RAPITLog

User's Manual

V2.9 - 30/03/22

Table of Contents

Introduction

RAPITLog datalogger is a versatile data recorder. It is designed to work both on industrial areas with a high load as well as less demanding remote situations. More over, and due to its high temperature range, from -20 till 80°C, it could be installed in several places with no need for extra cooling or heating.

One of its main advantages is that is multi-branding and highly configurable. Which makes it suitable for every kind of set up. From a small temperature datalogger to a complex one with multi parameter probes and calculated variables.

On the following chapters all kinds of available configurations will be described as well as all its options. Another key feature is its power when plotting and retrieving data, both historical and live one. As well as the extra functionality that its plugins give to him.

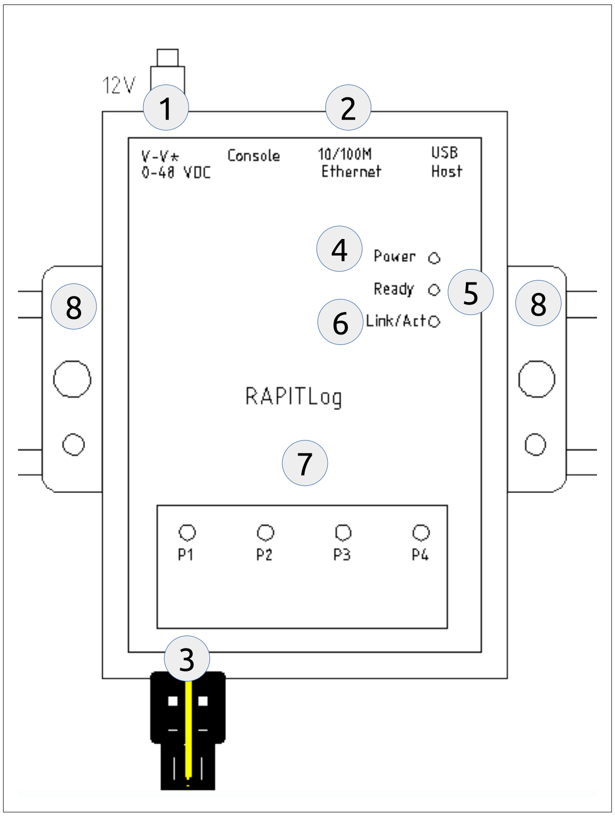
The datalogger comes with a web application both for configuration and for visualization, as well as for retrieving its stored data. This way no extra software is required and due to is mobile ready, it could be used on mobile, tablets and off course, computers.

The web application is secured to avoid non authorized people can get any data. Also it has two user profiles, Administrator, which can visualize, retrieve data and configure the datalogger. And plain users which are only allowed to visualize and export data.

One or several RapitLogs dataloggers can send its information to the RapitLog Scada. A client tailored scada to easily visualize all the information stored in it, as the client is specting.

Description

On the following schema all connections the datalogger has can be found. On the side the corresponding descriptions are described.



1. Power input. Allowed power from 9 to 48 VDC.
2. Network connection. Ethernet port for data exchange.
3. Probes communications port.
4. Power LED. Shows the board is properly plugged in.
5. Ready LED. Shows the board is working.
6. Link LED. Shows the network port is sending / receiving data.
7. Port LED. Shows that information its being transmitted on each port.
8. DIN rail support.

Product Specifications

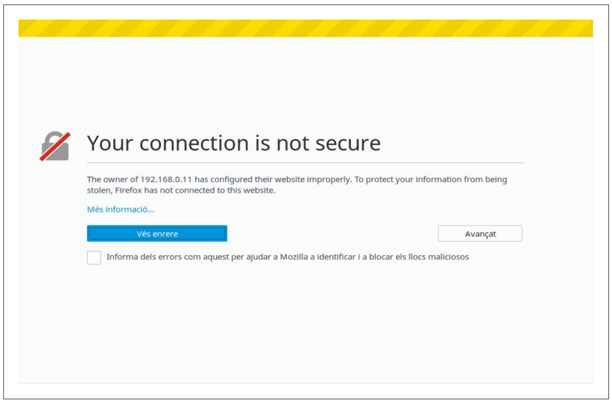
On the next table there is a description on all technical datalogger specifications.

Input Voltage:	9 -48 VDC
Power consumption:	120mA @ 24VDC
Dimensions	78 x 108 x 24 mm
Working Temperature:	-20 – 80 ºC
Regulation:	CE Class A, FCC Class A
Set up:	DIN Rail
485 / 232 Ports	4
Ethernet Ports	1
Internal Storage	16GB
Data Security Encryption	128 bits

First Steps

A web browser is the only thing needed to access the datalogger. After opening any web browser the next URL must be written https:// followed by the datalogger's IP address. By default the address is 192.168.1.10, the final URL is https://192.168.1.10.

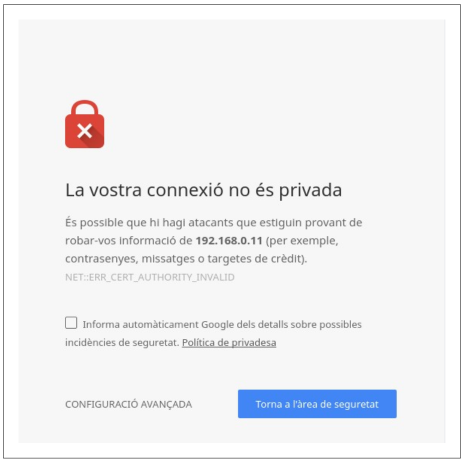
In some web browsers and advise is raised showing the system its not verified. The error will look like *UNKNOWN ISSUER* or *ERR_CERT_AUTHORITY_INVALID*. This is because the web browser does not recognizes the datalogger, although everything is fine. If that's the case, we will accept and add an exception on the advance button as shown in the following images. The web browser will keep it and never ask for it again.



Your connection is not secure

The owner of 192.168.0.11 has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

□ Informa dels errors com aquest per ajudar a Mozilla a identificar i a blocar els llocs maliciosos



La vostra connexió no és privada

És possible que hi hagi atacants que estiguin provant de robar-vos informació de **192.168.0.11** (per exemple, contrasenyes, missatges o targetes de crèdit).

NET::ERR_CERT_AUTHORITY_INVALID

□ Informa automàticament Google dels detalls sobre possibles incidències de seguretat. Política de privadesa

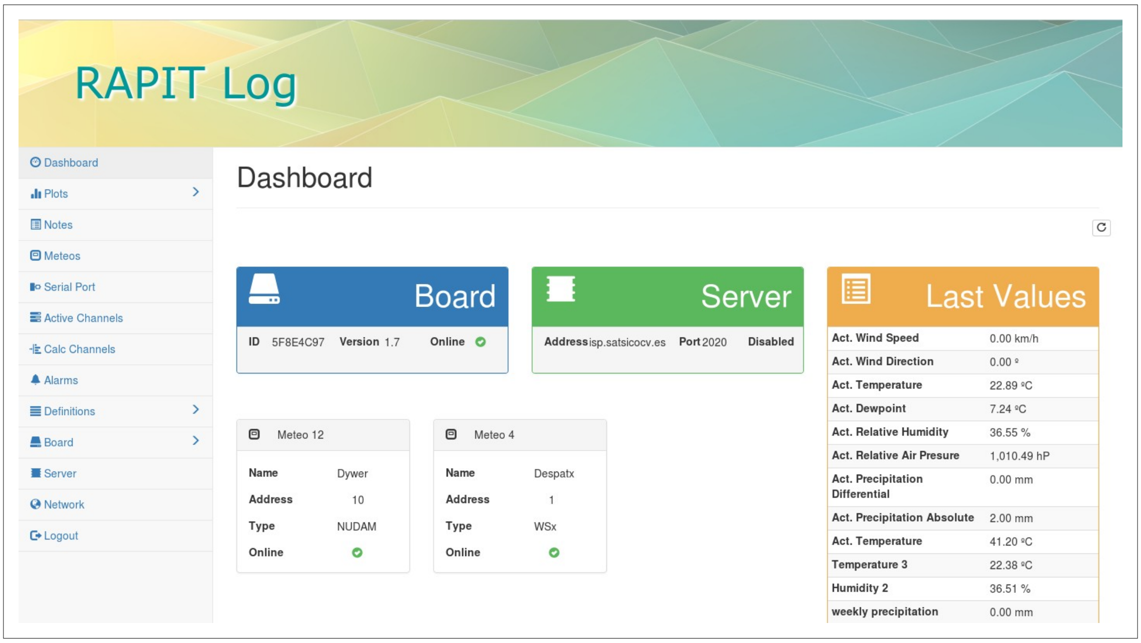
Once this is done, the login page should appear. After introducing the user and password access will be granted to the datalogger.

Visualization

On the next chapters a description to all available options to users is done.

Dashboard

This is the main window where you land after login. Here the datalogger’s general status can be found, as well as last read values.



Dashboard

Board

ID 5F8E4C97 Version 1.7 Online ✓

Server

Address isp.satsicocv.es Port 2020 Disabled

Meteo 12

Name	Dywer
Address	10
Type	NUDAM
Online	✓

Meteo 4

Name	Despatx
Address	1
Type	WSx
Online	✓

Last Values

Act. Wind Speed	0.00 km/h
Act. Wind Direction	0.00 º
Act. Temperature	22.89 ºC
Act. Dewpoint	7.24 ºC
Act. Relative Humidity	36.55 %
Act. Relative Air Presure	1,010.49 hP
Act. Precipitation Differential	0.00 mm
Act. Precipitation Absolute	2.00 mm
Act. Temperature	41.20 ºC
Temperature 3	22.38 ºC
Humidity 2	36.51 %
weekly precipitation	0.00 mm

As seen on the top image, information its enclosed in frames. Each one has information concerning one section. The first one, in blue, has information about the board. Its id, unique fore every datalogger. Firmware version and if its on-line, which means every thing works as expected.

On the next green frame it can be find out if the board is connected to the remote server for data transmission. Also it is possible to see the address and port where the data is been sent.

The yellow frame has a list of all configured variables on the datalogger with its last read value. In case there is any problem, that value is marked on red to show that. Leaving the mouse over any value will show the date the value was read. By clicking the small top right bottom an update of the data list will be triggered.

A gray frame will appear per each device plugged into the system. Here information over the device type, address, as well as its status can be found. A green check if everything its OK will be displayed. And a red cross in case there is a problem. This schema is used for *Board* and *Server* too.

With this *Dashboard* page we are able to have a global vision on our datalogger status and all its components.

