



Title EtherNet/IP communication between Gocator and Omron CS/CJ-Series PLC
Revision 1.1

Purpose This application note demonstrates how to interface to a Gocator from an Omron CS/CJ-Series PLC using the EtherNet/IP protocol. The programming tool used to create the examples is CX Programmer.

Equipment Any Gocator model sensor

Table of Contents

1 Overview	2
2 Configuring the Gocator.....	3
3 I/O Configuration in CX Programmer.....	3
4 Verifying Connection.....	4
5 Sending commands to the Gocator	6
6 Receiving Gocator measurement results	7
6.1 Managing 64-bit values	9
7 Other Communication.....	9
8 Example Ladder Program.....	9





1 Overview

This application note demonstrates how to interface to a Gocator using the EtherNet/IP protocol, which is an industrial protocol that allows bidirectional data transfer with PLCs. It encapsulates the object oriented Common Industrial Protocol (CIP).

The Gocator *only* supports Explicit Messaging via TCP. The user must create ladder logic to poll the sensor for measurement results. It is not possible to use Omron's Network Configurator for tag data links, as this type of communication is for devices using Implicit Messaging.

EtherNet/IP communication enables the PLC to control a Gocator sensor to:

- Command the sensor to Start, Stop and run a Calibration.
- Switch active configuration.
- Receive sensor state.
- Receive measurement values and decisions.

The Omron CS/CJ-Series PLC offers several instructions for Explicit EtherNet/IP communication. The general instruction is CMND(490) with FINS code 2810 Hex, which requires a full set of message parameters, but has no limitations in its use. The simplified instructions to get and set single attribute, EGATR(721) and ESATR(722), are easier to use, but only support 8-bit parameters for Class ID, Instance ID and Attribute ID. This means that EGATR and ESATR cannot be used with the Gocator's Command, Sensor State and Sample State Objects, since they all have Instance IDs above 300 Hex. The following table is a summary of compatibility between Gocator EIP objects and CJ/CS instructions for explicit messaging.

EIP Object/Instruction	CMND	EGATR
Identity	OK	OK
TCP/IP	OK	OK
Ethernet Link	OK	OK
Command	OK	-
Sensor State	OK	-
Sample State	OK	-

Table 1: Gocator EIP objects compatibility with CMND & EGATR instructions.

Before reading this guide it is strongly recommended to read the EtherNet/IP section of the Gocator User's Manual.





2 Configuring the Gocator

The Gocator supports many different output protocols. Before attempting to communicate to the sensor from a PLC, the sensor has to be configured to run the EtherNet/IP protocol. The Gocator sensor is configured through a web browser running on a computer. Figure 1 shows where this configuration takes place in the Output Panel. Refer to the Gocator Quick Start Guide for setup instructions, which can be downloaded here:

<http://www.lmi3d.com/product/gocator-2000-family/support/files/all>

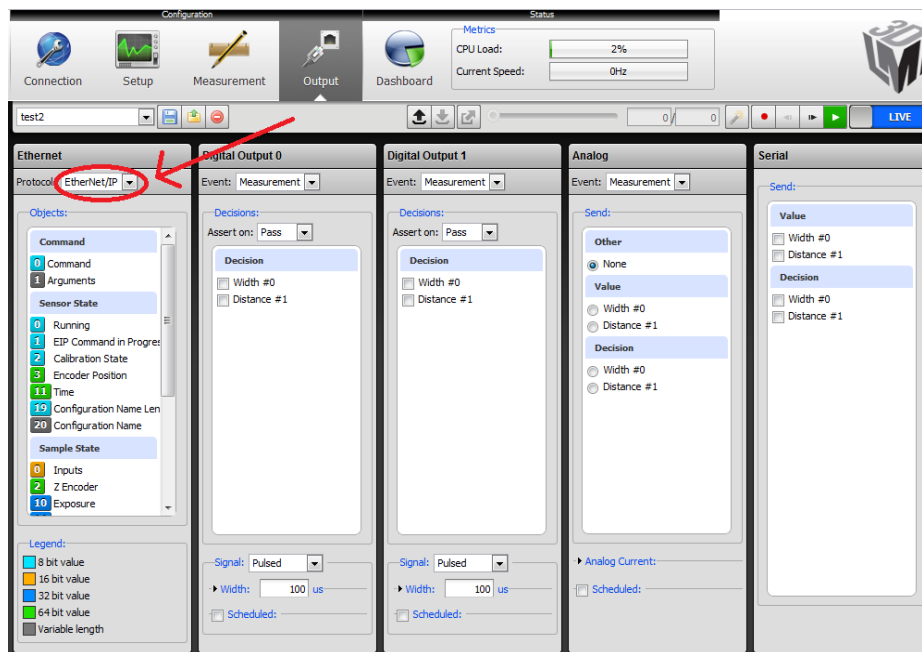


Figure 1: Configuring the EtherNet/IP protocol in the Output Panel.

3 I/O Configuration in CX Programmer

The PLC has to be equipped with an Ethernet card configured to be on the same subnet as the Gocator sensor (factory default Gocator IP address is 192.168.1.10). The Gocator supports unconnected or connected TCP messaging. Implicit UDP I/O messaging is **not** supported.

The default EtherNet/IP ports are used.

The IP address of the PLC is set through the IO Table in CX Programmer, in the Edit Parameters dialog of the EIP module. Verify that the IP Address is on the correct subnet. If the Omron PLC has a NODE No. setting on the unit, the last octet in the IP address must match the NODE setting, as shown in Figure 2.



