WOODWARD

Application Note 51321
(Revision A)
Original Instructions

SAE J1939 CAN Communications Used in Woodward GAP™ Controls

General Precautions	**Read this entire manual and all other publications pertaining to the work to be performed before installing, operating, or servicing this equipment.** **Practice all plant and safety instructions and precautions.** **Failure to follow instructions can cause personal injury and/or property damage.**

Revisions	**This publication may have been revised or updated since this copy was produced. To verify that you have the latest revision, check manual *26311* , *Revision Status & Distribution Restrictions of Woodward Technical Publications*, on the *publications page* of the Woodward website:** **www.woodward.com/publications** **The latest version of most publications is available on the *publications page*. If your publication is not there, please contact your customer service representative to get the latest copy.**

Proper Use	**Any unauthorized modifications to or use of this equipment outside its specified mechanical, electrical, or other operating limits may cause personal injury and/or property damage, including damage to the equipment. Any such unauthorized modifications: (i) constitute "misuse" and/or "negligence" within the meaning of the product warranty thereby excluding warranty coverage for any resulting damage, and (ii) invalidate product certifications or listings.**

Translated Publications	**If the cover of this publication states "Translation of the Original Instructions" please note:** **The original source of this publication may have been updated since this translation was made. Be sure to check manual *26311* , *Revision Status & Distribution Restrictions of Woodward Technical Publications*, to verify whether this translation is up to date. Out-of-date translations are marked with ⚠. Always compare with the original for technical specifications and for proper and safe installation and operation procedures.**

Revisions—Changes in this publication since the last revision are indicated by a black line alongside the text.

Contents

WARNINGS AND NOTICES III

ELECTROSTATIC DISCHARGE AWARENESS IV

CHAPTER 1. J1939 INTRODUCTION 1
What is SAE? 1
Overview 1
Physical Layer 2
Message Format Usage 5

CHAPTER 2. J1939 PROTOCOL SUPPORT 8
Address Claiming and the NAME 8
Communication Methods 10
Message Transmission Triggers 11
Message Length 11
Diagnostic Messages 12

CHAPTER 3. GAP™ BLOCK ARCHITECTURE 13
CAN_PORT 13
J1939_NTW 14
CAN_WRITE 16
CAN_READ 18
DM_MANGR 19
DM1_MSG 19
DM2_MSG 19
DM3_MSG 20
DM11_MSG 20
DM5_MSG 20
DM6_MSG 20
DM7_30_MSG 20
DM12_MSG 21
DM19_MSG 21
DM24_MSG 21
DM25_MSG 21
DM32_MSG 21
J1939_BAM 21
J1939_CURV 23
CAN_MSG 24
WRITE_B 25
WR_BYTE 25
WRITE_I 25
WRITE_A 26
READ_B 26
READ_BYTE 26
READ_I 26
READ_A 27

CHAPTER 4. APPLICATION EXAMPLES 28
Interpreting the J1939 standard 28
Transmit PGN61460 GAP Sample 30
Receive PGN61460 GAP Sample 32
RQST Message 34

Contents

APPENDIX. DEFINITIONS AND ABBREVIATIONS 35
Definitions 35
Abbreviations 39

Illustrations and Tables

Figure 1-1. Wiring Example with Isolation for Long Networks...............................3
Figure 1-2. Wiring Example without Isolation on Some Modes.............................4
Figure 1-3. The J1939 29-bit Identifier 5
Figure 1-4. Priority in 29-bit Header 5
Figure 1-5. PGN in 29-bit Header..6
Figure 1-6. Source Address in 29-bit Header..7
Figure 3-1. GAP J1939 Architecture..13
Figure 4-1. CAN_WRITE RQST Method 34
Figure 4-2. CAN_MSG RQST Method 34

Table 1-1. CAN Cable Specification 4
Table 1-2. Priority Rule of Thumb...6
Table 2-1. NAME Fields 9
Table 4-1. Valid Ranges 28

Warnings and Notices

Important Definitions

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

- **DANGER**—Indicates a hazardous situation which, if not avoided, will result in death or serious injury.
- **WARNING**—Indicates a hazardous situation which, if not avoided, could result in death or serious injury.
- **CAUTION**—Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
- **NOTICE**—Indicates a hazard that could result in property damage only (including damage to the control).
- **IMPORTANT**—Designates an operating tip or maintenance suggestion.

WARNING

Overspeed / Overtemperature / Overpressure

The engine, turbine, or other type of prime mover should be equipped with an overspeed shutdown device to protect against runaway or damage to the prime mover with possible personal injury, loss of life, or property damage.

The overspeed shutdown device must be totally independent of the prime mover control system. An overtemperature or overpressure shutdown device may also be needed for safety, as appropriate.

WARNING

Personal Protective Equipment

The products described in this publication may present risks that could lead to personal injury, loss of life, or property damage. Always wear the appropriate personal protective equipment (PPE) for the job at hand. Equipment that should be considered includes but is not limited to:

- **Eye Protection**
- **Hearing Protection**
- **Hard Hat**
- **Gloves**
- **Safety Boots**
- **Respirator**

Always read the proper Material Safety Data Sheet (MSDS) for any working fluid(s) and comply with recommended safety equipment.

WARNING

Start-up

Be prepared to make an emergency shutdown when starting the engine, turbine, or other type of prime mover, to protect against runaway or overspeed with possible personal injury, loss of life, or property damage.

WARNING

Automotive Applications

On- and off-highway Mobile Applications: Unless Woodward's control functions as the supervisory control, customer should install a system totally independent of the prime mover control system that monitors for supervisory control of engine (and takes appropriate action if supervisory control is lost) to protect against loss of engine control with possible personal injury, loss of life, or property damage.

NOTICE **Battery Charging Device**	**To prevent damage to a control system that uses an alternator or battery-charging device, make sure the charging device is turned off before disconnecting the battery from the system.**

Electrostatic Discharge Awareness

NOTICE **Electrostatic Precautions**	**Electronic controls contain static-sensitive parts. Observe the following precautions to prevent damage to these parts:** • **Discharge body static before handling the control (with power to the control turned off, contact a grounded surface and maintain contact while handling the control).** • **Avoid all plastic, vinyl, and Styrofoam (except antistatic versions) around printed circuit boards.** • **Do not touch the components or conductors on a printed circuit board with your hands or with conductive devices.** **To prevent damage to electronic components caused by improper handling, read and observe the precautions in Woodward manual *82715*, *Guide for Handling and Protection of Electronic Controls, Printed Circuit Boards, and Modules*.**

Follow these precautions when working with or near the control.

1. Avoid the build-up of static electricity on your body by not wearing clothing made of synthetic materials. Wear cotton or cotton-blend materials as much as possible because these do not store static electric charges as much as synthetics.
2. Do not remove the printed circuit board (PCB) from the control cabinet unless absolutely necessary. If you must remove the PCB from the control cabinet, follow these precautions:
   - Do not touch any part of the PCB except the edges.
   - Do not touch the electrical conductors, the connectors, or the components with conductive devices or with your hands.
   - When replacing a PCB, keep the new PCB in the plastic antistatic protective bag it comes in until you are ready to install it. Immediately after removing the old PCB from the control cabinet, place it in the antistatic protective bag.

Chapter 1. J1939 Introduction

What is SAE?

The Society of Automotive Engineers (SAE) is a non-profit educational and scientific organization dedicated to advancing mobility technology to better serve humanity. Over 90,000 engineers and scientists, who are SAE members, develop technical information on all forms of self-propelled vehicles including automobiles, trucks and buses, off-highway equipment, aircraft, aerospace vehicles, marine, rail, and transit systems. SAE disseminates this information through its meetings, books, technical papers, magazines, standards, reports, professional development programs, and electronic databases. SAE authors the J1939 standard.

Overview

The J1939 series of SAE Recommended Practices has been developed by the Truck & Bus Control and Communications Network Subcommittee of the Truck & Bus Electrical & Electronics Committee. The purpose of these Recommended Practices is to provide an open interconnect system for electronic systems. It is the intention of these Recommended Practices to allow Electronic Control Units (ECU) to communicate with each other by providing a standard architecture.

The usage of these Recommended Practices is not limited to truck and bus applications; other applications may be accommodated with immediate support being provided for construction and agricultural equipment and stationary power systems. J1939 is a high-speed communications network designed to support real-time closed loop control functions between devices that may be physically distributed throughout the system.

There is no procedure presently in place to test, validate, or provide formal approval for controls utilizing the J1939 network. Each developer is expected to design their products to the spirit of, as well as the specific content of, these recommended practices. The Woodward GAP implementation follows the SAE specification for the features supported. The remaining implementation is up to the GAP application engineer. The application engineer is expected to follow the specification and create a final network implementation that is compliant to the SAE J1939 standard. This Application Note will assist the GAP application engineer in that process.

The available J1939 standards and their purpose are as follows:

J1939	Base standard which describes the network in general and provides all preassigned NAME components
J1939-1	On-Highway application examples
J1939-2	Agriculture and Forestry applications. A primary intent of this document is to point out areas where there may be differences between SAE J1939 and ISO 11783.
J1939-11	Physical layer (250 kbit/s, twisted shielded pair) cabling
J1939-13	Standard diagnostic connector for a vehicle system
J1939-15	Reduced physical layer (250 kbit/s, unshielded twisted pair). Not for use in industrial applications.

J1939-21	Data link layer. Describes the use of the CAN 29-bit identifier and the commonly used messages such as Request, Acknowledgement, and Transport Protocol. The Transport Protocol specifies the breaking up of large amounts of data into multiple CAN-sized frames, along with adequate communication and timing to support effective frame transmission between nodes.
J1939-31	Network layer. Defines the requirements and services needed for communication between devices in different segments. Describes the various types of network interconnections allowed: bridges, routers, repeaters, gateways.
J1939-71	Vehicle application layer. Provides the definition of most data elements and messages. In the application profile all parameters as well as assembled messages called parameter groups are specified. Each CAN message is referenced by a unique number, the PGN.
J1939-73	Defines application layer diagnostics such as DM1, DM2, etc.
J1939-74	Defines configurable messaging. This is an application layer that allows users to use parameters that have previously been identified as configurable to be packed into messages in a proprietary manner for optimizing network performance. GAP/Coder does not support these messages.
J1939-75	Defines generator sets and industrial application layer messages. This particular document describes the parameters and parameter groups that are predominantly associated with monitoring and control of generators and driven equipment in electric power generation and industrial applications – but not with the engine or turbine driving the equipment.
J1939-81	Network management. Describes the management of source addresses for ECUs, the association of addresses with functionality, and minimum requirements for networked units.
J1939-82	Compliance test procedures for self-assessment of standards compliance.
J1939-84	On Board Diagnostic communications compliance test cases. These tests are for system level compliance with on road emissions standards.

Physical Layer

Hardware

The physical medium for J1939 devices is a differentially driven two-wire communications bus with common return according to ISO 11898-2 (and SAE J1939-11) commonly called CAN (Controller Area Network). CAN_L, CAN_H, and Shield (or drain wire) are the only mandatory lines. Some applications use the module power supply return (B–) in place of a separate or isolated CAN_GND.

Galvanic isolation between nodes is optional depending on the device it is used in and the distances required.

A termination impedance of 121 Ω ±2% is required. Only the two bus ends should be terminated.

Bit Rate

J1939 only supports 250 kbps. This baud rate is fixed in GAP/Coder and cannot be changed.

Bus Length

J1939 recommends a maximum bus length of 40 meters without repeaters. This limitation is due to the expected application on mobile equipment without isolation. When used in stationary applications with galvanic isolation, it is possible to reach distances as long as 250 m if all modules on the bus are properly isolated. Cable drops (stubs) from the main bus trunk to a node are limited to 1 m in either case.

SAE also recommends that no more than 30 devices be used on the same physical bus. This limitation is imposed to correlate with the CAN cable characteristics and transceiver requirements defined for use with J1939 networks. The GAP implementation does not have any limitations on the number of devices since this is a system level issue.

The shield drain wire must be grounded at one location only. This location is preferred to be near the center of the network. Each node may have a shield connection but that connection is normally expected to be a capacitive or capacitive and resistive connection to ground (also called an AC connection to ground). See Figure 1-1 for an example.

