USER MANUAL

Gocator 1300 Series

Document revision: C

Copyright

Copyright © 2015 by LMI Technologies, Inc. All rights reserved.

Proprietary
This document, submitted in confidence, contains proprietary information which shall not be reproduced or transferred to other documents or disclosed to others or used for manufacturing or any other purpose without prior written permission of LMI Technologies Inc.

No part of this publication may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine readable form without prior written consent of LMI Technologies, Inc.

Trademarks and Restrictions
Gocator™ is a registered trademark of LMI Technologies, Inc. Any other company or product names mentioned herein may be trademarks of their respective owners.

Information contained within this manual is subject to change.

This product is designated for use solely as a component and as such it does not comply with the standards relating to laser products specified in U.S. FDA CFR Title 21 Part 1040.

Contact Information
For more information, please contact LMI Technologies.

LMI Technologies, Inc. 1673 Cliveden Ave.
Delta, BC V3M 6V5
Canada

Telephone: +1 604 636 1011
Facsimile: +1 604 516 8368

www.lmi3D.com

Table of Contents

Introduction

The Gocator 1300 series of laser displacement sensors is designed for 3D measurement and control applications. Gocator sensors are configured using a web browser and can be connected to a variety of input and output devices.

This documentation describes how to connect, configure, and use a Gocator. It also contains reference information on the device's protocols and job files.

Notational Conventions

This guide uses the following notational conventions:

⚠	Follow these safety guidelines to avoid potential injury or property damage.

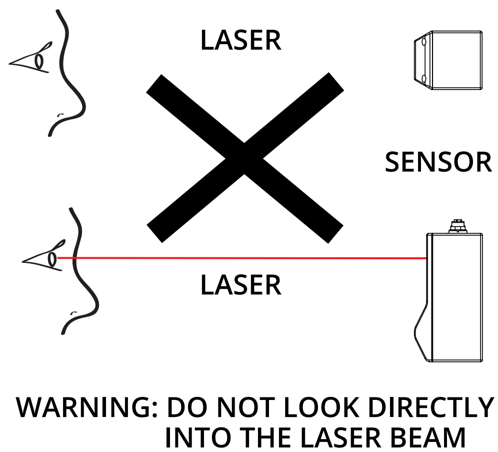
🗋	Consider this information in order to make best use of the product.

Safety and Maintenance

The following sections describe the safe use and maintenance of Gocator sensors.

Laser Safety

Gocator sensors contain semiconductor lasers that emit visible or invisible light and are designated as Class 2M, Class 3R, or Class 3B, depending on the chosen laser option.



Gocator sensors are referred to as *components*, indicating that they are sold only to qualified customers for incorporation into their own equipment. These sensors do not incorporate safety items that the customer may be required to provide in their own equipment (e.g., remote interlocks, key control; refer to references for detailed information). As such, these sensors do not fully comply with the standards relating to laser products specified in IEC 60825-1 and FDA CFR Title 21 Part 1040.

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

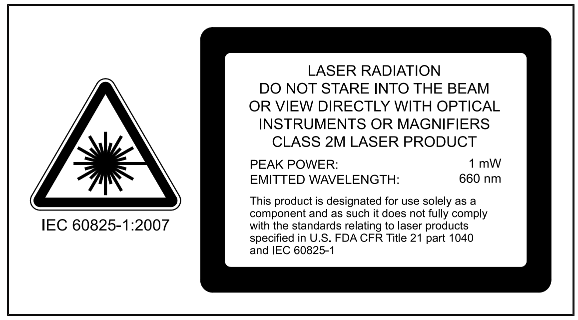
References

1. *International standard IEC 60825-1 (2001-08) consolidated edition*, Safety of laser products – Part 1: Equipment classification, requirements and user's guide.
2. *Technical report 60825-10*, Safety of laser products – Part 10. Application guidelines and explanatory notes to IEC 60825-1.
3. *Laser Notice No. 50*, FDA and CDRH http://www.fda.gov/cdrh/rad-health.html

Laser Classes

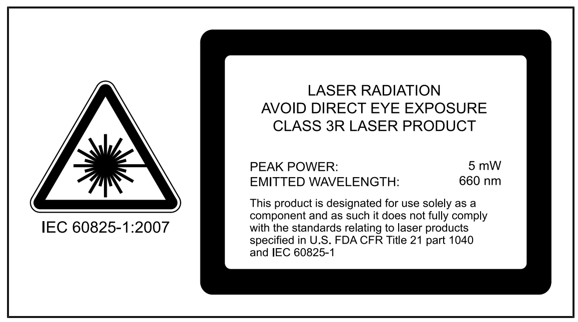
Class 2M laser components

Class 2M laser components would not cause permanent damage to the eye under reasonably foreseeable conditions of operation, provided that any exposure can be terminated by the blink reflex (assumed to take 0.25 seconds). Because classification assumes the blink reflex, the wavelength of light must be in the visible range (400 nm to 700 nm). The Maximum Permissible Exposure (MPE) for visible radiation for 0.25 seconds is 25 watts per square meter, which is equivalent to 1 mW entering an aperture of 7 mm diameter (the assumed size of the pupil).



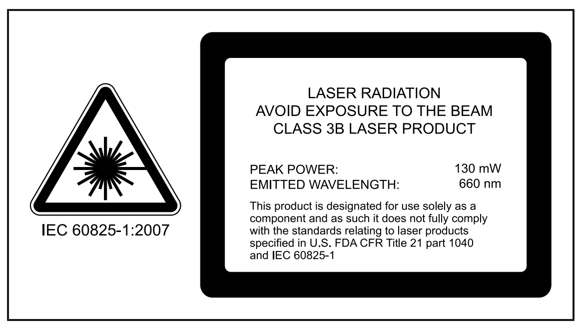
Class 3R laser components

Class 3R laser products emit radiation where direct intrabeam viewing is potentially hazardous, but the risk is lower with 3R lasers than for 3B lasers. Fewer manufacturing requirements and control measures for 3R laser users apply than for 3B lasers.



Class 3B laser components

Class 3B components are unsafe for eye exposure. Usually only ocular protection will be required. Diffuse reflections are safe if viewed for less than 10 seconds.



Labels reprinted here are examples only. For accurate specifications, refer to the label on your sensor.

Precautions and Responsibilities

Precautions specified in IEC 60825-1 and FDA CFR Title 21 Part 1040 are as follows:

Requirement	Class 2M	Class 3R	Class 3B
Remote interlock	Not required	Not required	Required*
Key control	Not required	Not required	Required – cannot remove

Requirement	Class 2M	Class 3R	Class 3B
			key when in use*
Power-on delays	Not required	Not required	Required*
Beam attenuator	Not required	Not required	Required*
Emission indicator	Not required	Not required	Required*
Warning signs	Not required	Not required	Required*
Beam path	Not required	Terminate beam at useful length	Terminate beam at useful length
Specular reflection	Not required	Prevent unintentional reflections	Prevent unintentional reflections
Eye protection	Not required	Not required	Required under special conditions
Laser safety officer	Not required	Not required	Required
Training	Not required	Required for operator and maintenance personnel	Required for operator and maintenance personnel

**LMI Class 3B laser components do not incorporate these laser safety items. These items must be added and completed by customers in their system design.*

Class 3B Responsibilities

LMI Technologies has filed reports with the FDA to assist customers in achieving certification of laser products. These reports can be referenced by an accession number, provided upon request. Detailed descriptions of the safety items that must be added to the system design are listed below.

Remote Interlock

A remote interlock connection must be present in Class 3B laser systems. This permits remote switches to be attached in serial with the keylock switch on the controls. The deactivation of any remote switches must prevent power from being supplied to any lasers.

Key Control

A key operated master control to the lasers is required that prevents any power from being supplied to the lasers while in the OFF position. The key can be removed in the OFF position but the switch must not allow the key to be removed from the lock while in the ON position.

Power-On Delays

A delay circuit is required that illuminates warning indicators for a short period of time before supplying power to the lasers.

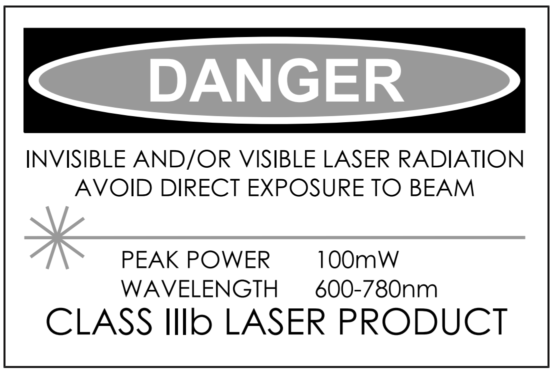
Beam Attenuators

A permanently attached method of preventing human access to laser radiation other than switches, power connectors or key control must be employed. On some LMI laser sensors, the beam attenuator is supplied with the sensor as an integrated mechanical shutter.

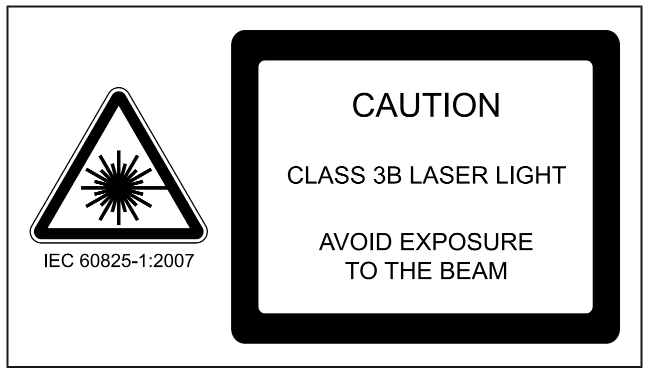
Emission Indicator
It is required that the controls that operate the sensors incorporate a visible or audible indicator when power is applied and the lasers are operating. If the distance between the sensor and controls is more than 2 meters, or mounting of sensors intervenes with observation of these indicators, then a second power-on indicator should be mounted at some readily-observable position. When mounting the warning indicators, it is important not to mount them in a location that would require human exposure to the laser emissions. User must ensure that the emission indicator, if supplied by OEM, is visible when viewed through protective eyewear.

Warning Signs
Laser warning signs must be located in the vicinity of the sensor such that they will be readily observed.

Examples of laser warning signs are as follows:



FDA warning sign example



IEC warning sign example

Systems Sold or Used in the USA

Systems that incorporate laser components or laser products manufactured by LMI Technologies require certification by the FDA.

Customers are responsible for achieving and maintaining this certification.

Customers are advised to obtain the information booklet *Regulations for the Administration and Enforcement of the Radiation Control for Health and Safety Act of 1968: HHS Publication FDA 88-8035*.

This publication, containing the full details of laser safety requirements, can be obtained directly from the FDA, or downloaded from their web site at http://www.fda.gov/cdrh.

Electrical Safety

⚠	Failure to follow the guidelines described in this section may result in electrical shock or equipment damage.

Sensors should be connected to earth ground
All sensors should be connected to earth ground through their housing. All sensors should be mounted on an earth grounded frame using electrically conductive hardware to ensure the housing of the sensor

is connected to earth ground. Use a multi-meter to check the continuity between the sensor connector and earth ground to ensure a proper connection.

Minimize voltage potential between system ground and sensor ground
Care should be taken to minimize the voltage potential between system ground (ground reference for I/O signals) and sensor ground. This voltage potential can be determined by measuring the voltage between Analog_out- and system ground. The maximum permissible voltage potential is 12 V but should be kept below 10 V to avoid damage to the serial and encoder connections.

See *Gocator 1300 I/O Connector* (page 235) for a description of connector pins.

Use a suitable power supply
The +24 to +48 VDC power supply used with Gocator sensors should be an isolated supply with inrush current protection or be able to handle a high capacitive load.

Use care when handling powered devices
Wires connecting to the sensor should not be handled while the sensor is powered. Doing so may cause electrical shock to the user or damage to the equipment.

Environment and Lighting

Avoid strong ambient light sources
The imager used in this product is highly sensitive to ambient light hence stray light may have adverse effects on measurement. Do not operate this device near windows or lighting fixtures that could influence measurement. If the unit must be installed in an environment with high ambient light levels, a lighting shield or similar device may need to be installed to prevent light from affecting measurement.

Avoid installing sensors in hazardous environments
To ensure reliable operation and to prevent damage to Gocator sensors, avoid installing the sensor in locations

- that are humid, dusty, or poorly ventilated;
- with a high temperature, such as places exposed to direct sunlight;
- where there are flammable or corrosive gases;
- where the unit may be directly subjected to harsh vibration or impact;
- where water, oil, or chemicals may splash onto the unit;
- where static electricity is easily generated.

Ensure that ambient conditions are within specifications
Gocator sensors are suitable for operation between 0–50° C and 25–85% relative humidity (non-condensing). Measurement error due to temperature is limited to 0.015% of full scale per degree C.

The Master 100/400/800/1200/2400 is similarly rated for operation between 0–50° C.

The storage temperature is -30–70° C.

⚠	The sensor must be heat-sunk through the frame it is mounted to. When a sensor is properly heat sunk, the difference between ambient temperature and the temperature reported in the sensor's health channel is less than 15° C.

⚠	Gocator sensors are high-accuracy devices, and the temperature of all of its components must therefore be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature in the sensor.

Sensor Maintenance

Keep sensor windows clean
Gocator sensors are high-precision optical instruments. To ensure the highest accuracy is achieved in all measurements, the windows on the front of the sensor should be kept clean and clear of debris.

Use care when cleaning sensor windows
Use dry, clean air to remove dust or other dirt particles. If dirt remains, clean the windows carefully with a soft, lint-free cloth and non-streaking glass cleaner or isopropyl alcohol. Ensure that no residue is left on the windows after cleaning.

Turn off lasers when not in use
LMI Technologies uses semiconductor lasers in 3D measurement sensors. To maximize the lifespan of the sensor, turn off the laser when not in use.

Avoid excessive modifications to files stored on the sensor
Settings for Gocator sensors are stored in flash memory inside the sensor. Flash memory has an expected lifetime of 100,000 writes. To maximize lifetime, avoid frequent or unnecessary file save operations.

Getting Started

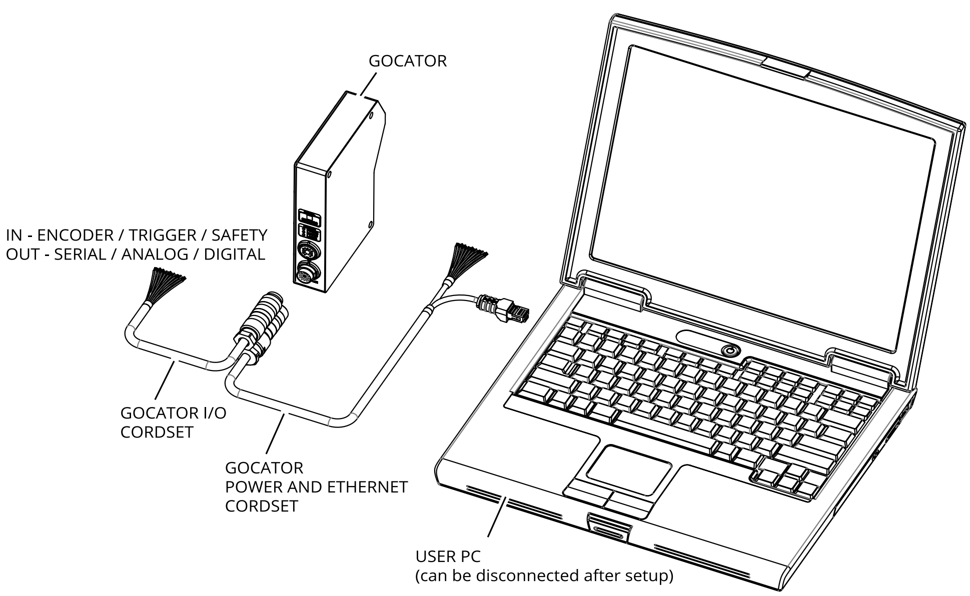
The following sections provide system and hardware overviews, in addition to installation and setup procedures.

System Overview

Gocator sensors can be installed and used in a variety of scenarios. Sensors can be connected as standalone devices, dual-sensor systems, or multi-sensor systems.

Standalone System

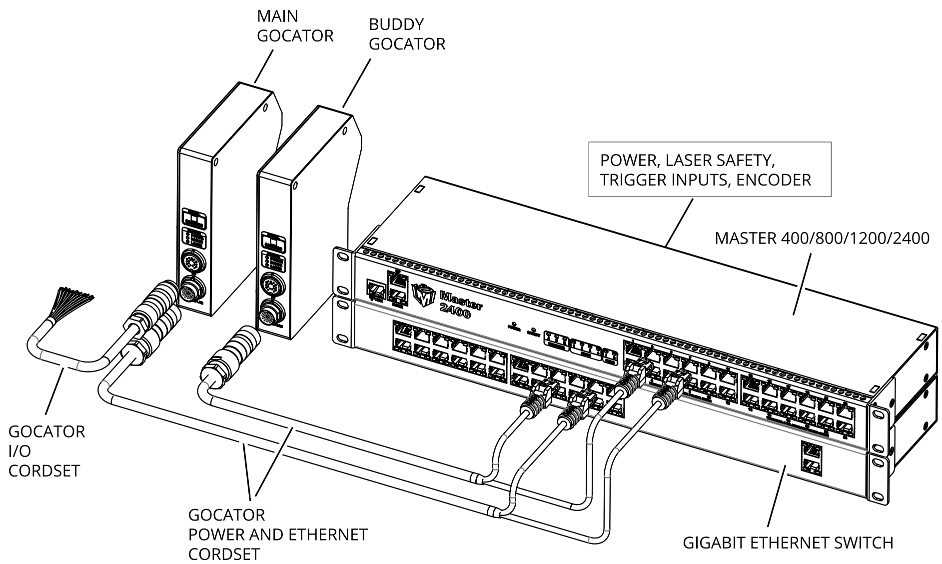
Standalone systems are typically used when only a single Gocator sensor is required. The sensor can be connected to a computer's Ethernet port for setup and can also be connected to devices such as encoders, photocells, or PLCs.



Dual-Sensor System

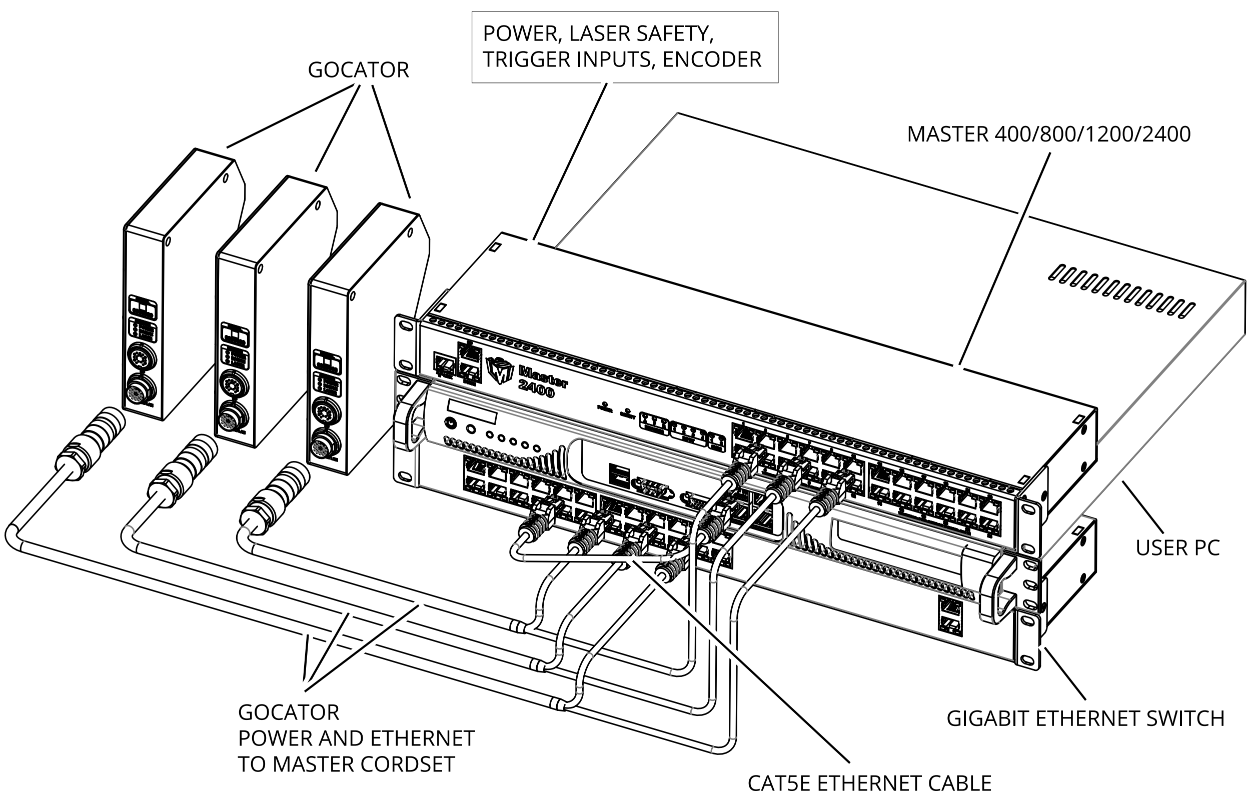
In a dual-sensor system, two Gocator sensors work together to perform ranging and output the combined results. The controlling sensor is referred to as the *Main* sensor, and the other sensor is referred to as the *Buddy* sensor. Gocator's software recognizes three installation orientations: *Opposite*, *Wide*, and *Reverse*.

A Master 400/800/1200/2400 must be used to connect two sensors in a dual-sensor system. Gocator Master cordsets are used to connect sensors to the Master.



Multi-Sensor System

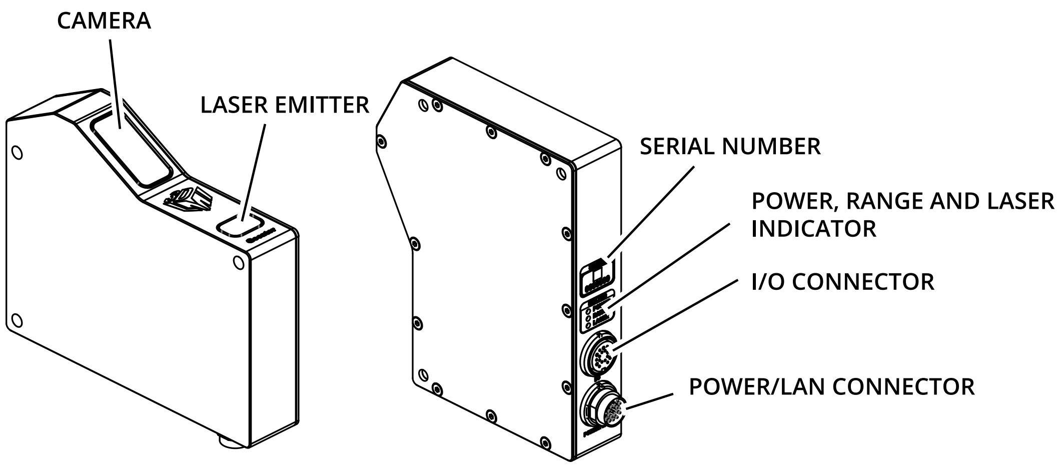
Master 400/800/1200/2400 networking hardware can be used to connect two or more sensors into a multi-sensor system. Gocator Master cordsets are used to connect the sensors to a Master. The Master provides a single point of connection for power, safety, encoder, and digital inputs. A Master 400/800/1200/2400 can be used to ensure that the scan timing is precisely synchronized across sensors. Sensors and client computers communicate via an Ethernet switch (1 Gigabit/s recommended).



Hardware Overview

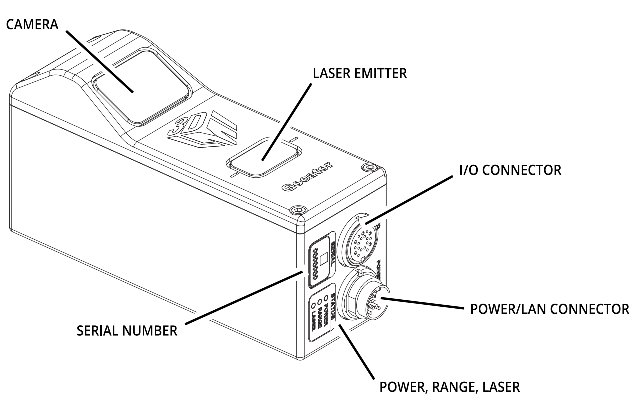
The following sections describe Gocator and its associated hardware.

Side Mount Package



Item	Description
Camera	Observes laser light reflected from target surfaces.
Laser Emitter	Emits structured light for laser ranging.
I/O Connector	Accepts input and output signals.
Power / LAN Connector	Accepts power and laser safety signals and connects to 1000 Mbit/s Ethernet network.
Power Indicator	Illuminates when power is applied (blue).
Range Indicator	Illuminates when camera detects laser light and is within the target range (green).
Laser Indicator	Illuminates when laser safety input is active (amber).
Serial Number	Unique sensor serial number.

Top Mount Package



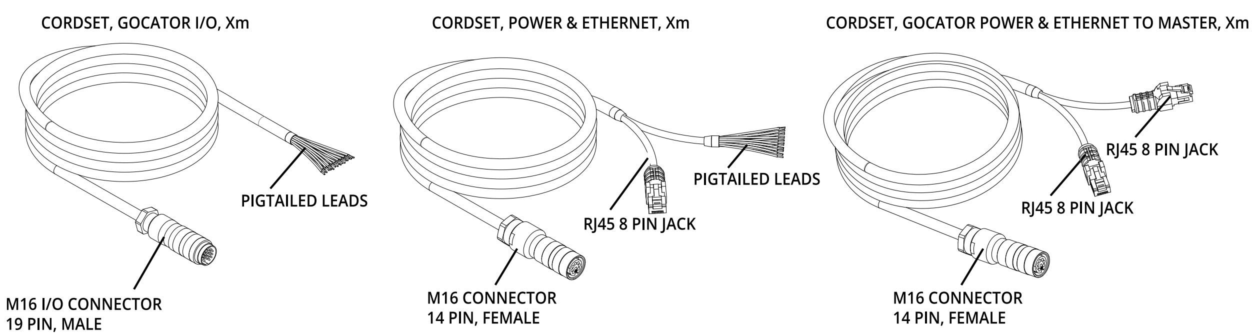
Item	Description
Camera	Observes laser light reflected from target surfaces.
Laser Emitter	Emits structured light for laser ranging.
I/O Connector	Accepts input and output signals.
Power / LAN Connector	Accepts power and laser safety signals and connects to 1000 Mbit/s Ethernet network.
Power Indicator	Illuminates when power is applied (blue).
Range Indicator	Illuminates when camera detects laser light and is within the target range (green).
Laser Indicator	Illuminates when laser safety input is active (amber).
Serial Number	Unique sensor serial number.

Gocator Cordsets

Gocator 1300 sensors use two types of cordsets.

The Power & Ethernet cordset is used for sensor communication via 1000 Mbit/s Ethernet over a standard RJ45 connector. The Master version of the Power & Ethernet cordset provides electrical connection between the sensor and a Master 400/800/1200/2400.

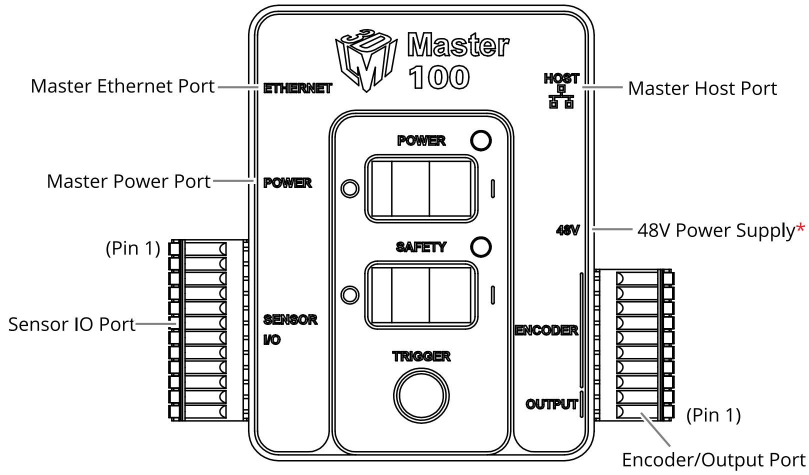
The Gocator I/O cordset provides power and laser safety interlock to sensors. It also provides digital I/O connections, an encoder interface, RS-485 serial connection, and an analog output.



See *Parts and Accessories* (page 248) for cordset lengths and part numbers. Contact LMI for information on creating cordsets with customized lengths and connector orientations.

Master 100

The Master 100 is used by the Gocator 1300 series for standalone system setup.

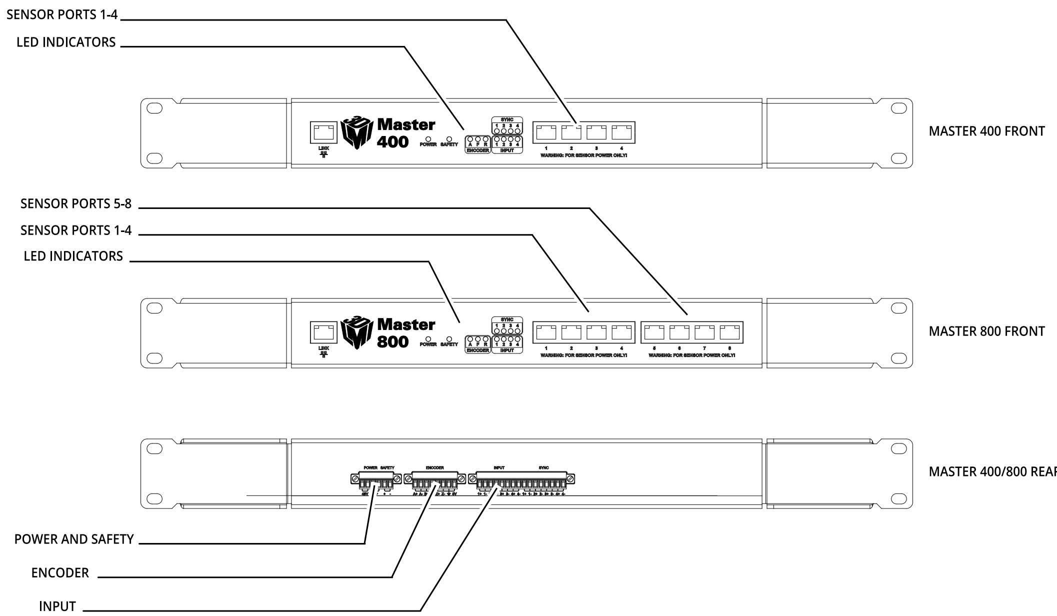


Item	Description
Master Ethernet Port	Connects to the RJ45 connector labeled Ethernet on the Power/LAN to Master cordset.
Master Power Port	Connects to the RJ45 connector labeled Power/Sync on the Power/LAN to Master cordset. Provides power and laser safety to the Gocator.
Sensor I/O Port	Connects to the Gocator I/O cordset.
Master Host Port	Connects to the host PC's Ethernet port.
Power	Accepts power (+48 V).
Power Switch	Toggles sensor power.
Laser Safety Switch	Toggles laser safety signal provided to the sensors [O= laser off, I= laser on].
Trigger	Signals a digital input trigger to the Gocator.
Encoder	Accepts encoder A, B and Z signals.
Digital Output	Provides digital output.

See *Master 100* (page 240) for pinout details.

Master 400/800

The Master 400 and the Master 800 allow you to connect more than two sensors. The Master 400 accepts four sensors, and the Master 800 accepts eight sensors.

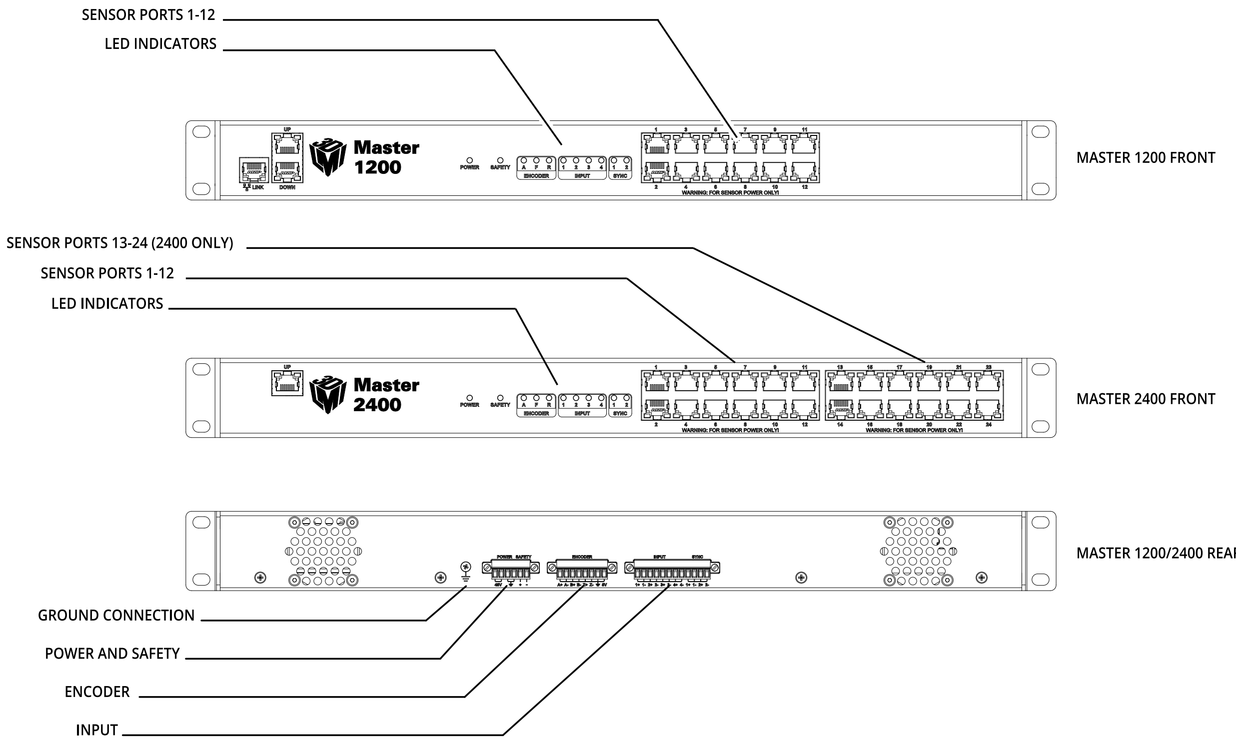


Item	Description
Sensor Ports	Master connection for Gocator sensors (no specific order required).
Ground Connection	Earth ground connection point.
Laser Safety	Laser safety connection.
Encoder	Accepts encoder signal.
Input	Accepts digital input.

See *Master 400/800* (page 242) for pinout details.

Master 1200/2400

The Master 1200 and the Master 2400 allow you to connect more than two sensors. The Master 1200 accepts twelve sensors, and the Master 2400 accepts twenty-four sensors.

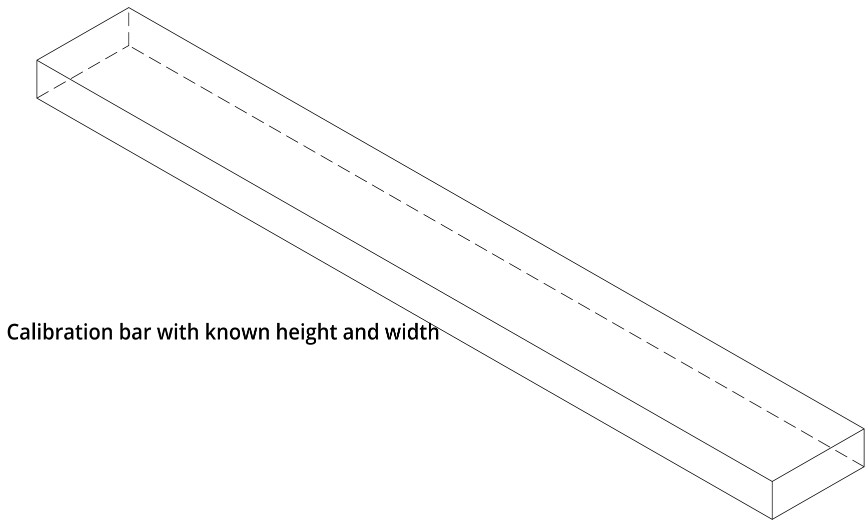


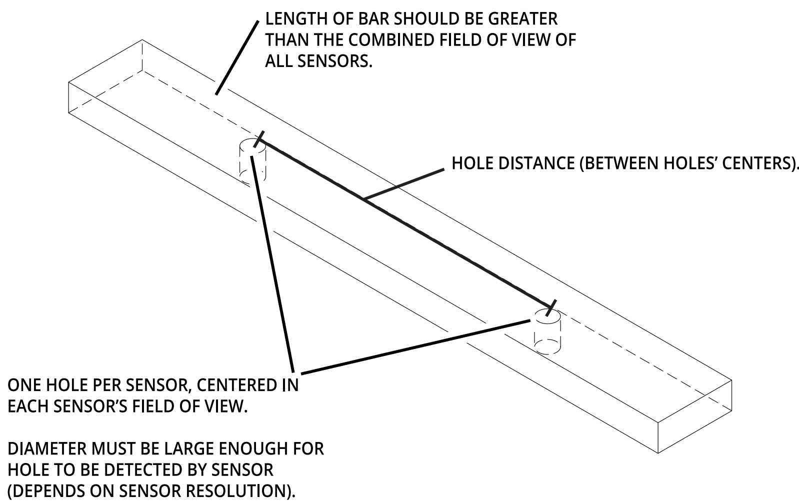
Item	Description
Sensor Ports	Master connection for Gocator sensors (no specific order required).
Ground Connection	Earth ground connection point.
Laser Safety	Laser safety connection.
Encoder	Accepts encoder signal.
Input	Accepts digital input.

See *Master 1200/2400* (page 245) for pinout details.

Calibration Targets

Targets are used for calibrating encoders.





See *Aligning Sensors* (page 77) for more information on alignment.

Installation

The following sections provide grounding, mounting, and orientation information.

Grounding - Gocator

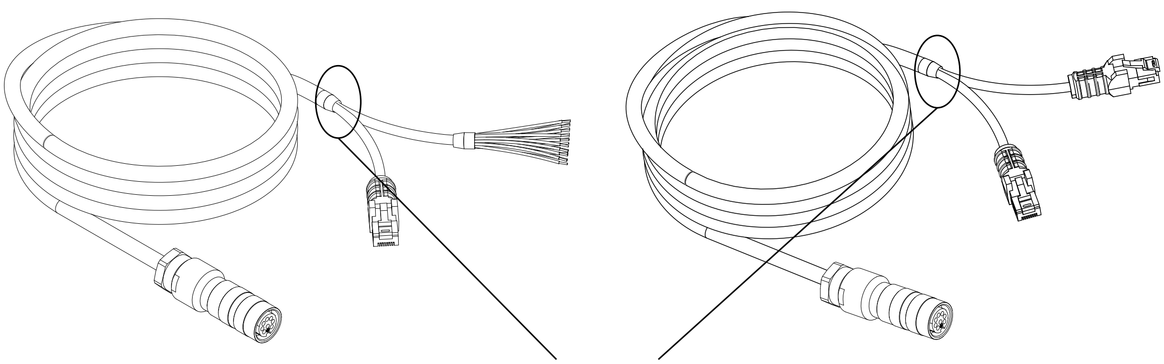
Gocators should be grounded to the earth/chassis through their housings and through the grounding shield of the Power I/O cordset. Gocator sensors have been designed to provide adequate grounding through the use of M6 x 1.0 pitch mounting screws. Always check grounding with a multi-meter to ensure electrical continuity between the mounting frame and the Gocator's connectors.

	The frame or electrical cabinet that the Gocator is mounted to must be connected to earth ground.

Recommended Grounding Practices - Cordsets

If you need to minimize interference with other equipment, you can ground the Power & Ethernet or the Power & Ethernet to Master cordset (depending on which cordset you are using) by terminating the shield of the cordset before the split. The most effective grounding method is to use a 360-degree clamp.

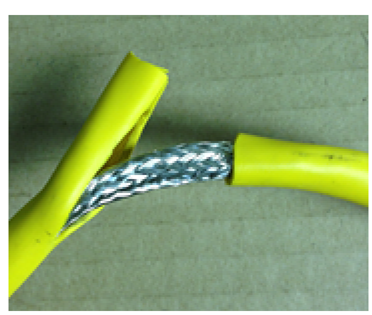
CORDSET, POWER & ETHERNET, Xm **CORDSET, GOCATOR POWER & ETHERNET TO MASTER, Xm**



Attach the 360-degree clamp before the split

To terminate the cordset's shield:

1. Expose the cordset's braided shield by cutting the plastic jacket before the point where the cordset splits.



2. Install a 360-degree ground clamp.

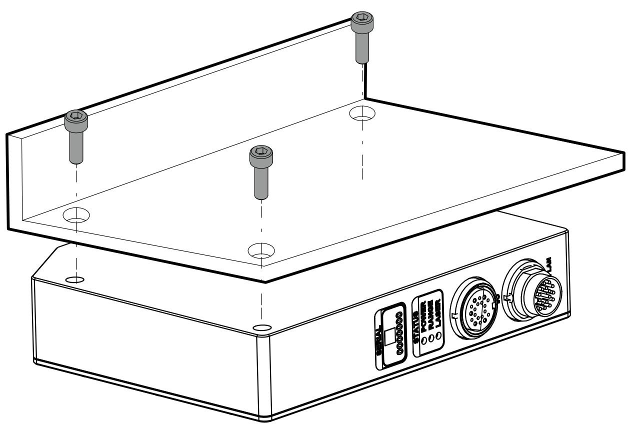
Grounding - Master 400/800/1200/2400

The mounting brackets of all Masters have been designed to provide adequate grounding through the use of star washers. Always check grounding with a multi-meter by ensuring electrical continuity between the mounting frame and RJ45 connectors on the front.

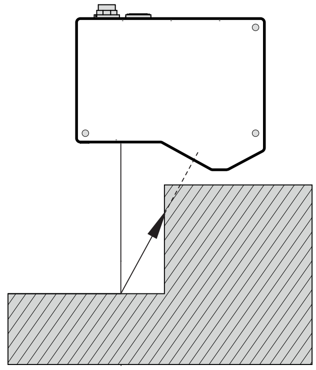
🗋	The frame or electrical cabinet that the Master is mounted to must be connected to earth ground.

Mounting - Side Mount Package

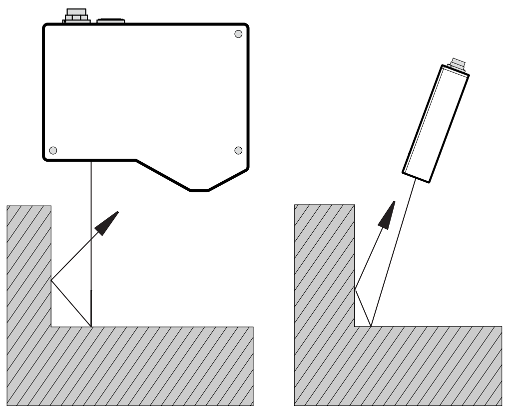
Sensors should be mounted using four M6 x 1.0 pitch screws of suitable length. The recommended thread engagement into the housing is 8-10 mm. Proper care should be taken in order to ensure that the internal threads are not damaged from cross-threading or improper insertion of screws.



Sensors should not be installed near objects that might occlude a camera's view of the laser.



Sensors should not be installed near surfaces that might create unanticipated laser reflections.

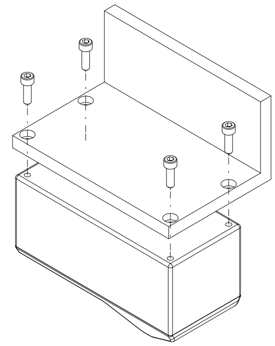


⚠	The sensor must be heat sunk through the frame it is mounted to. When a sensor is properly heat sunk, the difference between ambient temperature and the temperature reported in the sensor's health channel is less than 15° C.

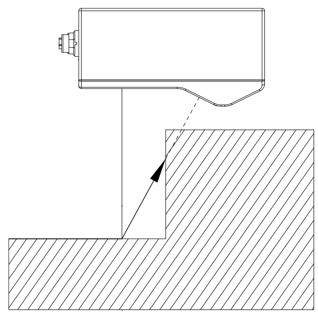
⚠	Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

Mounting - Top Mount Package

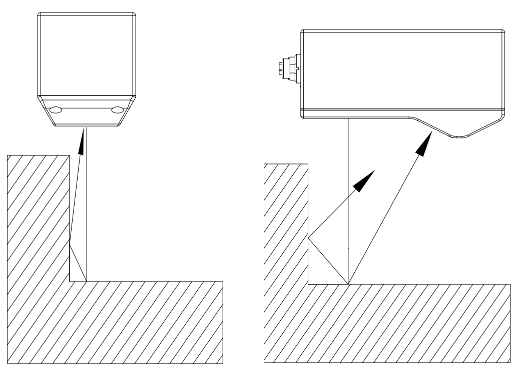
Sensors should be mounted using four M5 x 0.8 pitch screws of suitable length. The recommended thread engagement into the housing is 8-10 mm. Proper care should be taken in order to ensure that the internal threads are not damaged from cross-threading or improper insertion of screws.



Sensors should not be installed near objects that might occlude a camera's view of the laser.



Sensors should not be installed near surfaces that might create unanticipated laser reflections.



⚠	The sensor must be heat sunk through the frame it is mounted to. When a sensor is properly heat sunk, the difference between ambient temperature and the temperature reported in the sensor's health channel is less than 15° C.

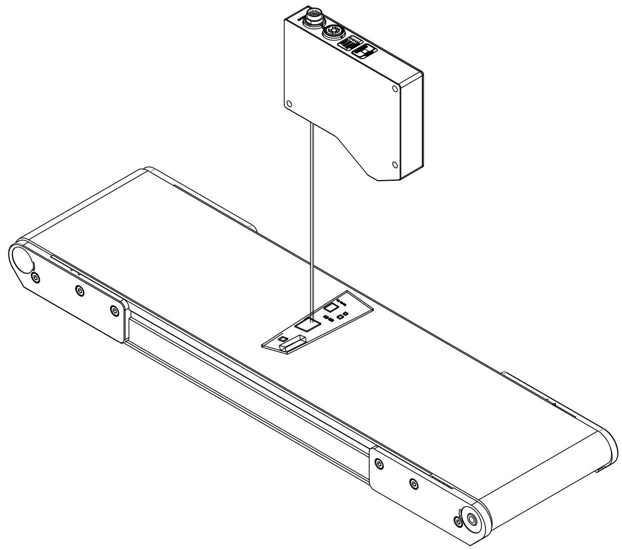
⚠	Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

Orientations

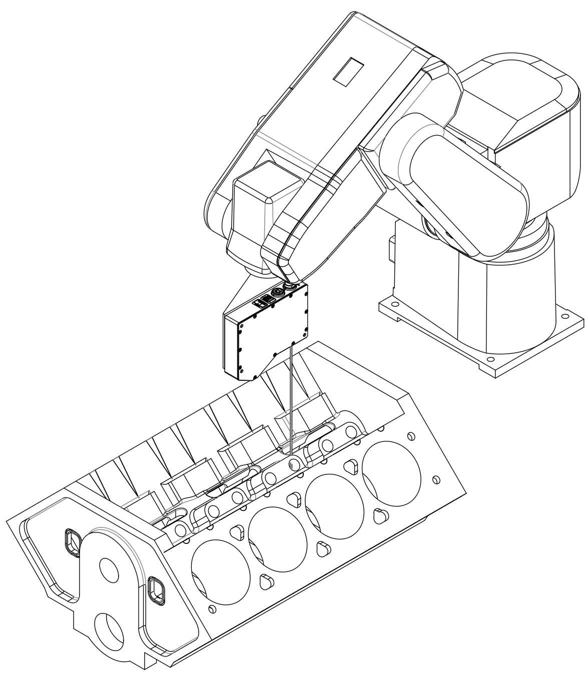
The examples below illustrate the possible mounting orientations for standalone and dual-sensor systems.

See *Dual-Sensor System Layout* (page 50) for more information on orientations.

Standalone Orientations

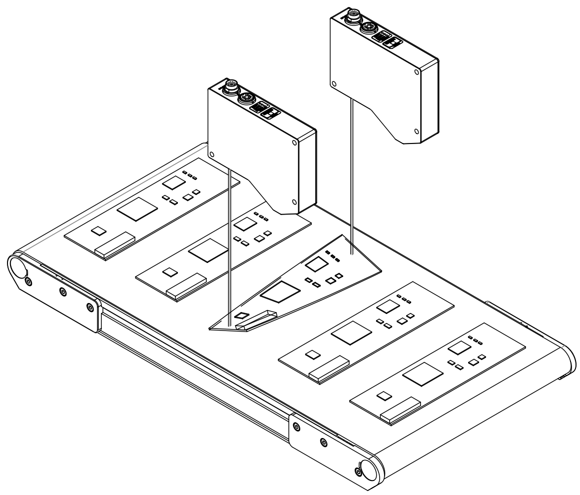


Single sensor above conveyor

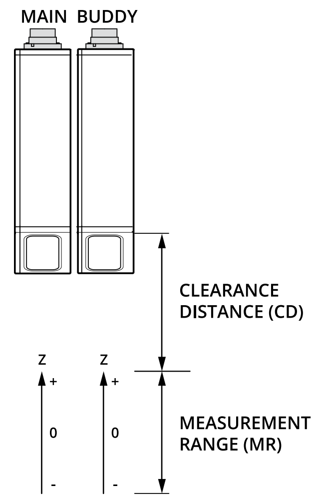


Single sensor on robot arm

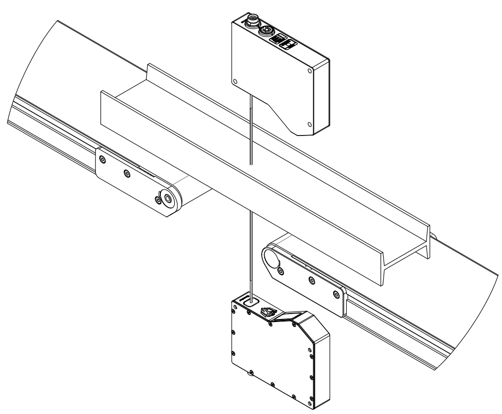
Dual-Sensor System Orientations:



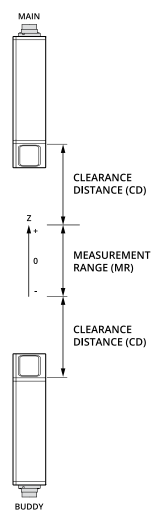
Side-by-side for wide-area measurement (Wide)



Main must be on the left side (when looking into the connector) of the Buddy (Wide)



Above/below for two-sided measurement (Opposite)



Main must be on the top with Buddy at the bottom (Opposite)

Network Setup

The following sections provide procedures for client PC and Gocator network setup.

Client Setup

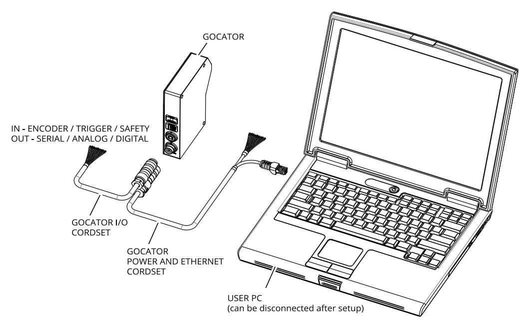
Sensors are shipped with the following default network configuration:

Setting	Default
DHCP	Disabled
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0

> All Gocator sensors are configured to 192.168.1.10 as the default IP address. For a dual-sensor system, the Main and Buddy sensors must be assigned unique addresses before they can be used on the same network. Before proceeding, connect the Main and Buddy sensors one at a time (to avoid an address conflict) and use the steps in *Running a Dual-Sensor System on page 32* to assign each sensor a unique address.

To connect to a sensor for the first time:

1. Connect cables and apply power.
   Sensor cabling is illustrated in *System Overview on page 15*.



2. Change the client PC's network settings.

   Windows 7

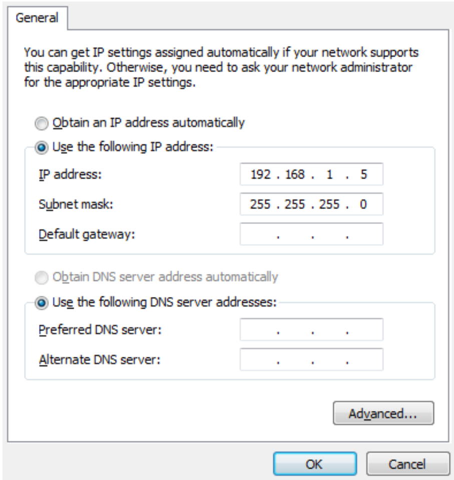
   a. Open the Control Panel, select **Network and Sharing Center**, and then click **Change Adapter Settings**.

   b. Right-click the network connection you want to modify, and then click **Properties**.

   c. On the **Networking** tab, click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.

   d. Select the **Use the following IP address** option.

   e. Enter IP Address "192.168.1.5" and Subnet Mask "255.255.255.0", then click **OK**.

