

# CAN: Controller Area Network Introduction and Primer

by Robert Boys rboys@dgtech.com

Copyright © 2004 Dearborn Group, Inc., September, 2004 Version 3.1

## Introduction

CAN was developed for the automotive market to reduce the weight and cost of wiring harnesses and add additional capabilities. It is also used in factory automation, medical, marine, military and anywhere a simple yet robust network is needed.

### What Exactly is CAN ?

CAN, by itself, is not necessarily a complete network system. It consists of only the physical layer (the two wires), the priority scheme (highest priority message always gets through first) and some error detection and handling circuitry. This allows simple messages of from zero to eight bytes to be passed on the system.

CAN, like most modern networks, is serial based. This means that the information travels along the network one bit at a time. A CAN network needs from one to two lines depending on the design. Parallel networks usually require more than 8 wires plus several handshaking lines to facilitate the data transfer.

Most network systems using CAN will employ a higher level protocol such J1939, CANopen or a proprietary scheme to create and process messages over the basic CAN network.

## A Vehicle Network and its Components

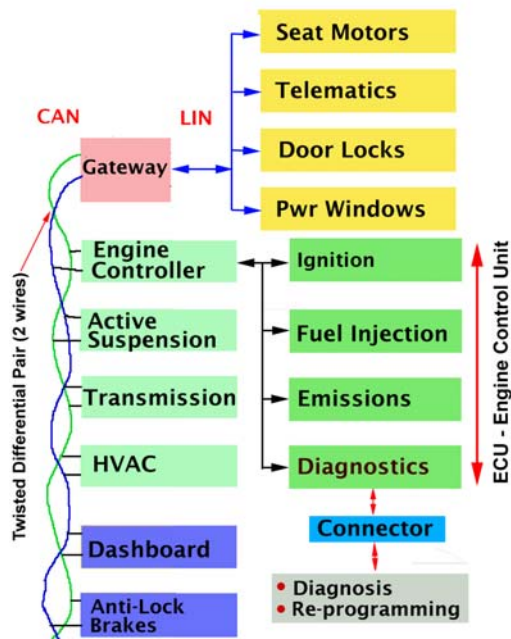


Figure 1: CAN & LIN Automotive Network

Figure 1 has two buses connected together (CAN and LIN) by a gateway that will be in one of the ECMs (Electronic Control Module), depending on the design. Often the gateway is in the ECU (Engine Control Unit). The ECU contains vital engine

control functions and is often the clearinghouse for vehicle diagnostics such as OBDII and/or a proprietary system. This is a representative system and there are many other variations.

### Engine Control Unit (ECU)

The Engine Control Unit is usually the most important module in a vehicle and is central to the entire system. The ECU is also another ECM. An ECU's possible components are shown in the four boxes on the right side of Figure 1.

The general term for vehicle modules in this article is ECM (Electronic Control Module). Each ECM can exchange information with other modules to accomplish certain tasks. For instance, the transmission module will supply the speedometer with the current speed as well as optionally to the radio to modify the volume: this is transmitted over the CAN bus as general network traffic. Figure 2 is an example of an ECM.



Figure 2: Toyota Airbag ECM

### Communication Paths

CAN messages are broadcasted to all ECMs. There is generally no way of knowing where a particular message came from or where its intended destination ECM is.

This makes it easy to add another or different ECM to modify vehicle features. The new ECM is simply plugged into the two wire CAN bus and all other modules will see its messages.

### Electronic Control Module Modes

A module can be in two general modes:

- 1) **Standard:** The modules and vehicle operate normally and network traffic consists of ordinary data needed for the operation of the vehicle. The translation of this data is usually, but not always, proprietary to the manufacturer.
- 2) **Diagnostic:** The module is put into the diagnostic mode by a scan tool that will make queries of the ECM. The ECM will return information to the scan tool concerning problems with the vehicle along with other data deemed important.

## Diagnostics - Public and Proprietary - Overview

### Public Diagnostic Protocols (OBDII)

Diagnostics can use a government-mandated standard such as OBDII (On Board Diagnostics Version 2) as specified by CARB (California Air Resources Board) and the EPA (Environmental Protection Agency) for automobiles and light trucks for model year 1996 and on. The European Community has also implemented OBDII as part of its Euro Stage III Directive in 2000.

These are normally concerned with vehicle emissions and the method and information needed to access and decipher them is public knowledge. OBDII legislation requires trouble codes to be reported with a generic scan tool such as the one in Figure 3.

Figure 3 also shows the special connector found under the steering column on all 1996 and later cars. This connector is specified by SAE J1962 and is equivalent to ISO15031-3. Normal vehicle network messages are also available on this connector. Depending on the vehicle, various protocols can be found on this connector including CAN, ISO9141, KWP2000 and J1850. Most vehicle makers are migrating to the CAN bus.

Figure 3: Generic Scan Tool



OBDII specifies a vehicle warning light called MIL (Malfunction Indicator Lamp). This light displays a message similar to “Service Engine Soon” to the vehicle operator. Pending errors will not illuminate the MIL but are available to the scan tool. If a pending error exists for a predetermined time duration, it will be turned into a Diagnostic Trouble Codes (DTC) and the MIL will be turned on indicating vehicle service is required.

