OilWear 2.0 (S120)

User Manual

April 2023

V1.01

Filtertechnik Ltd 1 Central Park, Lenton Lane, Nottingham, NG7 2NR

Email: sales@filtertechnik.co.uk **Web:** www.filtertechnik.co.uk

Tel: +44 (0)115 900 3600

Registered in England No: 03969985 VAT No: 760 821 731

Table of Contents

Introduction

This document describes the main features for the installation and commissioning of the OilWear 2.0 system. The main function of OilWear 2.0 is particle monitoring in industrial machinery fluids.

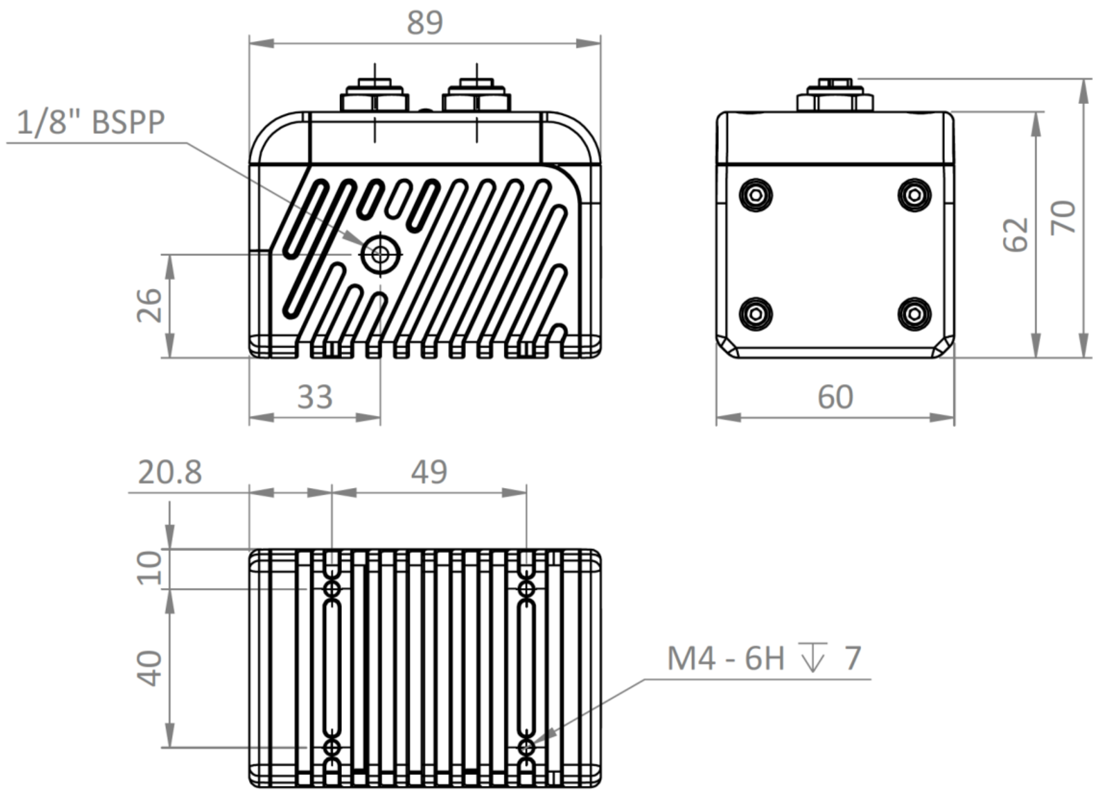
It is based on patented technology of the digital processing of images, which counts particles of more than 4µm present in fluids and are classified by size according to ISO, NAS or SAE standards, also providing a classification by shape and size to determine root cause origin.

The OilWear system is designed to be installed in a by-pass of the lubrication circuit taking a small sample from time to time and returning to the reservoir once measured.