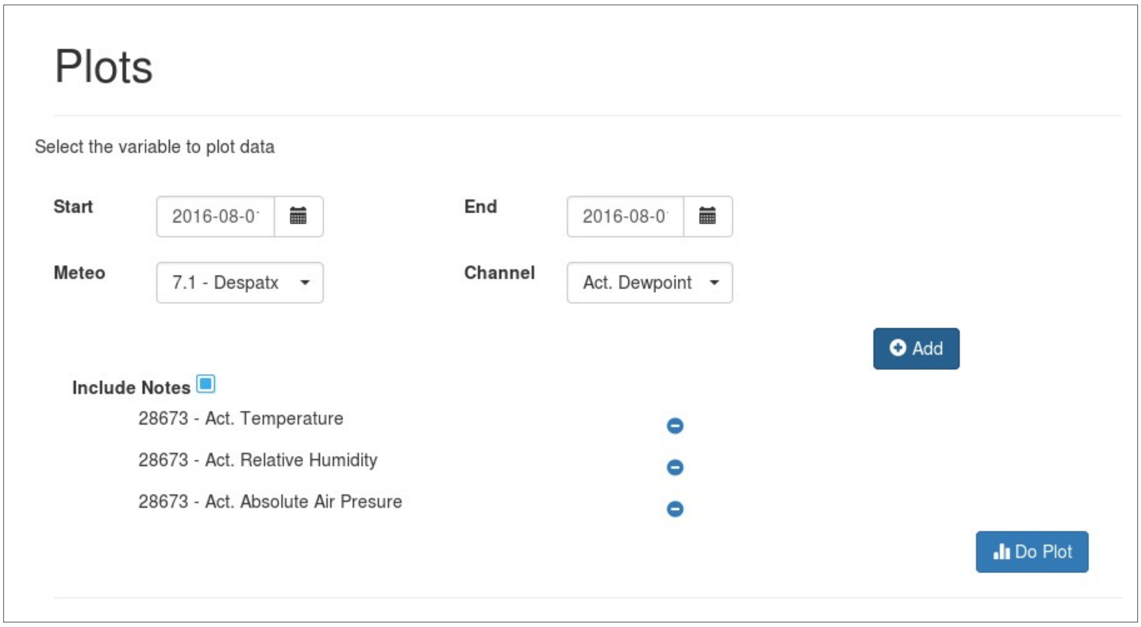
Historical Data

This section can be found under the *Data* menu entry. Here we are able to query and export all stored values in the datalogger.

To do so a start and end date must be chosen. By clicking over the calendar icon a nice calendar is displayed to easily allow us to choose the dates. After that a device among the available ones must be chosen. This will trigger the channel chooser to be filled with all available channels for that device. Picking the one wanted and clicking the *Add* button will add it to the visualization list. Up to four variables or channels can be added on every plot.

If a mistake is made by adding a non wanted channel, there is always the option to delete it by clicking on the small – button that appears on every channel.

Existing notes can be visualized by selecting the *Include Notes* check box. Those notes can be created on the *Online Plot* screen and can be used to store small amounts of text over the plots.



Plots

Select the variable to plot data

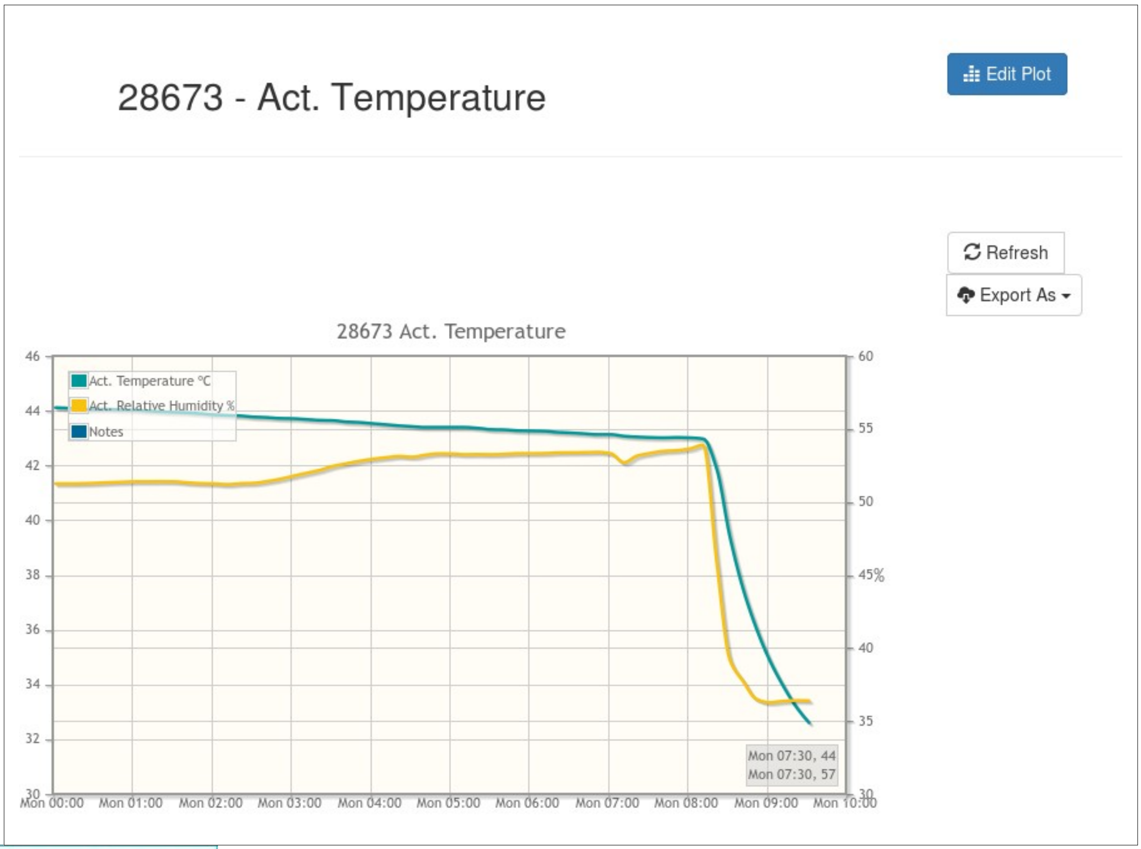
Start 2016-08-0·
End 2016-08-0·

Meteo 7.1 - Despatx
Channel Act. Dewpoint

Include Notes ☑

28673 - Act. Temperature
28673 - Act. Relative Humidity
28673 - Act. Absolute Air Presure

After all configuration its properly done, clicking the *Do Plot* button will generate the requested plot. On the next image an example can be seen. For each channel with a different unit a new axis is created. By doing it that way is easy to compare several different channels but related.



28673 - Act. Temperature

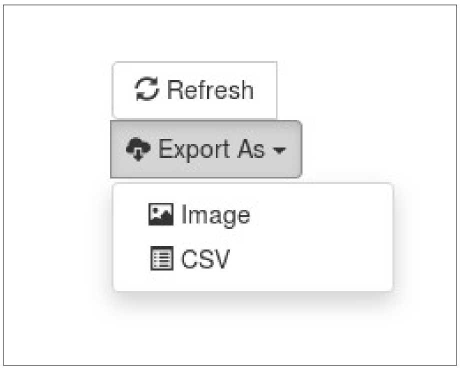
At the top left corner there is a legend with the color, name and unit of every plot line on it. Clicking over each one hides and shows each line for a cleaner view.

By clicking over the plot area and dragging the mouse, a box will appear to do zoom to better see that part with more detail. A double click will return it to the normal full view.

To find a specific value on the plot is as easy as leaving the mouse over the line and a legend will pop up with that value.

Outside the plot area to the right, two buttons can be found. The first one allows to refresh the plot in case of new values are available. The second one allows for data download.

By pressing over the second button a drop down menu shows the export options. Data can be downloaded as an image to attach it to a report or as CSV. This last format can be opened with a spread sheet software for further data processing or permanent storage.

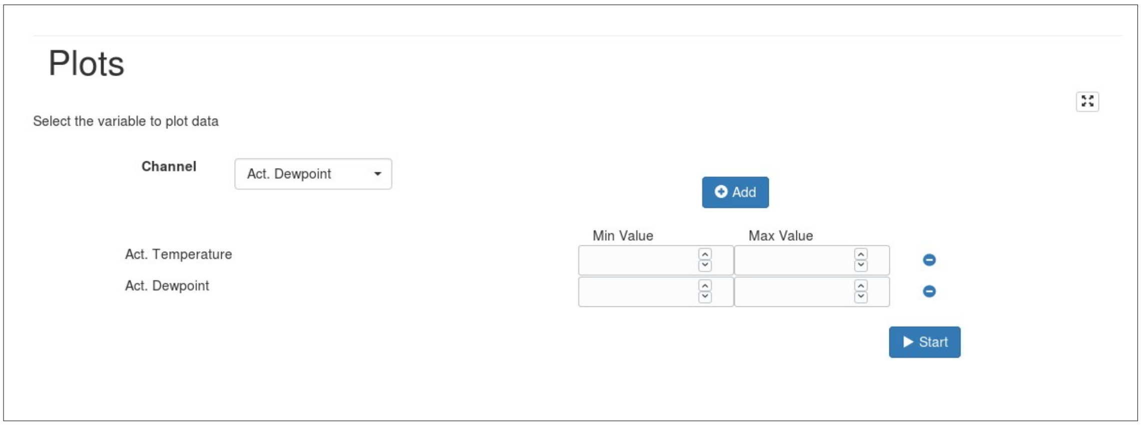


On-line Data

Inside the *Data* menu entry the on-line plots is found. The difference between the historical one is that the data plotted here is automatically updated as new data is available.

To get an *on-line* plot the wanted channels must be added from the drop down menu. As the channels are added, a *Min Value* and *Max Value* field appears as shown in the next image.

