DeviceNet is a popular industrial device networking technology. The Linear Topology and speed/distance limitations proscribed within the DeviceNet specification published by Open DeviceNet Vendors Association (ODVA) fits the vast majority of applications. This section describes technologies available to address applications where these topology and speed/distance limitations become limiting.

Introduction

DeviceNet is based upon CAN technology developed by BOSCH originally for under-hood automotive and transportation applications in the 80’s. In the early 90’s, the technology began to be applied to a variety of process, industrial and commercial applications. In 1994, Allen-Bradley introduced DeviceNet as a standard for Industrial Automation Device Level Communications based on the CAN technology. Subsequently, the technology was turned over to an independent organization, Open DeviceNet Vendors Association, charged with developing and promoting the technology as an open standard. This paper will cover some of the critical aspects dictating the performance of CAN chips, the implementation within DeviceNet, options based upon “local CAN bridging” that can be used to extend the performance of a DeviceNet system with illustrations of how “local CAN bridge” technology has been applied.

DeviceNet is a trademark of the Open DeviceNet Vendors Association

CAN- Chip technology

DeviceNet is based on CAN-chip technology. CAN technology is a semiconductor implementation of a serial bus system developed by Bosch in the early 1980s as a means to reliably replace wiring harnesses used in automobiles and other vehicles. The technology had to be extremely robust and reliable since lives would rely on it. The technology must be able to work in the deserts as well as in Alaska. It has to work on deteriorating pot-holed roads in Chicago, the roads in Afghanistan, as well as the Autobahns of Germany.

The technology has been very successful and is used now in millions of transportation vehicles each year as well as hundreds of thousands automation applications.

From the idea to the first chip

> In the early 1980s, engineers at Bosch were evaluating existing serial bus systems regarding their possible use inpassenger cars. Because none of the available network protocols were able to fulfill the requirements of theautomotive engineers, Uwe Kiencke started the development of a new serial bus system in 1983. The new busprotocol was mainly supposed to add new functionality – the reduction of wiring harnesses was just a by-product, but not the driving force behind the development of CAN. Engineers from Mercedes-Benz got involved early on in the specification phase of the new serial bus system, and so did Intel as the potential main semiconductor vendor. Professor Dr. Wolfhard Lawrenz from the University of Applied Science in Braunschweig-Wolfenbüttel, Germany, who had been hired as a consultant, gave the new network protocol the name ‘Controller Area Network’. ProfessorDr. Horst Wettstein from the University of Karlsruhe also provided academic assistance.
>
> In February of 1986, CAN was born: at the SAE congress in Detroit, the new bus system developed by Bosch wasintroduced as ‘Automotive Serial Controller Area Network’. Uwe Kiencke, Siegfried Dais and Martin Litschelintroduced the multi-master network protocol. It was based on a non-destructive arbitration mechanism, which would grant bus access to the message with the highest priority without any delays. There was no central busmaster. Furthermore, the fathers of CAN –the individuals mentioned above plus Bosch employees Wolfgang Borst,Wolfgang Botzenhard, Otto Karl, Helmut Schelling, and Jan Unruh –had implemented several error detection mechanisms. The error handling also included the automatic disconnection of faulty bus nodes in order to keep upthe communication between the remaining nodes.The transmitted messages were not identified by the node addressof the transmitter or the receiver of the message (as in almost all other bus systems), but rather by their content. Theidentifier representing the content of the message also had the function of specifying the priority of the message within the system. (1)

DeviceNet Specifications

The DeviceNet Specifications were originally developed by Allen-Bradley (now Rockwell Automation) but was transferred to an independent organization, The Open DeviceNet Vendors Association (ODVA) – http://www.odva.org/ . The specification is maintained by the ODVA with any changes going through thorough review, vetting, and voting by Technical Review Committees and Working Groups with identified specific interests and expertise.

