LW20/C microLiDAR® sensor

The world's lightest 100m IP67 rated LiDAR

Disclaimer

Information found in this document is used entirely at the reader's own risk and whilst every effort has been made to ensure its validity, neither LightWare Optoelectronics (Pty) Ltd, its subsidiaries, nor its representatives make any warranties with respect to the accuracy of the information contained herein.

Welcome to LightWare

Thank you for selecting LightWare as your **partner** in distance sensing technology.

LightWare is a pioneer in microLiDAR® distance sensors, drawing upon **four decades** of expertise in LiDAR technology to develop application-specific products renowned for their **accuracy, reliability, and durability**. LightWare's assembly process involves meticulous handling of sensors and optics, creating microLiDAR® sensors of world class quality. Our production methods benchmark the **ISO 9001:2015** standards at scale, with manufacturing capabilities reaching up to **45,000 units annually**, with each microLiDAR® unit crafted to the same exacting standards. Unsurprisingly, leading companies worldwide trust LightWare as their **preferred LiDAR partner**.

We are dedicated to ensuring **your success** when using LightWare microLiDAR® sensors to address your unique distance measuring and geospatial challenges.

Beyond this comprehensive product guide, our website's **resource center** (https://lightwarelidar.com/) offers a wealth of supplementary information, **including APIs, CAD drawings, and FAQs**.

Our dedicated technical support desk is at your service if you require assistance with integration or technical queries. Reach out to them at support@lightwarelidar.com.

LightWare products come with a **24-month limited warranty**, covering any defects in material or workmanship under normal use. For detailed warranty information, please refer to our website at https://lightwarelidar.com/terms-and-conditions/. We're here to support you on your journey — sensing your world with LightWare LiDAR.

Table of contents

1 Overview

This product guide is a comprehensive companion to your **LightWare LW20/C microLiDAR®** - the **lightest IP67 rated LiDAR with a 100m range in the world**.

This small yet robust and easy-to-use sensor is the industry standard for 100-meter range distance and height measurement. Its class-leading performance and functionality is available with plug-and-play ease of integration, and reliable results, while **the IP67 rating** ensures that it is dust and water resistant - making it suitable for direct external mounting without the need for additional protection.

The LW20/C uses the time-of-flight principle to measure distance, emitting a rapid succession of laser pulses that are reflected by target objects and then received back and processed immediately. It uses 905-nanometer laser technology, ensuring optimal performance at an affordable price, while meeting **class 1M eye safety** standards. Its accuracy is not affected by the color or texture of the target surface or the laser beam's angle of incidence, and it is virtually immune to background light, wind, and noise.

The LW20/C's **IP67 sealed enclosure**, together with a **selection of mounting brackets** makes it an ideal add-on component to already sealed environments. Should your application require an open frame component to integrate into your enclosure space, please use LightWare's SF20/C sensor.

Configurable features and dedicated mounting brackets make the LW20/C **easily compatible** with a range of **different controllers** and it **excels in 2D** and **3D applications**. Commonly mounted on a UAV airframe facing downward or forward, the user can select an update rate between 48 and 5000 readings per second, with faster rates giving excellent forward-facing obstacle detection range, and slower rates being preferable for scanning applications. The LW20/C's first and last pulse feature provides depth and distance insights from more than one target in its field of view, helping to penetrate foliage or dust.

By adding a small digital servomotor, the LW20/C's **onboard servo driver** enables customizable **beam steering and scanning**, ideal for autonomous vehicle sense-and-avoid navigation or SLAM mapping. The dedicated alarm channel provides two separate live alarm status outputs, signaling potentially hazardous conditions if an object is detected closer than the user-set alarm distances and angles.

With the LW20/C microLiDAR® now part of your toolkit, you have a solution delivering unparalleled precision, adaptability, and reliability across a diverse range of applications.

2 Safety

Always adhere to these product safety precautions and operate the sensor strictly in accordance with the guidelines outlined in this product guide. LightWare bears no responsibility or liability for any damage or injury, whether direct or indirect, arising from a failure to comply with these stipulations. Non-compliance with the precautions or warnings provided in this product guide constitutes a breach of safety standards intended for the proper use of the sensor.

2.1 Laser eye safety

LightWare LiDAR sensors comply with the United States Food and Drug Administration (FDA) laser eye safety regulations for safe use around humans and animals, based on the international standard IEC 60825-1 and utilizing LaserSafe PC Professional for the computations.

Caution: The sensor contains a laser and should never be aimed at a person or animal. Do not view the laser with magnifying optics such as microscopes or telescopes.

This laser product emits non-ionizing laser radiation. It is classified as Class 1M, indicating that the laser beam is safe to look at with the naked eye during normal use. However, avoid viewing it through magnifying optics such as binoculars, microscopes, telescopes, etc. Despite the safety rating, refrain from looking into the beam, switch off the device when in the vicinity, and never stare directly into the lens from less than half a meter.

Caution: Use of controls, adjustments, or performance of procedures other than those specified herein may result in hazardous radiation exposure.

Warning: Risk of permanent eye damage

- Class 1M lasers are **not safe** if viewed through **magnifying optics such as microscopes, binoculars, or telescopes from a distance less than the NOHD**.
- The laser eye safety rating of the sensor depends on the mechanical integrity of the optics and electronics. It must **not be disassembled or modified in any way**.
- **If the sensor is damaged, do not continue using it**.
- The sensor should be mounted using the mounting holes or product-specific brackets. **Do not attach to or clamp the lens tubes** as this may cause damage and adversely affect the laser safety rating.
- There are **no user-serviceable parts**, and maintenance or repair must only be done by the manufacturer or a qualified service agent.
- No regular maintenance is required, but if the lenses start collecting dust, they may be wiped with suitable lens-cleaning materials. Ensure that the device is switched off before looking into the lenses.

2.2 Labeling

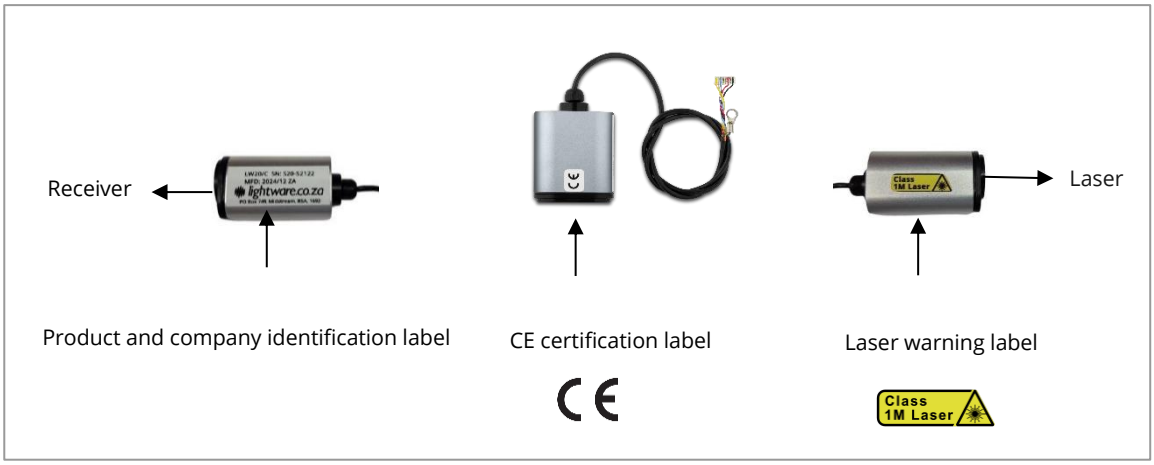

Figure 1: LW20/C laser warning label

2.3 Laser radiation information

Table 1: Laser radiation information

Specification	Value
LightWare product	LW20/C microLiDAR®
LiDAR type	Static single beam
Eye safety classification	Class 1M
Laser wavelength	905 nm
Pulse width	16 ns
Pulse frequency	20 kHz
Average laser power	1.8 mW
Maximum energy per pulse	90 nJ
Extended Nominal ocular hazard distance (Extended NOHD)*	17 m / 55.8ft

* Distance beyond which binoculars may be safely used.

Approximate values only. Please contact LightWare LiDAR if further information is required.

3 Key technical specifications

Table 2: LW20/C microLiDAR® key technical specifications

LW20/C microLiDAR® key technical specifications	
Performance	
Range	0.2 to 100m / 0.6 to 328ft (70% albedo in sunlight conditions, 0.9 x 0.9 m target size)
Update rate	48 to 5000 readings per second (customizable to suit application)
Resolution	1 cm / 0.4 in
Accuracy	± 5 cm / ± 2 in for update rates lower than 500 readings/s ± 10 cm / ± 4 in for update rates higher than 500 readings/s
Connections	
Power supply voltage	4.5 to 5.5 V
Power supply current	85mA typical, <200mA on startup
Outputs and interfaces	Serial UART and I^2C (3.3 V TTL, 5 V tolerant) and servo driver
Form factor	
Dimensions	30 mm x 20 mm x 43 mm / 1.2 in x 0.8 in x 1.7 in
Weight	19 g / 0.69 oz (including cable)
Optical	
Approvals	FDA: 1710193-000 (2020/09) CE certified ROHS3 Compliant REACH unaffected NDAA compliant (Section 848) Blue UAS listed
Laser safety	Class 1M (Please refer to the eye safety section of this user guide, above)
Optical aperture	12.7 mm / 0.5 in
Beam divergence	< 0.5°
Environmental	
Operating temperature	-10 to +50°C / 14 to 122°F
Storage temperature	–40 to 80 °C / –40 to 176 °F
Enclosure rating	IP67
Accessories	
USB adaptor	ACC_USB_Serial
Breakout board	ACC_BOB_08SUR-32S
Pixhawk adapter	ACC_PX_08SUR-32S
ABS Mounting bracket	LW 000_179
ABS servo mounting bracket	LW 000_180
Aluminum stand kit	725-28055
Default settings	
Serial port settings	Baud rate 115200, 8 data bits, 1 stop bit, no parity, no handshaking
I^2C address	0x66 (Hex), 102 (Dec)
Update rate	48 readings per second

4 Accessories

To support configuration and integration, the following accessories are available for purchase from the LightWare website:

4.1 USB adapter

To configure and test your sensor in LightWare Studio via the serial cable a USB adapter is required.

Figure 2: ACC_USB_Serial - Generic USB adapter

4.2 Pixhawk adapter

Each LW20/C is supplied with a 5-way Pixhawk adapter to simplify system integration. Additional adapters are available for purchase from our online store.

Figure 3: ACC_PX_08SUR-32S - Pixhawk adapter

4.3 Breakout board

An optional breakout board accessory is available to facilitate the integration of the LW20/C into a host controller such as a Pixhawk, PX4, Raspberry Pi, or Arduino. It consists of four reusable adapter boards to conveniently connect the LW20/C communication cable to other standard cables and host controllers, without requiring soldering.

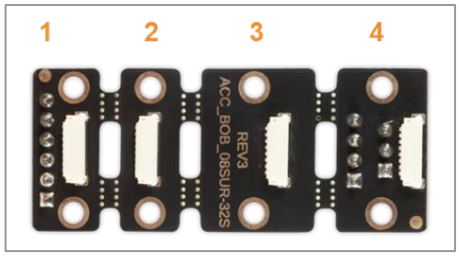

Figure 4: ACC_BOB_08SUR-32S - Breakout board for LW20/C

Table 3: ACC_BOB_08SUR-32S - Breakout board sub-boards

Board 1	Six-pin header connector	Six pins corresponding to the LW20/C communication cable pins. Pin 1 is labelled on the board.
Board 2	Six-way serial interface connector	Ideal for use on the telemetry *telem* port of the Pixhawk.
Board 3	Four-way I^2C interface connector	Use an I^2C cable to connect the board to the I^2C port of the Pixhawk. Twin connections allow a daisy chain connection to other sensors.
Board 4	Servo motor interface connector	The host controller can power, communicate with, and control both the servo motor and the LW20/C through this board, using serial communication.

5 Getting started

LightWare Studio is a free application (available for Windows, macOS, and Linux) and is the gateway to configuring your microLiDAR® sensor and visualizing your data. This software empowers you to customize settings, fine-tune sensor parameters, and easily analyze data. It also facilitates firmware upgrades and in-field diagnostics and support.

Detailed step-by-step videos are available on LightWare's YouTube channel: https://www.youtube.com/@LightWareLiDAR/videos

Follow these easy steps to get going with your LightWare microLiDAR®:

1. Download and install the version of LightWare Studio compatible with your operating system from the Resource section of LightWare's website at https://lightwarelidar.com/. You can safely install over an existing version of LightWare Studio if you are upgrading.

