A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-100 Heatcraft Smart Controller II

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

TABLE OF CONTENTS

1. Heatcraft Smart Controller II (HCSCII) Description

The Smart Controller is used for monitoring and programming a complete refrigeration system for optimum control. A single Smart Controller can control four independent systems with up to four evaporators on each system. The Heatcraft Smart Controller II Driver allows the FieldServer to transfer data to and from devices over RS-232 using Heatcraft Smart Controller II Driver protocol. The FieldServer can emulate either a Server or Client. The FieldServer acts as a common node, allowing communication between the Smart Controller and other nodes supporting different protocols. Contact FieldServer Technologies for list of supported protocols.

As a client, the FieldServer polls Smart Controller devices for system parameters and data and stores this information in Data Arrays. Logged error/alarm and logged data are captured for each sub subsystem per Smart Controller.

As a Server, the FieldServer emulates a Smart Controller and is able to be polled by a remote node for system parameters and data. Remote nodes are able to update the Data Arrays on the FieldServer and this updated data is written to the Smart Controller by the FieldServer. The FieldServer also serves logged errors/alarms and other logged data.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	The FieldServer can support one Smart Controller device or Main communication hub per port. A Main communication hub can be connected to up to four Smart Controllers or secondary communication hubs. Each secondary communication hub can support up to four Smart Controllers. Thus the FieldServer can support a maximum of 16 Smart Controllers per port with the use of one main and four secondary communication hubs.
Server	32	*[e.g: This is the limit per i/net panel. The 32 nodes correspond to the maximum of 32 mr's that an i/net panel supports.]*

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-23	RS-232 DB9F Connector
FS-8700-100	Driver Manual.

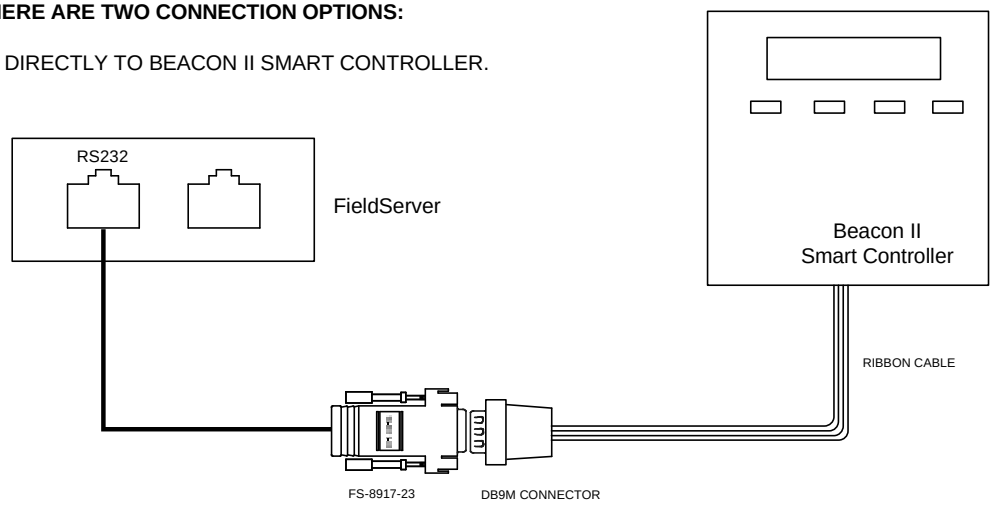
3. Hardware Connections

The FieldServer is connected to the Smart Controller Device as shown in connection drawing.

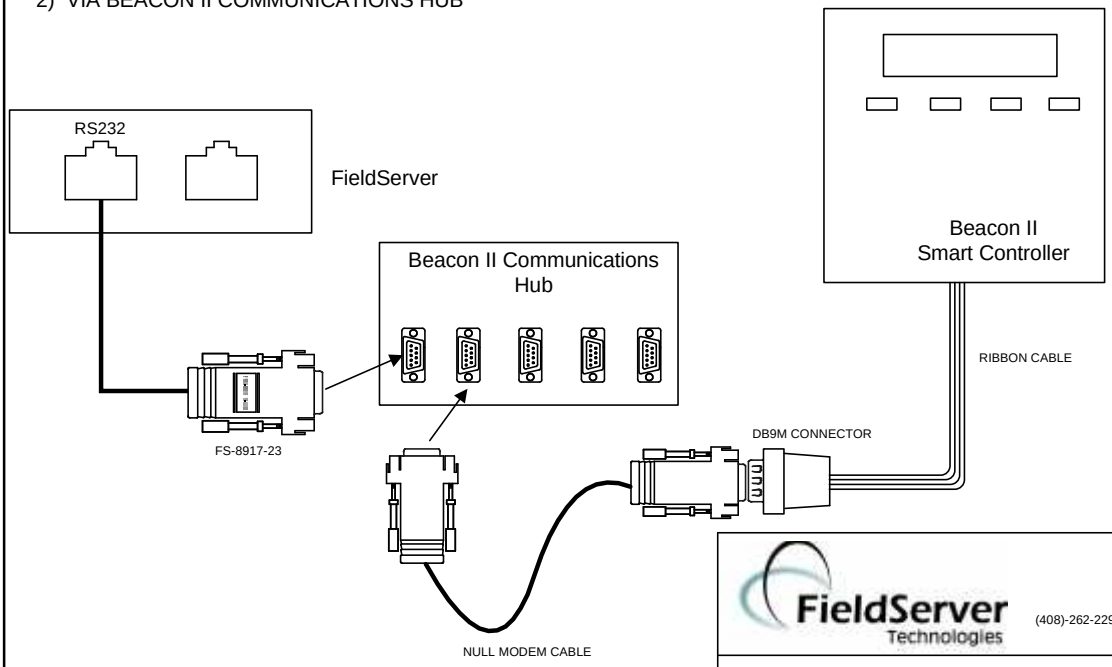
Configure the Smart Controller Device according to manufacturer's instructions

THERE ARE TWO CONNECTION OPTIONS:

1) DIRECTLY TO BEACON II SMART CONTROLLER.



