Benchmarking of CAN systems using the physical layer – car, truck and marine case studies

Dr. Chris Quigley, David Charles, Richard McLaughlin, Warwick Control Technologies

There are many reasons for a CAN system to be benchmarked or reverse engineered. An example is when full documentation unavailable and conversion to an electric powertrain is needed. A technique is described that uses an electrical signal fingerprint of a CAN message. This fingerprint is a way of associating messages to an ECU without any prior knowledge of the system. Its use is discussed in a number of case studies. In an automotive application, diagnostic responses from an ECU, whose identifiers are standardised, are matched with the unknown real-time CAN messages, so that the transmitting ECU is determined. Diagnostics parameters can then be used to discover real-time CAN signals by taking advantage of knowledge of typical automotive electronics. For example, wheel speed signals are transmitted by the braking ECU and the diagnostic parameters relating to vehicle speed can be correlated with only the braking real-time CAN messages. A similar approach is carried out on a truck based on the J1939 protocol it is typical that a significant number of messages are not standard and therefore unknown. Finally, in a marine application with little info known, electrical fingerprinting was used to confirm which ECUs were on the network.

Introduction

There are many reasons for a CAN system to be benchmarked or reverse engineered. An example is when full documentation on the CAN system is unavailable, sparse or incorrect and some engineering task needs to be carried out such as:

- conversion of a car or truck to an electric powertrain
- CAN system is exhibiting a fault that needs to be fixed
- Fitment of special controls is needed such as for a disabled driver
- Competitor product analysis

What is a CAN Message Signature?

A CAN Message Signature is something that is largely unique about any message sent by an ECU. Therefore, 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 in the CAN messages from each ECU due to a number of reasons:

- the physical makeup of the CAN bus (e.g. node position and distance on the bus).
- Components within the ECU
- Wiring characteristics and age

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 (which contains the CAN identifier) should not be considered for a CAN Message Signature based on the CAN electrical signature. This is because as there may be several ECUs communicating within this field and therefore influencing the electrical signal during the Arbitration field. Once the arbitration process is completed, there is just one ECU producing the Data Field. This is where you see unique characteristics in the electrical signal for each ECU. Therefore, 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.

To illustrate the unique electrical characteristics of each ECU in a vehicle, Figure 2 and Figure 3 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.

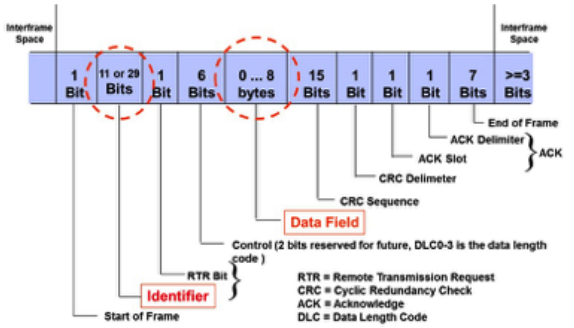

Figure 1: Construction of CAN Frame

Figure 2: ECU A - Electrical Characteristics

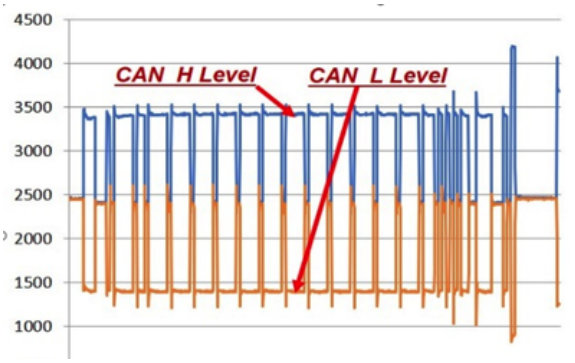

Figure 3: ECU B - Electrical Characteristics

If the latter half of each CAN frame is compared (where the data field is located), it can be seen that the CAN_H and CAN_L voltage levels are different for these two ECUs. For ECU A CAN_H sits at around 3500 millivolts and CAN_L sits around 1500 millivolts. The voltages differ for ECU B with CAN_H sitting below 3500 millivolts and CAN_L sitting below 1500 millivolts.

In this paper, this characteristic used to obtain a simple CAN Message Signature which is based on the CAN message electrical signal.

