

A LOW COST MOTION CONTROL PRODUCT ARCHITECTURE

Peter C. Di Giulio

PITNEY BOWES, INC.

35 Waterview Drive, Shelton, Ct. 06484

(203) 924-3109 FAX (203) 924-3385

April 15, 1992

ABSTRACT

This paper describes an innovative communications based architecture for motion control products. High performance, multi-axis motion control capabilities are provided to support a broad range of applications such as article processing, automotive systems, factory automation, and robotic systems. The architecture is independent of the communication media and features low overall life cycle costs. The benefits begin with an evolving set of re-usable "off-the-shelf" modules to accelerate product development. Integrated diagnostic capabilities are provided to facilitate product level testing. To achieve low module costs, a strong emphasis has been placed on commonality of parts and circuitry requirements have been minimized by developing a communication protocol optimized for motion control applications. In addition, the architecture offers a natural migration path to further reduce costs by employing PLD & ASIC electronics integration technologies.

BACKGROUND

In recent years, a number of communications based motion control architecture schemes have emerged in an attempt to overcome harnessing challenges and to provide a more flexible and modular development environment. On one extreme, these offerings emphasize a highly distributed processing environment based completely on peer-to-peer level communications. On the other extreme, they emphasize a highly centralized master/slave approach.

The SSD LINK Protocol¹ is an example of a distributed processing communications architecture that passes information over the network on an event driven basis. It is effective for applications (1) where there is a strong separation of functionality from one sub-system to another, (2) which do not have rigid deterministic inter-node timing requirements, and (3) which carry unaggressive system cost targets. During development, the strong partitioning of functions will ease the development of the individual functional modules. However, where there are tight timing relationships between these modules, final integration of many independent functional modules will likely be challenging.

SERCOS² is a highly centralized architecture which has been developed for numerical control applications by a consortium of German companies. It communicates low-level motion control information on a master/slave basis between a central motion control node and nodes which interface to motors, solenoids, sensors, etc. This architecture provides a natural support for tightly coupled, high performance motion control requirements where the reliability of the machine strongly depends on the smooth hand-off of articles from one motor to another during steep accelerations and high velocities. This is particularly true for variable load applications. With highly centralized architectures, the intelligence requirements for slave nodes which interface to motion control peripherals is very low. This provides the opportunity to reduce the cost of many slave nodes at an added cost to one centralized node.

The SERCOS protocol passes low-level data over the network to close servo loops in the central controller. This requires the communication of this information to be very regular. The high refresh rate required consumes significant channel bandwidth. This is exacerbated in SERCOS since all of the networked motors are communicated with at the same sample rate. Considerable channel bandwidth is wasted when the sample rate is very high to accommodate a single motor requiring a much smaller sample period than the rest.

The TRAJECTORY REGULATOR PROTOCOL³ offers a less highly centralized communications architecture scheme. It employs a master/slave approach to communicate higher level profile control information between a host and motor/sensor interface nodes which are based on Digital Signal Processing (DSP) technology. The capability provided at each motor is very robust and able to accommodate high performance motion control requirements for that motor. However, the total system cost resulting from placing a DSP on every motor node will be significant and, as with distributed systems, coordinating tightly coupled motion control requirements between motor nodes will be challenging for variable load applications.

ARCHITECTURE GOALS

The objective is to provide an effective communications based architecture for integrating system control elements within a product. This Intra-Product Communications Architecture (IPCA) should afford a low life cycle cost over the spectrum of communication requirements manifested in both the motion control industry and high volume products (see Figure 1). As lower cost motion control system solutions emerge, it is likely that a broader base of high volume motion control applications will materialize. To date, no known solution effectively leverages the trends in electronics integration to the motion control problem with the goal of minimizing the cost of (1) interfacing to and controlling motion control peripherals, and (2) supporting typical peripherals which are intrinsic to high volume products.

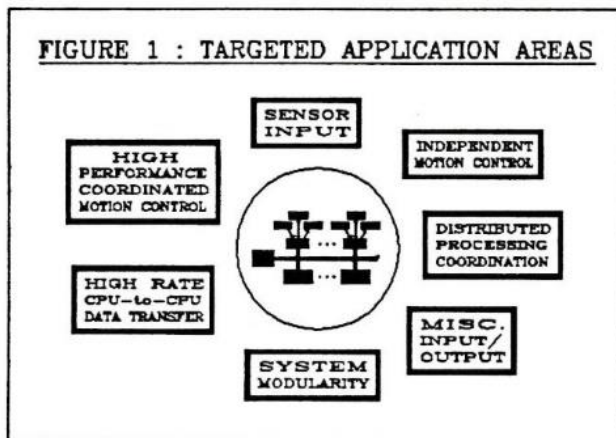


FIGURE 1 : TARGETED APPLICATION AREAS

The key life cycle cost reduction goals are to :

