Modbus RTU and SDI-12 bidirectional adapter

WiConvert

WiConvert Installation and Connection Manual

Manual

Changes Log

Review	Date	Reason for the amendment
1	07/05/2025	Initial issue of the document
2	19/08/2026	Content review

Content

Figures

1. Installation and wiring precautions

The power inputs are protected against incorrect wiring.

The controller board contains sensitive electronic components that may be damaged by electrostatic discharge (ESD), so it must be handled with care.

Before connecting the device, we recommend following these steps to avoid damaging it:

- Disconnect the device from the mains.
- Do not remove the electronic components from their casing to avoid damage caused by short circuits.
- Avoid touching the electronic components directly with your hands, as this could cause an electrostatic discharge in the components and damage some of them.
- Before switching on the device, check that all cables are correctly connected.
- Avoid handling the circuit board whilst the device is switched on to prevent damage.

2. Overview of the equipment

The **WiConvert is a protocol converter that enables bidirectional conversion between Modbus RTU and SDI-12.**

The device can operate in two modes:

- **Modbus RTU → SDI-12:** receives data via RS-485 or RS-232 using the Modbus RTU protocol and converts it to SDI-12.
- **SDI-12 → Modbus RTU:** receives data via SDI-12 and converts it to Modbus RTU, using RS-485 or RS-232 for communication.

Previously, the WiConvert was supplied in two different models, depending on the operating mode:

- **WiConvert-MS:** conversion from Modbus RTU to SDI-12.
- **WiConvert-SM:** conversion from SDI-12 to Modbus RTU.

Currently, the operating mode is configured via the device's configuration tool, so there is no need for a different product reference for each mode. The user can switch between both operating modes as required.

To configure the WiConvert, the WiLink4 configuration cable is required.

Important: Modbus compatibility

Of the three main variants of the Modbus protocol:

- **Modbus RTU**
- **Modbus ASCII**
- **Modbus TCP/IP**

WiConvert is only compatible with Modbus RTU.

3. General connection diagram

Figure 1 below shows the general connection diagram for the WiConvert and the location of its main components.

The unit comprises the following components:

- On the left, **the SDI-12 terminal block and the device's power input.**
- On the right, the Modbus terminal block and the power output for the device to be read.
- **Configuration jumpers:** located on the right-hand side of the unit and used to configure various operating options, which are described in later sections.
- **Upper 4-pin connector:** used to connect the WiLink4 configuration cable.