2) VIA BEACON II COMMUNICATIONS HUB



FieldServer Technologies (408)-262-2299

DRIVER, HEATCRAFT
CONNECTION DIAGRAM

BASE NAME:
FILE NAME: VisioDocument

DATE: 11/2/2004
BY: MN

4. Configuring the FieldServer as a Heatcraft Smart Controller II 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 Heatcraft Smart Controller II 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 Heatcraft Smart Controller II 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.	Byte, Uint16, Float, Int, Uint32 Recommended: Float
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.	1-33,000

Example

```
//  Data Arrays
Data_Array_Name,        Data_Format,    Data_Array_Length
DA_P1N11,               Float,          700
DA_ERRALM_LOG,          Float,          4804
DA_DATA_LOG,            Float,          25000
```

4.2. Client Side Connection Descriptions

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8[1]
Baud*	Specify baud rate	38400 (vendor limitation)
Parity*	Specify parity	Even, Odd, None, Mark, Space
Data_Bits*	Specify data bits	7, 8
Stop_Bits*	Specify stop bits	1
Protocol	Specify protocol used	HcscII , Smart-Controller
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, None
Poll Delay*	Time between internal polls	0-32000 seconds, **1s**
hcscII_hub_connected*	Specify if hub connected	YES/NO
hcscII_password_timeout*	Time in seconds after which the driver marks the node as offline if it did not get a Pass challenge from the Smart Controller.	1-2147483647 120
hcscII_dtr_toggle_time*	Time in seconds the Driver keeps DTR low before changing to high.	1-2147483647 5

Example

```
//  Client Side Connections
Connections
Port,   Baud,    Protocol,  hcscII_hub_connected,  hcscII_password_timeout,  hcscII_dtr_toggle_time
P8 ,    38400,   hcscII  ,  YES                ,  120,                        5
```

[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.

4.3. Client Side Node Descriptors

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	This commonly used parameter is not required or used by this driver.	11, 12, 13, 14 21, 22, 23, 24 31, 32, 33, 34 (all in Hex) 41, 42, 43, 44 only – anything else specified will be changed by driver to 11 (hex)
Protocol	Specify protocol used	HcscII , Smart-Controller
Port	Specify which port the device is connected to the FieldServer	P1-P8[2]

Example

```
//  Client Side Nodes
Nodes
Node_Name          Node_Id ,      Protocol ,      Port
SC_P1N11 ,         11     ,       HcscII   ,      P8
```

[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 (Not used internally, but declaration is essential)	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	RDBC, WRBC, WRBX**

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
Length	Length of Map Descriptor	1 to 4 depending on Map Descriptor's function: rdb, Rdbc - 4 Wrbx or Wrbc – 1 to 3 1-maximum if function declared in "Data Array" section is Server"
Address*	Not required by this driver.	
HcscII_cmd	This field is mandatory. It tells the driver what query or command is required.	60-FF (hexadecimal) *or* English name *or* one keyword for group of commands (See Appendix A.2)
HcscII_reset_on_reboot*	All read data from a Smart Controller will be reset on getting Password Challenge message.	YES, NO
Record_number*[3]	Specify record number to fetch from target device.	0-32767 0

4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	>0.1s

**WRBX is recommended for setting parameters at target device. Additional notes in Appendix A

[3] Note: Use this parameter only when HcscII_cmd parameter is set to EB or EC—F3

4.4.4. Map Descriptor Example 1 – Read data.

The Map Descriptor causes the driver to poll the Smart Controller II Device (SC_P1N11) every 1.0 seconds, read data and update the allocated memory for the command declared under HcscII_cmd parameters.

Map_Descriptor_Name, CMD_P1N11_89,	Scan_Interval, 1.0,	Data_Array_Name, DA_P1N11,	Data_Array_Offset, 214,	node_name, SC_P1N11,	Function, Rdbc,	Length, 4,	HcscII_cmd 89

Map Descriptor names need not be unique, but since some driver error messages use the name it is helpful to have meaningful names.

The name of an array which the driver will use to store data from the node specified under this Map Descriptor. Ensure that the array is long enough and that the data type is BYTE.

In this example the offset into the Data Array is set to 214 to correspond with the device (node_name) being polled using this Map Descriptor.
Data read for command 89

Although this offset is not used internally it will make sense if it is set according to the formula:
Offset = 50 + (cmd(dec)-96)*4
In this example hcscII_cmd = 89h = 137
Thus offset = 50 + (137-96)*4 = 214

Read Continuously.

4 data bytes are associated with every read query; therefore keep length 4 for every read poll.

This is the Command number that will be used by the hcscII driver to poll the Smart Controller to read system parameters or data.
In this example the command is 89h = 137(dec) i.e. this Map Descriptor is responsible for reading the desired preset temperature for all systems controlled by Smart Controller SC_P1N11.
89 can be replaced with its corresponding English name "Read All Desired Box Temps"

4.4.5. Map Descriptor Example 2 – "Commonset".

This single Map Descriptor can poll for multiple commands independent of the settings in the configuration file. Appendix A.2. lists the parameters that are read under this command. The Data_Array_Offset can be any valid offset as storage location is prefixed for every command. See Appendix A.3 for more detail.

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_commonset,	1.0,	DA_P1N11,	214,	SC_P1N11,	Rdbc,	4,	COMMONSET

4.4.6. Map Descriptor Example 3 - System #?

This a single Map Descriptor will read all data for sub system #1 not included in COMMONSET . Appendix A.2. lists the parameters that are read under this command. The Data_Array_Offset can be any valid offset as storage location is prefixed for every command. See Appendix A.3 for more detail.

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_system1,	1.0,	DA_P1N11,	218,	SC_P1N11,	Rdbc,	4,	SYSTEM#1

For second system use SYSTEM#2
For third system use SYSTEM#3
For fourth system use SYSTEM#4
OR

4.4.7. Map Descriptor Example 4 – Subset1#? and Subset2#?

Each system data can be read with two Map Descriptors with different updating time.

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_sbuset11,	1.0,	DA_P1N11,	222,	SC_P1N11,	Rdbc,	4,	SUBSET1#1

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_subset21,	3.0,	DA_P1N11,	226,	SC_P1N11,	Rdbc,	4,	SUBSET2#1

Here SUBSETx#y where x = subset number (1,2)
and y = sub system number (1,2,3 or 4)

4.4.8. Map Descriptor Example 5 - ERR_ALARM_LOG

This single Map Descriptor will issue commands to read logged data for errors/alarms. The FieldServer will store logged data for error/alarms in a Data Array Da_err_alrm_log. Refer to Appendix A.3 for more detail

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_system1,	1.0,	Da_err_alrm_log,	0,	SC_P1N11,	Rdbc,	4802,	ERR_ALARM_LOG

The latest record will always be the first record in this Data Array and can be found at offset 2.
The value at offset 0 indicates the record number for the latest record.
The value at offset 1 indicates the maximum number of records available.

