transmit data to the **GUBECLLOUD Service** . Through the dashboard on the Cloud service, you can remotely monitor the progress of recorded values and alarms. You can also activate push notifications (via the mobile app) and email alerts when certain conditions occur.



The screenshot shows the 'LOGGER' web interface. At the top, there is a menu with 'File', 'Logger', and 'Download'. Below the menu, the 'STATUS' section shows 'LOGGER: DISCONNECTED' and 'COM: ***'. There are icons for upload, download, link, and a menu, along with 'RESET COMPLETE' and 'RESET MEMORY' buttons. The 'SETTING' section has tabs for 'ADV SETTING', 'DOWNLOAD', 'MICRO SD', 'REAL-TIME', and 'CAN BRIDGE'. Under 'Modules', there are radio buttons for 'No module' (selected), 'WIFI module', and 'LTE module'. The 'WIFI' tab is active, showing fields for 'WIFI SSID' (TIM-19491119), 'WIFI PASSWORD' (9EuEBGqIfG2xBKohLinXfGg9), 'TRASMISION INTERVAL' (10 Sec), 'WIFI PORT' (9000), and 'CLOUD TOKEN' (ka6iO3U5W966g2ePOae0hn2Oj72XRpC). On the right side of the settings, there are checkboxes for 'DO NOT SAVE DATA', 'STAND ALONE CLOUD MODE', 'TRANSMIT MAX VALUE OF INTERVAL', 'WIFI ENABLE' (checked), and 'WEB CONFIGURATION'.

5.2.1 WIFI Network Settings

In this section you configure the device's connection to your local network:

- **WIFI ENABLE:** Check this box to turn on and physically enable transmission through the module.
- **WIFI SSID and WIFI PASSWORD:** Enter the name (SSID) of the wireless network and the corresponding security password to which the Datalogger will connect.
- **WIFI PORT:** Defines the network communication port for data exchange (the default value set by the system is 9000, but you can change it if it is already occupied).
- **TRANSMISSION INTERVAL:** Determines the default frequency at which data packets are sent. You can set it to "Off", "10 Sec", "60 Sec", "10 Min", or "60 Min".
- **WEB CONFIGURATION:** By checking this box, you authorize access to the Wi-Fi module's configuration via the module's remote web interface. In this case, all parameters (such as **Wi-Fi SSID, Wi-Fi PASSWORD, Wi-Fi PORT**) will be defined by the remote web interface. **NOTE:** The integrated module does not support this function, so do not select this flag.
- **ALARM MODE:** Defines the transmission priority in case of emergency. You can choose to transmit the alarm immediately, interrupting the standard cycle (IMMEDIATELY), or wait and append it to the next scheduled transmission (WAITING NEXT INTERVAL).
- **CLOUD TOKEN:** Enter the token code provided when activating the GUBECLOUD Service. The code is unique for each data logger.

5.2.2 Additional WIFI configurations

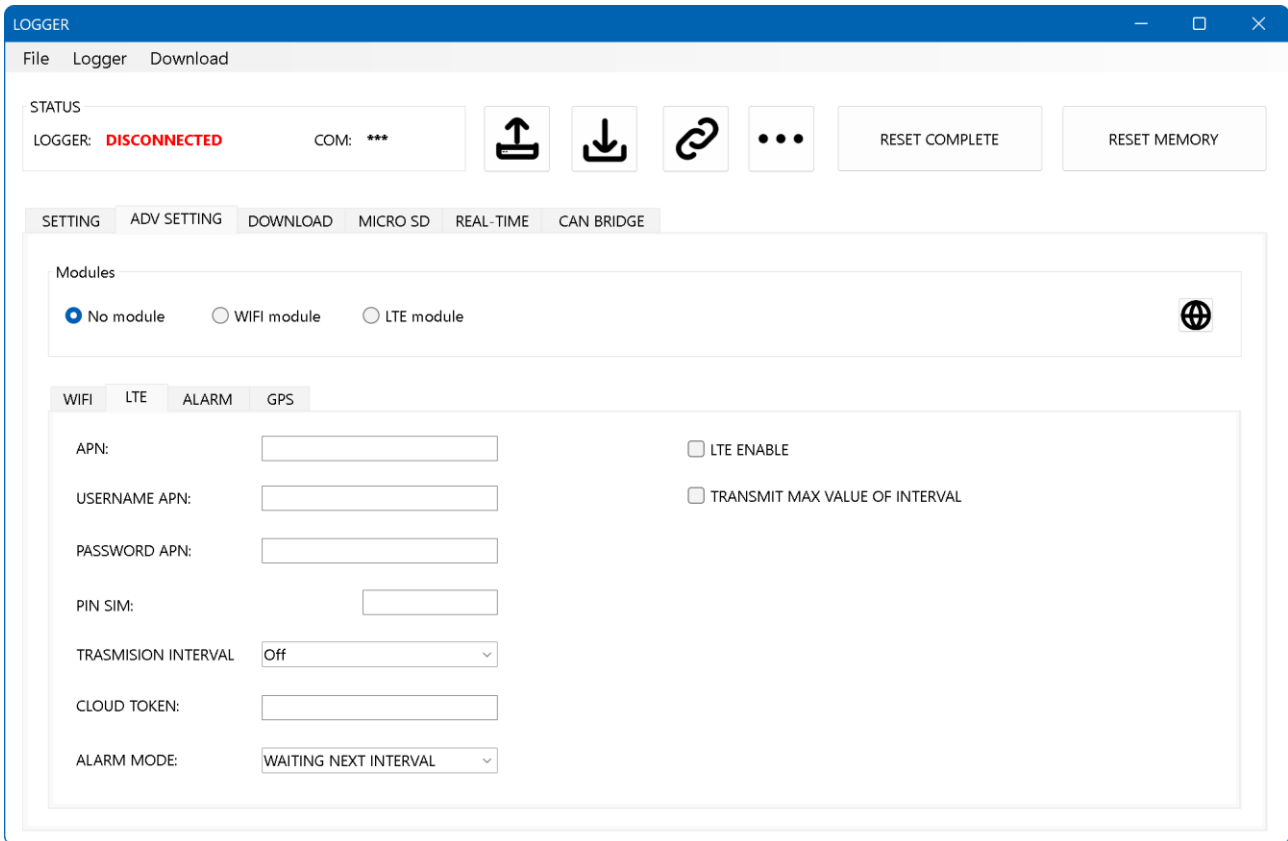
- **DO NOT SAVE DATA:** Decide whether or not to save data locally when the WIFI operates in "PC AS GETWAY" mode. Check (activate) the DO NOT SAVE DATA flag if you do not want to save data locally on the micro SD.
- **TRANSMIT MAX VALUE OF INTERVAL:** Data transmission to the cloud or local network is not high-frequency. Several seconds or minutes may pass between intervals (or between transmissions). Enable this option if you want to transmit the maximum value of each signal recorded between transmissions. Leave this option disabled if you only want to send the value detected at the time of transmission.

NOTE: When monitoring a critical parameter (such as engine temperature or oil pressure), it is important to track the critical value.

For example, if the oil pressure rises above the permitted threshold for only 20 seconds in a 1-minute interval, the data sent to the cloud may not reflect the actual situation: from "0 seconds" to "20 seconds" the pressure is normal, from "21 seconds" to "40 seconds" the pressure is too high, and from "41 seconds" to "60 seconds" the pressure returns to normal. The data is sent to the cloud at the 60th second, so the pressure will be normal. By enabling this function, the data logger will store the maximum value during the entire interval between one transmission and the next, and will send the maximum value at the sixtieth second. In this case, the data transmission will reflect the device's true "critical" situation.

5.3 LTE

GUBELOG-02 features a built-in LTE module. You can use the data logger's built-in LTE module to remotely send the values of all active inputs (GPS, IMU, Digital, Analog, CAN) and alarm values (Alarm1 .. 4).



How it works: Upon startup, the data logger automatically connects to the LTE network and begins sending data to our cloud (GUBECLOUD Service) at intervals set by the TRANSMISSION INTERVAL parameter. Data transfer from the data logger to the cloud service occurs completely automatically, without any external intervention. In this mode, the logger does not record data locally; data recording takes place in the cloud.

GUBECLOUD Service dashboard , you can remotely monitor the progress of recorded values and alarms. You can also activate push notifications (via the mobile app) and email alerts when certain conditions occur.

5.3.1 LTE connection settings

In this section you configure the device's connection to your local network:

- **LTE ENABLE:** Check this box to turn on and physically enable transmission through the module.
- **APN, APN USERNAME, and APN PASSWORD:** These values will be provided to you when you activate the cloud service and you will only have to enter them once.
- **SIM PIN:** The SIM is provided with cloud service activation. Leave this field blank.
- **TRANSMISSION INTERVAL:** Determines the default frequency at which data packets are sent. You can set it to "Off", "60 Sec", "10 Min", "60 Min", or "120 Min".

- **CLOUD TOKEN** : Enter the token code provided when activating the GUBECLOUD Service. The code is unique for each data logger.

5.2.2 Additional LTE configurations

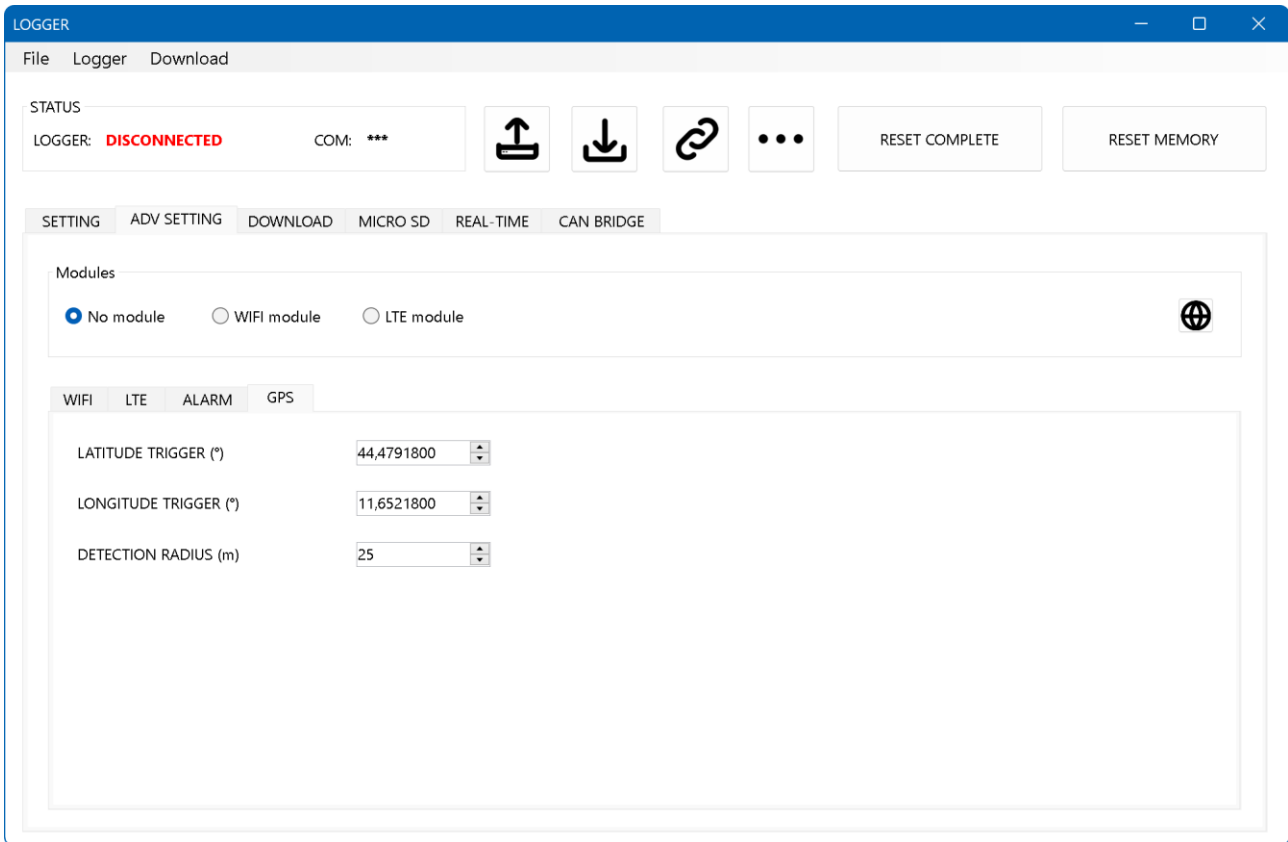
- **TRANSMIT MAX VALUE OF INTERVAL**: Data transmission to the cloud or local network is not high-frequency. Several seconds or minutes may pass between intervals (or between transmissions). Enable this option if you want to transmit the maximum value of each signal recorded between transmissions. Leave this option disabled if you only want to send the value detected at the time of transmission.