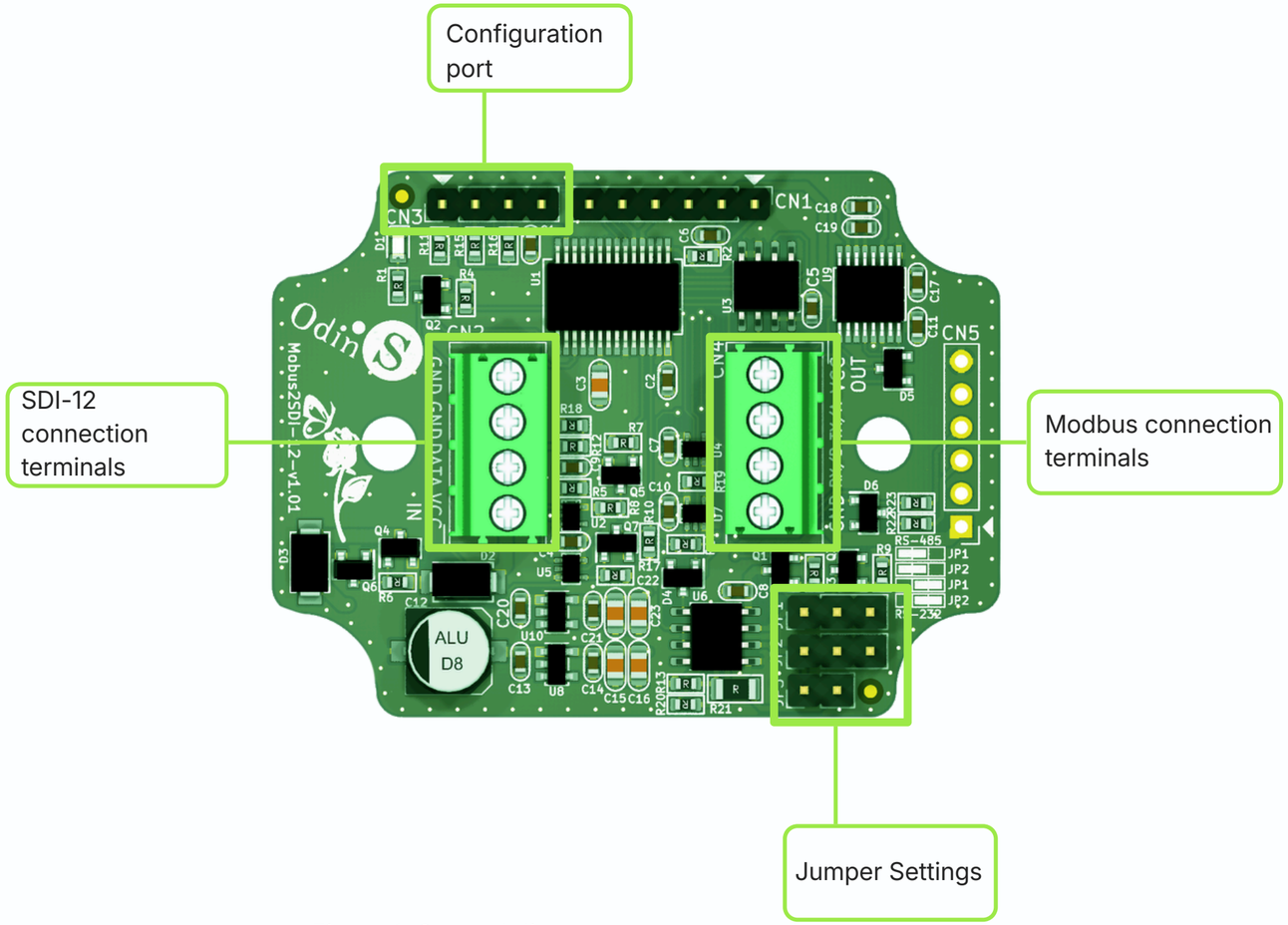

Figure 1. General diagram

4. Connecting the various components

4.1. SDI-12 interface connection

The **SDI-12 interface terminal block** has the following connections:

- VCC IN: 12 V DC power supply input.
- DATA: SDI-12 data bus connection.
- GND: has two terminals, one for the negative power supply terminal and the other for the SDI-12 bus ground connection.

The following figures show the **wiring diagrams** for each WiConvert operating mode:

- **Modbus RTU → SDI-12**
- **SDI-12 → Modbus RTU**

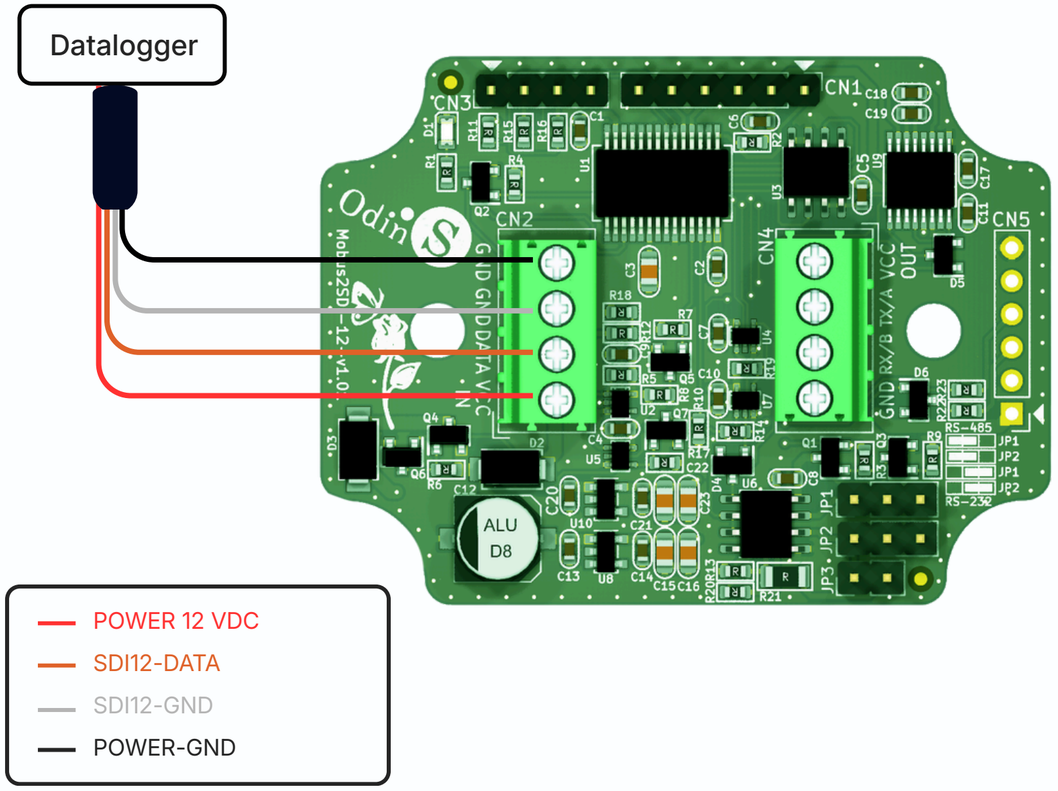

Figure 2. SDI-12 cabling for a WiConvert Modbus to SDI-12 converter

4.2. SDI-12 interface connection

Figure 3 shows the connection diagram for the SDI-12 interface, which may vary slightly depending on the intended use of the equipment.

