USER MANUAL

Gocator 3100 Series

Firmware version: 4.2.x.xx

Document revision: C

Copyright

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 3100 series of snapshot LED projection 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.

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 3100 I/O Connector* on page 305 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).

The Master 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.

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

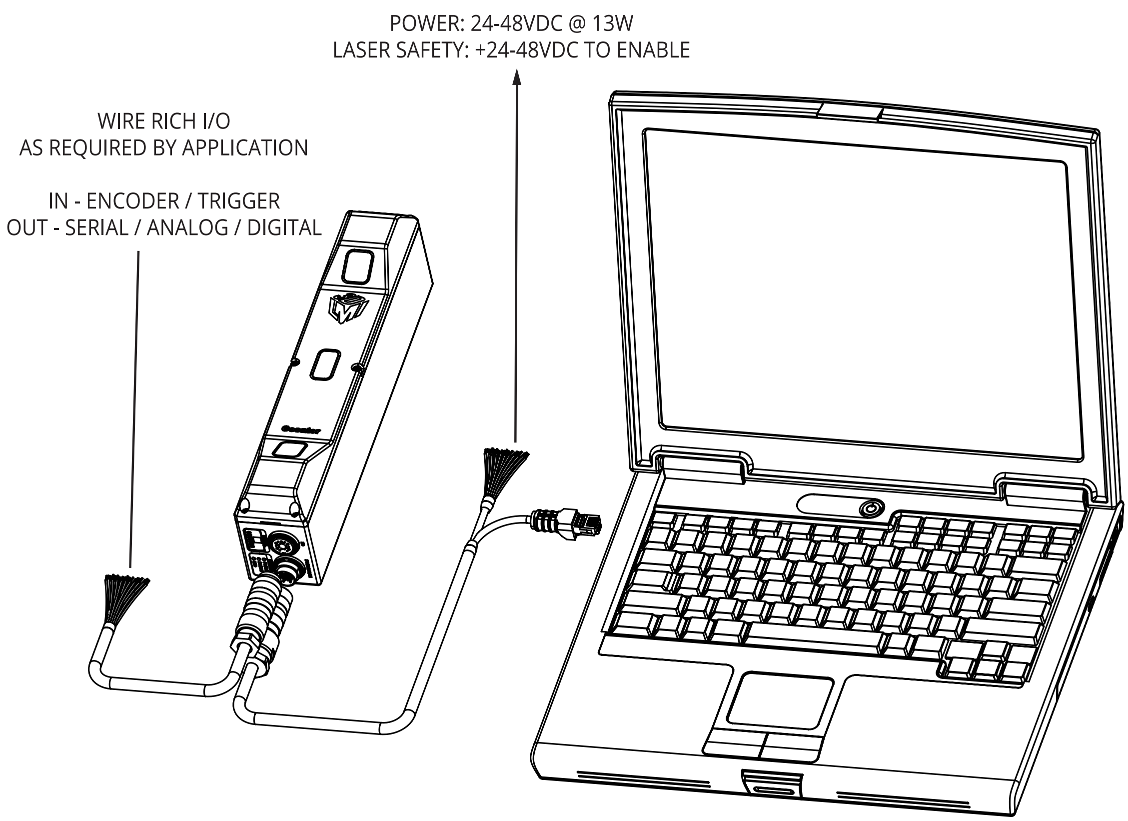
The following sections provide system and hardware overviews, in addition to installation and setup procedures.

System Overview

Gocator sensors can be installed and used in scenarios where the target to be scanned is static in position relative to the sensor for the short duration of camera exposure. Sensors can be connected as standalone devices or in a multi-sensor system.

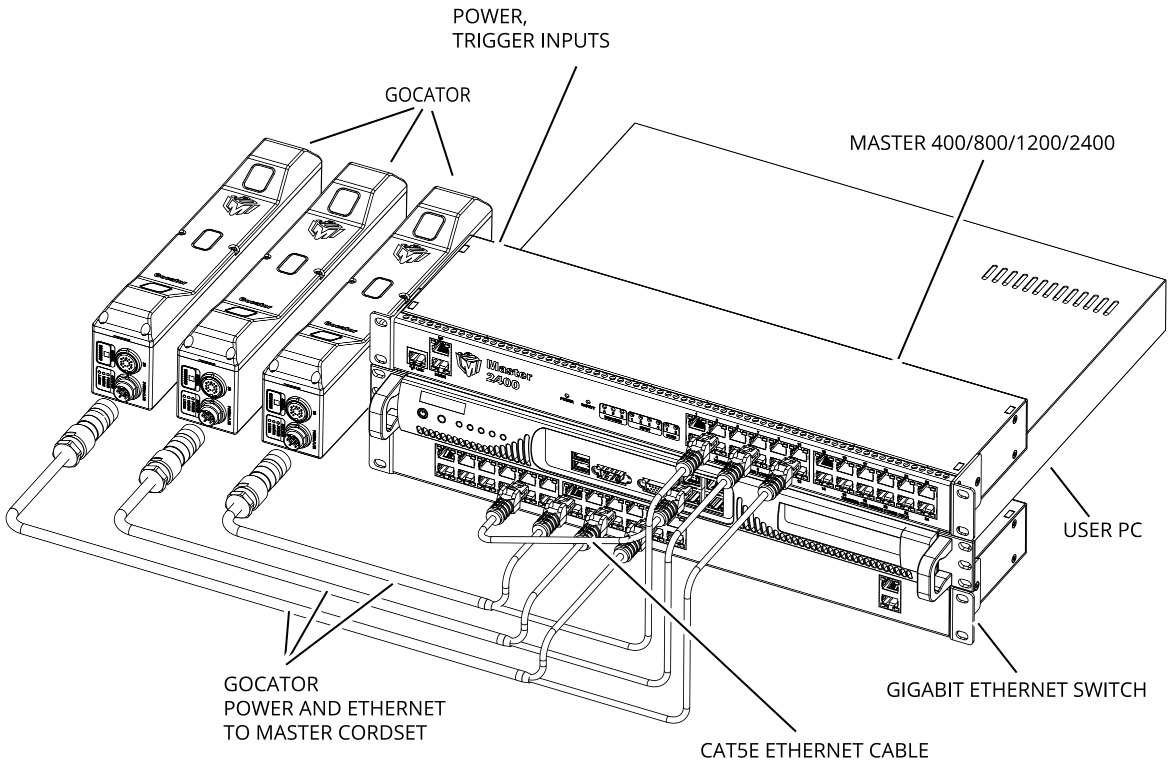
Standalone System

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.



Multi-Sensor System

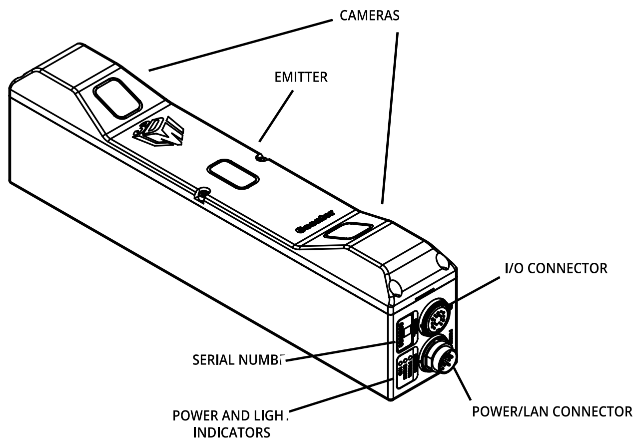
Master 400/800/1200/2400 networking hardware can be used to connect two or more sensors into a multi-sensor system. Gocator Master cordsets are used to connect the sensors to a Master. The Master provides a single point of connection for power, safety, encoder, and digital inputs. A Master 400/800/1200/2400 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.

Gocator 3110 Sensor



Item	Description
Camera	Observes light reflected from target surfaces.
Light Emitter	Emits structured light for 3D data acquisition.
I/O Connector	Accepts input/output signals.
Power/LAN Connector	Connects to 1000 Mbit/s Ethernet network.
Power Indicator	Illuminates when power is applied (blue).
Safety Indicator	Illuminates when laser safety input is active (amber).
Serial Number	Unique sensor serial number.

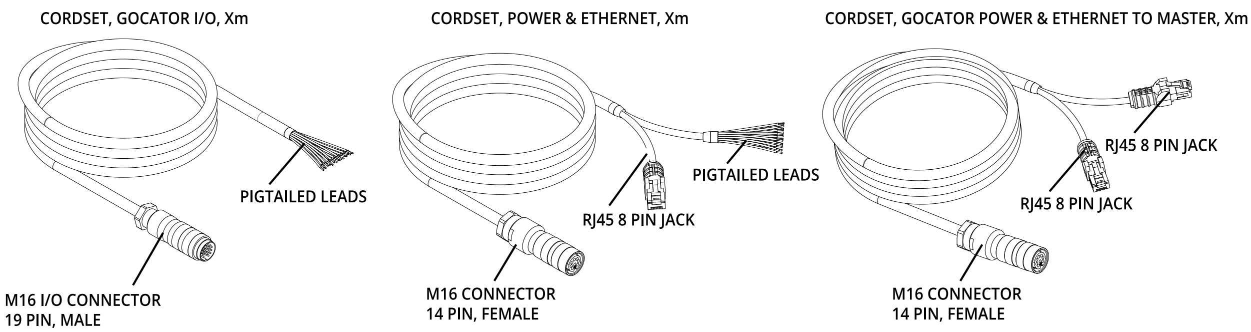
Gocator 3100 Cordsets

Gocator 3100 sensors use two types of cordsets.

The Power & Ethernet cordset provides power and safety interlock to the sensor. It is also used for sensor communication via 1000 Mbit/s Ethernet with a standard RJ45 connector. The Master version of the Power & Ethernet cordset provides direct connection between the sensor and a Master 400/800/1200/2400.

The Gocator 3100 I/O cordset provides digital I/O connections, an encoder interface, RS-485 serial connection, and an analog output.

> If you use a Master 100 with a Gocator 3100 series sensor, you must use the latest version of the Master 100 with a metal-shielded Master Power Port and a Power & Ethernet cordset with a metal-shielded Power/Sync RJ45 plug.

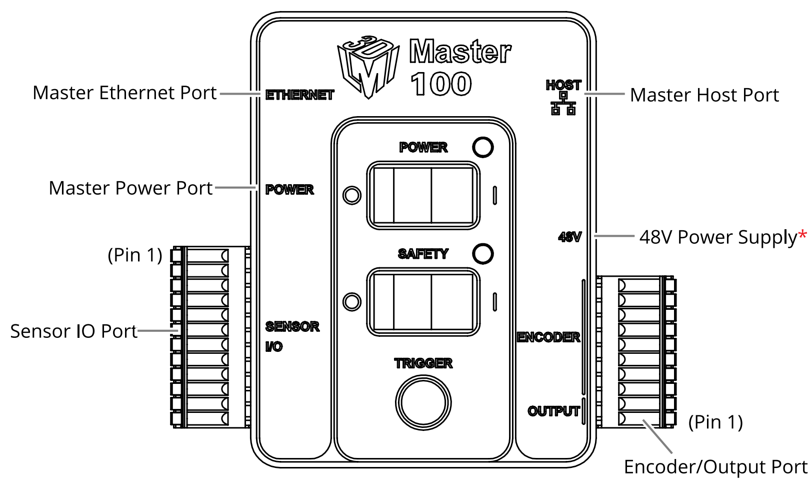


The maximum cordset length is 60 m. See *Gocator 3100 I/O Connector* on page 305 and See *Gocator Power/LAN Connector* on page 303 for pinout details.

See *Parts and Accessories* on page 317 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 3100 series for standalone system setup.

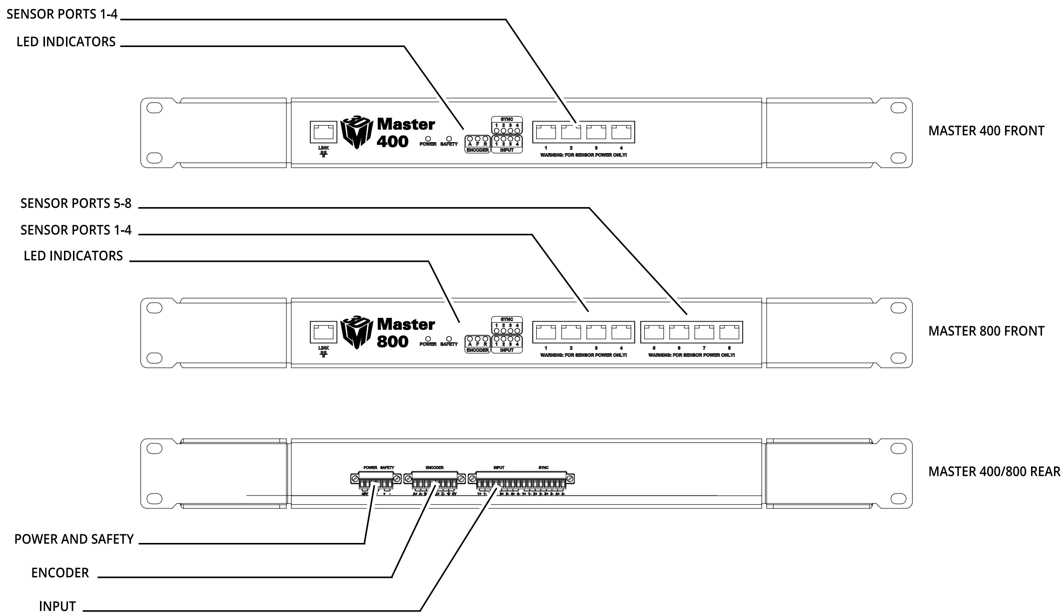


Item	Description
Master Ethernet Port	Connects to the RJ45 connector labeled Ethernet on the Power/LAN to Master cordset.
Master Power Port	Connects to the RJ45 connector labeled Power/Sync on the Power/LAN to Master cordset. Provides power and laser safety to the Gocator.
Sensor I/O Port	Connects to 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* on page 309 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.

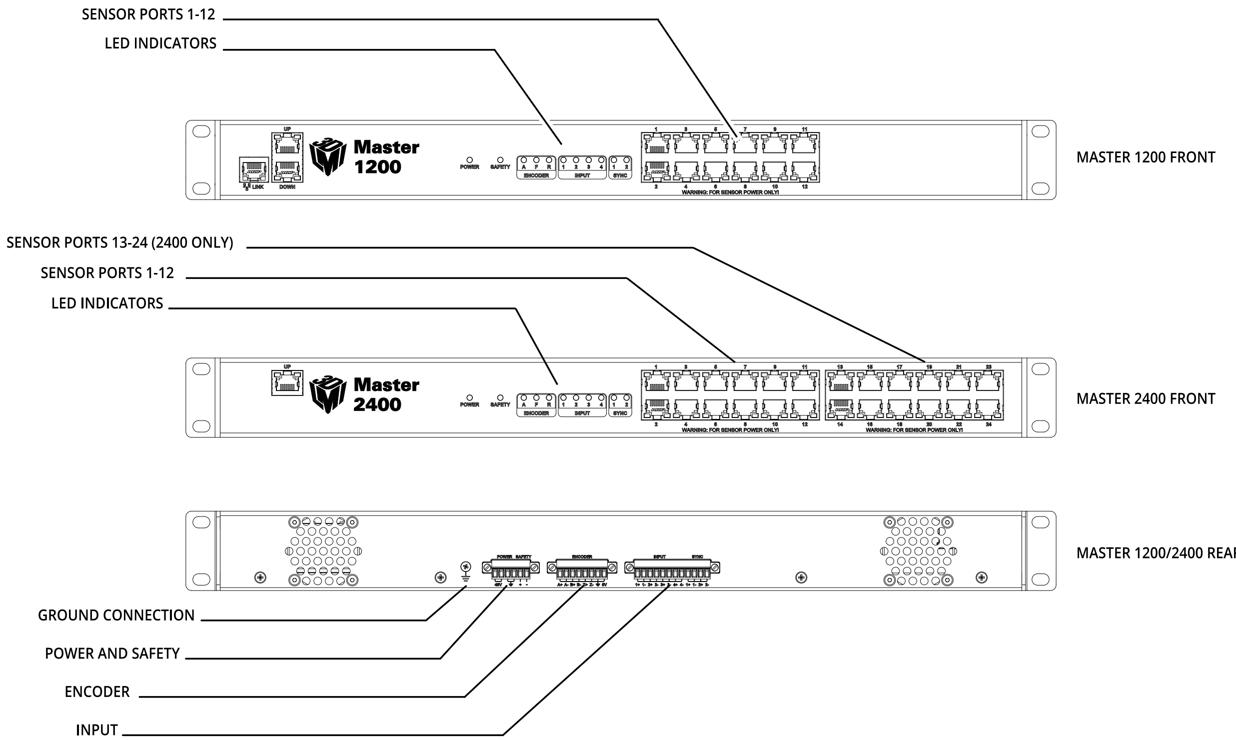


Item	Description
Sensor Ports	Master connection for Gocator sensors (no specific order required).
Ground Connection	Earth ground connection point.
Laser Safety	Laser safety connection.
Encoder	Accepts encoder signal.
Input	Accepts digital input.

See *Master 400/800* on page 311 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* on page 314 for pinout details.

Installation

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

Grounding - Gocator

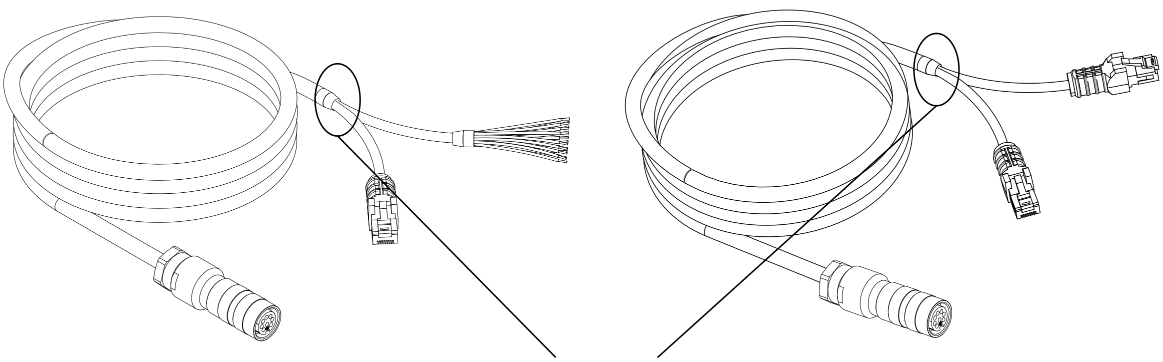
Gocators should be grounded to the earth/chassis through their housings and through the grounding shield of the Power I/O cordset. Gocator sensors have been designed to provide adequate grounding through the use of M5 x 0.8 pitch mounting screws. Always check grounding with a multi-meter to ensure electrical 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.

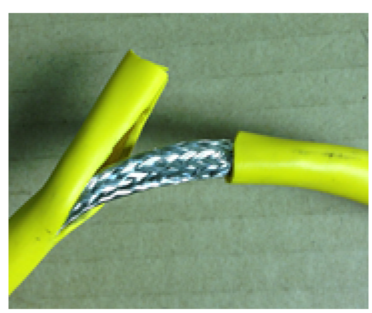
CORDSET, POWER & ETHERNET, Xm **CORDSET, GOCATOR POWER & ETHERNET TO MASTER, Xm**



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.

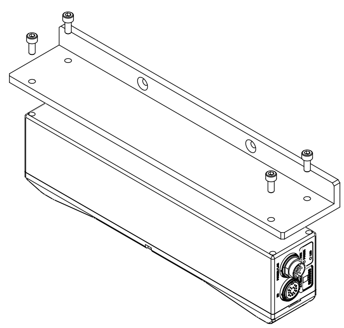
Grounding - Master 400/800/1200/2400

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.

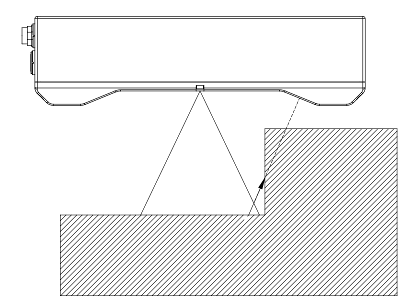
	The frame or electrical cabinet that the Master is mounted to must be connected to earth ground.

Mounting

Sensors should be mounted using four or six (depending on the model) 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.



With the exception of Gocator 2880, sensors should not be installed near objects that might occlude a camera's view of the LED light pattern. (Gocator 2880 is specifically designed to compensate for occlusions.)



⚠	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.

Network Setup

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

Client Setup

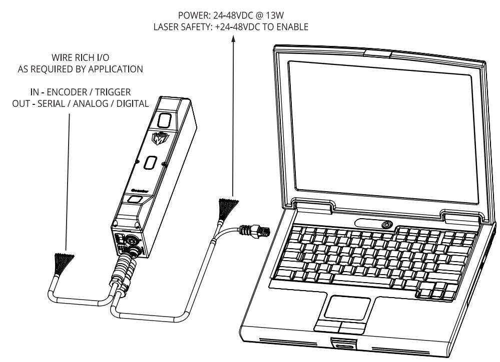
Sensors are shipped with the following default network configuration:

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

To connect to a sensor for the first time:

1. Connect cables and apply power.

   Sensor cabling is illustrated in *System Overview* on page 12.



2. Change the client PC's network settings.

   Windows 7

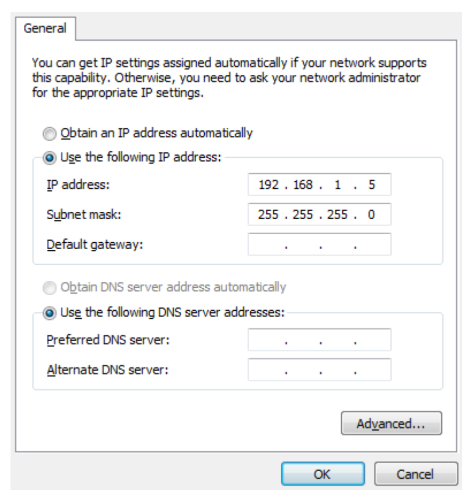
   a. Open the Control Panel, select **Network and Sharing Center**, and then click **Change Adapter Settings**.

   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**.

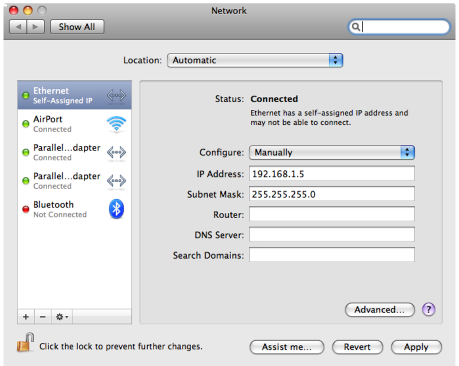


Mac OS X v10.6

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* on page 294 if you experience any problems while attempting to establish a connection to the sensor.

Gocator Setup

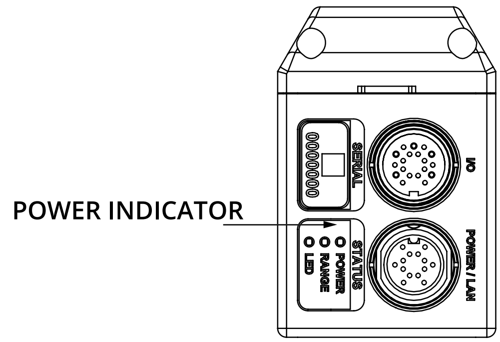
The Gocator is shipped with a default configuration that will produce 3D data on most targets.

The following sections walk you through the steps required to set up a standalone sensor system and a multi-sensor system for operations. After you have completed the setup, 3D data can be acquired to verify basic sensor operation.

Running a Standalone Sensor System

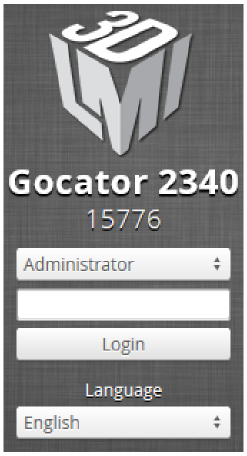
To configure a standalone sensor system:

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

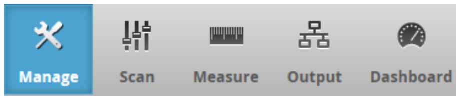


2. Enter the sensor's IP address (192.168.1.10) in a web browser.

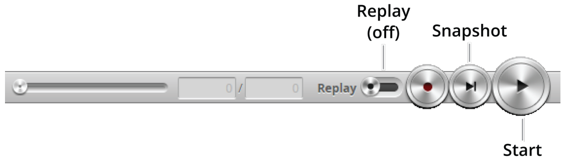
3. Log in as Administrator with no password.
   The interface display language can be changed using the language option. After selecting the language, the browser will refresh and the web interface will display in the selected language.



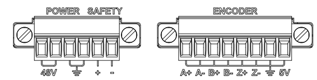
4. Go to the **Manage** page.



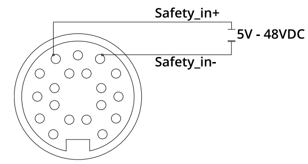
5. Ensure that **Replay** mode is off (the slider is set to the left).



6. Ensure that the LED Safety Switch is enabled or the LED 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.
   The **Snapshot** button is used to trigger the capture of a single 3D point cloud.



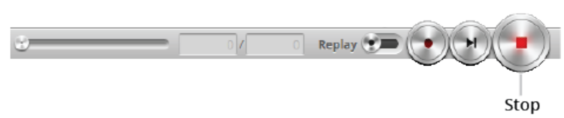
Master 400/800/1200/2400



Standalone

9. Move a target into the LED light pattern.
   If a target object is within the sensor's measurement range, the data viewer will display the shape of the target, and the sensor's range indicator will illuminate.
   If you cannot see the laser, or if a 3D heightmap is not displayed in the Data Viewer, see *Troubleshooting* on page 294.

10. Press the **Stop** button.
    The LED light projection should turn off.

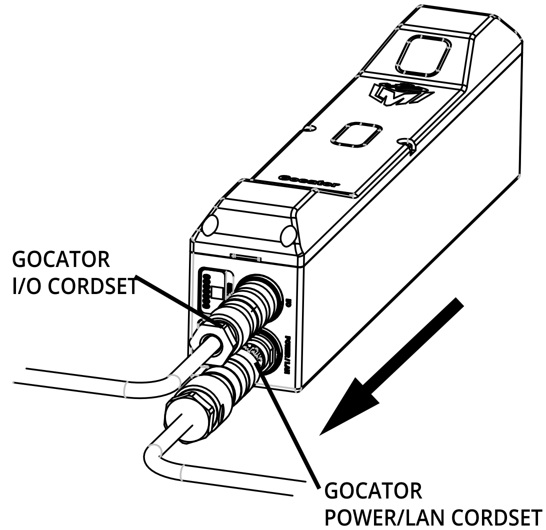


Running a Multi-Sensor System

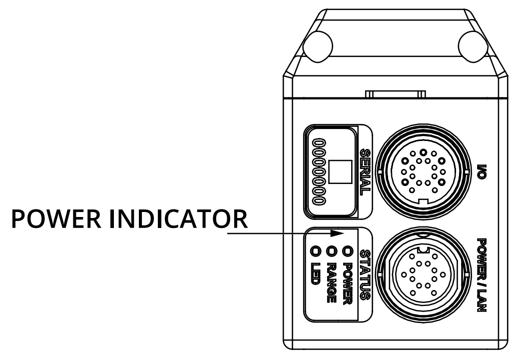
All sensors are shipped with a default IP address of 192.168.1.10. Ethernet networks require a unique IP address for each device, so you must set up a unique address for each sensor. For each additional sensor, follow the steps below.

To configure a multi-sensor system:

1. Turn off the sensor and unplug the Ethernet network connection of any configured sensors.

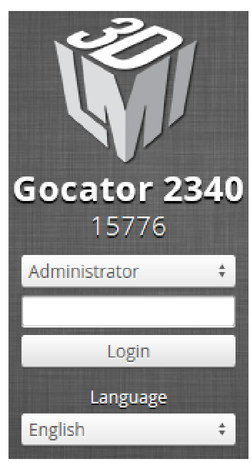


2. Power up the new sensor.
   The power LED (blue) of the new sensor should turn on immediately.

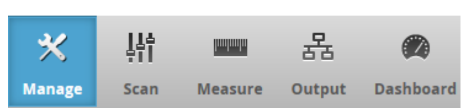


3. Enter the new sensor's default IP address (192.168.1.10) in a web browser.

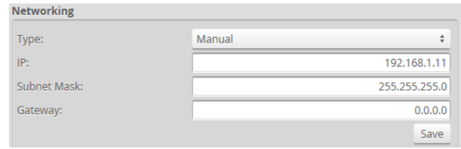
4. Log in as Administrator with no password.
   The interface display language can be changed using the language option. After selecting the language, the browser will refresh and the web interface will display in the selected language.



5. Go to the **Manage** page.



6. Modify the IP address in the **Networking** category and click the **Save** button.

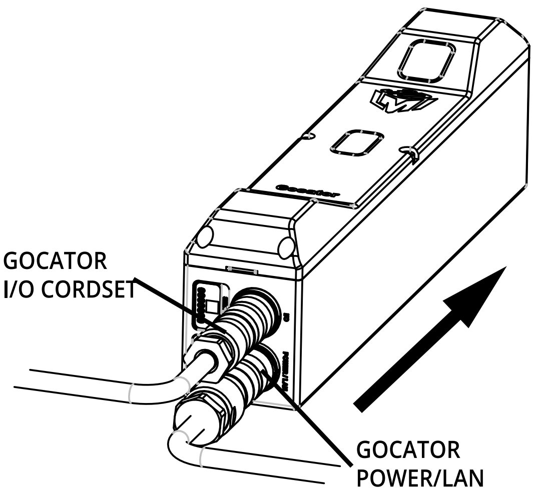
   

   You should increment the last octet of the IP address for each additional sensor you need to use. For example, if the IP address of the first sensor you configured is 192.168.1.10, use 192.168.1.11 for the second sensor; use 192.168.1.12 for the third; etc.

   When you click the **Save** button, you will be prompted to confirm your selection.

7. Power-cycle or reset the sensor.

   After changing a sensor's network configuration, the sensor must be reset or power-cycled before the change will take effect.

8. Repeat these steps for each additional sensor.

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 43)
Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance.

Scan Setup and Alignment (page 53)
Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment.

Measurement (page 97)
Contains built-in measurement tools and their settings.

Output (page 144)
Contains settings for configuring output protocols used to communicate measurements to external devices.

Dashboard (page 155)
Provides monitoring of measurement statistics and sensor health.

Toolbar (page 36)
Controls sensor operation, manages jobs, and replays recorded measurement data.

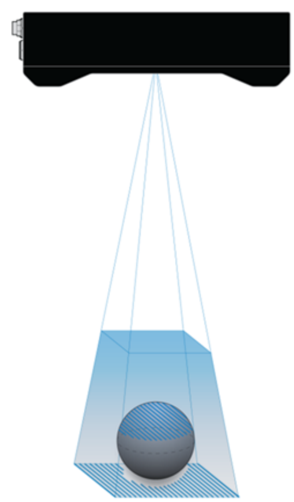
Theory of Operation

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

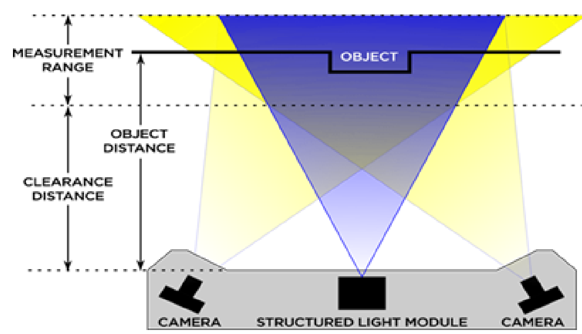
3D Acquisition

Principle of 3D Acquisition

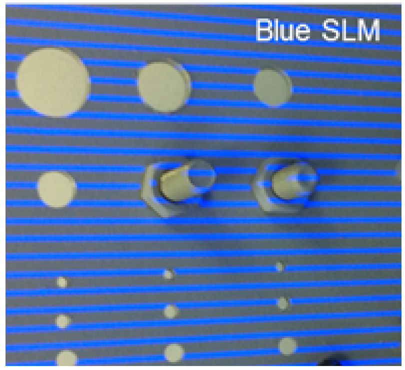
The Gocator 3100 is called a 3D snapshot sensor, meaning that it captures an entire surface in 3D in a single snapshot. The sensor projects several structured light patterns in a rapid sequence onto the target. The reflection of the pattern off the target is captured by two cameras. The target must remain stationary during the camera exposure of the light patterns. The required exposure time depends on the shape, color, and reflectiveness of the target, but is often shorter than 1 second.



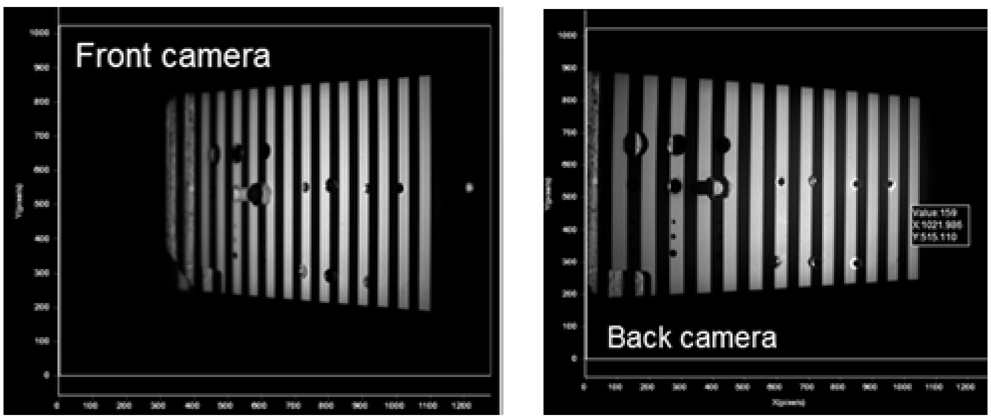
A Structured Light Modulator (SLM) produces a sequence of high resolution/high contrast light patterns using a blue LED. Two cameras capture the reflected light pattern from different viewing angles. The sensor can then use either stereo correlation or independent triangulation to generate 3D points from the light pattern. Gocator sensors are always pre-calibrated to deliver 3D data in engineering units throughout the specified measurement range.



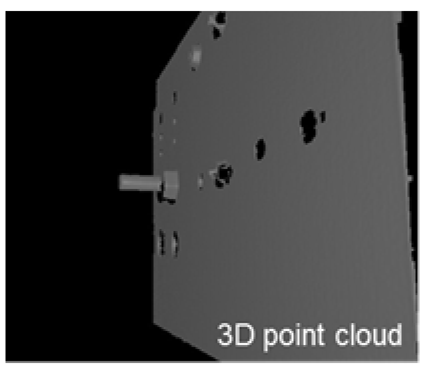
These are the steps of the acquisition of a 3D point cloud:



Step 1: Light pattern projected on target



Step 2: Reflected light captured by two cameras



Step 3: Use stereo correlation or independent triangulation to generate 3D point cloud

Stereo Correlation vs. Independent Triangulation

Stereo correlation means that the sensor locates the same point on the physical target in the two images captured at different viewing angles. Since the exact distance between the two cameras and the viewing angles are known, the distance to the point can be calculated. In order for stereo correlation to work and produce a 3D data point, the point on the target must be visible in both cameras. Stereo acquisition may produce more stable measurements on targets with a simple shape, but will be affected by occlusions on targets with complicated shapes and protruding features.

Independent triangulation means that each camera independently triangulates off the LED light pattern, based on the calibration process that takes place when the sensor is manufactured. Since the Gocator 3100 sensor has two cameras, a point on the physical target only needs to be visible to one of the cameras in order to generate a 3D point. Independent triangulation may improve performance on targets with complicated shapes that can cause occlusion, but it relies on the sensor's internal components being fully stable.

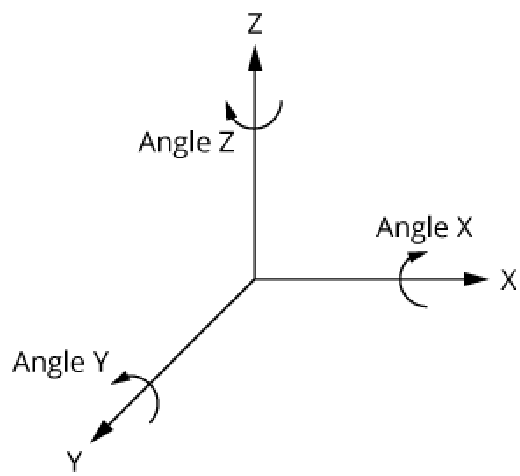
The **Reduce Occlusion** setting determines whether 3D data is acquired by using stereo correlation or both stereo correlation and independent triangulation. See *Reduce Occlusion* on page 57 for more details.

3D Data Output

Gocator measures the shape of the object calculated from either dual triangulation or stereo correlation. The Gocator reports a series of 3D coordinates from the surface of the target in the sensor's field of view.

Coordinate Systems

Gocator 3100 sensors use Cartesian left-hand notation for defining 3D coordinates.



The Z axis represents the sensor's measurement range (MR), where the values increase toward the sensor. The X axis and Y axis represent the sensor's field of view (FOV).

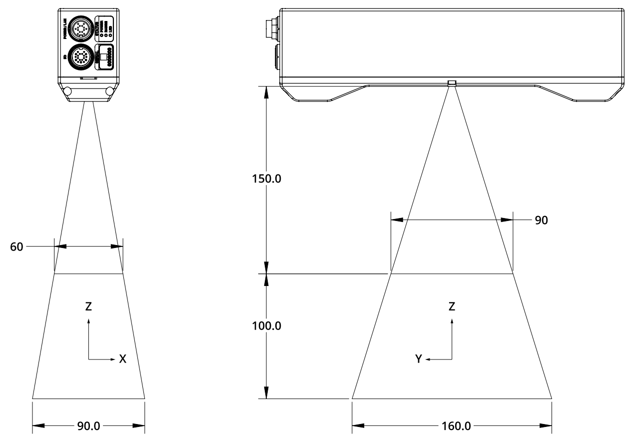
X offset, Y offset, and Z offset define the translations from the origin.

Rotations are specified based on rotating the target around the X axis (Angle X), followed by rotating around the Y axis (Angle Y), followed by rotating around the Z axis.

3D point cloud data is reported in sensor coordinates or system coordinates depending on the alignment state of the sensor. These coordinate systems are described below.

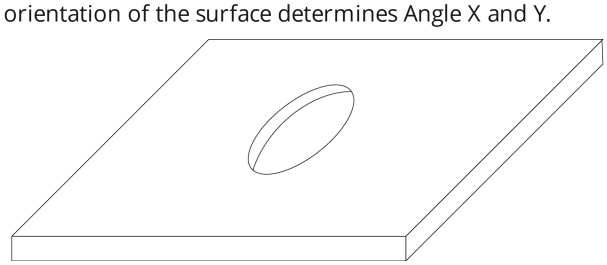
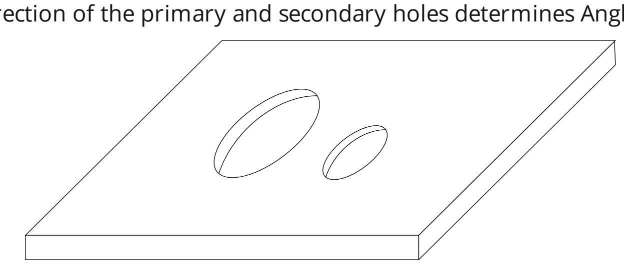
Sensor Coordinates

Before alignment, the origin of the sensor is at the center of the sensor's measurement range (MR) and field of view (FOV).



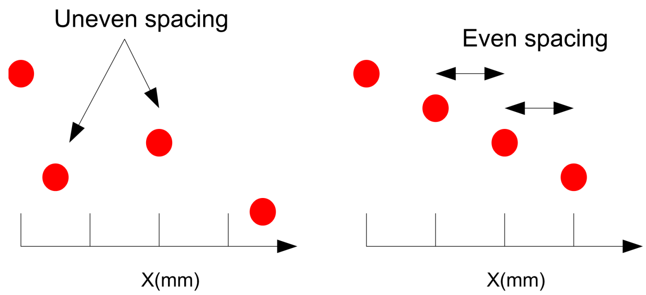
System Coordinates

Alignment can be used to establish a transformed coordinate system according to the user's needs. Alignment determines the adjustments to X, Y, and Z, as well as rotation angle around each axis. Transformed coordinate systems can be associated with specific sensor jobs. For details, see on page 65.

Target	Alignment	Explanation
Flat surface	Z offset Angle X Angle Y	System coordinates are aligned such that the system Z axis is perpendicular to the alignment plate surface. The system Z origin is set to the base of the alignment plate (using the Z Offset parameter).
Plate (single reference hole)	X offset Y offset Z offset Angle X Angle Y	The center of the hole defines the location of the X, Y and Z origins. The orientation of the surface determines Angle X and Y. 
Plate (with two reference holes)	X offset Y offset Z offset Angle X Angle Y Angle Z	The sizes of the primary and the secondary holes must be different. The center of the primary hole defines the location of the X, Y, and Z origins. The orientation of the surface determines Angle X and Y. The direction of the primary and secondary holes determines Angle Z. 

Resampling

The internal acquisition engine in the Gocator produces a random 3D point cloud where each individual point is an (X,Y,Z) coordinate triplet. In Surface mode, the random 3D point cloud is resampled to an even grid in the X-Y plane. The resampling divides the X-Y plane into fixed size square "bins." 3D points are projected along the Z axis, perpendicular to the X-Y plane, and points that fall into the same bin will be combined into a single Z value. The size of the resampling bins can be set with the **X Spacing Interval** setting; for details, see on page 64. The X-Y resampling plane is established through the Gocator's built-in alignment routine. That is, the resampling plane is set to match the plane described by the calibration plate; see *System Coordinates* on the previous page.



In the Ethernet data channel, only the Z values are reported and the X and Y positions can be reconstructed through the 2D-array index at the receiving end (the client). Resampling reduces the complexity for the algorithms in the Gocator's built-in measurement tools, allowing them to run on the embedded processors. All built-in measurement tools in the Gocator operate on resampled data in Surface mode.

Acquisition Speed

The following table provides estimates of the acquisition speed under different field-of-view, resolution, intensity, and occlusion reduction settings.

The exposure was set to 2 milliseconds with no measurement tools instantiated.

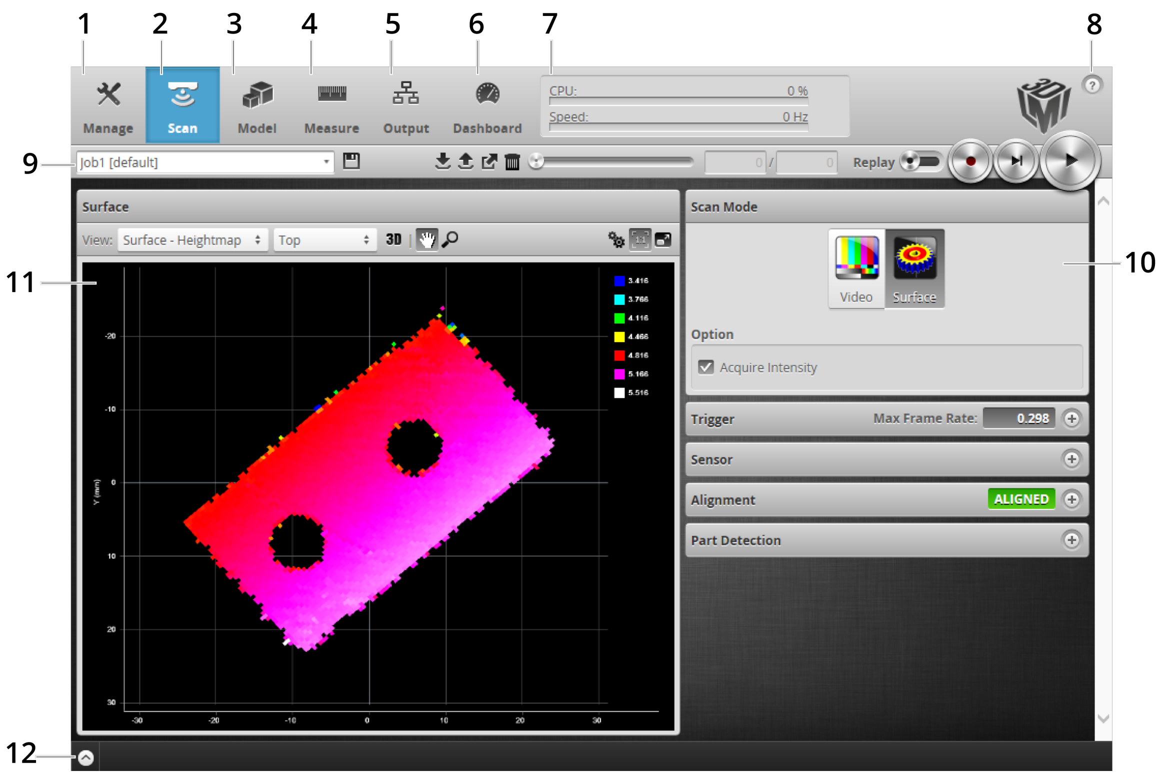
Field of View (X x Y x MR)	Resolution (mm)	Speed (Hz) Acquire Intensity and Reduce Occlusion options disabled
90x160x100	0.5	1.2
90x160x100	0.2	0.8
90x160x100	0.1	0.2
35x35x100	0.5	7.5
35x35x100	0.2	2.7
35x35x100	0.1	1.9
35x35x35	0.5	9
35x35x35	0.2	6.8
35x35x35	0.1	4.0

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.

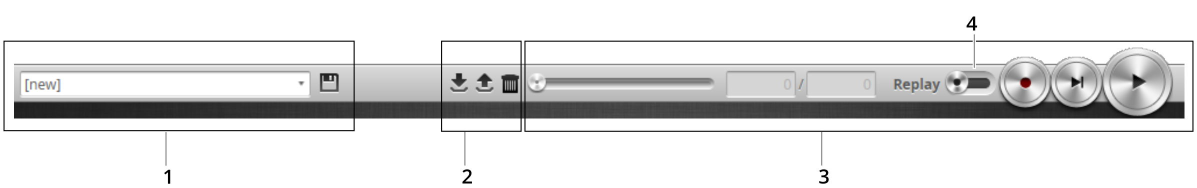


	Element	Description
1	Manage page	Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance. See *System Management and Maintenance* on page 43.
2	Scan page	Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment. See *Scan Setup and Alignment* on page 53.
3	Model page	Lets you set up part matching. See *Models and Part Matching* on page 82

	Element	Description
4	Measure page	Contains built-in measurement tools and their settings. See *Measurement* on page 97.
5	Output page	Contains settings for configuring output protocols used to communicate measurements to external devices. See *Output* on page 144.
6	Dashboard page	Provides monitoring of measurement statistics and sensor health. See *Dashboard* on page 155.
7	CPU Load and Speed	Provides important sensor performance metrics. See *Metrics Area* on page 42.
8	Help	Provides links to the user manual and SDK. Also lets you download support files from the sensor, which can be used with the emulator.See *Gocator Emulator* on page 158 for more information on the emulator.
9	Toolbar	Controls sensor operation, manages jobs, and replays recorded measurement data. See *Toolbar* below.
10	Configuration area	Provides controls to configure scan and measurement tool settings.
11	Data viewer	Displays sensor data, tool setup controls, and measurements. See *Data Viewer* on page 74 for its use when the **Scan** page is active and on page 98 for its use when the **Measure** page is active.
12	Log	Displays messages from the sensor (errors, warnings, and other information). See *Log* on page 41.

Toolbar

The toolbar is used for performing operations such as managing jobs, working with replay data, and starting and stopping the sensor.



	Element	Description
1	Job controls	For saving and loading jobs.
2	Replay 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 Jobs (Settings)

A Gocator can store several hundred jobs. Being able to switch between jobs is useful when a Gocator is used with different constraints during separate production runs. For example, width decision minimum and maximum values might allow greater variation during one production run of a part, but might allow less variation during another production run, depending on the desired grade of the part.

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.

When you change sensor settings using the Gocator web interfacein the emulator, 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
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.
Network Address	Network address changes are saved when you click the **Save** button in **Networking** on the **Manage** page. The sensor must be reset before changes take effect.

The job drop-down list in the toolbar shows the 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 create a job:

1. Choose **[New]** in the job drop-down list and type a name for the job.
2. Click the **Save** button or press **Enter** to save the job.
   The job is 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 save a job:

- Click the **Save** button .

The job is saved to sensor storage. Saving a job automatically sets it as the default, that is, the job loaded when then sensor is restarted.

To load (switch) jobs:

- Select an existing file name in the job drop-down list.

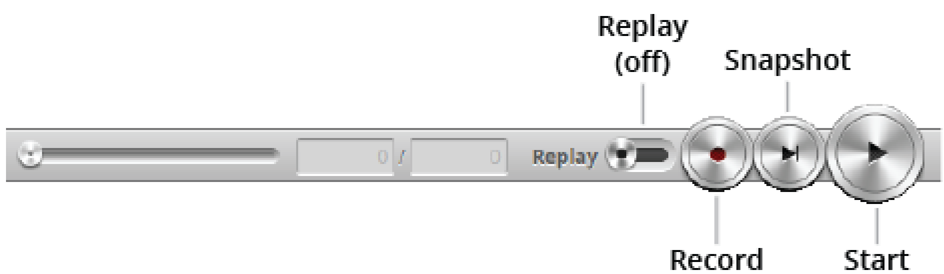
  The job is activated. If there are any unsaved changes in the current job, you will be asked whether you want to discard those changes.

You can perform other job management tasks—such as downloading job files from a sensor to a computer, uploading job files to a sensor from a computer, and so on—in the **Jobs** panel in the **Manage** page. See *Jobs* on page 46 for more information.

Recording, Playback, and Measurement Simulation

Gocator sensors can record and replay recorded scan data, and 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.

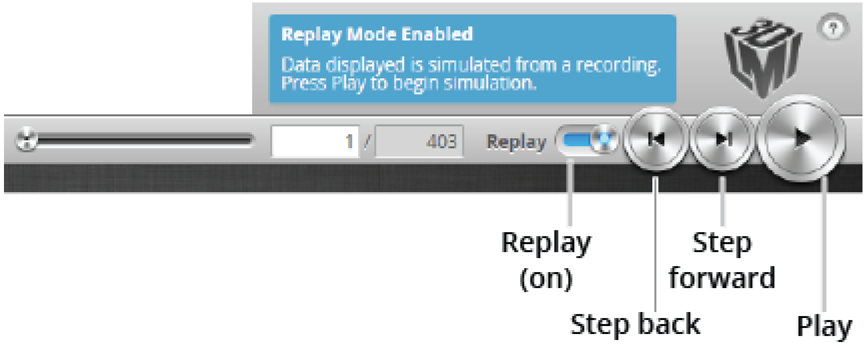


Recording and playback controls when replay is off

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 recording is enabled (and replay is off), the sensor will store the most recent data as it runs. Remember to disable recording if you no longer want to record live data. (Press the **Record** button again to disable recording).
3. Press the **Snapshot** button or **Start** button.
   The **Snapshot** button 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.



Playback controls when replay is on

To replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.
   To change the mode, **Replay Protection** must be unchecked.

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 until the end of the replay data.
   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 replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.
   To change the mode, **Replay Protection** must be unchecked.

2. Go to the **Measure** page.
   Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired. For information on adding and configuring measurements, see *Measurement* on page 97.

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; for more information on the dashboard, see *Dashboard* on page 155.

To clear replay data:

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

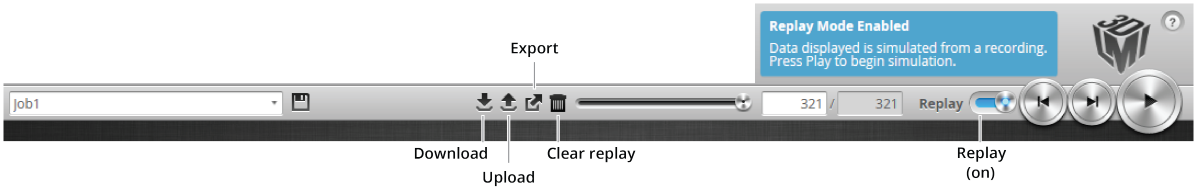
2. Click the **Clear Replay Data** button .

Downloading, Uploading, and Exporting Replay Data

Replay data (recorded scan data) can be downloaded from a Gocator to a client computer, or uploaded from a client computer to a Gocator.

Data can also be exported from a Gocator to a client computer in order to process the data using third-party tools.

> You can only upload replay data to the same sensor model that was used to create the data.



> Replay data is not loaded or saved when you load or save jobs.

To download replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.
   To change the mode, **Replay Protection** must be unchecked.
2. Click the **Download** button.

To upload replay data:

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

Replay data can be exported using the CSV format.

To export replay data in the CSV format:

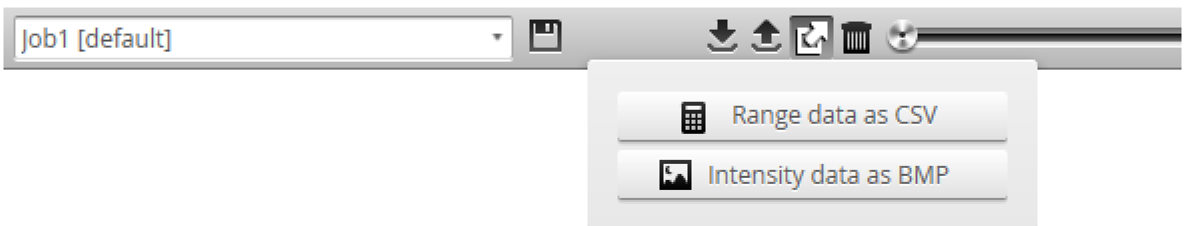
1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.

To change the mode, **Replay Protection** must be unchecked.

2. Click the **Export** button and select **Export Range Data as CSV**.
   Only data at the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Recording, Playback, and Measurement Simulation* on page 38.

3. Optionally, convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 292.

Recorded intensity data can be exported to a bitmap (.BMP format). **Acquire Intensity** must be checked in the **Scan Mode** panel while data was being recorded in order to export intensity data.



