Chipkin Automation Systems

Driver Manual

(Supplement to the FieldServer Instruction Manual)

FS-8705-18

Multistack Comput25 Master Controller Serial Driver

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after Sep 20, 2010

, 3495 Cambie St- Box 211, Vancouver, BC, Canada, V5Z 4R3

Tel: (866) 383-1657, **Fax:** (416) 915-4024

Email: dfs@chipkin.com **Website:** www.chipkin.com

TABLE OF CONTENTS

1. Multistack Comput25 Driver Description

The Compu25 protocol can be used to connect to suitably enabled legacy Multistack Inc, Chiller and Heat Pumps. This is a serial protocol using RS485. Multiple Field devices can be connected on a single trunk. The gateway supports one trunk.

The Gateway connects to the HVAC devices, reads data and stores it internally. When a remote system requests data, this data is served in a form that is appropriate to the requesting protocol. In the event that the connection to the HVAC controller is lost, or data cannot be read, the gateway can signal this to the remote data client.

The gateway requires minimal configuration and can be considered a plug and play component of a system, in that it is ready to operate out of the box with the default configuration.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Passive Client	1	Only one panel can be connected to a single FieldServer serial port.
Active Server (Simulate a Panel)	0	Not supported or documented.

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
-	No specific cables are shipped with this driver. A generic RJ45 Ethernet cable must be shipped with this driver.
-	A generic male and Female connector kit must be shipped with this driver.
FS-8705-18	Driver Manual.

