CAN Bus Performance Analysis of DeviceNet I/O Connection Services

R.T. McLaughlin & Eva D'souza – University of Warwick
Dr. K.H. Tang – Warwick Control Technologies

DeviceNet utilises a variety of possible I/O connection services that allow for deterministic optimisation of CAN bus load performance. This paper reviews the theory of bandwidth capability of contention-based communication schemes such as CAN and Ethernet. It studies the DeviceNet I/O connections for efficient use of CAN busload to minimise any possible message latencies due to high use of the bus bandwidth. It also shows the use of message prioritisation to ensure throughput of high priority information.

Considered in this paper are the various combinations of DeviceNet I/O message services: [1]

- Polling
- Strobing
- Change-of-State (COS)
- Cyclic
- Multicast

Several combinations of these message services are implemented to show the differences in bus bandwidth. This information is essential to system integrators and end-users to ensure an optimised high-performance control system.

DeviceNet, a CAN Layer 7 (Application Layer) Protocol, has been widely adopted as an industrial standard. DeviceNet uses connection-based communication method, which offers high reliability and stability. It defines two types of messaging: Explicit Messaging and I/O Messaging.

Explicit messages provide multi-purpose, point-to-point communication paths between two devices. They provide the typical request/response-oriented network communications used to perform node configuration and diagnostics. Explicit messages typically use low priority CAN identifiers and contain the specific meaning of the message right in the data field. This includes the service to be performed and the specific object attribute address.

DeviceNet I/O messages are for time-critical, control-oriented data. They provide a dedicated, special-purpose communication path between a producing application and one or more consuming applications. I/O messages use the high priority CAN identifiers. For this reason, it is critical to consider the busload performance to ensure minimum latency of data on a contention-based data link such as CAN.

Figure 1 illustrates the DeviceNet group messages through the utilisation of the CAN 11-bit identifiers. Recalling the basic CAN identifier rules, the lowest identifiers have the highest priority for bus contention [2]. Here it can be seen that Message Group 1 is the highest priority (msb = 0 or dominant), and its Message ID determines the message priority within that group.

Message Group 2 has the next highest priority, with Message Group 3 next in priority and Message Group 4 lowest in priority.

I/O messages generally fall within Groups 1 & 2, and Explicit messages generally fall within Groups 2 & 3. There are a few exceptions to this rule that will not be necessary to cover for the purposes of this paper. Group 4 messages are used for off-line communications. In this paper, I/O messages are only considered for the purposes of analysing message response performance in relation to busload utilisation.

CAN INDENTIFIER BITS											HEX RANGE	IDENTITY USAGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						000-3ff	Message Group 1
1	0	MAC ID						Group 2 Message ID			400-5ff	Message Group 2
1	1	Group 3 Message ID			Source MAC ID						600-7bf	Message Group 3
1	1	1	1	1	Group 4 Message ID (0-2f)						7c0-7ef	Message Group 4
1	1	1	1	1	1	1	1	x	x	x	7f0-7ff	Invalid CAN Identifiers
10	9	8	7	6	5	4	3	2	1	0		

Highest Priority ↑

Figure 1. DeviceNet Message Groups

CAN IDENTIFIER											**DESCRIPTION**
10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
0	*Group 1 Message*				*Source MAC ID*						*Group 1 Messages*
0	1	1	0	0	Source MAC ID						Slave's Multicast I/O Message
0	1	1	0	1	Source MAC ID						Slave's COS/Cyclic I/O Message
0	1	1	1	0	Source MAC ID						Slave's Bit-Strobe I/O Message
0	1	1	1	1	Source MAC ID						Slave's Poll I/O Message
1	*0*	*MAC ID*						*Grp 2 Msg ID*			*Group 2 Messages*
1	0	Source MAC ID						0	0	0	Master's Bit-Strobe Command Message
1	0	Destination MAC ID						0	0	1	Master's Multicast Command Message
1	0	Destination MAC ID						0	1	0	Master's COS/Cyclic Acknowledgement Message
1	0	Source MAC ID						0	1	1	Slave's Explicit Response Message
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Message
1	0	Destination MAC ID						1	0	1	Master's Poll Command Message
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Message
1	0	Destination MAC ID						1	1	1	Duplicate MAC ID Check Message

