Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-57 Profibus MS

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version: 1.01
Document Revision: 7
# TABLE OF CONTENTS

1. PROFIBUS MULTI-SLAVE 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

3. HARDWARE CONNECTIONS ................................................................... 5
   3.1. Hardware Connection Tips / Hints .................................................. 5

4. CONFIGURING THE FIELDSERVER AS A PROFIBUS MULTI-SLAVE MASTER 6
   4.1. Data Arrays/Descriptors .................................................................. 6
   4.2. Server Side Connection Descriptions ............................................. 7
   4.3. Server Side Node Descriptors ....................................................... 7
   4.4. Server Side Map Descriptors ......................................................... 8
       4.4.1. FieldServer Related Map Descriptor Parameters ..................... 8
       4.4.2. Driver Related Map Descriptor Parameters ............................... 8
       4.4.3. Timing Parameters ................................................................. 6
       4.4.4. Map Descriptor Example ......................................................... 9

APPENDIX A. ADVANCED TOPICS ......................................................... 10
   Appendix A.1. Using more than one Map Descriptor on a buffer .............. 10
   Appendix A.2. Understanding buffer types and Map Descriptor functions .......... 11
   Appendix A.3. Write-Throughs .............................................................. 12
   Appendix A.4. Map Descriptor Scan Interval affecting FieldServer performance ... 12
   Appendix A.5. The SST-5136 Profibus Multi-Slave Identification Number ........ 12
   Appendix A.6. Profibus Network Station limits ...................................... 12
   Appendix A.7. Interpreting Driver Connection Stats ................................ 12

APPENDIX B. TROUBLESHOOTING TIPS ............................................... 13
   Appendix B.1. FieldServer as a Profibus Slave .................................... 13
   Appendix B.2. Connection Tips & Hints ................................................ 14
1. Profibus Multi-Slave Description

The Profibus Multi-Slave driver allows the FieldServer to transfer data to and from a Profibus DP Master using Profibus DP protocol. The SST 5136-PBMS Communications Adapter card is included with the FieldServer. The FieldServer can emulate up to 125 DP Slave units each having an input and output buffer of 244 bytes each. The buffer data can be converted to other FieldServer data-types like Integers and Floating point numbers for seamless data-passing to other protocols. A maximum of 30,500 bytes of input and 30,500 bytes of output data can be transferred.

Max Nodes Supported

<table>
<thead>
<tr>
<th>FieldServer Mode</th>
<th>Nodes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>125</td>
<td>1-125 Slaves can be emulated</td>
</tr>
</tbody>
</table>
2. **Driver Scope of Supply**

2.1. **Supplied by FieldServer Technologies for this driver**

<table>
<thead>
<tr>
<th>FieldServer Technologies PART #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST 5136-PBMS</td>
<td>Communication Adapter complete with Phoenix Contact Screw Terminal and DB9 Connector</td>
</tr>
<tr>
<td>-</td>
<td>Profibus Tools CD</td>
</tr>
</tbody>
</table>

2.2. **Provided by the Supplier of 3rd Party Equipment**

2.2.1. **Required 3rd Party Hardware**

<table>
<thead>
<tr>
<th>Part #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Profibus Network With Cables And Terminators</td>
</tr>
</tbody>
</table>

2.2.2. **Required 3rd Party Software**

A Profibus Network Configuration Tool if not using the SST Tool supplied on the Profibus Tools CD.

2.2.3. **Required 3rd Party Configuration**

The FieldServer configured as a Profibus Multi-Slave can only transfer data with a configured DP Master on the Profibus network.
3. Hardware Connections

The FieldServer is connected to the Profibus network as shown in the connection drawing below:

**NOTES**
1) Connect only via the Phoenix or DB9 connector to the Profibus network.
2) Terminate DB9 by connecting pin 1-2 and 4-5 on the Phoenix connector

3.1. Hardware Connection Tips / Hints

See the 5136-PBMS Hardware Manual on the Tools CD for more information on connector pinouts and wiring practices.
4. Configuring the FieldServer as a Profibus Multi-Slave Master

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 an external Profibus DP Master.

The Profibus Multi-Slave driver builds a list of Slaves from the FieldServer configuration file. The sizes of the input and output buffers are calculated based on the mappings found. The maximum input and output buffer sizes are 244 bytes for each buffer.

Each emulated Slave is connected to the network with the Slave identification set to 0x0857. GS files are available on the Tools CD or obtainable from www.mysst.com.

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 Profibus Multi-Slave 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 “Server Side Nodes” section, and the data required from the Masters needs to be mapped in the “Server 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.