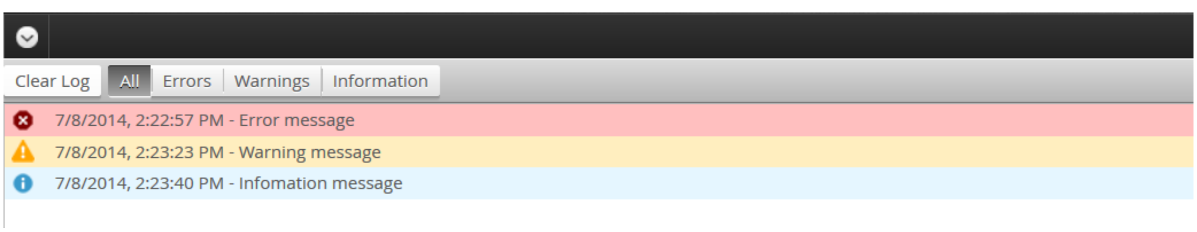
To export recorded intensity data to the BMP format:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue and a Replay Mode Enabled message is displayed.
   To change the mode, **Replay Protection** must be unchecked.

2. Click the **Export** button and select **Intensity data as BMP**.
   Only the intensity data in the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Recording, Playback, and Measurement Simulation* on page 38.

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:

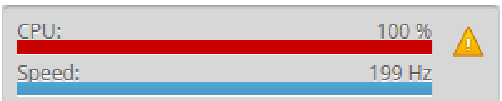
1. Click on the Log open button at the bottom of the web interface.

2. Click on the appropriate tab for the information you need.

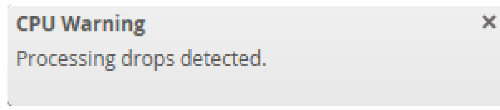
Metrics Area

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.



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* on the previous page 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 54), the data viewer can display video images or 3D surfaces. For details, see *Data Viewer* on page 74.

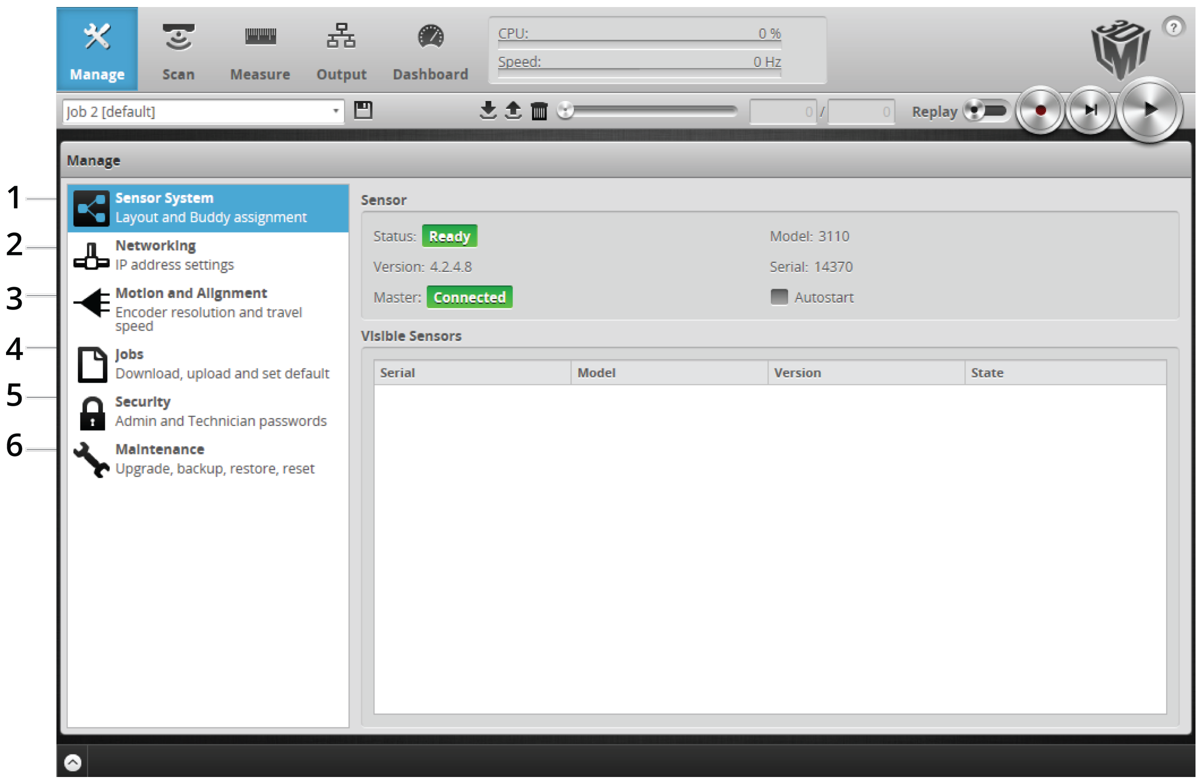
When the **Measure** page is active, the data viewer displays sensor data onto which representations of measurement tools and their measurements are superimposed. For details, see *Data Viewer* on page 98.

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.

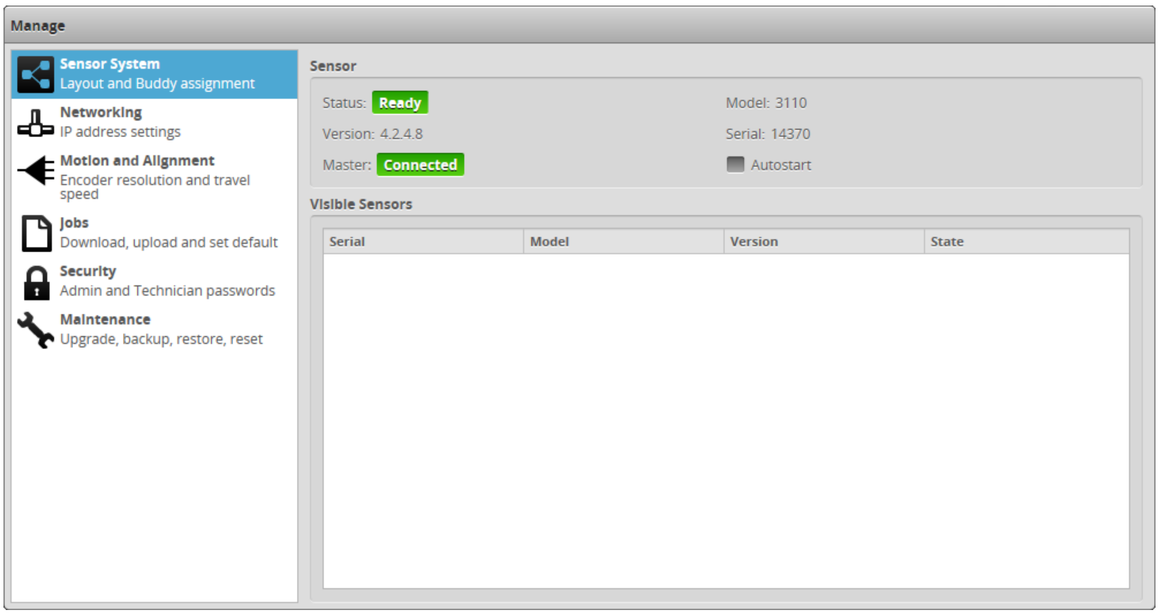


	Element	Description
1	Sensor System	Contains settings for configuring sensor system and boot-up. See *Sensor System* on the next page.
2	Networking	Contains settings for configuring the network. See *Networking* on page 45.
3	Motion and Alignment	Contains settings to configure the encoder. See *Motion and Alignment* on page 46.
4	Jobs	Lets you manage jobs stored on the sensor. See *Jobs* on

	Element	Description
		page 46.
5	Security	Lets you change passwords. See *Security* on page 48.
6	Maintenance	Lets you upgrade firmware, create/restore backups, and reset sensors. See *Maintenance* on page 49.

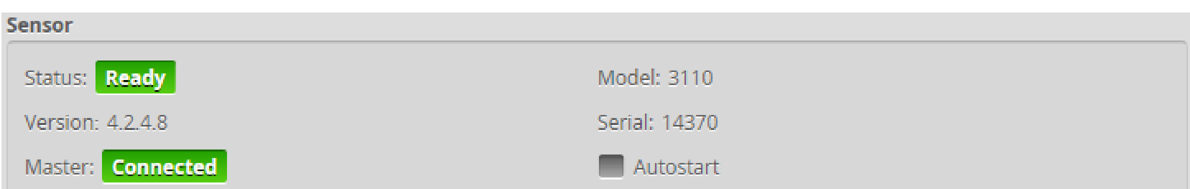
Sensor System

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.



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.

Over Temperature Protection

Sensors equipped with a 3B-N laser by default will turn off the laser if the temperature exceeds the safe operating range. You can override the setting by disabling the overheat protection.

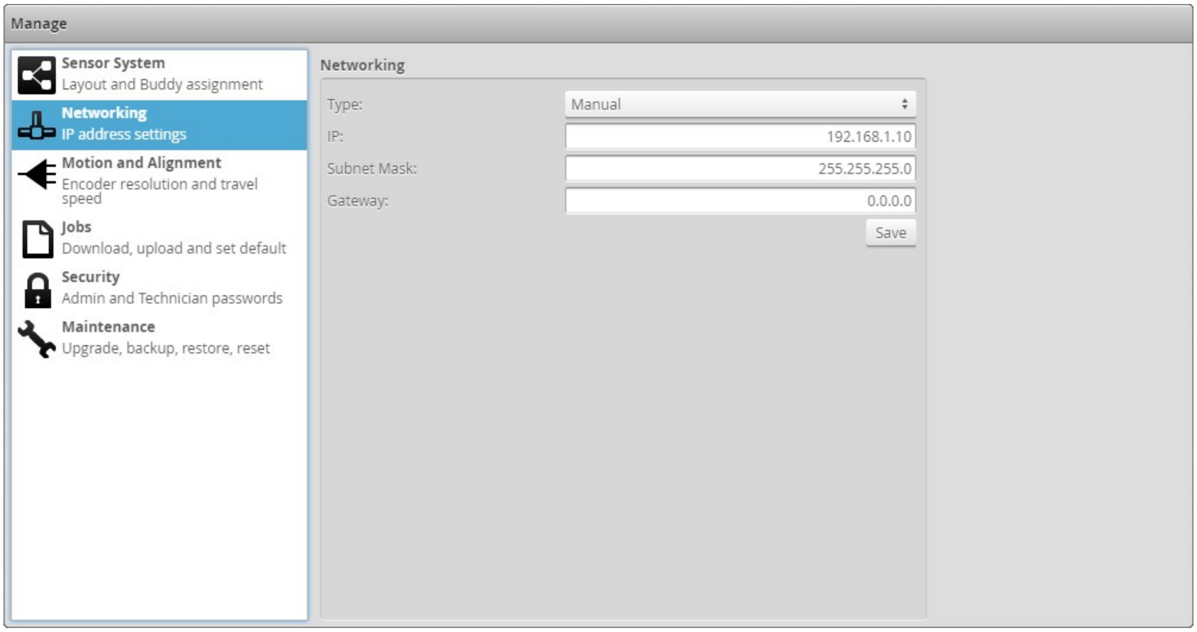
⚠	Disabling the setting is not recommended. Disabling the overheat protection feature could lead to premature laser failure if the sensor operates outside the specified temperature range.

To enable/disable overheat temperature protection:

1. Check/uncheck the **Over Temperature Protection** option.
2. Save the job file.

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.

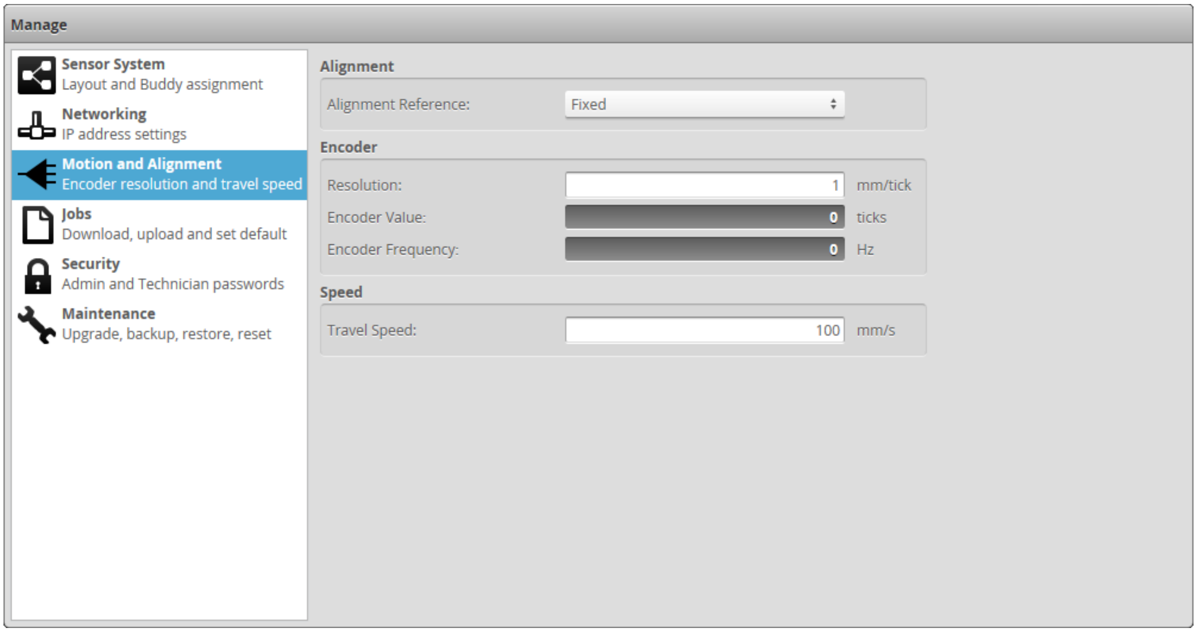


To configure the network settings:

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.

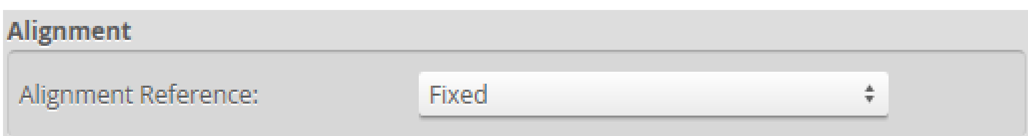
Motion and Alignment

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**.



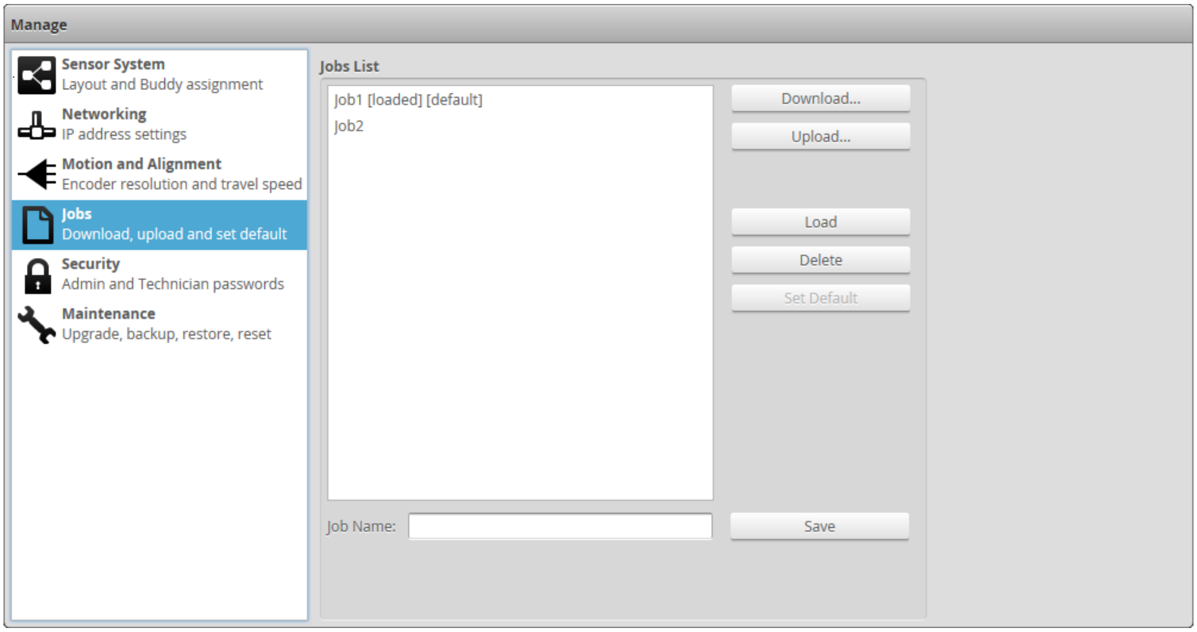
Setting	Description
Fixed	A single, global alignment is used for all jobs. This is typically used when the sensor mounting is constant over time and between scans, for example, when the sensor is mounted in a permanent position over a conveyor belt.
Dynamic	A separate alignment is used for each job. 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.

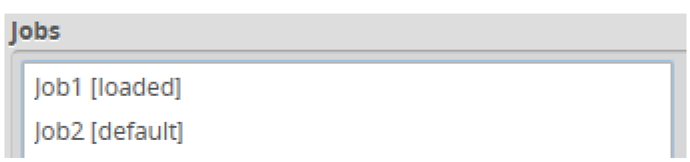
Jobs

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

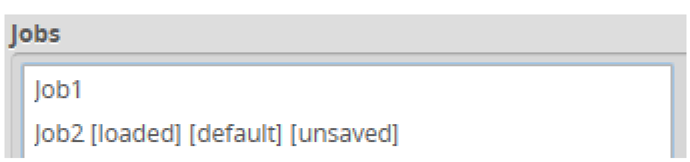


Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the sensor's flash storage.
Save button	Saves current settings to the job using the name in the **Job Name** field.
Load button	Loads the job that is selected in the job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

Jobs can be loaded (currently activated in sensor memory) 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 save a job:

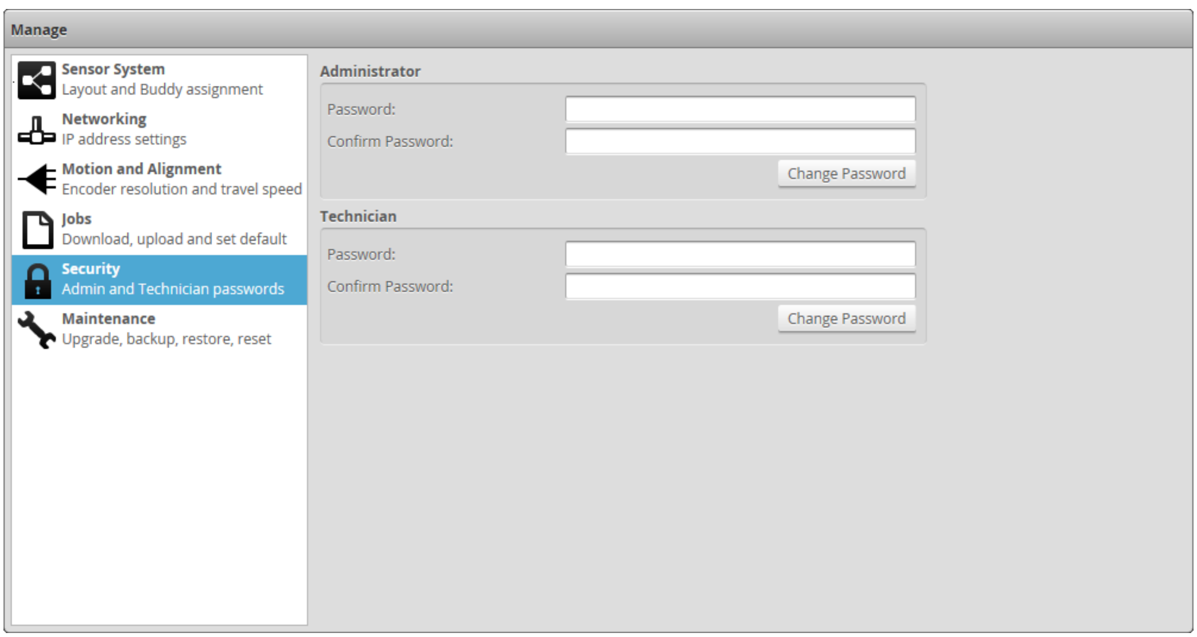
1. Go to the **Manage** page and click on the **Jobs** category.
2. Provide a name in the **Name** field.
   To save an existing job under a different name, click on it in the **Jobs** list and then modify it in the **Name** field.
3. Click on the **Save** button or press **Enter**.
   Saving a job automatically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

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.

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:

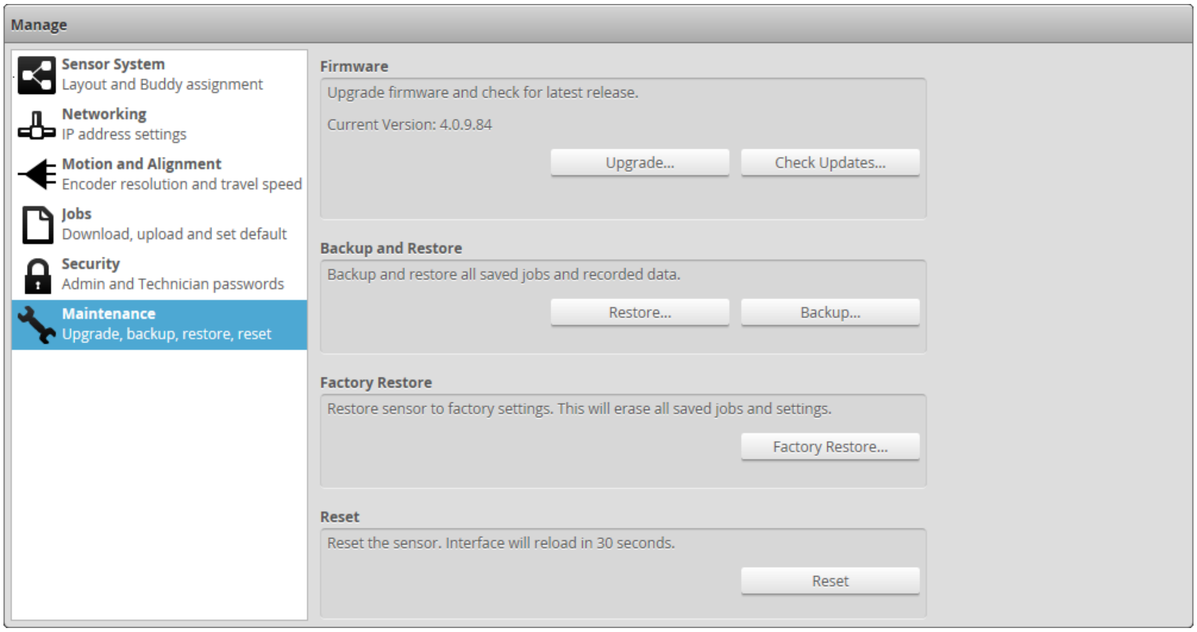
1. Go to the **Manage** page and click on the **Security** category.
2. In the **Technician** section, enter the Technician account password and password confirmation.
3. Click **Change Password**.
   The new password will be required the next time that a technician logs in to the sensor.

If the administrator or technician password is misplaced, the sensor can be recovered using a special software tool. See *Sensor Recovery Tool* on page 285 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.

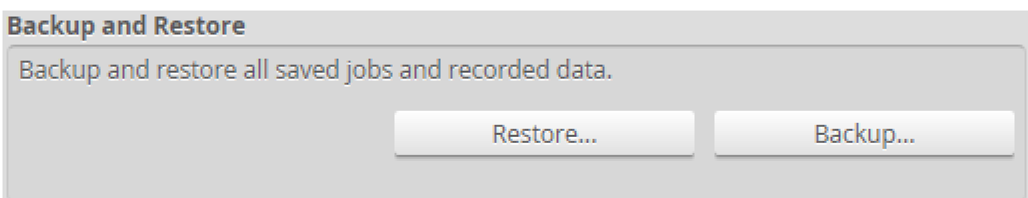


Sensor Backups and Factory Reset

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

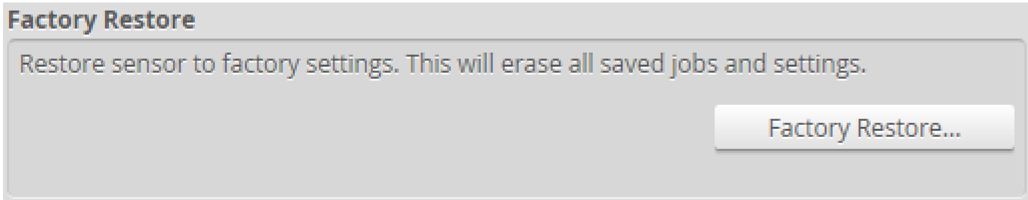
Backup files contain all of the information stored on a sensor, including jobs and alignment.

> 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:

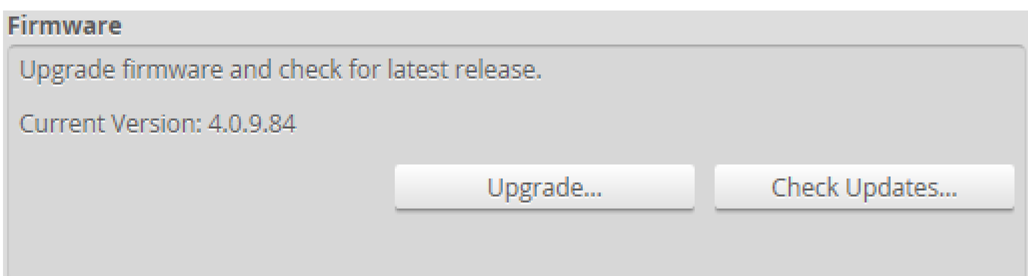
1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Restore...** button under **Backup and Restore**.
3. When you are prompted, select a backup file to restore.
   The backup file is uploaded and then used to restore the sensor. Any files that were on the sensor before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



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:

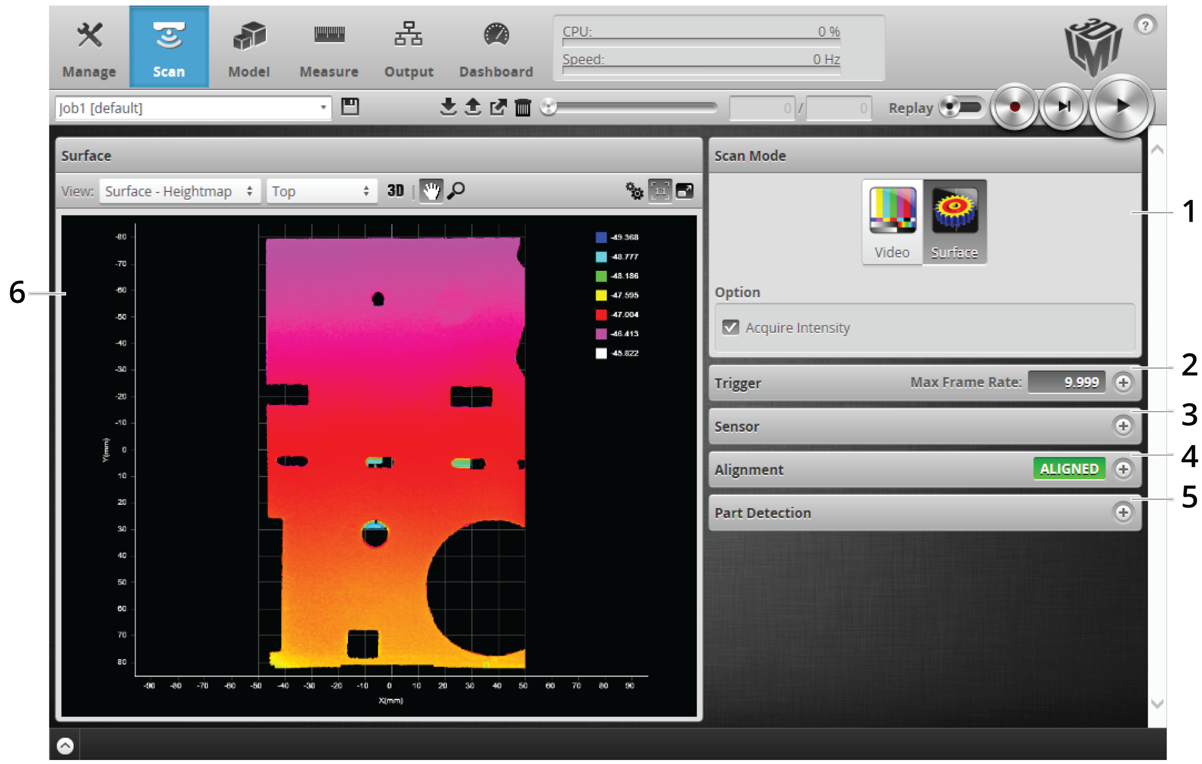
1. Go to the **Manage** page and click on the **Maintenance** category.
2. Click the **Upgrade...** button in the **Firmware** section.
3. Provide the location of the firmware file in the **File** dialog.
4. Wait for the upgrade to complete.
   After the firmware upgrade is complete, the sensor will 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 3D data acquisition 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.



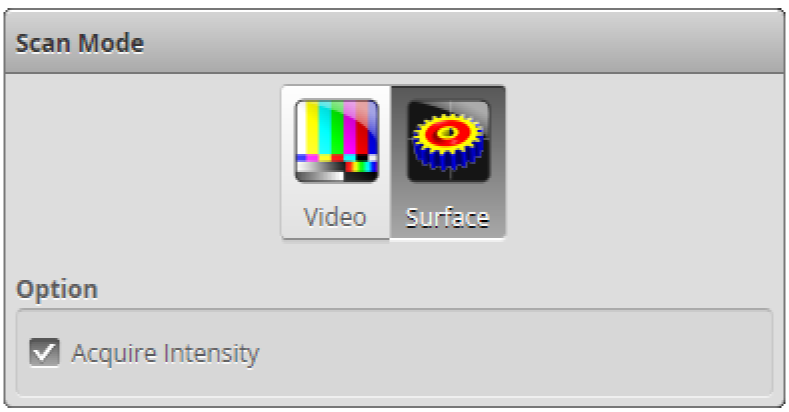
	Element	Description
1	Scan Mode panel	Contains settings for the current scan mode (Video or Surface) and other options. See *Scan Modes* on the next page.
2	Trigger panel	Contains trigger source and trigger-related settings. See *Triggers* on the next page.
3	Sensor panel	Contains settings for an individual sensor, such as active area or exposure. See *Sensor* on page 57.
4	Alignment panel	Used to perform alignment. See *Alignment* on page 64.
5	Part Detection panel	Used to set the part detection logic for sorting data into discrete objects. See *Part Detection* on page 70.
6	Data Viewer	Displays sensor data and adjusts regions of interest. Depending on the current operation mode, the data viewer can display video images or surface views. See *Data Viewer* on page 74.

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 (page 54)
Ensure that camera exposure is appropriate for 3D data acquisition.	Exposure (page 60)
Find the right balance between data quality, speed, and CPU utilization.	Active Area (page 58) Exposure (page 60) Job Files (page 169)
Calibrate the system so that 3D data can be aligned to a reference plane.	
Set up the part detection logic to create discrete objects from surfaces.	Part Detection (page 70)

Scan Modes

The Gocator web interface supports two scan modes: Video and Surface. 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.
Surface	Outputs 3D point clouds and performs surface measurements. Part detection can be enabled on a surface to identify discrete parts (see on page 70).
Acquire Intensity	When this option is enabled, an intensity value will be produced for each point in the 3D point cloud.

Triggers

A trigger is an event that causes a sensor to take a single 3D snapshot. Triggers are configured in the **Trigger** panel.

When a trigger is processed, the LED light pattern is strobed and the cameras expose to produce images. The resulting images are processed inside the sensor to yield a 3D point cloud, which can then be used for measurement.

The top-right of the **Trigger** panel displays the maximum speed at which an object could be captured at, calculated based on the exposure values, active area and the number of projection patterns required.

The light 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.
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.
Software	A network command can be used to send a software trigger. See *Protocols* on page 211 for more information.

Depending on the setup and measurement tools used, the CPU utilization may exceed 100%, which reduces the overall acquisition speed. If the **Clear Calibration** button is pressed, the calibration will be erased and the sensor will revert to using sensor coordinates. See *Acquisition Speed* on page 33 for the estimated acquisition speeds under different settings.

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

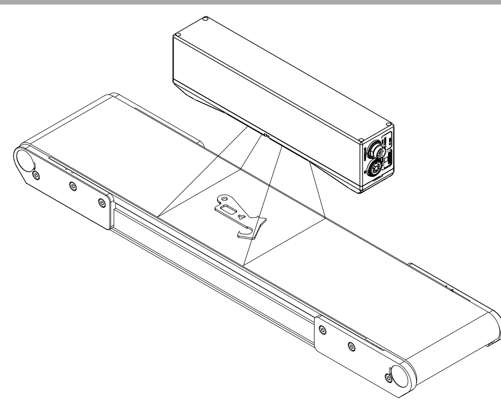
Trigger Examples

Example: External Input + Conveyor

External input triggering can be used to produce a snapshot for 3D 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.

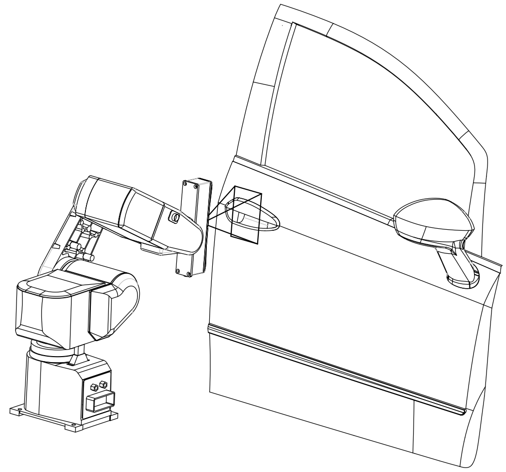
An external input can also be used to gate the trigger signals when time or encoder triggering is used. For example, a photocell could generate a series of trigger pulses as long as there is a target in position.

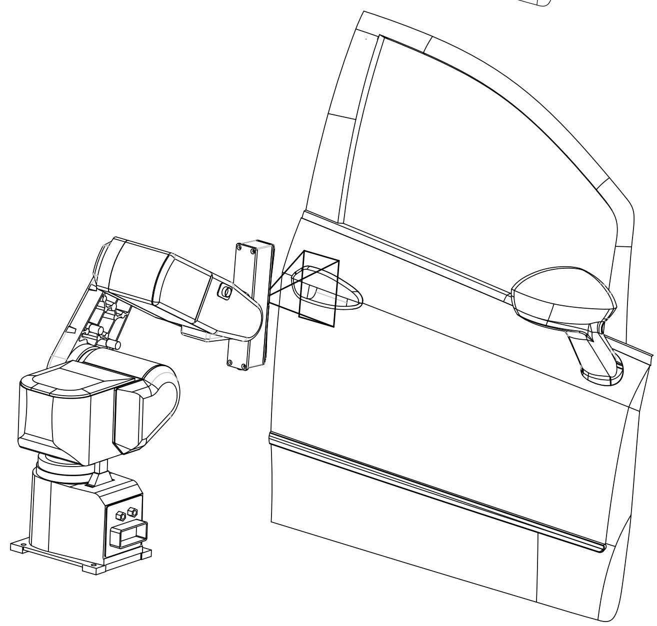


Example: Software Trigger + Robot Arm

Software triggering can be used to produce a snapshot for 3D measurement.

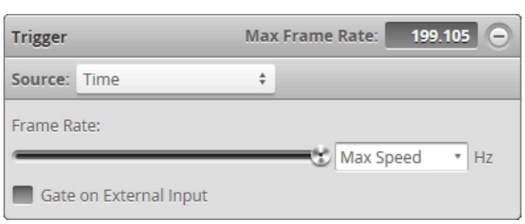
A software trigger can be used in systems that use external software to control the activities of system components.

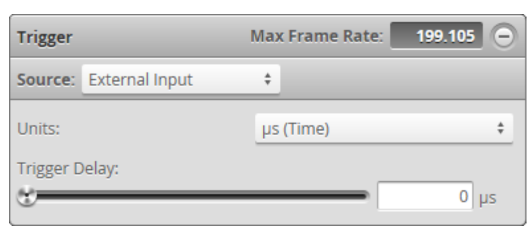


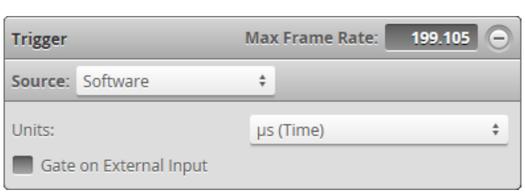


Trigger Settings

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**, **Encoder**, **External Input**, or **Software**).
Frame Rate	Time	Controls the frame rate. Select **Max Speed** from the drop-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 3D data acquisition 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 See *Digital Inputs* on page 306 for more information on connecting external input to Gocator sensors.
Units	External Input, Software	Specifies whether the trigger delay, output delay, and output scheduled command operate in the time domain. The unit is implicitly set to microseconds with Time 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* on page 60.

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.

Reduce Occlusion

When the option is disabled, 3D data is acquired using only stereo correlation between the two cameras, meaning that a point on the target must be visible in both cameras to produce a 3D data point.

When this option is enabled (default), each camera independently triangulates off the LED light pattern, which may improve performance on targets with complicated shapes that can cause occlusions. Stereo correlation is also used.

See *Stereo Correlation vs. Independent Triangulation* on page 29 for more information.

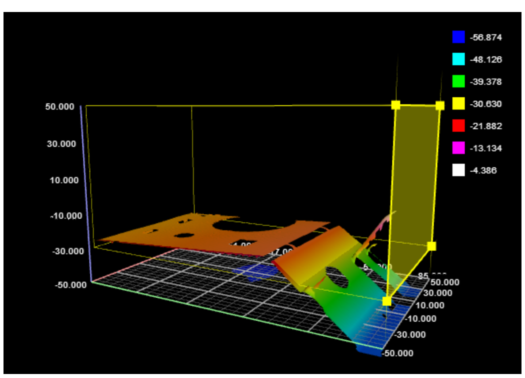
To enable or disable the Reduce Occlusion option:

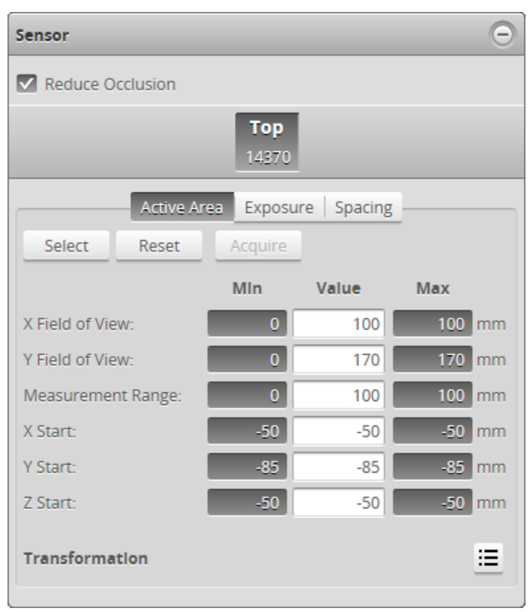
1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, the **Sensor** panel will not be displayed.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Check or uncheck the **Reduce Occlusion** checkbox.
5. Save the job in the **Toolbar** by clicking the **Save** button.

Active Area

Active area refers to the region within the sensor's maximum field of view that is used for 3D data acquisition.

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.





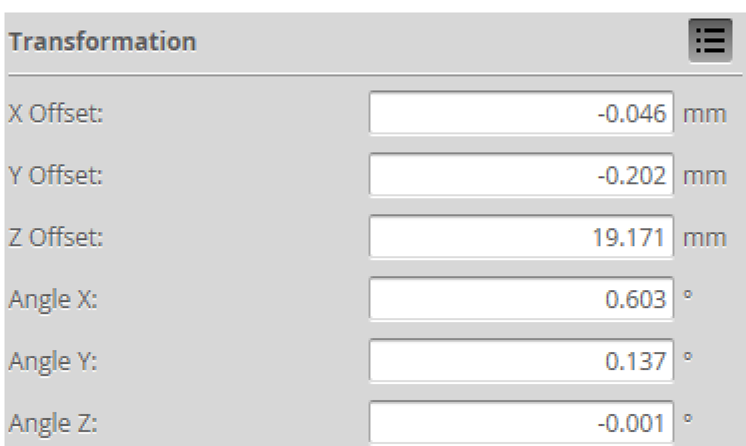
To set the active area:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click on the **Active Area** tab.
5. Click the **Select** button.
6. Click the **Acquire** button to see a scan while setting the active area.
7. Set the active area.
   Enter the active area values in the edit boxes or adjust the active area graphically in the data viewer.
   The 2D view allows you to adjust the sizes and positions in the X axis and Z axis. The 3D view allows you to adjust the sizes and positions in the X axis, Y axis, and Z axis. See *Regions* on page 101 for more information.
8. Click the **Save** button in the **Sensor** panel.
   Click the **Cancel** button to cancel setting the active area.
9. Save the job in the **Toolbar** by clicking the **Save** button.

> 3D acquisition 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 3D data are converted from sensor coordinates to system coordinates.



Parameter	Description
X Offset	Specifies the shift along the X axis. With Normal orientation, a positive value shifts the data to the

Parameter	Description
	right. With Reverse orientation, a positive value shifts the data to the left.
Y Offset	Specifies the shift along the Y axis.
Z Offset	Specifies the shift along the Z axis. A positive value shifts the data toward the sensor.
Angle X	Specifies the tilt around the X axis.
Angle Y	Specifies the tilt around the Y axis.
Angle Z	Specifies the tilt around the Z axis.

When applying the transformations, Angle is applied before the offsets: the object is first rotated around X, then Y, and then Z.

To configure transformation settings:

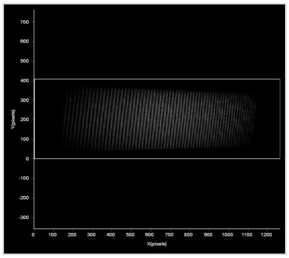
1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Expand the Transformations area by clicking on the expand button.
   See the table above for more information.
5. Set the parameter values.
   See the table above for more information.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the transformation settings are applied correctly after 3D acquisition is restarted.

Exposure

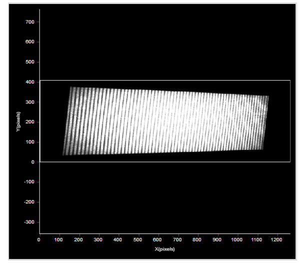
Exposure determines the duration of camera and light on-time. Longer exposures can be helpful to detect light patterns on dark or distant surfaces, but increasing exposure time decreases the maximum speed. Different target surfaces may require different exposures for optimal results. Gocator sensors 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.
Multiple	Uses multiple exposures to create a single scan. Used when the target surface has a varying reflectance within a single scan (e.g., white and black).

Video mode lets you see how the reflected light pattern appears on the camera and identify any stray light or ambient light problems. When exposure is tuned correctly, the light pattern 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.

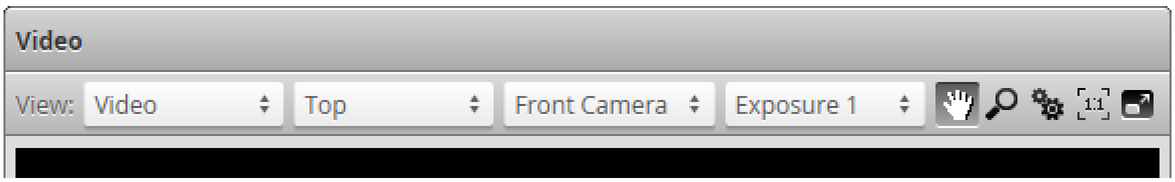


Under exposure
Light pattern is not fully detected.
Increase the exposure value.



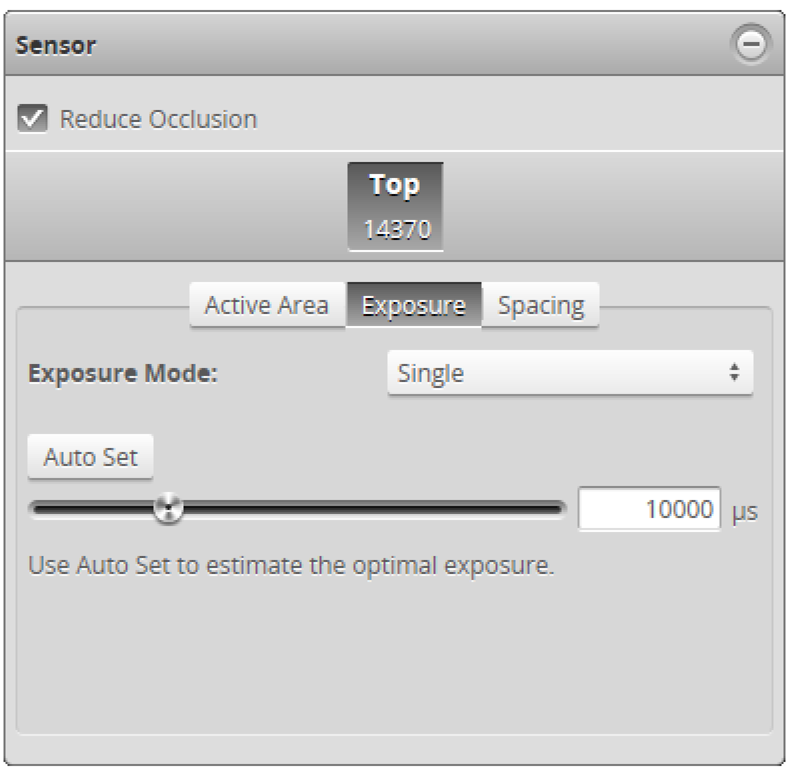
Over exposure
Light pattern saturated in the center.
Increase the exposure value.

When the Gocator is in Multiple exposure mode, select which exposure to view using the second drop-down box next to "View" in the data viewer. This drop-down is only visible in Video scan mode when the **Multiple** option is selected in the **Exposure** section in the **Sensor** panel.



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 parts.



To enable single exposure:

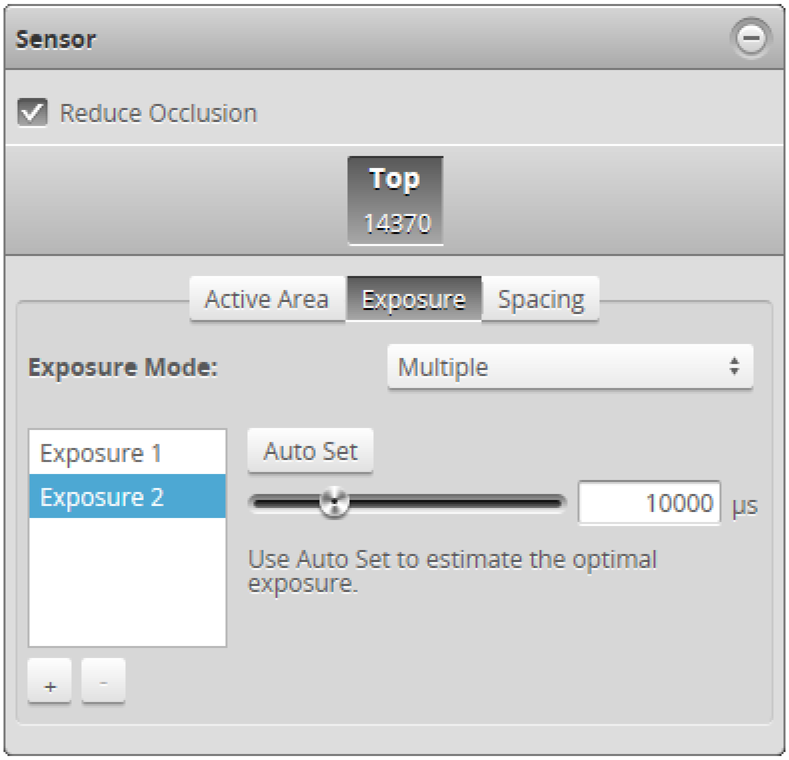
1. Place a representative target in view of the sensor.
   The target surface should be similar to the material that will normally be measured.
2. Go to the **Scan** page.
3. Expand the **Sensor** panel by clicking on the panel header.
4. Click on the **Exposure** tab.
5. Select **Single** from the **Exposure Mode** drop-down.
6. 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.
7. Run the sensor and check that 3D data acquisition is satisfactory.
   If 3D data acquisition is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* on page 60 for details.

Multiple Exposure

The sensor combines data from multiple exposures to create a single 3D point cloud. Multiple exposures can be used to increase the ability to detect light and dark materials that are in the field of view simultaneously.

Up to three exposures can be defined with each set to a different exposure level. For each exposure, the sensor will perform a complete scan at the current frame rate making the effective frame rate slower. For example, if two exposures are selected, then the speed will be half of the single exposure frame rate. The sensor will perform a complete multi-exposure scan for each external input.

The resulting 3D point cloud is a composite created by combing data collected with different exposures.

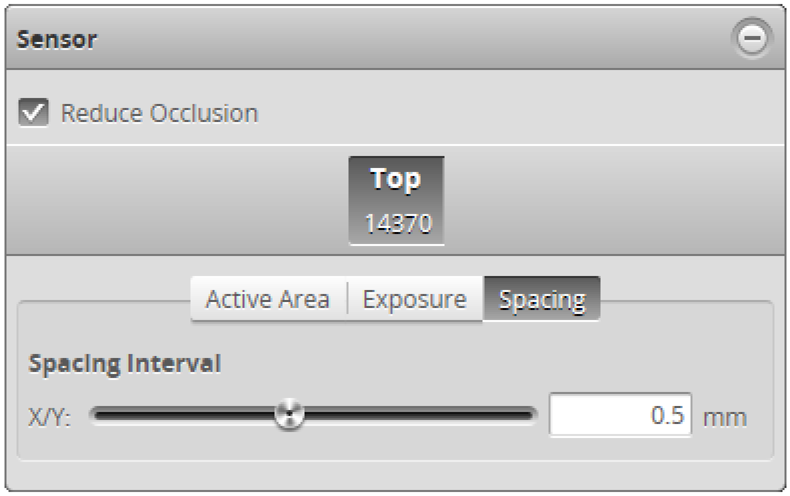


To enable multiple exposure:

1. Go to the **Scan** page.
2. Expand the **Sensor** panel by clicking on the panel header or the ⊕ button.
3. Click on the **Exposure** tab.
4. Select **Multiple** from the **Exposure Mode** drop-down.
5. Click the + button to add an exposure step.
   Up to a maximum of three exposure settings can be added.
   To remove an exposure, select it in the exposure list and click the - button.
6. Set the exposure level for each exposure to make the Gocator's camera less or more sensitive, as required.
   If **Acquire Intensity** is enabled, select the exposure step that is used to capture the intensity output.
7. If **Acquire Intensity** is enabled, select the exposure step that is used to capture the intensity output.
8. Run the sensor and check that 3D data acquisition is satisfactory.
   If 3D data acquisition is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* on page 60 for details.

Spacing

The **Spacing** tab lets you configure the spacing interval.



Spacing Interval

Spacing interval is the spacing between data points in resampled data. A larger interval creates scans with lower X/Y resolution, reduces CPU usage, and potentially increases the maximum frame rate. A larger interval also reduces the data output rate. For more information on resampled data, see *Resampling* on page 33.

To configure the spacing interval:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.

   If this mode is not selected, you will not be able to configure the spacing interval.
3. Expand the **Sensor** panel by clicking on the panel header or the button.
4. Click on the **Spacing** tab.
5. Select a spacing interval level.
6. Save the job in the **Toolbar** by clicking the **Save** button.

Alignment

Gocator sensors are pre-calibrated and ready to deliver 3D data in engineering units (mm) out of the box. However, alignment procedures are required to compensate for sensor mounting inaccuracies. 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* on page 59 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. 3D point clouds are reported in default sensor coordinates.
Manual	Transformations (see on page 59) 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.

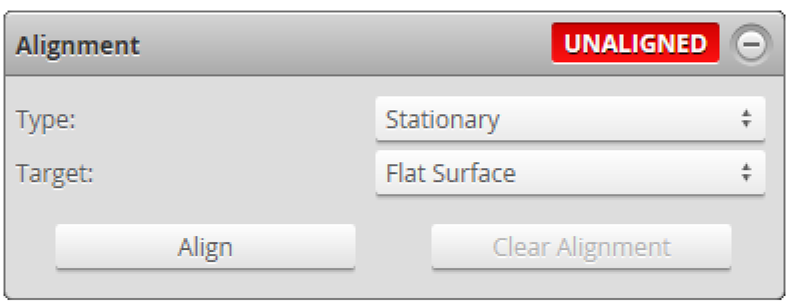
Alignment Types

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.

Aligning Sensors

Alignment can be used to compensate for mounting inaccuracies by aligning sensor data to a common reference surface.



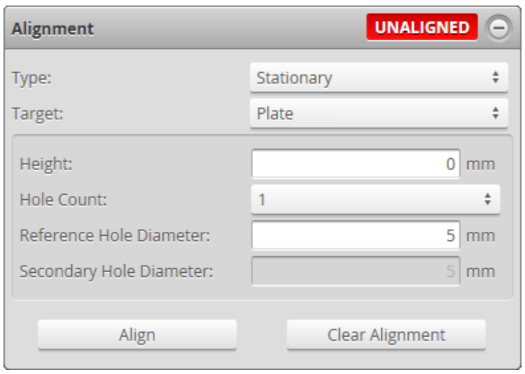
To prepare for alignment:

1. Choose an alignment reference in the **Manage** page if you have not already done so.
   See *Alignment Reference* on page 46 for more information.
2. Go to the **Scan** page.
3. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
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 **Plate** to use a custom calibration plate. If using a calibration plate, specify the bar dimensions and reference hole layout. See *System Coordinates* on page 32 for more information on plates.

     Configure the characteristics of the target.



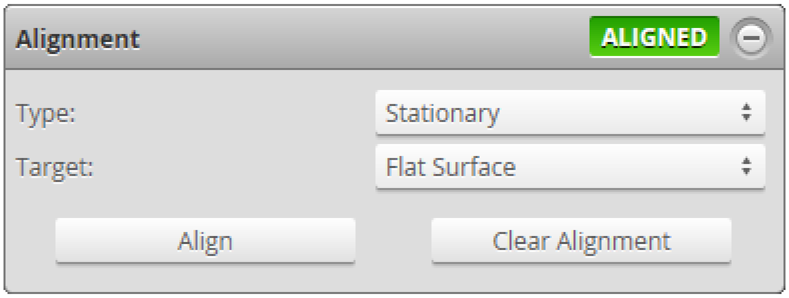
4. Place the target under the sensor
5. Click the **Align** button.
   The sensors will start, and the alignment process will take place. Alignment is performed simultaneously for all sensors. If the sensors do not align, check and adjust the exposure settings (page 60).

