



USER MANUAL

Gocator Point Profile Sensors

Gocator 1300 Series

Firmware version: 5.0.x.xx

Document revision: A

Copyright

Copyright © 2018 by LMI Technologies, Inc. All rights reserved.

Proprietary

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

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

Trademarks and Restrictions

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

Information contained within this manual is subject to change.

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

Contact Information

LMI Technologies, Inc.
9200 Glenlyon Parkway
Burnaby BC V5J 5J8
Canada

Telephone: +1 604-636-1011

Fax: +1 604-516-8368

www.lmi3d.com

Table of Contents

Copyright	2
Table of Contents	3
Introduction	11
Gocator Overview	12
Safety and Maintenance	13
Laser Safety	13
Laser Classes	14
Precautions and Responsibilities	15
Class 3B Responsibilities	15
Nominal Ocular Hazard Distance (NOHD)	16
Systems Sold or Used in the USA	16
Electrical Safety	17
Handling, Cleaning, and Maintenance	17
Environment and Lighting	18
Getting Started	19
Hardware Overview	20
Side Mount Package	20
Top Mount Package	20
Gocator Cordsets	21
Master 100	22
Master 400 / 800 / 1200 / 2400	22
Master 810 / 2410	24
Alignment Targets	25
System Overview	26
Standalone System	27
Dual-Sensor System	27
Multi-Sensor System	28
Installation	29
Mounting: Side Mount Package	29
Mounting - Top Mount Package	30
Orientations	31
Grounding	33
Gocator	33
Recommended Practices for Cordsets	34
Master Network Controllers	34
Grounding When Using a DIN Rail (Master 810/2410)	35
Installing DIN Rail Clips: Master 810 or 2410	35
Configuring Master 810	36
Setting the Divider	37

Encoder Quadrature Frequency	37
Setting the Debounce Period	38
Network Setup	39
Client Setup	39
Gocator Setup	41
Running a Standalone Sensor System	41
Running a Dual-Sensor System	42
Next Steps	44
How Gocator Works	46
3D Acquisition	46
Clearance Distance and Measurement Range	47
Resolution and Accuracy	47
Z Resolution	47
Z Linearity	48
Range Output	49
Coordinate Systems	49
Sensor Coordinates	49
System Coordinates	49
Data Generation and Processing	51
Profile Generation	51
Part Detection	51
Measurement and Anchoring	52
Output and Digital Tracking	52
Gocator Web Interface	53
Browser Compatibility	53
Internet Explorer 11 Issues	53
Internet Explorer Switches to Software Rendering	53
Internet Explorer Displays "Out of Memory"	53
User Interface Overview	54
Toolbar	56
Creating, Saving and Loading Jobs (Settings)	56
Recording, Playback, and Measurement Simulation	57
Recording Filtering	59
Downloading, Uploading, and Exporting Replay Data	61
Metrics Area	63
Data Viewer	64
Status Bar	64
Log	64
Frame Information	64
Quick Edit Mode	65

Interface Language	65
Management and Maintenance	67
Manage Page Overview	67
Sensor System	68
Dual- and Multi-sensor Systems	68
Buddy Assignment	68
Over Temperature Protection	69
Sensor Autostart	70
Layout	70
Networking	73
Motion and Alignment	74
Alignment Reference	74
Encoder Resolution	75
Encoder Value and Frequency	75
Travel Speed	75
Jobs	76
Security	77
Maintenance	78
Sensor Backups and Factory Reset	79
Firmware Upgrade	80
Support	81
Support Files	82
Manual Access	83
Software Development Kit	83
Scan Setup and Alignment	84
Scan Page Overview	84
Scan Modes	85
Triggers	85
Trigger Examples	89
Trigger Settings	90
Maximum Input Trigger Rate	92
Maximum Encoder Rate	92
Sensor	92
Active Area	92
Transformations	94
Exposure	95
Single Exposure	96
Dynamic Exposure	96
Advanced	97
Material	99
Camera Gain and Dynamic Exposure	99
Alignment	100

Aligning Sensors	100
Encoder Calibration	103
Alignment Types	103
Clearing Alignment	104
Filters	104
Median	105
Smoothing	105
Decimation	106
Slope	107
Profile Generation	108
Part Detection	111
Data Viewer	113
Data Viewer Controls	113
Video Mode	114
Spots and Dropouts	114
Range Mode	115
Profile Mode	117
Region Definition	118
Intensity Output	119
Measurement	120
Measure Page Overview	120
Data Viewer	120
Tools Panel	121
Adding and Configuring a Measurement	
Tool	121
Source	122
Regions	122
Feature Points	125
Fit Lines	127
Geometric Features	128
Decisions	128
Filters	130
Measurement Anchoring	131
Enabling and Disabling Measurements	133
Editing a Tool or Measurement Name	134
Changing a Measurement ID	135
Duplicating a Tool	135
Removing a Tool	136
Reordering Tools	136
Range Measurement	137
Position	137
Measurements and Settings	137

Thickness	138	System Requirements	205
Measurements and Settings	138	Limitations	205
Script	139	Downloading a Support File	206
Profile Measurement	140	Running the Emulator	207
Area	140	Adding a Scenario to the Emulator	208
Measurements, Features, and Settings ..	142	Running a Scenario	208
Bounding Box	144	Removing a Scenario from the Emulator	209
Measurements, Features, and Settings ..	145	Using Replay Protection	210
Circle	146	Stopping and Restarting the Emulator	210
Measurements, Features, and Settings ..	147	Running the Emulator in Default Browser	210
Closed Area	148	Working with Jobs and Data	211
Measurements and Settings	149	Creating, Saving, and Loading Jobs	211
Dimension	152	Playback and Measurement Simulation	212
Groove	154	Downloading, Uploading, and Exporting	
Intersect	159	Replay Data	213
Measurements, Features, and Settings ..	159	Downloading and Uploading Jobs	215
Line	161	Scan, Model, and Measurement Settings	217
Measurements, Features, and Settings ..	162	Calculating Potential Maximum Frame Rate ..	217
Panel	165	Protocol Output	218
Position	168	Remote Operation	218
Measurements, Features, and Settings ..	169	Gocator Accelerator	220
Round Corner	170	System Requirements	221
Strip	174	Benefits	221
Script	178	Installation	221
Feature Measurement	179	Gocator Accelerator Utility	221
Dimension	180	Dashboard and Health Indicators	224
Intersect	181	SDK Application Integration	224
Scripts	183	Gocator Device Files	226
Built-in Functions	184	Live Files	226
Output	189	Log File	226
Output Page Overview	189	Job File Structure	227
Ethernet Output	190	Job File Components	227
Digital Output	194	Accessing Files and Components	228
Analog Output	196	Configuration	228
Serial Output	198	Setup	229
Dashboard	201	Filters	230
Dashboard Page Overview	201	XSmoothing	230
State and Health Information	201	YSmoothing	230
Statistics	202	XGapFilling	231
Measurements	203	YGapFilling	231
Performance	203	XMedian	231
Gocator Emulator	205	YMedian	231

XDecimation	232
YDecimation	232
XSlope	232
YSlope	232
Trigger	233
Layout	234
Alignment	235
Disk	236
Bar	236
Plate	237
Polygon	237
Polygon/Corner	237
Devices / Device	238
SurfaceGeneration	244
FixedLength	244
VariableLength	244
Rotational	245
SurfaceSections	245
ProfileGeneration	245
FixedLength	246
VariableLength	246
Rotational	246
PartDetection	246
EdgeFiltering	248
PartMatching	248
Edge	248
BoundingBox	249
Ellipse	249
Replay	250
RecordingFiltering	250
Conditions/AnyMeasurement	250
Conditions/AnyData	251
Conditions/Measurement	251
Streams/Stream (Read-only)	251
ToolOptions	252
MeasurementOptions	253
FeatureOptions	253
StreamOptions	254
Tools	254
Profile Types	254
ProfileFeature	254
ProfileLine	255

ProfileRegion2d	255
Geometric Feature Types	255
Parameter Types	255
RangePosition	257
RangeThickness	258
ProfileArea	259
ProfileBoundingBox	261
ProfileCircle	262
ProfileDimension	264
ProfileGroove	265
ProfileIntersect	267
ProfileLine	269
ProfilePanel	271
ProfilePosition	273
ProfileRoundCorner	274
ProfileStrip	276
Script	279
Tool	279
Tool	280
Custom	282
Output	282
Ethernet	283
Ascii	285
EIP	285
Modbus	286
Digital0 and Digital1	286
Analog	287
Serial	288
Selcom	288
Ascii	289
Transform	289
Device	290
Protocols	291
Gocator Protocol	291
Data Types	292
Commands	292
Discovery Commands	293
Get Address	293
Set Address	294
Get Info	295
Control Commands	296
Protocol Version	297

Get Address	297	Backup	319
Set Address	298	Restore	320
Get System Info V2	298	Restore Factory	320
Get System Info	301	Get Recording Enabled	321
Get States	302	Set Recording Enabled	321
Log In/Out	303	Clear Replay Data	322
Change Password	303	Get Playback Source	322
Assign Buddies	304	Set Playback Source	322
Remove Buddies	305	Simulate	323
Set Buddy	305	Seek Playback	323
List Files	305	Step Playback	324
Copy File	306	Playback Position	324
Read File	306	Clear Measurement Stats	325
Write File	307	Read Live Log	325
Delete File	308	Clear Log	325
User Storage Used	308	Simulate Unaligned	326
User Storage Free	308	Acquire	326
Get Default Job	309	Acquire Unaligned	326
Set Default Job	309	Create Model	327
Get Loaded Job	309	Detect Edges	327
Get Alignment Reference	310	Add Tool	328
Set Alignment Reference	310	Add Measurement	328
Clear Alignment	311	Read File (Progressive)	329
Get Timestamp	311	Export CSV (Progressive)	329
Get Encoder	311	Export Bitmap (Progressive)	330
Reset Encoder	312	Get Runtime Variable Count	331
Start	312	Set Runtime Variables	331
Scheduled Start	313	Get Runtime Variables	332
Stop	313	Upgrade Commands	332
Get Auto Start Enabled	313	Start Upgrade	333
Set Auto Start Enabled	314	Start Upgrade Extended	333
Get Voltage Settings	314	Get Upgrade Status	333
Set Voltage Settings	315	Get Upgrade Log	334
Get Quick Edit Enabled	315	Results	334
Set Quick Edit Enabled	315	Data Results	334
Start Alignment	316	Stamp	335
Start Exposure Auto-set	316	Video	336
Software Trigger	317	Range	336
Schedule Digital Output	317	Range Intensity	337
Schedule Analog Output	318	Profile Point Cloud	337
Ping	318	Profile Intensity	338
Reset	319	Measurement	339

Alignment Result	339
Exposure Calibration Result	340
Event	340
Feature Point	341
Feature Line	341
Feature Plane	341
Feature Circle	342
Health Results	342
Modbus Protocol	348
Concepts	348
Messages	348
Registers	349
Control Registers	350
Output Registers	351
State	351
Stamp	352
Measurement Registers	353
EtherNet/IP Protocol	355
Concepts	355
Basic Object	356
Identity Object (Class 0x01)	356
TCP/IP Object (Class 0xF5)	356
Ethernet Link Object (Class 0xF6)	356
Assembly Object (Class 0x04)	357
Command Assembly	357
Runtime Variable Configuration Assembly	358
Sensor State Assembly	359
Sample State Assembly	360
Implicit Messaging Command Assembly	361
Implicit Messaging Output Assembly	362
ASCII Protocol	364
Connection Settings	364
Ethernet Communication	364
Serial Communication	365
Polling Operation Commands (Ethernet Only)	365
Command and Reply Format	366
Special Characters	366
Command Channel	366
Start	366
Stop	367
Trigger	367
LoadJob	368

Stamp	368
Clear Alignment	368
Moving Alignment	369
Stationary Alignment	369
Set Runtime Variables	369
Get Runtime Variables	370
Data Channel	370
Result	370
Value	371
Decision	372
Health Channel	372
Health	372
Standard Result Format	373
Custom Result Format	374
Selcom Protocol	375
Serial Communication	375
Connection Settings	375
Message Format	375
Development Kits	377
GoSDK	377
Setup and Locations	378
Class Reference	378
Examples	378
Example Project Environment Variable	378
Header Files	378
Functional Hierarchy of Classes	378
GoSystem	379
GoSensor	379
GoSetup	379
GoLayout	379
GoTools	380
GoTransform	380
GoOutput	380
Data Types	380
Value Types	380
Output Types	380
GoDataSet Type	381
Measurement Values and Decisions	382
Batching	382
Operation Workflow	383
Initialize GoSdk API Object	384
Discover Sensors	384

Connect Sensors	384	Profile	407
Configure Sensors	384	RawProfile	408
Enable Data Channels	385	Troubleshooting	409
Perform Operations	385	Specifications	411
Limiting Flash Memory Write Operations	386	Sensors	411
GDK	388	Gocator 1300 Series	411
Benefits	388	Gocator 1320 (Side Mount Package)	413
Supported Sensors	388	Gocator 1320 (Top Mount Package)	414
Typical Workflow	389	Gocator 1340 (Side Mount Package)	415
Installation and Class Reference	389	Gocator 1350 (Side Mount Package)	418
Required Tools	389	Gocator 1350 (Top Mount Package)	421
Getting Started with the Example Code	390	Gocator 1365 (Side Mount Package)	424
Building the Sample Code	390	Gocator 1370 (Side Mount Package)	427
Tool Registration	390	Gocator 1380 (Side Mount Package)	430
Tool Definitions	391	Gocator 1390 (Side Mount Package)	433
Entry Functions	391	Sensor Connectors	436
Parameter Configurations	392	Gocator Power/LAN Connector	436
Graphics Visualization	393	Grounding Shield	436
Debugging Your Measurement Tools	395	Power	437
Debugging Entry Functions	396	Laser Safety Input	437
Tips	396	Gocator I/O Connector	438
Backward Compatibility with Older Versions of Tools	396	Grounding Shield	438
Define new parameters as optional	396	Digital Outputs	438
Configuration Versioning	396	Inverting Outputs	439
Version	398	Digital Input	439
Common Programming Operations	398	Encoder Input	440
Input Data Objects	398	Serial Output	441
Setup and Region Info during Tool Initialization	399	Selcom Serial Output	441
Computing Region Based on the Offset from an Anchor Source	399	Analog Output	441
Part Matching	400	Master Network Controllers	443
Accessing Sensor Local Storage	400	Master 100	443
Print Output	400	Master 100 Dimensions	444
Tools	401	Master 400/800	445
Sensor Discovery Tool	401	Master 400/800 Electrical Specifications	446
CSV Converter Tool	402	Master 400/800 Dimensions	448
CSV File Format	403	Master 810/2410	449
Info	404	Electrical Specifications	451
DeviceInfo	405	Encoder	452
RecordingFilter	405	Input	454
Ranges	406	Master 810 Dimensions	456
		Master 2410 Dimensions	457
		Master 1200/2400	458

Master 1200/2400 Electrical Specifications .	459
Master 1200/2400 Dimensions	460
Accessories	461
Return Policy	463
Software Licenses	464
Support	470
Contact	471

Introduction

This documentation describes how to connect, configure, and use a Gocator. It also contains reference information on the device's protocols and job files, as well as an overview of the development kits you can use with Gocator. Finally, the documentation describes the Gocator emulator and accelerator applications.

The documentation applies to the following:

- Gocator 1300 series

Notational Conventions

This documentation 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.

Gocator Overview

Gocator laser displacement sensors are 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. Gocator sensors can also be configured using the provided development kits.

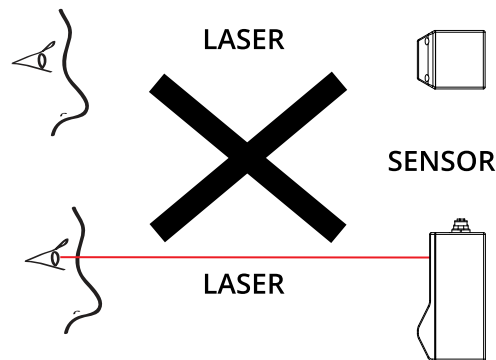
Safety and Maintenance

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

Laser Safety

Gocator sensors contain semiconductor lasers that emit visible or invisible light and are designated as Class 2, 2M, Class 3R, or Class 3B, depending on the chosen laser option. For more information on the laser classes used in Gocator sensors, *Laser Classes* on the next page.

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



**WARNING: DO NOT LOOK DIRECTLY
INTO THE LASER BEAM**

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

References

1. *International standard IEC 60825-1 (2001-08) consolidated edition*, Safety of laser products – Part 1: Equipment classification, requirements and user's guide.
2. *Technical report 60825-10*, Safety of laser products – Part 10. Application guidelines and explanatory notes to IEC 60825-1.
3. *Laser Notice No. 50*, FDA and CDRH (<https://www.fda.gov/Radiation-EmittingProducts/ElectronicProductRadiationControlProgram/default.htm>)

Laser Classes

Class 2 laser components

Class 2 laser components are considered to be safe, provided that:

- The user's blink reflex can terminate exposure (in under 0.25 seconds).

Class 2M laser components

Class 2M laser components should not cause permanent damage to the eye under reasonably foreseeable conditions of operation, provided that:

- No optical aids are used (these could focus the beam).
- The user's blink reflex can terminate exposure (in under 0.25 seconds).
- Users do not need to look repeatedly at the beam or reflected light.
- Exposure is only accidental.

Class 3R laser components

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

- Eye protection and protective clothing are not required.
- The laser beam must be terminated at the end of an appropriate path.
- Avoid unintentional reflections.
- Personnel must be trained in working with laser equipment.

Class 3B laser components

Class 3B components are unsafe for eye exposure.

- Usually only eye protection is required. Protective gloves may also be used.
- Diffuse reflections are safe if viewed for less than 10 seconds at a minimum distance of 13 cm.
- There is a risk of fire if the beam encounters flammable materials.
- The laser area must be clearly identified.
- Use a key switch or other mechanism to prevent unauthorized use.
- Use a clearly visible indicator to show that a laser is in use, such as "Laser in operation."
- Restrict the laser beam to the working area.
- Ensure that there are no reflective surfaces in the working area.



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

For more information, see *Precautions and Responsibilities* on the next page.

Precautions and Responsibilities

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

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

**LMI Class 3B laser components do not incorporate these laser safety items. These items must be added and completed by customers in their system design. For more information, see Class 3B Responsibilities below.*

Class 3B Responsibilities

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

Remote Interlock

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

Key Control

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

Power-On Delays

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

Beam Attenuators

A permanently attached method of preventing human access to laser radiation other than switches, power connectors or key control must be employed.

Emission Indicator

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

Warning Signs

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

Examples of laser warning signs are as follows:



FDA warning sign example



IEC warning sign example

Nominal Ocular Hazard Distance (NOHD)

In displacement sensors, the collimated light does not dissipate significantly over distance, so they retain the same laser class over this distance. For this reason, no NOHD is applicable.

Systems Sold or Used in the USA

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

Customers are responsible for achieving and maintaining this certification.

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

This publication, containing the full details of laser safety requirements, can be obtained directly from the FDA, or downloaded from their web site at <https://www.fda.gov/Radiation-EmittingProducts/ElectronicProductRadiationControlProgram/default.htm>.

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.

For a description of the connector pins, see *Gocator I/O Connector* on page 438.

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.

Handling, Cleaning, and Maintenance



Dirty or damaged sensor windows (emitter or camera) can affect accuracy. Use caution when handling the sensor or cleaning the sensor's windows.

Keep sensor windows clean

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

Turn off lasers when not in use

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

Avoid excessive modifications to files stored on the sensor

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

Environment and Lighting

Avoid strong ambient light sources

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

Avoid installing sensors in hazardous environments

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

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

Ensure that ambient conditions are within specifications

Gocator sensors are suitable for operation between 0–50° C and 25–85% relative humidity (non-condensing). Measurement error due to temperature is limited to 0.015% of full scale per degree C. The storage temperature is -30–70° C.

The Master network controllers are similarly rated for operation between 0–50° 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.

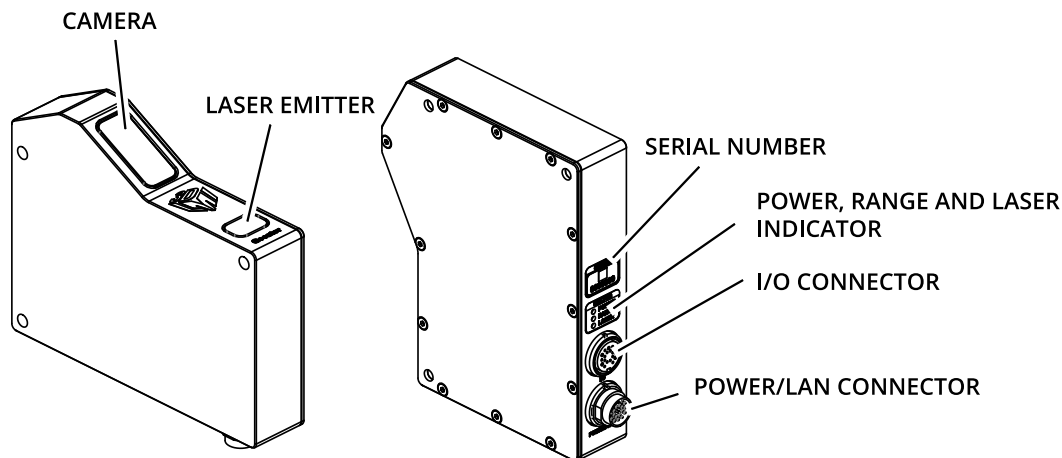
Getting Started

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

Hardware Overview

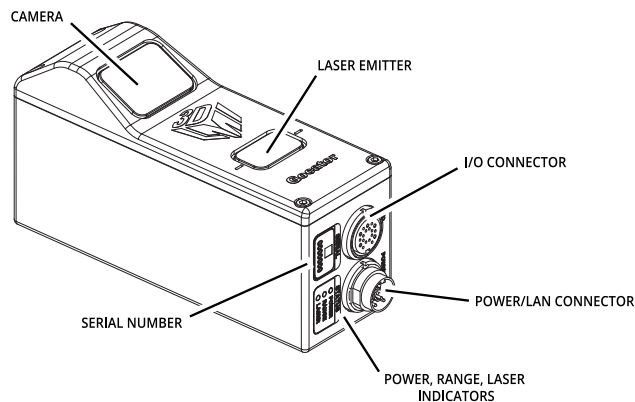
The following sections describe Gocator and its associated hardware.

Side Mount Package



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

Top Mount Package



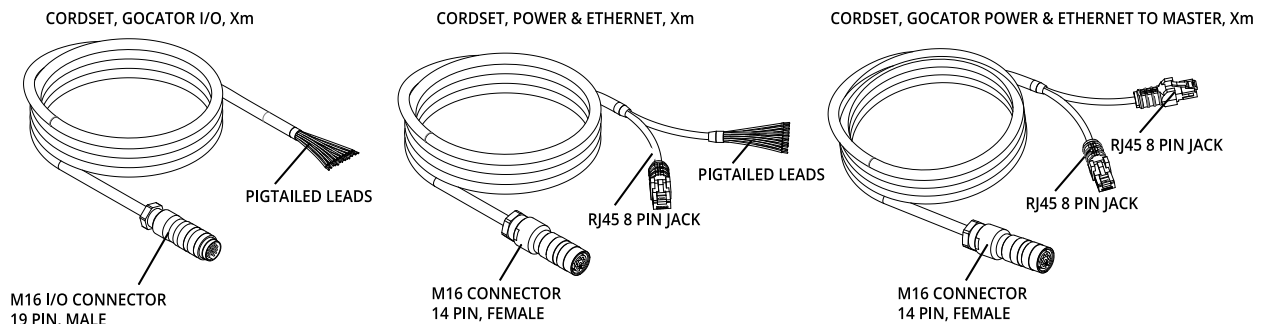
Item	Description
Camera	Observes laser light reflected from target surfaces.
Laser Emitter	Emits structured light for laser ranging.
I/O Connector	Accepts input and output signals.
Power / LAN Connector	Accepts power and laser safety signals and connects to 1000 Mbit/s Ethernet network.
Power Indicator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

Gocator Cordsets

Gocator 1300 sensors use two types of cordsets.

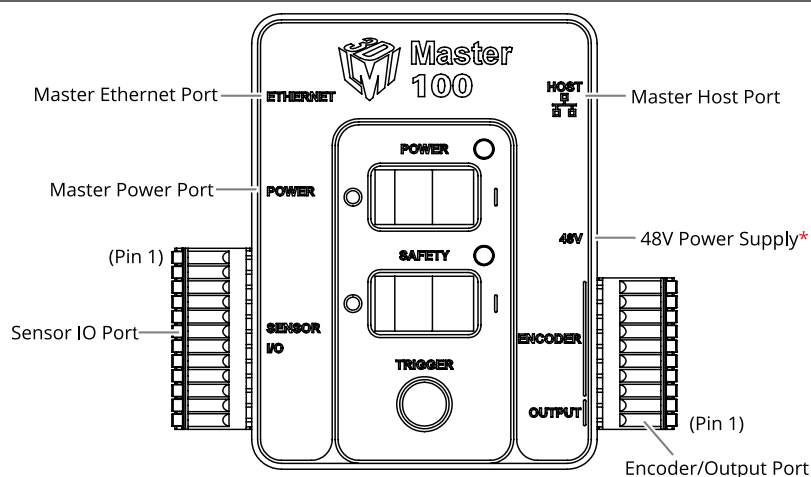
The Power & Ethernet cordset is used for sensor communication via 1000 Mbit/s Ethernet over a standard RJ45 connector. The Master version of the Power & Ethernet cordset provides electrical connection between the sensor and a [Master network controller](#) (excluding Master 100).

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



See *Accessories* on page 461 for cordset lengths and part numbers. Contact LMI for information on creating cordsets with customized lengths and connector orientations.

Master 100



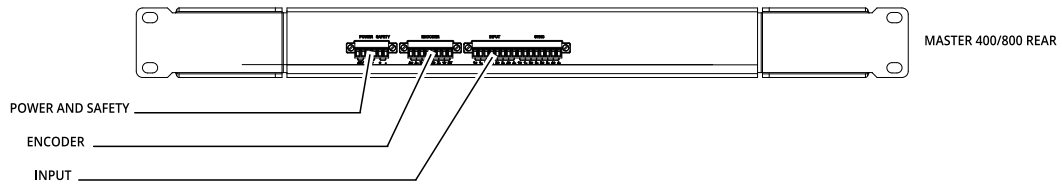
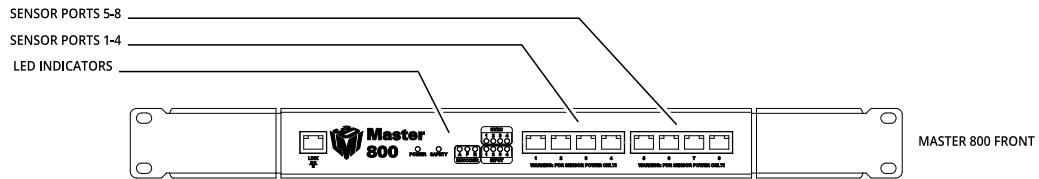
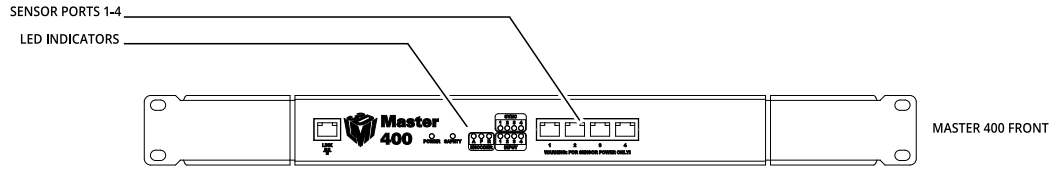
Item	Description
Master Ethernet Port	Connects to the RJ45 connector labeled Ethernet on the Power/LAN to Master cordset.
Master Power Port	Connects to the RJ45 connector labeled Power/Sync on the Power/LAN to Master cordset. Provides power and laser safety to the Gocator.
Sensor I/O Port	Connects to 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 443 for pinout details.

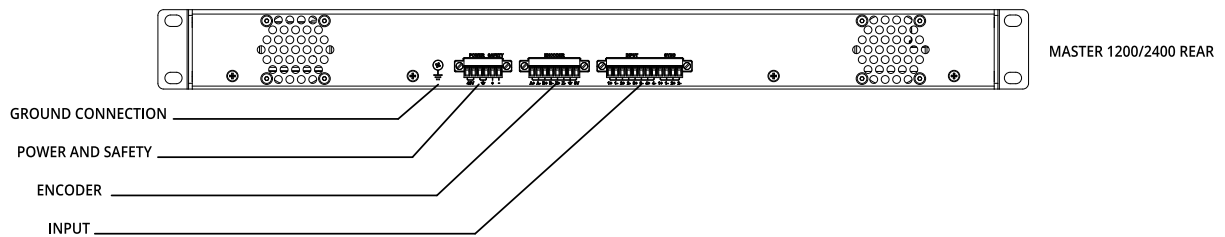
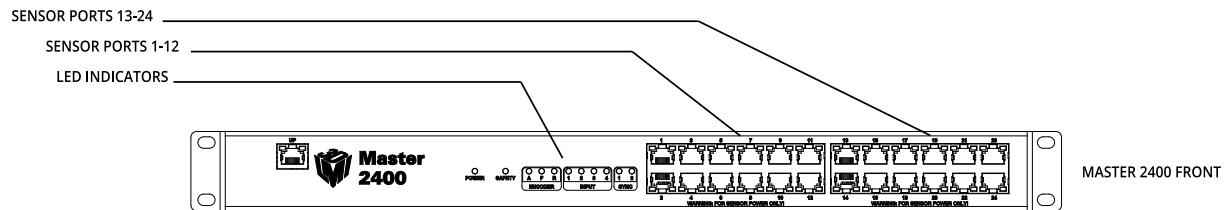
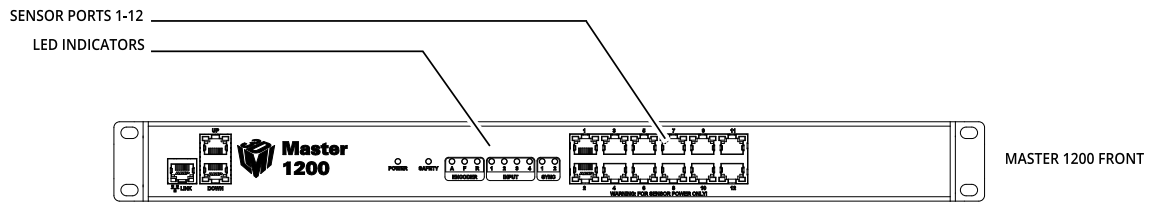
Master 400 / 800 / 1200 / 2400

The Master 400, 800, 1200, and 2400 network controllers let you connect more than two sensors:

- Master 400: accepts four sensors
- Master 800 accepts eight sensors
- Master 1200: accepts twelve sensors
- Master 2400: accepts twenty-four sensors



Master 400 and 800



Master 1200 and 2400

Item	Description
Sensor Ports	Master connection for Gocator sensors (no specific order required).

Item	Description
Ground Connection	Earth ground connection point.
Power and Safety	Power and laser safety connection.
Encoder	Accepts encoder signal.
Input	Accepts digital input.

For pinout details for Master 400 or 800, see *Master 400/800* on page 445.

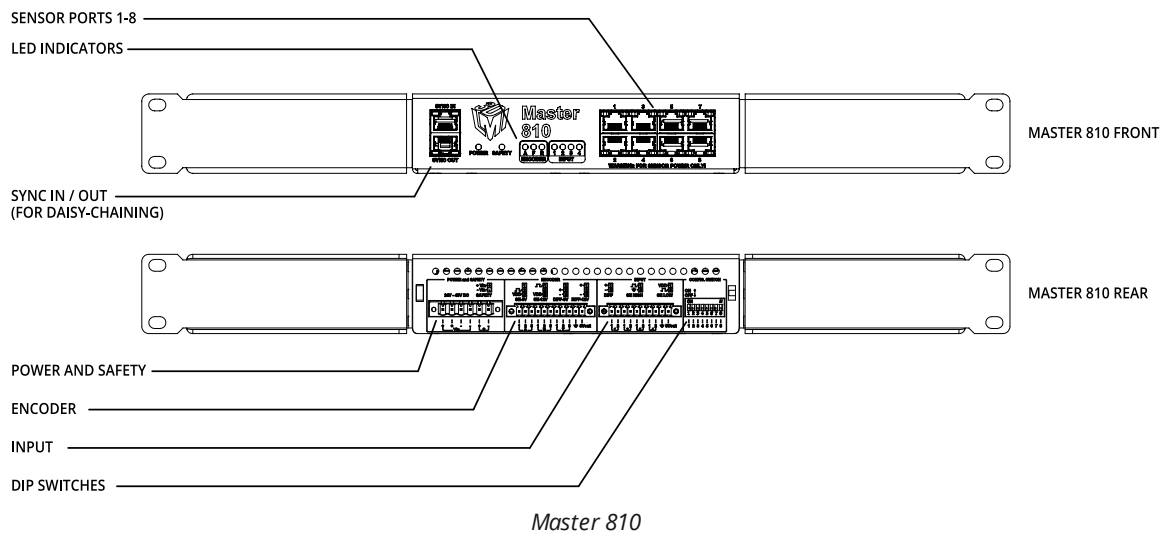
For pinout details for Master 1200 or 2400, see *Master 1200/2400* on page 458.

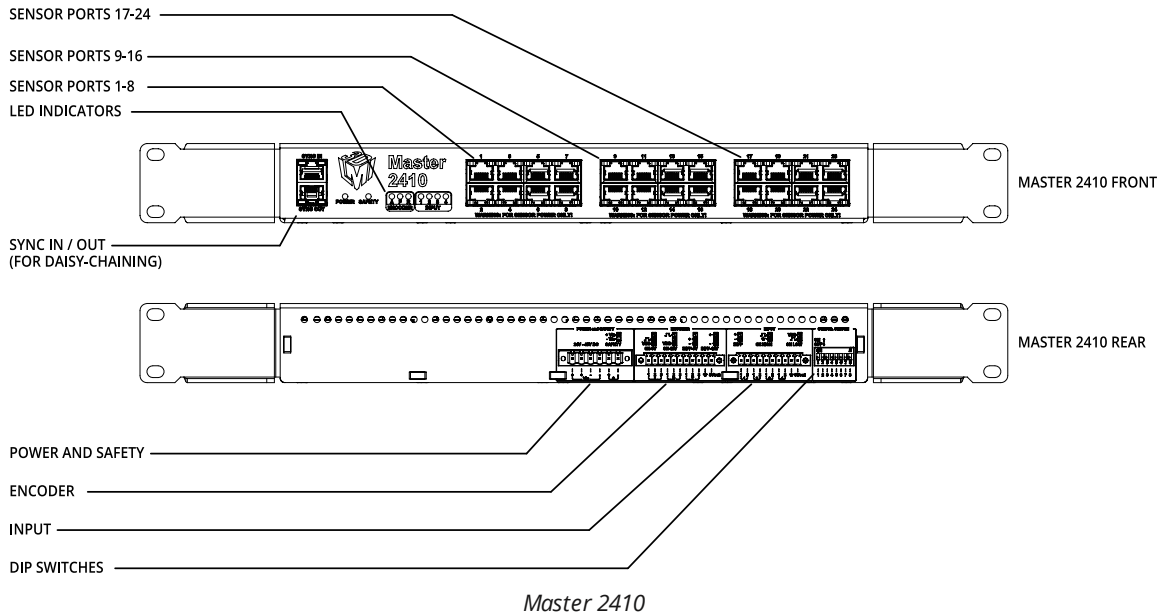
Master 810 / 2410

The Master 810 and 2410 network controllers let you connect multiple sensors to create a multi-sensor system:

- Master 810 accepts up to eight sensors
- Master 2410 accepts up to twenty-four sensors

Both models let you divide the quadrature frequency of a connected encoder to make the frequency compatible with the Master, and also set the debounce period to accommodate faster encoders. For more information, see *Configuring Master 810* on page 36. (Earlier revisions of these models lack the DIP switches.)



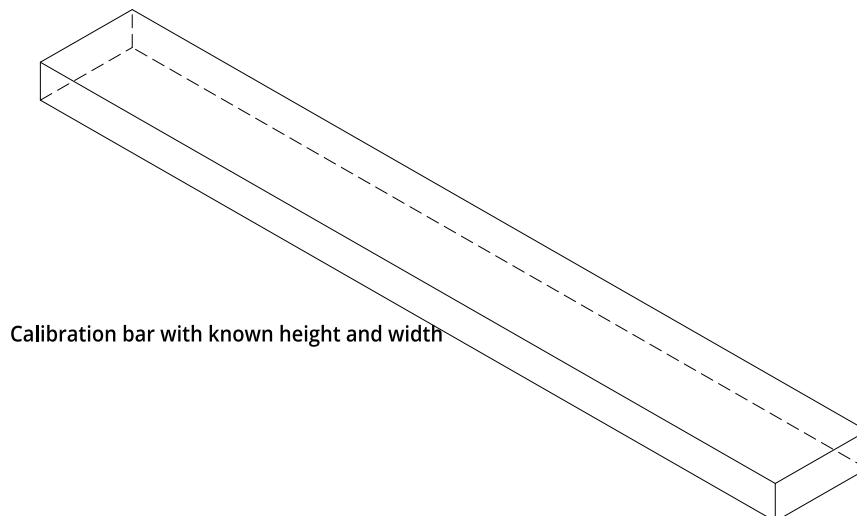


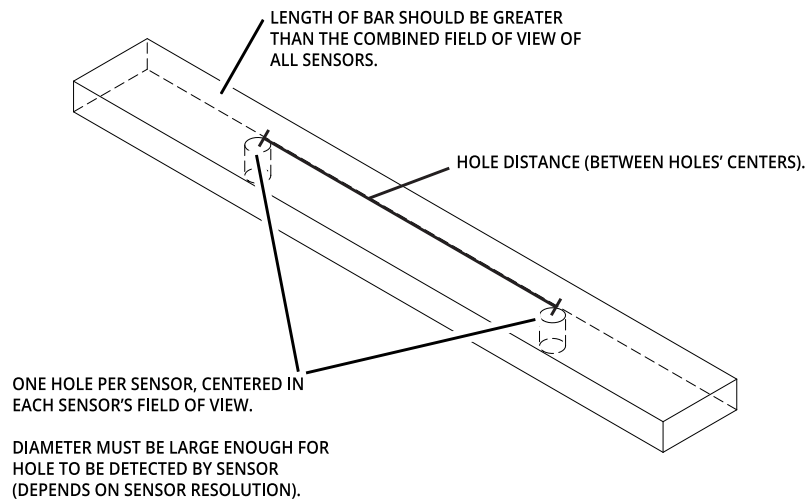
Item	Description
Sensor Ports	Master connection for Gocator sensors (no specific order required).
Power and Safety	Power and laser safety connection.
Encoder	Accepts encoder signal.
Input	Accepts digital input.
DIP Switches	Configures the Master (for example, allowing the device to work with faster encoders). For information on configuring Master 810 and 2410 using the DIP switches, see <i>Configuring Master 810</i> on page 36.

For pinout details, see *Master 810/2410* on page 449.

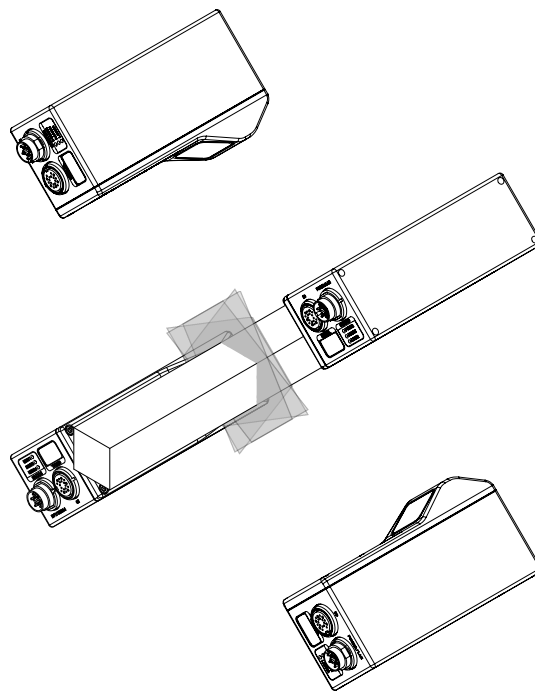
Alignment Targets

Targets are used for aligning sensors and calibrating transport systems.





For multi-sensor systems in a ring layout, use a polygon-shaped alignment target. The number of corners in the target should correspond with the number of sensors in the system. Sensors should be positioned so that each sensor can scan a corner and surrounding surface.



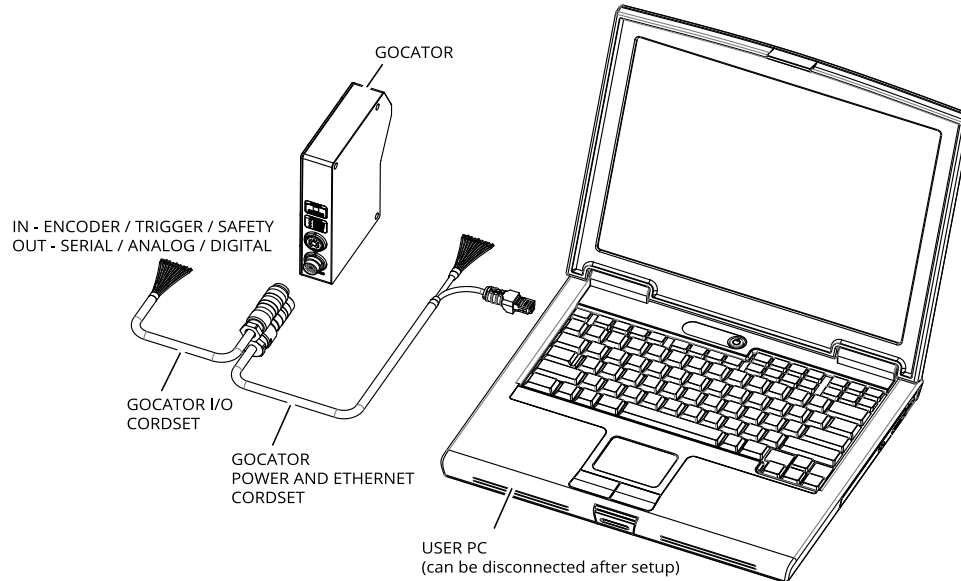
For more information on alignment, see *Aligning Sensors* on page 100.

System Overview

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

Standalone System

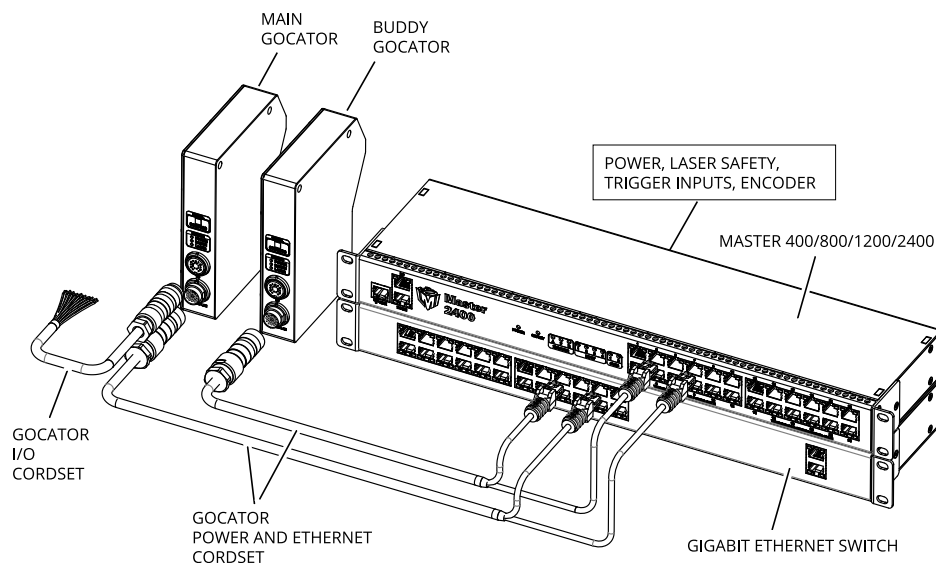
Standalone systems are typically used when only a single Gocator is required. The device can be connected to a computer's Ethernet port for setup and can also be connected to devices such as encoders, photocells, or PLCs.



Dual-Sensor System

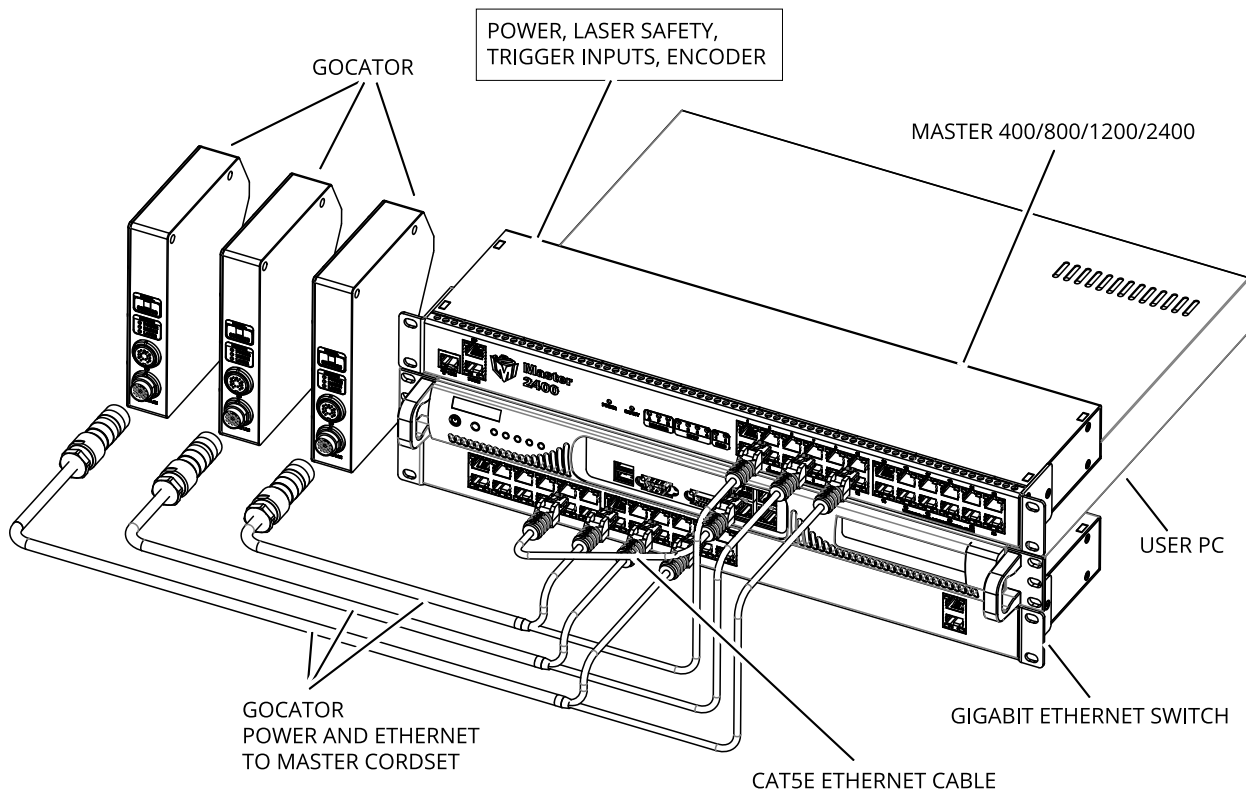
In a dual-sensor system, two Gocator sensors work together to perform ranging and output the combined results. The controlling sensor is referred to as the *Main* sensor, and the other sensor is referred to as the *Buddy* sensor. Gocator's software recognizes three installation orientations: *Opposite*,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

A [Master network controller](#) (excluding Master 100) must be used to connect two sensors in a dual-sensor system. Gocator Master cordsets are used to connect sensors to the Master.



Multi-Sensor System

A [Master network controller](#) (excluding Master 100) 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 400/800/810/1200/2400/2410 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).



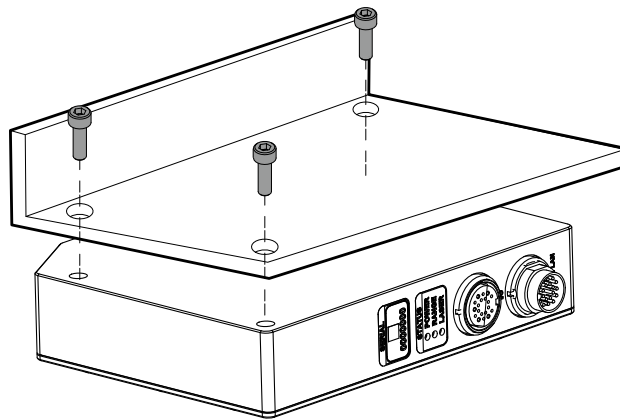
Installation

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

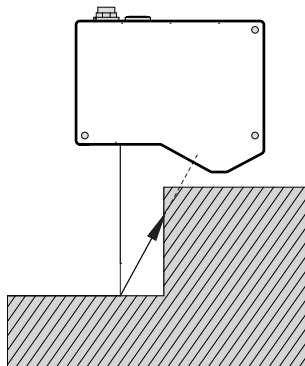
Mounting: Side Mount Package

Single point sensors are often mounted with the triangulation base perpendicular to the travel direction to avoid occlusions.

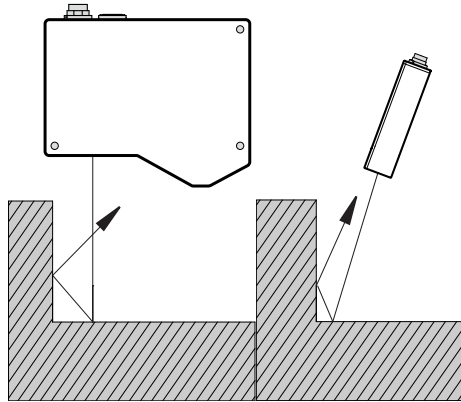
Sensors should be mounted using M6 x 1.0 pitch screws of suitable length. The recommended thread engagement into the housing is 8-10 mm. Proper care should be taken in order to ensure that the internal threads are not damaged from cross-threading or improper insertion of screws.



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



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



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



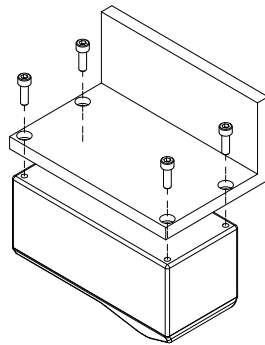
Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

Mounting - Top Mount Package

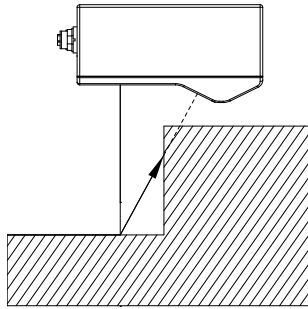


Single point sensors are often mounted with the triangulation base perpendicular to the travel direction to avoid occlusions.

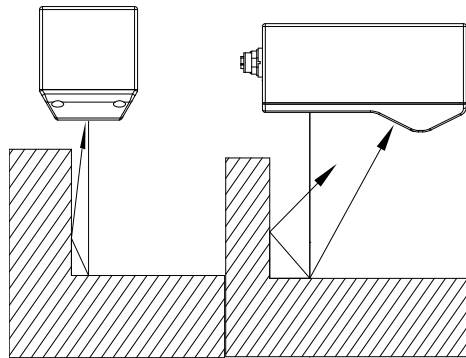
Sensors should be mounted using four M5 x 0.8 pitch screws of suitable length. The recommended thread engagement into the housing is 8-10 mm. Proper care should be taken in order to ensure that the internal threads are not damaged from cross-threading or improper insertion of screws.



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



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



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



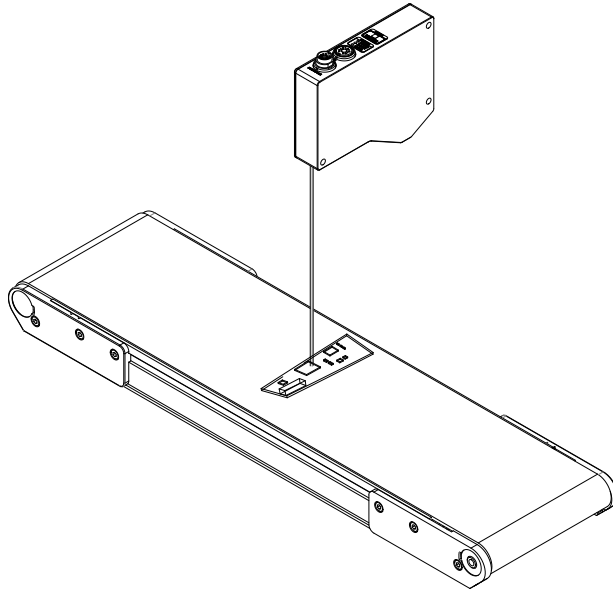
Gocator sensors are high-accuracy devices. The temperature of all of its components must be in equilibrium. When the sensor is powered up, a warm-up time of at least one hour is required to reach a consistent spread of temperature within the sensor.

Orientations

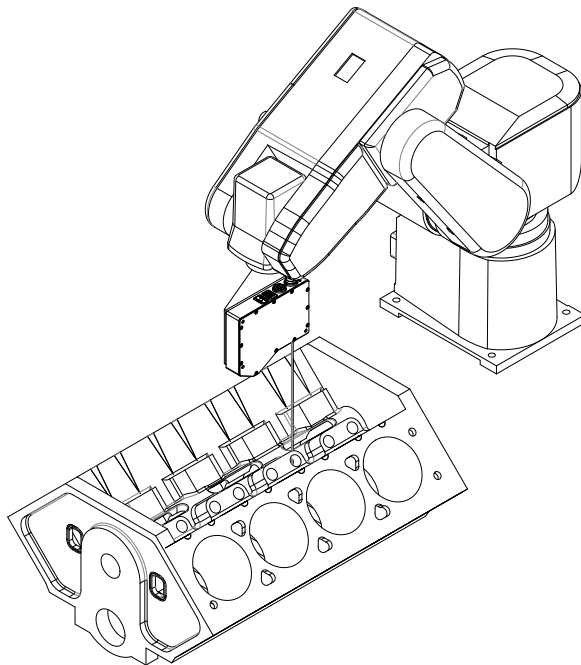
The examples below illustrate the possible mounting orientations for standalone and dual-sensor systems.

See *Layout* on page 70 for more information on orientations.

Standalone Orientations

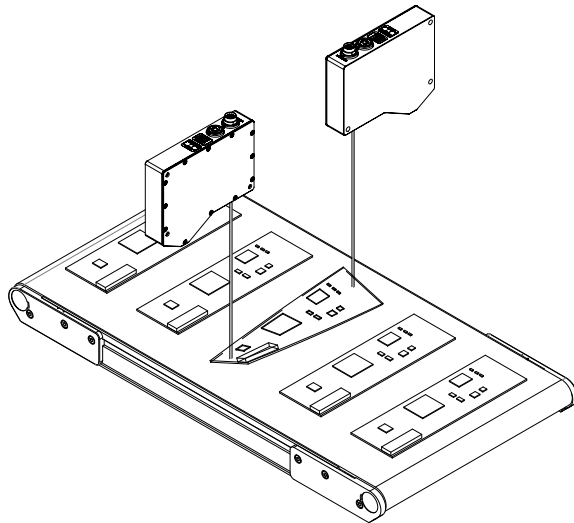


Single sensor above conveyor

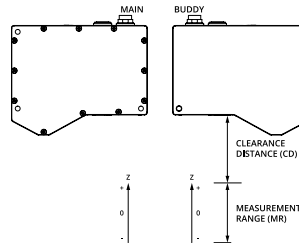


Single sensor on robot arm

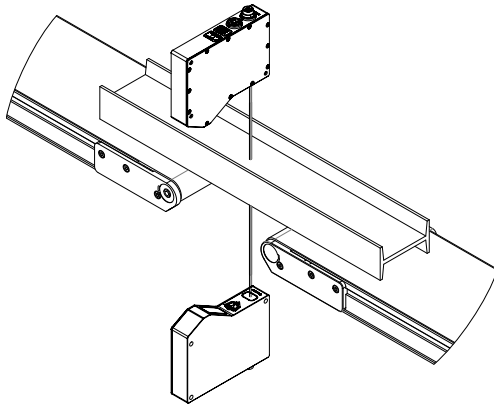
Dual-Sensor System Orientations:



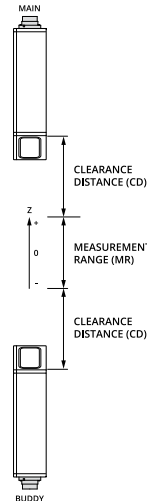
Side-by-side for wide-area measurement (Wide)



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



Above/below for two-sided measurement (Opposite)



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

Grounding

Components of a Gocator system should be properly grounded.

Gocator

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



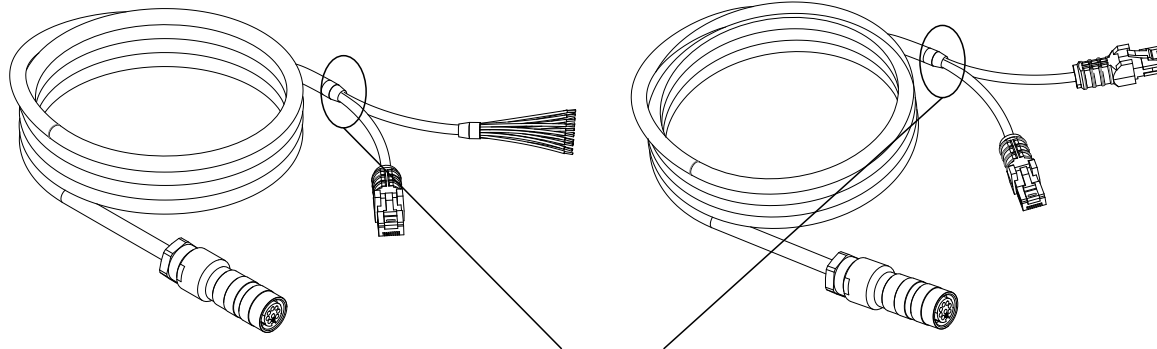
The frame or electrical cabinet that the Gocator is mounted to must be connected to earth ground.

Recommended Practices for 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.



Master Network Controllers

The rack mount brackets provided with all Masters are 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.



When using the rack mount brackets, you *must* connect the frame or electrical cabinet to which the Master is mounted to earth ground.

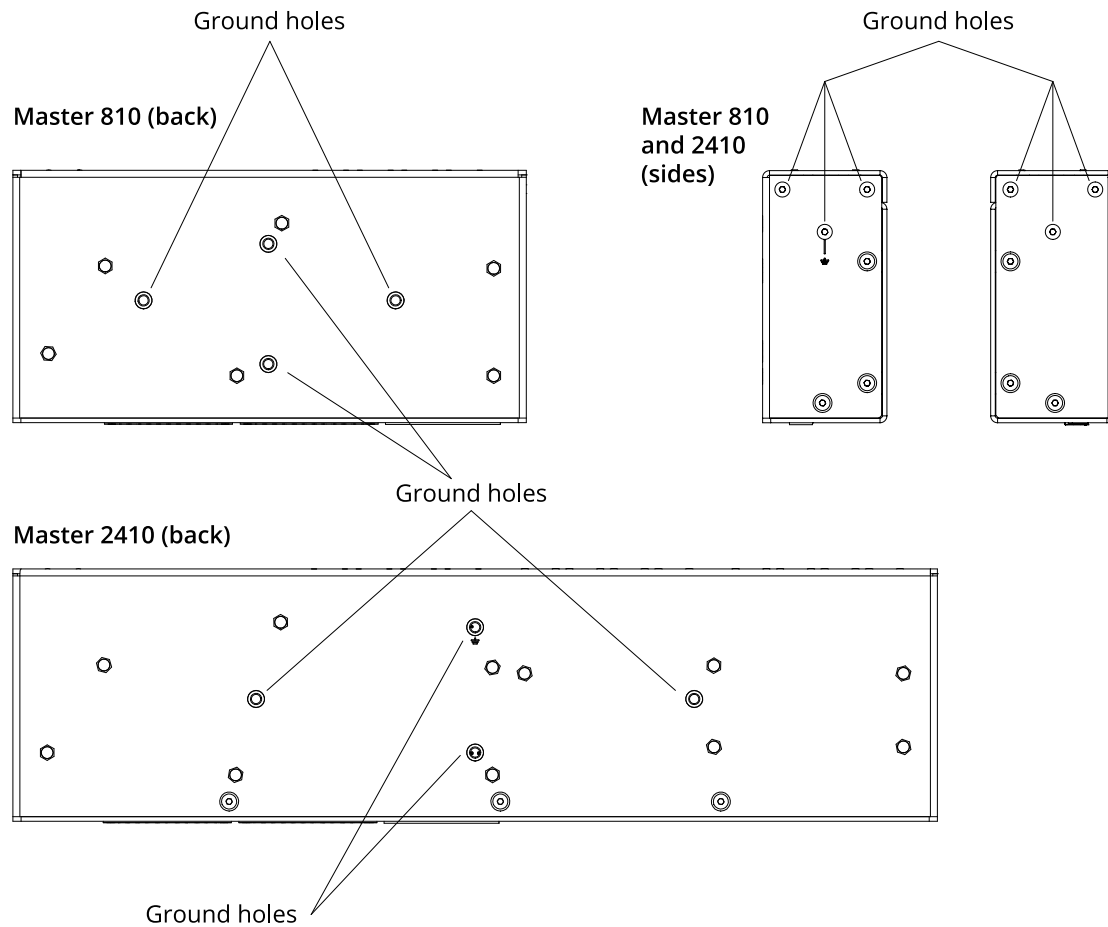


You *must* check electrical continuity between the mounting frame and RJ45 connectors on the front using a multi-meter.

If you are mounting Master 810 or 2410 using the provided DIN rail mount adapters, you must ground the Master directly; for more information, see *Grounding When Using a DIN Rail (Master 810/2410)* below.

Grounding When Using a DIN Rail (Master 810/2410)

If you are using DIN rail adapters instead of the rack mount brackets, you must ensure that the Master is properly grounded by connecting a ground cable to one of the holes indicated below. The holes accept M4x5 screws.



Installing DIN Rail Clips: Master 810 or 2410

You can mount the Master 810 and 2410 using the included DIN rail mounting clips with M4x8 flat socket cap screws. The following DIN rail clips ([DINM12-RC](#)) are included:

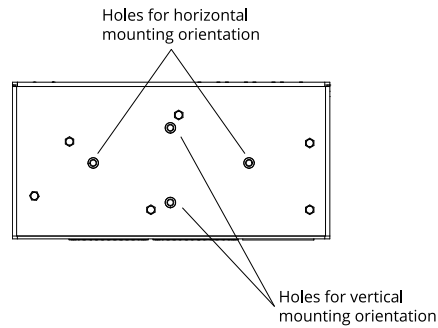


To install the DIN rail clips:

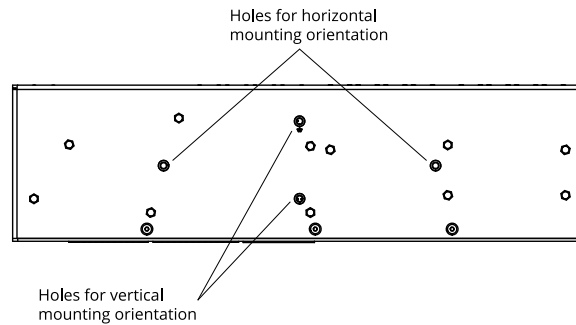
1. Remove the 1U rack mount brackets.

2. Locate the DIN rail mounting holes on the back of the Master (see below).

Master 810:

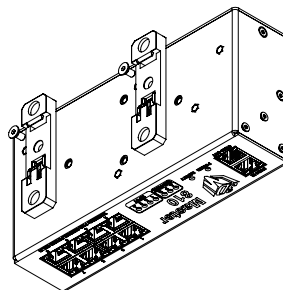


Master 2410:



3. Attach each of the two DIN rail mount clips to the back of the Master using an M4x8 flat socket cap screw for each one.

The following illustration shows the installation of clips on a Master 810 for horizontal mounting:



Ensure that there is enough clearance around the Master for cabling.

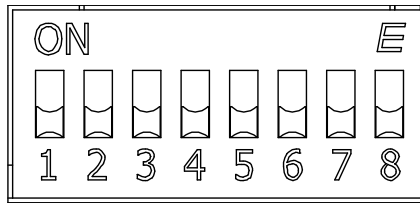
Configuring Master 810

If you are using Master 810 with an encoder that runs at a quadrature frequency higher than 300 kHz, you must use the device's divider DIP switches to limit the incoming frequency to 300 kHz.



Master 810 supports up to a maximum incoming encoder quadrature frequency of 6.5 MHz.

The DIP switches are located on the rear of the device.



 Switches 5 to 8 are reserved for future use.

This section describes how to set the DIP switches on Master 810 to do the following:

- Set the divider so that the quadrature frequency of the connected encoder is compatible with the Master.
- Set the debounce period to accommodate faster encoders.

Setting the Divider

To set the divider, you use switches 1 to 3. To determine which divider to use, use the following formula:

$$\text{Output Quadrature Frequency} = \text{Input Quadrature Frequency} / \text{Divider}$$

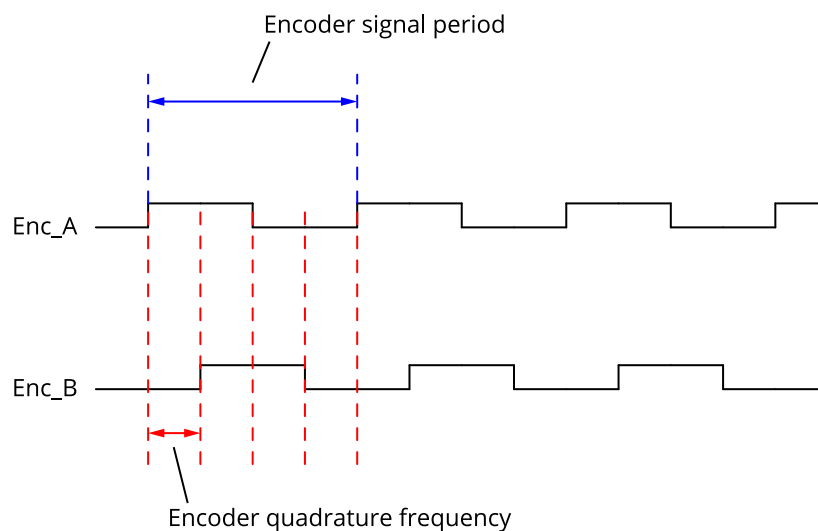
In the formula, use the *quadrature frequency* of the encoder (for more information, see *Encoder Quadrature Frequency* below) and a divider from the following table so that the Output Quadrature Frequency is no more than 300 kHz.

Divider	Switch 1	Switch 2	Switch 3
1	OFF	OFF	OFF
2	ON	OFF	OFF
4	OFF	ON	OFF
8	ON	ON	OFF
16	OFF	OFF	ON
32	ON	OFF	ON
64	OFF	ON	ON
128	ON	ON	ON

 The divider works on debounced encoder signals. For more information, see *Setting the Debounce Period* on the next page.

Encoder Quadrature Frequency

Encoder quadrature frequency is defined as illustrated in the following diagram. It is the frequency of encoder ticks. This may also be referred as the native encoder rate.



You must use a quadrature frequency when determining which divider to use (see *Setting the Divider* on the previous page). Consult the datasheet of the encoder you are using to determine its quadrature frequency.



Some encoders may be specified in terms of encoder signal frequency (or period). In this case, convert the signal frequency to quadrature frequency by multiplying the signal frequency by 4.

Setting the Debounce Period

If the quadrature frequency of the encoder you are using is greater than 3 MHz, you must set the debounce period to “short.” Otherwise, set the debounce period to “long.”

You use switch 4 to set the debounce period.

Debounce period	Switch 4
short debounce	ON
long debounce	OFF

Network Setup

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



DHCP is not recommended for Gocator sensors. If you choose to use DHCP, the DHCP server should try to preserve IP addresses. Ideally, you should use static IP address assignment (by MAC address) to do this.

Client Setup

To connect to a sensor from a client PC, you must ensure the client's network card is properly configured.

Sensors are shipped with the following default network configuration:

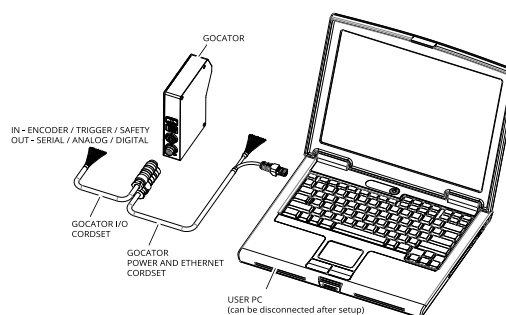
Setting	Default
DHCP	Disabled
IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Gateway	0.0.0.0



All Gocator sensors are configured to 192.168.1.10 as the default IP address. For a dual-sensor system, the Main and Buddy sensors must be assigned unique addresses before they can be used on the same network. Before proceeding, connect the Main and Buddy sensors one at a time (to avoid an address conflict) and use the steps in See *Running a Dual-Sensor System* on page 42 to assign each sensor a unique address.

To connect to a sensor for the first time:

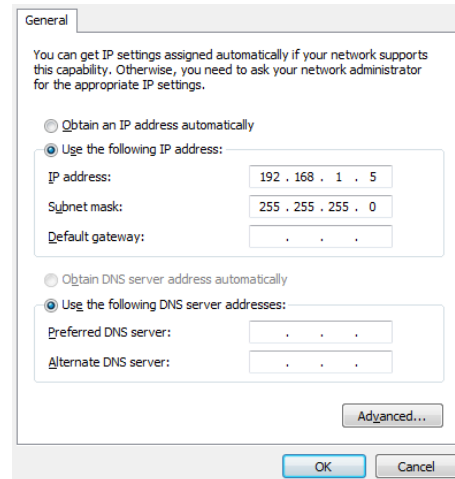
1. Connect cables and apply power.
Sensor cabling is illustrated in *System Overview* on page 26.



2. Change the client PC's network settings.

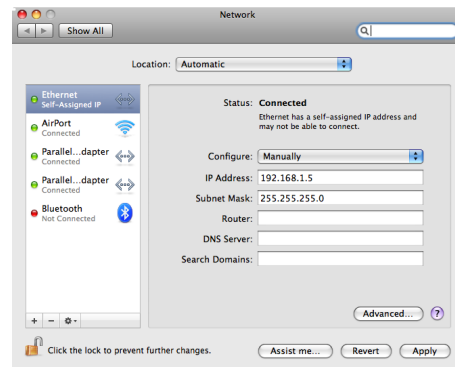
Windows 7

- 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 409 if you experience any problems while attempting to establish a connection to the sensor.

Gocator Setup

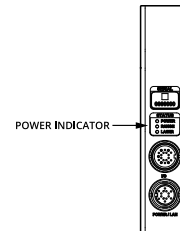
The Gocator is shipped with a default configuration that will produce laser ranges for most targets.

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

Running a Standalone Sensor System

To configure a standalone sensor system:

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



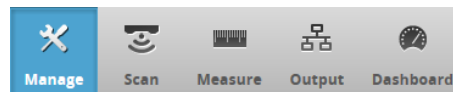
2. Enter the sensor's IP address (192.168.1.10) in a web browser.

The Gocator interface loads.

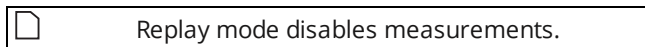
If a password has been set, you will be prompted to provide it and then log in.



3. Go to the **Manage** page.



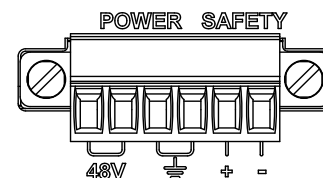
4. Ensure that Replay mode is off (the slider is set to the left).



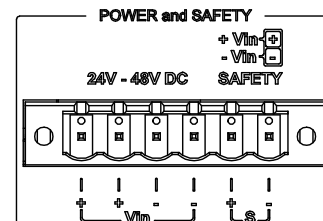
5. Ensure that the Laser Safety Switch is enabled or the Laser Safety input is high.
6. Go to the **Scan** page.
7. Observe the profile in the data viewer
8. Press the **Start** button or the **Snapshot** on the **Toolbar** to start the sensor.

The **Start** button is used to run sensors continuously.

The **Snapshot** button is used to trigger the capture of a single range.

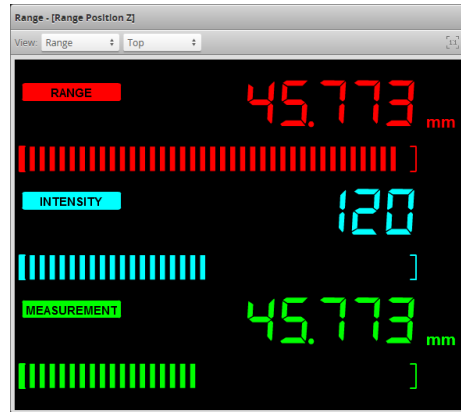


Master 400/800/1200/2400



Master 810/2410

9. Move a target into the laser plane.
If a target object is within the sensor's measurement range, the data viewer will display the distance to the target, and the sensor's range indicator will illuminate. If you cannot see the laser, or if a range is not displayed in the Data Viewer, see *Troubleshooting* on page 409.



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

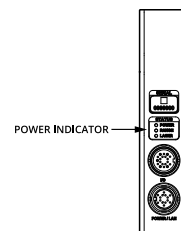
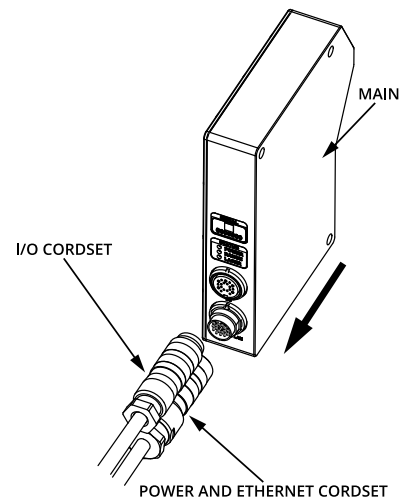


Running a Dual-Sensor System

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

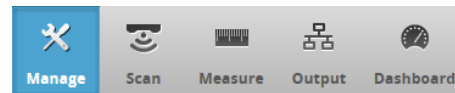
To configure a dual-sensor system:

1. Turn off the sensors and unplug the Ethernet network connection of the Main sensor.
All sensors are shipped with a default IP address of 192.168.1.10. Ethernet networks require a unique IP address for each device. Skip step 1 to 3 if the Buddy sensor's IP address is already set up with a unique address.
2. Power up the Buddy sensor.
The power LED (blue) of the Buddy sensor should turn on immediately.
3. Enter the sensor's IP address 192.168.1.10 in a web browser.

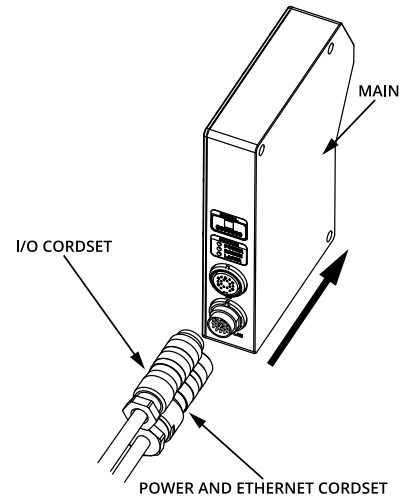


The Gocator interface loads.

4. Go to the **Manage** Page.
5. Modify the IP address to 192.168.1.11 in the **Networking** category and click the **Save** button.
When you click the **Save** button, you will be prompted to confirm your selection.
6. Turn off the sensors, re-connect the Main sensor's Ethernet connection and power-cycle the sensors.
After changing network configuration, the sensors must be reset or power-cycled before the change will take effect.

A screenshot of the 'Networking' configuration form. It has a 'Type' dropdown set to 'Manual'. Below are input fields for 'IP' (192.168.1.11), 'Subnet Mask' (255.255.255.0), and 'Gateway' (0.0.0.0). A 'Save' button is at the bottom right.

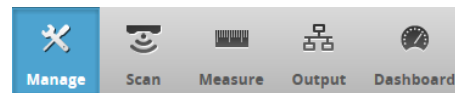
Type:	Manual
IP:	192.168.1.11
Subnet Mask:	255.255.255.0
Gateway:	0.0.0.0
<button>Save</button>	



7. Enter the sensor's IP address 192.168.1.10 in a web browser.
The Gocator interface loads.



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

A screenshot of the 'Visible Sensors' panel. It contains a table with the following data:

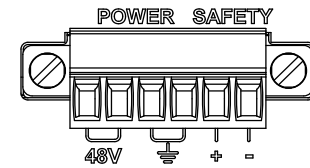
Serial	Model	Version	State
12079	Gocator 1350	4.0.9.94	Ready

Below the table is an 'Assign' button.

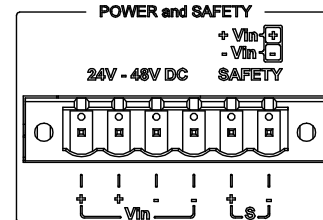
10. Select the Buddy sensor and click the **Assign** button.
The Buddy sensor will be assigned to the Main sensor and its status will be updated in the System panel.
The firmware on Main and Buddy sensors must be the same for Buddy assignment to be successful. If the firmware is different, connect the Main and Buddy sensor one at a time and follow the steps in *Firmware Upgrade* on

page 80 to upgrade the sensors.

11. Ensure that the Laser Safety Switch is enabled or the Laser Safety input is high.



Master 400/800/1200/2400



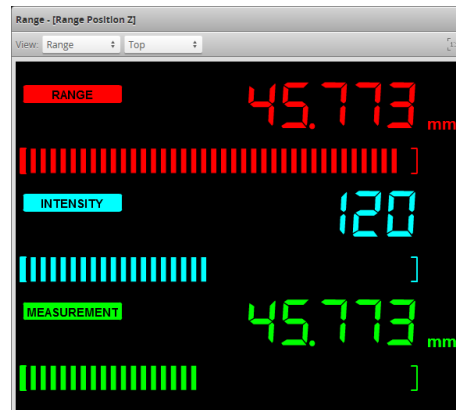
Master 810/2410

12. Ensure that **Replay** mode is off (the slider is set to the left).
13. Go to the the **Scan** page.
14. Press the **Start** or the **Snapshot** button on the **Toolbar** to start the sensors.

The **Start** button is used to run sensors continuously, while the **Snapshot** button is used to trigger a single measurement.



15. Move a target into the laser plane.
If a target object is within the sensor's measurement range, the data viewer will display the distance to the target, and the sensor's range indicator will illuminate.
If you cannot see the laser, or if a range is not displayed in the Data Viewer, see *Troubleshooting* on page 409.