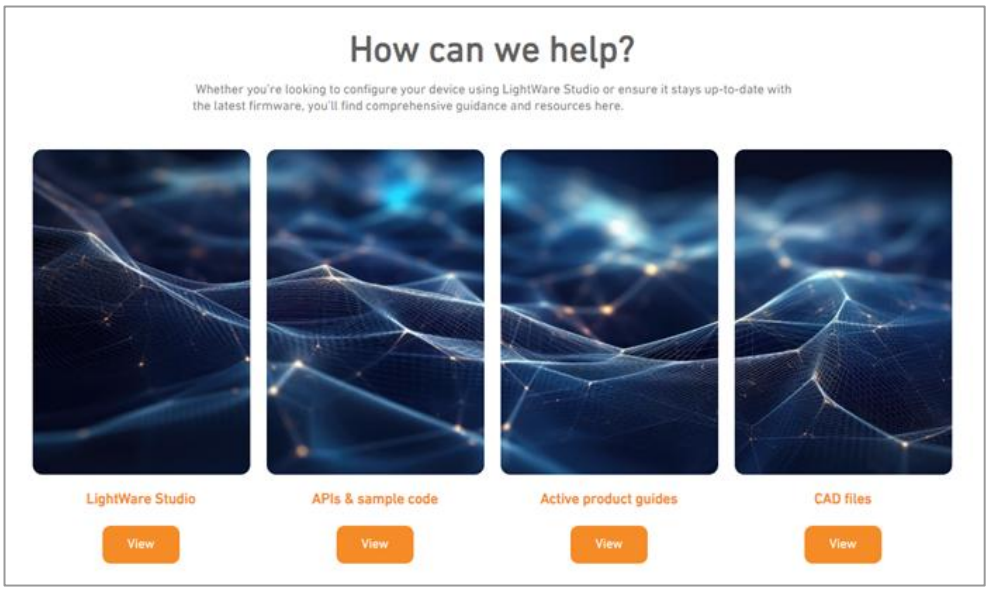

Figure 5: LightWare Studio website download page

2. Once the installation is complete, the *Welcome to LightWare Studio* page will open, prompting you to attach a device to your computer.

3. Carefully connect your LW20/C to the USB adapter using the communication cable. Check the connectors are correctly orientated and very gently squeeze them in until they click into place. To unplug the cable, use a small flat-head screwdriver in the groove on the connector.

Figure 6: LW20/C connection to a USB adapter

Caution: To avoid the risk of shorting the high voltage lines on the sensor circuit board, connect the USB cable to the adapter first before connecting it to the computer.

4. Insert the USB adapter or USB cable into your computer.

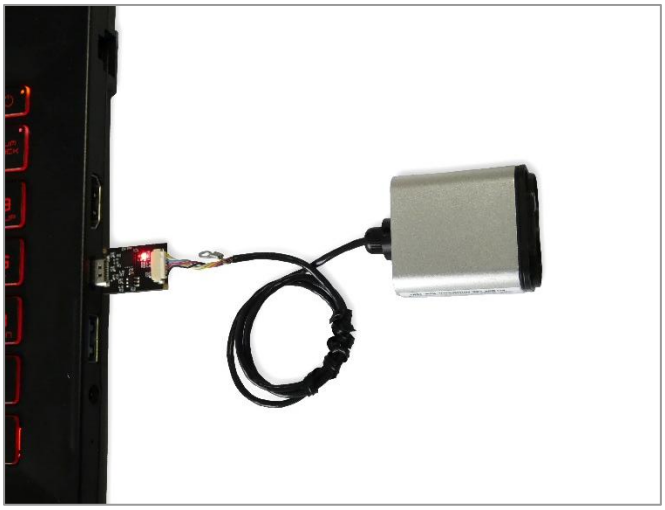

Figure 7: Red power LED on the USB adapter is illuminated

5. When connecting the sensor for the first time, Windows users may experience a brief delay as the operating system installs the necessary generic communication driver. Please allow the installation process to complete.

6. LightWare Studio will automatically detect the USB adapter and present it for selection on the Welcome page. The Welcome page may show other communications ports on your computer. Select the port that shows the connection to your LightWare USB adapter.

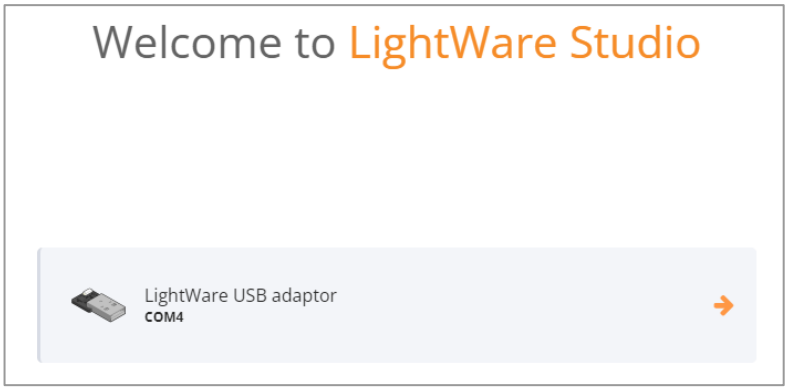

Figure 8: Connection established with the USB adaptor

7. LightWare Studio will start on the device's Info page, indicating the serial number, hardware version and firmware version of your device.

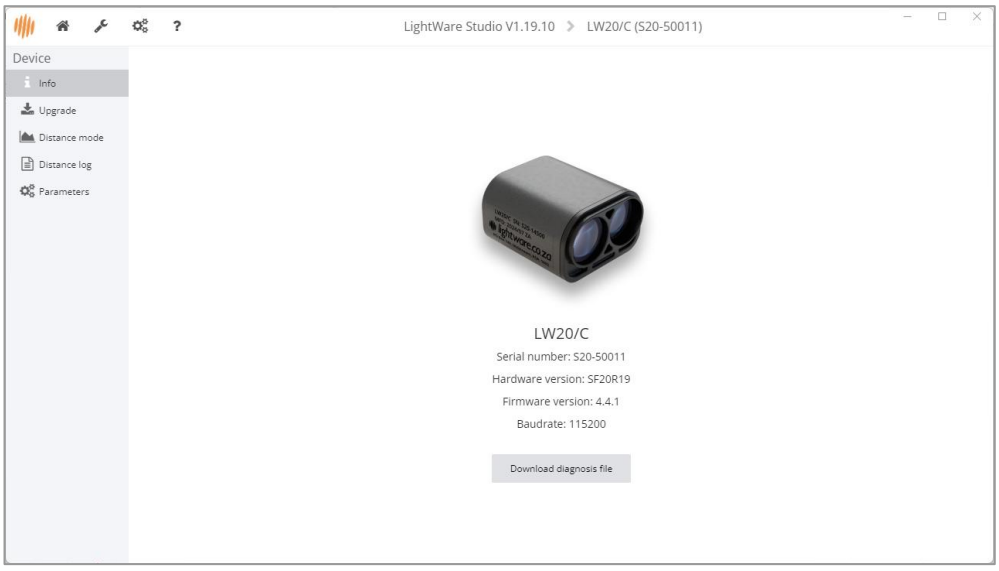

Figure 9: LightWare Studio device information page

8. Navigate to the *Distance mode* tool from the left panel. This streams live distance data in meters as it is scanned by the sensor as a graph.

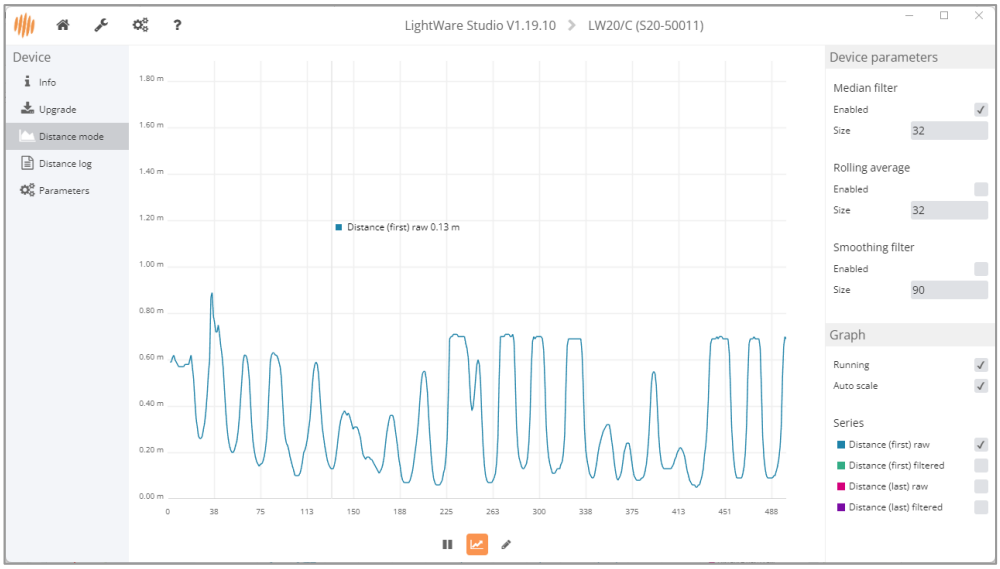

Figure 10: LightWare Studio LW20/C distance graph page

9. Data can be downloaded and saved using the pencil icon below the data.

10. On this display, you can control specific device parameters, which are also accessible from the dedicated parameters page. (Please refer to the section below for a more comprehensive understanding of these parameters.)

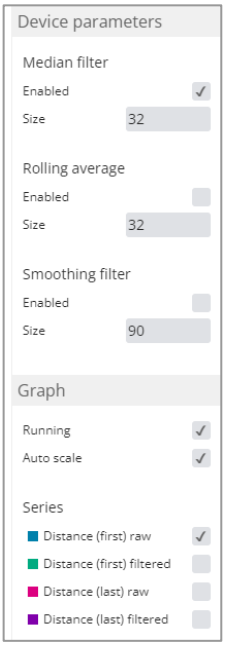

Figure 11: LightWare Studio LW20/C parameters in Distance mode page

11. Navigate to the *Distance log* tool from the left panel. This tool streams live distance data in meters as it is scanned by the sensor. Toggle the parameters on the right to stop or start the streaming, add line numbers or time stamps, or switch on different data types.

12. Data can be downloaded and saved using the *save* icon above the data.

Figure 12: LightWare Studio LW20/C distance log page showing measurements

6 Parameters, filters, settings, and tools

6.1 Setting the device parameters

Your LightWare LW20/C microLiDAR® sensor can be configured via LightWare Studio or from a host controller using the product commands through the serial UART or I²C communication interfaces.

To set the device parameters using LightWare Studio:

1. In the left panel, click on *Parameters* to open the detailed parameters page.

2. The scroll-down list of adjustable parameters will be displayed, with explanatory notes and dropdown options.

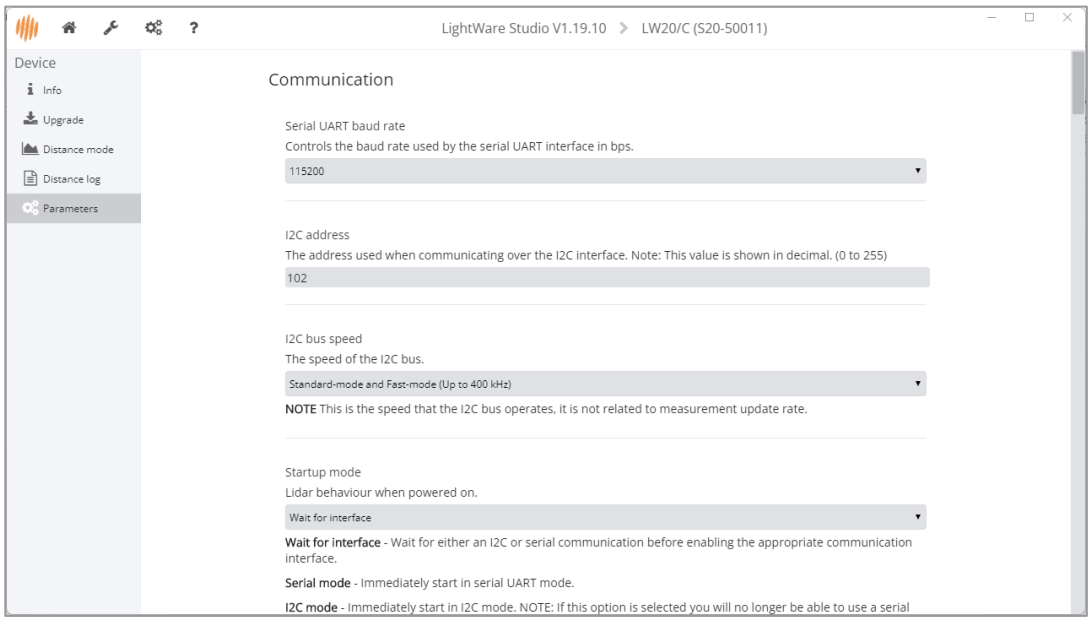

Figure 13: LightWare Studio LW20/C detailed parameters page

3. Set your device parameters according to your requirements. Refer to the table below for more information.

Table 4: LW20/C adjustable parameters

Parameter	Explanation	Options/range
Communication		
Serial UART baud rate	Select the serial UART interface baud rate, (in bps).	9 600 to 921600
I2C address	The address used when communicating over the I²C interface. A whole number in decimal.	0 to 127
I2C bus speed	Select I²C bus operating speed. **Note** that it is not related to the reading update rate. **Standard mode and fast mode**: Up to 400 kHz **Fast mode plus**: Up to 1 MHz **High-speed mode**: Up to 3.4 MHz	Standard mode and fast mode; Fast mode plus; or High-speed mode
Startup mode	Select the communication behavior when powered on **Wait for interface**: Wait for either an I²C or serial [UART] communication before enabling the appropriate interface. **Serial mode**: Immediately start in serial UART mode. **I2C mode**: Immediately start in I²C mode. Note: If this option is selected, you will no longer be able to use a serial UART adapter to communicate with the sensor. **Serial mode with legacy banner**: Immediately start in serial UART mode and output legacy diagnostics text on startup. **Serial mode with no preamble**: Immediately start in serial UART mode, but do not output the initial MMI message on startup.	Wait for interface; Serial mode; I²C mode; Serial mode with legacy banner; or Serial mode with no preamble
Measurement:		
Update rate	Select the number of measurements taken per second (Hz).	48 to 5 000
Update rate override	Specify a custom update rate which will take priority over the *Update rate* parameter. The value will be adjusted to the nearest rate allowed by the sensor. (Whole number, Hz.)	0 to 5000
Zero distance offset	The offset applied to the measured distance value. (In meters, up to three decimal places.)	-10 to 10
Lost signal threshold	The number of failed readings required before a loss of signal is reported, (whole number).	1 to 250
Lost signal type	Select the output value to be reported when signal is lost.	-1.00 or 230.0
Filtering:		
Median filter	Used to disregard short unwanted readings.	Select or de-select the checkbox
Median filter size	The response time of the median filter. (Whole number, in seconds.)	3 to 32
Rolling average	Used to average out a specified number of last-distance results.	Select or de-select the checkbox
Rolling average size	The number of distance results to use for the rolling average filter, (whole number).	2 to 32
Smoothing filter enabled	Used to remove noise from the readings.	Select or de-select the checkbox
Smoothing filter strength	The stronger the smoothing, the slower the response to change, (whole number).	0 to 100
	Refer to the next section of this product guide for more information on filters.	

