DeviceNet Interoperability Testing

Introduction

The term 'interoperability' has been mentioned in various electronic communications fields. In short it represents the "ability to operate between parties". 'Interoperability' is sometimes defined as "how data is effectively exchanged and utilised between devices". This statement reflects only the situation in the Application Layer. Interoperability should be the concern of all layers. In DeviceNet (and other protocols), an ideal and preferred system should contain products that are 100% interoperable with each other.

DeviceNet, a well proven CAN higher layer protocol, is said to fulfil three out of the ISO/OSI 7-layer model stack. It utilises the existing CAN protocol for its Data Link Layer and Physical Layer, and defines a powerful software protocol for its Application Layer.

Virtually, the Application Layer can be broken down into three sub-layers, i.e. the DeviceNet Application Layer, DeviceNet Session Layer, and DeviceNet Transport Layer, as seen in Figure 1. Although the names are identical to the ISO/OSI 7-layer model stack, these three 'virtual' layers do not implement all full features defined by the three layers in the ISO/OSI 7-layer model, instead only partial features are utilised by these three virtual layers. They are added merely to expand the discussion on the DeviceNet Protocol.

The three virtual Layers can be grouped together under a single "Software Layer", and the Data-Link Layer as well as the Physical Layer can be grouped under a single "Hardware Layer".

As can be seen in Figure 1, the Network Layer and the Presentation Layer in the ISO/OSI 7-layer model stack are eliminated in the DeviceNet protocol stack, since there exists only one network protocol. With the implementation of these five layers, DeviceNet is able to provide a highly reliable communication standard.

Similar to other systems that fulfil the ISO/OSI model, the modelling technique adopted by DeviceNet allows higher stacks to communicate with each other without having to know in detail the underlying stacks (end-to-end communication), i.e. they are communicating with each other via a 'virtual route', as seen in Figure 1. The term 'Interoperability' then represents the ability/effectiveness in exchanging data via these virtual routes.

Figure 1 The Virtual DeviceNet Protocol stack
DeviceNet Interoperability Issue

The DeviceNet protocol conformance test plays a vital role in the interoperability test, it verifies the conformance of the software implementation against the specification. It can therefore be confidently said that any DeviceNet devices that pass the conformance test will be guaranteed a certain level (high) of the interoperability. To filter out any problems caused by the incorrect implementation (non-conformance device), the basic requirement of the interoperability test is that the device must first pass the protocol conformance test.

The following sections identifies some important interoperability issues at different layers.

Software Layer Interoperability

DeviceNet Application Layer Interoperability

In the Application Layer, interoperability is simply a measure of the effectiveness of a device in generating/utilising (produce/consume) data. If the data is not effectively utilised, the device may fail to interoperate with others, even though this device passes the protocol conformance test.

For example, the DeviceNet protocol inherently implies, in theory, the possibility of transmitting unlimited number of fragmented message. Since the DeviceNet protocol conformance test engine does not (and cannot) restrict on the amount of I/O data produced/consumed by a device, improper handling of these data may flood the network with large amount of unnecessary data, preventing other messages to get through the bus.

Although experienced system integrators may avoid setting devices other than a DeviceNet master to be MAC ID zero, the DeviceNet protocol does not explicitly restrict any devices to use MAC ID zero. This may lead to an interoperability problem at the application layer.

To illustrate, consider a UCMM capable device with MAC ID 0, which has 256 input bytes, is configured as a producer, and utilising Group 1 message ID 0 (i.e. connection ID, or CAN ID, of 000Hex) for its I/O data producing. Assuming this particular device utilises the transport Class 0 Cyclic Connection with a cycle time of 1 ms, which is the minimum possible resolution. The large amount of I/O data produced not only increases the busload and reduces the bandwidth of the network, but also increase the latency of the messages transmitted by all other nodes, including the master, due to its high bus access priority. Worst still, these nodes may transition to the time-out state (see the later section).

