# VNA2-ELD

# Users Manual

Version 1.3

Revised July 9, 2018

Provider of ELD Adapters

Simma Software

# Warranty

Simma Softwares 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 CUSTOMERS 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.

# Contents

# Chapter 1

# Introduction

VNA2-ELD is a high performance Vehicle Network Adapter. The adapter interfaces with a computer, tablet, or phone using a Bluetooth SPP connection at speeds up to 1 Mbps. The adapter can also interface via RS-232 (disabled by default). The vehicle network adapter supports connections to two SAE J1939 networks or two OBDII networks and one J1708 network simultaneously.

Connection to the host computer is made via Bluetooth Classic or Bluetooth Low Energy, or through RS-232 using the female DB-9. Connections utilizing the DB-9 should use a straight through serial cable not a null modem cable. Connection to the vehicle network is made using the female DB15 connector found on the device. Power must be supplied to the VNA through this connection. All connections to the device are more thoroughly detailed in Chapter 4.

# Chapter 2

# RS-232 Framing

This chapter describes the message framing protocol used to define the start and body of messages sent between host device and the VNA2-ELD.

Data is sent to and from the VNA2-ELD 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 RS-232 communications are disrupted. The VNA2-ELD 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 VNA2-ELD 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 2s complement checksum over the length and data fields and is calculated before byte stuffing has been added.

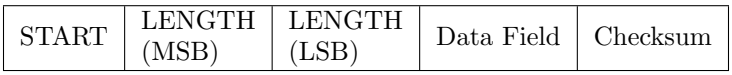


Figure 2.1: Framing Structure

### Byte Stuffing Example 1:

To transmit 10 zeros

| Description | Byte # | Value |
|---|---|---|
| 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)

| Description | Byte # | Value |
|---|---|---|
| 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) | 13 | 166 |

# Chapter 3

# RS-232 Messages

This chapter descries the commands that can be sent to and from the VNA2-ELD via Bluetooth. Note, all message definitions are in their unstuffed representations.

| Messages | Functional Descriptions |
|---|---|
| VMSG_ACK | Acknowledge a message |
| VMSG_REQ | Request a message |
| VMSG_FA_J1939 | Add a J1939 message filter |
| VMSG_FA_EXT_J1939 | Add a time based J1939 filter |
| VMSG_FD_J1939 | Delete a J1939 message filter |
| VMSG_FD_EXT_J1939 | Delete a time based J1939 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_TX_J1587 | Transmit a single J1587 message |
| VMSG_RX_J1587 | Receipt of a single J1587 message |
| VMSG_TX_I15765 | Transmit a single OBD2 message |
| VMSG_RX_I15765 | Receipt of a single OBD2 message |
| VMSG_ODOMETER | Set and Receive the digital odometer |
| VMSG_GPS | Receive GPS location |
| VMSG_REC_COUNT | Count of stored records |
| VMSG_REC_BATCH | Request and Receive stored records |
| VMSG_REC_DELETE | Delete received stored records |
| VMSG_LOGIN | Log a user into the VNA |
| VMSG_LOGOUT | Logout a user from the VNA |
| VMSG_ACONN | Control VNA auto-connect system |
| VMSG_MAC | Request VNA BT MAC address |
| VMSG_STATS | Periodic VNA status message |
| VMSG_CHIRPCON | Toggle periodic status messages |

Table 3.1: 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 |
|---|---|---|---|---|

Table 3.2: 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:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 0 | Message Identifier |
| vmsg_ack | 1 byte | variable | Message identifier of acknowledged msg |

## Response:

None

## Example:

Acknowledges the receipt of a TX_J1939 command

0/5

# REQ Request REQ

## Description:

This message is sent to request an intenral VNA message. You can request: ACONN, REC_TOTAL, REC_BATCH, and MAC. These are described later in the document.

## Syntax:

8b-vmsg, 8b-vmsg_req

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 255 | Message Identifier |
| vmsg_req | 1 byte | variable | Message identifier of requested msg |
| variable | variable | variable | Optional arguments for request |

## Response:

Requested Message

# FA_J1939 Filter Add J1939 FA_J1939

## Description:

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

## Syntax:

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

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| 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 PGN 65236 to port 0.