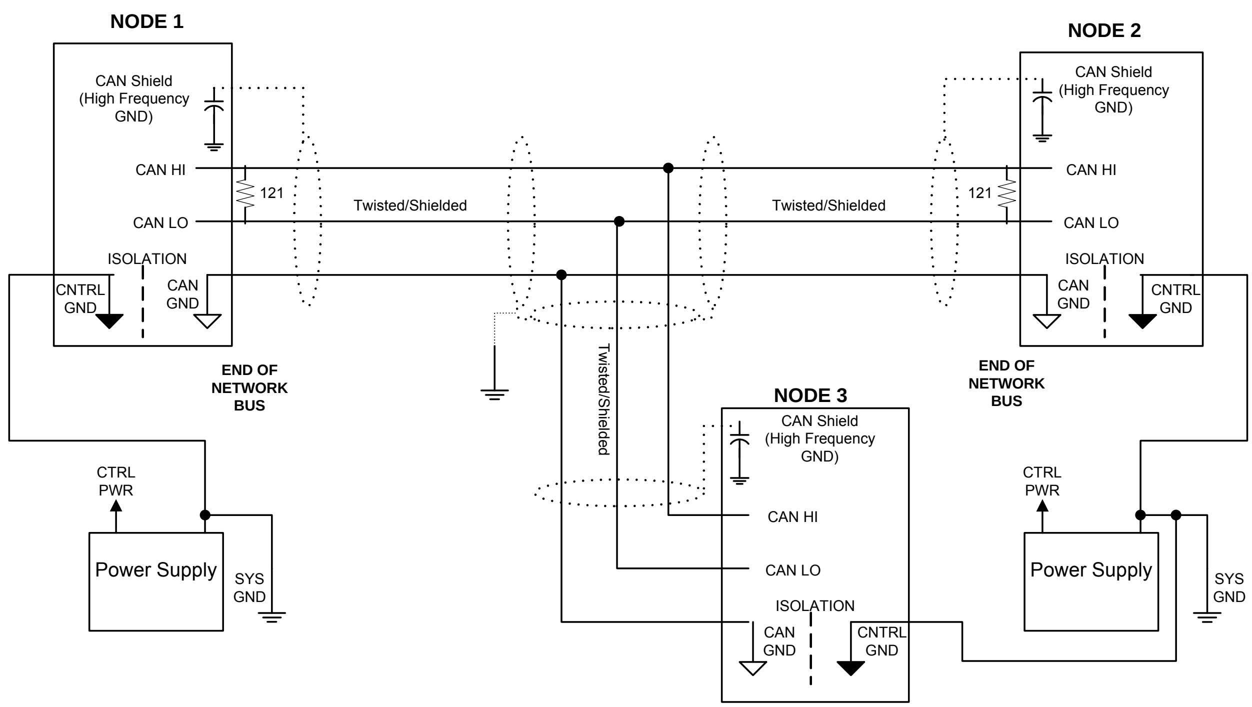


Figure 1-1. Wiring Example with Isolation for Long Networks

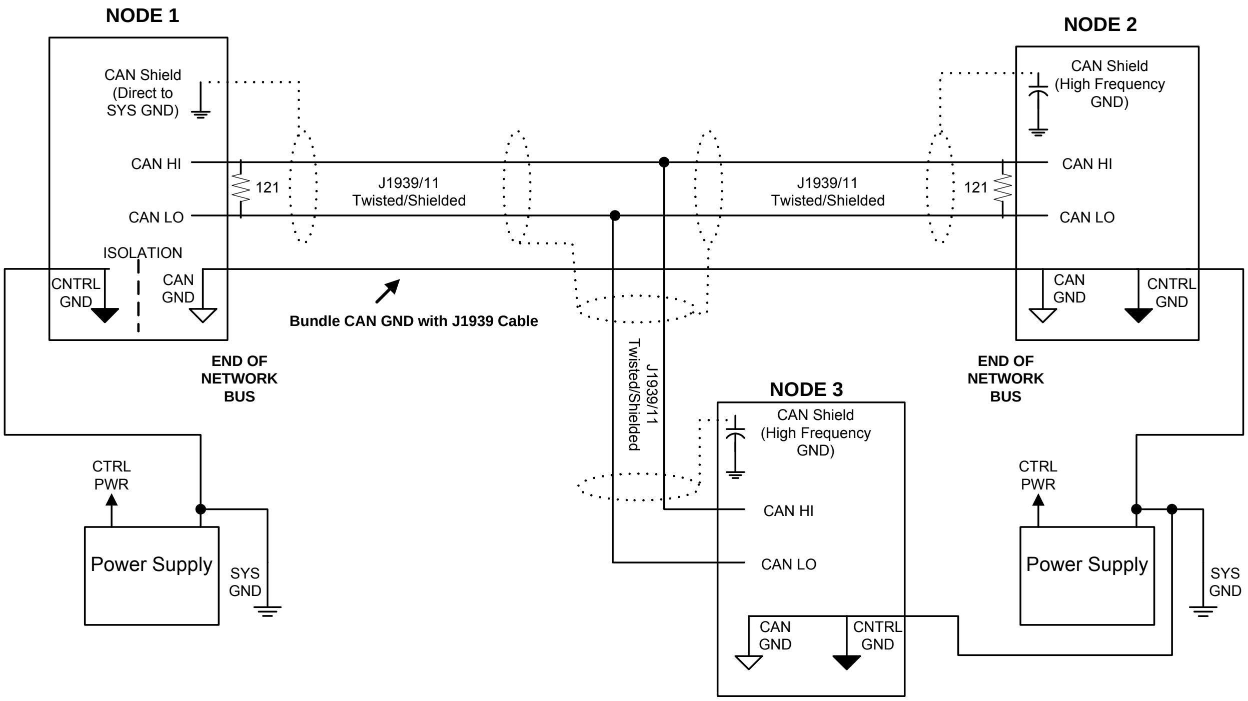


Figure 1-2. Wiring Example without Isolation on Some Modes

Cable Specification

SAE J1939 cable is defined to have two conductors and a shield drain wire (uninsulated wire). The two conductors are green for CAN_L and yellow for CAN_H. CAN cable is also manufactured for use in other fieldbus applications like DeviceNet™ * and CANopen. Those cables may be used in J1939 applications as long as they meet the environmental requirements of the application environment (like temperature rating).

*—DeviceNet is a trademark of ODVA (Open DeviceNet Vendor Association, Inc.).

A commonly used J1939 CAN cable is Raychem (Tyco Electronics) Cheminax 2019D0301. This part has two conductors + shield/drain wire. All wires are 0.75 mm².

The following table provides the cable requirements from the SAE J1939-11 standard.

Data pair impedance	120 Ohm ±10% at 1 MHz
Cable capacitance	40 pF/m at 1 kHz (nominal)
Propagation delay	5 ns/m (maximum) or >67% propagation velocity
DC Resistance	25 mΩ/m @ 20 °C (nominal)
Data Pair	0.8 mm² (approx. 18 AWG), individually tinned, 10 twists/meter OR 0.5 mm² (approx. 20 AWG), individually tinned, 10 twists/meter
Drain / Shield Wire	0.8 mm² (approx. 18 AWG), individually tinned OR 0.5 mm² (approx. 20 AWG), individually tinned
Shield Effectiveness	200 mΩ/m surface transfer impedance up to 1 MHz
Bend radius	4x cable diameter

Table 1-1. CAN Cable Specification

Message Format Usage

J1939 requires use of CAN2.0B revision transceivers. A CAN2.0A transceiver is not capable of supporting the 29-bit identifier (CAN Extended Frame). All Woodward products supporting the J1939 protocol use the CAN2.0B transceivers.

J1939 provides a complete network definition using the 29-bit identifier (CAN Extended Frame) defined within the CAN protocol shown in Figure 1-3. The CAN Data Frame Bits SOF, SRR, IDE, and RTR bits will not be discussed in the following description (see J1939-21 and ISO 11898 for details of these bits).

SOF	ID (11-bits)				SRR	IDE	EXT ID (18-bits)			RTR
		PGN					PGN			
	Priority	EDP	DP	PF			PF	PS	SA	
	3 bits	1 bit	1 bit	6 bits			2 bits	8 bits	8 bits	

Figure 1-3. The J1939 29-bit Identifier

The SRR and IDE bits are entirely defined and controlled by CAN and therefore not described or modified by J1939. They are not considered part of the PGN field.

Priority

The first three bits of the 29-bit identifier are used for determining message priority during the arbitration process. A value of 0 (bitwise 000) has the highest priority. Higher priority messages are typically used for high-speed control messages. An example of this is the torque control message from the transmission to the engine (see J1939-71). A lower priority is used for data that is not time critical. An example of this is the engine configuration message. The priority field may be configurable for each message type so that network tuning can be performed by an OEM, if necessary.

11-bit ID area				Extension for 29-bit ID	
Priority	PGN Number (MSG_NUM)				Node ID
Priority	EDP	DP	PDU Format (PF)	PDU Specific (PS)	Source Address (SA)

Figure 1-4. Priority in 29-bit Header

The table below can be used as a rule of thumb for setting priority levels. For example, use a priority of 2 rather than a priority of 1 to demote a message with respect to messages at priority level 1.

Priority (decimal)	Priority (bitwise)	Note
7	111	Configuration messages and ON_REQUEST messages (Very low priority)
6	110	Most status and diagnostic messages (Low priority)
5 or 4	101 100	High speed status messages (Medium priority)
3 or 2	011 010	Most control messages (High priority)
1	001	Control messages related to the closed loop control of combustion related features (Very high priority)
0	000	Reserved for extremely critical functions (Highest priority)

Table 1-2. Priority Rule of Thumb

Parameter Group Number (PGN)

The Parameter Group Number (PGN) consists collectively of the Extended Data page (EDP) bit, Data Page (DP) bit, PDU Format (PF) field, and the PDU Specific (PS) field. PDU stands for Protocol Data Unit (i.e. Message format).

All Parameter Groups (PG) have definitions which include the parameter (data) assignments within the 8-byte data field of each message as well as the transmission repetition rate and priority. The term "Parameter Group" is used because they are groups of specific parameters. Parameter Groups are identified by a Parameter Group Number (PGN), which uniquely identifies each Parameter Group. The PGN structure permits a total of up to 8672 different Parameter Groups to be defined per data page (DP). Parameter Groups and Parameter Group Numbers are described in J1939-71.

<table>
<tr><td colspan="4">11-bit ID area</td><td colspan="3">Extension for 29-bit ID</td></tr>
<tr><td>Priority</td><td colspan="5">PGN Number (MSG_NUM)</td><td>Node ID</td></tr>
<tr><td>Priority</td><td>EDP</td><td>DP</td><td colspan="2">PDU Format (PF)</td><td>PDU Specific (PS)</td><td>Source Address (SA)</td></tr>
</table>

Figure 1-5. PGN in 29-bit Header

The EDP and DP bits are used as page selectors. Page 0 contains all the messages which are presently defined in the SAE standard. Page 1 provides additional expansion capacity for the future, to be assigned after page 0 has been completed. Page 2 contains messages for other protocols which may coexist on the CAN bus simultaneously. Page 3 provides additional expansion capability for the future.

The PF field identifies one of two PDU formats able to be transmitted. PDU formats are described in J1939-21, Section 3.3. If the PF value is between 0 and 239 then it is considered a PDU1 type. A PDU1 message is destination specific meaning that it will be sent to one and only one destination node. If the PF field is between 240 and 255 then it is considered a PDU2 type. A PDU2 message is a broadcast message meaning that any node on the network may listen to the message.

The next 8 bits of the identifier are PDU Specific (PS), meaning that they are dependent on the value of the PF. If the PF value is between 0 and 239 (PDU1 type), the PS field contains a destination address. If the PF field is between 240 and 255 (PDU2 type), the PS field contains a Group Extension (GE) to the PDU Format. The Group Extension provides a larger set of values to identify messages that can be broadcast to all nodes on the network.

Most messages on J1939 are intended to be broadcast using the PDU2 format. Data transmitted on the network using PDU2 format cannot be directed to a specific destination. When a message must be directed to a particular node, it must be assigned a PGN in the PDU1 format range of numbers so a specific destination address can be included within the identifier of the message. An example of this is the power management device commanding a specific speed from the engine control.

Source Address (SA)

The last 8 bits of the identifier contain the address (Node ID) of the device transmitting the message (Source Address). For a given network, every address must be unique (254 available). Two different devices cannot use the same address at the same time. All PGNs are independent of the Source Address, thus any node can transmit any PGN.

<table>
<tr><td colspan="4">11-bit ID area</td><td colspan="2">Extension for 29-bit ID</td></tr>
<tr><td>Priority</td><td colspan="4">PGN Number (MSG_NUM)</td><td>Node ID</td></tr>
<tr><td>Priority</td><td>EDP</td><td>DP</td><td>PDU Format (PF)</td><td>PDU Specific (PS)</td><td>Source Address (SA)</td></tr>
</table>

Figure 1-6. Source Address in 29-bit Header

Chapter 2.
J1939 Protocol Support

This chapter explains the level of support provided by the Woodward GAP J1939 implementation. This support is provided in the GAP blocks used to create a functional J1939 network using a Woodward GAP programmable control with support for a CAN network.