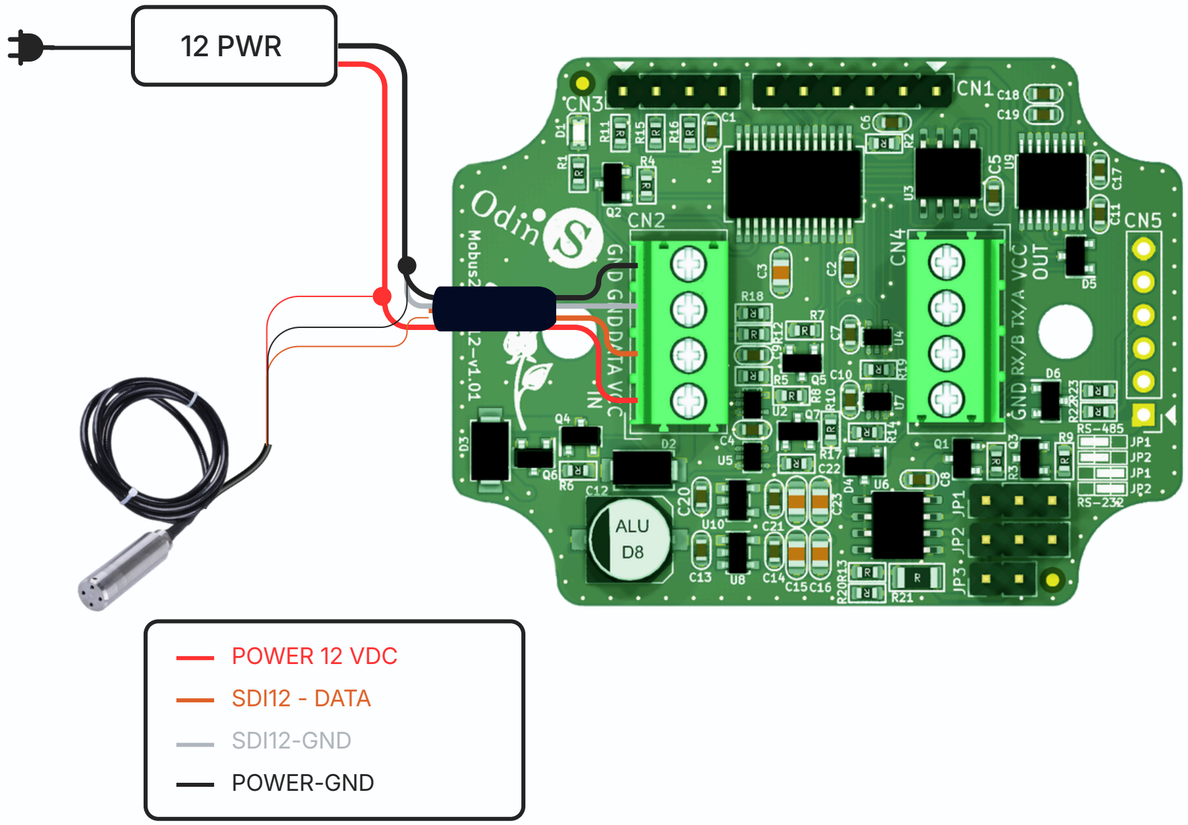

Figure 3. SDI-12 cabling for a WiConvert SDI-12 to Modbus converter

4.3. Modbus RTU interface connection

For the Modbus connection, the WiConvert has the following terminals:

- **VCC OUT:** 12 V power output.
- **TX/A and RX/B:** two communication terminals for the Modbus interface.
- **GND:** earth connection.

The WiConvert has two physical interfaces for Modbus communication: **RS-232 and RS-485**. The interface to be used is selected via jumpers **JP1 and JP2**.

Furthermore, when using the **RS-485 interface, jumper JP3** allows the bus terminating resistor to be enabled.

Figures 4 and 5 below show the different wiring diagrams for this terminal block, depending on the Modbus interface used.