1. Accelerate product time to market by developing a motion control architecture which will have broad applicability and facilitate the re-use of modules, designs, and development & manufacturing tools. Reducing development requirements will allow a greater focus on achieving product functionality. An evolving corporate engineering knowledge base will enable a rapid redeployment of internal resources. Note that broad applicability implies :
 - a. A low cost/performance ratio across a wide range of product requirements,
 - b. Support for both Centralized and Distributed Requirements as illustrated in Figure 2,
 - c. Tools to facilitate tailoring the architecture to varying design philosophies,

- d. Support for coordinated high performance control requirements, and
- e. CPU and communication media independence to accommodate varying application requirements and emerging technologies.

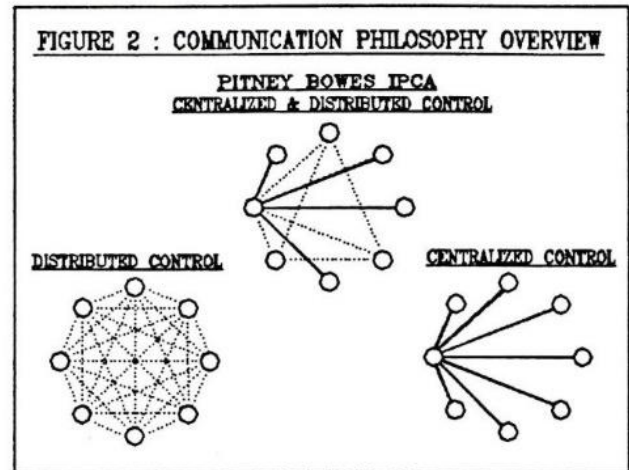


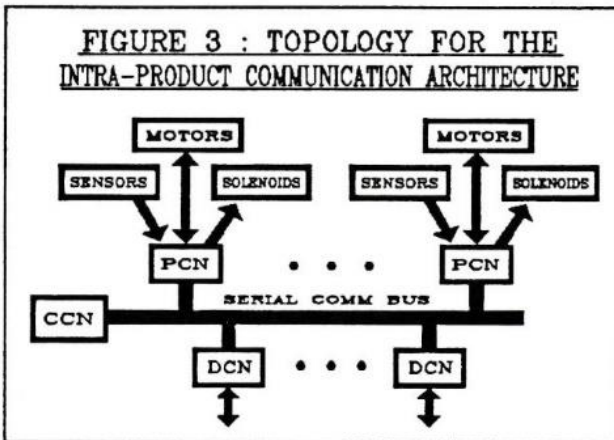
FIGURE 2 : COMMUNICATION PHILOSOPHY OVERVIEW

2. Reduce product costs by employing serial communications to promote standard electronic interfaces and packaging. A strong focus is placed on minimizing incremental communication node connect costs. Volumes resulting from part commonality and the use of the product architecture in multiple applications will enhance purchasing leverage and allow higher tooling levels to be employed to reduce recurring costs. Additionally, serial communications eliminates dedicated interface costs for option modules. The cost to allow an option module to be connected to a product is pushed into that module's ability to connect to the communications network.
3. Reduce manufacturing and after market product support costs. The reuse of proven modules and designs will foster a maturing reliability in future products. Built-In test features are needed to accelerate resolution of product problems. Part commonality is required to reduce overall manufacturing and spares stocking requirements.

In order to minimize the development risks, emphasis has been placed on building the proposed motion control architecture using proven communication and motion control structures and standards as much as possible.

ARCHITECTURE OVERVIEW

The IPCA Topology is provided in Figure 3. Three types of communication nodes are supported. They are a Central Control Node (CCN), a Peripheral Control Node (PCN), and a Distributed Control Node (DCN). The protocol definition allows most protocol functions to be implemented cost effectively in hardware.



The CCN provides the network controller function and, if PCNs are present, performs motion control functions as well. The key network control functions are (1) initializing the network and (2) controlling DCN and PCN access to the network in a manner which avoids contention. The motion control functions include multi-axis coordination, profile generation, and servo filtering.