2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Hardware

Part #	Description

2.2.2. Required 3rd Party Software

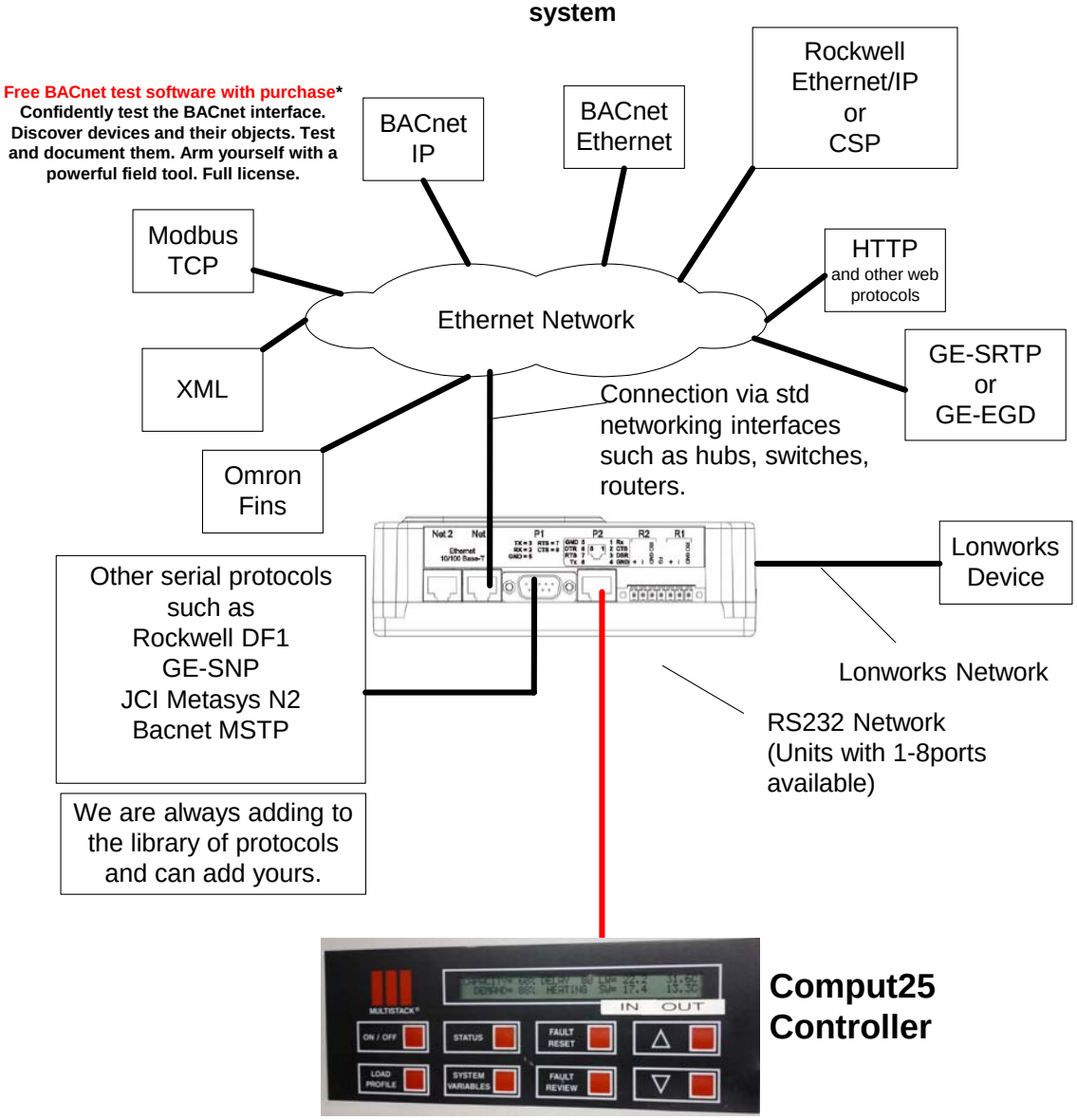
2.2.3. Required 3rd Party Configuration

None Known.