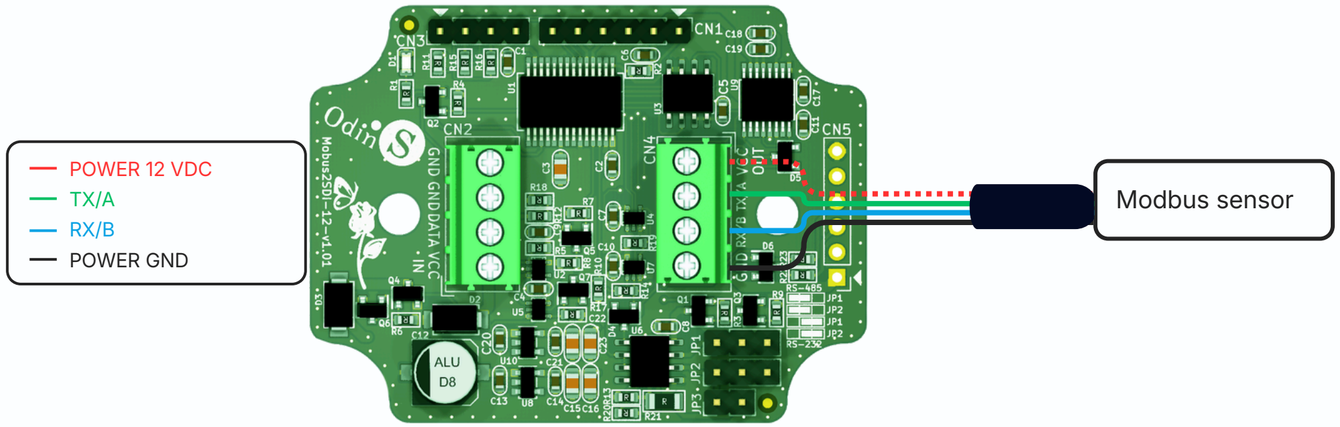

Figure 4. WiConvert Modbus connection from Modbus to SDI-12

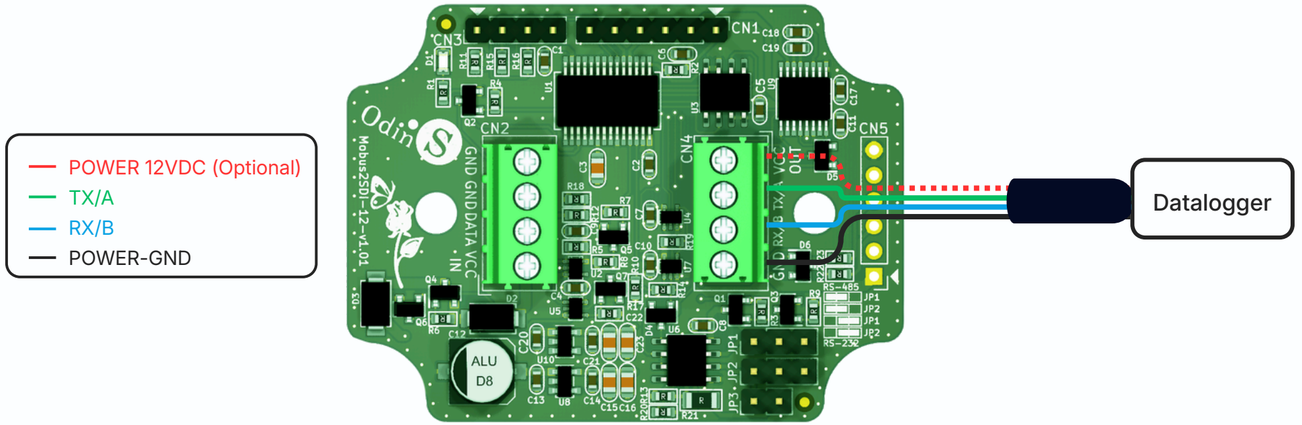

Figure 5. WiConvert SDI-12 to Modbus connection

4.4. Configuración de la interfaz física Modbus RTU

As mentioned above, the Modbus RTU physical interface can be either RS-232 or RS-485 and is configured using jumpers JP1 and JP2, as shown in Figure 6 below:

Figure 6. Selecting the Modbus RTU physical interface

When using the **RS-485 interface**, it may be necessary to enable the 120 Ω bus terminating resistor. The WiConvert incorporates this resistor, which is disabled by default.

The resistor is activated via jumper JP3.

Important: The terminating resistor should only be enabled when the WiConvert is at the start or end of the RS-485 bus. It must not be enabled if the device is located at an intermediate point on the bus.

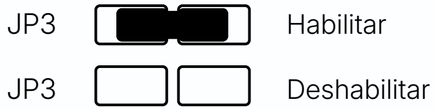

Figure 7. Enable/Disable termination resistor

4.5. Connecting the WiLink4 configuration cable

To set up the **WiConvert**, you need to use the **WiLink4** configuration cable.

The cable **has a 4-pin connector** which must be connected to the equipment in the correct orientation. **Pin 1 of the cable must line up with position 1 of the connector**, marked with a small white triangle, as shown in the figure.

Important: It is essential to ensure the connector is correctly oriented to avoid malfunctions or potential damage to the equipment or the configuration cable.

Once the USB cable has been connected to the computer, the operating system will detect a new serial port **COMxx** if you are using Windows, or **/dev/ttyUSBxx** if you are using Linux **(where xx represents the port number).**

Important: Throughout the entire setup process, the WiConvert must remain connected to a 12 VDC power supply.

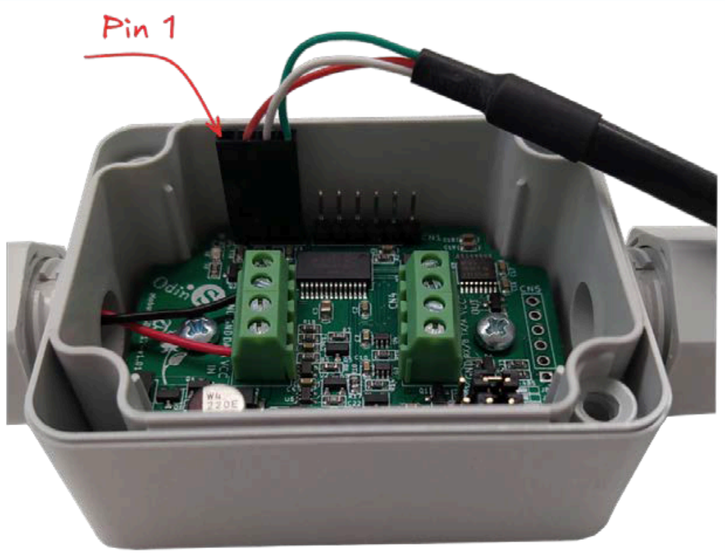

Figure 8. Connecting the USB configuration cable

5. Setting up a WiConvert

To configure WiConvert, follow these steps:

1. Access the web application: Go to the configuration platform at https://odct.odins.es/.

2. Download and run BridgeTool

If you have not installed it previously, download the BridgeTool from the web application. This tool enables communication between the web platform and the device.