### Proprietary Diagnostic Protocols

Diagnostics not prescribed by the government are usually proprietary to the vehicle or subsystems manufacturer. Each vehicle manufacturer or supplier has its own proprietary diagnosis system and this information is usually kept confidential.

The raw CAN messages can be seen with an oscilloscope, some scan tools or a CAN analyzer such as the Dearborn Group’s Gryphon but the meaning of the messages can be difficult to decipher without a translation list from the manufacturer.

**Dearborn Group, Inc.** [www.dgtech.com](http://www.dgtech.com)  
Phone (248) 488-2080

### CAN Physical layer

CAN usually consists of two wires - arranged as a differential pair. The data is impressed on the CAN bus by making the voltage between these two wires different: either 2 volts or zero volts. This differential arrangement provides excellent immunity from outside electrical interference and substantially reduces emitted RF radiation eliminating the need for shielded cables.

### LIN Physical layer

The LIN bus is less expensive than the CAN bus and is used for less demanding applications such as door locks and windows. Many of these applications do not need the power and speed of CAN. The LIN bus is a 12 volt single wire UART serial design.

### Other Serial Networks

There are several low cost networks available including LIN, ISO9141, KWP2000, J1850 and BEAN. BEAN (Body Electronic Area Network) is used in Toyota vehicles along with CAN. BEAN runs at 10 Kbits/sec and is a single wire, 0 to 10 volt bus. Since the Toyota CAN is 500 Kbits/sec, the raw CAN data transfer is 50 times that of BEAN. BEAN, however, is quite capable of handling the normal body application load.

### Polled and Periodic Messages

Polled messages are those that a ECM requests from another ECM (or even itself) for some specific information. For example, an ECM could request the status of the brakes or other vehicle subsystems. A scan tool could request some specified diagnostic information. Polled and periodic messages are used on both the Standard and Diagnostic modes of ECMs.

Periodic messages are broadcast on the network on a periodic basis, i.e. every 120 msec. They do not need to be requested by an ECM. Speeds, temperatures, pressures and commands are commonly sent in this manner at specific times.

Scan tools generally read both polled and periodic messages. You can view both types with an oscilloscope, but in order to request and subsequently see the polled messages, a CAN analyzer, scan tool or an ECM will be needed.

### Standard CAN Traffic - Public and Proprietary

Standard traffic consists of broadcast messages. It is easy to see the raw data with an oscilloscope or preferably a CAN analyzer, but knowing exactly what a message means can be challenging.

It is not clear what the following CAN message represents:

```
2C1 08 05 83 28 06 EC 00 75
```

All we know about it is that it is a standard 11 bit Identifier CAN frame with 8 bytes of data and the identifier is hex 2C1.

A public protocol such as J1939 or ISO9141 will specify what its frames mean as well as other details about the entire system.

A proprietary protocol such as found in many vehicle applications will not be disclosed except to approved vendors with the appropriate nondisclosure documents in place. It is possible to find out some things about these systems through trial and error and observation. It is impossible to derive all the secrets of these systems in this manner due to their complexity.

Most vehicles contain a mixture of public and proprietary messages. The public diagnostics system OBDII is required by law in the USA, Canada and Europe.

OEM scan tools usually use both public OBDII and proprietary codes. Aftermarket scan tools use only the public OBDII.

The scan tool receives these error codes and displays them. The user then must look in a book to decipher the code given or the translation of the codes is displayed depending on the scan tool design. The meaning of these codes is publicly available.

In Figure 3, the **P0742** means “Torque Converter Clutch Circuit Stuck On”. The “P” specifies a powertrain problem and other prefixes used are body (B), chassis (C) and network (U).

CARB and EPA allow a newer CAN based protocol specified by J2284-3 to be used as an alternative to OBDII in MY 2003. It specifies a CAN bus running at 500 Kbits/sec.

### SAE J2534 - Reprogramming

CARB and the EPA also specify that all automobiles and light trucks sold in the USA must provide a method of reprogramming their ECMs beginning in model year 2004. The same J1962 connector in Figure 3 is used.

The J2534 specification will soon include diagnostic capabilities. OEM scan tools already have reprogramming capabilities and J2534 is designed to extend this capability to the aftermarket.

#### Updating FLASH Memory Contents

Modern ECMs usually have their software programmed into FLASH nonvolatile memory that can easily be programmed to repair software bugs and provide various upgrades. These are normally emissions related but OEMs can also use J2534 to update their modules’ firmware.

#### Immobilizer and Security Features

Many vehicles have an immobilization feature. Each ECM is programmed with the serial numbers of the other ECMs in the vehicle. This makes it harder for thieves to swap ECMs to disable anti-theft features and get stolen vehicles running.

If a module is legitimately replaced as a service event, the new ECM(s) must be reprogrammed with the appropriate vehicle codes in order to operate.

Figure 4 is an example of a J2534 reprogramming tool. The software, because of many security concerns, is supplied directly by the vehicle manufacturer.



