A Comprehensible Guide to J1939

By Wilfried Voss

Published by
Copperhill Technologies Corporation
158 Log Plain Road
Greenfield, MA 01301

ISBN: 978-0-9765116-3-2

Printed in the United States of America

Limit of Liability/Disclaimer of Warranty

Copperhill Technologies Corporation
http://www.copperhilltech.com
158 Log Plain Road
Greenfield, MA 01301

About this book

After writing “A Comprehensible Guide to Controller Area Network”, documenting the SAE J1939 standard seemed to be a logical choice when it came to investigating CAN based higher layer protocols. As I have learned from a number or professionals in the CAN industry, J1939 is still gaining enormous popularity, even though it is already in business for some years. However, the quality and availability of documentation on J1939 is in utter contrast to its popularity.

According to Wikipedia, the biggest multilingual free-content encyclopedia on the Internet, "SAE J1939 is the vehicle bus standard used for communication and diagnostics among vehicle components, originally by the car and heavy duty truck industry in the United States." Beyond that statement there is only few information to find on J1939 and the same is true for the entire World Wide Web.

The SAE (Society of Automotive Engineers), like many standardization organizations, is keeping a close lock on their written works and profound information on their web site is sparse. Downloading the J1939 PDF documents comes with a hefty price tag, even for SAE members who receive a discount not worth mentioning.

At the time when this book was released the only available and complete technical reference on J1939 was provided by the SAE either as PDF downloads for a price tag of US$595.00 (Single-User, one-year subscription) or one could buy the complete SAE J1939 standard in one colossal work of 1600+ pages for a mere US$310.00. Regular books dedicated to J1939 and available for a reasonable price did not exist at the time when I started the research and, as mentioned previously, valid references on the World Wide Web were extremely sparse.

Beyond the commercial aspects there is also the issue of educational value and readability of these standards. Standards, like those developed for J1939, are not designed to educate or, God forbid, entertain.

Standardization organizations seem to be the worst when it comes to providing comprehensible and readable documentation of the products they are trying to promote, which ironically creates revenues for technical writers who actually know their job.

Despite the poor condition of the written standard, it was initially a pleasure to investigate the J1939 protocol functions. SAE J1939 is a very ingeniously designed protocol that takes a resourceful advantage of the CAN 29-Bit message identifier. Rather than relying on a myriad of protocol functions, SAE J1939 uses predefined parameter tables, which keeps the actual protocol on a comprehensible level. SAE J1939 is a prime example of good American engineering according to the KISS principle (KISS = Keep It Simple, Stupid!), but it is nevertheless at least as effective as, for instance, CANopen or DeviceNet.

I had originally contemplated to continue my "Comprehensible Guide" series with CANopen, but was overwhelmed by the amount of information that I would need to compile, which also indicates the effort it takes for the newcomer to get familiarized with the topic. SAE J1939 was so much more fun to investigate (again, initially), because it seemed simple and straight-forward. However, this conclusion can only be made after relentless digging through the standards and, after repeated reading, finally understanding what the authors were trying to convey to the reader.

This book is an attempt to create an enjoyable and readable J1939 reference for everybody. The information provided in this book is, besides the SAE J1939 Standards Collection, based on publicly available information such as, but not limited to web sites and printed literature as well as contributions by engineers familiar with Controller Area Network and the J1939 protocol. The information in this book, while based on the J1939 standard, is not a reproduction of any copyrighted SAE document.

Also, this book does not intend to replace the entire SAE J1939 Standards Collection, especially since the standards SAE J1939 and SAE J1939/71 contain mainly data references which account to more than 1000 pages of 8.5 x 11" in size. These data references are not part of this book. The mere intention was to explain the standard in the sense of being a comprehensible guide.

I also need to apologize in advance that the information in this book may seem to be repetitive at times.

First of all, I always try to provide a generic overview of the topic covered in my books. This will help people with a lesser technical background to understand the technology without having to read all details.

Initially I have been trying to describe the collective SAE J1939 Standard by going through the sequence of documents as numbered by the SAE (J1939, J1939/01…J1939/81). As it turns out, a great amount of information in these documents is redundant. It really seems to be the case that individual groups with different interests created their individual documents, sometimes referring to information in other documents, and in other cases reproducing the same information in different form.

In all consequence, the only documents needed to understand the protocol features are:

J1939-21 Data Link Layer
J1939-81 Network Management

The information in this book is based on these documents.

Let me point to the legal disclaimer that states that the publisher and author have used their best efforts in preparing this book.

I would also like to take the opportunity and apologize to all engineers of the SAE who worked on the J1939 standards collection. My comments throughout this book, regarding the condition of the documentation, are not favorable. You have created a great protocol, but the standard is poorly written and lacks any visible structure. Working through the standard was at times tiresome and frustrating.

It was especially irritating to learn that the SAE engineers who created the standard were not fully familiar with the CAN specification. The SAE J1939 Standards Collection contains a number of references to the CAN standard that are misleading in the best case, while others are plain wrong.

One would also expect that engineers, regardless of their special expertise, are familiar with the unit of time, "ms" or "msec" (milli-seconds). Instead the J1939 standard uses mS, which is officially milli-Siemens (electric conductance, equal to inverse Ohm - Ω).

Last, but not least, in case you have questions related to J1939 and/or you would like to contact me, please do so (by any means) through one of my web sites, http://www.J1939Forum.com. Just post an inquiry and either I or the community or both will respond.

About the author

Wilfried Voss is the President of esd electronics, Inc., a company specializing in CAN technology. The company is located in Greenfield, Massachusetts. Mr. Voss has worked in the CAN industry since 1997 and before that was a specialist in the paper industry. He has a master's degree in electrical engineering from the University of Wuppertal in Germany.

Mr. Voss has conducted numerous seminars on CAN and CANopen during various *Real Time Embedded And Computing Conferences* (RTECC), ISA (Instrumentation, Systems, and Automation Society) conferences and various other events all over the United States and Canada. He is also the founder of Copperhill Technologies, a software engineering and consulting company, and the creator of VisualSizer, a comprehensive servo motor sizing software.

Mr. Voss has traveled the world extensively, settling in New England in 1989. He presently lives in an old farmhouse in Greenfield, Massachusetts with his Irish-American wife, their son Patrick and their Rhodesian Ridgeback.

Acknowledgements by the author

This book would not have been possible without the help of my wife, Dr. Susan Marie Voss, a vigorous proof reader and source of many inspirations. Special appreciation is in order for my son Patrick (he was one year old at the release of this book) who taught me how to sufficiently type with only one hand, left or right, while the other was busy keeping him away from the keyboard’s *Sleep* button.

A great deal of gratitude shall be attributed to the Boston Red Sox, World Series Champions of 2004 and 2007. Go Sox!

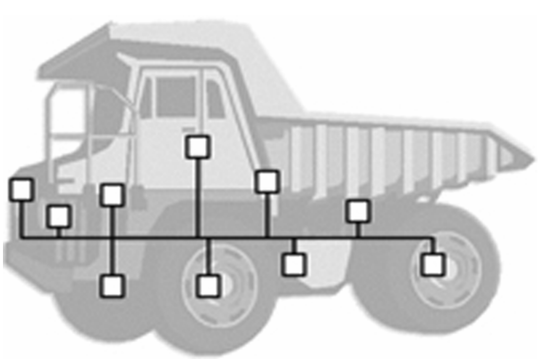
Table of Contents

Chapter
1

Introduction to J1939

The Society of Automotive Engineers (SAE) Truck and Bus Control and Communications Subcommittee has developed a family of standards concerning the design and use of devices that transmit electronic signals and control information among vehicle components. SAE J1939 and its companion documents have quickly become the accepted industry standard and the Controller Area Network (CAN) of choice for off-highway machines in applications such as construction, material handling, and forestry machines.