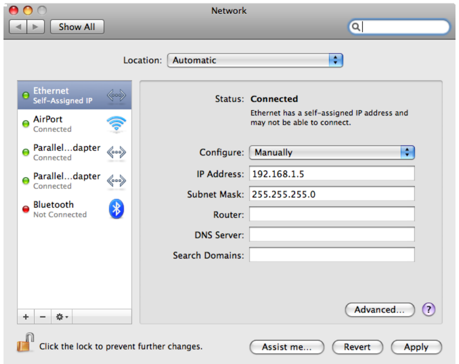


   Mac OS X v10.6

   a. Open the Network pane in **System Preferences** and select **Ethernet**.

   b. Set **Configure** to **Manually**.

   c. Enter IP Address "192.168.1.5" and Subnet Mask "255.255.255.0", then click **Apply**.



⚠	See *Troubleshooting* (page 211) if you experience any problems while attempting to establish a connection to the sensor.

Gocator Setup

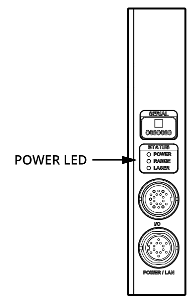
The Gocator is shipped with a default configuration that will produce laser ranges on most targets.

The following sections walk you through the steps required to set up a standalone sensor system and a dual-sensor system for operations. After you have completed the setup, you can perform laser ranging to verify basic sensor operation.

Running a Standalone Sensor System

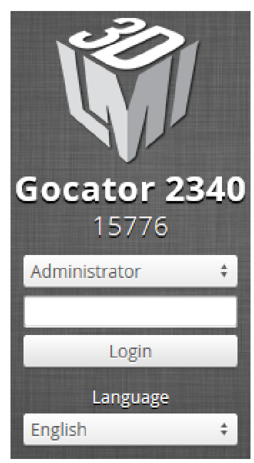
To configure a standalone sensor system:

1. Power up the sensor.
   The power indicator (blue) should turn on immediately.



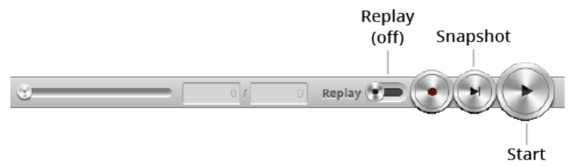
2. Enter the sensor's IP address (192.168.1.10) in a web browser.

3. Log in as Administrator with no password.
   The interface display language can be changed using the language option. After selecting the language, the browser will refresh and the web interface will display in the selected language.

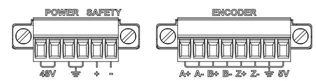


4. Go to the **Manage** page.

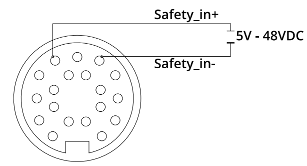
5. Ensure that **Replay** mode is off (the slider is set to the left).



6. Ensure that the Laser Safety Switch is enabled or the Laser Safety input is high.
7. Go to the **Scan** page.
8. Press the **Start** button or the **Snapshot** on the **Toolbar** to start the sensor.
   The **Start** button is used to run sensors continuously, whereas the **Snapshot** button is used to trigger a single capture.



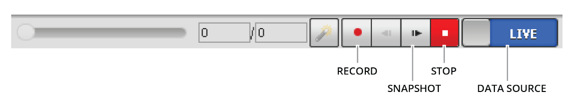
Master 400/800/1200/2400



Standalone

9. Move a target into the laser plane.
   If a target object is within the sensor's measurement range, the data viewer will display the distance to the target, and the sensor's range indicator will illuminate.
   If you cannot see the laser, or if a range is not displayed in the Data Viewer, see *Troubleshooting* (page 211).

10. Press the **Stop** button.
    The laser should turn off.

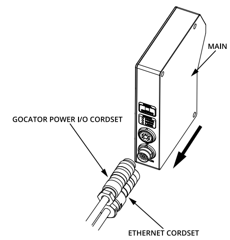


Running a Dual-Sensor System

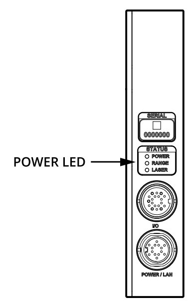
All sensors are shipped with a default IP address of 192.168.1.10. Ethernet networks require a unique IP address for each device, so you must set up a unique address for each sensor.

To configure a dual-sensor system:

1. Turn off the sensors and unplug the Ethernet network connection of the Main sensor.
   All sensors are shipped with a default IP address of 192.168.1.10. Ethernet networks require a unique IP address for each device. Skip step 1 to 3 if the Buddy sensor's IP address is already set up with an unique address.

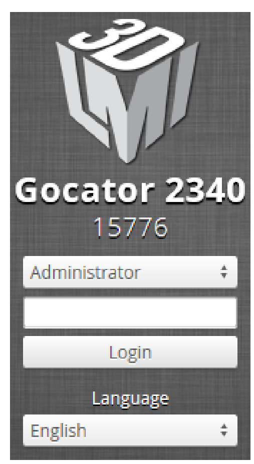


2. Power up the Buddy sensor.
   The power LED (blue) of the Buddy sensor should turn on immediately.



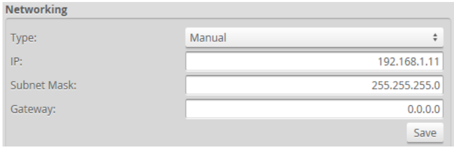
3. Enter the sensor's IP address 192.168.1.10 in a web browser.
   This will log into the Buddy sensor.

4. Log in as Administrator with no password.

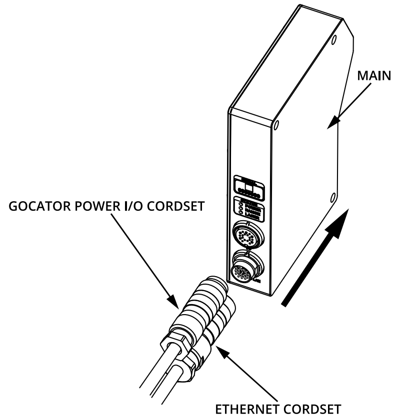


5. Go to the **Manage** Page.

6. Modify the IP address to 192.168.1.11 in the **Networking** category and click the **Save** button.
   When you click the **Save** button, you will be prompted to confirm your selection.

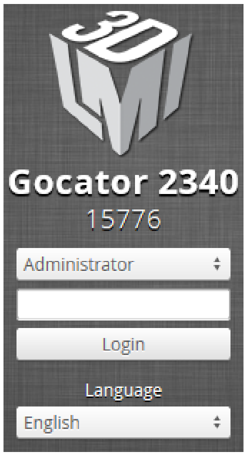


7. Turn off the sensors, re-connect the Main sensor's Ethernet connection and power-cycle the sensors.
   After changing network configuration, the sensors must be reset or power-cycled before the change will take effect.



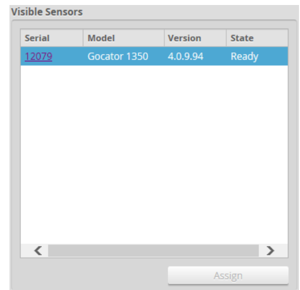
8. Enter the sensor's IP address 192.168.1.10 in a web browser.
   This will log into the Main sensor.

9. Log in as Administrator with no password.
   The interface display language can be changed using the language option. After selecting the language, the browser will refresh and the web interface will display in the selected language.

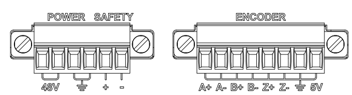


10. Select the **Manage** page.

11. Go to **Manage** page, **Sensor System** panel, and select the **Visible Sensors** panel.
    The serial number of the Buddy sensor is listed in the Available Sensors panel.

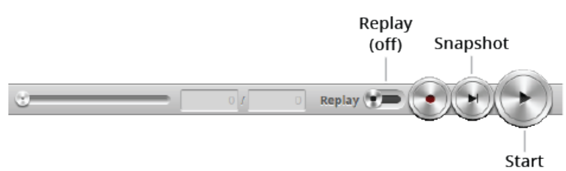


12. Select the Buddy sensor and click the **Assign** button.
    The Buddy sensor will be assigned to the Main sensor and its status will be updated in the System panel.
    The firmware on Main and Buddy sensors must be the same for Buddy assignment to be successful. If the firmware is different, connect the Main and Buddy sensor one at a time and follow the steps in *Firmware Upgrade on page 61* to upgrade the sensors.
13. Ensure that the Laser Safety Switch is enabled or the Laser Safety input is high.

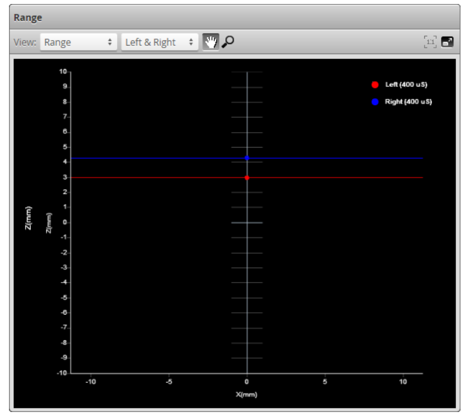


Master 400/800/1200/2400

14. Ensure that **Replay** mode is off (the slider is set to the left).
15. Go to the the **Scan** page.
16. Press the **Start** or the **Snapshot** button on the **Toolbar**to start the sensors.
    The **Start** button is used to run sensors continuously, while the **Snapshot** button is used to trigger a single measurement.



17. Move a target into the laser plane.
    If a target object is within the sensor's measurement range, the data viewer will display the distance to the target, and the sensor's range indicator will illuminate.
    If you cannot see the laser, or if a range is not displayed in the Data Viewer, see *Troubleshooting* (page 211).



18. Press the **Stop** button if you used the **Start** button to start the sensors.
    The laser should turn off.

Next Steps

After you complete the steps in this section, the Gocator measurement system is ready to be configured for an application using the software interface. The interface is explained in the following sections:

System Management and Maintenance (page 49)
Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance.

Scan Setup and Alignment (page 63)
Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment.

Measurement (page 86)
Contains built-in measurement tools and their settings.

Output (page 102)
Contains settings for configuring output protocols used to communicate measurements to external devices.

Dashboard (page 114)
Provides monitoring of measurement statistics and sensor health.

Toolbar (page 42)
Controls sensor operation, manages jobs, and replays recorded measurement data.

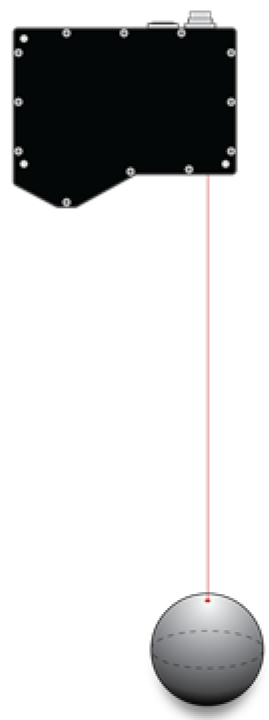
Theory of Operation

The following sections describe the theory of operation of Gocator sensors.

3D Acquisition

Principle of 3D Acquisition

The Gocator 1300 series sensors are displacement sensors, meaning that they capture a single 3D point for each camera exposure. The sensor projects a laser point onto the target. The sensor's camera views the laser from an angle, and captures the reflection of the light off the target. Because of this triangulation angle, the laser point appears in different positions on the camera depending on the 3D shape of the target. Gocator sensors are always pre-calibrated to deliver 3D data in engineering units throughout the specified measurement range.



Resolution and Accuracy

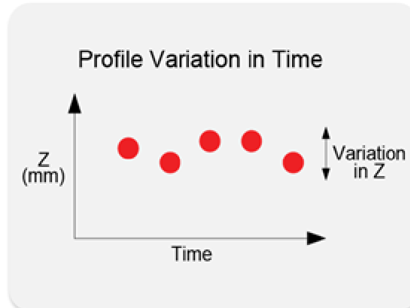
Delete this text and replace it with your own content.

Z Resolution

Z resolution is the variability of the height measurement with the target at a fixed position. This variability is caused by camera imager and sensor electronics.

Z resolution is better at the close range and worse at the far range. This is reflected in the Gocator data sheet as the two numbers quoted for Z resolution.

Z Resolution gives an indication of the smallest detectable height difference.

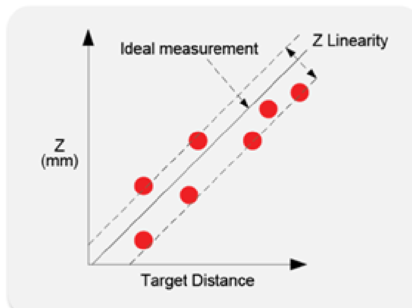


Z Linearity

Z Linearity is the difference between the actual distance to the target and the measured distance to the target, throughout the measurement range.

Z Linearity is expressed in the Gocator data sheet as a percentage of the total measurement range.

Z Linearity gives an indication of the sensor's ability to measure absolute distance



Range Output

Gocator measures the height of the object calculated from laser triangulation. The measurement is referred to as ranges and is reported as the distance from the sensor origin.

Coordinate Systems

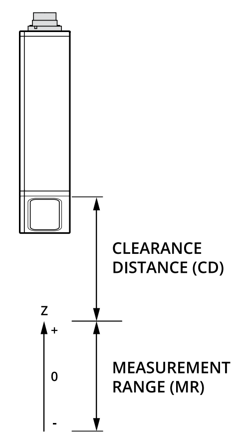
Range data is reported in sensor or system coordinates depending on the alignment state. The coordinate systems are described below.

Sensor Coordinates

Before alignment, individual sensors use the coordinate system shown here.

The Z axis represents the sensor's measurement range (MR), with the values increasing towards the sensor.

The origin is at the center of the MR.

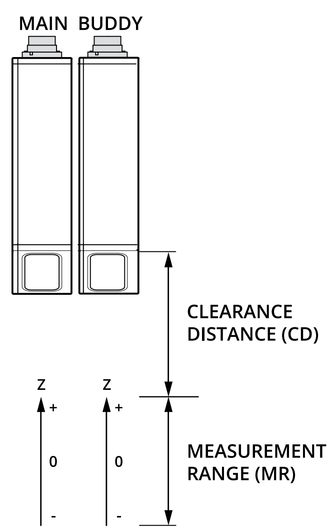


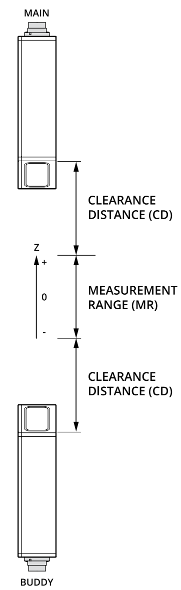
System Coordinates

Alignment is used with a single sensor to compensate for mounting misalignment and to set a zero reference, such as a conveyor belt surface. Alignment is also used to set a common coordinate system for dual-sensor systems. The adjustments resulting from alignment are called transformations. See *Alignment* (page 76) for more information on alignment.

System coordinates are aligned so that the system Z origin is set to the base of the alignment target object.

For Wide and Opposite layouts, ranges and measurements from the Main and Buddy sensors are expressed in a unified coordinate system. Isolated layouts express results using a separate coordinate system for each sensor.



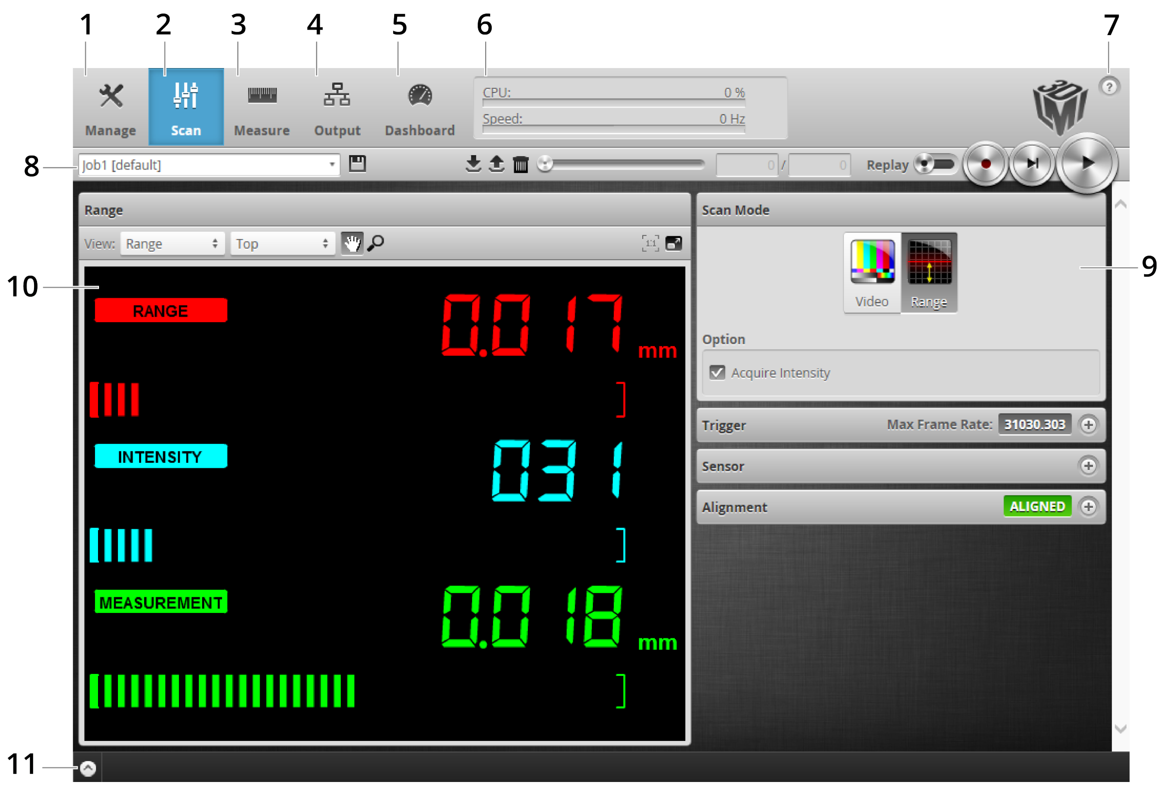

MAIN
CLEARANCE
DISTANCE (CD)
Z
+
0
-
MEASUREMENT
RANGE (MR)
CLEARANCE
DISTANCE (CD)
BUDDY

Gocator Web Interface

The following sections describe the Gocator web interface.

User Interface Overview

Gocator sensors are configured by connecting to a *Main* sensor with a web browser. The Gocator web interface is illustrated below.



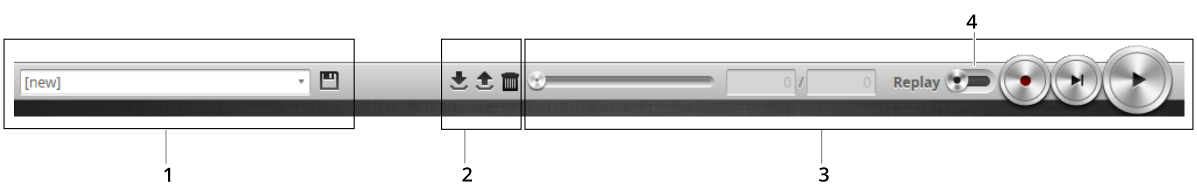
	Element	Description
1	Manage page	Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance. See *System Management and Maintenance* (page 49).
2	Scan page	Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment. See *Scan Setup and Alignment* (page 63).
3	Measure page	Contains built-in measurement tools and their settings. See *Measurement* (page 86).

	Element	Description
4	Output page	Contains settings for configuring output protocols used to communicate measurements to external devices. See *Output* (page 102).
5	Dashboard page	Provides monitoring of measurement statistics and sensor health. See *Dashboard* (page 114).
6	CPU Load and Speed	Provides important sensor performance metrics. See *Metrics Area* (page 47).
7	Help	Provides links to the user manual and SDK.
8	Toolbar	Controls sensor operation, manages jobs, and replays recorded measurement data. See *Toolbar* (below).
9	Configuration area	Provides controls to configure scan and measurement tool settings.
10	Data viewer	Displays sensor data, tool setup controls, and measurements. See *Data Viewer* on page 81 for its use when the **Scan** page is active and on page 86 for its use when the **Measure** page is active.
11	Log	Displays messages from the sensor (errors, warnings, and other information). See *Log* (page 47).

Common Elements

Toolbar

The toolbar is used for performing common operations. This section explains how to use the toolbar to manage jobs and to operate the sensor.



	Element	Description
1	Job controls	For saving and loading different jobs.
2	Recorded data controls	For downloading, uploading, and exporting recorded data.
3	Sensor operation / replay control	Use the sensor operation controls to start sensors, enable recording, and control recorded data.
4	Replay switch	Toggles the sensor data source between live and replay.

Saving and Loading Settings

When you change sensor settings using the Gocator web interface, some changes are saved automatically, while other changes are temporary until you save them manually. The following table lists the types of information that can be saved in a sensor.

Setting Type	Behavior
Network Address	Network address changes are saved when you click the **Save** button in **Networking** on

Setting Type	Behavior
	the **Manage** page. The sensor must be reset before changes take effect.
Job	Most of the settings that can be changed in the Gocator's web interface, such as the ones in the **Manage**, **Measure**, and **Output** pages, are temporary until saved in a job file. Each sensor can have multiple job files. If there is a job file that is designated as the default, it will be loaded automatically when the sensor is reset.
Alignment	Alignment can either be fixed or dynamic, as controlled by the **Alignment Reference** setting in **Motion and Alignment** in the **Manage** page. Alignment is saved automatically at the end of the alignment procedure when **Alignment Reference** is set to **Fixed**. When **Alignment Reference** is set to **Dynamic**, however, you must manually save the job to save alignment.

The job drop-down list shows the list of jobs stored in the sensor. The job that is currently active is listed at the top. The job name will be marked with "[unsaved]" to indicate any unsaved changes.

To save a job:

1. Select a job in the job drop-down list.
   - If you are creating a new job, choose **[New]** in the job drop-down list and enter a name for the job.
   - If you are saving changes to an existing job, choose the job in the job drop-down list.
2. Press the **Enter** key or click the **Save** button.
   The job will be saved to sensor storage using the name you provided. Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

To activate an existing job:

1. Select an existing file name in the job drop-down list.
   The job will be activated from sensor storage. If there are any unsaved changes to the current job, you will be asked whether you want to discard those changes.

Detailed management of jobs is handled in the **Jobs** panel in the **Manage** page. See *Jobs* (page 56) for more information.

Managing Multiple Settings

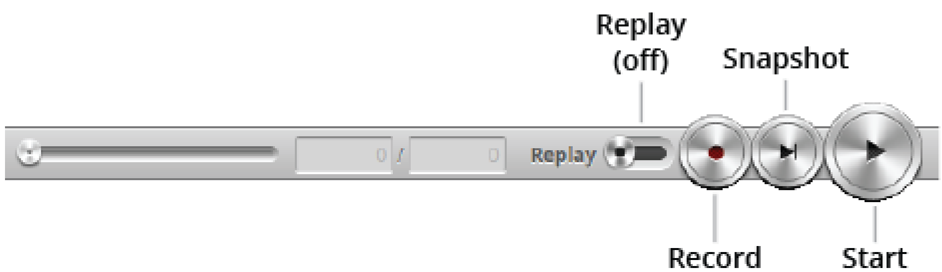
A Gocator can store several hundred jobs. Being able to switch between different jobs is useful when a Gocator is used with different constraints during separate production runs (for example, width decision constraints might be loose during one production run and tight during another depending on the desired grade of the part).

Switching active jobs can be done manually through the web interface as described under *To activate an existing job* in *Saving and Loading Settings on page 42*. Switching active jobs can also be done programmatically using the supported industrial protocols (Modbus, EtherNet/IP, and ASCII), the Gocator's native Ethernet protocol, and through the SDK.

Recording, Playback, and Measurement Simulation

Gocator sensors can record and replay data, and can also simulate measurement tools on recorded data. This feature is most often used for troubleshooting and fine-tuning measurements, but can also be helpful during setup.

Recording and playback are controlled by using the toolbar controls.

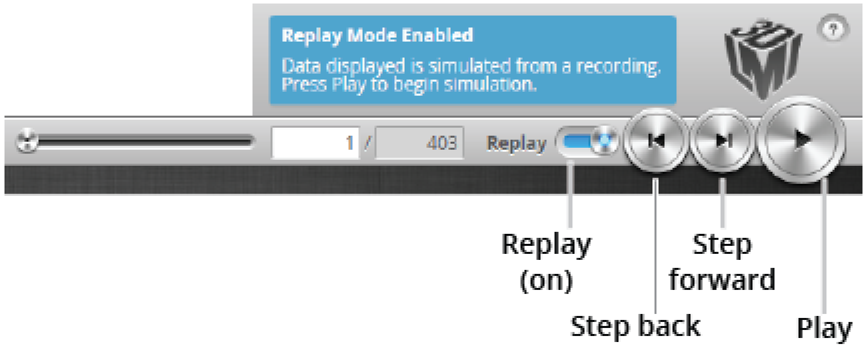


Recording and playback controls when replay is off

To record live data:

1. Toggle **Replay** mode off by setting the slider to the left in the **Toolbar**.
2. Press the **Record** button to enable recording.
   When replay is off and recording is enabled, the sensor will store the most recent data as it runs. Remember to disable recording if you no longer wish to record live data (press the **Record** button again to disable recording).
3. Press the **Snapshot** button or **Start** button.
   The **Snapshot** records a single frame. The **Start** button will run the sensor continuously and all frames will be recorded, up to available memory. When the memory limit is reached, the oldest data will be discarded.

🗋	Newly recorded data is appended to existing replay data unless the sensor job has been modified.



Recording and playback controls when replay is on

To replay recorded data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background will turn blue and a Replay Mode Enabled message will be displayed.

2. Use the **Replay** slider or the **Step Forward**, **Step Back**, or **Play** buttons to review data.
   The **Step Forward** and **Step Back** buttons move and the current replay location backward and forward by a single frame, respectively.
   The **Play** button advances the replay location continuously, animating the playback.
   The **Stop** button (replaces the **Play** button while playing) can be used to pause the replay at a particular location.
   The **Replay** slider (or **Replay Position** box) can be used to go to a specific replay frame.

To simulate measurements on recorded data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background will turn blue and a Replay Mode Enabled message will be displayed.

2. Go to the **Measure** page.
   Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired.

3. Use the **Replay Slider**, **Step Forward**, **Step Back**, or **Play** button to simulate measurements.
   Step or play through recorded data to execute the measurement tools on the recording.
   Individual measurement values can be viewed directly in the data viewer. Statistics on the measurements that have been simulated can be viewed in the **Dashboard** page; see *Dashboard* (page 114).

To clear recorded data:

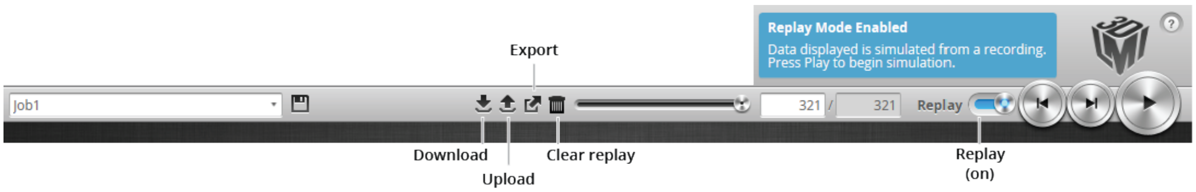
1. Stop the sensor if it is running by clicking on the **Stop** button.

2. Click on the **Clear Replay Data** button.

Downloading, Exporting, and Uploading Recorded Data

Recorded data can be downloaded or exported to the client computer or uploaded to the Gocator. Export is often used for processing the recorded data using third-party tools. Recorded data can also be downloaded in a binary format, which is used to back up the data for reviewing in the future.

	Recorded data is not saved or loaded when you save or activate jobs in the toolbar.



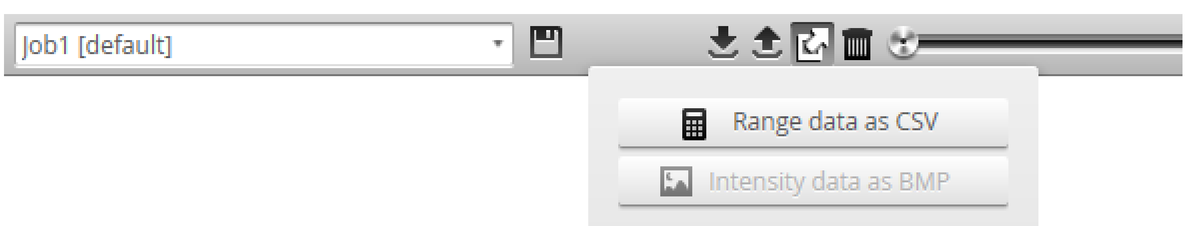
To download recorded data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background will turn blue and a Replay Mode Enabled message will be displayed.
2. Click the **Download** button.

To upload recorded data:

1. Toggle **Replay** mode on by setting the slider to the left in the **Toolbar**.
   The slider's background will turn blue and a Replay Mode Enabled message will be displayed.
2. Click the **Upload** button.
3. Select the directory and the file name to upload from the client computer and click on **OK**.

Recorded data can be exported using the CSV format. If **Acquire Intensity** has been enabled in the **Scan Mode** panel on the **Scan** page, intensity data will be included in the exported CSV file.



To export recorded data to CSV:

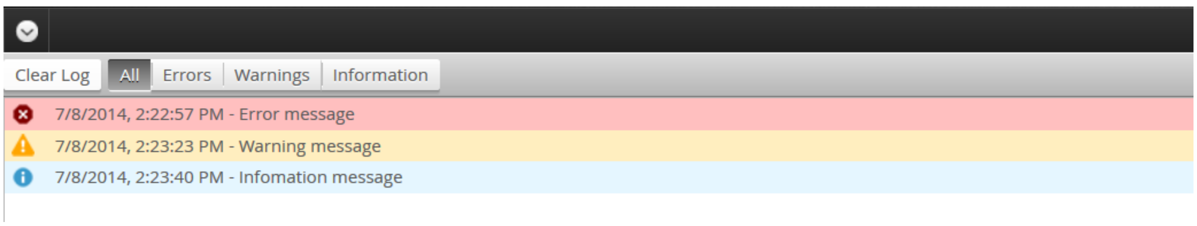
1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background will turn blue and a Replay Mode Enabled message will be displayed.
2. Click the **Export** button and select **Export Range Data as CSV**.

data at the current replay location is exported.

Use the playback control buttons to move to a different replay location; see *To replay recorded data* in *Recording, Playback, and Measurement Simulation on page 44* for more information on playback.

Log

The log, located at the bottom of the web interface, is a centralized location for all messages that the Gocator displays, including warnings and errors.



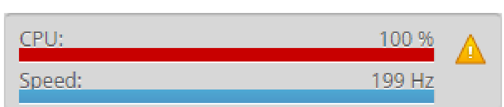
To use the log:

1. Click on the Log open button at the bottom of the web interface.
2. Click on the appropriate tab for the information you need.

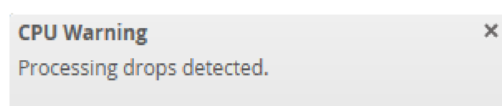
Metrics Area

The **Metrics** area displays two important sensor performance metrics: CPU load and speed (current frame rate).

The **CPU** bar in the **Metrics** panel (at the top of the interface) displays how much of the CPU is being utilized. A warning symbol () will appear next to the **CPU** bar if the sensor drops rangesbecause the CPU is over-loaded.



CPU at 100%



CPU warning message

The **Speed** bar displays the frame rate of the sensor. A warning symbol () will appear next to it if triggers (external input or encoder) are dropped because the external rate exceeds the maximum frame rate.

In both cases, a warning message will be temporarily displayed in the lower right corner of the web interface. Click on the warning symbol () to redisplay the warning message.

Open the log for details on the warning. See *Log* (above) for more information.

Data Viewer

The data viewer is displayed in both the **Scan** and the **Measure** pages, but displays different information depending on which page is active.

When the **Scan** page is active, the data viewer displays sensor data and can be used to adjust regions of interest. Depending on the selected operation mode (page 64), the data viewer can display video images or ranges. For details, see *Data Viewer* (page 81).

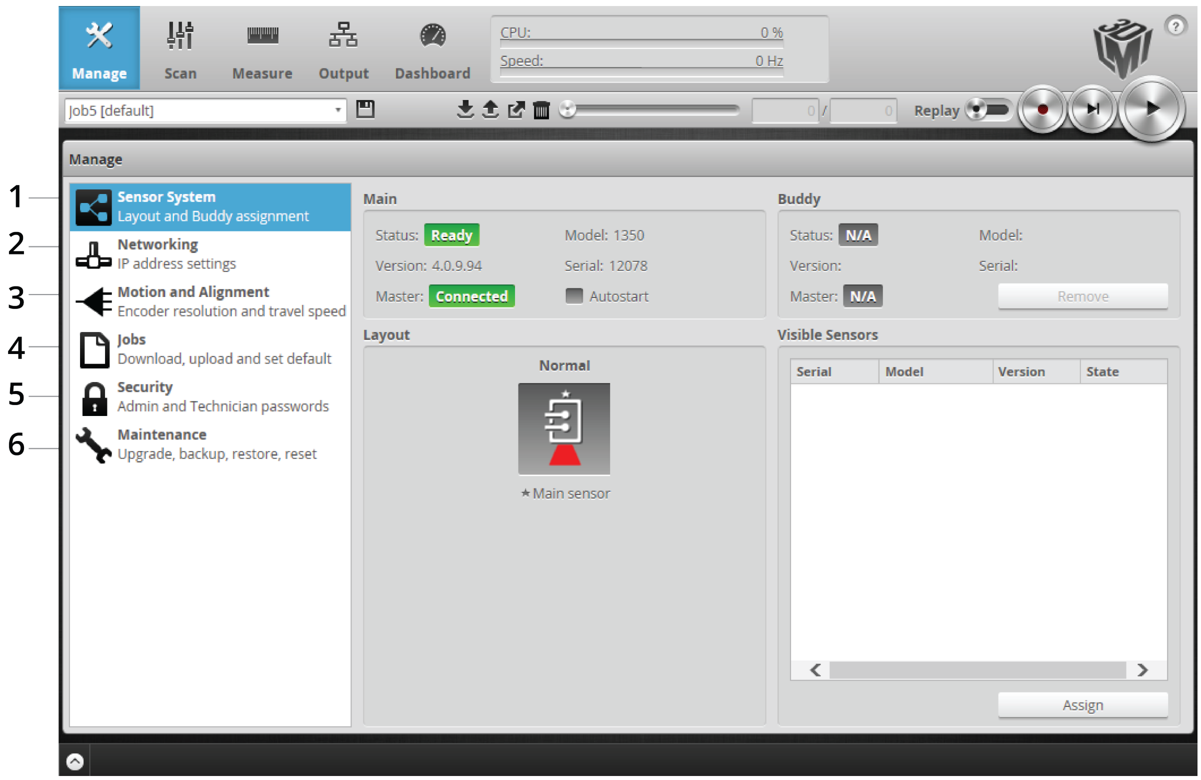
When the **Measure** page is active, the data viewer displays sensor data onto which representations of measurement tools and their measurements are superimposed. For details, see *Data Viewer* (page 86).

System Management and Maintenance

The following sections describe how to set up the sensor connections and networking, how to calibrate encoders and choose alignment reference, and how to perform maintenance tasks.

Manage Page Overview

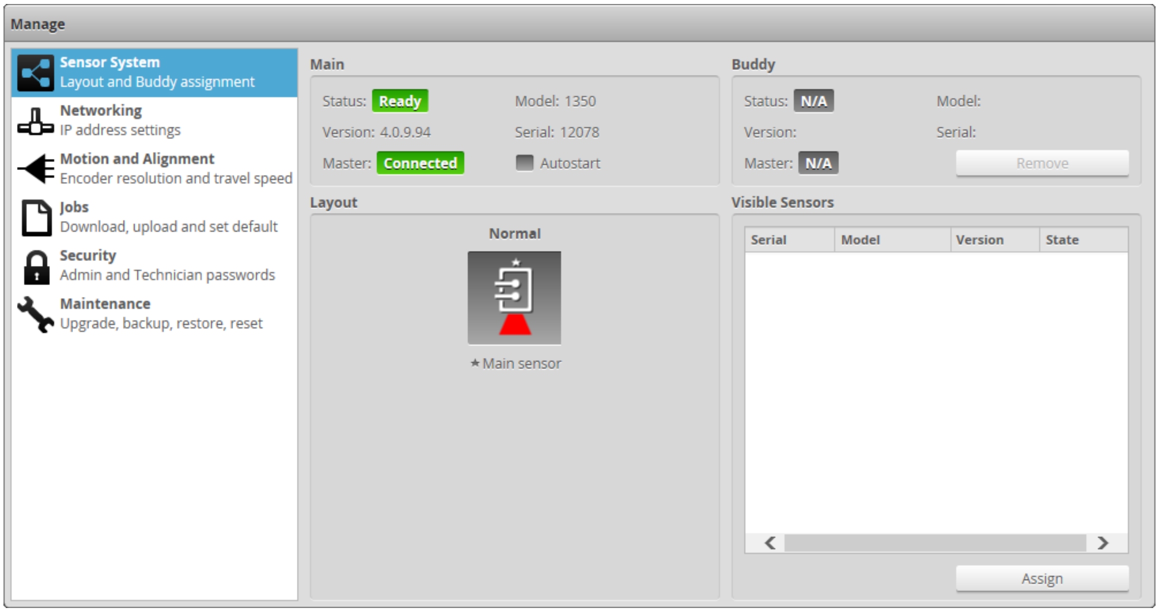
Gocator's system and maintenance tasks are performed on the **Manage** page.



Element	Description
1 Sensor System	Contains settings for configuring sensor system and layout, and boot-up. See *Sensor System* (next page).
2 Networking	Contains settings for configuring the network. See *Networking* (page 53).
3 Motion and Alignment	Contains settings to configure the encoder. See *Motion and Alignment* (page 54).
4 Jobs	Lets you manage jobs stored on the sensor. See *Jobs* (page 56).
5 Security	Lets you change passwords. See *Security* (page 58).
6 Maintenance	Lets you upgrade firmware, create/restore backups, and reset sensors. See *Maintenance* (page 59).

Sensor System

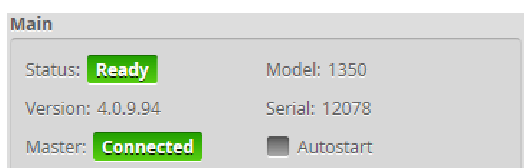
The following sections describe the **Sensor System** category on the **Manage** page. This category lets you choose the layout standalone or dual-sensor systems, and provides other system settings.



Dual-sensor layouts are only displayed when a Buddy sensor has been assigned.

Sensor Autostart

With the **Autostart** setting enabled, laser ranging profiling and measurement functions will begin automatically when the sensor is powered on. Autostart must be enabled if the sensor will be used without being connected to a computer.



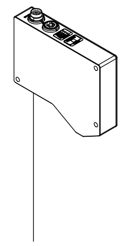
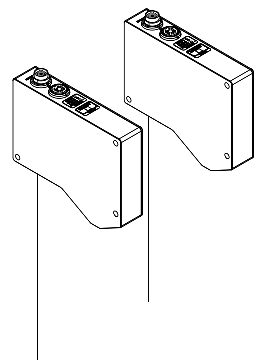
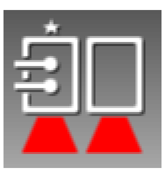
To enable/disable Autostart:

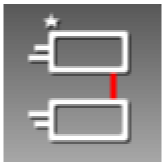
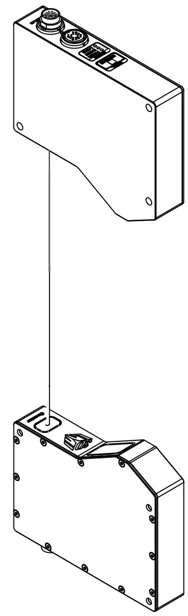
1. Go to the **Manage** page and click on the **Sensor System** category.
2. Check/uncheck the **Autostart** option in the **Main** section.

Dual-Sensor System Layout

Mounting orientations must be specified for a dual-sensor system. This information allows the alignment procedure to determine the correct system-wide coordinates for laser ranging and measurements. See *Coordinate Systems* (page 39) for more information on sensor and system coordinates.

Supported Layouts

	Orientation	Example
	**Standalone** The sensor operates as an isolated device.	
	**Wide** Sensors are mounted in Left (Main) and Right (Buddy) positions for measuring the height of the object at multiple points .	
	**Reverse** Sensors are mounted in a left-right layout as with the Wide layout, but the Buddy sensor is mounted such that it is rotated 180 degrees around the Z axis to prevent occlusion along the Y axis.	

	Orientation	Example
	**Opposite** Sensors are mounted in Top (Main) and Bottom (Buddy) positions for measuring thickness .	

To specify the layout:

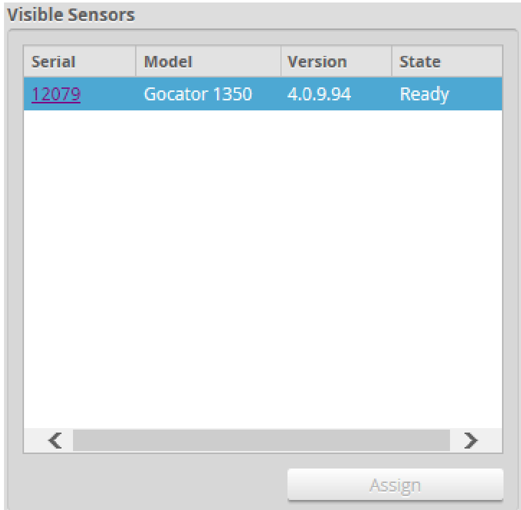
1. Go to the **Manage** page and click on the **Sensor System** category.
2. Select an assigned Buddy sensor in the **Visible Sensors** list.
   See *Buddy Assignment* (below) for information on assigning a Buddy Sensor.
3. Select a layout by clicking on one of the Layout buttons.
   See the table above for information on layouts.

Buddy Assignment

In a dual-sensor system, the *Main* sensor assumes control of the *Buddy* sensor after the Buddy sensor is assigned to the Main sensor. Configuration for both sensors can be performed through the Main sensor's interface.

> Main and Buddy sensors must be assigned unique IP addresses before they can be used on the same network. Before proceeding, connect the Main and Buddy sensors one at a time (to avoid an address conflict) and use the steps outline in Running a Dual-Sensor System (page 30) to assign each sensor a unique address.

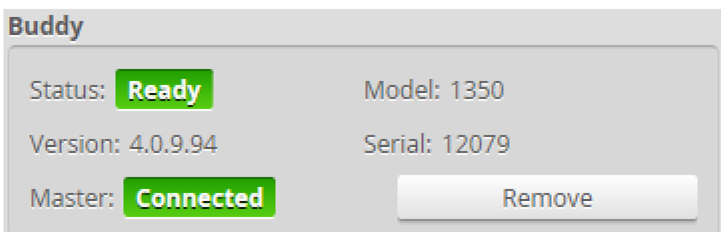
> When a sensor is acting as a Buddy, it is not discoverable and its web interface is not accessible.



To assign a Buddy sensor:

1. Go to the **Manage** page and click on the **Sensor System** category.
2. Select a sensor in the **Visible Sensors** list.
3. Click the **Assign** button.
   A sensor can only be assigned as a Buddy if its firmware and model number match the firmware and model number of the Main sensor. The **Assign** button will be greyed out if a sensor cannot be assigned as a Buddy.

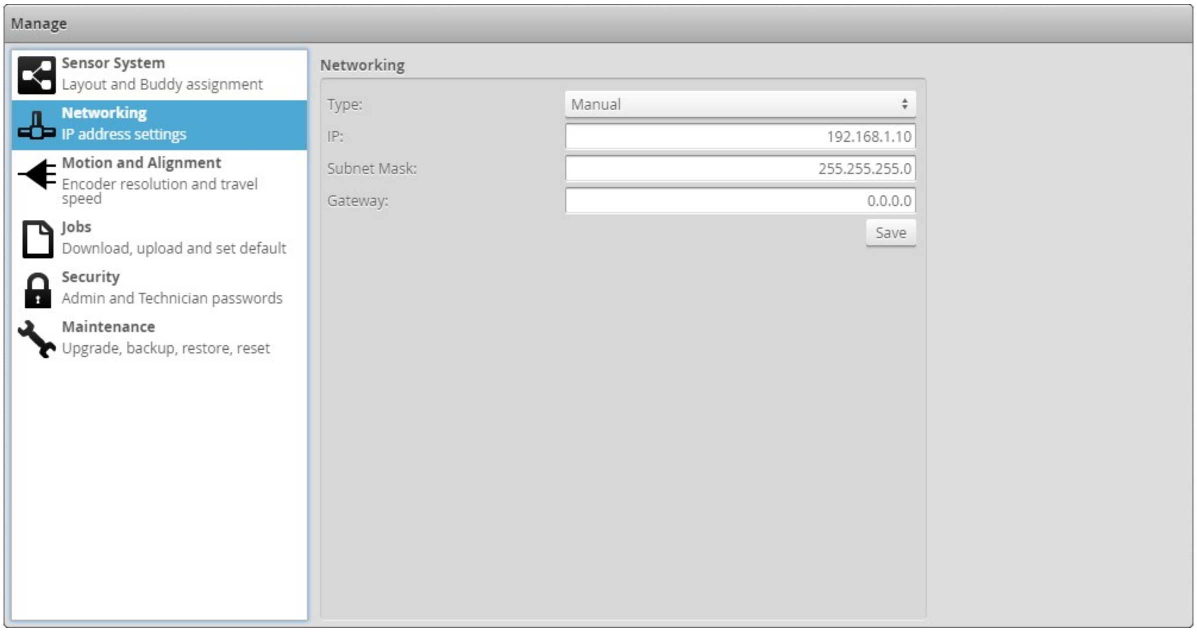
   The Buddy sensor will be assigned to the Main sensor and its status will be updated in the **System** panel.



To remove a Buddy, click on the **Remove** button.

Networking

The **Networking** category on the **Manage** page provides network settings. Settings must be configured to match the network to which the Gocator sensors are connected.

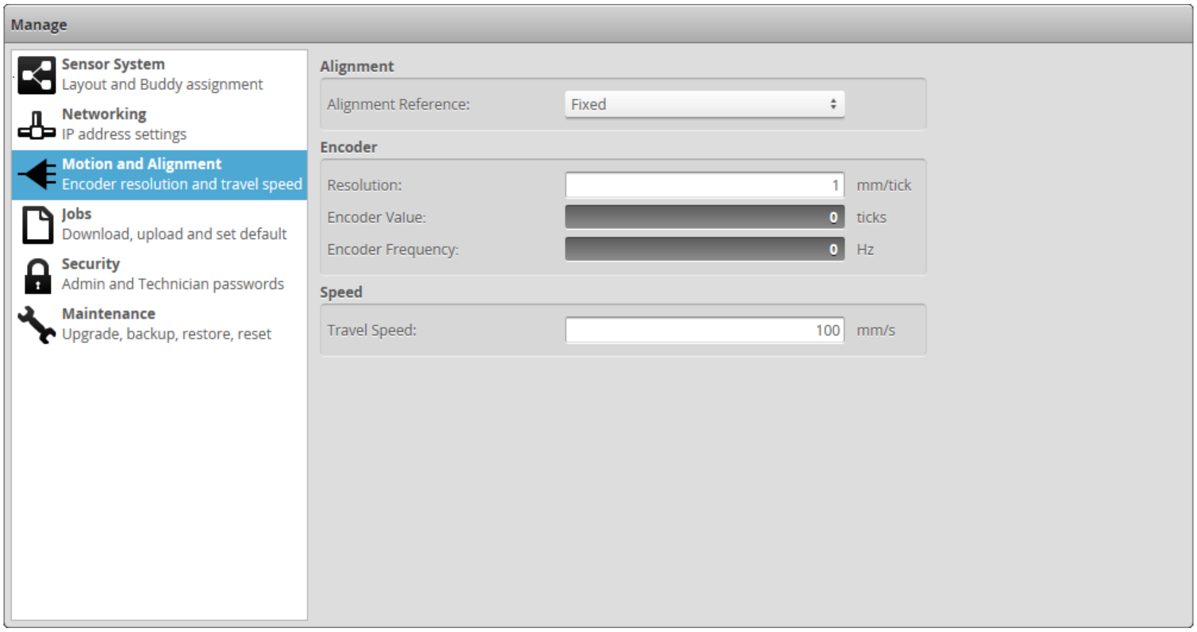


To configure the network settings:

1. Go to the **Manage** page.
2. In the **Networking** category, specify the Type, IP, Subnet Mask, and Gateway settings.
   The Gocator sensor can be configured to use DHCP or assigned a static IP address.
3. Click on the **Save** button.
   You will be prompted to confirm your selection.

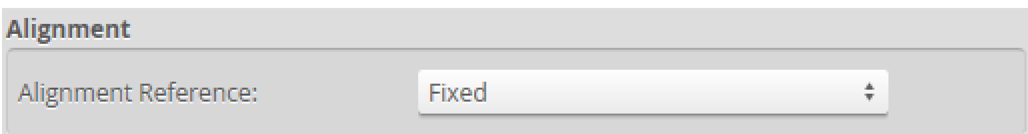
Motion and Alignment

The **Motion and Alignment** category on the **Manage** page lets you configure alignment reference, encoder resolution, and travel speed.



Alignment Reference

The **Alignment Reference** setting can have one of two values: **Fixed** or **Dynamic**.



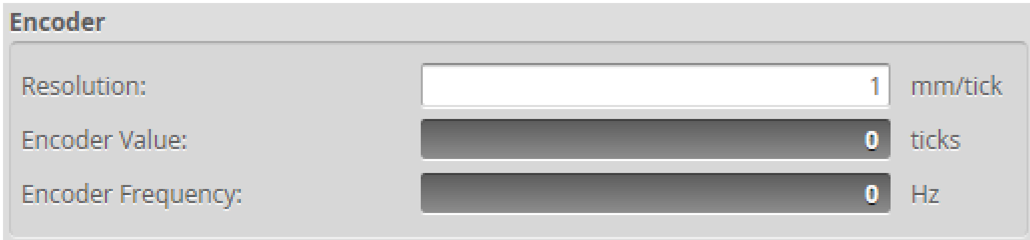
Setting	Description
Fixed	A single global alignment is used for all jobs. This is typically used when the sensor mounting is constant over time and between scans, for example, when the sensor is mounted in a permanent position over a conveyor belt.
Dynamic	A separate alignment is used for each job. This is typically used when the sensor's position relative to the object scanned is always changing, for example, when the sensor is mounted on a robot arm moving to different scanning locations.

To configure alignment reference:

1. Go to the **Manage** page and click on the **Motion and Alignment** category.
2. In the Alignment section, choose **Fixed** or **Dynamic** in the **Alignment Reference** drop-down.

Encoder Resolution

You can manually enter the encoder resolution in the **Resolution** setting , or it can be automatically set by performing an alignment with **Type** set to **Moving**. Establishing the correct encoder resolution is required for correct scaling of the scan of the target object in the direction of travel.



Encoder resolution is expressed in millimeters per tick.

To configure encoder resolution:

1. Go to the **Manage** page and click on the **Motion and Alignment** category.
2. In the **Encoder** section, enter a value in the **Resolution** field.

Encoder Value and Frequency

The encoder value and frequency are used to confirm the encoder is correctly wired to the Gocator and to manually calibrate encoder resolution (that is, by moving the conveyor system a known distance and making a note of the encoder value at the start and end of movement).

Travel Speed

The **Travel Speed** setting is used to correctly scale scans in the direction of travel in systems that lack an encoder but have a conveyor system that is controlled to move at constant speed. Establishing the correct travel speed is required for correct scaling of the scan in the direction of travel.

Travel speed is expressed in millimeters per second.

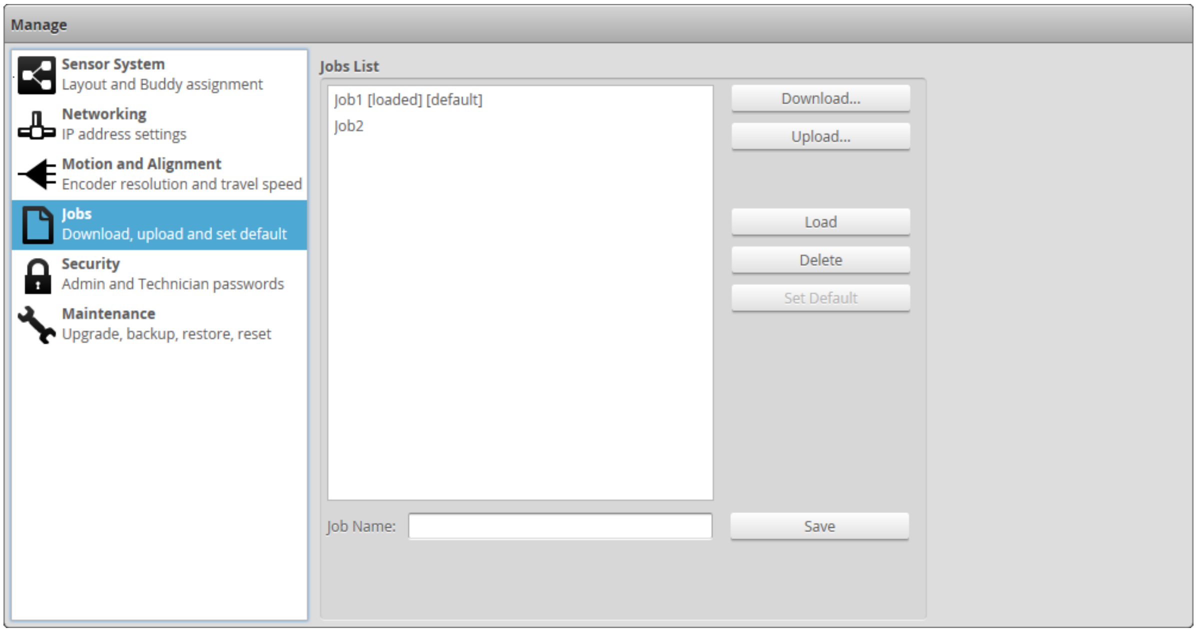
To manually configure travel speed:

1. Go to the **Manage** page and click on the **Motion and Alignment** category.
2. In the **Speed** section, enter a value in the **Travel Speed** field.

