

APN-083

CAN Bus Communication for OEM7 Products

Contents

Introduction	3
NovAtel CAN Implementation	4
Message Transmission Over CAN	4
Message Reception and Reassembly	5
Hardware Setup for CAN Communication	6
CAN Transceivers, Termination, and Resistors	6
CAN Pinouts for OEM7 Receiver Cards	6
CAN Pinouts for OEM7 Enclosure Products	7
Sample Setup	7
Software Configuration for CAN Communication.....	8
Software Configuration for CAN on OEM6	8
Software Configuration for CAN on OEM7	8
Enabling CAN.....	9
Verifying CAN Communication	10
Requesting NovAtel Messages or Corrections Over CAN	11
Configuring OEM7 CAN to behave as OEM6 CAN.....	15
Appendix A: CAN Commands.....	16
CAN Command Summary.....	16
Appendix B: CAN Logs	17
CAN Log Summary.....	17
NMEA2000 Fixed-Format Messages.....	17
NovAtel Proprietary Fixed-Format Messages	18
INSPVACMP	18
INSPVASDCMP.....	21
Appendix C: Full Configuration Examples	23
Example 1: Configuring the CAN bus to output NMEA2000 messages	23
Example 2: Configuring the CAN bus to output NovAtel messages	23

Introduction

CAN, which stands for Controller Area Network, is a communication medium becoming increasingly adopted by all markets. The purpose of CAN is to allow for data exchange between devices using a multi-master serial data communication model. The main advantage of CAN is that multiple devices can communicate through two inexpensive wires, that act as a BUS (Referred to as the CANBUS). This eliminates the need for one-to-one interconnect or dedicated network controllers.

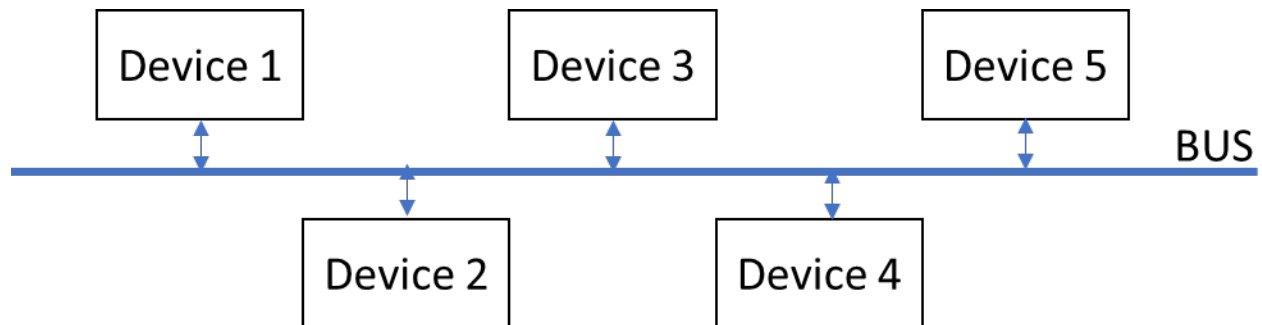


Figure 1: CANBUS Visualization

Figure 1 illustrates an example of a CAN network. Each device is electrically attached to the 2-wire BUS. BUS architectures like CAN allow devices to share data, without the need for each device to be specifically designed to work with each other device on the BUS. Each device (or “Node”) on the BUS can read all transmissions on the BUS.

Each node must identify itself on the BUS. This is what is known as “claiming an address”. Each node must claim a unique BUS address so the other nodes can communicate. Backup addresses can be set for the event that an address is already claimed. The system will attempt to use the backup as the next address.

This document will describe in detail how to install, configure, and use CAN as a communication method with NovAtel OEM7 products.

NovAtel CAN Implementation

NovAtel's CAN implementation is **J1939 based**, which is a standard originally built to support heavy duty vehicles such as large trucks and tractors. **For details on the J1939 CAN frame structure, refer to the J1939 standard.**

The NovAtel J1939 transport mechanism emulates a serial port, using CAN protocol as a physical layer. Standard CAN frames are used, with data fields up to 8 bytes long, 29-bit ID, and PGNs / Addresses specified per the [CCOMCONFIG](#) command.

