A Sierra Monitor Company

Driver Manual

(Supplement to the FieldServer Instruction Manual)

FS-8700-74 Veeder-Root

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after January 2010

Driver Version:	1.01
Document Revision:	8

TABLE OF CONTENTS

1 VEEDER-ROOT DRIVER DESCRIPTION

The Veeder-Root Serial Driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 ports using Veeder-Root protocol as defined in Veeder Root Document 576013-635 Revision J. Since the data protocol is the same for the TLS350+ as for TLS350, it is assumed that the driver will support the TLS350+ but this has not been tested.

The FieldServer emulates a Client.

The Veeder-Root Serial Driver is a poll response driver. Only one query or command can be processed at a time.

A limited set of the queries and commands defined in the protocol specification have been implemented. The reason for the limitation is two-fold. Firstly, not all commands/queries will have any meaning to a Server device as they are principally defined to configure the Veeder-Root Device. Secondly some commands return very complex data sets which cannot be processed in a method suitable for loading into the FieldServer's Data Arrays.

The driver is capable of exposing its communications statistics which allows them to be monitored using a Server device. Thus it is possible to ensure that the data is valid and to generate alarms if communication problems arise.

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-07	RJ45 to DB25M connection adapter
FS-8915-10	UTP (Cat5) cable (7 foot) for RS-232 use

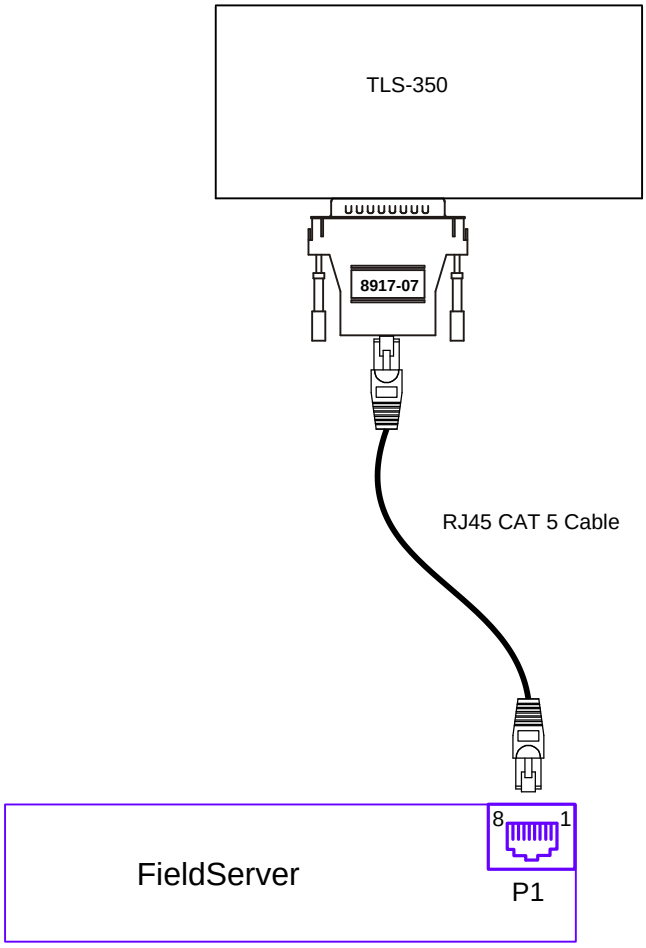
2.2 Provided by the Supplier of 3rd Party Equipment

2.2.1 Required 3rd Party Hardware

Part #	Description
	Veeder-Root Panel

3 HARDWARE CONNECTIONS

The FieldServer is connected to the Veeder-Root Panel as shown below.



Connector Pinouts

FS-X40 Function	RJ45 Pin#	DB25M Pin#	Color
RX	1	2	White
CTS	2	5	Brown
DSR	3	8	Yellow
GND	4	7	Green
DTR	6	20	Black
RTS	7	6	Orange
TX	8	3	Blue

3.1 Verifying comm. settings at the Veeder Root Panel

- From the front screen that shows the time/date, press MODE until "SETUP MODE" menu appears.
- Press FUNCTION until "COMMUNICATIONS SETUP" appears.
- Press STEP to get to "PORT SETTINGS" menu. Press ENTER to get into "PORT SETTINGS" menu.
- The "PORT SETTINGS" menu shows the following parameters:
  - Comm Board: 1 (RS-232)
  - Baud Rate:
  - Parity:
  - Stop Bit:
  - Data Length
  - Code:
- To scroll through the parameters, press the STEP button.
- To modify a parameter, press the STEP button until the parameter appears on the screen and push CHANGE to modify. Press ENTER, then follow the prompt by pressing STEP.
- If the operator "gets lost" in the menus, press MODE until the front screen showing time/date appears.

4 DATA ARRAY PARAMETERS

Data Arrays are "protocol neutral" data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

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, 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_Array_Format    , Data_Array_Length
DA_AI_01           , UInt16               , 200
DA_AO_01           , UInt16               , 200
DA_DI_01           , Bit                  , 200
DA_DO_01           , Bit                  , 200
```

5 CONFIGURING THE FIELDSERVER AS A VEEDER-ROOT 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.

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Veeder-Root Device.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Veeder-Root 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.

5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[1]
Baud*	Specify baud rate	30 - **9600**, standard baud rates only – (Vendor limitation)
Parity*	Specify parity	Even, Odd, **None**
Data_Bits*	Specify data bits	7, **8**
Stop_Bits*	Specify stop bits	**1,** 2
Protocol	Specify protocol used	Vroot, Veeder
Handshaking*	Specify hardware handshaking	**None**
Poll Delay*	Time between internal polls	0-32000 seconds, **1 second**

Example