J1939 is a higher-layer protocol based on Controller Area Network (CAN). It provides serial data communications between microprocessor systems (also called Electronic Control Units - ECU) in any kind of heavy duty vehicles. The messages exchanged between these units can be data such as vehicle road speed, torque control message from the transmission to the engine, oil temperature, and many more.



The main advantages of using CAN as a field-bus technology are reduced wiring (CAN requires only two wires between nodes), extremely reliable communication, easy implementation and improved maintenance and service capabilities, which consequently not only produce better vehicle performance, but also help to reduce production costs.

J1939-based protocols are used in:

- Diesel power-train applications
- In-Vehicle networks for trucks and buses
- Agriculture and forestry machinery (ISO 11783)
- Truck-Trailer connections
- Military vehicles (MiLCAN)
- Fleet management systems
- Recreational vehicles
- Marine navigation systems (NMEA2000)

The protocol features of J1939 are based on two older SAE (Society of Automotive Engineers) specifications:

1. SAE J1708

SAE J1708 specifies on the physical layer of the communication link. It uses RS485 as an electrical layer operating at 9600 baud. (Note: Unlike RS232/485 there are no message collisions under CAN). Messages under J1708 start with a Message Identification Character, followed by the data information and a checksum. The message length is 21 characters (or less) and each data character is 10 bits long. Each character starts with a start bit of low polarity.

2. SAE J1587

SAE J1587 is a joint SAE/TMS "Recommended Practices for Electronic Data Exchange Between Microcomputer Systems in Heavy-Duty Vehicle Applications". It regulates the communication and standardized data exchange between ECUs based on J1708 networks.

Note: The situation regarding documents/literature on J1708 and J1587 is as dire as with J1939.

J1939 is designed to replicate the functionality of J1708 and J1587 including control system support. Vehicle applications may utilize either one of both specifications.

Chapter
2

Overview – Controller Area Network and J1939

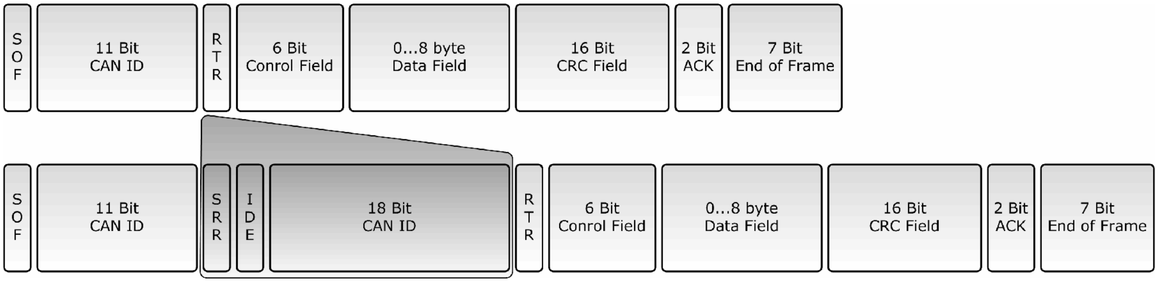
The standard CAN message frame uses an 11-bit message identifier (CAN 2.0A), which is sufficient for the use in regular automobiles and any industrial application, however, not necessarily for off-road vehicles.

The Society of Automotive Engineers (SAE) Truck and Bus Control and Communications Subcommittee had developed a family of standards concerning the design and use of devices that transmit electronic signals and control information among vehicle components. As a result, the higher layer protocol SAE J1939, based on CAN, was born, which was required to provide some backward-compatible functionality to older RS485-based communication protocols (J1708/J1587).

In order to serve these demands, the CAN standard needed to be enhanced to support a 29 bit message identifier. The ISO 11898 amendment for an extended frame format (CAN 2.0B) was introduced in 1995.

The 29 bit message identifier consists of the regular 11 bit base identifier and an 18 bit identifier extension. The distinction between CAN base frame format and CAN extended frame format is accomplished by using the IDE bit inside the Control Field. A low (dominant) IDE bit indicates an 11 bit message identifier, a high (recessive) IDE bit indicates a 29 bit identifier.

An 11 bit identifier (standard format) allows a total of 2^{11} (= 2048) different messages. A 29 bit identifier (extended format) allows a total of 2^{29} (= 536+ million) messages.



Picture 2.1 Extension from 11-Bit to 29-Bit CAN Identifier

The above picture shows a comparison between a standard CAN data frame with an 11-Bit identifier and a CAN data frame in extended format (29-Bit identifier). Both frames contain an Identifier Extension Bit (IDE)[3], which is at low level for the standard frame and at high for the extended data frame. CAN controllers must be designed in a way that they check the IDE in order to distinguish between the two possible frame formats.

Both formats, Standard (11 bit message ID) and Extended (29 bit message ID), may co-exist on the same CAN bus. During bus arbitration the standard 11 bit message ID frame will always have higher priority than the extended 29 bit message ID frame with identical 11 bit base identifier and thus gain bus access.

The Extended Format has some trade-offs: The bus latency time is longer (minimum 20 bit-times), messages in extended format require more bandwidth (about 20 %), and the error detection performance is reduced (because the chosen polynomial for the 15-bit checksum is optimized for frame length up to 112 bits).

[3] The IDE in an 11-Bit standard frame is embedded in the Control Field.

2.1 CAN Characteristics

Everything that has to do with CAN is based on maximum reliability with the maximum possible performance in mind. After all, CAN was originally designed for automobiles, definitely a very demanding environment for microprocessors, not only in regards to required electrical robustness, but also due to high speed requirements for a serial communication system.

Many companies in the field of medical engineering chose CAN because they have to meet particularly strict safety requirements. Similar problems have been faced by manufacturers of other equipment with very high safety or reliability requirements, including robots, lifts and transportation systems.

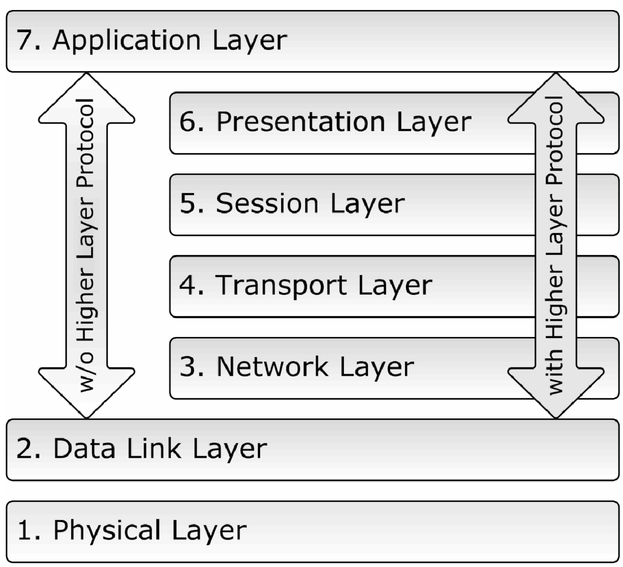
The CAN properties can be summarized as:

- Multi-Master priority based bus access
- Non-destructive contention-based arbitration
- Multicast message transfer by message acceptance filtering
- Remote data request
- Configuration flexibility
- System-wide data consistency
- Error detection and error signaling
- Automatic retransmission of messages that lost arbitration
- Automatic retransmission of messages that were destroyed by errors
- Distinction between temporary errors and permanent failures of nodes
- Autonomous deactivation of defective nodes

2.2 CAN Higher Layer Protocols

Even though extremely effective in automobiles and small applications, CAN alone is not suitable for machine automation, since its communication between devices is limited to only 8 bytes per message. As a consequence, higher layer protocols such as CANopen for machine control, DeviceNet for factory automation and J1939 for vehicles were designed to provide a real networking technology that support messages of unlimited length and allow a master/slave configuration.