Obtaining the CAN message signature for each CAN frame

A methodology for obtaining a CAN Message Signature based on the CAN_H and CAN_L voltage levels of each CAN message is described here. This can then be used to associated CAN messages to a transmitting ECU.

The process includes the following steps:

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

The average values of CAN_H, CAN_L Data field Dominant bits is the simple CAN Message Signature that is used throughout this paper.

The following figures show how the process for obtaining the CAN Message Signatures can be achieved using some tools. The X-Analyser with the PicoScope PC oscilloscope connected can be used to scan a CAN bus and grab the CAN_/CAN_L electrical signals for each CAN identifier that is on the network. Figure 4 shows the CAN frames are logged on the top half of the display. Each CAN frame can be 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).

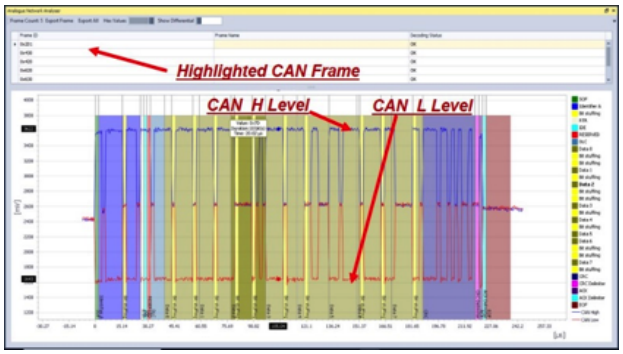

Figure 4: Highlighting a CAN Frame within a PicoScope Display

Each of 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	1CFFBC00		0	2.264	2.217	SOF	
3	DLC	8		0.144000009	2.303	2.257	SOF	
4	Data	C4 FB FF FF 0B 27 CB 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 includes:

- Frame ID (Hexadecimal)
- DLC
- Data Field
- Samples per Second
- Exported On (Date)
- Time of sample
- CAN-H and CAN-L Voltages
- Region of the CAN Frame the sample is from e.g. Data Field

Once this information is exported to Excel, post-processing is carried out to obtain the average of CAN_H and CAN_L voltages from the Data field Dominant bits. The single average voltage for CAN_H and CAN_L is a simple CAN Message Signature for that CAN message that is used for the remainder of paper in a number of CAN bus applications.

Using cluster plots to visualise CAN message signatures

As previously described, the CAN Message Signature is a pair of voltage readings; one each for CAN_H and CAN_L. These can be put onto a cluster plot so that the clustering of CAN messages transmitted by a particular ECU can be observed.

An example of this is illustrated in Figure 6 showing the cluster plot of CAN message signatures of the real-time CAN data of a passenger car.

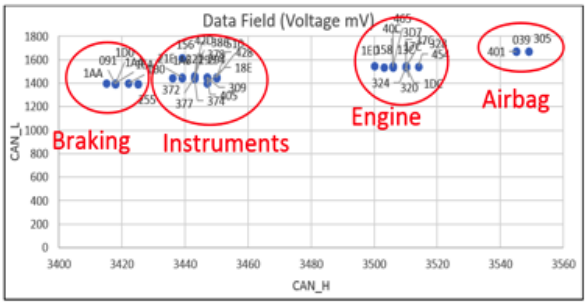

Figure 6 Cluster Plot of the Powertrain/ Chassis CAN Message Signatures for a passenger car

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

Therefore, the clustering of CAN message signatures can be used to help uncover which ECU is transmitting particular messages. The case studies below illustrate the use of this method in a number of applications allowing an engineer to solve a number of commonly experienced problems.

Case study 1 – Automotive reverse engineering using diagnostics

In the automotive industry, the real-time control CAN messages are proprietary and unknown. However, the identifiers of diagnostic messages used in manufacturing and service garages is standardised in specifications such as ISO15765 [1] and/or across an automotive manufacturer.

It is well known by automotive CAN and diagnostic engineers 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. This knowledge can be used to get the CAN Message Signature of the diagnostic response and use this to help identify which real-time CAN messages are transmitted by the engine controller. The diagnostic CAN identifiers for other ECUs are also usually standardised and therefore this information can be used to understand a CAN system better. If some kind of reverse engineering activity is taking place, one problem can be that there can be tens or even hundreds of CAN messages on a CAN bus. Therefore, identifying which ECU is transmitting each CAN message helps narrow down the search for CAN signals.

