

The Use of CAN Bus Message Electrical Signatures for Automotive Reverse Engineering

C Quigley, D Charles, R McLaughlin

Warwick Control Technologies

Abstract

There are many applications in which you may need to reverse engineer the Controller Area Network (CAN), e.g.:

- Automotive competitor analysis
- Telematics applications such fleet management
- Disabled driver applications

The typical reverse engineering process is concerned with moving a sensor and watching the CAN bus for message changes. For example, wind down a door window and see if this kicks off changes in CAN message data.

Many CAN buses have many messages originating from many Electronic Control Units (ECUs). This means it is difficult to watch all of them at the same time. It would be far easier if you could simply watch a smaller number of CAN messages to observe changes by isolating the ECUs the messages originate from.

This paper describes a process that allows the user to identify which CAN messages are transmitted by a particular ECU. This is achieved by getting the electrical signature of each CAN message and matching known CAN messages with unknown ones. Therefore, the transmitting ECU of the unknown CAN messages can be determined.

The method for determining which Identifiers come from a particular ECU is to first get electrical signature plots of known diagnostic response messages and compare with electrical signature plots of the real time control messages. We show how to achieve this using Warwick Control's tool X-Analyser coupled with a PicoScope PC oscilloscope and the a Kvaser CAN USB interface.

This paper requires prior basic knowledge of the workings of the CAN bus technology.

1 What is a CAN Message Electrical Signature?

A CAN message electrical signature is something that is largely unique about any message sent by an ECU. Therefore, you would expect all messages transmitted by an ECU to have the same electrical characteristics. For example, a CAN message comprising of the voltages of CAN High and CAN Low (CAN_H and CAN_L) should show something unique for each ECU due to the physical makeup of the CAN bus (e.g. node position and distance on the bus).

Figure 1 shows different fields that make-up a CAN frame. Due to the nature of the contention-based access method of CAN, the Arbitration field (CAN ID) should not be considered for the electrical signature, as there may be several ECUs communicating within this field and therefore influencing the electrical signal.

Once the arbitration process is completed, there is just one ECU producing the Data Field. This is where you see a unique electrical signature for this ECU. To obtain a unique signature for a CAN message

that represents its transmitting ECU, the measurements should be taken from this part of the CAN frame, which is when only one ECU is generating the CAN data.