NOTE: Detailed Logs and Commands information is available at:

docs.novatel.com/OEM7/Content/Home.htm

Message Transmission Over CAN

NovAtel messages are transmitted over CAN according to the following guidelines:

1. The entire message (Header + CRC + Data) is transported over CAN as-is, with no modifications
2. The message is broken into segments ≤ 8 bytes, in order for each segment to fit into the standard 8-byte J1939 frame.
3. When the segment is 8 bytes long, CAN DLC is set to 8.
4. When the segment is < 8 bytes (the last segment in the message could be less than 8 bytes), the DLC is set to the actual segment length. The remaining bytes in the CAN frame are set to FF. That is, no CAN frame ever carries data from more than 1 NovAtel message.
5. The segments are transmitted on CAN bus sequentially. There is no retransmission or acknowledgement, beyond what is specified by the J1939 protocol.
6. There is no guarantee as to the transmission timing of individual J1939 frames comprising NovAtel message.
7. There is no inter-message padding. That is, all frames carry some form of valid payload.

For example, the following is the result of sending the command "LOG CCOM1 BESTPOSA ONCE" to a receiver that is properly configured for CAN communication, as recorded by an open source CAN bus analysis program:

```
<Time><Tx/Rx><Channel><CAN ID><Type><DLC><DataBytes>
16:06:25:0840 Rx 1 0x1C06111C x 8 23 42 45 53 54 50 4F 53
16:06:25:0843 Rx 1 0x1C06111C x 8 41 2C 43 43 4F 4D 31 2C
16:06:25:0846 Rx 1 0x1C06111C x 8 30 2C 31 39 2E 30 2C 46
16:06:25:0851 Rx 1 0x1C06111C x 8 49 4E 45 53 54 45 45 52
16:06:25:0853 Rx 1 0x1C06111C x 8 49 4E 47 2C 32 30 34 34
16:06:25:0856 Rx 1 0x1C06111C x 8 2C 31 36 36 30 30 33 2E
16:06:25:0859 Rx 1 0x1C06111C x 8 30 30 30 2C 30 32 30 34
16:06:25:0862 Rx 1 0x1C06111C x 8 30 30 32 30 2C 62 31 66
16:06:25:0865 Rx 1 0x1C06111C x 8 36 2C 31 34 37 33 36 3B
16:06:25:0868 Rx 1 0x1C06111C x 8 53 4F 4C 5F 43 4F 4D 50
16:06:25:0871 Rx 1 0x1C06111C x 8 55 54 45 44 2C 53 49 4E
16:06:25:0875 Rx 1 0x1C06111C x 8 47 4C 45 2C 35 31 2E 31
16:06:25:0878 Rx 1 0x1C06111C x 8 35 30 33 38 35 34 31 36
```

```
16:06:25:0881 Rx 1 0x1C06111C x 8 34 36 2C 2D 31 31 34 2E
16:06:25:0884 Rx 1 0x1C06111C x 8 30 33 30 37 30 34 34 37
16:06:25:0887 Rx 1 0x1C06111C x 8 33 32 31 2C 31 30 39 38
16:06:25:0889 Rx 1 0x1C06111C x 8 2E 31 33 30 31 2C 2D 31
16:06:25:0892 Rx 1 0x1C06111C x 8 37 2E 30 30 30 30 2C 57
16:06:25:0895 Rx 1 0x1C06111C x 8 47 53 38 34 2C 31 2E 30
16:06:25:0899 Rx 1 0x1C06111C x 8 31 34 38 2C 30 2E 38 39
16:06:25:0902 Rx 1 0x1C06111C x 8 37 38 2C 32 2E 33 36 34
16:06:25:0905 Rx 1 0x1C06111C x 8 35 2C 22 22 2C 30 2E 30
16:06:25:0908 Rx 1 0x1C06111C x 8 30 30 2C 30 2E 30 30 30
16:06:25:0910 Rx 1 0x1C06111C x 8 2C 33 35 2C 33 31 2C 33
16:06:25:0913 Rx 1 0x1C06111C x 8 31 2C 33 31 2C 30 30 2C
16:06:25:0916 Rx 1 0x1C06111C x 8 30 36 2C 33 35 2C 33 33
16:06:25:0919 Rx 1 0x1C06111C x 8 2A 63 64 62 66 63 32 34
16:06:25:0921 Rx 1 0x1C06111C x 3 65 0D 0A
```

This demonstrates how a single message is broken into segments to fit into standard 8-byte CAN frames. If the data bytes within these messages are converted from Hexadecimal to ASCII, the original BESTPOSA message can be reconstructed, including the header and CRC:

```
#BESTPOSA,CCOM1,0,19.0,FINESTEERING,2044,166003.000,02040020,b1f6,147
36;SOL_COMPUTED,SINGLE,51.15038541646,-114.03070447321,1098.1301,-
17.0000,WGS84,1.0148,0.8978,2.3645,"",0.000,0.000,35,31,31,31,00,06,3
5,33*cdbfc24e
```

Message Reception and Reassembly

NovAtel messages are received and reassembled over CAN as follows:

1. The J1939 frame payload is extracted based on the DLC value.
2. The payloads are buffered the same way reads from a serial port would be buffered.
3. The buffer is parsed per NovAtel BINARY format.