4.4.9. Map Descriptor Example 6 - DATA_LOG_?

This Map Descriptor will issue commands to read logged data for sub systems in a Smart Controller. The user must define the number of subsystems to capture the logged data. The FieldServer will store logged data in a Data Array Da_data_log. Refer to Appendix A.3 for more detail

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
MD_system1,	1.0,	Da_data_log,	0,	SC_P1N11,	Rdbc,	25000,	DATA_LOG_1

Here DATA_LOG_? Where ? = 1 will collect logged data for subsystem number 1
? = 2 will collect logged data for subsystem number 2
? = 3 will collect logged data for subsystem number 3
? =4 will collect logged data for subsystem number 4

The latest record will always be the first record in this Data Array and can be found at offset 2.
The value at offset 0 indicates the record number for the latest record.
The value at offset 1 indicates the maximum number of records available.

4.4.10. Map Descriptor Example 7 - Write Data Strategy 1

This example illustrates how to write (Set) data to a Smart Controller II device (SC_P1_N11). There are always 4 values in a single write message sent to the Smart Controller, one for each system or subsystem. The data is read by the FieldServer, and when all 4 values have been updated, a trigger flag is set and the data is written to the target device. This strategy insures data integrity.

In this example whenever trigger flag at offset 8 is set, the FieldServer will initiate the 68h command to the Smart Controller with previously loaded data at offset 50 + (104 - 96)*4 = 82 where 104 =68h.

The trigger flag's offset (data_array_offset) can be obtain from the formula
Offset = cmd(hex) - 60 (hex)
Or
Offset = cmd (dec) – 96 (dec)
Or
simple mapping

Cmd(h)	trigger flag offset
60	0
61	1
--	--
68	8
---	----
88	40

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
CMD_P1N11_68,	DA_P1N11,	8,	SC_P1N11,	Wrbx,	1,	68

FieldServer will update this memory location automatically when remote device (scada system) sends data to FieldServer using 68h command to write to Smart Controller.

Write on change. When the value of the 9th element of the array called DA_P1N11 changes then this command will be trigger to write data to Smart Controller (SC_P1N11).

Always set to 1 for Wrbx commands. This means FieldServer will trigger this command only upon updation of data_array_offset location in Data Array DA_P1N11.

This is the command that will be triggered to write data to smart Controller SC_P1N11. Four data values will be taken from offset 82 in Data Array DA_P1N11.
Data Offset will calculated as
50 + (cmd (dec) - 96)*4
This can be replaced with its English name "**Set All Desired Box Temps"**

4.4.11. Map Descriptor Example 8 – Write Data Strategy 2

Whenever the upstream device updates the FieldServer with any or all of the four values for any write command, the FieldServer will initiate the command to write to the target device. If the FieldServer is updated with four values at the same time then all four values will be written to the target device at once and data integrity is ensured. In a case where all four values were not updated, the write will include some old values and data integrity may in **danger**.

In this example, FieldServer will watch four items at offset 82,83,84 and 85 and when any or all of these items is updated, then the HcscII driver will issue a command 68h with all these items as data values.

Map_Descriptor_Name, CMD_P1N11_68,	Scan_Interval, 1.0,	Data_Array_Name, DA_P1N11,	Data_Array_Offset, 82,	node_name, SC_P1N11,	Function, Wrbx,	Length, 4,	HcscII_cmd 68

Data values will be taken from this offset in Data Array DA_P1N11.
Offset 82 has been calculated from following formula.
50+(cmd (dec) -96)*4
in this example cmd=68h = 104 dec

Write On update. FieldServer will write values to Smart Controller when any or all values at given offset are updated.

Always set to 4 for Wrbx commands with this strategy. This means FieldServer will write four data values to Smart Controller in a single write command.

This is the command that will be used to write data to smart Controller SC_P1N11.
"Set All Desired Box Temps"

5. Configuring the FieldServer as a Heatcraft Smart Controller II Server

FieldServer with Heatcraft Smart Controller II (HCSCII) Driver can act as a server. For a detailed discussion on FieldServer configuration, please refer to the instruction manual for the FieldServer. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" files on the driver diskette).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Heatcraft Smart Controller II Driver Client

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Heatcraft Smart Controller II Driver communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual node(s) needs to be declared in the "Server Side Nodes" section, and the data to be provided to the clients needs to be mapped in the "Server 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. Server Side Connection Descriptors

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Server_Hold_Timeout*	Specifies time FieldServer will reserve server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s
Port	Specify which port the device is connected to the FieldServer	P1-P8[4]
Baud*	Specify baud rate	110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, None, Mark, Space
Data_Bits*	Specify data bits	7, 8
Stop_Bits*	Specify stop bits	1
Protocol	Specify protocol used	HcscII , Smart-Controller
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, None
hcscII_hub_connected	Specify if hub connected	YES/NO

[4] 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

```
//      Server Side Connections

Connections
Port,           Protocol,       Baud,           hcscII_hub_connected
P8,             hcscII,         38400,          YES
```

5.2. Server Side Node Descriptors

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	This commonly used parameter is not required by this driver.	11, 12, 13, 14 21, 22, 23, 24 31, 32, 33, 34 (all in Hex) 41, 42, 43, 44 only - anything else will be changed by the driver to 11 (hex)
Protocol	Specify protocol used	HcscII , Smart-Controller

Example

```
//  Server Side Nodes

Nodes
Node_Name       Protocol,       Port
SC_P1N11,       HcscII,         P1
SC_P2N11,       HcscII,         P2
```

5.3. Server Side Map Descriptors

5.3.1. 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 “Data Array” section above
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in “Data Array” section above
Function	Function of Server Map Descriptor	Server

5.3.2. Driver Specific 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	Register, Coil, AI, DI
Length	Length of Map Descriptor	1 - 1000
Address	Starting address of read block	40001, 30001, etc
Node_Name	Name of Node to fetch data from	One of the node names specified in “Client Node Descriptor” above
Data_Type	Not used	Not used
Length	Length of Map Descriptor	1 – 25000 for logged error/alarm length >4802 for logged data > 24747 else >606
Address	This driver does not use this commonly used parameter.	
HcscII_cmd	This field is mandatory. It tells the driver that what query or command is required.	60-FE (hexadecimal numbers) *or* English name *or* one keyword for group of commands *or* EVERYTHING *or* ERR_ALARM_LOG *or* DATA_LOG_? See Appendix A
hcscII_simulation	Used only to send Password Challenge to client	YES/ NO

5.3.3. Timing Parameters

Column Title	Function	Legal Values
Scada_Hold_Timeout	Specifies time server side waits before responding to client that node is offline on FieldServer client side.	>1.0s

5.3.4. Map Descriptor Example 1 - Everything.

The Map Descriptor will serve every thing other than logged items. It will serve a client using DA_P1N11 Data Array. Examples 3 and 4 demonstrate how to serve logged items.

```
Map_Descriptor_Name,    Data_Array_Name,    Data_Array_Offset,    node_name,    Function,    Length,    HcscII_cmd
SMD_P1N11,              DA_P1N11,           0,                    SC_P1N11,     Server,      700,       EVERYTHING
```

Status data is served and data for write will be store in this Data Array. Each Data Array represents one smart controller

This is Node declared in node sections and each node will represents a one Smart Controller device and its address.