Figure 2 CAN ID Distribution for Predefined Master/Slave Connection Set

Figure 2 illustrates some of the predefined master/slave connection set identifiers. The main areas of interest are the Group 1 and Group 2 I/O messages.

Before looking at the performance aspects of DeviceNet I/O messaging services, it is necessary to review what these messaging services are:

- Polling
- Strobing
- Change-of-State (COS)
- Cyclic
- Multicast

DeviceNet has the advantage over many other fieldbusses, in that it provides these versatile options. Any device can have the capability of these services, but it is not absolutely required by the DeviceNet Specification that a device supports them all.

Polling is a point-to-point communication between the master and a slave. Referring to Figure 2, the master will send a predefined Group 2 message to request I/O information from a particular device (node) using CAN ID 10xxxxxx101 (10 = Group 2 message, xxxxxx = Destination Node Address, 101 = Master I/O Poll Command). The node addressed will immediately respond with a Group 1 message supplying the I/O information using CAN ID 01111xxxxxx (0 = Group 1 message, 1111 = Slave Poll Response, xxxxxx = Source Node Address).

Figure 3 shows an example DeviceNet configuration with a PLC/Scanner controlling a photo sensor, three motor drives and an HMI for status and control. For example, referring to Figure 3, if all the slaves are set up for polling, the Master (PLC/Scanner) must send a poll request message to each node in order to receive a response from all of them. The master will send five Group 2 Poll request messages with the following CAN IDs: 10000001101 (MAC ID 1), 10000010101 (MAC ID 2), 10000011101 (MAC ID 3), 10000100101 (MAC ID 4), 10000101101 (MAC ID 5). Each node will respond with their respective Group 1 Poll response message: 01111000001 (MAC ID 1), 01111000010 (MAC ID 2), 01111000011 (MAC ID 3), 01111000100 (MAC ID 4), 01111000101 (MAC ID 5).

Strobing is a broadcast messaging scheme that allows a master to send one predefined Master/Slave message, and receive response from all of the nodes that are configured for strobing. Referring to Figure 2, the master will send a predefined Group 2 message to request I/O information from all strobe-configured devices using CAN ID 10xxxxxx000 (10 = Group 2 message, xxxxxx = Source/Master Node Address, 000 = Master I/O Strobe Command). All the nodes configured for strobing will immediately respond with a Group 1 message supplying the I/O information using CAN ID 01110xxxxxx (0 = Group 1 message, 1110 = Slave Strobe Response, xxxxxx = Source/Slave Node Address). Of course each of these messages will contain their relevant I/O data in the data field of the CAN frame.

Referring to Figure 3, if all the slaves are configured for strobing, the master will send one Goup2 Strobe request message using CAN ID 10000000000, and all of the slaves will respond with their respective Group 1 Strobe response message: 01110000001 (MAC ID 1), 01110000010 (MAC ID 2), 01110000011 (MAC ID 3), 01110000100 (MAC ID 4), 01110000101 (MAC ID 5).

Change-of-State (COS) allows a slave to transmit a Group 1 CAN data frame only when there is a change in its I/O status. Following a COS message, the master will immediately return an acknowledge message. This is particularly useful for simple a device such as a presence sensor that only has an on/off status for simply sensing the presence of an object, e.g. a box on a pallet. For example, the photo sensor (MAC ID 4) in Figure 3 can be configured so that it will send a Group 1 COS data message (CAN ID 01101000100) only when there is a change in its I/O status. Immediately following, the master (Node 0) will send an acknowledge message (CAN ID 10000000010) to indicate the receipt of the slave’s COS data message. A COS node also periodically sends a heartbeat message that can be set for up to 250 milliseconds.

