

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.

In Profile mode, the measurement results are updated after each profile is processed. In Surface mode, the measurement results are updated after each discrete part has been processed. If buffering is enabled in the Ethernet Output panel, reading the Extended Sample State Assembly Object automatically advances the buffer. See *Ethernet Output* on page 326 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	<p>A bit mask where setting the following bits will only perform the action with highest priority*:</p> <ul style="list-style-type: none">1 – Stop sensor2 – Start sensor4 – Perform stationary alignment8 – Perform moving alignment16 – Clear alignment32 – Set runtime variables64 – Load job file <p>*The priority of commands is currently as follows:</p> <ul style="list-style-type: none">1. Stop sensor2. Start sensor3. Perform stationary alignment4. Perform moving alignment5. Clear alignment6. Set runtime variables7. Load job file
1-31	Reserved (except for configuring runtime variables and loading job file)		<p>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.</p> <p>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".</p>

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: 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

Byte	Name	Type	Description
			0 – Fail Bits [1-7]: 0 – Measurement value OK 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 326.

For information on configuring the protocol with the Web interface (when using the protocol over Serial), see *Serial Output* on page 334.

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):