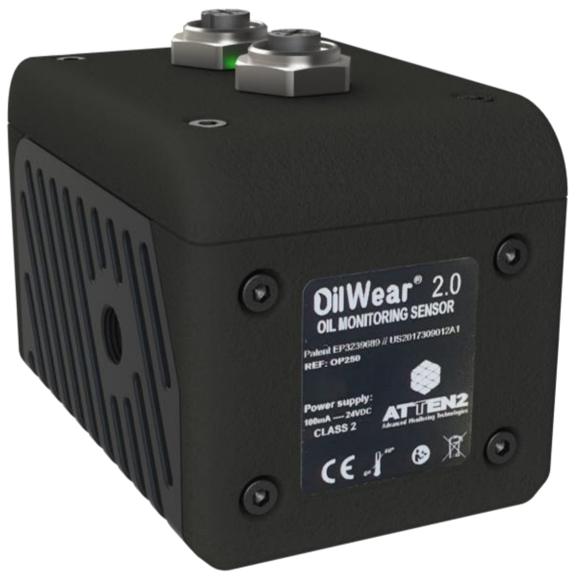
OilWear is designed to provide digital/analogue output data regarding fluid cleanliness measurement (according to the selected standard - ISO, NAS or SAE) of the oil sample, through the by-pass without interfering with a machines normal operation.

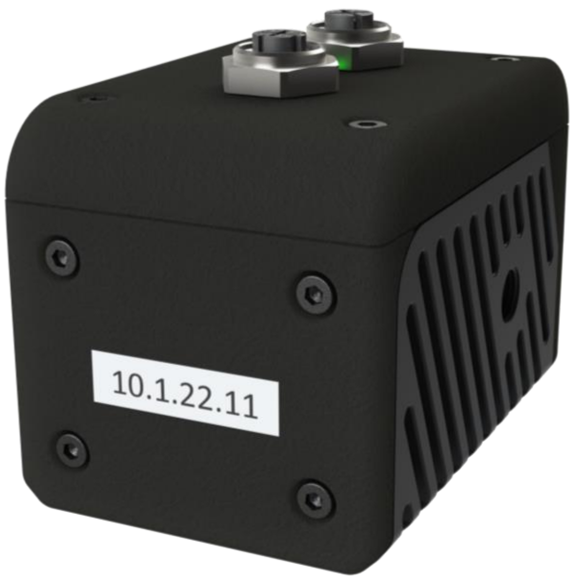
Technical Specifications

Specification	Detail
Power supply	24 VDC (20 minimum to 28 maximum)
Power consumption	100 mA (20 minimum to 150 maximum)
Operating temperature	0 to 70°C
Storage temperature	-30 to 70°C
Typical operating pressure	0.1 to 5 bar
Maximum pressure	160 bar
Operating oil flow	20 ml/min
External materials	Aluminium (anodised), Nylon
Wettable materials	BK7 optical glass, Aluminium, Viton
Ingress protection	IP65
Viscosity range (recommended)	1 - 460 cSt
Measuring standards	ISO4406, NAS1638, AS4059
ISO code range	7 – 25
Precision	±1 ISO codes (standard) ±2 ISO codes (unfavourable conditions)
Output	>4µm, >6µm, >14µm, >21µm, >38µm, >70µm Particle counts per ml & bubble counts per ml
Communication	Modbus RTU, RS485
Test time (adjustable)	60-3600 seconds (120 seconds minimum recommended)
Error log	Last 200 errors
Data log	Last 1000 measurements
Certifications	CE, UL, GL
Branding	Serial number, model, power supply, IP address
Calibration	ISO 11171
Weight	0.5 kg



Unit Identification



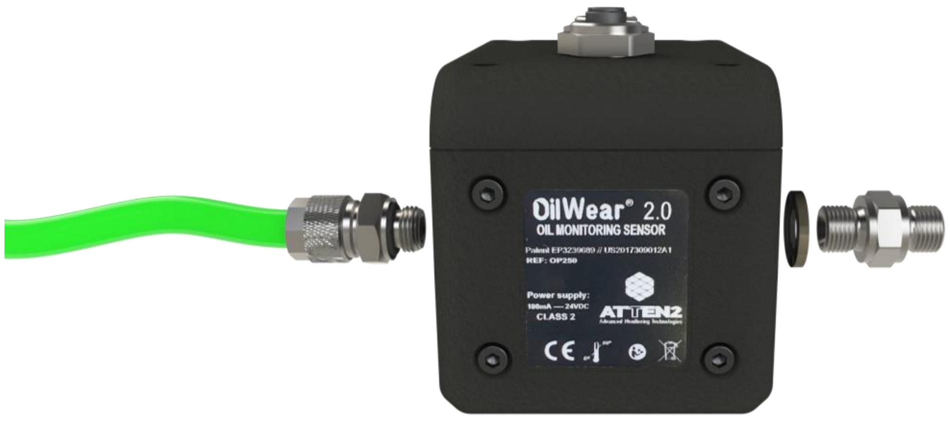


Each device is identified on the main label by a unique serial starting with OP (e.g. OP249). The other end of the device will be label with its IP address, required for connection over TCP/IP.