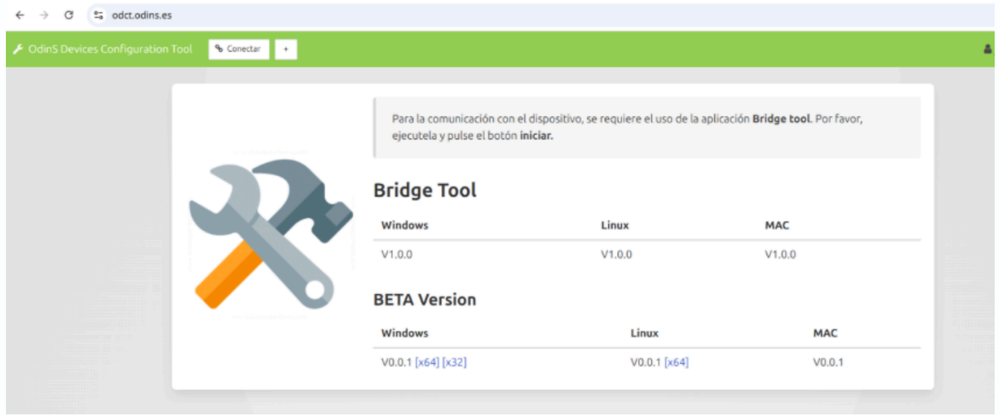

Figure 9. Main view of the web application.

Once downloaded, launch the app and tap the start button. You can then minimise the window, but do not close it, otherwise the app will be unable to communicate with the device.

Figura 10. BridgeTool

3. Connect the device via the platform

In the web application, click the **‘Connect’** button in the top-left corner.

A screen will appear where you must select:

- **The local serial port (COMxx)** associated with the WiLink4 cable.
- **The device type:** WiConvert.
- **The configuration application.**

4. Start the configuration

Once you have selected the above parameters, click the green **‘Start’** button to establish communication and begin configuring the device.

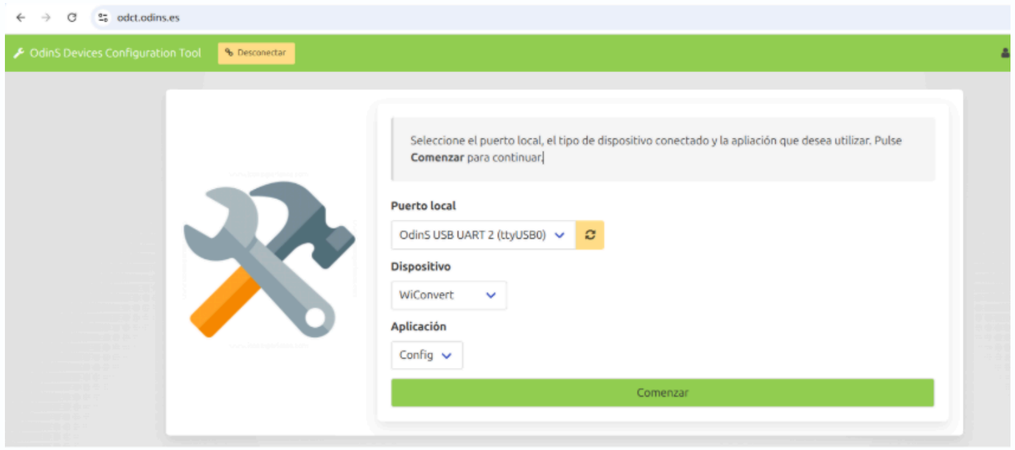

Figure 11. Port and device selection.

Figure 12 below shows the options available for the device:

- **Read device settings.** Allows you to read the settings stored on the device and modify them.
- **Load settings from file.** Allows you to load a file containing the desired settings from your PC.
- **Clear settings.** This deletes all settings from the device and resets it to its factory settings.

In all three of the above cases, once the relevant action has been carried out, a form will appear for you to enter or modify the device's settings, as shown in the image below.