In order to explain higher layer protocol we must refer to the ISO/OSI 7-Layer Reference Model[4] as shown in the picture below.



Picture 2.2.1 ISO/OSI 7-Layer Reference Model

The standard CAN implementation bypasses the connection between the Data Link Layer and the Application Layer. The layers above the Data Link Layer are covered by additional software, which represents per definition a higher layer protocol.

[4] For more information on the OSI Reference Model refer to:
CertificationZone.com OSI Reference Model Pocket Guide
by Howard C. Berkowitz - ISBN: 1890911143

2.3 J1939 Characteristics

J1939 is a higher-layer protocol based on Controller Area Network (CAN). It provides serial data communications between microprocessor systems (also called Electronic Control Units - ECU) in any kind of heavy duty vehicles.

Everything that has to do with CAN is based on maximum reliability with the maximum possible performance in mind, not only in regards to required electrical robustness, but also due to high speed requirements for a serial communication system.

While CAN itself is sufficiently suited for communication in a regular automobile or in small industrial applications, it comes with a few short-comings in regards to network management. In order to add these features CAN as the physical layer (the entire CAN protocol is on silicon) can be extended by additional software, the so-called higher layer protocols (such as J1939).

J1939 takes advantage of CAN features such as:

- Maximum reliability
- Excellent error detection & fault confinement
- Collision-free bus arbitration

Other than CAN, which supports up to 1 Mbit/sec, J1939 limits itself to 250 kbit/sec. CAN was designed to be as close to real-time applications as possible. This level of performance is not required for J1939.

It is troubling to learn that the SAE J1939 standard has no problem with compromising the CAN standard. The Network Management (SAE J1939/81), for instance, allows scenarios where two CAN nodes with the same message ID can access the bus. The result of such a situation is unpredictable. In addition, the SAE J1939 message format (as described in SAE J1939/21) does not take advantage of the message filtering as provided by all CAN controllers in the industry.

2.3.1 J1939 Quick Reference

- Higher-Layer Protocol using CAN as the physical layer
- Shielded twisted pair wire
- Max. network length of 40 meters (~120 ft.)
- Standard baud rate of 250 kBit/sec
- Max. 30 nodes (ECUs) in a network
- Max. 253 controller applications (CA) where one ECU can manage several CAs
- Peer-to-peer and broadcast communication
- Support for message length up to 1785 bytes
- Definition of Parameter Groups (Predefined vehicle parameters)
- Network Management[6] (includes address claiming procedure.

It must be emphasized that the maximum network length of 40 m (roughly 120 ft.), the baud rate of 250 kBit/sec and the maximum number of nodes (30) are self-inflicted restrictions by the SAE, most probably with the intention to keep everything on the extreme safe side and thus trying to prevent potential runtime problems.

In all consequence, the network length at 250 kBit/sec, according to ISO 11898, is 250 m (roughly 750 ft.).

There is no reason to believe that J1939 cannot be operated at the max. CAN baud rate of 1 MBit/sec. Naturally, the network length would drop, but the mere J1939 protocol features post no restriction in regards to the baud rate.

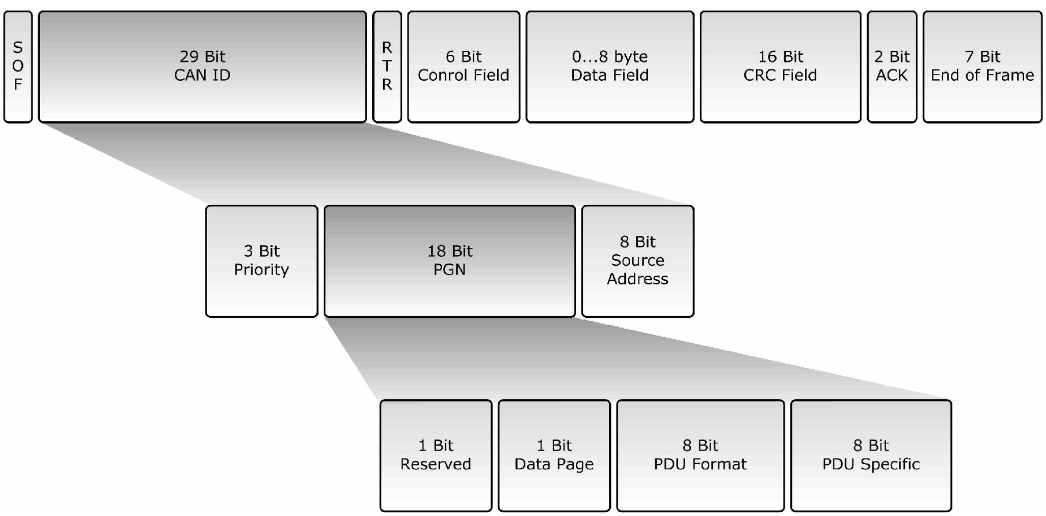
The J1939 protocol utilizes an 8 bit device (ECU) address, which would allow the operation of 256 nodes in the same network. It can only be assumed that the SAE was trying to keep the bus traffic on a low level by restricting the maximum number of nodes to 30. Elaborating comments on this restrictions may be embedded somewhere in the standard.

[6] The SAE J1939 Network Management does not include support for a Master/Slave configuration and it does not include node monitoring. These functions can nevertheless be implemented on an application level.

2.3.2 J1939 Message Format

CAN supports 11- and 29-Bit message identifiers. CAN is also designed in a way where the sending node is not concerned with which node(s) receive(s) the data. In turn a receiving node does not know/care who sent the data.

In contrast J1939 uses only the 29-Bit identifier (In fact, the CAN standard was extended from 11 to 29 bit per request by the SAE in order to support J1939). J1939 uses the identifier, among other features, to identify the source and, in some cases, the destination of data on the bus.



Picture 2.3.2.1 J1939 Message Format

As shown in the picture J1939 extends the use of the 29-Bit CAN identifier beyond the standard CAN message identification.

The CAN identifier is split into a priority field, a Parameter Group Number (PGN)[7] to identify the content of the data field, and the source address.

A message priority '0' indicates highest priority and a message priority of '7' indicates lowest priority. High priorities are usually assigned to time-critical data such as torque control

[7] For a peer-to-peer communication the PDU Specific field is used as destination address and, in this case, is not part of the PGN. See also chapter *Parameter Group Number Architecture*.

2.4 Other J1939 Based Protocols

Per definition, SAE J1939 provides serial data communications between microprocessor systems (also called Electronic Control Units - ECU) in any kind of heavy duty vehicles. The messages exchanged between these units can be data such as vehicle road speed, torque control message from the transmission to the engine, oil temperature, and many more.

SAE J1939 and its companion documents have quickly become the accepted industry standard of choice for off-highway machines. It was all too natural that organizations and manufacturers in the agricultural, military and marine industries, rather than re-inventing the wheel, adopted the proven combination of physical layer, Controller Area Network (CAN), and J1939 as the higher layer protocol for vehicles. However, it is in the specific nature of agricultural and military as well as marine applications that slight modifications, including a name change, were necessary.