Hardware Setup for CAN Communication

CAN uses two wires for communication. These signal lines are referred to as CAN High (CAN+) and CAN Low (CAN-). The difference between the two lines is known as the voltage differential. The inactive state between these two lines is 0V. When data is transmitted, the voltage differential is approximately 2.5V. This voltage differential, similar to a twisted pair differential, is resilient against EMI interference, electrical fields/spikes, and other noise.

CAN Transceivers, Termination, and Resistors

Both ends of the CAN BUS must be terminated. If is not properly terminated at both ends, then any signals on the bus get reflected from the ends and interfere with normal communication. When the ends of the BUS are properly terminated with 120Ω resistors, there are no reflections from the ends of the bus and all the nodes may communicate as intended.

There are no CAN transceivers on the OEM7 receiver cards. These cards require external CAN transceivers and proper bus terminations. See Figure 4 for an example of a CAN transceiver circuit. Some OEM7 enclosure products do include a CAN transceiver, which will be discussed in following sections.

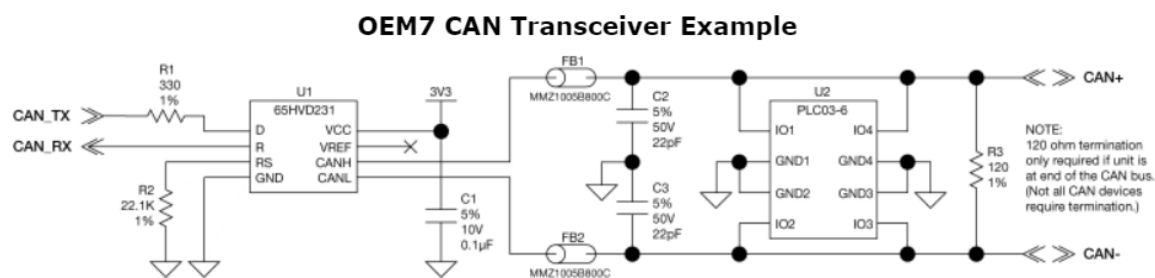


Figure 1: CAN Transceiver

For more information on the CAN transceiver including CAN transceiver components, please refer to:

https://docs.novatel.com/OEM7/Content/Interface_Circuits/CAN_Controller_Ports.htm

CAN Pinouts for OEM7 Receiver Cards

The following table shows the CAN pin designations for OEM7 receiver cards.

	OEM719	OEM729	OEM7600	OEM7700	OEM7720
Connector	P1701	P1803	P1701	P2001	P1901
CAN1TX	7	10	36	36	36
CAN1RX	6	11	38	38	38
CAN2TX	20	12	37	37	37
CAN2RX	8	13	35	35	35

Table 1: OEM Receiver CAN Pins

On the OEM719, CAN1 is multiplexed with user VARF and EVENT2, so the following commands must be issued before enabling CAN1:

```
FREQUENCYOUT DISABLE
MARKCONTROL MARK2 DISABLE
```

For complete OEM7 receiver port pinouts, please refer to the following documentation:

https://docs.novatel.com/OEM7/Content/Technical_Specs_Receiver/Technical_Specifications.htm

CAN Pinouts for OEM7 Enclosure Products

NovAtel PwrPak7 enclosure products come with an internal CAN transceiver. However, it still requires 120 Ohm bus terminators to function properly. Standalone OEM7 receiver card products require their own CAN transceiver, as was discussed in previous sections.

	PwrPak7/PwrPak7D
Connector	DSUB HD26
CAN1+ (High)	9
CAN1- (Low)	18

Table 2: PwrPak7/PwrPak7D CAN Pins

For complete PwrPak7/PwrPak7D enclosure port pinouts, please refer to the following documentation:

https://docs.novatel.com/OEM7/Content/Technical_Specs_Receiver/PwrPak7_Connectors.htm

Sample Setup

In the below picture, you can see a sample set-up. This includes a 120 Ohm terminator, the wiring coming from the two CAN lines on the PwrPak7 (CAN+, CAN-), and a Vector CAN interface convertor which takes the CAN output and converts it to USB to communicate with a PC. There are various third-party software programs available which can then decode and analyze the CAN output.



Figure 2: Sample Setup

Software Configuration for CAN Communication

This section will describe how to configure a receiver for CAN communication after the hardware has been correctly connected.

