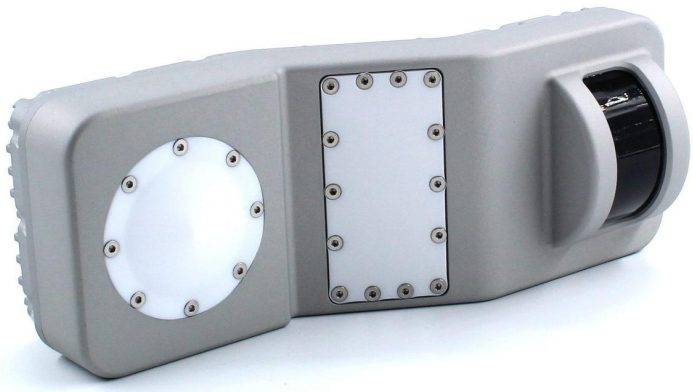
Geolux d.o.o. | Ljudevita Gaja 62 | HR-10430 Samobor Croatia | www.geolux-radars.com | geolux@geolux.hr

FX200



Non-Contact Open Channel Flow Meter

User Manual

v1.0.0.
November 2026

Copyright ©2026 Geolux d.o.o. All rights reserved.

Starting Point

Thank you for purchasing Geolux FX200 non-contact open channel flow meter! We have put together the experience of our engineers, the domain knowledge of our customers, the enthusiasm of our team, and the manufacturing excellence to deliver this product to you. You may freely rely on our field-proven radar technology. The use of top-quality components, state-of-the-art microwave antennas and advanced signal processing algorithms ensures that the Geolux flow meter can be used in various applications and environments. Although we are certain that you are more than capable of connecting the flow meter to your system, we have created this User Manual to assist you in setting up and using the Geolux FX200 flow meter. Should there be any questions left unanswered, please feel free to contact us directly.

Geolux d.o.o.
Ljudevita Gaja 62
10430 Samobor
Croatia

E-mail: ***geolux@geolux.hr***
Web: ***www.geolux-radars.com***

Contents

1. Introduction **4**
2. Device Characteristics **6**
3. Connector Pin-Out and Interfaces **7**
3.1. Serial RS-232 Interface 9
3.2. Serial RS-485 Interface 10
3.3. SDI-12 Interface 12
3.4. Analog 4 - 20 mA Output 13
4. Installing the Flow Meter **14**
4.1. Instrument Mounting and Radar Beam Size 15
4.2. Site and Mounting Requirements 20
4.3. Signal-to-Noise Ratio 23
4.4. Rain and Wind 24
4.5. Interference and Multiple Radars 24
4.6. Fogging and Evaporation 24
4.7. Reflections 25
5. Flow Meter Settings **26**
6. Data Protocols **37**
6.1. NMEA Protocol (RS-232) 37
6.2. Servicing Protocol (RS-232) 39
6.3. Modbus Protocol (RS-485) 55
6.4. SDI-12 Protocol 76
7. Geolux Instrument Configurator **98**
7.1. Establishing Connection 98
7.2. Device Configuration 101
7.3. Using LiDAR 105
7.4. Configuring Discharge Calculation 107
7.5. Echo Curve Acquisition 119
7.6. Raw Data View 120
8. Troubleshooting **122**
A. Appendix A: Mechanical Assembly **125**

1 Introduction

The Geolux FX200 flow meter uses advanced radar technology to accurately measure surface flow velocity and water level (distance) without requiring contact with the water. Because the sensor operates entirely above the water surface, installation is quick and simple, and long-term operation requires only minimal maintenance. The instrument is mounted above the channel or river, eliminating the need for submerged components and reducing the risk of damage from debris, sediment, or flooding.

The FX200 incorporates two independent radar sensors that simultaneously measure surface velocity and water level, and a third lidar sensor for automatic measurement of the riverbed profile.

Surface velocity is measured using a radar operating in the 60 GHz frequency range (V-band). The radar transmits a modulated electromagnetic wave toward the water surface at 60° angle. The signal reflects from the moving water surface and returns to the sensor. By analyzing the frequency shift of the reflected signal, which is caused by both the movement of the water and the radar signal modulation, the instrument determines the velocity of the surface flow.

This measurement method is known as Frequency-Modulated Continuous Wave (FMCW) radar. Using advanced signal processing techniques, the FX200 analyzes the returned radar signal to determine both the distance to the reflecting surface and the velocity of the moving surface. The processed data are represented internally as a range-Doppler map, which shows the velocity of objects at different distances from the sensor.

Water level is measured independently using a dedicated level radar sensor. The radar periodically transmits a linear frequency chirp ranging from 77 GHz to 81 GHz (W-band). The signal reflects from the water surface and returns to the sensor. The difference between the transmitted and received frequencies is proportional to the distance between the sensor and the reflecting surface. As this distance changes, the frequency difference changes accordingly, allowing the FX200 to determine the water level with high precision.

The radar modulation and signal processing algorithms used in the FX200 enable highly accurate measurements that are not affected by air temperature, humidity, or other environmental conditions.

Unlike conventional surface velocity meters that use simple Doppler radar, FMCW radar technology can separate reflections originating from objects at different distances. This capability is particularly important in hydrological applications. For example, during rainfall events the FX200 can clearly distinguish reflections from raindrops and those from the river surface. Conventional Doppler radars cannot fundamentally separate these signals and must rely on heuristic filtering techniques to reduce rain-induced measurement errors.

The FX200 measures surface velocity in the range 0.020 m/s to 16.000 m/s, with a resolution of 0.001 m/s.

The instrument also includes an integrated tilt sensor that continuously monitors the mounting angle of the device. Measured surface velocity values are automatically corrected using cosine compensation based on the detected tilt angle, ensuring accurate measurements even if the mounting structure shifts over time.

Flow (discharge) is calculated internally by combining the measured surface velocity and water level with the measured or configured cross-section profile of the river or channel. Built-in lidar can automatically measure the cross-section profile of the channel.

Once properly configured, the FX200 can calculate discharge with an accuracy of approximately ±1% compared to ADCP measurements at the same location. In addition to discharge, the sensor simultaneously

provides surface velocity and water level measurements via its digital communication interfaces, as well as the channel profile geometry.

2 Device Characteristics

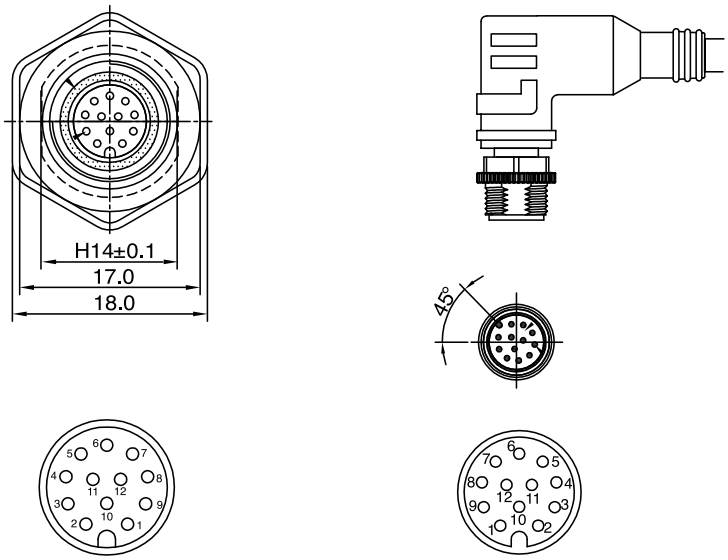
Table 1. Technical specifications

Velocity Radar	V-band 60 GHz FMCW radar
Level Radar	W-band 77 – 81 GHz FMCW radar
Channel Profile Measurement	Integrated LiDAR (up to 1000 points per scan)
Installation Height	0.2 – 30 m above the water surface
Velocity Measurement Range	0.02 – 16 m/s
Velocity Resolution	0.001 m/s
Velocity Accuracy	1% of measured value
Level Measurement Range	0.2 - 35 m
Level Resolution	1 mm
Level Accuracy	± 2 mm
LiDAR Range	50 m radial
Discharge Accuracy	Approx. ±1% compared to ADCP measurements
Tilt Compensation	Integrated tilt sensor, cosine compensation
Data Interfaces	RS-232 / NMEA RS-485 / Modbus-RTU SDI-12 Analog 4-20 mA Wi-Fi hotspot (configuration)
Connector	M12 circular, 12-pin, IP68
Ingress Protection	IP68
Input Voltage	9 – 27 VDC
Power Supply Requirement	Min. 500 mA
Power Consumption	1100 mW operational
Operating Temperature Range	-40°C to +85°C
Operating Humidity Range	0 – 100% RH

3 Connector Pin-Out and Interfaces

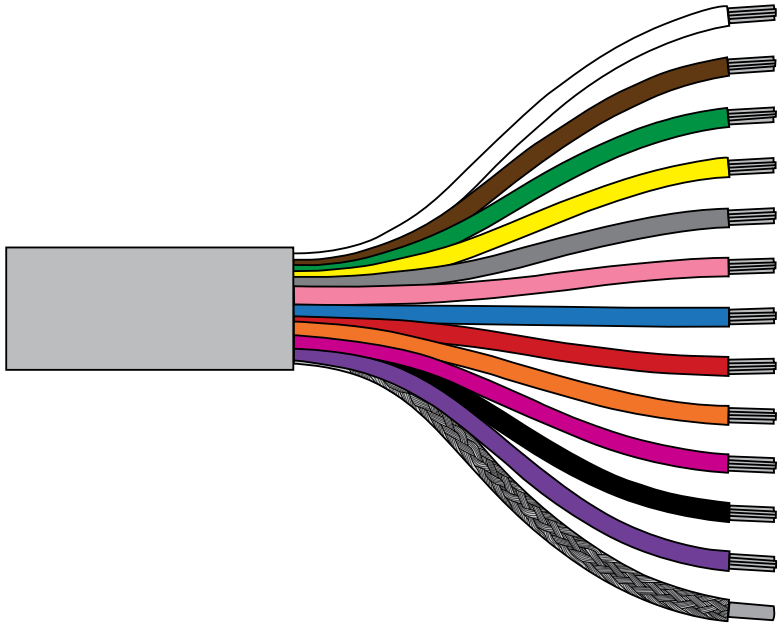
The flow meter is equipped with a robust IP68-rated circular M12 connector with 12 pins. The mating cable is supplied with the instrument. All communication and output interfaces are available on every model. These include RS-232, RS-485, SDI-12, and analog 4–20 mA. All interfaces can operate simultaneously, allowing parallel communication and data output. The exception is when the device is configured to operate in automatic SDI-12 sleep mode, when the other interfaces will be disabled, and only SDI-12 will be active.

Connector and Cable Drawing



Both the chassis connector (left) and cable connector (right) are shown from the front.

Connector Pin-Out



The table below describes the function of each pin.

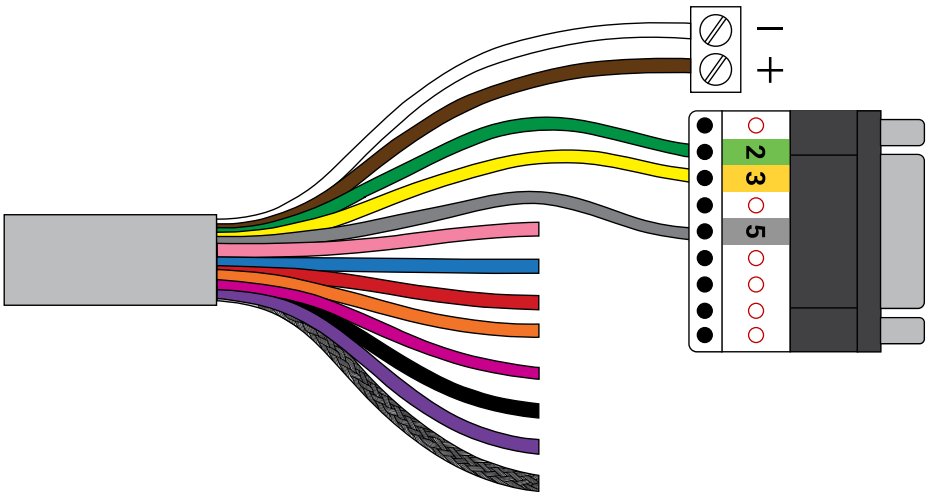
Pin No.	Wire Color		Pin Name	Pin Description
1	White		Power Ground	Connect to the negative (–) terminal of the power supply.
2	Brown		Power Input	Power supply input. The supply voltage must be between 9–27 VDC. The power source must be capable of providing at least 500 mA
3	Green		RS232 TX	RS-232 transmit data signal (device → host).
4	Yellow		RS232 RX	RS-232 receive data signal (host → device).
5	Grey		Signal Ground	Signal ground reference for RS-232 or SDI-12 communication.
6	Pink		CAN H	CAN 2.0 high signal line.
7	Blue		CAN L	CAN 2.0 low signal line.
8	Bright Red		4-20 mA Loop+	Positive terminal of the active 4–20 mA output.
9	Orange		RS485 B-	RS-485 data low (inverting) signal.

Pin No.	Wire Color		Pin Name	Pin Description
10	Dark Red		RS485 A+	RS-485 data high (non-inverting) signal.
11	Black		SDI-12 Data	SDI-12 communication data line.
12	Purple		4-20 mA Loop-	Negative terminal of the active 4–20 mA output.
13	Clear		Shield	Cable shield. Connect to grounding.

3.1. Serial **RS-232 Interface**

The instrument includes a standard full-duplex RS-232 serial interface. The voltage levels are fully compliant with the RS-232 standard and allow direct connection to a PC or other embedded devices that support RS-232 communication.

When connecting the instrument to a standard DB-9 PC connector, use the following wiring:



- Green wire - RS-232 TxD → DB-9 pin 2
- Yellow wire - RS-232 RxD → DB-9 pin 3
- Grey wire - Signal GND → DB-9 pin 5

Connection of the signal ground (GND) is required for proper operation of the serial interface.

For computers without a native RS-232 port, Geolux recommends using the Waveshare USB to RS232/485/TTL converter: ***https://www.waveshare.com/catalog/product/view/id/3629/s/usb-to-rs232-485-ttl/category/37/***

Serial RS-232 interface is used for direct connection of a single flow meter unit with the computer. The serial interface is used both for retrieving live flow measurements and for configuration of the flow meter. Geolux provides a PC application for unit configuration and flow monitoring free of charge. Default communication parameters are:

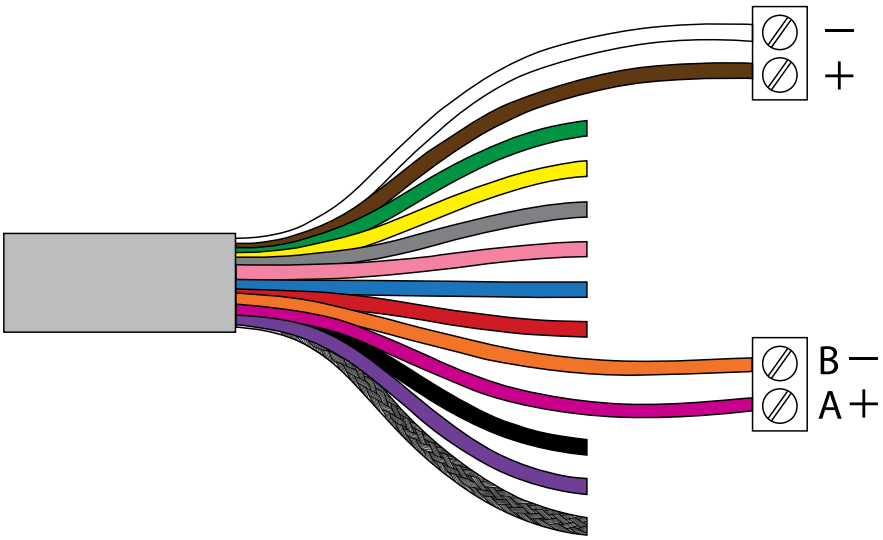
Bitrate: 115200 bps
Data bits: 8
Stop bits: 1
Parity: None

A NMEA-like communication protocol is used to deliver flow measurements over RS-232 interface. Detailed description of the protocol is given in 6.1. NMEA Protocol (RS-232).

3.2. Serial **RS-485 Interface**

The instrument includes a standard industrial half-duplex RS-485 communication interface. The interface is internally protected against short circuits and overvoltage.

The RS-485 interface is connected using two wires:



- Dark red wire - RS-485 A+ signal
- Orange wire - RS-485 B- signal

> *Important:*
>
> Carefully identify the dark red wire used for A+. The cable also contains a bright red wire and a pink wire, which serve different functions. Ensure the correct wire is selected before making the connection.

In some installations, an additional connection of signal GND (grey wire) may be required. The need for a ground connection depends on the receiving device. Always consult the receiver or data logger specification to confirm the correct wiring.

Serial RS-485 interface is used for connecting multiple flow meters to a single data logger. RS-485 interface uses a different protocol than the protocol used over RS-232 interface, in order to allow multiple flow meters connected on a single RS-485 bus. The main difference from the protocol used over RS-232 interface is that the flow measurements are not reported automatically, but are instead reported only after being requested by the master device (data logger unit). Geolux FX200 uses industry-standard Modbus protocol for data exchange over RS-485 interface. The detailed description of the protocol is given in 6.3. Modbus Protocol (RS-485).

Default communication parameters are:

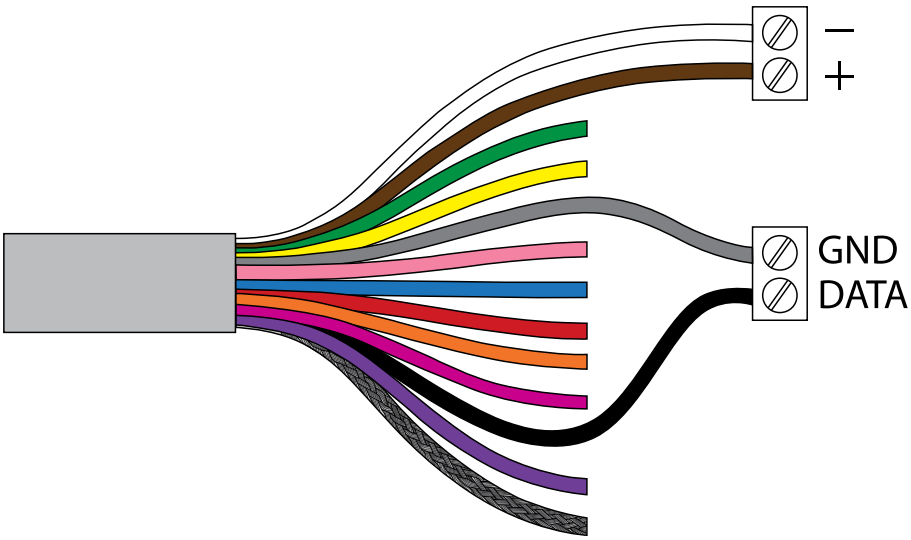
Bitrate: 9600 bps
Data bits: 8

Stop bits: 1
Parity: Even
Device ID: 1

3.3. SDI-12 **Interface**

The FX200 supports SDI-12 (Serial Data Interface at 1200 baud) protocol, which is widely used in environmental monitoring and hydrological measurement systems. SDI-12 communication enables connection to standard environmental data loggers using a single data line. The protocol supports low-power operation, making it suitable for battery or solar-powered installations.

For SDI-12 communication, connect the black wire (SDI-12 Data) to the SDI-12 data terminal of the data logger and the grey wire (Signal Ground) to the logger's signal ground (GND).



- Black wire – SDI-12 data
- Grey wire – signal ground

The instrument supports standard SDI-12 measurement commands as well as extended and custom commands for configuration, diagnostics, and service functions.

A complete list of supported SDI-12 commands and response formats is provided in 6.4. SDI-12 Protocol.

3.4. Analog **4 - 20 mA Output**

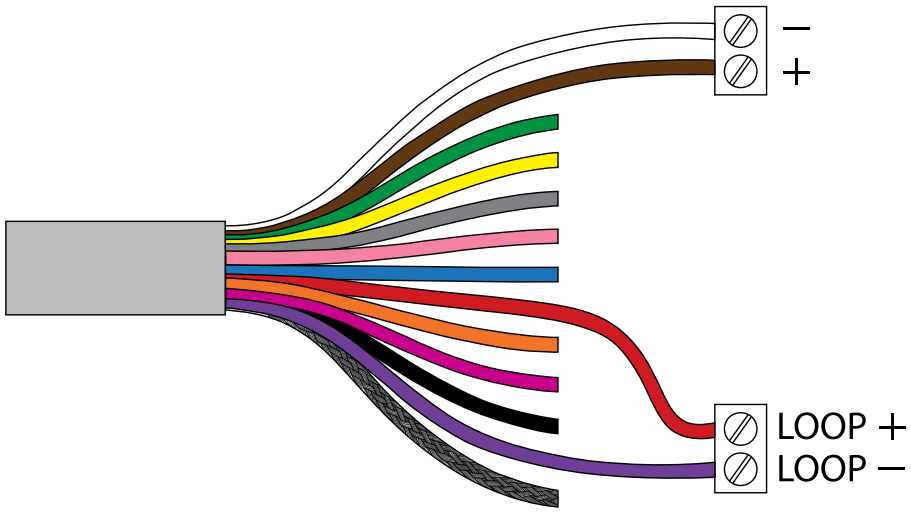
The FX200 flow meter is equipped with an analog 4–20 mA output for integration with PLCs, SCADA systems, and other industrial control equipment. The output is implemented as a 4-wire active current output (not a 2-wire loop-powered type). This means the device provides the current signal directly and requires a separate power supply connection.

The 4–20 mA output must be configured in the device settings to define the linear relationship between the measured value and the output current. The user must set:

- Minimum measurement value → corresponding to 4 mA
- Maximum measurement value → corresponding to 20 mA

The output signal is linearly scaled between these two configured points. The instrument can be configured to output either discharge, water level, surface velocity, distance to water or total volume measurement to the analog output.

Wiring identification:



- Bright red wire – 4–20 mA positive (+)
- Purple wire – 4–20 mA negative (–)

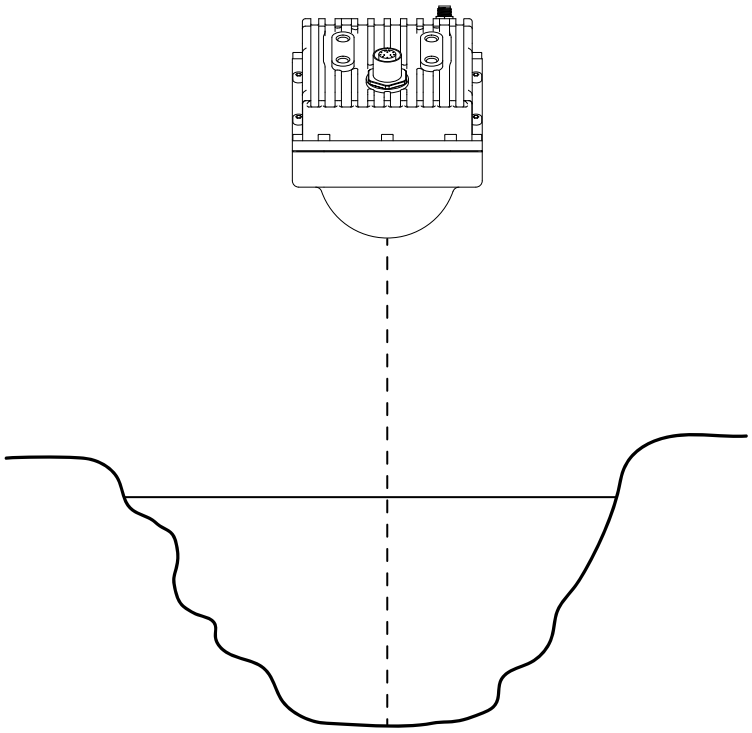
During sensor initialization, if low power mode is activated, or during sensor errors (such as tilt angle which is too big), the device will output 0 mA to differentiate between lowest measurement at 4 mA.

4 Installing the Flow Meter

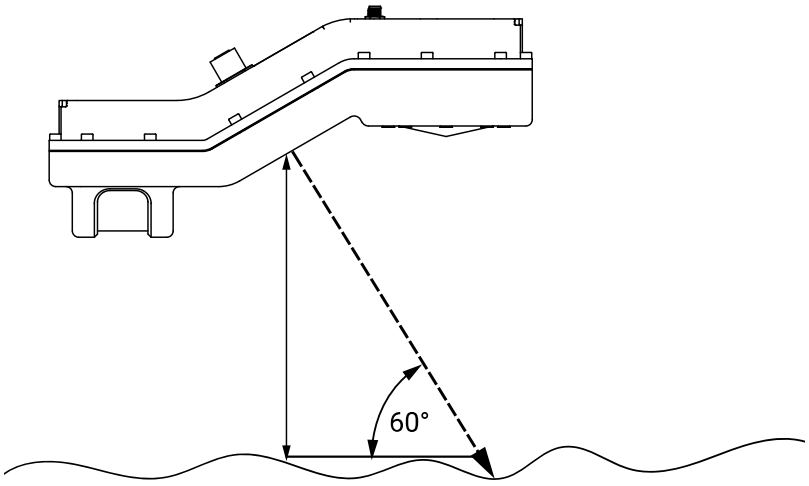
The flow meter must be installed above the water surface. During installation, the instrument should be properly leveled and must not be tilted either forward or sideways. A bubble level indicator is integrated into the top cover of the device to assist with accurate leveling during installation.

The recommended installation height is at least 1 meter above the water surface. The minimum possible installation height is 20 cm, while the maximum installation height is 30 meters above the water surface. For optimal performance, the sensor should be positioned above the deepest part of the river channel, to ensure that measurements remain available even during low water levels.

The picture illustrates the correct positioning of the radar in relation to the water surface.



Picture 1. Front view of the radar mounting.



Picture 2. Side view of the radar mounting: the instrument should be leveled - check bubble level indicator.

More detailed information on mounting and installation is available in the following chapters:

4.1. Instrument **Mounting and Radar Beam Size**

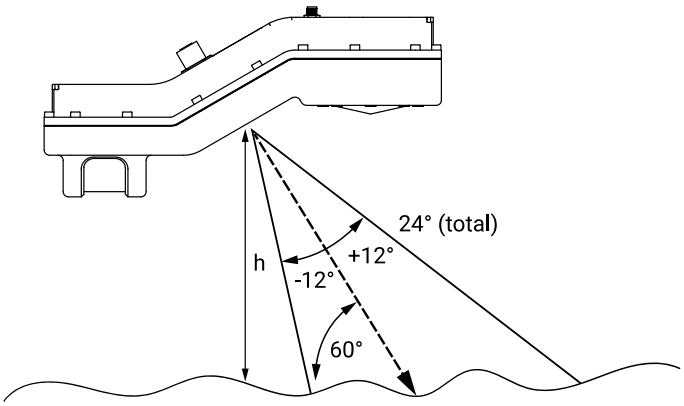
Instrument Mounting

Proper site selection and correct installation are essential to achieve the specified measurement accuracy.

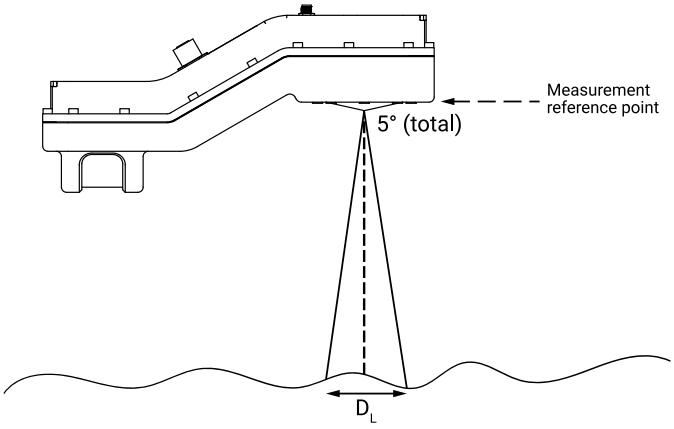
The instrument should be aligned parallel to the direction of water flow. Any deviation from the flow direction will reduce the measured velocity value. The measured value will always be lower than the actual surface velocity if the instrument is not properly aligned. It is recommended to point the instrument upstream, so the water flows toward the sensor.

Radar Beam Width

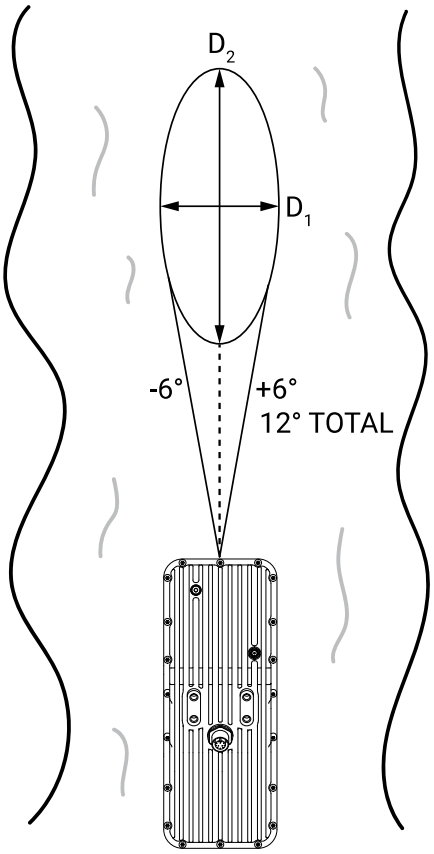
The installation height and tilt angle determine the size and position of the radar measurement areas on the water surface. The FX200 uses two separate radar beams: one for surface velocity measurement and one for water level measurement. The surface velocity radar illuminates an elliptical measurement area on the water surface. The instrument reports the average surface velocity within this area. The water level radar is directed vertically downward and forms a circular measurement area directly beneath the sensor.



Picture 3. Surface velocity radar beam in the elevation plane



Picture 4. Water level radar beam



Picture 5. Surface velocity radar beam in the azimuth plane

The dimensions of both the ellipse and the circle depend on the installation height of the instrument above the water surface. The formulas used to calculate the diameters of these measurement areas, along with a table showing the diameters for various installation heights, are provided below.

$$D_1 = 2h\frac{\tan(6°)}{\sin(60°)} \qquad D_2 = h\,[\tan(42°) - \tan(18°)] \qquad D_l = 2h\tan\left(\frac{5°}{2}\right)$$

Table 2. Beam footprint by installation height (metric)

Height (m)	D_1 (m)	D_2 (m)	D_I (m)
1	0.24	0.58	0.09
2	0.48	1.15	0.17
5	1.21	2.88	0.44
10	2.43	5.75	0.87
20	4.85	11.51	1.75
30	7.28	17.26	2.62

Table 3. Beam footprint by installation height (imperial)

Height (ft)	D_1 (ft)	D_2 (ft)	D_I (ft)
5	1.21	2.88	0.44
10	2.43	5.75	0.87
20	4.85	11.51	1.75
30	7.28	17.26	2.62
60	14.56	34.53	5.24
90	21.84	51.80	7.86

Controlling Radar Beam

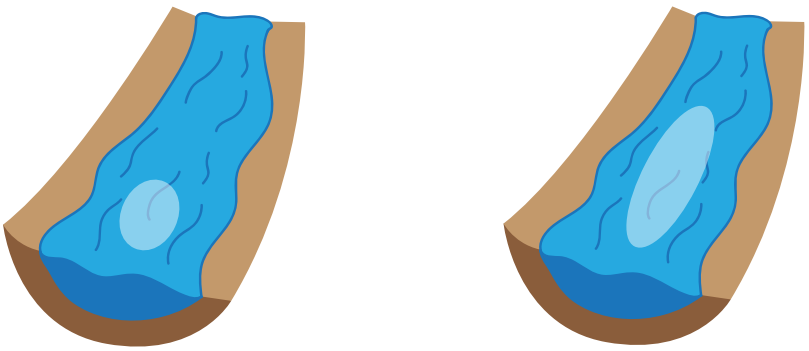
Unlike traditional Doppler-based radars, the surface velocity sensor of the FX200 uses frequency-modulated technology that enables advanced control of the effective measurement area on the water surface.

The previous section describes the *maximum* radar beam footprint, which is defined by the antenna geometry and installation conditions. However, the FX200 does not need to use the entire footprint. Through modulation control and signal processing, the instrument can limit measurements of the surface velocity to a selected portion of that area. This allows the user to focus on the most representative flow conditions and avoid disturbed or non-uniform regions.