These “new” protocols are[9]:

- ISO 11783 (a.k.a ISOBUS) – Agricultural Industry
- MilCAN – Military Applications
- NMEA 2000 – Marine Applications

[9] Note by the author: While information on NMEA 2000 and especially MilCAN is easily available, documentation of the ISO 11783 standard is “protected” by excessive price tags. One cannot deny the standardization organization’s objective to produce revenues, but one also needs to seriously question their marketing expertise.

Chapter
3

The J1939 Standards Collection

The “SAE Truck and Bus Control & Communications Network Standards Manual – 2007 Edition” is a colossal work of roughly 1600 pages where about 1000 pages refer to data such as Parameter Group Assignments, Address and Identify Assignments, Parameter Group Numbers, and more.

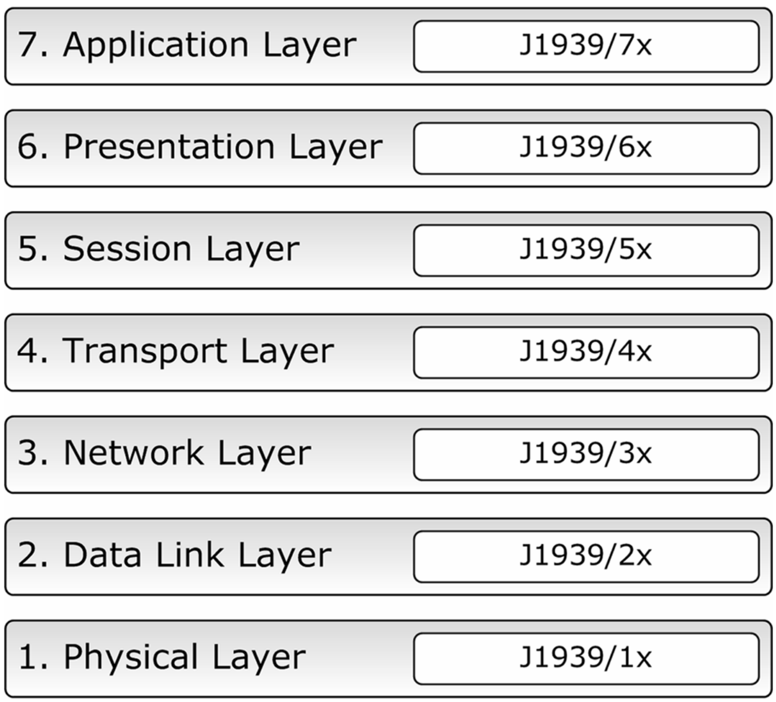
While it combines all standards in one work, it seems that the individual standard descriptions were developed independently from each other. There is no noticeable system to standardize the standards. Most documents provide a table of contents, some of them do not. Some standards repeat information from others or refer to others; many references actually lead nowhere. The page numbering is also inconsistent with the table of contents that exist.

It is apparent throughout the entire J1939 standards collection that the writers/creators of the standard indulged themselves with sophisticated, but nevertheless irritating terms and language.

The J1939 Standards Collection was created for mere documentation purposes without the necessary passion for the subject and without reader-friendliness or educational value in mind. Especially, unnecessary abbreviations (and their abbreviations) are used to an excessive degree and make reading these documents a demanding task.

3.1 ISO/OSI 7-Layer Reference Model

The J1939 Standards Collection was designed to follow the ISO/OSI 7-Layer Reference Model as far as necessary[13]. Each layer is addressed by a corresponding document.



Picture 3.1.1 SAE J1939 Standards

The Open Systems Interconnection Basic Reference Model or OSI Model for short is a layered, abstract description for communications and computer network protocol design.

[13] For more information on the OSI Reference Model refer to:
CertificationZone.com OSI Reference Model Pocket Guide
by Howard C. Berkowitz - ISBN: 1890911143

Layer	Title	Description
7	Application	Supports application and end-user processes.
6	Presentation	Provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa.
5	Session	Establishes, manages and terminates connections between applications.
4	Transport	Provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. Ensures complete data transfer.
3	Network	Provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node.
2	Data Link	Data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers, the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.
1	Physical	Conveys the bit stream (electrical impulse, light or radio signal) through the network at the electrical and mechanical level.

note

In a CAN (Controller Area Network) network both layers, Data Link and Physical Layer, are represented by the actual CAN controller. As a matter of fact, the actual CAN protocol, i.e. the entire data communication management including bus arbitration, error detection and fault confinement, etc., etc., is implemented into silicon. CAN controllers know per default what to do and how to do it.

Chapter

4

J1939 Message Format

The main document describing the J1939 message format is SAE J1939/21 – Data Link Layer. J1939/21 defines the use of the CAN data frame (29-bit identifier, Parameter Group Numbers – PGN, etc.) and the transport protocol functions, i.e. a definition of how messages longer than the standard CAN data length (8 bytes) are transmitted in a J1939 bus network[22].

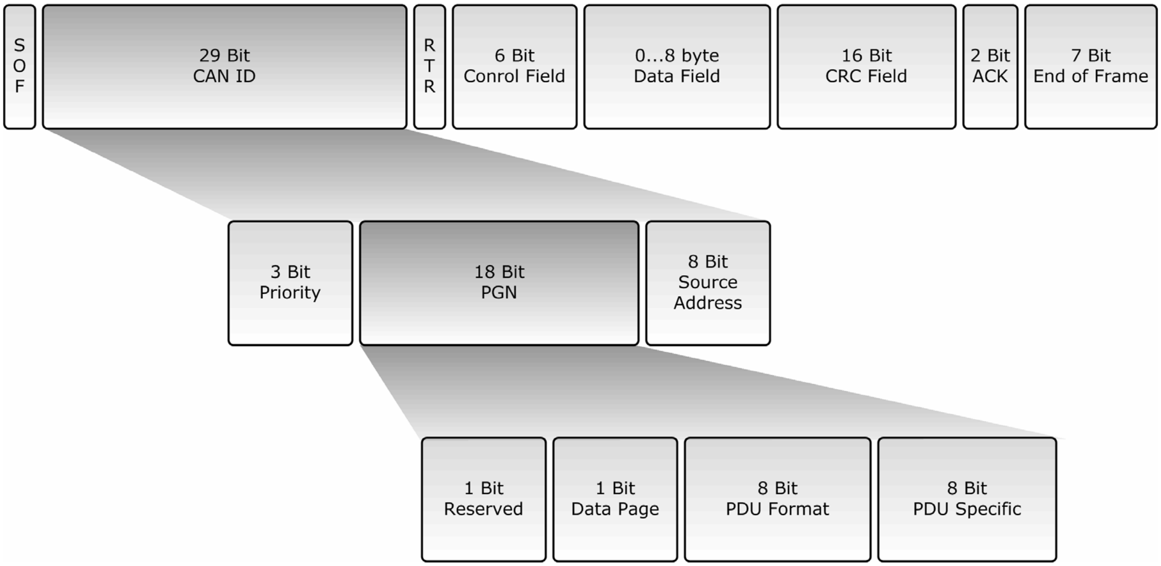
Without further ado, the J1939/21 document, assuming that the reader has already studied all applicable publications and knows what Parameter Groups are[23], jumps immediately into the detailed architecture of Parameter Group Numbers. It is like explaining the function of an automobile by starting with the details of the gas injection system. The document is poorly written and lacks any structure.

For instance, the mere fact that some J1939 messages may be longer than the standard CAN message of 8 bytes is mentioned in numerous chapters without offering any substantial details on how the messages are packetized. In the same spirit, topics like, for instance, PDU Format are explained repeatedly, spanning over several chapters.

