1x Hardware Manual

Release 4.8/1.3

Embention Sistemas Inteligentes, S.A.

2025-12-10

Contents

Scope of Changes

- Version 1.0
  - Added:
    - Pegasus Serie servo integration example
    - Pegasus CAN servo integration example
  - Changed:
    - Hours of operation for Target Drone/Loitering Munition Operation in Limited Operation section
- Version 1.1
  - Fixed:
    - Dynamic pressure sensor resolution value
- Version 1.2
  - Added:
    - GPIO Wiring for External Loads integration example
- Version 1.3
  - Added:
    - Operate in a Limited Operation Environment - Quick start section
    - UDP Failed Connection - Troubleshooting section

Introduction

Veronte Autopilot 1x

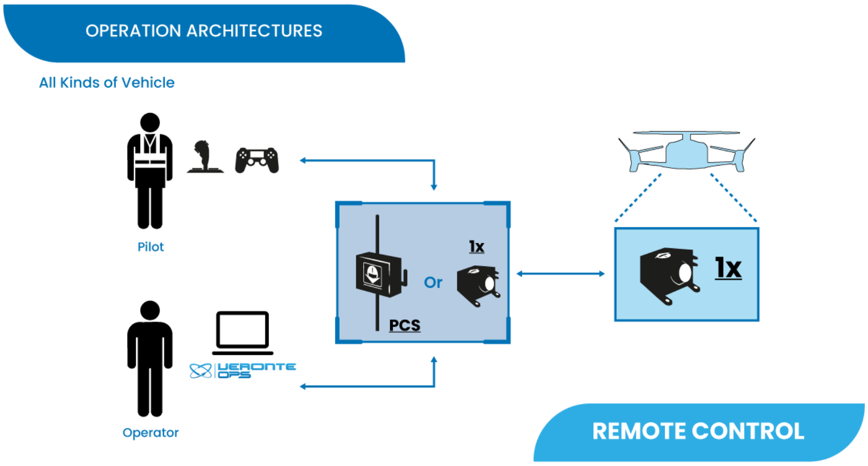
Veronte Autopilot 1x is a sensor-redundant control system designed to control any autonomous vehicle, either aircraft such as multirotors, helicopters, planes, VTOL, blimps, etc., as well as ground vehicles, surface vehicles or many others.

Veronte Autopilot 1x hardware embeds a redundant state-of-the-art suite of sensors, together with LOS and BLOS M2M datalink radio and a DAA module based on Remote ID or ADS-B, all with reduced size and weight.

Veronte Autopilot software tools are specifically designed for the operation and configuration of the **Veronte Autopilot**. **Veronte Ops** is the software employed to operate the autopilot from a user-configurable interface and **1x PDI Builder** permits configuring the autopilot and adapting it to the specific needs of the project.

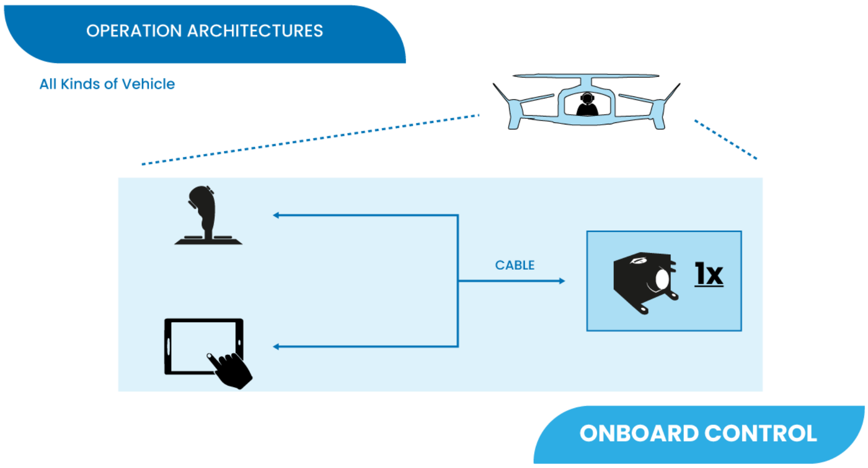
Typical Drone Setup

Veronte Autopilot is compatible with both LOS and BLOS communications for the remote control of autonomous vehicles. Physical or virtual sticks can be used for taking manual or assisted manual control at any time during the operation.



Typical eVTOL Setup

An onboard display and joystick interface are available for manned eVTOL and aerotaxi applications. Autonomous flight modes and fly-by-ware control can be configured according to the level of autonomy required.



Main Features

- **Highly configurable**: Fully user-configurable; payload, vehicle layout, control phases, control channels, etc.
- **Custom routines**: User-selectable automatic actions, activated on system events or periodically.
  - **Actions**: Phase change, activate payload, move servo, go to, onboard log, parachute release, etc.
  - **Events**: Waypoint arrival, inside/outside polygon, alarm, variable range, button, etc.

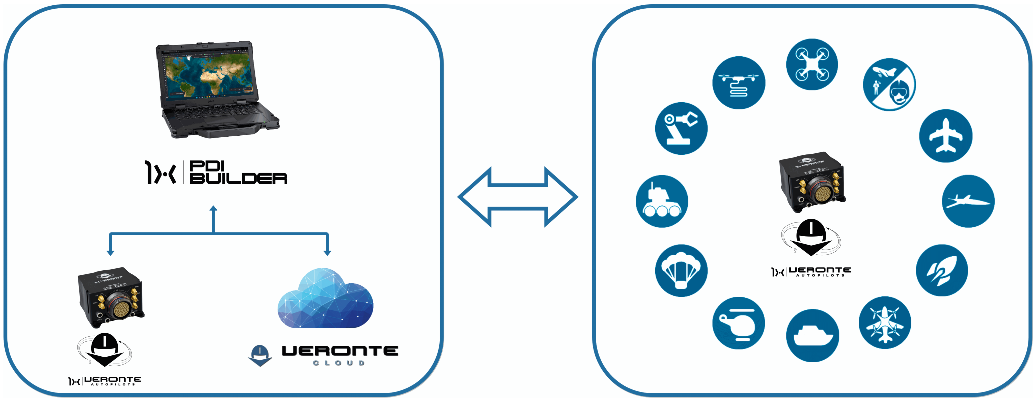
- **Telemetry & log**: Customizable telemetry and onboard log compatible with internal or external datalink modules, all with user-defined variables and recording frequency.
- **External sensor**: Support for external sensor connections: magnetometer, radar, LIDAR, RPM, temperature, fuel level, battery level, weather, etc.
- **Payload & Peripheral**: Transponder, secondary radios, satcom transceivers, camera gimbals, motor drivers, photo cameras, flares, parachute release systems, tracking antennas, pass through RS232, RS485 & CAN tunnel, etc.
- **Redundant Configurations**: **Veronte Autopilot 4x** is available for applications requiring redundancy and **Veronte Autopilot DRx** offers distributed redundancy architectures.

Quick Start

This user manual covers the mechanical and electrical assembly.

The software user manual explains how to configure and use the **Veronte Autopilot 1x**.

Veronte Autopilot 1x is the main element in our Flight Control System for UAS.

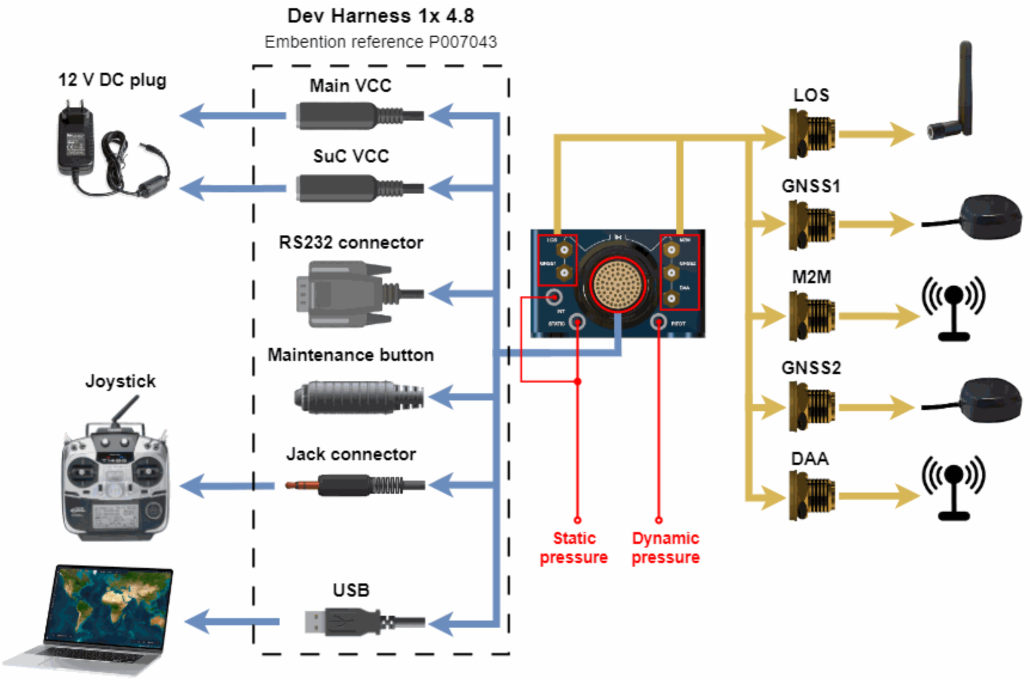


System Overview

Veronte Autopilot 1x contains all electronics and sensors required to properly execute all the UAV functions. A Veronte-based FCS contains the following elements:

- A **Veronte Autopilot 1x** installed in a vehicle to control. This autopilot executes GNC algorithms in real time in order to accomplish the planned mission and handle the payload.
- **Veronte Ops** - Software dedicated to mission planning, configuration and operation. It allows the user to monitor connected UAS in real time, to interact with them and to replay previous missions for post-flight analysis.
- An Autopilot 1x **GND unit** or **PCS** linked between **Veronte Ops** and **Veronte Autopilot 1x**. They support manual and arcade modes with conventional joysticks.

Basic Connection Diagram



For further information on the Dev Harness 1x 4.8 connectors, refer to the Dev Harness 1x 4.8 - Hardware Installation section of the present manual.

Warnings

- This user manual includes references to manuals for software applications. Select your software version to read them.
- Power out of range can cause irreversible damage to the system. Please read carefully the manual before powering the system.
- Each I/O pin withstands a maximum current of 1.65 mA. See pinout for more information.
- If the unit has an ADS-B, Remote ID and/or BLOS module activated, users must **NOT power on a Veronte Autopilot 1x** without **a suitable antenna** or **50 Ω load** connected to the SSMA port.

> ⚠ **Caution**
>
> Do not connect this type of termination to the ports for GNSS signals, as it may damage the module.

> **Danger**
>
> Disobeying this warning may damage the Autopilot 1x unit.

Limited Operation

Veronte Autopilot units are delivered with limited-operation installed and must be updated for enabling unlimited autonomous flight capabilities. **Operation limits** in Veronte Autopilot units can be **checked and unlocked in** Veronte Ops. For more information about this, see Platform license - Platform section of **Veronte Ops** user manual.

The different operating options available are explained below:

- **Non-Limited-Operation**: Allows fully autonomous operation with no time or distance limitation.
- **Limited-Operation**:
  - Allows fully autonomous flight performance in **LOS (500 m)** with no time limitations.
  - For **BLOS operations (>500 m)**, there is a limitation of 30min autonomous flights. After 30min from leaving LOS (500m), only internal navigation is permited and the autopilot can not estimate its position nor fly autonomously. Past this time limit, no changes of navigation source are allowed.

    > ⚠ **Caution**
    >
    > External navigation methods such as VectorNav can no longer be used.

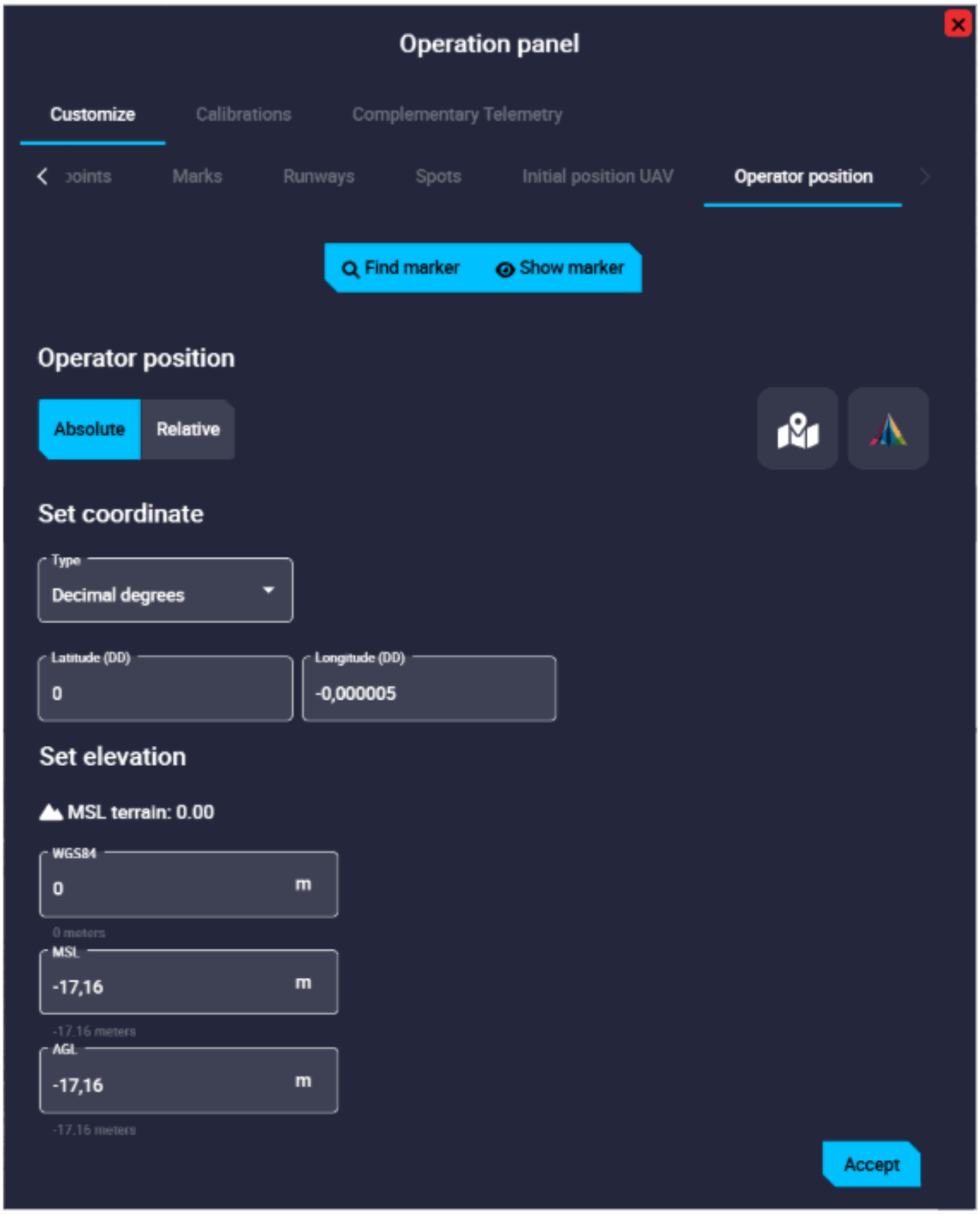
- **Target Drone/Loitering Munition Operation**: Allows fully autonomous flight performance with no time or distance limitation. This is restricted to 25h operation from the first startup. Once the operation time has expired the unit will not be able to restart.

If you have any questions regarding the operation capabilities and limitations please contact us at sales@embention.com.

Operate in a Limited Operation Environment

When operating under a limited license condition, users must adhere to specific time and distance limitations.

From Veronte Ops app, users can select an **Operator Position**, which is the reference point from which the distance allowed by the operating license is calculated (in this case 500 m radius).



Veronte Ops - Operator Position panel

> ⚠ **Caution**
>
> If the Operator Position is not set, the 30 minutes timer will start once the autopilot gains a Position Fix.

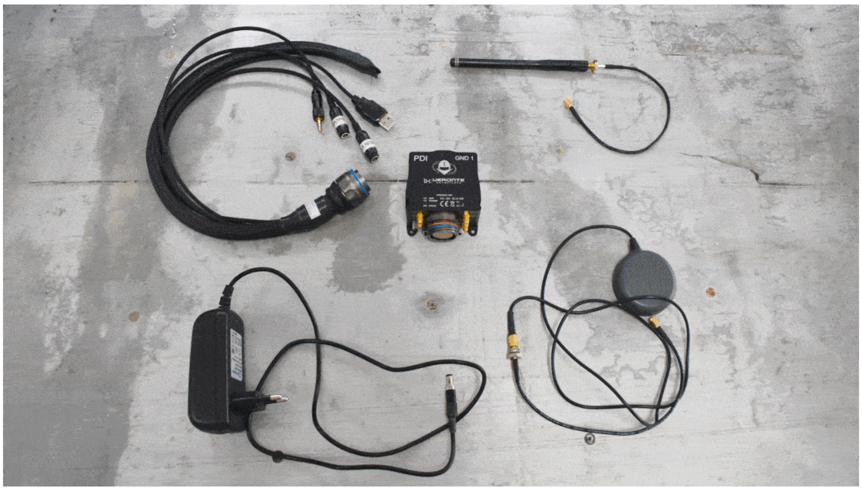
The operator is permitted to command a flight beyond the 500-meter radius, subject to a time limit. Upon exiting this 500-meter zone, a 30-minute timer activates. The timer pauses and resets when the platform re-enters the 500-meter radius.

To ensure safe operation, it is recommended to implement an automated Return to Home command. This command should activate if the platform flies beyond the 500-meter radius for more than a customizable limit of 15–20 minutes. The 30-minute timer will reset upon the platform's re-entry into the 500-meter zone.

Basic Connection for Operation

The steps described below cover the basic connection of a **ground unit** necessary for operation:

1. Connect the GNSS antenna to the **GNSS 1 port**:



Basic connection - Step 1

2. Connect the RF antenna to the **LOS port**:

Basic connection - Step 2

3. Connect the **autopilot harness** and **power it** using the power supply:

Basic connection - Step 3

4. Connect the **harness USB** to the computer and **configure Veronte Link** to detect the Autopilot 1x:

> ⓘ **Note**
>
> For Veronte Link to detect a Veronte device, the corresponding port must be properly configured. For further information regarding Veronte Link connections, please refer to the Serial connection - Integration examples section of **Veronte Link** user manual.

Basic connection - Step 4

Technical

Features

- **Communications**
  - 2 x isolated CAN buses
  - 1 x I2C bus
  - 1 x USB port
  - 1 x RS 232 bus
  - 1 x RS 485 full duplex bus
  - 1 x UART bus
  - Over USB, RS 485 or RS 232 firmware update
  - 1 x LOS module
  - 1 x BLOS module
- **Input / Output**
  - 16 x configurable input / output signals
  - 5 x analog input signals 0 - 3.3 V
  - 1 x EQEP bus
- **Power**
  - 6.5 - 36 V DC required for input power supply
  - 1 x output power with 3.3 V, up to 100 mA
  - 1 x output power with 5 V, up to 100 mA
- **Protection**
  - EMI shield
  - Against inrush current for connecting power supply

> **Note**
>
> The number of communication ports and signals can be increased with Veronte CEX or Veronte MEX.

Variants

The **Veronte Autopilot 1x** has 3 variants:

- **W/O DAA**
- **With Remote ID**
- **With ADS-B**

ADS-B module

Frequency band	1090 MHz
Current consumption	Averaged 140 mA
Sensitivity	-80 dBm
RF output power	Configurable +30 dBm (1W), +27 dBm (0.5W), +24 dBm (0.25W)
ESD protection	All lines protected
MAVLink (baud)	115200 bps
AERO (baud)	115200 bps (AT commands)

RemoteID module

Frequency	WiFi & Bluetooth bands
Developed according to	RIN 2120-AL31 Remote identification of Unmanned Aircraft FAA Standard
Parameters	Aircraft ID, position, altitude, and time mark
Compatibility	FAA Remote ID Scanner App

Sensor Specifications

Accelerometers (3-axis each one)			
Specification	**IMU 1**	**IMU 2**	**IMU 3**
Range	16 g	24 g	8 g
Maximum shock	20,000 g for 0.2 ms	10,000 g/ms	14,700 m/sec^2
Sensitivity	16,393 LSB/(m/sec^2)	10,920 LSB/(m/sec^2)	26,756,268 LSB/(m/sec^2)
Update Time	1 ms		
Error	3 mg (RMS noise)	190 Z axis 160 X & Y axis µg/Hz (noise density)	0.000167 X & Y axis 0.000243 Z axis (m/sec^2 / Hz) (noise density)
Offset	0 mg	± 20 mg	0.0196 m/sec^2

Gyroscopes (3-axis each one)			
Specification	**IMU 1**	**IMU 2**	**IMU 3**
Range			

Gyroscopes (3-axis each one)			
Specification	**IMU 1**	**IMU 2**	**IMU 3**
	125 to 2,000 °/sec	125 to 2,000 °/sec	2,000 °/sec
Sensitivity	228 to 14.2 LSB/°/sec	262 to 16 LSB/°/sec	655,360 to 10 LSB/°/sec
Update Time	1 ms		
RMS noise	0.075 °/sec	0.1 °/sec	0.152 °/sec
Offset	0 °/sec	±1 °/sec	0.14 X & Z axis 1.4 Y axis °/sec

Magnetometers			
Specification	**Magnetometer 0**	**Magnetometer 1**	**Magnetometer 2**
Range	4 G	8 G	11 G
Sensitivity	6,842 to 1,711 LSB/G	4,096 LSB/G	0.13 mG
Update Time	8.3 ms	12.5 ms	
RMS Noise		0.4 mG	0.3 mG

Magnetometers			
Specification	**Magnetometer 0**	**Magnetometer 1**	**Magnetometer 2**
	3.2 X & Y axis 4.1 Z axis mG		
Offset	0 G	0 G	0 G

Static Pressure		
Specification	**Sensor 1 (STATIC port)**	**Sensor 2 (INT port)**
Range	1,000 - 120,000 Pa	30,000 - 120,000 Pa
Band Error	500 Pa	200 Pa
Resolution	1.2 to 6.5 Pa	0.5 Pa
Update Time	20 ms	31.3 ms
RMS Noise	6.5 Pa	0.35 Pa

Dynamic Pressure Sensor	
Specification	**Pitot**
Range	3 Pa (5 kt / 8 km/h sea level) to 6,900 Pa (206 kt / 382 km/h sea level)
Band Error	140 Pa

Dynamic Pressure Sensor	
Specification	**Pitot**
Resolution	0.42 Pa
Update Time	20 ms
Bias	±7 Pa

GNSS Receivers	
Specification	**GNSS 1 & GNSS 2**
Constellations	BeiDou, Galileo, GLONASS, GPS / QZSS
Concurrent GNSS	4
Bands	L2OF, L2C, E1 B/C, B2I, E5b, L1 C/A, L1OF, B1I
Position Accuracy RTK	0.01 m + 1 ppm CEP
Update Rate RTK	Up to 20 Hz
Anti-jamming	Active CW detection and removal, Onboard bandpass filter
Anti-spoofing	Advanced anti-spoofing algorithms

<table>
<tr><th colspan="4">GNSS Receivers</th></tr>
<tr><th>Specification</th><th colspan="3">GNSS 1 & GNSS 2</th></tr>
<tr><td>Advanced Functions</td><td colspan="3">• Moving base capable
• GNSS-based attitude estimation</td></tr>
<tr><td>Update Time</td><td colspan="3">250 ms</td></tr>
<tr><td rowspan="3">Location Accuracy</td><td colspan="2">Vertical</td><td>0.01 m</td></tr>
<tr><td rowspan="2">Horizontal</td><td>SBAS</td><td>1 m</td></tr>
<tr><td>RTK</td><td>0.01 m</td></tr>
<tr><td>Velocity Accuracy</td><td colspan="3">0.05 m/s</td></tr>
</table>

Temperature

Device	Resolution	Bias
IMU 1	256 LSB/°C (16 bit)	15°C
IMU 2	8 LSB/°C	1°C
IMU 3	10 LSB/°C	5°C
MPU	-	15°C
Magnetometer 0	8 LSB/°C	-
Magnetometer 1	1.25 LSB/°C	-

Temperature		
Device	**Resolution**	**Bias**
Static pressure 1	0.01 °C	-
Static pressure 2	0.01 °C	-

> **Note**
>
> An external pressure sensor is required to measure below -20 °C.

Embedded Communications

LOS module

RF baudrate	115200 baud
Transmission power	19 dBm
Receiver sensitivity	-103 dBm
Frequency band	ISM 2.4 GHz
LOS range	Depends on the antenna employed and on the user's setup

> **Note**
>
> External modules can be used.

BLOS module

RF baudrate	115200 baud
Receiver sensitivity	-111 dBm
Frequency band	800 MHz, 850 MHz, 900 MHz, 1.8 GHz, 1.9 GHz, 2.1 GHz
Network	UMTS/HSPA+, GSM/(E)GPRS
eSIM	Included **Note** Its activation is optional

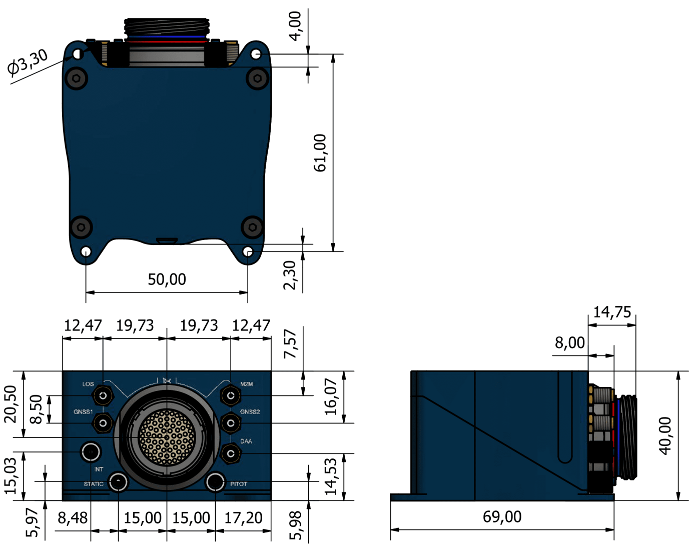
> **Note**
>
> External BLOS modules, such as satellites, can be used.

Mechanical and Electrical Specifications

Variable	**Value**
Weight	W/O DAA variant: 198 g With Remote ID or ADS-B: 210 g With Damping System: + 60 g
Temperature range	-40 to 65 ºC
Protection rating	IP67

Variable	Value
Maximum acceleration	32 g
Voltage input	6.5 V to 36 V
Power consumption	2.2 W in maintenance mode
	2.2 W in normal mode with CPU at 98 %, internal LOS module off and BLOS module off
	2.6 W in normal mode with CPU at 98 %, internal LOS module on and BLOS module on

Dimensions

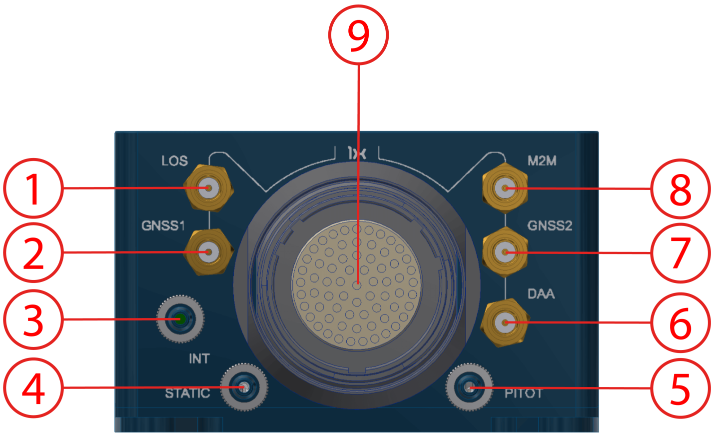


Veronte Autopilot 1x dimensions (mm)

M3 screws are recommended for mounting. In saline environments such as coastal and oceanic, the screw material must be stainless steel.

Interfaces

Connector Layout



Connectors

Index	Connector
1	LOS SSMA connector
2	GNSS1 SSMA connector
3	Static pressure port (Int. D. 2.5 mm x Out. D. 4 mm) for static pressure sensor 2
4	Static pressure port (Int. D. 2.5 mm x Out. D. 4 mm) for static pressure sensor 1
5	Dynamic pressure port (Int. D. 2.5 mm x Out. D. 4 mm)
6	ADS-B or remote ID SSMA connector **Warning** When using ADS-B or remote ID, there must be an adequate antenna or load connection to the DAA SMA.

Index	Connector
7	GNSS2 SSMA connector
8	M2M SSMA connector **Warning** If the BLOS module is enabled, a suitable antenna must be connected to this SSMA port. The 4G Antenna with the Embention reference **P000112** is recommended.
9	68-pin connector

Both static pressure ports must be used for sensor redundancy (Y tubing connection is strongly recommended).

> ⚠ **Warning**
>
> The static pressure port 4 is always used by **Autopilot 1x** to calculate speed (using the difference of pressure between ports 4 and 5), no matter which sensor is selected in configuration.

Mating Connectors

Index	Autopilot 1x Connector	Mating Connector
1	RF antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
2, 7		SSMA male Plug, low-loss cable is

Index	Autopilot 1x Connector	Mating Connector
	GNSS antenna (SSMA Jack Female)	recommended. Active Antenna GNSS: • Gain min 15dB (to compensate signal loss in RF Cable) • Gain max 50 dB • Maximum noise figure 1.5dB • Power supply 3.3V • Max current 20 mA
8	M2M antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
6	ADS-B or remote ID (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
9	Connector HEW.LM. 368.XLNP	Mating connector: FGW.LM.368.XLCT (Embention reference **P005550**) Mating harnesses available on demand: • Dev Harness 1x 4.8 (Embention reference **P007043**)

Index	Autopilot 1x Connector	Mating Connector
		• Veronte Harness Blue 68P (Embention reference **P001114**) • HIL Harness 1x 4.8 (Embention reference **P007740**)

Hardware Installation

Mechanical

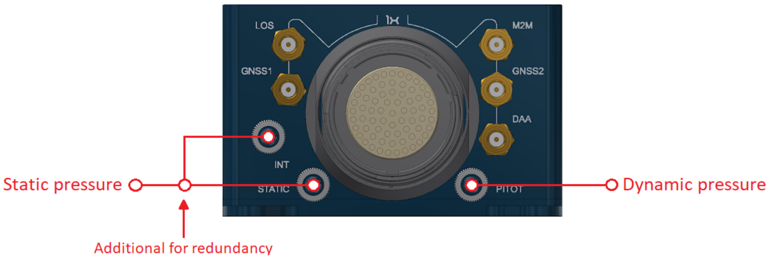
Veronte Autopilot 1x is manufactured using an anodized aluminium enclosure with enhanced EMI shielding and IP protection. A high reliability connector is also provided in this version. The total weight of W/O DAA variant is 198 g and it is 210 g for versions with Remote ID or ADS-B.

Pressure lines

Veronte Autopilot 1x has three pressure input lines, two for static pressure to determine the absolute pressure and one for pitot in order to determine the dynamic pressure.

For the fittings it is recommended to use a polyurethane tube of 2.5 mm inner diameter and 4 mm outer diameter.

- **Pressure Intake**
  - Pressure intakes must be located in order to prevent clogging.
  - Never install pressure intakes on the propeller flow.
  - Design pressure tubing path in order to avoid tube constriction.
- **Static Pressure**
  - It is not recommended to use inside fuselage pressure if it is not properly vented.
- **Pitot Tube**
  - Pitot tube must be installed facing the airflow.
  - It is recommended to install it near the aircraft's x axis in order to avoid false measures during manoeuvres.
  - For low-speed aircraft it is recommended at least 6.3mm tubes to prevent any rain obstruction.



> ⓘ **Note**
>
> In case of not using an input air connector, it is recommended to remove its corresponding nut. Vibrations may move and damage intake connectors with a nut that is not fixed with a tube.

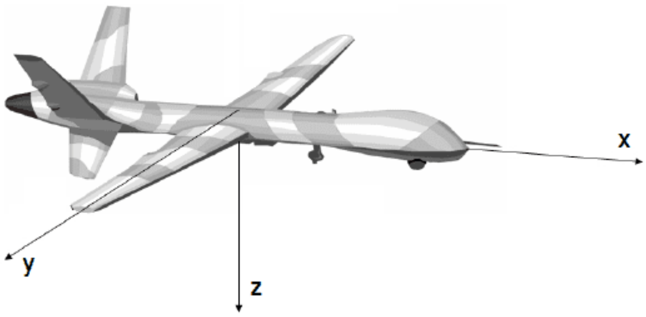
Location

The location of **Veronte Autopilot 1x** has no restrictions. You only need to configure its relative position with respect to the centre of mass of the aircraft and the GNSS antenna. The configuration of the location of **Veronte Autopilot 1x** can be easily configured using 1x PDI Builder.

Orientation

The orientation of **Veronte Autopilot 1x** has no restrictions either. You only need to configure axes with respect to the aircraft body axes by means of a rotation matrix or a set of correspondences between axes. The configuration of the orientation can be easily configured using **1x PDI Builder**.

Axes are printed on the **Autopilot 1x** box. Aircraft coordinates are defined by the standard aeronautical conventions (see image below).



Aircraft Coordinates (Standard Aeronautical Convention)

Vibration Isolation

Although **Veronte Autopilot 1x** rejects noise and high-frequency vibration modes with electronic filters, there may be situations where external isolation is needed.

Autopilot 1x can be mounted in different ways in order to reject the airframe vibration, but it is recommend to use the Damping System designed for that porpuse. It covers a wide frequency range of different aircraft types.

> ⓘ **Note**
>
> The user should take into account that wiring should be loose enough so that vibrations are not transmitted to **Autopilot 1x**.

Damping System

Embention offers the **Damping System** as a solution to isolate **Veronte Autopilot 1x** from vibrations.

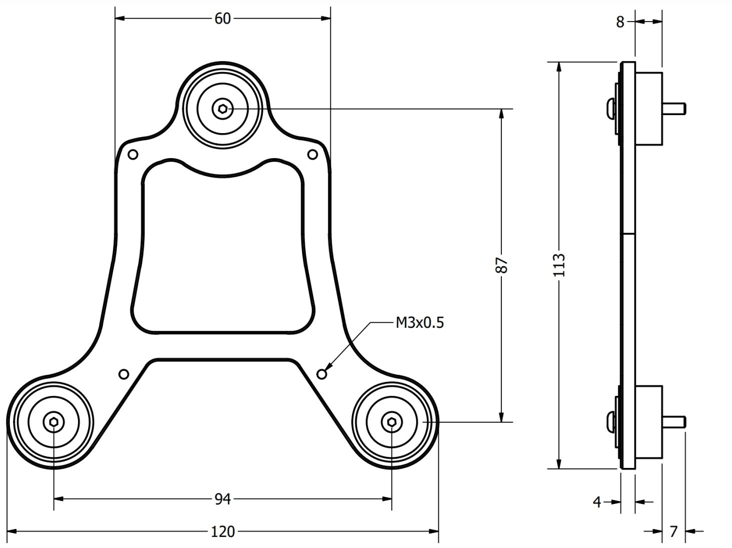
> **Important**
>
> Only effective with **Autopilot 1x** in horizontal position.

This damping system weighs **60 g**.

Damping System

> ⚠ **Warning**
>
> The **Damping System** is designed for hardware version **4.8** of **Autopilot 1x**.

Dimensions

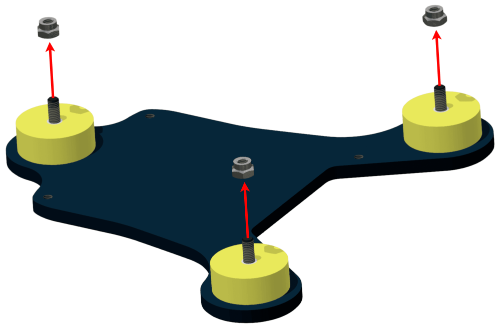


Damping system dimensions (mm)

Assembly steps

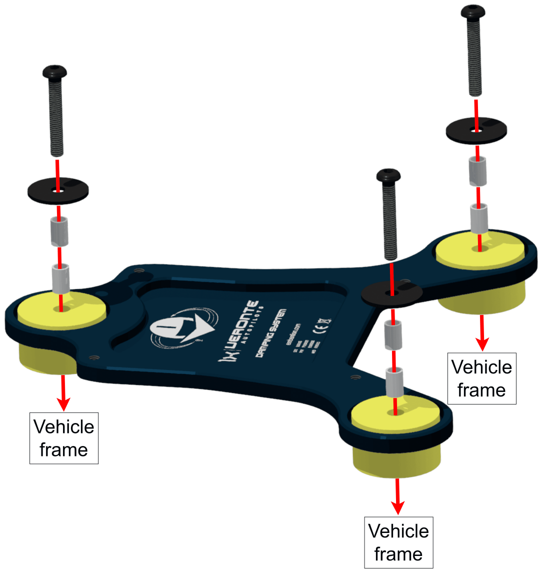
To assembly the Damping System into a vehicle with an **Autopilot 1x**, read the following steps.

1. Remove the three nuts located under the platform.



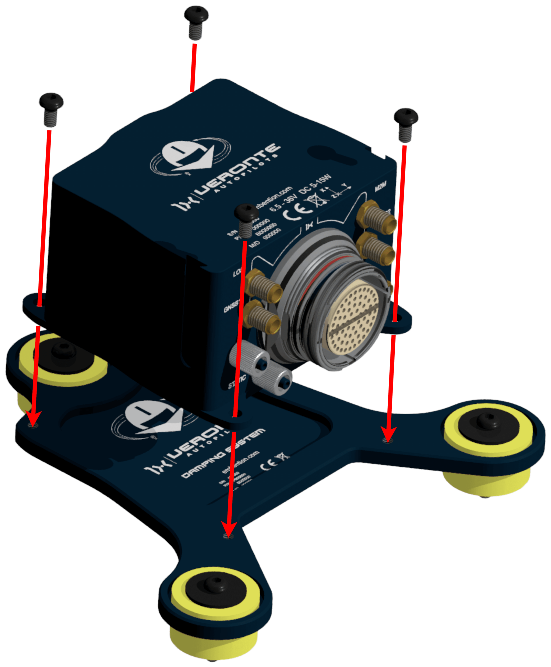
Step 1

2. Screw the platform on the aircraft frame. The included screws have M3.

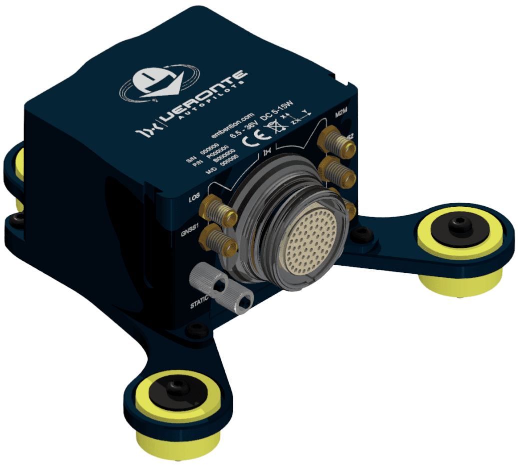


Step 2

3. Screw the **Autopilot 1x** on the **Damping system**.



Step 3

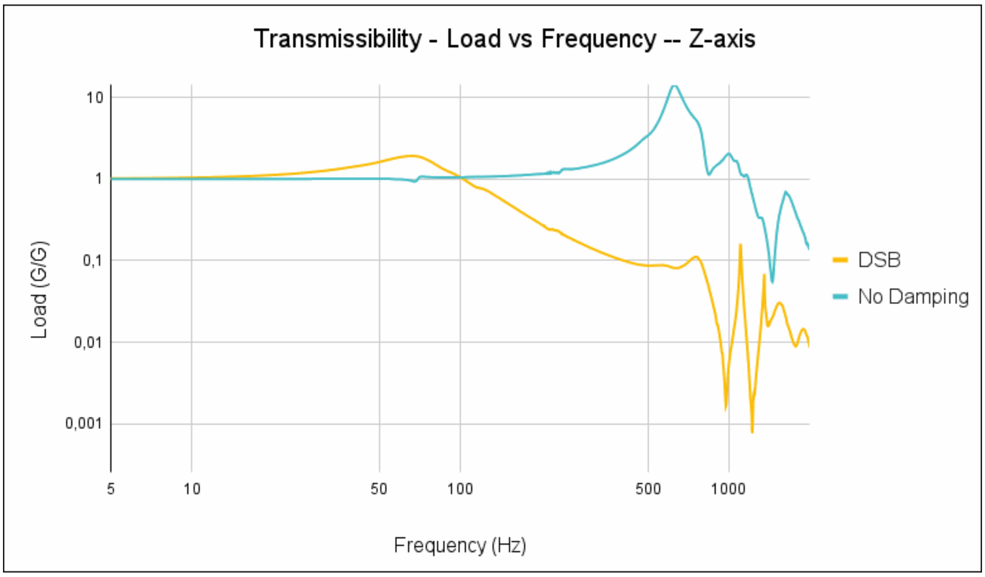


Result

Vibration analysis

- **Transmissibility (Z-axis)**

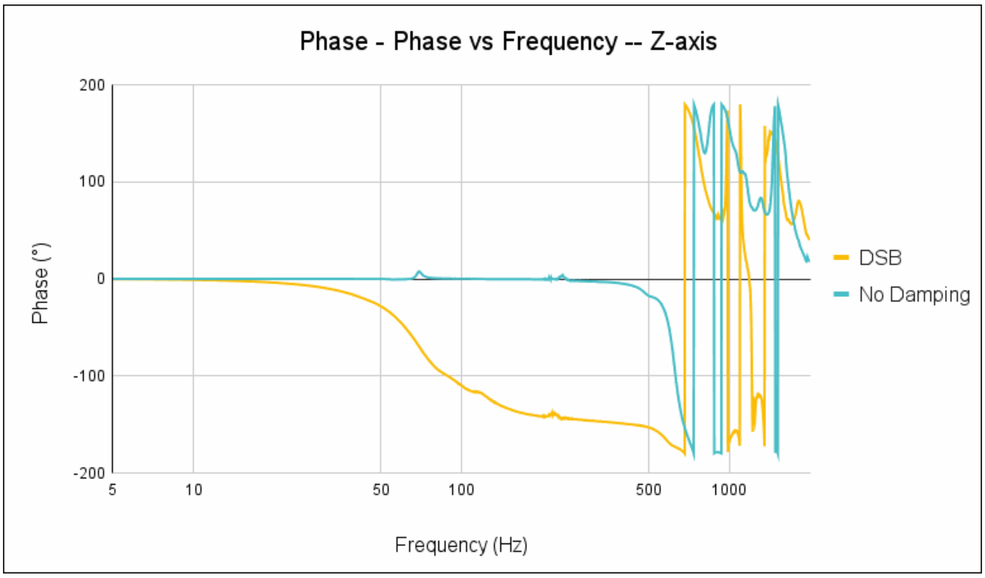
  Transmissibility graph, analysis of vibrations received by **Autopilot 1x**, with and without Damping System:



Transmissibility graph

- **Phase (Z-axis)**

  Phase graph, analysis of whether vibrations measured by **Autopilot 1x** are in phase or not, with and without Damping System:



Phase graph

Antenna Integration

The system uses different kinds of antennas to operate that must be installed on the airframe. Here you can find some advice for obtaining the best performance and for avoiding antenna interferences.

- **Antenna Installation**
  - Maximize separation between antennas as much as possible.
  - Keep them far away from alternators or other interference generators.
  - Always isolate antenna ground panel from the aircraft structure.
  - Make sure the antenna is securely mounted.
  - Always use high-quality RF wires minimising the wire length.
  - Always follow the antenna manufacturer manual.
  - SSMA connections shall be tightened applying 1Nm of torque
  - For all-weather aircraft, insert SSMA lightning protectors.
- **GNSS Antenna**
  - Antenna top side must point the sky.
  - Install it on a top surface with direct sky view.
  - Never place metallic / carbon parts or wires above the antenna.
  - It is recommended to install it on a small ground plane.
  - For all-weather aircraft, insert SSMA lightning protectors.

- **Recommended specifications for GNSS antennas**

Specifications	Range
Antenna frequency L1	1561.098 MHz to 1602 MHz
Antenna frequency L2	1207.14 MHz to 1246 MHz
Amplifier gain	17 dB to 35 dB
Out-of-band rejection	40 dB **Note** Higher values are preferable. 30 dB is considered the minimum acceptable value.
Polarization	RHCP (Right-Hand Circular Polarization)
Minimum supply voltage	2.7 V to 3.3 V
Maximum supply current	50 mA

Electrical

Power

Veronte Autopilot 1x can use unregulated DC **(6.5V to 36V)**. Pins used for power and ground are the same for both Ground and Air configurations.

LiPo batteries between 2S and 8S can be used without regulation needs. Remaining battery level can be controlled by the internal voltage sensor and by configuring the voltage warnings by software.

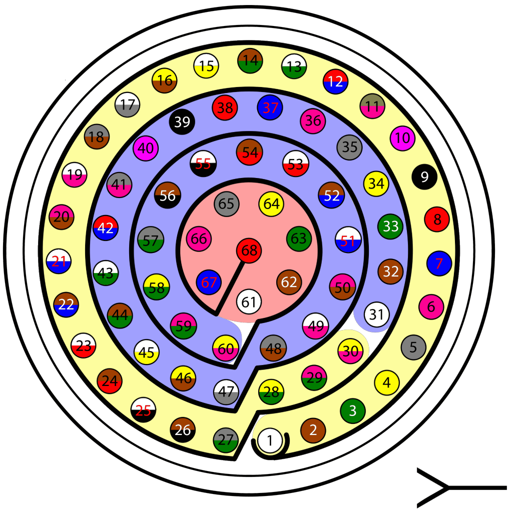
For higher voltage installations, voltage regulators must be used. For dimensioning voltage regulators take into account that a blocked servo can activate regulator thermal protection.

> ⚠ **Caution**
>
> **Caution!!** Power **Veronte Autopilot 1x** out of range can cause irreversible damage to the system. Please read carefully the manual before powering the system.

Autopilot 1x and servos can be powered by the same or different batteries. In case of having more than one battery on the system, a single point ground union is needed to ensure a good performance. The ground signal should be isolated from other noisy ground references (e.g. engines). If all grounds need to be connected, the connection should be made on the negative pole of the battery.

It is recommendable to use independent switches for autopilot and motor/actuators. During the system initialization, the PWM signal will be set to low level (0V), please make sure that actuators/motor connected support this behaviour before installing a single switch for the whole system.

Pinout



Connector for Autopilot 1x - HEW.LM.368.XLNP (frontal view)

> ⚠ **Warning**
>
> Check the pin number before connecting. The color code is repeated 3 times due to the amount of pins. First section (yellow) corresponds to pins 1-30, the second section (blue) to pins 31-60 and the third one (red) to pins 61-68. Pin number increases counterclockwise following the black line of the picture above.

<table>
<tr><th>PIN</th><th>Signal</th><th>Type</th><th>Description</th></tr>
<tr><td>1</td><td>I/O 0</td><td rowspan="7">I/O</td><td rowspan="7">Pins for PWM or digital I/O signals (0-3.3V). Protected against ESD and short circuit.
Warning
Each pin withstands a maximum current of 1.65 mA.</td></tr>
<tr><td>2</td><td>I/O 1</td></tr>
<tr><td>3</td><td>I/O 2</td></tr>
<tr><td>4</td><td>I/O 3</td></tr>
<tr><td>5</td><td>I/O 4</td></tr>
<tr><td>6</td><td>I/O 5</td></tr>
<tr><td>7</td><td>I/O 6</td></tr>
</table>

<table>
<tr><th>PIN</th><th>Signal</th><th>Type</th><th>Description</th></tr>
<tr><td>8</td><td>I/O 7</td><td></td><td></td></tr>
<tr><td>9</td><td>GND</td><td>GROUND</td><td>Ground signal for actuators 1-8.</td></tr>
<tr><td>10</td><td>I/O 8</td><td rowspan="8">I/O</td><td rowspan="8">Pins for PWM or digital I/O signals (0-3.3V). Protected against ESD and short circuit.
Warning
Each pin withstands a maximum current of 1.65 mA.</td></tr>
<tr><td>11</td><td>I/O 9</td></tr>
<tr><td>12</td><td>I/O 10</td></tr>
<tr><td>13</td><td>I/O 11</td></tr>
<tr><td>14</td><td>I/O 12</td></tr>
<tr><td>15</td><td>I/O 13</td></tr>
<tr><td>16</td><td>I/O 14</td></tr>
<tr><td>17</td><td>I/O 15</td></tr>
<tr><td>18</td><td>GND</td><td>GROUND</td><td>Ground signal for actuators 9-16.</td></tr>
<tr><td>19</td><td>RS232 TX</td><td>Output</td><td>RS 232 Output (-13.2V to 13.2V Max, -5.4V to 5.4V Typical). Protected against ESD and short circuit.</td></tr>
<tr><td>20</td><td>RS232 RX</td><td>Input</td><td>RS 232 Input (-25V to 25V Max, -0.6V Low and 2.4V High Threshold). Protected against ESD and short circuit.</td></tr>
</table>

PIN	Signal	Type	Description
21	GND	GROUND	Ground signal for buses.
22	ANALOG_3	Input	Input 0-3.3V. Protected against ESD and short circuit.
23	ANALOG_4	Input	Input 0-3.3V. Protected against ESD and short circuit.
24	GND	GROUND	Ground signal for buses.
25	CANA_P	I/O	CANbus interface, up to 1Mbps. Protected against ESD.
26	CANA_N	I/O	Twisted pair with a 120 ohms Zo recommended. Protected against ESD.
27	4XV_WD	I/O	Reserved. Do not connect.
28	CANB_P	I/O	CANbus interface. It supports data rates up to 1 Mbps. Protected against ESD.
29	CANB_N	I/O	Twisted pair with a 120 ohms Zo recommended. Protected against ESD.
30	GND	GROUND	Ground signal for buses.
31	I2C_CLK	Output	

PIN	Signal	Type	Description
			Clock line for I2C bus (0.3V to 3.3V). Protected against ESD and short circuit.
32	I2C_DATA	I/O	Data line for I2C bus (0.3V to 3.3V). Protected against ESD and short circuit.
33	GND	GROUND	Ground for 3.3V power supply.
34	3.3V	Power	3.3V - 100mA power supply. Protected against ESD short circuit with 100mA resettable fuse.
35	GND	GROUND	Ground for 5V power supply.
36	5V	Power	5V - 100mA power supply. Protected against ESD short circuit with 100mA resettable fuse.
37	GND	GROUND	Ground for analog signals.
38	ANALOG_0	Input	Analog input 0-3.3V. Protected against ESD and short circuit.
39	ANALOG_1	Input	

PIN	Signal	Type	Description
			Analog input 0-3.3V. Protected against ESD and short circuit.
40	ANALOG_2	Input	Analog input 0-3.3V. Protected against ESD and short circuit.
41	4XV_A	I/O	Reserved. Do not connect.
42	FTS1_OUT	Output	Deadman signal from comicro. Protected against ESD and short circuit.
43	FTS2_OUT	Output	!SystemOK Bit. Protected against ESD and short circuit.
44	4XV_B	I/O	Reserved. Do not connect.
45	UARTA_TX	Output	Microcontroller UART.
46	UARTA_RX	Input	Microcontroller UART.
47	GND	GROUND	Ground signal comicro power supply.
48	V_ARB_VCC	POWER	Veronte comicro power (6.5V to 36V). Protected against ESD and reverse polarity.

PIN	Signal	Type	Description
49	FTS3_OUT_MPU	Output	MPU alive voting signal, to use with 4xVeronte. It is a Square Wave at [100,125] Hz. Protected against ESD and short circuit.
50	OUT_RS485_P	Output	Non-inverted output from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit.
51	OUT_RS485_N	Output	Inverted output from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit.
52	IN_RS485_N	Input	Inverted input from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit.
53	IN_RS485_P	Input	Non-inverted input from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit.

PIN	Signal	Type	Description
54	OUT_GND	GND	Ground for RS-485 bus. **Warning** This is not a common GND pin.
55	EQEP_A	I/O	DIGITAL output / DIGITAL input / Encoder quadrature input A (0-3.3V). Protected against ESD and short circuit.
56	EQEP_B	I/O	DIGITAL output / DIGITAL input / Encoder quadrature input A (0-3.3V). Protected against ESD and short circuit. **Warning** Only use it as digital I/O with Veronte units of Hardware version 4.5 or lower.
57	EQEP_S	I/O	DIGITAL output / DIGITAL input / Encoder quadrature input A (0-3.3V). Protected against ESD and short circuit.
58	EQEP_I	I/O	

PIN	Signal	Type	Description
			DIGITAL output / DIGITAL input / Encoder quadrature input A (0-3.3V). Protected against ESD and short circuit.
59	GND	GROUND	Ground for encoders.
60	V_USB_DP	I/O	Veronte USB data line. Protected against ESD.
61	V_USB_DN	I/O	Veronte USB data line. Protected against ESD.
62	USB_SHIELD_GND	GROUND	USB cable shielding.
63	FTS_OUT_MPU	Output	Abort mission voting signal 1 from MPU, to use with 4x Veronte. Bit Low (0 V) if mission OK. High (3.3V) if mission wants to be terminated. Both pins are protected against ESD and short circuit. **Warning** Each pin withstands 2.5 A of maximum current.

PIN	Signal	Type	Description
64	FTS2_OUT_MPU	Output	Abort mission voting signal 2 from MPU, to use with 4x Veronte.
65	GND	GROUND	Veronte ground input.
66	GND	GROUND	Veronte ground input.
67	VCC	POWER	Veronte power supply (6.5V to 36V). Protected against ESD and reverse polarity. **Warning** Both pins are common. They MUST be connected to the same power supply.
68	VCC	POWER	

⚠ **Warning**

Remember!! All GND pins are common. Note that pin 54 is not a common GND pin.

Visit the following sections to know how to wire the **Autopilot 1x** to other devices via:

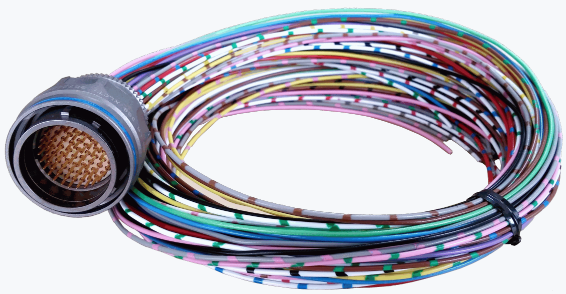
- RS232
- RS485/422
- CAN

To know the differences between version 4.5 and 4.8 (this one), read the Pinout changes from Autopilot 1x 4.5 - Troubleshooting section of the present manual.

Harnesses

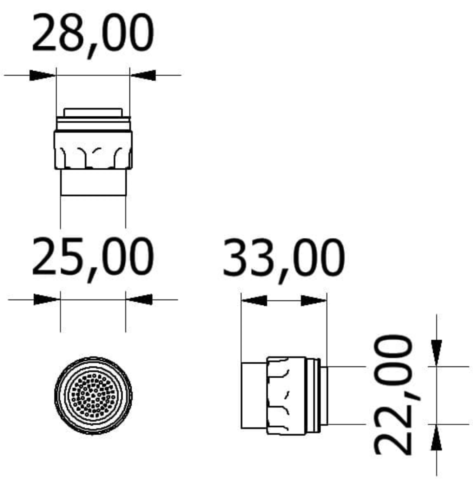
A wire harness is a structured assembly of cables and connectors used to organize and manage wiring in electrical and electronic systems. It is designed to ensure a tidy and secure installation of cables, preventing tangles, electromagnetic interference, and facilitating maintenance.

Veronte Autopilot 1x 4.8 has two compatible harnesses:

Veronte Harness Blue 68P	**Dev Harness 1x 4.8**
	
Harness available on demand with the Embention reference **P001114**	Harness available on demand with the Embention reference **P007043**

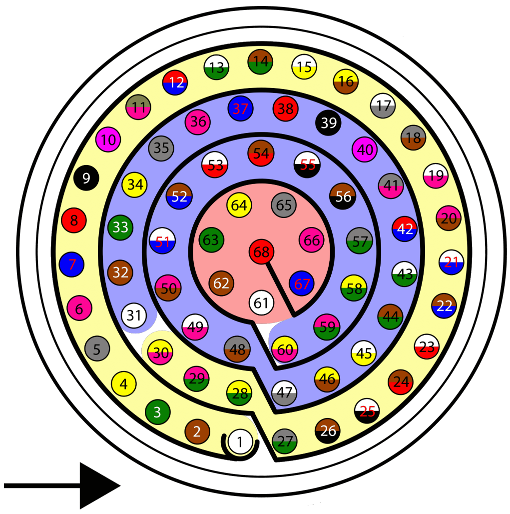
Dimensions

- **Harness Blue 68P wire gauge**: 22 AWG
- **Cables length**: 52 cm
- **Harness plug dimensions**:



Connector FGW.LM.368.XLCT dimensions (cm)

Pinout



Harness plug - FGW.LM.368.XLCT (frontal view)

Veronte Harness Blue 68P

The pinout of this harness is the same as the Veronte Autopilot 1x pinout above. The **color code** of the harness wires is given below.

> ⚠ **Warning**
>
> Check the pin number before connecting. The color code is repeated 3 times due to the amount of pins. First section (yellow) corresponds to pins 1-30, the second section (blue) to pins 31-60 and the third one (red) to pins 61-68. Pin number increases following the black line of the pictures above: counterclockwise for the connector and clockwise for the plug.

PIN	Color Code	PIN	Color Code
1	White	35	Gray
2	Brown	36	Pink
3	Green	37	Blue
4	Yellow	38	Red
5	Gray	39	Black
6	Pink	40	Violet
7	Blue	41	Gray - Pink
8	Red	42	Red - Blue
9	Black	43	White - Green
10	Violet	44	Brown - Green
11	Gray - Pink	45	White - Yellow
12	Red - Blue	46	Yellow - Brown
13	White - Green	47	White - Gray
14	Brown - Green	48	Gray - Brown
15	White - Yellow	49	White - Pink
16	Yellow - Brown	50	Pink - Brown
17	White - Gray	51	White - Blue
18	Gray - Brown	52	Brown - Blue
19	White - Pink	53	White - Red

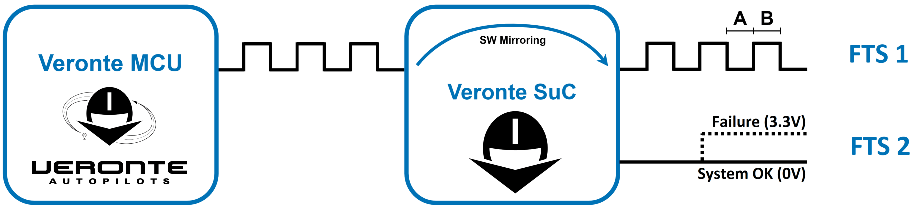
PIN	Color Code	PIN	Color Code
20	Pink - Brown	54	Brown - Red
21	White - Blue	55	White - Black
22	Brown - Blue	56	Brown - Black
23	White - Red	57	Gray - Green
24	Brown - Red	58	Yellow - Green
25	White - Black	59	Pink - Green
26	Brown - Black	60	Yellow - Pink
27	Gray - Green	61	White
28	Yellow - Green	62	Brown
29	Pink - Green	63	Green
30	Yellow - Pink	64	Yellow
31	White	65	Gray
32	Brown	66	Pink
33	Green	67	Blue
34	Yellow	68	Red

Dev Harness 1x 4.8

This harness has some connectors already implemented for easy operation. Below is detailed information on which pins these connectors are connected to:

Connector	PIN	Signal
Main VCC	65	GND
	66	GND
	67	VCC
	68	VCC
SuC VCC	47	GND
	48	V_ARB_VCC
RS232 Connector	19	RS 232 TX
	20	RS 232 RX
	21	GND
Maintenance Button	31	I2C_CLK
	32	I2C_DATA
Jack Connector	18	GND
	55	EQEP_A
USB	60	V_USB_DP
	61	V_USB_DN
	62	USB_SHIELD_GND

Flight Termination System (FTS)



Flight Termination System

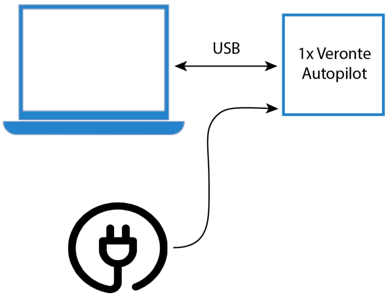
Veronte Autopilot 1x integrates two different FTS pins (42 and 43):

FTS1 - Deadman (Pin 42): On this pin, **Autopilot 1x** outputs a square wave with A = ~5ms and B = ~5ms (3.3V). Its frequency can be higher right after the rebooting (around 300-400Hz), but A and B must be always < 8ms.

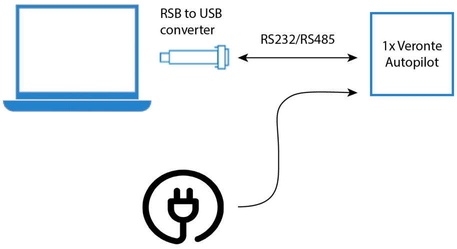
FTS2 - !SystemOK (Pin 43): Its output is 0V when the system is working as expected and 3.3V when some error is detected. In detail, pin 43 goes high if A > 8ms or B > 8ms in the deadman signal sent by the Main Processor Unit (MPU).

Software Installation

In order to configure and use **Veronte Autopilot 1x**, there are two ways to connect it to a computer: USB or serial.



USB connection



Serial connection

To install the required software and configure **Veronte Autopilot 1x**, read the 1x Software Manual.

An example of the configuration required for a correct communication between the PC and **Autopilot 1x** can be found in the AP communication with PC - Integration examples section of the **1x PDI Builder** user manual.

Operation

Types of operations

Veronte Autopilot 1x is an advanced system designed to enable the operation of autonomous vehicles, offering three control modes: automatic, assisted, and manual. This versatile autopilot can be used in both uncrewed and manned vehicles, integrating a **FLY-BY-WIRE** system that ensures precise and safe control at all times.

One of the main advantages of the Veronte Autopilot 1x is its configurability, allowing it to be adapted for different operational needs. Depending on the chosen configuration, the system can handle various types of takeoff, such as runway or catapult launches, among others.

Veronte Ops is the Veronte application dedicated to operating the system, providing an intuitive interface for mission management and monitoring. Additionally, for more flexibility, the system can also be operated through VCP (Veronte Communication Protocol), enabling the creation of custom control stations or integration with onboard mission computers for more specific or advanced applications.

In summary, **Autopilot 1x** stands out for its versatility, ease of integration, and customization options, offering a comprehensive solution for a wide range of autonomous vehicle applications.

In addition, for the different types of operations, the user may need to make different connections, configurations and/or integrations with external devices with **Veronte Autopilot 1x**. Therefore:

- Examples of how to make connections to **Autopilot 1x** such as **wiring connection via CAN** or with a **Serial to Ethernet Converter** are detailed in the Integration examples section of this manual.
- Examples of how to integrate **Autopilot 1x** with external devices such as datalinks are detailed in the Datalinks - Integration examples section of the present manual.

Please take a look at these sections for further explanations.

This section summarizes a list of possible options to operate an **Autopilot 1x** in different situations.

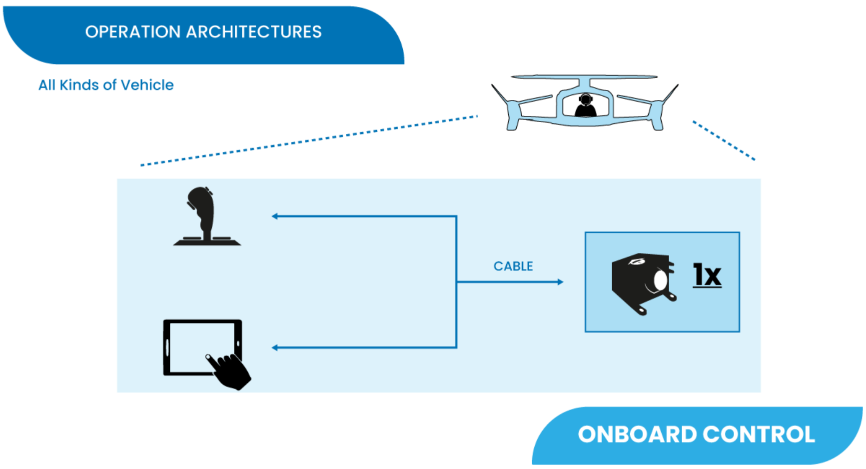
> **Tip**
>
> Most of the following diagrams can be used independently or combinated, to create redundant systems or backup solutions.

Operation Architectures

Veronte Autopilot 1x allows for a wide variety of communication and control solutions to adapt to each mission and platform specifications.

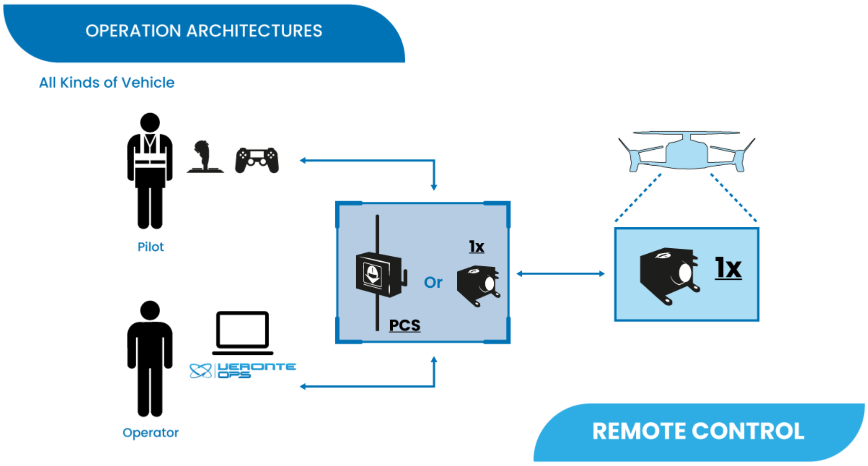
Onboard Control Setup

1x allows to control aircrafts (such as eVTOLs) by pilots on board in a flight deck. Pilots can use as controller joysticks, computers, tablets or any device able to communicate through PPM, CAN Bus, RS232 or RS485.



Remote Control Setup

The following image shows the standard Veronte System Layout for remote operation.