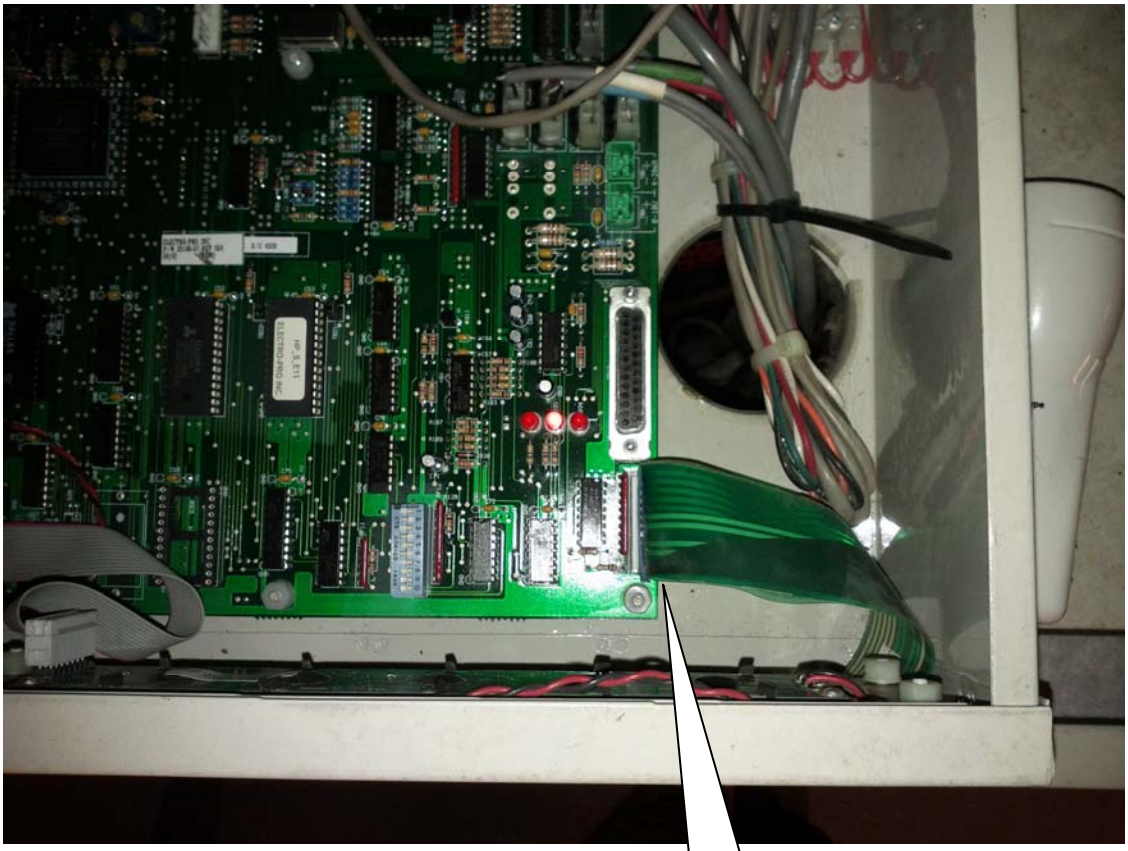
3. Hardware Connections

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

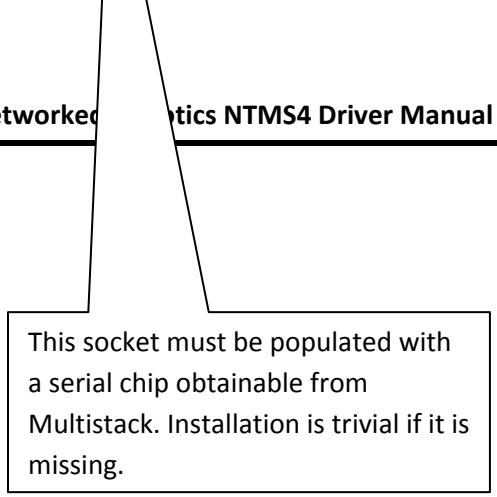
Monitor and Control DSC Security Panels with SIA codes using BACnet, Lonworks or a PLC system



3.1. Hardware Connection Tips / Hints



Gateway connects to this port.



4. Configuring the FieldServer as a Comput25 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 Comput25 controller.

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 FSC - Electronic Siren Controllers Serial 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 that in the tables, * indicates an optional parameter, with the bold legal value being the default.

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, UInt16, 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

Example

```
//   Data Arrays
Data_Arrays
Data_Array_Name,          Data_Format,          Data_Array_Length,

multistackstats,          UNT16,                200
```

4.2. Client Side Connection Descriptions

Create one connection for each PRO2000 port. Each connection can only be used to connect to a single PRO2000 interface/port.

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[1]
Protocol	Specify protocol used	multistack
Baud*	Specify baud rate	Driver Supports : 110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 28800; 38400; 57600 Baud Comput25 devices only support 2400 Baud.
Parity*	Specify parity	Driver Supports : 7,**8** Comput25 devices only support 8
Data_Bits*	Specify data bits	Driver Supports : **1**,2 Comput25 devices only support 1

[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.

Stop_Bits*	Specify stop bits	Driver Supports : Odd, Even, **None** Comput25 devices only support None.
Handshaking*	Specify hardware handshaking	**None**
Poll _Delay*	This parameter is required.	**0.05**

Example

```
//        Client Side Connections


Connections
Port,           Baud,           Parity,     Protocol,
P1,             2400,           None,       multistack,
```

4.3. Client Side Node Descriptors

Create one Node per connection only.

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
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.	1-258
Protocol	Specify protocol used	multistack
Connection	Specify which port the device is connected to the FieldServer The comput25 devices only support RS232 therefore only P ports may be used unless you have a converter.	P1-P8, R1-R2[2]

[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.

Example

```
//   Client Side Nodes


Nodes
Node_Name,          Node_ID,    Protocol,       Connection
HeatPump,           1,          multistack ,    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 The temperature is stored here.	One of the Data Array names from “Data Array” section above We recommend that you use a ‘FLOAT’ Data Array since the temperatures are reported as floating point numbers.
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in “Data Array” section above
Function	Function of Client Map Descriptor..	RDBC, ARB, WRBX, WRBC

4.4.2.

4.4.3. 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 to 1.	1
Address	This commonly used FieldServer parameter is not used by this protocol.	
MULT_Address	Speicified in hexadecimal. This is the Comput25 Controller memory location to be read.	**Valid addresses are 4 hex chars. Eg 0093**
MULT_Read_Word	Tells the driver to read two consecutive addresses and to combine the 2 bytes of data returned into one 16 bit value.	**No**, Yes

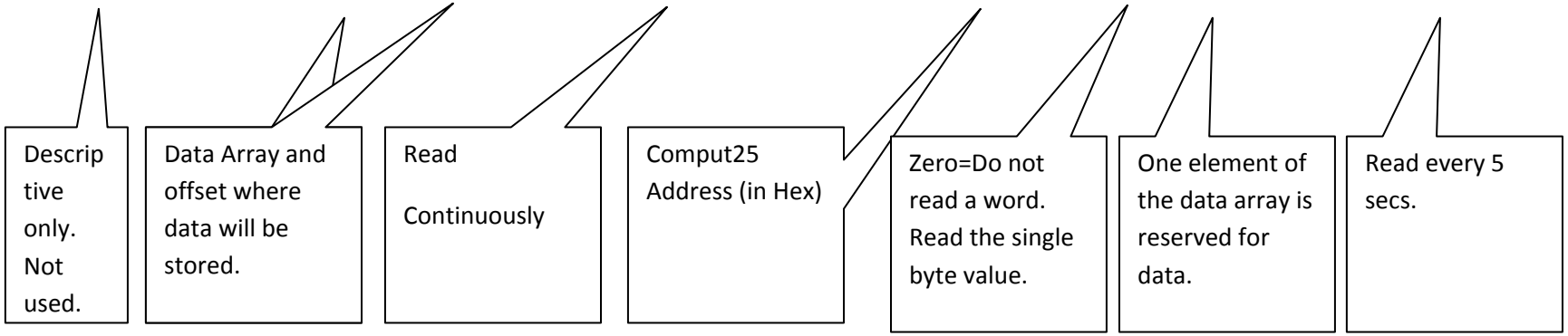
4.4.4. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥0.001s

4.4.5. Map Descriptor Example 1 – Read Compressor Status

In this example the driver reads the status. It is a 8 bit value. Each bit has an allocated meaning. This example does not break out the bits. This task is executed at best every 5 seconds.