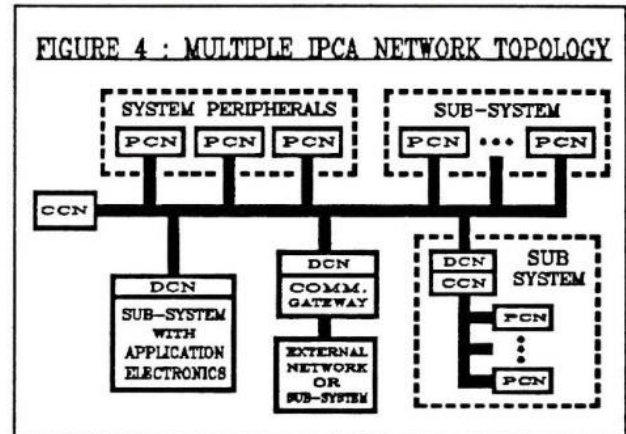
PCNs are smart I/O nodes which provide an interface to motors, solenoids, sensors, status indicators, etc. PCNs do not include CPUs but act under the control of the CCN. PCNs send sensor status messages to the CCN in response to CCN control messages directed at them. They can not send unsolicited messages on an event driven basis. As mentioned, the CCN provides a centralized motion control function for PCN smart I/O nodes. In a similarly manner to SERCOS, the CCN communicates low-level data to PCN motor/sensor controllers with a high refresh rate. As discussed later, a major advantage of the IPCA protocol is that some PCNs can be communicated to at a higher rate than others with proper scheduling by the CCN.

DCNs are CPU based nodes which can share the system control function in a distributed fashion. The communication architecture supports peer-to-peer communication amongst these nodes on an event driven basis. The CCN arranges DCNs' access to the network and can communicate with them. Note that a DCN's application is unrestricted and can take on a broad range

of functions such as a user interface, intelligent peripheral controller, system controller, communications interface, profile generation, servo filtering, and independent control of Motors, Sensors, and Solenoids.

Each physical network will support 1 CCN and a combination of up to 31 DCNs and PCNs. This is more than sufficient to support a wide range of motion control applications. For large systems, multiple physical networks can be employed to achieve the overall system requirement as discussed below.

The IPCA offers significant flexibility for partitioning control system requirements into multiple networks. Figure 4 shows a few ways for expanding the IPCA to accommodate large or expanding product requirements. Consider the control system peripherals shown in the upper left hand corner of Figure 4.



As other sub-systems are added to a product, the first option is to examine if there is sufficient capacity on the existing IPCA communications network to absorb the additional requirements so as to avoid the cost overhead in adding independent network. Figure 4 shows the connection of an optional control sub-system module to the existing network in the upper right hand corner.

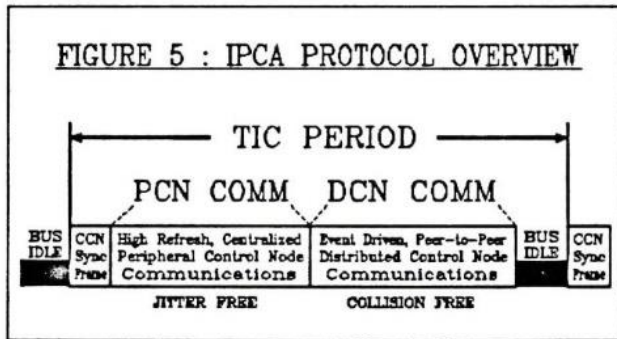
If there is not sufficient capacity to absorb the additional requirements, or other system level considerations make it undesirable to do so, a DCN can be used as a means to expand the capability of the architecture. As shown along the bottom of Figure 4, a DCN could be an independent sub-system with non-IPCA application electronics, a communication interface to an external network, or another IPCA network. In this last case, a DCN could be the CCN for an IPCA sub-network as shown in the figure. Alternatively, a single node can be a DCN on two independent IPCA networks to provide communications between those networks (not shown).

DETERMINING NODE PHYSICAL ORDER

The IPCA incorporates the means to determine the physical order of motion control nodes. This is required to allow product modules to be assembled in any order without requiring a service adjustment during product configuration set-up. As part of network initialization, the CCN determines the product configuration based on the presence and physical order of the nodes on the bus. The CCN will selectively download an alternate node address to provide unique node addresses for all the nodes. This will always need to occur when two or more identical modules are used in the product.

IPCA PROTOCOL OVERVIEW

An overview of the IPCA protocol is provided in Figure 5. A basic communication "TIC PERIOD" is defined to support the IPCA's communication requirements. Each Tic Period is broken down into three phases; these are (1) the CCN Sync Message, (2) PCN Communications, and (3) DCN Communications. Each of these three phases are tightly coordinated by the CCN (which contains the Tic Period timing function). The Tic Period time is user configurable and will typically be set between 0.5 and 2 msec.



The first phase consists of the CCN broadcasting a "Synchronization Message" to all of the DCNs and PCNs at the start of the Tic Period. This message, which can be thought of as the network "heart beat", is used to (1) synchronize PCN control/sensor activities, (2) provide a slow clock to be used for various node timing functions, and (3) as a system watchdog function. In this last case, DCN and PCN application functions can switch to safe state if a network failure prevents CCN Sync Messages from being communicated.