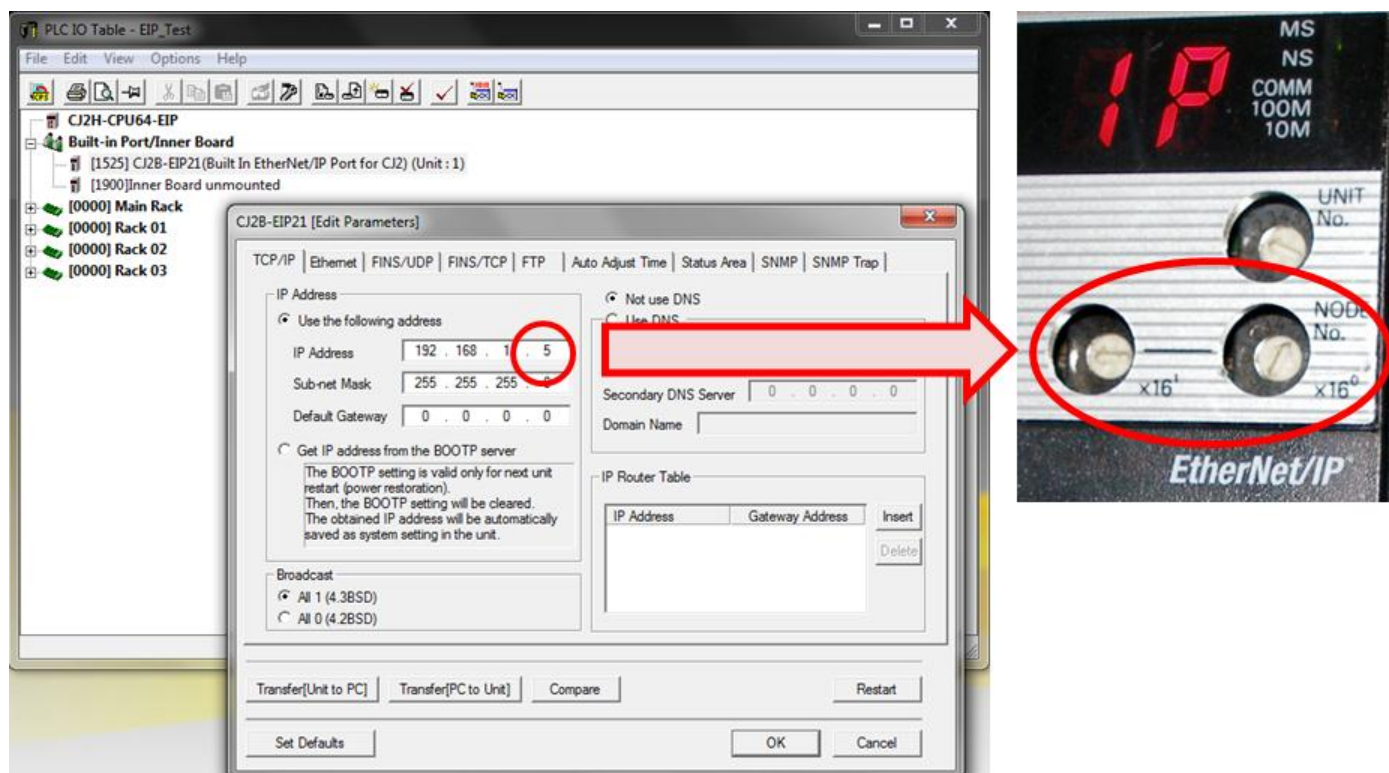


Figure 2: I/O Configuration in CX Programmer.

4 Verifying Connection

Before attempting to control and run the Gocator from the PLC, the user should always verify the connection first by reading an attribute from the Identity Object, for example the sensor's serial number.

Reading the sensor's serial number can be done with the EGATR instructions. Figure 3 shows the simplest form of ladder logic with an EGATR instruction to read the serial number from the sensor. The message send parameters are stored in memory starting at address D0 and the control data parameters start at address D10. The return message from the Gocator is received starting at address D20.

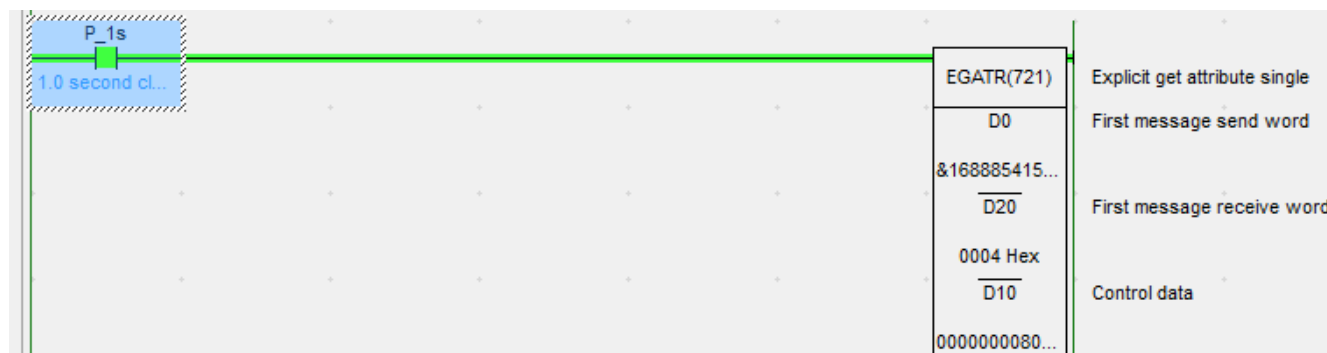


Figure 3: Ladder logic to read the sensor's serial number.





Figure 4 shows how to configure the message and control data parameters, where the values are displayed in Hex. The first word at D0 is the node address of the device the message is to be sent to. The node address is the last octet of the IP address, in this example A Hex (10 decimal) since the Gocator IP address is 192.168.1.10. The second word at D1 is the Class ID of the Identity object (1), the third word is the Instance ID (1) and the fourth word is the Attribute ID for the sensor's serial number (6). The Class, Instance and Attribute IDs are defined in the Gocator User's Manual.