Address Claiming and the NAME

Address claiming is the standard J1939 process used to enforce that each device on the network is using a unique source address (NODE_ID). Each device on the network will have at least one name and one address associated with it. There are examples, such as an engine and engine retarder residing in a common control unit, wherein multiple names and multiple addresses may coexist within a single device. The node address defines a specific communications source or destination for messages. The name includes identification of the primary function performed at that address and adds an indication of the instance of that functionality in case multiple devices with the same primary function coexist on the same network. As many as 254 different devices of the same function may coexist on the network, each identified by its own address and name.

To uniquely name each node, J1939 defines a 64-bit NAME consisting of the fields shown in Table 2-1. The Function Instance, ECU Instance, and Identity Number fields permit multiple devices of the same make and model to coexist on the same network but still have unique NAMEs for each. See J1939-81 for a full description of device naming and address assignment. See the J1939 base level standard Appendix B for current committee assignments.

NAMEs identify the primary vehicle function or functions which a controller performs and uniquely identify each device, even when there are more than one of the same type on the network. But with a length of 64 bits, a NAME is inconvenient to use in normal communications. Therefore, once the network is fully initialized, each device utilizes an 8-bit address as its source identifier or “handle” to provide a way to uniquely access a given device on the network. For example, an engine control may be assigned address 0, but if a second engine control is present, it needs a separate, unique address (e.g. 1) and instance.

Devices that accept destination specific commands may require multiple addresses. This permits distinguishing which action is to occur. For example, if the transmission is commanding a specific torque value from the engine control (address 0), this must be differentiated from commanding a specific torque value from the engine brake (retarder)(address 15). As can be seen by this example, a single device on the network may have multiple addresses and each address will have an associated NAME. However, at this time, Woodward GAP controls do not support the ability to assume multiple addresses.

Field	Range	GAP Configurable	Function
Arbitrary Address Capable	1 bit	No	Automatic/Manual address (node ID) assignment. GAP/Coder is manual only so this field is always 0.
Industry Group	3 bits (0 – 7)	Yes	Typically "Global" (0) or "Process Control" (5) for Woodward applications J1939 Standard Appendix B, Table B1.
Vehicle System	7 bits (0 – 127)	Yes	Provides a common name for a group of functions within a network. For example, "engine" for an engine+ generator application. J1939 Standard Appendix B, Table B12
Vehicle System Instance	4 bits (0 – 15)	Yes	Used to identify a particular occurrence of a particular Vehicle System
Function	8 bit (0 – 255)	Yes	When combined with the Industry Group and Vehicle System fields, this identifies a common name for a specific controller. For example, "ignition control". J1939 Standard Appendix B, Table B11
Function Instance	5 bits (0 – 31)	Yes	Used to identify a particular occurrence of a particular Function
ECU Instance	3 bits (0 – 7)	Yes	Used to identify a particular occurrence of a control function
Manufacturer Code	11 bits	Yes	Used to identify the control manufacturer. Woodward has been assigned 153 but GAP/Coder allows this field to be configured for customer specific needs. J1939 Standard Appendix B, Table B10
Identity Number	21 bits	No	A unique number to aid arbitration when two or more controls with the same default application are put on a network prior to proper configuration. This number is unique per controller and per CAN port on each controller. It is not adjustable via GAP/Coder.

Table 2-1. NAME Fields

To facilitate the initialization process of determining the address(es) for each device on the network, commonly used devices have Preferred Addresses assigned by the committee (Preferred Addresses are listed in Tables B2 - B9 of the J1939 base standard). Using the Preferred Address minimizes the frequency of multiple devices attempting to claim the same address. The application engineer must choose the appropriate Preferred Source Address for the application and use it in the NODE_ID field of the J1939_NTW block. In general, most devices (including the Woodward control) will use their Preferred Addresses immediately upon power up. A specific procedure (defined in J1939-81) for assigning addresses after power-up is used to resolve any conflicts that may occur. Each device must be capable of announcing which address(es) it intends to use. This is the address claim feature.

When an address conflict has been detected, the following four options are available, depending upon the capabilities of the device involved:

- Self-configurable controls—A self-configurable control is capable of dynamically computing and claiming an unused address. Most service tools and bridges will have this capability.
- Command configurable controls—A network interconnection device, such as a bridge or a service tool, may command another control to use a given address. The control having the unclaimable address would then issue an Address Claimed message to acknowledge acceptance of this new commanded address. The control may be commanded to accept a new address even though it has already claimed a valid address.
- Service configurable controls—Controls that are modifiable by service personnel, usually by the means of DIP switches or a proprietary service tool. When "commanded address" messages are used, this option differs from the Command Configurable in that a service tool is required and will often use proprietary techniques. GAP controls conform to this type but do not support the commanded address message. If the application engineer made the NODE_ID field tunable, the Source Address may be configured using a Woodward service tool.
- Non-configurable controls—These controls have an address that cannot be changed except by downloading a new program.

Communication Methods

Three primary communication methods exist within J1939 and appropriate use of each type allows effective use of the available Parameter Group Numbers. The three communications methods are:

- Destination specific communications, using PDU1 (PF values 0–239) (includes the use of the global destination address—255)
- Broadcast Communications using PDU2 (PF values 240–255)
- Proprietary Communications using either PDU1 or PDU2 format

Each of the communications methods has an appropriate use. Destination specific Parameter Group Numbers are needed where the message must be directed to one or another specific destination and not to both.

Broadcast Communications apply in several situations, including:

- Messages sent from a single or multiple sources to a single destination
- Messages sent from a single or multiple sources to multiple destinations

Broadcast Communications cannot be used where a message must be sent to one or another destination and not to both.

The third communications method in J1939, proprietary communications, is provided by the use of two proprietary Parameter Group Numbers. A Parameter Group Number has been assigned for broadcast proprietary communications and a Parameter Group Number has been assigned for destination specific proprietary communications. This allows for two functions. One, a specific source can send its proprietary message in a PDU2 type format (broadcast). Two, it allows for situations where a service tool must direct its communication to a specific destination out of a possible group of controls. For instance this case arises when an engine uses more than one controller but the service tool must be able to perform calibration/reprogramming while all devices are connected to the same network. In this case the proprietary protocol needs to be destination specific. Note that the destination device must be capable of properly interpreting the proprietary data.

Proprietary communications are useful in two situations:

- Where it is unnecessary to have standardized communications
- Where it is important to communicate proprietary information

Woodward GAP/Coder controls fully support all three PGN types for transmission and reception of 8-byte (or less) messages. Any PGN number in the valid range defined in J1939-71 or in the proprietary ranges may be used.

Message Transmission Triggers

J1939-71 lists an appropriate transmission frequency for each PGN defined in the standard. This message frequency is provided to communicate what the SAE committee anticipated when the message was defined. It is not a requirement to use this rate but if a slower rate is used, other devices on the link may incur timeouts in error.

Most PGNs are defined with a cyclic transmission frequency. This transmission type is supported in GAP in two ways. Both are described later in this application note.

Some PGNs are defined in J1939-71 as “On Request”. This message transmission type is used for identification parameters and seldom needed data. GAP supports creating messages to be sent upon request by another device. It is possible to create an “On Request” message for data within another device but it must be done by implementing the RQST PGN defined in J1939-21 within the GAP application environment. An example of this is provided in the Application Examples chapter of this application note.

J1939 does not define a Change of State method for PGN transmission trigger. However, the message trigger portions of the GAP CAN implementation are common for all protocols supported within GAP and CANopen does need this trigger method. Therefore, Woodward does provide a Change of State method for transmission trigger.

Message Length

CAN messages may only contain up to 8-bytes of data. This size constraint is common to all CAN based applications regardless of protocol. The majority of messages defined in J1939-71 are 8-bytes long. However, a few are longer and require a special protocol in order to transmit the larger data size. Support for larger data size is provided by a special protocol within J1939 that uses multiple messages to transmit the data and then reassemble the data at the receiving end. Woodward GAP controls support the ability to SEND messages with data longer than 8-bytes using the special J1939 transport protocol. The GAP controls cannot receive messages of this type though.

Diagnostic messages like DM1 also use BAM to transmit their data. Since it is undesirable to affect the frequency of the DM1 transmissions, the GAP control will not allow an application defined BAM message that would take more than 1 second to transmit when BAM-based DM1 messages are also needed. If such a long message is needed, DM1 messages must be disabled in order to transmit the large data packet.

Diagnostic Messages

SAE has defined many diagnostic messages in J1939-73. The diagnostic messages are primarily defined for compatibility with third party diagnostic tools. The Woodward GAP controls support DM1 (active diagnostics), DM2 (logged diagnostics), DM3 (reset active/logged diagnostics), DM5 (diagnostic readiness), DM6 (pending diagnostics), DM7 (command for non-continuously monitored tests), DM11 (reset active emissions related diagnostics), DM12 (active emissions related diagnostics), DM19 (calibration information), DM24 (SPN support), DM25 (expanded freeze frame), DM30 (scaled test results), DM32 (regulated exhaust emission level exceedance). Further support in the future is possible. All the DM messages require the EVENT_MGR block. Only those events defined within the EVENT_MGR block will be used in the diagnostic messages. The application engineer can use any Suspect Parameter Number (SPN) and Failure Mode Identifier (FMI) combination. SAE has also defined a range of SPN for proprietary use in case a control has diagnostic features that are not already outlined in J1939. Woodward supports the entire range of SPN numbers.

Chapter 3.
GAP™ Block Architecture

The GAP implementation for CAN is very flexible in order to accommodate many diverse needs. The arrangement of blocks is shown in Figure 3-1. Each block will be described in detail. This block suite allows any PGN to be sent and any 8-byte PGN (from J1939-71 or proprietary) to be received.

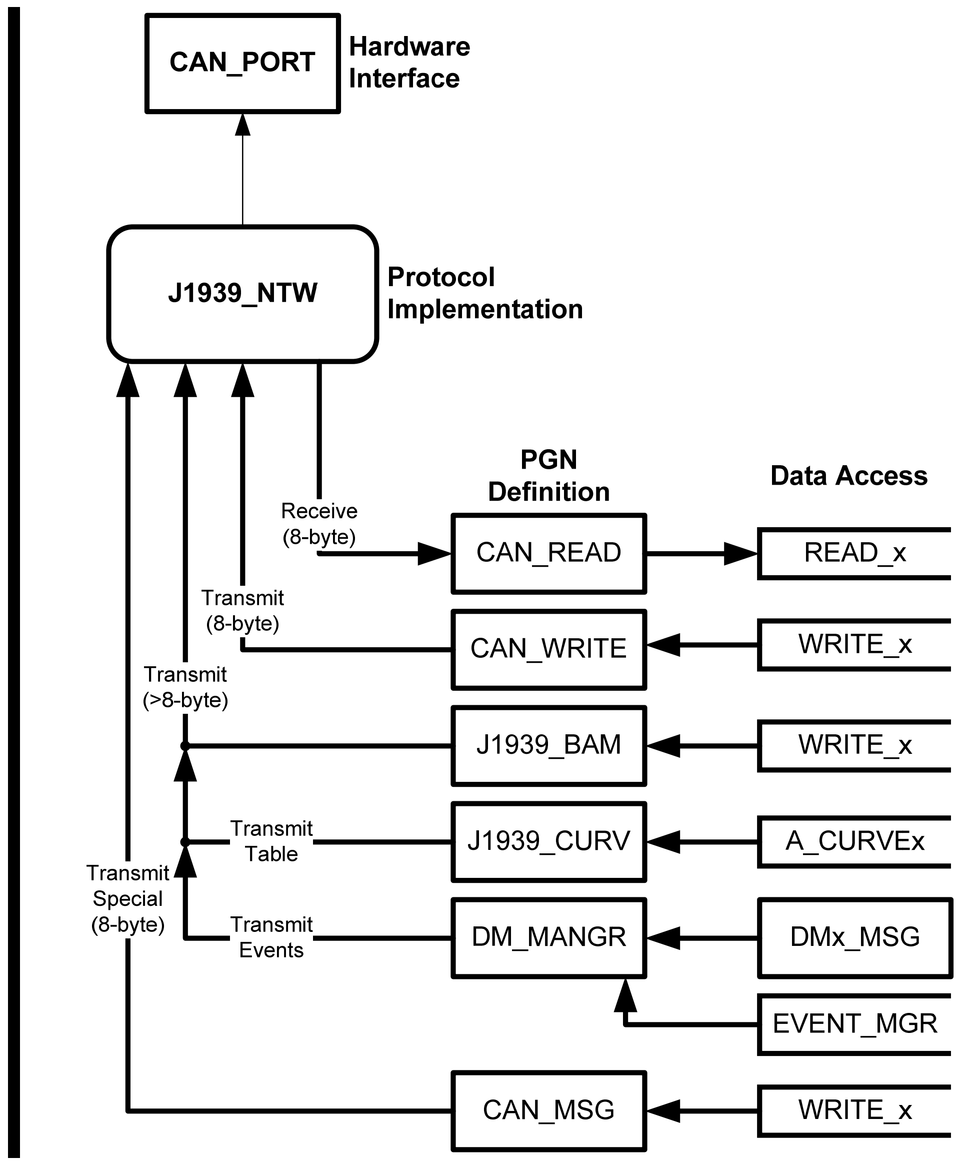


Figure 3-1. GAP J1939 Architecture

CAN_PORT

This block identifies the interface between software and hardware. Most Woodward GAP/Coder controls provide 2 or 3 CAN ports. The CAN_PORT block must be connected to the chassis (or module) block for the control in use. The chassis (or module) connection will define which of the physical ports is used for the network. A network protocol block and a CAN port block have a 1:1 relationship.