The summary of the methodology is described by the following steps:

- Send diagnostic requests for all ECUs on the network
- Record one oscilloscope trace for each CAN message on the network (diagnostic responses and real-time CAN messages)
- Calculate the CAN Message Signatures for each CAN message
- Visualise the data by plotting each CAN Message Signature and observe the clusters of data

Equipment required

Figure 7 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. The Kvaser CAN Interface is used to generate Diagnostic Request messages, and the PicoScope is used to receive the electrical signal of the Diagnostic Response. The electrical signal of the Diagnostic Response is then converted into a CAN Message Signature.

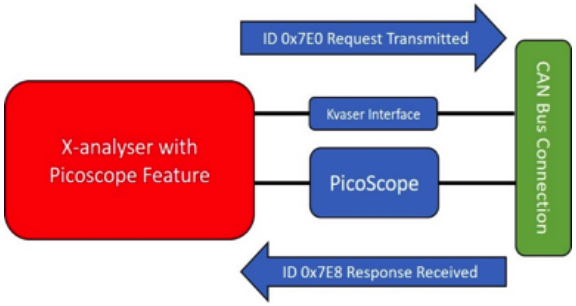

Figure 7 Equipment setup for CAN ID clustering capture

Diagnostic requests and response CAN identifiers

In this automotive example, more information about the diagnostic request can be found in ISO 15765-4:2016 [1]. The standard emission Diagnostic Request message is known to be CAN identifier 0x7E0 and the expected response from the ECM (Engine Control Module) is identifier 0x7E8. Referring to ISO 15765-4:2016, page 29, it also known that the TCM (Transmission Control Module) Diagnostic Request CAN identifier is 0x7E1, and the response message is on identifier 0x7E9. 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 0x7E2 and a response of 0x7EA. A diagnostic response's CAN identifier will increase in value by 8 and give the response i.e.

If there is no response to other requests, it means that this diagnostic function is not supported in this vehicle. Using this approach on the vehicle in question, it was ascertained that there were responses on CAN identifiers 0x728, 0x7E8, 0x738 and 0x768. From the manufacturer's specification, it is possible to establish the functions of these ECUs.

Visualisation of Collected Data

As described previously the plotting of each CAN Message Signature and then observing clusters of data, indicates which messages are transmitted by the same ECU. An example from a simple passenger car is shown below. Figure 8 shows the plot of CAN Message Signatures for the diagnostic responses.

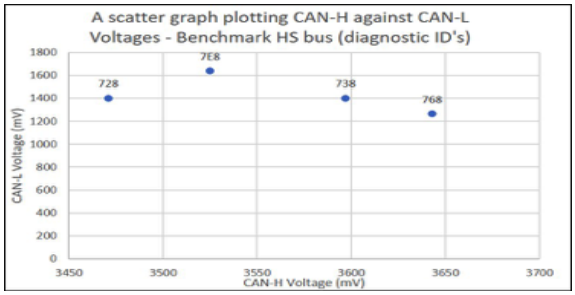

Figure 8 Cluster Plot of the Diagnostic Response CAN Messages for Vehicle Candidate 1 – CAN_H voltage versus CAN_L voltage

From the specification of this vehicle, the resulting diagnostic response messages are interpreted as follows:

- 0x728 – Instrument Cluster
- 0x7E8 – Engine ECU
- 0x738 – Steering ECU
- 0x768 – Brake Control Module ECU

Figure 9 shows the plot of CAN Messages Signatures for the unknown real-time CAN messages. By matching the location of the diagnostic response CAN Message Signatures with the clusters of unknown real-time CAN messages, the transmitting ECU can be ascertained. For example, diagnostic response 7E8 indicates that the second cluster from the left is for CAN messages transmitted by the engine ECU.