1/0/0/254/212

# FA_EXT_J1939 Low Pass Filter Add FA_EXT_J1939

## Description:

This message adds a PGN to the low pass acceptance filter the desired port. Filters of this type will only pass the requested message at the specified time base..

## Syntax:

8b-vmsg, 8b-port, 24b-pgn, 16b-time

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 50 | Message Identifier |
| port | 1 byte | 0 or 1 | J1939 Port Number |
| pgn | 3 bytes | variable | Parameter Group Number (MSB first) |
| time | 2 bytes | variable | Time between messages (ms) |

## Response:

ACK

## Example:

Add low pass acceptance filter for the PGN 65236 to port 0 at a rate of 1 second.

50/0/0/254/212/3/232

# FD_J1939 Filter Delete J1939 FD_J1939

## Description:

This message deletes a PGN from the acceptance filter for the desired port.

## Syntax:

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

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| 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 PGN 65236 on port 0.

2/0/0/254/212

# FD_EXT_J1939 Low Pass Filter Delete FD_EXT_J1939

## Description:

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

## Syntax:

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

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 51 | 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 PGN 65236 to port 0.

51/0/0/254/212

# FA_J1587 Filter Add J1587 FA_J1587

## Description:

This message adds a PID filter to the J1587 port.

## Syntax:

8b-vmsg, 16b-pid

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 3 | Message Identifier |
| pid | 2 bytes | variable | PID (MSB first) |

## Response:

ACK

## Example:

Adds acceptance filter for the PID 12345.

3/48/57

# FD_J1587 Filter Delete J1587 FD_J1587

## Description:

This message deletes a PID filter to the J1587 port.

## Syntax:

8b-vmsg, 16b-pid

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 4 | Message Identifier |
| pid | 2 bytes | variable | PID (MSB first) |

## Response:

ACK

## Example:

Deletes acceptance filter for the PID 12345.

4/48/57

# TX_J1939 Transmit J1939 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:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 5 | Message Identifier |
| port | 1 byte | 0 or 1 | J1939 Port Number |
| pgn | 3 bytes | variable | Parameter Group Number (MSB first) |
| dst | 1 byte | variable | Destination Address |
| src | 1 byte | 43 | Source Address (ignored) |
| 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=43, pri=6, data=0 to 10

5/1/0/48/57/255/43/6/0/1/2/3/4/5/6/7/8/9/10

# RX_J1939 Receive J1939 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:

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

## Response:

None

## Example:

Receipt of a J1939 packet on port 1 with pgn=12345, dst=255, src=43, pri=6, data=0 to 10

6/1/0/48/57/255/43/6/0/1/2/3/4/5/6/7/8/9/10

# TX_J1587 Transmit J1587 TX_J1587

## Description:

This message schedules a J1587 message for transmission.

## Syntax:

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

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 8 | Message Identifier |
| mid | 1 byte | variable | J1587 MID |
| pid | 2 bytes | 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 J1587 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:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 9 | Message Identifier |
| mid | 1 byte | variable | J1587 MID |
| pid | 2 bytes | variable | Parameter Identifier |
| data | variable | variable | Data |

## Response:

None

## 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

# TX_I15765 Transmit I15765 TX_I15765

## Description:

This message schedules an I15765 message for transmission. The message will be sent as soon as the attached CAN network allows.

## Syntax:

8b-vmsg, 8b-port, 8b-dst, 8b-src, 8b-pri, 8b-tat, nB-data

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 42 | Message Identifier |
| port | 1 byte | 0 or 1 | OBD2 Port Number |
| dst | 1 byte | variable | Destination Address |
| src | 1 byte | 252 | Source Address |
| pri | 1 byte | 0 to 7 | Priority |
| tat | 1 byte | 118, 119, 218, 219 | Target Address Type |
| data | variable | variable | Data |

## Response:

ACK

## Example:

Schedules an OBD2 RPM request packet on port 1 with , dst=1, src=252, pri=6, data=1, 12

42/1/1/252/6/119/1/12

# RX_I15765 Receive I15765 RX_I15765

## Description:

This message is sent to the host device when the VNA receives an I15765 message that passes acceptance filtering.