Although the CAN_PORT block has a rate group assignment, this assignment is only used to provide the CAN_P_STAT diagnostic functionality. The rate group selection will not affect the network transmit and receive behavior.

J1939_NTW

This block implements the J1939 protocol and provides network status feedback to the application. Each PGN to be sent or received goes through this protocol block so the CAN header can be properly formatted (or identified). Data formatting is done by the interface blocks that are children to this parent block.

Although the J1939_NTW block has a rate group assignment, this assignment is only used to provide the diagnostic functionality at the outputs of the block. The rate group selection will not affect the network transmit and receive behavior (with the exception of J1939_BAM and J1939_CURV messages).

NAME

The NAME described earlier in this application note is implemented in this block. Each configurable field for the NAME is an input to the J1939_NTW block. The values at these inputs upon boot-up (after the rate group runs once) or upon CAN_RST are used for the NAME used during address claiming.

It is up to the application engineer to properly set the values of these fields for the application installed in the control. Rules for assignment are provided in “Address Claiming and the NAME” in Chapter 2 as well as in the J1939 standard.

NODE_ID

The NODE_ID input to this block is used as the Source Address for this CAN port. All PGNs sent from this port will have this NODE_ID automatically assigned as the Source Address in the CAN header of the message. The NODE_ID detected at boot-up (after the rate group runs once) or upon CAN_RST will be used.

RCV_BF_SZ

All messages communicated via the connected CAN port are processed by the network block. This also includes messages that are not intended for use by the control application. All messages on the communication bus during the rate group period for the J1939 network are placed into a software buffer to be processed by the network block when the rate group runs. The rate group used for this processing is the fastest rate group used by any block in the TX_PGN_X and RCV_PGN_X repeat groups. The buffer size must be large enough to hold all the messages that will be communicated during this period (both transmit and receive). Woodward has made this buffer size configurable so that controls with a limited amount of RAM are not required to have a large buffer if only a few messages may be present on the network.

If this size is set too small, the maximum number of messages observed by the control to overflow the buffer during operation will be set on the RXOVRFLOW output of the block. The buffer size should always be set to a value larger than necessary to leave room for service tool communications or other special messages sent by some device on the communication bus.

Troubleshooting Outputs

The J1939_NTW block provides many outputs to facilitate network troubleshooting and for network status indication.

The CAN_LOAD output shows the average load seen in the past few seconds. No network should ever be loaded higher than 80% for reliable communications to be expected. A lower number like 60% is a better target for most networks.

The TX_WARN, TX_ERR, RX_WARN, RX_ERR, BUS_OFF outputs all come directly from the hardware CAN controller responsible for CAN communications on the bus. In general, if a wiring error is present (polarity switched, wiring short or open, termination resistor problem) the block will show one or more of these outputs TRUE.

The RXTXERRS output provides a cumulative count of receive and transmit errors observed by the CAN controller. In order to limit the processing needed for this feature, only 1 error per rate group will be counted although many more errors than that are possible. This output is useful to understand if an intermittent error is occurring like a wire sometimes shorting to the engine block. It is only reset upon reboot or CAN_RST but it will rollover and begin counting again at 0 if the number of errors exceeds 231.

If the control fails the Address Claim process, the DUP_ADDR output will be TRUE. This is an indication that another device on the network with a higher NAME priority is already using the Source Address defined at the NODE_ID input to this block. In this case, the ONLINE output will also be FALSE. The ONLINE output may also be FALSE if the controller is BUS_OFF due to a wiring error.

The RXOVRFLOW output will report the total number of messages that were ignored because the receive buffer was full. If this output ever exceeds 0, consider increasing the RCV_BUF_SZ.

The TXOVRFLOW output will report the total number of messages to be sent that were skipped because the transmit buffer was full. If this output ever exceeds 0, it is likely due to a bus load that is nearing 100%. The bus load must be reduced to eliminate this condition.

PGN Assignments

All PGNs to be received are configured using CAN_READ blocks and listed within the RPTrcv repeat group. All PGNs to be transmitted are configured using CAN_WRITE, J1939_BAM, or J1939_CURV blocks and listed within the RPTtx repeat group. If diagnostic messaging is required, the DM_MANGR block must be connected to the DM_MANGR input of the J1939_NTW block. The specific DMx_MSG blocks to be supported are then connected to the DM_MANGR block.

A special type of block called CAN_MSG may be used in J1939 as well. This block is used to transmit PGNs of 8-bytes or less that must only be sent upon a manual (or application generated) trigger rather than Change of State or Cyclic. It can also be used to send PGNs outside the range defined in J1939-71 if this is necessary. Such behavior should only be used by someone very familiar with J1939 at the base protocol level. All blocks of this type are listed within the RPTmsg repeat group.

CAN_WRITE

The CAN_WRITE block provides a method to define PGNs to be sent from the control. This block supports PGNs of 8-bytes or shorter length. Nearly all PGNs in J1939 are defined as 8-bytes even if just 1 byte is actually used to convey information. This is done to facilitate backward compatibility when bytes within the message are defined later and added to the standard.

Transmission Trigger

The MSG_TYPE field of this block is used to define how and when the PGN will be transmitted.

- ASYNC messages are transmitted as soon as the block runs within the rate group.
- CYCLIC messages are transmitted at the start of the next rate group depending on SYNC_SKIP. The SYNC_SKIP field provides a method to delay message transmission longer than the rate group time. It will provide a cyclic message transmission of SYNC_SKIP * Rate Group time.
- ONREQUEST messages are transmitted only when another device asks for the PGN defined by the MSG_NMBR input.
- Change of State (COS) messages are transmitted only when one or more data values within the message change. Alternatively, a timeout period may be defined using the COS_TMOUT input so the message is sent cyclically at the timeout rate even if the data has not changed but still sent immediately upon any data change. This facilitates using a timeout at the receiving end to verify receipt of valid data. A COS message trigger type is not normally used in J1939 but allowed by Woodward GAP controls.

CAN Header (ID)

The identifier (message header) used for the message transmission is created based on multiple block inputs. See the “Message Format Usage” section in Chapter 1 for details of the message header.

- PRIORITY input is used for the 3-bit priority portion of the header.
- MSG_NMBR input is used for the PGN portion of the header. If the PGN is a PDU1 type then the NODE_ID input to the CAN_WRITE block will be added to the MSG_NMBR to create the PGN number. If the PGN is a PDU2 type then the NODE_ID input is ignored and the PGN number will be set equal to the MSG_NMBR.
- The NODE_ID input on the J1939_NTW block will be used as the Source Address portion of the header for all transmitted messages.

If the MSG_NMBR input indicates a PDU1 type message, the PGN will be calculated based on using the MSG_NMBR and adding the NODE_ID of this CAN_WRITE block. This is done to make it easy to send a destination specific message to a device which may be at a variable node ID. Such an example is the TecJet™ control, which expects a destination specific message for the flow command, but the TecJet control may use 1 of 4 different addresses based on the harness code inputs.

The CAN_ID_EN and CAN_ID inputs to this block are ignored at compile time for J1939 applications. For the purpose of J1939 applications, they should be ignored as if they were not present on the block.

The CAN_ID_OUT output is the actual PGN number used based on the MSG_NUM and NODE_ID inputs. It will only represent the PGN portion of the CAN header. This output is provided since it is possible to have a MSG_NMBR come from a tunable or Name block. The MSG_NMBR and NODE_ID inputs are only used by the control at network initialization time, which occurs following a reboot or following CAN_RST on the J1939_NTW block. So it is possible that the input may be tuned to a new value for the PGN yet the network is still using the prior value because it has not been re-initialized. Hence, this output can be a helpful troubleshooting tool.

Endian

The J1939 standard requires the use of Little Endian (Intel format). This field must always be set False to use Little Endian for J1939.

Message Data

The data used in the message contents comes from the DATA_X repeat group inputs. The data packing in the message will happen in the order it appears in the repeat group. However, for data longer than 1-byte, the byte ordering of the individual data item will be according to the Little Endian rules.

Example 1:
CAN_WRITE.DATA_1 = WRITE_I (8-bits) = 1
CAN_WRITE.DATA_2 = WRITE_I (8-bits) = 2
CAN_WRITE.DATA_3 = WRITE_I (8-bits) = 3
CAN_WRITE.DATA_4 = WRITE_I (8-bits) = 4
CAN_WRITE.DATA_5 = WRITE_I (8-bits) = 5
CAN_WRITE.DATA_6 = WRITE_I (8-bits) = 6
CAN_WRITE.DATA_7 = WRITE_I (8-bits) = 7
CAN_WRITE.DATA_8 = WRITE_I (8-bits) = 8
CAN_WRITE.MSG_NMBR = 65144
CAN_WRITE.PRIORITY = 6
J1939_NTW.NODE_ID = 1

Header	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Footer
419330049	1	2	3	4	5	6	7	8	CRC

Example 2:
CAN_WRITE.DATA_1 = WRITE_B (2-bits) = True, Error=False, Not_Supp=False
CAN_WRITE.DATA_2 = WRITE_B (2-bits) = True, Error=False, Not_Supp=False
CAN_WRITE.DATA_3 = WRITE_B (2-bits) = True, Error=False, Not_Supp=False
CAN_WRITE.DATA_4 = WRITE_B (2-bits) = True, Error=False, Not_Supp=False
CAN_WRITE.DATA_5 = WRITE_I (8-bits) = 210 (D2 hex)
CAN_WRITE.DATA_6 = WRITE_I (16-bits) = 1234 (04D2 hex)
CAN_WRITE.DATA_7 = WRITE_A (32-bits) = 1234.0 (449A4000 hex)
CAN_WRITE.MSG_NMBR = 65144
CAN_WRITE.PRIORITY = 6
J1939_NTW.NODE_ID = 1

Header	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Footer
419330049	85	210	210	4	0	64	154	68	CRC
0x18FE7801	0x55	0xD2	0xD2	0x04	0x00	0x40	0x9A	0x44	CRC

CAN_READ

The CAN_READ block provides a method to define PGNs to be received by the control. This block supports PGNs of 8-bytes or shorter length. Nearly all PGNs in J1939 are defined as 8-bytes even if just 1 byte is actually used to convey information. This is done to facilitate backward compatibility when bytes within the message are defined later and added to the standard.

IMPORTANT **The MSG_TYPE, CAN_ID_EN, CAN_ID fields are all ignored and not used for J1939.**

CAN Header (ID)

The identifier (message header) used for the message acceptance filter is determined based on two block inputs. See the "Message Format Usage" section in Chapter 1 for details of the message header.

- The message priority is only used to arbitrate getting the message onto the bus. It is never used as a filter in J1939 for message receipt. These bits are masked off and ignored by the message acceptance filters.
- MSG_NMBR input is used for the PGN portion of the header. If the MSG_NMBR input indicates a PDU1 type message, the PGN number to use in the MSG_NMBR field should not include the Destination Address (should be the EDP+DP+PF portion). If the MSG_NMBR input indicates a PDU2 type message, the PGN number to use in the MSG_NMBR field must match the PGN portion of the CAN header entirely.
- The NODE_ID input on the CAN_READ block will be used to further filter messages so that only messages coming from a Source Address equal to the value of the NODE_ID field will be accepted. If it is desired to accept a given PGN from any node, then use a NODE_ID=255 to disable the filter feature.

The CAN_ID_EN and CAN_ID inputs to this block are ignored at compile time for J1939 applications. For the purpose of J1939 applications, they should be ignored as if they were not present on the block.