The first word of the control data at D10 is the maximum number of words in the receive message, which is 3 when polling for the serial number. The second word at D11 is set to 0x8011 - 8 sets the byte order to reversed, 0 is logical port 0, 11 is 10 hex + FINS unit number of the EIP module (which is 1 in the PLC used in this example).

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D00000	000A	0001	0001	0006	0000	0000	0000	0000	0000	0000
D00010	0003	8011	0000	0000	0000	0000	0000	0000	0000	0000
D00020	0004	1955	0000	0000	0000	0000	0000	0000	0000	0000

Figure 4: EGATR parameters to read the serial number.

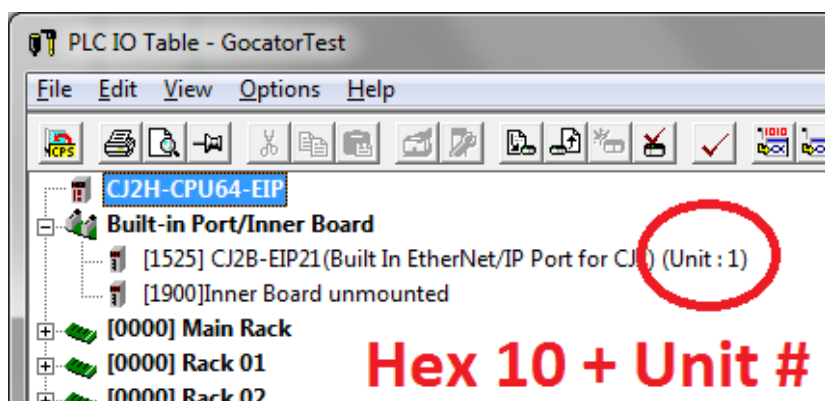


Figure 5: Formula for Control Data parameter for EIP module.

When this simple ladder logic is downloaded to the PLC and then executed, the serial number should be received at memory address D21, in this example 1955 Hex is 6485 Decimal. Successful reception of the serial number confirms that the communication path between the Gocator and the Omron PLC is functional.





5 Sending commands to the Gocator

The Command Assembly Object is used to start, stop, calibrate and switch configuration on the sensor. Since the Instance ID is a 310 Hex for the Command Object, a 16-bit Instance value is required, which is only supported by the CMND instruction. *Note that the ladder logic should be designed so that the command is not sent continuously with every scan of the logic.*

The command is written to the sensor as a 32 element byte array. The first byte is the command itself, whereas bytes 1-31 hold the name of the configuration file, which only applies if the command is Load Configuration (5). Figure 6 shows simple ladder logic to send a command on the first cycle of the program.

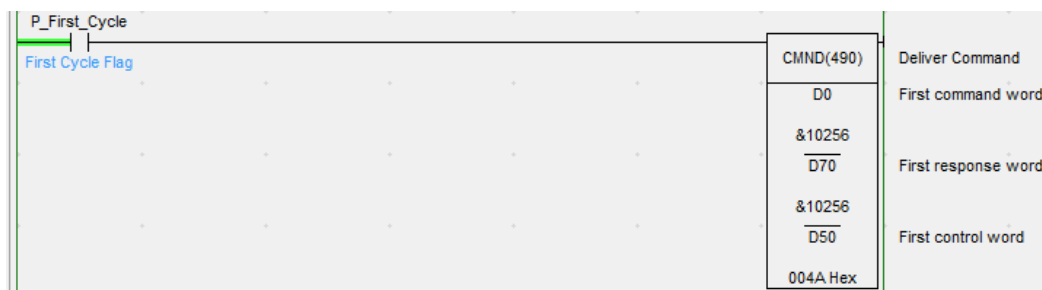


Figure 6: Ladder logic to send a command to the sensor.

The CMND(490) instruction for Explicit Messaging is configured as a FINS command with code 2810 Hex, which represents a CIP UCMM MESSAGE SEND command. For a detailed description of this command, please refer to the Omron Operation Manual for SYSMAC CS/CJ EtherNet/IP Units (W465-E1-06.pdf).