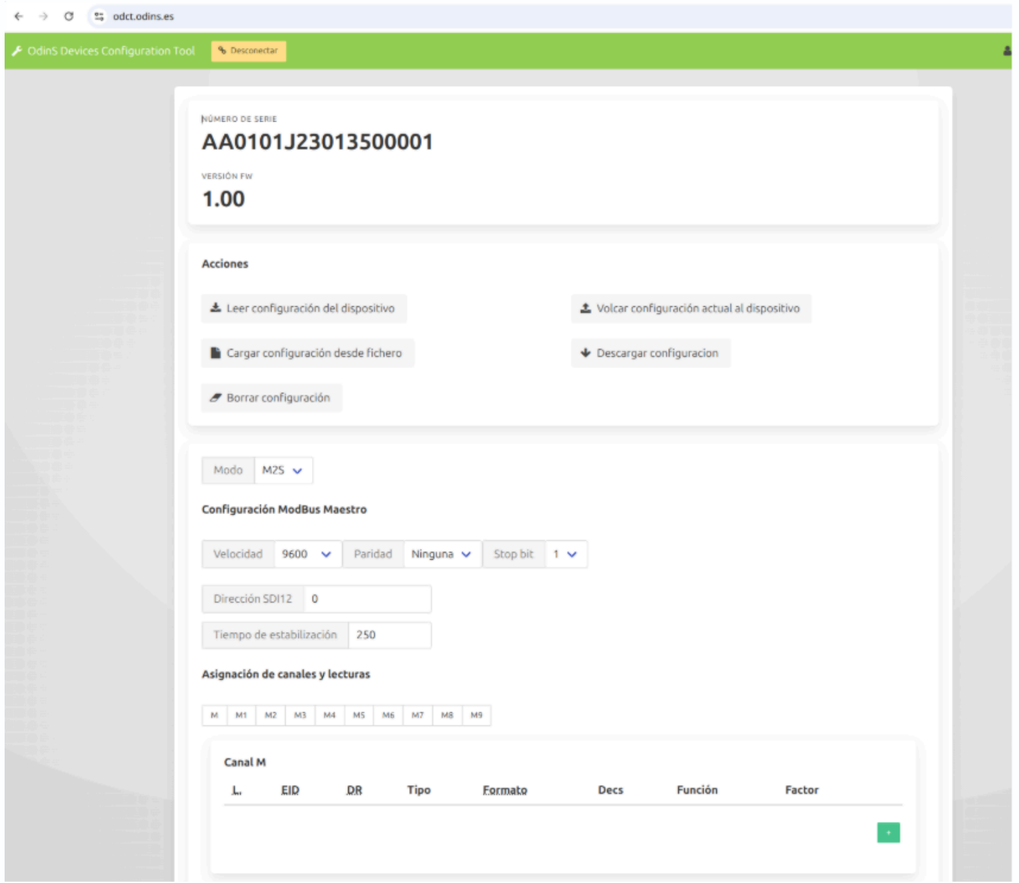

Figure 12. Configuration form

5.1. Configuring a WiConvert as a Modbus-to-SDI-12 converter

To configure the **WiConvert as a Modbus to SDI-12 converter, select M2S mode.** In this mode, the WiConvert manages the two communication interfaces as follows:

- **Modbus:** acts as the Master interface.
- **SDI-12:** acts as the Slave interface.

General parameters
Configure the following parameters:

- **Modbus baud rate:** the communication speed of the Modbus interface. The available values are 1200, 2400, 4800, 9600 and 19200 baud.

- **Parity:** allows you to select the type of parity used in Modbus communication: none, even or odd.

- **Stop bits:** number of stop bits used in Modbus communication: 1 or 2 bits.

- **SDI-12 address:** the address that the WiConvert will use to respond to SDI-12 requests and provide the data obtained via Modbus. Any character from 0 to 9, a–z or A–Z may be used. Numeric addresses from 0 to 9 are typically used.

- **Stabilisation time:** the time, expressed in milliseconds, that the WiConvert waits after powering up the Modbus sensor before taking a reading. This time allows the sensor to reach a stable state and become available for communication.

Configuring channels and readings
Once the general parameters have been configured, you must assign the Modbus registers to the SDI-12 channels and readings.

To add a reading:

1. **Select the channel you wish to configure (M, M1, ..., M9).**
2. **Press the green + button to the right of the channel form.**
3. **A new reading will be added to the selected SDI-12 channel.**
4. **The new reading is disabled by default. To enable it, press the button at the far right of the reading.**

In this way, you can configure the various SDI-12 readings associated with the Modbus registers you wish to monitor.

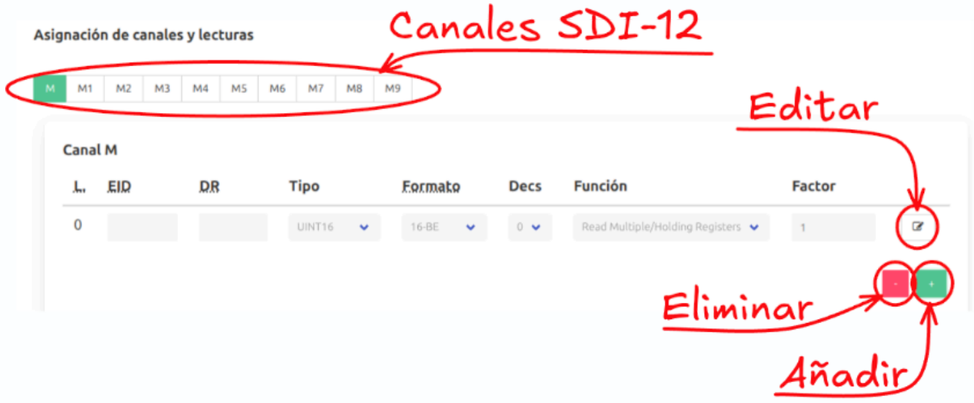

Figure 13. Add, remove or edit a channel reading.