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

6. Check the alignment results under **Transformation** in the **Active Area** tab in the **Sensor** panel.

Clearing Alignment

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



To clear alignment:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
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.

Filters

Filters are used to post-process data along the X or Y axis to remove noise or clean it up before it is output or is used by measurement tools. The following types of filters are supported:

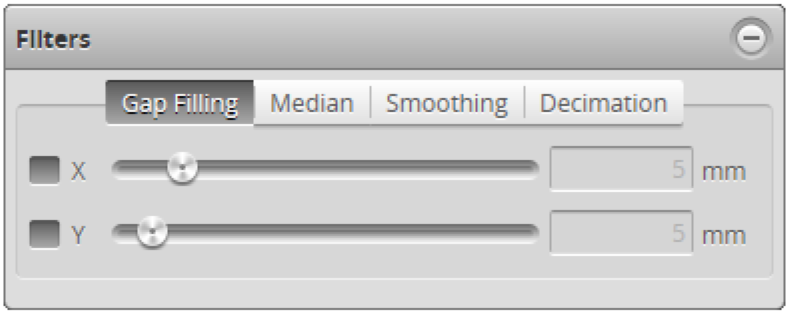
Filter	Description
Gap Filling	Fills in missing data caused by occlusions using information from the nearest neighbors. Gap filling also fills gaps where no data is detected, which can be due to the surface reflectivity, for example dark or specular surface areas, or to actual gaps in the surface.
Median	Substitutes the value of a data point with the median within a specified window around the data point.
Smoothing	Applies moving window averaging to reduce random noise.
Decimation	Reduces the number of data points.

Filters are applied in the order displayed in the table above. The filters are configured in the **Filters** panel on the **Scan** page.

Gap Filling

Gap filling works by filling in missing data points using either the lowest values from the nearest neighbors or linear interpolation between neighboring values (depending on the Z difference between neighboring values), in a specified window. The sensor can fill gaps along both the X axis and the Y axis.

If both X and Y gap filling are enabled, missing data is filled along the X and Y axes at the same time, using the available neighboring data.



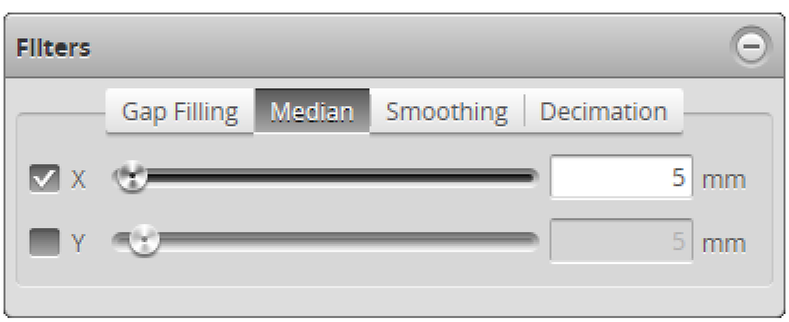
To configure X or Y gap filling:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure gap filling.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Gap Filling** tab.
5. Enable the **X** or **Y** setting and select the maximum width value.
   The value represents the maximum gap width that the Gocator will fill. Gaps wider than the maximum width will not be filled.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the laser profiling is satisfactory.

Median

The Median filter substitutes the value of a data point with the median calculated within a specified window around the data point.

	Missing data points will not be filled with the median value calculated from data points in the neighbourhood.



To configure X or Y median:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.

If this mode is not selected, you will not be able to configure gap filling.

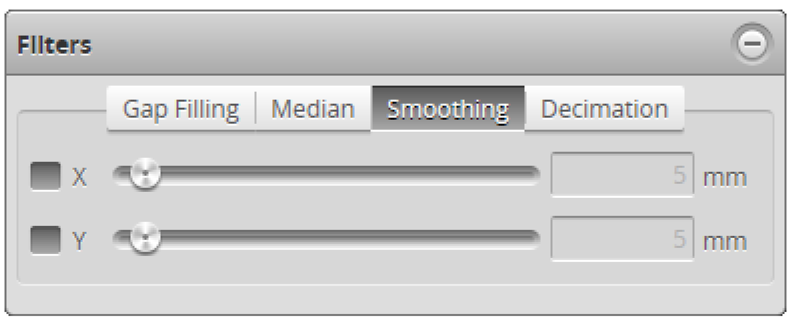
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Median** tab.
5. Enable the **X** or **Y** setting and select the maximum width value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the laser profiling is satisfactory.

Smoothing

Smoothing works by substituting a data point value with the average value of that data point and its nearest neighbors within a specified window. Smoothing can be applied along the X axis or the Y axis. X smoothing works by calculating a moving average across samples along the X axis. Y smoothing works by calculating a moving average along the X axis.

If both X and Y smoothing are enabled, the data is smoothed along X axis first, then along the Y axis.

	Missing data points will not be filled with the mean value calculated from data points in the neighbourhood.

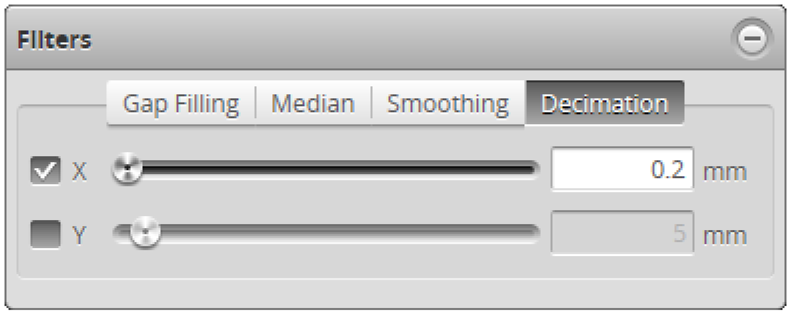


To configure X or Y smoothing:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure smoothing.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Smoothing** tab.
5. Enable the **X** or **Y** setting and select the averaging window value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the laser profiling is satisfactory.

Decimation

Decimation reduces the number of data points along the X or Y axis by choosing data points at the end of a specified window around the data point. For example, by setting X to .2, points will be used every .2 millimeters.

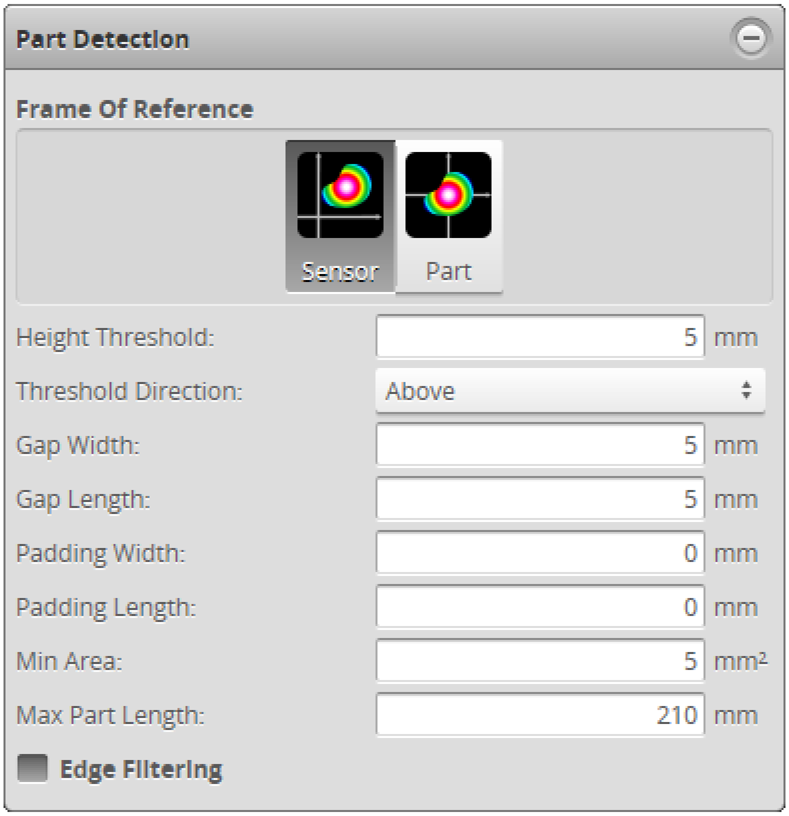


To configure X or Y decimation:

1. Go to the **Scan** page.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure smoothing.
3. Expand the **Filters** panel by clicking on the panel header or the button.
4. Click on the **Decimation** tab.
5. Enable the **X** or **Y** setting and select the decimation window value.
6. Save the job in the **Toolbar** by clicking the **Save** button.
7. Check that the laser profiling is satisfactory.

Part Detection

Multiple parts can be detected from a single surface and will be individually tracked.



The following settings can be tuned to improve the accuracy and reliability of part detection.

Setting	Description
Height Threshold	Determines the height threshold for part detection. The setting for **Threshold Direction** determines if parts should be detected above or below the threshold. Above is typically used to prevent the belt surface from being detected as a part when scanning objects on a conveyor.
Threshold Direction	Determines if parts should be detected above or below the height threshold.
Gap Width	Determines the minimum separation between objects on the X axis. If parts are closer than the gap interval, they will be merged into a single part.
Gap Length	Determines the minimum separation between objects on the Y axis. If parts are closer than the gap interval, they will be merged into a single part.
Padding Width	Determines the amount of extra data on the X axis from the surface surrounding the detected part that will be included. This is mostly useful when processing part data with third-party software such as HexSight, Halcon, etc.
Padding Length	Determines the amount of extra data on the Y axis from the surface surrounding the detected part that will be included. This is mostly useful when processing part data with third-party software such as HexSight, Halcon, etc.
Min Area	Determines the minimum area for a detected part. Set this value to a reasonable minimum in order to filter out small objects or noise.
Max Part Length	Determines the maximum length of the part object. When the object exceeds the

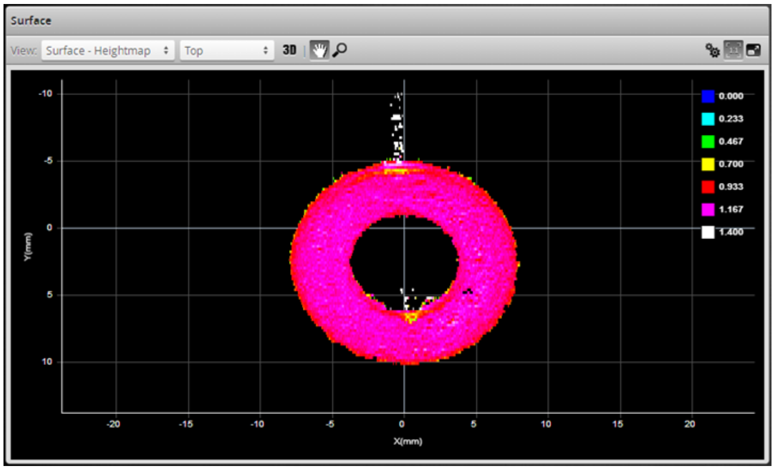
Setting	Description
	maximum length, it is automatically separated into two parts. This is useful to break a long object into multiple sections and perform measurements on each section.
Frame of Reference	Determines the coordinate reference for surface measurements. **Sensor** When **Frame of Reference** is set to **Sensor**, the sensor's frame of reference is used. When set to **Sensor**, all measurement values are relative to the sensor's field of view in X and relative to the encoder zero position in Y. **Part** When **Frame of Reference** is set to **Part**, all measurements except Bounding Box X and Y are relative to the center of the bounding box of the part. For Bounding Box X and Y, the measurement values are always relative to the sensor frame of reference (see *Bounding Box* on page 106).
Edge Filtering	See *Edge Filtering* below.

To set up part detection:

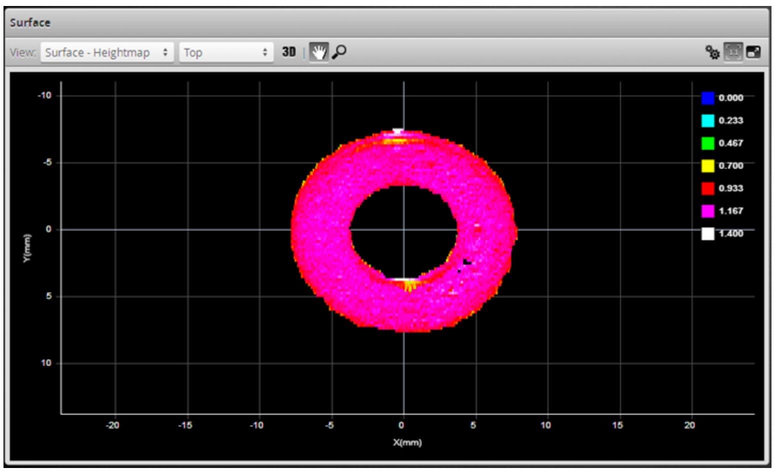
1. Go to the **Scan** page and choose **Surface** in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure part detection.
2. Expand the **Part Detection** panel by clicking on the panel header or the button.
3. If necessary, check the **Enabled** option.
   When **Generation** is set to **Continuous**, part detection is always enabled.
4. Adjust the settings.
   See the part detection parameters above for more information.

Edge Filtering

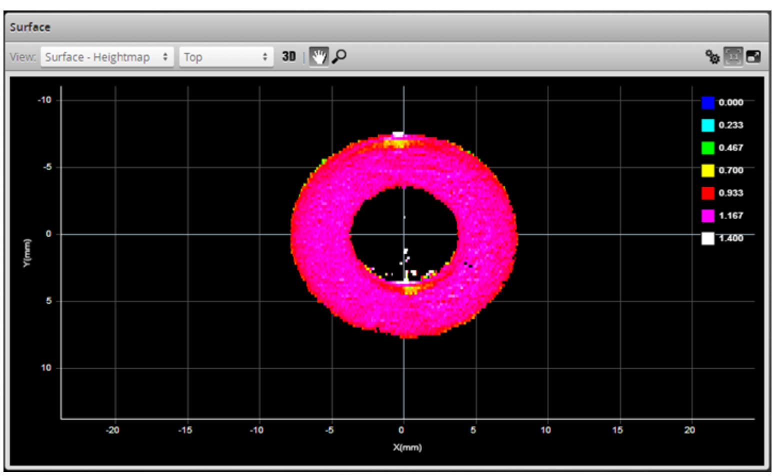
Part scans sometimes contain noise around the edges of the target. This noise is usually caused by the sensor’s light being reflected off almost vertical sides, rounded corners, etc. Edge filtering helps reduce edge noise in order to produce more accurate and repeatable volume and area measurements, as well as to improve positioning of relative measurement regions. Optionally, the **Preserve Interior Feature** setting can be used to limit filtering to the outside edges of the target.



Edge Filtering disabled (scan shows reflection noise)



Edge Filtering enabled (reflection noise eliminated or reduced)



Edge Filtering enabled, Preserve Interior Feature enabled.

To configure edge filtering:

1. Go to the **Scan** page and choose **Surface** in the **Scan Mode** panel.
   If this mode is not selected, you will not be able to configure part detection.
2. Expand the Part Detection panel by clicking on the panel header or the button and enable part detection if necessary.
3. Check the **Edge Filtering** checkbox to enable edge filtering.
4. Configure the **Width** and **Length** settings.
   The **Width** and **Length** settings represent the size of the filter on the X axis and the Y axis, respectively.
5. Set the **Preserve Interior Feature** setting if necessary.
   The **Preserve Interior Feature** setting limits filtering to the outside edges of the target.

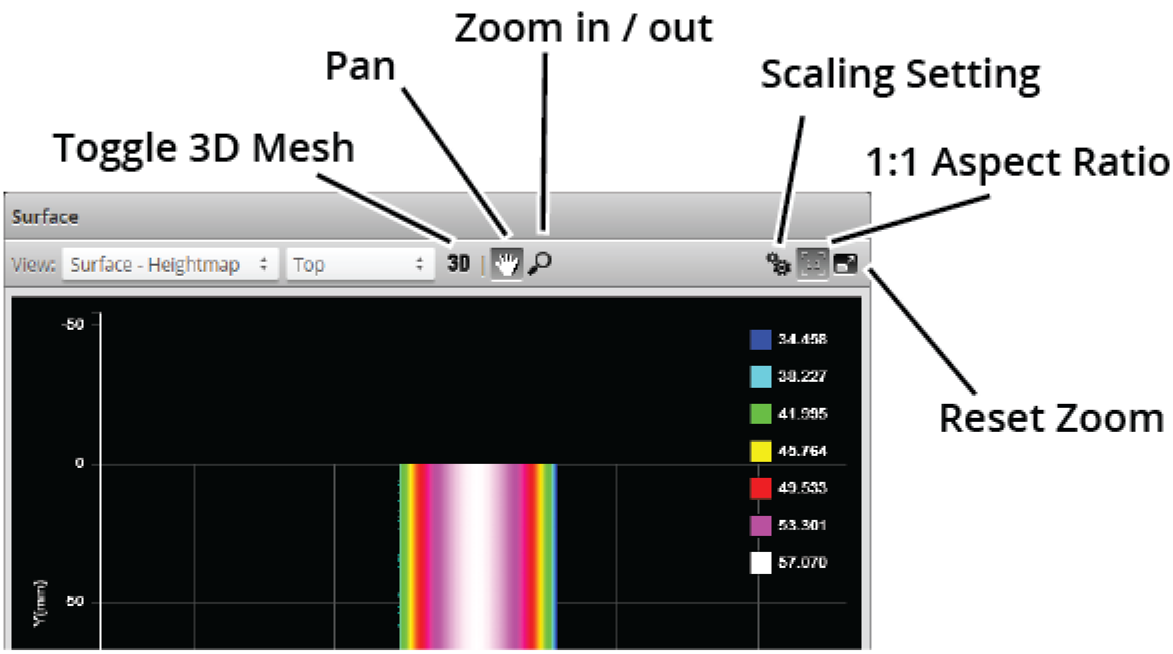
Data Viewer

The data viewer can display images in a 2D view and height map (resampled point cloud) or intensity in 2D or 3D views. The data viewer changes depending on the current operation mode and the panel that has been selected. Use the drop down list at the top-left corner to select the data source to view. The available data sources depend on the operation mode settings.

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

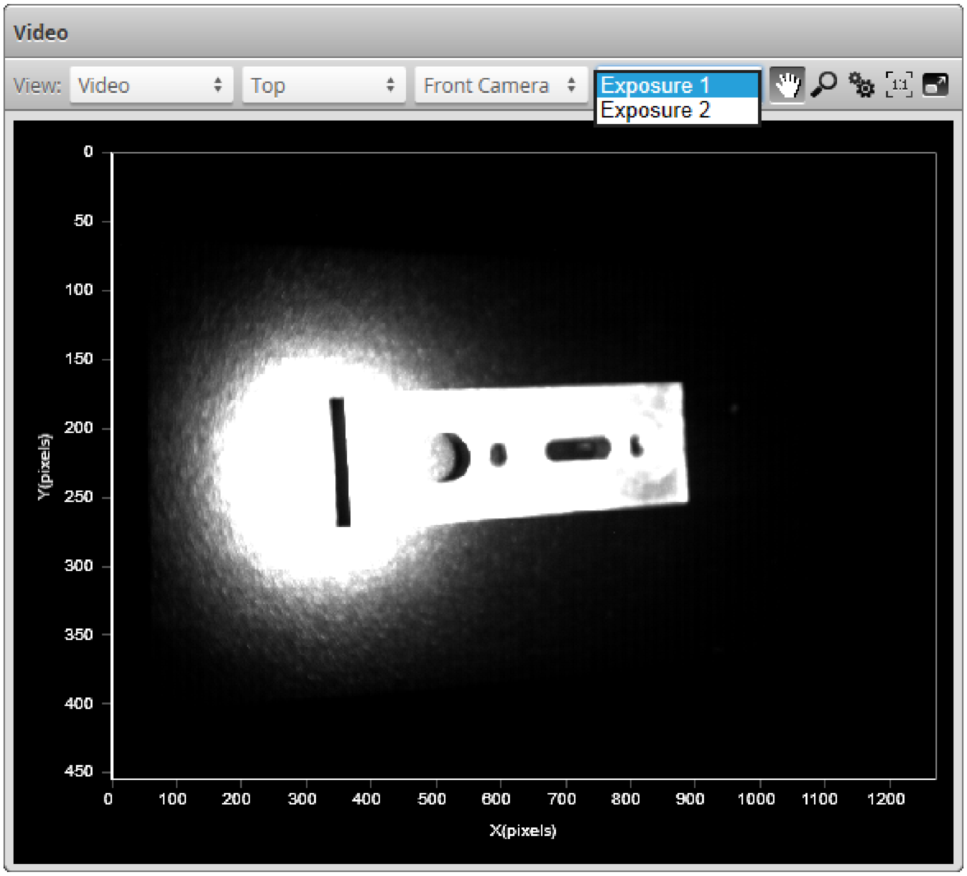
In Video scan mode, the data viewer displays a camera image from either of the two cameras in the sensor.

Exposure View

Exposure can be set to **Dynamic**, or **Multiple**. Different exposures can be displayed in the data viewer if more than one exposure has been set. For details on setting exposure in the **Exposure** tab in the **Sensor** panel, see on page 60.

If the **Multiple** option is selected in the **Exposure** tab and multiple exposures have been set, a drop-down is displayed at the top of the data viewer. This drop-down lists the exposures that have been added. Choosing an exposure changes the view of the data viewer to that exposure.

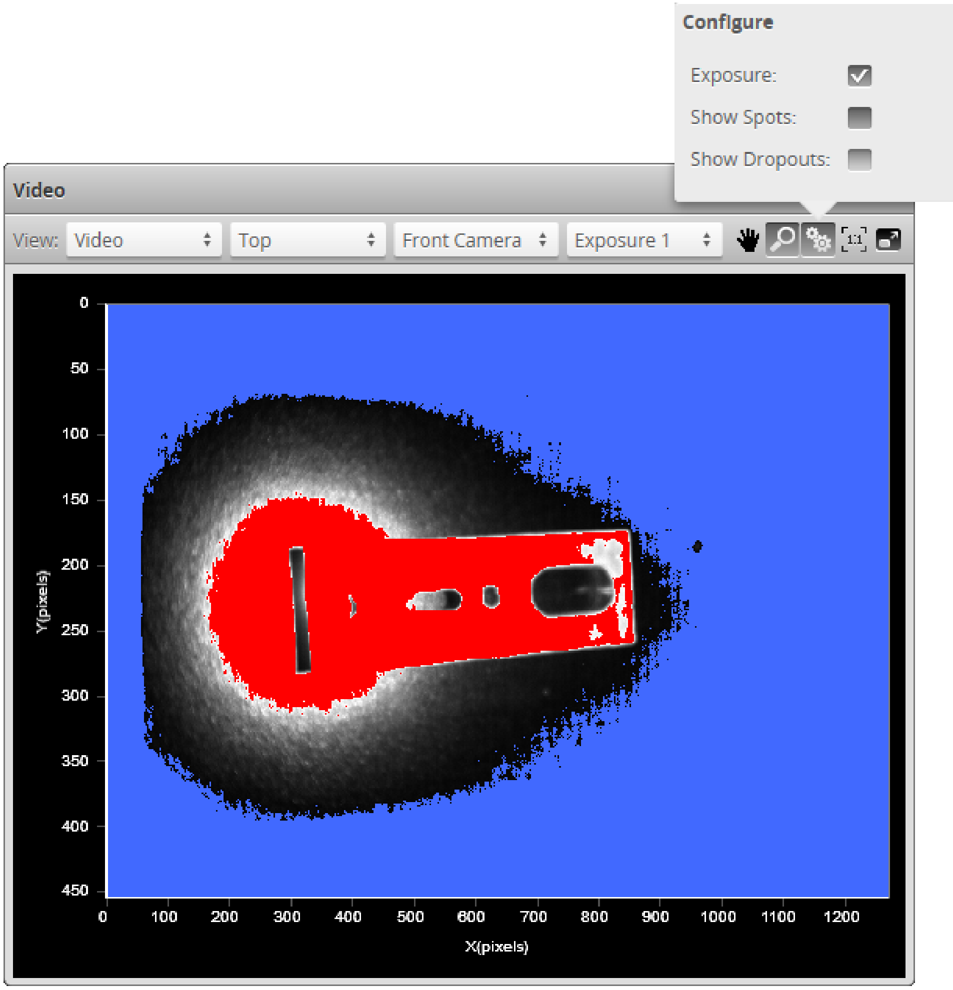
If the **Single** option is selected in the **Exposure** tab, the exposure drop-down will not be shown and only one exposure will be displayed.



To select the exposure view of the display:

1. Go to the **Scan** page and choose **Video** mode in the **Scan Mode** panel.
2. Select the camera view in the data viewer.
   Select **Front Camera** or **Back Camera** from the first drop-down list next to **View** at the top of the data viewer.
3. Select the exposure.
   Select the exposure from the second drop-down list next to **View** at the top of the data viewer. This drop-down is only be visible if the **Multiple** option has been selected.

To display a color exposure overlay on the video image to help set the correct exposure, check **Exposure Indication** at the top of the data viewer.



Exposure Indication uses the following colors:

- Blue: Indicates background pixels ignored by the sensor.
- Red: Indicates saturated pixels.

Correct tuning of exposure depends on the reflective properties of the target material and on the requirements of the application. Settings should be carefully evaluated for each application.

The Gocator 3100 cannot generate 3D points in over-saturated (areas indicated with red) or in under-exposed areas (indicated with blue). If it's not possible to set a single exposure to capture the entire object target without red areas appearing in the image, the **Multiple** exposure feature should be enabled. Use the drop-down selection box to view each exposure and tune one high exposure for dark areas on the target and one low exposure for bright areas on the target. Note that multiple exposures reduce the maximum speed the sensor can run at.

Surface Mode

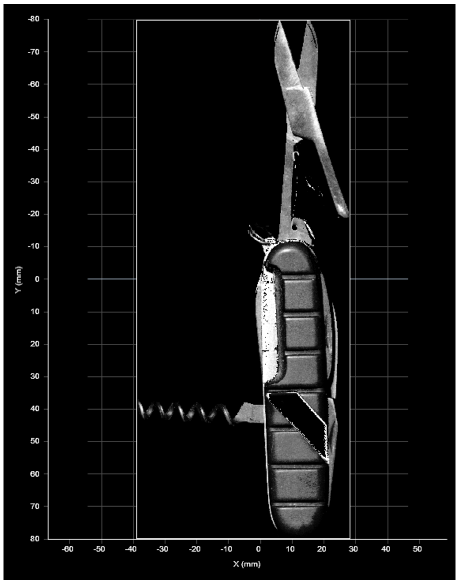
When the Gocator is in Surface scan mode, the data viewer can display height maps and intensity images. You can select the data to display from the **View** drop-down.

Clicking on the **3D** button displays Surface data in the 3D viewer. The 3D model is overlaid with the information that corresponds to the selected **View** option.

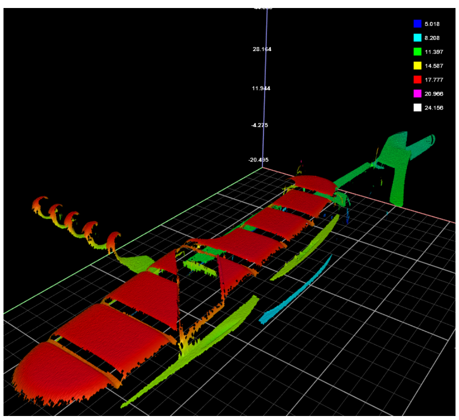
View Option	Information
Surface - Heightmap	In 2D view, displays the pseudo color height map. In 3D view, overlays the 2D pseudo color height map on the 3D model.
Surface - Grayscale	In 2D view, displays the grayscale height map. In 3D view, overlays the grayscale height map on the 3D model.
Surface - Uniform	Overlays a uniform shaded surface on the 3D model. (Only available in 3D view.)
Surface - Intensity	In 2D view, displays the intensity. In 3D view, overlays the intensity map on the 3D model. (**Acquire Intensity** must be checked in the **Scan Mode** panel.)

2D viewer with height map overlay

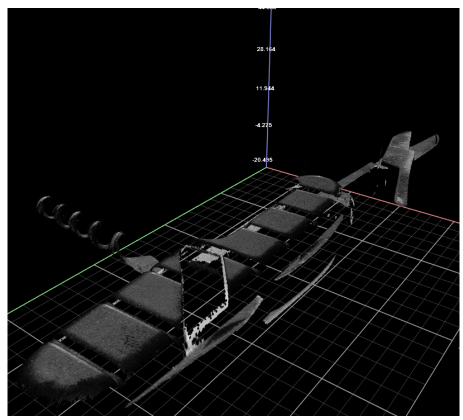
2D viewer with grayscale overlay



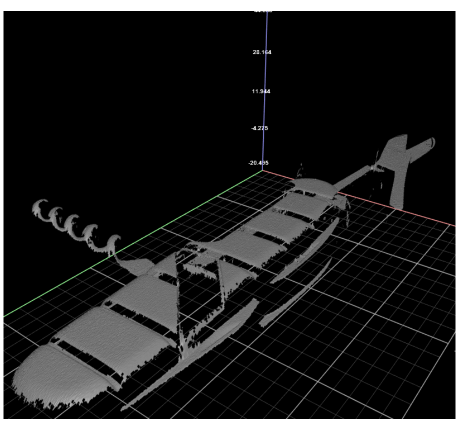
2D viewer with intensity overlay



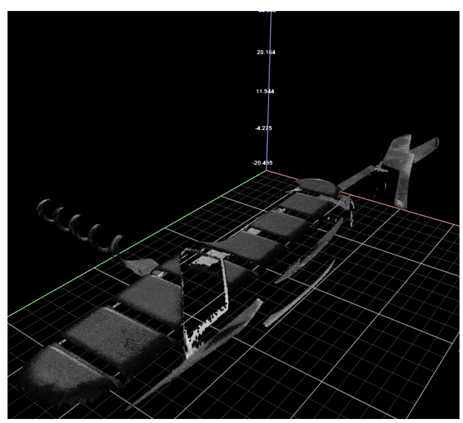
3D viewer with height map overlay



3D viewer with grayscale overlay



3D viewer with uniform overlay



3D viewer with uniform overlay

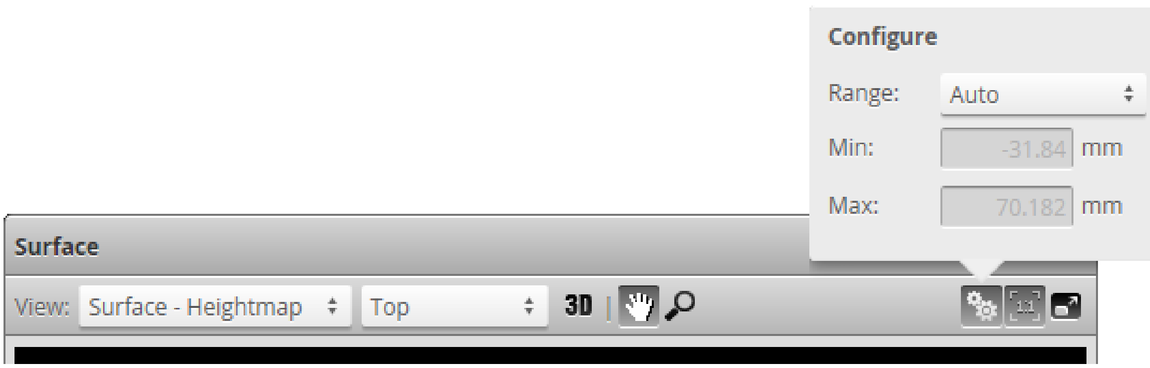
Clicking on the **3D** button toggles between the 2D and 3D viewer. See *Data Viewer Controls* on page 74 for explanations on the available controls.

To manually select the display type and the display view in the Scan page:

1. Go to the **Scan** page.
2. Select the View options in the data viewer.
   Heightmap, **Grayscale**, **Uniform**, or **Intensity** can be selected from the left drop-down list.

Height Map Color Scale

Height maps are displayed in pseudo-color. The height axis (Z) is color-coded. The scaling of the height map can be adjusted.



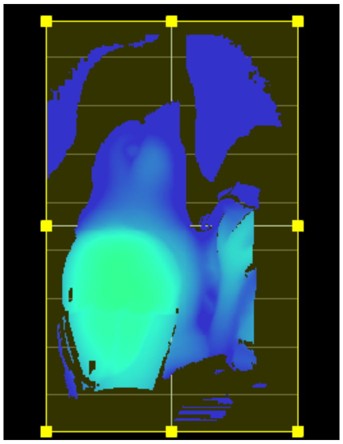
To change the scaling of the height map:

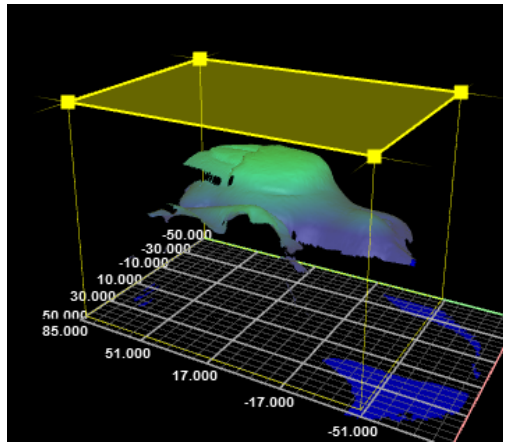
1. Select **Heightmap** from the drop-down in the data viewer.
2. Click the **Scaling** button.
   - To automatically set the scale, choose **Auto** in the **Range** drop-down.
   - To automatically set the scale based on a user-selected sub-region of the heightmap, choose **Auto - Region** in the **Range** drop-down and adjust the yellow region box in the data viewer to the desired location and size.
   - To manually set the scale, choose the **Manual** in the **Range** drop-down and enter the minimum and maximum height to which the colors will be mapped.

Region Definition

Regions, such as an active area or a measurement region, can be graphically set up using the data viewer in the 2D or in the 3D view.

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 on page 57.





To set up a region of interest:

1. Move the mouse cursor to the rectangle. In the 3D viewer, you must first select which side of the 3D rectangle to adjust by clicking on it.

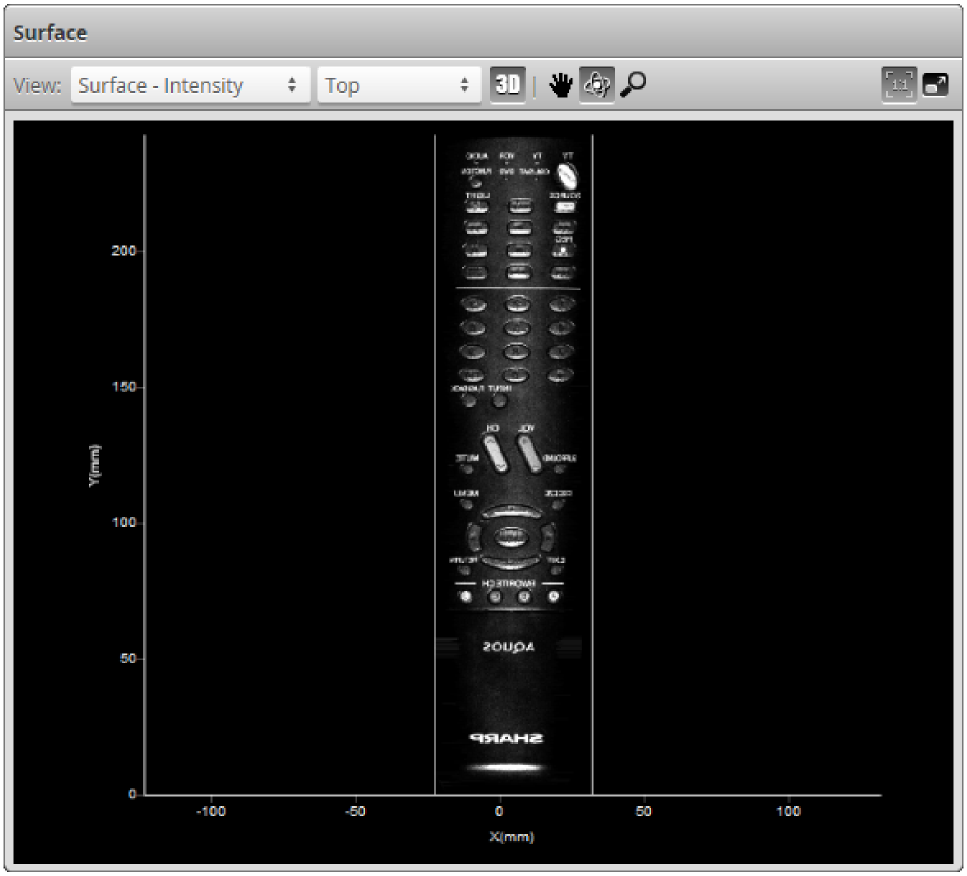
   The rectangle is automatically displayed when a setup or measurement requires an area to be specified.

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 images that measure the amount of light reflected by an object. An 8-bit intensity value is output for each point in the 3D point cloud.

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



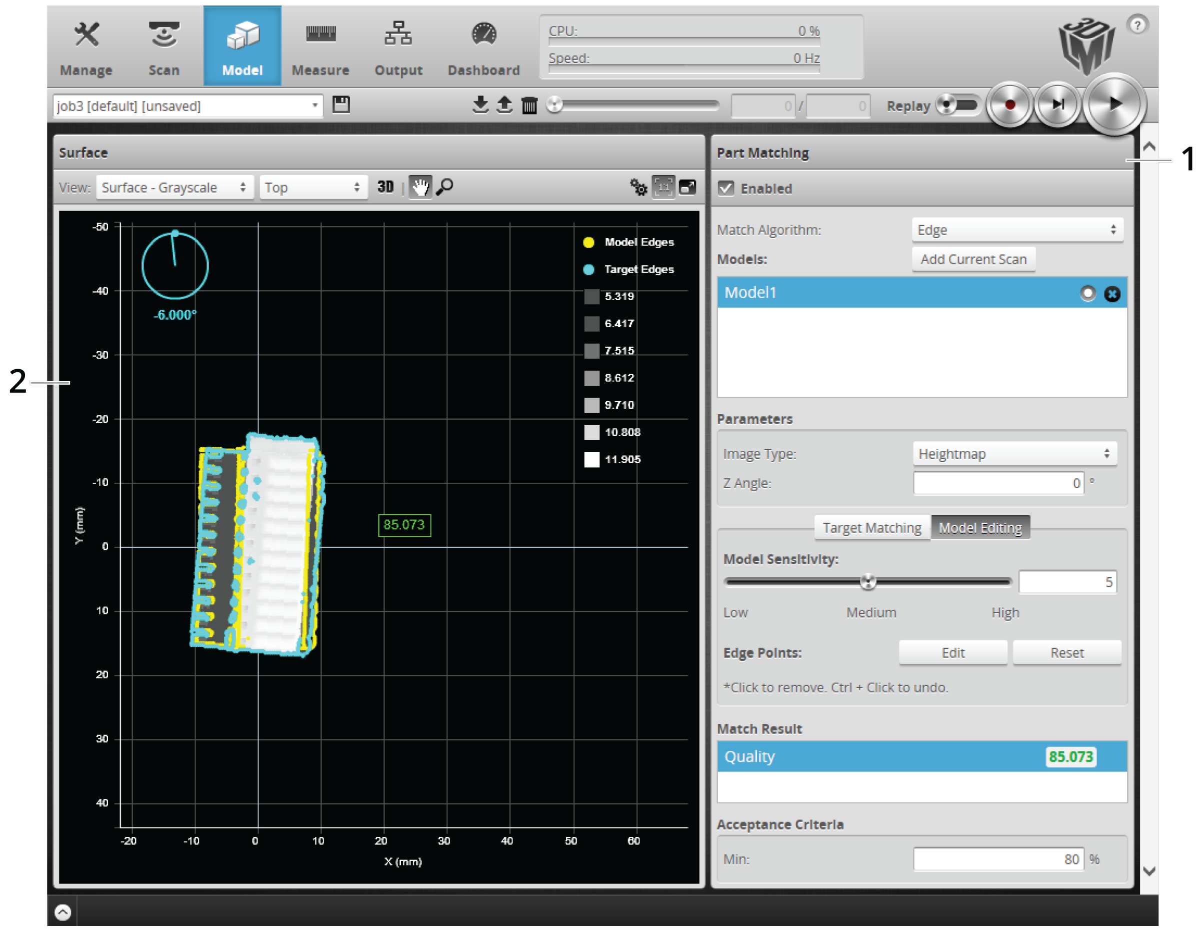
Models and Part Matching

The following sections describe how to set up part matching using a model, a bounding box, or an ellipse.

Model Page Overview

The **Model** page lets you set up part matching.

> The **Model** page is only displayed when the sensor is in **Surface** mode. See *Scan Modes* on page 54 for more information.



	Element	Description
1	Part Matching panel	Contains settings for configuring models and for part matching.
2	Data Viewer	Displays sensor data and lets you add and remove model edge points.

Part Matching

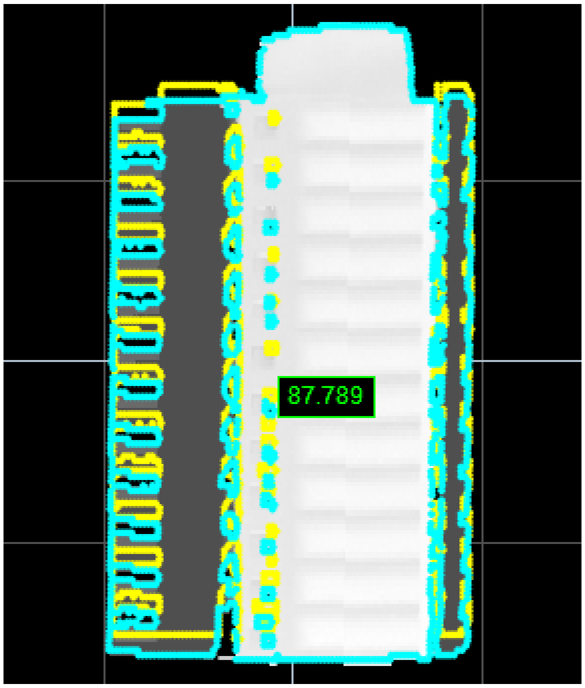
You can use Gocator to match parts to a previously saved model (see *Using Edge Detection* on page 84) or to the dimensions of a fitted bounding box or ellipse (see *Using Bounding Box and Ellipse* on page 93), regardless of the orientation of the part you are trying to match.

When the match quality between a model and a part reaches a minimum value (a percentage), or the bounding box or ellipse that encapsulates the part is between minimum and maximum dimension values, the part is "accepted" and any measurements that are added in the **Measure** page will return valid values, provided that the target is in range, etc. If the part is "rejected," any measurements added in the **Measure** page will return an Invalid value. For more information on measurements and decision values, see on page 97.

Using Edge Detection

When using edge detection for part matching, the Gocator compares a model that you must create from a previous scan to a "target" (one of the parts you want to match to the model).

In the data viewer, a model is represented as a yellow outline. The target is represented as a blue outline. If the part match quality above a minimum user-defined level, any measurements configured on the **Measure** page are applied.



Model (yellow outline) and target (blue outline).
Part match quality is 87.789%, which is greater
than the minimum set by the user, so the parts match.

When you create a model, the Gocator runs an edge detection algorithm on either the heightmap or intensity image of a scanned part. The resulting model is made up of the detected edge points. The scan used to create the model should be of a reference (or "golden") part to which all other parts will be compared.

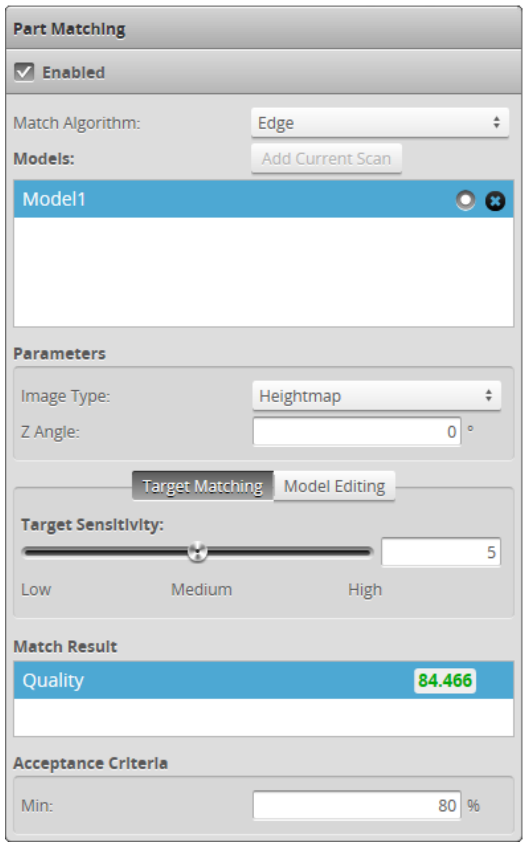
After the model has been created, you optionally modify the model by adjusting the sensitivity (how many edge points are detected), or selectively remove edge points from the model, to improve matching.

Models are saved as part of a job.

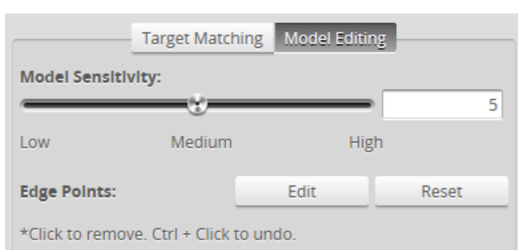
Once you have finished modifying the model, you can also modify target sensitivity, which controls how many edge points are detected on the subsequently scanned targets that will be compared to the model; the same edge detection algorithm used for creating models is used to compare a model to a part.

Typically, setting up edge detection to perform part matching involves the following steps:

1. Scan a reference part (you can also use replay data that you have previously saved).
2. Create a model based on the scan (using either heightmap or intensity data).
3. Adjust the model (edge detection algorithm sensitivity and selective removal of edge points).
4. Scan another part typical of the parts that would need to match the model.
5. Adjust the target sensitivity.
6. Set match acceptance level.



Part Matching panel showing Target Matching tab



Model Editing tab on Part Matching panel

The following settings are used to configure part matching using edge detection.

Setting	Description
Match Algorithm	Determines which algorithm the sensor will use to attempt a match. Set this to **Edge** for edge detection.
Image Type	Determines what kind of data the Gocator will use to detect edges and therefore for part matching. Choose this setting based on the kinds of features that will be used for part matching: **Heightmap**: Surface elevation information of the scanned part will be used to determine edges. This setting is most commonly used. **Intensity**: Intensity data (how light or dark areas of a scanned part are) will be used to determine edges. Use this setting if the main distinguishing marks are printed text or patterns on the parts. The **Acquire Intensity** option must be checked in the **Scan Mode** panel on the **Scan** page for this option to be available.
Z Angle	Corrects the orientation of the model to accurately match typical orientation and simplify measurements.
Target Sensitivity (**Target Matching** tab)	Controls the threshold at which an edge point is detected on the target's heightmap or intensity image. (The "target" is any part that is matched to the model and which will subsequently be measured if the match is accepted.) Setting **Target Sensitivity** higher results in more edge points. Setting it lower results in fewer edge points and results in higher performance. Use this setting to exclude noise from the detected edges and to make sure distinguishing features are properly detected. The level of this setting should generally be similar to the level of **Model Sensitivity**.
Model Sensitivity (**Model Editing** tab)	Controls the threshold at which an edge point is detected on the heightmap or intensity image used to create the model. Setting **Model Sensitivity** higher results in more edge points. Setting it lower results in fewer edge points and results in higher performance. Use this setting to exclude noise from the detected edges and to make sure distinguishing features are properly detected. The level of this setting should generally be similar to the level of **Target Sensitivity**. Changing this setting causes the edge detection algorithm to run again at the new threshold. If you have edited edge points manually (removing them selectively), those changes will be lost. See *Using Edge Detection* on page 84 for more information.
Edge Points (**Model Editing** tab)	The **Edit** button lets you selectively remove edge point that are detected by the edge detection algorithm at the current **Model Sensitivity** setting. See *Using Edge Detection* on page 84 for more information.
Acceptance Criteria	Determines the minimum quality level of the match as a percentage value.

Part rejected: Quality result is less than Min

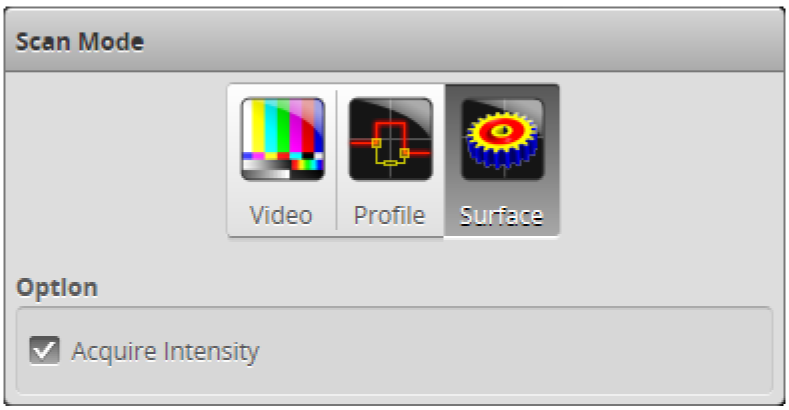
To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the Gocator is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the model), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Creating a Model

Gocator creates a model by running an edge detection algorithm on the heightmap or intensity image of a scan. The algorithm is run when a model is first created and whenever the **Model Sensitivity** setting is changed.

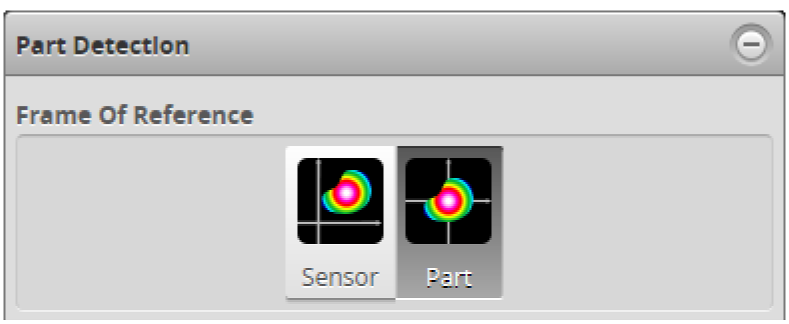
To create a model:

1. Go to the **Scan** page.
   a. In the **Scan Mode** panel, choose **Surface**.



   You must choose **Surface** in order to scan a part. Furthermore, the **Model** page is only displayed in Surface mode.

   b. If you want to use intensity data to create the model, make sure **Acquire Intensity** is checked.
   c. In the **Part Detection** panel, choose **Part** for the **Frame of Reference**.

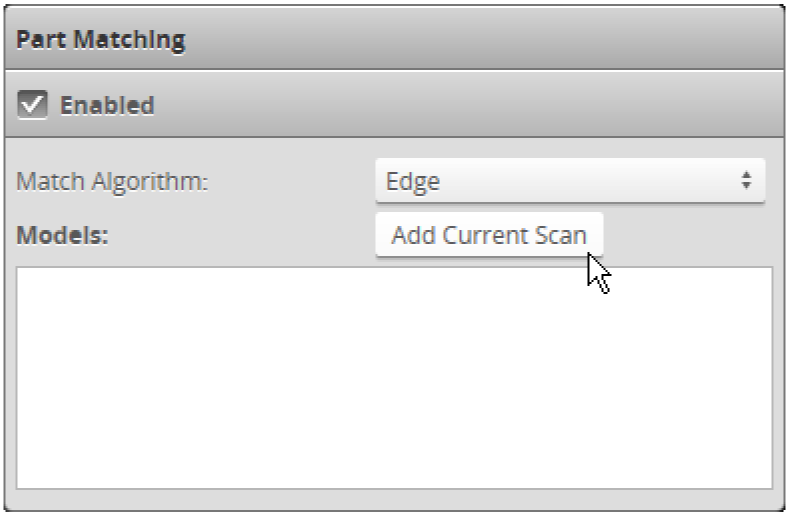


   Part matching is only available when **Part** has been selected.

2. Do one of the following:
   - Scan a reference part. See *Scan Setup and Alignment* on page 53 for more information on setting up and aligning Gocator. See *Running a Standalone Sensor System* on page 23 for more information on running a system to scan a part.

- Locate some previously recorded replay data and load it. See *Recording, Playback, and Measurement Simulation* on page 38 and *Downloading, Uploading, and Exporting Replay Data on page 40* for more information on replay data.

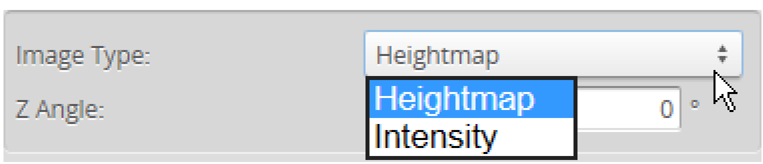
3. Go to the **Model** page.
   a. Make sure the **Enabled** option is checked in the **Part Matching** panel.
   b. In the **Match Algorithm** drop-down, choose **Edge**.



4. Click **Stop** on the toolbar if the sensor is running.
5. Click **Add Current Scan**.

> After adding the model, the Gocator will show that the match quality is 100%, because it is in effect comparing the model to the scan that was used to create the model. This value can be ignored.

6. In the **Image Type** drop-down, choose **Heightmap** or **Intensity**.



7. If you need to correct the orientation of the model, provide a value in the **Z Angle** field.

   Correcting the Z angle is useful if the orientation of the model is not close to the typical angle of target parts on the production line.

8. Save the job by clicking the **Save** button.

   Models are saved in job files.

   See *Saving and Loading Jobs (Settings)* on page 37 for more information on saving jobs.

After you have created a model, you may wish to modify it to remove noise to improve its matching capabilities. You may also wish to modify a model to exclude certain areas. See *Creating a Model* on the previous page for more information.

Model names can be renamed.

To rename a model:

1. In the **Models** list, double-click on a model name.
2. Type a new name in the model name field.
3. Press Enter or click outside the model name field.
4. Save the job by clicking the **Save** button.

