A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-107
Mitsubishi CHC Serial Driver

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

TABLE OF CONTENTS

1. Mitsubishi CHC Serial Driver Description

The Mitsubishi CHC Serial Driver allows the FieldServer to transfer data to and from devices over RS232 using Mitsubishi CHC Serial Driver protocol. The FieldServer can emulate a Client.

This driver is used to exchange data between a FieldServer and a Mitsubishi Heavy Industries communication interface known as a CHC-MF

The driver is a serial driver using a RS232 serial port to connect between the FieldServer and the CHC-MF. An RS485 port together with a converter can also be used for the connection.

The driver provides client and server functionality.

As a client the driver can poll for data from Air Conditioning Units via the CHC-MF interface as well as writing some control and set points.

Server functionality is provided only to support our ongoing quality assurance program by facilitating automated testing of the driver. It is not documented or supported. If required please contact the FST sales group to discuss your requirements.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 CHC node per connection
Server	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-8700-107	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

No special configuration of the CHC interface is required.

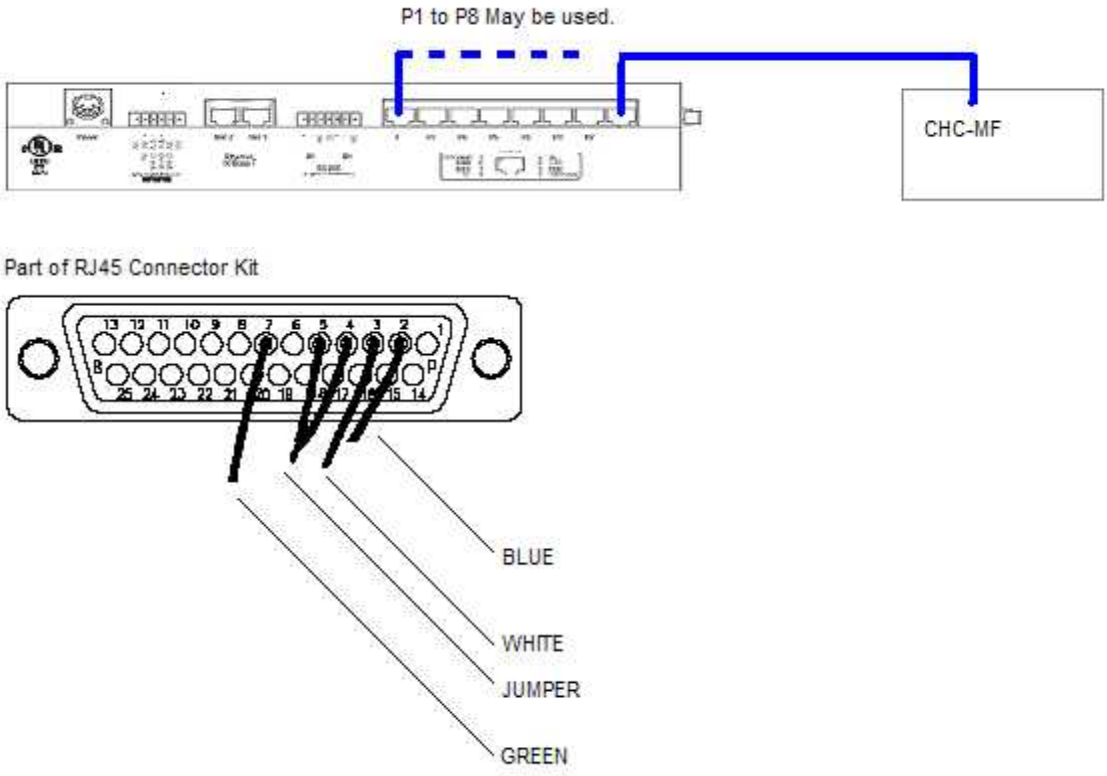
2.2.4. Optional Items

PART #	Vendor/Manufacturer	Description

3. Hardware Connections

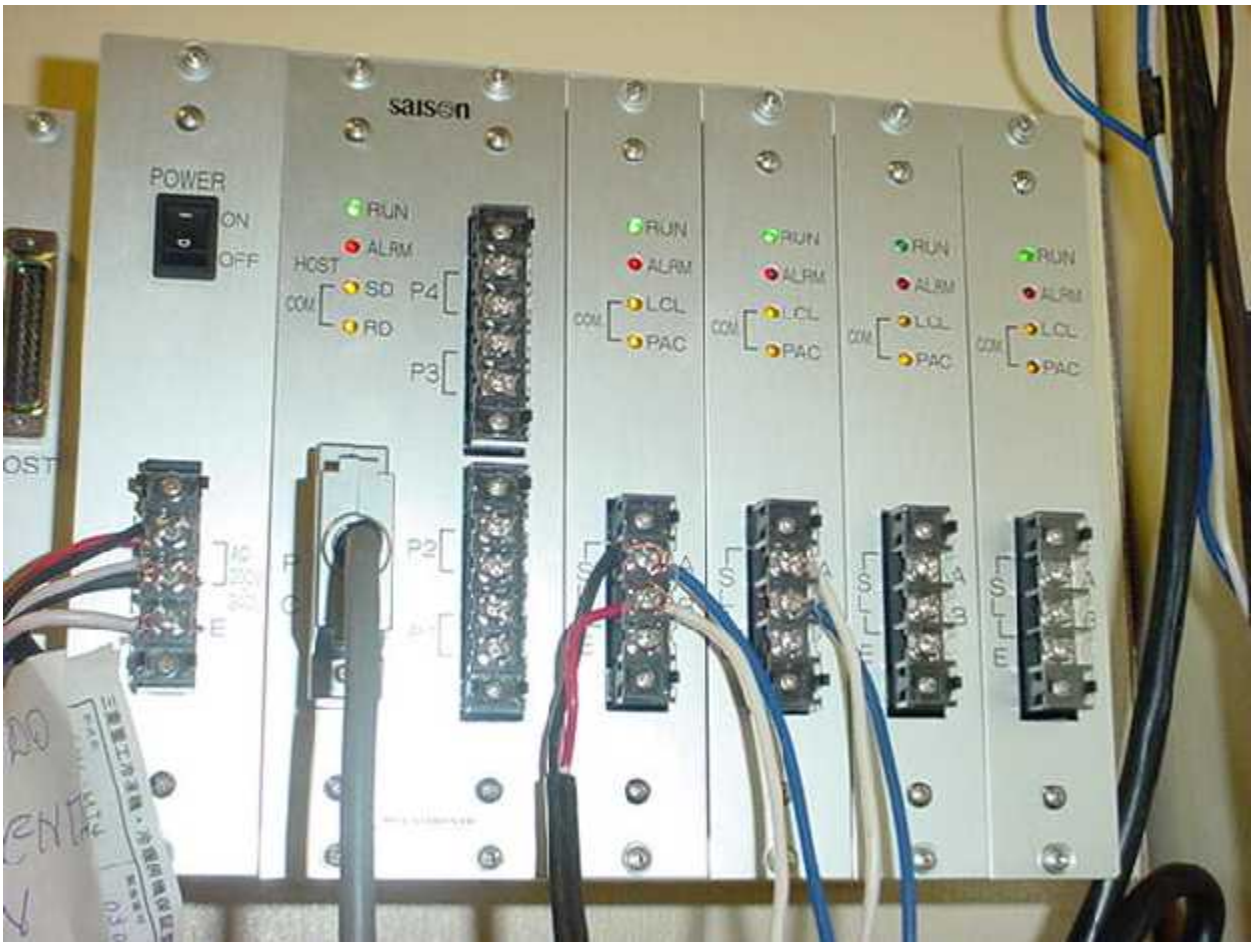
The FieldServer is connected to the CHC as shown in connection drawing.

Configure the CHC's Host Port according to manufacturer's instructions



The CHC appears to require wiring as if it were a DTE device. As the FieldServer is A DTE device this implies the cable is NULL Modem Like.

Pins 4 & 5 are connected internally in the CHC end. This is used to defeat the CTS/RTS Handshaking.



3.1. Hardware Connection Tips / Hints

This section is blank.

4. Configuring the FieldServer as a Mitsubishi CHC Serial Driver 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 Mitsubishi CHC Serial Driver Server.

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 Mitsubishi CHC 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,
DA_AI_01,               UInt16,         200
DA_AO_01,               UInt16,         200
DA_DI_01,               Bit,            200
DA_DO_01,               Bit,            200
```

4.2. Client Side Connection Descriptions

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

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	Mitsubishi-CHC Mitsubishi_CHC
Baud*	Specify baud rate	Driver Supports : 110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 28800; 38400; 57600; 115200 Baud Vendor Equipment Supports : 4800 Baud