## Syntax:

8b-vmsg, 8b-port, 8b-dst, 8b-src, 8b-pri, 8b-tat, nB-data

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 43 | Message Identifier |
| port | 1 byte | 0 or 1 | OBD2 Port Number |
| dst | 1 byte | variable | Destination Address |
| src | 1 byte | variable | Source Address |
| pri | 1 byte | 0 to 7 | Priority |
| tat | 1 byte | 118, 119, 218, 219 | Target Address Type |
| data | variable | variable | Data |

## Response:

None

## Example:

Receipt of an OBD2 RPM response packet on port 1 with dst=241, src=1, pri=6, data=65/12/172/1

43/1/241/1/6/118/65/12/172/1

# ODO Odometer ODO

## Description:

This message chirps the current digital odometer reading of the connected vehicle. The ODO message is sent to the host device every second unless disabled with the CHIRPCON message. The odometer is transmitted in tenths of a mile. The digital odometer is calculated using wheel speed from the vehicle network.

You can write a new odometer value by sending an ODO message with the desired value to the ELD. The new value becomes a starting point for continued accumulation.

## Syntax:

8b-vmsg, 32b-odo

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 46 | Message Identifier |
| odo | 4 bytes | variable | Odometer reading (tenth of mile) |

## Response:

None

## Example:

Receipt of the odometer reading with value of 25.5 miles.

43/0/0/0/255

# GPS GPS Location GPS

## Description:

This message chirps the current location from GPS (if equipped). The GPS message is sent to the host device every second.

## Syntax:

8b-vmsg, 1b-latdeg, 1b-latmin, 3b-latdec, 1b-latdir, 1b-londeg, 1b-lonmin, 3b-londec, 1b-londir

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 69 | Message Identifier |
| latdeg | 1 byte | Variable | Degrees of Latitude |
| latmin | 1 byte | Variable | Whole minutes of Latitude |
| latdec | 3 bytes | Variable | Decimal min of Latitude (1/100000) |
| latdir | 1 byte | Variable | ASCII latitude (N/S) |
| londeg | 1 byte | Variable | Degrees of Longitude |
| lonmin | 1 byte | Variable | Whole minutes of Longitude |
| londec | 3 bytes | Variable | Decimal min of Longitude (1/100000) |
| londir | 1 byte | Variable | ASCII Longitude (E/W) |

## Response:

None

# REC_CNT Record Total REC_CNT

## Description:

This message returns the number of records stored on the device. This message is sent upon request using the VMSG_REQ format described previously.

## Syntax:

8b-vmsg, 32b-rtotal

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 80 | Message Identifier |
| rtotal | 4 bytes | variable | Number of Records Stored |

## Response:

None

# REC_BATCH Record Batch REC_BATCH

## Description:

This message returns a number of records stored on the device. More information about the format of each record which could be returned is outlined in the VNA2-ELD Record Appendix. This message is sent upon request using the VMSG_REQ format described previously.

## Syntax:

8b-vmsg, 8b-recnum, vb-recdata

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 81 | Message Identifier |
| recnum | 1 byte | variable | Number of Records Returned |
| recdata | variable | variable | Records as outlined in Appendix |

## Request Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg_req | 1 byte | 255 | Message Identifier |
| vmsg | 1 byte | 81 | Message Identifier |
| recnum | 1 byte | variable | Number of Records Requested (max 8) |

## Response:

None

# REC_DEL Record Discard REC_DEL

## Description:

This message discards records which have been transferred to the connected host previously. This message is sent using the VMSG_REQ format described previously.

## Syntax:

8b-vmsg

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 82 | Message Identifier |

## Response:

ACK

# LOGIN Driver Login LOGIN

## Description:

This message allows a driver to be logged into the VNA. After the Login message is received, all records will be stored with the logged-in drivers driver id. If no driver is logged in, the driver id will be 0xFFFFFFFF. The driver can be logged out using the LOGOUT message described on the following page. If a login message is sent, while another driver is already logged in, the new driver id will be saved.

