



A Sierra Monitor Company

EUROPE: Paul Lambert, Regional Manager, Berlin, Germany

Tel: +49 (0) 30 510 65233 **Cell:** +49 (0) 174 1914702

email: plambert@fieldserver.com

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-146 KNX

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 2013

Driver Version:	1.05
Document Revision:	0

TABLE OF CONTENTS

1	KNX Description	3
2	Driver Scope of Supply	4
2.1	Supplied by FieldServer Technologies for this driver	4
2.2	Provided by the Supplier of 3 rd Party Equipment	4
2.2.1	<i>Required 3rd Party Hardware</i>	<i>4</i>
3	Hardware Connections.....	4
4	Data Array Parameters.....	5
5	Configuring the FieldServer as a KNX Client	6
5.1	Client Side Connection Parameters	6
5.2	Client Side Node Parameters	7
5.3	Client Side Map Descriptor Parameters.....	7
5.3.1	<i>FieldServer Specific Map Descriptor Parameters</i>	<i>7</i>
5.3.2	<i>Driver Related Map Descriptor Parameters</i>	<i>7</i>
5.4	Map Descriptor Examples	8
Appendix A. Reference		10
Appendix A.1. KNX Data Types		10
Appendix B. Troubleshooting.....		10
Appendix B.1. Driver Messages		10

1 KNX DESCRIPTION

The KNX driver allows the FieldServer to transfer data to and from devices using KNX protocol. The Fieldbus connection is included with the FieldServer. The FieldServer can emulate a Passive Client.

KNX driver enables data access from KNX networks to other FieldServer protocols. Most KNX data point types are supported, allowing communication to almost any kind of KNX device in the installation, such as temperature sensors, shutters, light switches, actuators, alarms etc. This allows BMS systems to access a KNX network using direct read and write of KNX configured groups. This setup does not require the use of ETS4 to configure the QuickServer KNX gateway.

The KNX protocol is a connectionless protocol and therefore supports multiple clients and multiple servers.

The QuickServer is intended to act as a Passive Client on the KNX bus and make information available to other protocols.

Supported Data point Types

Data point Name	Description
DPT1	1-bit Binary Switch
DPT2	2-bit Step Control
DPT3	4-bit Dimming
DPT4	8-bit Set
DPT5	8-bit Unsigned Value
DPT6	8-bit Signed Value
DPT7	16-bit Unsigned Value
DPT8	16-bit Signed Value
DPT9	16-bit Floating point value
DPT12	32-bit Unsigned Value
DPT13	32-bit Signed Value
DPT14	32-bit Float
DPT15	32-bit Access
DPT17	8-bit Scene Number
DPT18	8-bit Scene Control
DPT20	8-bit Enum Value

Maximum Nodes Supported

FieldServer Mode	Nodes	Comments
Client	255	Maximum number in a KNX bus zoned

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8915-10	UTP cable (7 foot) for Ethernet connection
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-02	RJ45 to DB9F connector adapter
FS-8917-01	RJ45 to DB25M connection adapter