Devices that allow large amounts of I/O data will then need to be tested with the largest possible I/O data size and the cycle time, either the cycle time of the slave’s COS/cyclic connection, or the DeviceNet master’s scan time.

DeviceNet Session Layer Interoperability

As defined in ISO 8327, the Session Layer takes care of the connection establishment. This includes:

1. Open, delete, and maintain various Connections. DeviceNet defines two messages, namely Explicit Message and I/O Message. I/O Message is further divided into Polled I/O, Bit-Strobe I/O, Change-of-State (Event-trigger) I/O, and

---

1 Busload is a measure of the number of messages over a time period. An increase in busload represents the increase in the number of messages per time.
Cyclic I/O. The Session Layer is responsible to process the connection request from a remote node.

2. The connection timer. The Session Layer in charge of maintaining various timer parameters associated with the connection states. This includes the Production Inhibit Time, Expected-Packet-Rate, etc.

As a norm in real-time control systems, every connection has a connection time associated to it. If an expected message is not received within a predefined time frame, the system should react to this situation, such as transition to the connection time-out state.

The DeviceNet Protocol Conformance Test engine has already verified this behaviour. The interoperability issue at this layer is: What is the proper connection time to set for the 'end-to-end' communication?

An example is the communication between a tool and a Group 2 only slave device which has already been owned by its master, as shown in Figure 2. As defined by DeviceNet, this (Group 2) slave will only respond to its master, and the master will proxy the slave responding to any third party request destined to its slave. The tool has no idea that the connection is actually proxy-ed by the master, i.e. the actual response time is double of the expected one. Therefore it may set a minimum connection time that "should be sufficient" for the request/response message. There may be no problem in a network with low busload. But in a heavily loaded network (perhaps with multi-masters), it may lead to the connection time-out problem due to the lose in the network access arbitration.

![Figure 2 DeviceNet Master proxy the slave responding to the tool](image)

**DeviceNet Transport Layer Interoperability**

Transport Layer is responsible for transporting the data from the lower layer (in this case, Data Link Layer) to the higher layer (Session Layer), or vice versa. One of the major tasks of this layer is to ensure the data received/transmitted is in the correct sequence, particularly the fragmented data. When a fragmented data package is being received, this layer ensures the complete data is received and reassembled correctly before sending it to the higher layer, otherwise the whole message should be discarded.

An example situation experienced by the author is that when a device (MAC ID 41) receives an explicit request message from a tool while it is communicating with its master (MAC ID 61). Occasionally this node floods the network by repeatedly transmitting its I/O data (at 100µs interval), causing all other nodes to transition to the time-out state since they fail to receive the I/O command messages (poll/bit-strobe) from the Master within the time frame. This is due to the internal logic error (timer/receive flag) of the device's firmware at a particular instance while it is processing the received I/O command.

To test this, the device needs to be interrupted with very high frequency. It is not possible for the DeviceNet Protocol Conformance Test engine to repeatedly interrupt the device at different instances, particularly due to the timer resolution of the
platform on which the test engine resides (Windows™). To correctly carry out this test, the platform should have a resolution of 10µs or better, which is almost impossible on a Windows™ platform. Therefore the interoperability test engine is required.

**Hardware Interoperability**

**Data Link Layer Interoperability**

This layer's interoperability deals with the data encapsulation, acknowledgement, fault detection and confinement, etc. A well-known example is the double-receiving problem that occurs on the receiver side. As defined in the Bosch CAN specification, a CAN message is considered valid to the transmitter if no errors occur until the last bit of the End-Of-Frame. However to the receiver, it is valid if no errors occur until the last but one bit of the End-Of-Frame. In this case, if the last bit of EOF fails, the transmitter will interpret this as an error whilst the receiver will accept it as a valid message. Hence the transmitter will perform the retransmission, causing the receiver to receive twice of the same message. Since this is the unavoidable problem, the number of occurrence of this phenomenon should be investigated, and this is one of the task performed by this research group at the University of Warwick.