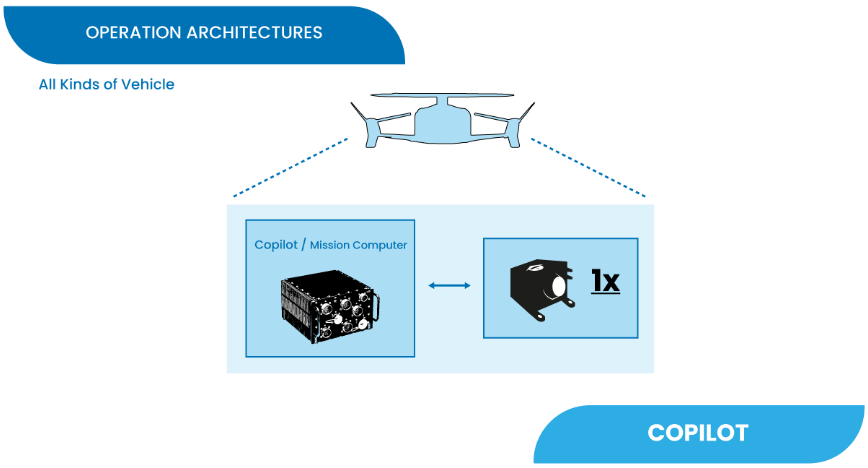


In the standard remote layout, an Operator (Internal Pilot) controls the UAV from the Ground Station using **Veronte Ops**.

Additionally, a Safety Pilot (External Pilot) is connected to the Ground Station using a radio controller. The stick commands are read by the Ground Unit and re-routed to the Air Unit. The Safety Pilot is able to take control of the flight at any point using an automation.

Copilot Control Setup

Veronte system allows integration with onboard mission computers for more specific or advanced applications.

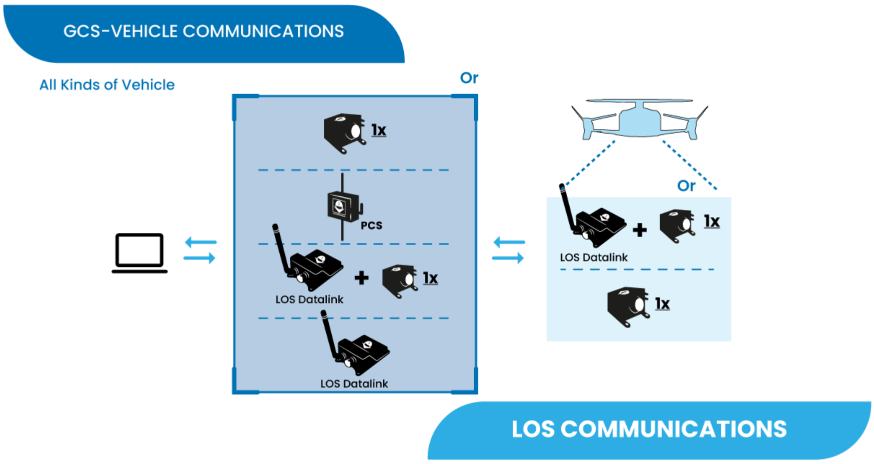


GCS-Vehicle Communications

The following are some examples and possible solutions for establishing communication between the ground control station and the vehicle.

LOS Communications

The following diagram shows the different options of GCS and in-vehicle solutions to establish correct Line of Sight (LOS) communications between them.



Depending on the requirements and needs of their mission, users can choose as GCS:

- **Autopilot 1x** with its internal LOS module
- **PCS**
- **Autopilot 1x** with an **external LOS Datalink**
- **LOS Datalink**

And on the vehicle side:

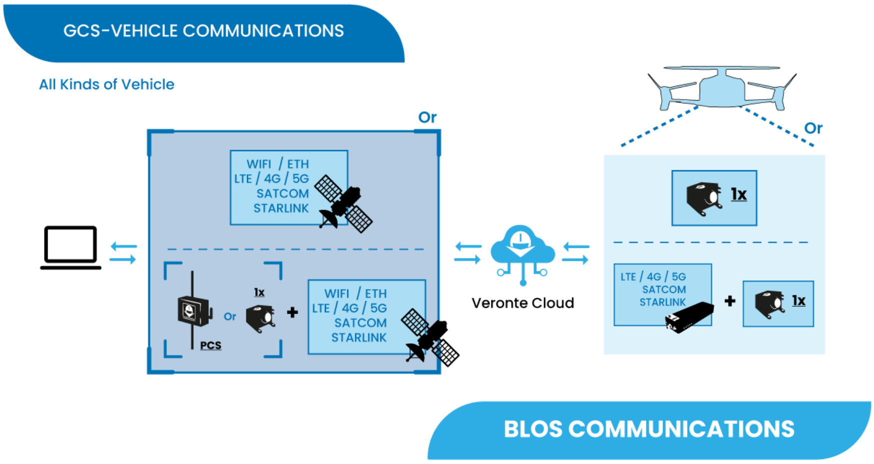
- **Autopilot 1x** with its internal LOS module
- **Autopilot 1x** with an **external LOS Datalink**

BLOS Communications

Veronte Cloud enables secure and efficient Beyond Line of Sight (BLOS) communication between the autopilot onboard a vehicle and the control station. It supports various communication methods, offering a flexible architecture to suit different operational requirements:

- **Autopilot 1x Internal Module**: Embedded **4G** module within **Autopilot 1x**.
- **LTE/4G/5G Module**: External LTE module for wireless communication.
- **Satcom Module**: Satellite communications device for global coverage.

- **Starlink**: High-bandwidth, global communications module.



These communication methods can be used both at the **ground control station** and **onboard**, enabling seamless switching between methods or simultaneous use for redundancy and enhanced reliability. They can also be combined to meet specific project requirements.

Control Station Connectivity Options

The control station connects to **Veronte Cloud** through two primary methods:

- **Option A: Direct PC Internet Connection**
  The **control station PC** connects directly to the Internet for communication with **Veronte Cloud**. This can be achieved using any available means of Internet communication:
  - Ethernet or Wi-Fi
  - LTE/4G/5G
  - Satellite Communication (Satcom)
  - Starlink
- **Option B: Connection via Veronte PCS/1x**
  The **control station PC** connects to the **Veronte PCS/1x module**, which manages the connection to the BLOS datalink module. The **PCS/1x** module supports:
  - Its **internal 4G module** for direct connectivity.
  - **External communication modules** (LTE/4G/5G, Satcom, Starlink, etc.).

This setup enhances communication reliability by leveraging Veronte's dedicated hardware for connection management and enabling the use of additional sensors integrated within the **PCS/1x** module.

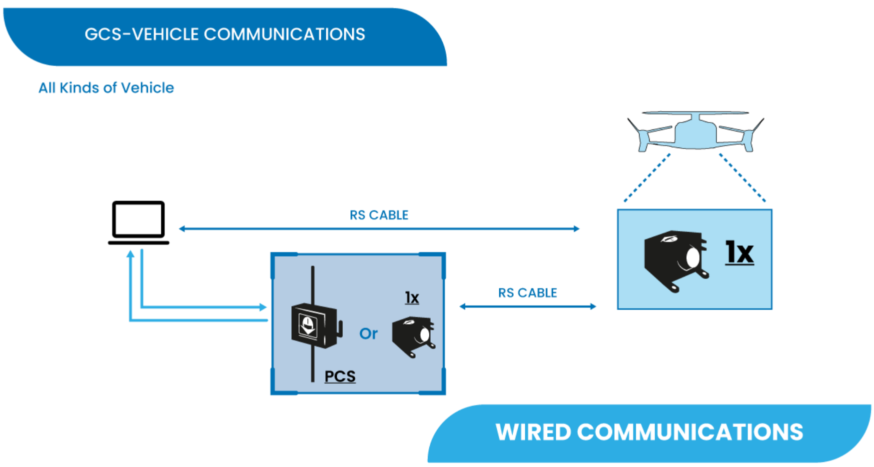
Onboard Connectivity Options

For onboard BLOS communications, **Veronte Autopilot 1x** system offers two main methods:

- **Option C: Internal 4G Module in Veronte Autopilot 1x**
  Autopilot 1x comes equipped with an internal 4G module that connects directly to **Veronte Cloud**. This option is compact and does not require additional external hardware.
- **Option D: External Module Connected to Autopilot 1x**
  The autopilot can integrate an **external communication module** (LTE/4G/5G, Satcom, Starlink, etc.) to enable BLOS communication with **Veronte Cloud**. This provides flexibility and allows for customization based on specific mission or environmental needs.

Wired Communications

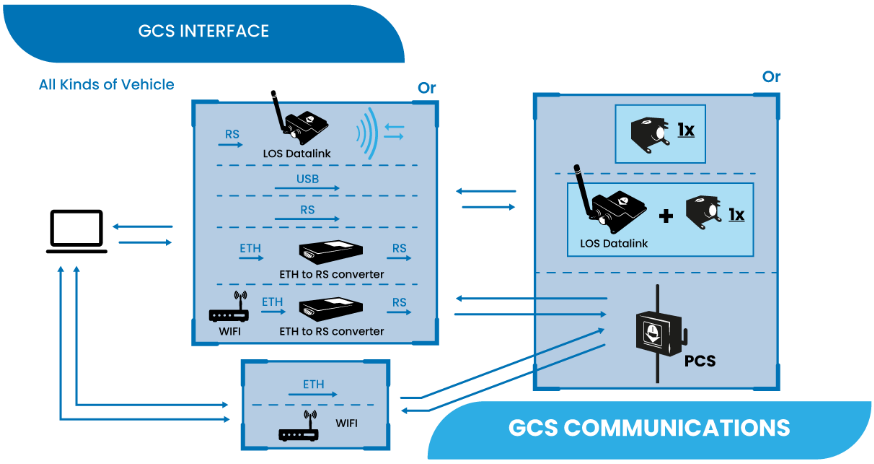
For operations where the control station is directly connected to the onboard autopilot by cable.



GCS Interface

This diagram represents some of the many ways to establish communication between the different parts of a Ground Control Station setup.

> ⓘ **Note**
>
> In a Ground Control Station setup there is usually a PC on one side and an **Autopilot 1x** with its internal LOS module, an **Autopilot 1x** with an **external LOS Datalink** or a **PCS** on the other side.



Direct connection

- The **PC** can directly connect a **PCS** via **USB**, **RS**, **Ethernet** or **Wifi**.
- The **PC** can directly connect an **Autopilot 1x** with its internal LOS module or **Autopilot 1x** with an **external LOS Datalink** via **USB/RS**.

Combined connections

Below are different connection methods that enable communication between the **PC** and an **Autopilot 1x** with its internal LOS module, an **Autopilot 1x** with an **external LOS Datalink** or a **PCS** via an additional device:

- PC connected via RS to a **LOS Datalink**, establishing a datalink connection to the other side of the GCS setup.
- Connection through an **Ethernet-to-RS converter**, i.e. Ethernet on the PC side and RS on the other side.
- The PC connects via **wifi** and the wifi modem then communicates with the other side of the GCS setup through an **Ethernet-to-RS converter**.

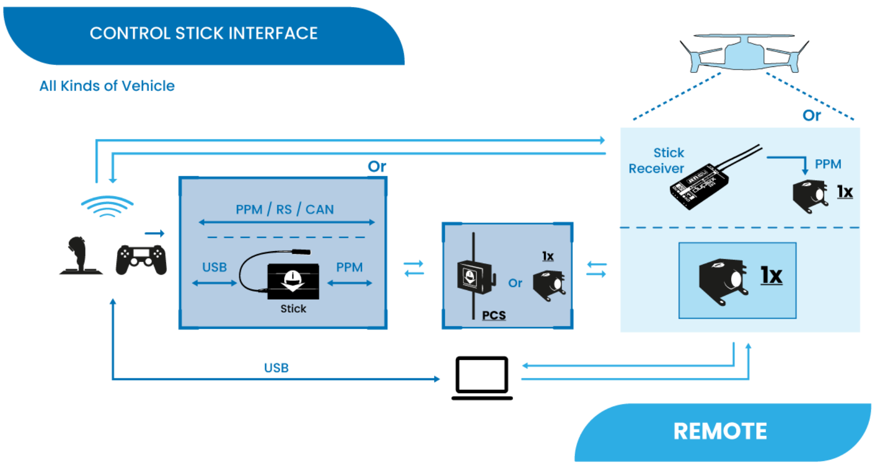
Control Stick Interface

This section presents the different types of manual control from stick to the onboard autopilot.

Veronte allows for a wide variety of pilot interface solutions in order to interact with manual flight modes, assisted flight modes (arcade) or payloads.

Remote Control Stick

A wide variety of controllers can be used to pilot manually aircrafts, such as RC transmitters, pedals, sticks or buttons. Veronte software allows the use of any device that is detected as a remote controller by the operative system.

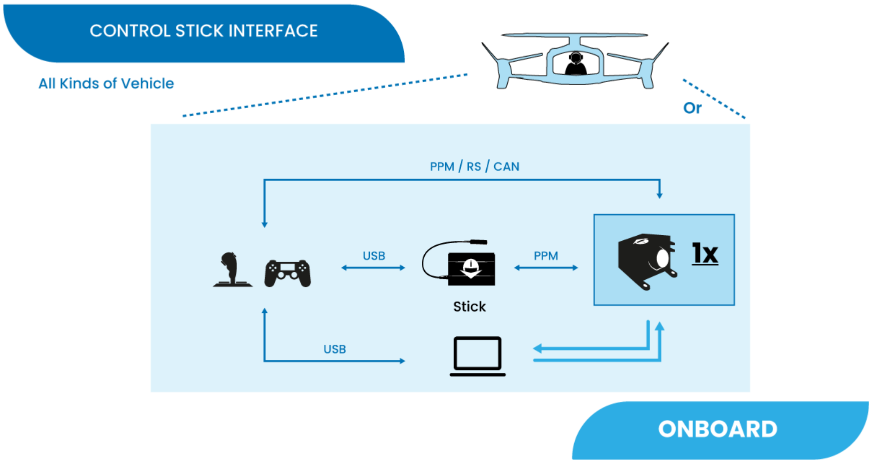


Although the most common way of control is to directly connect a **stick** via **PPM**, **RS** or **CAN** to a control station (**PCS/1x**) which then communicates with the onboard autopilot,

- It is possible to establish a link connection between a stick and a **stick receiver** integrated in the vehicle, which is connected via **PPM** to **Autopilot 1x**.
  This allows for a backup manual channel when there is a main channel loss and an emergency manual landing is needed. Recommended for initial developement stages where automatic landing phases are not defined yet.
- A **Veronte Stick** allows the connection of **USB sticks** to a control station (**PCS/1x**), converting **USB to PPM**. Then, the GCS communicates with the onboard autopilot for control.
- A **USB stick** can be connected directly to the **PC** to establish communication with the onboard autopilot for control.

Onboard Control Stick

In operations with pilots onboard in a flight deck (such as eVTOLs), the sticks can directly control the vehicle's **Autopilot 1x**.



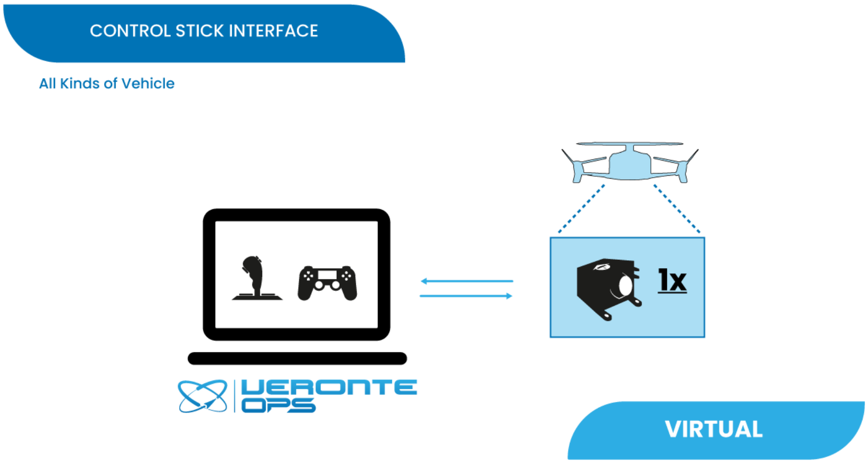
Some examples are:

- To directly connect a stick via **PPM**, **RS** or **CAN** to the autopilot.
- To use a **Veronte Stick** that converts **USB to PPM**, allowing connection between **USB sticks** and the autopilot.
- Connect a **USB stick** to a **PC** which establishes communication with the autopilot.

Virtual Stick

The Virtual stick feature allows to integrate as a stick controller any device that can interface with **Autopilot 1x** (RS232, RS485, ADC, CAN...) and can provide control reference values.

While the configuration is slightly more complex, this feature allows using a wide variety of devices as flight control interfaces.



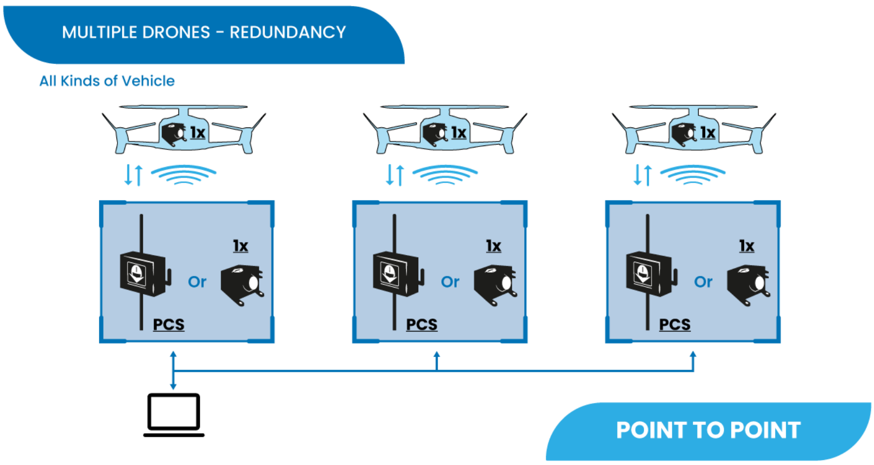
Multiple Drones/GCS - Redundancy

Due to Veronte's modular configuration, it is possible to integrate several onboard and ground units within the same network.

> **Note**
>
> Users are free to combine the different multiple drones solutions with the multiple GCS solutions.

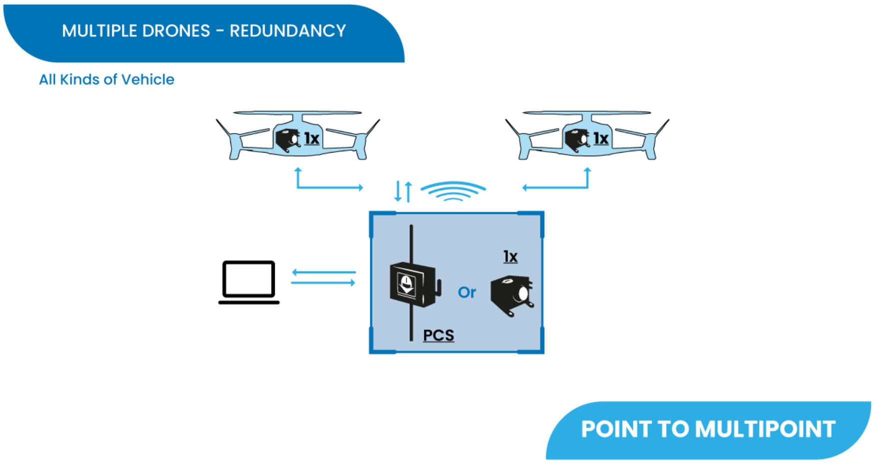
Multiple Drones - Point to Point

Standard multiplatorm setup.



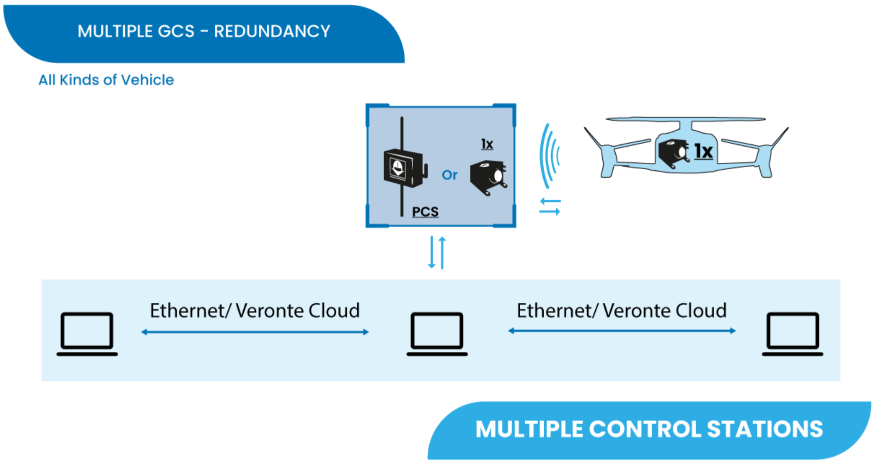
Multiple Drones - Point to Multipoint

Managing several platforms with a single radiolink.



Multiple GCS

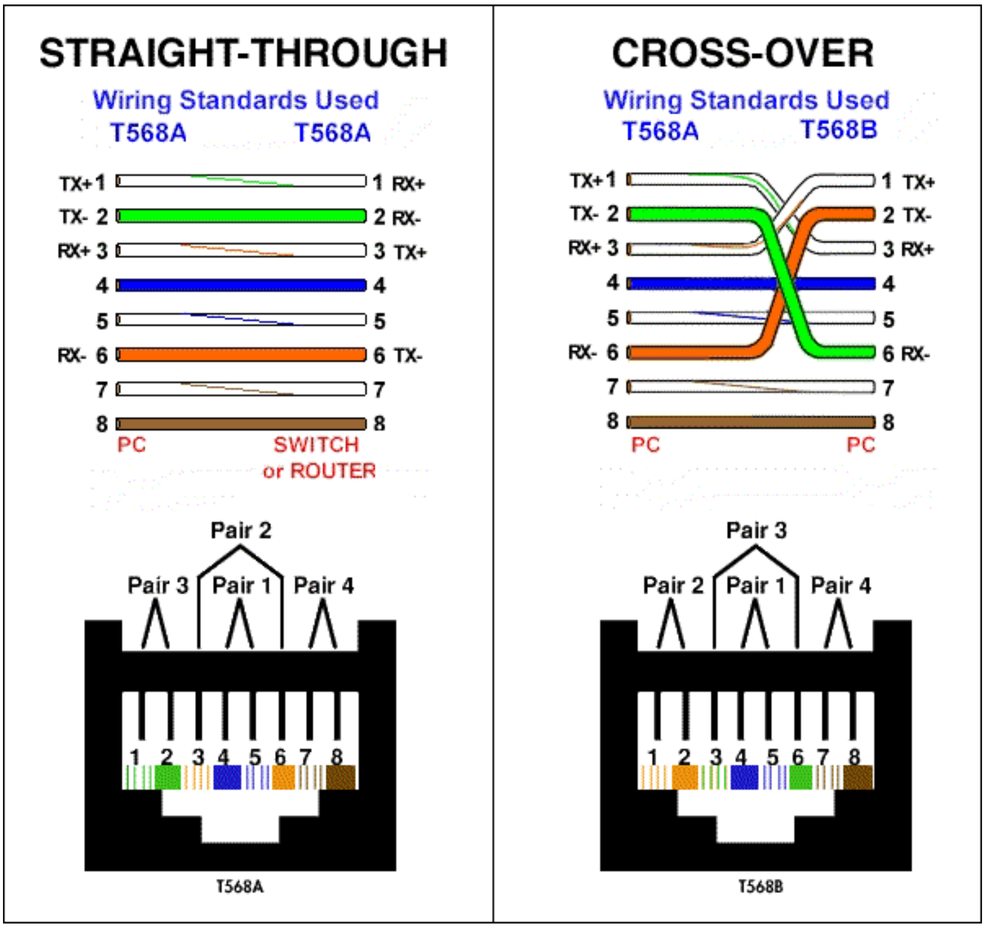
For long range operations with several LOS stations.



For remote solutions with LOS backup operator, **Veronte Cloud** allows the connection between PCs.

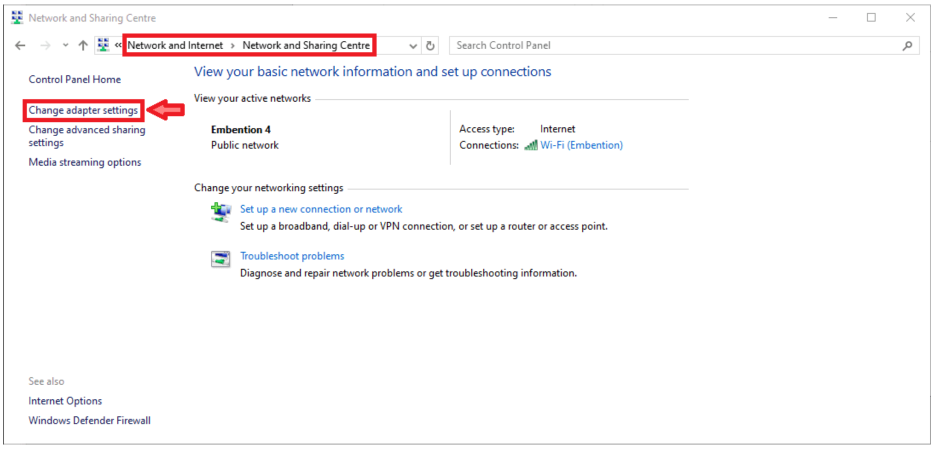
To correctly establish communication between the different PCs via **Ethernet** the following steps should be carried out:

1. Make the **physical connection** with ethernet cables, the two different types of ethernet cables can be used:
   - **Straight-Throught**
     Connect each PC to an **ethernet switch** with its Straight-Through ethernet cable (i.e. users will need 2 cables).
   - **Crossover**
     Connect the PCs directly to each other with a crossover ethernet cable.



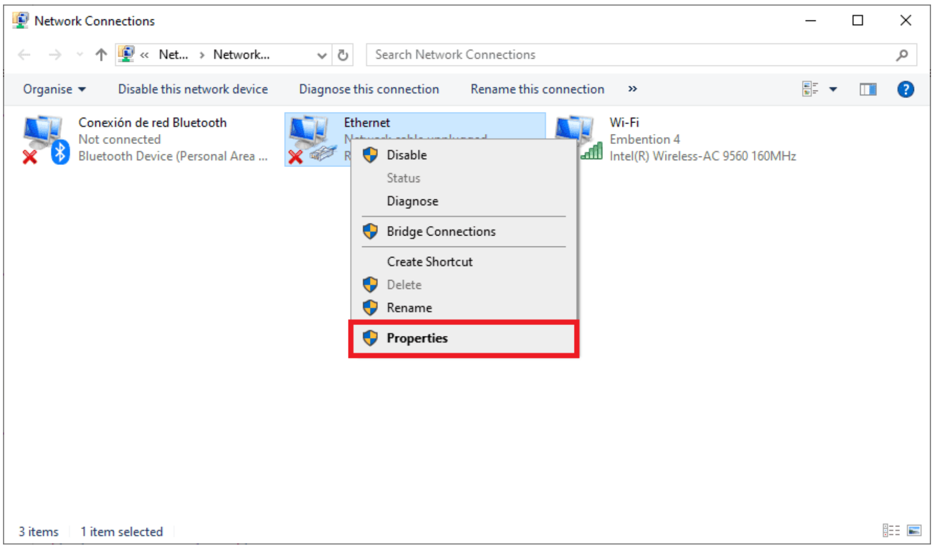
Straight-Through vs Crossover cables

2. On each PC, change the ethernet adapter settings to a static IP so that both are on the same subnet. To do this:
   - In the **Control Panel**, go to **Network and Internet**.
   - Open **Network and Sharing Centre** menu and click **Change adapter settings**.

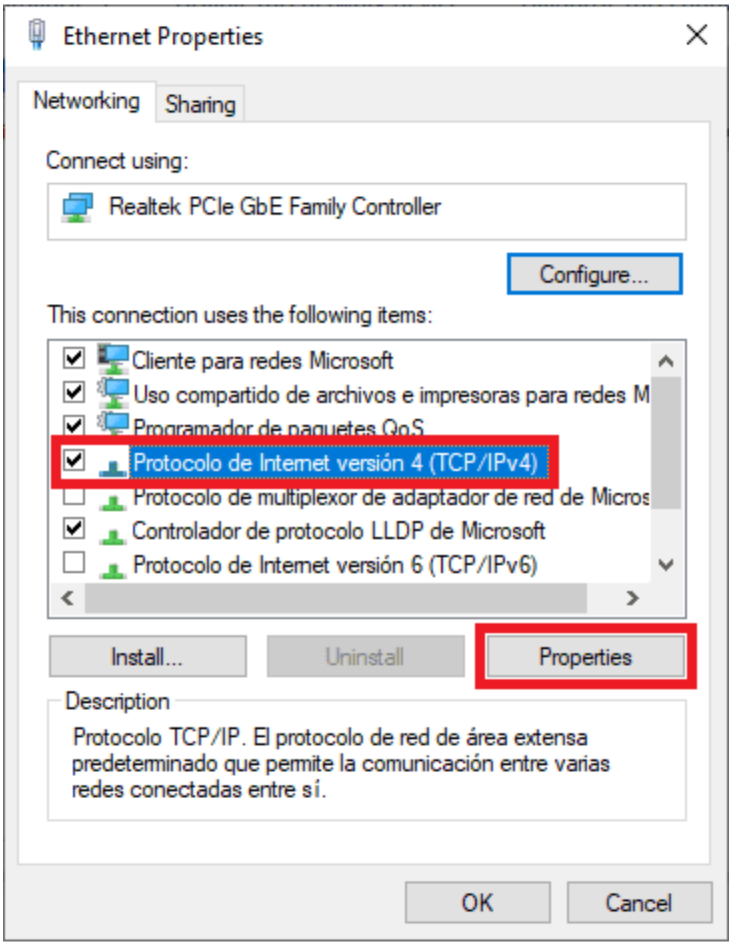
     Ethernet connection 1

   - Select **Local Area Connection**, right click, and select **Properties**.



Ethernet connection 2

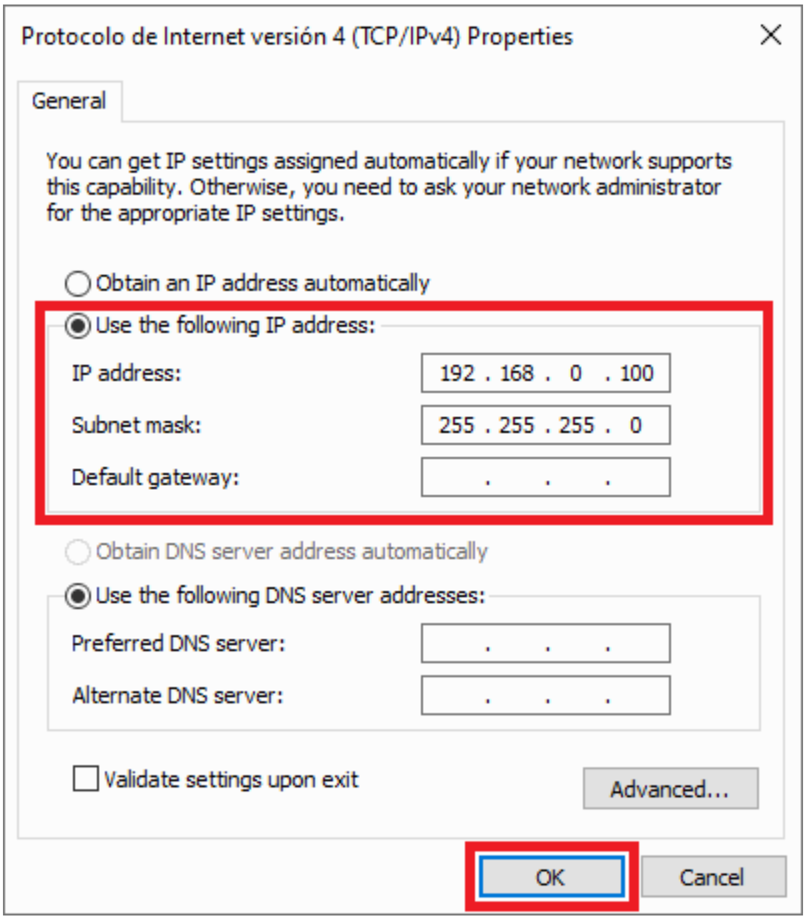
- Select **IPv4** and click **Properties**.



Ethernet connection 3

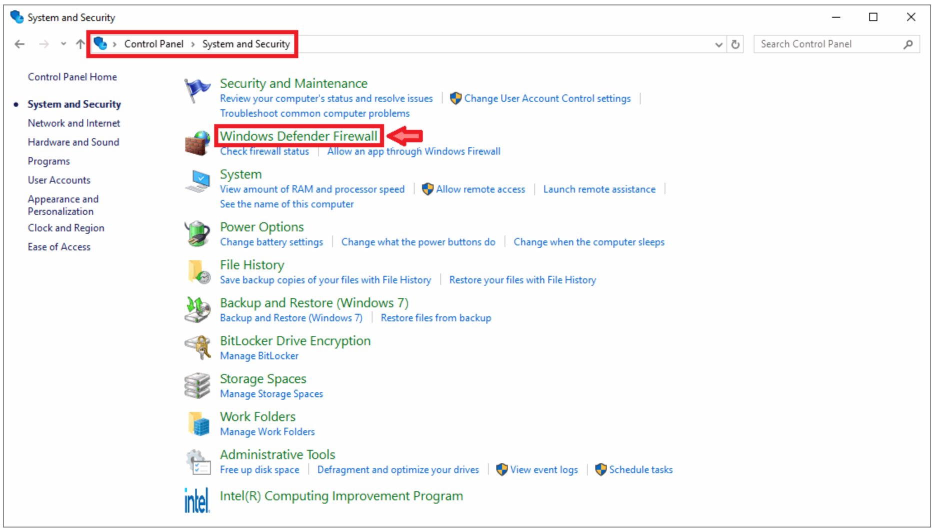
- Set **IP address** to a **static IP** (e.g. 192.168.0.100) and **Subnet mask** to 255.255.255.0. Click **OK**.

> **Important**
>
> If on this PC the IP address is set to 192.168.0.100, on the other PCs, the IP address must be set to **192.168.0.XXX** (e.g. 192.168.8.234), so that they are on the same subnet.



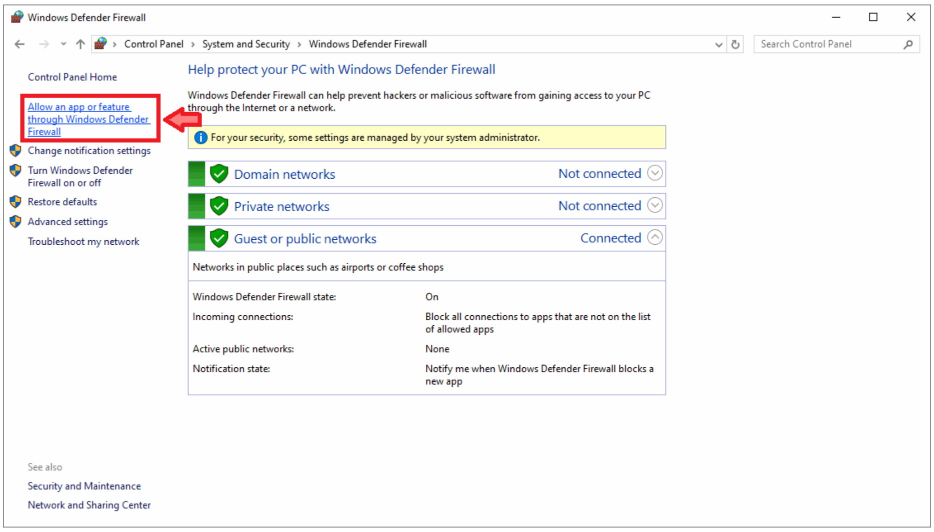
Ethernet connection 4

3. Allow **VeronteLink** to go through the Firewall on the PC that will run it, hereafter PC primary. To do so:
   - In the **Control Panel**, go to **System and Security**.



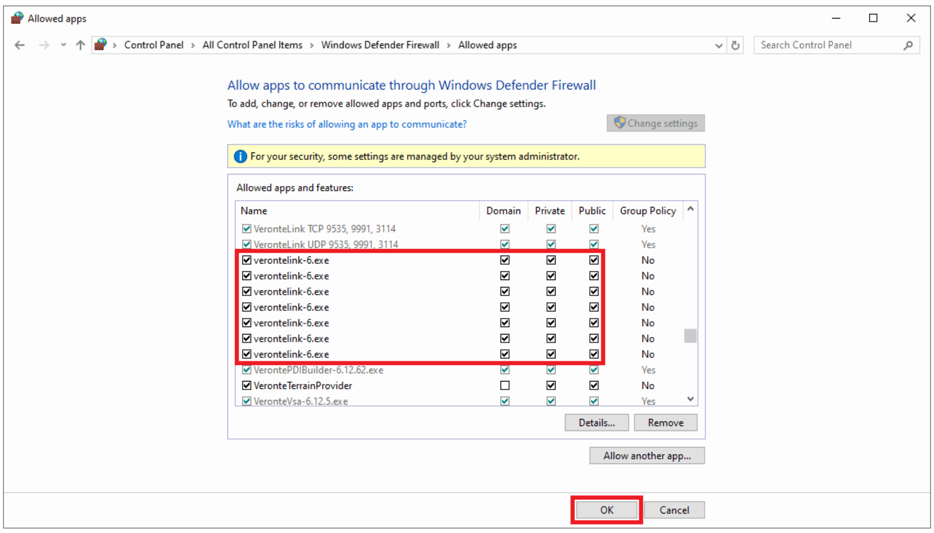
Windows Firewall 1

- Open **Windows Defender Firewall** and click on **Allow an app through Windows Defender Firewall**.



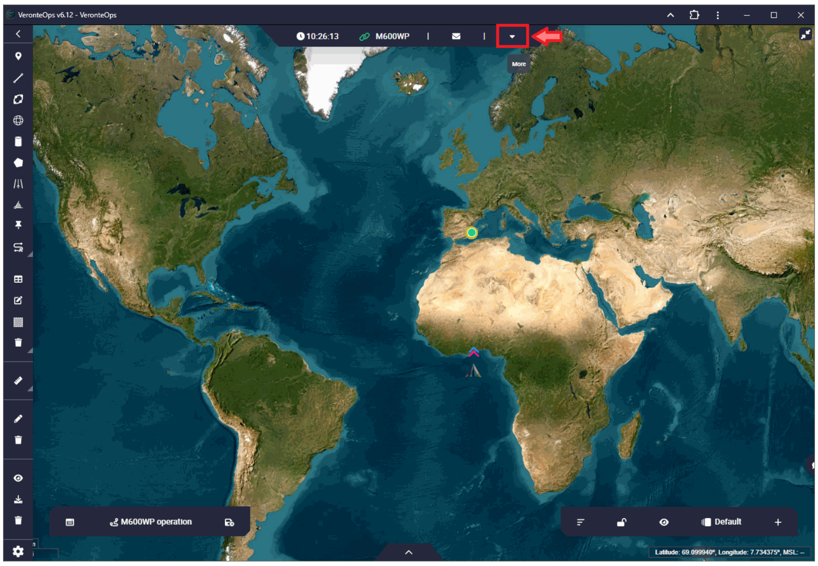
Windows Firewall 2

- Check that **Veronte Link** app is **allowed**.



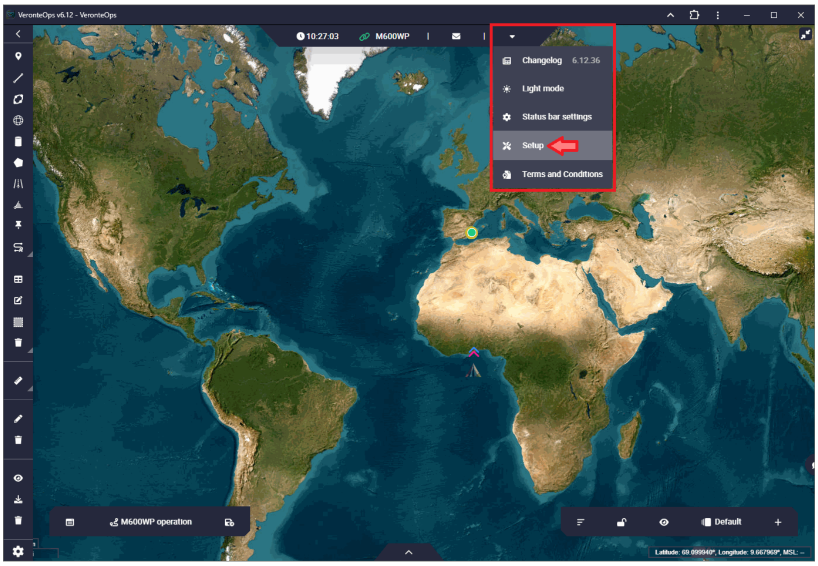
Windows Firewall 3

4. On the PC secondary, in **Veronte Ops** change the **Veronte Link Host** option setting to the **IP of the PC primary**. To do this:
   - Open **Veronte Ops**.
   - In the **Status bar**, click the **arrow** on the right of the bar to display a **drop-down menu**.



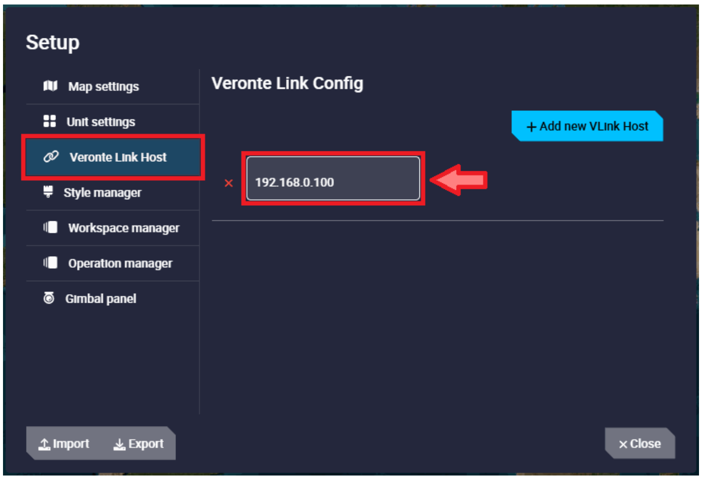
Veronte Ops - Status bar

   - In it, open the **Setup** menu.



Veronte Ops - Setup menu

   - Next, go to the **Veronte Link Host** settings.
   - Change the IP localhost to the IP of the PC primary.



Veronte Ops - Veronte Link Host settings

For more information on this settings, refer to the Setup - Veronte Ops configuration section of the **Veronte Ops** user manual

5. Finally, **Autopilot 1x** connected to the PC primary should be seen in the **Veronte Ops** open on this PC, as well as on the PC secondary.

   If users have any problems when trying to connect **Veronte Ops** to **Veronte Link**, refer to the Connecting to Veronte Link - Troubleshooting section of the **Veronte Ops** user manual.

If after following the steps described above users are not able to operate in this way, please contact support team by opening a **Ticket** in your Joint Collaboration Framework.

Preparation for operation

Here the user will find different checklists that the Embention team considers useful to follow as a guideline before performing an operation.

> **Important**
>
> This is only an **example**, it may vary depending on the operation to be performed and the user's platform.

1. Hardware revision
2. Software revision
3. Functional system test
4. Equipment checklist

Maintenance

Preventive maintenance

Apart from cleaning, no extra maintenance is required to guarantee the correct operation of the **Veronte Autopilot 1x**.

In order to clean **Veronte Autopilot 1x** properly follow the next recommendations.

- Turn off the device before cleaning.
- Use a clean, soft, damp cloth to clean the unit.
- Do not immerse the unit in water to clean it.

Software update

To update the software, an additional app is required: Veronte Updater.

> **Note**
>
> The file with the new software version will be shared with the customer in the **Joint Collaboration Framework** when it is requested.
>
> For more information about the **Joint Collaboration Framework**, read its user manual.

Compatible Devices

Actuators/Servos

Company	Comments
Volz	**I/O:** PWM, RS485 full duplex and RS485 half duplex (reduced functions) **Type:** simplex, redundant and OPV To know how to integrate it with Veronte Autopilot 1x, read the Volz DA26 integration example
Pegasus Actuators GmbH	**I/O:** PWM, RS485 full duplex and CAN-BUS **Type:** simplex

Company	Comments
	and redundant. To know how to integrate it with Veronte Autopilot 1x, read the Pegasus PA-R-135-4 integration example (CAN) or the Pegasus PA-R-135-4 integration example (Serial)
	I/O: PWM and CAN-BUS **Type:** linear actuator and servo To know how to integrate it with Veronte

Company	Comments
	Autopilot 1x, read the Ultra Motion integration example
KST® DIGITAL SERVO	**I/O:** PWM, RS232, RS485 and CAN-BUS
SAVÖX	**I/O:** PWM, RS232, RS485 and CAN-BUS

Other companies		
HiTEC	MULTIPLEX®	

ADS-B

Company	Comments
	Products: Ping20S (ADS-B Out)

Company	Comments
uAvionix	Ping1090 (ADS-B IN/ OUT)
Sagetech AVIONICS	**All products of the following families:** MX XP
EMBENTION	**Autopilot 1x** includes an internal ADS-B transceiver. To know how to configure it, read the Transponder/ ADS-B - Devices section of the **1x PDI Builder** user manual.

Company	Comments
DAEDALEAN APPLIED INTELLIGENCE	

Air Data Sensors

Company	Comments
EMBENTION	High Speed Pitot Sensor, read the High Speed Pitot Sensor integration example to know how to integrate it with Veronte Autopilot 1x.
MGL AVIONICS	**OAT 428:** For detailed information on its integration with Veronte Autopilot

Company	Comments
	1x, refer to OAT sensor 428 integration example
swiss-airdata www.swiss-airdata.com	
AEROPROBE Revolutionary Technology	

Altimeters

Company	Comments
AINSTEIN	**I/O:** UD-1 (CAN-bus) To know how to integrate it with Veronte Autopilot 1x, read the Ainstein CAN Radar integration example
HEXAGON	**I/O:** CAN-bus

Company	Comments
	I/O: I2C and CAN-bus **Product:** LW20, SF11, SF20 To know how to integrate them with Veronte Autopilot 1x, read the Lightware LW20 Lidar and Lightware SF20 Lidar integration examples
	I/O: PWM **Product:** LIDAR-Lite v3 To know how to integrate it with Veronte Autopilot 1x, read the Lidar

Company	Comments
	Garmin Lite v3 integration example

Cameras

Company	Comments
	Autopilot 1x reads identified objects by their cameras.

Control stations

Company	Comments
	Products: LCS PCS

Datalinks

BLOS Communications

Company	Comments

Company	Comments
	Broadband UAV Satcom: IMS-350 **Midband UAV Datalink and GPS System:** DLS-100
	Satellite Communications: RockBLOCK
	Requires Veronte COM

LOS Communications

Company	Comments
	Antenna: Tracking Systems T28 **Radio module:** • SDL, read the Veronte SDL integration example to know how to integrate it with Veronte Autopilot 1x • XDL
	I/O: RS232 communication tunnel **Microhard radio:** pDDL900-ENC, read the Microhard integration example to know how to integrate it with Veronte Autopilot 1x
	I/O: RS232 communication tunnel **DTC radio:** SOL8SDR-C, read the DTC integration example to know

Company	Comments
	how to integrate it with Veronte Autopilot 1x
	I/O: RS232 communication tunnel **Streamcaster radio:** 4200E, read the Silvus integration example to know how to use it with Veronte Autopilot 1x
	Radio: To know how to configure Digi radios, read its user guide and read the Digi integration example to know how to integrate it with Veronte Autopilot 1x

Engines

Jet Engines

Expansion modules

Company	Comments
	Products: CEX, read the CEX connection integration example to know how to integrate it with Veronte Autopilot 1x. MEX, read the MEX connection integration example to know how to integrate it with Veronte Autopilot 1x.

GNSS Receivers

Company	Comments
	GNSS, read the NexNav

Company	Comments
NexNav	GNSS integration example to know how to integrate it with Veronte Autopilot 1x.
ublox	**GNSS:** ZED-F9P-02B
NovAtel	

Company	Comments

IMUs & Compass

Company	Comments
	IMU: VN-300 To know how to integrate it with Veronte Autopilot 1x, read the Vectornav VN-300 integration example
	IMU: HWT905-232 To know how to integrate it with Veronte Autopilot 1x, read the WitMotion HWT905-232 integration example

Company	Comments
	Magnetometer: MEX To know how to integrate it with Veronte Autopilot 1x, read the MEX integration example
	Magnetometer: HMR2300-232 To know how to integrate it with Veronte Autopilot 1x, read the Magnetometer Honeywell HMR2300 integration example
	Magnetometer: RM3100-CB To know how to integrate it with Veronte Autopilot 1x, read the PNI RM3100 integration example

Motor controllers / ESC

Company	Comments
	Veronte products: MC110, read the MC110 connection integration example to know how to integrate it with Veronte Autopilot 1x. MC24, read the MC24 connection integration example to know how to integrate it with Veronte Autopilot 1x. MC01, read the MC01 connection integration

Company	Comments
	example to know how to integrate it with Veronte Autopilot 1x.
T-MOTOR THE SAFEST PROPULSION SYSTEM	**I/O:** PWM
Currawong ENGINEERING	**I/O:** CAN-bus

Other Companies	Comments
Hacker Brushless Motors Industrial Solutions	**I/O:** PWM, RS232, RS485, and CAN-bus
KDEDirect	
AXi	

Power management units

Company	Comments
EMBENTION	**Veronte product:** R12S

Company	Comments
	R12F R24F
VISION AIR TRONICS	

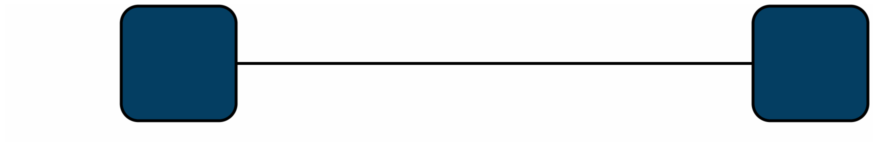
Transmitters

Company	Comments
Futaba	**Products:** 8J/10J/12K/14SG with 8 channels 12K/14SG with 12 channels T18SZ with 8 channels
EMBENTION	Stick

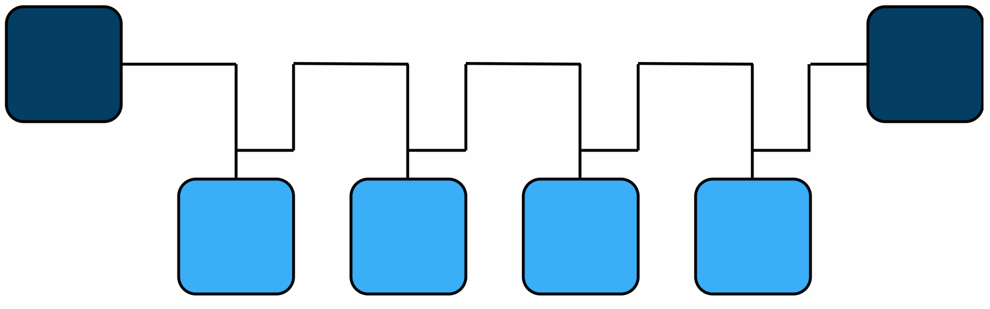
Integration examples

Wiring connection

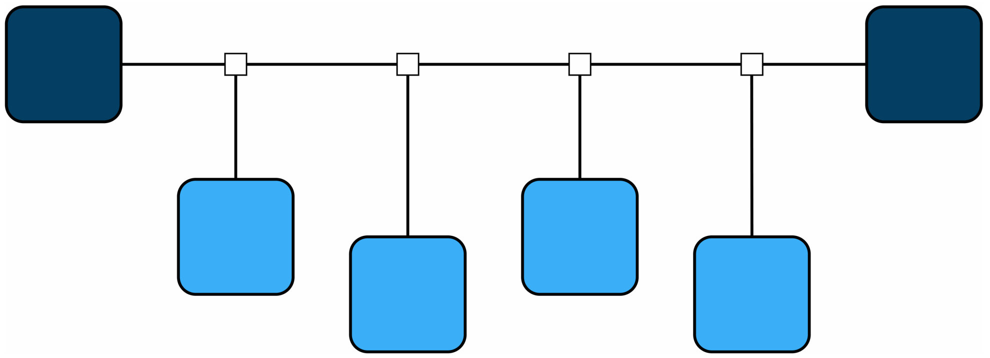
- Point to point



- Daisy chain



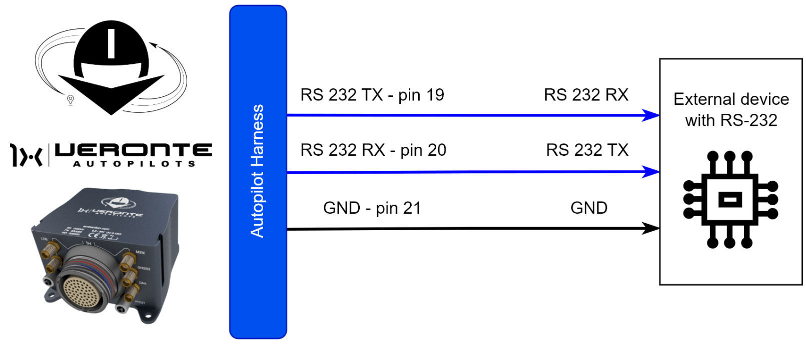
- Backbone with stubs



RS232

Point to point

This connection is recommended to establish with the computer while **1x** is commanding via CAN to Veronte MC110 or Veronte MC24, since the USB connection between the PC and the **1x** may be lost.



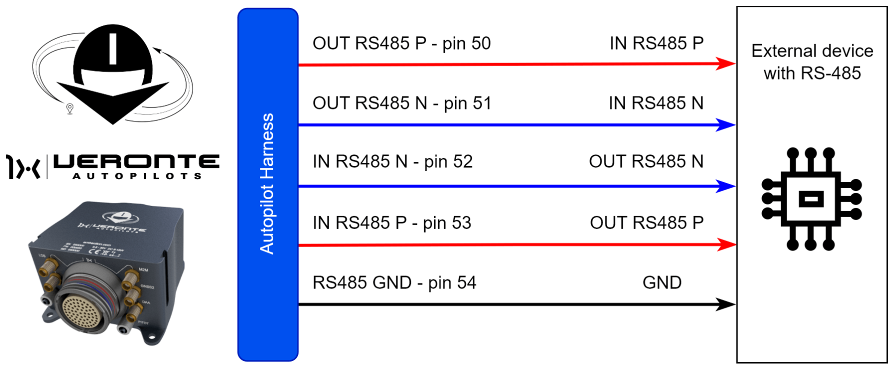
RS-232 connection

> **Note**
>
> Transmitter pin (TX) of one device is connected to the receiver pin (RX) of the other one, and vice versa.

RS485/422

Point to point

This connection is recommended to establish with the computer while **1x** is commanding via CAN to Veronte MC110 or Veronte MC24, since the USB connection between the PC and the **1x** may be lost.



RS-485 connection

> **Note**
>
> Output pin (OUT) of one device is connected to the input pin (IN) of the other one.
>
> Inverted signals (N) are connected each other; in the same way non-inverted signals (P) are linked each other.

Daisy chain

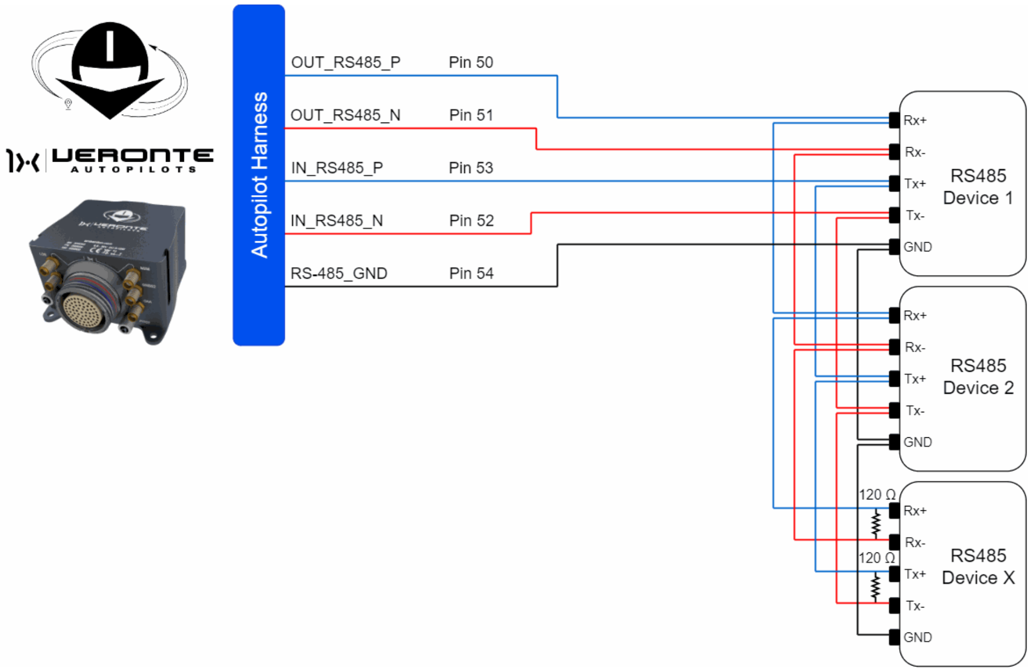
This section details the technical considerations that must be taken into account when connecting the Autopilot 1x to a chain of devices via RS485/422.

Full duplex

Autopilot 1x includes an internal resistor of 120 Ω. A second resistor is required at the end of the line (again 120 Ω) to allow the connection of multiple devices to the same line. This resistor may be placed on cable or PCB.

Full Duplex allows devices on an RS485 network to transmit and receive data **simultaneously**, enabling continuous bidirectional communication. This mode uses **four wires**, two for sending and two for receiving, which facilitates faster and more efficient data transfer.

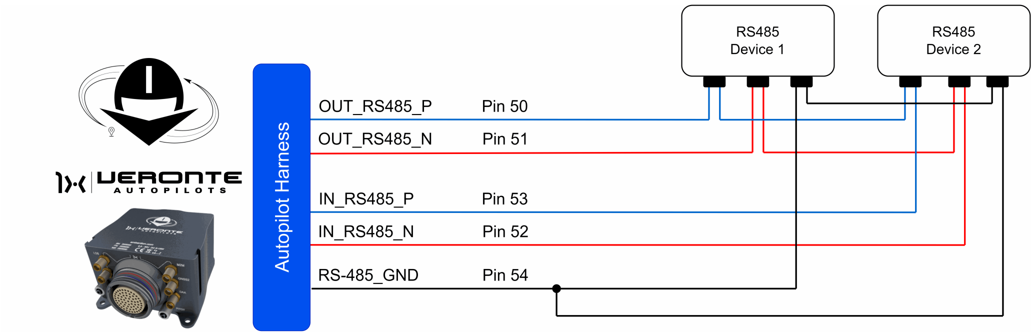
The following diagram shows how to connect the devices:

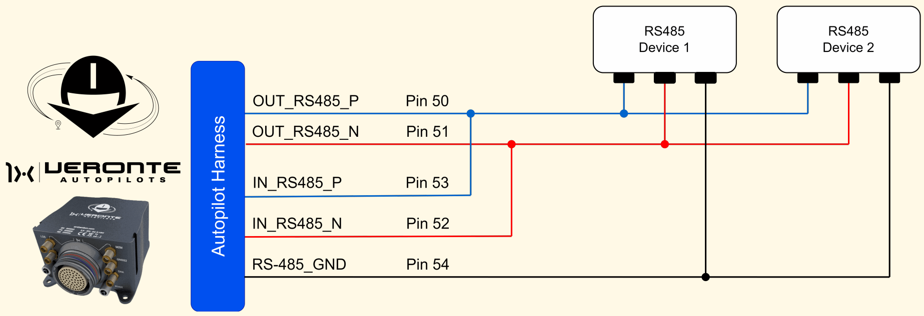


Half duplex

Half Duplex allows devices on an RS485 network to transmit and receive data, but **not at the same time**. In this mode, communication alternates between sending and receiving, using only **two wires** for both directions.

The following diagram shows how to connect the devices:

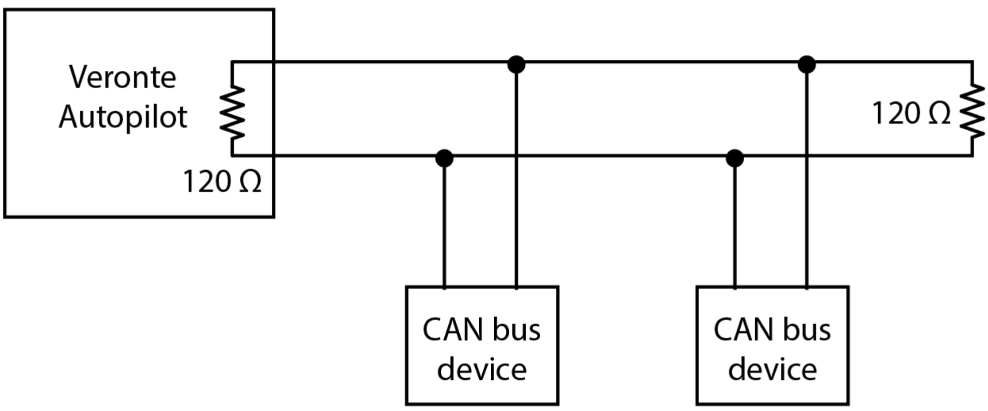


> ⚠ **Warning**
>
> The following mode of connection should be **avoided**.
>
> 
>

CAN

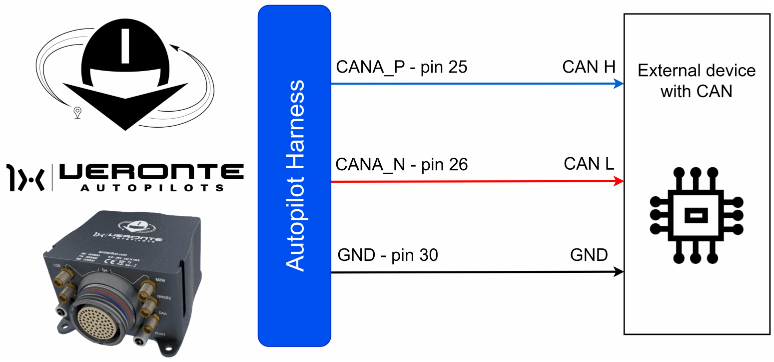
Electrical diagram of CAN bus

Autopilot 1x includes an internal resistor of 120 Ω. A second resistor is required at the end of the line (again 120 Ω) to allow the connection of multiple CAN Bus devices to the same line. This resistor may be placed on cable or PCB.



CAN assembly example diagram

Point to point



CAN connection

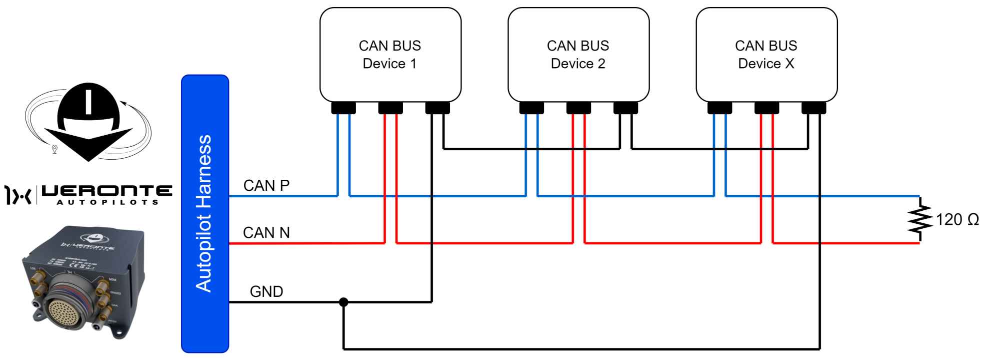
> ⓘ **Note**
>
> The user has the option to configure either of the two available CAN BUS lines on the Autopilot 1x: **CAN A** or **CAN B**.

The following sections detail the technical considerations to be taken into account when connecting the Autopilot 1x to a chain of devices via CAN.

Daisy chain

The daisy chain connection is the most common and recommended method for interconnecting multiple devices in a CAN system. This procedure involves sequentially connecting the CAN cable from one device to the next, ensuring that the total cable length is minimized to optimize network performance and reduce potential interference.

The following diagram illustrates an example of how to connect devices using this method:



Important

A resistor has been included in the diagram, which allows the connection of more devices, as explained in the Electrical diagram of CAN bus section.

Note

The standard designation for CAN lines in devices is typically **CAN H (High)** and **CAN L (Low)**. These designations correspond to those of the Autopilot 1x, where **CAN P (Positive)** and **CAN N (Negative)** represent the **High** and **Low** lines, respectively.

Note

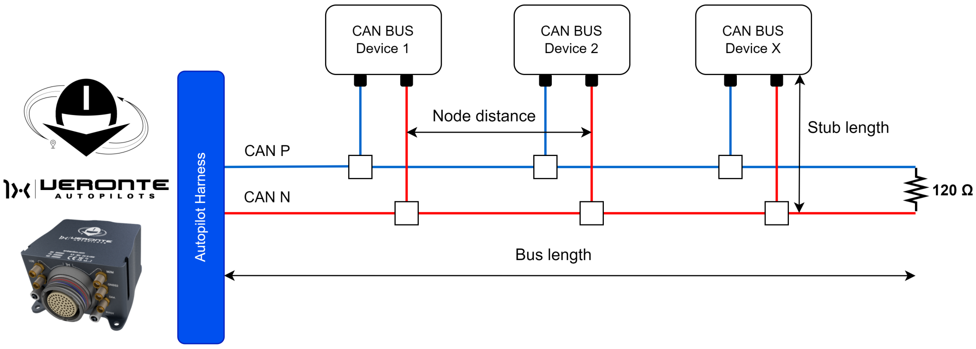
The user has the option to configure either of the two available CAN BUS lines on the Autopilot 1x: **CAN A** or **CAN B**.

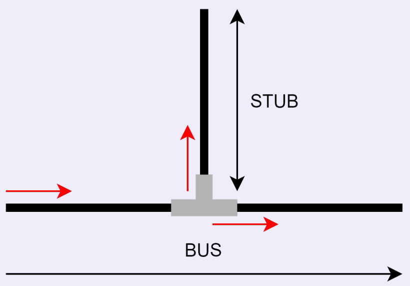
Backbone with stubs

Another method for connecting devices via CAN is to use a master cable from which the necessary stubs will be made to connect the devices. This connection method allows the creation of stubs from the master cable to integrate each device into the CAN network.

It is important for the user to note that when connecting devices in this topology, the total length of the CAN BUS cable is limited. Exceeding this length may adversaly affect network performance and communication integrity.

The following diagram illustrates an example of how to connect devices using this method:



> **Important**
>
> To create stub branches from the master line, **T-connectors** or **CAN splitter** are required.
>
> These connectors enable the connection of a device while allowing the main BUS line to continue, facilitating the connection of additional devices.
>
> 
>

The CAN protocol specifications limit the distance a device can be placed from the BUS. The following table shows the distance limitations to be considered when setting up the system.

