

# Software Manual for Stock Level Monitoring System



Technical support: [support@terabee.com](mailto:support@terabee.com)  
Sales and commercial support: [terabee-sales@terabee.com](mailto:terabee-sales@terabee.com)

# Table of contents

|   |           |
|---|-----------|
| <b>Introduction</b>   | <b>2</b>  |
| About Terabee Stock Level Monitoring System                 | 3         |
| Data triggering from sensors                                | 3         |
| Stock Level Supervisor data definition                      | 4         |
| Wi-Fi ID of the Stock Level Supervisor                      | 5         |
| <b>Connect with Stock Level Supervisor</b>                  | <b>6</b>  |
| Using embedded access point (Wi-Fi)                         | 6         |
| Using Ethernet  | 7         |
| <b>Configure The Stock Level Supervisor</b>                 | <b>7</b>  |
| Configurator setup guide                                    | 8         |
| <b>Get the IP address of the Stock Level Supervisor</b>     | <b>14</b> |
| Windows OS  | 14        |
| Plug the supervisor unit directly to your PC                | 14        |
| Passing through a router with DHCP server                   | 16        |
| Linux (Ubuntu 16.04)  | 19        |
| Plug the supervisor unit directly to your PC                | 19        |
| Passing through a router with DHCP server                   | 20        |
| <b>Quick Start: Communicate from a Modbus TCP/IP client</b> | <b>23</b> |
| Quick Start with QModBus for Linux OS                       | 23        |
| Connecting to QModBus                                       | 23        |
| Read raw distance registers example                         | 25        |
| Read volume data example                                    | 27        |
| Read level data example                                     | 28        |
| Manual trigger example                                      | 28        |
| Quick Start with Modbus Doctor on Windows                   | 30        |
| Connecting to Modbus Doctor                                 | 30        |
| Read raw sensor data example                                | 31        |
| Read volume data example                                    | 32        |
| Read level data example                                     | 33        |
| Manual trigger example                                      | 34        |
| <b>Stock Level Supervisor Operation</b>                     | <b>35</b> |
| What happens when a measurement is triggered successfully?  | 36        |
| How to detect an error during a measurement procedure?      | 36        |
| Error codes   | 37        |
| <b>Modbus Registers</b>                                     | <b>38</b> |
| Coils   | 39        |

|   |    |
|---|----|
| Trigger Signal                              | 39 |
| Discrete Inputs (Read Only Flags)           | 39 |
| Recent Measurement Flag                     | 40 |
| New Error Flag                              | 40 |
| Level Flags                                 | 41 |
| EMPTY                                       | 41 |
| LOW LEVEL                                   | 41 |
| ONE_QUARTER                                 | 41 |
| HALF FULL                                   | 41 |
| THREE_QUARTERS                              | 41 |
| FULL  | 41 |
| Input Registers (Read Only)                 | 41 |
| Error Register                              | 42 |
| Distance Registers                          | 43 |
| Latest Measurement State Register           | 43 |
| Volume Registers                            | 44 |
| Level Register                              | 44 |
| Sensor Failure Indicator registers          | 44 |
| Current Configuration Registers             | 45 |
| Container type                              | 45 |
| Material type                               | 45 |
| First Container's dimension                 | 45 |
| Second Container's dimension                | 45 |
| Third Container's dimension                 | 45 |
| Fourth Container's dimension                | 46 |
| Number of sensors                           | 46 |
| Current IP address                          | 46 |
| Ethernet Subnet Mask                        | 46 |
| Product ID Registers                        | 46 |
| Transformed Level Registers                 | 47 |
| Holding registers (Configurable Parameters) | 47 |

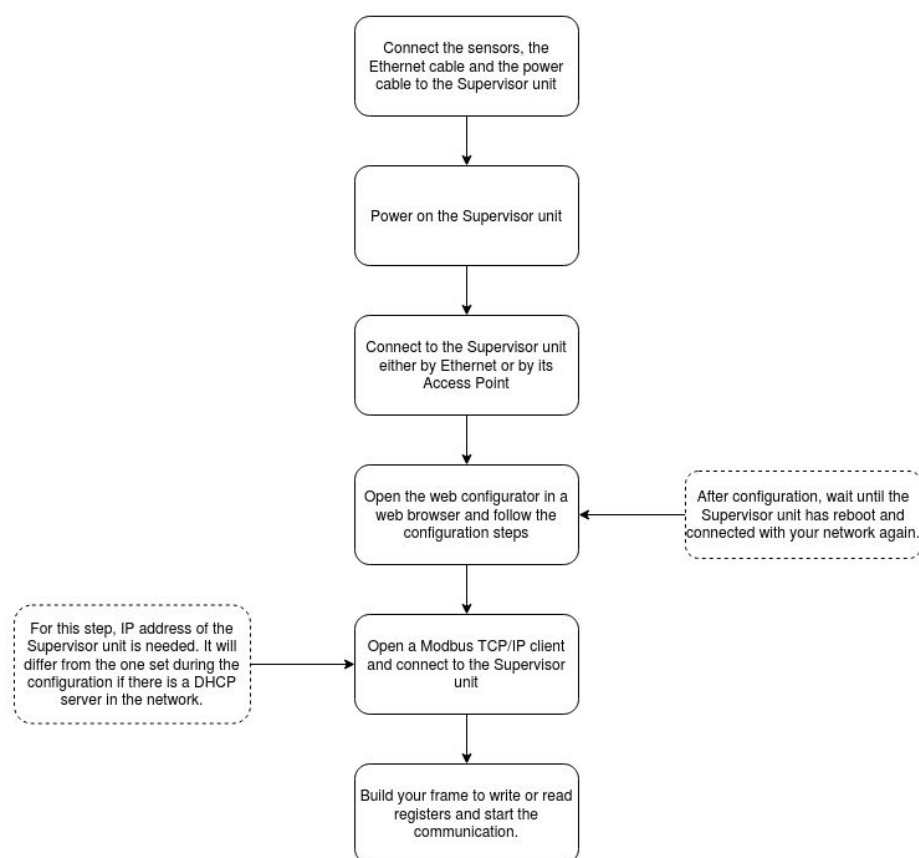
# Introduction

The purpose of this document is to give guidelines on the Terabee Stock Level Monitoring System configuration, using the web-based configuration software and connectivity setup for Ethernet and Wi-Fi interfaces. Other information includes: instructions for system communication using Linux and Windows OS; Modbus registers definition and operation.

## About Terabee Stock Level Monitoring System

The system consists of a central computing unit (Stock Level Supervisor), sensors (Stock Level TOF Sensors) and a free web-based configuration software for system setup. The supervisor unit collects and handles all distance measurements from configured sensors, and performs computation procedures to output material volume or level estimations via Ethernet. The system setup is performed in 3 simple steps:

1. Connect to the Supervisor via Wi-Fi or Ethernet
2. Navigate in the browser to stock level monitoring configurator web application
3. Input application parameters (e.g physical container characteristics) using the web-based configurator tool.



## Data triggering from sensors

After the Supervisor is configured, the system is ready for operation and allows the user to communicate with it. The Supervisor will start data collection (trigger) process, which includes the following steps:

- Activation of the connected sensors for distance measurements.
- Acquisition of the raw distance data from sensors via RS485
- Computation of volume and level data
- Storage of the computed information



*The whole data collection process can take several seconds. Depending on environmental conditions (ex. ambient light, distance to target), each configured sensor can take from 2 to 6 seconds to perform a measurement.*

The Supervisor offers two types of trigger processes: Automatic and Manual. Automatic trigger is performed periodically at a rate of 1 measurement per minute with no direct user control. Automatic triggers cannot be deactivated. Manual trigger is initiated only by the user by sending a Modbus TCP/IP request to the Supervisor unit.

## Stock Level Supervisor data definition

The following section presents the data information (possible data output) stored in the Supervisor unit, and their corresponding definitions.

**Distance 1 (Raw distance):** Distance between the sensor lens and the measured material surface or first target encountered in the sensor field of view, measured directly by the Stock Level TOF Sensor. The value is expressed in millimeters

**Distance 2 (Transformed level):** Distance between the lowest point of the container (selected during the configuration phase), and the target material detected by the sensor. This data output corresponds to the difference between the entire container height, and the distance between sensor and material. The value is expressed in millimeters.

**Level:** Ratio between the average of individual levels (transformed level) measured by the sensors and the entire height of the container, given in percentage.

**Volume:** Estimation of the current material volume, estimated from the container shape introduced during the configuration step and the current level.

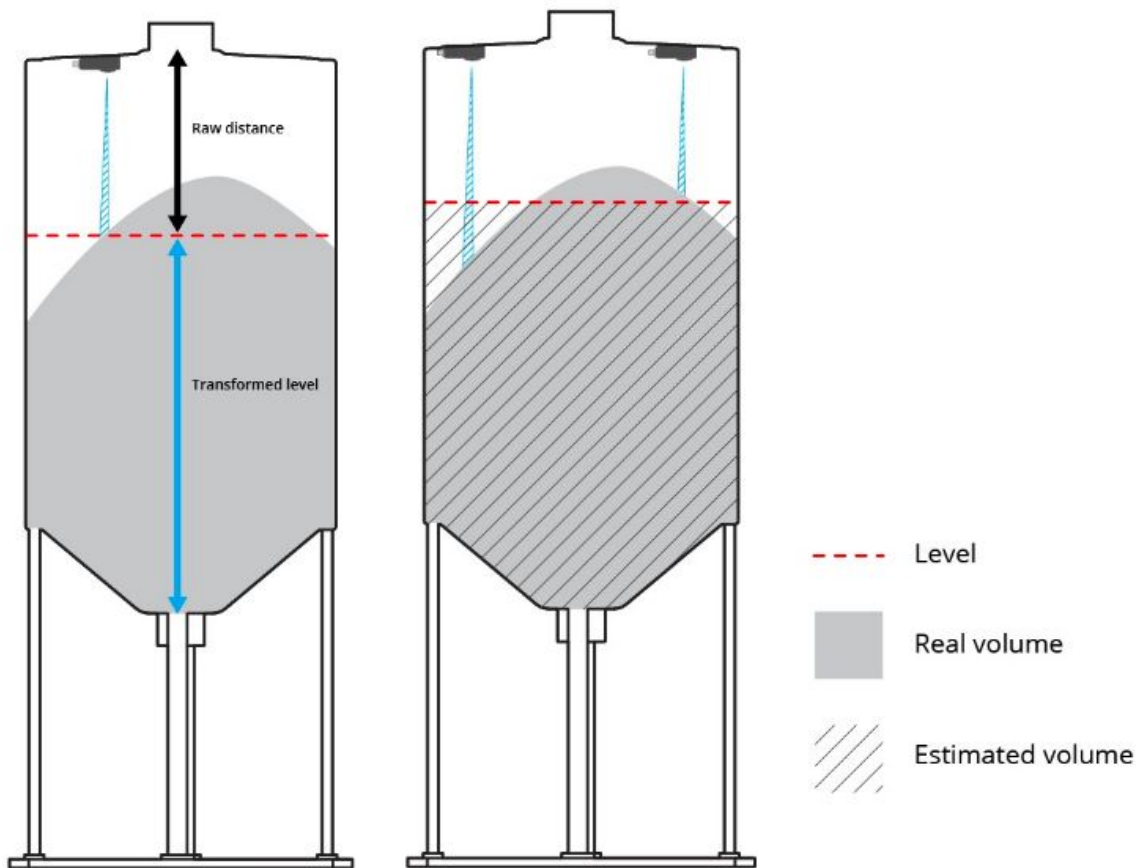


Figure 1 - Definition of the measurements made by the Stock Level Supervisor.

## Wi-Fi ID of the Stock Level Supervisor

Attached to the back of the Supervisor unit, you will find a square sticker containing its Wi-Fi ID (SSID), Wi-Fi Password and Country code, for further system configuration.




Figure 2 - Stickers on the back of the Supervisor unit

The **Wi-Fi ID (SSID)** is essential during the system configuration step as it is used (1) to connect to the Access Point of the Supervisor unit, (2) to open the onboard web-application software and (3) to resolve the IP address of its Ethernet interface inside a network.

The **Wi-Fi Password** is used to enable connection to the Supervisor's Access Point. Please note that the Wifi Password can not be changed by the user.

The **Country Code** is used to enable wireless networking. The usage of radio spectrum for wireless networks is regulated by governments to avoid interference and to guarantee a cooperative and free use of it. This code will be factory-set and indicates the country where the device can be used and the available channel.

 *In case the label has been damaged or worn off, the Wi-fi ID, Password and Country Code information can be obtained by contacting the Terabee team*

## 1. Connect with Stock Level Supervisor

### 1.1. Using embedded access point (Wi-Fi)

Once the Stock Level Supervisor is powered, it will act as an access point, allowing any wireless device with Wi-Fi to connect and communicate with it. To do so, navigate to your network settings and search for the network with the Wi-Fi ID of your device. Remember that the Wi-Fi ID is written on the sticker on the back of the case (Figure 2).