Cyclic allows a slave to transmit a Group 1 CAN data frame periodically according to times specified by the end user/system integrator during configuration. This is useful for nodes that have pre-determined parameters that the required update time is known, e.g. motor drive frequency is required to be updated every 10 milliseconds, or a temperature sensor is required to update every 100 milliseconds. For example, referring to Figure 3, the drives (MAC IDs 1, 2 &3) are required to update their voltage, current and frequency parameters every 10 milliseconds, and the HMI (MAC ID 5) is required to update its parameters every 100 milliseconds. Every 10 milliseconds, the master will receive three back-to-back Group 1 cyclic data messages from the Drives (CAN IDs 01101000001, 01101000010 & 01101000011), after which the master will send an acknowledge to each drive. Every 100 milliseconds, the master will receive a Group 1 cyclic data message from the HMI (CAN ID 01101000101), after which the master will send an acknowledge to the HMI.

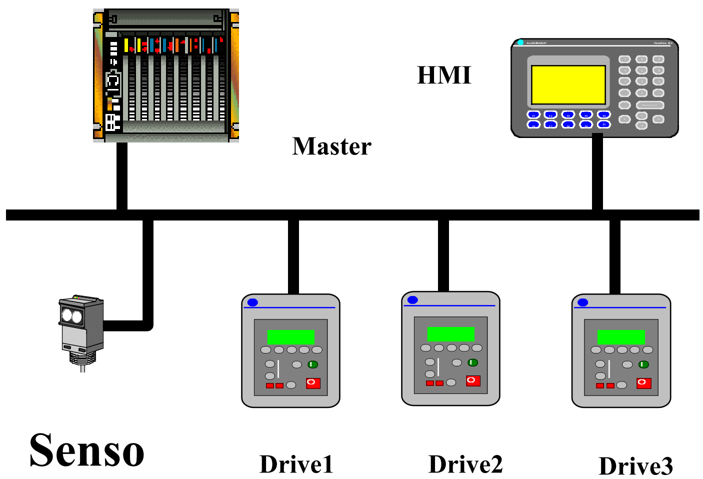

Figure 3. DeviceNet Example

Multicast is similar to Bit-Strobe, except that it allows the slave devices to be sub-divided into different groups, in which their I/O message production can be triggered differently. This prevents the transmission of unnecessary messages hence improving the bandwidth utilisation.

What also must be considered is the use of inter-scan delay within the Scanner to allow response to predefined I/O requests and COS/Cyclic messages. Inter-scan delay settings can help optimise bus loading through the control of space time between scans. During each scan, the scanner will send out its configured predefined Poll and Strobe messages. The user can set the "inter-scan delay" to facilitate a time delay between scans to allow responses, COS & Cyclic messages, and lower priority Group 3 explicit messages.

Here, some thought must go into the strategy needed for a particular automation configuration. If the system is a high-speed machine that has very fast time response requirements, and there are not many DeviceNet nodes, the inter-scan time delay can be set to quite a low figure (e.g. 2 milliseconds). This time is sufficient to allow response to predefined I/O messages, and capture any background explicit messages. If a system is very large (up to 64 DeviceNet nodes), and the I/O response is not as time critical (e.g. conveyer/assembly systems), longer inter-scan times are required (e.g.

10s of milliseconds) to allow sufficient time for Poll/Strobe response, COS/Cyclic inputs, and any explicit messages that may occur.