Beam Control Parameters

The FX200 provides two configurable parameters for adjusting the measurement area:

1. Beam Width

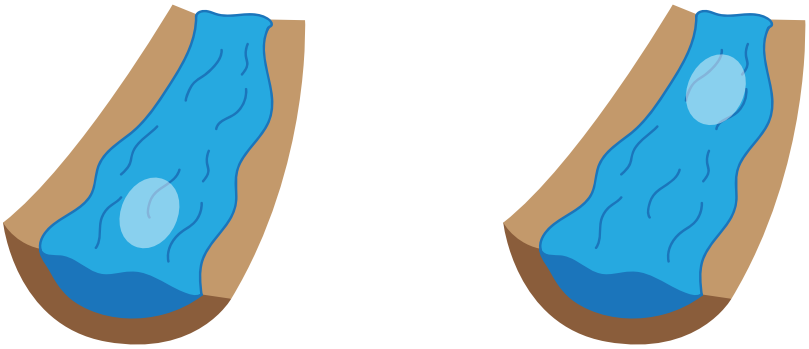


This parameter controls the size of the measurement area in the elevation direction (i.e., effectively reduces the D2 diameter of the ellipse).

Available options:

- **Narrow** – restricts measurements to a smaller, more focused region; recommended when flow disturbances are present and a more stable region must be isolated
- **Wide** – uses a larger portion of the radar beam; suitable for uniform flow conditions

2. Beam Offset

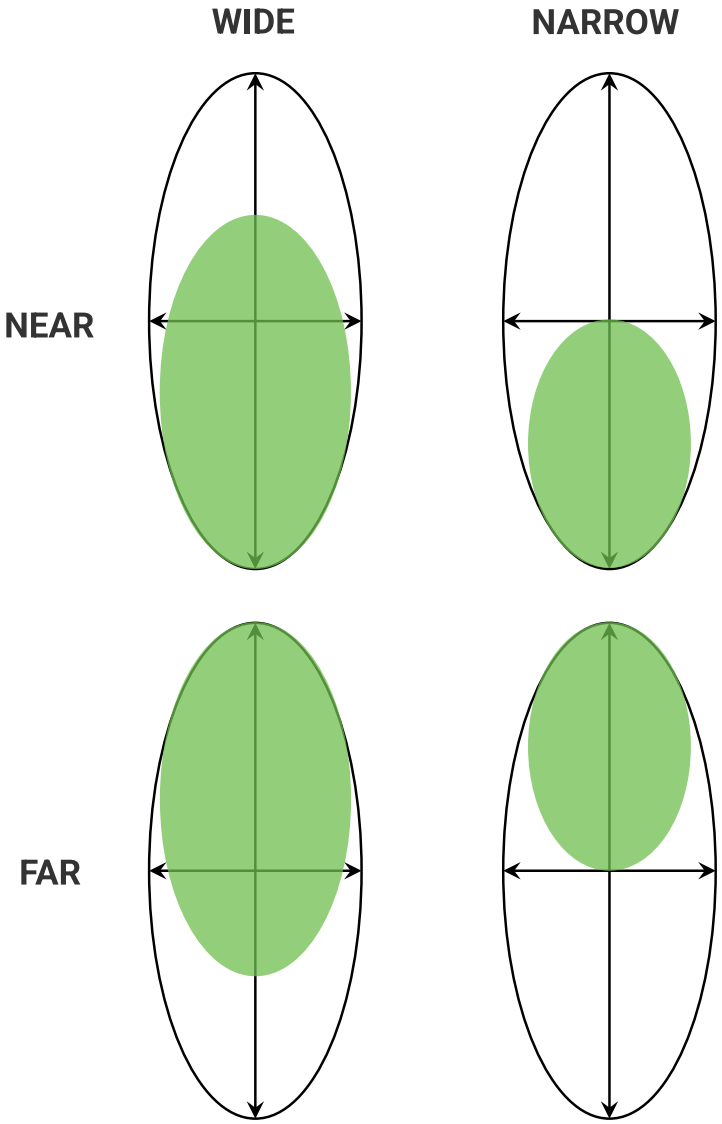


This parameter defines the position of the measurement area within the overall radar footprint. It allows the user to shift the active measurement zone along the flow direction.

Available options:

- **Near** – measurement area is closer to the sensor
- **Far** – measurement area is further away from the sensor

Radar Beam Configuration Matrix



Picture 6. Effect of beam parameters on active scanning area.

Correctly Adjusting Beam Width and Position

In real-world installations, flow conditions are often non-uniform. Turbulence, obstacles, or channel geometry can create areas where velocity measurements are not representative of the average surface flow. By combining **Beam Width** and **Beam Offset**, the VX60 allows the user to target the most stable portion of the flow.

Example 1 – Upstream cascade or turbulence away from the sensor
Disturbance located further from the mounting point.
Recommended settings:
- **Beam Width:** Narrow
- **Beam Offset:** Near

–

Example 2 – Turbulence near a bridge pier or mounting structure
Disturbance located close to the sensor.
Recommended settings:
- **Beam Width:** Narrow
- **Beam Offset:** Far

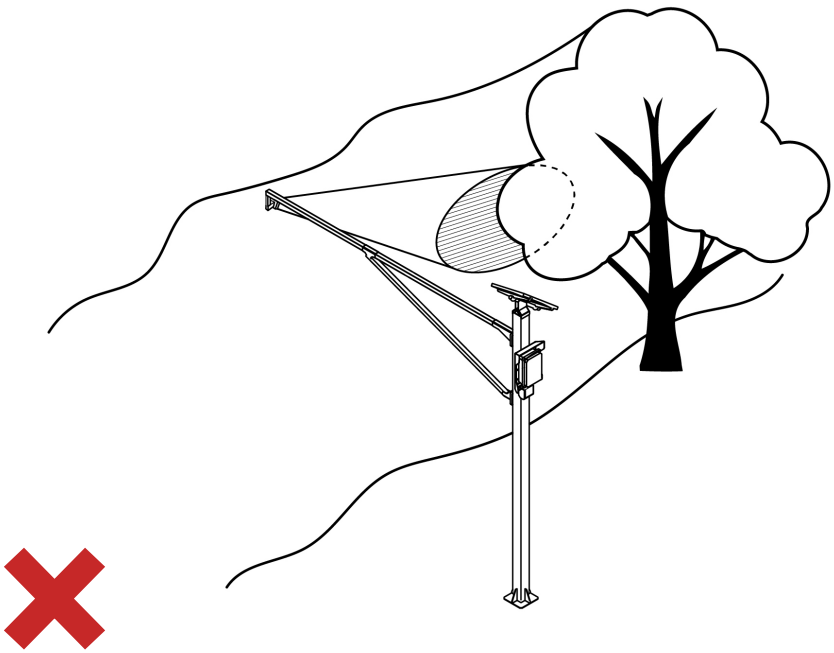
–

Example 3 – No disturbace or turbulence
The water flow is uniform.
Recommended settings:
- **Beam Width:** Wide
- **Beam Offset:** Near

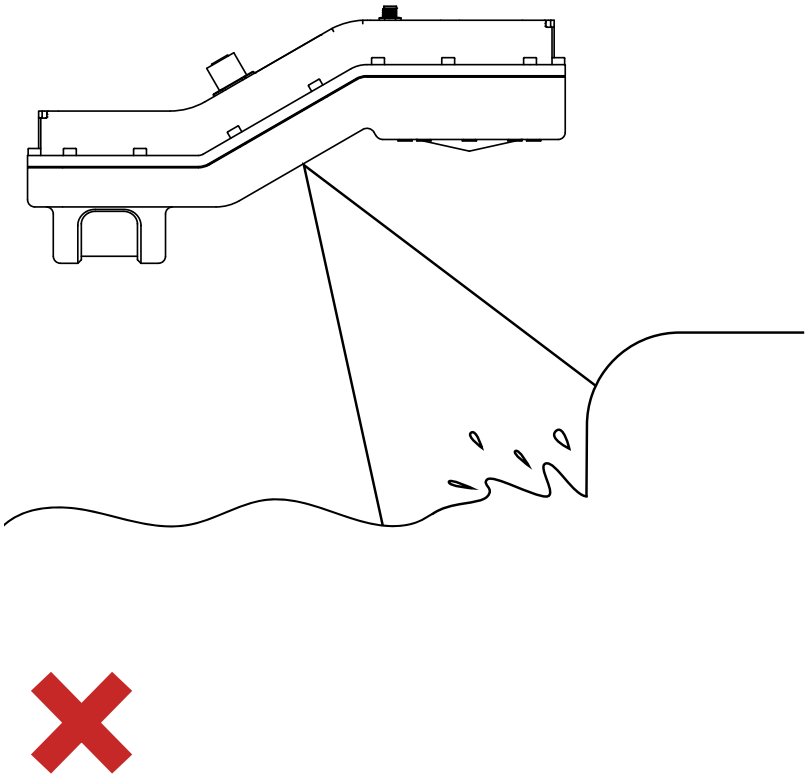
4.2. Site and Mounting Requirements

The measurement area must be free of obstacles, contain no vegetation, be free of rocks, sand deposits, cascades or waterfalls, must not include bridge structures or other fixed elements, and there must be no vegetation or objects between the radar and the water surface, as this can significantly affect measurement accuracy.

The mounting structure used to hold the instrument, such as a pole, bridge railing, or similar support, must be mechanically stable, free of vibrations, and withstand wind and environmental loads. Vibrations can negatively affect measurement stability and accuracy.

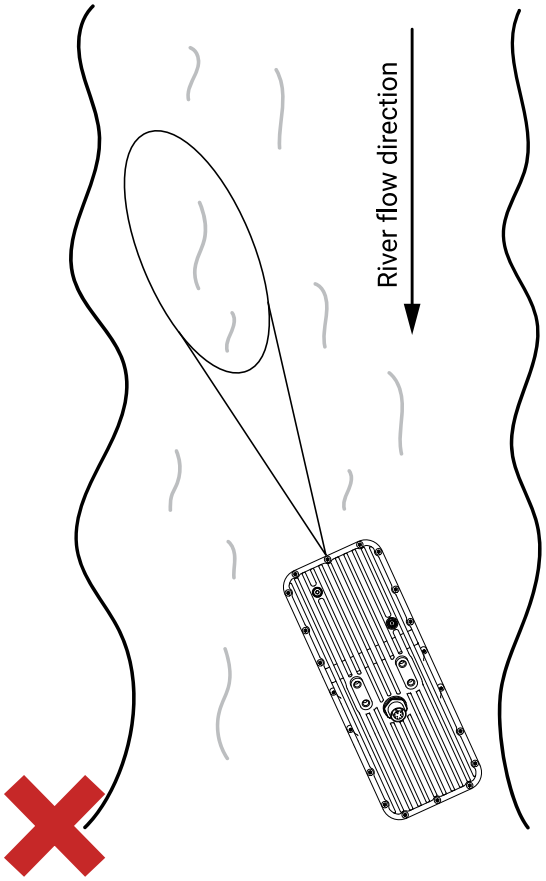


Picture 7. Ensure the radar beam is not obstructed.



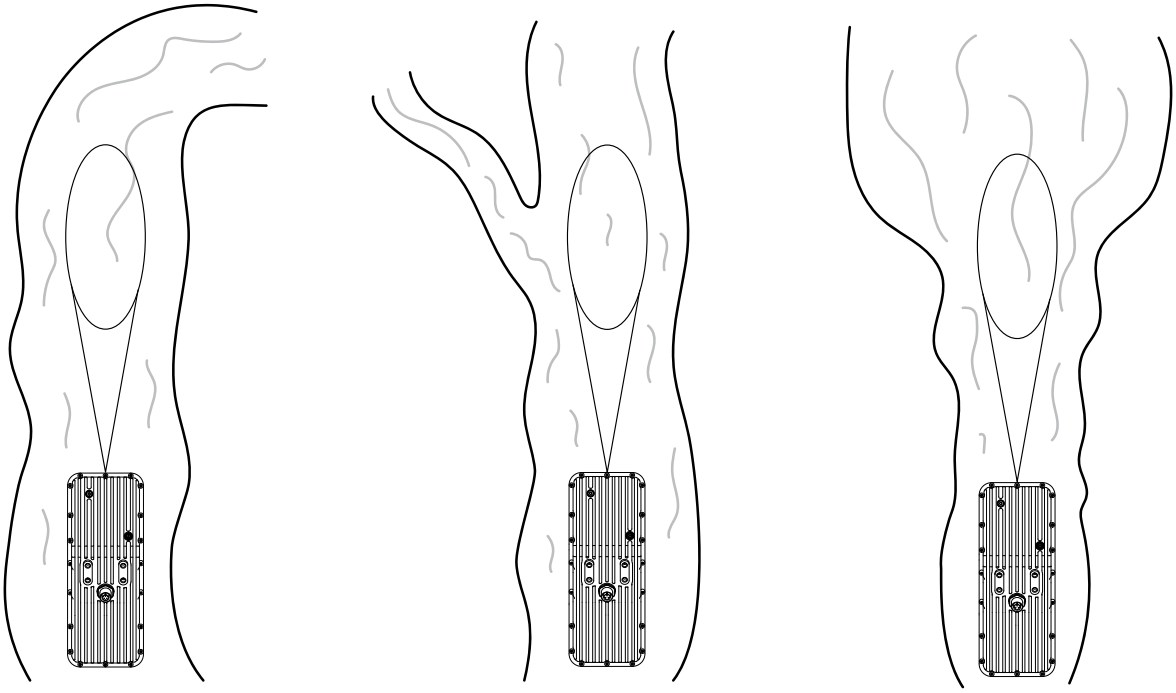
Picture 8. Waterfalls/cascades must be avoided.

The instrument must be aligned with the direction of water flow. Proper alignment ensures that the radar measures the true surface velocity of the moving water.



Picture 9. The instrument must be aligned with water flow direction.

For optimal measurement accuracy, install the instrument in a section of the river where the flow is uniform and stable. Avoid locations with channel bends or sharp curvatures, tributaries entering the measurement area, sudden widening or narrowing of the channel, and highly turbulent flow. As a general guideline, the minimum distance between the sensor location and any disturbance (such as a tributary, cascade, or channel bend) should be at least four times the width of the river.



The FX200 uses advanced signal processing and Kalman filtering with physical modeling of water flow. This ensures stable measurements even under turbulent conditions. Moderate surface waviness improves radar signal reflection and measurement quality. In cases of strong turbulence, increased fluctuations in measured values can occur which may result in reduced accuracy of the measurement. If strong turbulence is expected at the monitoring site, it is recommended to set the radar filter length to 60 seconds or higher.

4.3. Signal-to-Noise **Ratio**

The SNR (Signal-to-Noise Ratio) indicator represents the ratio between the useful Doppler signal reflected from the moving water surface and the background noise detected by the radar receiver. In surface velocity radar applications, SNR is a key quality parameter that indicates the reliability of the measured velocity. A higher SNR value means that the reflected signal from the water surface is significantly stronger than the noise floor, resulting in more accurate and stable measurements. Conversely, a low SNR indicates that the useful signal is weak compared to background noise, which may reduce measurement stability or increase uncertainty. SNR is typically influenced by installation geometry, mounting height, surface conditions, and environmental factors. Smooth, uniform flow with well-defined surface ripples generally produces stronger Doppler reflections and higher SNR values. Very calm water surfaces may reduce signal strength because fewer surface irregularities are present to reflect the radar signal. Excessive turbulence, splashing, or debris can also affect signal consistency, sometimes lowering the effective SNR. Proper positioning and installation of the radar significantly improves signal strength. Ensuring that the radar beam is unobstructed and correctly aimed at the target measurement zone is essential. Avoiding installation locations with heavy spray, foam accumulation, or floating debris can further improve signal quality. In some cases, adjusting internal parameters such as filter length or direction filter may help optimize performance in challenging conditions. Monitoring the SNR indicator during commissioning provides valuable feedback about installation quality and overall measurement reliability.

4.4. Rain and Wind

Unlike conventional Doppler radars, whose accuracy can be significantly affected by rainfall, the FX200 uses advanced FMCW radar signal processing. This technology enables the instrument to distinguish between reflections from raindrops and reflections from the water surface. As a result, the FX200 can measure surface flow velocity reliably even during rainfall.

Wind generally has only a minor influence on measurement accuracy and can usually be neglected. The main exception is strong wind. Strong wind can generate surface waves that travel in a different direction than the main water flow. This may introduce deviations in the measured surface velocity.

In locations where strong wind is common, the following measures are recommended:

- Install the instrument at a sheltered location, if possible (for example, under a bridge structure or in a naturally protected river section).
- Avoid installation at wide, open sections where wind can generate significant cross-waves.
- Increase the filter duration (filter length) to improve measurement stability under fluctuating surface conditions. A longer filter duration reduces short-term fluctuations caused by wind-driven waves and improves overall measurement stability.

4.5. Interference and Multiple Radars

Due to the FMCW modulation technique and very short transmission bursts, multiple FX200 radars can operate in close proximity with very low risk of mutual interference. The radars are not time-synchronized, so overlap of their modulation patterns is typically brief and does not produce measurable degradation in normal installations. In addition, the radar antenna is highly focused and the transmitted power is low. This further reduces the possibility of interference between devices. In Geolux internal tests, 10 radars were operated within 2 meters of each other, and no interference was observed.

The Geolux FX200 can also operate reliably in the presence of other radar systems, including:

- FMCW radars operating at 60 GHz and 80 GHz
- Radars operating at other frequencies (for example 24 GHz)
- Radars using different modulation techniques

4.6. Fogging and Evaporation

Radar sensors are generally not affected by fog or regular water evaporation. Measurement performance remains stable under typical environmental conditions.

The exception is in the case of very heavy evaporation, where the concentration of water vapor directly above the surface is extremely high, such as wastewater channels or sewage pipes. In such conditions, the surface

velocity measurement may be affected. If strong evaporation is present, the following configuration is recommended (this overrides general alignment guidance):

- Install the instrument so that it measures outgoing flow direction (water flowing away from the radar).
- Enable or configure the sensor to use outgoing-only directional filtering.

Evaporation rises vertically upward from the water surface and moves toward the radar. By enabling downstream directional filtering, signals traveling toward the radar (inbound direction) are rejected. In most cases, this effectively eliminates the influence of evaporation on the measurement.

4.7. Reflections

Water is a highly reflective surface for radar waves. Most of the transmitted radar energy is reflected from the water surface. Radar reflections follow the same basic physical principles as light reflection. When the radar beam reaches the water surface, a part of the energy is reflected away from the radar, a part of the energy is reflected back toward the radar, and only a small portion of the energy is absorbed by the water.

The amount of energy reflected back toward the radar depends mainly on the surface roughness of the water and the incident angle of the radar beam. As a general rule, a rougher water surface produces stronger reflections back toward the radar. This increases the signal-to-noise ratio (SNR) and improves measurement accuracy. Even a small surface roughness of approximately 1 mm is usually sufficient for precise measurement.

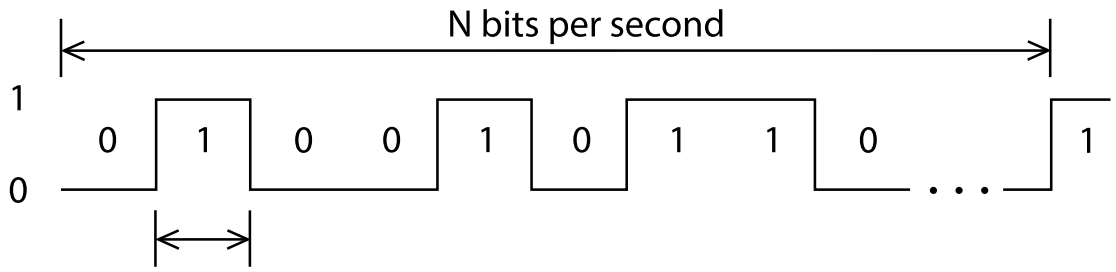
The Geolux flow meter is designed with advanced signal processing techniques that allow accurate operation even in low SNR conditions.

Indoor installation is generally not recommended. Radar signals may reflect from walls, ceilings, moving machinery, or rotating objects. These reflections can cause incorrect readings or false velocity measurements. For reliable operation, the instrument should be installed in an open outdoor environment with a clear and unobstructed measurement area.

5 Flow Meter Settings

This chapter groups the configurable device parameters exposed through the configurator and device protocols.

Baud Rate



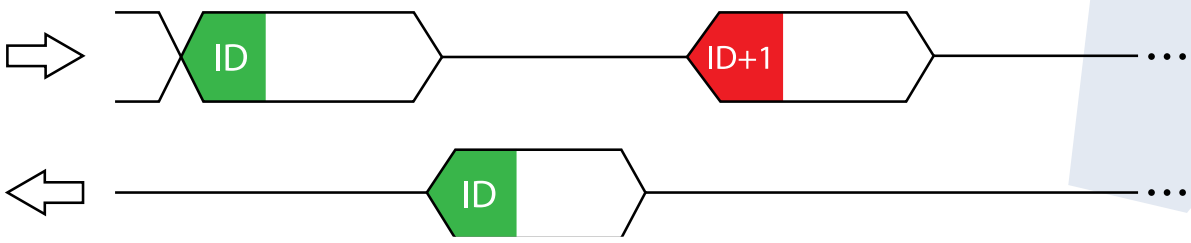
RS-232 Baud Rate

Configures the baud rate (bits per second) for serial communication on the RS-232 data line. This setting controls how many bits are sent on the communication line in one second. The available values are standardized. Using higher baud rate over longer lines may introduce errors in transferred data. The default instrument baud rate is 115200 bps.

RS-485 Baud Rate

Configures the baud rate (bits per second) for serial communication on the RS-485 data line. This setting controls how many bits are sent on the communication line in one second. The available values are standardized. Using higher baud rate over longer lines may introduce errors in transferred data. The default instrument baud rate is 9600 bps.

Device ID



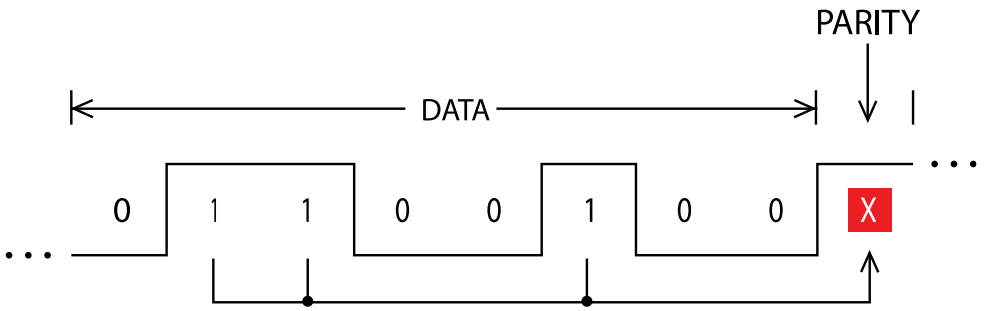
Modbus ID

Configures the device (slave) ID to be used on Modbus interface protocol. Modbus protocol uses request/response format and allow multiple instruments to be connected on the same bus. When a remote master transmits the request message, it will use the device ID as a device address. All instruments will receive the request, but only the instrument with matching device ID will answer to the received request.

SDI-12 ID

Configures the SDI-12 device ID to be used on SDI-12 interface. SDI-12 protocol uses request/response format and allows multiple instruments to be connected on the same bus. When a remote master transmits the request message, it will use the device ID as a device address. All instruments will receive the request, but only the instrument with matching device ID will answer to the received request.

Modbus Parity



Parity is used in serial communication as a basic method of bit-level error detection. When parity is set to *None*, no parity bit is added to the transmitted data. In this case, no bit-level error checking is performed. When *Even* or *Odd* parity is selected, an additional parity bit is appended to each transmitted byte. This bit is set so that the total number of logical “1” bits in the byte (including the parity bit) is either even or odd, depending on the selected mode. If the receiving device detects that the calculated parity does not match the received parity bit, a communication error is identified.

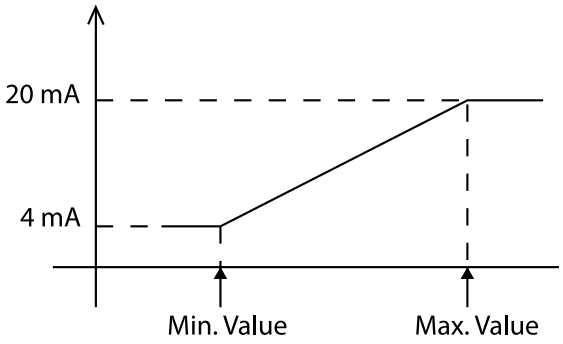
For most Modbus devices, the default and commonly used setting is Even parity.

RS-485 Stop Bits



Stop bits are added to the end of each data byte transferred over serial line, to allow pause between the two bytes of data. One or two stop bits may be used. The default setting for most Modbus devices is one stop bit, but some dataloggers require that two stop bits be used.

4 – 20 mA Parameters



4 - 20 mA Output Type

Selects which measurement parameter is transmitted via the analog 4–20 mA output. The user can choose between discharge (flow), water level, distance to the water surface, surface velocity, or total volume.

4 - 20 mA Minimum Value

To configure the 4-20 mA output range, the minimum measured value corresponding to 4 mA must be defined. The value is entered and the corresponding measurement unit (e.g., m³/s, cfs, L/h) is selected. This setting determines the lower point of the linear scaling between the measured velocity and the analog output current.

Example:

If the expected measurement range is between 0.700 m³/s and 1.500 m³/s, it is recommended to set the 4 mA value slightly below the lowest expected measurement (e.g., 0.500 m³/s). This ensures proper scaling and avoids signal saturation at the lower end of the operating range.

If measurement resolution is not critical, the minimum value for the 4-20 mA output may alternatively be left at the instrument minimum of 0 m³/s.

4 - 20 mA Maximum Value

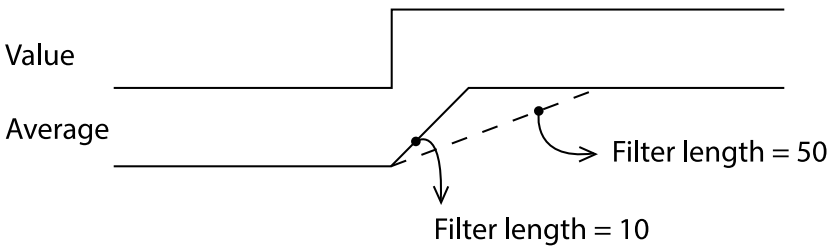
To configure the 4-20 mA output range, the maximum measured value corresponding to 20 mA must be defined. The value is entered and the corresponding measurement unit (e.g., m³/s, cfs, L/h) is selected. This setting determines the higher point of the linear scaling between the measured velocity and the analog output current.

Example:

If the expected measurement range is between 0.700 m³/s and 1.500 m³/s, it is recommended to set the 20 mA value slightly above the highest expected measurement (e.g., 1.700 m³/s). This ensures proper scaling and avoids signal saturation at the higher end of the operating range.

If measurement resolution is not critical, the maximum value for the 4-20 mA output may alternatively be left at the instrument maximum of 16.000 m³/s.

Filter Length

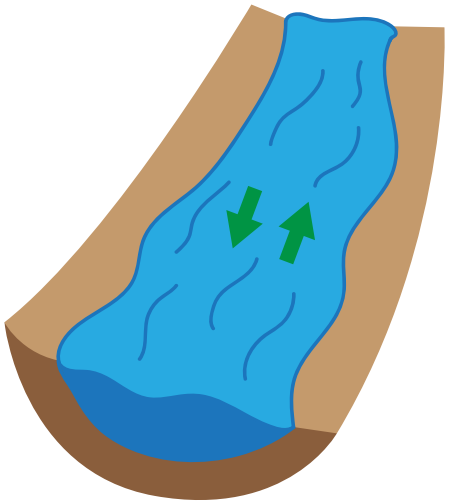


This parameter defines the length of the averaging filter, in seconds, used to smooth the measured velocity and water level values.

A longer filter length means that a greater number of individual measurements are included in the averaging process, resulting in smoother and more stable output data. However, increasing the filter length also increases the response time of the instrument. When surface velocity or water level changes, a longer filter will cause a slower transition to new reported values.

In most applications, the recommended setting is between 10 and 30 seconds. For highly turbulent flow conditions, a longer filter length is recommended to achieve more stable measurement results.

Direction Filter



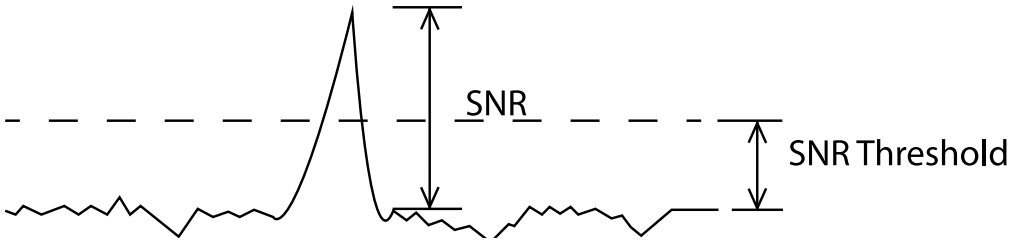
The Direction Filter parameter determines whether the instrument measures flow in both directions or restricts detection to a single flow direction (incoming or outgoing).

When set to *No Filter*, the radar measures velocity regardless of flow direction and reports both the velocity magnitude and the detected flow direction.

When set to *Incoming* or *Outgoing*, the instrument processes and reports only the selected flow direction. Radar returns corresponding to the opposite direction are ignored.

At monitoring sites where flow is expected to occur consistently in only one direction, it is strongly recommended to configure the Direction Filter accordingly. Restricting the measurement to the expected flow direction improves measurement stability and consistency by eliminating unwanted signal components.

SNR Threshold

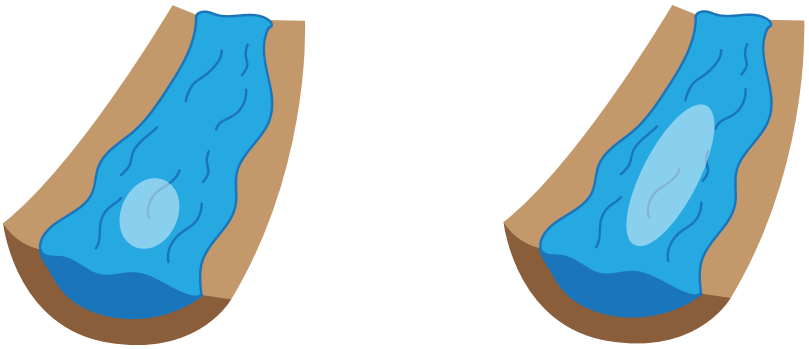


Defines the minimum signal-to-noise ratio (SNR), in dB, required for the radar to report a velocity measurement. Values below this threshold are ignored and zero velocity is reported.

A low SNR indicates that the radar signal is weak compared to the background noise , which can result in uncertain or incorrect velocity readings. Increasing this threshold helps filter out unreliable measurements. If you notice the radar reporting velocities when the water is not moving, check the SNR values during those periods. If they are consistently lower than during valid measurements, consider increasing the SNR threshold.

This parameter affects only surface velocity measurement, and not water level measurement.

Beam Width



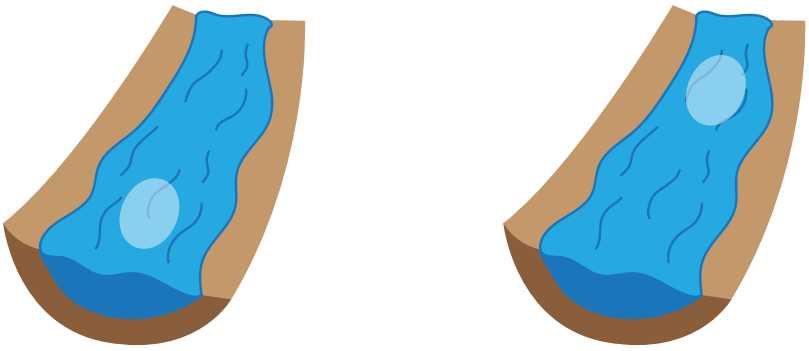
Picture 10. Narrow / wide radar beam width selection.

Beam Width defines the elevation spread of the radar beam and therefore how long a section of the water surface is analyzed. A narrower beam focuses on a shorter area close to the sensor, making it ideal when you want to exclude local disturbances such as waterfalls, cascades, or strong turbulence near the installation. A wider beam covers a longer surface area, allowing the radar to observe and average velocities over a broader section of the river or stream.

Because the radar reports an average velocity within the observed area, a wider beam is useful when surface velocities vary gradually, while a narrower beam provides more stable results in the presence of local disturbances.

The beam width of the water level radar is not affected by this parameter.

Beam Position



Picture 11. Near / far radar beam position selection.

The Beam Position parameter defines the measurement zone on the water surface (either near or far), placed along the radar beam, where the surface velocity is analyzed.

The default setting is *Near*, which configures the radar to measure velocities closer to the instrument. This setting is suitable for most standard installations.