```
//  Client Side Connections

Connections
Port    , Baud    , Parity    , Protocol    , Poll_Delay
P1      , 9600    , None      , Vroot       , 0.100s
```

[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.

5.2 Client Side Node Descriptors

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Protocol	Specify protocol used	Vroot, Veeder
Port	Specify which port the device is connected to the FieldServer.	P1-P8, R1-R2[2]
Route*	This parameter is only required if security is enabled on the Veeder-Root device. The route must be set to contain the 6 digits that form the security code. Refer to Appendix A.10	Decimal digits in the range 0 to 9 in the format a.b.c.d.e.f, **0**.

Example

```
// Client Side Nodes

Nodes
Node_Name          , Protocol   , Port
VR_Node1           , Vroot      , P1
```

5.3 Client Side Map Descriptor Parameters

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 Section 4
Data_Array_Location	Starting location in Data Array	0 to maximum specified in Section 4.
Function	Function of Client Map Descriptor	Rdbc, Wrbc, Wrbx

[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.

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 Section 5.2.
Data_Type	Data type	Refer to Appendix C.1 for supported Data Types.
Length	Length of Map Descriptor The length is used to tell the driver how many elements of the Data Array are under the control of the Map Descriptor. This is important because the driver uses the length to, for example, clear Data Array elements when a Tank/Sensor has no alarms.	1 – 1000 Refer to Appendix C.1 for required lengths for specific Data Types.
Address*[3]	The address is used to specify the Tank/Sensor/Input /Relay … Number to be used in the poll. For some data types the address has no meaning and should be set to 1. When zero is used as the value of the address then (in most cases) the driver polls for all Tanks/ Sensors belonging to that data type.	**0**, 1, 2, 3… Veeder-Root Tanks / Sensors … are numbered from 1. The maximum tank number is 12 and the maximum sensor number is 64.
Veed_Function	The Veed function number can be specified.	Specify using the notation 0xnnn where nnn is the function number, e.g. 0x901
DA_Byte_Name*	This is an optional Data Array that can be defined to store additional information. The information returned is specific to the Data Type. Refer to Appendix A.3 and 0 for examples using this parameter.	One of the Data Array names from Section 4, -
DA_Float_Name*	This is an optional Data Array that can be defined to store the Current value for the Configuration Data Type. Refer to Appendix A.3.	One of the Data Array names from Section 4, -

5.3.3 Timing Parameters

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

[3] See Appendix C.1 for Data Type, Address and Length Limitations

5.3.4 Map Descriptor Example 1 - Alarms.

This example provides one Map Descriptor to poll for alarm data and many Map Descriptors to store the response from the poll. One Map Descriptor is required for each tank/sensor/input/device defined for the system. The address of each of these passive Map Descriptors must correspond to the tank/sensor/input/device number. For example, where 5 tanks are defined as tanks 1,2,3,6,7 then 5 Map Descriptors with the data type 'Tank' are required and the address of each of these 5 Map Descriptors must be set to 1,2,3,6,7. The length parameter tells the driver how much array space is reserved for each tank/sensor for storing alarms.

Only the Map Descriptor used to generate the poll requires a scan interval.

The address must correspond to the tank/sensor/input/device number.
One Map Descriptor required for each tank/sensor/input/device.