Communication Strategy

DeviceNet provides a selection of communication mechanisms that enable the development of an efficient and responsive I/O system:

- Strobe/Poll I/O
- Cyclic I/O
- Change-of-State (COS) I/O
- Multicast

In Strobe/Poll I/O, the inputs and outputs are updated at the same rate. Strobing allows a master to address many nodes, and all the nodes that are configured for strobing reply in unison. This is considered to be what is known as the Producer/Consumer model. In polling, the master addresses one node at a time, and each node will respond to its respective request message. This is quite similar to the Profibus strategy. This method is easy to configure, but it makes less efficient use of the data bus bandwidth than the other methods. Both Strobe and Poll request messages are sent during each scan, and therefore they can cause a high percentage of data busload.

In real time control networks (industrial control networks) such as DeviceNet, it is recommended that the busload be kept below 35%. Referring to Figure 4, many studies have shown that in contention-based computer networks, message latencies are quite low up to approximately 30%. At 33% busload the latency begins to rise suddenly at a non-linear rate. Information level data busses such as Ethernet can tolerate these busload levels, but real-time based industrial networks must be given closer design considerations.

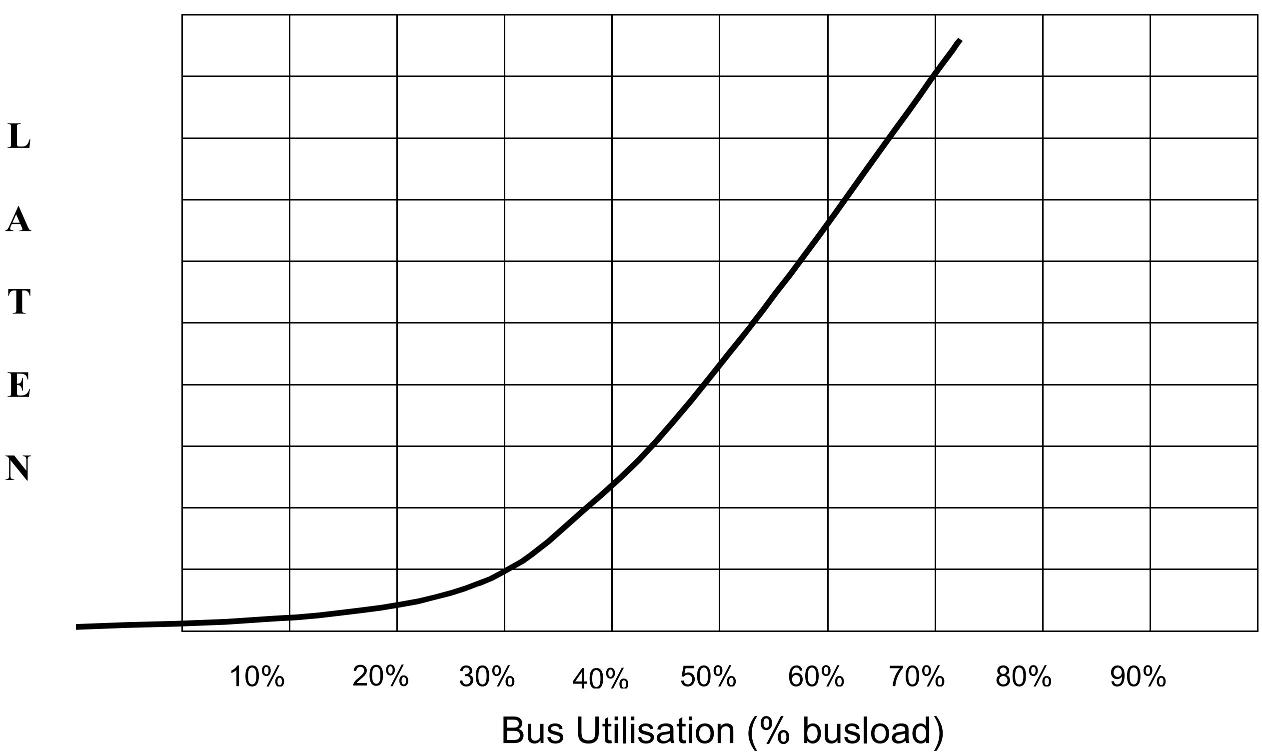

Figure 4. Latency caused by Bus Utilisation

Cyclic I/O messages are sent periodically as determined by the system designer. This makes more efficient use of the network bandwidth by reducing the input and output update rate of each station to its ideal minimum. This requires more effort to configure but can provide better system performance when the minimum required update rate for individual I/O devices can be identified. For example, temperature parameters need only to be updated at the 100s of milliseconds rate.