Parity*	Specify parity	Driver Supports : Odd, **Even**, None Vendor Equipment Supports: Even
Data_Bits*	Specify data bits	Driver Supports : 7,**8** Vendor Equipment Supports : 7
Stop_Bits*	Specify stop bits	Driver Supports : 1,2 Vendor Equipment Supports: 1
Handshaking*	Specify hardware handshaking	**None**

[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.

Poll _Delay*	Time between internal polls	0-32000 seconds, **0.2 second**

Example

```
//      Client Side Connections

Connections
Port,           Protocol,           Baud,     Parity,     Data_Bits,
P1,             Mitsubishi-CHC,     4800,     Even,       7,
```

4.3. Client Side Node Descriptors

Create one Node for each CHC Interface. Do NOT create a node for each Air Conditioner Group.

Section Title
Nodes

Column Title	**Function**	**Legal Values**
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Modbus 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	Mitsubishi-CHC Mitsubishi_CHC
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[2]

Example

```
//  Client Side Nodes

Nodes
Node_Name           Node_ID     Protocol            Connection
CHC_01              1           Mitsubishi-CHC      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.

4.4. 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, WRBX Passive_Client

4.4.2. 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
Data_Type	Data type This commonly used parameter is not used by this driver.	Register, Coil, AI, DI
Length	Length of Map Descriptor Tells the driver how many points to read. The maximum point number is 20 so when you add the length to the address the value may not exceed 21 ie. The largest point number that can be read is 20.	1-20
Address	Starting address of read block The following are valid point numbers.	1 to 20

	1,2,3,4,5,6,7,8,,9,10,20 If you set the address to 1 and the length to 20 the driver will read all valid the points in the range 1-20. If you set the address to 12 and the length to 1, the driver will print an error as point 12 cannot be read.	
Store_Errors*	If this parameter is set to yes then you must specify the CHC, GateWay, Outdoor_Addr and Indoor_Addr. In addition, the 'function' parameter must be set to 'Passive_Client' If the parameter is set to no or is omitted then you must specify CHC and Group.	Yes, **no**
CHC	This parameter is required for all Map Descriptors. Its typical value is 1.	0 to 9.
Group*	This is the Air Conditioning Group Number. This parameter is required when you read data from / write data to a CHC.	1 to 999
GateWay*	This parameter is only required when the 'Store_Errors' parameter is set to yes. It is the number of the gateway that connects a CHC to a network of Air conditioners. It is usually a number in the range 1 to 8	1 to 8