NOTE: When monitoring a critical parameter (such as engine temperature or oil pressure), it is important to track the critical value.

For example, if the oil pressure rises above the permitted threshold for only 20 seconds in a 1-minute interval, the data sent to the cloud may not reflect the actual situation: from "0 seconds" to "20 seconds" the pressure is normal, from "21 seconds" to "40 seconds" the pressure is too high, and from "41 seconds" to "60 seconds" the pressure returns to normal. The data is sent to the cloud at the 60th second, so the pressure will be normal. By enabling this function, the data logger will store the maximum value during the entire interval between one transmission and the next, and will send the maximum value at the sixtieth second. In this case, the data transmission will reflect the device's true "critical" situation.

5.3 GPS

GUBELOG-01 is equipped with a built-in GPS/GNSS module capable of detecting geographic coordinates and ground speed.



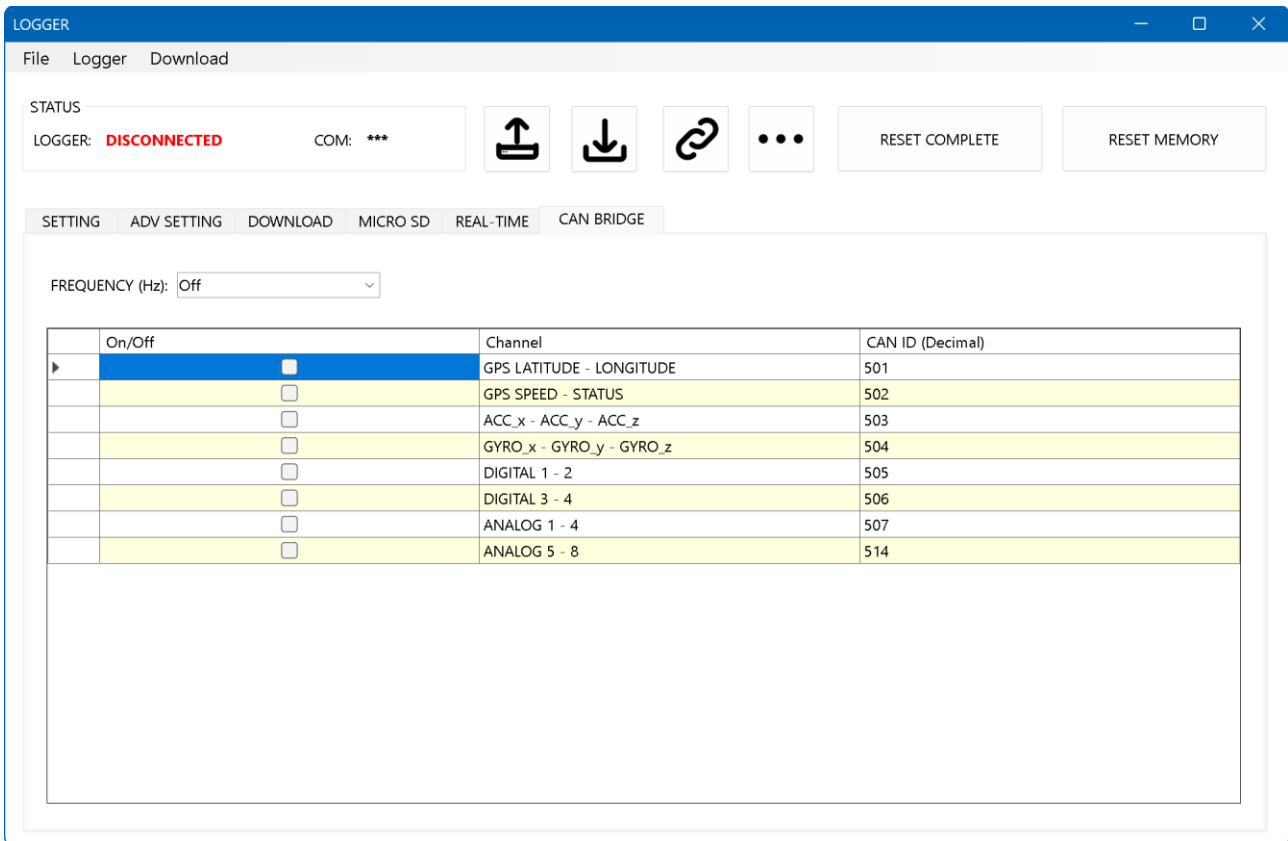
Using the LATITUDE TRIGGER, LONGITUDE TRIGGER, and DETECTION RADIUS parameters, you can set a “detection point.” The detection point is a circular area (with a radius equal to the DETECTION RADIUS) that, if crossed, triggers the “LAP” variable.

HOW IT WORKS : While recording data, the logger checks the GPS position and the detection point. If the data logger is mounted on a vehicle, and the vehicle passes the detection point, the “LAP” variable (which is zero at startup) is incremented by 1.

NOTE: for the datalogger to detect a subsequent crossing of the detection point, the vehicle/datalogger must have travelled at least 400m.

5.4 CAN BRIDGE

GUBELOG-01 can be used as a bridge between inputs and the CAN bus. This function allows you to read input values and retransmit them to the output on the CAN bus (with some limitations). To configure this function, from the main window you can click on **Datalogger** and then select the **CAN BRIDGE** folder .



5.4.1 Frequency

You'll find a drop-down menu where you can select the frequency at which to transmit input data via the CAN bus. Selecting OFF disables the function. Selecting one of the other options (1Hz, 10Hz, 50Hz, or 100Hz) enables the function.

5.4.2 CAN BRIDGE Settings

As can be seen from the configuration table, the inputs that can be "transferred" to the CAN bus are "packaged" into groups. The user can decide which packets to activate and which not by checking the On/Off column.

For each group, the user can specify the CAN ID (in decimal format) on which to transmit the packet. The format in which to transmit the packets is predefined and consists of 8 bytes.

NOTE: The endianness used by the CAN BRIDGE function is Little Endian (regardless of the one selected in the "SETTING" folder under "CAN ENDIANESS PROTOCOL". To specify whether the IDs follow the 11bit or 29bit (extended) protocol, select the correct value in the "SETTING" folder under "CAN ID PROTOCOL".

The inputs that can be transferred to the CAN bus are the following:

GPS : GPS Latitude GPS Longitude GPS Speed GPS Status	IMU : IMU Accelerometer X IMU Accelerometer Y IMU Accelerometer Z IMU Gyroscope X IMU Gyroscope Y IMU Gyroscope Z	DIGITAL : Digital 1 Digital 2 Digital 3 Digital 4	ANALOG : Analog 1 Analog 2 Analog 3 Analog 4 Analog 5 Analog 6 Analog 7 Analog 8
--	--	--	---

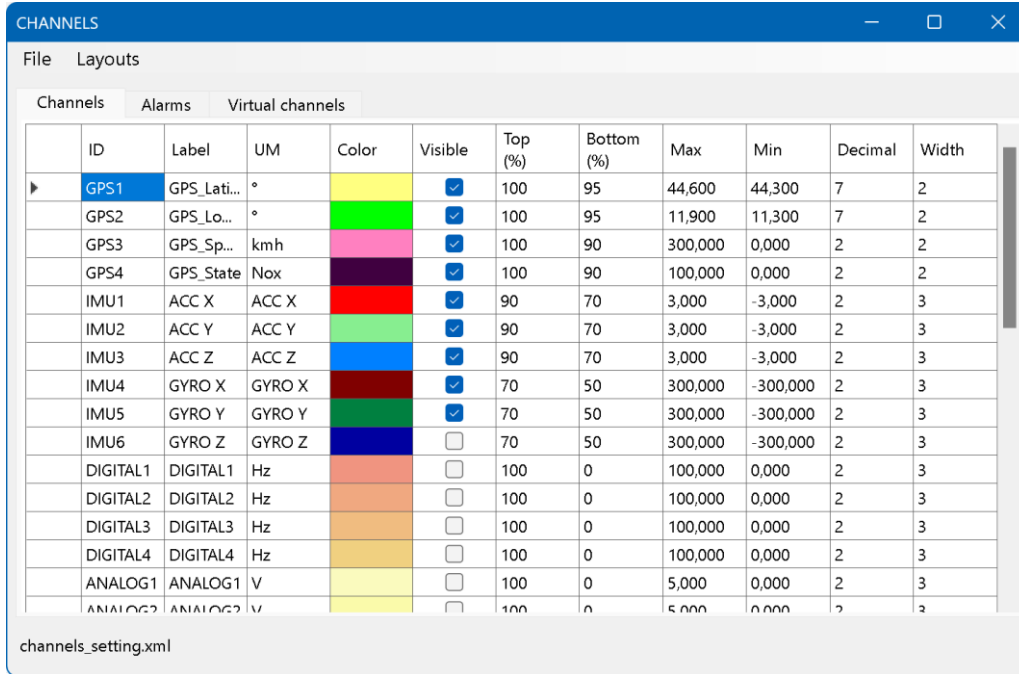
Using the CAN-BRIDGE function, the values of the inputs transmitted on the CAN bus are not converted to Float32 format. The values are retained in their original format (as read by the hardware) and sent on the CAN bus according to the following format:

Entrance	Group	Length (bytes)	Offset	Format	Formula
Latitude	1	4	0	Int32	Value/10 ⁷
Longitude	1	4	4	Int32	Value/10 ⁷
Speed	2	2	0	Word	Value/10
Status	2	1	2	Byte	Value
Accelerometer X	3	2	0	Int16	Value/1024
Y-Accelerometer	3	2	2	Int16	Value/1024
Z-Accelerometer	3	2	4	Int16	Value/1024
Gyroscope X	4	2	0	Int16	Value/16.4
Gyroscope Y	4	2	2	Int16	Value/16.4
Gyroscope Z	4	2	4	Int16	Value/16.4
Digital 1	5	4	0	Long32	Value/100
Digital 2	5	4	4	Long32	Value/100
Digital 3	6	4	0	Long32	Value/100
Digital 4	6	4	4	Long32	Value/100
Analog 1	7	2	0	Word	12-bit ADC
Analog 2	7	2	2	Word	12-bit ADC
Analog 3	7	2	4	Word	12-bit ADC
Analog 4	7	2	6	Word	12-bit ADC
Analog 5	8	2	0	Word	12-bit ADC
Analog 6	8	2	2	Word	12-bit ADC
Analog 7	8	2	4	Word	12-bit ADC
Analog 8	8	2	6	Word	12-bit ADC

The values of analog inputs 1-4 are read and converted with a 12-bit ADC: 0 Volts = 0 digits; 5 Volts = 4095 digits. The values of analog inputs 5-8 are read and converted with a 12-bit ADC: 0 mA = 0 digits; 20 mA = 4095 digits.

Chapter 6: Channels (Graphical Channel Configuration)

The **Channels module** is the tool dedicated to visually preparing your data. Accessible both from the main menu (Hub) and directly from within the analysis screen, this panel allows you to decide *how* each individual sensor should be plotted on the graph.