The CAN_ID_OUT output is the actual PGN number used based on the MSG_NUM input at the time of initialization. It will only represent the PGN portion of the CAN header. This output is provided since it is possible to have a MSG_NMBR come from a tunable or Name block. The MSG_NMBR (and NODE_ID) inputs are only used by the control at network initialization time, which occurs following a reboot or following CAN_RST on the J1939_NTW block. So it is possible that the input may be tuned to a new value for the PGN yet the network is still using the prior value because it has not been re-initialized. Hence, this output can be a helpful troubleshooting tool.

The READ_ID output shows the Source Address of the device that sent the data currently being published at the block outputs. This is particularly useful when the NODE_ID filter is disabled (=255) or if the NODE_ID input has been tuned to a new value since the network was last initialized.

Endian

The J1939 standard requires the use of Little Endian (Intel format). This field must always be set False to use Little Endian for J1939.

Message Data

The data from the message is published to the DATA_X repeat group outputs. The data will be unpacked from the message using the same method described in Message Data for the CAN_WRITE block.

In order to facilitate determining data validity, the block will publish reception timing information and also includes a timeout feature. The OLD_DATA value is the time since this message was last received. This value is used by the block internally to compare against the TIMEOUT input. If the OLD_DATA>TIMEOUT, then the MSG_TO field will be set TRUE and the DATA_X values will be forced to their Default inputs. The Default inputs should always be set by the application engineer to be a safe condition for engine operation or to cause shutdown.

The DATA_RATE output is a troubleshooting output that gives the average data rate observed in the past seconds.

The NEW_DATA output will toggle TRUE for 1 rate group following message reception. This is useful if a message is being sent very infrequently and its arrival is intended to trigger an action.

DM_MANGR

This block is used to coordinate diagnostic data content for the supported DMx_MSG blocks. The EVENT_MGR block must be used to define the SPN and FMI combinations and to control active and logged status for each event. The DM_MANGR block provides the interface for the lamp status and control bits which are also sent as part of many of the diagnostic messages.

Each diagnostic message to be supported is connected to this DM_MANGR block so that it can inherit the diagnostic and lamp information as necessary.

DM1_MSG

This block is used to send all active diagnostic data in the standard J1939 format. If more than 1 diagnostic event is active, a multi-packet message will automatically be used to send the data. This block implements the BAM (Broadcast) method when necessary as required by the J1939-73 standard. It also implements the CTS/RTS method upon a properly configured remote request. The message is sent cyclically every 1 second as long as the block is enabled.

DM2_MSG

This block is used to send all logged but not currently active diagnostic data in the standard J1939 format. If more than 1 diagnostic event is to be sent, a multi-packet message will automatically be used to send the data. This block implements the BAM (Broadcast) method when necessary as required by the J1939-73 standard. It also implements the CTS/RTS method upon a properly configured remote request. DM2 messages are sent only upon RQST message from another device per the J1939 standard.

DM3_MSG

When a DM3 message request is received requesting a reset of logged and not currently active events that are not “emissions related” and not “protected”, the reset action will occur automatically as long as this block is enabled and the INHIBIT input is FALSE. A DM3 message should always have a response in accordance with the J1939 standard. The response may be either a positive acknowledgement that the events have been reset or a negative acknowledgement indicating that the events will not be reset. This special ack/nack message will be sent automatically based on the ENABLE and INHIBIT status.

It is the intent of the J1939 standard that controls which are not part of the OBD process use the DM3 message to reset logged faults. It is assumed that active faults reset automatically when the condition no longer exists.

DM11_MSG

When a DM11 message request is received requesting a reset of all active, pending, and logged events that are not “protected”, the reset action will occur automatically as long as this block is enabled and the INHIBIT input is FALSE. A DM11 message should always have a response in accordance with the J1939 standard. The response may be either a positive acknowledgement that the events have been reset or a negative acknowledgement indicating that the events will not be reset. This special ack/nack message will be sent automatically based on the ENABLE and INHIBIT status.

It is the intent of the J1939 standard that controls which are part of the OBD process use the DM11 message to reset faults.

DM5_MSG

This block is used to send diagnostic readiness data in the standard J1939 format. Readiness states and test support are defined in the application. The same message also reports the number of active and the number of previously active diagnostics from the event manager log. An indication of what OBD standard is supported in the application is also sent in the message (defined by the application).

DM6_MSG

This block is used to send a listing of all pending diagnostic trouble codes using the standard J1939 format. Pending diagnostics are those flagged by the application in the event manager (using PEND_X) as an observed problem but without enough run time to be active.

DM7_30_MSG

This block is used to provide support for both DM7 and DM30. As defined in J1939, a DM7 message is sent by a service tool to command a test or set of tests to be performed. The test commands are provided to the application for execution. Once complete, the test results are reported using a DM30 message in the standard J1939 format.

DM12_MSG

This block is used to send emissions related active diagnostic trouble codes using the standard J1939 format. Emissions related codes are flagged in the event manager using the EMISS_x inputs.

DM19_MSG

This block is used to send calibration data using the standard J1939 format. It provides a 32-bit checksum and name.

DM24_MSG

This block is used to send a listing of SPNs that are used in reporting Freeze Frame data (in DM25), Test Data (in DM7_30), and in application defined PGNs using the standard J1939 format. This block is required when using the DM7_30_MSG or DM25_MSG blocks to be compliant with the SAE J1939 standard.

DM25_MSG

This block is used to send Freeze Frame data using the standard J1939 format. Data to be sent is application defined.

DM32_MSG

This block is used to send emissions related diagnostic trouble codes along with timers for each diagnostic code indicating the total active time, total previously active time, and the time until derate using the standard J1939 format. Both active and previously active emissions related trouble codes are reported. Emissions related trouble codes are identified in the event manager using the EMISS_X input.

J1939_BAM

This block is used to send PGN data that exceeds 8-bytes in length. J1939 defines a number of PGNs greater than 8-bytes in length. Some contain application data such as a torque map. Others contain ASCII data for control and software identification to service tools and displays. In all cases, these long messages are sent very infrequently. Two possible methods are defined in J1939 for sending multi-packet data. This block implements both methods. The BAM method will be used when the block is triggered from the application or remotely requested using a SA = 255 in the request message. The CTS/RTS method will be sued when the block is remotely requested using a SA ≠ 255 in the request message.

The J1939 multi-packet BAM protocol requires at least 50 ms between message packets. There is an announcement message followed by the data messages. As an example, a message of 9-bytes will require at least 150 ms to be transmitted because it needs the announcement message and 2 data messages.

The J1939 multi-packet CTS/RTS protocol allows shorter times between message packets. The GAP implementation will send the packets at the fastest rate group used by any of the J1939 associated blocks in the application. Unlike the BAM method, the CTS/RTS method has 2-way negotiation for sending packets so the transportation time will depend on the requesting node as well as the GAP rate groups used.

Transmission Trigger

Multi-packet messages are normally sent only occasionally so this block is provided with a Boolean trigger input to cause the message to begin sending. There is no cyclic trigger method provided in order to avoid possible conflicts with the Diagnostic Messages. The only other method that can cause this message to be sent is for another node to request the PGN number defined at the MSG_NMBR input using a J1939 RQST message. This “On Request” feature is automatically provided. The only way to disable it is to disable the block using the BAM_MS_EN input.

It is only possible to send 1 multi-packet message at a time due to the BAM protocol method. Therefore, if more than one multi-packet message is to be triggered, they should be triggered in sequence only after the prior message is finished sending. The SENT output may be useful as a trigger for the next block in sequence. However, remember that the SENT output will also be set TRUE if another device requests this PGN using a RQST message.

CAN Header (ID)

The identifier (message header) used for the message transmission is created based on multiple block inputs. See the “Message Format Usage” section in Chapter 1 for details of the message header.

- PRIORITY input is used for the 3-bit priority portion of the header.
- MSG_NMBR input is used for the PGN portion of the header.

Regardless of the PDU type, the PGN will be equal to the MSG_NMBR input. If the PGN number indicates a PDU1 type message, the Destination Address must be added to the PGN number manually since there is no NODE_ID input for this purpose.

The CAN_ID_OUT output is the actual PGN number used based on the MSG_NUM input. It will only represent the PGN portion of the CAN header. This output is provided since it is possible to have a MSG_NMBR come from a tunable or Name block. The MSG_NMBR input is only used by the control at network initialization time, which occurs following a reboot or following CAN_RST on the J1939_NTW block. So it is possible that the input may be tuned to a new value for the PGN yet the network is still using the prior value because it has not been re-initialized. Hence, this output can be a helpful troubleshooting tool.

Message Data

The data used in the message contents comes from the DATA_X repeat group inputs. The data packing in the message will happen in the order it appears in the repeat group. However, for data longer than 1-byte, the byte ordering of the individual data item will be according to the Little Endian rules. See CAN_WRITE for an example.

J1939_CURV

This block is used to send A_CURVE table data. It may be desirable to send some tables to other devices so they can be aware of a certain configuration like a torque curve. J1939 does not specify a method to send curve or table data. There is at least one example in J1939 of a 2D curve (PGN64912). This block implements a similar method but supports larger curves and also supports 3D curves. Two possible methods are defined in J1939 for sending multi-packet data. This block implements both methods. The BAM method will be used when the block is triggered from the application or remotely requested using a SA = 255 in the request message. The CTS/RTS method will be used when the block is remotely requested using a SA ≠ 255 in the request message.

The J1939 multi-packet BAM protocol used to send data longer than 8-bytes requires at least 50 ms between message packets. There is an announcement message followed by the data messages. As an example, a message of 9-bytes will require at least 150 ms to be transmitted because it needs the announcement message and 2 data messages.

The J1939 multi-packet CTS/RTS protocol allows shorter times between message packets. The GAP implementation will send the packets at the fastest rate group used by any of the J1939 associated blocks in the application. Unlike the BAM method, the CTS/RTS method has 2-way negotiation for sending packets so the transportation time will depend on the requesting node as well as the GAP rate groups used.

Transmission Trigger

Multi-packet messages are normally sent only occasionally so this block is provided with a Boolean trigger input to cause the message to begin sending. There is no cyclic trigger method provided in order to avoid possible conflicts with the Diagnostic Messages. The only other method that can cause this message to be sent is for another node to request the PGN number defined at the MSG_NMBR input using a J1939 RQST message. This “On Request” feature is automatically provided. The only way to disable it is to disable the block using the MS_EN input.

It is only possible to send 1 multi-packet message at a time due to the BAM protocol method. Therefore, if more than one multi-packet message is to be triggered, they should be triggered in sequence only after the prior message is finished sending. The SENT output may be useful as a trigger for the next block in sequence. However, remember that the SENT output will also be set TRUE if another device requests this PGN using a RQST message.

CAN Header (ID)

The identifier (message header) used for the message transmission is created based on multiple block inputs. See the “Message Format Usage” section in Chapter 1 for details of the message header.

- PRIORITY input is used for the 3-bit priority portion of the header.
- MSG_NMBR input is used for the PGN portion of the header.

Regardless of the PDU type, the PGN will be equal to the MSG_NMBR input. If the PGN number indicates a PDU1 type message, the Destination Address must be added to the PGN number manually since there is no NODE_ID input for this purpose.

The CAN_ID_OUT output is the actual PGN number used based on the MSG_NUM input. It will only represent the PGN portion of the CAN header. This output is provided since it is possible to have a MSG_NMBR come from a tunable or Name block. The MSG_NMBR input is only used by the control at network initialization time, which occurs following a reboot or following CAN_RST on the J1939_NTW block. So it is possible that the input may be tuned to a new value for the PGN yet the network is still using the prior value because it has not been re-initialized. Hence, this output can be a helpful troubleshooting tool.

Message Data

The data used in the message contents comes from the table identified by the CURVE_BLK input. Data from the table is scaled according to the scaling inputs and packed into the message for transmission using all X values followed by all Y values followed by all Z values. Before any data values though is a revision identifier and an indication of the number of X values, number of Y values, and the number of Z values. This allows the receiving end to easily unpack the data correctly.

CAN_MSG

This special block allows any generic CAN message to be sent upon manual trigger.

Transmission Trigger

This block is provided with a Boolean trigger input to cause the message to be sent. There is no cyclic trigger method provided since this block does not provide normal J1939 features. These messages cannot be requested using a J1939 RQST message from another device.

CAN Header (ID)

The identifier (message header) used for the message transmission is created based only on the CAN_ID input. CAN_ID_EN must be set TRUE. The complete 29-bit CAN ID must be identified and used at the CAN_ID input. See the “Message Format Usage” section in Chapter 1 for details of the message header.

Message Data

ENDIAN must be set FALSE for J1939. The data used in the message contents comes from the DATA_X repeat group inputs. The data packing in the message will happen in the order it appears in the repeat group. However, for data longer than 1-byte, the byte ordering of the individual data item will be according to the Little Endian rules. See CAN_WRITE for an example.