In certain situations, the water surface closer to the radar may be more turbulent or disturbed than areas farther away. In such cases, setting the Beam Position to *Far* shifts the measurement zone further from the instrument, which may result in more stable and representative velocity readings.

The beam position of the water level radar is not affected by this parameter.

Output Units

Discharge Output Unit

This parameter defines the measurement unit in which the radar's discharge readings are transmitted over all communication interfaces, including RS-232 (NMEA-like protocol), RS-485 (Modbus), and SDI-12.

Select the unit that matches the requirements of your data logger, SCADA system, or telemetry platform to ensure proper scaling, interpretation, and integration of the measurement data.

Water Level Output Unit

This parameter defines the measurement unit in which the radar's water level readings are transmitted over all communication interfaces, including RS-232 (NMEA-like protocol), RS-485 (Modbus), and SDI-12.

Select the unit that matches the requirements of your data logger, SCADA system, or telemetry platform to ensure proper scaling, interpretation, and integration of the measurement data.

Surface Velocity Output Unit

This parameter defines the measurement unit in which the radar's surface velocity readings are transmitted over all communication interfaces, including RS-232 (NMEA-like protocol), RS-485 (Modbus), and SDI-12.

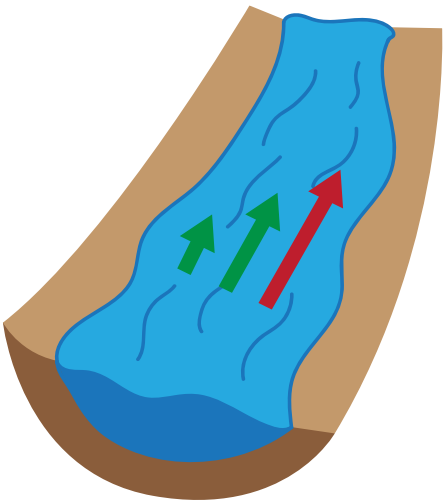
Select the unit that matches the requirements of your data logger, SCADA system, or telemetry platform to ensure proper scaling, interpretation, and integration of the measurement data.

Volume Output Unit

This parameter defines the measurement unit in which the radar's calculated totalized flow volume is transmitted over all communication interfaces, including RS-232 (NMEA-like protocol), RS-485 (Modbus), and SDI-12.

Select the unit that matches the requirements of your data logger, SCADA system, or telemetry platform to ensure proper scaling, interpretation, and integration of the measurement data.

Maximum Velocity



The Maximum Velocity setting defines the maximum expected surface flow velocity at the measurement site. The instrument will not report any velocity value higher than the configured maximum. Any detected value above this limit will be automatically rejected. This setting is useful for eliminating occasional spurious readings that may occur due to strong turbulence, debris passing through the radar beam, temporary signal disturbances, and environmental effects such as splashes or floating objects.

By setting a realistic maximum velocity based on site conditions, the reliability of the reported data can be improved.

> *Important:*
>
> The configured value should be slightly higher than the highest velocity expected at the site. If the value is set too low, valid high-flow measurements may be incorrectly discarded.

Minimum Distance

Defines the minimum valid distance between the radar and the water surface. Reflections detected closer than this value are ignored. This helps to eliminate unwanted signals from nearby objects such as bridge structures or mounting elements.

Important:

Set this value slightly below the shortest expected distance to the water surface.

Maximum Distance

Defines the maximum valid distance between the radar and the water surface. Reflections detected beyond this value are ignored. This prevents the instrument from processing signals outside the actual water surface area.

Important:

Set this value slightly above the longest expected distance to the water surface, considering water level variations.

Dynamic Flow Profiler

The Dynamic Flow Profiler (DFP) is an AI-based algorithm that improves measurement reliability under challenging conditions. When enabled, the instrument analyzes historical velocity data and learns the relationship between surface velocity and water level at the site. If the radar is temporarily unable to measure velocity due to low signal-to-noise ratio (SNR), smooth water surface, or other disturbances, DFP will estimate the velocity instead of reporting zero. The estimated value is calculated using previously recorded measurements and their correlation with the current water level. This function is especially useful at sites with very smooth water surfaces and weak radar reflections.

Important:

1. The instrument requires a training period to learn site-specific flow behavior. During this time, performance gradually improves.
2. If the radar is repositioned or installed at a new site, the DFP model must be reset. Click the Reset DFP button to clear previously learned data.

Power Management

Switching the device between continuous scanning mode and SDI-12 automatic sleep mode.

In continuous scanning mode, the device constantly performs measurements, which are transmitted over RS-232 interface and made available over Modbus, SDI-12, and analog interfaces. RS-232 measurement sen-

tences are transmitted when NMEA output is enabled. When awake, the device is available over RS-232 and Modbus interfaces, and is responding to commands on these interfaces.

In SDI-12 automatic sleep mode, the device remains in low-power sleep mode until SDI-12 measure command is received (**a**M!). While in sleep mode, the device will not be able to connect to Modbus or RS-232 interface. For the reconfiguration of the device which is set to operate in SDI-12 automatic sleep mode, it is necessary to power-cycle the device then use the PC configurator application to connect to the device within 1 minute after power-up. If there is no attempt to connect to the device over RS-232 within one minute, the device will automatically go back to the sleep mode. Additionally, it is possible to send the custom SDI-12 command **a**XGLWU! That will also wake up the device from the sleep mode, and keep it running for 1 minute to allow the PC application enough time to establish the connection to the device.

Operation Mode

Allows switching the device from operating mode to standby mode and back. While the device is in standby mode, it will communicate through communication interfaces, but no measurements will be made. In operating mode, the device continuously performs measurements. In standby mode, the power consumption is reduced.

WiFi

Enable WiFi

Enables on-baord WiFi access point which allows wireless connection to the device for re-configuration and data access. With WiFi access enabled, the device power consumption will be approx. 40 mA higher (at 12V).

WiFi Password

Set the WiFi access point password. When the device start connecting to the on-board WiFi access point, the instrument will ask for the conencting device to enter this password. The password must be at least 8 characters long.

LiDAR

LiDAR Mode

Enables or disables automatic scanning of the channel cross-section with the built-in LiDAR sensor. The available values are:

`off` = automatic scanning is disabled,
`automatic` = the instrument scans the channel cross-section periodically.

When automatic mode is enabled, the instrument performs a scan on every power-up and then repeats the scan at the interval defined by the LiDAR scan period. Each completed automatic scan replaces the stored channel profile, so the discharge calculation always uses the most recent cross-section. This is useful at sites where the channel bed changes over time, for example due to sediment transport or scouring.

When automatic mode is disabled, the channel profile changes only when a manual scan is performed and saved, or when the profile is edited manually.

Regardless of the selected mode, a scan can always be started manually, as described in Manual LiDAR Scan.

LiDAR Scan Period

Defines the interval, in minutes, between two consecutive automatic LiDAR scans. The valid range is between 1 and 32767 minutes, and the default value is 15 minutes.

This setting is used only when the LiDAR mode is set to *automatic*. Since each scan requires the LiDAR and its motor to run, more frequent scanning increases the average power consumption of the instrument. At sites with a stable channel bed, a long scan period is sufficient.

Left Limit Angle

Defines how far the LiDAR sweep extends towards the left side of the channel, in degrees measured from the vertical axis of the instrument. The default value of 70° corresponds to the full mechanical sweep range of the sensor, and values above 70° have no additional effect.

Reduce this value to exclude a part of the scanned sector from the measured channel profile, for example when a bridge structure, a bank wall, or vegetation on the left side of the channel is scanned instead of the channel bed.

Right Limit Angle

Defines how far the LiDAR sweep extends towards the right side of the channel, in degrees measured from the vertical axis of the instrument. The default value of 70° corresponds to the full mechanical sweep range of the sensor, and values above 70° have no additional effect.

Reduce this value to exclude a part of the scanned sector from the measured channel profile, for example when a bridge structure, a bank wall, or vegetation on the right side of the channel is scanned instead of the channel bed.

Manual LiDAR Scan

A single LiDAR scan can be started at any time, independently of the configured LiDAR mode. A manual scan takes 30 to 45 seconds to complete, and its result is not applied to the discharge calculation until it is reviewed and saved. The procedure is described in 7.3. Using LiDAR.

A manual scan can also be started remotely, over the servicing protocol (Start Manual LiDAR Scan), over Modbus (Block 3: Configuration Registers (R/W)), or over SDI-12 (Start Manual LiDAR Scan).

While a scan is in progress, no new scan can be started. Whether a scan is currently running, and whether it is an automatic or a manual one, is reported by the servicing protocol as described in Check LiDAR Scan Status.

6 Data Protocols

The Geolux FX200 Surface Velocity Radar supports the following communication protocols:

- NMEA-like protocol (RS-232). Continuously outputs discharge, water level, surface velocity, reflected signal strength, current tilt angle, and device temperature and relative humidity.
- Service protocol (RS-232). Used for instrument configuration and maintenance.
- Modbus RTU (RS-485). An industrial standard, supported by a wide range of third-party data loggers and control systems.
- SDI-12 protocol. Available for integration with SDI-12 compatible data loggers.

The following chapters describe the protocols in detail:

6.1. NMEA Protocol (RS-232)

NMEA protocol is based on the standard protocol family widely used by the navigation equipment. NMEA protocol is sentence oriented, and is capable of sending multiple sentences with different information. The sentence content is designated by the starting keyword which is different for each sentence type. NMEA sentences are terminated with the checksum which makes this protocol extremely reliable. NMEA protocol is single-direction protocol: data is only transmitted from the surface velocity radar.

The device automatically outputs NMEA sentences without any request or input required. The output rate is the same as internall processing rate, 10 samples per second. On low baud rates (4800, 9600, 14400 and 19200) the output rate is reduced to 5 samples per second since the low baud rate does not allow for 10 sps output.

At RS-232 interface the device periodically outputs following data sentences:

Discharge and total volume report

$DIS,<discharge>,<total_volume><checksum><CR><LF>*

- $DIS: The keyword sent on the beginning of each report. This sentence contains information about the discharge (flow).
- <discharge>: The discharge in selected measurement units, as decimal number.
- <total_volume>: The total accumulated volume, in selected measurement units, as decimal number.

Water level measurement report

$LVX,<water_level>,<distance_to_water>,<snr><checksum><CR><LF>*

- $LVX: The keyword sent on the beginning of each report. This sentence contains information about the measured water level.
- <water_level>: The water level in selected measurement units, as decimal number.
- <distance_to_water>: The distance to water in selected measurement units, as decimal number.
- <snr>: The signal-to-noise ratio of the returned signal, in dB.

Surface velocity measurement report

$VEL,<direction>,<velocity>,<snr>,<status><checksum><CR><LF>*

- $VEL: The keyword sent on the beginning of each report. This sentence contains information about the measured flow.
- <direction>: The direction of the flow, as detected by the instrument (1 incoming, -1 outgoing, 0 no flow).
- <velocity>: The absolute value of the surface flow velocity, in selected measurement units, as decimal number. If no flow is detected, the output will be zero.
- <snr>: The signal-to-noise ratio of the returned radar signal, in dB.
- <status>: The status indicator. If the device operates without any errors, the output will be zero. If non-zero, it is interpreted as error bitmask:

  Bit 0: forward tilt angle error (tilt angle outside of expected range),
  Bit 1: side tilt angle error (tilt is too big),
  Bit 2: radar co-processor update failed,
  Bit 3: surface velocity radar co-processor not responding,
  Bit 4: radar co-processor needs to be updated,
  Bit 5: tilt sensor is not working,
  Bit 6: water level co-processor not responding,
  Bit 7: DFP data corrupt, needs reset,
  Bit 8: power save mode is active,
  Bit 9: radar is initializing.

- <checksum>: Two uppercase hexadecimal digits containing the checksum value of the NMEA sentence.

Device status report

$STAT,<angle_elevation>,<angle_sideways>,<temperature>,<humidity><checksum><CR><LF>*

$STAT: The keyword sent to denote the device status report sentence.

- <angle_elevation>: The elevation angle, expressed in degrees, describes the tilt of the radar relative to the water surface. An elevation angle of 0° means the radar is oriented parallel to the water surface; an elevation angle of 90° means the radar is oriented perpendicular to the water surface.
- <angle_sideways>: The sideways tilt angle of the instrument. Negative values are tilted to the left, and positive to the right. The radar should be positioned so that this angle is close to zero.
- <temperature>: The temperature inside the instrument enclosure, in degrees Celsius.

- <humidity>: The relative humidity inside the instrument enclosure, in %.

NMEA Checksum Calculation

Each NMEA sentence ends with a checksum to ensure data integrity. The checksum (CSUM) is calculated as follows:

Start with a checksum value of 0.

Perform a bitwise XOR operation on all characters in the sentence between the $ and * symbols.

Do not include the $, *, the two checksum hex digits, or trailing CR/LF characters in the calculation.

If the input has no *, compute XOR over all characters after $.

The resulting checksum value is converted to two hexadecimal characters (0–9, A–F) and appended after the * symbol.

The following Python function shows how to calcualate the checksum:

```python
def nmea_checksum(nmea_sentence: str) -> str:
    s = nmea_sentence.rstrip("\\r\\n")

    # Remove leading '$' or '!' if present.
    if s and s[0] in "$!":
        s = s[1:]

    # Keep only payload part before checksum delimiter.
    s = s.split("*", 1)[0]

    checksum = 0
    for char in s.encode("ascii"):
        checksum ^= char

    return f"{checksum:02X}"
```

6.2. Servicing Protocol (RS-232)

The servicing protocol is used to read and modify the operating parameters of the flow meter. Device settings such as filtering parameters, measurement units selection, interface configuration, and other internal parameters are adjusted using this protocol.

Since the NMEA-like measurement protocol is unidirectional (measurement data is transmitted only from the device to the host), the servicing protocol provides independent two-way configuration and maintenance access on the same RS-232 interface when the device is awake. To simplify device setup and parameter adjustment, Geolux provides the Geolux Instrument Configurator software tool. End users typically do not need to interact directly with the servicing protocol, as all configuration tasks are performed through the Configurator application. A detailed description of the software is provided in 7. Geolux Instrument Configurator.

Each command must begin with the '#' character and be terminated with a newline character. All standard combinations of carriage return (CR) and line feed (LF) characters (CR, LF, or CR+LF) are accepted as valid line termination.

Reading All Parameters at Once

To read all parameters at once, send the following command to the device:

```
#get_info
```

The device will respond with a structured list of all current parameters and system information in the following format:

```
#device_type: 10003
#firmware: 1.2.3
#serial_number: 12345
#<parameter_name_1>: <value1>
#<parameter_name_2>: <value2>
...
#update_status: 0
```

The response includes device identification information (such as device type, firmware version, and serial number), followed by all configurable and status parameters. A complete description of all available parameters is provided later in this chapter.

Reading Single Parameter

Any configuration parameter can be read individually:

```
#get_<parameter_name>
```

The device will respond:

```
#<parameter_name>: <value>
```

Writing Single Parameter

Any configuration parameter that is not read-only can be changed by the following command format:

```
#set_<parameter_name>=<value>
```

The device will respond, depending on success or failure:

```
#set_<parameter_name>:OK
#set_<parameter_name>:ERR
```

Enumerated parameters accept both the string name and the numeric index, for example:

```
#set_direction=incoming
#set_direction=1
```

List of Parameters

Parameter Name	Get Command	Set Command
Maximum Velocity Treshold	Defines the upper bound for valid surface velocity measurements. Velocity readings above this threshold are rejected. The numeric value parameter is interpreted in the unit defined by `max_velocity_unit`. See Maximum Velocity for the detailed setting description.	
max_velocity	*#get_max_velocity*	*#set_max_velocity=<value>*
Maximum Velocity Unit	Defines the physical unit used by `max_velocity` command. The accepted units are: See Maximum Velocity for the detailed setting description. `mms` = mm/s, `ms` = m/s, `mph` = mph, `kmh` = km/h, `fps` = ft/s, `fpm` = ft/min, `cms` = cm/s.	
max_velocity_unit	*#get_max_velocity_unit*	*#set_max_velocity_unit=<value>*
Direction Filter	Defines the direction filter configuration. The accepted values are: See Direction Filter for the detailed setting description. `both` = measure both directions, `incoming` = incoming flow only, `outgoing` = outgoing flow only.	
direction	*#get_direction*	*#set_direction=<value>*
SNR Threshold	Defines the SNR threshold value for the surface velocity radar. See SNR Threshold for the detailed setting description.	
snr_threshold	*#get_snr_threshold*	*#set_snr_threshold=<value>*
Filter Length	Defines the number of seconds used for internal smoothing. Higher values reduce noise but increase response lag. The valid range is between 1 and 120. See Filter Length for the detailed setting description.	
filter_len	*#get_filter_len*	*#set_filter_len=<value>*

Discharge Unit	Defines the unit used for the discharge value in the SDI-12 measurement response, NMEA and on Modbus output. See Discharge Output Unit for the detailed setting description. The accepted units are: `m3/s` = m³/s, `m3/h` = m³/h, `l/s` = l/s, `l/h` = l/h, `cfs` = ft³/s, `gal/min` = gal/min, `ML/day` = ML/day.	
discharge_unit	*#get_discharge_unit*	*#set_discharge_unit=<value>*

Water Level Unit	Defines the unit used for the water level value in the SDI-12 measurement response, NMEA and on Modbus output. See Water Level Output Unit for the detailed setting description. The accepted units are: `mm` = millimeters, `cm` = centimeters, `m` = meters, `ft` = feet, `in` = inches.	
level_unit	*#get_level_unit*	*#set_level_unit=<value>*

Velocity Output Unit	Defines the unit used for the velocity value in the SDI-12 measurement response, NMEA and on Modbus output. See Surface Velocity Output Unit for the detailed setting description. The accepted units are: `mms` = mm/s, `ms` = m/s, `mph` = mph, `kmh` = km/h, `fps` = ft/s, `fpm` = ft/min, `cms` = cm/s.	
units	*#get_units*	*#set_units=<value>*

Volume Output Unit	Defines the unit used for the total flow volume value in the SDI-12 measurement response, NMEA and on Modbus output. See Volume Output Unit for the detailed setting description. The accepted units are: `m3` = cubic meters, `l` = liters, `ft3` = cubic feet, `gal` = galons, `ML` = mega (million) liters.	
volume_unit	*#get_volume_units*	*#set_volume_units=<value>*

Beam Width	Defines the radar beam width pattern for the surface velocity radar. Wide beam covers a larger area; narrow beam is more focused and rejects off-axis targets. The accepted values are: See Beam Width for the detailed setting description. `wide`, `narrow`.	
beam_width	*#get_beam_width*	*#set_beam_width=<value>*

Beam Offset	Defines the near or far measurement zone placed along the radar beam used for target detection, for the surface velocity radar. The accepted values are: See Beam Position for the detailed setting description. `near`, `far`.	
beam_offset	*#get_beam_offset*	*#set_beam_offset=<value>*

DFP Enable	Defines the enable status of Dynamic Flow Profiler algorithm, which tracks and accumulates flow history for improved accuracy in low-flow conditions. The accepted values are: See Dynamic Flow Profiler for the detailed setting description. `off`, `on`.	
use_dfp	*#get_use_dfp*	*#set_use_dfp=<value>*

Radar Height	Gets / sets the vertical distance between the instrument and the water level reference point (gauge zero). The instrument uses this value to calculate and report the water level relative to the gauge zero. The corresponding measurement unit is defined by `radar_height_unit` parameter.	
radar_height	*#get_radar_height*	*#set_radar_height=<value>*

Radar Height Unit	Defines the measurement unit for the `radar_height` parameter. The accepted units are: `mm` = millimeters, `cm` = centimeters, `m` = meters, `ft` = feet, `in` = inches.	
radar_height_unit	*#get_radar_height_unit*	*#set_radar_height_unit=<value>*

Minimum Valid Distance	Defines the minimum valid distance to water surface. Reflections detected closer than this distance are ignored. Use this parameter to exclude near-field returns e.g., mounting hardware, bridge elements, or other objects close to the sensor. The corresponding measurement unit is defined by `min_distance_unit` parameter. See Minimum Distance for the detailed setting description.	
min_distance	*#get_min_distance*	*#set_min_distance=<value>*

Minimum Valid Distance Unit	Defines the measurement unit used by `min_distance` parameter. The accepted units are: See Minimum Distance for the detailed setting description. `mm`, `cm`, `m`, `ft`, `in`.	
min_distance_unit	*#get_min_distance_unit*	*#set_min_distance_unit=<value>*

Maximum Valid Distance	Gets / sets the maximum valid distance to water surface. Reflections detected beyond this distance are ignored. Use this parameter to exclude far-field returns outside the intended measurement zone e.g., opposite bank, structures, or background targets. The corresponding measurement unit is defined by `max_distance_unit` parameter. See Maximum Distance for the detailed setting description.	
max_distance	*#get_max_distance*	*#set_max_distance=<value>*

Maximum Valid Distance Unit	Gets / sets the measurement unit used by `max_distance` parameter. The accepted units are: See Maximum Distance for the detailed setting description. `mm`, `cm`, `m`, `ft`, `in`.	
max_distance_unit	*#get_max_distance_unit*	*#set_max_distance_unit=<value>*

RS-232 Baud Rate	Gets / sets the baud rate of the RS-232 serial interface. The change takes effect immediately. The accepted values for baud rates are: See RS-232 Baud Rate for the detailed setting description. `4800`, `9600`, `14400`, `19200`, `38400`, `57600`, `115200`.	
baud_rate	*#get_baud_rate*	*#set_baud_rate=<value>*

Modbus Baud Rate	Gets / sets the baud rate of the RS-485/Modbus serial interface. The change takes effect immediately. The accepted values for baud rates are: See RS-485 Baud Rate for the detailed setting description. 4800, 9600, 14400, 19200, 38400, 57600, 115200.	
modbus_baud_rate	*#get_modbus_baud_rate*	*#set_modbus_baud_rate=<value>*

Modbus Device ID	Gets / sets the Modbus RTU slave address. Must be unique on the RS485 bus. The accepted value range for Modbus RTU slave address is 1 - 247. See Modbus ID for the detailed setting description.	
modbus_id	*#get_modbus_id*	*#set_modbus_id=<value>*

Modbus Serial Line Parity	Gets / sets the serial line parity for RS-485 / Modbus connection. The change takes place immediately. The accepted values are: See Modbus Parity for the detailed setting description. none, odd, even.	
modbus_parity	*#get_modbus_parity*	*#set_modbus_parity=<value>*

Modbus Serial Line Stop Bits	Gets / sets the number of stop bits for the serial line of RS-485 / Modbus connection. The change takes place immediately. The accepted values are: See RS-485 Stop Bits for the detailed setting description. one, two.	
modbus_stopbits	*#get_modbus_stopbits*	*#set_modbus_stopbits=<value>*

SDI-12 Address	Address of the device on the SDI-12 bus. Valid range is 0-61, which will be converted to characters 0-9, A-Z and a-z. See SDI-12 ID for the detailed setting description.	
sdi_id	*#get_sdi_id*	*#set_sdi_id=<value>*

Disable NMEA Output	Gets / sets the parameters that disables NMEA sentence output on RS-232. The accepted values are: off, on.	
disable_nmea	*#get_disable_nmea*	*#set_disable_nmea=<value>*

Analog Output Type	Gets / sets the which device measurement will be used to drive the analog output. See 4 - 20 mA Output Type for the detailed setting description. The accepted values are: `none`, `discharge`, `level`, `distance`, `velocity`, `volume`.	
analog_output	*#get_analog_output*	*#set_analog_output=<value>*

Analog Output Minimum Value	Gets / sets the measurement value that corresponds to the minimum output on the analog current interface (4 mA). The corresponding measurement unit is defined by `analog_min_unit` parameter. See 4 - 20 mA Minimum Value for the detailed setting description.	
analog_min	*#get_analog_min*	*#set_analog_min=<value>*

Analog Output Minimum Unit	Gets / sets the measurement unit for the value defined by `analog_min` parameter. The accepted units depend on the `analog_output` parameter. If the `analog_output` is set to `discharge`: `m3/s` = m³/s, `m3/h` = m³/h, `l/s` = l/s, `l/h` = l/h, `cfs` = ft³/s, `gal/min` = gal/min, `ML/day` = ML/day. If the `analog_output` is set to `level` or `distance`: `mm` = millimeters, `cm` = centimeters, `m` = meters, `ft` = feet, `in` = inches. If the `analog_output` is set to `velocity`: `mms` = mm/s, `ms` = m/s, `mph` = mph, `kmh` = km/h, `fps` = ft/s, `fpm` = ft/min, `cms` = cm/s. If the `analog_output` is set to `volume`: `m3` = cubic meters, `l` = liters, `ft3` = cubic feet, `gal` = galons, `ML` = mega (million) liters. See 4 - 20 mA Minimum Value for the detailed setting description.	
analog_min_unit	*#get_analog_min_unit*	*#set_analog_min_unit=<value>*

Analog Output Maximum Value	Gets / sets the measurement value that corresponds to the maximum output on the analog current interface (20 mA). The corresponding measurement unit is defined by `analog_max_unit` parameter. See 4 - 20 mA Maximum Value for the detailed setting description.	
analog_max	*#get_analog_max*	*#set_analog_max=<value>*

Analog Output Maximum Unit	Gets / sets the measurement unit for the value defined by `analog_max` parameter. The accepted units depend on the `analog_output` parameter. If the `analog_output` is set to `discharge`: `m3/s` = m³/s, `m3/h` = m³/h, `l/s` = l/s, `l/h` = l/h, `cfs` = ft³/s, `gal/min` = gal/min, `ML/day` = ML/day. If the `analog_output` is set to `level` or `distance`: `mm` = millimeters, `cm` = centimeters, `m` = meters, `ft` = feet, `in` = inches. If the `analog_output` is set to `velocity`: `mms` = mm/s, `ms` = m/s, `mph` = mph, `kmh` = km/h, `fps` = ft/s, `fpm` = ft/min, `cms` = cm/s. If the `analog_output` is set to `volume`: `m3` = cubic meters, `l` = liters, `ft3` = cubic feet, `gal` = galons, `ML` = mega (million) liters. See 4 - 20 mA Maximum Value for the detailed setting description.	
analog_max_unit	*#get_analog_max_unit*	*#set_analog_max_unit=<value>*

Configure Power Management Mode (SDI-12 Sleep Mode)	Gets / sets the automatic SDI-12 sleep mode parameter. When sleep is enabled, the device enters low-power sleep between SDI-12 transactions. When disabled, the sensor runs continuously and is ready at all times. The accepted values are: See Power Management for the detailed setting description. `on` = sleep enabled, `off` = sleep disabled (continuous scanning).	
sdi_sleep	*#get_sdi_sleep*	*#set_sdi_sleep=<value>*

Configure Operation Mode	Gets / sets the operation mode parameter. When the operation mode is set to standby, the device will go to standby mode. In standby mode, it will communicate on all communication interfaces, but the radar transmitters will be turned off to reduce the power, and the instrument will not be making any measurements. The accepted values are: See Operation Mode for the detailed setting description. `off` = normal operation, `on` = standby operation.	
power_save	*#get_power_save*	*#set_power_save=<value>*

Configure Wi-Fi Operation Mode	Gets / sets the WiFi hotspot status: `off` = default setting, the WiFi access point on the instrument is disabled, `on` = WiFi hotspot on the instrument is active. See Enable WiFi for the detailed setting description.	
enable_wifi	*#get_enable_wifi*	*#set_enable_wifi=<value>*

Configure Wi-Fi Password	Gets / sets the WiFi hotspot password. The password must be at least 8 characters long. See WiFi Password for the detailed setting description.	
wifi_password	*#get_wifi_password*	*#set_wifi_password=<value>*

Configure LiDAR Mode	Gets / sets the automatic LiDAR scanning mode. The accepted values are: `off` = automatic scanning is disabled, `automatic` = the channel cross-section is scanned periodically. See LiDAR Mode for the detailed setting description.	
lidar_mode	*#get_lidar_mode*	*#set_lidar_mode=<value>*

Configure LiDAR Scan Period	Gets / sets the interval between two consecutive automatic LiDAR scans, in minutes. The valid range is between 1 and 32767 minutes. This parameter is used only when `lidar_mode` is set to `automatic`. See LiDAR Scan Period for the detailed setting description.	
lidar_scan_period_min	*#get_lidar_scan_period_min*	*#set_lidar_scan_period_min=<value>*

Configure LiDAR Left Limit Angle	Gets / sets how far the LiDAR sweep extends towards the left side of the channel, in degrees measured from the vertical axis of the instrument. The default value of 70 corresponds to the full mechanical sweep range of the sensor. See Left Limit Angle for the detailed setting description.	
lidar_left_limit_angle	*#get_lidar_left_limit_angle*	*#set_lidar_left_limit_angle=<value>*

Configure LiDAR Right Limit Angle	Gets / sets how far the LiDAR sweep extends towards the right side of the channel, in degrees measured from the vertical axis of the instrument. The default value of 70 corresponds to the full mechanical sweep range of the sensor. See Right Limit Angle for the detailed setting description.	
lidar_right_limit_angle	*#get_lidar_right_limit_angle*	*#set_lidar_right_limit_angle=<value>*

Configure Fixed Mounting Angle	Override measured forward angle with a fixed value (degrees). Changing this parameter will affect the reported velocity as cosine correction behaviour will be changed. Use this parameter only during tow-tank testing. Set 0 to use real measured angle. The accepted value range for fixed angle is: 0 - 89.	
fixed_angle	*#get_fixed_angle*	*#set_fixed_angle=<value>*

Show Data on Error	Continue reporting velocity data even when angle errors are detected. The accepted values are: `off` = clear velocity on error (send 0 as velocity instead of the measured value), `on` = keep velocity.	
show_data_on_error	*#get_show_data_on_error*	*#set_show_data_on_error=<value>*

List of Commands

Perform Soft Reset

Restarts the device firmware. All settings are retained (stored in non-volatile memory). The device will be unresponsive for a few seconds after issuing this command.

Command: `#reset`

Reset DFP

Clears all accumulated DFP history data. Use this after a significant change in site conditions or sensor placement.

Command: `#reset_dfp`

Factory Reset

Restores all settings to their factory defaults and restarts the device. This is irreversible.

Command: `#factory_reset`

Set Staff Gauge Reading

Takes current reading from the staff gage to automatically adjust the radar_height parameter according to the given value and current measurement of the distance to water surface.

Command: `#set_staff_gauge=<value>`

Reset Totalizer

Resets the volume totalizer which keeps track of the total accumulated volume over time.

Command: `#reset_totalizer`

Start Manual LiDAR Scan

Starts a single LiDAR scan of the channel cross-section. The command can be used regardless of the configured LiDAR mode. The scan takes 30 to 45 seconds to complete. If a scan is already running, the command is ignored.

See Manual LiDAR Scan for the detailed description.

Command: `#start_manual_scan`

Check LiDAR Scan Status

The status of the LiDAR scanning is reported in the `#get_info` response, in the `#lidar_scan_status` line:

`off` = no scan is currently running,
`manual` = a manually started scan is in progress,
`automatic` = an automatic scan is in progress.

Send `#get_info` and read the reported value to check whether a scan is still ongoing:

```
#lidar_scan_status: manual
```

Reading LiDAR Scan

The channel cross-section produced by the most recent automatic or manual LiDAR scan can be read out over the servicing protocol. This is the same data shown on the **LiDAR** page of the configurator, see 7.3. Using LiDAR.

Unlike the simple text parameters, the scan is a relatively large data set (up to 1000 measured points). To transfer it efficiently it is serialized to a compact binary form, compressed, and returned in a single dedicated response rather than as a plain `#<name>: <value>` line.

Request

```
#read_lidarscan=manual
```

Use `manual` for the most recent manual scan, or `auto` / `automatic` for the most recent automatic scan.

Response

The device replies with a single line:

```
#lidarscan=<length>,<data>,<crc>\r\n
```

- `<length>` – the number of bytes in the `<data>` field, as a decimal number.
- `<data>` – the scan payload (see below), byte-stuffed so that it does not contain any character that would break the line-based protocol.
- `<crc>` – a four-character uppercase hexadecimal CRC-16 (see CRC) computed over the **compressed payload before byte-stuffing**.

If the scan cannot be prepared (for example, no scan is stored or the device is out of memory), the device responds with:

`#Lidarscan:ERR`

Payload encoding

The `<data>` field is produced in three steps:

1. **Serialization.** The scanned profile is encoded as a CBOR (RFC 8949) map. The relevant keys are:

Key	Type	Description
`unt`	text string	Unit of the coordinates. Always `mm`.
`x`	array of float32	X coordinates of the profile points, left to right.
`y`	array of float32	Y coordinates of the profile points. Values are negative, measured downward from the sensor.
`xof`, `yof`	float32	X and Y offsets applied to the coordinates. `0` for a raw scan.
`lvl`, `kfc`, `shp`, `sld`, `sdp`, `shn`	array / bool / uint	Discharge and channel-shape metadata. Empty / zero for a raw scan. **Can be ignored.**

   The `x` and `y` arrays have the same length and together define the cross-section points.

