

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8705-26

Barrington Lanstar Serial Driver

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after Aug 20, 2012

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1 Barrington Lanstar Driver Description	4
2 Driver Scope of Supply	5
2.1 Supplied with this driver.....	5
3 Hardware Connections	6
3.1 Hardware Connections	6
3.2 Block Diagram.....	10
4 Configuring the FieldServer as a Barrington Lanstar Client	11
4.1 Data Arrays	12
4.1.1 <i>Data Arrays - Example</i>	12
4.2 Client Side Connections.....	13
4.2.1 <i>Client Side Connection Descriptions - Example</i>	14
4.3 Client Side Nodes.....	15
4.3.1 <i>Client Side Nodes - Example</i>	15
4.4 Client Side Map Descriptors	16
4.4.1 <i>FieldServer Related Map Descriptor Parameters</i>	16
4.4.2 <i>Driver Related Map Descriptor Parameters</i>	17
4.5 Examples.....	19
4.5.1 <i>Map Descriptor Example 1 – Read Points</i>	19
4.5.2 <i>Map Descriptor Example 2 – Store Panel Troubles</i>	20
5 Configuring the FieldServer as a Barrington Lanstar Server	21
Appendix 1.1. Scaling	22
Appendix 1.2. Supported Communications functions	23

Appendix 1.3. Exposing Driver Stats 24

Appendix 1.4. Revision History 25

1 Barrington Lanstar Driver Description

This driver is used to exchange data between a FieldServer and a 'MicroSTAR' devices by polling the LanStar they are connected to. When this driver is used the gateway must be connected to 1 of the 4 ports on the Lanstar.

Each MicroSTAR device has 4x AO, 4x AI, 4x DI, 4x DO and 4x Counters. Setpoints, virtual Digital Outputs, any virtual points or virtual Microstars all reside in the memory of the LanStar and are thus accessible using this driver.

The driver can monitor the status of these points and write to the outputs.

This is an active client driver. The FieldServer is the client. The LanSTAR devices are the passive servers. The Client sends messages and processes responses from the LanSTAR's.

The driver is a serial driver using a RS232 serial ports to connect between the FieldServer and the LanSTAR's..

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.

Only Client functionality have been implemented. This means that you can use the driver as a client to monitor/command the devices. Server functionality is not supported thus you cannot use it a server to make another type of device appear as if it were a LanSTAR device.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
ACTIVE CLIENT	32	A MAXIMUM OF 16 DEVICES MAY BE CONNECTED PER TRUNK. THIS IS A LIMITATION OF THE PROTOCOL WHICH ONLY ALLOWS 32 POSSIBLE ADDRESSES AS 2 TRUNKS CAN BE CONNECTED TO ONE LANSTAR.
Active Server (Simulate a Panel)	0	Not supported or documented.

2 Driver Scope of Supply

2.1 Supplied with this driver

FieldServer Technologies PART #	Description
8915-10	No specific cables are shipped with this driver. A generic RJ45 Ethernet cable is shipped with the hardware.
FS-8705-26	Driver Manual.

3 Hardware Connections

Multiple WorkStation protocols and connection supported. See list of FieldServer Drivers.

3.1 Hardware Connections

TWO METHODS

For each Lanstar you are connecting to you will need one cable.

NULL MODEM

1. Obtain a Null Modem Cable
2. You need a female connector to plug into the lstar. If the other end is a male then cut it off. If both are female then cut any end off.
3. Strip back some of the wires
4. Using a multimeter find the 3 wires connected to pins 2,3 and 5 on the other end.

Null Modem 9-Pin connector End	Gateway (bottom TB)
Pin2	Terminal 1 (leftmost bottom)
Pin3	Terminal 2 (2 nd from left bottom)
Pin5	Terminal 3 (3 rd from left bottom)

OR

PATCH CABLE

1. Use an Ethernet Patch cable and the supplied RJ45-9pin converter.
2. Cut one end off the cable
3. Strip back orange/white brown/white blue/white
4. Connect Orange/White to Terminal 1 (leftmost bottom)
5. Connect Brown to Terminal 2 (2nd from left bottom)
6. Connect Blue/White to Terminal 3 (3rd from left bottom)

PATCH CABLE IMAGES



wn.