Parameter	Explanation	Options/range
Scanning		
Servo connected	Indicate that a servomotor is connected to the sensor.	Select or de-select the checkbox
Servo minimum PWM value	This value sets to PMW value of the left most position that the servo must move, viewed from above. (In µs, up to two decimal places. The default is 1000 µs.	0 to 2 999
Servo maximum PWM value	The maximum allowed PWM value that will turn the servomotor's shaft to its extreme right position. (In µs, up to two decimal places. The default is 2000 µs.)	0 to 2 999
Servo angular scale	The number of µs of the PWM that will cause a 1 degree of movement of the servo. (In µs per degree, up to two decimal places. The default is 10 µs per degree.)	0.1 to 1000
Steps per reading	The number of degrees moved by the servomotor after each reading. Note that the servo scan speed is determined by this parameter together with the sensor update rate (readings per second). (Whole number, in degrees.)	1 to 32
Servo lag	Compensates for the characteristic data angle mismatch when the servomotor changes direction. (In degrees, up to two decimal places.)	0 to 90
Field of view (lowest angle)	Instructs the sensor to only output readings taken when the servo angle is higher than this value. This cuts off unwanted data to create the left edge of the field of view. (In degrees, up to two decimal places.)	-180 to 180
Field of view (highest angle)	Instructs the sensor to only output readings taken when the servo angle is lower than this value. This cuts off unwanted data to create the right edge of the field of view. (In degrees, up to two decimal places.)	-180 to 180
Scan type	For higher precision, scanning in only a single direction can be selected.	Bi-directional or Uni-directional
Servo scan on startup	Start scanning automatically when the sensor is powered up, without requiring the servo start command to be sent.	Select or de-select the checkbox
Closest sweep distance	The sensor will output only the closest distance detected during each sweep.	Select or de-select the checkbox
	Refer to the LightWare website FAQ for more info on scanning parameters: https://lightwarelidar.com/resources-faqs/	
Alarms		
Alarm A distance	Warn when an object is detected closer than this user-set alarm distance. (In meters, up to two decimal places.) When scanning is activated, the object must also be between the left and right alarm angles.	0 to 100
Alarm B distance	Warn when an object is detected closer than this user-set alarm distance. (In meters, up to two decimal places.) When scanning is activated, the object must also be between the left and right alarm angles.	0 to 100 meters
Alarm hysteresis	The amount by which distance reading must decrease below the alarm distance before the alarm is cleared. Used to prevent alarm chatter. (In meters, up to two decimal places.)	0 to 10 meters
Scanning alarms		
Scanning Alarm A (low angle)	The lower (left) angle used with Alarm A during scanning. (In degrees, up to two decimal places.)	-180 to 180

Parameter	Explanation	Options/range
Scanning Alarm A (high angle)	The higher (right) angle used with Alarm A during scanning. (In degrees, up to two decimal places.)	-180 to 180
Scanning Alarm B (low angle)	The lower (left) angle used with Alarm B during scanning. (In degrees, up to two decimal places.)	-180 to 180
Scanning Alarm B (high angle)	The higher (right) angle used with Alarm B during scanning. (In degrees, up to two decimal places.)	-180 to 180
GPIO pins		
GPIO mode	Select the function of the server control / general-purpose input/output (GPIO) output.	Servo, Alarm A, or Alarm B
GPIO alarm confirmation count	The number of readings required before the alarm state on the GPIO pin is changed, (whole number).	1 to 10 000
Extra		
Legacy streaming frequency	Select legacy distance data streaming rate, in ASCII format over the serial UART interface, in Hz.	'Off' or 1 to 100
MMI/HMI measurement output resolution	Select the unit of measurement output in Machine-Machine Interface (MMI) or Human-Machine Interface (HMI) modes.	Centimeter or Millimeter
LED enabled	Turn the sensor LED on or off.	Select or de-select the checkbox
Custom selection	Select to enable specific device operation modes.	None or No legacy space prefix

6.2 Filters

Median Filter

This is a non-linear filter in which each output is the median, or middle, of the readings in the filter window. It is helpful for removing outliers and signal noise. A typical application is flying over terrain where quick changes in terrain height must not affect the altitude of the UAV, such as over water. The larger the filter size, the more immune the filter is to noise. This, however, does result in a delayed response to changes in the measurement.

Rolling average filter

A rolling average filter averages a fixed number of the newest data points. This smooths out short-term fluctuations and reveals trends. Typically applied when noisy fluctuations should be ignored, but the general profile of the data should be maintained, such as for terrain following.

Smoothing filter

This filter smooths sharp changes in the data while preserving slower changes. It helps to remove noise from the data and slows the response to sudden data changes. Typically used when monitoring the distance to stationary surfaces, such as when taking level measurements and filling rates.

6.3 Settings and tools

Additional application **settings** are available by clicking on the *gears* icon in the top menu:

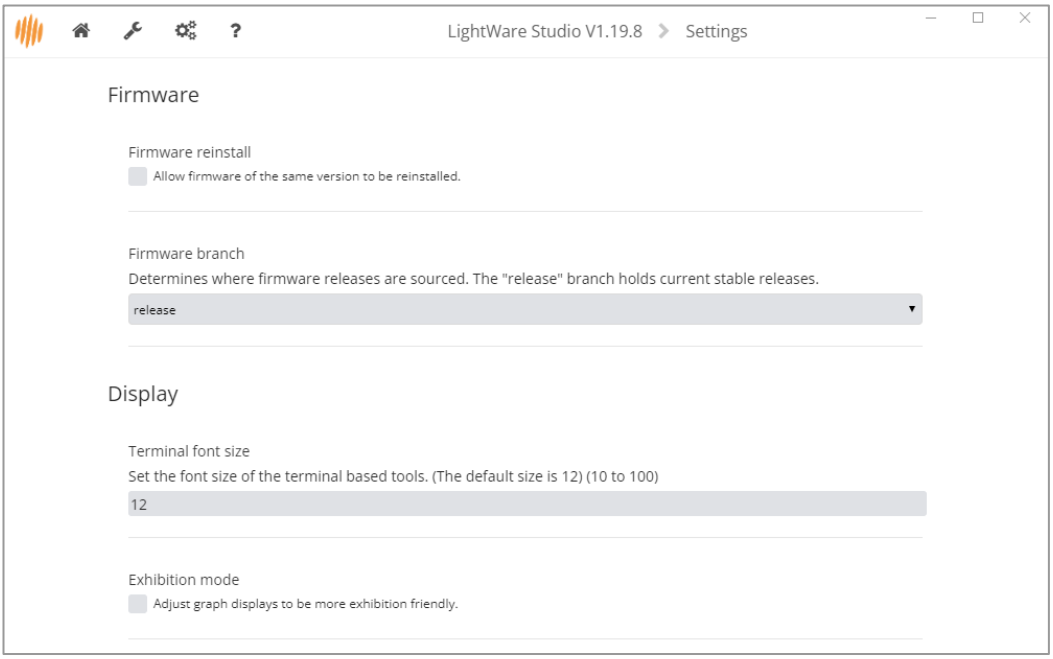

Figure 14: LightWare Studio application settings page

You can access the **specialized device tools page** by clicking on the *wrench* icon in the top menu, including a traditional terminal if needed:

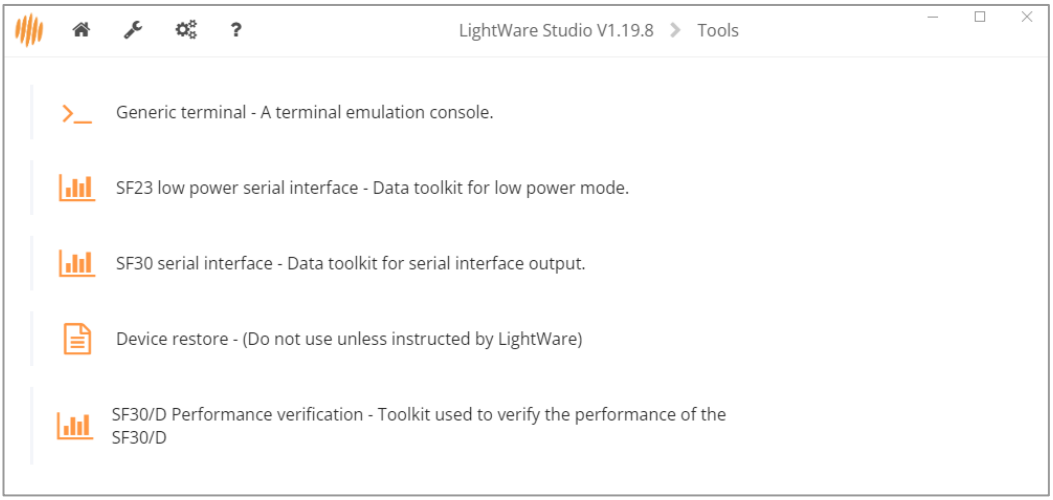

Figure 15: LightWare Studio specialized tools page

7 Installation, mounting and cabling

7.1 Mechanical interface

For detailed CAD files, please refer to the LightWare resource center at https://lightwarelidar.com/.

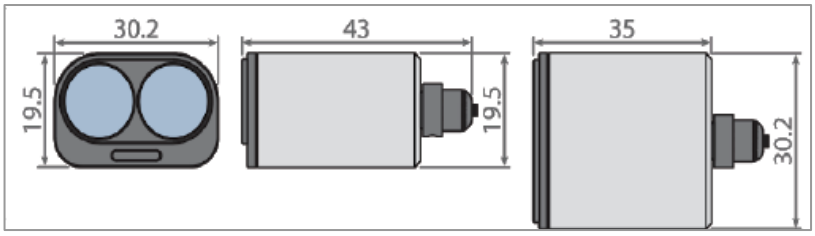

Figure 16: LW20/C dimensions

7.2 Mounting bracket

Mounting accessories can be purchased from the LightWare website. Alternatively the CAD file can be downloaded from the LightWare resource center at https://lightwarelidar.com/ to print your own bracket.

The ABS mounting brackets available for the LW20/C provide a convenient way to securely grasp and mount the IP67 enclosure of the sensor.

Figure 17: Mounting brackets for the LW20/C

There are two mounting bracket options available. Should through-hole mounting holes be required, the bracket with SKU: LW 000_179 (shown on the left below) would be suitable. In cases where the LW20/C needs to be mounted on a servo, the bracket with SKU: LW 000_180 (shown on the right below) would be preferable.

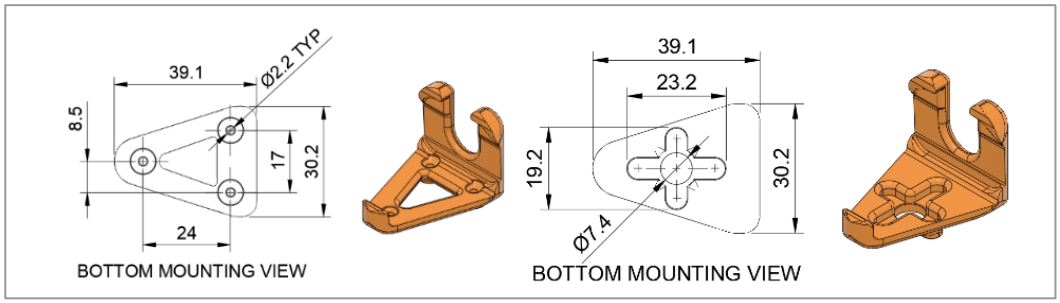

Figure 18: LW20/C mounting bracket options and dimensions

In cases where an offset stand is required, an aluminum stand can be purchased from the LightWare website. The kit (725-28055) is complete with a mounting bracket and mounting hardware to securely fasten the bracket to both the LW20/C and the desired application point.