You could use the synonyms *passive* or *server*.

The Length and offset collectively tell the driver on which portion it should act like a server.
Mostly length will be the same as that of length of Data Array defined in this Map Descriptor.
So the length of Map Descriptor or in turn length of Data Array must be long enough to emulate the whole memory area as in Smart Controller to support all required commands.

Here EVERYTHING means one Map Descriptor will fully emulate as one Smart Controller device and it will support all the commands supported by HcscII driver in server mode.

5.3.5. Map Descriptor Example 2 – FieldServer Emulates a portion of Smart Controller Device

In this example the FieldServer will serve suction temperatures for subsystems #1 and suction pressures for subsystem #1only.

```
Map_Descriptor_Name,    Data_Array_Name,    Data_Array_Offset,    node_name,    Function,    Length,    HcscII_cmd
SMD1_P1N11,             DA_P1N11,           398,                  SC_P1N11,     Server,      4,         Read Sub #1 Suction Temps
SMD2_P1N11,             DA_P1N11,           430,                  SC_P1N11,     Server,      4,         Read Sub #1 Suction Pressures
```

Status data is served from this Data Array.

Starting place for latest data available to upstream device.
Offset =
50+ (cmd(dec)-96)*4

The node name is declared in the Node Section of the .CSV file I represents a one Smart Controller device and its address.

The length is always 4 as there are 4 values associated with any HcscII messages.

English names for command numbers B7 and BF respectively. See Appendix A.1.

5.3.6. Map Descriptor Example 3.

This example illustrates a Map Descriptor used to serve logged error/alarm data. For more detail see Appendix A

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
SMD1_P1N11,	Da_err_alm_data,	0,	SC_P1N11,	Server,	4804 ,	ERR_ALARM_LOG

5.3.7. Map Descriptor Example 4.

This example illustrates a Map Descriptor used to serve logged data for all four sub systems – see Appendix A

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	node_name,	Function,	Length,	HcscII_cmd
SMD1_P1N11,	Da_log_data1,	0,	SC_P1N11	Server,	25000,	DATA_LOG_1
SMD2_P1N11,	Da_log_data2,	0,	SC_P1N11	Server,	25000,	DATA_LOG_2
SMD3_P1N11,	Da_log_data3,	0,	SC_P1N11	Server,	25000,	DATA_LOG_3
SMD4_P1N11,	Da_log_data4,	0,	SC_P1N11	Server,	25000,	DATA_LOG_4

Note. Examples 1, 3 and 4 collectively allow the FieldServer to simulate one complete Smart Controller.

Appendix A. Advanced Topics

Appendix A.1. Protocol Commands

The following list of commands has been implemented in the Heatcraft Smart Controller II Driver. The tables list the keywords and command numbers (hex) that may be used in the .CSV files to specify the HcscII_cmd parameter.

General Commands:
These commands are used to start communication. On detection of a DTR signal on the serial connection, the Smart Controller issues a "Password Challenge" command. Once the correct information is received from the FieldServer, the data on the Smart Controller is accessible for reading by the FieldServer. All System Parameter Commands (write commands) will be ignored by the Smart Controller until the "Validate Password" command is received from the FieldServer. The "Set Password" command can be used to override the Smart Controller default password.

Cmd(hex)	Cmd (Keywords)	Legal Values
60	Set All Sub Operating Modes	0 = "Off"; 1 = "Cooling"; 3 = "Defrost"; 4 = "Drain"; No change if > 7 (such as 0FFH)
61	Password Challenge	
62	Validate Password	
63	Set Password	
64…67		Spare

System Parameter Commands
These commands are used to write data to the Smart Controller device.

Cmd (Hex)	Cmd (keyword)	Legal Values
68	Set All Desired Box Temps	-30 to +70F
69	Set All Defrost Override Times	10 through 200 minutes
6A	Set All Alarm High Limit Temps	-40 to +80F
6B	Set All Alarm Low Limit Temps	-40 to +80F
6C	Set All Alarm Duration Times	2 to 120 minutes
6D	Set All Refrigerant Types	"R22" = 0, "R404" = 1, "R507" = 2
6E	Set All NV Modes	Bit 0 = Deg. F/C, Bit 1 = 12/24 Bit 2 = Normal / Demand Defrost Bit 4 = Air / Electric Def. Bit 6 = service off / service on Bit 7 = Unlock / Lock Param.
6F	Set Sub #1 Desired Superheats	+4 to +20 F
70	Set Sub #2 Desired Superheats	+4 to +20 F
71	Set Sub #3 Desired Superheats	+4 to +20 F
72	Set Sub #4 Desired Superheats	+4 to +20 F
73	Set Sub #1 Defrost Termination Temps	+40 to +100 F
74	Set Sub #2 Defrost Termination Temps	+40 to +100 F
75	Set Sub #3 Defrost Termination Temps	+40 to +100 F

Cmd (Hex)	Cmd (keyword)	Legal Values
76	Set Sub #4 Defrost Termination Temps	+40 to +100 F
77	Set Sub #1 Defrost Start Times 1 to 4	10 to 235, or 255 (none)
78	Set Sub #1 Defrost Start Times 5 to 8	10 to 235, or 255 (none)
79	Set Sub #1 Defrost Start Times 9 to 12	9 to 12: 10 to 235, or 255 (none)
7A	Set Sub #2 Defrost Start Times 1 to 4	1 to 4: 10 to 235, or 255 (none)
7B	Set Sub #2 Defrost Start Times 5 to 8	5 to 8: 10 to 235, or 255 (none)
7C	Set Sub #2 Defrost Start Times 9 to 12	9 to 12: 10 to 235, or 255 (none)
7D	Set Sub #3 Defrost Start Times 1 to 4	1 to 4: 10 to 235, or 255 (none)
7E	Set Sub #3 Defrost Start Times 5 to 8	5 to 8: 10 to 235, or 255 (none)
7F	Set Sub #3 Defrost Start Times 9 to 12	9 to 12: 10 to 235, or 255 (none)
80	Set Sub #4 Defrost Start Times 1 to 4	1 to 4: 10 to 235, or 255 (none)
81	Set Sub #4 Defrost Start Times 5 to 8	5 to 8: 10 to 235, or 255 (none)
82	Set Sub #4 Defrost Start Times 9 to 12	9 to 12: 10 to 235, or 255 (none)
83-88	83 – 88	Spares

System Parameter requests

These commands are used to read system parameters (readable and write memory area).