Figure 4: J2534 Pass-Thru Reprogramming Device

#### J Series and ISO Documents

J series (i.e. J2534) are standards from the SAE ([www.sae.org](http://www.sae.org)). The European ISO organization provides their own specifications. These two bodies overlap somewhat in automotive standards and these specifications are available for purchase.

## CAN Primer - Controller Area Network

### High Points of the CAN Bus

- 1) A twisted differential wire pair for robust noise immunity.
- 2) The highest priority message always gets through.
- 3) There are no masters or slaves - all nodes see all messages.  
CAN is a peer-to-peer network.
- 4) The priority level of a message is also its identifier.

### CAN Controllers

There are many microcontrollers available with CAN modules integrated including PowerPC, HC12, Coldfire, ARM, C166, 8051 and ST10. Stand-alone CAN controllers are also available from companies such as Intel, Philips and Infineon. These companies also supply the physical layer chip which creates the differential voltages. Most CAN controllers require this chip to connect to the physical wires. Most of the work is done by the CAN controller relieving this substantial burden from the CPU.

### CAN Physical layer

CAN is normally a differential twisted pair but there are single wire versions. The twisted pair can be shielded but this lowers the speed of the network. The most popular physical layer has a differential drive with *active* voltages at 3.5V and 1.5V and *passive* voltages, both at 2.5 volts; with respect to ground. See Figure 5. The two wires are designated CAN\_Hi and CAN\_Lo.

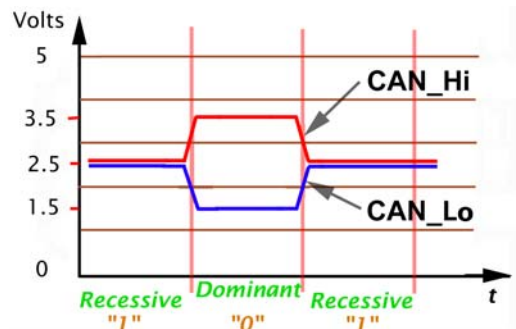


Figure 5: CAN Physical Layer Voltages

### CAN speeds

Maximum speed is 1 Mbits/sec with 250K and 500K common. The higher the frequency, the shorter the permissible network cable lengths. Maximum CAN cable lengths are typically a 1 meter drop line from the bus and total bus length up to 40 meters (@ 1 Mbit/s). A given system will have only one speed. It is not common to dynamically change speeds on a CAN bus.

### Identifier

A CAN message can be *Standard* with a 11 bit identifier or *Extended* with a 29 bit identifier. The identifier is used to provide unique frames for use in the system.

With 11 bit, 2,048 unique messages are possible and 29 bit provides 536 million. CAN frames are not addressed to a specific recipient. All recipients can see all messages and can accept or ignore any messages according to the system design.

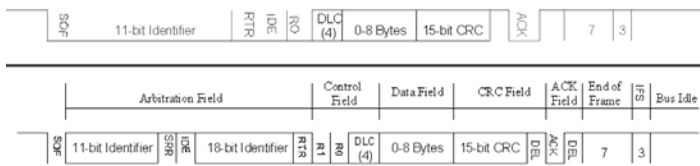
**Dearborn Group, Inc.** [www.dgtech.com](http://www.dgtech.com)  
Phone (248) 488-2080

The CAN specification CAN 2.0A has an 11 bit identifier and CAN 2.0B is both 11 and 29 bit. Figures 6 and 7 show the two general types of CAN data messages. The frame length depends on whether it is 11 or 29 bits, the number of data bytes (0 to 8) and the number of stuffing bits. More on stuffing bits later.

Figure 8 is an oscilloscope reading of a 500 Kbits/sec CAN data stream. They have 11-bit identifiers and each message is about 200 usecs wide and the space between them is about 100 usec. There are about 2,500 such messages every second on this bus.

**Data Frame**

CAN has 4 types of frames: data, remote, error and overload. A CAN data frame consists of a start bit, identifier (11 or 29 bits), data bytes of zero to eight bytes and various small fields including the CRC check. The start of a frame (SOF) is a falling edge and the end of frame is 7 bits high.



**Figure 6 (top): 11 bit Identifier  
Figure 7 (bottom): 29 bit Identifier CAN Frame**

**Error Frame**

CAN has a robust error mechanism and a CAN controller will take itself off the bus if it detects it has generated too many errors. The only way it can return to the bus is if it (the CAN controller) is forced to a RESET condition. A quality CAN analyzer such as the Gryphon/Hercules system will detect and record these errors for debugging purposes.

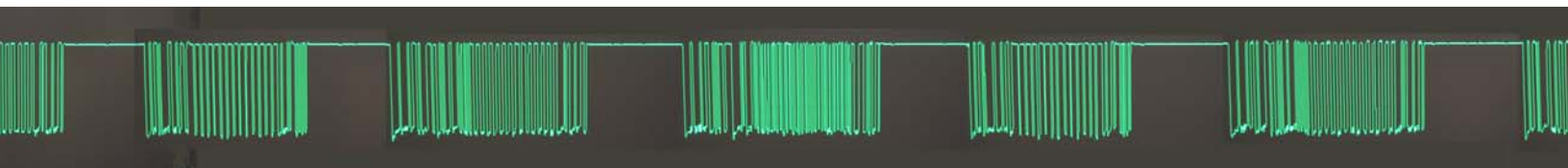
**Bit Stuffing:**

To help the CAN controller clocks stay synchronized, each CAN message can have no more than 5 bit times without at least one bit transition. The controller will add a bit to the CAN message to satisfy this rule and this is called bit stuffing. The CAN controller chip will automatically add and subtract these bits to the bus. They are normally seen only with an oscilloscope.