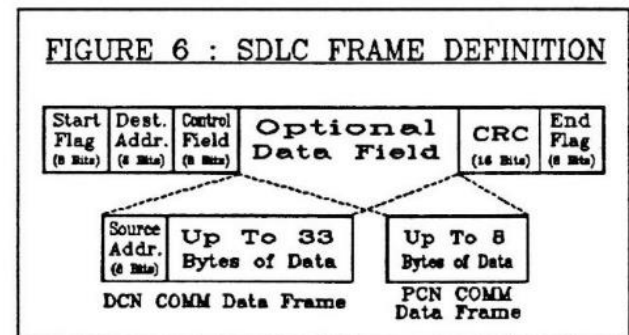
The second phase is reserved for CCN-to-PCN communications. This immediately follows the CCN Sync Message to facilitate the strict timing requirements for

PCN communications. Here, the CCN communicates with selected PCNs at a high refresh rate and in a manner which provides very predictable and jitter free timing. A Cyclic Redundancy Check (CRC) is used to detect communication errors. If errors are detected, the message is ignored. Since this data is updated frequently, no retry is performed.

The time following PCN communications and until the start of the next Tic Period is reserved for DCN communications. Predicated on an enable messages from the CCN, unsolicited DCN messages are sent directly between DCNs. The DCN access scheme eliminates the possibility of contention. A CRC and immediate acknowledge are used to detect communication errors. If a communication error occurs, a retry will be performed. A sequence bit is employed to ignore duplicate messages. More than one DCN message can be sent within a Tic Period. When the CCN determines that there is not enough time remaining in the Tic Period for another message to be sent, it will stop issuing the enable message. As shown in Figure 5, the bus will be idle until the start of the next Tic Period.

SDLC FRAME DEFINITION

The SDLC⁴ definition for a communications frame shown in Figure 6 has been selected as the foundation for the IPCA protocol due to its popularity and proven reliability. As shown in the figure, the frame consists of a start flag, Destination Address, Control Field, Optional Data Field, Cyclical Redundancy Check (CRC), and ending flag.



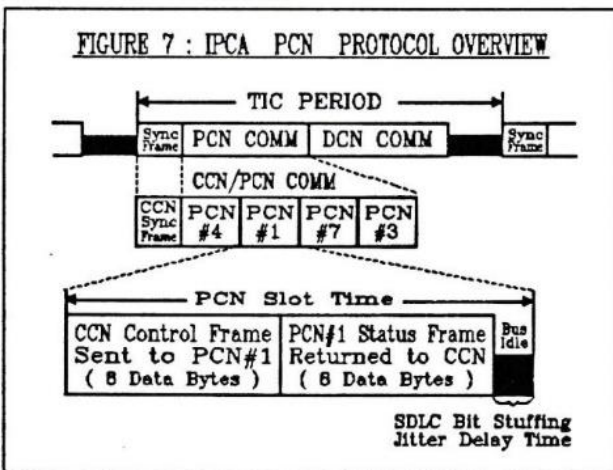
An NRZI scheme is used for clock encoding. "Bit stuffing" is used to guarantee that the 8-bit flag ('01111110') used to start and end the frame will be unique. Except when sending these flags, the transmitter sends an extra '0' bit after sending five consecutive '1' bits. The receiver always removes a '0' bit found in the data stream following five consecutive '1' bits.

The destination node address identifies the node to receive the frame. The control frame identifies the nature of the message frame (ie. a Network Control Function, PCN message, DCN message, or DCN message acknowledge). The CRC is a 16-bit error check on the frame content.

The data frame sent between the CCN and PCN contains up to 8 bytes in the data field. The data frame sent from one DCN (or CCN) to another contains up to 34 bytes in the data field. Here, the first byte will always be the source node address (useful for debugging) and the rest of the field will contain up to 33 bytes of data. Note that the DCN communications software driver provides facilities for sending long messages (such as a software download) as a series of these 33 data byte packets.

IPCA PCN PROTOCOL OVERVIEW

The IPCA PCN Protocol provides a high performance, well focused means of communicating critical control and sensor information between the CCN and smart I/O functions. Figure 7 provides an overview of PCN communications. The CCN is the only node that can send control messages to and receive sensor/status messages from PCNs. PCN communication commences following the CCN Sync Frame. As shown in the figure, the CCN sends up to 8 bytes of control information to a PCN and the PCN immediately acknowledges by returning up to 8 bytes of sensor information back to the CCN.