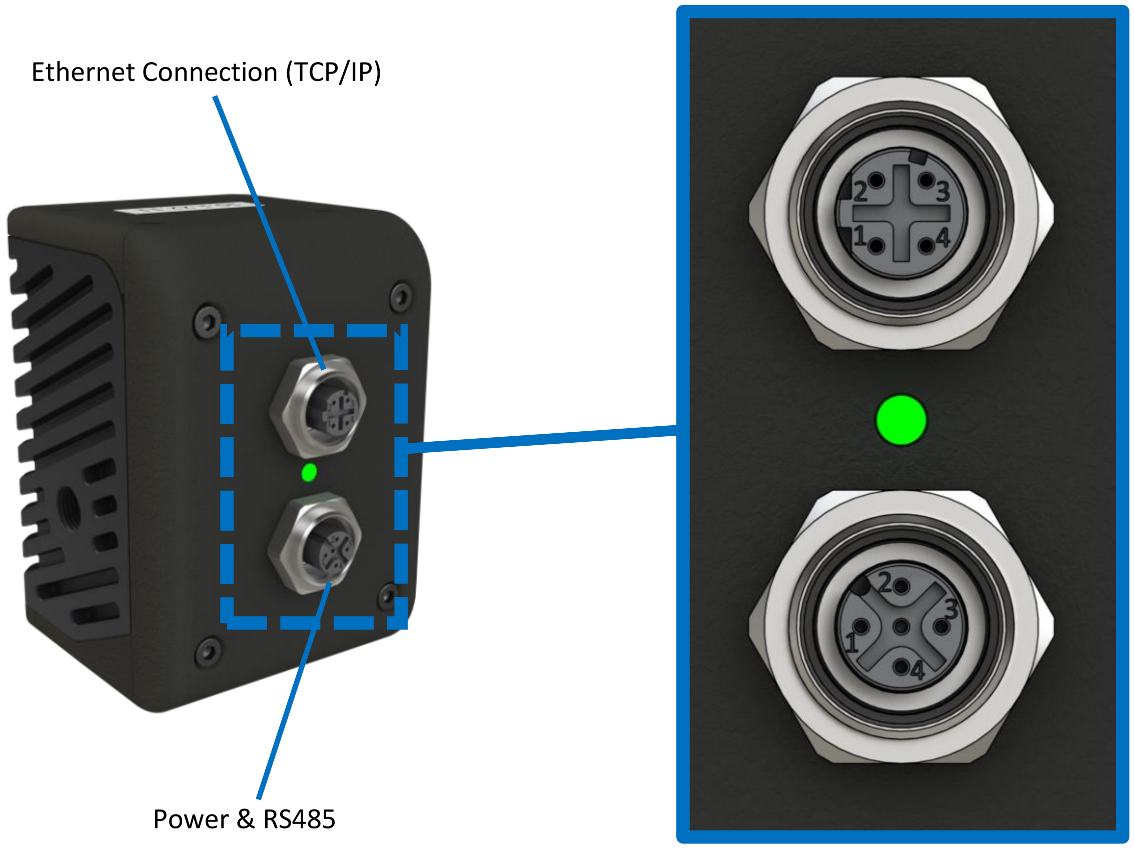
Fluid Connections & Flow Rate

The OilWear 2.0 uses 1/8" BSP parallel connections for the fluid circuit. It is advisable to use parallel threaded fittings with a dowty bonded seal or fittings which include an o-ring seal.

The units are calibrated for a flow rate of 20 ml/min. Flow rate can be regulated using a needle valve or similar flow control device, best positioned after the unit to minimise aeration of the oil.