**Priority Levels**

The CAN message with the highest priority always gets control of the bus and overrides lower priority messages. The priority is determined by the value of the message identifier. The lowest value corresponds to the highest priority. Zero (0) has a higher priority than two (2) and two has a higher priority than three (3) and so on up to the maximum of the 11 or 29 bit identifiers. This arbitration is automatically evaluated by the CAN controller hardware in real-time as the bits come in from the CAN bus.

↓ **Figure 8: Oscilloscope Display of a CAN bus**



**Differential Voltage levels in CAN**

It is important to know that the voltage levels from each of the two CAN lines to ground or to the vehicle chassis are not the important ones. What is crucially important to CAN is the voltage between the two lines or their difference voltage.

This is the essence of a differential pair. Therefore, to “pull the bus low” this really means pulling them apart by 2 volts. If all the CAN controllers do nothing, then the two bus lines come together with a difference of 0 volts. This is the idle state.

**Recessive and Dominant States:**

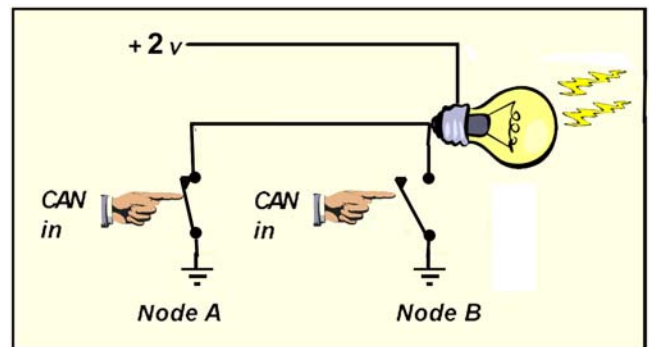
Voltages levels of 0 and 5 volts can be also expressed as low & high, 0 or 1, False or True... or in the case of CAN, Dominant or Recessive. The terms Dominant and Recessive refer to not only the voltage/logic levels, but also gives a sense of the priority scheme of CAN. Dominant is “0” and Recessive is “1”. Dominant overrides Recessive on a bit by bit basis. See Figure 5 to review these two possible states of the CAN bus.

Any node on the bus can only make the signal go from a “1” to a “0”. It cannot force a “0” into a “1” (Wired OR). At rest, the bus is at a “1”. A node therefore makes a “0” by pulling the bus apart by 2 volts and a “1” by not doing anything and allowing the bus to come together to its idle difference of zero volts.

**Arbitration Voltage Levels**

It makes sense then, that a node making a “0” will override another node making a “1” (by doing nothing). This is the essence of CAN arbitration.

Figure 9 illustrates this principle. If both switches are open, the voltage difference across the lamp is zero - this is the recessive state. Either switch can put a difference voltage across the lamp causing it to illuminate. This is the dominant state and a voltmeter will measure a difference voltage of 2 volts across the lamp. In this case, the right hand switch is unable to turn the lamp off - each switch can only turn the lamp on: not off if the other is already on. If both switches are open, the difference voltage will become 0 volts and the lamp will go out.



**Figure 9: CAN Output Circuit for Arbitration**

## Arbitration Bit by Bit

Each node can detect if its output matches what is on the bus. If a node is putting out a “1” (by doing nothing), and it sees a “1” on the bus, it continues with the next bit and so on until the frame is completed.

If it puts out a “1”, and sees a “0” (from another node), it immediately stops sending and this allows the node with the dominant bit to continue its frame. The highest priority message (i.e. the lowest identifier bit = 0) continues transmitting.

Remember any node can create a dominant state, but none can force a recessive state over a node creating a dominant state.

At the end of the identifier field, either 11 or 29 bits, one CAN message will prevail over all others. This is because it is illegal to put multiple CAN messages on the bus at the same time with the same identifier.

## Building Messages

A CAN message can contain from 0 to 8 data bytes. The reception of a frame with no data does pass useful information by its mere presence. Much like an unanswered telephone says that somebody called but left only their number on caller ID.

Messages will normally be created by embedding information inside the bytes of a CAN message or messages. For instance, byte 1 could be the engine temperature, byte 2 the RPM and byte 3 the vehicle speed. A byte can be split up to contain more than one data field and these fields can also span byte boundaries. A CAN analyzer can reconstruct and display these messages.

Higher level protocols such as OBDII, J1939 and proprietary schemes will be used to construct these special messages.

## CAN Traffic Examples

### Actual Vehicle CAN Traffic

Figure 10 is a stream of actual CAN traffic on a 2004 vehicle. These are periodic proprietary standard (non-diagnostic) messages for a period of about 140 msec. The messages were saved with Dearborn Group Gryphon and Hercules CAN analyzer and the timestamps are in msec:usec.