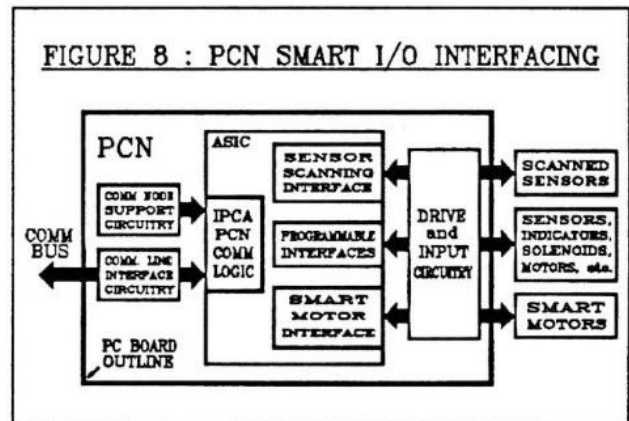


Jitter free timing is achieved by having the CCN-to-PCN communications commence at an integer number of "PCN Slot Times" following the CCN Sync Frame. The PCN Slot Time is based on the time to send the two 8 data byte frames under a worst case "bit stuffing" delay condition (many data bytes with all '1' bits).

The CCN incorporates programmable scheduling hardware to (1) allow communication to PCNs in any order, (2) reduce CCN CPU loading requirements, and (3) meet the network's very precise timing requirements.

PCN SMART I/O INTERFACING

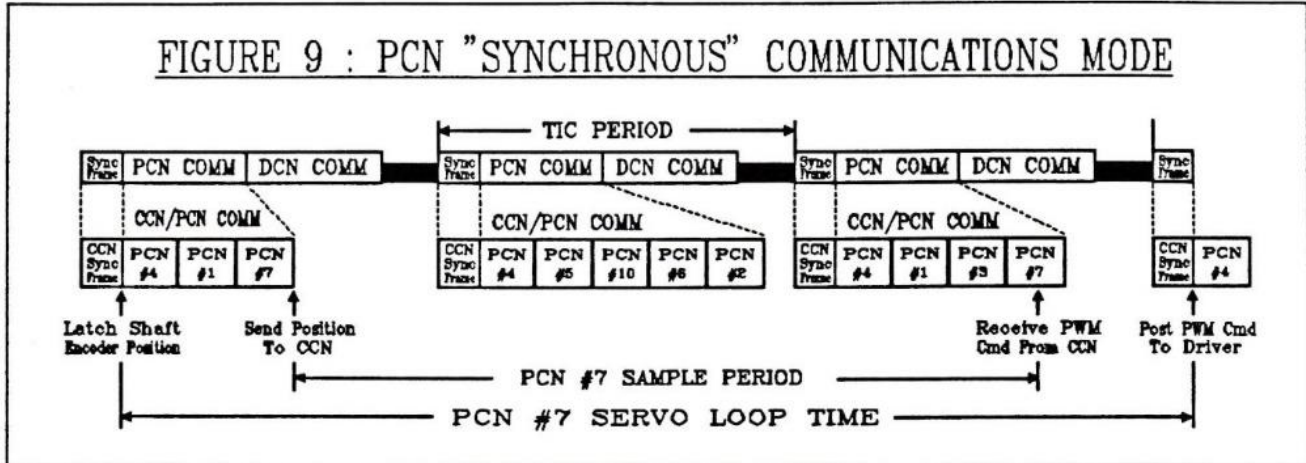
The IPCA provides a very efficient means of interfacing to motors, solenoids, sensors, indicators, etc. The IPCA's partitioning of the CCN-to-PCN motion control and communication requirements greatly simplifies PCN circuitry requirements by (1) locating complex motion control decision making in the CCN, (2) eliminating the need for a CPU environment with software download facilities in PCNs, and (3) leveraging the high PCN communication refresh rates to eliminate PCN communication error recovery requirements (PCNs have no retry capabilities). Figure 8 provides a block diagram for a typical low cost PCN resulting from this reduction in circuitry requirements.



As shown in the figure, the PCN minimally requires communication interface and support circuitry, smart I/O functions, and analog driver & interface circuitry. A small ASIC can be employed to highly integrate the PCN's streamlined communication and smart I/O logic requirements. Since the smart I/O requirements are optimized to the application's requirements, all of the defined logic device functions are used to capacity.

SYNCHRONOUS vs IMMEDIATE MODES

The IPCA offers two modes for coordinating the usage of information that is exchanged between CCN and PCNs. They are the Synchronous PCN Communication Mode and the Immediate PCN Communications Mode.

