



Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8705-11
FA403 Serial' Protocol
Driver

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version: 1.00a
Document Revision: 1

TABLE OF CONTENTS

1. FA403 DRIVER 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 3rd Party Equipment 4

 2.2.1. *Required 3rd Party Hardware* 4

 2.2.2. *Required 3rd Party Software* 4

 2.2.3. *Required 3rd Party Configuration*..... 4

 2.2.4. *Optional Items*..... 4

3. HARDWARE CONNECTIONS..... 5

 3.1. *Block Diagram* 5

 3.2. *Serial Connection Information*..... 6

 3.3. *Hardware Connection Tips / Hints*..... 7

4. CONFIGURING THE FIELDSEVER AS A FA403 PASSIVE CLIENT..... 8

 4.1. Data Arrays/Descriptors 8

 4.2. Client Side Connection Descriptions 9

 4.3. Client Side Node Descriptors 11

 4.4. Client Side Map Descriptors 11

 4.4.1. *FieldServer Related Map Descriptor Parameters* 11

 4.4.2. *Driver Related Map Descriptor Parameters* 12

 4.5. Map Descriptor (MD) Example 1 – Transmitter Status..... 13

5. CONFIGURING THE FIELDSEVER AS A FA403 ACTIVE SERVER 14

APPENDIX A. TROUBLESHOOTING TIPS..... 15

 Appendix A.1. Connection Tips & Hints 15

 Appendix A.2. Driver Error and Warning Messages 15

 Appendix A.3. Driver Stats – How an Upstream Device can Monitor Performance..... 17

6. REVISION HISTORY 18

1. FA403 Driver Description

This driver is used to exchange data between a FieldServer and a ‘Inovonics FA403 Serial Receiver’.

- The driver can will process ‘Transmitter Status’ Messages and ignore all others
- The driver will be a passive client – listening passively for messages from the FA403.
- The driver will store the date and time of the last message was received. This data will be stored in a Data Array and can be served to a remote client using another protocol.
- Each time a Transmitter Status message is received the driver will validate the message. If valid it will extract the System ID and the Transmitter ID. These will be used to build an index into a Data Array where the status information will be stored. Thus each system_id / transmitter_id combination will have its status stored at a unique location in a data array. Depending on how the server (say Modbus) is configured, it will be possible to allocate a unique address to each devices’ status. For example status of 1,1 could be associated with register 40001, of 1,2 with 40002 etc.
- The driver will be configured to accept / ignore messages containing status information for specific system_id / transmitter_id combinations for a range of combinations.
- The driver will expect the message to be sent in hex format.

- The driver is a serial driver using a RS232 / RS485 serial ports to connect between the FieldServer and the FA403 system.
-
- The driver is fully compatible with other FieldServer drivers and meets FieldServer’s quality assurance standards. The driver was developed by Chipkin Automation Systems, an Approved FieldServer Integrator.

FIELDSEVER MODE	NODES	COMMENTS
PASSIVE CLIENT	1	MESSAGES DO NOT CONTAIN DESTINATION AND SOURCE NODE ADDRESSES. THEREFORE ONLY ONE <i>FA403 SERIAL RECEIVER</i> CAN BE CONNECTED TO EACH PORT.
SERVER	N/A	

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

2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Hardware

Part #	Description
FA403 SERIAL RECEIVER	This device allows 3 rd party devices to receive data from FA series transmitters.