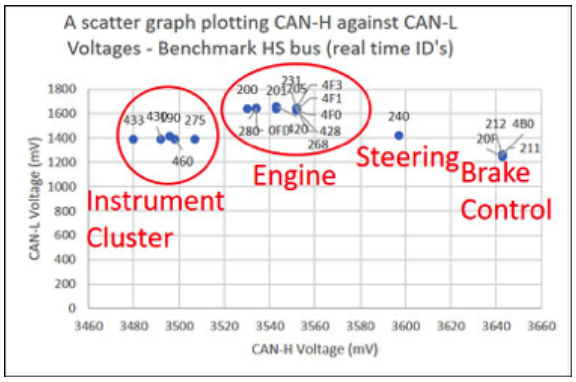

Figure 9 Cluster Plot of the Real-Time CAN Messages Signatures

Case Study 2 – Associating unknown/ undocumented CAN identifiers to transmitting ECU

A second case study is concerned when there are unknown and undocumented CAN messages on a CAN network and therefore it is unknown which ECU is transmitting. It might be that this is due to poor documentation or a bug in ECU software. By obtaining CAN Message Signatures of each CAN message on the network and then visualising the clusters of data on a plot can help indicate which ECU is transmitting the undocumented CAN messages. Once this has been done, the supplier of the device or ECU can be questioned about the findings. This approach would be similar to the plot shown in Figure 9. A cluster of messages indicate that they originate from the same ECU.

Case Study 3 - Clustering of message signatures to check for rogue ECUs on J1939/NMEA2000 type network

Another use for the previously described CAN Message Signature clustering methodology is in advanced fault finding of ECUs that are poorly configured or put onto a network which they should not be on. For example, it is required by the J1939 and NMEA2000 protocols to support an address claim process that results in each ECU having a unique Source Address [2]. If a system is poorly designed, the situation can arise in that two ECUs use the same Source Address. This is a practice that is illegal in these protocols.

The CAN Message Signatures can be used to check for duplicate source addresses in a J1939 or NMEA2000 system, which could be due to a poorly configured or rogue ECU.

During testing of such a network, an electrical signal analysis was performed, the CAN Message Signatures were generated and plotted for visualization. After reviewing the plot, it was noticed that there were two nodes with the same Source Address.

Figure 10 shows this plot. Since this is a NMEA2000 network, the last byte of the CAN identifier is the source address (i.e. for CAN identifier 9F20109, the Source Address is 0x09). For this particular network, each ECU should have been self-configurable and be able to dynamically change its Source Address. This is a type of plug and play feature of the NMEA2000 protocol that is mandatory.

CAN messages with identifiers 9F20109 and 19F50309 are messages from one sensor node. CAN message identifier 19F21109 is from a different device. The CAN Message Signatures of these three CAN messages can be seen on the cluster plot and the two are in completely different locations in the plot which indicates that the messages come from two distinct devices.

Messages with other Source Addresses are nicely clustered and close to each other (e.g. 03, 07, 30, 2C etc.), indicating that the messages are transmitted by the same device.