```
Map_Descriptors
Map_Descriptor_Name   , Scan_Interval   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Address   , Length   , Data_Type
Poller                , 1.0s            , DA_101           , 0                   , Rdbc       , Node_A      , 1         , 30       , System Status
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Address   , Length   , Data_Type
Sys_alms01            , DA_S1             , 0                   , Passive    , Node_A      , 1         , 30       , System
Tank_Alms1            , DA_T1             , 0                   , Passive    , Node_A      , 1         , 30       , Tank
LiqSensor_Alm1        , DA_L1             , 0                   , Passive    , Node_A      , 1         , 30       , Liquid Sensor
VaporSens_Alm1        , DA_V1             , 0                   , Passive    , Node_A      , 1         , 30       , Vapor Sensor
Input_Alm1            , DA_I1             , 0                   , Passive    , Node_A      , 1         , 30       , Input
VolLine_Alm1          , DA_O1             , 0                   , Passive    , Node_A      , 1         , 30       , Volumetric Line
GWater_Alms1          , DA_G1             , 0                   , Passive    , Node_A      , 1         , 30       , Groundwater Sensor
TypeB_SensALm1        , DA_B1             , 0                   , Passive    , Node_A      , 1         , 30       , Type-B Sensor
Univ_SensAlm1         , DA_U1             , 0                   , Passive    , Node_A      , 1         , 30       , Universal Sensor
AutoDial_Alms1        , DA_F1             , 0                   , Passive    , Node_A      , 1         , 30       , Auto-Dial
MechDisp_Alm1         , DA_M1             , 0                   , Passive    , Node_A      , 1         , 30       , Mech. Dispenser
ElecDisp_Alm1         , DA_E1             , 0                   , Passive    , Node_A      , 1         , 30       , Elec. Dispenser
Product_Alms1         , DA_P1             , 0                   , Passive    , Node_A      , 1         , 30       , Product
PressLine_Alm1        , DA_R1             , 0                   , Passive    , Node_A      , 1         , 30       , Press. Line
External_Alm1         , DA_X1             , 0                   , Passive    , Node_A      , 1         , 30       , External
WPLLD_Alms1           , DA_W1             , 0                   , Passive    , Node_A      , 1         , 30       , WPLLD
```

Data is stored in the Data Arrays at the offset tabulated in Appendix C.1.3.

Passive Map Descriptors do not generate messages. They are used to store data returned by the system status poll.

The length parameter tells the driver how much array space must be reserved for storing alarms for a given tank/sensor/input/device.

When an alarm occurs for tank 1 the driver stores the alarm type by setting the array position corresponding to the alarm type to a non-zero value as well as setting the first element of the reserved space non-zero as a summary. Refer to Appendix C.1.3 for alarm type numbers and descriptions.

Example: Tank 1 has two alarms

- 02 = Tank Leak Alarm
- 05 = Tank Low Product Alarm
- The driver sets array DA_T1 as follows:

  Index=0 : Value=1 (Summary); Index=2 : Value=2 (Leak); Index=5 : Value=5 (Product)

5.3.5 Map Descriptor Example 2 - Inventory

This example illustrates a poll for Tank Inventory Data. One Map Descriptor is required for each tank. The address should correspond to the tank number. The Length parameter tells the driver how much storage space to reserve for storing the inventory data. The inventory data is stored as follows.

Index	Contents
0.	Product Code (Example Array value = 49 indicates product code = ASCII '1')
1.	Tank Status (Bit 1 - (LSB) Delivery in Progress, Bit 2 - Leak Test in Progress, Bit 3 - Invalid Fuel Height Alarm (MAG Probes Only))
2.	Volume
3.	TC Volume
4.	Ullage
5.	Height
6.	Water
7.	Temperature
8.	Water Volume

.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Scan_Interval	, Node_Name	, Address	, Length	, Data_Type
InventoryPoll1	, DA_INVENTORY	, 0	, Rdbc	, 1.0s	, Node_A	, 1	, 10	, In-Tank Inventory

Reading the inventory requires an active Map Descriptor.

Set the address to the tank number. One Map Descriptor per tank.

This is the minimum number of Data Array elements required to store inventory data.
Tanks 1's data will be stored in the array DA_INVENTORY starting at the zero'th element (offset=0) and 10 elements of the array are reserved for this data.

Setting the data type generates the correct poll. Ensure that spelling, periods, hyphens in exactly the same as this example.

5.3.6 Map Descriptor Example 3 – Most Recent Delivery Report

This example illustrates a poll for Tank Delivery Data. One Map Descriptor is required for each tank. The address should correspond to the tank number. The Length parameter tells the driver how much storage space to reserve for storing the delivery data. The delivery data is stored as follows. (This function corresponds to the function 20C in the Veeder-Root protocol specification.).

Index	Contents
0.	Product Code
1.	Number of Deliveries
2.	Start Time
3.	Stop Time
4.	Starting Volume
5.	Starting TC Volume
6.	Starting Water
7.	Starting Temp
8.	Ending Volume
9.	Ending TC Volume
10.	Ending Water
11.	Ending Temp
12.	Starting Height
13.	Ending Height

.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Scan_Interval	, Node_Name	, Address	, Length	, Data_Type
DeliveryPoll01	, DA_DELIVERY	, 0	, Rdbc	, 1.0s	, Node_A	, 1	, 30	, In-Tank Delivery

6 CONFIGURING THE FIELDSERVER AS A (VEEDER-ROOT DRIVER) SERVER

The server functionality of this driver was developed for testing the Client. Its features are not documented and not intended for end-users. It may be possible to extend and document the server at the request of an end-user.

Appendix A. Useful Features

Appendix A.1. Polling for Tank / Sensor / Input / Relay / Device Status

This example illustrates the CSV configuration to poll directly for the status of a particular tank. The length parameter reserves space in the array for the driver to store the possible alarm states for the tank. Appendix C.1.3 provides details of the alarm types.

The first element of the array is used to store an alarm summary state. The array element is set non-zero if there are any alarms and is set to zero if there are none. The remaining array elements are set non-zero based on the types of alarms active for the tank. Refer to Appendix C.1.1 for other supported Data Types.

Example:

Tank 1 has two alarms 02 = Tank Leak Alarm; 05 = Tank Low Product Alarm

The driver sets array DA_TANK1_ALMS as follows

Index=0 : Value=1 (Summary) Index=2 : Value=2 (Leak) Index=5 : Value=5 (Product)

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Scan_Interval , Node_Name , Address , Length , Data_Type
Tank1_01_status , DA_TANK1_ALMS , 0 , Rdbc , 1.0s , Node_A , 1 , 30 , Tank
```

This is an active Map Descriptor producing a poll

The address must correspond to the tank number (in this example or the sensor / input / relay / device number depending on the data type.)

Amount of array space required to store the alarm types.

Appendix A.2. Resetting Remote Alarms

This example provides a 'write-on-change' Map Descriptor used to send the Veeder-Root device a remote alarm reset command. When the value of the first element of the array named DA_COMMANDS changes value, the driver will send the command. The driver stores the Veeder-Root System time in the array on completion of the command.

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Address , Length , Data_Type
Reset_Alms_01 , DA_COMMANDS , 0 , Wrbx , Node_A , 1 , 30 , Remote Alarm Reset
```

Appendix A.3. System Configuration

This example reads the configuration of the Veeder-Root Panel. The module type is stored in the location corresponding to the slot number. Slot 1's data is stored at index location 0, slot 2 at index location1, and if the DA_Byte_Name and DA_Float_Name arrays are defined then the driver also stores the Power on Reset and Current Values in the corresponding array locations.

Table 6.7 provides a listing of module types.

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Scan_interval , Function , Node_Name , Address , Length , Data_Type
Config_poll_1       , DA_MODULES      , 0                 , 10.0s         , Rdbc     , Node_A    , 1       , 30     , Configuration


Map_Descriptor_Name , Data_Array_Name , DA_Byte_Name , DA_Float_Name , Data_Array_Offset , Scan_interval , Function , Node_Name , Address , Length , Data_Type
Config_poll_1       , DA_MODULES      , DA_POWER     , DA_CURRENT    , 0                 , 10.0s         , Rdbc     , Node_A    , 1       , 30     , Configuration
```

Power-On-Reset and Current values are also stored if the DA_Byte_Name and the DA_Float_Name parameters are defined. The data format must be set to UINT32 or FLOAT.

