



# HACH Ultra Met One R4800 Counter Serial Driver FS-8705-45

Chipkin - Enabling Integration

[salesgroup1@chipkin.com](mailto:salesgroup1@chipkin.com)

Tel: +1 866 383 1657

© 2021 CHIPKIN AUTOMATION SYSTEMS

Driver Version: 8.0  
Document Revision: 3

**TABLE OF CONTENTS**

**1 HACH ULTRA MET ONE R4800 PARTICLE COUNTER DRIVER DESCRIPTION .....3**

**2 DRIVER SCOPE OF SUPPLY .....4**

    2.1 SUPPLIED WITH THIS DRIVER ..... 4

**3 HARDWARE CONNECTIONS.....5**

    3.1 BLOCK DIAGRAM..... 5

    3.2 CABLE CONNECTIONS..... 6

    3.3 TERMINATIONS ..... 7

**4 CONFIGURING THE FIELDSEVER AS A R4800 COUNTER CLIENT.....8**

    4.1 DATA ARRAYS..... 9

        4.1.1 *Data Arrays – Example*..... 9

    4.2 CLIENT SIDE CONNECTIONS ..... 10

        4.2.1 *Client Side Connection Descriptions – Example*..... 10

    4.3 CLIENT SIDE NODES ..... 11

        4.3.1 *Client Side Nodes – Example* ..... 11

    4.4 CLIENT SIDE MAP DESCRIPTORS ..... 12

        4.4.1 *FieldServer Related Map Descriptor Parameters* ..... 12

    4.5 DRIVER RELATED MAP DESCRIPTOR PARAMETERS ..... 12

    4.6 EXAMPLES..... 14

        4.6.1 *Map Descriptor Example 1 – Read Log Record* ..... 14

        4.6.2 *Map Descriptor Example 2 – Stop Counting*..... 15

**5 CONFIGURING THE FIELDSEVER AS TO EMULATE A COUNTER.....16**

**6 REVISION HISTORY .....17**

**APPENDIX A. MAP DESCRIPTOR LENGTHS .....18**

**APPENDIX B. HOW DATA IS STORED.....19**

**APPENDIX C. SUPPORTED SERVICES .....22**

**APPENDIX D. DRIVER ERROR MESSAGES.....23**

## 1 HACH Ultra Met One R4800 Particle Counter Driver Description

This serial driver connects to a trunk of 1 or more R4800 Counters that support the R4800 protocol as defined in the R4800 Operator Manual. It is capable of reading log records and status data from each transmitter as well as being able to send commands to the counter.

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 R4800 Counter's data and serve using other protocols such as BACnet and Modbus.

The driver is an active client driver in that it poll's for data – reading or writing data from counters. It cannot be used to simulate a R4800. Because only the client side of the protocol is implemented.



### Max Nodes Supported

FIELDSEVER MODE	NODES	COMMENTS
Client	Many	Number of counters is determined by vendor and RS485 limitations. Trunks of up to 10 devices supported.
Server	0	Not supported or documented.