Figure 7 shows an example how to configure the message and control data parameters, where the values are displayed in Hex. The send message parameters are from D0 to D36. The control data parameters are from D50 to D55. First control data parameter at D50 is number of command bytes 4A Hex = 74 bytes from D0 to D36. Second control data parameter at D51 is number of response bytes. Fourth control data parameter at D53 is 11 Hex, which again is 10 hex + FINS unit number of the EIP module (which is 1 in the PLC used in this example).

D										
Start Address:	0	On	Off	SetValue						
ChangeOrder		ForceOn	ForceOff	ForceCanc						
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D00000	2810	0000	05DC	5202	2006	2401	0A0C	2A00	1004	2004
D00010	2500	1003	3003	0100	0000	0000	0000	0000	0000	0000
D00020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0700
D00030	120C	3139	322E	3136	382E	312E	3130	0000	0000	0000
D00040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
D00050	004A	000C	0000	0011	0000	0000	0000	0000	0000	0000
D00060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
D00070	2810	0000	0000	0004	9000	0000	0000	0000	0000	0000

Figure 7: CMND parameters to send a start command.





The following table provides a detailed description of the send message parameters.

Word	15 08	07 00	Comment
S+0	28	10	FINS command code
S+1	00	00	Transport ID. If multiple messages are sent, this ID can be used to identify the response
S+2	05	DC	Message monitoring time
S+3	52	02	Routing service data for the Explicit Message (Service code and request path size)
S+4	20	06	Routing service data for the Explicit Message (Class ID)
S+5	24	01	Routing service data for the Explicit Message (Instance ID)
S+6	0A	0C	Routing service data for the Explicit Message (Priority/Time)
S+7	2A	00	Message size (bytes swapped). 2A Hex = 42 bytes of service code + request path + 32 byte command
S+8	10	04	Service code=10 Hex (Set Attribute Single), Request path size = 4 words (S+9 to S+12)
S+9	20	04	8-bit Class ID=20 Hex, Class ID=04 Hex (from Gocator User's Manual)
S+10	25	00	16-bit Instance ID=25 Hex
S+11	10	03	Instance ID (bytes swapped) = 0310 Hex (from Gocator User's Manual)
S+12	30	03	8-bit Attribute ID=30 Hex, Attribute ID=03 Hex (from Gocator User's Manual)
S+13	01	00	First byte in command array represents the command itself, 01 Hex=Start Sensor
S	00	00	S+14 to S+28 contains the configuration file name in ASCII Hex (for Load Configuration)
S+29	07	00	Route path size (bytes swapped). 07 Hex = 7 words from S+30 to S+36
S+30	12	0C	Ethernet Port=12 Hex (Extended Link Address=1 Hex, Port=2 Hex), Route path size=0C Hex=12 bytes (12 characters in IP address "192.168.1.10")
S+31	31	39	ASCII Hex code for "1"=31, "9"=39
S+32	32	2E	ASCII Hex code for "2"=32, "."=2E
S+33	31	36	"1", "6"
S+34	38	2E	"8", "."
S+35	31	2E	"1", "."
S+36	31	30	"1", "0"

Table 2: Send message parameters to send Start command to sensor at 192.168.1.10

6 Receiving Gocator measurement results

The Sample State Assembly Object is used to receive measurement results from the Gocator. Since the Instance ID is a 321 Hex for the Sample State Object, a 16-bit Instance value is required, which is only supported by the CMND instruction.

Figure 8 shows simple ladder logic to poll for measurement results from the Gocator once per second. The message returned from the sensor is a 180 byte array, as defined in the Gocator User's Manual. The first 80 bytes contain various stamp information such as input states, time stamp, encoder count and frame count. The last 100 bytes contain the measurement results (values and decisions). The position of the result is based on the ID of the measurement tool, e.g. a tool with ID 0 has its result located starting at byte 80 and a tool with ID 5 has its result located starting at byte 105.