2.2 Provided by the Supplier of 3rd Party Equipment

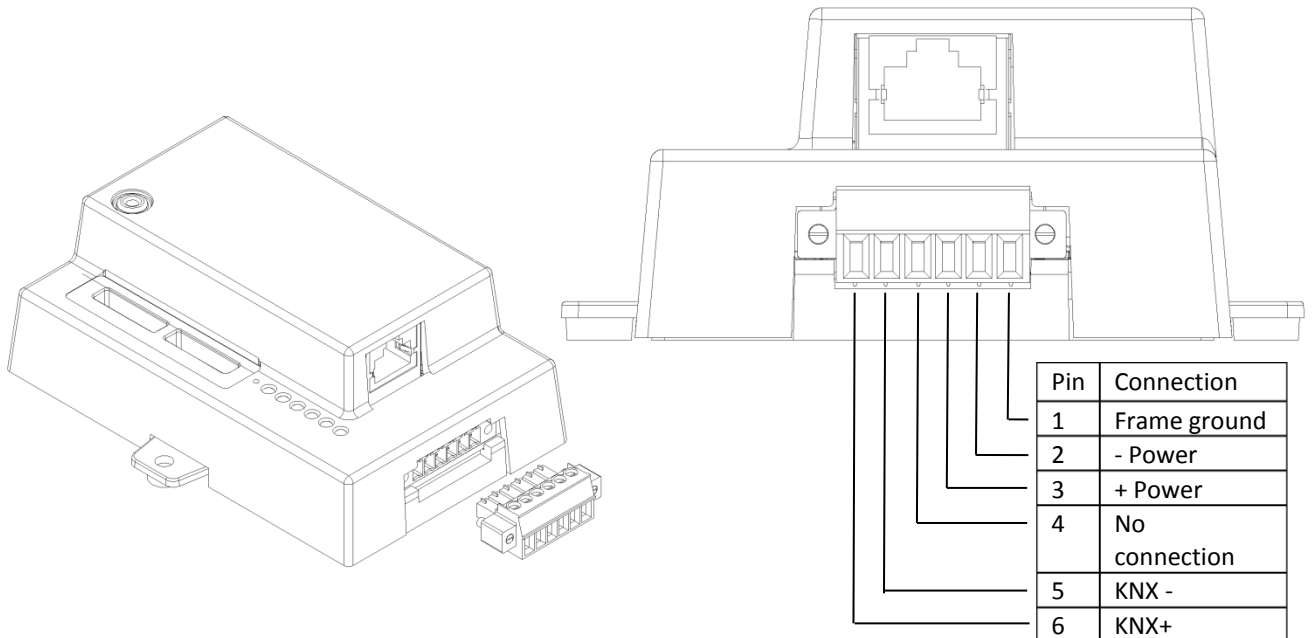
2.2.1 Required 3rd Party Hardware

Part #	Description
	KNX Bus power supply

3 HARDWARE CONNECTIONS

The FieldServer is connected to the KNX bus as shown in connection drawing.

Configure the KNX bus according to manufacturer’s instructions (i.e. ETS4).



4 DATA ARRAY PARAMETERS

Data Arrays are “protocol neutral” data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	FLOAT, UINT16, SINT16, UINT32, SINT32, BYTE.
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10,000

Datapoint Name	Description	Recommended Data Array Value
DPT1	1-bit Binary Switch	BYTE
DPT2	2-bit Step Control	BYTE
DPT3	4-bit Dimming	BYTE
DPT4	8-bit Set	BYTE
DPT5	8-bit Unsigned Value	BYTE
DPT6	8-bit Signed Value	SINT16
DPT7	16-bit Unsigned Value	UINT16
DPT8	16-bit Signed Value	SINT16
DPT9	16-bit Floating point value	FLOAT
DPT12	32-bit Unsigned Value	UINT32
DPT13	32-bit Signed Value	SINT32
DPT14	32-bit Float	FLOAT
DPT15	32-bit Access	UINT32
DPT17	8-bit Scene Number	BYTE
DPT18	8-bit Scene Control	BYTE
DPT20	8-bit Enum Value	BYTE

Example

```
// Data Arrays
Data_Arrays
Data_Array_Name , Data_Array_Format , Data_Array_Length
DA_KNX_01      , UINT16           , 50
DA_KNX_02      , UINT32           , 50
DA_KNX_03      , BYTE             , 50
DA_KNX_04      , FLOAT            , 50
```

5 CONFIGURING THE FIELDSEVER AS A KNX CLIENT

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FieldServer). This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a KNX device (Server).

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for KNX bus communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, the destination device addresses need to be declared in the “Client Side Nodes” section, and the data required from the servers needs to be mapped in the “Client Side Map Descriptors” section. Details on how to do this can be found below.

5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	R2
Protocol	Specify protocol used	KNX
Physical_Address	4 character Hex value (max FFFF), e.g. ffff = 15,15,255	0-FFFF
	2 part scheme, e.g. 1.15 = 0,1,15	[0-255].[0-255]
	3 part scheme, e.g. 1.2.31	[0-15].[0-15].[0-255]
Retries	Specify the number of retries (KNX bus handles this automatically)	0
Recovery_Interval	Specify the period between retries (recommend 60s)	0-32000s, 60s

Example1

The QuickServer is configured as a KNX Client with the KNX physical address of 1.1.128 using the hexadecimal address scheme:

```
// Client Side Connections

Connections
Port      , Protocol  , Physical_Address  , Retries  , Recovery_Interval
R2       , KNX       , 1180             , 0        , 60s
```

Example2

The QuickServer is configured as a KNX Client with the KNX physical address of 1.1.128 using the 2 part decimal address scheme:

```
// Client Side Connections

Connections
Port      , Protocol  , Physical_Address  , Retries  , Recovery_Interval
R2       , KNX       , 129.128          , 0        , 60s
```

Example3

The QuickServer is configured as a KNX Client with the KNX physical address of 1.1.128 using the 2 part decimal address scheme:

```
// Client Side Connections

Connections
Port      , Protocol  , Physical_Address  , Retries  , Recovery_Interval
R2       , KNX       , 1.1.128          , 0        , 60s
```