	ID	Label	UM	Color	Visible	Top (%)	Bottom (%)	Max	Min	Decimal	Width
▶	GPS1	GPS_Lati...	°	Yellow	<input checked="" type="checkbox"/>	100	95	44,600	44,300	7	2
	GPS2	GPS_Lo...	°	Green	<input checked="" type="checkbox"/>	100	95	11,900	11,300	7	2
	GPS3	GPS_Sp...	kmh	Pink	<input checked="" type="checkbox"/>	100	90	300,000	0,000	2	2
	GPS4	GPS_State	Nox	Purple	<input checked="" type="checkbox"/>	100	90	100,000	0,000	2	2
	IMU1	ACC X	ACC X	Red	<input checked="" type="checkbox"/>	90	70	3,000	-3,000	2	3
	IMU2	ACC Y	ACC Y	Light Green	<input checked="" type="checkbox"/>	90	70	3,000	-3,000	2	3
	IMU3	ACC Z	ACC Z	Blue	<input checked="" type="checkbox"/>	90	70	3,000	-3,000	2	3
	IMU4	GYRO X	GYRO X	Dark Red	<input checked="" type="checkbox"/>	70	50	300,000	-300,000	2	3
	IMU5	GYRO Y	GYRO Y	Dark Green	<input checked="" type="checkbox"/>	70	50	300,000	-300,000	2	3
	IMU6	GYRO Z	GYRO Z	Dark Blue	<input type="checkbox"/>	70	50	300,000	-300,000	2	3
	DIGITAL1	DIGITAL1	Hz	Light Orange	<input type="checkbox"/>	100	0	100,000	0,000	2	3
	DIGITAL2	DIGITAL2	Hz	Orange	<input type="checkbox"/>	100	0	100,000	0,000	2	3
	DIGITAL3	DIGITAL3	Hz	Light Yellow	<input type="checkbox"/>	100	0	100,000	0,000	2	3
	DIGITAL4	DIGITAL4	Hz	Yellow	<input type="checkbox"/>	100	0	100,000	0,000	2	3
	ANALOG1	ANALOG1	V	Light Yellow	<input type="checkbox"/>	100	0	5,000	0,000	2	3
	ANALOG2	ANALOG2	V	Light Yellow	<input type="checkbox"/>	100	0	5,000	0,000	2	3

6.1 Configuration Parameters (The Table)

The interface appears as a large table. Each row represents a channel in your system. Here's what each column means and how you can modify it:

- **ID (Read-only):** The original hardware name of the channel (e.g. GPS1, ANALOG4). Cannot be changed.
- **Label:** The custom name you want to give the sensor (e.g., "Lean Angle," "Brake Pressure"). This will be the name that will appear in the legend on the ANALYSIS screen (where the dataset graphs are displayed).
- **UM:** is the unit of measurement that will appear in the legend of the ANALYSIS screen (where the graphs relating to the datasets are displayed).
- **Color:** The color the line will be drawn in on the graph. **To change it, double-click the color cell** : a palette will open from which you can choose your preferred hue.
- **Visible:** A checkbox. If unchecked, the channel will be displayed in the values table but will *not* be drawn in the graph area to avoid visual clutter.
- **Top (%) and Bottom (%):** These values (from 0 to 100) allow you to confine a sensor's graph to a specific horizontal "portion" of the screen, preventing chaotic overlapping lines. For example, setting Top to 100 and Bottom to 50 will draw the channel only in the top half of the screen.
- **Max and Min:** These represent the physical limits of the vertical (Y) axis when the graph is in "Manual" mode. For example, if it's a water temperature sensor, you might set Min = 0 and Max = 120. If the graph zoom (in the Graphs window) is set to "Auto" mode, these values will be ignored.

- **Decimal:** Determines the number of decimal places with which the sensor value will be displayed in the value table.
- **Width:** The thickness (in pixels) of the line drawn on the graph. Increase it to emphasize the most important channels.

6.1.1 Tabs Channels, Alarms, Virtual channels

In the Channels window, you can select three tabs: Channels, Alarms, and Virtual Channels. The first contains the graphical settings for all 38 datalogger inputs (GPS, IMU, Digital inputs, Analog inputs, and CAN). The second contains all the alarms that can be set via the logger configuration (via the Logger window). The third contains all the virtual math channels you created via the Graphs window.

6.2 Managing Configuration Files

Using the menu above, you can:

- **Save As Global:** Saves the current changes to the channels_setting.xml file. This file is the configuration file that will open automatically when you open a dataset in the Graphs window.
- **Save Layout:** Create a new style file (template) to be reused in the future for other datasets (in the Layouts folder).
- **Open Layout:** Allows you to open a previously saved “Layout” file.

Layouts allow you to quickly switch between display styles without having to re-set all the signal parameters. You can save a layout file with only the IMU signals, or with only the engine signals, etc.

Chapter 7: Analysis (Data Visualization and Analysis)

Analysis button, you'll access the software's analytical core. When the window opens, the graph area will be empty and will only be populated if a dataset is opened/loaded.



7.1 Loading a Dataset

To view recorded data:

1. Click the folder icon in the top left toolbar (or go to **File -> Open dataset**).
2. Select the .dat file downloaded from the Datalogger.
3. The system will decode the binary file, extract the sample rates and display the entire trace on the screen, populating the side table with the list of active channels.
4. The first dataset will be loaded as dataset A. You can also load a second dataset (dataset B) to compare the values and trends of the various signals.

7.2 Chart Navigation (Zoom and Pan)

The top toolbar offers advanced controls for moving fluidly along the time (X) axis:

- **Zoom + and Zoom -:** Enlarge or reduce the chart area centered on the screen horizontally.
- **Zoom Undo:** Cancels the last zoom level applied, allowing you to go back to previous steps (the software stores your movement history!).
- **Zoom Max:** Restores the global view, showing the capture from second 0 to the end.
- **Scroll Arrows (Pan):** The <<, <, >, >> buttons allow you to scroll the graph right or left (forward and backward in time) in small (20% of the view) or large (80% of the view) jumps.

7.3 "Vertical Zoom" Mode

In the drop-down menu at the top right, you will find the **Vertical zoom item** which adjusts the behavior of the Y-axis (the height of the lines):

- **Auto:** The software automatically scales each curve so that its recorded minimum and maximum values perfectly fill the screen. Ideal for quick visual inspection.
- **Manual:** The graph strictly adheres to the *Min* and *Max* limits you set in the *Channels submenu* . Ideal for comparing different graphs while maintaining the same magnitude scale.

7.4 The Interactive Cursor and the Data Table

By clicking the mouse on the graph area, you'll notice a **dotted crosshair** that follows your movements. On the right of the screen is a data grid (**Channels**):

- When you click the mouse, **the grid on the right instantly updates**, showing the value recorded by each individual sensor at that specific moment. Alarms and virtual channels are also displayed. The visible channels are those activated from the Channels window.
- By clicking on a row in the values table (to the right of the graph area), the selected input's axis will be highlighted, displaying its graduated scale on the left side of the screen (the same color as the line). The selected graph line will be doubled in thickness (so it stands out from the others).

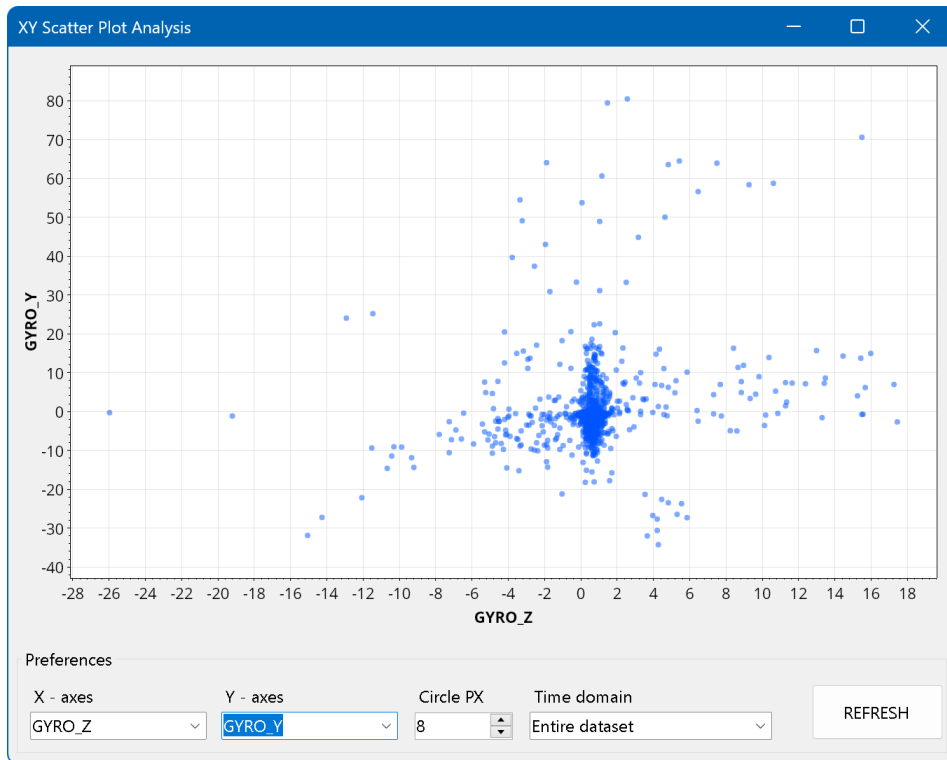
7.5 Advanced Analysis Tools

From the toolbar (or the **Graphs menu**) you can launch additional analysis modules based on the currently open dataset:

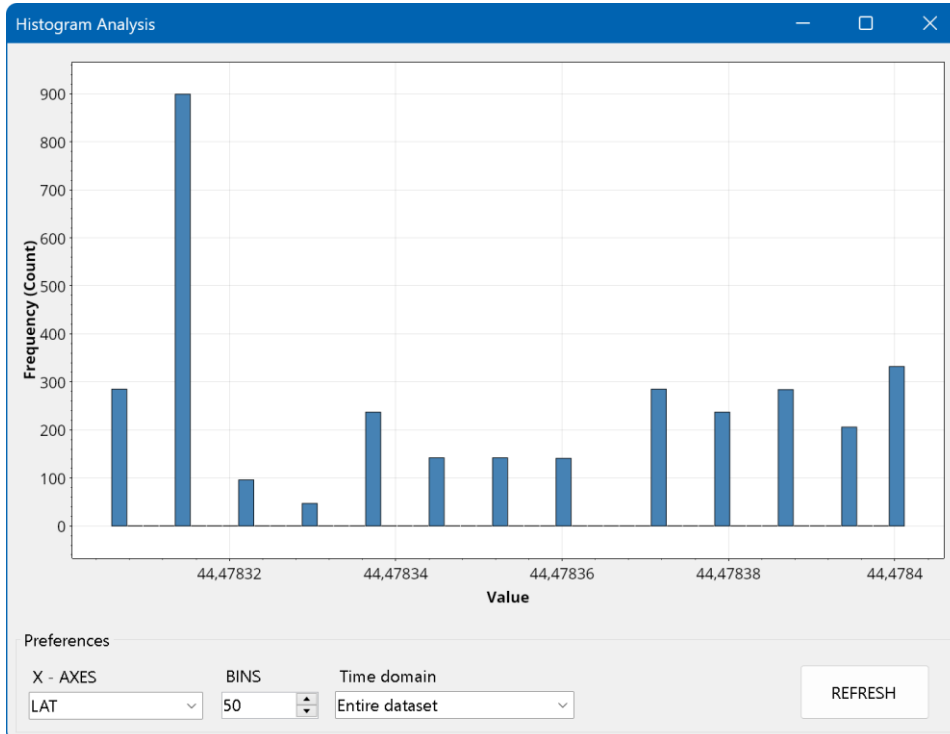
- **Statistics** [Calculator/Statistics Icon]: Generates a summary table that instantly calculates *the Minimum Value , Maximum Value , Range , Average and Standard Deviation* for each active channel in the entire session.