Bus Speed	Bus Length	Stub Length	Node Distance
1 Mbit/Sec	40 meters	0.3 meters	40 meters
500 kbits/Sec	100 meters	0.3 meters	100 meters
100 kbits/Sec	500 meters	0.3 meters	500 meters

Bus Speed	Bus Length	Stub Length	Node Distance
50 kbits/ Sec	1000 meters	0.3 meters	1000 meters

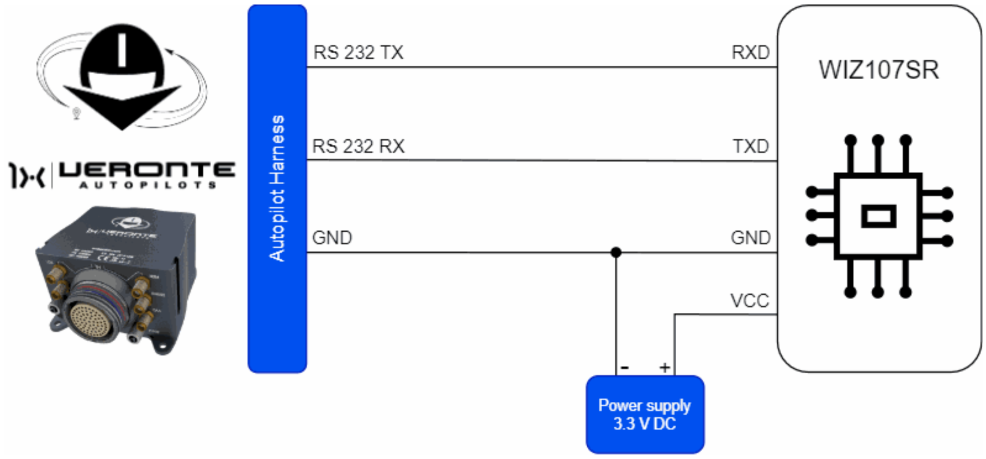
> ⚠ **Warning**
>
> If a cable stub (un-terminated cable) or a T-connector is used to connect to the bus line, then the stub distance should not exceed **0.3 meters**.

Serial to Ethernet Converter

This section provides the process to follow to integrate a **Serial to Ethernet Converter**, with **Veronte Autopilot 1x**. In this example, **WIZ108SR** and **WIZ107SR** converters are used.

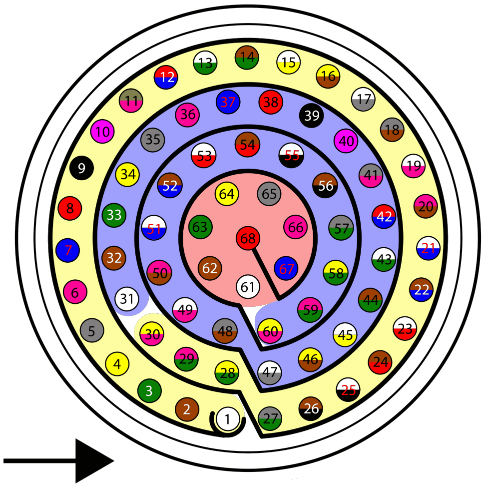
These protocol converters transmit the data sent by serial equipment as UDP data type, and converts back the UDP data received through the network into serial data to transmit back to the equipment.

- **WIZ107SR** connection diagram:



WIZ107SR - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, use an **external power supply** and **not to connect** it to the **3.3** line of **Autopilot 1x**. Please note that it **shares signal ground** with **Autopilot 1x**.

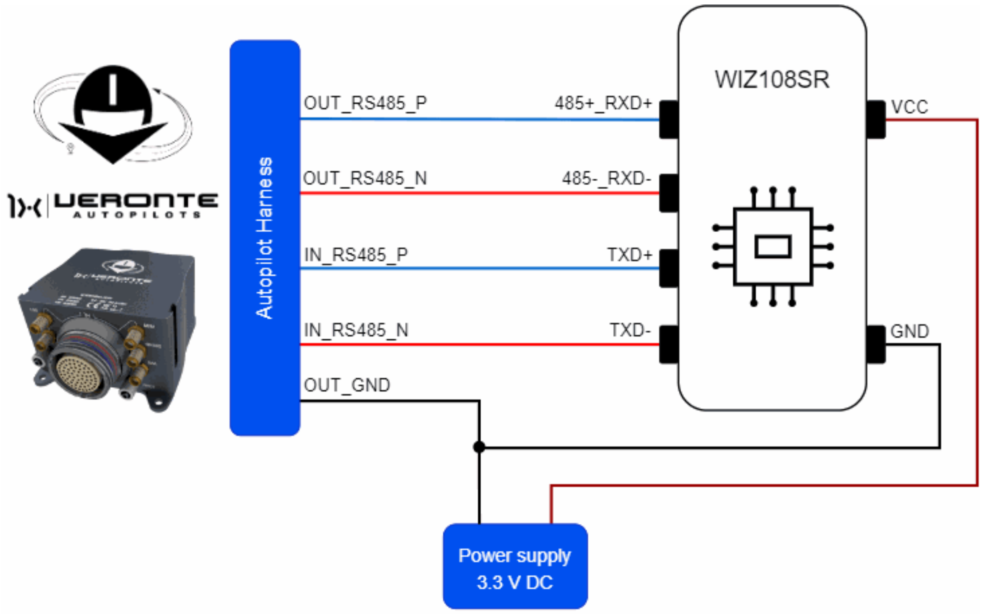


Autopilot 1x harness pinout

Autopilot 1x Harness			WIZ107SR Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
19	RS 232 TX	White-Pink	3	RXD
20	RS 232 RX	Pink-Brown	7	TXD
21	GND	White-Blue	11 / 12	GND

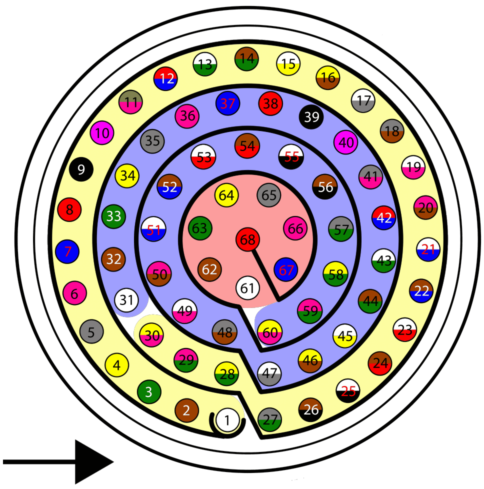
> **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

- **WIZ108SR** connection diagram:



WIZ108SR - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, use an **external power supply** and **not to connect** it to the **3.3** line of **Autopilot 1x**. Please note that it **shares signal ground** with **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness			WIZ108SR Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
50	OUT_RS485_P	Pink-Brown	3	485+_RXD+
51	OUT_RS485_N	White-Blue	5	485-_RXD-
53	IN_RS485_P	White-Red	7	TXD+
52	IN_RS485_N	Brown-Blue	9	TXD-
54	OUT_GND	Brown-Red	11 / 12	GND

> ⚠ **Warning**
>
> Note that this pin 54 is not a common GND pin.

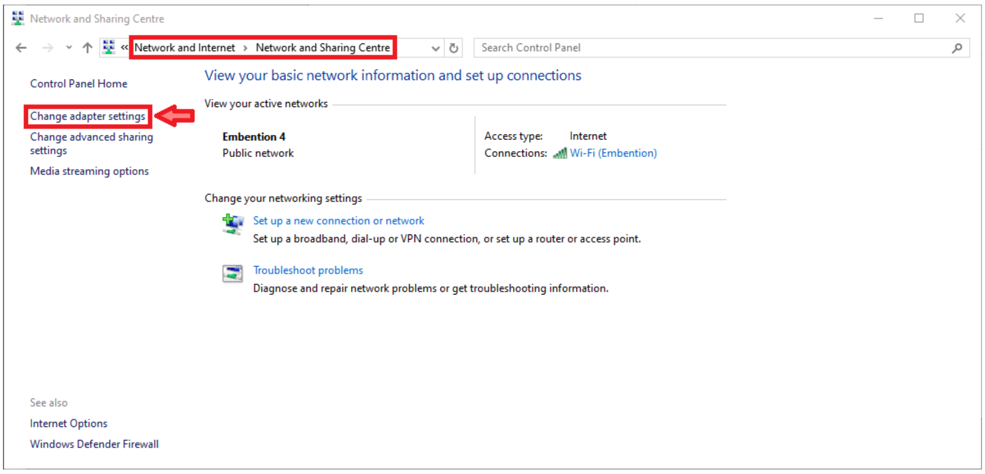
> ⓘ **Note**
>
> This integration is done for the **WIZ108SR**, this is for a connection via **RS485** with **Autopilot 1x**. However, the process for the **WIZ107SR** device is almost the same.

Once the hardware installation is complete, follow the steps below to properly configure the connection and ensure stable communication between the devices.

Ethernet Connection in Windows

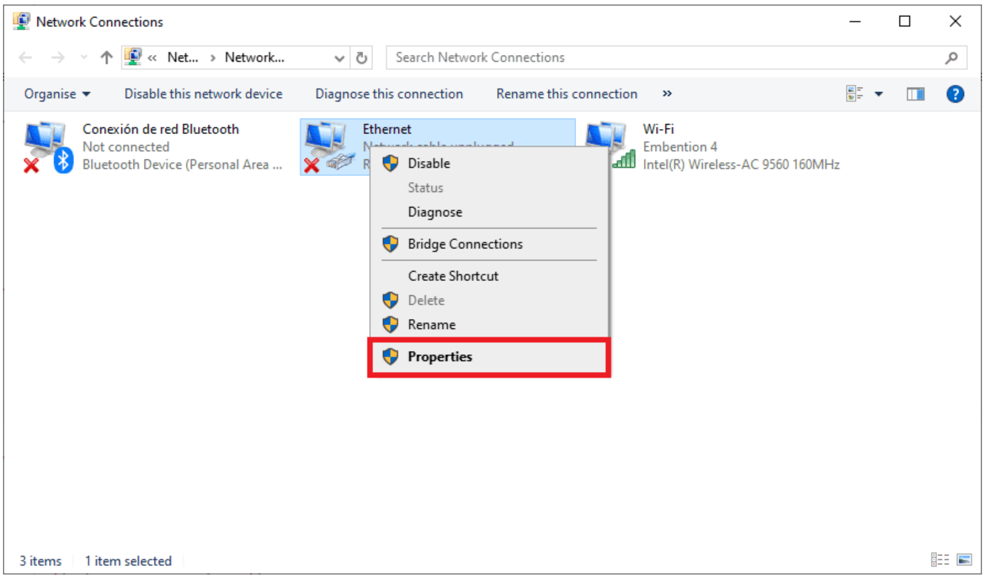
First, make sure computer is set to static IP address on same subnet as radio. The following substeps clarify how to set the IP address:

1. Connect the Ethernet cable of the WIZNet adapter, powered by 3.3V, to the Ethernet port of the PC.
2. Open **Network and Sharing Centre** menu and click **Change adapter settings**.



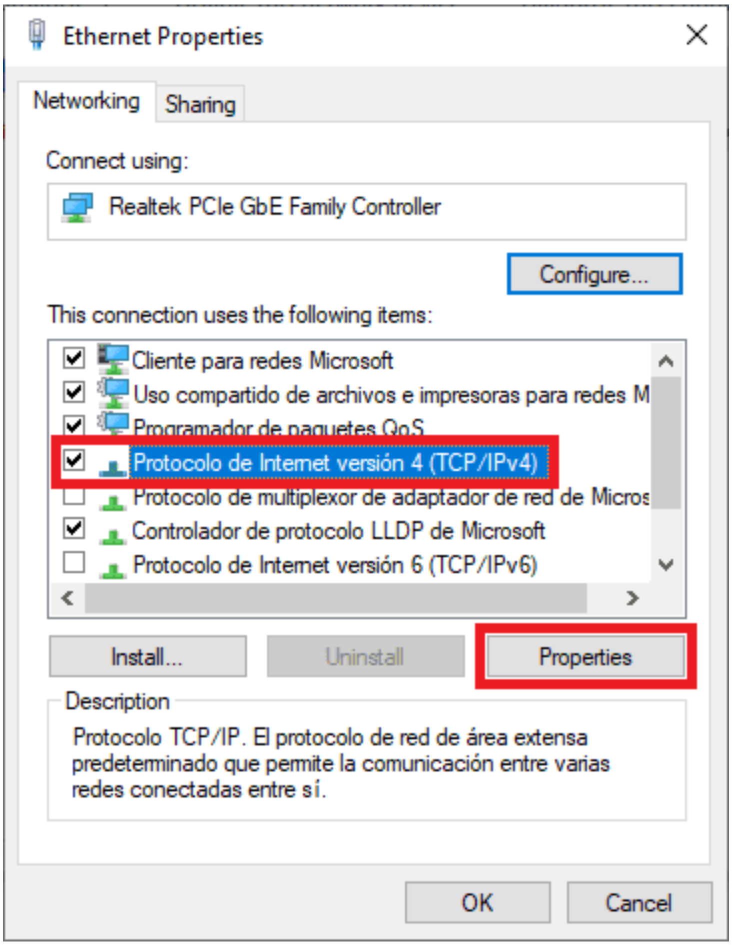
Ethernet connection 1

3. Select **Local Area Connection**, right click, and select **Properties**.



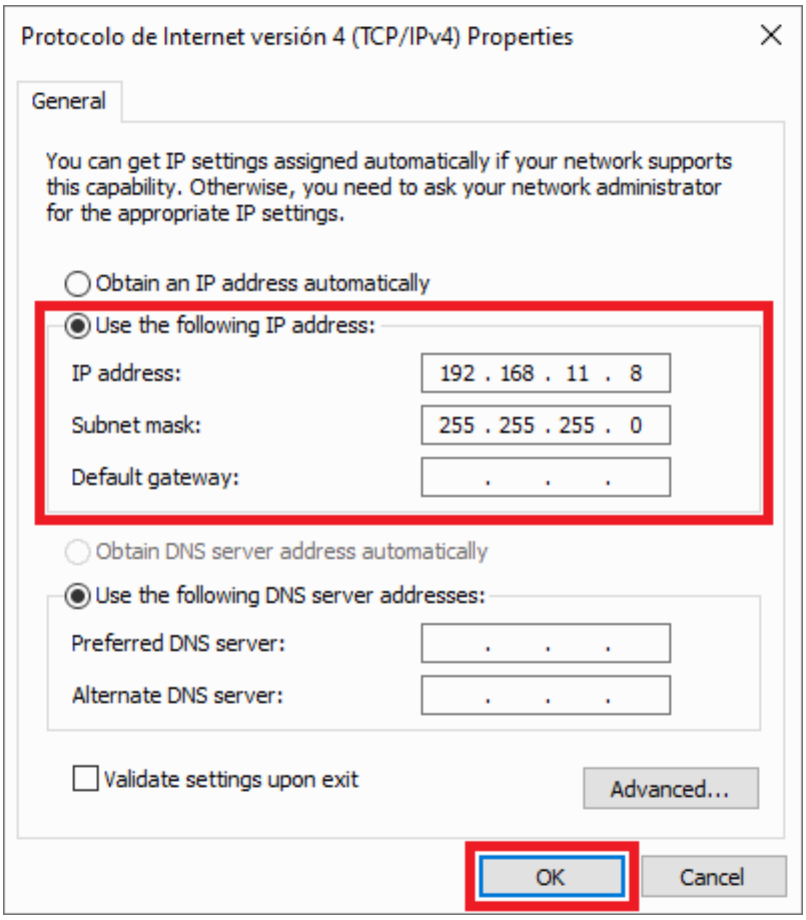
Ethernet connection 2

4. Select **IPv4** and click **Properties**.



Ethernet connection 3

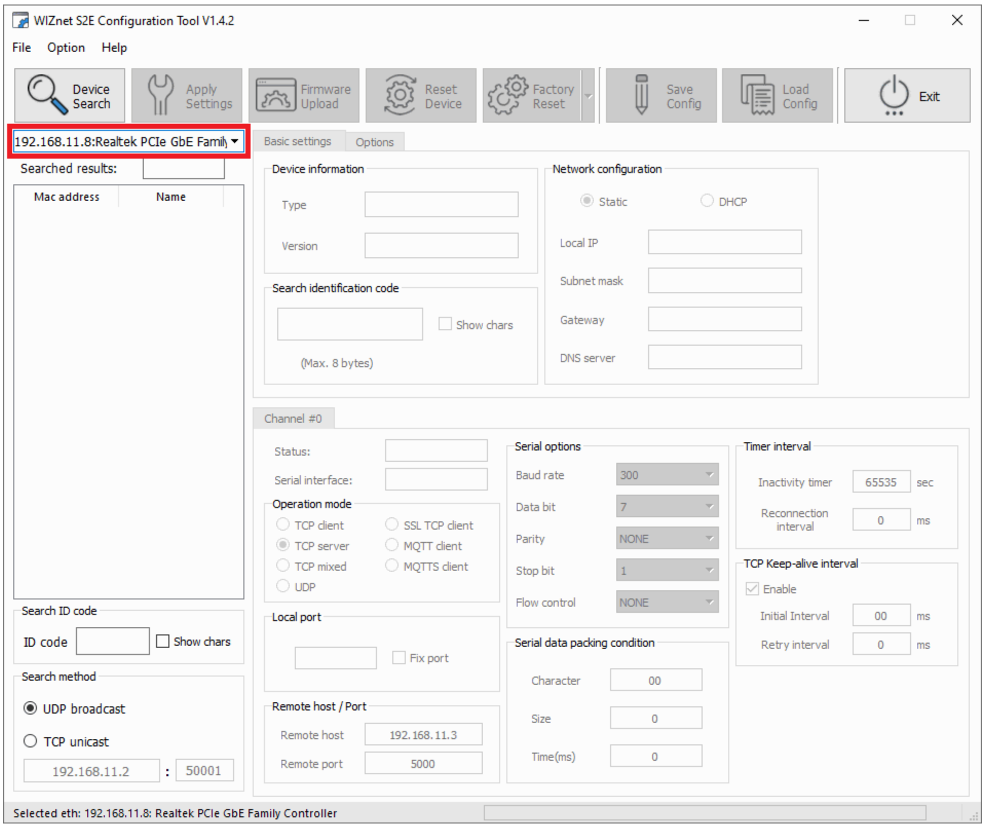
5. Set **IP address** to 192.168.11.X (in this example `192.168.11.8` is entered) and **Subnet mask** to `255.255.255.0`. Click **OK**.



Ethernet connection 4

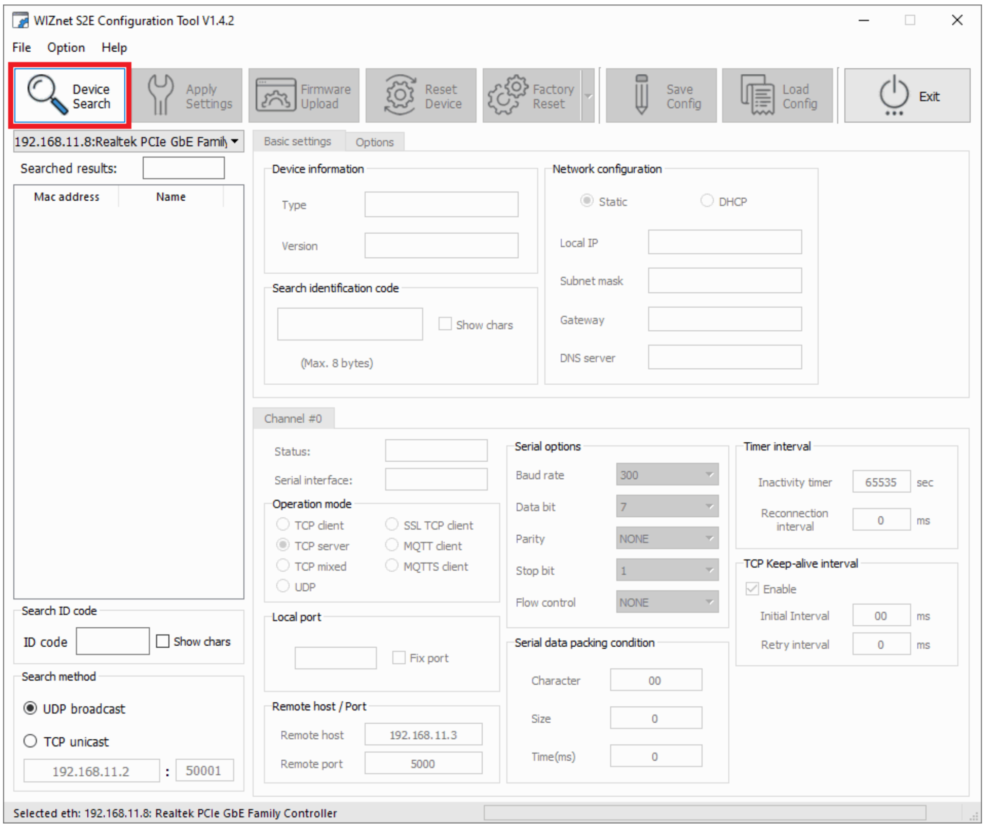
WIZNet software configuration

6. Open the **WIZNet Software** and select the Ethernet port with the IP assigned before in the dropdown menu.



WIZnet software - Ethernet port selected

7. Click on the **Device Search** option to **scan** for the device.



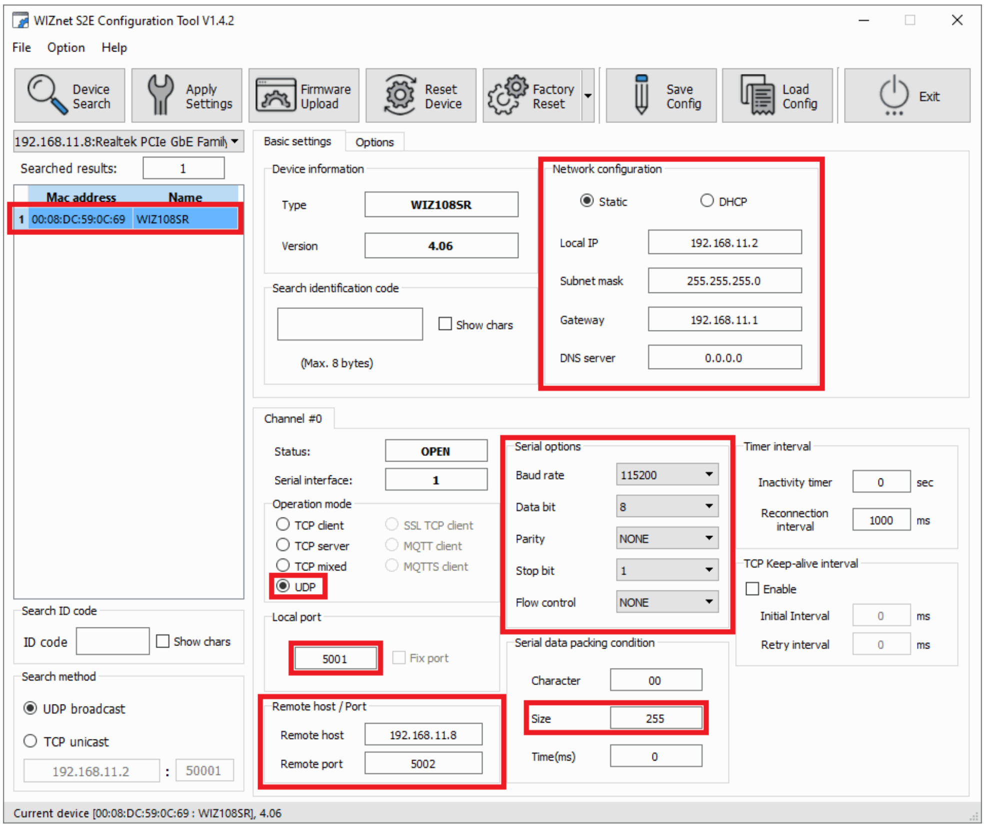
WIZnet software - Scan for device

8. Then, if the module is properly connected to the network, it will appear in the search list. Select the **WIZ108SR** device to access its configuration.

> ⚠ **Caution**
>
> If the module does not appear in the list and it has been previously connected to another PC, **reboot** the module (simply unpower it and power it up again).
>
> This is because when the **WIZ108SR** module establishes a connection to a PC's MAC address, it will only communicate with that PC unless it is rebooted.

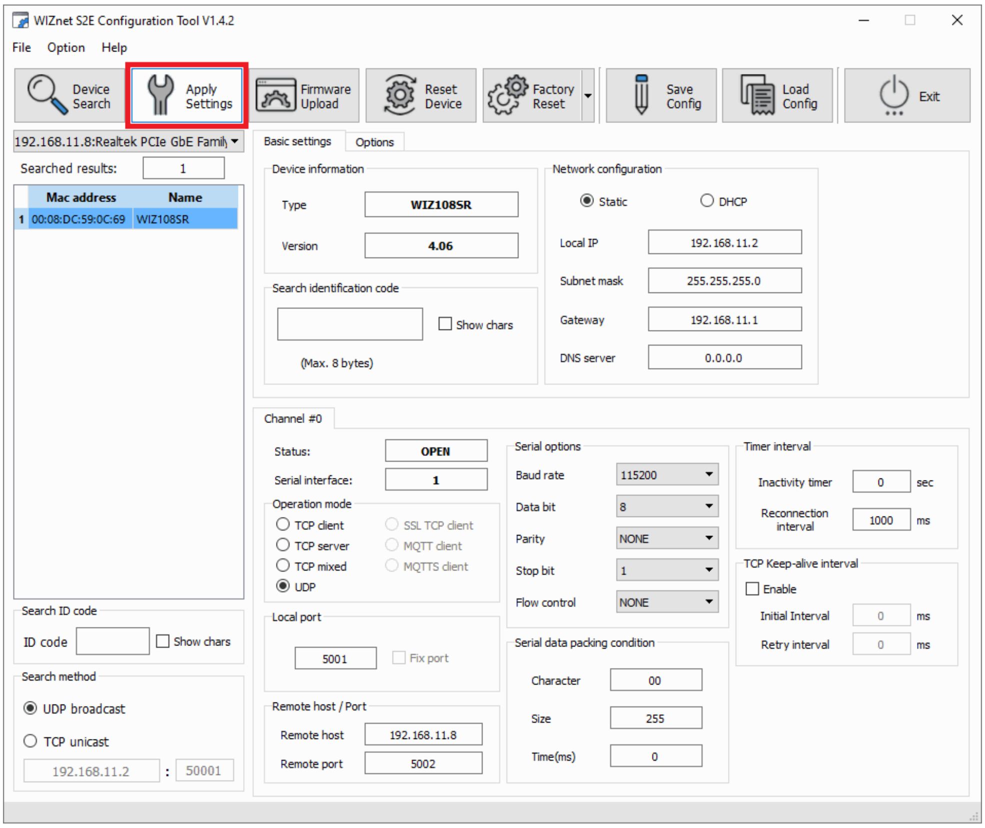
   The configuration must be as follows:

   - In the **Basic Settings** tab, make sure that the Network Settings parameters are:
     - Local IP: `192.168.11.2`
     - Subnet Mask: `255.255.255.0`
     - Gateway: `192.168.11.1`
   - In the **Channel #0** tab:
     - Change the Operation mode to **UDP**.
     - Set the Local port to `5001` (for example).
     - Set the Remote host to the **adapter IP** previously configured in step 5 and a Remote Port, in this example port `5002` has been chosen.
     - The Serial options must match those of the **Autopilot 1x**, so set them as follows:
       - Baud rate: `115200`
       - Data bit: `8`
       - Parity: `NONE`
       - Stop bit: `1`
     - Modify in the Serial data packing condition the **size** to `255` to send the data in 255 byte packets, so that each send will have more data.



WIZnet software - Device configuration

9. Finally, click **Apply Settings** to save the configuration to the device.



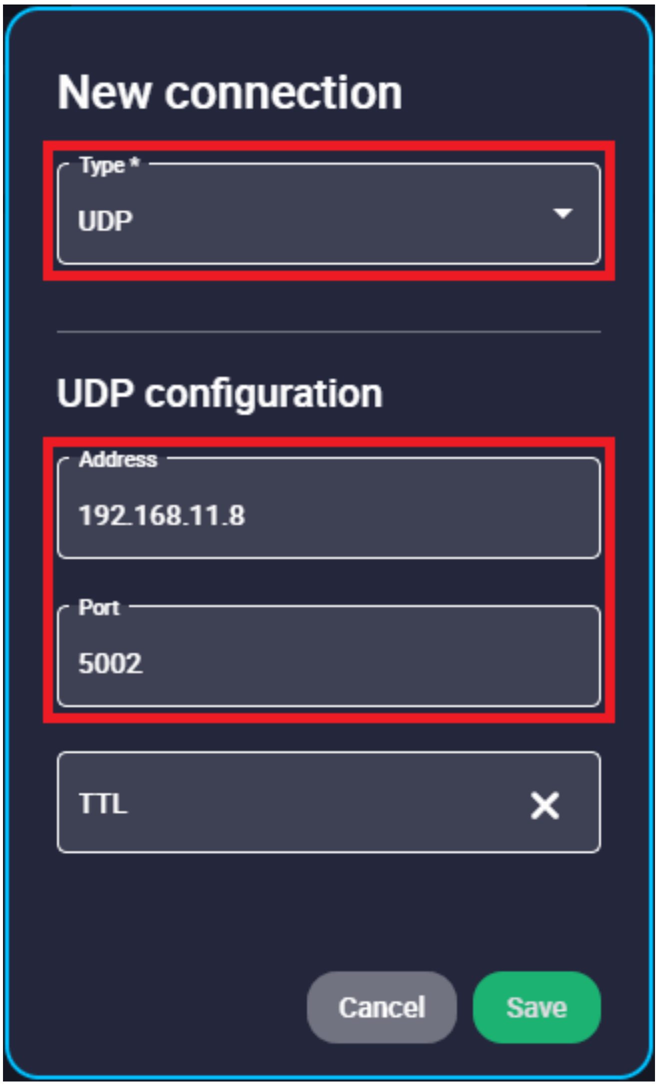
WIZnet software - Apply Settings

Veronte Link configuration

10. Add a new **UDP connection** in **Veronte Link** and configure it with the same **IP address and port** as previously set in the Remote options of the **WIZNet settings**.

    In this example:

    - Address: `192.168.11.1`
    - Port: `5002`



Veronte Link - Connection settings

For more information on Veronte Link connections, please refer to the UDP connection - Integration examples section of the **Veronte Link** user manual.

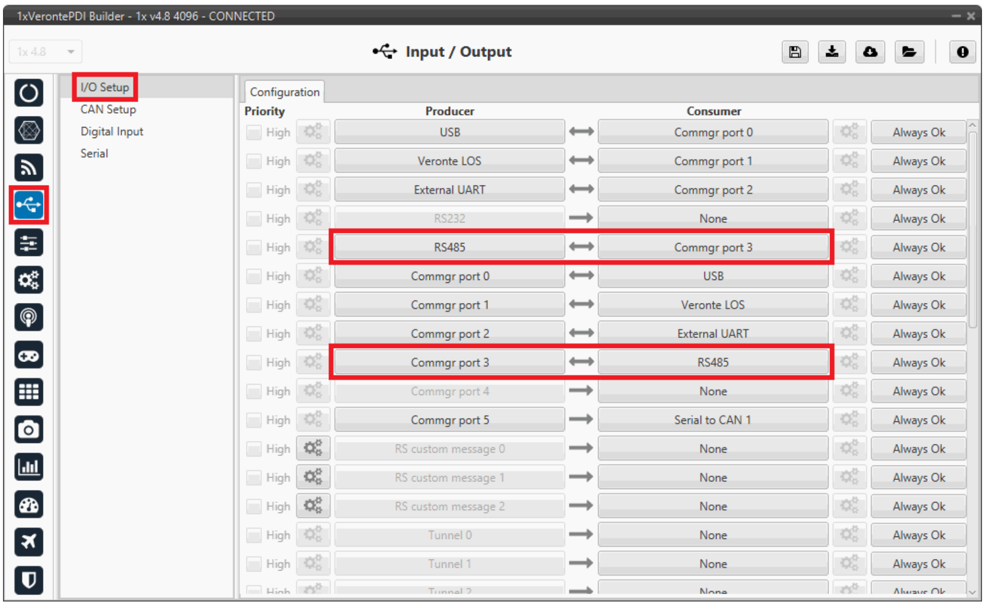
If the connection fails, please check UDP Failed Connection - Troubleshooting section of this manual.

1x PDI Builder configuration

11. Go to Input/Output menu → **I/O Setup panel**.

    Bidirectionally connect the **RS485 Producer** to a **Commgr port Consumer**, in this example it has been linked to Commgr port 3. Then, the same **Commgr port Producer**, in this case Commgr port 3, should be automatically connected to the **RS485 Consumer**:

    > ⓘ **Note**
    >
    > If the **WIZ107SR** module is used, i.e. a connection is established via **RS232**, bidirectionally connect the **RS232 port** to a **Commgr port** and not the RS485 port.



1x PDI Builder - I/O Setup configuration

12. Go to Input/Output menu → Serial panel → **RS485 tab**.

    > ⓘ **Note**
    >
    > If using the **WIZ107SR** module, check the **RS232** parameters instead.

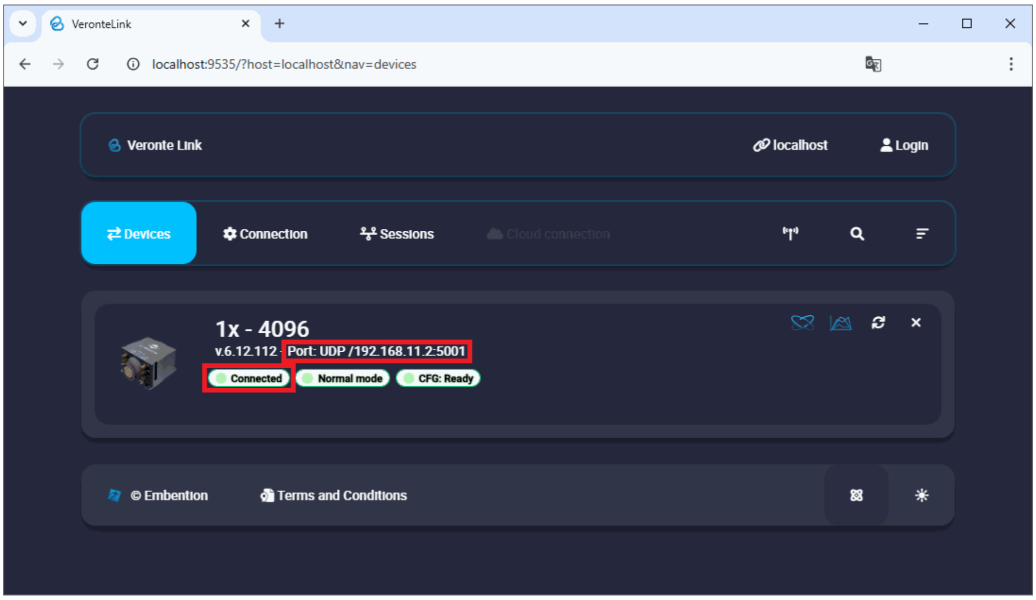
    Make sure that these parameters are the same as the parameter values previously set in the **WIZNet** software.

    - Baudrate: **115200**
    - Length: **8**
    - Stop: **1**

- Parity: **Disabled**

1x PDI Builder - Serial configuration

Following these steps, **Autopilot 1x** should appear in Veronte Link as follows:



Autopilot 1x connection via UDP

External devices

The step-by-step instructions for the following external devices will be explained in detail in the following sections:

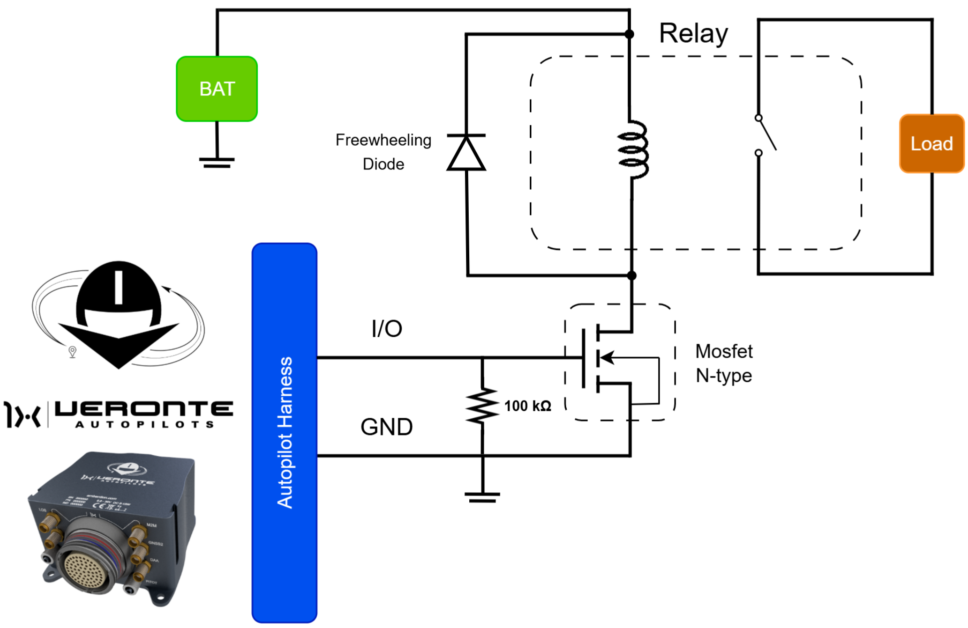
- Connection Examples
- Actuators/Servos
- Air Data Sensors
- Altimeters
- Datalinks
- GNSS Receivers
- IMUs & Compass
- RPM Sensors
- Stick
- Veronte products

GPIO Wiring for External Loads

Besides their use in communication buses, the I/O pins are frequently used as GPIOs to control external devices like relays, lights, or solenoids.

It is important to note that each **I/O pin** has a **maximum current output** of **1.65 mA**, as they are designed for data signals, not for directly powering devices.

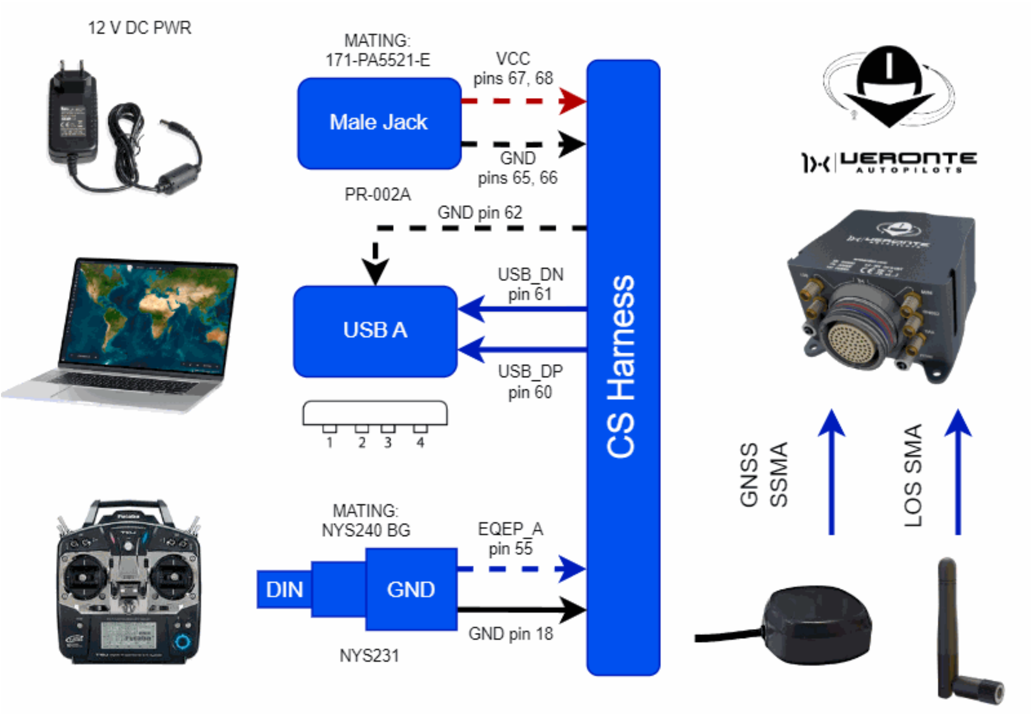
Due to this current limitation, an intermediate circuit is required to handle higher-power loads. The solution recommended is to use an N-type MOSFET as an electronic switch. The following diagram shows the recommended circuit for driving a relay:



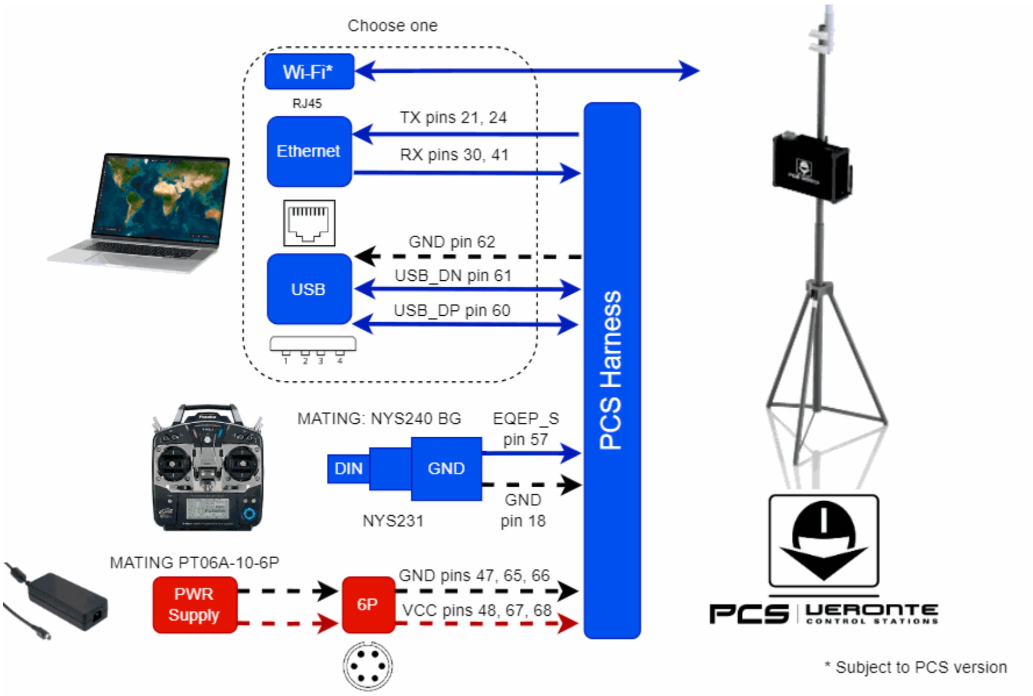
Circuit for driving a Relay

Connection Examples

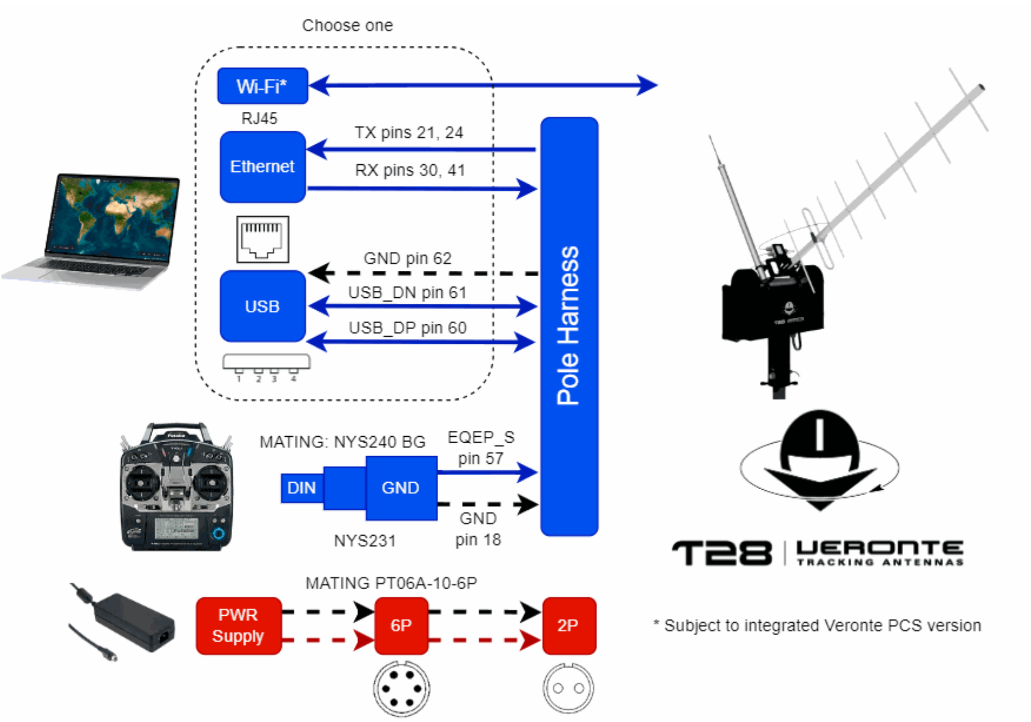
Ground Stations



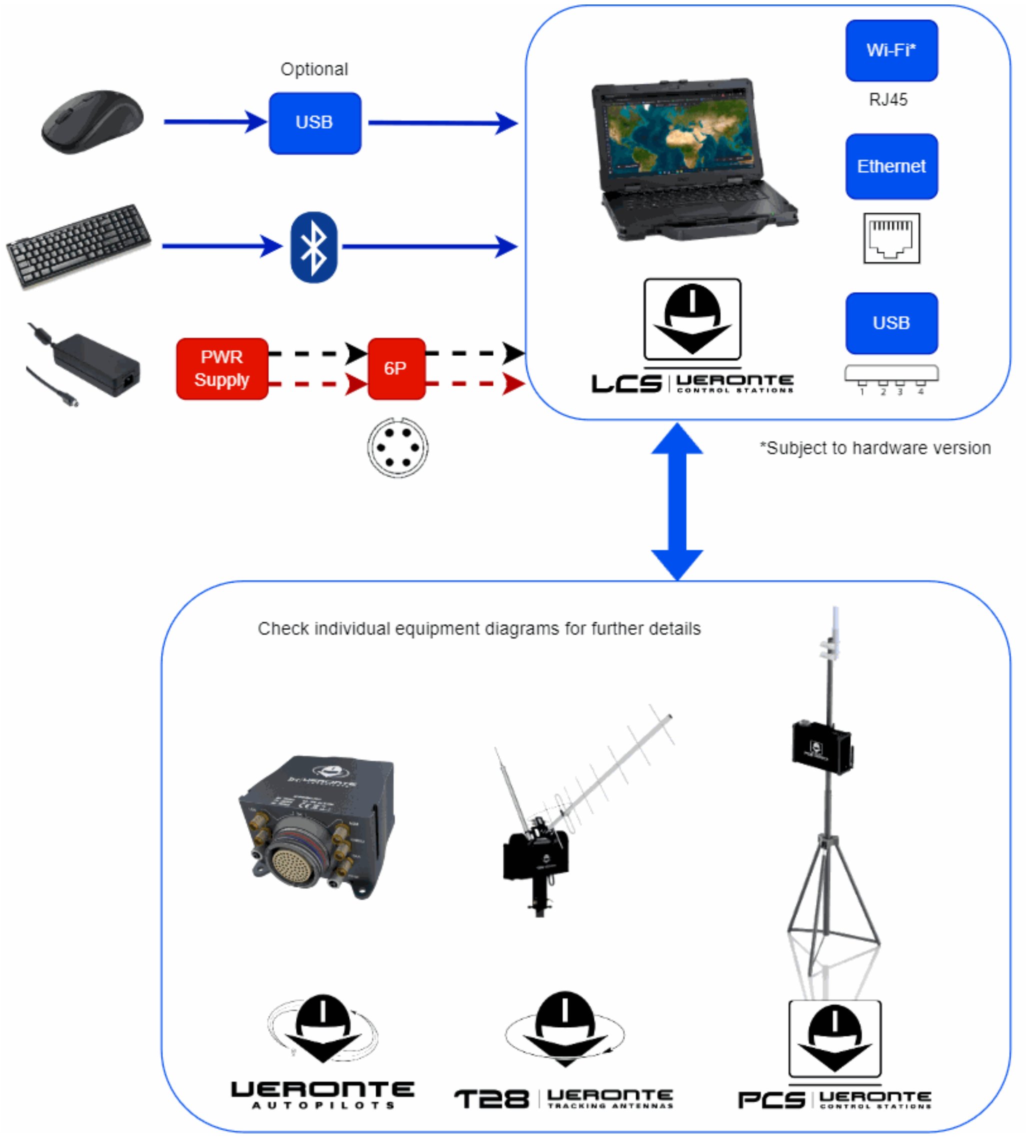
Basic Autopilot 1x Ground Station



Autopilot 1x PCS Ground Station



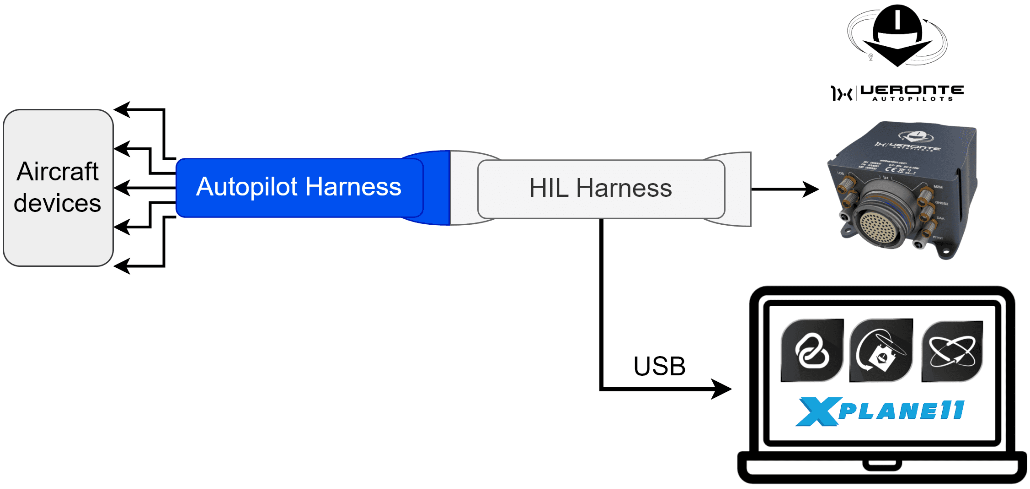
Autopilot 1x Tracker Ground Station



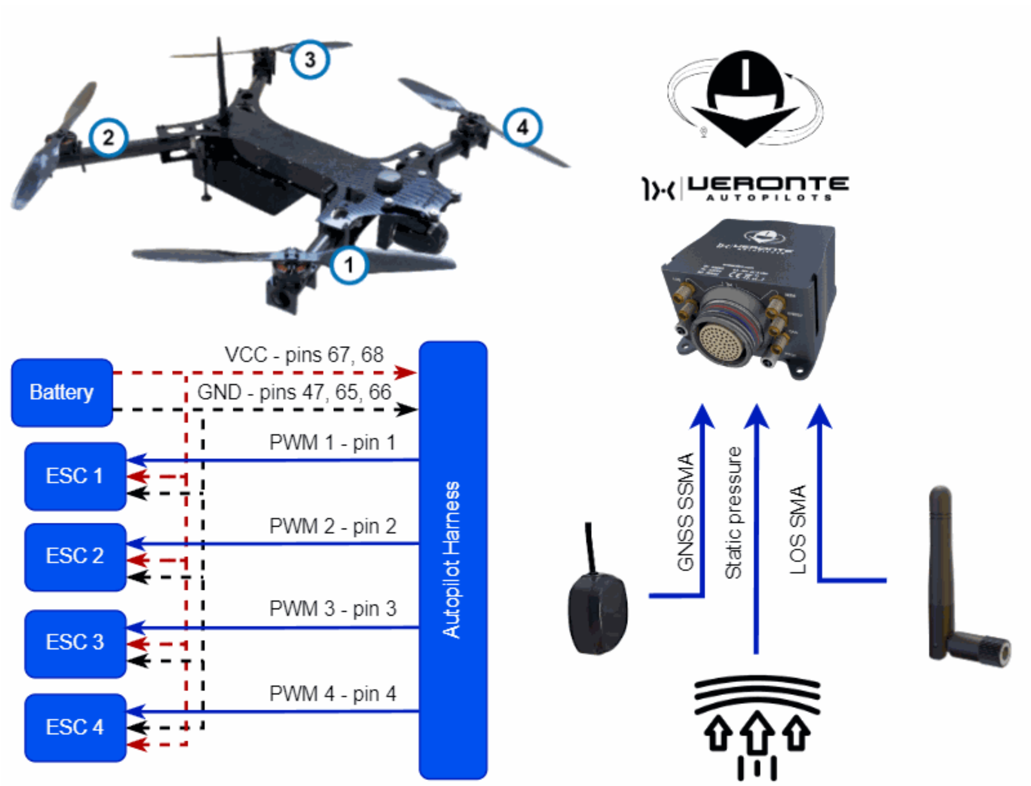
Autopilot 1x LCS Ground Station

> ⚠ **Warning**
>
> Veronte Autopilot 1x equipment harnesses have specific pin layouts. Only use their own matting connectors, do NOT mix harnesses: misuse can lead to destruction.

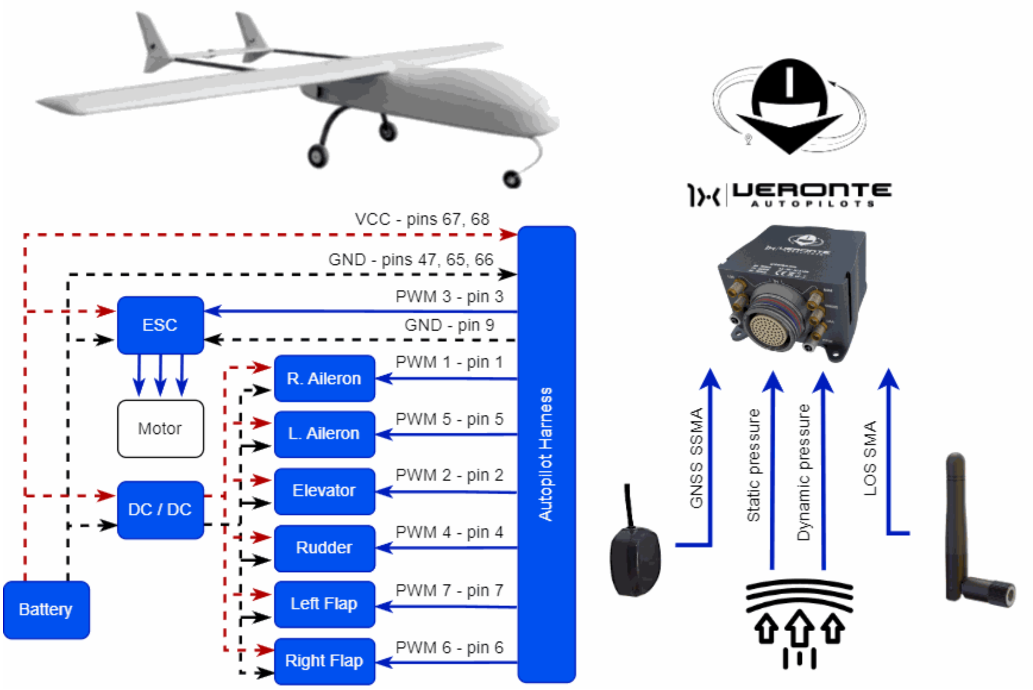
Aircrafts



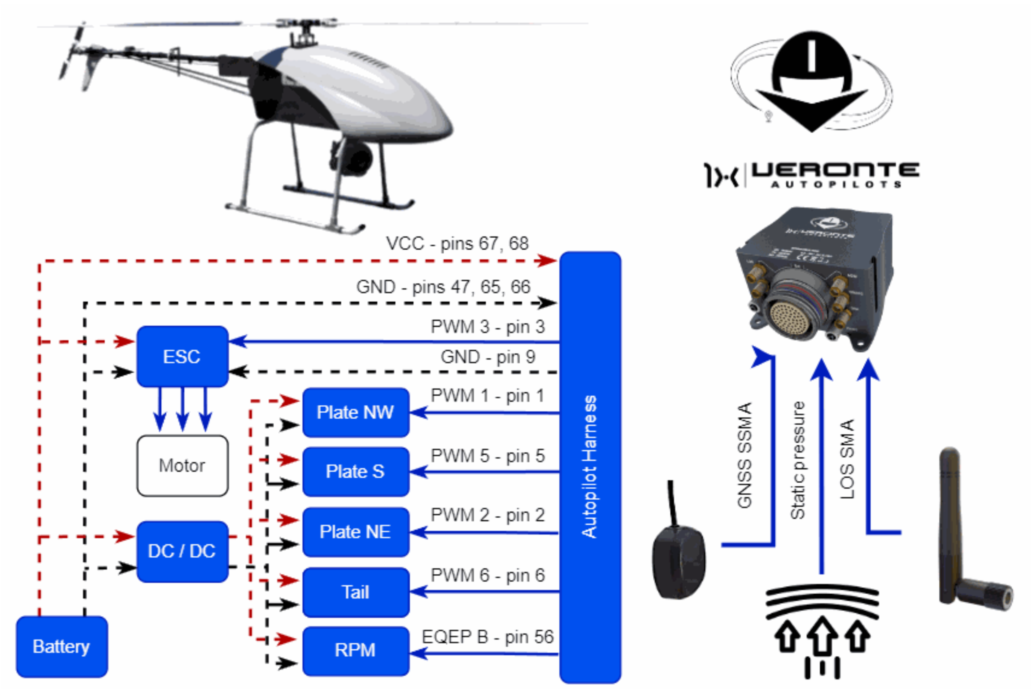
HIL Harness



Multicopter



Fixed Wing Airplane



Helicopter

Actuators/Servos

The user can configure any actuator compatible with the communication interfaces.

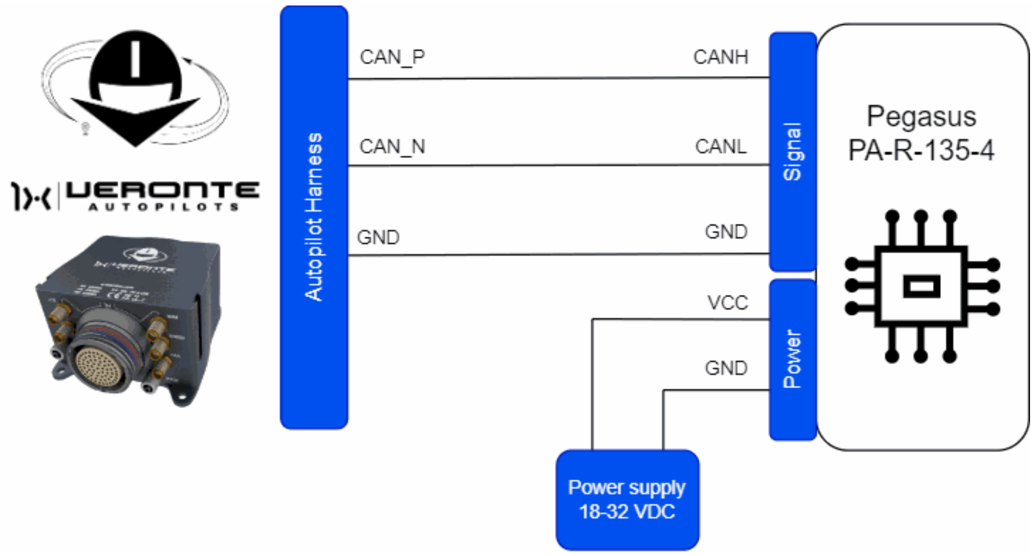
CAN

Pegasus PA-R-135-4

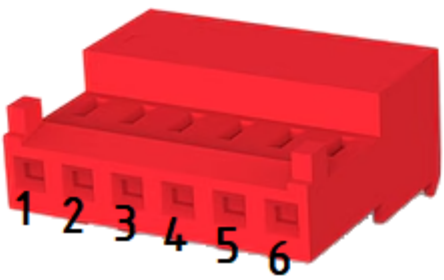
Pegasus PA-R-135-4 servo

Pegasus PA-R-135-4 micro-low-profile servo is currently the world's smallest professional servo-actuator of its class.

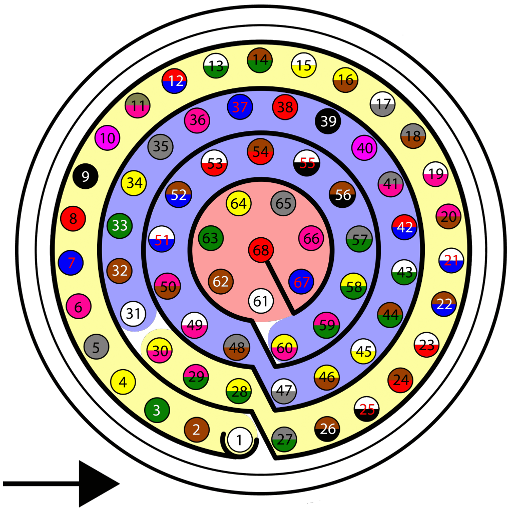
The setup to be performed by the user should be as follows:



Pegasus PA-R-135-4 - Autopilot 1x wiring diagram



Pegasus PA-R-135-4 Tyco MTA100 connector pinout



Autopilot 1x harness pinout

Autopilot 1x Harness			Pegasus PA-R-135-4 Tyco MTA100 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
25	CANA_P	White-Black	3	CANH
28	CANB_P	Yellow-Green		
26	CANA_N	Brown-Black	1	CANL
29	CANB_N	Pink-Green		
30	GND	Yellow-Pink	6	GND

ⓘ **Note**

CAN A and CAN B buses are equivalent and can be used interchangeably for the integration of this device.

For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the hardware integration of the device has been completed, it will be necessary to proceed with the **software integration**. To do this, follow the steps detailed in the Pegasus - Integration examples section of the **1x PDI Builder** user manual.

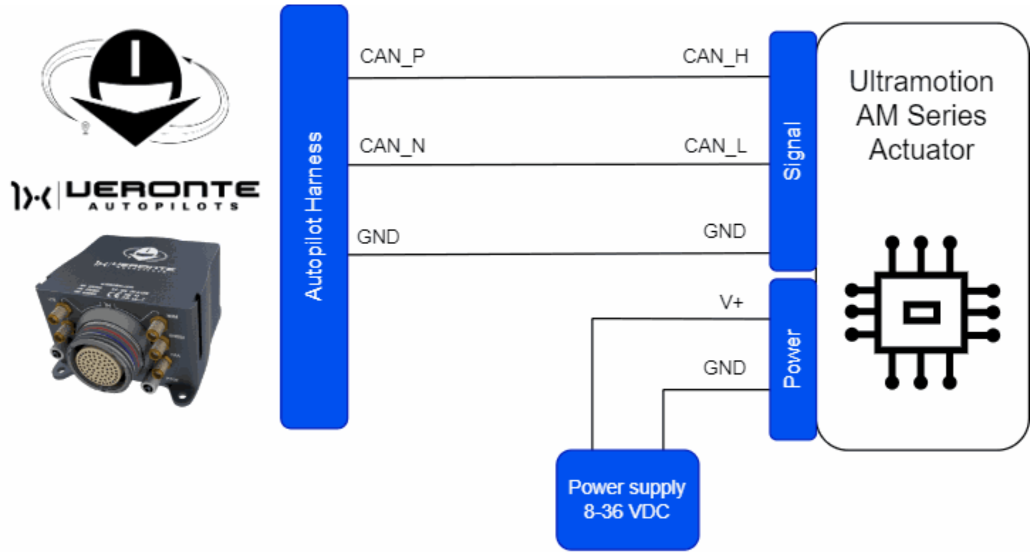
Ultra Motion

Ultra Motion servo

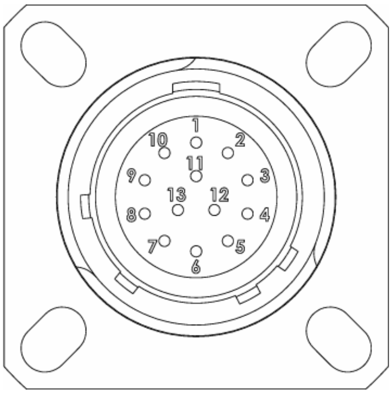
Ultra Motion servo is a high-precision actuator designed for demanding applications, with BLDC electronic control and non-contact absolute position feedback.

This device can be integrated with **Autopilot 1x** via **CAN 2.0B protocol** to ensure robust and efficient communication in the system.

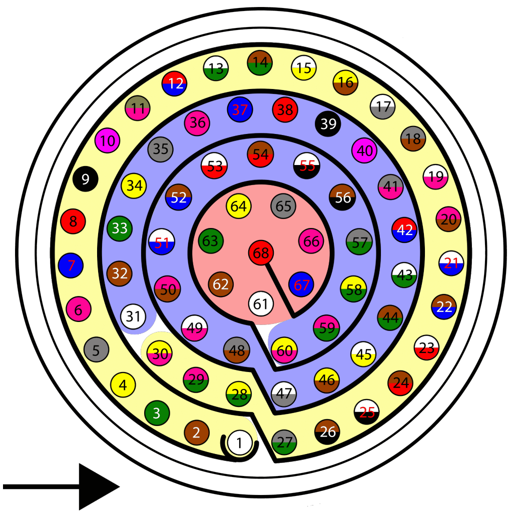
The setup to be performed by the user should be as follows:



Ultra Motion - Autopilot 1x wiring diagram



Ultra Motion - Pinouts for Signal connector on AM series actuators



Autopilot 1x harness pinout

Autopilot 1x Harness			Ultra Motion Signal Connector		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
25	CANA_P	White-Black	7	CAN_H	Red
28	CANB_P	Yellow-Green			
26	CANA_N	Brown-Black	9	CAN_L	White
29	CANB_N	Pink-Green			
30	GND	Yellow-Pink	8	GND	Black

ⓘ **Note**

CAN A and CAN B buses are equivalent and can be used interchangeably for the integration of this device.

For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

⚠ **Warning**

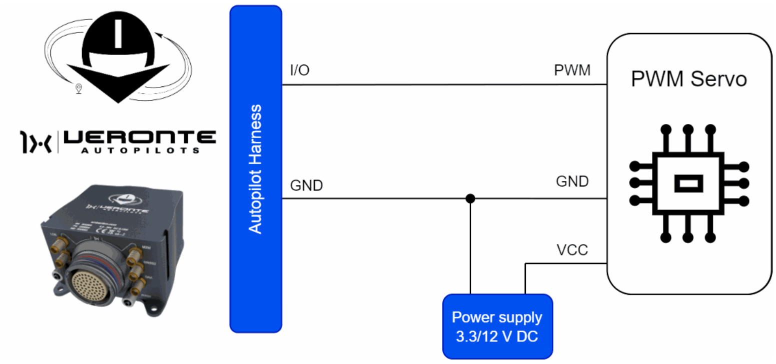
Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the hardware installation has been completed, the user can find the explanataion for the software installation in the Ultra Motion - Integration examples section of the **1x PDI Builder** user manual.

PWM

This section details the process of integrating a **PWM servomotor** with **Autopilot 1x**.