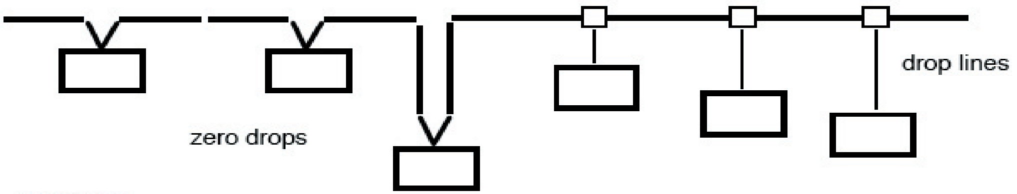


FIGURE 1

DeviceNet is described in the Specification as a Linear Bus Technology, illustrated in Figure 1, with the identified speed/distance limitations.

General Features of DeviceNet include:

- Trunk line, drop line configuration
- Node removal without breaking trunk line
- Up to 64 addressable nodes
- Signal and 24Vdc Power in same cable
- Selectable Data Rates (125k, 250k, 500k) as shown in Table 1
- Both Sealed and Open-Style connections
- zero node separation
- 121 ohm terminator at each trunk line end

TABLE 1

Data Rates	125 Kbps	250 Kbps	500 Kbps
Thick Trunk Length	500 m	250 m	100 m
Thin Trunk Length	100 m	100 m	100 m
Max Drop Length	6 m	6 m	6 m
Cum Drop Length	156 m	78 m	39 m

(2)

These characteristics work well with most applications. However, many applications in excess of these distances and or cumulative drop lengths would like to take advantage of the features and power provided within DeviceNet.

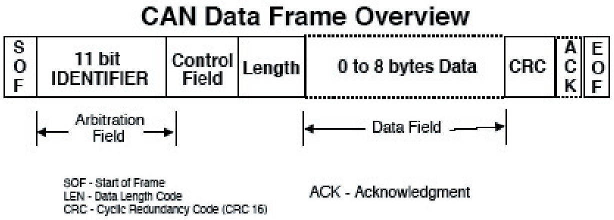
Distance and Topology Options

Traditional repeaters do not solve the problem since they generally amplify the signal. In the case of CAN chips, there is an underlying Bit Arbitration that mandates certain response requirements that effect the maximum length/speed specifications.

Bitwise Arbitration

The main characteristics of Bitwise Arbitration are:

- CSMA/NBA - Carrier Sense Multiple Access with Non-destructive Bitwise Arbitration
- Any node can access bus when quiet
- Non-destructive bit-wise arbitration allows 100% utilization and message priority based on 11-bit packet identifier
- CAN provides automatic error detection, signaling, and retries
- Data portion of packet can be 0 to 8 bytes long – see Figure 2



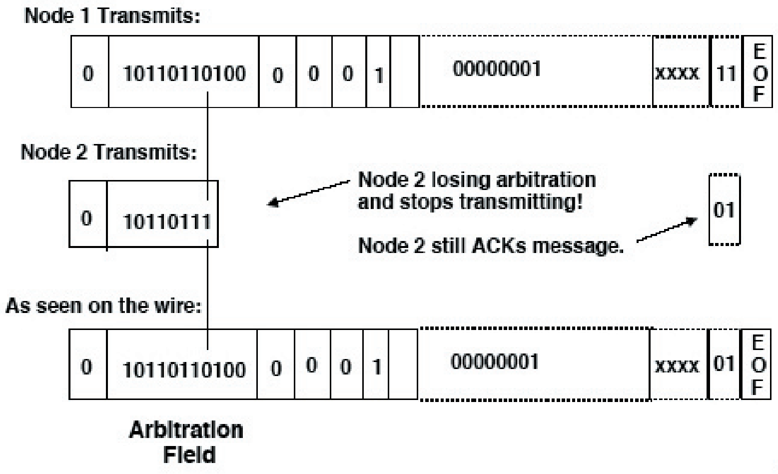
(3)

Figure 2

Figure 2 shows the CAN Data Frame as used in DeviceNet.

Figure 3 provides an illustration of the arbitration mechanism used in CAN chips to establish priority, to resolve access conflicts, and to provide higher levels of data integrity.