Software Configuration for CAN on OEM6

For CAN configuration on previous generation SPAN products (OEM6 and earlier), please refer to the “Configure CAN for SPAN” Application Note, available here:

<https://www.novatel.com/assets/Documents/Bulletins/apn046.pdf>

OEM6 products had the same NMEA2000 PGNs and NovAtel Proprietary Fixed-Format Messages available over CAN ports. See Appendix B for details.

Software Configuration for CAN on OEM7

By default, CAN is disabled on OEM7 receivers. Critical CAN configuration parameters such as Parameter Group Numbers (PGNs), addresses and priorities are system-specific and should be explicitly configured.

By default or after a FRESET, the receiver has the following CAN configuration:

- All CAN physical ports are disabled
- No J1939 addresses are claimed; default CAN NAME and address are configured
- CCOM ports are configured for NMEA2000 messages and serial emulation using single J1939 frames only

Enabling CAN

Optional

Use the [J1939CONFIG](#) command to specify a custom J1939 NAME and desired address. If not entered, default parameters for node, port, address ranges, and a manufacturing code of 305 will be used when address claiming begins.

Claiming an Address

To enable an OEM7 receiver to communicate over the CANBUS, use the [CANCONFIG](#) command to place the receiver “on bus”.

To begin the address claim procedure, issue the following command:

```
CANCONFIG <PORT> ON <PORT SPEED>
```

For example:

```
CANCONFIG CAN1 ON 500K
```

This address claim will use the parameters selected in the J1939CONFIG command (or the defaults, if the J1939CONFIG command was not used).

Releasing the Address

The receiver must release the address and be taken “off bus” to change CAN port settings.

```
CANCONFIG <PORT> OFF
```

For example:

```
CANCONFIG CAN1 OFF -
```

Verifying CAN Communication

To monitor the CAN status and verify that CAN communication is possible, use the [J1939STATUS](#) log. This log reports the status of the address claim procedure on the J1939 CAN node(s) as one of:

- DISABLED
- CLAIMING
- CLAIMED
- FAILED

If the status reports CLAIMED, this log also displays the claimed CAN address.

To monitor the CAN status using the J1939STATUS log, enter the following command:

```
LOG J1939STATUS ONCHANGED
```

The following is an example of what the J1939STATUS log might report before, during, and after the address claim procedure initiated by using the CANCONFIG command. By default, NODE1 is associated with CCOM1 using J1939 communication protocol.

```
[USB2]LOG J1939STATUS ONCHANGED
<OK

[USB2]<J1939STATUS USB2 1 80.0 UNKNOWN 0 0.000 02004020 e9ce 14970
<  NODE1 DISABLED 0 FE
<J1939STATUS USB2 0 80.0 UNKNOWN 0 0.000 02004020 e9ce 14970
<  NODE2 DISABLED 0 FE

[USB2]CANCONFIG CAN1 ON 500K
<OK

[USB2]<J1939STATUS USB2 0 76.0 FINESTEERING 2031 428150.317 0200c020
e9ce 14970
<  NODE1 CLAIMING 1 1C

[USB2]<J1939STATUS USB2 0 80.0 FINESTEERING 2031 428150.921 02004020
e9ce 14970
<  NODE1 CLAIMED 1 1C
```



Requesting NovAtel Messages or Corrections Over CAN

Log messages can be requested over the CAN bus ports once the software is configured for CAN and an address has been successfully claimed on the CAN bus. The receiver's CAN ports are designated as "CCOM" ports, i.e. CCOM1, CCOM2, etc.

Before being used, a CCOM port must first be associated with a J1939 node using the [CCOMCONFIG](#) command. The J1939 node must be active as per the "Enabling CAN" section above.

Once this is done, the user must decide what logs they wish to transmit over CAN. There are several NMEA2000 format logs, two NovAtel proprietary fixed-format logs, and the entire library of standard NovAtel logs to choose from. Each type will be described in more detail below.

NMEA2000 log configuration

All NMEA2000 logs are configured using the LOG command, where the destination port is a CAN port (CCOM<x>). For NMEA2000 logs, the desired CCOM port must first be associated with a J1939 node and set to "NMEA2000" CAN transport protocol using the [CCOMCONFIG](#) command. For example:

```
CCOMCONFIG CCOM1 NODE1 NMEA2000
```

Then, enable the desired NMEA2000 logs using the [LOG](#) command as usual.