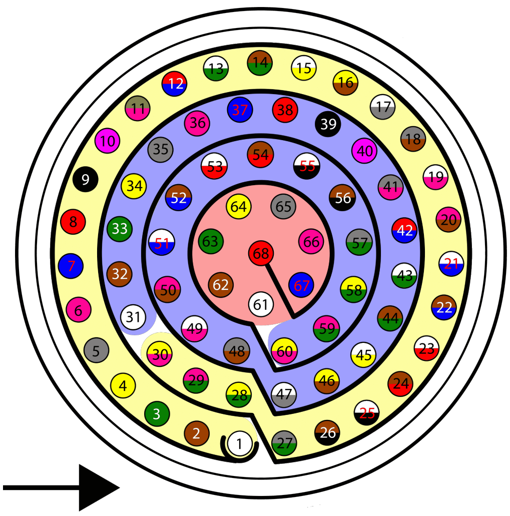
The connection diagram between the two devices should look like this:



PWM - Autopilot 1x wiring diagram

> **Important**
>
> Note that this servo must be connected to an **external power supply**, **sharing signal ground** with **Autopilot 1x**.

The **PWM servo** must be connected to one of the available I/O pins of **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness		
PIN	**Signal**	**Color Code**
1	I/O0	White
2	I/O1	Brown
3	I/O2	Green
4	I/O3	Yellow
5	I/O4	Gray
6	I/O5	Pink
7	I/O6	Blue
8	I/O7	Red
9	GND	Black
10	I/O8	Violet
11	I/O9	Gray-Pink
12	I/O10	Red-Blue
13	I/O11	White-Green
14	I/O12	Brown-Green
15	I/O13	White-Yellow
16	I/O14	Yellow-Brown
17	I/O15	White-Gray

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the hardware connection is made, it is necessary to **configure** the **I/O** pin used. Since these **I/O** pins are **preconfigured** as **GPIO**, they must be set as **PWM**.

To do so, refer to and follow the steps described in the PWM - Integration examples section of the **1x PDI Builder** user manual.

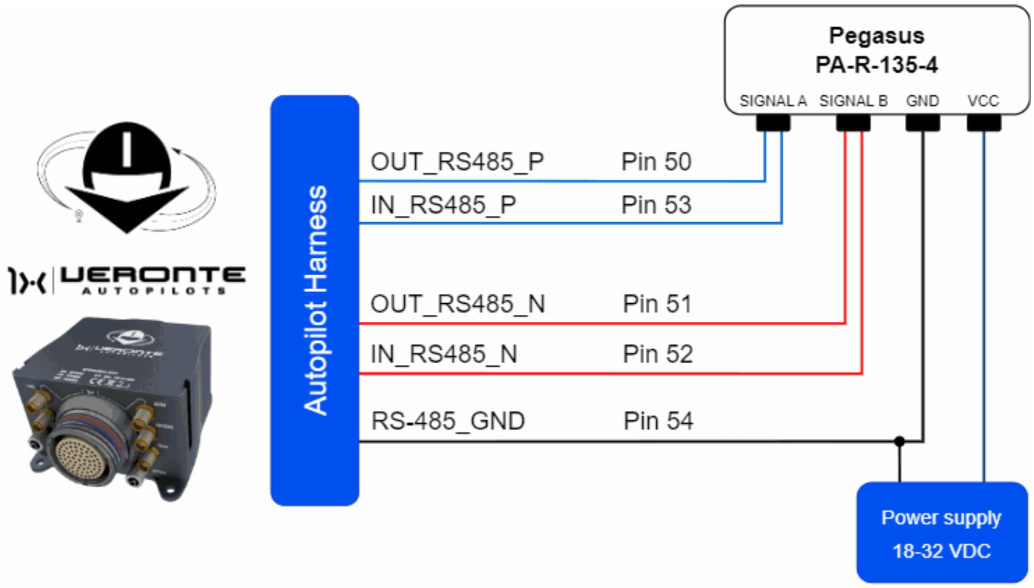
Serial

Pegasus PA-R-135-4

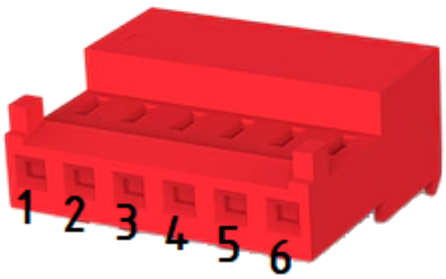
Pegasus PA-R-135-4 servo

Pegasus PA-R-135-4 micro-low-profile servo is currently the world's smallest professional servo-actuator of its class.

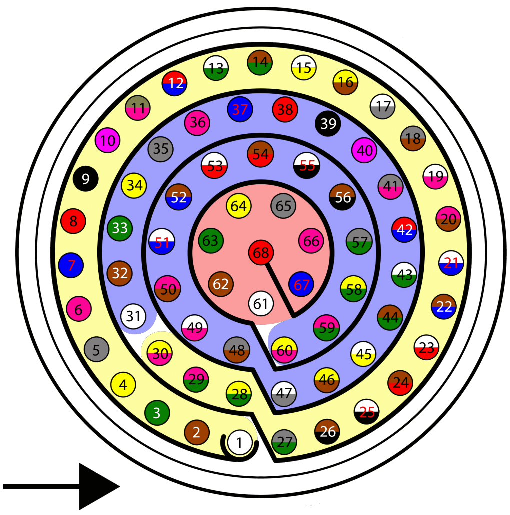
The setup to be performed by the user should be as follows:



Pegasus PA-R-135-4 - Autopilot 1x wiring diagram



Pegasus PA-R-135-4 Tyco MTA100 connector pinout

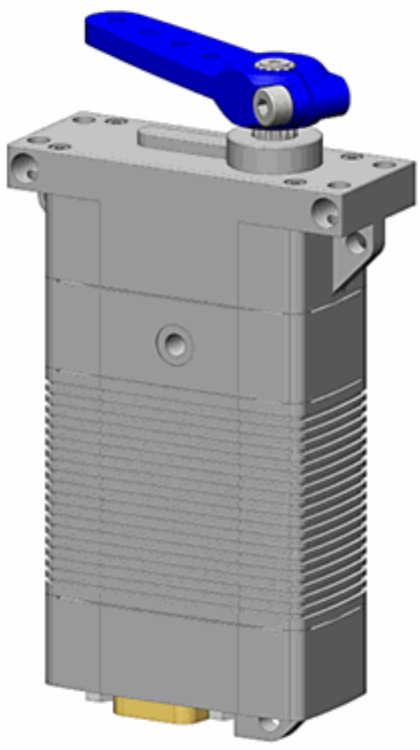


Autopilot 1x harness pinout

Autopilot 1x Harness			Pegasus PA-R-135-4 Tyco MTA100 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
50	OUT_RS485_P	Pink-Brown	3	SINGAL A
53	IN_RS485_P	White-Red		
51	OUT_RS485_N	White-Blue	1	SIGNAL B
52	IN_RS485_N	Brown-Blue		
54	OUT_GND	Brown-Red	6	GND

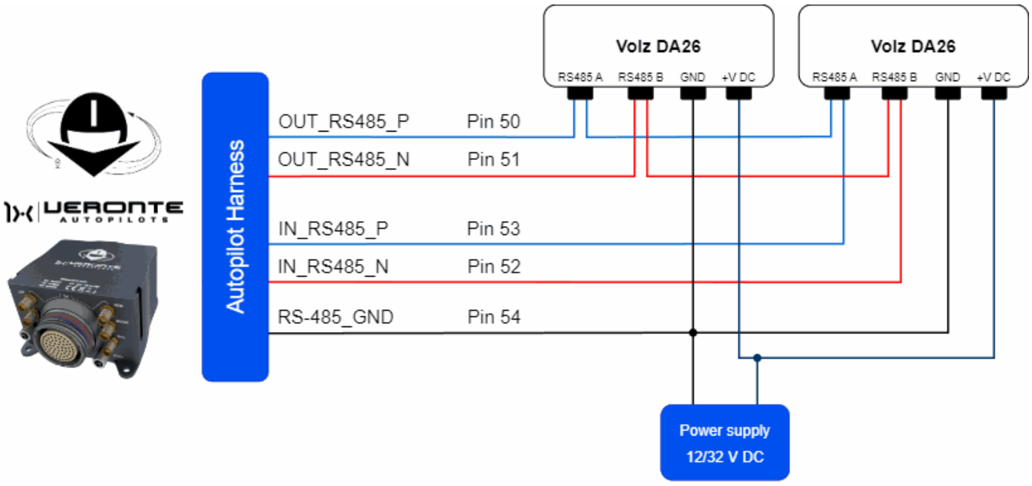
Once the hardware integration of the device has been completed, it will be necessary to proceed with the **software integration**. To do this, follow the steps detailed in the Pegasus - Integration examples section of the **1x PDI Builder** user manual.

Volz DA26 - RS485



Volz DA26

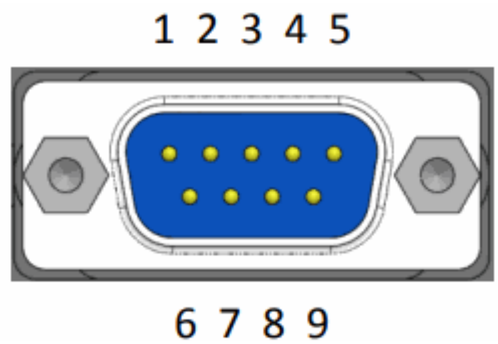
The following wiring connection is recommended for a **RS485 half-duplex** connection between **Volz DA26** servos and **Veronte Autopilot 1x**:



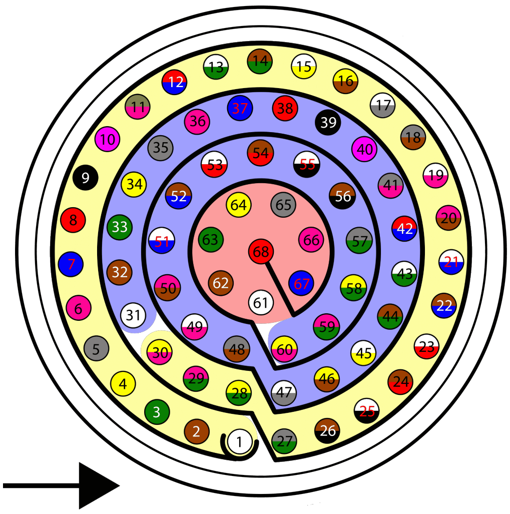
Volz DA26 - Autopilot 1x wiring connection

The above diagram is made for the case where 2 **Volz DA26** servos are connected, however, the connection is the same in case the user wants to connect only one or as many servos as the bus allows.

> **Important**
>
> Note that this servo must be connected to an **external power supply**, **sharing signal ground** with **Autopilot 1x**.



Volz DA26 connector pinout



Autopilot 1x harness pinout

Autopilot 1x Harness			Volz DA26 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
50	OUT_RS485_P	Pink-Brown	1	RS 485 A (Non-inverting Input/ Output)
53	IN_RS485_P	White-Red		

<table>
<tr><th colspan="3">Autopilot 1x Harness</th><th colspan="2">Volz DA26 Connector</th></tr>
<tr><th>PIN</th><th>Signal</th><th>Color Code</th><th>PIN</th><th>Signal</th></tr>
<tr><td>51</td><td>OUT_RS485_N</td><td>White-Blue</td><td rowspan="2">2</td><td rowspan="2">RS 485 B (Inverting Input/ Output)</td></tr>
<tr><td>52</td><td>IN_RS485_N</td><td>Brown-Blue</td></tr>
<tr><td rowspan="2">54</td><td rowspan="2">OUT_GND</td><td rowspan="2">Brown-Red</td><td>7</td><td>GND (1)</td></tr>
<tr><td>8</td><td>GND (2)</td></tr>
</table>

⚠ **Warning**

Note that this pin 54 is not a common GND pin.

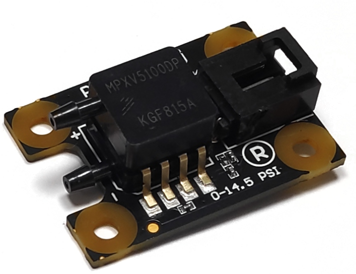
ⓘ **Note**

If users encounter any problems during wiring, please check the Half-duplex servo does not respond - Troubleshooting section of this manual.

Once the hardware integration of the device has been completed, it will be necessary to proceed with the **software integration**. To do this, follow the steps detailed in the Volz DA26-RS485 - Integration examples section of the **1x PDI Builder** user manual.

Air Data Sensors

High Speed Pitot Sensor



High Speed Pitot Sensor

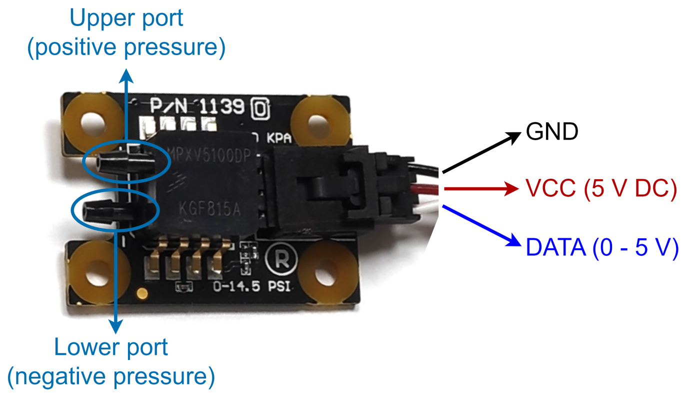
This section explains how to install the **High Speed Pitot Sensor** with an **Autopilot 1x** and configure it, so **1x** measures the air speed from the electrical signals of **High Speed Pitot Sensor**.

Required Material

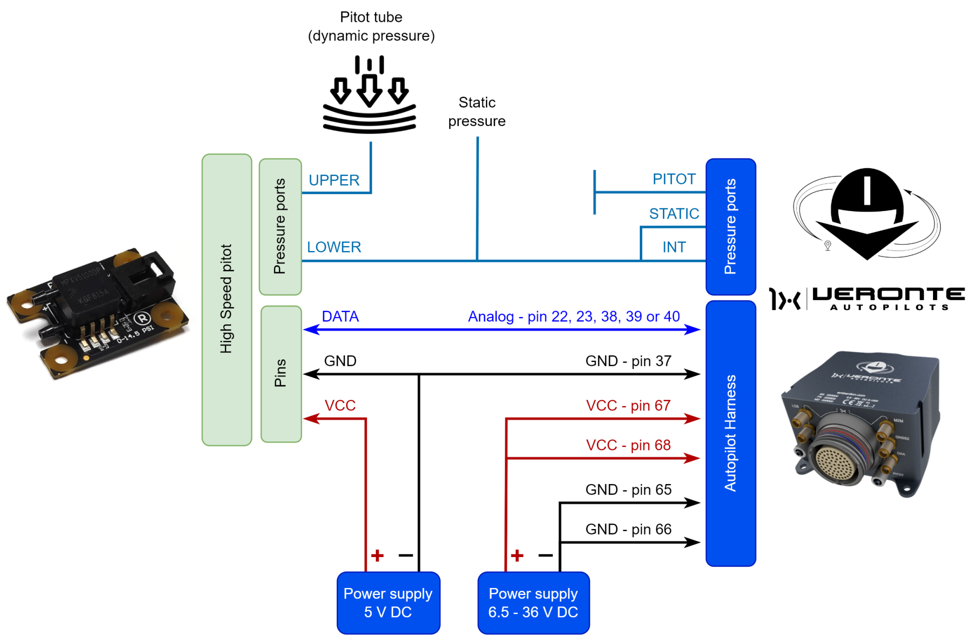
- **Veronte Autopilot 1x** with hardware version 4.8.
- **High Speed Pitot Sensor**.
- **Veronte Harness CS for 1x v4.8** with Embention reference **P001114**.
- Power supply for 1x (6.5 - 36 V DC).
- Power supply for **High Speed Pitot Sensor** (5 V DC).
- Pneumatic tubing, with 3 mm of outer diameter and 1 mm of inner. Silicone material is recommended.
- Two T-connectors for the tubing.

> ⚠ **Warning**
>
> Take caution when connecting the pressure ports. This sensor only measures a positive pressure differential, so the sensor might be damaged if it reverse connected.
>
> Ensure that the upper port is always at an equal or higher pressure than the lower port.

Connect the pneumatic tubes and wires according to the following diagrams:

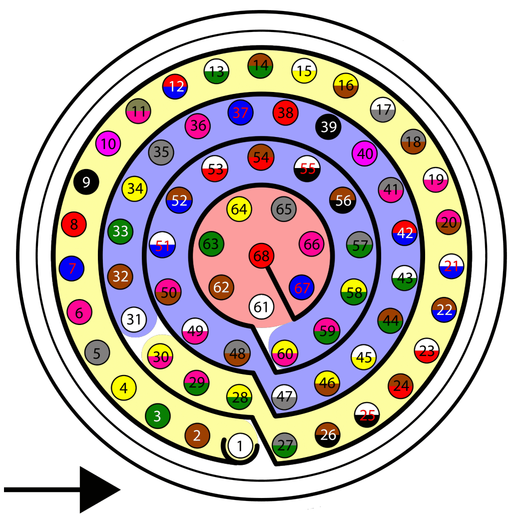


High Speed Pitot Sensor - Autopilot 1x connections diagram



High Speed Pitot Sensor - Autopilot 1x connections diagram

The analog pin defines which analog variable is used for configuration.



Autopilot 1x harness pinout

<table>
<tr><th colspan="3">Autopilot 1x Harness</th><th colspan="2">High Speed Pitot Sensor</th></tr>
<tr><th>PIN</th><th>Signal</th><th>Color Code</th><th>Signal</th><th>Color Code</th></tr>
<tr><td>38</td><td>ANALOG_0</td><td>Red</td><td rowspan="5">DATA</td><td rowspan="5">White</td></tr>
<tr><td>39</td><td>ANALOG_1</td><td>Black</td></tr>
<tr><td>40</td><td>ANALOG_2</td><td>Pink</td></tr>
<tr><td>22</td><td>ANALOG_3</td><td>Brown - Blue</td></tr>
<tr><td>23</td><td>ANALOG_4</td><td>White - Red</td></tr>
<tr><td>37</td><td>GND</td><td>Blue</td><td>GND</td><td>Black</td></tr>
</table>

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

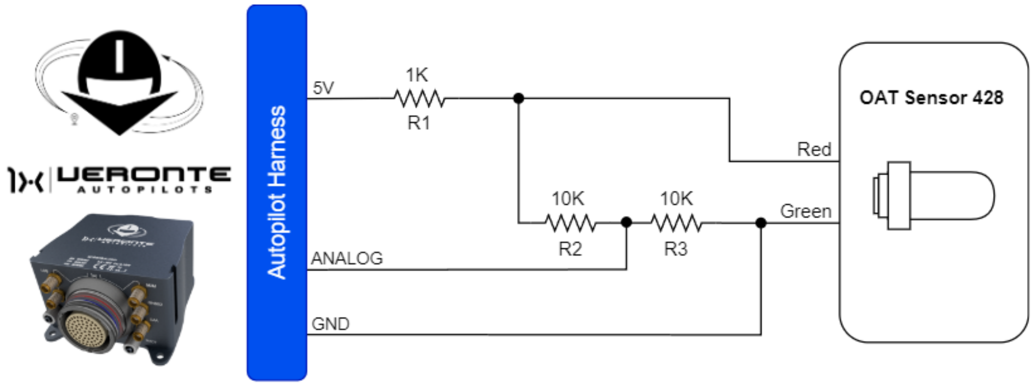
Once the **High Speed Pitot Sensor** is connected, air speed measurements can be monitored with **1x PDI Builder** using the variables ADC0 to ADC4. The integration of this device with **Autopilot 1x** as far as the software is concerned is explained in the High Speed Pitot Sensor - Integration examples section of **1x PDI Builder** user manual.

OAT sensor 428 of MGL Avionics

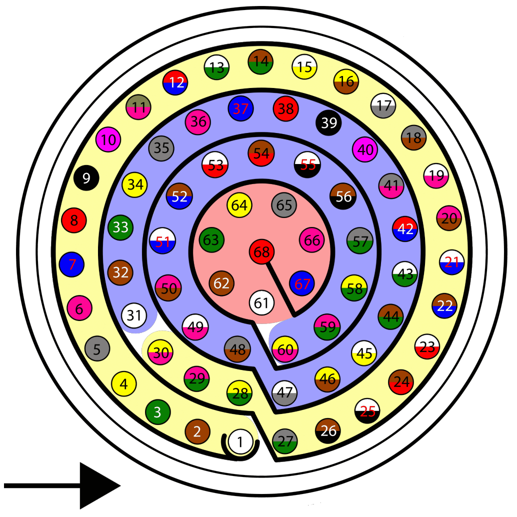
OAT sensor 428

The **OAT sensor 428** of MGL Avionics is an analogical temperature sensor that measures temperatures from -55°C to 150°C. It changes the voltage according to the temperature measured and therefore the connection to the autopilot is performed using the ADC pins.

The following resistors and wiring are necessary to connect an **OAT sensor 428** to the **Autopilot 1x**:



OAT sensor - Autopilot 1x wiring diagram



Autopilot 1x harness pinout

Autopilot 1x Harness		
PIN	**Signal**	**Color Code**
36	5V	Pink
37	GND	Blue
38	ANALOG_0	Red
39	ANALOG_1	Black
40	ANALOG_2	Pink
22	ANALOG_3	Brown - Blue
23	ANALOG_4	White - Red

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once connected the **OAT sensor**, the temperature can be monitored with **1x PDI Builder** using the variables ADC0 to ADC4.

The integration of this device with **Autopilot 1x** is explained in the OAT Sensor - Integration examples section of the **1x PDI Builder** user manual.

Altimeters

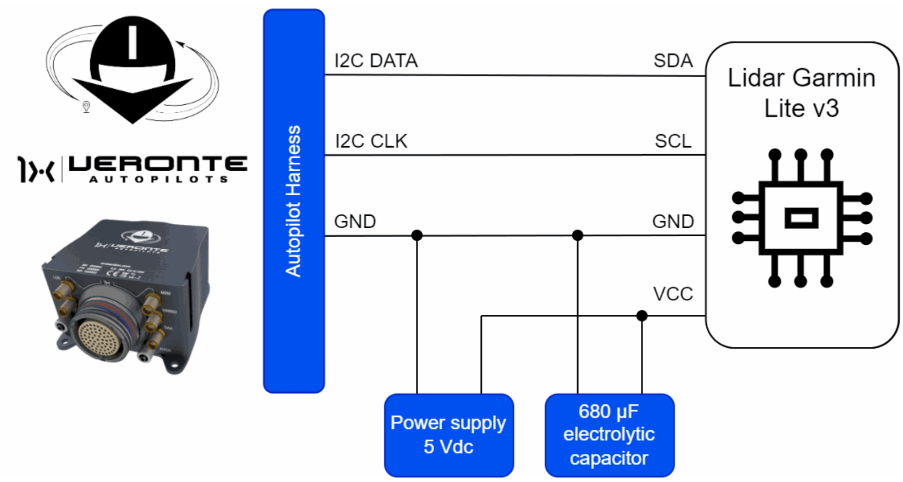
Lidar

The integration between **Veronte Autopilot 1x** and a lidar is performed using a variety of interfaces depending on the lidar device. The most common interfaces are I2C or analog although serial or CAN bus can also be used if the lidar is compatible.

Lidar Garmin Lite v3

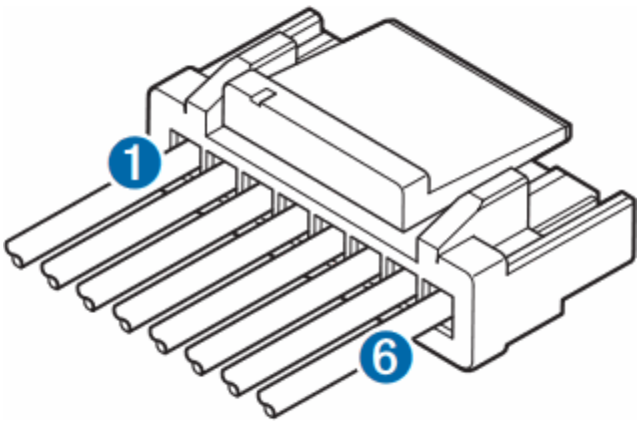
Lidar Garmin Lite v3

Lidar Garmin Lite v3 sensor integrates with **Autopilot 1x** via **I2C** connection.

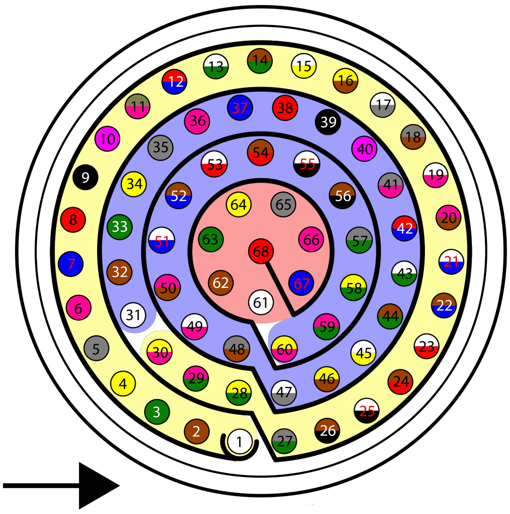


Lidar Garmin Lite v3 - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, it is recommended to use an **external power supply** and **not to connect** it to the **3.3-5 V** lines of **Autopilot 1x**.
>
> Please note that it **shares signal ground** with **Autopilot 1x**.



Lidar Garmin Lite v3 connector



Autopilot 1x harness pinout

Autopilot 1x Harness			Lidar Garmin Lite v3 Connector		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
31	I2C_CLK	White	4	I2C SCL	Green
32	I2C_DATA	Brown	5	I2C SDA	Blue
35	GND	Gray	6	Ground (-)	Black

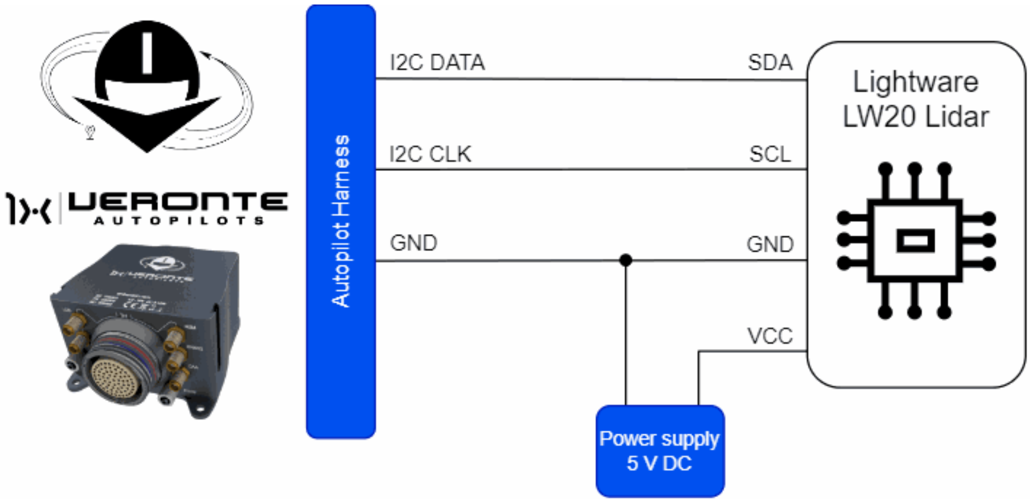
> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the lidar is connected, the user must proceed to its software integration with **Veronte Autopilot 1x** by referring to the Lidar Garmin Lite v3 - Integration examples section of the **1x PDI Builder** user manual.

Lightware LW20 Lidar

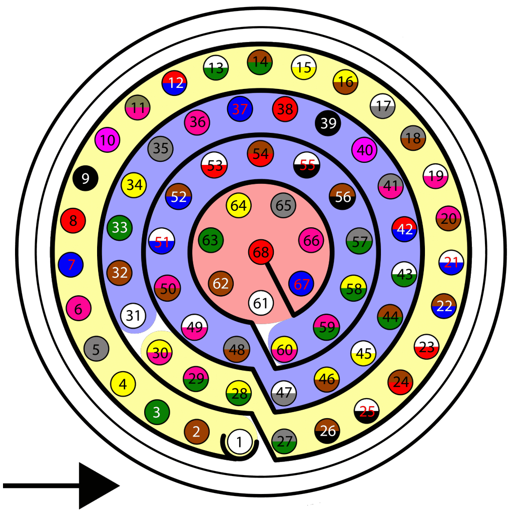
Lightware LW20 Lidar

Lightware LW20 Lidar sensor integrates with **Autopilot 1x** via **I2C** connection.



Lightware LW20 Lidar - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, it is recommended to use an **external power supply** and **not to connect** it to the **3.3-5 V** lines of **Autopilot 1x**.
>
> Please note that it **shares signal ground** with **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness			Lightware LW20 Lidar		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
31	I2C_CLK	White	2	SCL	White
32	I2C_DATA	Brown	1	SDA	Yellow
35	GND	Gray	8	GND	Black

⚠ **Warning**

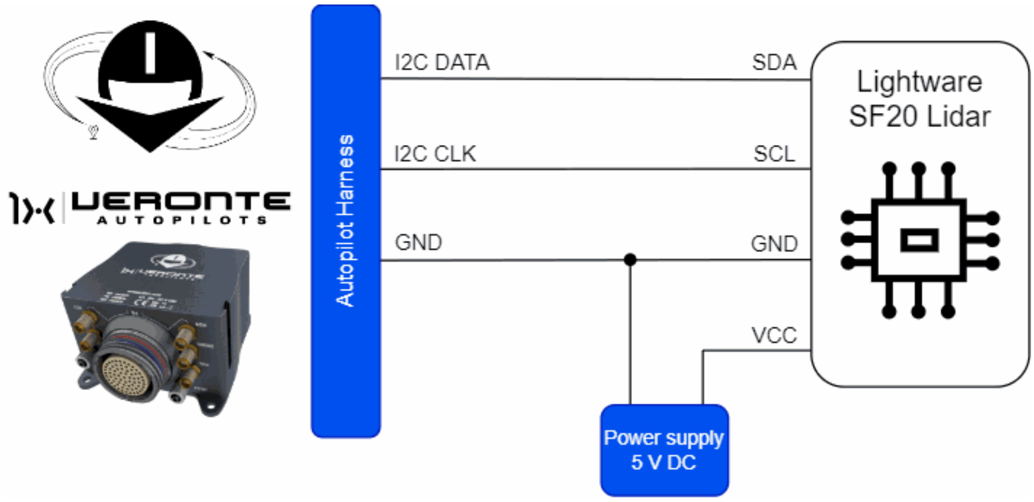
Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once **Lightware LW20 Lidar** is connected, the user must proceed to its software installation with **Veronte Autopilot 1x** by referring to the Lightware LW20 Lidar - Integration examples section of the **1x PDI Builder** user manual.

Lightware SF20 Lidar

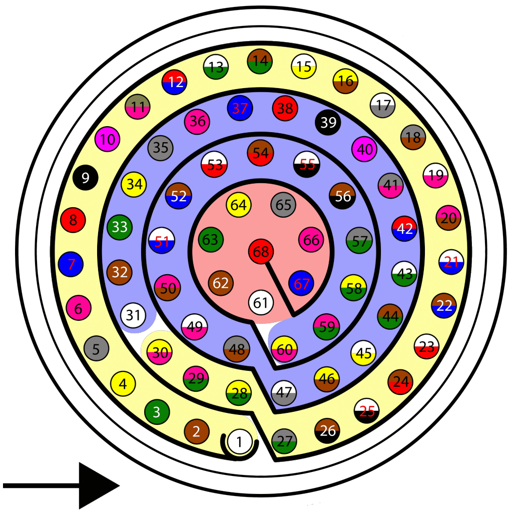
Lightware SF20 Lidar

Lightware SF20 Lidar sensor integrates with **Autopilot 1x** via **I2C** connection.



Lightware SF20 Lidar - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, it is recommended to use an **external power supply** and **not to connect** it to the **3.3-5 V** lines of **Autopilot 1x**.
>
> Please note that it **shares signal ground** with **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness			Lightware SF20 Lidar		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
31	I2C_CLK	White	2	SCL	White
32	I2C_DATA	Brown	1	SDA	Yellow
35	GND	Gray	8	GND	Black

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once **Lightware SF20 Lidar** is connected, the user must proceed to its software installation with **Veronte Autopilot 1x**. Since this setup is the same as for the **Lightware LW20**, please refer to the Lightware LW20 Lidar - Integration examples section of the **1x PDI Builder** user manual.

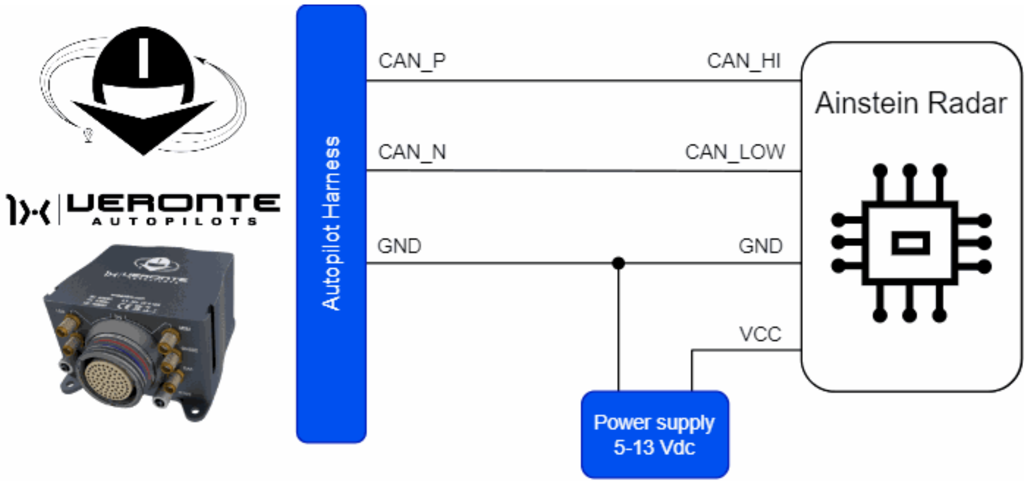
Radar

Radar altimeters are common devices on aircrafts.

Ainstein CAN Radar

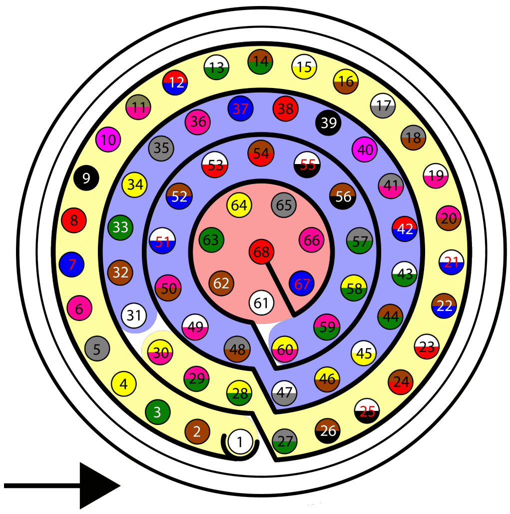
Ainstein CAN Radar

Ainstein CAN Radar sensor integrates with **Autopilot 1x** via **CAN** connection.



Ainstein CAN Radar - Autopilot 1x wiring diagram

> **Important**
>
> To ensure correct operation of the device, it is recommended to use an **external power supply** and **not to connect** it to the **5 V** line of **Autopilot 1x**.
>
> Please note that it **shares signal ground** with **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness			Ainstein CAN Radar		
PIN	Signal	Color Code	PIN	Signal	Color Code
25	CANA_P	White-Black	2	CAN_HI	Green
28	CANB_P	Yellow-Green			
26	CANA_N	Brown-Black	3	CAN_LOW	White
29	CANB_N	Pink-Green			
30	GND	Yellow-Pink	4	Ground	Black

> ⓘ **Note**
>
> CAN A and CAN B buses are equivalent and can be used interchangeably for the integration of this device.
>
> For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once **Ainstein CAN Radar** is connected, the user must proceed to its software installation with **Veronte Autopilot 1x** by referring to the Ainstein CAN Radar - Integration examples section of the **1x PDI Builder** user manual.

Datalinks

LOS

Amount of data sent via radiolink

Regardless of the radio used, the amount of telemetry data that can be sent by radiolink is limited.

> **Important**
>
> Refer to the Data Transmission section of the **1x Software Manual** for information on the amount of data that can be sent.

Digi radio (as internal radio)

Internal Digi radios can establish communication between Veronte Autopilots 1x.

The necessary configuration of Digi radios for proper communication between them and **1x** is described in the Digi internal radio - Integration examples section of the **1x PDI Builder** manual.

DTC (Domo Tactical) radio (SOL8SDR-C model)

System Layout

It is possible to operate DTC radios in two different ways, with or without amplifiers.

- **DTC**
  The following image shows the standard connection between **DTC** radios and **Autopilot 1x** for operation:

DTC radios and Autopilot 1x operation

- **DTC + Amplifier**

> ⓘ **Note**
>
> Amplifier information: **AMPD5W** model, 5W Linear RF Power Amplifier.

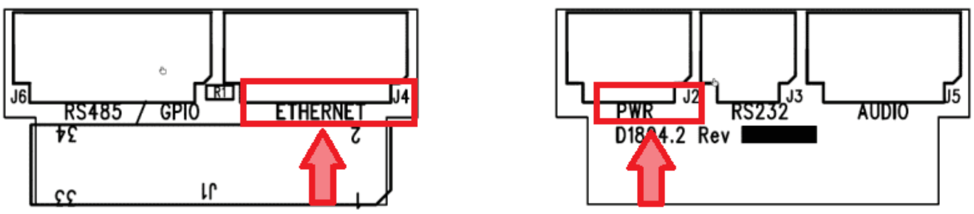
The following image shows the standard connection between **DTC** radios, amplifiers and **Autopilot 1x** for operation:

DTC + amplifier radios and 1x operation

Hardware Installation

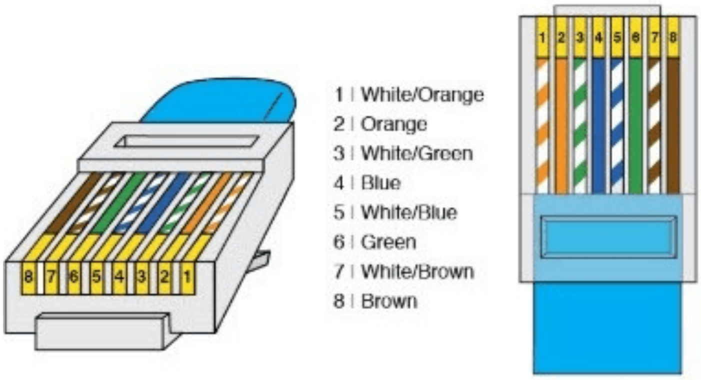
Depending on the action to be taken, different hardware installations are possible:

1. To **configure a DTC radio** it is required to carry out the installation of the ethernet and power connection:



DTC D1804 Gecko breakout PCB

   - **Ethernet**



RJ45 pinout T-568B

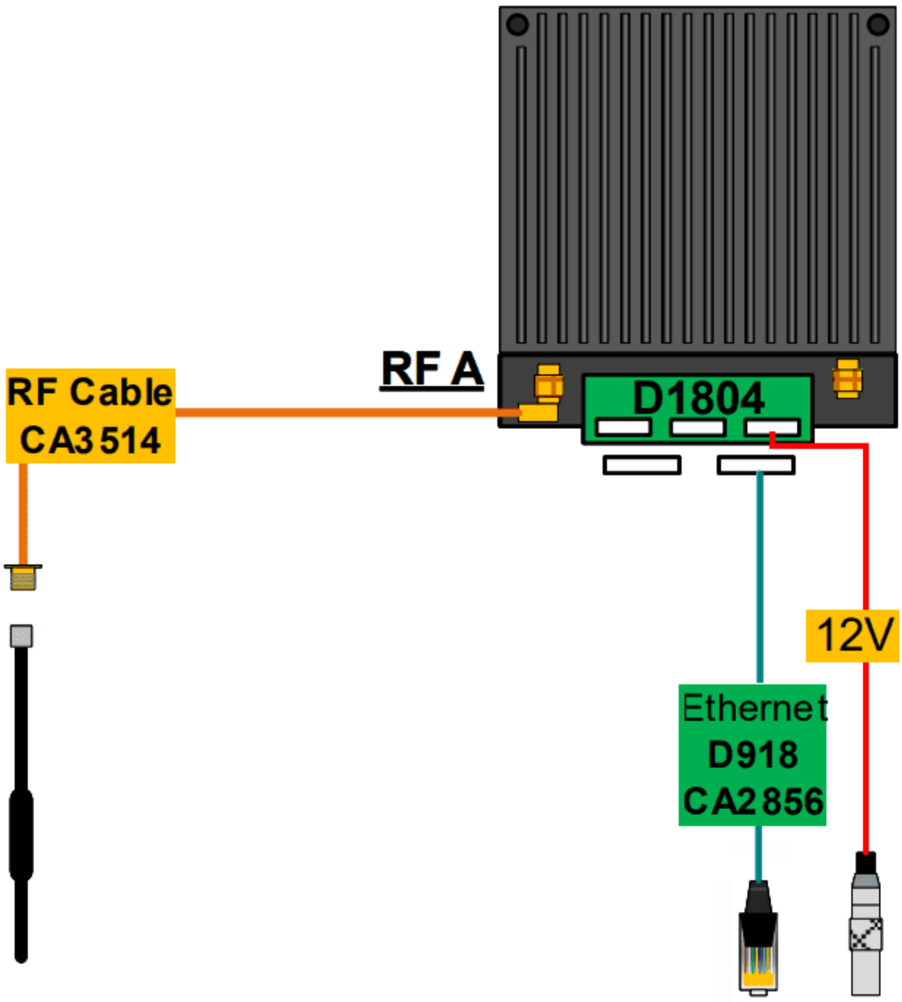
J4 (Ethernet) - D1804 Gecko breakout PCB		RJ45 Connector (T-568B)		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
1	Ethernet MDIP0	1	TX+	Orange-White
2	Ethernet MDIN0	2	TX-	Orange
3	Ethernet MDIP1	3	RX+	Green-White
4	Ethernet MDIN1	6	RX-	Green

- **Power supply**

Female DC Power Jack connector

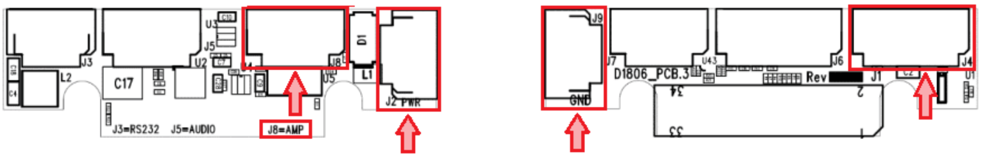
J2 (PWR) - D1804 Gecko breakout PCB		Power Connector
PIN	**Signal**	**Signal**
1	VIN	Power +
2	VIN	
3	GND	Power -
4	GND	

The full connection should look like this:



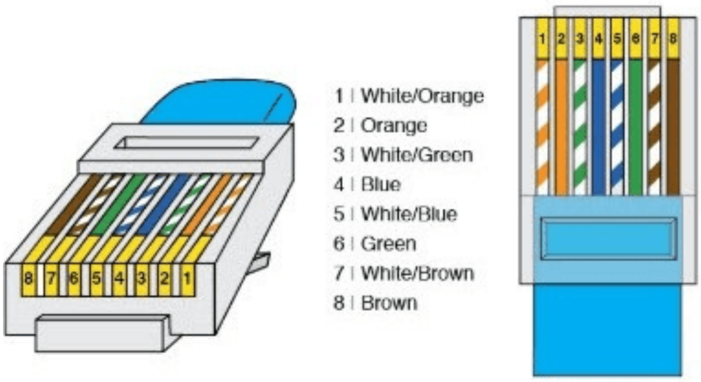
DTC connection - Configuration

2. To **configure a DTC + amplifier radio** it is required to carry out the installation of the ethernet, power and amplifier connection:



DTC D1806 Gecko active breakout PCB

- **Ethernet**



RJ45 pinout T-568B

J4 (Ethernet) - D1806 Gecko Active Breakout PCB		RJ45 Connector (T-568B)		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
1	Ethernet MDIP0	1	TX+	Orange-White
2	Ethernet MDIN0	2	TX-	Orange
3	Ethernet MDIP1	3	RX+	Green-White

J4 (Ethernet) - D1806 Gecko Active Breakout PCB		RJ45 Connector (T-568B)		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
4	Ethernet MDIN1	6	RX-	Green

- ◦ **Power supply**

Female DC Power Jack connector

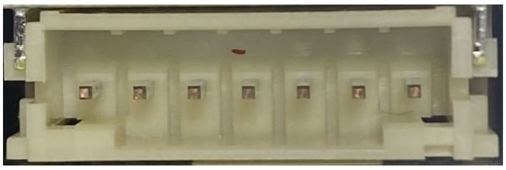
<

J2 (PWR) - D1806 Gecko Active Breakout PCB		Power Connector
PIN	**Signal**	**Power Signal**
1	VIN	Power +
2	VIN	
3	VIN	
4	VIN	

J2 (PWR) - D1806 Gecko Active Breakout PCB		Power Connector
PIN	**Signal**	**Power Signal**
5	VIN	
6	VIN	

J9 (GND) - D1806 Gecko Active Breakout PCB		Power Connector
PIN	**Signal**	**Power Signal**
1	GND	Power -
2	GND	
3	GND	
4	GND	
5	GND	
6	GND	

- **Amplifier**

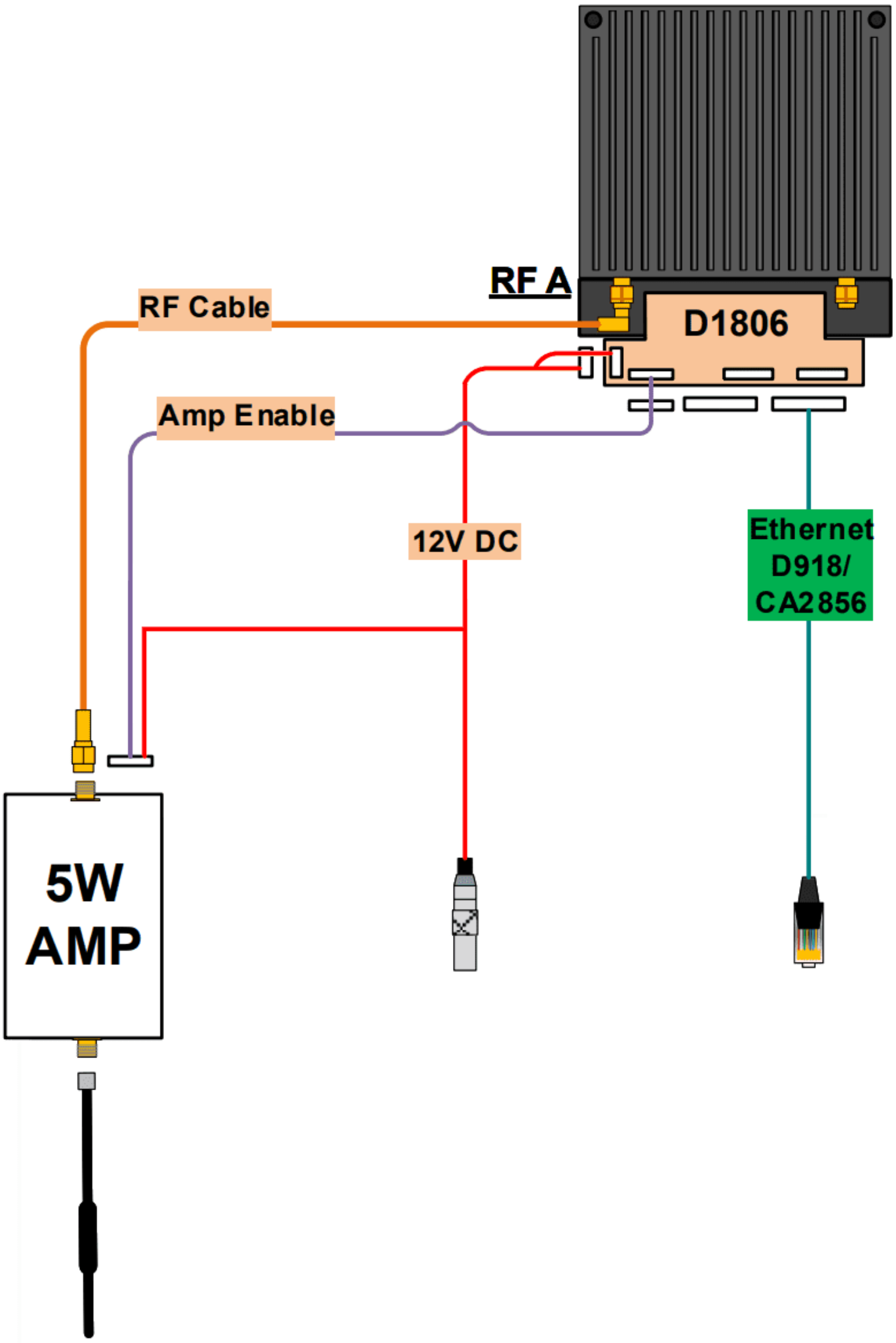
  Amplifier AMPD5W

J8 (AMP) - D1806 Gecko Active Breakout PCB		AMPD5W Connector	
PIN	**Signal**	**PIN**	**Signal**
1	5V_SDA	6	5V_SDA
2	GND	3	GND
4	5V_SCL	5	5V_SCL
7	PA_TDD	7	PA_TDD

J2 (PWR) - D1806 Gecko Active Breakout PCB		AMPD5W Connector	
PIN	**Signal**	**PIN**	**Signal**
1	VIN	1 & 2	Power +
2	VIN		
3	VIN		
4	VIN		
5	VIN		
6	VIN		

<table>
<tr><th colspan="2">J9 (GND) - D1806 Gecko Active Breakout PCB</th><th colspan="2">AMPD5W Connector</th></tr>
<tr><th>PIN</th><th>Signal</th><th>PIN</th><th>Signal</th></tr>
<tr><td>1</td><td>GND</td><td rowspan="6">3 & 4</td><td rowspan="6">Power -</td></tr>
<tr><td>2</td><td>GND</td></tr>
<tr><td>3</td><td>GND</td></tr>
<tr><td>4</td><td>GND</td></tr>
<tr><td>5</td><td>GND</td></tr>
<tr><td>6</td><td>GND</td></tr>
</table>

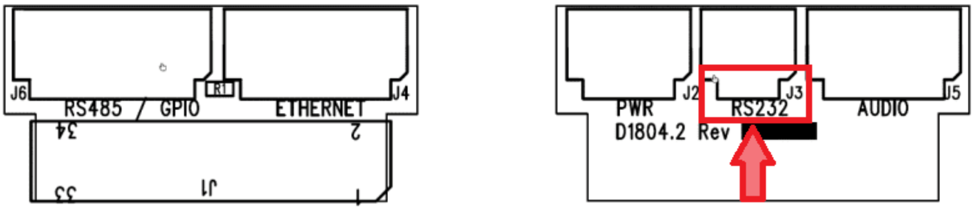
The full connection should look like this:



DTC + amplifier connection - Configuration

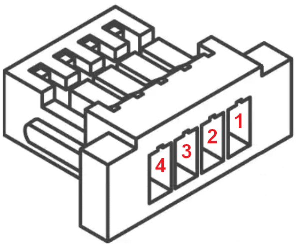
3. To **connect a DTC radio to a Veronte Autopilot 1x** the following installation must be carried out:

   As, the connection of a **DTC** radio to a **Veronte Autopilot 1x** must be made via **RS-232**, the connection will be the same as in the configuration case (1), but adding the wiring to RS-232 port.

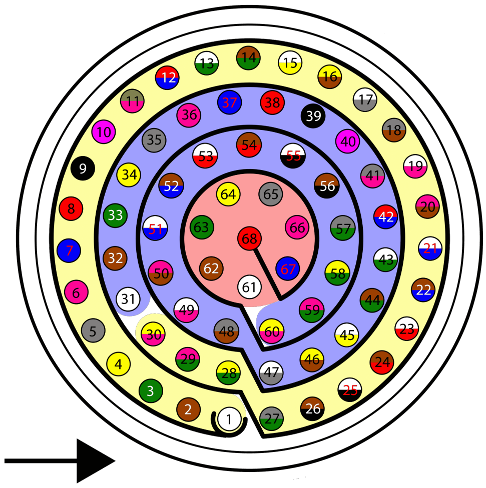


DTC D1804 Gecko breakout PCB - J3 (RS232)

This RS-232 should be connected to the RS-232 of **Autopilot 1x** harness.



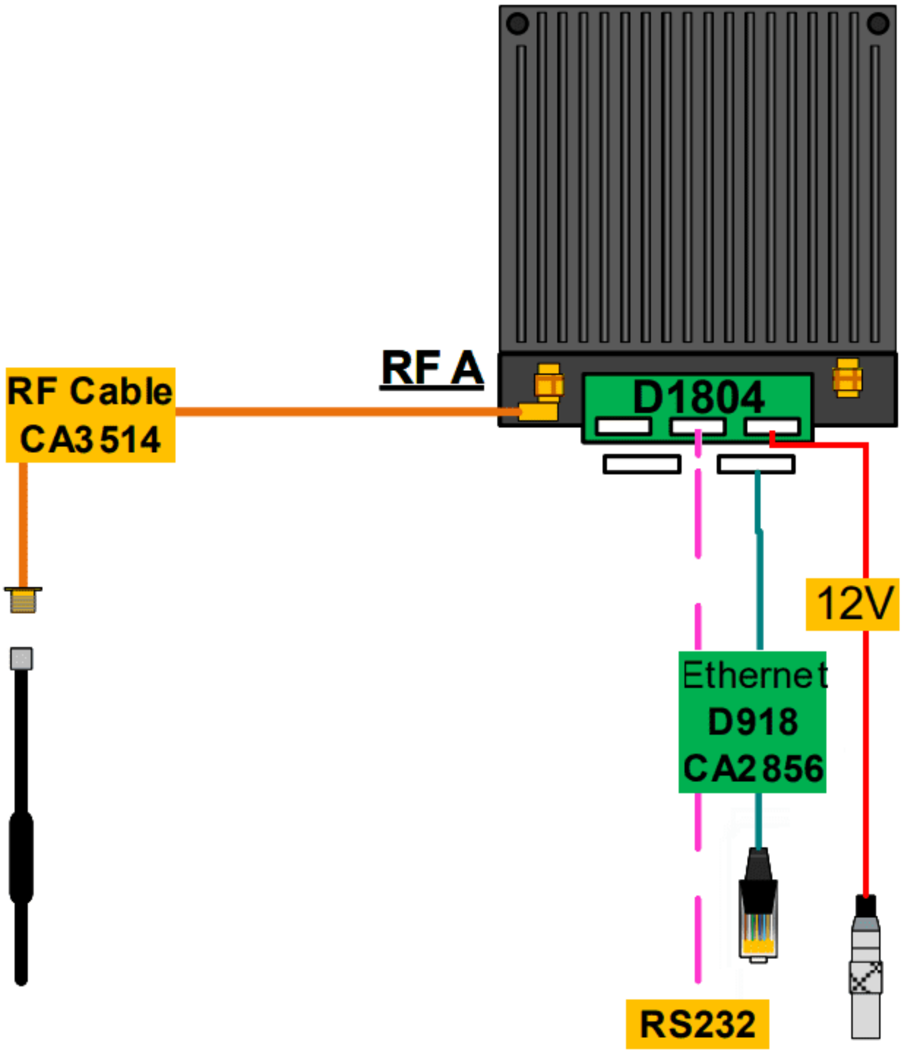
J3 (RS232) pinout of female connector



Autopilot 1x harness pinout

J3 (RS232) - D1804 Gecko Breakout PCB		Autopilot 1x Harness		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
2	RS232 RX	19	RS232 TX	White-Pink
1	RS232 TX	20	RS232 RX	Pink-Brown
3	GND	21	GND	White-Blue

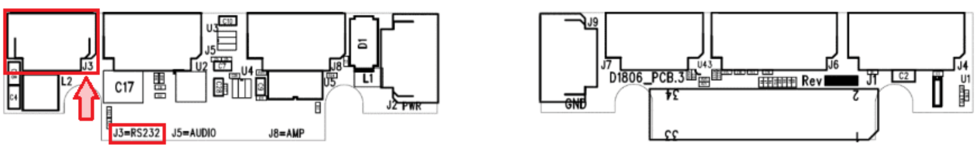
The full connection should look like this:



DTC connection - Veronte Autopilot 1x

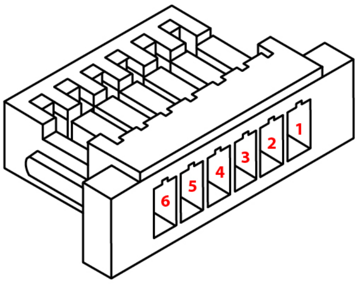
4. To **connect a DTC + amplifier radio to a Veronte Autopilot 1x** the following installation must be carried out:

   As, the connection of a **DTC** radio to a **Veronte Autopilot 1x** must be made via **RS-232**, the connection will be the same as in the configuration case (2), but adding the wiring to RS-232 port.

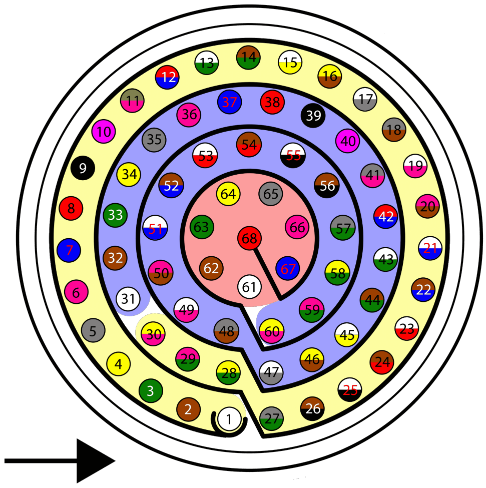


DTC D1806 Gecko active breakout PCB - J3 (RS232)

This RS-232 should be connected to the RS-232 of **Autopilot 1x** harness.



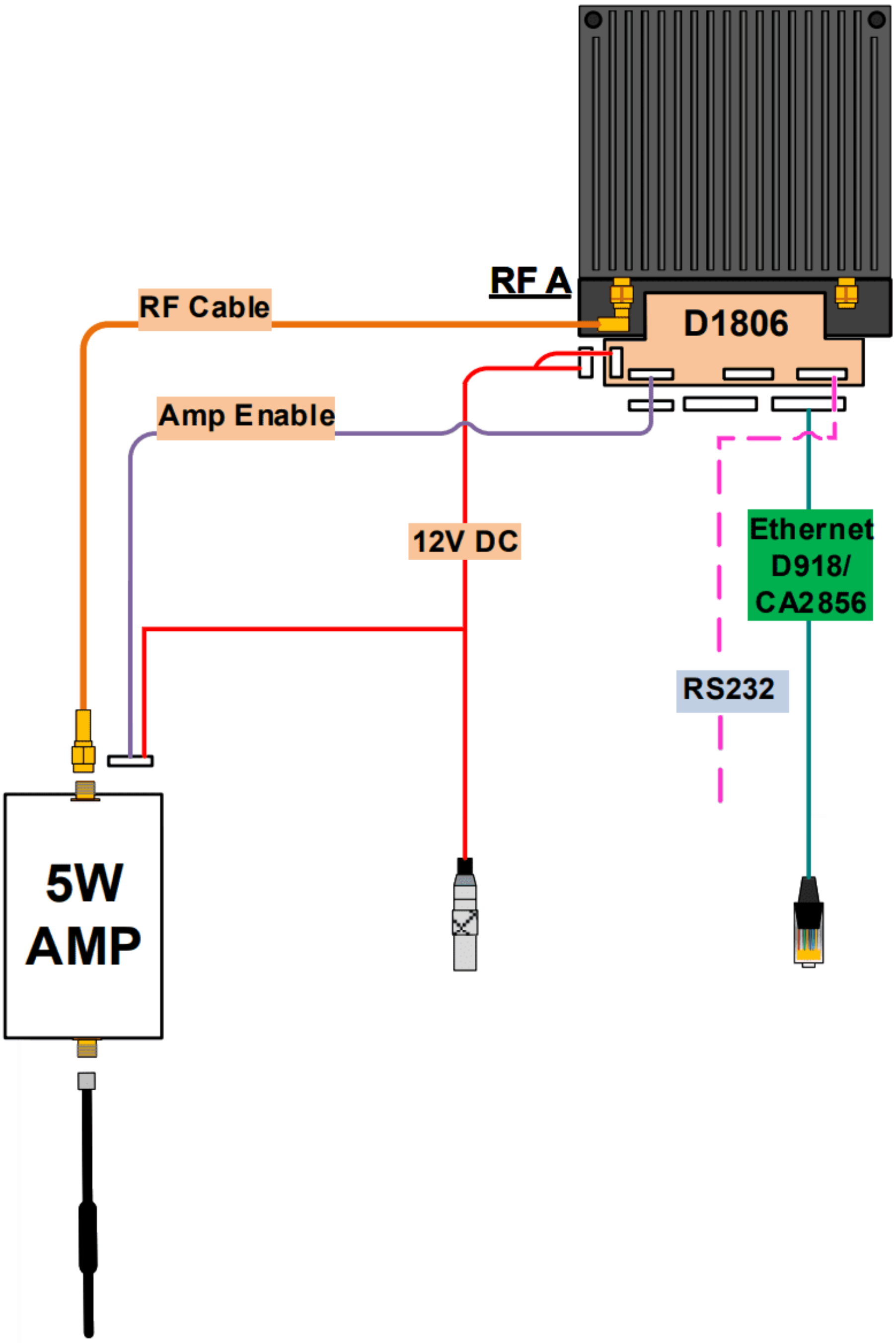
J3 (RS232) pinout of female connector



Autopilot 1x harness pinout

J3 (RS232) - D1806 Gecko Active Breakout PCB		Autopilot 1x Harness		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
2	RS232 RX	19	RS232 TX	White-Pink
1	RS232 TX	20	RS232 RX	Pink-Brown
3	GND	21	GND	White-Blue

The full connection should look like this:

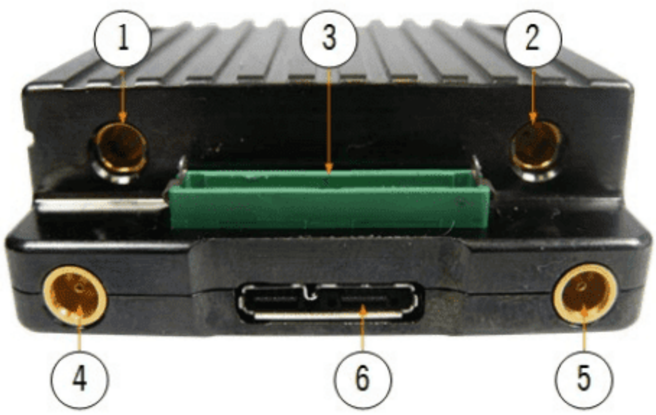


DTC + amplifier connection - Veronte Autopilot 1x

> ⚠ **Caution**
>
> It is also possible to **calibrate** the **power output** of **DTC** radios and **DTC** + amplifier radios.
>
> However, the radios are shipped with a factory calibration, it is **strongly recommended to not modify this calibration**. If the user wishes to modify it, please contact the support team (create a ticket in the customer's **Joint Collaboration Framework**; for more information, see Tickets section of the JCF manual).

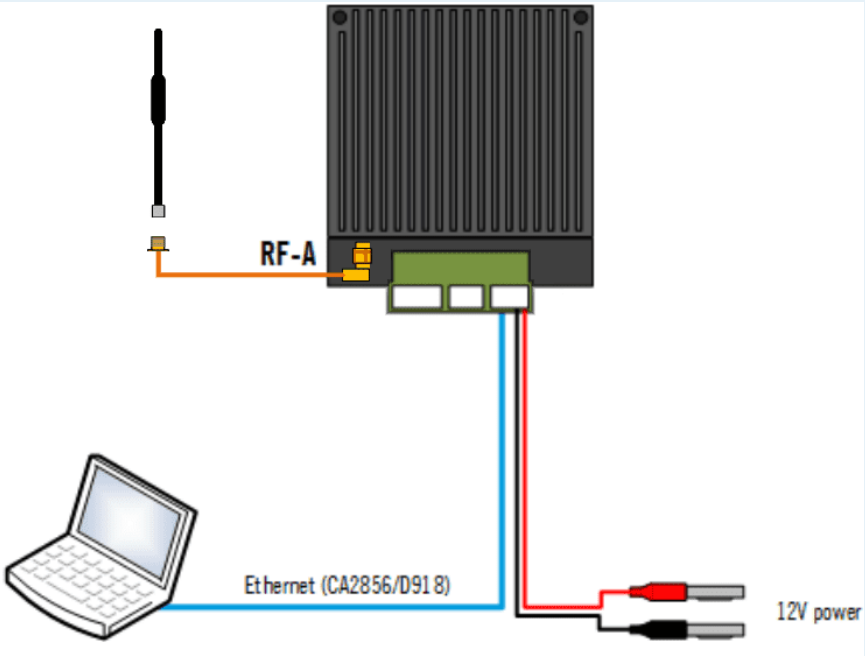
DTC radio configuration

First steps

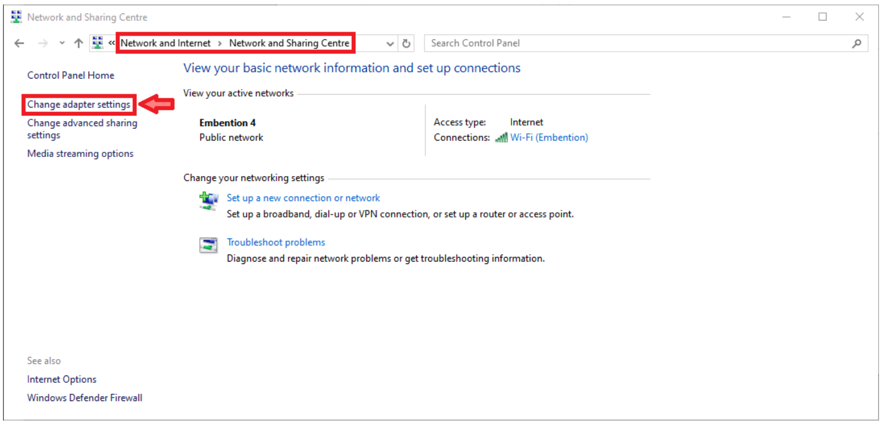


DTC ports

- **DTC without amplifier**
  1. Connect to 1 an SMP to SMA RF cable (this is the default transmit output).
  2. Connect this SMA RF cable to a 2.4 GHz antenna.
  3. Connect to 3 the **D1804 Gecko breakout PCB** supplied with the unit.
  4. Connect J2 (**PWR**) of the **D1804 PCB** to 12V power.
  5. In order to access the web browser control application, connect J4 (**ETHERNET**) of the **D1804 PCB** to a PC or network Ethernet port via CA2856 and D918.