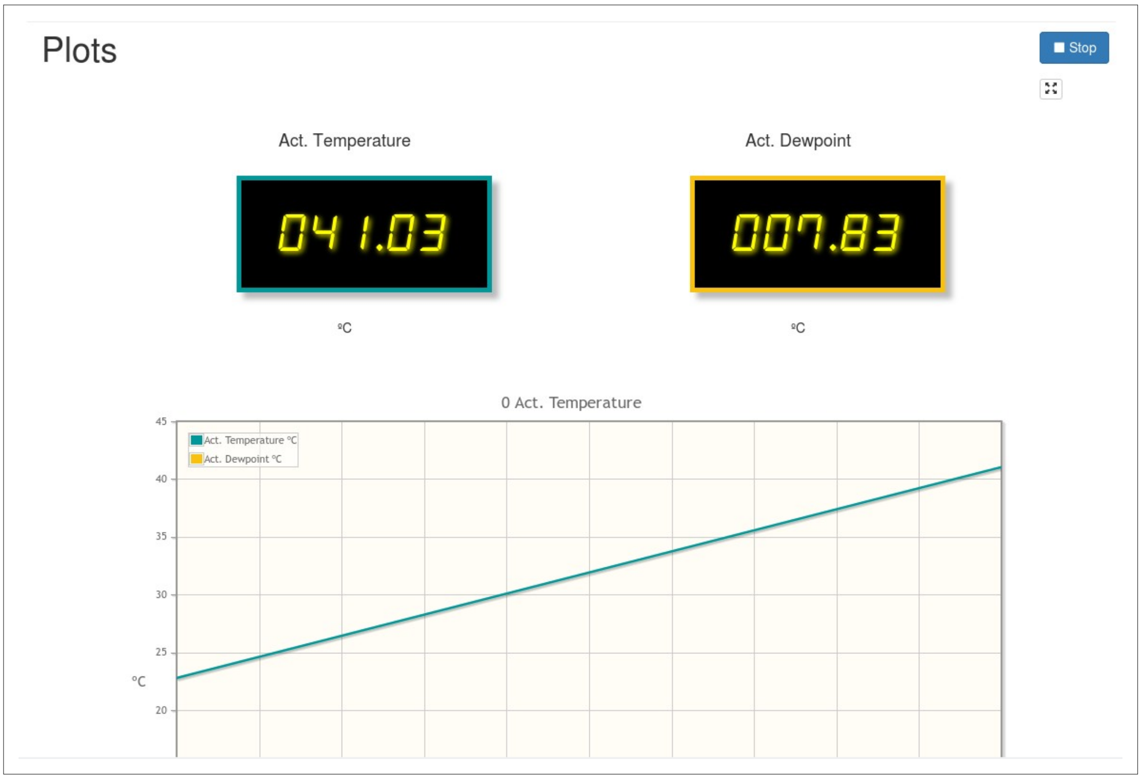
Those fields are not compulsory. But by filling them thresholds are added to the channels. If the thresholds is reached the channel indicator will be on error and it will start blinking on red to show the value is off limits.



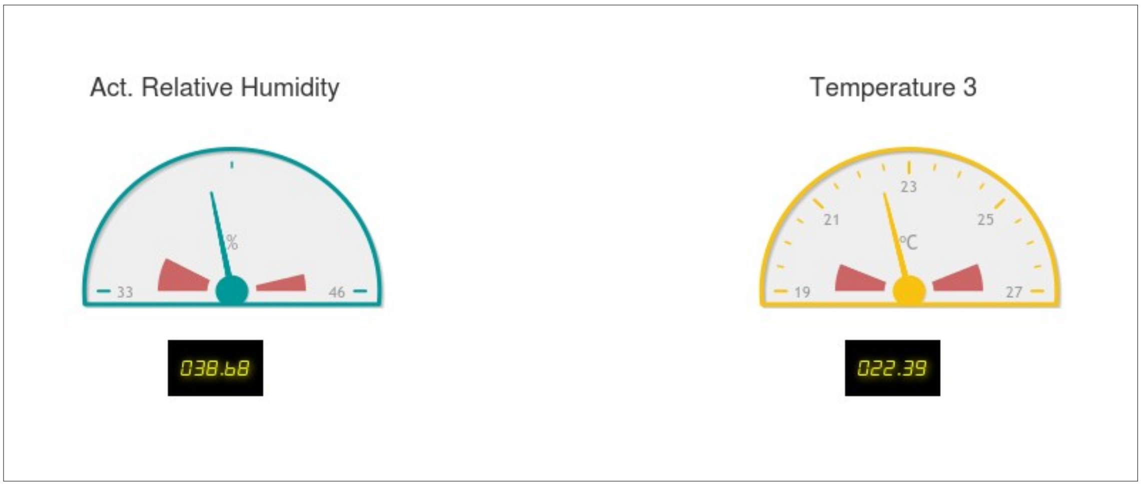
Channel: Act. Dewpoint

The visualization screen is shown on the next image. As observed a display with the current value is shown for every channel added. Bellow those, the plot, that will be automatically updated as new data comes in.

A *Stop* button is present on the top right area. This button stops the plotting and let for plotting reconfiguration. Underneath it another small button allows the plot in full screen for an easier and cleaner view.



Those displays are shown because no threshold value has been set as mentioned earlier. In case of filling those limits, the display changes to a gauge meter with the proper thresholds setup as shown in the next image.



By using this gauge meters it is easy to see if the value is close to reach an upper or lower threshold. In case a defined threshold is reached the gauge meter will start to blink in red. When it comes to its normal value the blinking stops. By working this way, even if the operator is away from the screen it is easy to spot something is going wrong.

But what happens if the operator is not always watching the screen? He could miss any incorrect value while he is away. To avoid such scenario, every time a threshold is reached a side counter it is activated. Even more, the plot it is filled with the exceed value. When the operator comes back to the screen it is easy for him to find out how many times a threshold has been reached and for how long it has been.

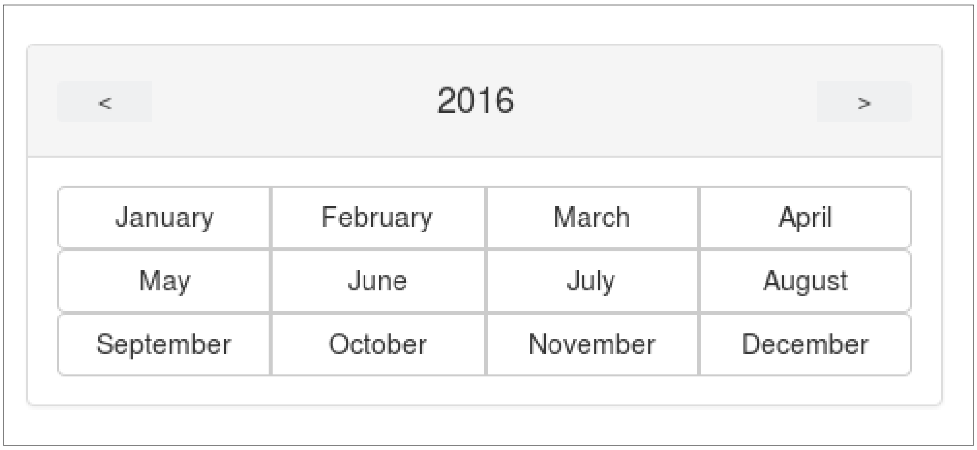
Notes can be added on this screen by clicking inside the plot area. On the next chapter a further explanation on its nature and behavior will have place.

Download Data

The last section inside the *Data* menu entry is the Download one. Here a monthly copy of the whole database can be downloaded in CSV format.

A calendar appears where the year and month can be selected as seen on the next image. To change the year two symbols appear at the edges of the year area. The right one increases a year, the left one decreases it by one. The month it’s selected by clicking over it.

Once the month is selected the CSV file must be generated by clicking on the *Generate* button. A progress bar will indicate the file’s generation progress. As soon as the file is ready for download the earlier button will change to allow the file to be downloaded.



< 2016 >

January	February	March	April
May	June	July	August
September	October	November	December

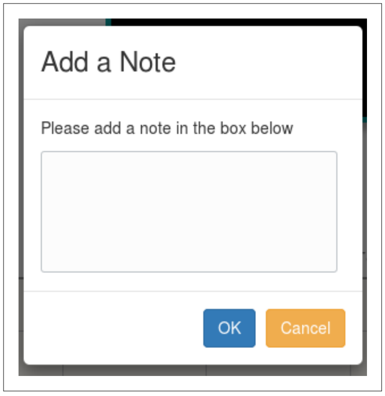
Notes

Notes can be added and consulted. Those notes are useful to add small descriptive texts to the plot to help understand the values, abnormal values, lack of value or simply to track a set of tests. Or anything else you can think of.

A list of available notes at the system is present at the *Notes* menu. Although this option is available, the common way to retrieve them is from the *Historical Plots* because the notes will be shown inside the plot with the values related that caused the motivation for the note.

For that to happen the *Include Notes* option must be checked. From this section, notes can be exported as normal data into CSV format for storage or further work.

As mentioned before, from the *Online Plots* and by clicking on the plot area notes can be added. Once one has clicked, a pop up shows up where the note text can be added. The side image show the creation note dialog. There is no need to introduce the note's date and time because the system uses the one clicked on the plot. The *Ok* button saves the note.



Add a Note

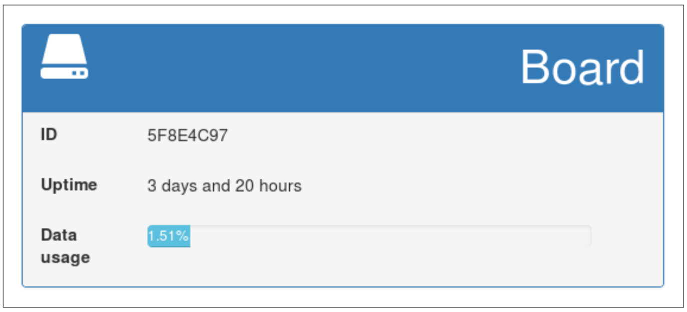
Please add a note in the box below

As many notes as wanted can be added, but it is not allowed to delete any. This is done that way by design, so no one can delete them by mistake or by will.

Board

Everything related to the board can be found in this section.

Inside the *Info* sub entry some board information it is displayed as we can observe on the side image. The board's unique id can be found. Other information available is the uptime time, as well as the amount of internal memory used by the board.



ID	5F8E4C97
Uptime	3 days and 20 hours
Data usage	1.51%

Another sub section is the actions one. Here the board can be rebooted in case we need it or a configuration change must be apply. It is compulsory to reboot the board for all the configuration changes done to take place.

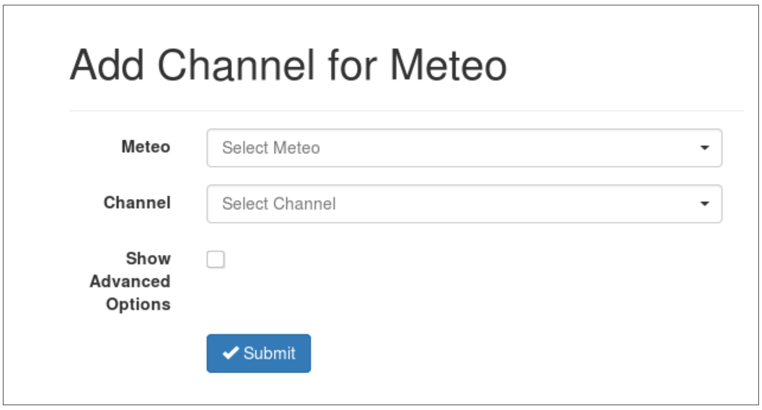
Configuration

In this section all the configuration options needed to set up the board will be described. An important thing is that a reboot is necessary to apply all changes made to the board. By not doing it so, changes may be pending till next reboot. The reboot can be done on Board → Actions menu.

