

# VNA-USB Converter Manual

Version 1.0  
Revised November 13<sup>th</sup>, 2011  
Created by the [J1939 to USB](#) Experts



**Simma Software**

## Warranty

Simma Software's products and software, when properly installed and used, are warranted for a period of one year from the date of order. If Simma Software receives notice of such defects during the warranty period, Simma Software will replace the adapter at no charge. Simma Software does not assume liability for its products or software, and does not warrant that the operation of the products or software shall be error free.

If during the investigation of a warranty claim, the customer and Simma Software both agree the suspected defect is not a defect, time spent by Simma Software will be paid for at \$120/hr and all expenses will be reimbursed.

THE WARRANTY SET FORTH ABOVE IS EXCLUSIVE AND NO OTHER WARRANTY, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED. SIMMA SOFTWARE SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR SUCH PURPOSE.

THE REMEDIES PROVIDED HEREIN ARE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. UNLESS REQUIRED BY APPLICABLE LAW, IN NO EFFECT SHALL SIMMA SOFTWARE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING LOSS OF PROFITS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY.

# TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION</b>	<b>4</b>
<b>2.0</b>	<b>USB FRAMING</b>	<b>5</b>
<b>3.0</b>	<b>USB MESSAGES</b>	<b>7</b>
<b>4.0</b>	<b>PHYSICAL CONNECTIONS</b>	<b>20</b>
<b>5.0</b>	<b>SPECIFICATIONS</b>	<b>23</b>

# Chapter 1

## Introduction

---

VNA-USB is a high performance Vehicle Network Adapter. The adapter interfaces with a computer using a standard full speed USB 2.0 port at speeds up to 12 Mbps. The vehicle network adapter supports connections to two SAE J1939 networks and one J1708 network simultaneously.

Connection to the host computer is made via the USB A connector on the device. A standard USB cable may be used to make this connection. Connection to the vehicle network is made using the female DB15 connector found on the device. Power is supplied to the VNA through the USB connection. All connections to the device are more thoroughly detailed in the Chapter 4.

# Chapter 2

## USB Framing

---

This chapter describes the message framing protocol used to define the start and body of messages sent between host device and the VNA-USB.

Data is sent to and from the VNA-USB using the industry standard method of byte stuffing. Byte stuffing is the process of reserving certain values for special purposes. This allows for error recovery if USB communications are disrupted. The VNA-USB protocol defines two special tokens, START and ESC. START has a value 192 decimal and ESC has a value of 219 decimal.

All messages start with a START token, and because of this, a START token can never appear in any part of the message. If the value of 192 is contained in the message length, data field, or checksum, an ESC token is inserted into the data stream followed by a 220, this indicates a value of 192. If an ESC token appears in the message length, data field, or checksum, an ESC token is inserted into the stream followed by a 221, this indicates a value of 219. All messages must be byte stuffed, following the START token, by the transmitter and unstuffed by the receiver.

To send a packet, a VNA-USB host starts by sending a START character followed by the size of the data to follow, along with the data and the checksum of the packet. The message length is a 2-byte field (MSB) equal to the size of the data field and checksum before byte stuffing has been executed. The checksum is a 2's complement checksum over the length and data fields and is calculated before byte stuffing has been added.

START	Length (MSB)	Length (LSB)	Data Field	Checksum
-------	-----------------	-----------------	------------	----------

Figure2-1 Framing Structure

**Byte Stuffing Example 1:**

To transmit 10 zeros (in chronological order)

<b>Description</b>	<b>Byte #</b>	<b>Value</b>
Start	0	192
MSB Length	1	0
LSB Length	2	11
Data 0	3	0
Data 1	4	0
Data 2	5	0
Data 3	6	0
Data 4	7	0
Data 5	8	0
Data 6	9	0
Data 7	10	0
Data 8	11	0
Data 9	12	0
Checksum (includes length and data)	13	245

**Byte Stuffing Example 2:**

To transmit 0x00 (0), 0xC0 (192), 0xDB (219), 0xDC (220), 0xDD (221)  
(Data bytes 1 and 2 require byte stuffing)

<b>Description</b>	<b>Byte #</b>	<b>Value</b>
Start	0	192
MSB Length	1	0
LSB Length	2	6
Data 0	3	0
Data 1	4	219
Data 2	5	220
Data 3	6	219
Data 4	7	221
Data 5	8	220
Data 6	9	221
Checksum (includes length and data bytes before stuffing)	10	166

# Chapter 3

## USB Messages

---

This chapter describes the commands that can be sent to and from the VNA-USB via USB. Note, all message definitions are in their unstuffed representations.