Outdoor_Addr*	This parameter is only required when the 'Store_Errors' parameter is set to yes.	0, 1 , 2 etc
Indoor_Addr*	This parameter is only required when the 'Store_Errors' parameter is set to yes.	0, 1 , 2 etc

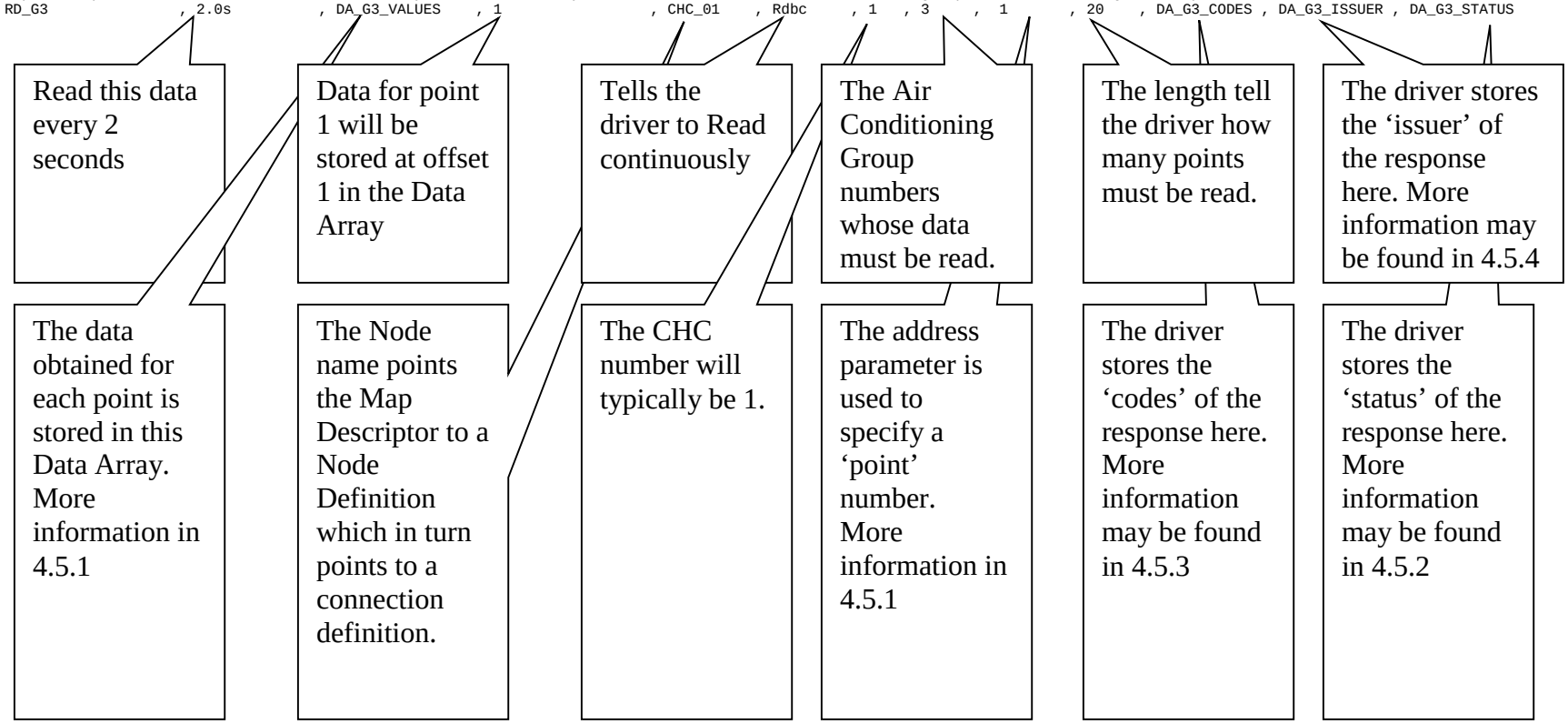
4.4.3. Timing Parameters

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

4.4.4. Map Descriptor Example 1 – Read all points from a single group.

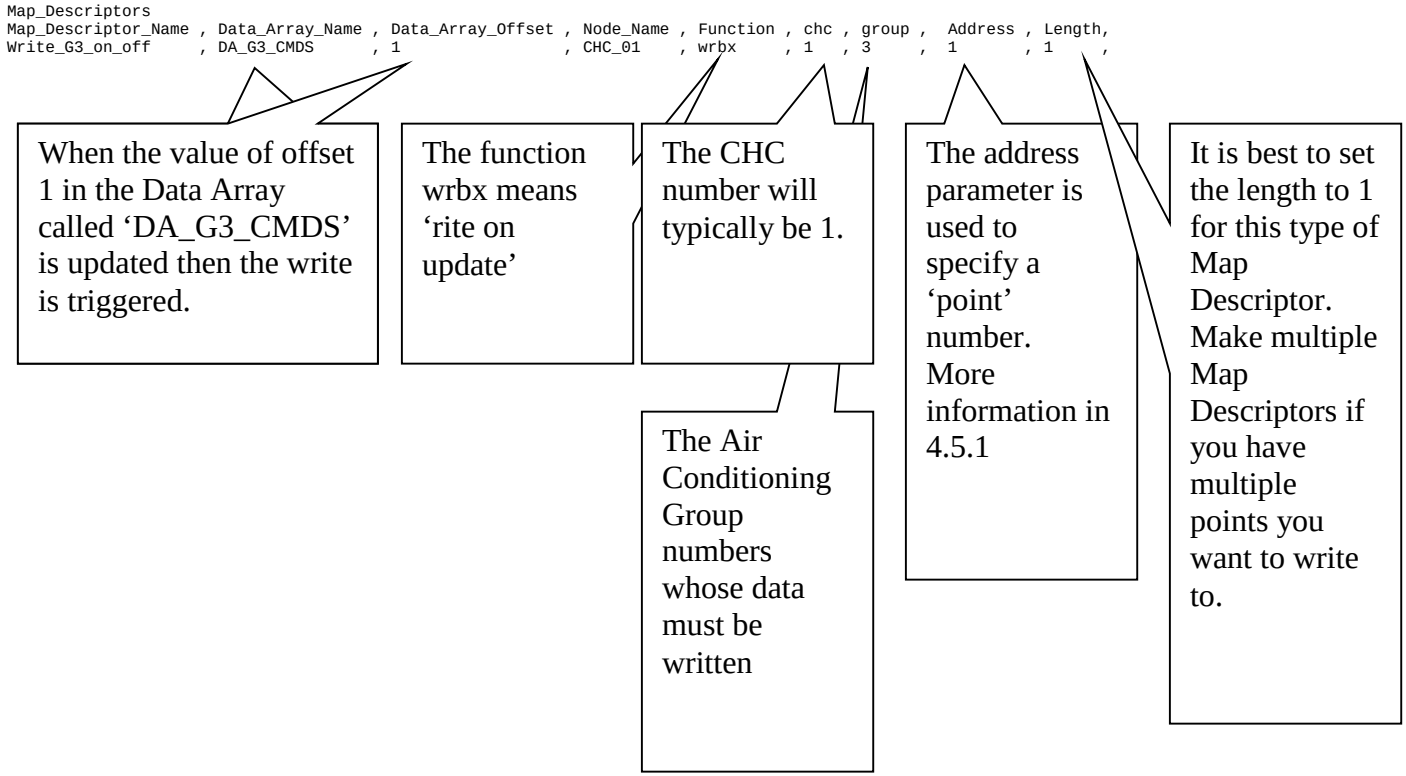
In this example the driver reads data for all the points of group 3 . It reads the data every 2 seconds because the `Scan_Interval` has been set to 2 seconds and the `function` has been set to RDBC – Read Block Continuous. It reads points 1 to 20 because the `Address` has been set to 1 and the `Length` has been set to 20. The driver knows that some of the points don't exist and will not produce an error. In addition to storing the current state / value of each point the driver also stores 2ndary information because 2ndary Data Arrays have been specified using the parameters named `DA_Byte_name, DA_Bit_Name , DA_Float_Name.` The secondary data is not stored if the parameters are omitted. Section 4.5 'Interpreting the values found in the Data Arrays' provides more information on how to interpret the data that is stored.

```
Map_Descriptors
Map_Descriptor_Name , Scan_Interval , Data_Array_Name , Data_Array_Offset , Node_Name , Function , chc , group ,  Address , Length, DA_Byte_name, DA_Bit_Name  , DA_Float_Name
RD_G3               , 2.0s          , DA_G3_VALUES    , 1                 , CHC_01    , Rdbc     , 1   , 3     ,  1      , 20    , DA_G3_CODES , DA_G3_ISSUER , DA_G3_STATUS
```



4.4.5. Map Descriptor Example 2– Write a single point

In this example the driver writes a single data value to the specified data point in the specified Air Conditioning group. This type of map descriptor is used when an upstream nodes write a new setpoint / state. The new setpoint / state is written to the Data Array. This Map Descriptor watches the Data Array. If it sees the value being updated then this triggers the write. One write is triggered for each update. An update of a Data Array does not require that the alue change, it only requires that a value is stored in the Data Array.