## 2 Driver Scope of Supply

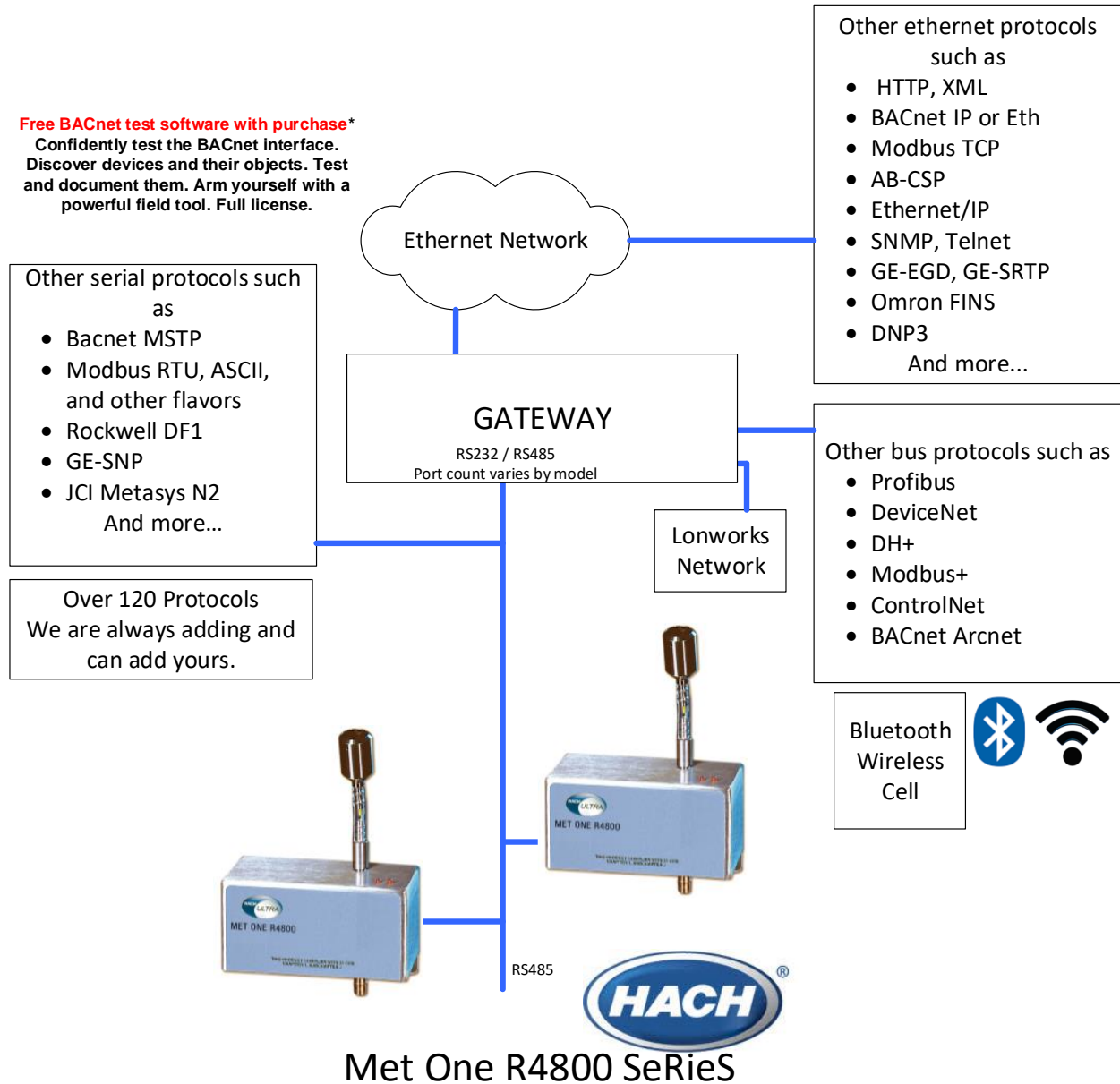
### 2.1 Supplied with this driver

FIELDSEVER TECHNOLOGIES PART #	DESCRIPTION
Cables	No specific cables are shipped with this driver.
FS-8705-45	Driver Manual.