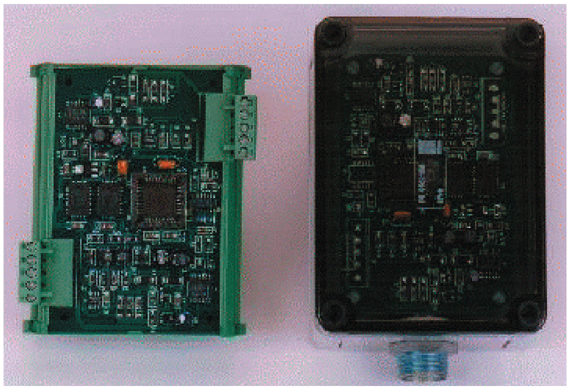
- Similar to Ethernet, each node attempts to transmit when the bus is free
  - Unlike Ethernet, there is no collision.
- If two or more nodes start transmitting simultaneously, bus conflict is resolved by Bitwise arbitration using the IDENTIFIER.
  - A “0” is dominant on wire and overrides a “1”
  - When a node transmits a “1”, but hears a “0”, it immediately stops transmitting
  - The “winning” node continues to transmit its message to completion
  - THIS MECHANISM GUARANTEES THAT NEITHER INFORMATION NOR TIME IS LOST.
- The value of the IDENTIFIER defines priority during arbitration (lowest IDENTIFIER “wins” arbitration). This means two nodes CAN NOT share the same IDENTIFIER.



(3)

Figure 3

- ALL nodes check the consistency of the message being received and will acknowledge a consistent message and flag an inconsistent message in the ACK SLOT.



The bottom line is that because of the Bitwise arbitration built into the CAN chip, certain timing and performance characteristics are enforced. The distance/speed limitations are chosen to ensure that installations will meet the CAN Bitwise Arbitration timing specifications. One cannot simply amplify the signal to increase the distance. Amplifying or even simply repeating the signal will not address the Bitwise arbitration requirements.

Options considered

Several techniques exist to address these restrictions. One technique, sometimes called "remote bridging", provides a gateway to a totally different network and then converts it back to DeviceNet at the other end. Remote bridging interconnects two physically separated but similar networks together using a different interconnecting medium.

New network segment

An alternative, lower cost technique is to complete the logical bus and to then generate a new network segment. A "local CAN bridge" would have two CAN chips, one for one segment and one for the other. A microprocessor would pass messages between the two CAN chips. Using this approach, the effective length of the complete network is doubled while requiring only one bridge. This technique is sometimes called "local CAN bridging".

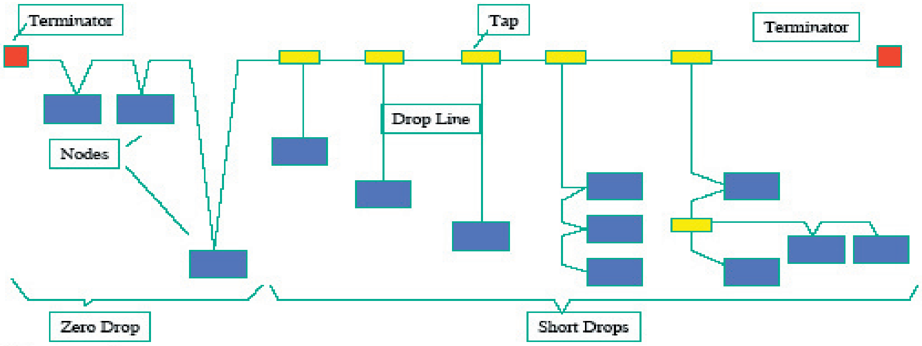


Figure 4

A traditional linear topology network has exactly two terminating resistors located at either end, by system definition as shown in Figure 4.a

Devices such as these providing "local CAN bridges" can be applied in a variety of ways to provide practical alternatives to overcome the inherent distance/speed limitations.