Most irritating, however, is the fact that the SAE engineers who created the standard are not familiar with the unit of time, “ms” or “msec” (milli-seconds). Instead they used mS, which is officially milli-Siemens (electric conductance, equal to inverse Ohm - Ω).

[22] This statement is based on SAE claims. In all consequence the document does not offer any detailed description that would enable the reader to understand the packaging of messages longer than 8 bytes.
[23] As a matter of fact, the document does not provide any explanation of what Parameter Groups are.

The following picture demonstrates the use of the 29-Bit message ID[27].



Picture 4.2.2 J1939 Message Format

The parameters embedded in the 29-Bit message identifier are divided into three sections, the priority field, PGN (Parameter Group Number)[28] and the 8 bit source address.

Priority

The first three bits in the identifier represent the priority during the arbitration process, thus providing eight priority levels. In compliance with the CAN standard a value of 0 (000) has the highest priority; a value of 8 (111) has the lowest priority.

[27] For further detailed technical information on the CAN message frame architecture please refer to "A Comprehensible Guide to Controller Area Network" by the same author and publisher. http://www.copperhillmedia.com.
[28] For a peer-to-peer communication the PDU Specific field is used as destination address and, in this case, is not part of the PGN. Also, the actual PGN is internally extended to 24 bits. See also chapter *Parameter Group Numbers*.

4.3 Communication Methods

SAE J1939 provides three communication methods, each serving a specific purpose.

1. Destination Specific Communications:

Destination specific communications use PDU1 (PF values 0 to 239), but also the global destination address 255. There are cases where this method will require the utilization of destination specific Parameter Group Numbers, for instance, in the case of more than one engine. A torque message, for example, must be sent only to the desired engine and not to both.[30]

2. Broadcast Communications

Broadcast communications use PDU2 (PF values 240 to 255) and, as the name implies, they can include:

- Sending a message from a single or multiple sources to a single destination.
- Sending a message from a single or multiple sources to multiple destinations.

3. Proprietary Communications

Proprietary communications use either PDU1 or PDU2 and, as the name implies, they are useful in case where standard communications are not practical[31].

The use of PDU1 or PDU2 indicates that there may be:

- **Broadcast Proprietary Communications**

and

- **Destination Specific Proprietary Communications**

A Parameter Group Number (PGN) has been assigned for both proprietary communication types.

---

[30] The SAE J1939 document uses the expression "…the message must be directed to one or another specific destination and not both."

[31] The SAE J1939 document uses the expression "Where it is important to communicate proprietary information." - yet another example of pointless redundancy.

The reception and processing of received messages are explained in detail in SAE J1939/21 (Data Link Layer) and J1939/7x (Application Layer).

In general a received message is handled according to the communication method[32]:

- **Destination Specific Request or Command**

  Each receiving ECU must determine whether the incoming destination address matches its own address and if yes, it must process the message and respond accordingly[33].

- **Global Request**

  Each ECU in the network, even the sender of the request, must process the message and respond if the requested data is available.

- **Broadcast**

  Each ECU must determine individually whether or not the message is relevant.

4.4 Parameter Group Numbers

SAE J1939 is a very ingeniously designed protocol that takes a resourceful advantage of the CAN 29-Bit message identifier. Rather than relying on a myriad of protocol functions, SAE J1939 uses predefined parameter tables, which keeps the actual protocol on a comprehensible level. However, these parameter tables (Parameter Groups) are also the biggest stumbling block when it comes to implementing the protocol into an embedded solution (ECU).

Parameters groups are, for instance, engine temperature which includes coolant temperature, fuel temperature, oil temperature, etc. Parameter Groups and their numbers (PGN) are listed in SAE J1939 (roughly 300 pages) and defined in SAE J1939/71, a document containing

[32] The SAE J1939 document uses the sentence "Several general observations can be made however regarding received messages." which not only lacks the proper grammar, but it also raises the question whether SAE J1939 is a research project (hence "observations") or a Standard where network specifics are outlined.

[33] SAE J1939 uses the vague wording "…provide some type of acknowledgement."

Note the high-lighted portion in the J1939 Frame Architecture that points to PDU Format Field (PF). The PF is divided into two sections, separated by the CAN SRR and IDE bit. The SRR and IDE bits are entirely defined by the CAN standard 2.0B and thus are not described or modified by the SAE J1939 standard. For the same reason, some documentation may refer to a PF length of 10 bits.

4.4.5 Parameter Group Number Range

With the definition of PDU Format (PF) and PDU Specific (PS) – as shown below - J1939 supports a total of 8672 Parameter Group numbers.

	PDU Format	**PDU Specific**	**Communication Mode**
PDU1 Format	0 – 239 0_{hex} - EF_{hex}	Destination Address	Peer-to-Peer
PDU2 Format	240 – 255 $F0_{hex}$ - FF_{hex}	Group Extension	Broadcasting

Table 4.4.5.1 PDU Format and PDU Specific

The Parameter Group Number range is divided into two sections:

1. Specific PGNs for peer-to-peer communication (PDU1 Format)
   Range: 00_{hex} - EF_{hex} (not including PDU Specific)
   Number of PGNs: 240

2. Generic PGNs for message broadcasting (PDU2 Format)
   Range: $F000_{hex}$ – $FFFF_{hex}$ (including PDU Specific)
   Number of PGNs: 4096

Considering the Data Page (DP) bit, the total number of PGNs is (240 + 4096) * 2 = 8672.

As a reminder:

The DP bit works as a page selector for the following PDU (Protocol Data Unit) Format (PF) field. Currently this bit is at 0, pointing to page 0, which in turn points to all currently defined messages. Page 1 will be used to provide extended capacity for the future, i.e. as soon as page 0 has reached its capacity.

The following shows a Parameter Group Number map:

DP	PGN Range (hex)	Number of PGNs	SAE or Manufacturer Assigned	Communication
0	000000 – 00EE00	239	SAE	PDU1 = Peer-to-Peer
0	00EF00	1	MF	PDU1 = Peer-to-Peer
0	00F000 – 00FEFF	3840	SAE	PDU2 = Broadcast
0	00FF00 – 00FFFF	256	MF	PDU2 = Broadcast
1	010000 – 01EE00	239	SAE	PDU1 = Peer-to-Peer
1	01EF00	1	MF	PDU1 = Peer-to-Peer
1	01F000 – 01FEFF	3840	SAE	PDU2 = Broadcast
1	01FF00 – 01FFFF	256	MF	PDU2 = Broadcast

Table 4.4.5.1 Parameter Group Number Range

The current range of Parameter Group Numbers as defined in SAE J1939/71 is from PGN 0 (Torque/Speed Control) to PGN 65279 (Water in Fuel Indicator). This range is not a real representation of the total number of PGNs, since there a gaps between PGN definitions. The same is true for SPNs, who range from SPN 16 (Engine Fuel Filter) to SPN 4096 (XBR Urgency).

The Request message type is associated with a specific PGN as described below.

Parameter Group Name	**Request**
Parameter Group Number	59904 (00EA00$_{hex}$)
Definition	Requests a Parameter Group from a single device or all devices in the network.
Transmission Rate	User defined (no more 2 to 3 times a second is recommended)
Data Length	3 bytes (CAN DLC = 3)
Extended Data Page (R)	0
Data Page	0
PDU Format	234
PDU Specific	Destination Address (Global or Specific – See following note)
Default Priority	6
Data Description	Byte 1, 2, 3 = Requested Parameter Group Number

While the destination address is defined to address a specific node (ECU) in the network, it can also be used to address all ECUs at the same time. A destination address (DA) of 255 is called a Global Destination Address[42] and it requires all nodes to listen and, if required, to respond. The SAE J1939/21 standard defines the Global Destination Address in one short sentence that can be easily overlooked.