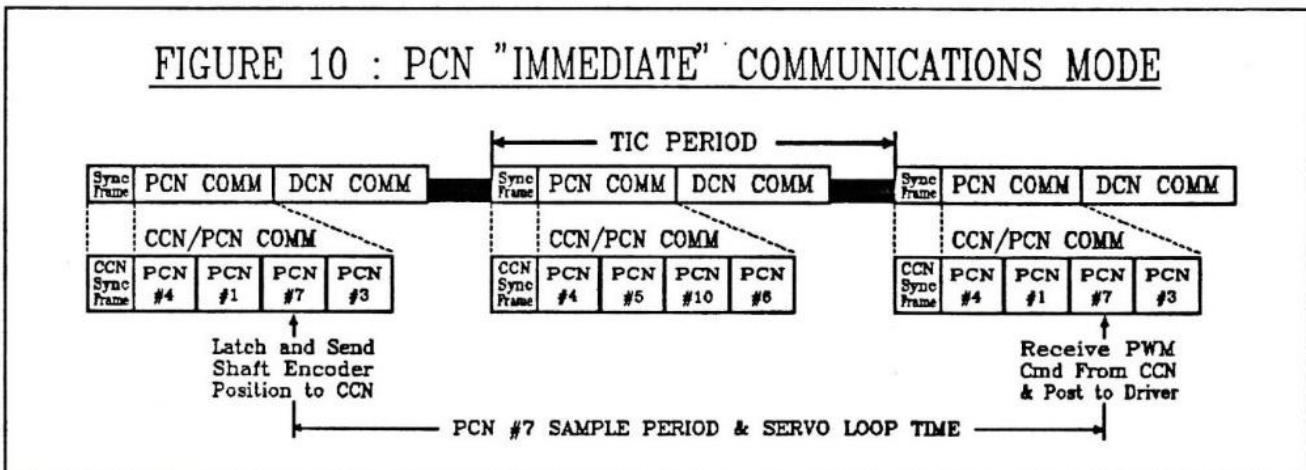


In the synchronous mode, control information passed from the CCN to the PCN is temporarily placed in a PCN holding buffer until the PCN receives the next CCN Sync Frame. At that time, this holding buffer information is applied to the output of the PCN smart I/O function. Similarly, synchronous sensor information passed back to the CCN is not what is immediately available at the time the PCN-to-CCN communication event occurs, but is based on the state of the the smart I/O function input when the most recent CCN Sync Frame was received. This permits a highly synchronized snap-shot of the state of many sensors in multiple PCNs to be achieved. Figure 9 illustrates this synchronous mode impact on servo operation.

Period. Since this is the synchronous mode, the communication between PCN #7 and the CCN can occur within any PCN time slot in the Tic Period. Jitter in PCN #7's sample period is eliminated since capturing of the encoder data and posting of PWM Command data both occur at the end of the CCN Sync Frame. Note, however, that the PCN #7 "Servo Loop Period" is one Tic Period greater than PCN #7's Sample Period. Here, the servo loop time is defined to be the time between when encoder information is captured to when the Servo Filter Result based on that encoder information is given to the PCN #7 PWM Generator Function.

In this illustration, PCN #7 is a single servo motor which is receiving PWM Command Control information from the CCN and providing encoder sensor information back to the CCN. PCN #7 has been selected to have a sample period which is twice the Tic Period and therefore is in communication with the CCN every other Tic

In the immediate mode, control information passed from the CCN to the PCN is applied to the PCN smart I/O outputs immediately. Sensor information available to the PCN smart I/O function at this time is returned to the CCN. This reduces the time it takes for the CCN to find out about a critical sensor event. Figure 10 illustrates immediate mode impact on servo operation.



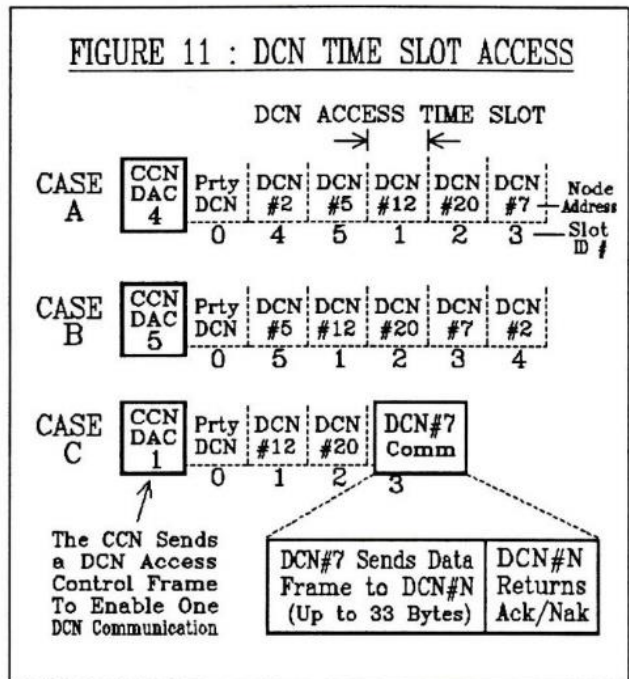
In this figure, PCN #7 is again a single servo motor with a sample period which is twice the Tic Period and therefore is in communication with the CCN every other Tic Period. Since this is the immediate mode, the communication between PCN #7 and the CCN must always occur in the same PCN time slot (#3 was selected) within a Tic Period to eliminate timing jitter in PCN #7's sample period. Although this implies greater scheduling restrictions on CCN-to-PCN #7 communications, it results in the Servo Loop Time of PCN #7 being equal to PCN #7's Sample Period. In some cases, this can improve a motor's performance without the need to increase the sample rate.