In COS, the I/O data is updated when the node input data changes. Inputs and outputs are periodically updated at a pre-determined rate to reassure the system that a node is still online if input data has not changed for a long period – much like a heart-beat message. This allows low data bus overhead, and it is very responsive to I/O changes.

In Multicast, the above-explained Strobing feature can be better utilised by strobing in groups. In most cases, at certain time the master requires only information from certain devices, hence with the use of multicast this can be achieved. It reduces the number of traffic generated because only intended devices are transmitting their I/O data.

However, to effectively make use of multicast, it requires the masters/scanners to intelligently acquire slaves' I/O information, i.e. update only when needed. With most of the existing masters/scanner's architectures the use of multicast does not deliver much interesting performance improvement, since most of these masters/scanners use time-trigger-transmission mechanism for their I/O commands. In other words, the same amount of I/O data from the slave devices are still sent on the network at each scan cycle, even though they may not be needed at that time. It merely splits a big chunk of back-to-back I/O data from the bit-strobe slaves into several smaller chunks of back-to-back I/O data.

It can be seen that the use of this information will allow a system designer to make use of all the above considerations to enable a time-optimised system. This will ensure efficient performance utilising minimum data bus bandwidth. The following is an analysis of various configurations used at the Warwick DeviceNet Test Labs to illustrate the effects of DeviceNet message service options, and the variation of the scanner inter-scan delay, on the bandwidth (or busload) of a DeviceNet system.

Analysis

Using a system with 40 Devices and a Baud Rate of 500 kbps, the following scenarios are considered at four inter-scan delay times (2 milliseconds, 5 milliseconds, 10 milliseconds, 20 milliseconds):

- All devices using the Polled I/O messaging service at the four inter-scan rates
- All devices using the Strobed I/O messaging service at the four inter-scan rates
- 20 devices using the Polled I/O messaging service at the four inter-scan rates, while the other 20 are set for COS at a heartbeat time of 250 milliseconds
- 20 devices using the Strobed I/O messaging service at the four inter-scan rates, while the other 20 are set for COS at a heartbeat time of 250 milliseconds

Consider the first scenario where all 40 Devices are polled with an inter-scan delay of 2 ms. Also consider for ease of understanding that the average CAN message is approximately 100 bits in length, i.e. each message will be approximately 200 microseconds. During the polling process, there will be 80 messages on the bus (40 poll requests, and 40 poll responses). These 80 messages it will take approximately 16 milliseconds (80 x 200 microseconds = 16 ms).

Now consider the second scenario where all 40 Devices are strobed with an inter-scan delay of 2 ms. During the strobing process there will be 41 messages (1 strobe request, and 40 strobe responses. These 41 messages it will take approximately 8.2 milliseconds (41 x 200 microseconds = 8.2 ms).

In the third scenario, where 20 devices are polled with an inter-scan delay of 2 ms, and the other 20 are set for COS with a heartbeat rate of 250 ms, the COS nodes are hardly to be considered due to the relatively long heartbeat rate. Here there will be 40 messages during the polling process, and the occasional COS message with the occasional heartbeat message from each of the other 20 nodes, which equates to 80+ messages per second. For the purposes of this analysis, due to the relative infrequency of the COS messages, only the 40 poll request/response messages are considered. Therefore the approximate time for messages during each scan is 8 milliseconds.

Utilising the same assumptions in the fourth scenario, where there will be 21 messages (1 strobe request, and 20 strobe responses), the approximate time for messages during each scan is 4.2 milliseconds.