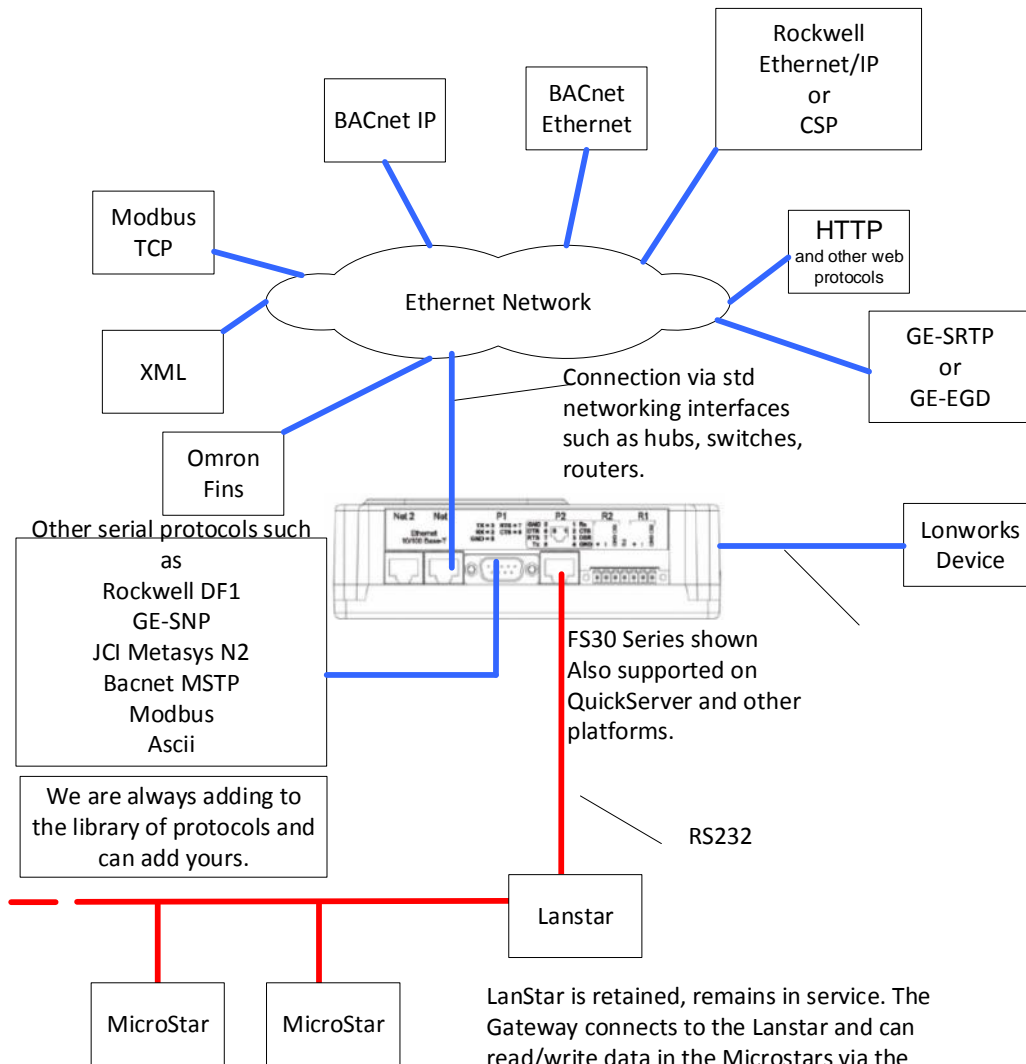


QuickServer Connections

The other end of this cable connect to RS232 port on the Lanstar. That connection is shown in other images.

3.2 Block Diagram

FS30 shown as representing FS20, FS30 and FS40



Connection via std networking interfaces such as hubs, switches, routers.

FS30 Series shown Also supported on QuickServer and other platforms.

RS232

LanStar is retained, remains in service. The Gateway connects to the Lanstar and can read/write data in the Microstars via the Lanstar. **A great solution if you need to keep the LanStar in service.**

FS-8705-26 Barrington LanStar Protocol (Get Microstar data via the LanStar)

4 Configuring the FieldServer as a Barrington Lanstar 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 FS).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with an Barrington Lanstar system.

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

4.1 Data Arrays

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.	Recommended: Bit, UInt16, Also Supported: Float, UInt32, 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

4.1.1 Data Arrays - Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,           Data_Format,           Data_Array_Length,
LanstarStats,              UNT16,                 200
```

4.2 Client Side Connections

Create one connection for each Barrington Lanstar serial port. Each connection can only be used to connect to a single Barrington Lanstar interface/port.

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8
Protocol	Specify protocol used	Barrington Lanstar
Baud*	Specify baud rate	Driver Supports : 110; 300; 600; 1200; 2400; 4800; 9600 ; 19200; 28800; 38400; 57600 Baud <i>Barrington Lanstar supports:</i> 9600
Data_Bits *	Specify parity	Driver Supports : 7, 8 <i>Barrington Lanstar supports:</i> 7
Stop_Bits*	Specify data bits	Driver Supports : 1,2 <i>Barrington Lanstar supports:</i> 1
Parity *	Specify stop bits	Driver Supports : Odd, Even, None <i>Barrington Lanstar supports:</i> Even

4.2.1 Client Side Connection Descriptions - Example