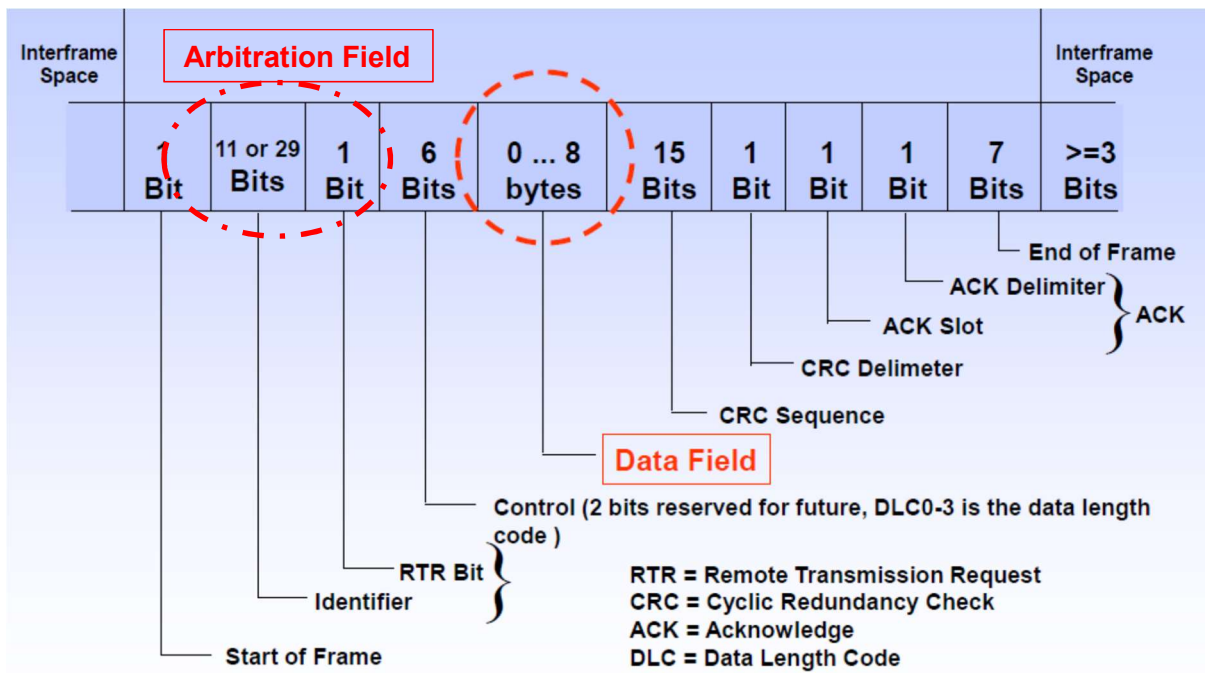


Figure 1 Construction of CAN Frame

To illustrate the unique electrical characteristics of each ECU in a vehicle, Figure 2 and Figure 3 below show the slight differences in the CAN_H and CAN_L voltages for two different ECUs from a modern passenger car. These are referred to as ECU A and ECU B.