Cmd(hex)	Cmd (Keyword)	Data Returned
89	Read All Desired Box Temps	-30 to +70 F
8A	Read All Defrost Override Times	10 through 200 minutes
8B	Read All Alarm High Limit temps	-40 to +80 F
8C	Read All Alarm Low Limit temps	-40 to +80 F
8D	Read All Alarm Duration times	2 to 120 minutes
8E	Read All Refrigerant Types	"R22" = 0, "R404" = 1, "R507" = 2
8F	Read All NV Modes	Bit 0 = Deg. F/C, Bit 1 = 12/24 Bit 2 = Normal / Demand Defrost Bit 4 = Air / Electric Def. Bit 6 = Service Off/ Service On Bit 7 = Unlock / Lock Param.
90	Read Sub #1 Desired Superheats	+4 to +20 F
91	Read Sub #2 Desired Superheats	+4 to +20 F
92	Read Sub #3 Desired Superheats	+4 to +20 F
93	Read Sub #4 Desired Superheats	+4 to +20 F
94	Read Sub #1 Defrost Termination Temps	+40 to +100 F
95	Read Sub #2 Defrost Termination Temps	+40 to +100 F
96	Read Sub #3 Defrost Termination Temps	+40 to +100 F
97	Read Sub #4 Defrost Termination Temps	+40 to +100 F
98	Read Sub #1 Defrost Start Times 1 to 4	10 to 235, or 255 (none)
99	Read Sub #1 Defrost Start Times 5 to 8	10 to 235, or 255 (none)
9A	Read Sub #1 Defrost Start Times 9 to 12	10 to 235, or 255 (none)
9B	Read Sub #2 Defrost Start Times 1 to 4	10 to 235, or 255 (none)
9C	Read Sub #2 Defrost Start Times 5 to 8	10 to 235, or 255 (none)
9D	Read Sub #2 Defrost Start Times 9 to 12	10 to 235, or 255 (none)
9E	Read Sub #3 Defrost Start Times 1 to 4	10 to 235, or 255 (none)
9F	Read Sub #3 Defrost Start Times 5 to 8	10 to 235, or 255 (none)
A0	Read Sub #3 Defrost Start Times 9 to 12	10 to 235, or 255 (none)

Cmd(hex)	Cmd (Keyword)	Data Returned
A1	Read Sub #4 Defrost Start Times 1 to 4	10 to 235, or 255 (none)
A2	Read Sub #4 Defrost Start Times 5 to 8	10 to 235, or 255 (none)
A3	Read Sub #4 Defrost Start Times 9 to 12	10 to 235, or 255 (none)
A4 – A9	A4 – A9	Spares

System Data Requests
Used to request system data (read only).

Cmd(hex)	Cmd (Keyword)	Data Returned
AAh	Read All Master Addresses And Units	Units in 3 MSBits, Master addresses in 5 LSBits
ABh	Read Main And IO Processor Firmware Ver	Major and Minor revision numbers (major *10 + minor)
ACh	Read Sub #1 Accumulated Comp Run Time	Hours * 10, Seconds * .5
ADh	Read Sub #2 Accumulated Comp Run Time	Hours * 10, Seconds * .5
AEh	Read Sub #3 Accumulated Comp Run Time	Hours * 10, Seconds * .5
AFh	Read Sub #4 Accumulated Comp Run Time	Hours * 10, Seconds * .5
B0h	Read Day Number	(0-255)
B1h	Read Master Mode and Status	"Mode" - Operating mode in 3 LSBits 0=Off; 1=Cooling; 2 = Pumpdown; 3=Defrost, 4=Draining; 5=Delay; 6=Test; 7=Service) Error Status (bit 3); Alarm Status (bit 4); sub-master control (bit 6); Smart Controller present (bit 7)
B2h	Read Master ModeX	Master extended status values "ModeX": optional sensor(s) attached and defrost mode bits (bit packed)
B3h	Read Master Alarm Codes	Master alarm codes "AlrBits" for: room temperature too high or room temperature too low (bit packed) bit 0 set = "Too High" bit 1 set = "Too Low bit 2 set = "Failure to Start" bit 3 set = "Sensor Failure"
B4h	Read Master Accumulated Comp Cycles	0 to 255
B5h	Read Master Room Temps	-55 to +125 F (-127= "Unknown", -128 = "Comm Error")
B6h	Read Master Outdoor Temps	-55 to +125 F
B7h	Read Sub #1 Suction Temps	-55 to +125 F
B8h	Read Sub #2 Suction Temps	-55 to +125 F
B9h	Read Sub #3 Suction Temps	-55 to +125 F

Cmd(hex)	Cmd (Keyword)	Data Returned
BAh	Read Sub #4 Suction Temps	-55 to +125 F
BBh	Read Sub #1 Saturated Suction Vapor Temps	-55 to +125 F
BCh	Read Sub #2 Saturated Suction Vapor Temps	-55 to +125 F
BDh	Read Sub #3 Saturated Suction Vapor Temps	-55 to +125 F
BEh	Read Sub #4 Saturated Suction Vapor Temps	-55 to +125 F
BFh	Read Sub #1 Suction Pressures	0 to 150 PSIA
C0h	Read Sub #2 Suction Pressures	0 to 150 PSIA
C1h	Read Sub #3 Suction Pressures	0 to 150 PSIA
C2h	Read Sub #4 Suction Pressures	0 to 150 PSIA
C3h	Read Sub #1 Superheats	-128 to +127 F
C4h	Read Sub #2 Superheats	-128 to +127 F
C5h	Read Sub #3 Superheats	-128 to +127 F
C6h	Read Sub #4 Superheats	-128 to +127 F
C7h	Read Sub #1 Defrost Temps	-128 to +127 F
C8h	Read Sub #2 Defrost Temps	-128 to +127 F
C9h	Read Sub #3 Defrost Temps	-128 to +127 F
CAh	Read Sub #4 Defrost Temps	-128 to +127 F
CBh	Read Sub #1 EXV Positions	0 to 255
CCh	Read Sub #2 EXV Positions	0 to 255
CDh	Read Sub #3 EXV Positions	0 to 255
Ceh	Read Sub #4 EXV Positions	0 to 255
CFh	Read Sub #1 EXV Step Sizes	0 to 255
D0h	Read Sub #2 EXV Step Sizes	0 to 255
D1h	Read Sub #3 EXV Step Sizes	0 to 255
D2h	Read Sub #4 EXV Step Sizes	0 to 255
D3h	Read Sub #1 AC Input Voltages	0*5 to 31.4*5 Vac
D4h	Read Sub #2 AC Input Voltages	0*5 to 31.4*5 Vac
D5h	Read Sub #3 AC Input Voltages	0*5 to 31.4*5 Vac
D6h	Read Sub #4 AC Input Voltages	0*5 to 31.4*5 Vac
D7h	Read Sub #1 Last Defrost Elapsed Times	0 to 255 minutes
D8h	Read Sub #2 Last Defrost Elapsed Times	0 to 255 minutes
D9h	Read Sub #3 Last Defrost Elapsed Times	0 to 255 minutes