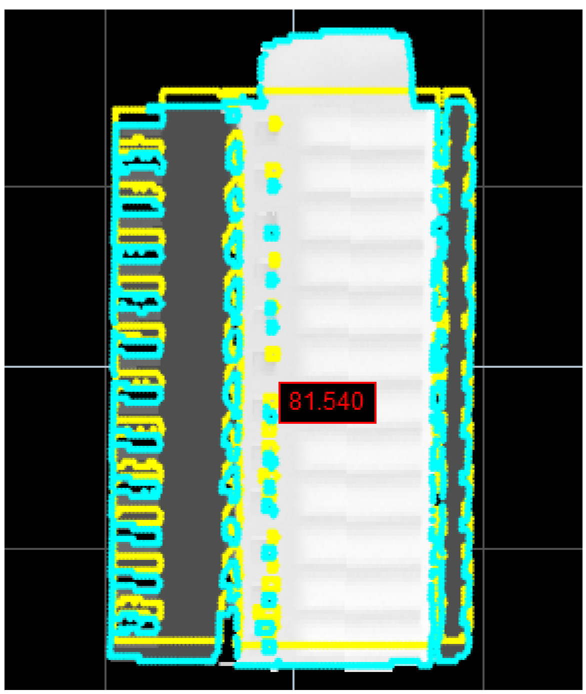
To delete a model, click the button.

Modifying a Model's Edge Points

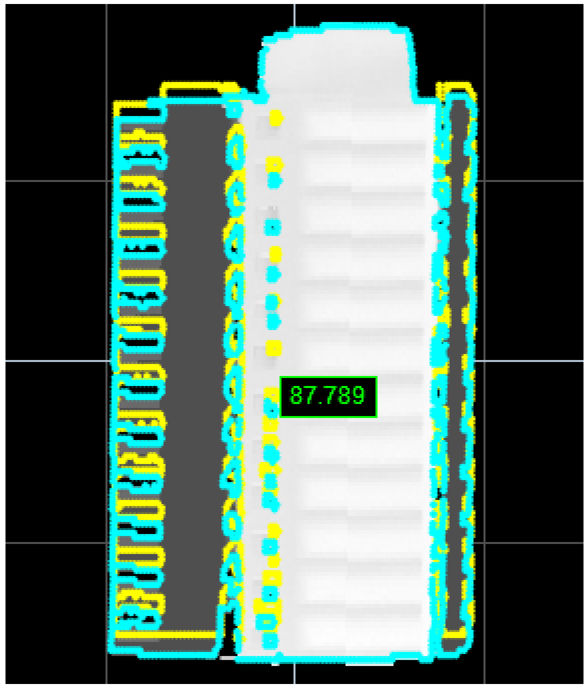
Modifying a model's edge points is useful to exclude noise in the detected edge points and to make sure distinguishing features are properly detected, which can improve matching. You can modify edge points in two ways.

First, you can control the overall number of edge points that are detected by the edge detection algorithm by raising and lowering the edge detection threshold (the **Model Sensitivity** setting). Modifying **Model Sensitivity** causes the edge detection algorithm to run again.

Second, you can fine-tune the model's edge points by selectively removing edge points that are detected by the edge detection algorithm. This could be useful, for example, if an edge on the target parts frequently presents minor variations such as flashing (excess material caused by leakage during molding): the edge points that make up the model can be edited to exclude that region. Editing the model can allow parts to match it more easily.



Edge points along top of model not removed.
Part is rejected. (Min set to 85%.)



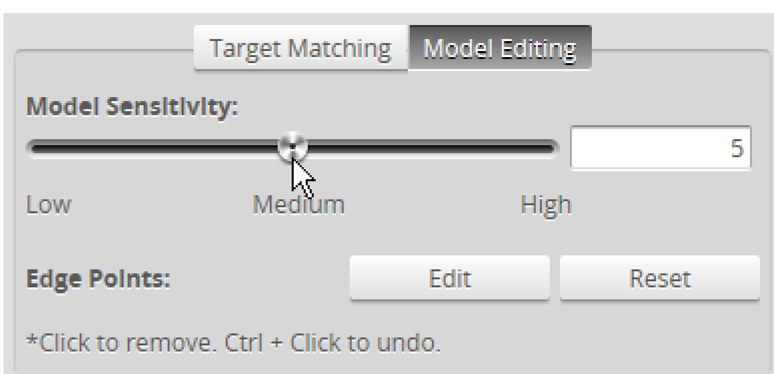
Edge points along top of model removed.
Part is accepted. (Min set to 85%.)

Removing edge points does not cause the edge detection algorithm to run again.

To change model senstivity:

1. In the **Models** list, select the model you want to configure by clicking on its selection control.

2. Click the **Model Editing** tab.
3. Adjust the **Model Sensitivity** slider to exclude noise and to properly detect the distinguishing features that will match parts.



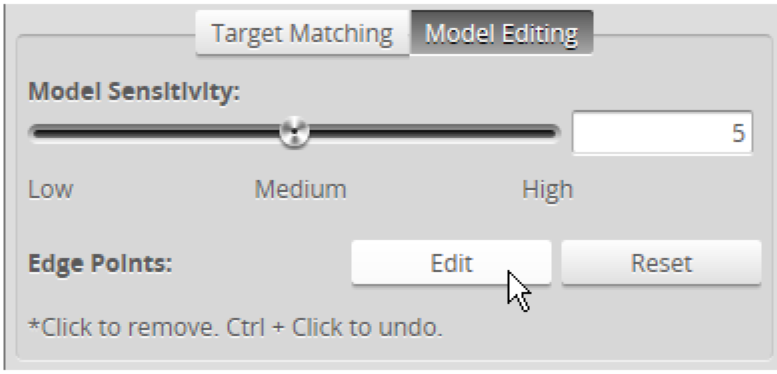
   You can also set the sensitivity value manually in the provided text box.

4. Save the job by clicking the **Save** button.

To manually remove model edge points:

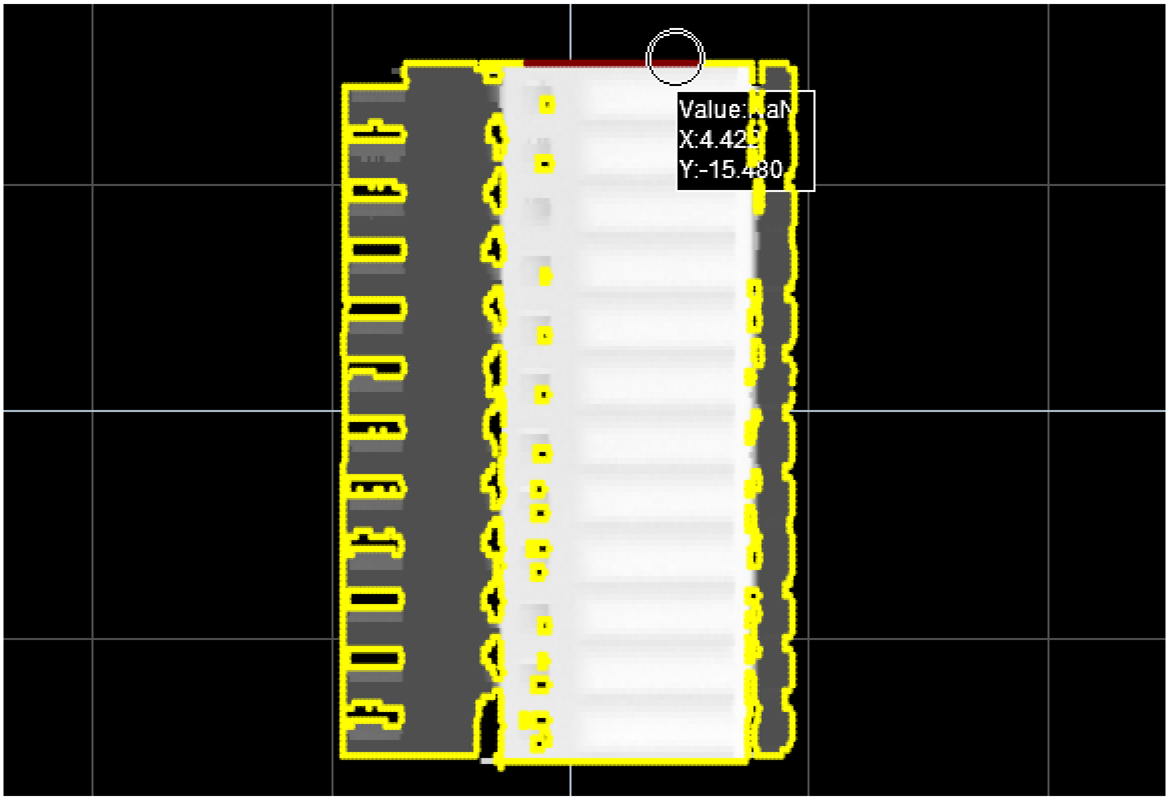
1. In the **Models** list, select the model you want to configure by clicking on its selection control.

2. In the **Model Editing** tab, click on the **Edit** button.



3. On the toolbar above the data viewer, make sure the **Select** tool is active.

4. Click in the data viewer and hold the mouse button while moving the pointer over the edge points you want to remove.



Points within the circular **Select** tool are removed from the model. Removed edge points turn red in the data viewer.

You can zoom in to see individual edge points by using the mouse wheel or by using the Zoom mode ( ).

5. If you have removed too many edge points, use Ctrl + Click in the data viewer to add the edge points back.
6. When you have finished editing the model, click **Save** in the **Model Editing** tab.
7. Save the job by clicking the **Save** button on the toolbar.

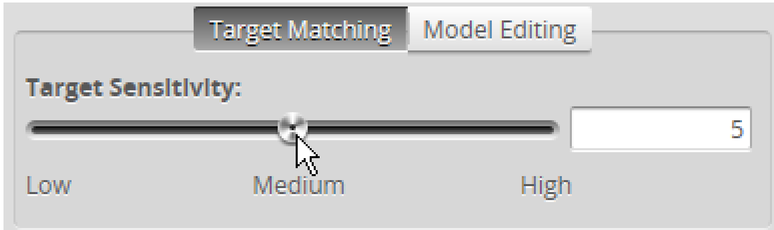
Adjusting Target Sensitivity

After you have added a model and optionally adjusted it, you must scan a different part, one that is typical of parts that must match the model.

Much in the same way that you can adjust a model's sensitivity, you can adjust the target sensitivity, that is, the threshold at which edge points are detected on the heightmaps or intensity images of parts that you want to match to the model. Adjusting the target sensitivity is useful to exclude noise, improving part matching.

To change target senstivity:

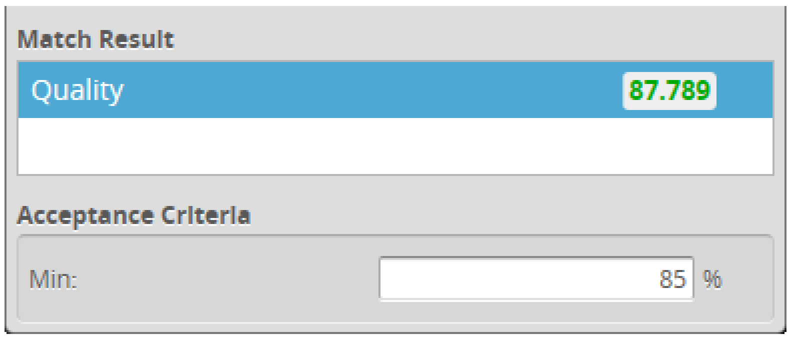
1. Click the **Target Matching** tab.
2. Adjust the **Target Sensitivity** setting to exclude noise in order to properly detect the distinguishing features that will allow parts to match.



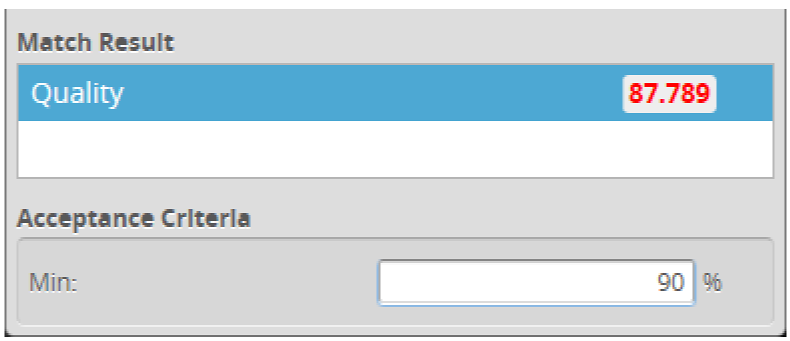
You can also set the sensitivity value manually in the provided text box.

Setting the Match Acceptance Criteria

In order for a part to match a model, the match quality must reach the minimum set in the **Min** field in **Acceptance Criteria** section of the **Part Matching** panel.



Part accepted: Quality result is greater than Min



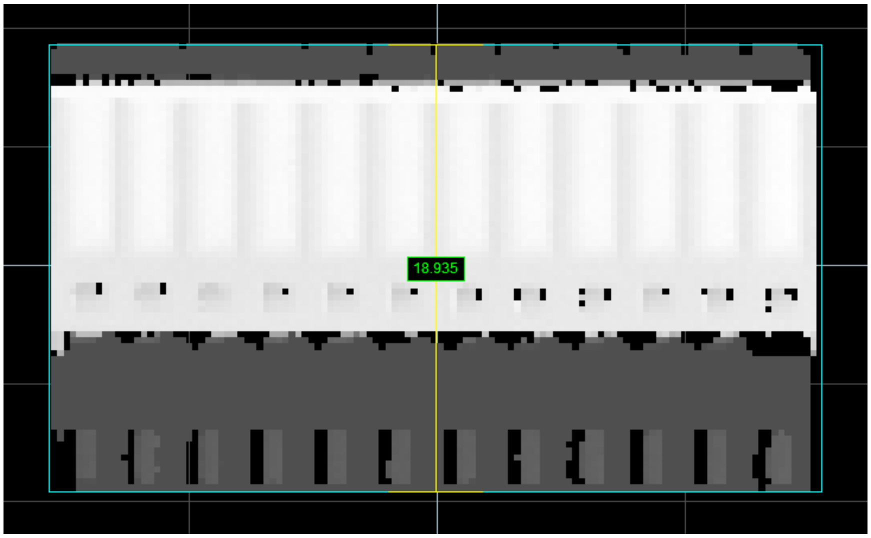
Running Part Matching

To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the Gocator is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the model), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Using Bounding Box and Ellipse

When using a bounding box or an ellipse to match parts, the Gocator tests whether a part fits into a bounding box or ellipse that you define. A match will occur regardless of orientation.

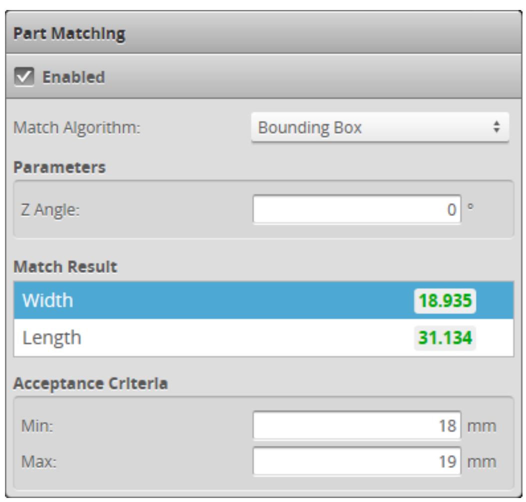
In the data viewer, a bounding box or ellipse is displayed with a blue outline. If a part fits in the bounding box or ellipse, any measurements configured on the **Measure** page are applied.



Blue bounding box around a part.
(Yellow lines show currently selected dimension in Part Matching panel.)

Typically, setting up a bounding box or an ellipse to perform part matching involves the following steps:

1. Scan a reference part (you can also use replay data that you have previously saved).
2. Set the characteristics of the bounding box (width and length) or ellipse (major and minor axes).



Part Matching panel (Bounding Box algorithm)

The following settings are used to configure part matching using a bounding box or ellipse.

Setting	Description
Match Algorithm	Determines which algorithm the sensor will use to attempt a match. Set this to **Bounding Box** or **Ellipse**.

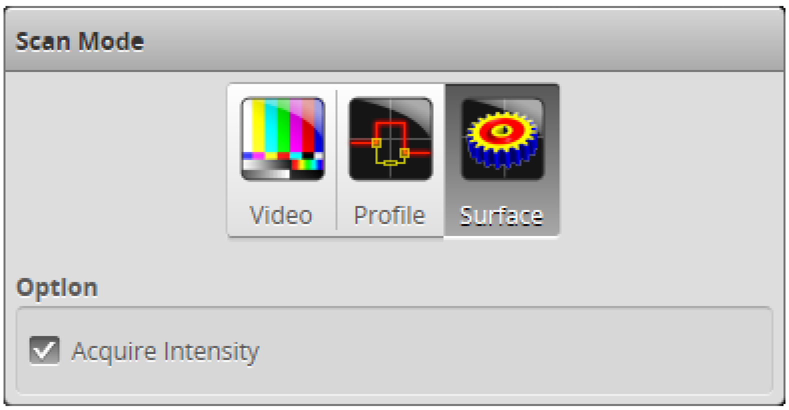
Setting	Description
Z Angle	Corrects the orientation of the bounding box or ellipse to accurately match typical orientation and simplify measurements.
Acceptance Criteria	Determines the minimum and maximum acceptable values of the selected dimension (Width and Length for bounding box, Major and Minor for ellipse) in **Match Result**.

Configuring a Bounding Box or an Ellipse

To use a bounding box or an ellipse to match a part, you must set its dimensions, taking into account expected acceptable variations when compared to a reference (or "golden") part.

To configure a bounding box or ellipse for part matching:

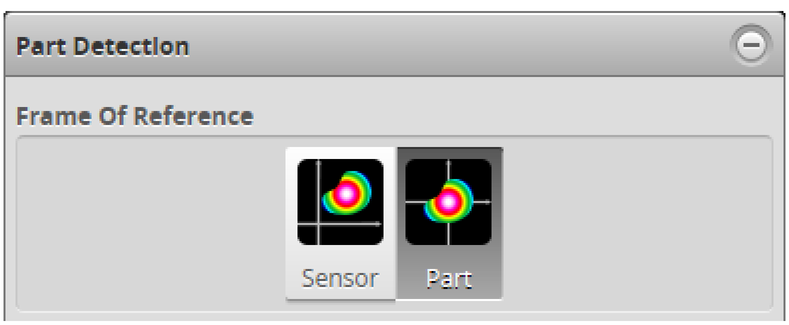
1. Go to the **Scan** page.
   a. In the **Scan Mode** panel, choose **Surface**.



   You must choose **Surface** in order to scan a part. Furthermore, the **Model** page is only displayed in Surface mode.

   Intensity data is not used when part matching using a bounding box or an ellipse, but you can enable the **Acquire Intensity** option if you need intensity data for other reasons.

   b. In the **Part Detection** panel, choose **Part** for the **Frame of Reference**.



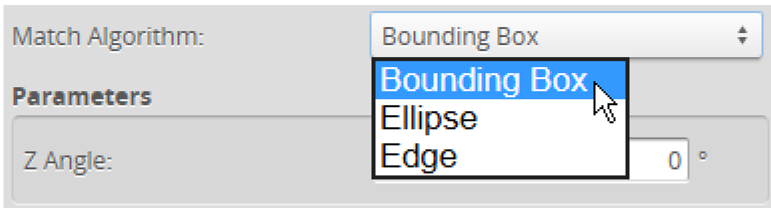
   Part matching is only available when **Part** has been selected.

2. Do one of the following:
   - Scan a reference part. See *Scan Setup and Alignment* on page 53 for more information on setting up and aligning Gocator. See *Running a Standalone Sensor System* on page 23 for more information on running a

system to scan a part.

- Locate some previously recorded replay data and load it. See *Recording, Playback, and Measurement Simulation* on page 38 and *Downloading, Uploading, and Exporting Replay Data on page 40* for more information on replay data.

3. Go to the **Model** page.
   a. Make sure the **Enabled** option is checked in the **Part Matching** panel.
   b. In the **Match Algorithm** drop-down, choose **Bounding Box** or **Ellipse**.



4. Set **Min** and **Max** of both of the dimensions of the selected match algorithm shape, taking into account expected acceptable variations.
   - If you chose **Bounding Box** for the match algorithm, select **Width** and then **Length** in **Match Result**, setting the minimum and maximum values acceptable for each dimension.
   - If you chose **Ellipse** for the match algorithm, select **Minor** and then **Major** in **Match Result**, setting the minimum and maximum values acceptable for each dimension.
5. Save the job by clicking the **Save** button.

   See *Saving and Loading Jobs (Settings)* on page 37 for more information on saving jobs.

Running Part Matching

To run part matching, simply make sure that the **Enabled** option is checked on the **Part Matching** panel when the Gocator is running. Any measurements that are added and configured on the **Measure** page will be applied to parts if a part match is accepted, regardless of the part's orientation (a successfully matched part is rotated to match orientation of the bounding box or ellipse), returning a value and decision (as long as the part is in range, etc.). If a part match is rejected, measurements will return an Invalid value.

Using Part Matching to Accept or Reject a Part

Part matching results only determine whether a measurement is applied to a part. Whether the measurement returns a pass or fail value—its decision—depends on whether the measurement's value is between the **Min** and **Max** values set for the measurement. This decision, in addition to the actual value, can in turn be used to control a PLC for example. The part matching "decision" itself is not passed to the Gocator output, but you can simulate this by setting up a measurement that will always pass if it is applied.

For example, you could set up a Position Z measurement, choosing Max Z as the feature type, and setting the **Min** and **Max** values to the measurement range of the sensor. This way, as long as a part matches and the target is in range, etc., the measurement will pass. This measurement decision, which is passed to the Gocator's output, could in turn be used to control a PLC.

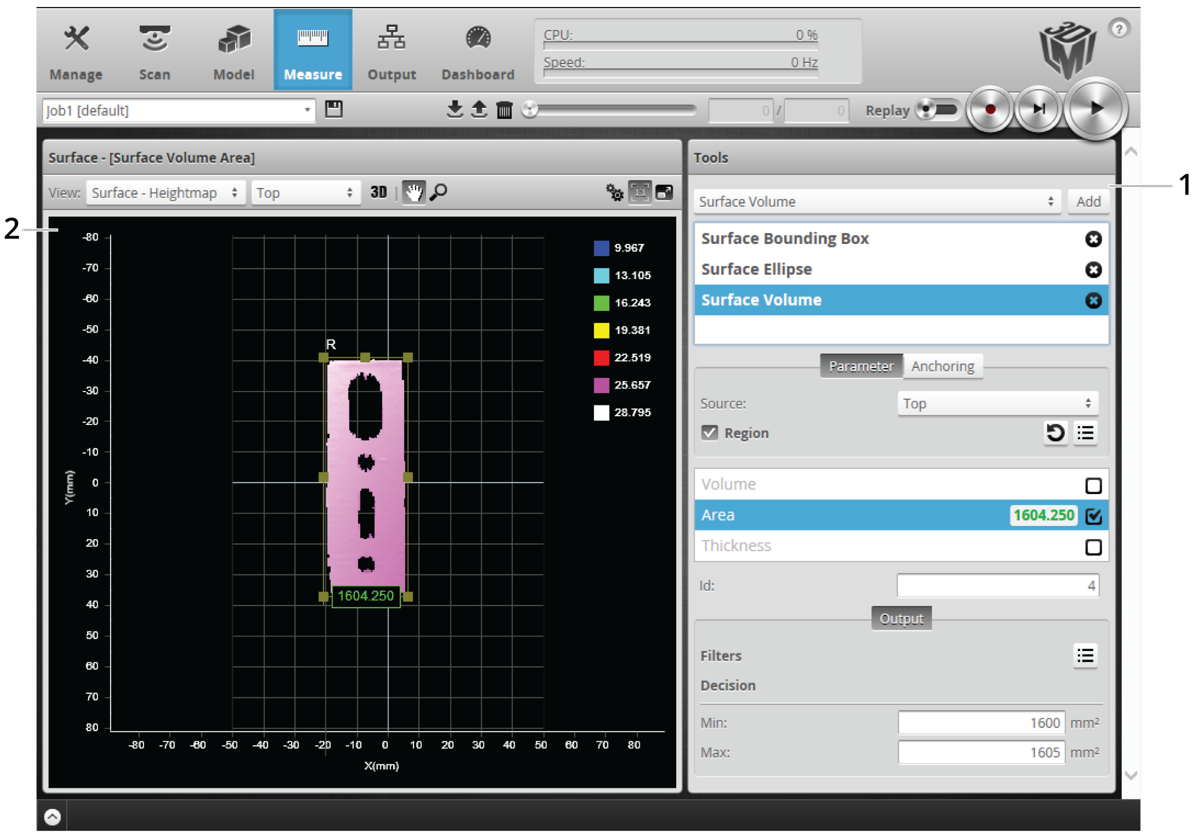
Measurement

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 Surface mode, the **Measure** page displays tools for surface measurement. In Video mode, tools are not accessible.



	Element	Description
1	Tools panel	Used to add, manage, and configure tools and measurements (see on the next page) and to choose anchors (see on page 104).
2	Data Viewer	Displays surface data, sets up tools, and displays result calipers related to the selected measurement. Parts are displayed using a height map, which is a top-down view of the XY plane, where color represents height. See *Data Viewer* on the next page.
3	Feature Area	Configurable region of interest from which feature points are detected. These feature points are used to calculate the measurements. The number of feature areas displayed depends on which measurement tool is currently selected.

Data Viewer

Regions, such as active area or measurement regions, can be graphically set up using the data viewer in the 2D or in the 3D view.

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 on page 101).

For instructions on how to set up measurement regions graphically, see on page 80.

Tools Panel

The **Tools** panel lets you add, configure, and manage tools. Tools contain related measurements. For example, the Position tool provides X, Y, and Z position measurements.

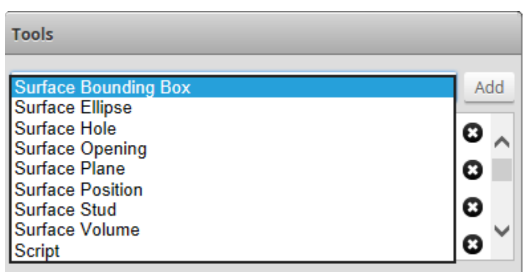
Some settings apply to tools, and therefore to all measurements, whereas some settings apply to specific measurements. See *Surface Measurement* on page 106 for information on the measurement tools and their settings.

> Tool names in the user interface include the scan mode, but not in the manual. So for example, you will see "Surface Bounding Box" in the user interface, but simply "Bounding Box" 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:

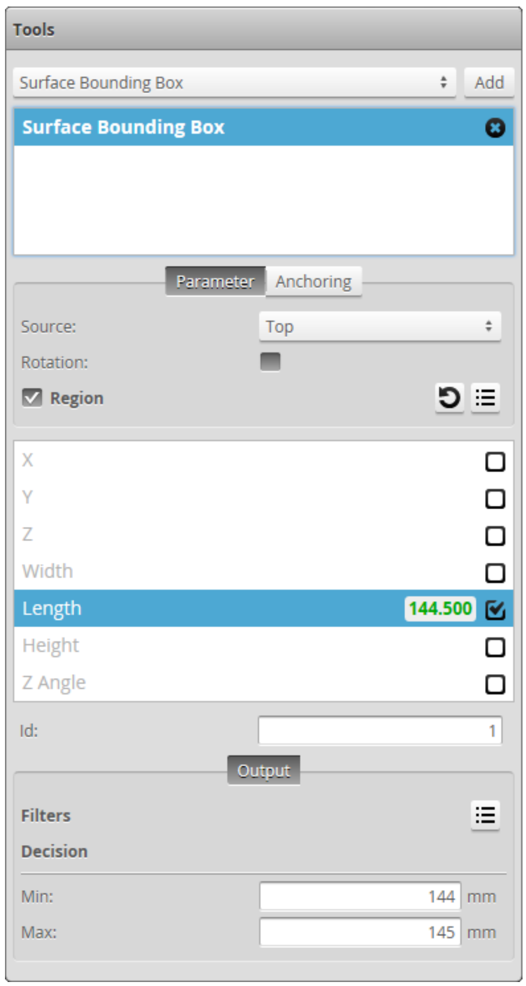
1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the Tools panel, select the tool 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 Surface 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, 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 Surface 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 Surface 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 Surface 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 Surface 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, 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 Surface 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* on page 99 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.

Many tools and measurements also have tool- and measurement-specific parameters. See the individual measurement tools for details.

Regions

Region parameters are used by many tools to limit the region in which a measurement will occur .

See the individual tools for details on using this parameter with each tool.

This parameter is also referred to as a measurement region.

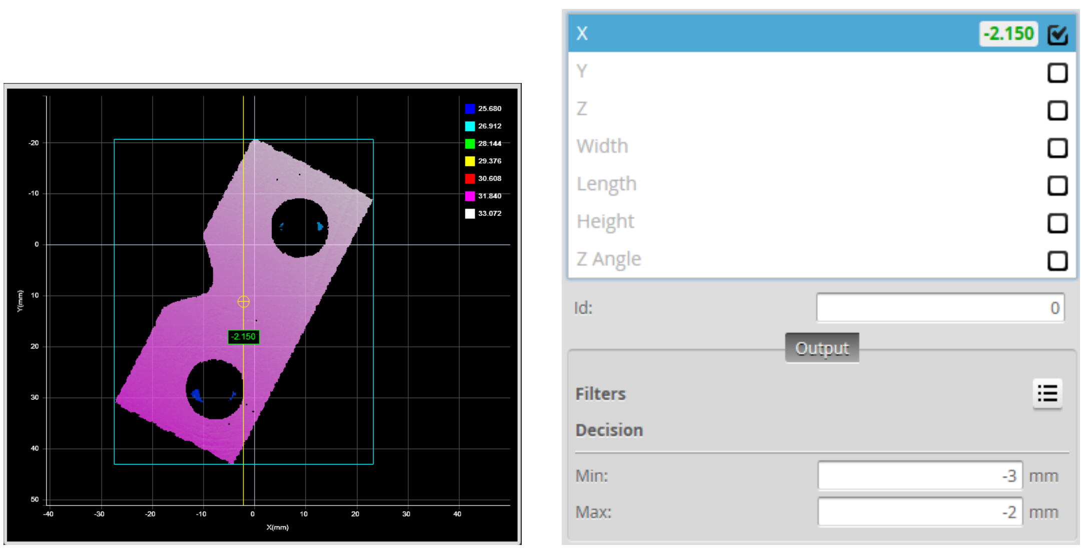
	In 3D mode, the region defaults to the global field of view.

To configure regions:

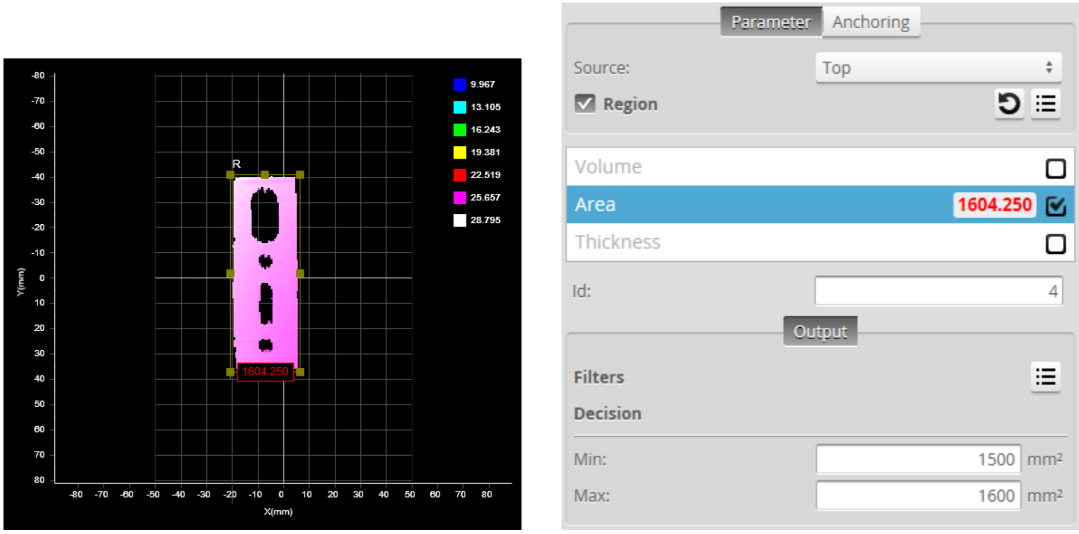
1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface mode in the **Scan Mode** panel.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the **Tools** panel, click on 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 or 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 (-2.150) within decision thresholds (Min: -3, Max: -2). Decision: Pass



Value (1604.250) outside decision thresholds (Min: 1500, Max: 1600). 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* on page 144 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 Surface 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* on page 99 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

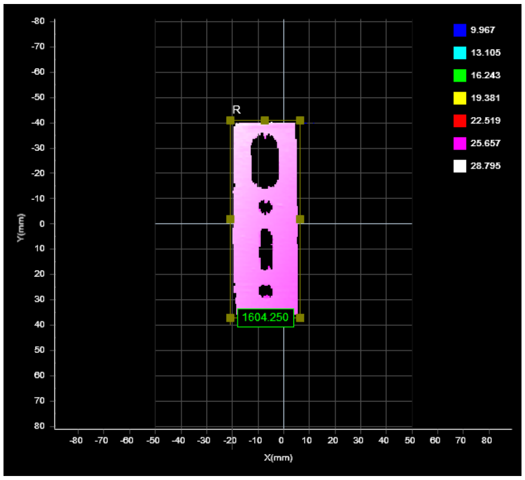
Filter	Description
	Scale and **Offset** can be used to transform the output without the need to write a script. For example, to convert the measurement value from millimeters to thousands of an inch, set **Scale** to 39.37. To convert from radius to diameter, set **Scale** to 2.
Hold Last Valid	Holds 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.

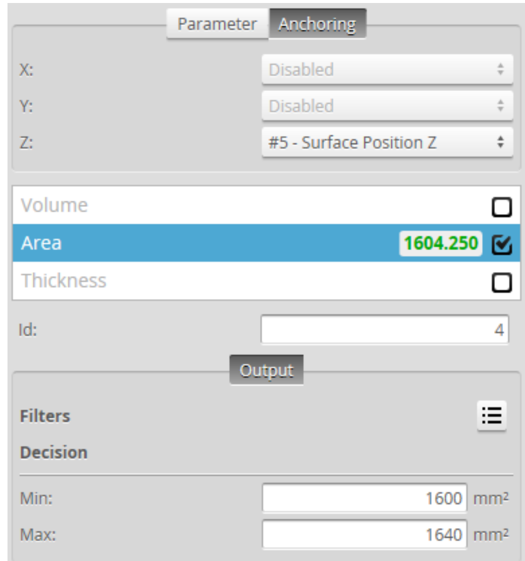
To configure the filters:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Surface 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* on page 99 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.

Measurement Anchoring

Measurement anchoring is used to track the movement of parts within the field of view of the sensor, compensating for variations in the height and position of parts. The movement is calculated as an offset from the position of a measured feature, where the offset is then used to correct the positions of measurement regions of other measurement tools. This ensures that the regions used to measure features are correctly positioned for every part.





Anchoring is not required in order to use measurement tools. This is an optional feature that helps make measurements more robust when the position and the height of the target varies from target to target.

Any X, Y, or Z measurement can be used as an anchor for a tool.

Several anchors can be created to run in parallel. For example, you could anchor some measurements relative to the left edge of a target at the same time as some other measurements are anchored relative to the right edge of a target.

To anchor a surface tool to a measurement:

1. Put a representative target object in the field of view.
   The target should be similar to the objects that will be measured later.
   a. Start the sensor, scan the target and then stop the sensor.
2. On the **Scan** page, in the **Scan Mode** panel, choose Surface mode.
   If this mode is not selected, tools will not be available in the **Measure** panel.
3. On the **Measure** page, add a suitable tool to act as an anchor.
   A suitable tool is one that returns an X, Y, or Z position as a measurement value.
4. Adjust the anchor tool's settings and measurement region, and choose a feature type (if applicable).
   You can adjust the measurement region graphically in the data viewer or manually by expanding the **Regions** area.
   The position and size of the anchor tool’s measurement regions define the zone within which movement will be tracked.
5. Add the tool that will be anchored.

Any tool can be anchored.

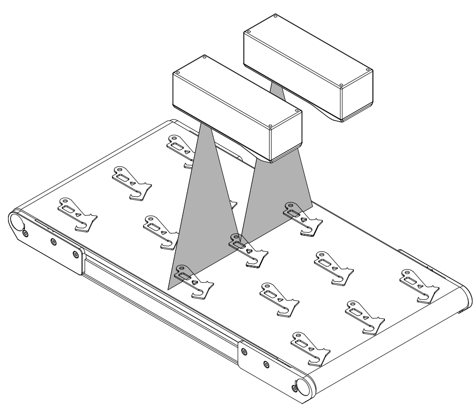
6. Adjust the tool and measurement settings, as well as the measurement regions.
7. Click on the tool's **Anchoring** tab.
8. Choose an anchor from the X, Y, or Z drop-down box.
   When you choose an anchor, a white “bulls-eye” indicator shows the position of the anchor in the data viewer.
   If the sensor is running, the anchored tool’s measurement regions are shown in white to indicate the regions are locked to the anchor. The measurement regions of anchored tools cannot be adjusted.
   The anchored tool’s measurement regions are now tracked and will move with the target’s position under the sensor, as long as the anchor measurement produces a valid measurement value. If the anchor measurement is invalid, for example, if there is no target under the sensor, the anchored tool will not show the measurement regions at all and an “Invalid-Anchor” message will be displayed in the tool panel.

To remove an anchor from a tool:

1. Click on the anchored tool's Anchoring tab.
   Select **Disabled** in the X, Y, or Z drop-down.

Surface Measurement

Surface measurement involves capturing 3D point cloud data, optionally identifying discrete objects, and measuring properties of the surface or the objects, such as the volume of the object or the height at a certain position of the object. All volumetric tools have the ability to operate either on the entire surface or the full object, or within a region of interest at a certain position in relation to the surface or an object.



Multiple measurements can be performed on the entire surface or each discrete object, limited only by the available CPU resources.

Measurement Tools

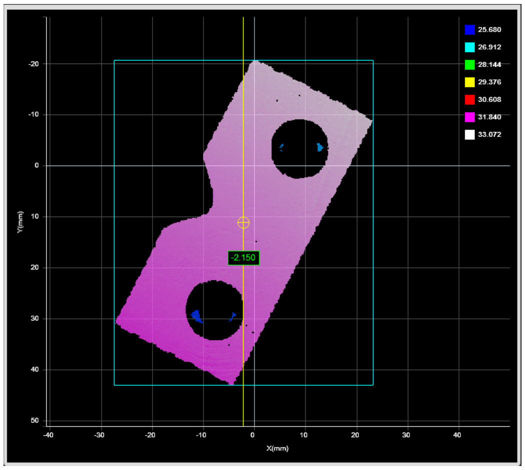
Bounding Box

The Bounding Box tool provides measurements related to the smallest rectangle box that encapsulates the part (for example, X position, Y position, width, length, etc.).

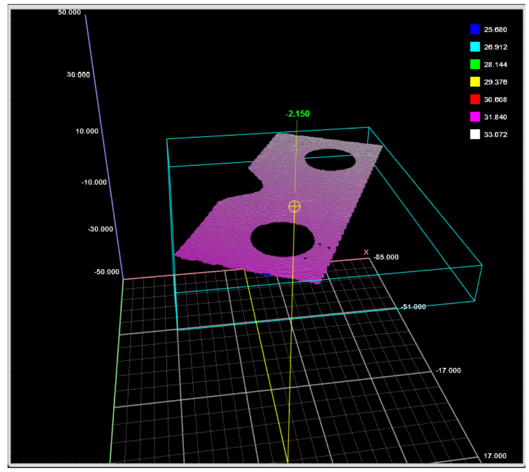
The measurement value can be compared with minimum and maximum constraints to yield a decision.

See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

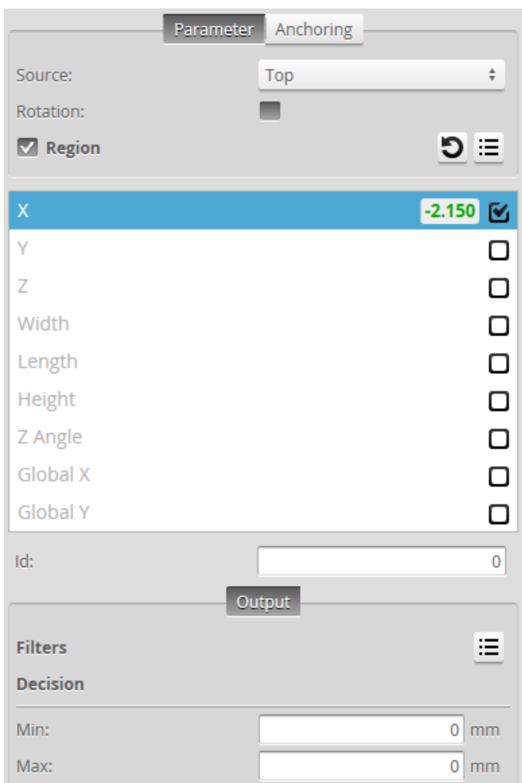
A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the Position centroids tools are referenced.



2D View

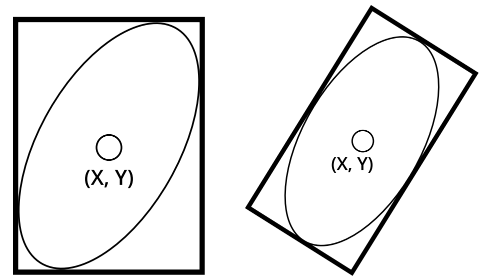
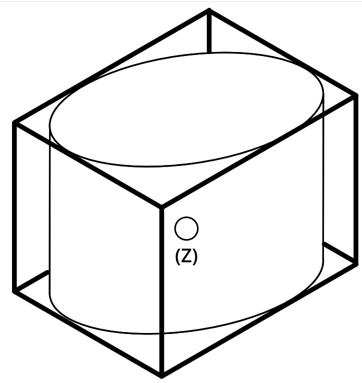
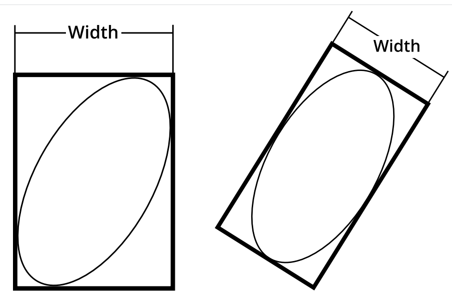
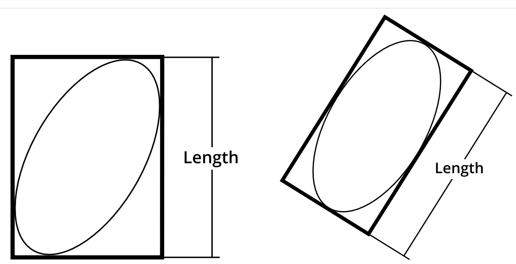


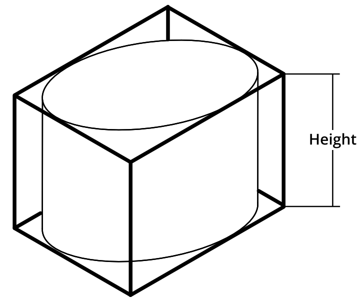
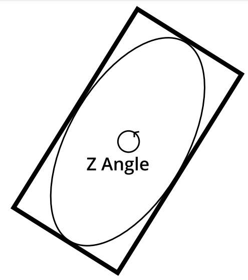
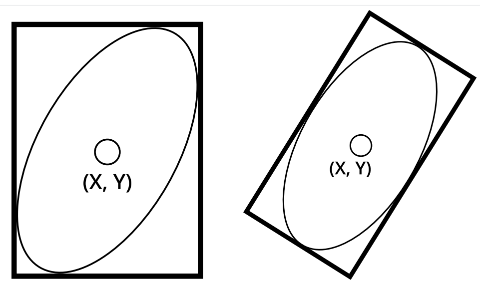
3D View



Measurement Panel

Measurements

<table>
<tr><th>Measurement</th><th>Illustration</th></tr>
<tr><td>X
Determines the X position of the center of the smallest rectangle that encapsulates the part.
The value returned is relative to the part.</td><td rowspan="2">
</td></tr>
<tr><td>Y
Determines the Y position of the center of the smallest rectangle that encapsulates the part.
The value returned is relative to the part.</td></tr>
<tr><td>Z
Determines the Z position of the center of the smallest rectangle that encapsulates the part.
The value returned is relative to the part.</td><td>
</td></tr>
<tr><td>Width
Determines the width of the smallest rectangle box that encapsulates the part. The width reports the dimension of the box in the direction of the minor axis.
When rotation is enabled, the bounding box is rotated by the angle of an ellipse fitted to the part data.</td><td>
</td></tr>
<tr><td>Length
Determines the length of the smallest rectangle box that encapsulates the part. The length reports the dimension of the box in the direction of the major axis.
When rotation is enabled, the bounding box is rotated by the angle of an ellipse fitted to the part data.</td><td>
</td></tr>
</table>

Measurement	Illustration
Height Determines the height (thickness) of the smallest rectangle box that encapsulates the part.	
Z Angle Determines the rotation about the Z axis and the angle of the bounding box relative to the X axis.	
Global X Determines the X position of the center of the smallest rectangle that encapsulates the part. The value returned is relative to the global/sensor coordinates.	
Global Y Determines the Y position of the center of the smallest rectangle that encapsulates the part. The value returned is relative to the global/sensor coordinates.	

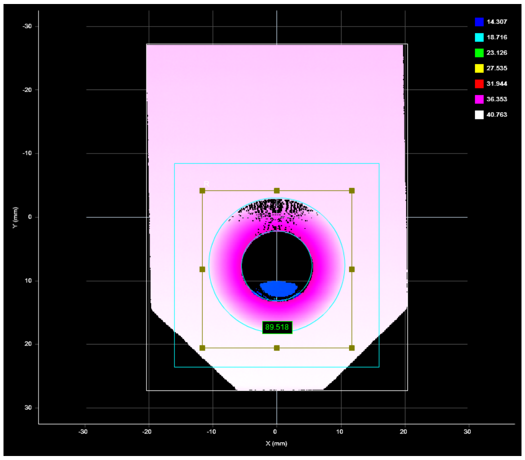
Parameters

Parameter	Description
Rotation	A bounding box can be vertical or rotated. A vertical bounding box provides the absolute position from which the part's Position centroid measurements are referenced. Check the **Rotation** checkbox to select rotated bounding box.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

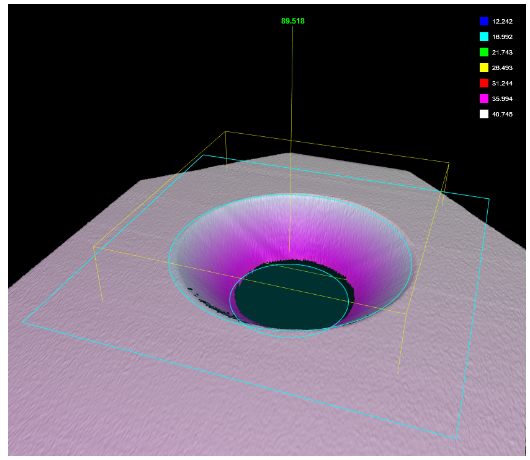
Countersunk Hole

The Countersunk Hole tool locates a countersunk circular opening within a region of interest on the surface and provides measurements to evaluate characteristics of countersunk holes, including the position (X, Y, and Z) of the center of the hole, outside radius of the hole, hole bevel angle, and the depth of the hole. The countersunk hole can be on a surface at an angle to the sensor.

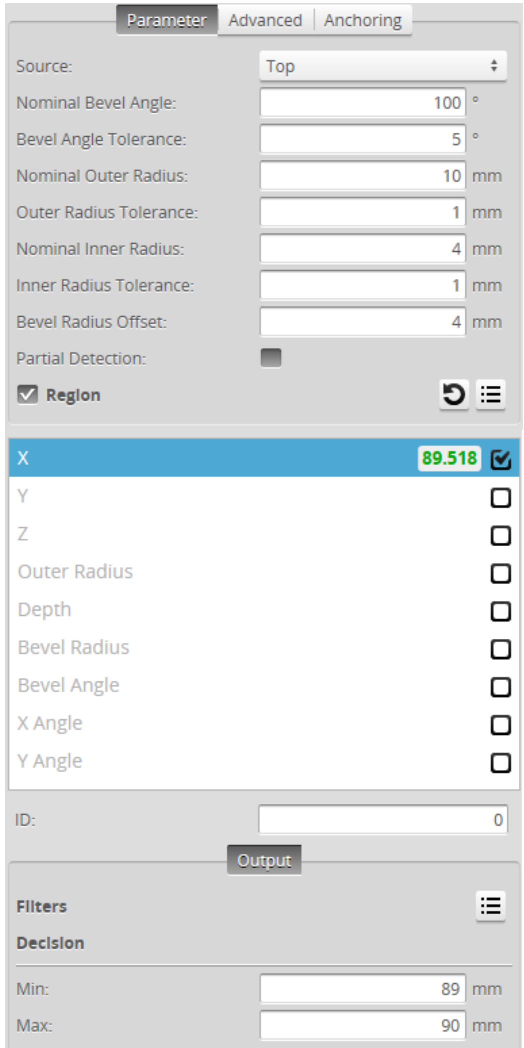
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

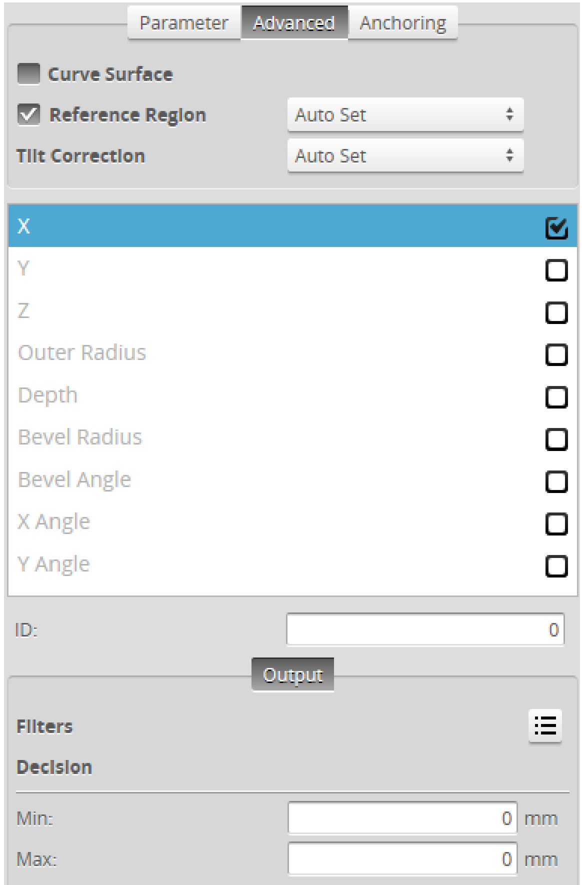


2D View

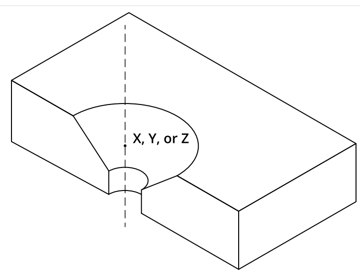


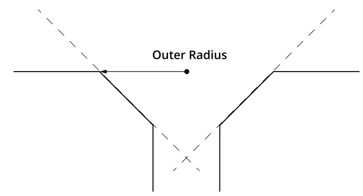
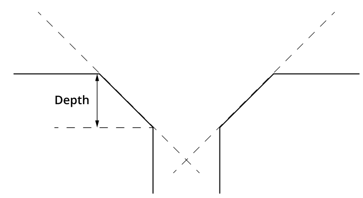
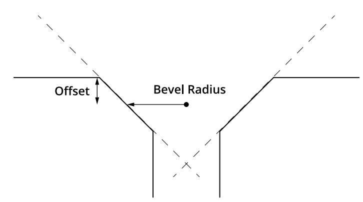
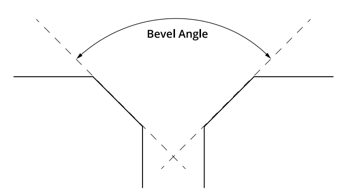
3D View

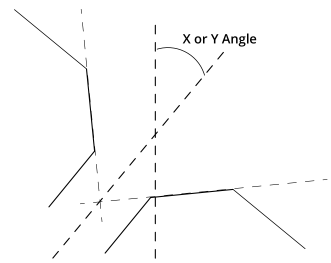

Parameter
Advanced
Anchoring
Source:
Top
Nominal Bevel Angle:
100 °
Bevel Angle Tolerance:
5 °
Nominal Outer Radius:
10 mm
Outer Radius Tolerance:
1 mm
Nominal Inner Radius:
4 mm
Inner Radius Tolerance:
1 mm
Bevel Radius Offset:
4 mm
Partial Detection:
Region
X
89.518
Y
Z
Outer Radius
Depth
Bevel Radius
Bevel Angle
X Angle
Y Angle
ID:
0
Output
Filters
Decision
Min:
89 mm
Max:
90 mm



Measurements

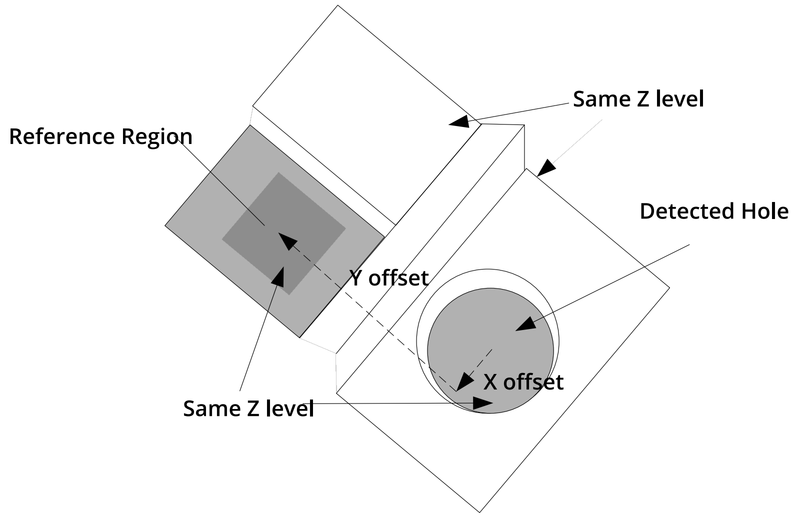
Measurement	Illustration
X Determines the X position of the center of the countersunk hole.	
Y Determines the Y position of the center of the countersunk hole.	
Z Determines the Z position of the center of the countersunk hole.	