Electrical Connections & Interfaces



Power & RS485 Wiring

Pin	Function	Colour (FT Cable)	Colour (Atten2 Cable)
1	N/A	Brown	Brown
2	Power In 24 VDC (+)	White	White
3	Power In -GND (-)	Blue	Blue
4	RS485 -RX (Receive)	Black	Grey
5	RS485 +TX (Transmit)	Green/Yellow	Black

Note: Wire colours may vary depending on the brand of cable used.

Ethernet (TCP/IP) Wiring

Pin	Function	Colour
1	+TX	White/Orange
2	+RX	White/Green
3	-TX	Orange
4	+TX	Green

Modbus TCP port: Industrial slave configuration protocol through port 502.

Powering Up

During the initial powering up allow at least 1 minute before establishing communication for the device to complete internal prechecks.

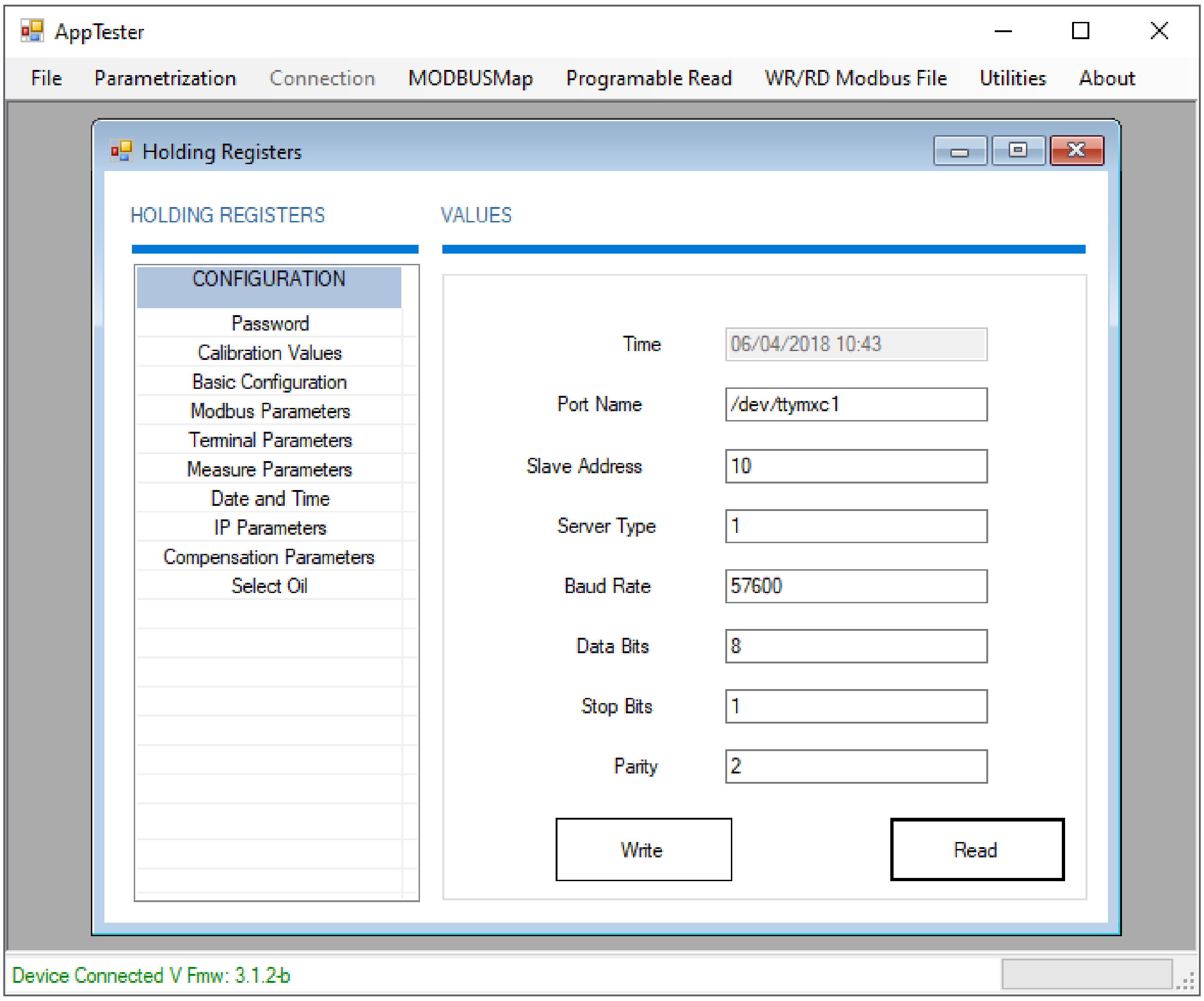
Sensor LED Indicator

Once powered up the LED provides an indication of the sensor condition and TCP/IP communication:

Colour	Status	Information
Red	Stable	Powered up
Orange	Stable	TCP/IP communication
Orange	Flashing	TCP/IP data transmission

Setting up RS485 connectivity from TCP/IP

By default, the device is configured for use over TCP/IP. To change the communication for RS485, Server Type needs to be modified by using the AppTester software and going to MODBUSMap>Holding-Configuration>Modbus Parameters>Read, then change Server Type from 2 to 1 and click "Write".



Time	06/04/2018 10:43
Port Name	/dev/ttymxc1
Slave Address	10
Server Type	1
Baud Rate	57600
Data Bits	8
Stop Bits	1
Parity	2

Device Connected V Fmw: 3.1.2-b

Server Type 1 = RS485, Server Type 2 = TCP/IP

Connecting to AppTester

A connection to the AppTester software is established by going to "Parametrization".

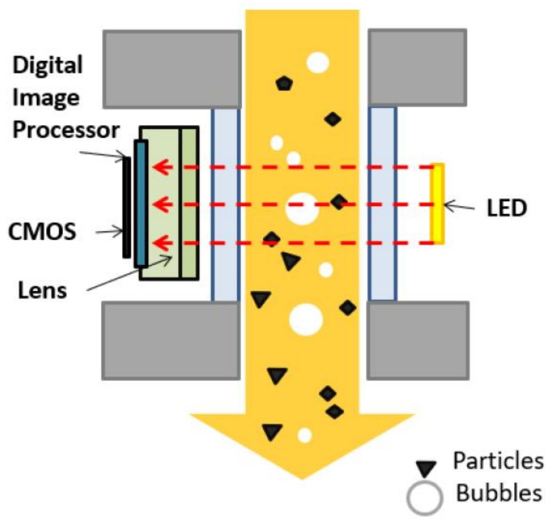
If the device is configured for TCP/IP select "MODBUS TCP", enter the IP address, click "Accept", then click the "Connection" tab.