2.2.2. Required 3rd Party Software

None required to enable this driver/

2.2.3. Required 3rd Party Configuration

None required to enable this driver/

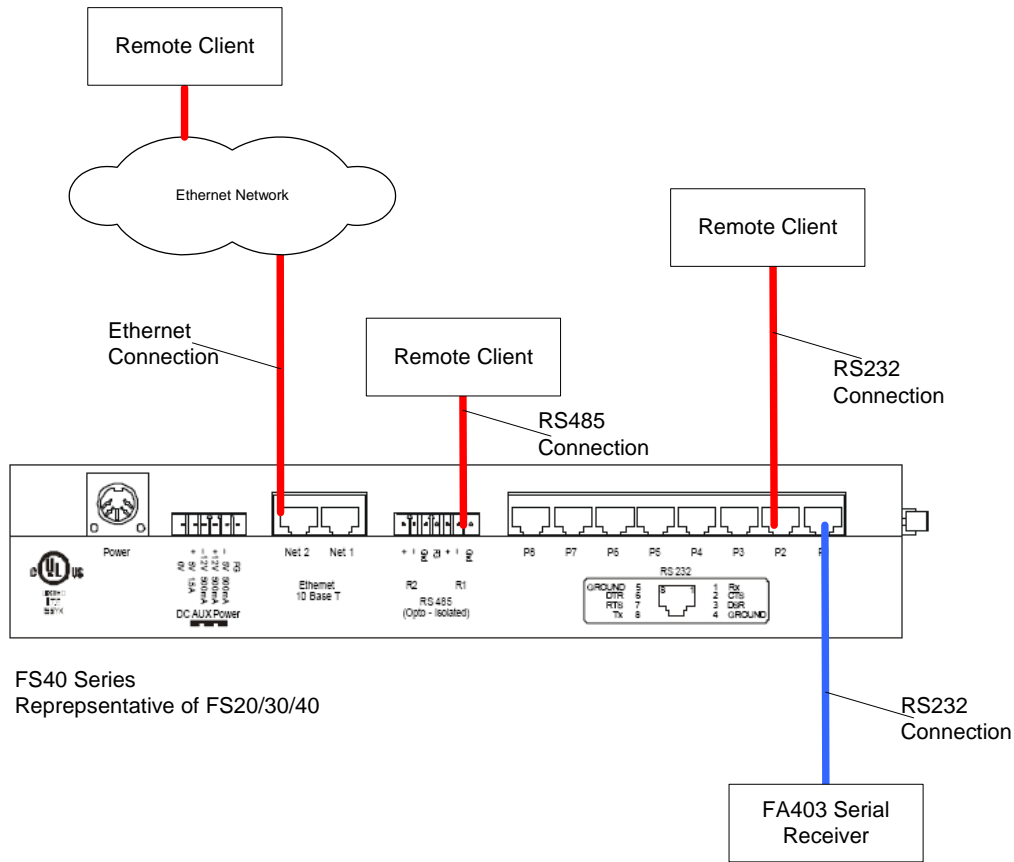
2.2.4. Optional Items

PART #	Vendor/Manufacturer	Description

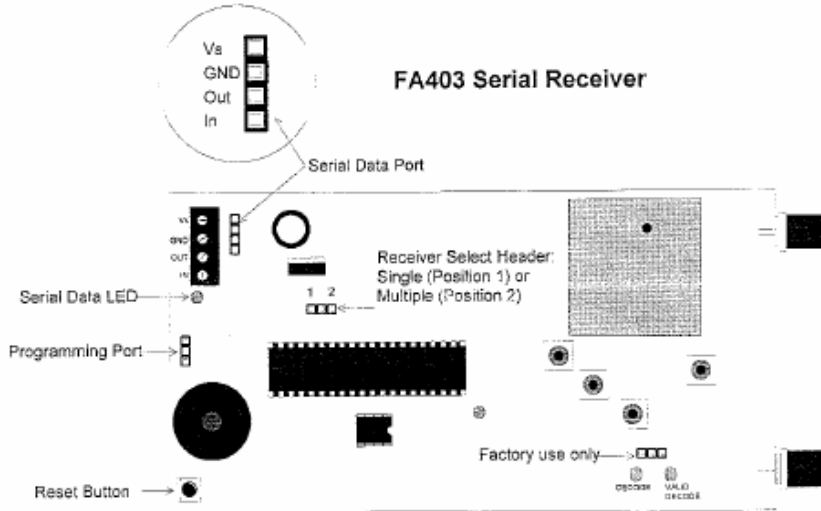
3. Hardware Connections

The FieldServer is connected to the R-Link Device as shown in connection drawing. Configure the R-Link according to manufacturer’s instructions