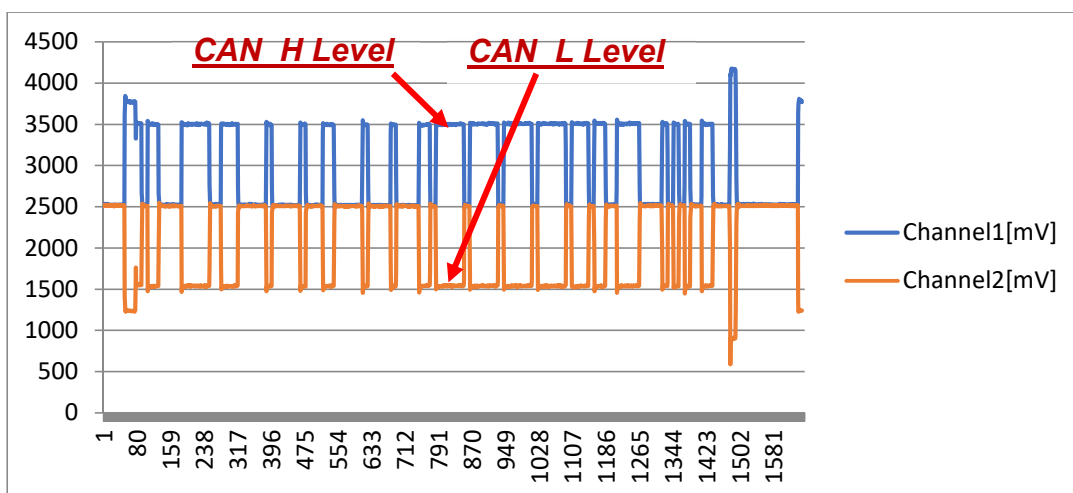


Figure 2 ECU A - Electrical Characteristics

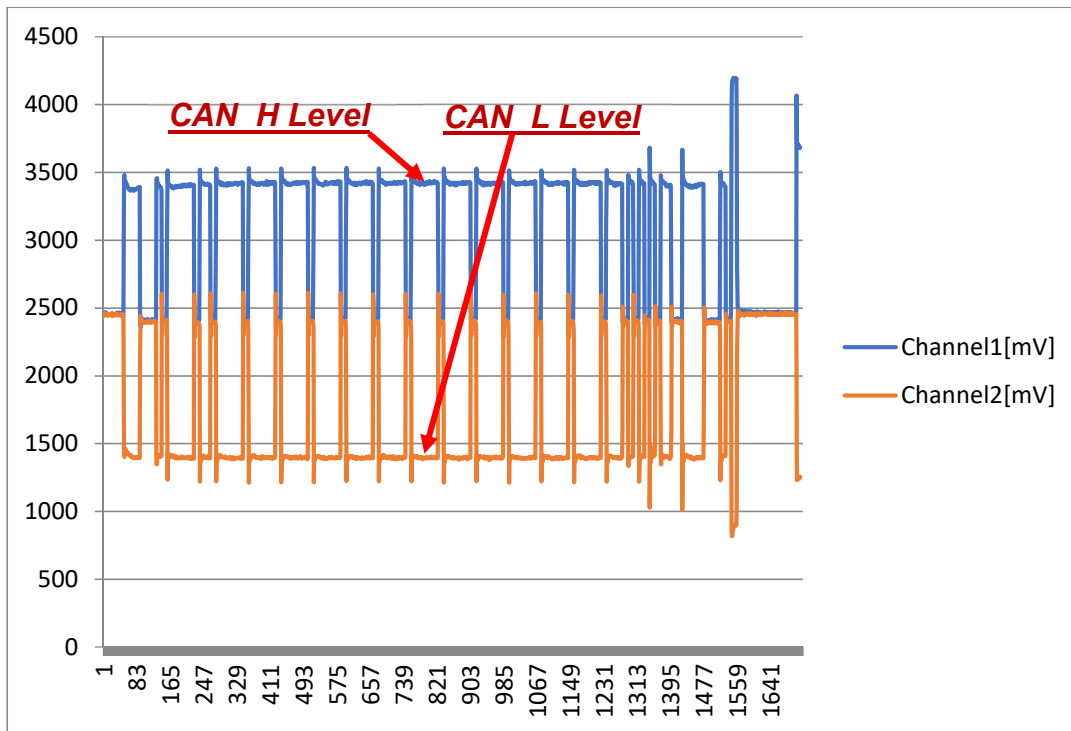


Figure 3 ECU B - Electrical Characteristics

It can be seen that the CAN_H and CAN_L voltage levels are different for the messages of these two ECUs.

2 Generating the Electrical Signature for Each CAN Frame

The methodology considered in gathering an electrical signature for each CAN message, allowing us to ascertain the ECU it comes from, is to consider the CAN_H and CAN_L voltage values to associate messages to ECUs.

Method– Analysing the voltages of CAN_H versus CAN_L

Process:

- Log one example of each CAN message oscilloscope trace
- Isolate the Data Field only
- Split Data Field bits into Dominant (logic 0) and Recessive (logic 1)
- Calculate modal average value of CAN_H and CAN_L voltage levels for Dominant bits only

Data is now ready for cluster plots.

Example in X-Analyser

Figure 4 shows the display in X-Analyser utilising the PicoScope interface. Here you can see CAN frames are logged on the top half of the display. One of the CAN frames are selected (highlighted), and the physical signalling of that frame is shown on the lower half of the display. Note that from this, we can gather the voltage levels of the dominant bits in the data field (CAN_H, CAN_L).