If the device is configured for RS485 select "MODBUS Serial RTU" and click "Accept". Next go to "Parametrization" > "Port COMM" and enter the relevant COM port being used by the device, then click "Change". Finally click the "Connection" tab.

If a connection is successfully established "Device Connected" will be displayed in green text at the bottom right. If the connection fails "Error Connecting with Device" will be displayed in red text.

Measuring Principle

The sensor measures fluid contamination and condition using a system of optical acquisition and digital image processing algorithms.



The sensor periodically captures images and processes these internally for the delivery of information to the user over Modbus.

The image processing algorithms allow for:

- The differentiation between particles and air bubbles.
- Classifying particles and bubbles by size.
- Identifying oil cleanliness by ISO, NAS & SAE classes.
- Shape recognition for root cause analysis (fatigue, sliding, cutting & fibre).

Modbus Communication

Introduction

Modbus is a communications protocol located at levels 1, 2 and 7 of the OSI Model, based on the master/slave (RTU) or client/server (TCP/IP) architecture, designed in 1979 by Modicon for its range of programmable logic controllers (PLCs). Having become a de facto industry standard communications protocol, it is the most widely available for the connection of industrial electronic devices.

Modbus allows the control of a network of devices and communicates the results to a computer. Modbus is also used for the connection of a supervisory computer with a remote unit (RTU) in supervisory systems data acquisition (SCADA). There are versions of the Modbus protocol for serial port and Ethernet (Modbus/TCP). Information from this section has been obtained from the Modbus organisation reference documentation.