Cmd(hex)	Cmd (Keyword)	Data Returned
DAh	Read Sub #4 Last Defrost Elapsed Times	0 to 255 minutes
DBh	Read Sub #1 Error Codes	room temp sensor (bit 0) defrost temp sensor (bit 1) suction temp sensor (bit 2) suction pressure sensor (bit 3) outdoor temp sensor (bit 4) low superheat temperature (Bit 5) compressor shutdown (bit 6)
DCh	Read Sub #2 Error Codes	
DDh	Read Sub #3 Error Codes	
DEh	Read Sub #4 Error Codes	
DFh	Read Sub #1 Firmware Versions	major and minor revisions (major*10 + minor)
E0h	Read Sub #2 Firmware Versions	major and minor revisions (major*10 + minor)
E1h	Read Sub #3 Firmware Versions	major and minor revisions (major*10 + minor)
E2h	Read Sub #4 Firmware Versions	major and minor revisions (major*10 + minor)
DFh	Read Sub #1 Firmware Versions	-55 to +125 F
E0h	Read Sub #2 Firmware Versions	-55 to +125 F
E1h	Read Sub #3 Firmware Versions	-55 to +125 F
E2h	Read Sub #4 Firmware Versions	-55 to +125 F
E3h	Read Sub #1 Spare Temp	-55 to +125 F
E4h	Read Sub #2 Spare Temp	-55 to +125 F
E5h	Read Sub #3 Spare Temp	-55 to +125 F
E6h	Read Sub #4 Spare Temp	-55 to +125 F
E7	E7	Spare
E8	E8	Spare
E9	E9	Spare
EA	EA	Spare

Requests for logged errors/alarms
Both commands should be used in pairs.

Cmd(hex)	Data Returned
F4	Current error/alarm que index and length
EB	Selected error/alarm logged record

Requests for logged data
Each group of three commands should be used in conjunction.

Cmd(hex)	Data Returned
F5	Current Subsystem #1 que index and length
EC	Selected Subsystem #1 log data (part 1 of selected record number)

ED	Selected Subsystem #1 log data (part 2 of selected record number)
F6	Current Subsystem #2 que index and length
EE	Selected Subsystem #2 log data (part 1 of selected record number)
EF	Selected Subsystem #2 log data (part 2 of selected record number)
F7	Current Subsystem #3 que index and length
F0	Selected Subsystem #3 log data (part 1 of selected record number)
F1	Selected Subsystem #3 log data (part 2 of selected record number)
F8	Current Subsystem #4 que index and length
F2	Selected Subsystem #4 log data (part 1 of selected record number)
F3	Selected Subsystem #4 log data (part 2 of selected record number)
F9-FE	Spares

Appendix A.2. Special Keywords and descriptions

These are special commands that have been developed by the driver to read multiple parameters independent of the configuration settings specified in the .CSV file.

"COMMONSET" - Reads all of the following parameters	
All Master Addresses And Units	AA
All Refrigerant Types	8E
All NV Modes	8F
Master Mode and Status	B1
Master ModeX	B2
Master Alarm Codes	B3
Master Accumulated Comp Cycles	B4
Master Room Temps	B5
Master Outdoor Temps	B6
All Desired Box Temps	89
All Defrost Override Times	8A
All Alarm High Limit Temps	8B
All Alarm Low Limit Temps	8C
All Alarm Duration Times	8D
Main And IO Processor Firmware Ver	AB
Day Number	B0
"Only for S1" - Reads all parameters and data for sub system #1 not included in COMMONSET.	
"SYSTEM#1" single command which polls for data from subset1#1 and subset2#1	
"SUBSET1#1" polls for the following data from the first subset only[5]:	
Sub #1 Error Codes	DB
Sub #1 Accumulated Comp Run Time	AC
Sub #1 Desired Superheats	90
Sub #1 Defrost Termination Temps	94
Sub #1 Saturated Suction Vapor Temps	BB
Sub #1 Suction Pressures	BF
Sub #1 Suction Temps	B7

[5] Updating time can be controlled seperately for subsets.

Sub #1 Superheats	C3
Sub #1 Defrost Temps	C7
"SUBSET2#1" polls for the following data from the second subset only[4]	
Sub #1 EXV Positions	CB
Sub #1 EXV Step Sizes	CF
Sub #1 Defrost Start Times 1 to 4	98
Sub #1 Defrost Start Times 5 to 8	99
Sub #1 Defrost Start Times 9 to 12	9A
Sub #1 Last Defrost Elapsed Times	D7
Sub #1 Spare Temp	E3
Sub #1 AC Input Voltages	D3
Sub #1 Firmware Versions	DF
Only for S2: - Reads all parameters and data for sub system #2, not included in COMMONSET.	
"SYSTEM#2" single command which polls for data from subset1#2 and subset2#2	
"SUBSET1#2" polls for the following data from the first subset only[4]:	
Sub #2 Error Codes	DC
Sub #2 Accumulated Comp Run Time	AD
Sub #2 Desired Superheats	91
Sub #2 Defrost Termination Temps	95
Sub #2 Saturated Suction Vapor Temps	BC
Sub #2 Suction Pressures	C0
Sub #2 Suction Temps	B8
Sub #2 Superheats	C4
Sub #2 Defrost Temps	C8
"SUBSET2#2" polls for the following data from the second subset only[4].	
Sub #2 EXV Positions	CC
Sub #2 EXV Step Sizes	D0
Sub #2 Defrost Start Times 1 to 4	9B
Sub #1 Defrost Start Times 5 to 8	9C
Sub #2 Defrost Start Times 9 to 12	9D
Sub #2 Last Defrost Elapsed Times	D8
Sub #2 Spare Temp	E4
Sub #2 AC Input Voltages	D4
Sub #2 Firmware Versions	E0
Only for S3: - Reads all parameters and data for sub system #3, not included in COMMONSET.	
"SYSTEM#3" single command which polls for data from subset1#3 and subset2#3	
"SUBSET1#3" polls for the following data from the first subset only[4]:	
Sub #3 Error Codes	DD
Sub #3 Accumulated Comp Run Time	AE
Sub #3 Desired Superheats	92
Sub #3 Defrost Termination Temps	96
Sub #3 Saturated Suction Vapor Temps	BD
Sub #3 Suction Pressures	C1

[4] Updating time can be controlled seperately for subsets.