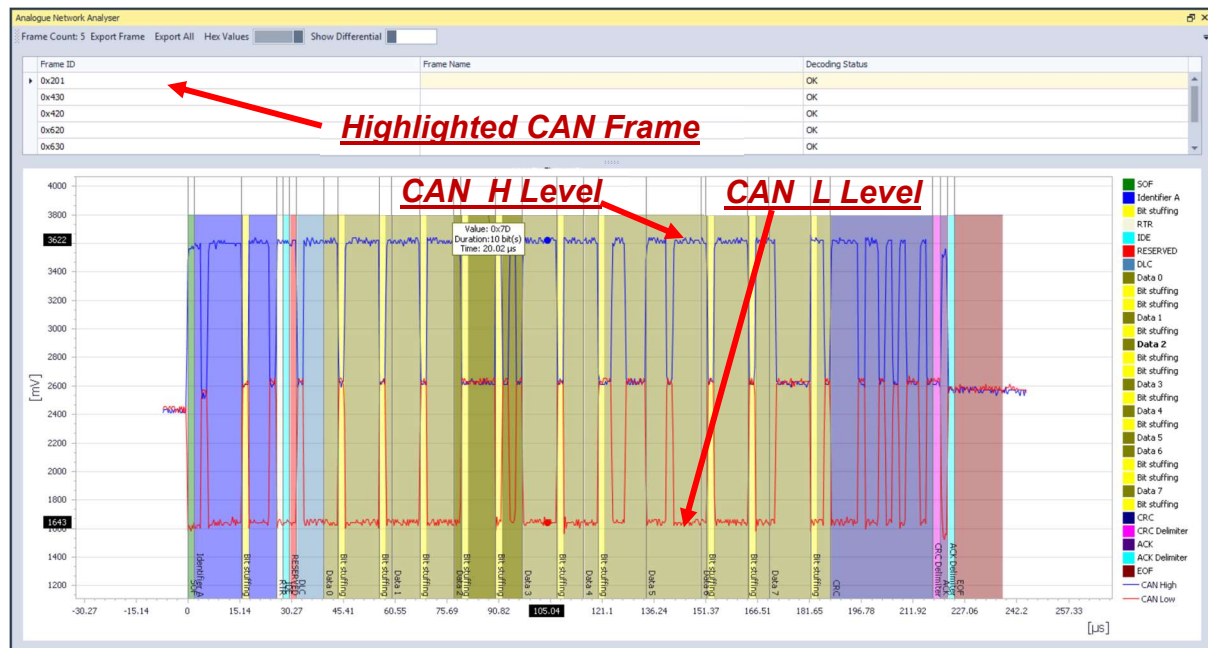


Figure 4 Highlighting a CAN Frame within a PicoScope Display

These waveforms can be exported as an Excel file to show readings of the CAN frame at a sample point. This is done within X-Analyser by the “Export Frame” button to export the selected frame and using the “Export All” button to export all the frame on that collection. An example of the data that is exported is shown in Figure 5.

	A	B	C	D	E	F	G	H
1	Export of Decoded CAN Frame			Time (us)	CAN High (V)	CAN Low (V)	Region Name	Additional Region
2	Frame ID	1CF7BC00		0	2.264	2.217	SOF	
3	DLC	8		0.144000009	2.303	2.257	SOF	
4	Data	C4 F8 FF FF 0B 27 C8 07		0.288000018	2.264	2.257	SOF	
5	Error Frame	FALSE		0.432000028	2.303	2.257	SOF	
6	Samples per Second	6944444		0.576000037	2.303	2.257	SOF	
7	Exported On	01/02/2018 10:43		0.720000046	2.303	2.257	SOF	
8				0.864000055	2.303	2.257	SOF	
9				1.008000065	2.303	2.217	SOF	
10				1.152000074	2.303	2.257	SOF	
11				1.296000083	2.264	2.257	SOF	
12				1.440000092	2.264	2.257	SOF	
13				1.584000101	2.303	2.257	SOF	
14				1.728000111	2.303	2.257	SOF	

Figure 5 Example Excel Data Exported for an Extended CAN Frame

The information given in the Excel file is;

- Frame ID (Hexadecimal)
- DLC
- Data (bytes in Hexadecimal)
- Error Frame (True or False) (False if a good CAN frame)
- Samples per Second
- Exported On (Date)
- Time (of sample for that frame, starts at zero)
- CAN-H and CAN-L Voltages
- Region Name (Region of the frame the data showing, is in)
- Additional Region (shows where bit stuffing occurs)

Once this information is exported to Excel, we can calculate cluster points using the method taking the Modal average of CAN_H and CAN_L voltages from data field (Dominant bits only).

Analysis and Clustering of the Data

The data is analysed by recording the level of CAN_H and CAN_L dominant bit voltage levels within the data field and coming up with a single modal average measure for both CAN_H and CAN_L. These can then be put onto a cluster plot so that the clustering of CAN messages from a particular ECU can be observed.