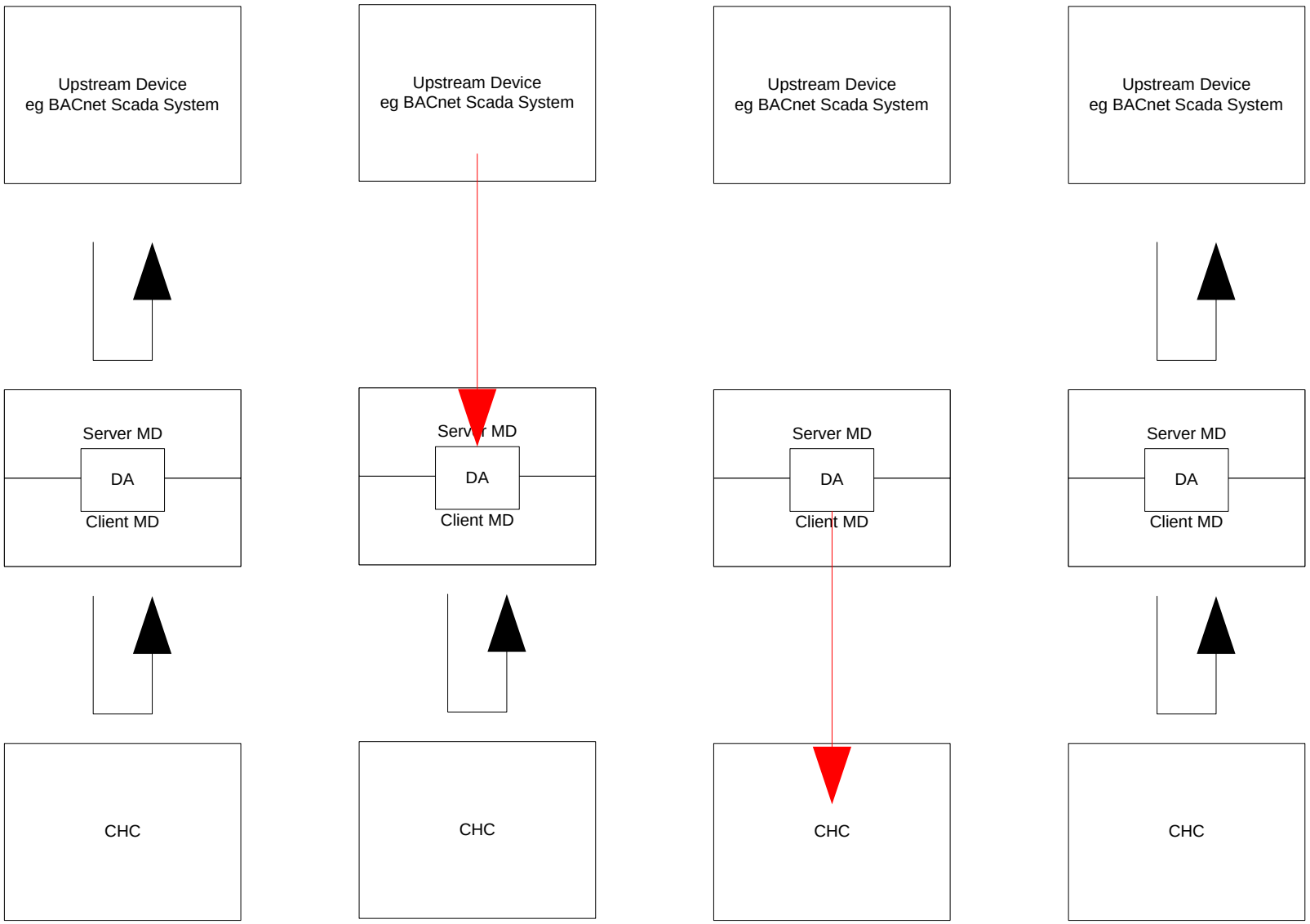


4.4.6. Map Descriptor Example 3 Write through a read

Write-Through is a FieldServer technology that allows you to skip the requirement for a write map descriptor if your are already reading data. Use the Map Descriptor In example 1 to read data. In that example, the Map Descriptor reads all the data for Group 3. Having read this data into a Data Array from the CHC, the FieldServer is normally configured to serve this data to an upstream device using a different protocol (eg BACnet). Normally the upstream device reads the data and the FieldServer serves the data. What happens when the upstream node wants to turn a downstream device on / off or change the set point ? It simply has to write to one of the points that it is reading. When the upstream node write into the Data Array, the driver recognizes the desire to have the state / set point 'Write-Through' the FieldServer. It creates a temporary one shot write (each time the upstream devices updates the Data Array) using the information from the existing Map Descriptor.

In the diagram on the next page 4 stages are indicated. In state 1 , the driver reads data from the CHC using 'rdbc' Map Descriptors and serves data to the upstream node using server Map Descriptors. In Stage 2, the Upstream node, stops reading and writes a value into the Data array being shared between the protocols. In stage 3 – the FieldServer generates the 'write-Through' using the rdbc Map Descriptor and the offset into the Data Array that was written to form a write to the downstream device. In stage 4 – the write has been completed and the upstream nodes returns to reading data from the FieldServer.

Using the Map Descriptor provided in example 1, if the upstream node write to offset 20 in the Data Array then the 'Write Through' would be generated for point 20. The driver uses the MD's length, address and offset information and the offset of the Data Array element that was updated to calculate the address to write to.


Upstream Device
eg BACnet Scada System
Server MD
DA
Client MD
CHC
Stage 1 - Reading
Stage 2 - Upstream Write
Stage 3 - Write Through
Stage 4 - Reading

4.4.7. Map Descriptor Example 4– Store Error Messages sent by the CHC

When curtain errors occur, the CHC sends an error message. The driver can process and store information from there messages. The driver cannot actively poll for this data. These messages are sent by the CHC without solicitation.

These MD's are used to store from error messages sent by the panel. Create one for each CHC, Gateway, Outdoor and Indoor Address combination. If an error is reported then offset 0 is updated with the most recent error number. reported and then the offset=Error_Number is also set to a non-zero value when an error is reported and cleared when the error is cleared.