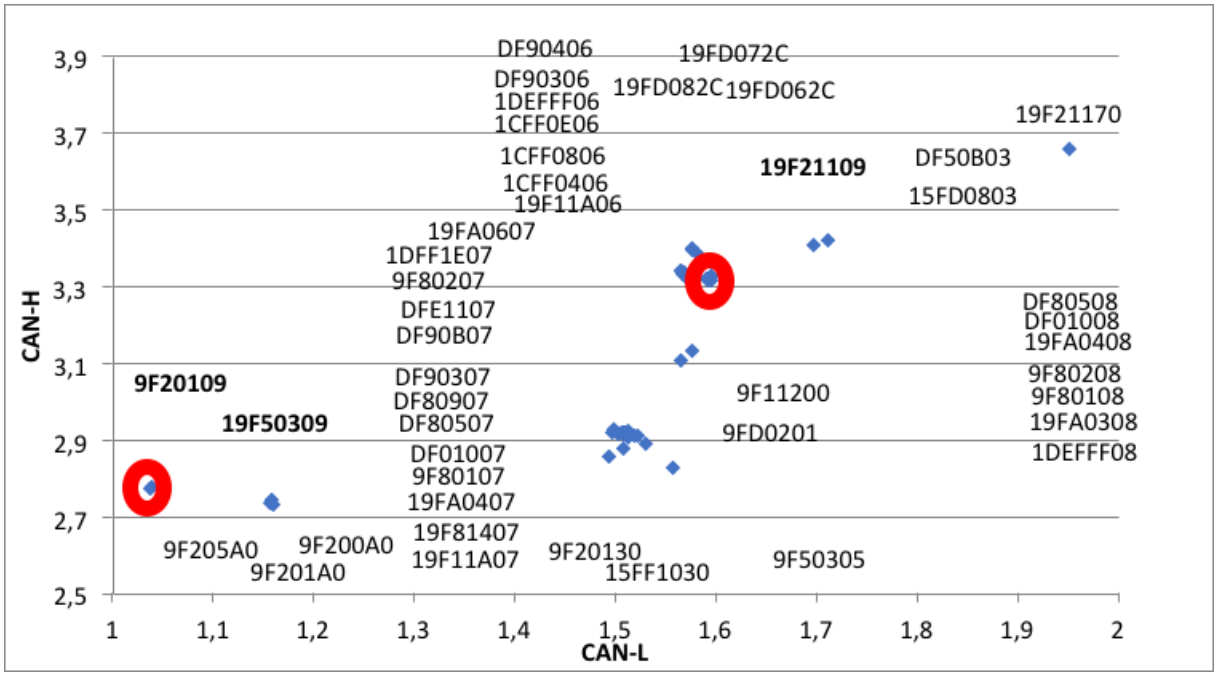

Figure 10: Cluster Plot of the Real-Time CAN Messages Signatures for NMEA2000 network

Upon further investigation of the two devices with Source Address 0x09, it was discovered that one of the devices was hard programmed with the source address of 0x09.

Case Study 4 – Identification of CAN architecture in a marine application

The next case study is a case in which there is a lack of documentation on the CAN system and CAN Message Signatures are used to help this out. This is situation that is common in the marine industry because it is often the case that each boat is a one-off or extremely small series run.

A racing yacht was investigated. It was based on a seven CAN bus system. There were four CAN buses based on a proprietary CAN protocol used for lithium battery management. A J1939-based system had a generator that was used to recharge the lithium batteries. A CANopen-based hydraulic system was used for the yacht's sail controls. Finally, a NMEA2000 system contained numerous devices for the yacht's navigation. A reported CAN bus system crash resulted in loss of sail control and the cause was to be investigated. The technical documentation for the yacht was very good in some areas and weaker in others. For example, information on the CAN architecture of the battery management system was not available.

During a review of the multiple CAN buses of the yacht with the chief engineer, he perceived that the Battery Management System (BMS) was configured as shown in Figure 11. The BMS was comprised of an Internal CAN bus and an External CAN bus. Various data was logged from the CAN buses including that necessary to obtain CAN message signatures.

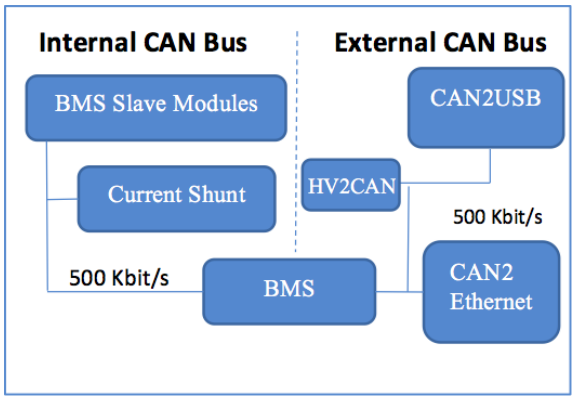

Figure 11: Perceived CAN Bus Architecture of BMS

Figure 12 shows the plot of CAN Message Signatures for the Internal and External CAN buses. These were generated from data collected from these yacht CAN buses.

Both buses should have a few devices on them transmitting CAN messages. However, the External CAN bus is showing only one cluster. This led to a further investigation and tear-down of the system. The single cluster on the shown in the plot for the External CAN Bus indicates that perhaps only one device is transmitting CAN messages. The further investigation proved that this was the case and the architecture was in fact as shown in Figure 13. The only device transmitting was that marked BMS and this discovery was led to by the plotting of CAN Message Signatures. The HV2CAN device was in fact on the Internal CAN Bus and not the External CAN bus.