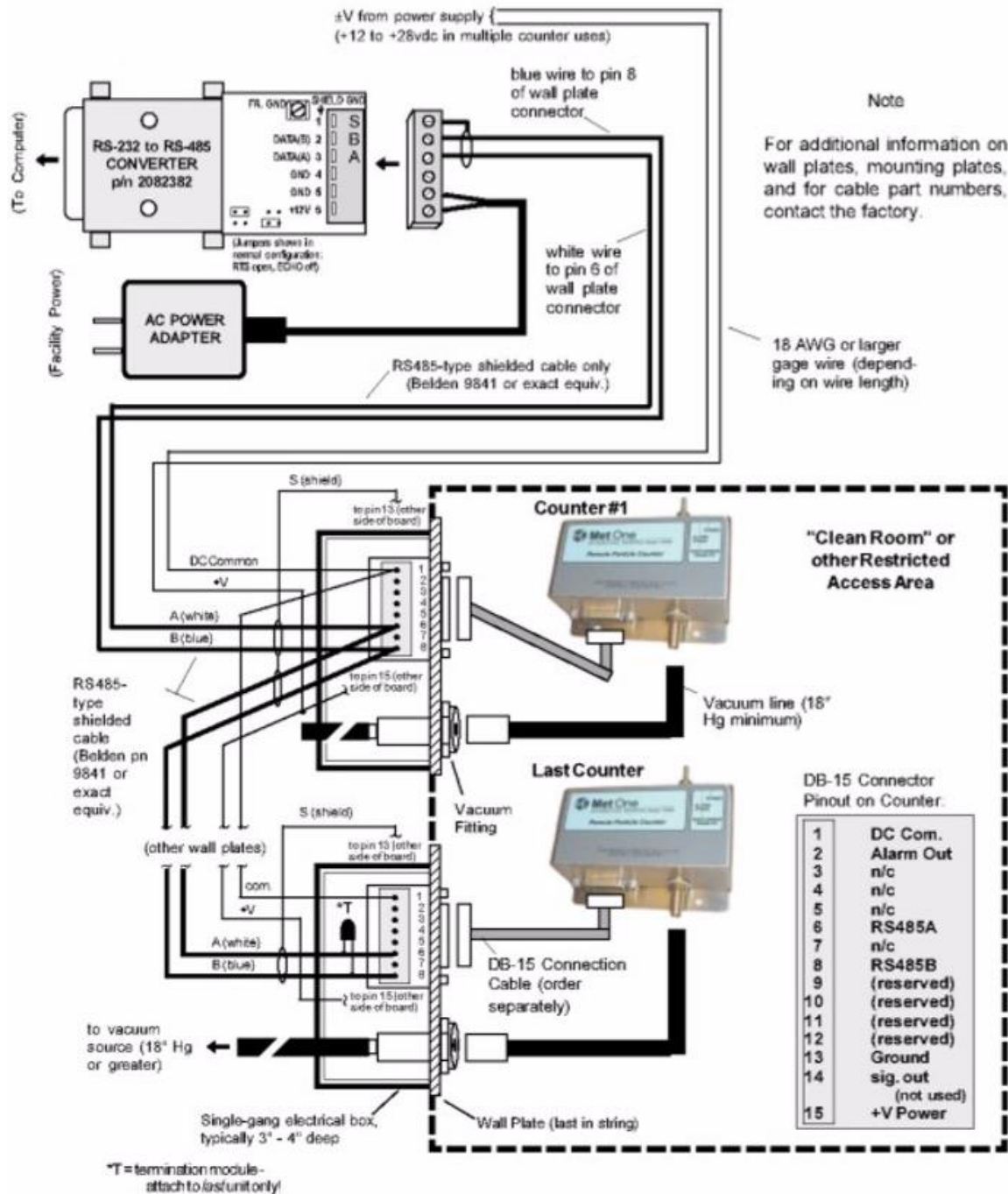
### 3 Hardware Connections

#### 3.1 Block Diagram

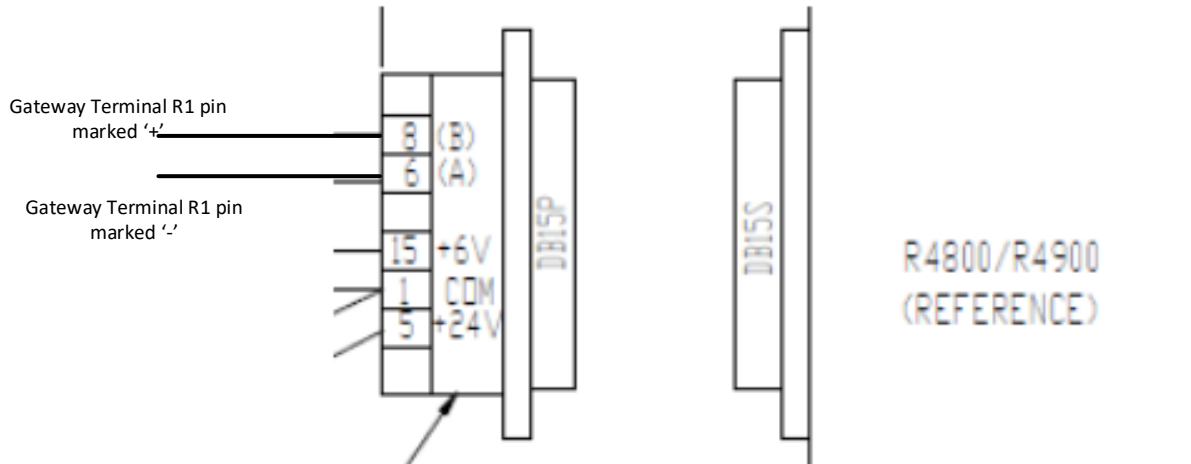
Multiple WorkStation protocols and connection supported. See list of FieldServer Drivers.



3.2 Cable Connections



3.3 Terminations



## 4 Configuring the FieldServer as a R4800 Counter 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 Hach Ultra Met One R4800 counters.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for R4800 monitoring and control, the driver’s 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

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.	<b>Recommended:</b> 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

4.1.1 Data Arrays – Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,          Data_Format,          Data_Array_Length,
metoneR4800 ,             UNT16,                200
DA_DATA,                  FLOAT,                200
```

## 4.2 Client Side Connections

Create one connection for each trunk of R4800 Counters.

SECTION TITLE		
Connections		
COLUMN TITLE	FUNCTION	LEGAL VALUES
Port	Specify which port the device is connected to the FieldServer	R1-R2
Protocol	Specify protocol used	metoneR4800