```
// Client Side Connections
```

Connections				
Port,	Baud,	Parity,	Protocol, Data_Bits ,	Stop_Bits
P1,	9600,	Even,	Barrington Lanstar , 7	, 1

4.3 Client Side Nodes

Create one Node per FACP in the network only.

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters NB ! The name does not need to correspond to the Node name configured in the Barrington Lanstar system.
Node_ID	Station address of physical server node This parameter is not used directly by the driver. We recommend that a unique Node ID's be given to each node.	0-64 Corresponds to the Node numbers of panels. Master node is typically zero.
Protocol	Specify protocol used	Barrington Lanstar

4.3.1 Client Side Nodes - Example

```
// Client Side Nodes

Nodes

Node_Name,           Node_ID,           Protocol,           Connection
MainPanel,           0,                 Barrington Lanstar , 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

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
Data_Type	This commonly used parameter is not used by this driver.	
Length	Length of Map Descriptor Reserves space in the Data Array.	Set the value to 1. One map descriptor is required for each point.
Address	The Point Number	0,1,2,3....
Lstar_Point_Type	Point Type	AI, AO, DI, DO, SP,

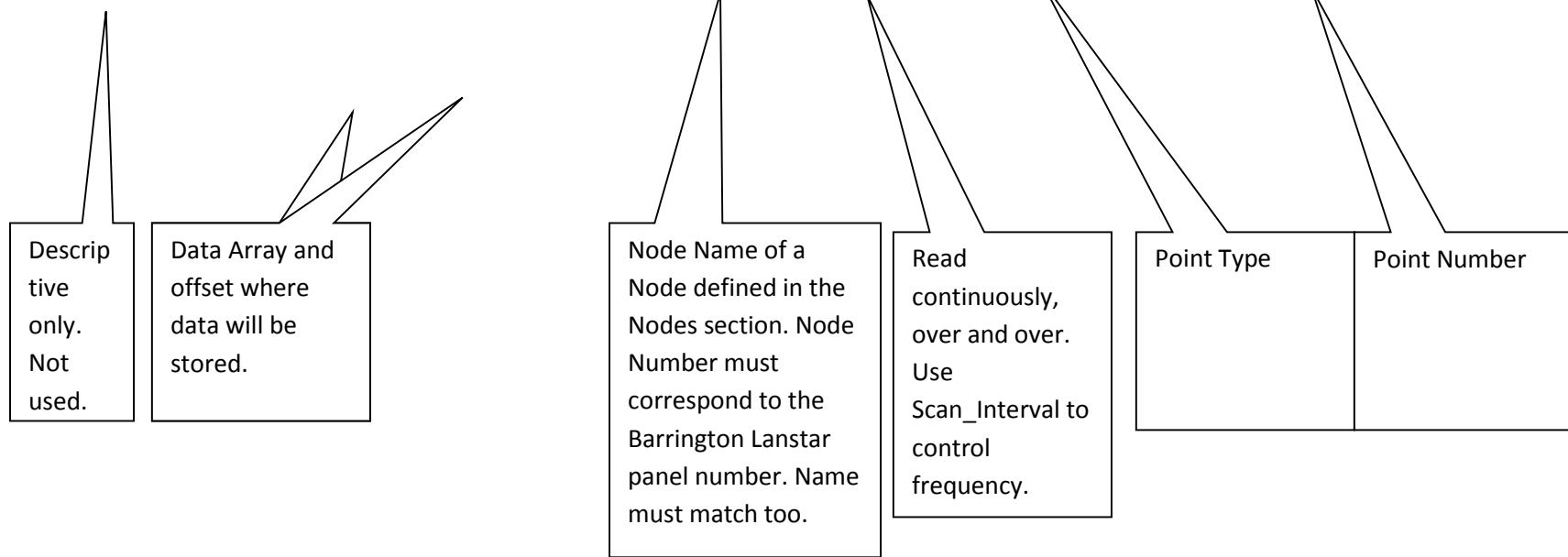
4.5 Examples

4.5.1 Map Descriptor Example 1 – Read Points

In this example three different points are reach, each from a different Node.

Map_Descriptors

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Node_Name	Function	Length	Lstar_Point_Type	Address
Read_DO	DA_MSTAR00_DO	15	MSTAR00	rdbc	1	DO	15
Read_SP	DA_MSTAR01_SP	13	MSTAR01	rdbc	1	SP	13
Read_AI	DA_MSTAR03_AI	0	MSTAR03	rdbc	1	AI	0



4.5.2 Map Descriptor Example 2 – Store Panel Troubles

In this example one point is written on update and another continuously.