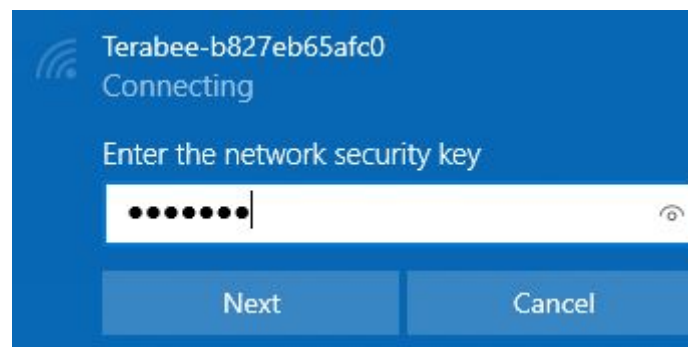


Figure 3 - Terabee-b827eb65afc0 network detected in the Wi-Fi connections

Click Connect and enter the corresponding network password. The password is also located on the label attached to the Supervisor unit hardware (Figure 2).

After a successful connection, a communication link is now established with the Supervisor unit.

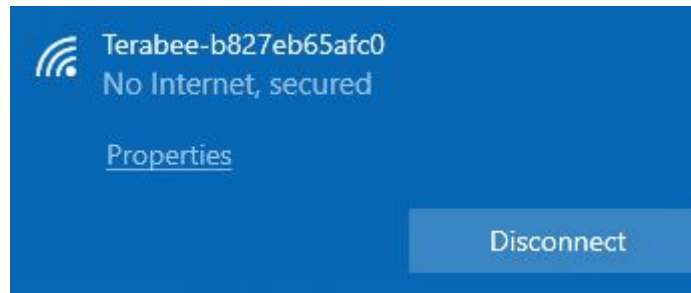


Figure 4 - Successful access point connection

## 1.2. Using Ethernet

This setup requires the use of an M12 male to RJ45 male cable to allow connectivity between the Supervisor and a PC. After a successful physical connection, please wait until the Ethernet network is recognized by your PC (this can take a moment). Once the connection is established, you will be able to see the Ethernet network in the connection settings.

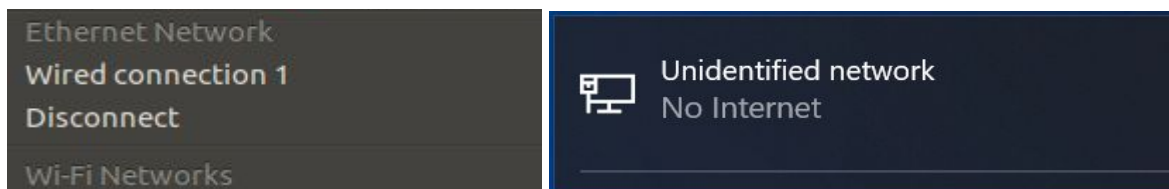


Figure 5 - Successful Ethernet network connection on Linux Ubuntu (left) and Windows OS (right).

Now that the Ethernet network connection is established, you can proceed to the Supervisor configuration step.

## 2. Configure The Stock Level Supervisor



*Please note that the configuration of the Stock Level Supervisor and associated sensors is an important and required step for the system to operate, and should be done before final physical installation.*

The Stock Level Supervisor unit offers a simple and interactive process for configuration. Once the communication link has been established (Wi-Fi or Ethernet), the user will be able to access the web application data input and parameter configuration. This is done through a graphical user interface that allows intuitive and quick setup. The following steps in this section will guide you through a full configuration process of the Terabee Stock Level Monitoring System.

Once you are connected to the Supervisor please navigate to the browser and type the Wi-Fi ID of the device followed by **.local** extension as shown in the image below. Remember that the Wi-Fi ID is written on the sticker on the back of the case (Figure 2).



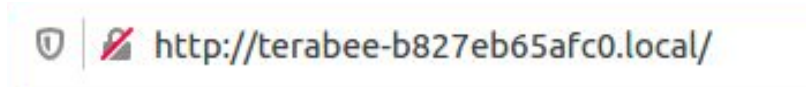


Figure 6 - Open the web configurator in a browser with the Wi-Fi ID of your supervisor unit.

By accessing the specified path, you will be taken to the web configurator of your Stock Level Supervisor.

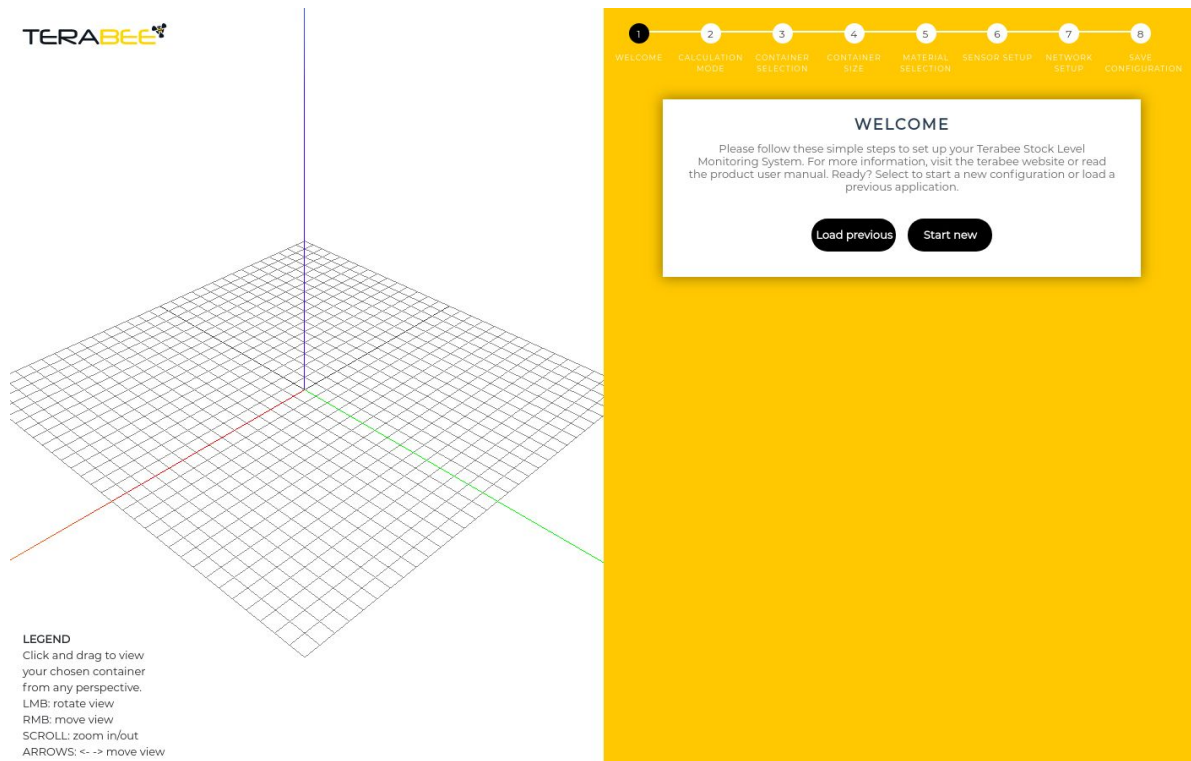


Figure 7 - Preview of the web configurator.

## 2.1. Configurator setup guide

The graphical user interface has two vertical panels. The left side displays an interactive 3D model of the container and application. You can interact with the model by using your mouse or keyboard (arrows) and the indications at the bottom left side of the interface. The right side allows application parameter input by following on-screen steps to complete the configuration process.

### Step 1 - Welcome

If you are configuring a new application, please select “Start New”. A previously saved configuration can be easily loaded by clicking “Load previous”. If there is no configuration saved, the application will inform you. The Supervisor unit only supports storage of 1 configuration at a time.

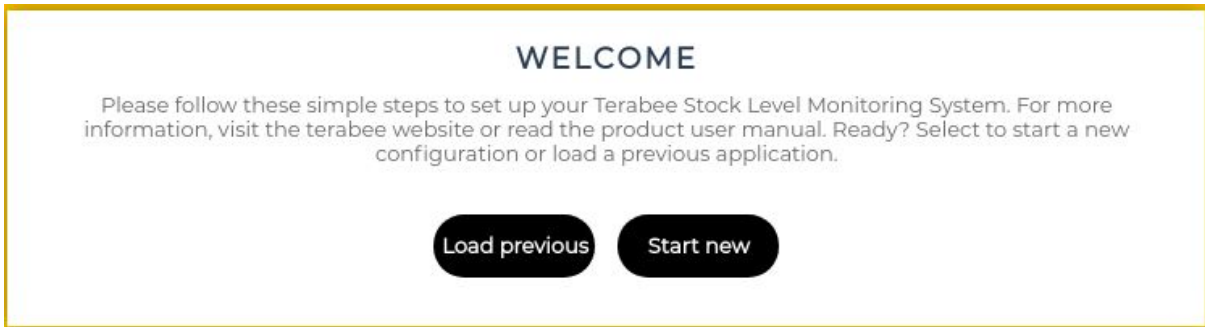


Figure 8 - Welcome window of the web configurator. Here you can start a new configuration or load a previous one.

## Step 2 - Data output type selection

In this step, please choose type of data output you are interested in:

- **Distance data** - offering raw distance data output. No additional parameters are required.
- **Volume and level data** - offering calculation of volume and level data. Container parameters and sensor positions have to be configured in the following steps.

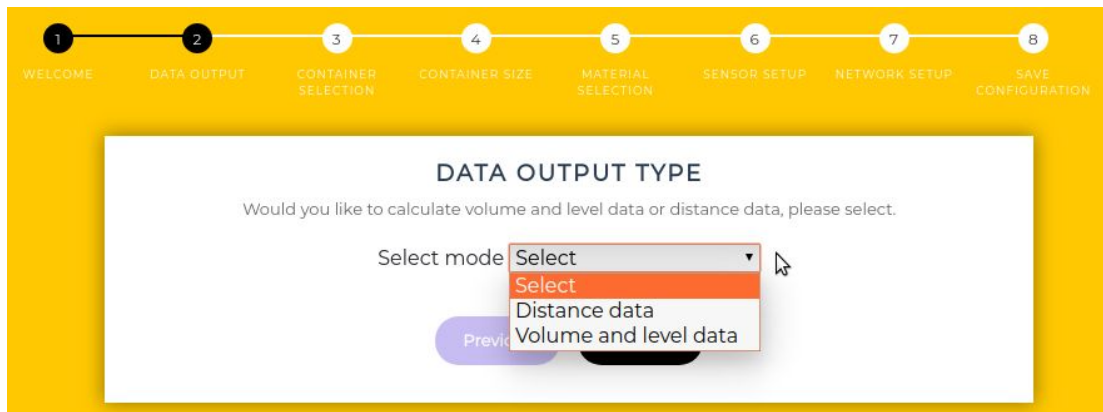


Figure 9. Output data type selection.



*Please note that if you select "Distance data" output type, no additional container parameters are required, you will be automatically advanced to step number 6.*

## Step 3 - Container selection

In this step, please choose from a pre-set list of containers corresponding to your application requirements. Two types of containers can be selected:

- **Silo** - offering a classic cylindrical container shape with the option to add coned top and bottom parts.
- **Container** - offering a rectangular container shape.

In case the pre-set container list does not match your application needs, please contact the Terabee team to explore opportunities of adding this to the software.



Figure 10 - Select the type of container you are going to use: Silo or Container (cuboid).

#### Step 4 - Container size

During this step, please input the exact dimensions of the selected container. Please make sure you have this information **before** the configuration process. Depending on the container type, the following fields are required:

- **Silo type** - Center Radius, Center Part Height, Top Part Height, Bottom Part Height
- **Container Type** - Width, Length, Height

A preview of the container 3D model is shown on the left side of the interface as the dimensions are modified

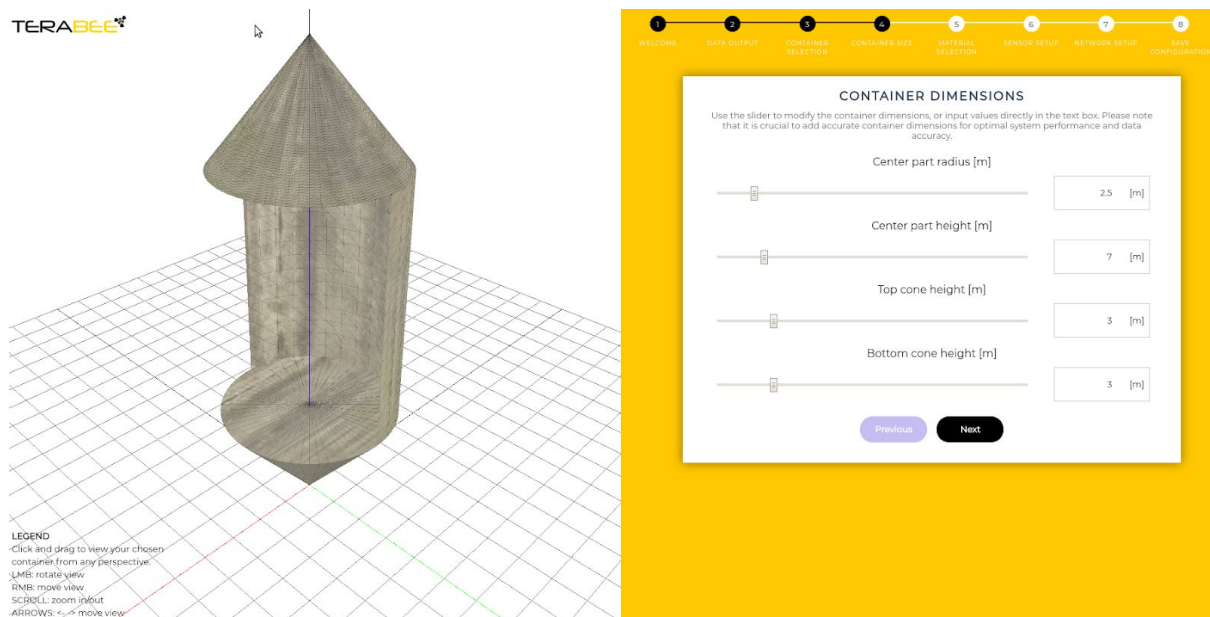


Figure 11 - Specify the characteristics of your container and visualize them in the left side panel.