The CAN_ID_OUT output is the actual CAN ID used based on the CAN_ID input. It will represent the entire CAN header – not just the PGN portion. This output is provided since it is possible to have a CAN_ID come from a tunable or Name block. The CAN_ID input is only used by the control at network initialization time, which occurs following a reboot or following CAN_RST on the J1939_NTW block. So it is possible that the input may be tuned to a new value for the CAN ID yet the network is still using the prior value because it has not been re-initialized. Hence, this output can be a helpful troubleshooting tool.

WRITE_B

This block is typically only used in J1939 networks. It provides the J1939 method of Boolean data which includes an Error indication and a Not Supported indication. Each WRITE_B consumes 2-bits of space in the message.

If a data item is supported but the source is faulted (like a discrete input is open circuit), then the ERROR input should be set TRUE to indicate to the receiving node(s) that the data state is invalid and cannot be used correctly. The actual data state at the BIN input is no longer available to the receiving node(s) because the ERROR input overrides it.

Since J1939 messages are pre-defined by the SAE committee, there may be data items in the message that are not supported by a given application. In these cases, the NOT_SUP input should be set TRUE to indicate to the receiving node(s) that this data should be ignored. The actual data state at the BIN input is no longer available to the receiving node(s) because the NOT_SUP input overrides it.

WR_BYTE

This block is not standard for J1939 applications but may be desired in some proprietary messaging. It simply provides 8 Boolean inputs to be packed into a single BYTE. Boolean input BIT_0 is put into the Least Significant Bit (LSB) position and Boolean input BIT_7 is put into the Most Significant Bit (MSB) position.

Since this block is not a standard J1939 format, there are no ERROR and NOT_SUP states available.

WRITE_I

This block is the most commonly used block for J1939 networks since all numeric data in J1939 is scaled, unsigned integer data. Most data ends on byte boundaries (8, 16, 24, 32-bits) but this block does allow odd sized data to be defined as well. This feature is provided since some J1939 PGNs require it. GAP Integer data is signed 32-bit but 32-bit values in J1939 are unsigned. A 32-bit unsigned value is not possible in GAP. If the full range must be supported for the application, contact Woodward engineering for a work-around solution.

If a data item is supported but the source is faulted (like an analog input is open circuit), then the ERROR input should be set TRUE to indicate to the receiving node(s) that the data is invalid and cannot be used correctly. The actual data at the IIN input is no longer available to the receiving node(s) because the ERROR input overrides it. This input is ignored if the BITS input is not a multiple of 8 (ends on a byte boundary).

Since J1939 messages are pre-defined by the SAE committee, there may be data items in the message that are not supported by a given application. In these cases, the NOT_SUP input should be set TRUE to indicate to the receiving node(s) that this data should be ignored. The actual data state at the IIN input is no longer available to the receiving node(s) because the NOT_SUP input overrides it. This input is ignored if the BITS input is not a multiple of 8 (ends on a byte boundary).

WRITE_A

This block is not standard for J1939 applications but may be desired in some proprietary messaging. It simply provides a method to send analog data in IEEE floating-point format. It consumes 32-bits of space in the message.

Since this block is not a standard J1939 format, there are no ERROR and NOT_SUP states available.

READ_B

This block is typically only used in J1939 networks. It supports the J1939 method of Boolean data which includes an Error indication and a Not Supported indication. Each READ_B unpacks 2-bits of space in the message.

If the Error flag is set in the data, the ERROR output on the block will be set TRUE. In this case the data at the BOUT output should be ignored because it is not valid.

If the Not Supported flag is set in the data, the NOT_SUP output on the block will be set TRUE. In this case the data at the BOUT output should be ignored because it is not valid.

If the MSG_TO conditions are TRUE, the DEF_BOUT input field will be applied to the BOUT output. To lock in the last read value, wrap the output around to the DEF_BOUT input. A ZMINUS1_B is not necessary for this as the functionality is built into the block already.

READ_BYTE

This block is not standard for J1939 applications but may be desired in some proprietary messaging. It simply unpacks a full byte into 8 Boolean outputs. Boolean output BIT_0 comes from the Least Significant Bit (LSB) position and Boolean output BIT_7 comes from the Most Significant Bit (MSB) position.

Since this block is not a standard J1939 format, there are no ERROR and NOT_SUP states available.

If the MSG_TO conditions are TRUE, the DEF_BIT_x input fields will be applied to the BIT_x outputs. To lock in the last read state, wrap the output around to the corresponding DEF_BIT_x input. A ZMINUS1_B is not necessary for this as the functionality is built into the block already.

READ_I

This block is the most commonly used block for J1939 networks since all numeric data in J1939 is scaled, unsigned integer data. Most data ends on byte boundaries (8, 16, 24, 32-bits) but this block does allow odd sized data to be expected as well. This feature is provided since some J1939 PGNs require it.

GAP Integer data is signed 32-bit. All J1939 values are unsigned (assuming they conform to the standard). Any data item on J1939 that is smaller than 32-bits will map into a GAP Integer perfectly. However, a 32-bit unsigned value from J1939 cannot be directly used in GAP. If the full range of a 32-bit unsigned integer must be supported for the application, contact Woodward engineering for a work-around solution.

If the Error flag is set in the data, the ERROR output on the block will be set TRUE. In this case the data at the IOUT output should be ignored because it is not valid. This output is not valid if the BITS input is not a multiple of 8 (ends on a byte boundary).

If the Not Supported flag is set in the data, the NOT_SUP output on the block will be set TRUE. In this case the data at the IOUT output should be ignored because it is not valid. This output is not valid if the BITS input is not a multiple of 8 (ends on a byte boundary).

If the MSG_TO conditions are TRUE, the DEF_IOUT input field will be applied to the IOUT output. To lock in the last read value, wrap the output around to the DEF_IOUT input. A ZMINUS1_I is not necessary for this as the functionality is built into the block already.

READ_A

This block is not standard for J1939 applications but may be desired in some proprietary messaging. It simply provides a method to receive analog data sent in IEEE floating-point format. It unpacks 32-bits of space in the message.

Since this block is not a standard J1939 format, there are no ERROR and NOT_SUP states available.

If the MSG_TO conditions are TRUE, the DEF_AOUT input field will be applied to the AOUT output. To lock in the last read value, wrap the output around to the DEF_AOUT input. A ZMINUS1 is not necessary for this as the functionality is built into the block already.

Chapter 4. Application Examples

J1939 always communicates numeric data using unsigned integer values. Therefore, all data that may be negative must first have an offset applied. Also, since the data is integer, any data with relevant information to the right of the decimal point must have a linear scale factor applied. Normally all J1939 data is linearly scaled to exactly fit the valid signal maximum range for the data size used (maximize resolution).

The below table shows the valid signal ranges for each data size. Both hexadecimal and decimal formats are shown. The hexadecimal format is shown as it helps to explain the range visually. If the data provided to the block could exceed the Valid Signal range, then an A_LIMITER or A_MAX function must be used otherwise the diagnostic bits will be set at the receiving end.

Range	1-byte Signal	2-byte Signal	4-byte Signal
Valid Signal	0 – 250 0 – FA hex	0 – 64255 0 – FAFF hex	0 – 4211081215 0 – FAFFFFFF hex
Special Purpose	251 FB hex	64256 – 64511 FB00 – FBFF hex	4211081216 – 4227858431 FBxxxxxx hex
Reserved	252 – 253 FC – FD hex	64512 – 65023 FC00 – FDFF hex	4227858432 – 4261412863 FC000000 – FDFFFFFF hex
Error Indication	254 FE hex	65024 – 65279 FExx hex	4261412864 – 4278190079 FExxxxxx hex
Not Available Indication	255 FF hex	65280 – 65535 FFxx hex	4278190080 – 4294967294 FFxxxxxx hex

Table 4-1. Valid Ranges

Interpreting the J1939 standard

A typical PGN definition from J1939-71 is shown below:

PGN 61460		**Blade Information**	**-BI**
Transmission Repetition Rate:		50 ms	
Data Length:		8	
Extended Data Page:		0	
Data Page:		0	
PDU Format:		240	
PDU Specific:		20	
Default Priority:		3	
Parameter Group Number:		61460 (0xF014)	
Start Position	**Length**	**Parameter Name**	**SPN**
1-2	2 bytes	Relative Blade Height	3365
3-4	2 bytes	Blade Rotation Angle	3331
5	1 byte	Relative Blade Height & Blade Rotation Angle Measurement Latency	3366
6.1	2 bits	Relative Blade Height Figure of Merit	3367
6.3	2 bits	Blade Rotation Angle Figure of Merit	3332

Observations:

- PDU Format (PF) is listed as 240 therefore this is a PDU2 type message.
- MSG_NUM = 61460 (PGN number)
- NODE_ID = don't care – input field will be ignored since this is a PDU2 type message.
- PRIORITY = 3
- Transmission Repetition Rate is listed as 50 ms. To create this transmission rate in GAP, use MSG_TYPE = "CYCLIC", RATE GROUP = 10, and SYNC_SKIP = 5.
- Note that the Data Length (3rd table row) is listed as 8 bytes. However, looking at the data definition in the last 5 table rows, there are only 6 bytes of data defined. This is typical in J1939. The "leftover" bits and bytes are filled with "Not Supported".

Below is the SPN definition from J1939-71 from the first SPN in the above PGN:

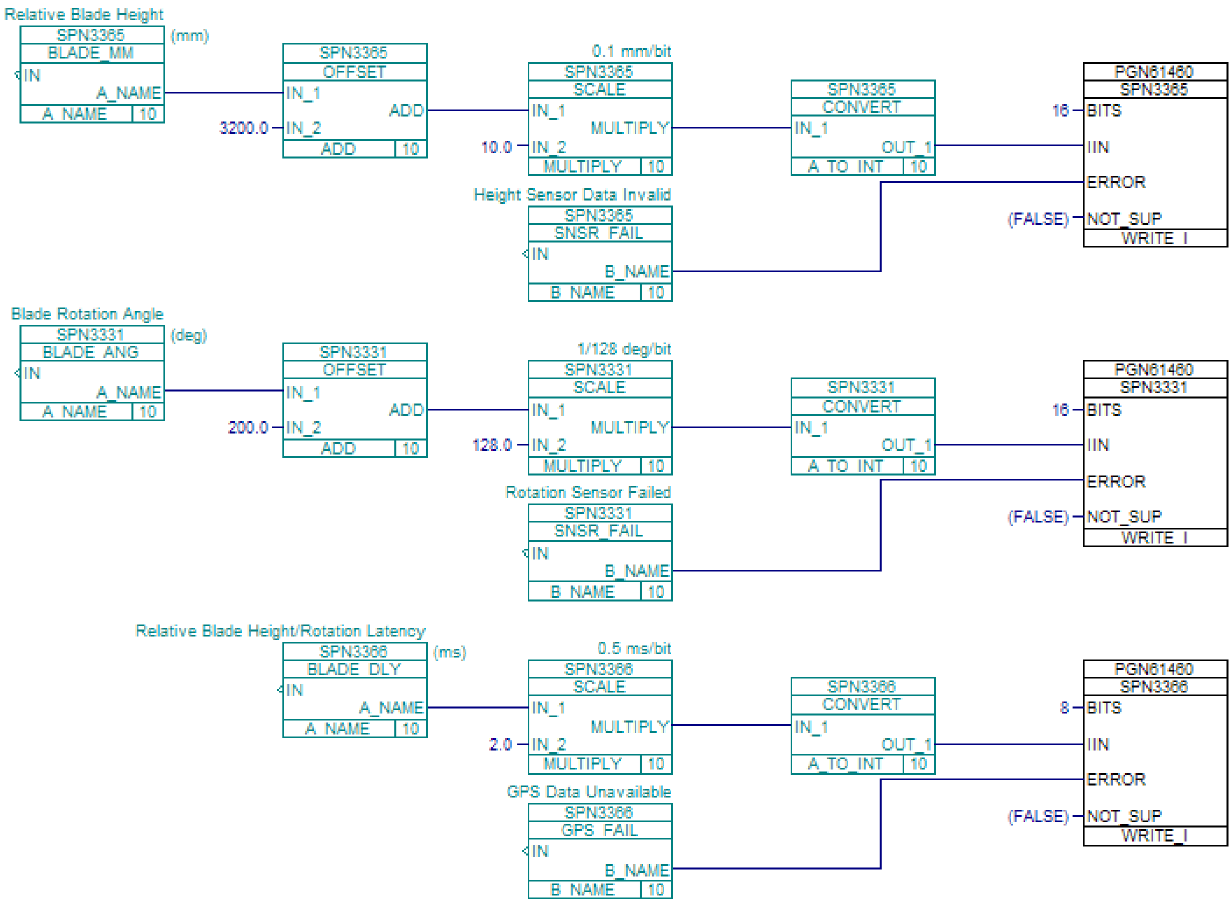
SPN 3365	**Relative Blade Height**	
Data Length:	2 bytes	
Resolution:	0.1 mm/bit, -3,200 mm offset	
Data Range:	-3,200 to 3,225.5 mm	Operational Range: same as data range
Type:	Measured	
PGN reference:	61460	Document: J1939-71