Travel speed can also be set automatically by performing an alignment with **Type** set to **Moving** (see page 77).

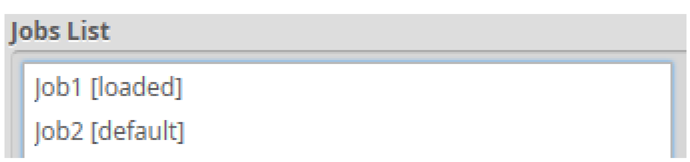
Jobs

The **Jobs** category on the **Manage** page lets you manage the jobs stored on the sensor.

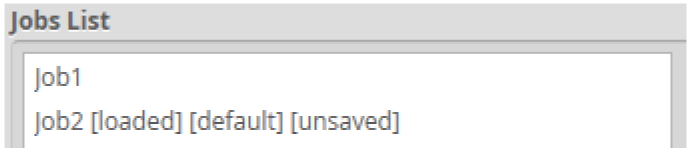


Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the sensor's flash storage.
Save button	Saves current settings to the job using the name in the **Job Name** field.
Load button	Loads the job that is selected in the file list. Reloading the current job discards any unsaved changes.
Delete button	Deletes the job that is selected in the jobs list.
Set as Default button	Sets the selected job as the default to be loaded at boot time. When the default job is selected, this button is used to clear the default.
Download... button	Downloads the selected jobs to the client computer.
Upload... button	Uploads a job from the client computer.

Jobs can be loaded and set as default independently. For example, Job1 could be loaded, while Job2 is set as the default. Default jobs load automatically when a sensor is power cycled or reset.



Unsaved jobs are indicated by "[unsaved]".



To download, load, or delete a job, or to set one as a default or clear a default:

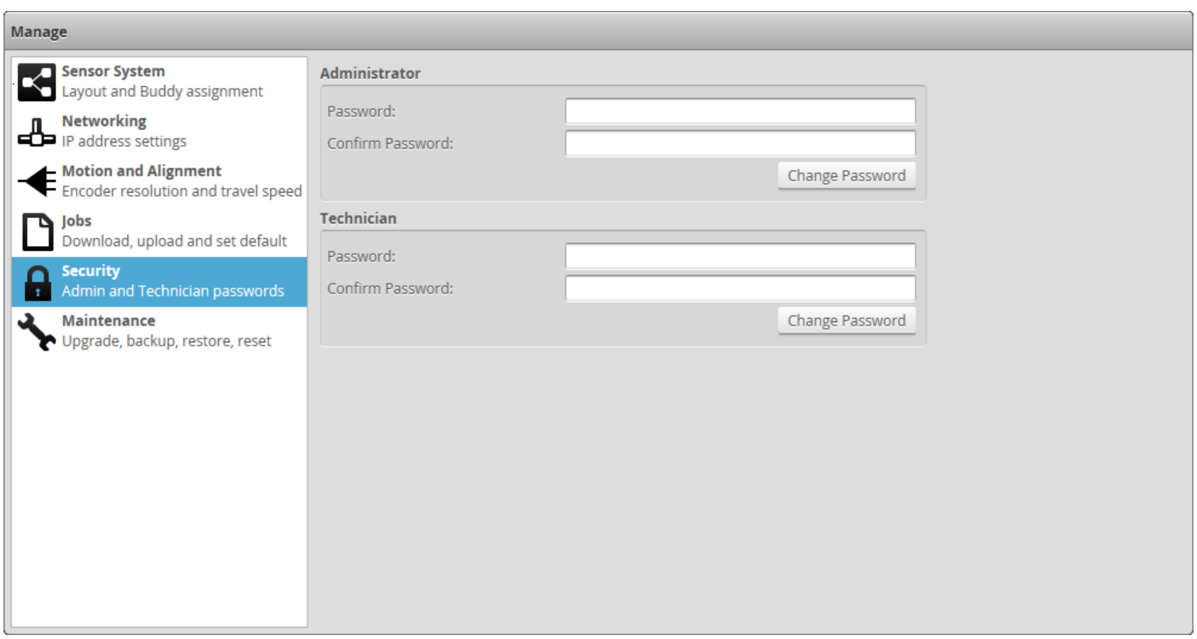
1. Go to the **Manage** page and click on the **Jobs** category.
2. Select a job in the **Jobs** list.
3. Click on the appropriate button for the operation.

To save a job:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Provide a name in the **Job Name** field.
   To save an existing job under a different name, click on it in the **Jobs** list and then modify it in the **Job Name** field.
3. Click on the **Save** button or press **Enter**.
   Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

Security

Gocator sensors can be secured with passwords to prevent unauthorized access. Each sensor has two accounts: Administrator and Technician.



Gocator Account Types

Account	Description
Administrator	The Administrator account has privileges to use the toolbar (loading and saving jobs, recording and viewing replay data), to view all pages and edit all settings, and to perform setup procedures such as sensor alignment.
Technician	The Technician account has privileges to use the toolbar (loading and saving jobs, recording and viewing replay data), to view the **Dashboard** page, and to start or stop the sensor.

The Administrator and Technician accounts can be assigned unique passwords. By default, passwords are blank (empty).

To set or change the password for the Administrator account:

1. Go to the **Manage** page and click on the **Security** category.
2. In the **Administrator** section, enter the Administrator account password and password confirmation.
3. Click **Change Password**.
   The new password will be required the next time that an administrator logs in to the sensor.

To set or change the password for the Technician account:

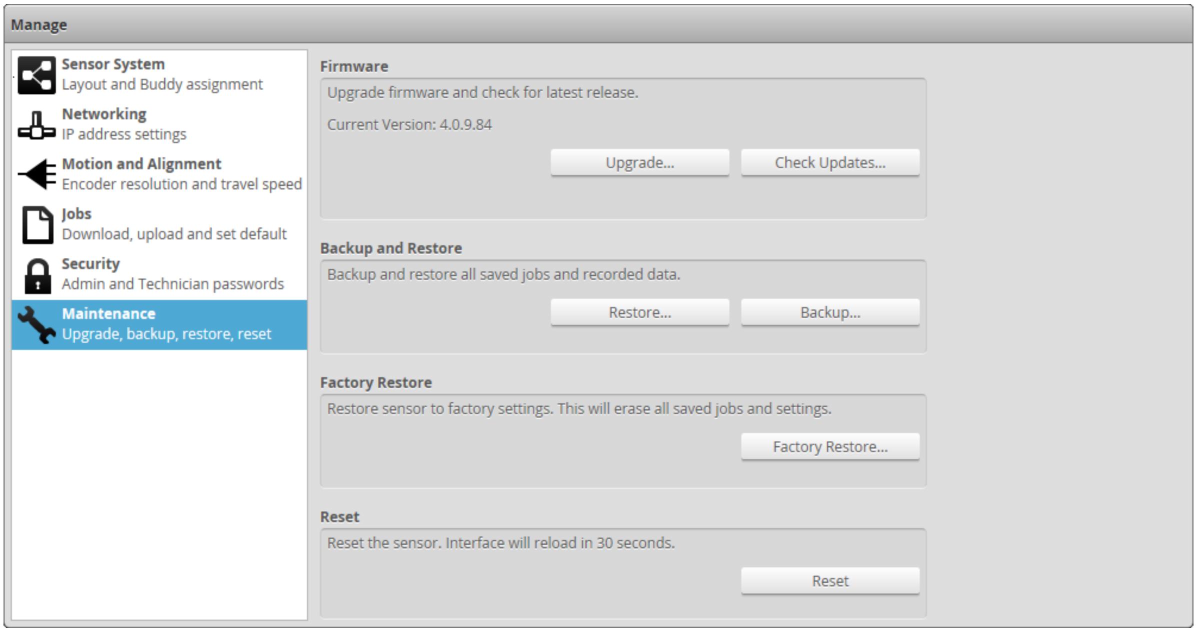
1. Go to the **Manage** page and click on the **Security** category.
2. In the **Technician** section, enter the Technician account password and password confirmation.
3. Click **Change Password**.
   The new password will be required the next time that a technician logs in to the sensor.

If the administrator or technician password is misplaced, the sensor can be recovered using a special software tool. See *Sensor Recovery Tool* (page 209) for more information.

Maintenance

The **Maintenance** category in the **Manage** page is used to do the following:

- upgrade the firmware and check for firmware updates;
- back up and restore all saved jobs and recorded data;
- restore the sensor to factory defaults;
- reset the sensor.

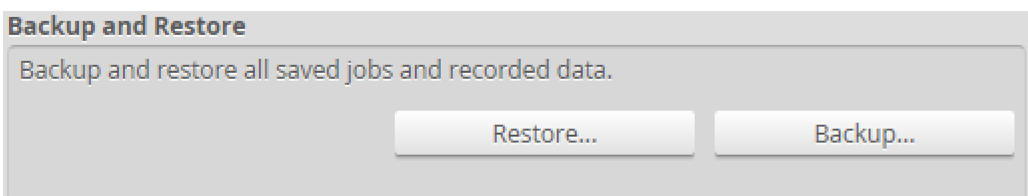


Sensor Backups and Factory Reset

You can create sensor backups, restore from a backup, and restore to factory defaults in the **Maintenance** category.

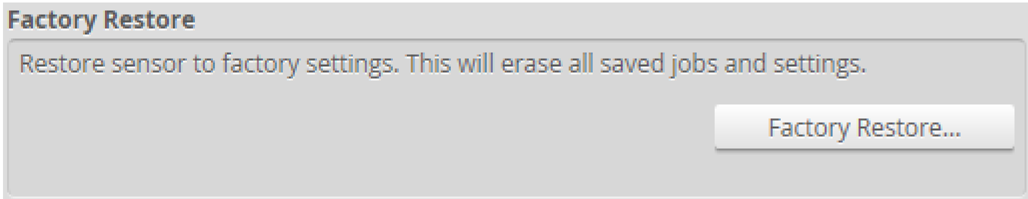
Backup files contain all of the information stored on a sensor, including jobs and alignment.

	An Administrator should create a backup file in the unlikely event that a sensor fails and a replacement sensor is needed. If this happens, the new sensor can be restored with the backup file.



To create a backup:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Backup...** button under **Backup and Restore**.
3. When you are prompted, save the backup.
   Backups are saved as a single archive that contains all of the files from the sensor.



To restore from a backup:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Restore...** button under **Backup and Restore**.
3. When you are prompted, select a backup file to restore.
   The backup file is uploaded and then used to restore the sensor. Any files that were on the sensor before the restore operation will be lost.

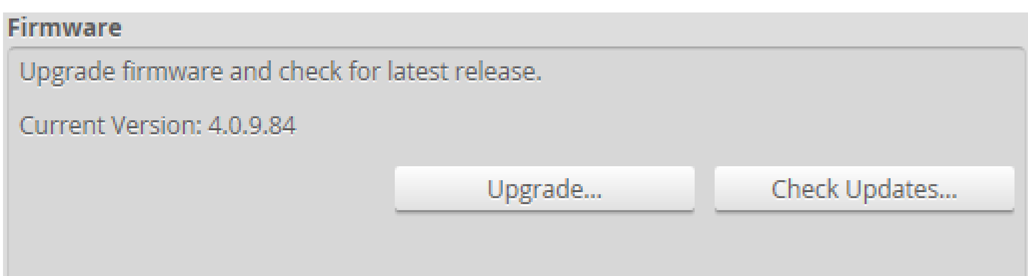
To restore a sensor to its factory default settings:

1. Go to the **Manage** page and click on **Maintenance**.
2. Consider making a backup.
   Before proceeding, you should perform a backup. Restoring to factory defaults cannot be undone.
3. Click the **Factory Restore...** button under **Factory Restore**.
   You will be prompted whether you want to proceed.

Firmware Upgrade

LMI recommends routinely updating firmware to ensure that Gocator sensors always have the latest features and fixes.

> In order for the Main and Buddy sensors to work together, they must be use the same firmware version. This can be achieved by upgrading through the Main sensor or by upgrading each sensor individually.



To download the latest firmware:

1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Check Updates...** button in the **Firmware** section.

3. Download the latest firmware.
   If a new version of the firmware is available, follow the instructions to download it to the client computer.

If the client computer is not connected to the Internet, firmware can be downloaded and transferred to the client computer by using another computer to download the firmware from LMI's website: *http://www.lmi3D.com/support/downloads*.

To upgrade the firmware:

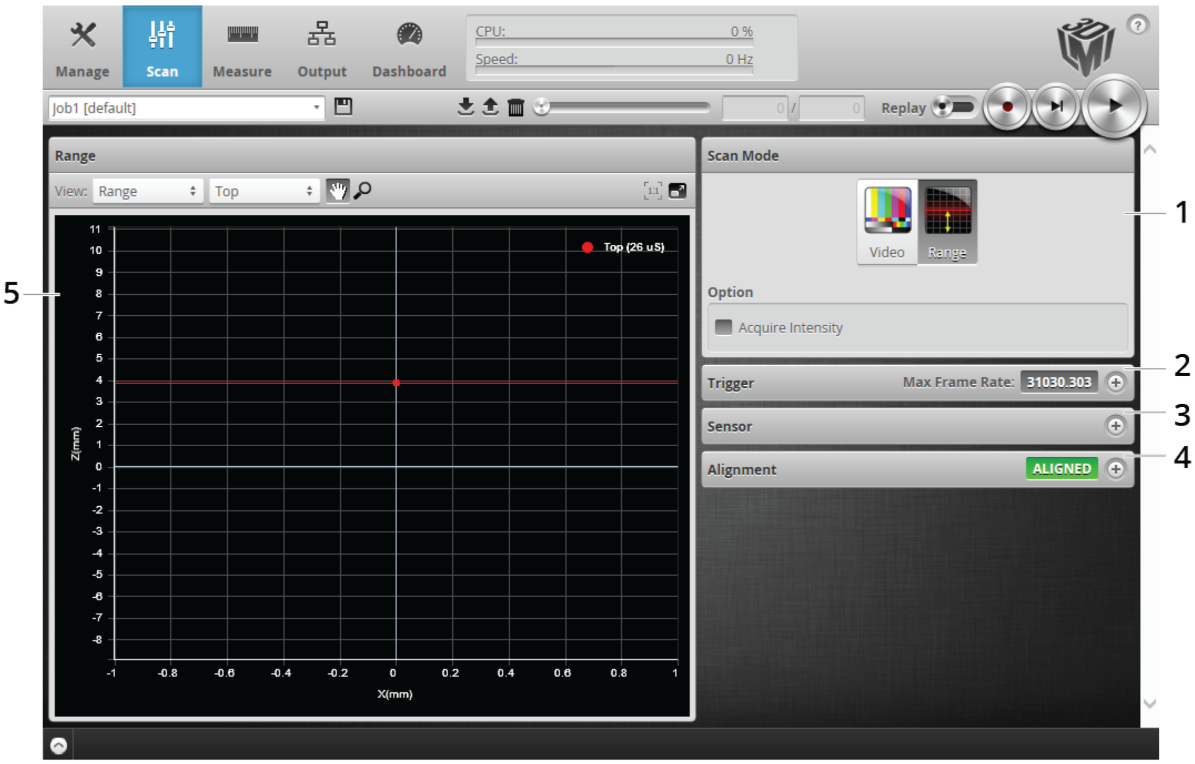
1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Upgrade...** button in the **Firmware** section.
3. Provide the location of the firmware file in the **File** dialog.
4. Wait for the upgrade to complete.
   After the firmware upgrade is complete, the sensor will self-reset. If a buddy has been assigned, it will be upgraded and reset automatically.

Scan Setup and Alignment

The following sections describe the steps to configure Gocator sensors for laser ranging using the **Scan** page. Setup and alignment should be performed before adding and configuring measurements or outputs.

Scan Page Overview

The **Scan** page lets you configure sensors and perform alignment.



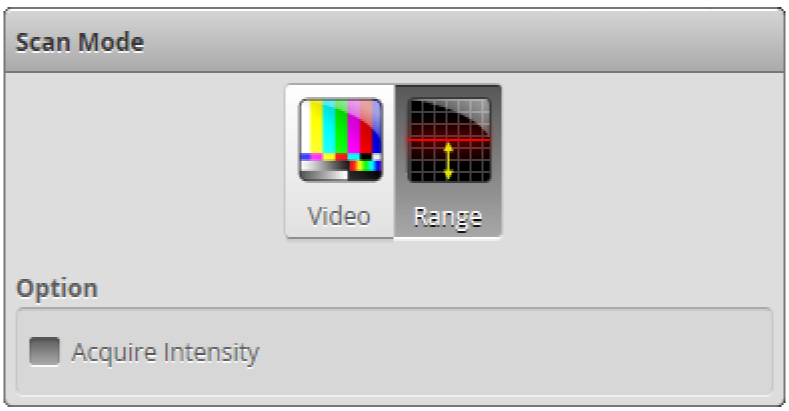
	Element	Description
1	Scan Mode panel	Contains settings for the current scan mode (Video or Range) and other options. See *Scan Modes* (next page).
2	Trigger panel	Contains trigger source and trigger-related settings. See *Triggers* (next page).
3	Sensor panel	Contains settings for an individual sensor, such as active area or exposure. See *Sensor* (page 69).
4	Alignment panel	Used to perform alignment. See *Alignment* (page 76).
5	Data Viewer	Displays sensor data and adjust regions of interest. Depending on the current operation mode, the data viewer can display video images or range plots . See *Data Viewer* (page 81).

The following table provides quick references for specific goals that you can achieve from the panels in the **Scan** page.

Goal	Reference
Select a trigger source that is appropriate for the application.	Triggers (below)
Ensure that camera exposure is appropriate for laser ranging .	Exposure (page 72)
Find the right balance between range quality, speed, and CPU utilization.	Active Area (page 69) Exposure (page 72) Gocator Device Files (page 117)
Calibrate the system so that laser range data can be aligned to a common reference and values can be correctly scaled in the axis of motion.	Aligning Sensors (page 77)

Scan Modes

The Gocator web interface supports two scan modes: Video and Range. The scan mode can be selected in the **Scan Mode** panel.



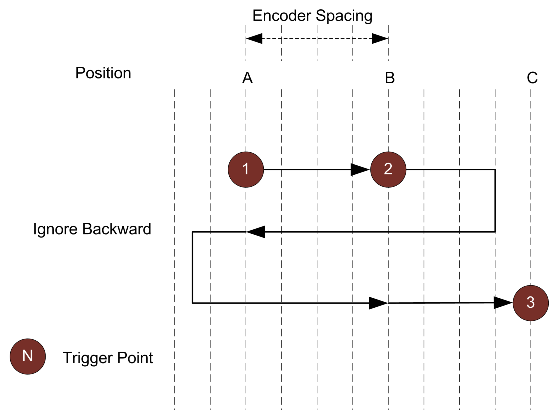
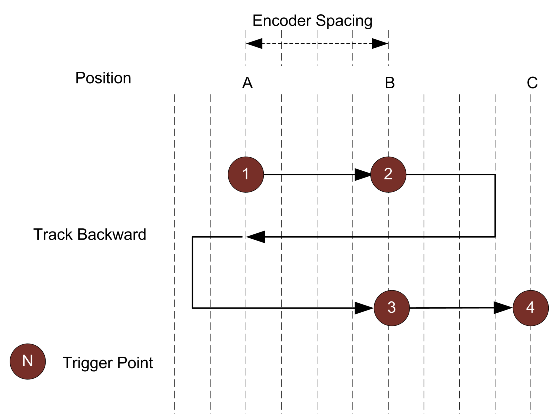
Mode and Option	Description
Video	Outputs video images from the Gocator. This mode is useful for configuring exposure time and troubleshooting stray light or ambient light problems.
Range	Outputs ranges and performs measurements. Video images are processed internally to produce laser ranges and measurements.
Surface	Outputs 3D point clouds and performs surface measurements.
Acquire Intensity	When this option is enabled, an intensity value will be produced for each laser range.

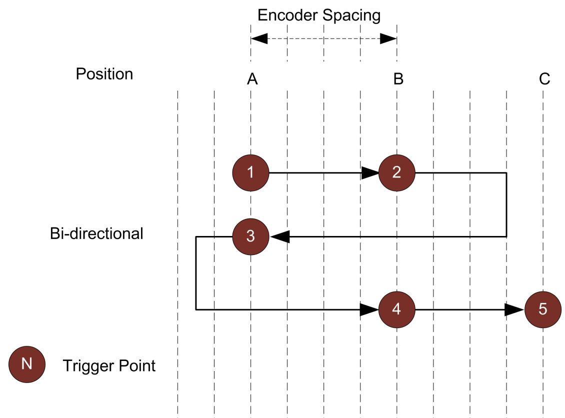
Triggers

A trigger is an event that causes a sensor to take a single picture. Triggers are configured in the **Trigger** panel on the **Scan** page.

When a trigger is processed, the laser is strobed and the camera exposes to produce an image. The resulting image is processed inside the sensor to yield a laser range (distance information), which can then be used for measurement.

The laser and camera inside a sensor can be triggered by one of the following sources:

Trigger Source	Description
Time	Sensors have an internal clock that can be used to generate fixed-frequency triggers. The external input can be used to enable or disable the time triggers.
Encoder	An encoder can be connected to provide triggers in response to motion. Three encoder triggering behaviors are supported. These behaviors are set using the **Behavior** setting. **Ignore Backward** A scan is triggered when the target object moves forward. If the target object moves backward, it must move forward by at least the distance that the target travelled backward, plus one encoder spacing, to trigger the next scan.  **Track Backward** A scan is triggered only when the target object moves forward. If the target object moves backward, it must move forward by at least the distance of one encoder spacing to trigger the next scan.  **Bi-directional** A scan is triggered when the target object moves forward or backward.

Trigger Source	Description
	 When triggers are received at a frequency higher than the maximum frame rate, some triggers may not be accepted. The **Trigger Drops Indicator** in the **Dashboard** can be used to check for this condition. The external input can be used to enable or disable the encoder triggers. See *Encoder Input* (page 237) for more information on connecting the encoder to Gocator sensors.
External Input	A digital input can provide triggers in response to external events (e.g., photocell). When triggers are received at a frequency higher than the maximum frame rate, some triggers may not be accepted. The **Trigger Drops Indicator** in the **Dashboard** page can be used to check for this condition. See *Digital Inputs* (page 236) for more information on connecting external input to Gocator.
Software	A network command can be used to send a software trigger. See *Protocols* (page 137) for more information.

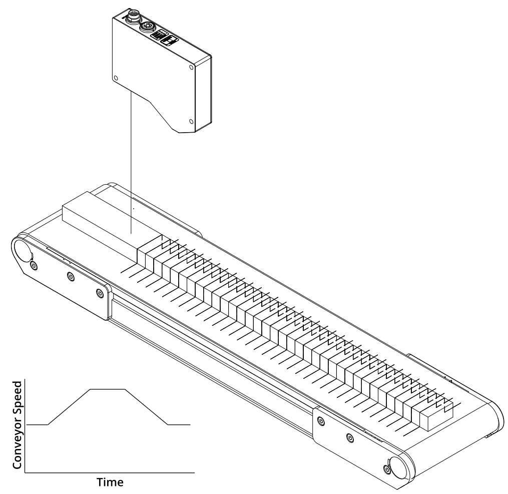
For examples of typical real-world scenarios, see next page. For information on the settings used with each trigger source, see page 68

Trigger Examples

Example: Encoder + Conveyor

Encoder triggering is used to perform range measurements at a uniform spacing.

The speed of the conveyor can vary while the object is being measured; an encoder ensures that the measurement spacing is consistent, independent of conveyor speed.

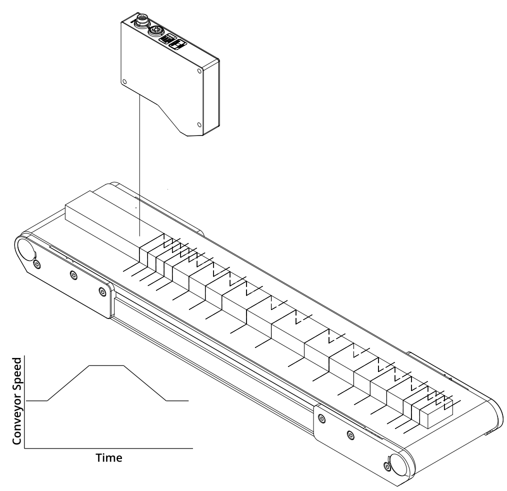


Example: Time + Conveyor

Time triggering can be used instead of encoder triggering to perform range measurements at a fixed frequency.

Measurement spacing will be non-uniform if the speed of the conveyor varies while the object is being measured.

It is strongly recommended to use an encoder with transport-based systems due to the difficulty in maintaining constant transport velocity.

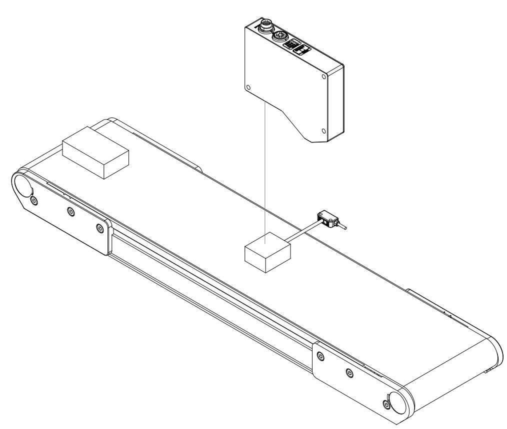


Example: External Input + Conveyor

External input triggering can be used to produce a snapshot for range measurement.

For example, a photocell can be connected as an external input to generate a trigger pulse when a target object has moved into position.

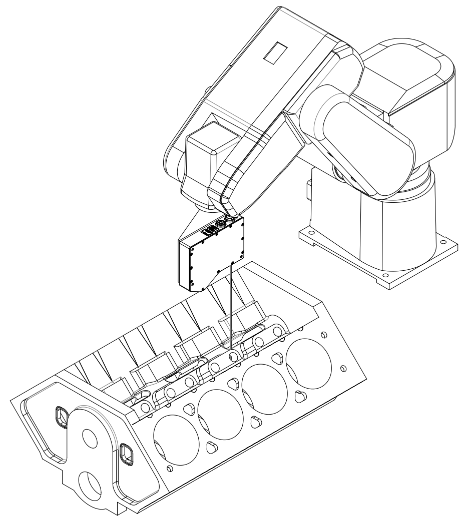
An external input can also be used to gate the trigger signals when time or encoder triggering is used. For example, a photocell could generate a series of trigger pulses as long as there is a target in position.



Example: Software Trigger + Robot Arm

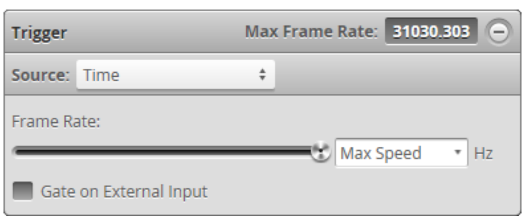
Software triggering can be used to produce a snapshot for range measurement.

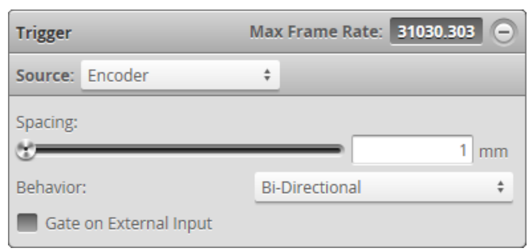
A software trigger can be used in systems that use external software to control the activities of system components.

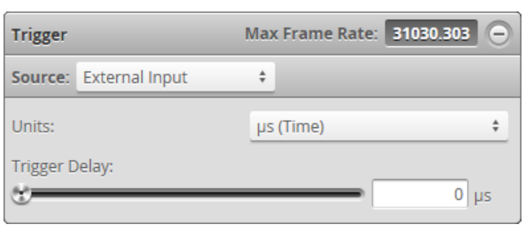


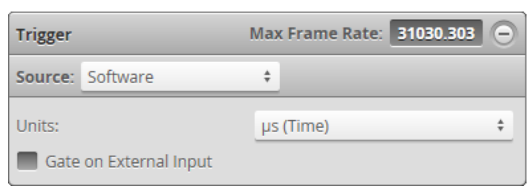
Trigger Settings

The trigger source is selected using the **Trigger** panel in the **Scan** page.









After specifying a trigger source, the **Trigger** panel shows the parameters that can be configured.

Parameter	Trigger Source	Description
Source	All	Selects the trigger source (**Time**, **External Input**, or **Software**).
Frame Rate	Time	Controls the frame rate. Select **Max Speed** from the drop-

Parameter	Trigger Source	Description
		down to lock to the maximum frame rate. Fractional values are supported. For example, 0.1 can be entered to run at 1 frame every 10 seconds.
Gate on External Input	Time, Encoder	External input can be used to enable or disable ranging in a sensor. When this option is enabled, the sensor will respond to time or encoder triggers only when the external input is asserted. See *Digital Inputs* (page 236) for more information on connecting external input to Gocator.
Behavior	Encoder	Specifies how the Gocator sensor is triggered when the target moves. Can be Track Backward, Ignore Backward, or Bi-Directional. See *Triggers* (page 64) for more information on these behaviors.
Spacing	Encoder	Specifies the distance between triggers (mm). Internally the Gocator sensor rounds the spacing to a multiple of the encoder resolution.
Units	External Input, Software	Specifies whether the trigger delay, output delay, and output scheduled command operate in the time or the encoder domain. The unit is implicitly set to microseconds with Time trigger source, and millimeters with Encoder trigger source.
Trigger Delay	External Input	Controls the amount of time or the distance the sensor waits before producing a frame after the external input is activated. This is used to compensate for the positional difference between the source of the external input trigger (e.g., photocells) and the sensor. Trigger delay is only supported in single exposure mode; for details, see *Exposure* (page 72).

To configure the trigger source:

1. Go to the **Scan** page.
2. Expand the **Trigger** panel by clicking on the panel header.
3. Select the trigger source from the drop-down.
4. Configure the settings.
   See the trigger parameters above for more information.
5. Save the job in the **Toolbar** by clicking the **Save** button.

Sensor

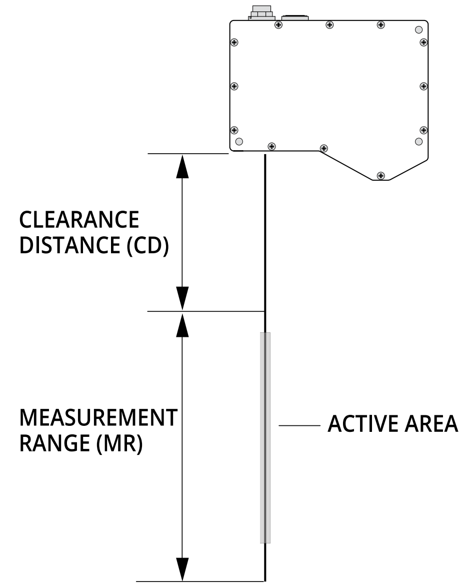
The following sections describe the settings that are configured in the **Sensor** panel on the **Scan** page.

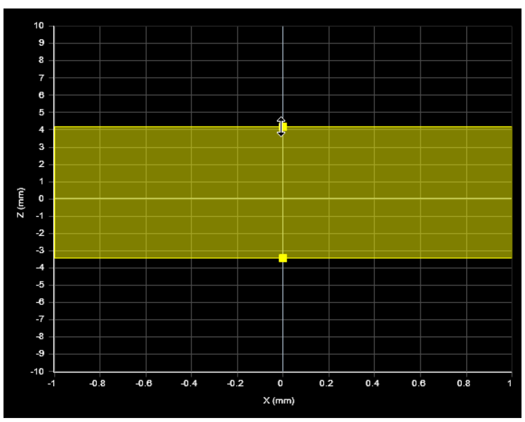
Active Area

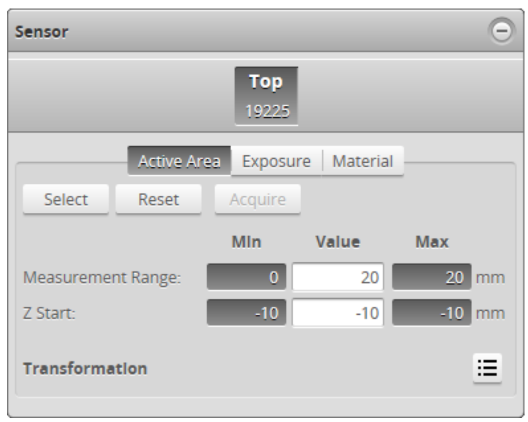
Active area refers to the region within the sensor's maximum field of view that is used for laser ranging.

By default, the active area covers the sensor's entire field of view. By reducing the active area, the sensor can operate at higher speeds.

Active area is specified in sensor coordinates, rather than in system coordinates. As a result, if the sensor is already alignment calibrated, press the **Acquire** button to display uncalibrated data before configuring the active area. See *Coordinate Systems* (page 39) for more information on sensor and system coordinates.







To set the active area:

1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure the active area.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click the button corresponding to the sensor you want to configure.
   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.

Active area is specified separately for each sensor.

5. Click on the **Active Area** tab.
6. Click the **Select** button.
7. Click the **Acquire** button to see a scan while setting the active area.
8. Set the active area.
   Enter the active area values in the edit boxes or adjust the active area graphically in the data viewer.
9. Click the **Save** button in the **Sensor** panel.
   Click the **Cancel** button to cancel setting the active area.
10. Save the job in the **Toolbar** by clicking the **Save** button.

> Laser ranging devices are usually more accurate at the near end of their measurement range. If your application requires a measurement range that is small compared to the maximum measurement range of the sensor, mount the sensor so that the active area can be defined at the near end of the measurement range.

Transformations

The transformation settings are used to control how ranges are converted from sensor coordinates to system coordinates.

Parameter	Description
Z Offset	Specifies the shift along the Z axis. A positive value shifts the toward the sensor.

When applying the transformations, Angle is applied before the Z offset.

To configure transformation settings:

1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure the transformations.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Click the button corresponding to the sensor you want to configure.
   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
   Transformations can be configured separately for each sensor.

5. Expand the Transformations area by clicking on the expand button.
   See the table above for more information.

6. Set the parameter values.
   See the table above for more information.

7. Save the job in the **Toolbar** by clicking the **Save** button.

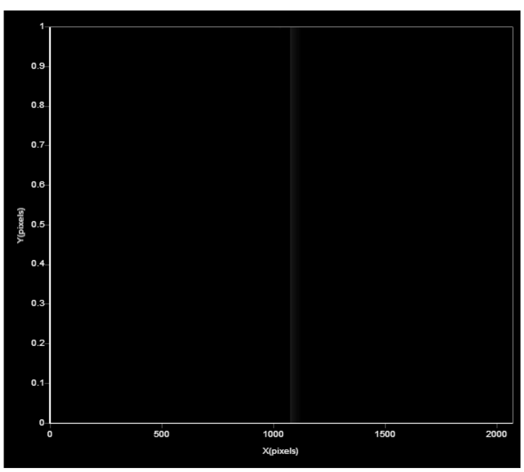
8. Check that the transformation settings are applied correctly after ranging is restarted.

Exposure

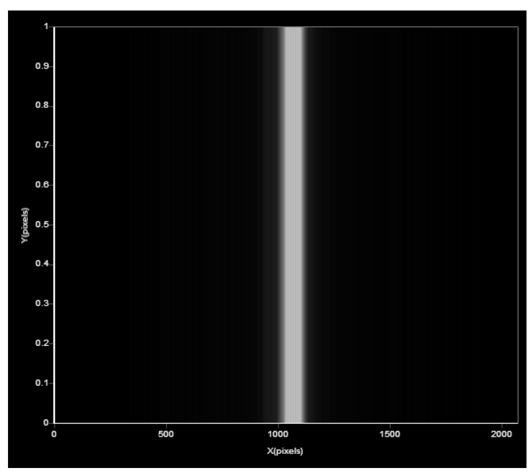
Exposure determines the duration of camera and laser on-time. Longer exposures can be helpful to detect laser signals on dark or distant surfaces, but increasing exposure time decreases the maximum speed. Different target surfaces may require different exposures for optimal results. Gocator sensors provide two exposure modes for the flexibility needed to scan different types of target surfaces.

Exposure Mode	Description
Single	Uses a single exposure for all objects. Used when the surface is uniform and is the same for all targets.
Dynamic	Automatically adjusts the exposure after each frame. Used when the target surface varies between scans.

Video mode lets you see how the laser point appears on the camera and identify any stray light or ambient light problems. When exposure is tuned correctly, the laser should be clearly visible along the entire length of the viewer. If it is too dim, increase the exposure value; if it is too bright decrease exposure value.



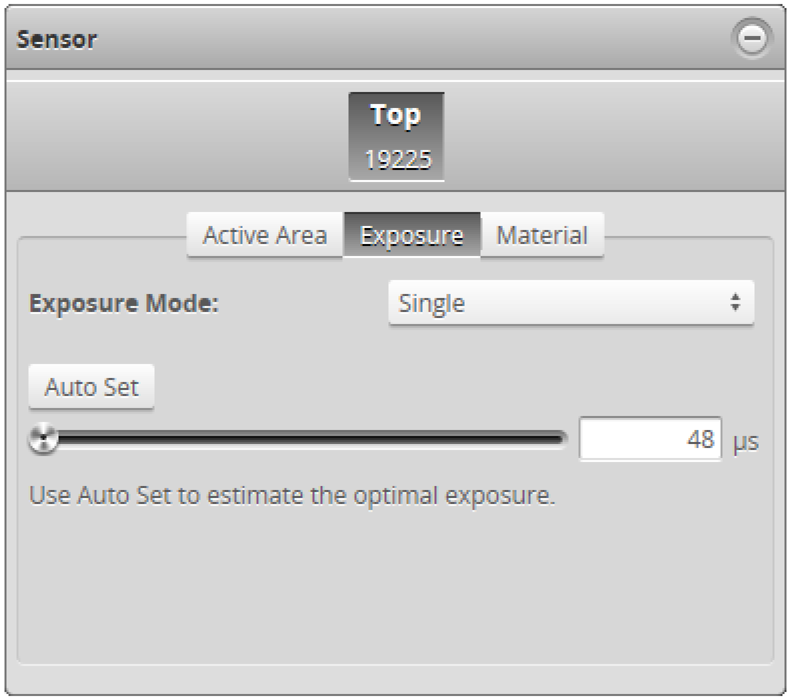
Under exposure
Laser point is not detected.
Increase the exposure value.



Over exposure
Laser point is too bright .
Increase the exposure value.

Single Exposure

The sensor uses a fixed exposure in every scan. Single exposure is used when the target surface is uniform and is the same for all targets.



To enable single exposure:

1. Place a representative target in view of the sensor.
   The target surface should be similar to the material that will normally be measured.

2. Go to the **Scan** page.

3. Expand the **Sensor** panel by clicking on the panel header.

4. Click the button corresponding to the sensor you want to configure.
   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
   Exposure can be configured separately for each sensor.

5. Click on the **Exposure** tab.

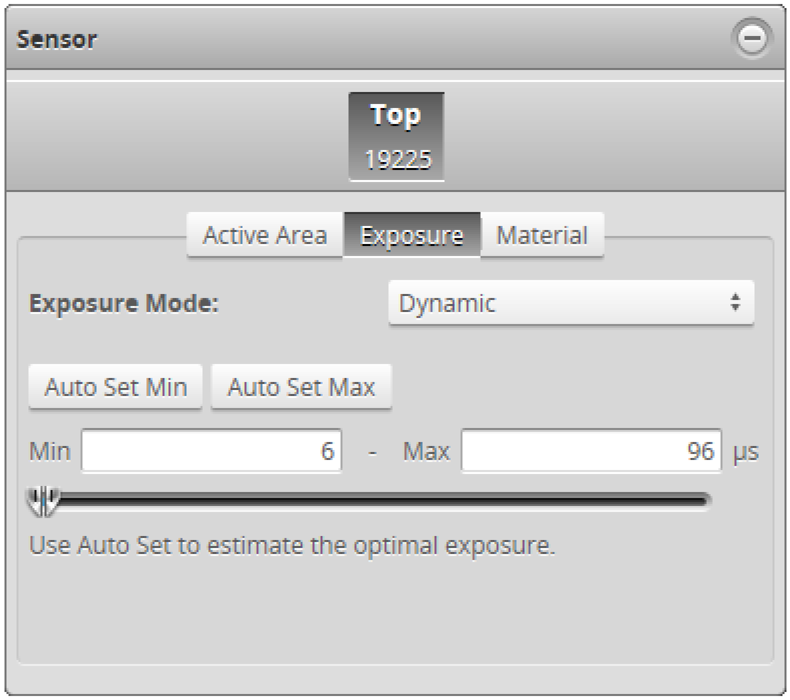
6. Select **Single** from the **Exposure Mode** drop-down.

7. Edit the **Exposure** setting.
   You can automatically tune the exposure by pressing the **Auto Set** button, which causes the sensor to turn on and tune the exposure time.

8. Run the sensor and check that laser ranging is satisfactory.
   If laser ranging is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* (previous page) for details.

Dynamic Exposure

The sensor automatically uses past range information to adjust the exposure to yield the best range . This is used when the target surface changes from scan to scan.



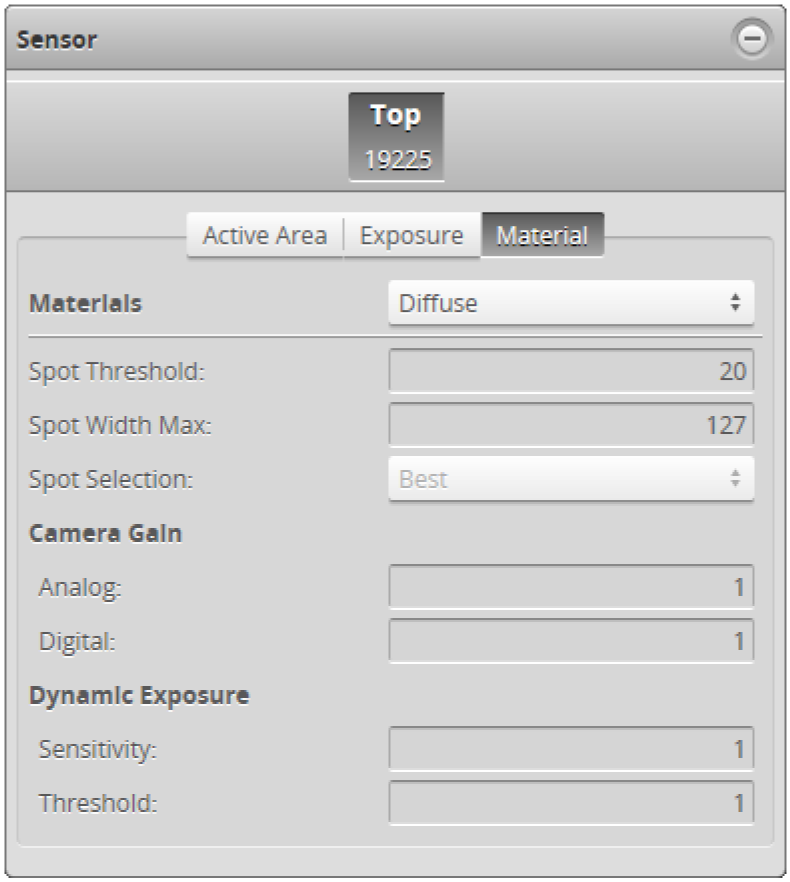
To enable dynamic exposure:

1. Go to the **Scan** page.
2. Expand the **Sensor** panel by clicking on the panel header or the button.
3. Click the button corresponding to the sensor you want to configure.
   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
   Exposure can be configured separately for each sensor.
4. Click on the **Exposure** tab.
5. Select **Dynamic** from the **Exposure Mode** drop-down.
6. Set the minimum and maximum exposure.
   The auto-set function can be used to automatically set the exposure. First, place the brightest target in the field of view and press the **Auto Set Min** button to set the minimum exposure. Then, place the darkest target in the field of view and press the **Auto Set Max** button to set the maximum exposure.
7. Run the sensor and check that laser ranging is satisfactory.
   If laser ranging is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* (page 72) for details.

Material

Range data acquisition can be configured to suit different types of target materials. For many targets, the setting it is not necessary, but it can make a great difference with others.

Preset material types can be selected in the **Materials** setting.



When **Materials** is set to **Custom**, the following settings can be configured:

Setting	Description
Spot Threshold	The minimum increase in intensity level between neighbouring pixels for a pixel to be considered the start of a potential spot. This setting is important for filtering false range spots generated by sunlight reflection.
Spot Width Max	The maximum number of pixels a spot is allowed to span. This setting can be used to filter out data caused by background light if the unwanted light is wider than the laser and does not merge into the laser itself. A lower **Spot Width** setting reduces the chance of false detection, but limits the ability to detect features/surfaces that elongate the spot.
Spot Selection	Determines the spot selection method (**Best**, **Top**, or **Bottom**). **Best** selects the strongest spot in a given column on the imager. **Top** (the one farthest to the left on the imager) and **Bottom** (the one farthest to the right on the imager) can

Setting	Description
	be useful in applications where there are reflections, flying sparks or smoke, which are always on one side of the laser.
Analog	**Analog** camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor.
Digital	**Digital** camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor.
Sensitivity	Controls the exposure that dynamic exposure converges to. The lower the value, the lower the exposure Gocator will settle on. The trade-off is between the number of exposure spots and the possibility of over-exposing.
Threshold	The minimum number of spots for dynamic exposure to consider the spot valid. If the number of spots is below this threshold, the algorithm will walk over the allowed exposure range slowly to find the correct exposure.

To configure material:

1. Go to the **Scan** page.
2. Expand the **Sensor** panel by clicking on the panel header or the button.
3. Click the button corresponding to the sensor you want to configure.
   The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
   Materials can be configured separately for each sensor.
4. Click on the **Materials** tab.
5. Choose a material in the **Materials** drop-down or choose **Custom** to manually configure settings.
   See the table above for the customizable settings.
6. Save the job in the **Toolbar** by clicking the **Save** button .
7. Check that laser ranging is satisfactory.
   After adjusting the setting, confirm that laser profiling is satisfactory.

Various settings can affect how the **Material** settings behave. You can use Video mode to examine how the settings interact. See *Spots and Dropouts* (page 82) for more information.

Alignment

Gocator sensors are pre-calibrated and ready to deliver ranges in engineering units (mm) out of the box. However, alignment procedures are required to compensate for sensor mounting inaccuracies, to align multiple sensors into a common coordinate system, and to determine the resolution (with encoder) and speed of the transport system. Alignment is performed using the **Alignment** panel on the **Scan** page.

Once alignment has been completed, the derived transformation values will be displayed under **Transformations** in the **Sensor** panel; see *Transformations* (page 71) for details.

Alignment States

A Gocator can be in one of three alignment states: None, Manual, or Auto.

Alignment State

State	Explanation
None	Sensor is not aligned. Ranges are reported in default sensor coordinates.
Manual	Transformations (see page 71) or encoder resolution (see page 68) have been manually edited.
Auto	Sensor is aligned using the alignment procedure (see below).

An indicator on the **Alignment** panel will display ALIGNED or UNALIGNED, depending on the Gocator's state.

Alignment Types

Gocator sensors support two types of alignment, which are related to whether the target is stationary or moving.

Type	Description
Stationary	**Stationary** is used when the sensor mounting is constant over time and between scans, e.g., when the sensor is mounted in a permanent position over a conveyor belt.
Moving	**Moving** is used when the sensor's position relative to the object scanned is always changing, e.g., when the sensor is mounted on a robot arm moving to different scanning locations.

Alignment: With and Without Encoder Calibration

For systems that use an encoder, encoder calibration can be performed while aligning sensors. The table below summarizes the differences between performing alignment with and without encoder calibration calibration.

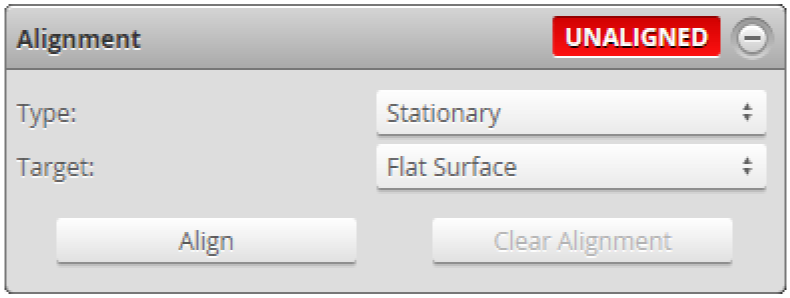
	With encoder calibration	Without encoder calibration
Target Type	Calibration bar	Flat surface or calibration bar
Target/Sensor Motion	Linear motion	Stationary
Calibrates Z axis Offset	Yes	Yes
Calibrates Encoder	Yes	No
Calibrates Travel Speed	Yes	No

See *Coordinate Systems* (page 39) for definitions of coordinate axes. See *Calibration Targets* (page 21) for descriptions of calibration disks and bars.

See *Aligning Sensors* (below) for the procedure to perform alignment. After alignment, the coordinate system for laser ranges will change from sensor coordinates to system coordinates.

Aligning Sensors

Alignment can be used to compensate for mounting inaccuracies by aligning sensor data to a common reference surface (often a conveyor belt).



To prepare for alignment:

1. Choose an alignment reference in the **Manage** page if you have not already done so.
   See *Alignment Reference* (page 55) for more information.

2. Go to the **Scan** page.

3. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, the **Alignment** panel will not be displayed.

4. Expand the **Alignment** panel by clicking on the panel header or the button.

5. Ensure that all sensors have a clear view of the target surface.
   Remove any irregular objects from the sensor's field of view that might interfere with alignment.

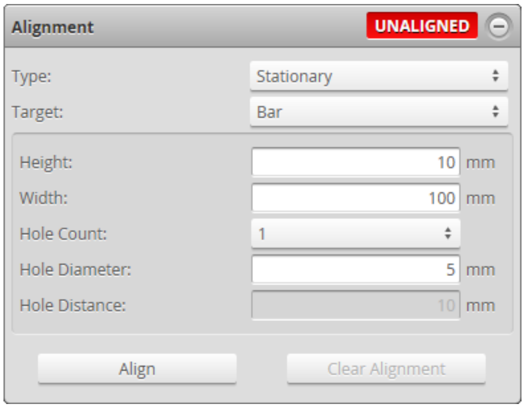
To perform alignment for stationary targets:

1. In the **Alignment** panel, select **Stationary** as the **Type**.

2. Clear the previous alignment if present.
   Press the **Clear Alignment** button to remove an existing alignment.

3. Select an alignment **Target**.
   - Select **Flat Surface** to use the conveyor surface (or other flat surface) as the alignment reference
   - Select **Bar** to use a custom calibration bar. If using a calibration bar, specify the bar dimensions and reference hole layout. See *Calibration Targets* (page 21) for details.

   Configure the characteristics of the target.



4. Place the target under the sensor

5. Click the **Align** button.
   The sensors will start, and the alignment process will take place. Alignment is performed simultaneously for all sensors. If the sensors do not align, check and adjust the exposure settings (page 72).

   > Alignment uses the exposure defined for single exposure mode, regardless of the current exposure mode

6. Use **Range** mode to inspect alignment results.
   Laser ranges from all sensors should now be aligned to the alignment target surface. The base of the alignment target (or target surface) provides the origin for the system Z axis.

To perform alignment for moving targets:

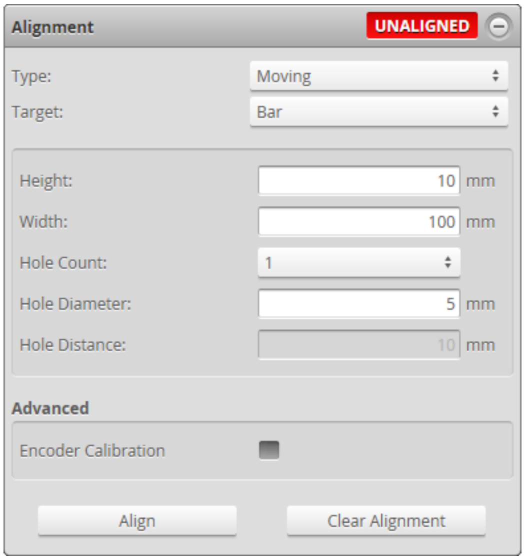
1. Do one of the following if you have not already done so.
   - If the system uses an encoder, configure encoder resolution. See *Encoder Resolution* (page 55) for more information.
   - If the system does not use an encoder, configure travel speed. See *Travel Speed* (page 56) for more information.