Messages	Functional Descriptions
VMSG_ACK	Acknowledge a message
VMSG_FA_J1939	Add a J1939 message filter
VMSG_FD_J1939	Delete a J1939 message filter
VMSG_FA_J1587	Add a J1587 message filter
VMSG_FD_J1587	Delete a J1587 message filter
VMSG_TX_J1939	Transmit a single J1939 message
VMSG_RX_J1939	Receipt of a single J1939 message
VMSG_PX_J1939	Periodically transmit a J1939 message
VMSG_TX_J1587	Transmit a single J1587 message
VMSG_RX_J1587	Receipt of a single J1587 message
VMSG_PX_J1587	Periodically transmit a J1587 message
VMSG_CPU_RESET	Requests CPU reset in 100ms

**Table 4-1: Messages and Functions**

Every message starts with a message identifier. This is an eight bit value that tells the receiver what type of message follows. This is followed by one or more additional data fields depending on the message. The message identifiers, data fields, responses, etc. can be found for each message on the following pages.

Message ID	Data Byte 0	Data Byte 1	...	Data Byte n
------------	-------------	-------------	-----	-------------

Figure 4-1 Message Structure

**ACK**

## Acknowledge

**ACK****Description:**

This message is sent to acknowledge the receipt of a message from the host system.

**Syntax:**

8b-vmsg, 8b-vmsg\_ack

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	0	Message Identifier
vmsg_ack	1 byte	variable	Message identifier of acknowledged message

**Response:**

None

**Example:**

Acknowledges the receipt of a TX\_J1939 command  
0/5



**FA\_J1939**

Filter Add

**FA\_J1939****Description:**

This message adds a PGN to the acceptance filter the desired port.

**Syntax:**

8b-vmsg, 8b-port, 24b-pgn

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	1	Message Identifier
port	1 byte	0 or 1	J1939 Port Number
pgn	3 bytes	variable	Parameter Group Number (MSB first)

**Response:**

ACK

**Example:**

Add acceptance filter for the PGN 65236 to port 0.

1/0/0/254/212

**FD\_J1939**

## Filter Delete

**FD\_J1939****Description:**

This message removes a PGN from the acceptance filter for the specified port.

**Syntax:**

8b-vmsg, 8b-port, 24b-pgn

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	2	Message Identifier
port	1 byte	0 or 1	J1939 Port Number
pgn	3 bytes	variable	Parameter Group Number (MSB first)

**Response:**

ACK

**Example:**

Deletes the acceptance filter for the PGN 65236 on port 0.

2/0/0/254/212

**FA\_J1587**

Filter Add

**FA\_J1587****Description:**

This message adds a PID filter to the J1587 port.

**Syntax:**

8b-vmsg, 16b-pid

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	3	Message Identifier
pid	2 byte	variable	PID (MSB first)

**Response:**

ACK

**Example:**

Adds acceptance filter for the PID 12345.

3/48/57

**FD\_J1587**

Filter Delete

**FD\_J1587****Description:**

This message deletes a PID filter from the J1587 port.

**Syntax:**

8b-vmsg, 16b-pid

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	4	Message Identifier
pid	2 byte	variable	PID (MSB first)

**Response:**

ACK

**Example:**

Removes the acceptance filter for the PID 12345.

4/48/57

**TX\_J1939**

Transmit

**TX\_J1939****Description:**

This message schedules a J1939 message for transmission. The message will be sent as soon as the attached CAN network allows.

**Syntax:**

8b-vmsg, 8b-port, 24b-pgn, 8b-dst, 8b-src, 8b-pri, nB-data

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	5	Message Identifier
port	1 byte	0 or 1	J1939 Port Number
pgn	3 byte	variable	Parameter Group Number (MSB first)
dst	1 byte	variable	Destination Address
src	1 byte	0 to 252	Source Address
pri	1 byte	0 to 7	Priority
data	variable	variable	Data

**Response:**

ACK

**Example:**

Schedules a J1939 packet on port 1 with pgn=12345, dst=255, src=252, pri=6, data=0 to 10

5/1/0/48/57/255/252/6/0/1/2/3/4/5/6/7/8/9/10

**RX\_J1939**

Receive

**RX\_J1939****Description:**

This message is sent to the host device when the VNA receives a J1939 message that passes acceptance filtering.