**Physical Layer Interoperability**

This layer’s interoperability deals with the bit timing, synchronisation and bit encoding/decoding. Although the Data Link Layer and the Physical Layer have been defined in the ISO 11898 standard, and the interoperability tests of these two layers should have already been carried out by the relative laboratories. However this does not mean the test can be neglected in the DeviceNet protocol. This is because DeviceNet has more tightened specification than ISO 11898, which in turn has more tightened specification than Bosch CAN specification. Figure 3 illustrates the Physical Layer relationship between the Bosch CAN Specification, the ISO 11898 standard and the DeviceNet protocol.

![Figure 3 The Physical Layer relationship between the Bosch CAN Specification, the ISO 11898 standard and the DeviceNet protocol](image)

An example is that the Bosch CAN Specification leaves the definition of CAN voltage representations open, i.e. it does not specify the voltage ranges for CAN_H and CAN_L. In ISO 11898, this voltage representation is specified. Also, ISO 11898 defines the valid network baudrate ranges from 10Kbps to 1Mbps, DeviceNet defines only three baudrate, i.e. 125Kbps, 250Kbps and 500kbps. In addition, the DeviceNet protocol specifies the sampling point of more than 80%, which shows another example of the tightened specification than the ISO 11898 protocol.

Another interoperability problem is that, the DeviceNet protocol specification does not define the minimum time the device should 'stay alive' during the power interruption. For instance, consider the network power is interrupted for 50ms. If the
power supply circuit of the DeviceNet master does not have sufficient capacitance to hold the power during the interruption, the DeviceNet Master will experience a power cycle reset, hence the connections between the master and its slaves are lost. If the network power is interrupted frequently, the master will do nothing but busy open/release the connections with its slaves.

Interoperability problems could also occur during the plug-and-play operation, where the device is being hot-swapped on a live network. Since the DeviceNet protocol specifies the necessity of drawing power from the network, incorrect implementation of the power supply section will likely bring down the whole network.

Although some of the physical layer problems should have been covered by the Physical Layer Conformance Testing, however since the DeviceNet Physical Layer Conformance Test Suite has not been fully developed and implemented, much of the work must be done in the interoperability test.

**Other Interoperability Issues**

**Version Compatibility**

All software inherently contain bugs, the DeviceNet protocol conformance software is no exception. These bugs may allow the device under test (DUT) to bypass some tests, hence there are compatibility problems between different versions. Although in theory any enhancements done on the DeviceNet specification should not affect the behaviour of a device, devices passed different versions of the conformance test may not work correctly with devices passed the later specification, due to the bugs.

**Hardware-Layer Only Devices**

Devices such as bridges, repeaters, analysers, network monitors, do not normally implement software layer within them, they implement only the hardware layer (see Figure 1). Therefore the Protocol Conformance Testing is not applicable to these devices. To ensure these devices conform to the DeviceNet specification, the Interoperability testing, as well as the Physical Layer Conformance Testing, should be applied to these devices.

For example, a bridge/repeater should, in theory, transport CAN messages from one network to the other without changing the order/sequence of the messages. Failure to do so may lead to serious problems. Some FullCAN chips implement internal arbitration within the message objects, which may change the sequence of the transmitted messages. Incorrect implementation of the firmware may then lead to interoperability problems. In addition, bridges/repeaters inherently introduce delay. This delay has to be carefully investigated to ensure it does not impact the performance of the network significantly.

**Interoperability Testing**

To systematically run the interoperability test, a bottom-up approach is adopted by the research group at the University of Warwick, i.e. the lowest layer is first tested, followed by the upper layers. This ensures any problems identified are not due to the bugs contained in the underlying stacks, thus ease the analysis and troubleshooting. Therefore the typical procedure exercised at the University of Warwick is:

1. Verify the Conformance of the Hardware (Physical and Data Link) Layer.²

² Although the Physical Layer Conformance Test procedure has not been fully formalised by the ODVA, the test procedure established by the University of Warwick Conformance Test Lab has been helping the vendors to identify the Physical Layer Conformance (and hence the interoperability problem) of their devices.