The following table demonstrates the use of fields of a Request message type.

Message Type	**PGN**	**PS (DA)**	**SA**	**Data 1**	**Data 2**	**Data 3**
Global Request	59904	255	Requester	PGN (LSB)	PGN	PGN (MSB)
Specific Request	59904	Responder	Requester	PGN (LSB)	PGN	PGN (MSB)

Table 4.4.10.2.1 Request Message Type - Use of Fields

[42] No acronym offered by SAE.

Parameter Group Name	**Proprietary A2**
Parameter Group Number	126720 ($01EF00_{hex}$)
Definition	Proprietary PG using the PDU1 Format for Peer-to-Peer communication.
Transmission Rate	Manufacturer Specific
Data Length	0 – 1785 bytes (multi-packet supported)
Extended Data Page (R)	0
Data Page	1
PDU Format	239
PDU Specific	8 bit Destination Address – Manufacturer Assigned
Default Priority	6
Data Description	Manufacturer Specific

Parameter Group Name	**Proprietary B**
Parameter Group Number	65280 - 65535 ($00FF00_{hex}$ – $00FFFF_{hex}$)
Definition	Proprietary PG using the PDU2 Format for Broadcast communication.
Transmission Rate	Manufacturer Specific
Data Length	0 – 1785 bytes (multi-packet supported)
Extended Data Page (R)	0
Data Page	0
PDU Format	255
PDU Specific	Group Extension – Manufacturer Assigned
Default Priority	6
Data Description	Manufacturer Specific

4.5 Transport Protocol Functions

Even though extremely effective in passenger cars and small industrial applications, CAN alone was not suitable to meet the requirements of truck and bus communications, especially since its communication between devices is limited to only 8 bytes per message. However, it is possible to extend the size of a CAN message by implementing additional software, i.e. so-called higher layer protocols. J1939 is such a higher layer protocol and it supports up to 1785 bytes per message.

In order to support a size of more than 8 bytes the message needs to be packaged into a sequence of 8 byte size messages. Consequently, the receiver of such a multi-packet message must re-assemble the data. Such functions are defined as Transport Protocol (TP) Functions and they are also described in SAE J1939/21. The two major parts of the TP Functions are Message Packaging[45] & Reassembly and Connection Management. In addition, the Transport Protocol Functions handle flow control and handshaking features for destination specific transmissions.

The SAE J1939/21 standard fails to mention that the multi-packet messages are only supported in PDU1 format (Peer-to-Peer communication). In order to broadcast messages they must be addressed to the global destination address (= 255).

4.5.1 Message Packaging and Reassembly

Certain parameter groups may require more than the 8 data bytes supported by the CAN standard. The SAE J1939 standard[46], namely the Transport Protocol Function, supports message lengths up 1785 bytes. In case a program group requires more than 8 data bytes

[45] The SAE Standard uses the term "Packetization", which is not an official English word.
[46] The SAE J1939/21 standard "wastes" less than one page on the subject of message "packetization" and re-assembly.

In order to broadcast a multi-packet message a node must first send a *Broadcast Announce Message* (BAM). A BAM message contains the following components:

- Parameter Group Number of the multi-packet message
- Size of the multi-packet message
- Number of packages

The BAM message allows all receiving nodes (= all nodes interested in the message) to prepare for the reception by allocating the appropriate amount of resources (memory).

The *Broadcast Announce Message* (BAM) is embedded in the Transport Protocol – Connection Management (TP.CM) PGN 60416 and the actual data transfer is handled by using the Data Transfer PGN 60160[48].

Parameter Group Name	**Transport Protocol – Connection Management (TP.CM)**
Parameter Group Number	60416 (00EC00$_{hex}$)
Definition	Used for Communication Management flow-control (e.g. Broadcast Announce Message).
Transmission Rate	According to the Parameter Group Number to be transferred
Data Length	8 bytes
Extended Data Page (R)	0
Data Page	0
PDU Format	236
PDU Specific	Destination Address (= 255 for broadcast)
Default Priority	7
Data Description	(For Broadcast Announce Message only)
Byte	1 - Control Byte = 32
	2,3 – Message Size (Number of bytes)
	4 – Total number of packages
	5 – Reserved (should be filled with FF$_{hex}$)
	6-8 – Parameter Group Number of the multi-packet message (6=LSB, 8=MSB)

[48] The SAE J1939/21 Standard – Chapter 5.10.2.1 Multipacket Broadcast – does not offer any reference regarding the exact structure of the BAM, the BAM PGN, or the Data Transfer PGN 60160.

Parameter Group Name	**Transport Protocol – Data Transfer (TP.DT)**
Parameter Group Number	60160 ($00EB00_{hex}$)
Definition	Data Transfer of Multi-Packet Messages
Transmission Rate	According to the Parameter Group Number to be transferred
Data Length	8 bytes
Extended Data Page (R)	0
Data Page	0
PDU Format	235
PDU Specific	Destination Address
Default Priority	7
Data Description	
Byte	1 – Sequence Number (1 to 255)
	2-8 - Data

The last packet of a multi-packet PGN may require less than eight data bytes. All unused data bytes in the last package are being set to FF_{hex}.

The transport of Multi-Packet Broadcast messages is not regulated by any flow-control functions and thus it is necessary to define timing requirements between the sending of a *Broadcast Announce Message* (BAM) and the Data Transfer PGN. The following picture demonstrates the message sequence and timing requirements for a broadcasted multi-packet message[49].

[49] One would expect that engineers, regardless of their special expertise, are familiar with the unit of time, “ms” or “msec” (milli-seconds). Instead the SAE J1939/21 standard uses mS, which is officially milli-Siemens (electric conductance, equal to inverse Ohm - Ω).

Chapter 5

J1939 Network Management

The SAE J1939 Network Management is defined in SAE J1939/81. This document is in far better condition than, for instance, SAE J1939/21. After all, there is a visible structure. Still, the authors use terms right from the beginning that are then being explained in later chapters. And, yet again, the authors indulge themselves in acronyms, especially CA (Controller Application), which is being excessively used throughout the document.

Network Management under J1939 is primarily represented by the Address Claiming Process. While other higher layer protocols based on Controller Area Network (CAN) do not support dynamic node address assignments per default, the SAE J1939 standard provides this ingeniously designed feature to uniquely identify ECUs and their primary function.

SAE J1939/81 prefers the use of CA (Controller Application) rather than ECU (Electronic Control Unit). In all consequence one ECU can run multiple CAs. Each Controller Application will have one address and associated NAME (See following chapters). The following chapters will continue using the term ECU, which is a synonym for CA[54].

[54] The SAE J1939 document provides an overview of the address claiming procedure and this document, while referring to SAE J1939/81 for more details, uses only the term ECU.

5.1 Address Claiming Procedure Overview

While other higher layer protocols based on CAN do not support dynamic node address assignments per default, the SAE J1939 standard provides yet another ingeniously designed feature to uniquely identify ECUs and their primary function.

The CAN standard in itself does not support node (ECU) addresses, only message IDs, where one node may manage multiple messages. However, the message ID must be hard-coded in the application program. Also, in a standard CANopen network the node address is usually hard-wired or mechanically adjustable (e.g. per dip switch).