2. **Compression.** The CBOR byte stream is compressed with Zstandard (zstd).
3. **Byte-stuffing.** The compressed bytes are escaped so that no reserved character remains in the payload. Each reserved byte is replaced by the `=` character followed by a single digit:

Original byte	Escaped as
`#` (0x23)	`=0`
`=` (0x3D)	`=1`
`,` (0x2C)	`=2`
CR (0x0D)	`=3`
LF (0x0A)	`=4`
NUL (0x00)	`=5`
`$` (0x24)	`=6`

   To recover the compressed payload, reverse this mapping: whenever a `=` is found, read the following digit and substitute the corresponding byte.

CRC

The CRC is a CRC-16 with polynomial `0x8005`, initial value `0`, and input and output reflected. This is the algorithm commonly known as **CRC-16/ARC**. It is computed over the compressed payload **before** byte-stuffing (i.e. over the same bytes obtained after un-escaping the received `<data>` field).

Python example

The following example reads the scan over the serial port and saves the profile as a CSV file with `x` and `y` columns in millimeters. It requires the `pyserial`, `zstandard`, and `cbor2` packages (`pip install pyserial zstandard cbor2`).

PYTHON

```python
import csv
import serial
import zstandard
import cbor2

PORT = "COM3"          # serial port of the FX200 (e.g. "/dev/ttyUSB0")
BAUD = 115200
OUTPUT_CSV = "lidar_profile.csv"

# Byte-stuffing table: index -> original byte (see the escaping table above).
SUBS = b"#=,\r\n\x00$"

def unescape(data: bytes) -> bytes:
    out = bytearray()
    i = 0
    while i < len(data):
        if data[i] == ord("="):
            i += 1
            idx = data[i] - ord("0")
            if 0 <= idx <= 6:
                out.append(SUBS[idx])
        else:
            out.append(data[i])
        i += 1
    return bytes(out)

def crc16_arc(data: bytes) -> int:
    """CRC-16/ARC: polynomial 0x8005, init 0, reflected in/out."""
    crc = 0
    for byte in data:
        crc ^= byte
        for _ in range(8):
            if crc & 1:
                crc = (crc >> 1) ^ 0xA001   # 0xA001 = bit-reversed 0x8005
            else:
                crc >>= 1
    return crc

with serial.Serial(PORT, BAUD, timeout=5) as port:
    port.write(b"#read_lidarscan=manual\r\n")

    # Read lines until the "#lidarscan=" response arrives.
    line = b""
    while True:
        line = port.readline()
        if not line:
            raise TimeoutError("no response from device")
        if line.startswith(b"#lidarscan"):
            break

if line.startswith(b"#lidarscan:ERR"):
    raise RuntimeError("device could not provide a LiDAR scan")
```

```
53
54  body = line[len("#lidarscan="):].rstrip(b"\r\n")
55  length_str, rest = body.split(b",", 1)
56  length = int(length_str)
57  escaped = rest[:length]
58  crc_received = int(rest[length + 1:], 16)
59
60  compressed = unescape(escaped)
61  if crc16_arc(compressed) != crc_received:
62      raise ValueError("CRC mismatch")
63
64  cbor_bytes = zstandard.ZstdDecompressor().decompress(compressed)
65  profile = cbor2.loads(cbor_bytes)
66
67  with open(OUTPUT_CSV, "w", newline="") as f:
68      writer = csv.writer(f)
69      writer.writerow(["x_mm", "y_mm"])
70      for x, y in zip(profile["x"], profile["y"]):
71          writer.writerow([x, y])
72
73  print(f"Saved {len(profile['x'])} points to {OUTPUT_CSV}")
```

6.3. Modbus Protocol (RS-485)

Protocol Overview

The FX200 instrument responds to Modbus RTU requests over the RS-485 data line. Communication parameters such as baud rate, parity, stop bits, and device ID are configured through the PC application or via Modbus itself. The default settings are:

Baud rate: 9600
Data bits: 8
Parity: Even
Stop bits: 1
Slave ID: 1

The Modbus slave address is latched at startup. After changing register `0x98` (register number `153`), continue using the **old address until the device is reset**. Reading that register returns the newly configured address, even while requests must still use the old address. Settings are saved asynchronously; allow the save to complete before resetting or removing power. After reset, use the new address. This applies whether the setting is changed through Modbus, RS-232, or SDI-12.

Modbus is a request-response protocol. A master device (such as a data logger or PLC) sends requests, and the slave device (the FX200 instrument) responds. The master can either read the value of one or more holding registers (using Modbus function code 0x03), or write the value of a holding register (function code 0x06), or write more holding registers (function code 0x10). All registers in the FX200 are Holding Registers (40001-range in Modbus convention).

Important:
Register Address vs. Register Number

The Modbus protocol uses zero-based register addressing in the actual request and response frames transmitted on the wire. This means that the first holding register is addressed as 0, the second as 1, and so on. However, many tools and documentation refer to one-based register numbers.

register number = register address + 1

For example, the first register has address **0** and register number **1**. The modpoll utility and most Modbus configuration tools use the 1-based register number.

To further complicate things, certain PLC software packages use 40001-based addressing. In this convention, holding registers are labeled starting from 40001, so that the register number 40001 corresponds to the first holding register at address 0. It is important to understand that the leading "4" in addresses such as 40001 does not form part of the address transmitted over Modbus. Instead, it indicates the register type (4 = Holding Register, not to be confused with function 3 for reading holding registers). The actual address sent in the Modbus frame is zero-based and does not include this prefix.

PLC register number = register address + 40001

Modbus is a very old protocol, so please do not judge its quirks too harshly.

Throughout this document, both the zero-based register address and the corresponding one-based register number are shown for clarity.

Each Modbus register holds a single 16-bit (2-byte) value. The FX200 uses the following data types:

- uint16 / int16 (1 register)

  A single 16-bit value occupying one register. Unsigned values (uint16) range from 0 to 65535. Signed values (int16) range from -32768 to 32767.

- uint32 (2 registers)

  A 32-bit unsigned integer spanning two consecutive registers. The low word is in the first register of the pair, and the high word is in the second, regardless of whether the starting address is even or odd.

  > *Example:*
  >
  > Value 0x00000003 would be transmitted as pair of registers 0x0003 and 0x0000.

- float32 (2 registers)

  A 32-bit IEEE 754 floating-point value spanning two consecutive registers. The encoding uses little-endian word order (low word first, high word second), which is common in industrial Modbus devices.

Example: The Value 1.5 is encoded per IEEE 754 as 0x3FC00000, and would be transmitted as pair of registers 0x0000 and 0x3FC0.
A fixed endianness check register at address 62 (register 63) always returns the value **-123.265625**. Read this register to verify your decoder handles the byte/word order correctly. The expected raw register values are 0x8800, 0xC2F6.

- string[n] (n/2 registers)

  The string data type allows a Modbus master to read and write textual data on the device. The parameter **n** (must be an even number) defines the maximum number of characters the string can contain. If the stored text is shorter than **n** characters, the remaining positions are filled with null characters (ASCII value 0). Each Modbus register is 16 bits (2 bytes) long and stores two characters, so the total number of registers used is **n**/2. Characters are stored in big-endian order within each register, so that the high byte contains the first (preceding) character and the low byte contains the second character.

Reading Registers with modpoll

modpoll is a command-line utility for reading and writing Modbus registers from a PC.

To read the first four float32 measurement values (discharge, water level, surface velocity, flow direction) from the FX200, run modpoll with the following arguments:

```
modpoll -m rtu -a 1 -r 1 -c 8 -t 4:float -e -b 9600 -d 8 -p even -s 1 COM3
```

To read a single uint16 configuration register (e.g., measurement unit at register 130):

```
modpoll -m rtu -a 1 -r 130 -c 1 -t 4 -b 9600 -d 8 -p even -s 1 COM3
```

These examples were tested with modpoll version 3.10.

Flag	Meaning
`-a 1`	Slave address 1
`-r 1`	Start at register number 1 (address 0x00)
`-c 8`	Read 8 registers (= 4 float32 values)
`-t 4:float`	Interpret as 32-bit float
`-e`	Little-endian word order (low word first)
`-b 9600`	Baud rate 9600
`-d 8`	8 data bits
`-p even`	Even parity
`-s 1`	1 stop bit
`COM3`	Serial port (use /dev/ttyUSB0 on Linux)

Report Server ID (Function 0x11)

Use Modbus function **17 (0x11), Report Server ID**, to identify the instrument. This is a command, not a holding register. The request format and complete device-specific response layout are described below.

The RTU request contains the slave address, function byte `0x11`, and the two-byte CRC; there are no register address or quantity fields. For slave address 1:

```
01 11 C0 2C
```

The response is 29 bytes long, including CRC:

Byte offset (zero-based)	Content
0	Active slave address
1	Function `0x11`
2	Payload byte count `0x18` (24 bytes)
3	Server identifier `0x47` (ASCII `G`)
4	Run indicator `0xFF`; firmware currently reports this constant, not the measurement error bitmask
5	Geolux identification format version `0x01`
6	ASCII space (`0x20`)
7–20	14 ASCII bytes: `GEOLUX FX-200`, padded with trailing spaces to 14 bytes
21–24	Serial number, uint32, most significant byte first
25–26	Firmware version, uint16, most significant byte first: `major × 100 + minor × 10 + patch`
27–28	RTU CRC, low byte first

Example response for serial number `0x12345678` and firmware 1.0.8:

```
01 11 18 47 FF 01 20 47 45 4F 4C 55 58 20 46 58 2D 32 30 30 20 12 34 56 78 00 6C D6 50
```

The serial number and firmware version in this command use most-significant-byte-first encoding; do not apply the holding-register low-word-first rule to this byte payload. Bootloader and hardware versions are not included in this response.

Configuration Writes and Read-Back

Read configuration values with function `0x03`. Write a single 16-bit setting with `0x06` or `0x10`. For writable 32-bit settings, use one `0x10` request containing both registers in ascending address order: low word first, then high word. Writing only the low word does not apply the new value. Read both words together with `0x03` to verify the result.

Read/write settings return their configured values, including nonzero values. In particular, receiver mode at address `0x8F` (register number 144) and analog output selection at `0x9B` (register number 156) are live settings, not reserved addresses. Invalid values for 16-bit settings return exception `0x03` (illegal data value). Reserved configuration addresses return zero on reads and exception `0x02` on writes; write-only command registers return exception `0x02` on reads.

The Modbus slave ID is an exception to immediate application: address `0x98` (register number 153) reads back the new setting immediately, but the old slave address remains active until reset.

Registers Overview

The register address space is divided into three distinct blocks:

- **Block 1** provides high-resolution measurement data as 32-bit floats. This is the recommended block for modern data loggers and SCADA systems.
- **Block 2** provides the same measurement data encoded as 16-bit integers for compatibility with legacy data loggers that cannot decode float32 values. Values are scaled as documented in the register table.
- **Block 3** contains all user-configurable device parameters. Some addresses are reserved. Reserved configuration addresses read as zero and reject writes with exception `0x02` (illegal data address). Command registers are write-only and reject reads with the same exception.

Block	Address Range	Register Range	Purpose
1	0 - 63 0x00 - 0x3F	1 - 64 0x01 - 0x40	Measurement data (float32, read-only)
2	64 - 127 0x40 - 0x7F	65 - 128 0x41 - 0x80	Measurement data (int16 compatibility, read-only)
3	128 - 194 0x80 - 0xC2	129 - 195 0x81 - 0xC3	Configuration parameters (read/write)

Block 1: Measurement Registers (float32)

Address(es)	Reg #	Data Type	Description
0, 1 0x00, 0x01	1, 2 0x01, 0x02	float32	Flow discharge in the selected measurement unit. The unit is set by configuration register `0x83`.
2, 3 0x02, 0x03	3, 4 0x03, 0x04	float32	Water level in the selected level measurement unit. The unit is set by configuration register `0x82`.
4, 5 0x04, 0x05	5, 6 0x05, 0x06	float32	Absolute surface velocity in the selected measurement unit (always >= 0). The unit is set by configuration register `0x81`.
6, 7 0x06, 0x07	7, 8 0x07, 0x08	float32	Flow direction: `+1.0` = incoming (towards sensor), `-1.0` = outgoing (away from sensor), `0.0` = no flow detected.
8, 9 0x08, 0x09	9, 10 0x09, 0x0A	float32	Discharge in m³/s. Not unit-converted, always in m³/s.
10, 11 0x0A, 0x0B	11, 12 0x0B, 0x0C	float32	Water level in m. Not unit-converted, always in m.

Address(es)	Reg #	Data Type	Description
12, 13 0x0C, 0x0D	13, 14 0x0D, 0x0E	float32	Surface velocity in m/s. The value can be positive or negative. Positive values – incoming flow, negative values – outgoing flow. Not unit-converted, always in m/s.
14, 15 0x0E, 0x0F	15, 16 0x0F, 0x10	float32	Signal-to-noise ratio (SNR) of the surface velocity radar, in dB. Higher values indicate stronger signal quality.
16, 17 0x10, 0x11	17, 18 0x11, 0x12	float32	Signal-to-noise ratio (SNR) of the water level radar in dB. Higher values indicate stronger signal quality.
18, 19 0x12, 0x13	19, 20 0x13, 0x14	float32	Forward tilt angle of the sensor in degrees. Ideally the sensor should be mounted at 0°.
20, 21 0x14, 0x15	21, 22 0x15, 0x16	float32	Side tilt angle of the sensor in degrees. This angle should be very close to 0°.
22, 23 0x16, 0x17	23, 24 0x17, 0x18	float32	Device temperature in units of degrees Celsius.
24, 25 0x18, 0x19	25, 26 0x19, 0x1A	float32	Relative humidity inside the device enclosure in percent (%).
26, 27 0x1A, 0x1B	27, 28 0x1B, 0x1C	uint32	Status flags (0 = all OK), as error bitmask: Bit 0: forward tilt angle error (tilt angle outside of expected range). Bit 1: side tilt angle error (tilt is too big). Bit 2: radar co-processor update failed. Bit 3: surface velocity radar co-processor not responding. Bit 4: radar co-processor needs to be updated. Bit 5: tilt sensor is not working. Bit 6: water level radar co-processor not responding. Bit 7: DFP data corrupt, needs reset. Bit 8: standby mode is active. Bit 9: radar is initializing.
28 0x1C	29 0x1D	uint16	Firmware upgrade status: `0` = idle, `1` = pending, `2` = active, `3` = done, `4` = failed.
29, 30 0x1D, 0x1E	30, 31 0x1E, 0x1F	float32	Total accumulated volume, in selected units. Low word at `0x1D`, high word at `0x1E`; read both in one request.
31, 32 0x1F, 0x20	32, 33 0x20, 0x21	float32	Total accumulated volume, in m³. Low word at `0x1F`, high word at `0x20`; read both in one request. Exported as float32 (about 7 significant decimal digits), while the internal accumulator uses double precision.
33, 34 0x21, 0x22	34, 35 0x22, 0x23	float32	Distance to water, in the selected level measurement unit. The unit is set by configuration register `0x82`.
35, 36 0x23, 0x24	36, 37 0x24, 0x25	float32	Distance to water in mm. Not unit-converted, always in mm.

Address(es)	Reg #	Data Type	Description
37 0x25	38 0x26	uint16	**LiDAR Scan Status (FX200 only).** Read-only live status: `0` = idle, `1` = automatic scan running, `2` = manual scan running. Idle means no scan is currently running; automatic scanning may still be enabled. This is the same status as RS-232 `#lidar_scan_status`. Reads use function `0x03`; writes return exception `0x02`. This address remains reserved on FX100 and VX60.
38 - 61 0x26-0x3D	39 - 62 0x27-0x3E		Reserved. Reads as 0.
62, 63 0x3E, 0x3F	63, 64 0x3F, 0x40	float32	Endianness check: This register will always return constant `-123.265625`, as a pair of values `0x8800`, `0xC2F6`. Use this to verify float32 decoding in your system.

Block 2: Measurement Registers (int16)

These registers provide the same measurement data as Block 1, but encoded as 16-bit integers for compatibility with data loggers that do not support float32 decoding. All values are read-only.

Address(es)	Reg #	Data Type	Description
64 0x40	65 0x41	uint16	Discharge, integer part, in selected measurement unit. Example: if discharge is 12.3456, this register returns 12.
65 0x41	66 0x42	uint16	Discharge, fractional part (4 decimal digits), in selected measurement units. Example: if discharge is 12.3456 this register returns 3456; if discharge is 12.34 then the register returns 3400.
66 0x42	67 0x43	int16	Water level, integer part, in selected measurement unit. Example: if the level is 12.3456, this register returns 12.
67 0x43	68 0x44	int16	Water level, fractional part (4 decimal digits), in selected measurement units. Example: if the level is 12.3456 this register returns 3456; if the level is 12.34 then the register returns 3400.
68 0x44	69 0x45	uint16	Absolute surface velocity, integer part, in selected measurement unit. Example: if velocity is 12.3456, this register returns 12.
69 0x45	70 0x46	int16	Absolute surface velocity, fractional part (4 decimal digits), in selected measurement units. Example: if velocity is 12.3456 this register returns 3456; if velocity is 12.34 then the register returns 3400.
70 0x46	71 0x47	int16	Flow direction: `+1` = incoming, `-1` = outgoing, `0` = no flow.
71 0x47	72 0x48	int16	Surface velocity in mm/s, integer part only. Positive values – incoming flow, negative values – outgoing flow. Clamped to range -16000 .. +16000 mm/s.
72 0x48	73 0x49	int16	Distance to water in mm, integer part only.

Address(es)	Reg #	Data Type	Description
73 0x49	74 0x4A	int16	Signal-to-noise ratio (SNR), of the returned radar signal of the surface velocity radar in dB integer part only. Higher values indicate stronger signal quality.
74 0x4A	75 0x4B	int16	Signal-to-noise ratio (SNR), of the returned radar signal of the water level radar in dB integer part only. Higher values indicate stronger signal quality.
75 0x4B	76 0x4C	int16	Forward tilt angle of the sensor in degrees. Ideally the sensor should be mounted at 0°.
76 0x4C	77 0x4D	int16	Side tilt angle of the sensor in degrees. This angle should be very close to 0°.
77 0x4D	78 0x4E	int16	Device temperature in degrees Celsius × 100. Example: if temperature is 23.456 °C then this register holds 2345.
78 0x4E	79 0x4F	int16	Relative humidity inside the device enclosure in percent (%) × 100. Example: if relative humidity is 24.17 % then this register holds 2417.
79, 80 0x4F, 0x50	80, 81 0x50, 0x51	uint32	Status flags (same encoding as Block 1).
81 0x51	82 0x52	uint16	Firmware upgrade status (same as Block 1 register `0x1C`).

Block 3: Configuration Registers (R/W)

Addr.	Reg #	Data Type	Description
128 0x80	129 0x81	uint16	**Current Firmware Version Number.** Read-only. Example: if the firmware version is 1.2.3 this register holds `123`.
129 0x81	130 0x82	uint16	**Velocity Measurement Unit.** Affects values in registers `0x04`, `0x05`, `0x44`, `0x45`. See Surface Velocity Output Unit for the detailed setting description. Available values: `0` = mm/s, `1` = m/s, `2` = mph, `3` = km/h, `4` = ft/s, `5` = ft/min, `6` = cm/s. **Default value:** `1` (m/s)

Addr.	Reg #	Data Type	Description
130 0x82	131 0x83	uint16	**Water Level Measurement Unit.** Affects values in registers `0x02`, `0x03`, `0x21`, `0x22`, `0x42`, `0x43`. See Water Level Output Unit for the detailed setting description. Available values: `0` = mm, `1` = cm, `2` = m, `3` = ft, `4` = in. **Default value:** `2` (m)
131 0x83	132 0x84	uint16	**Discharge Measurement Unit.** Affects values in registers `0x00`, `0x01`, `0x40`, `0x41`. See Discharge Output Unit for the detailed setting description. Available values: `0` = m³/s, `1` = m³/h, `2` = l/s, `3` = l/h, `4` = cfs (ft³/s), `5` = gal/min, `6` = ML/day (megaliters per day). **Default value:** `0` (m³/s)
132 0x84	133 0x85	uint16	**Velocity Filter Length.** In seconds. Longer values produce smoother but slower-responding readings. See Filter Length for the detailed setting description. **Valid range:** `1` - `120`. **Default value:** `10`
133 0x85	134 0x86	uint16	**Direction Filter.** See Direction Filter for the detailed setting description. Available values: `0` = measure both directions, `1` = incoming flow only, `2` = outgoing flow only. **Default value:** `0` (both)
134 0x86	135 0x87	uint16	**Radar Beam Width Configuration.** See Beam Width for the detailed setting description. Available values: `0` = wide, `1` = narrow. **Default value:** `0` (wide)
135 0x87	136 0x88	uint16	**Dynamic Flow Profiler (DFP).** Enable DFP calculation. See Dynamic Flow Profiler for the detailed setting description. Available values: `0` = disabled, `1` = enabled. **Default value:** `0` (off)

Addr.	Reg #	Data Type	Description
136, 137 0x88, 0x89	137, 138 0x89, 0x8A	float32	**Maximum Velocity Threshold.** Velocity readings above this threshold are rejected. See Maximum Velocity for the detailed setting description. **Default value:** `16.0`
138 0x8A	139 0x8B	uint16	**Maximum Velocity Unit.** See Maximum Velocity for the detailed setting description. Available values: `0` = mm/s, `1` = m/s, `2` = mph, `3` = km/h, `4` = ft/s, `5` = ft/min, `6` = cm/s. **Default value:** `1` (m/s)
139 0x8B	140 0x8C	uint16	**Radar Beam Offset Configuration.** See Beam Position for the detailed setting description. Available values: `0` = near, `1` = far. **Default value:** `0` (near)
140, 141 0x8C, 0x8D	141, 142 0x8D, 0x8E	float32	**Minimum Valid Distance.** Reflections detected closer than this distance are ignored. Use this parameter to exclude near-field returns e.g., mounting hardware, bridge elements, or other objects close to the sensor. See Minimum Distance for the detailed setting description. **Default value:** `0`
142 0x8E	143 0x8F	uint16	**Minimum Valid Distance Unit.** See Minimum Distance for the detailed setting description. Available values: `0` = mm, `1` = cm, `2` = m, `3` = ft, `4` = in. **Default value:** `2` (m)
143 0x8F	144 0x90	uint16	**Radar Receiver Mode.** Selects how the signals from the radar receive antennas are combined. Available values: `0` = coherent, `1` = sum. **Default value:** `0` (coherent)
144, 145 0x90, 0x91	145, 146 0x91, 0x92	float32	**Maximum Valid Distance.** Reflections detected beyond this distance are ignored. Use this parameter to exclude far-field returns outside the intended measurement zone e.g., opposite bank, structures, or background targets. See Maximum Distance for the detailed setting description. **Default value:** `30.0`

Addr.	Reg #	Data Type	Description
146 0x92	147 0x93	uint16	**Maximum Valid Distance Unit.** See Maximum Distance for the detailed setting description. Available values: 0 = mm, 1 = cm, 2 = m, 3 = ft, 4 = in. **Default value:** 2 (m)
147 0x93	148 0x94	uint16	**SNR Threshold** in dB. Reflections with a signal-to-noise ratio below this threshold are ignored. See SNR Threshold for the detailed setting description. **Valid range:** 0 - 255. **Default value:** 0
148 0x94	149 0x95	uint16	**RS-232 Serial Port Baud Rate.** See RS-232 Baud Rate for the detailed setting description. Available values: 0 = 4800, 1 = 9600, 2 = 14400, 3 = 19200, 4 = 38400, 5 = 57600, 6 = 115200. **Default value:** 6 (115200)
149 0x95	150 0x96	uint16	**Modbus RS-485 Baud Rate.** Caution: changing this will require the master to reconnect at the new baud rate. See RS-485 Baud Rate for the detailed setting description. Available values: 0 = 4800, 1 = 9600, 2 = 14400, 3 = 19200, 4 = 38400, 5 = 57600, 6 = 115200. **Default value:** 1 (9600)
150 0x96	151 0x97	uint16	**Modbus Parity Setting.** See Modbus Parity for the detailed setting description. Available values: 0 = none, 1 = odd, 2 = even. **Default value:** 2 (even)

Addr.	Reg #	Data Type	Description
151 0x97	152 0x98	uint16	**Modbus Stop Bits.** See RS-485 Stop Bits for the detailed setting description. Available values: `0` = one stop bit, `1` = two stop bits. **Default value:** `0` (one)
152 0x98	153 0x99	uint16	**Modbus Slave Address.** Must be unique on the RS-485 bus. The old address remains active until device reset; the configured value changes immediately. See Modbus ID for the detailed setting description. **Valid range:** `1` - `247`. **Default value:** `1`
153 0x99	154 0x9A	uint16	**SDI-12 Sensor Address.** The value is stored as a numeric integer (0 - 61) in the register. SDI-12 addressing is character-based, so the numeric value from the register is converted to the corresponding SDI-12 address character according to the following mapping: `0` - `9` -> characters `0`-`9`, `10` - `35` -> characters `A`-`Z`, `36` - `61` -> characters `a`-`z`. See SDI-12 ID for the detailed setting description. **Default value:** `0`
154 0x9A	155 0x9B	uint16	**Power Management Mode.** Gets / sets the automatic SDI-12 sleep mode parameter. When SDI-12 sleep is enabled, the device enters low-power sleep between SDI-12 transactions. When disabled, the sensor runs continuously and is ready at all times. See Power Management for the detailed setting description. Available values: `0` = sleep enabled (low power), `1` = sleep disabled (continuous scanning). **Default value:** `1` (off)
155 0x9B	156 0x9C	uint16	**Analog output selection.** Gets / sets the type of the measurement that will be sent to the analog output. See 4 - 20 mA Output Type for the detailed setting description. Available values: `0` = none, `1` = discharge, `2` = water level, `3` = distance to water, `4` = surface velocity, `5` = total volume, `6` = fixed 4 mA, `7` = fixed 12 mA, `8` = fixed 20 mA. **Default value:** `1` (discharge) **Note:** writing this register resets both the minimum value unit (register `0x9E`) and the maximum value unit (register `0xA2`) to `0`, because the set of valid units depends on the selected output type. When writing several of these registers in a single request, write this register first.

Addr.	Reg #	Data Type	Description
156, 157 0x9C, 0x9D	157, 158 0x9D, 0x9E	float32	**Minimum Value for Analog Output.** Measurements at or below this value produce minimum current. See 4 - 20 mA Minimum Value for the detailed setting description. **Default value:** `0`
158 0x9E	159 0x9F	uint16	**Minimum Value Unit for Analog Output.** See 4 - 20 mA Minimum Value for the detailed setting description. Available values, if analog output is set to discharge: `0` = m³/s, `1` = m³/h, `2` = l/s, `3` = l/h, `4` = cfs (ft³/s), `5` = gal/min, `6` = ML/day. Available values, if analog output is set to water level or distance to water: `0` = mm, `1` = cm, `2` = m, `3` = ft, `4` = in. Available values, if analog output is set to surface velocity: `0` = mm/s, `1` = m/s, `2` = mph, `3` = km/h, `4` = ft/s, `5` = ft/min, `6` = cm/s. Available values, if analog output is set to total volume: `0` = m³, `1` = l, `2` = ft³, `3` = gal, `4` = ML. **Default value:** `1` (with the factory default analog output of discharge, this is m³/h)
159 0x9F	160 0xA0		Reserved.
160, 161 0xA0, 0xA1	161, 162 0xA1, 0xA2	float32	**Maximum Value for Analog Output.** Measurements at or above this value produce maximum current. See 4 - 20 mA Maximum Value for the detailed setting description. **Default value:** `16.0`

Addr.	Reg #	Data Type	Description
162 0xA2	163 0xA3	uint16	**Maximum Value Unit for Analog Output.** See 4 - 20 mA Maximum Value for the detailed setting description. Available values, if analog output is set to discharge: 0 = m³/s, 1 = m³/h, 2 = l/s, 3 = l/h, 4 = cfs (ft³/s), 5 = gal/min, 6 = ML/day. Available values, if analog output is set to water level or distance to water: 0 = mm, 1 = cm, 2 = m, 3 = ft, 4 = in. Available values, if analog output is set to surface velocity: 0 = mm/s, 1 = m/s, 2 = mph, 3 = km/h, 4 = ft/s, 5 = ft/min, 6 = cm/s. Available values, if analog output is set to total volume: 0 = m³, 1 = l, 2 = ft³, 3 = gal, 4 = ML. **Default value:** 1 (with the factory default analog output of discharge, this is m³/h)
163 0xA3	164 0xA4	uint16	**Total Volume Measurement Unit.** Selects the unit used for the total accumulated volume reported in registers 0x1D, 0x1E. Available values: 0 = m³, 1 = l, 2 = ft³, 3 = gal, 4 = ML (megaliters). **Default value:** 0 (m³)

Addr.	Reg #	Data Type	Description
164 0xA4	165 0xA5	uint16	**Operation Mode.** When the operation mode is set to standby, the device will communicate on all communication interfaces, but the radar transmitters will be turned off to reduce the power, and the instrument will not be making any measurements. See Operation Mode for the detailed setting description. Available values: `0` = off (normal operation), `1` = on (standby operation). **Default value:** `0` (off)
165 0xA5	166 0xA6	uint16	**Disable NMEA Sentence Output.** On RS-232. Available values: `0` = NMEA enabled, `1` = NMEA disabled. **Default value:** `0` (off)
166 0xA6	167 0xA7	uint16	**Show Data on Error.** Continue reporting velocity data even when angle errors are detected. Available values: `0` = clear velocity on error (send 0 as velocity instead of measured value), `1` = keep velocity. **Default value:** `0` (off)
167 0xA7	168 0xA8	uint16	**Fixed Mounting Angle.** Override measured forward angle with a fixed value (degrees). Changing this parameter will affect the reported velocity as cosine correction behaviour will be changed. Use this parameter only during tow-tank testing. `0` = use measured angle. **Valid range:** `0` - `89`. **Default value:** `0` (off)
168, 169 0xA8, 0xA9	169, 170 0xA9, 0xAA	float32	**Radar Vertical Position.** Defines the vertical position of the radar sensor above the hydrological zero (gauge zero). This value is used to calculate the water level from the measured distance to the water surface and to determine the sensor position within the channel cross-section for discharge calculations. The unit is set by configuration register `0xAB`. **Default value:** `0`
170 0xAA	171 0xAB	uint16	**LiDAR Scan Period (FX200 only).** Read/write interval between automatic scans, in minutes. Zero-based address `0xAA`; one-based register number `171` (PLC `40171`). Used when LiDAR mode is automatic. **Valid range:** `1`–`32767` minutes. **Default:** `15`.
171 0xAB	172 0xAC	uint16	Unit for radar vertical position. `0` = mm, `1` = cm, `2` = m, `3` = ft, `4` = in. **Default value:** `2` (m)

Addr.	Reg #	Data Type	Description
172 0xAC	173 0xAD	uint16	Enable Wi-Fi. `0` = off, `1` = on. **Default value:** `0` (off)
173 - 180 0xAD - 0xB4	174 - 181 0xAE - 0xB5	string[16]	Wi-Fi password, up to 16 characters. Shorter passwords are padded with null characters.
181 0xB5	182 0xB6		Reserved. Reads as 0; writes return exception `0x02`.
182 0xB6	183 0xB7	uint16	**LiDAR Mode.** Enables automatic scanning of the channel cross-section with the built-in LiDAR sensor. See LiDAR Mode for the detailed setting description. Available values: `0` = off (automatic scanning disabled), `1` = automatic, `2` = semi-automatic (accepted by the register, but not yet implemented in the firmware - do not use). **Default value:** `0` (off)
183 0xB7	184 0xB8	uint16	**LiDAR Left Limit Angle.** Defines how far the LiDAR sweep extends towards the left side of the channel, in degrees measured from the vertical axis of the instrument. See Left Limit Angle for the detailed setting description. **Valid range:** `0` - `180`; values above `70` have no additional effect, since `70` corresponds to the full mechanical sweep range of the sensor. **Default value:** `70`
184 0xB8	185 0xB9	uint16	**LiDAR Right Limit Angle.** Defines how far the LiDAR sweep extends towards the right side of the channel, in degrees measured from the vertical axis of the instrument. See Right Limit Angle for the detailed setting description. **Valid range:** `0` - `180`; values above `70` have no additional effect, since `70` corresponds to the full mechanical sweep range of the sensor. **Default value:** `70`
185 0xB9	186 0xBA	uint16	Command register. Write value `0xAD10` to this register to perform a software reset. The device will restart immediately. The magic value prevents accidental resets.
186 0xBA	187 0xBB	uint16	Command register. Write value `0xBAC1` to this register to restore all parameters to factory defaults and restart. The magic value prevents accidental resets.
187 0xBB	188 0xBC	uint16	Command register. Write value `0xCDF0` to this register to clear the DFP (Dynamic Flow Profiler) learned data. The magic value prevents accidental resets.
188 0xBC	189 0xBD	uint16	Command register. Write `0xAD11` to reset the radar co-processor. Reads are not supported.