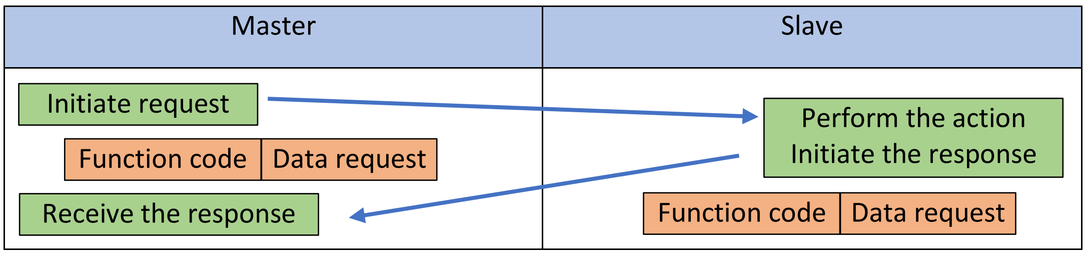
Each device in the Modbus network has a unique address. Any device can send Modbus commands, although usually only one master device is allowed. Each Modbus command contains the address of the device to which the command is sent. All devices receive the frame, but only the recipient executes it. Each of the messages includes redundant information that ensures its integrity at reception. The basic Modbus commands allow an RTU device to be controlled to modify the value of one of its registers or to request the contents of those registers.

Modbus is based on an approach of coils, registers, and functions. The Modbus data model distinguishes between digital inputs (discrete input), digital outputs (coils), input registers, and holding registers. The digital inputs and outputs occupy one bit, while the registers, both input and holding, occupy two bytes. MODBUS uses a ‘big-Endian’ representation for addresses and data items. This means that when a numerical quantity larger than a single byte is transmitted, the most significant byte is sent first.

Primary tables	Object type	Type of	Comments
Discrete input	Single bit	Read only	This type of data can be provided by an I/O system.
Coils	Single bit	Read/Write	This type of data can be altered by an application program.
Input registers	16-bit word	Read only	This type of data can be provided by an I/O system.
Holding registers	16-bit word	Read/Write	This type of data can be altered by an application program.

Types of data registers in the Modbus protocol.

Each device defines its coils and registers in their physical memory where information is stored, and the master send or extract this information. To extract the information, the master requires to send information regarding the function and the value.



Modbus Address Definition

Depending on the communication protocol selected, Modbus address information is different.

Type	Modbus TCP/IP	Modbus RTU
Interface	Ethernet	RS485-2W
Address	IP address*	COM port
Port	502	-
Baud rate*	-	57600
Slave ID*	10	10
Modbus configuration*		Data bits:8 Stop bits: 1 Parity: Even
*configurable upon request		

Results Data Registers

The next table displays the OilWear 2.0 results where the Modbus map can be found. The Modbus map shown below is defined equally for both communication modes, Modbus RTU-RS485-2W and Modbus TCP/IP. Two definitions of the map are shown.

Parameter		OW 2.0 Map - INPUT		OW 2.0 Compatible map – INPUT/HOLDING	
		Data type	Modbus register address	Data type	Modbus register address
Timestamp	-	Int	/*1356-1357*/	Int	/*1042-1043*/
ISO 4406	>4 microns	Float	/*1402-1403*/	Short	/*1030*/
	>6 microns		/*1404-1405*/	Short	/*1031*/
	>14 microns		/*1406-1407*/	Short	/*1032*/
Big particle count	>21 microns	Int	/*1366-1367*/	-	
	>38 microns	Int	/*1362-1363*/	-	
	>70 microns	Int	/*1358-1359*/	-	
Total particles	-	Int	/*1378-1379*/	Short	/*1029*/
Total bubbles	-	Int	/*1380-1381*/	Short	/*1028*/
OD	-	Unsigned char (2nd byte)	/*1416*/	Short	/*1018*/
Shape	Timestamp	Int	/*63000-63001*/	Int	/*1042-1043*/
	Cutting	Short	/*63003*/	Short	/*1035*/