The steps to enable NovAtel proprietary fixed-format messages over CAN are:

- 1) Configure the CAN Bus (see the "Enabling CAN" section above)
- 2) Use the [CCOMCONFIG](#) command to configure the CCOM port protocol for NMEA2000
- 3) Enable the desired logs using the [LOG](#) command as usual

The following example shows the configuration of CCOM1 to output two NMEA2000 logs:

```
CCOMCONFIG CCOM1 NODE1 NMEA2000
LOG CCOM1 PGN129025 ONTIME 1
LOG CCOM1 PGN129026 ONTIME 1
SAVECONFIG
```

Note that for transmitting NMEA logs, the INTERFACEMODE of the CCOM port must be set to NOVATEL.

See Appendix B for a full list of the available NMEA2000 logs.



NovAtel proprietary fixed-format log configuration

In addition to the standard NMEA2000 PGN messages, there are two NovAtel proprietary fixed-format SPAN messages that use the NMEA2000 Fast Packet Protocol. These two logs are called INSPVACMP (Optimized GNSS/INS Position, Velocity and Attitude) and INSPVASDCMP (Optimized GNSS/INS standard deviations and update status). They are intended for high-rate, fixed-format SPAN output over the CAN bus.

The INSPVACMP and INSPVASDCMP logs have specific PGN numbers, which default to 130816 and 130817, respectively. Optionally, the user may specify a custom PGN for these two logs by using the [PGNCONFIG](#) command. If custom PGNs are configured, it is strongly recommended to RESET the receiver after using the PGNCONFIG command (and issuing a SAVECONFIG to store your PGNCONFIG settings). This prevents PGN ambiguities and conflicts.

The steps to enable NovAtel proprietary fixed-format messages over CAN are:

- 1) Configure the CAN Bus (see the “Enabling CAN” section above)
- 2) Use the [CCOMCONFIG](#) command to configure the CCOM port protocol for NMEA2000. Note that the PGN, Priority, and Address parameters of the [CCOMCONFIG](#) command are ignored if NMEA2000 protocol is used.
- 3) Optionally, use the PGNCONFIG command to configure a custom PGN for these logs
- 4) Enable the desired logs using the [LOG](#) command as usual

For example:

```
CCOMCONFIG CCOM1 NODE1 NMEA2000
PGNCONFIG INSPVACMP 129500 3
LOG CCOM1 INSPVACMP ONTIME 1
SAVECONFIG
RESET
```

For more detail on these two logs, see Appendix B.

NovAtel log configuration

Standard NovAtel messages (commands, logs, responses) can also be sent and received on the CAN Bus using the CCOM ports.

- A single CCOM port cannot be used for both Binary and ASCII / NovAtel ASCII messages.
- A single CCOM port cannot be used for both Binary messages and corrections.
- If the CCOM port is configured as NOVATEL interface mode, all input is interpreted as NovAtel ASCII or Abbreviated ASCII. Unlike other COM ports, the receiver will not distinguish between ASCII and binary input.
- It is recommended to use one dedicated CCOM port for NovAtel messages and another dedicated CCOM port for corrections.
- NovAtel UI configuration does not affect standard NMEA logs such as GPGGA, GPHDT, etc. Any CCOM port can be used for standard NMEA logs irrespective of CCOMCONFIG settings.

The steps to enable NovAtel messages over CAN are:

- 1) Configure the CAN Bus (see the “Enabling CAN” section above)
- 2) Use the [CCOMCONFIG](#) command to configure the PGN and other desired CAN parameters
- 3) Use the [INTERFACEMODE](#) command to configure the CCOM port for the correct format (NOVATELBINARY recommended)
- 4) Enable the desired NovAtel logs using the [LOG](#) command as usual

For example:

```
CCOMCONFIG CCOM2 NODE1 J1939 61184 7 fe
INTERFACEMODE CCOM2 NOVATELBINARY NOVATELBINARY OFF
LOG CCOM2 BESTPOSB ONTIME 1
LOG CCOM2 BESTVELB ONTIME 1
SAVECONFIG
```

A single CCOM port cannot be used for both Binary and ASCII / NovAtel ASCII messages simultaneously, nor can it be used for both Binary messages and correction messages simultaneously. If using correction messages, it is recommended to use one dedicated CCOM port for NovAtel messages and another dedicated CCOM port for corrections.

If the CCOM port is configured with the NOVATEL interface mode, all input is interpreted as NovAtel ASCII or Abbreviated ASCII. Unlike other COM ports, the receiver will not distinguish between ASCII and binary input in this case.

Sending or Receiving Correction Messages over CAN

All NovAtel-supported correction formats are supported over CAN ports (CCOM).