Addr.	Reg #	Data Type	Description
189, 190 0xBD, 0xBE	190, 191 0xBE, 0xBF	float	Command register. Write the value of the current staff gauge reading to automatically adjust the vertical position of the radar. The units of the measurement are defined in register 171 (172). Writing to this register will automatically adjust the value in register 168-169 (169-170).
191 0xBF	192 0xC0	uint16	Command register. Write value `0xC003` to this register, to reset the totalizer to zero.
192 0xC0	193 0xC1	uint16	Command register. Write value `0xC004` to this register to delete the stored LiDAR channel cross-section scans. The magic value prevents accidental erasure. The scans are not deleted immediately: the request is queued and carried out once the LiDAR is idle, so that a scan that is currently running or being written to storage is not interrupted.
193 0xC1	194 0xC2		Reserved.
194 0xC2	195 0xC3	uint16	Command register. Write value `0xB61D` to this register to start a single (manual) LiDAR scan of the channel cross-section. The scan takes 30 to 45 seconds to complete, and the command is ignored if a scan is already running. The magic value prevents accidental scans. See Manual LiDAR Scan for the detailed description.
195 0xC3	196 0xC4	uint16	Command register. Write `0xD1E5` to stop the current manual LiDAR scan. Reads are not supported.

Reading LiDAR Scan

The channel cross-section produced by the most recent automatic or manual LiDAR scan can be read out over Modbus. This is the same data shown on the **LiDAR** page of the configurator, see 7.3. Using LiDAR.

The scan is a relatively large data set (up to 1000 measured points) and does not fit into the holding-register map. It is therefore transferred using a set of **custom Modbus function codes** that move an arbitrary-length byte stream (a "file") between the master and the device, rather than through registers. The same mechanism is used for firmware upgrades and echo-curve download.

The automatic LiDAR scan is exposed as **file number 2**. The manual LiDAR scan is exposed as **file number 3**.

Custom Function Codes

Three custom function codes are used. In all of them the first byte of the frame is the slave address and the last two bytes are the standard Modbus RTU CRC (CRC-16/MODBUS, low byte first). All multi-byte fields inside the frames are big-endian.

Code	Meaning
`0x46` (70)	File operation: open for read, open for write, or close.

Code	Meaning
0x47 (71)	Read a block of bytes from the currently open file.
0x48 (72)	Write a block of bytes to the currently open file (not used when reading a scan).

File operation (0x46) – request:

Field	Size	Value
Address	1	Slave address
Function	1	0x46
Operation	1	0x01 = open read, 0x02 = open write, 0xFF = close
File number	1	2 (0x02) for the automatic LiDAR scan, or 3 (0x03) for the manual LiDAR scan
File size	4	Ignored when opening the LiDAR scan for read; send 0
CRC	2	Modbus RTU CRC

File operation (0x46) – response to open read: the device prepares the scan and returns its total size in bytes:

Field	Size	Value
Address	1	Slave address
Function	1	0x46
File size	4	Total size of the scan payload, in bytes
CRC	2	Modbus RTU CRC

File read (0x47) – request:

Field	Size	Value
Address	1	Slave address
Function	1	0x47
Start address	4	Byte offset within the file to start reading from
Byte count	2	Number of bytes to read (keep within a single Modbus frame, e.g. 240)
CRC	2	Modbus RTU CRC

File read (0x47) – response:

Field	Size	Value
Address	1	Slave address
Function	1	0x47

Field	Size	Value
Byte count	2	Number of data bytes that follow
Data	n	The requested block of file bytes
CRC	2	Modbus RTU CRC

If the requested offset is at or beyond the end of the file, the device returns a byte count of `0`. This marks the end of the transfer.

An exception response (function code with the high bit set, e.g. `0xC6`) with exception code `6` (busy) is returned if the device cannot prepare the scan at that moment; retry after a short delay.

Transfer sequence

To read the scan:

1. Send an **open read** request for file `2` or `3` and read the returned file size.
2. Repeatedly send **read** requests, advancing the start address by the number of bytes returned each time, until the returned byte count is `0` or the accumulated length reaches the file size.
3. Send a **close** request (operation `0xFF`) to release the buffer on the device.

Example frames for slave address 1:

```
open read : 01 46 01 03 00 00 00 00 E3 D2
read      : 01 47 00 00 00 00 00 F0 B6 87   (240 bytes from offset 0)
close     : 01 46 FF 03 00 00 00 00 F6 0C
```

Payload format

The bytes returned for file `2` or `3` are the scanned profile serialized as a CBOR (RFC 8949) map and then compressed with Zstandard (zstd). Unlike the servicing protocol, the Modbus transfer carries the raw compressed bytes directly: there is **no byte-stuffing and no extra application CRC**, since the Modbus frame CRC already protects each block.

After concatenating all the blocks, decompress the payload with zstd and parse the resulting CBOR map. The relevant keys are:

Key	Type	Description
`unt`	text string	Unit of the coordinates. Always `mm`.
`x`	array of float32	X coordinates of the profile points, left to right.
`y`	array of float32	Y coordinates of the profile points. Values are negative, measured downward from the sensor.

The `x` and `y` arrays have the same length and together define the cross-section points.

Python example

The following example reads the scan over an RS-485 adapter and saves the profile as a CSV file with `x` and `y` columns in millimeters. Because the transfer uses custom function codes that standard Modbus libraries do not implement, the frames are built directly on top of a serial port. It requires the `pyserial`, `zstandard`, and `cbor2` packages (`pip install pyserial zstandard cbor2`).

PYTHON

```python
import csv
import struct
import time
import serial
import zstandard
import cbor2

PORT = "COM3"          # RS-485 adapter port (e.g. "/dev/ttyUSB0")
BAUD = 9600
SLAVE = 1
LIDAR_FILE = 3           # 2 = automatic scan, 3 = manual scan
CHUNK = 240            # bytes per read request
OUTPUT_CSV = "lidar_profile.csv"

def crc16_modbus(frame: bytes) -> int:
    crc = 0xFFFF
    for byte in frame:
        crc ^= byte
        for _ in range(8):
            if crc & 1:
                crc = (crc >> 1) ^ 0xA001
            else:
                crc >>= 1
    return crc

def send(port: serial.Serial, pdu: bytes) -> bytes:
    frame = bytes([SLAVE]) + pdu
    frame += struct.pack("<H", crc16_modbus(frame))   # CRC low byte first
    port.write(frame)
    time.sleep(0.05)
    resp = port.read(512)
    if len(resp) < 4 or crc16_modbus(resp[:-2]) != struct.unpack("<H", resp[-2:])[0]:
        raise IOError("bad or missing response: " + resp.hex(" "))
    if resp[1] & 0x80:
        raise IOError(f"Modbus exception {resp[2]}")
    return resp

with serial.Serial(PORT, BAUD, parity=serial.PARITY_EVEN, timeout=1) as port:
    # 1. Open the selected LiDAR file for read; response carries the total size (4 bytes).
    resp = send(port, bytes([0x46, 0x01, LIDAR_FILE]) + struct.pack(">I", 0))
    file_size = struct.unpack(">I", resp[2:6])[0]

    # 2. Read the file in chunks until the whole payload is received.
    payload = bytearray()
    while len(payload) < file_size:
        pdu = bytes([0x47]) + struct.pack(">I", len(payload)) + struct.pack(">H", CHUNK)
        resp = send(port, pdu)
        count = struct.unpack(">H", resp[2:4])[0]
        if count == 0:
            break
        payload += resp[4:4 + count]

```

```
    # 3. Close the file to release the device buffer.
    send(port, bytes([0x46, 0xFF, LIDAR_FILE]) + struct.pack(">I", 0))

cbor_bytes = zstandard.ZstdDecompressor().decompress(bytes(payload))
profile = cbor2.loads(cbor_bytes)

with open(OUTPUT_CSV, "w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["x_mm", "y_mm"])
    for x, y in zip(profile["x"], profile["y"]):
        writer.writerow([x, y])

print(f"Saved {len(profile['x'])} points to {OUTPUT_CSV}")
```

6.4. SDI-12 Protocol

Protocol Overview

The FX200 flow meter supports communication via the SDI-12 (Serial Data Interface at 1200 baud) protocol. The instrument responds to standard SDI-12 measurement commands by returning measured surface velocity and related data, and, in addition to standard measurement commands, the device implements a set of custom SDI-12 commands that allow users to read and modify internal configuration parameters, retrieve diagnostic information, and execute specific service operations such as restoring factory default settings.

In the command tables below, **a** represents the device address. CR and LF denote carriage return / line feed pair which is sent at the end of each response. The response contains no space character – in the table below the spaces are inserted between response elements to improve readability only.

Standard SDI-12 Commands

Request	Response	Example
Device Address Query	Device will identify by returning its SDI-12 address. The factory default address is 0.	
`?!`	`a CR LF`	→ `?!` ← `0`
Ping Sensor	The device will return its address if the address in the request matches the device address. This command is used to verify that the device is active, at a given address, and responding to requests.	
`a!`	`a CR LF`	→ `0!` ← `0`

Change Address	The device will reconfigure to change its SDI-12 device address, and will respond with the new address. All subsequent requests to the device should be addressed to the new address. Take care when changing address, if improperly set on multi-sensor bus this can cause address collision.	
`aAb!`	`b CR LF`	→ `0A5!` ← `5`

Get Device Identification	The device will respond with its identification information: • `14` = SDI protocol version 1.4 • `GEOLUX  FX200` = device name, with two space characters between • `fff` = current firmware version, one space character before • `ssssss` = six-digit decimal device serial number, padded with leading zeros and appended directly after the firmware version (no separator).	
`aI!`	`a14GEOLUX  FX200 fffssssss CR LF`	→ `0I!` ← `014GEOLUX  FX200 105100256`
	The example uses serial number `100256`.	

Verify Device	This command is executed in two steps. First, the master sends the command `aV!` to initiate the verification procedure. The device responds with a SDI-12 acknowledgment indicating that one value is immediately available for retrieval. No delayed service request is needed. The master must then issue the `aD0!` command to read the available value. The device responds with a single measurement representing the instrument's operational status: • `0` = not ready • `1` = ready Readiness means the application is running and the device status bitmask is zero. A sleeping or initializing instrument reports not ready; this command does not start a measurement.	

`aV!`
 	
`aD0!`
	`a0001 CR LF`
	
	`a+s CR LF`
		→ `0V!`
		← `00001`
		→ `0D0!`
		← `0+1`

<table>
<tr><td>Perform Measurement</td><td colspan="2">This command instructs the instrument to start a measurement cycle. Upon receiving aM!, the device immediately responds with the required warm-up time in seconds and the total number of measurement values that will be available once the measurement is completed.

After the specified warm-up time has elapsed, the master should retrieve the measurement results using the aD0!, aD1!, etc. commands as needed, until all reported values have been read.

If a CRC checksum is required to ensure data integrity, the aMC! command shall be used instead of aM!. This command functions identically to aM!, but the instrument appends a CRC checksum to each data response. The checksum in the response allows the master to verify the consistency and correctness of the transmitted values.

The Geolux FX200 returns 8 values, in the following order:
1. Discharge, in selected measurement unit,
2. Water level, in selected measurement unit,
3. Distance to water, in selected measurement unit,
4. Absolute surface velocity, in selected measurement unit,
5. Total volume, in selected measurement unit,
6. Flow direction flag (1 incoming, -1 outgoing, 0 no flow),
7. Surface velocity SNR, in dB,
8. Water level SNR, in dB.

The grouping of measurement values in responses to the aD0!, aD1!, aD2!, etc., commands does not necessarily follow the structure shown in the example above. The example is provided for illustration purposes only. In accordance with the SDI-12 protocol, the instrument may distribute the returned values across one or more data responses depending on the total length of the response string. Each SDI-12 response is limited in length, so the device dynamically groups values to fit within this constraint.</td></tr>
<tr><td>aM! or aMC!

aD0!

aD1!</td><td>asss8 CR LF

a discharge level dist vel CR LF

a volume dir svrsnr levsnr CR LF</td><td>→ 0M!
← 00158
… after 15 seconds …
→ 0D0!
← 0+1.023+1.12+2.21+0.54
→ 0D1!
← 0+10.212+1.0+48.0+42.0</td></tr>
</table>

Measure System Status	This command will instruct the instrument to measure and return system status. The most recent status values are immediately available; this command does not start a new radar measurement. If a CRC checksum is required to ensure data integrity, the `aMC1!` command shall be used instead of `aM1!`. This command functions identically to `aM1!`, but the instrument appends a CRC checksum to each data response. The checksum in the response allows the master to verify the consistency and correctness of the transmitted values. The Geolux FX200 returns 6 values, in the following order: 1. Forward tilt angle, in degrees, 2. Side tilt angle, in degrees, 3. Device temperature, in °C, 4. Device relative humidity, in %, 5. Device error status bitmask: Bit 0 = forward tilt angle error Bit 1 = side tilt angle error Bit 2 = radar co-processor update failed Bit 3 = velocity radar co-processor not responding Bit 4 = radar co-processor needs to be updated Bit 5 = tilt sensor is not working Bit 6 = level radar co-processor not responding Bit 7 = DFP data corrupt, needs reset Bit 8 = standby mode is active Bit 9 = radar is initializing 6. LiDAR scan status: `0` = idle, `1` = automatic scan running, `2` = manual scan running. LiDAR status is captured when `aM1!` or `aMC1!` is received. Idle means no scan is currently running, even if automatic scanning is enabled. This query does not start a scan. The concurrent/continuous `aC1!` and `aR1!` groups retain their existing five values. The grouping of measurement values in responses to the aD0!, aD1!, aD2!, etc., commands does not necessarily follow the structure shown in the example above. The example is provided for illustration purposes only. In accordance with the SDI-12 protocol, the instrument may distribute the returned values across one or more data responses depending on the total length of the response string. Each SDI-12 response is limited in length, so the device dynamically groups values to fit within this constraint.	

`aM1!` or `aMC1!`
 	
`aD0!`
	`a0006 CR LF`
	
	`a fwdtilt sidetilt temp rhum status lidar_status CR LF`
		→ `0M1!`
		← `00006`
		→ `0D0!`
		← `0+1.0+0.0+22.4+39.0+0+1`

<table>
<tr><td>Perform Concurrent Measurement</td><td colspan="2">The aC! command operates similarly to the previously described aM! command, with the key difference being that it enables concurrent measurements. After receiving aC!, the instrument responds with the required processing time and the number of values that will be available.

Unlike aM!, the instrument does not hold the SDI-12 bus during the measurement process. This allows other SDI-12 devices on the same bus to perform measurements simultaneously. Once the indicated processing time has elapsed, the master can retrieve the results using the aD0!, aD1!, etc., commands, in the same manner as for aM!.

In accordance with the SDI-12 specification for concurrent commands, the response to aC! uses two digits to indicate the number of measurement values that will be returned.

If a CRC checksum is required to ensure data integrity, the aCC! command shall be used instead of aC!. This command functions identically to aC!, but the instrument appends a CRC checksum to each data response. The checksum in the response allows the master to verify the consistency and correctness of the transmitted values.

The grouping of measurement values in responses to the aD0!, aD1!, aD2!, etc., commands does not necessarily follow the structure shown in the example above. The example is provided for illustration purposes only. In accordance with the SDI-12 protocol, the instrument may distribute the returned values across one or more data responses depending on the total length of the response string. Each SDI-12 response is limited in length, so the device dynamically groups values to fit within this constraint.</td></tr>
<tr><td>aC! or aCC!

aD0!

aD1!</td><td>asss08 CR LF

a discharge level dist vel CR LF

a volume dir svrsnr levsnr CR LF</td><td>→ 0C!
← 001508
… after 15 seconds …
→ 0D0!
← 0+1.023+1.12+2.21+0.54
→ 0D1!
← 0+10.212+1.0+48.0+42.0</td></tr>
</table>

Identify Parameter (SDI-12 Version 1.4)	The FX200 implements the SDI-12 version 1.4 **Identify Parameter** command. This command returns descriptive information about a measurement parameter supported by the instrument. The command format is `aIM_nnn!`, where `a` is the SDI-12 device address and `nnn` is the three-digit parameter number. The FX200 supports parameter numbers `001` through `008`, corresponding to the eight values returned by the standard measurement command. The response format is: • `a` = device address • `name` = parameter name • `unit` = parameter unit	
`aIM_nnn!`	`a,name,unit; CR LF`	→ `0IM_001!` ← `0,discharge,m3/s;` → `0IM_002!` ← `0,level,m;` → `0IM_003!` ← `0,distance,m;` → `0IM_004!` ← `0,velocity,mms;` → `0IM_005!` ← `0,volume,m3;` → `0IM_006!` ← `0,direction, ;` → `0IM_007!` ← `0,SNR,dB;` → `0IM_008!` ← `0,levelSNR,dB;`

Channel Profile (High Volume Commands)

The FX200 can return the channel cross-section profile measured by the LiDAR using the SDI-12 version 1.4 **high volume** commands. The profile is a list of points, each described by two coordinates:

- `profile_x` = horizontal offset from the instrument
- `profile_y` = vertical distance from the instrument, corrected by the configured LiDAR offset

Both coordinates are expressed in the configured water level unit (see Water Level Output Unit).

Because a profile is far too large to be produced within the time a sensor is allowed to answer a command, it is not measured on demand. The profile source must first be selected with the Channel Profile Source command, which starts the preparation in the background. Once the source reports that it is ready, the high volume commands return the prepared profile immediately.

Until a source has been selected and prepared, all high volume commands report that no data is available (a000000 for either aHB! or aHA!).

> *Note:*
>
> The profile is available in both a binary and an ASCII form and both return the same data. The binary form is strongly recommended: at 1200 baud a full 499-point profile takes roughly half a minute to transfer in binary form, and considerably longer in ASCII form.

<table>
<tr><th>Request</th><th>Response</th><th>Example</th></tr>
<tr><td>Read Profile in Binary Format</td><td colspan="2">The aHB! command reports how many values are available. The response contains ttt = time in seconds until the data is ready, which is always 000 because the profile is prepared in advance, and nnn = the number of values. A profile of N points returns 2N values: N profile_x values followed by N profile_y values.</td></tr>
<tr><td>aHB!</td><td>a ttt nnn CR LF</td><td>→ 0HB!
← 0000640</td></tr>
<tr><td></td><td colspan="2">The values are then read with the aDBn! commands, where n is the segment number starting at 0. Unlike every other SDI-12 response, the response to aDBn! is binary, is transmitted as 8 data bits with no parity, and is not terminated by CR LF:
<table>
<tr><th>Byte</th><th>Content</th></tr>
<tr><td>0</td><td>Device address</td></tr>
<tr><td>1 – 2</td><td>Number of data bytes in this segment, unsigned 16-bit, low byte first</td></tr>
<tr><td>3</td><td>Data type, always 9 (IEEE-754 32-bit floating point)</td></tr>
<tr><td>4 …</td><td>The data bytes, each value low byte first</td></tr>
<tr><td>last 2</td><td>CRC-16, low byte first, calculated over all preceding bytes of the response</td></tr>
</table>
Each segment carries at most 1000 data bytes, so 250 values. The master must keep requesting consecutive segments until a segment reports 0 data bytes and data type 0. The values are packed continuously across the segments: a segment boundary may fall inside the profile_x block, or between the profile_x and profile_y blocks. The master should therefore concatenate all segments first and split the result in half afterwards.</td></tr>
<tr><td>aDBn!</td><td>binary, see the table above</td><td>→ 0DB0!
← 0 E8 03 09 <1000 data bytes> <CRC>

→ 0DB1!
← 0 E8 03 09 <1000 data bytes> <CRC>

→ 0DB2!
← 0 30 02 09 <560 data bytes> <CRC>

→ 0DB3!
← 0 00 00 00 <CRC></td></tr>
</table>

Read Profile in ASCII Format	The aHA! command reports how many values are available, in the same way as aHB! but with a three-digit count. The values are then read with the standard aD0! … aD999! commands. Each response carries four values, except for a shorter final response when needed, and always ends with a three-character CRC. This applies to both manual and automatic profiles. Both forms return at most 499 profile points (998 coordinate values), so the value count fits in the three-digit response field.	
aHA!	a ttt nnn CR LF	→ 0HA! ← 0000640
aDn!	a value … CRC CR LF	→ 0D0! ← 0+1.234+1.310+1.387+1.463<CRC>

Identify Profile Value	The aIHB_nnnn! and aIHA_nnn! commands return the description of a single profile value, where nnnn / nnn is the value number starting at 1. The response ends with a three-character CRC.	
aIHB_nnnn!	a,name,unit,Element k; CRC CR LF	→ 0IHB_0001! ← 0,profile_x,m,Element 1; → 0IHB_0321! ← 0,profile_y,m,Element 1;
aIHA_nnn!	a,name,unit; CRC CR LF	→ 0IHA_001! ← 0,profile_x,m;

Custom (Extended) SDI-12 Commands Specific for FX200

Request	Response	Example
Maximum Velocity Threshold	Sets / gets the upper bound for valid surface velocity measurements. Velocity readings above this threshold are rejected. The numeric value parameter is interpreted in the unit set by aXGMVU!. See Maximum Velocity for the detailed setting description.	
Get: aXGMVL!	a value CR LF	→ 0XGMVL! ← 016
Set: aXGMVL value!	a value CR LF	→ 0XGMVL+15! ← 015

Maximum Velocity Unit	Gets / sets the physical unit used by `aXGMVL!` command. See Maximum Velocity for the detailed setting description. The units are represented by the numerical value as follows: • `0` = mm/s, • `1` = m/s, • `2` = mph, • `3` = km/h, • `4` = ft/s, • `5` = ft/min, • `6` = cm/s.	
Get: `aXGMVU!`	`a value CR LF`	→ `0XGMVU!` ← `01`
Set: `aXGMVU value!`	`a value CR LF`	→ `0XGMVU+6!` ← `06`

Direction Filter	Gets / sets the direction filter configuration. See Direction Filter for the detailed setting description. • `0` = measure both directions • `1` = incoming flow only • `2` = outgoing flow only	
Get: `aXGDFT!`	`a value CR LF`	→ `0XGDFT!` ← `00`
Set: `aXGDFT value!`	`a value CR LF`	→ `0XGDFT+2!` ← `02`

SNR Threshold	Gets / sets the SNR threshold value. See SNR Threshold for the detailed setting description.	
Get: `aXGSTH!`	`a value CR LF`	→ `0XGSTH!` ← `00`
Set: `aXGSTH value!`	`a value CR LF`	→ `0XGSTH+2!` ← `02`

Filter Length	Gets / sets the number of seconds used for internal smoothing. Higher values reduce noise but increase response lag. See Filter Length for the detailed setting description. The valid range is between 1 and 120.	
Get: `aXGFLN!`	`a value CR LF`	→ `0XGFLN!` ← `01`
Set: `aXGFLN value!`	`a value CR LF`	→ `0XGFLN+20!` ← `020`

Discharge Output Unit	Gets / sets the unit used for the discharge value in the SDI-12 measurement response, NMEA and on Modbus output. See Discharge Output Unit for the detailed setting description. The units are represented by the numerical value as follows: • `0` = m³/s, • `1` = m³/h, • `2` = L/s, • `3` = L/h, • `4` = ft³/s (cfs), • `5` = gal/min, • `6` = ML/day.	
Get: `aXGDSU!`	`a value CR LF`	→ `0XGDSU!` ← `01`
Set: `aXGDSU value!`	`a value CR LF`	→ `0XGDSU+3!` ← `03`

Water Level Output Unit	Gets / sets the unit used for the water level and distance to water values in the SDI-12 measurement response, NMEA and on Modbus output. See Water Level Output Unit for the detailed setting description. The units are represented by the numerical value as follows: • `0` = mm, • `1` = cm, • `2` = m, • `3` = ft, • `4` = in. `0XGLVU+2!` selects metres. To select mph for velocity output, use `0XGUNT+2!`.	
Get: `aXGLVU!`	`a value CR LF`	→ `0XGLVU!` ← `02`
Set: `aXGLVU value!`	`a value CR LF`	→ `0XGLVU+2!` ← `02`

Velocity Output Unit	Gets / sets the unit used for the velocity value in the SDI-12 measurement response, NMEA and on Modbus output. See Surface Velocity Output Unit for the detailed setting description. The units are represented by the numerical value as follows: • `0` = mm/s • `1` = m/s • `2` = mph • `3` = km/h • `4` = ft/s • `5` = ft/min • `6` = cm/s	
Get: `aXGUNT!`	`a value CR LF`	→ `0XGUNT!` ← `01`
Set: `aXGUNT value!`	`a value CR LF`	→ `0XGUNT+0!` ← `00`

Beam Width	Gets / sets the radar beam width pattern. Wide beam covers a larger area; narrow beam is more focused and rejects off-axis targets. See Beam Width for the detailed setting description. • `0` = wide • `1` = narrow	
Get: `aXGRBW!`	`a value CR LF`	→ `0XGRBW!` ← `00`
Set: `aXGRBW value!`	`a value CR LF`	→ `0XGRBW+1!` ← `01`

Beam Width in Azimuth (RX Mode)	Gets / sets how the radar combines the receive channels, controlling the azimuth beam pattern. This is separate from the elevation beam width selected by `aXGRBW!`. • `0` = coherent • `1` = sum The default is `0` (coherent).	
Get: `aXGRXM!`	`a value CR LF`	→ `0XGRXM!` ← `00`
Set: `aXGRXM value!`	`a value CR LF`	→ `0XGRXM+1!` ← `01`

Beam Position	Gets / sets the near or far measurement zone placed along the radar beam used for target detection. See Beam Position for the detailed setting description. • `0` = near • `1` = far	
Get: `aXGBOF!`	`a value CR LF`	→ `0XGBOF!` ← `00`
Set: `aXGBOF value!`	`a value CR LF`	→ `0XGBOF+1!` ← `01`

DFP Enable	Gets / sets the enable status of Dynamic Flow Profiler algorithm, which tracks and accumulates flow history for improved accuracy in low-flow conditions. See Dynamic Flow Profiler for the detailed setting description. • `0` = disabled • `1` = enabled	
Get: `aXGDFP!`	`a value CR LF`	→ `0XGDFP!` ← `00`
Set: `aXGDFP value!`	`a value CR LF`	→ `0XGDFP+1!` ← `01`

Radar Height	Gets / sets the vertical distance between the instrument and the water level reference point (gauge zero). The instrument uses this value to calculate and report the water level relative to the gauge zero. The corresponding measurement unit is set by `aXGRHU!`.	
Get: `aXGRHG!`	`a value CR LF`	→ `0XGRHG!` ← `00`
Set: `aXGRHG value!`	`a value CR LF`	→ `0XGRHG+12.11!` ← `0+12.11`

Radar Height Unit	Gets / sets the measurement unit used by the parameter set by aXGRHG! The valid units are: • `0` = mm, • `1` = cm, • `2` = m, • `3` = ft, • `4` = in. `0XGRHU+2!` selects metres for radar height. This is a distance unit, so mph is not supported.	
Get: `aXGRHU!`	`a value CR LF`	→ `0XGRHU!` ← `02`
Set: `aXGRHU value!`	`a value CR LF`	→ `0XGRHU+1!` ← `01`

Minimum Valid Distance	Gets / sets the minimum valid distance to water surface. Reflections detected closer than this distance are ignored. Use this parameter to exclude near-field returns e.g., mounting hardware, bridge elements, or other objects close to the sensor. See Minimum Distance for the detailed setting description. The corresponding measurement unit is set by `aXGDU0!`.	
Get: `aXGDZ0!`	`a value CR LF`	→ `0XGDZ0!` ← `00`
Set: `aXGDZ0 value!`	`a value CR LF`	→ `0XGDZ0+1!` ← `01`

Minimum Valid Distance Unit	Gets / sets the measurement unit used by minimum valid distance parameter (`aXGDZ0!`). See Minimum Distance for the detailed setting description. The valid units are: • `0` = mm • `1` = cm • `2` = m • `3` = ft • `4` = in	
Get: `aXGDU0!`	`a value CR LF`	→ `0XGDU0!` ← `02`
Set: `aXGDU0 value!`	`a value CR LF`	→ `0XGDU0+1!` ← `01`

Maximum Valid Distance	Gets / sets the maximum valid distance to water surface. Reflections detected beyond this distance are ignored. Use this parameter to exclude far-field returns outside the intended measurement zone e.g., opposite bank, structures, or background targets. See Maximum Distance for the detailed setting description. The corresponding measurement unit is set by `aXGDU1!`.	
Get: `aXGDZ1!`	`a value CR LF`	→ `0XGDZ1!` ← `030`
Set: `aXGDZ1 value!`	`a value CR LF`	→ `0XGDZ1+5!` ← `05`

Maximum Valid Distance Unit	Gets / sets the measurement unit used by maximum valid distance parameter (`aXGDZ1!`). See Maximum Distance for the detailed setting description. The valid units are: • `0` = mm • `1` = cm • `2` = m • `3` = ft • `4` = in	
Get: `aXGDU1!`	`a value CR LF`	→ `0XGDU1!` ← `02`
Set: `aXGDU1 value!`	`a value CR LF`	→ `0XGDU1+3!` ← `03`

RS-232 Baud Rate	Gets / sets the baud rate of the RS-232 serial interface. The change takes effect immediately. See RS-232 Baud Rate for the detailed setting description. The valid values for baud rates are: • `0` = 4800 • `1` = 9600 • `2` = 14400 • `3` = 19200 • `4` = 38400 • `5` = 57600 • `6` = 115200	
Get: `aXGSBD!`	`a value CR LF`	→ `0XGSBD!` ← `06`
Set: `aXGSBD value!`	`a value CR LF`	→ `0XGSBD+1!` ← `01`

Modbus Baud Rate	Gets / sets the baud rate of the RS-485/Modbus serial interface. The change takes effect immediately. See RS-485 Baud Rate for the detailed setting description. The valid values for baud rates are: • 0 = 4800 • 1 = 9600 • 2 = 14400 • 3 = 19200 • 4 = 38400 • 5 = 57600 • 6 = 115200	
Get: aXGMBD!	a value CR LF	→ 0XGMBD! ← 01
Set: aXGMBD value!	a value CR LF	→ 0XGMBD+6! ← 06

Modbus Device ID	Gets / sets the Modbus RTU slave address. Must be unique on the RS485 bus. See Modbus ID for the detailed setting description. The accepted value range for Modbus RTU slave address is 1 - 247.	
Get: aXGMID!	a value CR LF	→ 0XGMID! ← 01
Set: aXGMID value!	a value CR LF	→ 0XGMID+10! ← 010

Modbus Serial Line Parity	Gets / sets the serial line parity for RS-485 / Modbus connection. The change takes place immediately. See Modbus Parity for the detailed setting description. The valid values are: • 0 = none • 1 = odd • 2 = even	
Get: aXGMPA!	a value CR LF	→ 0XGMPA! ← 02
Set: aXGMPA value!	a value CR LF	→ 0XGMPA+0! ← 00

Modbus Serial Line Stop Bits	Gets / sets the number of stop bits for the serial line of RS-485 / Modbus connection. The change takes place immediately. See RS-485 Stop Bits for the detailed setting description. The valid values are: • 0 = one • 1 = two	
Get: aXGMSB!	a value CR LF	→ 0XGMSB! ← 00
Set: aXGMSB value!	a value CR LF	→ 0XGMSB+1! ← 01

Disable NMEA Output	Gets / sets the parameter that disables NMEA sentence output on RS-232. The valid values are: • `0` = NMEA enabled • `1` = NMEA disabled	
Get: `aXGNMA!`	`a value CR LF`	→ `0XGNMA!` ← `00`
Set: `aXGNMA value!`	`a value CR LF`	→ `0XGNMA+1!` ← `01`

Analog Output Type	Gets / sets the which device measurement will be used to drive the analog output. The accepted values are: • `0` = none, • `1` = discharge, • `2` = water level, • `3` = distance to water, • `4` = surface velocity, • `5` = total volume, • `6` = fixed 4 mA, • `7` = fixed 12 mA, • `8` = fixed 20 mA. Values outside `0` through `8` are invalid and leave the setting unchanged.	
Get: `aXGANU!`	`a value CR LF`	→ `0XGANU!` ← `01`
Set: `aXGANU value!`	`a value CR LF`	→ `0XGANU+2!` ← `02`