Sub #3 Suction Temps	B9
Sub #3 Superheats	C5
Sub #3 Defrost Temps	C9
"SUBSET2#3" polls for the following data from the second subset only[4]..	
Sub #3 EXV Positions	CD
Sub #3 EXV Step Sizes	D1
Sub #3 Defrost Start Times 1 to 4	9E
Sub #3 Defrost Start Times 5 to 8	9F
Sub #3 Defrost Start Times 9 to 12	A0
Sub #3 Last Defrost Elapsed Times	D9
Sub #3 Spare Temp	E5
Sub #3 AC Input Voltages	D5
Sub #3 Firmware Versions	E1
Only for S4: - Reads all parameters and data for sub system #4, not included in COMMONSET.	
"SYSTEM#4" single command which polls for data from subset1#4 and subset2#4	
"SUBSET1#4" polls for the following data from the first subset only[4]:	
Sub #4 Error Codes	DE
Sub #4 Accumulated Comp Run Time	AF
Sub #4 Desired Superheats	93
Sub #4 Defrost Termination Temps	97
Sub #4 Saturated Suction Vapor Temps	BE
Sub #4 Suction Pressures	C2
Sub #4 Suction Temps	BA
Sub #4 Superheats	C6
Sub #4 Defrost Temps	CA
"SUBSET2#4" polls for the following data from the second subset only[4].	
Sub #4 EXV Positions	CE
Sub #4 EXV Step Sizes	D2
Sub #4 Defrost Start Times 1 to 4	A1
Sub #4 Defrost Start Times 5 to 8	A2
Sub #4 Defrost Start Times 9 to 12	A3
Sub #4 Last Defrost Elapsed Times	E6
Sub #4 Spare Temp	DA
Sub #4 AC Input Voltages	D6
Sub #4 Firmware Versions	E2
ERR_ALARM_LOG - Reads error/alarm log data. F4 & EB commands handled in conjunction.	
The driver reads logged data on the Smart Controller starting with 0th record and stores it internally until it reaches the latest record. At this point, the driver updates the FieldServer putting the latest record in the first position. When the driver and FieldServer are synchronized, the driver updates the FieldServer with the latest records as they become available.	
DATA_LOG_?- Reads logged data for sub systems from Smart Controller	
DATA_LOG_1 captures logged data for system 1 DATA_LOG_2 captures logged data for system 2	

[4] Updating time can be controlled seperately for subsets.

DATA_LOG_3 captures logged data for system 3
DATA_LOG_4 captures logged data for system 4
The driver reads logged data on the Smart Controller starting with 0th record and stores it internally until it reaches the latest record. At this point, the driver updates the FieldServer putting the latest record in the first position. When the driver and FieldServer are synchronized, the driver updates the FieldServer with the latest records as they become available..

Appendix A.3. Storage Information

Most of the operations (commands) contain four data values, therefore the driver reserves four memory locations for each command.

Starting offsets for each command are determined using the following formulae:

Offset = cmd (hex) - 60 (hex) Or Offset = cmd (dec) - 60 (dec)

Note: When selecting data format consider that you may want to update the target device with -ve and fractional values.

As a client

Suppose client uses the following Data Arrays to read or write every thing with one Smart Controller.

DA_P1N11, Float, 700
DA_ERRALM_LOG, Float, 5000
DA_DATA_LOG_1, Float, 25000

DA_P1N11
If this Data Array will be used for reading and writing parameters to Smart Controller for every thing (limited use with logged items).
Here length is 686

Offsets	Cmd (hex)	Description
0 -39	60 – 88	Trigger flags for all write (Set) commands.
40- 49	Not used internally	
50 - 213	60 - 88	User determined memory locations for data written to Smart Controller by FieldServer.
214 - 690	89 - FF	Memory locations for storing the data for indicated commands

Defrost times

For commands (77 – 82 h and 98 – A3)
Each defrost time must fall on an hour or thirty – minute boundary. Valid values for these commands should be within 1.0 – 23.5 hr . Smart Controller may ignore setting values if out of range.

Accumulated compressor run time (AC - AF)
This time will be stored in seconds at corresponding starting offset of each command.

AC input voltages: (D3- D6)
FieldServer will store Ac voltage for each subsystem 0.0 - 31.4 Vac

Versions: (AB , DF – E2)
Firmware, Main and I/O versions stored as major.minor eg if major is 2 and Minor is 5 then FieldServer will store this as 2.5

Error/alarm data and logged data for sub-systems. (Client)

The driver starts reading the data starting with the 0th record, and stores all the records internally until it has read the latest record. The driver them updates the FieldServer with the records starting with the latest record. Once the FieldServer and the driver are synchronized, the FieldServer continues to be updated with the latest record.

The error/alarm log Data Array will be formatted as follows:

Offset	Information Stored
0	Record number of latest record
1	Total number of records
2	Day number (0-255) on which record was written.
3	Time (hrs) – Time boundary is 6mins, thus 22.5 represents 22h30 and 15.8 represents 15h48.
4	unit number or system numbers
5	Packed bits
6…10	Next Record

Each record is composed of 4 elements. The latest record is always stored in offsets 2 thru 5.

The data log Data Array for sub-systems will be formatted as follows:

Offset	Information Stored
0	Record number of latest record
1	Total number of records
2	day number (0-255)on which record was written.
3	Time in hrs (10.5 means 10 hr and 30 minutes)
4	Operating mode
5	Room temperature
6	EXV position
7	Superheat
8	Suction pressure
9	Accumulated compressor cycles since mid night
10…17	Next Record

Each record is composed of 8 elements. The latest record is always stored in offsets 2 thru 9.

Error/alarm data and logged data for sub-systems (Server)

The first element in the Data Array indicates the record number of the next record to be written and the second element (Length) indicates the number of records available. For error/alarm log each record has four elements. For logged data for sub-systems each record has eight elements

Appendix B. Driver Error Messages

Some configuration errors might produce an error every time a poll is generated. This will fill the error buffer quickly and not add any clarity. For this reason the driver suppresses subsequent similar messages. Thus it is possible for the same error produced by multiple Map Descriptors to produce only one error messages and subsequent error messages can be seen on the driver message screen.
Note : In the actual message you will observe that %d has been replaced by an integer , %s by text indicating a Data Array name or Map Descriptor name and %x by two hex characters.