Measurement	Illustration
Outer Radius Determines the outer radius of the countersunk hole. To convert the radius to a diameter, set the **Scale** setting in the **Output** panel (displayed after expanding the **Filters** section) to 2.	
Depth Determines the depth of the countersunk hole relative to the surface that the countersunk hole is on.	
Bevel Radius Determines the radius at a user-defined offset (**Offset** setting) relative to the surface that the countersunk hole is on. To convert the radius to a diameter, set the **Scale** setting in the **Output** panel (displayed after expanding the **Filters** section) to 2.	
Bevel Angle Determines the angle of the hole's bevel.	

Measurement	Illustration
X Angle Determines the angle the hole relative to the X axis. The measurement assumes that the hole is perpendicular to the surface, even though the surface itself is tilted.	
Y Angle Determines the angle of the hole relative to the Y axis. The measurement assumes that the hole is perpendicular to the surface, even though the surface itself is tilted.	

Parameters

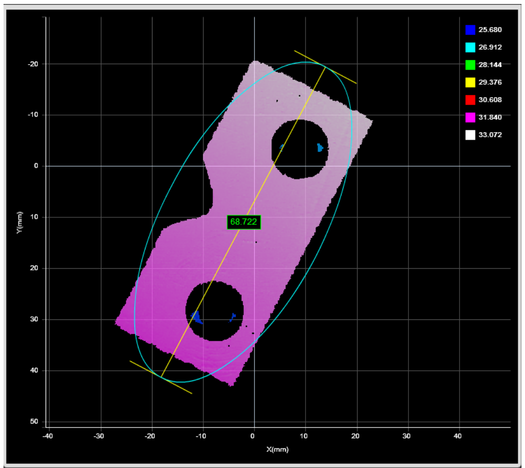
Parameter	Description
Nominal Bevel Angle	The expected bevel angle of the countersunk hole.
Bevel Angle Tolerance	The maximum variation from the nominal bevel angle (+/- from the nominal bevel angle).
Nominal Outer Radius	The expected outer radius of the countersunk hole.
Outer Radius Tolerance	The maximum variation from the nominal outer radius (+/- from the nominal outer radius).
Nominal Inner Radius	The expected inner radius of the countersunk hole.
Inner Radius Tolerance	The maximum variation from the nominal inner radius (+/- from the nominal inner radius).
Bevel Radius Offset	The offset, relative to the surface that the countersunk hole is on, at which the bevel radius will be measured.
Curve Surface	Whether the surface that the countersunk hole is on is curved. When enabled, specify the radius of the curvature in the **Curve Orientation** setting.
Reference Regions	The algorithm uses the Reference Regions option to calculate the Z position of the hole. It is typically used in cases where the surface around the hole is not flat.

Parameter	Description
	 When this option is set to Autoset, the algorithm automatically determines the reference region. When the option is not set to Autoset, the user manually specifies the reference region. The location of the reference region is relative to the detected center of the hole and positioned on the nominal surface plane. When the Reference Regions option is disabled, the tool measures the hole's Z position using all the data in the measurement region, except for a bounding rectangular region around the hole.
Tilt Correction	Tilt of the target with respect to the alignment plane. When this option is set to Autoset, the tool automatically detects the tilt. Otherwise, the user must enter the angles manually. Autoset requires the measurement region to cover more areas on the surface plane than other planes. The results from the Plane X and Y tool can be used for angles X and Y parameters.
Partial Detection	Enable if only part of the hole is within the measurement region. If disabled, the hole must be completely in the region of interest for results to be valid.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

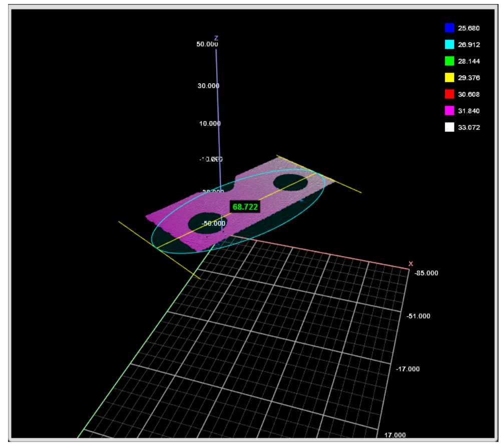
Ellipse

The Ellipse tool provides measurements for the major and minor axis lengths of an ellipse fitted to the part's shape in the XY plane, and also for the ratio of the major and minor axis lengths and for the orientation angle of the ellipse. The measurement value can be compared with minimum and maximum constraints to yield a decision.

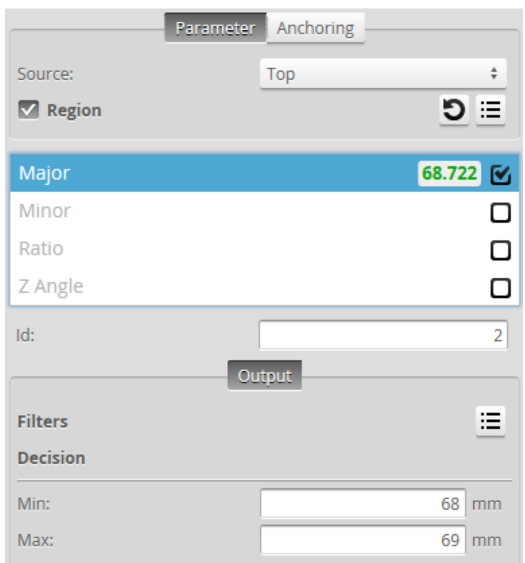
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.



2D View

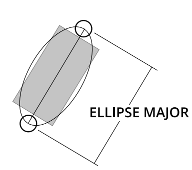


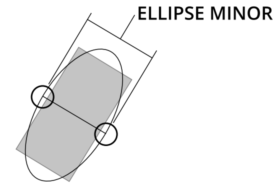
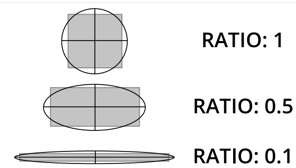
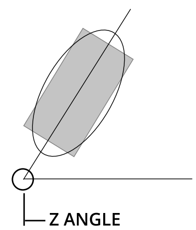
3D View



Measurement Panel

Measurements

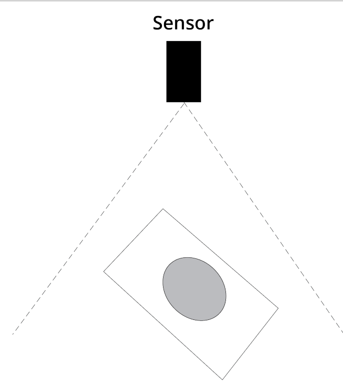
Measurement	Illustration
Major Determines the major axis length of an ellipse fitted to the part's area in the XY plane.	

Measurement	Illustration
Minor Determines the minor axis length of an ellipse fitted to the part's area in the XY plane.	
Ratio Determines the minor/major axis ratio of an ellipse fitted to the part's area in the XY plane.	
Z Angle Determines the orientation angle of an ellipse fitted to the part's area in the XY plane.	

Parameters

Parameter	Description
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

Hole



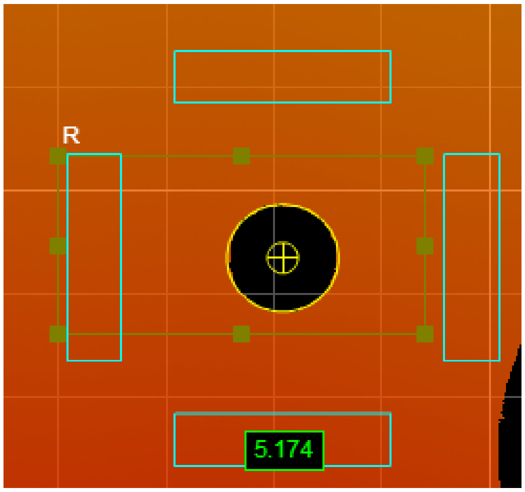
The Hole tool locates a circular opening within a region of interest on the surface and returns its position and radius.

See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

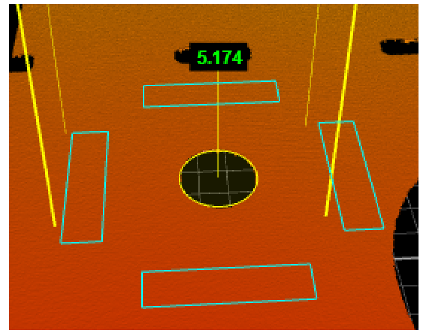
The hole can be on a surface at an angle to the sensor.

The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Hole Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

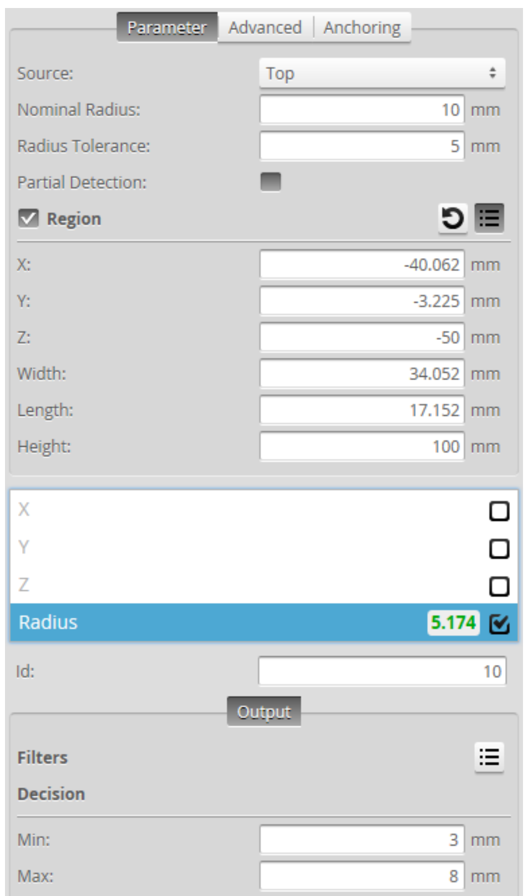
The measurement value can be compared with minimum and maximum constraints to yield a decision.

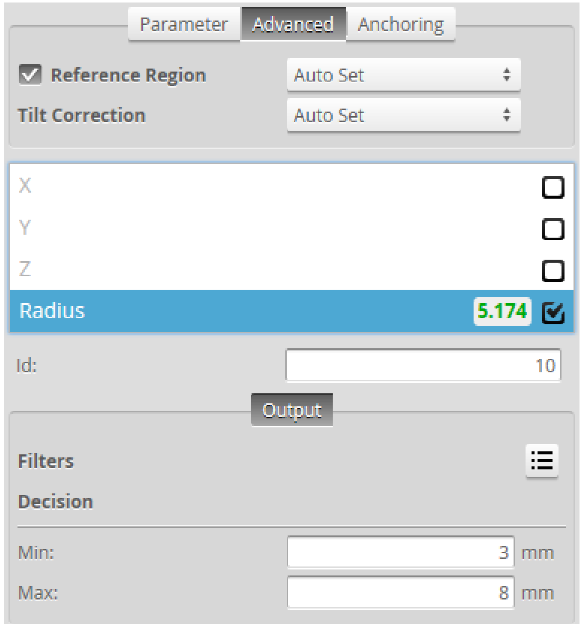


2D View



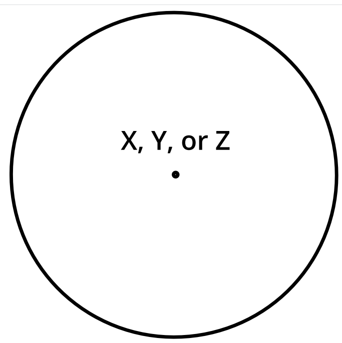
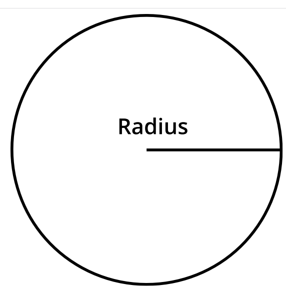
3D View





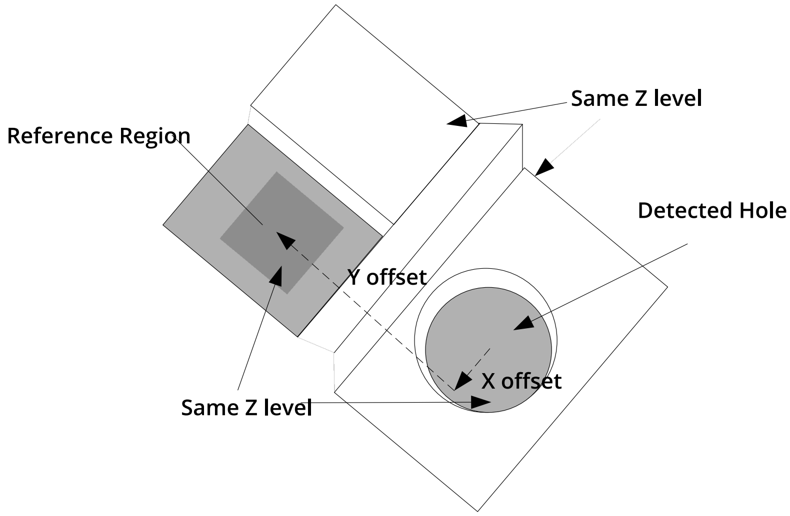
Measurement Panel

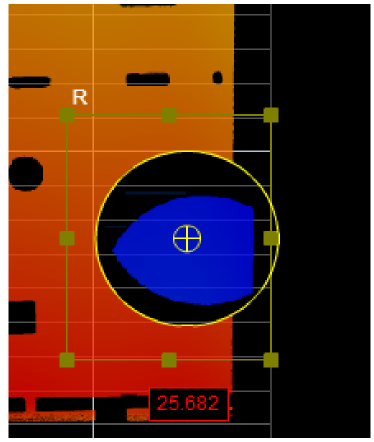
Measurements

Measurement	Illustration
X Determines the X position of the hole center.	
Y Determines the Y position of the hole center.	
Z Determines the Z position of the hole center.	
Radius Determines the radius of the hole.	

Parameters

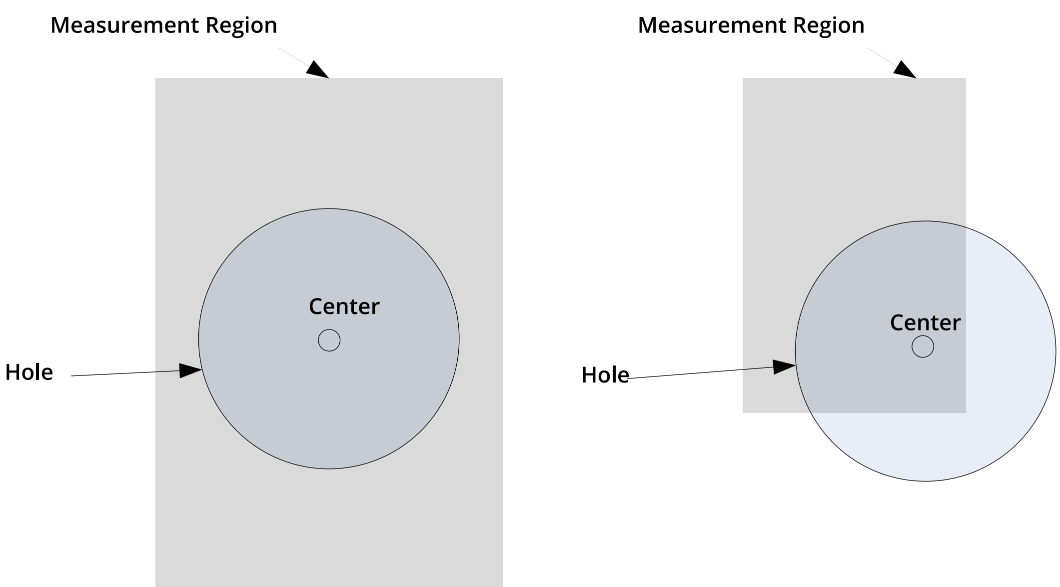
Parameter	Description
Nominal Radius	Expected radius of the hole.
Radius Tolerance	The maximum variation from the nominal radius (+/- from the nominal radius).

Parameter	Description
Reference Regions	The algorithm uses the Reference Regions option to calculate the Z position of the hole. It is typically used in cases where the surface around the hole is not flat.  When this option is set to Autoset, the algorithm automatically determines the reference region. When the option is not set to Autoset, the user manually specifies the reference region. The location of the reference region is relative to the detected center of the hole and positioned on the nominal surface plane. When the Reference Regions option is disabled, the tool measures the hole's Z position using all the data in the measurement region, except for a bounding rectangular region around the hole.
Tilt Correction	Tilt of the target with respect to the alignment plane. When this option is set to Autoset, the tool automatically detects the tilt. Otherwise, the user must enter the angles manually. Autoset requires the measurement region to cover more areas on the surface plane than other planes. The results from the Plane X and Y tool can be used for angles X and Y parameters.
Partial Detection	Enable if only part of the hole is within the measurement region. If disabled, the hole must be completely in the region of interest for results to be valid.

Parameter	Description
	
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

Measurement Region

The center of the hole must be inside the measurement region, even if the Partial Detection option is enabled.

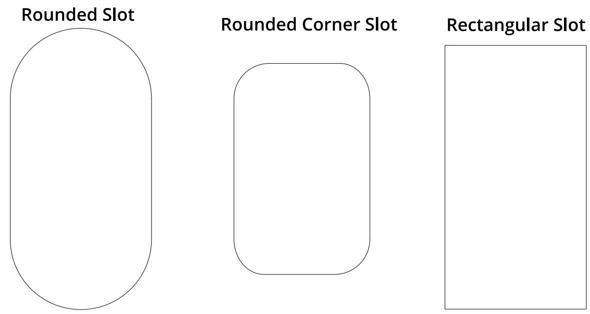


Opening

The Opening tool locates rounded, rectangular, and rounded corner openings. The opening can be on a surface at an angle to the sensor.

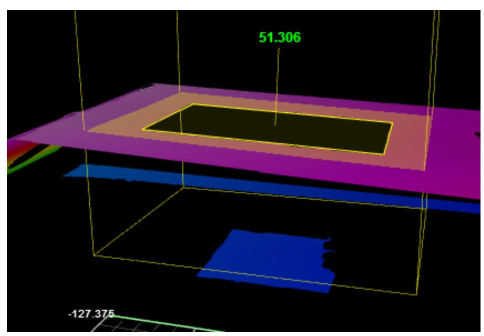
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Opening Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.



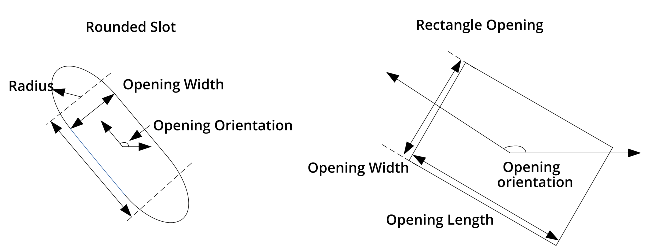
The measurement value can be compared with minimum and maximum constraints to yield a decision. You can select the measurement region in which the opening is expected to appear.

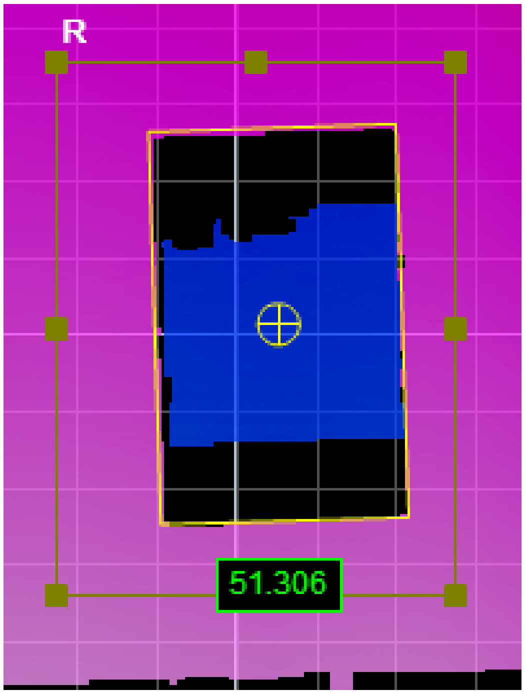
The algorithm can separate out background information that appears inside the opening. It can also detect a slot that only partially appears in the data.



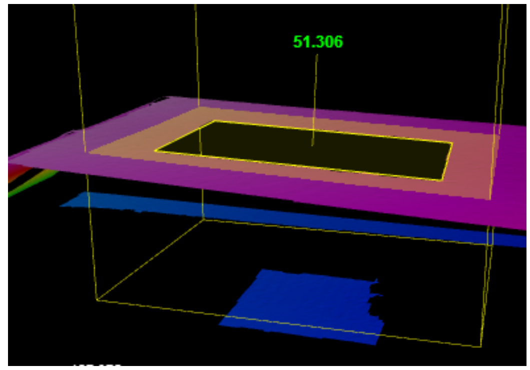
The shape of the opening is defined by its type and its nominal width, length, and radius.

The orientation defines the rotation around the normal of the alignment plane.

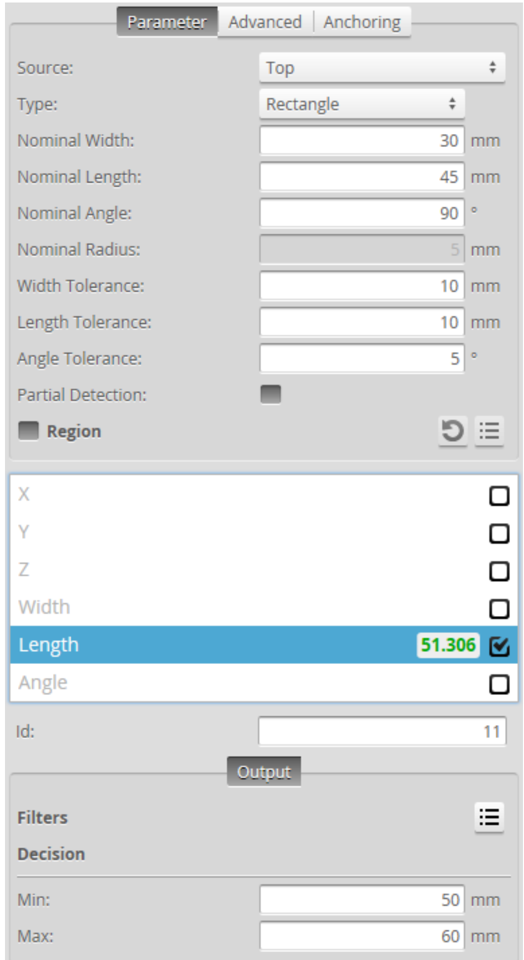


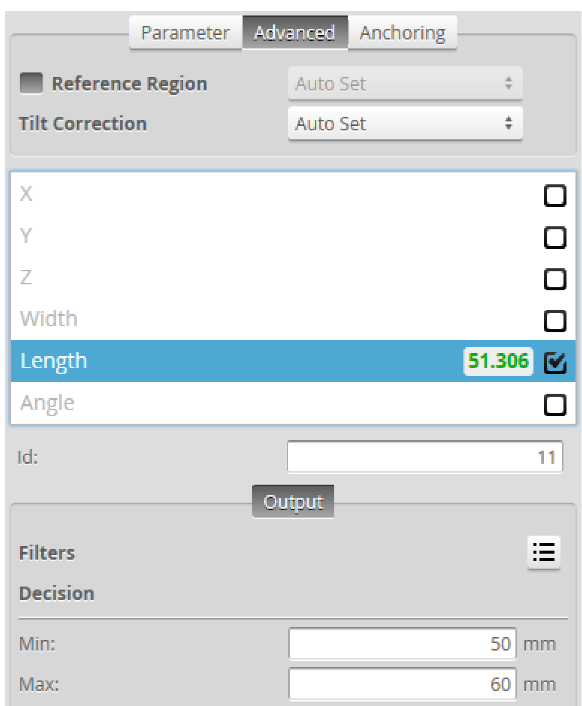


2D View

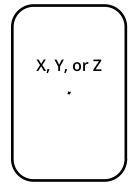
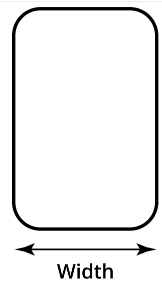
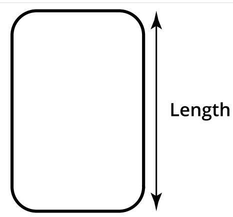
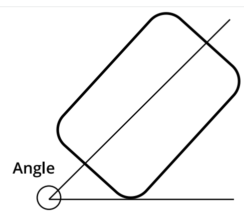


3D View


Parameter
Advanced
Anchoring
Source:
Top
Type:
Rectangle
Nominal Width:
30
mm
Nominal Length:
45
mm
Nominal Angle:
90
°
Nominal Radius:
5
mm
Width Tolerance:
10
mm
Length Tolerance:
10
mm
Angle Tolerance:
5
°
Partial Detection:
Region
X
Y
Z
Width
Length
51.306
Angle
Id:
11
Output
Filters
Decision
Min:
50
mm
Max:
60
mm

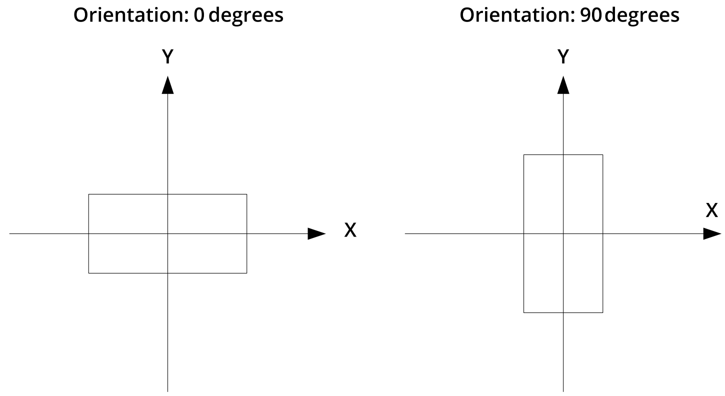
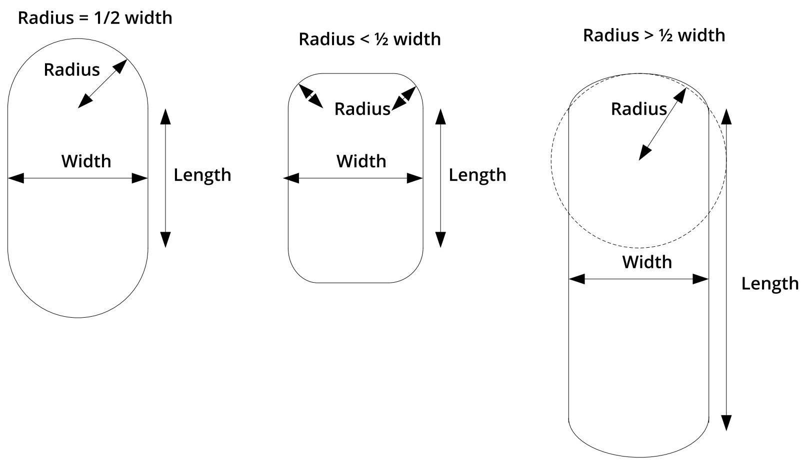

Parameter
Advanced
Anchoring
Reference Region
Auto Set
Tilt Correction
Auto Set
X
Y
Z
Width
Length
51.306
Angle
Id:
11
Output
Filters
Decision
Min:
50
mm
Max:
60
mm

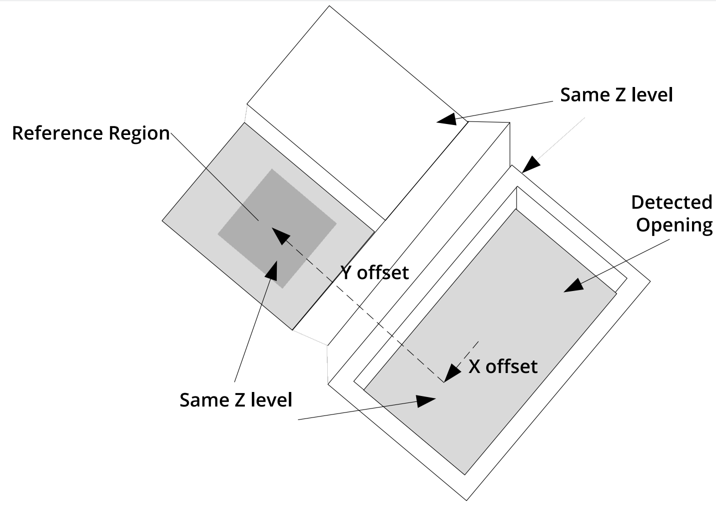
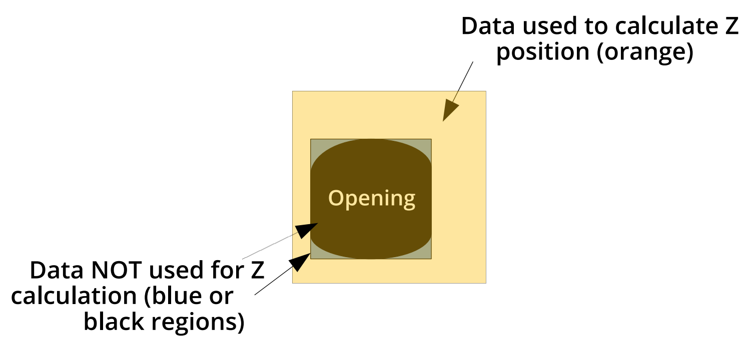
Measurements

Measurement	Illustration
X Determines the X position of the opening's center.	
Y Determines the Y position of the opening's center.	
Z Determines the Z position of the opening's center.	
Width Determines the width of the opening.	
Length Determines the length of the opening.	
Angle Determines the angle (rotation) around the normal of the alignment plane.	

Parameters

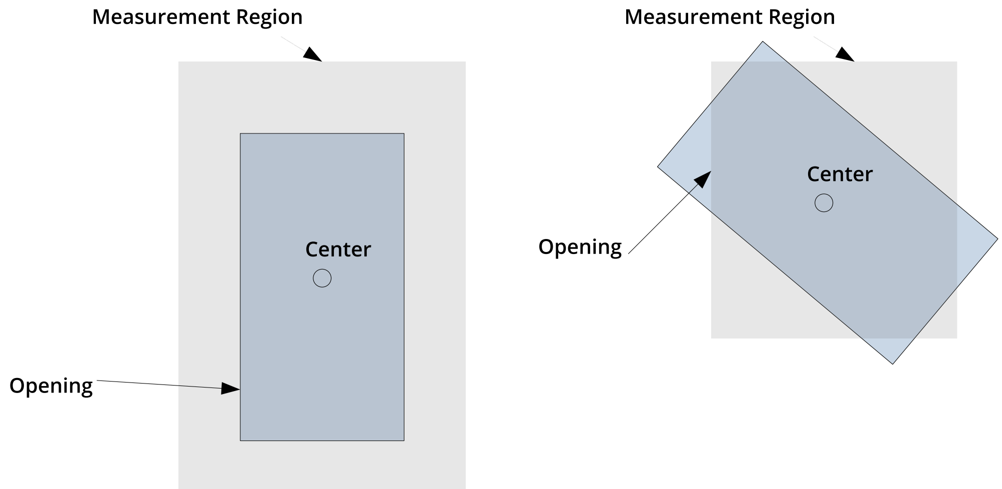
Parameter	Description
Type	Rounded Slot, Rectangle.
Nominal Width	Nominal width of the opening.
Nominal length	Nominal length of the opening.
Nominal Angle	Nominal angle of the opening. The default orientation is the length of the opening along the X axis.

Parameter	Description
	 The diagram above illustrates the case where the surface is not tilted. When the surface is tilted, the orientation is defined with respect to the normal of the surface, not with respect to the X-Y plane
Nominal Radius	Nominal radius of the opening ends. If the opening type is set to rectangular, the radius setting is disabled. The opening has an oval shape if the radius is equal to ½ of the width. The opening is a rounded rectangle when the radius is less than ½ of the width. 
Width Tolerance	The maximum variation from the nominal width (+/- from the nominal value).
Length Tolerance	The maximum variation from the nominal length (+/- from the nominal value).
Orientation Tolerance	The maximum variation from the nominal orientation (+/- from the nominal value).
Reference Regions	The algorithm uses reference regions to calculate the Z position of the opening. Reference regions are relative to the center location of the feature. This option is typically used in cases where the surface around the opening is not flat.

Parameter	Description
	 When the Reference Regions setting is disabled, the tool measures the opening's Z position using the all data in the measurement region, except for a bounding rectangular region around the opening.  With one or more reference region, the algorithm calculates the Z positions as the average values of the data within the regions. When the user places the reference region manually, all of the data is used, whether the data is inside or outside the opening. The user should place the reference region carefully.
Tilt Correction	Tilt of the target with respect to the alignment plane. Set to Auto-Set to have the tool automatically detect the target's tilt, or enter the angles manually. Auto-Set requires the measurement region to cover more areas on the surface plane than other planes. The results from the Plane X and Y tool can be used for angles X and Y parameters.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

Measurement Region

The center and the two sides and ends of the opening must be within the measurement region, even if Partial Detection is enabled.



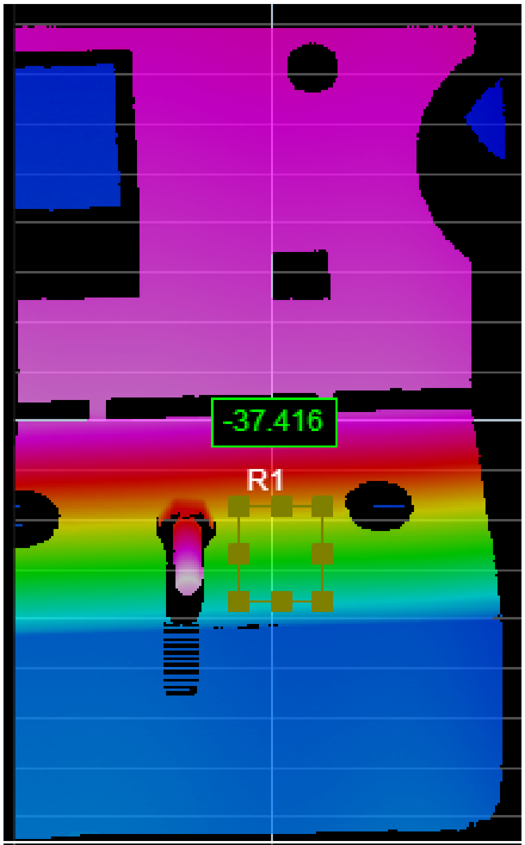
Plane

The Plane tool provides measurements that report angle X, angle Y, and offset Z of the surface with respect to the alignment target. The measurement value can be compared with minimum and maximum constraints to yield a decision.

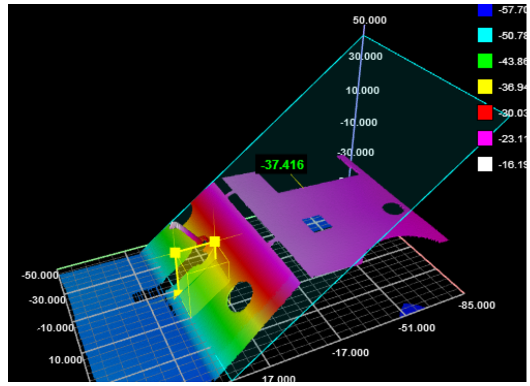
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

The Z offset reported is the Z position at zero position on the X axis and the Y axis.

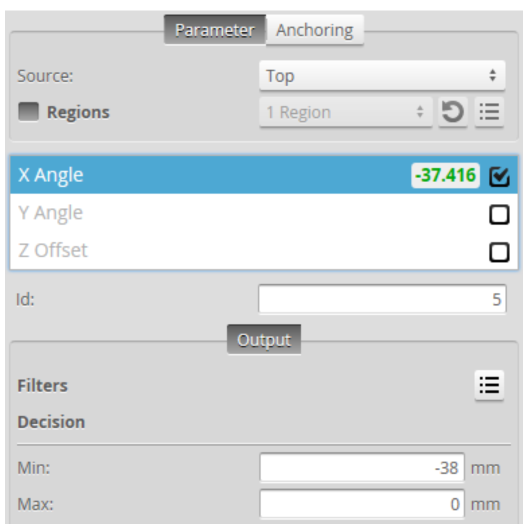
The results of the Plane Angle X and Plane Angle Y measurements can be used to customize the tilt angle in the Hole, Opening, and Stud tools.



2D View

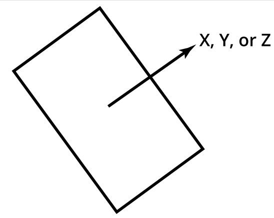
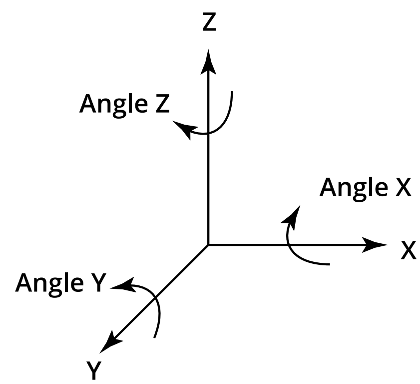


3D View



Measurement Panel

Measurements

Measurement	Illustration
Angle X Determines the X angle of the surface with respect to the alignment target.	
Angle Y Determines the Y angle of the surface with respect to the alignment target.	
Offset Z Determines the z offset of the surface with respect to the alignment target.	

Parameters

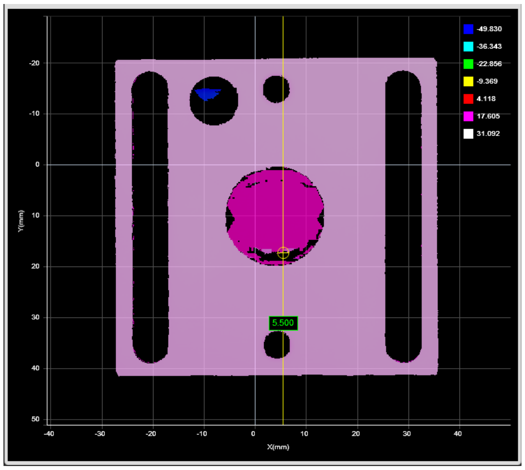
Parameter	Description
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

Position

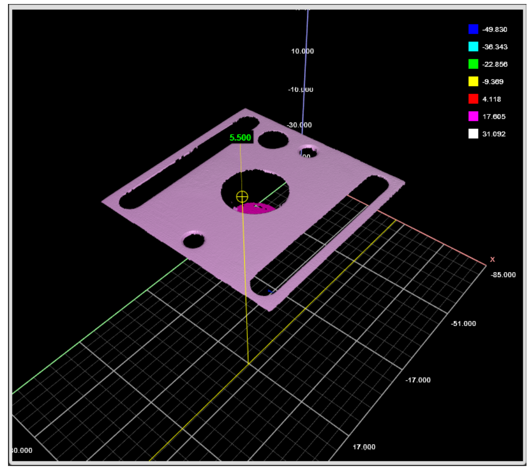
The Position tool reports the X, Y, or Z position of a part. The feature type must be specified and is one of the following: Average (the mean X, Y, and Z of the data points), Median (median X, Y, and Z of the data points), Centroid (the centroid of the data considered as a volume with respect to the z = 0 plane), Min X, Max X, Min Y, Max Y, Min Z, or Max Z.

The measurement value can be compared with minimum and maximum constraints to yield a decision.

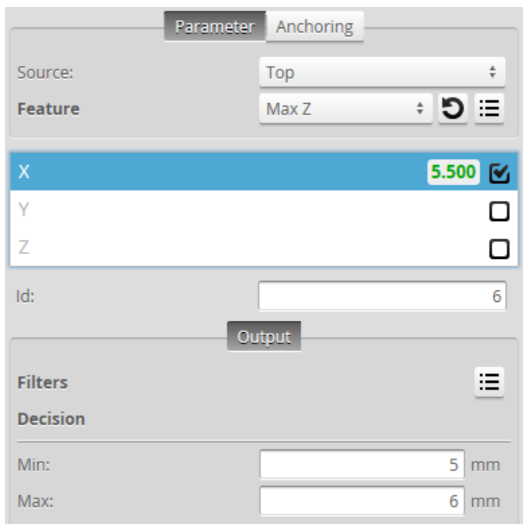
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.



2D View

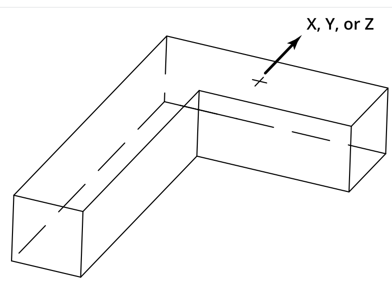


3D View



Measurement Panel

Measurements

Measurement	Illustration
X Determines the X position of the selected feature type.	
Y Determines the Y position of the selected feature type.	
Z Determines the Z position of the selected feature type.	

Parameters

Parameter	Description
Feature Type	One of the following: Average, Centroid, Min X, Max X, Min Y, Max Y, Min Z, Max Z, Median.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

Stud

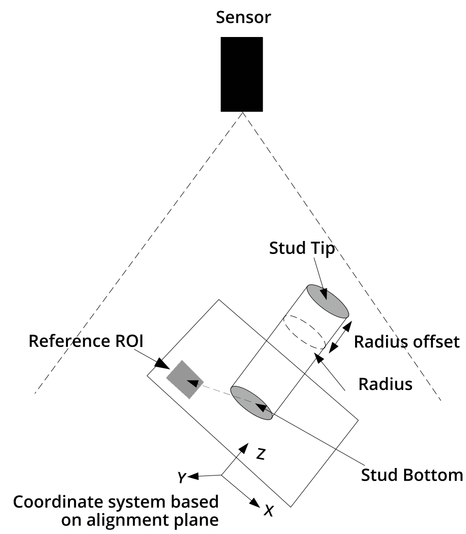
The Stud tool measures the location and radius of a stud.

See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.

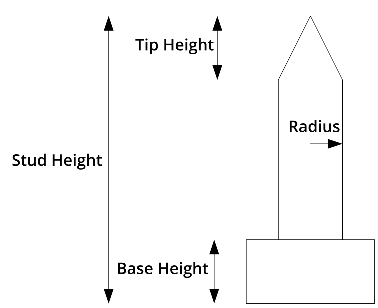
The tool uses a complex feature-locating algorithm to find a hold and then return measurements. See "Stud Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel.

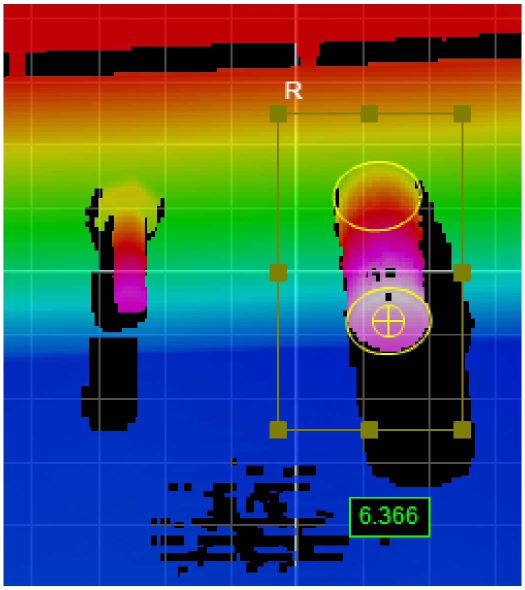
The measurement value can be compared with minimum and maximum constraints to yield a decision.

The location of the stud is defined at either the stud tip or the stud base. The tip is the intersection of the stud axis and the top of the stud; the base is the intersection of the stud axis and the surrounding plane.

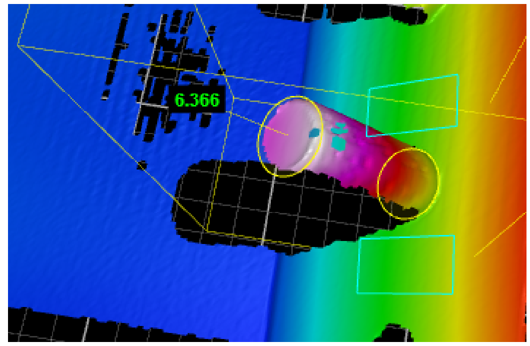


The stud shape is defined by the tip height and base height. The base and tip heights specify where the shaft with the nominal radius begins and ends.

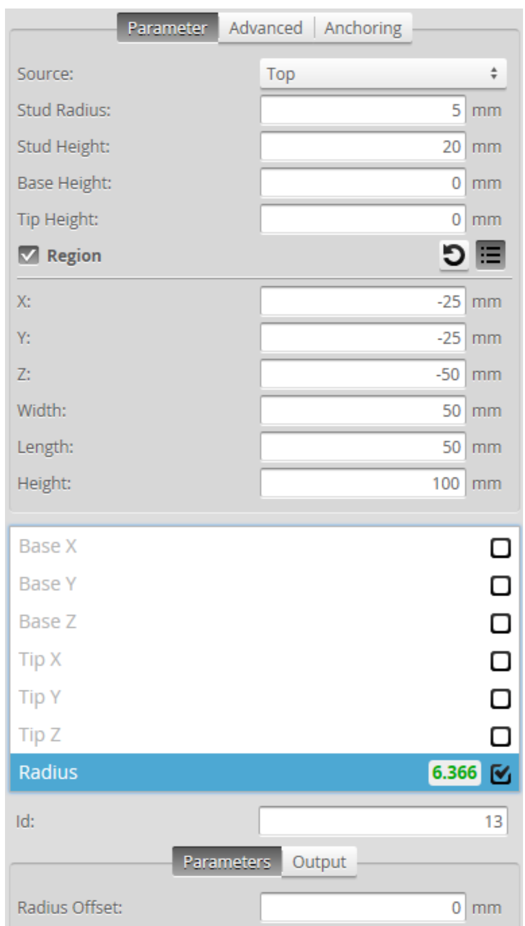


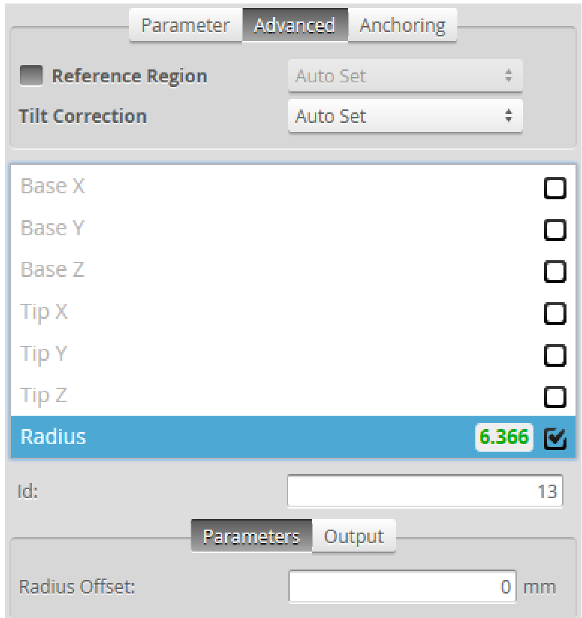


2D View



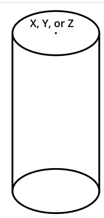
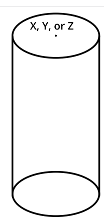
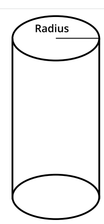
3D View





Measurement Panel

Measurements

Measurement	Illustration
Tip X Determines the X position of the stud tip.	
Tip Y Determines the Y position of the stud tip.	
Tip Z Determines the Z position of the stud tip.	
Base X Determines the X position of the stud base.	
Base Y Determines the Y position of the stud base.	
Base Z Determines the Z position of the stud base.	
Radius Determines the radius of the stud.	

Parameters

Parameter	Description
Nominal Stud Radius	Expected radius of the stud.
Nominal Stud Length	Expected length of the stud.
Base Height	The height above the base surface that will be ignored when the (truncated) cone is fit to the stud data.
Tip Height	The height from the top of the surface that will be ignored when the (truncated) cone is fit to the stud data.
Radius Offset *(Radius measurement only)*	The distance from the tip of the stud from which the radius is measured.
Reference Regions	The algorithm uses reference regions to calculate the base plane of the stud. Reference regions are relative to the base of the stud.
Tilt Correction	Tilt of the target with respect to the alignment plane. Set to Auto-Set to have the tool automatically detect the tilt, or enter the angles manually. Auto-Set requires the measurement region to cover more areas on the surface plane than other planes. The results from the Plane X and Y tool can be used for angles X and Y parameters.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

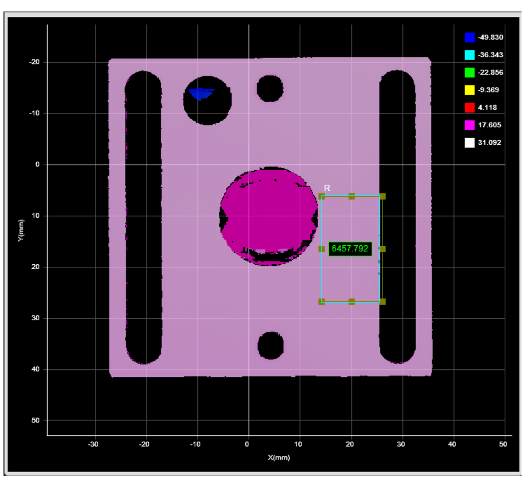
Measurement Region

The tip and the side of the stud must be within the measurement region.

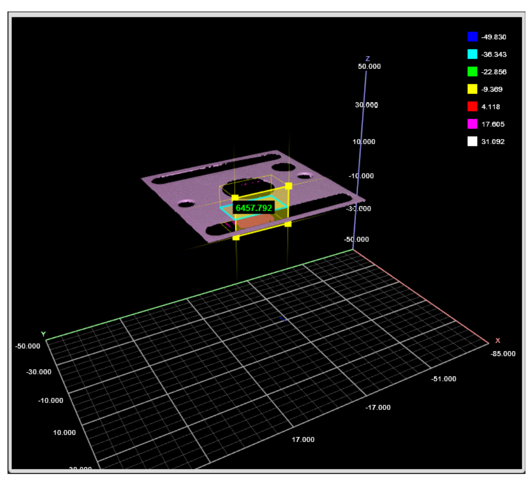
Volume

The Volume tool determines the volume, area, and thickness of a part. The measurement value can be compared with minimum and maximum constraints to yield a decision.

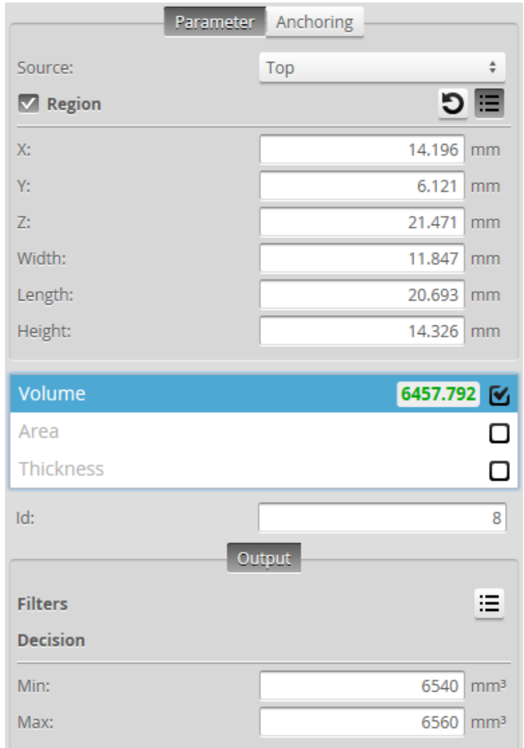
See *Adding and Removing Tools* on page 98 for instructions on how to add measurement tools.



2D View

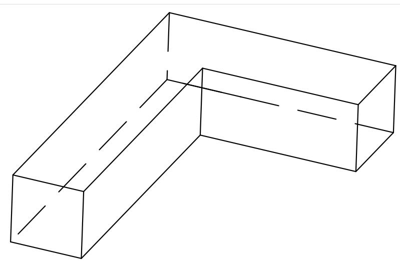
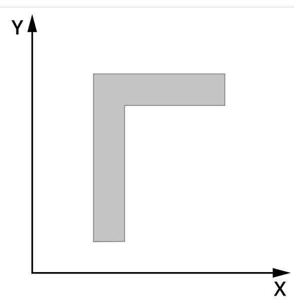


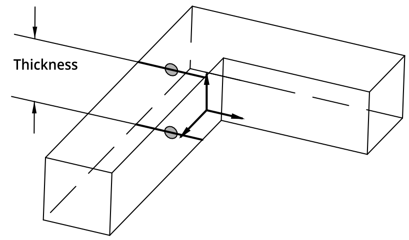
3D View



Measurement Panel

Measurements

Measurement	**Illustration**
Volume Measures volume in XYZ space.	
Area Measures area in the XY plane.	

Measurement	Illustration
Thickness Measures thickness (height) of a part.	

Parameters

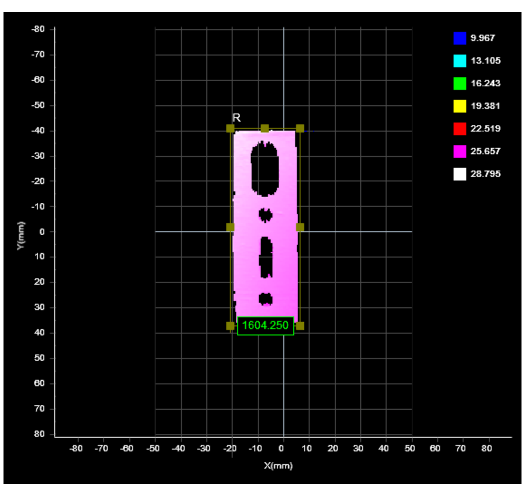
Parameter	Description
Location *(Thickness measurement only)*	One of the following: maximum height, minimum height, average height, median height, the height at the 2D centroid in the XY plane, or the height at the 3D centroid in XYZ space.
Decision	See *Decisions* on page 102.
Region	See *Regions* on page 101.
Output	See *Filters* on page 103.