Observations:

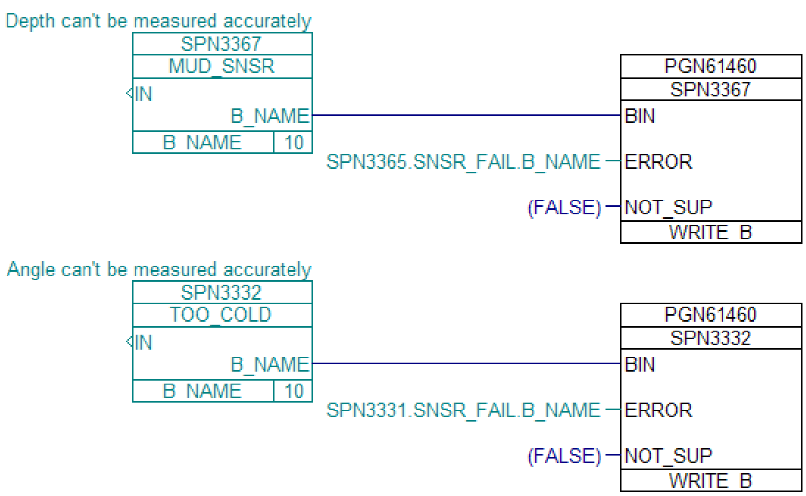
- Resolution provides the scale and offset of the data. In this case, to send the data it must first be added to 3200 so that all possible values in the data range will be positive (since J1939 only uses unsigned communication). Then the data must be scaled with a scale factor of 10.
- The entire data range is 3200 + 3225.5 = 6425.5
- To double check the scaling and offset are applied correctly, take the data range and apply the scale factor: 6425.5 * 10 = 64255. This is the largest number that will be communicated on J1939. This is also the largest data value allowed in J1939 for a 2-byte value.
- This is a "Measured" value. Hence, it is intended to be communication of sensor data and is not intended to be a Commanded value. There may be another PGN available to provide a Commanded value for this purpose.

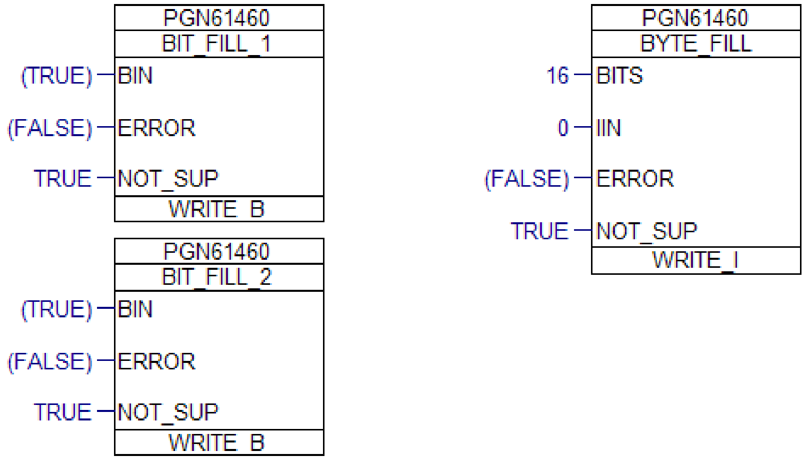
Transmit PGN61460 GAP Sample

Here is sample GAP logic used to create PGN 61460 for transmission from the control:

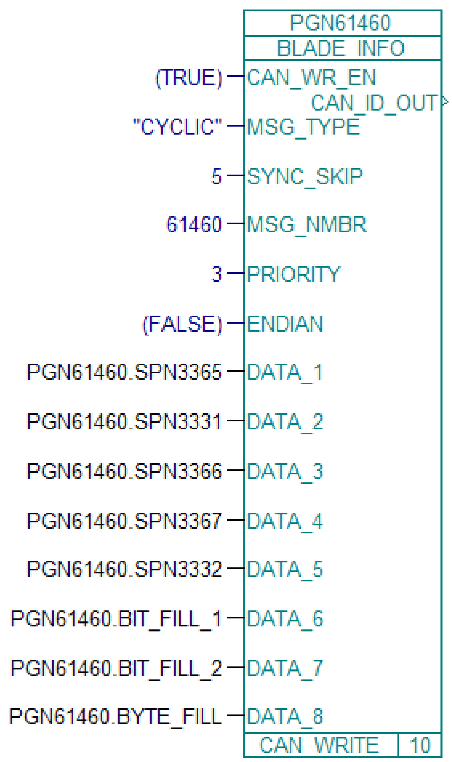


This example assumes that data for all three SPNs is available in the control. If one or more is not supported, the corresponding NOT_SUP input field would need to be set TRUE. Likewise, the data for the Boolean states below is also assumed to be available.





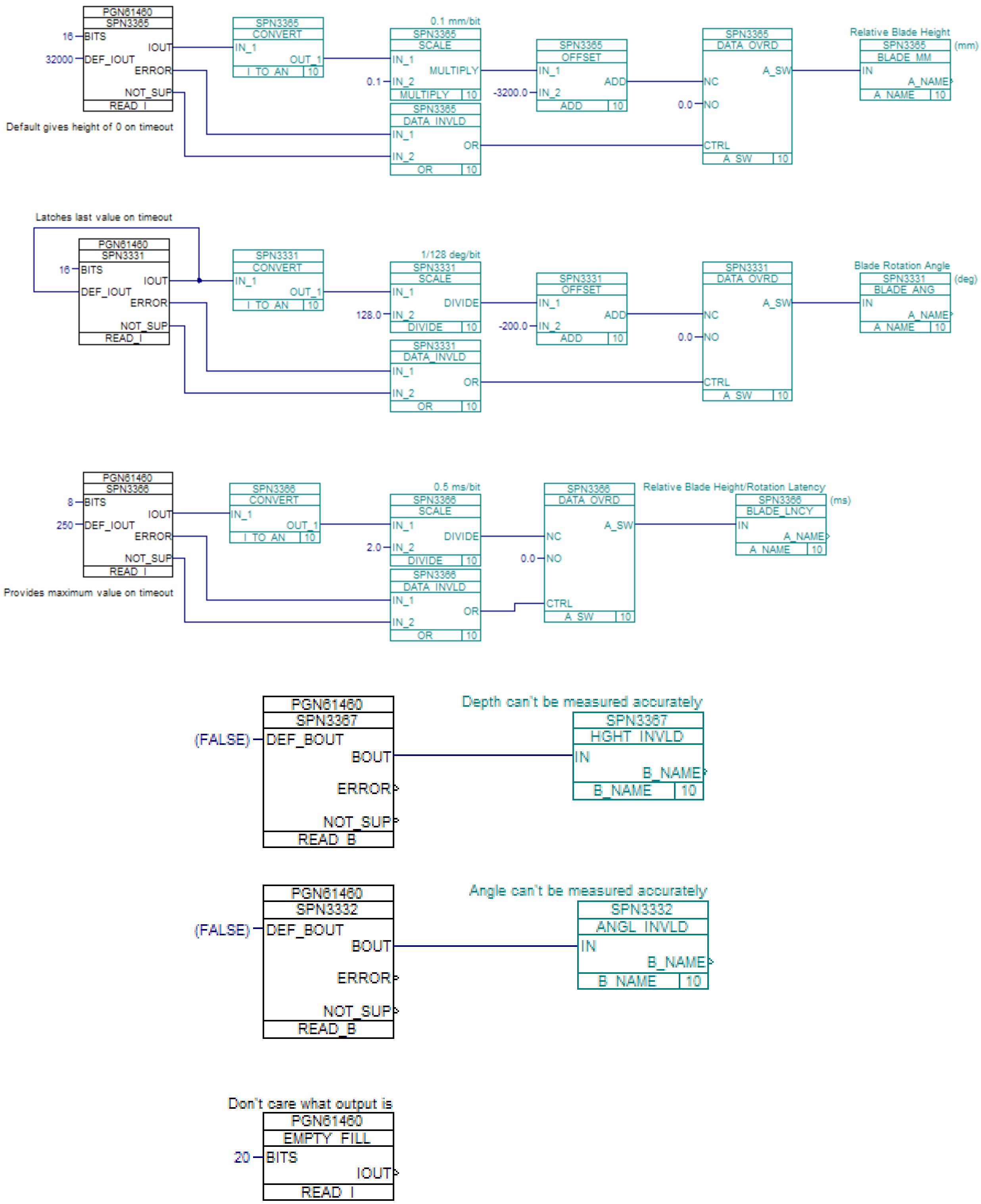
The two "bit filler" blocks are used to fill out the full byte. It is always required for CAN messages defined in GAP to end on a byte boundary once all data is added together. The "byte filler" block is used to make the message a full 8-bytes long as required in the J1939 message definition. If this block were omitted, GAP would only create a 6-byte message.



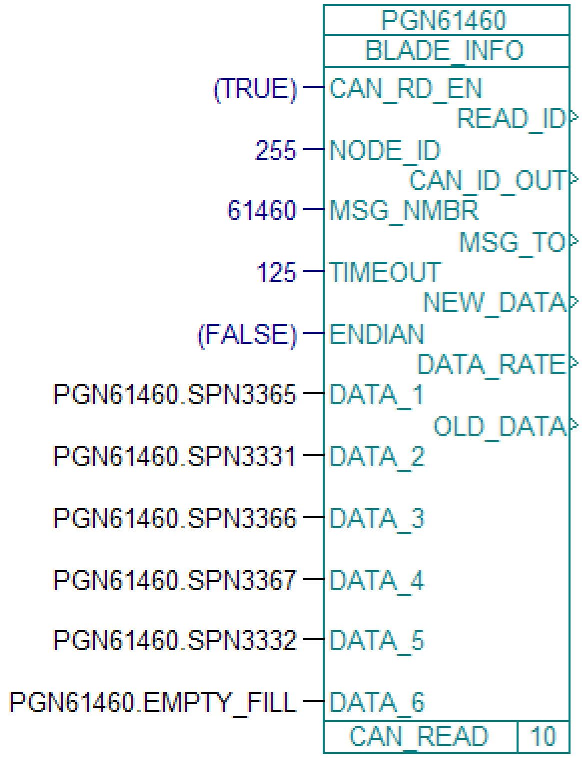
This block shows the correct order of the DATA_x blocks in order to create the order shown in the J1939 message definition. The MSG_TYPE and SYNC_SKIP values are also set to create a 50 ms transmission rate.

Receive PGN61460 GAP Sample

Here is sample GAP logic used to receive PGN 61460:



Default value conditions were selected for example ideas only. The 20-bit filler is required because the message is defined to be a full 8-bytes.



This example shows the NODE_ID set to 255 which disables Node ID filtering. This message will be received from any node. The TIMEOUT was set at 2.5x expected transmission frequency. Select a value appropriate for the application of the message.

RQST Message

Two different methods may be used to create a J1939 RQST (Request for PGN) message. The decision of which method to use will be based on how the message is to be triggered. If a cyclic trigger is desired to slowly retrieve data from another device, the CAN_WRITE method may be preferred. If the trigger is rare or manual, the CAN_MSG method may be preferred. Any given PGN should only be requested at 1 Hz or slower.