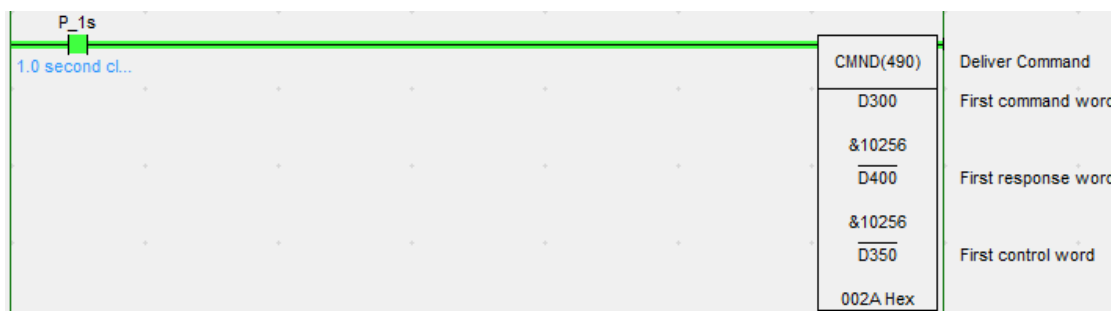


Figure 8: Ladder logic to poll for measurement results once per second.

Figure 9 shows an example how to configure the message and control data parameters, where the values are displayed in Hex. The send message parameters are from D300 to D320. The control data parameters are from D350 to D355. First control data parameter at D350 is number of command bytes 2A Hex = 42 bytes from D300 to D320. Second control data parameter at D351 is number of response bytes C0 Hex = 192 bytes (12 byte CIP response header, plus 180 bytes from the Gocator). Fourth control data parameter at D353 is 11 Hex, which again is 10 hex + FINS unit number of the EIP module (which is 1 in the PLC used in this example).

Also shown is an example measurement result. The 32-bit value is returned to the PLC with most significant byte first. Note that the Gocator transmits results scaled by a factor of 1000, so 71.630 mm is sent as 71630 = 117CE Hex.