The calculation of the bus loading is simplified using the following formula:

Busload (%) = (message time / message time + inter-scan delay) x 100%. This effectively the same as Busload (%) = (t_{active} / t_{active} + t_{idle}) x 100% [3]. It can be seen that the results using this formula with the 2 ms inter-scan delay are as follows:

- 88.8% (16/18) for the first scenario (all polled)
- 80.3% (8.2/10.2) for the second scenario (all strobed)
- 80% (8/10) for the third scenario (20 polled and 20 COS)
- 67.7% (4.2/6.2) for the fourth scenario (20 strobed and 20 COS)

From the above results, it can be seen that the busload is affected by the polling and strobing messaging services, and the combination of COS service. Applying the above calculations for different inter-scan delays, the table in Figure 5 illustrates the variety of bus loading with the many combinations, and how they affect the bus loading.

It is interesting to note that for the all-polled configuration, the lowest busload possible was 44%. It can be seen that wherever possible, it is advantageous to utilize Change-of-State or Cyclic messaging.

I/O Messaging	Inter-scan Delay	Busload
Polling	2 ms	89%
Strobing	2 ms	80%
Polling & COS	2 ms	80%
Strobing & COS	2 ms	68%
Polling	5 ms	76%
Strobing	5 ms	62%
Polling & COS	5 ms	62%
Strobing & COS	5 ms	46%
Polling	10 ms	62%
Strobing	10 ms	45%
Polling & COS	10 ms	44%
Strobing & COS	10 ms	30%
Polling	20 ms	44%
Strobing	20 ms	29%
Polling & COS	20 ms	29%
Strobing & COS	20 ms	17%

Figure 5 – Message Service & Inter-Scan Delay Effects on Busload

Effects of Inter-scan Delay

Referring to Figure 5, if the inter-scan delay is increased, the busload reduces. Inter-scan delay this is the time delay between consecutive I/O scans. The range of valid values is from 2 to 9,000 milliseconds.

