



Prominent Aquamatrix MultiFLEX 'CS Telnet' Ethernet Driver FS-8705-42

Chipkin - Enabling Integration

salesgroup1@chipkin.com

Tel: +1 866 383 1657

© 2021 CHIPKIN AUTOMATION SYSTEMS

Driver Version: 1.0
Document Revision: 2

TABLE OF CONTENTS

1 'CS TELNET' FUNCTIONAL SUMMARY3

2 'CS TELNET' DRIVER DESCRIPTION4

3 DRIVER SCOPE OF SUPPLY5

 3.1 SUPPLIED WITH THIS DRIVER 5

4 HARDWARE CONNECTIONS.....6

 4.1 HARDWARE CONNECTIONS..... 6

 4.2 BLOCK DIAGRAM #1 – GENERIC..... 6

5 CONFIGURING THE FIELDSEVER GATEWAY7

 5.1 DATA ARRAYS..... 8

 5.1.1 *Data Arrays – Example*..... 8

 5.2 CLIENT SIDE CONNECTIONS 8

 5.2.1 *Client Side Connection Descriptions – Example*..... 9

 5.3 CLIENT SIDE NODES 9

 5.3.1 *Client Side Nodes – Example* 9

 5.4 CLIENT SIDE MAP DESCRIPTORS 11

 5.4.1 *FieldServer Related Map Descriptor Parameters* 11

 5.4.2 *Driver Related Map Descriptor Parameters* 11

 5.5 EXAMPLES..... 12

 5.5.1 *Map Descriptor Example 1 – Read 'CS' Data*..... 12

 5.5.2 *Configuration Example*..... 13

6 CONFIGURING THE FIELDSEVER AS A 'CS TELNET' SYSTEM SERVER16

7 REVISION HISTORY17

1 'CS Telnet' Functional Summary

Driver connects to one or more Prominent Aquamatrix MultiFLEX controllers by logging in with a password. The driver expects the telnet session to be unencrypted. The driver may be sensitive to the version and implementation of telnet. It was only tested on one device. It is reasonable to expect that all MultiFLEX units have the same or similar versions of telnet.

Once connected the driver issues the CS command to read the state of the I/O in the MultiFLEX. The device responds with data. This data is parsed and each delimited value is stored in sequential items of a Data Array. By mapping another protocol(s) onto this Data Array this data can be exposed using the other protocol(s).

The driver also stores the number of items extracted and stored. This is monitored against the expected number of items (configurable). An alarm can be monitored for a mismatch. Under such conditions one would expect that the firmware in the MultiFlex has changed and hence the data stream has changed. New firmware is not required in this situation but a change to the configuration may be required.

The Scan Interval and other behaviors like timeouts and retries are also configurable.

The Telnet protocol provides a transport layer for end to end communication. The protocol does not define application layer usage. For this reason its almost impossible to provide a general purpose Telnet drive that can handle all situations.

This Driver has been specially developed to process the 'CS' command and its response over a Telnet session. The CS command is defined in "Application Note AN_T001"

http://www.aquatrac.com/literature/controllers/AN_T001.pdf

Only a client version of this driver has been developed. The server side of the protocol has not been developed. The driver therefore cannot be used to simulate a MultiFLEX controller. The client was tested against an operational unit.

It does not take a big effort to customize a driver like this to handle some other command and response.

The driver is capable of being linked with other FieldServer drivers to form regular FieldServer firmware that can be installed on QuickServer and other FieldServer gateways. Other drivers can access the MultiFLEX data and serve using other protocols such as BACnet and Modbus.

2 'CS Telnet' Driver Description

In this document we describe the gateway interface developed to provide a gateway to 'CS Telnet' products allowing for easy integration into Building and Industrial Automation Systems using protocols like BACnet and Modbus.

Gateway

The gateway consists of

- A Physical Device eg. FSB3510-Series or FS-QS-Series
- Firmware
 - Firmware Consists of
 - Protocol Driver for 'CS Telnet'
 - Protocol Driver (other eg Modbus) (More than 1 can be linked)
 - Gateway Engine (connects the 2 protocols. Provides all the gateway functionality.)

Therefore complete documentation of the delivered product consists of

1. Manual for 'CS Telnet' Driver – connections, settings, trouble shooting
2. Manual for Physical Gateway – Install, power, trouble shooting
3. Manual for Gateway Configuration and Software tools – How to configure the gateway
4. Manual for 'other' protocol. Such as Modbus

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.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Active Client	Large NUMBER OF NODE	Practical limitation IS GATEWAY MEMORY AND RESOURCES. A single gateway can talk to lots of multiflex controllers.
Active Server (Simulate a Panel)	0	Not supported or documented.

3 Driver Scope of Supply

3.1 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-8705-42	Driver Manual.