ID	Channel Name	U.M.	Min	Max	Range (P-P)	Average	Std. Dev.
GPS1	LAT	°	44,478	44,478	0,000	44,478	0,000
GPS2	LON	°	11,652	11,652	0,000	11,652	0,000
GPS3	SPEED	kmh	0,054	1,971	1,917	0,549	0,470
GPS4	STATUS	Nox	65,000	65,000	0,000	65,000	0,000
IMU1	ACC_X	g	-0,250	0,487	0,737	0,020	0,045
IMU2	ACC_Y	g	-0,249	0,452	0,701	0,005	0,052
IMU3	ACC_Z	g	0,578	1,465	0,887	1,034	0,034
IMU4	GYRO_X	°/s	-121,7...	91,768	213,537	1,713	13,712
IMU5	GYRO_Y	°/s	-34,268	80,427	114,695	-0,513	6,003

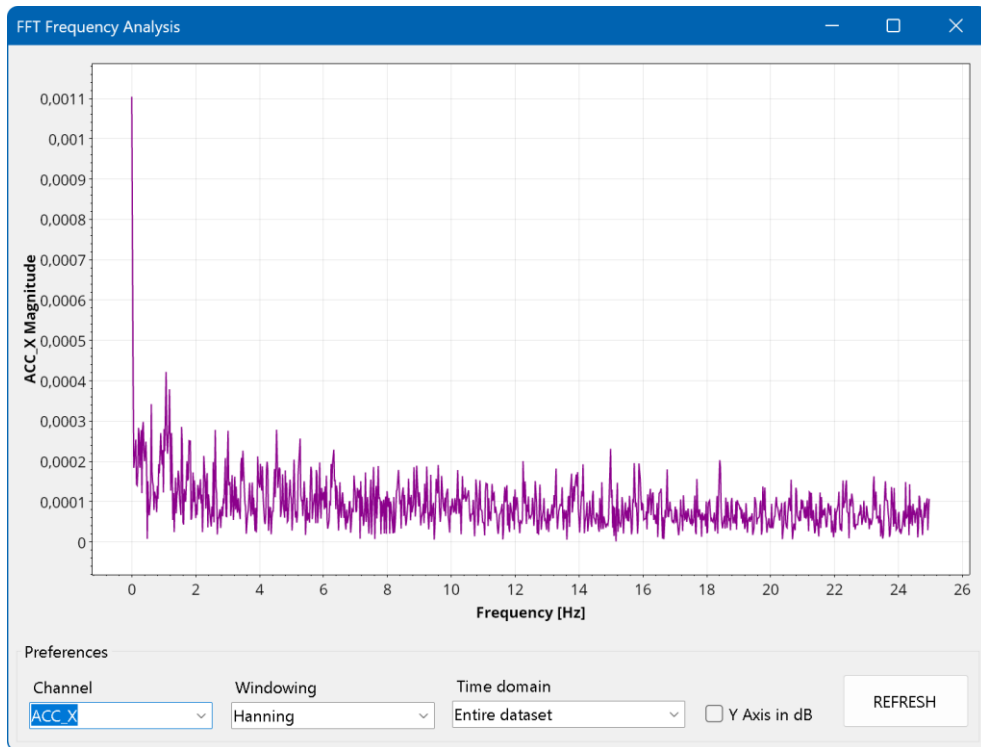
- **Scatter XY:** Opens the module for plotting one channel as a function of another (e.g. temperature and pressure).




- **Histogram:** Opens the histogram module to understand how long (statistical frequency) a sensor remained in a certain range of values (e.g. temperature map).



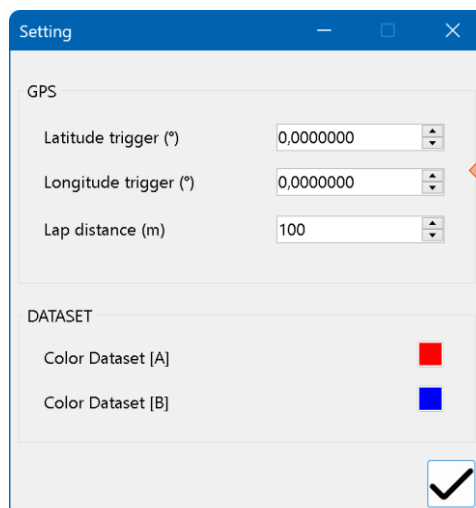
- **FFT (Fast Fourier Transform):** Starts the spectrum analyzer for frequency and vibration analysis (e.g. chatter or engine speed analysis).



7.6 GPS Trigger Detection

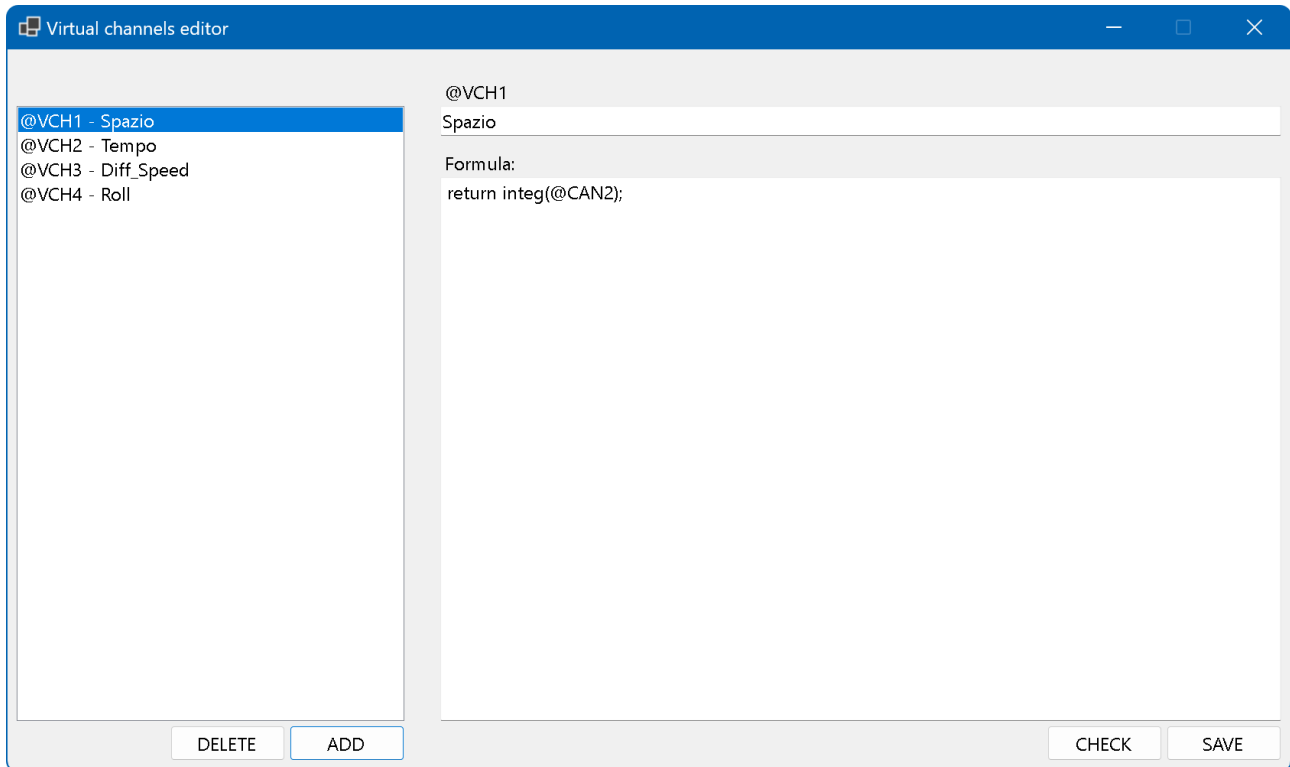
Using the button,  you can launch a GPS trigger search, which detects the device's passage through a specific GPS position. In motorsports, GPS trigger search can be used to determine when the device passes the finish line and thus determine lap times (LAPs).

To select the GPS position go to the top menu Tool > Setting and set the values related to the GPS card



7.7 Virtual Channels

Using the Tools > Virtual Channels Editor menu, you can open the Virtual Channels Editor module to generate virtual channels. The **Virtual Channels module** allows you to create new telemetry channels (e.g., speed, spacing, power, filters) by mathematically processing the data recorded by the Datalogger. The heart of the system is a high-performance vector calculation engine. You write the "recipe" (the script) for a single instant in time, and the software will automatically execute it at maximum speed for all the hundreds of thousands of samples recorded in the dataset.



7.7.1 The Graphical Interface (Virtual Channels Editor)

The editor window is divided into two main sections: on the left is the list management, on the right is the formula writing area.

- **ADD button:** Creates a new virtual channel at the bottom of the list. A unique ID will be automatically generated (e.g., @VCH1).
- **DELETE button:** Removes the currently selected virtual channel from the list (confirmation will be requested).
- **Channel List:** Displays a list of all virtual channels. **Important:** The order in this list is the order of execution. If the @VCH2 channel uses the @VCH1 channel in its formula, the @VCH1 channel must be higher in the list to be calculated first.
- **"Name" box:** Allows you to assign a human-readable name to the channel (e.g. "Path Space" or "RPM Filter").
- **"Formula" box:** The text area where you type the mathematical script.
- **CHECK button:** Performs instant code validation. Checks for typos, missing parentheses, or nonexistent variables *before* performing the actual calculation. Always use it before saving.
- **SAVE button:** Permanently saves the entire configuration to a file on your computer, so you can find it the next time you start the program.

7.7.2: Advanced Manual: Scripting and Mathematics of Virtual Channels

This document comprehensively defines the syntax, data types, and all the mathematical, algebraic, and logical functions available within the Virtual Channels Editor. The underlying language used is **C# (C-Sharp)**. Each expression is calculated instant by instant for each sample in the dataset.

1. The Semicolon (;)

In C#, **every single statement must end with a semicolon**. It signals to the engine where one command ends and the next begins.

- *Incorrect:* float x = 10
- *Correct:* float x = 10;

2. Case-Sensitive (Uppercase and Lowercase)

The language strictly distinguishes between uppercase and lowercase letters.

- Typing math.sin, MATH.SIN, or Math.Sin produces different results (only the third is correct).
- A variable called velocity is different from Velocity.

3. The mandatory "return" clause

The script must always conclude by telling the program **which number to save in the graph**. The return keyword is used.

C#

```
return @GPS1 * 2;
```

7.7.3: Data References (@ Tags)

To insert the sensor values relating to that specific moment in time into the formula, use tags preceded by the @ symbol.

- **Absolute Time:** @TIME (returns the elapsed time in seconds, e.g. 12.045).
- **Original Datalogger Channels:**
 - GPS (1-4): @GPS1, @GPS2, @GPS3, @GPS4
 - IMU (1-6): @IMU1 ... @IMU6
 - DIGITAL (1-4): @DIGITAL1 ... @DIGITAL4
 - ANALOG (1-8): @ANALOG1 ... @ANALOG8
 - CAN (1-16): @CAN1 ... @CAN16
- **Virtual Channels (already calculated):** @VCH1, @VCH2, @VCH3...

Example: Adding the analog accelerator to the brake on the CAN:

C#

```
return @ANALOG1 + @CAN5;
```

7.7.4: Data Types and Variables

Before using a value within the temporary script, it is useful to declare it by specifying its data type.

- **float (Single-precision decimal):** The standard format for telemetry. When writing a fixed number, add an f to the end to indicate it is a float.
 - *Example:* float radius = 12.5f;
- **int (Integer):** Whole numbers without a comma. Useful for counters or loops.
 - *Example:* int samples = 100;
- **bool (Boolean):** Can only take the values TRUE or FALSE. Useful for status flags.
 - *Example:* bool inBraking = true;
- **double (Double-precision decimal):** Used for astronomical or very high-precision calculations. Usually, float is sufficient.
 - *Example:* double constant = 0.123456789;

7.7.5: Algebraic and Logical Operators

Operators allow you to combine signals (recalled with the @ symbol, e.g. @GPS1).

Mathematical Operators

- **Addition (+):** return @ANALOG1 + 5.0f;
- **Subtraction (-):** return @GPS3 - @GPS2;
- **Multiplication (*):** return @CAN1 * 3.6f;
- **Division (/):** return @IMU1 / 9.81f;
- **Modulus (%):** Returns the remainder of a division (useful for loops).
 - *Example:* return @TIME % 60.0f; (Resets the time every 60 seconds.)

Logical and Relational Operators (Comparison)

They are used within conditions (If/While).

- **Same as (==):** @DIGITAL1 == 1.0f
- **Different from (!=):** @CAN1 != 0.0f
- **Greater / Less (>, <):** @GPS1 > 100.0f
- **Greater than or equal to / Less than or equal to (>=, <=):** @IMU2 <= 0.5f
- **AND (Logical AND - &&):** True if *both* conditions are true.

- *Example:* if (@GPS1 > 50.0f && @DIGITAL1 == 1.0f)
- **OR (Logical OR - ||):** True if *at least one* of the conditions is true.
 - *Example:* if (@CAN1 > 100.0f || @CAN2 > 100.0f)

7.7.6: Conditional Structures

They allow you to perform different calculations based on the state of the data.

1. The If ... Else block If ... Else

Executes a cascading block of code.