Active Channels

Active channels are those the datalogger asks and stores constantly. For each channel a data type, scale and offset can be customized.

To add a new active channel we only have to press over the *Activate Channel* at the top right side. Once in, a name must be given to new active channel, then a sensor must be chosen from the first list. After this, the second will be loaded with a list of all available channels for that device. Letting us to choose the desired one. Advanced options are present by enabling them in the check. After this it is possible to choose the data type and the scale and offset. By default this is set to average data type and the scale and offset is disabled. The next image shows the



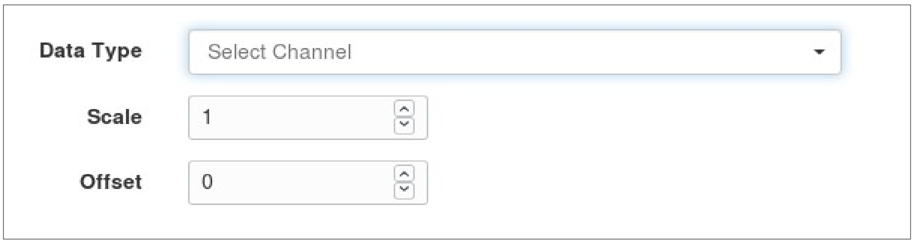
Add Channel for Meteo

Meteo: Select Meteo

Channel: Select Channel

Show Advanced Options: □

advanced options menu. On following chapters those options are described in detail.



Data Type: Select Channel

Scale: 1

Offset: 0

Data types

In this chapter all data types are explained in detail. The data type can be selected from the advance options when adding an active channel.

- **Average**.- By selecting this type the stored value is the average value between all read ones. This is the default option when creating a new active channel . And also is the more used data type.

- **Minimum**.- By choosing this one, the datalogger stores the minimum value among all read ones.

- **Maximum**.- This data type stores the maximum read value by the datalogger.

- **Incremental**.- This data type is the one it must be selected if the data is incremental. This kind of data is typically produced by meters. This devices produce data that is always equal or greater than the value before. As a water or power meter. Strange data will be store on a meter like this if this type is not selected.

- **Sum**.- This data type adds all read values from the datalogger. It could be useful if we have a pulse counter. In this case the datalogger only reads a pulse on every step if the meter reaches the configured value.

Scale and Offset

Scale an offset is a transformation factor. This can be apply to every read data to correct a known deviation. This option is found at the active channel advance form. This way the stored value is calculated multiplying the read value by the scale plus adding the offset value. The next box shows the applied formula to do so.

$$\boxed{Stored\ Value = Read\ Value * Scale + Offset}$$

An advance use for the scale and offset could be the pulse counter one. Lets say there is a pulse counter that gives a pulse every time a fixed amount of water passes through it, lets say 0,5 ml. The correct configuration for this is defining the data type to *Sum* and a *scale* equal to 0,5. The datalogger is able to store the right amount of water been consumed in ml.

Another example could be unit conversion. Lets say there is a probe that measures wind speed in m/s, but we are not used to that unit but knots. Knowing the knots speed equals 1,8 time the speed in m/s, a *scale* value of 1,8 can be setup for the datalogger do the unit conversion. The datalogger now will store and show all wind speed values in knots, making it easy for us.

Board

In this menu entry everything related to the board can be found. General information can be found at the Info section such as uptime, data usage or board's ID.

At the Actions chapter three actions can be done, reboot the board so all its changes can be applied. This is compulsory after any configuration change. Otherwise changes will be saved but not applied. The second action its a safety button to start the board's main logic in the strange case it went down. And the last action forces a read by the board. This is useful on high polling time systems where there is a need to check for a change and waiting for its natural polling is not an option.

Board Properties

Also the property chapter can be found in this *Board* section. Here internal datalogger properties can be setup, among those we find:

- **DISPLAY_WINDOW.**- This field sets the amount of time data is plot at the *online* plot. After this time, older values are fadeout from the plot. This value is set in minutes.

- **POLLING_TIME.**- This sets the polling time. This is how often is the datalogger going to read a value. This property it is defined in seconds.[1]

- **SAVE_RATIO.**- This is the save ratio property. Which tells the datalogger every how many pollings it should store a value.[2]

- **FORCE_SNOW_FILM.-** Only for DGT plugin. When this field is set, sends 0 as snow film when the road status is not either wet, icing or snowing. Otherwise, always sends the read snow film value.

Polling / data saving logic

The datalogger logic work as follows. Each POLLING_TIME seconds the datalogger reads a value and after SAVE_RATIO polls a value is stored. The saved value could be the average, maximum, minimum… depending on what is defined for each variable on its data type.

Lets say a POLLING_TIME of 60 and SAVE_RATIO of 5 is defined. And, also, *Average* has been set as data type. This means that each sixty seconds, every minute, a value is read by the datalogger. This will be repeated five times before saving the data. So at the datalogger's records, a value will be found every five minutes with the average value between all reads done every minute. The next formula shows how the saving times are calculated.

$$\boxed{Store\ time = POLLING\ TIME * SAVE\ RATIO}$$

[1] This field does not apply when the DGT plugin is active. In this case, this value is set to 10 seconds.

[2] This field does not apply when the DGT plugin is active. In this case, this value is set to follow the configured *Integration Period* value of the DGT protocol.

Plug-ins

This module can be found inside the *Board* menu entry. Here a list of available plug-ins will be shown. A plugin is an piece of software that adds extra functionality to the RapitLog. Notice that each plug-in requires a license to be enabled. Contact us at informatica@sicocv.es for more information.

After enabling it and introducing the correct license a new menu entry will be created where the plugin status or configuration can be checked..

The following plug-ins can be found at the RapitLog:

- 👁 **Monitor**.- Enables 24x7 monitoring over RapitLog. With this plug-in the datalogger is constantly monitored for possible problems. On a central service all RapitLogs are supervised and alarms can be set when the datalogger encounters a problem.

- 🔔 **Alarms**.- Adds the possibility to define alarms over data and act accordingly.

- ☰ **Calculus**.- Adds the ability to have calculus channels or virtual values such as counters, maximum, subtractions, formulas …

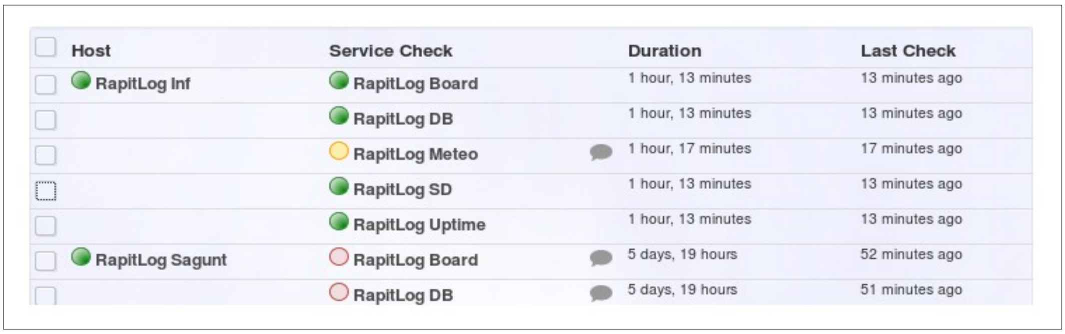
- 🔀 **Irs21**.- This plug-in translates the IRS21 ASCII protocol into the new IRS31 one. This enables old systems to use a new IRS31 sensor.

- 🛣 **DGT**.- This plug-in implements the UNE 199071-2014 spanish protocol to communicate with road stations that monitors atmospheric information in traffic control systems

- B **ModBus**.- This plug-in enables the datalogger to export all its instant values in ModBus standard. It can be ModBus RTU or ModBus TCP/IP.

Monitor Plug-in

This plug-in enables a 24x7 monitoring over RapitLog. With this plug-in the datalogger is constantly monitored by a central service that checks all RapitLog parameters are working as expected as it can be seen on the next image. When an error or warning occurs the central service generates an e-mail to notify the problem to the people in charge

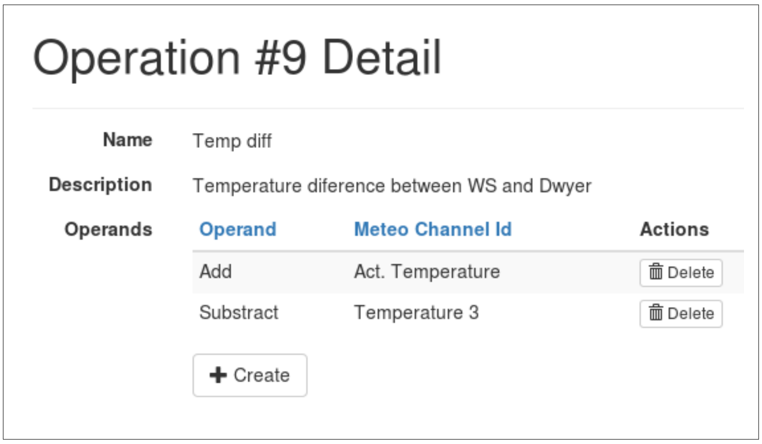


□	Host	Service Check	Duration	Last Check
□	● RapitLog Inf	● RapitLog Board	1 hour, 13 minutes	13 minutes ago
□		● RapitLog DB	1 hour, 13 minutes	13 minutes ago
□		○ RapitLog Meteo	1 hour, 17 minutes	17 minutes ago
□		● RapitLog SD	1 hour, 13 minutes	13 minutes ago
□		● RapitLog Uptime	1 hour, 13 minutes	13 minutes ago
□	● RapitLog Sagunt	○ RapitLog Board	5 days, 19 hours	52 minutes ago
□		○ RapitLog DB	5 days, 19 hours	51 minutes ago

Alarms Plug-in

This plug-in enables the possibility to define thresholds, that once passed will generate an event. Those events can be, simply stored at the datalogger, or send by e-mail or a whatsapp or telegram message to a predefined user. Or even to activate an external relay to notify about the alarm to another system.

Calculus Plug-in



Operation #9 Detail

Name: Temp diff

Description: Temperature diference between WS and Dwyer

Operands:

Operand	Meteo Channel Id	Actions
Add	Act. Temperature	Delete
Substract	Temperature 3	Delete

This is a plug-in and must be enabled at the *Board → Plugins* section. With this plug-in calculus channels, also known as virtual variables, can be created. Unlike physical or real channels, calculus channels are those that its value comes from a complex operation over one or more real values.

The addition of a new calculus channel can be found in the *Calc Channels* section at the *Create* button. The creation is split in two steps. At the first one a name and a description on the calculus channel must be set. Once this is done, and by entering on it, it is possible to add the real channels that are going to intervene on the operation. An example is shown on the side image.