The Case Study below illustrates the data collection methods, and process utilised in plotting the CAN ID clusters from the Excel modal average values. This allows a researcher/engineer to ascertain which ECUs the real time CAN messages originate.

3 Case Study – Automotive Reverse Engineering - Methodology for Identifying Transmitting ECU

The basis for this methodology is that each ECU on the CAN bus will exhibit its own unique electronic characteristics which are influenced by aspects such as its electrical components and tolerances, CAN transceiver, connector characteristics and location in the CAN bus. This can therefore be used to match unknown CAN frames with known CAN frames. In the automotive industry, the real-time control CAN messages are proprietary. However, the identifiers diagnostic messages used for manufacturing and service garages is standardised in specifications such as ISO15765 [1] and/or across an automotive manufacturer.

It is well known that many vehicles using standard CAN identifiers make a diagnostic request to the engine controller is made using CAN identifier 0x7E0 and that the engine controller will respond on CAN identifier 0x7E8.

Therefore, the summary of the methodology is described by the following steps:

- Send diagnostic requests
- Get signatures of all responses and real-time messages
- Analyse and plot the data on a cluster diagram

Equipment Setup for Reverse Engineering on a Car

Figure 6 below shows an example of the equipment setup utilising X-Analyser connected to the CAN bus via the Kvaser CAN USB interface and the PicoScope interface.

Referring to Figure 6, the Kvaser Interface is used to generate Diagnostic Request messages, and the PicoScope is used to receive the Diagnostic Response message for analysis of the physical signature. X-analyser software is used to create the transmitters of ID 0x7E0 (or 0x700-0x7FF for other ECUs) through the object transmitter and uses the Kvaser interface to send these messages onto the bus. The PicoScope will see the sent transmitter (0x7E0) and read the response to this message of ID 0x7E8. 0x7E8 can then be analysed through the Analogue Network Analyser in X-analyser.

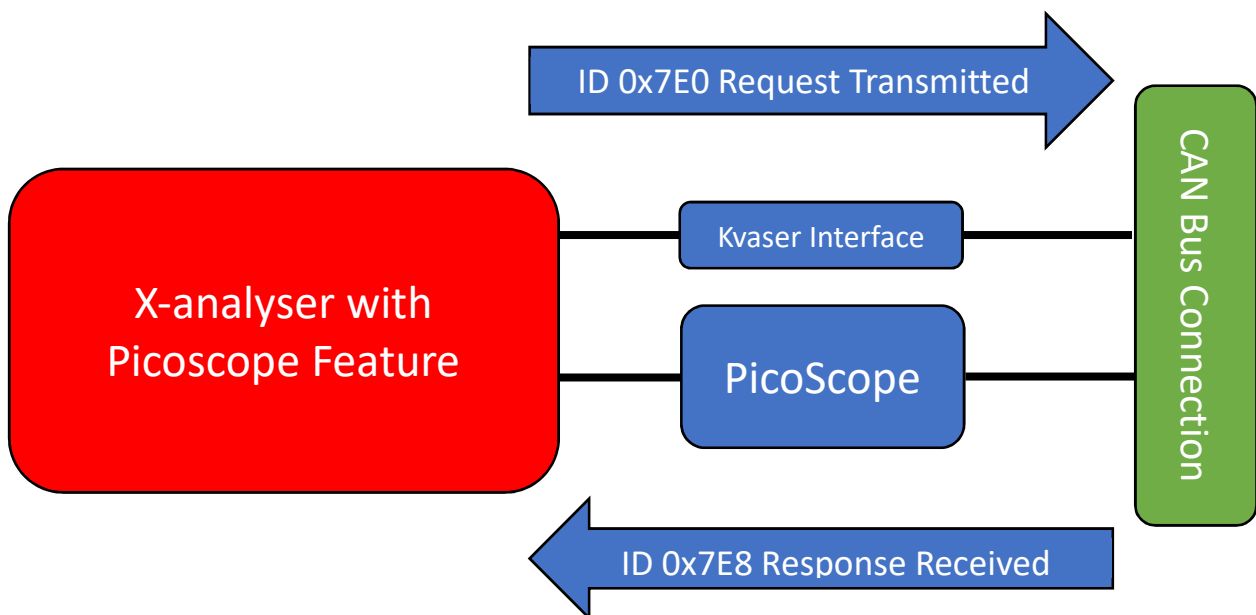


Figure 6 X-Analyzer connection to a car via Kvaser interface and PicoScope PC oscilloscope