These messages consist of the following identifiers: 025, 0B0, 0B2, 2C1, 223, 224, 2D0, 2D2, 2C4, 3D0, 4C1 and 4D7. Message 4D7 appears about every 2 seconds so doesn't show in this series of messages. 4D7 is therefore manually added out of sequence as the last message. To give some sense of timing with the CAN messages: 0B0 is purple, 0B2 is red and 2C4 is blue. 0B2 is also underlined for black and white printed versions.

It is easy to see that these messages are put on the bus periodically. Note that they are not always in the same order nor at exactly the same time. This illustrates the non-deterministic nature of CAN. You never know exactly when a message will get through. Note the timing relationship between 0B2 and 025.

From this information we know these messages have Standard 11 bit identifiers and their values, 1, 6 or 8 bytes of data, all the data values and when they occurred from the timestamp created by the Gryphon. These timestamps are not part of CAN.

Polled messages could be seen on this vehicle if a scan tool is connected. The scan tool makes requests to the ECM and the replies were displayed on the Gryphon's Hercules software.

Timestamp	ID	CAN Data
000:000	<u>0B2</u>	<u>00 48 00 48 00 00</u>
004:510	2D2	00
008:240	025	00 A2 00 00 00 00 00 CF
010:240	0B0	00 48 00 48 00 00
011:810	2C4	00 00 00 20 00 80 21 8F
012:260	<u>0B2</u>	<u>00 48 00 48 00 00</u>
014:360	223	00 00 00 00 00 00 00 2D
016:420	224	00 00 00 00 00 00 00 00
020:510	025	00 A2 00 00 00 00 00 CF
021:030	2C1	08 05 83 28 06 EC 00 75
022:500	0B0	00 48 00 48 00 00
024:540	<u>0B2</u>	<u>00 48 00 48 00 00</u>
029:220	2D0	00 00 08 00 10 00 00 F2
032:770	025	00 A2 00 00 00 00 00 CF
034:770	0B0	00 48 00 48 00 00
035:370	2C4	00 00 00 20 00 80 21 8F
036:260	2D2	00
036:840	<u>0B2</u>	<u>00 48 00 48 00 00</u>
038:930	223	00 00 00 00 00 00 00 2D
041:000	224	00 00 00 00 00 00 00 00
045:050	025	00 A2 00 00 00 00 00 CF
047:060	0B0	00 48 00 48 00 00
049:110	<u>0B2</u>	<u>00 48 00 48 00 00</u>
052:770	2C1	08 05 83 28 06 EC 00 75
057:350	025	00 A2 00 00 00 00 00 CF
058:920	2C4	00 00 00 20 00 80 21 8F
059:340	0B0	00 48 00 48 00 00
060:980	2D0	00 00 08 00 10 00 00 F2
061:390	<u>0B2</u>	<u>00 48 00 48 00 00</u>
061:860	3D0	00
063:480	223	00 00 00 00 00 00 00 2D
065:510	224	00 00 00 00 00 00 00 00
068:020	2D2	00
069:650	025	00 A2 00 00 00 00 00 CF
071:630	0B0	00 48 00 48 00 00
073:650	<u>0B2</u>	<u>00 48 00 48 00 00</u>
081:900	025	00 A2 00 00 00 00 00 CF
082:480	2C4	00 00 00 20 00 80 21 8F
083:890	0B0	00 48 00 48 00 00
084:530	2C1	08 05 83 28 06 EC 00 75
085:930	<u>0B2</u>	<u>00 48 00 48 00 00</u>
088:050	223	00 00 00 00 00 00 00 2D
090:090	224	00 00 00 00 00 00 00 00
092:720	2D0	00 00 08 00 10 00 00 F2
094:800	025	00 A2 00 00 00 00 00 CF
096:160	0B0	00 48 00 48 00 00
098:220	<u>0B2</u>	<u>00 48 00 48 00 00</u>
099:750	2D2	00
106:020	2C4	00 00 00 20 00 80 21 8F
106:450	025	00 A2 00 00 00 00 00 CF
108:450	0B0	00 48 00 48 00 00
110:490	<u>0B2</u>	<u>00 48 00 48 00 00</u>
112:600	223	00 00 00 00 00 00 00 2D
114:620	224	00 00 00 00 00 00 00 00
115:270	4C1	01 00 00 04 00 00 00 00
116:270	2C1	08 05 83 28 06 EC 00 75
118:740	025	00 A2 00 00 00 00 00 CF
120:730	0B0	00 48 00 48 00 00
122:770	<u>0B2</u>	<u>00 48 00 48 00 00</u>
124:480	2D0	00 00 08 00 10 00 00 F2
129:590	2C4	00 00 00 20 00 80 21 8F
131:020	025	00 A2 00 00 00 00 00 CF
131:510	2D2	00
133:010	0B0	00 48 00 48 00 00
135:030	<u>0B2</u>	<u>00 48 00 48 00 00</u>
137:130	223	00 00 00 00 00 00 00 2D
---:---		
820:550	4D7	17 00 05 00 00 00 00 00

Figure 10: Actual CAN Traffic

### hex 2D0 CAN Message Example

Identifier 2D0 contains, among other items, the engine RPM as a 16 bit (2 bytes) value as shown here: 10CE is the RPM data.