C#

```
if (@GPS1 > 200.0f) {
return 3.0f; // Very fast vehicle
}
else if (@GPS1 > 100.0f) {
return 2.0f; // Moderate vehicle
}
else {
return 1.0f; // Vehicle slow or stopped
}
```

2. The Ternary Operator (? :)

An If-Else condensed to a single line. Syntax: (Condition) ? ResultIfTrue : ResultIfFalse;

C#

```
// If CAN1 is negative, set it to 0, otherwise leave the value unchanged
return (@CAN1 < 0.0f) ? 0.0f : @CAN1;
```

3. The switch block

Useful when a variable (for example the state of a control unit) can take on many well-defined integer values.

C#

```
int mappaMotore = (int)@CAN5;
switch (mapMotor) {
case 1:
return @ANALOG1 * 1.0f; // Rain Map
case 2:
```

```
return @ANALOG1 * 1.5f; // Sports Map
    default:
return 0.0f; // Unrecognized case
}
```

7.7.7: Iterative Loops

They are used to repeat complex operations at the same instant in time.

1. For loop

Repeats an operation a predefined number of times. Requires a starting index, an end condition, and an increment (i++).

```
C#
float sumSeries = 0.0f;
for (int i = 1; i <= 5; i++) {
    sumSeries = sumSeries + (@CAN1 / i);
}
return sumSeries;
```

2. While loop

Repeats the block *as long as* the condition remains true. It is evaluated at the beginning of the loop.

```
C#
float threshold = @GPS1;
int iterations = 0;
while (threshold > 10.0f) {
    threshold = threshold - 5.0f;
    iterations++;
}
return iterations;
```

3. Do ... while loop

Similar to the while, but guarantees that the code will execute *at least once*, because the condition is evaluated at the end.

```
C#
float x = @ANALOG1;
do {
```

```
x = x * 0.9f; // Reduce by 10%
} while (x > 100.0f);
return x;
```

7.7.8: Standard Mathematical Functions

Callable by prefixing Math. (make sure to use capital letters).

- **Absolute Value (Math.Abs):** Removes the negative sign.
 - *Example:* return Math.Abs(@IMU1);
- **Standard Rounding (Math.Round):** Rounds to the nearest value.
 - *Example:* return Math.Round(@GPS1, 2); (*Round to 2 decimal places*) .
- **Round Up (Math.Ceiling):** Rounds up to the next integer.
 - *Example:* return Math.Ceiling(4.2f); (*Returns 5*) .
- **Round Down (Math.Floor):** Rounds down to the nearest integer.
 - *Example:* return Math.Floor(4.8f); (*Returns 4*) .
- **Truncate (Math.Truncate):** Removes the decimal part without rounding.
 - *Example:* return Math.Truncate(-5.9f); (*Returns -5*) .
- **Exponential (Math.Exp):** Calculates e^x .
 - *Example:* return Math.Exp(@CAN1);
- **Natural Logarithm (Math.Log):** Calculates the logarithm to base e .
 - *Example:* return Math.Log(@ANALOG3);
- **Logarithm base 10 (Math.Log10):**
 - *Example:* return Math.Log10(@ANALOG3);
- **Power (Math.Pow):** Raises a number to a specified exponent.
 - *Example:* return Math.Pow(@GPS1, 2.0f); (*Calculate the square of GPS1*) .
- **Square Root (Math.Sqrt):**
 - *Example:* return Math.Sqrt(@GPS2);
- **Sign (Math.Sign):** Returns 1 if positive, -1 if negative, 0 if zero.
 - *Example:* return Math.Sign(@IMU3);

7.7.9: Trigonometric Functions

All trigonometric functions require angles expressed in **radians** . To convert degrees to radians, multiply by (Math.PI / 180).

- **Constant Pi:** `Math.PI` (*It is approximately 3.14159*).
- **Sine (Math.Sin):** `return Math.Sin(@IMU1 * Math.PI / 180.0f);`
- **Cosine (Math.Cos):** `return Math.Cos(@IMU1);`
- **Tangent (Math.Tan):** `return Math.Tan(@IMU1);`
- **Arcsine (Math.Asin):** Returns the angle whose sine is the specified value. `return Math.Asin(@CAN1);`
- **Arccosine (Math.Acos):** `return Math.Acos(@CAN1);`
- **Arctangent (Math.Atan):** `return Math.Atan(@CAN1);`
- **Arctangent with 2 arguments (Math.Atan2):** Calculates the angle given by Y and X coordinates, useful for calculating trajectories. `return Math.Atan2(@GPS_Y, @GPS_X);`
- **Sine / Cosine / Tangent Hyperbolic:** `Math.Sinh(x)`, `Math.Cosh(x)`, `Math.Tanh(x)`.

7.7.10: Statistical Functions (Max, Min, Avg)

- **Maximum Value (Math.Max):** Returns the greater of two numbers.
 - *Example:* `return Math.Max(@GPS1, @GPS2);`
- **Minimum Value (Math.Min):** Returns the smaller of two numbers.
 - *Example:* `return Math.Min(@ANALOG1, 5.0f);` (*Keeps the signal below 5V*).
- **Simple Average (Avg):** It is calculated using algebraic operators.
 - *Example:* `return (@IMU1 + @IMU2 + @IMU3) / 3.0f;`

7.7.11: Telemetry Analysis and Signal Filters

These are special functions designed to process arrays of data in the time domain, analyzing how the signal behaves compared to previous samples.

Basic Dynamic Functions

- **Integral (Space from Time/Velocity):** Accumulates the area of the signal taking into account the sampling rate.
 - *Syntax:* `integral(value)`
 - *Example:* `return integral(@GPS3);`
- **Derivative (Acceleration from Velocity):** Calculates the rate of change from the previous sample.
 - *Syntax:* `derivative(value)`
 - *Example:* `return derivative(@GPS3);`

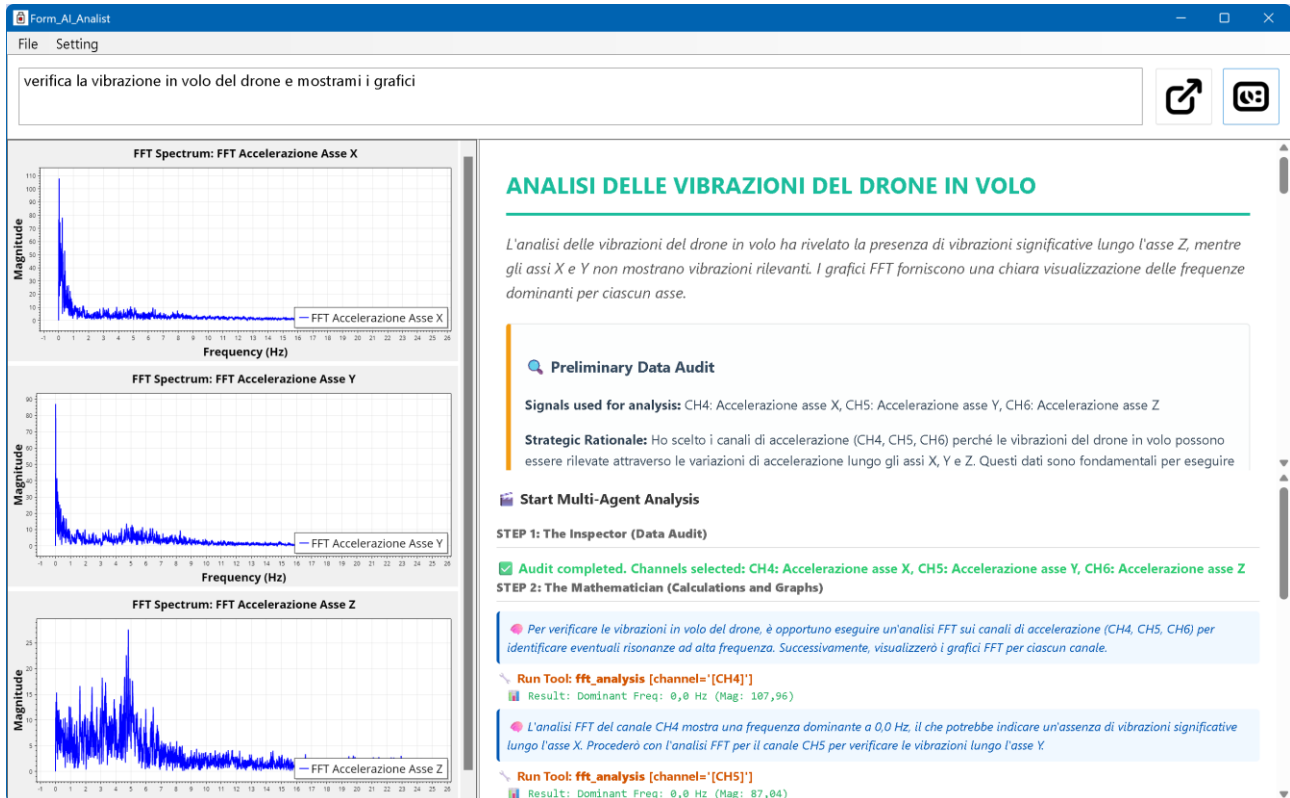
Filters (Signal Conditioning)

They allow you to clean up "noisy" signals (such as accelerometers or linear potentiometers).

- **Moving Average:** Smooths the signal by averaging the last N samples. Introduces a slight delay.
 - *Syntax:* `MovingAverage(value, number_of_samples)`
 - *Example:* `return MovingAverage(@IMU1, 10);` (*Average over the last 10 samples*).
- **Low Pass Filter (LP):** Cuts high frequencies (vibrations) and lets low frequencies (real movements) pass.
 - *Syntax:* `LPFilter(value, cutoff_frequency_Hz)`
 - *Example:* `return FilterLP(@ANALOG3, 5.0f);` (*Filters out anything that oscillates at more than 5 Hz*).
- **High Pass Filter (HP):** Cuts out continuous or slow frequencies and displays only fast variations.
 - *Syntax:* `FilterHP(value, cutoff_frequency_Hz)`
 - *Example:* `return FilterHP(@IMU2, 2.0f);`
- **Band Pass Filter (BP):** Allows only a specific window of frequencies to pass through.
 - *Syntax:* `FilterBP(value, min_frequency_Hz, max_frequency_Hz)`
 - *Example:* `return FilterBP(@CAN5, 10.0f, 20.0f);`
- **IIR (Infinite Impulse Response) Filter:** Autoregressive filter based on a damping coefficient (from 0.0 to 1.0). The closer it is to 1, the greater the damping.
 - *Syntax:* `IIRFilter(value, damping_factor)`
 - *Example:* `return IIRFilter(@GPS1, 0.85f);`
- **FIR (Finite Impulse Response) filter:** Filter calculated on the basis of a fixed window without feedback, excellent for phase stability.
 - *Syntax:* Because it requires a custom weight array, in telemetry it is usually approximated using weighted combinations of time-based moving averages or explicitly declared. In this module, using `MovingAverage` represents the purest and most applied form of rectangular FIR filtering.

Chapter 8: AI Analyst

AI Analyst is an innovative way to analyze data recorded by Datalogger. In the AI Analyst window, you'll find a field for entering queries, a menu for uploading a dataset, and an area displaying the AI's reasoning, results, and graphs. **AI Analyst** isn't just a chatbot for asking generic questions, but a true **multi-agent artificial intelligence system** integrated into the software. It's designed to assist you in analyzing telemetry data, automating the search for anomalies, creating complex graphs, and preparing professional reports.



1. What does an AI Analyst do?

When you enter a request (e.g., "Analyze vehicle cornering stability"), AI Analyst takes over the entire workflow typical of a data engineer:

- **Select sensors:** It automatically figures out which logger channels are needed to answer your question.
- **Apply math:** Leverage an advanced calculation engine to perform complex operations (Filters, Fast Fourier Transform (FFT), Integrals, Derivatives, Statistics).
- **Generate graphs:** Create dedicated visualizations (timelines, frequency spectra, histograms, XY scatterplots) to prove your points.
- **Write a Report:** Formats the results in a readable document, including the formulas used, the graphs generated and the technical conclusions.