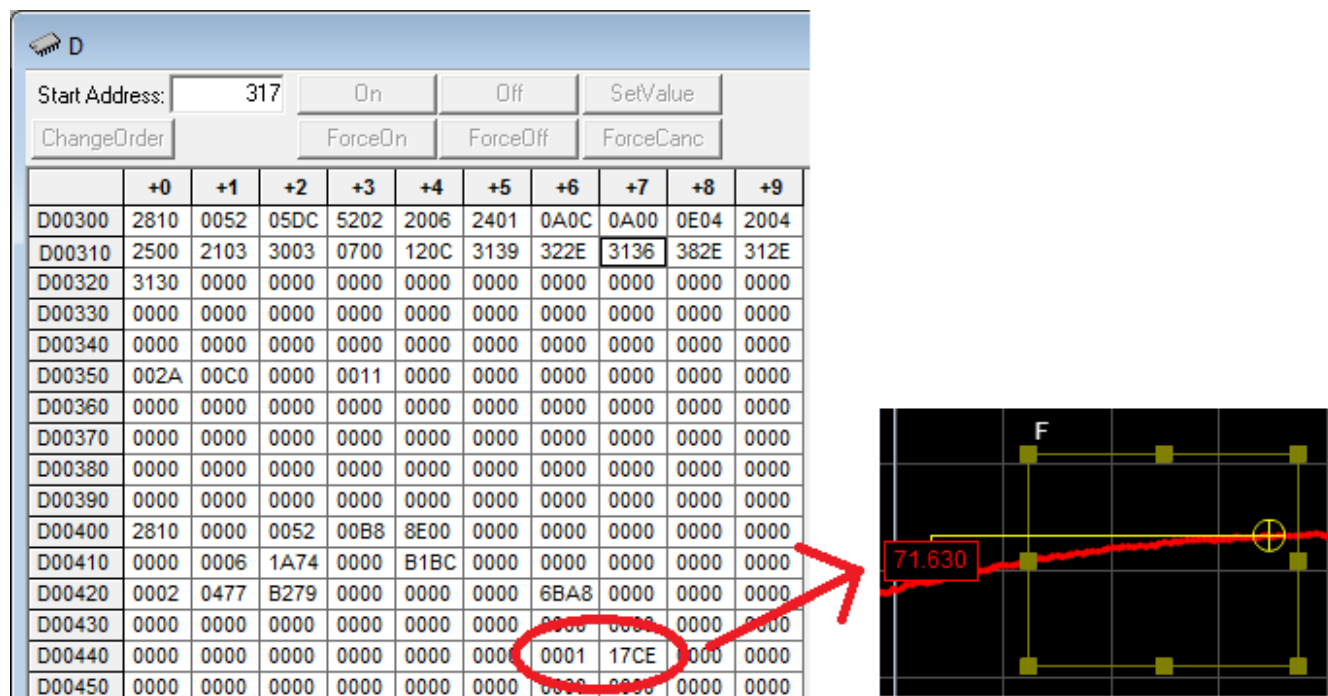


Figure 9: CMND parameters to poll for measurement results. Value returned 117CE Hex = 71630.





The following table provides a detailed description of the send message parameters.

Word	15 08	07 00	Comment
S+0	28	10	FINS command code
S+1	00	52	Transport ID. If multiple messages are sent, this ID can be used to identify the response
S+2	05	DC	Message monitoring time
S+3	52	02	Routing service data for the Explicit Message (Service code and request path size)
S+4	20	06	Routing service data for the Explicit Message (Class ID)
S+5	24	01	Routing service data for the Explicit Message (Instance ID)
S+6	0A	0C	Routing service data for the Explicit Message (Priority/Time)
S+7	0A	00	Message size (bytes swapped). 0A Hex = 10 bytes of service code + request path
S+8	0E	04	Service code=0E Hex (Get_Attribute_Single), Request path size=4 words (S+9 to S+12)
S+9	20	04	8-bit Class ID=20 Hex, Class ID=04 Hex (from Gocator User's Manual)
S+10	25	00	16-bit Instance ID=25 Hex
S+11	21	03	Instance ID (bytes swapped) = 0321 Hex (from Gocator User's Manual)
S+12	30	03	8-bit Attribute ID=30 Hex, Attribute ID=03 Hex (from Gocator User's Manual)
S+13	07	00	Route path size (bytes swapped). 07 Hex = 7 words from S+30 to S+36
S+14	12	0C	Ethernet Port=12 Hex (Extended Link Address=1 Hex, Port=2 Hex), Route path size=0C Hex=12 bytes (12 characters in IP address "192.168.1.10")
S+15	31	39	ASCII Hex code for "1" =31, "9"=39
S+16	32	2E	ASCII Hex code for "2"=32, "."=2E
S+17	31	36	"1", "6"
S+18	38	2E	"8", "."
S+19	31	2E	"1", "."
S+20	31	30	"1", "0"

Table 3: Send message parameters to poll for measurements from sensor at 192.168.1.10

6.1 Managing 64-bit values

The Gocator is using 64-bit values to represent encoder positions, time stamps and frame counts. Most PLC hardware does not support 64-bit data types, so the user can choose to either ignore the most significant bits, or manually handle the values as two 32-bit values.

7 Other Communication

The Gocator supports several more EtherNet/IP object classes in order to read network configuration attributes (such as sensor IP address) and sensor state. This communication is performed in the exact same way as outlined in this document.

8 Example Ladder Program

A very simple example ladder program comes with this Application Guide. The PLC model in the project file is a CJ2H CPU64-EIP. It's a program with two rungs, where a Start Command is sent to the sensor on the first cycle and then it's polling for measurement results once per second. Memory section D contains all the parameter settings for the CMND(490) instructions.