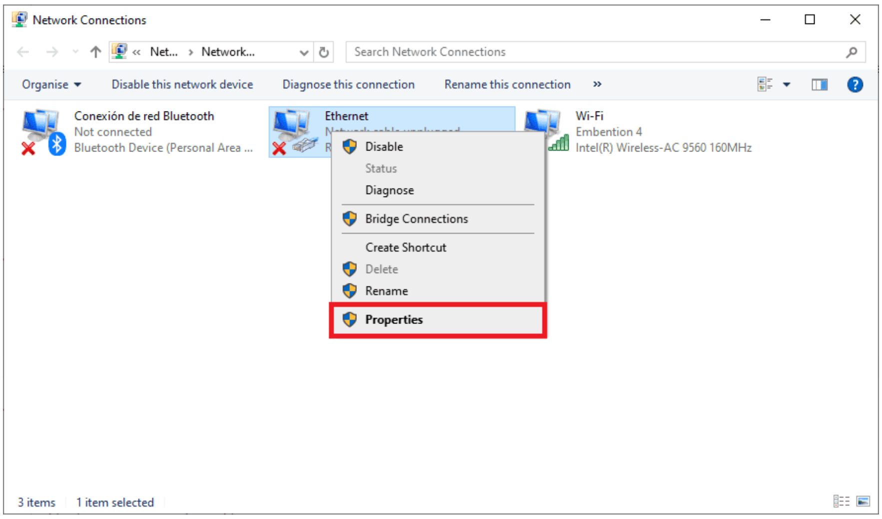
> **Note**
>
> The connections should look like this:
>
> 
>
>
> **First steps connection**

6. Make sure computer is set to static IP address on same subnet as radio. The following substeps clarify how to set the IP address in the **Control Panel**:
    1. Open **Network and Sharing Centre** menu and click **Change adapter settings**.

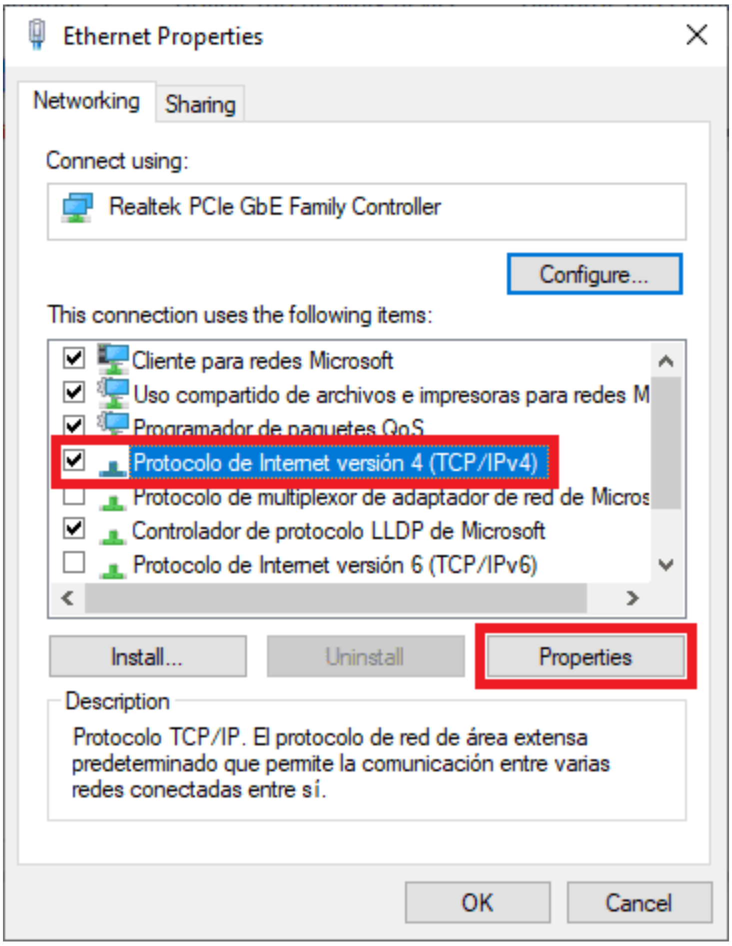
       Ethernet connection 1

    2. Select **Local Area Connection**, right click, and select **Properties**.



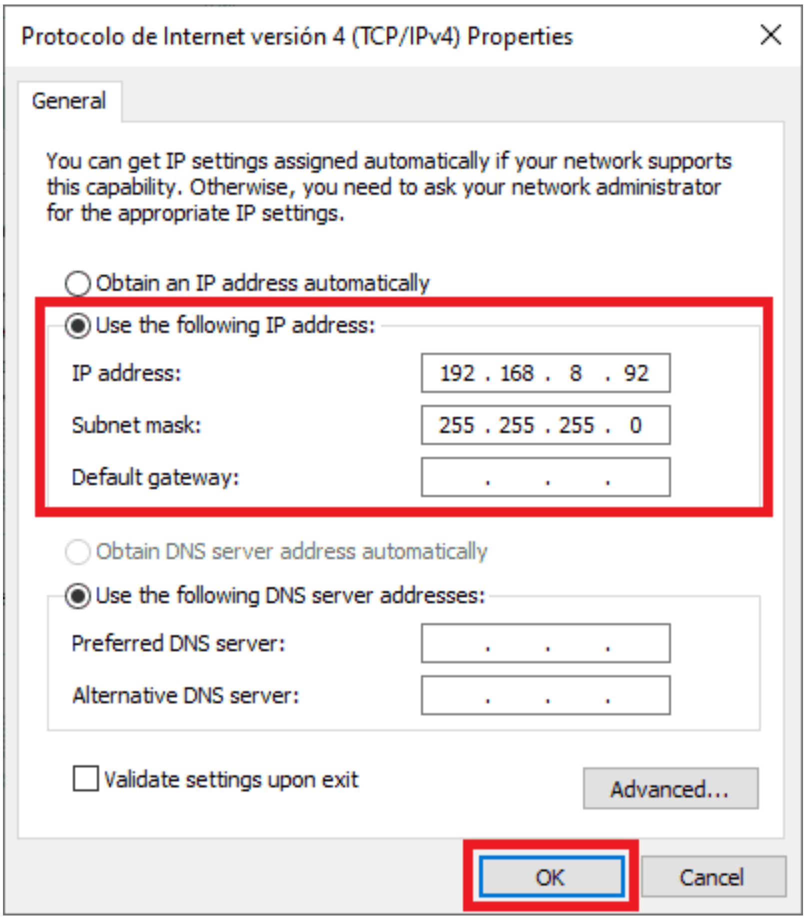
Ethernet connection 2

3. Select **IPv4** and click **Properties**.



Ethernet connection 3

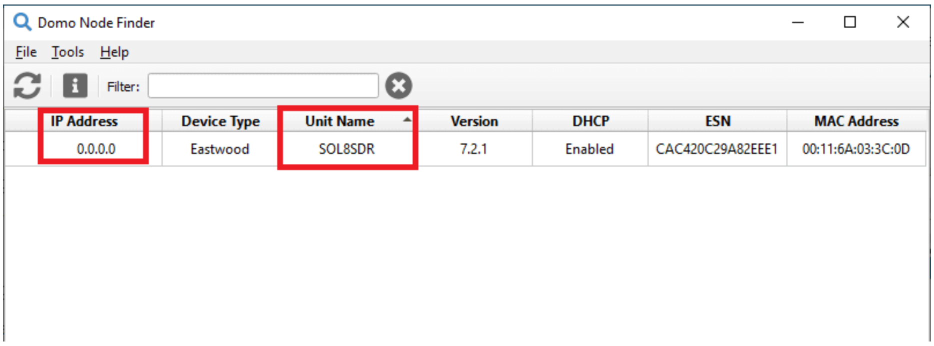
4. Set **IP address** to 192.168.8.YY (e.g. if the IP of the radio is 192.168.8.95, set the IP 192.168.8.92) and **Subnet mask** to 255.255.255.0. Click **OK**.



Ethernet connection 4

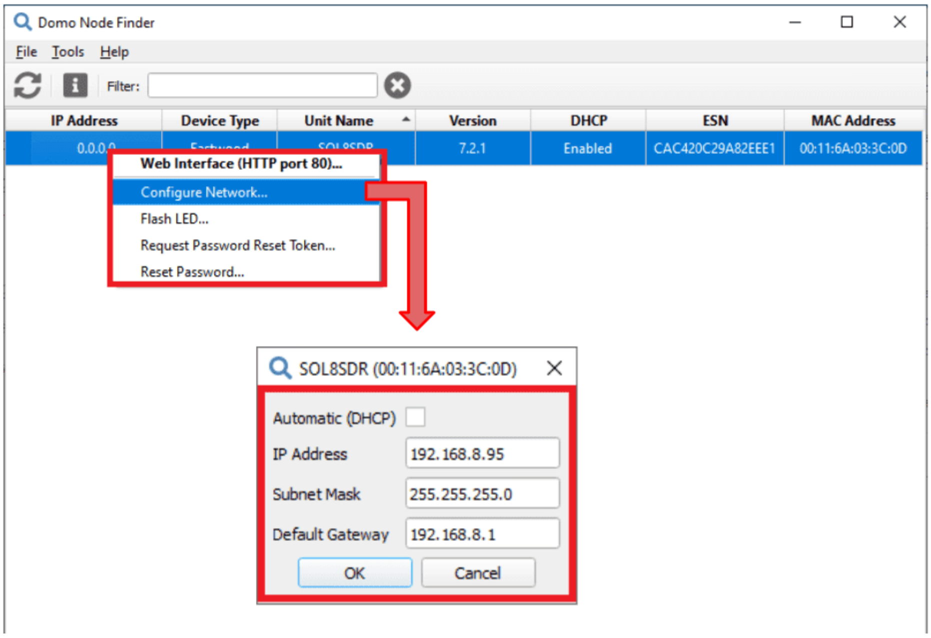
7. First, it is necessary to have the '**Domo Node Finder**' software installed.
8. Open **Domo Node Finder** and the connected radios will appear here as SOL8SDR.

   By default, 0.0.0.0 is the IP address of the radio:



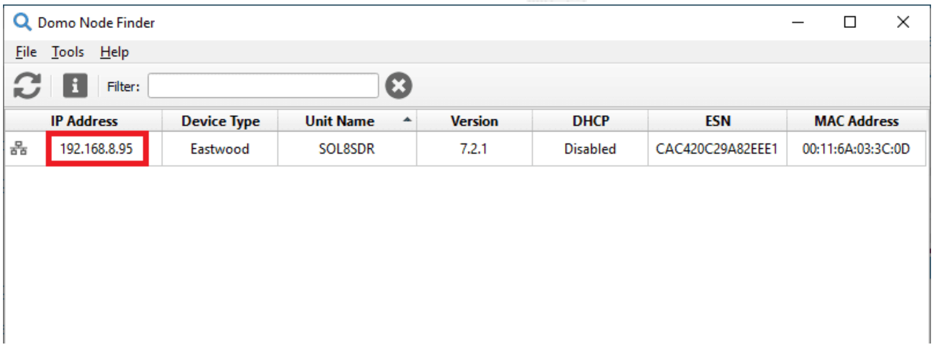
Domo Node Finder - Default IP address

9. To configure the IP address, **right-click** the IP address and select **Configure Network** to disable the DHCP setting and set the following **static IP address**:



Domo Node Finder - IP address configuration

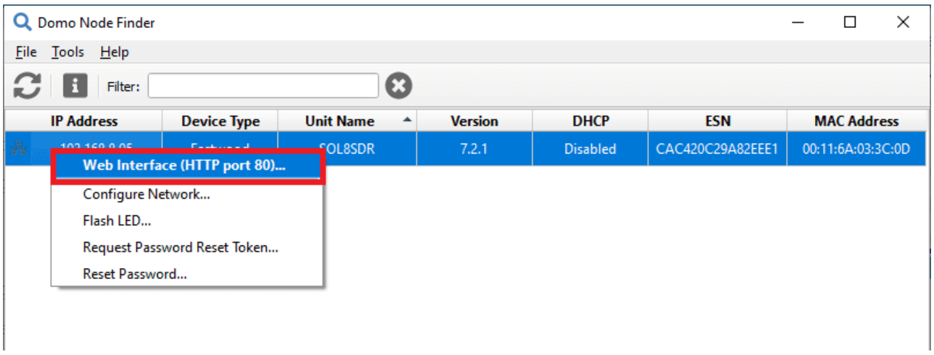
To confirm the change, click the  icon to update the IP address.



Domo Node Finder - Configured IP address

> ⓘ **Note**
>
> This IP address, **192.168.8.95**, is related to the radio linked to the **ground unit**. For the radio linked to the **air unit**, the IP address should be **192.168.8.96**.

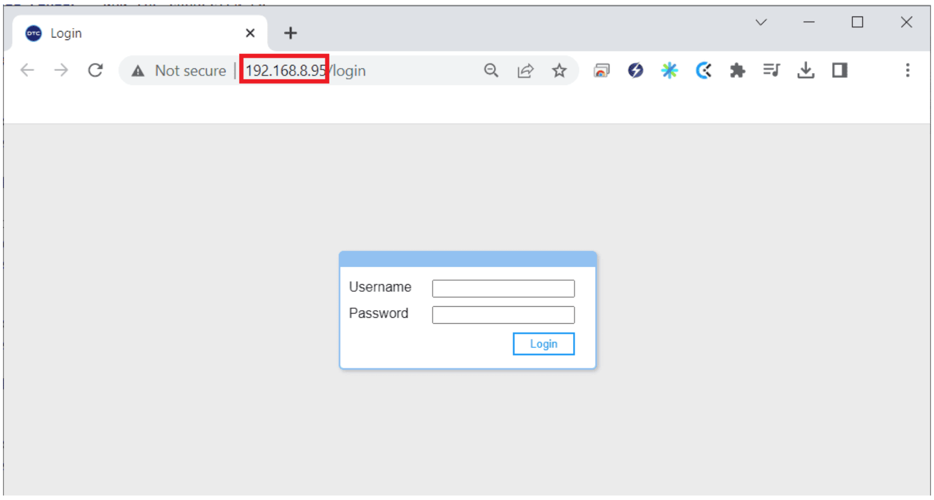
10. To open the **DTC** web browser control application, users can **right-click** the **IP address** and select **WEB Interface (HTTP port 80)**, **double-click** on the **IP address** or enter the IP address of the SOL8SDR-C on the address bar of a web browser.



Domo Node Finder - Open Web Browser Application

> ⓘ **Note**
>
> Although the application should work with any web browser, **DTC** recommends the use of Internet Explorer, Google Chrome or Firefox.

11. An authentication required dialogue box will open. Leave the **Username blank** and enter the **Password** as **Eastwood**.

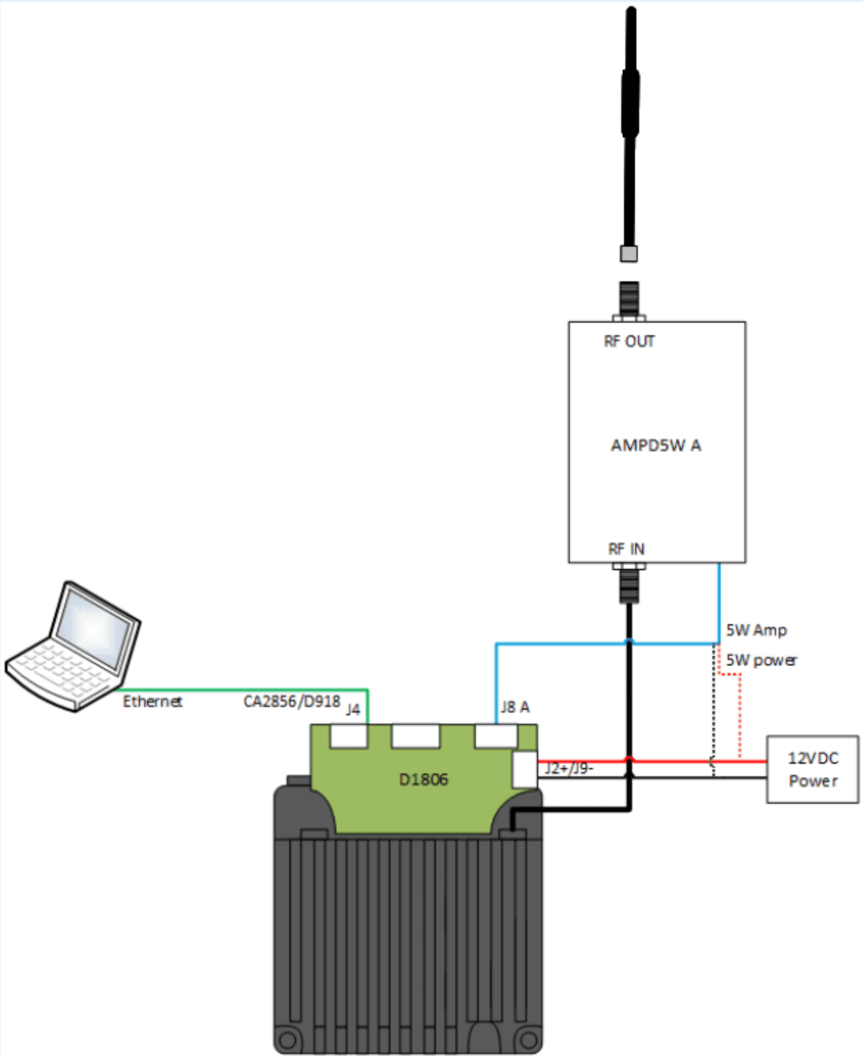


Domo Node Finder - Open Web Browser Application

12. Click **Login** and the web browser control application will open.

- **DTC with amplifier**
  1. Connect to 1 an SMP to SMA RF cable (this is the default transmit output).
  2. Connect this **SMA RF cable** to the amplifier **RF IN port**.
  3. Connect a 2.4 GHz **antenna** to the amplifier **RF OUT port**.
  4. Connect to 3 the **D1806 Gecko active breakout PCB**.

5. Connect J4 (**ETHERNET**) of the **D1806 PCB** to a PC or network Ethernet port via CA2856 and D918.
6. Connect J2 (**PWR**) and J9 (**GND**) of the **D1806 PCB** to 12V power.
7. Connect J8 (**AMP**) of the **D1806 PCB** to the control cable from the amplifier connector.

> ⓘ **Note**
>
> The connections should look like this:
>
> 
>
>
> **First steps connection + amplifier**

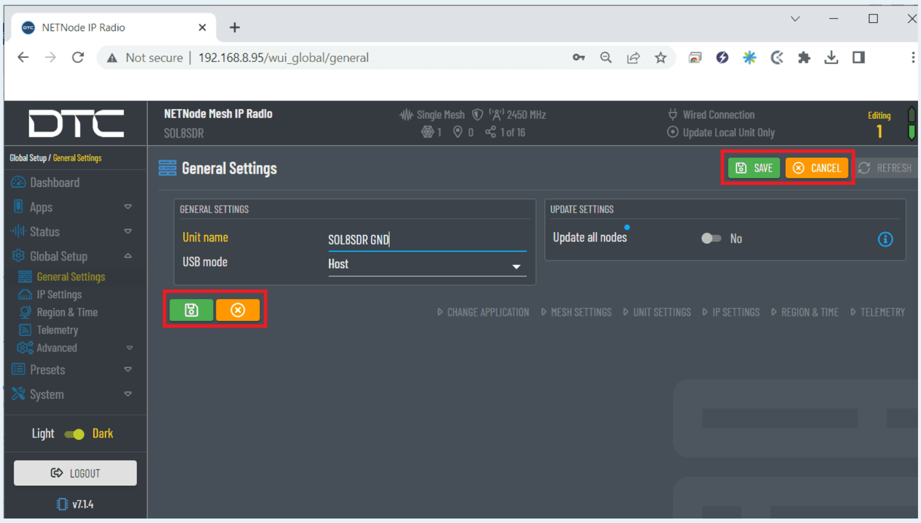
8. Now the steps to follow are the same as from step 6. of a DTC without amplifier, described above.

Point-to-Point configuration

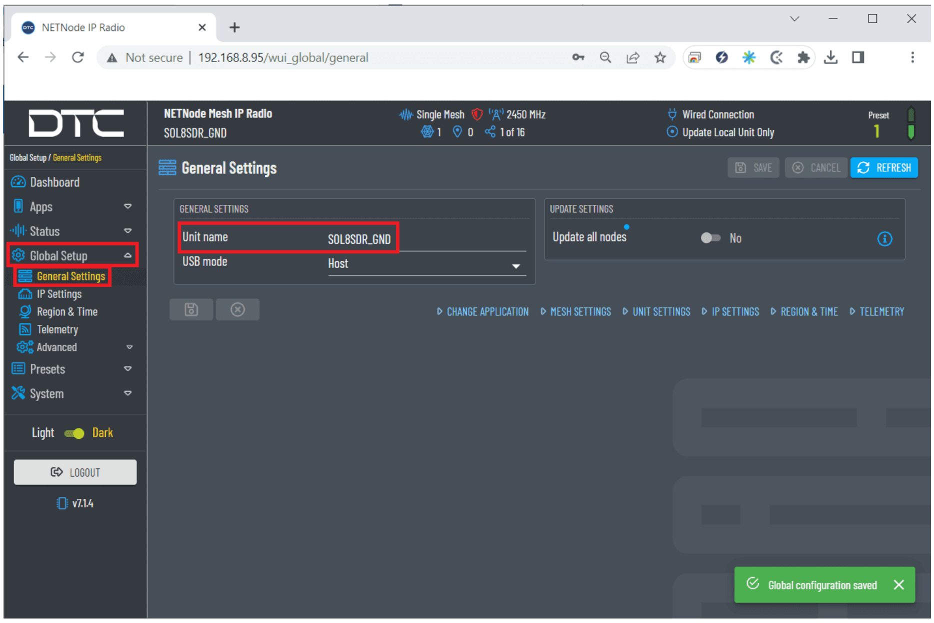
- **Basic radio configuration**

Once the website has been accessed, follow the steps below which show the parameters that need to be modified for a correct operation and pairing of the radios.

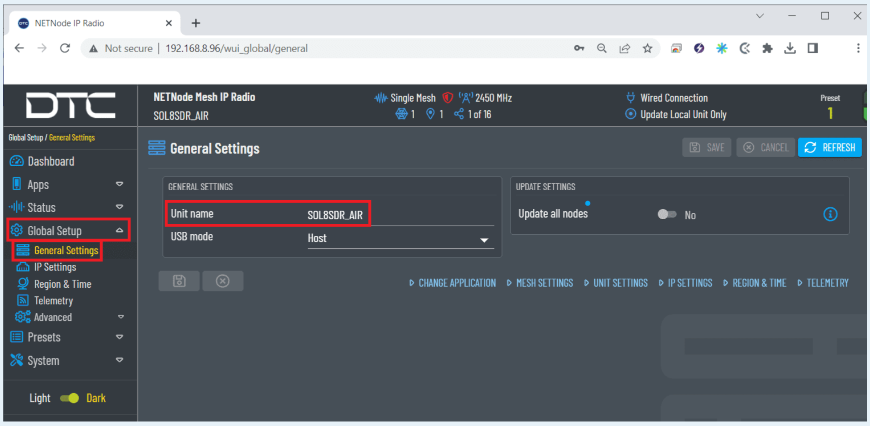
> ⓘ **Note**
>
> This is an example of the radio configuration linked to a **ground** unit.

> ⓘ **Note**
>
> After making any changes, the application will 'ask' to Save or Cancel the changes. An example is shown below:
>
> 
>
>
> **Save or Cancel changes**

1. Global Setup → **General Settings**: To easily identify each radio in a mesh, the user can rename the radio as desired:

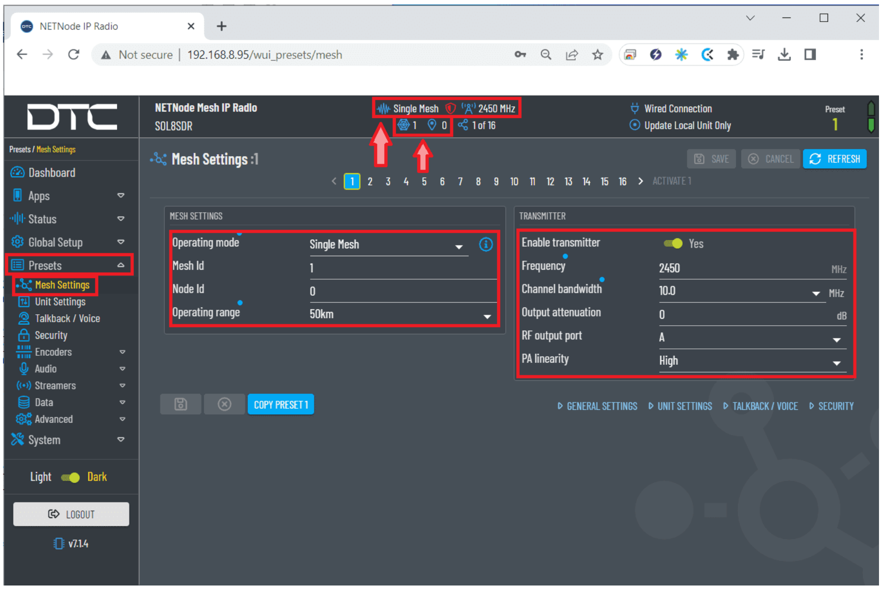


General settings configuration

> ### Note
>
> The radio related to the air unit also has its own customized name:
>
> 
>
>
> **General settings air configuration**

2. Presets.
   - **Mesh Settings**: Some of the parameters configured in this menu are always displayed at the top of the application.

     > ### ⚠ Caution
     >
     > It is recommended that software for all devices in a Mesh network should be at the same version to avoid potential compatibility issues.



Mesh settings configuration

- **Operation mode**: Select **Single Mesh**.
- **Mesh ID**: The **Mesh ID must be the same on all units in the Mesh network**. The Mesh ID tells the unit which network it belongs to, for example, all NETNodes on Mesh ID 1 will communicate with each other. The Mesh ID must be set to a **non-zero value**.
- **Node Id**: The **node ID must be unique in the Mesh network** for each device.

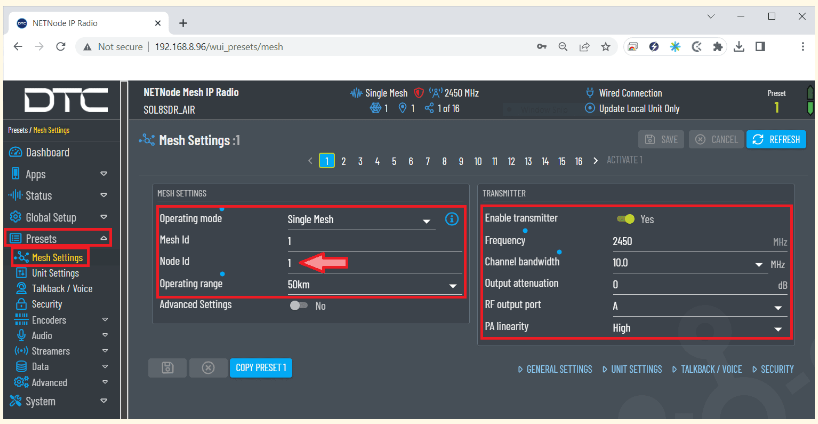
> **Note**
>
> A node can automatically reassign its Node ID at power up if it finds a conflict with an existing node.

- **Operating range**: A larger range allows the Mesh network to operate over a bigger distance at the expense of bitrate.
- **Enable transmitter**: Set the checkbox to switch the RF power on.
- **Frequency**: Set the desired transmission frequency. **2450 MHz recommended**.

> ⚠ **Warning**
>
> Be careful when choosing the frequency. The user may see interference with the Wifi frequency band, consult the radio spectrum.

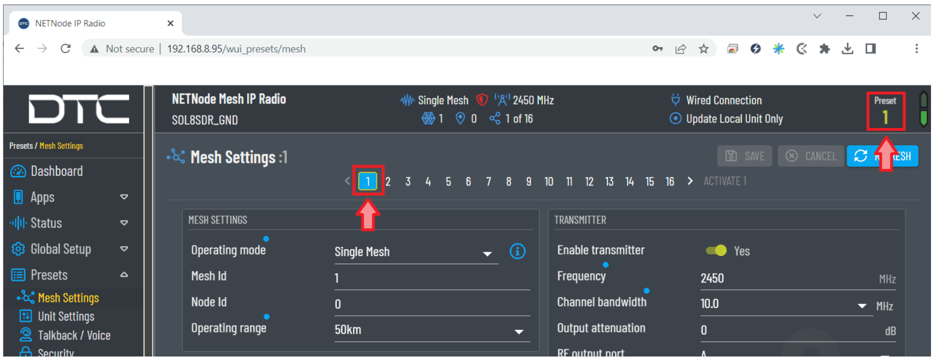
- **Channel bandwith**: Select the desired bandwidth from the drop-down list. Lower bandwidths provide greater range at the expense of data throughput. **10 MHz is recommended**.
- **Output attenuation**: The **level of attenuation** in **dB** that is applied to the output (from 0 to 32). **0 dB of attenuation is recommended**
- **RF output port**: The transmitter has two COFDM antennas, A and B. **A is selected as the output antenna by default**, but the user can select A or both if required.
- **PA linearity**: **High linearity** improves the COFDM shoulder performance at the expense of power consumption.

  Usually used when working with power amplifiers which must have excellent shoulder performance to operate, or for improved adjacent channel performance.

> ⚠ **Warning**
>
> To ensure proper communication between the two radios, the radio linked to the **air unit must have these same 'Mesh settings' except for the Node Id**, as each node in the mesh has its own Id (starting with Id 0):
>
> 
>
>
> **Mesh settings air configuration**

Moreover, there are up to **16 different preset configurations that can be setup**.

All these settings are made for preset 1, which is highlighted with a blue background in the 'Mesh settings' tab to indicate that it is active. In addition, a '**preset indicator**' with the current present is always displayed at the top right of the application, as shown in the figure below.

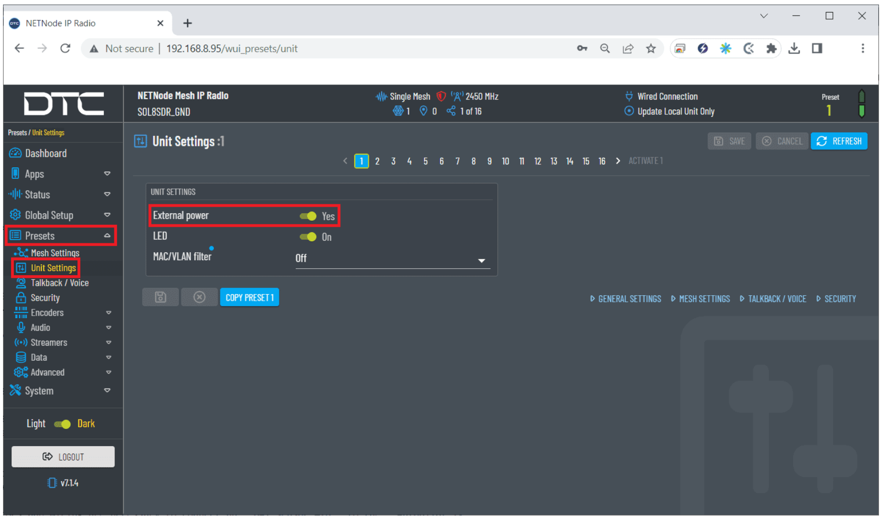


Current preset configuration

- **Unit Settings**:

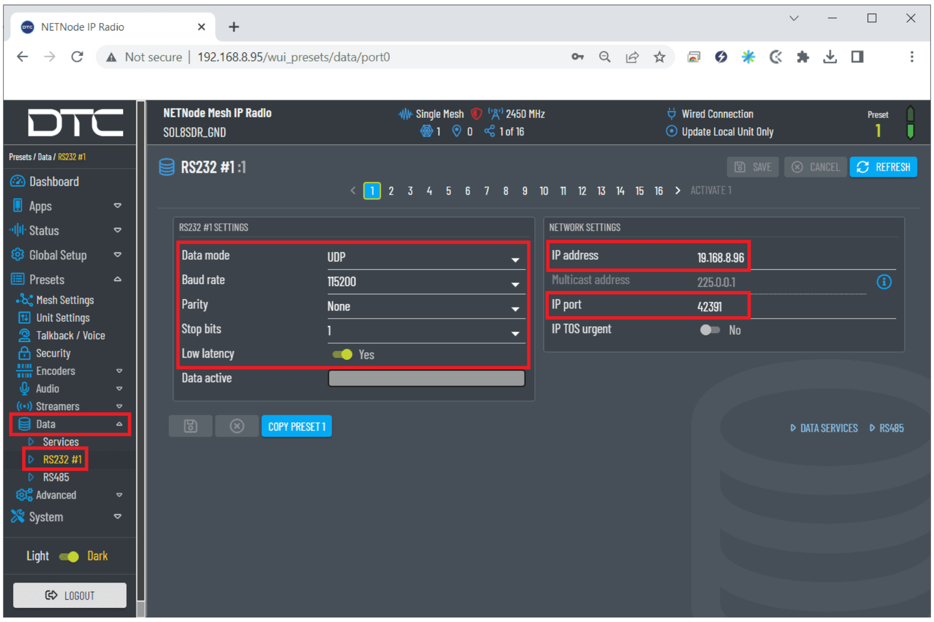
  External Power Enable: There is an external power output which can be used to supply 12VDC (1A) to an external device.

  This could be a camera, GNSS antenna or other device.



Unit settings configuration

3. **Data → RS232 #1**: In this menu the parameters of the RS232 port and the network settings are configured:



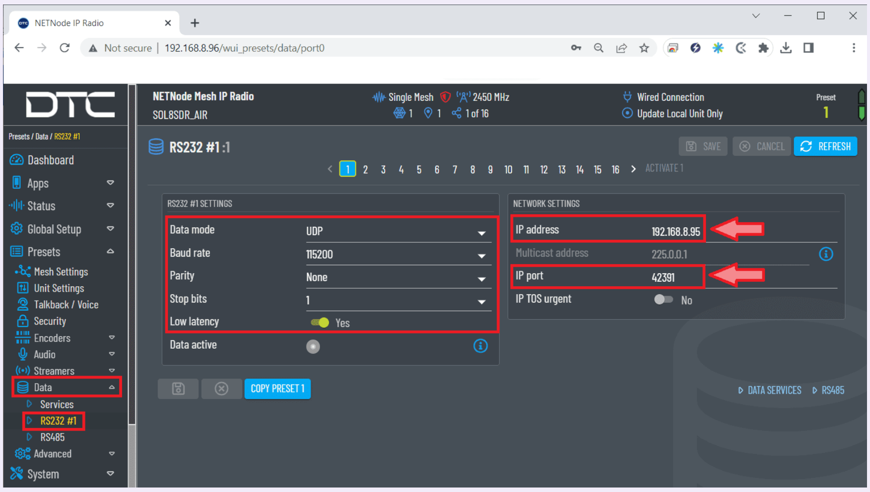
RS232 #1 configuration

- **Data mode**: **UDP** option is recommended. UDP packets are sent out and the system does not expect a reply. There is no way that the sending device can tell if the data arrived at the destination.
- The value of the **Baud rate**, **Parity** and **Stop Bits** parameters must be the same as those configured in **1x PDI Builder**.

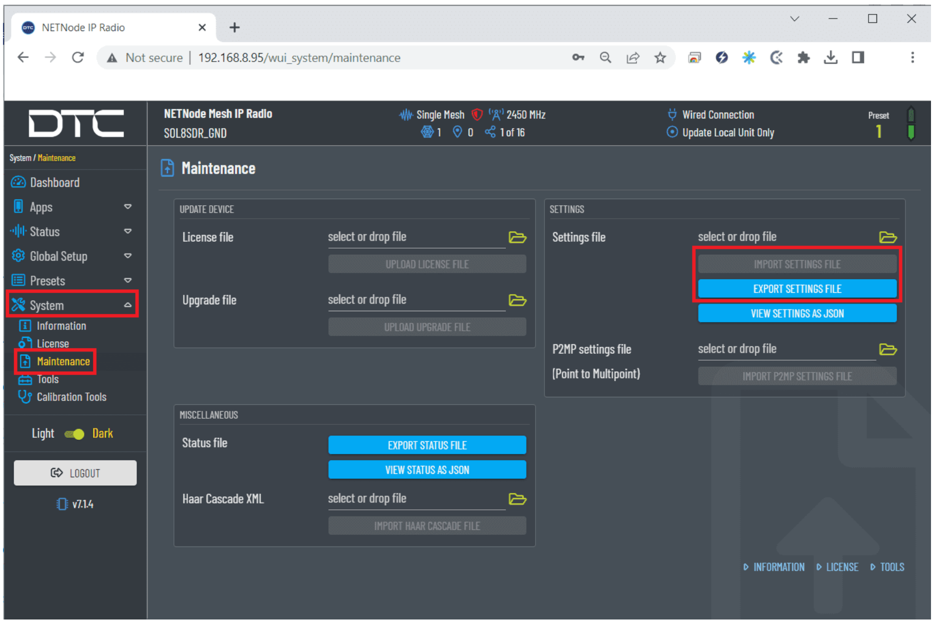
> ⓘ **Note**
>
> The data is assumed to be 8 bits.

- **Low latency**: Low latency will minimise delay at the expense of bitrate, so if set, data tranfer will be less prone to bursts. **Yes is recommended**.
- **IP address**: This should be the address of the radio receiving the data on the other end of the RS-232.

  In this case, as the **radio connected to the ground unit** is being configured, the **IP of the radio linked to the air unit will be set**.
- **IP port**: This set an IP port to and from which the data will be transferred. It must be **the same for both radios**.

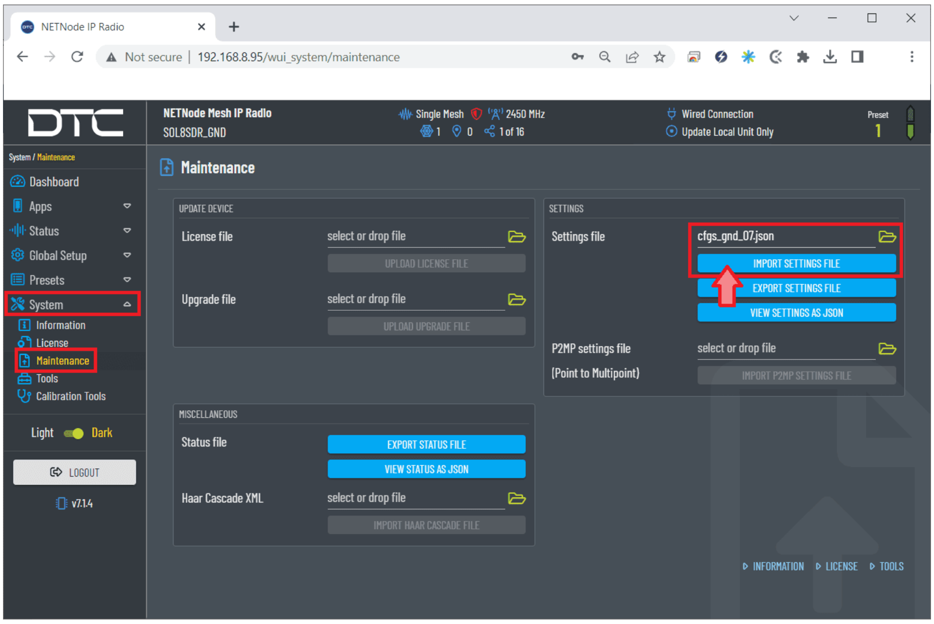
> **Important**
>
> For the **radio connected to the air unit**, the IP address to be configured is the **address of the radio linked to the ground unit**.
>
> 
>
>
> **RS232 #1 air unit configuration**

4. **System → Maintenance**: This menu allows to import and export radio configurations.



Maintenance configuration

To **import** a configuration into the radio, it is first necesarry to choose a configuration from the local storage by clicking on the icon. Then, the 'import button' will already be available (colored in blue) to click on and consequently import the selected configuration. An example is shown below:



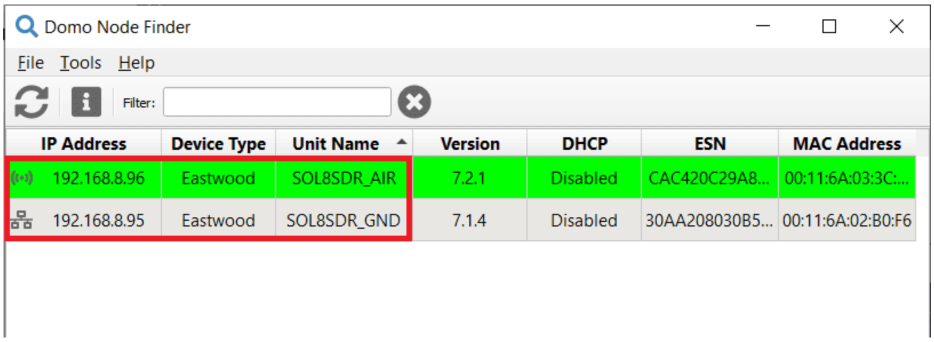
Configuration selected

- **Paired radios**

  Once both radios have been configured with these settings, they should be paired. Therefore, if we connect them to the power supply and only one of them to the computer, we can access the Domo Node Finder software or directly the Web Browser control application to check if they are correctly paired.

  - **Domo Node Finder** software

    When 2 radios are paired, they will both appear here:



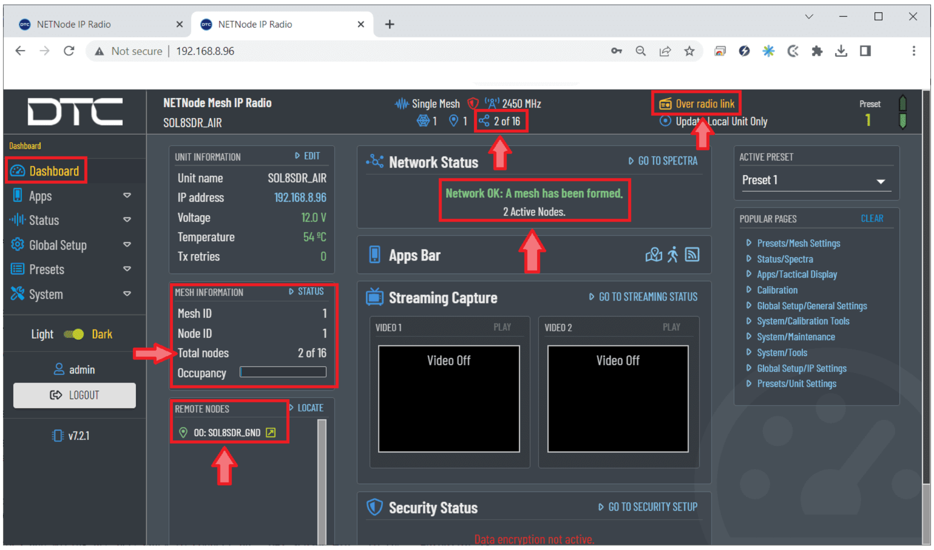
Domo Node Finder - Radios paired

As can be seen in the figure above, the connection type of each radio is indicated with different icons:

-  icon for the radio that is wiredly connected to the PC.
-  icon for the radio that is connected by link.

- **Web Browser control application**

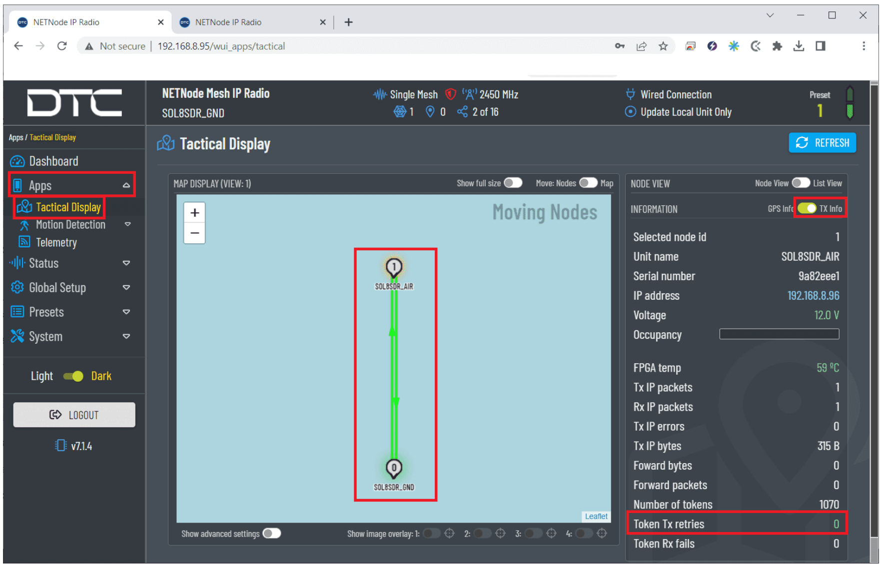
  When two radios are paired, this can be seen/checked directly in the '**Dashboard**' of both radios.



Radios paired - Dashboard

Furthermore, it can be seen that the above figure is related to the radio that is connected by link, as it is indicated at the top of the application with the label Over radio link.

- **Apps** → **Tactical Display**: Here the user can check the connection and the quality of the signal connection of both radios:



Radios paired - Tactical Display

  - **Map display**: The color of the link between nodes indicates the quality of the signal. The colors range from green (reliable link) to red (unreliable link). If no link is displayed, it means that communication has been lost.

- **TX info**: TX info should be selected to check the quality of the signal connection.
    - **Token Tx retries**: In a Mesh network, transmission is arbitrated by passing a token between nodes. This tab displays the number of token retries that have been needed for each node. It must be **0 with occasional 1 for a proper communication**.

      > ⚠ **Caution**
      >
      > Higher values will have an undesirable effect on system performance. If problems occur, check for interference and that there is no other Mesh system operating on the same or adjacent frequency.

Point-to-Multipoint configuration

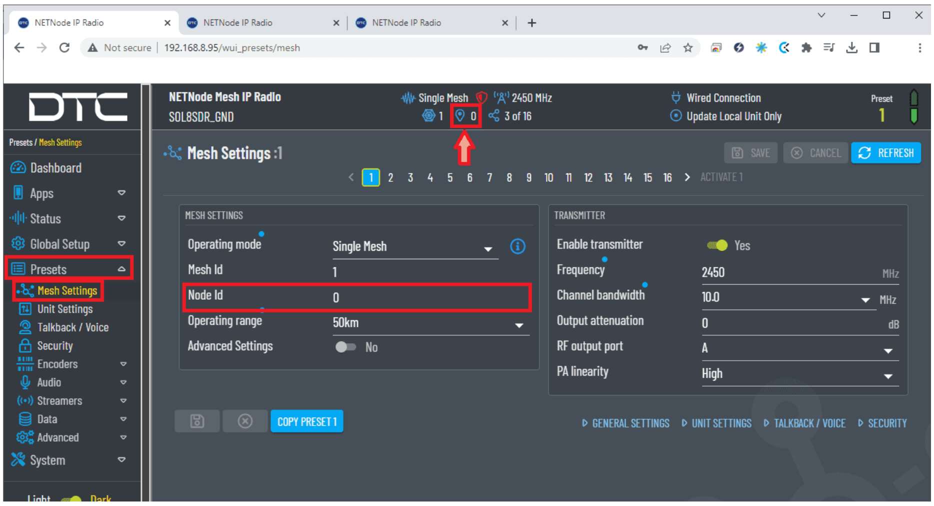
It is possible that the user wants to make a point-to-multipoint radio connection, i.e. there will be one radio sending commands to several radios, so there will be at least 3 radios.

The following is the configuration required for this type of connection.

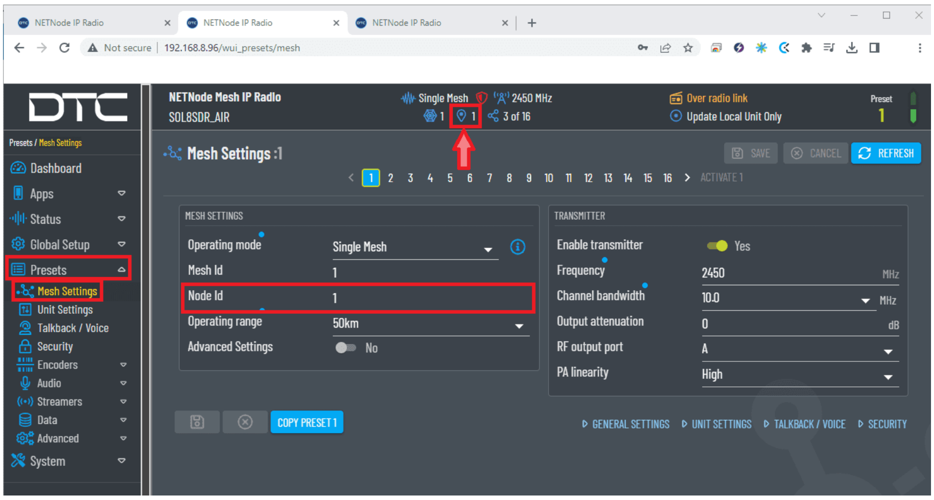
- **Radio configuration**
  The modifications to be made to the basic configuration explained above for the point-to-point application are detailed below.

  > ⓘ **Note**
  >
  > This example has been made with 3 radios (3 nodes in a mesh).

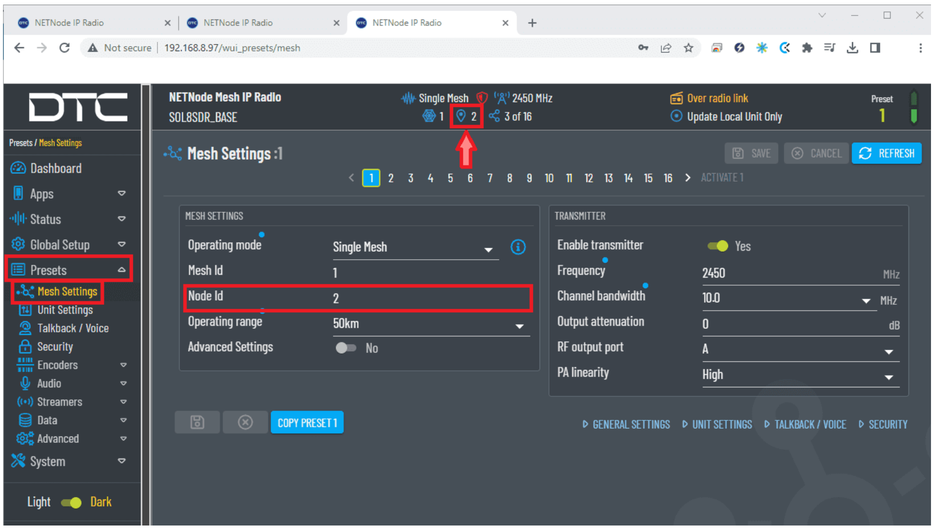
  1. **Presets** → **Mesh Settings**: The **Node ID** must be different for each node in the mesh.



Mesh settings ground configuration



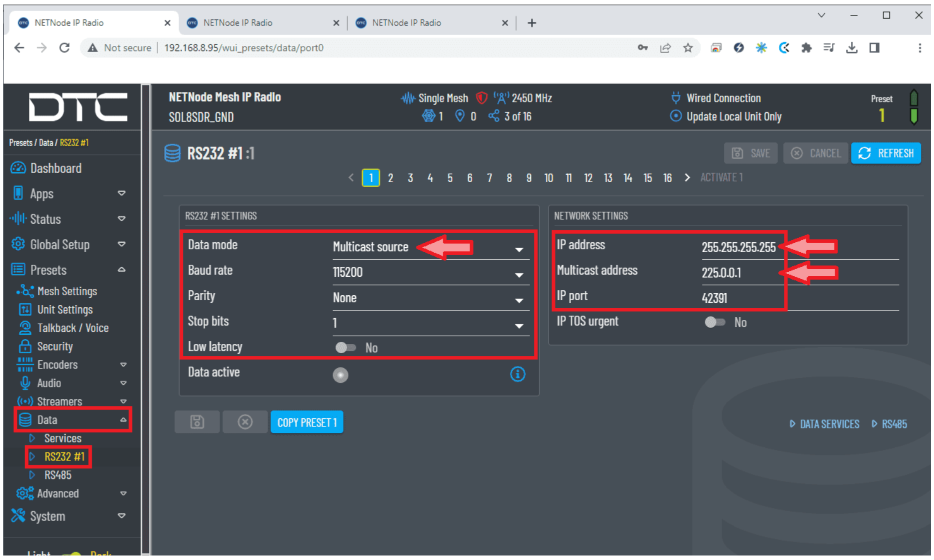
Mesh settings air configuration



Mesh settings base configuration

In the figures above, the user can see that the node ID is displayed at the top of the application at all times.

2. **Data** → **RS232 #1**: The **Multicast** data mode must be configured. This data mode allows a single node to send RS232/RS485 data to multiple nodes in the system. And it also creates a unicast data return channel. The **radio linked to the ground unit** is configured as the 'Point' that sends the commands.

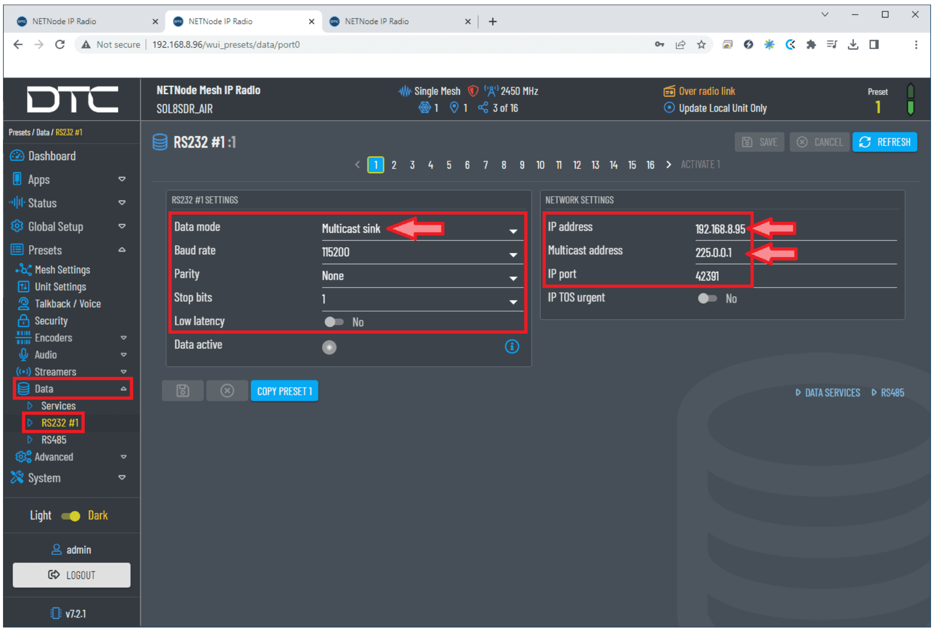


RS232 #1 ground configuration

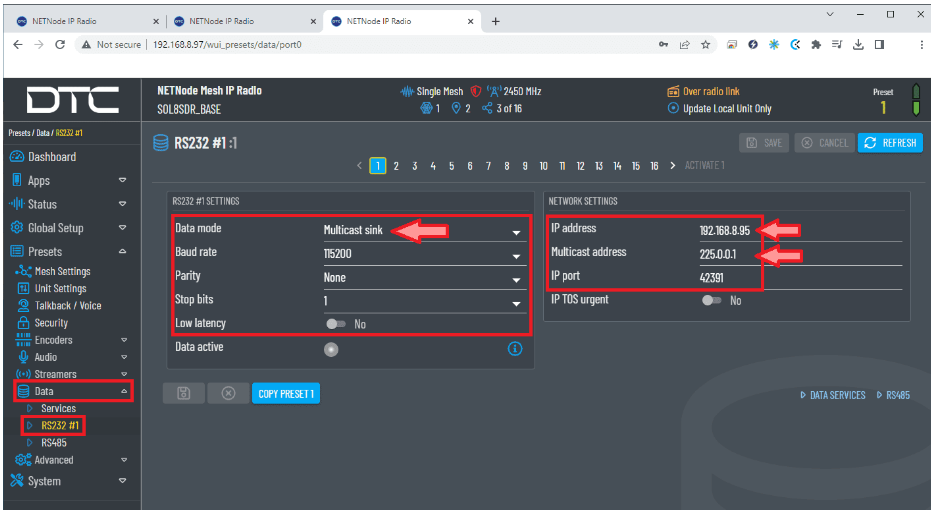
- **Data mode**: **Multicast source** must be selected.
- **IP address**: To send the data to all receivers the IP address must set to **255.255.255.255**.
- **Multicast address**: It must be **the same for all radios**, avoiding the 244.0.0.X address range.

  The address must be different from any multicast streaming and data channels.

Then, the **radios linked to the air and base units** receive those commands:



RS232 #1 air unit configuration



RS232 #1 base unit configuration

- **Data mode**: **Multicast sink** must be selected.
- **IP address**: The IP address of the **radio linked to the ground unit** is set.
- **Multicast address**: It must be **the same for all radios**, avoiding the 244.0.0.X address range.

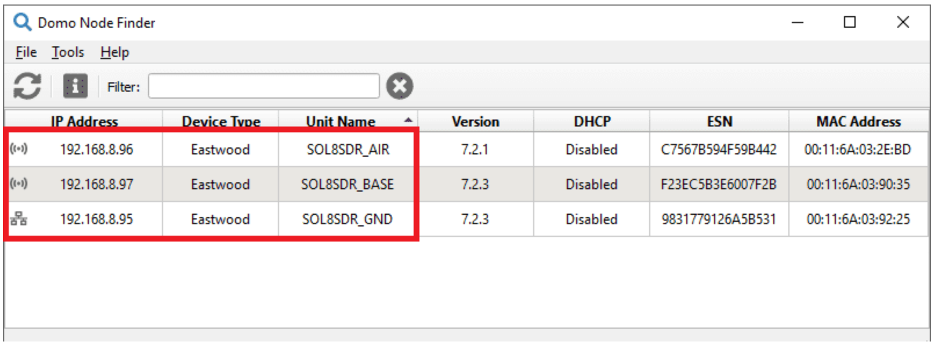
  The address must be different from any multicast streaming and data channels.

- **Paired radios**

  Once the radios have been configured with these settings, they should be paired. Therefore, if we connect them to the power supply and only one of them to the computer, we can access the Domo Node Finder software or directly the Web Browser control application to check if they are correctly paired.

- **Domo Node Finder** software

  When 3 radios are paired, they will appear here:

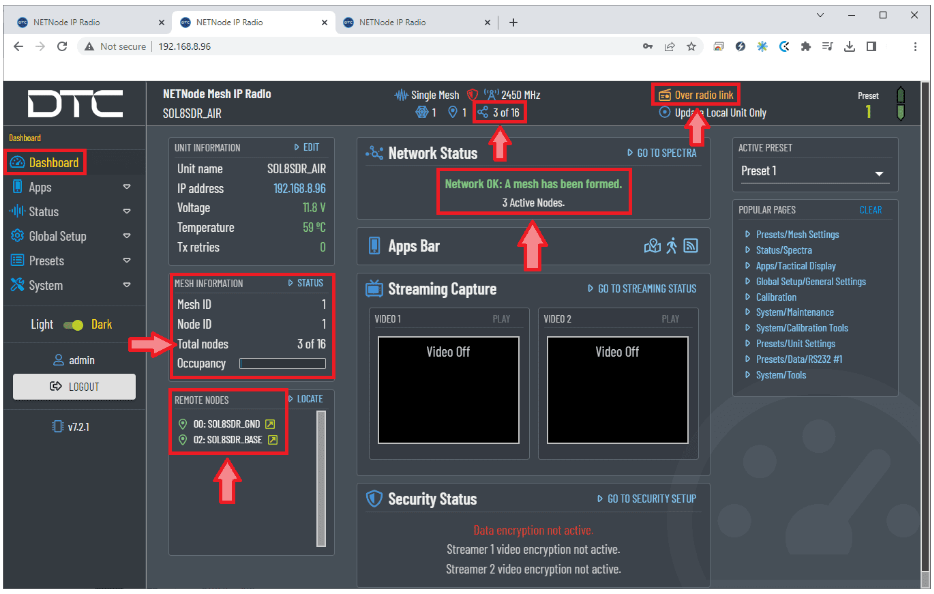


Domo Node Finder - 3 radios paired

  As can be seen in the figure above, there is 1 radio wiredly connected to the PC and 2 radios connected by link.

- **Web Browser control application**

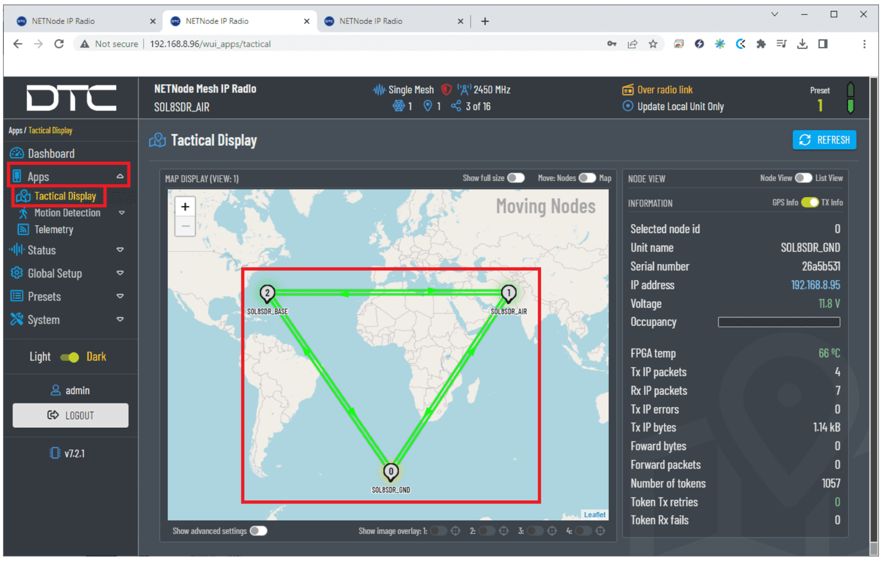
  When 3 radios are paired, this can be seen/checked directly in the '**Dashboard**' of the three radios.



Radios paired - Dashboard

  Furthermore, it can be seen that the above figure is related to a radio that is connected by link, as it is indicated at the top of the application with the label Over radio link.

  - **Apps** → **Tactical Display**: Here the user can check the connection and the quality of the signal connection of the radios:



Radios paired - Tactical Display

For more information on the configuration of **DTC** radios, please refer to the **DTC** documentation.

DTC radio configuration in 1x PDI Builder

The necessary configuration of **DTC** radio in **1x PDI Builder** is described in the External radios - Integration examples section of the **1x PDI Builder** manual.

Microhard pDDL900-ENC external

System Layout

It is possible to operate Microhard radios in two different ways, with or without amplifiers.

- **Microhard**
  The following image shows the standard connection between **Microhard** radios and **Autopilot 1x** for operation:

Microhard radios and Autopilot 1x operation

- **Microhard + Amplifier**

> ⓘ **Note**
>
> Amplifier information: **DDL900 Amplifier** model, 10W Linear Amplifier.

The following image shows the standard connection between **Microhard** radios, amplifiers and **Autopilot 1x** for operation:

Microhard + amplifier radios and Autopilot 1x operation

Hardware Installation

First, it is necessary to carry out the wiring of the **power** connector with the **4-pin power cable** supplied with the **Microhard** radio and with the power connector of the amplifier:

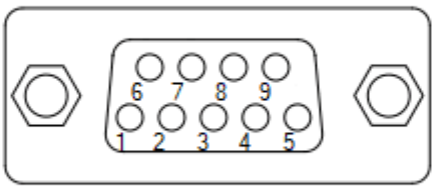
Female DC Power Jack connector

			Power Jack Connector
	PIN	**Signal**	**Signal**
RADIO	RED	Vin+ 9-30V	Power +
AMPLIFIER	1	DC 24-30V	
RADIO	BLACK	Vin-	Power -
AMPLIFIER	2	GND	

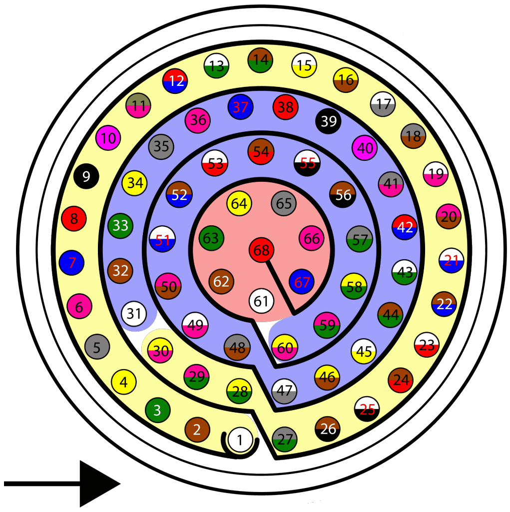
Danger

The specified supply voltage range for **Microhard** radios + amplifier is different from that of just **Microhard** radios. Please take this into account when powering.

Then, to physically connect the **Microhard** radio to **Veronte Autopilot 1x for operation**, connect an **RS-232** connector between the RS-232 port of the radio and the RS-232 of **Autopilot 1x** harness.



Serial port of Microhard radio



Autopilot 1x harness pinout

Serial port - Microhard radio		Autopilot 1x Harness		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
2	RXD	20	RS 232 RX	Pink-Brown
3	TXD	19	RS 232 TX	White-Pink
5	GND	21	GND	White-Blue

Microhard radio configuration

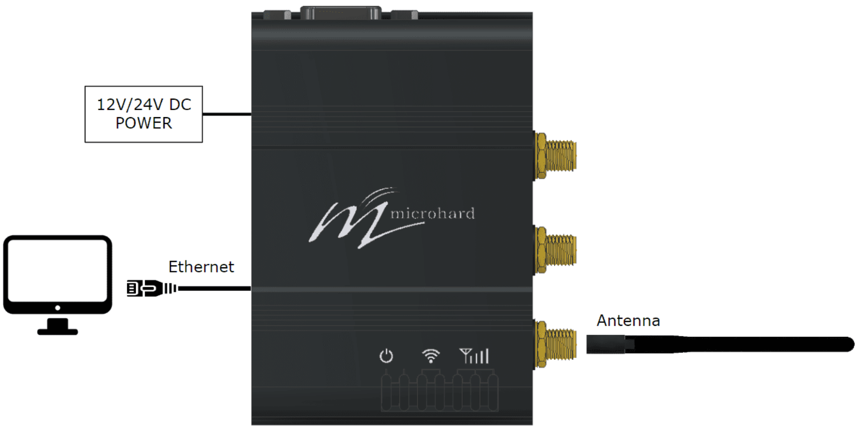
First steps

Microhard ports

- **Microhard without amplifier**
  1. Connect a suitable 900 MHz **antenna** to the **ANT1/MAIN** connector.
  2. Connect to the **4-pin molex power connector** the **4-pin power cable** supplied with the unit.
  3. Connect the power connector that has been previously wired with the cable to either 12V or 24V power.