Analog Output Minimum Value	Gets / sets the selected measurement value that corresponds to the minimum output on the analog current interface (4 mA). See 4 - 20 mA Minimum Value for the detailed setting description. The measurement unit is set by `aXGAU0!`.	
Get: `aXGAV0!`	`a value CR LF`	→ `0XGAV0!` ← `00`
Set: `aXGAV0 value!`	`a value CR LF`	→ `0XGAV0+0.1!` ← `00.1`

Analog Output Minimum Unit	Gets / sets the measurement unit for the value set by `aXGAV0!` command. See 4 - 20 mA Minimum Value for the detailed setting description. The valid units depend on the output type set by `aXGANU`: For discharge: • `0` = m³/s • `1` = m³/h • `2` = l/s • `3` = l/h • `4` = cfs • `5` = gal/min • `6` = ML/day For water level and distance to water: • `0` = mm • `1` = cm • `2` = m • `3` = ft • `4` = in For surface velocity: • `0` = mm/s • `1` = m/s • `2` = mph • `3` = km/h • `4` = ft/s • `5` = ft/min • `6` = cm/s For volume: • `0` = m³ • `1` = l • `2` = ft³ • `3` = gal • `4` = ML The set command takes the numeric unit index below. Get and set responses return the unit name, not its index. For velocity the returned names are `mms`, `ms`, `mph`, `kmh`, `fps`, `fpm` and `cms`. For distance they are `mm`, `cm`, `m`, `ft` and `in`; for discharge, `m3/s`, `m3/h`, `l/s`, `l/h`, `cfs`, `gal/min` and `ML/day`; for volume, `m3`, `l`, `ft3`, `gal` and `ML`. Unit selection is unavailable for output type none or fixed current. The examples below assume surface velocity output (`0XGANU+4!`).	
Get: `aXGAU0!`	`a unit_name CR LF`	→ `0XGAU0!` ← `0ms`
Set: `aXGAU0 value!`	`a unit_name CR LF`	→ `0XGAU0+4!` ← `0fps`

Analog Output Maximum Value	Gets / sets the selected measurement value that corresponds to the maximum output on the analog current interface (20 mA). See 4 - 20 mA Maximum Value for the detailed setting description. The measurement unit is set by `aXGAU1!`.	
Get: `aXGAV1!`	`a value CR LF`	→ `0XGAV1!` ← `016`
Set: `aXGAV1 value!`	`a value CR LF`	→ `0XGAV1+10!` ← `010`

Analog Output Maximum Unit	Gets / sets the measurement unit for the value set by `aXGAV1!` command. See 4 - 20 mA Maximum Value for the detailed setting description. The valid units depend on the analog output type set by `aXGANU`. For discharge: • `0` = m³/s • `1` = m³/h • `2` = l/s • `3` = l/h • `4` = cfs • `5` = gal/min • `6` = ML/day For water level and distance to water: • `0` = mm • `1` = cm • `2` = m • `3` = ft • `4` = in For surface velocity: • `0` = mm/s • `1` = m/s • `2` = mph • `3` = km/h • `4` = ft/s • `5` = ft/min • `6` = cm/s For volume: • `0` = m³ • `1` = l • `2` = ft³ • `3` = gal • `4` = ML The set command takes the numeric unit index below. Get and set responses return the unit name, not its index. For velocity the returned names are `mms`, `ms`, `mph`, `kmh`, `fps`, `fpm` and `cms`. For distance they are `mm`, `cm`, `m`, `ft` and `in`; for discharge, `m3/s`, `m3/h`, `l/s`, `l/h`, `cfs`, `gal/min` and `ML/day`; for volume, `m3`, `l`, `ft3`, `gal` and `ML`. Unit selection is unavailable for output type none or fixed current. The examples below assume surface velocity output (`0XGANU+4!`).	
Get: `aXGAU1!`	`a unit_name CR LF`	→ `0XGAU1!` ← `0ms`
Set: `aXGAU1 value!`	`a unit_name CR LF`	→ `0XGAU1+3!` ← `0kmh`

Configure Power Management Mode (Automatic SDI-12 Sleep)	Gets / sets the automatic SDI-12 sleep mode parameter. When sleep is enabled, the device enters low-power standby between SDI-12 transactions. When disabled, the sensor runs continuously and is ready at all times. See Power Management for the detailed setting description. The valid parameters are: • `0` = sleep enabled • `1` = sleep disabled (continuous scanning)	
Get: `aXGCON!`	`a value CR LF`	→ `0XGCON!` ← `01`
Set: `aXGCON value!`	`a value CR LF`	→ `0XGCON+0!` ← `00`

Configure Operation Mode	Gets / sets the operation mode parameter. When the operation mode is set to standby, the device will go to standby mode. In standby mode, it will communicate on all communication interfaces, but the radar transmitters will be turned off to reduce the power, and the instrument will not be making any measurements. See Operation Mode for the detailed setting description. The accepted values are: • `0` = off (normal operation) • `1` = on (standby operation)	
Get: `aXGMUT!`	`a value CR LF`	→ `0XGMUT!` ← `00`
Set: `aXGMUT value!`	`a value CR LF`	→ `0XGMUT+1!` ← `01`

Configure Fixed Mounting Angle	Override measured forward angle with a fixed value (degrees). Changing this parameter will affect the reported velocity as cosine correction behaviour will be changed. Use this parameter only during tow-tank testing. Set 0 to use real measured angle. The accepted value range for fixed angle is 0 - 89.	
Get: `aXGFIX!`	`a value CR LF`	→ `0XGFIX!` ← `00`
Set: `aXGFIX value!`	`a value CR LF`	→ `0XGFIX+1!` ← `01`

Enable Wi-Fi	Enable or disable the Wi-Fi hotspot. If automatic SDI-12 sleep is enabled, use `aXGLWU!` to keep the instrument awake while connecting to the hotspot. `aXGWEN!` is also accepted as an alias of `aXGWFE!`: • `0` = disabled • `1` = enabled	
Get: `aXGWFE!`	`a value CR LF`	→ `0XGWFE!` ← `00`
Set: `aXGWFE value!`	`a value CR LF`	→ `0XGWFE+1!` ← `01`

Set Wi-Fi Password	Change the Wi-Fi password. This parameter is write-only. The password can contain only letters, numbers, and dash or dot character. Other characters cannot be used in the password if it is changed via SDI-12. It must contain 8 to 16 characters. The response is `a1` on success or `a0` for an invalid password; an invalid password leaves the existing password unchanged. The hotspot restarts when an active password is changed.	
Set: `aXGWFP value!`	`a 1 CR LF`	→ `0XGWFP+NewPassword!` ← `01`

Show Data on Error	Continue reporting velocity data even when angle errors are detected. • `0` = clear velocity on error (send 0 as velocity instead of the measured value) • `1` = keep velocity	
Get: `aXGSDE!`	`a value CR LF`	→ `0XGSDE!` ← `00`
Set: `aXGSDE value!`	`a value CR LF`	→ `0XGSDE+1!` ← `01`

LiDAR Scan Period	Gets / sets the interval between two consecutive automatic LiDAR scans of the channel cross-section, in minutes. The valid range is between 1 and 32767 minutes. This parameter is used only when the automatic LiDAR mode is enabled. See LiDAR Scan Period for the detailed setting description. The automatic LiDAR mode and the LiDAR limit angles cannot be configured over SDI-12. Use the configuration application, the servicing protocol, or Modbus to change these settings.	
Get: `aXGLSP!`	`a value CR LF`	→ `0XGLSP!` ← `015`
Set: `aXGLSP value!`	`a value CR LF`	→ `0XGLSP+60!` ← `060`

Start Manual LiDAR Scan	Starts a single LiDAR scan of the channel cross-section. The command can be used regardless of the configured LiDAR mode. The scan takes 30 to 45 seconds to complete. If a scan is already running, the command is ignored. See Manual LiDAR Scan for the detailed description.	
`aXGSCB!`	`a CR LF`	→ `0XGSCB!` ← `0`

Channel Profile Source	Gets / sets which LiDAR scan is returned by the high volume commands, and reports how far the preparation of that profile has progressed. See Channel Profile (High Volume Commands) for how the profile itself is then read. The valid parameters are: • `0` = none, no profile is published • `1` = the last manual LiDAR scan (started with `aXGSCB!`) • `2` = the last completed automatic LiDAR scan Setting a source starts the preparation of the profile in the background. Preparing it takes a few seconds, because the recorded scan has to be read from the internal memory and the channel bed extracted from it. Both the ASCII and the binary form are prepared, so once the profile is ready either high volume command returns it immediately. The response to both the get and the set form is `source,state,points`, where `state` describes the preparation: • `0` = idle, no source selected • `1` = preparing, the high volume commands report no data yet • `2` = ready, the profile can be read • `3` = error, the selected scan could not be read; most often it has not been recorded yet and `points` is the number of profile points that are ready to be read. A profile of *N* points is returned by `aHB!` as *2N* values, *N* `profile_x` values followed by *N* `profile_y` values. The master should therefore select a source, then repeat `aXPRSRC!` until the state becomes `2`, and only then issue `aHA!` or `aHB!`.	
Get: `aXPRSRC!`	`a source,state,points CR LF`	→ `0XPRSRC!` ← `01,2,320`
Set: `aXPRSRC value!`	`a source,state,points CR LF`	→ `0XPRSRC+1!` ← `01,1,0` → `0XPRSRC!` ← `01,1,0` → `0XPRSRC!` ← `01,2,320`

> **Note:**
> The selected source is deliberately **not** stored in non-volatile memory. It returns to `0` whenever the device restarts, and also whenever the device returns to SDI-12 sleep mode, since the profile is held in volatile memory only. After a sleep period the source has to be selected again.

Selecting a source `1` or `2` while the automatic SDI-12 sleep mode is enabled (set with `aXGCON+0!`) wakes the device in the same way the `aXGLWU!` command does, so that the profile can be prepared. Every further SDI-12 command extends the awake period by approximately one minute, which keeps the device awake for the whole transfer. Approximately one minute after the last command the device returns to sleep mode.

> **Note:**
> At 1200 baud a full profile takes a considerable amount of time to transfer, so reading one from a sleeping device keeps the instrument awake, and consuming power, for that entire period.

Perform Soft Reset	Restarts the device firmware. All settings are retained (stored in non-volatile memory). The device will be unresponsive for a few seconds after issuing this command.	
`aXGRST!`	`a CR LF`	→ `0XGRST!` ← `0`

Reset DFP	Clears all accumulated DFP history data. Use this after a significant change in site conditions or sensor placement.	
`aXGRDP!`	`a CR LF`	→ `0XGRDP!` ← `0`

Factory Reset	Restores all settings to their factory defaults and restarts the device. This is irreversible.	
`aXGFAC!`	`a CR LF`	→ `0XGFAC!` ← `0`

Totalizer Reset	Resets the accumulated total volume to zero.	
`aXGRTL!`	`a CR LF`	→ `0XGRTL!` ← `0`

Set Staff Gauge Reading	Takes current reading from the staff gage to automatically adjust the radar height parameter according to the given value and current measurement of the distance to water surface. The radar height measurement unit set by `aXGRHU!` is used. The accepted reading is echoed with its sign. The height adjustment is applied after the acknowledgment, once the device is awake. Invalid input returns `aERR`. This command is write-only.	
`aXGSSG value!`	`a signed_value CR LF`	→ `0XGSSG+12.221!` ← `0+12.221`

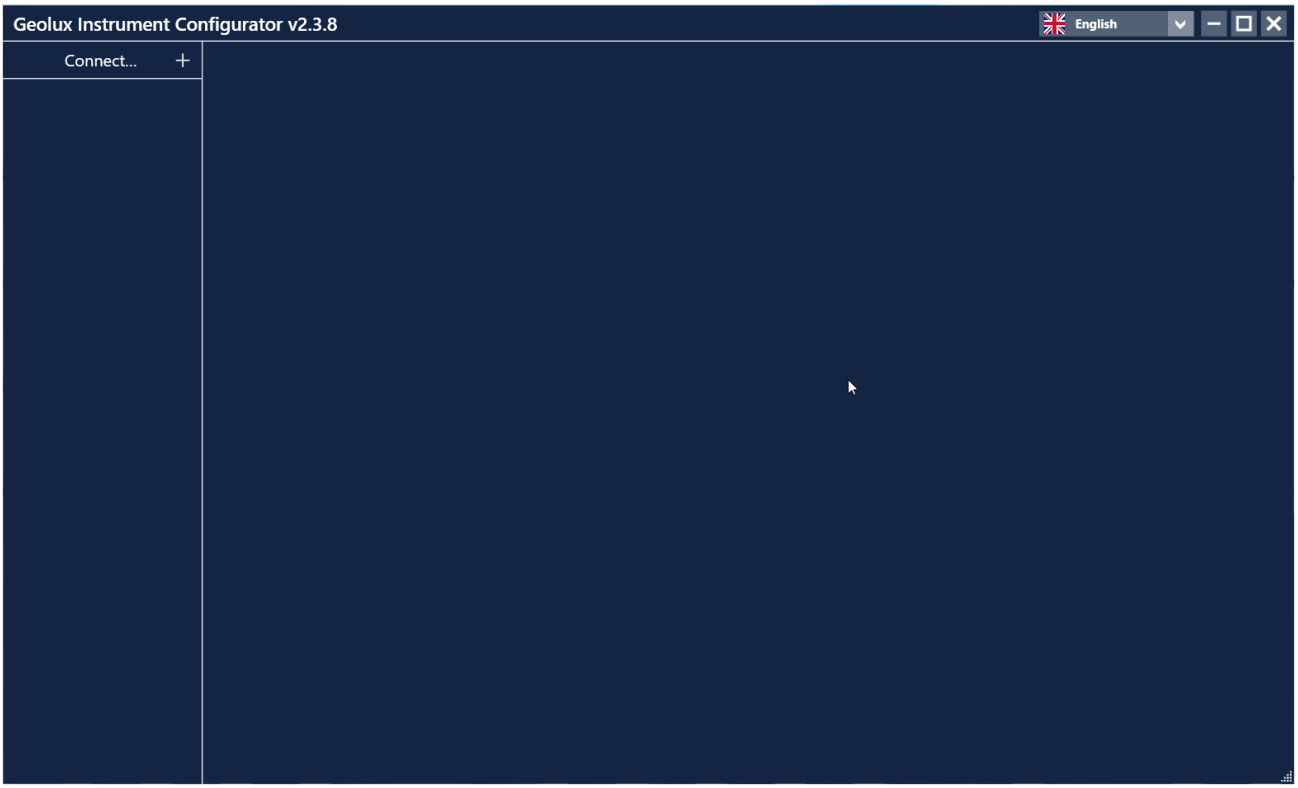
Wake Up	This command wakes the device from SDI-12 sleep mode and keeps it awake for approximately 1 minute. Use this command when the automatic SDI-12 sleep mode is enabled (set with `aXGCON+0!`) and temporary access to the device is required. While awake, RS-232 and RS-485/Modbus interfaces remain active, allowing a connection to be established with the PC configuration application. After the timeout period expires, the device automatically returns to sleep mode if no further activity is detected.	
`aXGLWU!`	`a CR LF`	→ `0XGLWU!` ← `0`

7 Geolux Instrument Configurator

Geolux provides a user-friendly PC application, Geolux Instrument Configurator, for configuring the operating parameters of the surface velocity radar and monitoring live measurement data.

The latest version of the application can be downloaded here: ***Setup-GeoluxInstrumentConfigurator.exe***.

When the application is launched, the main window is displayed. Initially, no data is shown because the connection to the radar has not yet been established.



Picture 12. Geolux Instrument Configurator.

7.1. Establishing Connection

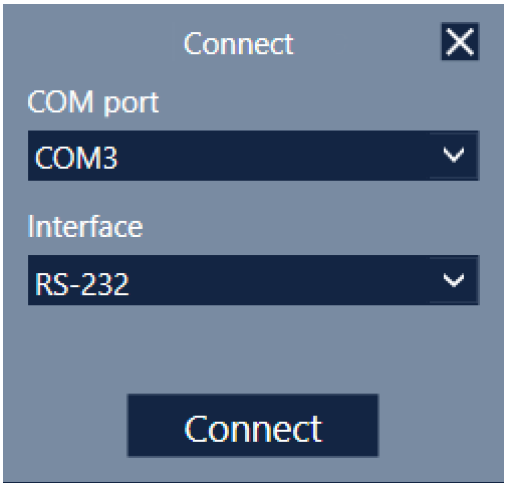
To connect the PC to the flow meter, first connect the radar to the PC using a serial communication cable. You can use either RS-232 port or RS-485 port to connect the device to the PC.

Click the Connect button in the upper-left corner of the application window, and select **FX100 / FX200** from the device selection window.

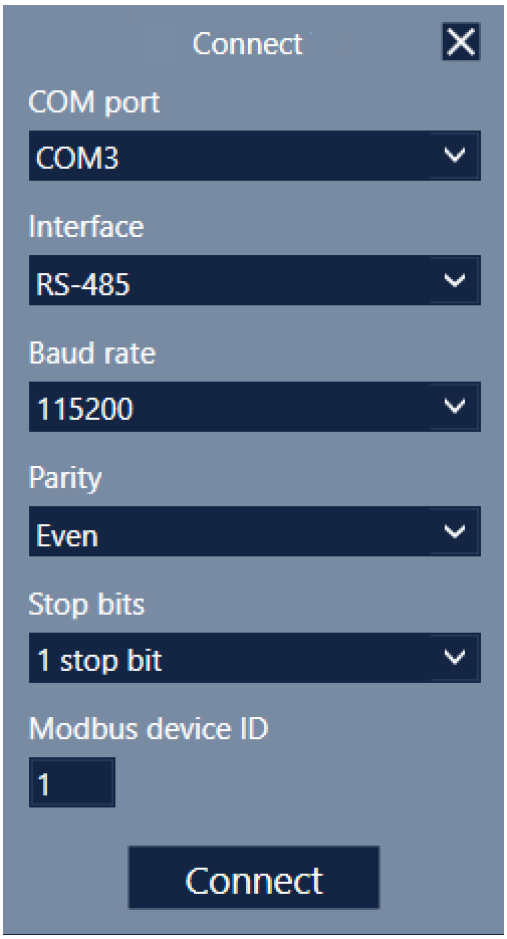


Picture 13. Device selection window.

Then, select the appropriate COM port and communication interface. If RS-485 is selected, you must also configure the baud rate, and the Modbus device ID. If multiple devices are connected to the same RS-485 bus, ensure that each device has a unique Modbus ID.



Picture 14. RS-232 connection parameters.



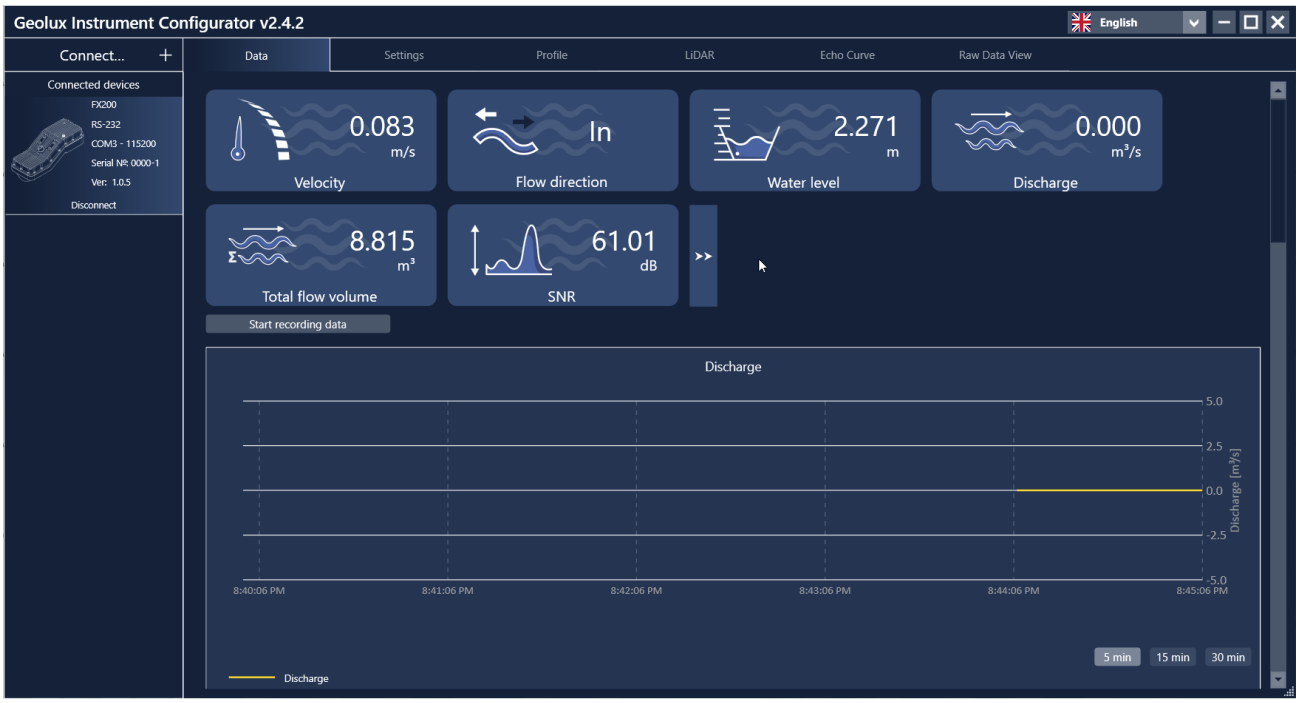
Picture 15. RS-485 connection parameters.

After clicking Connect, the application attempts to establish communication with the device. If the connection is successful, the device appears under Connected devices on the left. Live flow velocity measurements are displayed in the Data tab, as numerical values and time-series graphs.

Hovering the mouse pointer over a parameter icon displays a detailed description of that parameter. By default, only the most important parameters are shown. Clicking the arrow button next to the displayed parameters expands the view to show all available data.

Clicking Start recording data begins recording the primary displayed measurements and saves them to a .csv file.

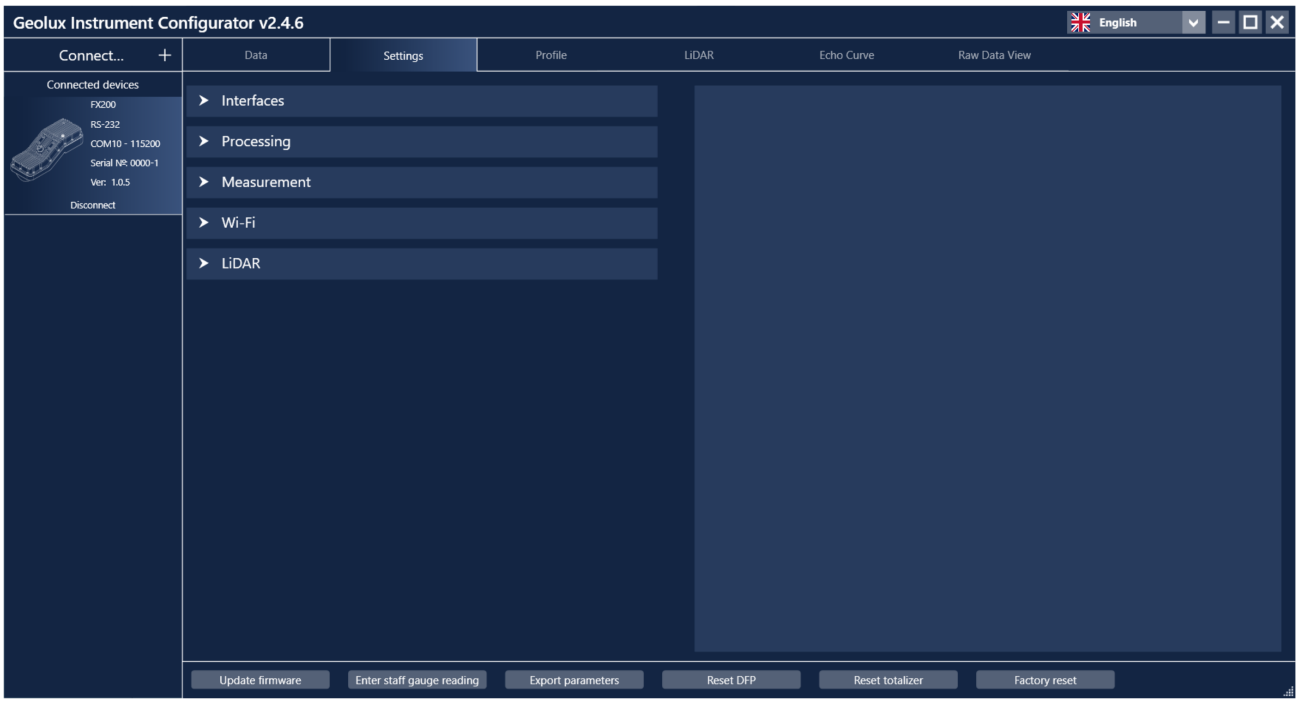
If the RS-232 interface is used and excessive checksum errors are detected, the interface, COM port, and baud rate text under Connected devices will turn red, indicating a communication error.



Picture 16. Data view page.

7.2. Device **Configuration**

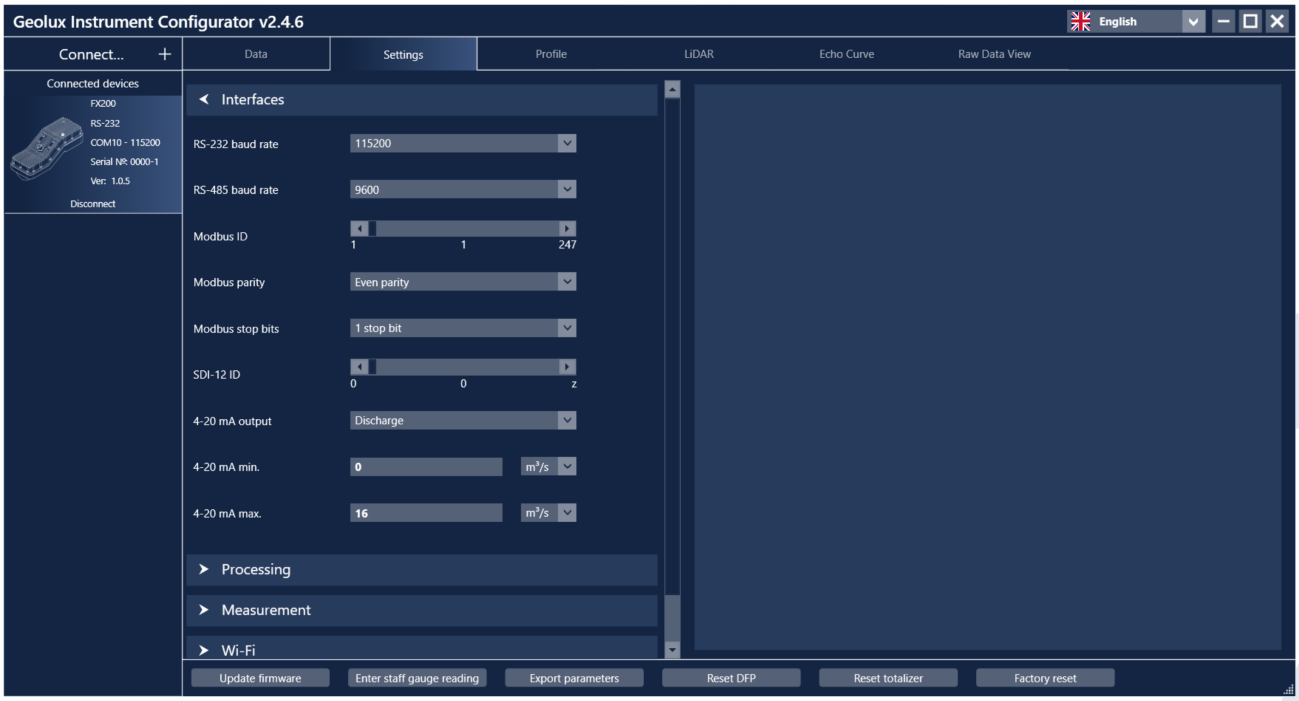
Clicking the Settings button opens the configuration interface, where the user can adjust operating parameters or upgrade the device firmware.



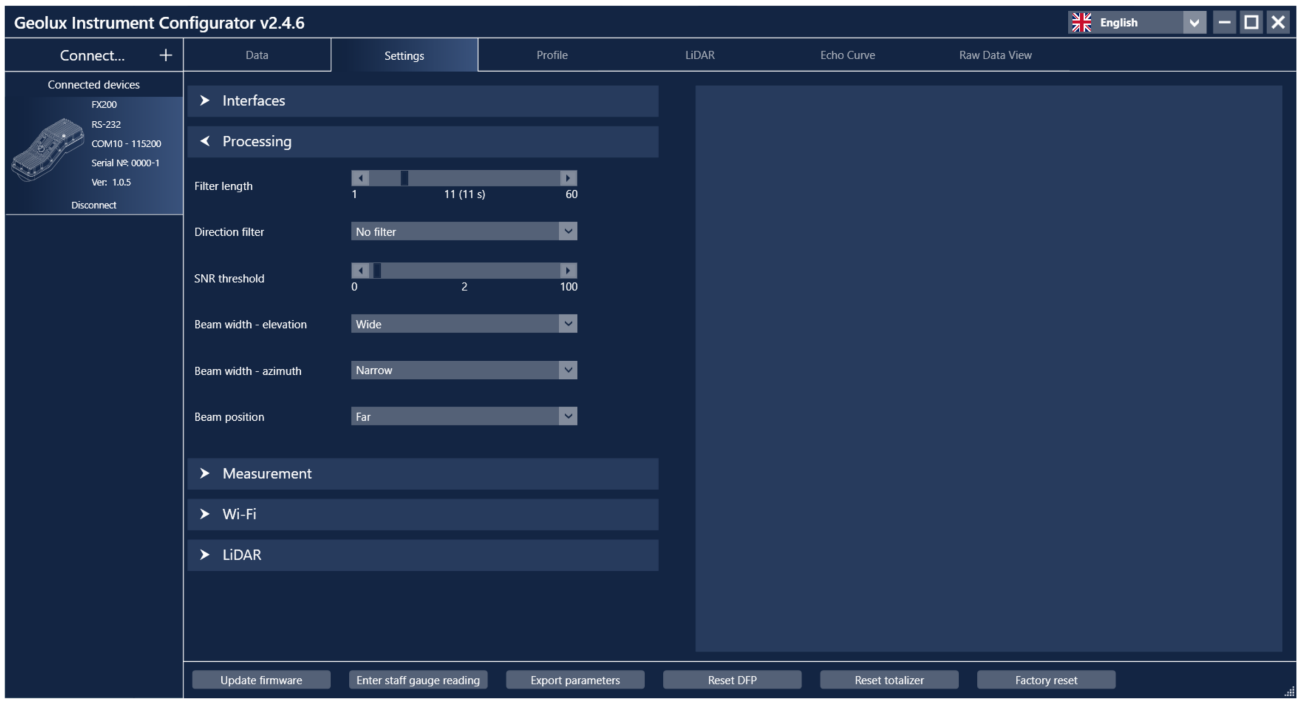
Picture 17. Settings page for FX200 flow meter.

Settings are organized into five groups: Interfaces, Processing, Measurements, LiDAR and Wi-Fi. Selecting a group displays all associated settings.