16. Press the **Stop** button if you used the **Start** button to start the sensors.

The laser should turn off.



Next Steps

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

Management and Maintenance (page 67)

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

Scan Setup and Alignment (page 84)

Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment.

Measurement (page 120)

Contains built-in measurement tools and their settings.

Output (page 189)

Contains settings for configuring output protocols used to communicate measurements to external devices.

Dashboard (page 201)

Provides monitoring of measurement statistics and sensor health.

Toolbar (page 56)

Controls sensor operation, manages jobs, and replays recorded measurement data.

How Gocator Works

The following sections provide an overview of how Gocator acquires and produces data, detects and measures parts, and controls devices such as PLCs. Some of these concepts are important for understanding how you should mount sensors and configure settings such as active area.



You can use the Gocator Accelerator to speed up processing of data. For more information, see *Gocator Accelerator* on page 220.

3D Acquisition

After a Gocator system has been set up and is running, it is ready to start capturing 3D data.

Gocator laser displacement sensors project a laser point onto the target.



The sensor's camera views the laser line on the target from an angle and captures the reflection of the laser light off the target. The camera captures a single 3D range for each camera exposure. The reflected

laser light falls on the camera at different positions, depending on the distance of the target from the sensor. The sensor's laser emitter, its camera, and the target form a triangle. Gocator uses the known distance between the laser emitter and the camera, and two known angles—one of which depends on the position of the laser light on the camera—to calculate the distance from the sensor to the target. This translates to the height of the target. This method of calculating distance is called *laser triangulation*.



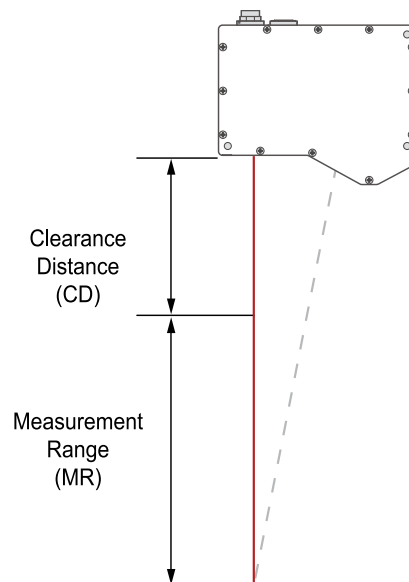
Gocator sensors are always pre-calibrated to deliver 3D data in engineering units throughout their measurement range.

Clearance Distance and Measurement Range

Clearance distance (CD) and measurement range (MR) are important concepts for understanding the setup of a Gocator sensor and for understanding results.

Clearance distance – The minimum distance from the sensor that a target can be scanned and measured. A target closer than this distance will result in invalid data.

Measurement range – The vertical distance, starting at the end of the clearance distance, in which targets can be scanned and measured. Targets beyond the measurement range will result in invalid data.

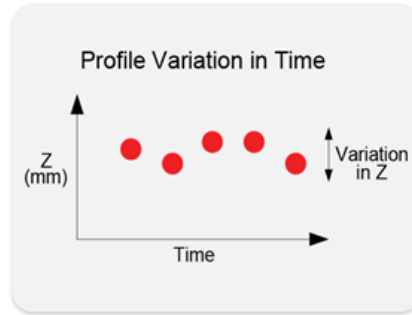


Resolution and Accuracy

The following sections describe Z Resolution and Z Linearity. These terms are used in the Gocator datasheets to describe the measurement capabilities of the sensors.

Z Resolution

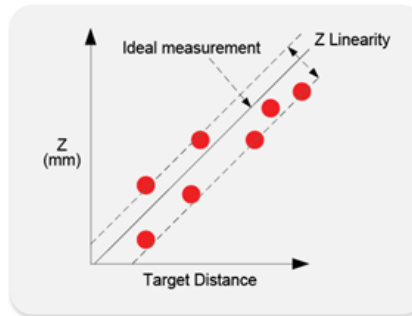
Z Resolution gives an indication of the smallest detectable height difference at each point, or how accurately height on a target can be measured. Variability of height measurements at any given moment, in each individual 3D point, with the target at a fixed position, limits Z resolution. This variability is caused by camera and sensor electronics.



Z resolution is better closer to the sensor. This is reflected in the Gocator data sheet as the two numbers quoted for Z resolution.

Z Linearity

Z linearity is the difference between the actual distance to the target and the measured distance to the target, throughout the measurement range. Z linearity gives an indication of the sensor's ability to measure absolute distance.



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

Range Output

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

Coordinate Systems

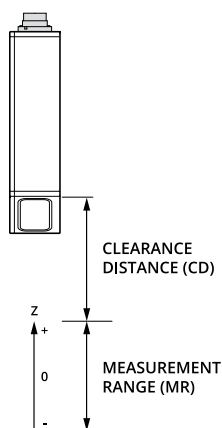
Range data is reported in one of three coordinate systems, which generally depends on the alignment state of the sensor.

- **Sensor coordinates:** Used on unaligned sensors.
- **System coordinates:** Used on aligned sensors. Applies to either standalone or multi-sensor systems.
- **Part coordinates:** Data can optionally be reported using a coordinate system relative to the part itself.

These coordinate systems are described below.

Sensor Coordinates

Unaligned sensors use *sensor* coordinates: The measurement range (MR) is along the Z axis. Most importantly, the origin is at the *center* of the measurement range.



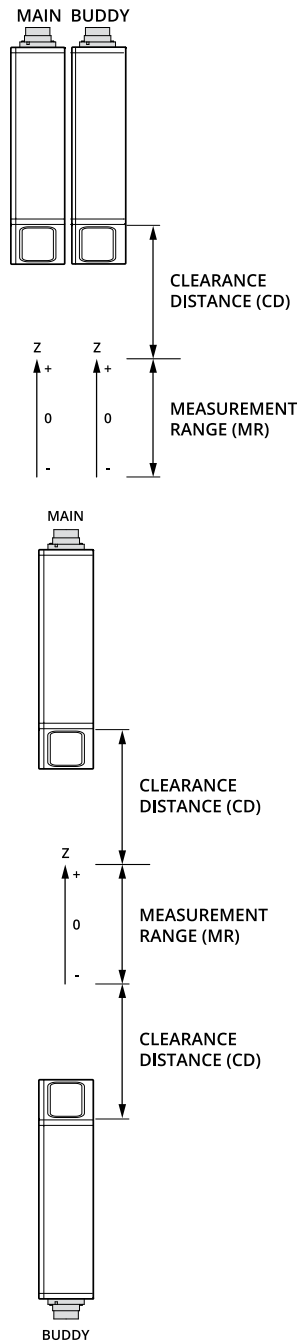
System Coordinates

Aligning sensors adjusts the coordinate system in relation to sensor coordinates, resulting in *system coordinates* (for more information on sensor coordinates, see *Sensor Coordinates* above). For more information on aligning sensors, see *Alignment* on page 100.

The adjustments resulting from alignment are called *transformations* (offsets along the axes and rotations around the axes). Transformations are displayed in the **Sensor** panel on the **Scan** page. For more information on transformations in the web interface, see *Transformations* on page 94.

Alignment is used with a single sensor to compensate for mounting misalignment and to set a zero reference, such as a conveyor belt surface.

Additionally, in multi-sensor systems, alignment sets a common coordinate system. That is, scan data and measurements from the sensors are expressed in a unified coordinate system.



Y angle is positive when rotating from positive X to positive Z axis.

X angle is positive when rotating from positive Y to positive Z. Z angle is positive when rotating from positive X to positive Y.

When applying the transformations, angular rotation is applied before the Z offset.

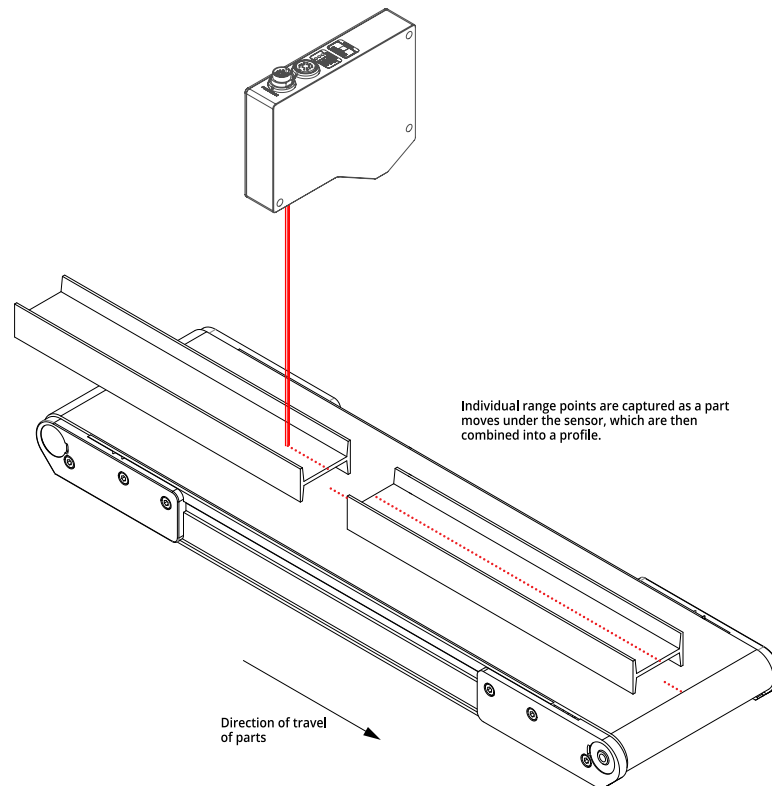
Data Generation and Processing

After scanning a target, Gocator can process the scan data to allow the use of more sophisticated measurement tools. This section describes the following concepts:

- Profile generation
- Part detection

Profile Generation

Gocator 1300 series, which are displacement sensors and only return a single range value, can combine a series of range values gathered as a target moves under the sensor to generate a profile.



You can then use all the standard [profile measurement tools](#) on the resulting profile:



Part Detection

After Gocator has generated a profile by combining single exposures into larger pieces of data, the firmware can isolate discrete parts in a generated profile into separate profiles representing parts. Gocator can then perform measurements on these isolated parts.

For more information on part detection, see *Part Detection* on page 111.

Measurement and Anchoring

After Gocator scans a target and, optionally, [further processes](#) the data, the sensor is ready to take measurements on the scan data.

Gocator provides several measurement tools, each of which provides a set of individual measurements, giving you dozens of measurements ideal for a wide variety of applications to choose from. The configured measurements start returning pass/fail decisions, as well as the actual measured values, which are then sent over the enabled [output](#) channels to control devices such as PLCs, which can in turn control ejection or sorting mechanisms. (For more information on measurements and configuring measurements, see *Measurement* on page 120.)



You can create custom measurement tools that run your own algorithms. For more information, see *GDK* on page 388.

A part's position can vary on a transport system. To compensate for this variation, Gocator can anchor a measurement to the positional measurement (X or Z) of an easily detectable feature. The calculated offset between the two ensures that the anchored measurement will always be properly positioned on different parts.

For more information on anchoring, see *Measurement Anchoring* on page 131.

Output and Digital Tracking

After Gocator has scanned and measured parts, the last step in the operation flow is to output the results and/or measurements.

One of the main functions of Gocator sensors is to produce pass/fail decisions, and then control something based on that decision. Typically, this involves rejecting a part through an eject gate, but it can also involve making decisions on good, but different, parts. This is described as “output” in Gocator. Gocator supports the following output types:

- Ethernet (which provides industry-standard protocols such as Modbus, EtherNet/IP, and ASCII, in addition to the Gocator protocol)
- Digital
- Analog
- Serial interfaces

An important concept is digital output tracking. Production lines can place an ejection or sorting mechanism at different distances from where the sensor scans the target. For this reason, Gocator lets you schedule a delayed decision over the digital interfaces. Because the conveyor system on a typical production line will use an encoder or have a known, constant speed, targets can effectively be “tracked” or “tagged.” Gocator will know when a defective part has traveled far enough and trigger a PLC to activate an ejection/sorting mechanism at the correct moment. For more information on digital output tracking, see *Digital Output* on page 194.

Gocator Web Interface

The following sections describe the Gocator web interface.

Browser Compatibility

LMI recommends Chrome, Firefox, or Edge for use with the Gocator web interface.

Internet Explorer 11 is supported with limitations; for more information, see below.

Internet Explorer 11 Issues

If you use Gocator with large datasets on Internet Explorer 11, you may encounter the following issues.

Internet Explorer Switches to Software Rendering

If the PC connected to a Gocator sensor is busy, Internet Explorer may switch to software rendering after a specific amount of time. If this occurs, data is not displayed in the data viewer, and the only reliable way to recover from the situation is to restart the browser.

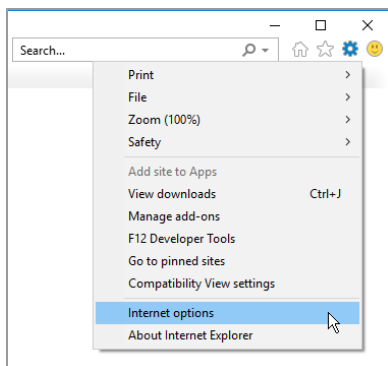
It is possible to remove the time limit that causes this issue, but you must modify the computer's registry. To do so, follow Microsoft's instructions at <https://support.microsoft.com/en-us/help/3099259/update-to-add-a-setting-to-disable-500-msec-time-limit-for-webgl-frame>.

Internet Explorer Displays "Out of Memory"

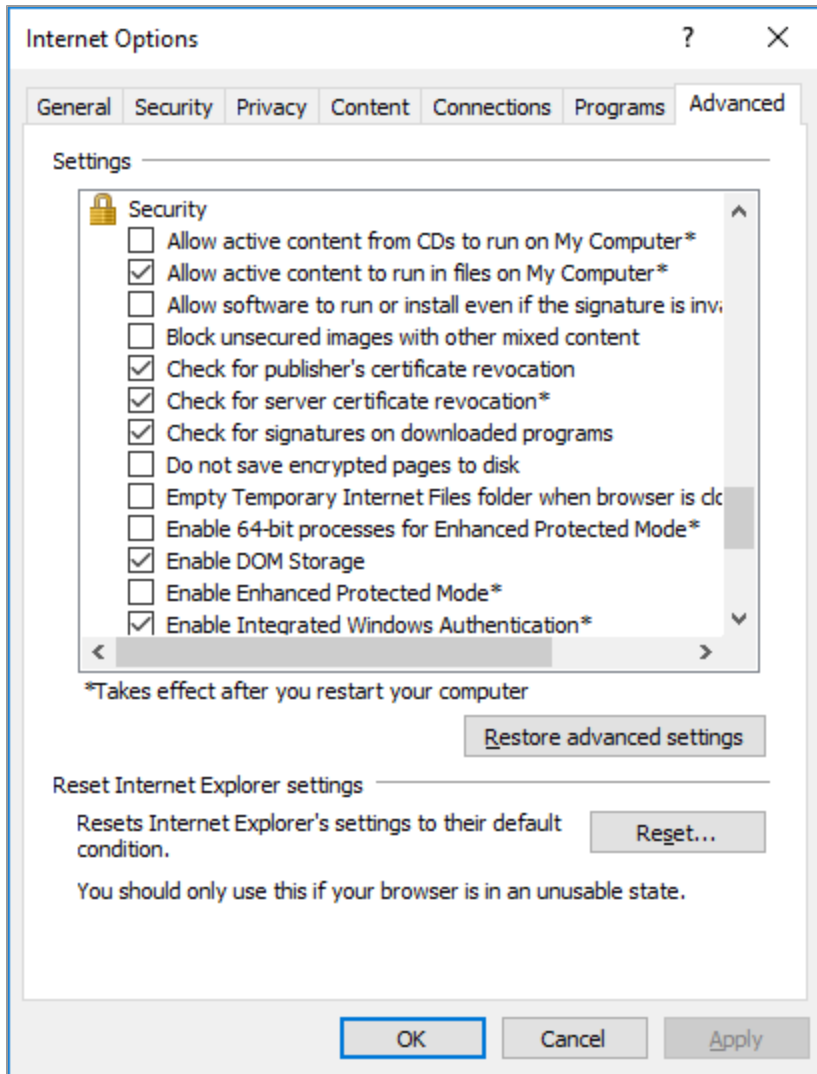
In some situations, you may encounter "Out of Memory" errors in the Gocator web interface. This issue can be resolved by checking two options in Internet Explorer.

To correct out of memory issues in Internet Explorer 11:

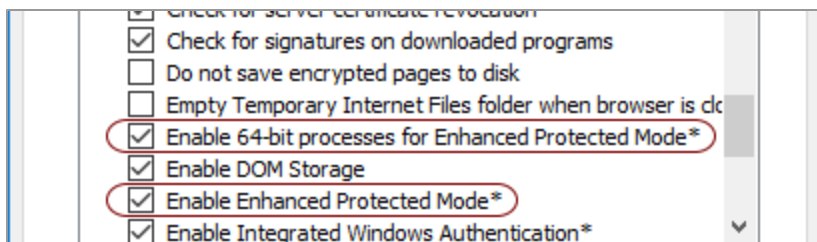
1. In upper right corner, click the settings icon (⚙️), and choose **Internet options**.



2. In the dialog that opens, click the **Advanced** tab, and scroll down to the **Security** section.



3. In the dialog, check *both* "Enable 64-bit processes for Enhanced Protected Mode" *and* "Enable Enhanced Protected Mode".

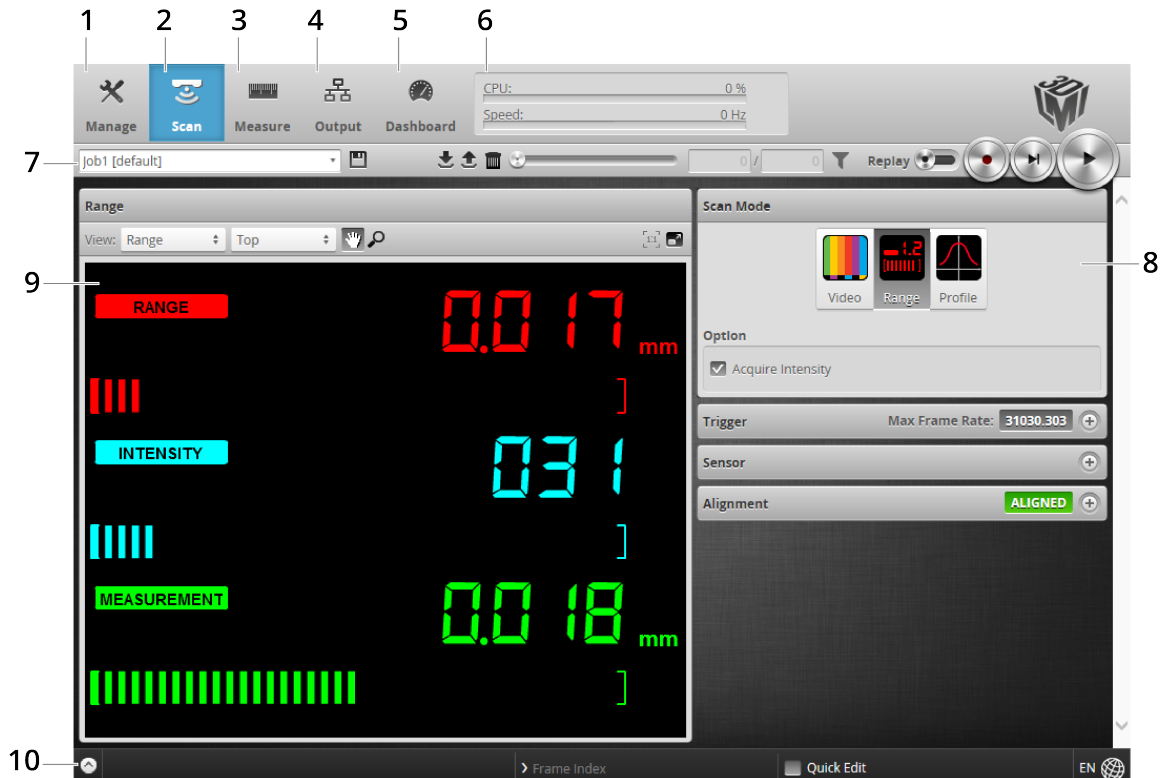


4. Click **OK** and then restart your computer for the changes to take effect.

User Interface Overview

Gocator sensors are configured by connecting to the IP address of a sensor with a web browser.

The Gocator web interface is shown below.

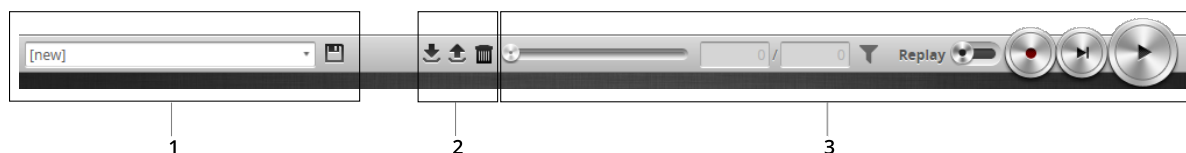


Element	Description
1	Manage page Contains settings for sensor system layout, network, motion and alignment, handling jobs, and sensor maintenance. See <i>Management and Maintenance</i> on page 67.
2	Scan page Contains settings for scan mode, trigger source, detailed sensor configuration, and performing alignment. See <i>Scan Setup and Alignment</i> on page 84.
3	Measure page Contains built-in measurement tools and their settings. See <i>Measurement</i> on page 120.
4	Output page Contains settings for configuring output protocols used to communicate measurements to external devices. See <i>Output</i> on page 189.
5	Dashboard page Provides monitoring of measurement statistics and sensor health. See <i>Dashboard</i> on page 201.
6	CPU Load and Speed Provides important sensor performance metrics. See <i>Metrics Area</i> on page 63.
7	Toolbar Controls sensor operation, manages jobs, and filters and replays recorded measurement data. See <i>Toolbar</i> on the next page.
8	Configuration area Provides controls to configure scan and measurement tool settings.
9	Data viewer Displays sensor data, tool setup controls, and measurements. See <i>Data Viewer</i> on page 113 for its use when the Scan page is active and on page 120 for its use when the Measure page is active.

Element	Description
10 Status bar	Displays log messages from the sensor (errors, warnings, and other information) and frame information , and lets you switch the interface language . For more information, see <i>Status Bar</i> on page 64.

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 and filter recording, and control recorded data.

Creating, 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 interface in 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

Setting Type	Behavior
	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 the 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 the 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 76 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 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**.

 Replay mode disables measurements.

2. (Optional) Configure recording filtering.
For more information on recording filtering, see *Recording Filtering* on the next page.
3. Click the **Record** button to enable recording.

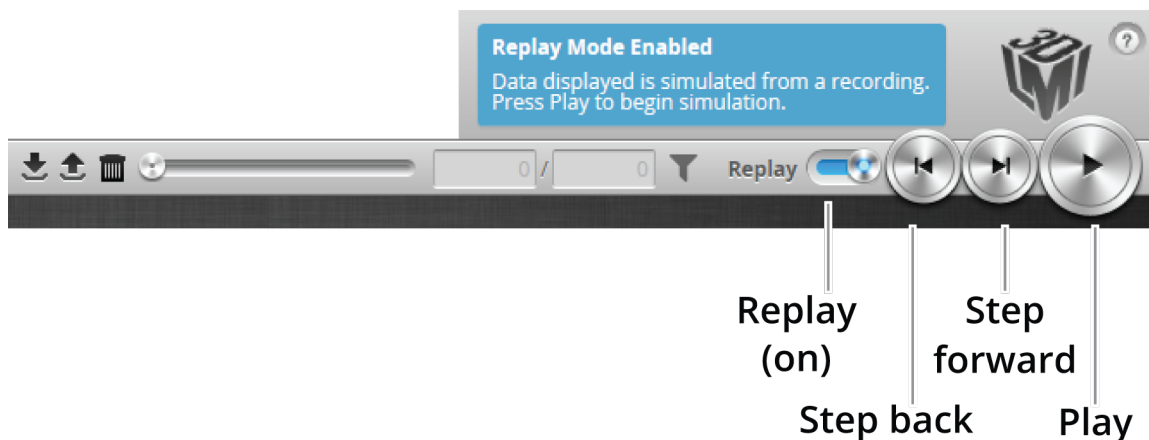


The center of the Record button turns red.

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).

4. 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.



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.
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:

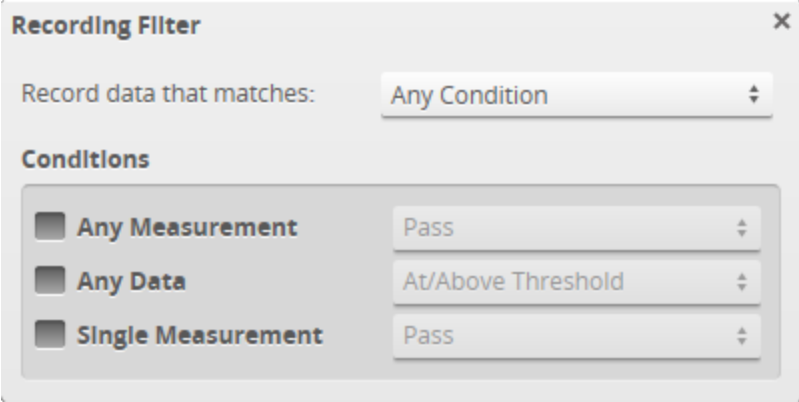
1. Toggle **Replay** mode on by setting the slider to the right in the **Toolbar**.
The slider's background turns blue and a Replay Mode Enabled message is displayed.
To change the mode, **Replay Protection** must be unchecked.
2. Go to the **Measure** page.
Modify settings for existing measurements, add new measurement tools, or delete measurement tools as desired. For information on adding and configuring measurements, see *Measurement* on page 120.
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 201.

To clear replay data:

1. Stop the sensor if it is running by clicking the **Stop** button.
2. Click the **Clear Replay Data** button .

Recording Filtering

Replay data is often used for troubleshooting. But replay data can contain thousands of frames, which makes finding a specific frame to troubleshoot difficult. Recording filtering lets you choose which frames Gocator records, based on one or more conditions, which makes it easier to find problems.



How Gocator treats conditions

Setting	Description
Any Condition	Gocator records a frame when any condition is true.
All Conditions	Gocator only records a frame if all conditions are true.

Conditions

Setting	Description
Any Measurement	Gocator records a frame when <i>any</i> measurement is in the state you select. The following states are supported: • pass • fail or invalid • fail and valid • valid • invalid
Single Measurement	Gocator records a frame if the measurement with the ID you specify in ID is in the state you select. This setting supports the same states as the Any Measurement setting (see above).
Any Data	At/Above Threshold: Gocator records a frame if the number of valid points in the frame is above the value you specify in Range Count Threshold . Below Threshold: Gocator records a frame if the number of valid points is below the threshold you specify.

To set recording filtering:

1. Make sure recording is enabled by clicking the Record button.



2. Click the Recording Filtering button .
3. In the Recording Filtering dialog, choose how Gocator treats conditions:
For information on the available settings, see *How Gocator treats conditions* above.
4. Configure the conditions that will cause Gocator to record a frame:
For information on the available settings, see *Conditions* above.

- Click the "x" button or outside of the Recording Filtering dialog to close the dialog.

The recording filter icon turns green to show that recording filters have been set.

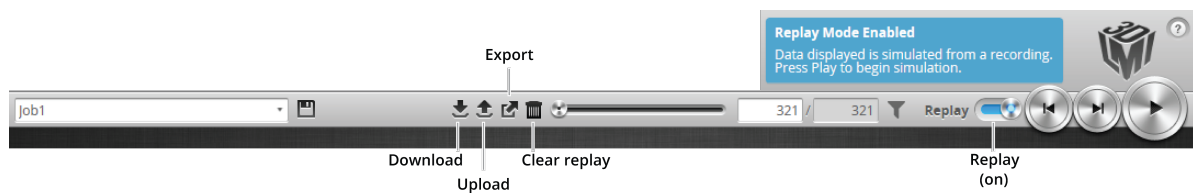
When you run the sensor, Gocator only records the frames that satisfy the conditions you have set.

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.



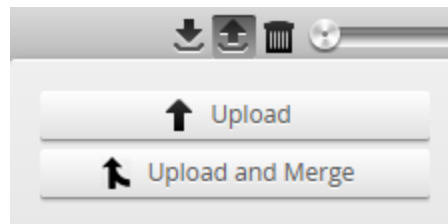
Replay data is not loaded or saved when you load or save jobs.

To download replay data:

- Click the Download button .
- In the **File Download** dialog, click **Save**.
- In the **Save As...** dialog, choose a location, optionally change the name (keeping the .rec extension), and click **Save**.

To upload replay data:

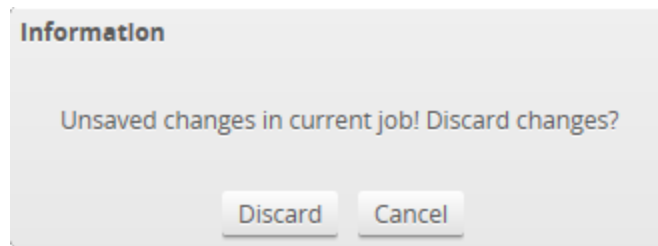
- Click the Upload button .
- The Upload menu appears.



- In the Upload menu, choose one of the following:
 - Upload:** Unloads the current job and creates a new unsaved and untitled job from the content of the replay data file.
 - Upload and merge:** Uploads the replay data and merges the data's associated job with the current job. Specifically, the settings on the **Scan** page are overwritten, but all other settings of the current job are preserved, including any measurements.

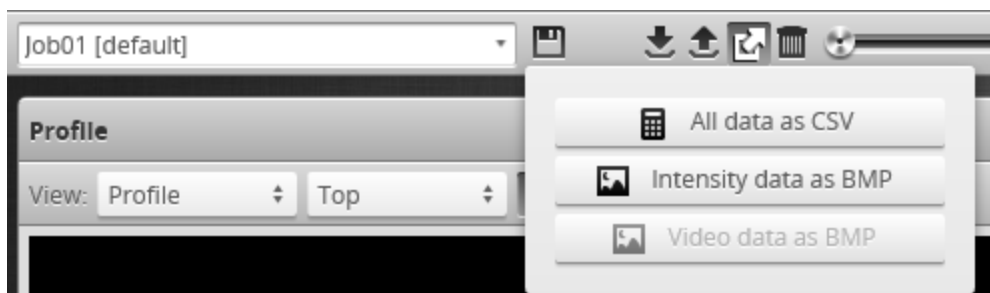
If you have unsaved changes in the current job, the firmware asks whether you want to discard the

changes.



3. Do one of the following:
 - Click **Discard** to discard any unsaved changes.
 - Click **Cancel** to return to the main window to save your changes.
4. If you clicked **Discard**, navigate to the replay data to upload from the client computer and click **OK**. The replay data is loaded, and a new unsaved, untitled job is created.

Replay data can be exported using the CSV format. If you have enabled **Acquire Intensity** in the **Scan Mode** panel on the **Scan** page, the exported CSV file includes intensity data.



To export replay data in the CSV format:

1. In the **Scan Mode** panel, switch to Range or Profile.
2. Click the Export button  and select **All Data as CSV**.

In Profile mode, all data in the record buffer is exported. 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 57.
3. (Optional) Convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 402.



The decision values in the exported data depend on the *current* state of the job, not the state during recording. For example, if you record data when a measurement returns a *pass* decision, change the measurement's settings so that a *fail* decision is returned, and then export to CSV, you will see a *fail* decision in the exported data.

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:

- 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 57.



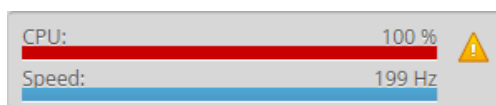
To export video data to a BMP file:

- In the **Scan Mode** panel, switch to Video mode.
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 57.
- Click the Export button  and select **Video data as BMP**.

Metrics Area

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

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

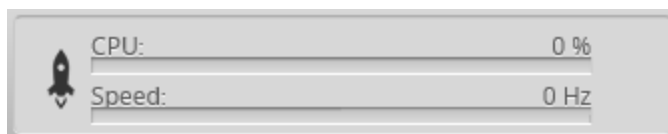


CPU at 100%

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.

Open the log for details on the warning. For more information on logs, see *Log* on the next page.

When a sensor is [accelerated](#) a "rocket" icon appears in the metrics area.



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 the active area and other settings. Depending on the selected operation mode (page 85), the data viewer can display video images, ranges, or profiles. For details, see *Data Viewer* on page 113.

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 120.

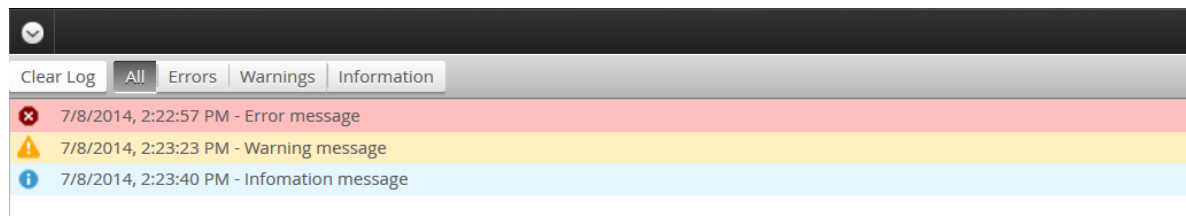
Status Bar

The status bar lets you do the following:

- See sensor messages in the [log](#).
- See [frame information](#).
- Change the [interface language](#).
- Switch to [Quick Edit mode](#).

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.



A number indicates the number of unread messages:



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.

Frame Information

The area to the right of the status bar displays useful frame information, both when the sensor is running and when viewing recorded data.



This information is especially useful when you have enabled [recording filtering](#). If you look at a recording playback, when you have enabled recording filtering, some frames can be excluded, resulting in variable "gaps" in the data.

The following information is available:

Frame Index: Displays the index in the data buffer of the current frame. The value resets to 0 when the sensor is restarted or when recording is enabled.

Master Time: Displays the recording time of the current frame, with respect to when the sensor was started.

Encoder Index: Displays the encoder index of the current frame.

Timestamp: Displays the timestamp the current frame, in microseconds from when the sensor was started.

To switch between types of frame information:

- Click the frame information area to switch to the next available type of information.

Quick Edit Mode

When working with a very large number of [measurement tools](#) (for example, a few dozen) or a very complex user-created [GDK tool](#), you can switch to a "Quick Edit" mode to make configuration faster.



When this mode is enabled, the data viewer and measurement results are not refreshed after each setting change. Also, when Quick Edit is enabled, in Replay mode, [stepping through frames](#) or playing back scan data does not change the displayed frame.

 When a sensor is running, Quick Edit mode is ignored: all changes to settings are reflected immediately in the data viewer.

Interface Language

The language button on the right side of the status bar lets you change the language of the Gocator interface.



To use the log:

1. Click the language button at the bottom of the web interface.



2. Choose a language from the list.



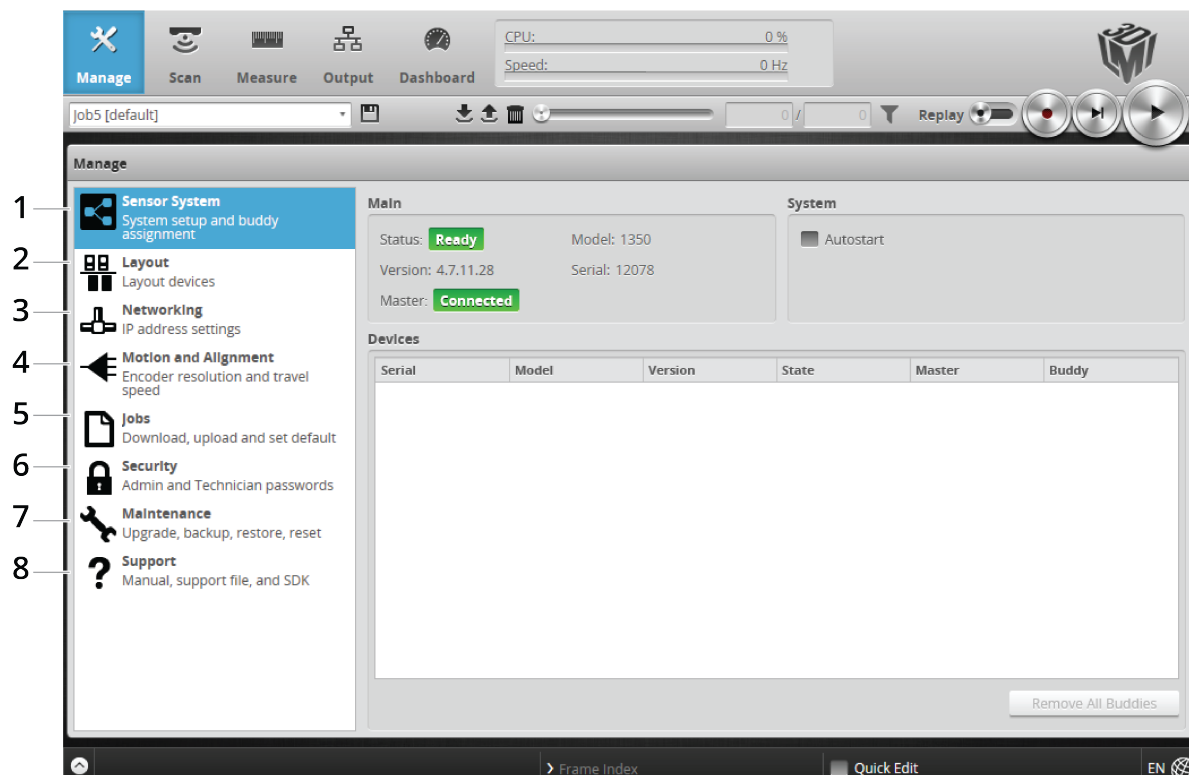
The Gocator interface reloads on the page you were working in, displaying the page using the language you chose. The sensor state is preserved.

Management and Maintenance

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

Manage Page Overview

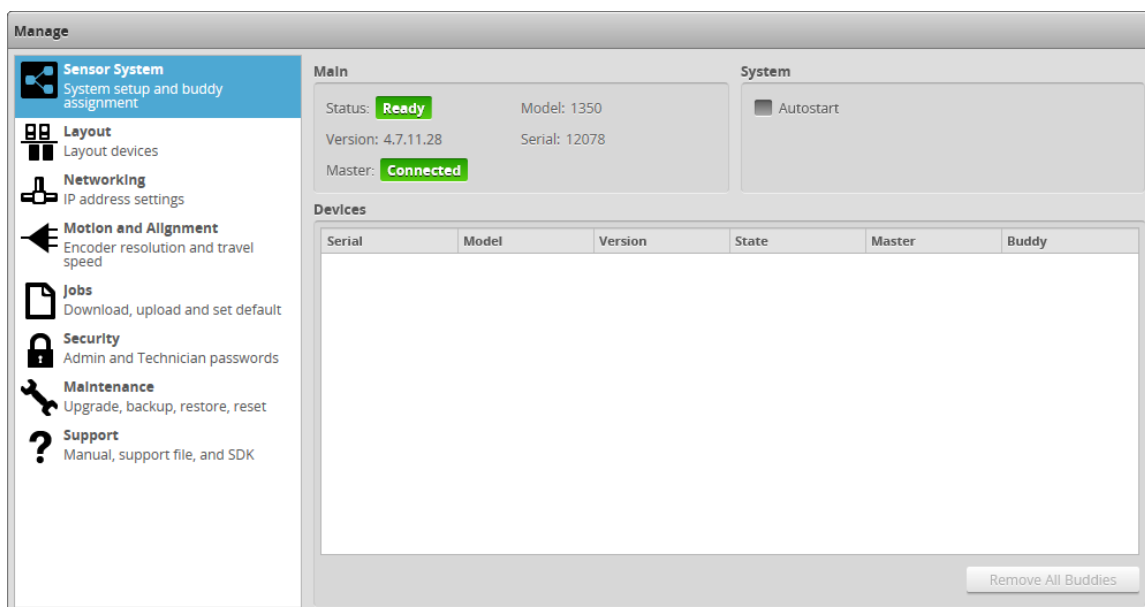
Gocator's system and maintenance tasks are performed on the **Manage** page.



Element	Description
1 Sensor System	Contains sensor information, buddy assignment, and the autostart setting. See <i>Sensor System</i> on the next page.
2 Layout	Contains settings for configuring dual-sensor system layouts.
3 Networking	Contains settings for configuring the network. See <i>Networking</i> on page 73.
4 Motion and Alignment	Contains settings to configure the encoder. See <i>Motion and Alignment</i> on page 74.
5 Jobs	Lets you manage jobs stored on the sensor. See <i>Jobs</i> on page 76.
6 Security	Lets you change passwords. See <i>Security</i> on page 77.
7 Maintenance	Lets you upgrade firmware, create/restore backups, and reset sensors. See <i>Maintenance</i> on page 78.
8 Support	Lets you open an HTML version or download a PDF version of the manual, download the SDK, or save a support file. Also provides device information. See <i>Support</i> on page 81

Sensor System

The following sections describe the **Sensor System** category on the **Manage** page. This category provides sensor information and the autostart setting. It also lets you choose which sensors to add to a dual-sensor system.



Dual- and Multi-sensor Systems

Gocator supports dual-sensor systems. In these systems, data from each sensor is combined into a single range or profile. Any [measurements](#) you configure work on the combined data.

Gocator lets you easily and quickly set up dual-sensor systems from the web interface. Setting up these systems involves two steps:

1. Assigning an additional sensor, called the *Buddy* sensor, to the *Main* sensor. For more information, see *Buddy Assignment* below.
2. Choosing the layout of the dual-sensor system. For more information, see *Layout* on page 70.

Buddy Assignment

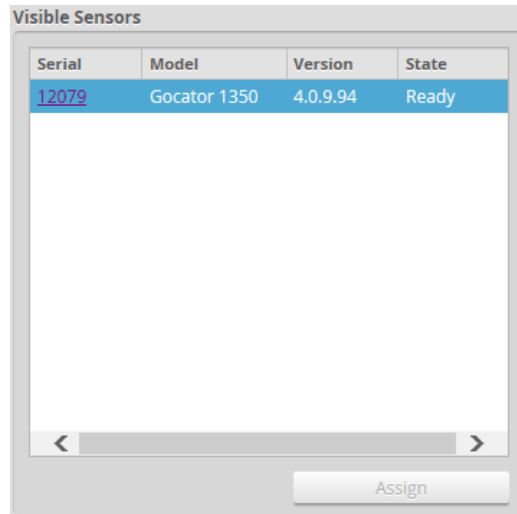
In a dual-sensor system, the *Main* sensor controls a second sensor, called the *Buddy* sensor, after the Buddy sensor is assigned to the Main sensor. You configure both sensors through the Main sensor's interface.



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



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



 A sensor can only be assigned as a Buddy if its firmware and model number match the firmware and model number of the Main sensor.

To assign a Buddy sensor:

1. Go to the **Manage** page and click on the **Sensor System** category.
2. In the **Visible Sensors** list, click the "plus" icon next to the sensor you want to add as a Buddy.
The sensor you added to the system appears in a **Buddies** list.

Devices					
Serial	Model	Version	State	Master	Buddy
Buddies					
40276	Gocator 2420	4.6.5.53	Connected	Connected	
Visible Sensors					
40166	Gocator 2420	4.6.5.53	Connectable	-	
40279	Gocator 2420	4.6.5.53	Connectable	-	
40278	Gocator 2420	4.6.5.53	Connectable	-	

3. Repeat the previous step to add more sensors to the system.

To remove a Buddy, click the "minus" icon next to the sensor you want to remove. To remove all Buddies, click **Remove All Buddies**.

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.



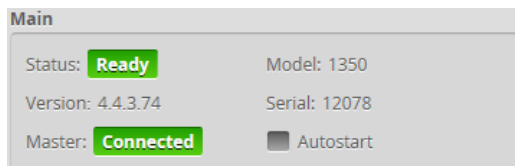
To enable/disable overheat temperature protection:

1. Check/uncheck the **Over Temperature Protection** option.

2. Save the job file.

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.



Main

Status: **Ready** Model: 1350

Version: 4.4.3.74 Serial: 12078

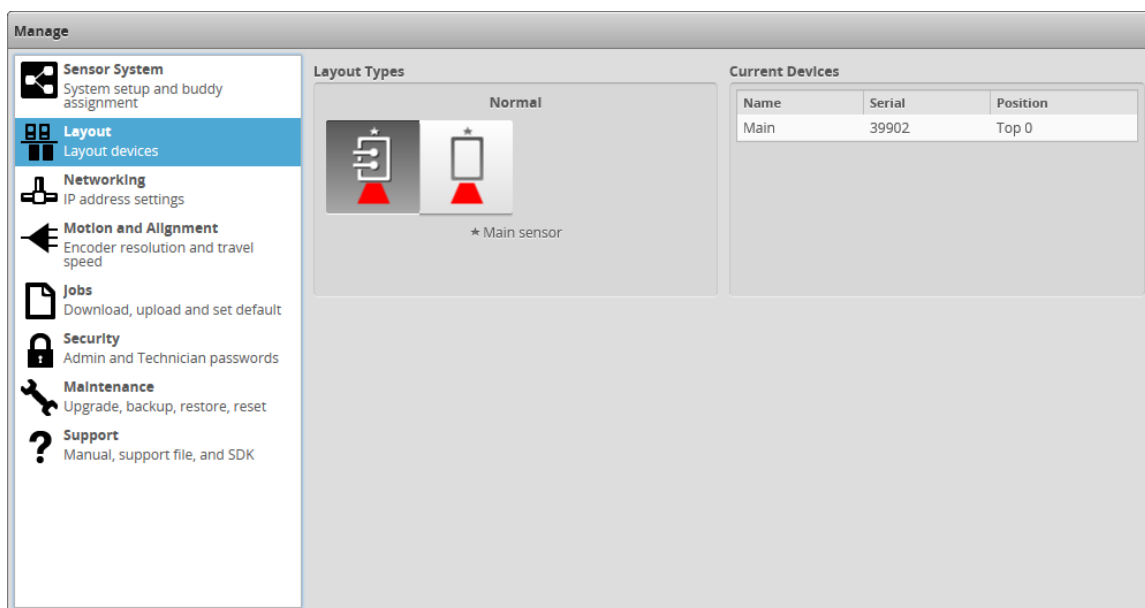
Master: **Connected** ☐ Autostart

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.

Layout

The following sections describe the **Layout** category on the **Manage** page. This category lets you configure dual-sensor systems.



Manage

Sensor System
System setup and buddy assignment

Layout
Layout devices

Networking
IP address settings

Motion and Alignment
Encoder resolution and travel speed

Jobs
Download, upload and set default

Security
Admin and Technician passwords

Maintenance
Upgrade, backup, restore, reset

Support
Manual, support file, and SDK

Layout Types

Normal

☒ ☐

* Main sensor

Current Devices

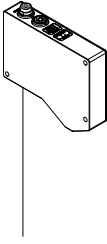
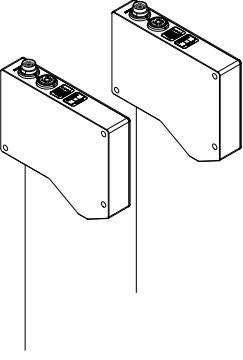
Name	Serial	Position
Main	39902	Top 0

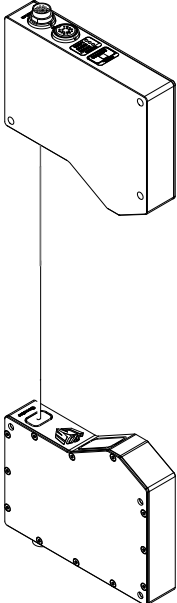
Mounting orientations must be specified for a dual- or multi-sensor system. This information allows the alignment procedure to determine the correct system-wide coordinates for laser ranging and measurements. For more information on sensor and system coordinates, see *Coordinate Systems* on page 49.



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

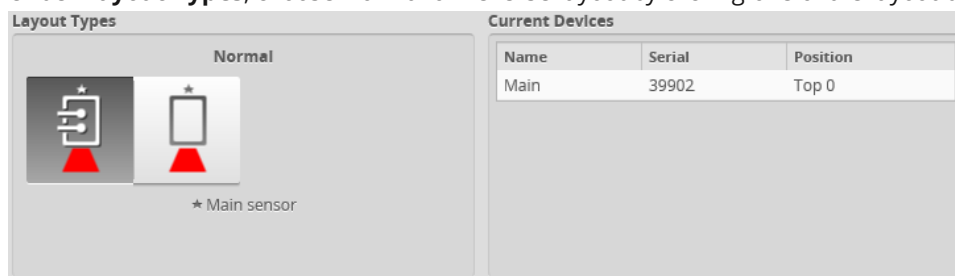
Supported Layouts

Layout Type	Example
	Normal The sensor operates as an isolated device. 
	Reverse The sensor operates as an isolated device, but in a reverse orientation. You can use this layout to change the handedness of the data.
	Wide Sensors are mounted in Left (Main) and Right (Buddy) positions. This allows for measuring the height of the object at multiple points. 
	Reverse Sensors are mounted in a left-right layout as with the Wide layout, but the Buddy sensor is mounted such that it is rotated 180 degrees around the Z axis to prevent occlusion along the Y axis. Sensors should be shifted along the Y axis so that the laser lines align.

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

To specify a standalone layout:

1. Go to the **Manage** page and click on the **Layout** category.
2. Under **Layout Types**, choose Normal or Reverse layout by clicking one of the layout buttons.



See the table above for information on layouts.

 Before you can select a dual-sensor layout, you must assign a second sensor as the Buddy sensor. For more information, see *Dual- and Multi-sensor Systems* on page 68.

To specify a dual-sensor layout:

1. Go to the **Manage** page and click on the **Layout** category.
2. Under **Layout Types**, choose a layout by clicking one of the layout buttons.

Layout Types

Wide

★ Main sensor

☐ Device Exposure Multiplexing

Current Devices

Name	Serial	Position
Main	22282	Top 0
Buddy 0	22285	Top 1

See the table above for information on layouts.

Before you can select a multi-sensor layout, you must assign two or more additional sensors as Buddy sensors. For more information, see *Dual- and Multi-sensor Systems* on page 68.

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.

Manage

Sensor System
System setup and buddy assignment

Layout
Layout devices

Networking
IP address settings

Motion and Alignment
Encoder resolution and travel speed

Jobs
Download, upload and set default

Security
Admin and Technician passwords

Maintenance
Upgrade, backup, restore, reset

Support
Manual, support file, and SDK

Networking

Type:

Manual

IP:

192.168.1.10

Subnet Mask:

255.255.255.0

Gateway:

0.0.0.0

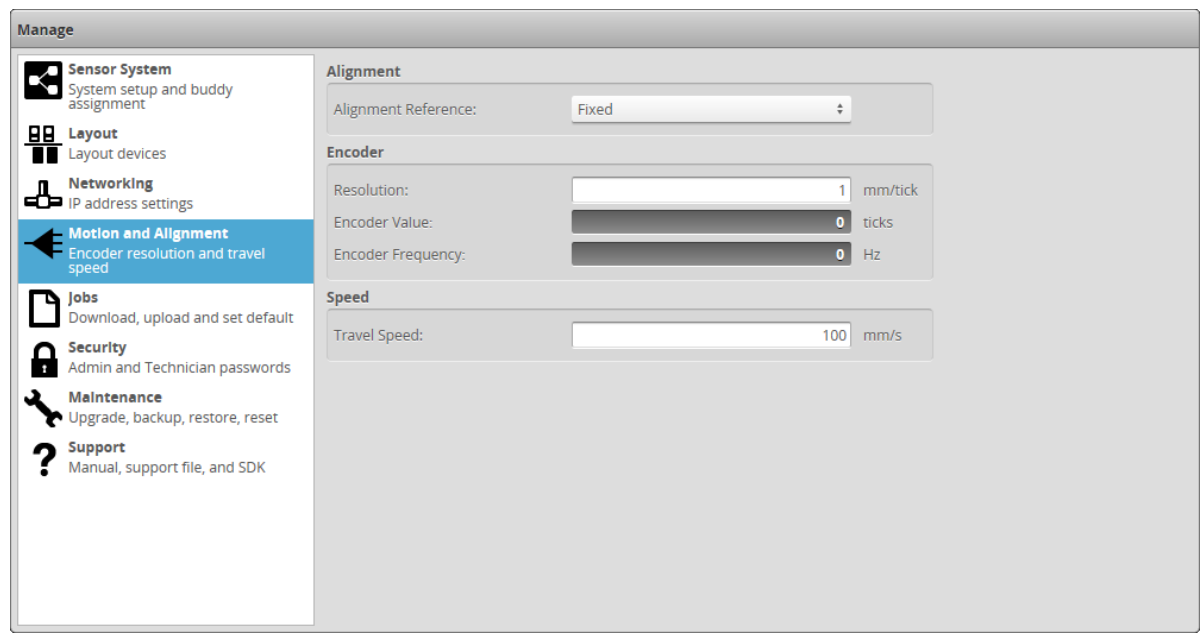
Save

To configure the network settings:

- Go to the **Manage** page.
- 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 by selecting the appropriate option in the **Type** drop-down.
- Click on the **Save** button.
You will be prompted to confirm your selection.

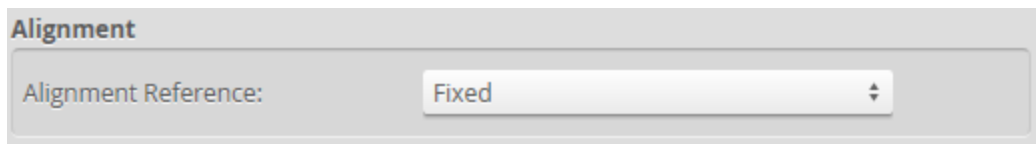
Motion and Alignment

The **Motion and Alignment** category on the **Manage** page lets you configure alignment reference, encoder resolution, and travel speed, and confirm that encoder signals are being received by the sensor.



Alignment Reference

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



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

To configure alignment reference:

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

Encoder Resolution

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



The image shows a configuration panel titled "Encoder". It contains three input fields: "Resolution:" with a value of "1" and unit "mm/tick", "Encoder Value:" with a value of "0" and unit "ticks", and "Encoder Frequency:" with a value of "0" and unit "Hz".

Encoder resolution is expressed in millimeters per tick, where one tick corresponds to *one* of the four encoder quadrature signals (A+ / A- / B+ / B-).



Encoders are normally specified in *pulses* per revolution, where each pulse is made up of the four quadrature *signals* (A+ / A- / B+ / B-). Because Gocator reads each of the four quadrature signals, you should choose an encoder accordingly, given the resolution required for your application.

To configure encoder resolution:

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

Encoder Value and Frequency

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

Travel Speed

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



The image shows a configuration panel titled "Speed". It contains one input field: "Travel Speed:" with a value of "100" and unit "mm/s".

Travel speed is expressed in millimeters per second.

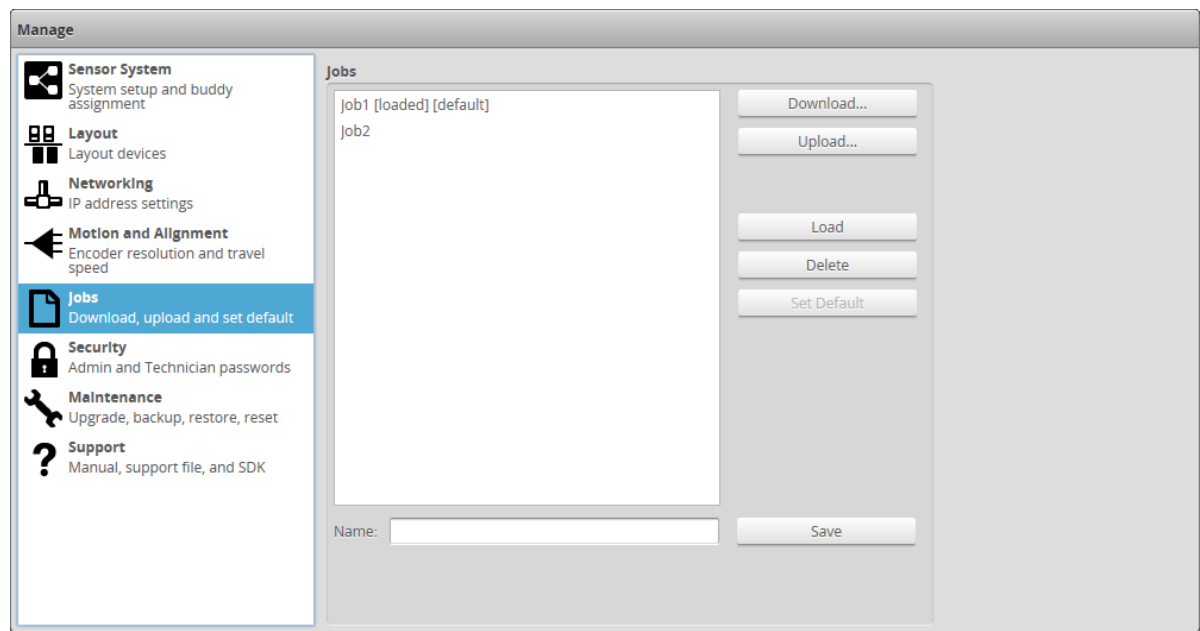
To manually configure travel speed:

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

Travel speed can also be set automatically by performing an alignment with **Type** set to **Moving** (see *Aligning Sensors* on page 100).

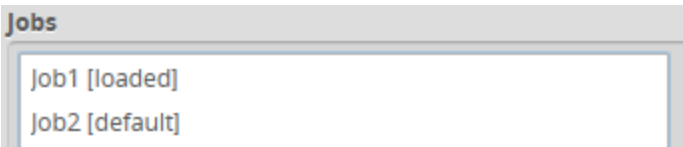
Jobs

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

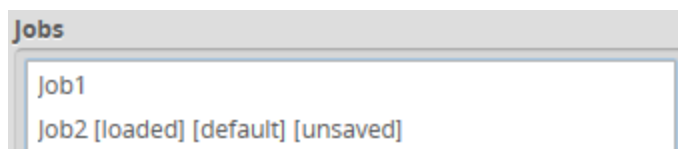


Element	Description
Name field	Used to provide a job name when saving files.
Jobs list	Displays the jobs that are currently saved in the sensor's flash storage.
Save button	Saves current settings to the job using the name in the Name field.
Load button	Loads the job that is selected in the job list. 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 the sensor is restarted.

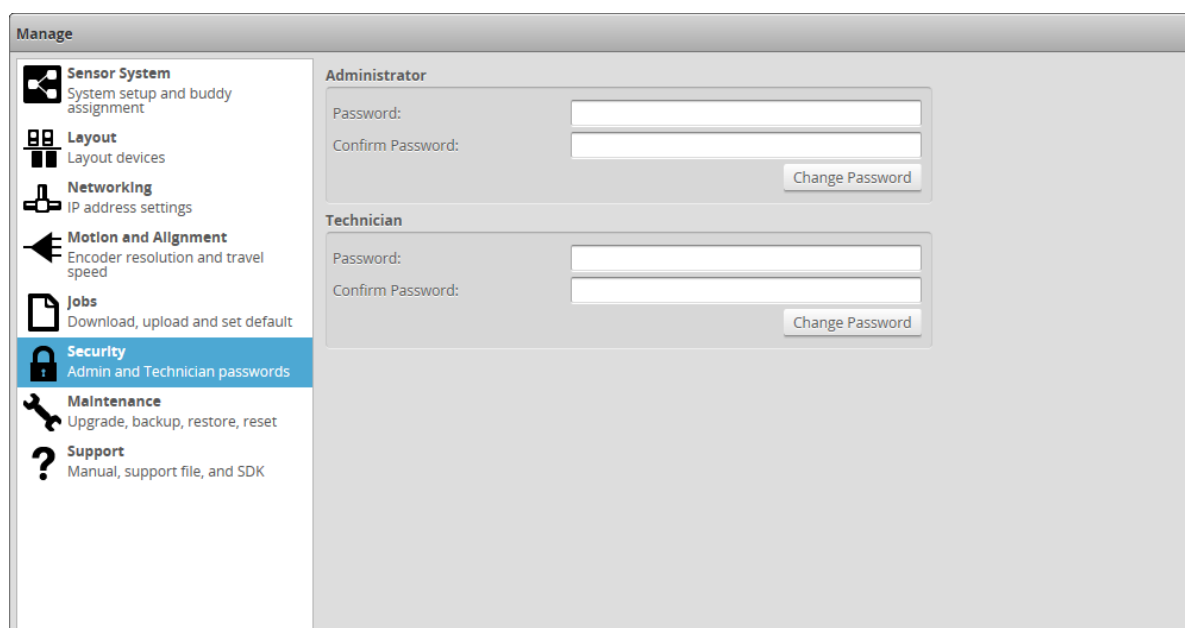
To download, load, or delete a job, or to set one as a default, or clear a default:

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

You can prevent unauthorized access to a Gocator sensor by setting passwords. Each sensor has two accounts: Administrator and Technician.

By default, no passwords are set. When you start a sensor, you are prompted for a password only if a password has been set.



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.

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:

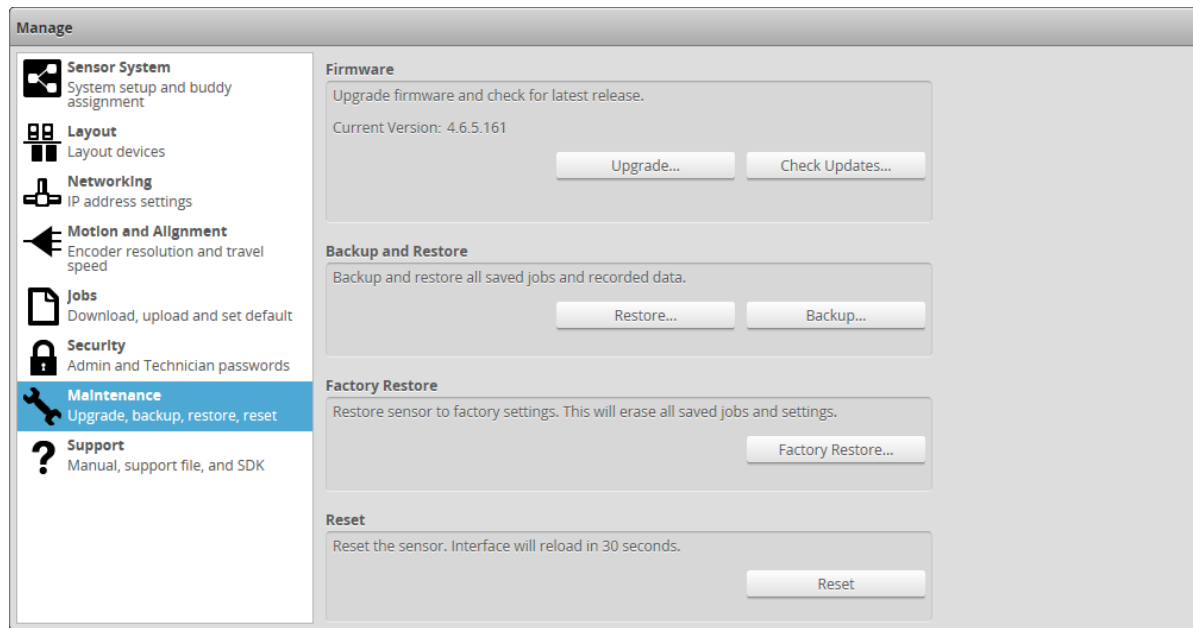
1. Go to the **Manage** page and click on the **Security** category.
2. In the **Technician** section, enter the Technician account password and password confirmation.
3. Click **Change Password**.
The new password will be required the next time that a technician logs in to the sensor.

If the administrator or technician password is lost, the sensor can be recovered using a special software tool. See *Sensor Discovery Tool* on page 401 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.



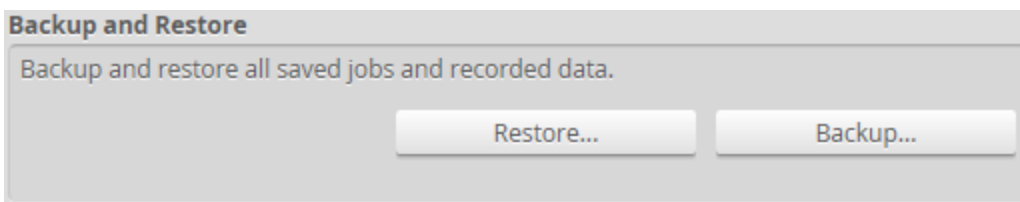
Sensor Backups and Factory Reset

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.

Factory Restore

Restore sensor to factory settings. This will erase all saved jobs and settings.

Factory Restore...

To restore from a backup:

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

To restore a sensor to its factory default settings:

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

Firmware Upgrade

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



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

Firmware

Upgrade firmware and check for latest release.

Current Version: 4.0.9.84

Upgrade...

Check Updates...

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. Locate the firmware file in the **File** dialog and then click open.
4. Wait for the upgrade to complete.

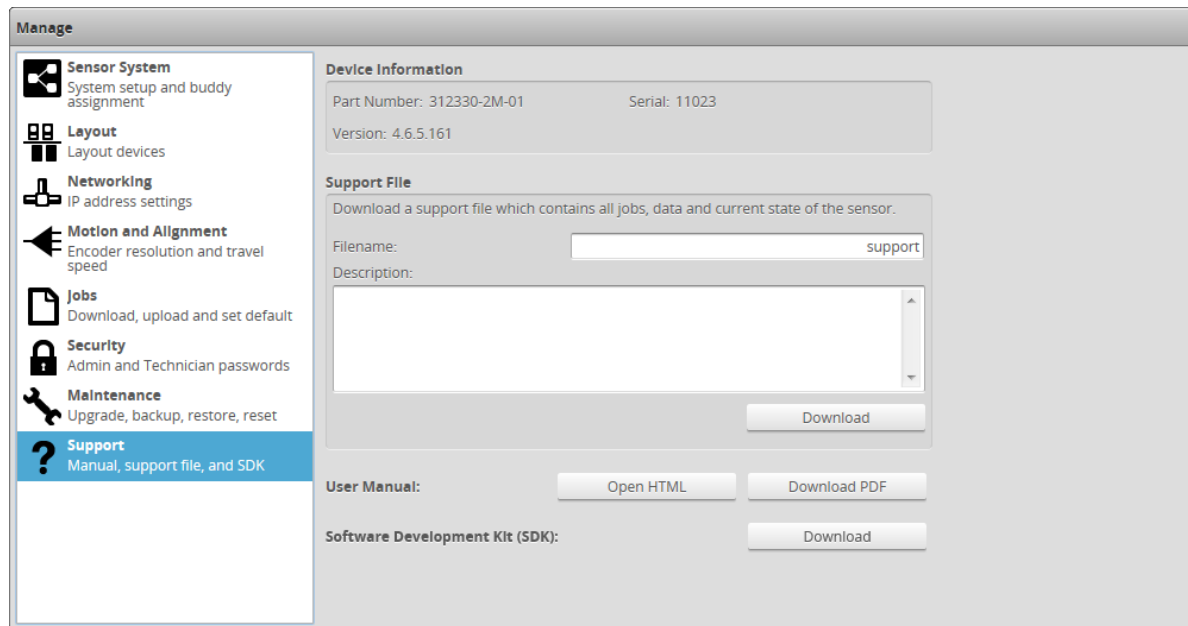
After the firmware upgrade is complete, the sensor will self-reset. If a buddy has been assigned, it will be upgraded and reset automatically.

Support

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

- open an HTML version or download a PDF version of the manual
- download the SDK
- save a support file
- get device information

The screenshot shows the 'Manage' page of the Gocator web interface. On the left is a sidebar with a tree view of categories: Sensor System, Layout, Networking, Motion and Alignment, Jobs, Security, Maintenance, and Support. The 'Support' category is highlighted with a blue background and a question mark icon. The main content area is divided into several sections. The 'Device Information' section shows 'Part Number: 311320-2M-01', 'Serial: 13434', and 'Version: 4.6.5.161'. Below this is the 'Support File' section, which includes a description: 'Download a support file which contains all jobs, data and current state of the sensor.' It has a 'Filename:' field with 'support' entered and a 'Description:' text area. A 'Download' button is at the bottom right of this section. At the bottom of the main area, there are two rows of buttons. The first row is labeled 'User Manual:' and has 'Open HTML' and 'Download PDF' buttons. The second row is labeled 'Software Development Kit (SDK):' and has a 'Download' button.



Support Files

You can download a support file from a sensor and save it on your computer. You can then use the support file to create a scenario in the Gocator emulator (for more information on the emulator, see *Gocator Emulator* on page 205). LMI's support staff may also request a support file to help in troubleshooting.

To download a support file:

1. Go to the **Manage** page and click on the **Support** category.
2. In **Filename**, type the name you want to use for the support file.
When you create a scenario from a support file in the emulator, the filename you provide here is displayed in the emulator's scenario list.
Support files end with the .gs extension, but you do not need to type the extension in **Filename**.
3. (Optional) In **Description**, type a description of the support file.

When you create a scenario from a support file in the emulator, the description is displayed below the emulator's scenario list.

4. Click **Download**, and then when prompted, click **Save**.

 Downloading a support file stops the sensor.

Manual Access

You can access the Gocator manuals from within the Web interface.

User Manual:

Open HTML

Download PDF

 You may need to configure your browser to allow pop-ups to open or download the manual.

To access the manuals:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **User Manual**, click one of the following:
 - **Open HTML**: Opens the HTML version of the manual in your default browser.
 - **Download PDF**: Downloads the PDF version of the manual to the client computer.

Software Development Kit

You can download the Gocator SDK from within the Web interface.

Software Development Kit (SDK):

Download

To download the SDK:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **Software Development Kit (SDK)**, click **Download**
3. Choose the location for the SDK on the client computer.

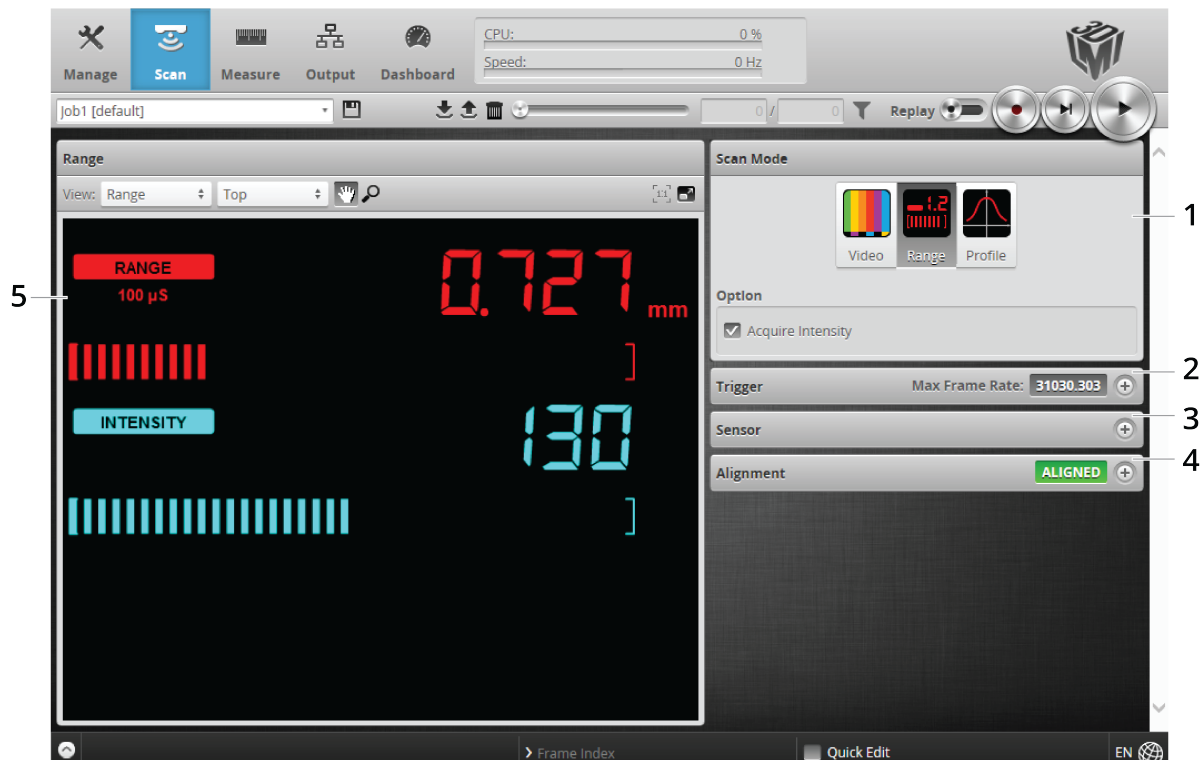
 For more information on the SDK, see *Development Kits* on page 377.

Scan Setup and Alignment

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

Scan Page Overview

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



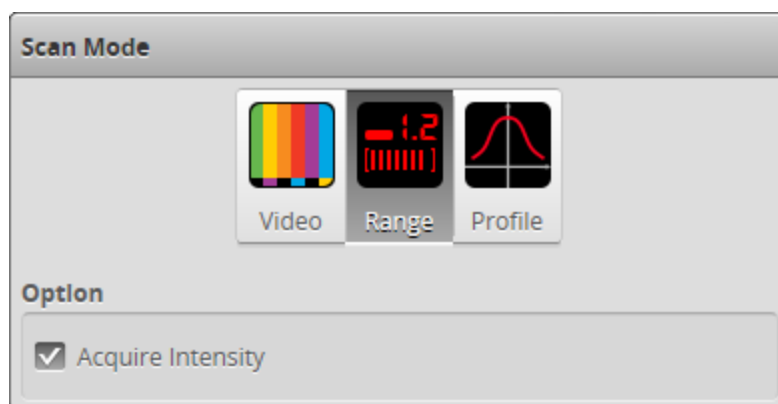
Element	Description
1	Scan Mode panel Contains settings for the current scan mode and other options. See <i>Scan Modes</i> on the next page.
2	Trigger panel Contains trigger source and trigger-related settings. See <i>Triggers</i> on the next page.
3	Sensor panel Contains settings for an individual sensor, such as active area or exposure. See <i>Sensor</i> on page 92.
4	Alignment panel Used to perform alignment. See <i>Alignment</i> on page 100.
5	Data Viewer Displays sensor data and adjusts regions of interest. Depending on the current operation mode, the data viewer can display video images or scan data. See <i>Data Viewer</i> on page 113.

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 85)
Ensure that camera exposure is appropriate for scan data acquisition.	Exposure (page 95)
Find the right balance between data quality, speed, and CPU utilization.	Active Area (page 92) Exposure (page 95) Job File Structure (page 227)
Align scan data to a common reference and so that values can be correctly scaled along the different axes.	Aligning Sensors (page 100)

Scan Modes

The Gocator web interface supports a video mode and one or more data acquisition modes. The scan mode can be selected in the **Scan Mode** panel.



Mode and Option	Description
Video	Outputs video images from the Gocator. This mode is useful for configuring exposure time and troubleshooting stray light or ambient light problems.
Range	Outputs ranges and performs measurements. Video images are processed internally to produce laser ranges and measurements.
Profile	Outputs profiles and performs profile measurements. The sensor uses various methods to generate a profile (see <i>Profile Generation</i> on page 108). Part detection can be enabled on a profile to identify discrete parts (see <i>Part Detection</i> on page 111). Video images are processed internally to produce laser profiles and cross-sectional measurements.
Acquire Intensity	When this option is enabled, an intensity value will be produced for each data point.

Triggers

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

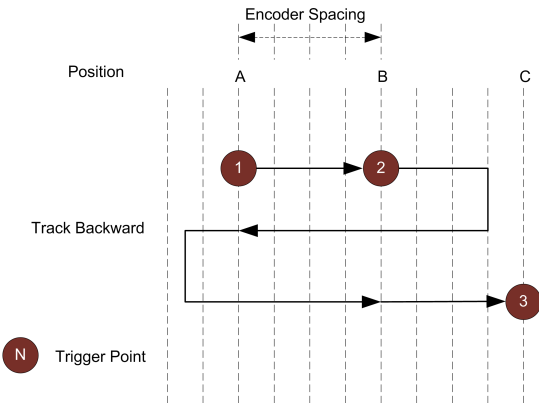
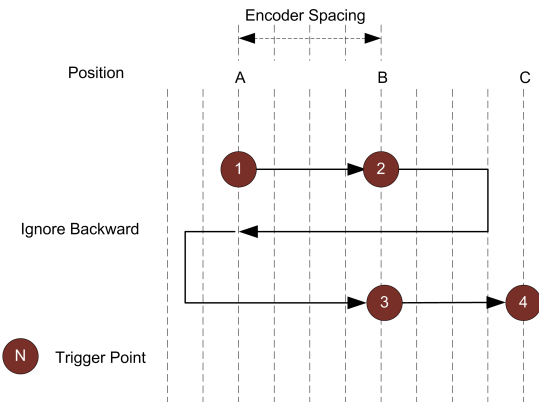
When a trigger is processed, the laser is strobed and the camera exposes to produce an image. The resulting image is processed inside the sensor to yield a laser range (distance information), which can then be used for measurement.

The sensor can be triggered by one of the sources described in the table below.

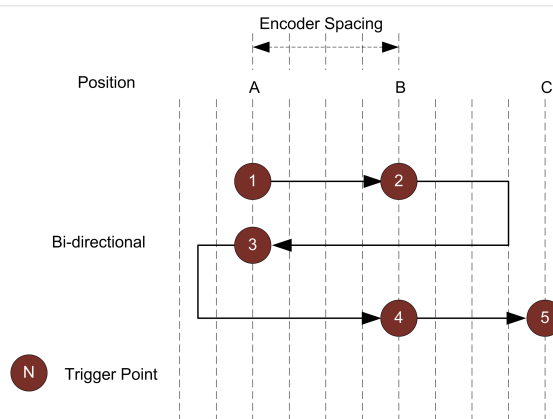
If the sensor is connected to a Master 400 or higher, encoder and digital (external) input signals over the IO cordset are *ignored*. The sensor instead receives these signals from the Master; for encoder and digital input pinouts on Masters, see the section corresponding to your Master in *Master Network Controllers* on page 443.

If the sensor is connected to a [Master 100](#) (or no Master is used), the sensor receives signals over the IO cordset. For information on connecting encoder and digital input signals to a sensor in these cases, see *Encoder Input* on page 440 and *Digital Input* on page 439*Encoder Input* on page 440

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.

Trigger Source	Description
Encoder	An encoder can be connected to provide triggers in response to motion. Three encoder triggering behaviors are supported. These behaviors are set using the Behavior setting. Track Backward A scan is triggered when the target object moves forward. If the target object moves backward, it must move forward by at least the distance that the target travelled backward (this distance backward is "tracked"), plus one encoder spacing, to trigger the next scan.  Ignore Backward A scan is triggered only when the target object moves forward. If the target object moves backward, it must move forward by at least the distance of one encoder spacing to trigger the next scan.  Bi-directional A scan is triggered when the target object moves forward or backward.