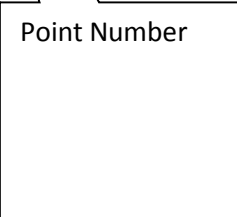
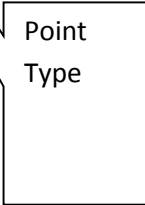
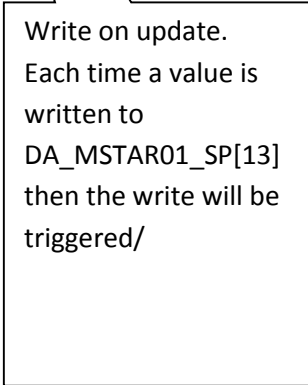
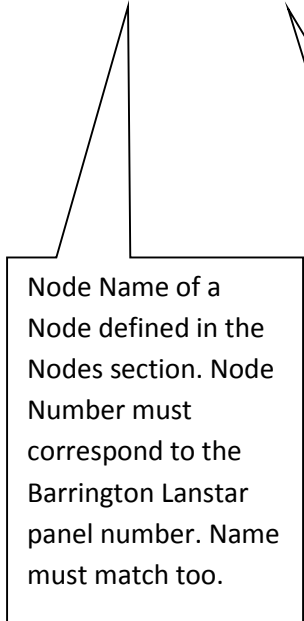
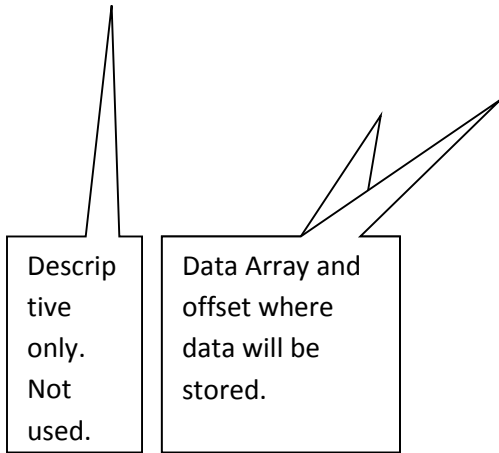
Map_Descriptors

Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length , Lstar_Point_Type , Address

Read_DO , DA_MSTAR00_DO , 15 , MSTAR00 , wrbc

Read_SP , DA_MSTAR01_SP , 13 , MSTAR01 , wrbx , 1 , SP , 13

Write continuously.



5 Configuring the FieldServer as a Barrington Lanstar Server

This driver cannot be used to emulate an Barrington Lanstar Panel

Appendix 1 – Advanced Topics

Appendix 1.1. Scaling

Points can be scaled when they are read (and thus a scaled version is stored in the Data Arrays). They can be scaled by a task that executes on data update or a fixed frequency or they can be scaled when they are served.

Example: Scale before stored.

Map_Descriptors

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length ,
Lstar_Point_Type , Address , Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale ,
Node_High_Scale
```

```
Read_AI           , DA_MSTAR03_AI   , 0           , MSTAR03   , rdbc     , 1           ,
AI                , 0,0,100,32,212
```

Example: Scale before served. ie scaled number is served.

Map_Descriptors

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length ,
Object_Type , Object_Instance , Relinquish_Default ,Data_Array_Low_Scale ,
Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
```

```
Serve_AI          , DA_MSTAR03_AI   , 0           , sMSTAR03  , server   , 1           ,
AI                , 0                , -           , 0,100,0,10
```

Appendix 1.2. Supported Communications functions

Always check the Data Sheet for an accurate and up to date list.

Function	Notes
Poll Request	Can Write to Do,Ao and SP points including virtual Do's.
Poll Request	Read. Can read Ai,Ao,Di,Do,SP,Counters including virtual DO's on a real MicroSTAR

Driver cannot read/write data to virtual microSTAR's.

Appendix 1.3. Exposing Driver Stats

The driver makes some of its operating statistics available in a Data Array where they can be read by a remote client. The lines from the example below can be cut and pasted into a configuration file.

```
Data_Arrays,
Data_Array_Name,      Data_Format,      Data_Array_length,
LanstarStats,        UINT32,          1000,
```

```
// Offset 37 IN THE STATS ARRAY CONTROL THE PRINTING OF DEBUG CODE.
```

```
//here we preset the value to 1 (debug on)
```

```
// so that the driver starts running with debug on
```

```
// This seems to slow the driver down enough to work well
```

```
// When debug is off we gets lots of errors.
```

Preloads

```
Data_Array_Name , Preload_Data_Index , Preload_Data_Value , length
```

```
Lstarstats , 37 , 1 , 1
```


Appendix 1.4. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
2014Aug20	PMC		1.0	1.0	Created