In summary, the benefits of the synchronous mode are less restrictive CCN scheduling and the ability to achieve a well timed snap-shot of the sensors of multiple PCNs. The benefits of the immediate mode are to shorten servo loop time and the time it takes to learn about a sensor event. As a final note, Immediate versus Synchronous Modes are designed into smart I/O functions, not PCNs. In this way, a given PCN can have some smart I/O functions which respond in the immediate mode and other functions which respond in the synchronous mode for maximum flexibility.

IPCA DCN PROTOCOL OVERVIEW

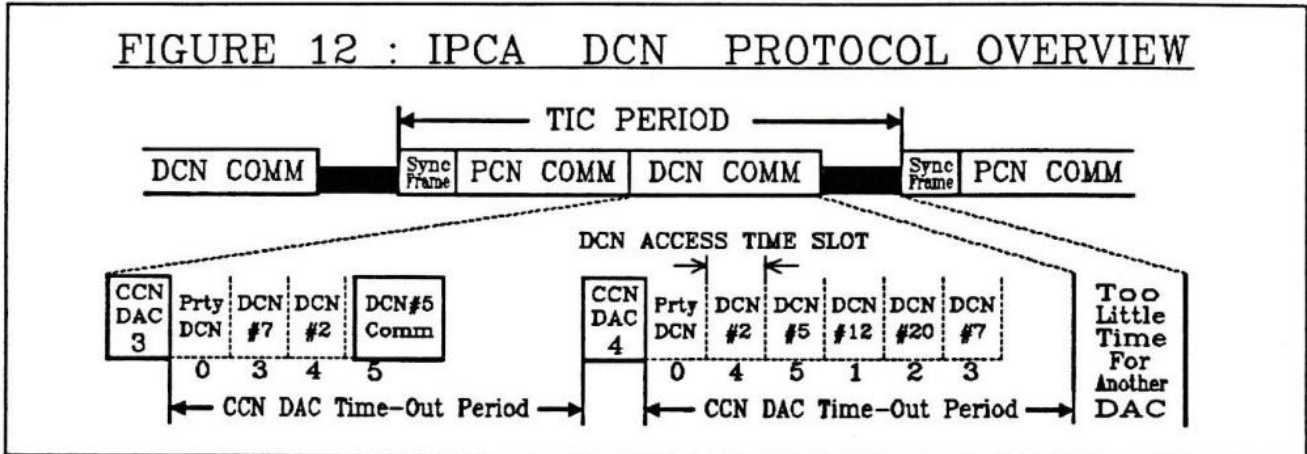
The IPCA DCN Protocol is the enabling feature which allows the IPCA to support mixed mode synchronous and event driven communications. Under CCN control, DCN communications occur in the time remaining after PCN communications have been completed within a Tic Period. This "use what time is available" strategy is acceptable for DCNs because DCN communication is event driven. Whereas PCN communications conform to strict synchronization timing requirements, the critical parameter for DCN communications is message latency. Operating at 10 MBps, DCN-to-DCN communication latency will be less than 5 msec under arduous conditions. In many applications, this latency will be less than 1 msec.

The DCN Protocol employs a Prioritized Slot Access Scheme to enable peer-to-peer communications between DCNs without contention. Each DCN has a unique Slot ID # (different from the Node Address) which is used to reserve the DCN a relative slot access position. As shown in Figure 11, the CCN enables DCN communication by broadcasting a "DCN Access Control" (DAC) frame. Figure 11 shows an example of three consecutive CCN DAC frame broadcasts which would occur on an IPCA network configured with five DCNs having node addresses #2, #5, #12, #20, and #7.



The time immediately following the CCN DAC is broken up into a number of "DCN Access Time Slots". A DCN assigned to a particular access slot will have the opportunity to send a message if that DCN's access time slot has arrived and no other DCN has initiated a message in an earlier access slot. If the DCN in the current access slot sends a message, then DCNs assigned to the remaining access slots will stay quiet until another CCN DAC is sent. If the DCN in the current access time slot does not send a message, then the DCN assigned to the next slot will have the opportunity to do so.