Trigger Source	Description
----------------	-------------



When triggers are received at a frequency higher than the maximum frame rate, some triggers may not be accepted. The **Trigger Drops Indicator** in the **Dashboard** can be used to check for this condition.

The external input can be used to enable or disable the encoder triggers.

For information on the maximum encoder rate, see *Maximum Encoder Rate* on page 92.

To verify that the sensor is receiving encoder signals, check whether **Encoder Value** is changing in the [Motion and Alignment](#) category on the **Manage** page, or in the [dashboard](#).

External Input	A digital input can provide triggers in response to external events (e.g., photocell). The external input triggers on the rising edge of the signal.
----------------	--

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.

For information on the maximum input trigger rate, see *Maximum Input Trigger Rate* on page 92.

Software	A network command can be used to send a software trigger. See <i>Protocols</i> on page 291 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.

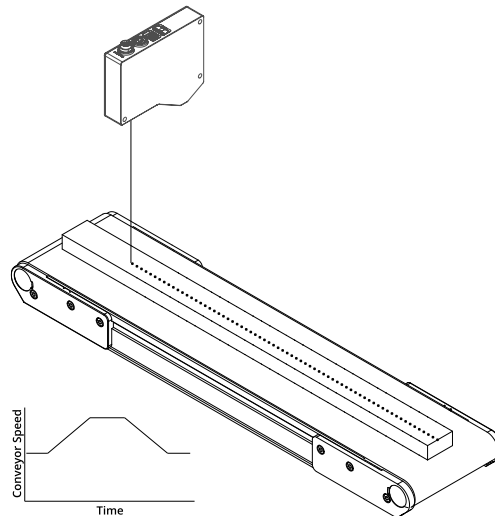
For examples of typical real-world scenarios, see *Trigger Examples* on the next page. For information on the settings used with each trigger source, see *Trigger Settings* on page 90

Trigger Examples

Example: Encoder + Conveyor

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

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

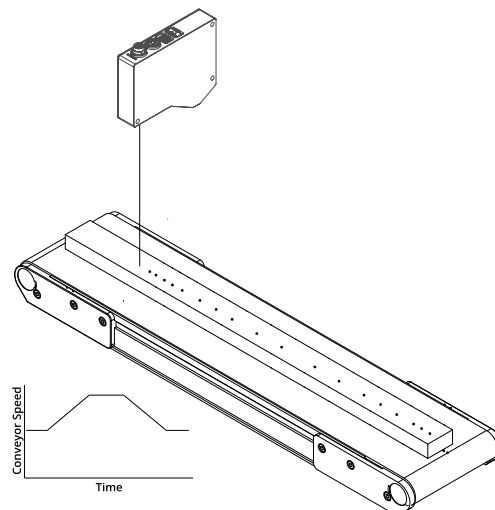


Example: Time + Conveyor

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

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

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

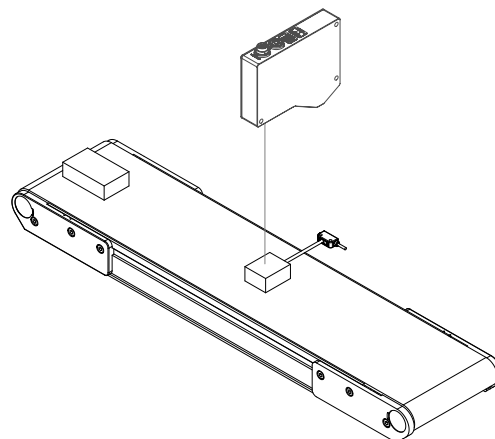


Example: External Input + Conveyor

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

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

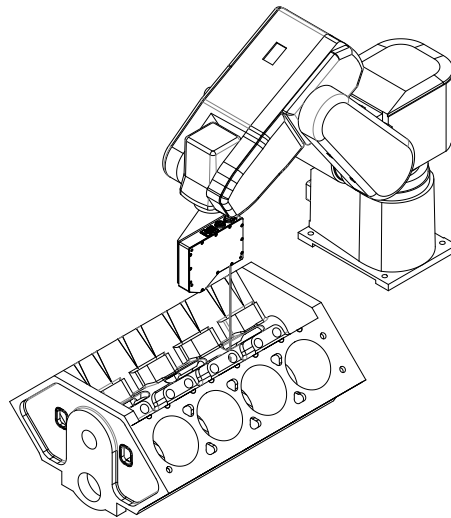
An external input can also be used to gate the trigger signals when time or encoder triggering is used. For example, a photocell could generate a series of trigger pulses as long as there is a target in position.



Example: Software Trigger + Robot Arm

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

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



Trigger Settings

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

Trigger Max Frame Rate: 31030.303

Source: Time

Frame Rate: Max Speed Hz

☐ Gate on External Input

Trigger Max Frame Rate: 31030.303

Source: Encoder

Spacing: 1 mm

Behavior: Bi-Directional

☐ Gate on External Input

Trigger Max Frame Rate: 31030.303

Source: External Input

Units: μ s (Time)

Trigger Delay: 0 μ s

Trigger Max Frame Rate: 31030.303

Source: Software

Units: μ s (Time)

☐ Gate on External Input

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

Gocator 1300 series sensors are limited to sending data at 10 kHz over the analog output channel.

Therefore, if you configure a sensor so that it runs at a speed higher than 10 kHz in the **Trigger** panel on the **Scan** page, and configure a measurement to be sent on the analog channel under **Analog** on the **Output** page, you will get analog data drops.

To achieve a 10 kHz analog output rate, you must check **Scheduled** on the **Output** page and configure scheduled output.

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 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 <i>Digital Input</i> on page 439 for more information on connecting external input to Gocator sensors.
Behavior	Encoder	Specifies how the Gocator sensor is triggered when the target moves. Can be Track Backward, Ignore Backward, or Bi-Directional. See <i>Triggers</i> on page 85 for more information on these behaviors.
Spacing	Encoder, External Input	Specifies the distance between triggers (mm). Internally the Gocator sensor rounds the spacing to a multiple of the encoder resolution.
Reversal Distance	Encoder	When encoder triggering is set to Bi-Directional, use this setting to ignore jitter or vibrations in your transport system by specifying what distance the target must travel before a direction change is triggered. One of the following: Auto: The distance is automatically set by multiplying the value in Spacing by 3. Custom: Set the distance (in millimeters). Various functions in the sensor depend on this value to explicitly determine the point where direction change is triggered. Set this value larger than the maximum vibrations you see in your transport system.
Units	External Input, Software	Specifies whether the trigger delay, output delay, and output scheduled command operate in the time or the encoder domain. The unit is implicitly set to microseconds with Time trigger source. The unit is implicitly set to millimeters with Encoder trigger source.
Trigger Delay	External Input	Controls the amount of time or the distance the sensor waits before producing a frame after the external input is activated. This is used to compensate for the positional difference between the source of the external input trigger (e.g., photocells) and the sensor. Trigger delay is only supported in single exposure mode; for details, see <i>Exposure</i> on page 95.

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 .

Maximum Input Trigger Rate

 The maximum external input trigger rate in a system including Master 400 or higher is 20 kHz.

When using a standalone sensor or a sensor connected to a Master 100, the maximum trigger rate is 32 kHz. This rate is limited by the fall time of the signal, which depends on the V_{in} and duty cycles. To achieve the maximum trigger rate, the V_{in} and duty cycles must be adjusted as follows:

Maximum Speed	V_{in}	Maximum Duty Cycle
32 kHz	3.3 V	88%
32 kHz	5 V	56%
32 kHz	7 V	44%
32 kHz	10 V	34%

At 50% duty cycle, the maximum trigger rates are as follows:

V_{in}	Maximum Speed
3.3 V	34 kHz
5 V	34 kHz
10 V	22 kHz

Maximum Encoder Rate

On a standalone sensor, with the encoder directly wired into the I/O port or through a Master 100, the maximum encoder rate is about 1 MHz.

For sensors connected through a Master 400 or higher, with the encoder signal supplied to the Master, the maximum rate is about 300 kHz.

Sensor

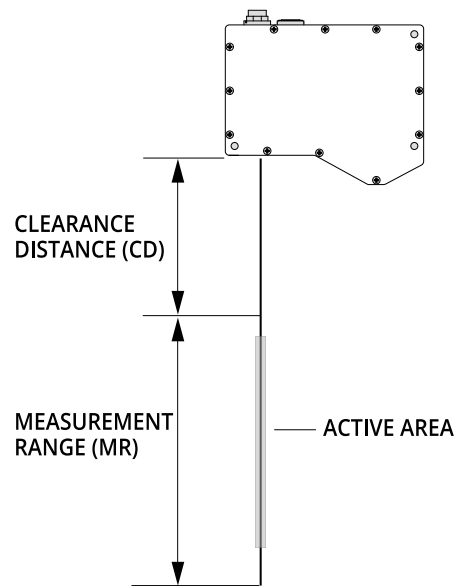
The following sections describe the settings that are configured in the **Sensor** panel on the **Scan** page.

Active Area

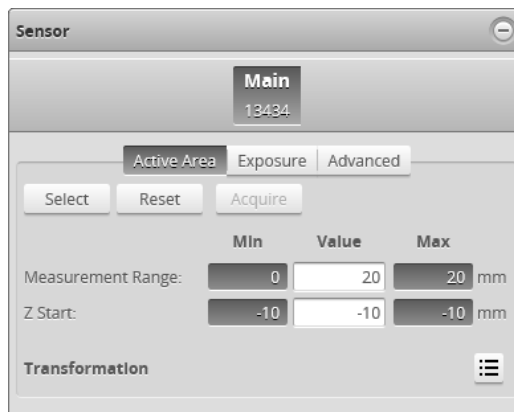
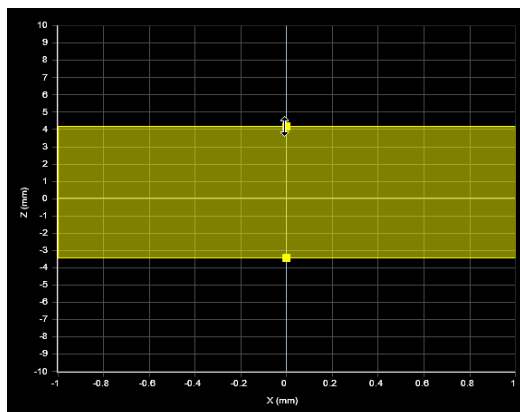
Active area refers to the region within the sensor's maximum field of view that is used for data acquisition.

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. You can also reduce the active area to exclude areas that are affected by ambient light.

Active area is specified in sensor coordinates, rather than in system coordinates. As a result, if the sensor is already alignment calibrated, press the **Acquire** button to display uncalibrated data before configuring the active area. See *Coordinate Systems* on page 49 for more information on sensor and system coordinates.



Active area is set in the **Active Area** tab on the **Sensor** panel.



To set the active area:

1. Go to the **Scan** page.

2. Choose a mode other than Video mode.
3. Expand the **Sensor** panel by clicking on the panel header or the  button.
4. Click the button corresponding to the sensor you want to configure.
The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
Active area is specified separately for each sensor.
5. Click on the **Active Area** tab.
6. Click **Select**.
7. Click **Acquire** to see a scan while setting the active area.
Acquiring a scan while setting the active area can help you determine where to size and place the active area.
8. Set the active area.
Adjust the active area graphically in the data viewer or enter the values manually in the fields.
9. Click the **Save** button in the **Sensor** panel.
Click the **Cancel** button to cancel setting the active area.
10. Save the job in the **Toolbar** by clicking the **Save** button .

Transformations

The transformation settings determine how data is converted from sensor coordinates to system coordinates (for an overview on coordinate systems, see *Coordinate Systems* on page 49). Typically, transformations are set when you [align a sensor](#). However, you can also manually set values using the **Transformations** section of the **Active Area** tab on the [Sensor](#) panel.



Parameter	Description
Z Offset	Specifies the shift along the Z axis. A positive value shifts the data toward the sensor.
Angle	Specifies the tilt (rotation in the X-Z plane). A positive value rotates the profile counter-clockwise.

When applying the transformations, angular rotation is applied before the Z offset.

To configure transformation settings:

1. Go to the **Scan** page.
2. Choose a mode other than Video mode in the **Scan Mode** panel.
If Video mode is selected, you will not be able to change the settings.
3. Expand the **Sensor** panel by clicking on the panel header.

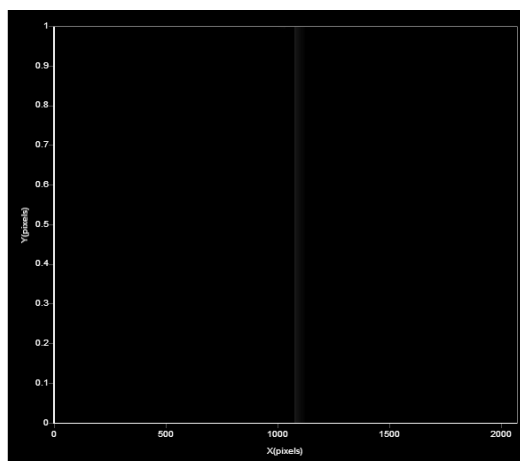
- Click the button corresponding to the sensor you want to configure.
The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
Transformations can be configured separately for each sensor.
- Expand the Transformations area by clicking on the expand button ☰.
See the table above for more information.
- Set the parameter values.
See the table above for more information.
- Save the job in the **Toolbar** by clicking the **Save** button 💾.
- Check that the transformation settings are applied correctly after the sensor is restarted.

Exposure

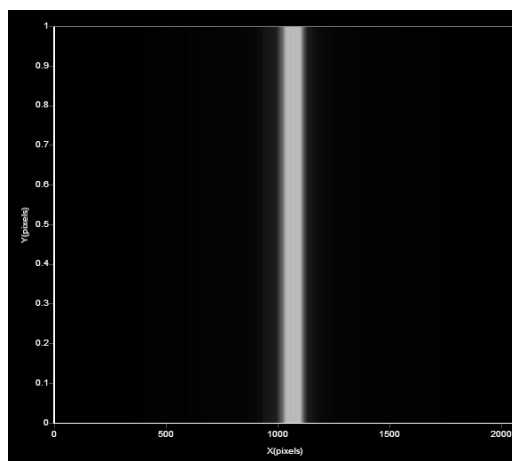
Exposure determines the duration of camera and light-source on-time. Longer exposures can be helpful to detect light on dark or distant surfaces, but increasing exposure time decreases the maximum speed. Different target surfaces may require different exposures for optimal results. Gocator sensors provide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

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

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



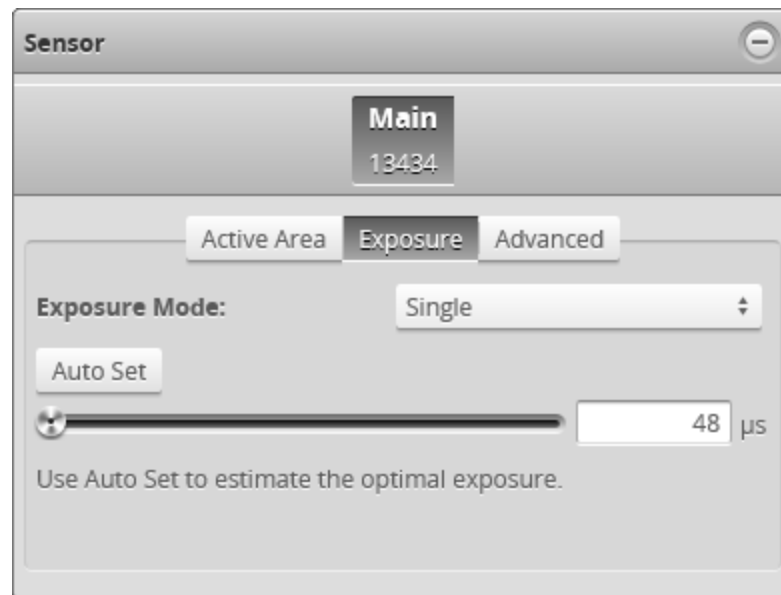
*Under-exposure:
Laser point is not detected.
Increase the exposure value.*



*Over-exposure:
Laser point is too bright.
Decrease the exposure value.*

Single Exposure

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



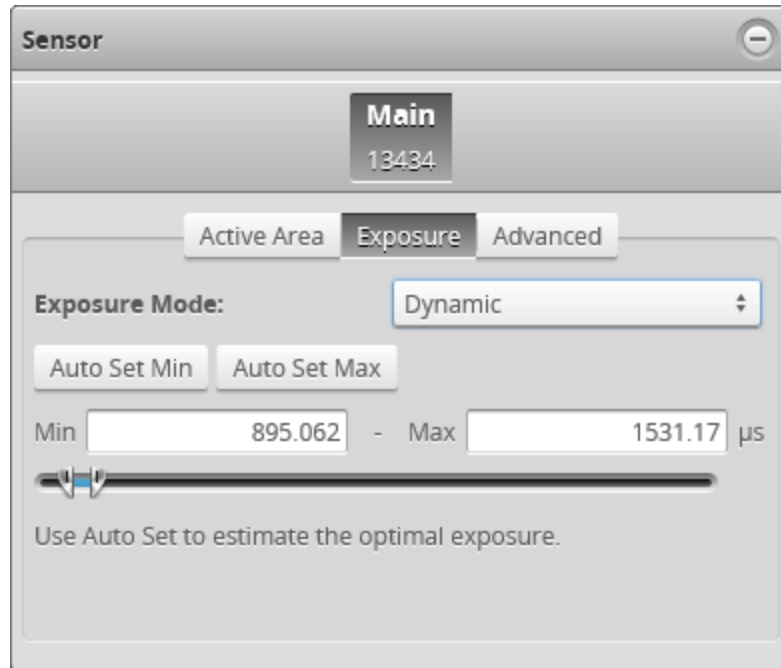
To enable single exposure:

1. Place a representative target in view of the sensor.
The target surface should be similar to the material that will normally be measured.
2. Go to the **Scan** page.
3. Expand the **Sensor** panel by clicking on the panel header or the  button.
4. Click the button corresponding to the sensor you want to configure.
The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
Exposure is configured separately for each sensor.
5. Click the **Exposure** tab.
6. Select **Single** from the **Exposure Mode** drop-down.
7. Edit the exposure setting by using the slider or by manually entering a value.
You can automatically tune the exposure by pressing the **Auto Set** button, which causes the sensor to turn on and tune the exposure time.
8. Run the sensor and check that laser ranging is satisfactory.

Dynamic Exposure

The sensor automatically uses past range information to adjust the exposure for subsequent exposures to yield the best range. This is used when the target surface changes from exposure to exposure (that is, from scan to scan).

 You can tune settings that control the exposure that is chosen by dynamic exposure in the [Material](#) tab.

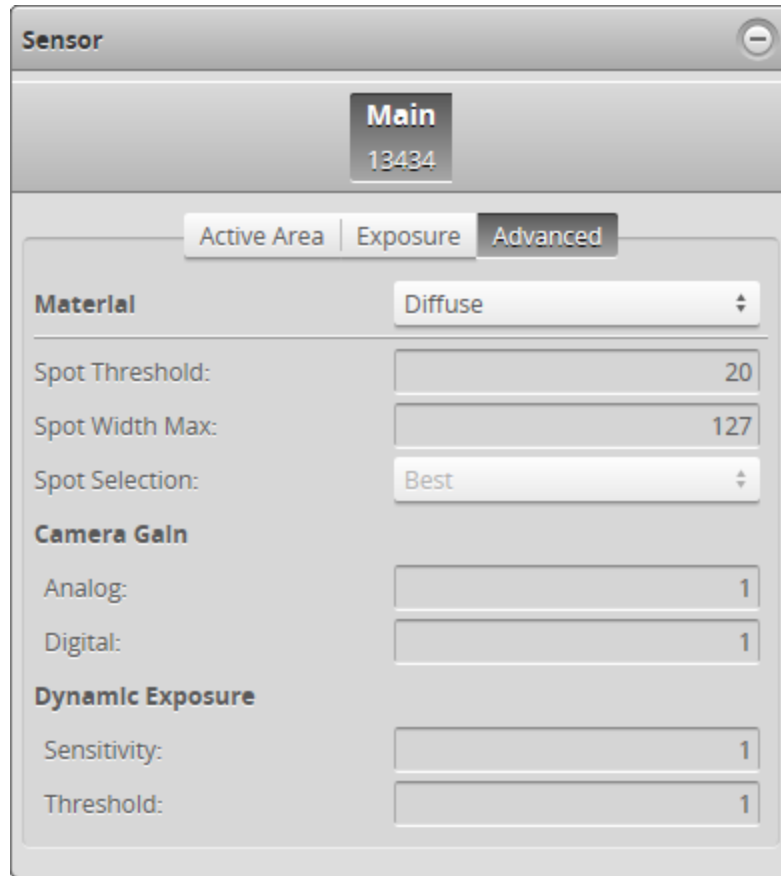


To enable dynamic exposure:

1. Go to the **Scan** page.
2. Expand the **Sensor** panel by clicking on the panel header or the  button.
3. Click the button corresponding to the sensor you want to configure.
The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
Exposure is configured separately for each sensor.
4. Click the **Exposure** tab.
5. Select **Dynamic** from the **Exposure Mode** drop-down.
6. Set the minimum and maximum exposure.
The auto-set function can be used to automatically set the exposure. First, place the brightest target in the field of view and press the **Auto Set Min** button to set the minimum exposure. Then, place the darkest target in the field of view and press the **Auto Set Max** button to set the maximum exposure.
7. Run the sensor and check that laser ranging is satisfactory.
If laser ranging is not satisfactory, adjust the exposure values manually. Switch to **Video** mode to use video to help tune the exposure; see *Exposure* on page 95 for details.

Advanced

The **Advanced** tab contains settings to configure material characteristics, camera gain, and dynamic exposure.



To configure advanced settings:

1. Go to the **Scan** page.
2. Switch to Video mode.
Using Video mode while configuring the settings lets you evaluate their impact.
3. Expand the **Sensor** panel by clicking on the panel header or the  button.
4. If you are configuring a dual- or multi-sensor system, click the button corresponding to the sensor you want to configure.
The button is labeled **Top**, **Bottom**, **Top-Left**, or **Top-Right**, depending on the system.
Settings can be configured separately for each sensor.
5. Click on the **Advanced** tab.
6. Configure material characteristics, camera gain, or dynamic exposure.
For more information, see *Material* on the next page and *Camera Gain and Dynamic Exposure* on the next page.
7. Save the job in the **Toolbar** by clicking the **Save** button .
8. Check that scan data is satisfactory.

Material

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

You can select preset material types in the **Materials** setting under the **Advanced** tab. The **Diffuse** material option is suitable for most materials.

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

Setting	Description
Spot Threshold	The minimum increase in intensity level between neighbouring pixels for a pixel to be considered the start of a potential spot. This setting is important for filtering false spots generated by sunlight reflection.
Spot Width Max	The maximum number of pixels a spot is allowed to span. This setting can be used to filter out data caused by background light if the unwanted light is wider than the laser and does not merge into the laser itself. A lower Spot Width Max setting reduces the chance of false detection, but limits the ability to detect features/surfaces that elongate the spot.
Spot Selection	Determines the spot selection method. Best selects the strongest spot in a given column on the imager. Top or Bottom : Top selects the spot farthest to the left on the imager, and Bottom selects the spot farthest to the right on the imager. These options can be useful in applications where there are reflections, flying sparks or smoke that are always on one side of the laser. None performs no spot filtering. If multiple spots are detected in an imager column, they are left as is. . Continuity considers adjacent horizontal data points on the imager to place spots on pixels, giving preference to more complete profile segments. The setting can improve scans in the presence of reflections and noise.

Various settings can affect how the **Material** settings behave. See *Spots and Dropouts* on page 114 for more information.

Camera Gain and Dynamic Exposure

You can set camera gain and dynamic exposure to improve data acquisition.

Setting	Description
Camera Gain	Analog camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor. Digital camera gain can be used when the application is severely exposure limited, yet dynamic range is not a critical factor.
Dynamic Exposure	Sensitivity controls the exposure that dynamic exposure converges to. The lower the

Setting	Description
	value, the lower the exposure Gocator will settle on. The trade-off is between the number of underexposed spots and the possibility of over-exposing. Threshold is the minimum number of spots for dynamic exposure to consider the profile point that make up the spot valid. If the number of spots is below this threshold, the algorithm will walk over the allowed exposure range slowly to find the correct exposure. Because this is slow, the Threshold value typically should be kept as low as possible, so this slow search is not used. These settings let you set tune how dynamic exposure settles on an exposure for a scan. For more information on Dynamic Exposure, see <i>Dynamic Exposure</i> on page 96.

Alignment

Alignment procedures are required to compensate for sensor mounting inaccuracies or to set a Z (height) reference plane. The alignment procedure also sets a common coordinate system for multi-sensor systems, and determines the encoder resolution (if present) and the speed of the transport system. (In many systems, the reference surface is a conveyor belt.)



Gocator sensors are pre-calibrated and ready to deliver data in engineering units (mm) out of the box. Alignment procedures do not affect sensor calibration.

A Gocator can be in one of two alignment states: Unaligned and Aligned. An indicator on the **Alignment** panel display UNALIGNED or ALIGNED, depending on the Gocator's state.

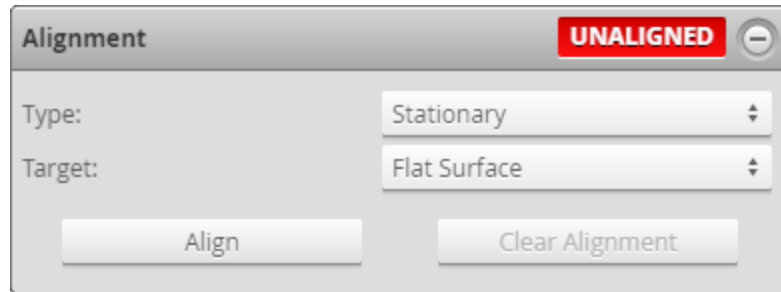
Alignment State

State	Explanation
Unaligned	The sensor or sensor system is not aligned. Data points are reported in sensor coordinates.
Aligned	The sensor is aligned using the alignment procedure (described below) or by manually modifying the values under Transformation in the Sensor tab on the Scan page (for more information, see <i>Transformations</i> on page 94). Data points are reported in system coordinates.

Once the alignment procedure has completed, the derived transformation values are displayed under **Transformations** in the **Sensor** panel.

Aligning Sensors

Alignment is configured and performed using the **Alignment** panel.



To prepare for alignment:

1. Choose an alignment reference in the **Manage** page if you have not already done so. For more information, see *Alignment Reference* on page 74.
2. Go to the **Scan** page.
3. Choose a mode other than Video mode in the **Scan Mode** 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 stationary alignment:

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**.
 - **Flat Surface:** Use this to align to a surface such as a conveyor.
 - **Bar:** Use this to align to a custom calibration bar.

Configure the characteristics of the target (bar dimensions and reference hole layout). For details on alignment targets, see *Alignment Targets* on page 25.



Degrees of Freedom: In stationary bar alignment, only one option is provided, namely, **X, Z, Y Angle**. This setting aligns X and Z offsets, as well as rotation around the Y axis.

4. 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 95).



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

5. Inspect alignment results.

Data points from all sensors should now be aligned to the alignment target surface. The base of the alignment target (or target surface) provides the origin for the system Z axis.

To perform moving alignment:

1. Do one of the following if you have not already done so.
 - If the system uses an encoder, configure encoder resolution. See *Encoder Resolution* on page 75 for more information.
 - If the system does not use an encoder, configure travel speed. See *Travel Speed* on page 75 for more information.
2. In the **Alignment** panel, select **Moving** as the **Type**.
3. Clear the previous alignment if present.
Press the **Clear Alignment** button to remove an existing alignment.
4. Select an alignment **Target**.
 - Select one of the disk **Disk** options to use a disk as the alignment reference.
 - Select **Bar** to use a custom calibration bar.

Configure the characteristics of the target (bar dimensions and reference hole layout); for details on alignment bars, see *Alignment Targets* on page 25.

Degrees of Freedom: In moving bar alignment, three options are available, which are combinations of different types of alignments. X, Y, and Z compensate for offsets on the X, Y, and Z axes, respectively. Y Angle and Z Angle compensate for rotation around the Y and Z axes, respectively. Compensating for X angle rotation is currently only possible by manually setting the rotation in the [Transformations](#) panel.



The Y offset, X angle, and Z angle [transformations](#) cannot be non-zero when **Uniform Spacing** is unchecked. Therefore, when aligning a sensor using a bar alignment target with **Uniform Spacing** unchecked, set the **Degrees of Freedom** setting to **X, Z, Y Angle**, which prevents these transformations from being non-zero.



On sensors aligned using Z angle or X angle, and to a lesser extent Y offset, CPU usage increases when scanning, which reduces the maximum scan speed.

Artifacts may appear in scan data on sensors aligned using Z angle or X angle if [encoder trigger spacing](#) is set too high (resulting in a low sampling rate).

5. Place the target under the sensor.
6. (Optional) Check the **Encoder or Speed Calibration** checkbox.
7. Click the **Align** button.
The sensors will start and then wait for the calibration target to pass through the laser plane. Alignment is performed simultaneously for all sensors. If the sensors do not align, check and adjust the [exposure settings](#).

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

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

Encoder Calibration

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

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

See *Coordinate Systems* on page 49 for definitions of coordinate axes. See *Alignment Targets* on page 25 for descriptions of calibration disks and bars.

See *Aligning Sensors* on page 100 for the procedure to perform alignment. After alignment, the coordinate system for laser ranges will change from sensor coordinates to system coordinates.

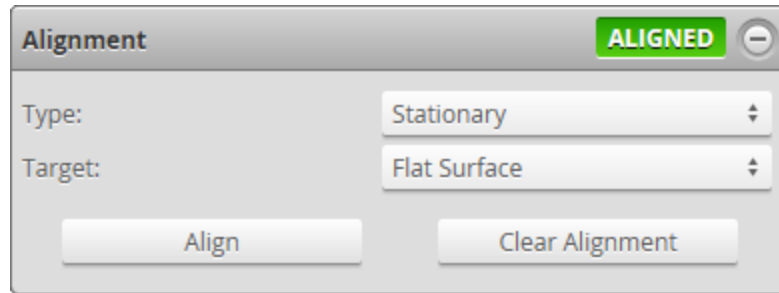
Alignment Types

Gocator sensors support two types of alignment: stationary or moving.

Type	Description
Stationary	Stationary is used when the alignment target does not move.
Moving	Moving is used when the alignment target moves beneath the sensor.

Clearing Alignment

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



To clear alignment:

1. Go to the **Scan** page.
2. Choose Range or Profile mode in the **Scan Mode** panel, depending on the type of measurement whose decision you need to configure.
If one of these modes 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 scan data along the X or Y axis to remove noise or clean it up *before* it is output or is used by measurement tools.

In some situations, such as when [Uniform Spacing](#) is disabled or when a sensor does not support filters, the filters panel is not displayed.

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.
Slope	Useful for measuring high-frequency height changes when they are surrounded by lower frequency changes on the surface.

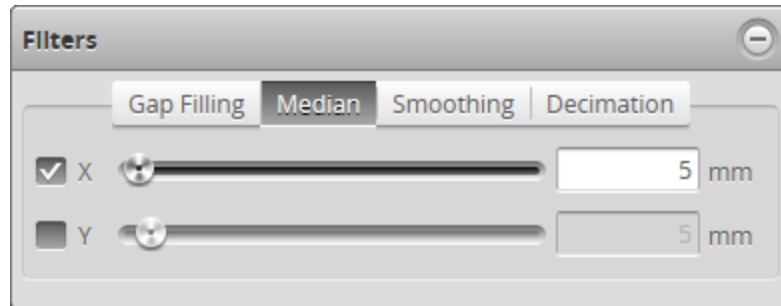
Filters are applied in the order displayed in the table above. The filters are configured in the **Filters** panel on the **Scan** page.

Median

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

 In Profile mode, gap filling is limited to the X axis. In Range mode, the filter is limited to the Y axis (direction of travel).

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



To configure X or Y median:

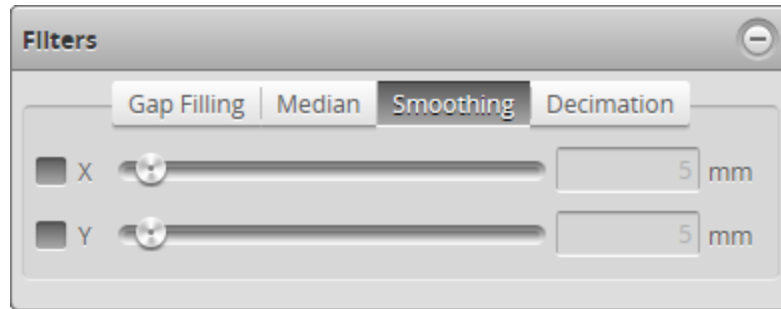
1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
If this mode is not selected, you will not be able to configure the median filter.
3. Expand the **Filters** panel by clicking on the panel header or the  button.
4. Click on the **Median** tab.
5. Enable the **X** or **Y** 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 within the same profile. Y smoothing works by calculating a moving average in the direction of travel at each X location.

 In Profile mode, gap filling is limited to the X axis. In Range mode, the filter is limited to the Y axis (direction of travel).

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

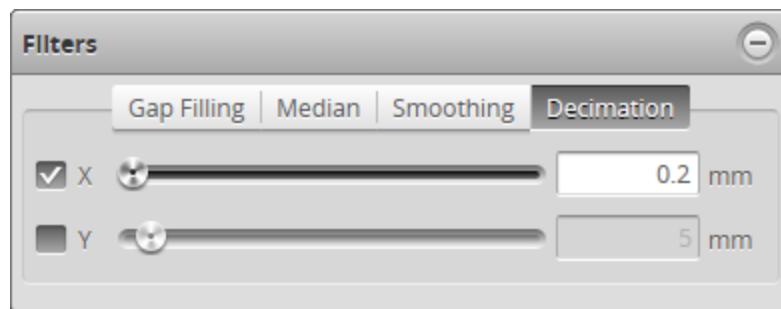


To configure X or Y smoothing:

1. Go to the **Scan** page.
2. Choose Range 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, only points every .2 millimeters will be used.



In Profile mode, gap filling is limited to the X axis. In Range mode, the filter is limited to the Y axis (direction of travel).

To configure X or Y decimation:

1. Go to the **Scan** page.
2. Choose Range mode in the **Scan Mode** panel.
If this mode is not selected, you will not be able to configure the decimation filter.

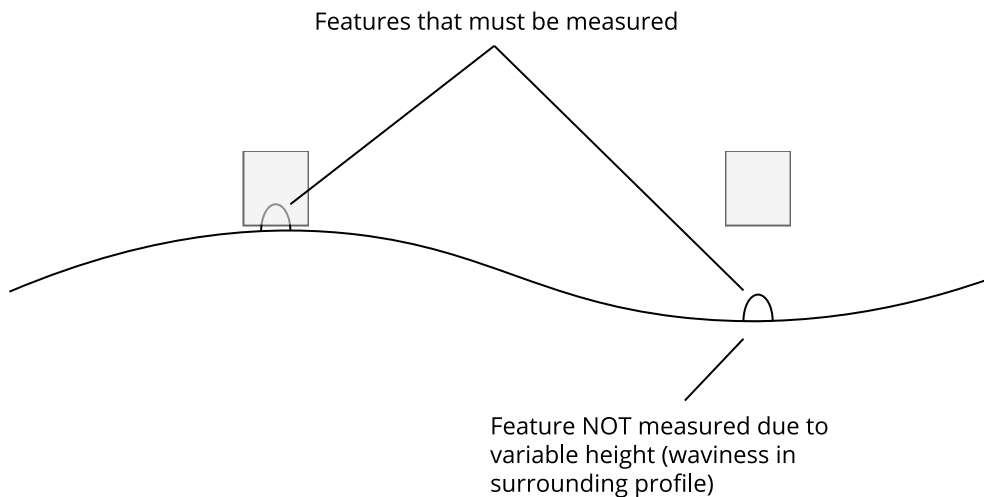
3. Expand the **Filters** panel by clicking on the panel header or the  button.
4. Click on the **Decimation** tab.
5. Enable the **X** or **Y** setting and select the decimation window value.
6. Save the job in the **Toolbar** by clicking the **Save** button .
7. Check that the laser profiling is satisfactory.

Slope

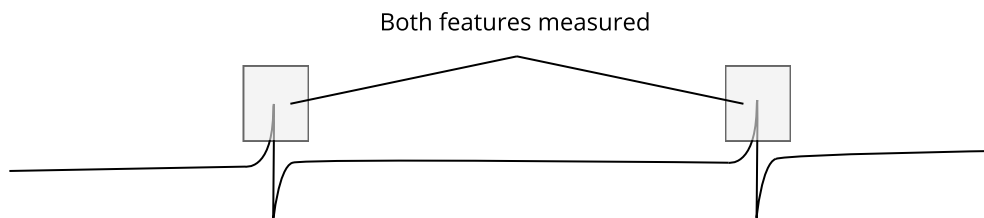
Slope modifies profile data in way that emphasizes high-frequency height changes when they are surrounded by lower frequency changes on the surface. You can use the filter, for example, to easily measure the position of edges on a wavy surface.

An example is a [generated profile](#) that looks like this:

Without Slope filter



With Slope filter



In the top profile (no filter applied), the second feature would be missed by a [Position Z](#) measurement, because the feature has moved beyond the region of interest defined for the measurement. When the filter is applied, the profile around the features is "evened out"—even though the overall height is

greater than the features that must be detected—and the more abrupt changes of the features are emphasized. As a result, the position of the features can easily be measured.

The filter can be used in both Range and Profile mode.

Profile Generation

The sensor can generate a profile by combining a series of ranges gathered along the direction of travel.

The sensor uses different methods to generate the data, depending on the needs of the application. Data generation is configured in the **Profile Generation** panel on the **Scan** page.

Profile Generation

Type:Continuous

*Part Detection is enabled

Profile Generation

Type:Fixed Length

Start Trigger:Sequential

Length:100mm

Profile Generation

Type:Variable Length

Max Length:100mm

Profile Generation

Type:Rotational

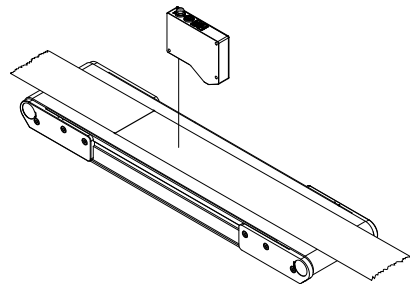
Encoder Resolution:11320.755ticks/rev

* Assuming one millimeter per degree.

The types in the table below correspond to the **Type** setting in the panel.

	When Type is set to Continuous , part detection is automatically enabled. When Type is set to any of the other settings, part detection can be enabled and disabled in the Part Detection panel. For descriptions of the settings that control part detection logic, <i>Part Detection</i> on page 111.
---	---

Continuous: The sensor continuously generates profiles of parts that are detected under the sensor. This type is typically used when the transport system continuously feeds material or parts under a sensor. The materials have a distinguishable start and stop edge, such as in when detecting wane (the curved part on boards cut from logs that must be removed).



Fixed Length: The sensor generates profiles of a fixed length (in mm) using the value in the **Length** setting. Like Continuous mode, Fixed Length mode is used when material or parts continuously pass under the sensor. Unlike Continuous mode, parts/material do not have distinguishable start and stop edge. Examples include measuring height characteristics of rubber extrusions or road roughness calculations.

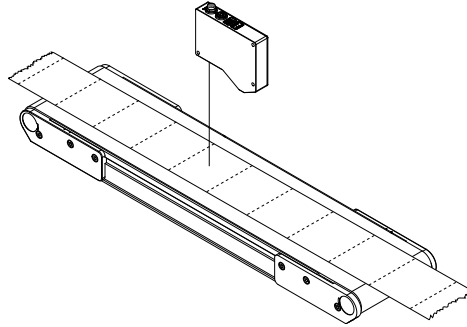
For correct length measurement, you should ensure that motion is calibrated (that is, encoder resolution for encoder triggers or travel speed time triggers).

Two types of start triggers are available:

- **Sequential:** Continuously generates back-to-back fixed length profiles.
- **External Input:** A pulse on the digital input triggers the generation of a single profile of fixed length.

For more information on connecting external input to a Gocator sensor, see *Digital Input* on page 439.

You can optionally enable part detection to process the profile after it has been generated, but the generation itself does not depend on the detection logic. To do this, check **Enabled** in the **Part Detection** panel.



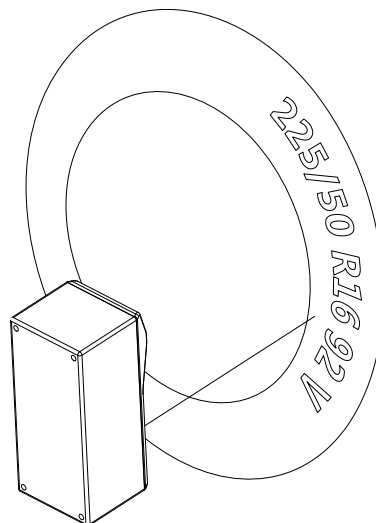
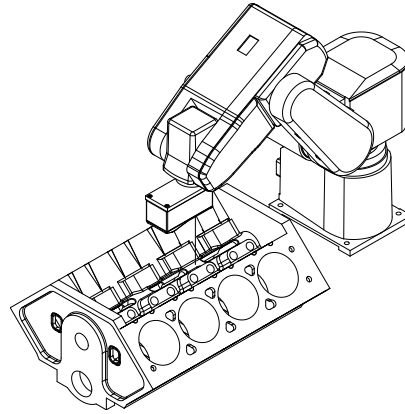
Variable Length: The sensor generates profiles of variable length while the external digital input is held high. If the value of the **Max Length** setting is reached while external input is still high, the next profile starts immediately. This mode is typically used in robot-mounted applications, for example, measuring the lengths of different parts on an engine block.

For correct length measurement, you should ensure that motion is calibrated (i.e., encoder resolution for encoder triggers or travel speed for time triggers).

For more information on connecting external input to a Gocator sensor, see *Digital Input* on page 439.

You can optionally enable part detection to process the profile after it has been generated, but the generation itself does not depend on the detection logic. To do this, check **Enabled** in the **Part Detection** panel.

Rotational: The sensor reorders ranges within a profile to be aligned with the encoder's index pulse so that the system knows when a full rotation has completed. That is, regardless of the radial position the sensor is started at, the generated profile always starts at the position of the index pulse. If the index pulse is not detected and the rotation circumference is met, the profile is dropped and the Encoder Index Drop indicator will be incremented. This mode is typically used in applications where measurements of circular objects or shafts need to be taken, such as tire tread inspection, or label



positioning on bottles.



To scan exactly one revolution of a circular target without knowing the circumference, manually set the encoder resolution (page 75) to 1, the encoder trigger spacing (page 85) to (number of encoder ticks per revolution) / (number of desired profiles per revolution), and **Encoder Resolution** in the **Profile Generation** panel to the number of encoder ticks per revolution.

You can optionally enable part detection to process the profile after it has been generated, but the generation itself does not depend on the detection logic. To do this, check **Enabled** in the **Part Detection** panel.

To configure profile generation:

1. Go to the **Scan** page and choose **Profile** in the **Scan Mode** panel.
If this mode is not selected, you will not be able to configure surface generation.
2. Expand the **Profile Generation** panel by clicking on the panel header or the  button.
3. Choose an option from the **Type** drop-down and any additional settings.
See the types and their settings described above.

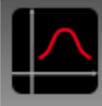
Part Detection

In Profile mode, the Gocator sensor can analyze profiles created by combining range values to identify discrete objects.

Part Detection

☒ Enabled

Frame Of Reference



Sensor



Part

Height Threshold:

mm

Threshold Direction:

Above

⬆

Gap Length:

mm

Padding Length:

mm

Min Part Length:

mm

Max Part Length:

mm

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 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 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 Part Length	Determines the minimum length of the part object.
Max Part Length	Determines the maximum length of the part object. When the object exceeds the 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. When Profile Generation is set to Continuous , only Part is available. See <i>Profile Generation</i> on page 108 for more information.

Sensor

When **Frame of Reference** is set to **Sensor**, the sensor's frame of reference is used.

Setting	Description
	The way the sensor's frame of reference is defined changes depending on the profile generation Type setting (<i>Profile Generation</i> on page 108 and for more information): - When parts are segmented from a continuous surface (the profile generation Type setting is set to Continuous), measurement values are relative to a Y origin at the center of the part (the same as for Part frame of reference; see below). - When parts are segmented from other types of profiles (the profile generation Type setting is set to Fixed Length, Variable Length, or Rotational), measurement values are relative to a Y origin at the center of the surface from which the part is segmented. Part When Frame of Reference is set to Part, all measurements are relative to the center of the bounding box of the part.

To set up part detection:

1. Go to the **Scan** page and choose **Profile** 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 **Profile Generation** is set to **Continuous**, part detection is always enabled.
4. Adjust the settings.
See the part detection parameters above for more information.

Data Viewer

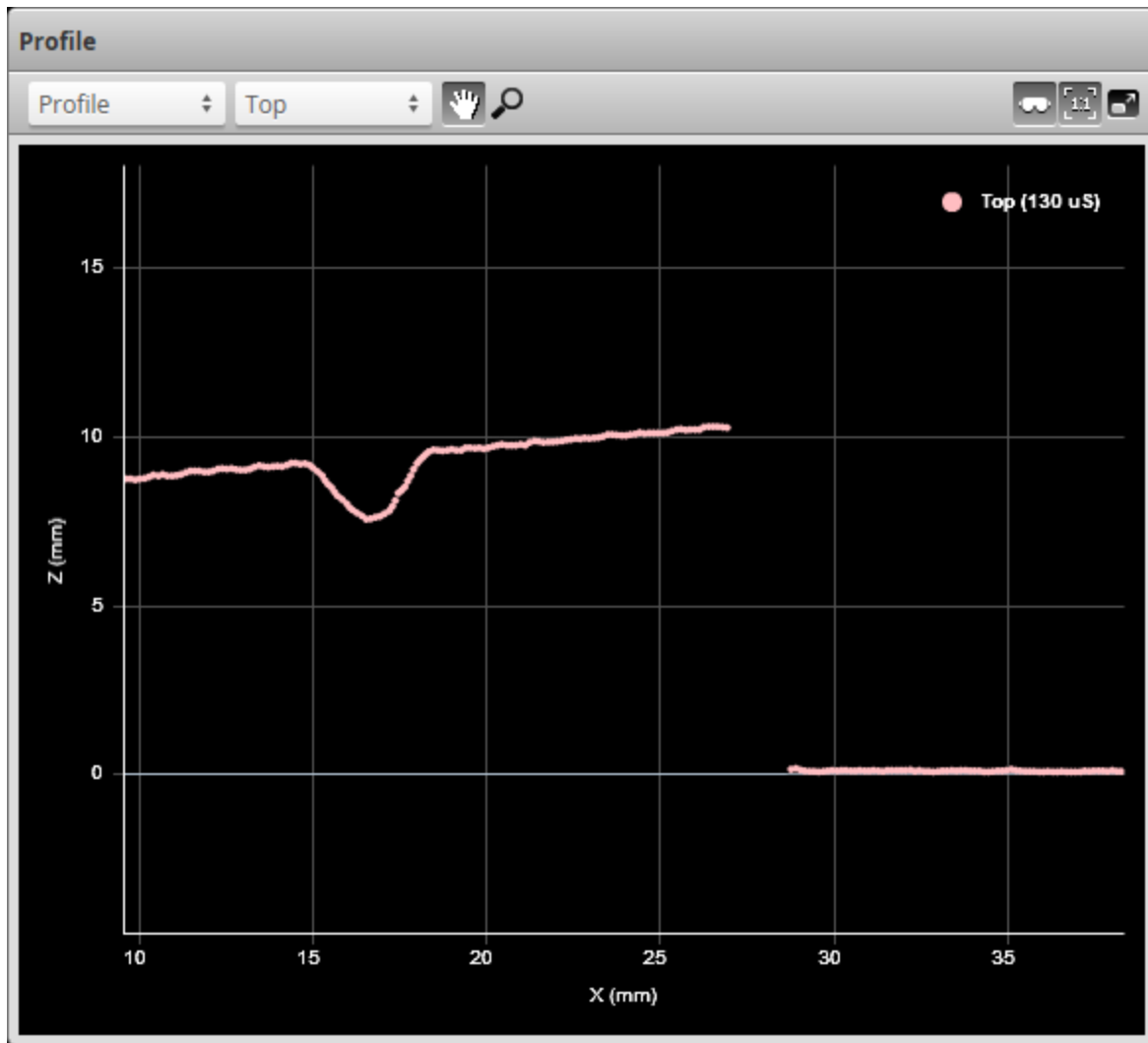
The data viewer can display video images, ranges, profiles, and intensity images. It is also used to configure the active area (*Active Area* on page 92) and measurement tools (see *Measurement* on page 120). The data viewer changes depending on the current operation mode and the panel that has been selected.

Data Viewer Controls

The data viewer is controlled by mouse clicks and by the buttons on the display toolbar. The mouse wheel can also be used for zooming in and out when viewing [generated profiles](#).

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

When the sensor is displaying profiles, a safety goggle mode button () is available in the data viewer. Enabling this mode changes some colors to ensure that profiles are visible in the data viewer when wearing laser safety goggles.



Video Mode

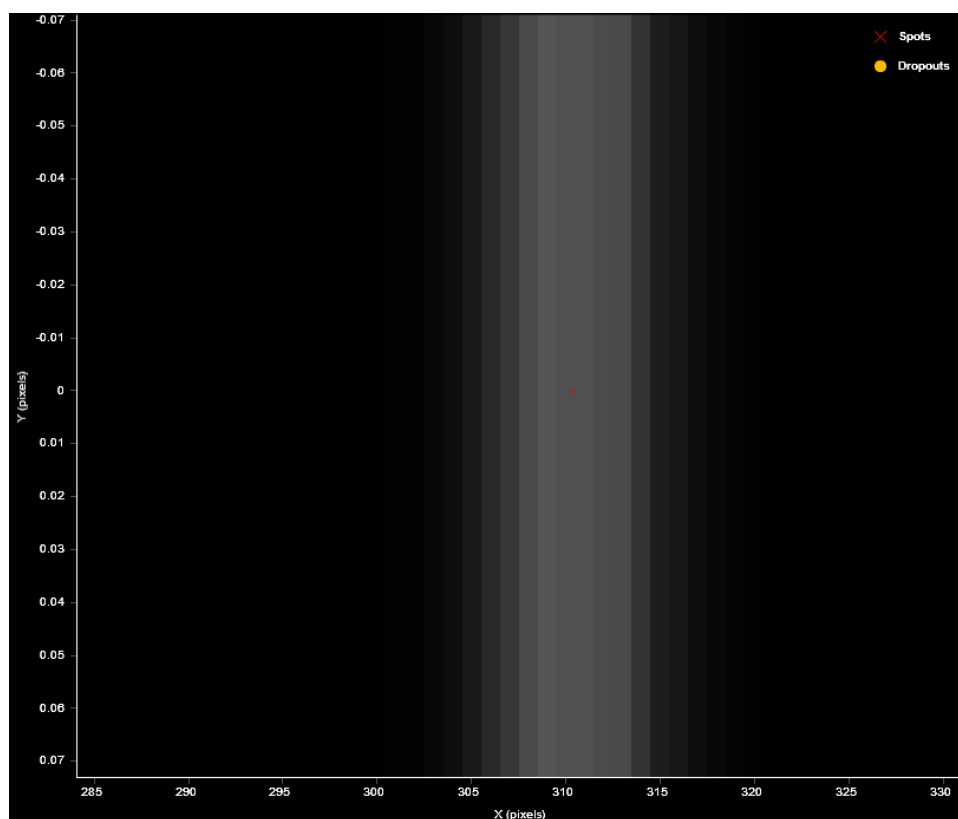
In Video mode, the data viewer displays images directly from the sensor's camera or cameras. In a dual-sensor system, camera images from either sensor can be displayed.

In this mode, you can configure the data viewer to display exposure, spot, and dropout information that can be useful in properly setting up the system for scanning.

Spots and Dropouts

Various settings can affect how the **Material** settings behave. In Video mode, you can examine how the **Material** settings are affected. To do this, in Video mode, check the **Show Spots** option at the top of the data viewer to overlay a representation of the spot in the data viewer.

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



To show data dropouts:

1. Go to the **Scan** page and choose **Video** mode in the **Scan Mode** panel.
2. check the **Show Dropouts** option at the top of the data viewer.

For more information on the material settings, see *Advanced* on page 97.

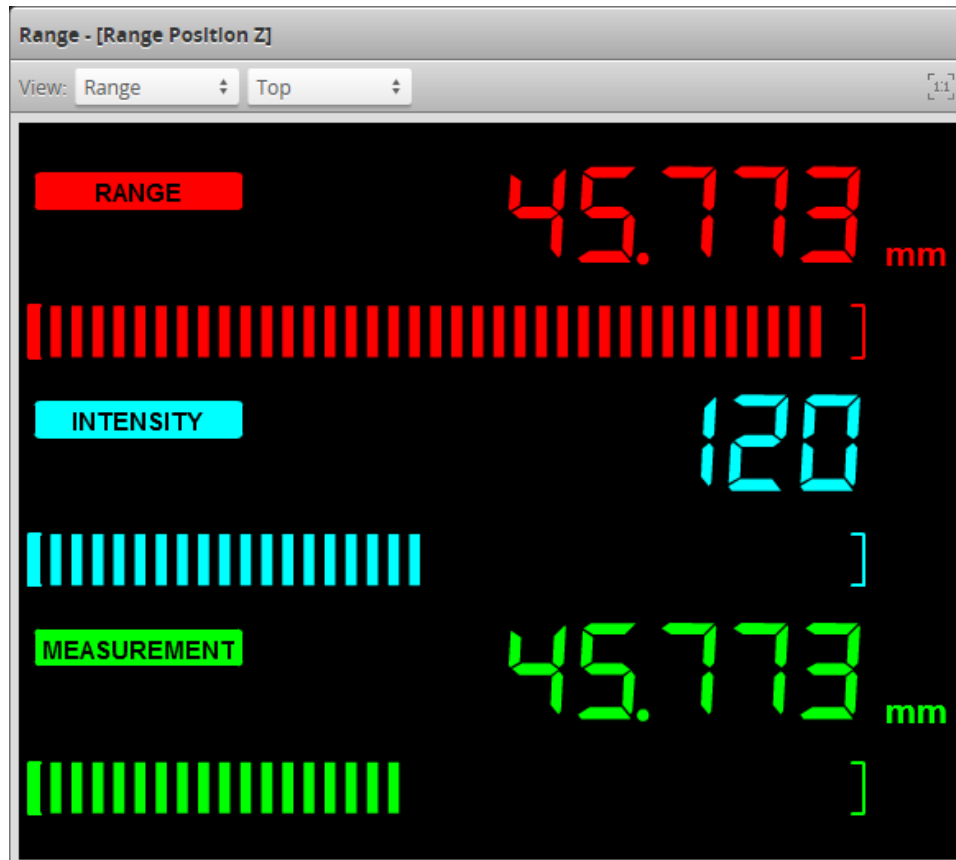
Range Mode

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

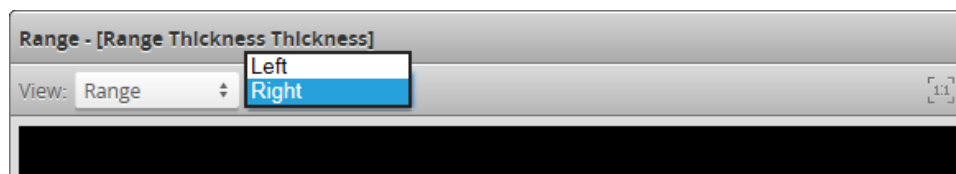
The Range value indicates where along the measurement range that the target falls. The bars indicate how close the target is to the near end (more bars, farther to the right) or the far end (fewer bars, farther to the left) of the measurement range. In the image below, the bars indicate that at 45.773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

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

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



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



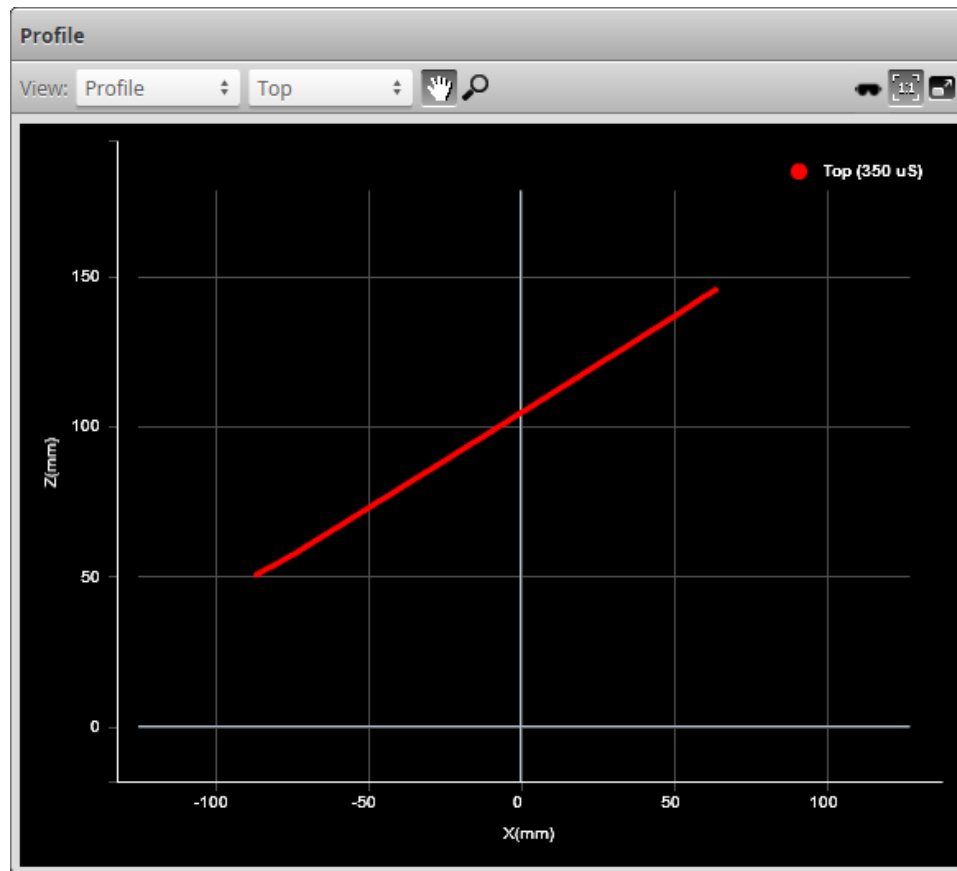
To manually select the display view in the Scan page:

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

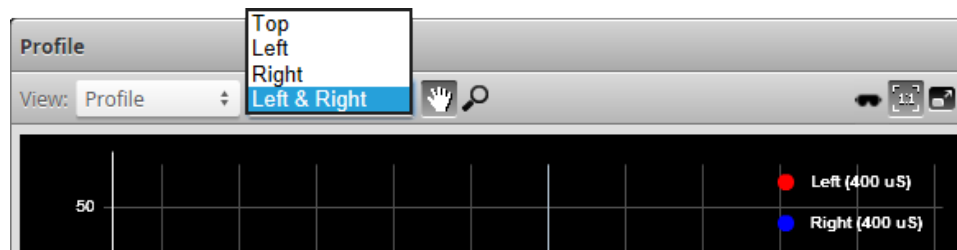
When the **Measure** page is active, the view of the display is set to the range source of the selected measurement tool (*Measurement* on page 120).

Profile Mode

When the Gocator is in Profile scan mode, the data viewer displays profile plots.



In a dual-sensor system, profiles from individual sensors or from a combined view can be displayed.



When in the **Scan** page, selecting a panel (e.g., **Sensor** or **Alignment** panel) automatically sets the display to the most appropriate display view.

To manually select the display view in the Scan page:

1. Go to the **Scan** page.
2. Choose **Profile** mode in the **Scan Mode** panel.
3. Select the view.

Top: View from a single sensor, from the top sensor in an opposite-layout dual-sensor system, or the combined view of sensors in the top position.

Bottom: View from the bottom sensor in an opposite-layout dual-sensor system.

Left: View from the left sensor in a dual-sensor system.

Right: View from the right sensor in a dual-sensor system.

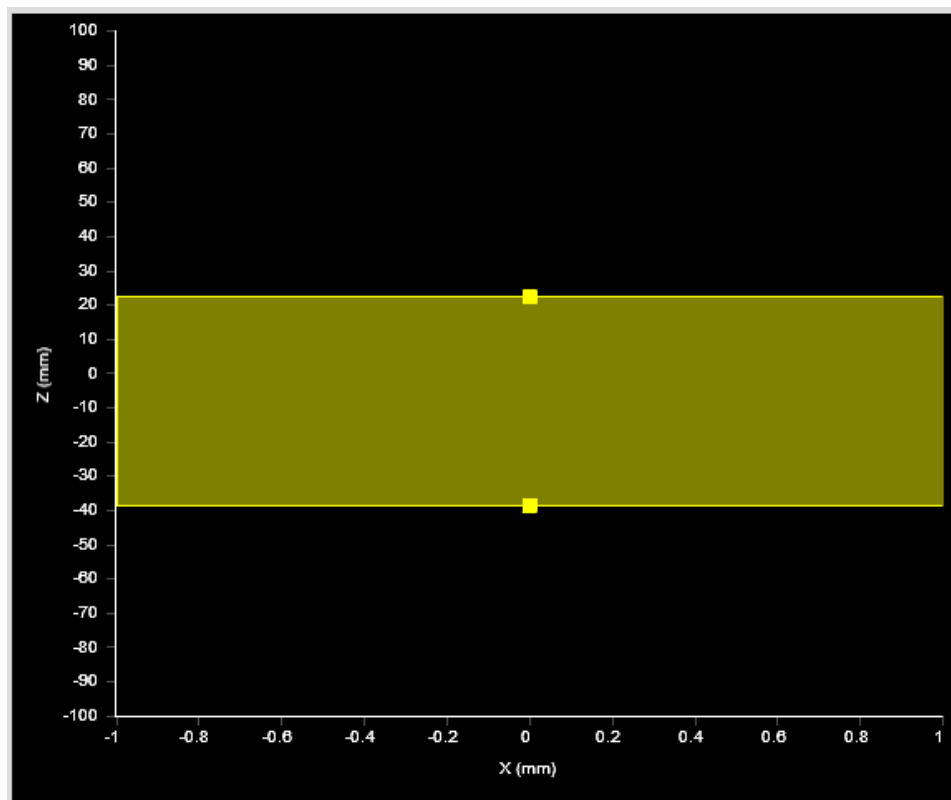
Left & Right: In a dual-sensor system, views from both sensors, displayed at the same time in the data viewer, using the coordinate systems of each sensor.

In the **Measure** page, the view of the display is set to the profile source of the selected measurement tool.

Region Definition

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

When the **Scan** page is active, the data viewer can be used to graphically configure the active area. The **Active Area** setting can also be configured manually by entering values into its fields and is found in the **Sensor** panel (see *Sensor* on page 92).



To set up a region of interest:

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

Intensity Output

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

To display intensity data, click the Intensity button (.



To be able to display intensity data, you must enable **Acquire Intensity** in the **Scan Mode** panel.

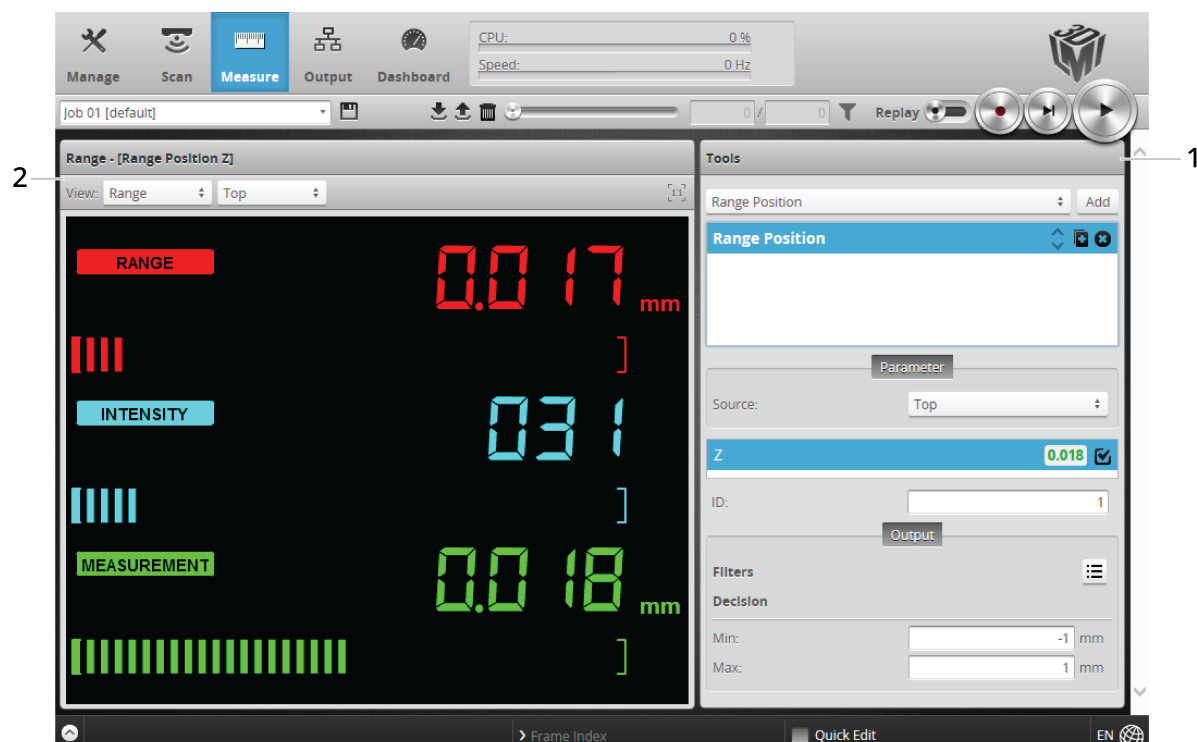
Measurement

The following sections describe Gocator's tools and measurements.

Measure Page Overview

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

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



Element	Description
1 Tools panel	Used to add, manage, and configure tools and measurements (see <i>Tools Panel</i> on the next page).
2 Data Viewer	Displays video and scan data, sets up tools, and displays result calipers related to the selected measurement. See <i>Data Viewer</i> below.

Data Viewer

When the **Measure** page is active, the data viewer can be used to graphically configure measurement regions. Measurement regions can also be configured manually in measurements by entering values into the provided fields (see *Regions* on page 122).

For information on controls in the data viewer, see *Data Viewer Controls* on page 113.

For instructions on how to set up measurement regions graphically, see *Region Definition* on page 118.

Tools Panel

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

Some settings apply to tools, and therefore to all measurements; these settings are found in the **Parameters** tab below the list of tools. Other settings apply to specific measurements, and are found in a **Parameters** tab below the list of measurements; not all measurements have parameters.

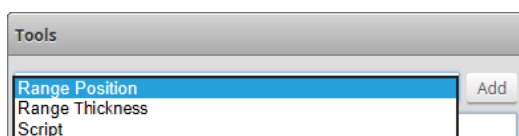
See *Range Measurement* on page 137 for information on the measurement tools and their settings.



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

Adding and Configuring a Measurement Tool

Adding a tool adds all of the tool's measurements to the **Tools** panel. You can then enable and configure the measurements selectively.



To add and configure a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range or Profile mode in the **Scan Mode** panel.
If one of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the 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 are added to the tool list. The tool parameters are listed in the area below the tool list.
6. (Optional) If you are running a dual-sensor system, choose the sensor that will provide data to the measurement tool in **Source**.
For more information on sources, see *Source* on the next page.
7. Select a measurement at the bottom of the tool panel.
8. Set any tool- or measurement-specific settings.
For tool- and measurement-specific settings, see the topics for the individual [range](#) or [profile](#) tools.
9. Set the **Min** and **Max** decision values.
For more information on decisions, see *Decisions* on page 128.
10. (Optional) Set one or more filters.

For more information on filters, see *Filters* on page 130.

11. (Optional) Set up anchoring.

For more information on anchoring, see *Measurement Anchoring* on page 131.

Source

For dual-sensor systems, you must specify which sensor, or combination of sensors, provides data for a measurement tool.

 The **Source** setting applies to all of a tool's measurements.

Depending on the layout you have selected, the **Source** drop-down will display one of the following (or a combination). For more information on layouts, see *Layout* on page 70.

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

To select the source:

1. Go to the **Measure** page by clicking on the **Measure** icon.

 The [scan mode](#) must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. If it is not already selected, click on the **Parameter** tab in the tool configuration area.
4. Select the profile source in the **Source** drop-down list.

Regions

Many measurement tools use user-defined regions to limit the area in which measurements occur. Unlike reducing the [active area](#), reducing the measurement region does not increase the maximum frame rate of the sensor.

 You can disable regions entirely and cause the measurement tool uses the entire [active area](#) by unchecking the checkbox next to the **Regions** setting.

All tools provide region settings under the upper **Parameters** tab. This region applies to all of a tool's measurements.



Region settings panel:

- ☒ **Region**
- X: -54.122 mm
- Z: -36.593 mm
- Width: 109.67 mm
- Height: 109.67 mm

Region settings are often found within expandable feature sections in the tool's panel.

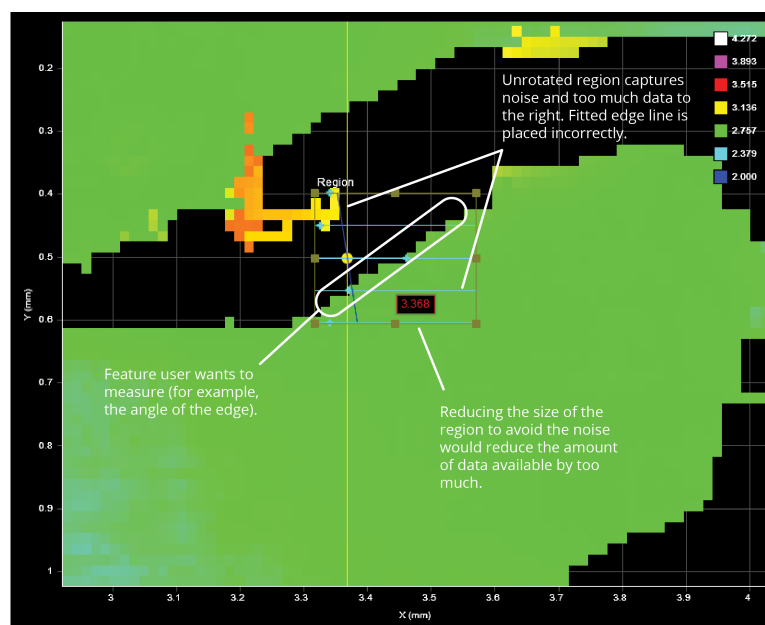
To configure regions:

1. Go to the **Measure** page by clicking on the **Measure** icon.

 The [scan mode](#) must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

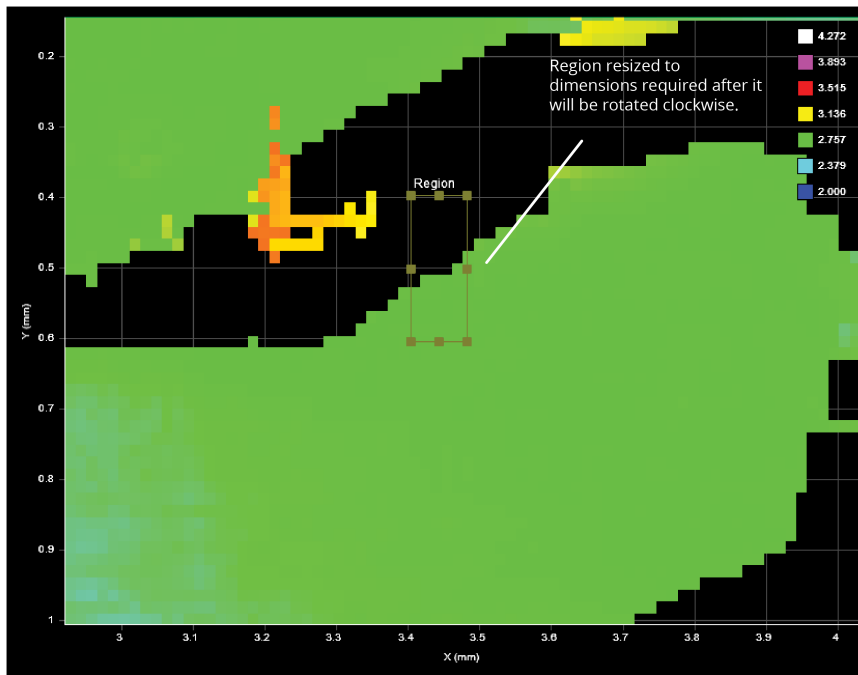
2. In the **Tools** panel, click on a tool in the tool list.
3. Configure the region using the mouse in the data viewer.
You can also configure regions manually by clicking the expand button (≡) and entering values in the fields. This is useful if you need to set precise values.

The measurement region of some tools can be rotated by setting the region's **Z Angle** to better accommodate features that are on an angle on a target. By rotating the measurement region, data not related to the feature can often be excluded, improving accuracy of measurements.



To rotate measurement regions:

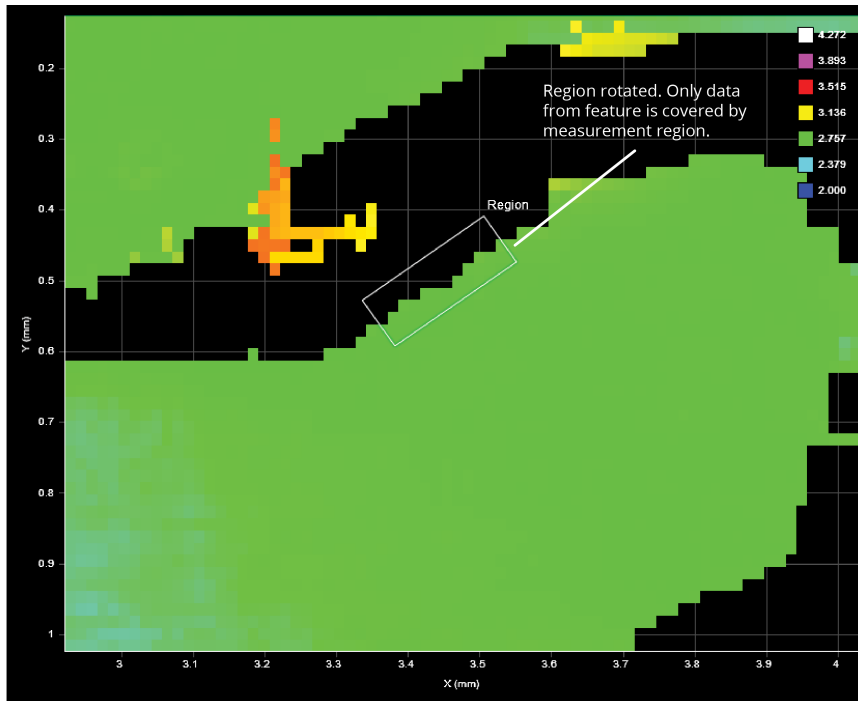
1. Determine the length and width of the region that will be required once it is rotated.



2. Expand the **Region** setting and then set a value in **Z Angle**.

Region	
X:	3.404 mm
Y:	0.397 mm
Z:	-16.725 mm
Width:	0.079 mm
Length:	0.207 mm
Height:	28.346 mm
Z angle:	55 °

The region rotates clockwise around the Z axis relative to the X axis.



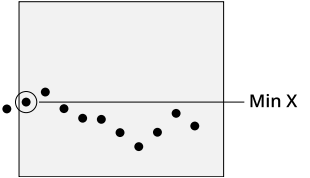
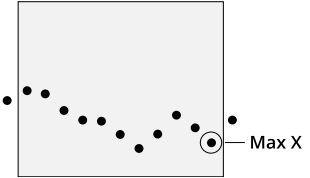
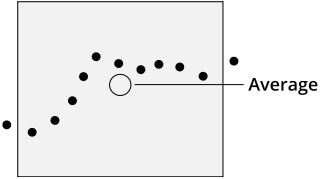
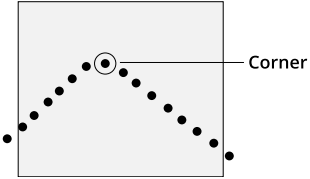
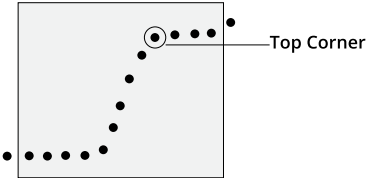
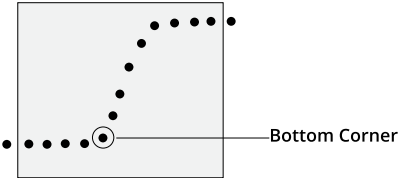
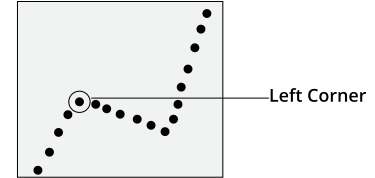
Once the region has been rotated, you cannot modify it in the data viewer using the mouse. You can however modify its dimensions and its location manually by changing the region's values in the **Region** setting.

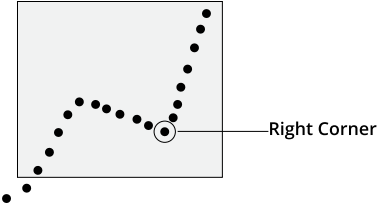
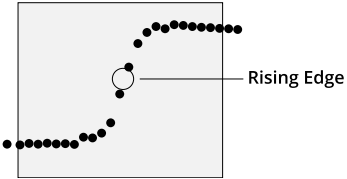
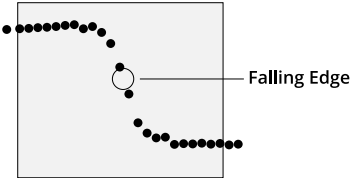
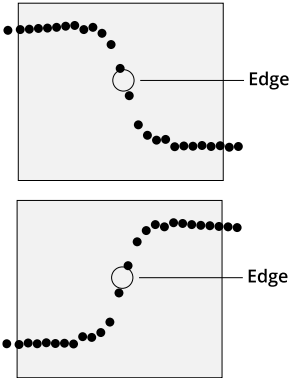
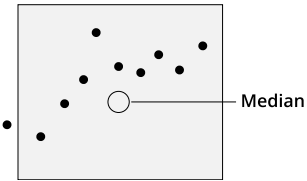
Feature Points

Dimensional and positional measurements detect *feature points* found within the defined [measurement region](#) and then compare measurement values taken at the selected point with minimum and maximum thresholds to produce a *decision*. Feature points are selected in one or more **Feature** dropdowns in a tool and are used for all of the tool's measurements.