2. In the **Alignment** panel, select **Moving** as the **Type**.

3. Clear the previous alignment if present.
   Press the **Clear Alignment** button to remove an existing alignment.

4. Select an alignment **Target**.
   - Select one of the disk **Disk** options to use a disk as the alignment reference.
   - Select **Bar** to use a custom calibration bar. If using a calibration bar, specify the bar dimensions and reference hole layout. See *Calibration Targets* (page 21) for details.

     Configure the characteristics of the target.



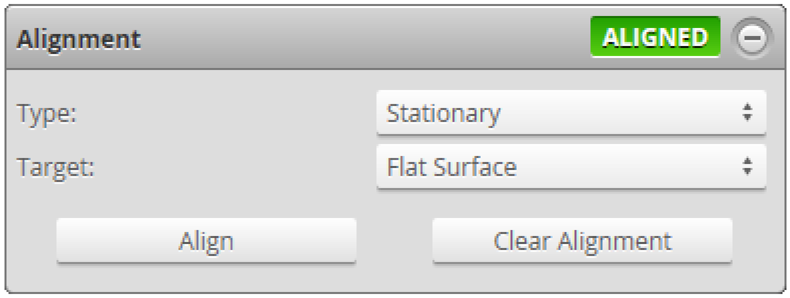
5. Place the target under the sensor
6. If the system uses an encoder and you want to calibrate it, check the **Encoder Calibration** checkbox.
7. Click the **Align** button.
   The sensors will start and then wait for the calibration target to pass through the laser plane. Alignment is performed simultaneously for all sensors. If the sensors do not align, check and adjust the exposure settings (page 72).

🗋	Alignment uses the exposure defined for single exposure mode, regardless of the current exposure mode

8. Engage the transport system.
   When the calibration target has passed completely through the laser plane, the calibration process will complete automatically. To properly calibrate the travel speed, the transport system must be running at the production operating speed before the target passes through the laser plane.
9. Use **Range** mode to inspect alignment results.
   Laser ranges from all sensors should now be aligned to the alignment target surface. The base of the alignment target (or target surface) provides the origin for the system Z axis.

Clearing Alignment

Alignment can be cleared to revert the sensor to sensor coordinates.



To clear alignment:

1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, the **Alignment** panel will not be displayed.
3. Expand the **Alignment** panel by clicking on the panel header or the button.
4. Click the **Clear Alignment** button.
   The alignment will be erased and sensors will revert to using sensor coordinates.

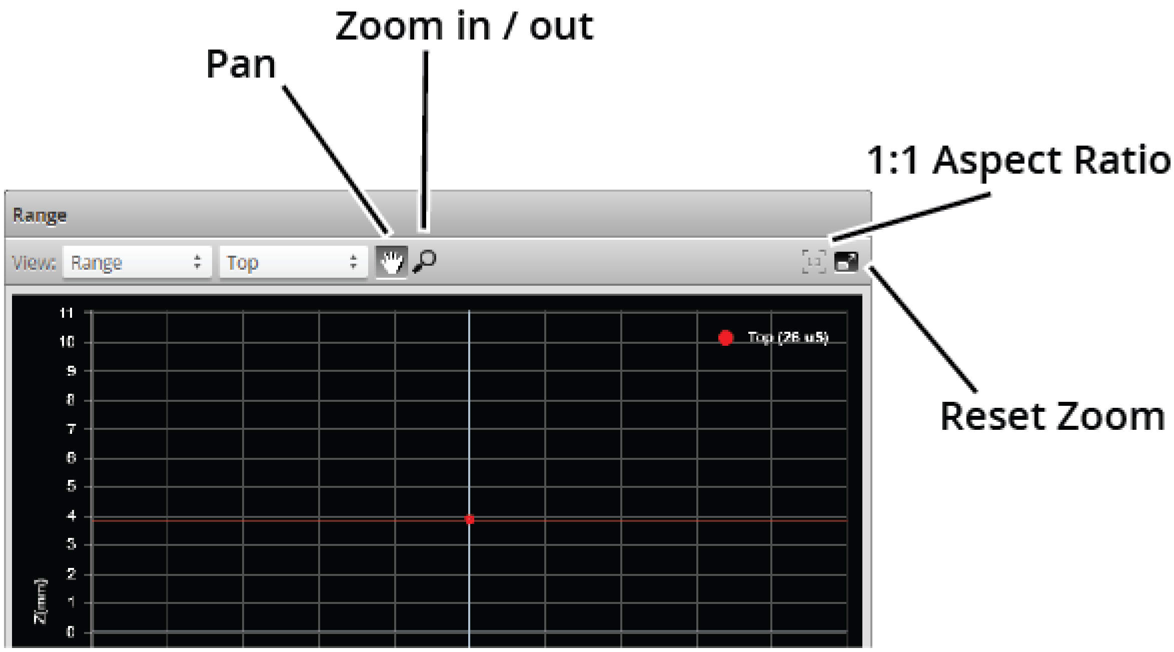
Data Viewer

The data viewer can display video images, range plots, and intensity images. It is also used to configure the active area (see page 69) and measurement tools (see page 86). The data viewer changes depending on the current operation mode and the panel that has been selected.

Data Viewer Controls

The data viewer is controlled by mouse clicks and by the buttons on the display toolbar. The mouse wheel can also be used for zooming in and out.

Press 'F' when the cursor is in the data viewer to switch to full screen.



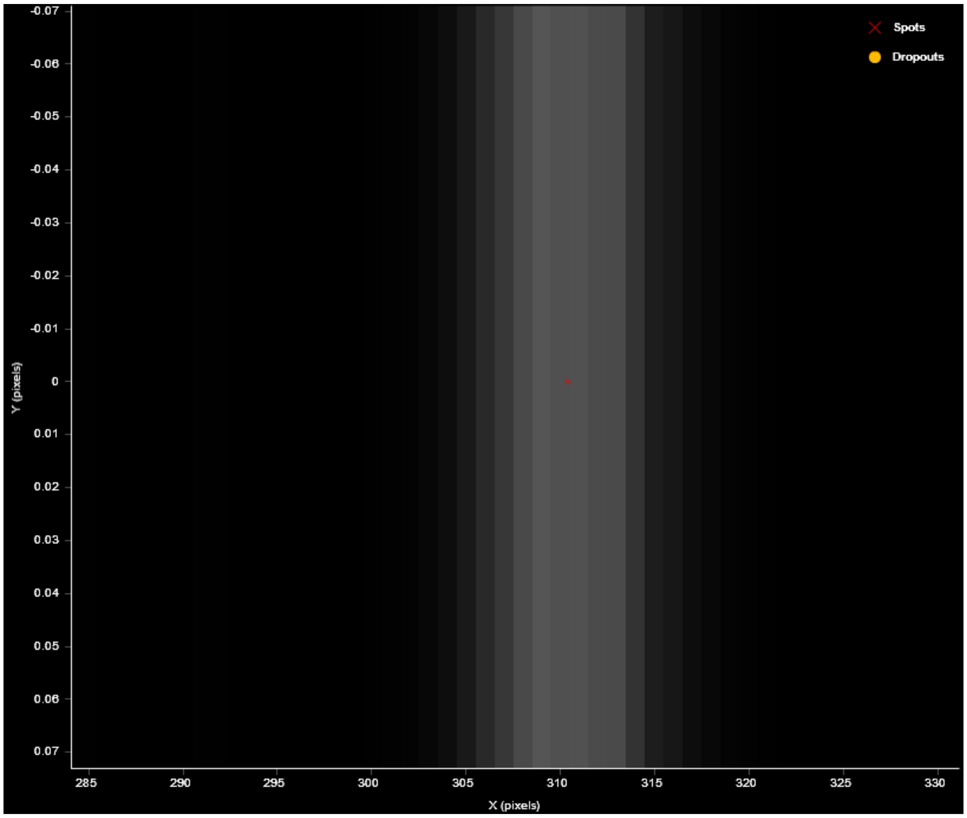
Video Mode

In Video scan mode, the data viewer displays a camera image. In a dual-sensor system, camera images from the Main or the Buddy sensor can be displayed.

Spots and Dropouts

Various settings can affect how the **Material** settings behave. Video mode can be used to examine how the **Material** settings are affected. To do this, check the **Show Spots** option at the top of the data viewer to overlay spot data in the data viewer. To show data dropouts, check the **Show Dropouts** option at the top of the data viewer.

In the image below, the white and gray squares represent the laser point as it appears on the camera sensor. The spot (which represents the center of the laser point on the camera sensor) is displayed as a red "x" symbol. A dropout would be displayed as a yellow dot.



See *Material* (page 75) for more information on settings for different materials.

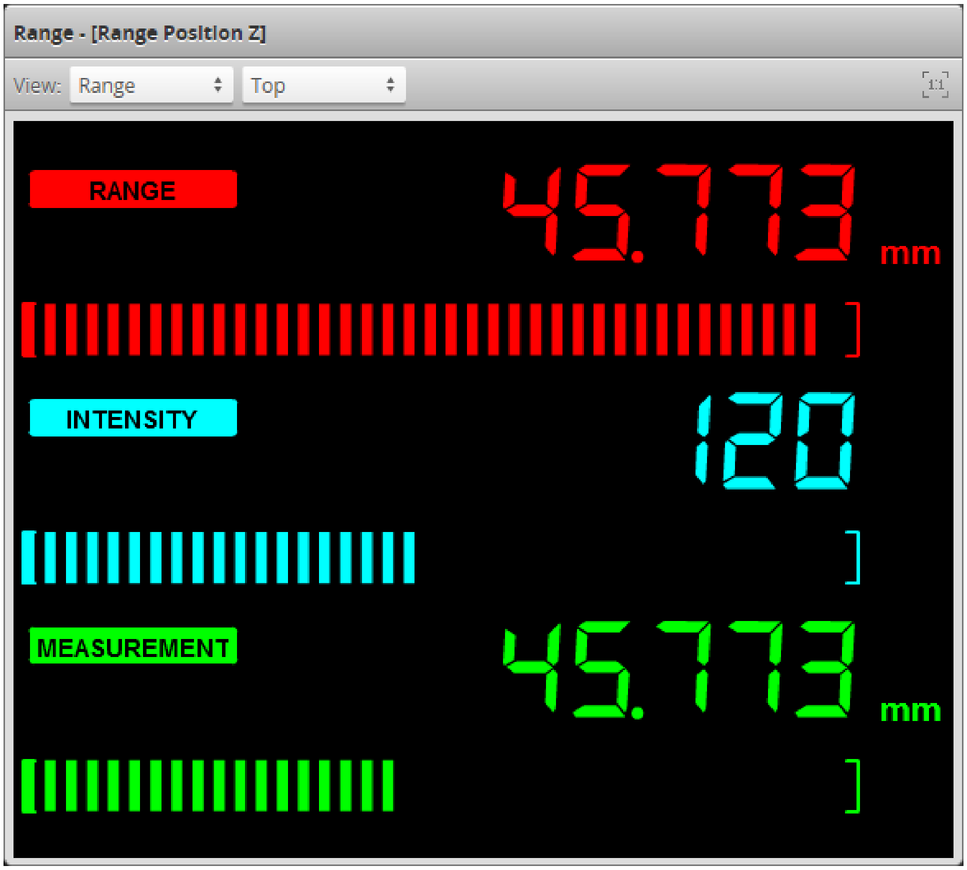
Range Mode

When the Gocator is in Range scan mode, the data viewer displays range, intensity, and measurement information as numerical values and bars. Color is used to indicate pass / fail in the case of measurement decisions.

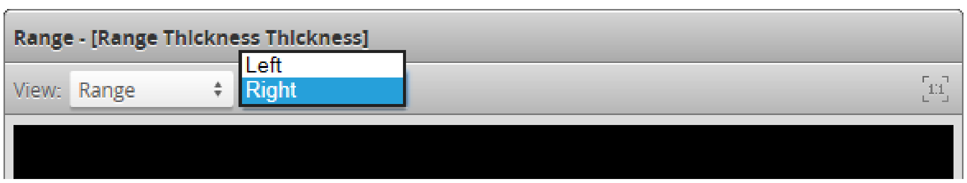
The Range value indicates where along the measurement range that the target falls. The bars indicate how close the target is to the near end (more bars, farther to the right) or the far end (fewer bars, farther to the left) of the measurement range. In the image below, the bars indicate that at 45.773 mm, the target is close to the near end of the measurement range of that sensor.

The Intensity value is on a scale of 0 to 255 and indicates the intensity at the laser point. The bars provide a graphical representation of this value. The **Acquire Intensity** option must be enabled in the **Scan Mode** panel for the Intensity value and bar to be displayed.

The Measurement value indicates the measured value of the target. If the measured value falls between the Min and Max decision values (a pass decision), the measurement and bars are green. The bars indicate graphically where, between the Min and Max decision values, the measured value falls. If the measured value falls outside the Min and Max decision values (a fail decision), the measurement is red and no bars are displayed. If the measurement is invalid, a "---" indicator is displayed instead of a value, and the bars and this indicator are red.



In a dual-sensor system, the sensor used to display ranges can be selected.



To manually select the display view in the Scan page:

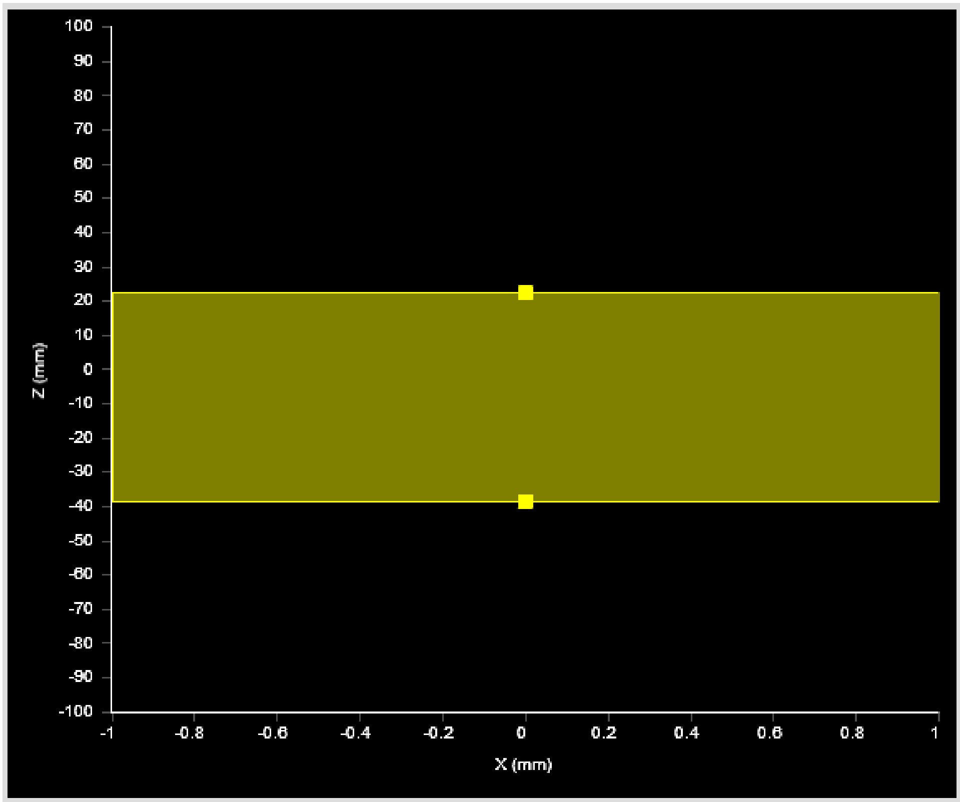
1. Go to the **Scan** page and choose **Range** mode in the **Scan Mode** panel.
2. Select the view in the data viewer.

When the **Measure** page is active, the view of the display is set to the range source of the selected measurement tool (see page 86).

Region Definition

Regions, such as an active area or a measurement region, can be graphically set up using the data viewer.

When the **Scan** page is active, the data viewer can be used to graphically configure the active area. The **Active Area** setting can also be configured manually by entering values into its fields and is found in the **Sensor** panel see page 69.



To set up a region of interest:

1. Move the mouse cursor to the rectangle.
   The rectangle is automatically displayed when a setup or measurement requires an area to be specified.
2. Drag the rectangle to move it, and use the handles on the rectangle's border to resize it.

Intensity Output

Gocator sensors can produce intensity data that measure the amount of light reflected by an object. An 8-bit intensity value is output for each range value.

Intensity output is enabled by checking the **Acquire Intensity** checkbox in the **Scan Mode** panel.

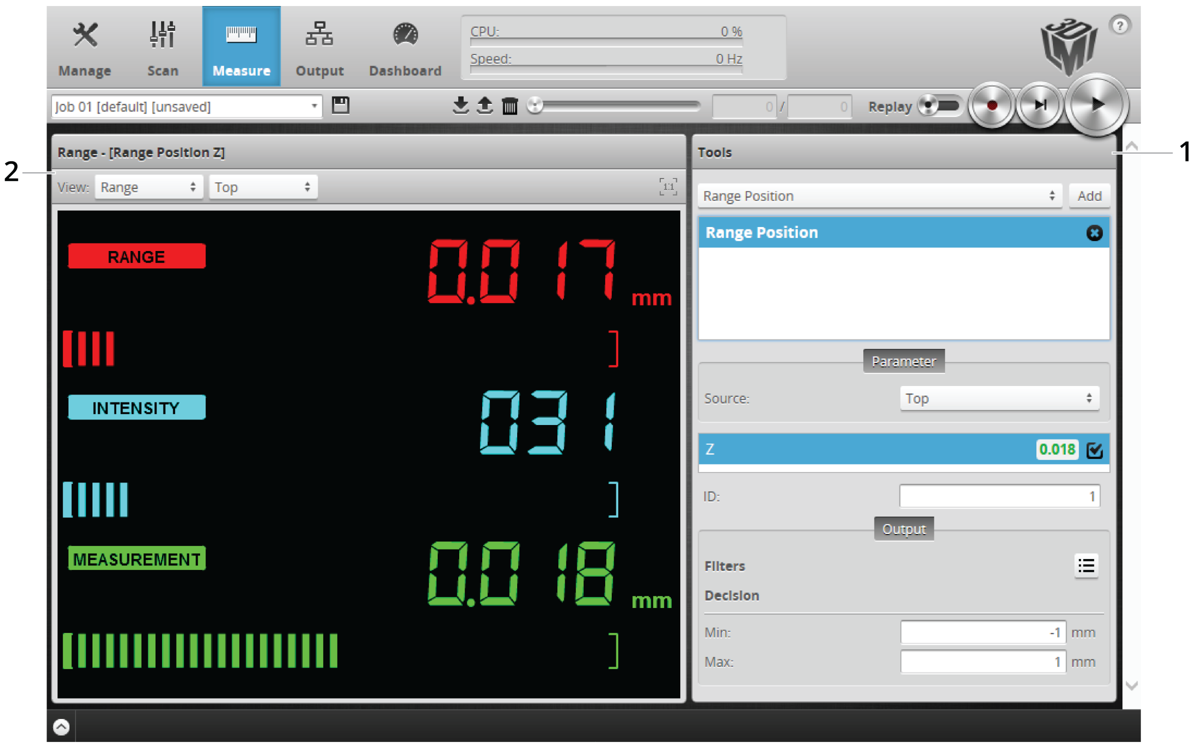
Measurement

The following sections describe the Gocator's tools and measurements.

Measure Page Overview

Measurement tools are added and configured using the **Measure** page.

The content of the **Tools** panel in the **Measure** page depends on the current scan mode. In Range mode, the **Measure** page displays tools for range measurement. In Video mode, tools are not accessible.



	Element	Description
1	Tools panel	Used to add, manage, and configure tools and measurements (see next page).
2	Data Viewer	Displays range data, sets up tools, and displays result calipers related to the selected measurement. See *Data Viewer* (below).

Data Viewer

Regions, such as active area or measurement regions, can be graphically set up using the data viewer.

When the **Measure** page is active, the data viewer can be used to graphically configure measurement regions. Measurement regions can also be configured manually in measurements by entering values into the provided fields (see page 90).

For instructions on how to set up measurement regions graphically, see page 84.

Tools Panel

The **Tools** panel lets you add, configure, and manage tools. Tools contain related measurements.

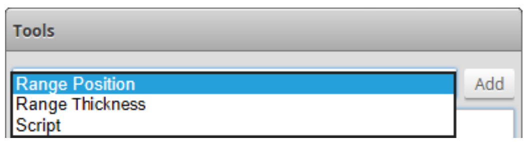
Some settings apply to tools, and therefore to all measurements, whereas some settings apply to specific measurements. See *Range Measurement* (page 94) for information on the measurement tools and their settings.

	Tool names in the user interface include the scan mode, but not in the manual. So for example, you will see "Range Position" in the user interface, but simply "Position" in the manual.

Measurement Tool Management

Adding and Removing Tools

Adding a tool adds all of the tool's measurements to the **Tools** panel, which can then be enabled and configured selectively.



To add a tool:

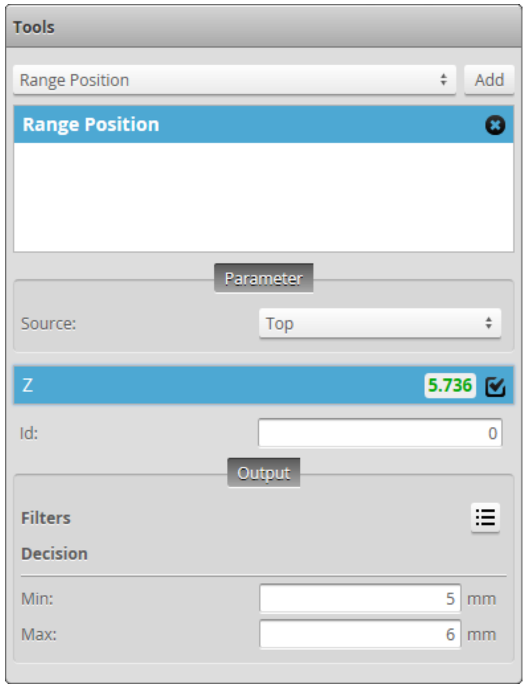
1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the Tools panel, select the tool you want to add from the drop-down list of tools.
5. Click on the **Add** button in the Tools panel.
   The tool and its available measurements will be added to the tool list. The tool parameters will be listed in the configuration area below the tool list.

To remove a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this modeone of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the tool list, click on the "x" button of the tool you want to delete.
   The tool will be removed from the tool list.

Enabling and Disabling Measurements

All of the measurements available in a tool are listed in the measurement list in the **Tools** panel after a tool has been added. To configure a measurement, you must enable it.



To enable a measurement:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurements list, check the box of the measurement you want to enable.
   The measurement will be enabled and selected. The **Output** tab, which contains output settings will be displayed below the measurements list. For some measurements, a **Parameters** tab, which contains measurement-specific parameters, will also be displayed.

To disable a measurement:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, uncheck the box of the measurement you want to disable.
   The measurement will be disabled and the **Output** tab (and the **Parameters** tab if it was available) will be hidden.

Editing a Tool or Measurement Name

You can assign a name to each tool and measurement. This allows multiple instances of tools and measurements of the same type to be more easily distinguished in the Gocator web interface. The measurement name is also referenced by the Script tool.

To edit a tool name:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the tool list, double-click on the tool name you want to change.
5. Type a new name in the ID field.
6. Press the Tab or Enter key, or click outside the name edit field.
   The name will be changed.

To change a measurement name:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this modeone of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, double-click on the measurement name you want to change.
5. Type a new name in the ID field.
6. Press the Tab or Enter key, or Click outside of the name edit field.
   The name change will be changed.

Changing a Measurement ID

The measurement ID is used to uniquely identify a measurement in the Gocator protocol or in the SDK. The value **must** be unique among all measurements.

To edit a measurement ID:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, select a measurement.
   To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* (previous page) for instructions on how to enable a measurement.

5. Click in the ID field.
6. Enter a new ID number.
   The value must be unique among all measurements.
7. Press the Tab or Enter key, or click outside the ID field.
   The measurement ID will be changed.

Common Measurement Settings

All tools provide region settings under the **Parameter** tab, and all measurements provide decision and filter settings under the **Output** tab.

Source

For dual-sensor systems, you must specify a range source for tools. The range source determines which sensor provides data for the measurement.

Depending on the layout you have selected, the **Source** drop-down will display a combination of the following. See *Dual-Sensor System Layout* (page 50) for more information on layouts.

Setting	Description
Top	Refers to the Main sensor in a standalone or dual-sensor system, the Main sensor in Opposite layout, or the combined data from both Main and Buddy sensors.
Bottom	Refers to a Buddy sensor in a dual-sensor system position in Opposite layout.
Top Left	Refers to a Main sensor in Wide layout or to a Buddy sensor in Reverse layout in a dual-sensor system position.
Top Right	Refers to a Buddy sensor in Wide layout or to a Main sensor in Reverse layout in a dual-sensor system position .

To select the source:

1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the **Tools** panel, click on a tool in the tool list.
5. Click on the **Parameter** tab in the tool configuration area.
6. Select the profile source in the **Source** drop-down list.

Regions

The Region parameter is used by tools to limit the region in which a measurement will occur. See the individual tools for details on the best way to use this parameter with each tool.

The parameter can be configured graphically using the mouse in the data viewer when the **Measure** page is active.

Some measurements use more than one region.

This parameter is also referred to as a measurement region.

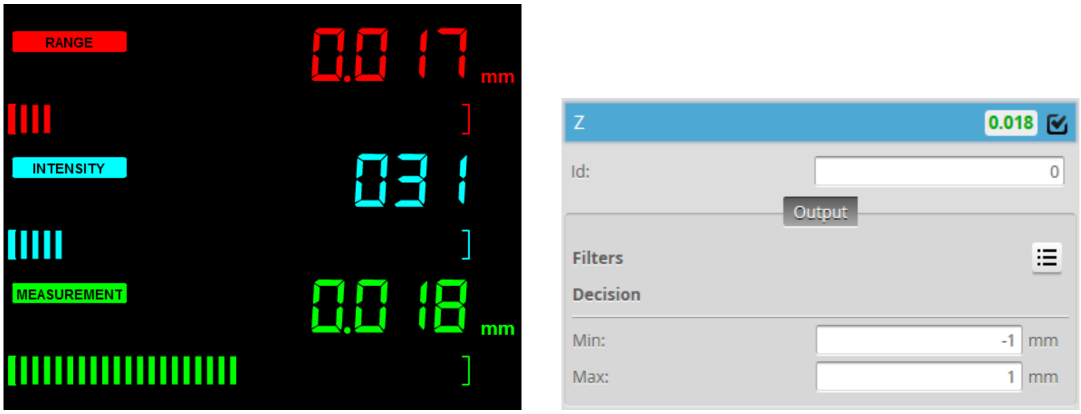
In 3D mode, the region defaults to the global field of view.

To configure regions:

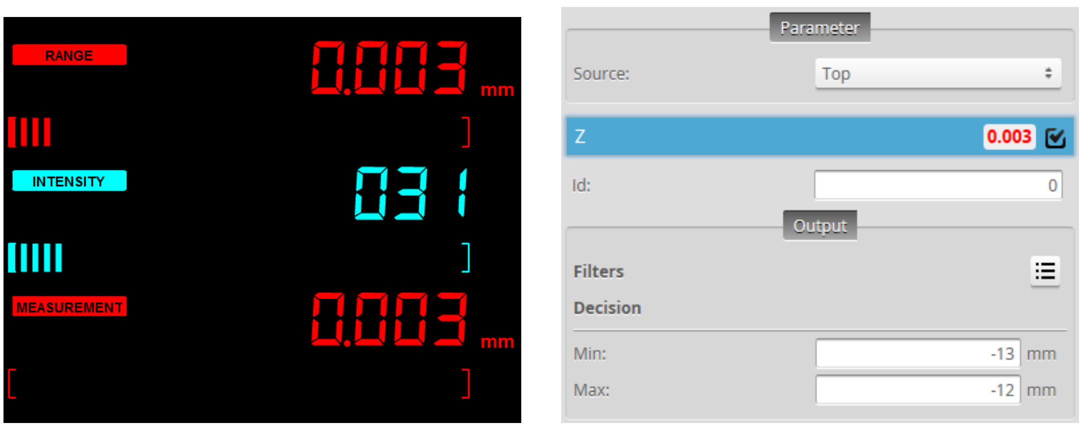
1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the **Tools** panel, click on a tool in the tool list.
5. Expand the region area by clicking on the expand button.
   Some region settings are found within other settings in this area.
6. Configure the region using the fields.
   You can also configure the region graphically using the mouse in the data viewer.

Decisions

Results from a measurement can be compared against minimum and maximum thresholds to generate pass / fail decisions. The decision state is pass if a measurement value is between the minimum and maximum threshold. In the user interface, these values are displayed in green. Otherwise, the decision state is fail. In the user interface, these values are displayed in red.



Value (5.736) within decision thresholds (Min: 5, Max: 6). Decision: Pass



Value (-13.880) outside decision thresholds (Min: -13, Max: -12). Decision: Fail

Along with measurement values, decisions can be sent to external programs and devices. In particular, decisions are often used with digital outputs to trigger an external event in response to a measurement. See *Output* (page 102) for more information on transmitting values and decisions.

To configure decisions:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the **Tools** panel, click on a tool in the tool list.
5. In the measurement list, select a measurement.
   To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* (page 88) for instructions on how to enable a measurement.
6. Click on the **Output** tab.

For some measurements, only the **Output** tab is displayed.

7. Enter values in the **Min** and **Max** fields.

Filters

Filters can be applied to measurement values before they are output from the Gocator sensors.

Filter	Description
Scale and Offset	The Scale and Offset settings are applied to the measurement value according to the following formula: Scale * Value + Offset Scale and Offset can be used to transform the output without the need to write a script. For example, to convert the measurement value from millimeters to thousands of an inch, set Scale to 39.37.
Hold Last Valid	Holds the last valid value when the measurement is invalid. Measurement is invalid if there is no valid value.
Smoothing	Applies moving window averaging to reduce random noise in a measurement output. The averaging window is configured in number of frames. If Hold Last Valid is enabled, smoothing uses the output of the Hold Last Valid filter.

To configure the filters:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the **Tools** panel, click on a tool in the tool list.
5. In the measurement list, select a measurement.
   To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* (page 88) for instructions on how to enable a measurement.
6. Click on the **Output** tab.

For some measurements, only the **Output** tab is displayed.

7. Expand the **Filters** panel by clicking on the panel header or the button.

8. Configure the filters.
   Refer to the table above for a list of the filters.

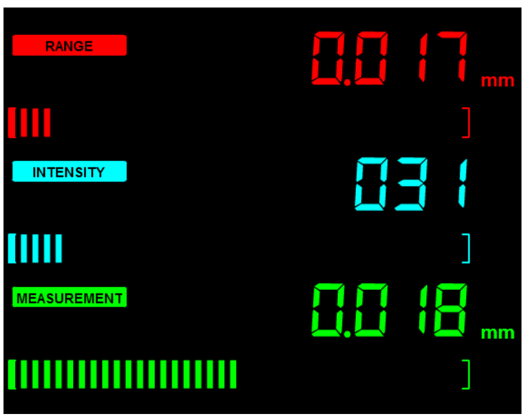
Range Measurement

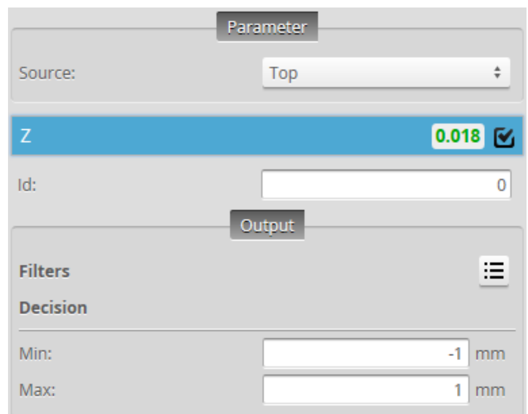
This section describes the range measurement tools available in the Gocator sensors.

Measurement Tools

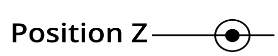
Position

The Position tool finds the Z axis position of the laser range. The measurement value can be compared with minimum and maximum constraints to yield a decision.





Measurements

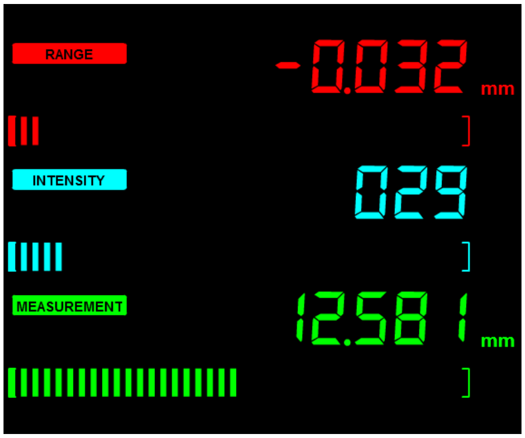
Measurement	Illustration
Position Z Determines the Z axis position of the laser range.	

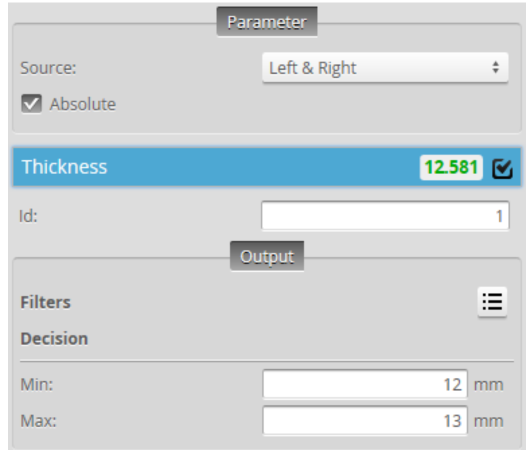
Parameters

Parameter	Description
Decision	See *Decisions* (page 91).
Output	See *Filters* (previous page).

Thickness

The Thickness tool determines the difference along the Z axis between two laser ranges. The measurement value can be compared with minimum and maximum constraints to yield a decision.

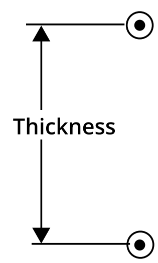




The difference can be expressed as an absolute or signed result. The difference is calculated by:

$$Thickness = Range_{Main} - Range_{Buddy}$$

Measurements

Measurement	Illustration
Thickness Determines the difference (thickness) along the Z axis between two laser ranges.	

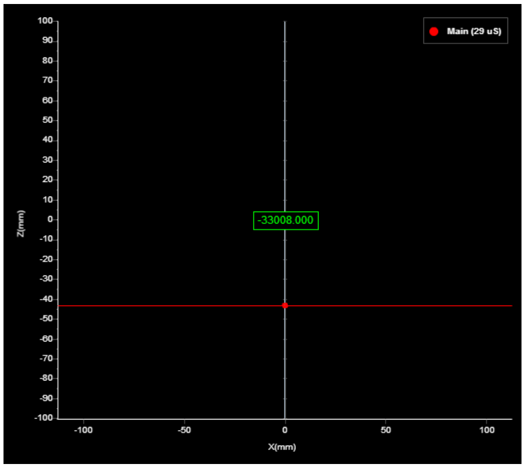
Parameters

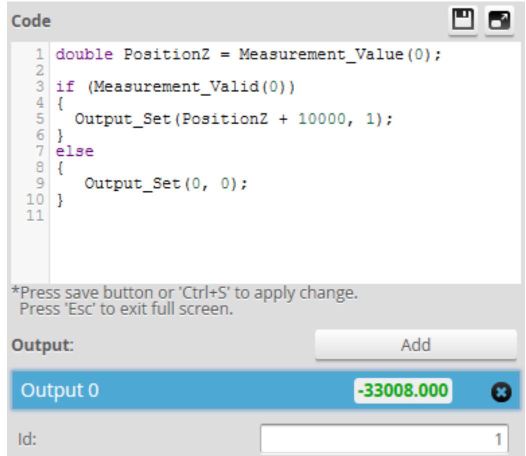
Parameter	Description
Absolute	Check the **Absolute** option to select absolute result
Decision	See *Decisions* (page 91).
Output	See *Filters* (page 93).

Script

A Script measurement can be used to program a custom measurement using a simplified C-based syntax. A script measurement can produce multiple measurement values and decisions for the output.

See *Adding and Removing Tools* (page 87) for instructions on how to add measurement tools.





See *Script Measurement* (below) for more information on the script syntax.

To create or edit a Script measurement:

1. Add a new Script tool or select an existing Script measurement.
2. Edit the script code.
3. Add script outputs using the **Add** button.
   For each script output that is added, an index will be added to the **Output** drop-down and a unique ID will be generated.
   To remove a script output, click on the button next to it.
4. Click the **Save** button to save the script code.
   If there is a mistake in the script syntax, the result will be shown as a "Invalid" with a red border in the data viewer when you run the sensor.
   Outputs from multiple measurement tools can be used as inputs to the script. A typical script would take results from other measurement tools using the value and decision function, and output the result using the output function. Stamp information, such as time and encoder stamps, are available in the script, whereas the actual data is not. (The script engine is not powerful enough to process the data itself.) Only one script can be created.

Script Measurement

A Script measurement can be used to program a custom measurement using a simplified C-based syntax. Similar to other measurement tools, a script measurement can produce multiple measurement values and decisions for the output.

The following elements of the C language are supported:

Supported Elements

Elements	Supported
Control Operators	if, while, do, for, switch and return.
Data Types	char, int, unsigned int, float, double, long long (64-bit integer).
Arithmetic and Logical Operator	Standard C arithmetic operators, except ternary operator (i.e., "condition? trueValue: falseValue"). Explicit casting (e.g., int a = (int) a_float) is not supported.
Function Declarations	Standard C function declarations with argument passed by values. Pointers are not supported.

Built-in Functions

Measurement Functions

Function	Description
int Measurement_Exists(int id)	Tests for the existence of a measurement by ID. Parameters: id – Measurement ID Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Valid(int id)	Determines if a measurement value is valid by its ID. Parameters: id - Measurement ID Returns 0 - Measurement is invalid 1 - Measurement is valid
double Measurement_Value (int id)	Gets the value of a measurement by its ID. Parameters: id - Measurement ID Returns: Value of the measurement 0 – if measurement does not exist 1 – if measurement exists
int Measurement_Decision (int id)	Gets the decision of a measurement by its ID. Parameters: ID - Measurement ID Returns: Decision of the measurement 0 – if measurement decision is false 1 – If measurement decision is true
int Measurement_NameExists(char* toolName, char* measurementName)	Determines if a measurement exist by name. Parameter: toolName – Tool name measurementName – Measurement name

Function	Description
	Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Id (char* toolName, char* measurementName)	Gets the measurement ID by the measurement name. Parameters: toolName – Tool name measurementName – Measurement name Returns: -1 – measurement does not exist Other value – Measurement ID

Output Functions

Function	Description
void Output_Set (double value, int decision)	Sets the output value and decision on Output index 0. Only the last output value / decision in a script run is kept and passed to the Gocator output. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetAt(0, INVALID_VALUE, 0)) Parameters: value - value output by the script decision - decision value output by the script. Can only be 0 or 1
void Output_SetAt(unsigned int index, double value, int decision)	Sets the output value and decision at the specified output index. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetAt(0, INVALID_VALUE, 0)) Parameters: index – Script output index value – value output by the script decision – decision value output by the script. Can only be 0 or 1
void Output_SetId(int id, double value, int decision)	Sets the output value and decision at the specified script output ID. To output an invalid value, the constant INVALID_VALUE can be used (e.g., Output_SetId(0, INVALID_VALUE, 0)) Parameters: id – Script output ID

Memory Functions

Function	Description
void Memory_Set64s (int id, long long value)	Stores a 64-bit signed integer in persistent memory. Parameters: id - ID of the value value - Value to store
long long Memory_Get64s (int id)	Loads a 64-bit signed integer from persistent memory. Parameters: id - ID of the value

Function	Description
	Returns: value - Value stored in persistent memory
void Memory_Set64u (int id, unsigned long long value)	Stores a 64-bit unsigned integer in the persistent memory Parameters: id - ID of the value value - Value to store
unsigned long long Memory_Get64u (int id)	Loads a 64-bit unsigned integer from persistent memory. Parameters: id - ID of the value Returns: value - Value stored in persistent memory
void Memory_Set64f (int id, double value)	Stores a 64-bit double into persistent memory. Parameters: id - ID of the value value - Value to store
double Memory_Get64f (int id)	Loads a 64-bit double from persistent memory. All persistent memory values are set to 0 when the sensor starts. Parameters: id - ID of the value Returns: value - Value stored in persistent memory
int Memory_Exists (int id)	Tests for the existence of a value by ID. Parameters: id – Value ID Returns: 0 – value does not exist 1 – value exists
void Memory_Clear (int id)	Erases a value associated with an ID. Parameters: id – Value ID
void Memory_ClearAll()	Erases all values from persistent memory

Stamp Functions

Function	Description
long long Stamp_Frame()	Gets the frame index of the current frame.
long long Stamp_Time()	Gets the time stamp of the current frame.
long long Stamp_Encoder()	Gets the encoder position of the current frame.
long long Stamp_EncoderZ()	Gets the encoder index position of the current frame.
unsigned int Stamp_Inputs()	Gets the digital input state of the current frame.

Math Functions

Function	Description
float sqrt(float x)	Calculates square root of x
float sin(float x)	Calculates sin(x) (x in radians)
float cos(float x)	Calculates cos(x) (x in radians)
float tan(float x)	Calculates tan(x) (x in radians)
float asin(float x)	Calculates asin(x) (x in radians)
float acos(float x)	Calculates acos(x) (x in radians)
float atan(float x)	Calculates atan(x) (x in radians)
float pow (float x, float y)	Calculates the exponential value. x is the base, y is the exponent
float fabs(float x)	Calculates the absolute value of x

Example: Accumulated Length

The following example shows how to create a custom measurement that is based on the values from other measurements and persistent values. The example calculates the length of the target using a series of position Z measurement tool values (Measurement ID 1)

```
/* Encoder Spacing is 0.5mm */
/* Z position measurement ID is set to 1 */


long long encoder_spacing = 500;
long long length = Memory_Get64s(0);


if (Measurement_Valid(1))
{
   length = length + encoder_spacing;
}
else
{
   length = 0;
}

Memory_Set64s(0, length);


if (length > 10000)
{
   Output_Set(length, 1);
}
else
{
   Output_Set(length, 0);
```

```
}
```

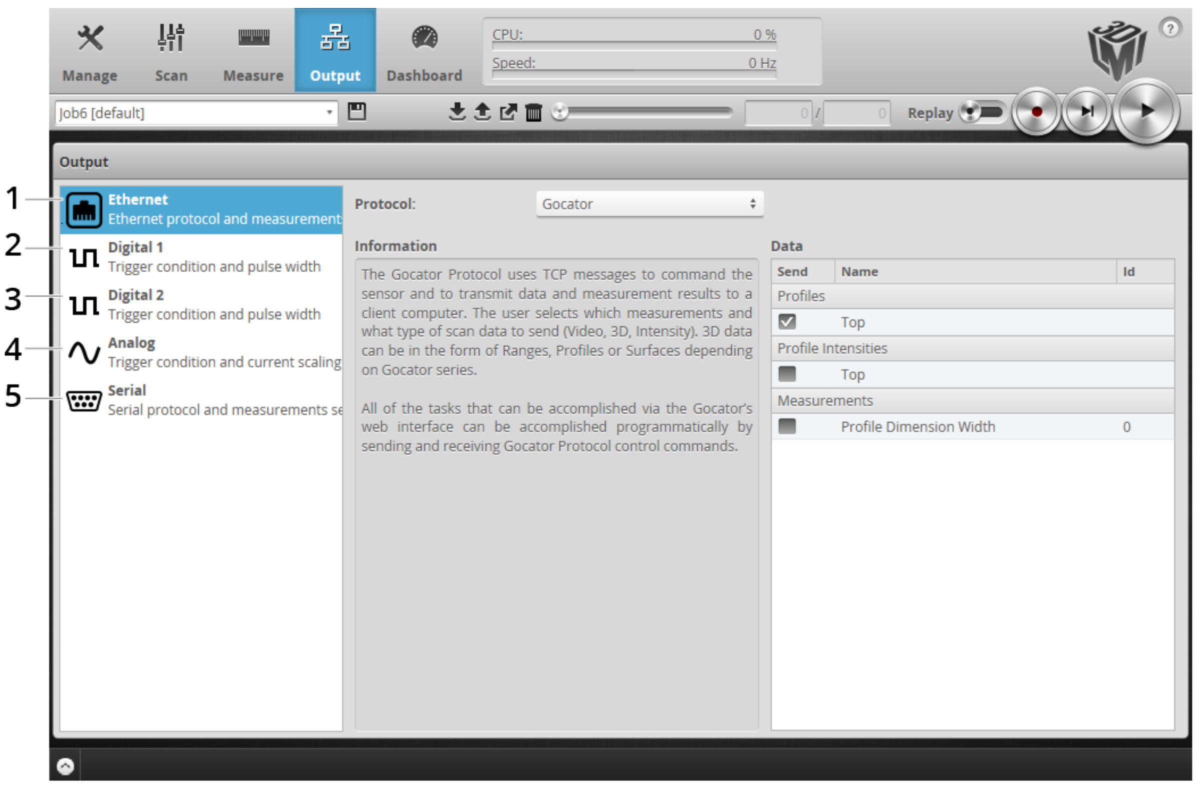
Output

The following sections describe the **Output** page.

Output Page Overview

Output configuration tasks are performed using the **Output** page. Gocator sensors can transmit laser ranges and measurement results to various external devices using several output interface options.

> Up to two outputs can have scheduling enabled with ASCII as the Serial output protocol. When Selcom is the current Serial output protocol, only one other output can have scheduling enabled.

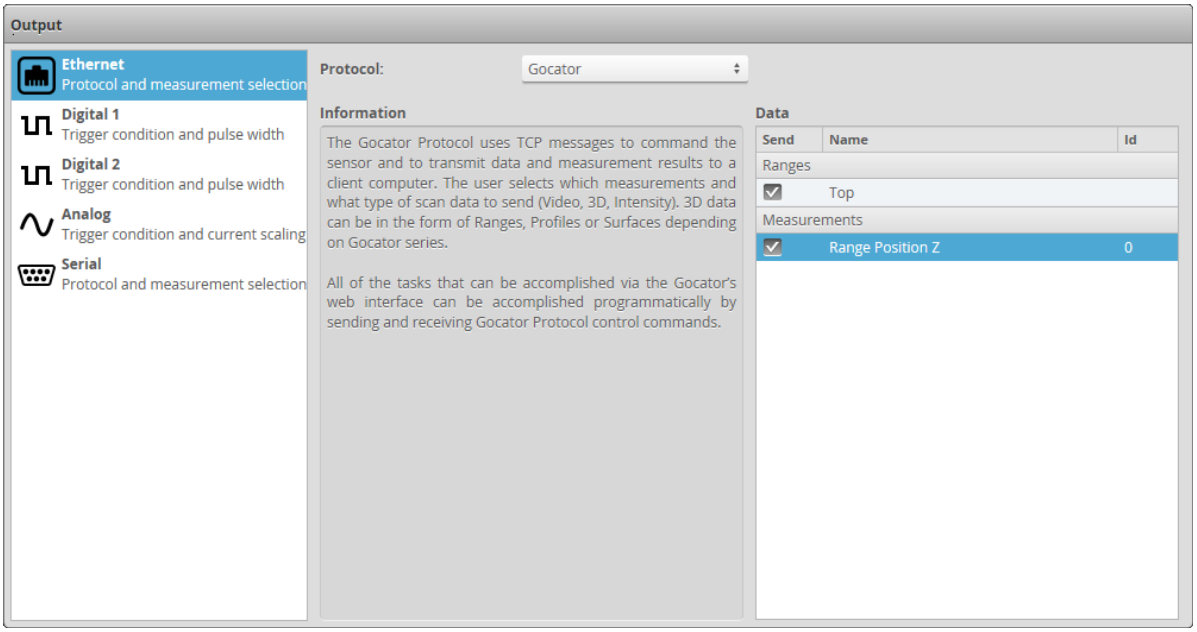


	Category	Description
1	Ethernet	Used to select the data sources that will transmit data via Ethernet. See *Ethernet Output* (next page).
2	Digital Output 1	Used to select the data sources that will be combined to produce a digital output pulse on Output 1. See *Digital Output* (page 107).
3	Digital Output 2	Used to select the data sources that will be combined to produce a digital output pulse on Output 2. See *Digital Output* (page 107).
4	Analog Panel	Used to convert a measurement value or decision into an analog output signal. See *Analog Output* (page 109).
5	Serial Panel	Used to select the measurements that will be transmitted via RS-485 serial output. See *Serial Output* (page 111).

Ethernet Output

A sensor uses TCP messages (Gocator protocol) to receive commands from client computers, and to send video, laser range, intensity, and measurement results to client computers. The sensor can also receive commands from and send measurement results to a PLC using ASCII, Modbus TCP, or EtherNet/IP protocol. See *Protocols* (page 137) for the specification of these protocols.

The specific protocols used with Ethernet output are selected and configured within the panel.

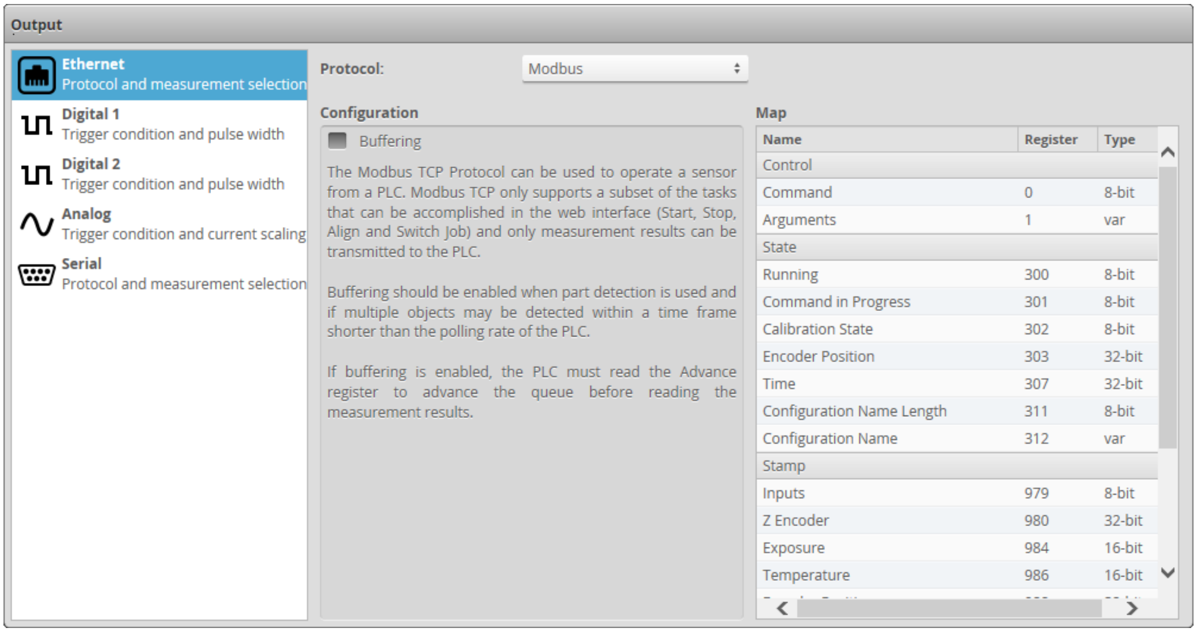


To receive commands and send results using Gocator Protocol messages:

1. Go to the **Output** page.
2. Click on the **Ethernet** category in the **Output** panel.
3. Select **Gocator** as the protocol in the **Protocol** drop-down.
4. Check the video, range, intensity, or measurement items to send.

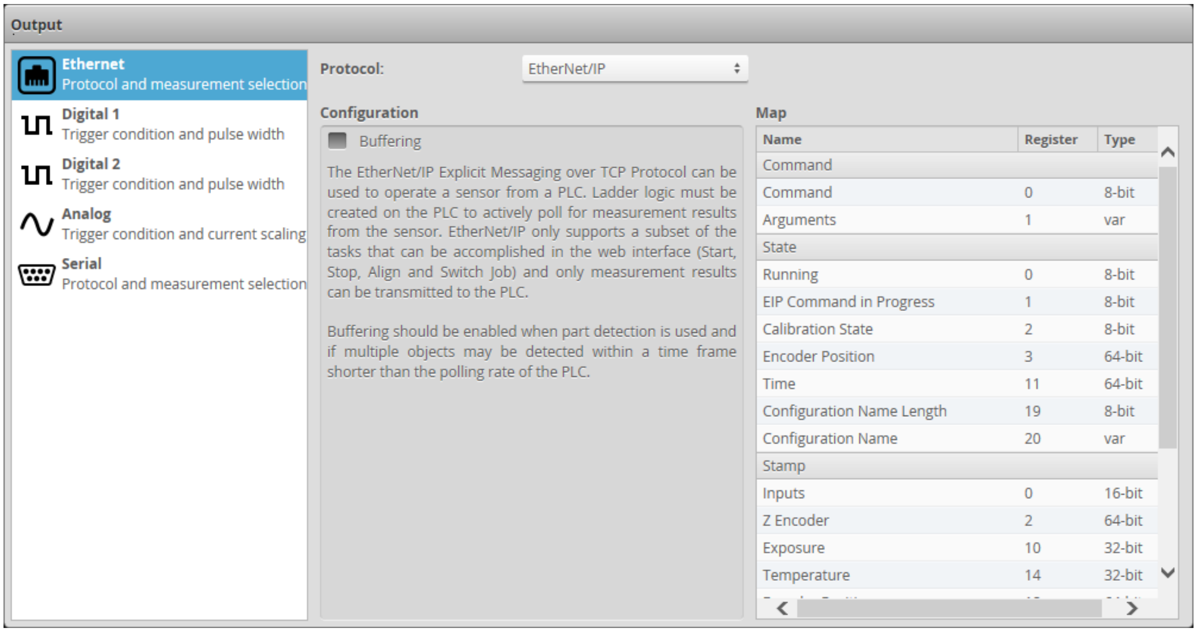
Measurements shown here correspond to measurements that have been added using the **Measure** page (see page 86).