3. Run the Interoperability test.

The interoperability test suggested and carried out by the research group at the University of Warwick is divided into three sub-tests, i.e. Network Test, Point-to-point Test, and Power Test.

**Network Test**

The Network Test aims to rectify the practical problems, i.e. those problems that could likely be encountered by the system integrators and the end users. This test covers all layers.

![Figure 4 Block Diagram showing the construction of interoperability test board](image)

To run the network test, one of the methods, which has been developed by the research group at the University of Warwick, is to connect the DUT onto a specially constructed DeviceNet Interoperability test board. This particular test board represents the worst case scenario, i.e. it consists of maximum of 64 nodes (including the DUT) with different product types from different vendors, maximum baud rate of 500K bps, maximum trunk cable length of 100 metres (for 500K bps), maximum cumulative drop budget of 39 metres (for 500K bps), different topologies (i.e. bus, daisy-chain, zero drops etc.), different connections (i.e. Explicit, Polled I/O, Bit-Strobe I/O, Cyclic I/O, Change of State I/O) with a bus load of 35%.

The DeviceNet master (scanner) is sitting at the far end of a daisy-chained drop cable that is connected at one end of a trunk. Refer to Figure 4 for illustration. The purpose of connecting the DeviceNet master (scanner) at one end and the device under test (DUT) at the other far end is to give the maximum propagation delay between these two nodes.

Other tools such as CANalyser™, CANScope™, Digital Oscilloscope, DeviceNet Monitor™, DeviceNet Manager™, DeviceNet Utilities⁴, DeviceNet busload meter, etc. are used for the data capture and analysis.

With this setup, many problems can be rectified, such as the inconsistency in the bit timing parameters (Physical Layer). For instance, if the sampling point of the DUT is

---

³ “Bus Load” indicates the percentage utilisation of the network.

⁴ ‘DeviceNet Utilities’ is a dedicated PC based DeviceNet monitor tool developed by the research group at the University of Warwick.
not set to the value as defined in the DeviceNet Specification (80% or more), it will cause the transmission of error frames on the network, and causing some devices to suspend to the bus-off state.

Other problems such as unexpected time out, incorrect I/O data, etc. will also be logged and analysed.

Since this test board gives the worst case scenario, we should be confident to say that any DeviceNet product that passes the worst case scenario test will be able to work on any practical DeviceNet networks that are not normally (practically) driven to these boundaries.

The typical slave explicit response time is shown in Figure 5. The average response time is 580µs. The quickest response time was recorded as 400µs. However as can be seen from the graph that, the response time can take as long as 2.2ms. This is due not only to the arbitration on the bus, but also the internal process delay.

![Typical Slave's Explicit Response Time](image)

**Figure 5 Typical explicit response time of a DeviceNet Slave**

The graph in Figure 6 shows the typical Poll response time of a DeviceNet Slave. As can be seen from the graph, the average response time is 320µs. The best (quickest) response time was recorded as 200µs, and the worst (slowest) response time observed is 700µs.

Since DeviceNet specifies that the minimum timer resolution of 1 ms, which correspond to the minimum time-out of 4ms, these values are perfectly acceptable. Nevertheless, any value falls outside 300% of the average value (in this case 1ms) represents the possibility of the latency problem, and should be analysed.

As compared to the result obtained in Figure 5, it can be clearly seen that the response time of the I/O message experiences less delay and has less fluctuation, i.e. it has higher consistency. This is the preferred result as I/O messages are more time critical than explicit messages.

---

5 This data was obtained from the interoperability test board with bus load of 35%, by using CANalyser™, and processed by using Microsoft® Excel™.
Figure 6 Typical polled I/O response time of a DeviceNet Slave

Point-to-point Test

The Point-to-point test aims to test the area that is not covered by the Network Test, including the generation of invalid DeviceNet message (such as remote frame, extended frame, etc.), error frames, unused data frame, etc. in high frequency, i.e. the network is heavily loaded.