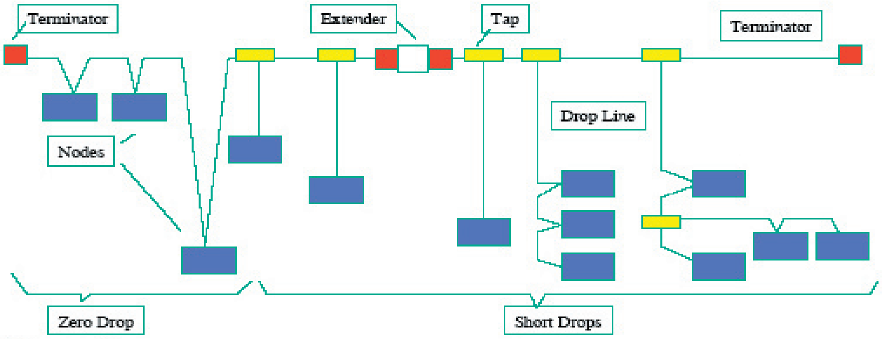


Figure 5

Using a single bridge creates two separate physical networks as highlighted by the two-pairs of termination resistors shown in Figure 5. Each segment complies with the requirements and is tied together with a communications microprocessor, which provides a store-and-forward function as well as electrically isolating each network.

This has NOT resulted in two DeviceNet "logical" networks. There are still only 63 addresses. All of the nodes are still working in the master-slave configuration with the same master and the same slaves.

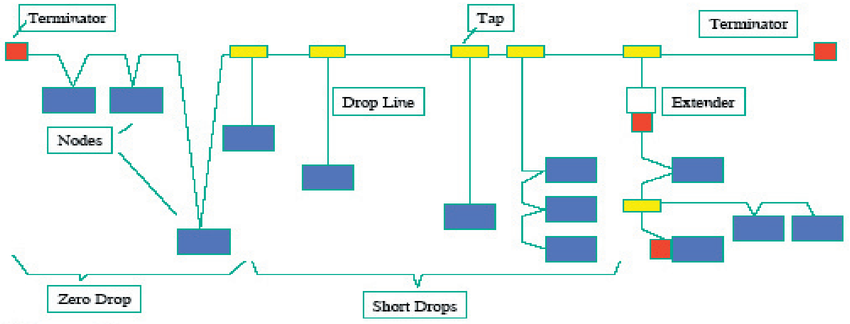


Figure 6

Bridges can also be used in the drop-line where the overall drop-line limits would have been exceeded as shown in Figure 6.

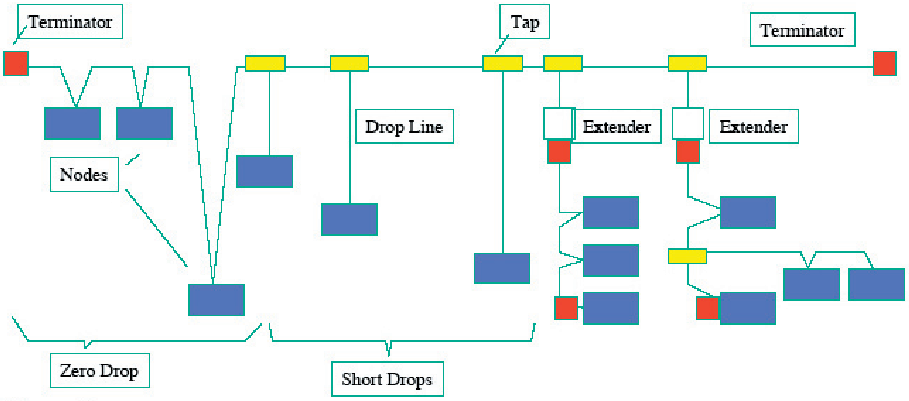


Figure 7

Since each network segment created using a "local CAN bridge" is electrically independent and self-compliant with the timing requirements of the CAN-chip technology, one can create multiple long-distance drop-lines, each as long as the trunk line. Figure 7 illustrates multiple long drop-lines.