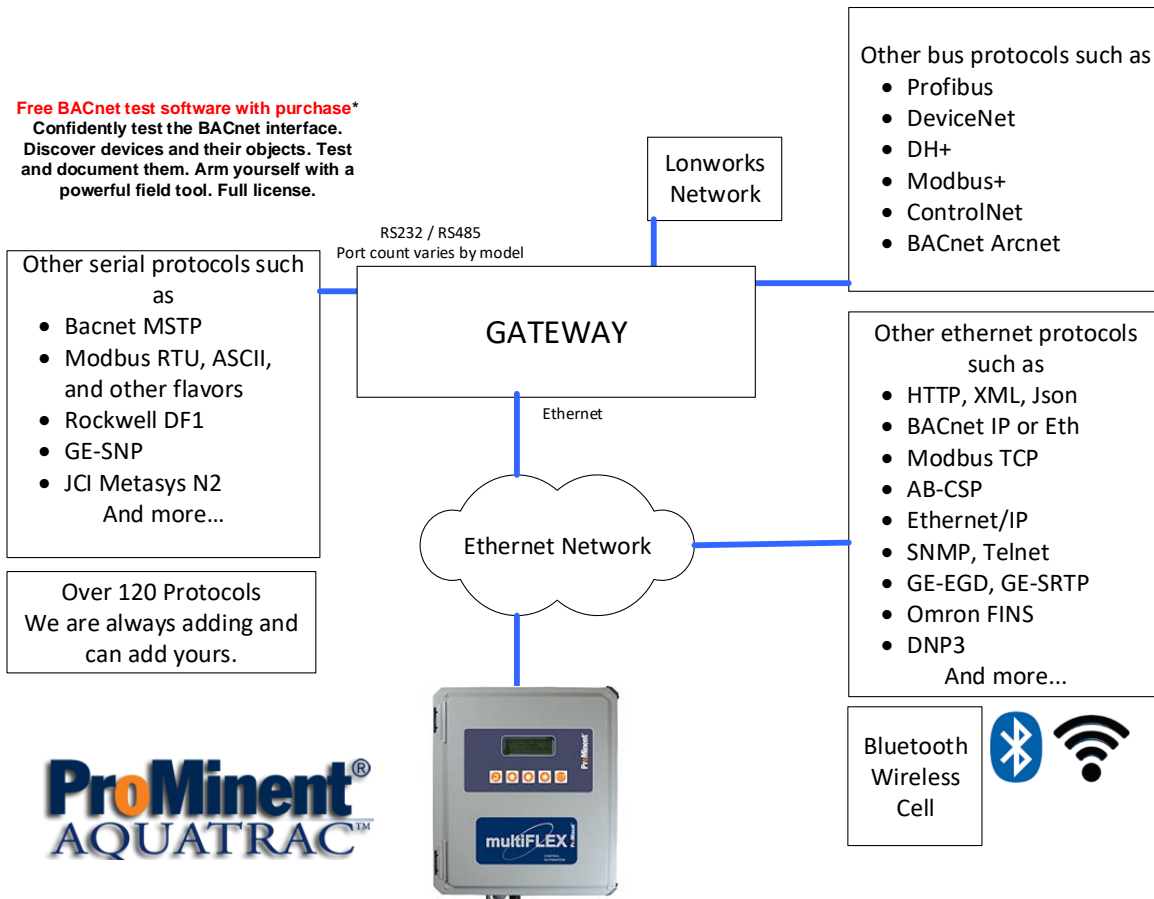
4 Hardware Connections

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

4.1 Hardware Connections

This is an Ethernet Driver. Connections are via standard networking equipment such as hubs, switches, routers and patch cables.

4.2 Block Diagram #1 – Generic



5 Configuring the FieldServer Gateway

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 'CS Telnet' System.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for 'CS Telnet' System 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.

5.1 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: FLOAT, 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

5.1.1 Data Arrays – Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,          Data_Format,          Data_Array_Length,
DA_Data,                  FLOAT,                200
```

5.2 Client Side Connections

Create one connection for each 'CS Telnet' System serial port. Each connection can only be used to connect to a single 'CS Telnet' System interface/port.

Section Title		
Adapters		
Column Title	Function	Legal Values
Adapter	Specify which Ethernet Port is used to connect to the 'CS Telnet' System	N1, N2
Protocol	Specify protocol used	CS telnet
CStelnet_LanPort	TCP port the 'CS Telnet' system has been configured to listen on.	Default = 1001

5.2.1 Client Side Connection Descriptions – Example

```
// Client Side Connections

Adapters

Adapter,      Protocol,

N1,          CS Telnet,
```

5.3 Client Side Nodes

Create one Node per 'CS Telnet' Panel. Create one of these for each CCTV system IP address you want to send messages to.

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID		This commonly used gateway parameter does not have a use in this protocol driver.
Protocol	Specify protocol used	CS telnet
Adapter	Which Adapter you are using at the connection level	N1, N2
IP_Address	The IP Address of the 'CS Telnet' System you want to send a message to.	Eg 192.168.1.168 (normal IP format)

5.3.1 Client Side Nodes – Example

```
// Client Side Nodes

Nodes

Node_Name,      Node_ID,  Protocol,  Adapter  IP_Address
MainPanel,      1,        CS Telnet, N1        192.168.1.168
```


5.4 Client Side Map Descriptors

5.4.1 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..	Rdbc – read continuously Wrbr – act on trigger

5.4.2 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
Address	This commonly used parameter is not used by this driver.	0
Length	Length of Map Descriptor	Set the value to 200. You are reserving space for all the data items.

5.5 Examples

5.5.1 Map Descriptor Example 1 – Read 'CS' Data

This single simple Map Descriptor is used to read the data from the Controller. You will need one of these for each controller you are reading.