2. How it works (Behind the scenes)

To ensure maximum precision and minimize errors, the system divides the work into three sequential phases (visible in real time in the Debug panel):

1. **The Inspector (Data Audit):** Checks the metadata of the uploaded dataset. It verifies which signals are available, discards unnecessary ones, and alerts you if any sensors that would have made the analysis more accurate are missing.
2. **The Mathematician (Processing):** Query real-world numerical data. Use mathematical tools to find peaks, calculate correlations, and draw graphs in the sidebar.
3. **The Editor (Drafting):** Collects the audit results, the mathematical results, and the graph images to draft the final document, writing it **exactly in the language you used** to submit the application.

3. How to make the most of it (Best Practices)

Artificial intelligence is powerful, but the best results are achieved by providing the right context. Here are the golden rules for achieving perfect analyses:

- **Configure Context (Essential):** Before using the AI, use the Settings > AI Configuration menu (Form_AI_Context). Give your channels clear names (e.g., "Acc Y" instead of "CH5") and add short descriptions (e.g., "Accelerometer located on the left arm"). The more information you provide, the better the AI will be at contextualizing the data.
- **Be specific in your requests:** Avoid overly vague questions like "What's wrong?" Instead, use direct prompts like: "Calculate the roll angle using the accelerometers and gyroscope. Look for any abnormal peaks above 2g."
- **Use your language:** You don't have to write in English. If you write in Italian, the AI will conduct the entire reasoning and write the report in Italian.

4. How to use Reports and Alarms

At the end of each run, the AI Analyst will present you with a hybrid HTML document in the center of the screen.

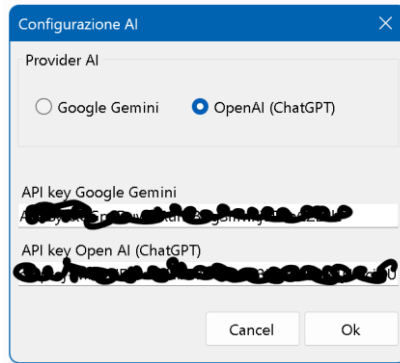
- **PDF Export:** Use the "Save Report" button to export the document as a PDF. The generated graphs will automatically be pasted into the document. It's the perfect tool for quickly sharing analyses with colleagues, pilots, or clients.
- **Automation (Alarm Suggestions):** At the end of each report, the AI will provide an "Automation Recommendations" section. It will suggest mathematical thresholds (e.g., If GyroZ > 150 deg/s for 0.5s). Use these suggestions to configure alarms in your data logger, automatically detecting the problem for future logs without having to request AI intervention again.

To make this "virtual engineer" work at its best, some initial configuration is required.

8.1 Initial Configuration and API Keys (Settings)

AI Analyst relies on top-notch generative AI models (such as **Google Gemini** or **OpenAI ChatGPT**). To use them, you must provide the software with your own access key (API Key).

1. From the AI Analyst module menu, open the **Settings item**.
2. Choose which "brain" you want to use (Gemini is often recommended for its reasoning speed, but you can opt for OpenAI).
3. Enter your **API Key**.



How to Obtain an API Key and Costs: API keys are generated by registering on the respective vendors' developer portals (for example, searching for *Google AI Studio* for Gemini or *OpenAI Developer Platform* for ChatGPT). Since these companies' web interfaces change frequently, simply follow their official online guides to generate the key. *Note on costs:* Using these APIs via the Datalogger incurs costs based on the amount of text processed. However, for typical telemetry analysis use, these costs are negligible (often fractions of a cent per analysis), and many vendors offer generous free monthly credits.

8.2 Setting the Stage: Channel Naming, Contexts, and URL Links

AI CONTEST

To get the most out of AI, assign an identifying name and description to each channel you wish to use. This will help AI generate the correct context and respond to your requests.

ID Channel	Name (ex. Water Temperature)	Description / Note to help the AI
GPS0	Latitudine	espressa in gradi decimali
GPS1	Longitudine	espressa in gradi decimali
GPS2	Velocità GPS	espressa in kmh
GPS3	Stato	65 rappresenta valori attendibili
IMU1	Accelerazione asse X	
IMU2	Accelerazione asse Y	
IMU3	Accelerazione asse Z	
IMU4	Giroscopio asse X	

Select the context in which you are using the logger and the data available. This will help the AI respond to your requests and suggest what to analyse.

ID	Context Name	System Prompt	Focus Topics
2	Drone & UAV Flight	Domain: Drone and UAV Flight. FOCUS: flight stability, motor vibrations (FFT), IMU anomalies, and battery voltage sag. ANOMALY DETECTION: Search for high-frequency resonance in motors or sudden altitude drops. PROACTIVE ACTIONS: If specific resonance is found, automatically run an FFT analysis, plot the spectrum, and suggest PID filter tuning.	Motor Vibrations (FFT), Pitch/Roll Stability, Voltage Sag, GPS Accuracy, IMU Noise
3	Go-Kart Racing	Domain: Go-Kart Racing. FOCUS: lateral G-forces, RPM drop in slow corners, braking lock-ups, and steering smoothness. ANOMALY DETECTION: Detect severe engine bogging out of corners or rear axle hopping under braking. PROACTIVE ACTIONS: If RPM drops below power band, suggest adjusting the rear sprocket ratio and	Lateral G, Corner RPM Drop, Brake Lock, Engine Temp, Steering Fluidity

Ok

AI is incredibly powerful, but it needs to know *what* it's looking at. Telling AI to "analyze the ANALOG4 channel" won't yield great results. If you tell it that ANALOG4 is "Brake pressure," however, AI will understand exactly what to look for.

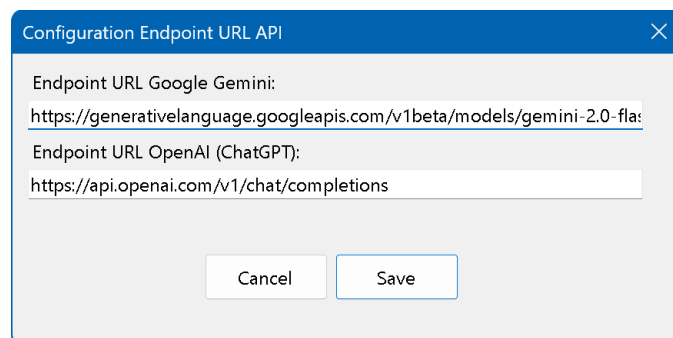
From the top menu, open **Settings -> AI Channel Info and Context** . This will open a table that's essential for training the AI:

- **Name (Channel Name):** Enter the real name of the sensor (e.g. *Water Temperature* , *Suspension Travel* , *Wheel Speed*).
- **Description:** Add a text note explaining to the AI what that channel is for. This is **crucial** for the mathematical algorithm to think clearly. (Ex: "*It helps understand how hard the driver brakes* ," or "*It helps identify front wheel bumps.*")

Context (Contexts): On the same screen, you can select the "Context" you're working in. Contexts pre-set the AI's "focus." Here are some built-in examples:

- **Drone & UAV Flight:** Focuses on flight stability, motor vibrations (FFT), and voltage drops. Looks for high-frequency resonances and suggests PID filter calibrations.
- **Precision Agriculture Tractor:** Focuses on GPS accuracy, wheel slip percentage, and power take-off (PTO) rpm stability. Suggests tire pressure adjustments based on tractive effort.
- **Industrial Machinery CMS:** Machine condition monitoring. It searches for bearing resonance frequencies and abnormal vibration peaks to prevent thermal failures.
- **Engine Dynamometer:** Engine dyno testing. Focus on torque/power curves, stoichiometric ratio (AFR), and knock peaks. Analyzes and suggests ignition delays in case of anomalies.

From the top menu, open **Settings -> AI Link** . This will open a window where you can enter links to access the Gemini and OpenAI model APIs.



Use this link for Gemini:

<https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent?key=>

Use this link for OpenAI:

<https://api.openai.com/v1/chat/completions>

8.3 How to query the AI (Best Practices)

Once you've uploaded a dataset and defined the context, you're ready to query the AI by typing in the search bar. For best results, **ask technical, targeted questions that require calculations** .

- *Weak question:* "Look at the graph and tell me how the vehicle performs." (The AI has no eyes and will give you a generic answer.)
- *Excellent question:* "Analyze the engine's vibration channels. Perform an FFT to find the dominant frequency and tell me if there are any anomalous resonances. Plot the spectrum for me."

Examples of Effective Prompts:

- ● *"Show me the frequency spectrum (FFT) of the motor vibration. Are there any dominant frequencies that indicate an imbalance?"*
- ● *"Is there a correlation between the sagging battery voltage and the analog 1 signal drain spikes? Show me a scatter plot."*
- ● *"Calculate the integral of the acceleration X to estimate the velocity and compare it to the GPS velocity."*

8.4 The Analysis Interface (Reports, Graphs and Debugging)

When you click on **Analyze** , the interface comes to life and is divided into three sections:

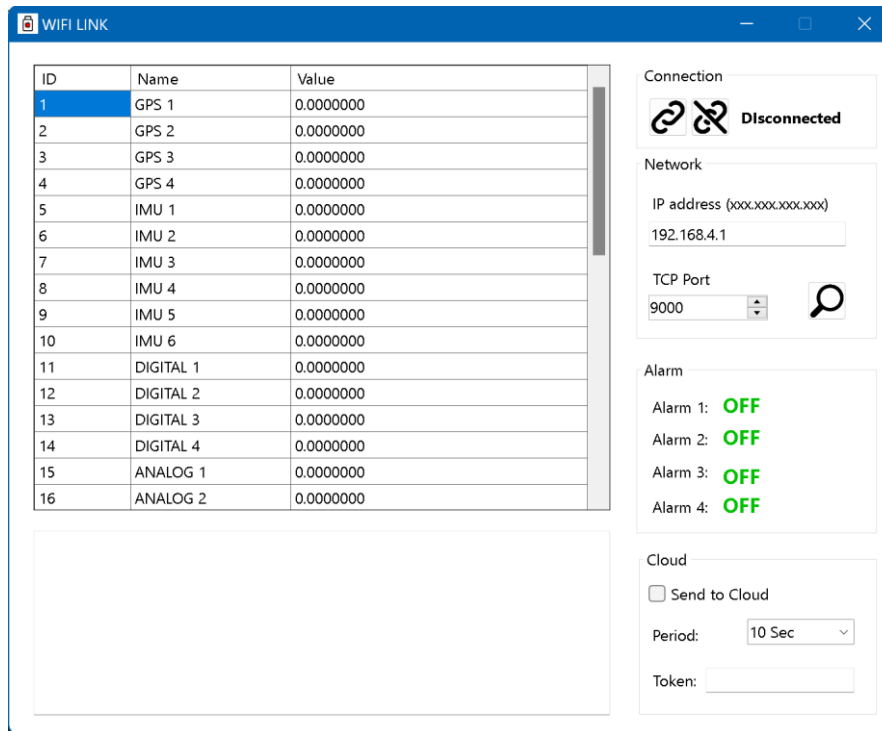
1. **The Debug Panel (Bottom Right):** This is your window into the AI's "brain." Here you'll see the machine's reasoning play out in real time (*"Looking for peaks..."*, *"Running the correlation tool..."*) and the raw mathematical results it extracts from the dataset.
2. **The Text Report (Top Right):** A readably formatted page where the AI gives you its final answer, its diagnostic conclusions and recommendations on preventative actions to take.
3. **Generated Graphs (left):** Based on its inferences, the AI will automatically draw new graphs below the text to support its thesis. The AI can generate complex time graphs (by mathematically crossing multiple channels), frequency histograms, XY scatterplots (Scatter plots), or frequency-based spectrum analysis (FFT).

8.5 Exporting the Report (Export PDF)

If you've performed an analysis that's particularly useful for a diagnosis (for example, to justify maintenance on industrial machinery or a setup for a customer), you can click the **Export PDF button** . The software will generate a professionally formatted document containing the dataset information, your initial question, the AI's step-by-step reasoning, the graphs it drew, and its final technical conclusion. A great tool to include in your test documentation!

Chapter 9: Wi-Fi Monitor (Real-Time Telemetry)

The **Wi-Fi Monitor module** is designed to transform your computer into a true remote telemetry station. With this panel, if your datalogger is equipped with a Wi-Fi module and connected to the same local network as your PC, you can view sensor data and alarm status live, without the need for USB cables.