Every real variable involve in the calculus channel it is set with an operand that defines its behavior. The next chapter describes all available operands in detail.

Operations Types

In this section existing operation types are explained. Each physical added variable to a calculus channel has an operation type that defines its behavior inside the calculated variable. Some operation types or operands can be combined in the same calculus channel, like *Add* and *Subtract*. By contrast, others only support one physical variable like counters and historical average.

The following list describes all available operations types.

- **Add**.- This type means the physical value will be added to the calculus one.

- **Subtract**.- This type means the physical value will be subtracted to the calculus one.

  This two operations, unlike the others, can be combined. So if we want our calculus channel be the addition of two physical ones, those real variables should be included with *Add* type. If what is wanted is the difference between two physical ones, this can be achieved including the first physical channel with type *Add* and the second with type *Subtract* to obtain it.

- **Minimum**.- This type, as the name suggest, stores the minimum value between all the physical channels added. This is why, all types included in the calculus channel must have *Minimum* as type.

- **Maximum**.- This type, is the opposite to the one before. It stores the maximum value between all the physical channels added. This is why, all types included in the calculus channel must have *Maximum* as type.

- **Average**.- By using this type the calculus variable is set to the average value between all the physical values included. This is why, all types included in the calculus channel must have *Average* as type as the two before.

- **Daily Counter**.- This type does a daily counter. Which means it increases the value read to a counter and resets it every day. This type only allows one physical channel.

- **Weekly Counter**.- This type behaves as the one before, but does the reset every week.

- **Monthly Counter**.- This one acts in the same way as the two types before but sets the counter to zero every month.

- **Yearly Counter**.- This one behaves as the others but it resets every year.

- **Uptime Counter**.- This counter resets on board's next reboot.

- **Hourly Counter.-** This counter resets every hour.

- **Trend Line**.- This type calculates an historical average. It keeps track of the last ten values read and calculates the average using them. This average is less prone to sudden changes because of that. It only admits one physical channel on it.

- **Energy**.- This type calculates the energy produced by a power channel. This is the amount of work done over time. This type only allows one physical variable.

- **Beaufort Scale.-** This type calculates the wind type according to the Beaufort Scale

- **Gust.-** This type calculates whether a wind gust has been produced.

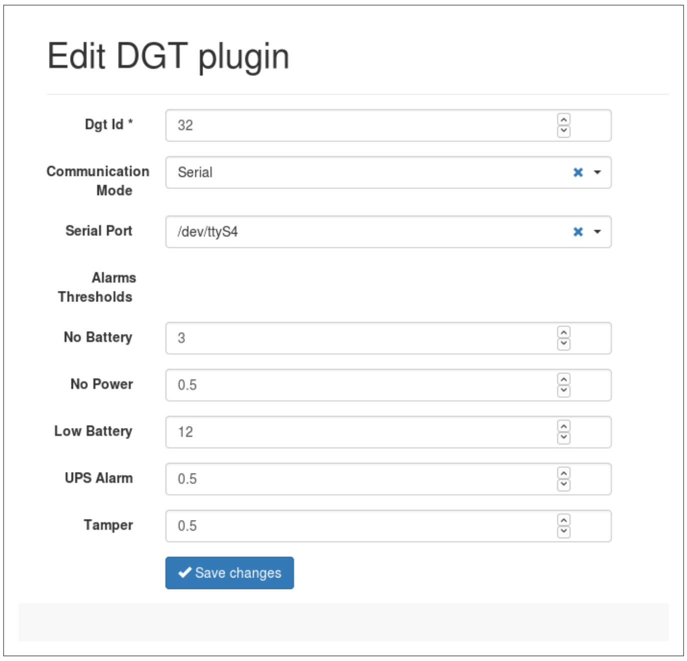
IRS21 Plug-in

This plugin acts as a translator for the new IRS31 devices on systems that only allow the old IRS21 ASCII protocol. This plugin listens on port TCP 4001 and it's able to understand IRS21 questions an answer it with the IRS31's values. This way old systems don't need to be modified in case an IRS21 sensor has to be replaced with the newer one.

DGT Plug-in

This is a plug-in and must be enabled at *Board→ Plugins* section. This enables the datalogger to act as a UNE 199071-2014 datalogger, commonly known as DGT protocol. This means that is able to talk with systems that manage weather information on traffic control.

The standard protocol defines communications to be serial only and at an specific speed. We have develop this plug-in in a more flexible way. The datalogger allows any kind of speed over the serial communication and also adds a more modern way of comunications, the TCP/IP protocol. For this reason, once enabled, communication mode must be defined. If serial communication is chosen, a configured port must be selected. This port must be created beforehand at the *Serial Port* section. If TCP mode is selected, only the port where the board will listen to, must be filled.



Edit DGT plugin

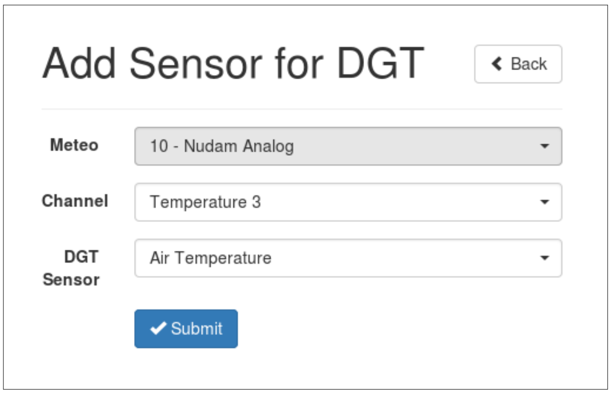
Dgt Id *	32
Communication Mode	Serial
Serial Port	/dev/ttyS4
Alarms Thresholds	
No Battery	3
No Power	0.5
Low Battery	12
UPS Alarm	0.5
Tamper	0.5

The other compulsory field is the DGT ID. The protocol defines that for multi point communications, (in which more than one system is plugged in), addresses will start on 0x20 and every new device will be set a consecutive number starting from this point. If otherwise, communication are point to point, this is, just one system, it will have ID equal to 0x20. Another important thing is that the protocol addresses are expressed in hexadecimal notation, and the datalogger expects them in decimal notation. So if we want to assign the ID 0x20 to the system, a 32 digit must be written on the datalogger configuration.

The DGT protocol also defines a set of alarms. The datalogger can use any probe its able to read as an alarm. For this purpose threshold are needed. At the Alarms Thresholds form this can be configured. So an alarm will be active if the probe value is lower than the one defined in its threshold and it will be deactivated if the value is higher than the threshold.

To let the plug-in know about the sensors and alarms available to use, it is needed to add those sensors at the top right Sensors button. Here all available sensor will be displayed. To add a new one the Activate Sensor button must be pressed.

On the form to add a new sensor we have to select three fields. The first one is the Meteorological device from where the sensor is going to be read. The second one is the actual sensor, or channel we want. And the last one is the specification of what type of sensor is in the DGT system. It is like a sensor type and It could be a meteorological sensor or an alarm. The system allows up to three sensors with the same type.


Add Sensor for DGT

Meteo	10 - Nudam Analog
Channel	Temperature 3
DGT Sensor	Air Temperature

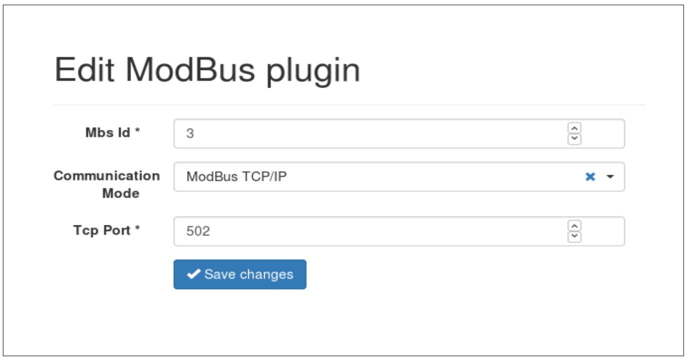
It must be mentioned that the protocol defines a configuration frame. In this case it’s only used to enable or disable the configured sensors using the above procedure. This is, for example, if we have configured three sensors using the upper procedure and the datalogger receives a configuration frame containing only two of the configured frames, the third one will be disabled. If for example, a configuration frame arrives with four sensor the system will raise an error because it does not know anything about that fourth sensor.

Thresholds are not well defined in the protocol. For this reason we do not support per sensor thresholds and only allow up to two thresholds per measurement type. In this case this implementation of the protocol takes the following rule. The first configured type threshold indicates a minimum threshold, which means the value cannot be underneath this value. The second type threshold, if defined, is a maximum threshold, which means the value cannot be higher than this threshold value.

This protocol also talks about a weather prediction alarms that must be raised with two hours in advance the condition appears. So far we don’t have any way to predict those alarms. This is why no atmospheric previsions alarms are defined in this implementation of the protocol.

ModBus Plug-in

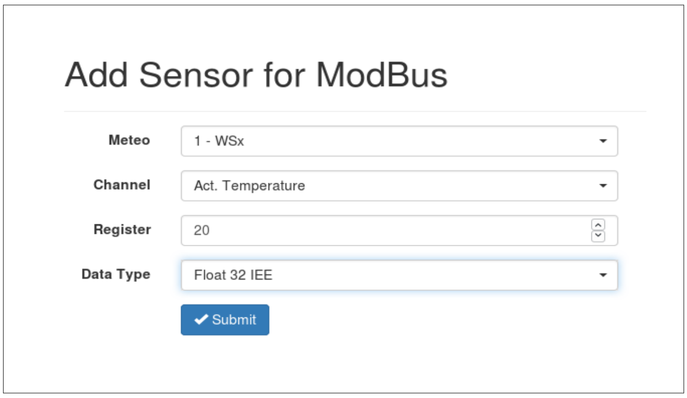
This is a plug-in and must be enabled at *Board→ Plugins* section. This plug-in adds the possibility to export all instant data with the ModBus slave protocol. This protocol is the facto standard protocol in industry.


Edit ModBus plugin

Mbs Id *	3
Communication Mode	ModBus TCP/IP
Tcp Port *	502

This protocol has several

variations depending on the type of communications. Here ModBus RTU and ModBus TCP/IP are implemented. The first one uses a serial communication port and the later, as the name suggest, TCP/IP. In the ModBus protocol, every slave device needs a unique address or Id to distinguish it from other devices in the same network. This parameter can be configured in the *ModBus ID* field as shown in the side image. Here also the communication type can be selected. If choosing ModBus TCP/IP a port must be provided where the datalogger will be listening for petitions. In case of choosing ModBus RTU a serial port must be selected from the available list. This port must be created beforehand at the *Serial Port* section.