Baud*	Specify baud rate	Driver Supports: 110; 300; 600; 1200; 2400; 4800; <b>9600</b> ; 19200; 28800; 38400; 57600 Baud Counter supports 1200, <b>9600</b>
Data_Bits *	Specify parity	Driver Supports: 7, <b>8</b> Counter supports: 8
Stop_Bits*	Specify data bits	Driver Supports: <b>1,2</b> Counter supports: 1
Parity *	Specify stop bits	Driver Supports: Odd, Even, <b>None</b> Counter supports: None

### 4.2.1 Client Side Connection Descriptions – Example

```
// Client Side Connections

Connections

Port,      Baud   Parity,   Data_Bits,  Stop_Bits,  Protocol
R1,        9600   None,     8,          1,          metoneR4800
```

### 4.3 Client Side Nodes

Create one Node per FACP in the network only.

SECTION TITLE		
Nodes		
COLUMN TITLE	FUNCTION	LEGAL VALUES
Node_Name	Provide name for node	Up 12 of the max of 32 alphanumeric characters possible to specify the Node name.
Node_ID	Not used directly by the driver	128-191 Where 128 is also known as Location 0 Do not use the location number.
Protocol	Specify protocol used	metoneR4800

#### 4.3.1 Client Side Nodes – Example

```

// Client Side Nodes

Nodes
Node_Name,      Node_ID,      Protocol,      Connection
Node_a,         128,          metoneR4800,  R1
    