     Then, once the radio is fully booted, the **Power LED** indicator (the one next to the power symbol) will be **solid blue**.
  4. To access the radio's WebUI for configuration, connect the **LAN** port (**not the WAN port**) to a PC, using an Ethernet cable.

  The connections should look like this:

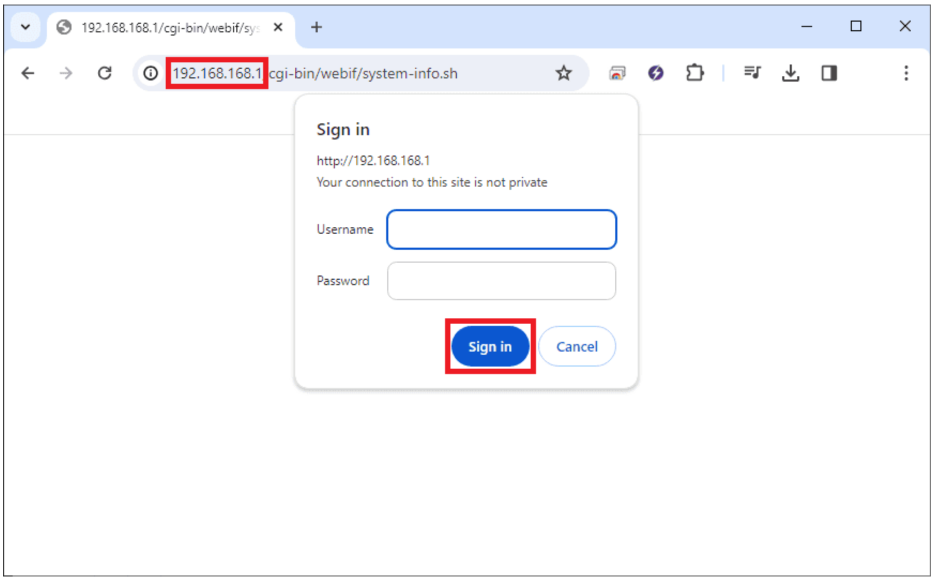


First steps connection

5. To open the **Microhard** WebUI, open a browser and enter the IP address of the radio into the address bar.

> ⓘ **Note**
>
> If users have problems accessing the **Microhard** radio WebUI, it may be because the PC's network connection settings (IP address and subnet of the adapter) are not configured properly. For further details, consult Microhard radio troubleshooting section below.

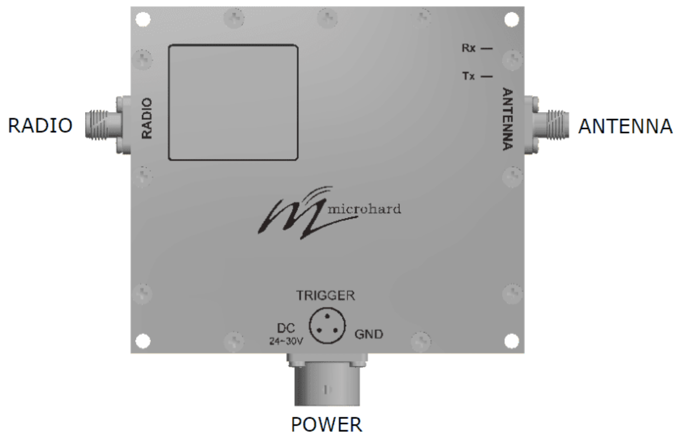
6. The website will then ask for a **Username** and **Password**. Enter the factory defaults:
   - Username: admin
   - Password: admin



Open WebUI

7. Click **Sign In** and the WebUI will open.
8. Once successfully logged in for the first time, the WebUI will force a password change.

- **Microhard with amplifier**

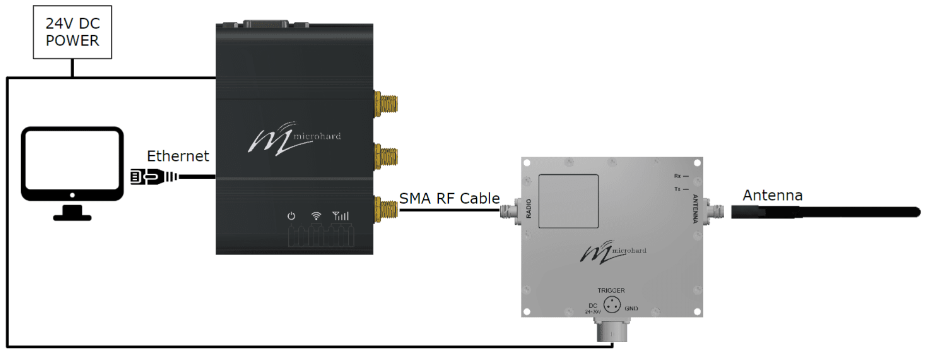


Amplifier ports

1. Connect to the **ANT1/MAIN** connector of the radio a **SMA RF cable**.
2. Connect this **SMA RF cable** to the amplifier **RADIO** port. Connect a suitable 900 MHz **antenna** to the amplifier **ANTENNA** port.
3. Connect to the **4-pin molex power connector** of the radio the **4-pin power cable** supplied with the unit.
4. Plug to the **POWER** port of the amplifier, the power connector supplied with it.
5. Connect the power connector that has been previously wired with the radio cable and the amplifier connector to **24V** power.

   Then, once the radio is fully booted, the **Power LED** indicator (the one next to the power symbol) will be **solid blue**.
6. Now the steps to follow are the same as **from step 4. of a Microhard without amplifier**, described above.

The connections should look like this:



First steps connection + amplifier

For more information on the configuration of **Microhard** radios, please refer to the **Microhard** documentation.

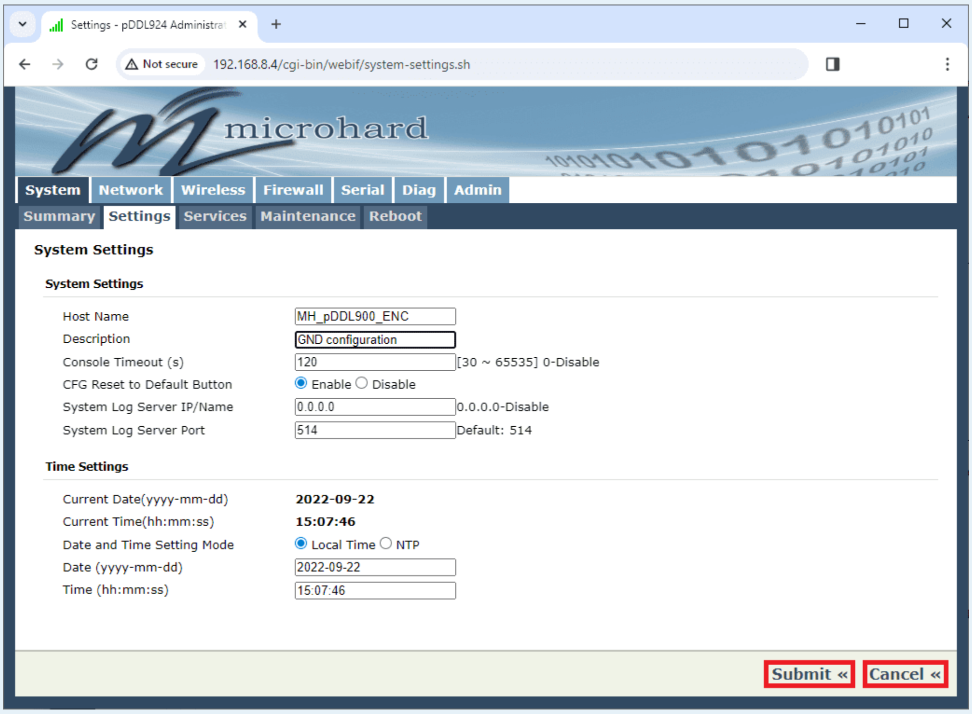
Basic radio configuration

Once the website has been accessed, follow the steps below which show the parameters that need to be modified for a correct **point-to-point** configuration and pairing of the radios.

> **Important**
>
> This example describes the parameters to be entered for both the radio linked to the 1x ground unit and the radio linked to the 1x air unit.
>
> If the values are common to both radios configurations, only one of them will be detailed.

ⓘ Note

After making any changes, the application will 'ask' to Submit or Cancel the changes. An example is shown below:

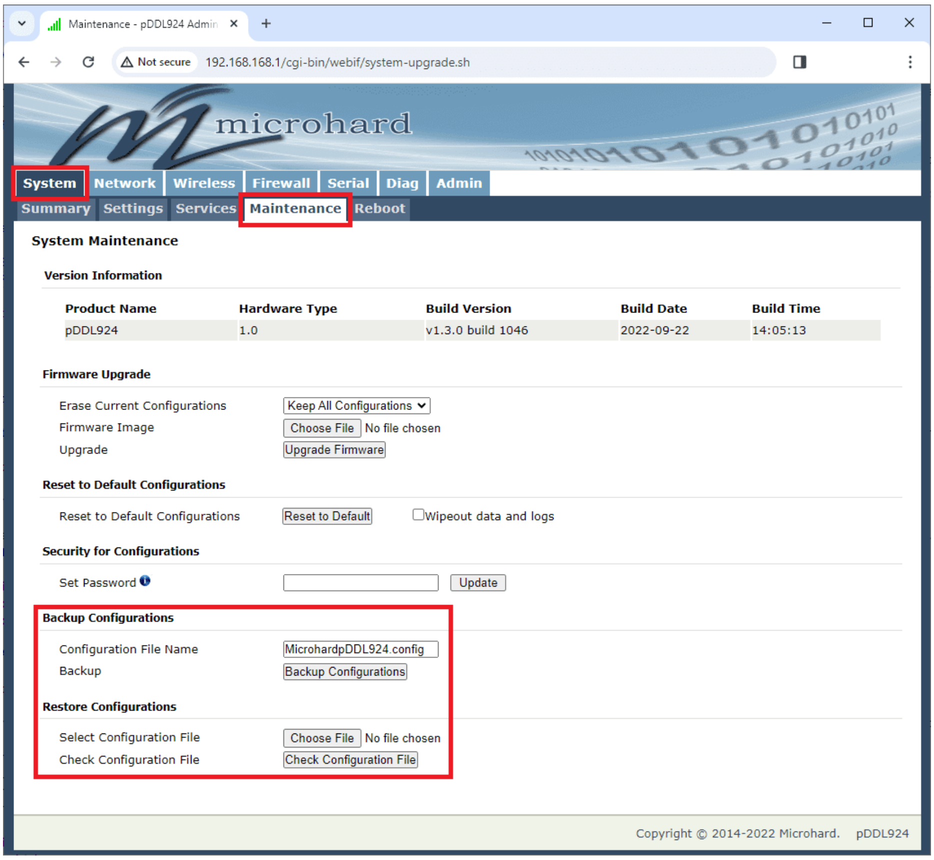


Submit or Cancel changes

1. **System**
   - **Maintenance**: From this menu, users can export (backup) and import (restore) configurations.
     - **Backup Configurations**: The radio configuration can be backed up to a file at any time using the Backup Configuration feature.
     - **Restore Configurations**: Using this option, a previously 'backed up' configuration can be uploaded to the radio.

ⓘ Note

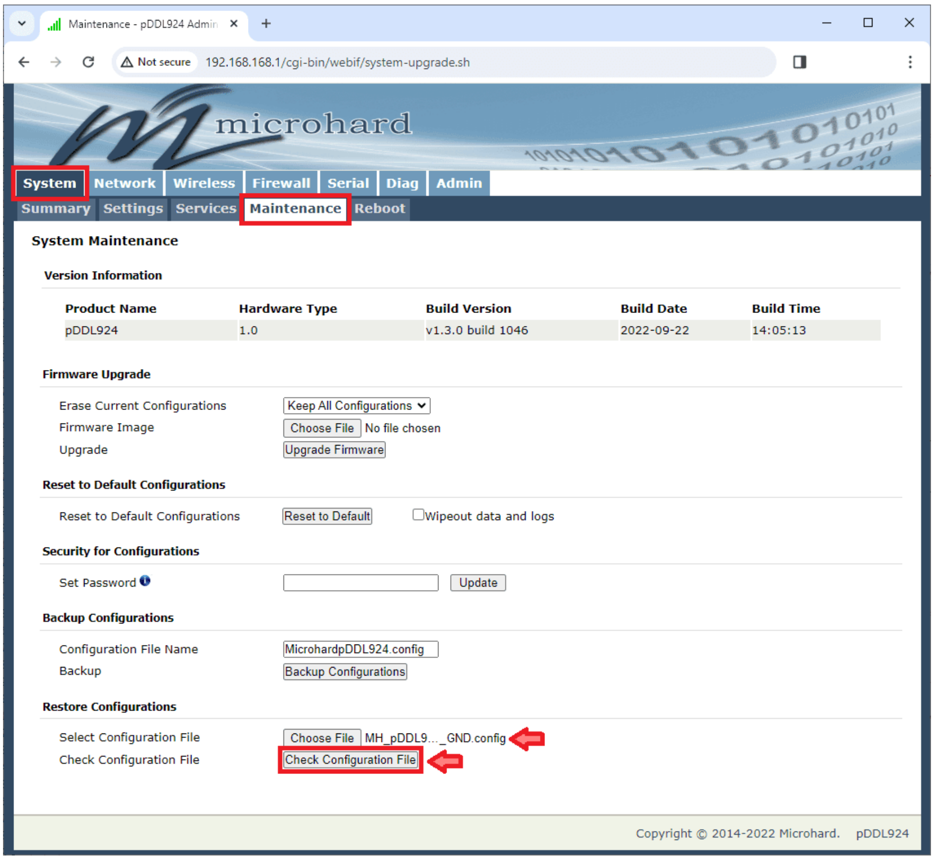
A password can be added to backup and restore files. If the password is lost, files that have been backed up with a password cannot be restored.



System Maintenance

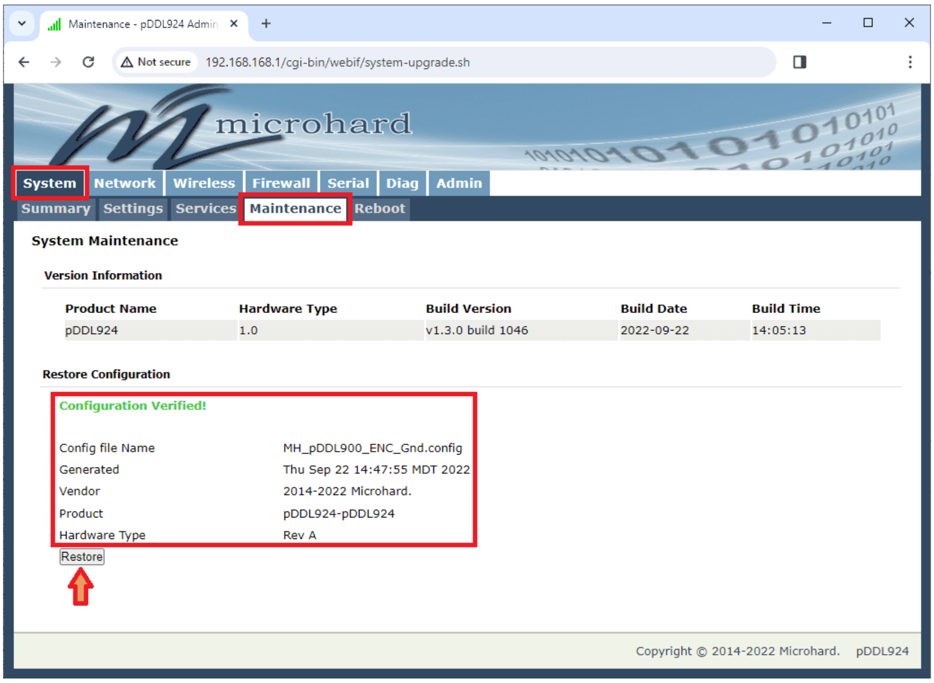
To **import** (restore) a configuration into the radio, it is first necessary to choose a configuration from the local storage by clicking on `Choose File`.

Then, once the configuration is loaded, click on the `Check Configuration File` button.



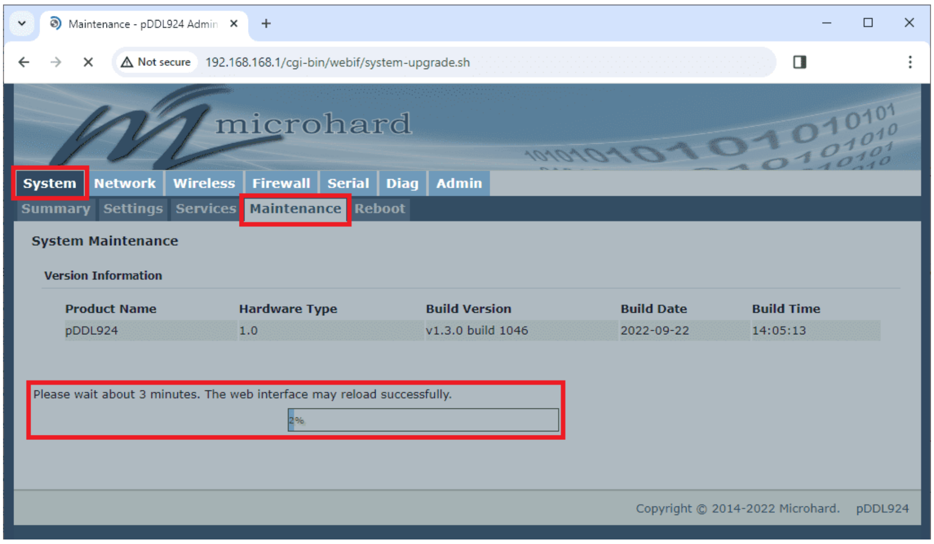
System Maintenance - Restore Configuration 1

After that, if the file is correct, a new window will appear to restore this configuration.



System Maintenance - Restore Configuration 2

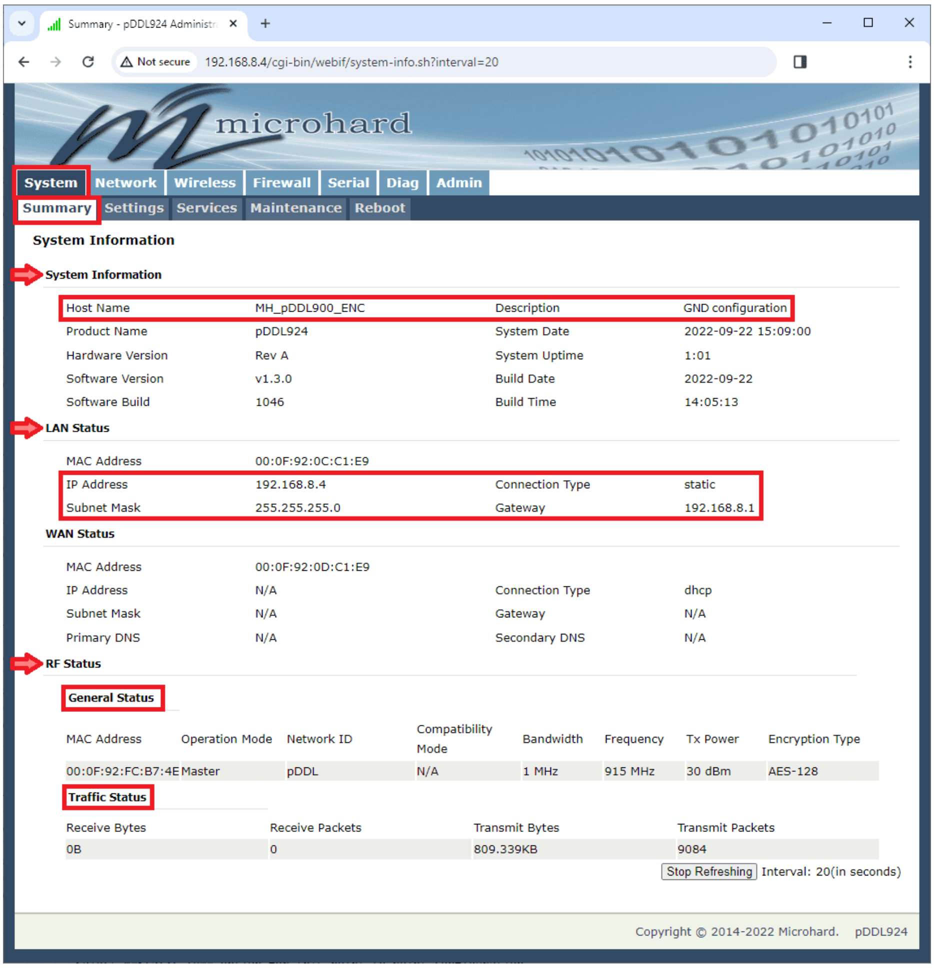
Finally, the WebUI will reloaded to applied the configuration.



System Maintenance - Restore Configuration 3

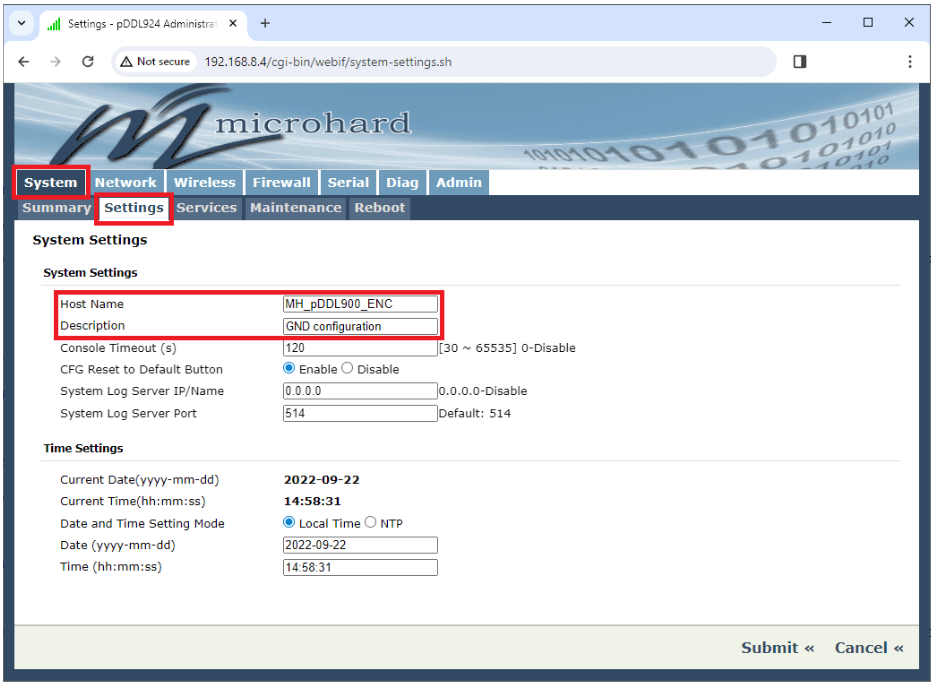
> **Error**
>
> If the restored configuration has a different IP address assigned to the radio, users may have to change again the **adapter settings** on the PC to be able to access the WebUI with the new IP address of the radio.

- **Summary**: This screen is displayed immediately after the initial login, showing a summary and status of all radio functions on a single display.
  This information includes system status, LAN network information and settings, version information, radio connection status, etc.

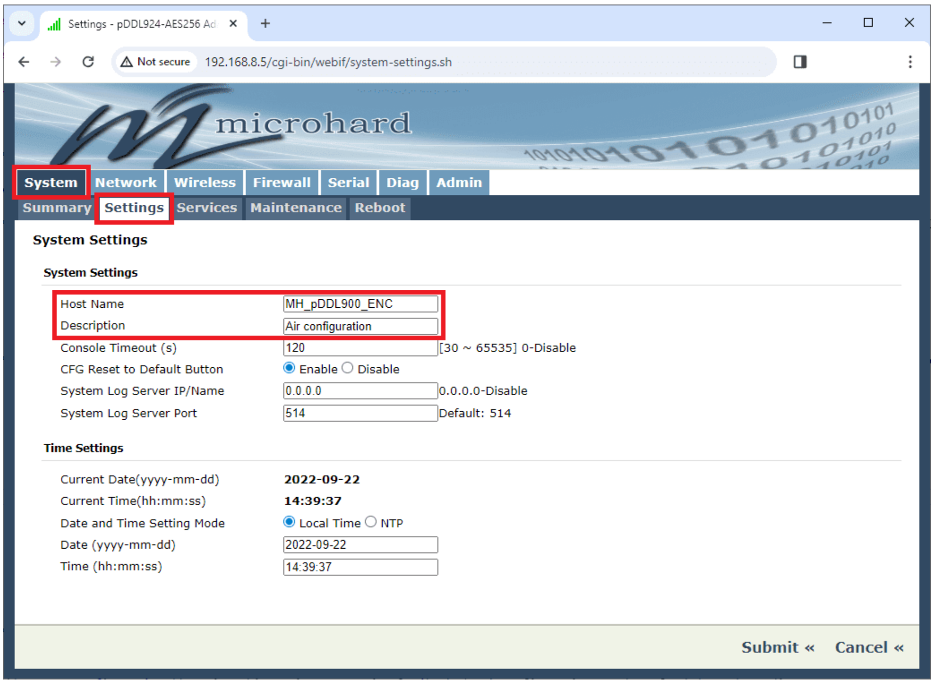


System Information

- **Settings**: To easily identify each radio in a mesh, the user can rename the radio as desired by configuring the **Host Name** and **Description** parameters:



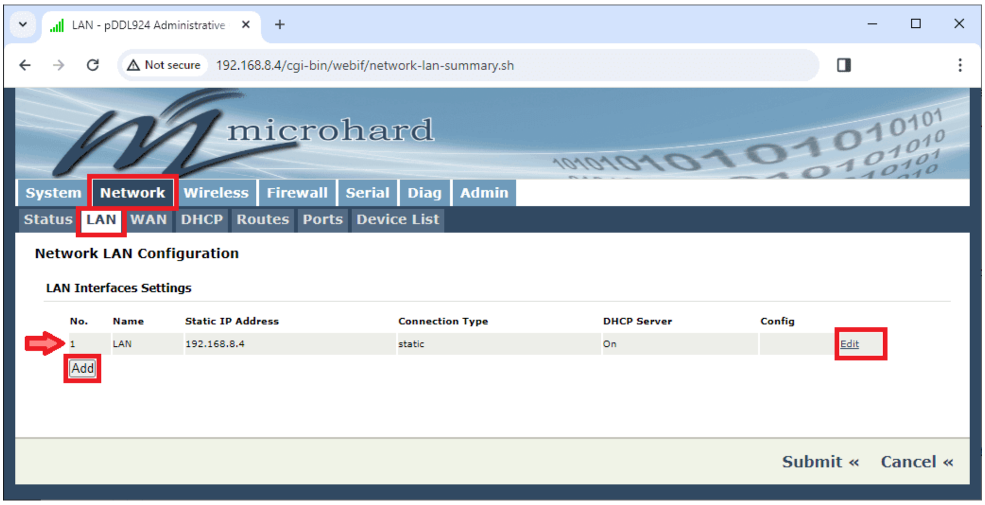
System Settings - GND unit



System Settings - Air unit

2. **Network** → **LAN**: Ethernet LAN port of the radio is for connecting devices to a local network.

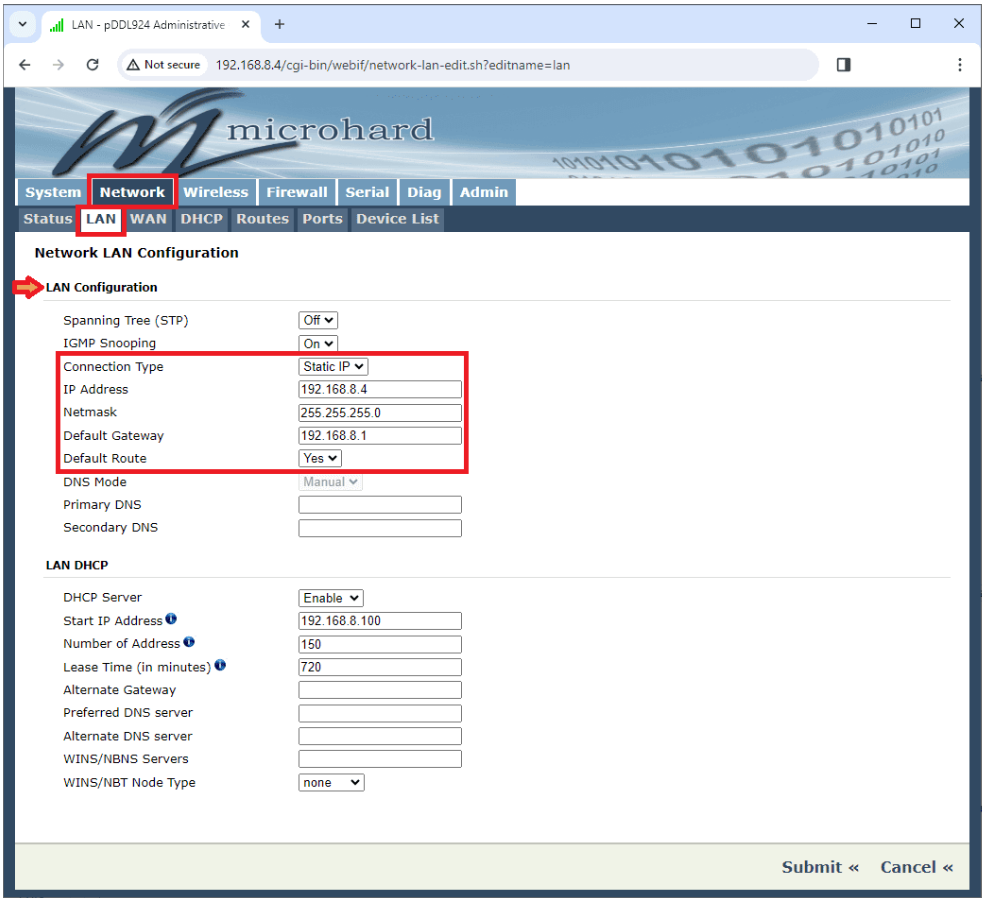
   By default, this port has a static IP address, and also, it is running a DHCP server to provide IP addresses to devices that are connected to the physical LAN port (directly or through a switch).



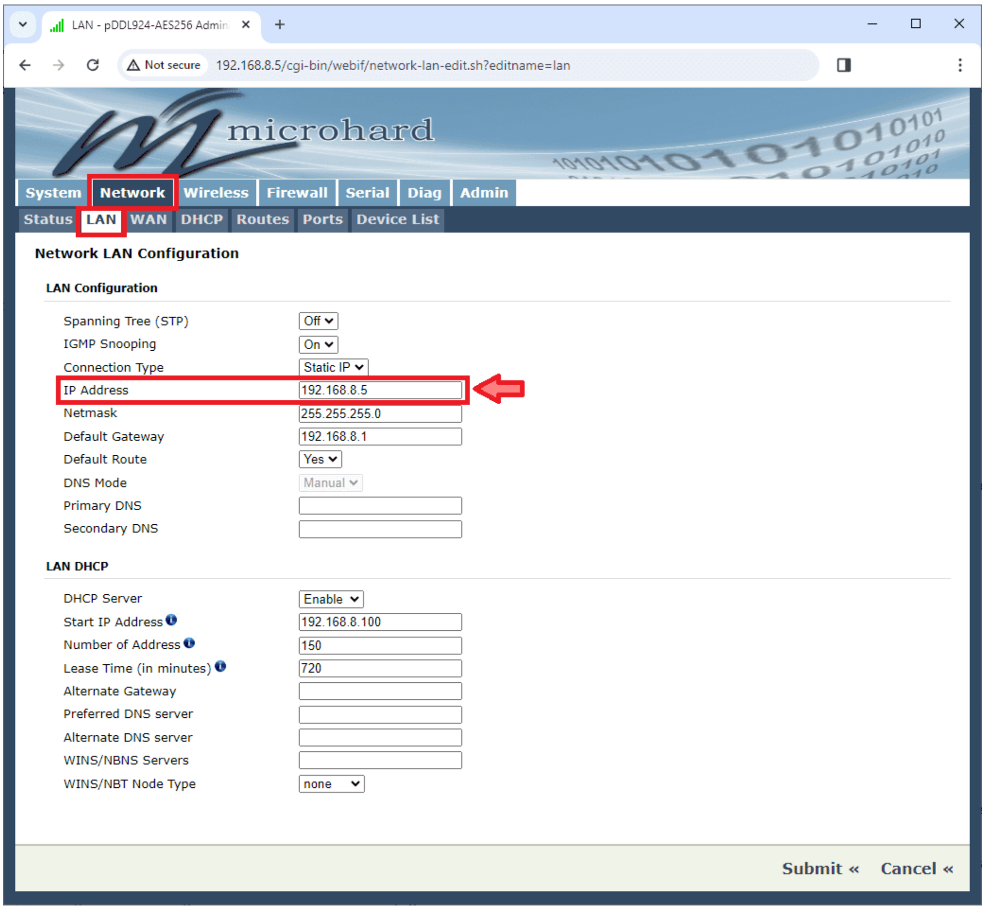
LAN Interfaces Settings

By selecting the `Edit` or `Add` buttons, the LAN network interface can be configured or additional LAN interfaces can be created.

LAN Configuration:



LAN Configuration - Ground unit



LAN Configuration - Air unit

- **Connection Type**: To determine whether the radio will obtain an IP address from a DHCP server on the connected network or whether a static IP address will be entered. Select the **Static IP** option for this point-to-point configuration.

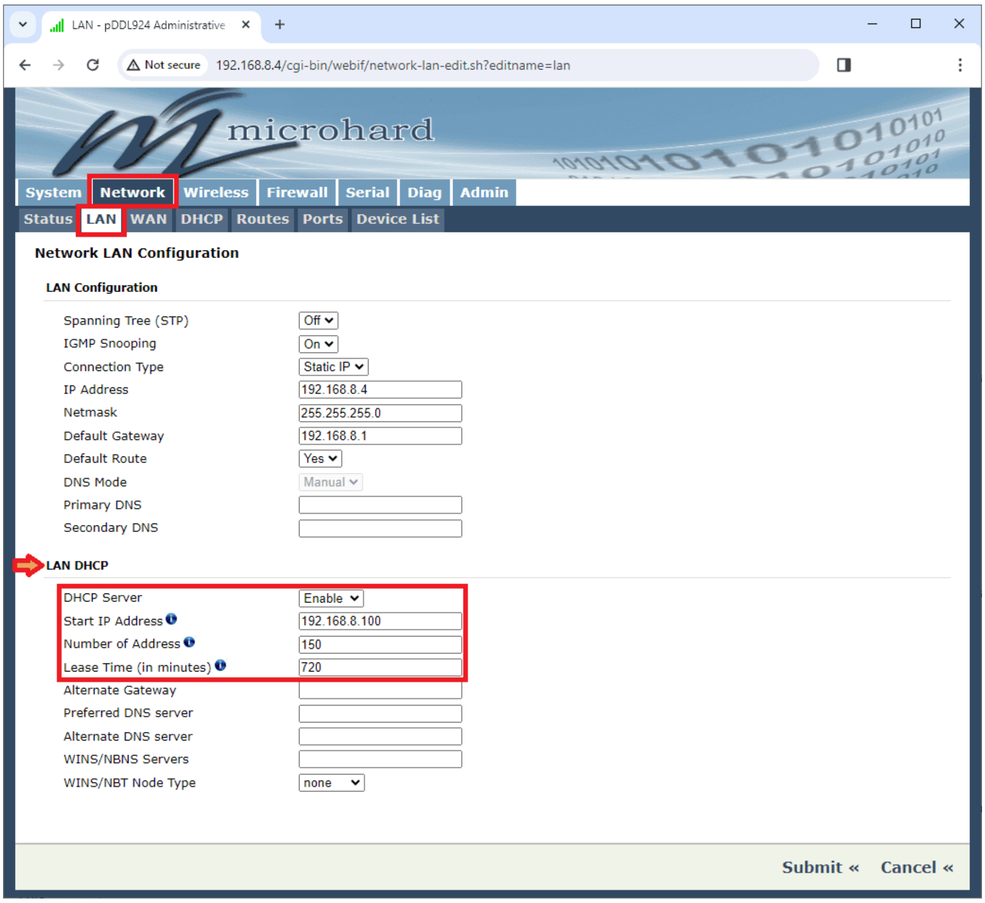
  Since the Static IP option has been chosen, the following fields must also be entered.
- **IP Address**: A valid IPv4 address for the network used must be entered in this field.
  - For the radio linked to the **ground** unit ⇒ **192.168.8.4** has been entered.
  - For the radio linked to the **air** unit ⇒ **192.168.8.5** has been entered.
- **Netmask**: The Network Mask for the network must be entered. The default netmask **255.255.255.0** is normally left.
- **Default Gateway**: If the radio is integrated into a network that has a defined gateway, then, as with other hosts on the network, the IP address of this gateway will be entered in this field.

  In case the user is using a **PCS**, **192.168.8.1** should be set here.
- **Default Route**: This parameter allows the user to set this interface as the default route in the routing table.

  In cases where the LAN is the primary connection, this would be set to **Yes**.

> **Important**
>
> Once the IP Address has been changed, users will have to type the new address in the browser in order to continue with the configuration.

LAN DHCP: A radio can be configured to provide Dynamic Host Control Protocol (DHCP) service to all connected devices (wired or wireless).



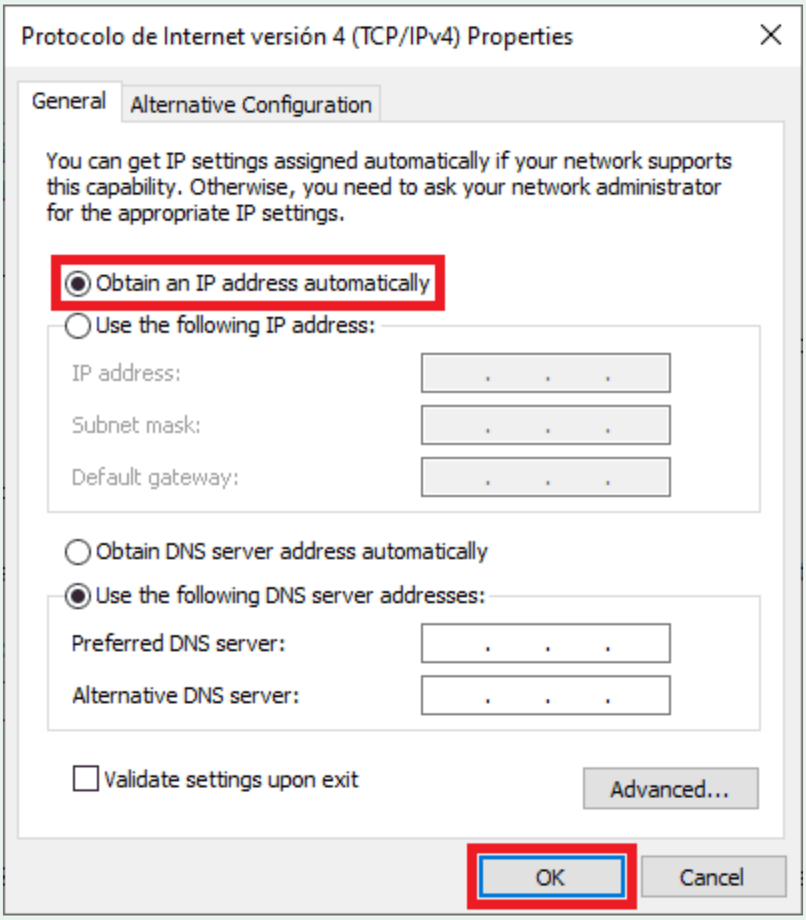
LAN DHCP

- **DHCP Server**: Enables/Disables the DHCP service for devices connected to the LAN port. Therefore, devices that are connected to the physical Ethernet LAN ports will be assigned an IP by the radio.

> ⚠ **Caution**
>
> Before enabling this service, check that there are no other devices, either wired or wireless, with an active DHCP Server service.

Hint

At this point, if **Enabled** and if users have previously set the PC to the static IP address on the same subnet of the radio, they can now change the **adapter settings** back to "Obtain an IP address automatically". Consequently, users will no longer have to change this setting on the PC every time they want to access the radio's WebUI.



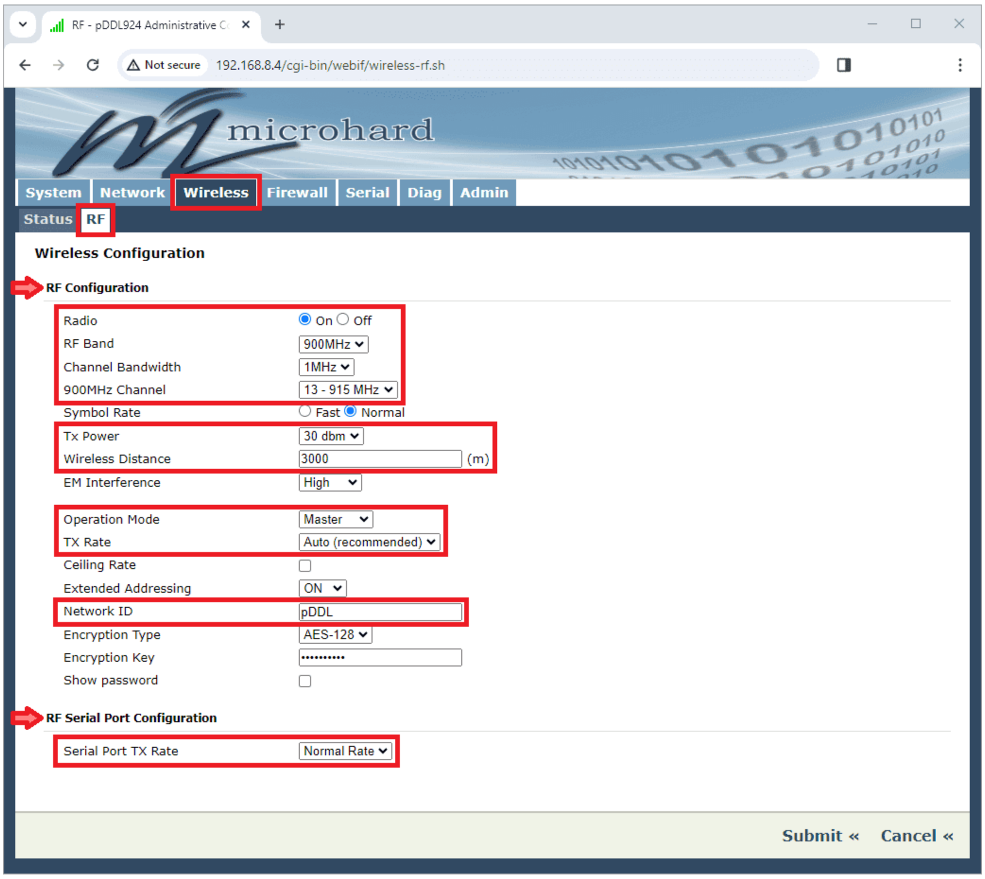
Adapter settings - IP address automatically

- **Start IP Address**: Select the starting DHCP address from which IP addresses will be assigned. The first octets of the subnet will be pre-set based on the LAN IP configuration, and cannot be changed. We enter here **192.168.8.100**.
- **Number of Address**: Set the maximum number of IP addresses that can be assigned by the radio. By default, 150 (this is an integer value).

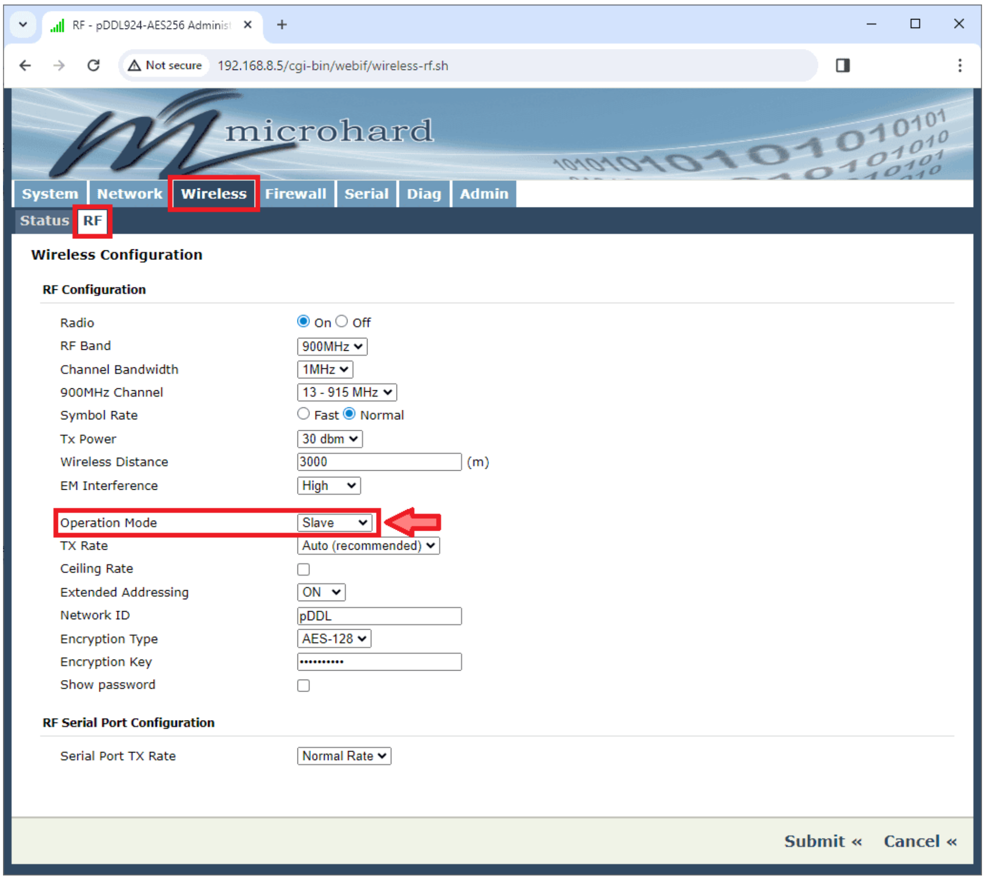
- **Lease Time (in minutes)**: The DHCP lease time is the amount of time before a new request for a network address must be made to the DHCP Server. Defaults to 720 minutes.

3. **Wireless** → **RF**: Allows to configure the radio module.

> **Important**
>
> Ensure that the **RF Band**, **Channel Bandwidth** and **Frequency-Channel** are set the same on each module.



Wireless Configuration - Ground unit



Wireless Configuration - Air unit

RF Configuration:

- **Radio**: Turns the radio module on/off.
- **RF Band**: Select the desired RF band to work, 900 MHz or 2.4 GHz. In this case, we select **900 MHz**.
- **Channel Bandwidth**: Select the channel bandwidth from the list. Generally a larger channel has higher throughput, at the cost of sensitivity, while a smaller channel tends to be more robust, but at the cost of throughput. **1 MHz** is recommended.

  > ⚠ **Caution**
  >
  > Refer to the radio specifications (**Microhard** documentation) in order to see the relationship and performance between channel bandwidth, throughput and sensitivity.

- **900 MHz Channel**: Set the Channel-Frequency. The frequency displayed is the center frequency and is available in 1 MHz increments. The values shown will vary with the Channel Bandwidth selected above. **13-915 MHz** is recommended.

> **Important**
> - This value must be **the same on each unit in a network**.
> - The noise floor of the specified channel will dramatically affect the link quality, it is essential to select the cleanest channel for superior performance.

- **Tx Power**: This setting establishes the transmit power level that will be presented to the antenna connector of the radio. Select **30 dbm** here.

  > **Caution**
  > For bench or proximity testing, it is best to use a lower power setting to avoid RF saturation.

- **Wireless Distance**: This parameter allows users to set the expected distance the wireless signal should travel.

  The radio sets various internal timeouts to account for this travel time. Longer distances will require a higher setting, and shorter distances may work better if the setting is reduced. **3000 m** is set.
- **Operation Mode**: A radio in **Master** mode can provide a wireless data connection to multiple slaves/remotes.

  And, in **Slave** mode, it can maintain a wireless connection, i.e. to a Master.
  - For the radio linked to the **ground** unit ⇒ **Master** mode must be selected.
  - For the radio linked to the **air** unit ⇒ **Slave** mode must be selected.
- **TX Rate**: This setting determines the type of modulation and, in turn, the rate at which the data will be transferred wirelessly.

  The default and **recommended** setting for both Master and Slave units is '**Auto**'.

  In 'Auto', the unit will transfer data at the highest possible rate depending on the receiving signal strength (RSSI).
- **Network ID**: Each radio module network must have a unique Network ID. In this case, **pDDL** has been entered.

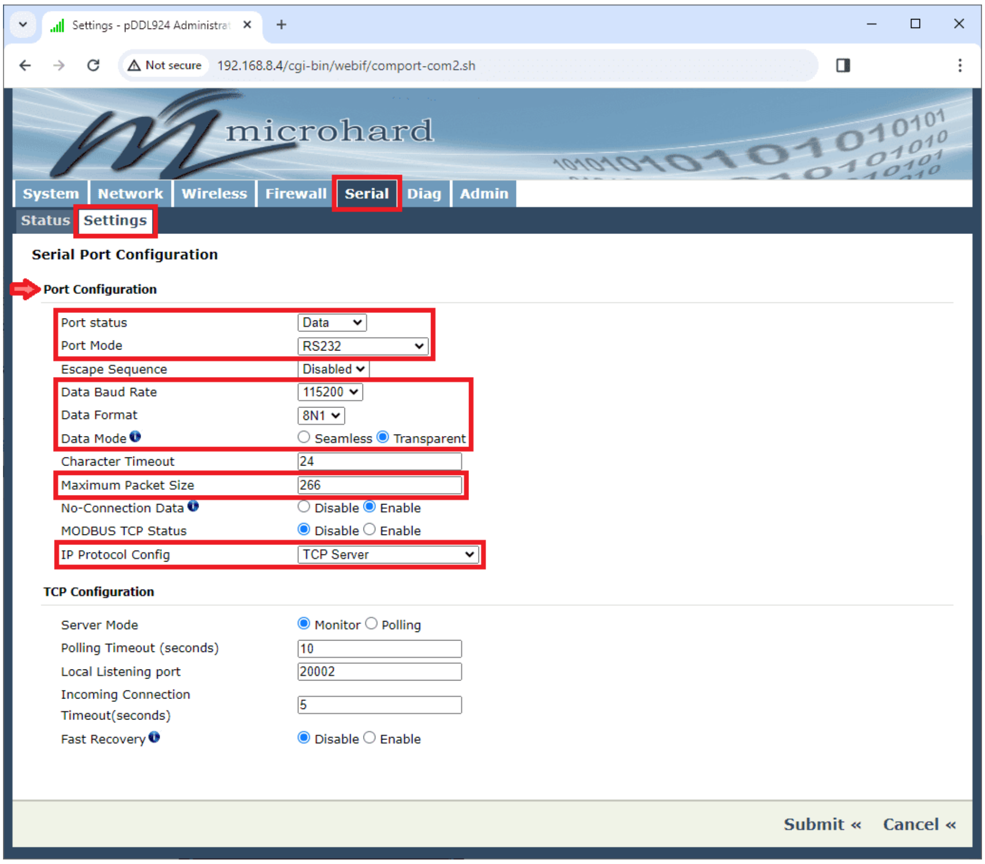
> **Important**
>
> This Network ID must be configured **the same in each unit on the network**.

RF Serial Port Configuration:

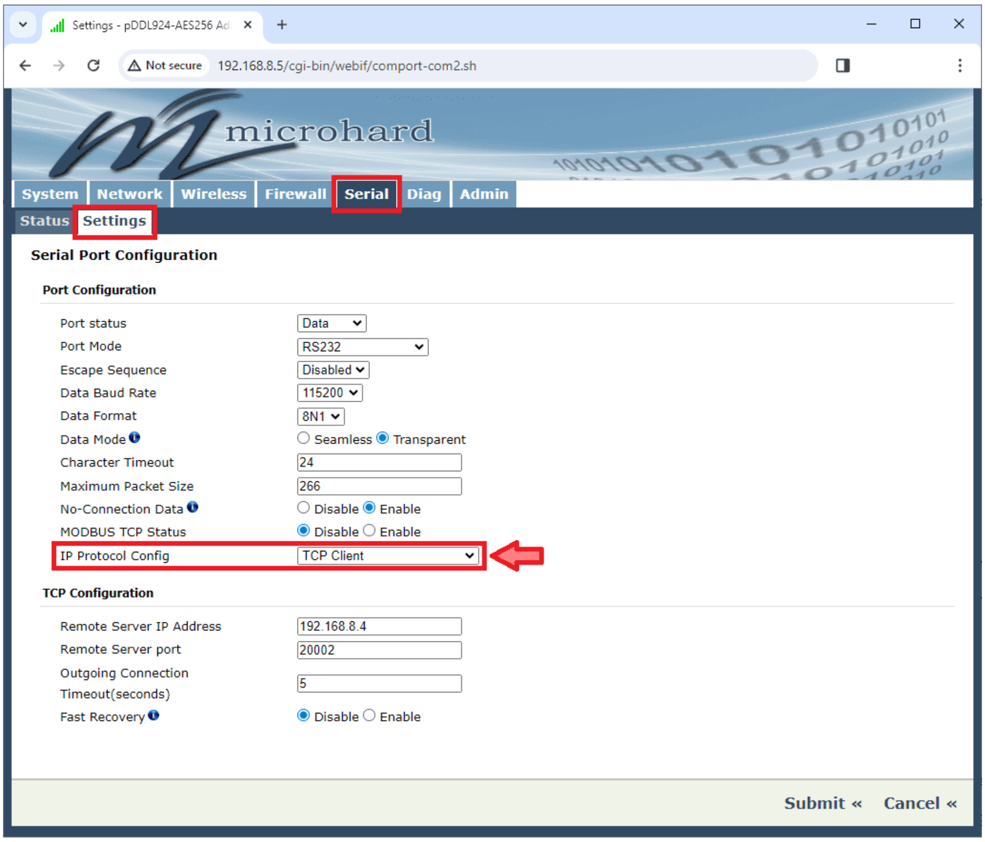
- **Serial Port TX Rate**: When using Ethernet and Serial data,
  - If the volume of **serial data** is high ⇒ Normal Rate (default option)
  - If the volume of **Ethernet data** is high ⇒ High Rate (compressed).

4. **Serial → Settings**: This menu allows configuring the serial device server for the serial communications port. Data from the serial device can be brought into the IP network via TCP, UDP or multicast; it can also come out of the radio network on another radio serial port.

Port Configuration:



Port Configuration - Ground unit



Port Configuration - Air unit

- **Port status**: Select the operating state of the serial port. By default, **Data**.
- **Port Mode**: **RS232** must be selected.
- **Data Baud Rate**: The serial baud rate is the rate at which the modem should communicate with the connected local asynchronous device.

> **Important**
>
> It must match the one configured in the **1x PDI Builder** software → **115200**.

- **Data Format**: This setting determines the format of the data on the serial port. It is the set of **Data Bits**, **Parity** and **Stop Bits** parameters.

> **Important**
>
> It must match those configured in the **1x PDI Builder** software → **8N1** (8 data bits, No parity, and 1 Stop bit).

- **Data Mode**: This setting defines the framing of the serial output data.
  - In Transparent mode (default), the received data will exit the radio quickly.
  - In Seamless mode, the serial port server will add a gap between data frames to comply with a specified protocol.

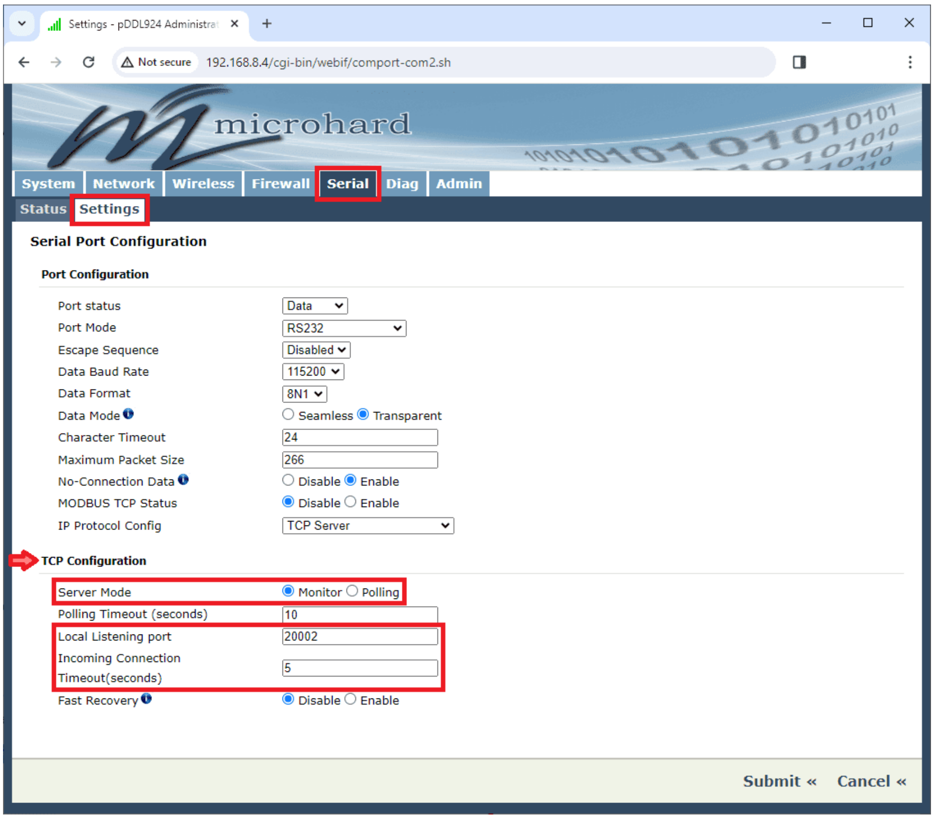
  In this case, **Transparent** mode is selected.
- **Character Timeout**: Defaults to 24 characters.
- **Maximum Packet Size**: Defines the size of the buffer that the serial server will use to receive data from the serial port.

  When the server detects that the Character Timeout criterion has been met, or the buffer is full, it packs the received frame and transmits it.

  It must be **266**, as this is the maximum packet size supported by Veronte Communication Protocol (VCP).
- **IP Protocol Config**: This setting determines which protocol the serial server will use to transmit serial port data over the radio network.
  - For the radio linked to the **ground** unit ⇒ **TCP Server**.
  - For the radio linked to the **air** unit ⇒ **TCP Client**.

TCP Configuration: The protocol selected in the IP Protocol Config field will determine which configuration options appear in this configuration menu.

- In the **TCP Server** mode, the radio series will not initiate a session, but will wait for a Client to request a session. The unit will 'listen' on a specific TCP port.
  - If a **session** is **established**, data will flow from the Client to the Server, and, if present, from the Server to the Client.
  - If a **session** is **not established**, both Client serial data and Server serial data, if present, will be discarded.



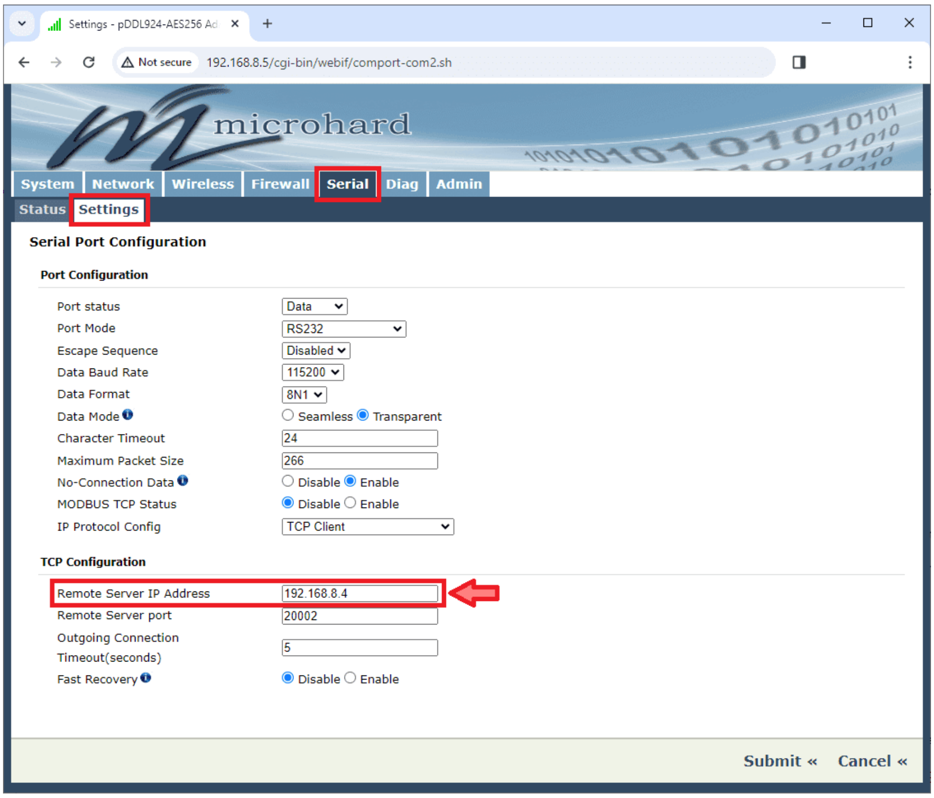
TCP Configuration - Ground unit

- **Server Mode**: **Monitor** is recommended.
- **Local Listening port**: The TCP port on which the server listens. Allows a TCP Client to create a TCP connection to transport data from the serial port. **20002** should be set.
- **Incoming Connection Timeout (seconds)**: Establishes when the TCP Server will terminate the TCP connection if the connection is in an idle state. It is recommended **5 seconds**.

◦ When **TCP Client** mode is selected and data is received on its serial port, the radio takes the initiative to find and connect to a remote TCP server.

The TCP session is terminated by this same unit when the data exchange session is completed and the connection timeout has expired.

If a TCP connection cannot be established, data is discarded from the serial port.



TCP Configuration - Air unit

- **Remote Server IP Address**: Enter an IP address of a TCP server which is ready to accept serial port data over a TCP connection. Users must set the **IP adddress** of the radio defined as **Master**, in this case, the IP address of the radio linked to the ground unit ⇒ **192.168.8.4**.

Connection status radio

- **Network → Status**: Provides an overview of the currently configured network interfaces, including the Connection Type (Static/DHCP), IP Address, Net Mask, Default Gateway, DNS, and IPv4 Routing Table.
- **Wireless → Status → General Status**: Shows the MAC address of the current radio, the Operating Mode (Master, Slave etc), the Network ID being used, Channel Bandwidth and Frequency information and the type of security used.
- **Serial → Status**: This window displays a number of status items that help to visualize the operation, statistics and troubleshooting of the RS232 port.

> ⓘ **Note**
>
> That is, the connection status with **Veronte Autopilot 1x**.

- Port Status: Shows whether the RS232 port has been enabled in the configuration.
- Baud Rate: The current baud rate used to interact with the connected device.
- Connect As: This shows the type of IP Protocol Config chosen (TCP, UDP, SMTP, PPP, etc).
- Connect Status: Shows if there is a current connection / if the port is active.
- Receive bytes/packets
- Transmit bytes/packets

In addition, to check the connection status between the **Autopilot 1x** and the radio, users can simply look at the LED indicators on the radio:

- When the **TX LED** turns **red**, that indicates that the modem is transmitting data over the radio.
- When the **RX LED** turns **green**, it indicates that the moden is synchronized and has received valid packets.

Paired radios

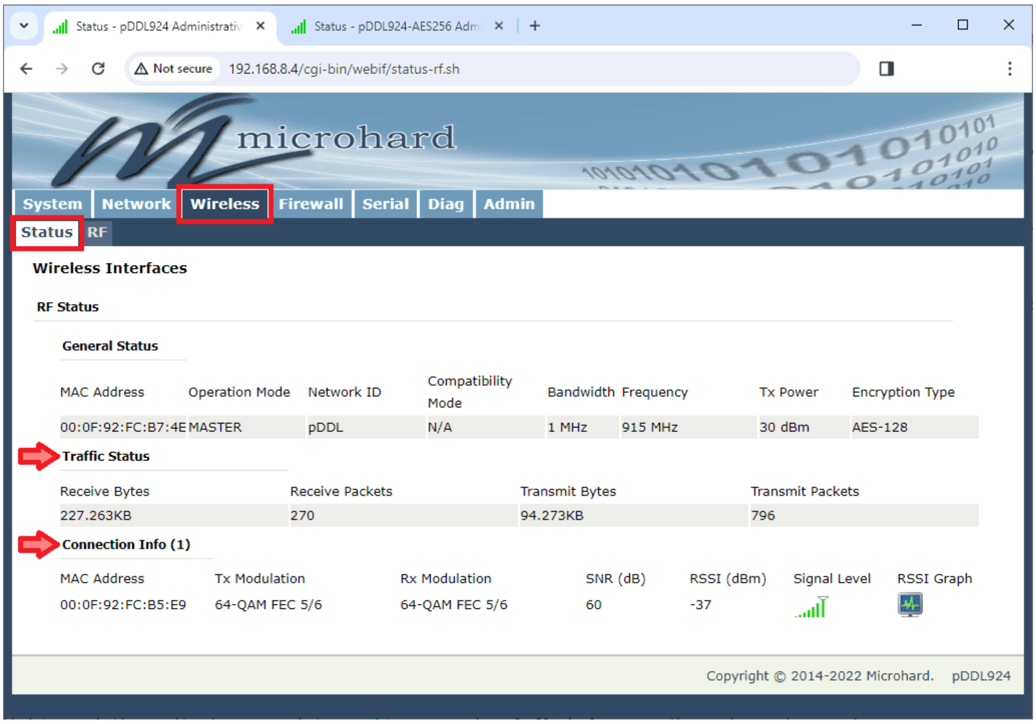
Once both radios are configured with these settings, they should be paired.

Therefore, by connecting both to the power supply, users can check that they are paired by simply observing the **RSSI LEDs**, they should be in a **solid green**. The more LEDs ilumitated ⇒ the stronger the link.