2D0 10 CE 08 00 10 00 00 D0

Figure 10 shows 2D0 is transmitted periodically on the CAN bus every 31.7 msec for use by any ECM that needs it.

Figure 11 is another CAN sequence that shows only the 2D0 messages on the bus. Many of the frames have been deleted for brevity. The Gryphon has filtered all the other messages out in the same manner an ECM would ignore all messages except the ones it needs to see. In addition, four bytes of the frame has been replaced by the “~” for clarity.

The vehicle is started and after 10 seconds the engine is revved up four times by quickly pressing on the accelerator pedal.

The CAN data clearly varies according to the RPM value. It is not clear why the RPM value stays at zero until the accelerator is pressed once. Perhaps 2D0 represents something other than RPM. Without access to the proprietary data from the manufacturer, it will be rather difficult to determine exactly what 2D0 is.

Figure 12 is a graph of this data and the accelerator activations can be more clearly be seen than the data in Figure 11.

Time	ID	CAN Data 8 bytes	Comments
0	2D0	0000 08 ~ F2	start engine
0.031	2D0	0000 08 ~ F2	
~ 300 lines deleted (about 9 sec)			
9.842	2D0	0000 00 ~ EA	
9.874	2D0	0000 00 ~ EA	
9.906	2D0	010B 00 ~ F6	accel pedal on
10.826	2D0	0E8F 08 ~ 8F	
10.858	2D0	0E5A 08 ~ 5A	
10.89	2D0	0E23 08 ~ 23	
11.779	2D0	0A24 08 ~ 20	off
11.811	2D0	0A1B 08 ~ 17	
11.842	2D0	0A17 08 ~ 13	
12.795	2D0	0B16 08 ~ 13	
12.827	2D0	0B12 08 ~ 0F	
12.859	2D0	0B0C 08 ~ 09	
13.779	2D0	0A49 08 ~ 45	
13.811	2D0	0A47 08 ~ 43	
13.843	2D0	0A47 08 ~ 43	
14.795	2D0	096D 08 ~ 68	
14.827	2D0	0966 08 ~ 61	
14.859	2D0	0960 08 ~ 5B	accel pedal on
15.78	2D0	14B2 08 ~ B8	
15.811	2D0	146E 08 ~ 74	off
15.843	2D0	13F3 08 ~ F8	
16.795	2D0	09D7 08 ~ D2	
16.827	2D0	09CB 08 ~ C6	
16.859	2D0	09B6 08 ~ B1	
17.78	2D0	1A8B 08 ~ 97	accel pedal on
17.811	2D0	1B16 08 ~ 23	
17.843	2D0	1B38 08 ~ 45	
18.796	2D0	0C2B 08 ~ 29	off
18.827	2D0	0BAF 08 ~ AC	
18.859	2D0	0B74 08 ~ 71	
19.78	2D0	2240 08 ~ 54	
19.812	2D0	2214 08 ~ 28	
19.843	2D0	21B9 08 ~ CC	
20.796	2D0	0F51 08 ~ 52	accel pedal on
20.828	2D0	0ECD 08 ~ CD	
20.859	2D0	0E45 08 ~ 45	off

Figure 11: 2D0 CAN Messages (RPM)

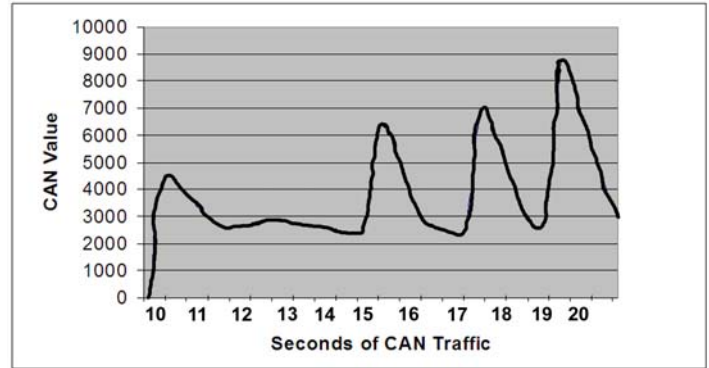


Figure 12: Graph of 2D0 Engine RPM

### Rear Height CAN Message Example

Another example shown in Figure 13 involves a different vehicle. The message with identifier 479 is periodically sent out about twice a second. It is a standard 11 bit identifier with three bytes of data.

The second and third bytes contain the code that operates the rear suspension adjustment air pump.

When a CAN message with an identifier of 479 is sent with a data word of hex 0000 is sent on the CAN bus the air pump is turned off. A data word of 0190 adjusts the height up and FED4 adjusts it down.

The vehicle operator presses a button and these codes are sent out in response. The ECM responsible for the air pump receives the messages, recognizes it and acts accordingly.