To send or receive corrections over a CAN port:

- 1) Configure the CAN Bus (see the “Enabling CAN” section above)
- 2) Use the [CCOMCONFIG](#) command to configure the PGN and other CAN parameters used by the RTK corrections CAN messages.
 - a. PGN: There is no designated PGN for corrections. PGN is up to the user and BUS specific.
 - b. Address:
 - i. Use 0xFF to receive corrections from any CAN address and to broadcast corrections to all CAN nodes.
 - ii. Use 0x00 to 0xFD to send corrections to or receive corrections from a specific CAN node.
- 3) Use the [INTERFACEMODE](#) command to configure the CCOM interface mode.
 - a. To transmit corrections, use the desired INTERFACEMODE, e.g. RTCMV3
 - b. To receive corrections, it is recommended to use INTERFACEMODE AUTO.

For example, to configure a CCOM port to receive corrections from any source:

```
CCOMCONFIG CCOM2 NODE1 J1939 61184 6 0xFF
INTERFACEMODE CCOM2 AUTO NONE OFF
```

For example, to configure a CCOM port to transmit RTCMV3 corrections to a 0x1c node:

```
CCOMCONFIG CCOM2 NODE1 J1939 61184 6 0x1c
INTERFACEMODE CCOM2 NONE RTCMV3 OFF
LOG CCOM2 RTCM1004 ONTIME 1
LOG CCOM2 RTCM1012 ONTIME 1
LOG CCOM2 RTCM1006 ONTIME 10
LOG CCOM2 RTCM1033 ONTIME 10
LOG CCOM2 RTCM1019 ONTIME 120
LOG CCOM2 RTCM1020 ONTIME 120
```

See the “Transmitting and Receiving Corrections” section in the OEM7 documentation for more information, available here:

https://docs.novatel.com/OEM7/Content/Operation/Transmit_Receive_Corrections.htm

Configuring OEM7 CAN to behave as OEM6 CAN

To configure an OEM7 receiver to log the same commands and use the same logging rate and CAN port bit rate as the OEM6 CAN defaults, enter the following commands:

```
CCOMCONFIG CCOM1 NODE1 NMEA2000
CANCONFIG CAN1 ON 250K
LOG CCOM1 PGN129025 ONTIME 1
LOG CCOM1 PGN129026 ONTIME 1
LOG CCOM1 PGN129029 ONTIME 1
SAVECONFIG
```

Appendix A: CAN Commands

The following commands are relevant for CAN configuration and usage on OEM7 products.

For CAN commands on previous generation SPAN products (OEM6 and earlier), please refer to the “Configure CAN for SPAN” Application Note, available here:

<https://www.novatel.com/assets/Documents/Bulletins/apn046.pdf>

CAN Command Summary

Command/Log	Description
CANCONFIG	Controls the CAN transceiver hardware and places the receiver <i>on bus</i> or <i>off bus</i>
J1939CONFIG	Assigns the CAN J1939 NAME and address parameters to a Node
PGNCONFIG	Configures the NovAtel-proprietary NMEA2000 messages (to change the PGN and its priority)
CCOMCONFIG	Configures the CCOM port parameters used to output messages over CAN

Table 3: CAN Command Summary

Appendix B: CAN Logs

This section describes the logs messages available over CAN on OEM7 products. Six NMEA2000 format messages and two NovAtel proprietary format messages are available in a fixed-format structure that is optimized for CAN communication.

The remainder of the NovAtel message library is also available over CAN, but those messages do not have a fixed format and are output over CAN as ASCII strings in a CAN-compatible “wrapper”. The fixed-format messages are defined within a NovAtel DBC file (available upon request), while the remainder of the NovAtel messages are not.

All of the commands and logs mentioned in this document are described in full detail in the OEM7 documentation (<http://docs.novatel.com/OEM7/Content/Home.htm>) or in the OEM6 Firmware Reference Manual (www.novatel.com/assets/Documents/Manuals/om-20000129.pdf).

For CAN logging for SPAN on previous generation products (OEM6 and earlier), please refer to the “Configure CAN for SPAN” Application Note, available here:

<https://www.novatel.com/assets/Documents/Bulletins/apn046.pdf>

CAN Log Summary

Log Name	Log Description
J1939STATUS	Reports the status a Node on the J1939 CAN network, such as the claimed address
PGN126992	NMEA2000 System Time
PGN129025	NMEA2000 GNSS Position Rapid Update
PGN129026	NMEA2000 COG & SOG Rapid Update
PGN129027	NMEA2000 Position Delta High Precision Rapid Update
PGN129029	NMEA2000 GNSS Position
PGN129551	NMEA2000 GNSS Differential Signal
INSPVACMP	Optimized INS Position, Velocity and Attitude
INSPVASDCMP	Optimized INS Standard Deviation and Update Status

Table 4: CAN log summary

These logs will be described in greater detail below under their respective categories.