## Syntax:

8b-vmsg, 32b-driverid

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 85 | Message Identifier |
| driverid | 1 byte | variable | Driver Identifier |

## Response:

ACK

# LOGOUT Driver Logout LOGOUT

## Description:

This message logs a driver out from the VNA2-ELD record system. After this message is received, records will no longer contain a valid driver id, instead 0xFFFFFFFF will be stored.

## Syntax:

8b-vmsg

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 86 | Message Identifier |

## Response:

ACK

# ACONN Automatic Connection ACONN

## Description:

This message returns the status of the automatic baud-rate and network detection schemes. It is sent upon request using the VMSG_REQ format described previously. It is also sent upon completion of network detection on OBD2 enabled devices.

## Syntax:

8b-vmsg, 8b-port, 32b-baudrate, 32b-nettype

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 52 | Message Identifier |
| port | 1 byte | 0 or 1 | CAN port |
| baudrate | 4 bytes | variable | Baud Rate of Network |
| nettype | 4 bytes | variable | Network Type |

## Request:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg_req | 1 byte | 255 | Message Identifier |
| vmsg | 1 byte | 52 | Message Identifier |
| port | 1 byte | 0 or 1 | CAN port |

## Response:

None

# MAC BT MAC Address MAC

## Description:

This message returns the MAC address of the VNA2-ELD's Bluetooth radio. It is sent upon request using the VMSG_REQ format described previously.

## Syntax:

8b-vmsg, 48b-mac

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 71 | Message Identifier |
| mac | 6 bytes | variable | MAC Address of Bluetooth Radio |

## Response:

ACK

# 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. It can be disabled with chirpcon.

## Syntax:

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

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| 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

# CHIRPCON Chirp Control CHIRPCON

## Description:

This message controls the periodic transmission of STATS1 (23) and ODO (46). To enable these messages set the corresponding bit to 1 in the bmask argument. To disable set the corresponding bit to 0. The top 28 bits are reserved for future messages.

| B31-B4 | B2 | B1 | B0 |
|---|---|---|---|
| X | ODO | X | STATS1 |

## Syntax:

8b-vmsg, 32b-bmask

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 64 | Message Identifier |
| stat0 | 4 bytes | variable | Control bitmask |

## Response:

None

# Chapter 4

# Bootloader

The VNA2-ELD is equipped with a robust bootloader which allows the firmware to be upgraded during a deployment. The bootloader messages receive an image at runtime, verifies the image, and resets the hardware to copy the image into runtime execution.

We require users to include the ability to interface with our bootloader in their applications. Simma Software may need to deploy bug fixes or feature enhancements while the VNA2 hardware is deployed.

The VNA2-ELD uses three messages which are described on the following pages.

# BL_START Bootloader Start BL_START

## Description:

The bootloader start message begins the bootloading process. It should be sent one time, before any data is transfered.

## Syntax:

8b-vmsg

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 100 | Message Identifier |

## Response:

Success: MSG_RESULT, BL2_START, 0
Failure: MSG_RESULT, BL2_START, non-zero

# BL_DATA Bootloader Data BL_DATA

## Description:

The bootloader data message transmits data packets to the bootloader. You may send up to 250 bytes of data per message. You must wait for an ACK before sending another message. If no ACK is received after 1 second, restart the process with BL_START.

## Syntax:

8b-vmsg; 4b-addr; vb-data

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 100 | Message Identifier |
| addr | 4 bytes | Variable | Bootloader Address |
| data | 1-100 bytes | Variable | Bootloader Data |

## Response:

Success: MSG_RESULT, BL2_DATA, 0
Failure: MSG_RESULT, BL2_DATA, non-zero

# BL_STOP Bootloader Stop BL_STOP

## Description:

The bootloader stop message ends the bootloading transfer. Upon receipt of the message, the VNA2-ELD marks the transferred image as valid and reboots. The image will be transferred into working flash during the reboot. This message **should not** be sent unless the image was transferred successfully.