Figure 19: Aluminum mounting bracket for LW20/C

7.3 Mounting and alignment instructions

Take careful note of the following points when mounting the sensor:

- When choosing a position, ensure that there is **nothing in the path** of the laser beam and that there are **no shiny or highly reflective surfaces near the beam path** that could result in false signals.
- Do not mount the sensor recessed within a cavity of the airframe. This can cause false readings in short-range distances (side lobes) or out-of-range conditions. Mount the sensor **flush with the exterior** or keep the recess conical and shallow.
- The aluminum enclosure of the LW20/C provides environmental protection while also acting as a heat sink. In cases where the sensor is incorporated into a custom enclosure, please make provision for adequate dissipation of heat from the LW20/C. Thermally conductive tape/pads may be applied to the bottom of the chassis of the LW20/C.
- The LightWare microLiDAR® sensor is designed for installation with exposed lenses. If it is to be mounted behind glass, ensure use of non-reflective glass and mount the sensor flush with the glass to prevent false readings. The glass must have good transmission at 905 nm wavelength, with an anti-reflective coating optimized for this wavelength.
- Secure the communication cable to prevent it from pulling on the connector.
- Make sure the sensor is securely mounted to prevent false readings or damage.

7.4 Orientation

The sensor requires a clear line-of-sight to measure distance to a target surface. It can be mounted with a vertical or horizontal lens orientation.

Figure 20: Sensor mounting orientations

It can be mounted in a downward-facing, angled, or forward-facing orientation, depending on your application:

- Mount with a downward-facing orientation for altimetry, terrain following, or precision landing applications.
- Mount at an angle to reduce reaction lag time for terrain following. The ideal angle depends on the speed traveled and the overall system lag but should be between 20° and 45°.
- Mount in a forward-facing orientation for sense-and-avoid or position-hold applications.

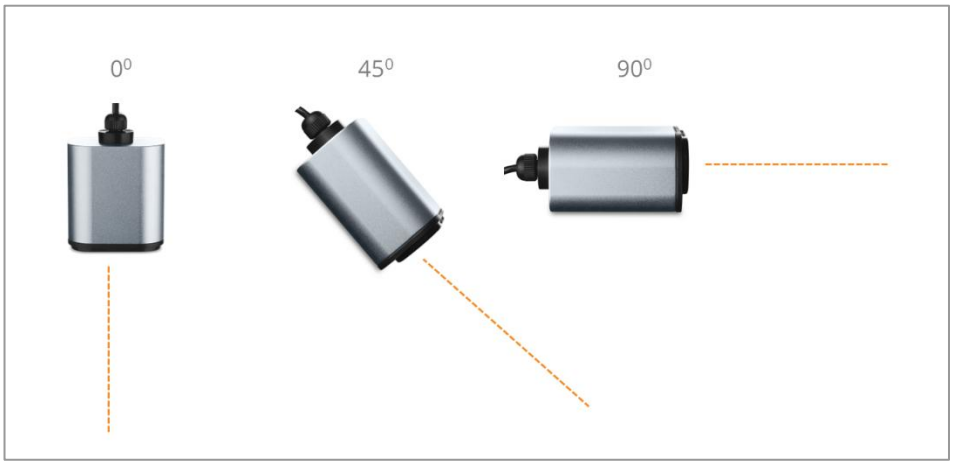

Figure 21: Sensor mounting angle

7.5 Communication and power cable

The LW20/C's 40-cm five-conductor cable is permanently connected via an IP67-rated gland. This cable carries the power supply, communications signals, and servo driver signals. The cable is shielded and must be earthed to reduce electromagnetic interference (EMI).

Figure 22: LW20/C communication cable and grounding lug

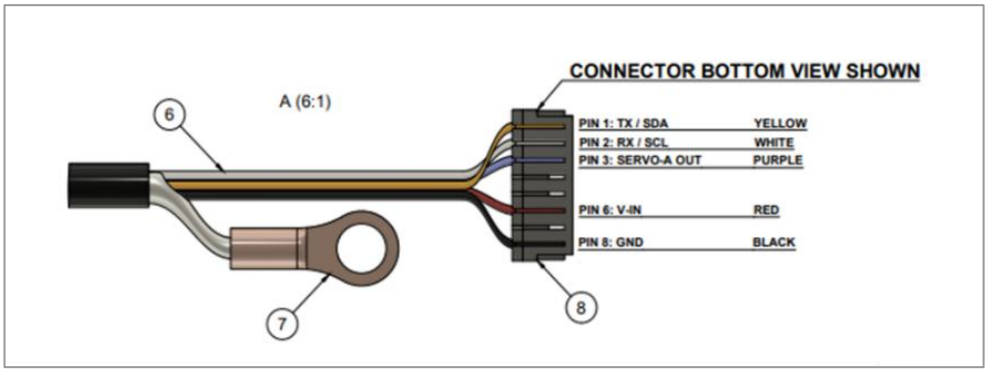

Figure 23: LW20/C communication pinout

Figure 24:LW20/C pinout table

Connector Pin	Wire	Serial Function	I²C Function
1	Yellow	TXD, transmit data for serial connections	SDA, serial data for I²C connections
2	White	RXD, receive data for serial connections	SCL, serial clock for I²C
3	Blue	Servo control line, PWM, GPIO	
4		[pin not used]	
5		[pin not used]	
6	Red	VIN, +5 V power supply positive (4.5 V to 5.5 V at 85 mA typical, 200 mA max)	
7		[pin not used]	
8	Black	GND, power supply negative, power or logic	
	Shield	Earth to reduce EMI	

Note: The serial UART, I²C, and servo interfaces use 3.3 V TTL logic, (5 V tolerant). The 5-volt power supply should be sized appropriately for startup power.

8 Advanced features

8.1 First and last pulse detection

This LightWare microLiDAR® sensor features *first and last pulse* processing, capturing both initial and final laser return signals in scenarios where multiple objects are within the sensor's line of sight. It is important to note that objects must be separated by approximately five meters or more for separate return signals to be recognized.

First and last pulse capability allows the microLiDAR® sensor to measure its altitude above the ground while simultaneously monitoring its height above treetops or structures for collision avoidance, and enhances performance in challenging environmental conditions like dust, rain, fog, and snow. By discerning both pulses, the sensor can effectively penetrate these elements and accurately report the furthest distance as the actual target. This feature also allows the sensor to measure the distance to objects through foliage.

A glass window in the sensor's line of sight will reflect some laser energy back toward the receiver, potentially resulting in false readings. The sensor's *first and last pulse* detection feature can usually mitigate this issue, depending on the type of glass used.

Although *first and last pulse* detection is helpful when the sensor needs to be positioned behind a protective window, this type of mounting is not recommended, as LightWare sensors are designed to be integrated with exposed lens elements.

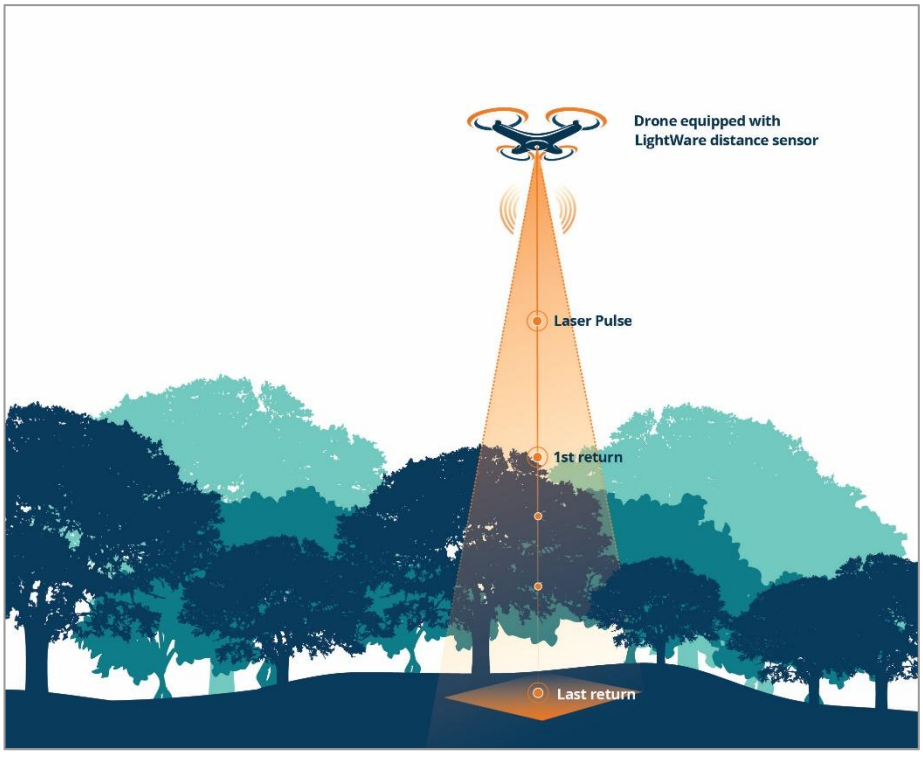

Figure 25: First and Last pulse returns

8.2 Servo Drivers

The LW20/C microLiDAR® sensor is equipped with an onboard servo driver and can easily be paired with a digital servo motor using the onboard servo driver hardware and software to create a 2D or 3D scanning LiDAR. The sensor can then be aimed in selected directions or used for autonomous scanning. With live-streamed data, this creates an effective SLAM (simultaneous localization and mapping) device and collision sensor by activating the two internal collision avoidance warning alarms in real time. Standard digital servo motors are recommended, as the control signal response for analog servo motors is too slow.

Connect the microLiDAR® sensor and servomotor as shown in the diagram below. To reduce the risk of power supply spikes influencing the sensor's performance, it is crucial to run the servo from a separate power supply. Check the servo specifications for the correct voltage and current ratings and ensure a common connection to the negative rails of the sensor and servo power supplies. Most standard digital servomotors are compatible, but analog servomotors are not compatible due to their slow control signal response.

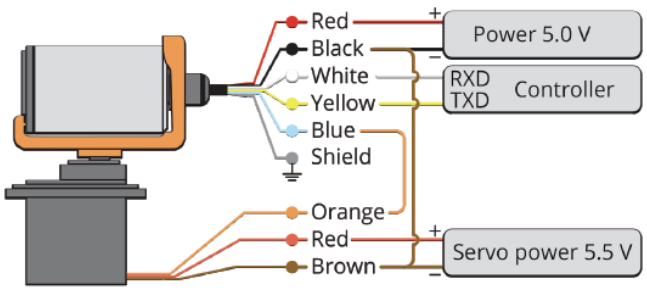

Figure 26: Servo wiring schematic diagram for SF20/C and LW20/C

Refer to the LightWare website FAQs https://lightwarelidar.com/resources-faqs/ for more information on setting parameters for the servomotor sweep limits and speed, servo lag, field of view, and scan type.

8.3 Alarms

Your LightWare LW20/C microLiDAR® measures and reports distances and has a dedicated alarm channel providing two separate live alarm status outputs, warning of potentially hazardous conditions.

Alarm A and Alarm B give separate warnings when the ground (or another object) is detected closer than their user-set alarm distances. Each time a distance measurement is taken, the data is analyzed internally by the sensor and the alarm statuses are updated in real time.

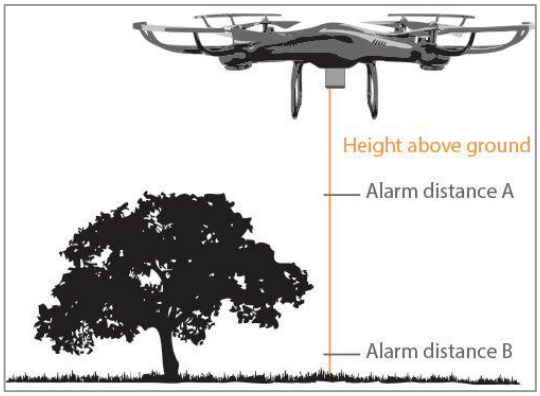

Figure 27: Alarm A and B without scanning

When used in scanning mode (with a servomotor) the alarms check both distance and angle settings and warn when the ground or another object is detected closer than the user-set alarm distance within the alarm angle. These two-dimensional alarms update at the end of each sweep and remain fixed for the duration of the next sweep.

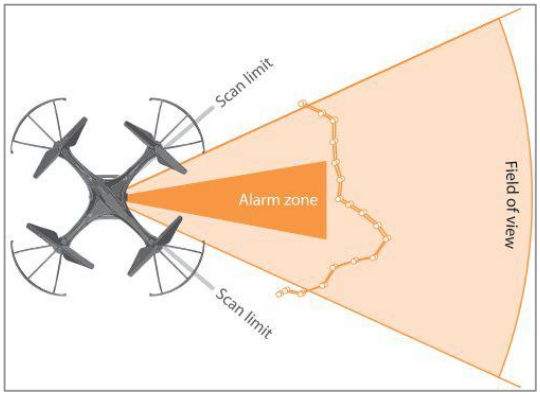

Figure 28: Alarm A and B with scanning

These alarm parameters (including hysteresis) can be set using the LightWare Studio parameters page or by using commands through your host controller.

9 Communication interfaces

The LightWare microLiDAR® sensor can be connected to a host controller, transmitting results and receiving commands with a serial UART or an I^2C communication interface.

- The one-to-one serial UART interface allows one sensor to communicate with a single host controller.
- The configurable address of the I^2C communication interface allows multiple sensors to be connected to one host controller on a common bus.

Once a sensor is connected to a host controller, the first command from the host controller will inform it which of the two communication interfaces is being used. Subsequent commands sent from the host controller to the sensor will request values, change settings, or alter the sensor's performance. The sensor will reply to a single command with a single reply, although the streaming command allows the sensor to continuously update the reply without the host resending the command. Note that streaming data is only available through the serial UART interface. The complete command list is contained in this product guide.