Message	Description and Action
HCSCII:#01 Err. Insufficient space Reqd/Exist <%d>/<%d> da <%s>	The defined Data Array Length is too short to store data from Smart Controller. Increase Data Array Length in the CSV file.
HCSCII:#02 Err. Insufficient space Reqd/Exist <%d>/<%d> da <%s>	The defined Data Array Length is too short to store data from Upstream node. Increase Data Array Length in the CSV file.
HCSCII:#03 ERR. Insufficient space Reqd/Exist <%d>/<%d> da <%s>Logging err/alarm stopped.	The defined Data Array Length is too short to store logged errors/alarms. Increase Data Array Length in the CSV file.
HCSCII:#04 ERR. Insufficient space Reqd/Exist <%d>/<%d> da <%s>Logging data stopped.	The defined Data Array Length is too short to store logged data for sub-systems. Increase Data Array Length in the CSV file.
HCSCII:#05 ERR. Err/Alarm Table-length/Max <%d>/<%d> MD <%s>	Indicates that length of the err/alarm table in Smart Controller is more than 4000. Error/alarm log capturing will be stopped.
HCSCII:#06 ERR. Err/Alarm Insufficient memory on FS MD <%s>	There is not enough free memory on FieldServer to store error/alarm log internally. Call for support.
HCSCII:#07 ERR. Data Log Table-length/Max <%d>/<%d> MD <%s>Logging data stopped.	Indicates that length of the data logging table in Smart Controller is more than 4000. Error/alarm log capturing will be stopped.
HCSCII:#08 FYI. Data log Insufficient memory on FS MD <%s>.	There is not enough free memory on FieldServer to store data logs internally. Call for support.Call for support
HCSCII:#11 Err. Insufficient space Reqd/Exist <%d>/<%d> da <%s>	The defined Data Array Length is too short to store data from Upstream node. Increase Data Array Length in the CSV file.
HCSCII:#12 FYI Device addr <%X> at port <%d> can not be Communicated!	FieldServer is connected to Smart Controller device and no Map Descriptor has been defined to communicate with this device. Add some Map Descriptors to configuration file if communication is desired with this device.
HCSCII:#13 FYI Device at port <%d> can not be Communicated!	Similar to #12 but now only this is the device on this connection either hub is not connected.
HCSCII:#14 Err. Insufficient space Reqd/Exist <%d>/<%d>	FieldServer is acting as a server and is responding to logged data query but Data Array is not enough long.

Message	Description and Action
da <%s>	Increase the length of corresponding Data Array.
HCSCII:#21 Err. Insufficient space Reqd/Exist <%d>/<%d> da <%s>	Write operation with a command for which Data Array does not contain data. Write operation will be cancelled. Increase the Data Array length to be required.
HCSCII:#31 FYI Command <%2X> Data Items<%d> Not in Range" Data : %d %d %d %d	FieldServer is going to set parameters at Smart Controller but found few or all values are out of range. Smart Controller may ignore invalid values, check with protocol command set for valid range. The shown values will be scaled.
hcscII:#32 FYI. You could use an Array called <%s> to expose diagnostic info.	You could add an array to expose some predefined stats see Appendix B.1.

Appendix B.1. Driver Stats

The following lines may be added to the configuration file to expose certain special stats.
//

```
//   HcscII Stats
//
Data_Arrays
Data_Array_Name,          Data_Format ,          Data_Array_Length
hcscII-stats      ,       UINT32      ,          12000
```

To obtain information on connection port only:
The offset for the port is calculated according to the formula: Offset_port = (port-1)*10
Thus: For P1 Offset_port = (1-1)*10 = 0
For P7 Offset_port = (7-1)*10 = 60

For stat no 0

Stat_number	Stats	Description
0	HCSCII_MODEM_MSG	Number of modem msg received on this port

To get modem message number for port P7
Offset = offset_port + stat_number = 60+0 = 60

To obtain information on individual nodes on a connection port:
Offset formula:
Fixed_offset + (port-1)*Mx_Nodes*stats_per_node +node_num*stats_per_node + stats_number

Where: Fixed_offset = 100
Port = 1 for P1, 2 for P2 …….8 for P8
Mx_nodes = Maximum nodes per connection = 16
Stats_per_node = 25
Node_num is related to the node_id declared in node section as follows:

Node_id	11	12	13	14	21	22	23	24	31	32	33	34	41	42	43	44
Node_num	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Stats_number can be determined from the following table:

Stat_number	Stats	Description
0	HCSCII_PWD_CHALLENGE	Number of times node received and responded to password challenge message.
1	Not used	Not used
2	HCSCII_ERRALM_QUE_INDEX	Smart Controller's que index for err/alarm log
3	HCSCII_ERRALM_QUE_LENGTH	Smart Controller's que length for err/alarm log
4	HCSCII_DATALOG_1_QUE_INDEX	Smart Controller's que index for system 1 data log.
5	HCSCII_DATALOG_1_QUE_LENGTH	Smart Controller's que length for system 1 data log.
6	HCSCII_DATALOG_2_QUE_INDEX	Smart Controller's que index for system 2 data log.
7	HCSCII_DATALOG_2_QUE_LENGTH	Smart Controller's que length for system 2 data log.
8	HCSCII_DATALOG_3_QUE_INDEX	Smart Controller's que index for system 3 data log.
9	HCSCII_DATALOG_3_QUE_LENGTH	Smart Controller's que length for system 3 data log.
10	HCSCII_DATALOG_4_QUE_INDEX	Smart Controller's que index for system 4 data log.
11	HCSCII_DATALOG_4_QUE_LENGTH	Smart Controller's que length for system data log.
12	HCSCII_FS_ERRALM_QUE_INDEX	Number of err/alarm log records captured by FieldServer (usable before synchronization).
13	HCSCII_FS_DATALOG_1_QUE_INDEX	Same as above for data log system 1
14	HCSCII_FS_DATALOG_2_QUE_INDEX	Same as above for data log system 2
15	HCSCII_FS_DATALOG_3_QUE_INDEX	Same as above for data log system 3
16	HCSCII_FS_DATALOG_4_QUE_INDEX	Same as above for data log system 4

To see the number of records captured by the FieldServer for logged data for system 1 (Stat_number = 13); node_id 13 (node_num = 2); connected at Port P2 (port = 2)

Offset = 100 +(2-1)*16*25+2*25 +13 = 563

Appendix C. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
1/6/04	SSS		1.00	0	Initial release.
2/2/04	SSS		1.01	0	Restructured logged error/alarms and data, added section 6.5 driver stats and other cleanup.
2/25/04	JD		1.01	1	Formatting Changes
11/02/04	Meg		1.01	2	Start of complete overhaul of manual updating formatting and language – DUR0445. Connection diagram added – DUR0444
11/02/04	Meg	Meg	1.01	3	Further work on overhaul of manual updating formatting and language – DUR0445.
11/11/04	Meg	Meg	1.01	4	Further work in updating language in document. DUR0445.