```
Map_Descriptors
Map_Descriptor_Name ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name    , MULT_Address ,MULT_Read_Word ,Length ,Scan_Interval ,
Compressor 01 Status,DA_Data         ,0                 ,RDBC     ,HeatPumpCtrl , 2068         ,0              ,1      ,5.000s       ,
```



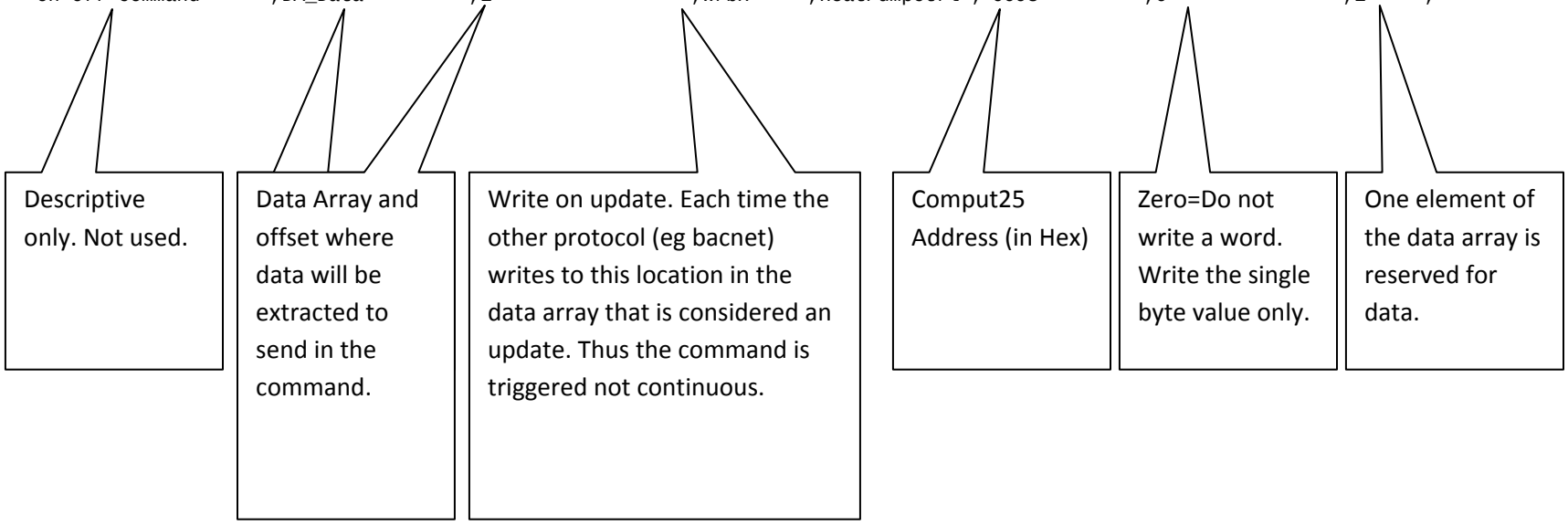
4.4.6. Map Descriptor Example 2 – Command Run/Stop

In this example the driver writes the unit on / off. This only occurs when the source data is updated – ie when the other protocol is used to write a value to the DA[offset] location. Note Command On with value=63. Command off with value=31. An alternate solution is provided in example 3

```
Map_Descriptors

Map_Descriptor_Name ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name    , MULT_Address ,MULT_Read_Word ,Length ,