We suggest using LightWare's pre-built APIs wherever possible, which are available via the LightWare website resource center. If you require more control or do not find a suitable pre-built API, you can use the information below to build a compatible system. The packet-based binary protocol is compatible with higher-level APIs like C, Python, and JavaScript. Please contact LightWare for assistance with APIs or programming if required.

9.1 Serial UART interface

For serial UART communication, the sensor uses encapsulated packets to send and receive data. A packet sent **to** the sensor is a request. A correctly formatted request will always be **replied** to with a response. Streaming is available through the serial UART interface. In this case, the sensor sends request streaming packets without a direct request from the host, and they do not require a response from the host.

Requests are made using one of the sensor commands. The complete command list is contained in this product guide. Commands are flagged as either read or write. When a read request is issued, the response will contain the requested data. When a write request is issued, the contents of the response will vary depending on the command.

Default serial UART interface properties:

- Baud rate: 115200 (configurable)
- Data: 8 bit
- Parity: none
- Stop: 1 bit
- Flow control: none

9.2 I²C interface

For I²C communication, the sensor will always be the slave on the I²C interface and only transmit data when requested by the master.

Multiple sensors can be connected to an I²C bus. The I²C serial bus configurable address allows connecting multiple devices on a common bus. Default I²C interface Address: 0x66 or 102. The sensor's I²C interface SDA and SCL pins use 3.3 V logic levels with a 3.3kΩ pull-up resistors, but are also 5 V tolerant.

Requests are made using one of the sensor commands. The complete command list is contained below in this product guide. When a read request is issued the response will contain the requested data. When a write request is issued there is no response generated.

10 Commands

Your LightWare microLiDAR® use three communication protocols for both serial UART and I²C communication:

- The packet-based binary protocol is a register based protocol that is compatible with higher-level APIs like C, Python, and JavaScript. This is LightWare's recommended protocol as it allows for various data streaming from a single request.
- An **ASCII protocol**, where command strings are simple mnemonics and replies are text strings, both human-readable. These ASCII strings' maximum length is 32 characters.
- A **legacy protocol**, with some special commands compatible with older systems, in binary format.

The first command sent by the host to the sensor after powerup will be used to detect whether serial UART or I²C mode is in use. The sensor will not return a response to the first command. Subsequently, for each command sent by the host controller, a single reply will be returned by the sensor.

To initialize the communication with the sensor, send the command to request the Product name. It is advisable to send the command to query the Product name twice in succession shortly after powerup. As described above the first request will not return a response, however the second request will return the product name, indicating that the sensor has indeed initialized successfully and a handshake has been successfully established with the sensor.

The streaming ($) command can be used to command the sensor to continuously update the reply without waiting for the host controller to resend the command.

10.1 Binary protocol

10.1.1 Binary protocol - Command structure

Both request and response **packets** are composed of the following bytes:

Table 5: Packet composition

	Header			Payload		Checksum	
Byte	start	flags low	flags high	ID	data	CRC low	CRC high

Table 6: Header Flag byte explanation

Byte	**Flags high**								**Flags low**							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Payload length (0 to 1023)								Reserved							Write

- The **start** byte is always 0xAA and indicates the beginning of a packet.
- The **flags** bytes form a 16-bit integer representing the packet's payload length and read/write status.
- The **payload** includes the ID byte, the data bytes, and the write bit. Its length is between 1 and 1023 bytes, inclusive depending on the command type.
- The **ID** byte indicates which command the request/response relates to.
- The **command list** is contained later in this product guide.
- The **write** bit is 1 to indicate write mode, or 0 to indicate read mode.
- The **CRC** bytes form a 16-bit/2-byte checksum value used to validate the integrity of the packet data. The sensor will not accept and process a packet if the CRC is not correctly formed. Every byte in the packet except for the CRC itself is included in the checksum calculation.

10.1.2 Binary protocol - Checksum algorithm

The **checksum** algorithm is CRC-16-CCITT 0x1021. Below are two CRC calculation examples:

Table 7: Checksum algorithm

C/C++	JavaScript

uint16_t createCRC(uint8_t* Data, uint16_t Size)
{
  uint16_t crc = 0;
 	
  for (uint32_t i = 0; i < Size; ++i)
  {
  uint16_t code = crc >> 8;
  code ^= Data[i];
  code ^= code >> 4;
  crc = crc << 8;
  crc ^= code;
  code = code << 5;
  crc ^= code;
  code = code << 7;
  crc ^= code;
  }
 	
  return crc;
}
	function createCRC(data, size) {
	  let crc = 0;
	
	  for (let i = 0; i < size; ++i) {
	  let code = crc >>> 8 & 0xFF;
	  code ^= data[i] & 0xFF;
	  code ^= code >>> 4;
	  crc = crc << 8 & 0xFFFF;
	  crc ^= code;
	  code = code << 5 & 0xFFFF;
	  crc ^= code;
	  code = code << 7 & 0xFFFF;
	  crc ^= code;
	  }
	
	  return crc;
	}

10.1.3 Binary protocol – Reading bytes

Once a packet is successfully read it can be processed based on its command ID. It is vital to **verify the payload length and checksum** before processing.

If either of the following errors are received, “invalid packet length” or “checksum is invalid”, please roll the incoming stream back to one byte after where the start byte was detected.

Below is the process for reading the raw serial byte stream and identifying packets:

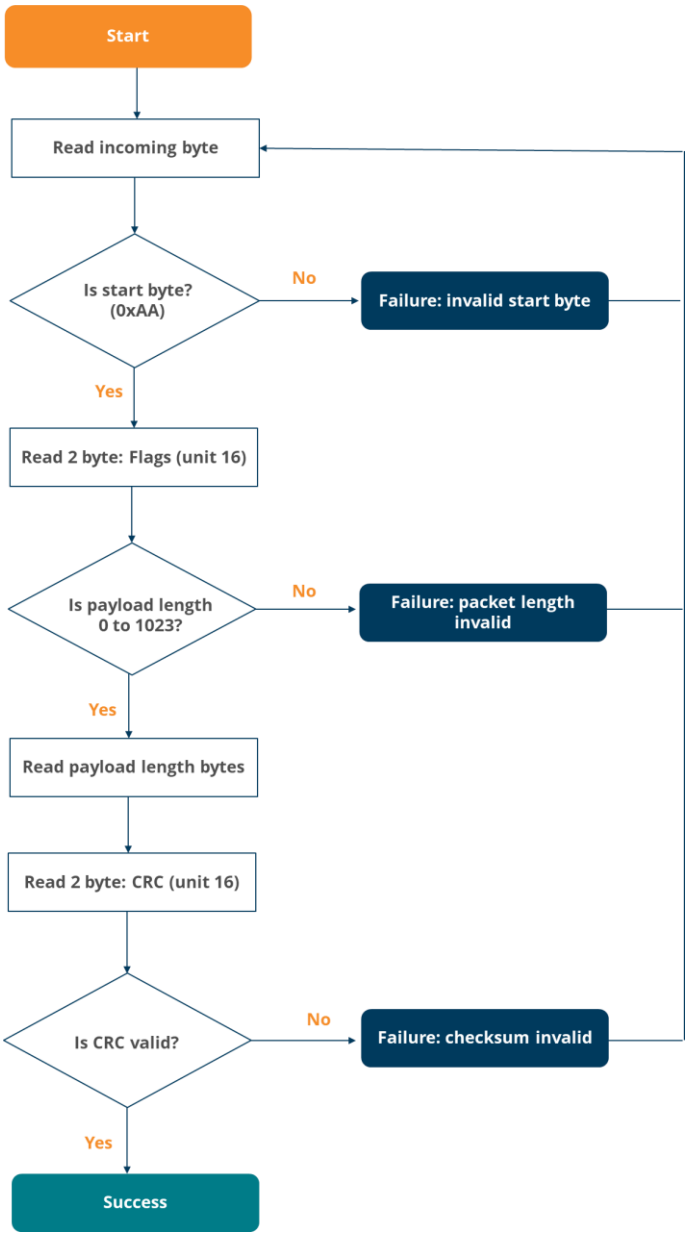

Figure 29: Process flow for reading bytes

10.1.4 Binary protocol - Sending commands

Every request sent to the sensor will receive a response. The response also confirms that the request was received and processed. The timeout value and number of retries should be optimized for the specific application.

Below is the process for sending a command request and reading the response:

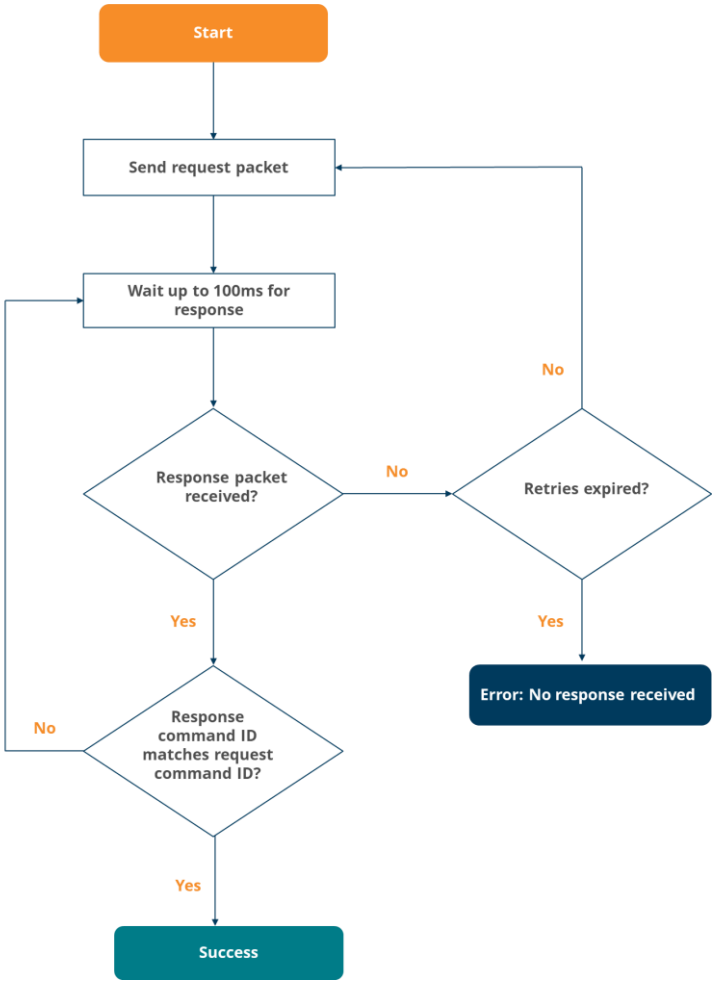

Figure 30: Process flow for sending commands

10.1.5 Binary protocol - Saving

Parameters listed in the command list below, and indicated to persists across power cycles, must be saved to onboard flash once changed.

To save the parameters, the Token (ID 10) must be read from the unit by sending a read command. The value received must then be sent as the data in the Save Parameters command (ID 12) to the unit.

The Token expires every time after use and consecutive save commands will require the request of a new token prior to the save commands sent.