Add Sensor for ModBus

Meteo: 1 - WSx
Channel: Act. Temperature
Register: 20
Data Type: Float 32 IEE

Datalogger data can be mapped into Modbus Input Registers, function code 4. Apart from the configured sensors, the first ten registers are fixed and cannot be configured. On those registers, useful information about the board and its status is represented. So starting with address 10 input register can be mapped to the different values the datalogger is able to read as shown in the image on the side.

To maximize interoperability with other devices or PLC's, several non standard data types are been implemented. ModBus standard describes data as a 16 bits integer, but often this is not enough in modern applications that require better precision or bigger values. This is why the datalogger is able to output data as 16bits integer, or with a decimal multiplier, or with a centimal multiplier or even as a 32 bits IEEE float number. Bellow a deeper description on the implemented data types can be found.

- **Int16**.- Standard format occupying one register. The value is written as integer, with no decimal value at all.

- **Decimal Multiplier**.- Non standard format for real numbers. The value written is the value read multiplied by ten. Allowing then, for one decimal precision. This data type occupies one register.

- **Centimal Multiplier**.- Non standard format for real numbers. The value written is the value read multiplied by a hundred. Allowing then, for two decimals precision. This data type occupies one register.

- **Float 32 IEEE**.- Non standard format for real numbers. The value written complies with the IEEE standard known as a single-precission floating-point format. This data type is the most precise of all the above but it uses two registers.

As mentioned before, this plugin always export a set of register with status and board information. So board Id or version can be retrieved, as well as sensor status or uptime. Here the whole list with it's data type and a more detailed explanation is presented.

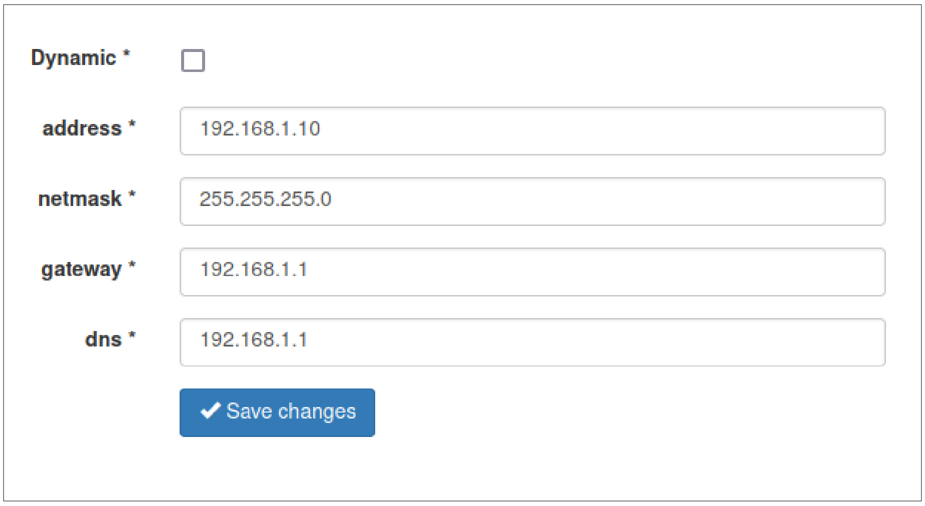
Register Address	Description	Data type	Length
0	Board ID	Int32. Hexadecimal Board's Identification	2
2	Board's Version	Decimal Multiplier	1
3	Current date and time	Int32. Unix timestamp of current date and time	2
5	Board Uptime	Int32. Amount of board's working minutes	2
7	Sensors error	Boolean. 0 .- All sensors are ok. 1 .- Some sensors are on error.	1
8	Used memory percentage	Centimal Multiplier	1
9	Board error	Boolean. 0 .- Board works fine. 1 .- There are some problems.	1

Network

Inside this menu entry the datalogger Ip address can be changed. This datalogger works with static addresses as well with dynamic ones managed by DHCP. It is highly recommended to use this last dynamic option if network properties are unknown.

If Internet connection is available the datalogger is able to sync its clock with Internet services through NTP protocol to guaranty the best accuracy on date and time.

The default network configuration the datalogger has, is a manual configuration with 192.168.1.10 as IP address. On the next image this default configuration can be checked.



Dynamic *	□
address *	192.168.1.10
netmask *	255.255.255.0
gateway *	192.168.1.1
dns *	192.168.1.1

To configure the datalogger to use DHCP the field *Dynamic* must be selected at the form.

Otherwise, if what is required is a static configuration, the later field must be unselected. When doing that, additionals fields for the static configuration will appear. Please ask the network administrator for those values. A wrong configuration could lead to an unaccessible board.

The fields to configure are:

- **Address**.- datalogger ip address. This is a unique identifier in the network that allows devices to talk with each other. Please choose a unique and valid number in the network.

- **Netmaks**.- Device network mask. A network mask represents all of the networks hosts at the same local area network.

- **Gateway**.- Ip address of the gateway, if applicable. A gateway is a network computer that acts as an entrance to the Internet or another network.

- **DNS**.- Domain name server’s address. This Internet server is used to identify domain names and translate them into ip addresses.

Server

In this section the data sending to a remote server can be enabled and configured. The server address and port must be supplied. By default this information is already filled with Sico's servers parameters.

This data sending is useful as a backup in case something serious happens to the datalogger, such a thunder burns it, drowns in water, ...

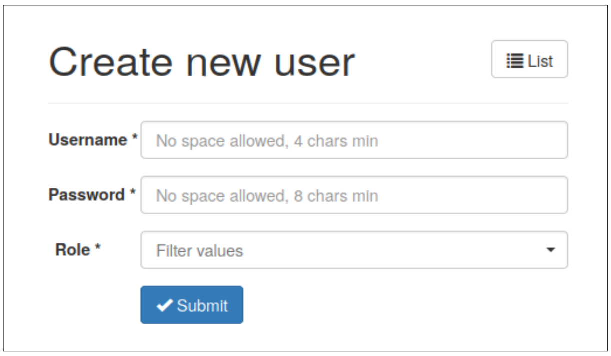
But not just as a backup, this data sending to a Sico remote server also is useful to allow further data processing and the creation of complex reports the datalogger is not able to manage.

To enable this option it is compulsory to do some configuration at the remote server by Sico. Please don't hesitate to ask us for more details at informatica@sicocv.es.

Users

As the title refers, in this section dataloger's users can be managed. Users can be added, deleted on modified. To be able to access to this menu the logged user must have administrative rights.

When creating a new user three fields must be filled, as the side image shows. The username, used to log in into the RapitLog web page. The password, used for that login. And finally, the role the user will have. The role can be either *User* or *Administrator*. If the role is set to *User*, *t*he new user will only have access to visualize and export data. In the other hand, if the role chosen for the new user is set to *Admninistrator*, this new user will have access to visualize and export data, as the plain user, but also to configure the dataloger and manage users.



Create new user

Username *	No space allowed, 4 chars min
Password *	No space allowed, 8 chars min
Role *	Filter values

Important: Do not delete the last administrative user, or otherwise it will be impossible to manage the datalogger, and therefore, to edit or add new users.

Configuration

The configuration menu entry, allows to export and import the dataloger configuration. The network configuration, created users, and board properties are not backed up when exporting the dataloger configuration.

Be aware, that importing an invalid or corrupted configuration file might leave the RapigLog on an unstable situation and therefore unable to reboot again.

Time

In this section the dataloger time-zone and time synchronization can be configured.

The zone indicates to the datalogger how to display the local time and how to do the daylight saving, if applies.

If *NTP Enabled* is set, the datalogger will synchronize its clock with the server specified in the next field.

On the other hand, if it is not checked, the datalogger time will be synchronized with the computers' clock every time the dashboard page is visited.

Advanced Configuration

In this section datalogger’s advance configuration will be described in detail. In most cases it won’t be necessary to setup any advance configuration. But there is always some special needs that requires to setup something more advance.

Serial Port

The datalogger comes with four serial communication ports. Each one of those can be plugged to probes or devices and allows to communicate either in 232 or 485 protocol. The first communication protocol only allows only one device per port, in the other hand, the second one allows more than one device per port. If the probes or devices plugged in allow it, 485 communication bus it is highly recommended to be used.

To create a new port, a physical device must be selected on which the communication bus will be created. Each physical bus has a name like `ttyS[1-4]` where the number matches the number printed on datalogger’s label. After that, bus speed has to be selected. This speed must match device’s speed, so check the device manual for that information. The last step is to select the bus type that will be used 232 or 485.

Meteos

In this section devices or probes can be created or edited. Each device is plugged to a communication port defined beforehand. A descriptive name can be setup for easier identification. This name is the one used later on through all the system to refer the device. Another field that must be filled is the device address. In case the devices operates over a 232 communication port this field won't be taken into account.

The next step is to select the device type among the available ones. Last, an enable check allows to enable / disable this device. It needs to be mentioned that a disabled device won't be taken into account by the datalogger, so all values depending on it won't be recorded at all.

Integrated devices

This chapter can be found inside the *Definitions* menu entry with the name *Device Types*. Here a list of integrated devices by the datalogger can be checked. Till now the following devices are recognized by the datalogger:

- **NUDAM**.- Devices with several analog inputs, pt100 or thermocouple.
- **EOLOS**.- Lambrecht multisensor probe.
- **ARS31.-** Luft Active road sensors.
- **NIRS31**.- Lufft Non Invasive road sensors.
- **ANACON**.- Lufft device with analog or pt100 inputs.
- **WSx**.- Lufft meteorological multisensor sensor set.
- **IRS31Pro**.- Lufft road sensor.
- **VS20 – VS2K**.- Luft visibility sensor
- **RS2-UMB**.- Lufft precipitation sensor.

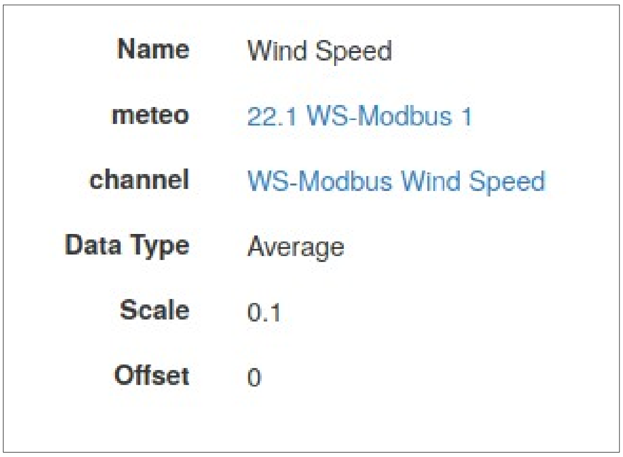
- **ModBus.-** Generic ModBus sensor. Holding and Input registers.