Appendix A.4. Advanced Example 4 – System Diagnostics

This example reads the System Diagnostics Report from the Veeder-Root Panel. The results are stored in 3 consecutive array elements. If the value is set non-zero then the test FAILED. A value of zero means the test passed.

Index Contents
I/O test result
Ram test result
Prom test result

```
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Scan_interval , Function , Node_Name , Address , Length , Veed_Function , Data_Type
System1poll_1       , DA_SELF_TEST    , 0                 , 10.0s         , Rdbc     , Node_A    , 1       , 3      , 0x901         , Misc.
```

Appendix A.5. System Revision Level Report I

This example reads the System Revision Level from the Veeder-Root Panel. The result is stored in Ascii using up to 100 consecutive array elements. Use an array format BYTE or STRING for this function.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Scan_interval	, Function	, Node_Name	, Address	, Length	, Veed_Function	, Data_Type
System1poll_1	, DA_REVISION	, 0	, 10.0s	, Rdbc	, Node_A	, 1	, 100	, 0x902	, Misc.

The following string will be returned and can be interpreted as described below.

YYMMDDHHmmSOFTWARE# nnnnnn-vvv-rrrCREATED - YY.MM.DD.HH.mm&&CCCC<ETX>

YYMMDDHHmm - Current Date and Time
nnnnnn-vvv - Software version number (ASCII text string)
rrr - Software revision level (ASCII text string)
YY.MM.DD.HH.mm - Date and time of software creation
&& - Data Termination Flag
CCCC - Message Checksum

Appendix A.6. System Revision Level Report II

This example reads additional System Revision Level data from the Veeder-Root Panel. The result is stored in Ascii using up to 100 consecutive array elements. Use an array with format BYTE or STRING for this function. The notes below explain how the data should be interpreted.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Scan_interval	, Function	, Node_Name	, Address	, Length	, Veed_Function	, Data_Type
System1poll_1	, DA_REVISION2	, 0	, 10.0s	, Rdbc	, Node_A	, 1	, 100	, 0x905	, Misc.

The following string will be returned and can be interpreted as described below.

i90500YYMMDDHHmmSOFTWARE# 346abb-Tvv-rrrCREATED - YY.MM.DD.HH.mmnnAABBCCDDEEFFGGHHIIJJS-MODULE# nnnnnn-vvv-r

YYMMDDHHmm	Current Date and Time
346	Software Base number (fixed)
a	Platform (0 = Standard CPU, PLLD only 1 = Enhanced CPU 2 = (Unused) 3 = Enhanced CPU 16 Tank 4 = Standard CPU without PLLD & WPLLD 5 = Standard CPU, WPLLD only)
bb	Version level (eg version "15")
T	Software Type (1 = "Real" 2 = "Demo" 3 = "IFSF")
vv	Language
rrr	Revision level (e.g. revision "AX1")
YY.MM.DD.HH.mm	Date and time of software creation
nn	number of 2 byte values to follow (Hex)
AA	PERIODIC IN TANK TESTS (00 = DISABLE, 01 = ENABLE)
BB	ANNUAL IN TANK TESTS (00 = DISABLE, 01 = ENABLE)
CC	CSLD (00 = DISABLE, 01 = ENABLE)
DD	BIR (00 = DISABLE, 01 = ENABLE)
EE	FUEL MANAGER (00 = DISABLE, 01 = ENABLE)
FF	PRECISION PLLD (00 = DISABLE, 01 = ENABLE)
GG	TANKER LOAD (00 = DISABLE, 01 = ENABLE)
HH	0.2 GPH PLLD (00 = DISABLE, 01 = ENABLE)
II	PRECISION PLLD ON DEMAND (00 = DISABLE, 01 = ENABLE)
JJ	SPECIAL 3 TANK/LINE CONSOLE (00 = DISABLE, 01 = ENABLE)
SEM Info 3 parts, if none "NO SOFTWARE MODULE	
nnnnnn	SEM number (ASCII text string)
vvv	SEM Software version number (ASCII text string)
r	SEM Software Revision number (ASCII text string)

Appendix A.7. Stick Height Report

This example reads the Stick height Report from the Veeder-Root Panel. The result is stored as a real number. Scaling may be applied if specified in the Map Descriptor. Refer to the FieldServer Configuration manual. Use one Map Descriptor per tank.

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset , Scan_Interval , Function , Node_Name , Address , Length , Data_Type
Poll_Height1          , DA_STICKHEIGHT    , 0                , 0.8s          , Rdbc     , Node_A    , 1       , 1      I, n-Tank Stick Height
```

Appendix A.8. Active Alarm Report

This example reads the Veeder-Root Panel for active alarms (includes acknowledged and active). If there are any active alarms then the 1st element of the Data Array is set non-zero. If there are no active alarms then the array element is set to zero. A number of passive are required Map Descriptors to store alarms for the tanks. Sensor / inputs / devices as described in Section 5.3.4

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset , Scan_Interval , Function , Node_Name , Address , Length , Data_Type
AlarmPoll-11          , DA_ACTIVEALM      , 0                , 2.0s          , Rdbc     , Node_A    , 1       , 1      , Active Alarm Report
```

Appendix A.9. Cleared Alarm Report

This example reads the Veeder-Root Panel for cleared alarms. If there are any cleared alarms then the 1st element of the Data Array is set non-zero. If there are no cleared alarms then the array element is set to zero. A number of passive are required Map Descriptors to store the cleared alarm info for the tanks. Sensor / inputs / devices as described in Section 5.3.4

The data is stored in the array specified using the DA_Byte_Name parameter of the passive Map Descriptors.