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Data_Array_Name</th>
<th>Data_Format</th>
<th>Data_Array_Length</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data_Array_Name</strong></td>
<td>Provide name for Data Array</td>
<td>Up to 15 alphanumeric characters</td>
<td></td>
</tr>
<tr>
<td><strong>Data_Array_Format</strong></td>
<td>Provide data format. Each Data Array can only take on one format.</td>
<td>Float, Bit, UInt16, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte</td>
<td></td>
</tr>
<tr>
<td><strong>Data_Array-Length</strong></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>
<td></td>
</tr>
</tbody>
</table>

Example

```
// Data Arrays
Data_Arrays
Data_Array_Name, Data_Format, Data_Array_Length
Float_Input, Float, 20
Word_Input, UInt16, 20
Byte_Input, Byte, 20
Bit_Input, Bit, 16
```
4.2. **Server Side Connection Descriptions**  

<table>
<thead>
<tr>
<th><strong>Section Title</strong></th>
<th><strong>Function</strong></th>
<th><strong>Legal Values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Column Title</strong></td>
<td><strong>Function</strong></td>
<td><strong>Legal Values</strong></td>
</tr>
<tr>
<td>Adapter</td>
<td>Adapter Name</td>
<td>Prof_DP_MS</td>
</tr>
<tr>
<td>Protocol</td>
<td>Specify protocol used</td>
<td>Profibus_MS</td>
</tr>
<tr>
<td>Profibus_MS_Baud</td>
<td>Profibus Multi-Slave Network Baudrate</td>
<td>9600, 19200, 93.75K, 187.5K, 500K, 750K, 1.5M, 3M, 6M, 12M</td>
</tr>
</tbody>
</table>

**Example**  

```//  Server Side Connections
Connections
Adapter, Protocol, Profibus_MS_Baud
Prof_DP_MS, Profibus_MS, 93.75K```

4.3. **Server Side Node Descriptors**  

<table>
<thead>
<tr>
<th><strong>Section Title</strong></th>
<th><strong>Function</strong></th>
<th><strong>Legal Values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Column Title</strong></td>
<td><strong>Function</strong></td>
<td><strong>Legal Values</strong></td>
</tr>
<tr>
<td>Node_Name</td>
<td>Provide name for node</td>
<td>Up to 32 alphanumeric characters</td>
</tr>
<tr>
<td>Node_ID</td>
<td>Slave station number</td>
<td>1-125</td>
</tr>
<tr>
<td>Protocol</td>
<td>Specify protocol used</td>
<td>Profibus_MS</td>
</tr>
<tr>
<td>Adapter</td>
<td>Adapter Name</td>
<td>Prof_DP_MS</td>
</tr>
</tbody>
</table>

**Example**  

```//  Server Side Nodes
Nodes
Node_Name, Node_ID, Protocol, Adapter
SLV01, 1, Profibus_MS, Prof_DP_MS```
4.4. Server 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 Server Map Descriptor</td>
<td>RDBC, WRBC</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 “Server Node Descriptor” above</td>
</tr>
<tr>
<td>DP_Type</td>
<td>Data type in which buffer data will be arranged</td>
<td>FLOAT, WORD, BYTE, BIT</td>
</tr>
<tr>
<td>DP_Buffer</td>
<td>Indicates which Slave buffer to access</td>
<td>INPUT for data sent to the Master, OUTPUT for data received from the Master</td>
</tr>
<tr>
<td>Address</td>
<td>Offset from beginning of data buffer</td>
<td>BIT offsets are in bits, other in bytes</td>
</tr>
</tbody>
</table>
| Length       | Number of DP_Type elements that is mapped to/from the Data Array | FLOAT – 1 .. 61  
WORD – 1 .. 122  
BYTE – 1 .. 244  
BIT – 1 .. 1952 |

4.4.3. Timing Parameters

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Function</th>
<th>Legal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan_Interval</td>
<td>Rate at which data is updated to / from the Multi-Slave buffers</td>
<td>≥0.001s</td>
</tr>
</tbody>
</table>
4.4.4. Map Descriptor Example.

The following Map Descriptor will produce input data for the Master to read:

// Server Side Map Descriptors
Map Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, Node_name, DP_Type, DP_Buffer, Address, Length, Scan_Interval
ByteToServer_01, Byte_Output, 0, WRBC, SLV01, Byte, Input, 0, 20, 1s

Active Map Descriptor sends the data to the card for transmission
Profibus Data type to convert the data to/from
Offset into the connection buffer. In bits for BIT, bytes for others
Slave which is sending the data
INPUT buffer is sent to the Server
Offset into the connection buffer. In bits for BIT, bytes for others

The following Map Descriptor will consume output data received from the Master:

// Server Side Map Descriptors
Map Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, Node_name, DP_Type, DP_Buffer, Address, Length, Scan_Interval
ByteFromServer_01, Byte_Input, 0, RDBC, SLV01, Byte, Output, 0, 20, 1s