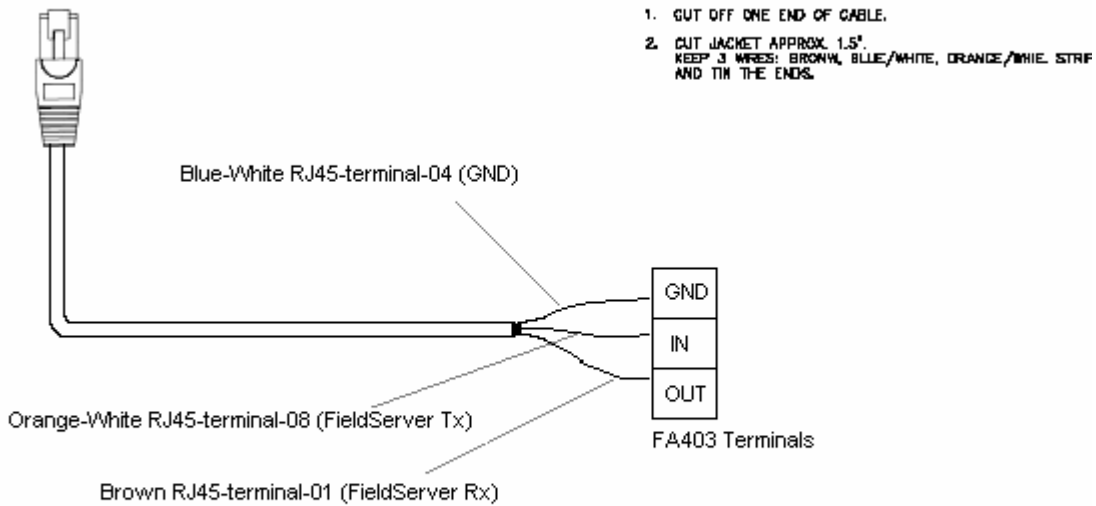
3.1. Block Diagram



3.2. Serial Connection Information



Signal	Host RS-232 Connector		FA403 Connector
	9-pin	25-pin	
Ground	5	7	Ground
Data Out (TXD)	3	2	IN
Data In (RXD)	2	3	OUT



3.3. Hardware Connection Tips / Hints

None

4. Configuring the FieldServer as a FA403 Passive 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 FA403 Serial Receiver.

4.1. Data Arrays/Descriptors

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for FA403 Driver 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.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

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, Bit, UInt16, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_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

Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,      Data_Array_Format,      Data_Array_Length
DA_AI_01,             UInt16,                 200
DA_AO_01,             UInt16,                 200
DA_DI_01,             Bit,                    200
DA_DO_01,             Bit,                    200
```

4.2. Client Side Connection Descriptions

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer. For FS30 Series bridges, only P2 can be used as the Primary serial port because of the DTR signaling which is not present on P2.	P1-P8, R1-R2 ¹
Protocol	Specify protocol used	FA403
Baud*	Specify baud rate FA403 only supports 9600 The FieldServer supports all standard baud rates between 300 and 115200	110 – 115200, standard baud rates only
Parity*	Specify parity	FA403 supports None Driver Supports: Even, Odd, None , Mark, Space
Data_Bits*	Specify data bits	FA403 supports 8 , Driver supports 7,8
Stop_Bits*	Specify stop bits	FA403 supports 1 Driver supports 1,2
Handshaking*	Specify hardware handshaking	None

¹ Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

Example

//	Client Side Connections				
Connections					
Port,	Protocol,	Baud,	Parity,	Handshaking,	Poll_Delay
P1,	FA403,	9600,	None,	None,	0.100s

4.3. Client Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	This commonly used Fieldserver parameter is not used by this driver	
Protocol	Specify protocol used	FA403,
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ²

Example