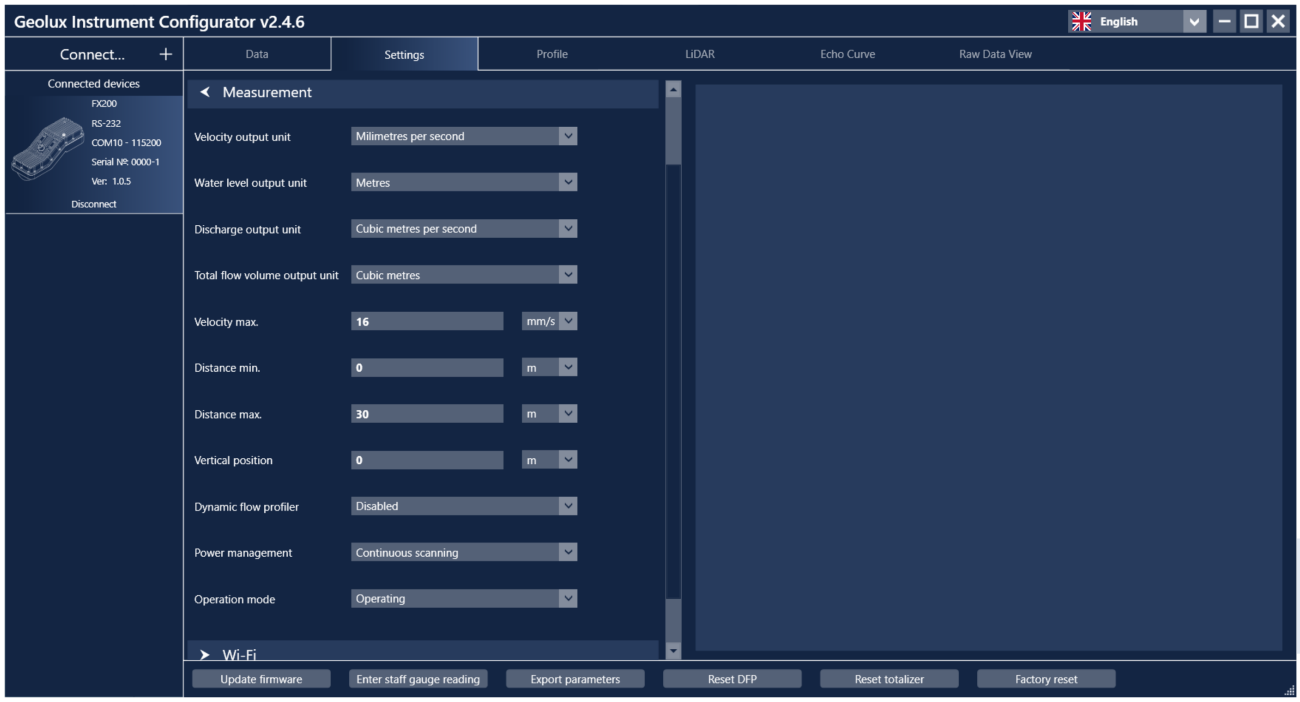
When a setting is selected or modified, a detailed description is displayed on the right side of the window. Detailed descriptions of all parameters are also given in 5. Flow Meter Settings. After changing a parameter, a Set button appears next to the modified field. Clicking Set confirms and saves the new value to the device.



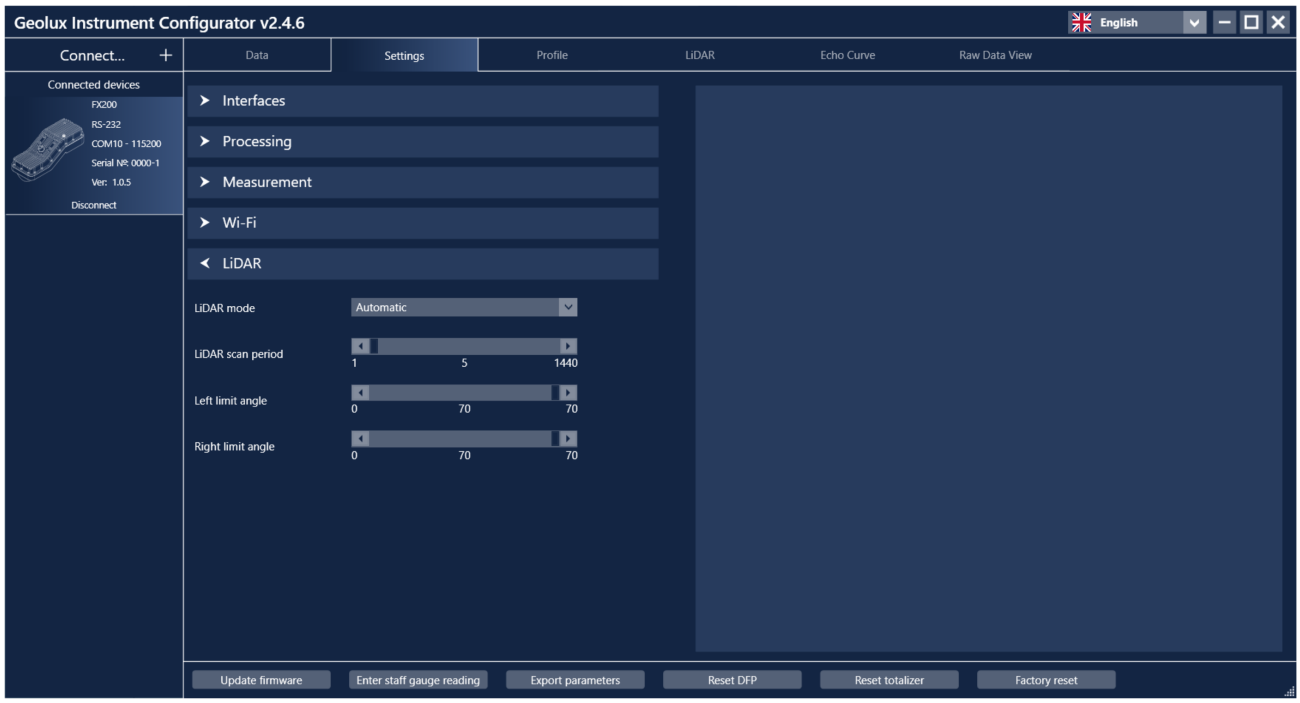
Picture 18. Interface settings.



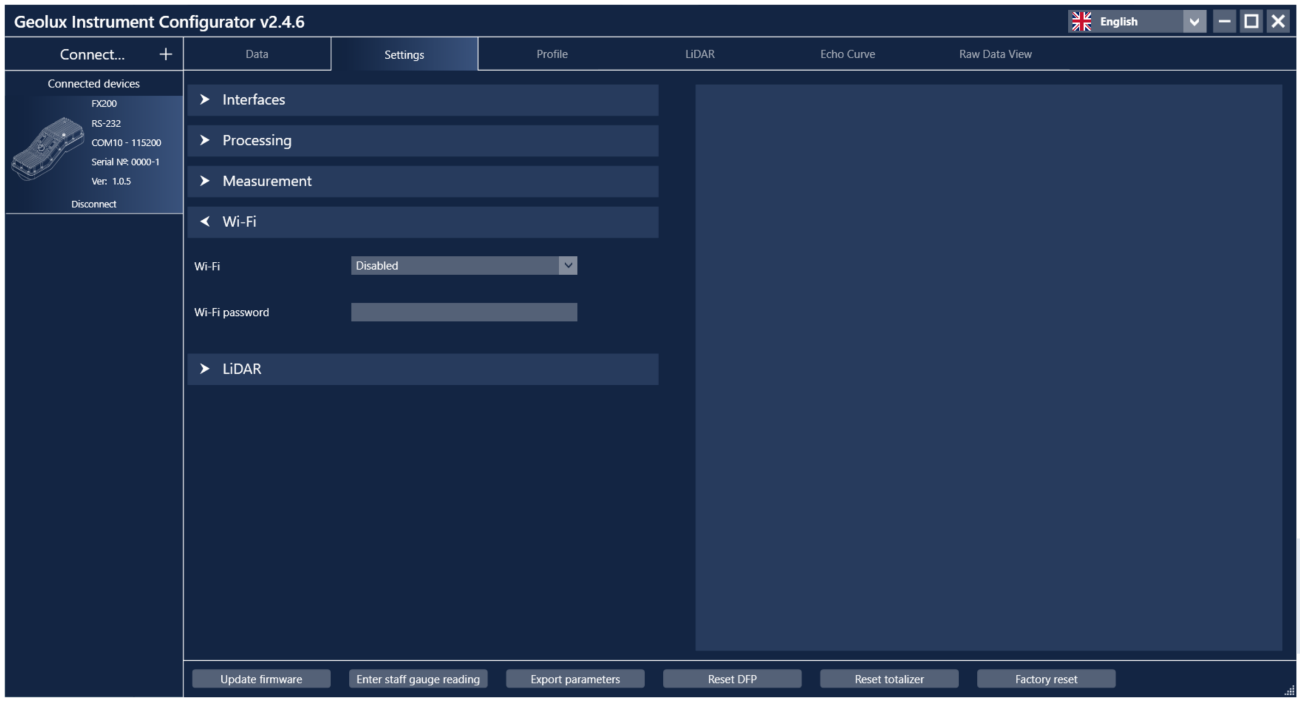
Picture 19. Data processing settings.



Picture 20. Measurement settings.



Picture 21. LiDAR settings.



Picture 22. Wi-Fi settings.

Additional functions available in the Settings tab include:

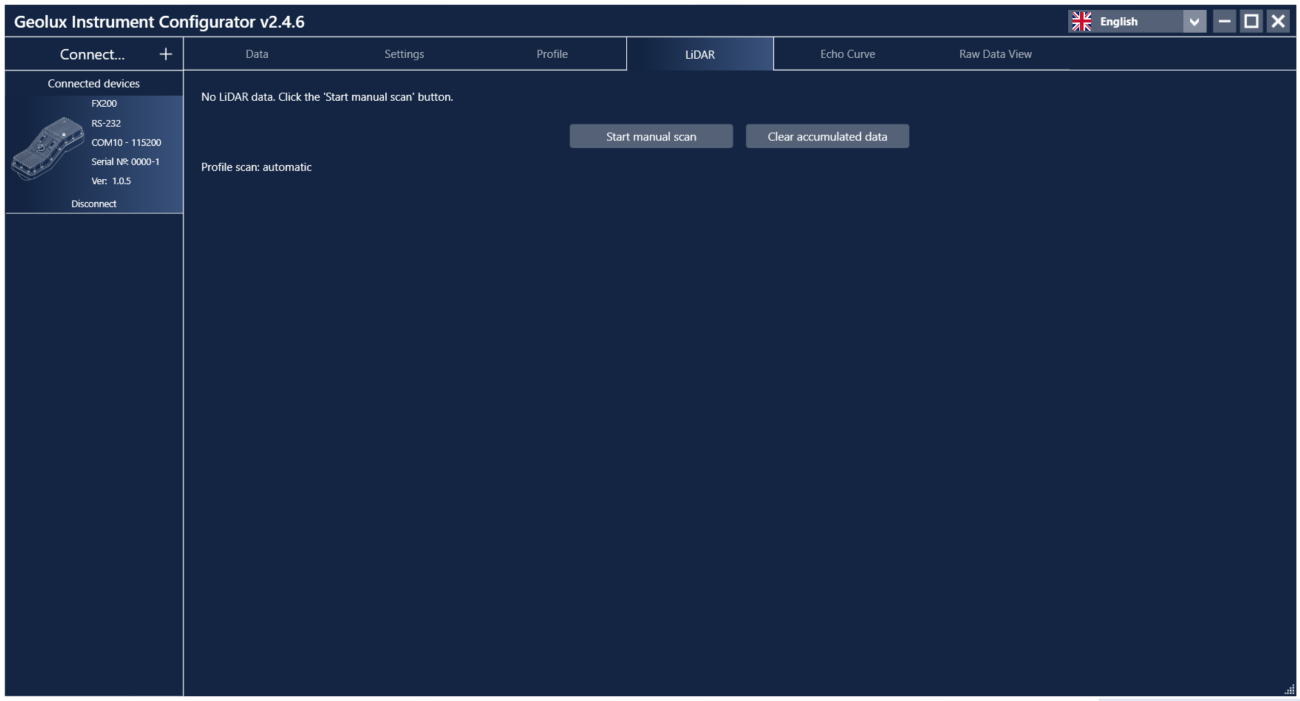
- **Update firmware** – Updates the device firmware to a newer version.
- **Enter staff gauge reading** – Changes radar height parameter based on the current reading of the staff gage at the monitoring site.

- **Export parameters** – Saves all device parameters to a text file for review.
- **Reset DFP** – Clears all accumulated DFP history data.
- **Reset totalizer** – Resets the volume totalizer to zero.
- **Factory reset** – Restores all parameters to their default values.

7.3. Using LiDAR

The FX200 uses the built-in LiDAR sensor to measure the cross-section profile of the channel. A scan can be started manually from the configurator application, or the instrument can perform scans automatically at a configurable interval. Automatic scanning and the scanned angular sector are configured in the **LiDAR** group on the **Settings** page, as described in 5. Flow Meter Settings.

To run a LiDAR scan manually, open the **LiDAR** page in the configurator application and click **Start manual scan**.



Picture 23. Empty LiDAR scan page.

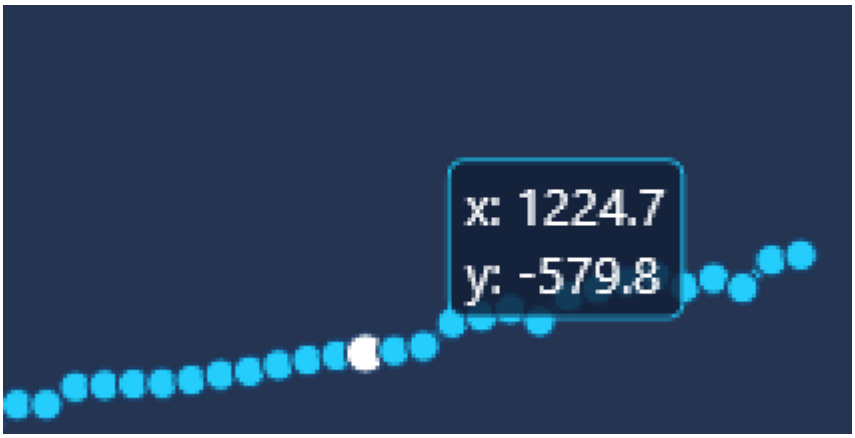
The instrument starts scanning immediately. A manual scan takes 30 to 45 seconds. When the scan is complete, a preview of the scanned channel profile is displayed:



Picture 24. LiDAR scan preview.

The preview window shows the result of the most recent LiDAR scan. A manually started scan is not yet used as the actual channel profile for discharge calculation. In the preview window, the scan results can be reviewed, and the scanned channel points can be modified, added, or deleted:

- Click a point to select it. The coordinates of the selected point are displayed.
- Drag the selected point to change its position, or right-click it to enter its coordinates manually.
- Press the Delete key to remove the selected point.
- Double-click a line segment between two points to insert a new point at the middle of the segment.
- Drag a rectangular selection over a range of points to select several points at once.



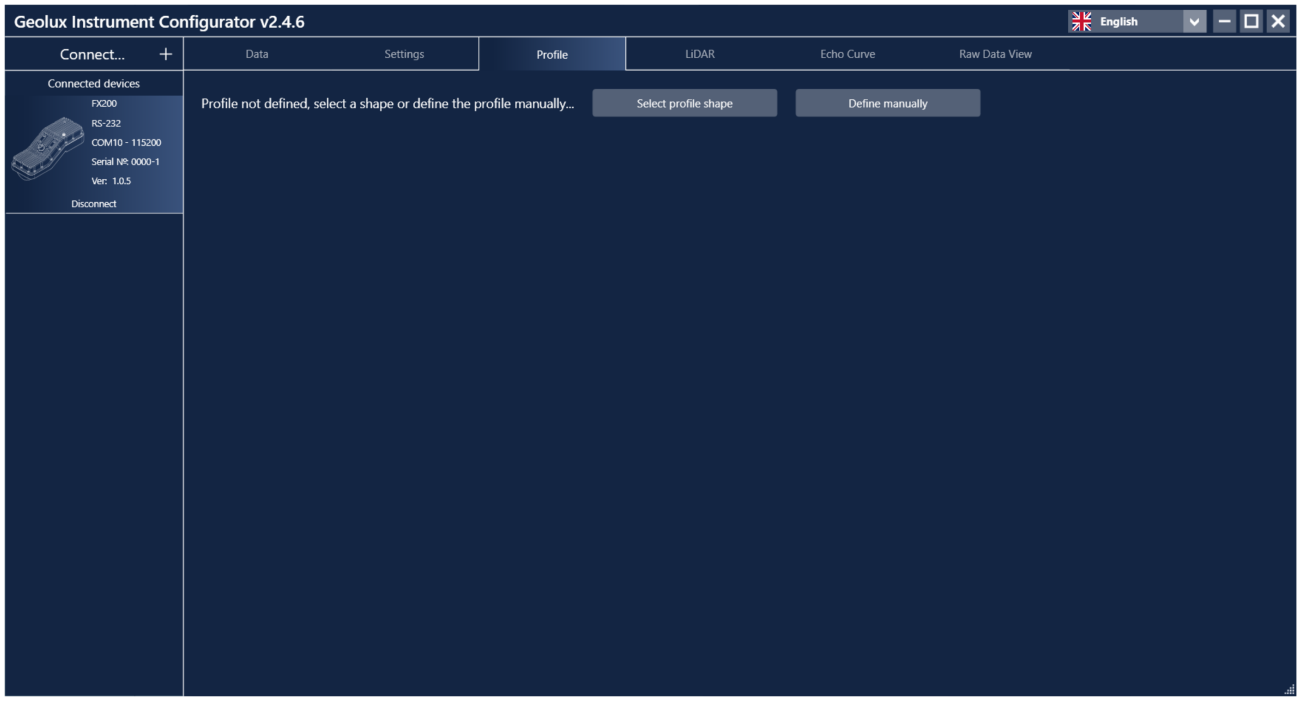
Picture 25. Selecting a channel point.

To accept the scan and start using it as the actual channel profile, click **Save changes**. The scan is saved as the main profile for discharge calculation, and the **Profile** page is displayed immediately.

To discard the scanned data and empty the LiDAR page, click **Clear accumulated data**. The scanned points can also be exported for external processing by clicking **Save to .csv**.

7.4. Configuring **Discharge Calculation**

If no channel profile is stored on the device, or if the existing profile has been deleted, the **Profile** page will be empty and a new channel profile must be defined.



Picture 26. Empty channel profile page.

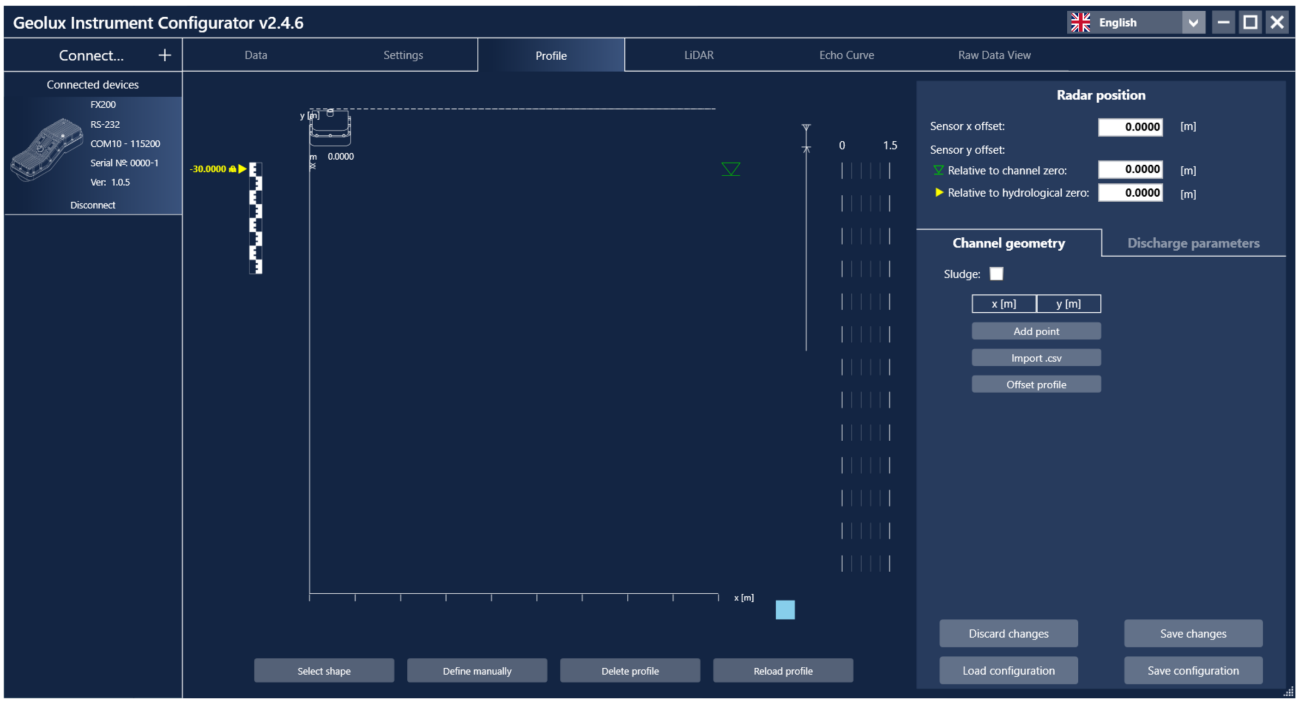
There are two ways to define the channel profile:

- **Manual definition** – Enter the X and Y coordinates of each profile point. This method is typically used for natural channels where bathymetric survey data is available.
- **Predefined shapes** – Select one of the predefined channel shapes, such as rectangular, trapezoidal, or circular. This method is intended for concrete channels, culverts, and sewer systems.

If a LiDAR scan has been saved on the **LiDAR** page, as described in 7.3. Using LiDAR, the scanned cross-section is displayed on the **Profile** page as the channel profile and is used for the discharge calculation. The scanned profile can always be overridden manually: its points can be edited in the same way as manually entered points, or the profile can be deleted and defined again using either of the two methods described above.

Manual Channel Profile Configuration

Click **Define manually** to open an empty channel profile.



Picture 27. Manual channel profile definition.

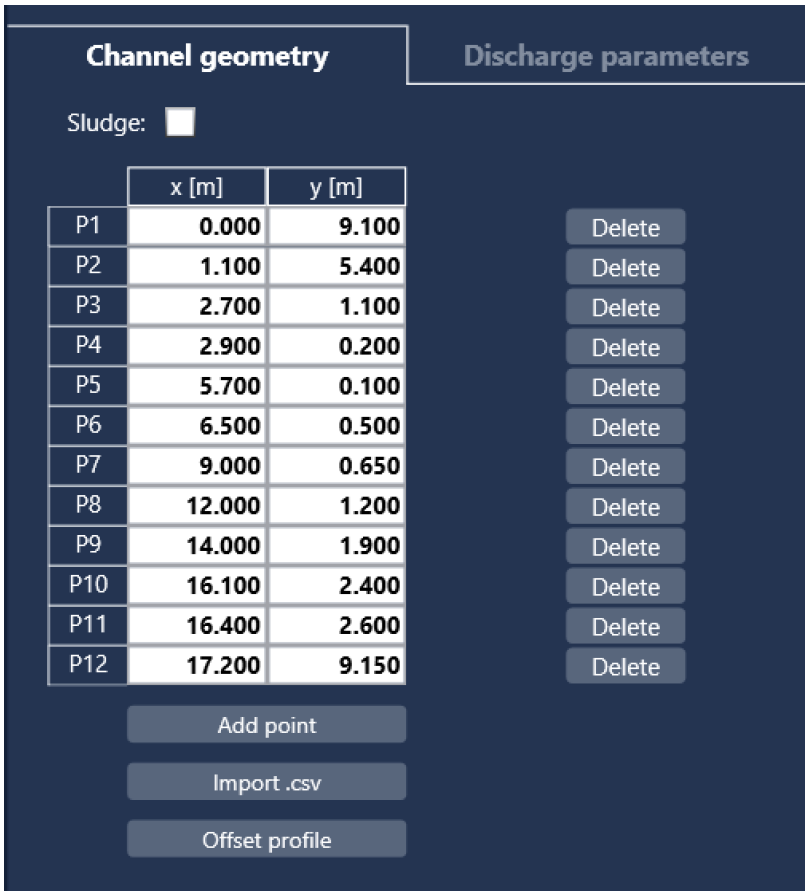
Begin by entering the channel profile points. Add points individually by clicking **Add point**, then enter the X and Y coordinates.

The X coordinate starts at **0** at the left edge of the channel and increases towards the right.

The Y coordinate may use any vertical datum. For example, it can represent:

- elevation above mean sea level,
- depth referenced to the channel bottom,
- any other local vertical datum.

After entering the coordinates, click **Set** to save the point. Additional points cannot be added until the current point has been confirmed. To remove a point, click **Delete**.

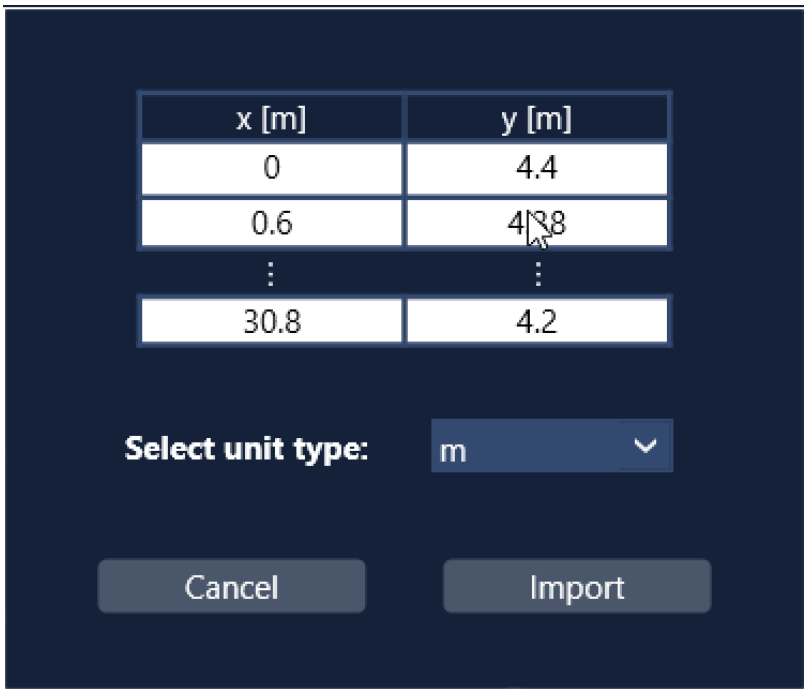


Picture 28. Channel profile coordinate editor.

Importing Profile from a CSV File

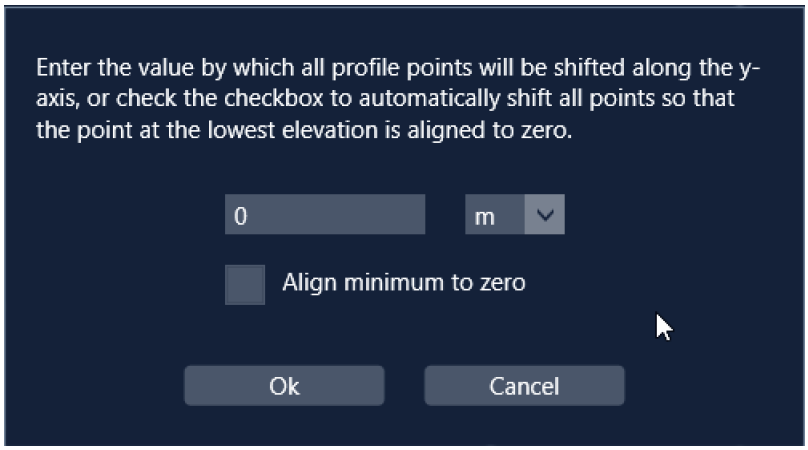
If channel bathymetry is available in a CSV file, the profile can be imported instead of entering points manually.

Click **Import .csv**, select the file, and review the imported coordinates. Before importing, select the measurement unit used in the CSV file.



Picture 29. CSV profile import preview.

After clicking **Import**, an optional vertical offset can be applied to all Y coordinates. This is useful when the CSV data uses a different vertical datum than the one intended for the FX200 configuration.



Picture 30. Applying a vertical offset to imported profile coordinates.

Configuring Radar Position

After defining the channel profile, configure the radar position in the **Radar position** pane.



Picture 31. Radar position settings.

- **Sensor X offset**

  Horizontal distance from the leftmost channel point.
- **Sensor Y offset (relative to channel zero)**

  Vertical sensor position using the same datum as the channel profile. For example, if the profile uses elevations above mean sea level, this value must use the same reference. If the profile uses the channel bottom as zero, this value is the distance from the radar to the channel bottom.

  This parameter is used to calculate the wetted cross-sectional area from the measured distance to the water surface.
- **Sensor Y offset (relative to hydrological zero)**

  Equivalent to the **Vertical position** setting on the **Settings** page. It is used only to convert the measured distance to the reported water level and does not affect discharge calculation.

  This value may be the same as the channel reference, but different vertical datums can also be used.

After the radar position has been configured, the channel profile is displayed together with the sensor location.



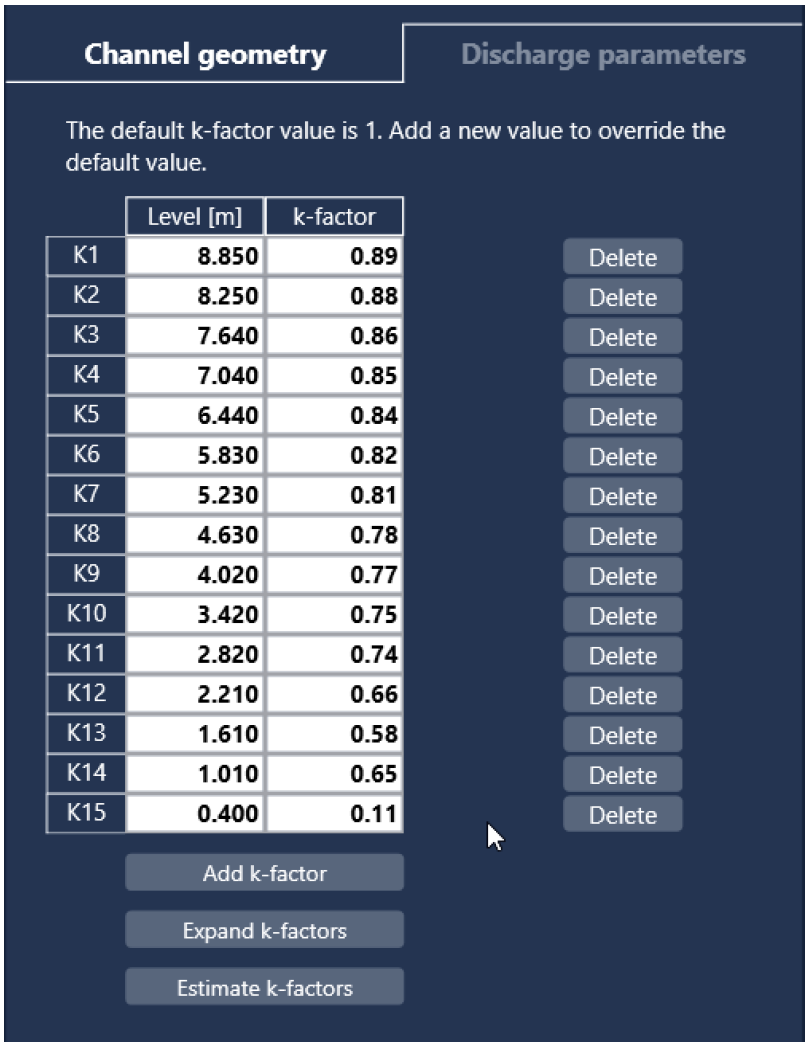
Picture 32. Channel profile with radar position.

Defining Discharge Parameters

The final step is to define the k-factor table.

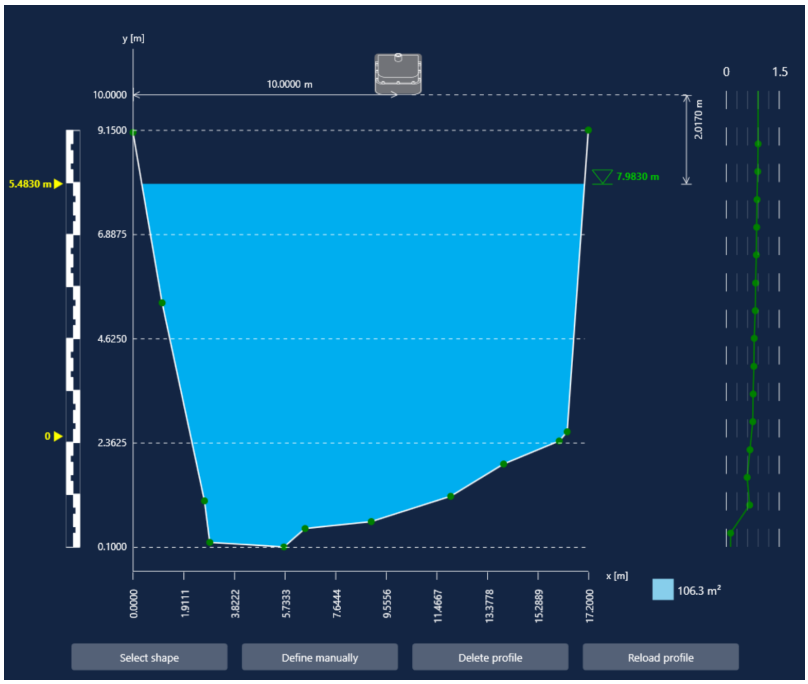
Open **Discharge parameters** and add k-factor values for the required water levels using **Add k-factor**. Alternatively, click **Estimate k-factors** to generate an initial estimate.