Active Map Descriptor fetches data from the card
Node name for this Slave
Profibus Data type to convert the data to/from
OUTPUT buffer is data received from the Server
Offset into the connection buffer. In bits for BIT, bytes for others
Appendix A. Advanced Topics

Appendix A.1. Using more than one Map Descriptor on a buffer

It is important to correctly calculate Map Descriptor address and length values when using more than one Map Descriptor on a buffer. The driver does not allow overlaps in the address ranges used by Map Descriptors to prevent incorrect data mappings and double-polling of buffer data. The following byte lengths exist for the available DP_Type data types:

<table>
<thead>
<tr>
<th>DP_Type</th>
<th>Byte Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td>4</td>
</tr>
<tr>
<td>Word</td>
<td>2</td>
</tr>
<tr>
<td>Byte</td>
<td>1</td>
</tr>
<tr>
<td>Bit</td>
<td>Depends on number of bits. 8 bits per byte.</td>
</tr>
</tbody>
</table>

Example on using three Map Descriptors on the same buffer:

First Map Descriptor is used to read 1 Float from the start of the Master’s output buffer:
Address = 0 (start of buffer); Length = 1 (one float).

Second Map Descriptor is used to read 2 Words from the next available space in the Master’s output buffer:
Address = 0 (start of buffer) + 1 * 4 (byte length of first Map Descriptor) = 4; Length = 2 (ten words).

Third Map Descriptor used to read 18 Bits from the next available space:
(Note: Address has to be specified as a bit offset now)
Address = 0 (start of buffer) * 8 (bits per byte) + 1 * 4 * 8 (bit length of first Map Descriptor) + 2 * 2 * 8 (bit length of second Map Descriptor) = 64 bits
Length = 18 (eighteen bits).

Mapping of the buffer is shown below:
Appendix A.2. Understanding buffer types and Map Descriptor functions

The following DP_Buffer types and Map Descriptor functions are normally used together:

<table>
<thead>
<tr>
<th>DP_Buffer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>RDBC</td>
</tr>
<tr>
<td>Input</td>
<td>WRBC</td>
</tr>
</tbody>
</table>

The diagram below explains the data flows for both buffers.

Note that for the Output buffer, the data is coming from the external DP Master and is transferred into the Multi-Slave card’s internal input buffer. The Map Descriptor being of the RDBC function (Read Data Block Continuous) transfers the data from the card buffer into the FieldServer’s Data Array packing it correctly according to the specified DP_Type.

For the Input buffer, the data is written from the FieldServer’s Data Array by the WRBC function Map Descriptor (Write Data Block Continuous) into the Multi-Slave card’s internal output buffer from where it is transferred across the Profibus network to the DP Master’s input buffer.
Appendix A.3. Write-Throughs

Some FieldServer drivers support “write-throughs” which is simply defined as the ability of a Read Map Descriptor to automatically create a temporary write Map Descriptor to update the value of an external R/W (Read or Write) memory location. Since the Profibus DP Master contains separate Read-Only (Output) and Write-Only (Input) Buffers, write-throughs are disabled and not implemented on the Profibus Multi-Slave driver.

Appendix A.4. Map Descriptor Scan Interval affecting FieldServer performance

Note that the Map Descriptor scan interval simply sets the rate at which data is moved to or from the Data Arrays and the Profibus Multi-Slave’s local card buffers. The Profibus Network baudrate as set on the Connection under Profibus_MS_Baud determines the speed at which data is transmitted on the Profibus network itself. Very low scan intervals (less than one second) on multiple Map Descriptors of full buffer length are permitted although it could affect overall FieldServer performance significantly, even more so when using Bit Data Arrays. Generally, 1 second scan intervals are sufficient on Map Descriptors for monitoring and open loop control functions. Lower scan intervals may be more appropriate for closed loop control functions.

Appendix A.5. The SST-5136 Profibus Multi-Slave Identification Number

The SST-5136 Profibus Multi-Slave’s Identification Number is 0x0857.

Appendix A.6. Profibus Network Station limits

The Profibus Multi-Slave driver does not allow station address zero or station address 126 to be used for an emulated Slave. Station zero is reserved for a DP Master and station 126 is reserved for special systems. Station addresses from 1 to 125 are allowed.

Appendix A.7. Interpreting Driver Connection Stats

When viewing the connection stats with Ruinet, it is important to note that the transmit (Tx) stats on the Connection simply reflect messages that have been submitted to the Profibus card for transmission on the network. They do not necessarily indicate actual transmissions to a Profibus Master.