All of the tasks that can be accomplished with the Gocator's web interface (creating jobs, performing alignment, sending data and health information, and software triggering, etc.) can be accomplished programmatically by sending Gocator protocol control commands.



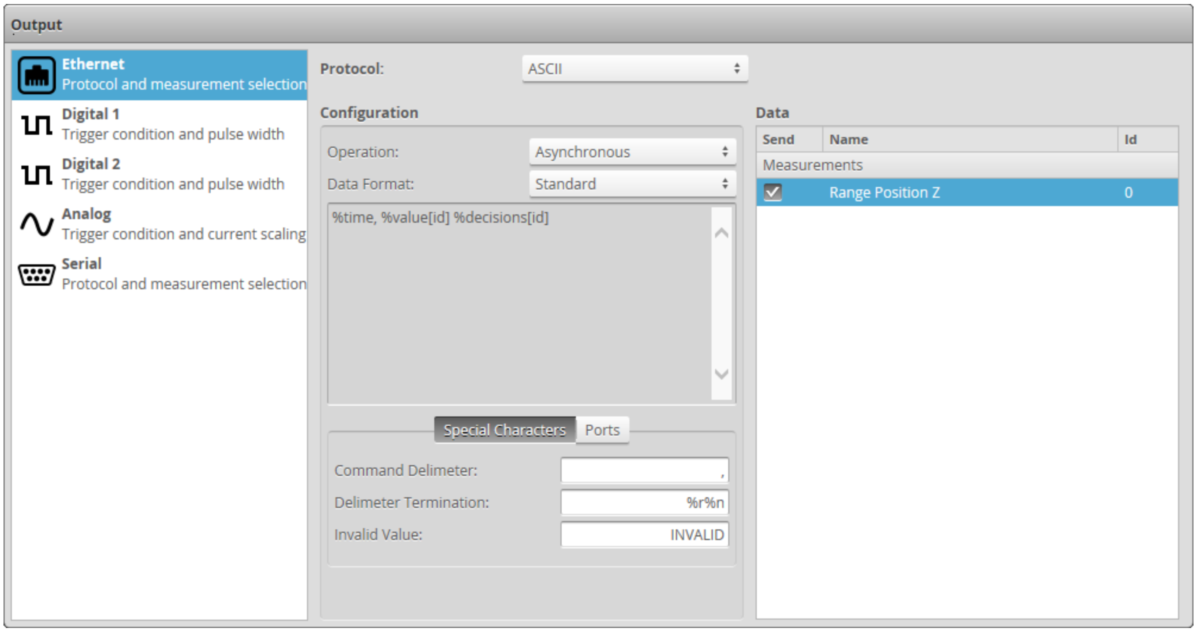
To receive commands and send results using Modbus TCP messages:

1. Go to the **Output** page.
2. Click on **Ethernet** in the **Output** panel.
3. Select **Modbus** as the protocol in the **Protocol** drop-down.
   Unlike the Gocator Protocol, you do not select which measurement items to output. The Ethernet panel will list the register addresses that are used for Modbus TCP communication.
   The Modbus TCP protocol can be used to operate a sensor. Modbus TCP only supports a subset of the tasks that can be performed in the web interface. A sensor can only process Modbus TCP commands when Modbus is selected in the **Protocol** drop-down.
4. Check the **Buffering** checkbox, if needed.
   Buffering is needed, for example, in Surface mode if multiple objects are detected within a time frame shorter than the polling rate of the PLC.
   If buffering is enabled with the Modbus protocol, the PLC must read the Advance register to advance the queue before reading the measurement results.



To receive commands and send results using EtherNet/IP messages:

1. Go to the **Output** page.
2. Click on **Ethernet** in the **Output** panel.
3. Select **EtherNet/IP** in the **Protocol** option.
   Unlike using the Gocator Protocol, you don't select which measurement items to output. The **Ethernet** panel will list the register addresses that are used for EtherNet/IP messages communication.
   The EtherNet/IP protocol can be used to operate a sensor. EtherNet/IP only supports a subset of the tasks that can be accomplished in the web interface. A sensor can only process EtherNet/IP commands when the EtherNet/IP is selected in the **Protocol** option.
4. Check the **Buffering** checkbox, if needed.
   Buffering is needed, for example, in Surface mode if multiple objects are detected within a time frame shorter than the polling rate of the PLC. If buffering is enabled with the EtherNet/IP protocol, the buffer is automatically advanced when the Sample State Assembly Object (see page 184) is read.



To receive commands and send results using ASCII messages:

1. Go to the **Output** page.

2. Click on **Ethernet** in the **Output** panel.

3. Select **ASCII** as the protocol in the **Protocol** drop-down.

4. Set the operation mode in the **Operation** drop-down.
   In asynchronous mode, the data results are transmitted when they are available. In polling mode, users send commands on the data channel to request the latest result. See *Asynchronous and Polling Operation* (page 187) for an explanation of the operation modes.

5. Select the data format from the **Data Format** drop-down.
   Select **Standard** to use the default result format of the ASCII protocol. Select the measurement to send by placing a check in the corresponding checkbox. See *Standard Result Format* (page 188) for an explanation of the standard result mode.
   Select **Custom** to enable the custom format editor, and then use the replacement patterns listed in **Replacement Patterns** to create a custom format in the editor.

6. Set the special characters in the **Special Characters** tab.
   Set the command delimiter, delimiter termination, and invalid value characters. Special characters are used in commands and standard-format data results.

7. Set the TCP ports in the **Ports** tab.
   Select the TCP ports for the control, data, and health channels. If the port numbers of two channels are the same, the messages for both channels are transmitted on the same port.

Digital Output

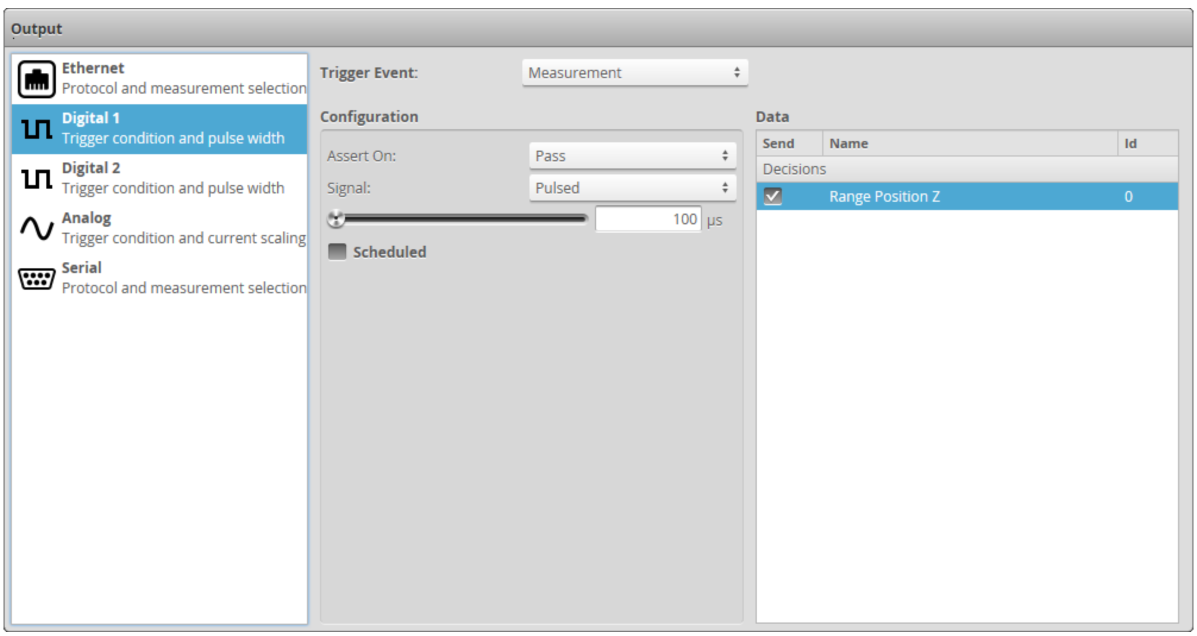
Gocator sensors can convert measurement decisions or software commands to digital output pulses, which can then be used to output to a PLC or to control external devices, such as indicator lights or air ejectors.

A digital output can act as a measurement valid signal to allow external devices to synchronize to the timing at which measurement results are output. In this mode, the sensor outputs a digital pulse when a measurement result is ready.

A digital output can also act as a strobe signal to allow external devices to synchronize to the timing at which the sensor exposes. In this mode, the sensor outputs a digital pulse when the sensor exposes.

Each sensor supports two digital output channels. See *Digital Outputs* (page 236) for information on wiring digital outputs to external devices.

Trigger conditions and pulse width are then configured within the panel.



To output measurement decisions:

1. Go to the **Output** page.
2. Click on **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. In **Configuration**, set **Assert On** and select the measurements that should be combined to determine the output.
   If multiple measurement decisions are selected and **Assert On** is set to **Pass**, the output is activated when all selected measurements pass.
   If **Assert On** is set to **Fail**, the output is activated when any one of the selected measurements fails.

5. Set the **Signal** option.
   The signal type specifies whether the digital output is a continuous signal or a pulsed signal. If **Signal** is set to **Continuous**, the signal state is maintained until the next transition occurs. If **Signal** is set to is **Pulsed**, you must specify the pulse width and how it is scheduled.
6. Specify a pulse width using the slider.
   The pulse width is the duration of the digital output pulse, in microseconds.
7. Specify whether the output is immediate or scheduled.
   Check the **Scheduled** option if the output needs to be scheduled.
   A scheduled output becomes active after a specified delay from the start of Gocator exposure. A scheduled output can be used to track the decisions for multiple objects as these objects travel from the sensor to the eject gates. The **Delay** setting specifies the distance from the sensor to the eject gates.
   An immediate output becomes active as soon as measurement results are available. The output activates after the sensor finishes processing the data. As a result, the time between the start of sensor exposure and output activates can vary and is dependent on the processing latency. The latency is reported in the dashboard and in the health messages.
8. Specify a delay.
   The delay specifies the time or spatial location between the start of sensor exposure and when the output becomes active. The delay should be larger than the time needed to process the data inside the sensor. It should be set to a value that is larger than the processing latency reported in the dashboard or in the health messages.
   The unit of the delay is configured with the **Delay Domain** setting.

To output a measurement valid signal:

1. Go to the **Output** page.
2. Click on **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. In **Configuration**, set **Assert On** to **Always**.
5. Select the measurements.
   The output activates when the selected decisions produce results. The output activates only once for each frame even if multiple decision sources are selected.
6. Specify a pulse width using the slider.
   The pulse width determines the duration of the digital output pulse, in microseconds.

To respond to software scheduled commands:

1. Go to the **Output** page.
2. Click on **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Software**.

4. Specify a **Signal** type.
   The signal type specifies whether the digital output is a continuous signal or a pulsed signal. If the signal is continuous, its state is maintained until the next transition occurs. If the signal is pulsed, user specifies the pulse width and the delay.
5. Specify a **Pulse Width**.
   The pulse width determines the duration of the digital output pulse, in microseconds.
6. Specify if the output is **Immediate** or **Scheduled**.
   A pulsed signal can become active immediately or scheduled. Continuous signal always becomes active immediately.
   Immediate output becomes active as soon as a scheduled digital output (see page 154) is received.
   Scheduled output becomes active at a specific target time or position, given by the Scheduled Digital Output command. Commands that schedule event in the past will be ignored. An encoder value is in the future if the value will be reached by moving in the forward direction (the direction that encoder calibration was performed in).

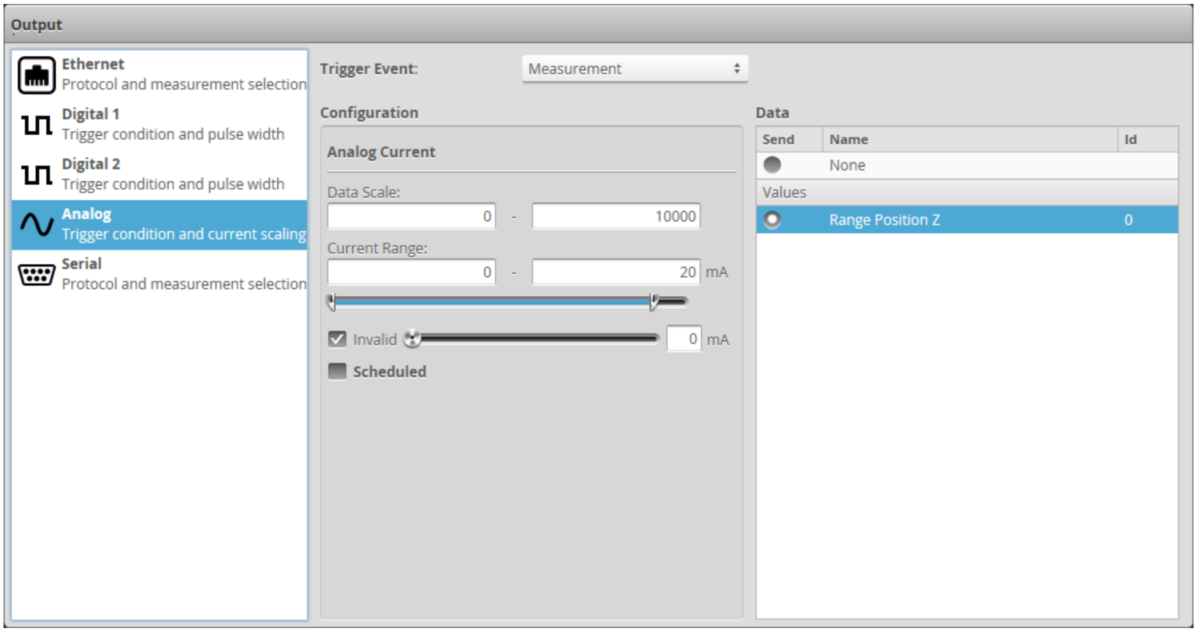
To output an exposure signal:

1. Go to the **Output** page.
2. Click on **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Exposure**.
4. Set the **Pulse Width** option.
   The pulse width determines the duration of the digital output pulse, in microseconds.

Analog Output

Gocator sensors can convert a measurement result or software request to an analog output. Each sensor supports one analog output channel.

See *Analog Output* (page 239) for information on wiring analog output to an external device.



To output measurement value or decision:

1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Measurement**.
4. Select the measurement that should be used for output.
   Only one measurement can be used for analog output. Measurements shown here correspond to measurements that have been programmed using the **Measurements** page.
5. Specify **Data Scale** values.
   The values specified here determine how measurement values are scaled to the minimum and maximum current output. The **Data Scale** values are specified in (um) for dimensional measurement, (0.001 mm2) for area, (mm3) for volume, and (0.001 degree) for angle results.
   The values specified here determine the minimum and maximum current values in milliamperes. The invalid current value is used when a measurement value is not valid. If invalid is not checked, the output holds the last value when a measurement value is not valid.
6. Specify **Current Range** and **Invalid** current values.
   The values specified here determine the minimum and maximum current values in milliamperes. If **Invalid** is checked, the current value specified with the slider is used when a measurement value is not valid. If **Invalid** is not checked, the output holds the last value when a measurement value is not valid.
7. Specify if the output is immediate or scheduled.
   An analog output can become active immediately or scheduled. Check the **Scheduled** option if the output needs to be scheduled.
   A scheduled output becomes active after a specified delay from the start of Gocator exposure. A scheduled output can be used to track the decisions for multiple objects as these objects travel from

the sensor to the eject gates. The delay specifies the distance from the sensor to the eject gates.

An Immediate output becomes active as soon as the measurement results are available. The output activates after the Gocator finishes processing the data. As a result, the time between the start of Gocator exposure and output activates depends on the processing latency. The latency is reported in the dashboard and in the health messages.

8. Specify a delay.

   The delay specifies the time or spatial location between the start of Gocator exposure and the output becomes active. The delay should be larger than the time needed to process the data inside the Gocator. It should be set to a value that is larger than the processing latency reported in the dashboard and in the health messages.

   The unit of the delay is configured in the trigger panel. See *Triggers* (page 64) for details.

> The analog output takes about 75 us to reach 90% of the target value for a maximum change, then another ~40 us to settle completely.

To respond to software scheduled commands:

1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Software**.
4. Specify if the output is immediate or scheduled.

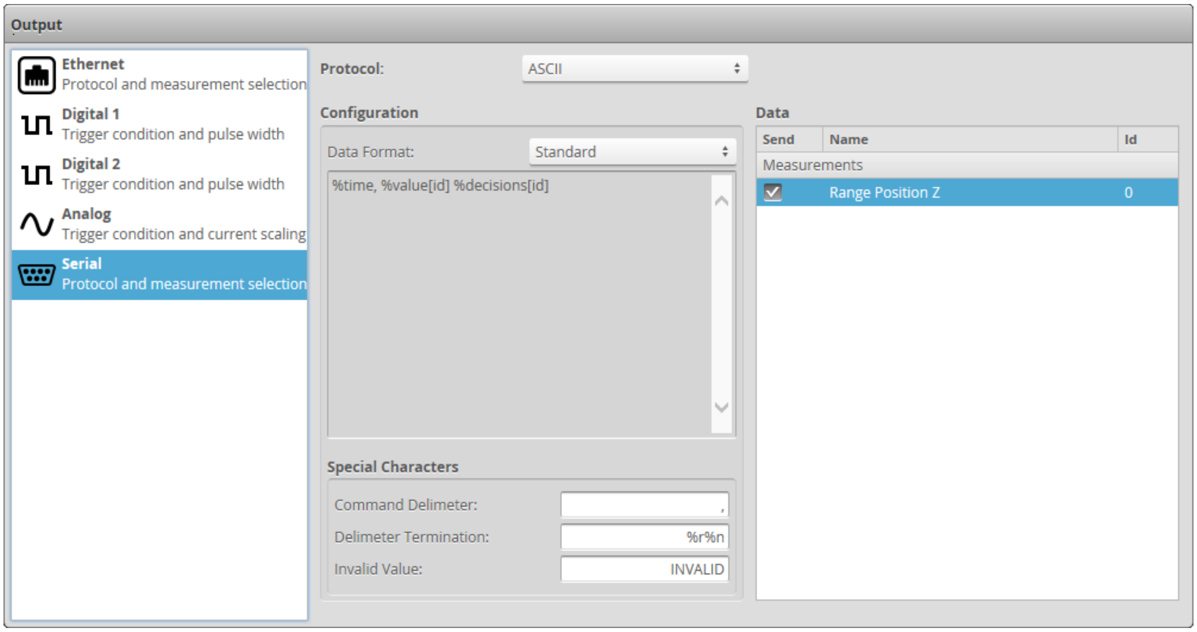
   An analog output value becomes active immediately or scheduled. Immediate output becomes active as soon as a Scheduled Analog Output command (see page 154) is received.

   Software scheduled command can schedule an analog value to output at a specified future time or encoder value, or changes its state immediately. The Delay setting in the panel is ignored. Commands that schedule event in the past will be ignored. An encoder value is in future if the value will be reached by moving in the forward direction (the direction that encoder calibration was performed in).

Serial Output

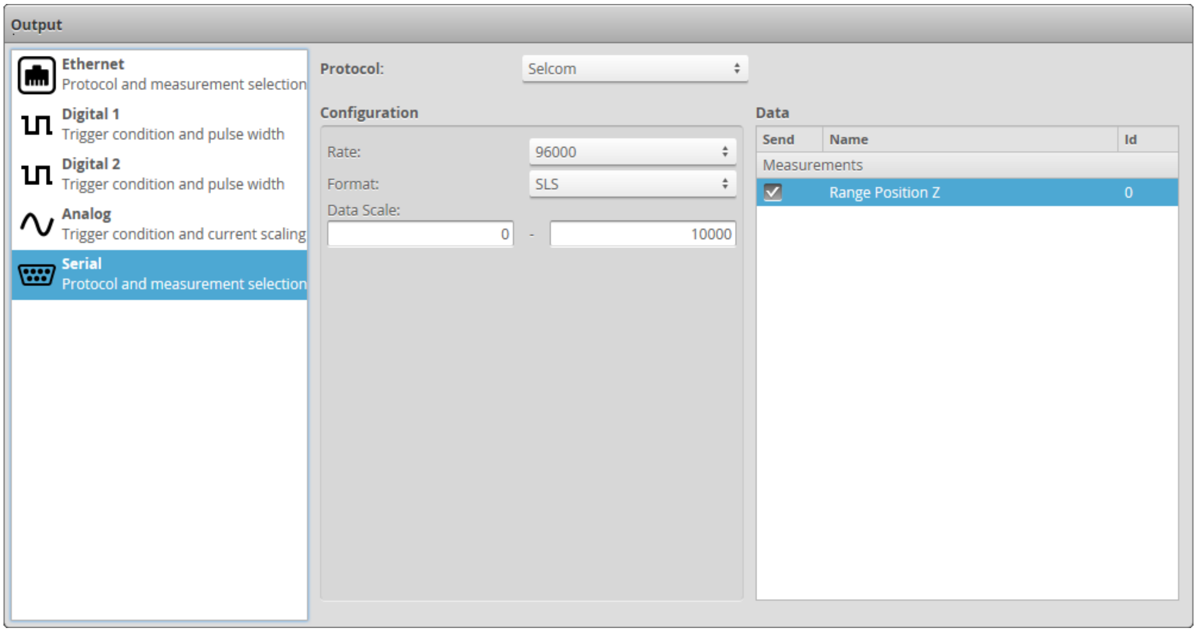
The Gocator's web interface can be used to select measurements to be transmitted via RS-485 serial output. Each sensor has one serial output channel.

Two protocols are supported: ASCII Protocol and Selcom Serial Protocol. The ASCII protocol outputs data asynchronously using a single serial port. The Selcom Serial Protocol outputs synchronized serial data using two serial ports. See *ASCII Protocol* (page 187) for the ASCII Protocol parameters and data formats. See *Serial Protocol* (page 197) for the Selcom serial protocol and data formats.



To exchange results using ASCII messages:

1. Go to the **Output** page.
2. Click on **Serial** in the **Output** panel.
3. Select **ASCII** in the **Protocol** option.
4. Select the **Data Format**.
   Select **Standard** to use the default result format of the ASCII protocol. Select value and decision to send by placing a check in the corresponding check box. See *Standard Result Format* (page 188) for an explanation of the standard result mode.
   Select **Custom** to customize the output result. A data format box will appear in which you can type the format string. See *Custom Result Format* (page 189) for the supported format string syntax.
5. Select the measurments to send.
   Select measurements by placing a check in the corresponding check box.
6. Set the **Special Characters**.
   Select the delimiter, termination and invalid value characters. Special characters are used in commands and standard-format data results.



To configure Selcom serial output:

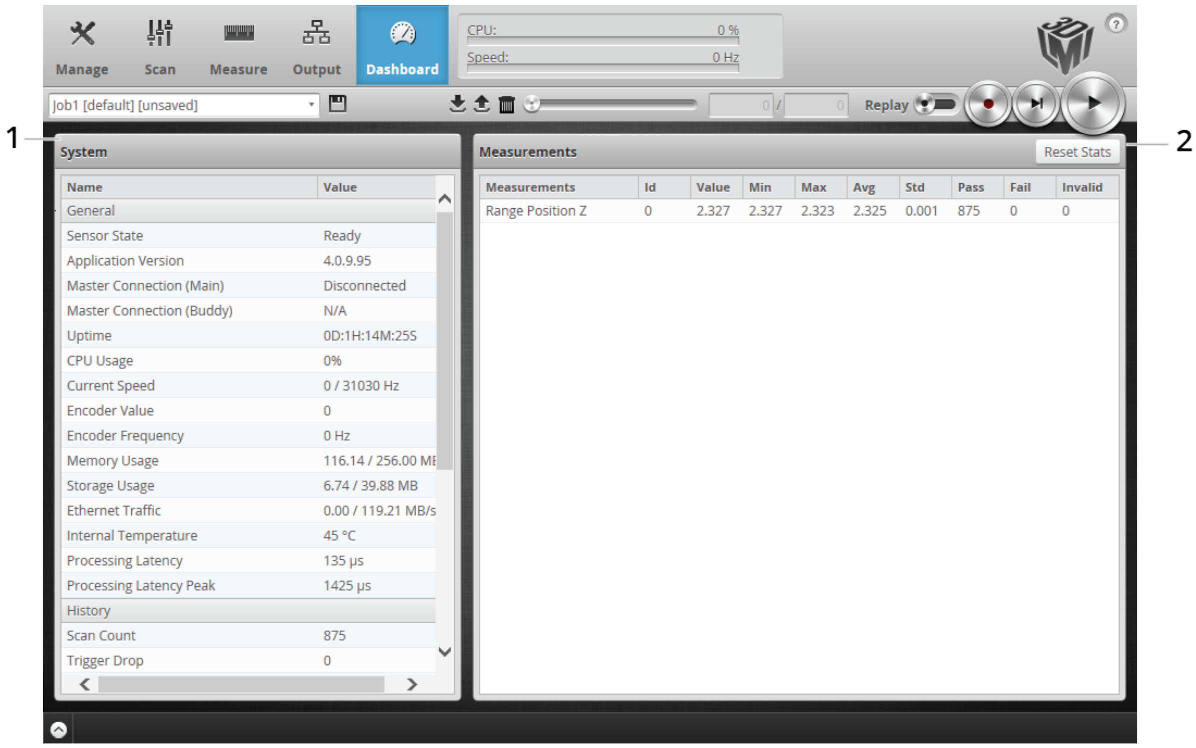
1. Go to the **Output**page.
2. Click on **Serial** in the **Output** panel.
3. Select **Selcom** in the **Protocol** option.
4. Select the measurements to send.
   To select an item for transmission, place a check in the corresponding check box. Measurements shown here correspond to measurements that have been programmed using the **Measurements** page.
5. Select the **Output** speed.
6. Specify **Data Scale** values.
   The values specified here determine how measurement values are scaled to the minimum and maximum current output. The **Data Scale** value is specified in (um) for dimensional measurement, (0.001 mm^2) for area, (mm^3) for volume and (0.001 degree) for angle results.
   The results are scaled according to the number of serial bits used to cover the data scale range. For example, the 12-bit output would break a 200 mm data scale range into 4096 increments (0.0488 mm/bit), and the 14-bit output would break a 200 mm data scale range into 16384 increments (0.0122 mm/bit).
7. Specify the **Data Format**.
   See *Serial Protocol* (page 197) for definitions of the formats.

Dashboard

The following sections describe the **Dashboard** page.

Dashboard Page Overview

The **Dashboard** page summarizes sensor health information, and measurement statistics.



	Element	Description
1	System	Displays sensor state and health information. See *System Panel* (below).
2	Measurements	Displays measurement statistics. See *Measurements* (next page).

System Panel

The following state and health information is available in the **System** panel on the **Dashboard** page:

Dashboard General System Values

Name	Description
Sensor State	Current sensor state (Ready or Running).
Application Version	Gocator firmware version.
Master Connection	Whether Master is connected.
Uptime	Length of time since the sensor was power-cycled or reset.
CPU Usage	Sensor CPU utilization (%).

Name	Description
Encoder Value	Current encoder value (ticks).
Encoder Frequency	Current encoder frequency (Hz).
Memory Usage	Sensor memory utilization (MB used / MB total available).
Storage Usage	Sensor flash storage utilization (MB used / MB total available).
Temperature	Sensor internal temperature (C).
Ethernet Traffic	Network output utilization (MB/sec).
Internal Temperature	Internal sensor temperature.
Processing Latency	Last delay from camera exposure to when results can be scheduled to.
Processing Latency Peak	Peak latency delay from camera exposure to when results can be scheduled to Rich I/O. Reset on start.

Dashboard History Values

Name	Description
Scan Count	Number of scans performed since sensor state last changed to Running.
Trigger Drop	Count of camera frames dropped due to excessive trigger speed.
Analog Output Drop	Count of analog output drops because last output has not been completed.
Digital Output Drop	Count of digital output drops because last output has not been completed.
Serial Output Drop	Count of serial output drops because last output has not been completed.
Processing Drop	Count of frame drops due to excessive CPU utilization.
Ethernet Drop	Count of frame drops due to slow Ethernet link.
Digital Output High Count	Count of high states on digital outputs.
Digital Output Low Count	Count of low states on digital outputs.
Range Valid Count	Count of valid ranges.
Range Invalid Count	Count of invalid ranges.
Anchor Invalid Count	Count of invalid anchors.
Valid Spot Count	Count of valid spots detected in the last frame.
Max Spot Count	Maximum number of spots detected since sensor was started.
Camera Search Count	Count of camera frame where laser has lost tracked. Only applicable when tracking window is enabled.

Measurements

Measurement statistics are displayed for each measurement that has been configured on the **Measure** page. Use the **Reset** button to reset the statistics.

The following information is available for each measurement:

Dashboard Measurement Statistics

Name	Description

Measurements	The measurement ID and name.
Value	The most recent measurement value.
Min/Max	The minimum and maximum measurement values that have been observed.
Avg	The average of all measurement results collected since the sensor was started.
Std	The standard deviation of all measurement results collected since the sensor was started.
Pass/Fail	The counts of pass or fail decisions that have been generated.
Invalid	The count of frames from which no feature points could be extracted.

Gocator Device Files

This section describes the device files found on a Gocator.

Job Files

Job files contain settings that control system behavior when a sensor is running. If **Alignment Reference** is set to **Dynamic**, a job file will also contain transformation settings.

Elements in job files contain three types of values: settings, constraints, and properties. Settings are input values that can be edited. Constraints are read-only limits that define the valid values for settings. Properties are read-only values that provide supplemental information related to sensor setup.

When a job file is received from a sensor, it will contain settings, constraints, and properties. When a job file is sent to a sensor, any constraints or properties in the file will be ignored.

Changing the value of a setting can affect multiple constraints and properties. After you upload a job file, you can download the job file again to access the updated values of the constraints and properties.

All Gocator sensors share a common job file structure.

When editing the job file manually, you should only edit the elements that are applicable and leave the other elements in the structure.

> The following sections correspond to the XML structure used in job files.

Configuration (Root)

Configuration Child Elements

Element	Type	Description
@version	32u	Configuration version (101).
Setup	Section	See *Setup* (below) for a description of the Setup elements.
Tools	Collection	Collection of sections. Each section is an instance of a tool and is named by the type of the tool it describes. See the topics for each tool for more information.
Tools.options	String (CSV)	List of available tool types.
Output	Section	See *Output* (page 129) for a description of the Outputs elements.

Setup

The Setup element contains settings related to system and sensor setup.

Setup Child Elements

Element	Type	Description
AutoStartEnabled	Bool	Enables automatically data capture after boot-up.
TemperatureSafetyEnabled	Bool	Enables laser temperature safety control.
TemperatureSafetyEnabled.used	Bool	Whether or not this property is used.
ScanMode	32s	The default scan mode.
ScanMode options	String (CSV)	List of available scan modes.
OcclusionReductionEnabled	Bool	Enables occlusion reduction.
UniformSpacingEnabled	Bool	Enables uniform spacing.
UniformSpacingEnabled.used	Bool	Whether or not property is used.
UniformSpacingEnabled.value	Bool	Actual value used if not configurable.
IntensityEnabled	Bool	Enables intensity data collection.
IntensityEnabled.used	Bool	Whether or not property is used.
IntensityEnabled.value	Bool	Actual value used if not configurable.
ExternalInputZPulseEnabled	Bool	Enables the External Input based encoder Z Pulse feature.
Trigger	Section	See *Triggers* (page 125).
Layout	Section	See *Layout* (below).
Alignment	Section	See *Alignment* (next page).
Devices	Collection	A collection of two Device sections (with roles main and buddy).
SurfaceGeneration	Section	See *SurfaceGeneration* (page 123).
PartDetection	Section	See *PartDetection* (page 124).
Custom	Custom	Used by specialized sensors.

Layout

Layout Child Elements

Element	Type	Description
DataSource	32s	Data source of the layout output (read-only): 0 – Top 1 – Bottom 2 – Top left 3 – Top right
TransformedDataRegion	Region3D	Transformed data region of the layout output.
Orientation	32s	Sensor orientation: 0 – Wide 1 – Opposite 2 – Reverse

Element	Type	Description
Orientation options	32s (CSV)	List of available orientation options.
Orientation value	32s	Actual value used if not configurable.
MultiplexBuddyEnabled	Bool	Enables multiplexing for buddies.
MultiplexSingleEnabled	Bool	Enables multiplexing for a single sensor configuration.
MultiplexSingleExposureDuration	64f	Exposure duration in us (currently rounded to integer when read by the sensor)
MultiplexSingleDelay	32u	Delay in µs. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod	32u	Period in µs. (Currently gets rounded up when read by the sensor.)
XSpacingCount	32u	Number of points along X when data is resampled.
YSpacingCount	32u	Number of points along Y when data is resampled.
MultiplexSinglePeriod min	64f	Minimum period in µs.

Alignment

The Alignment element contains settings related to alignment and encoercalibration.

Alignment Child Elements

Element	Type	Description
InputTriggerEnabled	Bool	Enables digital input-triggered alignment operation.
Type	32s	Type of alignment operation: 0 – Stationary 1 – Moving
Type.options	32s (CSV)	List of available alignment types.
StationaryTarget	32s	Stationary alignment target: 0 – None 1 – Disk 2 – Bar 3 – Plate
StationaryTarget.options	32s (CSV)	List of available stationary alignment targets.
MovingTarget	32s	Moving alignment target: 0 – None 1 – Disk 2 – Bar 3 – Plate
MovingTarget.options	32s (CSV)	List of available moving alignment targets.
EncoderCalibrateEnabled	Bool	Enables encoder resolution calibration.
Disk	Section	See *Disk* (next page).
Bar	Section	See *Bar* (next page).
Plate	Section	See *Plate* (next page).

Disk

Disk Child Elements

Element	Type	Description
Diameter	64f	Disk diameter (mm).
Height	64f	Disk height (mm).

Bar

Bar Child Elements

Element	Type	Description
Width	64f	Bar width (mm).
Height	64f	Bar height (mm).
HoleCount	32u	Number of holes.
HoleDistance	64f	Distance between holes (mm).
HoleDiameter	64f	Diameter of holes (mm).

Plate

Plate Child Elements

Element	Type	Description
Height	64f	Plate height (mm).
HoleCount	32u	Number of holes.
RefHoleDiameter	64f	Diameter of reference hole (mm).
SecHoleDiameter	64f	Diameter of secondary hole(s) (mm).

Devices / Device

Devices / Device Child Elements

Element	Type	Description
@role	32s	Sensor role: 0 – Main 1 – Buddy
DataSource	32s	Data source of device output (read-only): 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
ActiveArea	Region3D	Active area. (Contains min and max attributes for each element.)
TransformedDataRegion	Region3D	Active area after transformation (read-only).

Element	Type	Description
FrontCamera	Window	Front camera window (read-only).
BackCamera	Window	Back camera window (read-only).
ExposureMode	32s	Exposure mode: 0 – Single exposure 2 – Dynamic exposure
ExposureMode.options	32s (CSV)	List of available exposure modes.
Exposure	64f	Single exposure (μs).
Exposure.min	64f	Minimum exposure (μs).
Exposure.max	64f	Maximum exposure (μs).
DynamicExposureMin	64f	Dynamic exposure range minimum (μs).
DynamicExposureMax	64f	Dynamic exposure range maximum (μs).
ExposureSteps	64f (CSV)	Mutiple exposure list (μs).
ExposureSteps.countMin	32u	Minimum number of exposure steps.
ExposureSteps.countMax	32u	Maximum number of exposure steps.
IntensityStepIndex	32u	Index of exposure step to use for intensity when using multiple exposures.
XSubsampling	32u	Subsampling factor in X.
XSubsampling.options	32u (CSV)	List of available subsampling factors in X.
ZSubsampling	32u	Subsampling factor in Z.
ZSubsampling.options	32u (CSV)	List of available subsampling factors in Z.
SpacingInterval	64f	Uniform spacing interval (mm).
SpacingInterval.min	64f	Minimum spacing interval (mm).
SpacingInterval.max	64f	Maximum spacing interval (mm).
SpacingInterval.used	Bool	Whether or not field is used.
SpacingInterval value	64f	Actual value used if not configurable.
SpacingIntervalType	32s	Spacing interval type: 0 – Maximum resolution 1 – Balanced 2 – Maximum speed
SpacingIntervalType.used	Bool	Whether or not field is used.
Tracking	Section	Described below.
Material	Section	Described below.
Custom	Custom	Used by specialized sensors.
XSpacingCount	32u	Number of resampled points along X (read-only).
YSpacingCount	32u	Number of resampled points along Y (read-only).

Tracking

Tracking Child Elements

Element	Type	Description
Enabled	Bool	Enables tracking.
SearchThreshold	64f	Percentage of spots that must be found to remain in track.
Height	64f	Tracking window height (mm).
Height.min	64f	Minimum tracking window height (mm).
Height.max	64f	Maximum tracking window height (mm).

Material

Material Child Elements

Element	Type	Description
Type	32s	Type of Material settings to use. 0 – Custom 1 – Diffuse
Type.used	Bool	Determines if the setting's value is currently used.
Type.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotThreshold	32s	Spot detection threshold.
SpotThreshold.used	Bool	Determines if the setting's value is currently used.
SpotThreshold.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotWidthMax	32s	Spot detection maximum width.
SpotWidthMax.used	Bool	Determines if the setting's value is currently used.
SpotWidthMax.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotWidthMax min	32s	Minimum allowed spot detection maximum value.
SpotWidthMax max	32s	Maximum allowed spot detection maximum value.
SpotSelectionType	32s	Spot selection type 0 – Best. Picks the strongest spot in a given column. 1 – Top. Picks the spot which is most Top/Left on the imager 2 – Bottom. Picks the spot which is most Bottom/Right on the imager
SpotSelectionType.used	Bool	Determines if the setting's value is currently used.
SpotSelectionType.value	32s	Value in use by the sensor, useful for determining value when used is false.
CameraGainAnalog	64f	Analog camera gain factor.
CameraGainAnalog.used	Bool	Determines if the setting's value is currently used.
CameraGainAnalog.value	64f	Value in use by the sensor, useful for determining value when used is false.

Element	Type	Description
CameraGainAnalog.min	64f	Minimum value.
CameraGainAnalog.max	64f	Maximum value.
CameraGainDigital	64f	Digital camera gain factor.
CameraGainDigital.used	Bool	Determines if the setting's value is currently used.
CameraGainDigital.value	64f	Value in use by the sensor, useful for determining value when used is false.
CameraGainDigital.min	64f	Minimum value.
CameraGainDigital.max	64f	Maximum value.
DynamicSensitivity	64f	Dynamic exposure control sensitivity factor. This can be used to scale the control setpoint.
DynamicSensitivity.used	Bool	Determines if the setting's value is currently used.
DynamicSensitivity.value	64f	Value in use by the sensor, useful for determining value when used is false.
DynamicSensitivity.min	64f	Minimum value.
DynamicSensitivity.max	64f	Maximum value.
DynamicThreshold	32s	Dynamic exposure control threshold. If the detected number of spots is fewer than this number, the exposure will be increased.
DynamicThreshold.used	Bool	Determines if the setting's value is currently used.
DynamicThreshold.value	32s	Value in use by the sensor, useful for determining value when used is false.
DynamicThreshold.min	32s	Minimum value.
DynamicThreshold.max	32s	Maximum value.
GammaType	32s	Gamma type.
GammaType used	Bool	Value in use by the sensor, useful for determining value when used is false.
GammaType value	32s	Determines if the setting's value is currently used.

SurfaceGeneration

SurfaceGeneration Child Elements

Element	Type	Description
Type	32s	Surface generation type: 0 – Continuous 1 – Fixed length 2 - Variable length 3 – Rotation
FixedLength	Section	Described below.
VariableLength	Section	Described below.
Rotational	Section	Described below.

FixedLength

FixedLength Child Elements

Element	Type	Description
StartTrigger	32s	Start trigger condition: 0 – Sequential 1 – Digital input
Length	64f	Surface length (mm).
Length.min	64f	Minimum surface length (mm).
Length.max	64f	Maximum surface length (mm).

VariableLength

VariableLength Child Elements

Element	Type	Description
MaxLength	64f	Maximum surface length (mm).
MaxLength.min	64f	Minimum value for maximum surface length (mm).
MaxLength.max	64f	Maximum value for maximum surface length (mm).

Rotational

Rotational Child Elements

Element	Type	Description
Circumference	64f	Circumference (mm).
Circumference.min	64f	Minimum circumference (mm).
Circumference.max	64f	Maximum circumference (mm).

PartDetection

PartDetection Child Elements

Element	Type	Description
Enabled	Bool	Enables part detection.
Enabled.used	Bool	Whether or not this field is used.
Enabled value	Bool	Actual value used if not configurable.
Threshold	64f	Height threshold (mm).
Threshold.min	64f	Minimum height threshold (mm).
Threshold.max	64f	Maximum height threshold (mm).
ThresholdDirection	64f	Threshold direction: 0 – Above 1 – Below
GapWidth	64f	Gap width (mm).
GapWidth.min	64f	Minimum gap width (mm).
GapWidth.max	64f	Maximum gap width (mm).

Element	Type	Description
GapLength	64f	Gap length (mm).
GapLength.min	64f	Minimum gap length (mm).
GapLength.max	64f	Maximum gap length (mm).
PaddingWidth	64f	Padding width (mm).
PaddingWidth.min	64f	Minimum padding width (mm).
PaddingWidth.max	64f	Maximum padding width (mm).
PaddingLength	64f	Padding length (mm).
PaddingLength.min	64f	Minimum padding length (mm).
PaddingLength.max	64f	Maximum padding length (mm).
MaxLength	64f	Max length (mm).
MaxLength.min	64f	Minimum value of max length (mm).
MaxLength.max	64f	Maximum value of max length (mm).
FrameOfReference	32s	Part frame of reference: 0 – Sensor 1 – Scan 2 – Part
EdgeFiltering	Section	See *EdgeFiltering* (below).
MinArea	64f	Minimum area (mm2).
MinArea.min	64f	Minimum value of minimum area.
MinArea.max	64f	Maximum value of minimum area.

EdgeFiltering

EdgeFiltering Child Elements

Element	Type	Description
Enabled	Bool	Enables edge filtering.
PreserveInteriorEnabled	Bool	Enables preservation of interior.
ElementWidth	64f	Element width (mm).
ElementWidth.min	64f	Minimum element width (mm).
ElementWidth.max	64f	Maximum element width (mm).
ElementLength	64f	Element length (mm).
ElementLength.min	64f	Minimum element length (mm).
ElementLength.max	64f	Maximum element length (mm).

Triggers

The Triggers element contains settings related trigger source, speed, and encoder resolution.

Triggers Child Elements

Element	Type	Description
Source	32s	Trigger source: 0 – Time 1 – Encoder 2 – Digital Input 3 – Software
Source.options	32s (CSV)	List of available source options.
Units	32s	Sensor triggering units when source is not clock or encoder: 0 – Time 1 – Encoder
FrameRate	64f	Frame rate for time trigger (Hz).
FrameRate.min	64f	Minimum frame rate (Hz).
FrameRate.max	64f	Maximum frame rate (Hz).
FrameRate.maxSource	32s	Source of maximum frame rate limit: 0 – Imager 1 – Surface generation
MaxFrameRateEnabled	Bool	Enables maximum frame rate (ignores FrameRate).
EncoderSpacing	64f	Encoder spacing for encoder trigger (mm).
EncoderSpacing.min	64f	Minimum encoder spacing (mm).
EncoderSpacing.max	64f	Maximum encoder spacing (mm).
EncoderSpacing.minSource	32s	Source of minimum encoder spacing: 0 – Resolution 1 – Surface generation
EncoderTriggerMode	32s	Encoder triggering mode: 0 – Tracking backward 1 – Bidirectional 2 – Ignore backward
Delay	64f	Trigger delay (µs or mm).
Delay.min	64f	Minimum trigger delay (µs or mm).
Delay.max	64f	Maximum trigger delay (µs or mm).
GateEnabled	Bool	Enables digital input gating.
GateEnabled.used	Bool	True if this parameter can be configured.
GateEnabled.value	Bool	Actual value if the parameter cannot be configured.

Tools

The Tools element contains measurement tools. The following sections describe each tool and its available measurements.

Tools Child Elements

Element	Type	Description
@options	String (CSV)	A list of the tools available in the currently selected scan mode.
<ToolType>	Section	An element for each added tool.

RangePosition

A RangePosition element defines settings for a range position tool and its measurement.

RangePosition Child Elements

Element	Type	Description
Name	String	Setting for measurement name.
Source	32s	Setting for profile source.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Measurements\Z	PositionMeasurement	Z measurement

PositionMeasurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set)
Name	String	Measurement name
Enabled	Boolean	Measurement enable state 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state 0 – Disable 1 – Enable
SmoothingWindow	32s	Smoothing window
Scale	64f	Output scaling factor
Offset	64f	Output offset factor
DecisionMin	64f	Minimum decision threshold
DecisionMax	64f	Maximum decision threshold

RangeThickness

A RangeThickness element defines settings for a range thickness tool and its measurement.

RangeThickness Child Elements

Element	Type	Description
Name	String	Setting for measurement name.
Source	32s	Setting for profile source.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Absolute	Boolean	Setting for selecting absolute or signed result. 0 – Signed 1 – Absolute
Measurements\Thickness	ThicknessMeasurement	Thickness measurement

ThicknessMeasurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set)
Name	String	Measurement name
Enabled	Boolean	Measurement enable state 0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state 0 – Disable 1 – Enable
SmoothingWindow	32s	Smoothing window
Scale	64f	Output scaling factor
Offset	64f	Output offset factor
DecisionMin	64f	Minimum decision threshold
DecisionMax	64f	Maximum decision threshold

Script

A Script element defines settings for a script measurement.

Script Child Elements

Element	Type	Description
Name	String	Setting for measurement name.
Code	String	Script code.
Measurements\Output	(Collection)	Dynamic list of Output elements.

Output

Element	Type	Description
id (attribute)	32s	Output ID. Optional (output disabled if set to -1).
Name	String	Output name

Output

The Output element contains the following sub-elements: Ethernet, Serial, Analog, Digital0, and Digital1. Each of these sub-elements defines the output settings for a different type of Gocator output.

For all sub-elements, the source identifiers used for measurement outputs correspond to the measurement identifiers defined in each tool's Measurements element. For example, in the following XML, in the options attribute of the Measurements element, 2 and 3 are the identifiers of measurements that are enabled and available for output. The value of the Measurements element (that is, 2) means that only the measurement with id 2 (Position Z) will be sent to output.

```
<RangePosition>   ...
  <Measurements>
    <Z id="2">   ...
<RangeThickness>   ...
  <Measurements>
    <Thickness id="3">   ...


<Output>
  <Ethernet>     ...
    <Measurements options="2,3">2</Measurements>
```

Ethernet

The Ethernet element defines settings for Ethernet output.

In the Ethernet element, the source identifiers used for video, range, profile, and surface output, as well as range, profile, and surface intensity outputs, correspond to the **sensor** that provides the data. For example, in the XML below, the options attribute of the Profiles element shows that only two sources are available (see the table below for the meanings of these values). The value of the Ranges element—0—indicates that only data from that source will be sent to output.

```
<RangePosition>   ...
  <Measurements>
    <Z id="2">   ...
<RangeThickness>   ...
  <Measurements>
    <Thickness id="3">   ...


<Output>
  <Ethernet>     ...
```

```
<Ranges options="0,1">0</Profiles>     ...
<RangeIntensities options="0" />     ...
<Measurements options="2,3">2</Measurements>
```

Ethernet Child Elements

Element	Type	Description
Protocol	32s	Ethernet protocol: 0 – Gocator 1 – Modbus 2 – EtherNet/IP 3 – ASCII
Ascii		See *ASCII* (page 134).
EIP		See EIP.
Modbus		See Modbus
Videos	32s (CSV)	Selected video sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right
Videos.options	32s (CSV)	List of available video sources (see above).
Ranges	32s (CSV)	Selected range sources. 0 – Top 1 – Bottom 2 – Top left 3 – Top right
Ranges.options	32s (CSV)	List of available range sources (see above).
RangeIntensities	32s (CSV)	Selected range intensity sources. 0 – Top 1 – Bottom 2 – Top left 3 – Top right
RangeIntensities.options	32s (CSV)	List of available range intensity sources (see above).
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.

ASCII

ASCII Child Elements

Element	Type	Description
Operation	32s	Operation mode: 0 – Asynchronous 1 – Polled

Element	Type	Description
ControlPort	32u	Control service port number.
HealthPort	32u	Health service port number.
DataPort	32u	Data service port number.
Delimiter	String	Field delimiter.
Terminator	String	Line terminator.
InvalidValue	String	String for invalid output.
CustomDataFormat	String	Custom data format.
CustomFormatEnabled	Bool	Enables custom data format.

EIP

EIP Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables EIP output buffering
EndianOutputType	32s	Endian Output Type 0 – Big endian 1 – Little endian
ImplicitOutputEnabled	Bool	Enables Implict (I/O) Messaging.

Modbus

Modbus Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables Modbus output buffering

Digital0 and Digital1

The Digital0 and Digital1 elements defines settings for the Gocator's two digital outputs.

Digital0 and Digital1 Child Elements

Element	Type	Description
Event	32s	Triggering event: 0 – None (disabled) 1 – Measurements 2 – Software 3 – Alignment state 3 – Alignment state
SignalType	32s	Signal type: 0 – Pulse 1 – Continuous
ScheduleEnabled	Bool	Enables scheduling.
PulseWidth	64f	Pulse width (µs).

Element	Type	Description
PulseWidth.min	64f	Minimum pulse width (µs).
PulseWidth.max	64f	Maximum pulse width (µs).
PassMode	32s	Measurement pass condition: 0 – AND of measurements is true 1 – AND of measurements is false 2 – Always assert
Delay	64f	Output delay (µs or mm, depending on delay domain defined below).
DelayDomain	32s	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.

Analog

The Analog element defines settings for Analog output.

The range of valid measurement values [DataScaleMin, DataScaleMax] is scaled linearly to the specified current range [CurrentMin, CurrentMax].

Only one Value or Decision source can be selected at a time.

Analog Child Elements

Element	Type	Description
Event	32s	Triggering event: 0 – None (disabled) 1 – Measurements 2 - Software
ScheduleEnabled	Bool	Enables scheduling.
CurrentMin	64f	Minimum current (mA).
CurrentMin.min	64f	Minimum value of minimum current (mA).
CurrentMin.max	64f	Maximum value of minimum current (mA).
CurrentMax	64f	Maximum current (mA).
CurrentMax.min	64f	Minimum value of maximum current (mA).
CurrentMax.max	64f	Maximum value of maximum current (mA).
CurrentInvalidEnabled	Bool	Enables special current value for invalid measurement value.
CurrentInvalid	64f	Current value for invalid measurement value (mA).
CurrentInvalid.min	64f	Minimum value for invalid current (mA).
CurrentInvalid.max	64f	Maximum value for invalid current (mA).
DataScaleMin	64f	Measurement value corresponding to minimum current.

Element	Type	Description
DataScaleMax	64f	Measurement value corresponding to maximum current.
Delay	64f	Output delay (µs or mm, depending on delay domain defined below).
DelayDomain	32s	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Measurement	32u	Selected measurement source.
Measurement.options	32u (CSV)	List of available measurement sources.

> The delay specifies the time or position at which the analog output activates. Upon activation, there is an additional delay before the analog output settles at the correct value.

Serial

The Serial element defines settings for Serial output.

Serial Child Elements

Element	Type	Description
Protocol	32s	Serial protocol: 0 – ASCII 1 – Selcom
Protocol.options	32s (CSV)	List of available protocols.
Selcom	Section	Described below.
Ascii	Section	Described below.
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.

Selcom

Selcom Child Elements

Element	Type	Description
Rate	32u	Output bit rate.
Rate.options	32u (CSV)	List of available rates.
Format	32s	Output format: 0 – 12-bit 1 – 12-bit with search 2 – 14-bit 3 – 14-bit with search
Format.options	32s (CSV)	List of available formats.
DataScaleMin	64f	Measurement value corresponding to minimum word value.
DataScaleMax	64f	Measurement value corresponding to maximum word value.

ASCII

ASCII Child Elements

Element	Type	Description
Delimiter	String	Field delimiter.
Terminator	String	Line terminator.
InvalidValue	String	String for invalid output.
CustomDataFormat	String	Custom data format.
CustomFormatEnabled	Bool	Enables custom data format.