10.1.6 Binary protocol – Command list

Table 8: Binary protocol command list

<table>
<tr><th>ID</th><th>Name</th><th>RW</th><th>Read bytes</th><th>Write bytes</th><th>Persists</th><th>Description</th></tr>
<tr><td>0</td><td>Product name</td><td>R</td><td>16</td><td>-</td><td>-</td><td>A 16-byte string indicating product model name.
Always LW20 followed by a null terminator.
Use to verify the LW20 is connected and operational over the selected interface.</td></tr>
<tr><td>1</td><td>Hardware version</td><td>R</td><td>4/ uint32</td><td>-</td><td>-</td><td>The hardware revision number as a uint32.</td></tr>
<tr><td>2</td><td>Firmware version</td><td>R</td><td>4</td><td>-</td><td>-</td><td>The currently installed firmware version as 4 bytes.
Used to identify the product for API compatibility.
<table>
<tr><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>Patch</td><td>Minor</td><td>Major</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>3</td><td>Serial number</td><td>R</td><td>16</td><td>-</td><td>-</td><td>A 16-byte string (null-terminated) of the serial identifier assigned during production.</td></tr>
<tr><td>9</td><td>User data</td><td>RW</td><td>16</td><td>16</td><td>Yes</td><td>16 bytes of user data stored and read for any purpose.</td></tr>
<tr><td>10</td><td>Token</td><td>R</td><td>2 / unit16</td><td>-</td><td>-</td><td>Next usable safety token / Current safety token.
Once used, it will expire, and a new token will be created.</td></tr>
<tr><td>12</td><td>Save parameters</td><td>W</td><td>-</td><td>2/ uint16</td><td>-</td><td>Commands written to, that must be stored and persist across power cycles will be saved to flash memory on the receipt of the latest Token (ID 10) value sent to this command.
The safety token prevents unintentional writes.
The token expires once a successful save has completed.</td></tr>
<tr><td>14</td><td>Reset</td><td>W</td><td>-</td><td>2/ uint16</td><td>-</td><td>Writing the safety token to this command will restart the sensor.</td></tr>
<tr><td>27</td><td>Distance output</td><td>RW</td><td>4/ uint32</td><td>4/ uint32</td><td>No</td><td>Configures the (44) distance data command data output. Each bit toggles the output of specified data.
<table>
<tr><th>Bit</th><th>Output</th></tr>
<tr><td>0</td><td>First return raw</td></tr>
<tr><td>1</td><td>First return closest</td></tr>
<tr><td>2</td><td>First return median</td></tr>
<tr><td>3</td><td>First return furthest</td></tr>
<tr><td>4</td><td>First return strength</td></tr>
<tr><td>5</td><td>Last return raw</td></tr>
<tr><td>6</td><td>Last return closest</td></tr>
<tr><td>7</td><td>Last return median</td></tr>
<tr><td>8</td><td>Last return furthest</td></tr>
<tr><td>9</td><td>Last return strength</td></tr>
<tr><td>10</td><td>Noise</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>RW</th><th>Read bytes</th><th>Write bytes</th><th>Persists</th><th>Description</th></tr>
<tr><td>28</td><td>Communication Mode</td><td>RW</td><td>1/ uint8</td><td>1/ uint8</td><td>Yes</td><td>The Communication mode sets the startup state. The following options are available:
Bit Mode Description
0 Auto Detect Interface (Default)
1 Serial UART
2 I²C
3 Serial UART (Legacy Header)
4 Serial UART (no startup message)</td></tr>
<tr><td>30</td><td>Stream</td><td>RW</td><td>4/ uint32</td><td>4/ uint32</td><td>No</td><td>Serial and USB interface only.
(If used on I²C, the data will not be retrievable.)
Reading from the stream command will indicate what type of data is currently being streamed.
Writing to the stream command will set the type of data to be streamed.
<table>
<tr><th>Value</th><th>Streamed data</th></tr>
<tr><td>0</td><td>disabled</td></tr>
<tr><td>5</td><td>(44) stream distance data cm</td></tr>
<tr><td>6</td><td>(45) stream distance data mm</td></tr>
</table></td></tr>
<tr><td>44</td><td>Distance data in cm</td><td>R</td><td>varies</td><td>-</td><td>-</td><td>Distance data in cm as measured by the LW20.
This command can be read any time, but if (30) stream is set to 5, this command will automatically output at the measurement update rate.
The data included will vary and be packed in order based on the configuration of the (27) distance output command.
<table>
<tr><th>Data output bit</th><th>Description</th><th>Size</th></tr>
<tr><td>0</td><td>First return raw (cm)</td><td>int16</td></tr>
<tr><td>1</td><td>First return closest (cm)</td><td>int16</td></tr>
<tr><td>2</td><td>First return median (cm)</td><td>int16</td></tr>
<tr><td>3</td><td>First return furthest (cm)</td><td>int16</td></tr>
<tr><td>4</td><td>First return strength (%)</td><td>int16</td></tr>
<tr><td>5</td><td>Last return raw (cm)</td><td>int16</td></tr>
<tr><td>6</td><td>Last return closest (cm)</td><td>int16</td></tr>
<tr><td>7</td><td>Last return median (cm)</td><td>int16</td></tr>
<tr><td>8</td><td>Last return furthest (cm)</td><td>int16</td></tr>
<tr><td>9</td><td>Last return strength (%)</td><td>int16</td></tr>
<tr><td>10</td><td>Background noise</td><td>int16</td></tr>
</table></td></tr>
<tr><td>45</td><td>Distance data in mm</td><td>R</td><td>varies</td><td>-</td><td>-</td><td>Distance data in mm as measured by the LW20.
This command can be read any time, but if (30) stream is set to 6, this command will automatically output at the measurement update rate.
The data included will vary and be packed in order based on the configuration of the (27) distance output command.
<table>
<tr><th>Data output bit</th><th>Description</th><th>Size</th></tr>
<tr><td>0</td><td>First return raw (mm)</td><td>Int32</td></tr>
<tr><td>1</td><td>First return closest (mm)</td><td>Int32</td></tr>
<tr><td>2</td><td>First return median (mm)</td><td>Int32</td></tr>
<tr><td>3</td><td>First return furthest (mm)</td><td>Int32</td></tr>
<tr><td>4</td><td>First return strength (%)</td><td>Int32</td></tr>
<tr><td>5</td><td>Last return raw (mm)</td><td>Int32</td></tr>
<tr><td>6</td><td>Last return closest (mm)</td><td>Int32</td></tr>
<tr><td>7</td><td>Last return median (mm)</td><td>Int32</td></tr>
<tr><td>8</td><td>Last return furthest (mm)</td><td>Int32</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>RW</th><th>Read bytes</th><th>Write bytes</th><th>Persists</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>9 | Last return strength (%) | Int32
10 | Background noise | Int32</td></tr>
<tr><td>50</td><td>Laser firing</td><td>RW</td><td>1/
uint8</td><td>1/
uint8</td><td>No</td><td>Reading this command will indicate the current laser firing state.
Writing to this command will enable or disable laser firing.
Value | Description
0 | Disabled
1 | Enabled</td></tr>
<tr><td>55</td><td>Temperature</td><td>R</td><td>4/
uint32</td><td>-</td><td>-</td><td>Reading this command will return the measured temperature in 0.01 of a degree.</td></tr>
<tr><td>92</td><td>Update rate</td><td>RW</td><td>4/
uint32</td><td>4/
uint32</td><td>Yes</td><td>This variable when read will indicate the current update rate in Hz.
When writing to this variable, the update rate can be set in Hz and the closest possible update rate to the request will be used and then reported back.</td></tr>
<tr><td>93</td><td>Update rate preselect list</td><td>RW</td><td>1/
uint8</td><td>1/
uint8</td><td>Yes</td><td>Controls the LW20's sampling update rate.
Reading this command will return the current update rate.
Writing this command will set the update rate.
Command value | Update rate samples/second
1 | 48
2 | 55
3 | 64
4 | 77
5 | 97
6 | 129
7 | 194
8 | 388
9 | 625
10 | 1250
11 | 2500
12 | 5000</td></tr>
<tr><td>85</td><td>Noise</td><td>R</td><td>4/
uint32</td><td>-</td><td>-</td><td>Reading this command will return the level of measured background noise.</td></tr>
<tr><td>94</td><td>Zero offset</td><td>RW</td><td>4/
int32</td><td>4/
int32</td><td>Yes</td><td>Changing this offset value will change the zero-distance position for the output, written and read in mm.</td></tr>
<tr><td>95</td><td>Lost signal counter</td><td>RW</td><td>4/
int32</td><td>4/
int32</td><td>Yes</td><td>Sets the number of lost signal returns before a lost signal indication is output on the distance value.
The distance output lost signal indication -1000.</td></tr>
<tr><td>90</td><td>Baud rate</td><td>RW</td><td>1/
uint8</td><td>1/
uint8</td><td>Yes</td><td>The serial baud rate used by the serial interface.
This parameter only takes effect when the serial interface is first enabled after power-up or restart.
Reading this command will return the baud rate.
Writing to this command will set the baud rate.
Value | Baud rate (bps)
0 | 9600
1 | 19200
2 | 38400
3 | 57600
4 | 115200
5 | 230400
6 | 460800</td></tr>
</table>

<table>
<tr><th>ID</th><th>Name</th><th>RW</th><th>Read bytes</th><th>Write bytes</th><th>Persists</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>7 | 921600</td></tr>
<tr><td>91</td><td>I2C address</td><td>RW</td><td>1/ uint8</td><td>1/ uint8</td><td>Yes</td><td>The I²C address value is in decimal.
Reading this command will return the I²C address.
Writing this command will set the I²C address.</td></tr>
<tr><td>137</td><td>Median filter enable</td><td>RW</td><td>1/ uint8</td><td>1/ uint8</td><td>Yes</td><td>Reading this command will return the status of the median filter.
Writing this command will set the status of the median filter.
Value | Description
0 | Disabled
1 | Enabled</td></tr>
<tr><td>138</td><td>Median filter size</td><td>RW</td><td>4/ int32</td><td>4/ int32</td><td>Yes</td><td>Reading this command will return the size of the median filter.
Writing this command will set the size of the median filter.
The valid range is 3 to 32.</td></tr>
<tr><td>139</td><td>Smoothing filter enable</td><td>RW</td><td>1/ uint8</td><td>1/ uint8</td><td>Yes</td><td>Reading this command will return the status of the smoothing filter.
Writing this command will set the status of the smoothing filter.
Value | Description
0 | Disabled
1 | Enabled</td></tr>
<tr><td>140</td><td>Smoothing factor</td><td>RW</td><td>4/ uint32</td><td>4/ uint32</td><td>Yes</td><td>Reading this command will return the strength of the smoothing filter.
Writing this command will set the strength of the smoothing filter.
The valid range is 1 to 99.</td></tr>
<tr><td>141</td><td>Rolling average enable</td><td>RW</td><td>1/ uint8</td><td>1/ uint8</td><td>Yes</td><td>Reading this command will return the status of the rolling average filter.
Writing this command will set the status of the rolling average filter.
Value | Description
0 | Disabled
1 | Enabled</td></tr>
<tr><td>142</td><td>Rolling average size</td><td>RW</td><td>4/ uint32</td><td>4/ uint32</td><td>Yes</td><td>Reading this command will return the size of the rolling average filter.
Writing this command will set the size of the rolling average filter.
The valid range is 2 to 32.</td></tr>
</table>

10.2 ASCII protocol

10.2.1 ASCII protocol – Command structure

A command string is made up of:

- A **command type** indicating what action the sensor must take
- A **channel**
- A **parameter** that is to be acted on (if applicable)
- An **index** indicating which parameter value is to be acted on, such as the high or the low setting (if applicable)

- A **comma**, if it is to be followed by a numeric value
- A **numeric value** (if required)
- **<CR><LF>** indicates the end of the command

<command type><channel><parameter><index><,><numeric value><CR><LF>

There are four command types, each using a single ASCII character:

? The host controller instructs the sensor to read and reply with a single setting or result value.

The host controller instructs the sensor to change a setting value. All sensor parameters can be changed in this way. (These parameters can also be changed through LightWare Studio). These commands include instructions to connect or enable/disable functions.

% The host controller instructs the sensor to save specific setting changes to memory. The setting remains when the sensor is switched off.

$ The host controller instructs the sensor to stream live value results continuously or update a result or setting every time it changes. Streaming has its own ASCII command strings format, as detailed further below.

There are six channels, each containing a group of related information:

P For information about the LightWare product

L For all laser information such as distance measurements, (including alarm distances) and related laser settings. (All distances are in meters.)

S For all servo information including servo position and settings and alarm angles

A For alarm state information

C The communication channel handles serial UART and I^2C data settings

10.2.2 ASCII protocol - The reply to the ASCII command

The sensor will reply to the command in the following format:

<channel/parameter/index/numeric value as received in command><:><reply value><CR><LF>

If there is more than one reply value, they will be separated by commas. For purposes of the document, replies are show in lower case to help distinguish from commands.

10.2.3 ASCII protocol - Example command and reply

A typical command sent by the host controller:

?PN<CR><LF> which means, “Reply with your product name”

The sensor replies with:

pn:LW20 <cr><lf> which means, “Product Name: LW20”

The commands list table on the following pages contains many common commands and shows how to build the command you require.

10.2.4 ASCII protocol - Saving

The command **%P<CR>** will save all settings to permanent memory. Note that saving uses the sensor’s flash memory, which has limited writes: This is not a limitation for everyday use, but the user must be careful not to accidentally call the save command in a tight loop, as the flash memory will quickly be exhausted with hundreds of thousands of saves.

10.2.5 ASCII protocol - Command list

Table 9: LW20/C channel commands list

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
Product channel commands									
?						**?**	p:LW20,1.0.0,11	Reply with the product name, firmware version, hardware version	Same return for both commands
?	P					**?P**			
?	P	N				**?PN**	pn:LW20	Reply with the product **name**	
?	P	S				**?PS**	ps:1.0.0	Reply with the product **firmware version**	
?	P	F				**?PF**	pf:11	Reply with the product **hardware version**	
%	P					**%P**	%p:	Save all new settings in **all channels** to permanent memory.	
Communication channel commands									
?	C	B				**?CB**	cb:7	Reply with the serial port **baud rate**	**Numeric value** for baud rate: 0 = 9 600 1 = 19 200 2 = 38 400 3 = 57 600 4 = 115 200 5 = 230 400 6 = 460 800 7 = 921 600
#	C	B		,		**#CB,6**	cb:6	Change the serial port **baud rate** to 460 800 (rate 6)	
?	C	I				**?CI**	ci:0x66	Reply with the **I²C address** in hexadecimal	
#	C	I		,		**#CI,0x78**	ci:0x78	Change the **I²C address** to 0x78	Adjustment range: 0 to 0x7F
%	C					**%C**	%c:	Save all new communication channel settings to permanent memory	
Laser channel commands									
?	L	M				**?LM**	lm:5	Reply with which **measuring mode** (update rate) is active	**Numeric value** for readings per second:

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
#	L	M		,		**#LM,1**	lm:1	Change the update rate to 388 readings per second	1 = 48 2 = 55 3 = 65 4 = 78 5 = 97 6 = 129 7 = 194 8 = 388 9 = 625 10 = 1250 11 = 2500 12 = 5000
?	L	O				**?LO**	lo:0.14	Reply with distance measurement **datum offset**	
#	L	O		,		**#LO,0.56**	lo:0.56	Change to a new **zero distance offset** to 0.56 meters	Adjustment range is -10.00 to +10.00
?	L	C				**?LC**	lc:8	Reply with the number of **lost signal confirmations**	
#	L	C		,		**#LC,16**	lc:16	Change the number of **lost signal threshold** to 16	Adjustment range: 1 to 250
?	L	F				**?LF**	lf:1	Reply with the laser **state**, (whether it is running or not)	**Numeric value** for laser state: 0 = laser is off 1 = laser is running
#	L	F		,		**#LF,0**	lf:0	Change the laser **state** to off	
?	L	T				**?LT**	lt:35.7	Reply with the sensor's **internal temperature** in °C	
?	L	N				**?LN**	ln:0.5	Reply with the level of **background noise**	
?	L					?L	l:3.35	Reply with the default raw distance	
?	L	D				**?LD**	ld,0:23.67	Reply with the default *median* **distance**	
?	L	D	F			**?LDF**	ldf,0:56.98	Reply with the *median* **distance to the first return**	