## Syntax:

8b-vmsg

## Definition:

| *Name* | *Size* | *Value* | *Description* |
|---|---|---|---|
| vmsg | 1 byte | 100 | Message Identifier |

## Response:

Success: MSG_RESULT, BL2_STOP, 0
Failure: MSG_RESULT, BL2_STOP, non-zero

# Chapter 5

# Additional Functionality

The VNA2-ELD has a number of additional functional features which expand the usability in data logging, long term deployments, and interoperability with iOS devices. These features are described in detail below.

## Reset

The VNA2-ELD is equipped with reset functionality which causes the device to reset if the VNA believes the Bluetooth connection has failed. This functionality prevents the VNA from entering a condition which would require driver intervention to clear. The VNA will reset 15 seconds after a disconnection to make sure everything is in order for a new connection. The Reset function requires the user to send a message to the VNA once a second while connected. We recommend either sending a filter for a message which is already set, or sending an ACK.

## Sleep

The VNA2-ELD is equipped with a sleep mode designed to preserve battery life on OBD vehicles. The VNA2-ELD will go to sleep after 5 minutes when connected to a vehicle which has its engine turned off. The VNA2-ELD will wake up without intervention when it detects the vehicle has started again. Users should be aware that the VNA2-ELD cannot be connected to during sleep mode. If reconnection is required while the VNA2-ELD is in sleep mode, the engine must be restarted. Users should still be conscious about leaving the VNA2-ELD or any other device connected to the diagnostic port if the vehicle will be unused for an extended period of time.

## Bluetooth Low Energy

The VNA2-ELD uses a dual mode Bluetooth transceiver to allow for standard VNA commands to be transferred over either Bluetooth Classic (2.0) or Bluetooth Low Energy.

When the VNA is connected to via Bluetooth Classic it connects using SPP and behavior is the same as outlined in this document.

Over Bluetooth Low Energy the VNA connects via a custom GATT profile. This profile reveals a service with two characteristics. One characteristic is writeable and the other characteristic is both readable and notifiable. The writable characteristic should be used to send data to the VNA. The readable characteristic will reveal no data when read; instead, a host should subscribe to notifications on this characteristic. The VNA will return data via notifications in the same format as specified in the above document.

The service and characteristics can be identified by the following UUIDs:

Service: 4439C0C0-BD26-11E4-AAA3-0002A5D5C51B

Read Characteristic: F8287520-BD28-11E4-B64D-0002A5D5C51B

Write Characteristic: C188C680-BD2A-11E4-BC8D-0002A5D5C51B

In both connection modes the message framing and format protocols are the same as outlined in Chapters 2 and 3 of the VNA-232 user's manual.

# Chapter 6

# Examples

Simma Software provides numerous examples of VNA2-ELD functionality for PC, Android, and iOS. The following is a list of programs available for each platform. All of the example applications are available as both source code and pre-built executables.

## PC

- pc232: Complete functionality of the VNA2-ELD
- pc232-gps: Processes the 1 second GPS message
- pc232-records: Requests, prints, and removes all records from the VNA2-ELD
- pc232-j1939: Sets filters and prints received J1939 data
- pc232-obd2: Requests and prints received OBD2 data

## Android

- Android-j1939: Sets filters and prints received J1939 data

## iOS

- iOS-j1939: Sets filters and prints received J1939 data

# Chapter 7

# Physical Connections

This chapter describes how to connect the VNA2-ELD to both to your vehicle and your computer if using RS-232.

## Vehicle

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.

Power must be provided to the VNA-232 through the DB15. The power supply internal to the device features an input range of 8 to 30V.