eg Err 40 is reported. Offset0 set to 40 and Offset40 set to 40

eg Err 20 is reported. Offset0 set to 20 and Offset20 set to 20

eg Err 40 is cleared. Offset0 set to 40 and Offset40 set to 0

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Node_Name , Function       , chc , Gateway , Outdoor_Addr , Indoor_Addr , Store_Errors, Length, DA_Byte_name , DA_Float_Name
Store_Errs          , DA_121_ERRORS   , 0                 , CHC_01    , Passive_Client , 1   , 1       , 2            , 3          , Yes          , 100   , DA_121_CODES   , DA_121_STATUS
```

Tells the driver to listen passively for an 'Error' message whose attributes match those specified on the Map Descriptor.

For each combination of Gateway, Outdoor Address and Indoor Address that can generate an error, a new Map Descriptor must be created.

Setting 'Store_Errors' to yes tell the drive to use this Map Descriptor to store error message data.

4.5. Interpreting the values found in the Data Arrays

4.5.1. Point Addresses and Values

Values are stored in the primary data array specified by the 'Data_Array_Name' parameter in the Map Descriptor.

Where is the data stored in the Data Array ?

The driver adds the value of the 'address' to the value of the 'Data_Array_Offset' parameter.

Therefore if the value of 'Data_Array_Offset' is zero, the value for the 'Temperature Setting' (point 2) is stored at location 2 in the Data Array. Data Array locations are numbered from zero, the the point value is stored in the 3rd element of the data array (known as location 2).

Point / Address	Description	Type	Values
1	Start/Stop Status	Discrete	0 = Off 1 = On
2	Temperature Setting	Analog	180 to 300 corresponding to a temperature of 18 to 30°C
3	Automatic temperature control mode setting	Discrete	0 = Manual Mode 1 = Automatic Mode
4	Temperature control mode setting	Discrete	0 = Heating Mode 1 = Cooling Mode 2 = Ventilation Mode
5	Room Temperature	Analog	130 to 300 corresponding to a temperature of 13 to 30°C
6	Remote Control Mode Setting 1	Discrete	0 = Start/Stop Allowed 1 = Prohibited
7	Remote Control Mode Setting 2	Discrete	0 = Temperature Setting Allowed 1 = Prohibited
8	Remote Control Mode	Discrete	0 = Temperature

	Setting 3		control mode setting allowed 1 = Prohibited
9	Filter sign reset	Discrete	0 = Off / Reset
10	Forced thermo off (electricity demand)		0 = Forced thermo off enabled 1 = Forced thermo off disabled
20	Electric Energy	Analog	x 10 actual value

4.5.2. Point Status

The point status may also be used to determine additional information. The point status is stored when the (misleadingly named) 'DA_Float_Name' is specified with a Data Array Name.

The driver inspects the status when storing the point value in the primary data array. If it sees the "??" status then it stores a special value in the primary data array. In the case of a discrete point the value 99 is stored and in the case of an analog point the value of -1 is stored. As a consequence of this an upstream device is not dependent on interpreting the 'Status' information.

Point / Address	Description	Status
1	Start/Stop Status	00 = Normal – All units not operating 01 = Normal – One or more units operating 08 = Error in one or more units while all units are not operating 09 = Error in one or more units while one or more units are operating ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
2	Temperature Setting	00 = Normal ?? = Communication Error – When this condition is detected by the driver the stores the value -1 in the primary data array so that the upstream device can detect the error.
3	Automatic temperature	00 = Manual Mode

	control mode setting	01 = Automatic Mode ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
4	Temperature control mode setting	00 = Heating Mode 01 = Cooling Mode 02 = Ventilation Mode ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
5	Room Temperature	00 = Normal ?? = Communication Error – When this condition is detected by the driver the stores the value -1 in the primary data array so that the upstream device can detect the error.
6	Remote Control Mode Setting 1	00 = Start/Stop allowed 01 = Prohibited Mode ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
7	Remote Control Mode Setting 2	00 = Temp Setting Allowed 01 = Prohibited Mode ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
8	Remote Control Mode Setting 3	00 = Temp Control Mode Setting Allowed 01 = Prohibited Mode ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
9	Filter sign reset	00 = Filter Sign Off – all units 01 = Filter Sign is on for one or more units ?? = Communication Error – When this condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
10	Forced thermo off (electricity demand)	00 = Forced Thermo off enabled (representative air conditioning unit) 01 = Forced Thermo on enabled (representative air conditioning unit) ?? = Communication Error – When this

		condition is detected by the driver the stores the value 99 in the primary data array so that the upstream device can detect the error.
20	Electric Energy	00 = Normal ?? = Communication Error – When this condition is detected by the driver the stores the value -1 in the primary data array so that the upstream device can detect the error.

Status data are stored by writing the two bytes directly to the Data Array. See section "4.5.3 Response Codes" for a list of typical values.

4.5.3. Response Codes

Response Codes are stored by writing the two bytes directly to the Data Array

'00' = 0x3030 = 12336 decimal
'01' = 0x3031 = 12337 decimal
....
'09' = 0x3039 = 12345 decimal

'??' = 0x3f3f = 16191 decimal

'A0' = 0x4130 = 16688 decimal
'A1' = 0x4131 = 16689 decimal
'A2' = 0x4132 = 16690 decimal
'A3' = 0x4133 = 16691 decimal
'A4' = 0x4134 = 16692 decimal
'A5' = 0x4135 = 16693 decimal
'A6' = 0x4136 = 16694 decimal

'B0' = 0x4230 = 16994 decimal
'C0' = 0x4330 = 17200 decimal
'C4' = 0x4334 = 17204 decimal
'D0' = 0x4430 = 17456 decimal
'L4' = 0x4C34 = 19508 decimal

It is beyond the scope of this manual to provide 'meaning' to these codes. Contact your CHC vendor for additional information

4.5.4. Issuer Information

Typically the issuer is

0x3030 = '00 -> Typical'