**Syntax:**

8b-vmsg, 8b-port, 24b-pgn, 8b-dst, 8b-src, 8b-pri, nB-data

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	6	Message Identifier
port	1 byte	0 or 1	J1939 Port Number
pgn	3 byte	variable	Parameter Group Number (MSB first)
dst	1 byte	variable	Destination Address
src	1 byte	0 to 252	Source Address
pri	1 byte	0 to 7	Priority
data	variable	variable	Data

**Response:**

ACK

**Example:**

Receipt of a J1939 packet on port 1 with pgn=12345, dst=255, src=252, pri=6, data=0 to 10

6/1/0/48/57/255/252/6/0/1/2/3/4/5/6/7/8/9/10

**PX\_J1939**

## Periodic Transmit

**PX\_J1939****Description:**

This message schedules a message for periodic transmission on the specified port. Message transmission time may be slightly delayed from the value specified depending on the usage of the network the device is attached to.

**Syntax:**

8b-vmsg, 8b-port, 24b-pgn, 8b-dst, 8b-src, 8b-pri, nB-data, 16b-rate

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	7	Message Identifier
port	1 byte	0 or 1	J1939 Port Number
pgn	3 byte	variable	Parameter Group Number (MSB first)
dst	1 byte	variable	Destination Address
src	1 byte	0 to 252	Source Address
pri	1 byte	0 to 7	Priority
data	variable	variable	Data
rate	2 bytes	variable	Rate of re-transmission in mS

**Response:**

ACK

**Example:**

Schedules a J1939 packet on port 1 to transmit every second with pgn=12345, dst=255, src=252, pri=6, data=0 to 10

7/1/0/48/57/255/252/6/0/1/2/3/4/5/6/7/8/9/10/3/232

**TX\_J1587**

Transmit

**TX\_J1587****Description:**

This message schedules a J1587 message for transmission

**Syntax:**

8b-vmsg, 8b-mid, 16b-pid, 8b-pri, nB-data

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	8	Message Identifier
mid	1 byte	variable	J1587 MID
pid	2 byte	variable	Parameter Identifier
pri	1 byte	1 to 8	Priority
data	variable	variable	Data

**Response:**

ACK

**Example:**

Schedules a J1587 packet to transmit with mid=123, pid=321, pri=4, data=0 to 10

8/123/321/4/0/1/2/3/4/5/6/7/8/9/10



**RX\_J1587**

Receive

**RX\_J1587****Description:**

This message is sent to the host device when the VNA receives a J1587 message that passes acceptance filtering.

**Syntax:**

8b-vmsg, 8b-mid, 16b-pid, nB-data

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	9	Message Identifier
mid	1 byte	variable	J1587 MID
pid	2 byte	variable	Parameter Identifier
data	variable	variable	Data

**Response:**

ACK

**Example:**

Receipt of a J1587 packet with mid=123, pid=321, pri=4, data=0 to 10  
9/123/321/4/0/1/2/3/4/5/6/7/8/9/10

**RESET**

## CPU RESET

**RESET****Description:**

This message requests a CPU reset in 100 ms. Note, correct key must be supplied.

**Syntax:**

8b-vmsg, 24b-key

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	17	Message Identifier
key	3 bytes	0x5A69A5	Unlock key

**Response:**

ACK

**Example:**

Requests a CPU reset.  
17/90/105/165

**STATS**

## Message Statistics

**STATS****Description:**

This message is sent to the host device every second. It contains the total number of valid J1708 messages, invalid J1708 bytes, and CAN frames.

**Syntax:**

8b-vmsg, 32b-stat0, 32b-stat1, 32b-stat2, 8b-verhw, 8b-versw

**Definition:**

<i>Name</i>	<i>Size</i>	<i>Value</i>	<i>Description</i>
vmsg	1 byte	23	Message Identifier
stat0	4 bytes	Variable	Total received J1708 valid messages
stat1	4 bytes	Variable	Total received J1708 invalid bytes
stat2	4 bytes	Variable	Total received CAN frames
verhw	1 byte	Variable	Version of hardware
versw	1 byte	Variable	Version of software

**Response:**

None

# Chapter 4

## Physical Connections

---

This chapter describes how to connect the VNA-USB to your vehicle. For further implementation details see later chapters.