  Two ModBus sub-types devices have been defined. One to read *Holding Registers*, function code 0x3, and another to read *Input Registers*, function code 0x4.

  When defining a channel type for a ModBus device, the *minimum* and *maximum* values are not taken into account as the ModBus protocol does not require any conversion.

  Usually ModBus integrators present its values with a scaling factor. Often they use non ModBus standard formats like IEEE-754 -float32- or 32 bits wide integers. RapitLog is able to understand those formats but require a little configuration as described below:

  - **Scaling Factor.-** To apply an scaling factor to an address, when activating it, the *Advance Options* must be selected. Here the *Scale* option must be filled. Take into account that if the integrator provides its value with an scaling factor of 10, this means the *Scale* field must be filled with a 0.1 value. More information about this fields can be found on chapter Scale and Offset. The following image shows an example of a ModBus configuration for an address with a scaling factor of 10.

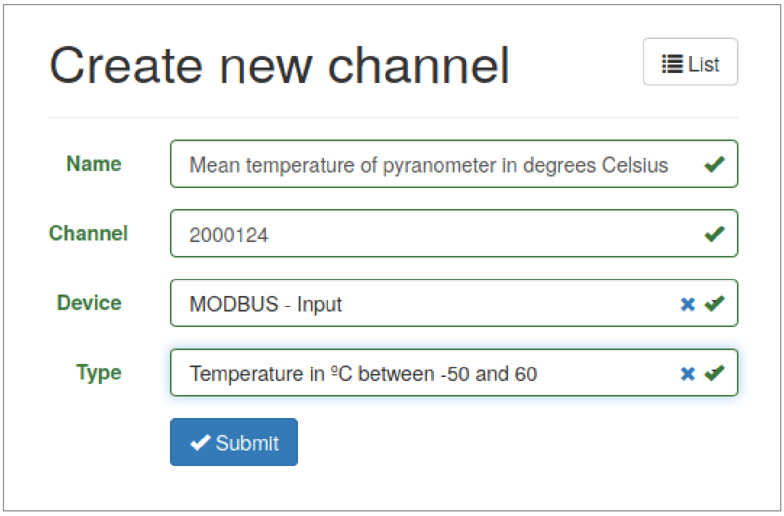


Name	Wind Speed
meteo	22.1 WS-Modbus 1
channel	WS-Modbus Wind Speed
Data Type	Average
Scale	0.1
Offset	0

  - **IEE 754.-** Also known as float32. To let RapitLog know that the required address must be interpreted as a float32 variable, when adding the channel for the ModBus device, the *Channel* field must be set with the appropriate address number plus three million (3.000.000).

  - **IEE 754 LSW.-** Same format as before, but this time the LSW comes before the MSW. When adding a channel for this type of Modbus, the *Channel* field must be set with the given address plus five million (5.000.000).

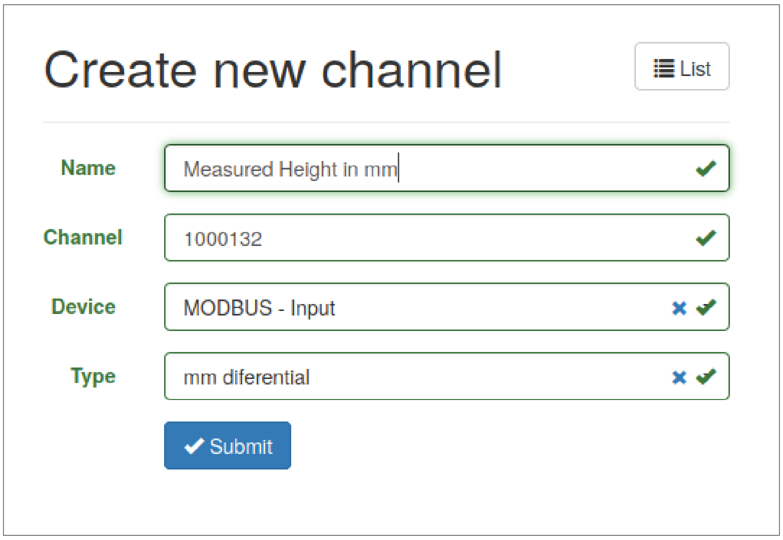
- **uInt32.-** In the same way that the float32, to let RapiLog know that the required address must be interpreted as an unsigned int32 variable, when adding the address for the ModBus device, the *Channel* field must be set with the appropriate address number plus two million (2.000.000). A configuration for an unsigned int32 variable on address 124 can be shown on the next image, where the Channel field has been set to 2000124, (124 + 2.000.000).



Create new channel

Name	Mean temperature of pyranometer in degrees Celsius
Channel	2000124
Device	MODBUS - Input
Type	Temperature in ºC between -50 and 60

- **sInt32.-** In the same way as the one before, to tell RapiLog know that the required address must be interpreted as a signed int32 variable, when adding the address for the ModBus device, the *Channel* field must be set with the appropriate address number plus four million (4.000.000).



Create new channel

Name	Measured Height in mm
Channel	1000132
Device	MODBUS - Input
Type	mm diferential

  - **sInt16.-** For the signed int of 16 bits wide. RapitLog must be configured to add one million (1.000.000) to the address for the ModBus devce in the *Channel* field. As the previous image showed.

- **SPN1.-** Sunshine Pyranometer by Deta-T Devices.

- **Lambrecht 15189.-** Lambrecht precipitation sensor series 15189

Important: At Sico we are constantly working to enlarge the recognized device list. If the device you are familiar with it is not listed here, please don't hesitate to contact us at informatica@sicocv.es to work on the possibility to integrate it. Thank you.

Note: In order for Lufft's products to work with the datalogger UMB-ASCII must be selected as communication protocol in those devices.

Channel Types

This section included inside the *Definitions* menu entry allows for the channel types to be managed. A channel type refers to the unit that is linked to a probe, as well as the probes' range, if necessarily.

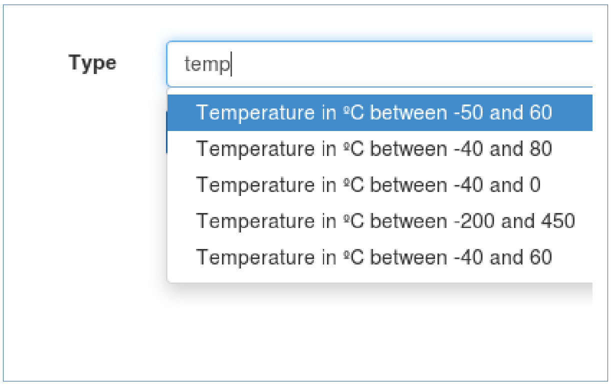
All units and ranges required by the multisensor probes are already defined. However it could happen that a probe plugged in does not have its range already defined. Let's think of an analog temperature sensor with a working range of -20 to 80 ºC that it is not defined by default. A new channel type must be created with a name, a small description, -20 as minimum value and 80 as maximum value. As well as ºC as unit. By doing it this way the analog value will be read correctly by the datalogger.

A channel type can be reused by several sensors. For this purpose when creating a new sensor, apart from creating a new channel type, an available channel must be created too. This is described in the next section.

Available Channels

This section can be found inside the *Definitions* menu entry and allows device channels or probes to be defined. The same way as the earlier chapter, all channels for the integrated devices are already setup. However sometimes it is required to add a new channel.

To add a new one a name must be provided. Devices that are able to manage several probes or sensors use to distinguish them with an index, channel or number. This identifier must be filled out at the *channel* field. The new channel needs a device type under where it should be created. The last field to choose is the channel type, this is the unit and the upper and lower values allowed. Due to its extension this dropdown has an auto filter, so it will filter the channels types as we write, as shown on the side image.



Type: temp

Temperature in ºC between -50 and 60
Temperature in ºC between -40 and 80
Temperature in ºC between -40 and 0
Temperature in ºC between -200 and 450
Temperature in ºC between -40 and 60

It must be said that by creating a new available channel does not mean the datalogger is going to read and store its value. As the name suggest this channel is available for the datalogger but not active. In this sections definitions are created, at the *Active Channel* section is where a channel can be *activated* for the datalogger to read its value.

Advance configuration example

Here a practical advance example is described.

Lets say there are two analog pressure probes with a working range between 0 and 300 psi. Those probes record the pressure on two different presses. Lets also say that what it is really wanted is to know the difference between them and also this information it is required in bares, not in psi.

The next steps shows the procedure to achieve such an advance configuration:

1. Crete a *channel type* with a minimum value of 0 and a maximum one of 300. This will automatically convert the 4 – 20 mA signal to 0 – 300 psi.

2. Create two *Available channels*. Select *Nudam* as device type. Introduce 0 as a channel for the first probe and 1 for the second one. This is because the first and the second Nudam inputs is where the analog probes are plugged in.

3. Add a new *Meteo* and add the communication port where it is plugged and the meteo type. Lets suppose the Nudam module its plugged into the first communication port. So the first port and Nudam as type must be selected. At the address field the Nudam address used when configuring it, must be provided here.

4. Now everything its defined. It is time to activate the defined channels. To do so at the *Active Channels* section two new channels must be created selecting the *Available Channels* defined before. To be able to do the unit conversion it is required to activate the advance options and set the conversion. As it is well known that a psi equals to 0,06895 bars, this value must be filled at the *Scale* field to make the datalogger do the unit conversion.

5. At this point the datalogger is able to read and store the two pressures. Now lets create what it is really needed, the pressure difference. To do so a calc channel must be added with a name and a comprehensive description.

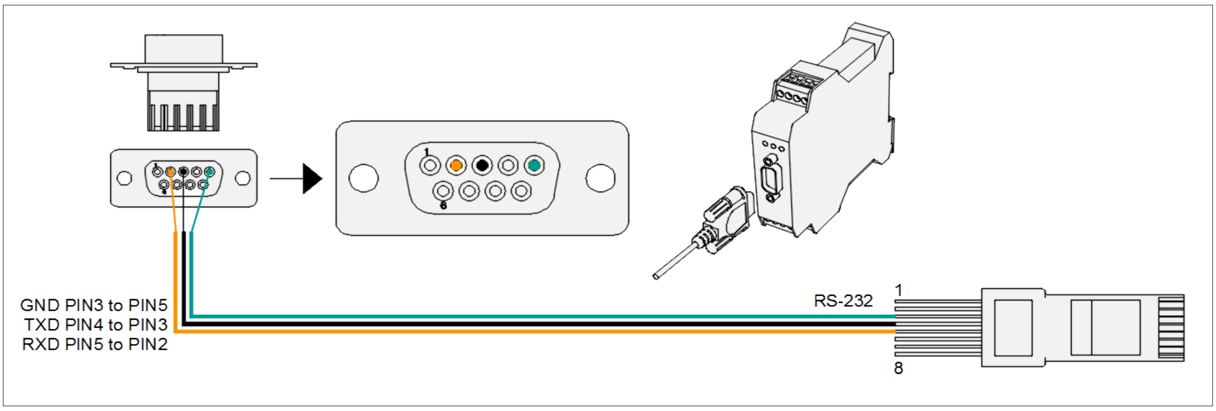
6. To finish the calc channel configuration it must be edited. Adding the first pressure channel as type *Add* and the second one as type *Subtract* the datalogger will calculate the pressure difference. Now everything is properly setup.