Each ECU in a J1939 vehicle network must hold at least one NAME and one address for identification purposes. Single electronic units are allowed, however, to control multiple names and addresses.

The 8-bit ECU address defines the source or destination for messages.

The ECU NAME includes an indication of the ECU's main function performed at the ECU's address. A function instance indicator is added in cases where multiple ECUs with the same main function share the same network.

The J1939 standard allows up to 253 ECUs with the same function to share the same network, where each ECU is identified by their individual address and NAME[55].

[55] This statement was derived from the SAE J1939 standard and is somewhat misleading. J1939 allows a maximum of 30 ECUs, but a maximum of 253 Controller Applications.

SAE J1939 defines a 64 bit NAME, as shown in the picture below, to uniquely identify each ECU in a network.

Arbitrary Address Capable	Industry Group	Vehicle System Instance	Vehical System	Reserved	Function	Function Instance	ECU Instance	Manufacturer Code	Identity Number
1 bit	3 bit	4 bit	7 bit	1 bit	8 bit	5 bit	3 bit	11 bit	21 bit

Picture 5.1.1 J1939 NAME Fields

While the 64 bit NAME is certainly appropriate to uniquely identify nodes (ECUs) and their function in a J1939 network, it will nevertheless necessitate unreasonable resources to maintain standard communications.

In order to provide a more efficient solution, the SAE J1939 Standard defines an address claim procedure[56], where each ECU utilizes an 8 bit address to identify the source of a message or to access (destination address) another ECU in the network. The address claim procedure is designed to assign addresses to ECUs right after the network has been initialized and thus assuring that the assigned address is unique to the ECU. For instance, an engine may be assigned the address 0 while another engine is present, which will be assigned another address (e.g. 1) and instance.

ECUs designed to accept destination specific commands may require multiple addresses, each with their corresponding NAME, in order to distinguish the required action. For instance, the torque from the engine as commanded by the transmission must be separated from the torque commanded by the brake.

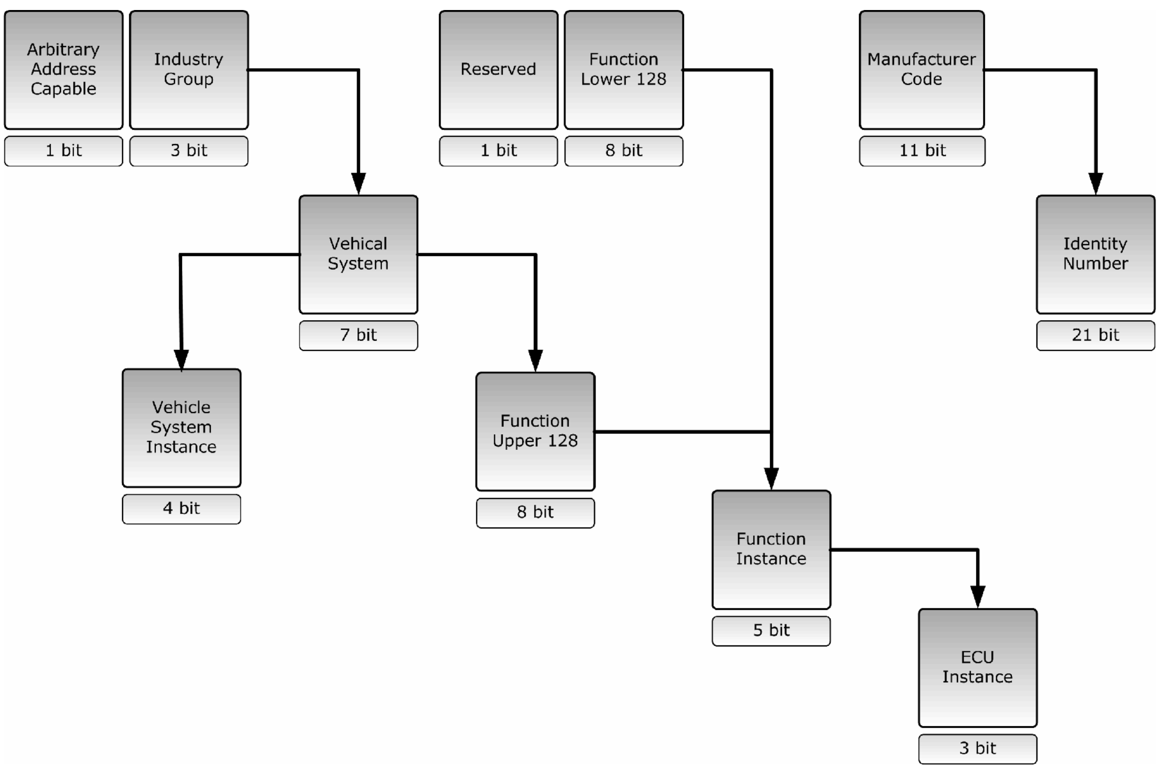
56 The address claim procedure is defined through the SAE J1939/81 Standard and elaborated on in SAE J1939/01.

5.2.1.10 NAME Field: Identity Number

The 21 bit Identity Number is assigned by the manufacturer of the ECU and should be used to guarantee unique NAMEs within a product line. The manufacturer is also allowed to add further information to the Identity Number such as, for instance, serial number, date of manufacture, etc.

5.2.1.11 NAME Field Dependencies

The following picture demonstrates the dependencies inside the NAME field according to the definition of all NAME fields.



Picture 5.2.11.1 Name Field Dependencies

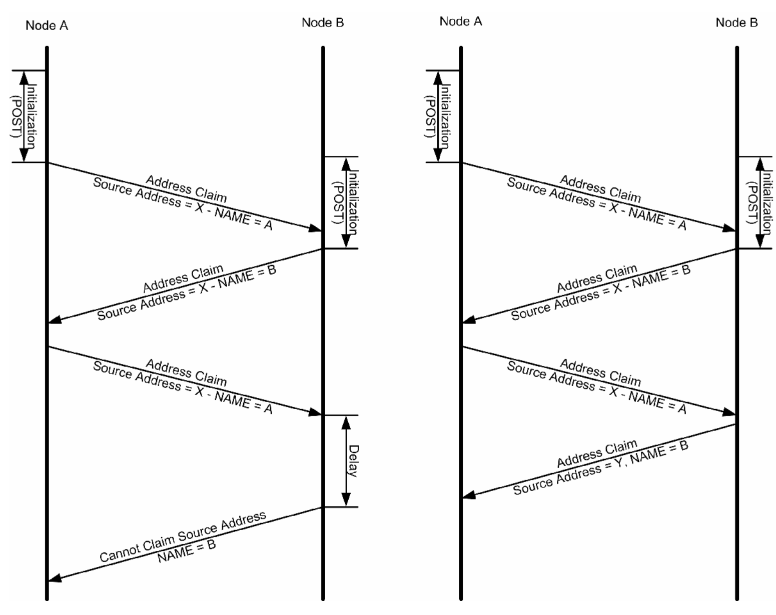
immediately, while other ECUs should not begin until a time of 250 ms after claiming an address. This allows competing claims to be resolved before the address is being used.

In the event that two ECUs attempt to claim the same address, the ECU with the lowest NAME value will succeed and use the address as claimed. The remaining ECUs must claim a different address by sending another *Address Claimed* message containing a different address or send a *Cannot Claim Address* message.

The destination address for an address claim is always the global address (255) in order to address all nodes in the network.

A node, that has not yet claimed an address, must use the NULL address (254) as the source address when sending a *Request for Address Claimed* message.

The following picture demonstrates two possible address claim scenarios.