```
Map_Descriptors
Map_Descriptor_Name      ,Scan_Interval ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,Length
Read Data                ,0s              ,DA_Data         ,0                ,rdbc    ,Node01     ,200
```

Wrbx means – write on update.
Whenever the data in DA_CONTROL[0] is updated (value might not change) then the action is triggered.

Reserver enough space

5.5.2 Configuration Example

```
//=====
//
//   Common Information
//
Bridge
Title
CStelnet rev100
//=====
//
//   Data Arrays
//
Data_Arrays
Data_Array_Name ,Data_Format ,Data_Array_Length
DA_Data          ,FLOAT          ,2000
DA_CStelnet      ,Uint16         ,2000

// This value is compared to the number of items converted and stored. If they are different then
DA_CStelnet[11]=1

// Connect the point to a server object for a remote clien to monitor for an alarm condition

Preloads,
Data_Array_Name ,Preload_Data_Index ,Preload_Data_Value,Preload_Data_Format
DA_CStelnet     ,9                   ,160           ,UINT16 // Number of Items expected
```

DA_CSTELNET is a special array name. It is used by the driver to control some diagnostic information and to produce a conversion mismatch alarm. It should always be present.

Driver stores the number of converted items in DA_CSTELNET[10] and compares to CSTELNET[9] and report ok/alm in CSTELNET[11] which can be mapped onto the other protocol for monitoring.

```

Connections
Adapter ,protocol ,CStelnet_LanPort
N1      ,cstelnet ,1001

//-----
//
// Client Side Nodes
//

Nodes
Node_Name ,IP_Address ,Node_ID ,Hunter_SiteId ,Protocol ,Adapter
Node01    ,209.92.119.232 ,1      ,0          ,cstelnet ,N1

//-----
//
// Client Side Map Descriptors
//

Map_Descriptors
Map_Descriptor_Name      ,Scan_Interval ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,Length
Read Data                ,0s            ,DA_Data         ,0                  ,rdbc    ,Node01    ,1
    
```

Enable 'CS Telnet' protocol on Ethernet Port N1. Expects the remote controller to be listening on port 1001

The Address of the MultiFLEX controller

This single task reads the CS data. In this case the Scan Interval of zero means 'as often as possible'

```

Connections
Adapter ,Protocol
N1      ,Bacnet_IP

Nodes
Node_Name      ,Node_ID ,Protocol
CS_DEVICE     ,389001 ,Bacnet_IP

Map_Descriptors
Map_Descriptor_Name      ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name      ,Object_Type ,Object_Instance ,Units
ALARM Reconfiguraton Rqd - Unexpected Data ,DA_CS_Telnet    ,11                ,Passive ,CS_DEVICE     ,bI          ,00                ,No_Units
ServerOffset 01          ,DA_Data         ,01                ,Passive ,CS_DEVICE     ,AI          ,01                ,No_Units
ServerOffset 02          ,DA_Data         ,02                ,Passive ,CS_DEVICE     ,AI          ,02                ,No_Units
ServerOffset 03          ,DA_Data         ,03                ,Passive ,CS_DEVICE     ,AI          ,03                ,No_Units
ServerOffset 04          ,DA_Data         ,04                ,Passive ,CS_DEVICE     ,AI          ,04                ,No_Units
ServerOffset 05          ,DA_Data         ,05                ,Passive ,CS_DEVICE     ,AI          ,05                ,No_Units
ServerOffset 06          ,DA_Data         ,06                ,Passive ,CS_DEVICE     ,AI          ,06                ,No_Units
ServerOffset 07          ,DA_Data         ,07                ,Passive ,CS_DEVICE     ,AI          ,07                ,No_Units
ServerOffset 08          ,DA_Data         ,08                ,Passive ,CS_DEVICE     ,AI          ,08                ,No_Units
ServerOffset 09          ,DA_Data         ,09                ,Passive ,CS_DEVICE     ,AI          ,09                ,No_Units,
    
```

Enable BACnet (Could have been Modbus or any other protocol)

BACnet Device. This is the 'virtual' controller. This BACnet device represents the MultuFLEX controller

Each converted data item is stored in sequential positions in the Data Array = DA_DATA (in this example). Each of these positions can be mapped onto BACnet. In this example the 1st 10 data values have been mapped. They could have been allocated meaningful names and Engineering units.

6 Configuring the FieldServer as a 'CS Telnet' System Server

This driver cannot be used to emulate a MultiFLEX Controller. If you really need this feature contact our sales dept.

7 Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
Oct 2019	PMC		1.0	1	Created
11 May 2021	PMC		1.0	2	Document format updated