Transformation File

The transformation file contains information about the physical system setup that is used to:

- Transform data from sensor coordinate system to another coordinate system (e.g., world)
- Define encoder resolution for encoder-based triggering
- Define the travel offset (Y offset) between sensors for staggered operation

The transformation file exists as a separate file when **Alignment Reference (page 55)** is set to **Fixed**. When **Alignment Reference** is set to **Dynamic**, transformation information is included in each job file (see page 117) under the Transform element.

Use the Read File and Write File commands to modify the transformation file. See the following sections for the elements contained in the transformation file.

Alignment Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transform version="100">
  <EncoderResolution>1</EncoderResolution>
  <Speed>100</Speed>
  <Devices>
    <Device role="0">
      <X>-2.3650924829</X>
      <Y>0.0</Y>
      <Z>123.4966803469</Z>
      <XAngle>5.7478302588</XAngle>
      <YAngle>3.7078302555</XAngle>
      <ZAngle>2.7078302556</XAngle>
      <Orientation>0</Orientation>
    </Device>
    <Device id="1">
      <X>0</X>
      <Y>0.0</Y>
      <Z>123.4966803469</Z>
      <XAngle>5.7478302588</XAngle>
      <YAngle>3.7078302555</XAngle>
      <ZAngle>2.7078302556</XAngle>
      <Orientation>0</Orientation>
    </Device>
  </Devices>
</Transform>
```

Transform

The Transform element contains the alignment record for both the Main and the Buddy sensor. The version attribute defines the version of the record format.

```
<Transform version="100">
```

Transform Child Elements

Element	Type	Description
EncoderResolution	64f	Encoder Resolution (mm/tick).
Speed	64f	Travel Speed (mm/s).
Devices	(Collection)	Contains two Device elements.

Device

A Device element defines the transformation for a sensor. There is one entry element per sensor, identified by a unique role attribute (0 for main and 1 for buddy):

```
<Device role="0">
```

Device Child Elements

Element	Type	Description
X	64f	Translation in the X axis (mm).
Y	64f	Translation in the Y axis (mm).
Z	64f	Translation in the Z axis (mm).
XAngle	64f	Rotation about X axis (degrees).
YAngle	64f	Rotation about Y axis (degrees).
ZAngle	64f	Rotation about Z axis (degrees).

The rotation (counter-clockwise in the X-Z plane) is performed before the translation.

Protocols

The following sections describe the protocols that Gocator sensors support.

Gocator Protocol

This section describes TCP and UDP commands and data formats used by a client computer to communicate with Gocator sensors. Network communication enables the client to:

- Discover Main and Buddy sensors on an IP network and re-configure their network addresses.
- Configure Main and Buddy sensors.
- Send commands to run sensors, provide software triggers, read/write files, etc.
- Receive data, health, and diagnostic messages.
- Upgrade firmware.

The Concepts section defines network connection types (Discovery, Control, Upgrade, Data, and Health), common data types, and other terminologies. Subsequent sections provide details about network commands and data formats.

The Gocator SDK provides open source C language libraries that implement the network commands and data formats defined in this section. See *Software Development Kit* (page 199) for more information.

General

Modes

A Gocator system can operate in the following modes.

System Modes

Mode	Description
Video	Sends raw video.
Range	Performs displacement measurements (default mode).

Buddy Communication Channels

The peer-to-peer control channels are used by Gocator sensors to communicate between sensors.

Channel	Port	Description
Discovery	2002, 2005, 2008	Gocator peer discovery port. UDP broadcasts on the subnet are sent once every second.

Channel	Port	Description
Command	2002 to 2015	Gocator request and response ports. Gocator uses UDP communications on these ports for configuration and reporting.
Data	2500	Main Gocator listens on this port for TCP traffic from the Buddy sensor. Buddy sensor communicates using a free port available at the time.

States

A Gocator system can be in one of three states: Conflict, Ready, or Running. The Start and Stop commands are sent by the client to change the current state. The sensor can be configured to boot in either the Ready or Running state.

In the Ready state, a sensor can be configured. In the Running state, a sensor will respond to input signals, perform measurements, drive its outputs, and send data messages to the client. Disconnecting to command channel will change the sensor from the Running state to the Ready state.

The Conflict state indicates that a sensor has been configured with a Buddy sensor but the Buddy sensor is not present on the network. The sensor will not accept some commands until the Change Buddy command is used to remove the configured Buddy.

Data Types

The table below defines the data types and associated type identifiers used throughout this document. All values are transmitted in little endian format (least significant byte first) unless stated otherwise.

Data Types

Type	Description	Null Value
char	Character (8-bit, ASCII encoding)	-
byte	Byte.	-
8u	8-bit unsigned integer.	-
8s	8-bit signed integer.	-
16s	16-bit signed integer.	-32768 (0x8000)
16u	16-bit unsigned integer.	65535 (0xFFFF)
32s	32-bit signed integer.	-2147483648 (0x80000000)
32u	32-bit unsigned integer.	4294967295 (0xFFFFFFFF)
64s	64-bit signed integer.	-9223372036854775808 (0x8000000000000000)
64u	64-bit unsigned integer.	18446744073709551615 (0xFFFFFFFFFFFFFFFF)
64f	64-bit floating point	-1.7976931348623157e+308
Point16s	Two 16-bit signed integers	-

> IP addresses are an exception to the little endian rule. The bytes in the address "a.b.c.d" will always be transmitted in the order a, b, c, d (big endian).

Status Codes

Each reply on Discovery (page 139), Control (page 141), and Upgrade (page 164) channels contains a *status* field containing a status code indicating the result of the command. The following status codes are defined.

Status Codes

Label	Value	Description
OK	1	Command succeeded.
Failed	0	Command failed.
Invalid State	-1000	Command is not valid in the current state.
Item Not Found	-999	A required item (e.g., file) was not found.
Invalid Command	-998	Command is not recognized.
Invalid Parameter	-997	One or more command parameters are incorrect.
Not Supported	-996	The operation is not supported.

Discovery Commands

Sensors ship with the following default network configuration:

Setting	Default
DHCP	0 (disabled)
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0 (disabled)

The Get Address and Set Address discovery commands can be used to modify a sensor's network configuration. Discovery commands are UDP broadcast messages:

Destination Address	Destination Port
255.255.255.255	3220

When a sensor accepts a discovery command, it will send a UDP broadcast response:

Destination Address	Destination Port
255.255.255.255	Port of command sender.

The use of UDP broadcasts for discovery enables a client computer to locate a sensor when the senor and client are configured for different subnets. All you need to know is the serial number of the sensor in order to locate it on an IP network.

Get Address

The Get Address command is used to discover Gocator sensors across subnets.

Command

Field	Type	Offset	Description
length	64u	0	Command length.
type	64s	8	Command type (0x1).
signature	64u	16	Message signature (0x0000504455494D4C).
deviceId	64u	24	Serial number of the device whose address information is queried. 0 selects all devices.

Reply

Field	Type	Offset	Description
length	64u	0	Reply length.
type	64s	8	Reply type (0x1001).
status	64s	16	Operation status.
signature	64u	24	Message signature (0x0000504455494D4C).
deviceId	64u	32	Serial number.
dhcpEnabled	64u	40	0 – Disabled 1 – Enabled
reserved[4]	byte	48	Reserved.
address[4]	byte	52	The IP address in left to right order.
reserved[4]	byte	56	Reserved.
subnetMask[4]	byte	60	The subnet mask in left to right order.
reserved[4]	byte	64	Reserved.
gateway[4]	byte	68	The gateway address in left to right order.
reserved[4]	byte	72	Reserved.
reserved[4]	byte	76	Reserved.

Set Address

The Set Address command modifies the network configuration of a Gocator sensor. On receiving the command, the Gocator will perform a reset. You should wait 30 seconds before re-connecting to the Gocator.

Command

Field	Type	Offset	Description
length	64u	0	Command length.
type	64s	8	Command type (0x2).
status	64s	16	1 – OK 0 – Error
signature	64u	24	Message signature (0x0000504455494D4C).
deviceId	64u	32	Serial number of the device whose address information is queried. 0 selects all devices.

Field	Type	Offset	Description
dhcpEnabled	64u	40	0 – Disabled 1 – Enabled
reserved[4]	byte	48	Reserved.
address[4]	byte	52	The IP address in left to right order.
reserved[4]	byte	56	Reserved.
subnetMask[4]	byte	60	The subnet mask in left to right order.
reserved[4]	byte	64	Reserved.
gateway[4]	byte	68	The gateway address in left to right order.
reserved[4]	byte	72	Reserved.
reserved[4]	byte	76	Reserved.

Reply

Field	Type	Offset	Description
length	64u	0	Reply length.
type	64s	8	Reply type (0x1002).
status	64s	16	Operation status.
signature	64u	24	Message signature (0x0000504455494D4C).
deviceId	64u	32	Serial number.

Control Commands

A client can send commands and receive responses over the Control and Upgrade TCP channels.

Command Channels

Channel	Port	Description
Control	3190	Sensor accepts commands for most operations.
Upgrade	3192	Sensor accepts commands for firmware upgrades.

The Control and Upgrade channels can be connected simultaneously, but the sensor will accept only a single connection on each port. If an additional connection is attempted on a port that is already connected, the previous connection will be closed and the new connection will be accepted.

Protocol Version

The Protocol Version command reports the Control protocol version of the connected sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4511)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4511).
status	32s	6	Reply status.
majorVersion	8u	10	Major version.
minorVersion	8u	11	Minor version.

Get Address

The Get Address command is used to discover Gocator sensors across subnets.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x3012)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x3012).
status	32s	6	Reply status.
dhcpEnabled	byte	10	0 – DHCP not used 1 – DHCP used
address[4]	byte	11	IP address (most significant byte first).
subnetMask[4]	byte	15	Subnet mask.
gateway[4]	byte	19	Gateway address.

Set Address

The Set Address command modifies the network configuration of a Gocator sensor. On receiving the command, the Gocator will perform a reset. You should wait 30 seconds before re-connecting to the Gocator.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x3013)
dhcpEnabled	byte	6	0 – DHCP not used 1 – DHCP used
address[4]	byte	7	IP address (most significant byte first).
subnetMask[4]	byte	11	Subnet mask.
gateway[4]	byte	15	Gateway address.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x3013).
status	32s	8	Reply status.

Get System Info

The Get System Info command reports information for sensors that are visible in the system.

Firmware version refers to the version of the Gocator's firmware installed on each individual sensor. The client can upgrade the Gocator's firmware by sending the Start Upgrade command (see page 165). Firmware upgrade files are available from the downloads section under the support tab on the LMI web site. See *Firmware Upgrade* (page 61) for more information on obtaining the latest firmware.

Every Gocator sensor contains factory backup firmware. If a firmware upgrade command fails (e.g., power is interrupted), the factory backup firmware will be loaded when the sensor is reset or power cycled. In this case, the sensors will fall back to the factory default IP address. To avoid IP address conflicts in a multi-sensor system, connect to one sensor at a time and re-attempt the firmware upgrade.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4002)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4002).
status	32s	6	Reply status.
localInfo	Sensor Info	10	Info for this device.
remoteCount	32u	66	Number of discovered sensors.
remoteInfo [remoteCount]	Sensor Info	70	List of info for discovered sensors.

SensorInfo

Field	Type	Offset	Description
deviceId	32u	0	Serial number of the device.
address[4]	byte	4	IP address (most significant byte first).
modelName[32]	char	8	Model name.
firmwareVersion[4]	byte	40	Firmware version (most significant byte first).
state	32s	44	Sensor state -1 – Conflict

Field	Type	Offset	Description
			0 – Ready 1 – Running
role	32s	48	Sensor role 0 – Main 1 – Buddy
buddyId	32s	52	Serial number of paired device (main or buddy). 0 if unpaired.

Get States

The Get System Info command a list of useful system states.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4525)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4525).
status	32s	6	Reply status.
count	32u	10	Number of state variables.
sensorState	32s	14	Sensor state -1 – Conflict 0 – Ready 1 – Running
loginState	32s	18	Device login state 0 – No user 1 – Administrator 2 – Technician
alignmentReference	32s	22	Alignment reference 0 – Fixed 1 – Dynamic
alignmentState	32s	26	Alignment state 0 – Unaligned 1 – Aligned
recordingEnabled	32s	30	Whether or not recording is enabled 0 – Disabled 1 – Enabled
playbackSource	32s	34	Playback source 0 – Live data

Field	Type	Offset	Description
			1 – Recorded data
uptimeSec	32s	38	Uptime (whole seconds component)
uptimeMicrosec	32s	42	Uptime (remaining microseconds component)
playbackPos	32s	46	Playback position
playbackCount	32s	50	Playback frame count
autoStartEnabled	32s	54	Auto-start enable (boolean)

Log In/Out

The Log In/Out command is used to log in or out of a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4003)
userType	32s	6	Defines the user type 0 – None (log out) 1 – Administrator 2 – Technician
password[64]	char	10	Password (required for log-in only).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4003).
status	32s	6	Reply status.

Change Password

The Change Password command is used to change log-in credentials for a user.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4004)
user type	32s	6	Defines the user type 0 – None (log out) 1 – Administrator 2 – Technician
password[64]	char	10	New password.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4004).
status	32s	6	Reply status.

Set Buddy

The Set Buddy command is used to assign or unassign a Buddy sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4005)
buddyId	32u	6	Id of the sensor to acquire as buddy. Set to 0 to remove buddy.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4005).
status	32s	6	Reply status.

List Files

The List Files command returns a list of the files in the sensor's file system.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101A)
extension[64]	char	6	Specifies the extension used to filter the list of files (does not include the "."). If an empty string is used, then no filtering is performed.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101A).
status	32s	6	Reply status.
count	32u	10	Number of file names.
fileNames[count][64]	char	14	File names

Copy File

The Copy File command copies a file from a source to a destination within the connected sensor. Copy a saved configuration to "_live.job" to make the configuration live.

Command

Field	Type	Offset	Description
length	64s	0	Command size – in bytes.
id	64s	4	Command identifier (0x101B).
source[64]	char	6	Source file name.
destination[64]	char	70	Destination file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101B).
status	32s	6	Reply status.

Read File

Downloads a file from the connected sensor. Read the file "_live.job" and "_live.prof" to download the live configuration and template.

Command

Field	Type	Offset	Description
length	32u	0	Command size – in bytes.
id	16u	4	Command identifier (0x1007).
name[64]	char	6	Source file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size – in bytes.
id	16u	4	Reply identifier (0x1007).
status	32s	6	Reply status.
length	32u	10	File length
data[length]	byte	14	File contents

Write File

The Write File command uploads a file to the connected sensor. Write to "_live.job"to write the make the job file live. Except for writing to the live files, the file is permanently stored on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1006)
name[64]	char	6	Source file name.
length	32u	70	File length
data[length]	byte	74	File contents

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1006).
status	32s	6	Reply status.

Delete File

The Delete File command removes a file from the connected sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1008)
name[64]	char	6	Source file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1008).
status	32s	6	Reply status.

Get Default Job

The Get Default Job command gets the name of a default job file that will be loaded at boot time.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4100)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4100).
status	32s	6	Reply status.
name[64]	char	10	File name (null-terminated).

Set Default Job

The Set Default Job command sets the name of a default job file that will be loaded at boot time.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4101).
fileName[64]	char	6	File name (null-terminated).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4101).
status	32s	6	Reply status.

Get Loaded Job

The Get Loaded Job command returns the currently loaded (i.e., live) file name and modified status for a file type.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4512).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4512).
status	32s	6	Reply status.
fileName[64]	char	10	Name of the currently loaded file.
changed	8u	74	Whether or not the currently loaded file has been changed (1: yes; 0: no).

Set Alignment Reference

The Set Alignment Reference command is used to set the sensor's alignment reference.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4103).
reference	32s	6	Alignment reference 0 – Fixed 1 – Dynamic

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4103).
status	32s	6	Reply status.

Get Alignment Reference

The Get Alignment Reference command is used to get the sensor's alignment reference.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4104).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4104).
status	32s	6	Reply status.
reference	32s	10	Alignment reference 0 – Fixed 1 – Dynamic

Clear Alignment

The Clear Alignment command clears sensor alignment.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4102).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4102).
status	32s	6	Reply status.

Get Timestamp

The Get Timestamp command retrieves the sensor's timestamp, in clock ticks. All devices in a system are synchronized with the system clock; this value can be used for diagnostic purposes, or used to synchronize the start time of the system.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100A).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x100A).
status	32s	6	Reply status.
timestamp	64u		Timestamp, in clock ticks.

Get Encoder

This command retrieves the current system encoder value.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101C).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101C).
status	32s	6	Reply status.
encoder	64s	10	Current encoder position, in ticks.

Reset Encoder

The Reset Encoder command is used to reset the current encoder value.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101E).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101E).
status	32s	6	Reply status.

Start

The Start command starts the sensor system (system enters the Running state).

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100D).
reserved	64s		Reserved field – set to 0.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x100D).
status	32s	6	Reply status.

Scheduled Start

The scheduled start command starts the sensor system (system enters the Running state) at target time or encoder value (depending on the trigger mode).

Command

Field	Type	Description
length	64s	Command size – in bytes.
id	64s	Command identifier (0x100F).
target	64s	Target scheduled start value (in ticks or microseconds, depending on the trigger type).

Reply

Field	Type	Description
length	64s	Reply size – in bytes.
id	64s	Reply identifier (0x101D).
status	64s	Reply status.

Stop

The Stop command stops the sensor system (system enters the Ready state).

Command

Field	Type	Type	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1001).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1001).
status	32s	6	Reply status.

Start Alignment

The Start Alignment command is used to start the alignment procedure on a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4600).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4600).
status	32s	6	Reply status.
opId	32s	10	Operation ID.

Start Exposure Auto-set

The Start Exposure Auto-set command is used to start the exposure auto-set procedure on a sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4601).
role	32s	6	Role of sensors to auto-set. 0 – Main 1 – Buddy

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4601).
status	32s	6	Reply status.
opId	32s	10	Operation ID.

Software Trigger

The Software Trigger command causes the sensor to take a snapshot while in software mode and in the Running state.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4510).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4510).
status	32s	6	Reply status.

Schedule Digital Output

The Schedule Digital Output command schedules a digital output event. The digital output must be configured to accept software-scheduled commands and be in the Running state. See *Digital Output* (page 107) for information on setting up the digital output.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4518).
index	16u	6	Index of the output (starts from 0)
target	64s	8	Specifies the time (clock ticks) or position (um) of when the event should happen. The target value is ignored if the **Signal** setting in **Digital** in the **Output** panel is not set to **Scheduled**. The output will be triggered immediately. See *Digital Output* (page 107) for information on setting **Signal**.
value	8u	16	Specifies the target state: 0 – Set to low (continuous) 1 – Set to high (continuous) Ignored if output type is pulsed.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4518).
status	32s	6	Reply status.

Schedule Analog Output

The Schedule Analog Output command schedules an analog output event. The analog output must be configured to accept software-scheduled commands and be in the Running state. See *Analog Output* (page 109) for information on setting up the analog output.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4519).

Field	Type	Offset	Description
index	16u	6	Index of the output. Must be 0.
target	64s	8	Specifies the time (clock ticks) or position (encoder ticks) of when the event should happen. The target value is ignored if the **Signal** setting in **Analog** in the **Output** panel is not set to **Scheduled**. The output will be triggered immediately. See *Analog Output* (page 109) for information on setting the Signal type.
value	32s	16	Output current (micro-amperes).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4519).
status	32s	6	Reply status.

> The analog output takes about 75 us to reach 90% of the target value for a maximum change, then roughly another 40 us to settle completely.

Ping

The Ping command can be used to test the control connection. This command has no effect on sensors.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x100E).
timeout	64u	6	Timeout value (microseconds).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x100E).
status	32s	6	Reply status.

> If a non-zero value is specified for timeout, the client must send another ping command before the timeout elapses; otherwise the server would close the connection. The timer is reset and updated with every command.

Reset

The Reset command reboots the Main sensor and any Buddy sensors. All sensors will automatically reset 3 seconds after the reply to this command is transmitted.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4300).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4300).
status	32s	6	Reply status.

Backup

The Backup command creates a backup of all files stored on the connected sensor and downloads the backup to the client.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1013).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1013).
status	32s	6	Reply status.
length	32u	10	Data length.
data[length]	byte	14	Data content.

Restore

The Restore command uploads a backup file to the connected sensor and then restores all sensor files from the backup.

> The sensor must be reset or power-cycled before the restore operation can be completed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1014).
length	32u	10	Data length.
data[length]	byte	14	Data content.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1014).
status	32s	6	Reply status.

Restore Factory

The Restore Factory command restores the connected sensor to factory default settings. This command has no effect on connected Buddy sensors.

Note that the sensor must be reset or power-cycled before the factory restore operation can be completed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4301).
resetAddress	8u	6	Specifies whether IP address should be restored to default: 0 – Do not reset IP 1 – Reset IP

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4301).
status	32s	6	Reply status.

Set Recording Enabled

The Set Recording Enabled command enables recording for replay later.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4516).
enable	8u	6	0: disabled; 1: enabled.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4516).
status	32s	6	Reply status.

Get Recording Enabled

The Get Recording Enabled command retrieves whether recording is enabled.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4517).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4517).
status	32s	6	Reply status.
enable	8u	10	0: disabled; 1: enabled.

Clear Replay Data

The Clear Replay Data command clears the sensors replay data..

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4513).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4513).
status	32s	6	Reply status.

Set Playback Source

The Set Playback Source command sets the data source for data acquisition.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4523).
source	32s	6	Source 0 – Live 1 – Replay buffer

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4523).
status	32s	6	Reply status.

Get Playback Source

The Get Playback Source command gets the data source for data acquisition.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4524).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4524).
status	32s	6	Reply status.
source	32s	10	Source 0 – Live 1 – Replay buffer

Simulate

The Simulate command simulates the last frame if playback source is live, or the current frame if playback source is the replay buffer.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4522).
source	32s	6	Source 0 – Live 1 – Replay buffer

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4522).
status	32s	6	Reply status.
bufferValid	8u	10	Whether or not the buffer is valid.

A reply status of -996 means that the current configSuration (mode, sensor type, etc.) does not support simulation.

A reply status of -992 means that the simulation buffer is empty. Note that the buffer can be valid even if the simulation buffer is actually empty due to optimization choices. This scenario means that the simulation buffer would be valid if data were recorded.

Seek Playback

The Seek Playback command seeks to any position in the current playback dataset. The frame is then sent.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4503).
frame	32u	6	Frame index.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4503).
status	32s	6	Reply status.

Step Playback

The Step Playback command advances playback by one frame.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4501).
direction	32s	6	Define step direction 0 – Forward 1 – Reverse

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4501).
status	32s	6	Reply status.

When the system is running in the Replay mode, this command advances replay data (playback) by one frame. This command returns error if no live playback data set is loaded. You can use the Copy File command to load a replay data set to live.

Playback Position

The Playback Position command retrieves the current playback position.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4502).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4502).
status	32s	6	Reply status.
Frame Index	32u	10	Current frame index (starts from 0).
Frame Count	32u	14	Total number of available frames/objects.

Clear Measurement Stats

The Clear Measurement Stats command clears the sensor's measurement statistics.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4526).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4526).
status	32s	6	Reply status.

Simulate Unaligned

The Simulate Unaligned command simulates data before alignment transformation.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452A).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x452A).
status	32s	6	Reply status.

Acquire

The Acquire command acquires a new scan.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4528).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4528).
status	32s	6	Reply status.

Acquire Unaligned

The Acquire Unaligned command acquires a new scan without performing alignment transformation.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4527).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4527).
status	32s	6	Reply status.

Read File (Progressive)

The progressive Read File command reads the content of a file as a stream.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4529).
name[64]	char	6	Source file name.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4529).
status	32s	6	Reply status.

Field	Type	Offset	Description
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

The reply may be followed by a series of continue replies.

Export CSV (Progressive)

The progressive Export CSV command exports replay data as a CSV stream.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4507).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4507).
status	32s	6	Reply status.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

All recorded range data is exported in the CSV.

The reply may be followed by a series of continue replies. See the section on progressive reply.

Export Bitmap (Progressive)

The progressive Export Bitmap command exports replay data as a bitmap stream.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4508).
type	32s	6	Data type: 0 – Range or video 1 – Intensity
source	32s	10	Data source to export.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4508).
status	32s	6	Reply status.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

> The reply may be followed by a series of continue replies. See the section on progressive reply.

Upgrade Commands

A client can send commands and receive responses over the Control and Upgrade TCP channels.

Command Channels

Channel	Port	Description
Control	3190	Sensor accepts commands for most operations.
Upgrade	3192	Sensor accepts commands for firmware upgrades.

The Control and Upgrade channels can be connected simultaneously, but the sensor will accept only a single connection on each port. If an additional connection is attempted on a port that is already connected, the previous connection will be closed and the new connection will be accepted.

After connecting to a Gocator device, you can use the Get Protocol Version command to retrieve the protocol version.

Protocol version refers to the version of the Gocator Protocol supported by the *connected sensor* (the sensor to which a command connection is established), and consists of major and minor parts. The minor part is updated when backward-compatible additions are made to the Gocator Protocol. The major part will be updated in the event that breaking changes are made to the Gocator Protocol.

Get Protocol Version

The Get Protocol Version command reports the Upgrade protocol version of the connected sensor.

Command

Field	Type	Description
length	64s	Command size – in bytes.
id	64s	Command identifier (0x0100).

Reply

Field	Type	Description
length	64s	Reply size – in bytes.
id	64s	Reply identifier.

Field	Type	Description
status	64s	Reply status.
majorVersion	64s	Major version.
minorVersion	64s	Minor version.

Start Upgrade

The Start Upgrade command begins a firmware upgrade for the Main sensor and any Buddy sensors. All sensors will automatically reset 3 seconds after the upgrade process is complete.

Command

Field	Type	Offset	Description
length	64s	0	Command size – in bytes.
id	64s	4	Command identifier (0x0000).
skipValidation	byte	6	Whether or not to skip validation (0 – do not skip, 1 – skip).
length	32u	7	Length of the upgrade package (bytes).
data[length]	byte	11	Upgrade package data.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x0000).
status	32s	6	Reply status.

Get Upgrade Status

The Get Upgrade Status command determines the progress of a firmware upgrade.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1).
status	32s	6	Reply status.
state	32s	10	Upgrade state: -1 – Failed 0 – Completed 1 – Running 2 – Completed, but should run again
progress	32u	14	Upgrade progress (valid when in the Running state)

Get Upgrade Log

The Get Upgrade Log command can retrieve an upgrade log in the event of upgrade problems.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x2)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x2).
status	32s	6	Reply status.
length	32u	10	Length of the log (bytes).
log[length]	char	14	Log content.

Data Results

A client can receive data messages from a Gocator sensor by connecting to the Data or Health TCP channels.

Result Channels

Channel	Port	Description
Data	3196	Sensor sends data messages.
Health	3194	Sensor sends health messages.

The ports above can be connected simultaneously and the sensor will also accept multiple connections on each port.

Messages that are received on the Data and Health channels use a common structure, called Gocator Data Protocol (GDP). Each GDP message consists of a 6-byte header followed by a variable-length, message-specific content section. The structure of the GDP message is defined below.

Gocator Data Protocol

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag Bits 0-14: Message type identifier.

GDP messages are always sent in groups. The last message flag in the control field is used to indicate the final message within a group. If there is only one message per group, this bit will be set in each message.

Stamp

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
count (C)	32u	6	Count of stamps in this message.
size	16u	10	Stamp size, in bytes (min: 56, current: 56).
source	8u	12	Source (0 – Main, 1 – Buddy).
reserved	8u	13	Reserved.
stamps[C]	Stamp	14	Array of stamps (see below).

Stamp

Field	Type	Offset	Description
frameIndex	64u	0	Frame index (counts up from zero).
timestamp	64u	8	Timestamp (us).
encoder	64s	16	Current encoder value (ticks).
encoderAtZ	64s	24	Encoder value latched at z/index mark (ticks).
status	64u	32	Bit field containing various frame information: Bit 0: sensor digital input state Bit 4: master digital input state Bit 8-9: inter-frame digital pulse trigger (Master digital input if master is connected, otherwise sensor digital input. Value is cleared after each frame and clamped at 3 if more than 3 pulses are received).
serialNumber	32u	40	Sensor serial number (main if buddied).
reserved[2]	32u	44	Reserved.

Video

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
attributesSize	16u	6	Size of attributes, in bytes (min: 20, current: 20).
height (H)	32u	8	Image height, in pixels.
width (W)	32u	12	Image width, in pixels.
pixelSize	8u	16	Pixel size, in bytes.
pixelFormat	8u	17	Pixel format: 1 – 8-bit greyscale

Field	Type	Offset	Description
			2 – 8-bit color filter 3 – 8-bits-per-channel color (B, G, R, X)
colorFilter	8u	18	Color filter array alignment: 0 – None 1 – Bayer BG/GR 2 – Bayer GB/RG 3 – Bayer RG/GB 4 – Bayer GR/BG
source	8u	19	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
cameraIndex	8u	20	Camera index.
exposureIndex	8u	21	Exposure index.
exposure	32u	22	Exposure (ns).
reserved[2]	8u	26	Reserved.
pixels[H][W]	(Variable)	28	Image pixels. (Depends on pixelSize above.)

Range

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
attributeSize	16u	6	Size of attributes, in bytes (min: 20, current: 20).
count (C)	32u	8	Number of profile arrays.
zScale	32u	12	Z scale (nm).
zOffset	32s	16	Z offset (µm).
source	8u	20	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	21	Exposure (ns).
reserved[3]	8u	25	Reserved.
range[C]	16s	28	Range values.

Range Intensity

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
attributeSize	16u	6	Size of attributes, in bytes (min: 12, current: 12).
count (C)	32u	8	Number of profile arrays.
source	8u	12	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	13	Exposure (ns).
reserved[3]	8u	17	Reserved.
range[C]	8u	20	Range intensity values.

Measurement

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
count (C)	32u	6	Count of measurements in this message.
reserved[2]	8u	10	Reserved.
id	16u	12	Measurement identifier.
measurements[C]	Measurement	16	Array of measurements (see below).

Measurement

Field	Type	Offset	Description
value	32s	0	Measurement value.
decision	8u	4	Measurement decision bitmask. Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 – Invalid anchor
reserved[3]	8u	5	Reserved.

Alignment Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 8).
opId	32u	8	Operation ID.
status	32s	12	Operation status.

Exposure Calibration Result

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
attributesSize	16u	6	Size of attributes, in bytes (min: 12, current: 12).
opId	32u	8	Operation ID.
status	32s	12	Operation status.
exposure	32s	16	Exposure result (ns).

Health Results

A client can receive data messages from a Gocator sensor by connecting to the Data or Health TCP channels.

Result Channels

Channel	Port	Description
Data	3196	Sensor sends data messages.
Health	3194	Sensor sends health messages.

The ports above can be connected simultaneously and the sensor will also accept multiple connections on each port.

Messages that are received on the Data and Health channels use a common structure, called Gocator Data Protocol (GDP). Each GDP message consists of a 6-byte header followed by a variable-length, message-specific content section. The structure of the GDP message is defined below.

Gocator Data Protocol

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag Bits 0-14: Message type identifier.

GDP messages are always sent in groups. The last message flag in the control field is used to indicate the final message within a group. If there is only one message per group, this bit will be set in each message.

A Health Result contains a single data block for health *indicators*. Each indicator reports the current status of some aspect of the sensor system, such as CPU usage or network throughput.

Health Result Header

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).
control	16u	4	Bit 15: Last message flag. Bits 0-14: Message type identifier.
count (C)	32u	6	Count of indicators in this message.
source	8u	10	Source (0 – Main, 1 – Buddy).
reserved[3]	8u	13	Reserved
indicators[C]	Indicator	16	Array of indicators (see format below).

The health indicators block contains a 2 dimensional array of indicator data. Each row in the array has the following format:

Indicator Format

Field	Type	Offset	Description
id	64s	0	Unique indicator identifier (see below).
instance	64s	4	Indicator instance.
value	64s	8	Value (identifier-specific meaning).

The following health indicators are defined for Gocator sensor systems:

Health Indicators

Indicator	Id	Instance	Value
Encoder Value	1003	-	Current system encoder tick.
Encoder Frequency	1005	-	Current system encoder frequency (ticks/s).
App Version	2000	-	Firmware application version.
Uptime	2017	-	Time elapsed since node boot-up or reset (seconds).
Internal Temperature	2002	-	Internal temperature (centidegrees Celsius).
Projector Temperature	2404	-	Projector module temperature (centidegrees Celsius). Only available on projector based devices.
Control Temperature	2028	-	Control module temperature (centidegrees Celsius). Available only on 3B-class devices.
Memory Usage	2003	-	Amount of memory currently used (bytes).
Memory Capacity	2004	-	Total amount of memory available (bytes).

Indicator	Id	Instance	Value
Storage Usage	2005	-	Amount of non-volatile storage used (bytes).
Storage Capacity	2006	-	Total amount of non-volatile storage available (bytes).
CPU Usage	2007	-	CPU usage (percentage of maximum).
Net Out Capacity	2009	-	Total available outbound network throughput (bytes/s).
Net Out Link Status	2034	-	Current Ethernet link status.
Sync Source	2043	-	Gocator synchronization source. 1 - FireSync Master device 2 - Sensor
Digital Inputs	2024	-	Current digital input status (one bit per input).
Event Count	2102	-	Total number of events triggered.
Camera Search Count	2217	-	Number of search states. (Only important when tracking is enabled.)
Camera Trigger Drops	2201	-	Number of dropped triggers.
Analog Output Drops	2501	Output Index	Number of dropped outputs.
Digital Output Drops	2601	Output Index	Number of dropped outputs.
Serial Output Drops	2701	Output Index	Number of dropped outputs.
Sensor State	20000	-	Gocator sensor state. -1 – Conflict 0 – Ready 1 – Running
Current Sensor Speed	20001	-	Current sensor speed. (Hz)
Maximum Speed	20002	-	The sensor's maximum speed.
Spot Count	20003	-	Number of found spots in the last profile.
Max Spot Count	20004	-	Maximum number of spots that can be found.
Scan Count	20005	-	Number of surfaces detected from a top device.
Laser Overheat	20020	-	Indicates whether laser overheat has occurred. 0 – Has not overheated 1 – Has overheated Only available on certain 3B laser devices.
Laser Overheat Duration	20021	-	The length of time in which the laser overheating state occurred. Only available on certain 3B laser devices.
Playback Position	20023	-	The current replay playback position.
Playback Count	20024	-	The number of frames present in the replay.
FireSync Version	20600	-	The FireSync version used by the Gocator build.

Indicator	Id	Instance	Value
Processing Drops	21000	-	Number of dropped frames. The sum of various processing drop related indicators.
Last IO Latency	21001	-	Last delay from camera exposure to when rich IO scheduling occurs. Valid only if rich IO is enabled.
Max IO Latency	21002	-	Maximum delay from camera exposure to when rich IO scheduling occurs. Valid only if rich IO is enabled. Reset on start.
Ethernet Output	21003	-	Number of bytes transmitted.
Ethernet Rate	21004	-	The average number of bytes per second being transmitted.
Ethernet Drops	21005	-	Number of dropped Ethernet packets.
Trigger Drops	21010		Number of dropped triggers. The sum of various triggering-related drop indicators.
Output Drops	21011		Number of dropped output data. The sum of all output drops (analog, digital, serial, host server, and ASCII server).
Host Server Drops	21012		The number of bytes dropped by the host data server. Not currently emitted.
ASCII Server Drops	21013		The number of bytes dropped by the ASCII Ethernet data server. Not currently emitted.
Range Valid Count	21100	-	Number of valid ranges.
Range Invalid Count	21101	-	Number of invalid ranges.
Anchor Invalid Count	21200	-	Number of frames with anchoring invalid.
Z-Index Drop Count	22000	-	The number of dropped surfaces due to a lack of z-encoder pulse during rotational part detection.
Value	30000	Measurement ID	Measurement Value.
Pass	30001	Measurement ID	Number of pass decision.
Fail	30002	Measurement ID	Number of fail decision.
Max	30003	Measurement ID	Maximum measurement value.
Min	30004	Measurement ID	Minimum measurement value.
Average	30005	Measurement ID	Average measurement value.
Std. Dev.	30006	Measurement ID	Measurement value standard deviation.
Invalid Count	30007	Measurement ID	Number of invalid values.
Overflow	30008	Measurement ID	Number of times this measurement has overflown on any output. Multiple simultaneous overflows result in only a sinlge increment to this counter. Overflow conditions include: -Value exceeds bit representation available for

Indicator	Id	Instance	Value
			given protocol -Analog output (mA) falls outside of acceptable range (0-20 mA) When a measurement value overflow occurs, the value is set to the null value appropriate for the given protocol's measurement value output type. The Overflow health indicator increments.

Additional undocumented indicator values may be included in addition to the indicators defined above.

Modbus TCP Protocol

Modbus TCP is designed to allow industrial equipment such as Programmable Logic Controllers (PLC), sensors, and physical input/output devices to communicate over an Ethernet network.

Modbus TCP embeds a Modbus frame into a TCP frame in a simple manner. This is a connection-oriented transaction, and every query expects a response.

This section describes the Modbus TCP commands and data formats. Modbus TCP communication enables the client to:

- Switch to a different active job.
- Align and run sensors.
- Receive sensor states, stamps, and measurement results.

Modbus TCP is enabled in the **Output** panel. For more information, see *Ethernet Output* (page 103).

If buffering is enabled with the Modbus protocol, the PLC must read the Buffer Advance output register (see page 178) to advance the queue before reading the measurement results.

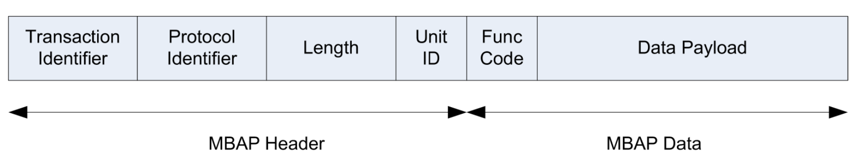
Concepts

A PLC sends a command to start each Gocator. The PLC then periodically queries each Gocator for its latest measurement results. In Modbus terminology, the PLC is a Modbus Client. Each Gocator is a Modbus Server which serves the results to the PLC.

The Modbus TCP protocol uses TCP for connection and messaging. The PLC makes a TCP connection to the Gocator on port 502. Control and data messages are communicated on this TCP connection. Up to eight clients can be connected to the Gocator simultaneously. A connection will be closed after 10 minutes of inactivity.

Messages

All Modbus TCP messages consist of an MBAP header (Modbus Application Protocol), a function code, and a data payload.



The MBAP header contains the following fields:

Modbus Application Protocol Header

Field	Length (Bytes)	Description
Transaction ID	2	Used for transaction pairing. The Modbus Client sets the value and the Server (Gocator) copies the value into its responses.

Field	Length (Bytes)	Description
Protocol ID	1	Always set to 0.
Length	1	Byte count of the rest of the message, including the Unit identifier and data fields.
Unit ID	1	Used for intra-system routing purpose. The Modbus Client sets the value and the Server (Gocator) copies the value into its responses.

Modbus Application Protocol Specification describes the standard function codes in detail. Gocator supports the following function codes:

Modbus Function Code

Function Code	Name	Data Size (bits)	Description
3	Read Holding Registers	16	Read multiple data values from the sensor.
4	Read Input Registers	16	Read multiple data values from the sensor.
6	Write Single Register	16	Send a command or parameter to the sensor.
16	Write Multiple Registers	16	Send a command and parameters to the sensor.

The data payload contains the registers that can be accessed by Modbus TCP messages. If a message accesses registers that are invalid, a reply with an exception is returned. Modbus Application Protocol Specification defines the exceptions and describes the data payload format for each function code.

The Gocator data includes 16-bit, 32-bit, and 64-bit data. All data are sent in big endian format, with the 32-bit and 64-bit data spread out into two and four consecutive registers.

32-bit Data Format

Register	Name	Bit Position
0	32-bit Word 1	31 .. 16
1	32-bit Word 0	15 .. 0

64-bit Data Format

Register	Name	Bit Position
0	64-bit Word 3	63 .. 48
1	64-bit Word 2	47 .. 32
2	64-bit Word 1	31 .. 16
3	64-bit Word 0	15 .. 0

Registers

Modbus registers are 16 bits wide and are either control registers or output registers.

Control registers are used to control the sensor states (e.g., start, stop, or calibrate a sensor).

The output registers report the sensor states, stamps, and measurement values and decisions. You can read multiple output registers using a single Read Holding Registers or a single Read Input Registers command. Likewise, you can control the state of the sensor using a single Write Multiple Register command.

Control registers are write-only, and output registers are read-only.

Register Map Overview

Register Address	Name	Read/Write	Description
0 - 124	Control Registers	WO	Registers for Modbus commands. See *Control Registers* (below) for detailed descriptions.
300 -371	Sensor States	RO	Report sensor states. See *State* (next page) for detailed descriptions.
900 - 999	Stamps	RO	Return stamps associated with each range. See *State* (next page) for detailed descriptions.
1000 - 1060	Measurements & Decisions	RO	20 measurement and decision pairs. See *Measurement Registers* (page 179) for detailed descriptions.

Control Registers

Control registers are used to operate the sensor. Register 0 stores the command to be executed. Registers 1 to 21 contain parameters for the commands. The Gocator executes a command when the value in Register 0 is changed. To set the parameters before a command is executed, you should set up the parameters and the command using a single Multiple Write register command.

Control Register Map

Register Address	Name	Read/Write	Description
0	Command Register	WO	Command register. See the Command Register Values table below for more information.
1 – 21	Job Filename	WO	Null-terminated filename. Each 16-bit register holds a single character. Only used for Load Job Command. Specifies the complete filename, including the file extension ".job".

The values used for the Command Register are described below.

Command Register Values

Value	Name	Description
0	Stop running	Stop the sensor. No effect if sensor is already stopped.
1	Start Running	Start the sensor. No effect if sensor is already started.
2	Align (stationary target)	Start the alignment process. State register 301 will be set to 1 (busy) until the alignment process is complete.
3	Align (moving target)	Start alignment process and also calibrate encoder resolution. State

Value	Name	Description
		register 301 will be set to 1 (busy) until the motion calibration process is complete.
4	Clear Alignment	Clear the alignment.
5	Load Job	Activate a job file. Registers 1 - 21 specify the filename.

Output Registers

Output registers are used to output states, stamps, and measurement results. Each register address holds a 16-bit data value.

State

State registers report the current sensor state.

State Register Map

Register Address	Name	Type	Description
300	Stopped / Running		Sensor State: 0 - Stopped 1 - Running
301	Busy		Busy State: 0 - Not busy 1 - Busy Registers 302 to 363 below are only valid when the Busy State is not Busy
302	Alignment State		Current Alignment State: 0 - Not aligned 1- Aligned
303 – 306	Encoder Value	64s	Current Encoder value (ticks).
307 – 310	Time	64s	Current time (us).
311	Job File Length	16u	Number of characters in the current job file name.
312 – 371	Live Job Name		Current Job Name. Name of currently loaded job file. Does not include the extension. Each 16-bit register contains a single character.

Stamp

Stamps contain trigger timing information used for synchronizing a PLC's actions. A PLC can also use this information to match up data from multiple Gocator sensors.

Stamps are updated after each range data is processed.

Stamp Register Map

Register Address	Name	Type	Description
976	Buffer Advance		If buffering is enabled this address must be read by the PLC Modbus client first to advance the buffer. After the buffer advance read operation, the Modbus client can read the updated Measurements & Decisions in addresses 1000-1060.
977	Buffer Counter		Number of buffered messages currently in the queue.
978	Buffer Overflow		Buffer Overflow Indicator: 0 - No overflow 1 - Overflow
979	Inputs		Digital input state.
980	zPosition High	64s	Encoder value when the index is last triggered.
981	zPosition		
982	zPosition		
983	zPosition Low		
984	Exposure High	32u	Laser exposure (us).
985	Exposure Low		
986	Temperature High	32u	Sensor temperature (degress celcius * 1000).
987	Temperature Low		
988	Position High	64s	Encoder position
989	Position		
990	Position		
991	Position Low		
992	Time Low	64u	Timestamp (us).
993	Time		
994	Time		
995	Time Low		
996	Frame Index High	64u	Frame counter. Each new sample is assigned a frame number.
997	Frame Index		
998	Frame Index		
999	Fame Index Low		

Measurement Registers

Measurement results are reported in pairs of values and decisions. Measurement values are 32 bits wide and decisions are 8 bits wide.

The measurement ID defines the register address of each pair. The register address of the first word can be calculated as (1000 + 3 * ID). For example, a measurement with ID set to 4 can be read from registers 1012 (high word) and, 1013 (low word), and the decision at 1015.

The measurement results are updated after each range data is processed.

Measurement Register Map

Register Address	Name	Type	Description
1000	Measurement 0 High	32s	Measurement value in um (0x80000000 if invalid)
1001	Measurement 0 Low		
1002	Decision 0	16u	Measurement decision. A bit mask, where: Bit 0: 1 - Pass 0 - Fail Bits 1-7: 0 - Measurement value OK 1 = Invalid value 2 = Invalid anchor
1003	Measurement 1 High		
1004	Measurement 1 Low		
1005	Decision 1		
1006	Measurement 2 High		
1007	Measurement 2 Low		
1008	Decision 2		
...	...	...	...
1057	Measurement 19 High		
1058	Measurement 19 Low		
1059	Decision 19		

EtherNet/IP Protocol

EtherNet/IP is an industrial protocol that allows bidirectional data transfer with PLCs. It encapsulates the object-oriented Common Industrial Protocol (CIP).

This section describes the EtherNet/IP messages and data formats. EtherNet/IP communication enables the client to:

- Switch to a different active job.
- Align and run sensors.
- Receive sensor states, stamps, and measurement results.

EtherNet/IP is enabled in the **Output** panel. For more information, see *Ethernet Output* (page 103).

Concepts

To EtherNet/IP-enabled devices on the network, the sensor information is seen as a collection of objects, which have attributes that can be queried. For example, an "assembly object" is a type of object with a data attribute that can be accessed with the GetAttribute and SetAttribute commands. The Gocator uses assembly objects to take commands and provide sensor state and measurement values.

The PLC sends a command to start a Gocator. The PLC then periodically queries the attributes of the assembly objects for its latest measurement results. In EtherNet/IP terminology, the PLC is a scanner and the Gocator is an adapter.

The Gocator supports unconnected or connected explicit messaging (with TCP). Implicit I/O messaging is not supported.

The default EtherNet/IP ports are used. Port 44818 is used for TCP connections and UDP queries (e.g., list Identity requests). Port 2222 for UDP I/O Messaging is not supported.

Basic Object

Identity Object (Class 0x01)

Attribute	Name	Type	Value	Description	Access
1	Vendor ID	UINT	1256	ODVA Provided Vendor ID	Get
2	Device Type	UINT	43	Device Type	Get
3	Product Code	UINT	2000	Product Code	Get
4	Revision	USINT USINT	x.x	Byte 0 - Major Revision Byte 1 - Minor Revision	Get
6	Serial number	UDINT	32-bit value	Sensor serial number.	Get
7	Product Name	SHORT STRING 32	"Gocator"	Gocator Product Name	Get

TCP/IP Object (Class 0xF5)

The TCP/IP Object contains read-only network configuration attributes such as IP Address. TCP/IP configuration via Ethernet/IP is not supported. See Volume 2, Chapter 5-3 of the CIP Specification for a complete listing of TCP/IP object attributes.

Attribute	Name	Type	Value	Description	Access
1	Status	UDINT	0	TCP interface status	Get
2	Configuration Capability	UINT	0		Get
3	Configuration Control	UINT	0	Product Code	Get
4	Physical Link Object	Structure (See description)		See 5.3.3.2.4 of CIP Specification Volume 2: Path size (UINT) Path (Padded EPATH)	Get
5	Interface Configuration	Structure (See description)		See 5.3.3.2.5 of CIP Specification Volume 2" IP Address (UDINT) Network Mask (UDINT), Gateway Address (UDINT) Name Server (UDINT) Secondary Name (UDINT) Domain Name (UDINT)	Get

Ethernet Link Object (Class 0xF6)

The Ethernet Link Object contains read-only attributes such as MAC Address (Attribute 3). See Volume 2, Chapter 5-4 of the CIP Specification for a complete listing of Ethernet Link object attributes.

Attribute	Name	Type	Value	Description	Access
1	Interface Speed	UDINT	1000	Ethernet interface data rate (mbps)	Get
2	Interface Flags	UDINT		See 5.4.3.2.1 of CIP Specification Volume 2: Bit 0: Link Status 0 – Inactive 1 - Active Bit 1: Duplex 0 – Half Duplex 1 – Full Duplex	Get
3	Physical Address	Array of 6 USINTs		MAC Address (for example: 00 16 20 00 2E 42)	Get

Assembly Object (Class 0x04)

The Gocator Ethernet/IP object model includes the following assembly objects: Command, Sensor State, and Sample State.

All assembly object instances are static. Data in a data byte array in an assembly object are stored in the big endian format.

Command Assembly

The command assembly object is used to start, stop, and align the sensor, and also to switch jobs on the sensor.

Command Assembly

Information	Value
Class	0x4
Instance	0x310
Number of Attributes	3
Length	32 bytes
Supported Service	0x10 (Write Single Attribute)

Attributes 1 and 2 are not implemented, as they are not required for the static assembly object.

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See Below	Command parameters Byte 0 - Command. See table below for specification of the values.	Get, Set

Command Definitions

Value	Name	Description
0	Stop running	Stop the sensor. No action if the sensor is already stopped
1	Start Running	Start the sensor. No action if the sensor is already started.
2	Stationary Alignment	Start the stationary alignment process. Byte 1 of the sensor state assembly will be set to 1 (busy) until the alignment process is complete, then back to zero.
3	Moving Alignment	Start the moving alignment process. Byte 1 of the sensor state assembly will be set to 1 (busy) until the alignment process is complete, then back to zero.
4	Clear Alignment	Clear the alignment.
5	Load Job	Load the job.

Sensor State Assembly

The sensor state assembly object contains the sensor's states, such as the current sensor temperature, frame count, and encoder values.

Sensor State Assembly

Information	Value
Class	0x4

Information	Value
Instance	0x320
Number of Attributes	3
Length	100 bytes
Supported Service	0x0E (Get Single Attribute)

Attributes 1 and 2 are not implemented, as they are not required for the static assembly object.

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Sensor state information. See below for more details.	Get

Sensor State Information

Byte	Name	Type	Description
0	Sensor's state		Sensor state: 0 - Ready 1 - Running
1	Command in progress		Command busy status: 0 - Not busy 1 - Busy performing the last command
2	Alignment state		Alignment status: 0 - Not aligned 1 - Aligned The value is only valid when byte1 is set to 0.
3-10	Encoder	64s	Current encodesr position
11-18	Time	64s	Current timestamp
19	Current Job Filename Length	16u	Number of characters in the current job filename. (e.g., 9 for "myjob.job"). The length includes the .job extension. Valid when byte 1 = 0.
20-43	Current Job Filename		Name of currently loaded job file. Includes the ".job" extension. Each byte contains a single character. Valid when byte 1 = 0.
44 - 99	Reserved		Reserved bytes

Sample State Assembly

The sample state object contains measurements and their associated stamp information.

Sample State Assembly

Information	Value
Class	0x04
Instance	0x321

Information	Value
Number of Attributes	3
Length	380 bytes
Supported Service	0x0E (Get Single Attribute)

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array		Sample state information. See below for more details	Get

Sample State Information

Byte	Name	Type	Description
0-1	Inputs		Digital input state
2-9	Z Index Position	64s	Encoder position at time of last index pulse (64-bit signed integer)
10-13	Exposure	32u	Laser exposure in us
14-17	Temperature	32u	Sensor temperature in degrees Celsius * 1000 (centidegrees)
18-25	Position	64s	Encoder position 64-bit signed integer)
26-33	Time	64u	Time (64-bit unsigned integer)
34-41	Frame Counter	64u	Frame counter (64-bit unsigned integer)
42	Buffer Counter	8u	Number of buffered messages currently in the queue.
43	Buffer Overflow		Buffer Overflow Indicator: 0 - No overflow 1 - Overflow
44 - 79	Reserved		Reserved bytes
80-83	Measurement 0	32s	Measurement value in um (0x80000000 if invalid)
84	Decision 0	8u	Measurement decision. A bit mask, where: Bit 0: 1 - Pass 0 - Fail Bits 1-7: 0 - Measurement value OK 1 = Invalid value 2 = Invalid anchor
85-88	Measurement 1		
89	Decision 1		
...	...		
375-378	Measurement 59		
379	Decision 59		

Measurement results are reported in pairs of values and decisions. Measurement values are 32 bits wide and decisions are 8 bits wide.

The measurement ID defines the byte position of each pair within the state information. The position of the first word can be calculated as (80 + 5 * ID). For example, a measurement with ID set to 4 can be read from byte 100 (high word) to 103 (low word) and the decision at 104.

If buffering is enabled in the Ethernet Output panel, reading the Extended Sample State Assembly Object automatically advances the buffer. See See *Ethernet Output* (page 103) for information on the **Output** panel.

ASCII Protocol

This section describes the ASCII protocol available over the Ethernet and serial outputs. The protocol communicates using ASCII strings. The output result format from the sensor is user-configurable.

Ethernet Communication

Gocator's Ethernet communication is bidirectional. Measurement results are sent on the Ethernet output in one of two modes: Polling or Asynchronous. The ASCII protocol over Ethernet enables the client to:

- Switch to a different job.
- Align, run, and trigger sensors.
- Receive sensor states, health indicators, stamps, and measurement results

As with the Gocator Protocol (see page 137), there are separate channels for Control, Data, and Health messages. The Control channel is used for commands. The Data channel is used to receive and poll for measurement results. The Health channel is used to receive health indicators.

The port number used for each channel is configurable. Each port can accept multiple connections, up to a total of 16 connections for all ports.

Channels can share the same port or operate on individual ports. The default port number is 8190 for all channels. The following port numbers are reserved for Gocator internal use: 80, 843, 2000-2100, 2500-2600, 3100-3250.

Asynchronous and Polling Operation

On the Ethernet output, the Data channel can operate asynchronously or by polling. Under asynchronous operation, measurement results are automatically sent on the Data channel when the sensor is in the running state and results become available. The result is sent on all connected data channels.

Under polling operation, when the sensor receives a Get Result command, it will send the latest measurement results on the same data channel that the request is received.

Command and Reply Format

Commands are sent from the client to the Gocator. Command strings are not case sensitive The command format is:

<COMMAND><DELIMITER><PARAMETER><TERMINATION>

If a command has more than one parameter, each parameter is separated by the delimiter. Similarly, the reply has the following format:

<STATUS><DELIMITER><OPTIONAL RESULTS><DELIMITER>

The status can either be "OK" or "ERROR". The optional results can be relevant data for the command if successful, or a text based error message if the operation failed. If there is more than one data item, each item is separated by the delimiter.

The delimiter and termination characters are configured in the Special Character settings.

Special Characters

The ASCII Protocol has three special characters.

Special Characters

Special Character	Explanation
Delimiter	Separates input arguments in commands and replies, or data items in results. Default value is ",".
Terminator	Terminates both commands and result output. Default value is "%r%n".
Invalid	Represents invalid measurement results. Default value is "INVALID"

The values of the special characters are defined in the Special Character settings. In addition to normal ASCII characters, the special characters can also contain the following format values.

Format values for Special Characters

Format Value	Explanation
%t	Tab
%n	New line
%r	Carriage return
%%	Percentage (%) symbol

Standard Result Format

Measurement results can either be sent in the standard format or in a custom format. In the standard format, you select in the web interface which measurement values and decisions to send. For each measurement the following message is transmitted:

M	t_n	,	i_n	,	V	v_n	,	D	d_1	CR

Field	Shorthand	Length	Description
MeasurementStart	M	1	Start of measurement frame.
Type	t_n	n	Hexadecimal value that identifies the type of measurement. The measurement type is the same as defined elsewhere (see page 166).
Id	i_n	n	Decimal value that represents the unique identifier of the measurement.
ValueStart	V	1	Start of measurement value.
Value	v_n	n	Measurement value, in decimal. The unit of the value is measurement-specific.

Field	Shorthand	Length	Description
DecisionStart	D	1	Start of measurement decision.
Decision	d_1	1	Measurement decision, a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 - Invalid anchor

Custom Result Format

In the custom format, you enter a format string with place holders to create a custom message. The default format string is "%time, %value[0], %decision[0]".

Result Placeholders

Format Value	Explanation
%time	Timestamp
%encoder	Encoder position
%frame	Frame number
%value[Measurement ID]	Measurement value of the specified measurement ID. The ID must correspond to an existing measurement. The value output will be displayed as an integer in micrometers.
%decision[Measurement ID]	Measurement decision, where the selected measurement ID must correspond to an existing measurement. Measurement decision is a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits 1-7: 0 – Measurement value OK 1 – Invalid value 2 - Invalid anchor

Control Commands

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Start

The Start command starts the sensor system (causes it to enter the Running state). This command is only valid when the system is in the Ready state. If a start target is specified, the sensor starts at the target time or encoder (depending on the trigger mode).

Formats

Message	Format
Command	Start, start target The start target (optional) is the time or encoder position at which the sensor will be started. The time and encoder target value should be set by adding a delay to the time or encoder position returned by the Stamp command. The delay should be set such that it covers the command response time of the Start command.
Reply	OK or ERROR, <Error Message>

Examples:

```
Start

OK

Start,1000000

ok

Start

ERROR, Could not start the sensor
```

Stop

The stop command stops the sensor system (causes it to enter the Ready state). This command is valid when the system is in the Ready or Running state.

Formats

Message	Format
Command	Stop
Reply	OK or ERROR, <Error Message>

Examples:

```
Stop

OK
```

Trigger

The Trigger command triggers a single frame capture. This command is only valid if the sensor is configured in the Software trigger mode and the sensor is in the Running state. If a start target is specified, the sensor starts at the target time or encoder (depending on the unit setting in the Trigger panel; see page 64).

Formats

Message	Format
Command	Trigger, start target The start target (optional) is the time or encoder position at which the sensor will be started. The time and encoder target value should be set by adding a delay to the time or encoder position returned by the Stamp command. The delay should be set such that it covers the command response time of the Start command.
Reply	OK or ERROR, <Error Message>

Examples:

```
Trigger

OK

Trigger,1000000

OK
```

Load Job

The Load Job command switches the active sensor configuration.

Formats

Message	Format
Command	LoadJob, job file name If the job file name is not specified, the command returns the current job name. An error message is generated if there is no job loaded. ".job" is appended if the filename does not have an extension.
Reply	OK or ERROR, <Error Message>

Examples:

```
LoadJob,test.job

OK,test.job loaded successfully

LoadJob

OK,test.job

LoadJob,wrongname.job

ERROR, failed to load wrongname.job

OK
```

Stamp

The Stamp command retrieves the current time, encoder, and/or the last frame count.

Formats

Message	Format
Command	Stamp,time,encoder,frame If no parameters are given, time, encoder, and frame will be returned. There could be more than one selection.
Reply	If no arguments are specified OK, time, <time value>, encoder, <encoder position>, frame, <frame count> ERROR, <Error Message> If arguments are specified, only the selected stamps will be returned.

Examples:

```
Stamp

OK,Time,9226989840,Encoder,0,Frame,6

Stamp,frame

OK,6

OK,test.job

LoadConfig,wrongname.job
```

Stationary Alignment

The Stationary Alignment command performs an alignment based on the settings in the sensor's live job file. A reply to the command is sent when the alignment has completed or failed. The command is timed out if there has been no progress after one minute.

Formats

Message	Format
Command	StationaryAlignment
Reply	If no arguments are specified OK or ERROR, <Error Message>

Examples:

```
StationaryAlignment

OK

StationaryAlignment

ERROR,ALIGNMENT FAILED
```

Moving Alignment

The Moving Alignment command performs an alignment based on the settings in the sensor's live job file. A reply to the command is sent when the alignment has completed or failed. The command is timed out if there has been no progress after one minute.

Formats

Message	Format
Command	MovingAlignment
Reply	If no arguments are specified OK or ERROR, <Error Message>

Examples:

```
MovingAlignment

OK

MovingAlignment

ERROR, ALIGNMENT FAILED
```

Clear Alignment

The Clear Alignment command clears the alignment record generated by the alignment process.

Formats

Message	Format
Command	ClearAlignment
Reply	OK or ERROR, <Error Message>

Examples:

```
ClearAlignment

OK
```

Data Commands

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Get Result

The Get Result command retrieves measurement values and decisions.

Formats

Message	Format
Command	Result, measurement ID, measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified, OK, <data string in standard format> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Result,0,1
```

```
OK,M00,00,V151290,D0,M01,01,V18520,D0
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Result,2
```

```
ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %value[0], %decision[0]):

```
Result
```

```
OK,1420266101, 151290, 0
```

Get Value

The Get Value command retrieves measurement values.

Formats

Message	Format
Command	Value, measurement ID, measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified, OK, <data string in standard format, except that the decisions are not sent> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Value,0,1
```

```
OK,M00,00,V151290,M01,01,V18520
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Value,2
```

```
ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %value[0]):

```
Value

OK, 1420266101, 151290
```

Get Decision

The Get Decision command retrieves measurement decisions.

Formats

Message	Format
Command	Decision, measurement ID, measurement ID...
Reply	If no arguments are specified, the custom format data string is used. OK, <custom data string> ERROR, <Error Message> If arguments are specified, OK, <data string in standard format, except that the values are not sent> ERROR, <Error Message>

Examples:

Standard data string for measurements ID 0 and 1:

```
Decision,0,1

OK,M00,00,D0,M01,01,D0
```

Standard formatted measurement data with a non-existent measurement of ID 2:

```
Decision,2

ERROR,Specified measurement ID not found. Please verify your input
```

Custom formatted data string (%time, %decision[0]):

```
Decision

OK,1420266101, 0
```

Health Commands

Optional parameters are shown in italic. The placeholder for data is surrounded by brackets (<>). In the examples, the delimiter is set to ','.

Get Health

The Get Health command retrieves health indicators. See *Health Results* (page 170) for details on health indicators.

Formats

Message	Format
Command	Health, health indicator ID.health indicator instance ... More than one health indicator can be specified. Note that the health indicator instance is optionally attached to the indicator ID with a '.'. If the health indicator instance field is

Message	Format
	used the delimiter cannot be set to '.'.
Reply	OK, <health indicator of first ID>, <health indicator of second ID> ERROR, <Error Message>

Examples:

```
health,2002,2017
```

```
OK,46,1674
```

```
Health
```

```
ERROR,Insufficient parameters.
```

Serial Protocol

This section describes the Selcom Serial Protocol settings and message formats supported by Gocator sensors.

Data communication is synchronous using two unidirectional (output only) RS-485 serial channels: data (Serial_Out0) and clock (Serial_Out1). See *Serial Output* (page 238) for cable pinout information.

Serial Communication

Gocator's serial communication is unidirectional (output only). Measurement results are sent on the Serial output in Asynchronous mode. While measurement values and decisions can be transmitted to an RS-485 receiver, job handling and control operations must be performed through the Gocator's web interface or through communications on the Ethernet output.

See *Serial Output* (page 238) for cable pinout information.

Gocator serial communication uses the following connection settings:

Serial Connection Settings

Parameter	Value
Start Bits	1
Stop Bits	1
Parity	None
Data Bits	8
Baud Rate (b/s)	115200
Format	ASCII

Connection Settings

Selcom Serial Protocol uses the following connection settings:

Serial Connection Settings

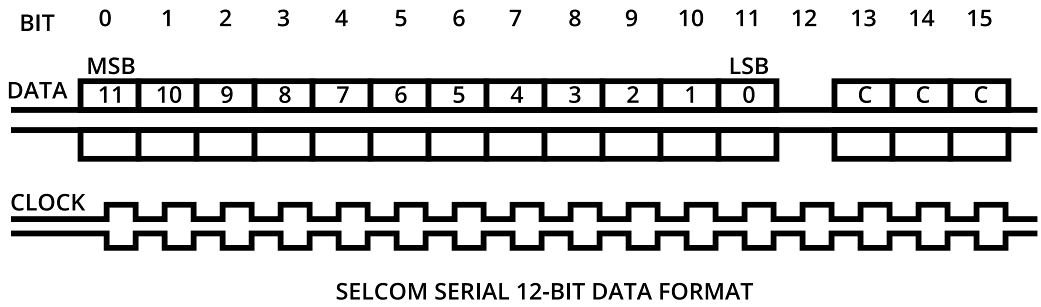
Parameter	Value
Data Bits	16
Baud Rate (b/s)	96000, 512000, 1024000
Format	Binary

Message Format

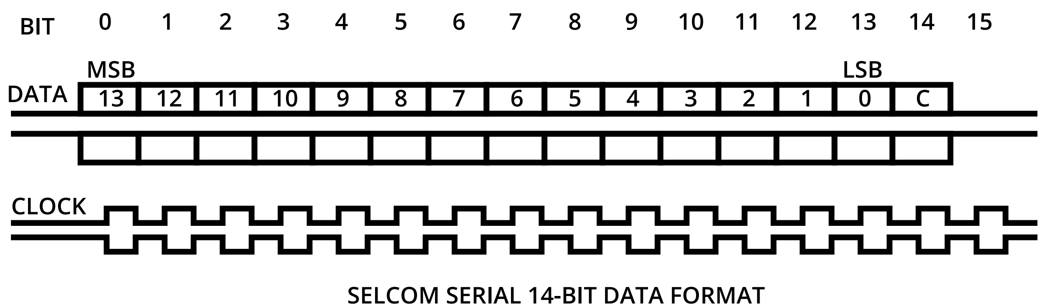
The data channel is valid on the rising edge of the clock and data is output with the most significant bit first, followed by control bits for a total of 16 bits of information per frame. The time between the start of the camera exposure and the delivery of the corresponding range data is fixed to a deterministic value.

The sensor support can output data in either 12 or 14 bits, illustrated below:

MSB = most significant bit, LSB = least significant bit, C = data valid bit (high = invalid)



Selcom serial 12-bit data format



Selcom serial 14-bit data format

Software Development Kit

The Gocator Software Development Kit (SDK) includes open-source software libraries and documentation that can be used to programmatically access and control Gocator sensors. The latest version of the SDK can be downloaded by going to http://lmi3d.com/support/downloads/, selecting a product type, and clicking on the *Product User Area* link.

Applications compiled with previous versions of the SDK are compatible with Gocator firmware if the major version numbers of the protocols match. For example, an application compiled with version 4.0 of the SDK (which uses protocol version 4.0) will be compatible with a Gocator running firmware version 4.1 (which uses protocol version 4.1). However, any new features in firmware version 4.1 would not be available.

If the major version number of the protocol is different, for example, an application compiled using SDK version 3.x being used with a Gocator running firmware 4.x, you must recompile the application.

The Gocator API, included in the SDK, is a C language library that provides support for the commands and data formats used with Gocator sensors. The API is written in standard C to allow the code to be compiled for any operating system. A pre-built DLL is provided to support 32-bit and 64-bit Windows operating systems. Projects and makefiles are included to support other editions of Windows and Linux.

For Windows users, code examples explaining how to wrap the calls in C# and VB.NET are provided in the tools package, which can be downloaded at http://lmi3d.com/support/downloads/.

For more information about programming with the Gocator SDK, refer to the class reference and sample programs included in the Gocator SDK.

Setup and Locations

Class Reference

The full SDK class reference is found by accessing 14400-4.x.x.xx_SOFTWARE_GO_SDK\GO_SDK\doc\GoSdk\Gocator_1x00\GoSdk.html.

Examples

Examples showing how to perform various operations are provided, each one targeting a specific area. All of the examples can be found in *GoSdkSamples.sln*.

To run the SDK samples, make sure *GoSdk.dll* and *kApi.dll* (or *GoSdkd.dll* and *kApid.dll* in debug configuration) are copied to the executable directory. All sample code, including C examples, is now located in the *Tools* package, which can be downloaded by going to http://lmi3d.com/support/downloads/.

Sample Project Environment Variable

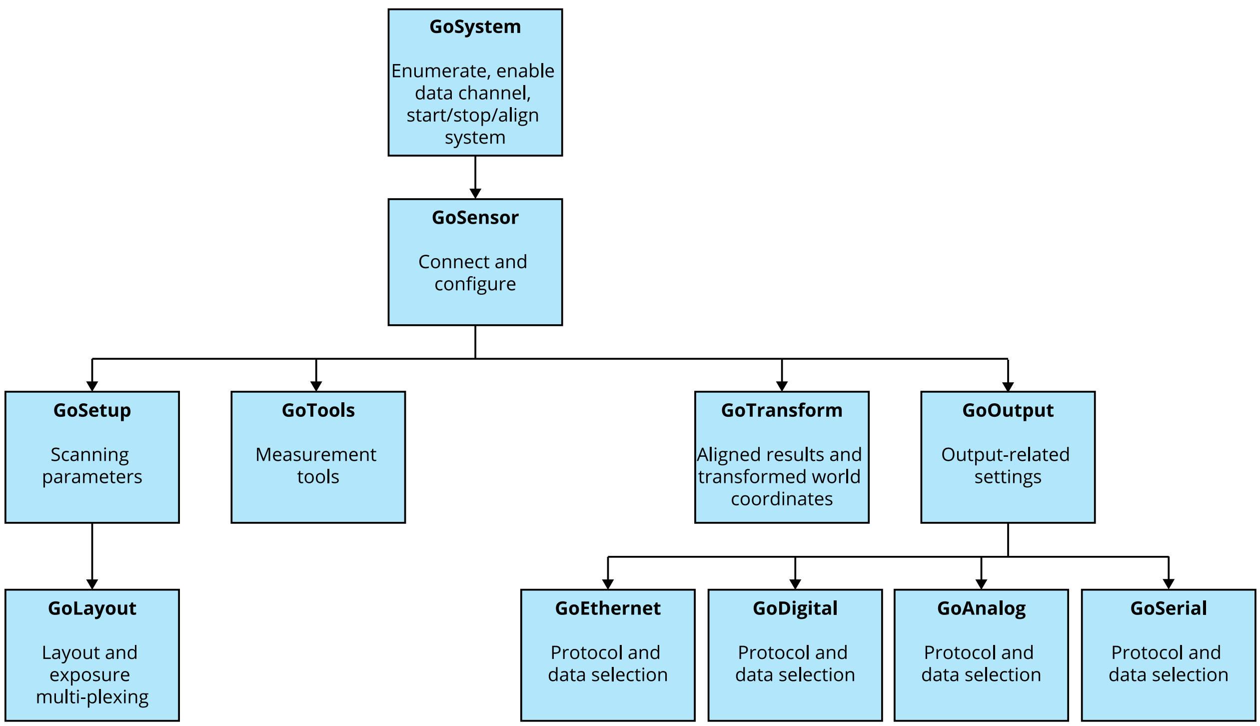
All sample projects use the environment variable *GO_SDK_4*. The environment variable should point to the *GO_SDK* directory, for example, *C:\14400-4.0.9.156_SOFTWARE_GO_SDK\GO_SDK*.

Header Files

Header files are referenced with GoSdk as the source directory, for example: #include <GoSdk/GoSdk.h>. The SDK header files also reference files from the *kApi* directory. The include path must be set up for both the GoSdk and the kApi directories. For example, the sample projects set the include path to $(GO_SDK_4)\Gocator\GoSdk and $(GO_SDK_4)\Platform\kApi.

Class Hierarchy

This section describes the class hierarchy of the Gocator 4.0 SDK.



GoSystem

The *GoSystem* class is the top-level class in Gocator 4.x. Multiple sensors can be enabled and connected in one *GoSystem*. Only one *GoSystem* object is required for multi-sensor control.

Refer to the *How To Use The Open Source SDK To Fully Control A Gocator Multi-sensor System* how-to guide on http://lmi3d.com/resources/gocator/ for details on how to control and operate a multi-sensor system using the SDK.

All objects that are explicitly created by the user or passed via callbacks should be destroyed by using the *GoDestroy* function.

GoSensor

GoSensor represents a physical sensor. If the physical sensor is the Main sensor in a dual-sensor setup, it can be used to configure settings that are common to both sensors.

GoSetup

The *GoSetup* class represents a device's configuration. The class provides functions to get or set all of the settings available in the Gocator web interface.

GoSetup is included inside *GoSensor*. It encapsulates scanning parameters, such as exposure, resolution, spacing interval, etc. For parameters that are independently controlled for Main and Buddy sensors, functions accept a role parameter.

GoLayout

The *GoLayout* class represents layout-related sensor configuration.

GoTools

The *GoTools* class is the base class of the measurement tools. The class provides functions for getting and setting names, retrieving measurement counts, etc.

GoTransform

The *GoTransform* class represents a sensor transformation and provides functions to get and set transformation information, as well as encoder-related information.

GoOutput

The *GoOutput* class represents output configuration and provides functions to get the specific types of output (Analog, Digital, Ethernet, and Serial). Classes corresponding to the specific types of output (*GoAnalog*, *GoDigital*, *GoEthernet*, and *GoSerial*) are available to configure these outputs.

Data Types

The following sections describe the types used by the SDK and the kApi library.

Value Types

GoSDK is built on a set of basic data structures, utilities, and functions, which are contained in the *kApi* library.

The following basic value types are used by the *kApi* library.

Value Data Types

Type	Description
k8u	8-bit unsigned integer
k16u	16-bit unsigned integer
k16s	16-bit signed integer

Type	Description
k32u	32-bit unsigned integer
k32s	32-bit signed integer
k64s	64-bit signed integer
k64u	64-bit unsigned integer
k64f	64-bit floating number
kBool	Boolean, value can be kTRUE or kFALSE
kStatus	Status, value can be kOK or kERROR
kIpAddress	IP address

Output Types

The following output types are available in the SDK.

Output Data Types

Data Type	Description
GoDataMsg	Represents a base message sourced from the data channel. See *GoDataSet Type* (below) for more information.
GoMeasurementMsg	Represents a message containing a set of GoMeasurementData objects.
GoProfileIntensityMsg	Represents a data message containing a set of profile intensity arrays.
GoProfileMsg	Represents a data message containing a set of profile arrays.
GoRangeIntensityMsg	Represents a data message containing a set of range intensity data.
GoRangeMsg	Represents a data message containing a set of range data.
GoResampledProfileMsg	Represents a data message containing a set of resampled profile arrays.
GoStampMsg	Represents a message containing a set of acquisition stamps.
GoSurfaceIntensityMsg	Represents a data message containing a surface intensity array.
GoSurfaceMsg	Represents a data message containing a surface array.
GoVideoMsg	Represents a data message containing a video image.

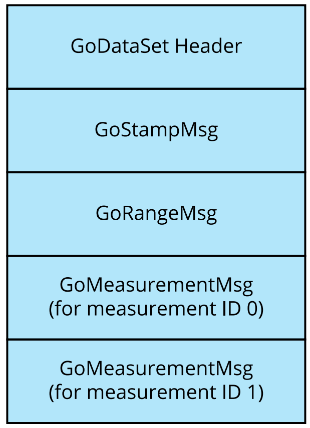
Refer to the *GoSdkSamples* sample code for examples of acquiring data using these data types.

> See *Setup and Locations* (page 199) for more information on the code samples.

GoDataSet Type

Data are passed to the data handler in a *GoDataSet* object. The *GoDataSet* object is a container that can contain any type of data, including scan data (ranges), measurements, and results from various operations. Data inside the *GoDataSet* object are represented as messages.

The following illustrates the content of a *GoDataSet* object of a range mode setup with two measurements.



After receiving the *GoDataSet* object, you should call *GoDestroy* to dispose the *GoDataSet* object. You do not need to dispose objects within the *GoDataSet* object individually.

> All objects that are explicitly created by the user or passed via callbacks should be destroyed by using the *GoDestroy* function.

Measurement Values and Decisions

Measurement values and decisions are 32-bit signed values (k32s). See *Value Types* (page 201) for more information on value types.

The following table lists the decisions that can be returned.

Measurement Decisions

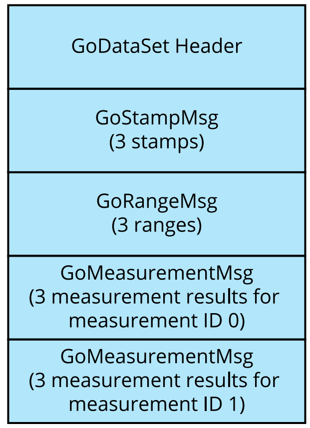
Decision	Description
1	The measurement value is between the maximum and minimum decision values. This is a pass decision.
0	The measurement value is outside the maximum and minimum. This is a fail decision.
-1	The measurement is invalid (for example, the target is not within range). Provides the reason for the failure.
-2	The tool containing the measurement is anchored and has received invalid measurement data from one of its anchors. Provides the reason for the failure.

Refer to the *SetupMeasurement* example for details on how to add and configure tools and measurements. Refer to the *ReceiveMeasurement* example for details on how to receive measurement decisions and values.

> You should check a decision against <=0 for failure or invalid measurement.

Batching

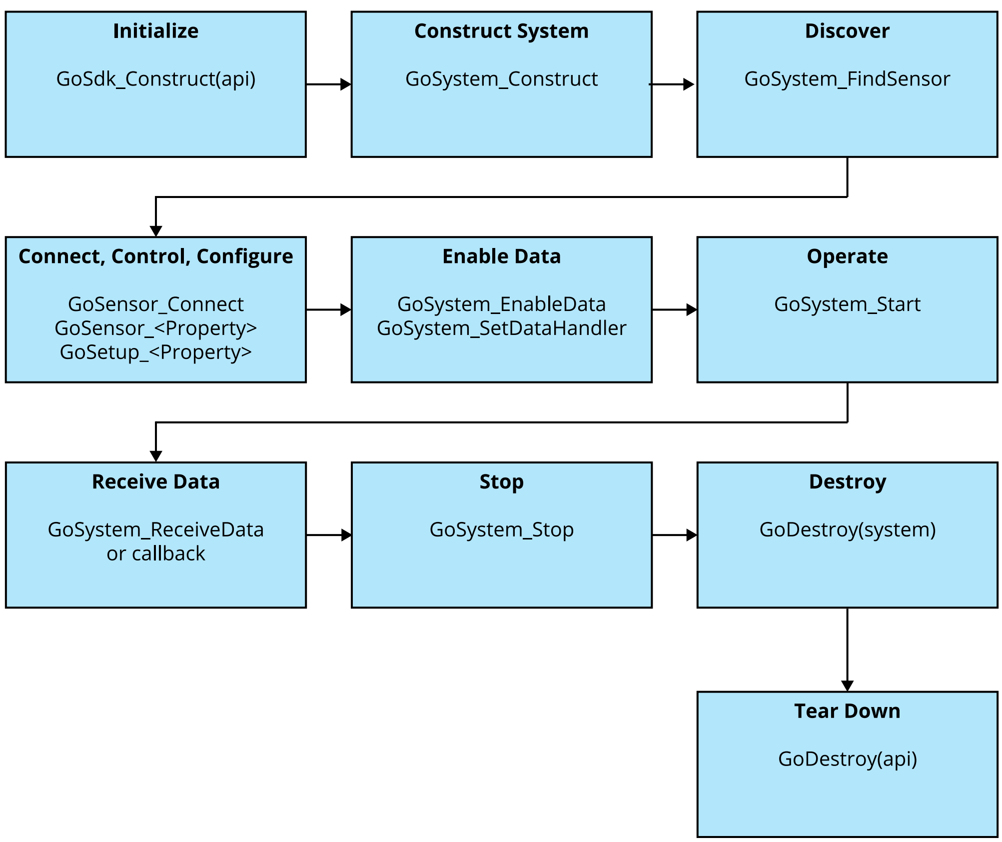
One *GoDataSet* object can contain data from multiple frames. Each message has a *Count* property that specifies how many frames of data are included. The following illustrates the data structure when three frames of data are contained inside a *GoDataSet* object.



The batching size is dynamically adjusted to ensure the sensor's CPU keeps up with the messages delivered with the shortest latency.

Operation Workflow

Applications created using the SDK typically use the following programming sequence:



See *Setup and Locations* (page 199) for more information on the code samples referenced below.

> Sensors must be connected before the system can enable the data channel.

> All data functions are named *Go<Object>_<Function>*, for example, *GoSensor_Connect*. For property access functions, the convention is *Go<Object>_<Property Name>* for reading the property and *Go<Object>_Set<Property Name>* for writing it, for example, *GoMeasurement_DecisionMax* and *GoMeasurement_SetDecisionMax*, respectively.

Initialize GoSdk API Object

Before the SDK can be used, the *GoSdk* API object must be initialized by calling *GoSdk_Construct(api)*:

```
kAssembly api = kNULL;
if ((status = GoSdk_Construct(&api)) != kOK)
{
    printf("Error: GoSdk_Construct:%d\n", status);
    return;
}
```

When the program finishes, call *GoDestroy(api)* to destroy the API object.

Discover Sensors

Sensors are discovered when *GoSystem* is created, using *GoSystem_Construct*. You can use *GoSystem_SensorCount* and *GoSystem_SensorAt* to iterate all the sensors that are on the network.

GoSystem_SensorCount returns the number of sensors physically in the network.

Alternatively, use *GoSystem_FindSensorById* or *GoSystem_FindSensorByIpAddress* to get the sensor by ID or by IP address.

Refer to the *Discover* example for details on iterating through all sensors. Refer to other examples for details on how to get a sensor handle directly from IP address.

Connect Sensors

Sensors are connected by calling *GoSensor_Connect*. You must first get the sensor object by using *GoSystem_SensorAt*, *GoSystem_FindSensorById*, or *GoSystem_FindSensorByIpAddress*.

Configure Sensors

Some configuration is performed using the *GoSensor* object, such as managing jobs, uploading and downloading files, scheduling outputs, setting alignment reference, etc. Most configuration is however performed through the *GoSetup* object, for example, setting scan mode, exposure, exposure mode, active area, speed, alignment, filtering, subsampling, etc. Surface generation is configured through the *GoSurfaceGeneration* object and part detection settings are configured through the *GoPartDetection* object.

See *Class Hierarchy* (page 200) for information on the different objects used for configuring a sensor. Sensors must be connected before they can be configured.

Refer to the *Configure* example for details on how to change settings and to switch, save, or load jobs. Refer to the *BackupRestore* example for details on how to back up and restore settings.

Enable Data Channels

Use *GoSystem_EnableData* to enable the data channels of all connected sensors. Similarly, use *GoSystem_EnableHealth* to enable the health channels of all connected sensors.

Perform Operations

Operations are started by calling *GoSystem_Start*, *GoSystem_StartAlignment*, and *GoSystem_StartExposureAutoSet*.

Refer to the *StationaryAlignment* and *MovingAlignment* examples for details on how to perform alignment operations. Refer to the *ReceiveRange*, *ReceiveProfile*, and *ReceiveWholePart* examples for details on how to acquire data.

Example: Configuring and starting a sensor with the Gocator API

```
#include <GoSdk/GoSdk.h>

void main()
{
    kIpAddress ipAddress;
    GoSystem system = kNULL;
    GoSensor sensor = kNULL;
    GoSetup setup = kNULL;

    //Construct the GoSdk library.
    GoSdk_Construct(&api);
    //Construct a Gocator system object.
    GoSystem_Construct(&system, kNULL);
    //Parse IP address into address data structure
    kIpAddress_Parse(&ipAddress, SENSOR_IP);
    //Obtain GoSensor object by sensor IP address
    GoSystem_FindSensorByIpAddress(system, &ipAddress, &sensor)
    //Connect sensor object and enable control channel
    GoSensor_Connect(sensor);
    //Enable data channel
    GoSensor_EnableData(system, kTRUE)
    //[Optional] Setup callback function to receive data asynchronously
    //GoSystem_SetDataHandler(system, onData, &contextPointer)
    //Retrieve setup handle
    setup = GoSensor_Setup(sensor);
    //Reconfigure system to use time-based triggering.
    GoSetup_SetTriggerSource(setup, GO_TRIGGER_TIME);
    //Send the system a "Start" command.
    GoSystem_Start(system);
```

```
    //Data will now be streaming into the application
    //Data can be received and processed asynchronously if a callback function has been
    //set (recommended)
    //Data can also be received and processed synchronously with the blocking call
    //GoSystem_ReceiveData(system, &dataset, RECEIVE_TIMEOUT)
    //Send the system a "Stop" command.
    GoSystem_Stop(system);


    //Free the system object.
    GoDestroy(system);