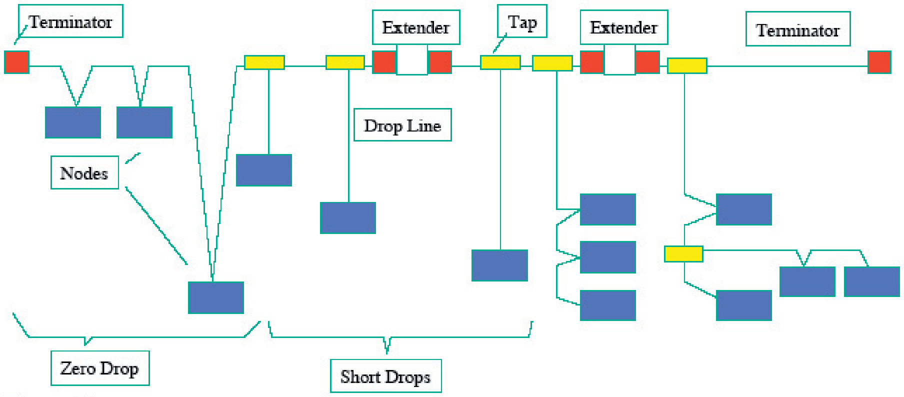


Figure 8

Figure 8 illustrates the trunk line can be extended using this technology. From a theoretical perspective, there is no limit to the number of extenders and network segments that can be implemented in this manner. Practically, there are trade-offs and performance issues which must be evaluated in all system configurations. These trade-offs will be covered later.

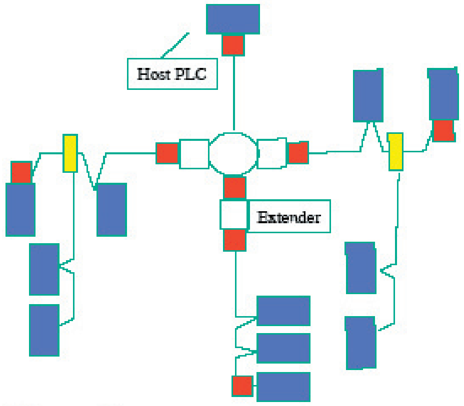


Figure 9

Other topologies, such as this modified Star, Figure 9, can be implemented by taking advantage of this technology.

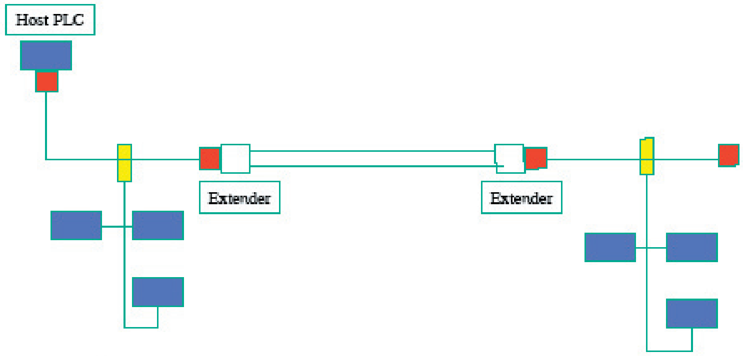


Figure 10

Figure 10 illustrates fiber optic versions are available for applications that are subject to lightning strikes, damaging chemical environments, or severe electrical disturbances.