Picture 5.3.1.1 Address Claim Procedure

SAE J1939/81 explains that, during this process, both nodes would go eventually into BUS OFF mode. However, this statement is not backed up by either the CAN standard (which categorically does not allow two messages with the same ID) or empirical tests.

Tests (not accomplished by the SAE) have shown that the two competing nodes will go into Error Passive mode and both nodes will eventually return to the regular Error Passive mode. However, the time until both nodes return to regular activities is unpredictable and so are the consequences for the application.

5.3.2 Address Management Messages

The network management messages have the same characteristics as all other J1939 messages. The messages are:

Message	**PGN**	**PF**	**PS**	**SA**	**Data Length**	**Data**
Request for Address Claimed	59904	234	DA	SA[1)]	3 bytes	PGN 60928
Address Claimed	60928	238	255	SA	8 bytes	NAME
Cannot Claim Source Address	60928	238	255	254	8 bytes	NAME
Commanded Address	65240	254	216	SA	9[2)]	NAME, new SA

Table 5.3.2.1 Address Management Messages

1) In case no address has been claimed as of yet the source address could be set to 254.
2) The commanded address, since it is longer than 8 bytes, is sent using the Transport Protocol as described in chapter *Transport Protocol*.

5.3.2.1 Request for Address Claimed

The Request for Address Claimed message (PGN 59904) is identical to the Request message type as described in SAE J1939/21 and chapter *Parameter Group Numbers* in this book.

Parameter Group Name	**Request**
Parameter Group Number	59904 ($00EA00_{hex}$)
Definition	Requests a Parameter Group from a single device or all devices in the network.
Transmission Rate	User defined (no more 2 to 3 times a second is recommended)
Data Length	3 bytes (CAN DLC = 3)
Extended Data Page (R)	0
Data Page	0
PDU Format	234
PDU Specific	Destination Address (Global or Peer-to-Peer)
Default Priority	6
Data Description	Requested Parameter Group Number = PGN 60928

The *Request for Address Claimed* message is used to request the sending of an *Address Claimed* message from either a particular node in the network or from all nodes (use of global destination address = 255). The Address Claimed message (as described in the following chapter) will provide the requested information, i.e. address and NAME of the responding node(s).

The purpose of sending such a request may be for several reasons, for instance:

- A node is checking whether or not it can claim a certain address.
- A node is checking for the existence of another node (Controller Application) with a certain function.

The response to a Request for Address Claimed message can be multiple:

- Any addressed node that has already claimed an address will respond with an *Address Claimed* message.
- Any addressed node that was unable to claim an address will respond with a *Cannot Claim Address* message.
- Any addressed node that has not yet claimed an address should do so by responding with their own *Address Claimed* message where the source address is set to NULL (254).
- A node sending the Request for Address Claimed message should respond to its own request in case the global destination address (255) was used.

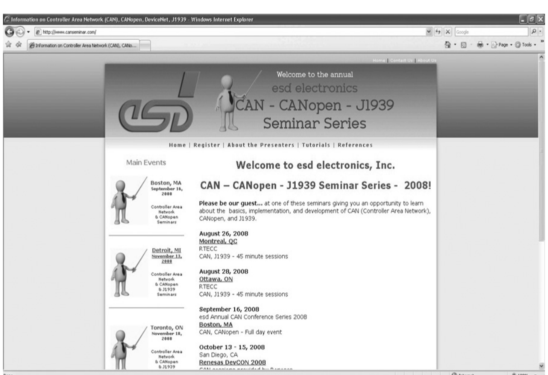
Appendix A – Web Site References

http://www.cannewsletter.com

CANNewsletter.com provides vast information on all aspects of Controller Area Network including CANopen and J1939. The web site contains all kinds of information beyond the standards including many articles and links to CAN, CANopen and J1939 seminars and literature.

http://www.j1939forum.com

J1939Forum.com is the Online meeting place where to find additional information on SAE J1939 and get help with issues related to SAE J1939.



http://www.canseminar.com

Check out this web site, CANSeminar.com, for seminars on CAN, CANopen, or J1939 in your area.

Appendix B – Literature References

1. **A Comprehensible Guide to Controller Area Network**
   by Wilfried Voss
   Published 2005 by Copperhill Technologies Corporation
   http://www.copperhillmedia.com

2. **Embedded Networking with CAN and CANopen**
   by Olaf Pfeiffer, Andrew Ayre and Christian Keydel
   Republished by Copperhill Technologies Corporation
   http://www.copperhillmedia.com

3. **SAE Truck and Bus Control & Communications Network Standards Manual**
   2007 Edition
   HS-1939
   SAE International

4. **Application of J1939 Networks in Agricultural Equipment**
   by Marvin L. Stone and Mark Zachos
   Oklahoma State University - Dearborn Group
   Stillwater, Oklahoma - Farmington Hills, Michigan

5. **J1939-based application profiles**
   by Holger Zeltwanger - CiA
   Reference found at: http://www.can-expo.ru/files/J1939.pdf

6. **High Speed Networking in Construction and Agricultural Equipment**
   by Marvin L. Stone
   Department of Biosystems and Agricultural Engineering
   Oklahoma State University

7. **CAN und J1939 unter extremen Einsatzbedingungen**
   Elektronik automotive 5.2006

8. **A Framework for Developing SAE J1939 Devices**
   by Joachim Stolberg, IXXAT Automation
   March 21, 2007

9. **Introduction to J1939**
   Application Note AN-ION-1-3100
   by Markus Junger
   2004 - Vector Informatik GmbH

10. **Dynamic Address Configuration in SAE J1939**
    by Marvin L. Stone
    Department of Biosystems and Agricultural Engineering

Oklahoma State University

11. **On calculating guaranteed message response times on the SAE J1939 bus**
by Roger Johansson and Jan Torin
Chalmers Lindholmen University College
Goeteborg, Sweden 2002

12. **J1939 C Library for CAN-Enabled PICmicro® Microcontrollers**
by Kim Otten, Kim Otten Software Consulting
and Caio Gübel, Microchip Technology Inc.
Source: http://ww1.microchip.com/downloads/en/AppNotes/00930a.pdf

13. **NMEA 2000 Network Design**
Jack Rabbit Marine
Source: http://www.jackrabbitmarine.com/files/NMEA2K_Network_Design_v2.pdf

14. **Q&A - What is SAE J1939?**
Axiomatic Global Electronic Solutions
July 6, 2006 – Application Note

Appendix C – Picture Index

Appendix D - Abbreviations

ABS	Antilock Braking System
ACK	Acknowledgement
BAM	Broadcast Announce Message
CA	Controller Application
CAN	Controller Area Network
CM	Connection Management
CRC	Cyclic Redundancy Check
CTS	Clear to Send
DA	Destination Address
DLC	Data Length Code
DP	Data Page
DT	Data Transfer
ECU	Electronic Control Unit
EDP	Extended Data Page
EOF	End of Frame
GE	Group Extension
ID	Identifier
IDE	Identifier Extension Bit
LLC	Logical Link Control
LSB	Least Significant Bit or Byte
MSB	Most Significant Bit or Byte
NA	Not Allowed
NACK	Negative Acknowledgement
P	Priority
PDU	Protocol Data Unit
PF	PDU Format
PG	Parameter Group
PGN	Parameter Group Number
PS	PDU Specific
RTR	Remote Transmission Request
RTS	Request to Send
SA	Source Address
SAE	Society of Automotive Engineers
SOF	Start of Frame
SPN	Suspect Parameter Number
SRR	Substitute Remote Request
TP	Transport Protocol

Index