Driver Manual

FS-8700-115 ASCII Driver (General Purpose)

APPLICABILITY & EFFECTIVITY
Effective for all systems manufactured after February 2021.

Driver Revision: 1.03
Document Revision: 4.A

MSA Safety
1991 Tarob Court
Milpitas, CA 95035
Website: www.MSAsafety.com

U.S. Support Information:
+1 408 964-4443
+1 800 727-4377
Email: smc-support@msasafety.com

EMEA Support Information:
+31 33 808 0590
Email: smc-support.emea@msasafety.com

Contents

1 Description

The serial GPA (General Purpose ASCII) Driver allows the FieldServer to accept data from remote devices which produce an ASCII byte stream. A typical example of such a device is an electronic scale producing an output similar to the one below.

:weight 0.57 Kg Tare 44.3 Kg 1 2 3 4 -5 -6.7

The driver waits passively for messages. When a message is received, the driver will extract the numbers from a string of characters and numbers. The numbers so formed are stored in consecutive elements of a Data Array. Referring to the example above, the driver will store the value .57 in the 1st element of the Data Array (DA), the value 44.3 in the next element, the value 1 in the next, the value 2 in the next etc.

The driver is also capable of sending custom poll message to a remote device. Some devices may require a character or stream of characters sent to it before it will output its data on a serial port.

The driver can process negative numbers.

1.1 Driver Limitations

- Only one data stream per connection – if two different streams of string data are sent to the same port, the data from the one will overwrite data from the other.
- The driver can only process numbers that are presented in a simple numeric form. Hexadecimal, Exponent-mantissa and other complex forms cannot be processed.
- The driver will overwrite the existing values with the new values. Values will be appended only if the new message has more values than the previous message, e.g. if a message with 5 values follows a message with 3 values, the first 3 values will be overwritten, and the last two values will be appended.

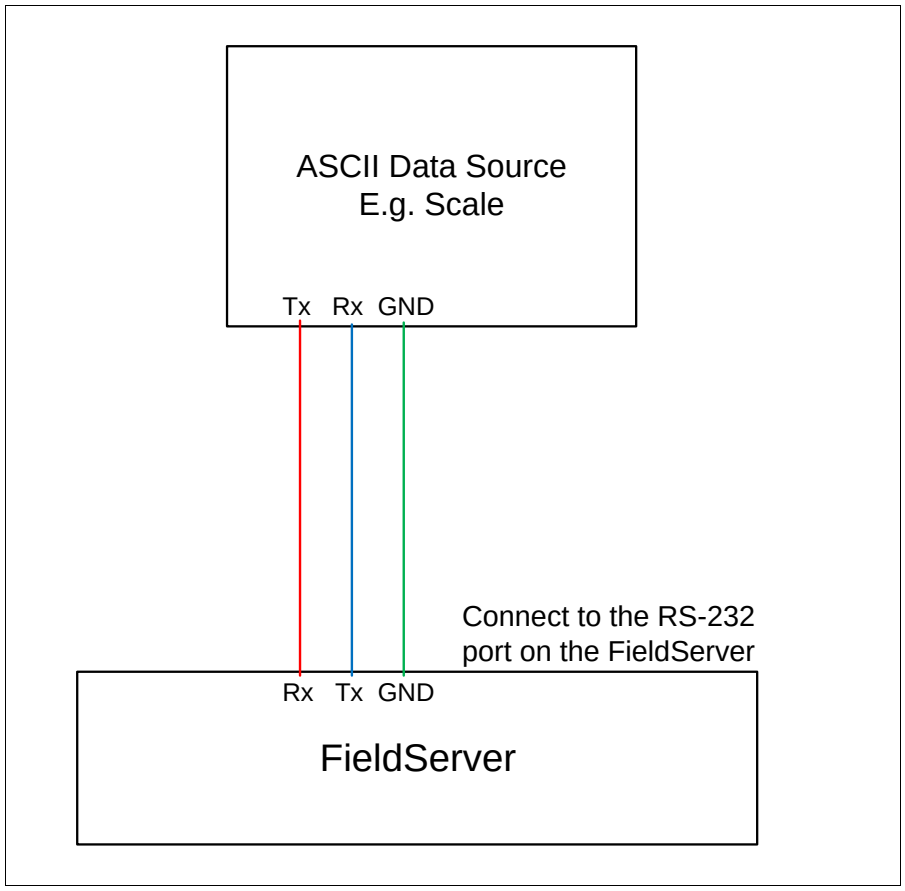
Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 node per port
Server		The driver cannot serve data

2 Hardware Connections

The FieldServer is connected to the vendor device as shown in connection drawing.

Configure the ASCII Passive Client according to manufacturer's instructions.



3 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, Bit, Byte, Uint16, Uint32, Sint16, Sint32
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.	1-10000

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 Client Side Configuration

For detailed information on FieldServer configuration, 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 GPA (General Purpose ASCII) Driver Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for GPA (General Purpose ASCII) 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: In the tables below, * indicates an optional parameter, with the bold legal value being the default.