Vehicle:

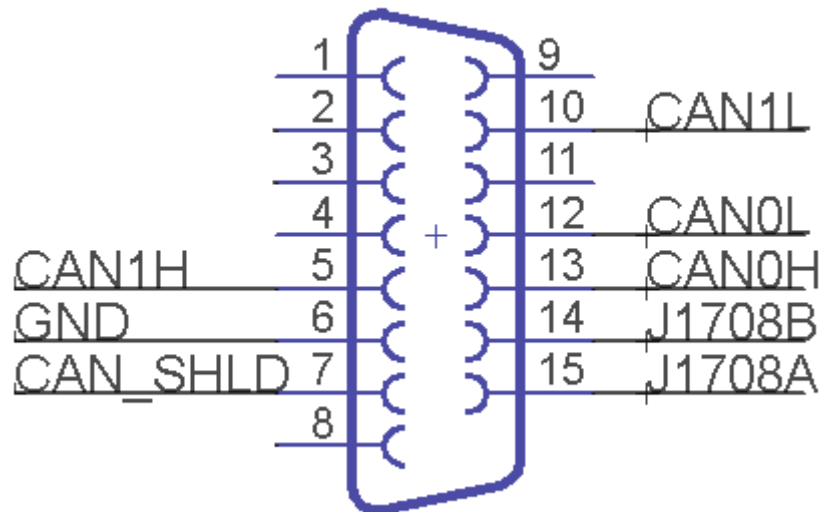


Figure 4-1 VNA-USB Vehicle Connection

The VNA is designed to connect to a heavy duty truck using a Deutsch 9 pin or 6 pin HD connector. Pinout is industry standard, so cable adapters are interchangeable.

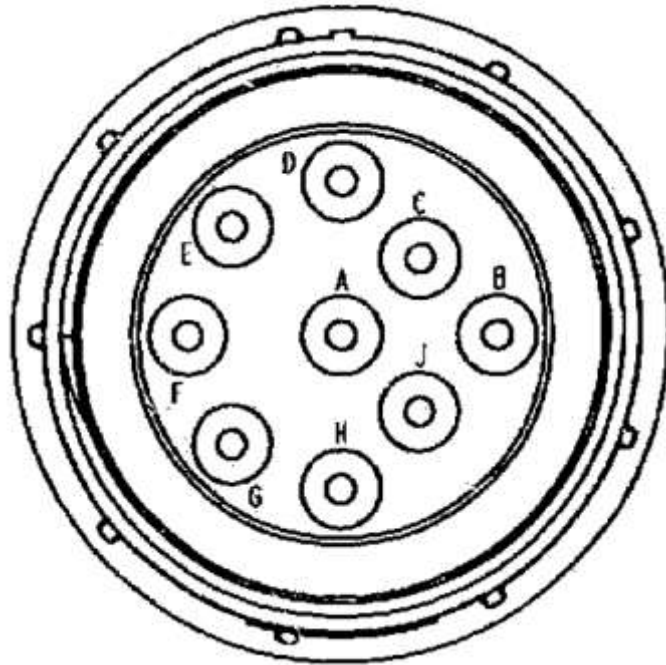


Figure 4-2 Standard Deutsch HD Connector (Plug)

Function	Deutsch 9 Pin	DB 15 Pin
Ground (-)	1/A	6
Battery (+)	2/B	8
CAN0 High (+)	3/C	13
CAN0 Low (-)	4/D	12
CAN Shield	5/E	7
J1708 (+)	6/F	15
J1708 (-)	7/G	14
CAN1 High (+)	8/H	5
CAN1 Low (-)	9/J	10

Table 4-1 Standard DB15 compared to Deutsch HD pinout

### Host computer:

The VNA is designed to communicate to a host computer/embedded device via a standard USB type B connector. Connection on the VNA is implemented using a standard USB cable.



Figure 4-3 VNA-USB USB Type B Connector

# Chapter 5

# Specifications

---

## Physical

Serial Connection: USB type B  
Vehicle Connection: DB-15 Female  
Temperature Rating: -40 to +85C (+125C available)

## Electrical

5 VDC Input, 0.3W max  
All communications lines feature 8kV or higher IEC ESD protection  
CAN transceiver meets or exceeds ISO 11898 standard  
J1708 transceiver fully compliant with TIA/EIA-485A

## Functional

10 Individual J1939 periodic message transmissions  
10 Individual J1708 periodic message transmissions  
100 Individual J1939 acceptance message filters  
100 Individual J1587/J1708 acceptance message filters  
CAN 125kbps to 1 Mbps  
USB Full speed, up to 12 Mbps