9.1 Network Configuration and Auto-Sweep

To receive data, the software needs to know the data logger's IP address and communication port. The **Network section** contains all the necessary settings:

- **TCP Port:** The default communication port is **9000**. Make sure it matches the one set in the *Logger -> Adv Setting module*.
- **Automatic Search (Search for Logger):** No need to guess the device's IP address! By clicking the search icon, the software will identify your local network (e.g., 192.168.1.xxx) and launch a high-speed scan of all 254 possible IP addresses. In less than a second, the program will locate the Datalogger and automatically fill in the "IP address" box.
- **IP Address:** If you prefer, or if you have a complex network configuration, you can enter the IP address manually. The software will remember the last IP address and port used for subsequent sessions.

9.2 Connection and Data Grid (Real-Time)

Once you have set up your network, go to the **Connection pane** :

1. **Connect** button (green icon) to start communication.
2. The status label will change from *Disconnected* to **Connected** (green).

At this point, the **Central Grid (DataGrid)** will come to life. This table is preset to display the values of all **38 channels** supported by the system in real time:

- 4 GPS channels
- 6 IMU channels
- 4 Digital Channels
- 8 Analog channels
- 16 CAN Bus channels

The system reads a continuous stream of data and instantly updates the corresponding sensor line, allowing you to monitor the behavior of the vehicle or machinery while it is in operation.

9.3 Alarm Monitoring (Alarm 1-4)

On the right of the screen is the **Alarm box** , closely linked to the logical settings you defined previously (see *Chapter 5*). The Datalogger transmits the alarm status in real time. For each of the 4 available alarms, you will see a very intuitive text and color indicator:

- **OFF (Green Background):** The monitored sensor value is within the safety parameters.
- **ON (Red Background):** The logical alarm condition has been triggered (e.g., temperature too high or pressure dropping). This allows you to intervene promptly.

9.4 Cloud Features

Send to Cloud checkbox . Enabling this option will cause the Wi-Fi module to act as a gateway and, in addition to displaying data on screen, will also forward the telemetry packet to the cloud servers in real time for remote global storage and analysis. (You can request more information about the Wi-Fi and Cloud service directly from our website or by email.)

To safely close the telemetry session, simply click the **Disconnect button** in the Connection pane.

PART 2: Hardware

Chapter 10: Hardware Architecture and Inputs

The datalogger is designed to acquire values from 38 inputs simultaneously. This chapter analyzes the electrical and logical characteristics of each input type available on the 26-pin JAE MX23A26NF1 main connector.

10.1 Integrated Sensors (GPS and IMU)

The device is equipped with two integrated sensors that are essential for the dynamic analysis of the vehicle or machinery:

- **GPS/GNSS receiver:** Built-in module with a refresh rate of 1 Hz (for GUBELOG-01 and GUBELOG-02) and 10 Hz (for GUBELOG-03). The antenna is either integrated into the device (as with GUBELOG-01 and GUBELOG-02) or external (as with GUBELOG-03).
- **Inertial Measurement Unit (IMU):** 6-axis sensor (3-axis accelerometers + 3-axis gyroscopes) rigidly mounted on the internal PCB. It measures longitudinal/lateral accelerations and roll, pitch, and yaw angles.
Accelerometers +/- 30g, internal sampling rate 1125Hz, low-pass filter 68.8Hz. Gyroscopes +/- 2000°/sec internal sampling rate 1125Hz, low-pass filter 73.3Hz.

10.2 Analog Inputs (AN1 - AN8)

The datalogger features 8 analog inputs with a 12-bit ADC converter. The inputs are divided into two categories to ensure maximum industrial and motorsport compatibility:

- **Standard 0-5V Channels (AN1 - AN4):** High-impedance voltage inputs, ideal for reading potentiometers, pressure sensors, and pre-amplified thermocouples. Maximum tolerated voltage: -0.3V to 5.3V.
- **0-20mA / 4-20mA Current Channels (AN5 - AN8):** These 4 inputs are equipped internally with a precision 250 Ohm pull-down resistor to ground. They allow "Plug & Play" connection of industrial current sensors without additional hardware.

10.3 Digital Inputs / Frequency (DI1 - DI4)

The 4 digital inputs are designed for reading pulse signals (wheel speed sensors, engine RPM, flow meters, tone wheels).

- **Frequency and Precision:** The Hertz calculation is entrusted to a very high frequency hardware counter that guarantees almost zero latency for signals from 1Hz to 20KHz.
- **Logic Thresholds (Trigger):** The signal is considered "HIGH" above 2.5V and "LOW" below 1.0V. The maximum voltage tolerated on the pin is the battery voltage (VBatt).

10.4 Digital Outputs

The system integrates two digital outputs connected to two programmable software alarms (based on the input values) that allow the generation of a 0-5V signal. The same two outputs can drive two relays to activate external actuators (such as motors, solenoid valves, pumps, or other).

- **Voltage signal:** 0-5V.
- **Relay:** Relay drive via external power supply.

- **Maximum load:** Max. drain voltage 60V, Max. current 1.5A

10.5 CAN Bus Interface

The system integrates a CAN Bus capable of extracting up to 16 channels simultaneously.

- **Transceiver:** High-Speed CAN 2.0B.
- **Protocols:** Using the GUBELLINI DataStudio software it is possible to configure the CAN bus according to the OBD II, SAE J1939, ISOBUS (ISO 11783), OpenCAN (EN-50325-4) protocols.
- **Termination:** The 120 Ohm termination resistor is present internally in the datalogger.

10.6 Wi-Fi Telemetry Module

For applications requiring remote real-time monitoring and data transmission to the cloud, the GUBELOG-01 data logger is equipped with a Wi-Fi module. (NOTE: The GUBELOG-02 data logger integrates an LTE module instead of a Wi-Fi module.)

WizFi360 industrial network coprocessor , which autonomously manages the entire TCP/IP stack, lightening the load on the main datalogger.

Technical Features and Wi-Fi Specifications:

- **Network Processor:** WIZnet WizFi360 (Industrial Grade).
- **Wireless Standard:** Full compatibility with IEEE 802.11 b/g/n networks.
- **Operating Frequency:** 2.4 GHz (Channels 1-13), which ensures excellent obstacle penetration and a long range.
- **Data Interface:** UART [Enter the Baud Rate, e.g. 115200 bps or 2 Mbps].
- **Network Protocols:** Native TCP/IPv4 stack (used for real-time data streaming on TCP port 9000).
- **Security and Encryption:** Support for WPA / WPA2-PSK protected networks.
- **Operating Mode:** Station (STA). The module seamlessly and automatically connects to the vehicle's hotspot, garage router, or corporate Wi-Fi network.
- **Power supply and consumption:** The module is powered directly by the Datalogger [e.g. 5V or 12V via the expansion connector] with a peak transmission (TX) consumption of approximately [e.g. 230 mA].
- **Case and Connection:** [Describe the physical appearance, e.g., water-resistant ABS plastic case with quick-connect cable to the logger's B connector.]

10.7 Main Connector: JAE MX23A26NF1 (Automotive Grade)

To ensure maximum reliability in signal transmission and maintain the IPX7 waterproof certification, the physical interfacing of all 38 channels (analog, digital, CAN Bus and power) is entrusted to a single automotive grade connector: the **26-pin JAE Electronics MX23A26NF1** .

Why this standard? The MX23A series was specifically designed for the automotive and motorcycle industries for use in exposed areas (engine compartments, external frames, agricultural machinery). It features the highest level of technical features:

- **Waterproof:** The connector is equipped with silicone gaskets (sealing rings) on the coupling and individual rubber grommets for each inserted cable, preventing the entry of water, oils and dust.
- **Vibration Resistance:** The mechanical click-lock system ensures that the connector will not accidentally come loose, even when subjected to the extreme stresses of a single-cylinder engine or the curbs of a racetrack.
- **Electrical Protection:** Low-profile housing physically separates pins to prevent short circuits due to moisture.

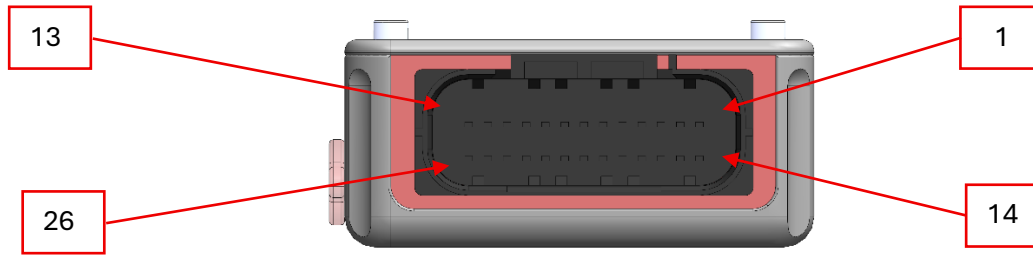
Wiring Tips (Best Practices)

If you decide to build a custom wiring harness starting from the bare connector, we recommend following these basic rules:

1. **Crimping:** Use only the crimping tool specifically designed for JAE MX23A terminals. Incorrect crimping is the primary cause of false contacts and data loss.
2. **Dummy Plugs:** If your wiring doesn't use all 26 available pins, you **must** insert the appropriate rubber "dummy plugs" into the unused holes of the flying lead. Otherwise, water will enter the connector, compromising the watertightness of the entire data logger.
3. **Wire gauge:** Use automotive grade wire (e.g. FLRY or TXL insulation) with appropriate gauge (generally between 0.3 and 0.5 mm² for signals, commonly known as 22-20 AWG).

Pin-out Table (Connector Mapping)

Below is the function of each pin on the connector.



Pin	Name	Description	I
1	DI1	Digital input 1	THE
2	DI2	Digital input 2	THE
3	D3	Digital input 3	THE
4	D4	Digital input 4	THE
5	AN1	Analog Input 1	THE
6	AN2	Analog Input 2	THE
7	AN3	Analog input 3	THE
8	AN4	Analog input 4	THE
9	DO1	Digital output 1	OR
10	DO2	Digital output 2	OR
11	CAN H	CAN line	CAN BUS
12	CAN L	CAN line	CAN BUS
13	VBatt	Diet	POWER SUPPLY
14	+12V OUT (VBatt)	Sensor power supply	POWER OUT
15	+5V OUT	5V sensor power supply	POWER OUT
16	GND	Mass	POWER
17	GND	Mass	POWER
18	AN5	Analog input 5	THE
19	AN6	Analog input 6	THE
20	AN7	Analog input 7	THE
21	AN8	Analog input 8	THE
22	***	Reserved	PROG
23	***	Reserved	PROG
24	***	Reserved	PROG
25	***	Reserved	PROG
26	GND	Mass	POWER