Diagnostic Requests

More information about the diagnostic request can be found in ISO 15765-4:2016. The basic information needed is diagnostic request have the hexadecimal CAN IDs ranging from 0x700 – 0x7FF. The standard emission Diagnostic Request message is known to be ID 0x7E0 and the expected response from the ECM (Engine Control Module) is ID 0x7E8. Referring to ISO 15765-4:2016, page 29, it also known that the TCM (Transmission Control Module) Diagnostic Request ID is 7E1, and the response message is 7E9. Many of the other ECUs are manufacturer specific, but most can be ascertained utilising an OBD tool for a particular car model. For example, in many models, the ABS ECU is known to have a request of 7E2 and a response of 7EA.

A diagnostic response's ID will increase in value by 8 and give the response i.e.;

Request ID = 0x7E0 Response ID = 0x7E8 8 = 7E8 – 7E0

An example of a 0x7E0 diagnostic request CAN frame is;

ID = 0x7E0 DLC = 8 Data = 02 10 01 00 00 00 00

Therefore, we expect a response from the Emissions (Engine) ECU of on CAN ID 7E8.

If there is no response to other requests, it means that this diagnostic function is not supported in this vehicle. The plotted chart in Figure 7 shows the Diagnostic Response messages in the 1st candidate car. From this, we ascertained the Electrical Signatures of CAN IDs 728, 7E8 738 and 768. From the manufacturer’s specification, it is possible to establish the functions of these ECUs.

Data Capture on X-Analyser and PicoScope

The clusters will show which messages are associated with the same ECU. The results from two candidate vehicles are shown below.

Candidates 1 and 2 were electrically good CAN buses i.e. good earthing and less noise. The methodology used here was to plot the modal CAN-H and CAN-L values from the data segment of the CAN frame to produce the clusters shown. This modal value would be taken from the Region of the Data Field for Dominant bits only.

Utilising the Method plotting CAN_H versus CAN_L

Candidate 1

In the 1st Candidate vehicle, the diagnostic request messages were sent with the response results that plots the Electrical Signature shown in Figure 7.