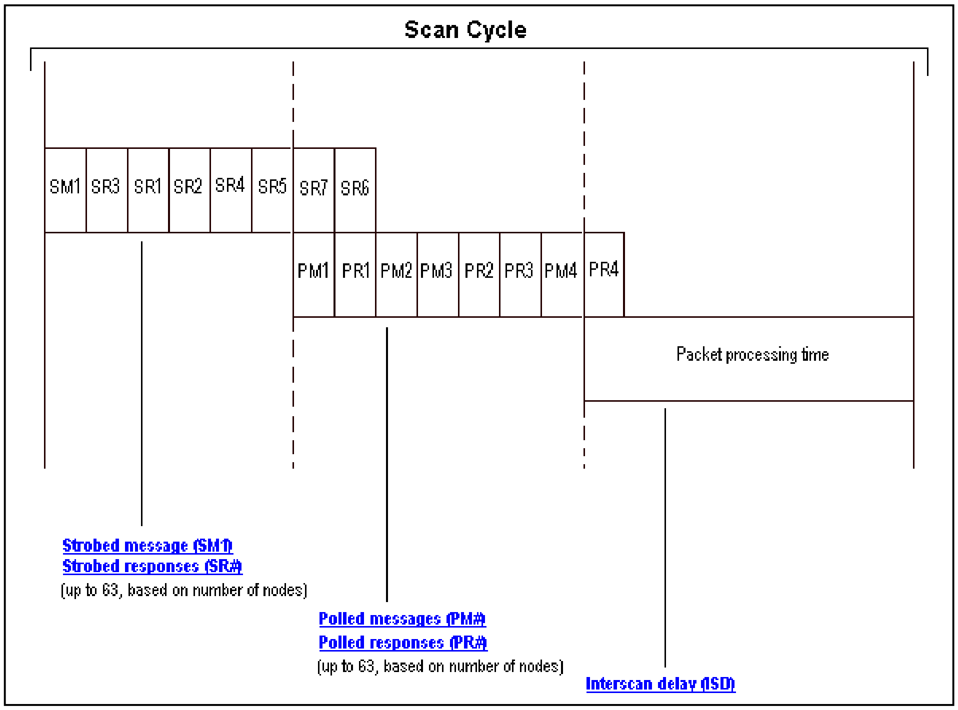

Figure 6 – Scan Cycle Timing

Figure 6 illustrates the time spacing effects of inter-scan delay [4]. This allow affective scheduling of messages, while leaving room for non-time critical messages [5] The scanner uses this period of time to perform non-time-critical communications on the DeviceNet network, such as communicating with RSNetWorx for DeviceNet software. Setting this parameter to a very low value increases the latency for non-time-critical scanner operations, including the time required to respond to RSLinx software and configuration functions. Setting this parameter to a very large value reduces the freshness of the I/O data being collected by the scanner and is not advisable. As there are limitations on the inter-scan delay, and there are choices of I/O messaging, it can be better utilised in reducing the busload and thus reducing the latency. Note the Strobed messages take priority over the Polled messages.

Besides analysing the busload, it is interesting to note the time from the beginning of a scan to the last message returned from a node. In the all-polled scenario, from the start of the scan to the last poll response message, the latency incurred is approximately16 milliseconds at 500 kbps. In many control systems, this is not a critical time for lower priority messages, but at 125 kbps, it may become a bit more noticeable where the latency incurred is approximately 64 milliseconds. In the all-strobed scenario, the approximate incurred latency of the last message on the scan is 8.2 milliseconds at 500 kbps, and 32.8 milliseconds at 125 kbps.

Prioritisation of Nodes through MAC ID Assignment

Of course the alternative in these extreme conditions would be to designate as many nodes as possible as Change-of-State or Cyclic. Another alternative is to set the higher priority nodes to the lower node numbers. If there are a combination of Strobed and Polled messages, the higher priority nodes should be set to Strobed. During each scan all the messages will be arbitrating according to the standard CAN bit-wise arbitration method.

Referring to Figure 2, it can be seen that the Group 1 I/O messages are prioritised by message ID in the following order:

- Change-of-State/Cyclic
- Strobe Response
- Poll Response

After which, each message group is prioritised by the MAC ID (node number). The lower the MAC ID, the higher the priority of the message in the above groups.

A system integrator has a variety of choices here when considering which messages are the most time critical, and therefore of the highest priority. Utilising these choices wisely will ensure rapid throughput of high priority data, and reduce busload, thus increasing the throughput of all data.

References:

[1] DeviceNet Specification – Open DeviceNet Vendors Association
[2] CAN Specification; Robert Bosch
[3] Quigley, Roxburgh, McLaughlin, Tang; A Low Cost CAN Bus Transducer and its Applications; Proceedings 7th International CAN Conference; 2000
[4] RSNetworx Help screen; Rockwell Automation
[5] Guaranteeing Message Latencies on CAN; K. Tindell, A. Burns: Proceedings 1st International CAN Conference;1994

Richard T. McLaughlin
Warwick Manufacturing Group
University of Warwick
Coventry, CV4 7AL
United Kingdom
44 2476 524711
44 2476 524307
r.mclaughlin@warwick.ac.uk
www.warwick.ac.uk/devicenet

Eva B. D’souza
Warwick Manufacturing Group
University of Warwick
Coventry, CV4 7AL
United Kingdom
44 2476 573002
44 2476 524307
e.dsouza@warwick.ac.uk
www.warwick.ac.uk/devicenet

Kiah Hion Tang
Warwick Control Technologies
University of Warwick Science Park
Coventry, CV4 7EZ
United Kingdome
44 2476 524711
44 2476 524307
xiong@tangskingdom.com
www.warwickcontrol.com