



A Sierra Monitor Company

FieldServer Driver - Fieldbus FS-8704-12 GE-EGD (Ethernet Global Data)

Driver Code:	GE-EGD
Version:	1.00a
Protocol Documentation:	N/A
Protocol Version:	11/30/2000
Physical Interface:	
Adapter:	N1,N2

Driver Description

The GE-EGD (Ethernet Global Data) driver allows the FieldServer to transfer data to and from devices over Ethernet using GE-EGD (Ethernet Global Data) protocol. There are two Ethernet ports standard on the FieldServer. The FieldServer can emulate either a Server or Client.

GE Fanuc Automation and GE Drive Systems developed an Ethernet Global Data, or EGD, exchange for PLC and computer data in 1998. EGD uses UDP or datagram messages for fast transfer of up to 1400 bytes of data from a producer to one or more consumers. UDP messages have much less overhead than the streaming TCP connection used for programming or CommReq's over SRTP Ethernet. Like Genius® broadcast input or directed control messages, UDP messages are not acknowledged. They can be sent at short intervals. Chances of one or more messages being dropped are small on a local area network.

As a client the FieldServer acts as a EGD consumer. As a master the FieldServer acts as a EGD producer.



FieldServer Specific Map Descriptor Parameters (Passive Client)

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters. The map descriptor name can be any name that has meaning to you and in fact duplicate name will not produce an error. This driver recognizes a special map descriptor name; "EGD-ii". It stands for EGD Internal Indications. Its use is more fully explained in section 6 of this manual.
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Location	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	Passive



Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in "Client Node Descriptor" above
Length	Number of points being consumed for Bit values this represents the number of bytes (i.e. number of points divided by 8)	1 - 1000
The following keywords apply only to the GE-EGD protocol.		
ge_producerId	<p>This identifies the GE device producing the EGD data. Although it is in decimal dot format like an IP address it is not an IP address and does not correspond to the IP address of the GE-Ethernet port producing the message. It corresponds to the producer ID configured for the CPU producing the data.</p> <p>When processing an incoming message the IP address of the source is not considered by this driver. Rather, the ge_producerID is considered in deciding where to store the incoming data.</p> <p>The default value is typically the same as the IP address of the producer but the value can be changed and it is possible for one device to have multiple Ethernet interfaces and hence multiple IP addresses.</p>	<p>Nnn.nnn.nnn.nnn</p> <p>Where nnn are in the range 0-255.</p>
ge_exchangeId	This and the producerID uniquely identify a packet of EGD data. Thus, this driver uses these two parameters to match a produced data packet with one or more passive map descriptors.	Integer values ≥ 1
ge_data_type	Each produced data packet contains raw packed data. Nothing in the message identifies the structure or type of the incoming data. The Driver, thus cannot differentiate between byte, integer, real ... numbers and requires the specification of this keyword to unpack the data buffer.	Byte, Bit, Word, Dword, Int , Long Float (4 byte IEEE real number) or Double (8 byte IEEE real number).
ge_offset	If the producer has been configured to produce data of multiple type in one data packet then you will need multiple map descriptors to decode them. The ge_offset is used to point to the first byte in the data packet to be processed by the map descriptor. The default value is zero. Typically the map descriptor for the 2nd, 3rd ... map descriptors associated with one data packet will be non-zero.	Zero , Any positive integer



FieldServer Specific Map Descriptor Parameters (Active Server)

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	<p>Up to 32 alphanumeric characters.</p> <p>The map descriptor name can be any name that has meaning to you and in fact duplicate name will not produce an error.</p> <p>This driver recognizes a special map descriptor name; "EGD-ii". It stands for EGD Internal Indications. Its use is more fully explained in section 6 of this manual.</p>
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Location	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	WRBC



Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in "Producer Node Descriptor" above
Length	Length of Map Descriptor	1 - 1000
The following keywords apply only to the GE-EGD protocol.	Only one map descriptor may be configured for each exchangeID. This is different from the configuration of consumer map descriptors. The effect of this limitation is to limit each produced exchange to contain only one data type and to contain data from only one data array.	
Ge_producerId	This identifies the GE device producing the EGD data. Although it is in decimal dot format like an IP address it is not an IP address and does not necessarily correspond to the IP address of the GE-Ethernet port producing the message. It corresponds to the producer ID configured for the CPU producing the data. The default value is typically the same as the IP address of the producer but the value can be changed and it is possible for one device to have multiple Ethernet interfaces and hence multiple IP addresses. Any change to the producerID must be matched by a similar change in the consumer's configuration.	Nnn.nnn.nnn.nnn Where nnn are in the range 0-255.
Ge_exchangeId	This and the producerID uniquely identify a packet of EGD data. Thus, the consumer uses these two parameters to update. Any change to the exchangeID must be matched by a similar change in the consumer's configuration.	Integer values ≥ 1
Ge_data_type	Each produced data packet contains raw packed data. This keyword is used to tell the driver how to pack the data into the message. Thus you can read from a BIT array in the FieldServer but send the data as words for storage in %R (register memory) in the GE-PLC. Any change to the data type must be matched by a similar change in the consumer's configuration.	Byte, Bit, Word, Dword, Int , Long Float (4 byte IEEE real number) or Double (8 byte IEEE real number). See section 6.3 for a full list.
Ge_offset	Not required for producer map descriptors.	



When the driver sees this map descriptor it uses the data array EGD_DIAG to store driver specific data. Only one of these map descriptors may be specified per FieldServer.

The driver stores the following data.

Array Element	Contents	
0-31	The first 32 bytes of the most recently received UDP packet received on port 0x4746 (The GE EGD port).	
32	PDUTypeVersion	
33	RequestID	
34	ProducerID	As a UINT32. Not in dot format.
35	ExchangeID	
36	TimeStampSec	
37	TimeStampNanoSec	
38	Status	Read section 4 of Chapter 4 of GE-Fanuc document GFK-1541 for more information.
39	ConfigSignature	
40	Reserved	
41	Source IP Address	As a UINT32. . Not in dot format.

The following data types are recognized by the driver

Byte
Bit (translated as 8bits aligned with a byte boundary)
Word (unsigned 16bit integer)
Dword (unsigned 32bit integer)
Int (signed 16bit integer)
Long (signed 32bit integer)
Float (translated as an IEEE 4 byte real number)
Double (translated as an IEEE 8 byte real number)



As well as the following GE specific types.

Type	Description	P-Producer	C-Consumer
%R	Register memory in word mode	P/C	
%AI	Analog input memory in word mode	P/C	
%AQ	Analog output memory in word mode	P/C	
%I	Discrete input memory in byte mode	P/C	
%Q	Discrete output memory in byte mode	P/C	
%T	Discrete temporary memory in byte mode	P/C	
%M	Discrete momentary memory in byte mode	P/C	
%SA	Discrete system memory group A in byte mode	P/C	
%SB	Discrete system memory group B in byte mode	P/C	
%SC	Discrete system memory group C in byte mode	P/C	
%S	Discrete system memory in byte mode	P	
%G	Discrete global data table in byte mode	P/C	