	Cutting [%]	Unsigned char (2nd byte)	/*63014*/	-	
	Sliding	Short	/*63006*/	Short	/*1036*/
	Sliding [%]	Unsigned char (1st byte)	/*63016*/	-	
	Fatigue	Short	/*63004*/	Short	/*1037*/
	Fatigue [%]	Unsigned char (1st byte)	/*63015*/??	-	
	Fibre	Short	/*63005*/	Short	/*1038*/
	Fibre [%]	Unsigned char (2nd byte)	/*63015*/ ??	-	
	Air	Short	/*63002*/	Short	/*1039*/
	Air [%]	Unsigned char (1st byte)	/*63014*/	-	
	Unknown	Short	/*63007*/	Short	/*1040*/
	Unknown [%]	Unsigned char (2nd byte)	/*63016*/	-	
Temperature	[°C]	Float	/*62480-62481*/		
Particle Count	>4 microns	Int	/*1378-1379*/		
	>6 microns	Int	/*1374-1375*/		
	>14 microns	int	/*1370-1371*/		

Both the above maps (orange and grey) contain the same data. The orange map has extended features, such as ISO 4406 decimal values and shape percentages. The grey map is for ease of implementation in terms of data reading, interpreting, and INPUT/HOLDING presence.

Modbus Integration

Successful Modbus integration in an acquisition system requires that several topics are considered:

- Register Addresses and Functions: Modbus data is obtained through a combination of functions and registers. Some Modbus acquisition systems require introduction of both functions and registers in the same area. For example, reading a Modbus INPUT register “1001”, requires introducing “41001”, that is the combination of function 4 “READ INPUT REGISTERS” plus the register 1001.

				Function codes			
				Code	Sub code	Hex	Section
Data access	Bit access	Physical discrete inputs	Read discrete inputs	02		02	6.2
		Internal bits or physical coils	Read coils	01		01	6.1
			Write single coil	05		02	6.5
			Write multiple coils	15		0F	6.11
	16-bit access	Physical input registers	Read input register	04		04	6.4
		Internal registers or physical output registers	Read holding registers	03		03	6.3
			Write single register	06		06	6.6
			Write multiple register	16		10	6.12
			Read/write multiple registers	23		17	6.17
			Mask write register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access		Read file record	20		14	6.14
			Write file record	21		15	6.15

- Modbus offset: In the Modbus/RTU and Modbus/TCP protocols, the addresses are encoded using 16 bits with a number between 0 and 65,535. These are 0-based addresses. Therefore, the Modbus protocol address is equal to the Holding Register Offset minus one. Some acquisition devices have the offset predefined, but others not. So, it is essential to check that the address is pointing to the correct register.
- Confusion about Little-Endian vs Big-Endian Word Order: Although Modbus.org standard documents provide some guidance for implementing the Modbus protocol, they do not address the question of word order beyond the register level. Modbus implementers must make an arbitrary choice as to which address of the register pair contains the most significant word of 32-bit values such as IEEE-754 single-precision floats and signed or unsigned 32-bit integers. Most programs for communicating with Modbus slaves can be configured for either register word order, but the most common default word order today is Little-Endian.
- Register sectioning: Some of the data contained in the sensor Modbus map encapsulates two different data values in the same register, one in each byte. The acquisition system should be able to section this data and interpret this data separately.
- Data type interpretation: Different programming languages offer different names for the variable types available. Once collected, registers must be interpreted correctly. The following table defines the variables as proposed by the sensor interface, with expected ranges.

Variable type	Bytes	Range	Definition
INT	4	0-42949697295	Unsigned 32-bit integer
FLOAT	4	±1,5 x 10^{-45} -±3,4 x 10^{38}	Floating number
SHORT	2	0-65535	Unsigned 16-bit integer
CHAR	1	0-255	Unsigned 8-bit integer

Integration Test

When lubricant is not flowing through the device an ISO code of 8/7/6 will be present. This can be used to confirm correct the Modbus offset.

Maintenance

OilWear 2.0 is designed to operate autonomously during the lifetime of the equipment that it is attached to. Nevertheless, it is advised to carry out a cleaning procedure once a year to ensure the sensor is operating effectively.

To clean the sensor, power it down and safely remove it from the hydraulic system. Flush the sensor using petroleum ether (either with a pump or syringe) followed by a thorough drying with moisture free compressed air. Once clean and dry the sensor can be reinstalled into the system.