4.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[1]
Baud*	Specify baud rate.	110 – 115200, standard baud rates only, **9600**
Parity*	Specify parity.	Even, Odd, **None**, Mark, Space
Data_Bits*	Specify data bits.	7, **8**
Stop_Bits*	Specify stop bits.	**1**
Start_Char*	Specify the decimal value of the character with which message will start. The driver will wait for the specified character to detect a valid start of an incoming message. Refer to **Section 6.1.1**.	0-255, **-**
EndPart_String*	Specify the last part of string with which the incoming message will end. This parameter can also be used in conjunction with Termination_Char. Refer to **Section 6.1.1**.	Any case in-sensitive string up to 99 characters. e.g. The End**, -**
Termination_Char*	Specify the decimal value of the character with which the message will end. The driver will wait for specified character to detect a valid end of an incoming message. If used in conjunction with EndPart_String, the Driver will wait for the Termination_Char only if it has received EndPart. Refer to **Section 6.1.1**.	0-255, **Default is 13 (CR – Carriage Return)**

[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.

IC_Timeout*	If this parameter is set, the driver will consider a message complete when it receives the Termination_Char message or the IC_Timeout period has elapsed. If data is received after the IC_Timeout period has elapsed, but before the next Start_Char message, the driver will report an IC_Timeout error, and all characters will be ignored until the next Start_Char message. Refer to **Section 6.1.1**.	0 to 65.5s, **0.5s**

Example

```
//    Client Side Connections
Connections
Port    , Protocol    , Baud    , Parity
P1      , ASCII       , 9600    , None
```

4.2 Client Side Node Parameters

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters
Node_ID	This parameter has no special meaning for this driver. Incoming messages could come from any Node. A Node_ID may be allocated.	Any value may be used
Protocol	Specify protocol used.	ASCII
Connection	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[2]

Example

```
//    Client Side Nodes

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

[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.

4.3 Client Side Map Descriptor Parameters

4.3.1 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 3**
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length -1) as specified in **Section 3**
Function	Function of Client Map Descriptor. Refer to the FieldServer configuration manual for more information. Note that in the case of a Passive Client Map Descriptor the Map Descriptor owns the Data Array elements and no active Map Descriptor can address the same data.	Passive, Rdbc

4.3.2 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 4.2**
Data_Type	This parameter determines how data is stored. Digital data is stored as 1's or Zero's. Non-Float types truncate the values, thus 0.6 is stored as zero. ASCII stores the actual received ASCII characters.	Float_Reg, Int, Bit, ASCII
Length	Length of Map Descriptor. This parameter specifies the maximum number of string items that can be converted to numbers and stored. Other than 'using up' point count, there is no danger in setting the length to a larger number than required.	1, 2, 3…
Poll_Msgs	This parameter allows the user to set the number of times a polling message should be sent. E.g. If this parameter is set to 3, 3 polls are issued at a frequency determined by the scan interval. No further polls are sent even if the Data Array changes. Setting this parameter to 0 will cause polling to continue indefinitely.	**0** to 2147483647
Scan_Interval	The amount of time between successive poll requests.	0-32000s, **2s**

4.4 Map Descriptor Examples

4.4.1 Monitoring an MXL/XLS Device for Events

In this example, we provide a Map Descriptor to capture data from a scale and store it in a Data Array called 'DA_Scale'. The 1st element converted from the ASCII steam will be store at offset zero in the Data Array. The driver will convert and store floating point numbers because the Data_Type is a floating point type. A maximum of 10 number fields can be processed and stored. The driver only converts and stores the data when the ASCII stream ends. Refer to **Section 6.1** for information on how the driver knows when a stream has ended and it should process and store data.

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Data_Type   , Length
Store_Weights         , DA_Scale          , 0                  , Passive    , Node_A      , Float_Reg   , 10
```

In the above example:

- **Map_Descriptor_Name** – Map Descriptor Names do not need to be unique but unique names may facilitate trouble shooting.
- **Data_Array_Name** – The name of the Data Array where the incoming data should be stored.
- **Data_Array_Offset** – The 1st element of converted data will be stored at this location in the DA. Subsequent data is stored in consecutive locations.
- **Function** – Driver waits passively for incoming data. There is no polling.
- **Node_Name** – Connects this Map Descriptor to a Node Descriptor which in turn points to a connection descriptor.
- **Data_Type** – When the driver processes the stream it should (in this case) treat the data as floating point data. Ensure that the format of the Data is suitable to store a floating point value.
- **Length** – A maximum of 10 value items will be stored.

4.4.2 Polling

In this example, we provide a Map Descriptor to send a poll to a remote device. The Map_Descriptor will send ten characters from the Data_Array DA_Poll. This Data Array could be preloaded in the configuration file or its contents could be changed by other protocols. This Map Descriptor is used to send data only, no data is stored by this Map Descriptor.