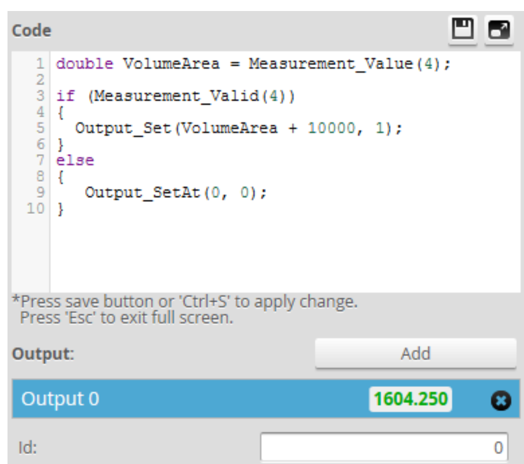
Script

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* on page 98 for instructions on how to add measurement tools.

See *Script Measurement* on the next page for more information on scripts.





See *Script Measurement* on the next page 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 3D point cloud 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.

> Scripts must be less than 27000 characters long.

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:

Function	Description
	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 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 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

Function	Description
	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)

Function	Description
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

The following example shows how to create a custom measurement that is based on the values from other measurements. The example calculates the 3-Dimensional Euclidean distance between the center of a hole and the center of the base of a nearby stud, then checks decision limits on this metric before sending the output.

```
/* Calculate the 3-Dimensional Euclidean distance between two points in 3D space*/

/* Retrieve 3D coordinate from Hole X, Y and Z tools (assumes these tools have been
configured as ID 0, ID 1 and ID 2 respectively) */
double HoleX = Measurement_Value(0);
double HoleY = Measurement_Value(1);
double HoleZ = Measurement_Value(2);

/* Retrieve 3D coordinate from StudBase X, Y and Z tools (assumes these tools have been
configured as ID 3, ID 4 and ID 5 respectively) */
double StudX = Measurement_Value(3);
double StudY = Measurement_Value(4);
double StudZ = Measurement_Value(5);

/* Calculate distance between points in 3D space */
double Distance = sqrt((HoleX - StudX)*(HoleX - StudX) + (HoleY - StudY)*(HoleY - StudY)
+ (HoleZ - StudZ)*(HoleZ - StudZ));

/* Min and Max Decision Limits */
/* Note that measurement values are in the unit of thousands of a millimeter in the
script */
/* In this example the distance is considered good if it's between 17.9 mm and 18.1 mm */
double MinDecisionLimit = 17.900;
double MaxDecisionLimit = 18.100;

if (Distance > MinDecisionLimit && Distance < MaxDecisionLimit)
{
   Output_Set(Distance, 1);
}
else
```

```
{
   Output_Set(Distance, 0);
}
```

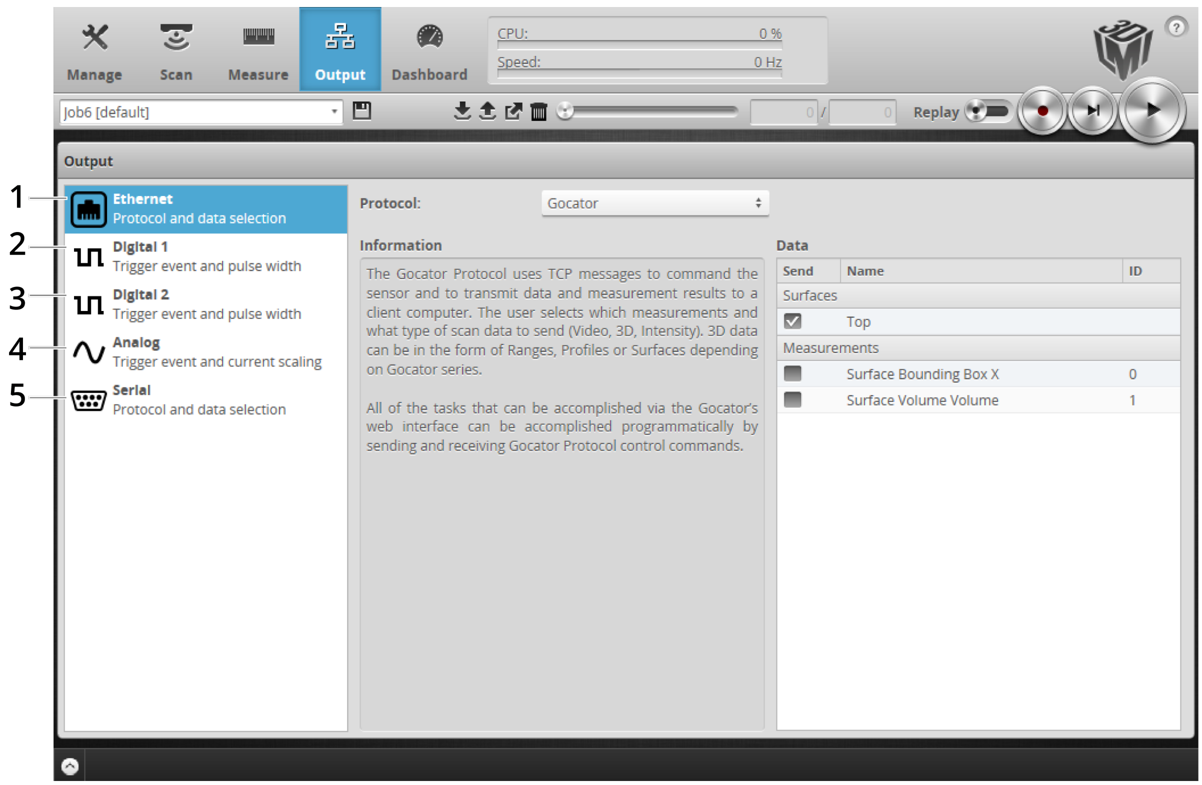
Output

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

Output Page Overview

Output configuration tasks are performed using the **Output** page. Gocator sensors can transmit 3D point cloud data 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.

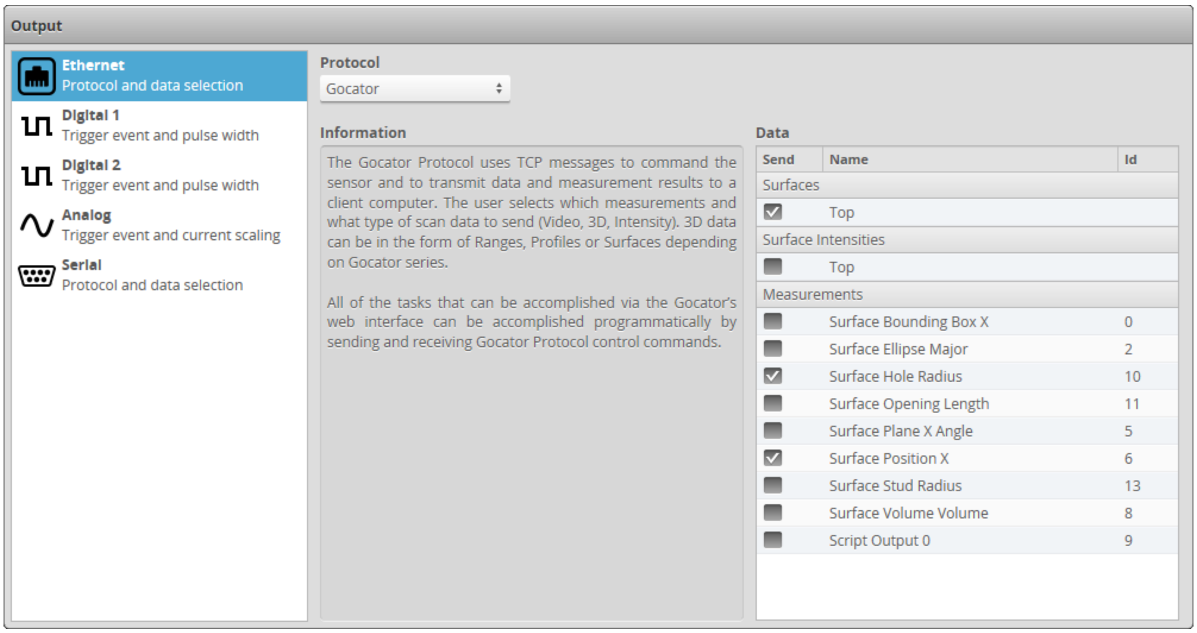


	Category	Description
1	Ethernet	Used to select the data sources that will transmit data via Ethernet. See *Ethernet Output* on the next page.
2	Digital Output 1	Used to select the data sources that will be combined to produce a digital output pulse on Output 1. See *Digital Output* on page 148.
3	Digital Output 2	Used to select the data sources that will be combined to produce a digital output pulse on Output 2. See *Digital Output* on page 148.
4	Analog Panel	Used to convert a measurement value or decision into an analog output signal. See *Analog Output* on page 151.
5	Serial Panel	Used to select the measurements that will be transmitted via RS-485 serial output. See *Serial Output* on page 153.

Ethernet Output

A sensor uses TCP messages (Gocator protocol) to receive commands from client computers, and to send video, 3D point clouds, 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* on page 211 for the specification of these protocols.

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

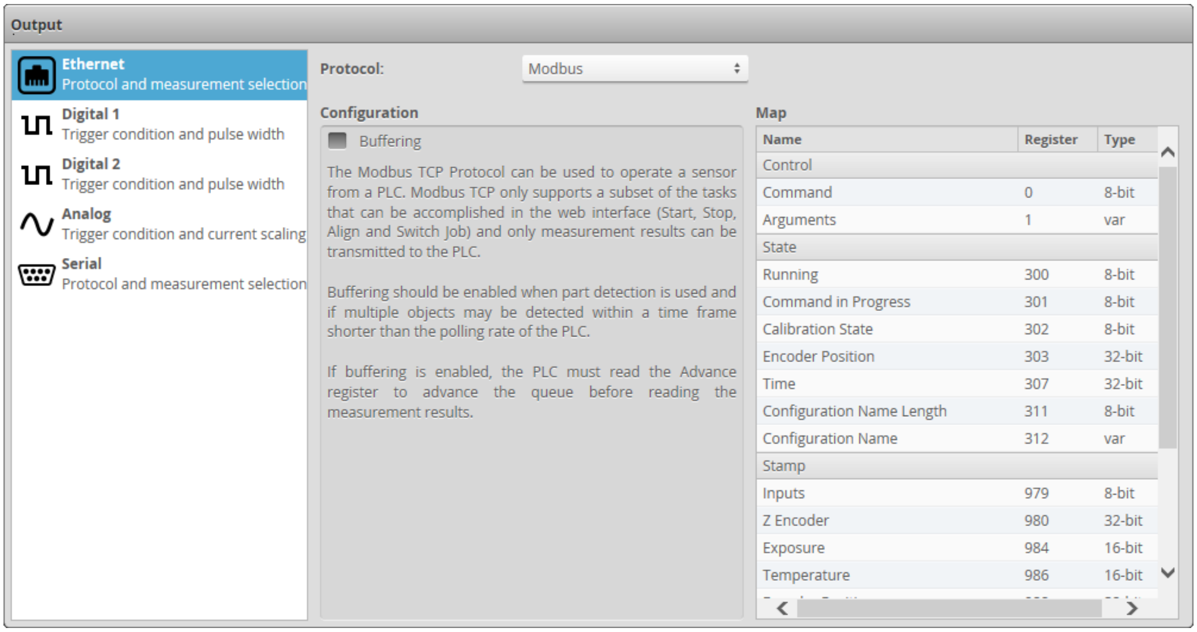


To receive commands and send results using Gocator Protocol messages:

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, data, intensity, or measurement items to send.

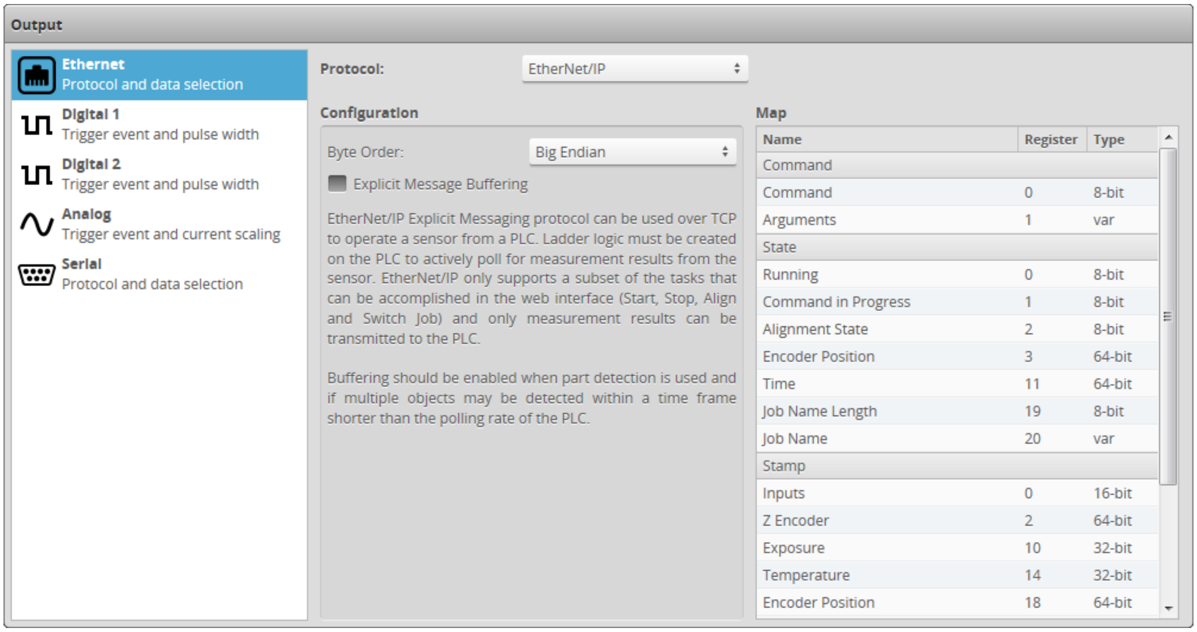
> Measurements shown here correspond to measurements that have been added using the **Measure** page (see on page 97).

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.



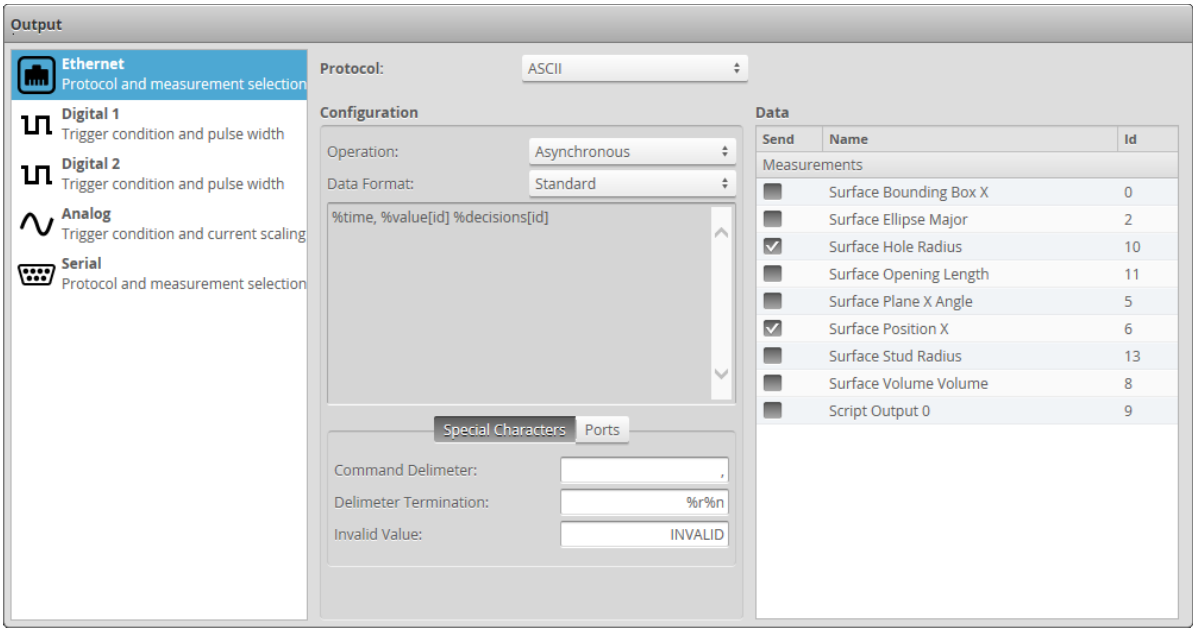
To receive commands and send results using Modbus TCP messages:

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 **Explicit Message 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 on page 264) is read.
5. Choose the byte order in the **Byte Order** dropdown.



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 *Polling Operation Commands (Ethernet Only)* on page 267 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* on page 273 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

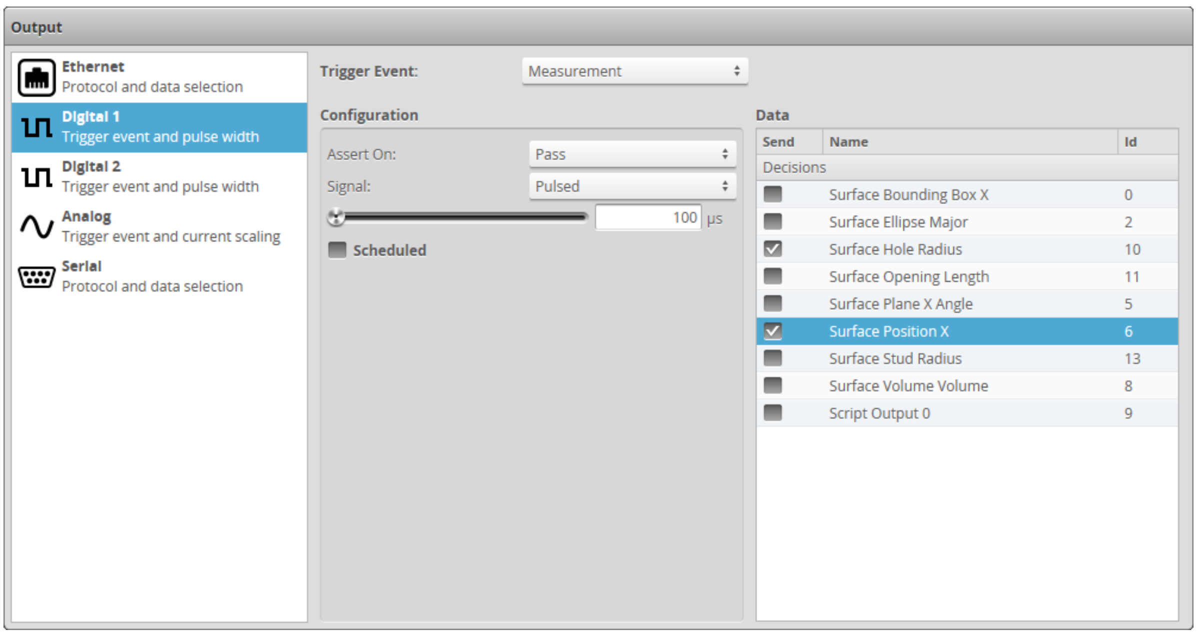
Gocator sensors can convert measurement decisions or software commands to digital output pulses, which can then be used to output to a PLC or to control external devices, such as indicator lights or air ejectors.

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* on page 305 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 on page 229) 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).

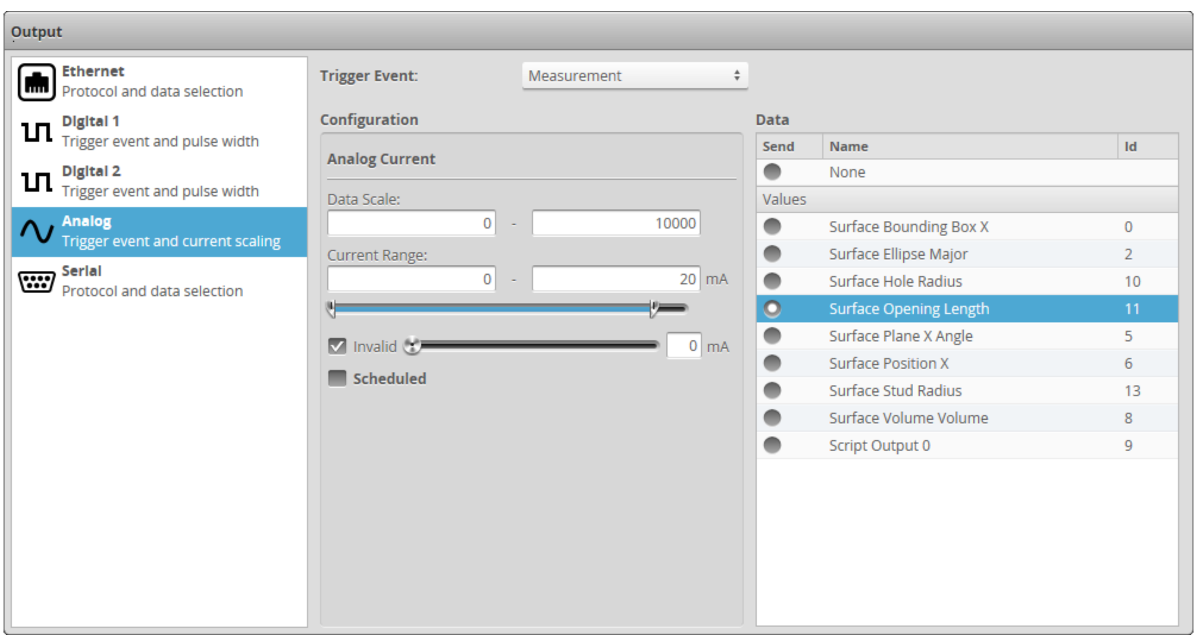
To output an exposure signal:

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.

For information on wiring analog output to an external device, see *Analog Output* on page 307



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 millimeters for dimensional measurements such as distance, square millimeters for areas, cubic millimeters for volumes, and degrees for angle results.
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* on page 54 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:

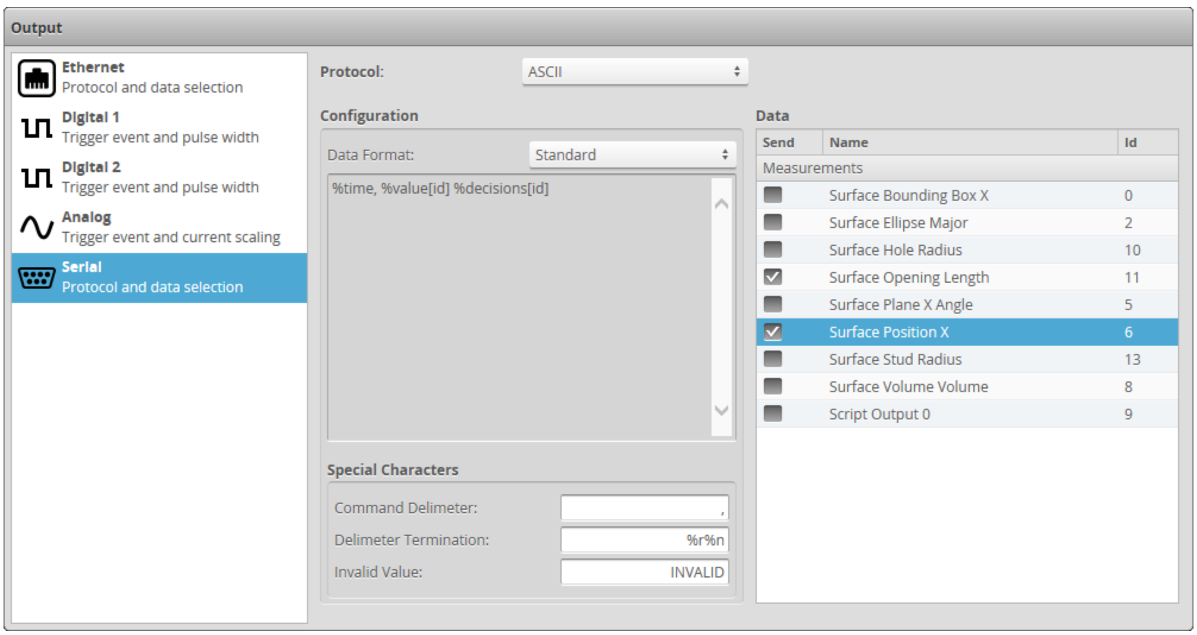
1. Go to the **Output** page.
2. Click on **Analog** in the **Output** panel.
3. Set **Trigger Event** to **Software**.

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 on page 230) 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.

The ASCII protocol outputs data asynchronously using a single serial port. See *ASCII Protocol* on page 265 for the ASCII Protocol parameters and data formats.For information on wiring serial output to an external device, see *Serial Output* on page 307.



To configure ASCII output:

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* on page 273 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* on page 274 for the supported format string syntax.

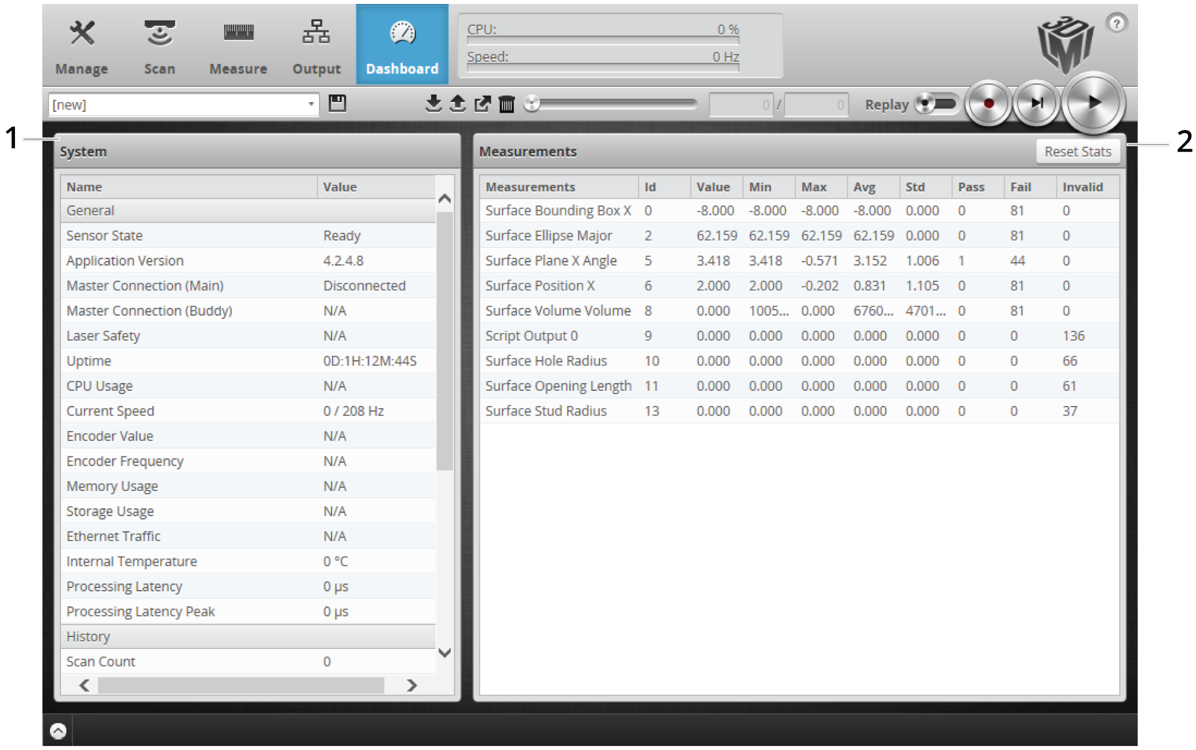
5. Select the measurments to send.
   Select measurements by placing a check in the corresponding check box.
6. Set the **Special Characters**.
   Select the delimiter, termination and invalid value characters. Special characters are used in commands and standard-format data results.

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* on the 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.
Laser Safety	Whether Safety is enabled.
Uptime	Length of time since the sensor was power-cycled or reset.

Name	Description
CPU Usage	Sensor CPU utilization (%).
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:

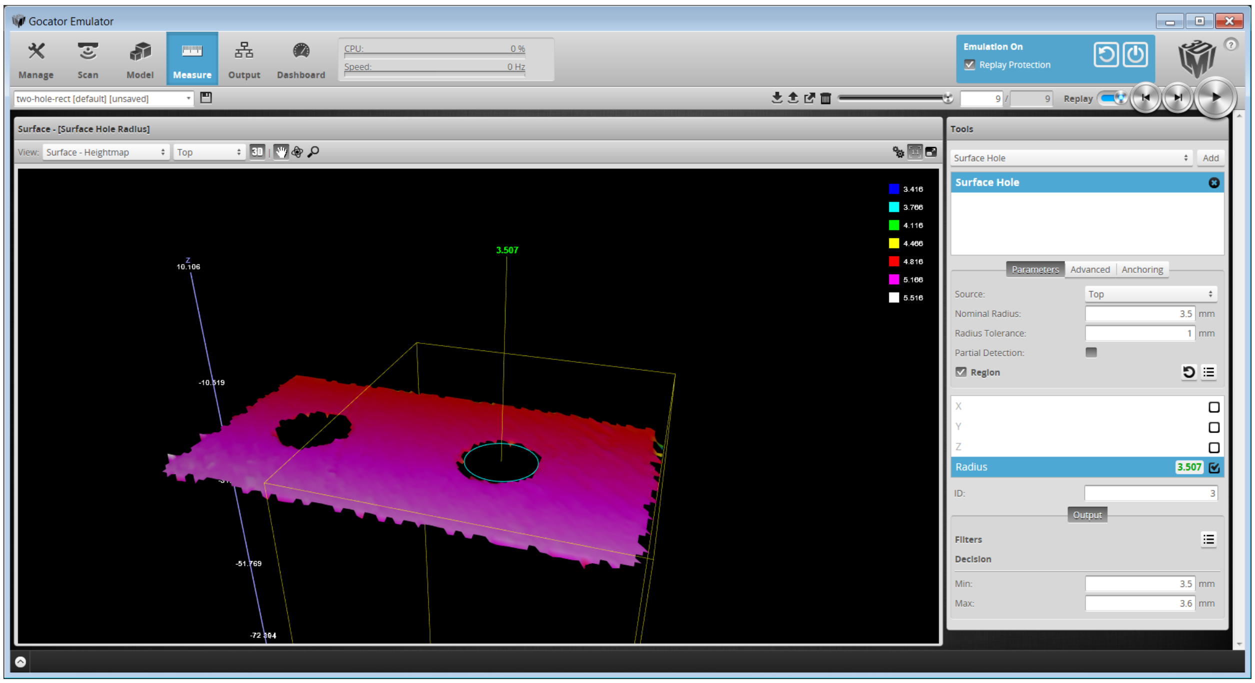
Dashboard Measurement Statistics

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 Emulator

The Gocator emulator is a stand-alone application that lets you run a "virtual" sensor. In a virtual sensor, you can test jobs, evaluate data, and even learn more about new features, rather than take a physical device off the production line to do this. You can also use a virtual sensor to familiarize yourself with the overall interface if you are new to Gocator.

> The Gocator emulator is only supported on Windows 7 and 8.



Emulator showing a part in recorded data.
A measurement is applied to the recorded data.

Limitations

In most ways, the emulator behaves like a real sensor, especially when visualizing data, setting up models and part matching, and adding and configuring measurement tools. The following are some of the limitations of the emulator:

- Changes to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). However, you can keep modified jobs by first saving them and then downloading them from the **Jobs** list on the **Manage** page to a client computer. The job files can then be loaded into the emulator at a later time or even onto a physical sensor for final testing.
- Output is not supported.
- Performing alignment in the emulator has no effect and will never complete.
- Only one instance can be run at a time.

For information on saving and loading jobs in the emulator, see *Creating, Saving, and Loading Jobs* on page 162 .

For information on uploading and downloading jobs between the emulator and a computer, and performing other job file management tasks, see *Downloading and Uploading Jobs* on page 166.

Downloading a Support File

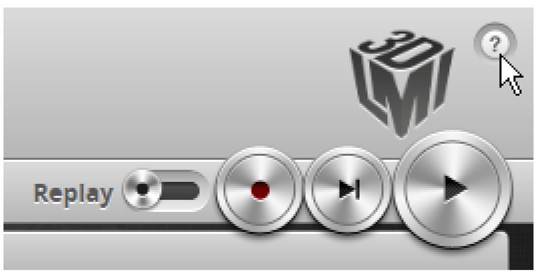
The emulator is provided with several virtual sensors preinstalled.

You can also create virtual sensors yourself by downloading a support file from a physical Gocator and then adding it to the emulator.

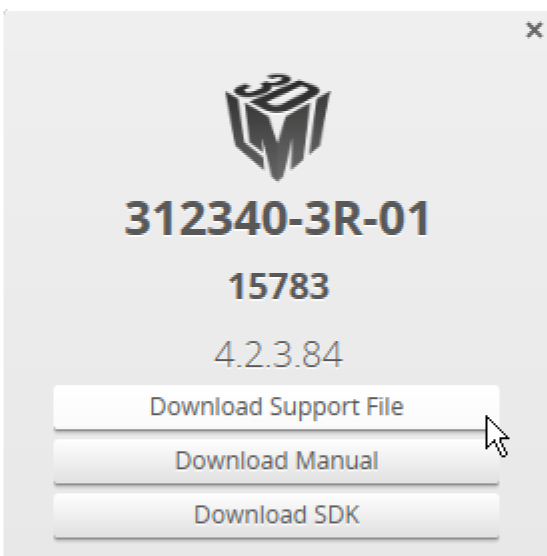
Support files can contain jobs, letting you configure systems and add measurements. They can also contain replay data, letting you test measurements and some configurations on real data.

To download a support file:

1. Click on the Help icon in the upper right corner of the Web interface.



2. Click on **Download Support File** in the dialog that opens.



3. If you want, change the name from the default **support.gs**.

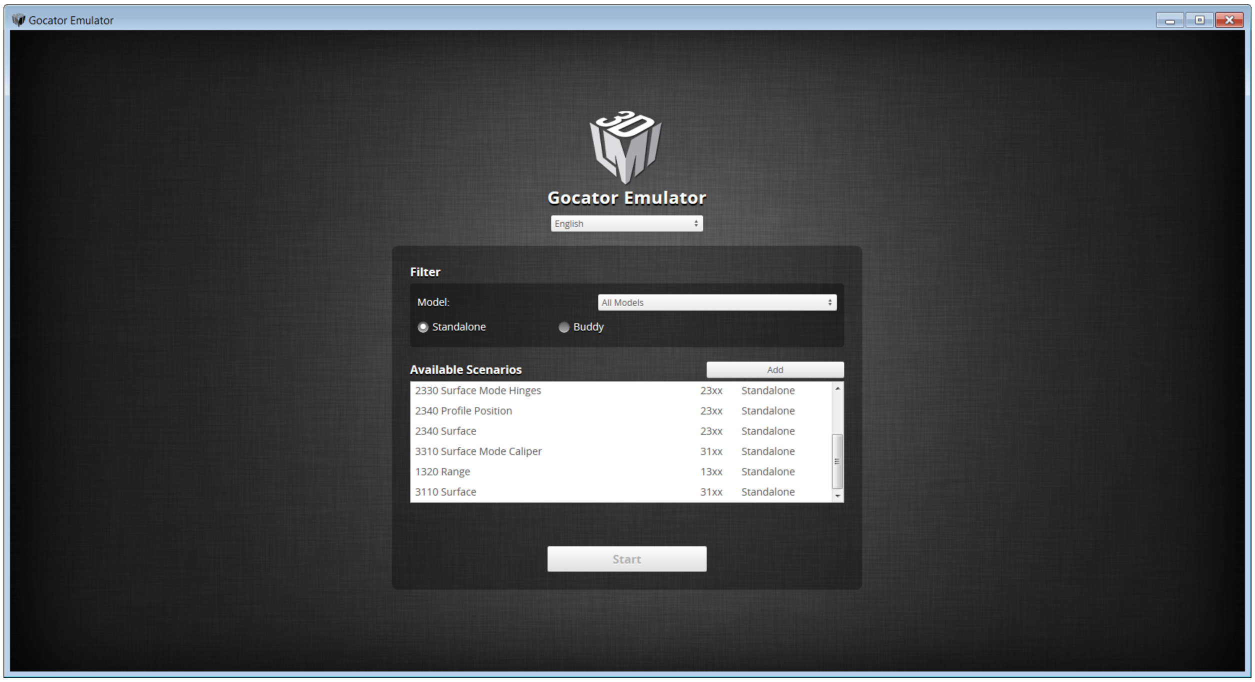
   The name of the .gs file, excluding the extension, is used in the **Available Scenarios** list in the emulator launch screen.

4. Save the file.

Running the Emulator

The emulator is contained in the Gocator tools package (14405-x.x.x.x_SOFTWARE_GO_Tools.zip). You can download the package by going to http://lmi3d.com/support/downloads/, selecting a product type, and clicking on the *Product User Area* link.

To run the emulator, unzip the package and double-click on *\Emulator\bin\win32\GoEmulator.exe*.



Emulator launch screen

You can change the language of the emulator's interface from the launch screen. To change the language, choose a language option from the top drop-down:

Selecting the emulator interface language

Adding a Scenario to the Emulator

If you have downloaded a support file from a physical sensor, you must add it as a scenario in the emulator.

You can add support files downloaded from any series of Gocator sensors to the emulator.

To add a scenario:

1. Launch the emulator if it isn't running already.
2. Click the **Add** button and choose a previously saved support file (.gs extension) in the **Choose File to Upload** dialog.

Running a Scenario

After you have added a virtual sensor by uploading a support file to the emulator, you can run it from the **Available Scenarios** list on the emulator launch screen. You can also run any of the scenarios included in the installation.

To run a scenario:

1. If you want to filter the scenarios listed in **Available Scenarios**, do one or both of the following:
   - Choose a model family in the **Model** drop-down.
   - Choose **Standalone** or **Buddy** to limit the scenarios to single-sensor or dual-sensor scenarios, respectively.
2. Select a scenario in the **Available Scenarios** list and click **Start**.

Removing a Scenario from the Emulator

To remove a scenario from the emulator, you must remove it manually from the computer's file system.

The following procedure applies to Windows 7.

To remove a scenario:

1. Close the Gocator emulator.
1. Click the Windows **Start** button.
2. In the **Search programs and files** field, type **%LOCALAPPDATA%\LMI\GoEmulate\datasets** and press Enter.

   Windows will open Windows Explorer showing the folder containing the scenarios.

   You must select **Show hidden files, folders, and drives** in the Windows Explorer **Folder Options** dialog (**View** tab) to see this folder.
3. Delete the folder of the scenario you want to remove.

Using Replay Protection

Because making changes to certain settings on the **Scan** page causes replay data to be flushed, you can use the **Replay Protection** option in the **Emulation On** message to protect replay data. (The

Emulation On message is displayed whenever the emulator is running a virtual sensor).

When **Replay Protection** is on, you cannot switch from Replay mode. Settings that do not affect replay data can be changed.

Replay Protection is on by default.

Stopping and Restarting the Emulator

To stop the emulator, click on the **Stop Emulator** button in the **Emulation On** message. Stopping the emulator returns you to the launch screen. (To close the application, click on the Close button in the upper right corner of the application.)

To restart the emulator, click the **Restart Emulator** button. Restarting the emulator restarts the currently running simulation.

Working with Jobs and Data

The following topics describe how to work with jobs and replay data (data recorded from a physical sensor) in the emulator.

Creating, Saving, and Loading Jobs

Changes saved to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). To keep jobs permanently, you must first save the job in the emulator and then download the job file to a client computer. See below for more information on creating, saving, and switching jobs. For information on downloading and uploading jobs between the emulator and a computer, see *Downloading and Uploading Jobs* on page 166.

The job drop-down list in the toolbar shows the jobs available in the emulator. 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 create a job:

1. Choose **[New]** in the job drop-down list and type a name for the job.
2. Click the **Save** button or press **Enter** to save the job.
   The job is saved to the emulator using the name you provided.

To save a job:

- Click the **Save** button .

  The job is saved to the emulator.

To load (switch) jobs:

- Select an existing file name in the job drop-down list.

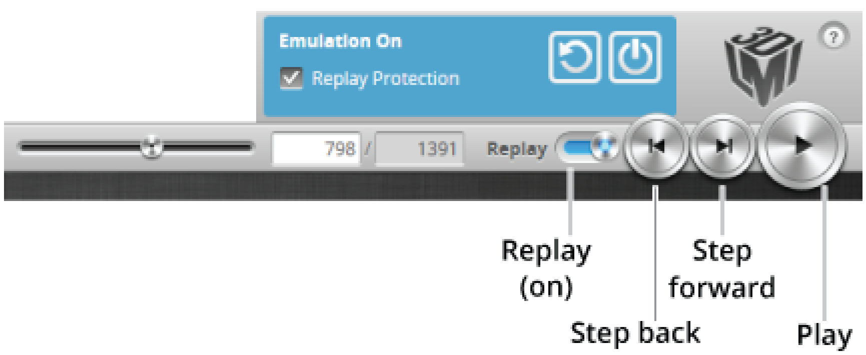
  The job is activated. If there are any unsaved changes in the current job, you will be asked whether you want to discard those changes.

Playback and Measurement Simulation

The emulator can replay scan data previously recorded by a physical sensor, and 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.

Playback is controlled by using the toolbar controls.

	Recording is not functional in the emulator.



Playback controls when replay is on

To replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.
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 until the end of the

replay data.

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 replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Go to the **Measure** page.
   Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired. For information on adding and configuring measurements, see *Measurement* on page 97.

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; for more information on the dashboard, see *Dashboard* on page 155.

To clear replay data:

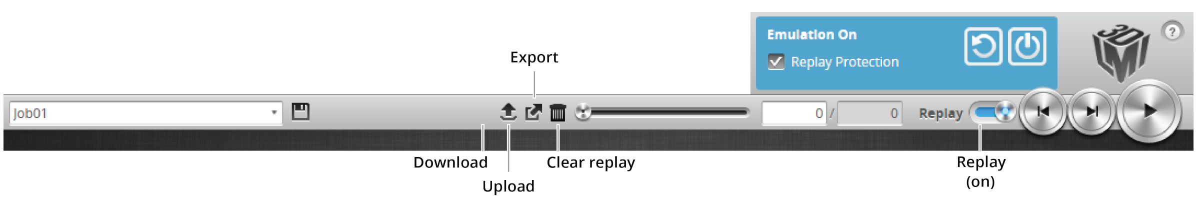
- Click the **Clear Replay Data** button.

Downloading, Uploading, and Exporting Replay Data

Replay data (recorded scan data) can be downloaded from the emulator to a client computer, or uploaded from a client computer to the emulator.

Data can also be exported from the emulator to a client computer in order to process the data using third-party tools.

You can only upload replay data to the same sensor model that was used to create the data.



Replay data is not loaded or saved when you load or save jobs.

To download replay data:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Click the **Download** button.

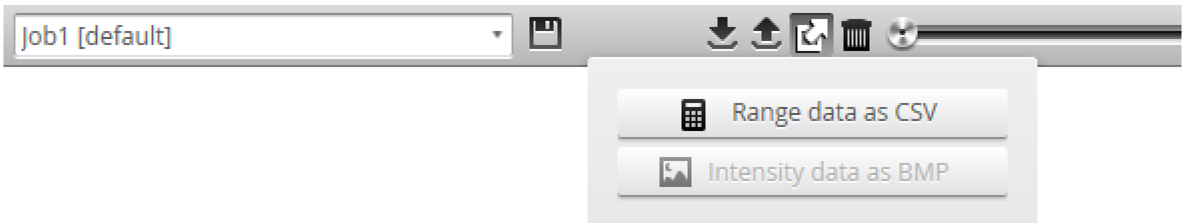
To upload replay data:

1. Toggle **Replay** mode on by setting the slider to the left in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Click the **Upload** button.

3. Select the directory and the file name to upload from the client computer and click **OK**.

Replay data can be exported using the CSV format.



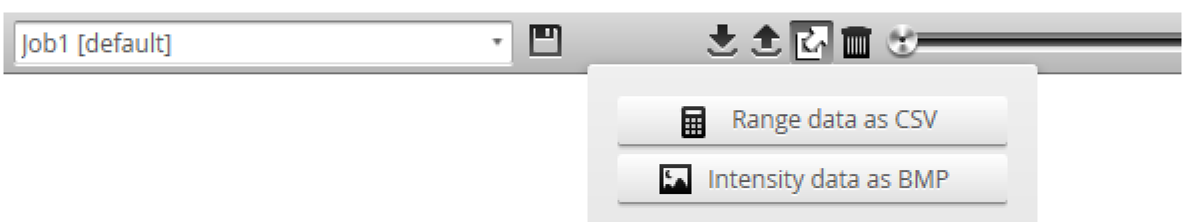
To export replay data in the CSV format:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Click the **Export** button and select **Export Range Data as CSV**.
   Only data at the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Playback and Measurement Simulation* on page 163.

3. Optionally, convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 292.

Recorded intensity data can be exported to a bitmap (.BMP format). **Acquire Intensity** must be checked in the **Scan Mode** panel while data was being recorded in order to export intensity data.



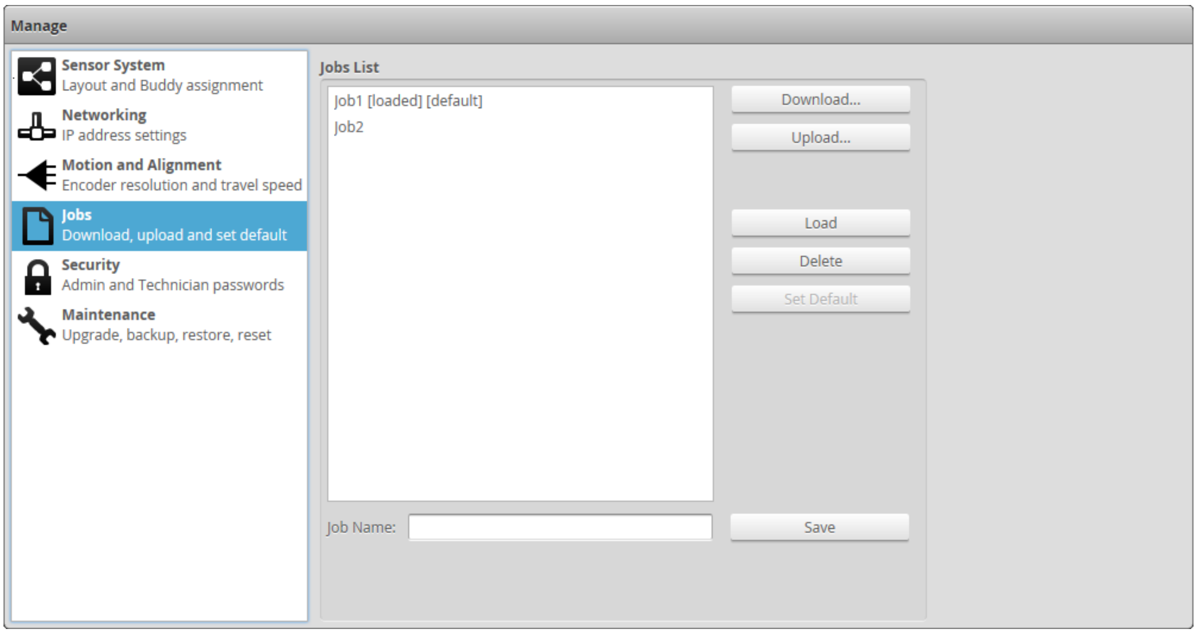
To export recorded intensity data to the BMP format:

1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
   The slider's background turns blue.
   To change the mode, **Replay Protection** must be unchecked.

2. Click the **Export** button and select **Intensity data as BMP**.
   Only the intensity data in the current replay location is exported.
   Use the playback control buttons to move to a different replay location; for information on playback, see *To replay data* in *Playback and Measurement Simulation* on page 163.

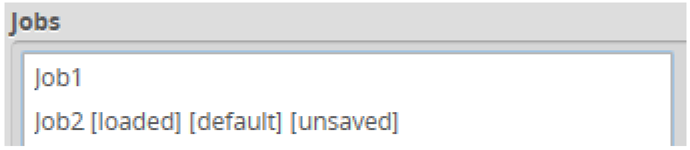
Downloading and Uploading Jobs

The **Jobs** category on the **Manage** page lets you manage the jobs in the emulator.



Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the emulator.
Save button	Saves current settings to the job using the name in the **Job Name** field. Changes to job files are not persistent in the emulator. To keep changes, first save changes in the job file, and then download the job file to a client computer. See the procedures below for instructions.
Load button	Loads the job that is selected in the job list. Reloading the current job discards any unsaved changes.
Delete button	Deletes the job that is selected in the job list.
Set as Default button	Setting a different job as the default is not persistent in the emulator. The job set as default when the support file (used to create a virtual sensor) was downloaded is used as the default whenever the emulator is started.
Download... button	Downloads the selected job to the client computer.
Upload... button	Uploads a job from the client computer.

Unsaved jobs are indicated by "[unsaved]".



> Changes to job files in the emulator are *not* persistent (they are lost when you close or restart the emulator). However, you can keep modified jobs by first saving them and then downloading them to a client computer.

To save a job:

1. Go to the **Manage** page and click on the **Jobs** category.
2. Provide a name in the **Name** field.
   To save an existing job under a different name, click on it in the **Jobs** list and then modify it in the **Name** field.
3. Click on the **Save** button or press **Enter**.

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.

Scan, Model, and Measurement Settings

The settings on the **Scan** page related to actual scanning will clear the buffer of any scan data that is uploaded from a client computer, or is part of a support file used to create a virtual sensor. If **Replay Protection** is checked, the emulator will indicate in the log that the setting can't be changed because the change would clear the buffer. For more information on Replay Protection, see *Using Replay Protection* on page 161.

Other settings on the **Scan** page related to the post-processing of data can be modified to test their influence on scan data, without modifying or clearing the data, for example edge filtering (page 72), and filters on the X axis (page 67). Note that modifying the Y filters causes the buffer to be cleared.

For information on creating models and setting up part matching, see *Models and Part Matching* on page 82. For information on adding and configuring measurement tools, see *Measurement* on page 97.

Calculating Potential Maximum Frame Rate

You can use the emulator to calculate the potential maximum frame rate you can achieve with different settings.

For example, when you reduce the active area, in the **Active Area** tab on the **Sensor** panel, the maximum frame rate displayed on the **Trigger** panel is updated to reflect the increased speed that would be available in a physical Gocator sensor. (See *Active Area* on page 58 for more information on active area.)

Similarly, you can adjust exposure on the **Exposure** tab on the **Sensor** panel to see how this affects the maximum frame rate. (See *Exposure* on page 60 for more information on exposure.)

> To adjust active area in the emulator, **Replay Protection** must be turned off. See *Using Replay Protection* on page 161 for more information.

> Saving changes to active area causes replay data to be flushed.

Gocator Device Files

This section describes the user-accessible device files stored on a Gocator.

Live Files

Various "live" files stored on a Gocator sensor represent the sensor's active settings and transformations (represented together as "job" files), the active replay data (if any), and the sensor log.

By changing the live job file, you can change how the sensor behaves. For example, to make settings and transformations active, write to or copy to the _live.job file. You can also save active settings or transformations to a client computer, or to a file on the sensor, by reading from or copying these files, respectively.

> The live files are stored in volatile storage. Only user-created job files are stored in non-volatile storage.

The following table lists the live files:

Live Files

Name	Read/Write	Description
_live.job	Read/Write	The active job. This file contains a Configuration component containing the current settings. If Alignment Reference in the active job is set to Dynamic, it also contains a Transform component containing transformations. For more information on job files (live and user-created), accessing their components, and their structure, see *Job Files* on the next page.
_live.cfg	Read/Write	A standalone representation of the Configuration component contained in _live.job. Used primarily for backwards compatibility.
_live.tfm	Read/Write	**If Alignment Reference of the active job is set to Dynamic:** A copy of the Transform component in _live.job. Used primarily for backwards compatibility. **If Alignment Reference of the active job is set to Fixed:** The transformations that are used for *all* jobs whose Alignment Reference

Name	Read/Write	Description
		setting is set to Fixed.
_live.log	Read	A sensor log containing various messages. For more information on the log file, see *Log File* below.
_live.rec	Read/Write	The active replay simulation data.
ExtendedId.xml	Read	Sensor identification.

Log File

The log file contains log messages generated by the sensor. The root element is *Log*.

To access the log file, use the Read File command, passing "_live.log" to the command. The log file is read-only.

Log Child Elements

Element	Type	Description
List of (Info | Warning | Error)	List	An ordered list of log entries.

Log/Info | Log/Warning | Log/Error Elements

Element	Type	Description
@time	64u	Log time, in uptime (µs).
@value	String	Log content; may contain printf-style format specifiers (e.g. %u).
List of (IntArg | FloatArg | Arg)	List	An ordered list of arguments: IntArg – Integer argument FloatArg – Floating-point argument Arg – Generic argument

The arguments are all sent as strings and should be applied in order to the format specifiers found in the content.

Job Files

The following sections describe the structure of job files.

Job files, which are stored in a Gocator's internal storage, control system behavior when a sensor is running. Job files contain the settings and potentially the transformations associated with the job (if Alignment Reference is set to Dynamic).

There are two kinds of job files:

- A special job file called "_live.job." This job file contains the *active* settings and potentially the transformations associated with the job. Changing this file (or its components) changes the active settings or transformations. It is stored in volatile storage.
- Other job files that are stored in non-volatile storage.

Job File Components

A job file contains components that can be loaded and saved as independent files. The following table lists the components of a job file:

Job File Components

Component	Path	Description
Configuration	config.xml	The job's configurations. This component is always present.
Transform	transform.xml	Transformation values. Present only if Alignment Reference is set to Dynamic.
Part model	<name>.mdl	One or more part model files. Part models are created using models and part matching.

Elements in the components 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.

Accessing Files and Components

Job file components can be accessed individually as XML files using path notation. For example, the configurations in a user-created job file called *productionRun01.job* can be read by passing "productionRun01.job/config.xml" to the Read File command. In the same way, the configurations in the active job could be read using "_live.job/config.xml".

> If Alignment Reference is set to Fixed, the active job file (_live.job) will not contain transformations. To access transformations in this case, you must access them via _live.tfm.

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

Configuration

The Configuration component of a job file contains settings that control how a Gocator sensor behaves.

You can access the Configuration component of the active job as an XML file, either using path notation, via "_live.job/_config.xml", or directly via "_live.cfg".