The data to be entered by the user are:

- **L.** SDI-12 reading offset assigned to the Modbus register within the channel. The offset is assigned automatically when a reading is added and is a value between 0 and 8 (9 readings in total).
- **EID.** Modbus slave ID, i.e. the address assigned to the Modbus sensor so that its registers can be read. This is normally a value between 1 and 247.
- **DR.** Start offset of the Modbus register to be read. Value between 0 and 65535.
- **Tipo.** Data type stored in the Modbus register. It can be one of the following: 16-bit integer (INT16), 16-bit unsigned integer (UINT16), 32-bit integer (INT32), 32-bit unsigned integer (UINT32), 32-bit floating-point (FLOAT32), 64-bit floating-point (DOUBLE64), BCD16 and BCD32.
- **Formato.** Specifies how the bytes within the register or registers to be read are encoded. This can be 16-bit big-endian, 16-bit little-endian, 32-bit big-endian, 32-bit little-endian, 32-bit big-endian swapped bytes, 64-bit big-endian or 64-bit little-endian.
- **Decs.** The number of decimal places used to represent the value read from the Modbus register. A value between 0 and 3.
- **Función.** Modbus function code for reading the register. The two supported functions are 'Read Holding Registers' and 'Read Input Registers'.
- **Factor.** Multiplier factor for the value read from the register. In some cases, the value stored in the Modbus register is multiplied by 10, 100, etc., so if we wish to obtain its original value, we must apply the inverse of this multiplier, i.e. 1/10 or 1/100.

Note: *The maximum number of values that can be mapped from Modbus to SDI-12 is 90, i.e. 10 channels × 9 readings per channel.*

Once the configuration is complete, **it can be uploaded to the device and/or saved to a file on the PC for later use.** To do this, click the relevant button.

5.1.1. Formato de datos del registro Modbus

As mentioned in the previous section, data in Modbus registers can be encoded in two main ways when it exceeds the size of a single register (16 bits): big-endian and little-endian.

By default, Modbus uses **Big-Endian** encoding, where bytes are stored in their natural read order; that is, the most significant byte is located at the highest memory address and the least significant byte at the lowest. Conversely, **Little-Endian** encoding stores bytes in the opposite order, with the most significant byte at the lowest memory address and the least significant byte at the highest.

The following table shows an example of the three possible formats for 32-bit data, which occupy two Modbus registers.

Data to be stored: 1-2-3-4			
Format		**Registro offset 1**	**Registro offset 0**
Big-endian	1-2-3-4	1-2	3-4
Bytes exchanged in big-endian format	3-4-1-2	3-4	1-2
Little-endian	4-3-2-1	4-3	2-1

Table 1. Modbus data format

5.2. Configuring a WiConvert as an SDI-12 to Modbus converter

As in the previous case, the first step is to select the S2M operating mode on the WiConvert. In this instance, this means that the SDI-12 interface acts as the master and the Modbus interface as the slave.

The general parameters to be configured are:

- **Modbus speed.** Baud rate or operating speed of the Modbus interface. The supported speeds are 1200, 2400, 4800, 9600 and 19200.
- **Parity.** Whether or not to use parity on the Modbus interface. The possible options are none, even or odd.
- **Stop bits.** The number of stop bits to be used on the Modbus interface. The available options are 1 or 2 bits.
- **Modbus address.** The assigned Modbus slave address. This is a value between 1 and 247.
- **Overall time between readings.** This is the total reading time, in seconds, for the connected SDI-12 sensors. If set to 0, the value configured individually for each sensor slot is used.

In this operating mode, up to 16 SDI-12 sensors (slots) can be configured, as shown in the figure below. Furthermore, to assist with configuration, a summary table containing the Modbus registers corresponding to the configured sensors is included at the end of the form.