The following types of points can be identified in a measurement region.

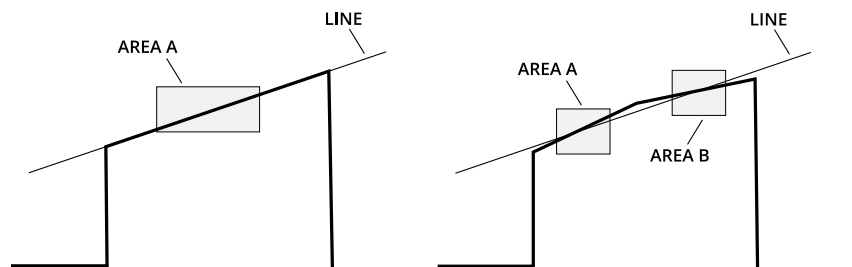
Point Type	Examples
Max Z Finds the point with the maximum Z value in the region of interest.	
Min Z Finds the point with the minimum Z value in the region of interest.	

Point Type	Examples
Min X Finds the point with the minimum X value in the region of interest.	
Max X Finds the point with the maximum X value in the region of interest.	
Average Determines the average location of points in the region of interest.	
Corner Finds a dominant corner in the region of interest, where corner is defined as a change in profile slope.	
Top Corner Finds the top-most corner in the region of interest, where corner is defined as a change in profile shape.	
Bottom Corner Finds the bottom-most corner in the region of interest, where corner is defined as a change in profile shape.	
Left Corner Finds the left-most corner in the region of interest, where corner is defined as a change in profile shape.	

Point Type	Examples
Right Corner Finds the right-most corner in the region of interest, where corner is defined as a change in profile shape.	
Rising Edge Finds a rising edge in the region of interest (moving from left to right).	
Falling Edge Finds a falling edge in the region of interest (moving from left to right).	
Any Edge Finds a rising or falling edge in the region of interest.	
Median Determines the median location of points in the region of interest.	

Fit Lines

Some measurements involve estimating lines in order to measure angles or intersection points. A fit line can be calculated using data from either one or two fit areas.



A line can be defined using one or two areas. Two areas can be used to bypass discontinuity in a line segment.

Geometric Features

Many [Profile tools](#) can output features that [Feature tools](#) can take as input to produce measurements. These features are called *geometric features*. Feature tools use these entities to produce measurements based on more complex geometry. (For more information on Feature tools, see *Feature Measurement* on page 179.)

Gocator's measurement tools can currently generate the following kinds of geometric features:

Points: A 2D point. Can be used for point-to-point or point-to-line measurements.

Lines: A straight line that is infinitely long. Useful for locating the orientation of an enclosure or part, or to intersect with another line to form a reference point that can be consumed by a Feature tool.

The following tables list Gocator's measurement tools and the geometric features they can generate:

Geometric features generated by Profile tools

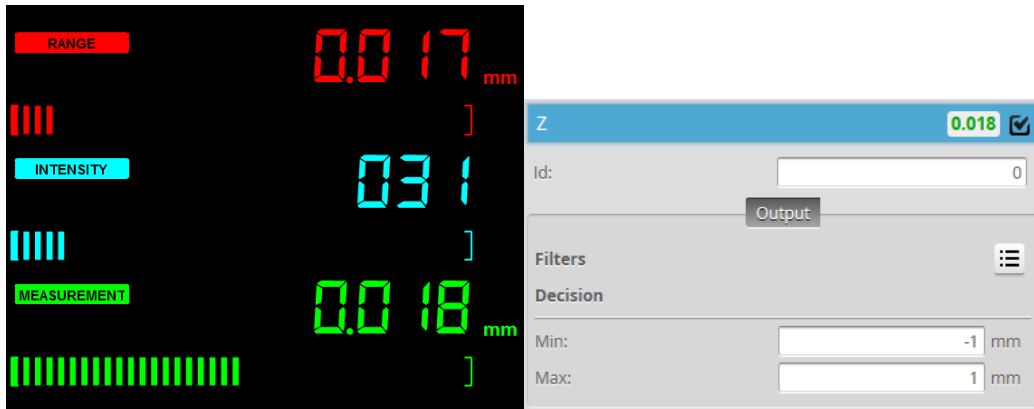
	Point						Line	
	Center	Point	Intersect	Corner	Error Min Point	Error Max Point	Line	Base
Tool								
Area	X							
Bounding Box	X			X				
Circle	X							
Dimension								
Groove								
Intersect			X				X	X
Line					X	X	X	
Panel								
Position		X						
Round Corner								
Strip								

The [Feature Intersect](#) tool can also produce an intersect point. [Script tools](#) do not currently take geometric features as input.

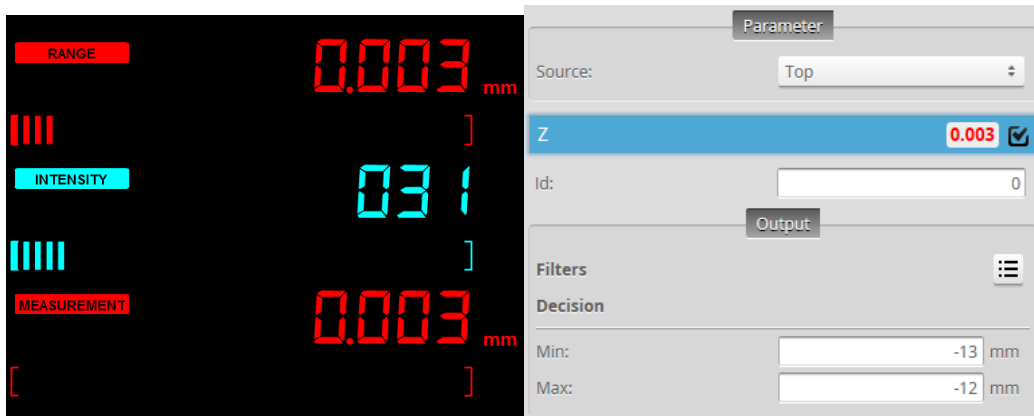
Decisions

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 data viewer and next to the measurement, these values are displayed in green. Otherwise, the decision state is *fail*. In the user interface, these values are displayed in red.

All measurements provide decision settings under the **Output** tab.



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



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

Along with measurement values, decisions can be sent to external programs and devices. In particular, decisions are often used with digital outputs to trigger an external event in response to a measurement. See *Output* on page 189 for more information on transmitting values and decisions.

To configure decisions:

1. Go to the **Measure** page by clicking on the **Measure** icon.

 The [scan mode](#) must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. In the measurement list, select a measurement.
To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 133 for instructions on how to enable a measurement.
4. Click on the **Output** tab.
For some measurements, only the **Output** tab is displayed.
5. Enter values in the **Min** and **Max** fields.

Filters

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

All measurements provide filter settings under the **Output** tab. The following settings are available.

Filter	Description
Scale and Offset	The Scale and Offset settings are applied to a measurement value according to the following formula: $\text{Scale} * \text{Value} + \text{Offset}$ Scale and Offset can be used to transform the output without the need to write a script. For example, to convert the measurement value from millimeters to thousands of an inch, set Scale to 39.37. To convert from radius to diameter, set Scale to 2. For more information on scripts, see <i>Scripts</i> on page 183.
Hold Last Valid	Holds the last valid value when the measurement is invalid.
Smoothing	Averages the <i>valid</i> measurements in the number of preceding frames specified in Samples. Use this to reduce the impact of random noise on a measurement's output. If Hold Last Valid is enabled, the smoothing filter uses the last valid measurement value until a valid value is encountered.
Preserve Invalid	When enabled, smoothing is only applied to valid measurements and not to invalid results: invalid results are not modified and are sent to output as is. When disabled, smoothing is applied to both valid and invalid results. (This setting is only visible when Smoothing is enabled.) If Hold Last Valid is enabled, results will always be valid, in which case this setting does nothing.

To configure the filters:

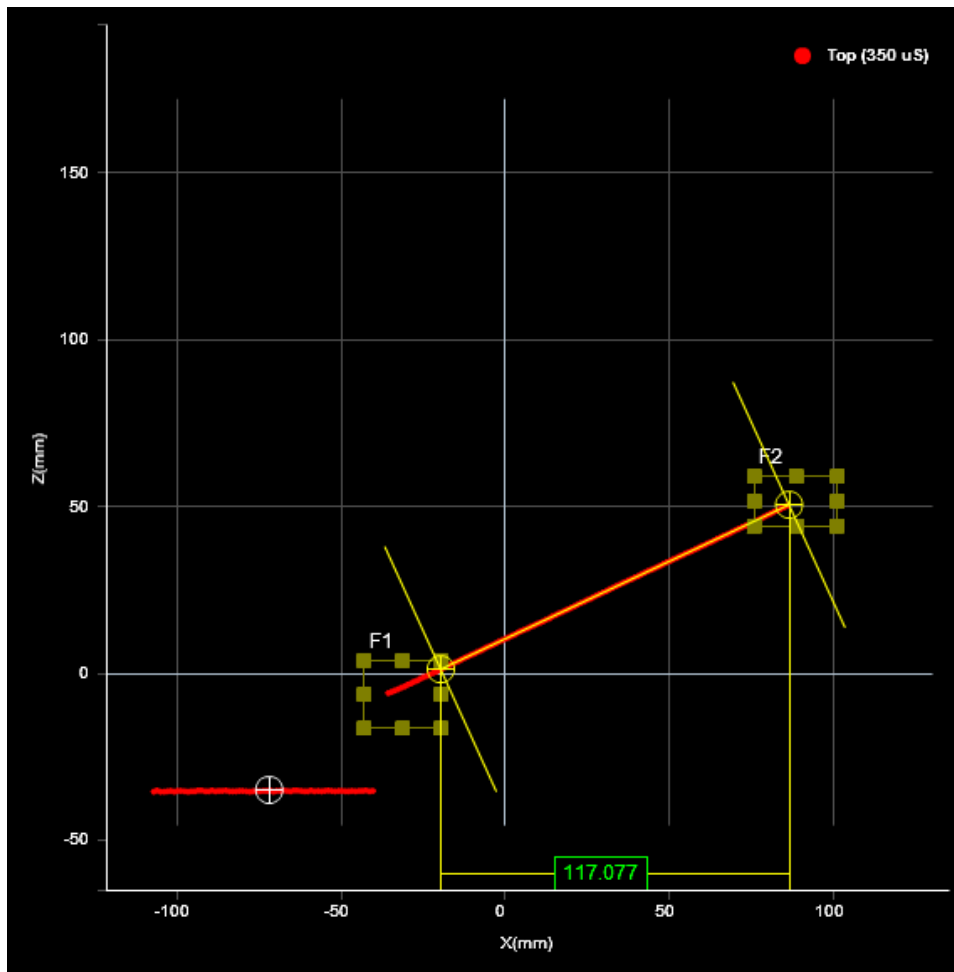
1. Go to the **Measure** page by clicking on the **Measure** icon.

The [scan mode](#) must be set to the type of measurement you need to configure. Otherwise, the wrong tools, or no tools, will be listed on the **Measure** page.

2. In the **Tools** panel, click on a tool in the tool list.
3. In the measurement list, select a measurement.
To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 133 for instructions on how to enable a measurement.
4. Click on the **Output** tab.
For some measurements, only the **Output** tab is displayed.
5. Expand the **Filters** panel by clicking on the panel header or the $\oplus$ button.
6. 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.



Parameter

Anchoring

X:

#3 - Profile Position X

Z:

Disabled

Width

☐

Height

☐

Distance

117.077

☒

Center X

☐

Center Z

☐

Id:

2

Output

Filters



Decision

Min:

110

mm

Max:

125

mm

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 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 profile tool to a measurement:

1. Place a representative target object in the field of view.
In Profile mode
 - a. Use the **Start** or **Snapshot** button to view live profile data to help position the target.
2. 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.

3. Adjust the anchoring tool's settings and measurement region.
You can adjust the measurement region graphically in the data viewer or manually by expanding the **Regions** area.

The position and size of the anchoring tool's measurement regions define the zone within which movement will be tracked.

See *Feature Points* on page 125 for more information on feature types.

4. Add the tool that you want to anchor.
Any tool can be anchored.
5. Adjust the tool and measurement settings, as well as the measurement regions, on a scan of the representative target.
6. Click on the tool's **Anchoring** tab.
7. Choose an anchor from one of the drop-down boxes.
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 anchoring measurement produces a valid measurement value. If the anchoring measurement is invalid, for example, if part moves outside its measurement region, the anchored tool will not show the measurement regions at all and an "Invalid-Anchor" message will be displayed in the tool panel.
8. Verify that the anchored tool works correctly on other scans of targets in which the part has moved slightly.

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.

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.

The screenshot shows the 'Tools' panel in the Gocator software. At the top, there is a search bar with 'Range Position' entered and an 'Add' button. Below this, the 'Range Position' tool is listed with expand, add, and delete icons. Under the 'Parameter' tab, the 'Source' is set to 'Top'. The 'Z' measurement is highlighted in blue, showing a value of '5.736' and a checked checkbox. Below this, the 'Id' is set to '0'. The 'Output' tab is active, showing 'Filters' and 'Decision' settings. Under 'Decision', the 'Min' value is '5 mm' and the 'Max' value is '6 mm'.

To enable a measurement:

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

To disable a measurement:

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

Editing a Tool or Measurement Name

You can change the names of tools you add in Gocator. You can also change the names of their measurements. 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 change a tool or measurement name:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
If this 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. Do one of the following:
 - **Tool:** In the tool list, double-click the tool name you want to change
 - **Measurement:** In a tool's measurement list, double-click the measurement name you want to change.
5. Type a new name.
6. Press the Tab or Enter key, or click outside the field.
The name will be changed.

Changing a Measurement ID

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

To edit a measurement ID:

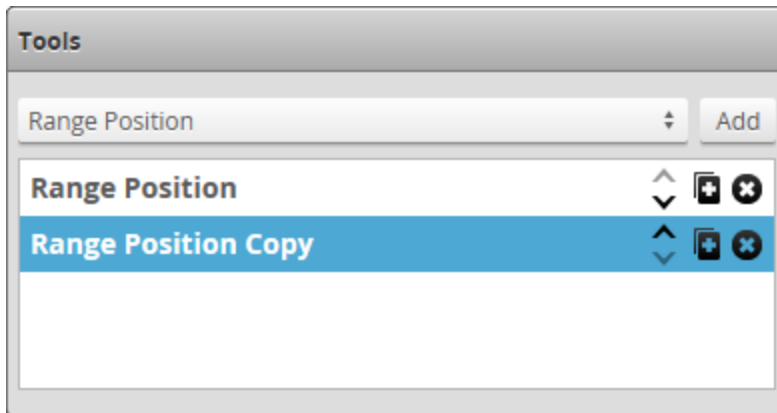
1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range mode in the **Scan Mode** panel.
If this mode is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the measurement list, select a measurement.
To select a measurement, it must be enabled. See *Enabling and Disabling Measurements* on page 133 for instructions on how to enable a measurement.
5. Click in the ID field.
6. Type 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.

Duplicating a Tool

You can quickly create a copy of a previously added tool in Gocator. All settings of the original are copied. This is useful, for example, when you need almost identical tools with only minor variations, such as different Min and Max values.

To duplicate a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range or Profile mode in the **Scan Mode** panel.
If one of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the tool list, click the Duplicate button (📄) of the tool you want to duplicate.
A copy of the tool appears below the original.



5. Configure the copy as desired and rename it if necessary.
For information on renaming a tool, see *Editing a Tool or Measurement Name* on page 134.

Removing a Tool

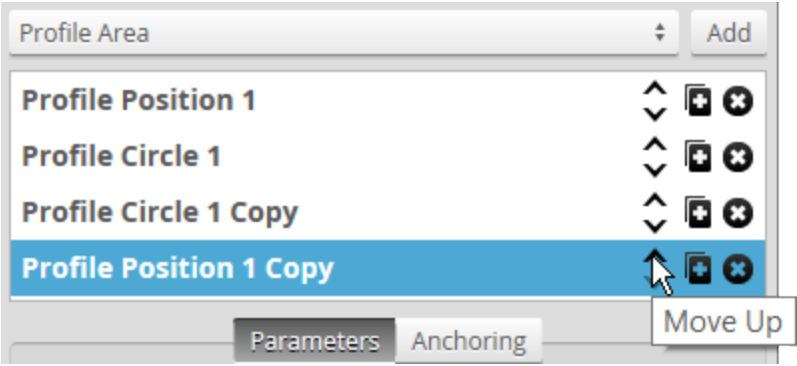
Removing a tool removes all of its associated measurements.

To remove a tool:

1. Go to the **Scan** page by clicking on the **Scan** icon.
2. Choose Range or Profile mode in the **Scan Mode** panel.
If this mode one of these modes is not selected, tools will not be available in the **Measure** panel.
3. Go to the **Measure** page by clicking on the **Measure** icon.
4. In the tool list, click on the Duplicate button (✖) of the tool you want to duplicate.
A copy of the tool appears below the original.

Reordering Tools

When you [add](#) or [duplicate](#) a tool, the tool is added to the bottom of the list in the **Tools** panel. You can reorder tools in the web interface to organize tools more logically. For example, you could group tools that output [geometric features](#) with the tools that use them. Or you could group tools you use as anchors with the tools that use those anchors.

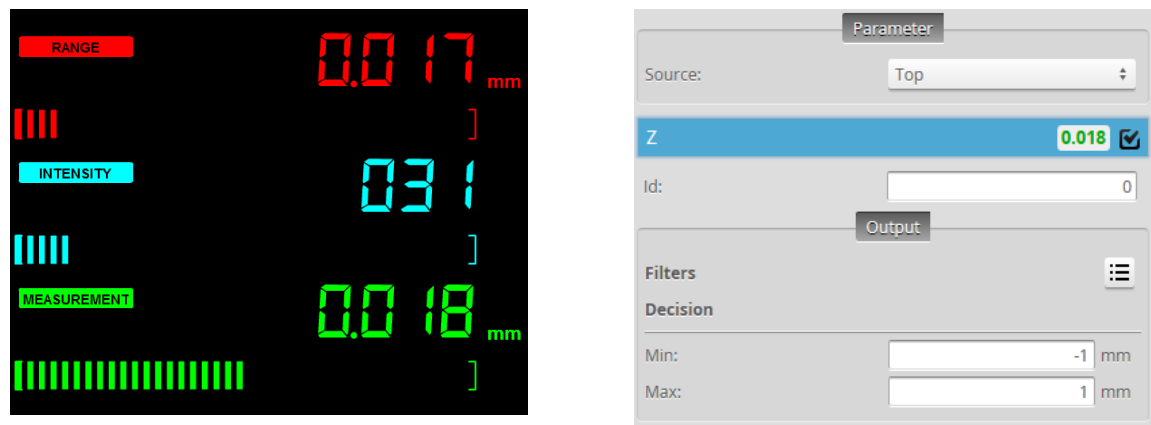


Range Measurement

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

Position

The Position tool finds the Z axis position of the laser range. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.



Measurements and Settings

Measurements

Measurement

Position Z

Determines the Z axis position of the laser range.

Illustration

Position Z —

Parameters

Parameter

Source

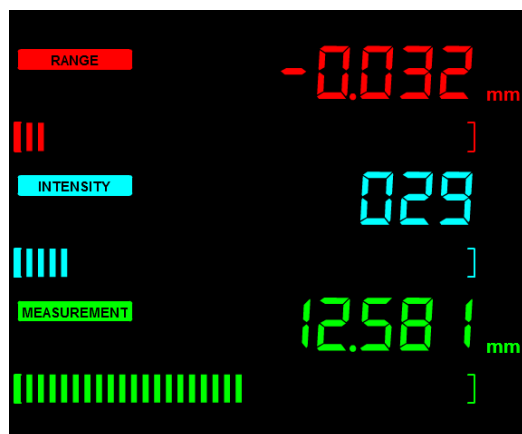
Description

The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see *Source* on page 122.

Parameter	Description
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Thickness

The Thickness tool determines the difference along the Z axis between two laser ranges. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.



Parameter

Source: Left & Right

☒ Absolute

Thickness 12.581

Id: 1

Output

Filters

Decision

Min: 12 mm

Max: 13 mm

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

$$\text{Thickness} = \text{Range}_{\text{Main}} - \text{Range}_{\text{Buddy}}$$

Measurements and Settings

Measurements

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

Parameters

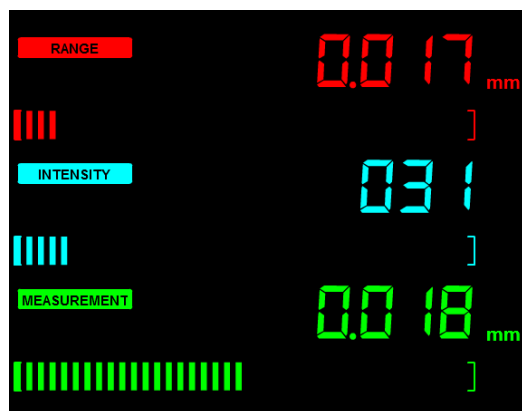
Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Absolute	Check the Absolute option to select absolute result
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

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 Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.

See *Scripts* on page 183 for more information on scripts.



Code

```

1 double PositionZ = Measurement_Value(0);
2
3 if (Measurement_Valid(0))
4 {
5     Output_Set(PositionZ + 10000, 1);
6 }
7 else
8 {
9     Output_Set(0, 0);
10 }
11

```

*Press save button or 'Ctrl+S' to apply change.
Press 'Esc' to exit full screen.

Output:

Add

Output 0

10000.018

✖

Id:

1

See *Scripts* on page 183 for more information on the script syntax.

To create or edit a Script measurement:

1. Add a new Script tool or select an existing Script measurement.

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

Profile Measurement

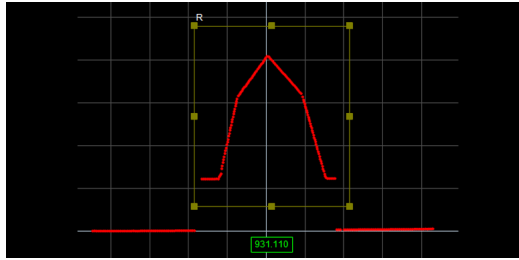
This section describes the profile measurement tools available in Gocator sensors.

Area

The Area tool determines the cross-sectional area within a region. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

All tools can use the measurements of other tools as anchors, to improve reliability and repeatability; for more information, see *Measurement Anchoring* on page 131.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameters
Anchoring

Source: Top
Type: Object
Baseline: Line
☒ Region
Line: 1 Region

Measurements
Features

Area 931.110 ☒
Centroid X ☐
Centroid Z ☐

ID: 0

Output

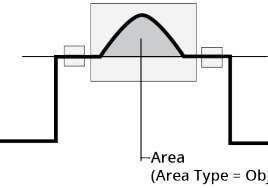
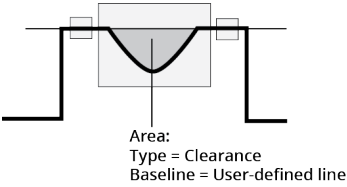
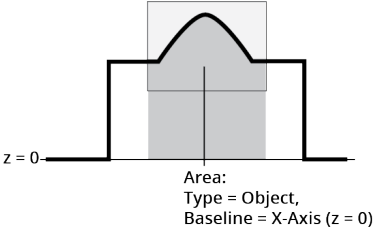
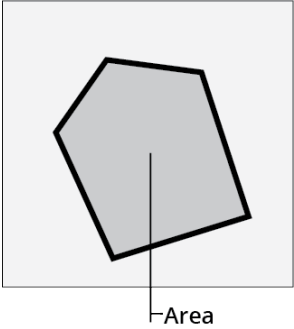
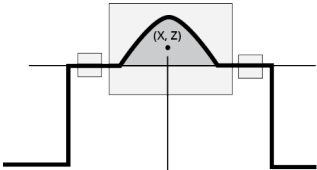
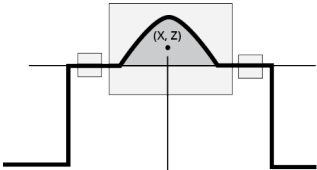
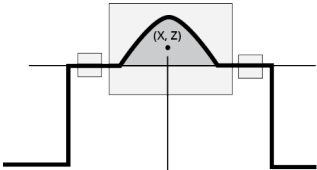
Filters
Decision

Min: 930 mm²
Max: 940 mm²

Areas are positive in regions where the profile is above the X axis. In contrast, areas are negative in regions where the profile is below the X axis.

Measurements, Features, and Settings

Measurements

Measurement	Illustration		
Area Measures the cross-sectional area within a region that is above or below a fitted baseline.	 Area (Area Type = Object)  Area: Type = Clearance Baseline = User-defined line Standalone, or dual-sensor setup in Wide orientation  z = 0 Area: Type = Object, Baseline = X-Axis (z = 0) Dual-sensor setup in Opposite orientation  Area <tr><td> Centroid X Determines the X position of the centroid of the area. </td><td>  (X, Z) Centroid: Type = Object Baseline = User-defined line </td></tr>	Centroid X Determines the X position of the centroid of the area.	 (X, Z) Centroid: Type = Object Baseline = User-defined line
Centroid X Determines the X position of the centroid of the area.	 (X, Z) Centroid: Type = Object Baseline = User-defined line		

Features

Type	Description
Center Point	The center point of the area.



For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Type	Object area type is for convex shapes above the baseline. Regions below the baseline are ignored. Clearance area type is for concave shapes below the baseline. Regions above the baseline are ignored.
Baseline	Baseline is the fit line that represents the line above which (Object clearance type) or below which (Clearance area type) the cross-sectional area is measured. When this parameter is set to Line, you must define a line in the Line parameter. See <i>Fit Lines</i> on page 127 for more information on fit lines. When this parameter is set to X-Axis, the baseline is set to $z = 0$.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.
Line	When Baseline (see above) is set to Line, set this to one of the following: 1 Region or 2 Regions: Lets you set one or two regions whose data the tool will use to fit a line. All Data: The tool uses all of the data in the active area. For more information on regions, see <i>Regions</i> on page 122). For more information on fit lines, see <i>Fit Lines</i> on page 127.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.



For more information on anchoring, see *Measurement Anchoring* on page 131.

Bounding Box

The Bounding Box tool provides measurements related to the smallest box that contains the profile (for example, X position, Z position, width, etc.).

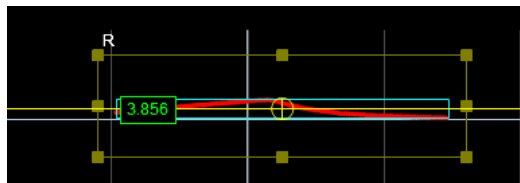
Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.

The bounding box provides the absolute position from which the Position centroids tools are referenced.



When you use measurement tools on parts, the coordinates returned are relative to the part. You can use the values returned by the Bounding Box tool's "Global" (see below) measurements as an offset in a Gocator script to convert the positional (X or Z) measurements of other measurement tools to [sensor](#) or [system](#) coordinates (depending on whether the sensor is aligned). For more information on Gocator scripts, see *Scripts* on page 183.



Parameters Anchoring

Source: Top

☒ Region

Measurements Features

X	4.703	☒
Z		☐
Width		☐
Height		☐
Global X		☐
Global Y		☐
Global Angle		☐

ID: 1

Output

Filters

Decision

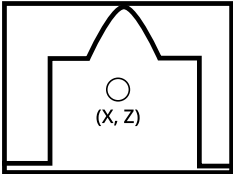
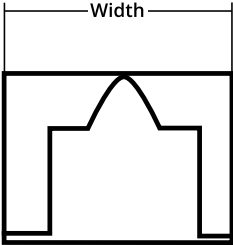
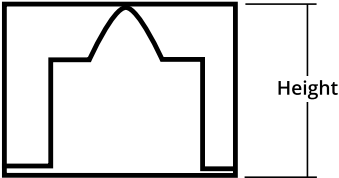
Min: 4 mm

Max: 5 mm

Measurement Panel

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Determines the X position of the center of the bounding box that contains the profile. The value returned is relative to the profile.	
Z Determines the Z position of the center of the bounding box that contains the profile. The value returned is relative to the profile.	
Width Determines the width of the bounding box that contains the profile. The width reports the dimension of the box in the direction of the minor axis.	
Height Determines the height (thickness) of the bounding box that contains the profile.	
Global X* This measurement is not intended for use with Gocator 1300 sensors.	
Global Y* This measurement is not intended for use with Gocator 1300 sensors.	
Global Angle* This measurement is not intended for use with Gocator 1300 sensors.	

Features

Type	Description
Center Point	The center point of the bounding box.
Corner Point	The lower left corner of the bounding box.

 For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.



For more information on anchoring, see *Measurement Anchoring* on page 131.

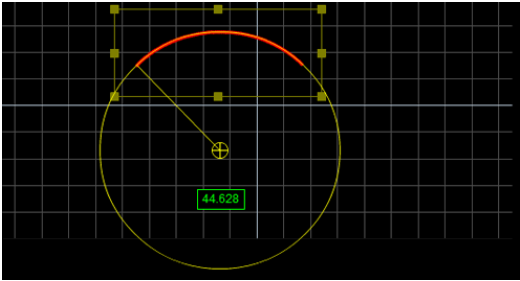
Circle

The Circle tool provides measurements that find the best-fitted circle to a profile and measure various characteristics of the circle. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



The tool may be unable to fit a circle to the profile when attempting the fit on a small number of relatively collinear data points.



ParametersAnchoring

Source:

Top

☒ Region

↺

⋮

MeasurementsFeatures

X

Z

Radius

44.628

⊗

ID:

2

Output

Filters

⋮

Decision

Min:

44 mm

Max:

46 mm

Measurements, Features, and Settings

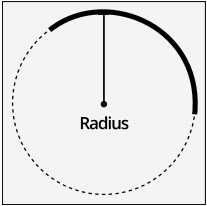
Measurements

Measurement

Illustration

Radius

Measures the radius of the circle.

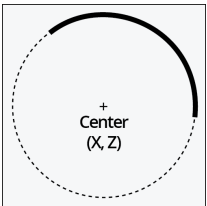


X

Finds the circle center position in the X axis.

Z

Finds the circle center position in the Z axis.



Features

Type

Description

Center Point

The center point of the fitted circle.

For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.



For more information on anchoring, see *Measurement Anchoring* on page 131.

Closed Area

The Closed Area tool determines the cross-sectional area within a region using point cloud data from a dual- or multi-sensor system.

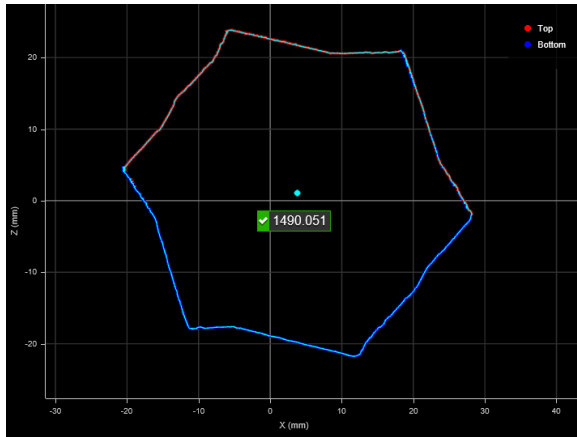
The tool is intended for use with roughly circular shaped profiles, or profiles that do not contain excessive concavity. The tool renders a polygon corresponding to the profile in the data viewer. Use this polygon to decide whether the tool can correctly calculate an acceptable representation of the profile. Minor gaps in the profile are permitted; the size of these gaps is configurable.

When the tool is used in conjunction with a script tool, you can calculate the volume of a target; for more information on the Script tool, see *Script* on page 178.

Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

All tools can use the measurements of other tools as anchors, to improve reliability and repeatability; for more information, see *Measurement Anchoring* on page 131.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameters
Anchoring

Source:
Top & Bottom

☐ Use Region

☐ Use Max Gap

Sample Spacing
1 degree

Measurements

Area
1490.834

ID:
1

Parameters
Output

Filters

Decision

Min:
1400 mm

Max:
1500 mm

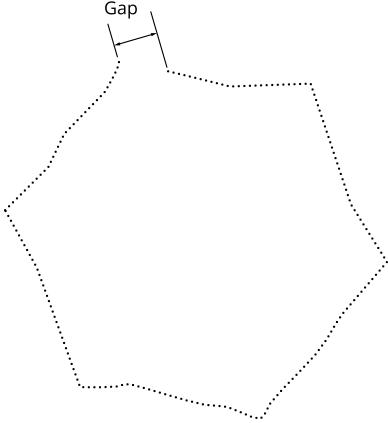
Measurements and Settings

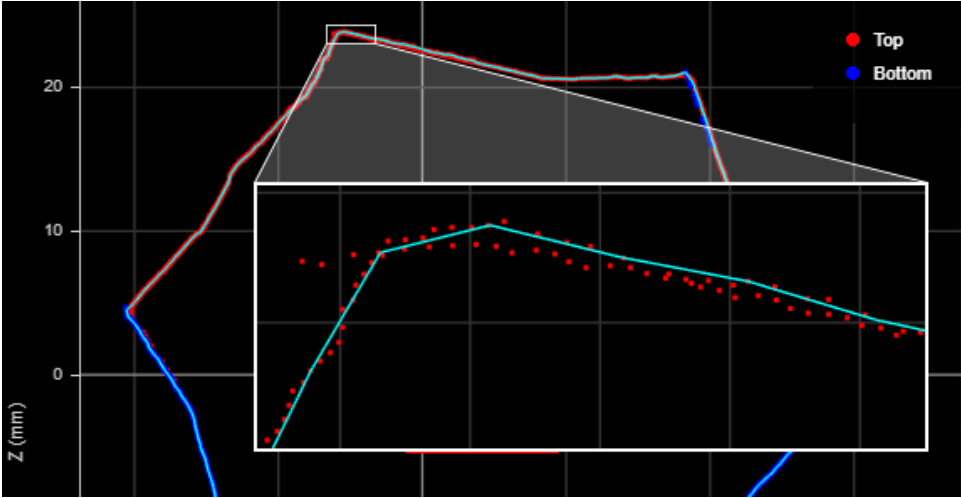
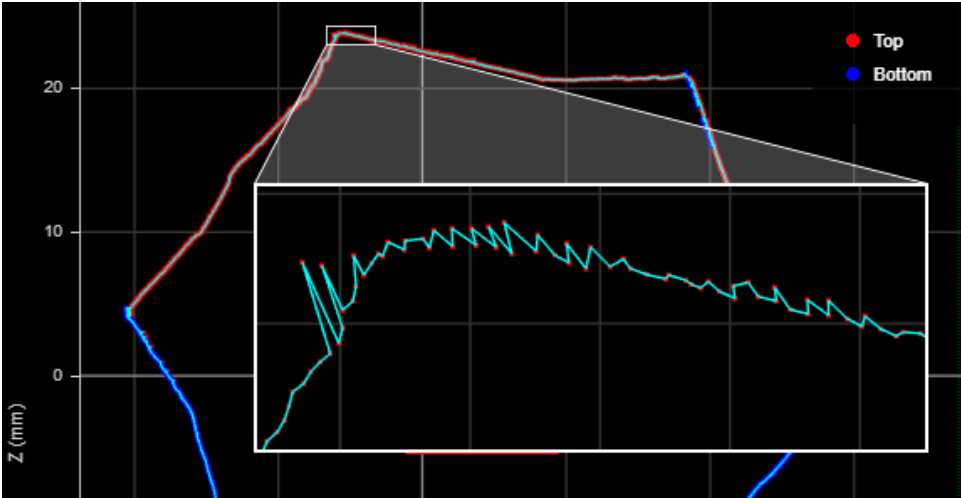
Measurements

Measurement	Illustration
Closed Area Measures the cross-sectional area within a region using data from a dual- or multi-sensor system.	

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Use Region	Indicates whether the tool uses a user-defined region. If this option is not checked, the tools uses data from the entire active area.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.
Use Max Gap	Indicates whether the tool uses the Max Gap setting (see below).

Parameter	Description
Max Gap	The maximum gap allowed between any two profile points on the contour of the target, in millimeters. In the following illustration of a profile, if the gap were greater than the value set in Max Gap, the tool would return an invalid value. 

Parameter	Description
Sample Spacing	The angle interval around the center of the profile the tool uses to calculate area. Enabling this setting and setting a value can increase the tool's performance. In the following image, the spacing is set to 1 degree. The polygon calculated from the profile points, which is then used to calculate the area, is simplified, increasing performance but reducing accuracy.  In the following image, Sampling Spacing is set to 0. Accuracy is increased, but performance is reduced.  If you set the value to 0, the tool uses the smallest angle permitted internally by the sampling engine.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

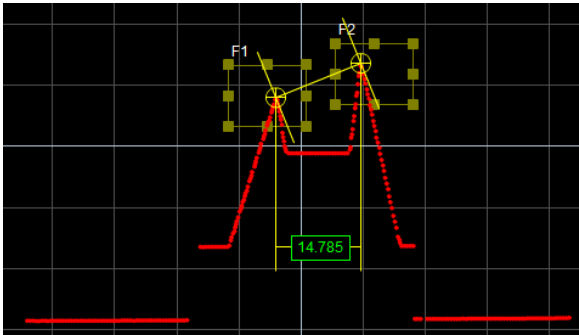
 A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

 For more information on anchoring, see *Measurement Anchoring* on page 131.

Dimension

The Dimension tool provides Width, Height, Distance, Center X, and Center Z measurements.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



ParametersAnchoring

Source:Top

Feature 1Max Z

Feature 2Max Z

Width

Height

Distance14.785

Center X

Center Z

Id:4

Output

Filters

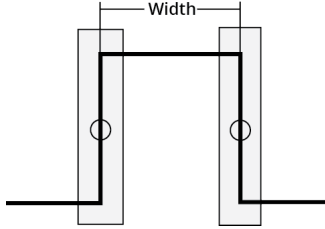
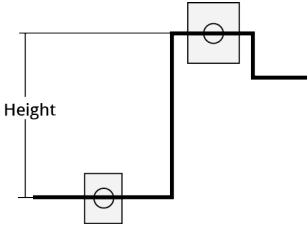
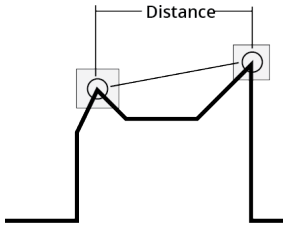
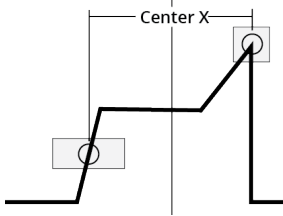
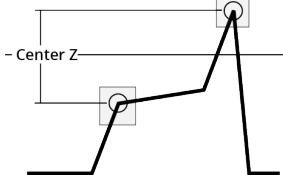
Decision

Min:14 mm

Max:15 mm

The tool's measurements require two feature points. See *Feature Points* on page 125 for information on point types and how to configure them.

Measurements

Measurement	Illustration
Width Determines the difference along the X axis between two feature points. The difference can be calculated as an absolute or signed result. The difference is calculated by: $\text{Width} = \text{Feature 2}_{X \text{ position}} - \text{Feature 1}_{X \text{ position}}$	
Height Determines the difference along the Z axis between two feature points. The difference can be expressed as an absolute or signed result. The difference is calculated by: $\text{Height} = \text{Feature 2}_{Z \text{ position}} - \text{Feature 1}_{Z \text{ position}}$	
Distance Determines the direct, Euclidean distance between two feature points.	
Center X Finds the average location of two features and measures the X axis position of the average location	
Center Z Finds the average location of two features and measures the Z axis position of the average location.	

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.

Parameter	Description
Feature 1 Feature 2	The Feature 1 and Feature 2 settings represent the two features the tool uses to perform measurements. For each, one of the following: • Max Z • Min Z • Max X • Min X • Corner • Average • Rising Edge • Falling Edge • Any Edge • Top Corner • Bottom Corner • Left Corner • Right Corner • Median To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button (⋮) and enter the values in the fields. For more information on regions, see <i>Regions</i> on page 122.
Absolute (Width and Height measurements only)	Determines if the result will be expressed as an absolute or a signed value.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

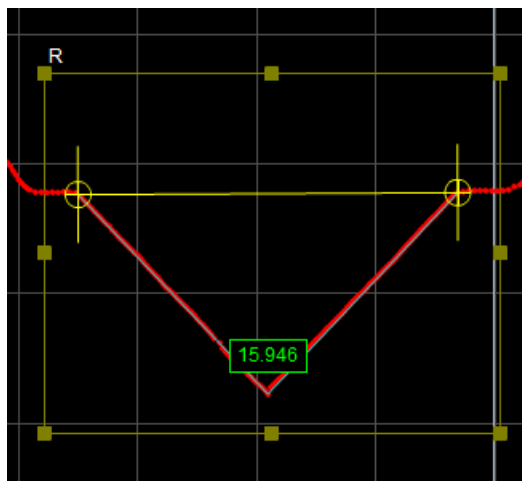


For more information on anchoring, see *Measurement Anchoring* on page 131.

Groove

The Groove tool provides measurements of V-shape, U-shape, or open-shape grooves. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameter Anchoring

Source: Top

Shape: U-Shape

Min Depth: 0 mm

Min Width: 0 mm

Max Width: 0 mm

☒ Region

Width 15.946

X

Z

Width 15.946

Depth

Id: 6

Parameters Output

Select Type: Max Depth

Index: 0

The Groove tool uses a complex feature-locating algorithm to find a groove and then return measurements. See "Groove 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 Groove tool lets you add multiple measurements of the same type to receive measurements and set decisions for multiple grooves. Multiple measurements are added by using the drop-down above the list of measurements and clicking on the **Add** button.

For example, if a target has three grooves, by adding two measurements, choosing **Index From The Left** in the **Select Type** setting of those measurements, and providing values of 0 and 2 in the **Index** setting of the measurements, respectively, the Groove tool will return measurements and decisions for the first and third grooves.

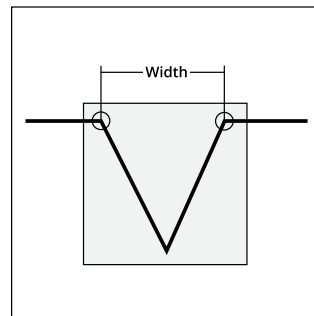
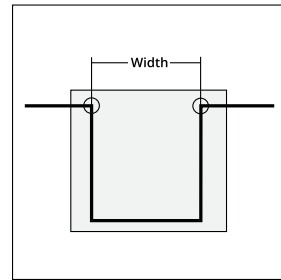
Measurements

Measurement

Illustration

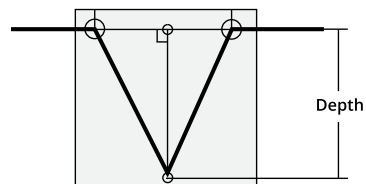
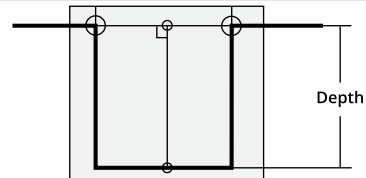
Width

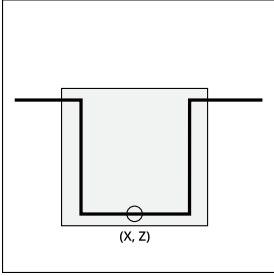
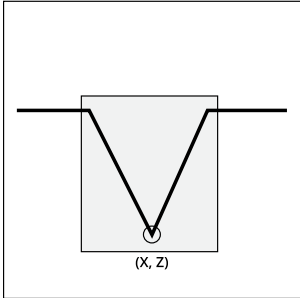
Measures the width of a groove.



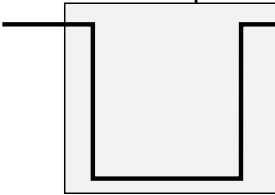
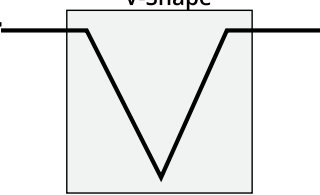
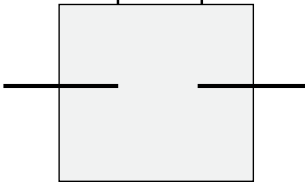
Depth

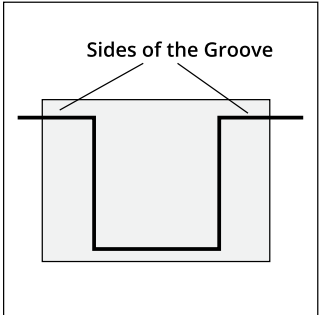
Measures the depth of a groove as the maximum perpendicular distance from a line connecting the edge points of the groove.



Measurement	Illustration
X Measures the X position of the bottom of a groove.	
Z Measures the Z position of the bottom of a groove.	

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Shape	Shape of the groove U-Shape  V-Shape  Open Shape 
Min Depth	Minimum depth for a groove to be considered valid.
Min Width	Minimum width for a groove to be considered valid. The width is the distance between the groove corners.
Max Width	Maximum width of a groove to be considered valid. If set to 0, the maximum is set to the width of the measurement area.

Parameter	Description
Region	The measurement region defines the region in which to search for the groove. For a stable measurement, the measurement region should be made large enough to cover some laser data on the left and right sides of the groove.  For more information on regions, see <i>Regions</i> on page 122.
Location (Groove X and Groove Z measurements only)	Specifies the location type to return Bottom - Groove bottom. For a U-shape and open-shape groove, the X position is at the centroid of the groove. For a V-shape groove, the X position is at the intersection of lines fitted to the left and right sides of the groove. See algorithm section below for more details. Left - Groove's left corner. Right - Groove's right corner.
Select Type	Specifies how a groove is selected when there are multiple grooves within the measurement area. Maximum Depth - Groove with maximum depth. Index from The Left - 0-based groove index, counting from left to right Index from the Right - 0-based groove index, counting from right to left.
Index	0-based groove index.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

 A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

 For more information on anchoring, see *Measurement Anchoring* on page 131.

Intersect

The Intersect tool determines intersect points and angles. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

The Intersect tool's measurements require two fit lines, one of which is a reference line set to the X axis (z = 0), the Z axis (x = 0), or a user-defined line.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



ParametersAnchoring

Source:TopReference Type:LineLine (L)2 RegionsRef Line (RL)2 Regions

MeasurementsFeatures

X-0.291ZAngleID:3

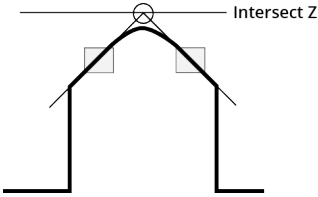
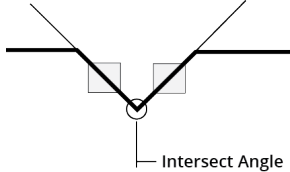
OutputFiltersDecision

Min:-3 mmMax:-2 mm

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Finds the intersection between two fitted lines and measures the X axis position of the intersection point.	

Measurement	Illustration
Z Finds the intersection between two fitted lines and measures the Z axis position of the intersection point.	
Angle Finds the angle subtended by two fitted lines.	

Features

Type	Description
Intersect Point	The point of intersection.
Line	The intersect line.
Base Line	The base line.

 For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Reference Type	Determines the type of the reference line. X-Axis: The reference line is set to the X axis. Z-Axis: The reference line is set to the Z axis Line: The reference line is defined manually using the Ref Line parameter. One or two regions can be used to define the line.
Line	You can use one or two fit areas for the fit line. To set the region (or regions) of the fit line, adjust it graphically in the data viewer, or expand the feature using the expand button () and enter the values in the fields. For more information on regions, see <i>Regions</i> on page 122. For more information on fit lines, see <i>Fit Lines</i> on page 127.

Parameter	Description
Ref Line	Used to define the reference line when Line is selected in the Reference Type parameter. To set the region (or regions) of the reference line, adjust it graphically in the data viewer, or expand the feature using the expand button (⋮) and enter the values in the fields. For more information on regions, see <i>Regions</i> on page 122. For more information on fit lines, see <i>Fit Lines</i> on page 127.
Angle Range (Angle measurement only)	Determines the angle range. The options are: -90 – 90 0 – 180
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

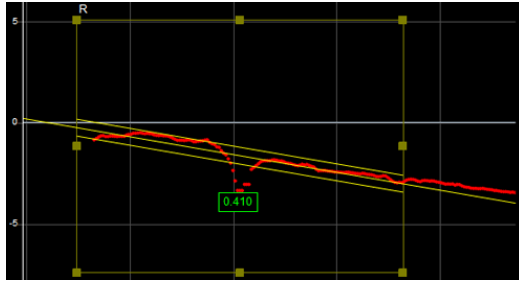


For more information on anchoring, see *Measurement Anchoring* on page 131.

Line

The Line tool fits a line to the live profile and measures the deviations from the best-fitted line. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameters
Anchoring

Source: Top

☒ Region

X: -25.63 mm

Z: -22.355 mm

Width: 52.404 mm

Height: 52.404 mm

☐ Fitting Regions All Data

Measurements
Features

Standard Deviation 0.410

Min Error

Max Error

Percentile

Offset

Angle

Min Error X

Min Error Z

Max Error X

Max Error Z

ID: 4

Output

Filters

Decision

Min: 0 mm

Max: 1 mm

Measurements, Features, and Settings

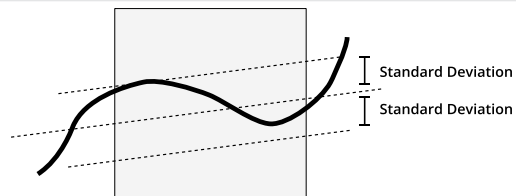
Measurements

Measurement

Standard Deviation

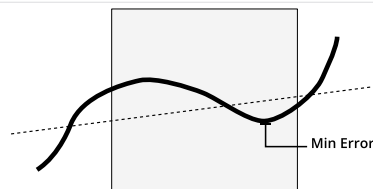
Finds the best-fitted line and measures the standard deviation of the laser points from the line.

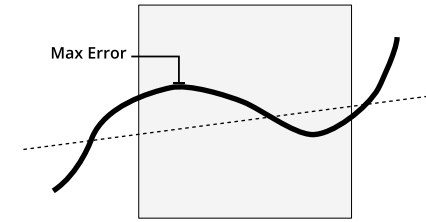
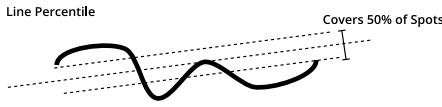
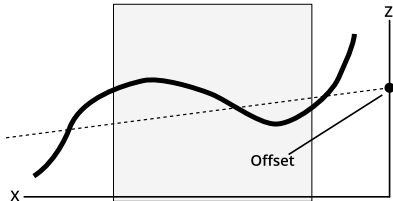
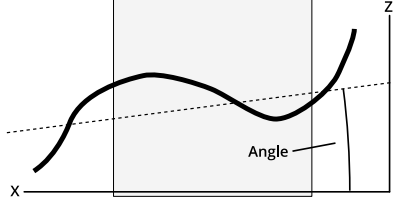
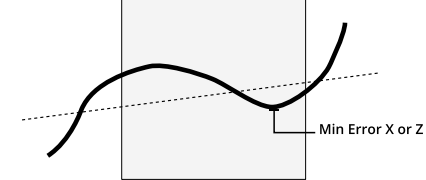
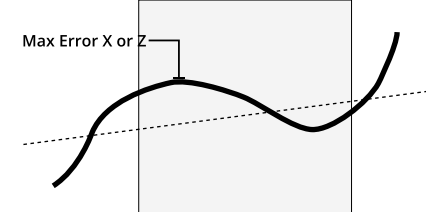
Illustration



Min Error

Finds the best-fitted line and measures the minimum error from the line (the maximum distance below the line).



Measurement	Illustration
Max Error Finds the best-fitted line and measures the maximum error from the line (the maximum distance above the line).	
Percentile Finds the best-fitted line and measures the range (in Z) that covers a percentage of points around the line.	
Offset Finds the best-fitted line and returns the intersection point between that line and the Z axis.	
Angle Finds the best-fitted line and returns the angle relative to the X axis.	
Min Error X Min Error Z Finds the best-fitted line and returns the X or Z position of the minimum error from the line (the maximum distance below the line).	
Max Error X Max Error Z Finds the best-fitted line and returns the X or Z position of the maximum error from the line (the maximum distance above the line).	
Features	
Type	Description
Line	The fitted line.
Error Min Point	The point of minimum error.
Error Max Point	The point of maximum error.
 For more information on geometric features, see <i>Geometric Features</i> on page 128.	

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.
Fitting Regions	Determines which data Gocator uses to <i>fit the line</i> over the profile. When Fitting Regions is enabled, Gocator uses the data indicated by one of the following options: • All Data: All of the data in the profile is used to fit the line. • 1 Region: Data from a fitting region you define in the data viewer is used to fit the line. • 2 Regions: Data from two fitting regions you define is used to fit the line. When Fitting Regions is disabled, to fit the line, Gocator uses the measurement region if Region is enabled, or the entire profile if Region is disabled. When Fitting Regions is enabled and 1 Region or 2 Regions is selected, you can set the region (or regions) graphically in the data viewer, or you can expand the feature using the expand button (⋮) and enter the values in the fields. For more information on regions, see <i>Regions</i> on page 122.
Percent (Percentile measurement only)	The specified percentage of points around the best-fitted line.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

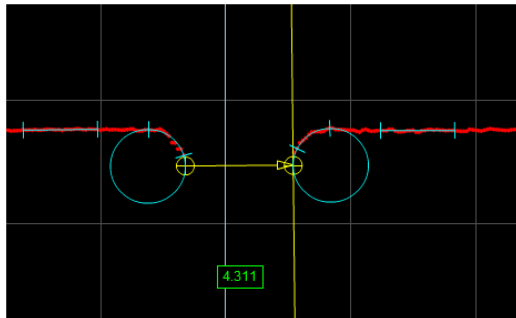


For more information on anchoring, see *Measurement Anchoring* on page 131.

Panel

The Panel tool provides Gap and Flush measurements. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameters Anchoring

Source: Top

Reference Side: Left

Max Gap Width: 0 mm

Left

Right

Gap 4.311 ☒

Flush ☐

Left Gap X ☐

Left Gap Z ☐

Left Flush X ☐

Left Flush Z ☐

Left Surface Angle ☐

Right Gap X ☐

Right Gap Z ☐

Right Flush X ☐

Right Flush Z ☐

Right Surface Angle ☐

ID: 0

Parameters Output

Measurement Axis Edge

The Panel tool uses a complex feature-locating algorithm to find the gap or calculate flushness and return measurements. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel. See "Gap and Flush Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed explanation of the algorithm.



You must make sure that there are enough data points to define the edge in the profile, by properly setting up exposure, etc. If not, the algorithm will not function.

Measurement	Illustration
Gap Measures the distance between two surfaces. The surface edges can be curved or sharp.	
Flush Measures the flushness between two surfaces. The surface edges can be curved or sharp.	
Left Gap X Returns the X position of the edge feature on the left side used to measure the gap.	
Left Gap Z Returns the Z position of the edge feature on the left side used to measure the gap.	
Left Flush X Returns the X position of the feature on the left side used to measure flushness.	
Left Flush Z Returns the Z position of the feature on the left side used to measure flushness.	
Left Surface Angle The angle of the left side surface relative to the X axis.	

Measurement	Illustration
Right Gap X Returns the X position of the edge feature on the right side used to measure the gap.	
Right Gap Z Returns the Z position of the edge feature on the right side used to measure the gap.	
Right Flush X Returns the X position of the feature on the right side used to measure flushness.	
Right Flush Z Returns the Z position of the feature on the right side used to measure flushness.	
Right Surface Angle The angle of the right side surface relative to the X axis.	

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Reference SideDirection	Defines the side used to calculate the measurement axis (see below) rounded corner.
Max Gap Width	The maximum width of the gap. Allows the tool to filter gaps greater than the expected width. This can be used to single out the correct gap when there are multiple gaps in the field of view.
Measurement Axis <i>Gap measurement only</i>	Defines the direction that the gap is calculated, in relation to the reference side (see above). Surface: In the direction of the fitted surface line of the reference surface. Edge: In the direction perpendicular to the edge of the reference surface. Distance: The Cartesian distance between the two feature locations.
Absolute <i>Flush measurement only</i>	When enabled, returns an absolute value rather than a signed value.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Left/Right SideEdge Parameters

Parameter	Description
Max Void Width	The maximum allowed width of missing data caused by occlusion or data dropout.
Min Depth	Defines the minimum depth before an opening could be considered to have a potential edge. The depth is the perpendicular distance from the fitted surface line.
Surface Width	The width of the surface area in which laser data is used to form the fitted surface line. This value should be as large as the surface allows.
Surface Offset	The distance between the edge region and the surface region. Setting a small value allows the edge within a tighter region to be detected. However, the measurement repeatability could be affected if the data from the edge are considered as part of the surface region (or vice versa). A rule of thumb is to set Surface Offset equal to Nominal Radius .
Nominal Radius	The radius of the curve edge that the tool uses to locate the edge region.
Edge Angle	A point on the best fit circle to be used to calculate the feature point. The selected point is on the circumference at the specified angle from the start of the edge region. The angle is measured from the axis perpendicular to the fitted surface line.
Edge Type	Defines the type of feature point to use for the edge (Corner or Tangent). A tangent edge point is the point selected based on the defined Edge Angle. A corner edge point is the intersect point between the fitted surface line and a edge line formed by interpolating the points at and after the tangent within the edge region.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.



For more information on anchoring, see *Measurement Anchoring* on page 131.

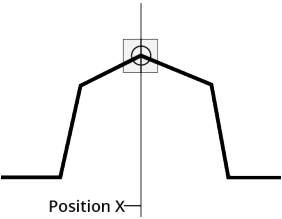
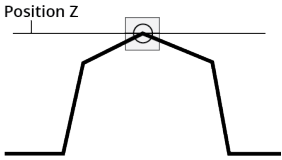
Position

The Position tool finds the X or Z axis position of a feature point. The feature type must be specified and is one of the following: Max Z, Min Z, Max X, Min X, Corner, Average (the mean X and Z of the data points), Rising Edge, Falling Edge, Any Edge, Top Corner, Bottom Corner, Left Corner, Right Corner, or Median (median X and Z of the data points). Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.

Measurements, Features, and Settings

Measurements

Measurement	Illustration
X Finds the position of a feature on the X axis.	
Z Finds the position of a feature on the Z axis.	

Features

Type	Description
Point	The returned position.

 For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.

Parameter	Description
Feature	The feature the tool uses for its measurements. One of the following: • Max Z • Min Z • Max X • Min X • Corner • Average • Rising Edge • Falling Edge • Any Edge • Top Corner • Bottom Corner • Left Corner • Right Corner • Median To set the region of a feature, adjust it graphically in the data viewer, or expand the feature using the expand button () and enter the values in the fields. For more information on regions, see <i>Regions</i> on page 122.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.



A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.



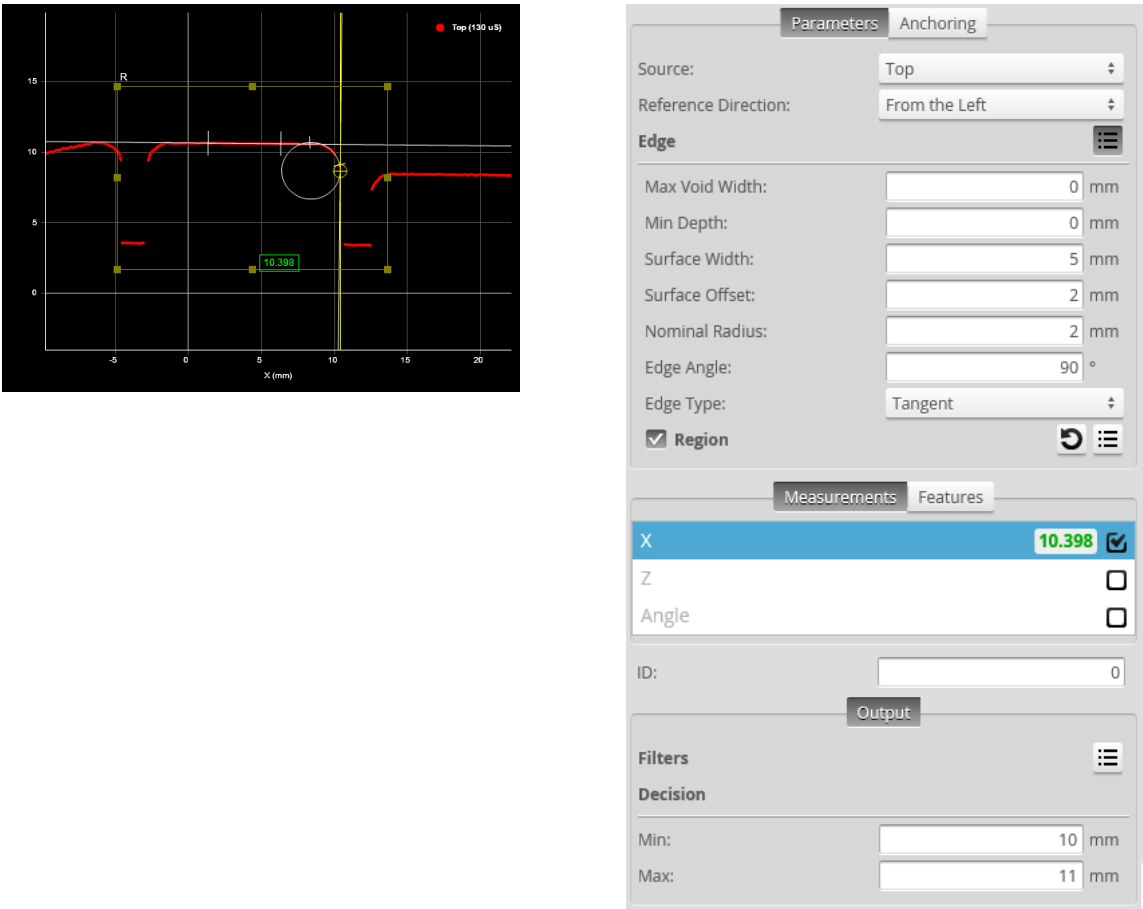
For more information on anchoring, see *Measurement Anchoring* on page 131.

Round Corner

The Round Corner tool measures corners with a radius, returning the position of the edge of the corner and the angle of adjacent surface with respect to the X axis.

Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.

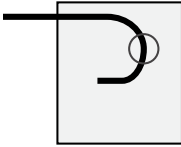


The Round Corner tool uses a complex feature-locating algorithm to find the edge and return measurements. The behavior of the algorithm can be adjusted by changing the parameters in the measurement panel. See "Gap and Flush Algorithm" in the *Gocator Measurement Tool Technical Manual* for a detailed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



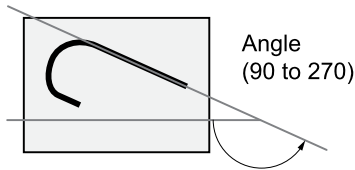
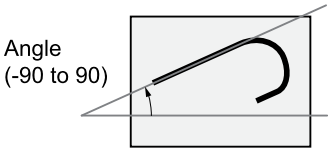
You must make sure that there are enough data points to define the edge (proper exposure, etc.). If not, the algorithm will not function.

Measurements

Measurement	Illustration
X Measures the X position of the location where the tangent touches the edge, or intersect of the tangent and the line fitted to the surface used by the measurement (see Reference Side , below).	
Z Measures the Z position of the location where the tangent touches the edge, or intersect of the tangent and the line fitted to the surface used by the measurement (see Reference Side , below).	

Angle

Measures the angle of the line fitted to the surface next to the corner (see **Reference Side**, below), with respect to the x-axis. Left edge angles are from -90 to 90. Right edge angles are from 90 to 270.



Features

Type	Description
Edge Point	The position of the edge.
Radius Center Point	The center of the radius.

 For more information on geometric features, see *Geometric Features* on page 128.

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.
Reference Direction	Defines the side used to calculate the rounded corner.
Max Gap Width	The maximum width of the gap. Allows the tool to filter gaps greater than the expected width. This can be used to single out the correct gap when there are multiple gaps in the field of view.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Edge Parameters

Parameter	Description
Max Void Width	The maximum allowed width of missing data caused by occlusion or data dropout.
Min Depth	Defines the minimum depth before an opening could be considered to have a potential edge. The depth is the perpendicular distance from the fitted surface line.
Surface Width	The width of the surface area in which laser data is used to form the fitted surface line. This value should be as large as the surface allows.
Surface Offset	The distance between the edge region and the surface region. Setting a small value allows the edge within a tighter region to be detected. However, the measurement repeatability could be affected if the data from the edge are considered as part of the surface region (or vice versa). A rule of thumb is to set Surface Offset equal to Nominal Radius.
Nominal Radius	The radius of the curve edge that the tool uses to locate the edge region.
Edge Angle	A point on the best fit circle to be used to calculate the feature point. The selected point is on the circumference at the specified angle from the start of the edge region. The angle is measured from the axis perpendicular to the fitted surface line.
Edge Type	Defines the type of feature point to use for the edge (Corner or Tangent). A tangent edge point is the point selected based on the defined Edge Angle. A corner edge point is the intersect point between the fitted surface line and a edge line formed by interpolating the points at and after the tangent within the edge region.
Region	The region to which the tool's measurements will apply. For more information, see <i>Regions</i> on page 122.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

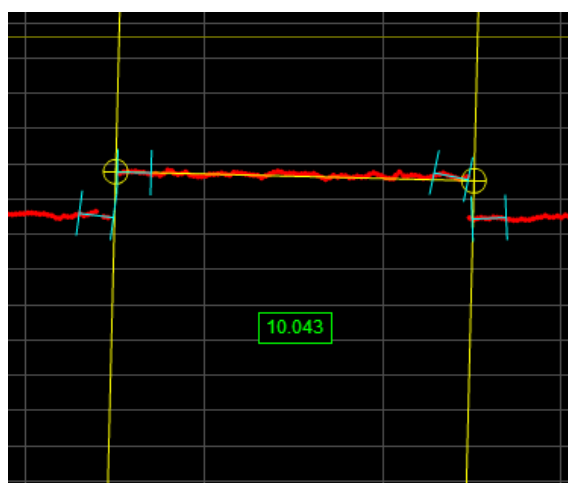
ⓘ A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

ⓘ For more information on anchoring, see *Measurement Anchoring* on page 131.

Strip

The Strip tool measures the width of a strip. Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Parameters Anchoring

Source: Top

Base Type: Flat

Left Edge

☒ Rising ☒ Data End

☒ Falling ☒ Void

Right Edge

Tilt Enabled: ☒

Support Width: 5 mm

Transition Width: 0 mm

Min Width: 0 mm

Min Height: 2 mm

Max Void Width: 0 mm

☒ Region

X Add

X ☐ *

Z ☐ *

Width 10.043 ☒ *

Height ☐ *

Id: 12

Parameters Output

Select Type: Index Left

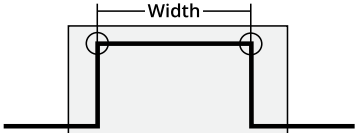
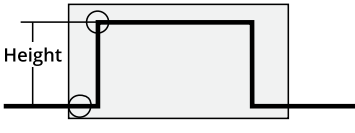
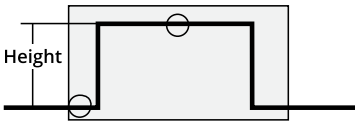
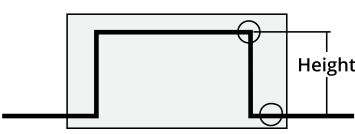
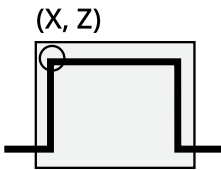
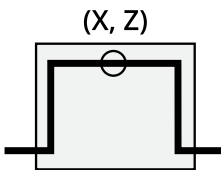
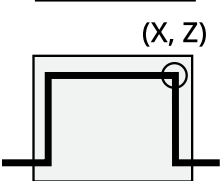
Index: 0

The Strip tool uses a complex feature-locating algorithm to find a strip and then return measurements. See "Strip 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 Strip tool lets you add multiple measurements of the same type to receive measurements and set decisions for multiple strips. Multiple measurements are added by using the drop-down above the list of measurements and clicking on the **Add** button.

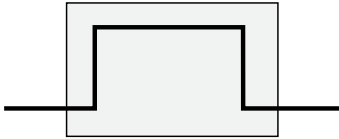
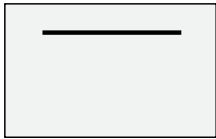
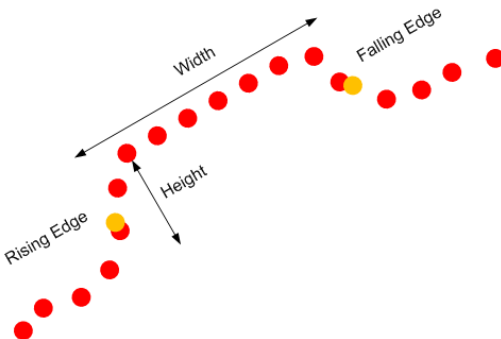
For example, if a target has three strips, by adding two measurements, choosing **Index From The Left** in the **Select Type** setting, and providing values of 1 and 3 in the **Index** of field of the measurements, respectively, the Strip tool will return measurements and decisions for the first and third strip.

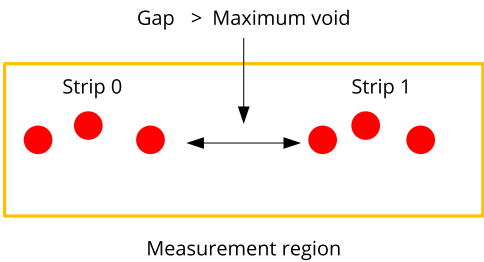
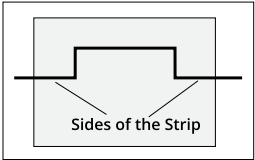
Measurements

Measurement	Illustration
Width Measures the width of a strip.	
Height Measures the height of a strip.	  
X Measures the X position of a strip.	
Z Measures the Z position of a strip.	 

Parameters

Parameter	Description
Source	The sensor, or combination of sensors, that provides data for the tool's measurements. For more information, see <i>Source</i> on page 122.

Parameter	Description
Base Type	Affects detection of rising and falling edges. Base Type = Flat  Base Type = None  When Base Type is set to Flat, both strip (raised area) and base support regions are needed. When set to None, only a point that deviates from a smooth strip support region is needed to find a rising or falling edge.
Left Edge Right Edge	Specifies the features that will be considered as the strip's left and right edges. You can select more than one condition. Rising - Rising edge detected based on the strip edge parameters. Falling - Falling edge detected based on the strip edge parameters. Data end - First valid profile data point in the measurement region. Void - Gap in the data that is larger than the maximum void threshold. Gaps connected to the measurement region's boundary are not considered as a void. See "Strip Start and Terminate Conditions" in the <i>Gocator Measurement Tool Technical Manual</i> for the definitions of these conditions.
Tilt Enabled	Enables/disables tilt correction. The strip may be tilted with respect to the sensor's coordinate X axis. This can be caused by conveyor vibration. If the Tilt option is enabled, the tool will report the width and height measurements following the tilt angle of the strip. 
Support Width	Specifies the width of the region around the edges from which the data is used to calculate the step change. See "Strip Step Edge Definitions" in the <i>Gocator Measurement Tool Technical Manual</i> on how this parameter is used by different base types.

Parameter	Description
Transition Width	Specifies the nominal width needed to make the transition from the base to the strip. See "Strip Step Edge Definitions" in the <i>Gocator Measurement Tool Technical Manual</i> on how this parameter is used by different base types.
Min Width	Specifies the minimum width for a strip to be considered valid.
Min Height	Specifies the minimum deviation from the strip base. See "Strip Step Edge Definitions" in the <i>Gocator Measurement Tool Technical Manual</i> on how this parameter is used for different base types.
Max Void Width	The maximum width of missing data allowed for the data to be considered as part of a strip when Void is selected in the Left or Right parameter. This value must be smaller than the edge Support Width.  Measurement region
Region	The measurement region defines the region in which to search for the strip. If possible, the region should be made large enough to cover the base on the left and right sides of the strip.  For more information, see <i>Regions</i> on page 122.
Location (Strip Height, Strip X, and Strip Z measurements only)	Left - Left edge of the strip. Right - Right edge of the strip. Center - Center of the strip.
Select Type	Specifies how a strip is selected when there are multiple strips within the measurement area. Best - The widest strip. Index Left - 0-based strip index, counting from left to right. Index Right - 0-based strip index, counting from right to left.
Index	0-based strip index.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Anchoring

Anchor	Description
X or Z	Lets you choose the X or Z measurement of another tool to use as a positional anchor for this tool.

 A measurement *must* be enabled in the other tool for it to be available as an anchor. The anchor measurement should also be properly configured before using it as an anchor.

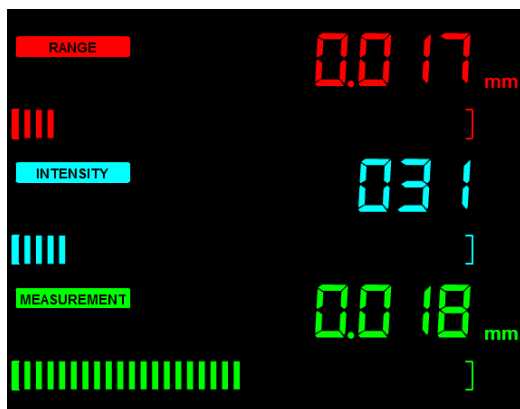
 For more information on anchoring, see *Measurement Anchoring* on page 131.

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 Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.

See *Scripts* on page 183 for more information on scripts.



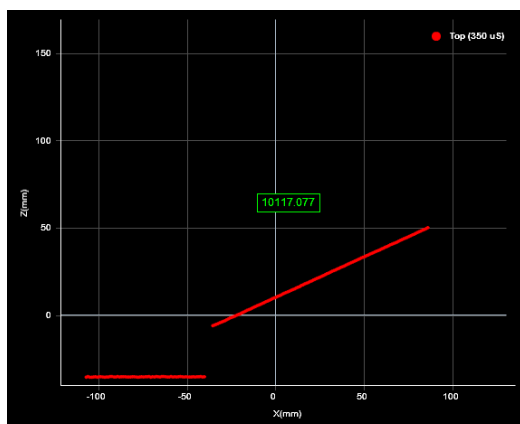
```
Code
1 double PositionZ = Measurement_Value(0);
2
3 if (Measurement_Valid(0))
4 {
5     Output_Set(PositionZ + 10000, 1);
6 }
7 else
8 {
9     Output_Set(0, 0);
10 }
11
```

*Press save button or 'Ctrl+S' to apply change.
Press 'Esc' to exit full screen.

Output: Add

Output 0 10000.018

Id: 1



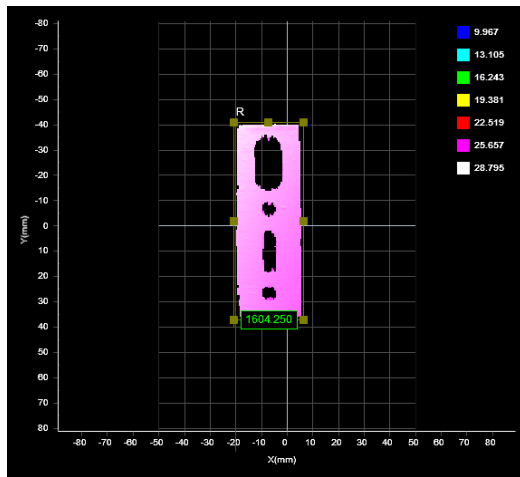
```
Code
1 double DimensionDistance = Measurement_Value(2);
2
3 if (Measurement_Valid(2))
4 {
5     Output_Set(DimensionDistance + 10000, 1);
6 }
7 else
8 {
9     Output_SetAt(0, 0);
10 }
```

*Press save button or 'Ctrl+S' to apply change.
Press 'Esc' to exit full screen.

Output: Add

Output 0 10117.077

Id: 0



Code

```

1 double VolumeArea = Measurement_Value(4);
2
3 if (Measurement_Valid(4))
4 {
5     Output_Set(VolumeArea + 10000, 1);
6 }
7 else
8 {
9     Output_SetAt(0, 0);
10 }

```

*Press save button or 'Ctrl+S' to apply change.
Press 'Esc' to exit full screen.

Output:

Add

Output 0

1604.250

X

Id:

0

See *Scripts* on page 183 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 profile3D point cloud data is not. (The script engine is not powerful enough to process the data itself.) Only one script can be created.

Feature Measurement

The following sections describe Gocator's Feature tools.

Feature tools produce measurements based on more complex geometry, letting you implement applications more quickly by reducing dependence on writing scripts to accomplish these kinds of measurements. Feature tools take [geometric features](#) generated by other tools as input and perform measurements on those features.

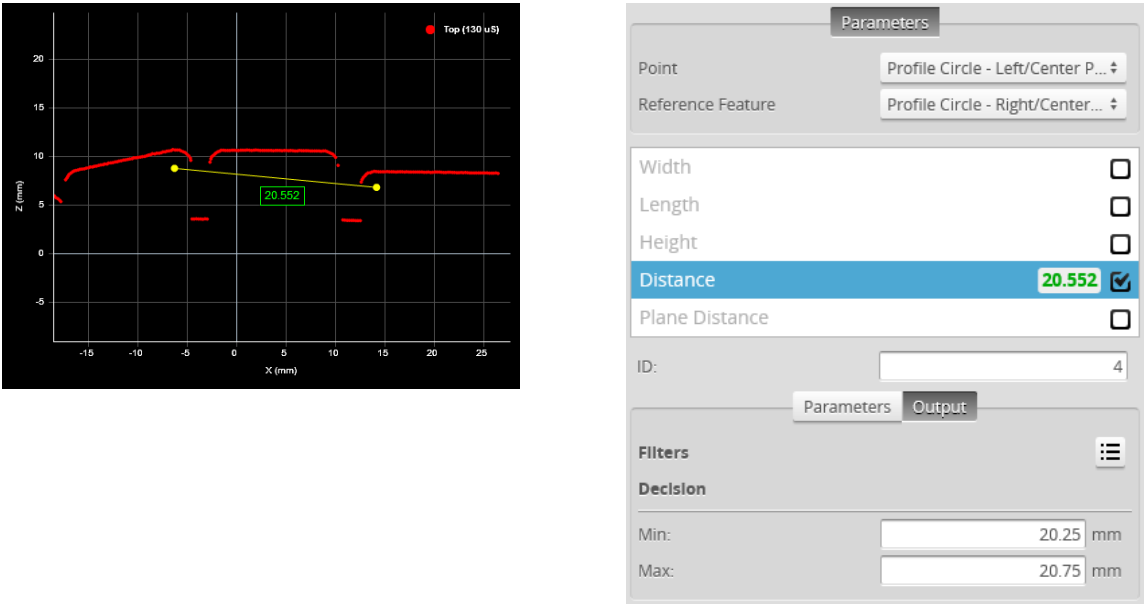
Feature tools are available in Profile mode.

Dimension

The Feature Dimension tool provides dimensional measurements from a point [geometric feature](#) to a reference point or line geometric feature.

Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Measurement Panel

 In the following measurement descriptions, the first geometric feature is set in the **Point** drop-down. The second geometric feature is set in the **Reference Feature** drop-down.

Measurements

Measurement

Width

Point-point: The difference on the X axis between the points.

Point-line: The difference on the X axis between the point and a point on the line. For profiles, the point on the line is at the same Z position as the first point.

Length

Point-point: The difference on the Y axis between the points.

Point-line: The difference on the Y axis between the point and, for profiles, the nearest point on the line; currently, always zero.

Measurement

Height

Point-point: The difference on the Z axis between the points.

Point-line: The difference on the Z axis between the point and, for profiles, a point on the line at the same X position as the first point.

Distance

Point-point: The direct, Euclidean distance between two point geometric features.

Point-line: The direct, Euclidean distance between a point and the nearest point on the line.

Plane Distance

Point-point: The distance between two point geometric features. For profile data, the points are projected onto the XZ plane (always the same as the Distance measurement).

Point-line: The distance between a point and a line. For profile data, projected onto the XZ plane (always the same as the Distance measurement).

Parameters

Parameter	Description
Point	A point geometric feature generated by another tool.
Reference Feature	A point or line geometric feature generated by another tool. Dimensional measurements are calculated <i>from</i> the reference feature <i>to</i> the point in the Point setting.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Intersect

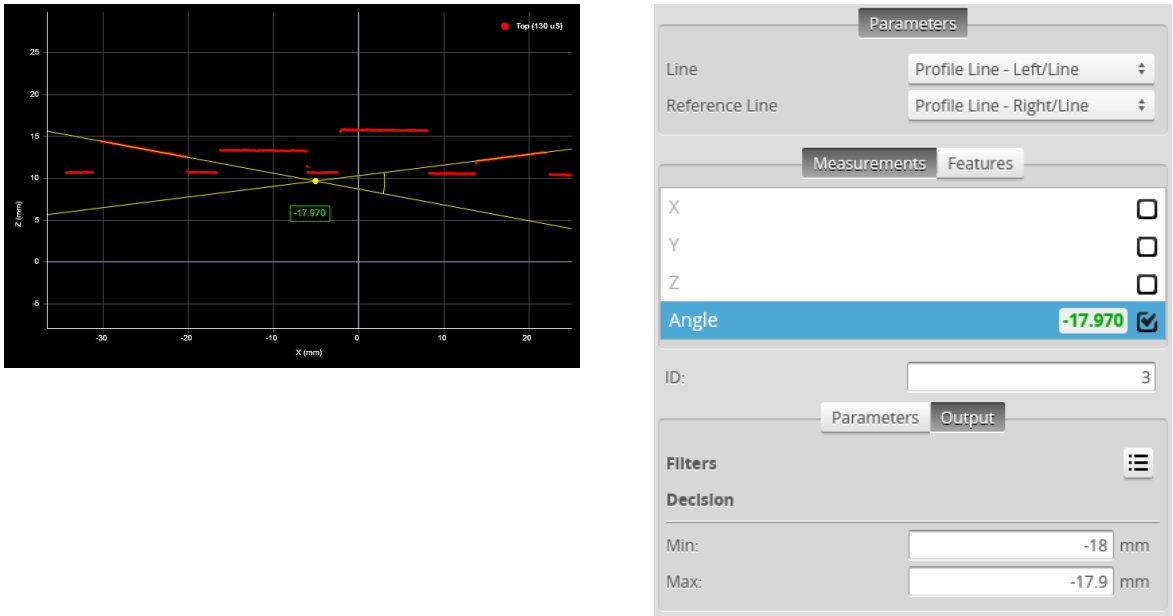
The Feature Intersect tool returns the intersection of a line [geometric features](#) and a reference line or plane geometric feature. For line-line intersections, the lines are projected onto the Y = 0 plane for features extracted from a profile. The angle measurement between the two lines is also returned.

The Feature Intersect tool saves you from having to write complicated calculations in [script tools](#) to find intersect point between lines. Previously, calculating the intercept point of two lines was difficult and prone to bugs, involving finding lines in indirect ways.

The Feature Intersect tool can also generate a point [geometric feature](#) representing the point of intersection of the lines that the [Feature Dimension](#) tool can use in measurements.

Gocator compares the measurement value with the values in **Min** and **Max** to yield a decision. For more information on decisions, see *Decisions* on page 128.

See *Adding and Configuring a Measurement Tool* on page 121 for instructions on how to add measurement tools.



Measurement Panel

Measurements

Measurement

X

Line-Line: The X position of the intersect point between the lines.

Line-Plane: The X position of the intersect point between the line and the plane.

Y

Line-Line: The Y position of the intersect point between the lines.

Line-Plane: The Y position of the intersect point between the line and the plane.

Z

Line-Line: The Z position of the intersect point between the lines.

Line-Plane: The Z position of the intersect point between the line and the plane.

Angle

Line-Line: The angle between the lines, as measured from the line selected in **Reference Feature** to the line selected in **Line**. Line-line angles can range over 360 degrees, expressed either as an angle from -180 to 180 or as an angle from 0 to 360 degrees.

*Line-Plane:*The angle between the line and the perpendicular projection of the line onto the plane, as measured from the plane geometric feature selected in **Reference Feature** to the line selected in **Line**. Line-plane angles are expressed as an angle from 90 to -90 degrees, which can be expressed as an absolute value.

In both cases, use the **Angle Range** setting to determine how angles are expressed.

Features

Type	Description
Intersect Point	The intersect point of the two edge lines.

Parameters

Parameter	Description
Line	A line geometric feature generated by another tool.
Reference Feature	A line or plane geometric feature generated by another tool. For the Angle measurement, the angle is measured <i>from</i> the reference feature.
Angle Range (Angle measurement only)	Determines the angle range.
Filters	The filters that are applied to measurement values before they are output. For more information, see <i>Filters</i> on page 130.
Decision	The Max and Min settings define the range that determines whether the measurement tool sends a pass or fail decision to the output. For more information, see <i>Decisions</i> on page 128.

Scripts

Scripts use outputs from other measurement tools to produce custom measurements.

Similar to other measurement tools, a script measurement can output multiple measurement values and decisions. Scripts are added, configured, and removed much like other measurement tools; for more information on this, see *Script* under *Profile Measurement* on page 140.

 Scripts must be less than 27,000 characters long.

Scripts use a simplified C-based syntax. 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)	Determines if a measurement exists by ID. Parameters: id – Measurement ID Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Valid(int id)	Determines if a measurement value is valid by its ID. Parameters: id - Measurement ID Returns 0 - Measurement is invalid 1 - Measurement is valid
double Measurement_Value (int id)	Gets the value of a measurement by its ID. Parameters: id - Measurement ID Returns: Value of the measurement 0 – if measurement does not exist 1 – if measurement exists
int Measurement_Decision (int id)	Gets the decision of a measurement by its ID. Parameters: ID - Measurement ID Returns: Decision of the measurement 0 – if measurement decision is false 1 – If measurement decision is true
int Measurement_NameExists(char* toolName, char* measurementName)	Determines if a measurement exist by name. Parameter: toolName – Tool name measurementName – Measurement name Returns: 0 – measurement does not exist 1 – measurement exists
int Measurement_Id (char* toolName, char* measurementName)	Gets the measurement ID by the measurement name.

Function	Description
	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:

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

Runtime Variable Functions

Function	Description
int RuntimeVariable_Count()	Returns the number of runtime variables that can be accessed. Returns: The count of runtime variables.

Function	Description
int RuntimeVariable_Get32s(int id)	Returns the value of the runtime variable at the given index. Parameters: Id – ID of the runtime variable Returns: Runtime variable value

Stamp Functions

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

Math Functions

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

Example: Accumulated Length

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

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

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

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

```
Memory_Set64s(0, length);
```

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

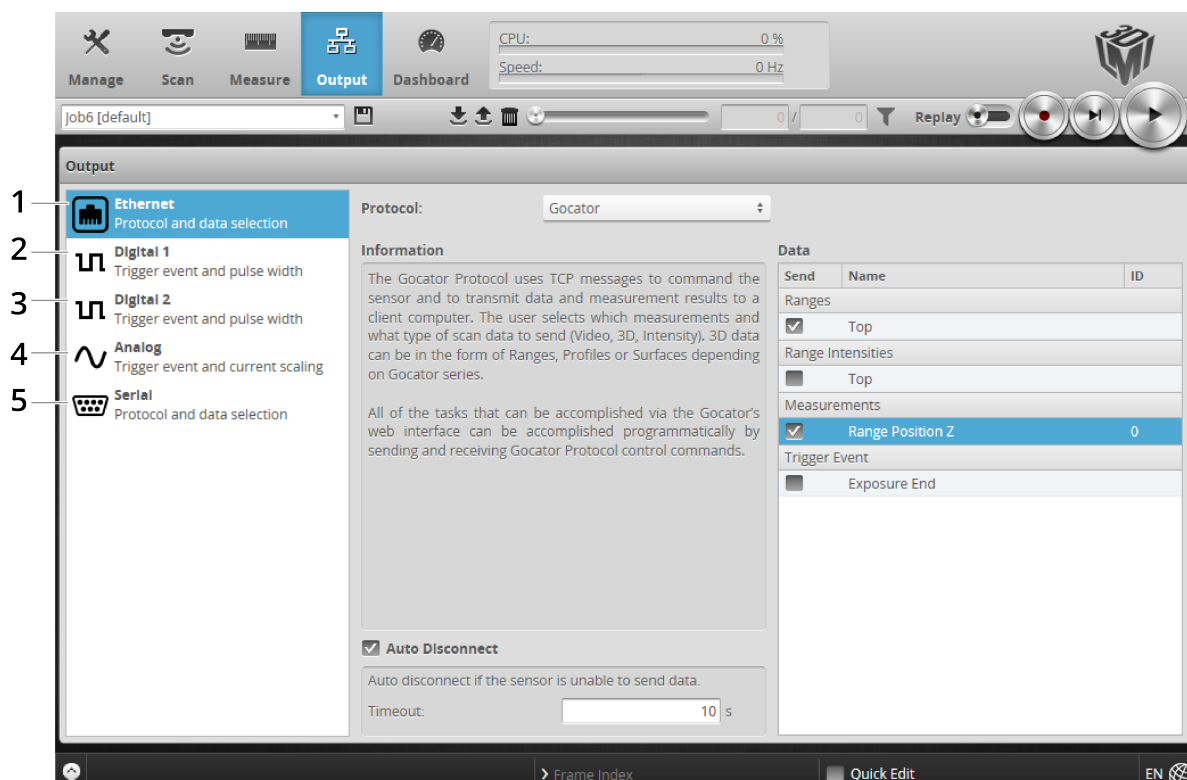

Output

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

Output Page Overview

Output configuration tasks are performed using the **Output** page. Gocator sensors can transmit 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 <i>Ethernet Output</i> 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 <i>Digital Output</i> on page 194.
3 Digital Output 2	Used to select the data sources that will be combined to produce a digital output pulse on Output 2. See <i>Digital Output</i> on page 194.
4 Analog Panel	Used to convert a measurement value or decision into an analog output signal. See <i>Analog Output</i> on page 196.
5 Serial Panel	Used to select the measurements that will be transmitted via RS-485 serial output. See <i>Serial Output</i> on page 198.

Ethernet Output

A sensor uses TCP messages (Gocator protocol) to receive commands from client computers, and to send video, laser range, intensity, and measurement results to client computers. The sensor can also receive commands from and send measurement results to a PLC using ASCII, Modbus TCP, or EtherNet/IP protocol. See *Protocols* on page 291 for the specification of these protocols.

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

Output

Ethernet
Protocol and data selection

Digital 1
Trigger event and pulse width

Digital 2
Trigger event and pulse width

Analog
Trigger event and current scaling

Serial
Protocol and data selection

Protocol: Gocator

Information

The Gocator Protocol uses TCP messages to command the sensor and to transmit data and measurement results to a client computer. The user selects which measurements and what type of scan data to send (Video, 3D, Intensity). 3D data can be in the form of Ranges, Profiles or Surfaces depending on Gocator series.

All of the tasks that can be accomplished via the Gocator's web interface can be accomplished programmatically by sending and receiving Gocator Protocol control commands.

☒ **Auto Disconnect**

Auto disconnect if the sensor is unable to send data.

Timeout: 10 s

Data

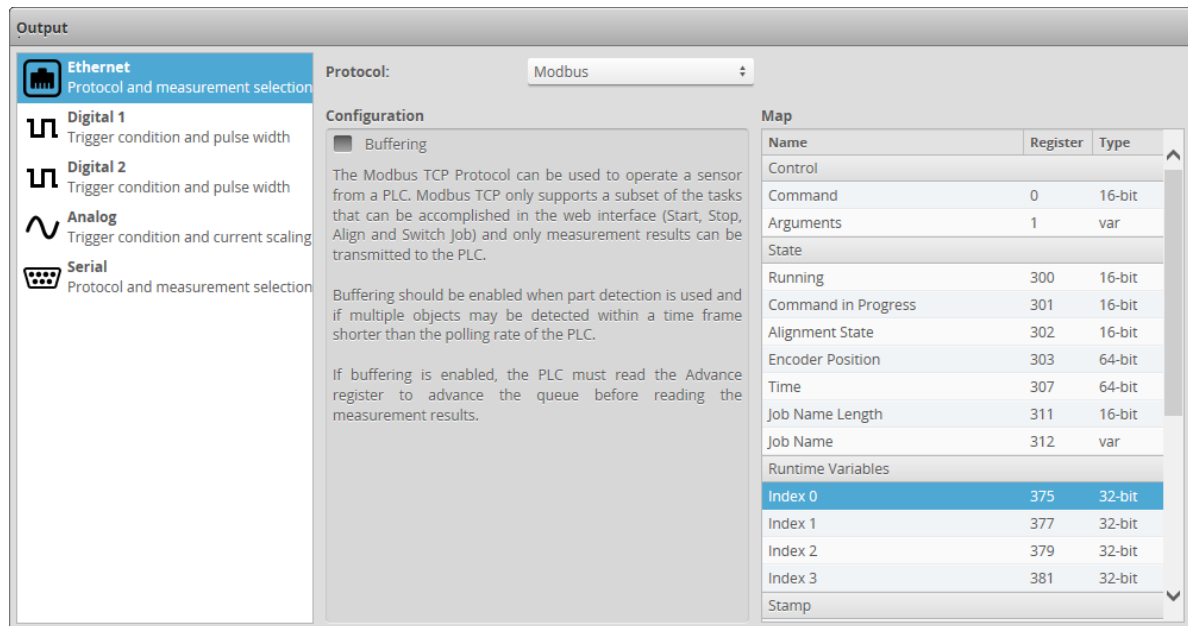
Send	Name	Id
Ranges		
☒	Top	
Measurements		
☒	Range Position Z	0
Trigger Event		
☒	Exposure End	

To receive commands and send results using Gocator Protocol messages:

1. Go to the **Output** page.
2. Click on the **Ethernet** category in the **Output** panel.
3. Select **Gocator** as the protocol in the **Protocol** drop-down.
4. Check the video, range, intensity, or measurement items to send.
5. (Optional) Uncheck the Auto Disconnect setting.
By default, this setting is checked, and the timeout is set to 10 seconds.

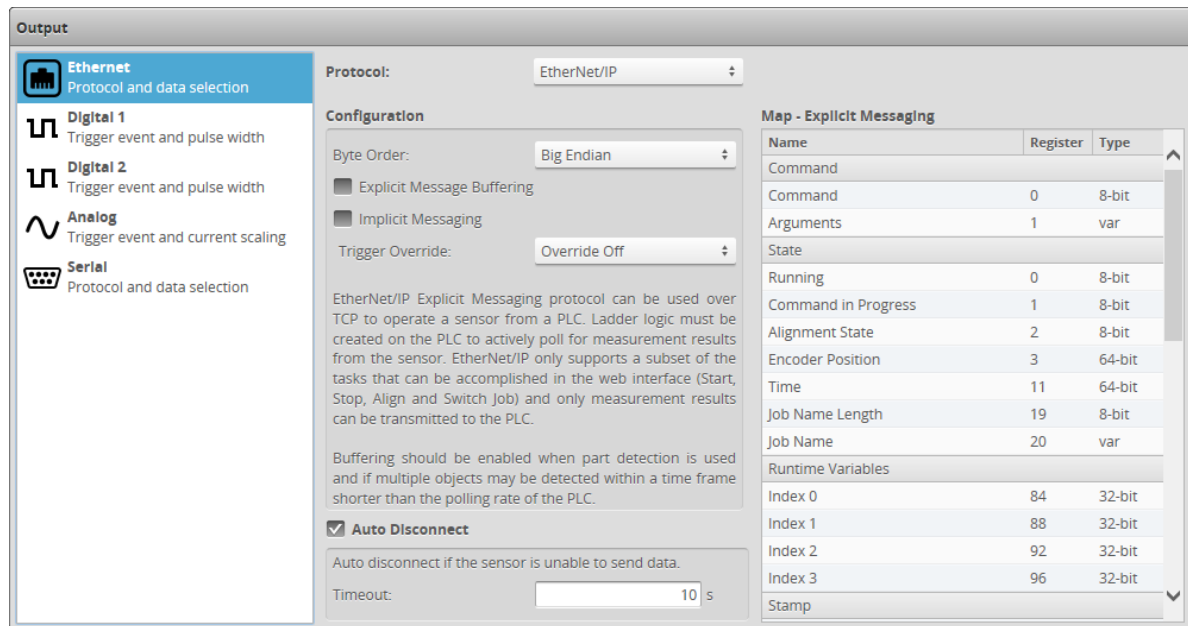
 Measurements shown here correspond to measurements that have been added using the **Measure** page (see *Measure Page Overview* on page 120).

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.



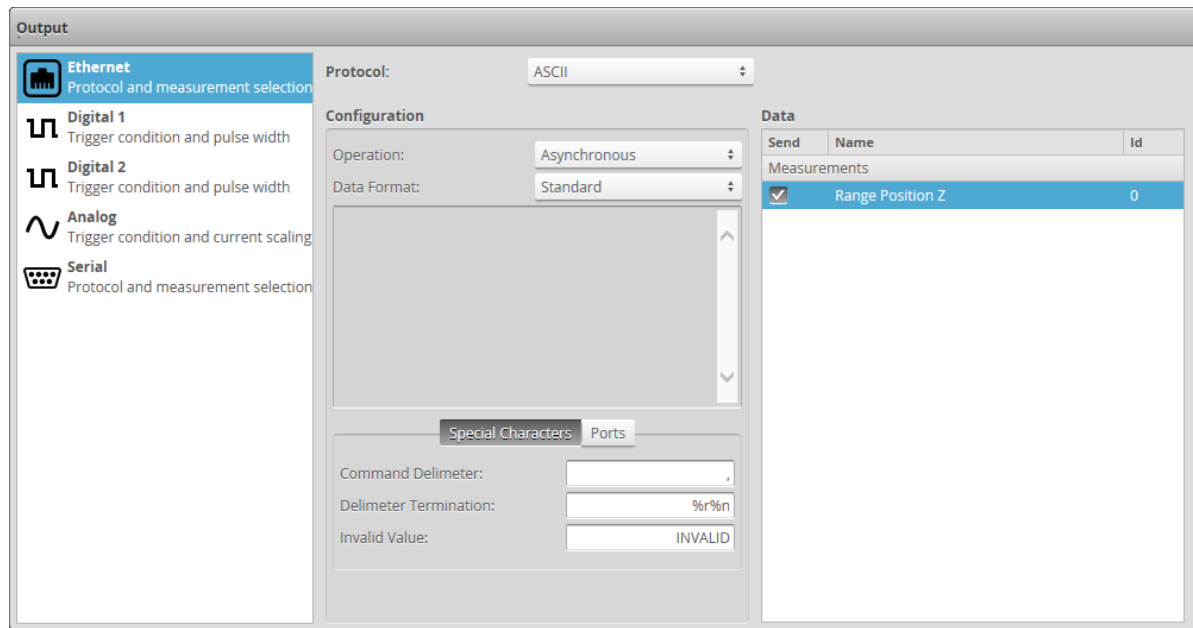
To receive commands and send results using Modbus TCP messages:

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** option, 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 is read (*Sample State Assembly* on page 360).
5. Check the **Implicit Messaging** option, if needed.
Implicit messaging uses UDP and is faster than explicit messaging, so it is intended for time-critical applications. However, implicit messaging is layered on top of UDP. UDP is connectionless and data delivery is not guaranteed. For this reason, implicit messaging is only suitable for applications where occasional data loss is acceptable.
For more information on setting up implicit messaging, see http://lmi3d.com/sites/default/files/APPNOTE_Implicit_Messaging_with_Allen-Bradley_PLCs.pdf.
6. 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 365 for an explanation of the operation modes.
5. Select the data format from the **Data Format** drop-down.
Standard: 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 373 for an explanation of the standard result mode.
Standard with Stamp: Select the measurement to send by placing a check in the corresponding checkbox. See *Standard Result Format* on page 373 for an explanation of the standard result mode.
Custom: Enables the custom format editor. Use the replacement patterns listed in **Replacement Patterns** to create a custom format in the editor. C language *printf*-style formatting is also supported: for example, `%sprintf[%09d, %value[0]]`. This allows fixed length formatting for easier input parsing in PLC and robot controller logic.
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 438 for information on wiring digital outputs to external devices.

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

Output

Ethernet
Protocol and data selection

Digital 1
Trigger event and pulse width

Digital 2
Trigger event and pulse width

Analog
Trigger event and current scaling

Serial
Protocol and data selection

Trigger Event: Measurement

Configuration

Assert On: Pass

Signal: Pulsed

100 µs

☐ Scheduled

☐ Invert Output Signal

Data

Send	Name	Id
Decisions		
☒	Range Position Z	0

To output measurement decisions:

1. Go to the **Output** page.
2. Click **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 **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. Check the **Scheduled** option if the output needs to be scheduled; otherwise, leave it unchecked for immediate output.
A scheduled output becomes active after the 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. If you checked **Scheduled**, specify a delay and a delay domain.
The **Delay** specifies the time or encoder distance 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.
9. If you want to invert the output signal, check **Invert Output Signal**.

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 **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 be scheduled. A continuous signal always becomes active immediately.

Immediate output becomes active as soon as a scheduled digital output (*Schedule Digital Output* on page 317) is received.

Scheduled output becomes active at a specific target time or position, given by the Scheduled Digital Output command. Commands that schedule an 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 **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Exposure Begin** or **Exposure End**.
4. Set the **Pulse Width** option.

The pulse width determines the duration of the digital output pulse, in microseconds.

To output an alignment signal:

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

The digital output state is High if the sensor is aligned, and Low if not aligned. Whether the sensor is running does not affect the output.

To respond to exposure begin/end:

1. Go to the **Output** page.
2. Click **Digital 1** or **Digital 2** in the **Output** panel.
3. Set **Trigger Event** to **Exposure Begin** or **Exposure End**.

Analog Output

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

sensor supports one analog output channel.



Gocator 1300 series sensors are limited to sending data at 10 kHz over the analog output channel.

Therefore, if you configure a sensor so that it runs at a speed higher than 10 kHz in the **Trigger** panel on the **Scan** page, and configure a measurement to be sent on the analog channel under **Analog** on the **Output** page, you will get analog data drops.

To achieve a 10 kHz analog output rate, you must check **Scheduled** on the **Output** page and configure scheduled output.

See *Analog Output* on page 441 for information on wiring analog output to an external device.

Send	Name	Id
None	Range Position Z	0

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 85 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 *Schedule Analog Output* on page 318) 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

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

Two protocols are supported: ASCII Protocol and Selcom Serial Protocol.

The ASCII protocol outputs data asynchronously using a single serial port. For information on the ASCII Protocol parameters and data formats, see *ASCII Protocol* on page 364.

The Selcom Serial Protocol outputs synchronized serial data using two serial ports. For information on the Selcom serial protocol and data formats, see *Selcom Protocol* on page 375.

For information on wiring serial output to an external device, see *Serial Output* on page 441.

Send	Name	Id
☒	Range Position Z	0

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 373 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 374 for the supported format string syntax.
5. Select the measurements 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.

Output

Ethernet
Protocol and data selection

Digital 1
Trigger event and pulse width

Digital 2
Trigger event and pulse width

Analog
Trigger event and current scaling

Serial
Protocol and data selection

Protocol: Selcom

Configuration

Rate: 96000

Format: SLS

Data Scale: 0 - 10000

Specifies the range of output values to use in scaling to the range selected in the Format above.
Default output units are in mm, mm², mm³ and degrees depending on the output type. Values outside this range are clamped to the minimum or maximum values.

Scheduled

Delay: 3000 µs

Data

Send	Name	Id
Measurements		
☒	Range Position Z	0

To configure Selcom output:

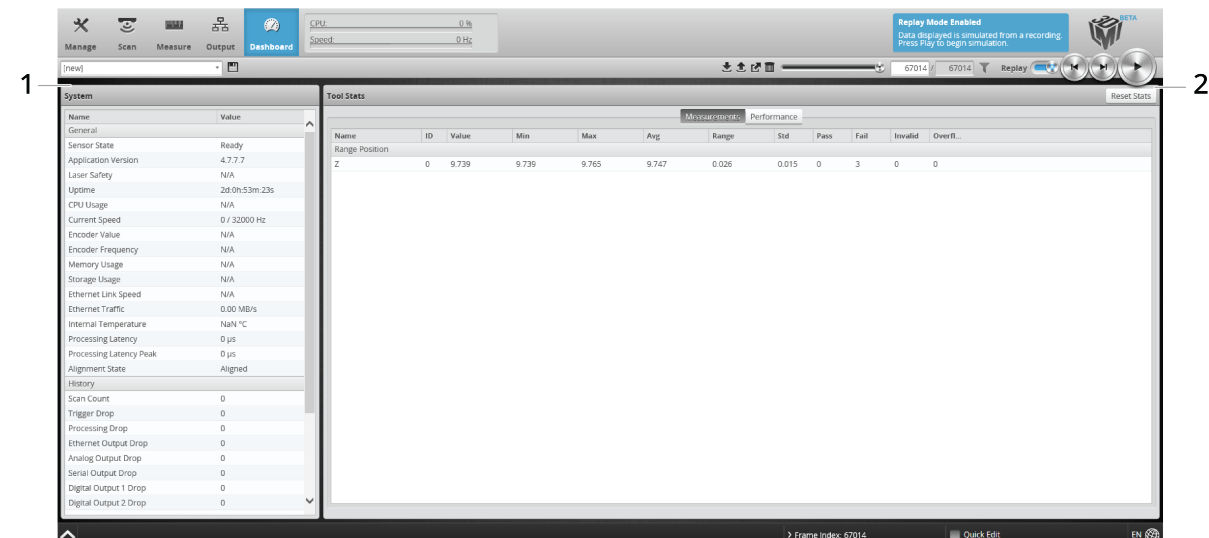
1. Go to the **Output** page.
2. Click on **Serial** in the **Output** panel.
3. Select **Selcom** in the **Protocol** option.
4. Select the measurements to send.
To select an item for transmission, place a check in the corresponding check box. Measurements shown here correspond to measurements that have been programmed using the **Measurements** page.
5. Select the baud rate in **Rate**.
6. Select the **Data Format**.
See *Selcom Protocol* on page 375 for definitions of the formats.
7. Specify **Data Scale** values.
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.
The results are scaled according to the number of serial bits used to cover the data scale range. For example, the 12-bit output would break a 200 mm data scale range into 4096 increments (0.0488 mm/bit), and the 14-bit output would break a 200 mm data scale range into 16384 increments (0.0122 mm/bit).
8. Set the output delay in **Delay**.
The scheduled delay must be longer than the processing latency to prevent drops.

Dashboard

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

Dashboard Page Overview

The **Dashboard** page summarizes sensor health information and provides measurement statistics. It also provides tool performance statistics. Use this information to troubleshoot your system.



Element	Description
1 System	Displays sensor state and health information. See <i>State and Health Information</i> below.
2 Tool Stats	Displays measurement and tool performance statistics. See <i>Statistics</i> on the next page.

State and Health Information

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 (Conflict, Ready, or Running).
Application Version	Gocator firmware version.
Laser Safety	Whether Laser Safety is enabled.
Uptime	Length of time since the sensor was power-cycled or reset.
CPU Usage	Sensor CPU utilization (%).
Current Speed*	Current speed of the sensor.
Encoder Value	Current encoder value (ticks).
Encoder Frequency	Current encoder frequency (Hz).

Name	Description
Memory Usage	Sensor memory utilization (MB used / MB total available).
Storage Usage	Sensor flash storage utilization (MB used / MB total available).
Ethernet Link Speed	Speed of the Ethernet link (Mbps).
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.
Alignment State	Whether the sensor or sensor system has been aligned.
Over Temperature State	Whether the internal temperature of the sensor is over a predetermined level.

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.
Processing Drop**	Count of frame drops due to excessive CPU utilization.
Ethernet Output Drop**	Count of frame drops due to slow Ethernet link.
Analog Output Drop**	Count of analog output drops because last output has not been completed.
Serial Output Drop**	Count of serial output drops because last output has not been completed.
Digital Output 1 Drop**	Count of digital output drops because last output has not been completed.
Digital Output 2 Drop**	Count of digital output drops because last output has not been completed.
Digital Output 1 High Count	Count of high states on digital output.
Digital Output 2 High Count	Count of high states on digital output.
Digital Output 1 Low Count	Count of low states on digital output.
Digital Output 2 Low Count	Count of low states on digital output.
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.

* When the sensor is accelerated, the indicator's value is reported from the accelerating PC.

** When the sensor is accelerated, the indicator's value is the sum of the values reported from the sensor and the accelerating PC.

Statistics

In the **Tool Stats** pane, you can examine measurement and tool statistics in two tabs: **Measurements** and **Performance**.

To reset statistics in both tabs, use the **Reset Stats** button.

Measurements

The **Measurements** tab displays statistics for each measurement enabled in the **Measure** page, grouped by the tool that contains the measurement.

Tool Stats												Reset Stats
Measurements												
Name	ID	Value	Min	Max	Avg	Range	Std	Pass	Fail	Invalid	Overflow	
Profile Area												
Centroid X	1	-13.803	-13.803	-13.803	-13.803	0	0	0	1	0	0	
Area	2	0.083	0.083	0.083	0.083	0	0	0	1	0	0	
Profile Line												
Offset	3	0.074	0.074	0.074	0.074	0	0	0	1	0	0	
Angle	4	-0.062	-0.062	-0.062	-0.062	0	0	0	1	0	0	
Profile Position												
X	5	-8.448	-8.448	-8.448	-8.448	0	0	0	1	0	0	
Z	6	0.118	0.118	0.118	0.118	0	0	0	1	0	0	

For each measurement, Gocator displays the following information:

Measurement Statistics

Name	Description
ID	The measurement ID as set in the measurement's ID field on the Measure page .
Value	The most recent measurement value.
Min	The minimum measurement value that has been observed.
Max	The maximum measurement value that has been observed.
Avg	The average of all measurement values collected since the sensor was started.
Range	The difference between Max and Min.
Std	The standard deviation of all measurement values collected since the sensor was started.
Pass	The number of pass decisions the measurement has generated.
Fail	The number of fail decisions the measurement has generated.
Invalid	The number of frames that returned no valid measurement value.
Overflow	The number of frames that returned an overflow.

Performance

The **Performance** tab displays performance statistics (execution time) for each tool added in the **Measure** page.

Tool Stats						Reset Stats
Measurements						
Name	Last (ms)	Min (ms)	Max (ms)	Avg (ms)	▼ Avg (%)	
Profile Area	0.022	0.006	0.098	0.025	21.3	
Profile Line	0.013	0.006	0.123	0.016	13.9	
Profile Position	0.006	0.004	0.046	0.007	6.1	

For each tool, Gocator displays the following information:

Performance Statistics

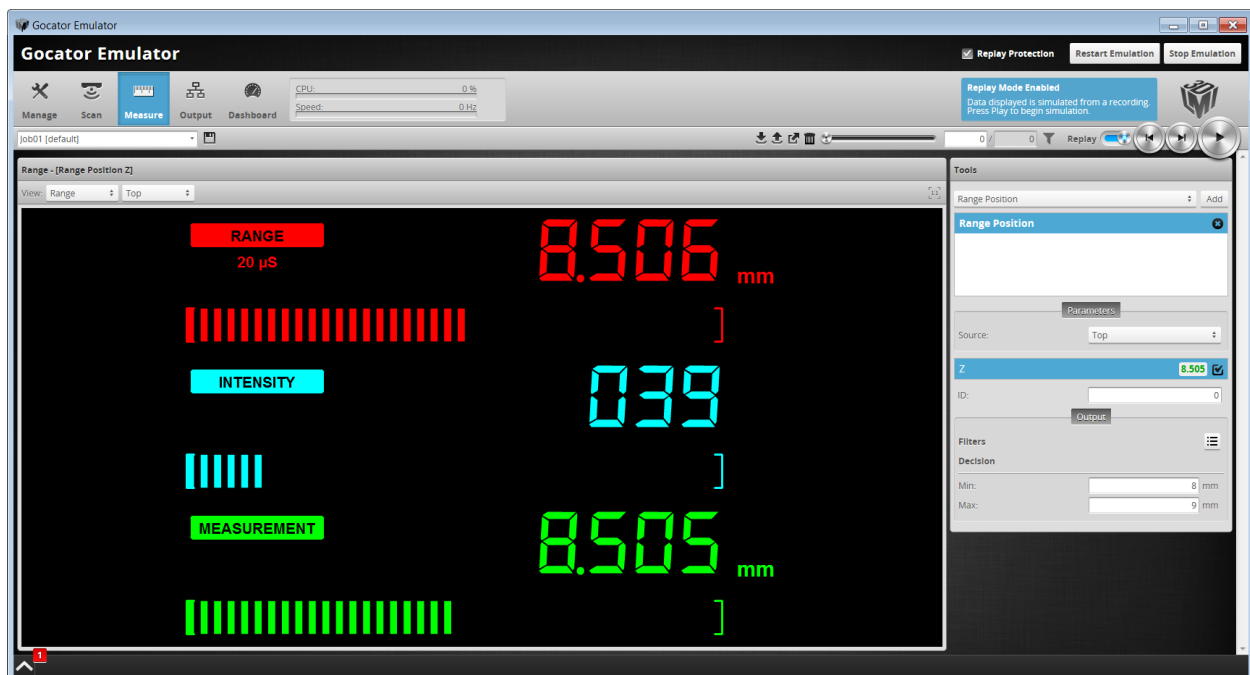
Name	Description
Last (ms)	The last execution time of the tool.
Min (ms)	The minimum execution time of the tool.
Max (ms)	The maximum execution time of the tool.
Avg (ms)	The average execution time of the tool.
Avg (%)	The average percentage the CPU the tool uses.



Tools are sorted by the *Avg (%)* column in descending order.

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.



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

System Requirements

The following are the system requirements for the software:

- Processor: Intel Core i3 or equivalent (64-bit)
- RAM: 4 GB
- Hard drive: 500 GB
- Operating system: Windows 7, 8, or 10 (64-bit)

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 a modified job by first [saving](#) it and then [downloading](#) it from the **Jobs** list on the **Manage** page to a client computer. The job file can then be loaded into the emulator at a later time or even onto a physical sensor for final testing.
- Performing alignment in the emulator has no effect and will never complete.

For information on saving and loading jobs in the emulator, see *Creating, Saving, and Loading Jobs* on page 211 .

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 215.

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 in an emulated sensor. Support files can also contain replay data, letting you test measurements and some configurations on real data. Dual-sensor systems are supported.

Support File

Download a support file which contains all jobs, data and current state of the sensor.

Filename:

Description:

Download

To download a support file:

1. Go to the **Manage** page and click on the **Support** category.
2. In **Filename**, type the name you want to use for the support file.
When you create a scenario from a support file in the emulator, the filename you provide here is displayed in the emulator's scenario list.
Support files end with the .gs extension, but you do not need to type the extension in **Filename**.
3. (Optional) In **Description**, type a description of the support file.
When you create a scenario from a support file in the emulator, the description is displayed below the emulator's scenario list.

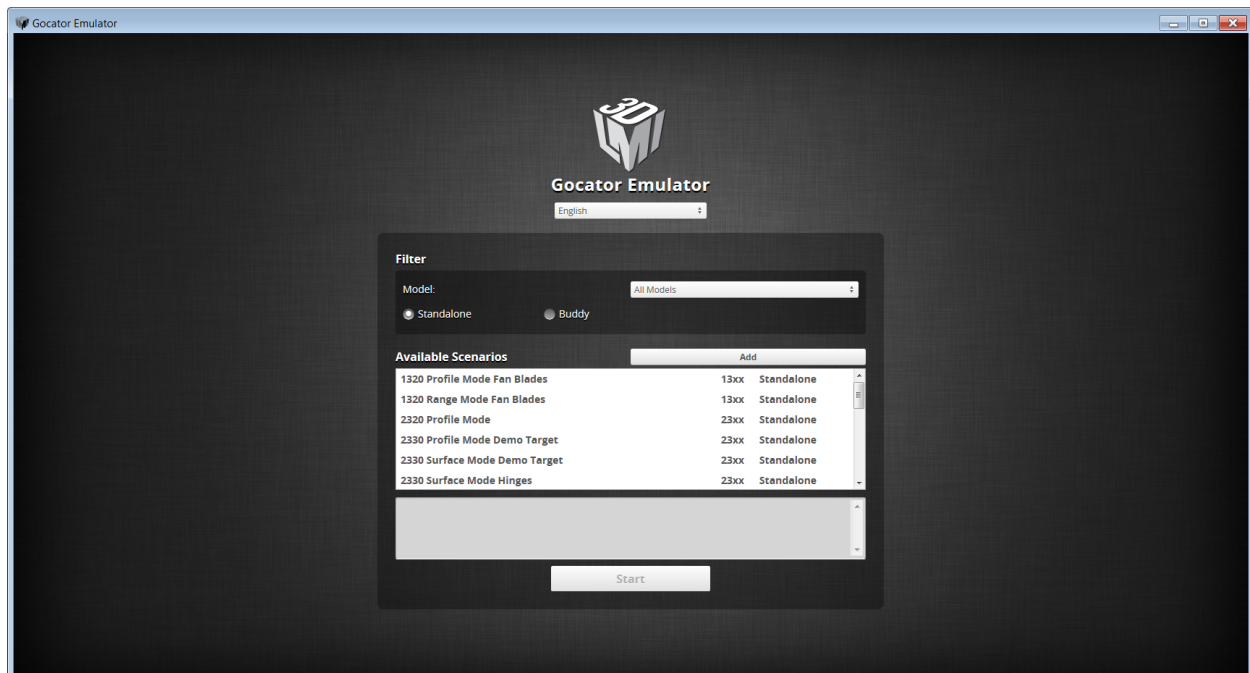
- Click **Download**, and then when prompted, click **Save**.

 Downloading a support file stops the sensor.

Running the Emulator

The emulator is contained in the Gocator tools package (14405-x.x.x.x_SOFTWARE_GO_Tools.zip). To get the package, go to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download the package from the Download Center.

To run the emulator, unzip the package and double-click the *GoEmulator* link in the unzipped emulator folder.



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

To simulate a physical sensor using a support file downloaded from a sensor, you must add it as a scenario in the emulator.

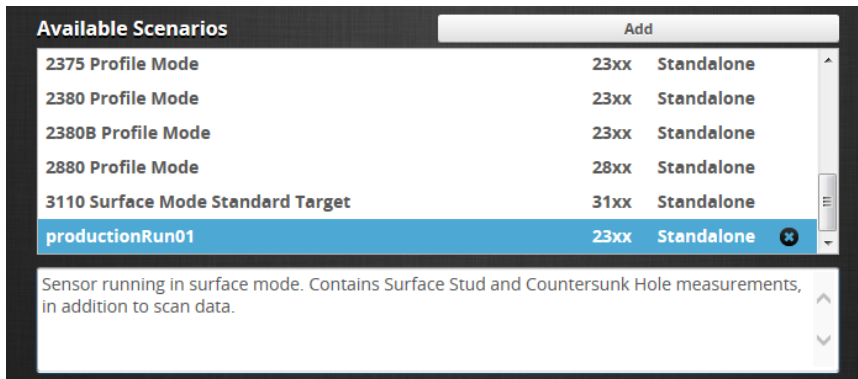
 You can add support files downloaded from any series of Gocator sensors to the emulator.

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.



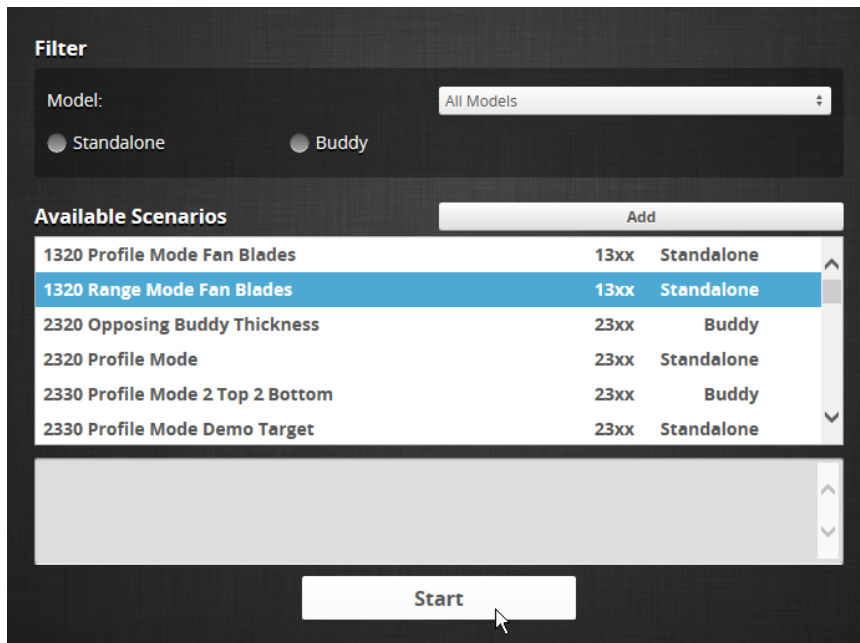
3. (Optional) In **Description**, type a description.



 You can only add descriptions for user-added scenarios.

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-/multi-sensor scenarios, respectively.
2. Select a scenario in the **Available Scenarios** list and click **Start**.

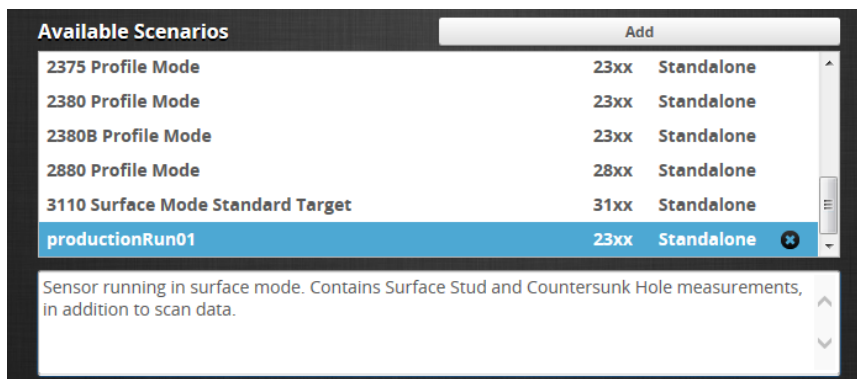
Removing a Scenario from the Emulator

You can easily remove a scenario from the emulator.

 You can only remove user-added scenarios.

To remove a scenario:

1. If the emulator is running a scenario, click **Stop Emulation** to stop it.
2. In the **Available Scenarios** list, scroll to the scenario you want to remove.

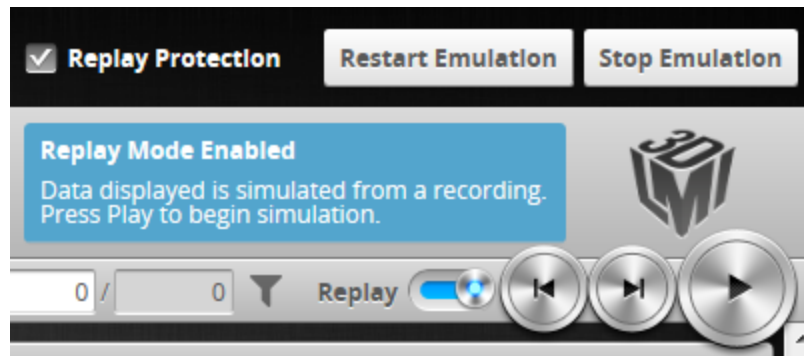


3. Click the  button next to the scenario you want to remove.

The scenario is removed from the emulator.

Using Replay Protection

Making changes to certain settings on the **Scan** page causes the emulator to flush replay data. The **Replay Protection** option protects replay data by preventing changes to settings that affect replay data. Settings that do not affect replay data can be changed.



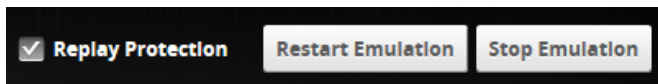
If you try to uncheck **Replay Protection**, you must confirm that you want to disable it.

Replay Protection is on by default.

Stopping and Restarting the Emulator

To stop the emulator:

- Click **Stop Emulation**.



Stopping the emulator returns you to the launch screen.

To restart the emulator when it is running:

- Click **Restart Emulation**.

Restarting the emulator restarts the currently running simulation.

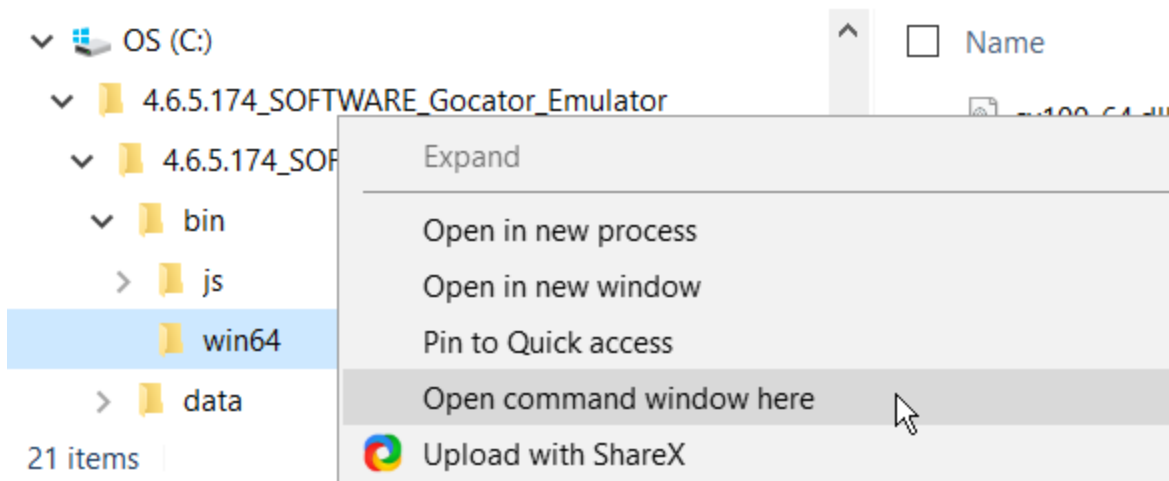
Running the Emulator in Default Browser

When you use the `/browser` command line parameter, the emulator application launches normally but also launches in your default browser. This provides additional flexibility when using the emulator. For example, you can resize the emulator running in a browser window.

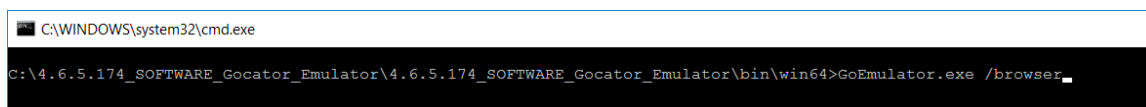
To run the emulator in your default browser:

1. In Windows Explorer (Windows 7) or File Explorer (Windows 8 or 10), browse to the location of the emulator.
The emulator is under `bin\win64`, in the location in which you installed the emulator.

2. Press and hold **Shift**, right-click the `win64` folder containing the emulator, and choose **Open command window here**.



3. In the command prompt, type `GoEmulator.exe /browser`, followed by an IPV4 address.



After the emulator application starts, the emulator also launches in your default browser.

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 215.

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.

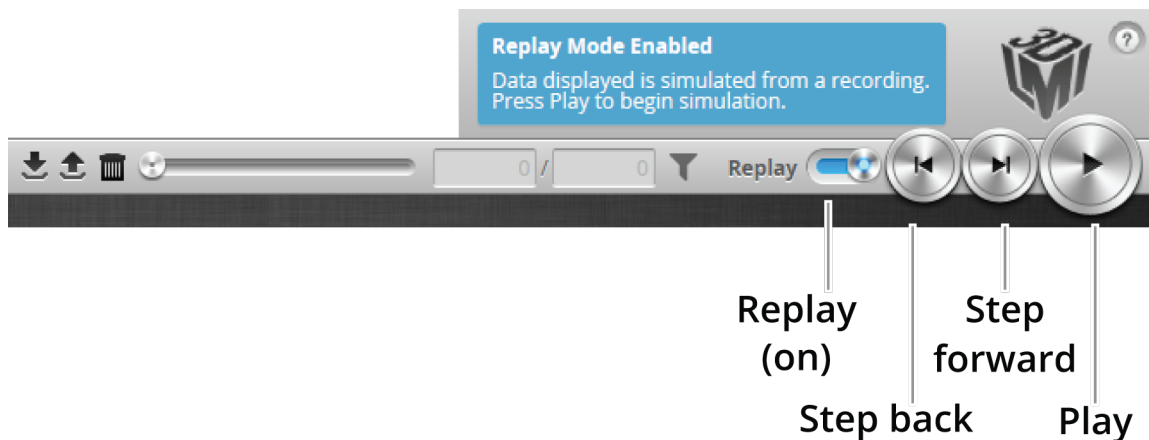
The job is activated. If there are any unsaved changes in the current job, you will be asked whether you want to discard those changes.

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 using the toolbar controls.

 Recording is not functional in the emulator.



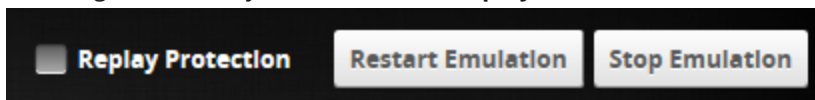
Playback controls when replay is on

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, you must uncheck **Replay Protection**.



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 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 120.
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 201.

To clear replay data:

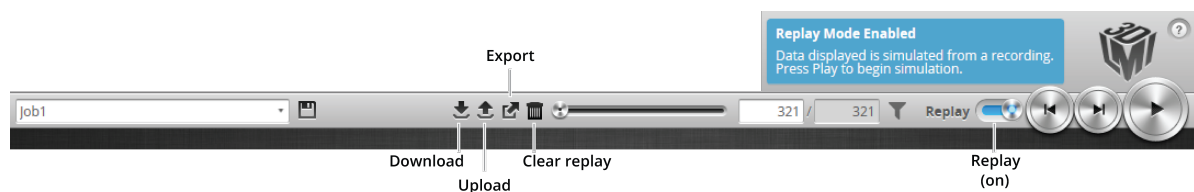
- Click the **Clear Replay Data** button .

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.



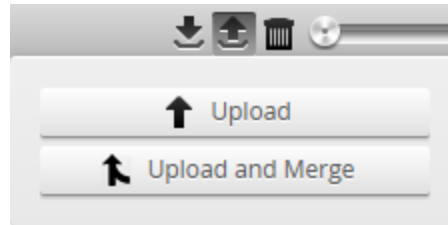
 Replay data is not loaded or saved when you load or save jobs.

To download replay data:

1. Click the Download button .
2. In the **File Download** dialog, click **Save**.
3. In the **Save As...** dialog, choose a location, optionally change the name (keeping the .rec extension), and click **Save**.

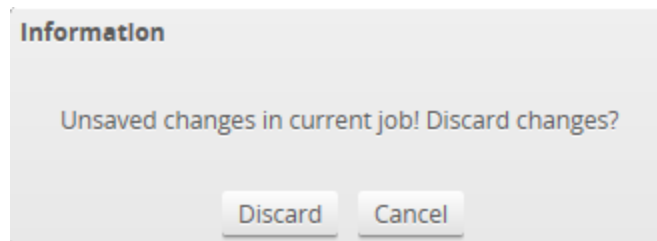
To upload replay data:

1. Click the Upload button .
The Upload menu appears.



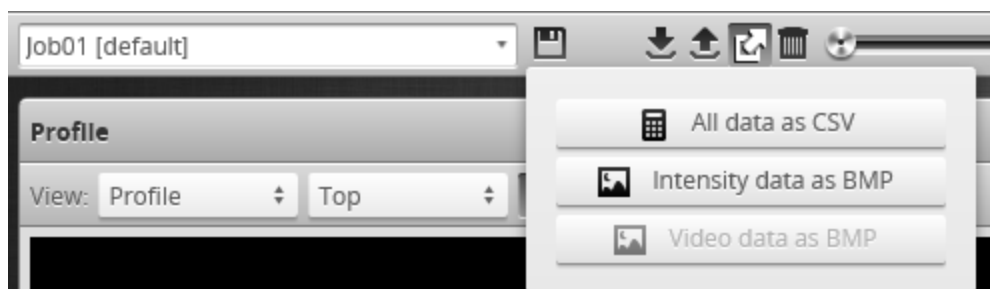
2. In the Upload menu, choose one of the following:
 - **Upload:** Unloads the current job and creates a new unsaved and untitled job from the content of the replay data file.
 - **Upload and merge:** Uploads the replay data and merges the data's associated job with the current job. Specifically, the settings on the **Scan** page are overwritten, but all other settings of the current job are preserved, including any measurements.

If you have unsaved changes in the current job, the firmware asks whether you want to discard the changes.



3. Do one of the following:
 - Click **Discard** to discard any unsaved changes.
 - Click **Cancel** to return to the main window to save your changes.
4. If you clicked **Discard**, navigate to the replay data to upload from the client computer and click **OK**.
The replay data is loaded, and a new unsaved, untitled job is created.

Replay data can be exported using the CSV format. If you have enabled **Acquire Intensity** in the **Scan Mode** panel on the **Scan** page, the exported CSV file includes intensity data.



To export replay data in the CSV format:

1. In the **Scan Mode** panel, switch to Range or Profile.
2. Click the Export button  and select **All Data as CSV**.
In Profile mode, all data in the record buffer is exported. 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 212.
3. (Optional) Convert exported data to another format using the CSV Converter Tool. For information on this tool, see *CSV Converter Tool* on page 402.



The decision values in the exported data depend on the *current* state of the job, not the state during recording. For example, if you record data when a measurement returns a *pass* decision, change the measurement's settings so that a *fail* decision is returned, and then export to CSV, you will see a *fail* decision in the exported data.

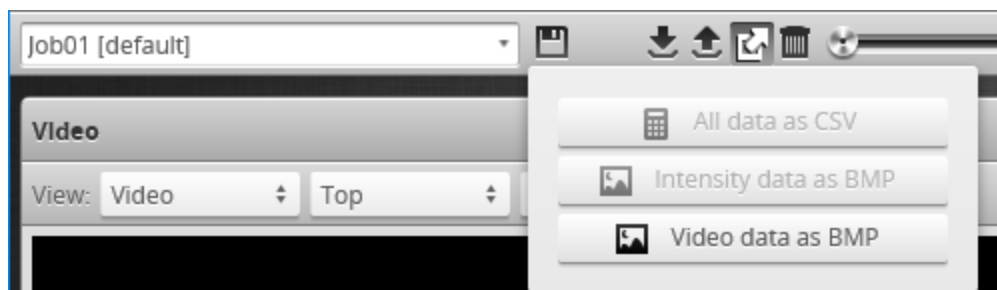
Recorded intensity data can be exported to a bitmap (.BMP format). **Acquire Intensity** must be checked in the **Scan Mode** panel while data was being recorded in order to export intensity data.

To export recorded intensity data to the BMP format:

- 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 212.

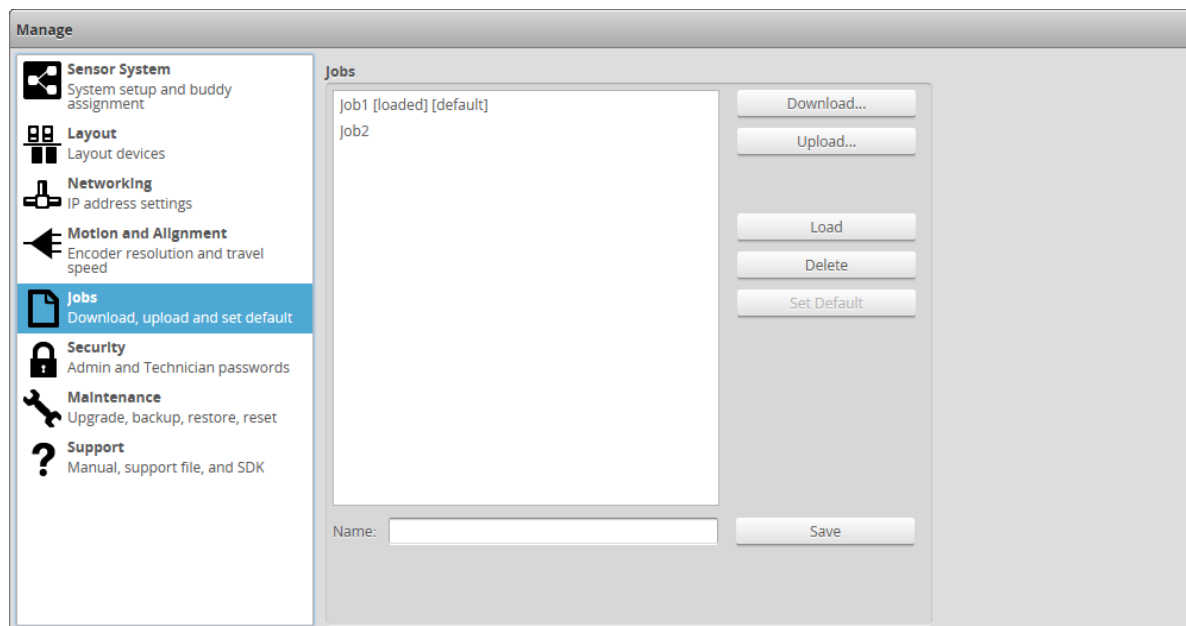


To export video data to a BMP file:

1. In the **Scan Mode** panel, switch to Video mode.
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 212.
2. Click the Export button  and select **Video data as BMP**.

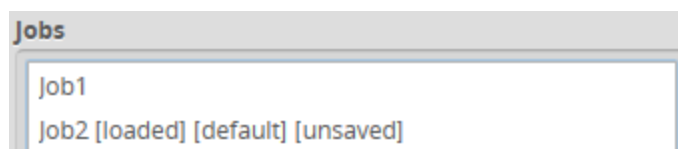
Downloading and Uploading Jobs

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 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 210.

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 and filters on the X axis. Note that modifying the Y filters causes the buffer to be cleared. (For more information on these features, see the Gocator Laser Profile Sensors user manual.)

For information on creating models and setting up part matching, see *Models and Part Matching* in the Gocator 2100, 2300 and 2880 user manual. For information on adding and configuring measurement tools, see *Measurement* on page 120.

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 92 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 95 for more information on exposure.)



To adjust active area in the emulator, **Replay Protection** must be turned off. See *Using Replay Protection* on page 210 for more information.



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

Protocol Output

The emulator simulates output for all of Gocator's Ethernet-based protocols.

- [Gocator](#)
- [ASCII](#)
- [Modbus](#)
- [EtherNet/IP](#)

Clients (such as PLCs) can connect to the emulator to access the simulated output and use the protocols as they would with a physical sensor.

The emulator allows connections to emulated sensors on localhost (127.0.0.1). You can also allow connections to emulated sensors on your computer's network card; for more information, see *Remote Operation* below.

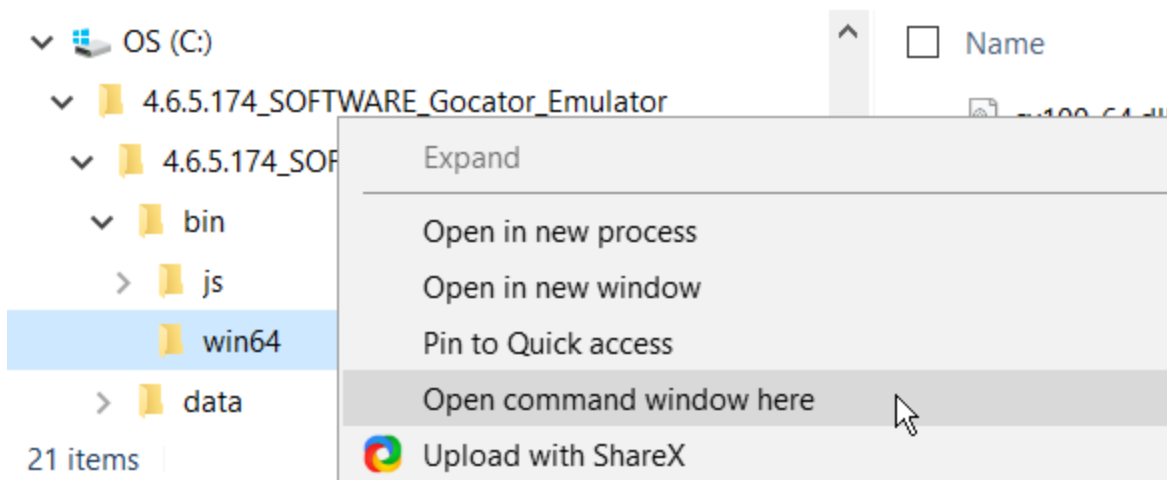
Remote Operation

You can specify the IP address of one of your computer's network cards to allow clients to connect remotely to an emulated sensor using the `/ip` command line parameter. When the `/ip` parameter is not used, emulated sensors are only available on the local machine (that is, 127.0.0.1 or localhost).

-  Clients can only connect to emulated sensors, not to the emulator's launch page.
-  You may need to contact your network administrator to allow connections to the computer running the emulated sensor.

To allow remote connections to an emulated sensor:

1. In Windows Explorer (Windows 7) or File Explorer (Windows 8 or 10), browse to the location of the emulator. The emulator is under `bin\win64`, in the location in which you installed the emulator.
2. Press and hold **Shift**, right-click the `win64` folder containing the emulator, and choose **Open command window here**.



3. In the command prompt, type `GoEmulator.exe /ip`, followed by an IPV4 address, for example:

```
C:\WINDOWS\system32\cmd.exe  
C:\4.6.5.174_SOFTWARE_Gocator_Emulator\4.6.5.174_SOFTWARE_Gocator_Emulator\bin\win64>GoEmulator.exe /ip 192.168.1.42
```

The emulator application starts.

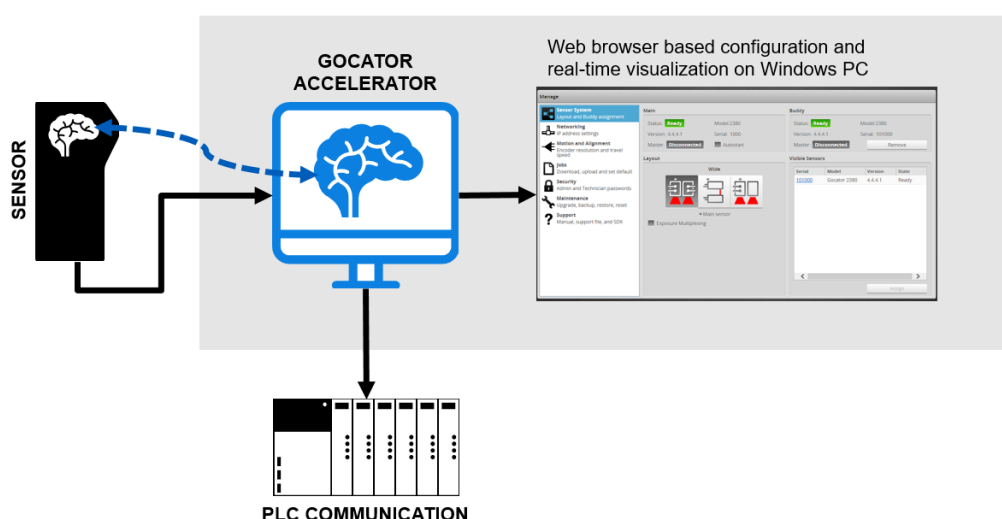
 The emulator does not check that the IP address is valid.

4. From the emulator launch page, start a scenario.
For more information, see *Running a Scenario* on page 208.
5. Provide the IP address you used with the `/ip` parameter, followed by port number 3191, to users who want to connect to the emulated sensor, for example:
`192.168.1.42:3191`

Gocator Accelerator

The Gocator Accelerator improves a Gocator system's processing capability by transferring the processing to a PC in the system. It can accelerate one or more standalone sensors or multi-sensor systems.

You can implement acceleration capabilities in client applications that you create using the [Gocator SDK](#). LMI also provides a standalone utility (GoAccelerator.exe) that you can use to accelerate systems.



The [web interface](#) on an accelerated sensor is identical to the interface on an unaccelerated sensor. The Ethernet-based [output protocols](#) (Gocator, EtherNet/IP, ASCII, and Modbus) are also identical to those found on an unaccelerated sensor, and are fully supported.



The Gocator Accelerator supports digital, analog, and serial output from sensors. However, because output must be passed to the PC and then back to the sensor, network latency will have an impact on performance.



The firmware version of the sensor you want to accelerate must match the version of the SDK used to build an accelerator-based application (or the version of the GoAccelerator utility).

When a sensor is accelerated, it sends data directly to the accelerating application. Users access the Gocator web interface using the IP address of the computer running the application, rather than the IP of the sensor.



Some health indicators behave differently when a sensor is accelerated. For information on which indicators in the Dashboard that acceleration affects, see *State and Health Information* on page 201. For information on which indicators available through the Gocator protocol acceleration affects, see *Health Results* on page 342.

Once a system is accelerated, an SDK application can interface to the accelerator application the same way as is possible with a physical sensor, although the IP of the accelerating PC must be used for the connection.

System Requirements

The following are the system requirements for the software:

- Processor: Intel Core i3 or equivalent (32- or 64-bit)
- RAM: 4 GB
- Hard drive: 500 GB
- Operating system: Windows 7, 8, or 10 (32- or 64-bit)

Benefits

Accelerated sensors provide several benefits.

Acceleration is completely transparent: because the output protocols of an accelerated sensor are identical to those of an unaccelerated sensor, SDK and PLC applications require no changes whatsoever for controlling accelerated sensors and receiving health information and data.

Measurement latency is reduced on accelerated sensors, which results in shorter cycle times. This means a sensor can scan more targets in a given time period.

The memory of accelerated sensors is limited only by the memory of the PC on which the Accelerator is running.

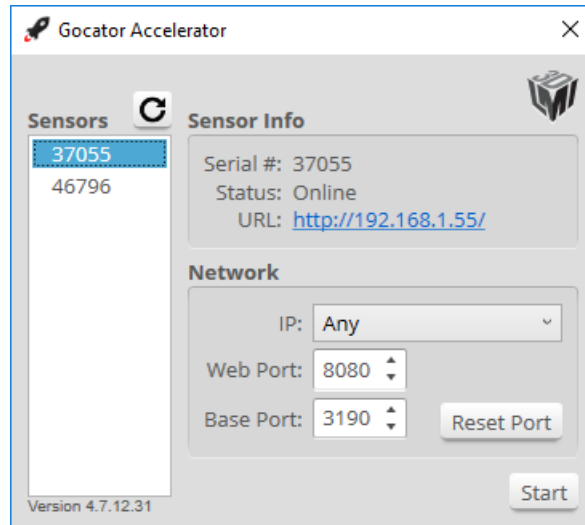
Installation

To get the necessary packages, go to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download it from the Download Center.

- For the GoAccelerator utility, download the 14405-X.X.X_X_SOFTWARE_GO_Tools.zip package.
- For the SDK libraries and DLL for integrating acceleration into a client application, download the 14400-X.X.X_X_SOFTWARE_GO_SDK.zip.

Gocator Accelerator Utility

The Gocator Accelerator utility accelerates the standalone sensors or multi-sensor systems you choose.



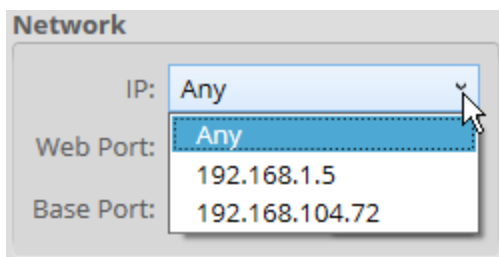
To accelerate a sensor using the Gocator Accelerator utility:

1. Power up the sensor system you want to accelerate.
2. Launch the Gocator Accelerator utility.
3. If a Windows Security alert asks whether you want to allow GoAccelerator.exe to communicate on networks, make sure **Public** and **Private** are checked, and then click **Allow Access**.
4. In the **Sensors** list, click the sensor you want to accelerate.

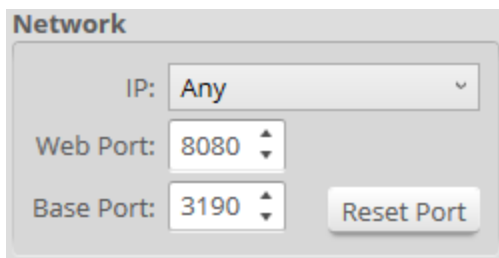
If you do not see the sensor, you may need to wait a few seconds and then click the Refresh button ().

In multi-sensor systems, only the Main sensor is listed.

5. (Optional) In the **IP** drop-down, choose an IP or choose **Any** to let the application choose.



6. (Optional) Set **Web Port** to a port for use with the accelerated sensor's URL.



If port 8080 is already in use, set **Web Port** to an unused port.

- (Optional) If you are accelerating multiple systems, click on another sensor in the **Sensors** list, and repeat the steps above.

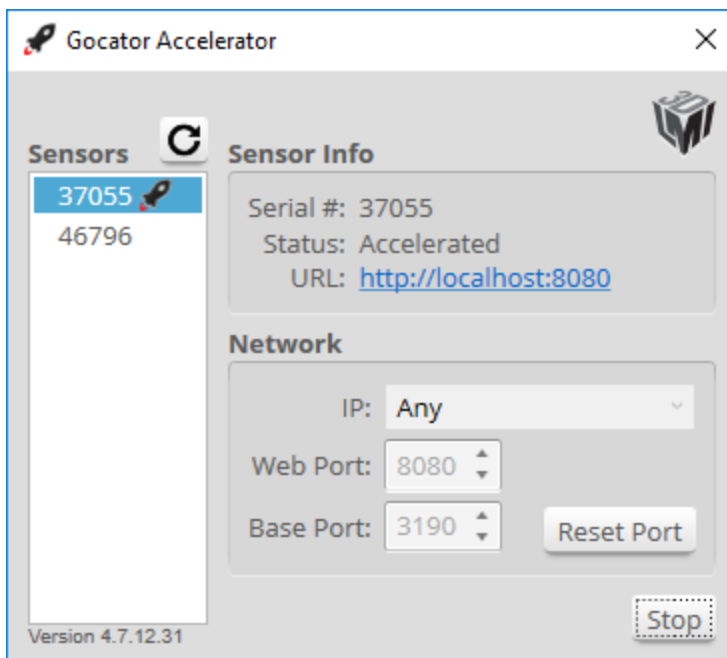
The application uses **Base Port** as an offset for several communication port numbers.

To avoid port conflicts, you should increment the base port number by at least 10 for each accelerated sensor.

Port 3190 is the default base port number, allowing connections from SDK-based applications and the web UI without manually specifying ports.

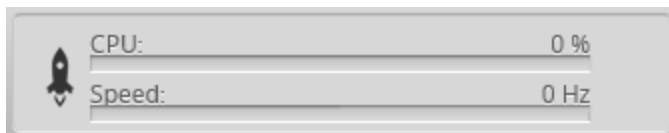
- Click **Start**.

The sensor system is now accelerated. An icon appears next to the accelerated sensor in the **Sensors** list to indicate this.



- To open the accelerated sensor's web interface, in the Gocator Accelerator application, click the link next to **URL**.

When a sensor is accelerated, a "rocket" icon appears in the [metrics area](#).



If you restart an accelerated sensor, the sensor will continue to be accelerated when it restarts.

To stop an accelerated sensor in the Gocator Accelerator application:

- Select the sensor in the **Sensors** list.
- Click **Stop**.

To exit the Gocator Accelerator application:

- Right-click the icon Gocator Accelerator icon (🚀) in the notification tray.

Clicking the X icon in the application only minimizes the application.

2. Choose **Exit**.

Dashboard and Health Indicators

After a sensor is accelerated, the values of some health indicators come from the accelerating PC instead of the sensor. Others come from a combination of the accelerated sensor and the accelerating PC.

- For information on which indicators are affected in the Dashboard in the web interface, see *State and Health Information* on page 201.
- For information on which indicators accessed through the Gocator protocol are affected, see *Health Results* on page 342.

SDK Application Integration

Gocator acceleration can be fully integrated into an SDK application. Users simply need to instantiate the GoAccelerator object and connect it to a sensor object.

```
GoAccelerator accelerator = kNULL;

// obtain GoSensor object by sensor IP address
if ((status = GoSystem_FindSensorByIpAddress(system, &ipAddress, &sensor)) != kOK)
{
    printf("Error: GoSystem_FindSensorByIpAddress:%d\n", status);
    return;
}

// construct accelerator
if ((status = GoAccelerator_Construct(&accelerator, kNULL)) != kOK)
{
    printf("Error: GoAccelerator_Construct:%d\n", status);
    return;
}

// start accelerator
if ((status = GoAccelerator_Start(accelerator)) != kOK)
{
    printf("Error: GoAccelerator_Start:%d\n", status);
    return;
}

printf("GoAccelerator_Start completed\n");
if ((status = GoAccelerator_Attach(accelerator, sensor)) != kOK)
{
    printf("Error: GoAccelerator_Attach:%d\n", status);
    return;
}

// create connection to GoSensor object
if ((status = GoSensor_Connect(sensor)) != kOK)
{
    printf("Error: GoSensor_Connect:%d\n", status);
    return;
}
```

After, the SDK application can control an accelerated sensor in the same way as an unaccelerated sensor.

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
<code>_live.job</code>	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 <i>Job File Structure</i> on the next page.
<code>_live.cfg</code>	Read/Write	A standalone representation of the Configuration component contained in <code>_live.job</code> . Used primarily for backwards compatibility.
<code>_live.tfm</code>	Read/Write	If Alignment Reference of the active job is set to Dynamic: A copy of the Transform component in <code>_live.job</code> . Used primarily for backwards compatibility. If Alignment Reference of the active job is set to Fixed: The transformations that are used for <i>all</i> jobs whose Alignment Reference setting is set to Fixed.
<code>_live.log</code>	Read	A sensor log containing various messages. For more information on the log file, see <i>Log File</i> below.
<code>_live.rec</code>	Read/Write	The active replay simulation data.
<code>ExtendedId.xml</code>	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
@idStart	64s	Identifier of the first log.
@idEnd	64s	Identifier of the final log.
List of (Info Warning Error)	List	An ordered list of log entries. This list is empty if idEnd < idStart.

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

Element	Type	Description
@time	64u	Log time, in uptime (µs).
@source	32u	The serial number of the sensor the log was produced by.
@id	32u	The Identifier, or index, of the log
@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 File Structure

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. 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. For more

Component	Path	Description
		information, see <i>Configuration</i> below.
Transform	transform.xml	Transformation values. Present only if Alignment Reference is set to Dynamic. For more information, see <i>Transform</i> on page 289.

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.

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.

All Gocator sensors share a common job file structure and settings for all features are included in job files, regardless of the model.



If a setting in a job file is not used by a sensor, the setting's *used* property is set to 0.

Configuration Child Elements

Element	Type	Description
@version	32u	Configuration version (101).

Element	Type	Description
@versionMinor	32u	Configuration minor version (9).
Setup	Section	For a description of the Setup elements, see <i>Setup</i> below.
Replay	Section	Contains settings related to recording filtering (see <i>Replay</i> on page 250).
Streams	Section	Read-only collection of available data streams (see <i>Streams/Stream (Read-only)</i> on page 251).
ToolOptions	Section	List of available tool types and their information. See <i>ToolOptions</i> on page 252 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 <i>Tools</i> on page 254.
Tools.options	String (CSV)	Deprecated. Replaced by ToolOptions .
Outputs	Section	For a description of the Output elements, see <i>Output</i> on page 282.

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.
OcclusionReductionEnabled.used	Bool	Whether or not property is used.
OcclusionReductionEnabled.value	Bool	Actual value used if not configurable.
OcclusionReductionAlg	32s	The Algorithm to use for occlusion reduction: 0 – Standard 1 – High Quality
OcclusionReductionAlg.used	Bool	Whether or not property is used
OcclusionReductionAlg.value	Bool	Actual value used if not configurable
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.

Element	Type	Description
IntensityEnabled.used	Bool	Whether or not property is used.
IntensityEnabled.value	Bool	Actual value used if not configurable.
FlickerFreeModeEnabled	Bool	Enables flicker-free operation.
FlickerFreeModeEnabled.used	Bool	Whether flicker-free operation can be used on this sensor.
ExternalInputZPulseEnabled	Bool	Enables the External Input based encoder Z Pulse feature.
Filters	Section	See <i>Filters</i> below. Used by Gocator profile and snapshot sensors.
Trigger	Section	See <i>Trigger</i> on page 233.
Layout	Section	See <i>Layout</i> on page 234.
Alignment	Section	See <i>Alignment</i> on page 235.
Devices	Collection	A collection of two Device sections (with roles main and buddy). See <i>Devices / Device</i> on page 238.
SurfaceGeneration	Section	See <i>SurfaceGeneration</i> on page 244. Used by Gocator profile sensors.
SurfaceSections	Section	See <i>SurfaceSections</i> on page 245. Used by Gocator profile and snapshot sensors.
ProfileGeneration	Section	See <i>ProfileGeneration</i> on page 245.
PartDetection	Section	See <i>PartDetection</i> on page 246.
PartMatching	Section	See <i>PartMatching</i> on page 248. Used by Gocator profile and snapshot 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 profile and snapshot sensors.

XSmoothing

XSmoothing Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
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
@used	Bool	Whether or not this field is used

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
@used	Bool	Whether or not this field is used
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
@used	Bool	Whether or not this field is used
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
@used	Bool	Whether or not this field is used
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
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.

Element	Type	Description
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
@used	Bool	Whether or not this field is used
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
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

XSlope

	This filter is only available on displacement sensors.
---	--

XSlope Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
Enabled	Bool	Enables filtering.
Window	64f	Window size (mm).
Window.min	64f	Minimum window size (mm).
Window.max	64f	Maximum window size (mm).

YSlope

	This filter is only available on displacement sensors.
---	--

YSlope Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used

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.