ID	Data	Comments
479	70 0000	- neutral
479	70 0190	- up
479	70 FED4	- down

Figure 13: Data of Rear Height Adjustment

## Testing and Measuring a CAN bus:

### General Information

Standard equipment such as voltmeters and oscilloscopes (scopes) are limited in value with CAN and other automotive serial networks. The data traffic on a CAN bus is not normally repetitive enough to make meaningful measurements as with a steady state signal. Special techniques and equipment are necessary to effectively design and troubleshoot these networks.

### Voltmeter

**Voltage:** At idle, a digital voltmeter connected to the two CAN wires will measure 0 volts - this is the "1": the difference of CAN\_Hi and CAN\_Lo wires. This is the rest or idle state.

With CAN messages on the bus, the voltmeter will indicate perhaps 0.5 volt or so depending on the nature of the bus traffic. This will be the average voltage differential of the bus less those effects too high in frequency for the voltmeter to respond to. This makes voltage readings almost useless for servicing CAN.

**Resistance:** Voltmeters are useful in detecting CAN line shorts to ground, battery or to each other using the ohmmeter function. Typical default resistances are published by the automotive OEMs and will vary depending on the number and type of a vehicle's ECMs. The vehicle key must be off to accurately measure the resistance of the CAN bus.

**Problems with Voltage on the CAN Bus:** Ohmmeters will not give accurate readings if there is any voltage present on the bus. Ohmmeters work by putting a small DC voltage across a circuit and measuring the resulting current. The resistance in ohms is calculated and displayed. Therefore, an external voltage will cause an erroneous measurement by adding or subtracting to the ohmmeter voltage. A clue is if the ohmmeter gives two different resistance readings when the probes are reversed. If in doubt, use the voltmeter to test for zero volts across the CAN lines and to the vehicle ground before making resistance measurements.

To test for the resistance from the battery to either CAN\_Hi or CAN\_Lo, the battery must be disconnected. Otherwise the leakage voltages will render the measurement useless.

**Shorts and Opens:** The CAN controllers will tolerate a short circuit of one of the two lines to ground because of the characteristics of the differential bus. It cannot tolerate both CAN bus wires shorted to ground or to each other. It will tolerate one of the CAN lines being an open circuit.

Corrosion can cause a higher or lower resistance of the CAN bus and this will probably degrade the network often causing erratic or intermittent failures that are difficult to diagnose. An ohmmeter is the perfect instrument for detecting anomalous DC resistance values. You can tell a great deal about the health of a CAN network by its resistance values.

**Termination Resistors:** The CAN specification requires a 120 ohm termination resistor at each end of the CAN bus. This will result in a resistance of about 60 ohms across the CAN\_Hi and CAN\_Lo wires if these resistors are installed. The reason is the resistance of two equal resistors in parallel is one half of one of the resistances. The load from the ECMs will lower this value further but only slightly and is likely negligible.

**Practical Measurements Example:** On a new Toyota vehicle the resistance between the CAN lines was 62 ohms, from either line to ground was 7 Kohm on one vehicle and 13 Kohm on another model. These values were well within the vehicles' specifications. The resistance for the CAN lines to the battery was invalid due to the voltage issue as described above. We did not remove the battery to test this.

**Frequency:** Some voltmeters have a frequency function. This will not normally be very useful as the CAN messages are not periodic enough for the meter to properly measure. CAN messages generally confuse frequency measurement instruments.

### Oscilloscope (scope)

**Non-repetitive Signals:** The network traffic on many buses, including CAN, is not repetitive and hence rather difficult for a standard oscilloscope to trigger on and display. Digital storage scopes are needed to effectively view these waveforms.

Ordinary scopes rely on displaying the same image repeatedly and this requires a stable and repetitive signal for a clear and jitter free display. Figure 14 is the blurred image from a analog scope. Compare this with the same frame digitally sampled and stored in Figure 15 and it is clear to see the difference.

Figure 14 is a result of many CAN messages on top of each other and at different times as the scope sweeps overwrites its screen multiple times. Figure 15 is one CAN message sampled once and stored in the scope's memory for display.

From Figure 14 there is some useful information. It can be seen that the approximate voltage of the bus is correct and that the signals are present and not shifted high or low and the top or bottom is not clipped. It does not prove they are CAN frames. Circuit defects can cause waveforms to show these problems.

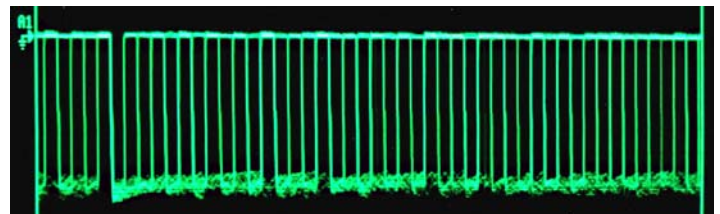


Figure 14: CAN Frames on Analog Scope

Figure 15 is a 11 bit CAN frame with identifier 010 and one byte of data of value hex 55. It was captured with a HP 54645D 100 MHz digital storage scope. While it is easy to see the overall waveform is clean, of proper voltage and frequency, it is difficult to calculate the identifier and data values.