## Step 5 - Material Selection

The fifth step involves selection of the material type stocked by the corresponding container. The Terabee software offers two simple selections:

- **Solids and Powders** (e.g grains, animal feed, construction materials, etc)
- **Liquids** (water, fuel, milk, other opaque liquids)

Please note that the selected materials are not visualized after their selection, nevertheless these are automatically saved and stored inside the Supervisor unit.

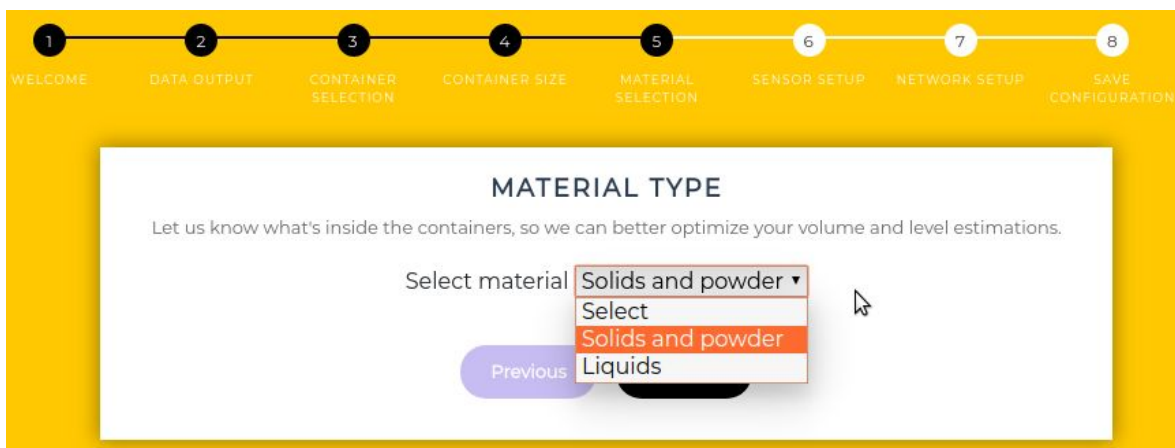


Figure 12 - Select the material type between solids and powder or liquids.

## Step 6 - Sensor Setup

Step 6 requires the user to determine suitable sensor positioning inside the selected container. As a first step, from the dropdown field, please select the number of sensors used for the level monitoring application.

Next, specify sensor position/s on the container. Input X and Y positions (measured in meters from the center of the container) using either the slide bar or the value input field. Please use technical drawings of the containers or access most suitable installation points by inspecting the container in real-life.

On the left side of the interface, the sensors will appear as black circles on the top and center of the container. By adjusting X and Y positions of each sensor on the right side, it will automatically change positions in the 3D model view to allow better visual representation.

Once all sensors are well positioned, please click "Next".



*Please note that the Terabee configuration software is only used as a means to set up the system, including volume and level computation logic. The software is not intended to provide any recommendations on most suitable mounting possibilities, necessary number or configuration of sensors, or other information related to set up of the*

application. This information is to be evaluated and confirmed by the user before system configuration



Please note, if you have selected distance data output type in step 2, only a number of sensors has to be selected. The sensor's respective position is not required.

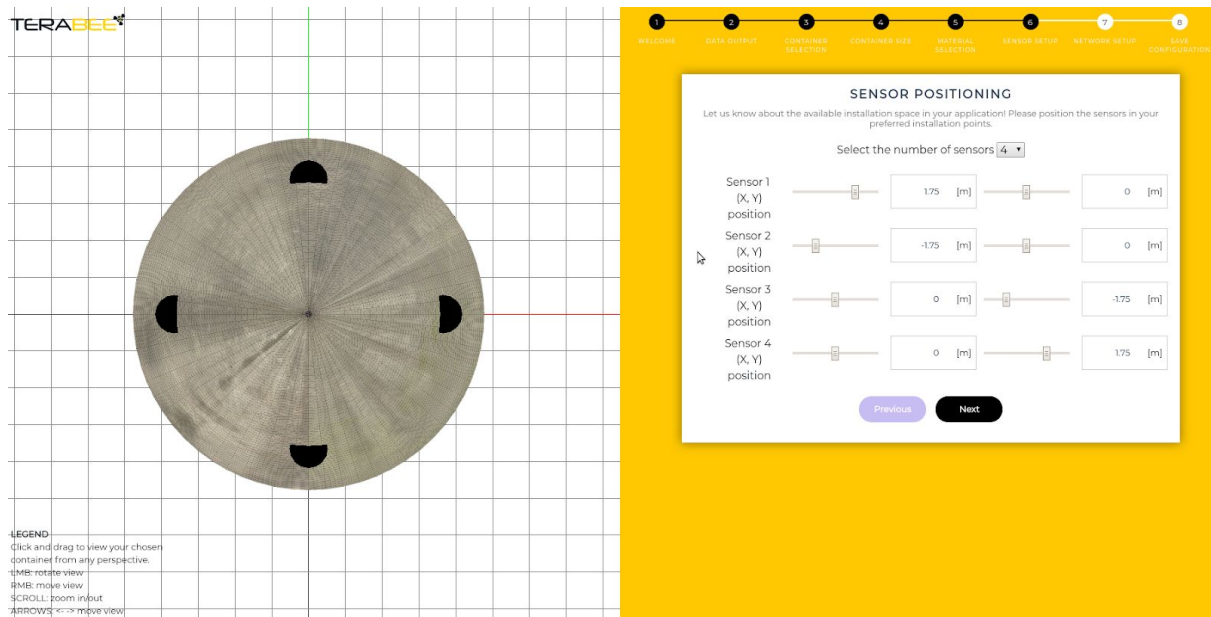


Figure 13 - Preview of the sensors position on top of your container.

### Step 7 - Network Setup

Finally, please specify a static IP address for networks which do not support DHCP (Dynamic Host Configuration Protocol) protocol, or where DHCP service is disabled.

Note that the IP address must be an IPv4 address and should be provided in CIDR (Classless Inter-Domain Routing) notation. To demonstrate an example:

- for an IPv4 address of *192.168.100.14* with a subnet mask of *255.255.255.0* the CIDR notation is *192.168.100.14/24*

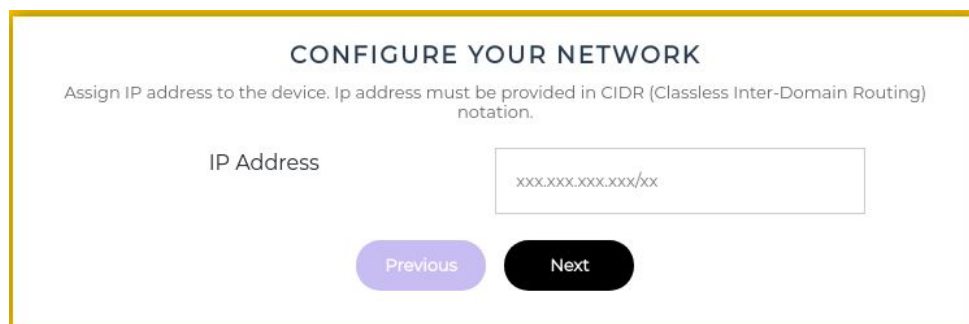




Figure 14 - Set a fixed IP in case you do not have a DHCP Server in your network.

 Please respect the format of the IP address, otherwise, you will not be able to finalize the system configuration

 If you have a DHCP Server in your network, the Stock Level Supervisor will prioritize the IP address given by the DHCP Server. If you do not have a DHCP Server or DHCP server is disabled, the static IP address will be utilized.

### Step 8 - Configuration saving

To finalize the configuration process of the Stock Level Supervisor unit, please click "Save" and your application configuration will be stored inside the Supervisor unit.

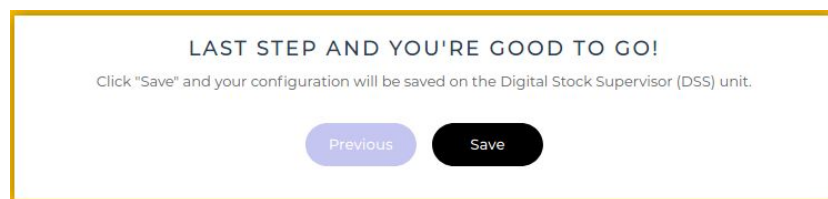


Figure 15 - Click on save to finish the configuration.

After successful configuration, the Terabee Stock Level Supervisor will start running. Connect to the device using Ethernet and start triggering or reading modbus registers.

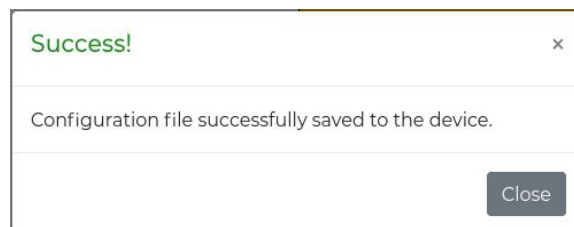



Figure 16 - If you get this message you have successfully configured your device.

 Note that the Stock Level Supervisor can reboot after the configuration if the network settings have been changed during the configuration process. The status LED will blink 10 times, following the device restart, please wait until the network is available again to access the web configurator if necessary. This can take up to 2 minutes.

### Existing configuration modification

In case some of the parameters were set incorrectly during the initial configuration process, you can access the saved configuration file and make necessary adjustments without needing to go through the full setup steps. To do so, navigate to the Homepage of the software, click on "Load previous" and modify the current configuration.

## 3. Get the IP address of the Stock Level Supervisor

To connect to the Supervisor unit from your TCP/IP client via Ethernet, its IP address is required. There are two ways of assigning an IP address to the Supervisor:

1. via DHCP (*Dynamic Host Configuration Protocol*);
2. by a fixed Ethernet IP address set during the configuration step (Section 2.1).

If you are using a DHCP server in your network, the IP address of the Ethernet interface for the Supervisor unit will be assigned automatically. However, if you do not have a DHCP server, the fixed IP address selected during the configuration step is used.



*Note that if you have a DHCP server in your network, the dynamically assigned IP address will be prioritized over the fixed IP address set during the configuration step*

### 3.1. Windows OS

#### 3.1.1. Plug the supervisor unit directly to your PC

On Windows OS, there is no DHCP server running by default. If you are planning to connect the supervisor unit directly to your PC, you will have to set up the network manually between your PC and the supervisor unit. To achieve it, please follow these steps for Windows network configuration:

1. Go to *Control Panel* → *Network and Sharing Center* → *Change adapter settings*
2. Right click on the Ethernet connection and click *Properties*
3. In the new window select *Internet Protocol Version 4 (TCP/IPv4)* and click the *Properties* button

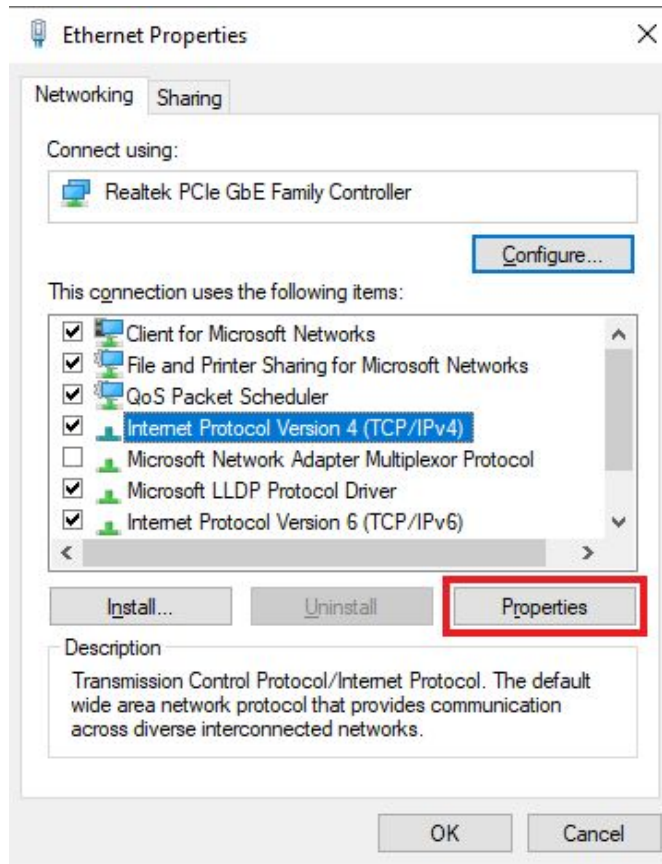


Figure 17. Ethernet connection configuration window (Windows 10).