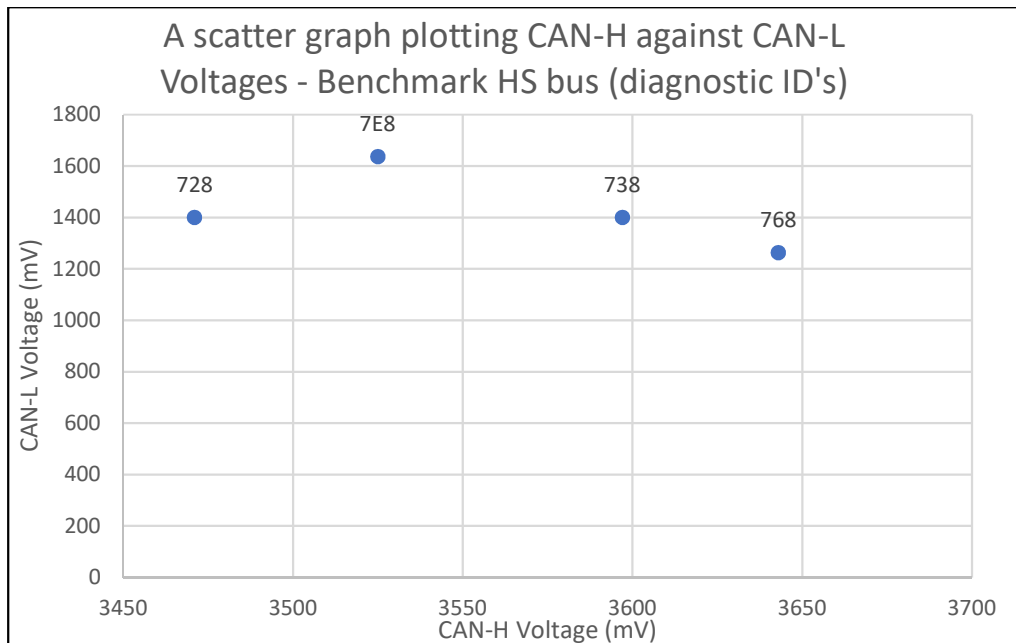


Figure 7 Cluster Plot of the Diagnostic Response CAN Messages for Vehicle Candidate 1 – CAN_H modal voltage versus CAN_L modal voltage

Here we are plotting the cluster points using CAN_H versus CAN_L. From the specification of this vehicle, the resulting diagnostic response messages are interpreted as follows:

- 728 – Instrument Cluster
- 7E8 – Engine ECU
- 738 – Steering ECU
- 768 – Brake Control Module ECU

After the Diagnostic Response Signature is established, we then collected the real-time CAN control messages and plot the Electrical Signature shown below in Figure 8.

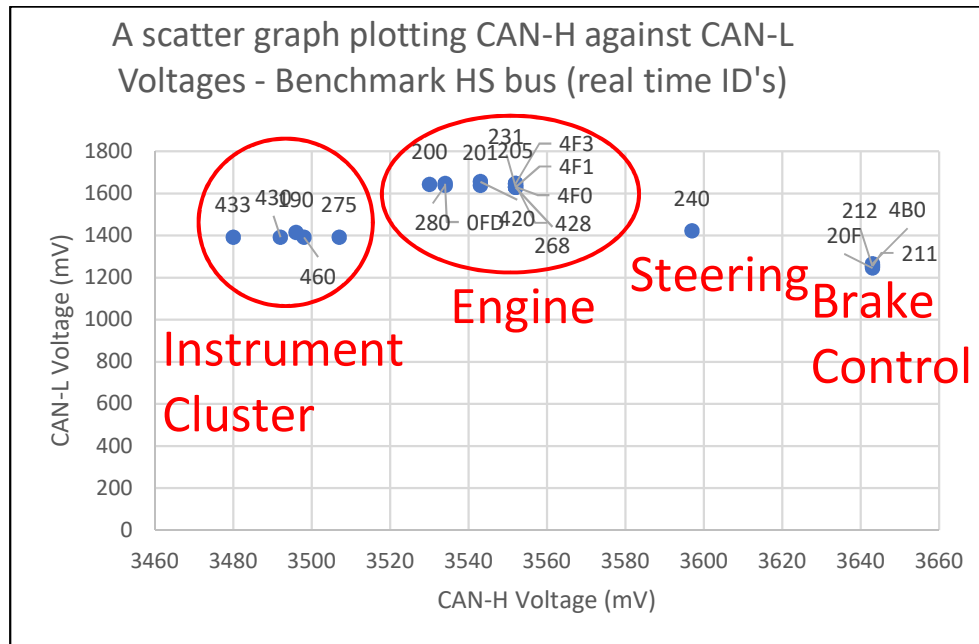


Figure 8 Cluster Plot of the Real-Time CAN Messages for Vehicle Candidate 1 – CAN_H modal voltage versus CAN_L modal voltage

Here we have established that the general Electrical Signatures of the real time CAN messages closely match up with the diagnostic response messages. Therefore, we can ascertain that the messages come from the following ECUs:

- Instrument ECU – CAN IDs 190, 275, 430, 433, 460
- Engine ECU - CAN IDs 200, 201, 205, 231, 268, 280, 420, 428, 4F0, 4F1, 4F3
- EHPAS ECU – 240
- Brake Control Module ECU – 20F, 211, 212, 4B0

This information will allow reverse engineering methods to help ascertain the functions of these CAN messages. In X-Analyzer, it is possible to isolate these messages and perform various investigation methods to determine the functions of the individual signals within these messages.