```
// Client Side Nodes

Nodes
Node_Name,          Node_ID,          Protocol,          Connection
PLC 1,              1,              FA403,           P1
```

4.4. Client Side Map Descriptors

4.4.1. FieldServer Related 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 “Data Array” section above
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in “Data Array” section above
Function	Function of Client Map Descriptor	Passive

² Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

4.4.2. 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 “Client Node Descriptor” above
Length	<p>Length of Map Descriptor</p> <p>See the examples for more specific information.</p> <p>The driver will always use 256 consecutive storage locations – one for each transmitter. The length parameter is not used by the driver.</p>	Positive whole numbers.
Address	Not used by this driver	Zero and Positive whole numbers.
fa403_sys_id	Transmitters with whose System_ID matches the value specified here will be stored using this Map Descriptor	Numbers 0,1,2...
fa403_xmit_id	<p>Transmitters with whose Transmitter_ID is equal to or larger than the value specified by this parameter.</p> <p>Typically this parameter is set to zero.</p>	Numbers 0,1,2...

4.5. Map Descriptor (MD) Example 1 – Transmitter Status.

This Map Descriptor is used to store incoming status messages from Transmitters. The driver located the Map Descriptor (MD) to be used to store the message by matching the System_ID. If the driver cant find an MD with a matching System_ID then the message is ignored and no data is stored. If a MD is found, then the driver matches the Xmitter ID. The transmitter ID in the received message must be equal to or larger than the value specified by the Xmit_ID parameter.

For example: Is a message is received with System_ID=1 and Transmitter_ID=12 then the driver will use the MD below and store the data at offset 11.

// Client Map Descriptors							
Map_Descriptors							
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name	FA403_Sys_ID	FA403_Xmit_ID	Length
Store_Sys1_Status,	DA_SYS_01,	0,	passive,	Node_FA403,	1,	0,	256,

We suggest each MD is named uniquely

The status data obtained will be stored in the DA named here, starting at this offset.

The driver will store the status byte's value.

Driver wait passively for messages

The Node name connects the MD to a Node Descriptor which in turn is connected to a Port.

The Node_ID parameter on the Node Descriptor tells the driver the panel number.

Only Status data from field units with a System_ID code =1 will be stored using this Map Descriptor. Others will be ignored or stored using other MD's if defined.

Read and store data for 48 relays.

The driver doesn't use this parameter. It assumes that 256 consecutive locations in the data array will be used for the 256 possible transmitter ID's.

5. Configuring the FieldServer as a FA403 Active Server

The server side is used to test the client side as part of an ongoing quality assurance program. If you are interested in using it please contact us.

Appendix A. Troubleshooting tips

Appendix A.1. Connection Tips & Hints

None

Appendix A.2. Driver Error and Warning Messages

A number of corrective actions require that you edit the configuration file. When you have completed the edit, you must download the modified configuration file to the FieldServer and reset the FieldServer for the changes to take effect.*

Error Message	Meaning and Corrective Action
FA403:#01 FYI. Rcvd Stat=%#02x for Id=%d:%d	<p>Each time the driver receives a valid status message it prints this message on the driver message screen. You can safely ignore the messages. They may be used to verify that the driver received a message.</p> <p>If you see one of these messages and it is not immediately followed by msg #2 then you know that the driver is not storing data from the message. This is typically the case if the driver has not been configured to store message from that transmitter.</p>
FA403:#02 FYI. Store Stat=%#02x for Id=%d:%d in DA=%s Off=%d	<p>Each time the driver receives a valid status message and find a place to store the data it prints this message on the driver message screen. You can safely ignore these messages.</p> <p>They may be used to verify that the driver received a message and that the message was not ignored.</p>
FA403:#03 FYI. Use a DA called <%s> to expose diagnostic info.	Read appendix Appendix A.3 for more info.
FA403:#04 Err. Undefined sys_id	Every Map Descriptor must have the FA403_Sys_ID defined. Correct the configuration file. Download the modified file to the FieldServer and reset the FieldServer for the changes to take effect.
FA403:#05 Err. Undefined xmit_id	Every Map Descriptor must have the FA403_Xmit_ID defined. Correct the configuration file. Download the modified file

	to the FieldServer and reset the FieldServer for the changes to take effect.

Appendix A.3. Driver Stats – How an Upstream Device can Monitor Performance.

If this driver is appropriately configured , it can expose operation statistics in a Data Array which can be monitored by a remote device to check that the driver is performing without error.

The lines from the example below can be cut and pasted into a configuration file to expose these stats.

Data_Arrays,		
Data_Array_Name,	Data_Format,	Data_Array_Length,
FA403-Stats,	UINT32,	1000,

Offset	Description
1	Number of bytes received from the FA403 Serial Receiver
2	Number of message ignored by the driver. The driver has only been programmed to recognize status messages. This count may include fragments of other messages.
3	Number of status messages received.
4	Number of status messages received that had checksum errors.
5	The time in seconds since 1970 that the last status message stored.
6	Extracted from the most recent status message. The Factory Code's 1 st byte. Overwritten by each new status message
7	Extracted from the most recent status message. The Factory Code's 2nd byte. Overwritten by each new status message
8	Extracted from the most recent status message. The System ID. Overwritten by each new status message
9	Extracted from the most recent status message. The TRansmitter ID. Overwritten by each new status message
10	Extracted from the most recent status message. The Status byte. Overwritten by each new status message
11	Each time the driver stores data this count is incremented by 1.

6. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
08Apr06	PMC	PMC	1.00a	1	Created