	Gocator 1300 series sensors are limited to sending data at 10 kHz over the analog output channel.
	Therefore, if you configure a sensor so that it runs at a speed higher than 10 kHz, and configure a measurement to be sent on the analog channel, you will get analog data drops.
	To achieve a 10 kHz analog output rate, you must enable and configure scheduled output.

Trigger Child Elements

Element	Type	Description
Source	32s	Trigger source: 0 – Time 1 – Encoder 2 – Digital Input 3 – Software
Source.options	32s (CSV)	List of available source options.
Units	32s	Sensor triggering units when source is not clock or encoder: 0 – Time 1 – Encoder
FrameRate	64f	Frame rate for time trigger (Hz).
FrameRate.min	64f	Minimum frame rate (Hz).
FrameRate.max	64f	Maximum frame rate (Hz).
FrameRate.maxSource	32s	Source of maximum frame rate limit: 0 – Imager 1 – Surface generation
TracheidRate	64f	The frame rate of Tracheid data (Read Only)
TracheidRate.used	Bool	Whether the sensor has a Tracheid data rate.
FrameDataRate	64f	The frame rate of normal (range/profile/surface) data (Read Only)
FrameDataRate.used	Bool	Whether the sensor has a separate FrameDataRate
EncoderSpacing.max	64f	Maximum encoder spacing (mm).
EncoderSpacing.minSource	32s	Source of minimum encoder spacing: 0 – Resolution 1 – Surface generation

Element	Type	Description
EncoderSpacing.used	Bool	Whether or not this parameter is configurable.
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.
BurstEnabled	Bool	Enables burst triggering.
BurstEnabled.Used	Bool	Whether or not this parameter is configurable.
BurstCount	32u	Number of scans to take during burst triggering.
BurstCount.used	Bool	Whether or not this parameter is configurable.
BurstCount.max	32u	Maximum burst count.
ReversalDistanceAutoEnabled	Bool	Whether or not to use auto-calculated value.
ReversalDistanceAutoEnabled.used	Bool	Whether or not this parameter can be configured.
ReversalDistance	64f	Encoder reversal threshold (for jitter handling)
ReversalDistance.used	Bool	Whether or not this parameter is used.
ReversalDistance.value	64f	Actual value.
LaserSleepMode.used	Bool	Whether or not this feature can be configured
LaserSleepMode/Enabled	Bool	Enables or disables the feature.
LaserSleepMode/IdleTime	64u	Idle time before laser is turned off (µs).
LaserSleepMode/WakeupEncoderTravel	64u	Minimum amount of encoder movement before laser turns on (mm).

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 4 – Top Bottom 5 – Left Right

Element	Type	Description
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 – Normal (single-sensor system) / Wide (dual-sensor system) 1 – Opposite 2 – Reverse 3 – Grid
Grid	Grid	Grid representation of the multi-sensor layout.
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 μ s (currently rounded to integer when read by the sensor)
MultiplexSingleDelay	64f	Delay in μ s. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod	64f	Period in μ s. (Currently gets rounded up when read by the sensor.)
MultiplexSinglePeriod.min	64f	Minimum period in μ s.

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).
ZAngle	64f	Z Angle start (degrees).
ZAngle.used	Bool	Whether or not this property is used.

Grid Elements

Element	Type	Description
ColumnCount	32u	Column count.
ColumnCount.value	32u	Column count value.

Alignment

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

Alignment Child Elements

Element	Type	Description
@used	Bool	Whether or not this field is used
InputTriggerEnabled	Bool	Enables digital input-triggered alignment operation.
InputTriggerEnabled.used	Bool	Whether or not this feature can be enabled. This feature is available only on some sensor models.
InputTriggerEnabled.value	Bool	Actual feature status.
Type	32s	Type of alignment operation: 0 – Stationary 1 – Moving
Type.options	32s (CSV)	List of available alignment types.
StationaryTarget	32s	Stationary alignment target: 0 – None 1 – Disk 2 – Bar 3 – Plate
StationaryTarget.options	32s (CSV)	List of available stationary alignment targets.
MovingTarget	32s	Moving alignment target: 1 – Disk 2 – Bar
MovingTarget.options	32s (CSV)	List of available moving alignment targets.
EncoderCalibrateEnabled	Bool	Enables encoder resolution calibration.
Disk	Section	See <i>Disk</i> below.
Bar	Section	See <i>Bar</i> below.
Plate	Section	See <i>Plate</i> on the next page.
Polygon	Section	See <i>Polygon</i> 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).

Element	Type	Description
HoleCount	32u	Number of holes.
HoleCount.value	32u	Actual number of holes expected by system.
HoleCount.used	Bool	Whether the hole count will be used in the bar alignment procedure.
HoleDistance	64f	Distance between holes (mm).
HoleDistance.used	Bool	Whether the hole distance will be used in the bar alignment procedure.
HoleDiameter	64f	Diameter of holes (mm).
HoleDiameter.used	Bool	Whether the hole diameter will be used in the bar alignment procedure.
DegreesOfFreedom	32s	Degrees of freedom (DOF) to align: 42 – 3 DOF: x, z, y angle 58 – 4 DOF: x, y, z, y angle 59 – 5 DOF: x, y, z, y angle, z angle

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).

Polygon

Polygon Child Elements

Element	Type	Description
Corners	List	Contains a list of Corners (described below).
Corners.minCount	32s	Minimum number of corners.

Polygon/Corner

Corner Child Elements

Element	Type	Description
X	64f	X Position
Y	64f	Y Position
Devices	List of 32u	List of devices this corner is assigned to.
Devices.options	List of 32u	List of valid options for this field.

Devices / Device

Devices / Device Child Elements

Element	Type	Description
@index	32u	Ordered index of devices in device list.
@role	32s	Sensor role: 0 – Main 1 – Buddy
Layout	Layout	Multiplexing bank settings.
DataSource	32s	Data source of device output (read-only): 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
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).
FrontCamera	Window	Front camera window (read-only).
BackCamera	Window	Back camera window (read-only).
BackCamera.used	Bool	Whether or not this field is used.
PatternSequenceType	32s	The projector pattern sequence to display when a projector equipped device is running. The following types are possible: -1 – None 0 – Default 100 – Nine Lines
PatternSequenceType.options	32s	List of available pattern sequence types.
PatternSequenceType.used	Bool	Whether or not this field is used.
PatternSequenceCount	32u	Number of frames in the active sequence (read-only).
ExposureMode	32s	Exposure mode: 0 – Single exposure 2 – Dynamic exposure
ExposureMode.options	32s (CSV)	List of available exposure modes.
Exposure	64f	Single exposure (μs).
Exposure.min	64f	Minimum exposure (μs).
Exposure.max	64f	Maximum exposure (μs).
Exposure.used	Bool	Whether or not this field is used.

Element	Type	Description
DynamicExposureMin	64f	Dynamic exposure range minimum (μs).
DynamicExposureMax	64f	Dynamic exposure range maximum (μs).
ExposureSteps	64f (CSV)	Mutiple exposure list (μs).
ExposureSteps.countMin	32u	Minimum number of exposure steps.
ExposureSteps.countMax	32u	Maximum number of exposure steps.
IntensitySource	32s	Intensity source: 0 – Both cameras 1 – Front camera 2 – Back camera
IntensitySource.options	32s (CSV)	List of available intensity sources.
IntensityMode	32s	Intensity Mode: 0 – Auto 1 - Preserve
IntensityMode.used	Bool	Whether intensity mode is used
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 this field is used.
Tracking	Section	See <i>Tracking Child Elements</i> on the next page.
Material	Section	See <i>Material Child Elements</i> on page 241.
Tracheid	Section	See <i>Tracheid Child Elements</i> on page 243.
IndependentExposures	Section	See <i>IndependentExposures Child Elements</i> on page 243
Custom	Custom	Used by specialized sensors.

Region3D Child Elements

Element	Type	Description
X	64f	X start (mm).
Y	64f	Y start (mm).

Element	Type	Description
Z	64f	Z start (mm).
Width	64f	X extent (mm).
Length	64f	Y extent (mm).
Height	64f	Z extent (mm).
ZAngle	64f	Z Angle start (degrees).
ZAngle.used	Bool	Whether or not this property is used.

Window Child Elements

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

Layout Child Elements

Element	Type	Description
Grid	Grid	Layout grid information.
MultiplexingBank	32u	Multiplexing bank ID
MultiplexingBank.used	32u	Whether or not this field can be specified
MultiplexingBank.value	32u	Actual value used by system

Grid Child Elements

Element	Type	Description
@used	Bool	Whether or not this section is used.
Row	32s	Device row position in grid layout.
Row.value	32s	Value in use by the sensor, useful for determining value when used is false.
Column	32s	Device column position in grid layout.
Column.value	32s	Value in use by the sensor, useful for determining value when used is false.
Direction	32s	Sensor orientation direction.
Direction.value	32s	Value in use by the sensor, useful for determining value when used is false.



Tracking is only available on Gocator 2300 and 2400 series sensors.

Tracking Child Elements

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

Element	Type	Description
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 Child Elements

Element	Type	Description
Type	32s	Type of Material settings to use. 0 – Custom 1 – Diffuse 3 – Reflective
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.
Type.options	32u (CSV)	List of available material types.
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 3 – None. All spots are available. This option may not be available in some configurations. 4 – Continuity. Picks the most continuous spot.
SpotSelectionType.used	Bool	Determines if the setting's value is currently used.
SpotSelectionType.value	32s	Value in use by the sensor, useful for determining value when used is false.
SpotSelectionType.options	32s (CSV)	List of available spot selection types.

Element	Type	Description
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.
SensitivityCompensationEnabled	Bool	Sensitivity compensation toggle. Used in determining analog and digital gain, along with exposure scale.
SensitivityCompensationEnabled.used	Bool	Determines if the setting's value is currently used.
SensitivityCompensationEnabled.value	Bool	Value in use by the sensor, useful for determining value when used is false.
GammaType	32s	Gamma type.
GammaType used	Bool	Determines if the setting's value is currently used.
GammaType value	32s	Value in use by the sensor. Useful for determining value when used is false.
SpotContinuitySorting	Section	See <i>SpotContinuitySorting Child Elements</i> on the next page.
SurfaceEncoding	32s	Surface encoding type: 0 – Standard

Element	Type	Description
		1 – Interreflection (advanced use only)
SurfaceEncoding.used	Bool	Determines if the setting's value is currently used.
SurfaceEncoding.value	Bool	Value in use by the sensor, useful for determining value when used is false.
SurfacePhaseFilter	32s	Surface phase filter (correction type) 0 – None 1 – Reflective 2 – Translucent
SurfacePhaseFilter.used	Bool	Determines if the setting's value is currently used.
SurfacePhaseFilter.value	Bool	Value in use by the sensor, useful for determining value when used is false.

SpotContinuitySorting Child Elements

Element	Type	Description
MinimumSegmentSize	32u	Smallest continuous segment considered in continuity sorting.
SearchWindow/X	32u	X component of continuity sorting search window size.
SearchWindow/Y	32u	Y component of continuity sorting search window size.

 IndependentExposures settings are only supported by Gocator 3000 series sensors.

IndependentExposures Child Elements

Element	Type	Description
@used	Bool	Whether this field is used
Enabled	Bool	Whether to allow using separate exposure values for each camera
FrontCameraExposure	64f	The exposure value to use for the front camera
FrontCameraExposure.min	64f	The minimum exposure value possible for front camera
FrontCameraExposure.max	64f	The maximum exposure value possible for back camera
BackCameraExposure	64f	The exposure value to use for the front camera
BackCameraExposure.min	64f	The minimum exposure value possible for front camera
BackCameraExposure.max	64f	The maximum exposure value possible for back camera

 Tracheid settings are only supported by Gocator 200 series multi-point sensors.

Tracheid Child Elements

Element	Type	Description
@used	Bool	Whether this field is used
TracheidExposureEnabled	Bool	Whether to use a unique exposure for tracheid capture
TracheidExposure	64f	The exposure value to use for tracheid measurements
TracheidExposure.min	64f	The minimum exposure value possible tracheid measurements

Element	Type	Description
TracheidExposure.max	64f	The maximum exposure value possible for tracheid measurements
Camera0Threshold	32u	The tracheid threshold for camera 0
Camera1Threshold	32u	The tracheid threshold for camera 1

SurfaceGeneration

The SurfaceGeneration element contains settings related to surface generation.

This element is used by Gocator laser profile sensors.

SurfaceGeneration Child Elements

Element	Type	Description
Type	32s	Surface generation type: 0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational
FixedLength	Section	See <i>FixedLength</i> below.
VariableLength	Section	See <i>VariableLength</i> below.
Rotational	Section	See <i>Rotational</i> on the next page.

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).

SurfaceSections

SurfaceSections Child Elements

Element	Type	Description
@used	Bool	Whether surface sectioning is enabled.
@xMin	64f	The minimum valid X value to be used for section definition.
@xMax	64f	The maximum valid X value to be used for section definition.
@yMin	64f	The minimum valid Y value to be used for section definition.
@yMax	64f	The maximum valid Y value to be used for section definition.
Section	Collection	A series of Section elements.

Section Child Elements

Element	Type	Description
@id	32s	The ID assigned to the surface section.
@name	String	The name associated with the surface section.
StartPoint	Point64f	The beginning point of the surface section.
EndPoint	Point64f	The end point of the surface section.
CustomSpacingIntervalEnabled	Bool	Indicates whether a user specified custom spacing interval is to be used for the resulting section.
SpacingInterval	64f	The user specified spacing interval.
SpacingInterval.min	64f	The spacing interval limit minimum.
SpacingInterval.max	64f	The spacing interval limit maximum.
SpacingInterval.value	64f	The current spacing interval used by the system.

ProfileGeneration

The ProfileGeneration element contains settings related to profile generation.

ProfileGeneration Child Elements

Element	Type	Description
Type	32s	Profile generation type: 0 – Continuous 1 – Fixed length 2 – Variable length 3 – Rotational

Element	Type	Description
FixedLength	Section	See <i>FixedLength</i> below.
VariableLength	Section	See <i>VariableLength</i> below.
Rotational	Section	See <i>Rotational</i> 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.
Enabled.used	Bool	Whether or not this field is used.
Enabled value	Bool	Actual value used if not configurable.
MinArea	64f	Minimum area (mm ²).
MinArea.min	64f	Minimum value of minimum area.
MinArea.max	64f	Maximum value of minimum area.

Element	Type	Description
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.
Threshold	64f	Height threshold (mm).
Threshold.min	64f	Minimum height threshold (mm).
Threshold.max	64f	Maximum height threshold (mm).
ThresholdDirection	32u	Threshold direction: 0 – Above 1 – Below
FrameOfReference	32s	Part frame of reference: 0 – Sensor 1 – Scan 2 – Part
FrameOfReference.used	Bool	Whether or not this field is used.

Element	Type	Description
FrameOfReference.value	32s	Actual value.
IncludeSinglePointsEnabled	Bool	Enables preservation of single data points in Top+Bottom layout
IncludeSinglePointsEnabled. used	Bool	Whether or nto this field is available to be modified
EdgeFiltering	Section	See <i>EdgeFiltering</i> below. (Not used by G1 sensors.)

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. This element is used by Gocator profile and snapshot sensors.

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 <i>Edge</i> below.
BoundingBox	Section	See <i>BoundingBox</i> on the next page.
Ellipse	Section	See <i>Ellipse</i> 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).
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and, if enabled, which dimension is the basis of detection. The possible values are: 0 – None 1 – Length 2 – Width
Acceptance/Width/Min	64f	Minimum width (mm).
Acceptance/Width/Max	64f	Maximum width (mm).
Acceptance/Width/Tolerance	64f	Width acceptance tolerance value
Acceptance/Width/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
Acceptance/Length/Min	64f	Minimum length (mm).
Acceptance/Length/Max	64f	Maximum length (mm).
Acceptance/Length/Tolerance	64f	Length acceptance tolerance value
Acceptance/Length/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
X	64f	X value
X.deprecated	Bool	Whether this X field is deprecated
Y	64f	Y value
Y.deprecated	Bool	Whether this Y field is deprecated
Width	64f	Width value
Width.deprecated	Bool	Whether this width field is deprecated
Length	64f	Length value
Length.deprecated	Bool	Whether this length field is deprecated

Ellipse

Ellipse Child Elements

Element	Type	Description
ZAngle	64f	Z rotation to apply to ellipse (degrees).
AsymmetryDetectionType	32s	Determine whether to use asymmetry detection and, if enabled, which dimension is the basis of detection. The possible values are: 0 – None 1 – Major 2 – Minor
Acceptance/Major/Min	64f	Minimum major length (mm).

Element	Type	Description
Acceptance/Major/Max	64f	Maximum major length (mm).
Acceptance/Major/Tolerance	64f	Major acceptance tolerance value
Acceptance/Major/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
Acceptance/Minor/Min	64f	Minimum minor length (mm).
Acceptance/Minor/Max	64f	Maximum minor length (mm).
Acceptance/Minor/Tolerance	64f	Minor acceptance tolerance value
Acceptance/Minor/Tolerance.deprecated	Bool	Whether this tolerance field is deprecated
X	64f	X value
X.deprecated	Bool	Whether this X field is deprecated
Y	64f	Y value
Y.deprecated	Bool	Whether this Y field is deprecated
Width	64f	Width value
Width.deprecated	Bool	Whether this width field is deprecated
Length	64f	Length value
Length.deprecated	Bool	Whether this length field is deprecated

Replay

Contains settings related to recording filtering.

RecordingFiltering

RecordingFiltering Child Elements

Element	Type	Description
ConditionCombineType	32s	0 – Any: If any enabled condition is satisfied, the current frame is recorded. 1 – All: All enabled conditions must be satisfied for the current frame to be recorded.
Conditions	Collection	A collection of AnyMeasurement , AnyData , or Measurement conditions.

Conditions/AnyMeasurement

Conditions/AnyMeasurement Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
Result	32s	The measurement decision criteria to be included in the filter. Possible values are: 0 – Pass 1 – Fail

Element	Type	Description
		2 – Valid 3 – Invalid

Conditions/AnyData

Conditions/AnyData Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
RangeCountCase	32s	The case under which to record data: 0 – Range count at or above threshold of valid data points. 1 – Range count below threshold.
RangeCountThreshold	32u	The threshold for the number of range points that are valid.

Conditions/Measurement

Conditions/Measurement Elements

Element	Type	Description
Enabled	Bool	Indicates whether the condition is enabled.
Result	32s	The measurement decision criteria for the selected ID to be included in the filter. Possible values are: 0 – Pass 1 – Fail 2 – Valid 3 – Invalid
Ids	32s	The ID of the measurement to filter.

Streams/Stream (Read-only)

Streams/Stream Child Elements

Element	Type	Description
Step	32s	The data step of the stream being described. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Id	32u	The stream ID.
Cadenceld	32u	Represents a stage in the data processing pipeline. The greater the number, the farther removed from the initial acquisition stage. One of the following: 0 – Primary 1 – Auxiliary

Element	Type	Description
		10 - Diagnostic
DataType	32s	The stream data type 0 - None 4 - Uniform Profile 16 - Uniform Surface
ColorEncoding	32s	The color encoding type. Only appears for Video stream steps (1). 0 - None 1 - Bayer BGGR 2 - Bayer GBRG 3 - Bayer RGGB 4 - Bayer GRBG
IntensityEnabled	Bool	Whether the stream includes intensity data
Sources	Collection	A collection of Source elements as described below.

Source Child Elements

Element	Type	Description
Id	32s	The ID of the data source. Possible values are: 0 - Top 1 - Bottom 2 - Top Left 3 - Top Right 4 - Top Bottom 5 - Left Right
Capability	32s	The capability of the data stream source. Possible values are: 0 - Full 1 - Diagnostic only 2 - Virtual
Region	Region3d	The region of the given stream source.
AdditionalRegions	Collection	Collection of additional regions (for example, for the second camera).
AdditionalRegions/Region	Region3d	Additional regions.

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.
@format	32s	Format type of the tool: 0 – Standard built-in tool 1 – GDK user-defined tool 2 – Internal GDK tool
MeasurementOptions	Collection	See <i>MeasurementOptions</i> below
FeatureOptions	Collection	See <i>FeatureOptions</i> below.
StreamOptions	Collection	See <i>StreamOptions</i> 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.

FeatureOptions

FeatureOptions Child Elements

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

<Feature Name> Child Elements

Element	Type	Description
@displayName	String	Display name of the feature.
@minCount	32u	Minimum number of instances in a tool.
@maxCount	32u	Maximum number of instances in a tool.
@dataType	String	The data type of the feature. One of: – PointFeature – LineFeature – CircleFeature – PlaneFeature

StreamOptions

StreamOptions Child Elements

Element	Type	Description
@step	32s	The data step of the stream being described. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
@ids	CSV	A list representing the available IDs associated with the given step.

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.

Profile Types

The following types are used by various measurement tools.

ProfileFeature

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

ProfileFeature Child Elements

Element	Type	Description
Type	32s	Determine how the feature is detected within the area: 0 – Max Z 1 – Min Z 2 – Max X 3 – Min X 4 – Corner 5 – Average 6 – Rising Edge 7 – Falling Edge 8 – Any Edge 9 – Top Corner 10 – Bottom Corner 11 – Left Corner 12 – Right Corner

Element	Type	Description
		13 – Median
RegionEnabled	Bool	Indicates whether feature detection applies to the defined Region or to the entire active area.
Region	ProfileRegion2D	Element for feature detection area.

ProfileLine

An element of type ProfileLine defines measurement areas used to calculate a line.

ProfileLine Child Elements

Element	Type	Description
RegionCount	32s	Count of the regions.
Regions	(Collection)	The regions used to calculate a line. Contains one or two Region elements of type ProfileRegion2D , with RegionEnabled fields for each.

ProfileRegion2d

An element of type ProfileRegion2d defines a rectangular area of interest.

ProfileRegion2d Child Elements

Element	Type	Description
X	64f	Setting for profile region X position (mm).
Z	64f	Setting for profile region Z position (mm).
Width	64f	Setting for profile region width (mm).
Height	64f	Setting for profile region height (mm).

Geometric Feature Types

The Geometric Feature type is used by various measurement tools.

Feature Child Elements

Element	Type	Description
@id	32s	The identifier of the geometric feature. -1 if unassigned.
@dataType	String	The data type of the feature. One of: – PointFeature – LineFeature
@type	String	Type name of feature.
Name	String	The display name of the feature.
Enabled	Bool	Whether the given feature output is enabled.
Parameters	Collection	Collection of GdkParam elements.

Parameter Types

The following types are used by internal and custom (user-created) GDK-based tools.

GDK Parameter Child Elements

Element	Type	Description
@label	String	Parameter label.
@type	String	Type of parameter. It is one of the following (see tables below for elements found in each type): - Bool - Int - Float - ProfileRegion - SurfaceRegion2d - SurfaceRegion3d - GeometricFeature
@options	Variant (CSV)	Options available for this parameter.
@optionNames	String (CSV)	Names

GDK Parameter Bool Type

Element	Type	Description
	Bool	Boolean value of parameter.

GDK Parameter Int Type

Element	Type	Description
	32s	Integer value of parameter of integer type.

GDK Parameter Float Type

Element	Type	Description
	64f	Floating point value of parameter.

GDK Parameter String Type

Element	Type	Description
	String	String value of parameter.

GDK Parameter Profile Region Type

Element	Type	Description
X	64f	X value of region.
Z	64f	Z value of region.
Width	64f	Width value of region.
Height	64f	Height value of region.

GDK Parameter Surface Region 2D Type

Element	Type	Description
X	64f	X value of region.
X	64f	X value of region.

Element	Type	Description
Y	64f	Y value of region.
Width	64f	Width value of region.
Length	64f	Length value of region.

GDK Parameter Surface Region 3D Type

Element	Type	Description
X	64f	X value of region.
Y	64f	Y value of region.
Z	64f	Z value of region.
Width	64f	Width value of region.
Length	64f	Length value of region.
Height	64f	Height value of region.
ZAngle	64f	ZAngle value of region.

GDK Parameter Geometric Feature Type

Element	Type	Description
	32s	Geometric feature Id for parameter.

RangePosition

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

RangePosition Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Range source.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	Not used.
Stream\Step	32s	Not used.
Stream\ld	32u	Not used.
Measurements\Z	Position tool measurement	Z measurement. Determines the Z axis position of the laser range. Position Z 

Position Tool Measurement

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

Element	Type	Description
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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

RangeThickness

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

RangeThickness Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Range source.
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	Not used.
Stream\Step	32s	Not used
Stream\ld	32u	Not used.
Absolute	Boolean	Setting for selecting absolute or signed result: 0 – Signed 1 – Absolute
Measurements\Thickness	Thickness tool measurement	Thickness measurement.

Thickness 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileArea

A ProfileArea element defines settings for a profile area tool and one or more of its measurements.

ProfileArea Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfileArea</i> above.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video

Element	Type	Description
		2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
Type	Boolean	Area to measure: 0 – Object (convex shape above the baseline) 1 – Clearance (concave shape below the baseline)
Type.used	Boolean	Whether or not field is used.
Baseline	Boolean	Baseline type: 0 – X-axis 1 – Line
Baseline.used	Boolean	Whether or not field is used.
RegionEnabled	Boolean	If enabled, the defined region is used for measurements. Otherwise, the full active area is used.
Region	ProfileRegion2d	Measurement region.
Line	ProfileLine	Line definition when Baseline is set to Line.
Measurements\Area	Area tool measurement	Area measurement.
Measurements\CentroidX	Area tool measurement	CentroidX measurement.
Measurements\CentroidZ	Area tool measurement	CentroidZ measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.
<i>Area Tool Measurement</i>		
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

Element	Type	Description
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileBoundingBox

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

ProfileBoundingBox Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfileBoundingBox</i> above.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Bounding Box tool measurement	X measurement.
Measurements\Z	Bounding Box tool measurement	Z measurement.
Measurements\Width	Bounding Box tool measurement	Width measurement.

Element	Type	Description
Measurements\Height	Bounding Box tool measurement	Height measurement.
Measurements\GlobalX	Bounding Box tool measurement	GlobalX measurement
Measurements\GlobalY	Bounding Box tool measurement	GlobalY measurement
Measurements\GlobalAngle	Bounding Box tool measurement	GlobalAngle measurement
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.
Features\CornerPoint	GeometricFeature	CornerPoint PointFeature.

Bounding Box 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileCircle

A ProfileCircle element defines settings for a profile circle tool and one or more of its measurements.

ProfileCircle Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfileCircle</i> on the previous page.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Circle tool measurement	X measurement.
Measurements\Z	Circle tool measurement	Z measurement.
Measurements\Radius	Circle tool measurement	Radius measurement.
Features\CenterPoint	GeometricFeature	CenterPoint PointFeature.

Circle 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:

Element	Type	Description
		0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileDimension

A ProfileDimension element defines settings for a profile dimension tool and one or more of its measurements.

ProfileDimension Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RefFeature	ProfileFeature	Reference measurement region.
Feature	ProfileFeature	Measurement region.
Measurements\Width	Dimension tool measurement	Width measurement.
Measurements\Height	Dimension tool measurement	Height measurement.
Measurements\Distance	Dimension tool measurement	Distance measurement.

Element	Type	Description
Measurements\CenterX	Dimension tool measurement	CenterX measurement.
Measurements\CenterZ	Dimension tool measurement	CenterZ measurement.

Dimension 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
Absolute (Width and Height measurements only)	Boolean	Setting for selecting absolute or signed result: 0 – Signed 1 – Absolute

ProfileGroove

A ProfileGroove element defines settings for a profile groove tool and one or more of its measurements.

The profile groove tool is dynamic, meaning that it can contain multiple measurements of the same type in the Measurements element.

ProfileGroove Child Elements

Element	Type	Description
Name	String	Tool name.

Element	Type	Description
Features	Collection	Not used.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
Shape	32s	Shape: 0 – U-shape 1 – V-shape 2 – Open
MinDepth	64f	Minimum depth.
MinWidth	64f	Minimum width.
MaxWidth	64f	Maximum width.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
Measurements\X	Groove tool measurement	X measurement.
Measurements\Z	Groove tool measurement	Z measurement.
Measurements\Width	Groove tool measurement	Width measurement.
Measurements\Depth	Groove tool measurement	Depth measurement.

Groove 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

Element	Type	Description
		1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
SelectType	32s	Method of selecting a groove when multiple grooves are found: 0 – Max depth 1 – Ordinal, from left 2 – Ordinal, from right
SelectIndex	32s	Index when SelectType is set to 1 or 2.
Location (X and Z measurements only)	32s	Setting for groove location to return from: 0 – Bottom 1 – Left corner 2 – Right corner

ProfileIntersect

A ProfileIntersect element defines settings for a profile intersect tool and one or more of its measurements.

ProfileIntersect Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfileIntersect</i> above.
Source	32s	Profile source.
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X measurements (IDs) available for anchoring.

Element	Type	Description
Anchor\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
RefType	32s	Reference line type: 0 – Fit 1 – X Axis
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RefLine	ProfileLine	Definition of reference line. Ignored if RefType is not 0.
Line	ProfileLine	Definition of line.
Measurements\X	Intersect tool measurement	X measurement.
Measurements\Z	Intersect tool measurement	Z measurement.
Measurements\Angle	Intersect tool measurement	Angle measurement.
Features\IntersectPoint	GeometricFeature	IntersectPoint PointFeature.
Features\Line	GeometricFeature	Line LineFeature.
Features\BaseLine	GeometricFeature	BaseLine LineFeature.
Intersect 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

Element	Type	Description
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
Absolute (Angle measurement only)	Boolean	Setting for selecting the angle range: 0 – A range of -90 to 90 degrees is used. 1 – A range of 0 to 180 degrees is used.

ProfileLine

A ProfileLine element defines settings for a profile line tool and one or more of its measurements.

ProfileLine Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfileLine</i> above.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Measurement region.
FittingRegions	ProfileLine	ProfileLine describing up to 2 regions to fit to.
FittingRegionsEnabled	Bool	Whether the fitting regions are enabled.
Measurements\StdDev	Line tool	StdDev measurement.

Element	Type	Description
	measurement	
Measurements\MaxError	Line tool measurement	MaxError measurement.
Measurements\MinError	Line tool measurement	MinError measurement.
Measurements\Percentile	Line tool measurement	Percentile measurement.
Measurements\Offset	Line tool measurement	Offset measurement.
Measurements\Angle	Line tool measurement	Angle measurement.
Measurements\MinErrorX	Line tool measurement	Minimum Error in X measurement.
Measurements\MinErrorZ	Line tool measurement	Minimum Error in Z measurement.
Measurements\MaxErrorX	Line tool measurement	Maximum Error in X measurement.
Measurements\MaxErrorZ	Line tool measurement	Maximum Error in Z measurement.
Features\Line	GeometricFeature	Line LineFeature.
Features\ErrorMinPoint	GeometricFeature	ErrorMinPoint PointFeature.
Features\ErrorMaxPoint	GeometricFeature	ErrorMaxPoint PointFeature.

Line 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements enable state 0 – Disable 1 – Enable

Element	Type	Description
SmoothingWindow	32u	Smoothing window.
Scale	64f	Output scaling factor.
Offset	64f	Output offset factor.
DecisionMin	64f	Minimum decision threshold.
DecisionMax	64f	Maximum decision threshold.
Percent	64f	Error percentile.
<i>(Percentile measurement only)</i>		

ProfilePanel

A ProfilePanel element defines settings for a profile panel tool and one or more of its measurements.

ProfilePanel Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RefSide	32s	Setting for reference side to use.
MaxGapWidth	64f	Setting for maximum gap width (mm).
LeftEdge	ProfilePanelEdge	Element for left edge configuration.
RightEdge	ProfilePanelEdge	Element for right edge configuration.
Measurements\Gap	Gap/Flush measurement	Gap measurement.
Measurements\Flush	Gap/Flush measurement	Flush measurement.
Measurements\LeftGapX	Gap/Flush measurement	Left Gap X measurement.
Measurements\LeftGapZ	Gap/Flush	Left Gap Z measurement.

Element	Type	Description
	measurement	
Measurements\LeftFlushX	Gap/Flush measurement	Left Flush X measurement.
Measurements\LeftFlushZ	Gap/Flush measurement	Left Flush Z measurement.
Measurements\LeftSurfaceAngle	Gap/Flush measurement	Left Surface Angle measurement.
Measurements\RightGapX	Gap/Flush measurement	Right Gap X measurement.
Measurements\RightGapZ	Gap/Flush measurement	Right Gap Z measurement.
Measurements\RightFlushX	Gap/Flush measurement	Right Flush X measurement.
Measurements\RightFlushZ	Gap/Flush measurement	Right Flush Z measurement.
Measurements\RightSurfaceAngle	Gap/Flush measurement	Right Surface Angle measurement.

ProfilePanelEdge

Element	Type	Description
EdgeType	32s	Edge type: 0 – Tangent 1 – Corner
MinDepth	64f	Minimum depth.
MaxVoidWidth	64f	Maximum void width.
SurfaceWidth	64f	Surface width.
SurfaceOffset	64f	Surface offset.
NominalRadius	64f	Nominal radius.
EdgeAngle	64f	Edge angle.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Edge region.

Gap/Flush 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

Element	Type	Description
		1 – Enable
HoldEnabled	Boolean	Output hold enable state: 0 – Disable 1 – Enable
SmoothingEnabled	Boolean	Smoothing enable state: 0 – Disable 1 – Enable
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
Axis (Gap measurement only)	32s	Measurement axis: 0 – Edge 1 – Surface 2 – Distance
Absolute (Flush measurement only)	Boolean	Setting for selecting absolute or signed result: 0 – Signed 1 – Absolute

ProfilePosition

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

ProfilePosition Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Collection of geometric feature outputs available in the tool. See <i>ProfilePosition</i> above.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.

Element	Type	Description
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
Feature	ProfileFeature	Element for feature detection.
Measurements\X	Position tool measurement	X measurement.
Measurements\Z	Position tool measurement	Z measurement.
Features\Point	GeometricFeature	Point PointFeature

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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileRoundCorner

A ProfileRoundCorner element defines settings for a profile round corner tool and one or more of its measurements.

ProfileRoundCorner Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile 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\Z	String (CSV)	The Z measurements (IDs) used for anchoring.
Anchor\Z.options	String (CSV)	The Z measurements (IDs) available for anchoring.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
RefDirection	32s	Setting for reference side to use: 0 – Left 1 – Right
Edge	ProfilePanelEdge	Element for edge configuration
Measurements\X	Round Corner tool measurement	X measurement.
Measurements\Z	Round Corner tool measurement	Z measurement.
Measurements\Angle	Round Corner tool measurement	Angle measurement.
Features\CenterPoint	Geometric Feature	Circle Center PointFeature.
Features\EdgePoint	Geometric Feature	Edge PointFeature.

ProfilePanelEdge

Element	Type	Description
EdgeType	32s	Edge type: 0 – Tangent 1 – Corner
MinDepth	64f	Minimum depth.
MaxVoidWidth	64f	Maximum void width.
SurfaceWidth	64f	Surface width.
SurfaceOffset	64f	Surface offset.
NominalRadius	64f	Nominal radius.

Element	Type	Description
EdgeAngle	64f	Edge angle.
RegionEnabled	Bool	Whether or not to use the region. If the region is disabled, all available data is used.
Region	ProfileRegion2d	Edge region.

Round Corner 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.

ProfileStrip

A ProfileStrip element defines settings for a profile strip tool and one or more of its measurements.

The profile strip tool is dynamic, meaning that it can contain multiple measurements of the same type in the Measurements element.

ProfileStrip Child Elements

Element	Type	Description
Name	String	Tool name.
Features	Collection	Not used.
Source	32s	Profile source.

Element	Type	Description
Anchor\X	String (CSV)	The X measurements (IDs) used for anchoring.
Anchor\X.options	String (CSV)	The X 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.
StreamOptions	Collection	A collection of <i>StreamOptions</i> on page 254 elements.
Stream\Step	32s	The stream source step. Possible values are: 1 – Video 2 – Range 3 – Surface 4 – Section
Stream\ld	32u	The stream source ID.
BaseType	32s	Setting for the strip type: 0 – None 1 – Flat
LeftEdge	Bitmask	Setting for the left edge conditions: 1 – Raising 2 – Falling 4 – Data End 8 – Void
RightEdge	Bitmask	Setting for the right edge conditions: 1 – Raising 2 – Falling 4 – Data End 8 – Void
TiltEnabled	Boolean	Setting for tilt compensation: 0 – Disabled 1 – Enabled
SupportWidth	64f	Support width of edge (mm).
TransitionWidth	64f	Transition width of edge (mm).
MinWidth	64f	Minimum strip width (mm).
MinHeight	64f	Minimum strip height (mm).
MaxVoidWidth	64f	Void max (mm).
Region	ProfileRegion2d	Region containing the strip.
Measurements\X	Strip tool measurement	X measurement.
Measurements\Z	Strip tool	Z measurement.

Element	Type	Description
	measurement	
Measurements\Width	Strip tool measurement	Width measurement.
Measurements\Height	Strip tool measurement	Width measurement.

Strip 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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
SelectType	32s	Method of selecting a groove when multiple grooves are found: 0 – Best 1 – Ordinal, from left 2 – Ordinal, from right
SelectIndex	32s	Index when SelectType is set to 1 or 2.
Location	32s	Setting for groove location to return from: (X, Z, and Height measurements only) 0 – Left 1 – Right 2 – Center

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.

Tool

A Tool element of type FeatureDimension defines settings for a feature dimension tool and one or more of its measurements.

Tool Child Elements

Element	Type	Description
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
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.
Parameters\RefPoint	GdkParamGeometricFeature	Reference point feature.
Parameters\Feature	GdkParamGeometricFeature	Reference feature.
Measurements\Measurement @type=Width	Dimension Measurement	Width measurement.
Measurements\Measurement @type=Length	Dimension Measurement	Length measurement.
Measurements\Measurement @type=Height	Dimension Measurement	Width measurement.
Measurements\Measurement @type=Distance	Dimension Measurement	Distance measurement.
Measurements\Measurement @type=PlaneDistance	Dimension Measurement	Plane distance measurement.

Dimension Measurement Child Elements

@id	32s	Measurement ID. Optional (measurement disabled if not set).
@type	String	Type name of measurement.
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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
Parameters\WidthAbsolute (Width measurement only)	GdkParamBool	Absolute width enabled boolean.
Parameters\LengthAbsolute (Length measurement only)	GdkParamBool	Absolute length enabled boolean.
Parameters\HeightAbsolute (Height measurement only)	GdkParamBool	Absolute height enabled boolean.

Tool

A Tool element of type FeatureIntersect defines settings for a feature intersection tool and one or more of its measurements.

Tool Child Elements

Element	Type	Description
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
Name	String	Tool name.
Source	32s	Surface source.

Element	Type	Description
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.
Parameters\Line	GdkParamGeometricFeature	Line feature input.
Parameters\RefLine	GdkParamGeometricFeature	Reference line feature input.
Measurements\Measurement @type=X	Intersect Measurement	X measurement.
Measurements\Measurement @type=Y	Intersect Measurement	Y measurement.
Measurements\Measurement @type=Z	Intersect Measurement	Z measurement.
Measurements\Measurement @type=Angle	Intersect Measurement	Angle measurement.
Features\IntersectPoint	GDK Feature	Intersect point feature.

Intersect Measurement Child Elements

@id	32s	Measurement ID. Optional (measurement disabled if not set).
@type	String	Type name of measurement.
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
PreserveInvalidsEnabled	Boolean	Preserve invalid measurements 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.
Parameters\AngleRange	GdkParamInt	Angle range option choice. Is one of: 0 – -180 To 180 1 – 0 To 360

Custom

A Custom element defines settings for a user-created GDK-based tool and one or more of its measurements.

Custom Child Elements

Element	Type	Description
@type	String	Type name of the tool.
@version	String	Version string for custom tool.
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.
Parameters	GDK Parameter	Collection of parameters . The element name in the job file is the name of the parameter.
Measurements	GDK Measurement	Collection of measurements .
Features	GDK Feature	Collection of features .

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 (Range Position Z) will be sent to output.

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

```
<Thickness id="3"> ...
```

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

Ethernet

The Ethernet element defines settings for Ethernet output.

In the Ethernet element, the source identifiers used for video, range, profile, and surface output, as well as range, profile, and surface intensity outputs, correspond to the *sensor* that provides the data. For example, in the XML below, the *options* attribute of the Ranges 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="0,1">0</Ranges>
    <Profiles options=""/>
    <Surfaces options=""/>
    ...
```

Ethernet Child Elements

Element	Type	Description
Protocol	32s	Ethernet protocol: 0 – Gocator 1 – Modbus 2 – EtherNet/IP 3 – ASCII
TimeoutEnabled	Boolean	Enable or disable auto-disconnection timeout. Applies only to the Gocator protocol.
Timeout	64f	Disconnection timeout (seconds). Used when TimeoutEnabled is true and the Gocator protocol is selected.
Ascii	Section	See <i>Ascii</i> on page 285.
EIP	Section	See <i>EIP</i> on page 285.
Modbus	Section	See <i>Modbus</i> on page 286.
Videos	32s (CSV)	Selected video sources: 0 – Top 1 – Bottom 2 – Top left 3 – Top right

Element	Type	Description
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 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).
SurfaceSections	32s (CSV)	Selected surface section sources.
SurfaceSections.options	32s (CSV)	List of available surface section sources.
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.
SurfaceIntensities.options	32s (CSV)	List of available surface intensity sources (see above).
SurfaceSectionIntensities	32s (CSV)	Selected surface section intensity sources

Element	Type	Description
SurfaceSectionIntensities.options	32s (CSV)	List of available surface section intensity sources.
Tracheids	32s (CSV)	Selected tracheid sources.
Tracheids.options	32s (CSV)	List of available tracheid sources.
Measurements	32u (CSV)	Selected measurement sources.
Measurements.options	32u (CSV)	List of available measurement sources.
Events	32u (CSV)	Selected events
Events.Options	32u (CSV)	CSV list of possible event options: 0 – Exposure Begins 1 – Exposure Ends
Features	32u (CSV)	Selected feature sources.
Features.options	32u (CSV)	List of available feature sources.
ToolData	32u (CSV)	Selected tool data sources.
ToolData.options	32u (CSV)	List of available tool data sources.

Ascii

Ascii Child Elements

Element	Type	Description
Operation	32s	Operation mode: 0 – Asynchronous 1 – Polled
ControlPort	32u	Control service port number.
HealthPort	32u	Health service port number.
DataPort	32u	Data service port number.
Delimiter	String	Field delimiter.
Terminator	String	Line terminator.
InvalidValue	String	String for invalid output.
CustomDataFormat	String	Custom data format.
CustomFormatEnabled	Bool	Enables custom data format.
StandardFormatMode	32u	The formatting mode used if not a custom format: 0 – Standard 1 – Standard with Stamp

EIP

EIP Child Elements

Element	Type	Description
BufferEnabled	Bool	Enables EtherNet/IP output buffering.

Element	Type	Description
EndianOutputType	32s	Endian output type: 0 – Big endian 1 – Little endian
ImplicitOutputEnabled	Bool	Enables Implicit (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 – Acquisition start 5 – Acquisition end
SignalType	32s	Signal type: 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).

Element	Type	Description
DelayDomain	32s	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Inverted	Bool	Whether the sent bits are flipped.
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.



Gocator 1300 series sensors are limited to sending data at 10 kHz over the analog output channel.

Therefore, if you configure a sensor so that it runs at a speed higher than 10 kHz, and configure a measurement to be sent on the analog channel, you will get analog data drops.

To achieve a 10 kHz analog output rate, you must enable and configure scheduled output.

Analog Child Elements

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

Element	Type	Description
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 <i>Selcom</i> below.
Ascii	Section	See <i>Ascii</i> 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 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.
Delay	64u	Output delay in μs.

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.
StandardFormatMode	32u	The formatting mode used if not a custom format: 0 – Standard 1 – Standard with Stamp

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 both the Main and the Buddy sensor.

Transform Child Elements

Element	Type	Description
@version	32u	Major transform version (100).
@versionMinor	32u	Minor transform version (0).
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.

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](#)
- [Selcom](#)

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:

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



The Gocator 4.x firmware uses mm, mm², mm³, 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, mm²/1000, mm³/1000, 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 *GoSDK* on page 377.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 190.

For information on job file structures (for example, if you wish to create job files programmatically), see *Job File Structure* on page 227.

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	-
Point64f	Two 64-bit floating point values	-
Point3d64f	Three 64-bit floating point values	-
Rect64f	Four 64-bit floating point values	-
Rect3d64f	Eight 64-bit floating point values	-

Commands

The following sections describe the commands available on the Discovery (page 293), Control (page 296), and Upgrade (page 332) 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.
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 sensor 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	64s	0	Command length.
type	64s	8	Command type (0x1).

Field	Type	Offset	Description
signature	64s	16	Message signature (0x0000504455494D4C)
deviceld	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.

Reply

Field	Type	Offset	Description
length	64s	0	Reply length.
type	64s	8	Reply type (0x1001).
status	64s	16	Operation status.
signature	64s	24	Message signature (0x0000504455494D4C)
deviceld	64s	32	Serial number.
dhcpEnabled	64s	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	64s	0	Command length.
type	64s	8	Command type (0x2).
signature	64s	16	Message signature (0x0000504455494D4C)
deviceld	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.
dhcpEnabled	64s	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.

Field	Type	Offset	Description
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	64s	0	Reply length.
type	64s	8	Reply type (0x1002).
status	64s	16	Operation status. For a list of status codes, see <i>Commands</i> on page 292.
signature	64s	24	Message signature (0x0000504455494D4C).
deviceld	64s	32	Serial number.

Get Info

The Get Info command is used to retrieve sensor information.

Command

Field	Type	Offset	Description
length	64s	0	Command length.
type	64s	8	Command type (0x5).
signature	64s	16	Message signature (0x0000504455494D4C).
deviceld	64s	24	Serial number of the device whose address information is queried. 0 selects all devices.

Reply

Field	Type	Offset	Description
length	64s	0	Reply length.
type	64s	8	Reply type (0x1005).
status	64s	16	Operation status. For a list of status codes, see <i>Commands</i> on page 292.
signature	64s	24	Message signature (0x0000504455494D4C).
attrCount	16u	32	Byte count of the attributes (begins after this field and ends before propertyCount).
id	32u	34	Serial number.
version	32u	38	Version as a 4-byte integer (encoded in little-endian).
uptime	64u	42	Sensor uptime (microseconds).
ipNegotiation	byte	50	IP negotiation type: 0 – Static 1 – DHCP

Field	Type	Offset	Description
addressVersion	byte	51	IP address version (always 4).
address[4]	byte	52	IP address.
reserved[12]	byte	56	Reserved.
prefixLength	32u	68	Subnet prefix length (in number of bits).
gatewayVersion	byte	72	Gateway address version (always 4).
gatewayAddress[4]	byte	73	Gateway address.
reserved[12]	byte	77	Reserved.
controlPort	16u	89	Control channel port.
upgradePort	16u	91	Upgrade channel port.
healthPort	16u	93	Health channel port.
dataPort	16u	95	Data channel port.
webPort	16u	97	Web server port.
propertyCount	8u	99	Number of sensor ID properties.
properties [propertyCount]	Property	100	List of sensor ID properties.

Property

Field	Type	Description
nameLength	8u	Length of the name.
name[nameLength]	char	Name string.
valueLength	8u	Length of the value.
value[valueLength]	char	Value string.

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. For more information on Upgrade commands, see *Upgrade Commands* on page 332.

States

A Gocator system can be in one of three states: Conflict, 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.

The state of the sensor can be retrieved using the [Get States](#) or [Get System Info](#) command.

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

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 <i>Commands</i> on page 292.
majorVersion	8u	10	Major version.
minorVersion	8u	11	Minor version.

Get Address

The Get Address command is used to get a sensor address.

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 <i>Commands</i> on page 292.
dhcpEnabled	byte	10	0 – DHCP not used 1 – DHCP used

Field	Type	Offset	Description
address[4]	byte	11	IP address (most significant byte first).
subnetMask[4]	byte	15	Subnet mask.
gateway[4]	byte	19	Gateway address.

Set Address

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

Command

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

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x3013).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Get System Info V2

The Get System Info command reports information about the local node, remote nodes and assigned buddies.

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 *Start Upgrade* on page 333). 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 80.

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 (0x4010)

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4010).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.
localInfoSize	16u	10	Size of localInfo structure. Current: 52.
localInfo	Device Info	12	Info for this device.
remoteCount	32u	-	Number of discovered sensors.
remoteInfoSize	16u	-	Size of remoteInfo structure. Current 60.
remoteInfo [remoteCount]	Remote Info	-	List of info for discovered sensors.
buddyInfoCount	32u	-	Number of buddies assigned (can be 0).
buddyInfoSize	16u	-	Size of buddyInfo structure. Current: 8.
Buddies[buddyCount]	Buddy Info	-	List of info for the assigned buddies.

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 -1 – Conflict 0 – Ready 1 – Running For more information on states, see <i>Control Commands</i> on page 296.
role	32s	48	Sensor role 0 – Main 1 – Buddy

Remote Info

Field	Type	Offset	Description
deviceId	32u	0	Serial number of the device.

Field	Type	Offset	Description
address[4]	byte	4	IP address (most significant byte first).
modelName[32]	char	8	Model name.
firmwareVersion[4]	byte	40	Firmware version (most significant byte first).
state	32s	44	Sensor state -1 – Conflict 0 – Ready 1 – Running For more information on states, see <i>Control Commands</i> on page 296.
role	32s	48	Sensor role 0 – Main 1 – Buddy
mainId	32u	52	Serial number of the main device, or zero.
buddiableStatus	32s	56	Whether or not the device can be buddied: 1 – Can be buddied Errors: 0 – Unbuddiable (General Error) -100 – Already buddied -99 – Invalid State (e.g. running) -98 – Version Mismatch -97 – Model Mismatch

Buddy Info

Field	Type	Offset	Description
deviceId	32u	2	Serial number of the device.
state	k32s	6	Buddy state 2 - Connecting 1 – Connected Errors: 0 – Unbuddiable (General Error) -100 – Already buddied -99 – Invalid State (e.g. running) -98 – Version Mismatch -97 – Model Mismatch -95 – Device Missing

Get System Info



This version of the Get System Info command is deprecated. Use [Get System Info \(v2\)](#) instead.

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 *Start Upgrade* on page 333). 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 80.

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 <i>Commands</i> on page 292.
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 -1 – Conflict 0 – Ready 1 – Running

Field	Type	Offset	Description
			For more information on states, see <i>Control Commands</i> on page 296.
role	32s	48	Sensor role 0 – Main 1 – Buddy
buddyId	32s	52	Serial number of paired device (main or buddy). 0 if unpaired.

Get States

The Get 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 <i>Commands</i> on page 292.
count	32u	10	Number of state variables.
sensorState	32s	14	Sensor state -1 – Conflict 0 – Ready 1 – Running For more information on states, see <i>Control Commands</i> on page 296.
loginState	32s	18	Device login state 0 – No user 1 – Administrator 2 – Technician
alignmentReference	32s	22	Alignment reference 0 – Fixed 1 – Dynamic
alignmentState	32s	26	Alignment state 0 – Unaligned 1 – Aligned
recordingEnabled	32s	30	Whether or not recording is enabled

Field	Type	Offset	Description
			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 <i>Commands</i> on page 292.

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

Field	Type	Offset	Description
			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 <i>Commands</i> on page 292.



Passwords can only be changed if a user is logged in as an administrator.

Assign Buddies

The Assign Buddies command is used to set the list of buddies assigned to the system.

This command can be used to both add and remove buddies by changing the list of buddies. A serial number of 0 can be used to add device slots that are not assigned a physical sensor. Collections associated with the devices (e.g. <Device> element in the configuration) grow or shrink accordingly. Items are added to or removed from the end of these collections. For example: the system starts with 2 devices, [A, B]. A new list [A, B, C] is sent. The configuration for A and B are preserved, and a new record is created for C. If now the system changes back to [A, B], the record for C is deleted. Adding or removing items in the middle of the list has the same behaviour. Example: the system starts with 3 devices, [A, B, C]. A new list [A, C] is sent. The configuration for B is now used for C, and the configuration for C is deleted. To ensure consistency when adding and removing devices, add only to the end of the list and remove using the Remove Buddies command.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4011).
buddyCount	32u	6	Number of buddies or 0 to unbuddy all devices.
buddies[buddyCount]	32u	10	Serial Numbers of the buddies to assign (can be 0).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4011).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Remove Buddies

The Remove Buddies command is used to remove one or more buddies using 0-based buddy indices.

Use this command to remove a buddy devices along with its associated configuration resources. If the system starts with 3 devices: [A, B, C], and this command is called to remove B, the configuration items for A and C remain unchanged.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4013).
buddyCount	32u	6	Number of buddies.
buddyIds[buddyCount]	32u	10	Indices of the buddies to remove. Note that the first buddy has index 0 (i.e. it's the index of buddies, not all devices including the main).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4013).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Set Buddy

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

Command

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

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4005).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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; for more information, see *Job File Structure* on page 227).

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	32u	0	Command size including this field, in bytes.
id	16u	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 <i>Commands</i> on page 292.

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 File Structure* on page 227).

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 including this field, 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 including this field, in bytes.
id	16u	4	Reply identifier (0x1007).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.
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 File Structure* on page 227).

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 <i>Commands</i> on page 292.

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 File Structure* on page 227).

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 <i>Commands</i> on page 292.

User Storage Used

The User Storage Used command returns the amount of user storage that is used.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1021).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1021).
status	32s	6	Reply status.
spaceUsed	64u	10	The used storage space in bytes.

User Storage Free

The User Storage Free command returns the amount of user storage that is free.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x1022).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x1022).
status	32s	6	Reply status.
spaceFree	64u	10	The free storage space in bytes.

Get Default Job

The Get Default Job command gets the name of the job the sensor loads when it powers up.

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 <i>Commands</i> on page 292.
name[64]	char	10	The file name (null-terminated) of the job the sensor loads when it powers up.

Set Default Job

The Set Default Job command sets the job the sensor loads when it powers up.

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 the sensor loads when it powers up.

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
timestamp	64u	10	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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.

Start

The Start command starts the sensor system (system enters the Running state). For more information on states, see *Control Commands* on page 296.

Command

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

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Reply identifier (0x100D).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

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 296.

Command

Field	Type	Offset	Description
length	32u	0	Command size – in bytes.
id	16u	4	Command identifier (0x100F).
target	64s	6	Target scheduled start value (in ticks or μ s, depending on the trigger type).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size – in bytes.
id	16u	4	Reply identifier (0x100F).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Stop

The Stop command stops the sensor system (system enters the Ready state). For more information on states, see *Control Commands* on page 296.

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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 296).

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 <i>Commands</i> on page 292.

Get Voltage Settings

The Get Voltage Settings command returns the sensor's voltage and cable length settings.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4539).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4539).
Voltage	k16u	10	0: 48 Volts; 1: 24 Volts.
Cable Length	k16u	12	0 – 100: Meters

Set Voltage Settings

The Set Voltage Settings command sets the sensor's voltage and cable length settings.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4538).
Voltage	k16u	6	0: 48 Volts; 1: 24 Volts.
Cable Length	k16u	8	0 – 100: Meters

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4538).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Get Quick Edit Enabled

The Get Quick Edit Enabled command returns whether Quick Edit mode is enabled on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4541).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4541).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.
Enable	8u	10	0: disabled; 1: enabled.

Set Quick Edit Enabled

The Set Quick Edit Enabled command enables or disables Quick Edit mode on the sensor.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4540).
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 (0x4540).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
opId	32u	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 <i>Commands</i> on page 292.
opId	32u	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 <i>Commands</i> on page 292.

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.

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: 0 – Set to low (continuous)

Field	Type	Offset	Description
			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 <i>Commands</i> on page 292.

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.

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 (microamperes).

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 <i>Commands</i> on page 292.



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 <i>Commands</i> on page 292.



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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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	6	Data length.
data[length]	byte	10	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 <i>Commands</i> on page 292.

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
resetip	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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.

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).

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 <i>Commands</i> on page 292.
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.

Field	Type	Offset	Description
id	16u	4	Reply identifier (0x4503).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.



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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.

Read Live Log

The Read Live Log command returns an XML file containing the log messages between the passed start and end indexes.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x101F).
Start	32u	6	First log to read
End	32u	10	Last log to read

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x101F).
status	32s	6	Reply status.
length	32u	10	File length
data[length]	byte	14	XML Log File

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.

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 <i>Commands</i> on page 292.



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 <i>Commands</i> on page 292.

	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 <i>Commands</i> on page 292.

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	16u	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 <i>Commands</i> on page 292.

Add Tool

The Add Tool command adds a tool to the live job.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4530).
typeName[64]	char	6	Type name of the tool (e.g., ProfilePosition)
name[64]	char	70	User-specified name for tool instance

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4530).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Add Measurement

The Add Measurement command adds a measurement to a tool instance.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4531).
toolIndex	32u	6	Index of the tool instance the new measurement is added to.
typeName[64]	char	10	Type name of the measurement (for example, X).
name[64]	char	74	User-specified name of the measurement instance.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4531).
status	32s	6	Reply status. For a list of status codes, see <i>Commands</i> on page 292.



This command can only be used with dynamic tools (tools with a dynamic list of measurements). The maximum number of instances for a given measurement type can be found in the [ToolOptions](#) node. For dynamic tools, the maximum count is greater than one, while for static tools it is one.

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
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).

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
data[size]	byte	22	Chunk data.



All recorded range or profile data 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).
type	32s	6	Data type: 0 – Range or video 1 – Intensity
source	32s	10	Data source to export.

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 <i>Commands</i> on page 292.
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 <i>Commands</i> on page 292.
progressTotal	32u	10	Progress indicating completion (100%).
progress	32u	14	Current progress.
size	32u	18	Size of the chunk in bytes.
data[size]	byte	22	Chunk data.

Get Runtime Variable Count

The Get Runtime Variable Count command gets the number of runtime variables that can be accessed.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4537).

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4537).
status	32s	6	Reply status.
valueLength	32u	10	The count of runtime variables.

Set Runtime Variables

The Set Runtime Variables command sets the runtime variables at the given index for the given length.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.

Field	Type	Offset	Description
id	16u	4	Command identifier (0x4536).
index	32u	6	The starting index of the variables to set.
length	32u	10	The number of values to set from the starting index.
values[length]	32s	14	The runtime variable values to set.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4536).
status	32s	6	Reply status.

Get Runtime Variables

The Get Runtime Variables command gets the runtime variables for the given index and length.

Command

Field	Type	Offset	Description
length	32u	0	Command size including this field, in bytes.
id	16u	4	Command identifier (0x4535).
index	32u	6	The starting index of the variables to retrieve.
length	32u	10	The number of values to retrieve from the starting index.

Reply

Field	Type	Offset	Description
length	32u	0	Reply size including this field, in bytes.
id	16u	4	Reply identifier (0x4535).
status	32s	6	Reply status.
index	32u	10	The starting index of the variables being returned.
length	32u	14	The number of values being returned.
values[length]	32s	18	The runtime variable values.

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. For more information on Control commands, see *Control Commands* on page 296.

After connecting to a Gocator sensor, you can use the [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 including this field, in bytes.
id	64s	8	Command identifier (0x0000).
length	64s	16	Length of the upgrade package (bytes).
data[length]	byte	24	Upgrade package data.

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x0000).
status	64s	16	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

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 including this field, in bytes.
id	64s	8	Command identifier (0x0003).
skipValidation	64s	16	Whether or not to skip validation (0 – do not skip, 1 – skip).
length	64s	24	Length of the upgrade package (bytes).
data[length]	byte	32	Upgrade package data.

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x0003).
status	64s	16	Reply status. For a list of status codes, see <i>Commands</i> on page 292.

Get Upgrade Status

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

Command

Field	Type	Offset	Description
length	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x1)

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x1).
status	64s	16	Reply status. For a list of status codes, see <i>Commands</i> on page 292.
state	64s	24	Upgrade state: -1 – Failed 0 – Completed 1 – Running 2 – Completed, but should run again
progress	64s	32	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	64s	0	Command size including this field, in bytes.
id	64s	8	Command identifier (0x2)

Reply

Field	Type	Offset	Description
length	64s	0	Reply size including this field, in bytes.
id	64s	8	Reply identifier (0x2).
status	64s	16	Reply status. For a list of status codes, see <i>Commands</i> on page 292.
length	64s	24	Length of the log (bytes).
log[length]	char	32	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 342.

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).
source	8u	12	Source (0 – Main, 1 – Buddy).
reserved	8u	13	Reserved.
stamps[C]	Stamp	14	Array of stamps (see below).

Stamp

Field	Type	Offset	Description
frameIndex	64u	0	Frame index (counts up from zero).
timestamp	64u	8	Timestamp (µs).
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. (In a dual-sensor system, the serial number of the main sensor.)
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 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).
flippedX	8u	26	Indicates whether the video data must be flipped horizontally to match up with profile data.
flippedY	8u	27	Indicates whether the video data must be flipped vertically to match up with profile data.
pixels[H][W]	(Variable)	28	Image pixels. (Depends on pixelSize above.)

Range

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

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

Range Intensity

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

Profile Point Cloud

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 5.

Field	Type	Offset	Description
attributeSize	16u	6	Size of attributes, in bytes (min: 32, current: 32).
count (C)	32u	8	Number of profile arrays.
width (W)	32u	12	Number of points per profile array.
xScale	32u	16	X scale (nm).
zScale	32u	20	Z scale (nm).
xOffset	32s	24	X offset (μm).
zOffset	32s	28	Z offset (μm).
Source	8u	32	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	33	Exposure (ns).
cameraIndex	8u	37	Camera index.
reserved[2]	8u	38	Reserved.
ranges[C][W]	Point16s	40	Profile ranges.

Profile 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 7.
attributesSize	16u	6	Size of attributes, in bytes (min: 24, current: 24).
count (C)	32u	8	Number of profile intensity arrays.
width (W)	32u	12	Number of points per profile intensity array.
xScale	32u	16	X scale (nm).
xOffset	32s	20	X offset (μm).
source	8u	24	Source 0 – Top 1 – Bottom 2 – Top Left 3 – Top Right
exposure	32u	25	Exposure (ns).
cameraIndex	8u	29	Camera index.
reserved[2]	8u	30	Reserved.

Field	Type	Offset	Description
streamStep	32s	31	Stream step.
streamStepId	32s	35	Stream ID.
points[C][W]	8u	39	Intensity arrays.

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	14	Array of measurements (see below).

Measurement

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

Alignment Result

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

Field	Type	Offset	Description
			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 -7 – Too few profiles in target 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.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 8).
opId	32u	8	Operation ID.
status	32s	12	Operation status.
exposure	32u	16	Exposure result (ns).

Event

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 22.
attributesSize	16u	6	Size of attributes, in bytes (min: 8, current: 8).
eventType	32u	8	- The type of event: 0 – Exposure Begin 1 – Exposure End
length	32u	12	The number of bytes containing additional data.
data[length]	8u	16	Additional data.

Feature Point

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 24.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10^6)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10^6)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10^6)

Feature Line

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 25.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10^6)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10^6)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10^6)
Direction.x	64s	32	X Component of Direction Vector (Scaled by 10^6)
Direction.y	64s	40	Y Component of Direction Vector (Scaled by 10^6)
Direction.z	64s	48	Z Component of Direction Vector (Scaled by 10^6)

Feature Plane

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 26.
id	16u	6	Feature Id
Normal.x	64s	8	X Component of Normal Vector (Scaled by 10^6)
Normal.y	64s	16	Y Component of Normal Vector (Scaled by 10^6)
Normal.z	64s	24	Z Component of Normal Vector (Scaled by 10^6)
originDistance	64s	32	Distance to Origin (Scaled by 10^6)

Feature Circle

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 27.
id	16u	6	Feature Id
Point.x	64s	8	X Coordinate of Point (Scaled by 10 ⁶)
Point.y	64s	16	Y Coordinate of Point (Scaled by 10 ⁶)
Point.z	64s	24	Z Coordinate of Point (Scaled by 10 ⁶)
Normal.x	64s	32	X Component of Normal Vector (Scaled by 10 ⁶)
Normal.y	64s	40	Y Component of Normal Vector (Scaled by 10 ⁶)
Normal.z	64s	48	Z Component of Normal Vector (Scaled by 10 ⁶)
radius	64s	56	Radius of Circle (Scaled by 10 ⁶)

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 334.

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

Field	Type	Offset	Description
size	32u	0	Count of bytes in message (including this field).

Field	Type	Offset	Description
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.
source	8u	10	Source (0 – Main, 1 – Buddy).
reserved[3]	8u	11	Reserved
indicators[C]	Indicator	14	Array of indicators (see format below).

The 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 <i>Indicator identifiers</i> below table below).
instance	32u	4	Indicator instance.
value	64s	8	Value (identifier-specific meaning).

The following health indicators are defined for Gocator sensor systems.

	When a sensor is accelerated, some health indicators report values from the <i>PC</i> that is accelerating the sensor, or a combination of both. In the table below, values are reported from the sensor unless otherwise indicated.
	Undocumented indicators may be included in addition to the indicators defined below.

Indicator identifiers

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.
Internal Temperature	2002	-	Internal temperature (centidegrees Celsius).
Uptime	2017	-	Time elapsed since node boot-up or reset (seconds).
Projector Temperature	2404	-	Projector module temperature (centidegrees Celsius). Only available on projector based devices.
Control Temperature	2028	-	Control module temperature (centidegrees Celsius). Available only on 3B-class devices.
Memory Usage	2003	-	Amount of memory currently used (bytes).
Memory Capacity	2004	-	Total amount of memory available (bytes).

Indicator	ID	Instance	Value
Storage Usage	2005	-	Amount of non-volatile storage used (bytes).
Storage Capacity	2006	-	Total amount of non-volatile storage available (bytes).
Alignment State	20008	-	Alignment state: 0 – not aligned 1 - aligned
CPU Usage	2007	-	CPU usage (percentage of maximum).
Net Out Capacity	2009	-	Total available outbound network throughput (bytes/s).
Net Out Link Status	2034	-	Current Ethernet link status.
Sync Source*	2043	-	Gocator synchronization source. 1 - FireSync Master device 2 - Sensor
Digital Inputs*	2024	-	Current digital input status (one bit per input).
Event Count	2102	-	Total number of events triggered.
Camera Search Count	2217	-	Number of search states. (Only important when tracking is enabled.)
Camera Trigger Drops	2201	-	Number of dropped triggers.
Analog Output Drops	21014 (previously 2501)	Output Index	Number of dropped outputs.
Digital Output Drops	21015 (previously 2601)	Output Index	Number of dropped outputs.
Serial Output Drops	21016 (previously 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.
Master Status*	20006	0 for main	Master connection status:

Indicator	ID	Instance	Value
		1 for buddy	0 – Not connected 1 – Connected The indicator with instance = buddy does not exist if the buddy is not connected.
Cast Start State*	20007		The state of the second digital input. (NOTE: Only available on XLine capable licensed devices)
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 Processing Latency	21001	-	Last delay from camera exposure to availability of all results.
Max Processing Latency	21002	-	Maximum value of processing latency.
Ethernet Output	21003	-	Number of bytes transmitted.
Ethernet Rate	21004	-	The average number of bytes per second being transmitted.
Ethernet Drops	21005	-	Number of dropped Ethernet packets.
Trigger Drops**	21010		Number of dropped triggers. The sum of various triggering-related drop indicators.
Output Drops**	21011		Number of dropped output data. The sum of all output drops (analog, digital, serial, host server, and ASCII server).
Controlled Trigger Drops	21017		Trigger drops from the Controlled Triggering System (Grouped with “Trigger Drops” indicator)
Surface Processing Time	21018		Processing Time of Frame on 35XX/32XX (microseconds)
Max Frame Rate	21019		Max Configurable frame rate given above Processing Time (scaled by 1x10 ⁻⁶)
Range Valid Count**	21100	-	Number of valid ranges.
Range Invalid Count**	21101	-	Number of invalid ranges.

Indicator	ID	Instance	Value
Anchor Invalid Count**	21200	-	Number of frames with anchoring invalid.
First Log Id	21301		ID of the first available log entry.
Last Log Id	21300		ID of the last available log entry. It is inclusive: for example, if first = 3 and last = 5, the available log IDs are 3, 4, 5. If no log is available, the last ID is less than the first ID.
Z-Index Drop Count	22000	-	The number of dropped surfaces due to a lack of z-encoder pulse during rotational part detection.
Tool Run Time	22004	Tool Index	The most recent time taken to execute the tool.
Part Total Emitted	22006	-	Total number of parts emitted by profile part detection.
Part Length Limit	22007	-	Number of parts emitted due to reaching the length limit.
Part Min Area Drops	22008	-	Number of parts dropped due to being smaller than the minimum area.
Part Backtrack Drops	22009	-	Number of parts dropped due to backtracking.
Parts Currently Active	22010	-	Number of parts currently being tracked.
Part Length	22011	-	Length of largest active part.
Part Start Y	22012	-	Start Y position of the largest active part.
Part Tracking State	22013	-	Tracking state of the largest active part.
Part Capacity Exceeded	22014	-	Part detection part or run capacity has been exceeded.
Part X Position	22015	-	Center X position of the largest active part.
Tool Runtime Minimum	22016	-	Minimum time spent for tool to process a sample
Tool Runtime Maximum	22017	-	Maximum time spent for tool to process a sample
Tool Runtime Average	22018	-	Average time for tool to process a sample
Tool Runtime Percent Average	22019	-	Average percentage of total time spent running this tool
Value	30000	Measurement ID	Measurement Value.
Pass	30001	Measurement ID	Number of pass decision.
Fail	30002	Measurement ID	Number of fail decision.
Min	30003	Measurement ID	Minimum measurement value.
Max	30004	Measurement ID	Maximum 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 overflowed

Indicator	ID	Instance	Value
			on any output. Multiple simultaneous overflows result in only a single 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.

* When the sensor is accelerated, the indicator's value is reported from the accelerating PC.

** When the sensor is accelerated, the indicator's value is the sum of the values reported from the sensor and the accelerating PC.

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, mm², mm³, 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, mm²/1000, mm³/1000, and deg/1000 in the protocols.

If buffering is enabled with the Modbus protocol, the PLC must read the Buffer Advance output register (see *State* on page 351) to advance the queue before reading the measurement results.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 190.

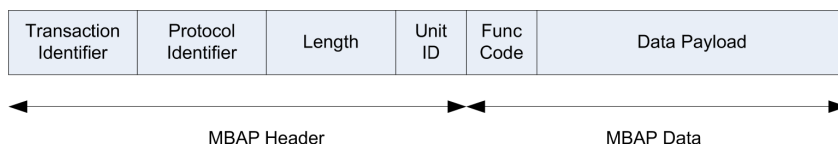
Concepts

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	2	Always set to 0.
Length	2	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 <i>Control Registers</i> below for detailed descriptions.
300 - 899	Sensor States	RO	Report sensor states. See <i>State</i> on the next page for detailed descriptions.
900 - 999	Stamps	RO	Return stamps associated with each range. See <i>State</i> on the next page for detailed descriptions.
1000 - 1059	Measurements & Decisions	RO	20 measurement and decision pairs. See <i>Measurement Registers</i> on page 353 for detailed descriptions.

Control Registers

Control registers are used to operate the sensor. Register 0 stores the command to be executed. Subsequent registers contain parameters for the commands if applicable. 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	Takes a 16-bit command. For a list of the available commands, see table below.
1 - 64	Command Parameters	WO	For Load Job (5) command: Null-terminated filename. Each 16-bit register holds a single character. Specifies the complete filename, including the file extension ".job". For Set Runtime Variables (6) command: Registers 1-8 are used to set the values of the runtime variables.

The 16-bit values used for Command Register are described below.

Command Register Values

Value	Name	Description
0	Stop Running	Stops the sensor. No effect if sensor is already stopped.

Value	Name	Description
1	Start Running	Starts the sensor. No effect if sensor is already started.
2	Align (stationary target)	Starts the stationary alignment process. State register 301 will be set to 1 (busy). When the alignment process is complete, the register is set back to zero.
3	Align (moving target)	Starts moving alignment process and also calibrate encoder resolution. State register 301 will be set to 1 (busy). When the alignment process is complete, the register is set back to zero.
4	Clear Alignment	Clears the alignment.
5	Load Job	Activates the specified job file. Set registers 1-64 to the null-terminated filename, one filename character per 16-bit register, including the null terminator character.
6	Set Runtime Variables	Sets the runtime variables. Set registers 1 through 8 to the values of all four 32-bit runtime variables.

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	Sensor State	16u	Sensor State: 0 - Stopped 1 - Running
301	Modbus Command in Progress	16u	1 when the sensor is busy performing the last command, 0 when done. Registers 302 and 311-371 below are only valid when there is no command in progress.
302	Alignment State	16u	Current Alignment State: 0 - Not aligned 1 - Aligned (Valid when register 301 = 0.)
303 – 306	Encoder Value	64u	Current Encoder value (ticks).
307 – 310	Time	64s	Current time (µs).
311	Job File Name Length	16u	Number of characters in the current job file name. (Valid when register 301 = 0.)
312 – 371	Live Job Name	16u	Name of currently loaded job file. Does not include

Register Address	Name	Type	Description
			the extension. Each 16-bit register contains a single character. (Valid when register 301 = 0.)
375	Runtime Variable 0 High	32s	Runtime variable value stored in two register locations.
376	Runtime Variable 0 Low		
...	...	...	...
381	Runtime Variable 3 High	32s	Runtime variable value stored in two register locations.
382	Runtime Variable 3 Low		

Stamp

Stamps contain trigger timing information used for synchronizing a PLC's actions. A PLC can also use this information to match up data from multiple Gocator sensors.

Stamps are updated after each range data is processed.

Stamp Register Map

Register Address	Name	Type	Description
960-975	reserved		Not used.
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 1000-1059.
977	Buffer Counter	16u	Number of buffered messages currently in the queue.
978	Buffer Overflow Flag	16u	Buffer Overflow Indicator: 0 - No overflow 1 - Overflow. (Indicates data is being lost.)
979	Inputs	16u	Digital input state.
980	zPosition High	64u	Encoder value when the index is last triggered.
981	zPosition		
982	zPosition		
983	zPosition Low		
984	Exposure High	32u	Laser exposure (µs).
985	Exposure Low		
986	Temperature High	32u	Sensor temperature in degrees Celcius * 100 (centidegrees).

Register Address	Name	Type	Description
987	Temperature Low		
988	Position High	64u	Encoder position
989	Position		
990	Position		
991	Position Low		
992	Time Low	64u	Timestamp (µs).
993	Time		
994	Time		
995	Time Low		
996	Frame Index High	64u	Frame counter. Each new sample is assigned a frame number.
997	Frame Index		
998	Frame Index		
999	Fame Index Low		

Measurement Registers

Measurement results are reported in pairs of values and decisions. Measurement values are 32 bits wide and decisions are 8 bits wide.

The measurement ID defines the register address of each pair. The register address of the first word can be calculated as $(1000 + 3 * ID)$. For example, a measurement with ID set to 4 can be read from registers 1012 (high word) and, 1013 (low word), and the decision at 1015.

The measurement results are updated after each range data is processed.

Measurement Register Map

Register Address	Name	Type	Description
1000	Measurement 0 High	32u	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

Register Address	Name	Type	Description
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, mm², mm³, 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, mm²/1000, mm³/1000, and deg/1000 in the protocols.

For information on configuring the protocol using the Web interface, see *Ethernet Output* on page 190.

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 four assembly objects: the command assembly (32 bytes), the runtime variable configuration assembly (64 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 also supported. For more information, see http://lmi3d.com/sites/default/files/APPNOTE_Implicit_Messaging_with_Allen-Bradley_PLCs.pdf.

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	x.x	Byte 0 - Major revision	Get
		USINT		Byte 1 - Minor revision	
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, runtime variable configuration, sensor state, and sample state, implicit messaging command, and implicit messaging output.

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 (SetAttributeSingle)

Attributes 1 and 2 are not implemented, as they are not required for the static assembly object.

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See Below	Command parameters Byte 0 - Command. See table below for specification of the values.	Get, Set

Command Definitions

Value	Name	Description
0	Stop running	Stop the sensor. No action if the sensor is already stopped
1	Start Running	Start the sensor. No action if the sensor is already started.
2	Stationary Alignment	Start the stationary alignment process. Byte 1 of the sensor state assembly will be set to 1 (busy) until the alignment process is complete, then back to zero.
3	Moving Alignment	Start the moving alignment process. Byte 1 of the sensor state assembly will be set to 1 (busy) until the alignment process is complete, then back to zero.
4	Clear Alignment	Clear the alignment.
5	Load Job	Load the job. Set bytes 1-31 to the file name (one character per byte, including the extension). File name must be null-terminated and end with ".job".

Runtime Variable Configuration Assembly

The runtime variable configuration assembly object contains the sensor's intended runtime variables.

Runtime Variable Configuration Assembly

Information	Value
Class	0x04
Instance	0x311
Attribute Number	3
Length	64 bytes
Supported Service	0x10 (SetAttributeSingle)

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Runtime variable configuration information. See below for more details.	Get

Sensor State Information

Byte	Name	Type	Description
0-3	Runtime Variable 0	32s	Stores the intended value of the Runtime Variable at index 0.
4-7	Runtime Variable 1	32s	Stores the intended value of the Runtime Variable at index 1.
8-11	Runtime Variable 2	32s	Stores the intended value of the Runtime Variable at index 2.
12-15	Runtime Variable 3	32s	Stores the intended value of the Runtime Variable at index 3.
16-63	Reserved		

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	0x04
Instance	0x320
Attribute Number	3
Length	100 bytes
Supported Service	0x0E (GetAttributeSingle)

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 State		Sensor state: 0 - Stopped 1 - Running
1	EtherNet/IP Command in Progress		Command busy status: 0 - Not busy 1 - Busy performing the last command Bytes 2 and 19-83 below are only valid when there is no command in progress.
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 encoder position
11-18	Time	64s	Current timestamp
19	Current Job Filename Length	8u	Number of characters in the current job filename. (e.g., 11 for "current.job"). The length includes the .job extension. Valid when byte 1 = 0.
20-83	Current Job Filename		Name of currently loaded job, including the ".job" extension. Each byte contains a single character. Valid when byte 1 = 0.
84-87	Runtime Variable 0	32s	Runtime variable value at index 0

Byte	Name	Type	Description
...	...		
96-99	Runtime Variable 3	32s	Runtime variable value at index 3

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 (GetAttributeSingle)

Attribute 3

Attribute	Name	Type	Value	Description	Access
3	Command	Byte Array	See below	Sample state information. See below for more details.	Get

Sample State Information

Byte	Name	Type	Description
0-1	Inputs	16u	Digital input state.
2-9	Z Index Position	64u	Encoder position at time of last index pulse.
10-13	Exposure	32u	Laser exposure in μ s.
14-17	Temperature	32u	Sensor temperature in degrees Celsius * 100 (centidegrees).
18-25	Position	64u	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 Overflowing	8u	Buffer overflow indicator: 0 - No overflow 1 - Overflow
44 - 79	Reserved		Reserved bytes.
80-83	Measurement 0	32s	Measurement value in μ m (0x80000000 if invalid).
84	Decision 0	8u	Measurement decision. A bit mask, where: Bit 0:

Byte	Name	Type	Description
			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 $(80 + 5 * \text{ID})$. For example, a measurement with ID set to 4 can be read from byte 100 (high word) to 103 (low word) and the decision at 104.

If buffering is enabled in the Ethernet Output panel, reading the Extended Sample State Assembly Object automatically advances the buffer. See *Ethernet Output* on page 190 for information on the **Output** panel.

Implicit Messaging Command Assembly

Implicit Messaging Command Assembly

Information	Value
Class	0x04
Instance	0x64
Attribute Number	3
Length	32 bytes

Implicit Messaging Command Assembly Information

Byte	Name	Type	Description
0	Command	8u	A bit mask where setting the following bits will only perform the action with highest

Byte	Name	Type	Description
			priority*: 1 – Stop sensor 2 – Start sensor 4 – Perform stationary alignment 8 – Perform moving alignment 16 – Clear alignment 32 – Set runtime variables 64 – Load job file *The priority of commands is currently as follows: 1. Stop sensor 2. Start sensor 3. Perform stationary alignment 4. Perform moving alignment 5. Clear alignment 6. Set runtime variables 7. Load job file
1-31	Reserved (except for configuring runtime variables and loading job file)		If you are setting the runtime variables, use bytes 4-19 to define the values of each of the four runtime variables in little endian format. If you are loading job file, use bytes 1-31 for the filename, one character per byte. The filename must be null terminated and must end with ".job".

Implicit Messaging Output Assembly

Implicit Messaging Output Assembly

Information	Value
Class	0x04
Instance	0x322
Attribute Number	3
Length	376 bytes

Implicit Messaging Output Assembly Information

Byte	Name	Type	Description
0	Sensor State	8u	Sensor state is a bit mask where: Bit 0:

Byte	Name	Type	Description
			1 – Running 0 – Stopped Bits [1-7]: 0 – No state issue Non-zero – Conflict
1	Alignment and Command state	8u	A bit mask where: Bit 0: 1 – Explicit or Implicit Command in progress 0 – No Explicit or Implicit command is in progress Bit 1 1 – Aligned 0 – Not aligned
2-3	Inputs	16u	Digital input state
4-11	Z Index Position	64u	Encoder position at time of last index pulse
12-15	Exposure	32u	Laser exposure in μ s
16-19	Temperature	32u	Sensor temperature in degrees celsius * 100 (centidegrees)
20-27	Encoder Position	64s	Encoder position
28-35	Time	64u	Time
36-43	Scan Count	64u	Represents the number of scans
44-55	Reserved		
56	Decision 0	8u	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
...	...		
119	Decision 63	8u	Measurement decision is a bit mask where: Bit 0: 1 – Pass 0 – Fail Bits [1-7]: 0 – Measurement value OK

Byte	Name	Type	Description
			1 – Invalid Value 2 – Invalid Anchor
120-123	Measurement 0	32s	Measurement value in μm . (0x80000000 if invalid)
...	...		
372-375	Measurement 63	32s	Measurement value in μm . (0x80000000 if invalid)

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 can use polling. For more information on polling commands, see *Polling Operation Commands (Ethernet Only)* on the next page.

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, mm², mm³, 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, mm²/1000, mm³/1000, 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 190.

For information on configuring the protocol with the Web interface (when using the protocol over Serial), see *Serial Output* on page 198.

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 *Command Channel* 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 Channel* on page 370 for more information.

The Health channel is used to receive health indicators (see *Health Channel* on page 372).

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

Command Channel

The following sections list the actions available on the command channel.

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.

Formats

Message	Format
Command	Trigger
Reply	OK or ERROR, <Error Message>

Examples:

Command: Trigger

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
```

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

Moving Alignment

The Moving Alignment command performs an alignment based on the settings in the sensor's live job file. A reply to the command is sent when the alignment has completed or failed. The command is timed out if there has been no progress after one minute.