```

**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	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,wrbc,wrbcx

**4.5 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.	See Table below this one.
Address	This commonly used parameter is not used by this driver.	
metoneR4800CMD	Specify the command you wish to use. Use one of the strings indicated in the next column	See Table below this one.
metoneR4800CRLF	Do not change this parameter in the configuration file. I	

<u>metoneR4800CMD</u>	<u>Length</u>
Use these strings in the configuration file to execute the different services	Reserve space in the Data Arrays
send buffered record	15
send current record	15
Clear buffer	2
number of records	3
Hold time	5
Sample period	5
Mode request	3
Device select It is not required that you implement this service because the driver does counter selection automatically. Each time it is about to send a command to the counter, it checks to see if the command is being sent to the same counter as the previous one. If this is not the case, the driver automatically does counter selection before executing the configured commands.	2
Auto	2
Manual	2
Start counting by computer	2
Start counting by counter	2
Stop counting	2
Active mode	2
Standby mode	2
Universal Select	2

## 4.6 Examples

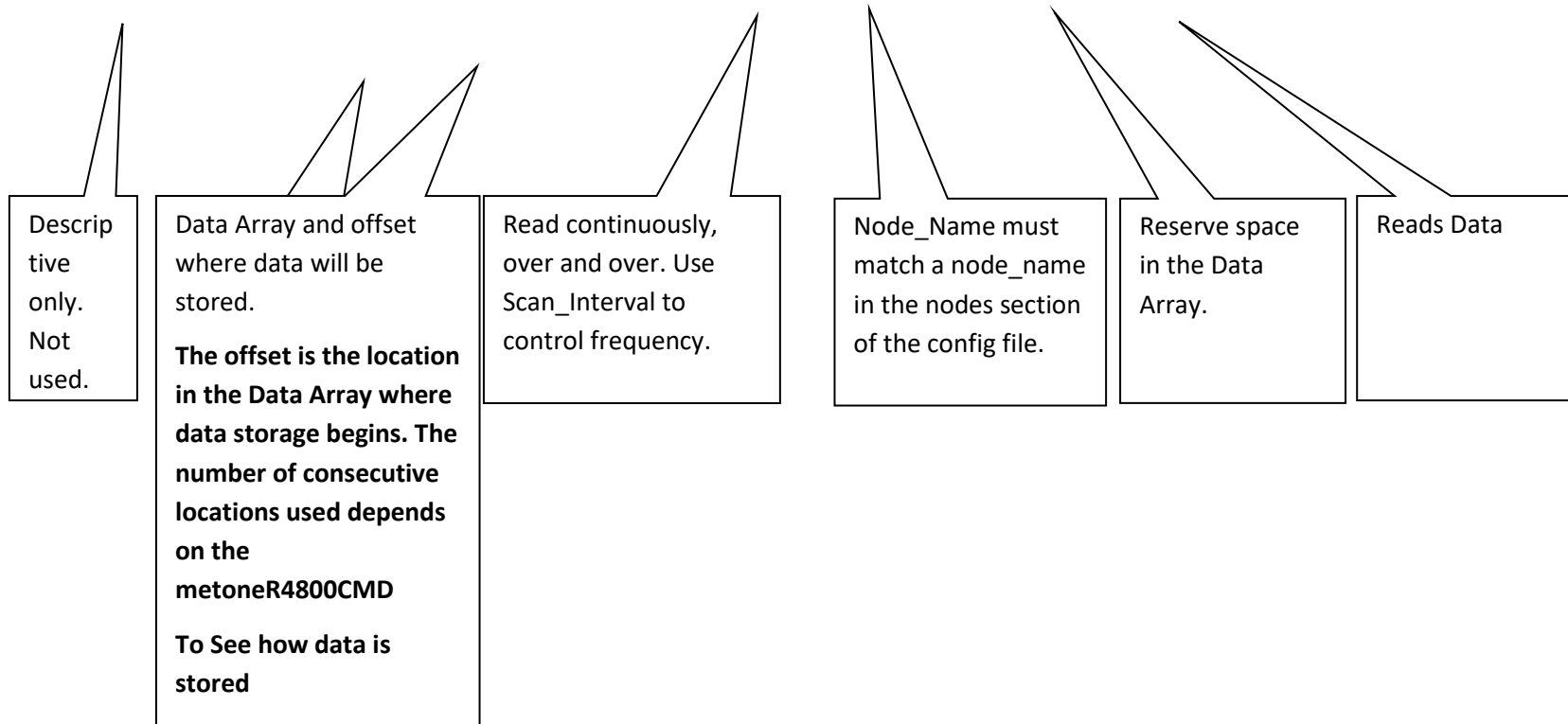
### 4.6.1 Map Descriptor Example 1 – Read Log Record

In this example the current count record will be retrieved. If there is no record to retrieve then the DA\_Data[1] will be set to zero, else to 1. 14 data values are extracted and stored in consecutive locations in the Data Array. An appendix contains a listing of what data you will find at each offset.