A stored value of "1" indicates Alarm Cleared. A stored value of "1 2" shows alarm occurred.
The value is stored at the location which corresponds to the alarm type.
Thus if a 02 = Tank Leak Alarm was cleared then at index=2 the value stored will be 1.

Active alarms are stored in the primary array.

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Scan_Interval	Function	Node_Name	Address	Length	Data_Type
AlarmPoll-11	DA_ACTIVEALM	0	2.0s	Rdbc	Node_A	1	1	Active Alarm Report
Map_Descriptors								
Map_Descriptor_Name	Data_Array_Name	DA_Byte_Name	Data_Array_Offset	Function	Node_Name	Address	Length	Data_Type
Sys_alms01	DA_S1	DA_S1_CLEAR	0	passive	Node_A	1	30	System
Tank_Alms1	DA_T1	DA_T1_CLEAR	0	passive	Node_A	1	30	Tank

The cleared alarm information is stored in the secondary array specified with the DA_Byte_Name

Appendix A.10. Security Codes

The RS-232 security code is an optional six-digit code used to limit external serial access to the system for security purposes. It can be set to any unique set of characters using either the front panel switches or the external communication interface setup commands. The system will not respond to a command without the proper security code if the DIP switch is set to enable RS-232 security.

A four-position DIP switch is located on the CPU board, which is mounted in the right-back of the console printer compartment beside the battery switch. The DIP switch enables the security code and the switch positions are assigned as follows:

Switch	
1	Front Panel Setup Security Enable
2	RS-232 Security Enable
3, 4	Unused

If RS-232 security is enabled then the **Route** parameter must be specified for the node. Set the route parameter equal to the security code. Use a period to separate each digit of the code.

Example: Security Code=234419
Route = 2.3.4.4.1.9

Alphabetic characters cannot be directly specified in the route field. They need to be specified as ASCII values.

Example : Security Code = 123abc
Route = 1.2.3.97.98.99

If the security feature has been disabled on the panel the set the route to zero

Appendix B. Troubleshooting

1. The number of messages received should equal the number of messages sent.

2. The number of errors should represent a small percentage of the total number of messages sent (< 5%).

3. Several errors in consecutive messages may cause the FieldServer kernel to place the node offline in which case polling is slowed significantly until good communications are re-established.

4. If the number of messages received is zero and the number of timeouts is equal to the number of messages sent then
    a. The connection is bad. Check the cables
    b. The security configuration is invalid
    c. The port settings are incorrect. Check the baud rate
5. If the number of messages received and sent are roughly equal and the number of errors is small but the Data Arrays do not update then
    a. If polling for System Status – Check the 1st element of the Data Array associated with the poll Map Descriptor. The driver sets or clears the element as a summary alarm state for the Veeder-root device.
    b. Check that the data age is less than the scan interval.
6. If the number of messages ignored is non-zero then this indicates that some data cannot be stored and is being discarded. Generally this arises when the driver cannot find an appropriate Map Descriptor to store data received in response to a poll (for composite data such as system status). For example: A system alarm occurs but Map Descriptors have not been defined to store system alarms.
7. Monitor the error log to see if the driver has reported any errors or important information. These messages arise in two ways.
    a. Configuration errors and warnings arise from the way that the CSV file has been configured. These should be eliminated during set-up.
    b. Errors may arise from some run-time condition. Many of these errors are produced in the error log once and then suppressed so that the log does not overflow or hide other meaningful information. These messages are preceded by an *

Appendix C. Reference

Appendix C.1. Supported Functions

Appendix C.1.1. Data Types Supported by the Driver

The Veeder-Root protocol specification defines a large number of enquiries and commands that may be sent to a Veeder-Root device. This driver supports the following functions:

System
Tank
Liquid Sensor
Vapor Sensor
Input
Volumetric Line
Groundwater Sensor
Type-B Sensor
Universal Sensor
Auto-Dial
Mech. Dispenser
Elec. Dispenser
Product
Press. Line
External
WPLLD

Appendix C.1.2. Veeder Root Support

The revision number indicates the minimum Veeder-Root firmware revision required for support of the function. The function numbers are hexadecimal numbers. Either convert them to decimal or specify them in the CSV file using the notation 0xnnn where nnn is the function number.

Function_	Revision_	Description
SYSTEM REPORTS (7.2.1)		
101	1	System Status Report
102	1	System Configuration Report
113	14	Active Alarm Report
114	19	Cleared Alarm Report
IN-TANK REPORTS (7.2.2)		
201	1	In-Tank Inventory Report
202	1	In-Tank Delivery Report
203	1	In-Tank Leak Detect Report
204	1	In-Tank Shift Inventory Report
205	1	In-Tank Status Report
20C	15	In-Tank Most Recent Delivery Report
20D	15	In-Tank Stick Height Report

SENSOR REPORTS (7.2.3)		
301	1	Liquid Sensor Status Report
306	1	Vapor Sensor Status Report
311	1	Groundwater Sensor Status Report
341	2	Type A (2 Wire CL) Sensor Status Report
346	2	Type B (3 Wire CL) Sensor Status Report
34B	4	Universal Sensor Status Report
LINE LEAK REPORTS (7.2.4)		
381	7	Pressure Line Leak Status
386	10	WPLLD Line Leak Status
I/O DEVICE REPORTS (7.2.6)		
401	1	Input Status Report
406	1	Relay Status Report
SYSTEM DIAGNOSTIC REPORTS (7.4.1)		
901	1	Self Test Results Report
902	1	System Revision Level Report
905	15	System Revision Level Report II
CONTROL FUNCTIONS (7.1)		
1	1	System Reset
2	1	Clear Power Reset Flag
3	1	Remote Alarm Reset
31	10	Confirm Clear Function
51	1	Clear In-Tank Delivery Reports
52	1	Start In-Tank Leak Detect Test
53	1	Stop In-Tank Leak Detect Test
54	5	Delete CSLD Rate Table
81	7	Start Pressure Line Leak Test (3.0 GPH only in V18)
82	7	Stop Pressure Line Leak Test
83	10	Start WPLLD Line Leak Test (3.0 GPH only in V18)
84	10	Stop WPLLD Line Leak Test