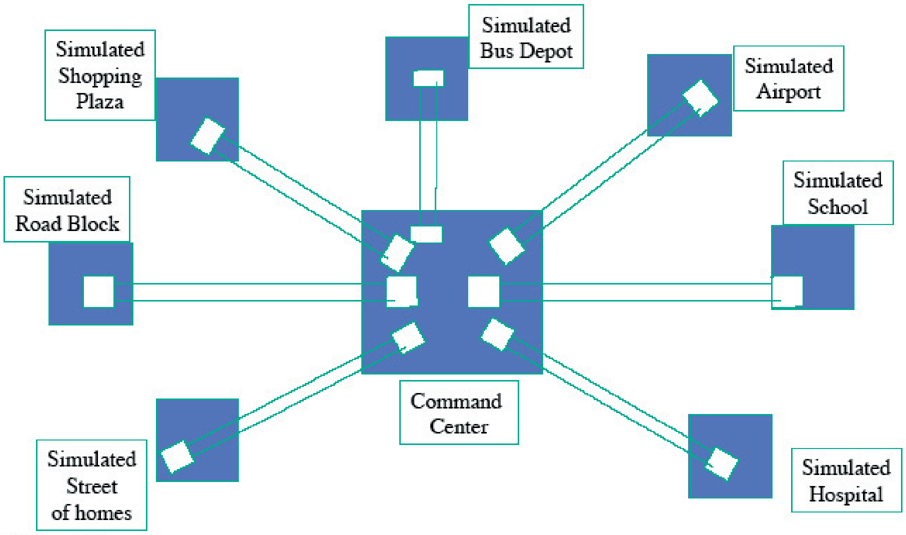


Figure 11

The technology has been implemented in Anti-Terror Training Applications at Fort Knox as illustrated in Figure 11. In this case, the fiber-optic version was used since it was an outdoors application.

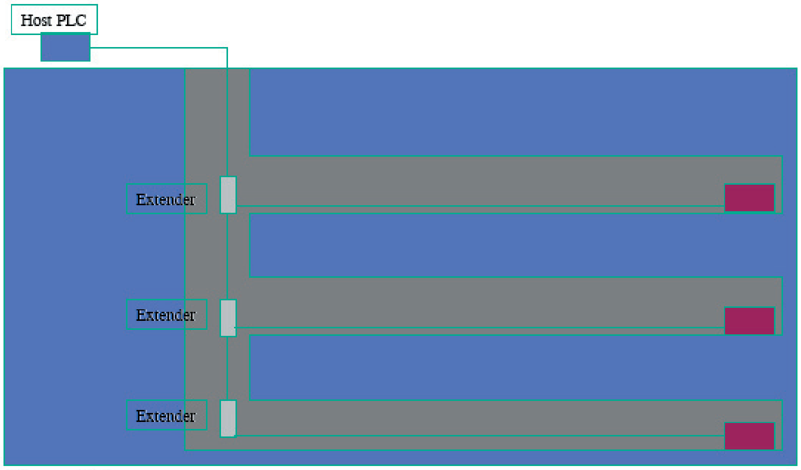


Figure 12

The technology has been applied in deep-mining applications. Figure 12 is from a South African diamond mine where the requirements were to not only meet distance requirements, but to isolate each level so that a short on one level would not effect operation on other levels.

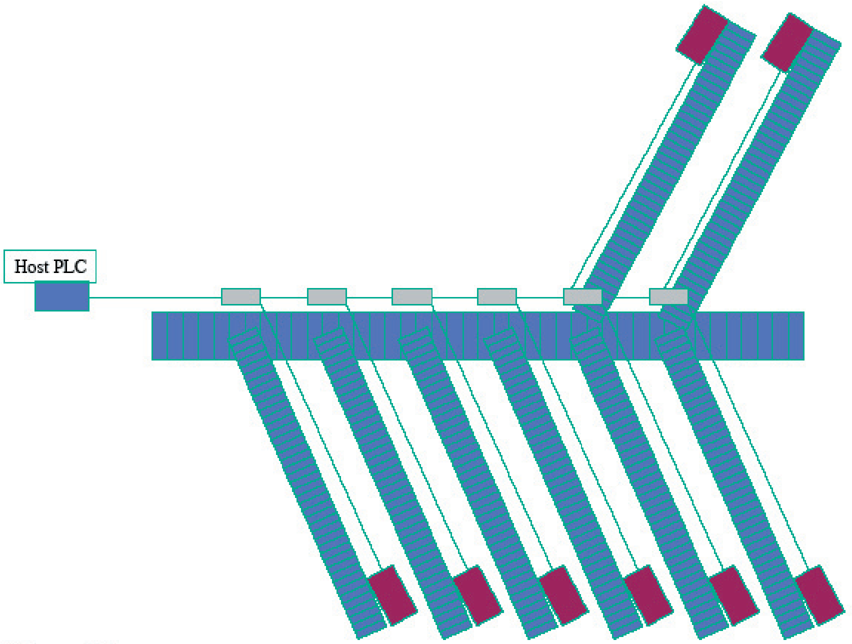


Figure 13

Other typical applications included material handling. Figure 13 is from a clothing distribution center with a central conveyor fed from a number of storage aisles. The DeviceNet Trunk-line ran down the length of the central conveyor. A "local CAN bridge" was used at the intersection of each aisle to create its own network segment as drop-lines.