Moreover, users can access the radio's WebUI to check it in the following menu:

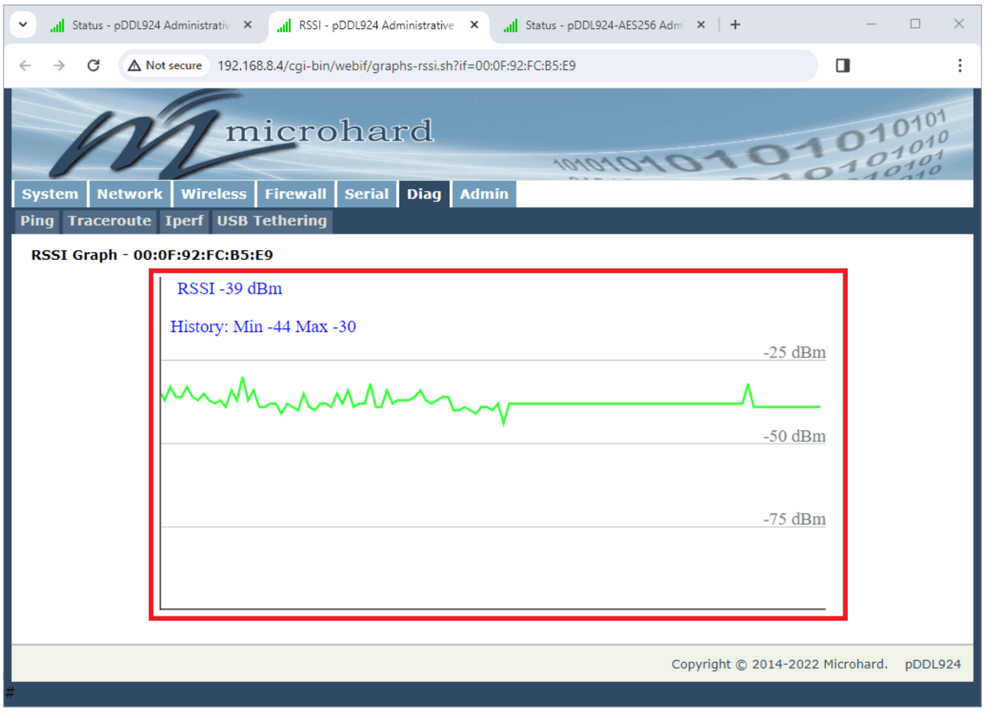
> ⓘ **Note**
>
> If they are correctly linked, by connecting only one of the radios to the PC, users will be able to access the WebUI of both radios.

Wireless → **Status**: The Status window provides an overview of all wireless or radio settings and connections.



Wireless Status

- **Traffic Status** shows statistics on transmitted and received data.
- **Connection Info** displays information about all wireless connections. The MAC address, TX & RX Modulation, Signal to Noise ratio (SNR), Signal Strength (RSSI), and a graphical representation of the signal level or quality, as well as a **RSSI Graph Link**. By clicking on , a new window with the **RSSI Graph Link** will be displayed:



Wireless Status - RSSI Graph Link

Microhard radio configuration in 1x PDI Builder

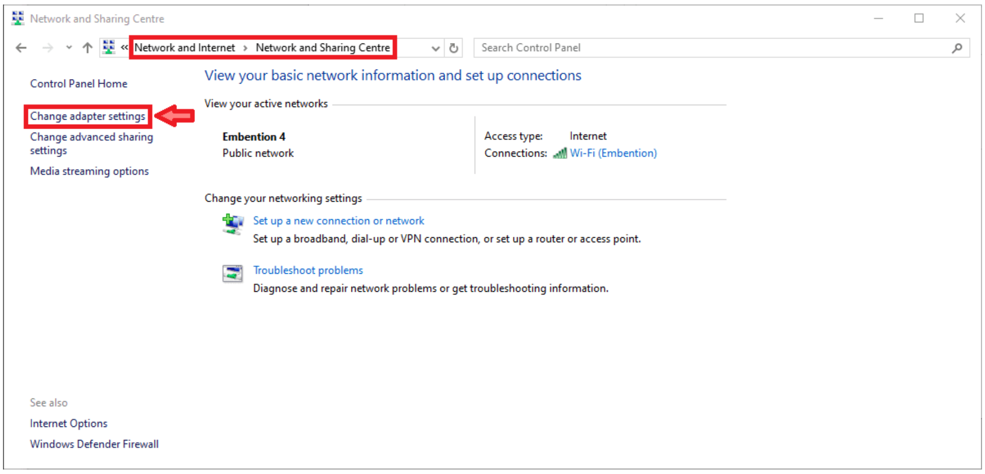
The necessary configuration of **Microhard** radio in **1x PDI Builder** is described in the External radios - Integration examples section of the **1x PDI Builder** manual.

Microhard radio troubleshooting

If the user has problems accessing the radio's WebUI, try setting the computer to the static IP address on the same subnet as the radio.

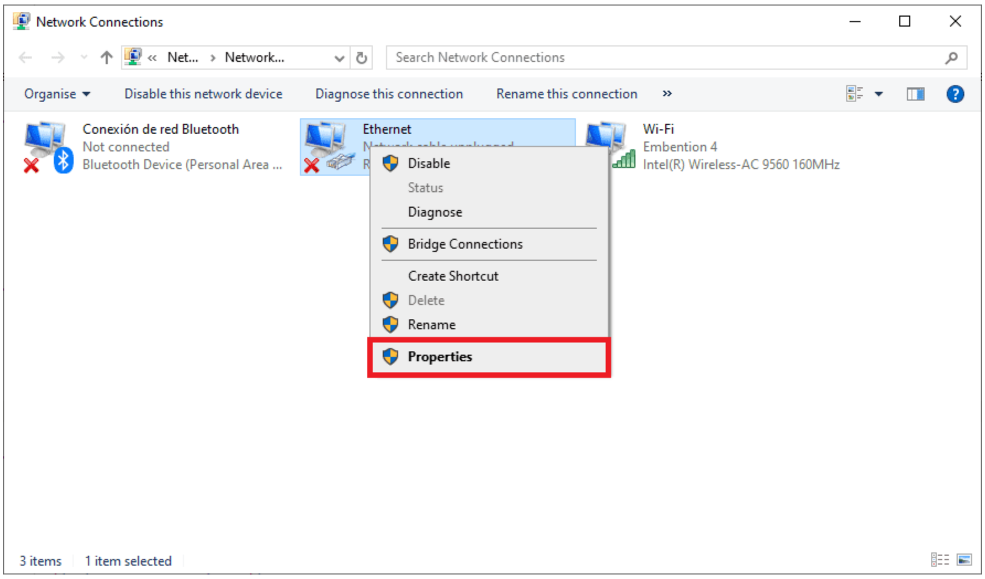
The following steps clarify how to set the IP address in the **Control Panel**:

1. Open **Network and Sharing Centre** menu and click **Change adapter settings**.



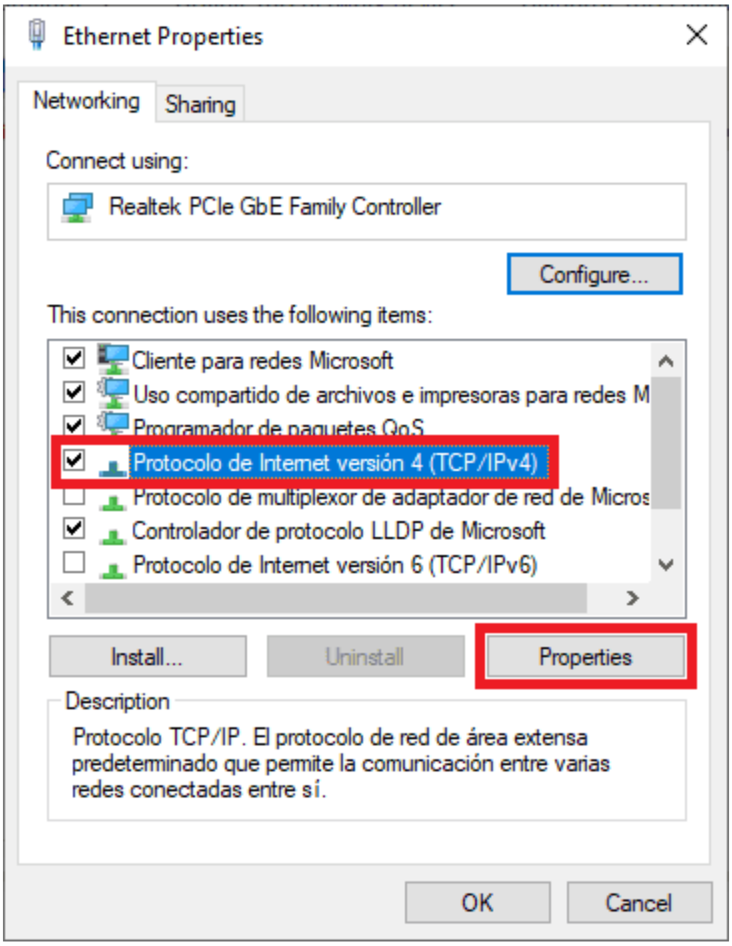
Ethernet connection 1

2. Select **Local Area Connection**, right click, and select **Properties**.



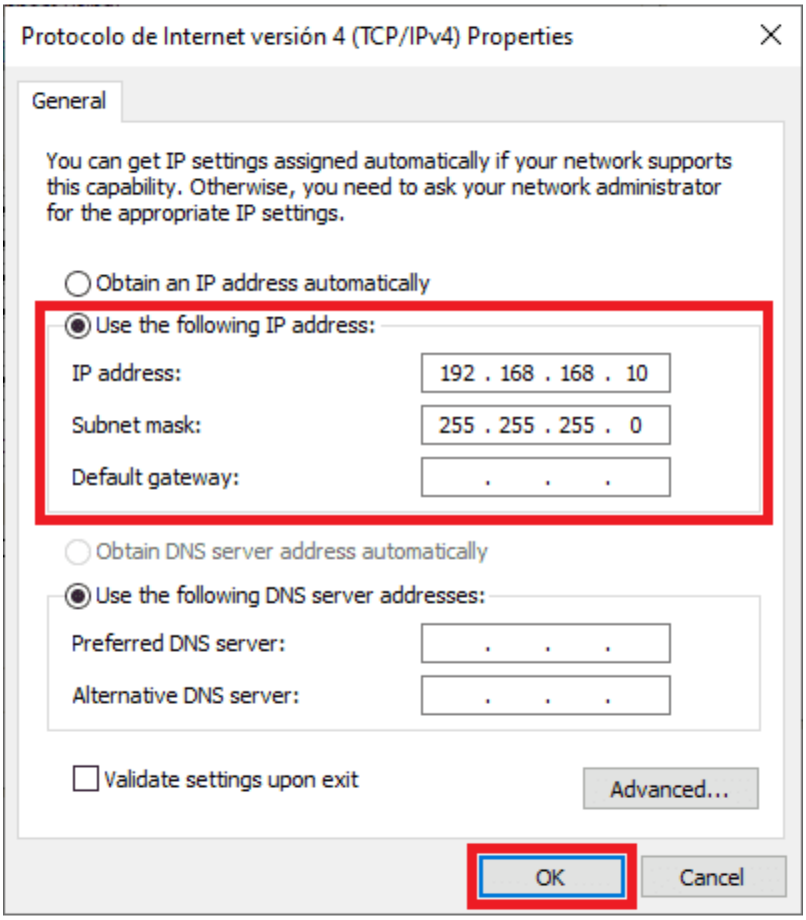
Ethernet connection 2

3. Select **IPv4** and click **Properties**.



Ethernet connection 3

4. Set **IP address** to 192.168.168.YY (e.g. if the IP of the radio is 192.168.168.1, set the IP 192.168.168.10) and **Subnet mask** to 255.255.255.0. Click **OK**.

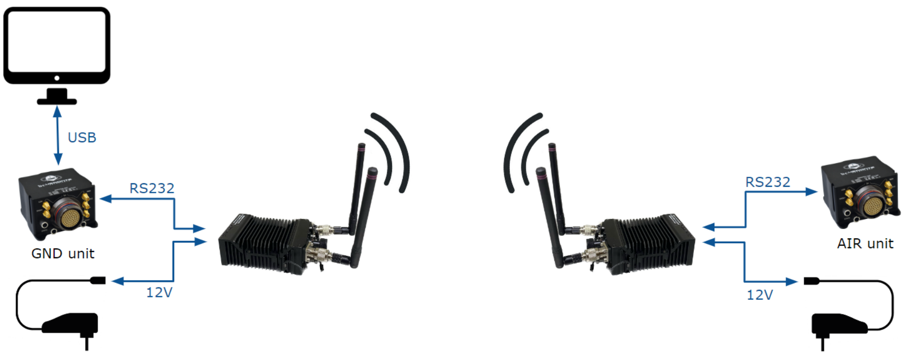


Ethernet connection 4

Silvus radio (StreamCaster 4200E model)

System Layout

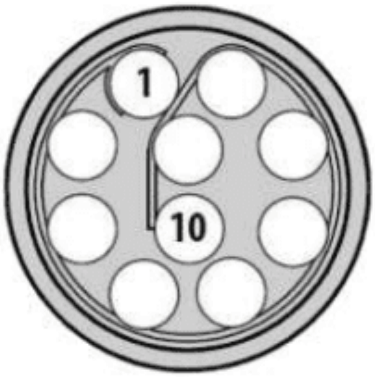
The following image shows the standard connection between **Silvus** radios and **Autopilot 1x** for operation:



Silvus and 1x connection

Hardware Installation

A wiring configuration of the PRI cable connected to the PRI port of the radio is required, in order to connect to the power supply, ethernet and RS-232.



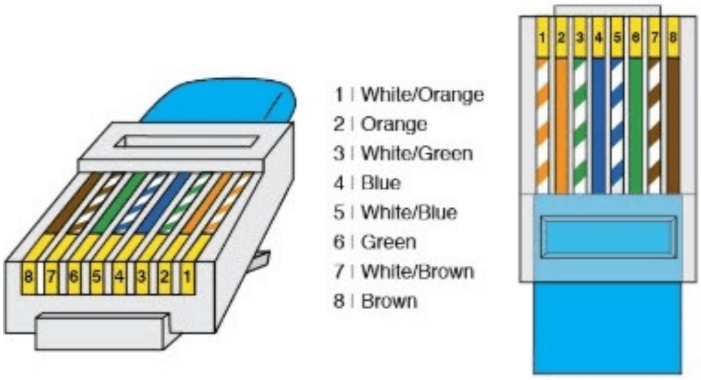
PRI port connector (mounted in radio)

- **Power supply**

Female DC Power Jack connector

PRI port Connector - Silvus Radio		Power Connector
PIN	**Signal**	**Signal**
2	GND IN	Power -
3	VCC IN	Power +

- **Ethernet**

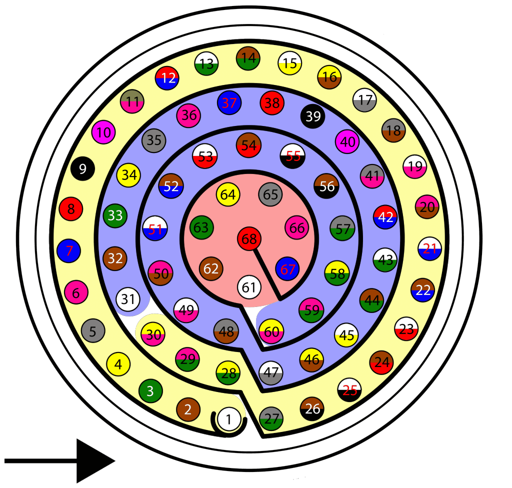


RJ45 pinout T-568B

PRI port Connector - Silvus Radio		RJ45 Connector (T-568B)		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
4	ETH0_MX2N (RX-)	6	RX-	Green
5	ETH0_MX2P (RX+)	3	RX+	Green-White
6	ETH0_MX1P (TX+)	1	TX+	Orange-White
10	ETH0_MX1N (TX-)	2	TX-	Orange

- **RS-232**

  The RS-232 from the PRI cable should be connected to the RS-232 of **Autopilot 1x** harness.



Autopilot 1x harness pinout

PRI port Connector - Silvus Radio		Autopilot 1x Harness		
PIN	**Signal**	**PIN**	**Signal**	**Color Code**
7	RS232_RXD	19	RS 232 TX	White-Pink
8	RS232_TXD	20	RS 232 RX	Pink-Brown
9	GND	21	GND	White-Blue

Silvus radio configuration

This section shows a basic configuration of the **Silvus** radio.

First steps

1. Connect antennas (or attenuators) with male TNC ends to 2 RF ports.
2. Connect power supply to power port on PRI cable.
3. Connect non-forked female side of PRI cable to radio's PRI port.

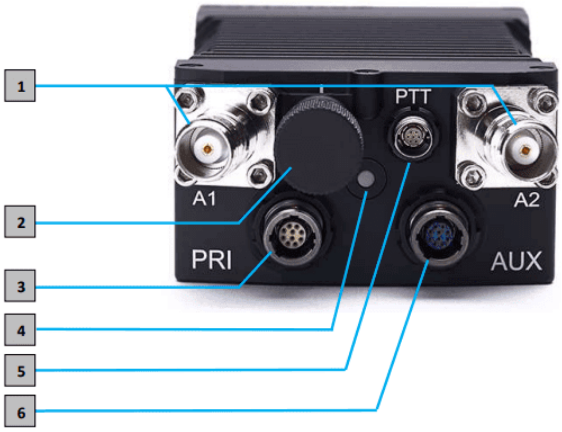


Figure 3 StreamCaster 4200E Ruggedized Enclosure

1 RF Channels 1-2 Connectors [TNC Female]

2 Power Switch [15-Position Rotating]

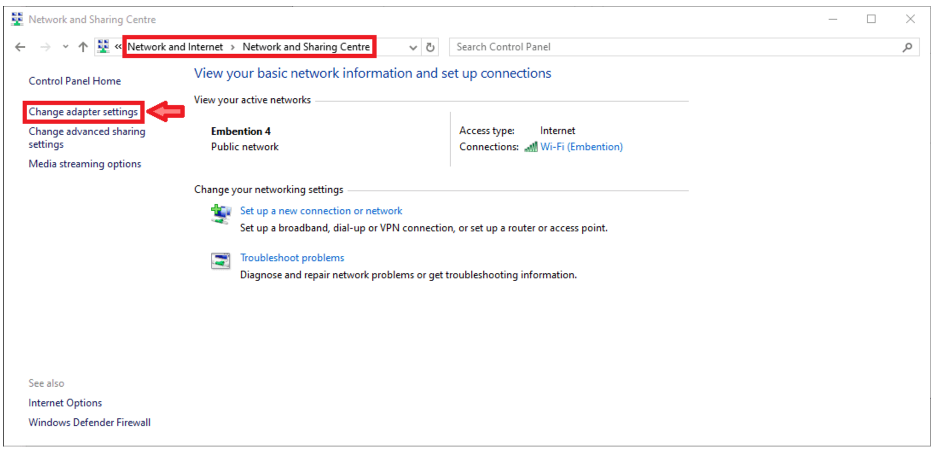
3 Power (EB Version Only, 9-20V), Ethernet, and Serial Port Connector [ODU GK0YAR-P10UC00-000L]

4 Bi-Color Status LED

- Red – Radio is in the process of booting up
- Flashing Green – Radio is fully booted but not wirelessly connected to any other radio
- Green – Radio is wirelessly connected to at least one other radio

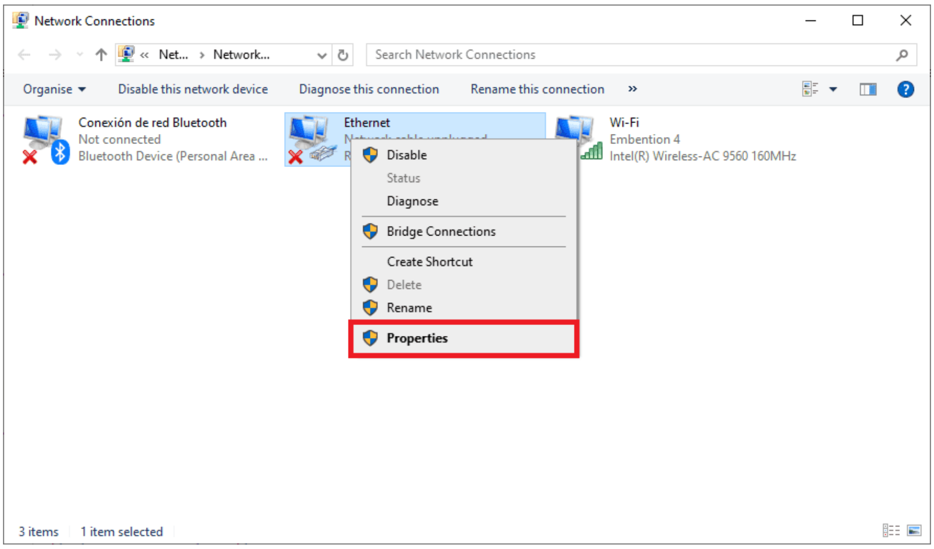
Silvus connectors

4. When looking at the rotary multi position switch from the top, pull the knob towards you while rotating the knob towards the 1 position.
   This turns radio on. LED indicator will turn to fix red.
5. In order to access the StreamScape graphical user interface (GUI), connect Ethernet (RJ45) connector of PRI cable to Ethernet port of laptop/computer.
6. Make sure computer is set to static IP address on same subnet as radio. The following substeps clarify how to set the IP address:
   1. Open **Network and Sharing Centre** menu and click **Change adapter settings**.



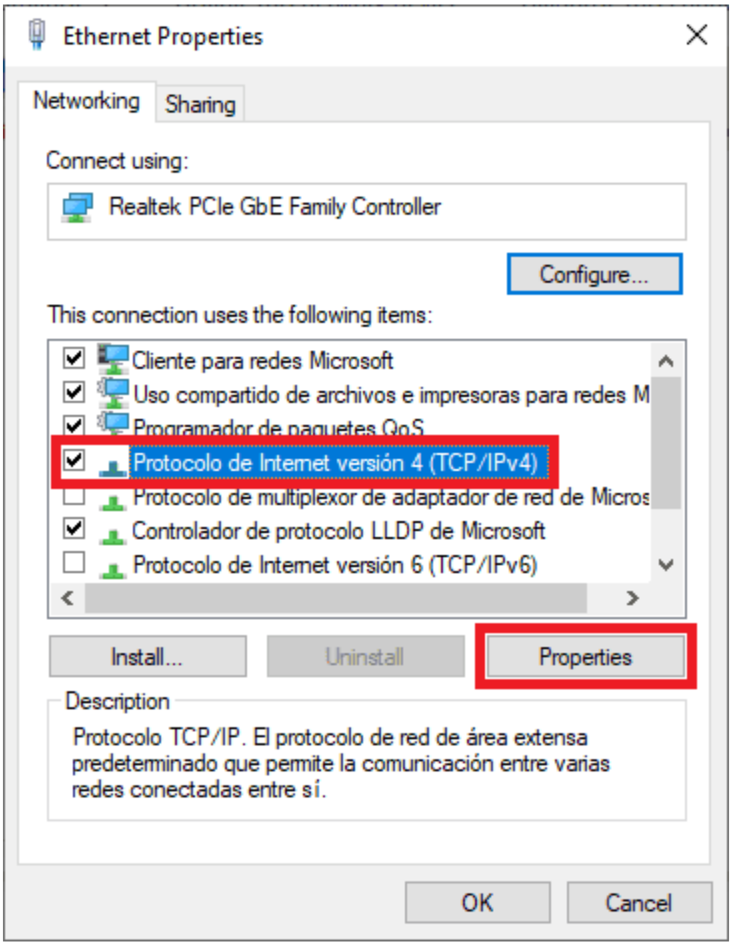
Ethernet connection 1

2. Select **Local Area Connection**, right click, and select **Properties**.



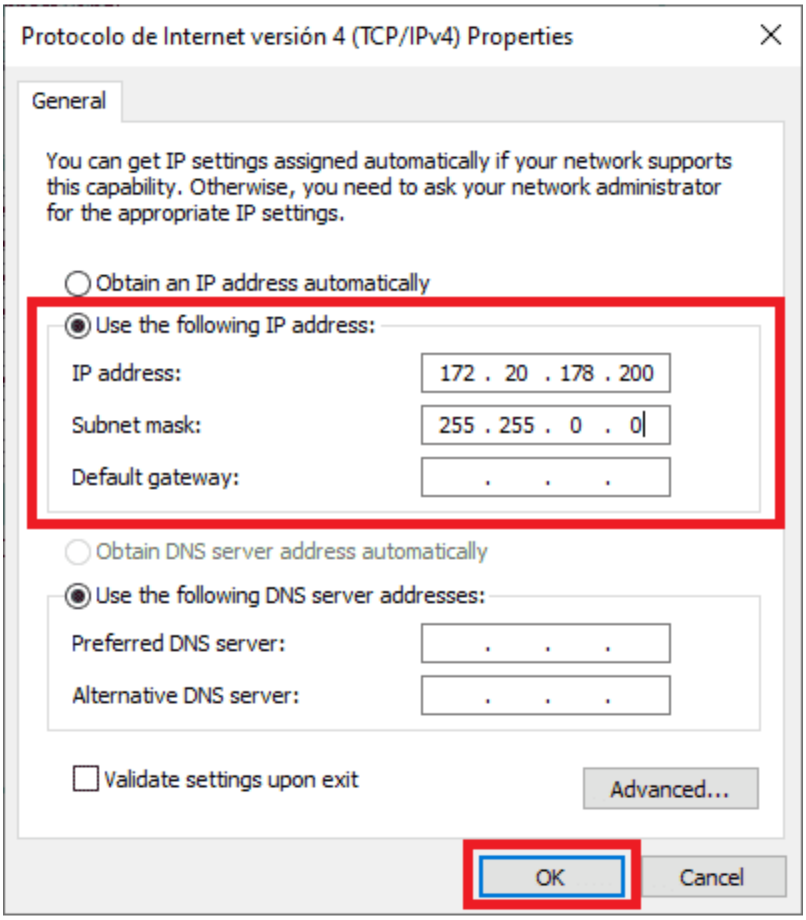
Ethernet connection 2

3. Select **IPv4** and click **Properties**.



Ethernet connection 3

4. Set **IP address** to 172.20.XX.YY (e.g. if the IP of the radio is 172.20.178.203, set the IP 172.20.178.200) and **Subnet mask** to 255.255.0.0. Click **OK**.



Ethernet connection 4

7. Wait for LED indicator to turn to blinking green.
8. Access **StreamScape** GUI in web browser. To access, enter IP address of radio into web browser and press enter.

> ⓘ **Note**
>
> Latest version of Firefox or Google Chrome are preferred. Internet Explorer or others are not recommended.

Silvus initial menu

9. User manual can be accessed by clicking the book icon in the GUI (Next to **Basic Configuration** in the previous screenshot).

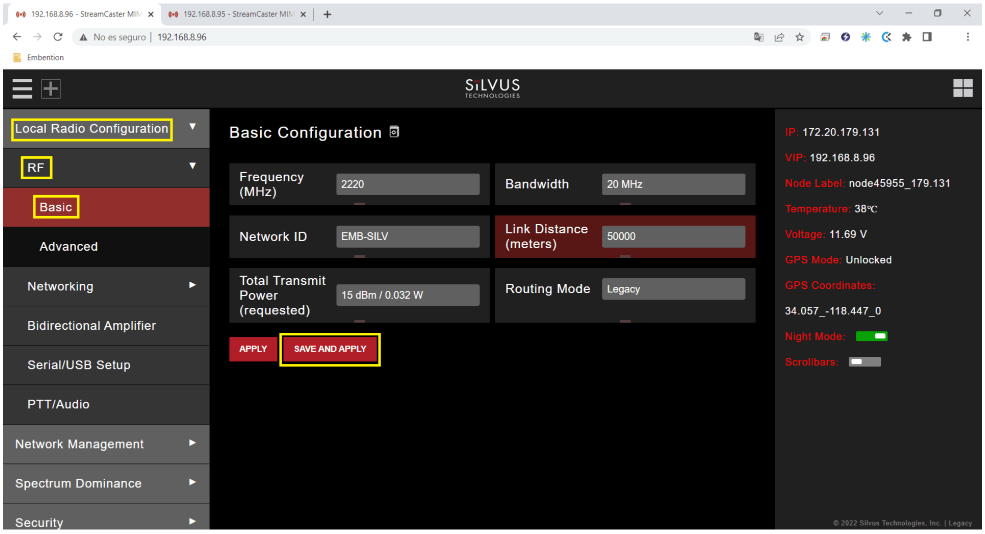
Basic radio configuration

Once the website has been accessed, follow the steps below which show the parameters that need to be modified for correct operation and pairing of the radios.

> ⓘ **Note**
>
> This is an example of the radio configuration linked to a 1x air unit.

> ⓘ **Note**
>
> After making changes to each window, it is important to click on "Save and apply".

1. Basic Configuration.



Basic configuration panel

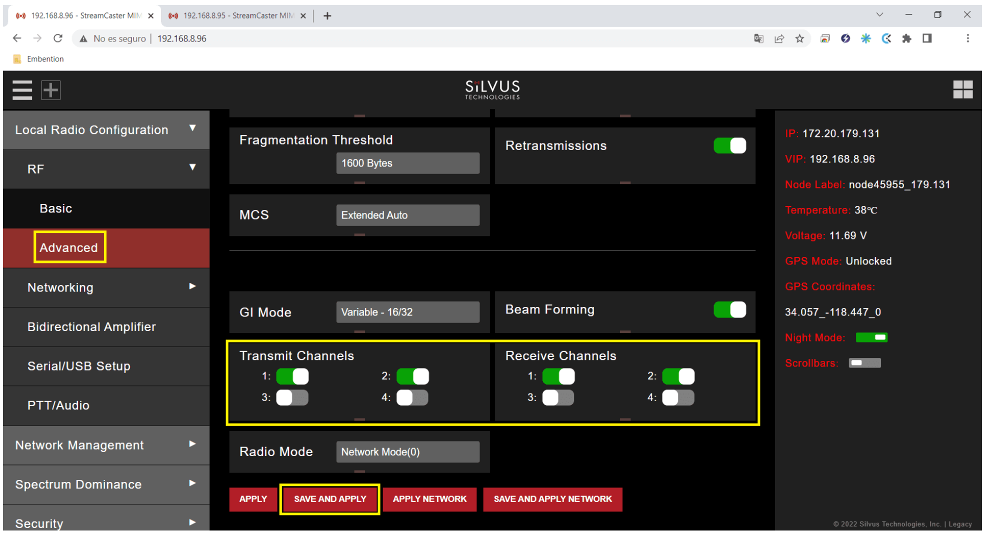
- **Frequency (MHZ)**: This defines the frequency of the signal. There is a drop-down menu for frequency selection. We recommend 2220 MHz.

  > ⚠ **Warning**
  >
  > Be careful when choosing the frequency. The user may see interference with the Wifi frequency band, consult the radio spectrum.

- **Bandwith**: This defines the RF bandwidth of the signal. Default value.

- **Network ID**: Network ID allows for clusters of radios to operate in the same channel, but remain independent.
  A radio with a given Network ID will only communicate with other radios with the same Network ID.
- **Link Distance (meters)**: Set to an approximate maximum distance between any two nodes in meters. It is important to set the link distance to allow enough time for packets to propagate over the air. **It is recommended to set the link distance 10-15% greater than the actual maximum distance**.
- **Total Transmit Power (requested)**: This defines the total power of the signal (power is divided equally between the radio antenna ports). Set the appropriate power for each application. The power that has been set is small, as it is sufficient for our tests.
- **Routing Mode**: As Large Network mode requires a license and is not available outside USA, we set Legacy mode.

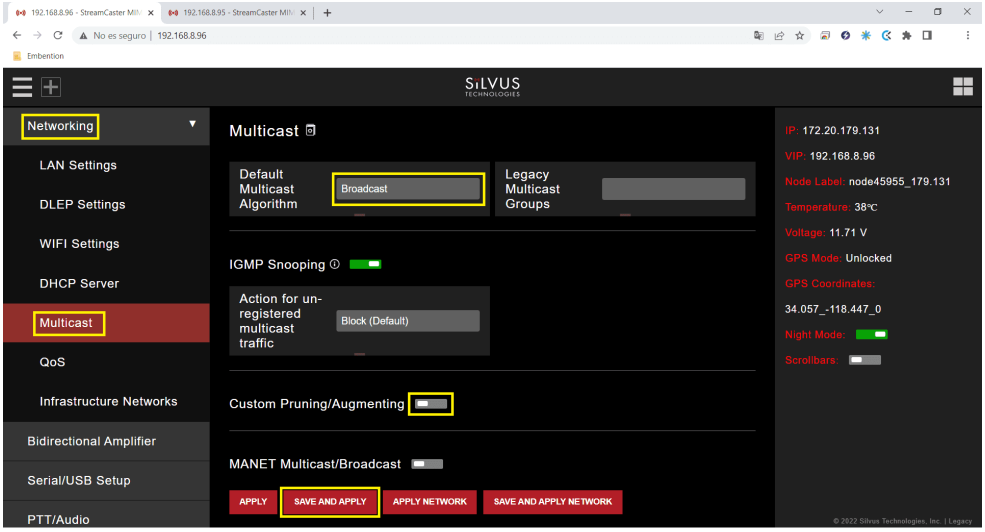
2. Advanced configuration.



Advanced configuration panel

- **Transmit/Receive Channels**: Allows user to enable or disable each channel on the radio for TX/RX (each RF port is a channel).
  We have enabled both channels.

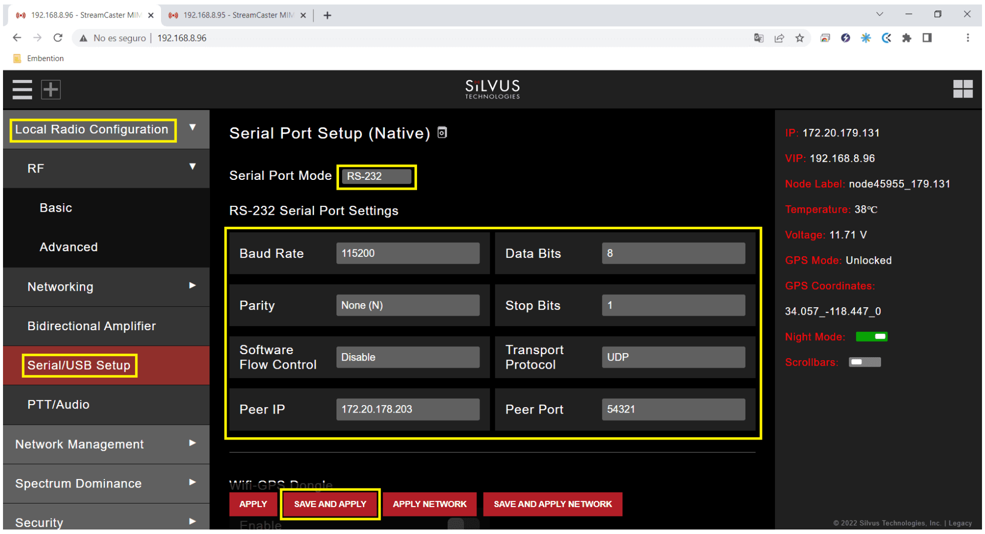
3. Networking. Multicast.



Multicast panel

- **Default Multicast Algorithm**: Broadcast.
- **Custom Pruning/Augmenting**: Disable.

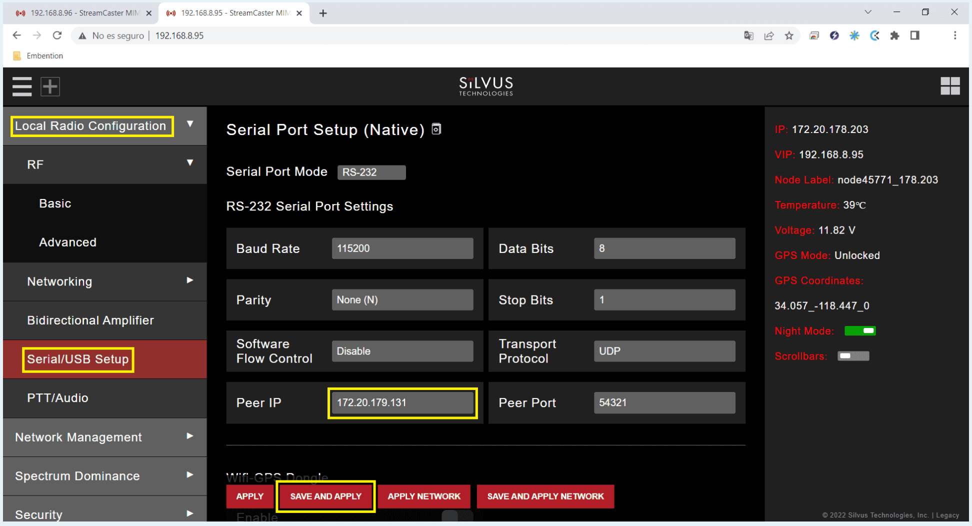
4. Serial/USB Setup



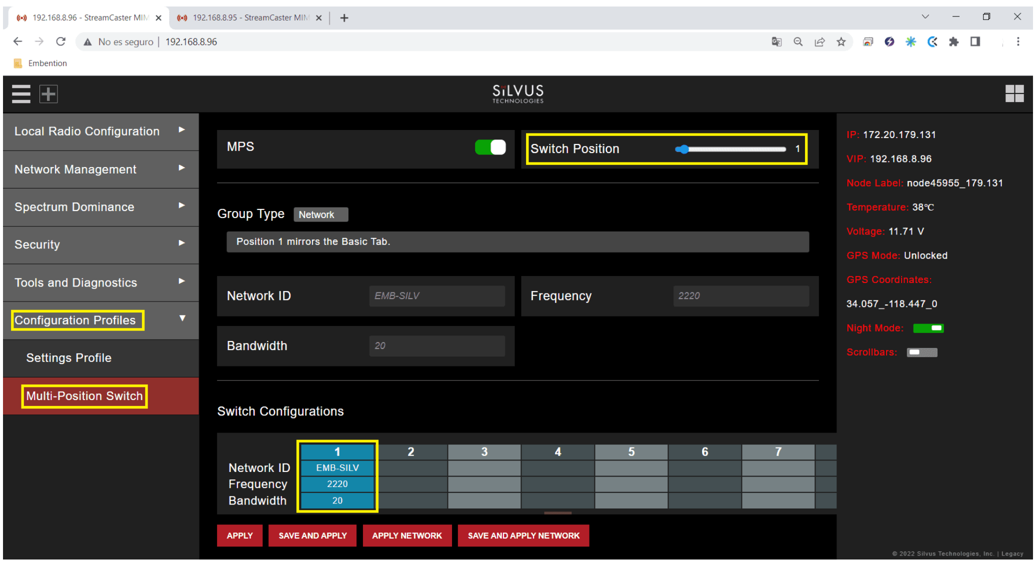
RS-232 settings

- **Serial Port Setup**: RS-232.
- RS-232 Serial Port Settings
  - The value of the **Baudrate**, **Data Bits**, **Parity** and **Stop Bits** parameters must be the same as those configured in **1x PDI Builder**.
  - **Software Flow Control**: Disable.
  - **Transport Protocol**: We recommend **UDP**. If no data loss can be tolerated, change this setting to TCP on the radio corresponding to the 1x **air** unit.

- **Peer IP**: This should be the IP address of the radio on the other end of the RS-232. In this example, we must set the IP address of the radio linked to the ground unit.

> ⓘ **Note**
>
> Both radios (the one connected to the GND unit and the one connected to the AIR unit) have the same configuration except for the **Peer IP**.
>
> 
>
>
> **Peer IP in radio linked to the GND unit**

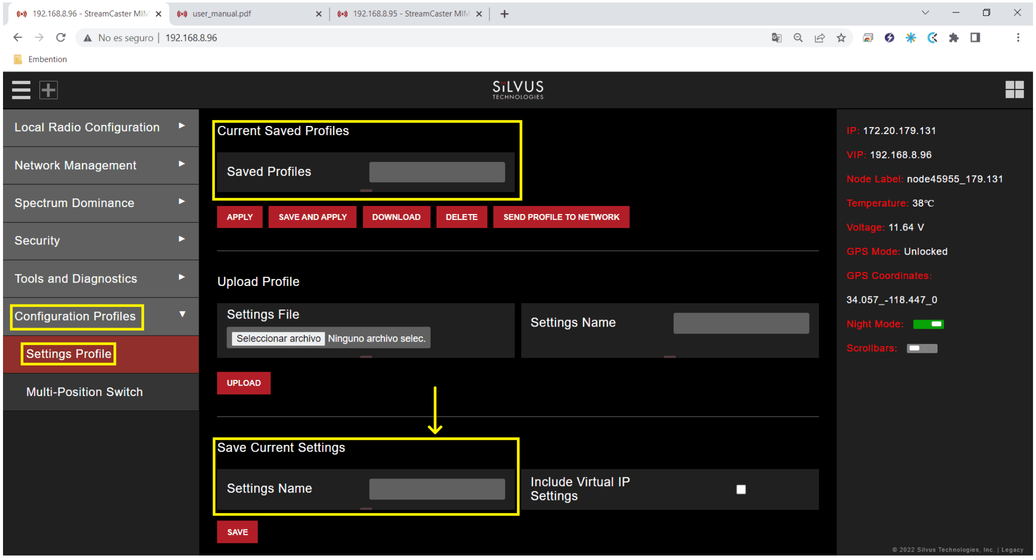
In addition to these settings, different configurations can be stored in the same radio, on the **Multi-Position Switch** panel. The user can select the one that will work, with the radio's switch position.



Multi-Position Switch panel

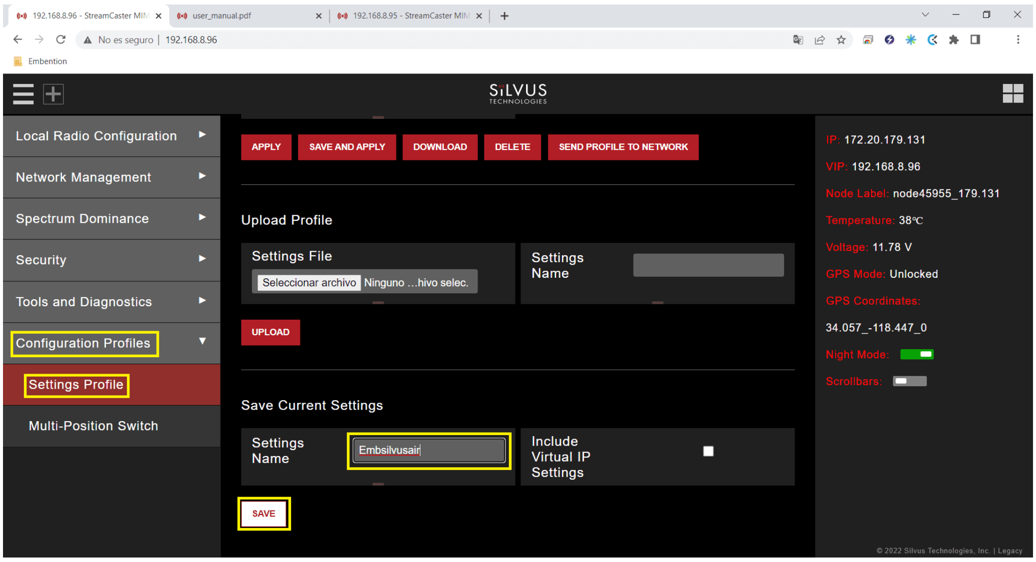
In this example only one configuration has been created.

With the above settings the configuration is finished. Furthermore, this configuration can be saved and downloaded in the **Settings Profile** window of the Configuration Profiles section.

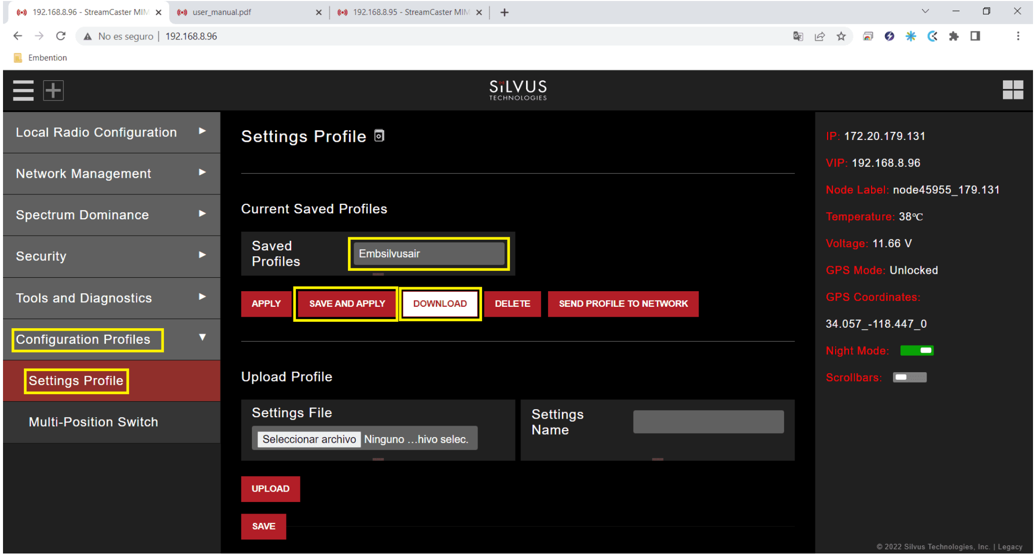


Settings Profile panel

Before downloading the configuration, it is necessary to save it.



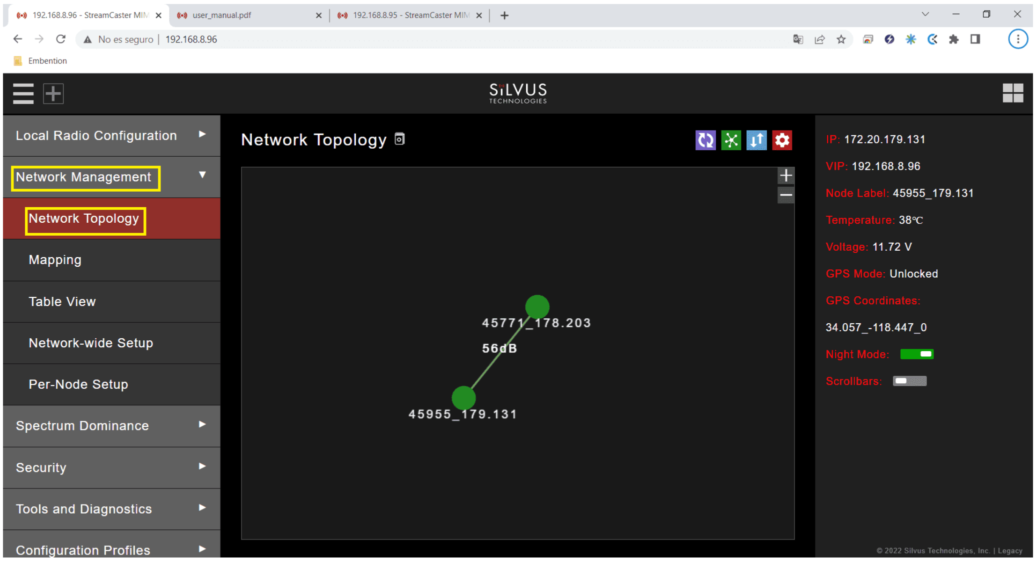
Save settings



Download settings

After configuring both radios with these settings they should be paired. Therefore, if we connect them to the power supply, when we switch them on, the LED will turn from fix red to fix green, this indicates that it is connected to at least one radio. Also, if we connect only one of them to the computer, we can access the **StreamScape** GUI of both.

And, in the **Network Topology** window of the Network Management section, we can see the link between them.



Connection between radios

Silvus radio configuration in 1x PDI Builder

The necessary configuration of **Silvus** radio in **1x PDI Builder** is described in the External radios - Integration examples section of the **1x PDI Builder** manual.

Veronte SDL

SDL radio module has its own user manual, click here to read it.

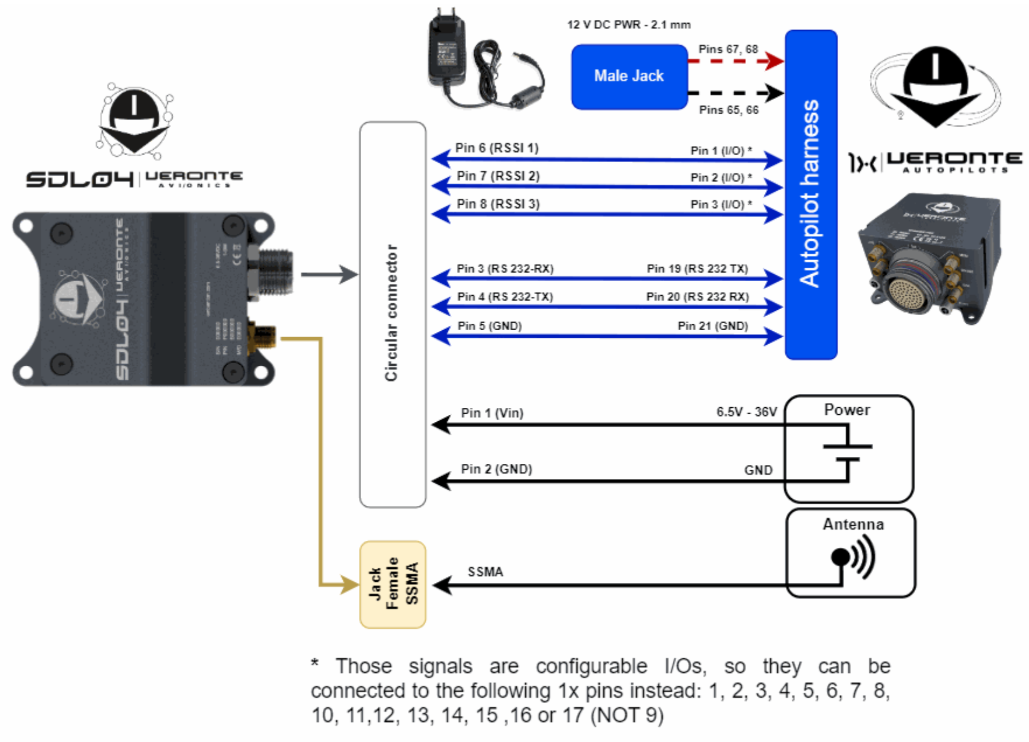
This section shows how to connect and configure a **SDL** (any variant) with an **Autopilot 1x air unit** and an **Autopilot 1x ground unit**. The following diagram summarizes the flight communication system between a 1x ground unit and a 1x air unit installed on an aircraft.

Operation diagram

First of all, **Autopilots 1x** and **SDL** require software configuration:

- To configure **Autopilots 1x**, use **1x PDI Builder** by reading External radios - Integration examples section of the **1x PDI Builder** user manual.
- To configure **SDL**, read the following sections of the **SDL** user manual:
  - To know the basics and start configuring it: Software Installation.
  - To configure according to the application requirements: Veronte Autopilot 1x - Integration examples.

Once all devices have been configured, the electrical connections can be established with the following figure, which shows the required connections according to 1x pinout, and SDL pinout.



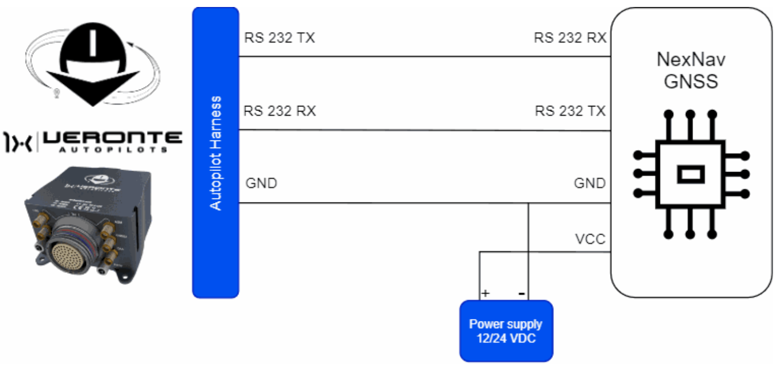
Connections diagram

GNSS Receivers

NexNav GNSS

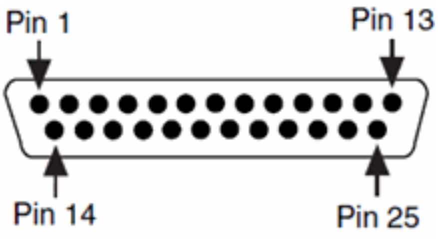
NexNav GNSS

NexNav GNSS sensor integrates with **Autopilot 1x** via **RS232** connection.

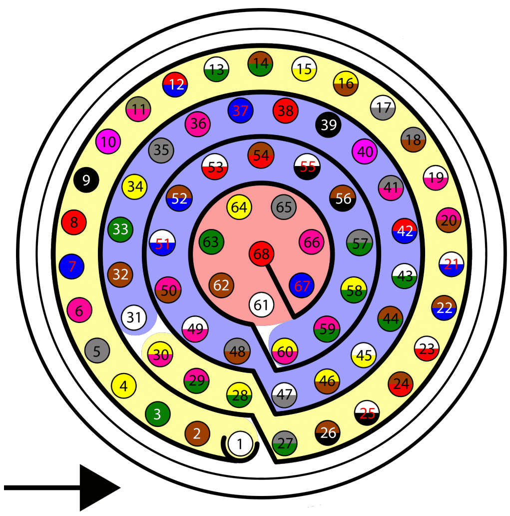


NexNav GNSS - Autopilot 1x wiring diagram

> **Important**
>
> Note that it must be connected to an **external power supply**, **sharing signal ground** with **Autopilot 1x**.



NexNav GNSS connector pinout



Autopilot 1x harness pinout

Autopilot 1x Harness			NexNav GNSS Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
19	RS 232 TX	White-Pink	20	RS-232 RX, Port 1
20	RS 232 RX	Pink-Brown	7	RS-232 TX, Port 1
21	GND		10	

Autopilot 1x Harness			NexNav GNSS Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
		White-Blue		RS-232 Ground

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

The software installation of this device with **Autopilot 1x** is explained in the NexNav GNSS - Integration examples section of the **1x PDI Builder** user manual.

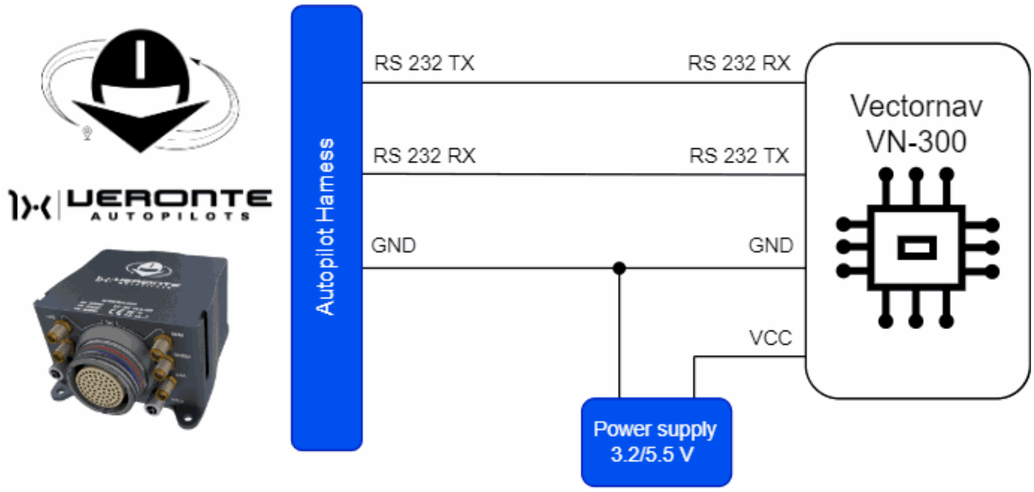
IMUs & Compass

IMUs

Vectornav VN-300

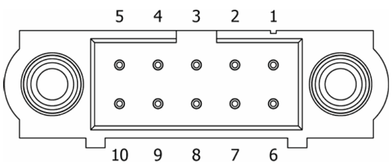
Vectornav VN-300

Vectornav VN-300 is an **external IMU** that can be connected via **RS-232** (serial interface) to **Veronte Autopilot 1x**.

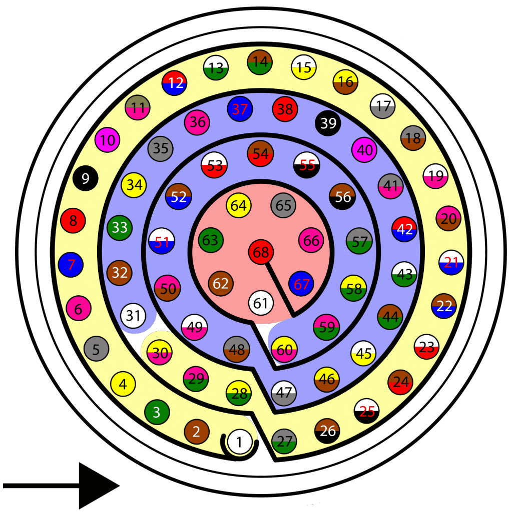


Vectornav VN-300 - Autopilot 1x wiring diagram

> **Important**
>
> Note that it must be connected to an **external power supply**, **sharing signal ground** with **Autopilot 1x**.



Vectornav VN-300 connector pinout



Autopilot 1x harness pinout

Autopilot 1x Harness			VectorNav VN-300 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
19	RS 232 TX	White-Pink	1	RX1
20	RS 232 RX	Pink-Brown	2	TX1
21	GND	White-Blue	5	GND

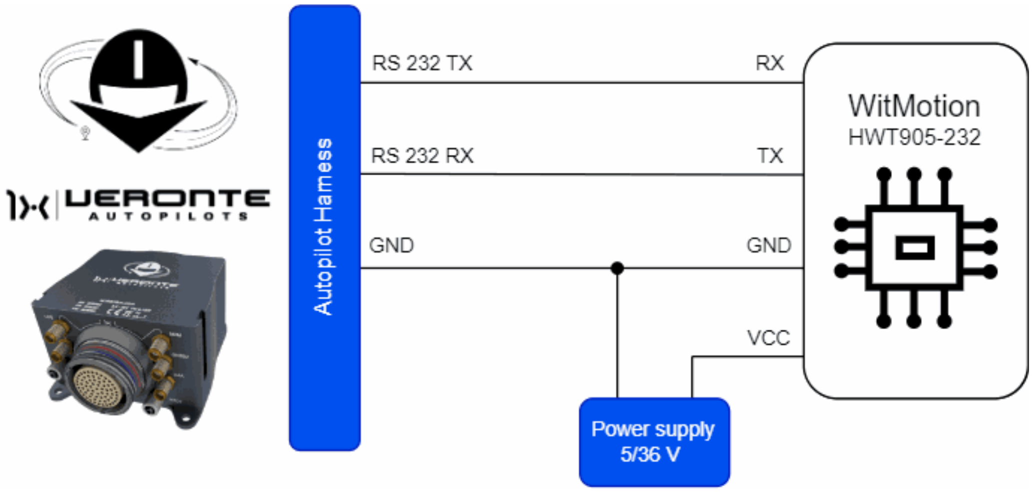
> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the IMU is connected, the user must proceed to its software integration with **Veronte Autopilot 1x** by referring to the Vectornav VN-300 - Integration examples of the **1x PDI Builder** user manual.

WitMotion HWT905-232

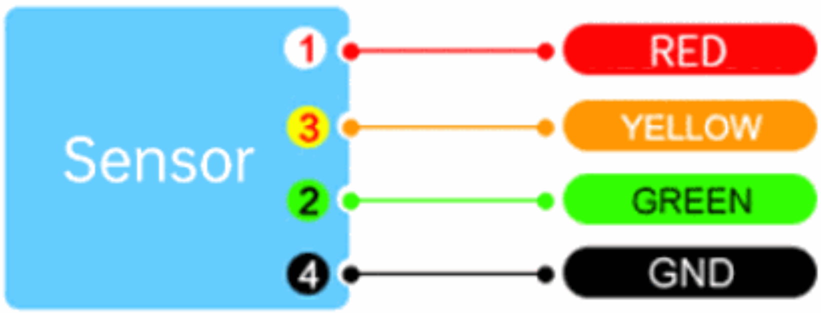
WitMotion HWT905-232

Vectornav VN-300 is an **external IMU** that can be connected via **RS-232** (serial interface) to **Veronte Autopilot 1x**.

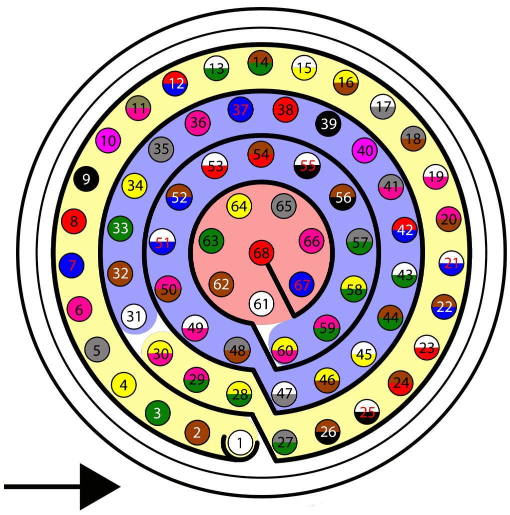


WitMotion HWT905-232 - Autopilot 1x wiring diagram

> **Important**
>
> Note that it must be connected to an **external power supply**, **sharing signal ground** with **Autopilot 1x**.



WitMotion HWT905-232 connector pinout



Autopilot 1x harness pinout

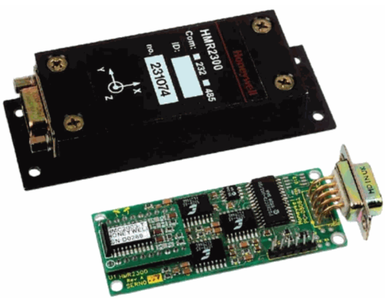
Autopilot 1x Harness			WitMotion HWT905-232 Connector		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
19	RS 232 TX	White-Pink	2	RX	Green
20	RS 232 RX	Pink-Brown	3	TX	Yellow
21	GND	White-Blue	4	GND	Black

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the IMU is connected, the user must proceed to its software integration with **Veronte Autopilot 1x** by referring to the WitMotion HWT905-232 - Integration examples of the **1x PDI Builder** user manual.

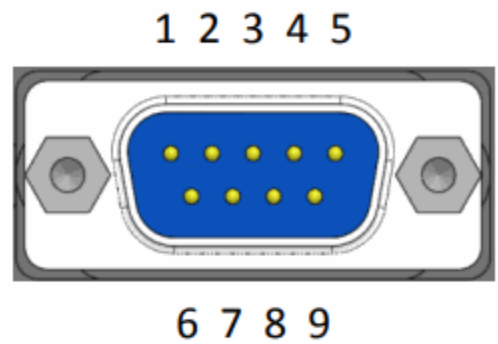
Magnetometers

Magnetometer Honeywell HMR2300

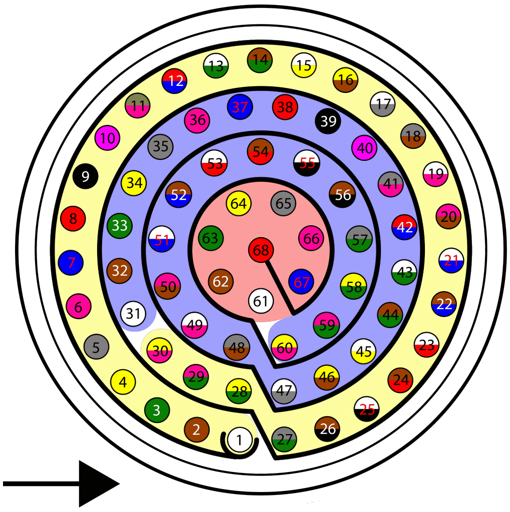


Magnetometer Honeywell HMR2300

Magnetometer Honeywell HMR2300 is an **external magnetometer** that can be connected to **Veronte Autopilot 1x** via **RS232** or **RS485** (serial interfaces).



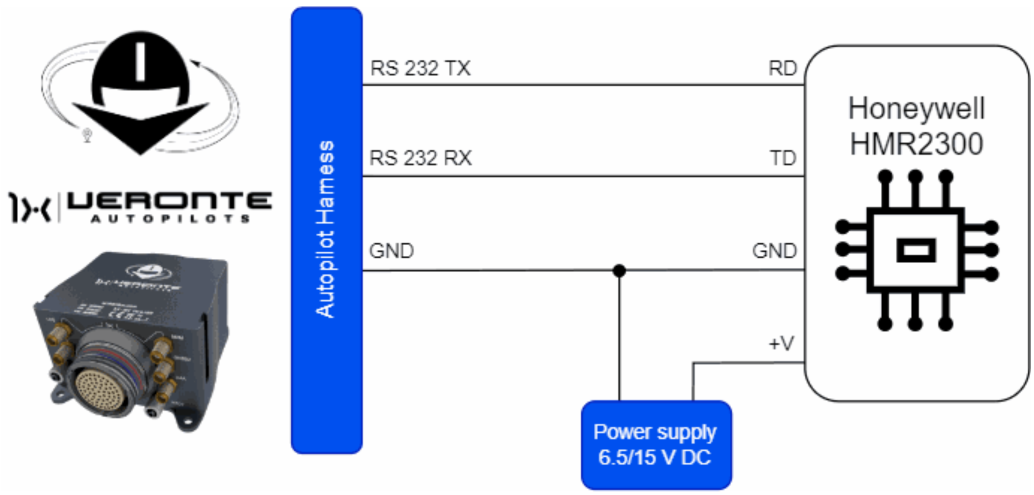
Magnetometer Honeywell HMR2300 connector pinout



Autopilot 1x harness pinout

Connections via **RS232** and **RS485** interfaces are explained separately.

RS232



Magnetometer Honeywell HMR2300 (RS232) - Autopilot 1x wiring diagram

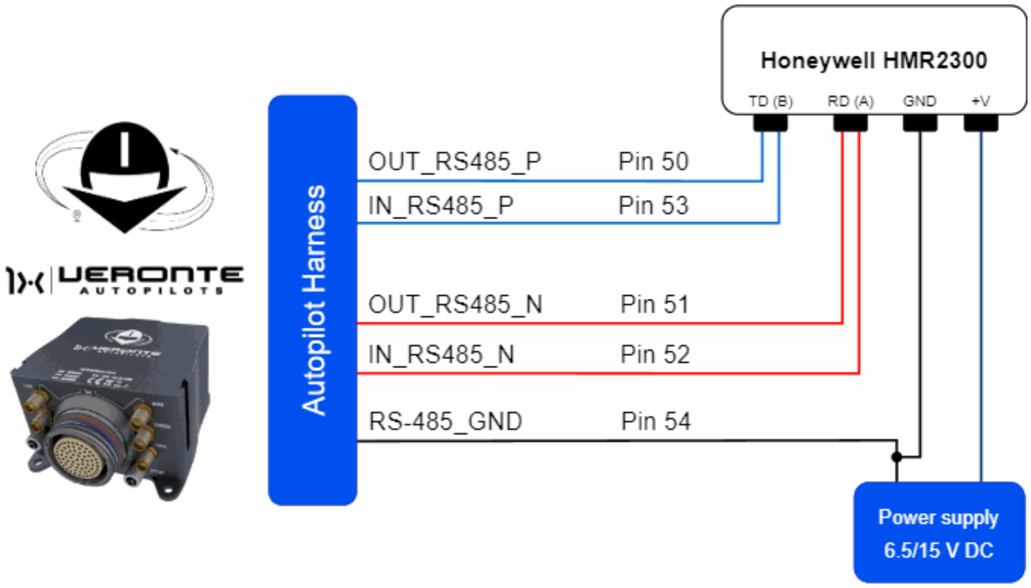
For proper operation via **RS232**, the connection between **Magnetometer Honeywell HMR2300** and **Autopilot 1x** pins should be like this:

Autopilot 1x Harness			Magnetometer Honeywell HMR2300 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
19	RS 232 TX	White-Pink	3	RD
20	RS 232 RX	Pink-Brown	2	TD
21	GND	White-Blue	5	GND

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the **Magnetometer Honeywell HMR2300** is connected, the user must proceed to its software installation with **Veronte Autopilot 1x** by referring to the Magnetometer Honeywell HMR2300 (RS232) - Integration examples of the **1x PDI Builder** user manual.