You can access the Configuration component in user-created job files in non-volatile storage, for example, "productionRun01.job/config.xml". You can only access configurations in user-created job files using path notation.

See the following sections for the elements contained in this component.

Configuration Child Elements

Element	Type	Description
@version	32u	Configuration version (101).
@versionMinor	32u	Configuration minor version (3).
Setup	Section	See *Setup* below for a description of the Setup elements.
ToolOptions	Section	List of available tool types and their information. See *ToolOptions* on page 185 for details.
Tools	Collection	Collection of sections. Each section is an instance of a tool and is named by the type of the tool it describes. For more information, see the sections for each tool under *Tools* on page 186.
Tools.options	String (CSV)	List of available tool types.
Output	Section	See *Output* on page 201 for a description of the Output elements.

Setup

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

Setup Child Elements

Element	Type	Description
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.
Filters	Section	See *Filters* on the next page. Used by Gocator 2300, 2880, and 3100 series sensors.
Trigger	Section	See *Trigger* on page 174.
Layout	Section	See *Layout* on page 175.
Alignment	Section	See *Alignment* on page 175.
Devices	Collection	A collection of two Device sections (with roles main and buddy). See *Devices / Device* on page 177.

Element	Type	Description
SurfaceGeneration	Section	See *SurfaceGeneration* on page 180. Used by Gocator 2300 and 2880 series sensors.
ProfileGeneration	Section	See *ProfileGeneration* on page 181. Used by Gocator 1300 series sensors.
PartDetection	Section	See *PartDetection* on page 182.
PartMatching	Section	See *PartMatching* on page 184. Used by Gocator 2300, 2880, and 3100 series sensors.
Custom	Custom	Used by specialized sensors.

Filters

The Filters element contains settings related to post-processing profiles before they are output or used by measurement tools. This element is used by Gocator 2300, 2880, and 3100 series sensors.

XSmoothing

XSmoothing Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YSmoothing

YSmoothing Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XGapFilling

XGapFilling Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YGapFilling

YGapFilling Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XMedian

XMedian Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YMedian

YMedian Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XDecimation

XDecimation Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YDecimation

YDecimation Child Elements

Element	Type	Description
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

Trigger

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

Trigger Child Elements

Element	Type	Description
Source	32s	Trigger source: 0 – Time 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.

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
XSpacingCount	32u	Number of points along X when data is resampled.
YSpacingCount	32u	Number of points along Y when data is resampled.
TransformedDataRegion	Region3D	Transformed data region of the layout output.
Orientation	32s	Sensor orientation: 0 – Wide 1 – Opposite 2 – Reverse
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	64f	Delay in µs. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod	64f	Period in µs. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod.min	64f	Minimum period in µs.

Region3D Child Elements

Element	Type	Description
X	64f	X start (mm).
Y	64f	Y start (mm).
Z	64f	Z start (mm).
Width	64f	X extent (mm).
Length	64f	Y extent (mm).
Height	64f	Z extent (mm).

Alignment

The Alignment element contains settings related to alignment and encoder calibration.

Alignment Child Elements

Element	Type	Description
InputTriggerEnabled	Bool	Enables digital input-triggered alignment operation.
Type	32s	Type of alignment operation:

Element	Type	Description
		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* below.
Bar	Section	See *Bar* below.
Plate	Section	See *Plate* on the 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
DataSource	32s	Data source of device output (read-only): 0 – Top
XSpacingCount	32u	Number of resampled points along X (read-only).
YSpacingCount	32u	Number of resampled points along Y (read-only).
ActiveArea	Region3D	Active area. (Contains min and max attributes for each element.)
TransformedDataRegion	Region3D	Active area after transformation (read-only).
PatternSequenceType	32s	G3 projection sequence for video mode. 0 – Normal 100 – Nine Lines
PatternSequenceType.options	32s (CSV)	List of available projection sequences.
PatternSequenceType.used	Bool	Whether or not the type can be selected.
PatternSequenceCount	32u	Number of frames in the active sequence (read-only)..
FrontCamera	Window	Front camera window (read-only).
BackCamera	Window	Back camera window (read-only).
BackCamera.used	Bool	Whether or not this field is used.
ExposureMode	32s	Exposure mode: 0 – Single exposure 1 – Multiple exposures
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).

Element	Type	Description
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.
SpacingIntervalType	32s	Spacing interval type: 0 – Maximum resolution 1 – Balanced 2 – Maximum speed 3 – Custom
SpacingIntervalType.used	Bool	Whether or not field is used.
Tracking	Section	See *Tracking* on the next page.
Material	Section	See *Material* on the next page.
Custom	Custom	Used by specialized sensors.

Region3D Child Elements

Element	Type	Description
X	64f	X start (mm).
Y	64f	Y start (mm).
Z	64f	Z start (mm).
Width	64f	X extent (mm).
Length	64f	Y extent (mm).
Height	64f	Z extent (mm).

Window Child Elements

Element	Type	Description
X	32u	X start (pixels).
Y	32u	Y start (pixels).

Element	Type	Description
Width	32u	X extent (pixels).
Height	32u	Y extent (pixels).

Tracking

Tracking Child Elements

Element	Type	Description
Enabled	Bool	Enables tracking.
Enabled.used	Bool	Whether or not this field is used.
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

Element	Type	Description
		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.
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

The SurfaceGeneration element contains settings related to surface generation.

This element is used by Gocator 2300 and 2880 series sensors.

SurfaceGeneration Child Elements

Element	Type	Description
Type	32s	Surface generation type:

Element	Type	Description
		0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational
FixedLength	Section	See *FixedLength* below.
VariableLength	Section	See *VariableLength* below.
Rotational	Section	See *Rotational* 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).

ProfileGeneration

The ProfileGeneration element contains settings related to profile generation.

This element is used by Gocator 1300 series sensors.

ProfileGeneration Child Elements

Element	Type	Description
Type	32s	Profile generation type: 0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational
FixedLength	Section	See *FixedLength* below.
VariableLength	Section	See *VariableLength* below.
Rotational	Section	See *Rotational* below.

FixedLength

FixedLength Child Elements

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

VariableLength

VariableLength Child Elements

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

Element	Type	Description
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
MinArea	64f	Minimum area (mm^2).
MinArea.min	64f	Minimum value of minimum area.
MinArea.max	64f	Maximum value of minimum area.
MinArea.used	Bool	Whether or not this field is used.
GapWidth	64f	Gap width (mm).
GapWidth.min	64f	Minimum gap width (mm).
GapWidth.max	64f	Maximum gap width (mm).
GapWidth.used	Bool	Whether or not this field is used.
GapLength	64f	Gap length (mm).
GapLength.min	64f	Minimum gap length (mm).
GapLength.max	64f	Maximum gap length (mm).
GapLength.used	Bool	Whether or not this field is used.
PaddingWidth	64f	Padding width (mm).
PaddingWidth.min	64f	Minimum padding width (mm).
PaddingWidth.max	64f	Maximum padding width (mm).
PaddingWidth.used	Bool	Whether or not this field is used.
PaddingLength	64f	Padding length (mm).
PaddingLength.min	64f	Minimum padding length (mm).
PaddingLength.max	64f	Maximum padding length (mm).
PaddingLength.used	Bool	Whether or not this field is used.
MinLength	64f	Minimum length (mm).
MinLength.min	64f	Minimum value of minimum length (mm).
MinLength.max	64f	Maximum value of minimum length (mm).
MinLength.used	Bool	Whether or not this field is used.
MaxLength	64f	Maximum length (mm).
MaxLength.min	64f	Minimum value of maximum length (mm).
MaxLength.max	64f	Maximum value of maximum length (mm).
MaxLength.used	Bool	Whether or not this field is used.

Element	Type	Description
FrameOfReference	32s	Part frame of reference: 0 – Sensor 1 – Scan 2 – Part
FrameOfReference.used	Bool	Whether or not this field is used.
FrameOfReference.value	32s	Actual value.
EdgeFiltering	Section	See *EdgeFiltering* below.

EdgeFiltering

EdgeFiltering Child Elements

Element	Type	Description
@used	Bool	Whether or not this section is used.
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).

PartMatching

The PartMatching element contains settings related to part matching.

PartMatching Child Elements

Element	Type	Description
Enabled	Bool	Enables part matching.
Enabled.used	Bool	Whether or not this field is used.
MatchAlgo	32s	Match algorithm. 0 – Edge points 1 – Bounding Box 2 – Ellipse
Edge	Section	See *Edge* on the next page.
BoundingBox	Section	See *BoundingBox* on the next page.
Ellipse	Section	See *Ellipse* on the next page.

Edge

Edge Child Elements

Element	Type	Description
ModelName	String	Name of the part model to use. Does not include the .mdl extension.
Acceptance/Quality/Min	64f	Minimum quality value for a match.

BoundingBox

BoundingBox Child Elements

Element	Type	Description
ZAngle	64f	Z rotation to apply to bounding box (degrees).
Acceptance/Width/Min	64f	Minimum width (mm).
Acceptance/Width/Max	64f	Maximum width (mm).
Acceptance/Length/Min	64f	Minimum length (mm).
Acceptance/Length/Max	64f	Maximum length (mm).

Ellipse

Ellipse Child Elements

Element	Type	Description
ZAngle	64f	Z rotation to apply to ellipse (degrees).
Acceptance/Major/Min	64f	Minimum major length (mm).
Acceptance/Major/Max	64f	Maximum major length (mm).
Acceptance/Minor/Min	64f	Minimum minor length (mm).
Acceptance/Minor/Max	64f	Maximum minor length (mm).

ToolOptions

The ToolOptions element contains a list of available tool types, their measurements, and settings for related information.

ToolOptions Child Elements

Element	Type	Description
<Tool Names>	Collection	A collection of tool name elements. An element for each tool type is present.

Tool Name Child Elements

Element	Type	Description
@displayName	String	Display name of the tool.
@isCustom	Bool	Reserved for future use.
MeasurementOptions	Collection	See *MeasurementOptions* on the next page

MeasurementOptions

MeasurementOptions Child Elements

Element	Type	Description
<Measurement Names>	Collection	A collection of measurement name elements. An element for each measurement is present.

Measurement Name Child Elements

Element	Type	Description
@displayName	String	Display name of the tool.
@minCount	32u	Minimum number of instances in a tool.
@maxCount	32u	Maximum number of instances in a tool.

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.

Surface Types

The following types are used by the various measurement tools.

Region3D

An element of type Region3D defines a rectangular area of interest in 3D.

Region3D Child Elements

Element	Type	Description
X	64f	Volume X position (mm).
Y	64f	Volume Y position (mm).
Z	64f	Volume Z position (mm).
Width	64f	Volume width (mm).
Length	64f	Volume length (mm).
Height	64f	Volume height (mm).

SurfaceFeature

An element of type SurfaceFeature defines the settings for detecting a feature within an area of interest.

SurfaceFeature Child Elements

Element	Type	Description
Type	32s	Setting to determine how the feature is detected within the area:

Element	Type	Description
		0 – Average (formerly Centroid 2d) 1 – Centroid (formerly Centroid 3d) 2 – X Min 3 – X Max 4 – Y Min 5 – Y Max 6 – Z Min 7 – Z Max 8 – Median
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Element for feature detection volume.

SurfaceRegion2d

An element of type SurfaceRegion2d defines a rectangular area of interest on the X-Y plane.

SurfaceRegion2d Child Elements

Element	Type	Description
X	64f	Setting for surface region X position (mm).
Y	64f	Setting for surface region Y position (mm).
Width	64f	Setting for region width (mm).
Length	64f	Setting for region length (mm).

Script

A Script element defines settings for a script measurement.

Script Child Elements

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

Output

Element	Type	Description
@id	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	32u	Measurement enabled.

SurfaceBoundingBox

A SurfaceBoundingBox element defines settings for a surface bounding box tool and one or more of its measurements.

SurfaceBoundingBox Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
ZRotationEnabled	Boolean	Setting to enable/disable rotation of bounding box
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
Measurements\X	Bounding Box tool measurement	X measurement.
Measurements\Y	Bounding Box tool measurement	Y measurement.
Measurements\Z	Bounding Box tool measurement	Z measurement.
Measurements\Width	Bounding Box tool measurement	Width measurement.
Measurements\Height	Bounding Box tool measurement	Height measurement.
Measurements\ZAngle	Bounding Box tool measurement	ZAngle measurement.
Measurements\GlobalX	Bounding Box tool measurement	Global X measurement.
Measurements\GlobalY	Bounding Box tool measurement	Global Y measurement.
Measurements\GlobalZAngle	Bounding Box tool measurement	Global Z Angle measurement.

Bounding Box Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).
Name	String	Measurement name.
Enabled	Boolean	Measurement enable state:

Element	Type	Description
		0 – Disable 1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfaceCsHole

A SurfaceCsHole element defines settings for a surface countersunk hole tool and one or more of its measurements.

SurfaceCsHole Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
NominalBevelAngle	64f	Nominal bevel angle (mm).
BevelAngleTolerance	64f	Bevel angle tolerance (mm).
NominalOuterRadius	64f	Nominal outer radius (mm).
OuterRadiusTolerance	64f	Outer radius tolerance (mm).
NominalInnerRadius	64f	Nominal inner radius (mm).
InnerRadiusTolerance	64f	Inner radius tolerance (mm).
BevelRadiusOffset	64f	Bevel radus offset (mm).
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable

Element	Type	Description
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions which are to be used
RefRegions	(Collection)	Reference regions. Contains 2 SurfaceRegion2D elements.
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for manual tilt correction angle X.
TiltYAngle	64f	Setting for manual tilt correction angle Y.
CurveFitEnabled	Boolean	Setting to enable/disable curve fitting: 0 – Disable 1 – Enable
CurveOrientation	64f	Setting for curve orientation angle.
Measurements\X	Countersunk Hole tool measurement	X measurement.
Measurements\Y	Countersunk Hole tool measurement	Y measurement.
Measurements\Z	Countersunk Hole tool measurement	Z measurement.
Measurements\OuterRadius	Countersunk Hole tool measurement	Outer Radius measurement.
Measurements\Depth	Countersunk Hole tool measurement	Depth measurement.
Measurements\BevelRadius	Countersunk Hole tool measurement	Bevel Radius measurement.
Measurements\BevelAngle	Countersunk Hole tool measurement	Bevel Angle measurement.
Measurements\XAngle	Countersunk Hole tool measurement	X Angle measurement.
Measurements\YAngle	Countersunk Hole tool measurement	Y Angle measurement.
Measurements\Radius	Countersunk Hole tool measurement	Radius measurement.

Countersunk Hole Tool Measurement

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	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfaceEllipse

A SurfaceEllipse element defines settings for a surface ellipse tool and one or more of its measurements.

SurfaceEllipse Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
Measurements\Major	Ellipse tool measurement	Major measurement.
Measurements\Minor	Ellipse tool measurement	Minor measurement.
Measurements\Ratio	Ellipse tool	Ratio measurement.

Element	Type	Description
	measurement	
Measurements\ZAngle	Ellipse tool measurement	ZAngle measurement.

Ellipse Tool Measurement

Element	Type	Description
@id	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	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfaceHole

A SurfaceHole element defines settings for a surface hole tool and one or more of its measurements.

SurfaceHole Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.

Element	Type	Description
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
NominalRadius	64f	Nominal radius (mm).
RadiusTolerance	64f	Radius tolerance (mm).
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used.
RefRegions	(Collection)	Reference regions. Contains two RefRegion elements of type SurfaceRegion2D.
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for manual tilt correction angle X.
TiltYAngle	64f	Setting for manual tilt correction angle Y.
Measurements\X	Hole tool measurement	X measurement
Measurements\Y	Hole tool measurement	Y measurement
Measurements\Z	Hole tool measurement	Z measurement
Measurements\Radius	Hole tool measurement	Radius measurement

Hole Tool Measurement

Element	Type	Description
@id	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:

Element	Type	Description
		0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfaceOpening

A SurfaceOpening element defines settings for a surface opening tool and one or more of its measurements.

SurfaceOpening Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Type	32s	Type of the opening: 0 – Rounded 1 – Slot
NominalWidth	64f	Nominal width (mm).
NominalLength	64f	Nominal length (mm).
NominalAngle	64f	Nominal angle (degrees).
NominalRadius	64f	Nominal radius (mm).
WidthTolerance	64f	Radius tolerance (mm).
LengthTolerance	64f	Length tolerance (mm).
AngleTolerance	64f	Angle tolerance (degrees).
PartialDetectionEnabled	Boolean	Setting to enable/disable partial detection: 0 – Disable 1 – Enable

Element	Type	Description
RegionEnabled	Boolean	Setting to enable/disable region: 0 – Disable 1 – Enable
Region	Region3D	Measurement region.
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used.
RefRegions	(Collection)	Reference regions. Contains two RefRegion elements of type SurfaceRegion2D.
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for manual tilt correction angle X.
TiltYAngle	64f	Setting for manual tilt correction angle Y.
Measurements\X	Opening tool measurement	X measurement.
Measurements\Y	Opening tool measurement	Y measurement.
Measurements\Z	Opening tool measurement	Z measurement.
Measurements\Width	Opening tool measurement	Width measurement.
Measurements\Length	Opening tool measurement	Length measurement.
Measurements\Angle	Opening tool measurement	Angle measurement.

Opening Tool Measurement

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

Element	Type	Description
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfacePlane

A SurfacePlane element defines settings for a surface plane tool and one or more of its measurements.

SurfacePlane Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
RegionsEnabled	Boolean	Setting to enable/disable regions: 0 – Disable 1 – Enable
RegionCount	32s	Count of the regions.
Regions	(Collection)	Measurement regions. Contains one or two Region elements of type Region3D.
Measurements\XAngle	Plane tool measurement	XAngle measurement.
Measurements\YAngle	Plane tool measurement	YAngle measurement.
Measurements\ZOffset	Plane tool measurement	ZOffset measurement.

Plane Tool Measurement

Element	Type	Description
id (attribute)	32s	Measurement ID. Optional (measurement disabled if not set).

Element	Type	Description
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	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfacePosition

A SurfacePosition element defines settings for a surface position tool and one or more of its measurements.

SurfacePosition Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
Feature	SurfaceFeature	Measurement feature.
Measurements\X	Position tool measurement	X measurement.
Measurements\Y	Position tool measurement	Y measurement.
Measurements\Z	Position tool measurement	Z measurement.

Position Tool Measurement

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	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.

SurfaceStud

A SurfaceStud element defines settings for a surface stud tool and one or more of its measurements.

SurfaceStud Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StudRadius	64f	Radius of stud (mm).
StudHeight	64f	Height of stud (mm).
BaseHeight	64f	Height of stud's base.
TipHeight	64f	Height of stud's tip.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.

Element	Type	Description
RefRegionsEnabled	Boolean	Setting to enable/disable reference regions: 0 – Disable 1 – Enable
RefRegionCount	32s	Count of the reference regions that are to be used.
RefRegions	(Collection)	Reference regions. Contains two RefRegion elements of type SurfaceRegion2D.
AutoTiltEnabled	Boolean	Setting to enable/disable tilt correction: 0 – Disable 1 – Enable
TiltXAngle	64f	Setting for manual tilt correction angle X.
TiltYAngle	64f	Setting for manual tilt correction angle Y.
Measurements\BaseX	Stud tool measurement	BaseX measurement.
Measurements\BaseY	Stud tool measurement	BaseY measurement.
Measurements\BaseZ	Stud tool measurement	BaseZ measurement.
Measurements\TipX	Stud tool measurement	TipX measurement.
Measurements\TipY	Stud tool measurement	TipY measurement.
Measurements\TipZ	Stud tool measurement	TipZ measurement.
Measurements\Radius	Stud tool measurement	Radius measurement.

Stud Tool Measurement

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

Element	Type	Description
		1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
RadiusOffset *(Radius measurement only)*	64f	Radius offset of the stud.

SurfaceVolume

A SurfaceVolume element defines settings for a surface volume tool and one or more of its measurements.

SurfaceVolume Child Elements

Element	Type	Description
Name	String	Tool name.
Source	32s	Surface source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.
Anchor\Y	String (CSV)	The Y measurements (IDs) used for anchoring.
Anchor\Y.options	String (CSV)	The Y measurements (IDs) available for anchoring.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
RegionEnabled	Boolean	Setting to enable/disable region.
Region	Region3D	Measurement region.
Measurements\Volume	Volume tool measurement	Volume measurement.
Measurements\Area	Volume tool measurement	Area measurement.
Measurements\Thickness	Volume tool measurement	Thickness measurement.

Volume Tool Measurement

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

Element	Type	Description
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Location *(Thickness measurement only)*	32s	Measurement type: 0 – Maximum 1 – Minimum 2 – 2D Centroid 3 – 3D Centroid 4 – Average 5 – Median

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 (Surface Hole X) will be sent to output.

```
<SurfaceHole>   ...
  <Measurements>
    <X id="2">   ...
    <Y 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 Surfaces element shows that only two sources are available (see the table below for the meanings of these values). The value in this element—0—indicates that only data from that source will be sent to output.

```
<Output>
  <Ethernet>
     ...
    <Ranges options=""/>
    <Profiles options=""/>
    <Surfaces options="0,1">0</Surfaces>
    ...
```

Ethernet Child Elements

Element	Type	Description
Protocol	32s	Ethernet protocol: 0 – Gocator 1 – Modbus 2 – EtherNet/IP 3 – ASCII
Ascii	Section	See *Ascii* on the next page.
EIP	Section	See *EIP* on page 204.
Modbus	Section	See *Modbus* on page 204.
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).
Profiles	32s (CSV)	Selected profile sources: 0 – Top 1 – Bottom 2 – Top left

Element	Type	Description
		3 – Top right
Profiles.options	32s (CSV)	List of available profile sources (see above).
Surfaces	32s (CSV)	Selected surface sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right
Surfaces.options	32s (CSV)	List of available surface 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).
ProfileIntensities	32s (CSV)	Selected profile intensity sources. 0 – Top 1 – Bottom 2 – Top left 3 – Top right
ProfileIntensities.options	32s (CSV)	List of available profile intensity sources (see above).
SurfaceIntensities	32s (CSV)	Selected surface intensity sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right
SurfaceIntensities.options	32s (CSV)	List of available surface 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
ControlPort	32u	Control service port number.

Element	Type	Description
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 EtherNet/IP output buffering.
EndianOutputType	32s	Endian output type: 0 – Big endian 1 – Little endian
ImplicitOutputEnabled	Bool	Enables Implict (I/O) Messaging.
ImplicitTriggerOverride	32s	Override requested trigger type by client: 0 – No override 1 – Cyclic 2 – Change of State

Modbus

Modbus Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables Modbus output buffering.

Digital0 and Digital1

The Digital0 and Digital1 elements define 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 4 – Exposure
SignalType	32s	Signal type:

Element	Type	Description
		0 – Pulse 1 – Continuous
ScheduleEnabled	Bool	Enables scheduling.
PulseWidth	64f	Pulse width (µs).
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).

Element	Type	Description
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.
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	See *Selcom* below.
Ascii	Section	See *Ascii* on the next page.
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

Element	Type	Description
		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.

Transform

The transformation component 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

You can access the Transform component of the active job as an XML file, either using path notation, via "_live.job/transform.xml", or directly via "_live.tfm".

You can access the Transform component in user-created job files in non-volatile storage, for example, "productionRun01.job/transform.xml". You can only access transformations in user-created job files using path notation.

See the following sections for the elements contained in this component.

Transformation 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>
    </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>
    </Device>
  </Devices>
</Transform>
```

The Transform element contains the alignment record for the sensor.

Transform Child Elements

Element	Type	Description
@version	32u	Transform version (100).
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 Child Elements

Element	Type	Description
@role	32s	Role of device described by this section: 0 – Main 1 – Buddy
X	64f	Translation on the X axis (mm).
Y	64f	Translation on the Y axis (mm).
Z	64f	Translation on the Z axis (mm).
XAngle	64f	Rotation around the X axis (degrees).
YAngle	64f	Rotation around the Y axis (degrees).
ZAngle	64f	Rotation around the Z axis (degrees).

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

Part Models

Part models represent models created using the part matching feature.

You can access a model in the active job using path notation. For example, to access a model called scan.mdl, use "_live.job/scan.mdl".

You can access part models in user-created job files in non-volatile storage, for example, "productionRun01.job/model1.mdl". You can only access part models in user-created job files using path notation.

See the following sections for the elements contained in a model.

Part models contain the following subcomponents. You can access the subcomponents using path notation, for example, "productionRun01.job/myModel.mdl/config.xml".

Part Model Child Elements

Element	Type	Description
Configuration	config.xml	Model configuration XML. It is always present. (See *Configuration* on the next page.)
Edge Points	edge-height-top	Edge points for the top heightmap. (See *Edge Points* below.)
Edge Points	edge-height-bottom	Edge points for the bottom heightmap.
Edge Points	edge-intensity-top	Edge points for the top intensity map.
Edge Points	edge-intensity-bottom	Edge points for the bottom intensity map.

The edge points file exists only when the model contains the source data for the edge points.

Edge Points

Edge Points Data

Field	Type	Offset	Description
id	16s	0	Sender ID -1 – Part matching
source	8s	2	Source 0 – Model 1 – Target
imageType	8s	3	Image type 0 – Height map 1 – Intensity map
imageSource	8s	4	Image source 0 – Top 1 – Bottom
width	32u	5	Width of model space, in units of xScale

Field	Type	Offset	Description
length	32u	9	Length of model space, un units of yScale
xScale	32u	13	X scale (nm)
yScale	32u	17	Y scale (nm)
xOffset	32s	21	X offset (µm)
yOffset	32s	25	Y offset µm
zAngle	32s	29	Z rotation (microdegrees)
pointCount	32u	33	Number of edge points
points[pointCount]	(32u, 32u)	37	Edge points collection. Each point is a tuple of x and y values, in units of xScale and yScale, respectively.

Configuration

Delete this text and replace it with your own content.

Configuration Child Elements

Element	Type	Description
@version	32u	Major version (1).
@versionMinor	32u	Minor version (0).
Edges	Collection	Collection of Edge items (described below).
EdgeSensitivity	64f	Sensitivity recorded during model edges generation (read-only).
TransformedDataRegion	Region3d	Data region of the model.
ZAngle	64f	Additional rotation applied to the model (degrees).
TargetEdgeSensitivity	64f	Sensitivity used to generate target edges.
ImageType	32s	Selects type of image used to generate edges: 0 – Height map 1 – Intensity map
ImageType.options	32s (CSV)	List of available image types.

Protocols

Gocator supports protocols for communicating with sensors over Ethernet (TCP/IP) and serial output. For a protocol to output data, it must be enabled and configured in the active job.

Protocols Available over Ethernet

- Gocator
- Modbus
- EtherNet/IP
- ASCII

Protocols Available over Serial

- ASCII

Gocator Protocol

This section describes the TCP and UDP commands and data formats used by a client computer to communicate with Gocator sensors using the Gocator protocol. It also describes the connection types (Discovery, Control, Upgrade, Data, and Health), and data types. The protocol enables the client to:

- Send commands to run sensors, provide software triggers, read/write files, etc.
- Receive data, health, and diagnostic messages.
- Upgrade firmware.

> The Gocator 4.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

To use the Gocator protocol, it must be enabled and configured in the active job.

> Gocator sensors send UDP broadcasts over the network over the Internal Discovery channel (port 2016) at regular intervals during operation to perform peer discovery.

> The Gocator SDK provides open source C language libraries that implement the network commands and data formats defined in this section. For more information, see *Software Development Kit* on page 276.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 145.

For information on job file structures (for example, if you wish to create job files programmatically), see *Job Files* on page 169.

Data Types

The table below defines the data types and associated type identifiers used in this section.

All values except for IP addresses are transmitted in little endian format (least significant byte first) unless stated otherwise. The bytes in an IP address "a.b.c.d" will always be transmitted in the order a, b, c, d (big endian).

Data Types

Type	Description	Null Value
char	Character (8-bit, ASCII encoding)	-
byte	Byte.	-
8s	8-bit signed integer.	-128
8u	8-bit unsigned integer.	255U
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	-

Commands

The following sections describe the commands available on the Discovery (page 213), Control (page 215), and Upgrade (page 242) channels.

When a client sends a command over the Control or Upgrade channel, the sensor sends a reply whose identifier is the same as the command's identifier. The identifiers are listed in the tables of each of the commands.

Status Codes

Each reply on the Discovery, Control, and Upgrade 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.

Label	Value	Description
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.
Simulation Buffer Empty	-992	The simulation buffer is empty.

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)

Use the Get Address and Set Address commands to modify a sensor's network configuration. These 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. For a list of status codes, see *Commands* on page 212.
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).
signature	64u	16	Message signature (0x0000504455494D4C).
deviceId	64u	24	Serial number of the device whose address information is queried. 0 selects all devices.
dhcpEnabled	64u	32	0 – Disabled 1 – Enabled
reserved[4]	byte	40	Reserved.
address[4]	byte	44	The IP address in left to right order.
reserved[4]	byte	48	Reserved.
subnetMask[4]	byte	52	The subnet mask in left to right order.
reserved[4]	byte	56	Reserved.
gateway[4]	byte	60	The gateway address in left to right order.
reserved[4]	byte	64	Reserved.
reserved[4]	byte	68	Reserved.

Reply

Field	Type	Offset	Description
length	64u	0	Reply length.
type	64s	8	Reply type (0x1002).
status	64s	16	Operation status. For a list of status codes, see *Commands* on page 212.
signature	64u	24	Message signature (0x0000504455494D4C).
deviceId	64u	32	Serial number.

Control Commands

A client sends control commands for most operations over the Control TCP channel (port 3190).

The Control channel and the Upgrade channel (port 3192) 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. For more information on Upgrade commands, see *Upgrade Commands* on page 242.

States

A Gocator system can be in one of two states: Ready or Running. The client sends the Start and Stop control commands to change the system's current state to Running and Ready, respectively. The sensor can also be configured to boot in either the Ready or Running state, by enabling or disabling autostart, respectively, using the Set Auto Start Enabled command.

In the Ready state, a sensor can be configured. In the Running state, a sensor responds to input signals, performs measurements, drives its outputs, and sends data messages to the client. Disconnecting the command channel changes the sensor from the Running state to the Ready state.

The state of the sensor can be retrieved using the Get States or Get System Info command.

Progressive Reply

Some commands send replies progressively, as multiple messages. This allows the sensor to stream data without buffering it first, and allows the client to obtain progress information on the stream.

A progressive reply begins with an initial, standard reply message. If the *status* field of the reply indicates success, the reply is followed by a series of “continue” reply messages.

A continue reply message contains a block of data of variable size, as well as status and progress information. The series of continue messages is ended by either an error, or a continue message containing 0 bytes of data.

Protocol Version

The Protocol Version command returns the 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. For a list of status codes, see *Commands* on page 212.
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. For a list of status codes, see *Commands* on page 212.
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).

Field	Type	Offset	Description
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. For a list of status codes, see *Commands* on page 212.

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 on page 242). Firmware upgrade files are available from the downloads section under the support tab on the LMI web site. For more information on getting the latest firmware, see *Firmware Upgrade* on page 51.

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. For a list of status codes, see *Commands* on page 212.
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.

Sensor Info

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 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 215.
role	32s	48	Sensor role 0 – Main

Get States

The Get States command returns various 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. For a list of status codes, see *Commands* on page 212.
count	32u	10	Number of state variables.
sensorState	32s	14	Sensor state 0 – Ready 1 – Running For more information on states, see *Control Commands* on page 215.
loginState	32s	18	Device login state 0 – No user 1 – Administrator 2 – Technician
alignmentReference	32s	22	Alignment reference 0 – Fixed

Field	Type	Offset	Description
			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 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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
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 (a .job file, a component of a job file, or another type of file, see *Job Files* on page 169).

To make a job active (to load it), copy a saved job to "_live.job". To "save" the active job, copy from "_ live.job" to another file.

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. For a list of status codes, see *Commands* on page 212.

Read File

Downloads a file from the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job Files* on page 169).

To download the live configuration, pass "_live.job" in the *name* field. To read the configuration of the live configuration only, pass "_live.job/config.xml" in the *name* field.

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. For a list of status codes, see *Commands* on page 212.
length	32u	10	File length
data[length]	byte	14	File contents

Write File

The Write File command uploads a file to the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job Files* on page 169).

To make a job file live, write to "_live.job". Except for writing to the live file, 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. For a list of status codes, see *Commands* on page 212.

Delete File

The Delete File command removes a file from the connected sensor (a .job file, a component of a job file, or another type of file; for more information, see *Job Files* on page 169).

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. For a list of status codes, see *Commands* on page 212.

Get Default Job

The Get Default Job command gets the name of the job 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. For a list of status codes, see *Commands* on page 212.
name[64]	char	10	The file name (null-terminated) of the job that will be loaded at boot time.

Set Default Job

The Set Default Job command sets the job 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) of the job that will be loaded at boot time.

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. For a list of status codes, see *Commands* on page 212.

Get Loaded Job

The Get Loaded Job command returns the name and modified status of the currently loaded file.

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. For a list of status codes, see *Commands* on page 212.
fileName[64]	char	10	Name of the currently loaded job.
changed	8u	74	Whether or not the currently loaded job has been changed (1: yes; 0: no).

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. For a list of status codes, see *Commands* on page 212.
reference	32s	10	Alignment reference 0 – Fixed 1 – Dynamic

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
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. For a list of status codes, see *Commands* on page 212.
encoder	64s	10	Current encoder position, in ticks.

Reset Encoder

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

The encoder value can be reset only when the encoder is connected directly to a sensor. When the encoder is connected to the master, the value cannot be reset via this command.

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. For a list of status codes, see *Commands* on page 212.

Start

The Start command starts the sensor system (system enters the Running state). For more information on states, see *Control Commands* on page 215.

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. For a list of status codes, see *Commands* on page 212.

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). For more information on states, see *Control Commands* on page 215.

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. For a list of status codes, see *Commands* on page 212.

Stop

The Stop command stops the sensor system (system enters the Ready state). For more information on states, see *Control Commands* on page 215.

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. For a list of status codes, see *Commands* on page 212.

Get Auto Start Enabled

The Get Auto Start Enabled command returns whether the system automatically starts after booting.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452C).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x452C).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.
enable	8u	10	0: disabled 1: enabled

Set Auto Start Enabled

The Set Auto Start Enabled command sets whether the system automatically starts after booting (enters Running state; for more information on states, see *Control Commands* on page 215).

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x452B).
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 (0x452B).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
opId	32s	10	Operation ID. Use this ID to correlate the command/reply on the Command channel with the correct Alignment Result message on the Data channel. A unique ID is returned each time the client uses this command.

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. For a list of status codes, see *Commands* on page 212.
opId	32s	10	Operation ID. Use this ID to correlate the command/reply on the Command channel with the correct Exposure Calibration Result message on the Data channel. A unique ID is returned each time the client uses this command.

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. For a list of status codes, see *Commands* on page 212.

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* on page 148 for more information on setting up 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) when or position (µm) at which the digital output event should happen. The target value is ignored if ScheduleEnabled is set to false. (**Scheduled** is unchecked in **Digital** in the **Output** panel.) The output will be triggered immediately.
value	8u	16	Specifies the target state:

Field	Type	Offset	Description
			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. For a list of status codes, see *Commands* on page 212.

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* on page 151 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).
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 ScheduleEnabled is set to false. (**Scheduled** is unchecked in **Analog** in the **Output** panel.) The output will be triggered immediately.
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. For a list of status codes, see *Commands* on page 212.

> 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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
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. For a list of status codes, see *Commands* on page 212.

Restore Factory

The Restore Factory command restores the connected sensor to factory default settings.

> The command erases the non-volatile memory of the main device.

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).

Field	Type	Offset	Description
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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
enable	8u	10	0: disabled; 1: enabled.

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

Get Playback Source

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

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. For a list of status codes, see *Commands* on page 212.
source	32s	10	Source 0 – Live 1 – Replay buffer

Set Playback Source

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

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
bufferValid	8u	10	Whether or not the buffer is valid.

> A reply status of -996 means that the current configuration (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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

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

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. For a list of status codes, see *Commands* on page 212.

Field	Type	Offset	Description
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. For a list of status codes, see *Commands* on page 212.

Clear Log

The Clear Log command clears the sensor's log.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101D).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101D).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.

The command returns after the scan has been captured and transmitted.

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. For a list of status codes, see *Commands* on page 212.

The command returns after the scan has been captured and transmitted.

Create Model

The Create Model command creates a new part model from the active simulation scan.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4602).
modelName[64]	char	6	Name of the new model (without .mdl extension)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4602).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

Detect Edges

The Detect Edges command detects and updates the edge points of a part model.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4604).
modelName[64]	char	6	Name of the model (without .mdl extension)
sensitivity	16s	70	Sensitivity (in thousandths)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4604).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

Read File (Progressive)

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

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

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.

Initial 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. For a list of status codes, see *Commands* on page 212.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in byes.
data[size]	byte	22	Chunk data.

Export CSV (Progressive)

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

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4507).
name[64]	char	6	Source file name.

Initial 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. For a list of status codes, see *Commands* on page 212.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in byes.
data[size]	byte	22	Chunk data.

> Only the current surface scan, as determined by the playback position, is exported to the CSV stream.

Export Bitmap (Progressive)

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

This command returns an initial reply, followed by a series of "continue" replies if the initial reply's *status* field indicates success. The continue replies contain the actual data, and have 0x5000 as their identifier.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4508).
name[64]	char	6	Source file name.

Initial 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. For a list of status codes, see *Commands* on page 212.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.

Continue Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x5000).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

Field	Type	Offset	Description
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in byes.
data[size]	byte	22	Chunk data.

Upgrade Commands

A client sends firmware upgrade commands over the Upgrade TCP channel (port 3192).

The Control channel (port 3190) and the Upgrade channel 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. For more information on Control commands, see *Control Commands* on page 215.

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 is updated when breaking changes are made to the Gocator Protocol.

Start Upgrade

The Start Upgrade command begins a firmware upgrade for the sensors in a system. All sensors 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).
length	32u	6	Length of the upgrade package (bytes).
data[length]	byte	10	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. For a list of status codes, see *Commands* on page 212.

Start Upgrade Extended

The Start Upgrade Extended command begins a firmware upgrade for the sensors in a system. All sensors 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 (0x0003).
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 (0x0003).
status	32s	6	Reply status. For a list of status codes, see *Commands* on page 212.

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. For a list of status codes, see *Commands* on page 212.
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. For a list of status codes, see *Commands* on page 212.
length	32u	10	Length of the log (bytes).
log[length]	char	14	Log content.

Results

The following sections describe the results (data and health) that Gocator sends.

Data Results

A client can receive data messages from a Gocator sensor by connecting to the Data TCP channel (port 3196).

The Data channel and the Health channel (port 3194) can be connected at the same time. The sensor accepts multiple connections on each port. For more information on the Health channel, see *Health Results* on page 250.

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, containing *size* and *control* fields, 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. (See individual data result sections.)

GDP messages are always sent in groups. The Last Message flag in the *control* field is used to indicate the final message in 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. For this message, set to 1.
count (C)	32u	6	Count of stamps in this message.
size	16u	10	Stamp size, in bytes (min: 56, current: 56).

Field	Type	Offset	Description
source	8u	12	Source (0 – Main).
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. For this message, set to 2.
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 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

Field	Type	Offset	Description
			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.)

Surface

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. For this message, set to 8.
attributeSize	16u	6	Size of attributes, in bytes (min: 40, current: 40).
length (L)	32u	8	Surface length (rows).
length (W)	32u	12	Surface width (columns).
xScale	32u	16	X scale (nm).
yScale	32u	20	Y scale (nm).
zScale	32u	24	Z scale (nm).
xOffset	32s	28	X offset (µm).
yOffset	32s	32	Y offset (µm).
zOffset	32s	36	Z offset (µm).
source	8u	40	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	41	Exposure (ns).
rotation	32s	45	Rotation (microdegrees).
reserved[3]	8u	49	Reserved.
ranges[L][W]	16s	50	Surface ranges.

Surface 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. For this message, set to 9.
attributeSize	16u	6	Size of attributes, in bytes (min: 32, current: 32).
length (L)	32u	8	Surface length (rows).
width (W)	32u	12	Surface width (columns).
xScale	32u	16	X scale (nm).
yScale	32u	20	Y scale (nm).
xOffset	32s	24	X offset (µm).
yOffset	32s	28	Y offset (µm).
source	8u	32	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	33	Exposure (ns).
reserved[3]	8u	37	
intensities[H][W]	8u	40	Surface intensities.

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. For this message, set to 10.
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

Field	Type	Offset	Description
			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. For this message, set to 11.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 8).
opId	32u	8	Operation ID.
status	32s	12	Operation status. 1 – OK 0 – General failure -1 – No data in the field of view for stationary alignment -2 – No profiles with sufficient data for line fitting for travel alignment -3 – Invalid target detected. Examples include: - Calibration disk diameter too small. - Calibration disk touches both sides of the field of view. - Too few valid data points after outlier rejection. -4 – Target detected in an unexpected position. -5 – No reference hole detected in bar alignment. -6 – No change in encoder value during travel calibration -988 – User aborted -993 – Timed out -997 – Invalid parameter

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. For this message, set to 12.

Field	Type	Offset	Description
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).

Edge Match 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. For this message, set to 16.
decision	byte	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (µm).
yOffset	32s	11	Target y offset in model space (µm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
quality	32s	19	Match quality (thousandth).
qualityDecision	byte	24	Quality match decision.
reserved[2]	byte	25	Reserved.

Bounding Box Match 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. For this message, set to 17.
decision	byte	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (µm).
yOffset	32s	11	Target y offset in model space (µm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
width	32s	19	Width axis length (µm)
widthDecision	8u	23	Width axis decision.
length	32s	24	Length axis length (µm)
lengthDecision	8u	28	Length axis decision.

Ellipse Match Result

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

Field	Type	Offset	Description
			Bits 0-14: Message type identifier. For this message, set to 18.
decision	byte	6	Overall match decision.
xOffset	32s	7	Target x offset in model space (µm).
yOffset	32s	11	Target y offset in model space (µm).
zAngle	32s	15	Target z rotation in model space (microdegrees).
minor	32s	19	Minor axis length (µm)
minorDecision	8u	23	Minor axis decision.
major	32s	24	Major axis length (µm)
majorDecision	8u	28	Major axis decision.

Health Results

A client can receive health messages from a Gocator sensor by connecting to the Health TCP channel (port 3194).

The Data channel (port 3196) and the Health channel can be connected at the same time. The sensor accepts multiple connections on each port. For more information on the Data channel, see *Data Results* on page 244.

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, containing *size* and *control* fields, 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. (See individual data result sections.)

GDP messages are always sent in groups. The Last Message flag in the *control* field is used to indicate the final message in 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. Always 0.
count (C)	32u	6	Count of indicators in this message.

Field	Type	Offset	Description
source	8u	10	Source (0 – Main, 1 – Buddy).
reserved[3]	8u	11	Reserved
indicators[C]	Indicator	14	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	32u	0	Unique indicator identifier (see below).
instance	32u	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).
Laser safety status	1010	-	0 if laser is disabled; 1 if enabled.
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).
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.

Indicator	Id	Instance	Value
			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 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.
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

Indicator	Id	Instance	Value
			transmitted.
Ethernet Drops	21005	-	Number of dropped Ethernet packets.
Digital Output Pass	21006	Output Index	Number of pass digital output pulse.
Digital Output Fail	21007	Output Index	Number of fail digital output pulse.
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 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 Protocol

Modbus is designed to allow industrial equipment such as Programmable Logic Controllers (PLCs), sensors, and physical input/output devices to communicate over an Ethernet network.

Modbus 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 lets the client:

- Switch jobs.
- Align and run sensors.
- Receive measurement results, sensor states, and stamps.

To use the Modbus protocol, it must be enabled and configured in the active job.

> The Gocator 4.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

If buffering is enabled with the Modbus protocol, the PLC must read the Buffer Advance output register (see on page 257) to advance the queue before reading the measurement results.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 145.

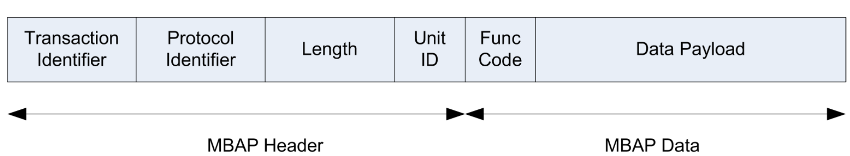
Concepts

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 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 closes 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.
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* on the next page for detailed descriptions.
900 - 999	Stamps	RO	Return stamps associated with each surface. See *State* on the next page for detailed descriptions.
1000 - 1060	Measurements & Decisions	RO	20 measurement and decision pairs. See *Measurement Registers* on page 258 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.

Value	Name	Description
3	Align (moving target)	Start alignment process and also calibrate encoder resolution. State 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 Name 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.

In Surface mode, the stamps are updated after each surface has been 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.

In Surface mode, the measurement results are updated after each discrete part has been processed.

Measurement Register Map

Register Address	Name	Type	Description
1000	Measurement 0 High	32s	Measurement value in µm (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 jobs.
- Align and run sensors.
- Receive sensor states, stamps, and measurement results.

To use the EtherNet/IP protocol, it must be enabled and configured in the active job.

> The Gocator 4.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 145.

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.

Gocator supports all required objects, such as the Identity object, TCP/IP object, and Ethernet Link object. In addition, assembly objects are used for sending sensor and sample data and receiving commands. There are three assembly objects: the command assembly (32 bytes), the sensor state assembly (100 bytes), and the sample state assembly object (380 bytes). The data attribute (0x03) of the assembly objects is a byte array containing information about the sensor. The data attribute can be accessed with the GetAttribute and SetAttribute commands.

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
Attribute Number	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

Value	Name	Description
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.
4	Clear Alignment	Clear the alignment.
5	Load Job	Load the job. Set bytes 1-31 to the file name (one character per byte, including the extension).

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
Instance	0x320
Attribute Number	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

Byte	Name	Type	Description
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
Attribute Number	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
14-17	Temperature	32u	Sensor temperature in degrees Celsius * 1000 (centidegrees)
18-25	Position	64s	Encoder position
26-33	Time	64u	Time
34-41	Frame Counter	64u	Frame counter
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 µm (0x80000000 if

Byte	Name	Type	Description
			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
...	...		
375-378	Measurement 59	32s	Measurement value in µm (0x80000000 if invalid)
379	Decision 59	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

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.

In Surface mode, the measurement results are updated after each discrete part has been processed. If buffering is enabled in the Ethernet Output panel, reading the Extended Sample State Assembly Object automatically advances the buffer. See See *Ethernet Output* on page 145 for information on the **Output** panel.

ASCII Protocol

This section describes the ASCII protocol. The ASCII protocol is available over either serial output or Ethernet output. Over serial output, communication is asynchronous (measurement results are automatically sent on the Data channel when the sensor is in the running state and results become available). Over Ethernet, communication can be asynchronous or use polling. For more information on polling commands, see

The protocol communicates using ASCII strings. The output result format from the sensor is user-configurable.

To use the ASCII protocol, it must be enabled and configured in the active job.

> The Gocator 4.x firmware uses mm, mm^2, mm^3, and degrees as standard units. In all protocols, values are scaled by 1000, as values in the protocols are represented as integers. This results in effective units of mm/1000, mm^2/1000, mm^3/1000, and deg/1000 in the protocols.

For information on configuring the protocol with the Web interface (when using the protocol over Ethernet), see *Ethernet Output* on page 145.

For information on configuring the protocol with the Web interface (when using the protocol over Serial), see *Serial Output* on page 153.

Connection Settings

Ethernet Communication

With Ethernet ASCII output, you can set the connection port numbers of the three channels used for communication (Control, Data, and Health):

Ethernet Ports for ASCII

Name	Description	Default Port
Control	To send commands to control the sensor.	8190
Data	To retrieve measurement output.	8190
Health	To retrieve specific health indicator values.	8190

Channels can share the same port or operate on individual ports. The following port numbers are reserved for Gocator internal use: 2016, 2017, 2018, and 2019. Each port can accept multiple connections, up to a total of 16 connections for all ports.

Serial Communication

Over serial, Gocator ASCII communication uses the following connection settings:

Serial Connection Settings for ASCII

Parameter	Value
Start Bits	1
Stop Bits	1
Parity	None
Data Bits	8
Baud Rate (b/s)	115200
Format	ASCII
Delimiter	CR

Up to 16 users can connect to the sensor for ASCII interfacing at a time. Any additional connections will remove the oldest connected user.

Polling Operation Commands (Ethernet Only)

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, a client can:

- Switch to a different job.
- Align, run, and trigger sensors.
- Receive sensor states, health indicators, stamps, and measurement results

Gocator sends Control, Data, and Health messages over separate channels. The Control channel is used for commands such as starting and stopping the sensor, loading jobs, and performing alignment (see *Control Commands* on the next page).

The Data channel is used to receive and poll for measurement results. When the sensor receives a Result command, it will send the latest measurement results on the same data channel that the request is received on. See *Data Commands* on page 271 for more information.

The Health channel is used to receive health indicators (see *Health Commands* on page 273).

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

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:

```
Command: Start
Reply: OK
Command: Start,1000000
Reply: OK
Command: Start
Reply: 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:

```
Command: Stop
Reply: 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 on page 54).

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:

```
Command: Trigger
Reply: OK
Command: Trigger,1000000
Reply: OK
```

LoadJob

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 no job is loaded. ".job" is appended if the filename does not have an extension.
Reply	OK or ERROR, <Error Message>

Examples:

```
Command: LoadJob,test.job
Reply: OK,test.job loaded successfully
Command: LoadJob
Reply: OK,test.job
Command: LoadJob,wrongname.job
Reply: ERROR, failed to load wrongname.job
```

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:

```
Command: Stamp
Reply: OK,Time,9226989840,Encoder,0,Frame,6
Command: Stamp,frame
Reply: OK,6
```

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:

```
Command: StationaryAlignment
Reply: OK
Command: StationaryAlignment
Reply: 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:

```
Command: ClearAlignment
Reply: 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 ','.

Result

The 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
```

Value

The 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
```

Decision

The 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 ','.

Health

The Health command retrieves health indicators. See *Health Results* on page 250 for details on health indicators.