If the messages reach a Master and it communicates correctly with the FieldServer, the receive (Rx) stats are indicative of polls from a connected Profibus Master.
Appendix B. Troubleshooting tips

Appendix B.1. FieldServer as a Profibus Slave

If no communications is being experienced between a Profibus Master and a FieldServer Profibus Slave, check for the following most common issues:

- The baud rate for this driver must be specified in the Connections section of the configuration.
- Baud rate must match the baud rate being used on the network.
- The Node_ID must match the System_Station_Address in the configuration - this tells the FieldServer that the buffer referenced in the Node_ID refers to the buffers in the Profibus card attached to the FieldServer.
- The gse file must match the FieldServer hardware being used
  - FieldServer offers several hardware platforms that support Profibus. Each hardware platform requires its own gse file. Make sure that the gse file being used is the correct file for the hardware platform in use.
  - The gse file can be opened with a text editor. The Vendor and model name can be checked in this file. For the FieldServer X40 Profibus Master/Slave, the Vendor is “SST”, and the Model Name is “5136-PFB-ISA”. The current revision is 1.9, which uses Hardware revision 1.0 and software release 1.8.
- The byte count for the gse profile must match the byte count configured in the FieldServer
  - The gse being used is either going to allow the Slave (via the network configuration tool) to select the bytes to be used, or the byte make-up will be fixed and specified in the gse. Either way, the largest byte offset used by the Slave must match the largest byte offset configured in the FieldServer.
  - When selecting byte count with a network configuration tool, the total amount of bytes/words/bits setup per Input/Output buffer in the FieldServer configuration Map Descriptors must match those selected in the network configuration tool otherwise a connection length mismatch will occur and connection will not be established. The FieldServer shows the connection size it expects on the F (driver message) screen in the RUI utility in order that this value may be checked against the network configuration tool.
- The FieldServer may have bridge.bss loaded, which will configure it as a Master
  - The bridge.bss file is loaded by default when FieldServers ships with the default Profibus Master Configuration. This file must not exist in applications where FieldServer is being used as a Slave.
  - To check for the file, try uploading the file from the FieldServer using the RUI Utility. If it uploads successfully, then it will need to be deleted. See the FieldServer Utilities manual for instructions on how to do this.
- The FieldServer may be missing the pfbdp.ss1 file
  - This file is compulsory for all SST Profibus applications, and may be missing for some reason.
  - To check for the file, try uploading the file from the FieldServer using the RUI Utility. If it uploads successfully, then this is not the problem.
- The direction of the communications (Input/Output Buffer) may be incorrectly configured.
  - Buffer names can be confusing, especially when looking at the Slave. The buffers are named according to their direction of communication in the Master.
Hence, an Input buffer in the Slave will write data to the Master, and data will be received from the Master in the output buffer.

- Remember that as a Slave, the FieldServer will have to write data into the input buffer (function=wrbc), and read data from the output buffer (function=rdbc).

- The Profibus card may be faulty.
  - If this is the case, then there will be an error message in the FieldServer RUI error screen indicating that the card could not be reached. Check for this message. If it is there, try opening up the FieldServer and re-seating the card. If this does not help, contact FieldServer Technical support for jumper settings on the card, and further possible troubleshooting or return authorization.

- The Data Type/Offset/Length combination may be incorrectly set up
  - In the FieldServer configuration, the Offset Parameter almost always refers to the buffer offset in bytes (starting at 0), regardless of data type. The only exception is the BIT data type which specifies offset in bits to allow for offsets that are not a multiple of 8. The Length parameter always refers to length in items, which means Float data will be the number of float values, Word data will be the number of words, etc.

Appendix B.2. Connection Tips & Hints

The SST 5136-PBMS card contains a SYS and a COM LED. The SYS LED indicates the card’s system status as follows:

<table>
<thead>
<tr>
<th>SYS state</th>
<th>LED</th>
<th>Meaning</th>
<th>Status / Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear / OFF</td>
<td></td>
<td>Card is Offline</td>
<td>The DP Master is not online</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td>Card error, network Error or disconnected from network</td>
<td>Check network wiring</td>
</tr>
<tr>
<td>Color</td>
<td>Description</td>
<td>Reason</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td>Card is Online</td>
<td>Card is functioning OK</td>
<td></td>
</tr>
<tr>
<td>Alternating Red / Green</td>
<td>Every time an emulated Slave is in error the LED goes Red.</td>
<td>Network configuration on DP Master may be scanning buffer data not defined in Master. Slave not defined in CSV file.</td>
<td></td>
</tr>
<tr>
<td>Yellow</td>
<td>Some of the emulated Slaves are scanned in stop mode</td>
<td>Card is functioning OK</td>
<td></td>
</tr>
</tbody>
</table>

The COM LED is Green when the card transmits on the network. A Red COM LED indicates a reception error occurred. Make sure both ends of the network is terminated. Also make sure your choice and length of network cable supports the baud rate used. Please refer to the 5136-PBMS Hardware Guide on the Tools CD for more information on network cable types.