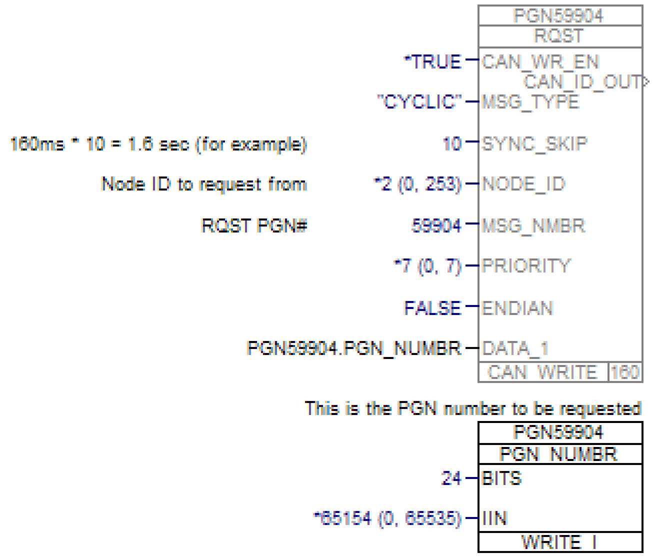


Figure 4-1. CAN_WRITE RQST Method

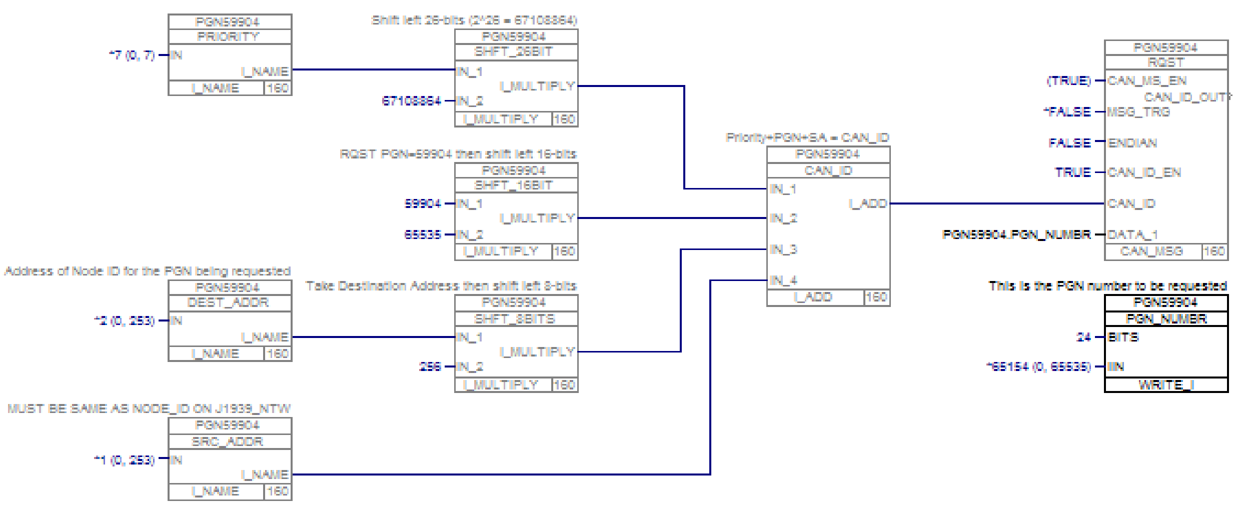


Figure 4-2. CAN_MSG RQST Method

Appendix. Definitions and Abbreviations

Definitions

Acknowledgment (ACK)	Confirms that the requested action has been understood and performed.
Address	The 8 bit field (or fields) used to define the source (and destination when applicable) of a message (e.g. engine, transmission, etc.).
Arbitration	The process by which one or more devices resolve conflicts in obtaining access to a shared network bus.
Bit Stuffing	A procedure used to assure the transmitted and received messages maintain a minimum number of dominant to recessive edges, and vise versa, to maintain the proper resynchronization within the string of bits in a CAN Data Frame.
Bridge	A device which stores and forwards messages between two J1939 network segments. This permits changes in the media, the electrical interface, and data rate between segments. The protocol and address space remain the same on both sides of the bridge. Note that a bridge may selectively filter messages going across it so that the bus load is minimized on each segment.
Bus	See Segment.
CAN Data Frame	The ordered bit fields necessary to create a CAN frame used to convey data, beginning with an SOF and ending with an EOF.
Cyclic Redundancy Check (CRC)	An error control mechanism. A 15 bit cyclic redundancy check is performed for detecting transmission errors. Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a frame check sequence so that the resulting frame, consisting of k + n bits is exactly divisible by some predefined number. The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error.
Data Field	A 0 to 64-bit field normally placed in a CAN data frame which contains the data as defined in the Application Layer (document J1939-71).
Data Page	One bit in the Identifier portion of the CAN Arbitration Field is used to select one of two pages of Parameter Group Numbers. This provides for the future growth of Parameter Group definitions. It also is one of the fields used to determine the Parameter Group Number which labels the data field of the CAN Data Frame.
Destination Address (DA)	This is a Protocol Data Unit (PDU) specific field in the 29 bit CAN identifier used to indicate the address of the device intended to receive the J1939 message.
Device	A physical component with one or more ECUs and network connections.

Electronic Control Unit (ECU)	A computer based electronic assembly from which J1939 messages may be sent or received.
End of Frame (EOF)	A 7 bit field marking the ending of a CAN data frame.
Extended Frame	A CAN data frame using a 29 bit identifier as defined in the CAN 2.0 specification.
Frame	A series of data bits making up a complete message. The frame is subdivided into a number of fields, each field containing a predefined type of data. See CAN Data Frame
Function	A capability of a vehicle system having one or more ECUs that are connected to a J1939 bus segment of a Vehicle System. The function value is used in the 8 bit Function field in the 64 bit NAME entity (See J1939-81, Section 4.1)
Gateway	This device permits data to be transferred between two networks with different protocols or message sets. The gateway provides a means to repackage parameters into new message groups when transferring messages from one segment to another.
Group Extension (GE)	This is a PDU specific field of a J1939 CAN Data Frame that is used as part of the information necessary to determine the Parameter Group Number.
Identifier	The identifier portion of the CAN arbitration field.
Idle	A state on the CAN bus where no node is transmitting or attempting to transmit data.
Media	The physical entity which conveys the electrical transmission (or equivalent means of communication) between devices on the network. For J1939-11, the media consists of shielded twisted pair copper wires.
Message	A "message" is equivalent to one or more "CAN Data Frames" that have the same Parameter Group Number. For instance the information related to a single Parameter Group Number to be transferred on the bus may take several CAN data frames.
Multipacket Messages	A type of J1939 message which is used when more than one CAN data frame is required to transmit all data specific to a given Parameter Group Number. Each CAN data frame will have the same identifier but will contain different data in each packet.
NAME	An 8-byte value which uniquely identifies the primary function of an ECU and its instance on the network. A device's NAME must be unique, no two devices may share the same NAME value on a given vehicle network.
Node	A specific hardware connection of a device to the physical media. A specific node may have more than one address claimed on the network.
Non-Volatile	Retention of changeable memory values even though power is turned off for any reason. This term is used with respect to data values, such as node addresses or NAMEs, that are changed during use.

Read Only Memory (ROM)	This type of memory is technically non-volatile, but is not changeable during use and thus not what is referred to in these documents.
Negative-Acknowledgment (NACK)	A response which indicates that a message has not been understood or a requested action could not be performed.
Packet	A single CAN data frame. This can also be a message if the Parameter Group to be transferred can be expressed in one CAN data frame.
Parameter Group (PG)	A collection of parameters that are conveyed in a J1939 message. Parameter Groups include commands, data, requests, acknowledgments, and negative-acknowledgments. The PG identifies the data in a message, regardless of whether it is a single packet or multipacket message. Parameter Groups are not dependent on the source address field thus allowing any source to send any Parameter Group.
Parameter Group Number (PGN)	A three byte, 24 bit, representation of the Reserved Bit, Data Page, PDU Format, and GE fields. The Parameter Group Number uniquely identifies a particular Parameter Group.
PDU Format (PF)	An 8 bit field in the 29 bit identifier that identifies the PDU format and is used in whole or in part to provide a label for a Parameter Group. It also is one of the fields used to determine the Parameter Group Number which labels the data field of the CAN Data Frame.
PDU Specific (PS)	An 8-bit field in the 29 bit identifier whose definition depends upon the value of the PDU Format field. It can be either a destination address (DA) or Group Extension (GE). It also is one of the fields used to determine the Parameter Group Number which labels the data field of the CAN Data Frame.
PDU1 Format	A PDU format used for messages that are to be sent to a destination address (DA). The PS field contains the destination address (specific or global).
PDU2 Format	A PDU format used to send information that has been labeled using the Group Extension technique. This PDU does not contain a destination address. The PS field contains the Group Extension in the case of PDU2 formats.
Preferred Address	The address that an ECU will attempt to use first when claiming an address. Preferred Addresses are assigned in the J1939 standard.
Priority	A 3-bit field in an identifier that establishes the arbitration priority of the information communicated. The highest priority is zero and the lowest priority is seven.
Protocol Data Unit (PDU)	A PDU is a J1939 specific CAN Data Frame format.
Remote Transmission Request (RTR)	A feature of the CAN protocol allowing a device to request that another device (or devices) send a message. This feature of CAN is not used in J1939. An alternate request mechanism is specified for J1939.

Repeater	A device which regenerates the bus signal onto another segment of media. This permits the network to connect more devices onto the bus, or to connect to another type of media (Physical Layer Expansion). The speed (data rate), protocol (data link layer), and address space are the same on both sides of the repeater. For J1939, any delays in regenerating the data signal must be kept to a very small fraction of one bit interval.
Reserved Bit	A bit in a J1939 29-bit identifier reserved for future definition by SAE. It also is one of the fields used to determine the Parameter Group Number which labels the data field of the CAN Data Frame.
Router	A device which allows segments with independent address spaces, data rates, and media to exchange messages. A router may permit each segment to operate with minimum bus loading yet still obtain critical messages from remote segments. The protocol remains the same across all segments. Note that the router must have look up tables to permit the translation and routing of a message with ID X on segment 1 to ID Y on segment 2.
Segment	The physical media and attached nodes of a network not interconnected by network interconnection devices. A single segment of a network is characterized by all of the devices “seeing” the signal at the same time (i.e., there is no intermediate device between electrical sections of the network). Multiple segments can be connected together by network interconnection devices including repeaters, bridges, and routers.
Source Address (SA)	An 8-bit field in the 29-bit identifier which allows for the unique identification of the source of a message. The SA field contains the address of the device that is sending the message.
Standard Frame	A CAN data frame using an 11-bit identifier as defined in the CAN 2.0b specification.
Start of Frame (SOF)	The initial bit in a CAN frame serving only to indicate the beginning of the frame.
Subnetwork	This refers to the network activity (message traffic) on a specific J1939 segment when multiple segments are used. Subnetworks may include: Tractor; Trailer, Implement, and Braking System. Note that they may be separated by a bridge or router to minimize total bus loading. Collectively the subnetworks are the J1939 Vehicle Network.
Vehicle	A machine which, in most applications, includes a capability to propel itself and includes one or more J1939 segments. A vehicle may be assembled of one or more Vehicle systems which are connected together to form the whole vehicle.

Vehicle System	A subcomponent of a vehicle, or a component that is analogous to a subcomponent of a vehicle, that includes one or more J1939 segments and may be connected or disconnected from the vehicle. A Vehicle System may be made up of one or more Functions, which have ECU's that are connected to a J1939 segment of the Vehicle System.

Abbreviations

ACK	Acknowledgment
AP	Accelerator Pedal
ASCII	American Standard Code for Information Interchange
BAM	Broad Announcement Message
CAN	Controller Area Network
Con-Ag	Construction-Agriculture Industry
CRC	Cyclic Redundancy Check
DA	Destination Address
DLC	Data Length Code
DP	Data Page
ECU	Electronic Control Unit
EOF	End of Frame
GE	Group Extension
ID	Identifier
IDE	Identifier Extension Bit
LLC	Logical Link Control
LSB	Least Significant Byte or Least Significant Bit
MAC	Medium Access Control
MID	Message Identifier
MSB	Most Significant Byte or Most Significant Bit
NA	Not Allowed
NA	Not Available
NACK	Negative-Acknowledgment
OSI	Open System Interconnect
P	Priority
PDU	Protocol Data Unit
PF	PDU Format
PG	Parameter Group
PGN	Parameter Group Number
PID	Parameter Identifier
PS	PDU Specific
PS_GE	PDU Specific - Group Extension
PS_DA	PDU Specific - Destination Address
R	Reserved
RTR	Remote Transmission Request
SA	Source Address
SID	Subsystem Identifier
SLOT	Scaling, Limits, Offset, and Transfer Function
SOF	Start of Frame
SPN	Suspect Parameter Number
SRR	Substitute Remote Request

We appreciate your comments about the content of our publications.

Send comments to: icinfo@woodward.com

Please reference publication 51321A.

B51321:A

PO Box 1519, Fort Collins CO 80522-1519, USA
1000 East Drake Road, Fort Collins CO 80525, USA
Phone +1 (970) 482-5811 • Fax +1 (970) 498-3058

Email and Website—www.woodward.com

Woodward has company-owned plants, subsidiaries, and branches, as well as authorized distributors and other authorized service and sales facilities throughout the world.

Complete address / phone / fax / email information for all locations is available on our website.

2012/12/Colorado