0x3f3f = '??' -> Communication error
0x3039 = '09' -> Point changed because of 'on the spot' manipulation

It is beyond the scope of this manual to provide ‘meaning’ to these codes. Contact your CHC vendor for additional information

5. Configuring the FieldServer as a Mitsubishi CHC Serial Driver Server

This driver has a server side implemeted but it is used for FieldServer's Quality Assurance program and is not docuemnted or supported. If you are interested in using Sever Side features then please contact FieldServer's marketing Group.

Appendix 1. Advanced Topics

This section is blank.

Appendix 2. Troubleshooting tips

2.1.1. Connection Tips & Hints

This section is blank.

2.1.2. Driver Error Messages

Eror Message	Explanation and corrective action
CHC:#1 FYI. Use an array called <%s> to expose diagnostic info.	Read section 2.1.3
CHC:#2 FYI. Poll Delay set to %5.2f Secs	This message is informational. No corrective action is required. The poll delay is the minimum aount of time that must pass between successive messages. This parameter is normally controllable with the parameter called 'Poll_Delay' specified on the connection. This driver has a special minimum delay requirement. This message prints the driver default.
CHC:#3 Err. CHC, Gateway, In/Outdoor required.	When the Map Descriptor parameter called 'Store_Errors' Is set to yes then the same map descriptor must also specify the following parameters; Gateway, Indoor_Address, Outdoor_Adress. Correct the configuration CSV file by editing it, specifying the additional parameters. The download the corrected configuration file and reset the FieldServer for the changes to take effect.
CHC:#4 Err. CHC, Group Required.	The group number parameter must always be specified. Valid group numbers are in the range 1-999. Correct the configuration CSV file by editing it, specifying the additional parameters. The download the corrected configuration file and reset the FieldServer for the changes to take effect.

CHC:#5 Err. CHC, Address out of range (1-20).	The 'Adress' parameter's legal values are in the range 1 to 20. This parameter is used to specify the point that must be read / written. See section 4.5.1 for a list of points. Correct the configuration CSV file by editing it, specifying the correct value for this parameter. The download the corrected configuration file and reset the FieldServer for the changes to take effect.
CHC:#6 Err. CHC, Address & Length out of range (1-20).	When the length is added to the Address a value larger than 20 is obtained. Thsu the driver is being asked to read points whose numbers are out of range. The largest point number that can be read is 20. Adjust the length parameter accordingly. Correct the configuration CSV file by editing it, specifying the correct value for this parameter. The download the corrected configuration file and reset the FieldServer for the changes to take effect.
CHC:#7 Err. CHC not specified.	The 'CHC" parameter must be specified on all Map Descriptors. Typically the value of this parameter is 1. Correct the configuration CSV file by editing it, specifying the correct value for this parameter. The download the corrected configuration file and reset the FieldServer for the changes to take effect.
CHC:#8 Err. Unexpected response to select=0x%x	Each time this message is printed the driver produces a 'protocol error' stat. The message is printed when the driver expects one kind of response and gets another. This could occur because of noise, occupation a change in firmware or because of a driver error.

	If the error occurs occasionally and does not affect operations then it may be safe to ignore the message – assuming it is an occasional corrupt message. In such cases you should expect to see a different character printed in the message each time. If the error occurs frequently, affects operations or if the message is printed and always has the same unexpected byte then you should contact Tech Support after taking a log.
CHC:#9 Err. Unexpected NAK response to select txt=0x%x	Read the notes for Error 8.
CHC:#10 Err. Unexpected response to select txt=0x%x	Read the notes for Error 8.
CHC:#11 Err. Polled %d times(s). No response yet.	If this error occurs often or affects normal operation then take a log and contact Tech Support. The message is printed when the driver attempt to poll for data and does not receive a response. This is possible if the CHC is very busy.
CHC:#12 FYI No Data back yet. NAK	This message may be safely ignored. After sending a poll the driver looks to see if there is a response. Sometime the CHC may be busy and may send a NAK back instead of the response. This message record this event. You do not need to take any corrective action. The driver will wait for a response and produce a proper error should one occur.
CHC:#13 Err. Parse Failed.	The driver prints the whole message that could not be parsed. Parsing is the process of interpreting a message. An error message preceding this one probably gives the reason for the failure. This message is printed to help Tech support analyze the problem. There is no corrective action you can take. If the error occurs occasionally then it may be th e result of a corrupt message.. If it occurs frequently then take a log and call Tech support.

CHC:#14 FYI. Expected EOT. Got=0x%x. Assuming text msg	The driver expected an EOT (a message which indicates that there is no / no more data). Instead the drive received a message beginning with the byte printed in the error line. The driver will treat this as additional data and try and interpret it. If that process fails then an error message will be printed. There is no corrective action you can take.
CHC:#15 Err. Extract of Extract of Request Error. ret=%d %x,%x,%x,%x,%x,%x,%x,%x	Parsing (the process of interpreting a message) failed when it tried to extract the status of a Request Error. This message will help Tech support analyze the problem. If the message is printed rarely then treat it as a corrupt message and ignore it. Otherwise take a log and contact Tech Support.