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Scan_Interval , Length , Poll_Msgs
MD_Poll             , DA_Poll         , 0                 , Rdbc     , Node_A    , 5s            , 10     , 3
```

In the above example:

- **Map_Descriptor_Name** – Map Descriptor Names do not need to be unique but unique names may facilitate troubleshooting.
- **Data_Array_Name** – The name of the Data Array from where the polling data is extracted.
- **Data_Array_Offset** – The 1st character of polling data will be extracted from this position in the Data_Array. Subsequent data will be extracted from consecutive locations.
- **Function** – The Driver will send this poll continuously at time intervals specified by the scan interval.
- **Scan_Interval** – Connects this Map Descriptor to a Node Descriptor which in turn points to a connection descriptor.
- **Length** – Max. of 10 characters sent to the remote device.
- **Poll_Msgs** – The Driver will send the poll message three times.

5 Server Side Configuration

Driver cannot be used to serve data. Contact the FieldServer sales group if you are interested in this functionality.

6 Useful Features

6.1 Separating Data Streams

The driver processes the ASCII stream and stores the converted data when the stream ends. By default, the driver closes the stream when it receives the CR character (Hex code = 0x0d Decimal code = 13), or when the default IC_Timeout period (5s) elapses.

The default end of stream character can be changed using the registry or connection parameter.

6.1.1 Customizing Data Stream Separation Using Connection Parameters

If the connection parameter Termination_Char is specified, the user can define the character used to indicate the end of the data stream. If the connection parameter EndPart_String is specified, a string can be used to specify the end of the data stream. The Termination_Char and EndPart_String parameters can be used in conjunction. The IC_Timeout period can also be configured by the user.

Example

```
//  Client Side Connections

Connections
Port   , Protocol   , Baud   , Parity   , EndPart_String   , Termination_Char   , IC_Timeout
P1     , ASCII      , 9600   , None     , The End          , 13                 , 2s
```

6.1.2 Changing the Default End of Stream Character Using the Registry

The example below is an extract from registry.ini. The Registry group is for port P1. Similar groups can exist for other ports and a group may have more settings than those shown below. If you wished to change the end of stream character to a NEW LINE character (Hex=0x0a Decimal=10) then change the 13's to 10's in the example below. The default_ value is used by the registry when it receives a 'restore defaults' command.

```
[FieldServer_P1]
Termination_Char = 13
default_Termination_Char = 13
```

6.2 Interaction with the WebServer Driver

This driver is specially configured to watch for registry changes that are initiated from the WebServer driverCreate web pages to change the registry settings.

- Browse to the web page.
- Change a setting.
- When the Webserver sees that the settings have been updated it sets a signal for the ASCII driver to use the new settings.

The html fragment below can be inserted in a Web Page. It will allow a user via a browser to change the connection settings for P1 using a browser. The FieldServer configuration file will not have to be changed. Ensure that registry.ini is installed on the FieldServer.

```
<FST_COMBO Reg_name= "FieldServer_P1:Baud"
List_Items="300;600;1200;2400;4800;9600;19200;38400;115200;">
<FST_TEXT Reg_name = "FieldServer_P1:Data_Bits" >
<FST_TEXT Reg_name = "FieldServer_P1:Stop_Bits" >
<FST_TEXT Reg_name = "FieldServer_P1:Parity" >
```

Similar fields may be added for any parameter in the registry file but note that not all registry settings are used by this driver.

7 Troubleshooting

7.1 Driver Specific Stats

Message	Description
IC Timeout	If the Driver is waiting for EndPart_String to arrive and the inter-character timeout (default 0.5s) elapses before the end of stream character has been received the driver stats it as IC_Timeout, clears the incoming buffer and waits for the beginning of the next stream.
Timeout	If driver is configured to send a poll and is unable to transmit the entire poll message within the Timeout period (default 2s), the Driver will start this as Timeout.

7.2 Driver Error Messages

Error	Description and Corrective Action
ASCII:#1 FYI. Port \"%s\" Connection Parameters Can Be Changed Remotely	You can safely ignore this message. It is designed to make you aware of the capability of the driver.
ASCII:#2 FYI. Port \"%s\" Setting Changed Remotely. New Setting: %s	You can safely ignore this message if the setting reported match your expectations. The driver is reporting that connection settings have been changed. The change resulted from a change to the FieldServer registry settings and this driver has been notified to start working with the new settings.
ASCII:#3 FYI. Port \"%s\" Initial Setting:%s Start Char 0x%2X End Char 0x%02X Termination Time %fs	The driver is reporting initial settings for the port. The message is for your information only and no corrective action is required if it confirms you expectations. If it doesn't review and correct the configuration file or registry.ini file.
ASCII:#4 Extracted max %d values, ignoring others.	The message will get printed out when the driver has extracted the maximum number of supported values from the ASCII message. The driver won't process the message further. If more values are required, contact FieldServer sales.
ASCII : Retiring MD<%s> from polling, All poll messages have been sent	The driver is reporting that all configured polling message have been sent. No further polling message will be sent using the Map Descriptor. The message is for your information only and no corrective action is required if it confirms your expectations. If it doesn't review and correct the configuration file.