In the point-to-point test, the DUT are connected to a specially constructed test jig, which is controlled by the computer. A resistor of 600 ohms and a capacitor of 2nF is used to load the network, emulating the worst case impedance of 64 nodes, as shown in Figure 7.

Figure 7 The Point-to-point Interoperability Test Jig Setup

To run the test, the DUT is mapped to the DeviceNet Master so that it works normally. The Error Frames Injector\(^6\) (EFI) is used to test the behaviour of the slave when it is forced to the bus-off state. This is particular useful if the DUT supports the off-line communication.

The CAN Frames Generator is used to randomly generate any possible CAN messages to the network. The frequency of the injection can be varied from low (every 100ms) to high (every 100us). In this case the busload can go up to 90%. This provides good data to the analysis of the interoperability, such as:

\[^6\] The EFI unit is developed by the research group at the University of Warwick. It is capable of detecting a message frame of a particular connection from a particular node and inserts error frames to this particular message frame to zap this device to the bus-off state.
1. **Stress.** The message injected on the network with high frequency can test the ability of the DUT in handling the high interrupt rate. For example, devices that support I/O Bit-Strobe connection, such as a Photoelectric sensor, must accept all messages at the Data Link Layer, and process it at the Software Layer. If the bus load is high (i.e. high message rate), the microcontroller within the device may have no time to process other routines but services only the interrupts. For instance, the worst case interrupt rate at 500K baud is reckoned to be 1 message per every 96\(\mu\)s. Consider a microcontroller with system clock of 1\(\mu\)s. The average instruction time is 2.5 cycles, i.e. 2.5\(\mu\)s. This means that only 96/2.5 = 38 instructions can be processed before the next interrupt takes place. Failure to do so may lead to the lost of messages. If the DUT cannot handle this rate of interrupt, then the DUT will be suggested to reduce the supported baud rate down to 250Kbps or 125Kbps.

2. **Invalid message Handling.** Invalid messages include the CAN error frames, CAN identifier greater than 7C0hex, CAN extended frame, unused/reserved bits. In theory these messages should not affect the device. However, for instance, if a device uses a CAN controller that adheres to CAN Version 2.0A, it will interpret the extended frame as an error. Although DeviceNet protocol does not allow the extended CAN frame to be exerted on the network, using Version 2.0B allows the future development/enhancement.

In general, there should not be any connection failures (due to the message latency) between the master and the slaves at the bus load of 35%. However as the bus load increases, we should expect the increase in the latency too, see Figure 3. Analysis is carried out to investigate the behaviour and the capability of the device working on a highly loaded DeviceNet network.

### Power Test

The Power Test aims to investigate the behaviour/performance of the DUT during the power interruption/cycle. In this test, the Network power is cycled several time. Power cycle generates power interruptions. These interruptions can be used to test the capacity of the DUT in holding the power within the interrupted period. There is no pass/fail criteria issued on this test, however devices are recommended to provide larger storage for the power whenever possible.

This test is still under investigation by the research group at the University of Warwick, and there is no significant result obtained at the time this paper was submitted.

### Conclusion

DeviceNet Interoperability Test requires long-hours of running on the test board/jig and constant monitoring of the network parameters. In most of the time, unlike the Protocol Conformance Testing, there is no clear card on the pass/fail criteria. In this case, the analysed results can only be used to suggest or comment on the potential bug/instability of the DUT.

Many high-end/dedicated tools are required for this research and investigation. In addition, devices with multiple device profiles from multiple vendors with different versions are required. Continuously updating the devices is also necessary. This group is appealing to companies to contribute their devices to the test labs, so that further research can be carried out to improve the performance of the testing, hence the reliability of the products.

Although the test procedure has not been officially formalised by the ODVA (Open DeviceNet Vendor Association), at the time this paper was submitted, this procedure proved to be valuable and useful to identify most interoperability issues. Work is still
underway to formalise the test procedure by the research group at the University of Warwick.

References