Appendix C.1.3. Alarm Tables

The following data will be stored for each alarm type in Data Arrays as set up in the example in Section 5.3.4

Data_Type	Data Array Offset	Description
System	1	Printer out of Paper
	2	Printer Error
	3	EEPROM Configuration Error
	4	Battery Off
	5	Too Many Tanks
	6	System Security Warning
	7	ROM Revision Warning
	8	Remote Display Communications Error
	9	Autodial Error
	10	Software Module Warning
	11	Tank Test Shutdown Warning
	12	Protective Cover Alarm
	13	BIR Shift Close Pending
	14	BIR Daily Close Pending
	15	PC(H8) Revision Warning
	16	System Self Test Error
	17	System Clock Incorrect Warning
	18	System Device Poll Timeout
Tank	1	Tank Setup Data Warning
	2	Tank Leak Alarm
	3	Tank High Water Alarm
	4	Tank Overfill Alarm
	5	Tank Low Product Alarm
	6	Tank Sudden Loss Alarm
	7	Tank High Product Alarm
	8	Tank Invalid Fuel Level Alarm
	9	Tank Probe Out Alarm
	10	Tank High Water Warning
	11	Tank Delivery Needed Warning
	12	Tank Maximum Product Alarm
	13	Tank Gross Leak Test Fail Alarm
	14	Tank Periodic Leak Test Fail Alarm
	15	Tank Annual Leak Test Fail Alarm
	16	Tank Periodic Test Needed Warning
	17	Tank Annual Test Needed Warning
	18	Tank Periodic Test Needed Alarm
	19	Tank Annual Test Needed Alarm
	20	Tank Leak Test Active
	21	Tank No CSLD Idle Time Warning
	22	Tank Siphon Break Active Warning

Data_Type	Data Array Offset	Description
	23	Tank CSLD Rate Increase Warning
	24	Tank AccuChart Calibration Warning
	25	Tank HRM Reconciliation Warning
	26	Tank HRM Reconciliation Alarm
	27	Tank Cold Temperature Warning
	28	Tank Missing Delivery Ticket Warning
	29	Tank/Line Gross Leak Alarm
Liquid Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning
Vapor Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning
Input	1	Input Setup Data Warning
	2	Input Normal
	3	Input Alarm
Volumetric Line Leak	1	VLLD Setup Data Warning
	2	VLLD Self Test Alarm
	3	VLLD Shutdown Alarm
	4	VLLD Leak Test Fail Alarm
	5	VLLD Selftest Invalid Warning
	6	VLLD Continuous Handle On Warning
	7	VLLD Gross Line Test Fail Alarm
	8	VLLD Gross Line Selftest Fail Alarm
	9	VLLD Gross Pump Test Fail Alarm
	10	VLLD Gross Pump Selftest Fail Alarm
	11	VLLD Periodic Test Needed Warning
	12	VLLD Annual Test Needed Warning
	13	VLLD Periodic Test Needed Alarm
	14	VLLD Annual Test Needed Alarm
	15	VLLD Periodic Line Test Fail Alarm

Data_Type	Data Array Offset	Description
	16	VLLD Periodic Line Selftest Fail Alarm
	17	VLLD Periodic Pump Test Fail Alarm
	18	VLLD Periodic Pump Selftest Fail Alarm
	19	VLLD Annual Line Test Fail Alarm
	20	VLLD Annual Line Selftest Fail Alarm
	21	VLLD Annual Pump Test Fail Alarm
	22	VLLD Annual Pump Selftest Fail Alarm
	23	VLLD Pressure Warning
	24	VLLD Pressure Alarm
	25	VLLD Gross Test Fault Alarm
	26	VLLD Periodic Test Fault Alarm
	27	VLLD Annual Test Fault Alarm
	28	VLLD Fuel Out Alarm
Groundwater Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning
Type-A Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning
Type-B Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning

Data_Type	Data Array Offset	Description
Universal Sensor Alarm	2	Sensor Setup Data Warning
	3	Sensor Fuel Alarm
	4	Sensor Out Alarm
	5	Sensor Short Alarm
	6	Sensor Water Alarm
	7	Sensor Water Out Alarm
	8	Sensor High Liquid Alarm
	9	Sensor Low Liquid Alarm
	10	Sensor Liquid Warning
Auto-Dial	1	Autodial Setup Data Warning
	2	Autodial Failed Alarm
	3	Autodial Service Report Warning
	4	Autodial Alarm Clear Warning
	5	Autodial Delivery Report Warning
Elec. Dispenser	2	DIM Disabled Alarm
	3	DIM Communication Failure Alarm
	4	DIM Transaction Alarm
Product Alarm	1	BIR Setup Data Warning
	2	BIR Threshold Alarm
	3	BIR Close Shift Warning
	4	BIR Close Daily Warning
Pressure Line Leak	1	PLLD Setup Data Warning
	2	PLLD Gross Test Fail Alarm
	3	PLLD Annual Test Fail Alarm
	4	PLLD Periodic Test Needed Warning
	5	PLLD Periodic Test Needed Alarm
	6	PLLD Sensor Open Alarm
	7	PLLD High Pressure Alarm (Obsolete V19)
	8	PLLD Shutdown Alarm
	9	PLLD High Pressure Warning (Obsolete V19)
	10	PLLD Continuous Handle On Warning (Obsolete V19)
	11	PLLD Periodic Test Fail Alarm
	12	PLLD Annual Test Needed Warning
	13	PLLD Annual Test Needed Alarm
	14	PLLD Low Pressure Alarm
	15	PLLD Sensor Short Alarm (Obsolete V19)
	16	PLLD Continuous Handle On Alarm
	17	PLLD Fuel Out Alarm
	18	PLLD Line Equipment Alarm