CHC:#16 Err. No place to store #%c%c Grp=%d Pnt=%d Port=%d	Sometime the CHC will include in a response data for a point that wasn't polled. It does this when the status of that point changes under certain conditions. When this message is printed the driver is telling you that it found data for the group / point identified but it could not find a Map Descriptor to use to store the data. If the reported point is of no interest to you then ignore the message. Otherwise, edit the config file, adding a Map Descriptor (MD) for the point indicated. The MD's function could be set to 'passive' if you don't want to actively poll for the point's status. In this case the MD will be used to store the points data preventing this message being printed again. Once you have corrected the CSV file, download the modified file to the FieldServer and reset the FieldServer for you changes to take effect. This particular message is printed when

	the driver is processing a Request Error.
CHC:#17 Err. Extract of discrete. ret=%d %x,%x,%x,%x,%x,%x,%x,%x	Parsing (the process of interpreting a message) failed when it tried to extract the status of a discrete point. This message will help Tech support analyze the problem. If the message is printed rarely then treat it as a corrupt message and ignore it. Otherwise take a log and contact Tech Support.
CHC:#18 Err. No place to store #%c%c Grp=%d Pnt=%d Port=%d	This particular message is printed when the driver is processing a discrete point. Read the notes for Error #16 for additional information.
CHC:#19 Err. Extract of ana. ret=%d %x,%x,%x,%x,%x,%x,%x,%x,%x,%x	Parsing (the process of interpreting a message) failed when it tried to extract the status of an analog point. This message will help Tech support analyze the problem. If the message is printed rarely then treat it as a corrupt message and ignore it. Otherwise take a log and contact Tech Support.
CHC:#20 Err. No place to store #%c%c Grp=%d Pnt=%d Port=%d	This particular message is printed when the driver is processing an analog point. Read the notes for Error #16 for additional information.
CHC:#21 Err. Extract of err. ret=%d %x,%x,%x,%x,%x,%x,%x,%x,%x,%x	Parsing (the process of interpreting a message) failed when it tried to extract the status of an error point. This message will help Tech support analyze the problem. If the message is printed rarely then treat it as a corrupt message and ignore it. Otherwise take a log and contact Tech Support.
CHC:#22 Err. No place to store Err Report. #%c%c gateway=%d outdoor_addr=%d indoor_addr=%d port=%d	This particular message is printed when the driver is processing an unsolicited error message sent by the CHC. Read the notes for Error #16 for additional information.
CHC:#23 Err. Bad Response Header: No STX	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.

CHC:#24 Err. Bad Response Header: No SA	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#25 Err. Bad Response Header: No SI	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#26 Err. Bad Response chksum	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#27 Err. Bad Rcvd Seq Number	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#28 Err. Parse failed. Beyond eob.	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#29 Err. No # at begin of msg	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#30 Err. No way to process Code=%c%c	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#31 Err. No ETX.	There is no corrective action that you can take. If the error is rare then ignore it and assume it was produced by noise / corruption. Otherwise, take a log and contact Tech Support.
CHC:#32 Err. Point=%d Cant be Read.	The point indicated cannot be read. It could be that it is a write only point. More likely the points address is invalid. Valid points are tabulated in section 4.5.1

CHC:#33 Err. Point=%d Cant be Read.	The point indicated cannot be read. It could be that it is a write only point. More likely the points address is invalid. Valid points are tabulated in section 4.5.1
CHC:#34 Err. Point=%d Cant be Written.	The point indicated cannot be written . It could be that it is a read only point. Valid points are tabulated in section 4.5.1. Only some points can be written to.
CHC:#35a Err. Parse failed. Beyond eob.	All Error #35's are produced by the server side of the driver. If you see any one of them then take a log and contact tech Support as they should only be printed when FieldServer's QA procedure are executed.
CHC:#35b Err. Parse failed. No $ at begin.	See message #35a.
CHC:#35c Err. DA=%s not found	See message #35a.
CHC:#35d Err. DA=%s to short. Act=%d Rqd=%d	See message #35a.
CHC:#35e Err. DA=%s not found	See message #35a.
CHC:#35f Err. DA=%s to short. Act=%d Rqd=%d	See message #35a.
CHC:#35g Err. Point=%d Cant be served.	See message #35a.
CHC:#35g Err. Point=%d Cant be served.	See message #35a.
CHC:#35h Err. DA=%s not found	See message #35a.
CHC:#35i Err. DA=%s to short. Act=%d Rqd=%d	See message #35a.
CHC:#35k Err. DA=%s not found	See message #35a.
CHC:#35m Err. DA=%s to short. Act=%d Rqd=%d	See message #35a.
CHC:#35n Err. Point=%d Cant be Set.	See message #35a.
CHC:#35o Err. Point=%d Cant be served.	See message #35a.
CHC:#35p Err. No way to process Code=%c%c	See message #35a.
CHC:#35q Err. No ETX.	See message #35a.
CHC:#35r Err. Slave: Parse: Failed.	See message #35a.
CHC:#36 Err. Checksum calc3 is wrong.	If you see this message then contact Tech Support after taking a log.
CHC:#36 Err. Checksum calc1 is wrong.	If you see this message then contact Tech Support after taking a log.

CHC:#36 Err. Checksum calc2 is wrong.	If you see this message then contact Tech Support after taking a log.
CHC:#37 Err. Diagnostic 1	If you see this message then contact Tech Support after taking a log.
CHC:#38 Err. Cant open DA_UNSOL	If you see this message then contact Tech Support after taking a log.
CHC:#38 Err. Cant open DA_ERROR	If you see this message then contact Tech Support after taking a log.

2.1.3. Exposing Driver Stats

The diver makes some of its operating statistics available in a Data Array where they can be read by an upstream device. 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,
mits-chc-stats,         UINT32,             1000,
```