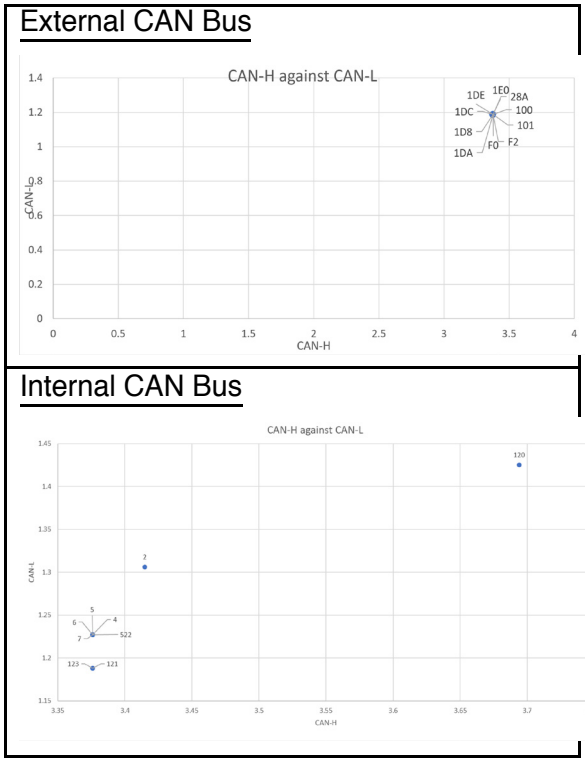

Figure 12: Plots of CAN message signatures of marine CAN buses

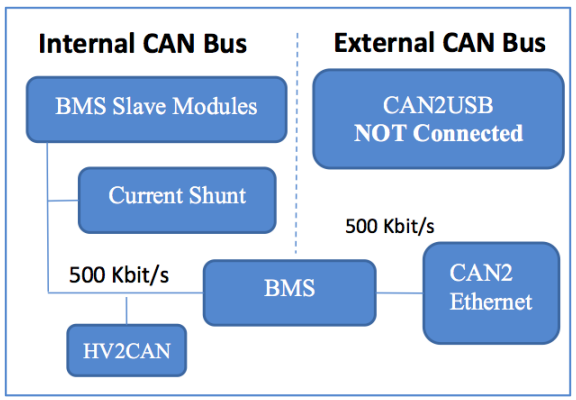

Figure 13: Actual CAN bus architecture

Summary and Conclusion

The method for obtaining simple CAN Message Signatures shown in this paper can be used as evidence to support hypotheses when carrying out a number of engineering and testing activities on a CAN bus system.

It has been shown that the CAN Message Signature can be used to help identify which device is transmitting a particular CAN message of a certain identifier. The method shown in this article can be used as evidence to support hypotheses when reverse engineering a CAN bus architecture or carrying out advanced fault finding. Many times, during reverse engineering exercises, we want to isolate CAN messages from a particular ECU. This method of plotting CAN Message Signatures which are based on the average of CAN_H versus CAN_L levels for each message data field has shown that it is a very good methodology in accomplishing this.

A number of case studies have briefly described how to use the clustering and visualisation of CAN Message Signatures for advanced CAN bus system testing and analysis.

The approaches shown in this paper have been on CAN. However, it could potentially be applied to other network technologies using differential signalling such as CAN-FD or FlexRay.

Definitions/Abbreviations

ABS	Anti-skid Braking System
BMS	Battery Management System
CAN	Controller Area Network
CAN_H	CAN High
CAN_L	CAN Low
CAN ID	CAN Identifier
ECM	Engine Control Module
ECU	Electronic Control Unit
OBD	On Board Diagnostics
TCM	Transmission Control Module

Dr. Chris Quigley
Warwick Control Technologies Ltd.
Unit 8 Ladbroke Park
Millers Road
GB-CV34 5AN Warwick
www.warwickcontrol.com

David Charles
Warwick Control Technologies Ltd.
Unit 8 Ladbroke Park
Millers Road
GB-CV34 5AN Warwick
www.warwickcontrol.com

Richard McLaughlin
Warwick Control Technologies Ltd.
Unit 8 Ladbroke Park
Millers Road
GB-CV34 5AN Warwick
www.warwickcontrol.com

References

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

[2] J1939-81 Network Management