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>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barrington Lanstar Driver Description</td>
</tr>
<tr>
<td>2</td>
<td>Driver Scope of Supply</td>
</tr>
<tr>
<td>3</td>
<td>Hardware Connections</td>
</tr>
<tr>
<td>4</td>
<td>Configuring the FieldServer as a Barrington Lanstar Client</td>
</tr>
<tr>
<td>5</td>
<td>Configuring the FieldServer as a Barrington Lanstar Server</td>
</tr>
</tbody>
</table>

## 1 Barrington Lanstar Driver Description

### 2 Driver Scope of Supply

2.1 Supplied with this driver

### 3 Hardware Connections

3.1 Hardware Connections

3.2 Block Diagram

### 4 Configuring the FieldServer as a Barrington Lanstar Client

4.1 Data Arrays

#### 4.1.1 Data Arrays - Example

4.2 Client Side Connections

#### 4.2.1 Client Side Connection Descriptions - Example

4.3 Client Side Nodes

#### 4.3.1 Client Side Nodes - Example

4.4 Client Side Map Descriptors

#### 4.4.1 FieldServer Related Map Descriptor Parameters

#### 4.4.2 Driver Related Map Descriptor Parameters

4.5 Examples

#### 4.5.1 Map Descriptor Example 1 – Read Points

#### 4.5.2 Map Descriptor Example 2 – Store Panel Troubles

### 5 Configuring the FieldServer as a Barrington Lanstar Server

Appendix 1.1. Scaling

Appendix 1.2. Supported Communications functions
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

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

2 Driver Scope of Supply

2.1 Supplied with this driver

<table>
<thead>
<tr>
<th>FieldServer Technologies PART #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8915-10</td>
<td>No specific cables are shipped with this driver.</td>
</tr>
<tr>
<td></td>
<td>A generic RJ45 Ethernet cable is shipped with the hardware.</td>
</tr>
</tbody>
</table>


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 Istar. 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\textsuperscript{nd} from left bottom)
Pin5 Terminal 3 (3\textsuperscript{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 (2\textsuperscript{nd} from left bottom)
6. Connect Blue/White to Terminal 3 (3\textsuperscript{rd} from left bottom)
PATCH CABLE IMAGES

Connect the RS232 cable to TB4 as shown.
Use one of these 4 ports
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

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 a 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

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Function</th>
<th>Legal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data_Array_Name</td>
<td>Provide name for Data Array</td>
<td>Up to 15 alphanumeric characters</td>
</tr>
<tr>
<td>Data_Array_Format</td>
<td>Provide data format. Each Data Array can only take on one format.</td>
<td><strong>Recommended</strong>: Bit, Ulnt16, Also Supported: Float, Uint32, SInt16, Packed_Bit, Byte, Packed_Bit, Swapped_Bit</td>
</tr>
<tr>
<td>Data_Array_Length</td>
<td>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.</td>
<td>1-10,000</td>
</tr>
</tbody>
</table>

4.1.1 Data Arrays - Example

```c
// 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.

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

```markdown
// Client Side Connections

<table>
<thead>
<tr>
<th>Connections</th>
<th>Port,</th>
<th>Baud,</th>
<th>Parity,</th>
<th>Protocol, Data_Bits, Stop_Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1,</td>
<td>9600,</td>
<td>Even,</td>
<td>Barrington Lanstar, 7, 1</td>
<td></td>
</tr>
</tbody>
</table>
```
4.3 Client Side Nodes

Create one Node per FACP in the network only.

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Nodes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Function</th>
<th>Legal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node_Name</td>
<td>Provide name for node</td>
<td>Up to 32 alphanumeric characters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NB! The name does not need to correspond to the Node name configured in the Barrington Lanstar system.</td>
</tr>
<tr>
<td>Node_ID</td>
<td>Station address of physical server node</td>
<td>0-64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corresponds to the Node numbers of panels. Master node is typically zero.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Specify protocol used</td>
<td>Barrington Lanstar</td>
</tr>
</tbody>
</table>

4.3.1 Client Side Nodes - Example

```plaintext
// 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

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Function</th>
<th>Legal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map_Descriptor_Name</td>
<td>Name of this Map Descriptor</td>
<td>Up to 32 alphanumeric characters</td>
</tr>
<tr>
<td>Data_Array_Name</td>
<td>Name of Data Array where data is to be stored in the FieldServer</td>
<td>One of the Data Array names from “Data Array” section above</td>
</tr>
<tr>
<td>Data_Array_Offset</td>
<td>Starting location in Data Array</td>
<td>0 to maximum specified in “Data Array” section above</td>
</tr>
<tr>
<td>Function</td>
<td>Function of Client Map Descriptor..</td>
<td>Passive</td>
</tr>
</tbody>
</table>
### 4.4.2 Driver Related Map Descriptor Parameters

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Function</th>
<th>Legal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node_Name</td>
<td>Name of Node to fetch data from</td>
<td>One of the node names specified in “Client Node Descriptor” above</td>
</tr>
<tr>
<td>Data_Type</td>
<td>This commonly used parameter is not used by this driver.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length of Map Descriptor</td>
<td>Set the value to 1. One map descriptor is required for each point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserves space in the Data Array.</td>
</tr>
<tr>
<td>Address</td>
<td>The Point Number</td>
<td>0,1,2,3,...</td>
</tr>
<tr>
<td>Lstar_Point_Type</td>
<td>Point Type</td>
<td>AI, AO, DI, DO, SP,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.5 Examples

4.5.1 Map Descriptor Example 1 – Read Points

In this example three different points are read, 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

Data Array and offset where data will be stored.

Node Name of a Node defined in the Nodes section. Node Number must correspond to the Barrington Lanstar panel number. Name must match too.

Read continuously, over and over. Use Scan_Interval to control frequency.

Point Type

Point Number
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 , 1 , DO , 15

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

Data Array and offset where data will be stored.

Node Name of a Node defined in the Nodes section. Node Number must correspond to the Barrington Lanstar panel number. Name must match too.

Write on update. Each time a value is written to DA_MSTAR01_SP[13] then the write will be triggered/

Write continuously.

Descriptive only. Not used.

Point Type

Point Number
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. I.e. 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poll Request</td>
<td>Can Write to Do, Ao and SP points including virtual Do’s.</td>
</tr>
<tr>
<td>Poll Request</td>
<td>Read. Can read Ai, Ao, Di, Do, SP, Counters including virtual DO’s on a real MicroSTAR</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Resp</th>
<th>Format</th>
<th>Driver Ver.</th>
<th>Doc. Rev.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014Aug20</td>
<td>PMC</td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>Created</td>
</tr>
</tbody>
</table>