Data_Type	Data Array Offset	Description
Wireless PLD	1	WPLLD Setup Data Warning
	2	WPLLD Gross Test Fail Alarm
	3	WPLLD Periodic Test Fail Alarm
	4	WPLLD Periodic Test Needed Warning
	5	WPLLD Periodic Test Needed Alarm
	6	WPLLD Sensor Open Alarm
	7	WPLLD Communications Alarm
	8	WPLLD Shutdown Alarm
	9	WPLLD Continuous Handle On Warning (Obsolete V19)
	10	WPLLD Annual Test Fail Alarm
	11	WPLLD Annual Test Needed Warning
	12	WPLLD Annual Test Needed Alarm
	13	WPLLD High Pressure Warning (Obsolete V19)
	14	WPLLD High Pressure Alarm (Obsolete V19)
	15	WPLLD Sensor Short Alarm (Obsolete V19)
	16	WPLLD Continuous Handle On Alarm
	17	WPLLD Fuel Out Alarm
	18	WPLLD Line Equipment Alarm
Externally Detected Alarm	1	Externally Detected Communication Alarm
	2	Communications - Data Reception Timeout
	3	Communications - Failed Checksum
	4	Communications - Parity Error
	5	Modem - Line Busy
	6	Modem - No Answer
	7	Modem - No Carrier
	8	Modem - No Dial Tone
	9	Modem - Modem Error
	10	Modem - Modem Not Responding
	11	Modem - Port Not Available
	12	Polling - Could Not Update Queue
	13	Polling - Invalid Data Type Requested

Appendix C.2. Data Type and Required Lengths

The following table provides a list of possible values for the data type parameter as well as the minimum value that the length parameter should be set to for each Map Descriptor of that data type.

Data Type	Length Parameter Setting
Normal	1
System	30
Tank	30
Liquid Sensor	30
Vapor Sensor	30
Input	30
Volumetric Line	30
Groundwater Sensor	30
Type-A Sensor	30
Type-B Sensor	30
Universal Sensor	30
Auto-Dial	30
Mech. Dispenser	30
Elec. Dispenser	30
Product	30
Press. Line	30
WPLLD	30
External	30
Relay	30
Configuration	1
In-Tank Inventory	10
Misc	20
In-Tank Delivery	100
System Reset	20
Clear Power Reset	20
Remote Alarm Reset	20
Confirm Clear	20
Clear Delivery Reports	20
Start In-Tank Leak Detect Test	20
Stop In-Tank Leak Detect Test	20
Delete CSLD Rate Table	20
Start P-Line Leak Detect Test	20
Stop P-Line Leak Detect Test	20
Start WPLLD Leak Detect Test	20
Stop WPLLD Leak Detect Test	20
System Status	1

Appendix C.3. Module Types

00 - Not used	0A - Four Probe w/ Ground Temp Module
01 - FourProbe Module	0B - Groundwater Sensor Module
02 - Vapor Sensor Module	0C - Type A Sensor Module
03 - Liquid Sensor Module	0D - Remote Display Module
04 - FourRelay Module	
05 - I/O Combo Module	
06 - Printer Module	
07 - RS-232 Module	
08 - Modem Module	
09 - Volumetric Line Leak Module	
10 - Type B Sensor Module	1A - Pressure Line Leak Sensor Module
11 - Universal Sensor Module	1B - Pressure Line Leak Controller Module
12 - Fax/Modem (1785) Module	1D - Remote Printer Module
13 - Remote/Local Printer Module	1E - External Fax/Modem Module
14 - Pump Sensor Module	1F - RS-485 Module
15 - European RS-232 Module	
17 - EightProbe Module	
18 - Mechanical Dispenser Interface Module	
19 - Electronic Dispenser Interface Module	
20 - Wireless PLLD AC Interface Module	
21 - Wireless PLLD Communications Module	
22 - Wireless PLLD Controller Module	
23 - Hughes Satellite J-Box Module	
24 - Fax/Modem (1786) Module	
25 - Serial Satellite Module	
26 - Three Probe / Three Liquid Sensor Module	
27 - Three PLLD Sensor Module	

Appendix C.4. Error Messages

Error	Action Required
VRoot:#1 FYI. The mapDesc called <%s> is too short.[4]	The length of the Map Descriptor used to expose driver statistics is too short. Set the length to at least 600.
VRoot:#2 FYI. You could have used a mapDesc called <%s> to expose diagnostic info.	You can safely ignore this message. It is a prompt. Refer to Appendix C.5.
*VRoot:#3 Err. Data Type=%d is unknown. Data will be discarded.	The driver has encountered a message reporting an alarm for an equipment category/data type the driver does not recognize. Refer to FST..

[4] Download the CSV file, make the relevant changes, then upload the file and cycle power to the FieldServer.