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
?	L	D	F	,	0	**?LDF,0**	ldf,0:45.32	Reply with the *median* **distance to the first return**	**Numeric value** for returns: 0 = median 1 = raw 2 = closest 3 = furthest
?	L	D	F	,	1	**?LDF,1**	ldf,1:32.78	Reply with the *raw* **distance to the first return**	
?	L	D	F	,	2	**?LDF,2**	ldf,2:65.12	Reply with the *closest* **distance to the first return**	
?	L	D	F	,	3	**?LDF,3**	ldf,3:34.23	Reply with the *furthest* **distance to the first return**	
?	L	D	L			**?LDL**	ldl,0:56.98	Reply with the *median* **distance to the last return**	Same return as including ",0" numeric value, see next entry
?	L	D	L	,	0	**?LDL,0**	ldl,0:45.32	Reply with the *median* **distance to the last return**	**Numeric value** for returns: 0 = median 1 = raw 2 = closest 3 = furthest
?	L	D	L	,	1	**?LDL,1**	ldl,1:32.78	Reply with the *raw* **distance to the last return**	
?	L	D	L	,	2	**?LDL,2**	ldl,2:65.12	Reply with the *closest* **distance to the last return**	
?	L	D	L	,	3	**?LDL,3**	ldl,3:34.23	Reply with the *furthest* **distance to the last return**	
?	L	H				**?LH**	lh:100	Reply with the **signal strength** of first default signal	As a percentage of full signal
?	L	H	F			**?LHF**	lhf:100	Reply with the **signal strength of first return**	
?	L	H	L			**?LHL**	lhl:78	Reply with the **signal strength of last return**	
?	L	A				**?LA**	la:3.00	Reply with **Alarm A** distance	Same return for both commands
?	L	A	A			**?LAA**	laa:3.00	Reply with **Alarm A** distance	
#	L	A	A	,		**#LAA,5.00**	laa:5.00	Change **Alarm A** distance to 5.00 meters	Adjustment range is 0.00 to 100.00

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
?	L	A	B			**?LAB**	lab:3.00	Reply with **Alarm B** distance	
#	L	A	B	,		**#LAB,6.00**	lab:6.00	Change **Alarm B** distance to 6.00 meters	Adjustment range is 0.00 to 100.00
?	L	A	H			**?LAH**	lh:0.00	Reply with the **alarm hysteresis** value	
#	L	A	H	,		**#LAH,0.05**	lh:0.05	Change the **alarm hysteresis** value to 0.05 meters	Adjustment range is 0.00 to 10.00
%	L					**%L**	%l:	Save all new settings to permanent memory	
Servo channel commands									
?	S					**?S**	s:0	Reply with whether a servo is **connected**	**Numeric value** for servo connection: 0 = not connected 1 = connected
?	S	C				**?SC**	sc:0	Reply with whether a servo is **connected**	
#	S	C		,		**#SC,1**	sc:1	**Connect** the servo	
?	S	S				**?SS**	ss:0	Reply with whether the servo is **scanning** or not	**Numeric value** for scanning state: 0 = scanning off 1 = scanning on
#	S	S		,		**#SS,1**	ss:1	Switch servo **scanning** on	
?	S	P				**?SP**	sp:0.0	Reply with the current servo **position** in degrees	
#	S	P		,		**#SP,20.5**	sp:20.5	Move the servo to a new **position** of 20.5°	Adjustment range: -180 to 180
?	S	M				**?SM**	sm:15.0	Reply with the current servo position in degrees	
#	S	M				**#SM**	sm:0.0	Move the servo to the **middle** position	
?	S	W	L			**?SWL**	swl:1000.0	Reply with the **minimum allowed PWM time** in µs	
#	S	W	L	,		**#SWL,1100.0**	swl:1100.0	Change the **minimum allowed PWM time** to 1100.0 µs	Adjustment range: 0.00 to 2999.00
?	S	W	H			**?SWH**	swh:2000.0	Reply with the **maximum allowed PWM time** in µs	
#	S	W	H	,		**#SWH,1900**	swh:1900.0	Change the **maximum allowed PWM time** to 1900.0 µs	Adjustment range: 0.00 to 2999.00

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
?	S	W	S			**?SWS**	sws:9.00	Reply with the **PWM scale** of the servo, in µs per degree	
#	S	W	S	,		**#SWS,10.00**	sws:10.00	Change the **PWM scale** of the servo to 10.00 µs per degree	Adjustment range: 0.10 to 1000.00
?	S	R				**?SR**	sr:4	Reply with the number of servo **steps per reading**	
#	S	R		,		**#SR,8**	sr:8	Change the number of servo **steps per reading** to 8 degrees	Adjustment range: 1 to 32
?	S	L				**?SL**	sl:1.32	Reply with the **servo lag angle** in degrees	
#	S	L		,		**#SL,2.57**	sl:2.57	Change the **servo lag angle** to 2.57°	Adjustment range: 0.00 to 90.00
?	S	F	L			**?SFL**	sfl:-45.0	Reply with the **field of view lower angle** in degrees	
#	S	F	L	,		**#SFL,-30.0**	sfl:-30.0	Change the **field of view lower angle** to -30.0°	Adjustment range: -180.00 to 180.00
?	S	F	H			**?SFH**	sfh:45.0	Reply with the **field of view higher angle** in degrees	
#	S	F	H	,		**#SFH,30.0**	sfh:30.0	Change the **field of view higher angle** to 30.0°	Adjustment range: -180.00 to 180.00
?	S	T				**?ST**	st:1	Reply with the **scan type** being used	**Numeric value** for scan type: 0 = bidirectional scan 1 = unidirectional scan
#	S	T		,		**#ST,0**	st:0	Change the **scan type** to bidirectional	
?	S	A	L			**?SAL**	sal:-45.0	Reply with the **scanning Alarm A lower angle** in degrees	
#	S	A	L	,		**#SAL,-15.0**	sal:-15.0	Change the **scanning Alarm A lower angle** to -15.0°	Adjustment range: -180.00 to 180.00
?	S	A	H			**?SAH**	sah:45.0	Reply with the **scanning Alarm A higher angle** in degrees	

Build Command						Command Each command must end with <CR><LF>	Typical reply Each reply will end with <CR><LF>	Description	Notes
Command type	Channel	Parameter	Index	Comma	Numeric value				
#	S	A	H	,		**#SAH,0.0**	sah:0.0	Change the **scanning Alarm A higher angle** to 0.0°	Adjustment range: -180.00 to 180.00
?	S	B	L			**?SBL**	sbl:-45.0	Reply with the **scanning Alarm B lower angle** in degrees	
#	S	B	L	,		**#SBL,0.0**	sbl:0.0	Change the **scanning Alarm B lower angle** to 0.0°	Adjustment range: -180.00 to 180.00
?	S	B	H			**?SBH**	sbh:45.0	Reply with the **scanning Alarm B higher angle** in degrees	
#	S	B	H	,		**#SBH,15.0**	sbh:15.0	Change the **scanning Alarm B higher angle** to 15°	Adjustment range: -180.00 to 180.00
%	S					**%S**	%s:	Save all new servo channel settings to permanent memory	
Alarm channel commands									
?	A					**?A**	a:0,1	Reply with the states of both alarms	**Numeric value** for alarm state: 0 = alarm off 1 = alarm on
?	A	A				**?AA**	aa:0	Reply with the state of **Alarm A**	
?	A	B				**?AB**	ab:1	Reply with the state of **Alarm B**	

10.3 Legacy protocol

10.3.1 Legacy protocol – Command structure

The sensor supports a legacy protocol, with some special commands compatible with older systems in binary format.

Table 10: LW20/C legacy commands list

Command **Each command must end with <CR><LF>**	**Typical reply** **Each return will end with <CR><LF>**	**Description**	**Notes**
Legacy commands for the serial port			
D' or **'d'**	78.32	Output the current **distance** once only	
?SU	su:0	Reply with whether distance **streaming** is active	**Numeric value** for streaming state: 0 = off 1 = on
#SU,1	su:1	Switch distance **streaming** on or off	
0'	No response	Enable I²C legacy distance **streaming**	
Legacy commands for the I²C port			
0' or **0** or **1** or **129**	78.23	Enable output binary coded **distance** in centimeters	Send any other command to disable

10.4 Streaming commands list

The sensor can be instructed to stream live data as it is scanned, without the host controller having to repeatedly resend a '**?**' command. The sensor has five streams (denoted by $1, $2, $3, $4, and $5), and each can be set to stream a different reading. These five streams can be seen as five columns in LightWare Studio.

The streaming ASCII command strings format is as follows:

<$><stream number><,>< channel/parameter/index/numeric to be streamed><CR><LF>

Streams are a settable parameter like any other, and the stream can be saved to persist through boot up using the command **%P<CR>** which saves all settings. You can clear all streams by sending **$<CR>**.

The commands list table on the following pages contains several useful examples of streaming commands.

Table 11: LW20/C streaming commands list

Build Command				Command Each command must end with <CR><LF>	Typical reply Each return will end with <CR><LF>	Description	Notes
Command Type	Stream number	Comma	Stream parameter				
Streaming commands							
$	1	,	SS	**$1,SS**	ss:30.5,15.56,27.43	Stream **servo scanning** data on Stream 1: angle, first return, last return	
$	2	,	LT	**$2,LT**	Data steam	Stream the sensor's **internal temperature** on Stream 2	
$	1	,	LDF	**$1,LDF**	Data steam	Stream the *median* **distance to the first return** on Stream 1	
$	2	,	LHF	**$2,LHF**	Data steam	Stream the **signal strength of first return** on Stream 2	
$				**$**	Data steam	**Stop** all streams	

11 Firmware updates

Occasionally, LightWare will release new firmware for your sensor, to address bug fixes or introduce additional features. All registered customers will receive an email notification when new firmware is released for their LightWare sensor.

Caution: LightWare strongly advises that all LightWare sensors are kept up to date with their latest firmware revision.

You can check whether your sensor is equipped with the latest firmware and access updates directly through LightWare Studio as follows:

1. Select *Upgrade* from the left panel.

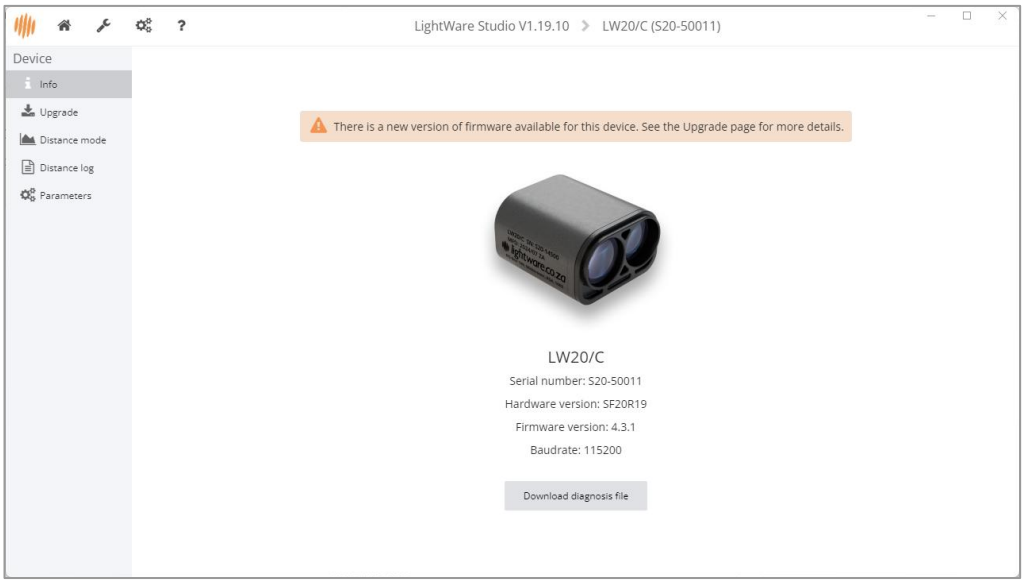

Figure 31: LightWare Studio device information page

2. The page will display the firmware version currently installed on the sensor and indicate whether any recent upgrades are available for download.

3. If you need to upgrade, click the *Install* button, and follow the instructions.

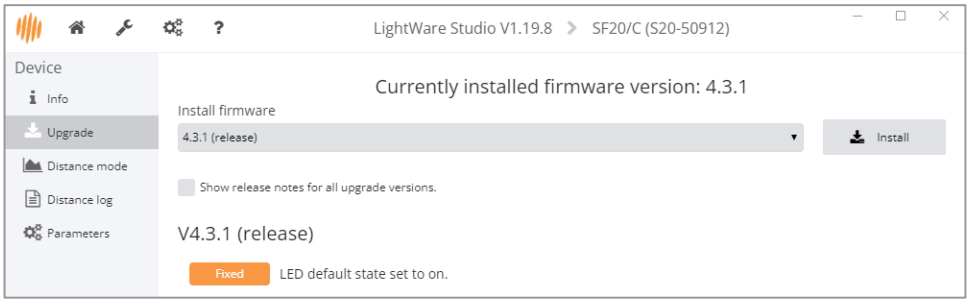

Figure 32: LightWare Studio firmware upgrade page

4. The page will display the currently installed firmware version on the sensor, and it will indicate whether any recent upgrades are available for download.