The network isolation features are illustrated in Figure 15, an OEM application of a large drilling machine controlled by a PLC. The PLC controlled not only the electric motors driving the drill, but auxiliary services such as water and lubrication pumps as well as an instrumentation pack which was installed near the drill-head at the end of the shaft. The problem that they had was when the network connection to the drill-head sensors was damaged, it brought down the complete network, which slowed down the process and made repairs more difficult. By isolating the sensor package, they were able to keep the rest of the equipment running and thus better able to facilitate repairs.

This application was not on DeviceNet, but on CANopen, an international standard Superset including DeviceNet.

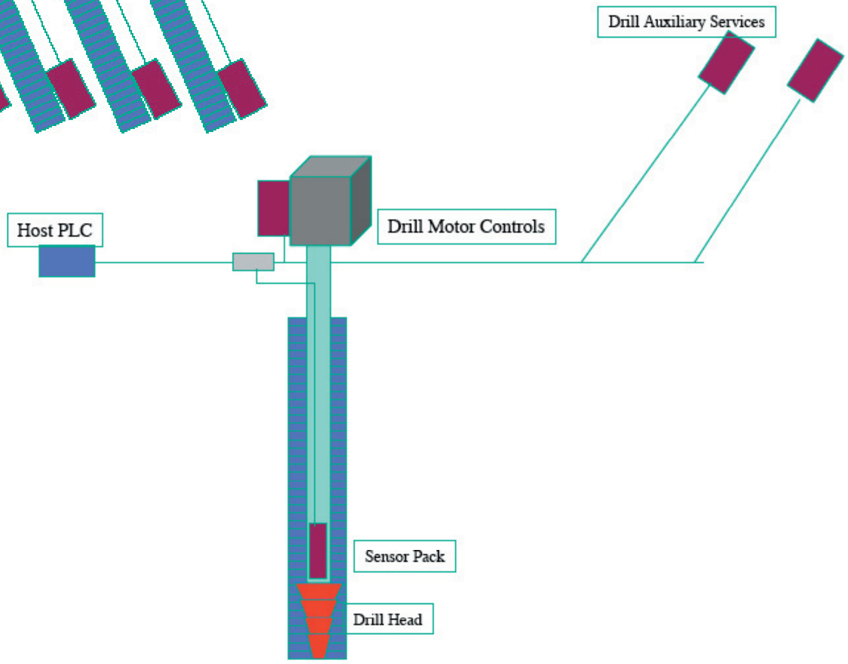


Figure 15

There is no Free Lunch

"Local CAN Bridges" incorporate a technique called "Store and Forward". This entails receiving the message in its entirety, storing it momentarily, and then sending it. The implication is that this introduces latency into the message – typically in the 0.5 millisecond order of magnitude. The latency will be introduced at each of the "Local CAN Bridges". The Applications Engineer needs to include this as part of their consideration for the overall system design. If the application is very time critical or if there is tight process synchronization, the introduction of any latency may dictate that this technology should not be used.

CONCLUSIONS

"Local CAN Bridges" can be used to extend the length of DeviceNet as well as expand the manner in which network topologies may be implemented, thus allowing DeviceNet to be used in a broader array of applications. It is up to the Applications Engineer to assess the features and trade-offs involved to determine if this technology helps to meet the project requirements or not.

REFERENCES

1) http://www.can-cia.org/can/protocol/history/history.html, CAN in Automation international users' and manufacturers' group

2) DeviceNet Technical Overview http://www..odva.org

3) DeviceNet Technical Overview Ray Romito, DeviceNet trainer for Rockwell Automation.Allen-Bradley First presented to SI/OEM User Group April 30, 1996 http://www.odva/org