The driver maintains one set of stats for each communication port.
To determine the base offset use the following formula:

Base Offset = Port Number * 100 where Port Number is the port number printed in the error log by message #22.

Relative Offset : Calculate the offset of each stat by adding the relative offset to the base offset.

Relative Offset	Description
1	Increments each time point data is recbd but there is no MD to store the point
2	CHC number of the most recent Point status message received for which there was no storage location (gateway number for errors)
3	Group number of the most recent Point status message received for which there was no storage location (outdoor addr for errors)
4	Point number of the most recent Point status message received for which there was no storage location (indoor addr for errors)
5	Code (1st Byte) of the most recent Point status message received for which there was no storage location
6	Code (2nd Byte) of the most recent Point status message received for which there was no storage location
7	For diagnostics. Controls Server response. When set no responses are sent
8	For diagnostics. Controls Server response. Text responses are never sent. Only EOT's
9	Increments each time a sequence of polls don’t result in a text response
10	For diagnostics.
11	For diagnostics.
12	For diagnostics.
13	For diagnostics.

Relative Offset	Description
14	Increments each time a select msg is sent
15	Increments each time an ACK is sent
16	Increments each time an EOT is sent
17	Increments each time a polling msg is sent
18	Increments each time a text msg is sent
19	Increments each time a text read msg is sent
20	Increments each time a text write msg is sent
21	Increments each time a text msg is received

Appendix 3. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
23 Aug 2004	PMC		1.00a	0	Document Created