Formats

Message	Format
Command	Health,health indicator ID.Optional 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 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.
```

Standard Result Format

Gocator can send measurement results either 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 on page 244).
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.
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.

Format Value	Explanation
	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

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 Gocator series, 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 rewrite the application with the SDK version corresponding to the sensor firmware in use.

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_3x00\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

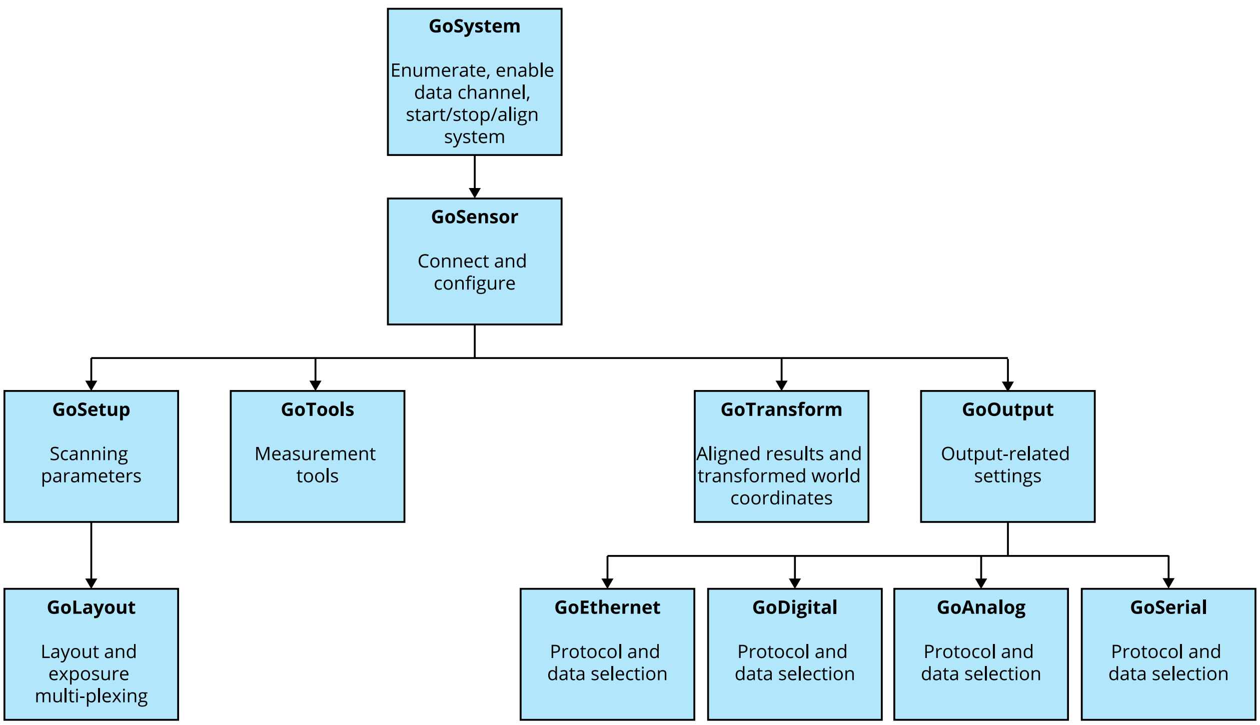
All sample projects use the environment variable *GO_SDK_4*. The environment variable should point to the *GO_SDK* directory, for example, *C:\14400-4.0.9.156_SOFTWARE_GO_SDK\GO_SDK*.

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 in http://lmi3d.com/sites/default/files/APPNOTE_Gocator_4.x_Multi_Sensor_Guide.zip 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.

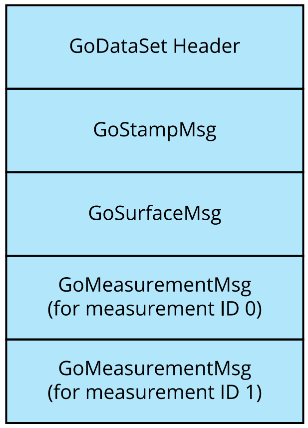
Refer to the *GoSdkSamples* sample code for examples of acquiring data using these data types.

See *Setup and Locations* on page 276 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 (surfaces), 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 surface 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* on page 278 for more information on value types.

The following table lists the decisions that can be returned.

Measurement Decisions

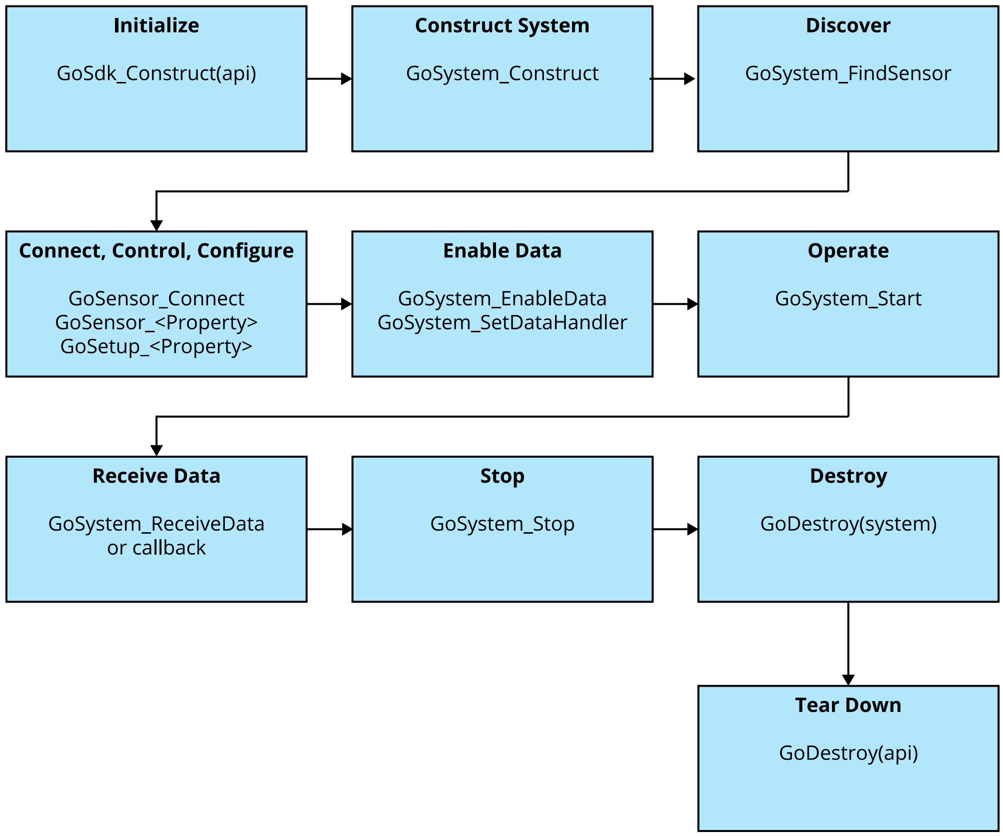
Decision	Description
1	The measurement value is between the maximum and minimum decision values. This is a pass decision.
0	The measurement value is outside the maximum and minimum. This is a fail decision.
-1	The measurement is invalid (for example, the target is not within range). Provides the reason 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.

Operation Workflow

Applications created using the SDK typically use the following programming sequence:



> See *Setup and Locations* on page 276 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* on page 277 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_ 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.

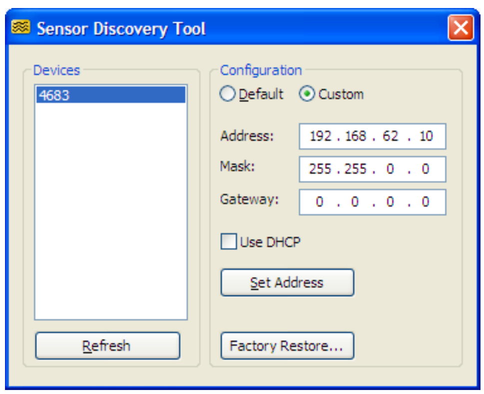
Tools and Native Drivers

The following sections describe the tools and native drivers 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.

GenTL Driver

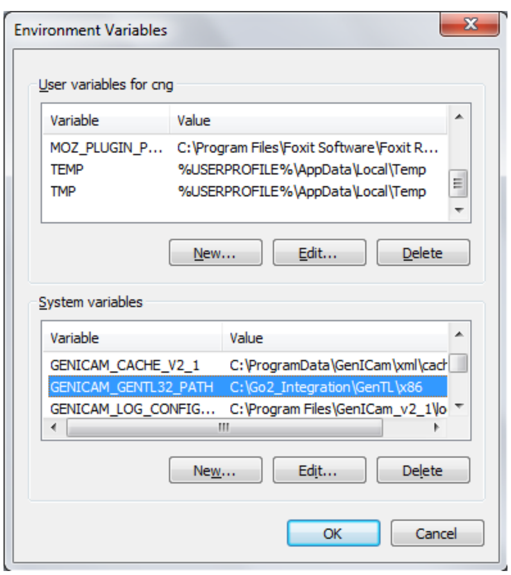
GenTL is an industry standard method of controlling and acquiring data from an imaging device. Gocator provides a GenTL driver that allows GenTL-compliant third-party software (e.g., Halcon and Common Vision Blox) to acquire and process 3D point clouds and intensity generated from the Gocator's Surface mode in real-time.

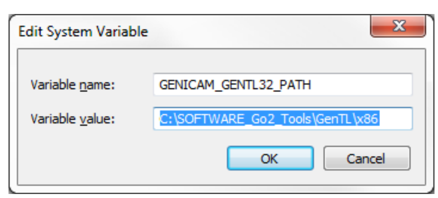
You can download the toolset package containing the driver from the LMI Technologies website at http://lmi3d.com/support/downloads/. Click on the link for your sensor, click on Product User Area, and log in.

After downloading the tool package (14405-x.x.x.x_SOFTWARE_GO_Tools.zip), unzip the file. The driver is found under the GenTL\x86 directory.

To install the driver in Windows 7:

1. Open the **Control** panel.
2. Select **System and Security** and then click **System**.
3. Click **Advanced System Settings**.
   The **Advanced System Settings** link is typically in the left column of the window.
4. For 32-bit systems, click **New** to create a system environment variable GENICAM_GENTL32_PATH and point it to the GenTL\x86 directory.
   If the system environment variable already exists, click **Edit**.
5. For 64-bit systems, click **New** to create a system environment variable GENICAM_GENTL64_PATH and point it to the GenTL\x64 directory.
   If the system environment variable already exists, click **Edit**.





To work with the Gocator GenTL driver, the Gocator must operate in Surface mode with its the appropriate output enabled in the **Ethernet** panel in the **Output** page. Check **Acquire Intensity** in the

Scan Mode panel on the **Scan** page and enable intensity output in the **Ethernet** panel if intensity data is required.

Refer to the documentations in the GenTL\ directory for instructions on how to interface to various third party software.

Gocator GenTL driver packs the part output, intensity and stamps (e.g., time stamp, encoder index, etc.) into either a 16-bit RGB image or a 16-bit grey scale image. You can select the format in the Go2GenTL.xml setting file.

The width and height of the 16-bit RGB or grey scale image is calculated from the maximum number of columns and rows required to support the sensor's FOV and the maximum part length.

16-bit RGB Image

When the 16-bit RGB format is used, the height map, intensity, and stamps are stored in the red, green, and blue channel respectively.

Channel	Details
Red	Height map information. The width and height of the image represent the dimensions in the X and Y axis. Together with the pixel value, each red pixel presents a 3D point in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = Z offset + Pz * Z resolution Refer to the blue channel on how to retrieve the offset and resolution values. If Pz is 0 if the data is invalid. The Z offset is fixed to -32768 * Z resolution. Z is zero if Pz is 32768.
Green	Intensity information. Same as the red channel, the width and height of the image represent the dimension in the X and the Y axis. Together with the pixel value, each blue pixel represents an intensity value in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = 16-bit intensity value The intensity value is 0 if the intensity image is not available. Gocator outputs 8-bit intensity values. The values stored in the 16-bit RGB image is multiplied by 256. To obtain the original values, divide the intensity values by 256. Refer to the blue channel on how to retrieve the offset and resolution values.
Blue	Stamp information. Stamps are 64-bit auxiliary information related to the height map and intensity content. The next table explains how the stamps are packed into the blue

Channel	Details
	pixel channel See *Data Results* on page 244 for an explanation of the stamp information.

The following table shows how the stamp information is packed into the blue channel. A stamp is a 64-bit value packed into four consecutive 16-bit blue pixels, with the first byte position storing the most significant byte.

Stamp Information from GenTL driver

Stamp Index	Blue Pixel Position	Details
0	0..3	Version
1	4..7	Frame Count
2	8..11	Timestamp (us)
3	12..15	Encoder value (ticks)
4	16..19	Encoder index (ticks) This is the encoder value when the last index is triggered
5	20..23	Digital input states
6	24..27	X offset (nm)
7	28..31	X resolution(nm)
8	32..35	Y offset (nm)
9	36..39	Y resolution (nm)
10	40..43	Z offset (nm)
11	44..47	Z resolution (nm)
12	48..51	Height map Width (in pixels)
13	52..55	Height map length (in pixels)
14	56..59	Specify if the intensity is enabled

16-bit Grey Scale Image

When the 16-bit grey scale format is used, the height map, intensity, and stamps are stored sequentially in the grey scale image.

The last row of the image contains the stamp information.

Rows	Details
0 .. (max part height - 1)	Height map information. The width and height of the image represent the dimensions in the X and Y axis. Together with the pixel value, each pixel presents a 3D point in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): X = X offset + Px * X resolution

Rows	Details
	Y = Y offset + Py * Y resolution Z = Z offset + Pz * Z resolution Refer to the blue channel on how to retrieve the offset and resolution values. If Pz is 0 if the data is invalid. The Z offset is fixed to -32768 * Z Resolution. Z is zero if Pz is 32768.
(max part height) .. 2* (max part height) If intensity is enabled	Intensity information. The width and height of the image represent the dimension in the X and the Y axis. Together with the pixel value, each blue pixel represents an intensity value in the real-world coordinates. The following formula can be used to calculate the real-world coordinates (X, Y, Z) from pixel coordinates (Px, Py, Pz): The following formula assumes Py is relative to the first row of the intensity information, not the first row of the whole 16-bit grey scale image. X = X offset + Px * X resolution Y = Y offset + Py * Y resolution Z = 16-bit intensity value This intensity value is 0 if the intensity image is not available. Gocator outputs 8-bit intensity values. The values stored in the 16-bit Grey scale image is multiplied by 256. To obtain the original values, divide the intensity values by 256. Refer to the stamps on how to retrieve the offset and resolution values.
The last row of the 16-bit grey scale image	Stamp information. Stamps are 64-bit auxiliary information related to the height map and intensity content. The next table explains how the stamps are packed into the blue pixel channel See *Data Results* on page 244 for an explanation of the stamp information.

The following table shows how the stamp information is packed into the last row. A stamp is a 64-bit value packed into four consecutive 16-bit pixels, with the first byte position storing the most significant byte.

Stamp Information from GenTL driver

Stamp Index	Column Position	Details
0	0..3	Version
1	4..7	Frame Count
2	8..11	Timestamp (us)
3	12..15	Encoder value (ticks)
4	16..19	Encoder index (ticks) This is the encoder value when the last index is triggered
5	20..23	Digital input states
6	24..27	X offset (nm)
7	28..31	X resolution(nm)
8	32..35	Y offset (nm)

Stamp Index	Column Position	Details
9	36..39	Y resolution (nm)
10	40..43	Z offset (nm)
11	44..47	Z resolution (nm)
12	48..51	Height map Width (in pixels)
13	52..55	Height map length (in pixels)
14	56..59	Specify if intensity is enabled or not

Registers

GenTL registers are multiple of 32 bits. The registers are used to control the operation of the GenTL driver, send commands to the sensors, or to report the current sensor information.

Register Map Overview

Register Address	Name	Read/Write	Length (bytes)	Description
260	WidthReg	RO	4	Specify the width of the returned images. The part height map is truncated if it is wider than the specified width.
264	HeightReg	RO	4	Specify the height of the returned images (i.e., length of the part). The part height map is truncated if it is longer than the specified length.
292	ResampleMode	RO	4	Enable the resampling logic in the GenTL driver 0 – Disable resampling 1 – Enable resampling When resampling is enabled, the GenTL driver will resample the height map so that the pixel spacing is the same in the X and Y axis.
296	EncoderValue0	RO	4	Report the current encoder value (least significant 32-bit). The current encoder value is latched from the sensor when this register is read.
300	EncoderValue1	RO	4	Report the current encoder value (most significant 32-bit). The encoder value is latched when EncoderValue0 register is read. User should read EncoderValue0 before reading EncoderValue1.
304	Configuration File	RW	16	Read the name of sensor live configuration file or switch (write) the sensor configuration file. The configuration name is NULL terminated and includes the extension ".job". Writing to this register causes the sensor to switch to the specified configuration.

Register Address	Name	Read/Write	Length (bytes)	Description
320	Transformation X offset	RO	4	Return the sensor transformation X offset
324	Transformation Z offset	RO	4	Return the sensor transformation Z offset
328	Transformation Angle	RO	4	Return the sensor transformation angle
332	Transformation Orientation	RO	4	Return the sensor transformation orientation
336	Clearance distance	RO	4	Return the sensor clearance distance

XML Settings File

The settings file, Go2GenTL.xml, resides in the same directory as the Gocator GenTL driver. Users can set the resample mode and output format by changing the setting in this file.

Element	Type	Description
ResampleMode	32u	Settings to disable or enable resampling mode: 0 – Disable 1 – Enable When resampling mode is enabled, the GenTL driver will resample the height map so that the pixel spacing is the same in the X and Y axis. The default value is 1.
DataFormat	32u	Settings to choose 16-bit RGB or 16-bit grey scale image output: 0 – 16-bit RGB Image 1 – 16-bit grey scale Image The default value is 0

CSV Converter Tool

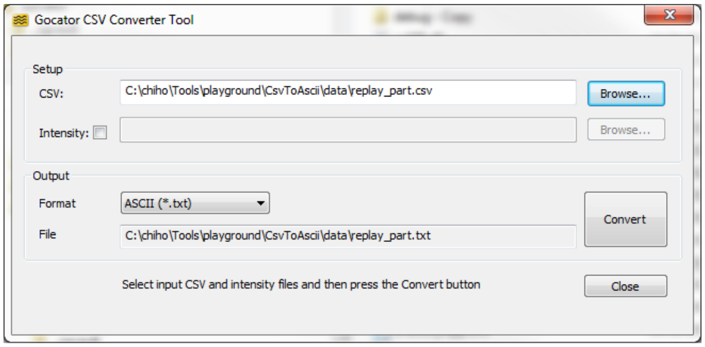
After you have exported recorded data to CSV, you can use the Gocator CSV Converter Tool to convert the exported part data into the following formats:

- ASCII (XYZI)
- 16-bit BMP
- 16-bit PNG
- GenTL

You can get the tool package (14405-x.x.x.x_SOFTWARE_GO_Tools.zip) from the LMI Technologies website at http://lmi3d.com/support/downloads/. Click on the link for your sensor, click on Product User Area, and log in.

For more information on exporting recorded data, see see *Downloading, Uploading, and Exporting Replay Data* on page 40.

After downloading the tool package, unzip the file and run the Gocator CSV Converter tool [bin>win32>kCsvConverter.exe].



The software tool supports data exported from Surface mode.

The GenTL format is a 48-bit RGB or grey scale PNG. Height map, intensity and stamp information are stored as defined in the GenTL Driver section (see on page 287). You can load the exported data into image processing software to provide simulation data for developing applications using the GenTL driver.

To convert exported CSV into different formats:

1. Select the CSV file to convert.
2. If intensity information is required, check the **Intensity** box and select the intensity bitmap.
   Intensity information is only used when converting to ASCII or GenTL format. If intensity is not selected, the ASCII format will only contain the point coordinates (XYZ).
3. Select the output format.
   The converted file will reside in the same directory as the CSV file. It will also have the same name but with a different file extension. The converted file name is displayed in the **Output File** field.
4. Press the **Convert** button.

Troubleshooting

Review the guidance in this chapter if you are experiencing difficulty with a Gocator sensor system. See *Return Policy* on page 319 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* on page 285 for more information.

When attempting to log in, the password is not accepted.

- See *Sensor Recovery Tool* on page 285 for steps to reset the password.

3D Data Acquisition

When the Start button or the Snapshot button is pressed, the sensor does not emit light.

- The safety input signal may not be correctly applied. See *Specifications* on page 296 for more information.
- The exposure setting may be too low. See *Exposure* on page 60 for more information on configuring exposure time.
- Use the Snapshot button instead of the Start button to capture 3D point cloud data. If the LED light 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* on page 54 for information on configuring the trigger source.

Performance

The sensor CPU level is near 100%.

- Consider reducing the speed. If you are using a time trigger source, see *Triggers* on page 54 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.

- Consider reducing the resolution.
  See *Spacing* on page 63 for more information on configuring resolution.
- 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 3100 Series

The Gocator 3100 series consists of the sensor models defined below.

MODEL	3109	3110
Clearance Distance (CD) (mm)	157	150
Measurement Range (MR) (mm)	70	100
Field of View (mm)	67 x 86 - 93 x 88	60 x 105 - 90 x 160
Linearity Z (+/- % of MR)	0.038	0.050
Resolution Z (mm)	0.023 - 0.044	0.035 - 0.108
Resolution XY (mm)	0.065 x 0.071 - 0.091 x 0.092	0.090 x 0.100 - 0.150 x 0.160
Dimensions (mm)	49 x 100 x 155	49 x 74.4 x 276
Weight (kg)	1.04	1.35

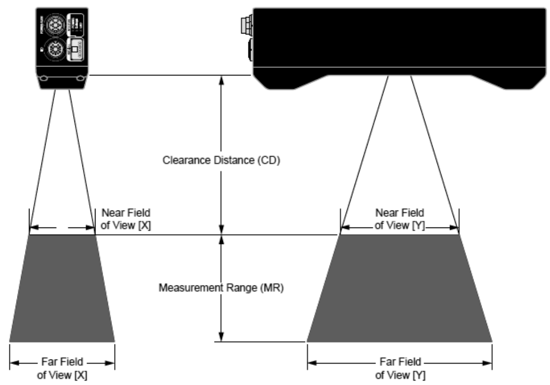
Linearity Z values and Resolution Z are typical values.

Field of View and Resolution XY are specified as [X] x [Y], near to far.

ALL 3100 SERIES MODELS	
Scan Rate	Up to 5 Hz
Light Source	Blue LED (465 nm)
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Light Safety Enable, Trigger
Outputs	2x Digital output, RS-485 Serial (115 kBaud) Analog Output (4 – 20 mA)
Input Voltage (Power)	+24 to +48 VDC (25 Watts); Ripple +/- 10%
Housing	Gasketed aluminum enclosure, IP67
Operating Temp.	0 to 50° C
Storage Temp.	-30 to 70° C
Vibration Resistance	10 to 55 Hz, 1.5 mm double amplitude in X, Y and Z directions, 2 hours per direction
Shock Resistance	15 g, half sine wave, 11 ms, positive and negative for X, Y and Z directions

Differential encoder requires the use of Master 400/800/1200/2400.

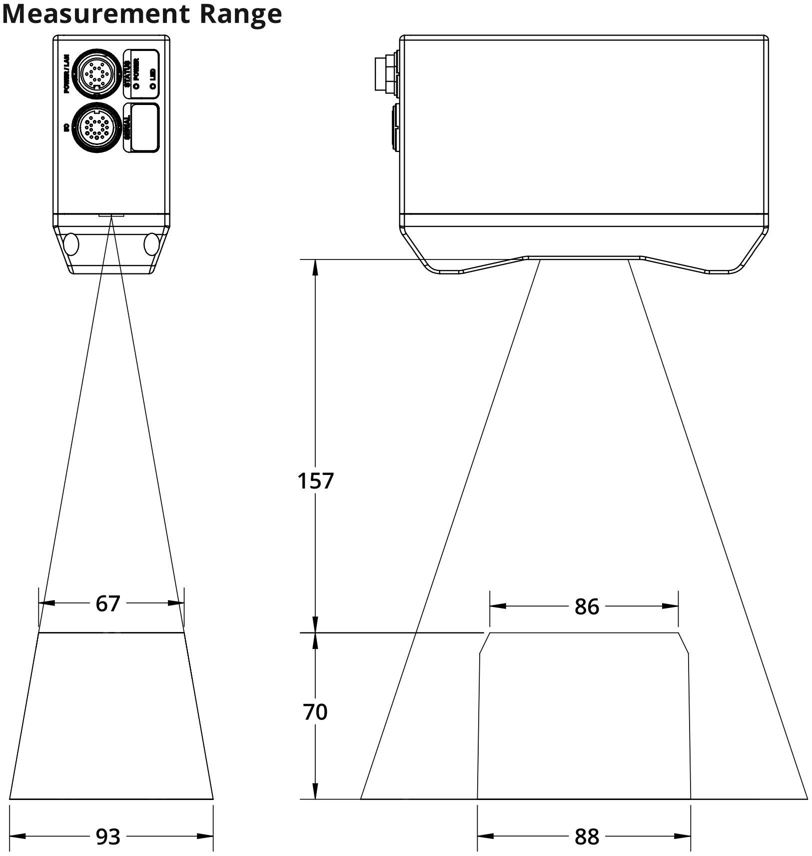
The following diagram indicates the names corresponding to various measurements in the subsequent sections.



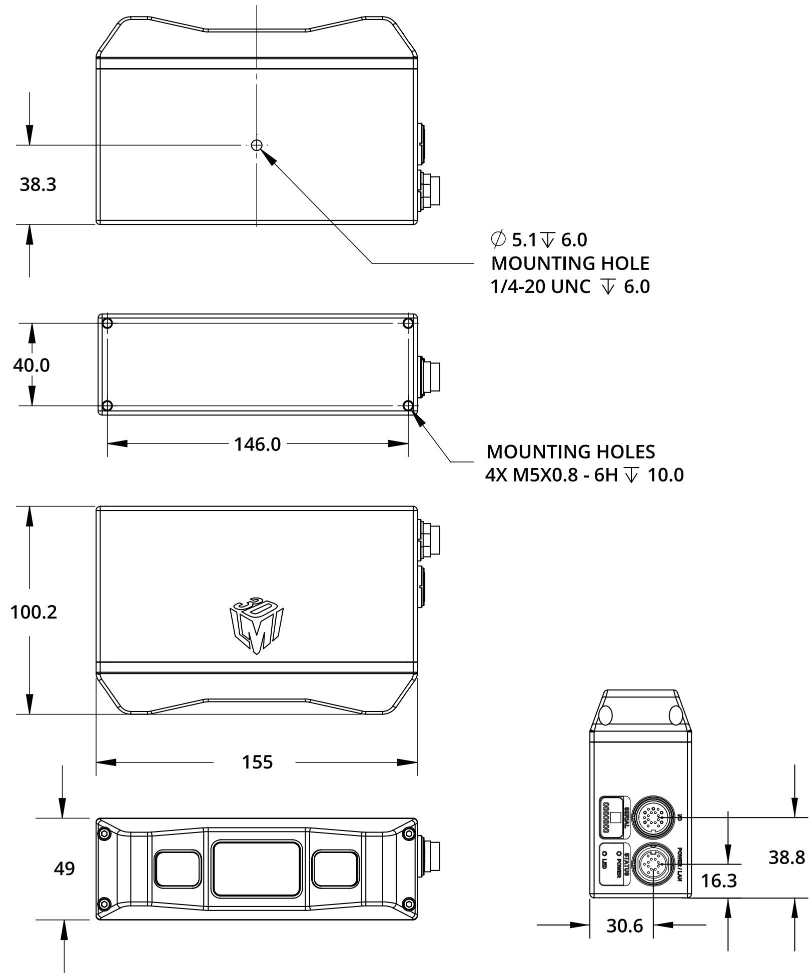
Mechanical dimensions for each sensor model are illustrated on the following pages.

Gocator 3109

Field of View / Measurement Range

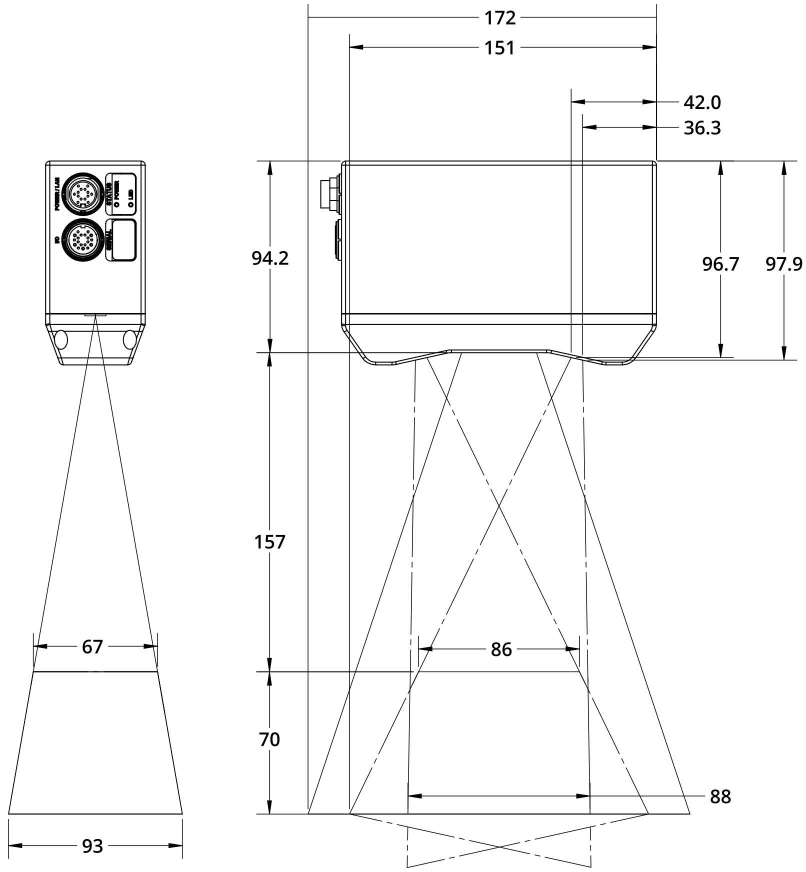


Dimensions



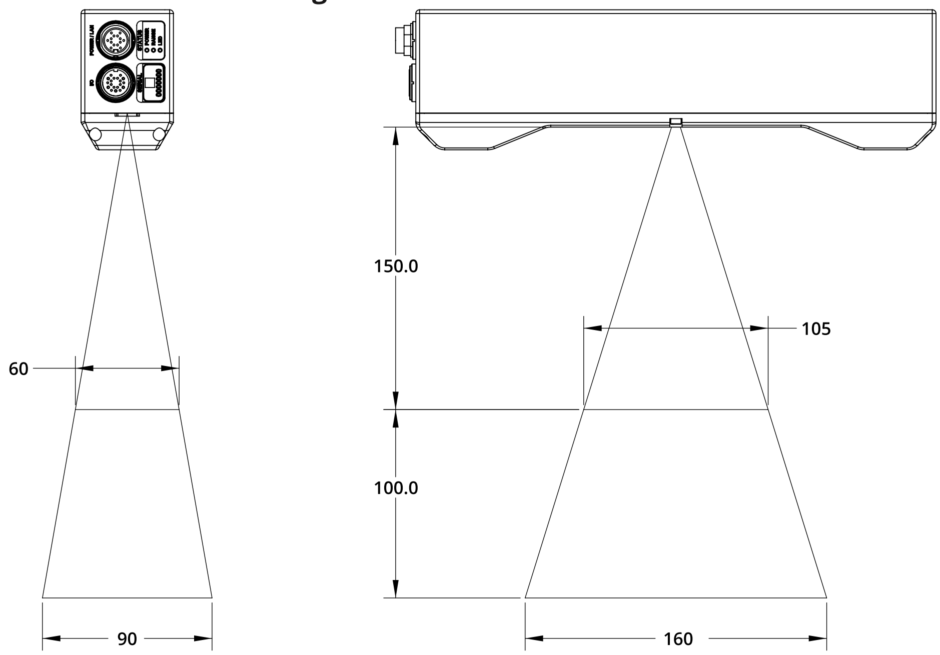
38.3
∅ 5.1 ↧ 6.0
MOUNTING HOLE
1/4-20 UNC ↧ 6.0
40.0
146.0
MOUNTING HOLES
4X M5X0.8 - 6H ↧ 10.0
100.2
155
49
38.8
16.3
30.6

Envelope

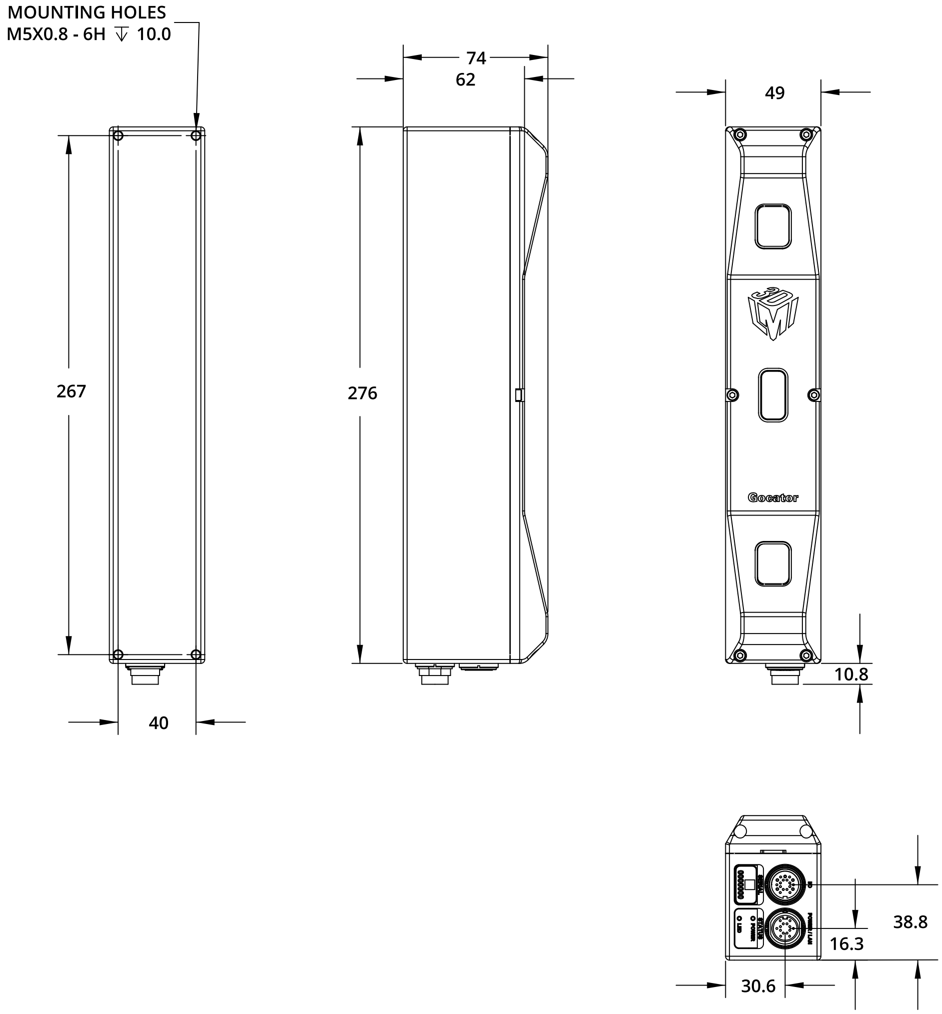


Gocator 3110

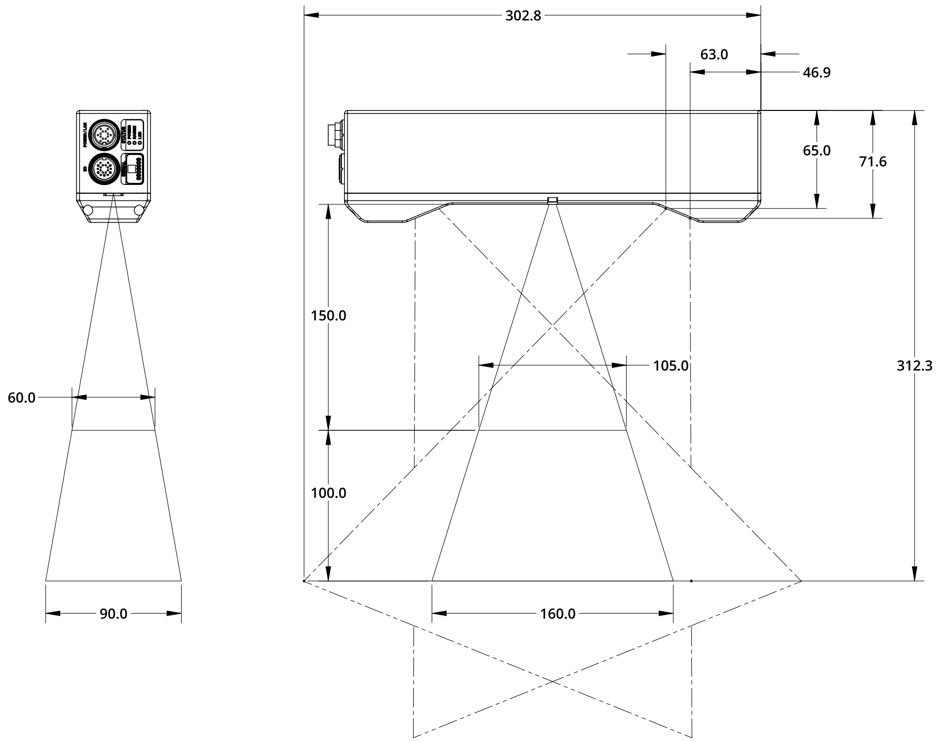
Field of View / Measurement Range



Dimensions


MOUNTING HOLES
M5X0.8 - 6H ↧ 10.0
74
62
49
267
276
40
10.8
Gocator
38.8
16.3
30.6

Envelope



Gocator Power/LAN Connector

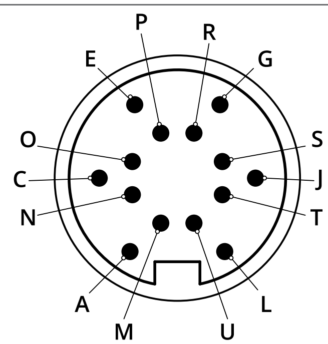
The Gocator Power/LAN connector is a 14 pin, M16 style connector that provides power input, safety input and Ethernet.

	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	L	White/ Orange & Black
GND_24-48V	L	Orange/ Black
DC_24-48V	A	White/ Green & Black
DC_24-48V	A	Green/ Black
Safety-	G	White/ Blue & Black
Safety+	J	Blue/ Black
Sync+	E	White/ Brown & Black
Sync-	C	Brown/ Black
Ethernet MX1+	M	White/ Orange
Ethernet MX1-	N	Orange
Ethernet MX2+	O	White/ Green
Ethernet MX2-	P	Green
Ethernet MX3-	S	White/ Blue
Ethernet MX3+	R	Blue
Ethernet MX4+	T	White/ Brown
Ethernet MX4-	U	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

Apply positive voltage to DC_24-48V. See *Gocator 3100 Series* on page 296 for the sensor's power requirement. Apply ground to GND_24-48VDC.

Power requirements

Function	Pins	Min	Max
DC_24-48V	A	24 V	48 V
GND_24-48VDC	L	0 V	0 V

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+	J	24 V	48 V
Safety_in-	G	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 3100 I/O Connector

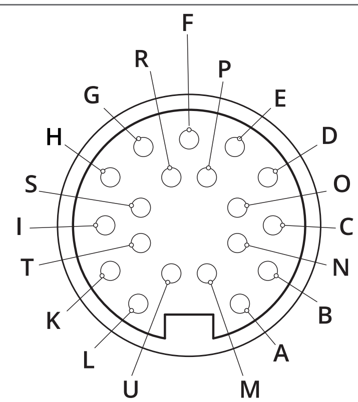
The Gocator 3100 I/O connector is a 19 pin, M16 style connector that provides encoder, digital input, digital outputs, serial output, and analog output signals.

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+	D	Grey
Trigger_in-	H	Pink
Out_1+ (Digital Output 0)	N	Red
Out_1- (Digital Output 0)	O	Blue
Out_2+ (Digital Output 1)	S	Tan
Out_2- (Digital Output 1)	T	Orange
Encoder_A+	M	White / Brown & Black
Encoder_A-	U	Brown / Black
Encoder_B+	I	Black
Encoder_B-	K	Violet
Encoder_Z+	A	White / Green & Black
Encoder_Z-	L	Green / Black
Serial_out+	B	White
Serial_out-	C	Brown
Reserved	E	Blue / Black
Reserved	G	White / Blue & Black
Analog_out+	P	Green
Analog_out-	F	Yellow & Maroon / White
Reserved	R	Maroon



*View: Looking into the connector **on** the sensor*

Grounding Shield

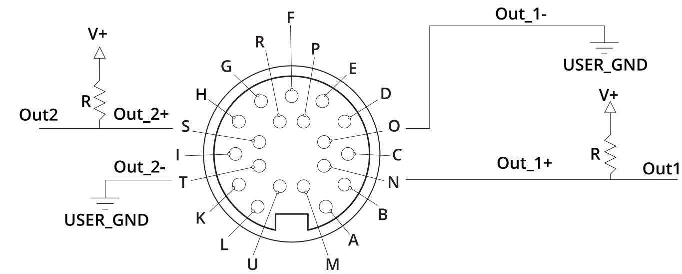
The grounding shield should be mounted to the earth ground.

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 N and Emitter – Pin O) and Out_2 (Collector – Pin S and Emitter Pin T) 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	N, O	40 mA	70 V	20 us
Out_2	S, T	40 mA	70 V	20 us

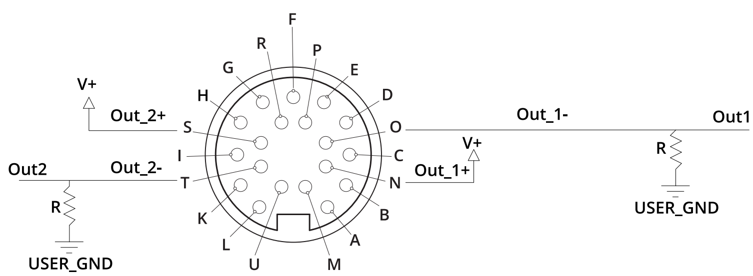


The resistors shown above are calculated by R = (V+) / 2.5 mA.

The size of the resistors is determined by power = (V+)^2 / R.

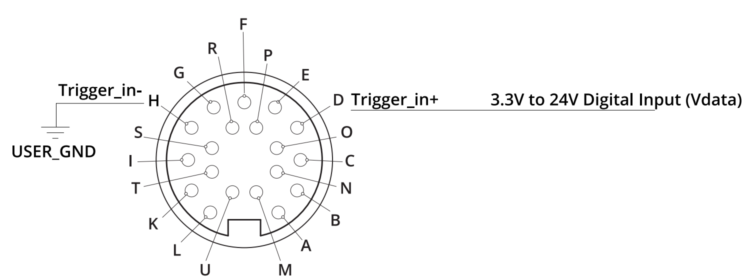
Inverting Outputs

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.



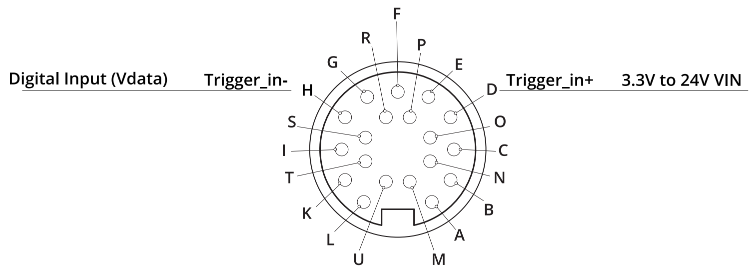
Digital Inputs

Every Gocator sensor has a single optically isolated input. To use this input without an external resistor, supply 3.3 - 24 V to Pin D and GND to Pin H.



Active High

If the supplied voltage is greater than 24 V, connect an external resistor in series to Pin D. The resistor value should be R = [(Vin-1.2V)/10mA]-680.



Active Low

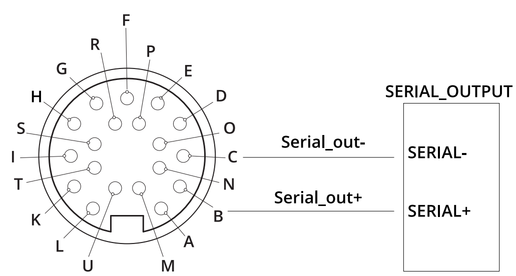
To assert the signal, the digital input voltage should be set to draw a current of 3 mA to 40 mA from Trigger_In+. The current that passes through Trigger_In+ is I = (Vin – 1.2 – Vdata) / 680. To reduce noise sensitivity, we recommend leaving a 20% margin for current variation (i.e., uses a digital input voltage that draws 4mA to 25mA).

Function	Pins	Min Voltage	Max Voltage	Min Current	Max Current	Min Pulse Width
Trigger_in	D, H	3.3 V	24 V	3 mA	40 mA	20 us

Serial Output

Serial RS-485 output is connected to Serial_out as shown below.

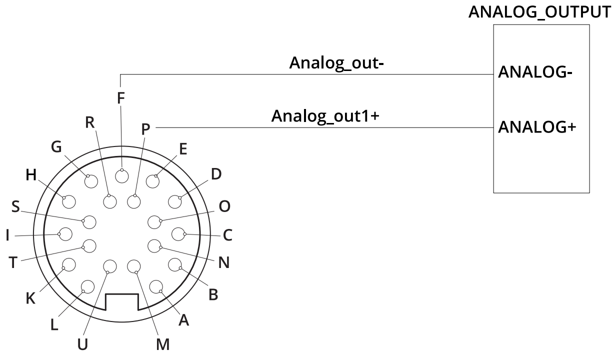
Function	Pins
Serial_out	B, C



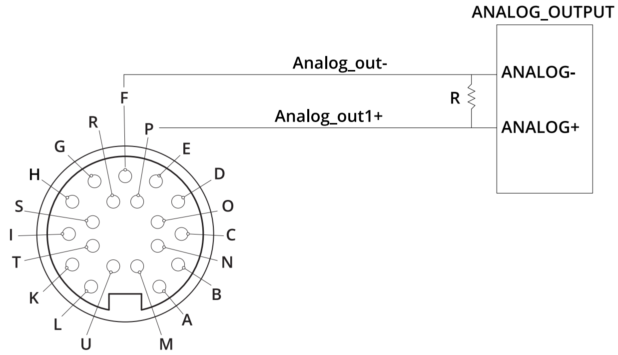
Analog Output

The Sensor I/O Connector defines one analog output interface: Analog_out.

Function	Pins	Current Range
Analog_out	P, F	4 – 20 mA

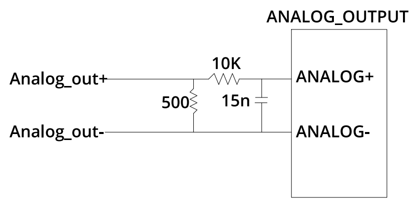


Current Mode



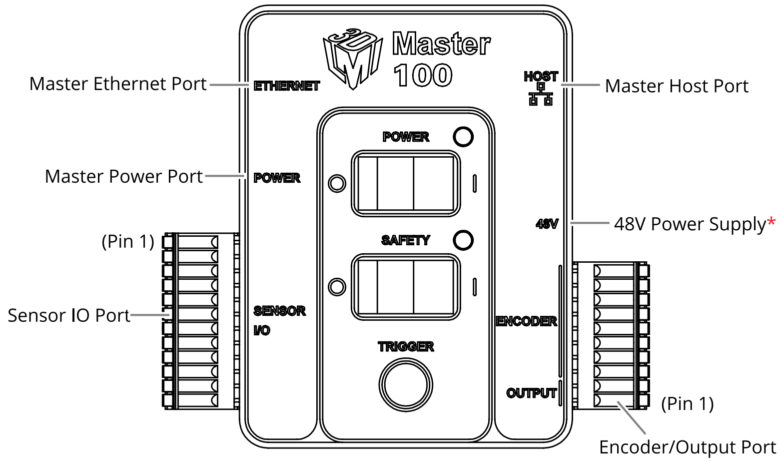
Voltage Mode

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

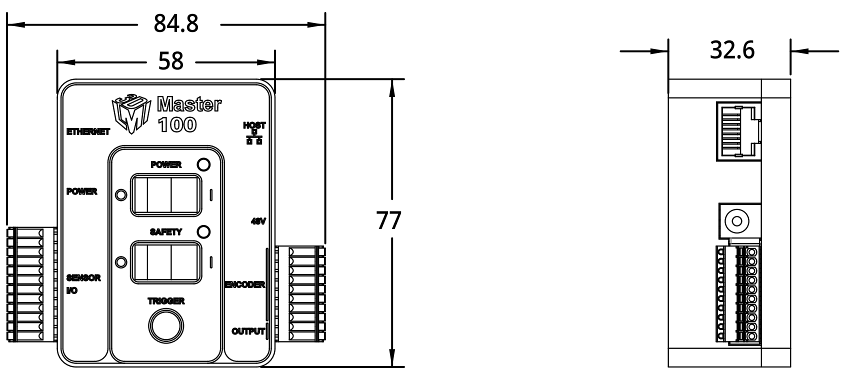
Gocator I/O Pin	Master Pin	Conductor Color
Encoder_A+	1	White/Brown & Black
Encoder_A-	2	Brown/Black
Encoder_Z+	3	White/Green & Black
Encoder_Z-	4	Green/Black
Trigger_in+	5	Grey
Trigger_in-	6	Pink
Out_1-	7	Blue
Out_1+	8	Red
Encoder_B+	11	Black
Encoder_B-	12	Violet

The rest of the wires in the Gocator I/O cordset are not used.

Encoder/Output Port Pins

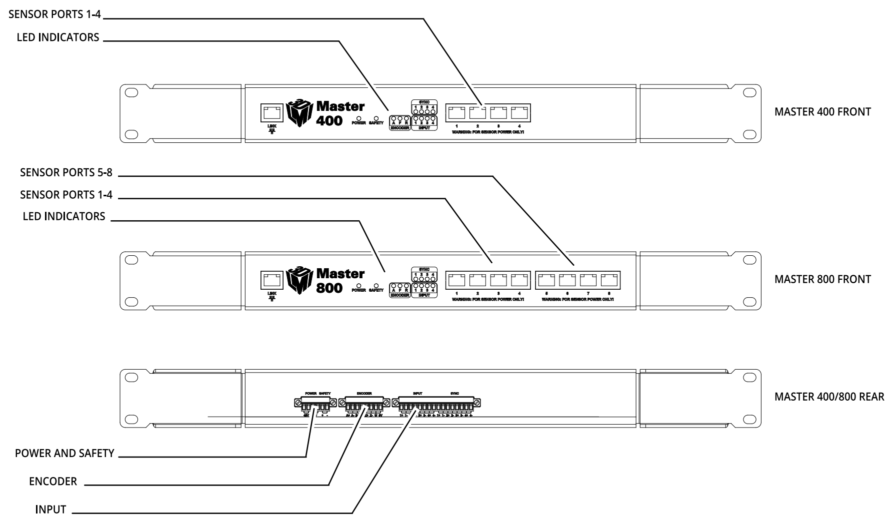
Function	Pin
Output_1+ (Digital Output 0)	1
Output_1- (Digital Output 0)	2
Encoder_Z+	3
Encoder_Z-	4
Encoder_A+	5
Encoder_A-	6
Encoder_B+	7
Encoder_B-	8
Encoder_GND	9
Encoder_5V	10

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
Reserved	4
Reserved	5

Function	Pin
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

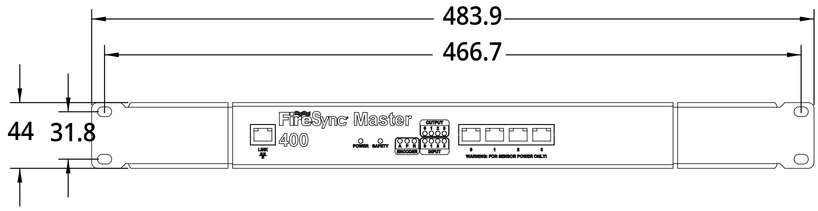
When using a Master 400/800, its chassis must be well grounded.

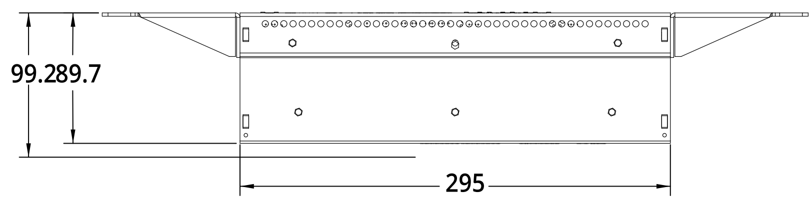
	The +48VDC power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

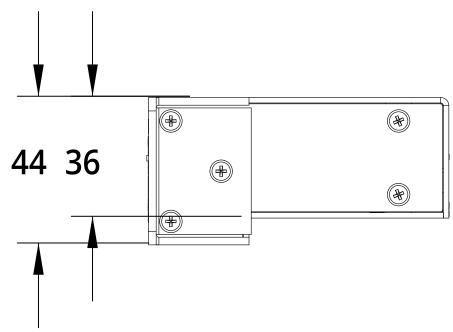
	The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements which need to be considered when calculating total system power requirements.

Master 400/800 Dimensions

The dimensions of Master 400 and Master 800 are the same.

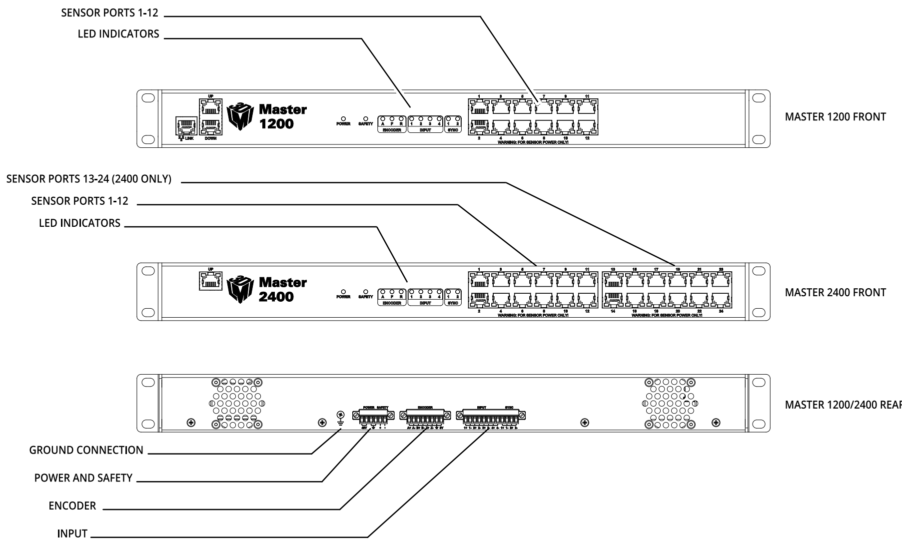






Master 1200/2400

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
Reserved	4

Function	Pin
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

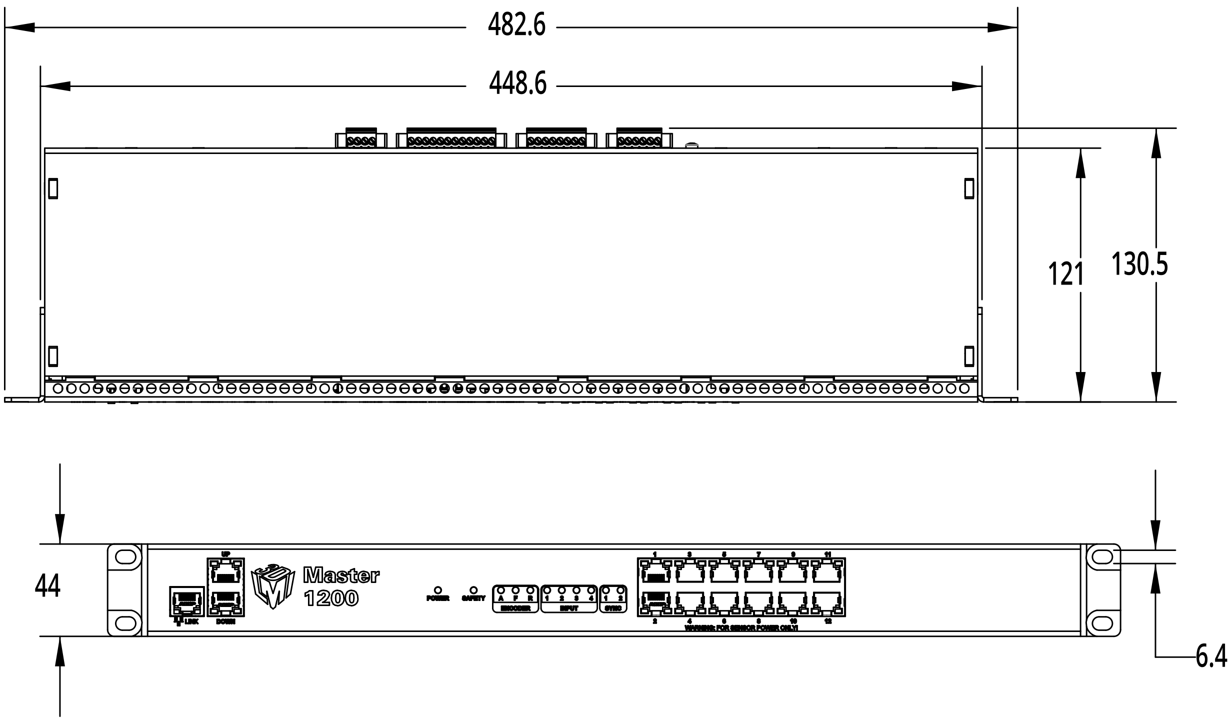
⚠ When using a Master 1200/2400, its chassis must be well grounded.

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 3100 Sensors

Description	Part Number
Gocator 3109, No tools	313109A-LED-B-00
Gocator 3109, Measurement tools	313109A-LED-B-01
Gocator 3110, No tools	313110A-LED-B-00
Gocator 3110, Measurement tools	313110A-LED-B-01

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
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

Description	Part Number
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

BlowFish

Website:

http://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html

License:

CodeMirror

Website:

http://codemirror.net

License:

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:

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)

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

Americas	EMEAR	ASIA PACIFIC
LMI Technologies (Head Office)	LMI Technologies GmbH	LMI (Shanghai) Trading Co., Ltd.
Vancouver, Canada	Berlin, Germany	Shanghai, China
+1 604 636 1011	+49 (0)3328 9360 0	+86 21 5441 0711

LMI Technologies has sales offices and distributors worldwide. All contact information is listed at lmi3D.com/contact/locations.