    //Free the GoSdk library
    GoDestroy(api);
}
```

Limiting Flash Memory Write Operations

Several operations and Gocator SDK functions write to the Gocator's flash memory. The lifetime of the flash memory is limited by the number of write cycles. Therefore it is important to avoid frequent write operation to the Gocator's flash memory when you design your system with the Gocator SDK.

Power loss during flash memory write operation will also cause Gocators to enter rescue mode.

This topic applies to all Gocator sensors.

Gocator SDK Write-Operation Functions

Name	Description
GoSensor_Restore	Restores a backup of sensor files.
GoSensor_RestoreDefaults	Restores factory default settings.
GoSensor_CopyFile	Copies a file within the connected sensor. The flash write operation does not occur if GoSensor_CopyFile function is used to load an existing job file. This is accomplished by specifying "_live" as the destination file name.
GoSensor_DeleteFile	Deletes a file in the connected sensor.
GoSensor_SetDefaultJob	Sets a default job file to be loaded on boot.
GoSensor_UploadFile	Uploads a file to the connected sensor.
GoSensor_Upgrade	Upgrades sensor firmware.
GoSystem_StartAlignment	When alignment is performed with alignment reference set to fixed, flash memory is written immediately after alignment. GoSensor_

Name	Description
	SetAlignmentReference() is used to configure alignment reference.
GoSensor_SetAddress	Configures a sensor's network address settings.
GoSensor_ChangePassword	Changes the password associated with the specified user account.

System created using the SDK should be designed in a way that parameters are set up to be appropriate for various application scenarios. Parameter changes not listed above will not invoke flash memory write operations when the changes are not saved to a file using the GoSensor_CopyFile function. Fixed alignment should be used as a means to attach previously conducted alignment results to a job file, eliminating the need to perform a new alignment.

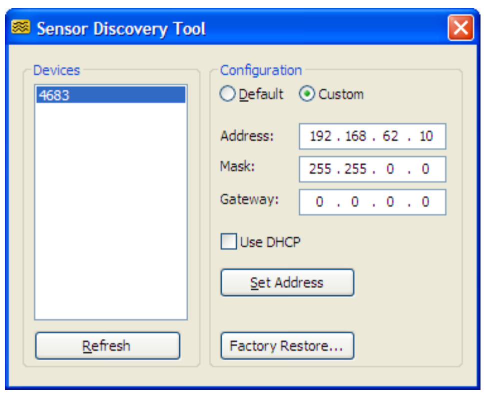
Tools

The following sections describe the tools you can use with a Gocator.

Sensor Recovery Tool

If a sensor's network address or administrator password is forgotten, the sensor can be discovered on the network and/or restored to factory defaults by using a special software tool called the Sensor Discovery tool. This software tool can be obtained from the downloads area of the LMI Technologies website: http://www.lmi3D.com.

After downloading the tool package [14405-x.x.x.x_SOFTWARE_GO_Tools.zip], unzip the file and run the Sensor Discovery Tool [bin>win32>kDiscovery.exe].



Any sensors that are discovered on the network will be displayed in the Devices list.

To change the network address of a sensor:

1. To change the network address of a sensor.
2. Select the **Custom** option.
3. Enter the new network address information.
4. Press the Set Address button.

To restore a sensor to factory defaults:

1. Select the sensor serial number in the **Devices** list.
2. Press the **Factory Restore...** button.
   Confirm when prompted.

The Sensor Discovery tool uses UDP broadcast messages to reach sensors on different subnets. This enables the Sensor Discovery tool to locate and re-configure sensors even when the sensor IP address or subnet configuration is unknown.

Troubleshooting

Review the guidance in this chapter if you are experiencing difficulty with a Gocator sensor system. See *Return Policy* (page 250) for further assistance if the problem that you are experiencing is not described in this section.

Mechanical/Environmental

The sensor is warm.

- It is normal for a sensor to be warm when powered on. A Gocator sensor is typically 15° C warmer than the ambient temperature.

Connection

When attempting to connect to the sensor with a web browser, the sensor is not found (page does not load).

- Verify that the sensor is powered on and connected to the client computer network. The Power Indicator LED should illuminate when the sensor is powered.
- Check that the client computer's network settings are properly configured.
- Ensure that the latest version of Flash is loaded on the client computer.
- Use the LMI Discovery tool to verify that the sensor has the correct network settings. See *Sensor Recovery Tool* (page 209) for more information.

When attempting to log in, the password is not accepted.

- See *Sensor Recovery Tool* (page 209) for steps to reset the password.

Laser Ranging

When the Start button or the Snapshot button is pressed, the sensor does not emit laser light.

- Ensure that the sticker covering the laser emitter window (normally affixed to new sensors) has been removed.
- The laser safety input signal may not be correctly applied. See *Specifications* (page 213) for more information.
- The exposure setting may be too low. See *Exposure* (page 72) for more information on configuring exposure time.
- Use the Snapshot button instead of the Start button to capture a laser range. If the laser flashes when you use the **Snapshot** button, but not when you use the **Start** button, then the problem could be related to triggering. See *Triggers* (page 64) for information on configuring the trigger source.

The sensor emits laser light, but the Range Indicator LED does not illuminate and/or points are not displayed in the Data Viewer.

- Verify that the measurement target is within the sensor's field of view and measurement range. See *Specifications* (page 213) to review the measurement specifications for your sensor model.
- Check that the exposure time is set to a reasonable level.See *Exposure* (page 72) for more information on configuring exposure time.

Performance

The sensor CPU level is near 100%.

- Consider reducing the speed. If you are using a time or encoder trigger source, see *Triggers* (page 64) for information on reducing the speed. If you are using an external input or software trigger, consider reducing the rate at which you apply triggers.
- Review the measurements that you have programmed and eliminate any unnecessary measurements.

Specifications

The following sections describe the specifications of the Gocator and its associated hardware.

Gocator 1300 Series

The Gocator 1300 series consists of the sensor models defined below.

MODEL	1320	1340	1350	1365	1370	1390
Clearance Distance (mm)	40	162.5	200	562	237.5	500
Measurement Range (MR) (mm)	20	95	200	375	412.5	2000
Linearity Z (+/- % of MR)	0.05	0.05	0.05	0.11	0.07	0.10
Linearity Z (+/- mm)	0.010	0.05	0.100	0.4	0.3	2.0
Resolution Z (mm)	0.0004 - 0.0004	0.0005 - 0.0010	0.0015 - 0.0035	0.0025 - 0.0040	0.0025 - 0.0070	0.0250 - 0.0600
Spot Size (mm)	0.11	0.37	0.50	1.80	0.90	2.60
Recommended Laser Class	3R	3B	3B	3B	3B	3B
Other Laser Class		3R				
Recommended Package Dimensions (mm)	Side Mount 30x120x149	Side Mount 30x120x149	Side Mount 30x120x149	Side Mount 30x120x220	Side Mount 30x120x149	Side Mount 30x120x277
Other Package Dimensions (mm)			Top Mount 49x75x162			
Weight (kg)	0.8	0.8	0.75 / 0.8	1.1	0.8	1.4

Optical models, laser classes, and packages can be customized. Contact LMI for more details.

Specifications stated are based on standard laser classes. Linearity Z and Resolution Z may vary for other laser classes.

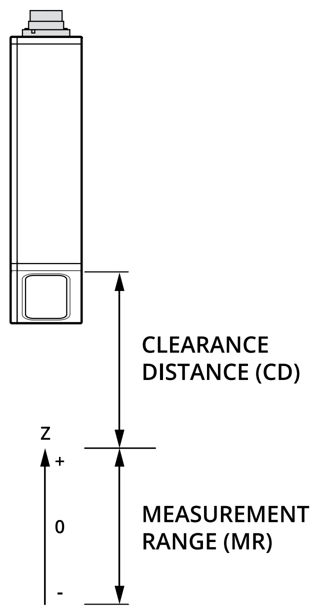
All specification measurements are performed on LMI's standard calibration target (a diffuse, painted white surface).

Linearity Z is the worst case difference in average height measured, compared to the actual position over the measurement range.

Resolution Z is the maximum variability of height measurements across multiple frames, with 95% confidence.

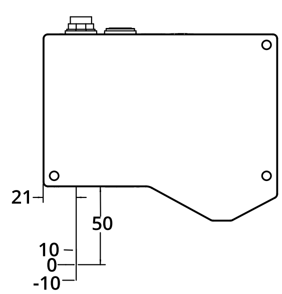
See *Resolution and Accuracy* (page 37) for more information.

ALL 1300 SERIES MODELS	
Scan Rate	32kHz
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Laser Safety Enable, Trigger
Outputs	2x Digital Output, RS-485 Serial, Selcom Serial, 1x Analog Output (4 - 20 mA)
Input Voltage (Power)	+24 to +48 VDC (13 Watts); Ripple +/- 10%
Housing	Gasketed Aluminum Enclosure, IP67
Operating Temp.	0 to 50° C
Storage Temp.	-30 to 70 ° C

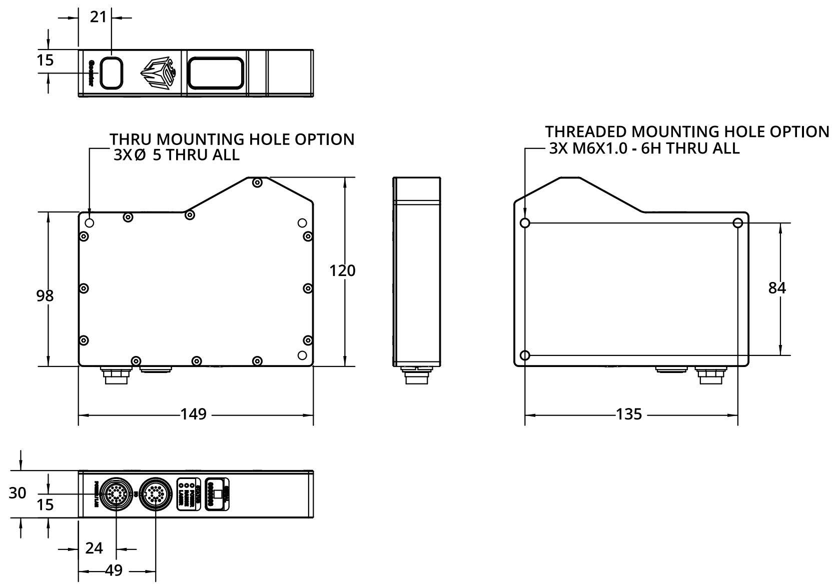


Gocator 1320 (Side Mount Package)

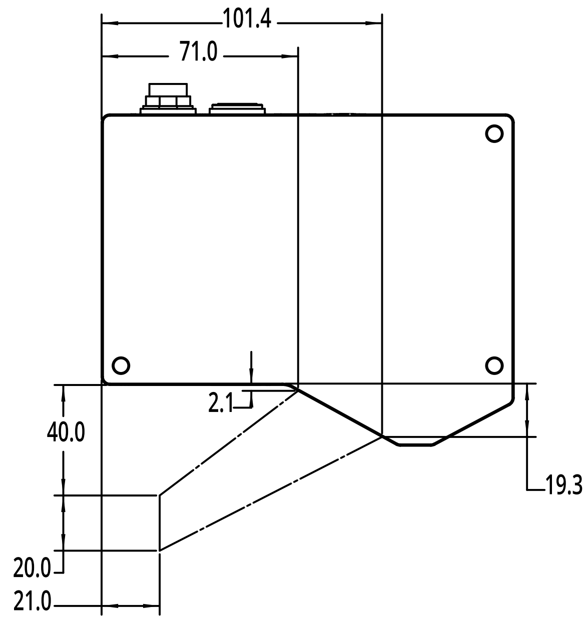
Field of View / Measurement Range



Dimensions

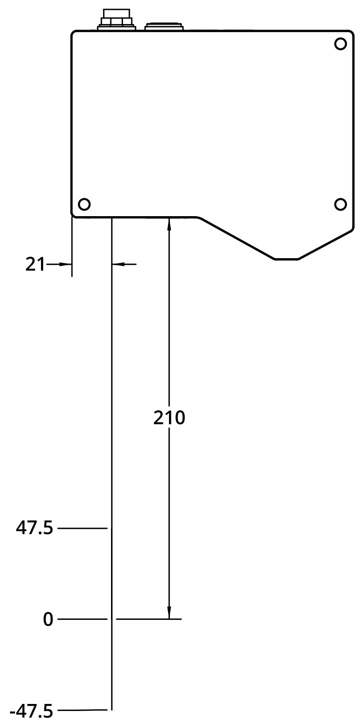


Envelope

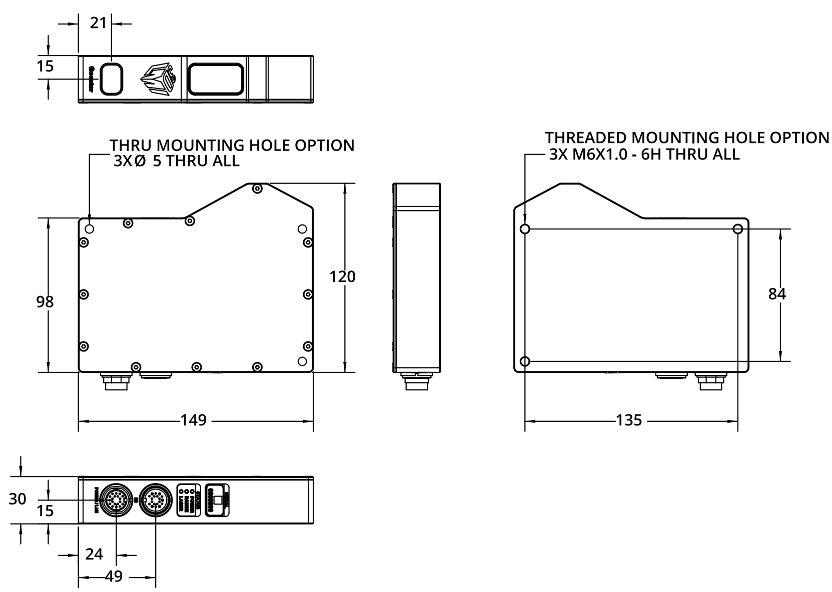


Gocator 1340 (Side Mount Package)

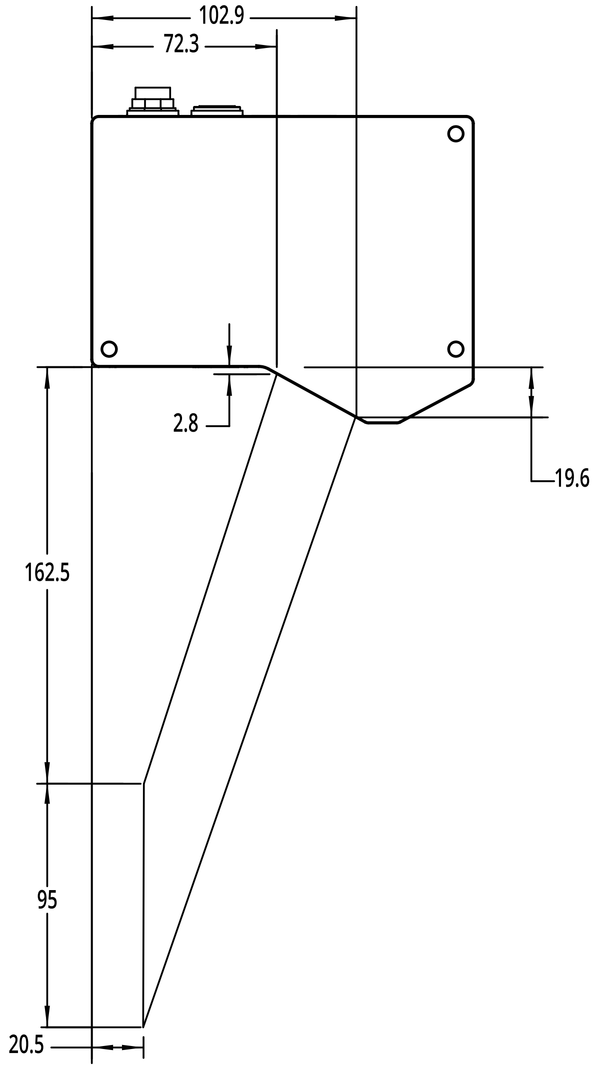
Field of View / Measurement Range



Dimensions

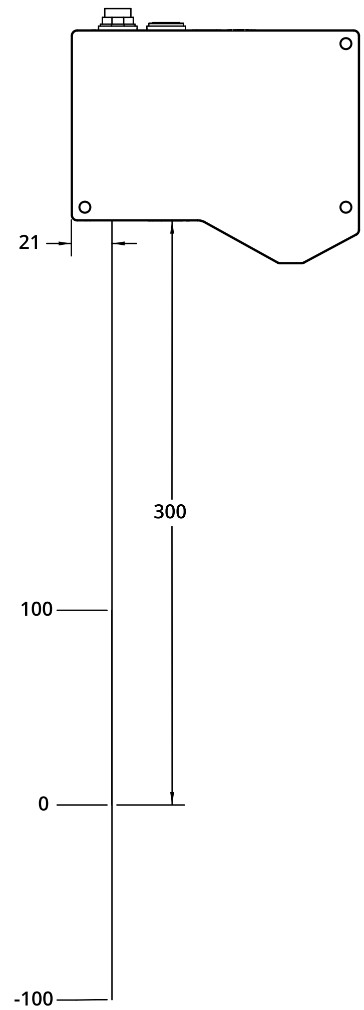


Envelope

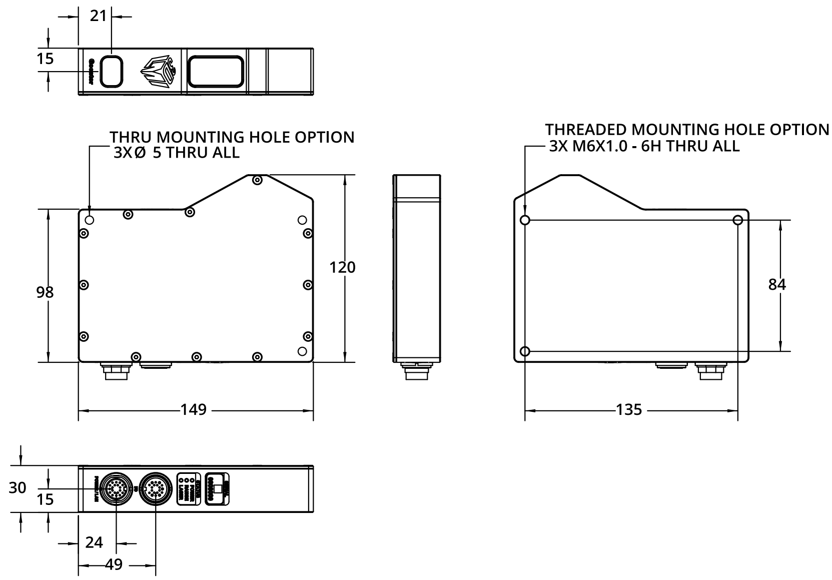


Gocator 1350 (Side Mount Package)

Field of View / Measurement Range

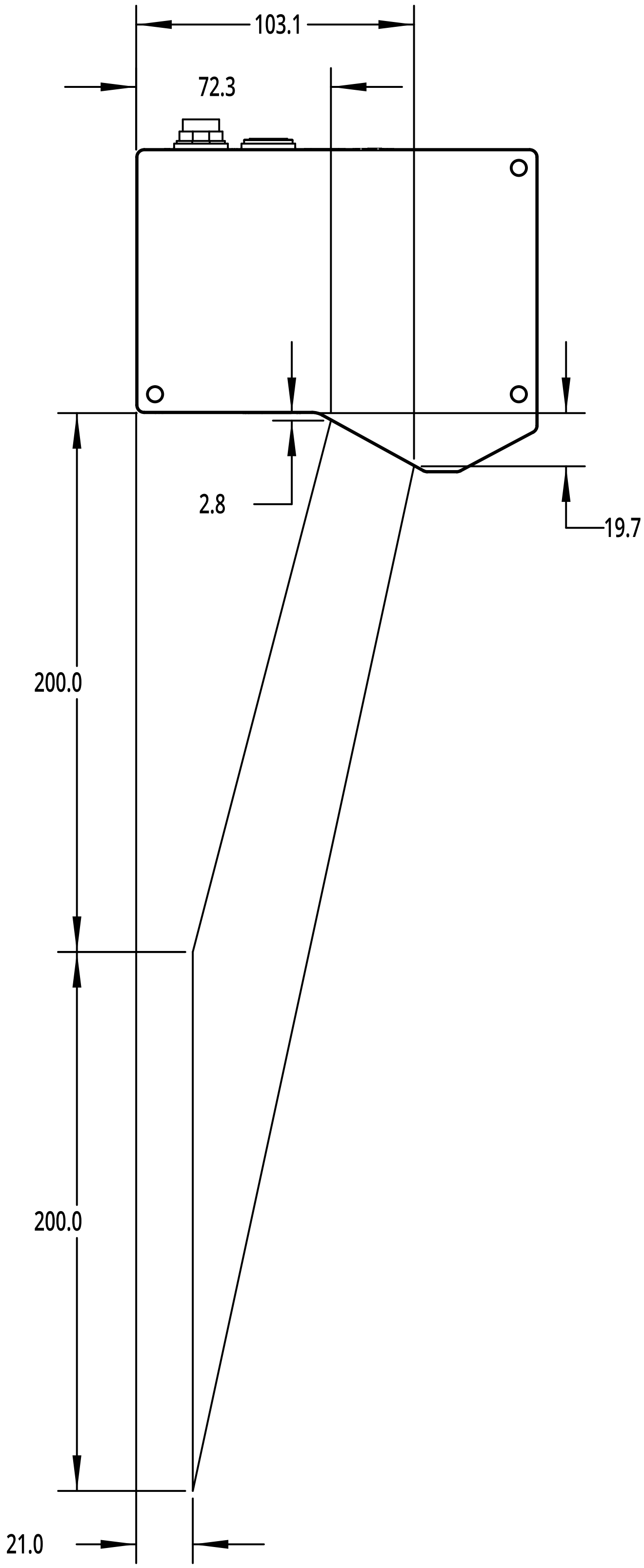


Dimensions



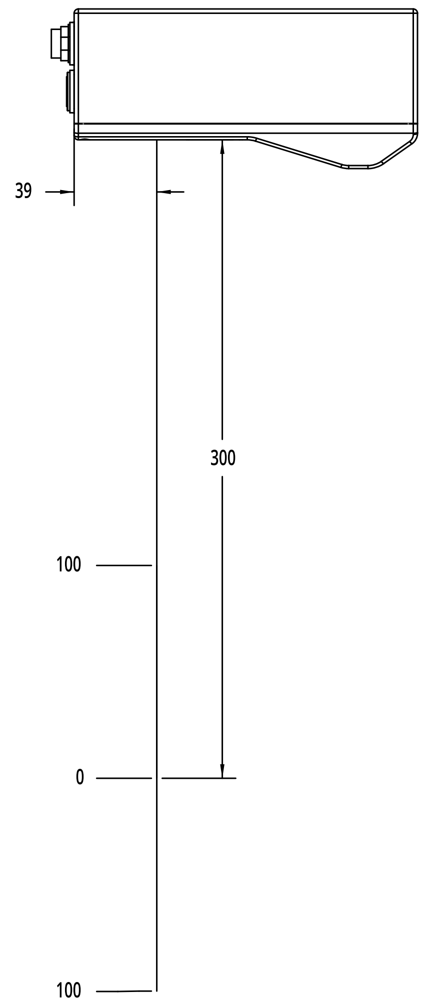
21
15
THRU MOUNTING HOLE OPTION
3XØ 5 THRU ALL
THREADED MOUNTING HOLE OPTION
3X M6X1.0 - 6H THRU ALL
98
120
84
149
135
30
15
24
49

Envelope

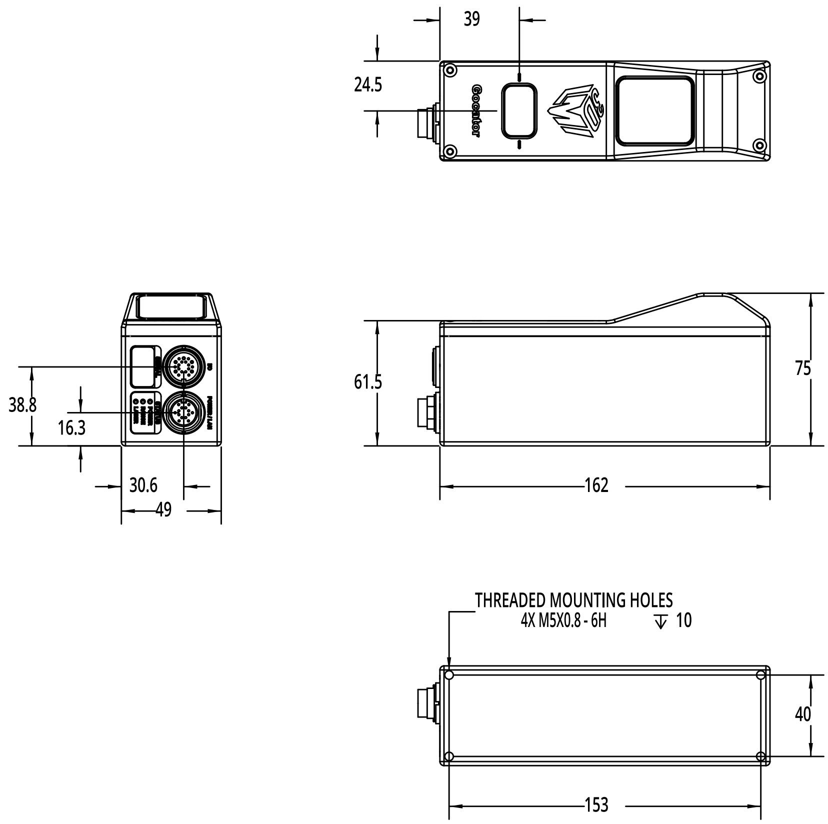


Gocator 1350 (Top Mount Package)

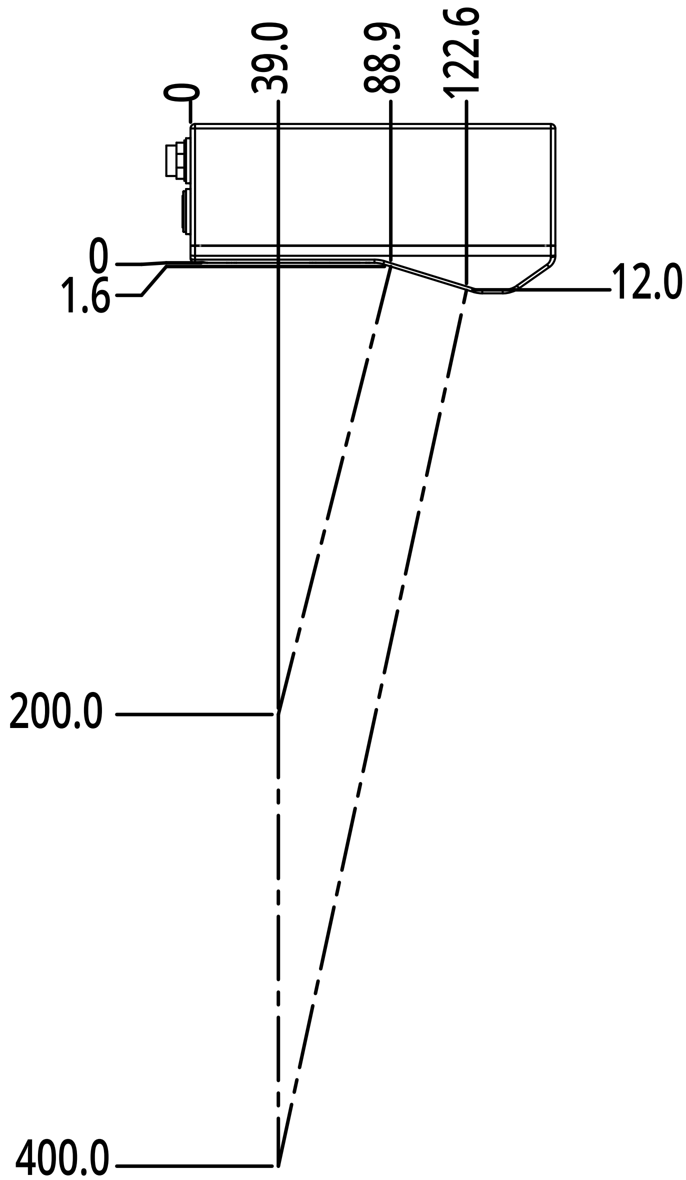
Field of View / Measurement Range



Dimensions

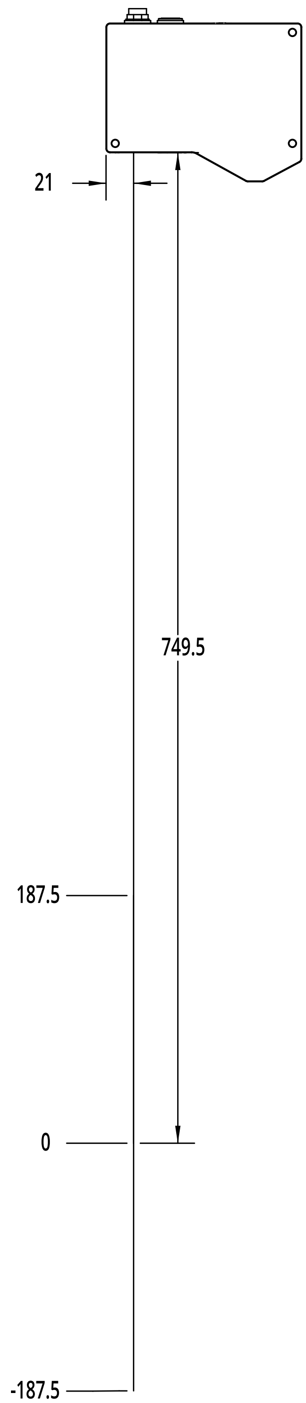

39
24.5
38.8
16.3
30.6
49
61.5
75
162
THREADED MOUNTING HOLES
4X M5X0.8 - 6H ↧ 10
40
153

Envelope

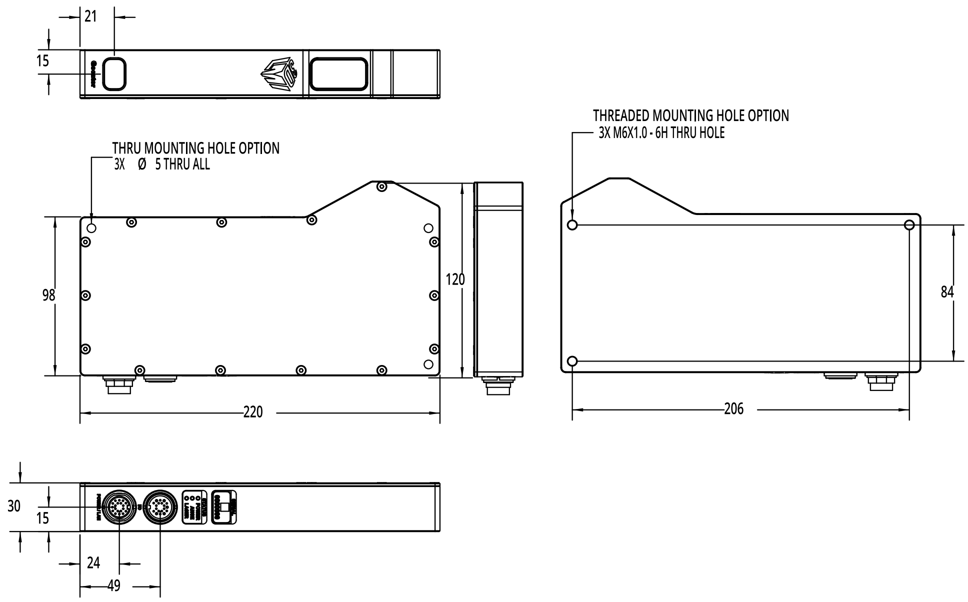


Gocator 1365 (Side Mount Package)

Field of View / Measurement Range

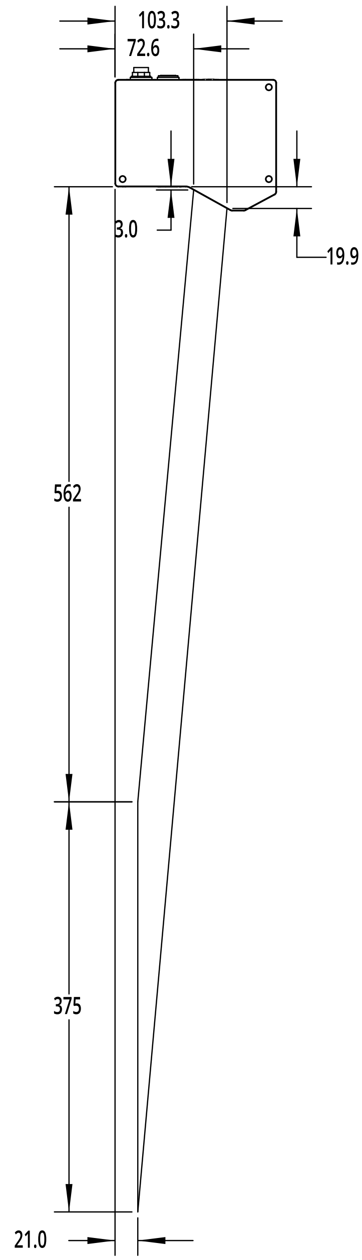


Dimensions

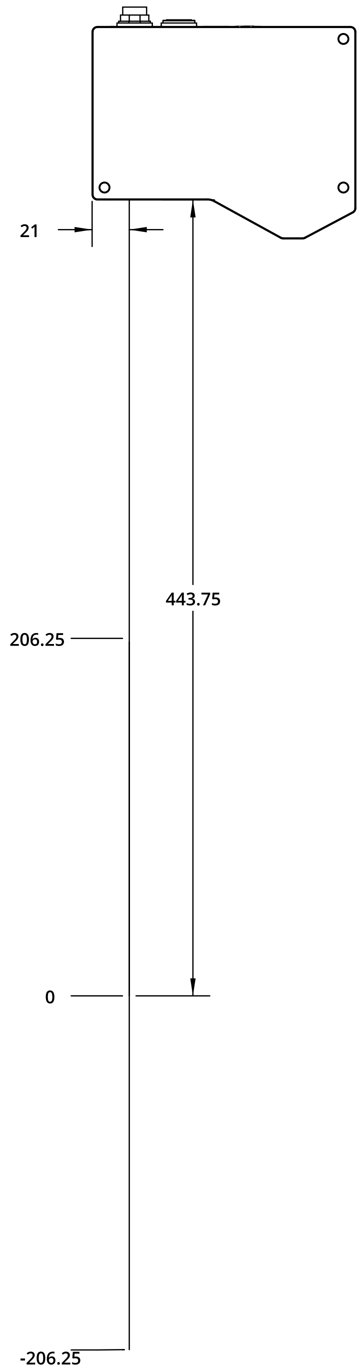


21
15
THRU MOUNTING HOLE OPTION
3X Ø 5 THRU ALL
THREADED MOUNTING HOLE OPTION
3X M6X1.0 - 6H THRU HOLE
98
120
84
220
206
30
15
24
49

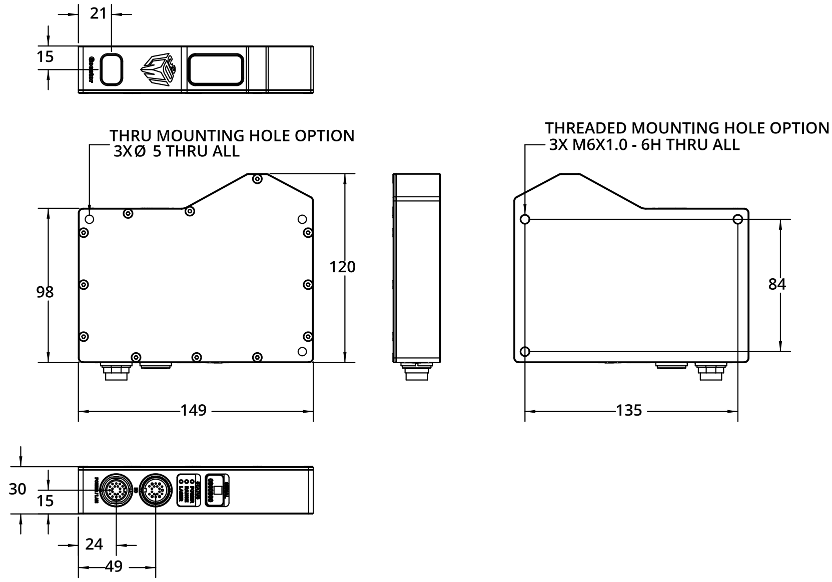
Envelope



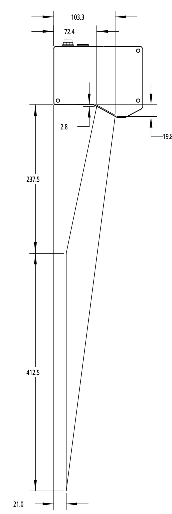
Field of View / Measurement Range



Dimensions

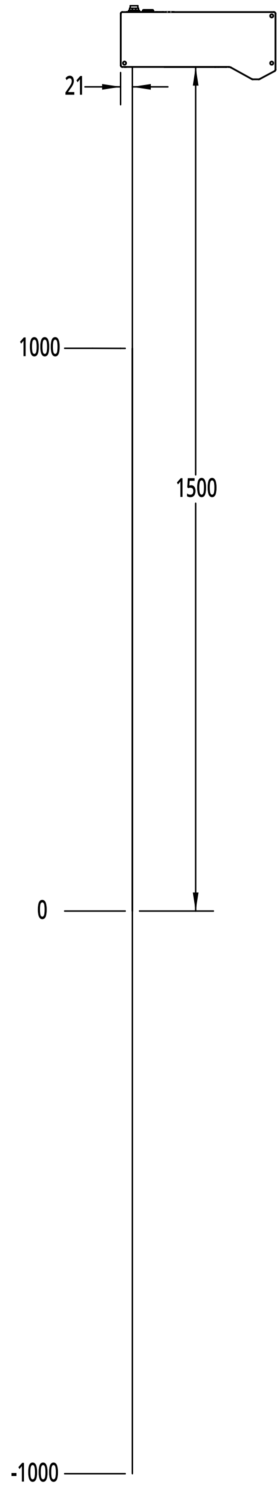


Envelope

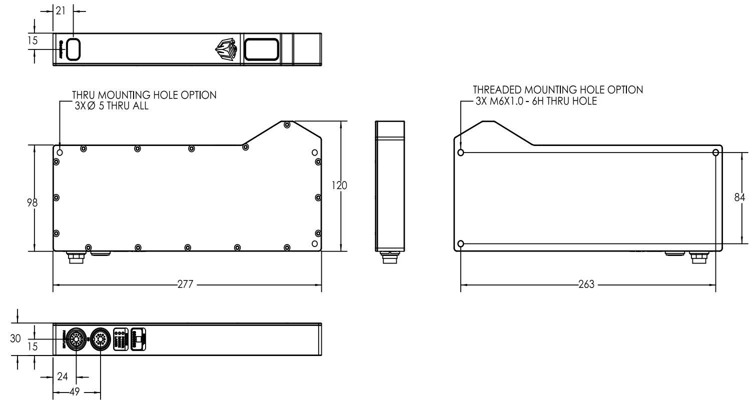


Gocator 1390 (Side Mount Package)

Field of View / Measurement Range

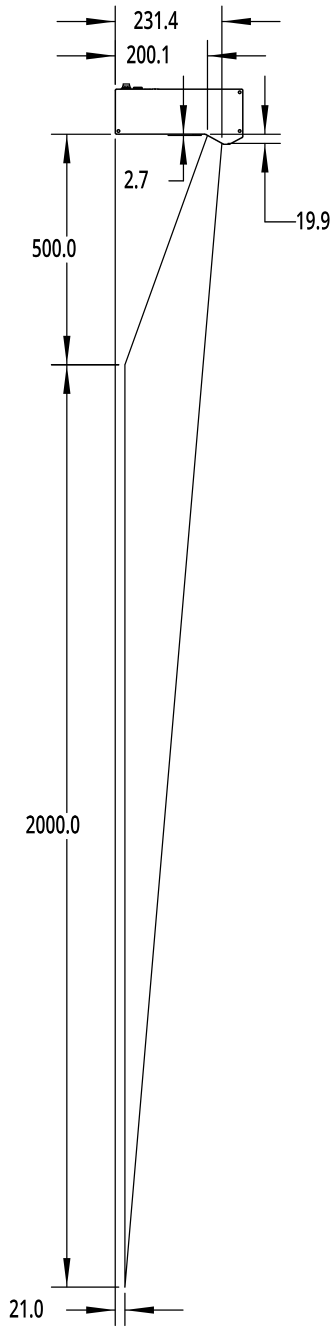


Dimensions



21
15
THRU MOUNTING HOLE OPTION
3X Ø 5 THRU ALL
98
120
277
THREADED MOUNTING HOLE OPTION
3X M6X1.0 - 6H THRU HOLE
84
263
30
15
24
49

Envelope



Gocator Power/LAN Connector

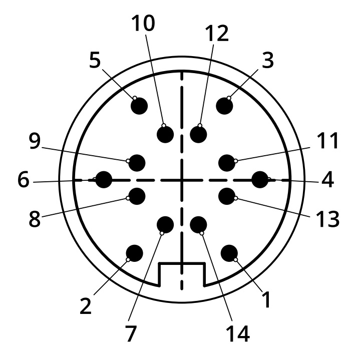
The Gocator Power/LAN connector is a 14 pin, M16 style connector that provides power input, laser safety input and Ethernet.

> This connector is rated IP67 only when a cable is connected or when a protective cap is used.

This section defines the electrical specifications for Gocator Power/LAN Connector pins, organized by function.

Gocator Power/LAN Connector Pins

Function	Pin	Lead Color on Cordset
GND_24-48V	1	White/ Orange & Black
GND_24-48V	1	Orange/ Black
DC_24-48V	2	White/ Green & Black
DC_24-48V	2	Green/ Black
Safety-	3	White/ Blue & Black
Safety+	4	Blue/ Black
Sync+	5	White/ Brown & Black
Sync-	6	Brown/ Black
Ethernet MX1+	7	White/ Orange
Ethernet MX1-	8	Orange
Ethernet MX2+	9	White/ Green
Ethernet MX2-	10	Green
Ethernet MX3-	11	White/ Blue
Ethernet MX3+	12	Blue
Ethernet MX4+	13	White/ Brown
Ethernet MX4-	14	Brown



View: Looking into the connector ***on*** *the sensor*

Two wires are connected to the ground and power pins.

Grounding Shield

The grounding shield should be mounted to the earth ground.

Power

Positive voltage is applied to DC_24-48V. See *Gocator 1300 Series* (page 214) for the sensor's power requirement. Ground is applied to GND_24-48VDC.

Power requirements

Function	Pins	Min	Max
DC_24-48V	2	24 V	48 V
GND_24-48VDC	1	0 V	0 V

Laser Safety Input

The Safety_in+ signal should be connected to a voltage source in the range listed below. The Safety_in- signal should be connected to the ground/common of the source supplying the Safety_in+.

Laser safety requirements

Function	Pins	Min	Max
Safety_in+	4	24 V	48 V
Safety_in-	3	0 V	0 V

⚠ Confirm the wiring of Safety_in- before starting the sensor. Wiring DC_24-48V into Safety_in- may damage the sensor.

Gocator 1300 I/O Connector

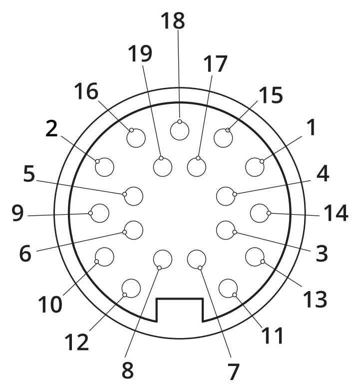
The Gocator 1300 I/O connector is a 19 pin, M16 style connector that provides encoder, digital input, digital outputs, serial output, and analog output signals.

This connector is rated IP67 only when a cable is connected or when a protective cap is used.

This section defines the electrical specifications for Gocator I/O Connector pins, organized by function.

Gocator I/O Connector Pins

Function	Pin	Lead Color on Cordset
Trigger_in+	1	Grey
Trigger_in-	2	Pink
Out_1+ (Digital Output 0)	3	Red
Out_1- (Digital Output 0)	4	Blue
Out_2+ (Digital Output 1)	5	Tan
Out_2- (Digital Output 1)	6	Orange
Encoder_A+	7	White/Brown & Black
Encoder_A-	8	Brown / Black
Encoder_B+	9	Black
Encoder_B-	10	Violet
Encoder_Z+	11	White/Green & Black
Encoder_Z-	12	Green / Black
Serial_out+	13	White
Serial_out-	14	Brown
Serial_out2+	15	Blue / Black
Serial_out2-	16	White / Blue & Black
Analog_out+	17	Green
Analog_out-	18	Yellow & Maroon/ White
Reserved	19	Maroon



*View: Looking into the connector **on** the sensor*

Grounding Shield

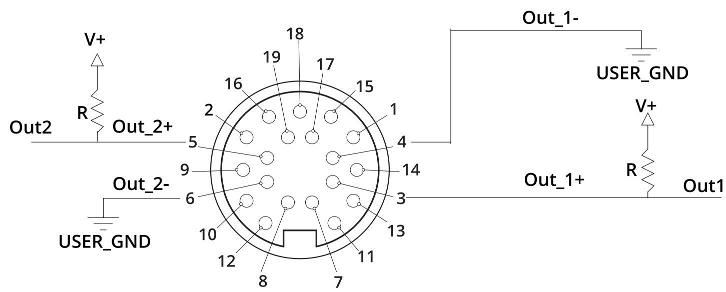
The grounding shield should be mounted to the earth ground.

Digital Outputs

Each Gocator sensor has two optically isolated outputs. Both outputs are open collector and open emitter, this allows a variety of power sources to be connected and a variety of signal configurations.

Out_1 (Collector – Pin 3 and Emitter – Pin 4) and Out_2 (Collector – Pin 5 and Emitter Pin 6) are independent and therefore V+ and GND are not required to be the same.

Function	Pins	Max Collector Current	Max Collector–Emitter Voltage	Min Pulse Width
Out_1	3, 4	40 mA	70 V	20 us
Out_2	5, 6	40 mA	70 V	20 us

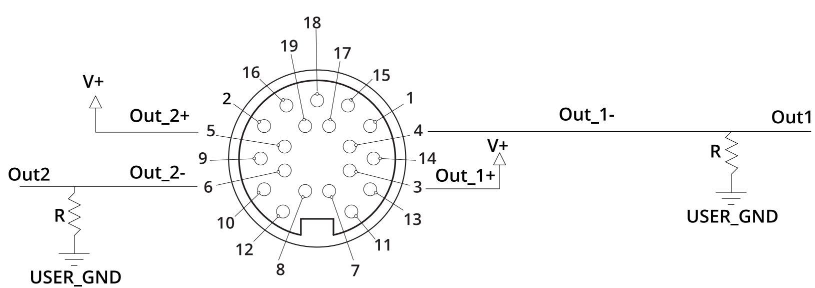


The resistors shown above are calculated by R = (V+) / 2.5 mA.

The size of the resistors is determined by power = (V+)^2 / R.

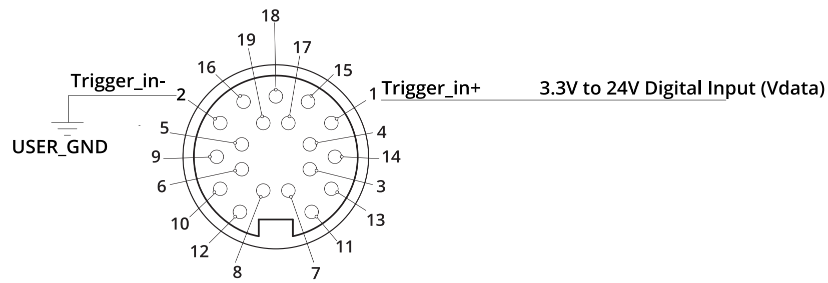
Inverting Outputs

To invert an output, connect a resistor between ground and Out_1- or Out_2- and connect Out_1+ or Out_2+ to the supply voltage. Take the output at Out_1- or Out_2-. The resistor selection is the same as what is shown above.



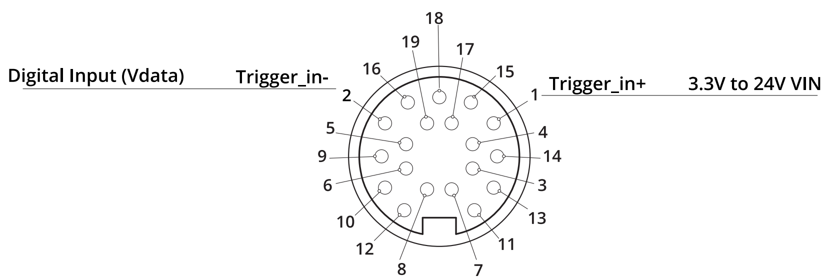
Digital Inputs

Every Gocator sensor has a single optically-isolated input. To use this input without an external resistor, supply 3.3 - 24 V to Pin 1 and GND to Pin 2.



Active High

If the supplied voltage is greater than 24 V, connect an external resistor in series to Pin 1. The resistor value should be R = [(Vin-1.2V)/10mA]-680.



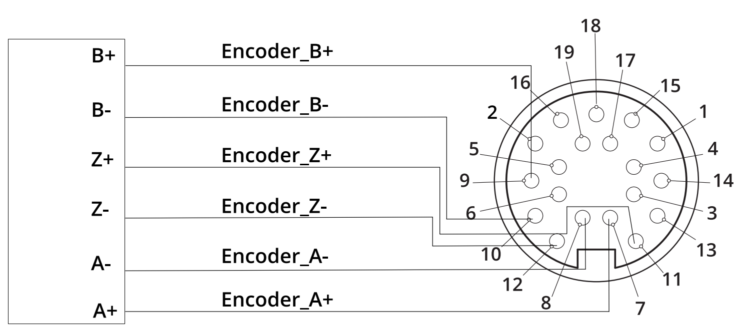
Active Low

To assert the signal, the digital input voltage should be set to draw a current of 3 mA to 40 mA from Trigger_In+. The current that passes through Trigger_In+ is I = (Vin – 1.2 – Vdata) / 680. To reduce noise sensitivity, we recommend leaving a 20% margin for current variation (i.e., uses a digital input voltage that draws 4mA to 25mA).

Function	Pins	Min Voltage	Max Voltage	Min Current	Max Current	Min Pulse Width
Trigger_in	1, 2	3.3 V	24 V	3 mA	40 mA	20 us

Encoder Input

Encoder input is provided by an external encoder and consists of three RS-485 signals. These signals are connected to Encoder_A, Encoder_B, and Encoder_Z.



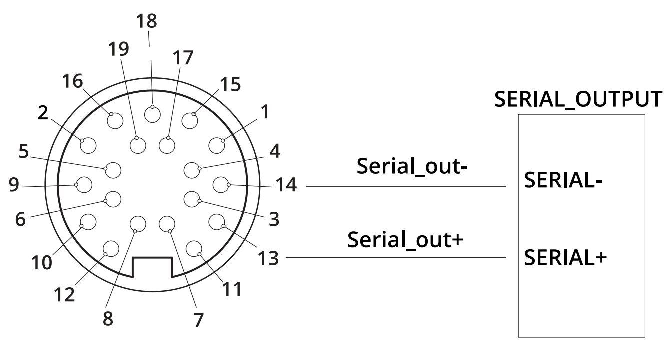
Function	Pins	Common Mode Voltage		Differential Threshold Voltage			Max Data Rate
		Min	Max	Min	Typ	Max	
Encoder_A	7, 8	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Encoder_B	9, 10	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Ecnoder_Z	11, 12	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz

Gocator only supports differential RS485 signalling. Both + and - signals must be connected.

Serial Output

Serial RS-485 output is connected to Serial_out as shown below.

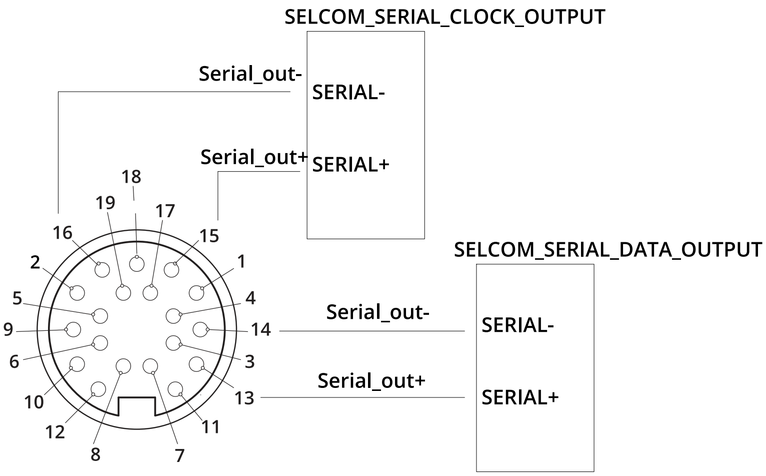
Function	Pins
Serial_out	13, 14



Selcom Serial Output

Serial RS-485 output is connected to Serial_out and Serial_out2 as shown below.

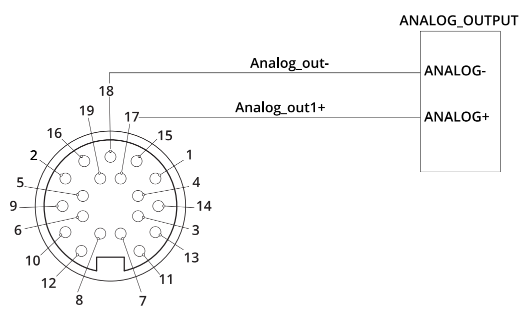
Function	Pins
Serial_out (data)	13, 14
Serial_out2 (clock)	15, 16



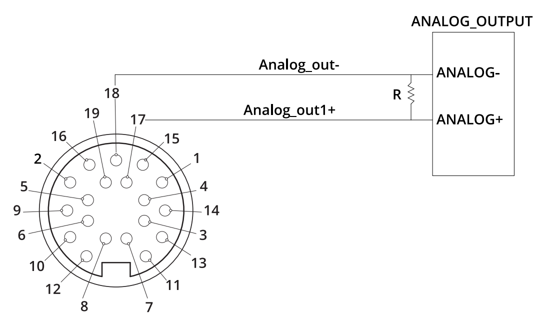
Analog Output

The Sensor I/O Connector defines one analog output interface: Analog_out.

Function	Pins	Current Range
Analog_out	17, 18	4 – 20 mA

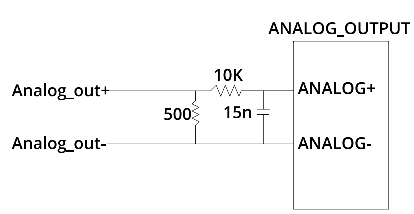


Current Mode



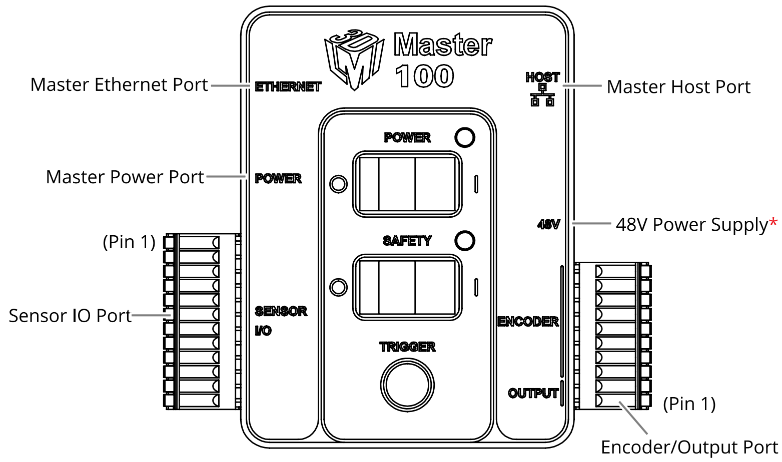
Voltage Mode

To configure for voltage output, connect a 500 Ohm ¼ Watt resistor between Analog_out+ and Analog_out- and measure the voltage across the resistor. To reduce the noise in the output, we recommend using an RC filter as shown below.



Master 100

The Master 100 accepts connections for power, safety, and encoder, and provides digital output.



*Contact LMI for information regarding this type of power supply.

Connect the Master Power port to the Gocator's Power/LAN connector using the Gocator Power/LAN to Master cordset. Connect power RJ45 end of the cordset to the Master Power port. The Ethernet RJ45 end of the cordset can be connected directly to the Ethernet switch, or connect to the Master Ethernet port. If the Master Ethernet port is used, connect the Master Host port to the Ethernet switch with a CAT5e Ethernet cable.

To use encoder and digital output, wire the Master's Gocator Sensor I/O port to the Gocator IO connector using the Gocator I/O cordset.

Sensor I/O Port Pins

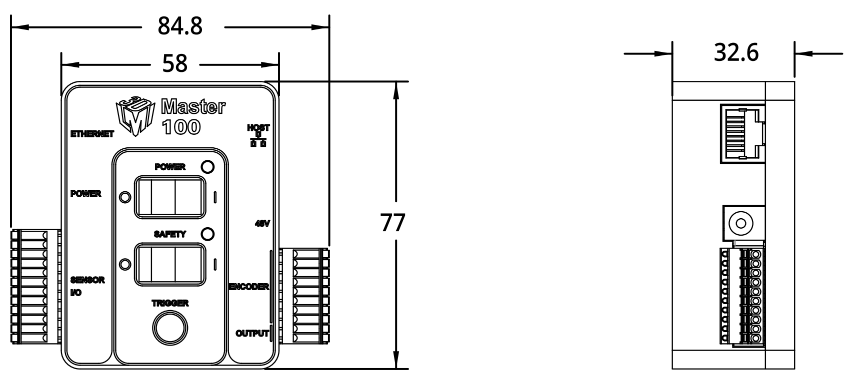
Gocator I/O Pin	Master Pin	Conductor Color
Encoder_A+	1	White/Brown & Black
Encoder_A-	2	Brown/Black
Encoder_Z+	3	White/Green & Black
Encoder_Z-	4	Green/Black
Trigger_in+	5	Grey
Trigger_in-	6	Pink
Out_1-	7	Blue
Out_1+	8	Red
Encoder_B+	11	Black
Encoder_B-	12	Violet

The rest of the wires in the Gocator I/O cordset are not used.

Encoder/Output Port Pins

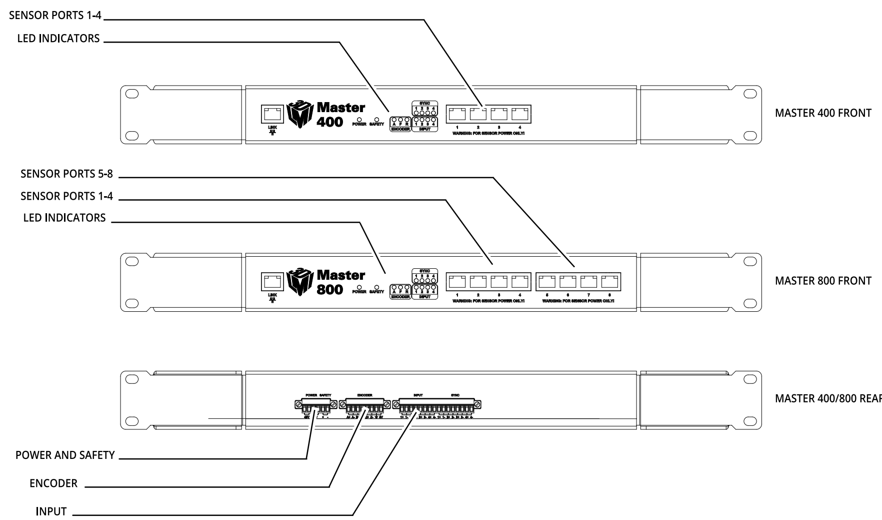
Function	Pin
Output_1+ (Digital Output 0)	1
Output_1- (Digital Output 0)	2
Encoder_Z+	3
Encoder_Z-	4
Encoder_A+	5
Encoder_A-	6
Encoder_B+	7
Encoder_B-	8
Encoder_GND	9
Encoder_5V	10

Master 100 Dimensions



Master 400/800

The Master 400/800 provides sensor power and safety interlock, and broadcasts system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND(48VDC)	3
GND(48VDC)	4
Safety Control+	5
Safety Control–	6

> The +48VDC power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

> The Safety Control requires a voltage differential 12VDC to 48VDC across the pin to enable the laser.

Digital Input (16 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3

Function	Pin
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12
Reserved	13
Reserved	14
Reserved	15
Reserved	16

> This connector does not need to be wired up for proper operation.

Encoder (8 pin connector)

Function	Pin
Encoder_A+	1
Encoder_A-	2
Encoder_B+	3
Encoder_B-	4
Encoder_Z+	5
Encoder_Z-	6
GND	7
+5VDC	8

Master 400/800 Electrical Specifications

Electrical Specifications for Master 400/800

	Master 400 / 800
Power Supply Voltage	+48VDC
Power Supply current (Max.)	10A
Power Draw (Min.)	15W
Safety Voltage	+12 to +48VDC
Encoder signal voltage range	RS485 Differential
Digital input voltage range	Logical LOW: 0 VDC to +0.1VDC Logical HIGH: +11 VDC to +22.5VDC

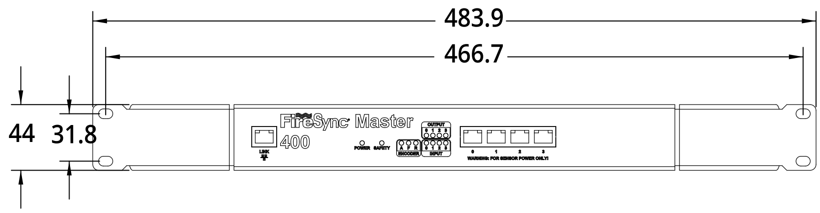
⚠	When using a Master 400/800, its chassis must be well grounded.

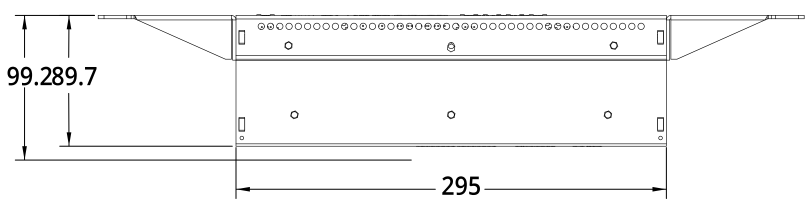
	The +48VDC power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

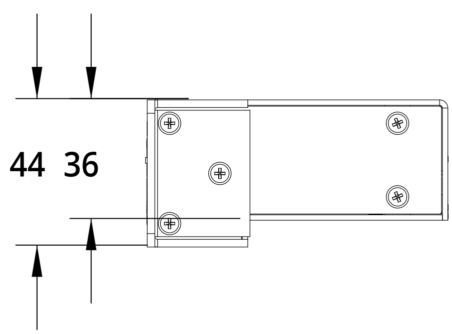
	The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements which need to be considered when calculating total system power requirements.

Master 400/800 Dimensions

The dimensions of Master 400 and Master 800 are the same.

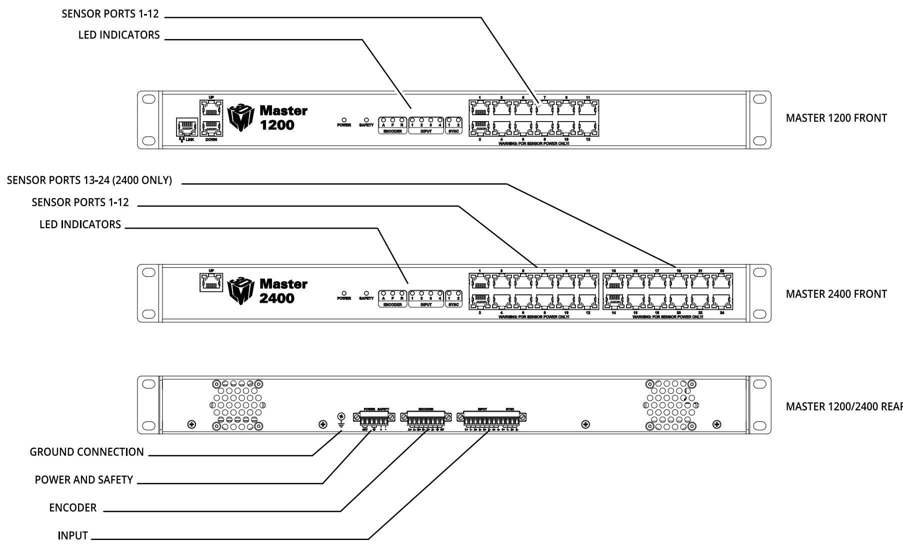






Master 1200/2400

The Master 1200/2400 provides sensor power and safety interlock, and broadcasts system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND(48VDC)	3
GND(48VDC)	4
Safety Control+	5
Safety Control–	6

The +48VDC power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

The Safety Control requires a voltage differential 12VDC to 48VDC across the pin to enable the laser.

Digital Input (16 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3

Function	Pin
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12

This connector does not need to be wired up for proper operation.

Encoder (8 pin connector)

Function	Pin
Encoder_A+	1
Encoder_A-	2
Encoder_B+	3
Encoder_B-	4
Encoder_Z+	5
Encoder_Z-	6
GND	7
+5VDC	8

Master 1200/2400 Electrical Specifications

Electrical Specifications for Master 1200/2400

	Master 1200 / 2400
Power Supply Voltage	+48VDC
Power Supply current (Max.)	10A
Power Draw (Min.)	15W
Safety Voltage	+12 to +48VDC
Encoder signal voltage range	RS485 Differential
Digital input voltage range	Logical LOW: 0 VDC to +0.1VDC Logical HIGH: +3.5 VDC to +6.5VDC

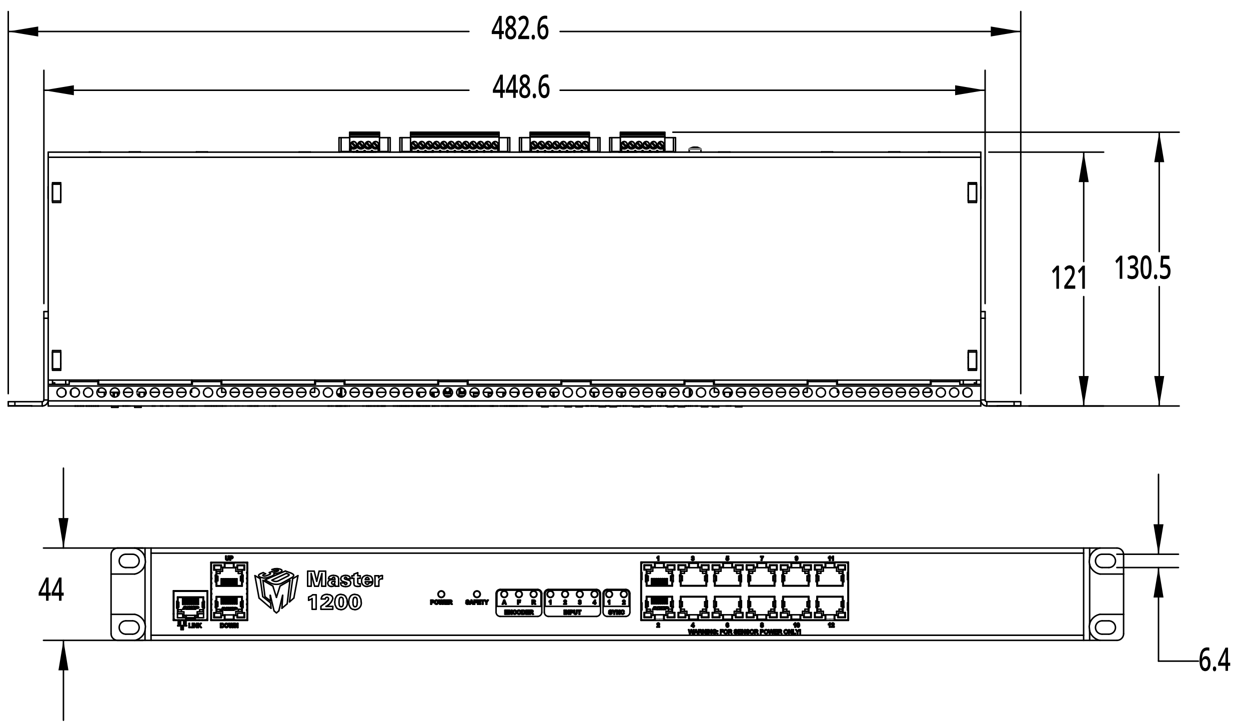
⚠ When using a Master 1200/2400, its chassis must be well grounded.

🗋	The +48VDC power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

🗋	The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements which need to be considered when calculating total system power requirements.

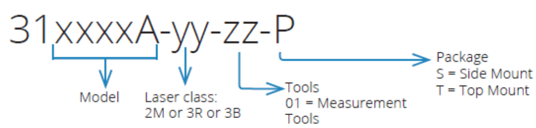
Master 1200/2400 Dimensions

The dimensions of Master 1200 and Master 2400 are the same.



Parts and Accessories

Gocator Part Number Legend



Gocator 1300 Sensors (Side Mount Package)

Description	Part Number
Gocator 1320 with Class 3R laser, Side Mount Package	311320A-3R-01-S
Gocator 1340 with Class 3B laser, Side Mount Package	311340A-3B-01-S
Gocator 1340 with Class 3R laser, Side Mount Package	311340A-3R-01-S
Gocator 1350 with Class 3B laser, Side Mount Package	311350B-3B-01-S
Gocator 1365 with Class 3B laser, Side Mount Package	311365B-3B-01-S
Gocator 1370 with Class 3B laser, Side Mount Package	311370B-3B-01-S
Gocator 1390 with Class 3B laser, Side Mount Package	311390B-3A-01-S

Gocator 1300 Sensors (Top Mount Package)

Description	Part Number
Gocator 1350 with Class 3B laser, Top Mount Package	311350B-3B-01-T

Masters

Description	Part Number
Master 100 - for single sensor (development only)	30705
Master 400 - for networking up to 4 sensors	30680
Master 800 - for networking up to 8 sensors	30681
Master 1200 - for networking up to 12 sensors	30649
Master 2400 - for networking up to 24 sensors	30650

Cordsets

Description	Part Number
2m I/O cordset, open wire end	30864-2m
5m I/O cordset, open wire end	30862
10m I/O cordset, open wire end	30863

Description	Part Number
15m I/O cordset, open wire end	30864-15m
20m I/O cordset, open wire end	30864-20m
25m I/O cordset, open wire end	30864-25m
2m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-2m
5m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30859
10m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30860
15m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-15m
20m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-20m
25m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-25m
2m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-2m
5m Power and Ethernet to Master cordset, 2x RJ45 ends	30856
10m Power and Ethernet to Master cordset, 2x RJ45 ends	30857
15m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-15m
20m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-20m
25m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-25m

Contact LMI for information on creating cordsets with custom length or connector orientation. The maximum cordset length is 60 m.

Return Policy

Return Policy
Before returning the product for repair (warranty or non-warranty) a Return Material Authorization (RMA) number must be obtained from LMI. Please call LMI to obtain this RMA number.

Carefully package the sensor in its original shipping materials (or equivalent) and ship the sensor prepaid to your designated LMI location. Please ensure that the RMA number is clearly written on the outside of the package. Inside the return shipment, include the address you wish the shipment returned to, the name, email and telephone number of a technical contact (should we need to discuss this repair), and details of the nature of the malfunction. For non-warranty repairs, a purchase order for the repair charges must accompany the returning sensor.

LMI Technologies Inc. is not responsible for damages to a sensor that are the result of improper packaging or damage during transit by the courier.

Software Licenses

Pico-C

Website:

http://code.google.com/p/picoc/

License:

picoc is published under the "New BSD License".

http://www.opensource.org/licenses/bsd-license.php

Copyright (c) 2009-2011, Zik Saleeba

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Zik Saleeba nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BlowFish

Website:

http://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html

License:

PuTTY is copyright 1997-2011 Simon Tatham.

Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad Khalifa, Markus Kuhn, Colin Watson, and CORE SDI S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL SIMON TATHAM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CodeMirror

Website:

http://codemirror.net

License:

Copyright (C) 2011 by Marijn Haverbeke <marijnh@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANT ABILITY,FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jQuery

Website:

http://jquery.com/

License:

Copyright (c) 2011 John Resig, http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Closure Library

Website:

http://code.google.com/closure/library/index.html

License:

Copyright 2006 The Closure Library Authors. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS-IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

jQuery.CopyEvents

Website:

http://brandonaaron.net

License:

Copyright (c) 2006 Brandon Aaron

Licensed under the MIT License (http://www.opensource.org/licenses/mit-license.php)

jQuery.history

License:

jQuery history plugin

Copyright (c) 2006 Taku Sano (Mikage Sawatari)

Licensed under the MIT License (http://www.opensource.org/licenses/mit-license.php)

Modified by Lincoln Cooper to add Safari support and only call the callback once during initialization for msie when no initial hash supplied. API rewrite by Lauris Bukis-Haberkorns

jQuery.mouseWheel

Website:

http://brandonaaron.net

License:

Copyright (c) 2010 Brandon Aaron

Licensed under the MIT License (http://www.opensource.org/licenses/mit-license.php)

jQuery.scaling

Website:

http://eric.garside.name

License:

Scaling 1.0 - Scale any page element

Copyright (c) 2009 Eric Garside

Licensed under the MIT License (http://www.opensource.org/licenses/mit-license.php)

jQuery.scrollFollow

Website:

http://kitchen.net-perspective.com/

License:

Copyright (c) 2008 Net Perspective

Licensed under the MIT License (http://www.opensource.org/licenses/mit-license.php)

Flex SDK

Website:

http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK

License:

Copyright (c) 2010 Adobe Systems Incorporated

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

EtherNet/IP Communication Stack

Website:

sourceforge.net/projects/opener

License:

SOFTWARE DISTRIBUTION LICENSE FOR THE

ETHERNET/IP(TM) COMMUNICATION STACK (ADAPTED BSD STYLE LICENSE)

Copyright (c) 2009, Rockwell Automation, Inc. ALL RIGHTS RESERVED.

EtherNet/IP is a trademark of ODVA, Inc.

Support

For assistance regarding a component or product, please contact LMI Technologies.

World

Email	support@lmi3D.com
Web	http://www.lmi3D.com

North America

Phone	+1 604 636 1011
Fax	+1 604 516 8368

Europe

Phone	+31 45 850 7000
Fax	+31 45 574 2500

For more information on safety and laser classifications, please contact:

U.S. Food and Drug Administration
Center for Devices and Radiological Health
WO66-G609
10903 New Hampshire Avenue
Silver Spring, MD 20993-0002
USA

Contact

North America	Europe
LMI Technologies Inc.	LMI Technologies BV
1673 Cliveden Avenue	Valkenburgerweg 223
Delta BC V3M 6V5	NL-6419AT Heerlen
Canada	The Netherlands
Phone: +1 604 636 1011	Phone: +31 45 850 7000
Fax: +1 604 516 8368	Fax: +31 45 574 2500