Chapter 11: Dimensions and Mechanical Installation

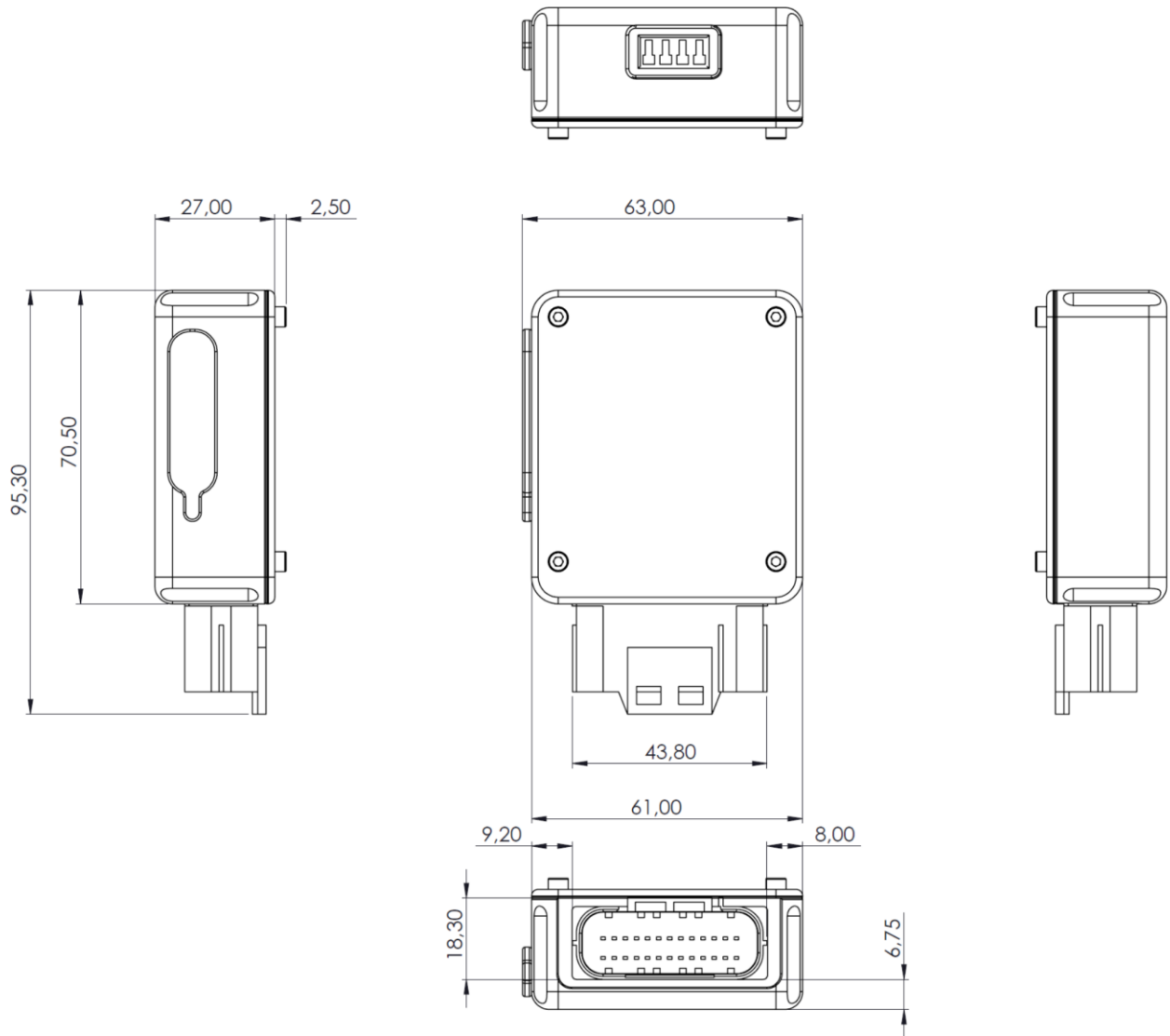
To ensure system reliability (IPX7 certified against immersion) and correct reading of the inertial sensors (IMU), the data logger must be installed respecting the dimensions and tolerances reported below.

11.1 Mechanical Specifications

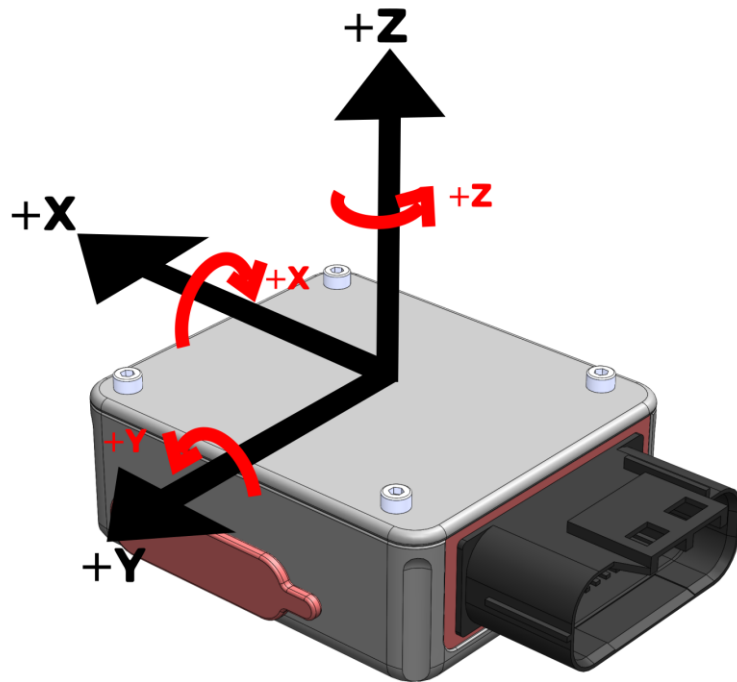
- **Casing Material:** Nylon PA12.
- **External Dimensions (L x W x H):** E.g. 93 mm x 63 mm x 30 mm.
- **Weight:** 180 grams.
- **Protection rating:** IPX7.
- **Operating Temperatures:** From -20°C to +85°C.

11.2 Technical Drawing and Dimensions

Note: The dimensions shown are expressed in millimeters (mm).



11.3 Positioning and Fixing (Guidelines)



1. **IMU Orientation:** Since the device contains accelerometers and gyroscopes, it must be mounted orthogonally to the axes of the vehicle or machinery whose movements/vibrations are to be detected.
2. **Vibration Damping:** For applications with very high vibrations (e.g. single-cylinder engines), installation using rubber silent-blocks or dual-lock Velcro is recommended.

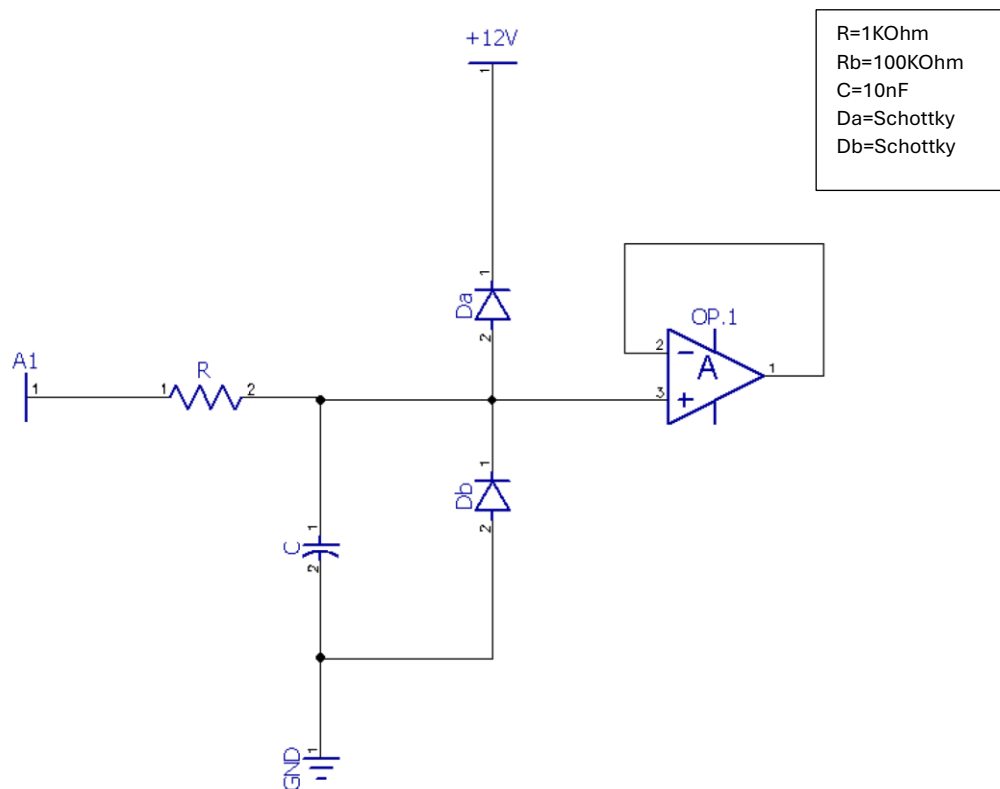
Chapter 12: Equivalent Electrical Schematics (I/O)

This section provides simplified wiring diagrams of the datalogger's internal circuitry. This information is essential for designing wiring and verifying the electrical compatibility of third-party sensors.

12.1 Analog Inputs

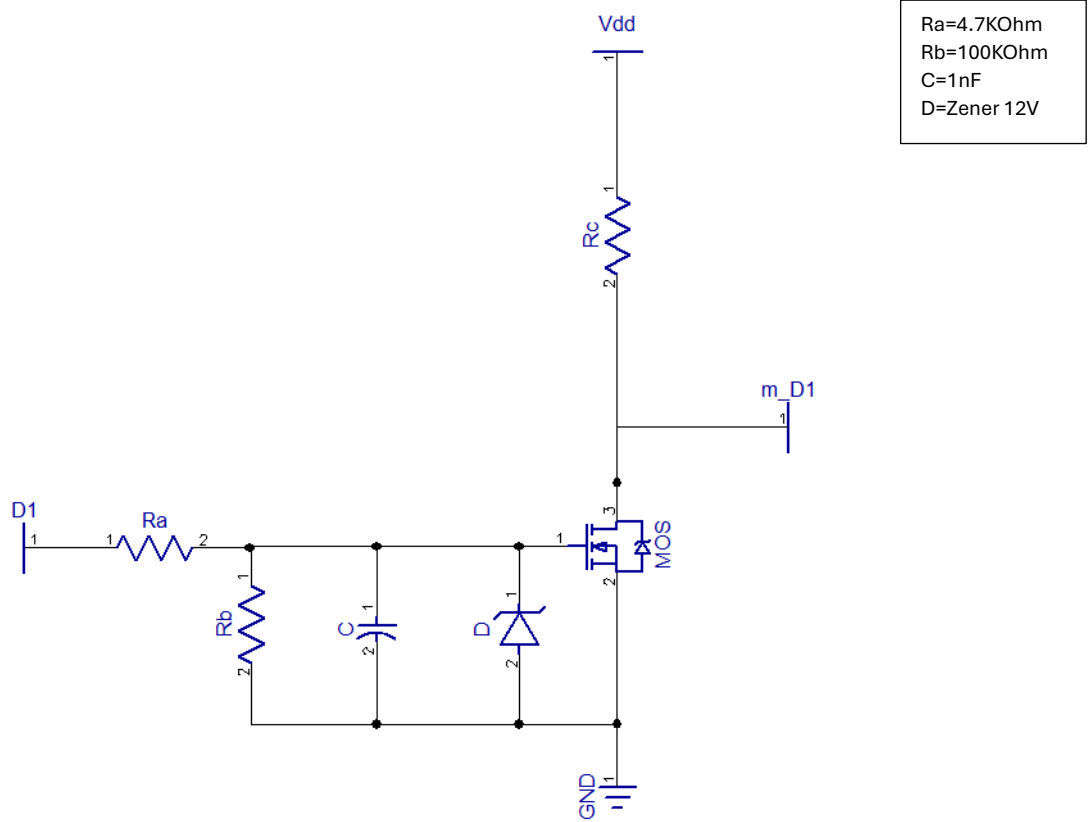
The schematic shows the equivalent input network before the ADC converter.

- Channels AN1 through AN4 feature a low-pass RC circuit with input impedance.
- Channels AN5 to AN8 include the 250 Ohm resistor for Current/Voltage conversion (between A1 and GND).



12.2 Digital Inputs (Digital / Speed Inputs)

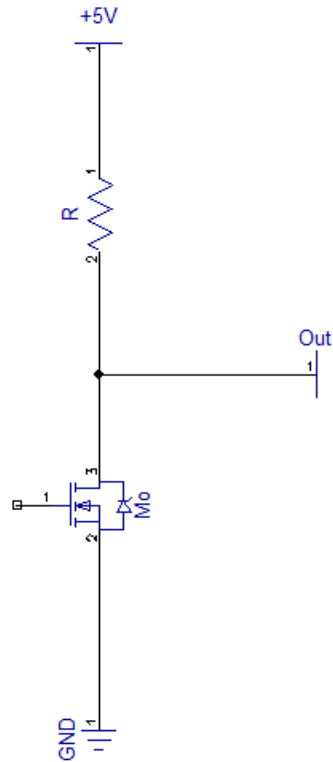
The digital input is protected against overvoltage. D1 is the digital input pin on the main connector.



12.3 Digital Outputs (Relay Module Control)

The datalogger's digital outputs are designed to drive external loads such as optional relays. The circuit uses a low-side switch architecture, meaning the output closes the circuit to ground (GND) when activated.

- Maximum current draw (I-max): 1.5A per channel.
- Protection: Equipped with internal flyback diode for loads.



R=10KOhm
M0=IRLML0060TRPbF
D=Zener 60V



GUBELLINI sas by Diego Gubellini & C.

Via Euridia Bergianti 10B 40059 Medicina BO Italy | VAT IT 03466001207

URL. <http://www.gubellinielectronics.com> – MAIL. info@gubellinielectronics.com