NMEA2000 Fixed-Format Messages

The NMEA2000 data format is based on the “NATIONAL MARINE ELECTRONICS ASSOCIATION (NMEA) NMEA2000 Standard rev 2.0 - January/2013”. NovAtel is not permitted to publish detailed descriptions of the NMEA2000 message structure, but the standard may be purchased directly from the NMEA.

OEM7 receivers support the following NMEA2000 Parameter Group Messages (PGN) over the CAN bus:

Log name	Log Description
PGN126992	NMEA2000 System Time
PGN129025	NMEA2000 GNSS Position Rapid Update
PGN129026	NMEA2000 COG & SOG Rapid Update
PGN129027	NMEA2000 Position Delta High Precision Rapid Update
PGN129029	NMEA2000 GNSS Position

PGN129551	NMEA2000 GNSS Differential Signal
-----------	-----------------------------------

Table 5: NMEA2000 Fixed-Format Message Summary

NovAtel Proprietary Fixed-Format Messages

The NovAtel Proprietary fixed-format logs are sent using the NMEA2000 Fast Packet Protocol and have specific PGN numbers set by the OEM7 PGNCONFIG command. The following table lists these logs which can be used to output high-rate position information over CAN:

Log Name	Log Description	PGN	Protocol	Data Size (Bytes)
INSPVACMP	Optimized INS Position, Velocity and Attitude	Defined at System Initialization	NMEA2000 <i>Fast Packet Protocol</i>	34
INSPVASDCMP	Optimized INS Standard Deviation and Update Status	Defined at System Initialization	NMEA2000 <i>Fast Packet Protocol</i>	31

Table 6: NovAtel Proprietary Fixed-Format Message Summary

INSPVACMP

INS Position, Velocity and Attitude Optimized for Ground Vehicles

This log contains INS position, velocity and attitude, with respect to the SPAN frame. By default, these values are output at the IMU center of navigation but can be offset and rotated to any point of interest via the [SETINSTRANSULATION](#) and [SETINSROTATION](#) commands, respectively.

The Position Status indicates whether the subsequent information in the message is valid. If the Position Status is not zero, then the position is valid, with a statistical accuracy estimated by the standard deviations given in the INSPVASDCMP log.

When Position Type = 0 ("INS_INACTIVE"), Latitude, Longitude, Altitude are not valid, nor are the velocities, attitude, and azimuth rate. The data should not be used for any purpose in this case.

Table 3-2: INSPVACMP Log Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Message Time	Message time, corresponding to time from the beginning of the GPS week time generated by the CGR	Long	4	0
2	INS Status	INS solution status (See “Inertial Solution Status” table in the online documentation portal under the INSATT log)	UChar	1	4
3	GNSS Position Type	Position type (see “Position or Velocity Type” table in the online documentation portal under the “BESTPOS” log)	UChar	1	5
4	Latitude	WGS84 latitude	5Byte Long	5	6
5	Longitude	WGS84 longitude	5Byte Long	5	11
6	Height	WGS84 height	Long	4	16
7	North Velocity	North velocity in local frame	Short	2	20
8	East Velocity	East velocity in local frame	Short	2	22
9	Up Velocity	Up velocity in local frame	Short	2	24
10	Roll	local frame	Short	2	26
11	Pitch	local frame	Short	2	28
12	Azimuth	local frame	UShort	2	30
13	Azimuth Rate	Rate of change of the Azimuth	Short	2	32

Table 7: INSPVACMP Log Field Description

The Scale Factors for the measurements above are presented in the following table.

Field	Description	Units	Scale Factor
Message Time	Message time, corresponding to time from the beginning of the GPS week time generated by the CGR	seconds	0.001
INS Status	INS solution status	unitless	1
Position Status	position type	unitless	1
Latitude	WGS84 latitude	degrees	180/239
Longitude	WGS84 longitude	degrees	180/239
Height	WGS84 height	meters	0.0001
North Velocity	north velocity in local frame	m/s	0.002
East Velocity	east velocity in local frame	m/s	0.002
Up Velocity	up velocity in local frame	m/s	0.002
Roll	local frame	degrees	0.01
Pitch	local frame	degrees	0.01
Azimuth	local frame	degrees	0.01
Azimuth Rate	Rate of change of the Azimuth	degrees/s	0.01

Table 8: INSPVACMP Log Field Formats and Scale Factors

INSPVASDCMP

Optimized INS Standard Deviation and Update Status

This log contains Standard Deviation information for the position, velocity, and attitude fields in the INSPVACMP log. It also includes indicators as to which update types have been applied to the inertial filter. This log is output at 1Hz only.