If no k-factor is defined, a default value of **1.0** is used.



Picture 33. K-factor editor.

The configured k-factors are displayed next to the channel profile and aligned with their corresponding water levels.



Picture 34. Channel profile with configured k-factors.

After configuring the profile and discharge parameters, click **Save changes** to store the configuration on the device.

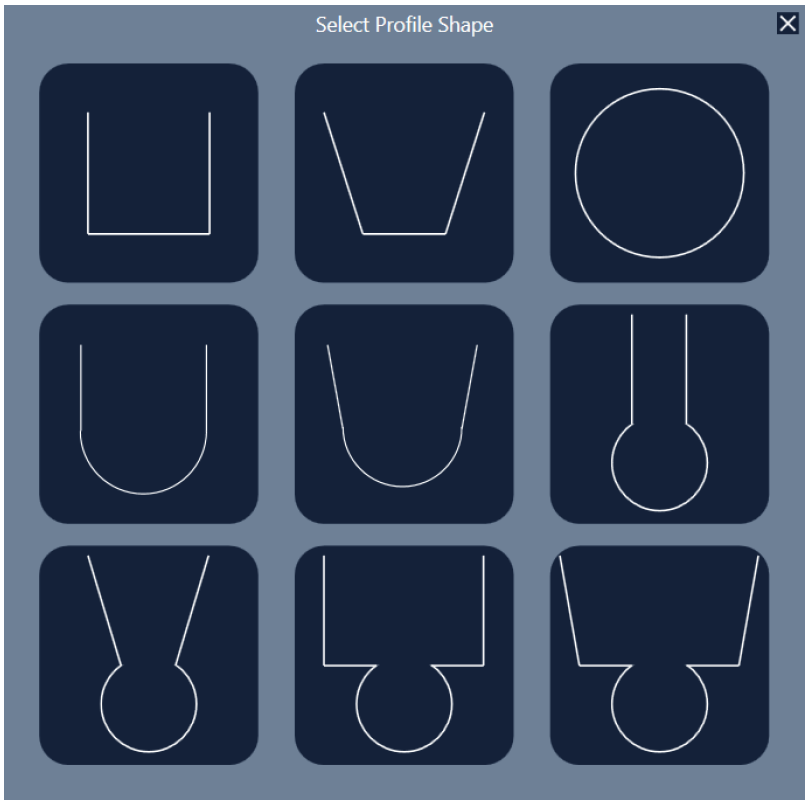
To discard all unsaved modifications and restore the current device configuration, click **Discard changes**.

The configuration can also be saved to an XML file using **Save configuration**. To restore a previously saved configuration, click **Load configuration**, then save it to the device using **Save changes**.

To remove the current channel profile, click **Delete profile**. To immediately create a new profile, click **Define manually** or **Select profile shape**.

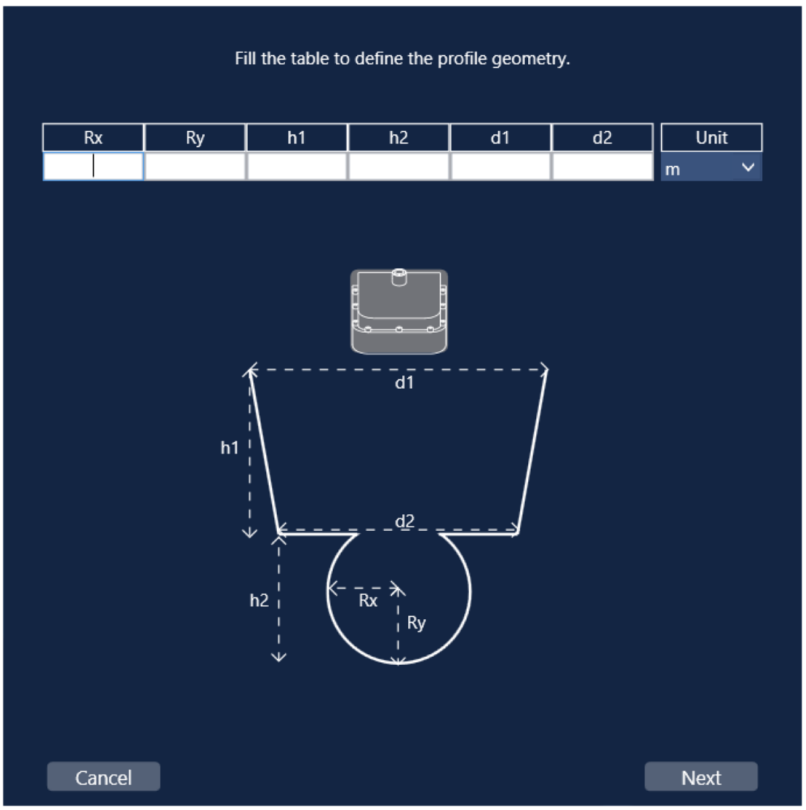
Defining Channel Profile from a Predefined Shape

Click **Select profile shape** to choose one of the available predefined channel geometries.



Picture 35. Selecting a predefined channel shape.

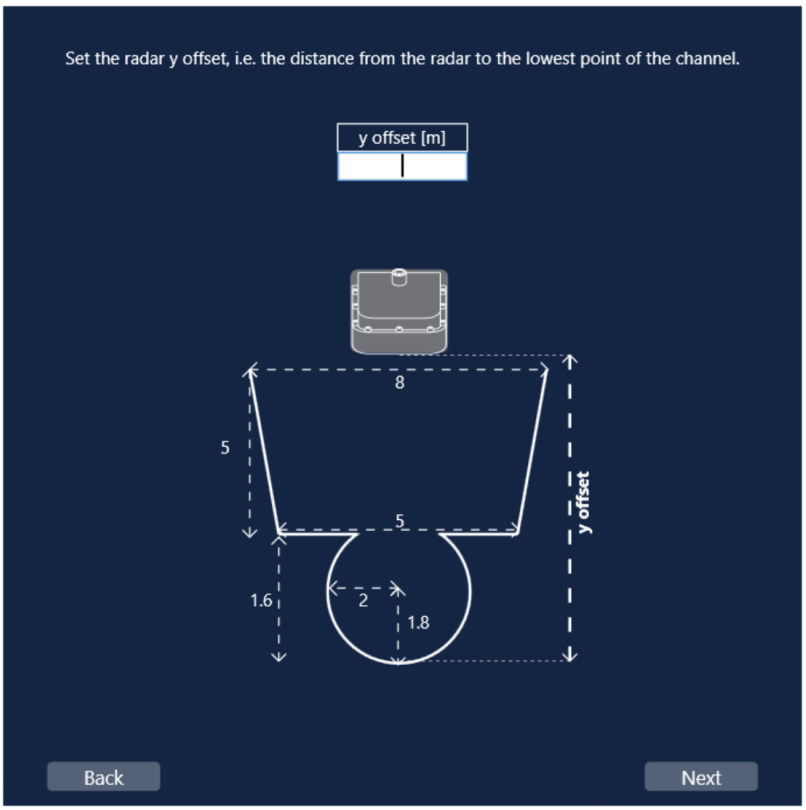
After selecting a shape, enter the required dimensions and select the measurement unit. The entered dimensions are illustrated in the preview.



Picture 36. Entering channel dimensions.

Click **Next** to configure the radar height. The required measurement is illustrated together with the previously entered dimensions.

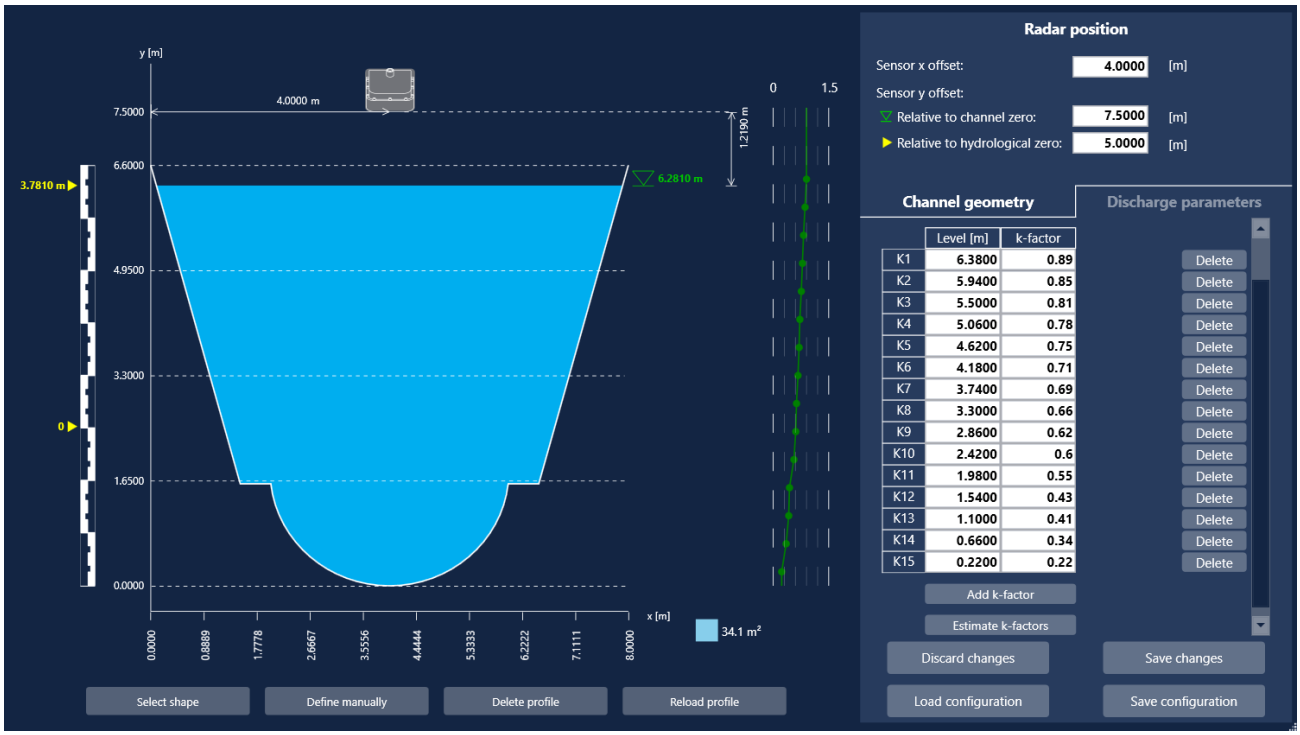
If necessary, click **Back** to modify the channel dimensions.



Picture 37. Configuring radar position.

After the radar position has been configured, the channel profile is displayed in the main window.

The remaining step is to configure the k-factors in the **Discharge parameters** section, either manually or by using **Estimate k-factors**, as described earlier.

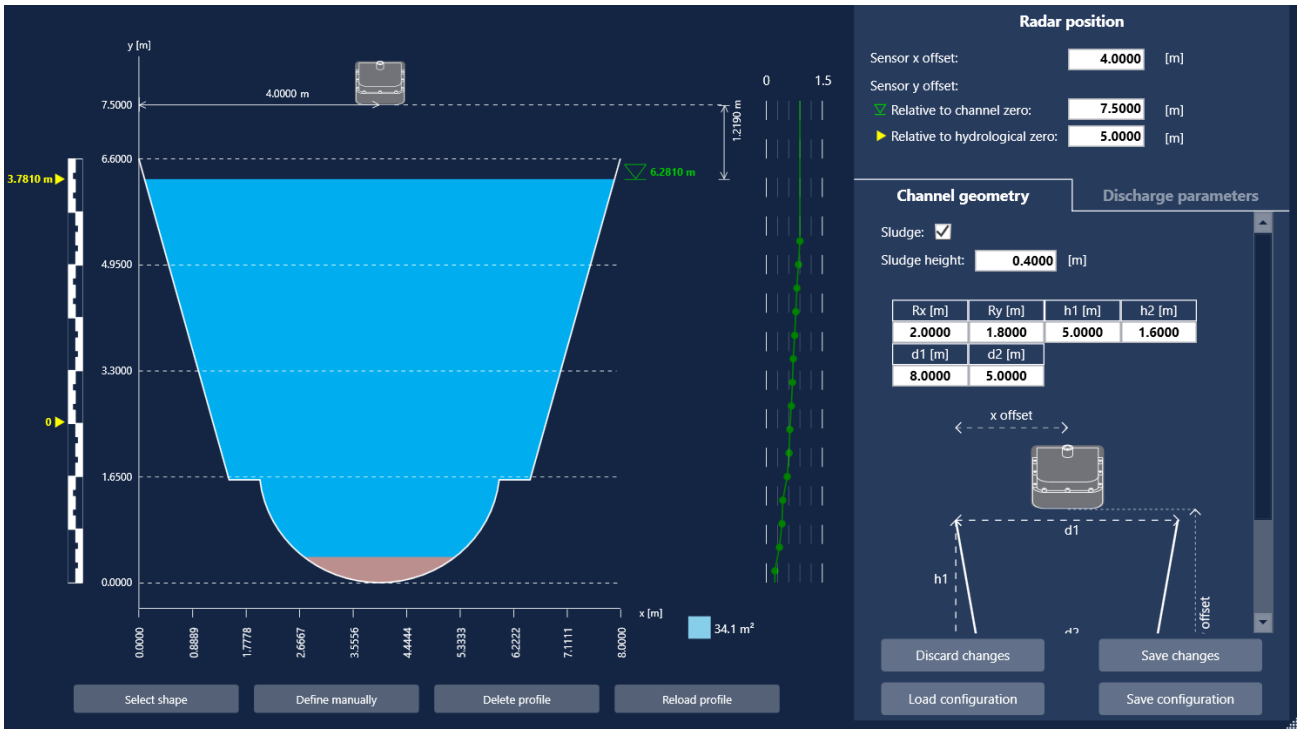


Picture 38. Completed discharge configuration using a predefined channel shape.

Defining Sludge Depth

In sewer systems and other sediment-prone channels, sludge accumulation reduces the effective flow area. Configure the sludge depth so the FX200 can correctly calculate the wetted cross-section.

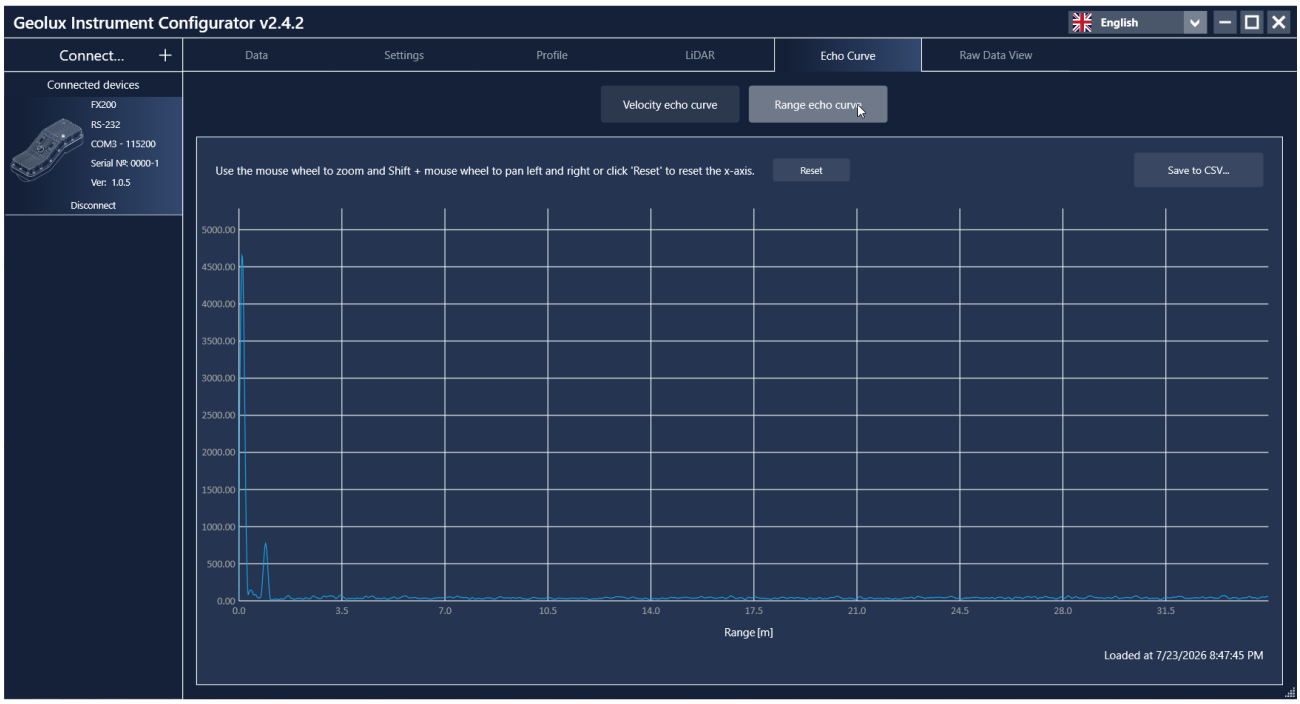
Enable **Sludge** in the **Channel geometry** pane and enter the sludge thickness. The accumulated sludge is immediately displayed in the channel profile.



Picture 39. Channel profile with sludge compensation.

7.5. Echo **Curve Acquisition**

The Echo Curve tab allows visualization and acquisition of the radar echo signal. This is useful for diagnosing potential problems with weak signal or incorrect measurements. The Echo Curve tab can display two types of echo curves: the rangefinder echo curve and the velocity distribution echo curve. The echo curves will start loading and updating automatically as soon as the Echo Curve tab is opened.

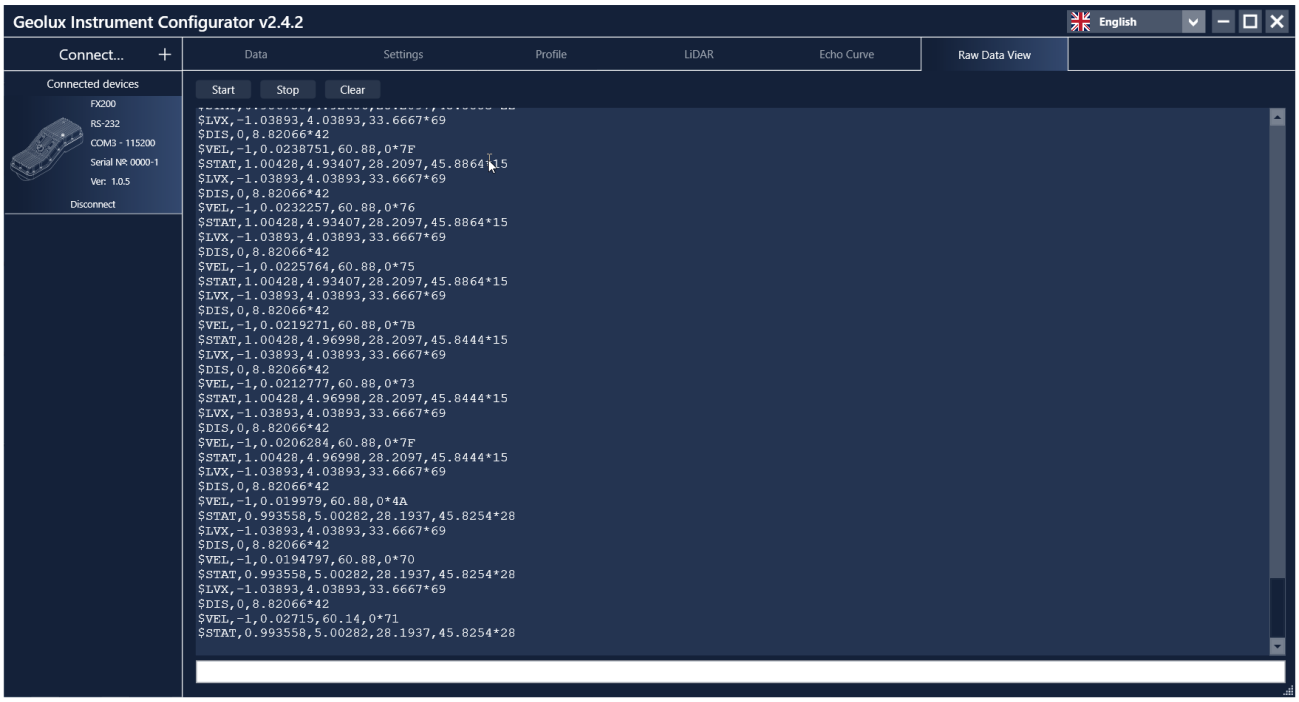


Picture 40. Echo curve display.

7.6. Raw **Data View**

The Raw Data tab enables monitoring of the raw data stream transmitted by the device. This is useful for diagnosing problems with the communication interfaces.

- Clicking Start begins raw data acquisition.
- Clicking Stop ends data acquisition.
- Clicking Clear removes previously displayed data.

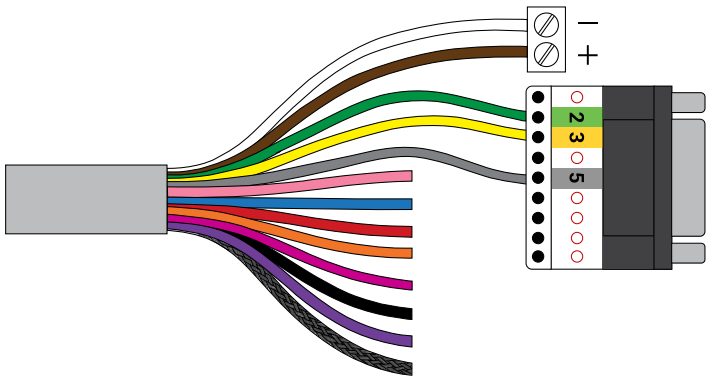


Picture 41. Raw data view page.

8 Troubleshooting

The Instrument Does Not Connect To The PC Application Over RS-232 Connection

1. Make sure that the cable is properly connected to the instrument. Check that the cable M12 circular connector is firmly connected and screwed to the instrument. It is not sufficient to simply attach the cable connector to the instrument connector, the cable connector must be screwed into the instrument connector.
2. Check power supply. Make sure that the power is being supplied to the instrument. A direct-current voltage between 9 and 27 Volts must be connected to the brown (+) and white (–) wires of the instrument cable. The power supply must be able to deliver at least 500 mA of current.
3. Check the RS-232 connection. The yellow, green, and grey wires from the instrument cable must be properly connected to the serial port on the computer. Make sure that the grey wire (signal ground) is connected – the RS-232 connection will not work if the signal ground is not connected. Also, make sure that the yellow (device Rx) and green (device Tx) wires are properly connected. If you are using a standard DB-9 connector on the PC, the wires should be connected as in the following diagram:



4. Make sure that you are using the correct COM port. If there are multiple COM ports available on your computer, make sure that you are selecting the correct COM port in the PC application. If you are not certain which COM port number is assigned to the port that is being used to connect to the instrument, try setting up the connection with each COM port available in the system, until the connection is established.
5. Make sure that the COM port is not already open. Only one application may use a single COM port in the system. Make sure that no other open application uses the same COM port that you are trying to open.

6. Try restarting the application and/or the computer. Close and reopen the configurator application and try to establish the connection again. Restart your computer and try to establish the connection to the instrument again.

The Instrument Does Not Respond Over Modbus (RS-485) Interface

1. Make sure that the cable is properly connected to the instrument. Check that the cable M12 circular connector is firmly connected and screwed to the instrument.
2. Check power supply. A direct-current voltage between 9 and 27 Volts must be connected to the brown (+) and white (–) wires of the instrument cable. The power supply must be able to deliver at least 500 mA of current.
3. Check the RS-485 connection. Make sure that the RS-485 lines on the instrument cable are properly connected. The dark red wire should be connected to the A+ line, and the orange wire should be connected to the B- line. Make sure that the A+ and B- lines are not swapped. Make sure that you have correctly identified the dark red wire, as there is also a bright red wire that is used for the 4-20 mA positive output terminal. It is possible to misidentify the wire and to connect the bright red wire to A+ instead of the dark red wire.
4. Check that you are using the correct slave device ID. The default Modbus device ID is 1. The Modbus device ID can be changed by connecting the instrument to the PC application over the RS-232 connection. Open the Geolux Instrument Configurator application, establish the connection, then check the Device ID parameter and make sure that it is the same as the slave device ID used in the issued Modbus requests.
5. Make sure that there are no two devices on the bus with the same Device ID. Modbus allows multiple devices to be connected to the same bus simultaneously. Each device must have a unique slave device ID, so that the bus master can distinguish between the devices. If two or more devices are assigned the same ID, a bus conflict will prevent the master from communicating correctly with the slave devices. To resolve this problem, change the instrument's slave device ID to a unique number through the Geolux Instrument Configurator.
6. Check that you are using the correct Modbus connection parameters (baud rate, parity, stop bits). The default Modbus connection parameters are 9600 bps, even parity, 1 stop bit. These parameters can be checked and changed in the Geolux Instrument Configurator application over the RS-232 connection.

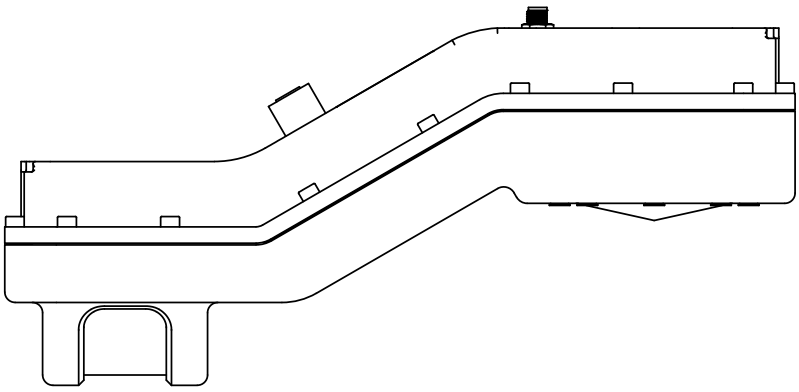
The 4-20 mA Output Is Not Correct

1. Make sure that the cable is properly connected to the instrument. Check that the cable M12 circular connector is firmly connected and screwed to the instrument.
2. Check power supply. A direct-current voltage between 9 and 27 Volts must be connected to the brown (+) and white (–) wires of the instrument cable. The power supply must be able to deliver at least 500 mA of current.
3. Make sure that the wires from the cable are properly connected. For 4-20 mA readout, the bright red wire from the instrument cable must be connected to the positive (+) input, and the purple wire must be connected to the negative (–) input of the 4-20 mA interface.
4. Check the instrument configuration parameters related to the 4-20 mA output. Connect the instrument over RS-232 to the Geolux Instrument Configurator application and check the 4-20 mA output type, minimum and maximum values. Configure these parameters properly.

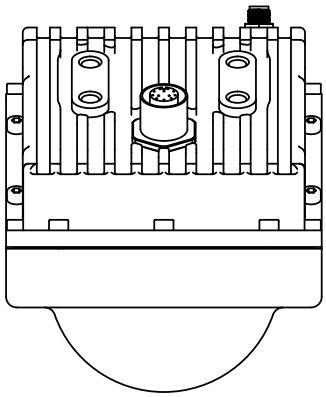
The Instrument Is Operating, But Does Not Report Measurements Or Reports Incorrect Values

1. Check that the instrument is positioned properly. Make sure that the instrument is installed above the water surface and that it is properly leveled – use the integrated bubble level indicator. The instrument must not be tilted forward or sideways. The recommended installation height is at least 1 meter above the water surface.
2. Check that there are no obstructions between the instrument and the water surface, so that the radar waves can freely travel from the instrument's antennas to the water surface and back.
3. Check that the distance between the instrument and the water surface is within the instrument's operating range (0.2 m to 30 m). Use a tape measure or a similar tool to measure the distance between the instrument and the water surface.
4. Check the configured channel cross-section and the instrument mounting height. Discharge is calculated from the measured surface velocity, water level and the configured cross-section profile, so an incorrect profile or mounting height results in incorrect discharge values.

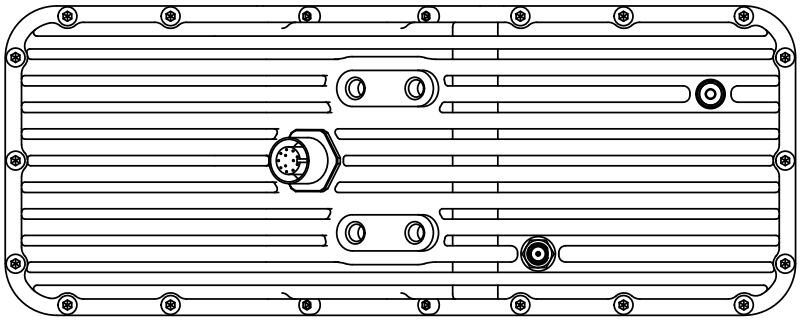
A Appendix A: Mechanical Assembly



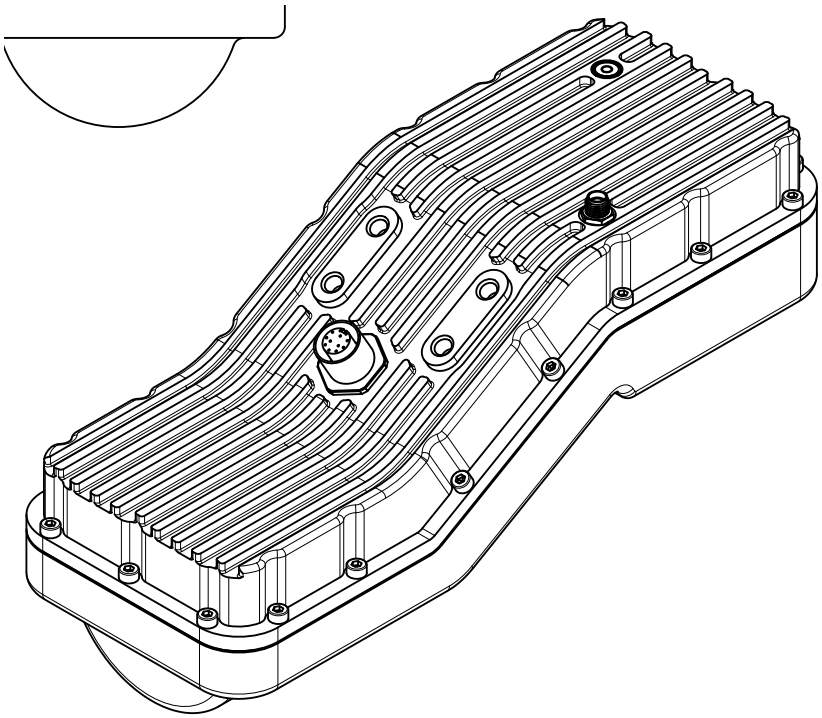
Picture 42. Side view



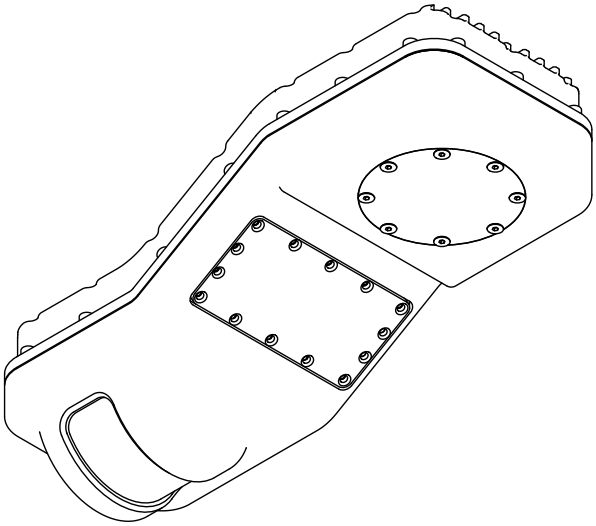
Picture 43. Front view



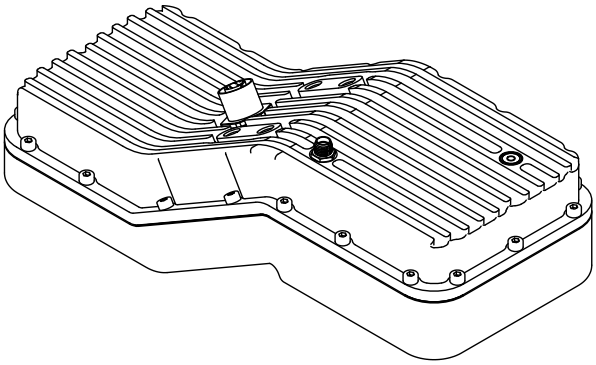
Picture 44. Top view



Picture 45. Isometric view from above



Picture 46. Isometric view from below



Picture 47. Isometric view of the housing