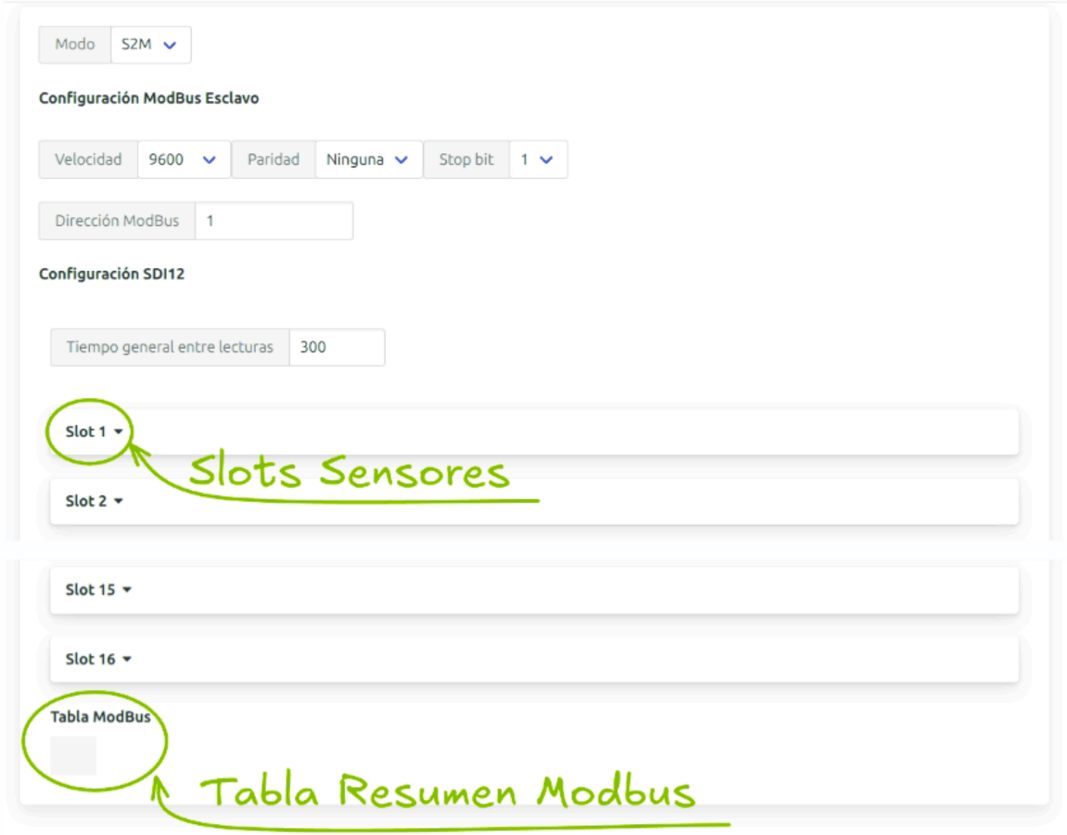

Figure 14. SD-12 to Modbus configuration form

To configure an SDI-12 sensor, you must expand the relevant slot and enable it. Once enabled, you can enter the parameters required to read the SDI-12 sensor.

Configurable slot parameters:

- **Enabled.** Enables or disables the slot.
- **Address.** The address assigned to the SDI-12 sensor to be read from that slot (addresses must not be duplicated, otherwise there will be a bus conflict).
- **Time between readings.** The time in seconds between two sensor readings, if the general time between readings is not enabled.
- **Stabilisation time.** The wait time, in milliseconds, from when power is supplied until communication with the sensor is possible.
- **Active channels.** Enables or disables the channels we wish to read.
- **Readings.** Enables or disables the specific reading within the channel we wish to read.

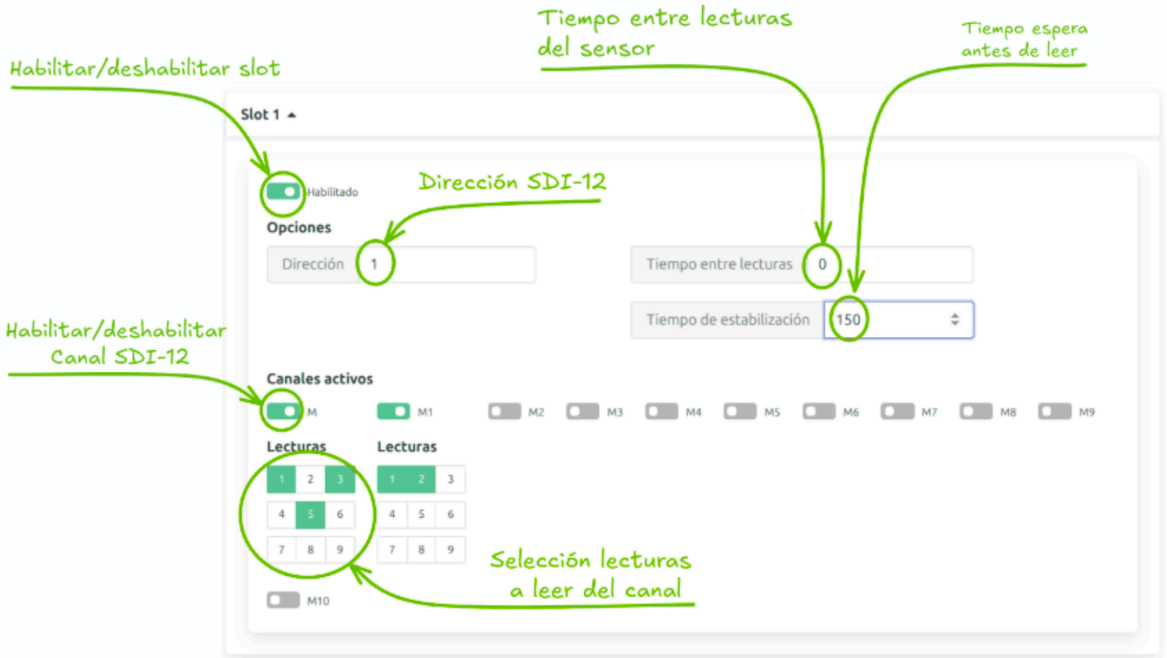

Figure 15. SDI-12 sensor configuration

Tabla ModBus

```
Slot 1 - SDI12 Address: 1
 Ch:M
   SDI12 Read:1 ModBus Address:0x07d0 Type:FLOAT32
   SDI12 Read:3 ModBus Address:0x07d4 Type:FLOAT32
   SDI12 Read:5 ModBus Address:0x07d8 Type:FLOAT32
 Ch:M1
   SDI12 Read:1 ModBus Address:0x07e2 Type:FLOAT32
   SDI12 Read:2 ModBus Address:0x07e4 Type:FLOAT32
```

Figure 16. Summary table of Modbus from the previous example.

5. References

[1] Odin Solutions, WiConvert – Technical specifications

WiConvert Visual

Let´s invent Smart Solutions!

Odin Solutions, S.L.

Calle Palma de Mallorca 2,
30009, Murcia, ESPAÑA
Tel.: +34 868 123 395

Email: comercial@odins.es
Web: www.odins.es

Scan the QR code and get in touch with us.

Discover our vision, projects and solutions designed with you in mind.