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	GPS Week	GPS Week	UShort	2	0
2	Message Time	Message time corresponding to time from the beginning of the GPS week time generated by the CGR	ULong	4	2
3	Sigma Latitude	Sigma Latitude Standard Deviation	UShort	2	6
4	Sigma Longitude	Sigma Longitude Standard Deviation	UShort	2	8
5	Sigma Height	Height Standard Deviation	UShort	2	10
6	Sigma North Velocity	North Velocity Standard Deviation	UShort	2	12
7	Sigma East Velocity	East Velocity Standard Deviation	UShort	2	14
8	Sigma Up Velocity	Up Velocity Standard Deviation	UShort	2	16
9	Sigma Roll	Roll Standard Deviation	UShort	2	18
10	Sigma Pitch	Pitch Standard Deviation	UShort	2	20
11	Sigma Azimuth	Azimuth Standard Deviation	UShort	2	22
12	Update Counter	Time since last ZUPT or position update	UChar	1	23
13	Position Type	INS Update Position Type (see “Position or Velocity Type” table under the “BESTPOS” log)	UChar	1	24
14	Extended Solution Status	Bitwise INS alignment and update information (see “Extended Solution Status” table under the “INSATTX” log)	Hex	4	26
15	ALIGNAge	Seconds since a valid ROVERPOS solution was available	UChar	1	30

Table 9: INSPVASDCMP Log Field Description

The Scale Factors for the measurements above are presented in the following table.

Field	Description	Units	Scale Factor
GPS Week	Weeks since GPS Week 0 (00:00, 6 January, 1980)	Weeks	1
Message Time	Message time, corresponding to time from the beginning of the GPS week time generated by the CGR	s	0.001
Sigma Latitude	Latitude Standard Deviation	meters	0.001
Sigma Longitude	Longitude Standard Deviation	meters	0.001
Sigma Height	Height Standard Deviation	meters	0.001
Sigma North Velocity	North Velocity Standard Deviation	m/s	0.001
Sigma East Velocity	East Velocity Standard Deviation	m/s	0.001
Sigma Up Velocity	Up Velocity Standard Deviation	m/s	0.001
Sigma Roll	Roll Standard Deviation	degrees	0.01
Sigma Pitch	Pitch Standard Deviation	degrees	0.01
Sigma Azimuth	Azimuth Standard Deviation	degrees	0.01
Update Counter	Time since last ZUPT or position update	seconds	1
Position Type	GNSS Update Position Type	unitless	1
Extended Solution Status	Bitwise INS alignment and update information, see “INS Extended Solution Status” Table.	unitless	N/A
ALIGN Age	Seconds since a valid ROVERPOS solution was available for updating the inertial filter.	seconds	1

Table 10: INSPVASDCMP Log Field Formats and Scale Factors

Appendix C: Full Configuration Examples

The following examples show complete configuration workflows to output various message types over the CAN bus. The user must be connected to the receiver over a different port than the one they are trying to configure in order to send these commands. For example, if configuring the CCOM1 port, the user should send these commands over a different receiver port such as COM1, USB1, or ICOM1.

Example 1: Configuring the CAN bus to output NMEA2000 messages

The following example assumes that the user wants to send NMEA2000 format standard PGNs and NovAtel proprietary fixed-format messages (using default PGNs) over the receiver's CCOM1 CAN bus port at 500000 bps.

```
CANCONFIG CAN1 OFF
J1939CONFIG NODE1 CAN1
LOG J1939STATUS ONCHANGED
CCOMCONFIG CCOM1 NODE1 NMEA2000
INTERFACEMODE CCOM1 NOVATEL NOVATEL OFF
CANCONFIG CAN1 ON 500K
LOG CCOM1 PGN129025 ONTIME 1
LOG CCOM1 PGN129026 ONTIME 1
LOG CCOM1 PGN129029 ONTIME 1
LOG CCOM1 INSPVACMP ONTIME 1
LOG CCOM1 INSPVASDCMP ONTIME 1
SAVECONFIG
```

Example 2: Configuring the CAN bus to output NovAtel messages

The following example assumes that the user wants to send NovAtel format binary messages over the receiver's CCOM1 CAN bus port at 500000 bps.

```
CANCONFIG CAN1 OFF
J1939CONFIG NODE1 CAN1
LOG J1939STATUS ONCHANGED
CCOMCONFIG CCOM1 NODE1 J1939 1536 7 0x11
INTERFACEMODE CCOM1 NOVATELBINARY NOVATELBINARY OFF
CANCONFIG CAN1 ON 500K
LOG CCOM1 BESTPOSB ONTIME 1
LOG CCOM1 BESTVELB ONTIME 1
SAVECONFIG
```