Error	Action Required
VRoot:#4 FYI. If there is more than one VRoot node then connect each node to a dedicated port	You may ignore this message if only one Veeder-Root node is connected to the FieldServer. If more than one node is connected then each node must be connected to a separate port.
VRoot:#5 Err. Data Type invalid for this driver. Md=<%s>[4]	The Data types must be spelled, spaced and punctuated exactly as in Appendix C.1
VRoot:#6 Err. No Default Poll Function for this data type. Md= <%s>[4]	Some data types require an additional Map Descriptor parameter 'VEED_Function' to be specified..
VRoot:#7 Err. Length too short. Rqd=%d. Md=<%s>[4]	The message reports the offending Map Descriptor and the number of array elements required. Adjust the length parameter accordingly. Ensure that the new length coupled with the offset fits in the array size.
VRoot:#11 Err. Command 0x54 cannot specify all tanks[4]	For the Data Type (or Veed_Function) specified the address may not be set to zero. You need one Map Descriptor per tank/sensor/input/device.
VRoot:#12 Err. Command 0x54. Bad confirmation code.	Refer to FST.
*VRoot:#13 Err. Device Rejected message. Md=<%s>	This message is generated the first time that the Error Statistic 'Function Error' is produced. This error can arise in several ways. The firmware version of the Veeder-Root device may not support the poll generated by the Map Descriptor. Refer to Appendix C.1 The Veeder-Root device may not have the tank/sensor/device specified or the message received by the Veeder-Root device is corrupted. Check the devices configured against the address specified in the Map Descriptor.
*VRoot:#16 Err. Array too short to store. Md=<%s> Rqd=%d	When this error is produced incoming data is being discarded and alarm or other data will not be updated. There are a few variations on this message. Some do not report the Rqd length. This message is produced as a run-time error. It may arise from a configuration error that has not been corrected. Check for Error #7 messages. If there are any correct the errors by editing the CSV file. Then reset the FieldServer. If the driver is still producing error 16 then you should report this to FieldServer support. Provide a copy of your configuration file and a port log to assist in the diagnosis of this error.
*VRoot:#17 Err. Array too short to store. Md= <%s> ByteArray	This error is similar to #16 except that it applies to the array named in the DA_Byte_Name parameter for the Map Descriptor.
*VRoot:#18 Err. Array too short to store. Md= <%s> Float Array	This error is similar to #16 except that it applies to the array named in the DA_Float_Name parameter for the Map Descriptor.
VRoot:#19 Err. Cant process msg type= %x	You cannot take any corrective action to eliminate this error. Report the message to FieldServer support. It will assist in the diagnosis of the error if you can take a port log and provide a copy of your CSV file.

Error	Action Required
*VRoot:#20 FYI. Incoming data is being abandoned. Func= %x DT= <%s> Addr=%d	This message is a warning. It indicates that a response to a poll contains data for which a Map Descriptor cannot be found to store the data. For example: The response to poll for System Status returns an alarm for Liquid Sensor #3 but you have not defined a Map Descriptor for Liquid sensor #3. You may be able to use the Data Type and the Address to determine what kind of Map Descriptor is required. If, however, you are satisfied that you have Map Descriptors for the data of interest to you then you may safely ignore this message.
*VRoot:#21 Err. Md=<%s> not bound to a port. Can't be used.	Read the notes for Error #4 to see how to correct this problem.
Vroot:#22 FYI. Config reported zero modules.	You may ignore this message. The Veeder-Root device reported zero configured modules in response to a configuration enquiry.
*VROOT:#23 Err. Address= 0 invalid for this command. Md= <%s>[5]	Set the address to a positive number corresponding to the tank/sensor/input/device whose data you are polling for.
*VROOT:#24 Err. Function=%x not supported. Md= <%s>[5]	Check the data type's spelling spacing and punctuation. If the Veed_Function parameter has been specified in the Map Descriptor, check that the function is supported by the driver and that it is specified correctly. Refer to Appendix C.1.2
VROOT:#25 FYI. Node=<%s> No password. Is security disabled on panel?	This message may be safely ignored. It does not indicate an error. This message is printed once for each node which does not have a password. This message provides confirmation that your configuration is not using a password and suggests that you check the panel configuration for the corresponding node to ensure that the security feature has been disabled. Additional information is provided in Appendix A.10
VROOT:#26 FYI. Node=<%s> Password=<%s>. Is security enabled on panel?	

[5] Download the CSV file, make the relevant changes, then upload the file and cycle power to the FieldServer.

Appendix C.5. Driver Stats

In addition to the standard FieldServer communication statistics described in the FieldServer Configuration Manual, Veeder-Root Driver can also expose some driver statistics by writing data to a Data Array. A special Map Descriptor named "vroot-stats" is required.

The following example shows how this special Map Descriptor can be configured. This section of text may be copied directly into your CSV file.

```
Nodes
Node_Name           , Protocol
null_node           , VRoot

Data_Arrays
Data_Array_Name     , Data_Format     , Data_Array_Length
VROOT_STATS         , UINT32          , 1000

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Node_Name      , Length , Data_Type
VRoot-stats         , VROOT_STATS     , Null_Node      , 1000   , Stats
```

The driver uses the Data Array VROOT _STATS (in this example) to store driver specific statistics. Only one of these Map Descriptors may be specified per FieldServer.

The driver stores the following data.

1 VEED_MASTER_SENDS_MSG
2 VEED_MASTER_SENDS_BYTES
3 VEED_MASTER_TIMEOUT
4 VEED_MASTER_STREAMING
5 VEED_MASTER_RCVS_NAK
6 VEED_MASTER_RCVS_CHECKSUM
7 VEED_MASTER_RCVS_PROTOCOL
8 VEED_MASTER_COMPLETE_ERR
9 VEED_MASTER_RCVS_MSG
10 VEED_MASTER_RCVS_BYTES
11 VEED_MASTER_IC_TIMEOUT
12 VEED_MASTER_SEND_BAD_MSG
13 VEED_STAT_MSG_IGNORED
14 VEED_MASTER_RCV_BAD_FUNC_RESPONSE