Formats

Message	Format
Command	MovingAlignment
Reply	If no arguments are specified OK or ERROR, <Error Message>

Examples:

Command: MovingAlignment

Reply: OK

Command: MovingAlignment

Reply: ERROR, ALIGNMENT FAILED

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

Set Runtime Variables

The Set Runtime Variables command sets the runtime variables, using the specified index, length, and data. Values are integers.

Formats

Message	Format
Command	setvars,index,length,data
	Where <i>data</i> is the delimited integer values to be set.
Reply	OK or ERROR

Examples:

Command: setvars,0,4,1,2,3,4

Reply: OK

Get Runtime Variables

The Get Runtime Variables command gets the runtime variables, using the specified index and length.

Formats

Message	Format
Command	setvars,index,length
Reply	OK,data

Where *data* is the delimited data for the passed length.

Examples:

Command: getvars,0,4

Reply: OK,1,2,3,4

Data Channel

The following sections list the actions available on the data channel.

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 Channel

The following sections list the actions available on the health channel.

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 342 for details on health indicators.

Formats

Message	Format
Command	Health,health indicator ID.Optional health indicator instance ...

Message	Format
	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 <i>Data Results</i> on page 334).
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. 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

C language *printf*-style formatting is also supported: for example, %sprintf[%09d, %value[0]]. This allows fixed length formatting for easier input parsing in PLC and robot controller logic.

Selcom Protocol

This section describes the Selcom serial protocol settings and message formats supported by Gocator sensors.

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

For information on configuring the protocol using the Web interface, see *Serial Output* on page 198.



Units for data scales use the standard units (mm, mm², mm³, and degrees).

Serial Communication

Data communication is synchronous using two unidirectional (output only) RS-485 serial channels: data (Serial_Out0) and clock (Serial_Out1). See *Serial Output* on page 441 for cable pinout information.

Measurement results are sent on the serial output (data) in asynchronous mode. Measurement values and decisions can be transmitted to an RS-485 receiver, but job handling and control operations must be performed through the Gocator's web interface or through communications on the Ethernet output.

Connection Settings

The Selcom protocol uses the following connection settings:

Serial Connection Settings

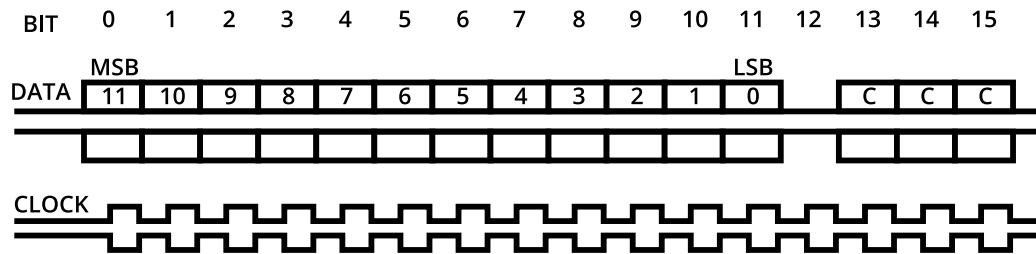
Parameter	Value
Data Bits	16
Baud Rate (b/s)	96000, 512000, 1024000
Format	Binary

Message Format

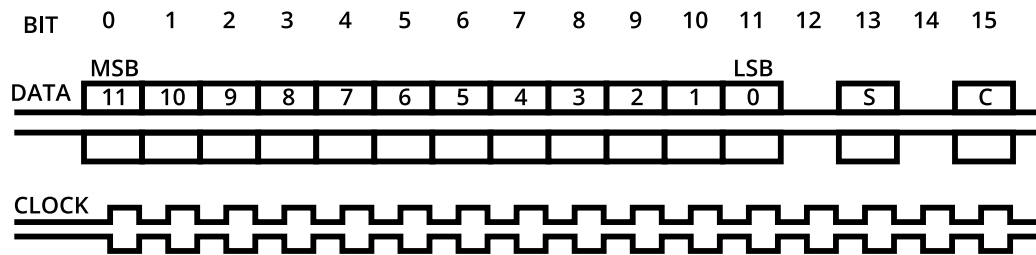
The data channel is valid on the rising edge of the clock and data is output with the most significant bit first, followed by control bits for a total of 16 bits of information per frame. The time between the start of the camera exposure and the delivery of the corresponding range data is fixed to a deterministic value.

The sensor can output data using one of four formats, illustrated below, where:

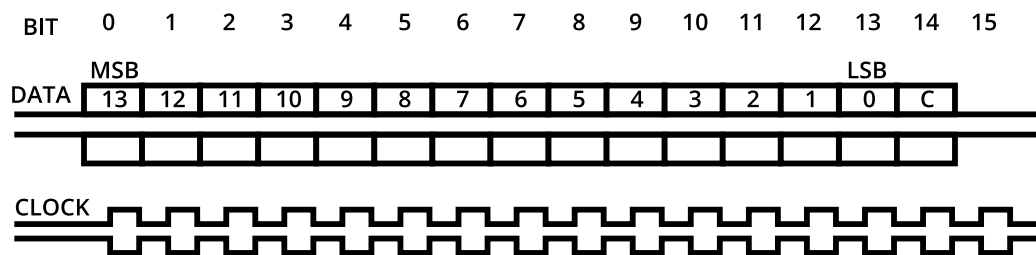
- MSB = most significant bit
- LSB = least significant bit
- C = data valid bit (high = invalid)
- S = whether data is acquired in search mode or track mode (high = search mode)



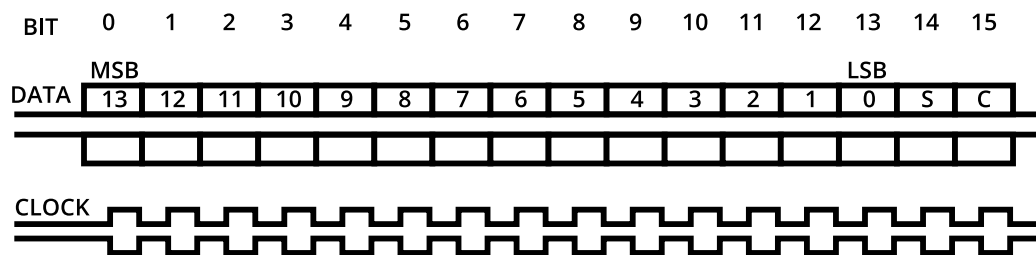
12-bit data format (SLS mode; "SLS" in Gocator web interface)



12-bit data format with Search/Track bit



14-bit data format



14-bit data format with Search/Track bit

Development Kits

These sections describe the following development kits:

- [Software Development Kit \(GoSDK\)](#)
- [Gocator Development Kit \(GDK\)](#)

GoSDK

The Gocator Software Development Kit (GoSDK) includes open-source software libraries and documentation that can be used to programmatically access and control Gocator sensors. To get the latest version of the SDK, go to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download it from the Download Center.

You can download the Gocator SDK from within the Web interface.



To download the SDK:

1. Go to the **Manage** page and click on the **Support** category
2. Next to **Software Development Kit (SDK)**, click **Download**
3. Choose the location for the SDK on the client computer.

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 go to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download it from the Download Center.

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 GoSDK class reference is found by accessing the following file:

14400-4.x.x.xx_SOFTWARE_GO_SDK\GO_SDK\doc\GoSdk\Gocator_1x00\GoSdk.html

.

Examples

Examples showing how to perform various operations are provided, each one targeting a specific area. For Visual Studio, the examples can be found in solution files specific to different versions of Visual Studio. For example, *GoSdk-2017.sln* is for use with Visual Studio 2017. A make file for Linux systems is also provided.

To run the GoSDK examples, make sure *GoSdk.dll* and *kApi.dll* (or *GoSdkd.dll* and *kApid.dll* in debug configuration) are copied to the executable directory.

Example Project Environment Variable

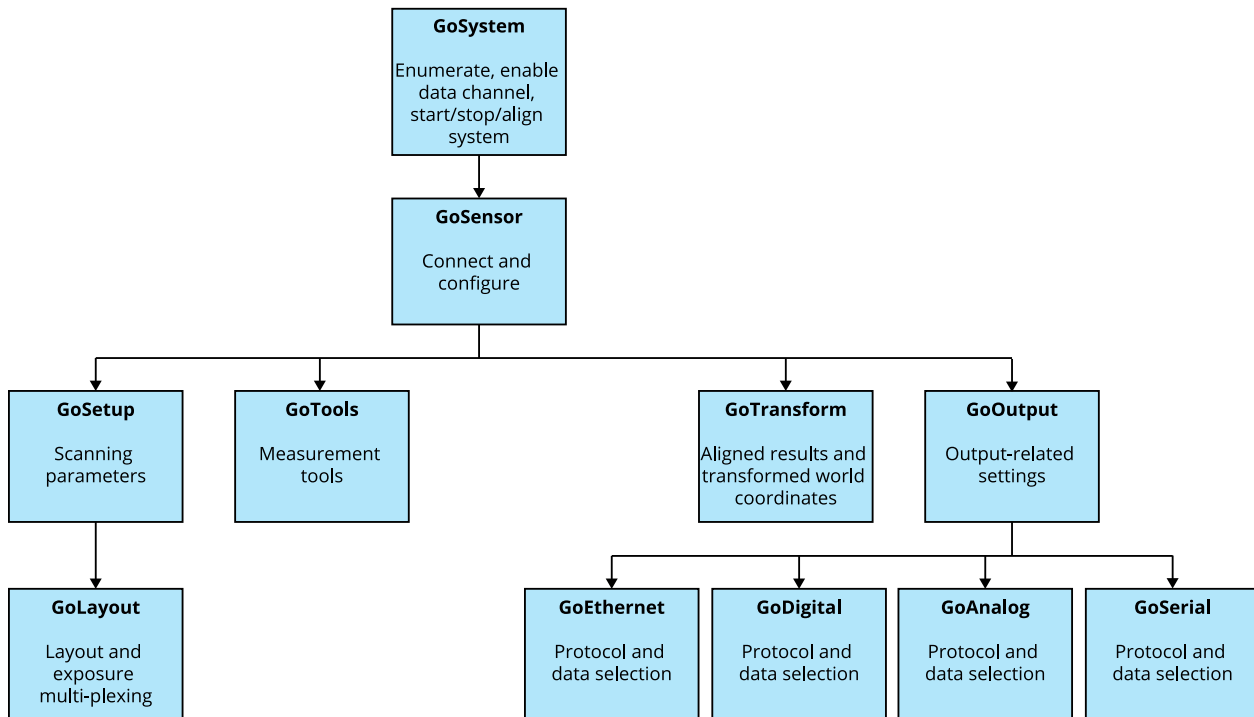
All GoSDK example projects use the environment variable *GO_SDK_4*. The environment variable should point to the *GO_SDK* directory, for example, *C:\14400-4.0.9.156_SOFTWARE_GO_SDK\GO_SDK*.

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`.

Functional Hierarchy of Classes

This section describes the functional hierarchy of the classes in the Gocator 4.x SDK ("GoSDK"). In the following diagram, classes higher in the hierarchy often provide resources for classes lower in the hierarchy, and for this reason should be instantiated earlier in a client application.



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
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
GoAlignMsg	Represents a message containing an alignment result.

Data Type	Description
GoBoundingBoxMatchMsg	Represents a message containing bounding box based part matching results.
GoDataMsg	Represents a base message sourced from the data channel. See <i>GoDataSet Type</i> below for more information.
GoEdgeMatchMsg	Represents a message containing edge based part matching results.
GoEllipseMatchMsg	Represents a message containing ellipse based part matching results.
GoExposureCalMsg	Represents a message containing exposure calibration results.
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.
GoSectionMsg	Represents a data message containing a set of section arrays.
GoSectionIntensityMsg	Represents a data message containing a set of profile intensity 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.

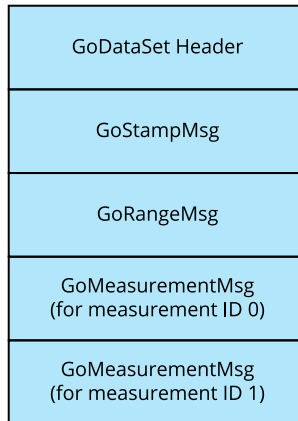


See *Setup and Locations* on page 378 for more information on the code samples.

GoDataSet Type

Data are passed to the data handler in a *GoDataSet* object. The *GoDataSet* object is a container that can contain any type of data, including scan data (ranges or profiles), measurements, and results from various operations. Data inside the *GoDataSet* object are represented as messages.

The following illustrates the content of a *GoDataSet* object of a range mode setup with two measurements.



After receiving the *GoDataSet* object, you should call *GoDestroy* to dispose the *GoDataSet* object. You do not need to dispose objects within the *GoDataSet* object individually.

 All objects that are explicitly created by the user or passed via callbacks should be destroyed by using the *GoDestroy* function.

Measurement Values and Decisions

Measurement values and decisions are 32-bit signed values (k32s). See *Value Types* on page 380 for more information on value types.

The following table lists the decisions that can be returned.

Measurement Decisions

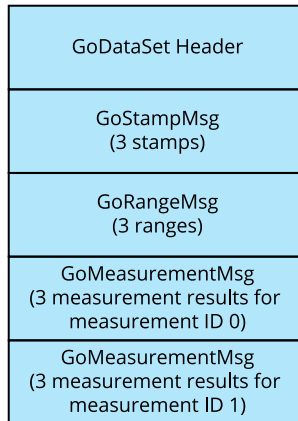
Decision	Description
1	The measurement value is between the maximum and minimum decision values. This is a pass decision.
0	The measurement value is outside the maximum and minimum. This is a fail decision.
-1	The measurement is invalid (for example, the target is not within range). Provides the reason for the failure.
-2	The tool containing the measurement is anchored and has received invalid measurement data from one of its anchors. Provides the reason for the failure.

Refer to the *SetupMeasurement* example for details on how to add and configure tools and measurements. Refer to the *ReceiveMeasurement* example for details on how to receive measurement decisions and values.

 You should check a decision against ≤ 0 for failure or invalid measurement.

Batching

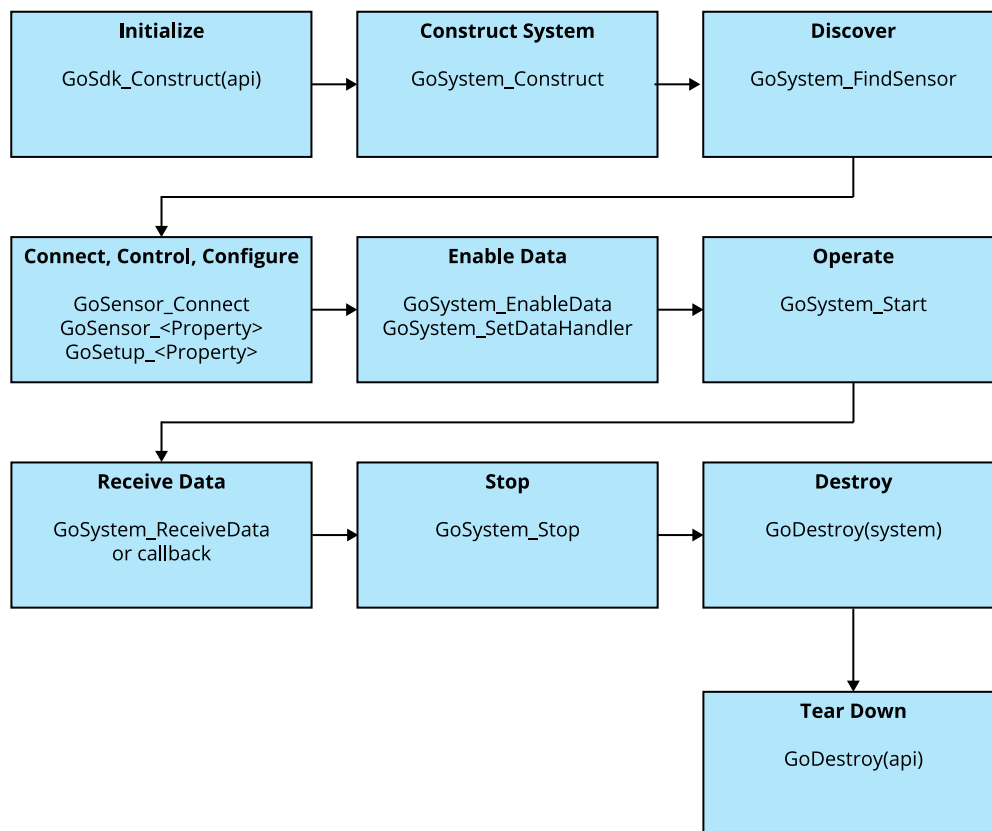
One *GoDataSet* object can contain data from multiple frames. Each message has a *Count* property that specifies how many frames of data are included. The following illustrates the data structure when three frames of data are contained inside a *GoDataSet* object.



The batching size is dynamically adjusted to ensure the sensor's CPU keeps up with the messages delivered with the shortest latency.

Operation Workflow

Applications created using the SDK typically use the following programming sequence



See *Setup and Locations* on page 378 for more information on the code samples referenced below.



Sensors must be connected before the system can enable the data channel.



All GoSDK 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 *Functional Hierarchy of Classes* on page 378 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

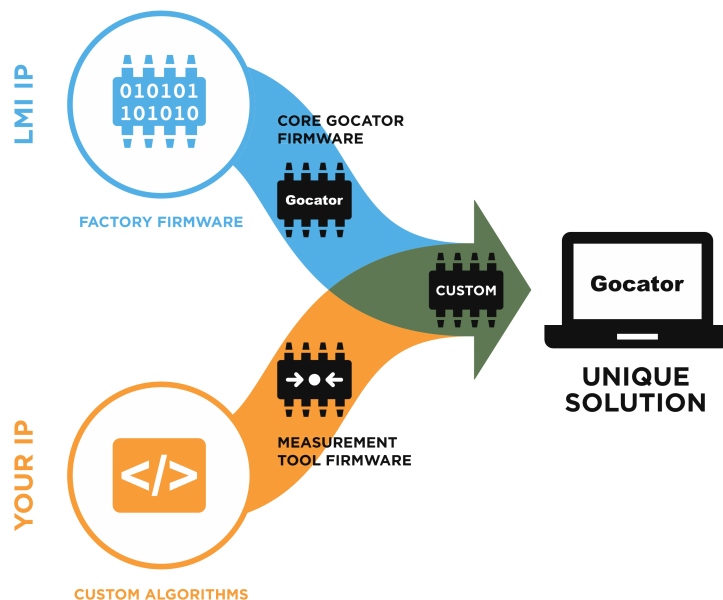
Name	Description
GoSensor_Restore	Restores a backup of sensor files.
GoSensor_RestoreDefaults	Restores factory default settings.
GoSensor_CopyFile	Copies a file within the connected sensor. The flash write operation does not occur if GoSensor_CopyFile function is used to load an existing job file. This 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.

Name	Description
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.

GDK

The Gocator Development Kit (GDK) is a framework for developing and testing custom Gocator measurement tools containing your own algorithms, and then deploying them to Gocator sensors.



Custom tools created with the GDK act much like native Gocator measurement tools, running at native speeds and taking advantage of features such as anchoring. The GDK supports all data types (range and profile), and tools created with the GDK use the same data visualization as native tools.

Benefits

When you use the GDK to create custom measurement tools, you have complete control over how and where your custom measurement tools can be used, which protects your intellectual property.

You can also easily troubleshoot and modify your tools on-site, letting you respond quickly to your customers' urgent issues.

Supported Sensors

The GDK is available for free for the following Gocator sensors:

- Gocator 1300 series
- Gocator 2100 series

- Gocator 2300 series
- Gocator 2400 series
- Gocator 2880
- Gocator 3210 and 3506

Typical Workflow

The following is the typical workflow for creating and deploying custom measurement tools:

- Develop and build measurement tools using the GDK project files and libraries in Microsoft Visual Studio, targeting Win32.
- Debug the measurement tools using the emulator on a PC.
- Build the measurement tools into a custom firmware binary.
- Upload the custom firmware to a sensor.

Installation and Class Reference

The GDK project and library files are in the 14524-x.x.x.xxx_SOFTWARE_GDK.zip GDK package. You can download the GDK package by going to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download it from the Download Center, selecting a Gocator series, and clicking on the *Product User Area* link.

After downloading the package, extract the package to a directory.

You can access full installation and setup instructions, as well as the complete class reference documentation, by clicking the *Guide* shortcut under the root directory.

 bin	8/4/2016 2:08 AM	File folder
 doc	8/4/2016 2:10 AM	File folder
 Gocator	8/4/2016 2:14 AM	File folder
 lib	8/4/2016 2:15 AM	File folder
 pkg	8/4/2016 2:16 AM	File folder
 Platform	8/4/2016 2:16 AM	File folder
 res	8/4/2016 2:16 AM	File folder
 Guide	8/3/2016 1:39 PM	Shortcut

Required Tools

The GDK requires the following tools:

- Microsoft Visual Studio 2013
- Python 3.4 or later
- GCC ARM tools 4.9.3-p2 (for Gocator 2300C and 2400 series sensors)
- Texas Instruments C6000 Code Generation Tools 7.4.13 (for Gocator 1300, 2300A, 2300B, 3210, and 3506 series sensors)

You can download Microsoft Visual Studio 2013 from Microsoft's website.

The 14525_1.0.0.0_SOFTWARE_GDK_Prerequisites.zip package contains the installation files for Python 5.2, the GCC ARM tools, and the Texas Instruments code generation tools. For information on installing the package, see *Installation and Class Reference* on the previous page.

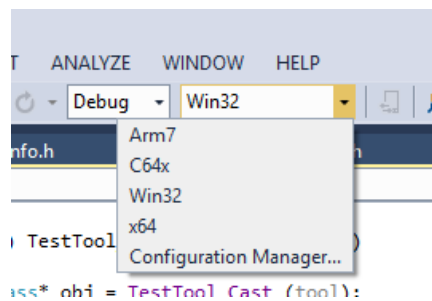
Getting Started with the Example Code

The best way to get started is with the GDK sample code. You can find the sample projects under Gocator\GDKSampleApp. This project is ready for you to build and use as a template for new projects.

Start by opening GDK.sln in Visual Studio 2013.

Building the Sample Code

You can build the sample code for working with either the emulator or a sensor. To do this, choose the target and then build the solution



The following targets are available:

- Win32/x64 for debugging code and emulating a sensor to test tools (on a PC)
- Arm7 for building for Gocator 2300C and 2400 series sensors
- C64x for Gocator 1300, 2300A, 2300B, 3210, and 3506 series sensors

The Win32 target supports Debug and Release builds. The Arm7 and C64x targets (Gocator sensors) only support Release builds.

Tool Registration

For a tool to be available to a user in the Gocator web interface, you must add it to the project assembly in Asm.c.

```
#include <GdkSampleApp/Asm.h>
#include <GdkSampleApp/TestProfileSelect.h>
#include <GdkSampleApp/TestSurfaceSelect.h>
#include <GdkSampleApp/TestSurfaceConfiguration.h>
#include <GdkSampleApp/TestSurfaceGraphics.h>
#include <Gdk/GdkLib.h>
#include <GoSensor/Version.h>
#include <GoSensorAppLib/GsaDef.h>
#include <GoSensorAppLib/GsaAsm.h>

kBeginAssembly(Tool, ToolAsm, TOOL_VERSION, GOCATOR_VERSION)
    kAddDependency(GdkLib)
    kAddType(TestProfileSelect)
    kAddType(TestSurfaceSelect)
    kAddType(TestSurfaceConfiguration)
```

```
kAddType (TestSurfaceGraphics)
kEndAssembly ()
```

You can add multiple tools in a GDK project. As seen above, `TestProfileSelect`, `TestSurfaceSelect`, `TestSurfaceConfiguration`, etc. will be available for users from the drop-down menu in the **Tools** panel in sensor's web interface.

Tool Definitions

You must add standard entry functions (methods) for each tool. The class table declares the entry functions:

```
kBeginClass (Tool, TestTool, GdkTool)
    kAddVMethod (TestTool, kObject, VRelease)
    kAddVMethod (TestTool, GdkTool, VInit)
    kAddVMethod (TestTool, GdkTool, VDescribe)
    kAddVMethod (TestTool, GdkTool, VNewToolConfig)
    kAddVMethod (TestTool, GdkTool, VNewMeasurementConfig)
    kAddVMethod (TestTool, GdkTool, VUpdateConfig)
    kAddVMethod (TestTool, GdkTool, VStart)
    kAddVMethod (TestTool, GdkTool, VStop)
    kAddVMethod (TestTool, GdkTool, VProcess)
kEndClass ()

ToolFx (kStatus) TestTool_VDescribe (GdkToolInfo toolInfo)
{
    GdkMeasurementInfo mmt;
    GdkParamsInfo params;
    GdkParamInfo paramInfo;

    kCheck (GdkToolInfo_SetTypeName (toolInfo, TEST_PROFILE_SELECT_TOOL_NAME));
    kCheck (GdkToolInfo_SetLabel (toolInfo, TEST_PROFILE_SELECT_TOOL_LABEL));

    kCheck (GdkToolInfo_SetSourceType (toolInfo, GDK_DATA_TYPE_UNIFORM_PROFILE));

    ...
}
```

The function `<Tool Name>_VDescribe` describes the tool and its basic configuration. This function is called during sensor start-up. For more information on entry functions, see *Entry Functions* below.



The `VDescribe` function for each tool is properly formed. Significant issues with this function (for example, overwriting memory) could prevent the sensor from starting.



You should use the emulator to debug tools *before* deploying tools to sensors.

Entry Functions

The following table describes the main entry functions.

Function	Description
VDescribe	Defines the tool's name, data types, acceptable source options, configuration parameters, and at least one measurement.

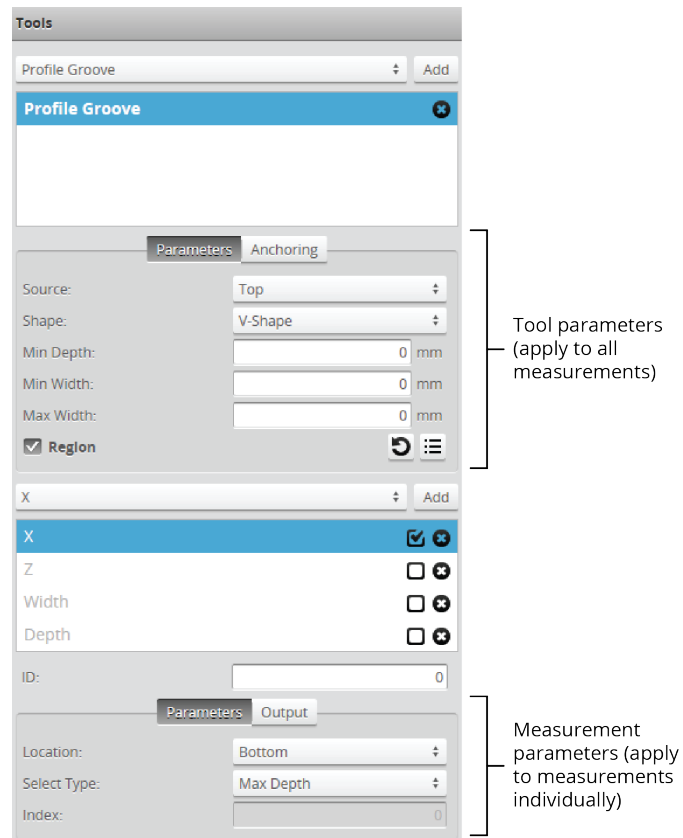
Function	Description
VStart	Called when the sensor starts running (that is, the user clicks the Run button). The function gets parameters from <code>GtTool</code> . You typically allocate memory in this function.
VProcess	Called every time data is received while the sensor is running.
VStop	Called when the user clicks the Stop button.

The `TestSurfaceConfiguration` example shows how to create and modify parameters based on other user settings.

For full descriptions of these functions, see the GDK class reference documentation (see *Installation and Class Reference* on page 389 for information on installing the documentation).

Parameter Configurations

Each tool has two levels of parameters: tool parameters and measurement parameters.



A tool can contain multiple measurements. In the image above, the Groove tool contains four measurements: X, Z, Width, and Depth. Each tool has one set of tool parameters and each measurement in a tool has one set of measurement parameters.

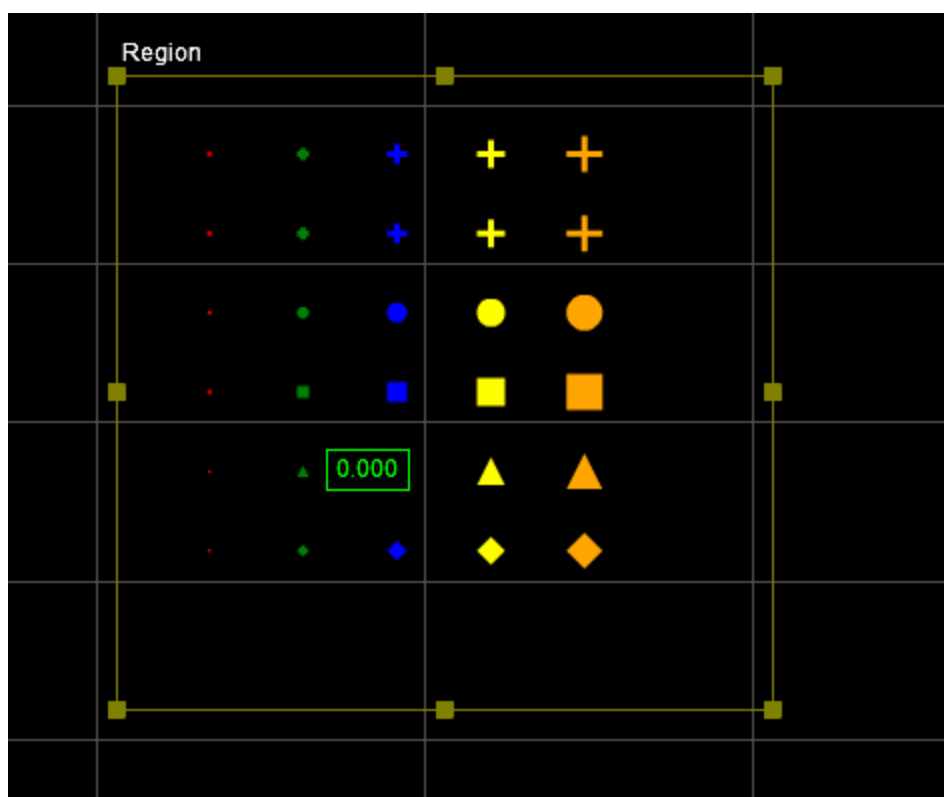
The following table lists the functions that provide advanced or interactive control for setting up tool and measurement parameters:

Function	Description
VNewToolConfig	Advanced method for setting default values of tool parameters based on the current sensor configuration (for example, active area). Called when a new tool is added in the interface.
VNewMeasurementConfig	Advanced method for setting default values of measurement parameters based on the current sensor configurations (for example, active area). Called when measurements in a tool is are added in the interface.
VUpdateConfig	Advanced method for updating the configuration based on parameters set by users.

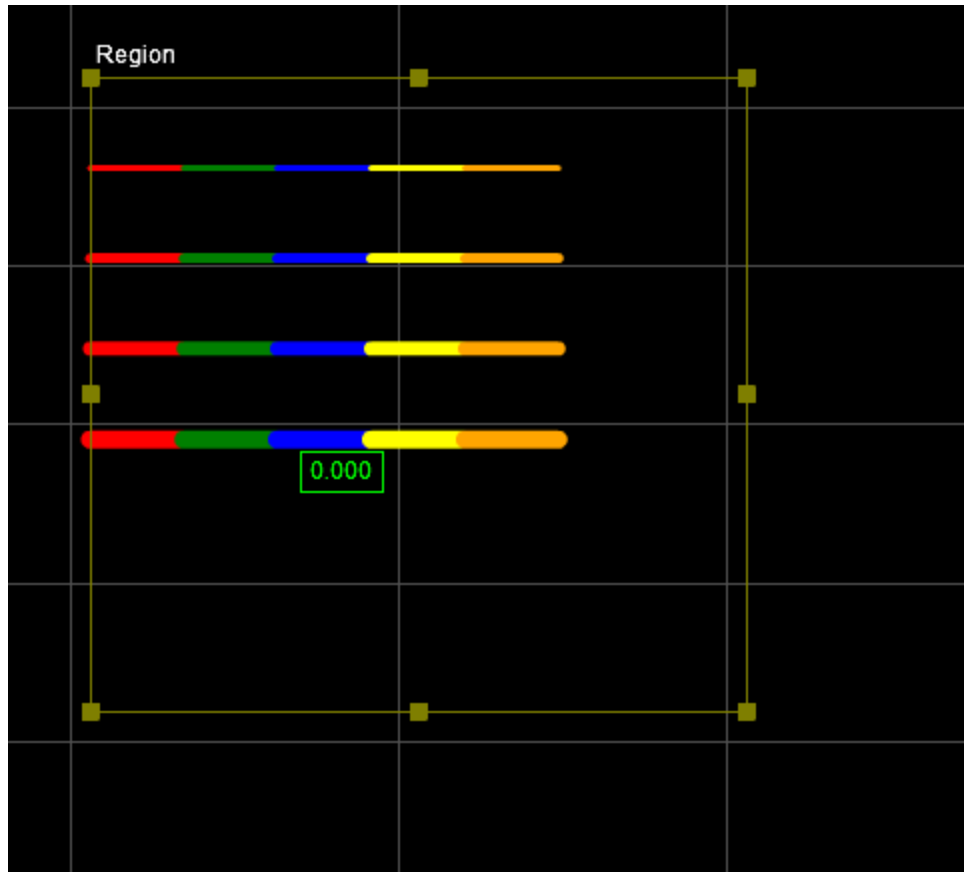
For full descriptions of these functions, see the GDK class reference documentation (see *Installation and Class Reference* on page 389 for information on installing the documentation).

Graphics Visualization

The `GdkGraphic` function supports points and lines.



Point graphics



Line graphics

To create graphics:

1. Use `GdkGraphic_Construct` to create a graphic object.
2. Use `GdkGraphicPointSet_Construct` to create points or `GdkGraphicLineSet_Construct` to create lines.
3. Add the points and lines to the graphic object using `GdkGraphic_AddPointSet` and `GdkGraphic_AddLineSet`.
4. Output using `GdkToolOutput_SetRendering`.

The following illustrates the process:

```
kTest(GdkGraphic_Construct(&graphic, kObject_Alloc(tool)));
kTest(GdkGraphicPointSet_Construct(&pointSet, 4.0, kMARKER_SHAPE_CROSS, kCOLOR_LIME,
&point32f, 1, kObject_Alloc(tool)));
kTest(GdkGraphic_AddPointSet(graphic, pointSet));
kTest(GdkToolOutput_SetRendering(output, measurementIndex, graphic));
```

The GDK example `TestSurfaceGraphics` shows how to use the graphics functions.

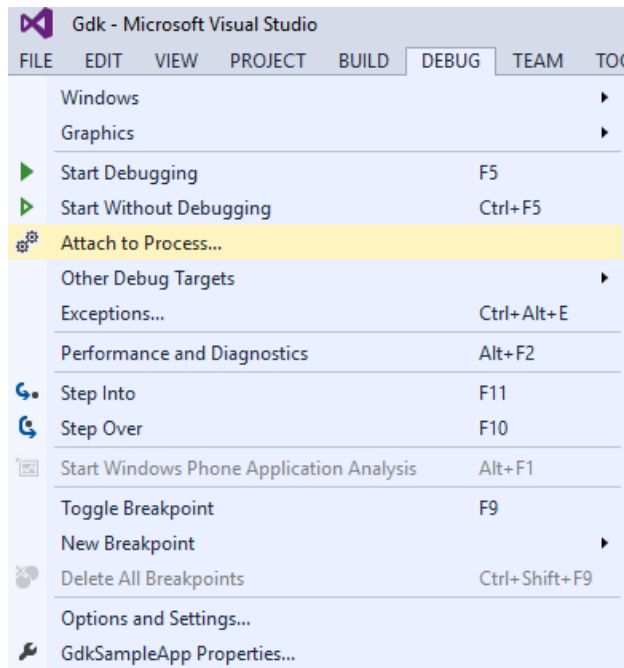
 Graphic functions take an array of `kPoint3d32f`. It does NOT accept `kPoint3d64f`.

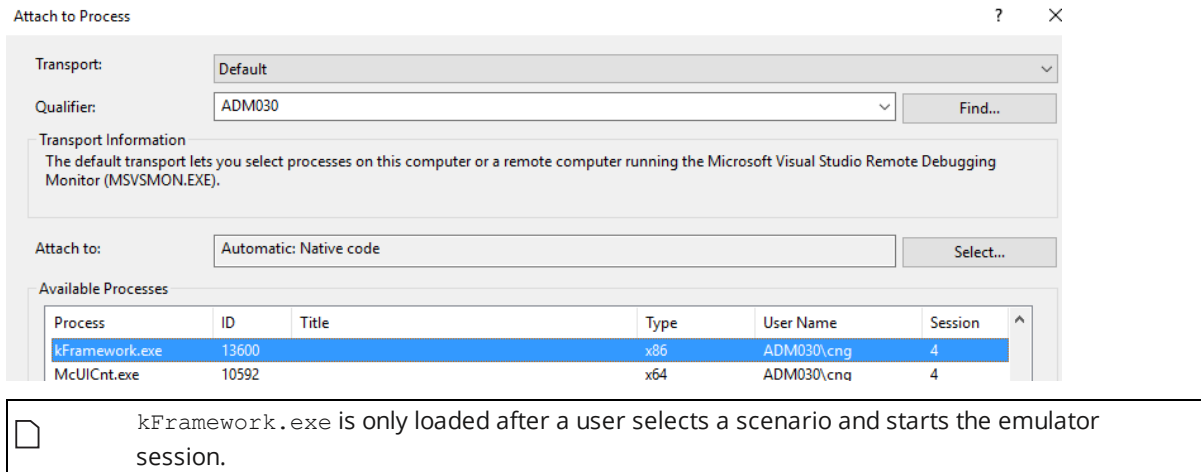
Debugging Your Measurement Tools

We highly recommend using the emulator to debug measurement tools you create with the GDK. By using a Gocator support file and previously recorded scan data, downloaded from a physical sensor, you can completely simulate standalone and dual-sensor configurations on a PC to test your tools.

To debug your tools in the emulator:

1. Compile your code using the Win32 target (Debug or Release).
2. In the output directory, rename the DLL with the same name as your project to GdkApp.dll.
For example, if your project is called `MyGDKTools`, the resulting DLL should be called `MyGDKTools.dll`. You rename this DLL to `GdkApp.dll`.
The output directories are as follows:
Release: `win32`
Debug: `win32d`
3. Launch the emulator from same output directory as in step 2.
4. In the emulator, choose a scenario and start it.
5. In Visual Studio, attach the debugger to the `kFramework.exe` process.





Debugging Entry Functions

VStart, *VProcess*, and *VStop* are called whenever a data record is played back in the emulator (that is, when a user clicks on the Next button or types the frame number in the frame field) with at least one tool instance. For more information on playback controls, see *Recording, Playback, and Measurement Simulation* in the Gocator user manual.

VDescribe however is called when the DLL loads, before the debugger can attach to the `kFramework.exe` process. To debug *VDescribe*, we recommend testing the function calls by putting them in *Vinit*.

For information on building targets for testing in the emulator, see the GDK class reference documentation.

Tips

The following sections provide useful information for creating custom measurement tools.

Backward Compatibility with Older Versions of Tools

When loading a recording or job file that contains a custom measurement tool, the parameters in the loaded recording or job file must match those in the firmware.

By default, if declared parameters are missing from the configuration, a job file or a recording will fail to load.

There are two ways to provide backward compatibility with older parameter sets.

Define new parameters as optional

Mark a parameter as optional with the function `GdkParamInfo_SetIsOptional`. When a parameter is marked as optional, parameter parsing functions succeed even if the parameter is missing from the configuration. The missing parameter is initialized with default value.

Configuration Versioning

Over the lifetime of a tool, you may need to make changes to its interface (for example, changing or removing parameters). The user-defined aspects of a tool interface—its parameters and

measurements—are captured by `GDKToolVersionInfo` objects.

By default, a tool has just one version (`GdkToolInfo_FirstVersion`), but more versions may be added using `GdkToolInfo_AddVersion`. Whenever the interface of a tool has changed, a new version can be registered so that the new interface can be correctly parsed by the framework.

When the configuration of a tool instance is saved, the version used at the time is also saved. This saved version is used by the framework to parse the configuration. If a version not defined by the firmware implementation, then that tool instance will not be inactive.

During run-time, you can query the version of the configuration of a tool instance by using `GdkToolCfg_Version`. You can then interpret the parameters depending on the version the configuration is saved in.

```
GdkFx(kStatus) GdkExampleTool_VDescribe(GdkToolInfo info)
{
    kCheck(GdkToolInfo_SetLabel(info, "Example"));

    kCheck(GdkToolInfo_SetSourceType(info, GDK_DATA_TYPE_UNIFORM_PROFILE));
    kCheck(GdkToolInfo_AddSourceOption(info, GDK_DATA_SOURCE_TOP));

    kCheck(GdkExampleTool_DescribeV0(info));
    kCheck(GdkExampleTool_DescribeV1(info));

    kCheck(GdkToolInfo_SetDefaultVersion(info, GdkToolInfo_VersionAt(info, 1)));

    return kOK;
}

GdkFx(kStatus) GdkExampleTool_DescribeV0(GdkToolInfo info)
{
    kCheck(GdkParamsInfo_Add(GdkToolInfo_Params(info), "RefRegion", GDK_PARAM_TYPE_PROFILE_
REGION, "Ref Region", kNULL));
    kCheck(GdkParamsInfo_Add(GdkToolInfo_Params(info), "Region", GDK_PARAM_TYPE_PROFILE_
REGION, "Region", kNULL));
    kCheck(GdkToolInfo_SetFirstVersionName(info, ""));

    return kOK;
}

GdkFx(kStatus) GdkExampleTool_DescribeV1(GdkToolInfo info)
{
    GdkToolVersionInfo versionInfo;

    // Auto-version

    kCheck(GdkToolInfo_AddVersion(info, kNULL, &versionInfo));
    kCheck(GdkToolVersionInfo_UseBase(versionInfo, GdkToolInfo_FirstVersion(info)));
    kCheck(GdkParamsInfo_AddFloat(GdkToolVersionInfo_Params(versionInfo), "BaseScale",
kNULL, 2.0, kNULL));

    return kOK;
}
```

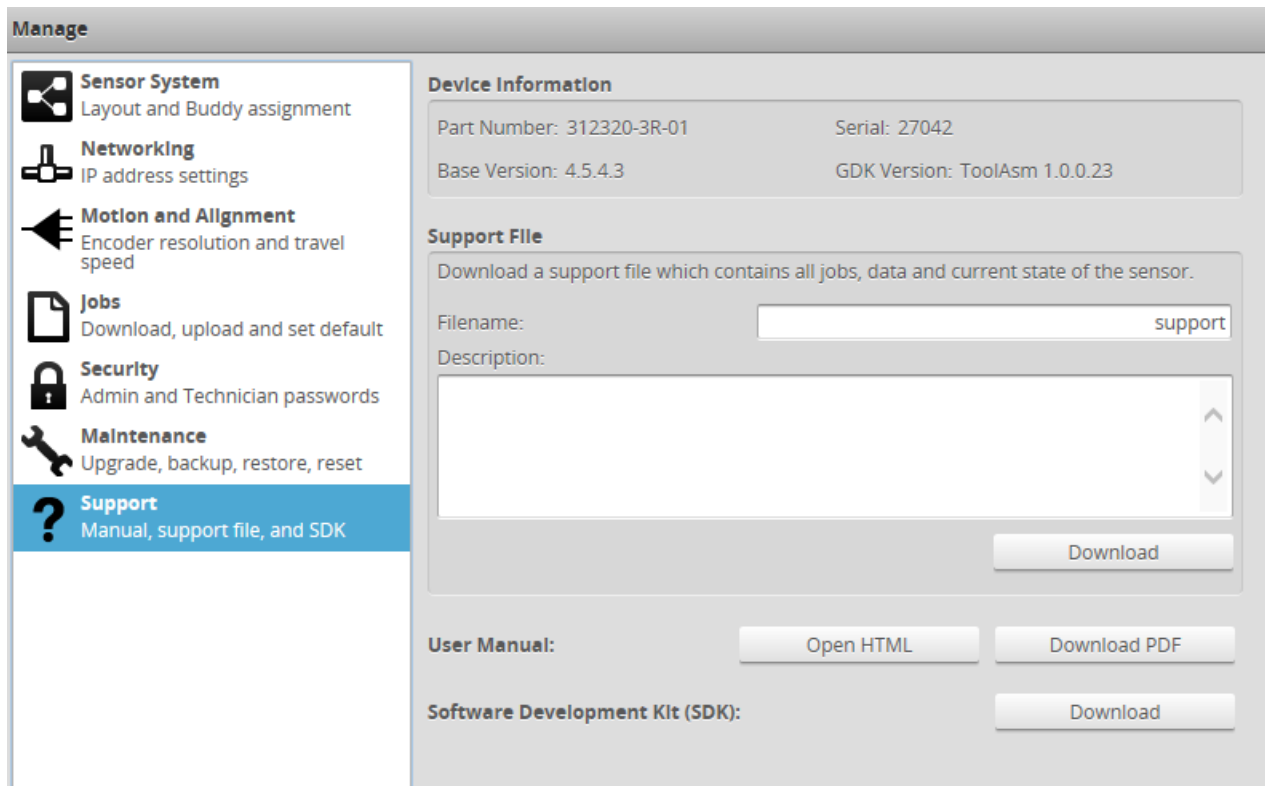
Adding a new measurement does not require special handling. The new measurement is just not instantiated in a previous configuration.

Version

You can define the version number of your tools in `Asm.x.h`.

```
#define TOOL_VERSION    kVersion_Stringify_(1, 0, 0, 23)
```

The version is displayed on the **Manage** page, in the **Support** category.



Common Programming Operations

The following sections describe common programming operations.

Input Data Objects

The `VProcess` function receives a `GdkToolInput` object as input. This object is a container where the information and actual data of the received input is stored.

```
GdkInputItem item = GdkToolInput_Find(input, obj->dataSource);  
GdkDataInfo itemInfo = GdkInputItem_Info(item);
```

The `GdkToolInput_Find` and `GdkInputItem_Info` functions are used to extract the item and info objects. These objects can then be used to retrieve the input data and information (for example, offset and resolution) associated to the input. The following are some examples:

Retrieving offset and scales

```
const kPoint3d64f* scale = GdkDataInfo_Scale(GtInputItem_Info(item));
const kPoint3d64f* offset = GdkInputItem_Offset(item);
```

Retrieving a pointer to surface and intensity surface data

```
const k16s* rangeSrc = GdkSurfaceInput_RangeRowAt(item, 0);
const k8u* intensitySrc = GdkSurfaceInput_IntensityRowAt(item, 0);
```

Similar functions are available for retrieving information from range and profile data.

Computing actual height information using a offset and scales

```
k64f height = rangeSrc[index] * scale->z + offset->z;
```

Extracting height information from profiles and surfaces.

The `TestProfileSelect` and `TestSurfaceSelect` examples show how to perform these operations.

Setup and Region Info during Tool Initialization

Memory allocation is often done in the `VInit` or `VStart` function. To retrieve sensor and data information such as active area settings and data scale outside of `VProcess`, you can use the following function:

```
GdkDataInfo info = GdkSensorInfo_DataSource(GdkTool_SensorInfo(tool), GDK_DATA_SOURCE_TOP);
```

Computing Region Based on the Offset from an Anchor Source

Just like built-in measurement tools, custom tools created with the GDK can be anchored to another tool (GDK-based tools or built-in tools).

To compute the offset region:

```
TestToolClass* obj = TestTool_Cast_(tool);
GdkParams params = GdkToolCfg_Parameters(config);
const kPoint3d64f* anchor = GdkToolInput_AnchorPosition(input);
GdkRegionXZ64f offsetRegion = { k64F_NULL, k64F_NULL, k64F_NULL, k64F_NULL };

param = GdkParams_Find(params, "Region");
obj->region = *GdkParam_AsProfileRegion(param);

offsetRegion = obj->region;
offsetRegion.x += anchor->x;
offsetRegion.z += anchor->z;
```

In the code above, we first retrieve the tool's region settings (before anchoring is applied), and then adjust the region based on the results from the anchored source in `VProcess`. If the anchored source fails, the tools will not be invoked.

The `TestProfileSelect` and `TestSurfaceSelect` examples show how to extract height information from anchored regions.

For more information on anchoring, see *Measurement Anchoring* in the Gocator user manual.

Part Matching

When part matching is enabled, the tool receives translated and corrected surface data. If part matching fails for the current scan (for example, the quality score is too low), the tools will not be invoked.

For more information on part matching, see *Part Matching* in the Gocator user manual.

Accessing Sensor Local Storage

You can access a sensor's local storage by using the kFile API.

For example, to read and write a file to a sensor's storage, you could use the following:

```
#include <kApi/Io/kFile.h>

...

ToolFx(kStatus) TestTool_VStart(TestTool tool)
{
    ...

    kFile_Save("test.txt", stringBuffer, (kSize) 1024);
    kFile_Load("test.txt", stringBuffer, &bufLen, kNULL);
}
```

Print Output

In the emulator, you can send output to Visual Studio or to programs such as DebugView by using the `OutputDebugString` function.

```
GtsFx(kStatus) TestTool_Trace(const kChar* format, ...)
{
    kStatus status = kOK;
    kChar debugLine[256];

    kVarArgList argList;
    kVarArgList_Start_(argList, format);
    {
        status = kStrPrintvf(debugLine, 256, format, argList);
    }
    kVarArgList_End_(argList);
    OutputDebugStringA(debugLine);
    return status;
}
```



`OutputDebugString` is NOT supported on sensor targets. Use `#ifdef` to comment out the code when compiling against sensor targets.

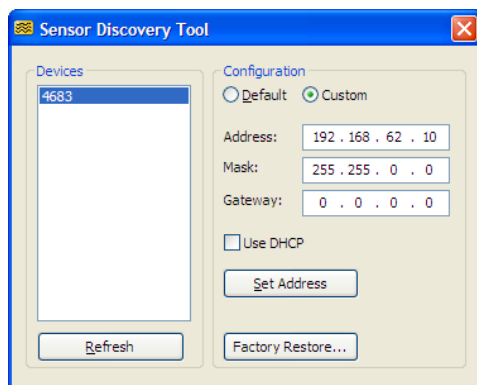
Tools

The following sections describe the tools you can use with a Gocator.

Sensor Discovery 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 the Sensor Discovery software tool. This 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 [>Discovery>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. Select the **Custom** option.
2. Enter the new network address information.
3. Click **Set Address**.

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.

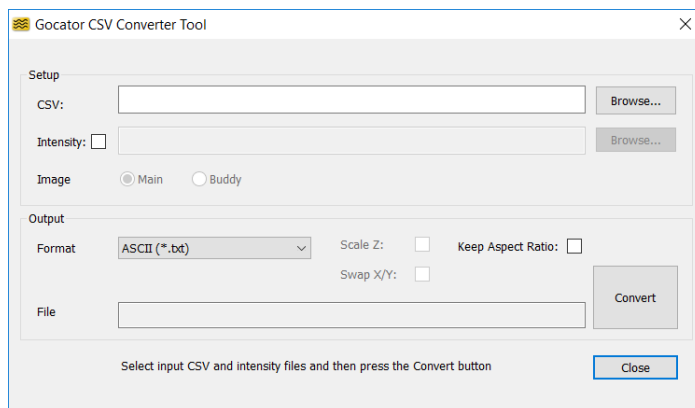


The Sensor Discovery tool uses UDP broadcast messages to reach sensors on different subnets. This enables the Sensor Discovery tool to locate and re-configure sensors even when the sensor IP address or subnet configuration is unknown.

CSV Converter Tool

The CSV Converter tool lets you convert data exported from Gocator in the CSV format to several formats (see table below). For more information on exporting recorded data, see *Downloading, Uploading, and Exporting Replay Data* on page 61.

For information on the CSV file format that Gocator exports, see the next section.



The tool supports data exported from Profile mode.

To get the tool package (14405-x.x.x.x_SOFTWARE_GO_Tools.zip), go to <http://lmi3d.com/support>, choose your product from the Product Downloads section, and download it from the Download Center.

After downloading the tool package, unzip the file and run the Gocator CSV Converter tool [CsvConverter>kCsvConverter.exe].

The tool supports the following output formats:

Output formats

Format	Description
ASCII (XYZI)	Comma-separated points in X, Y, Z, Intensity (if available) format.
16-bit BMP	Heightmap with 16bit height values in a 5-5-5 RGB image. Not intended for visualization.
16-bit TIFF	Heightmap as grayscale image.
16-bit PNG	Heightmap as grayscale image.
GenTL RGB	Not intended for use with this sensor family.
GenTL Mono	Not intended for use with this sensor family.
Raw CSV	LMI Gocator CSV format for a single frame.
HexSight HIG	LMI HexSight heightmap.
STL ASCII	Mesh in standard STL text format (can become very large).
STL Binary	Mesh in binary STL format.
Wavefront OBJ	Mesh with comma-separated vertices and facets in text

Format	Description
	format.
ODSCAD OMC	GFM ODSCAD heightmap.
MountainsMap SUR	DigitalSurf MountainsMap heightmap.
24-bit Spectrum	Color spectrum bitmap for visualization of heightmap. Does not contain height values.

With some formats, one or more of the following options are available:

Output options

Option	Description
Scale Z	Resamples the Z values to use the full value range.
Swap X/Y	Swaps the X and Y axes to obtain a right-handed coordinate system.
Keep Aspect Ratio	Resamples the X and Y axes to obtain the proper aspect ratio.

To convert exported CSV into different formats:

1. Select the CSV file to convert in the **CSV** field.
2. (Optional) 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. If a dual-sensor system was used, choose the source sensor next to **Image**.
4. Select the output format.
For more information on output formats, see *Output formats* on the previous page.
5. (Optional) Set the **Scale Z**, **Swap X/Y**, and **Keep Aspect Ratio** options.
Availability of these options depends on the output format you have chosen. For more information, see *Output options* above.
6. Click **Convert**.
The converter converts the input files.
The converted file will be in the same directory as the input file. It will also have the same name as the input file but with a different file extension. The converted file name is displayed in the **Output File** field.

CSV File Format

The CSV Converter tool can convert from the CSV format that Gocator can export to several other formats. If you want to work with the exported file directly, use the following information.

An exported CSV file contains a series of "sections." Each section begins with a row containing the name of the section, and ends with a row containing the string "End." An empty line separates each section.

Each section usually contains one or more subsections. Each subsection has a header row containing a list of field names, followed by one or more rows of data. There is usually no empty line between the subsections.

Other structures within sections are possible.

Example:

```
Info
CSV Version,Sensor Count,Trigger Mode,...
2,1,0,32000.00000,...
End
```

```
DeviceInfo
ID,Model,Version,...
13434,311320-2M-01,4.8.2.29,...
End
```

```
Ranges
...
End
```

Info

This section contains basic system information. It has one header row and one value row. The fields are described below:

Info Fields

Field	Description
CSV Version	Version of the CSV file format.
Sensor Count	Number of sensors in the system.
Trigger Mode	Trigger source: 0 – Time 1 – Encoder 2 – Digital input 3 – Software
Trigger Rate	Frame rate for time trigger (Hz).
Trigger Delay Domain	Output delay domain: 0 – Time (µs) 1 – Encoder (mm)
Trigger Delay	Output delay (µs or mm, depending on delay domain defined above).
Operation Mode	The scan mode.
XResolution	System X resolution (mm).
YResolution	System Y resolution (mm).

Field	Description
ZResolution	System Z resolution (mm).
Yspeed	Y Speed (mm/s).
Layout	Sensor orientation: 0 – Normal (single-sensor system) / Wide (dual-sensor system) 1 – Opposite 2 – Reverse 3 – Grid

DeviceInfo

This section contains information about each device in the system. There is one header row, and one value row per device.

DeviceInfo Fields

Field	Description
ID	Device serial number
Model	Device part number
Version	Firmware version
Exposure Mode	Exposure mode: 0 – Single exposure 1 – Multiple exposures 2 – Dynamic exposure
Exposure 0 through Exposure 4	Multiple exposures
Exposure Min	Dynamic exposure min
Exposure Max	Dynamic exposure max
FOV X	Active area X
FOV Y	Active area Y
FOV Z	Active area Z
FOV Width	Active area width
FOV Height	Active area length (Y). (Note difference in terminology.)
FOV Depth	Active area height (Z). (Note difference in terminology.)
Transform X	Transform X offset (mm)
Transform Y	Transform Y offset (mm)
Transform Z	Transform Z offset (mm)
Transform X Angle	Transform X Angle (degrees)
Transform Y Angle	Transform Y angle (degrees)
Transform Z Angle	Transform Z angle (degrees)

RecordingFilter

This section lists the filters used during recording. Unlike the other sections, it contains multiple sub-

sections within, separated by spaces (but not the “End” keyword).

Example:

```
RecordingFilter
Section1 Param 1, Section1 Param2
value, value
Section2 Param 1
value
Section3 Param1, Section3 Param2
value
End
```

Each section will be described by a separate table below. They appear in the same order as documented.

RecordingFilter Fields

Field	Description
Condition Combination Type	Any or All

"Any Measurement" Filter Fields

Field	Description
Type	Any Measurement
Enabled	Whether or not is enabled. Yes/No
Result	Accepted result type: Pass/Fail/Invalid/Valid

"Any Data" Filter Fields

Field	Description
Type	Any Data
Enabled	Whether or not is enabled: Yes/No
Threshold Case	How to threshold: At or Above, or Below
Range Count Threshold	Threshold value (point count)

"Measurement" Filter Fields

Field	Description
Type	Measurement
Enabled	Whether or not is enabled: Yes/No
Result	Accepted result type: Pass/Fail/Invalid/Valid
Selection ID	First measurement ID

Ranges

This section describes single-point range data. It has two sub-sections: attributes and data.

The attribute section has only one row of data

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (us)
Y	Y value (mm)
Axis	Axis: Z (range) or I (Intensity)
Value	Range value (mm) or intensity (count)

Profile

This section describes uniform (or resampled) profile data, which is produced when the sensor is in Profile mode and uniform spacing is enabled. It has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)

Field	Description
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (µs)
Y	Y value (mm)
Axis	Axis: Z (range) or I (Intensity)
(x values)	Each column in header is a resampled X position Each column in data is the range (mm) or intensity (count)

RawProfile

This section describes point cloud profile data (or unresampled / raw data), which is produced when the sensor is in Profile mode and uniform spacing is disabled. It has two sub-sections: attributes and data.

The attribute section has only one row of data.

Attribute Section Fields

Field	Description
Frame Count	Total number of frames
Column Count	Number of columns
X Offset	X offset (mm)
Y Offset	Y offset (mm)
Z Offset	Z offset (mm)

The data section has one or more rows of data per frame (for example, range and intensity).

Data Section Fields

Field	Description
Frame	Frame index
Source	Source (for example, 0 for Top)
Time	Stamp time
Encoder	Stamp encoder
Z Encoder	Stamp encoder Z
Inputs	Stamp inputs
Exposure	Stamp exposure (µs)
Y	Y value (mm)
Axis	Axis: X, Z, or I (Intensity)
(x values)	Each column in header is an index. Each column in data is the X/Z value (mm) or intensity (count)

Troubleshooting

Review the guidance in this chapter if you are experiencing difficulty with a Gocator sensor system. If the problem that you are experiencing is not described in this section, see *Return Policy* on page 463.

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 Discovery Tool* on page 401 for more information.

When attempting to log in, the password is not accepted.

- See *Sensor Discovery Tool* on page 401 for steps to reset the password.

Laser Ranging

When the Start button or the Snapshot button is pressed, the sensor does not emit laser light.

- Ensure that the sticker covering the laser emitter window (normally affixed to new sensors) has been removed.
- The laser safety input signal may not be correctly applied. See *Specifications* on page 411 for more information.
- The exposure setting may be too low. See *Exposure* on page 95 for more information on configuring exposure time.
- Use the Snapshot button instead of the Start button to capture a laser range. If the laser flashes when you use the **Snapshot** button, but not when you use the **Start** button, then the problem could be related to triggering. See *Triggers* on page 85 for information on configuring the trigger source.

The sensor emits laser light, but the Range Indicator LED does not illuminate and/or points are not displayed in the Data Viewer.

- Verify that the measurement target is within the sensor's field of view and measurement range. See *Specifications* on page 411 to review the measurement specifications for your sensor model.

- Check that the exposure time is set to a reasonable level. See *Exposure* on page 95 for more information on configuring exposure time.

Performance

The sensor CPU level is near 100%.

- Consider reducing the speed. If you are using a time or encoder trigger source, see *Triggers* on page 85 for information on reducing the speed. If you are using an external input or software trigger, consider reducing the rate at which you apply triggers.
- Review the measurements that you have programmed and eliminate any unnecessary measurements.

Specifications

The following sections describe the specifications of Gocator sensors and connectors, as well as Master hubs.

Sensors

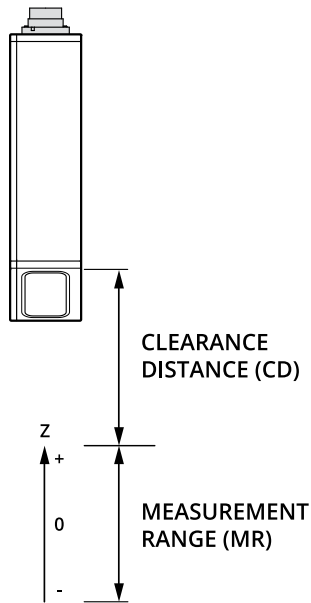
The following sections provide the specifications of Gocator sensors.

Gocator 1300 Series

The Gocator 1300 series consists of the following models:

MODEL	1320	1340	1350	1365	1370	1380	1390
Clearance Distance (mm)	40	162.5	200	562	237.5	127	500
Measurement Range (MR) (mm)	20	95	200	375	412.5	1651	2000
Linearity Z (+/- % of MR)	0.05	0.05	0.05	0.11	0.07	0.18	0.10
Linearity Z (+/- mm)	0.010	0.05	0.100	0.4	0.3	3.0	2.0
Resolution Z (mm)	0.0004 - 0.0004	0.0005 - 0.0010	0.0015 - 0.0035	0.0025 - 0.0040	0.0025 - 0.0070	0.0100 - 0.0450	0.0250 - 0.0600
Spot Size (mm)	0.11	0.37	0.50	1.80	0.90	2.60	2.60
Recommended Laser Class	3R	3B	3B	3B	3B	3B	3B
Other Laser Class	3B	2M, 3R					
Recommended Package Dimensions (mm)	Side Mount (3R) 30x120x149	Side Mount 30x120x149	Side Mount 30x120x149	Side Mount 30x120x220	Side Mount 30x120x149	Side Mount 30x120x149	Side Mount 30x120x277
Other Package Dimensions (mm)	Top Mount (3B) 49x75x162	Top Mount 49x75x162					
Weight (kg)	0.8	0.8	0.75 / 0.8	1.1	0.8	0.8	1.4

The following diagram illustrates some of the terms used in the table above.



Optical models, laser classes, and packages can be customized. Contact LMI for more details.

Specifications stated are based on standard laser classes. Linearity Z and Resolution Z may vary for other laser classes.

All specification measurements are performed on LMI's standard calibration target (a diffuse, painted white surface).

Linearity Z is the worst case difference in average height measured, compared to the actual position over the measurement range.

Resolution Z is the maximum variability of height measurements across multiple frames, with 95% confidence.

See *Resolution and Accuracy* on page 47 for more information.

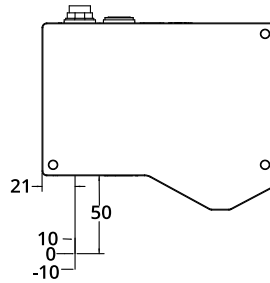
ALL 1300 SERIES MODELS

Scan Rate	32kHz
Interface	Gigabit Ethernet
Inputs	Differential Encoder, Laser Safety Enable, Trigger
Outputs	2x Digital Output, RS-485 Serial, Selcom Serial, 1x Analog Output (4 - 20 mA)
Input Voltage (Power)	+24 to +48 VDC (13 Watts); Ripple +/- 10%
Housing	Gasketed Aluminum Enclosure, IP67
Operating Temp.	0 to 50° C
Storage Temp.	-30 to 70 ° C

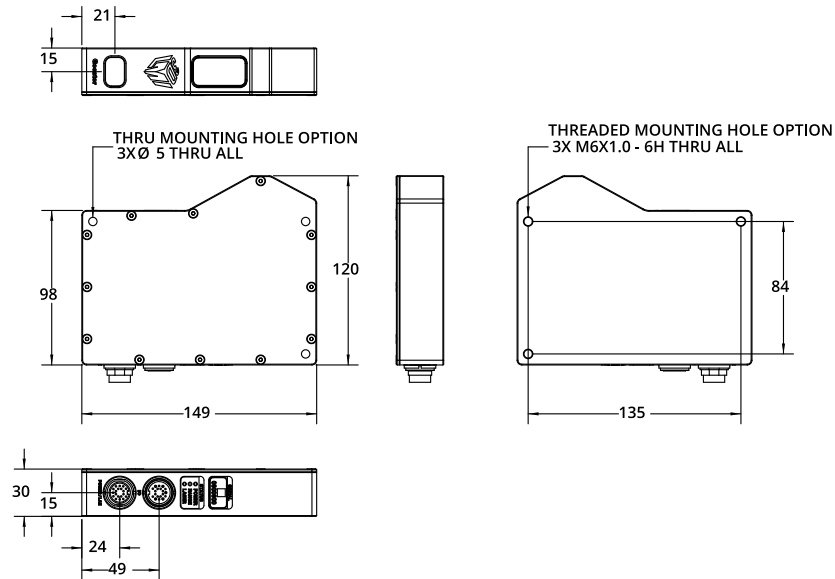
Mechanical dimensions, CD/MR, and the envelope for each sensor model are illustrated on the following pages.