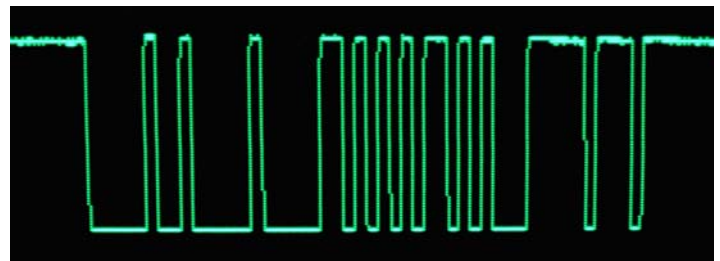


Figure 15: CAN Frame ID hex 010, data hex 55

**Frequency:** The scope can easily determine the frequency of a CAN message. Simply determine the width of the most narrow pulse of the waveform and divide this number into 1 (invert it). For example, in Figure 15, the first pulse will suffice as there are no pulses more narrow. This pulse measures 2 usec in duration which is 2 millionths of a second ! (this is easy for scopes to do). This is equivalent to 0.000002 seconds or  $2 \times 10^{-6}$  second. Inverting this (divide 1 by 0.000002) with a calculator returns 500,000 or 500 Kbits/sec which is the speed of this CAN bus. Some scopes have methods to calculate this automatically. All you need to do select the pulse to be measured.

**Signal Faithfulness:** The smallest bit pulse of Figure 15 could be represented by a 250 kHz sine wave and the wider pulses with lower frequencies. They will then have rounded corners and sloped sides. A scope with a 250 kHz bandwidth would display a waveform in this manner diminishing its usefulness.

The square corners of the waveform, caused by odd harmonics at multiples of 250 kHz (750, 1,250, 1,750 etc.), any noise pulses and other anomalies with frequencies somewhat greater than 250 kHz will not be displayed or will be severely attenuated.

This would have severe limits on the usefulness of such a scope. Some digital scopes found in scan tools have such limitations and software techniques might be used to artificially “square up” the waveform. This is not good engineering practice.

Figure 16 shows three CAN frames with a 10 kHz digital automotive scope. Compare this signal with Figure 15. It is clear the waveform in Figure 16 has been “squared up” with mixed results. The CAN frame is difficult to decipher and does not faithfully reproduce the waveform. It does provide some basic information that is useful.



**Figure 16: 100 kHz Digital Scope CAN Waveform**

The signal in Figure 15 was reproduced with a 100 MHz scope although this is far in excess of what is needed for CAN work. A 10 to 30 MHz scope will be sufficient.

#### Connecting Scopes to a CAN bus

With the CAN differential pair, the CAN\_Hi and CAN\_Lo lines will operate opposite of each other. If you connect a scope hot lead to one of the CAN lines and the scope ground lead to the vehicle ground, you will only see one half of the CAN waveform.

You could use the second channel of a dual channel scope to see the other side of the waveform but to properly combine them a special differential probe is needed. This probe will take each of the CAN lines, subtract them and present this to the scope.

However, you can take advantage of the fact that the ground of the scope and of the vehicle are not connected together. Connect

the scope hot lead to one CAN line and the ground to the other CAN lead. The differential waveform will display correctly. This is how Figures 5, 15 and 16 were made. If the vehicle and scope grounds are connected, this would short the one CAN lead to the vehicle ground through the scope ground lead.

#### CAN Analyzer Tools

Standard CAN bus analyzers can read CAN traffic, put user specified messages on the bus, provide statistical information such as bus loading and bus error information.



**Figure 17: Gryphon CAN Analyzer**

Figure 17 is the Gryphon. It connects directly to the OBDII connector on the vehicle and will capture OBDII codes but also proprietary messages similar to that shown in Figure 18.

See [www.dgtech.com/product/gryphon/flier/gryphon.pdf](http://www.dgtech.com/product/gryphon/flier/gryphon.pdf) for more information on CAN analyzer capabilities.

A quality CAN analyzer will display the CAN messages giving the identifier and any data values. Many analyzers will display the acronym for a particular CAN message.

Figure 18 shows a typical display from the Dearborn Group Gryphon/Hercules analyzer. Not only are the identifiers, data values and a timestamp shown, but also the acronym of the identifier if available.

Line No	TimeStamp (µs)	Channel	Frame ID	Frame Acronym	Protocol	Data	Rx/Tx
67	00:00:08:604:590	CH#A	00000130	AC ON	CAN - EXT	00 00	Tx
68	00:00:08:604:670	CH#B	00000130	AC ON	CAN - EXT	00 00	Rx
69	00:00:09:084:040	CH#A	110	Brake ON	CAN - STD	00 00	Tx
70	00:00:09:084:120	CH#B	110	Brake ON	CAN - STD	00 00	Rx
71	00:00:09:092:880	CH#A	00000130	AC ON	CAN - EXT	00 00	Tx
72	00:00:09:092:960	CH#B	00000130	AC ON	CAN - EXT	00 00	Rx

**Figure 18: Analyzer Display of CAN Frames**

**Dearborn Group, Inc.** [www.dgtech.com](http://www.dgtech.com)  
 Phone (248) 488-2080