```
Map_Descriptors
```

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Length , metoneR4800CMD
```

```
Send Current Record , DA_Data , 0 , Rdbc , Counter128 , 15 , Send Current Record
```



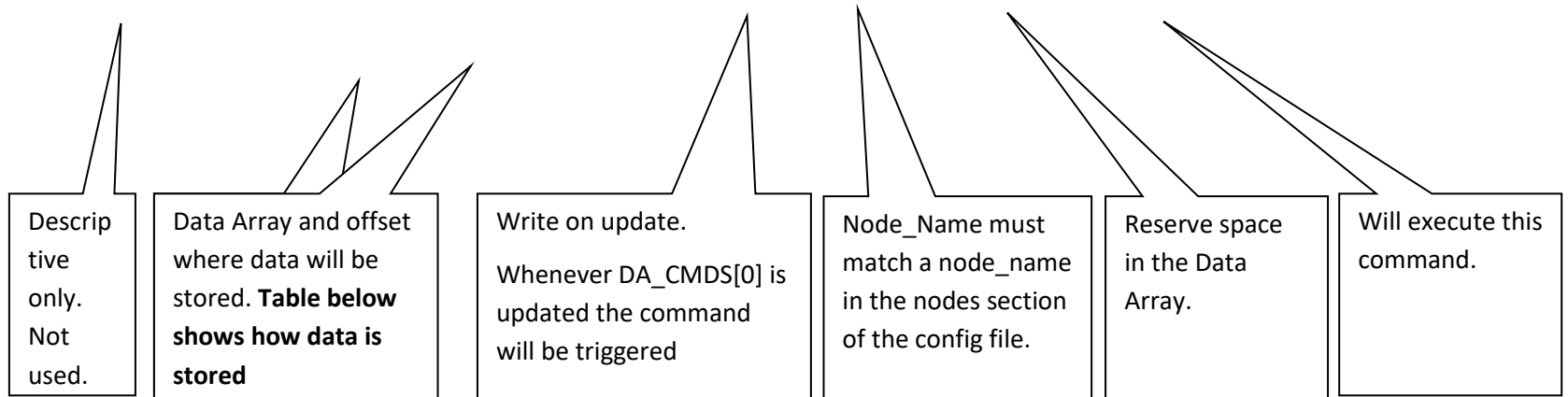
### 4.6.2 Map Descriptor Example 2 – Stop Counting

Sends the Stop Counting command. In this example when the value in DA\_CMDS[0] is updated (does not have to change) then the command will be triggered. On completion the driver will store an 'e' in DA\_CMDS[0]. The driver will count how many times this command has executed. The count will be stored in DA\_CMDS[1]

Map\_Descriptors

Map\_Descriptor\_Name , Data\_Array\_Name , Data\_Array\_Offset , Function , Node\_Name , Length , Vasia;a\_CMD

Stop Counting , DA\_CMDS , 0 , wrbx , Counter128 ,2 ,Stop Counting



## 5 Configuring the FieldServer as to Emulate a Counter

This driver cannot be used to emulate a R4800 Counter. For some protocols we implement the client and server sides – like Modbus. In such cases the protocol can be used to emulate a device. We do not normally do this for protocols where we expect our customer will always want the Client functionality. Ask our sales department if you need to emulate a device.