4. You will see settings of the Ethernet interface in your computer. Select *Use the following IP address* and set IP address, Subnet mask and Default gateway (optional). Remember that the prefix of IP address and subnet mask must match the setup in the Supervisor configuration. For example, if you defined a fixed IP address during the configuration step with the following prefix 10.42.0.XXX, then in your windows PC you should define an IP address with the same prefix. In the same way, if during the configuration step you defined a subnet mask with a CIDR prefix of /24, then in this step you should write the corresponding common notation 255.255.255.0.

For example: A Stock level supervisor unit has been configured with a fixed IP 10.42.0.13/24. PC connected to this device will be configured with a fixed IP 10.42.0.2 and subnet mask 255.255.255.0. Figure below demonstrates an example of this.



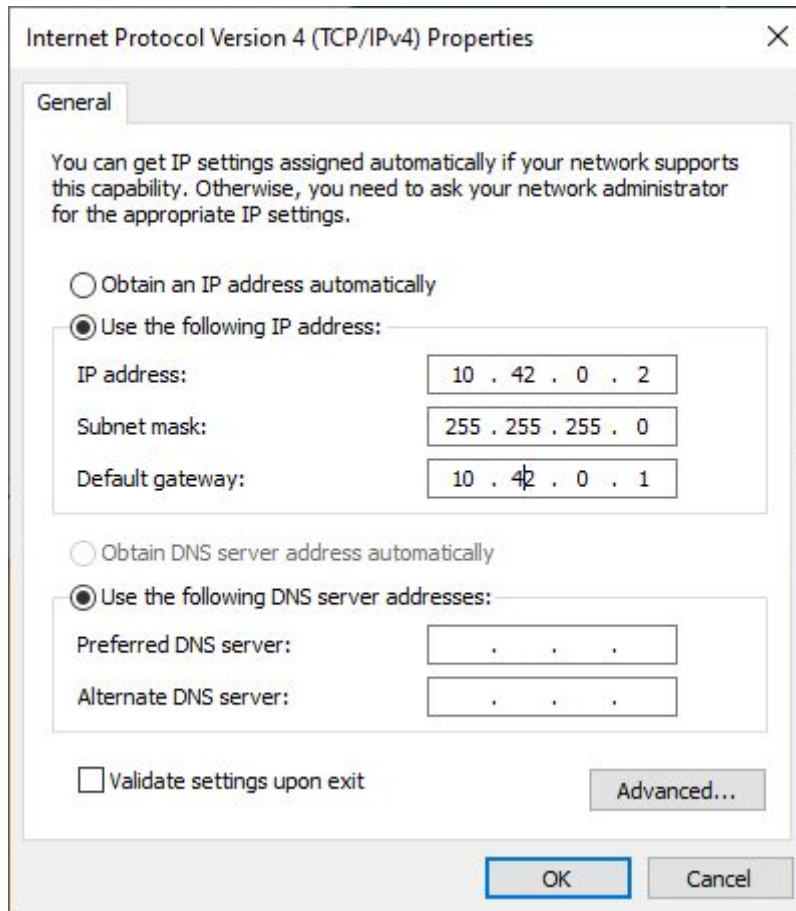


Figure 18. Internet Protocol Version 4 configuration window (Windows 10).

5. Click OK and Close.

Now, you will be able to communicate with the Supervisor unit using the defined fixed IP address in the configuration step and a Modbus TCP/IP client.

### 3.1.2. Passing through a router with DHCP server

If you are connected to a router with a DHCP server, then the IP address of the supervisor unit will be assigned automatically. In consequence, you need to do a scan in the network to recover it. Before, ensure you have the proper configuration of your network settings. Please follow these steps:

1. Go to *Control Panel* → *Network and Sharing Center* → *Change adapter settings*
2. Right click on the Ethernet connection and click *Properties*
3. In the new window uncheck the option *Internet Protocol Version 6 (TCP/IPv6)* as we want to work only with IPv4 IP address.
4. Then select *Internet Protocol Version 4 (TCP/IPv4)* and click the *Properties* button.

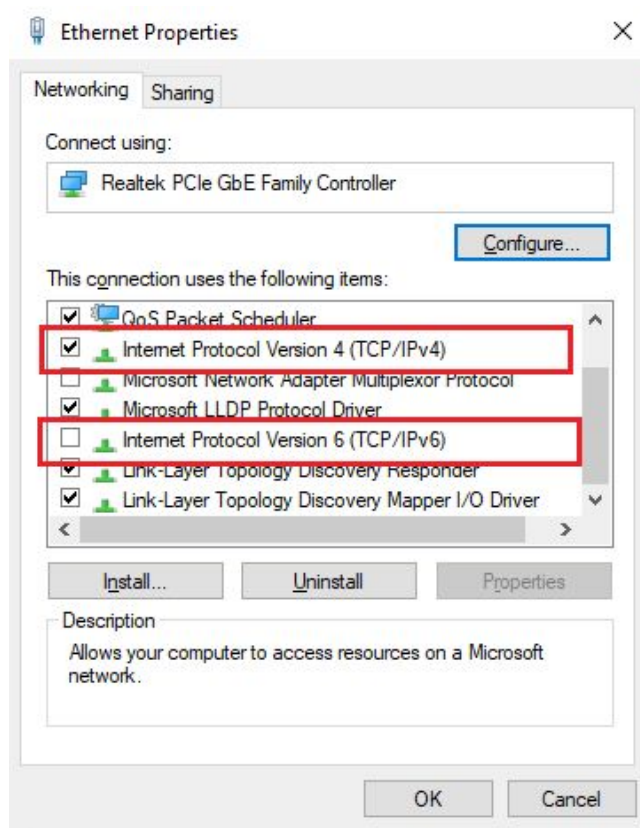


Figure 19. Ethernet properties configuration window (Windows 10).

6. You will see settings of the Ethernet interface in your computer. Select *Obtain IP address automatically*.

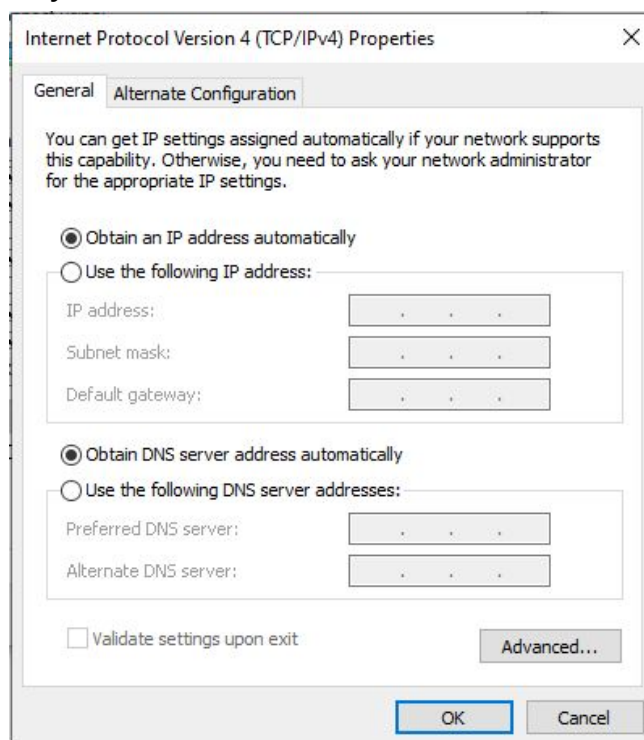


Figure 20. Internet Protocol Version 4 configuration window (Windows 10).

7. Click OK and Close.

Now, you can plug your windows PC to the network and it will get an automatic IPv4 IP address from the DHCP server. The next step is to resolve the IP address of the supervisor unit. From your PC, open up a command prompt by typing "cmd" into the start menu search (Windows Vista, 7, or newer) or by opening a Run window and then running "cmd" (All windows versions).

Type in the console:

```
ipconfig
```

This will display all the current TCP/IP network configuration of your computer.

```
Wireless LAN adapter Local Area Connection* 11:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . :
    IPv4 Address. . . . . : 192.168.0.102
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Wireless LAN adapter Wi-Fi 2:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::3550:a941:57fa:3e37%15
    IPv4 Address. . . . . : 10.0.0.133
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
```

Figure 21. Ethernet settings of your windows PC.

Then, type the next command in your terminal to obtain the supervisor's IP address:

```
ping -S <Ethernet-IP-PC> <name-of-your-device>.local
```

Where <Ethernet-IP-PC> is the IPv4 IP address found with the previous command and <name-of-your-device> is the same as Wi-Fi ID of your device. As a reminder, the Wi-Fi ID can be found on the label sticker on the back of the Supervisor unit case (Figure 2). For this example we will use: **terabee-b827eb65afc0.local**

```
C:\Users\Testing Platform>ping -S 192.168.0.102 terabee-b827eb65afc0.local
Pinging terabee-b827eb65afc0.local [192.168.0.100] from 192.168.0.102 with 32 bytes of data:
Reply from 192.168.0.100: bytes=32 time<1ms TTL=64
Reply from 192.168.0.100: bytes=32 time=1ms TTL=64
Reply from 192.168.0.100: bytes=32 time<1ms TTL=64
Reply from 192.168.0.100: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figure 22 - get the IP address of the Stock Level Supervisor Ethernet port.



*In case you experience issues discovering the Ethernet IP address of the Supervisor unit, this could be because you are connected to it through Access Point and Ethernet interface, at the same time. To ensure you get the correct IP address, please disconnect your laptop from the Access Point and leave only the Ethernet connection*

Finally, you will be able to communicate between the supervisor unit and Modbus TCP/IP client using IP discovered in this section.

## 3.2. Linux (Ubuntu 16.04)

### 3.2.1. Plug the supervisor unit directly to your PC

For direct connectivity of the Supervisor unit and a Linux OS-based PC , the network must be set up manually.

Start by executing the following command in your terminal:

```
nm-connection-editor
```

The network-connections window will appear. Next, click on the “Add” button, select “Ethernet” and click on “Create...”. Give a name to your network and open the “IPv4 Settings” tab. Once done, click on “Method” options and select “Manual”. Next, in the “Addresses” section, select “Add”.

Insert an IP address and a subnet mask. You must define these two values according to the IP address specified during the configuration step. For example, if you defined a fixed IP address during the configuration step with the following prefix 10.42.0.XXX, then in this window you should define an IP address with the same prefix. In the same way, if during the configuration step you defined a subnet mask with a CIDR prefix of /24, then in this step you should write the corresponding common notation 255.255.255.0. Click on “Save” to continue.

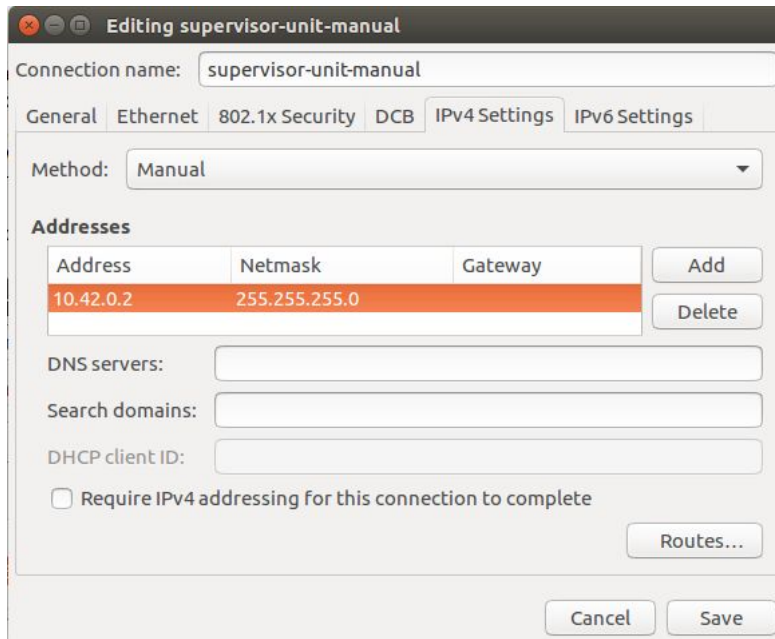



Figure 23. Ethernet connection configuration window (Linux Ubuntu 16.04).

Next please connect the Supervisor unit to Modbus TCP/IP client device (ex. PC, PLC) via Ethernet. From now on, you will be able to communicate with the Supervisor unit using the fixed IP address you defined in the configuration step and a Modbus TCP/IP client.

 During the connection step with Ethernet, please make sure you're not connected to the Supervisor unit via its Access Point

### 3.2.2. Passing through a router with DHCP server

In case of a router connection using a DHCP server, please configure and scan the network to obtain the IP address of the Supervisor as explained in the following section.

Start by executing the following command in your terminal:

```
nm-connection-editor
```

The network-connections window will appear. Next, click on the "Add" button, select "Ethernet" and click on "Create...". Give a name to your network and open the "IPv4 Settings" tab. Once done, click on "Method" options and select "Automatic (DHCP)". Click on "Save" to continue.

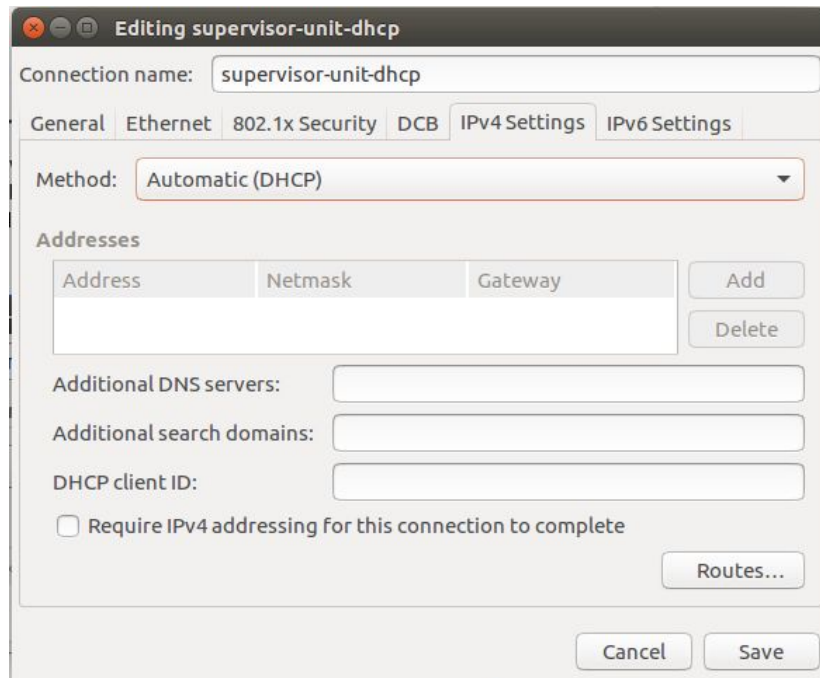


Figure 24. Ethernet connection configuration window (Linux Ubuntu 16.04).

Next, open your terminal and run the following command to obtain the Ethernet interface name of your computer:

```
ip address show
```

```
> ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp60s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 82:50:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 brd 192.168.0.255 scope global dynamic enp60s0
        valid_lft 4066sec preferred_lft 4066sec
    inet6 fe80::6fa1:289c:43f5:9e5c/64 scope link
        valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 82:50:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.198/24 brd 10.0.0.255 scope global dynamic wlo1
        valid_lft 83199sec preferred_lft 83199sec
    inet6 fe80::b3c0:c4b0:7724:1551/64 scope link
        valid_lft forever preferred_lft forever
```

Figure 25 - Find the name of the Ethernet interface of your laptop, in this case it is **enp60s0**.

Once done, execute the following command to obtain the IP address of the Supervisor unit:

```
ping -I <interface> <name-of-your-device>.local
```

Where <interface> is the name of the Ethernet interface for your computer (in case of this example **enp60s0**), and <name-of-your-device> is the same as the Wi-Fi ID of your device. As a reminder, the Wi-Fi ID can be found on the label sticker on the back of the Supervisor unit case (Figure 2). For this example we will use: **terabee-b827eb65afc0.local**

```
> ping -I enp60s0 Terabee-b827eb65afc0.local
PING Terabee-b827eb65afc0.local (192.168.0.100) from 192.168.0.101 enp60s0: 56(84) bytes of data:
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=0.582 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from 192.168.0.100: icmp_seq=3 ttl=64 time=0.425 ms
64 bytes from 192.168.0.100: icmp_seq=4 ttl=64 time=0.308 ms
^C
--- Terabee-b827eb65afc0.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 0.308/0.618/1.159/0.327 ms
```

Figure 26 - With this command you can get the Ethernet IP address of your Stock Level Supervisor.



*Please note that you might be experiencing issues with the Ethernet IP Address discovery in case you are already connected to the Supervisor unit through its Access point and Ethernet interface, at the same time. Disconnect from the access point before establishing connection with the Ethernet interface.*

You will now be able to communicate with the Supervisor unit and the Modbus TCP/IP client.

## 4. Quick Start: Communicate from a Modbus TCP/IP client

To communicate and read data from the Supervisor unit, a **Modbus TCP/IP Client** is required. The Supervisor unit offers a TCP/IP protocol for data transport and networking. This protocol communicates using a client-server relationship. To access the information stored inside the Supervisor unit, a TCP/IP client must first establish a connection with the Supervisor unit, and then send a read - or write - request. Measurement and computation results are stored in registers and coils, and can be accessed by sending corresponding Modbus frames.

There are a number of Modbus TCP/IP client softwares supporting such communication, and use will depend on the operating system of your PC. The following section covers quick start guidelines using:

- (1) **QModBus** software for **Linux OS**
- (2) **Modbus Doctor** software for **Windows OS**

For more information about the available function codes and register descriptions, please refer to section [Modbus Registers](#) of this document.

### 4.1. Quick Start with QModBus for Linux OS

QModBus software is a free Qt-based implementation of a Modbus RTU Master or Modbus TCP/IP Client. Its graphical interface will allow a quick setup and communication with the Supervisor unit. To install the software please execute the following command in your terminal:

```
sudo add-apt-repository ppa:js-reynaud/qmodbus
sudo apt-get update
sudo apt install qmodbus
qmodbus
```

#### 4.1.1. Connecting to QModBus

After successful installation, please execute command: `qmodbus` on your terminal, which will open QModBus software graphical user interface. Modbus TCP tab will start by default.

Next, select the checkbox "Active" and introduce the IP address of the Supervisor unit. Once done, please set the port to **502** and click "Apply" to establish the connection.





Currently the Stock Level Supervisor only supports a single connection, meaning that only one TCP/IP client can be connected at a time

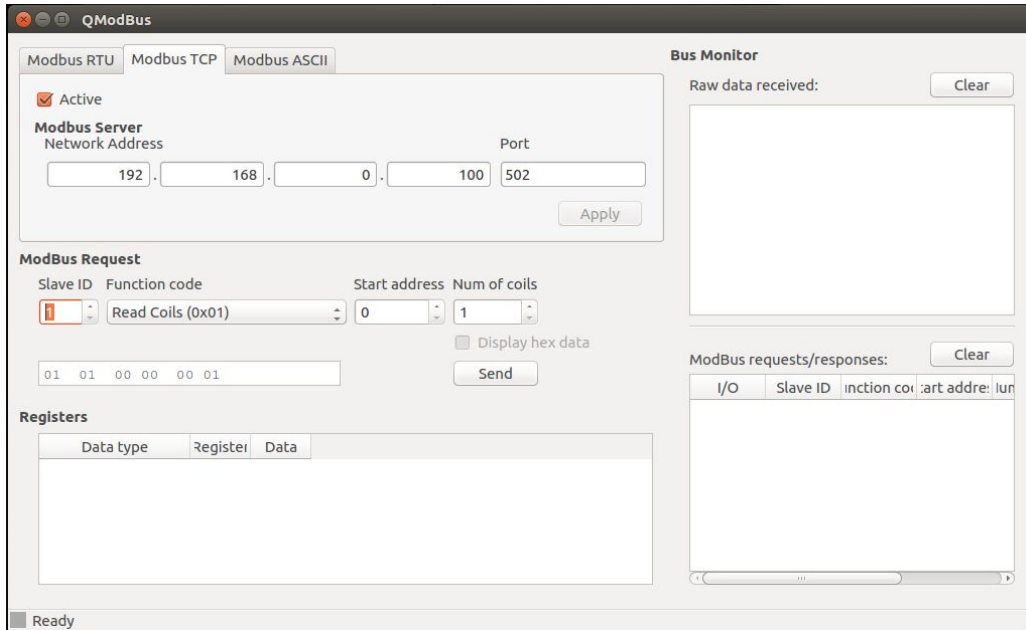


Figure 27 - QModBus software: graphical user interface window.



Please pay attention to the bottom left side of the window. If you receive the following message (Could not connect tcp/ip port!) the following scenarios might apply: (1) using an incorrect IP address, (2) introducing the wrong port, or (3) connecting before having plugged in the Ethernet cable

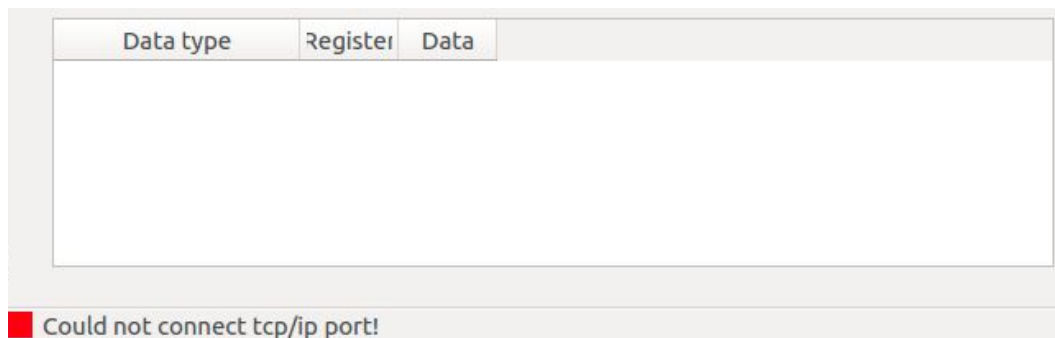


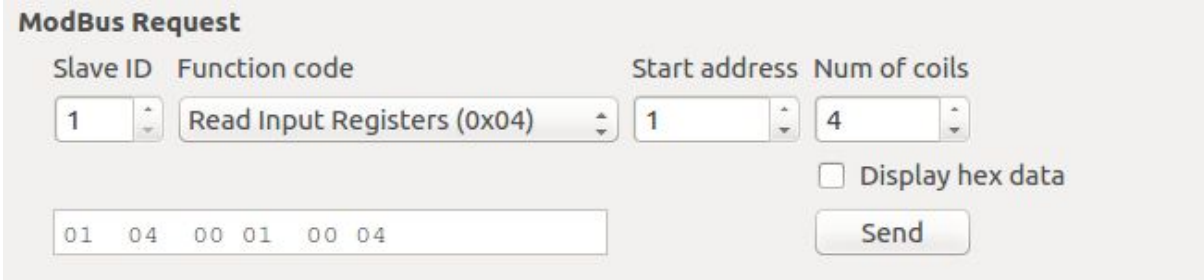
Figure 28 -QModBus software: error message example for unsuccessful connection.

If the last step succeeded, the “Apply” button in the **Modbus Server** section will be disabled. You are now connected with the Terabee Stock Level Supervisor. The **Modbus Request** section can now be used to build your request. In the next sections, you will be able to follow a few examples on reading registers and sending a *Manual Trigger* request.

## 4.1.2. Read raw distance registers example

In this section, an example on how to read raw distance registers will be presented. The Supervisor unit can report raw distance in millimeters [mm] collected from each sensor.

For the purposes of these guidelines, an example of 4 sensors connected to the Supervisor is selected. Obtaining distance measurements from the 4 sensors can be done by reading registers 1-4. In order to read (inquire) raw distance data from the Stock Level Monitoring System, please configure the next settings in the **Modbus Request** section of QModbus:



The screenshot shows the 'ModBus Request' configuration window. It has four input fields: 'Slave ID' with the value '1', 'Function code' with a dropdown menu showing 'Read Input Registers (0x04)', 'Start address' with the value '1', and 'Num of coils' with the value '4'. To the right of these fields is a checkbox labeled 'Display hex data' which is currently unchecked. Below the 'Function code' field is a 'Send' button. At the bottom of the window, a white rectangular box displays the resulting Modbus frame in hexadecimal: '01 04 00 01 00 04'.

Figure 29 - QModbus software: ModBus Request section configuration

In QModbus software, the **Modbus Request** section contains all the elements to construct a Modbus TCP/IP frame.

1. First field is the SLAVE ID. As TCP/IP is a connection-based protocol and uses an IP address to recognize the devices inside a network, it does not require a slave ID. Thus, you will not need to modify this field and any value from 1 to 254 will work.
2. The second step requires selecting the FUNCTION CODE. To see which function codes are available for the Supervisor unit please refer to section [Modbus Registers](#) of this document. For this example, the function code 0x04 (Read Input Registers) is used.
3. The third step (value) to be set is the START ADDRESS. This corresponds to the address location of the first register requested. For this example, the address location 0x01 is set.
4. Last, select the number of registers to be read under the field NUM OF COILS. For this example, four registers area read

You will observe that a short frame is created at the bottom of the ModBus Request section (inside the white rectangle). This contains the Slave ID and the Protocol Data Unit (PDU) of the frame, which includes one byte for the function code, two bytes for the starting address and two bytes for the number of registers to be read. Next, click "Send" to continue.



Note that if an incorrect number of registers is selected or a non-existent start address, a Modbus exception will be received back from the Supervisor unit and the following message will be shown in QModbus.

The screenshot shows the 'Modbus Request' configuration window. The 'Slave ID' is set to 1, the 'Function code' is 'Read Input Registers (0x04)', the 'Start address' is 33, and the 'Num of coils' is 100. The 'Display hex data' checkbox is unchecked. Below the configuration fields, a hex data field contains '01 04 00 21 00 64'. A 'Send' button is visible. Below the configuration, the 'Registers' section is empty. At the bottom, a red error message reads: 'Protocol error: Slave threw exception 'Illegal data address' or function not implemented'.

Figure 30 - Qmodbus software: illegal data address error message

Click **Send** to execute the command. The distance data is now read from the system, and displayed in the **Registers** section. To get more information about the meaning of the values stored by these registers, please go to section [Modbus Registers](#) of this document.

The screenshot shows the 'Registers' section of the software. It contains a table with the following data:

| Data type               | Register | Data  |
|-------------------------|----------|-------|
| Input Register (16 bit) | 1        | 34666 |
| Input Register (16 bit) | 2        | 34836 |
| Input Register (16 bit) | 3        | 39276 |
| Input Register (16 bit) | 4        | 34566 |

Below the table, the status 'Ready' is displayed.

Figure 31 - QModBus software: example of raw distance data, registers 1-4

The **Registers** section contains a table specifying the values of the requested registers. The first column indicates the type of register and the size of its value, the second column the address location of each register, and the third column displays the value inside the register.

Values can also be displayed in hexadecimal format, if the DISPLAY HEX DATA checkbox (in **Modbus Request** section) is selected before reading registers.



Please note that in the **ModBus requests/responses** section you will see an incoherence in the received CRC value marked in red. This occurs because QModBus looks for a ModBus CRC check at the end of the frame. Contrary to RTU, the Modbus PDU wrapped by the TCP/IP layers does not contain any CRC however QModBus software does not implement this difference

| ModBus requests/responses: |         |          |               |               |             |             | Clear |
|----------------------------|---------|----------|---------------|---------------|-------------|-------------|-------|
|                            | I/O     | Slave ID | Function code | Start address | Num of coil | CRC         |       |
| 1                          | Req >>  | 1        | 4             | 33            | 4           | 0000        |       |
| 2                          | << Resp | 1        | 4             | 33            | 2           | 0028 (0064) |       |

Figure 32 -QModbus software: As TCP/IP frames do not have CRC in their PDU, QModBus will show an incoherence in the ModBus requests/responses section. This behavior is normal when using QModbus.

### 4.1.3. Read volume data example

In this section, an example of reading Volume registers is presented. The Supervisor unit can output estimated volume data in [dm<sup>3</sup>] (liters), and it can store values from 0 to 4,294,967,295 dm<sup>3</sup>, which represents a word of 32 bits. Thus, the Supervisor unit utilises 2 registers to save the current material volume of the container. In order to read (inquire) the Volume data, please follow below configuration input parameters in the **Modbus Request** section of QModbus software.

**ModBus Request**

|                                |  |                                 |                                |
|--------------------------------|--|---------------------------------|--------------------------------|
| Slave ID                       | Function code  | Start address                   | Num of coils                   |
| <input type="text" value="1"/> | <input type="text" value="Read Input Registers (0x04)"/> | <input type="text" value="34"/> | <input type="text" value="2"/> |

Display hex data

Figure 33. QModBus software: example of reading volume data.

Once all necessary parameters are selected and input, click **Send** to execute the command. The volume data is now read from the system, and displayed in the **Registers** field.

| Registers              |          |        |
|------------------------|----------|--------|
| Data type              | Register | Data   |
| Input Register (16 ... | 34       | 0x01cf |
| Input Register (16 ... | 35       | 0xa368 |

Figure 34. Qmodbus software: volume data example (registers section)

In this example, volume estimation has been obtained using 4 configured sensors. Output values are respectively **0x01CF**, for register 34, and **0xA368**, for register 35. The volume is a combination of these words, the Most Significant Word is the one stored in register 34 and the Less Significant Word is the one stored in register 35. Thus, the current material volume of the container for this example is 0x01CFA368, which converted into decimal format is **30,385,000 dm<sup>3</sup>**.

#### 4.1.4. Read level data example

For this example we are going to read the Level register. The Supervisor unit can also output remaining level data of the monitored material, expressed in percentage [%]. In order to read (inquire) the level data from Supervisor unit, please follow below configuration input parameters in the **Modbus Request** section of QModbus software.

The screenshot shows the 'Modbus Request' configuration window. The 'Slave ID' is set to 1, the 'Function code' is 'Read Input Registers (0x04)', the 'Start address' is 36, and the 'Num of coils' is 1. The 'Display hex data' checkbox is unchecked. A 'Poll' button is visible. Below the fields, a hex data field contains the value '01 04 00 24 00 01'.

Figure 35. QModbus software: example of reading level data.

Once all necessary parameters are selected an input, click **Send** to execute the command. The level data is now read from the system, and displayed in the **Registers** section.

| Data type               | Register | Data |
|-------------------------|----------|------|
| Input Register (16 bit) | 36       | 47   |

Figure 36. Qmodbus software: level data example (registers section)

For this example, the obtained level is 47%.

#### 4.1.5. Manual trigger example

The Supervisor unit allows manual trigger of the data measurements through Modbus TCP/IP. This function is helpful in cases when the measurement update rate of your application is required to be other than once per minute (default). However, please note that the Manual Trigger function will be performed only after user request (unlike the Automatic Trigger Process). The following steps show an example of executing a manual Trigger:

1. Ensure you are connected to the supervisor unit. To connect, enter the IP address discovered in the section 3 of this manual.
2. Click Apply to connect.
3. Select the function code write single coil (0x05) and the start address 0.
4. Write **1** in the "Data" cell at the bottom section.
5. Click on "Send"

The screenshot shows the QModBus software interface with the following configuration:

- Modbus Server:** Network Address is 10.42.0.13 (highlighted with a red box and '1'). Port is 502. The 'Apply' button is highlighted with a red box and '2'.
- Modbus Request:** Slave ID is 1. Function code is 'Write Single Coil (0x05)' (highlighted with a red box and '3'). Start address is 0 (highlighted with a red box and '3'). Num of coils is 1. The 'Send' button is highlighted with a red box and '5'. The hex data field shows '01 05 00 00'.
- Registers:** A table with columns 'Data type', 'Register', and 'Data'. The first row shows 'Coil (binary)', '0', and '1' (highlighted with a red box and '4').

The status bar at the bottom left shows 'Ready'.

Figure 37 - Sending a *Manual Trigger* request.

**Protocol error: Slave threw exception 'Connection timed out' or function not implemented.**

Figure 38 - This error message can possibly appear at the bottom left side of the QModBus window if the *Manual Trigger* has taken more time than the internal timeout of QModBus. This message does not mean that the *Manual Trigger* has not been performed successfully.

To verify if the measurement has been done properly, please take a look at the section [What happens when a measurement is triggered successfully?](#)

In case of the following error message receipt: protocol error (see figure below for detailed example), please reevaluate the physical connection between the Supervisor unit and your PC. Repeat connection in the Modbus Server section of the QModBus software.

■ Protocol error: Slave threw exception 'Broken pipe' or function not implemented.

Figure 39 - Error message on the QModBus window if the connection is broken



The QModBus TCP/IP client is configured to work with a ModBus RTU timeout timer. This means that QModBus will always wait for an immediate response coming from a TCP/IP server. However, the Terabee Stock Level Supervisor may take some seconds to perform the Manual Trigger and thus the reply to this action can be delayed. This depends on the number of sensors used for the application and the state of every sensor. You may then receive this message on the bottom of the QModBus window after a Manual Trigger. To verify if the measurement has been done properly, please take a look at the section [What happens when a measurement is triggered successfully?](#)

■ Protocol error: Slave threw exception 'Connection timed out' or function not implemented.

Figure 40 - This error message can possibly appear at the bottom left side of the QModBus window if the *Manual Trigger* has taken more time than the internal timeout of QModBus. This message does not mean that the *Manual Trigger* has not been performed successfully

## 4.2. Quick Start with Modbus Doctor on Windows

The Terabee Supervisor unit uses a Modbus TCP/IP protocol for data communication via Ethernet interface. For Windows OS based operations, Terabee recommends using the Modbus Doctor freeware utility software (KScada). Please follow the link below to the official page of KScada and download the MODBUS Doctor software.

<http://www.kscada.com/modbusdoctor.html>

### 4.2.1. Connecting to Modbus Doctor

After successful download, initiate the software on your Windows PC. Next, select the TCP/IP field and make sure the following settings are configured accordingly:

- Address IP: **192.168.0.100** ([Get the IP address of the Stock Level Supervisor](#))
- NumPort: **502**
- TimeOut: **10000**

For visual instructions please consult Figure 41. Once the TCP/IP field parameters are configured, click "Connection". Connection with the Supervisor unit should now be established and the Status field (bottom left window) should display "Connected".

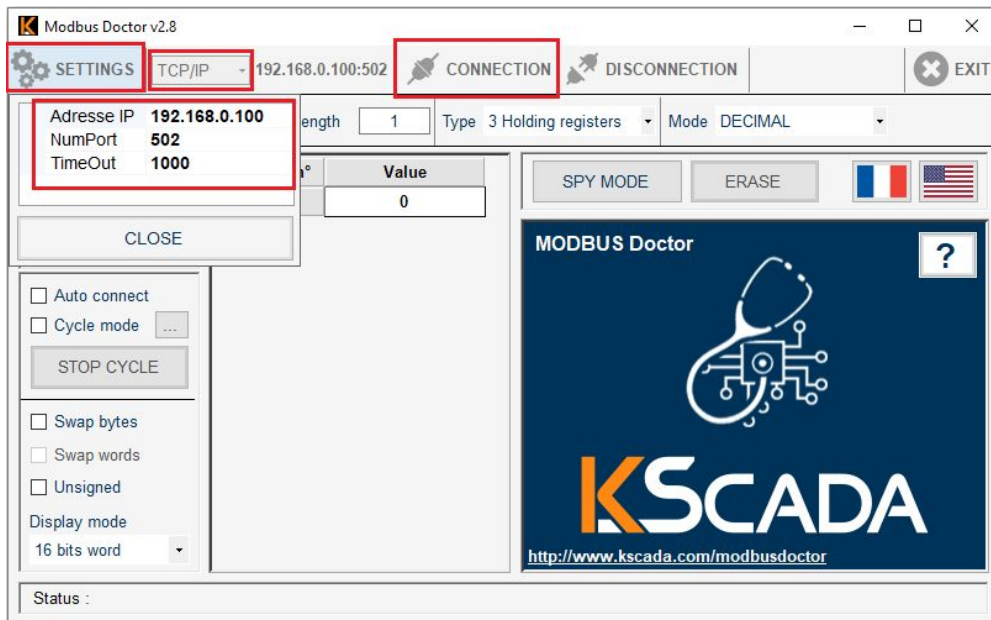


Figure 41 - Setting connection parameters.

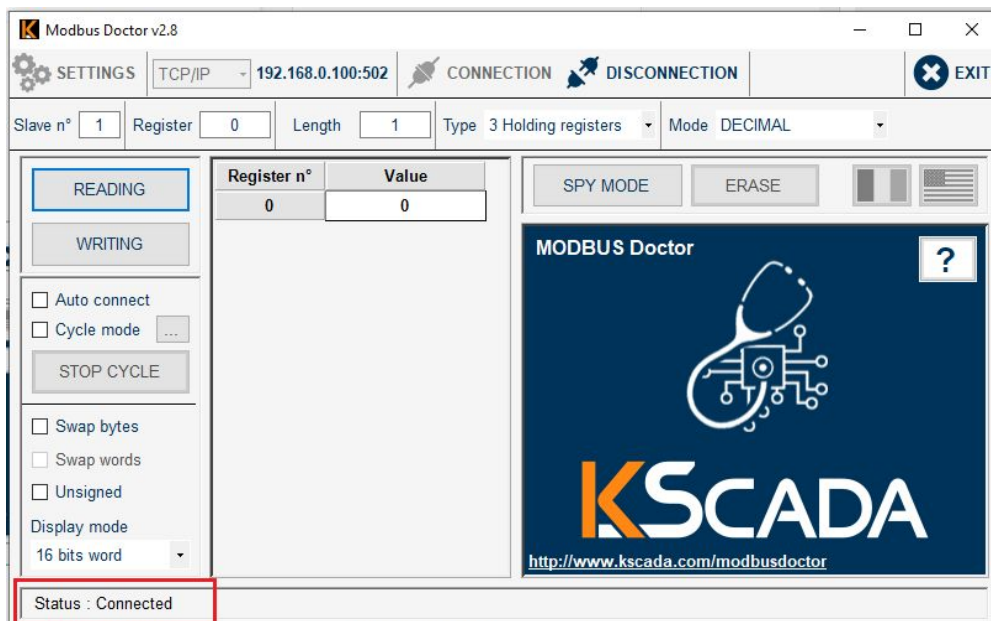


Figure 42 - Connection successfully established.

#### 4.2.2. Read raw sensor data example

The Supervisor unit can report raw distance in millimeters [mm]. For the purpose of this guideline, an example of 4 sensors is selected. Obtaining distance measurements from the 4 sensors can be done by reading registers 1-4. In order to read (inquire) raw distance data from the Stock Level Monitoring System, please execute the following READ command:



- Register (First sensor ID): 1
- Type (function): 04 Input Register
- Length (Quantity): 4

Next, click “READING” to execute the command. The distance data is now read from the system, and displayed in the “Value” field.

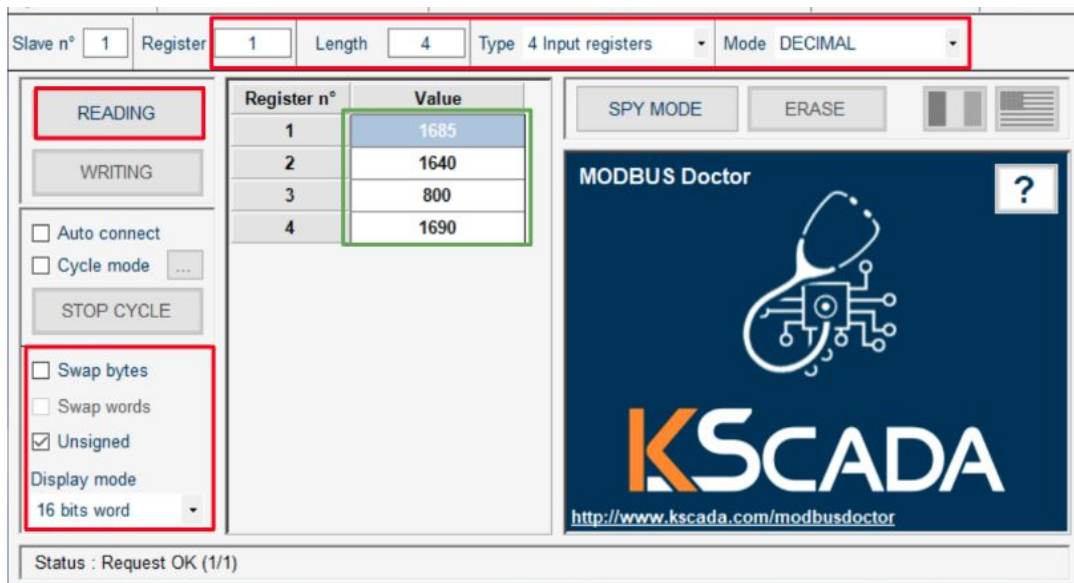


Figure 43 - Reading raw distance data.

To display in correct format, please make sure to set the following fields with corresponding parameters::

- **Mode:** DECIMAL
- **Swap bytes:** OFF
- **Swap words:** OFF
- **Unsigned:** ON
- **Display mode:** 16 bits word

### 4.2.3. Read volume data example

The Terabee Stock level Supervisor can report material volume in [dm<sup>3</sup>] (Liters). It can store values from 0 to 4,294,967,295 dm<sup>3</sup>, which represents a word of 32 bits. Thus, the Stock Level Supervisor utilises 2 registers to save the current material volume of the container. In order to read (inquire) material volume from the Stock Level Monitoring System, please execute the following READ command:

- Register (Volume): 34
- Type (function): 04 Input Register
- Length (Quantity): 2

Click "READING" to execute the command. The volume data is now read from the system, and displayed in the "Value" field.

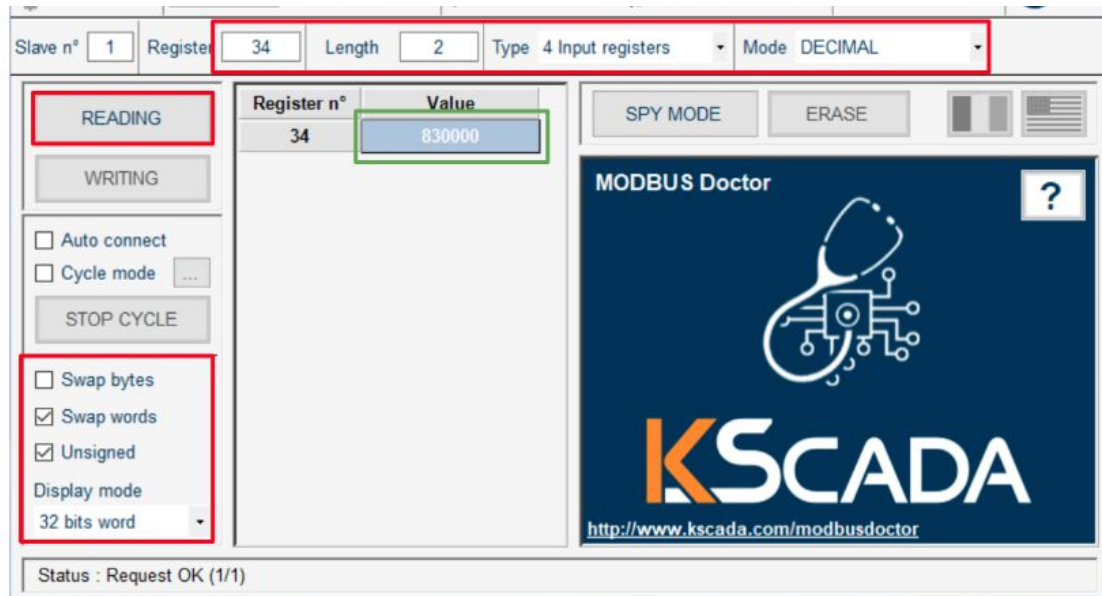


Figure 44 - Reading material volume data.

To display in correct format, please make sure:

- Mode: DECIMAL
- Swap bytes: OFF
- Swap words: ON
- Unsigned: ON
- Display mode: 32 bits word

#### 4.2.4. Read level data example

The Terabee Stock level Supervisor can report material level [0-100 %]. In order to read (inquire) material level from the Stock Level Monitoring System, please execute the following READ command:

- Register (Level): 36
- Type (function): 04 Input Register
- Length (Quantity): 1

Click "READING" to execute the command. The level data is now read from the system, and displayed in the "Value" field.

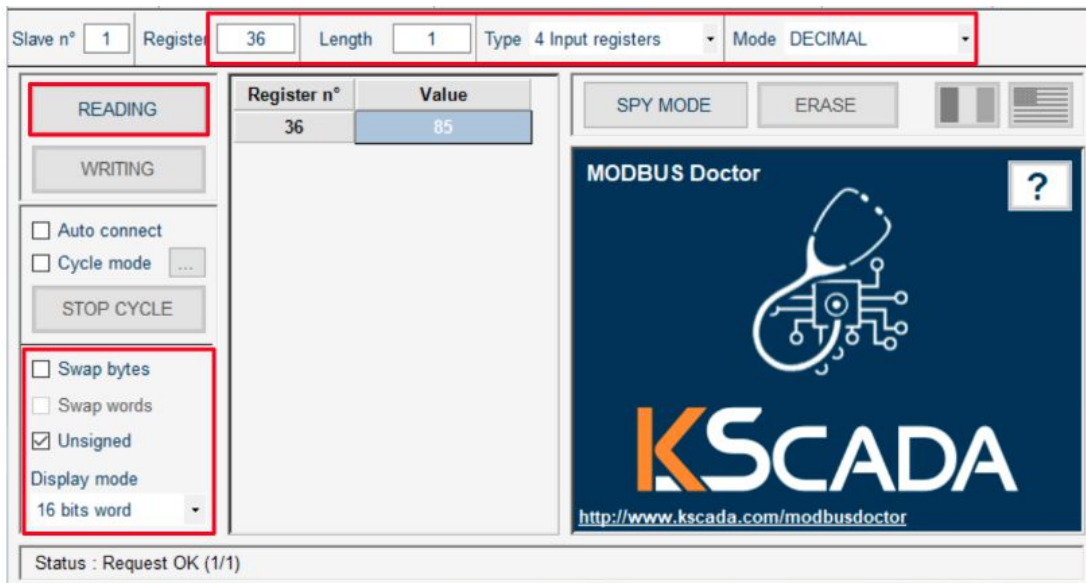


Figure 45 - Reading material level data.

To display in correct format, please make sure:

- Mode: DECIMAL
- Swap bytes: OFF
- Swap words: OFF
- Unsigned: ON
- Display mode: 16 bits word

#### 4.2.5. Manual trigger example

The Supervisor unit allows manual trigger of the data measurements through Modbus TCP/IP. This function is helpful in cases when the measurement update rate of your application is required to be other than once per minute (default). However, please note that the Manual Trigger function will be performed only after user request (unlike the Automatic Trigger Process). The following steps show an example of executing a manual Trigger:


1. Connect to ModbusDoctor as explained in the section [Connecting to Modbus Doctor](#)
2. Select type **"1 Coil status"**
3. Write **"1"** to Value field
4. Click **WRITING**



Figure 46 - Sending a *Manual Trigger* request.

## 5. Stock Level Supervisor Operation

After the configuration step, the Terabee Stock Level Supervisor will be ready to start measurements, volume and level monitoring. There are two possible measurement procedures carried out by the Supervisor unit: (1) *Automatic Trigger* and (2) *Manual Trigger*. The Automatic trigger (default) will start immediately after the configuration is done, and the triggering process is performed once every minute, updating the internal registers of the Supervisor unit.

 Please note that in the case when no sensors are connected to the Supervisor unit after configuration is completed, a new error message will be stored in the Error register. However, the Supervisor unit will not stop its normal operation. For More information about possible errors please refer to section [Modbus Registers](#).

If there are no sensors connected to the Stock Level Supervisor after the configuration step, you may then see the New Error flag raised and a new error value inside the Error Register the first time you connect to the Stock Level Supervisor after the configuration step. This is normal behavior and does not imply that the system has an actual failure. After connecting the sensors, you can either wait a minute or start a manual trigger and verify the New Error Flag and the Error register again.

## 5.1. What happens when a measurement is triggered successfully?

The measurement procedure consists of: (1) activation of sensors, (2) acquisition of their data, (3) computation of the volume and level and (4) finally storage of the results to the respective registers. The following registers change when a measurement is triggered successfully:

1. If the measurement procedure was started by a *Manual Trigger*, the register **Trigger Signal** (coil with address location 0) has been set to 1. After the measurement procedure has finished, it will go back to 0 to allow the user to perform another *Manual Trigger*.
2. The **Recent Measurement** flag (discrete input with address location 0) is set to 1.
3. One of the **Container Level** flags (discrete inputs with address locations from 1 to 7) is raised depending on the current level of the container.
4. The raw distance in mm of each configured sensor is stored in the corresponding **Raw Distance** register (Input registers with address locations from 1 to 32). The registers of the non-configured sensors are set to 0x00 FF.
5. The **Latest Measurement State** (Input register with address location 33) is set to 0x00 01.
6. The computed volume in  $dm^3$  (Liters) is stored in the **Volume** registers (Input register with address locations 34 and 35). The register with address location 34 will store the Most Significant Bytes (MSB) of the volume and the register with address location 35 the Less Significant Bytes (LSB).
7. The current level of the container in percentage is stored in the **Level** register (Input register with address location 36).
8. The **Sensor Failure Indicators** (Input registers with address locations 37 and 38) are set to zero.
9. The **Transformed Levels** (Input registers with address locations from 55 and 86) are set with their respective value.



To get a more detailed description of every register please refer to section [Modbus Registers](#).

## 5.2. How to detect an error during a measurement procedure?

In case of error detection during the operation of the Stock Level Monitoring System some flags and registers will change, providing enough information to track the source of the error. The following, indicate the register change process, when an error is detected during a measurement procedure.

1. If the measurement procedure was started by a *Manual Trigger*, the **Trigger Signal** coil (coil with address location 0) has been set to 1. Even if an error occurs during the measurement the TCP/IP client will receive a confirmation message from the server when trying to modify this register. This is because the confirmation message indicates that the coil was properly set to 1 and not that the measurement was performed correctly. In any case, this register will go back to 0 after the end of the measurement procedure.
2. The **New Error** Flag will be raised if a new error code is stored in the **Error** register. This flag will go back to zero once the **Error** register has been read by the client.
3. The **Error** Register will be set with the corresponding error code. To read more about the existing error codes please refer to the section [Error codes](#).
4. **Sensor Failure Indicators** will denote the sensors which failed during the measurement procedure. Each bit represents the current state of the sensor with id equal to the position of the bit.
5. Finally, to get more information about the exact cause of the error for a specific sensor, read its raw data inside the **Raw Distance** register. The possible error codes are described in the next section.

### 5.3. Error codes

The following table summarizes error cases and the corresponding codes. Terabee Stock Level Supervisor stores the error codes inside the Input Register with address 00. Once a new error is stored in this register, the New Error Flag is raised (Discrete Input 01 is set to true). Table 1 and 2 shows the meaning of every possible error code for this register.

| Error case                    | Output value | LED signalization  |
|-------------------------------|--------------|--|
| No Error                      | 0x0000       | This is the value of the register by default at the start up of the device. It indicates no error.   |
| Sensor Failure                | 0xFF01       | If one or more sensors fail, the Error Register will store this error code. In addition, the Sensor Failure Indicators will show which sensor has failed and the Distance Register corresponding to the ID of the sensor which has failed will show the reason of the error (Table 2). |
| Impossible to compute results | 0xFF02       | If all the sensors fail, the Error Register will store this error code. In consequence, the Stock Level Supervisor will not be able to compute a volume or a level. Then, both Volume registers and the Level register will be set to 0xFFFF.  |

Table 1 - Error Register Codes

| Error case            | Output value | LED signalization   |
|-----------------------|--------------|---|
| Target too close      | 0x0000       | The target is closer than the minimum range of the sensor.  |
| Target too far        | 0xFFFF       | The target is further than the maximum range of the sensor.   |
| Invalid reading       | 0x0001       | Ambient light is too high.<br>Target surface is too reflective.   |
| Communication failure | 0x0002       | There was an error with the communication between the Stock Level Supervisor and the sensor. This can be caused by e.g. a disconnected sensor, a damaged cable or a bad hookup. |
| Sensor not configured | 0x00FF       | If there is no configured sensor with the corresponding ID, the register is set to this value.  |

Table 2 - Specific Sensor Failure codes for the **Raw Distance** registers.

## 6. Modbus Registers

This section details all the registers available in the Stock Level Supervisor. For every type of register, first, a table with a short description is presented. Then, a definition of every register is specified.

### 6.1. Coils

| Name           | Functionality   | Address | Value Limits |
|----------------|---|---------|--------------|
| Trigger Signal | Indicates the Stock Level Supervisor to start a <i>Manual Trigger</i> . | 0       | 0 or 1       |

Table 3 - Available Coils

#### Trigger Signal

Setting this coil to true will start a measurement action. This implies that the Stock Level Supervisor will collect the distance measured by every configured sensor and will compute a volume and a level based on the container information received through the configurator. After this process, the Stock Level Supervisor will reply to the client. If the new measurement has been triggered correctly, this register will go back to 0 automatically. If there has been an error, this coil will also be set automatically to 0, but the New Error Flag will be raised and the Error Register will be modified.

**Address location** : 0

**Value** : True (1) or False (0)



*Please note that in any case the reply of the Stock Level Supervisor when trying to modify this register is an ACK. If the client wants to track an error, it has to read the New Error Flag. A Modbus exception will only be replied to if the request frame is wrong.*



## 6.2. Discrete Inputs (Read Only Flags)

| Name               | Address | Value | Description   |
|--------------------|---------|-------|---|
| Recent Measurement | 0       | 0/1   | Recent measurement inside the <b>Volume</b> or <b>Level</b> Registers |
| New Error          | 1       | 0/1   | New Error In the <b>Error</b> register                                |
| Empty              | 2       | 0/1   | The container level is below 5%                                       |
| Low Level          | 3       | 0/1   | The container level is below 25% and above 5%                         |
| One Quarter        | 4       | 0/1   | The container level is below 50% and above 25%                        |
| Half Full          | 5       | 0/1   | The container level is below 75% and above 50%                        |
| Three Quarters     | 6       | 0/1   | The container level is below 95% and above 75%                        |
| Full               | 7       | 0/1   | The container level is above 95%                                      |

Table 4 - Available Discrete Inputs (read only Flags)

### Recent Measurement Flag

This boolean Flag defines whether the values in any of the Input Registers 34 (Volume MSB), 35 (Volume LSB) or 36 (Level) have been read or not.

**RECENT MEASUREMENT FLAG = 1:** The values stored in the Volume or Level registers have not been read.

**RECENT MEASUREMENT FLAG = 0:** The values stored in the Volume or Level registers have been read. If a value spans more than one register (for example volume register), reading only one register will set the flag to 0.

**Address location** : 0  
**Value** : True (1) or False (0)

### New Error Flag

This Flag defines whether the value in the Input Registers 0 (Error Register) has been read or not.

**NEW ERROR FLAG = 1:** The value stored in the Error Register has not been read and is a new value. It has not been read yet by a client.

**NEW ERROR FLAG = 0:** The value stored in the Error Register has been read and is not a new value. It has been read already by a client.

**Address location** : 1  
**Value** : True (1) or False (0)

### Level Flags

The Following registers are booleans flags indicating the current level of the supervised container. When the current level is inside the range defined for every flag, it is set to true, otherwise it is set to false. In consequence, only one of those flags will be true at a time.

#### EMPTY

**Range** : Level  $\leq$  5%  
**Address location** : 2  
**Value** : True (1) or False (0)

#### LOW LEVEL

**Range** : 5% < Level < 25%  
**Address location** : 3  
**Value** : True (1) or False (0)

#### ONE\_QUARTER

**Range** : 25%  $\leq$  Level < 50%  
**Address location** : 4  
**Value** : True (1) or False (0)

#### HALF FULL

**Range** : 50%  $\leq$  Level < 75%  
**Address location** : 5  
**Value** : True (1) or False (0)

#### THREE\_QUARTERS

**Range** : 75%  $\leq$  Level < 95%  
**Address location** : 6  
**Value** : True (1) or False (0)

#### FULL

**Range** : 95%  $\leq$  Level  $\leq$  100%  
**Address location** : 7  
**Value** : True (1) or False (0)

### 6.3. Input Registers (Read Only)

The following section lists the Input Registers. Those registers contain the information acquired by the sensors and the results of the computations made by the Stock Level Supervisor. For every register, this section defines a name, a description of the contents, an address location and the limits of the stored value.

| Name                        | Address   | Description   |
|-----------------------------|-----------|---|
| Error                       | 0         | Contains the error codes  |
| Raw Distance                | 1 to 32   | Raw data measured by the sensors  |
| Latest Measurement State    | 33        | Indicates the state of the latest measurement   |
| Volume (First 16-bit word)  | 34        | Contains the Most Significant Bytes of the current Volume of the container                            |
| Volume (Second 16-bit word) | 35        | Contains the Less Significant Bytes of the current Volume of the container                            |
| Level                       | 36        | Contains the current Level of the container   |
| Sensor Failure Indicators   | 37 and 38 | Indicates which sensors have failed   |
| Container Type              | 39        | Indicates the container Type  |
| Material type               | 40        | Indicates the Material type   |
| Container Dimensions        | 41 to 44  | Contains the configured dimensions of the container being monitored                                   |
| Reserved for Future Use     | 45-47     | RFU   |
| Number of Sensors           | 48        | Contains the number of the current configured sensors   |
| Ethernet Static IP Address  | 49 and 50 | IP address of the Ethernet port assigned when there is no DHCP server in the network                  |
| Ethernet Subnet Mask        | 51        | Contains the Subnet Mask for the current IP address in CIDR notation                                  |
| Product ID                  | 52 to 54  | Contains the product ID as a Mac Address  |
| Transformed Levels          | 55 to 86  | Contains the distance between the lowest point of the container and the point measured by the sensor. |

Table 5 - Available Input registers (read only)

## Error Register

This register stores the last error code found by the Stock Level Supervisor, which indicates its general state. Please refer to the Error Codes section for more information about the error codes.

**Address location** : 0

**Value Limits** : The possible values of this register are

- 0x00 00 - Default value at start up. No errors.
- 0xFF 01 - Sensor Failure.
- 0xFF 02 - Impossible to compute the results.



To get more detailed information about the error management please refer to section [Error codes](#).

## Distance Registers

These registers store the raw distance in millimeters measured by the configured sensors. A configured sensor is a sensor defined during the configuration step. The register whose address location matches the ID of a configured sensor will store its raw distance. In other words, register with address location 1 will store the raw distance of sensor with ID 1, register with address location 2 will store the raw distance of sensor with ID 2 and so on. If a sensor ID has not been configured, the corresponding register will be set to 0x00FF.

**Address location** : 1 to 32

**Value Limits** : 500 to 60000 mm



Please note that there are some special cases for when the sensor has failed during the measurement process. Please refer to Table 2 to check the special cases.

## Latest Measurement State Register

This register contains the state of the latest measurement. If it has been performed successfully, this register will be set to 1. Once the result values have been read, this register will go back to 0. Finally, if an error was found during the measurement it is going to be set to 0xFF FF.

**Address location** : 33

**Value Limits** : The possible values of this register are

- 0x00 01 - The last measurement has been performed successfully.
- 0x00 00 - The result values have been already read by a client.
- 0x00 FF - An error occurred during the measurement.



Please note that this register exists to allow the user to read the state of the Stock Level Supervisor in an easier way. It opens the possibility to read many registers starting from its address and get a general overview at once. For example, the client can use the next frame to read together the Previous Measurement state, the Volume registers and the Level in a single command:

```
0x00 0xAA 0x00 0x00 0x00 0x06 0x01 0x04 0x00 0x21 0x00 0x04  
(Where AA is a transaction identifier in the Modbus TCP/IP protocol)
```

### Volume Registers

These registers store the volume of the container computed after the last measurement of the Stock Level Supervisor. As the volume is computed in Liters, two registers are available to store 32 bit values. The value is then stored in both registers in a big endian format. It means for a volume value of 0xAA BB CC DD, the register 34 will store the value 0xAA BB and the register 35 the value 0xCC DD. Note that the volume will be computed if at least one sensor performed a measurement successfully, even if some sensors have failed during the process. However, if all the sensors failed this register will be set to 0xFF FF

**Address location** : 34 and 35  
**Value Limits** : together obtain values from 0 to 4294967295 L.  
 0xFF FF - Value when it is impossible to compute the volume.

### Level Register

This register stores the level of the container computed after the latest measurement of the Stock Level Supervisor. It is stored as a percentage of the measured material surface to the height of the container. Note that the level will be computed if at least one sensor performed a measurement successfully, even if some sensors have failed during the process. However, if all the sensors failed this register will be set to 0xFF FF

**Address location** : 36  
**Value Limits** : 0 to 100%  
 0xFF FF - Value when it is impossible to compute the Level.

### Sensor Failure Indicator registers

Each bit stored in these registers represents the failure status of a sensor. If the bit at the position  $i$  is set to 1, the sensor with ID  $i$  has failed during the last measurement. However, there are 32 available IDs for sensors. As each register can only store a word of 16 bits, the sensor failure indicator is divided into two words. The first word, stored inside the register 37, sets the sensor failure indicator for the first 16 sensor's ID. The second word, stored in the register 38, sets the sensor failure indicator for the rest of the available IDs. Note that the smaller ID represented in the word is defined by its LSB.

**Address location** : 37 and 38  
**Value Limits** : 0x00 00 to 0xFF FF



For example let's suppose that after the last measurement the sensors with Id 1, 10 and 25 have failed. Then, the value stored in the register 37 will be 0x0201 and the value stored in the register 38 will be 0x0100.

## Current Configuration Registers

These registers show the current configuration stored in the Stock Level Supervisor. They are read only registers as they client is not intended to modify the configuration at runtime. The configuration parameters include: Container type, Material Type, Container dimensions, number of sensors and current IP address.

### Container type

Stores the code for a specific container: Silo or Cuboid.

**Address location** : 39

**Value** :

- 0x00 00 - No container (distance readings only)
- 0x00 01 - Silo.
- 0x00 02 - Cuboid.

### Material type

Stores the code for a specific material: solid or liquid.

**Address location** : 40

**Value** :

- 0x00 00 - No container (distance readings only)
- 0x00 01 - Powder and Solids.
- 0x00 02 - Liquid.

### First Container's dimension

If it is a prism it will contain the width (x in configuration step). If it is a silo it will contain the center radius. Value in mm.

**Address location** : 41

**Value** : 0 to 65535 mm

- 0x00 00 - No container (distance readings only)

### Second Container's dimension

If it is a prism it will contain the length (y in configuration step). If it is a silo it will contain the center height. Value in mm.

**Address location** : 42

**Value** : 0 to 65535 mm

- 0x00 00 - No container (distance readings only)

### Third Container's dimension

If it is a prism it will contain the height (z in configuration step). If it is a silo it will contain the upper height. Value in mm.

**Address location** : 43

**Value** : 0 to 65535 mm

0x00 00 - No container (distance readings only)

### Fourth Container's dimension

If it is a prism it will contain 0. If it is a silo it will contain the bottom height. Value in mm.

**Address location** : 44

**Value** : 0 to 65535 mm

0x00 00 - No container (distance readings only)

### Number of sensors

Contains the number of currently configured sensors.

**Address location** : 48

**Value** : 1 to 32

### Current IP address

This register stores the current IP address of the Stock Level Supervisor's ethernet port. As it is composed of 4 bytes, it is stored in two registers. The value is stored in hexadecimal format. For an IP address of 10.42.0.40, the register 49 will contain the value 0x0A 2A and the register 50 the value 0x00 28

**Address location** : 49 and 50

**Value** : 0x00 00 to 0xFF FF.

### Ethernet Subnet Mask

This register contains the subnet mask of the IP in CIDR notation. It is given in hexadecimal format. For example, for a subnet mask of 255.255.255.0 which has 24 leading 1-bits, the value of this register will be 0x00 18.

**Address location** : 51

**Value** : 0x00 00 to 0x00 20.

## Product ID Registers

The product ID is the same as device MAC address. The value is stored in hexadecimal format. For example MAC address "AA:BB:CC:DD:EE:FF" is stored in register 52 (0xAA BB), the register 53 (0xCC DD) and the register 54 (0xEE FF).


**Address location** : 52 to 54  
**Value** : 0x00 00 to 0xFF FF.

## Transformed Level Registers


These registers store a **transformation** of the raw distance measured by the configured sensors. A configured sensor is a sensor defined during the configuration step. The registers whose address location matches the ID of a configured sensor will store its transformed level. If a sensor ID has not been configured, the corresponding register will be set to 0x00FF.

The transformation corresponds to the distance between the lowest point of the container and the target detected by the sensor. In other words, these registers store the difference between the height of the container and the raw distance measured by the sensor.

**Address location** : 55 to 86  
**Value Limits** : 0 to 65535

 Please note: If the distance data mode has been selected in the step 2 of the configuration, transformed level registers are the same as raw distance registers.

## 6.4. Holding registers (Configurable Parameters)

 The Stock Level Supervisor does not have any holding register.