Pinouts Diagrams

RapitLog to DB9 (232)

To be able to connect the datalogger to a DB9 232 serial interface a specific RapitLog to DB9 cable must be used. In case this cable is not available there is always the option to build it. To do so an RJ45 terminal must be wired to a DB9 according the following table and as shown on the next diagram.

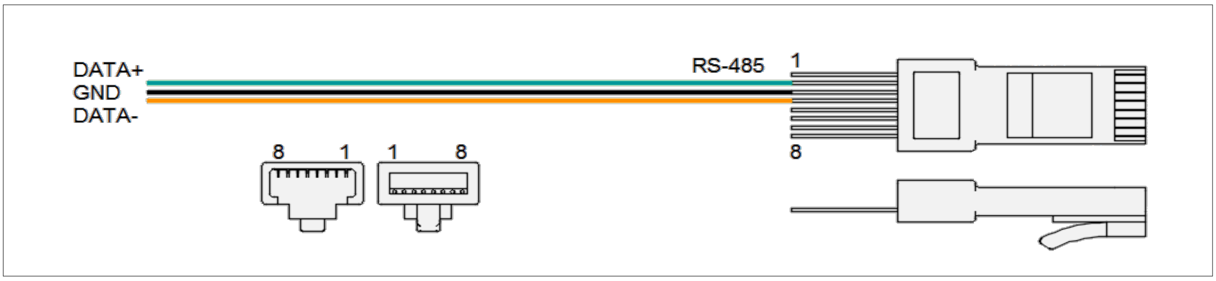
Name	**RJ45**	**DB9**
GND	Pin 3	Pin 5
TX	Pin 4	Pin 3
RX	Pin 5	Pin 2



RapitLog to terminal blocks (485)

To be able to connect the datalogger to the 485 bus a special RapitLog 485 cable must be used. In case of not having it, there is always the option to build id. To do so an RJ45 terminal must be wired according the following table and as shown on the next diagram.

Name	RJ45
Data +	Pin 2
Data -	Pin 4
GND	Pin 3

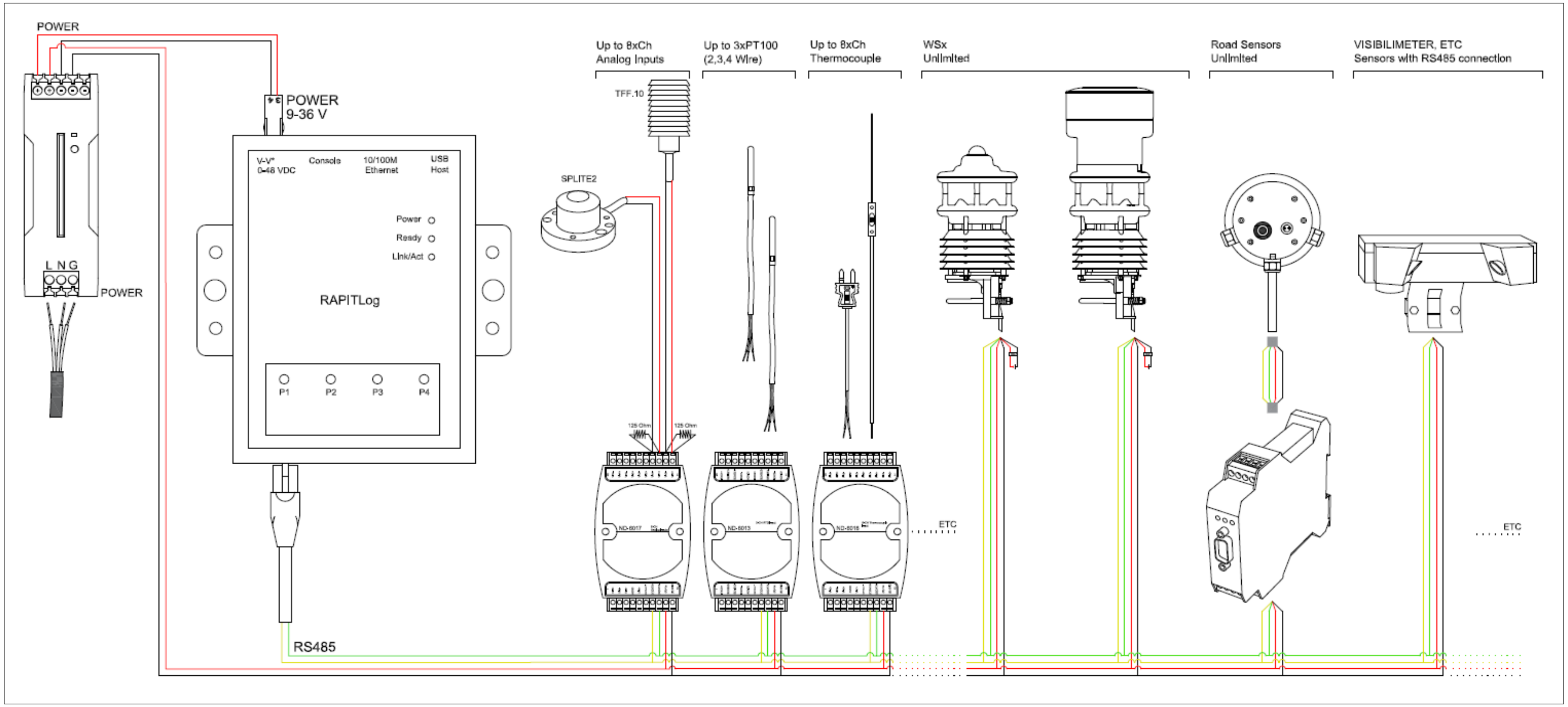


Appendix A

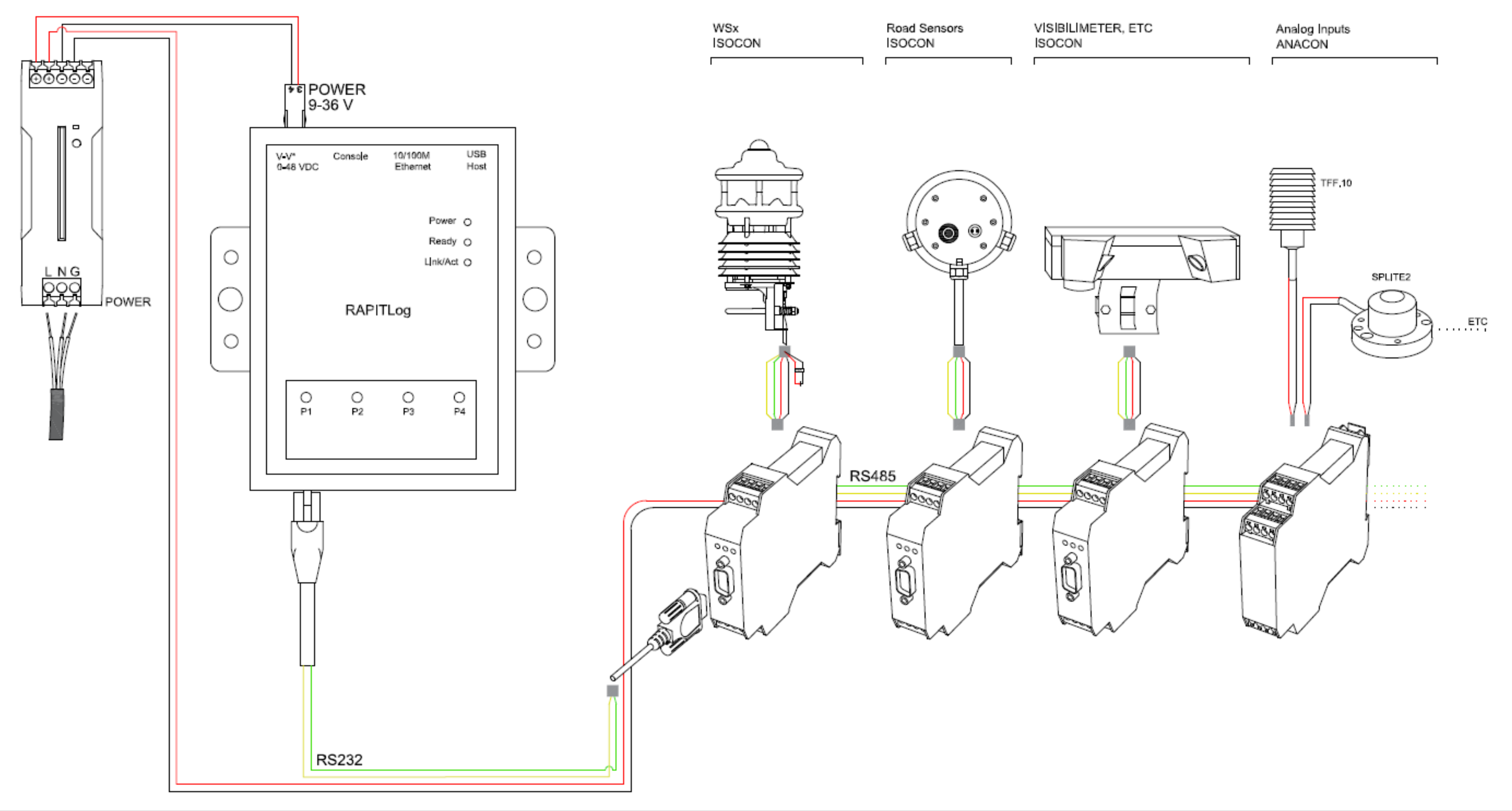
In this appendix two wiring diagrams are shown. The first one is an example on how to connect the datalogger to several devices. Analog probes, Lufft's WSx, IRS31 etc.

The second one shows how to connect the datalogger to an existing system with the minimum changes on the wiring to make the installation as easy as possible. Note that this setup requires 232 communication mode to be selected.

RapitLog wiring example



RapitLog wiring example to an existing system



Appendix B

A list of available accessories are listed in this appendix

RapitLog Scada

The RapitLog Scada it’s a 15,6” HD display that allows for live, non stop data visualization from RapitLog. It has both Ethernet and wireless communications for easer installation. With its multiple set-up modes, standing on a table, hanging from ceiling or attached to a wall fits in any place. The screen its tailored for every client so the user only sees what he/she really needs.