Gocator 1320 (Side Mount Package)

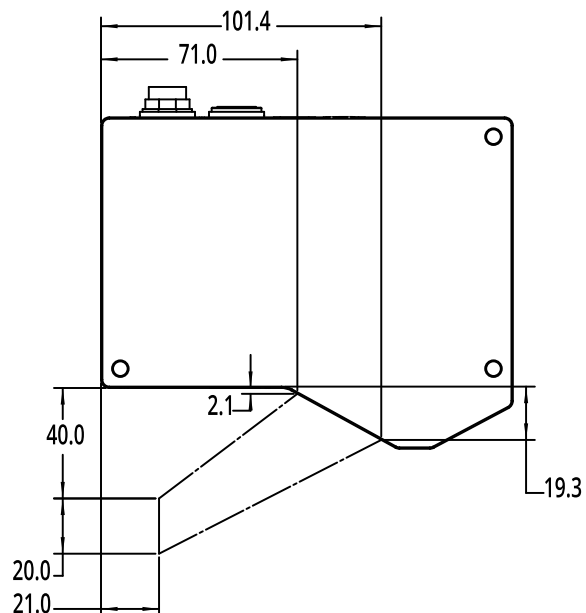
Measurement Range



Dimensions

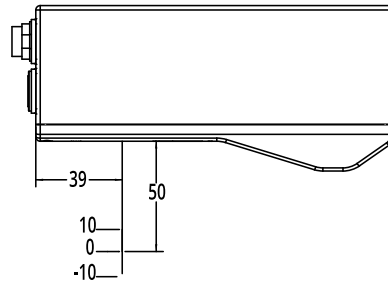


Envelope

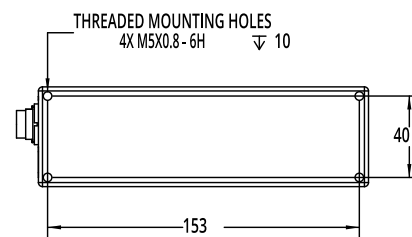
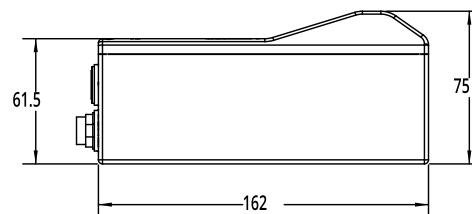
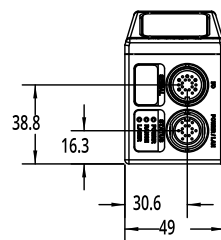
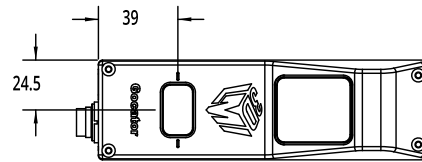


Gocator 1320 (Top Mount Package)

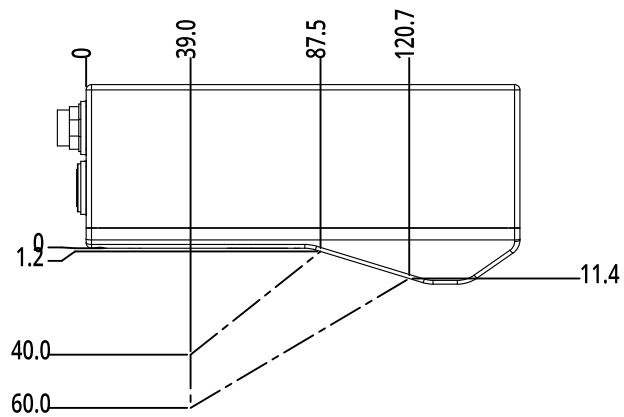
Field of View / Measurement Range



Dimensions

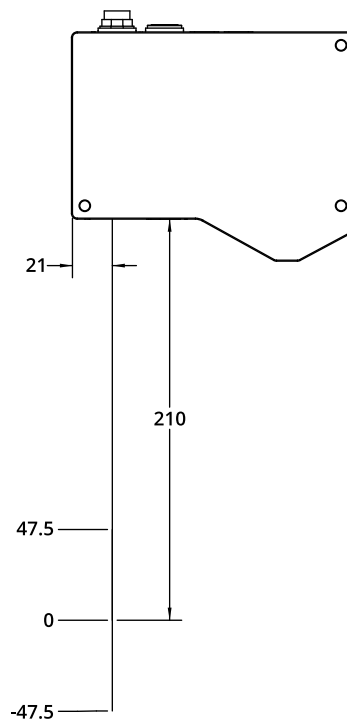


Envelope

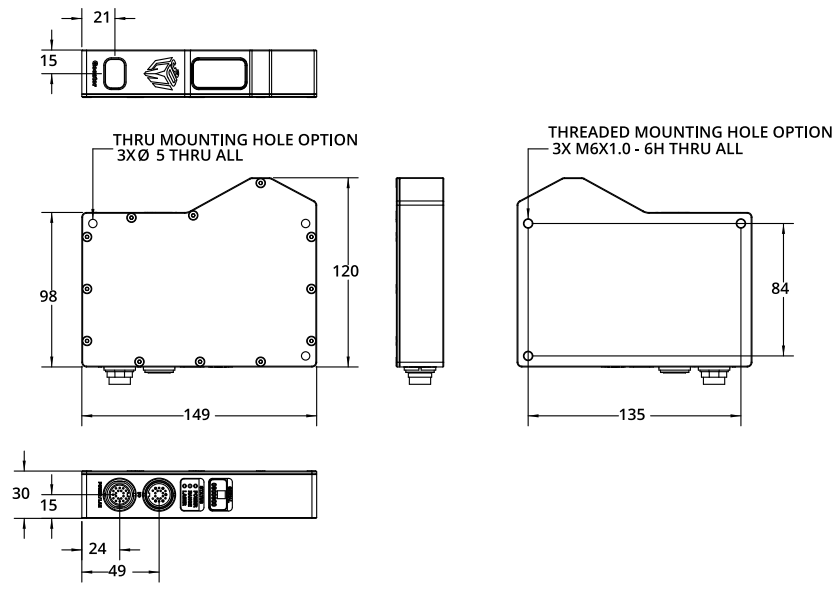


Gocator 1340 (Side Mount Package)

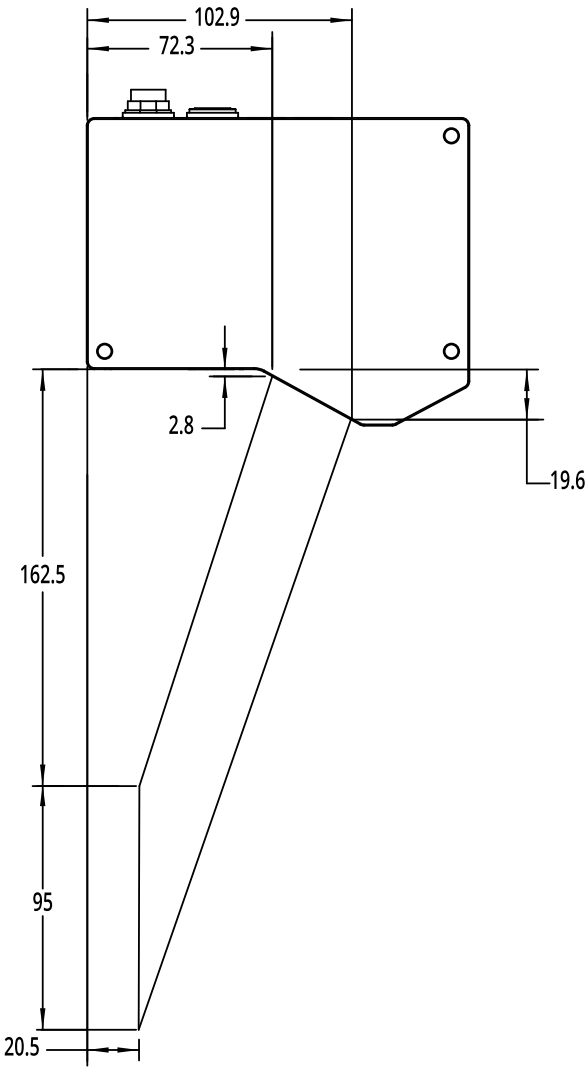
Measurement Range



Dimensions

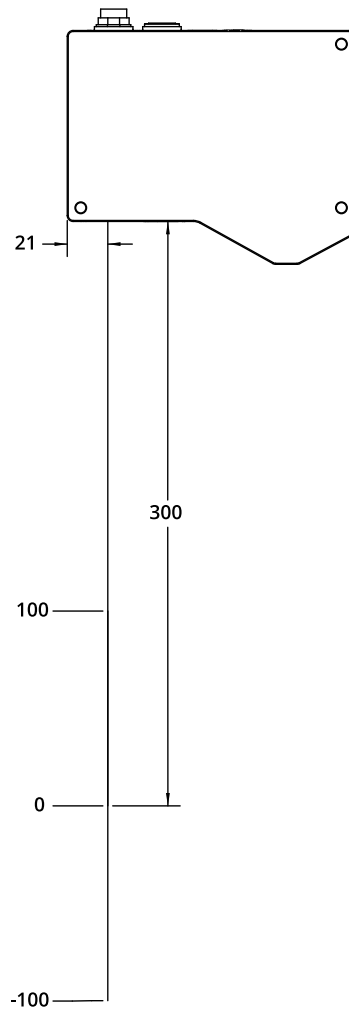


Envelope

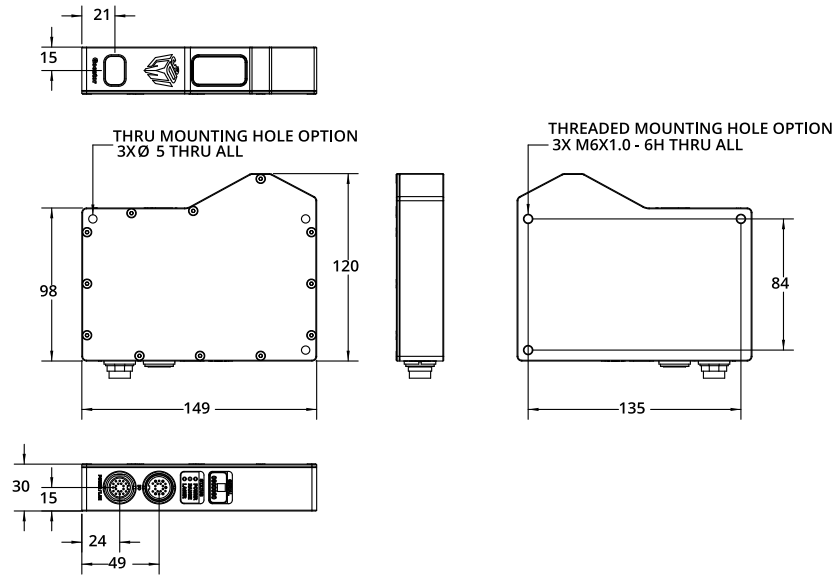


Gocator 1350 (Side Mount Package)

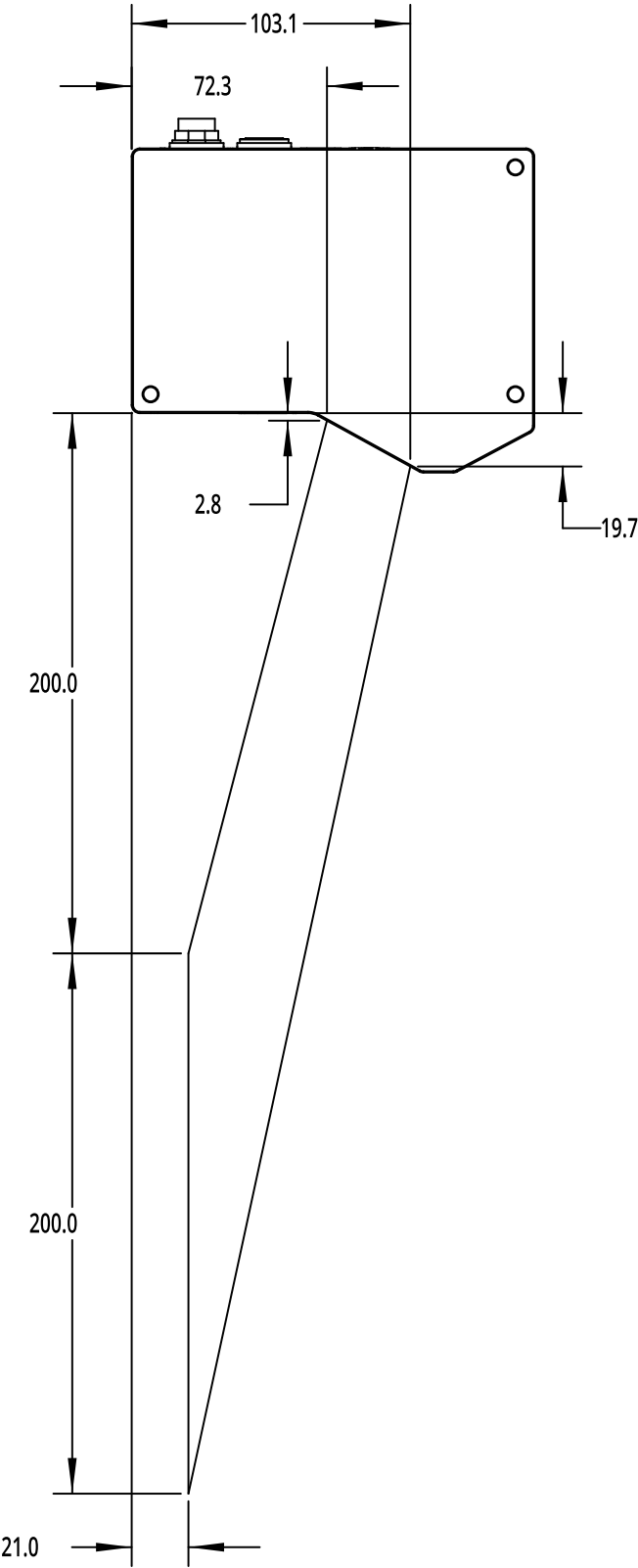
Measurement Range



Dimensions

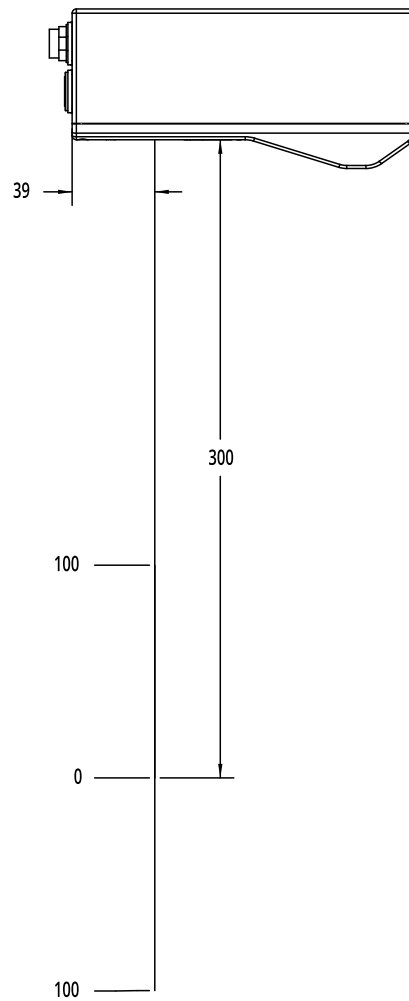


Envelope

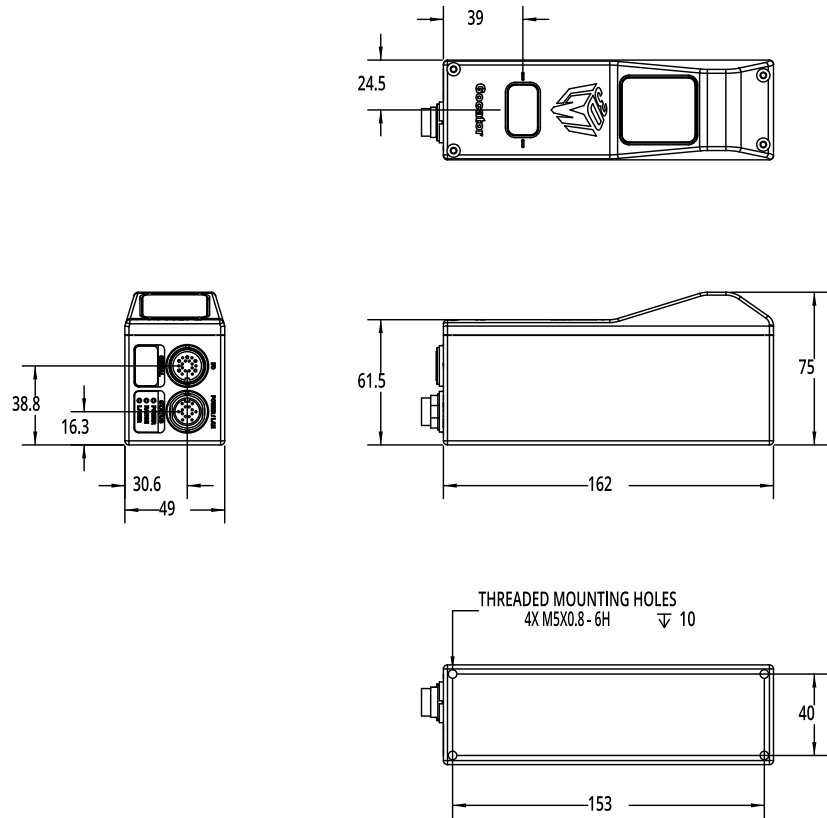


Gocator 1350 (Top Mount Package)

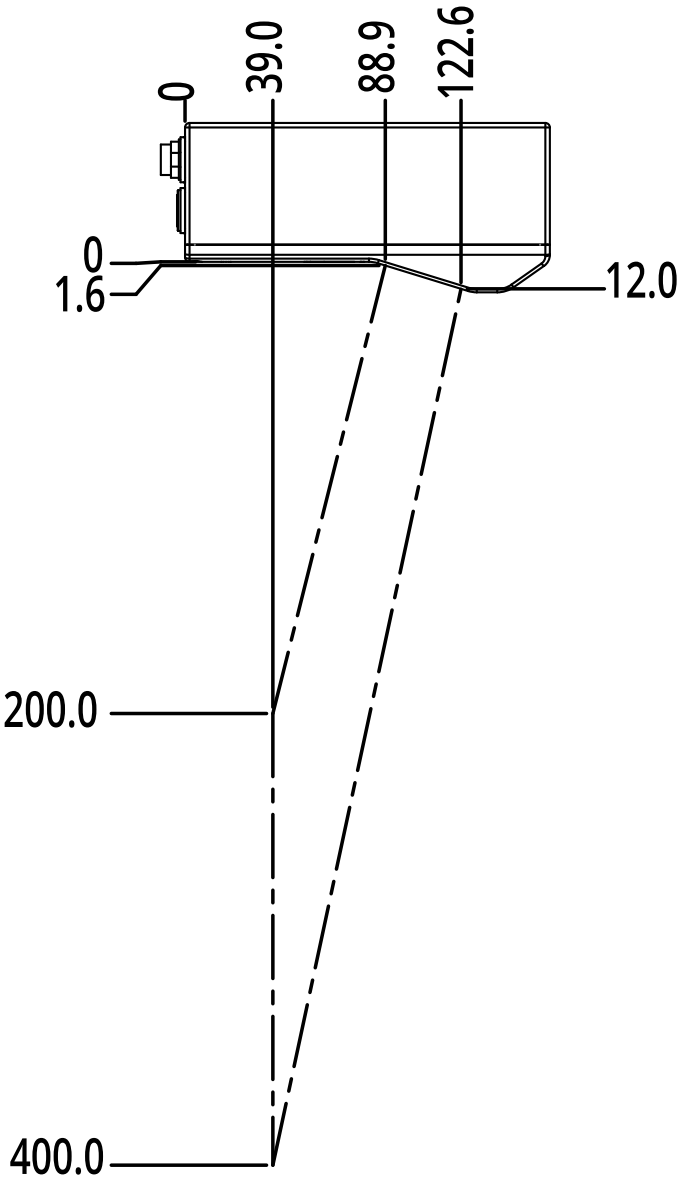
Measurement Range



Dimensions

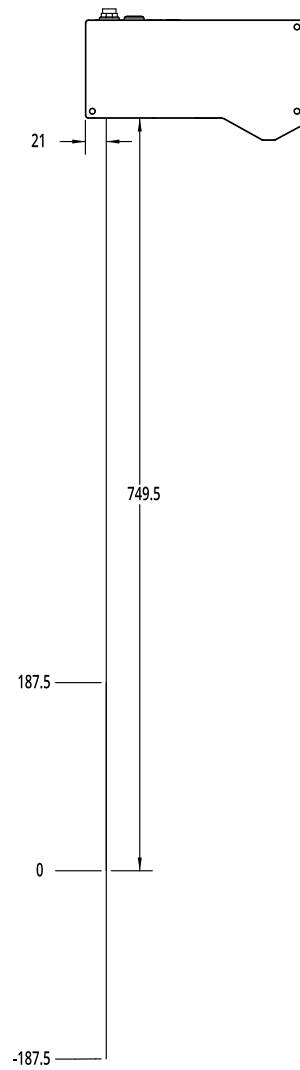


Envelope

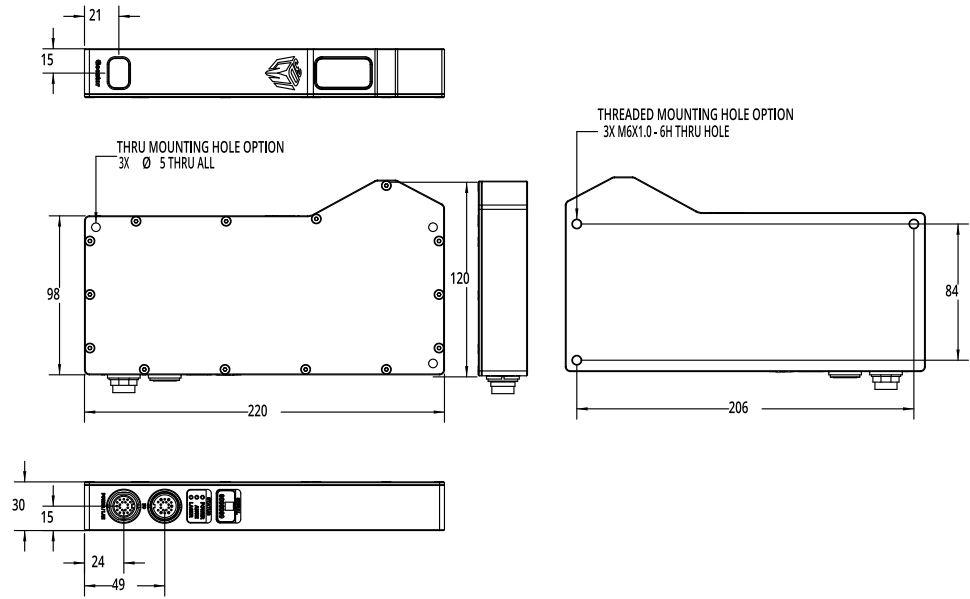


Gocator 1365 (Side Mount Package)

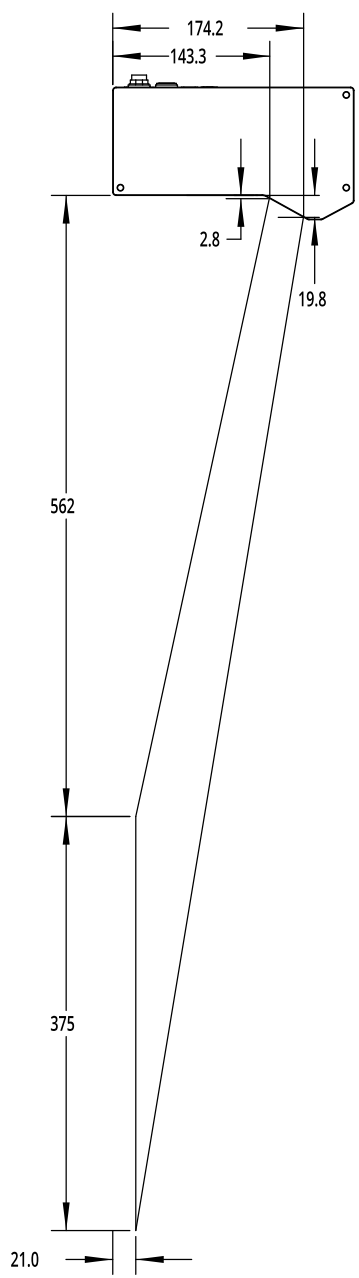
Measurement Range



Dimensions

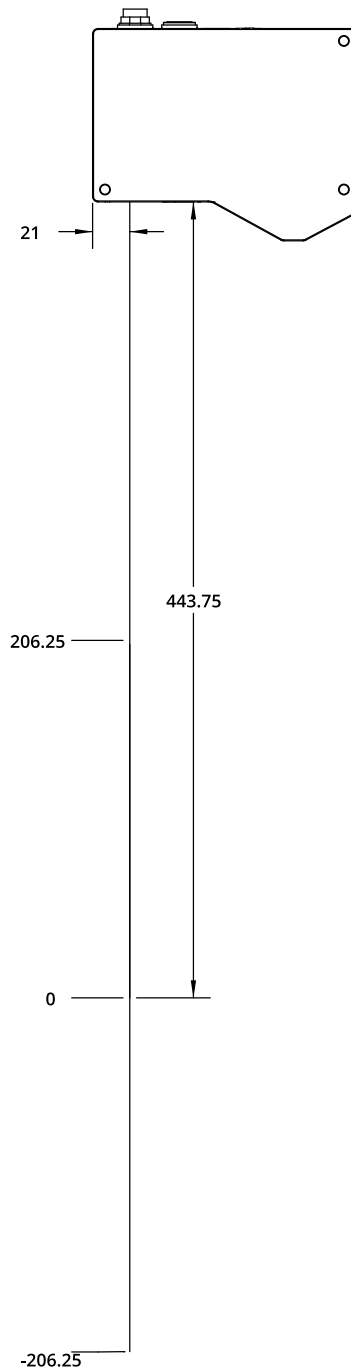


Envelope

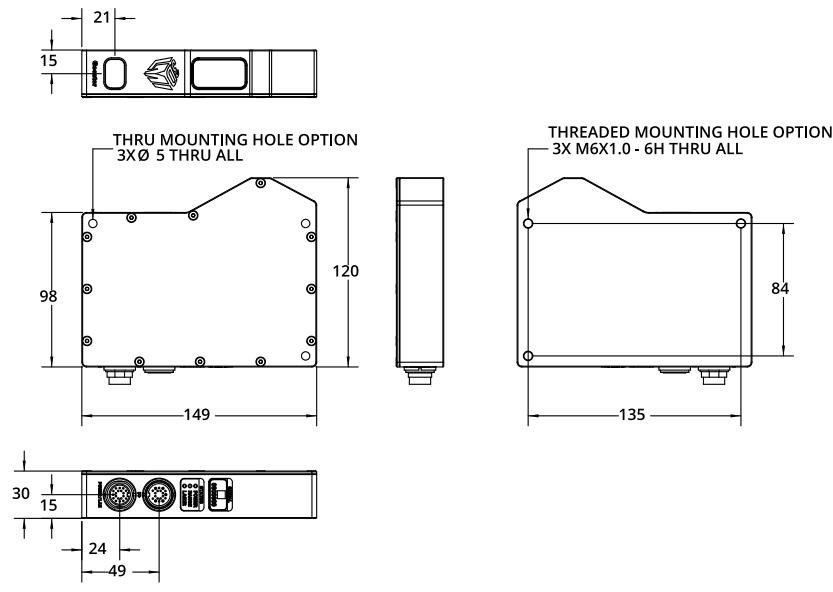


Gocator 1370 (Side Mount Package)

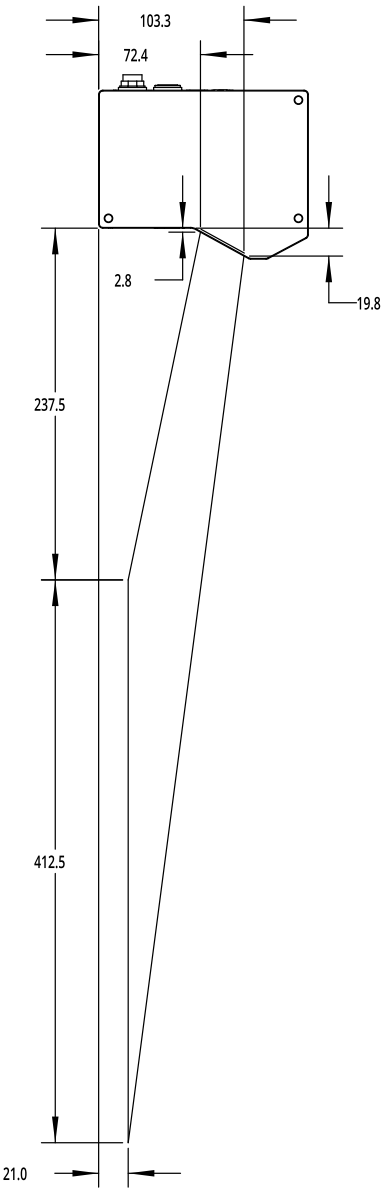
Measurement Range



Dimensions

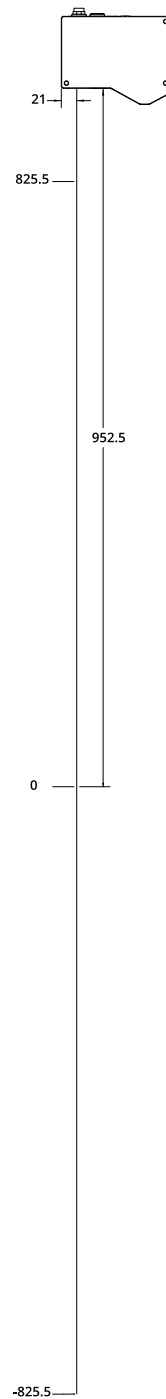


Envelope

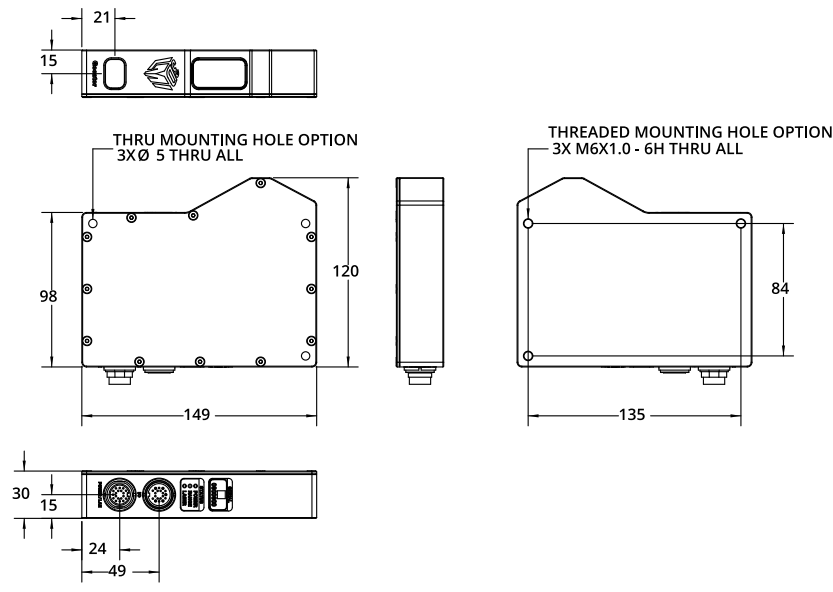


Gocator 1380 (Side Mount Package)

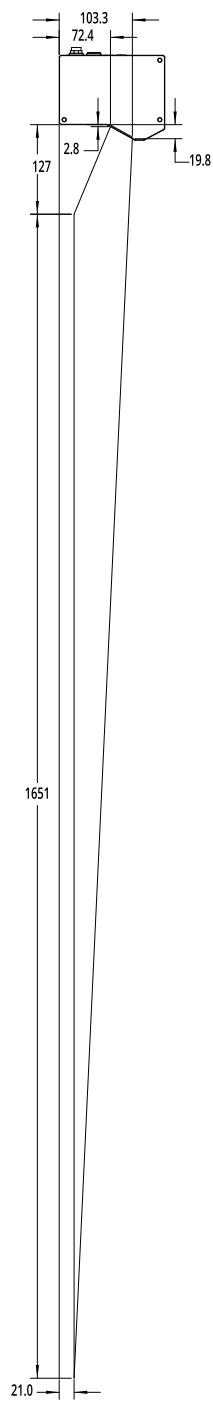
Measurement Range



Dimensions

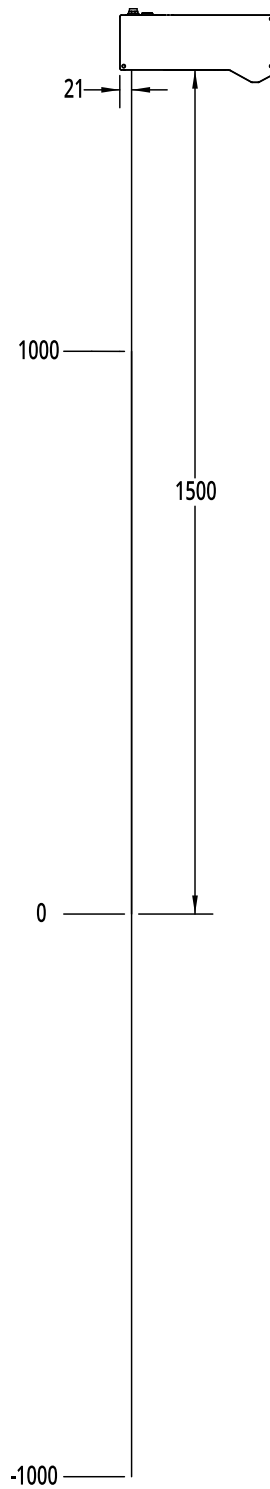


Envelope

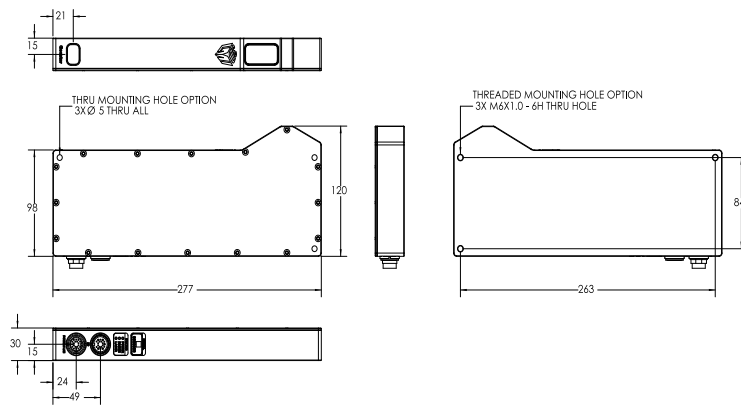


Gocator 1390 (Side Mount Package)

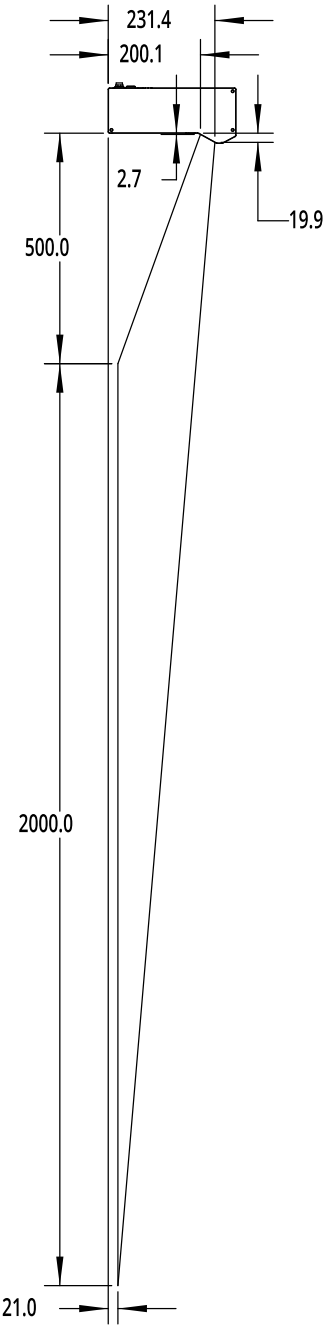
Measurement Range



Dimensions



Envelope



Sensor Connectors

The following sections provide the specifications of the connectors on Gocator sensors.

Gocator Power/LAN Connector

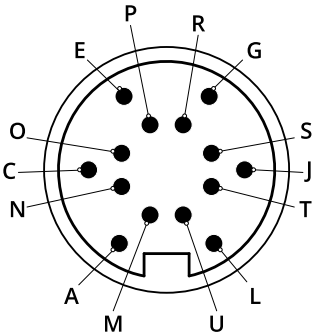
The Gocator Power/LAN connector is a 14 pin, M16 style connector that provides power input, laser safety input and Ethernet.

 This connector is rated IP67 only when a cable is connected or when a protective cap is used.

This section defines the electrical specifications for Gocator Power/LAN Connector pins, organized by function.

Gocator Power/LAN Connector Pins

Function	Pin	Lead Color on Cordset
GND_24-48V	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 1300 Series* on page 411

Power requirements

Function	Pins	Min	Max
DC_24-48V	A	24 V	48 V
GND_24-48VDC	L	0 V	0 V

Laser Safety Input

The Safety_in+ signal should be connected to a voltage source in the range listed below. The Safety_in- signal should be connected to the ground/common of the source supplying the Safety_in+.

Laser safety requirements

Function	Pins	Min	Max
Safety_in+	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 I/O Connector

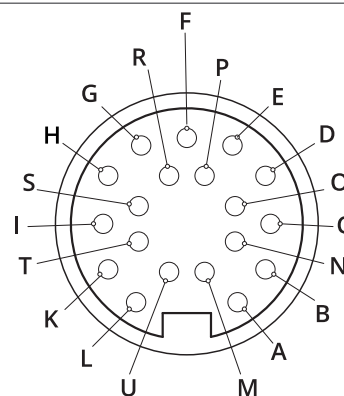
The Gocator I/O connector is a 19 pin, M16 style connector that provides encoder, digital input, digital outputs, serial output, and analog output signals.

 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
Serial_out2+	E	Blue / Black
Serial_out2-	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

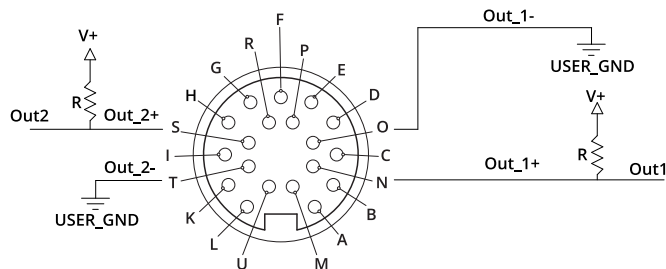
The grounding shield should be mounted to the earth ground.

Digital Outputs

Each Gocator sensor has two optically isolated outputs. Both outputs are open collector and open emitter, which 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 μ s
Out_2	S, T	40 mA	70 V	20 μ s

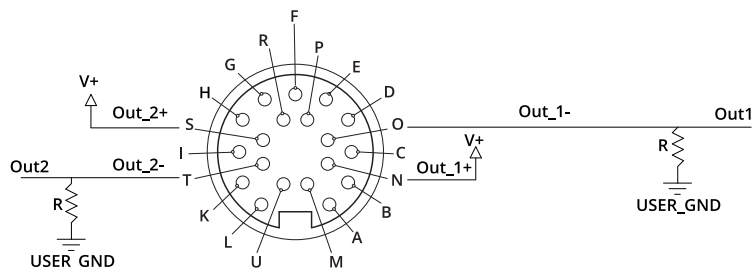


The resistors shown above are calculated by $R = (V+) / 2.5 \text{ mA}$.

The size of the resistors is determined by power = $(V+)^2 / R$.

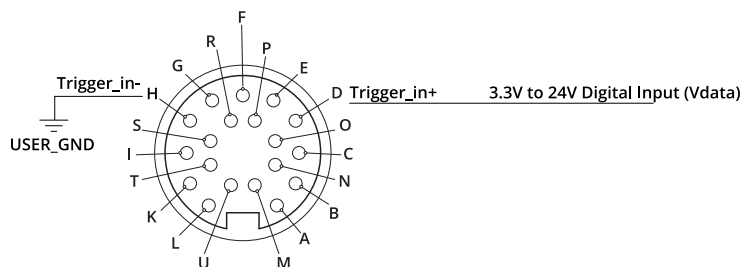
Inverting Outputs

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-. For resistor selection, see above.



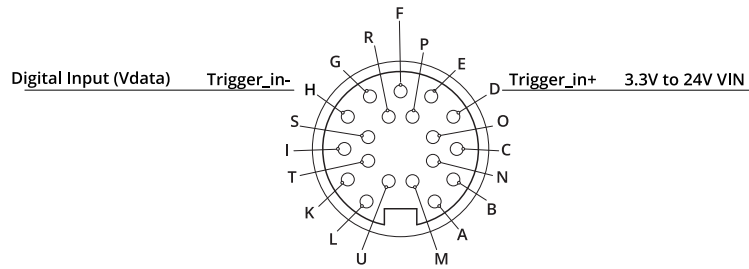
Digital Input

Every Gocator sensor has a single optically isolated input. To use this input without an external resistor, supply 3.3 - 24 V to the positive pin and GND to the negative.



Active High

If the supplied voltage is greater than 24 V, connect an external resistor in series to the positive. The resistor value should be $R = [(V_{in}-1.2V)/10mA]-680$.



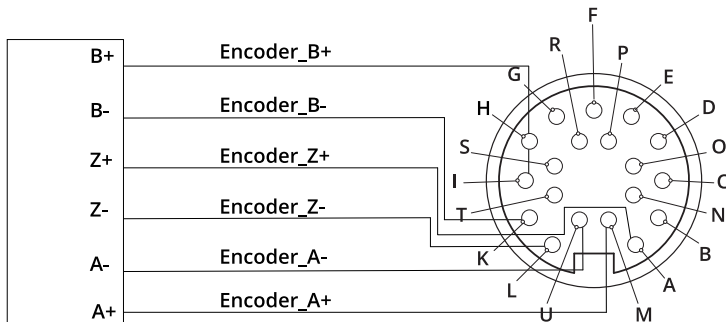
Active Low

To assert the signal, the digital input voltage should be set to draw a current of 3 mA to 40 mA from the positive pin. The current that passes through the positive pin is $I = (V_{in} - 1.2 - V_{data}) / 680$. 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 μ s

Encoder Input

Encoder input is provided by an external encoder and consists of three RS-485 signals. These signals are connected to Encoder_A, Encoder_B, and Encoder_Z.



Function	Pins	Common Mode Voltage		Differential Threshold Voltage			Max Data Rate
		Min	Max	Min	Typ	Max	
Encoder_A	M, U	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Encoder_B	I, K	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz
Encoder_Z	A, L	-7 V	12 V	-200 mV	-125 mV	-50 mV	1 MHz

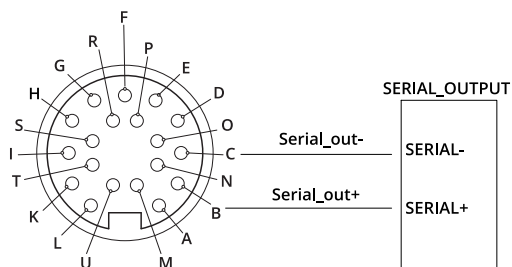
 Gocator only supports differential RS485 signalling. Both + and - signals must be connected.

 Encoders are normally specified in *pulses* per revolution, where each pulse is made up of the four quadrature *signals* (A+ / A- / B+ / B-). Because Gocator reads each of the four quadrature signals, you should choose an encoder accordingly, given the resolution required for your application.

Serial Output

Serial RS-485 output is connected to Serial_out as shown below.

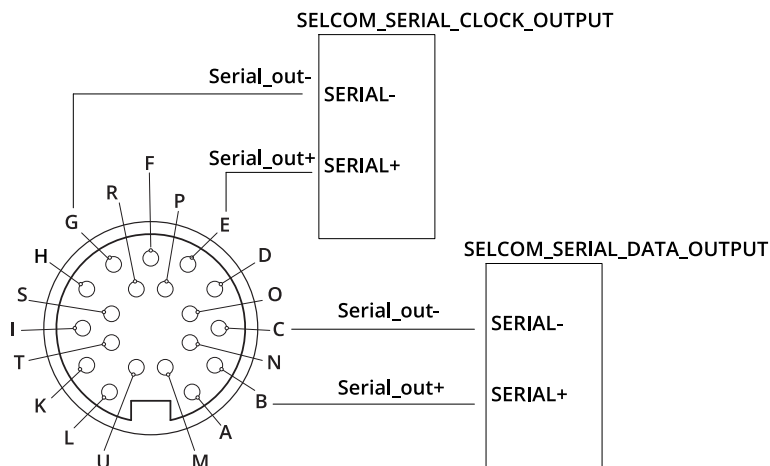
Function	Pins
Serial_out	B, C



Selcom Serial Output

Serial RS-485 output is connected to Serial_out and Serial_out2 as shown below.

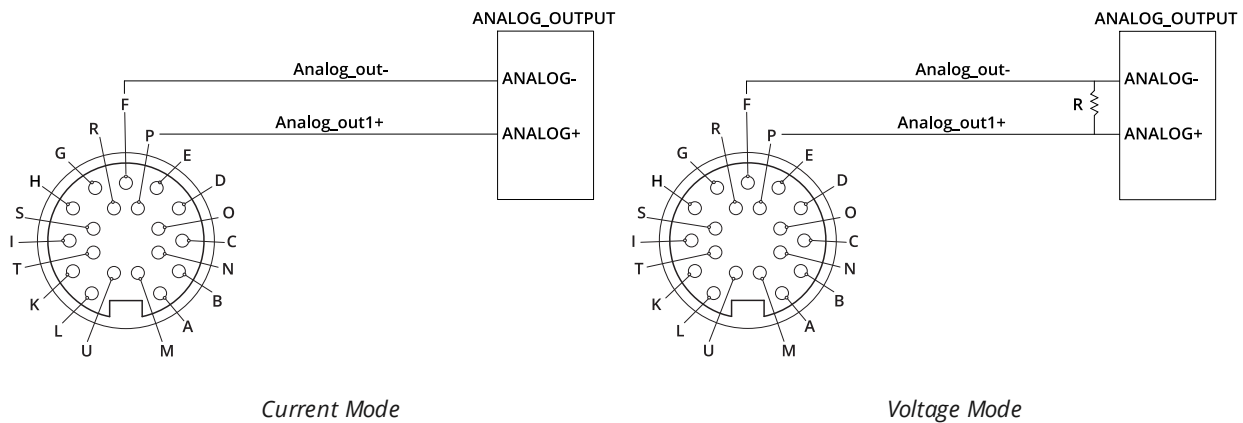
Function	Pins
Serial_out (data)	B, C
Serial_out2 (clock)	E, G



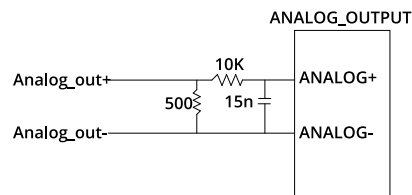
Analog Output

The Sensor I/O Connector defines one analog output interface: Analog_out.

Function	Pins	Current Range
Analog_out	P, F	4 – 20 mA



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.



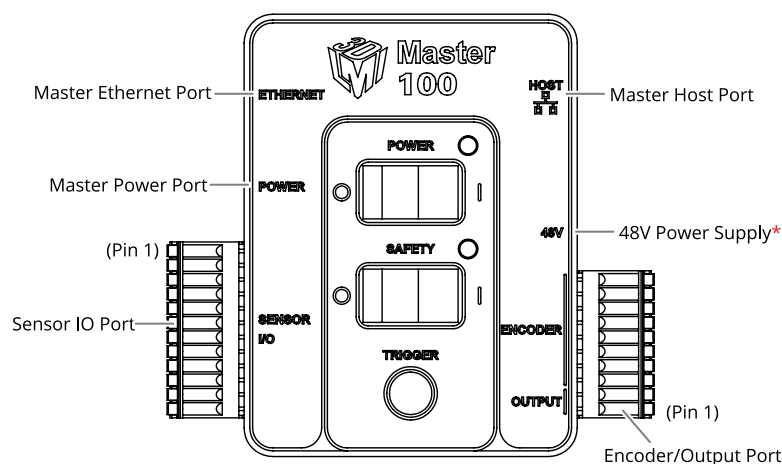
Master Network Controllers

The following sections provide the specifications of Master network controllers.

For information on maximum external input trigger rates, see *Maximum Input Trigger Rate* on page 92.

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

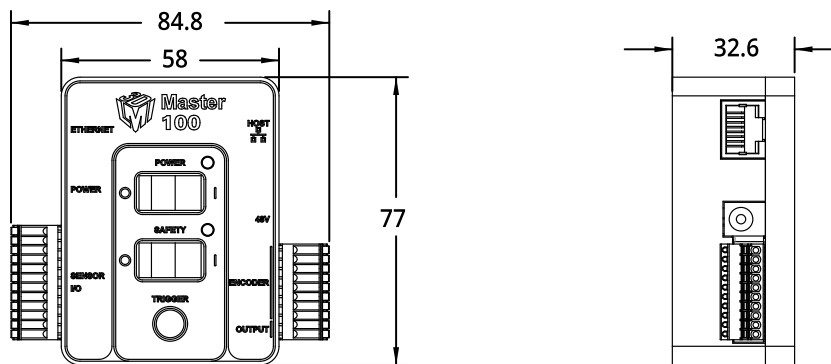
Gocator I/O Pin	Master Pin	Conductor Color
Encoder_A+	1	White/Brown & Black
Encoder_A-	2	Brown/Black
Encoder_Z+	3	White/Green & Black
Encoder_Z-	4	Green/Black
Trigger_in+	5	Grey
Trigger_in-	6	Pink
Out_1-	7	Blue
Out_1+	8	Red
Encoder_B+	11	Black
Encoder_B-	12	Violet

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

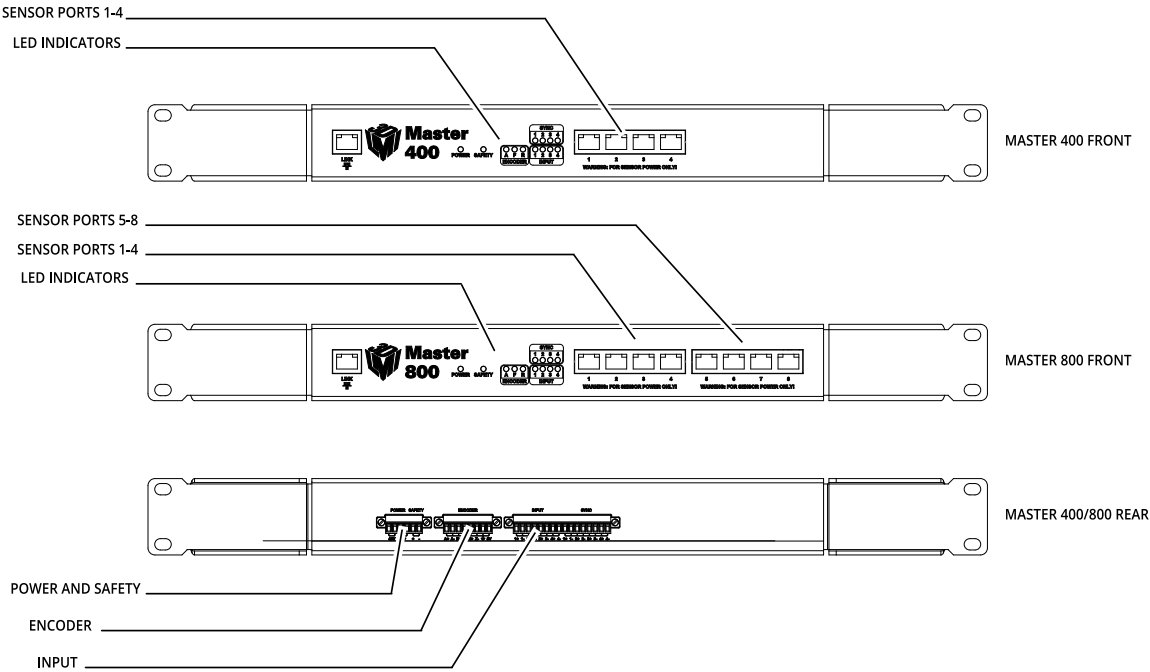


Master 400/800

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.



The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4
Safety Control+	5
Safety Control-	6



The 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 of 24 VDC to 48 VDC across the pin to enable the laser.

Input (16 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12
Reserved	13
Reserved	14
Reserved	15
Reserved	16



The Input 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

Specification	Value
Power Supply Voltage	+48 VDC
Power Supply Current (Max.)	10 A

Specification	Value
Power Draw (Min.)	5.76 W
Safety Input Voltage Range	+24 VDC to +48 VDC
Encoder Signal Voltage	Differential (5 VDC)
Digital Input Voltage Range	Logical LOW: 0 to +0.1 VDC Logical HIGH: +3.3 to +24 VDC

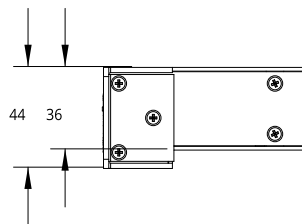
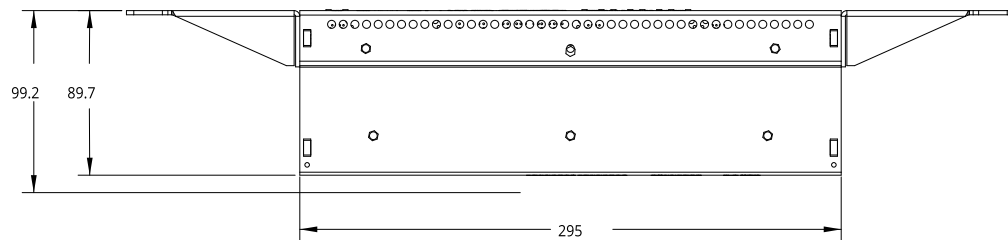
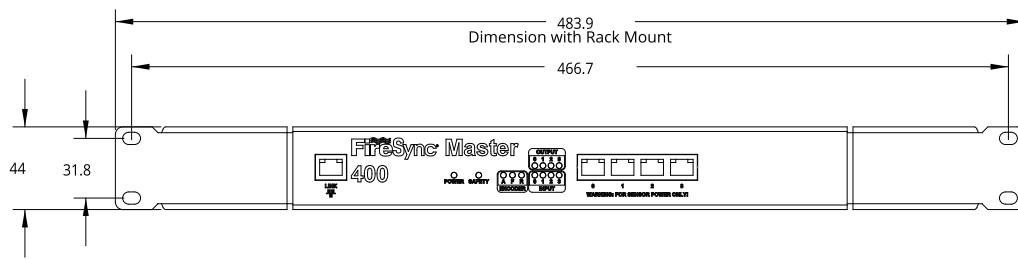
 When using a Master hub, the chassis must be well grounded.

 The 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 that 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 810/2410

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.

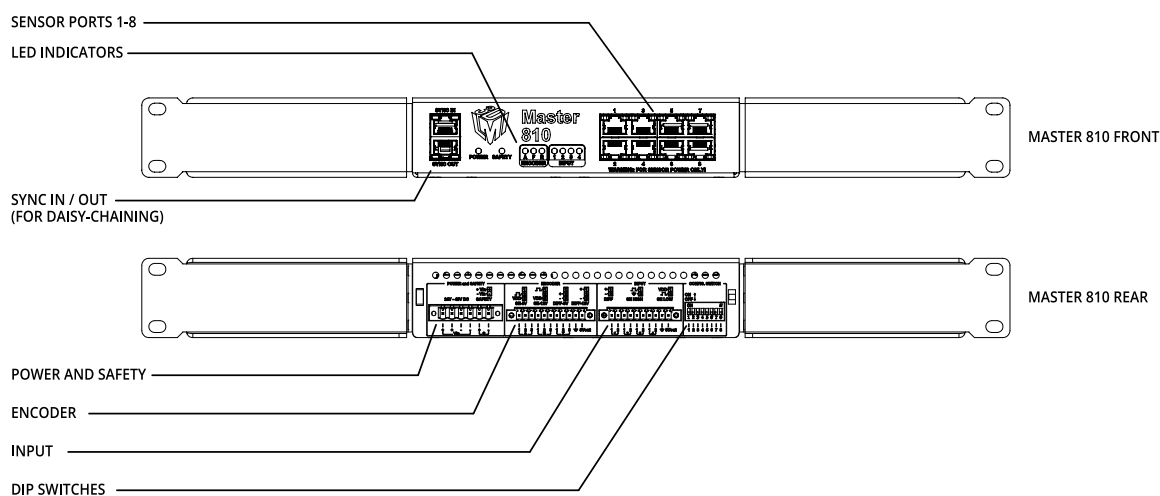
Master 810 and 2410 can be mounted to DIN rails using the provided adapters (for more information, see *Installing DIN Rail Clips: Master 810 or 2410* on page 35). The units are also provided with removable adapters for 1U rack mounting; the mounting holes for this option are compatible with older Master models (400/800/1200/2400).

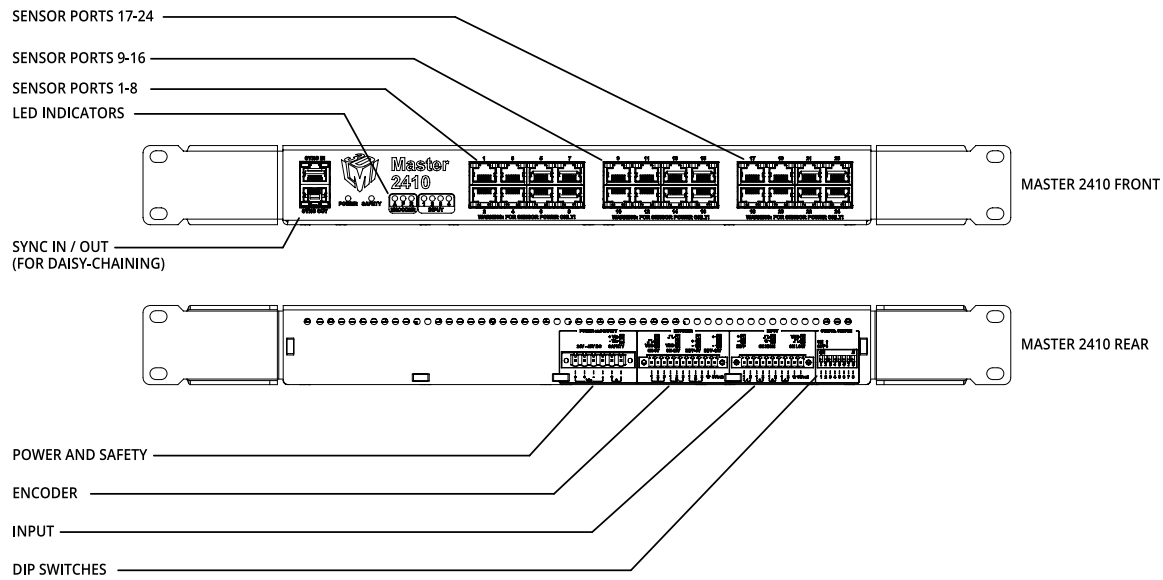


The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.

Master 2410 can currently be used with encoders with a maximum quadrature frequency of 300 kHz.

Master 810 can be configured to work with a maximum encoder quadrature frequency of 6.5 MHz. For more information, see *Configuring Master 810* on page 36.





For information on configuring the DIP switches, see *Configuring Master 810* on page 36.

Power and Safety (6 pin connector)

Function	Pin
Power In+	1
Power In+	2
Power In-	3
Power In-	4
Safety Control+	5
Safety Control-	6

 The 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 of 24 VDC to 48 VDC across the pin to enable the laser.

 On earlier revisions of Master 810 and Master 2410, the inputs are labeled 0-3.

Input (10 pin connector)

Function	Pin
Input 1 Pin 1	1
Input 1 Pin 2	2
Reserved	3
Reserved	4
Reserved	5

Function	Pin
Reserved	6
Reserved	7
Reserved	8
GND (output for powering other devices)	9
+5VDC (output for powering other devices)	10

 The Input connector does not need to be wired up for proper operation.

 For Input connection wiring options, see *Input* on page 454.

Encoder (11 pin connector)

Function	Pin
Encoder_A_Pin_1	1
Encoder_A_Pin_2	2
Encoder_A_Pin_3	3
Encoder_B_Pin_1	4
Encoder_B_Pin_2	5
Encoder_B_Pin_3	6
Encoder_Z_Pin_1	7
Encoder_Z_Pin_2	8
Encoder_Z_Pin_3	9
GND (output for powering external devices)	10
+5VDC (output for powering external devices)	11

 For Encoder connection wiring options, see *Encoder* on the next page.

Electrical Specifications

Electrical Specifications

Specification	Value
Power Supply Voltage	+24 VDC to +48 VDC
Power Supply Current (Max.)*	Master 810: 9 A Master 2410: 25 A * Fully loaded with 1 A per sensor port.
Power Draw (Min.)	Master 810: 1.7 W Master 2410: 4.8 W
Safety Input Voltage Range	+24 VDC to +48 VDC
Encoder Signal Voltage	Differential (5 VDC, 12 VDC)

Specification	Value
	Single-Ended (5 VDC, 12 VDC) For more information, see <i>Encoder</i> below.
Digital Input Voltage Range	Single-Ended Active LOW: 0 to +0.8 VDC Single-Ended Active HIGH: +3.3 to +24 VDC Differential LOW: 0.8 to -24 VDC Differential HIGH: +3.3 to +24 VDC For more information, see <i>Input</i> on page 454.
	If the input voltage is above 24 V, use an external resistor, using the following formula: $R = [(V_{in} - 1.2V) / 10mA] - 680$

 When using a Master hub, the chassis must be well grounded.

 The power supply must be isolated from AC ground. This means that AC ground and DC ground are not connected.

 24 VDC power supply is only supported if all connected sensors support an input voltage of 24 VDC.

 The Power Draw specification is based on a Master with no sensors attached. Every sensor has its own power requirements that need to be considered when calculating total system power requirements..

Encoder

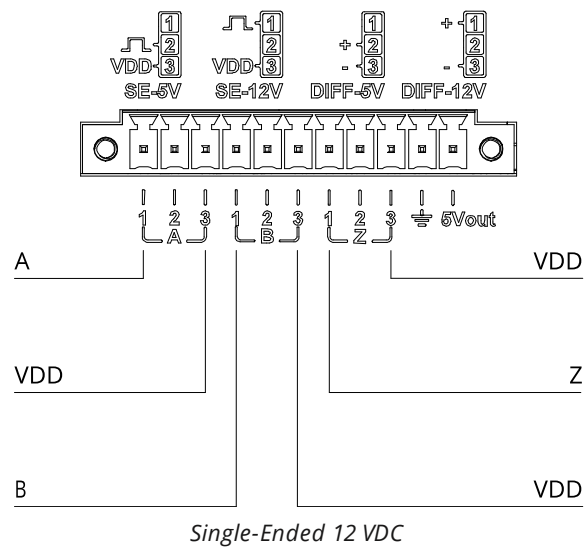
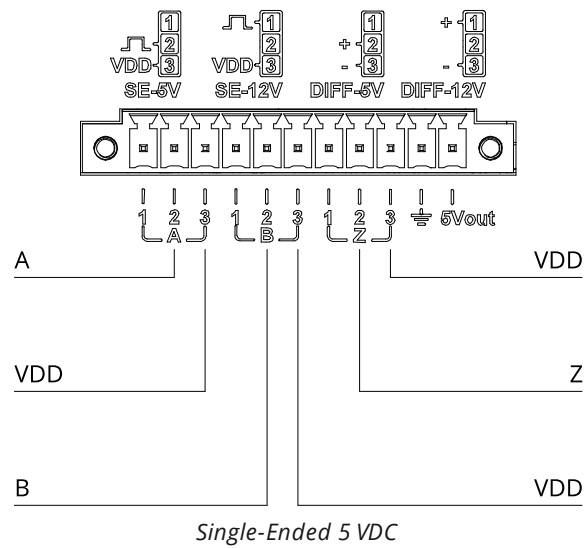
Master 810 and 2410 support the following types of encoder signals: Single-Ended (5 VDC, 12 VDC) and Differential (5 VDC, 12 VDC).

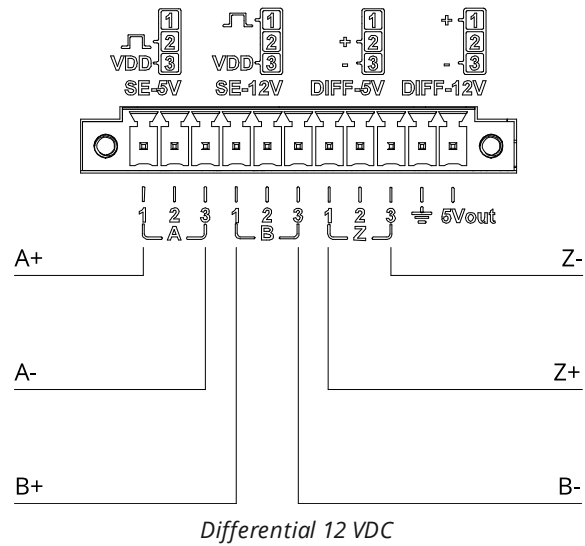
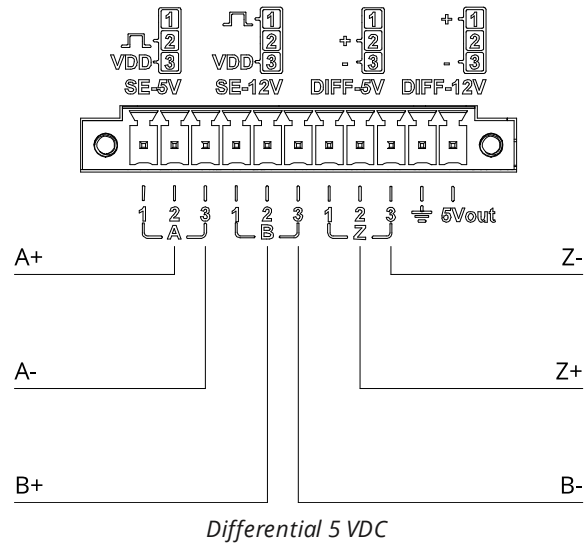
For 5 VDC operation, pins 2 and 3 of each channel are used.

For 12 VDC operation, pins 1 and 3 of each channel are used.

 The 5-volt encoder input supports up to 12 volts for compatibility with earlier Master network controllers. However, we strongly recommend connecting 12-volt output encoders to the appropriate 12-volt input to attain maximum tolerance.

To determine how to wire a Master to an encoder, see the illustrations below.





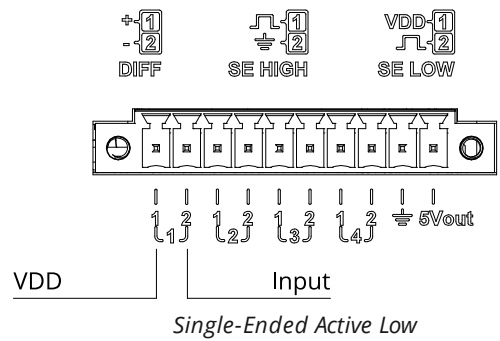
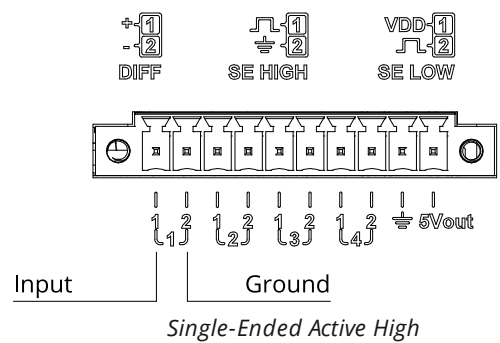
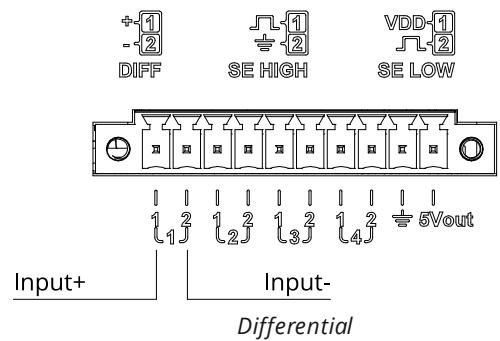
Input

Master 810 and 2410 support the following types of input: Differential, Single-Ended High, and Single-Ended Low.



Currently, Gocator only supports Input 0.

For digital input voltage ranges, see the table below.

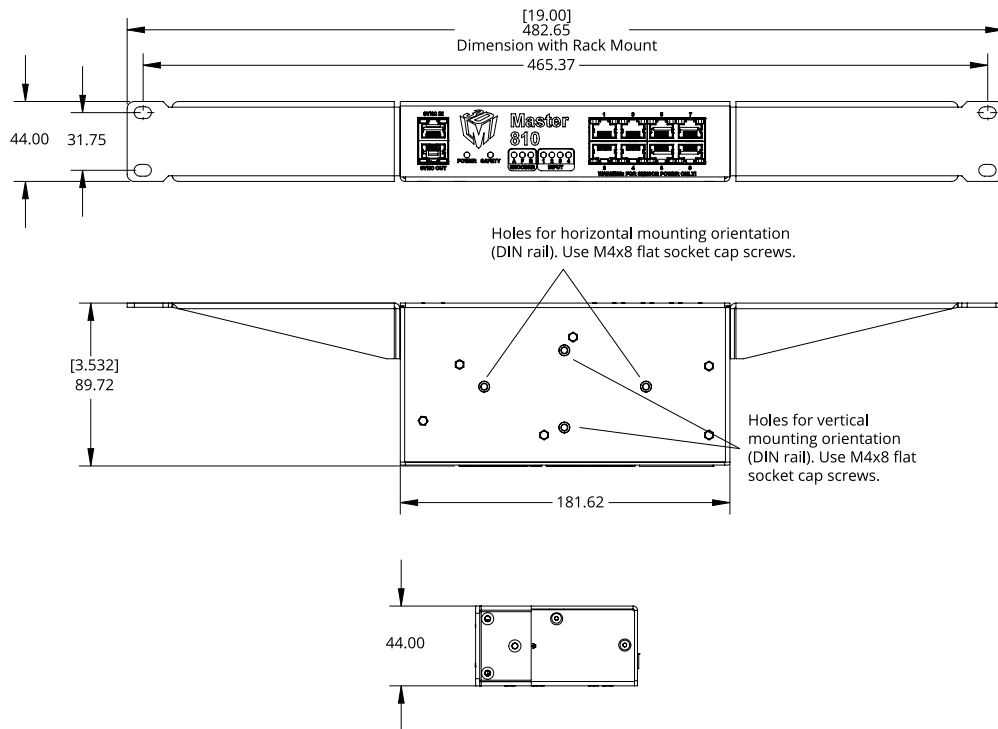


Digital Input Voltage Ranges

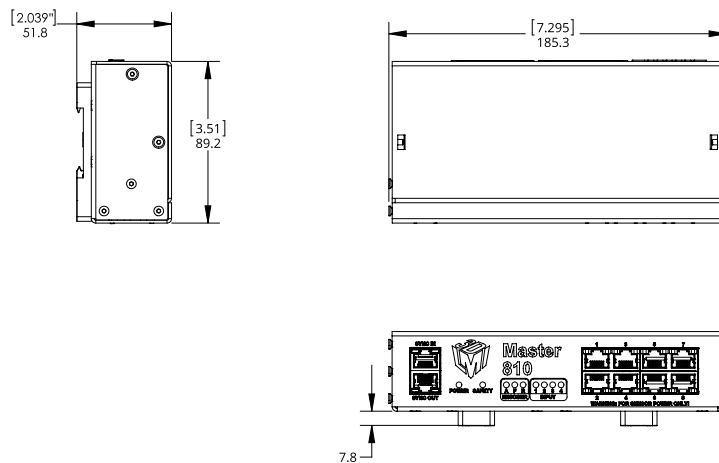
	Input Status	Min (VDC)	Max (VDC)
Single-ended Active High	Off	0	+0.8
	On	+3.3	+24
Single-ended Active Low	Off	(V _{DD} - 0.8)	V _{DD}
	On	0	(V _{DD} - 3.3)
Differential	Off	-24	+0.8
	On	+3.3	+24

Master 810 Dimensions

With 1U rack mount brackets:



With DIN rail mount clips:

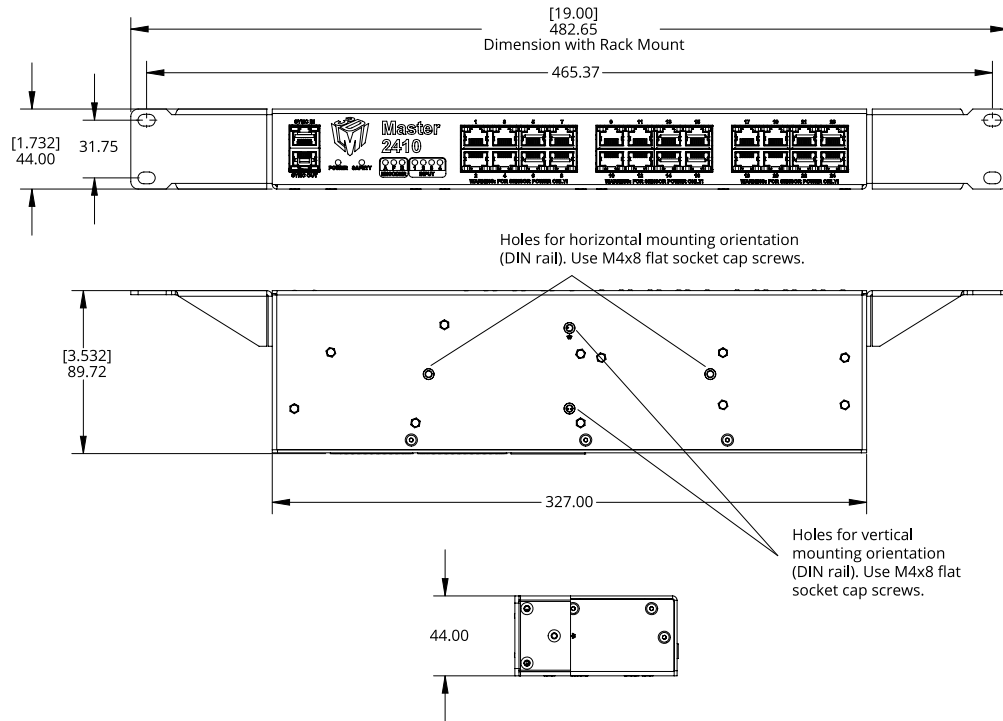


For information on installing DIN rail clips, see *Installing DIN Rail Clips: Master 810 or 2410* on page 35.

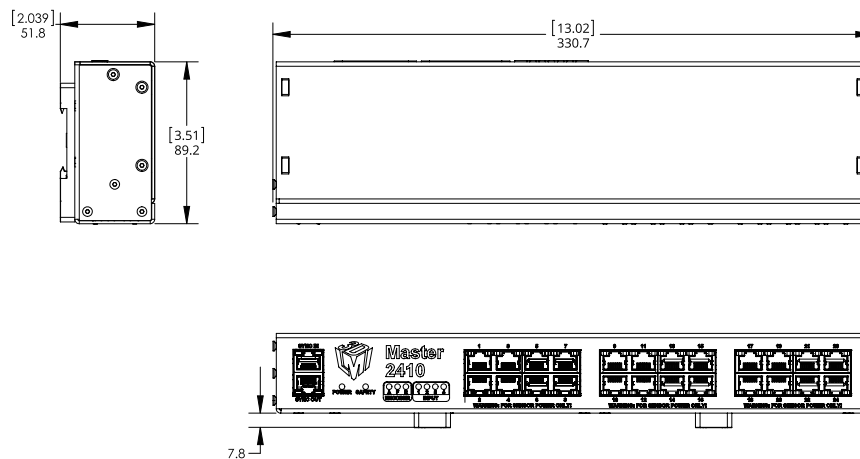
The CAD model of the DIN rail clip is available at <https://www.winford.com/products/cad/dinm12-rc.igs>.

Master 2410 Dimensions

With 1U rack mount brackets:



With DIN rail mount clips:



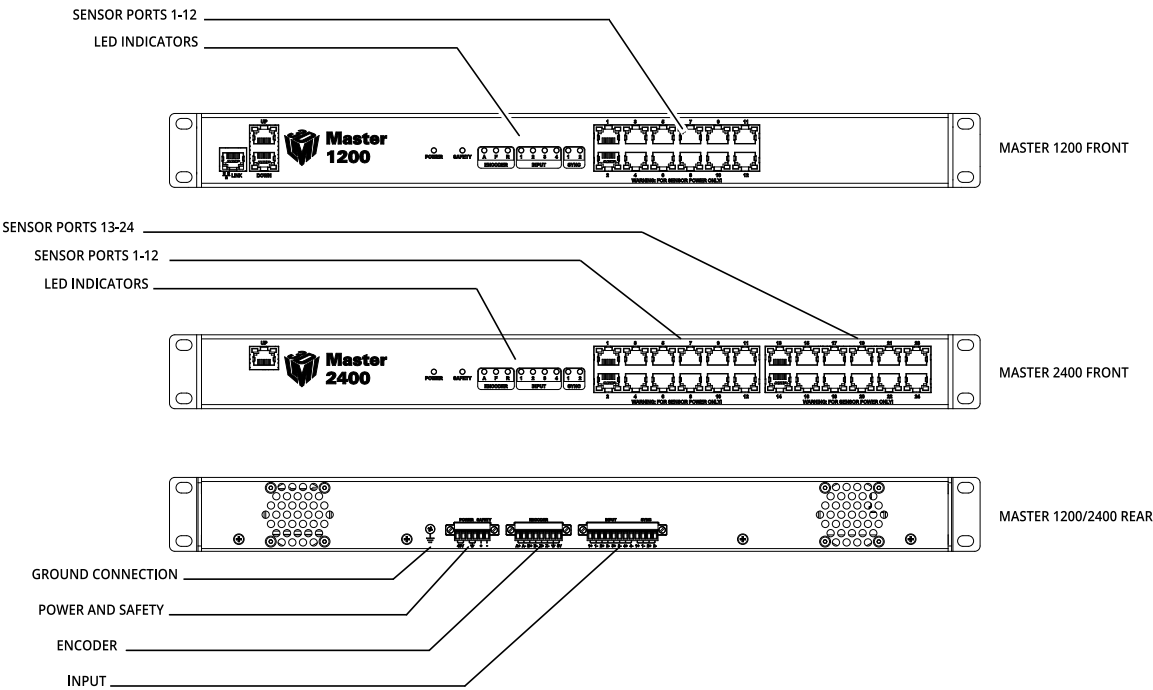
For information on installing DIN rail clips, see *Installing DIN Rail Clips: Master 810 or 2410* on page 35.

The CAD model of the DIN rail clip is available at <https://www.winford.com/products/cad/dinm12-rc.igs>.

Master 1200/2400

Master network controllers provide sensor power and safety interlock, and broadcast system-wide synchronization information (i.e., time, encoder count, encoder index, and digital I/O states) to all devices on a sensor network.

The Phoenix connectors on Master 400/800/1200/2400 are not compatible with the connectors on Master 810/2410. For this reason, if you are switching models in your network, you must rewire the connections to the Master.



Power and Safety (6 pin connector)

Function	Pin
+48VDC	1
+48VDC	2
GND (24-48VDC)	3
GND (24-48VDC)	4
Safety Control+	5
Safety Control-	6

The 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 of 24 VDC to 48 VDC across the pin to enable the laser.

Input (12 pin connector)

Function	Pin
Input 1	1
Input 1 GND	2
Reserved	3
Reserved	4
Reserved	5
Reserved	6
Reserved	7
Reserved	8
Reserved	9
Reserved	10
Reserved	11
Reserved	12



The Input 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

Specification	Value
Power Supply Voltage	+48 VDC
Power Supply Current (Max.)	10 A
Power Draw (Min.)	5.76 W
Safety Input Voltage Range	+24 VDC to +48 VDC
Encoder Signal Voltage	Differential (5 VDC)
Digital Input Voltage Range	Logical LOW: 0 to +0.1 VDC Logical HIGH: +3.5 to +6.5 VDC



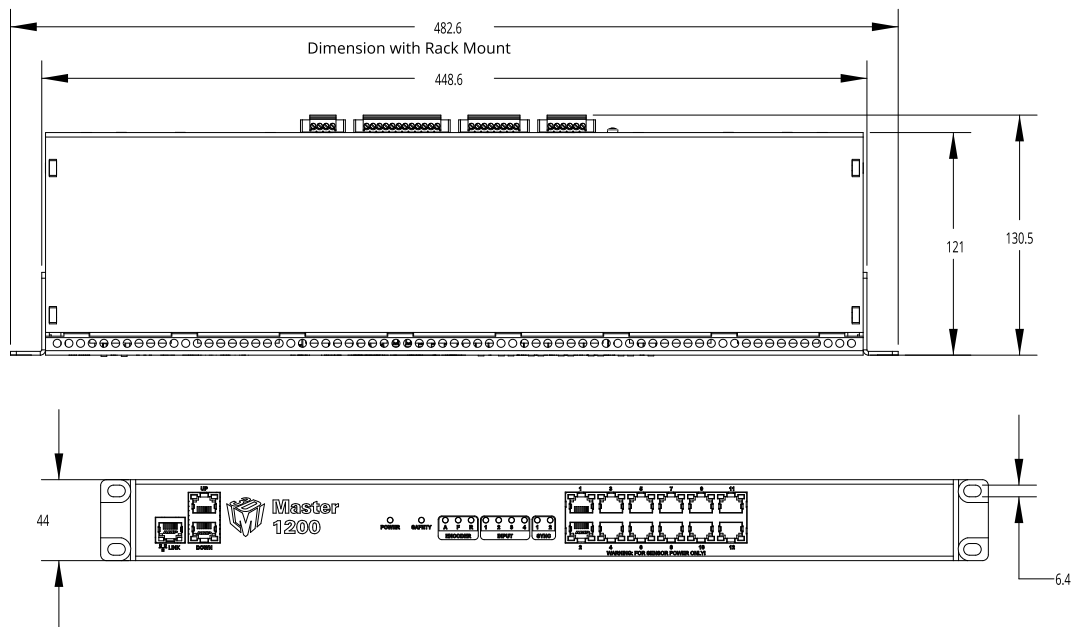
When using a Master hub, the chassis must be well grounded.

 The 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 that 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.



Accessories

Masters

Description	Part Number
Master 100 - for single sensor (development only)	30705
Master 810 - for networking up to 8 sensors	301114
Master 2410 - for networking up to 24 sensors	301115

Cordsets

Description	Part Number
1.2m I/O cordset, open wire end	30864-1.2m
2m I/O cordset, open wire end	30864-2m
5m I/O cordset, open wire end	30864-5m
10m I/O cordset, open wire end	30864-10m
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	30861-5m
10m Power and Ethernet cordset, 1x open wire end, 1x RJ45 end	30861-10m
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
1.2m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-1.2m
2m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-2m
5m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-5m
10m Power and Ethernet to Master cordset, 2x RJ45 ends	30858-10m
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

Cordsets - 90-degree

Description	Part Number
2m I/O cordset, 90-deg, open wire end	30883-2m
5m I/O cordset, 90-deg, open wire end	30883-5m

Description	Part Number
10m I/O cordset, 90-deg, open wire end	30883-10m
15m I/O cordset, 90-deg, open wire end	30883-15m
20m I/O cordset, 90-deg, open wire end	30883-20m
25m I/O cordset, 90-deg, open wire end	30883-25m
2m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-2m
5m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-5m
10m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-10m
15m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-15m
20m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-20m
25m Power and Ethernet cordset, 90-deg, 1x open wire end, 1x RJ45 end	30880-25m
2m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-2m
5m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-5m
10m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-10m
15m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-15m
20m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-20m
25m Power and Ethernet to Master cordset, 90-deg, 2x RJ45 ends	30877-25m

Contact LMI for information on creating cordsets with custom length or connector orientation. The maximum cordset length is 60 m.

Return Policy

Return Policy

Before returning the product for repair (warranty or non-warranty) a Return Material Authorization (RMA) number must be obtained from LMI. Please call LMI to obtain this RMA number.

Carefully package the sensor in its original shipping materials (or equivalent) and ship the sensor prepaid to your designated LMI location. Please ensure that the RMA number is clearly written on the outside of the package. Inside the return shipment, include the address you wish the shipment returned to, the name, email and telephone number of a technical contact (should we need to discuss this repair), and details of the nature of the malfunction. For non-warranty repairs, a purchase order for the repair charges must accompany the returning sensor.

LMI Technologies Inc. is not responsible for damages to a sensor that are the result of improper packaging or damage during transit by the courier.

Software Licenses

Pico-C

Website:

<http://code.google.com/p/picoc/>

License:

picoc is published under the "New BSD License".

<http://www.opensource.org/licenses/bsd-license.php>

Copyright (c) 2009-2011, Zik Saleeba

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Zik Saleeba nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BlowFish

Website:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html>

License:

PuTTY is copyright 1997-2011 Simon Tatham.

Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad Khalifa, Markus Kuhn, Colin Watson, and CORE SDI S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL SIMON TATHAM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CodeMirror

Website:

<http://codemirror.net>

License:

Copyright (C) 2011 by Marijn Haverbeke <marijnh@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 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.

jQuery

Website:

<http://jquery.com/>

License:

Copyright (c) 2011 John Resig, <http://jquery.com/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Closure Library

Website:

<http://code.google.com/closure/library/index.html>

License:

Copyright 2006 The Closure Library Authors. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS-IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

jQuery.CopyEvents

Website:

<http://brandonaaaron.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://brandonaaaron.net>

License:

Copyright (c) 2010 Brandon Aaron

Licensed under the MIT License (<http://www.opensource.org/licenses/mit-license.php>)

jQuery.scaling

Website:

<http://eric.garside.name>

License:

Scaling 1.0 - Scale any page element

Copyright (c) 2009 Eric Garside

Licensed under the MIT License (<http://www.opensource.org/licenses/mit-license.php>)

jQuery.scrollFollow

Website:

<http://kitchen.net-perspective.com/>

License:

Copyright (c) 2008 Net Perspective

Licensed under the MIT License (<http://www.opensource.org/licenses/mit-license.php>)

Flex SDK

Website:

<http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK>

License:

Copyright (c) 2010 Adobe Systems Incorporated

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

EtherNet/IP Communication Stack

Website:

sourceforge.net/projects/opener

License:

SOFTWARE DISTRIBUTION LICENSE FOR THE
ETHERNET/IP(TM) COMMUNICATION STACK (ADAPTED BSD STYLE LICENSE)

Copyright (c) 2009, Rockwell Automation, Inc. ALL RIGHTS RESERVED.

EtherNet/IP is a trademark of ODVA, Inc.

Support

For help with a component or product, please submit an online support ticket using LMI's [Help Desk](http://support.lmi3d.com/newticket.php) at <http://support.lmi3d.com/newticket.php>.

If you are unable to use the Help Desk or prefer to contact LMI by phone or email, use the contact information below.



Response times for phone or email support requests are longer than requests submitted through the Help Desk.

North America

Phone	+1 604 636 1011
Fax	+1 604 516 8368
Email	support@lmi3d.com

Europe

Phone	+31 45 850 7000
Fax	+31 45 574 2500
Email	support@lmi3d.com

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) Burnaby, Canada +1 604 636 1011	LMI Technologies GmbH Berlin, Germany +49 (0)3328 9360 0	LMI (Shanghai) Trading Co., Ltd. Shanghai, China +86 21 5441 0711

LMI Technologies has sales offices and distributors worldwide. All contact information is listed at lmi3d.com/contact/locations.