5. If you need to upgrade, click the *Install* button, followed by OK to confirm.

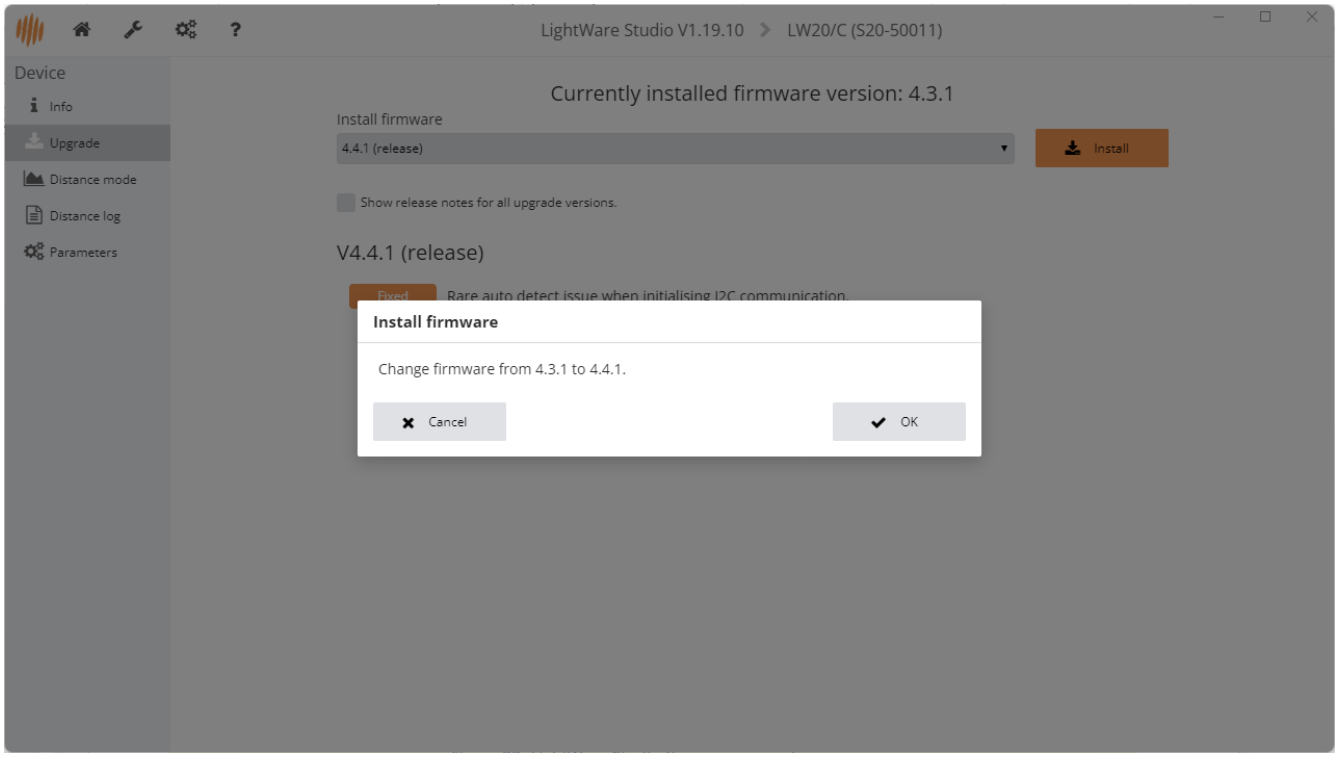

Figure 33: Confirmation of firmware upgrade

6. The firmware will be installed to your device, and the device will automatically reboot.

12 Troubleshooting

Table 10: LW20/C troubleshooting

Problem	Solution
1. Sensor outputs a short distance reading or distorted distance reading	• The sensor is receiving a signal caused by scattered light from a close-by object in the vicinity of the beam, such as a desk surface, landing gear, pole, or highly reflective object. Relocate your sensor or the object and test again.
2. Sensor outputs -1 / 230	• This is an out-of-range condition. There is no measurable object within the sensor's range.
3. Sensor is not communicating with the serial UART controller at all.	• Ensure that the sensor's baud rate is compatible with the controller. • Ensure that the sensor's TXD and RXD lines are connected to the controller's RXD and TXD lines, respectively. • If using ArduPilot or PX4, ensure that the correct parameters for sensor integration have been set. • Ensure that the sensor supply voltage is within the specified range and is not dropping below the specified minimum level. If using a separate power supply, ensure a common ground.
4. Sensor is not communicating with the I²C controller at all.	• Ensure that the sensor SDA and SCL lines are connected to the controller SDA and SCL lines, respectively. • If using ArduPilot or PX4, ensure that the correct parameters for sensor integration have been set. • Ensure that the sensor supply voltage is within the specified range and is not dropping below the specified minimum level. • If using a separate power supply, ensure a common ground.
5. Servo does not scan on startup	• Ensure "Servo scan on startup" is enabled in LightWare Studio. • • Ensure servo ground and sensor ground is connected to the same ground plane • • Servo motors are rated for a certain maximum and minimum PWM value. Set the servo max/min PWM value in LightWare Studio according to the servo you are using.
6. Alarms not visible	• Check alarm zone distances are set correctly • Ensure that GPIO mode is set to output either alarm A or alarm B • Alarm takes time to reset, check "GPIO alarm confirmation count"
7. Sensor stops communicating during flight	• Check the power supply to the sensor. • Ensure all cable connections are properly seated and secured.
8. Readings are erratic or changing too fast	• Check the update rate and ensure it is suitable for the application. (Slower update rates are advised for altimetry.) • Consider using the built-in filters to remove background noise.

	• Check the grounding via the shielded cable. • Investigate possible sources of electromagnetic interference (EMI).
9. The sensor is running hot	• Ensure adequate ventilation and heat sinking to prevent heat build-up.
10. Filtered output appears slow/irregular	• Ensure that the filter selected and filter size selected matches your application. • Ensure only one filter is selected at a time, unless bench tested
11. Sensor not working via Serial adapter	• If the sensor was set to startup in I^2C mode, or set to a baud please contact our technical support team for assistance. • The maximum baud rate supported by the LightWare USB adaptor is 921600. If the baud rate was set to a value higher than 921600, please contact our technical support team for assistance.

For issues not covered above, refer to the FAQs in the LightWare website resource center or contact LightWare's dedicated technical support team for assistance with remote testing of your LightWare sensor.

13 Repair and maintenance

13.1 Maintenance and calibration

The LightWare microLiDAR® sensor contains no moving parts, and **no regular maintenance** is required. The sensor **does not need regular calibration** and will remain true to specification throughout its lifespan if used as directed.

13.2 Cleaning

If the LightWare microLiDAR® lenses collect dust, use a clean, soft cloth or air compressor to remove it. The lenses are coated with an anti-reflective, non-scratch coating. Only appropriate lens cleaning materials should be used to avoid scratching the sensor's lens or damaging the coating. Keep the device free from moisture in accordance with its IP rating.

13.3 Electrical safety

- Check all electrical connections are isolated and that there are no exposed wires.
- Ensure the power supplied to the device does not exceed the maximum rated voltages specified in the technical specifications section.
- Keep the device free from moisture in accordance with the IP rating.

13.4 Service and repairs

If you experience any problems with your sensor, please contact the LightWare technical support desk for in-field diagnostics before sending the unit to LightWare. During in-field support, you may be requested to supply the device's diagnostics file, which can be downloaded from LightWare Studio from the device *Info* page.

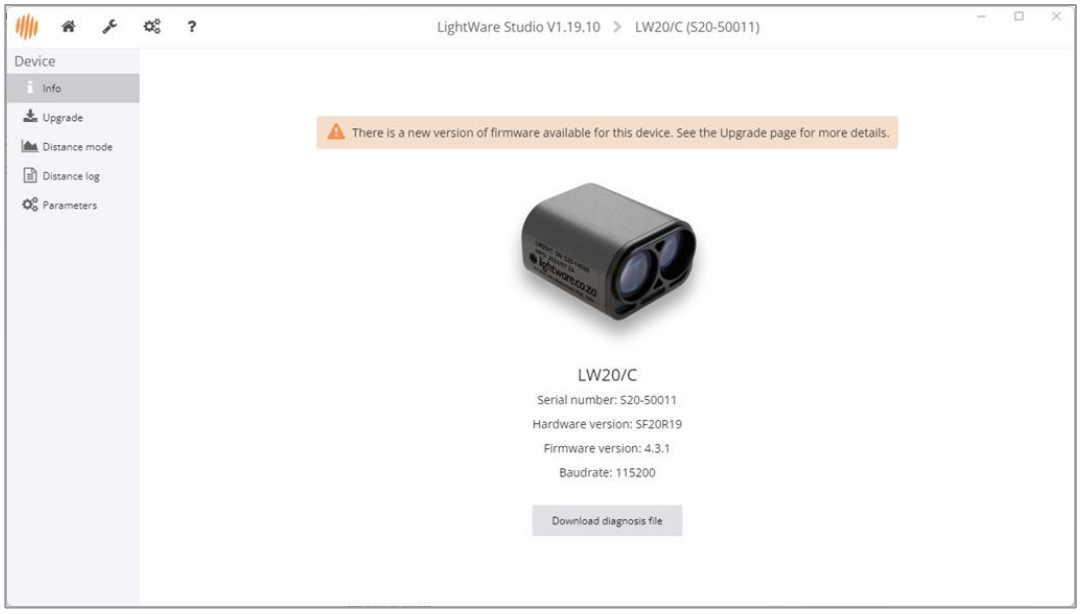

Figure 34: LightWare Studio device information page

If the unit needs to be returned to LightWare for repairs, LightWare support will assist you with the Return Merchandise Authorization (RMA) procedure.

14 End-of-life safe disposal

At LightWare, we are committed to protecting the environment and ensuring that our products have minimal impact on the planet at the end of their lifecycle. As your device reaches the end of its operational life, we encourage you to dispose of it in a responsible and environmentally friendly manner.

Please do not dispose of LightWare sensors with general household or commercial waste.

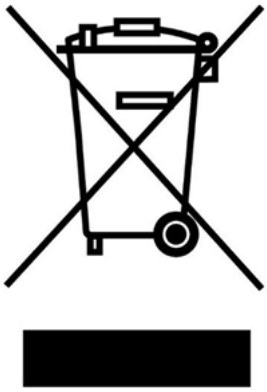

Figure 35: End-of-life disposal

LightWare sensors consist of ABS and other plastics, acrylic, and glass components, which are widely recyclable. The electronic PC board assembly should be disposed of through a reputable electronic waste recycler in your area. Alternatively, return your device to LightWare for safe disposal.

15 Document revision history

Table 11: Revision history

Revision	Date	Comments
Rev 13	2024/12/19	Review entire manual and move to new standard product guide template. Check and update product information and LightWare Studio interface. Remove GUI and USB info. Review specifications entirely. Correct eye safety data to refer to non-ionizing radiation.
Rev 12	2020/09/28	Correct average laser power, pulse frequency, NOHD distance, and cable layout. Update FDA Accession number.
Rev 11	2020/05/22	Separate LW20 information from LW20. Update branding, layout. Include CE certification. New datasheet format. Update power supply current to 100 mA. Remove housing details. Include HMI commands description table.
Rev 10	2019/11/28	Include CE certification. New datasheet format. Separate LW20 information from LW20.
Rev 9	2018/07/23	Included heat sink contact area figure.
Rev 8	2018/07/04	Remove “Encoded laser pulses prevent interference for other lasers” and “Power saving mode to save energy when not in use”. Update power supply current to 130 mA. Add “Objects must be separated by approximately 5 meters before they are seen as separate return signals.” Include “The LW20 is an OEM module that requires the customer to provide appropriate heat sinking and EMI shielding”. Updated "The LW20 will wait for the first command to confirm which communication mode to select (serial or I^2C). The first command will not get a response.” Correct serial port baud rate to 9600 to 921600. Updated pinout diagram. Replace “%C – Save the latest communications settings to permanent memory" with “%P – Save all settings to permanent memory”. Update command list table. Update LightWare Terminal and HMI interface instructions.
Rev 7	2018/04/26	Update new models to replace obsolete models. Update with single I^2C and serial communication cable. Update USB cable figures. Remove references to 5 V logic. Include “Ensure that adequate ventilation or heat sinking is provided if the LW20 is incorporated into a custom enclosure, as heat build-up could occur.”
Rev 6	2018/04/05	Update range to “0.2 to 100 m (sunlit white wall, 45 readings per second)” and beam divergence to 0.3°. Update USB-ISS module figure.
Rev 5	2017/07/27	Add pinout diagram.
Rev 4	2017/05/25	Highlight factory default I^2C bus address as 0x66 and serial baud rate as 115200.
Rev 3	2017/05/05	Change baud rate from 921600 to 115200. Updated USB-ISS module figure and LightWare Terminal screens version. Include data streaming section.
Rev 2	2017/04/10	Include section “How to communicate with the LW20 using the Devantech USB to I^2C module”.
Rev 1	2017/03/03	Include FDA approval. Update cable figures to include ferrite core.
Rev 0	2017/02/20	First edition