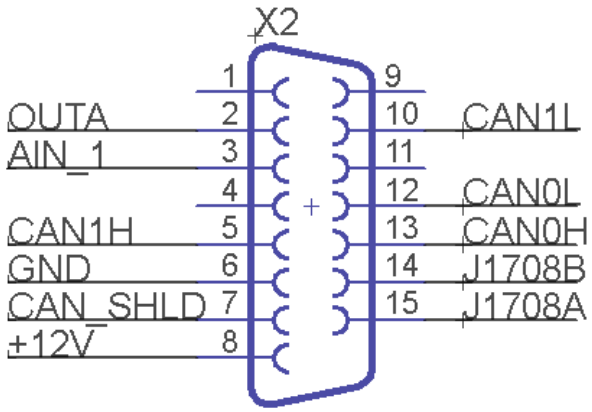


Figure 7.1: VNA2-ELD Vehicle Connection

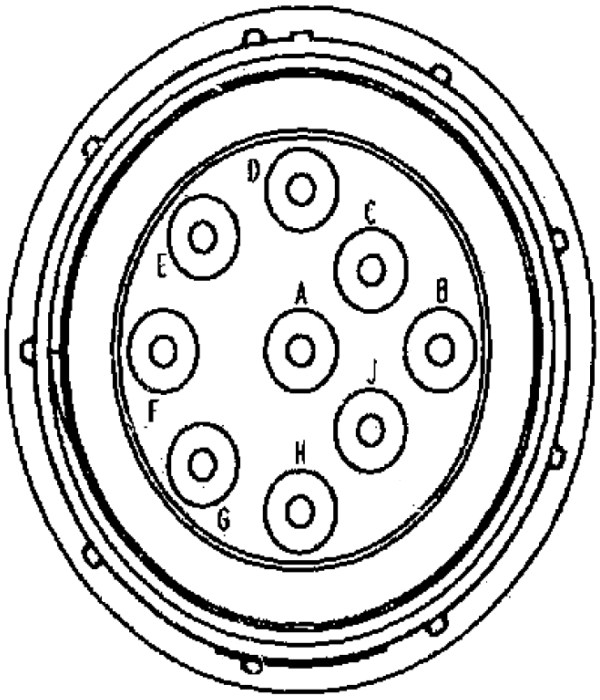


Figure 7.2: Deutsch Connector Pinout

| 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 7.1: Standard DB15 compared to Deutsch HD pinout

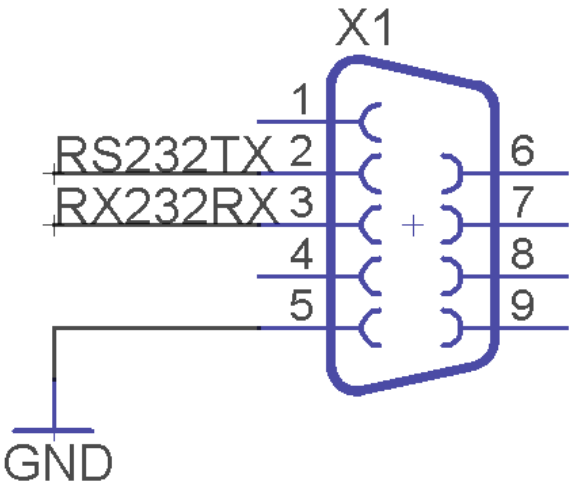


Figure 7.3: DB9 Pinout

## Computer

The VNA is designed to communicate to a host computer/embedded device via a standard RS-232 connection. Connection on the VNA is implemented using a female DB9, which should allow for the use of a standard straight-through DB9 serial cable. Refer to figure 4-3 for DB9 pinout schematic.

# Chapter 8

# Specifications

## Physical

- Serial Connection: DB-9 Female
- Vehicle Connection: DB-15 Female
- Temperature Rating: -40 to +85C (+125C available)

## Electrical

- 8-60 VDC Input, 0.5W max
- Load Dump and Input Power Reverse Bias Protection
- All communications lines feature 8kV or higher IEC ESD protection
- CAN transceiver meets or exceeds ISO 11898 standard
- J1587 transceiver fully compliant with TIA/EIA-485A

## Functional

- 100 Individual J1939 acceptance message filters
- 100 Individual J1587/J1708 acceptance message filters
- 5 Individual Low Pass J1939 acceptance message filters
- CAN 125kbps to 1Mbps
- RS-232 9.6kps to 1Mbps
- Bluetooth 4.1 Dual-Mode