Candidate 2

To further verify the validity of this method, a similar method was performed on a 2nd candidate vehicle for which the CAN specification was available. The result is illustrated in Figure 9 below showing the Electrical Signatures of the real-time CAN data of this vehicle.

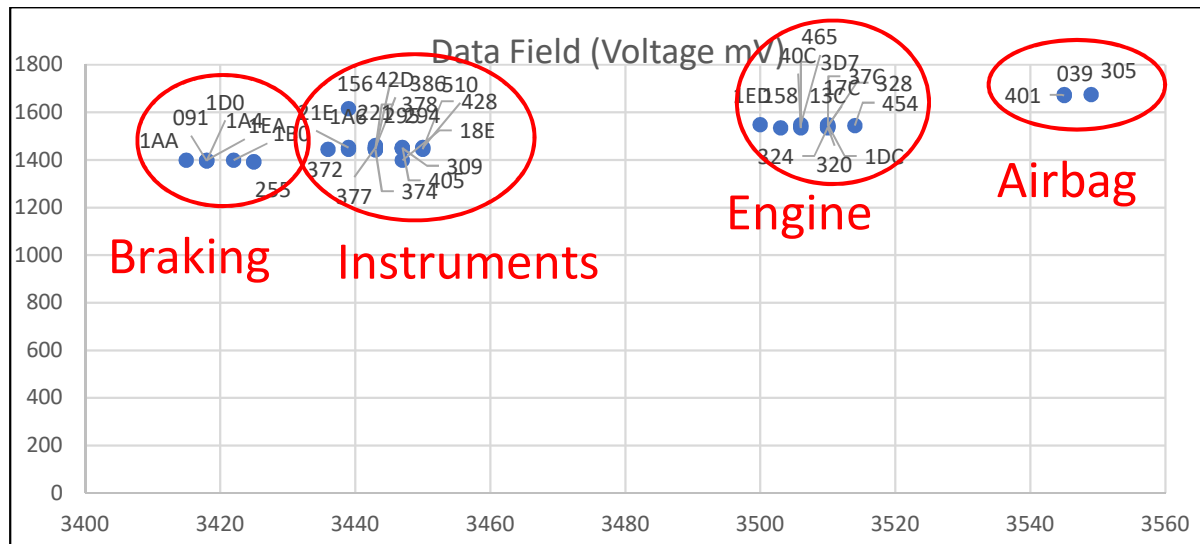


Figure 9 Cluster Plot of the Real-Time CAN Messages for Vehicle Candidate 2 – CAN_H modal voltage versus CAN_L modal voltage

Here we can observe that the messages come from the following ECUs:

- Braking ECU – CAN IDs 091, 1AA, 1A4, 1B0, 1D0, 1EA, 255
- Instrument ECU – CAN IDs 156, 18E, 1A6, 21E, 221, 294, 295, 309, 372, 374, 377, 378, 386, 405, 428, 42D 510
- Engine ECU – CAN IDs 13C, 158, 17C, 1DC, 1ED, 320, 324, 328, 376, 3D7, 40C, 454, 465
- Airbag ECU - CAN IDs 039, 305, 401

4 Summary and Conclusion

The method shown in this paper can be used as evidence to support hypotheses when reverse engineering. Many times, during reverse engineering exercises, we want to isolate CAN messages from a particular ECU. This method of plotting Electrical Signatures by noting the modal average of CAN_H versus CAN_L levels for each message data field has shown that it is a very good assistance in accomplishing this.

The approach shown in this paper is not limited to CAN bus technology. CAN-FD is the obvious next bus to look at. However, electrical signatures could be obtained for many other bus and network technologies e.g. FlexRay which uses a differential signalling approach. It may be possible to characterise the signals on a LIN bus. However, a slightly revised approach would need to be adopted for deriving an electrical signature since it does not use differential signalling.

References

1. ISO 15765-4 (2016) - Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) Part 4: Requirements for emissions-related systems