On Off Command      ,DA_Data         ,1                 ,wrbx     ,HeatPumpCtrl , 0093         ,0              ,1     ,
```



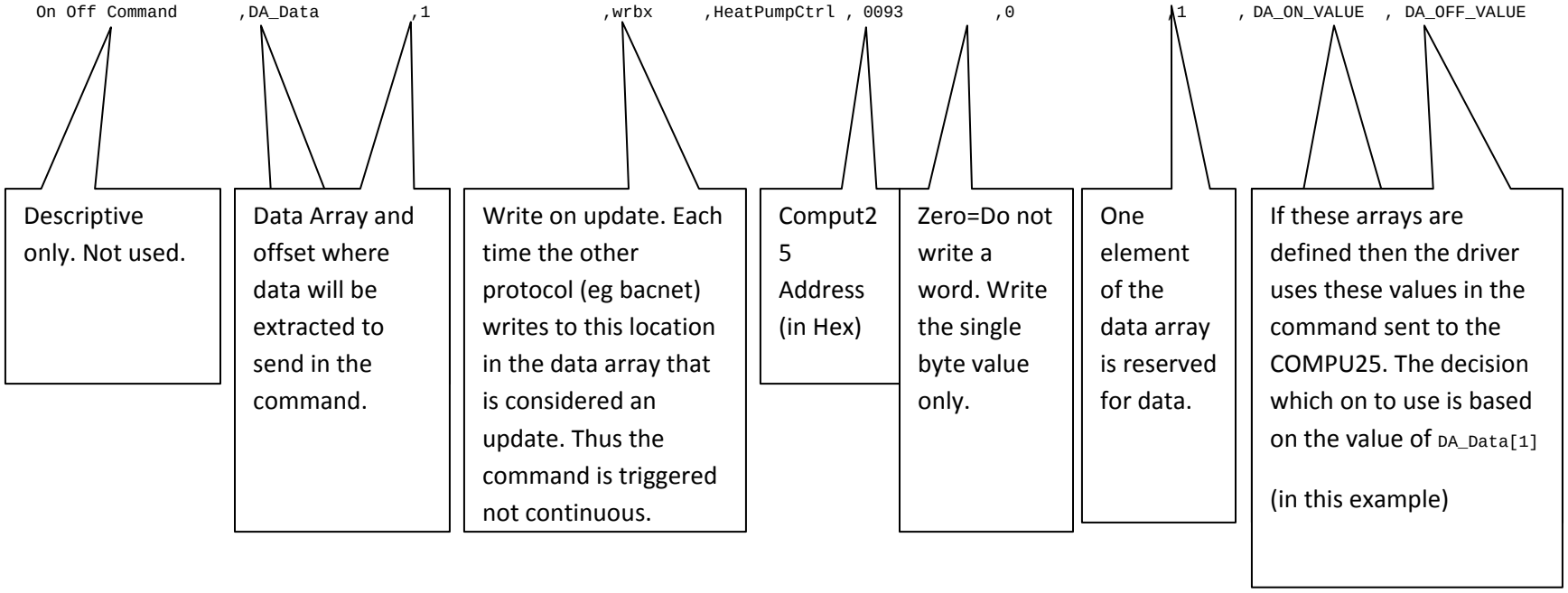
4.4.7. Map Descriptor Example 3 – Command Run/Stop - Alternate

While your upstream protocol (eg BACnet would like to send a 1 or zero to turn the unit on or off the actual value that must be sent to the COMPU25 controller may be different. Typical values for Off=31 and On=63. This is achieved as shown in this example. The driver looks in the normal data array, extracts a value. If zero then it looks in the corresponding position of the array defined by '`DA_Byte_Name'`. If non-zero then it looks in the corresponding position of the array defined by '`DA_Float_Name'`. Some value must be preloaded (See next page)

```
Map_Descriptors

Map_Descriptor_Name ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name    , MULT_Address ,MULT_Read_Word ,Length , DA_Byte_Name , DA_FLoat_Name

On Off Command      ,DA_Data         ,1                ,wrbx     ,HeatPumpCtrl , 0093         ,0             ,1      , DA_ON_VALUE  , DA_OFF_VALUE
```



Data Arrays are pre-loaded like this. Place this section immediately after the Data Arrays section of the configuration file.