RS485



Magnetometer Honeywell HMR2300 (RS485) - Autopilot 1x wiring diagram

For proper operation via **RS485**, the connection between **Magnetometer Honeywell HMR2300** and **Autopilot 1x** pins should be like this:

Autopilot 1x Harness			Magnetometer Honeywell HMR2300 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
50	OUT_RS485_P	Pink-Brown	2	TD (B) Transmit Data, RS-485 (B+)
53	IN_RS485_P	White-Red		
51	OUT_RS485_N	White-Blue	3	RD (A) Receive Data,

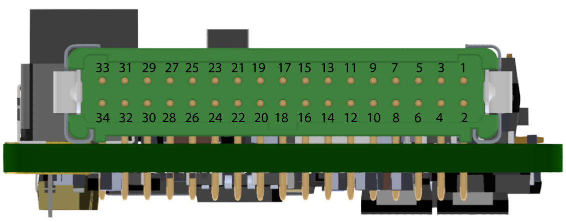
Autopilot 1x Harness			Magnetometer Honeywell HMR2300 Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
52	IN_RS485_N	Brown-Blue		RS-485 (A-)
54	OUT_GND	Brown-Red	5	GND

> ⚠ **Warning**
>
> Note that this pin 54 is not a common GND pin.

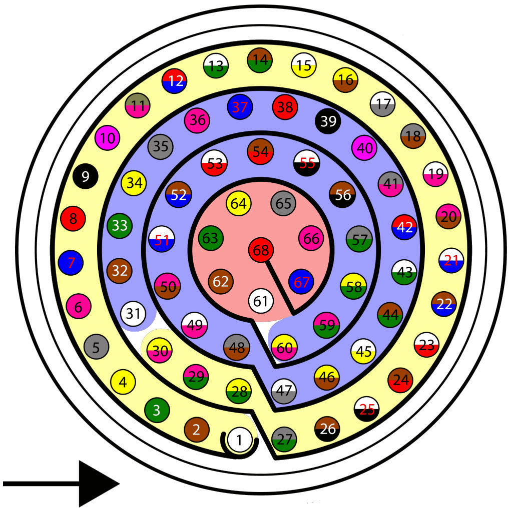
> ⓘ **Note**
>
> If users encounter any problems during wiring, please check the Half-duplex servo does not respond - Troubleshooting section of this manual.

Once the **Magnetometer Honeywell HMR2300** is connected, the user must proceed to its software installation with **Veronte Autopilot 1x** by referring to the Magnetometer Honeywell HMR2300 (RS485) - Integration examples of the **1x PDI Builder** user manual.

MEX as Magnetometer Honeywell HMR2300



MEX connector pinout



Autopilot 1x harness pinout

MEX can be used as an **external magnetometer Honeywell HMR2300** connected to **Veronte Autopilot 1x** via **serial** (RS232/RS485) or **CAN** interfaces.

Connections via **CAN**, **RS232** and **RS485** interfaces are explained separately.

CAN

For proper operation via **CAN**, the connection between **MEX** and **Autopilot 1x** pins should be like this:

Autopilot 1x Harness			MEX Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
25	CANA_P	White-Black	22	CAN A (P)
26	CANA_N	Brown-Black	23	CAN A (N)
28	CANB_P	Yellow-Green	20	CAN B (P)
29	CANB_N	Pink-Green	21	CAN B (N)

Autopilot 1x Harness			MEX Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
30	GND	Yellow-Pink	24	CAN GND

> **Note**
>
> If only CAN A or CAN B has been configured in the software for communications, only the corresponding pins must be connected.
>
> For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

> **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

> **Important**
>
> Integration is also possible by connecting CAN A of the **Autopilot 1x** to CAN B of the **MEX** and vice versa, i.e. it does not necessarily have to be CAN A-CAN A or CAN B-CAN B.
>
> However, any connections made must be **consistent** with the **configuration** made at software level in 1x PDI Builder and MEX PDI Builder.

RS232

For proper operation via **RS232**, the connection between **MEX** and **Autopilot 1x** pins should be like this:

<table>
<tr><th colspan="3">Autopilot 1x Harness</th><th colspan="2">MEX Connector</th></tr>
<tr><th>PIN</th><th>Signal</th><th>Color Code</th><th>PIN</th><th>Signal</th></tr>
<tr><td rowspan="2">19</td><td rowspan="2">RS 232 TX</td><td rowspan="2">White-Pink</td><td>16</td><td>RS-232 (A) RX</td></tr>
<tr><td>19</td><td>RS-232 (B) RX</td></tr>
<tr><td rowspan="2">20</td><td rowspan="2">RS 232 RX</td><td rowspan="2">Pink-Brown</td><td>15</td><td>RS-232 (A) TX</td></tr>
<tr><td>18</td><td>RS-232 (B) TX</td></tr>
<tr><td>21</td><td>GND</td><td>White-Blue</td><td>17</td><td>GND</td></tr>
</table>

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Important

Integration is possible by connecting the RS-232(A) or RS-232(B) from **MEX** to the RS232 of **Autopilot 1x**.

However, any connections made must be **consistent** with the **configuration** made at software level in 1x PDI Builder and MEX PDI Builder.

RS485

For proper operation via **RS485**, the connection between **MEX** and **Autopilot 1x** pins should be like this:

Autopilot 1x Harness			MEX Connector	
PIN	**Signal**	**Color Code**	**PIN**	**Signal**
50	OUT RS485_P	Pink-Brown	33	IN RS-485 (P)
51	OUT RS485_N	White-Blue	31	IN RS-485 (N)
52	IN RS485_N	Brown-Blue	32	OUT RS-485 (N)
53	IN RS485_P	White-Red	30	OUT RS-485 (P)
54	OUT_GND	Brown-Red	34	RS-485 GND

> ⚠ **Warning**
>
> Note that, in **Autopilot 1x**, this pin 54 is not a common GND pin.

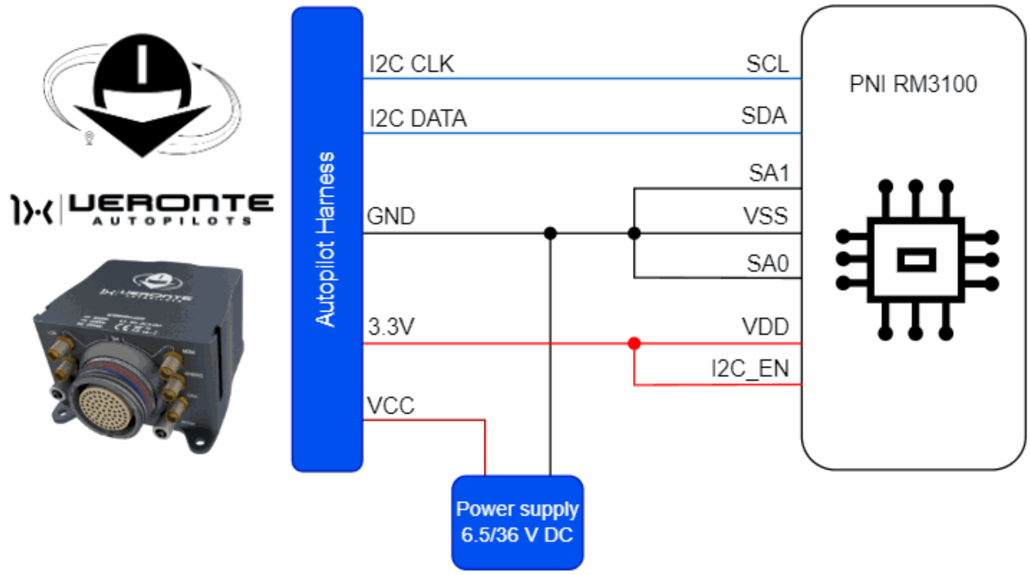
> **Important**
>
> Any connections made must be **consistent** with the **configuration** made at software level in 1x PDI Builder and MEX PDI Builder.

PNI RM3100

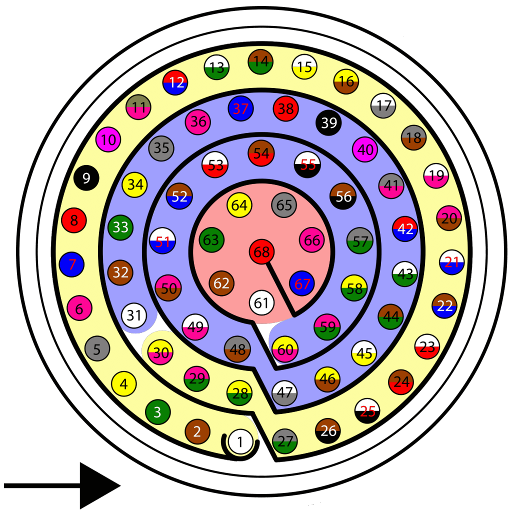
PNI RM3100-CB

PNI RM3100-CB magnetometer must be connected to **Autopilot 1x** via **I2C**.

> **Important**
>
> This integration example is described for the **RM3100-CB**.



PNI RM3100-CB - Autopilot 1x wiring diagram



Autopilot 1x harness pinout

<table>
<tr><th colspan="3">Autopilot 1x Harness</th><th colspan="2">PNI RM3100-CB Connector</th></tr>
<tr><th>PIN</th><th>Signal</th><th>Color Code</th><th>PIN</th><th>Signal</th></tr>
<tr><td>31</td><td>I2C_CLK</td><td>White</td><td>1</td><td>SCL</td></tr>
<tr><td>32</td><td>I2C_DATA</td><td>Brown</td><td>3</td><td>SDA</td></tr>
<tr><td rowspan="3">33</td><td rowspan="3">GND</td><td rowspan="3">Green</td><td>4</td><td>VSS
Ground</td></tr>
<tr><td>2</td><td>SA1
Bit 1 of slave address</td></tr>
<tr><td>7</td><td>SA0
Bit 0 of slave address</td></tr>
<tr><td>34</td><td>3.3V</td><td>Yellow</td><td>8</td><td>VDD
Supply Voltage</td></tr>
</table>

Autopilot 1x Harness			PNI RM3100-CB Connector	
PIN	Signal	Color Code	PIN	Signal
			5	I2C_EN I2C enable pin (HIGH = I2C)

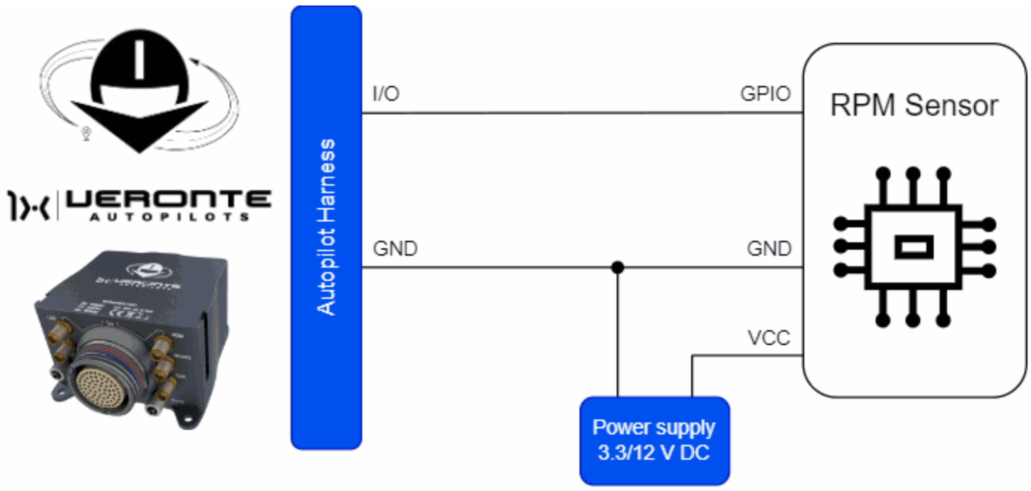
⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

Once the magnetometer is connected, the user must proceed to its software integration with **Veronte Autopilot 1x** by referring to the PNI RM3100 - Integration examples section of the **1x PDI Builder** user manual.

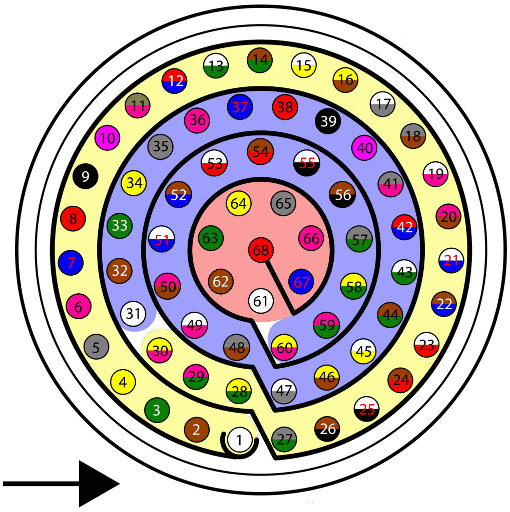
RPM Sensors

RPM sensors typically use different types of wiring depending on their technology and application. However, most RPM sensors, such as Hall effect, optical, or inductive sensors, generally have the following connection:



RPM Sensors - Autopilot 1x wiring diagram

RPM sensor must be connected to one of the available I/O pins of **Autopilot 1x**.



Autopilot 1x harness pinout

Autopilot 1x Harness		
PIN	**Signal**	**Color Code**
1	I/O0	White
2	I/O1	Brown
3	I/O2	Green
4	I/O3	Yellow
5	I/O4	Gray
6	I/O5	Pink
7	I/O6	Blue
8	I/O7	Red
9	GND	Black

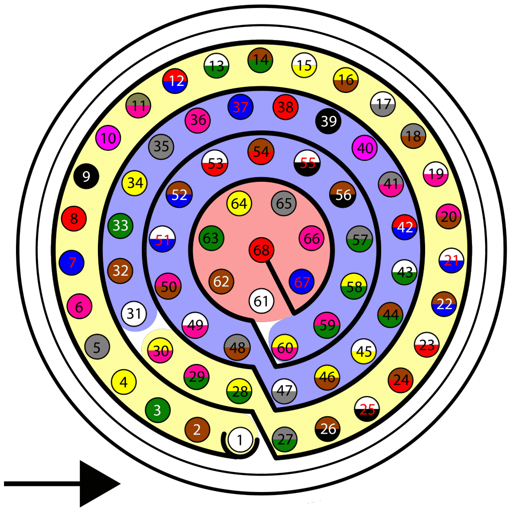
Autopilot 1x Harness		
PIN	**Signal**	**Color Code**
10	I/O8	Violet
11	I/O9	Gray-Pink
12	I/O10	Red-Blue
13	I/O11	White-Green
14	I/O12	Brown-Green
15	I/O13	White-Yellow
16	I/O14	Yellow-Brown
17	I/O15	White-Gray
55	EQEP_A	White-Black
56	EQEP_B	Brown-Black
57	EQEP_S	Gray-Green
58	EQEP_I	Yellow-Green
59	GND	Pink-Green

Once the **hardware installation** is complete, to properly integrate the device with **Autopilot 1x** follow the steps detailed in the RPM Sensors - Integration examples section of the **1x PDI Builder** user manual.

Stick

Veronte Autopilot 1x is compatible with joysticks that use PPM, CAN bus, USB, Serial, etc.

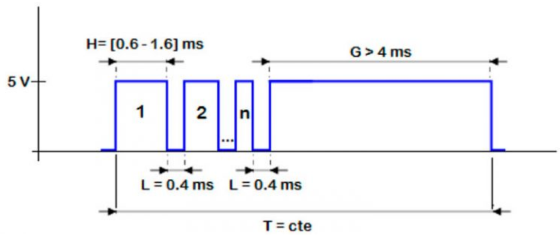
If the PPM level is 3.3V, the following **Autopilot 1x** pins can be used:



Autopilot 1x harness pinout

Autopilot 1x Harness		
PIN	**Signal**	**Color Code**
1	I/O0	White
2	I/O1	Brown
3	I/O2	Green
4	I/O3	Yellow
5	I/O4	Gray
6	I/O5	Pink
7	I/O6	Blue
8	I/O7	Red
9	GND	Black
10	I/O8	Violet
11	I/O9	Gray-Pink

Autopilot 1x Harness		
PIN	Signal	Color Code
12	I/O10	Red-Blue
13	I/O11	White-Green
14	I/O12	Brown-Green
15	I/O13	White-Yellow
16	I/O14	Yellow-Brown
17	I/O15	White-Gray
55	EQEP_A	White-Black
56	EQEP_B	Brown-Black
57	EQEP_S	Gray-Green
58	EQEP_I	Yellow-Green
59	GND	Pink-Green

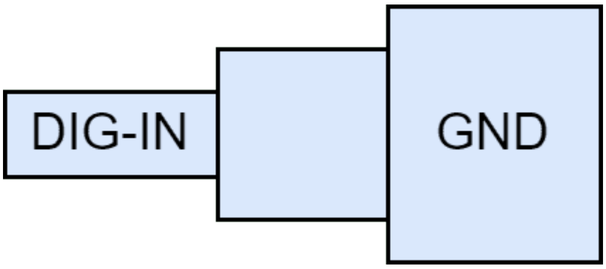


PPM signal

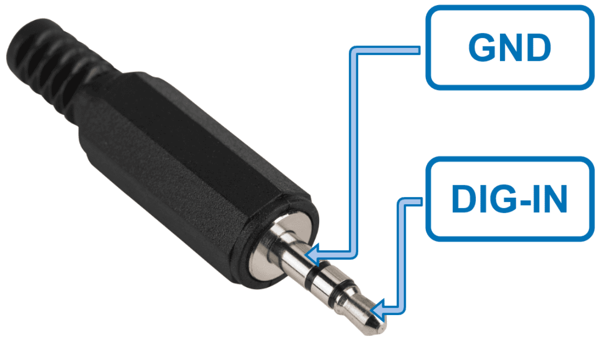
⚠ **Caution**

PPM signal must be into the **Veronte Autopilot 1x** voltage ranges. Some joysticks may need an adaptation board, please ask our team to check compatibility.

Connector for harness is provided with 3.5mm stereo plug connector as follows:



PPM pinout



PPM connector

- To use the joystick with **PPM** in the system, connect the PPMout of the trainer port to a digital input of **Veronte Autopilot 1x** and configure that digital input according to the PPM Stick - Integration examples section of the **1x PDI Builder** user manual.
- When using a **USB joystick**, the software installation with **Autopilot 1x** is detailed in the USB joystick - Integration examples section of the **1x PDI Builder** user manual.
- For joysticks with signals **different from PPM or USB**, read the Virtual Stick - Integration examples section of the **1x PDI Builder** user manual.

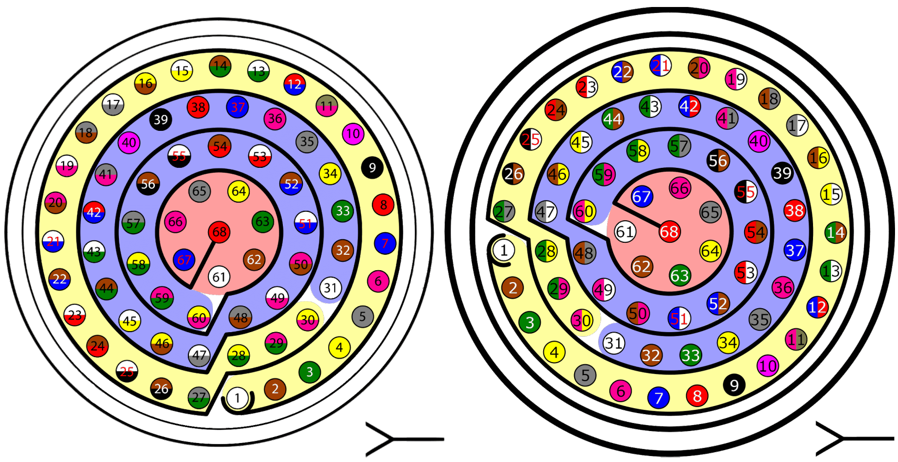
Veronte products

This section explains how integrate **Autopilot 1x** with Veronte products.

CEX connection

When communication is established between the PC and the **CEX** using the **Veronte Autopilot 1x** as a tunnel, the connection between the **CEX** and **Autopilot 1x** is via **CAN**.

The pin connection between the two devices should be like this:



Connector for Autopilot 1x - HEW.LM.368.XLNP (frontal view)

Connector for CEX - HEW.LM.368.XLNP (frontal view)

Autopilot 1x Connector			CEX Connector		
PIN	**Signal**	**Color Code**	**PIN**	**Signal**	**Color Code**
25	CANA_P	White-Black	5	CAN (A) P	Gray
26	CANA_N	Brown-Black	6	CAN (A) N	Pink
28	CANB_P	Yellow-Green	8	CAN (B) P	Red
29	CANB_N	Pink-Green	9	CAN (B) N	Black
30	GND		7		Blue

Autopilot 1x Connector			CEX Connector		
PIN	Signal	Color Code	PIN	Signal	Color Code
		Yellow-Pink		CAN GND	

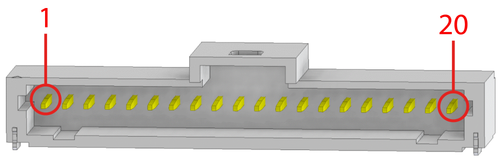
> **Note**
>
> If only CAN A or CAN B has been configured in the software for communications, only the corresponding pins must be connected.
>
> For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

> **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

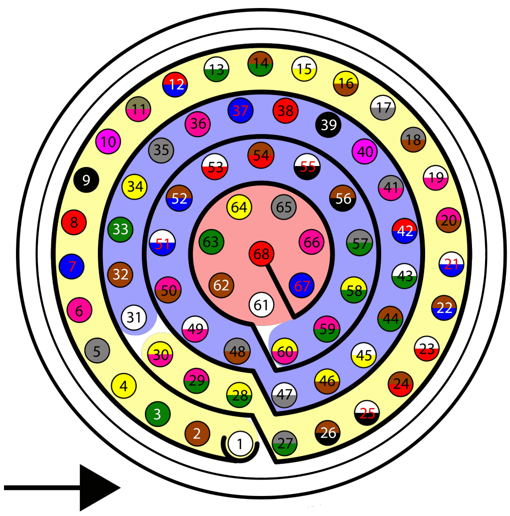
> **Important**
>
> Integration is also possible by connecting CAN A of the **Autopilot 1x** to CAN B of the **CEX** and vice versa, i.e. it does not necessarily have to be CAN A-CAN A or CAN B-CAN B.
>
> However, any connections made must be **consistent** with the **configuration** made at software level in 1x PDI Builder and CEX PDI Builder.

MC01 connection

For proper operation via **CAN**, the connection between **MC01** and **Autopilot 1x** pins should be like this:



MC01 connector pinout



Autopilot 1x harness pinout

Autopilot 1x harness			MC01 connector		
PIN	**Signal**	**Color code**	**PIN**	**Signal**	**Color code**
25	CANA_P	White-Black	9	CAN (P)	White
28	CANB_P	Yellow-Green			
26	CANA_N	Brown-Black	8	CAN (N)	Gray
29	CANB_N	Pink-Green			

Autopilot 1x harness			MC01 connector		
PIN	**Signal**	**Color code**	**PIN**	**Signal**	**Color code**
30	GND	Yellow-Pink	10	GND	Black

Note

CAN A and CAN B buses are equivalent and can be used interchangeably for the integration of this device.

For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

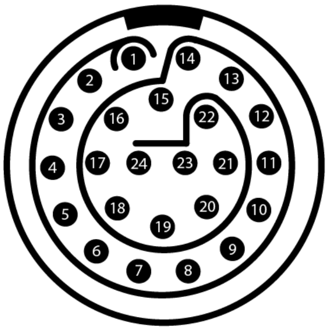
Warning

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

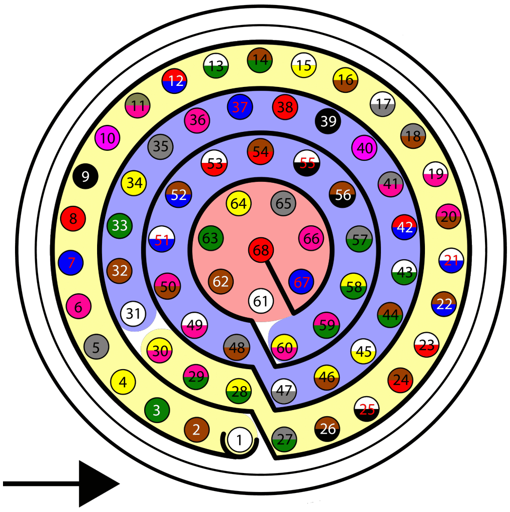
Once **MC01** has been properly wired with the **Autopilot 1x**, users can proceed to the software integration detailed in the MC01 - Integration examples section of the **1x PDI Builder** user manual.

MC24 connection

For proper operation via **CAN**, the connection between **MC24** and **Autopilot 1x** pins should be like this:



MC24 connector pinout



Autopilot 1x harness pinout

Autopilot 1x harness			MC24 connector	
PIN	**Signal**	**Color code**	**PIN**	**Signal**
25	CANA_P	White-Black	5	CANA_P
26	CANA_N	Brown-Black	6	CANA_N
28	CANB_P	Yellow-Green	18	CANB_P
29	CANB_N	Pink-Green	7	CANB_N

Autopilot 1x harness			MC24 connector	
PIN	Signal	Color code	PIN	Signal
30	GND	Yellow-Pink	24	CAN_GND

Note

If only CAN A or CAN B has been configured in the software for communications, only the corresponding pins must be connected.

For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

Warning

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

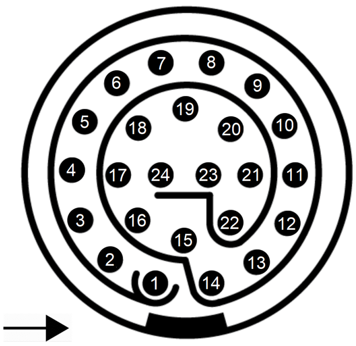
Important

Integration is also possible by connecting CAN A of the **Autopilot 1x** to CAN B of the **MC24** and vice versa, i.e. it does not necessarily have to be CAN A-CAN A or CAN B-CAN B.

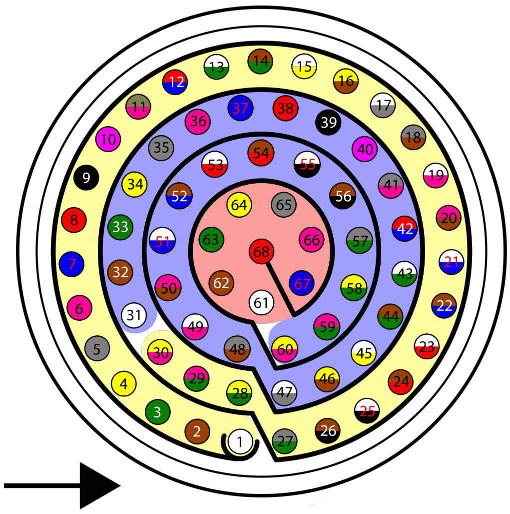
However, any connections made must be **consistent** with the **configuration** made at software level. For this, refer to the MC110/MC24 - Integration examples section of the **1x PDI Builder** user manual.

MC110 connection

For proper operation via **CAN**, the connection between **MC110 hardware version 1.2** and **Autopilot 1x** pins should be like this:



MC110 1.2 harness pinout



Autopilot 1x harness pinout

Autopilot 1x harness			MC110 harness	
PIN	**Signal**	**Color code**	**PIN**	**Signal**
25	CANA_P	White-Black	5	CANA_P
26	CANA_N	Brown-Black	6	CANA_N

Autopilot 1x harness			MC110 harness	
PIN	**Signal**	**Color code**	**PIN**	**Signal**
28	CANB_P	Yellow-Green	18	CANB_P
29	CANB_N	Pink-Green	7	CANB_N
30	GND	Yellow-Pink	24	CAN_GND

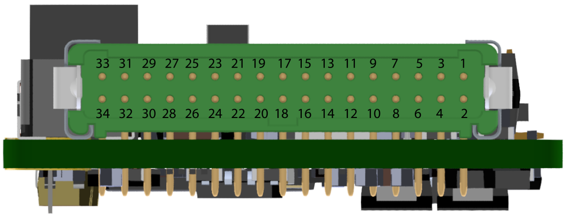
> ⓘ **Note**
>
> If only CAN A or CAN B has been configured in the software for communications, only the corresponding pins must be connected.
>
> For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

> ⚠ **Warning**
>
> **Remember!!** In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin. [!IMPORTANT] Integration is also possible by connecting CAN A of the **Autopilot 1x** to CAN B of the **MC110** and vice versa, i.e. it does not necessarily have to be CAN A-CAN A or CAN B-CAN B.
>
> However, any connections made must be **consistent** with the **configuration** made at software level. For this, refer to the MC110/MC24 - Integration examples section of the **1x PDI Builder** user manual.

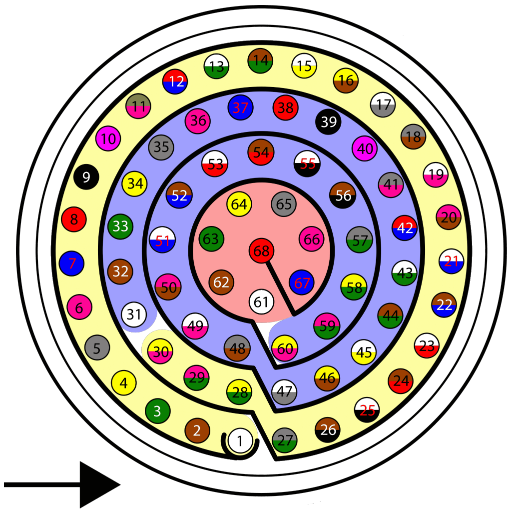
MEX connection

When communication is established between the PC and the **MEX** using the **Veronte Autopilot 1x** as a tunnel, the connection between the **MEX** and **Autopilot 1x** is via **CAN**.

The pin connection between the two devices should be like this:



MEX connector pinout



Autopilot 1x harness pinout

Autopilot 1x harness			MEX connector	
PIN	**Signal**	**Color code**	**PIN**	**Signal**
25	CANA_P	White-Black	22	CAN A (P)
26	CANA_N	Brown-Black	23	CAN A (N)

Autopilot 1x harness			MEX connector	
PIN	**Signal**	**Color code**	**PIN**	**Signal**
28	CANB_P	Yellow-Green	20	CAN B (P)
29	CANB_N	Pink-Green	21	CAN B (N)
30	GND	Yellow-Pink	24	CAN GND

ⓘ **Note**

If only CAN A or CAN B has been configured in the software for communications, only the corresponding pins must be connected.

For more information on CAN connection, please visit CAN - Wiring connection section of this manual.

⚠ **Warning**

Remember!! In **Autopilot 1x**, all GND pins are common. Note that pin 54 is not a common GND pin.

> **Important**
>
> Integration is also possible by connecting CAN A of the **Autopilot 1x** to CAN B of the **MEX** and vice versa, i.e. it does not necessarily have to be CAN A-CAN A or CAN B-CAN B.
>
> However, any connections made must be **consistent** with the **configuration** made at software level in 1x PDI Builder and MEX PDI Builder.

Troubleshooting

In case of any issue with software, read the Troubleshooting section of the **1x PDI Builder** user manual.

Maintenance mode

Maintenance mode is the main recovery mode that Veronte system puts at the user disposal. The main use of **maintanance mode** is to solve issues related to the current configuration, mainly related with communication or memory writting issues.

While in **maintenance mode**, **all communications channels are enabled** by default, so it is possible to connect **Veronte Autopilot 1x** through any of its configuration interfaces, regardless of its current configuration. Thus allowing to re-establish communications with it in case they have been lost for any reason.

> **Tip**
>
> It is heavily recommended to always use **maintenance mode** to load a new configuration that is very different from the current one.

> ⚠ **Warning**
>
> **Autopilot 1x** might enter in **maintenance mode** if a problem with the power supply is detected upon boot up (voltage or current is out of range).

How to enter in maintenance mode

There are two ways to enter in **maintenance mode**: by software or hardware (forcing it).

Using software to enter in maintenance mode

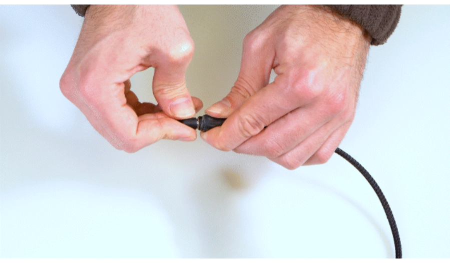
To enter in **maintenance mode** by software, read the Maintenance Mode - Troubleshooting section of the **1x PDI Builder** user manual.

Forcing maintenance mode

There are two ways to force the maintenance mode: using **power supply** or using the **I2C pins**.

Power supply

In order to active **maintenance mode**, power cycle the **Veronte Autopilot 1x** repetively with periods of 700 ms (with a margin range between 380 and 965 ms). After 30 cycles, the autopilot will enter in **maintenance mode**.



How to power cycle an autopilot

I2C pins

To enter in **maintenance mode** with **I2C**:

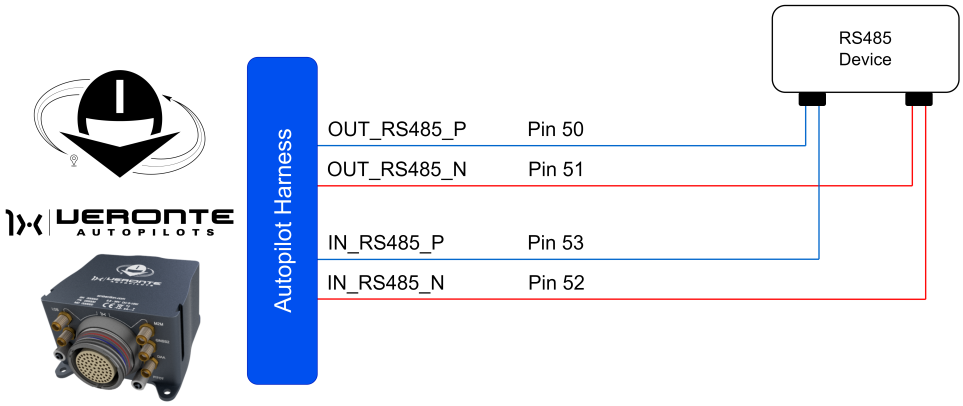
1. Unplug **Veronte Autopilot 1x**
2. Connect both I2C pins each other
3. Then, power up **Autopilot 1x**
4. Finally disconnect both pins

Both pins are I2C_CLK (number 31) and I2C_DATA (number 32) according to the pinout.

> **Note**
>
> **Dev Harness 1x 4.8** (Embention reference `P007043`) has already included a button with this 2 pins to easily enter maintenance mode.
>
> The procedure is the same as for the pins, but instead of connecting and disconnecting the pins, press and release the button.

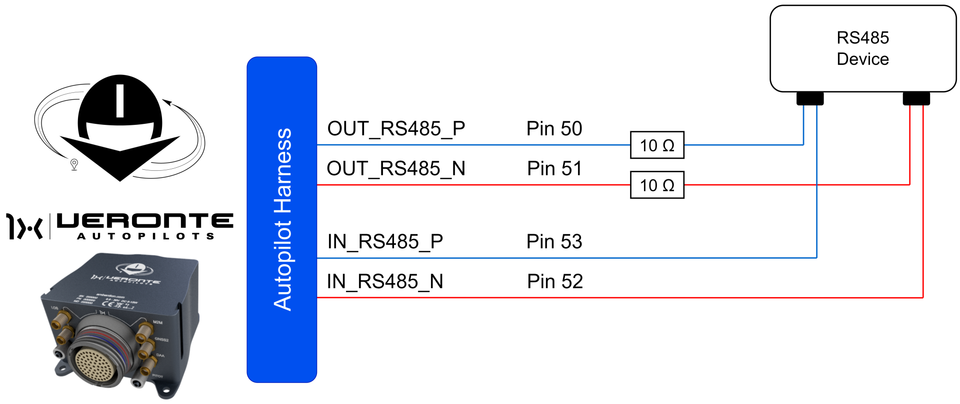
Half-duplex servo does not respond

Any servo with half duplex RS-485 should communicate with **Autopilot 1x** following the connection diagram:



Normal connection between 1x and servo

Sometimes this connection does not work, because the servo has not enough transmission power. In this case, a couple of 10 Ω resistors may solve the problem. Both resistors have to be placed at the trasmission line of the **Autopilot 1x**.



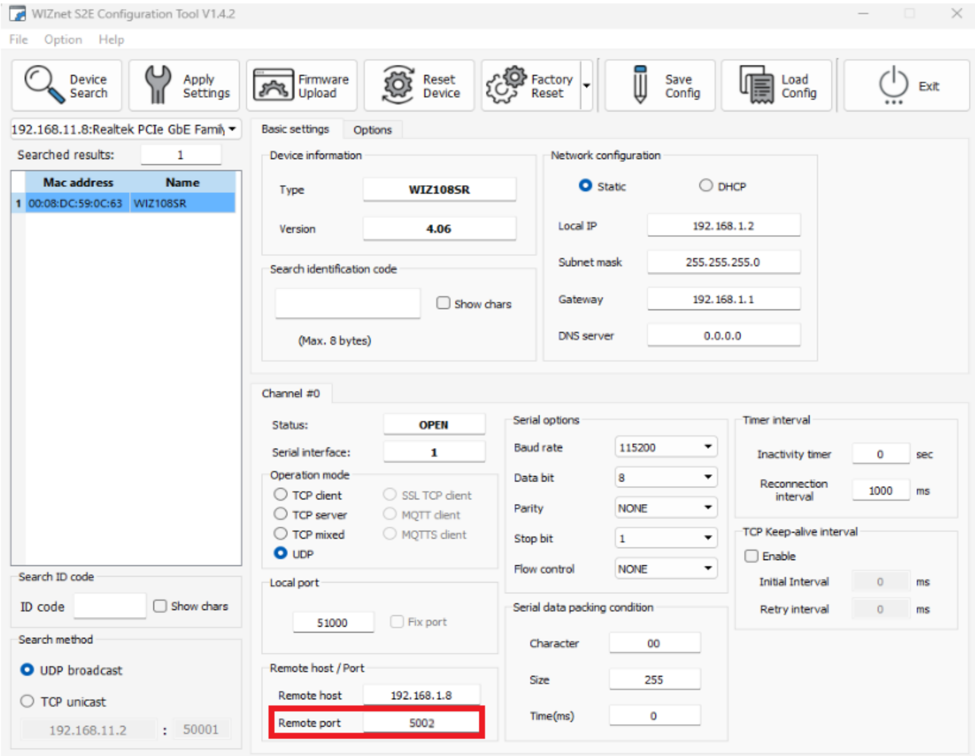
Resistors connection between 1x and servo

If the couple of resistors does not solve the issue, the user should contact the support team (create a ticket in the customer's **Joint Collaboration Framework**; for more information, see Tickets section of the **JCF** manual).

UDP Failed Connection

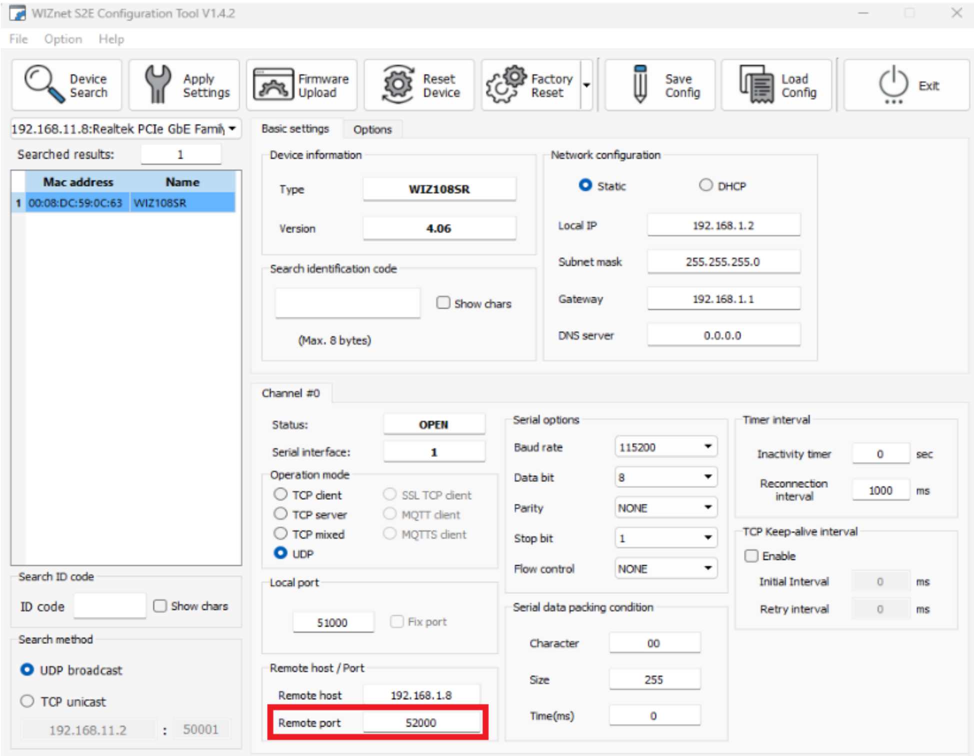
If setting up the autopilot with an RS-485 connection via UDP on port 5002, and it is not detected by **Veronte Link**, please check the following steps:

1. Open **Wiznet Software** and check Remote Port.



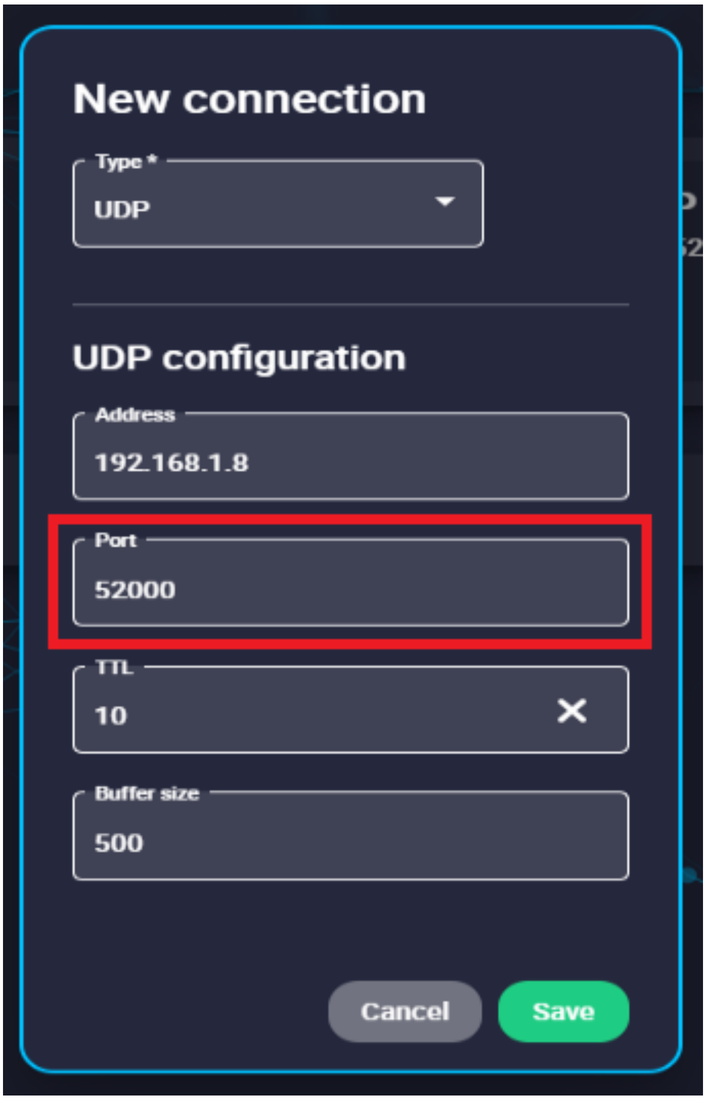
WIZnet Configuration Panel

2. Change Remote Port 5002 to a different port. In this example, Remote Port 52000 is set.



WIZnet Configuration Panel

3. Afterward, configure the UDP Connection in **Veronte Link** with the same parameter value.



UDP Connection settings

4. Click on **Save**.

Hardware Changelog

Hereby are described the main differences between the latest release of the **Veronte Autopilot 1x** hardware (v **4.8**) and the previous commercial version (v **4.5**).

	
Veronte Autopilot 1x 4.5	Veronte Autopilot 1x 4.8

Specifications

Mechanical		
	4.5	**4.8**
Mating connector	Circular mating connector with 68 pins	
Enclosure	Anodized aluminum	
Weight	190 g	198 g
Dimensions	78 x 63 x 40 mm	76 x 65 x 40 mm
Protection rating	IP67	
Mounting	M4 screws	M3 screws
	-40 to 65 ºC	

Mechanical		
	4.5	**4.8**
Temperature range (no convection)		
Pressure port	2.4 mm	SMC M5 series system. Usable with 2.5 x 4 mm polyurethane
RF connectors (LOS/BLOS/ GNSS)	SSMA jack female	

Sensors		
	4.5	**4.8**
Number of static pressure sensors	3	2
Static pressure range	**0)** 0 - 103,000 Pa **1)** 1,000 - 120,000 Pa **2)** 30,000 - 120,000 Pa	**1)** 1,000 - 120,000 Pa **2)** 30,000 - 120,000 Pa
Static pressure band error	**0)** 1,030 Pa **1)** 500 Pa **2)** 200 Pa	**1)** 500 Pa **2)** 200 Pa

Sensors		
	4.5	**4.8**
Static pressure resolution	**0)** 25 Pa **1)** 1.2 to 6.5 Pa **2)** 0.5 Pa	**1)** 1.2 to 6.5 Pa **2)** 0.5 Pa
Number of dynamic pressure sensors	1	
Dynamic pressure range	3 Pa (5 kt 8 km/h sea level) to 6,900 Pa (206 kt 382 km/h sea level)	
Dynamic pressure band error	0.01 %	
Dynamic pressure resolution	25,500 Pa	
Number of accelerometers (3 axis each)	2	3
Accelerometer range	**1)** ±16 g **2)** ±24 g	**1)** ±16 g **2)** ±24 g **3)** ±8 g
Accelerometer max. shock	**1)** 20,000 for 0.2 ms **2)** 10,000 g/ms	**1)** 20,000 g for 0.2 ms **2)** 10,000 g/ms

Sensors		
	4.5	**4.8**
		3) 14,700 m/s²
Accelerometer sensitivity	**1)** 16,393 LSB/(m/s²) **2)** 10,920 LSB/(m/s²)	**1)** 16,393 LSB/(m/s²) **2)** 10,920 LSB/(m/s²) **3)** 26,756,268 LSB/(m/s²)
Number of gyroscopes (3 axis each)	2	3
Gyroscope range	**1)** 125 to 2,000 º/s **2)** 125 to 2,000 º/s	**1)** 125 to 2,000 º/s **2)** 125 to 2,000 º/s **3)** 2,000 º/s
Gyroscope sensitivity	**1)** 228 to 14.2 LSB/º/s **2)** 262 to 16 LSB/º/s	**1)** 228 to 14.2 LSB/º/s **2)** 262 to 16 LSB/º/s **3)** 655,360 to 10 LSB/º/s
Number of magnetometers	2	3
Magnetometer range	**1)** 4 gauss **2)** 8 gauss	

Sensors		
	4.5	**4.8**
		0) 4 gauss **1)** 8 gauss **2)** 11 gauss
Magnetometer sensitivity	**1)** 6,842 to 1,711 LSB/gauss **2)** 4,096 LSB/gauss	**0)** 6,842 to 1,711 LSB/gauss **1)** 4,096 LSB/gauss **2)** 0.13 mgauss
Number of GNSS units	2	
GNSS constellations	BeiDou, GLONASS, GPS	BeiDou, Galileo, GLONASS, GPS/QZSS
Concurrent GNSS constellations	Up to 2	Up to 4
GNSS bands	L2OF, L2C, E1B/C, B2I, E5b, L1C/A, L1OF, B1I	
Position accuracy RTK	0.025 m + 1 ppm CEP	> 0.01 m + 1 ppm CEP
Update rate	8 Hz	up to 20 Hz

I/O (on base hardware - expansion boards available)		
	4.5	**4.8**
Vin	2 x (6.5 - 36 V) DC	
PWM / GPIO	Up to 16	
RS232	1 x	
RS485	1 x	
Vout	5 & 3.3 V	
FTS	Deadman output (GPIO)	
CAN bus	2 x	
ADC	5 x	
EQEP	1 x	
I2C	1 x	
UART	1 x (FTS - SuC)	1 x Microcontroller (i.e. external datalink)
USB	1 x	

Architecture and Computing Power

Veronte Autopilot 1x 4.8 and 4.5 share the same internal architecture, powered by the same dual core microprocessor from the Texas Instruments EP (Enhanced Performance) series.

Line of Sight Communications

Veronte Autopilot 1x 4.5 had the possibility to install internal datalinks from Microhard (Pico Series), available in 2.4 GHz, 900 MHz and 400 MHz. In the new version (4.8), this datalink has been replaced by a short range module for testing, configuration and telemetry download.

Embention has decided to replace the 1W Microhard datalink for the following reasons:

- As per past experience, each aircraft manufacturer has different datalink needs, so Embention has preferred that customers can have more flexibility choosing the appropriate brand, model and frequency.
- By removing the Microhard modules, there is now more space to include better sensors with extended accuracy.
- Reducing the power of the internal RF module from the autopilot permits to reduce electromagnetic interferences.
- Reducing the power consumption also reduces the heat dissipation needs on the autopilot and improves performance in warm environments.

Thus, Autopilot 1x 4.8 internal datalink has been replaced by Digi XBee® 3 PRO Zigbee 3.0, a 2.4 GHz short range module. This module has the mission to allow the integration and PDI tuning of the autopilot into the customer's platform. An external datalink (Microhard or others) can be installed in the additional UART port available in 4.8.

Performance for Digi XBee® 3 PRO Zigbee 3.0	
Data rate	RF 250 kbps, serial up to 1 Mbps
Indoor/urban range	Up to 300 ft (90 m)

Performance for Digi XBee® 3 PRO Zigbee 3.0	
Outdoor/RF line-of-sight range	Up to 2 miles (3200 m)
Transmit power	+19 dBm
Receiver sensitivity (1% PER)	-103 dBm normal model

Features	
Frequency band	ISM 2.4 GHz
Interference immunity	DSSS (Direct Sequence Spread Spectrum)

Networking and Security	
Protocol	Zigbee 3.0
Encryption	128/256 bit AES
Reliable packet delivery	Retries / acknowledgements
IDS	

Networking and Security	
	PAN ID and addresses, cluster IDs and endpoints (optional)
Channels	16 channels

New Features

One of our fundamentals at Embention is the continuous evolution of our products and services. The UAM sector is evolving everydays and new needs and technical requirements are demanded almost every week. For this reason, in order to help our customers to achieve their goals, Embention has released a new version of its hardware, with the following features.

- **Enhanced sensor accuracy and reliability**: Dead reckoning navigation is a critical tool for degraded GNSS scenarios, the new sensors in Veronte Autopilot 1x are significantly more accurate than previous ones.

  These enhanced sensors allows to improve the navigation estimation during the standard operation and in GNSS denied navigation.
- **Improved RTK and GNSS heading**: New GNSS sensors in Veronte Autopilot 1x are significantly better in various aspects (supported constellations, accuracy, number of concurrent constellations and frequency).

  The positioning accuracy has improved from 0.025 m + 1 ppm to 0.01 m + 1 ppm CEP, it also improves RTK and GNSS heading estimation.
- **Enhanced high-temperature hardware design and production materials**: In order to offer the best performances in extreme scenarios and to pass more demanding quality and environmental tests.

  New materials are used in PCB manufacturing. These new fibers have better performance in heat conditions, extending the lifetime of the product.
- **Additional failure detection means**: Development of failure detection algorithms and redundant hardware.

The new PCB design includes additional signals for failure detection that can communicate each one of the autopilot cores with the arbiter. Also, the internal CAN Bus network has been modified to enhance robustness.

- **Short-range 2.4 GHz LOS included in all units**: Short range communications module included, in order to facilitate system integration.

  This module is based on Digi Xbee and permits to have an independent link for telemetry download and configuration.
- **BLOS module with extended coverage**: BLOS communications and UTM capabilities through a physical or eSIM card.
- **Extended GNSS-denied navigation capabilities**: Flexible and custom Kalman filters, high precision IMU sensors and several tools in order to build safety automations if GNSS is lost.

  Inertial navigation highly depends on the IMU accuracy, which has been significantly improved with the new setup.
- **Internal Remote ID or ADS-B with IN/OUT**: Extended collaborative sense and avoid capabilites, allowing the compliance of new FAA unmanned operation requirements.

  WIFI and Bluetooth modules have been integrated to enable Remote ID communications. This module can be replaced by a 1W ADS-B IN & OUT module if preferred.
- **Embedded global DEM, geoid, magnetic and gravity fields**: Advanced custom possibilites for the user during the map configuration, in order to carry out high-accuracy operations in the safest way possible.

  Worldwide terrain altitude and magnetic field are included in Veronte Autopilot 1x.
- **Internal enclosure pressure port**: A new pressure port has been included for measuring the pressure inside the autopilot.

Collaborative Sense & Avoid and UTM

Some of the most important aeronautical authorities in the world have been very clear about the future lines in the certification process: all aircraft have to have collaborative detection capabilities. For example, the American FAA has expressed in several documents that a Remote ID system for drone identification and detection for smaller drones. For larger drones, it is expected that the ADS-B technology will prevail.

Remote ID

Remote ID is a young technology based on personal area networks (Bluetooth) and local area networks (WiFi). The system is emitting a constant signal with the ID of the drone and its positioning at each moment. Users can see the track of the operation and customers have the ability to configure the information shown.

Frequency	WiFi & Bluetooth bands
Developed according to	RIN 2120-AL31 Remote identification of Unmanned Aircraft FAA Standard
Parameters	Aircraft ID, position, altitude and time mark
Compatibility	FAA Remote ID Scanner App

To comply with the final rule of Remote Identification of Unmanned Aircraft (Part 89), manufacturers need to produce standard Remote ID drones and meet the requirements of this rule by September 16th, 2022.

Embention's Remote ID is being adapted in order to be in complicance with Active Standard ASTM F3411 requirements, developed by subcommittee F38.02.

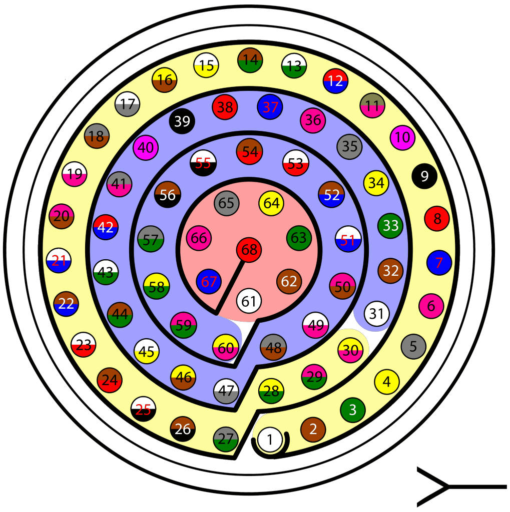
ADS-B

Internal ADS-B with IN & OUT capabilities is another option in order to reach the collaborative detection capabilities and to enhance the fleet management performance.

Frequency band	1090 MHz
Current consumption	Averaged 140 mA
Sensitivity	-80 dBm
RF output power	Configurable + 30 dBm (1W), + 27 dBm (0.5 W), + 24 dBm (0.25 W)
ESD protection	All lines protected
MAVlink (baud)	115200 bps
AERO (baud)	115200 bps (AT commands)

Pinout changes from Autopilot 1x 4.5

The pinout for `4.5` and `4.8` versions are very similar, but they have several differences. To prevent any confusion, the following table shows the pinout for both versions. The different pins are marked with ⚠, all the rest have the same function.



68 pin connector for both versions

PIN	Signal	Type	Description
1	I/O0	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
2	I/O1	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
3	I/O2	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
4	I/O3	I/O	PWM / Digital I/O signal (0-3.3V). Protected against

PIN	Signal	Type	Description
			ESD and short circuit
5	I/O4	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
6	I/O5	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
7	I/O6	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
8	I/O7	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
9	GND	GROUND	Ground signal for actuators 1-8
10	I/O8	I/O	PWM / Digital I/O signal (0-3.3V). Protected against

PIN	Signal	Type	Description
			ESD and short circuit
11	I/O9	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
12	I/O10	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
13	I/O11	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
14	I/O12	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
15	I/O13	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
16	I/O14	I/O	PWM / Digital I/O signal (0-3.3V).

PIN	Signal	Type	Description
			Protected against ESD and short circuit
17	I/O15	I/O	PWM / Digital I/O signal (0-3.3V). Protected against ESD and short circuit
18	GND	GROUND	Ground signal for actuators 9-16
19	RS 232 TX	Output	RS 232 Output (-13.2V to 13.2V Max, -5.4V to 5.4V Typical). Protected against ESD and short circuit
20	RS 232 RX	Input	RS 232 Input (-25V to 25V Max, -0.6V Low and 2.4V High Threshold). Protected against ESD and short circuit
21	GND	GROUND	Ground signal for buses
22	ANALOG_3	Input Analog	Input 0-3.3V. Protected against

PIN	Signal	Type	Description
			ESD and short circuit
23	ANALOG_4	Input Analog	Input 0-3.3V. Protected against ESD and short circuit
24	GND	GROUND	Ground signal for buses
25	CANA_P	I/O	CANbus interface, up to 1Mbps (2.3V Typical, 1.2V-2.3V Differential). Protected against ESD
26	CANA_N	I/O	Twisted pair with a 120 ohms Zo recommended (2.3V Typical, 1.2V-2.3V Differential). Protected against ESD
27 ⚠	`4.5:` GND	`4.5:` GROUND	`4.5:` Ground signal for buses
	`4.8:` 4XV_WD	`4.8:` I/O	`4.8:` Reserved. Do not connect
28	CANB_P	I/O	CANbus interface. It supports data

PIN	Signal	Type	Description
			rates up to 1 Mbps. Protected against ESD
29	CANB_N	I/O	Twisted pair with a 120 ohms Zo recommended. Protected against ESD
30	GND	GROUND	Ground signal for buses
31	I2C_CLK	Output	Clk line for I2C bus (0.3V to 3.3V). Protected against ESD and short circuit
32	I2C_DATA	I/O	Data line for I2C bus (0.3V to 3.3V). Protected against ESD and short circuit
33	GND	GROUND	Ground for 3.3V power supply
34	3.3V	POWER	3.3V - 100mA power supply. Protected against ESD short circuit with 100mA resettable fuse

PIN	Signal	Type	Description
35	GND	GROUND	Ground for 5V power supply
36	5V	POWER	5V - 100mA power supply. Protected against ESD short circuit with 100mA resettable fuse
37	GND	GROUND	Ground for analog signals
38	ANALOG_0	Input	Analog input 0-3.3V. Protected against ESD and short circuit
39	ANALOG_1	Input	Analog input 0-3.3V. Protected against ESD and short circuit
40	ANALOG_2	Input	Analog input 0-3.3V. Protected against ESD and short circuit
41 ⚠	`4.5:` GND	`4.5:` GROUND	`4.5:` Ground for FTS signals
	`4.8:` 4XV_A	`4.8:` I/O	`4.8:` Reserved. Do not connect

PIN	Signal	Type	Description
42	FTS1_OUT	Output	Deadman signal from comicro. Protected against ESD and short circuit
43	FTS2_OUT	Output	!SystemOK Bit. Protected against ESD and short circuit
44 ⚠	`4.5:` GND	`4.5:` GROUND	`4.5:` Ground signal for safety buses
	`4.8:` 4XV_B	`4.8:` I/O	`4.8:` Reserved. Do not connect
45 ⚠	`4.5:` V_ARB_TX	`4.5:` Output	`4.5:` Veronte comicro UART output to activate safety mechanism. Protected against ESD and short circuit
	`4.8:` UARTA_TX	`4.8:` Output	`4.8:` Microcontroller UART
46 ⚠	`4.5:` V_ARB_RX	`4.5:` Input	`4.5:` Veronte comicro UART output to activate

PIN	Signal	Type	Description
			safety mechanism. Protected against ESD and short circuit
	`4.8:` UARTA_RX	`4.8:` Input	`4.8:` Microcontroller UART
47	GND	GROUND	Ground signal comicro power supply
48	V_ARB_VCC	POWER	Veronte comicro power (6.5V to 36V). Protected against ESD and reverse polarity
49	FTS3_OUT_MPU	Output	MPU alive voting signal, to use with 4xVeronte. It is a Square Wave at [100,125] Hz. Protected against ESD and short circuit
50	OUT_RS485_P	Output	Non-inverted output from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against

PIN	Signal	Type	Description
			ESD and short circuit
51	OUT_RS485_N	Output	Inverted output from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit
52	IN_RS485_N	Input	Inverted input from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit
53	IN_RS485_P	Input	Non-inverted input from RS485 bus (-7V to 12V Max, -2.3V to 2.3V Typical). Protected against ESD and short circuit
54	OUT_GND	GND	Ground for RS-485 bus **Warning** This is not a common GND pin.

PIN	Signal	Type	Description
55	EQEP_A	I/O	DIGITAL output / DIGITAL input / Encoder quadrature input A (0-3.3V). Protected against ESD and short circuit
56	EQEP_B	I/O	DIGITAL output / DIGITAL input / Encoder quadrature input B (0-3.3V). Protected against ESD and short circuit **Warning** Only use it as digital I/O with Veronte units of Hardware version 4.5 or lower.
57	EQEP_S	I/O	DIGITAL output / DIGITAL input / Encoder strobe input (0-3.3V). Protected against ESD and short circuit

PIN	Signal	Type	Description
58	EQEP_I	I/O	DIGITAL output / DIGITAL input / Encoder index input A (0-3.3V). Protected against ESD and short circuit
59	GND	GROUND	Ground for encoders
60	V_USB_DP	I/O	Veronte USB data line. Protected against ESD
61	V_USB_DN	I/O	Veronte USB data line. Protected against ESD
62 ⚠	`4.5:` V_USB_ID	`4.5:` I/O	`4.5:` Veronte USB ID line. Protected against ESD and short circuit
	`4.8:` USB_SHIELD_GND	`4.8:` GROUND	`4.8:` USB cable shielding
63	FTS_OUT_MPU	Output	Abort mission voting signal from MPU, to use with 4xVeronte. Bit Low (0V) if mission OK. High (3.3V) if mission

PIN	Signal	Type	Description
			wants to be terminated. Protected against ESD and short circuit
64	FTS2_OUT_MPU	Output	Abort mission voting signal 2 from MPU, to use with 4xVeronte. Bit Low (0V) if mission OK. High (3.3V) if mission wants to be terminated. Protected against ESD and short circuit
65	GND	GROUND	Veronte ground input
66	GND	GROUND	Veronte ground input
67	VCC	POWER	Veronte power supply (6.5V to 36V). Protected against ESD and reverse polarity. **Warning** Both pins are common. They MUST be

PIN	Signal	Type	Description
			connected to the same power supply.
68	VCC	POWER	

> ⚠ **Warning**
>
> **Remember!!** All GND pins are common. Note that pin 54 is not a common GND pin.

Acronyms and Definitions

Acronyms

Acronym	Description
16 VAR	16 Bits variables (Integers)
32 VAR	32 Bits variables (Reals)
ADC	Analog to Digital Converter
ADSB	Automatic Dependent Surveillance-Broadcast
AGL	Above Ground Level
AoA	Angle of Attack
ARC	Arcade Mode
AUTO	Automatic Mode
BIT	Bit Variables
BLOS	Beyond Line Of Sight
CAN	Controller Area Network
CAP	Capture Module

Acronym	Description
CEP	Circular Error Probability
CMB	Climb Phase
CRU	Cruise Phase
DAA	Detect And Avoid
DC	Direct Current
DGPS	Differential GPS
ECAP	Enhanced CAP
ECEF	Earth Centered - Earth Fixed
EGNOS	European Geostationary Navigation Overlay Service
(E)GPRS	Enhanced Data Rates for GSM Evolution
EKF	Extended Kalman Filter
EQEP	Enhanced Quadrature Encoder Pulse
ESC	

Acronym	Description
	Electronic Speed Controller
eVTOL	Electric Vertical Take Off and Landing
FCS	Flight Control System
FHSS	Frequency Hopping Spread Spectrum
FLR	Flare Phase
FTS	Flight Termination System
GIS	Geographical Information System
GND	Ground
GNSS	Global Navigation Satellite Systems
GPIO	General Purpose Input Output
GPS	Global Positioning System
GS	Ground Speed
GS	Ground Segment
GSM	

Acronym	Description
	Global System for Mobile Communications
HLD	Hold Phase
HSPA+	High Speed Packet Access Plus
HUM	Hardware User Manual
I2C	Inter-Integrated Circuit
IAS	Indicated Air Speed
ID	Identification
Int. D.	Internal Diameter
IMU	Inertial Measurement Unit
ISM	Industrial Scientific and Medical
LED	Light-Emitting Diode
LND	Landing Phase
LOS	Line Of Sight
M2M	Machine To Machine
MSL	Mean Sea Level

Acronym	Description
OPV	Optionally Piloted Vehicle
Out. D.	Outer Diameter
PCS	Pole Control Station
PFD	Primary Flight Display
PID	Proportional Integral Derivative
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
QNH	Barometric atmospheric pressure adjusted to sea level
QZSS	Quasi-Zenith Satellite System
RC	Radio Control Mode
RF	Radio Frequency
RPAS	Remotely Piloted Aircraft System
RPM	

Acronym	Description
	Revolutions Per Minute
RS 232	Recommended Standard 232
RS 485	Recommended Standard 485
RX	Reception
SMA	SubMiniature Version A Connector
SSMA	Miniature-SMA
STB	Standby Phase
SU	Servo-Output matrix
TAS	True Air Speed
TKO	Take Off Phase
TX	Transmission
UART	Universal asynchronous receiver-transmitter
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle

Acronym	Description
UMTS	Universal Mobile Telecommunications System
US	Output-Servo matrix
VTOL	Vertical Take Off and Landing
WGS 84	World Geodetic System 84
WP	Waypoint

Definitions

- **Control Phase:** The operation is divided into phases in which the UAV has a specific performance. Each of this phases is called a control phase.
- **Control Channel:** It is each of the signals used to control a behaviour or action.
- **Control Mode:** It is possible to make a manual control of the UAV by stick, assisted control and fully automatic control.
- **Actuator:** It is a mechanic device to provide force to move or "act" another mechanical device.

Contact Data

For support-related inquiries, customers have access to a dedicated portal through the Joint Collaboration Framework. This platform facilitates communication and ensures traceability of all support requests, helping us to address your needs efficiently.

For other questions or general inquiries, you can reach us via email at sales@embention.com or by phone at (+34) 965 115 421