5.2 Client Side Node Parameters

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Protocol	Specify Protocol used	KNX
Port	Specify through which port the device is connected to the FieldServer	R2

Example

```
// Client Side Nodes

Nodes
Node_Name      , Protocol  , Port
KNX_Bridge_1  , KNX      , R2
```

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from Section 0.
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 0.
Function	Function of Client Map Descriptor	Rdbc, Wrbc, Wrbx, ARS

5.3.2 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the Node names specified in Section 5.2
Data_Type	Data type	As per Appendix A
Group_Address	KNX Group Address Value associated with the data array (Hexidecimal value max FFFF)	[0-15]/[0-15]/[0-255]
Linked_Map_Descriptor	If a Group address should be linked to another Group address' data array	One of the Map descriptor names specified in Section 5.23, otherwise leave blank with a "-"
Read_Response	QuickServer is not intended to respond to Read requests ("No" recommended), but this could be set ("Yes) to do so.	No, Yes

5.4 Map Descriptor Examples.

Example1

The QuickServer is configured to store writes or other read responses. It will also generate a write should the data array update. KNX_VAR1_LINK data array is updated with a group 0.5.2 value update as it is linked to KNX_VAR1 data array.

```
// Client Side Map Descriptors

// Linked map descriptor example, any store to a linked map descriptor will update the parent.
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Group_Address , Linked_Map_Descriptor
KNX_VAR1 , DA_KNX1 , 00 , ARS , KNX_Read_1 , DPT1 , 0/5/01 , -
KNX_VAR1_LINK , DA_KNX1 , 01 , PASSIVE , KNX_Read_1 , DPT1 , 0/5/02 , KNX_VAR1
```

Map descriptor
KNX_VAR1 maps the value for the KNX group address 0:5:1
Map descriptor
KNX_VAR1_LINK maps the value for the KNX group address 0:5:2

Both map descriptors use the same data array (DA_KNX1), but reference the data at different offsets.

ARS is the most common function for KNX clients, as it will send out a KNX read to the group on start-up, and from then on it will be passive.

The data point type must be compatible with the KNX group address it is associated with.

The following data types are also available:
DPT1, DPT2,
DPT3, DPT4,
DPT5, DPT6,
DPT7, DPT8,
DPT9, DPT12,
DPT13, DPT14,
DPT15, DPT17,
DPT18, DPT20

The associated group address for the map descriptor is specified in this field.

The linked map descriptor pair must be of the same data type.

Example2

The QuickServer is configured to have a readable KNX point. By default the QuickServer is not intended to respond to group read message, so this map descriptor is specifically set to yes. This will be used to provide data to the KNX network if a read response is requested.

```
// Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Group_Address , Read_Response
KNX_VAR1 , DA_KNX1 , 00 , PASSIVE , KNX_Read_2 , DPT1 , 0/5/05 , Yes
```

The map descriptor is set to respond to the group request.

Example3

The QuickServer is configured to write to a group address when the data array is updated.

```
// Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Group_Address
KNX_LIGHT1 , DA_KNX2 , 00 , WRBX , KNX_Write_1 , DPT1 , 0/5/03
KNX_LIGHT2 , DA_KNX2 , 01 , WRBX , KNX_Write_2 , DPT1 , 0/5/04
```

The map descriptor is set to write a change of value to the group specified.

Appendix A. REFERENCE

Appendix A.1. KNX Data Types

KNX Data point Name	Description	Recommended Data Array Value
DPT1	1-bit Binary Switch	BYTE
DPT2	2-bit Step Control	BYTE
DPT3	4-bit Dimming	BYTE
DPT4	8-bit Set	BYTE
DPT5	8-bit Unsigned Value	BYTE
DPT6	8-bit Signed Value	SINT16
DPT7	16-bit Unsigned Value	UINT16
DPT8	16-bit Signed Value	SINT16
DPT9	16-bit Floating point value	FLOAT
DPT12	32-bit Unsigned Value	UINT32
DPT13	32-bit Signed Value	SINT32
DPT14	32-bit Float	FLOAT
DPT15	32-bit Access	UINT32
DPT17	8-bit Scene Number	BYTE
DPT18	8-bit Scene Control	BYTE
DPT20	8-bit Enum Value	BYTE

Appendix B. TROUBLESHOOTING

Appendix B.1. Driver Messages

Msg #	Msg Screen	Screen message	Meaning	Suggested Solution
1	DRIVER	KNX : KNX DEV CMD=%x, SEQ=%d, ACK NAK	Driver is ACKing or NACKing a KNX protocol specific command message on the network	Informational to aid in checking network health