```
Preloads
Data_Array_Name ,Preload_Data_Index ,Preload_Data_Value ,
DA_ON_VALUE     ,00                 ,31                 ,
DA_OFF_VALUE    ,00                 ,63                 ,
```

5. Configuring the FieldServer as a Multistack Comput25 Server

This driver cannot be used to emulate an comput25 device.

Appendix 1. Advanced Topics

Appendix 1.1. Driver Error Messages

Error Message We have shown place holders for the parts of the message which change. %s is a place holder for a text string. %d is a place holder for a number %c is a place holder for an alpha character.	Explanation and corrective action ***FYI messages are informational and do not require a corrective action. Simply use them to confirm configuration / behaviors are what you expect.***
MULT:#01 FYI. Md=%s Rqsted=%d Sent=%d MaxVal=255	Only a single byte can be written at a time. The max value that fits in a byte = 255. Its possible that the way the gateway is configured may result in the other protocol (eg BACnet) may send a value greater than 255. In such cases the driver truncates the data and prints this message to make you aware of the fact.
MULT:#03 FYI. multi-byte read. Order=abcd	Its possible to configure the driver to read two consecutive locations and to combine the two single bytes into 1 16 bit value

	before storing it in the Data Array. This message is printed once to tell you the order in which the bytes are combined. Location A = ab (hex) Location B = cd (hex) Stored Value = abcd (hex)
MULT:#04 FYI. multi-byte read. Order=cdab	Its possible to configure the driver to read two consecutive locations and to combine the two single bytes into 1 16 bit value before storing it in the Data Array. This message is printed once to tell you the order in which the bytes are combined. Location A = ab (hex) Location B = cd (hex) Stored Value = cdab (hex)
MULT:#20 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	This message should not be printed during normal operation. It should only be printed during factory testing under specific conditions. If you see this message capture a log and report the fact to Tech Support
MULT:#21 FYI. Write Responses suppressed.	This message should not be printed during normal operation. It should only be printed during factory testing under specific conditions. If you see this message capture a log and report the fact to Tech Support
MULT:#9e Err. Diagnostic MULT:#9f Err. Diagnostic	This message should not be printed during normal operation. It should only be printed during factory testing under specific conditions. If you see this message capture a log and report the fact to Tech Support

Appendix 1.2. Driver Statistics

Appendix 1.3. Exposing Driver Stats

The diver 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,
multistackstats,        UINT32,             1000,
```

Offset	Description
1	Increments each time a Read message is sent
2	Counts the total number of bytes sent in read messages.
3	Increments each time a Write message is sent
4	Counts the total number of bytes sent in Write messages.
5	Increments each time a message cannot be sent for some reason.
6	If the buffer used to stored responses from the comput25 overflows this value increments. It should remain zero. If it increments often, capture a log and contact Tech Support.
7	Increments each time a response is received that for one reason or another is invalid in its format or length and cannot be processed correctly. Such responses are dropped.

Offset	Description
8	Increments each time a response is received whose format allows it to be processed.
9	Increments each time a NAK response is received
10	The exception code received with the most recent NAK is stored here.
11	Increments each time a response is received whose format allows it to be processed and which was processed without error.
12	Increments each time a response is received whose format allows it to be processed but which failed the processing.
13	Increments each time no response is received within the timeout period.
14	Increments each time a response is received which does not begin with the designated begin of message character. Such messages are dropped.
15	Increments each time a response is received which has a checksum error. Such messages are dropped.
16	Increments each time a response is received which does not contain the correct message termination character. Such messages are dropped.
17	Increments each time a response is received whose byte value cannot be extracted.
18	Increments each time a response is received which does not contain the correct message termination character. Such messages are dropped. Context is different to #16
19	Increments each time a response is received whose byte value cannot be extracted. Context is different to #17
20	Increments each time a message was processed ok and resulted in data being stored.
21	The most recent value read is stored here.
22	Set this value to 1 to have the server side of the driver suppresses responses to polls. This is required for testing only.

Offset	Description
23	Set this value to 1 to have the client side of the driver to not expect responses – it times out and continues to work ok.
24	Increments by 1 each time we return Md with OK on timeout from a write. Should only occur if MULT_STAT_NO_WRITE_RESPONSE is non-zero

Appendix 1.4. Memory Map

Typically you are provided a pre-configured device and a document accompanying the device provides the map of BACnet or Modbus objects. The list of comput25 memory locations is, thus, not required. It is available on request.

Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
2012Jan19	PMC		1.0	1.0	Created.
2012Apr13	PMC		1.5	2.0	New Feature for writing values. See example #3.