## 6 Revision History

DATE	RESP	FORMAT	DRIVER VER.	DOC. REV.	COMMENT
Dec 2019	PMC		0.00	0	Created.
28 Jan 2020	PMC		1.00	1	Updated connections
1 Feb 2021	PMC		8.00	2	Added timestamp
10 May 2021	PMC		8.00	3	Document format updated

## Appendix A. Map Descriptor Lengths

metoneR4800CMD Use these strings in the configuration file to execute the different services	Length Reserve space in the Data Arrays
Send buffered record	15
Send current record	15
Clear buffer	2
Number of records	3
Hold time	5
Sample period	5
Mode request	3
Device select	2
Auto	2
Manual	2
Start counting by computer	2
Start counting by counter	2
Stop counting	2
Active mode	2
Standby mode	2
Universal Select	2

## Appendix B. How Data is Stored

### Table - How Data is stored

Command / Service	Offset	Contents
send buffered record	0	Trigger
	1	Set to zero if no records are available. Set to 1 if there is a record. Ditto for response '?' which means command not understood.
	2	status indicator
	3	month
	4	day
	5	year
	6	hour
	7	minute
	8	second
	9	period
	10	ch1_size
	11	ch1_count
	12	ch2_size
	13	Ch2_count
	14	location
15	Time Stamp – secs since 1970Jan01	
		<b>status indicator</b>
		blank space = No Alarms (value stored == 32)
		! = Check Sensor (value stored=33)
		\$ = alarm/count alarm (value stored = 36)
		% = check sensor alarm (value stored = 37)
Send Current Record		Same arrangement as buffered record
Clear Buffer	0	Trigger. Write here to trigger cmd. When complete a "C" is stored here (value 67)
	1	Count of how many times this has been executed
Number of Records	0	Trigger. Write here to trigger cmd. When complete a "D" is stored here (value 68)
	1	Number of Records
	2	Count of how many times this has been executed
Hold Time	0	Trigger. Write here to trigger cmd. When complete a "H" is stored here (value 72)
	1	Hour

	2	Minutes
	3	Seconds
	4	Count of how many times this has been executed
		Read setting with RDBC. Write settings with WRBC, WRBX (preferred) When you send a new hold time to the counter, the counter responds with the hold time and it is stored in offsets 1,2,3
Sample Period	0	Trigger. Write here to trigger cmd. When complete a "H" is stored here (value 72)
	1	Hour
	2	Minutes
	3	Seconds
	4	Count of how many times this has been executed
		Read setting with RDBC. Write settings with WRBC, WRBX (preferred) When you send a new sample period to the counter, the counter responds with the sample period and it is stored in offsets 1,2,3
Mode Request	0	Trigger. Write here to trigger cmd.
	1	When complete a "C" (counting) "H" (holding) or "S" (stopped) will be stored here when cmd has completed
	2	Count of how many times this has been executed
Device select	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Auto	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Manual	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Start counting by computer	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed

Start counting by counter	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Stop counting	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Active mode	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Standby mode	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed
Universal Select	0	Trigger. Write here to trigger cmd.
	1	Count of how many times this has been executed

## Appendix C. Supported Services

Always check the Data Sheet for an accurate and up to date list.

Services	
send buffered record	Read
send current record	Read
Clear buffer	Write
number of records	read
Hold time	Read & write
Sample period	Read & write
Mode request	Read
Device select	Write
Auto	Write
Manual	Write
Start counting by computer	Write
Start counting by counter	Write
Stop counting	Write
Active mode	Write
Standby mode	Write
Universal Select	Write

**Appendix D. Driver Error Messages**

<b>Error Message</b>	<b>Explanation and corrective action</b>
<p>We have shown place holders for the parts of the message which change.</p> <p>%s is a place holder for a text string.</p> <p>%d is a place holder for a number</p> <p>%c is a place holder for an alpha character.</p>	<p><i><b>FYI messages are informational and do not require a corrective action. Simply use them to confirm configuration / behaviors are what you expect.</b></i></p>