The access slot position immediately following the CCN DAC can be dynamically assigned to any one DCN on the network and is used to allow a critical DCN to assume a high priority access position. The remaining access time slots are used to conduct a round robin priority access in the following manner. Each DCN is assigned a unique Slot ID # (different from the node address). When the CCN sends a DAC, it identifies the Slot ID # with which to begin prioritized slot access (starting in the second access time slot following the CCN DAC). In Figure 11A, setting this "start" Slot ID # equal to 4 results in providing DCN #2 with the highest round robin priority access. Every time the CCN sends another DAC, it always increments this "start" Slot ID # to give each of the round robin DCN's equal access to the network. When it counts up to the maximum Slot ID # value (configurable) it resets to 1.



When a DCN sends a message, the receiving DCN will verify the message and return an immediate acknowledgement. If the sending DCN doesn't get an acknowledgement, it will retry at its next opportunity to send a message. Sequence bits are used between every pair of DCNs to prevent duplicate message errors. This DCN communication process occurs in hardware to enhance network reliability by providing an immediate response capability and eliminating the impact of the DCN CPU and software. The DCN CPU simply loads the communication buffer with a message, sets a flag to activate the communication, and waits for a flag (or interrupt) indicating that the message is transmitted. The receiving DCN will hear nothing until a validated message arrives.

Figure 12 shows how multiple DCN communications are possible within a Tic Period. Following the end of PCN communication, the CCN immediately enables DCN communication by broadcasting the DAC frame along with the appropriate Slot ID #. The CCN then waits a "DAC Time-Out Period". This is the maximum slot time delay period + the worst case DCN message transmission time with acknowledge. After this time-out period, if there is time for another "DAC Time-Out Period" before the end of the Tic Period, then the CCN will send out another DAC to enable another DCN communication. This process repeats until there is not enough time left in the Tic Period, at which time the bus will remain idle until the CCN Tic Period Sync Frame.

NETWORK PERFORMANCE

Significant simulation tools have been developed for the IPCA. At 10 MBps, the IPCA will support a 1 msec average data refresh for 31 PCNs. Where there is a mix of DCN and PCN nodes, the worst case DCN message latency will be less than 5 msec assuming 30% of the network bandwidth is allocated to DCN communications. In most cases, this latency is less than 2 msec.

COMMUNICATION MEDIA INDEPENDENCE

Flexibility has been provided for the IPCA protocol to be implemented in a multidrop or ring topology. Few restrictions exist on the selection of the communications media. This allows the media to be optimally selected so as to minimize the IPCA's cost/performance.

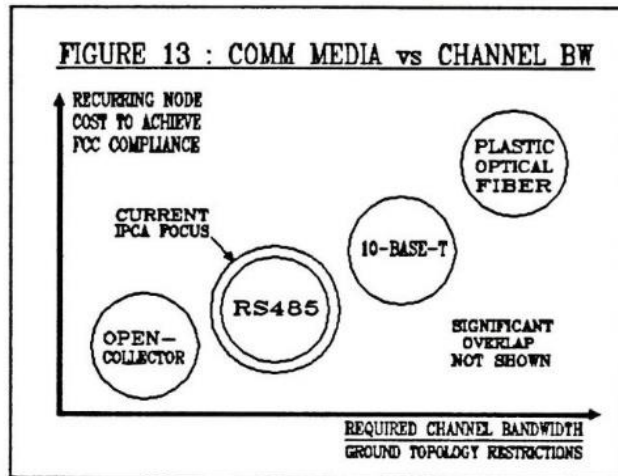
The critical media parameter required to adjust to different communication medias is the "Network Response Time". This is the time for data to reach all of the nodes on the network and return to the sender. The Network Response Time, which is typically between 1 and 10 usec, is affected by :

1. The communications baud rate,
2. The media delay time,
3. The media interface circuitry response time,
4. Clock resynchronization timing requirements, and
5. The IPCA communication logic response time (which is usually small since real time communication logic is implemented in hardware).

To modify the IPCA to accommodate different media requirements, the DCN Slot Time is configurable in the CCN and DCNs, and the wait time for a PCN return message to the CCN is configurable in the CCN. At power-up, these times default to the longest time available. During network initialization, the CCN adjusts these times before real-time communications begin.

Although the communications baud rate is the most obvious factor driving media selection, another major factor is the system grounding topology. The ability to select a less expensive media for a given channel frequency is enhanced by the IPCA's product focus. Node interfacing requirements can be reduced by applying grounding topology restrictions to applications where IPCA network nodes share a common power distribution system (as they would in a typical product).

Figure 13 illustrates the cost and capability of various communication media. As shown, media cost is a function of its ability to support a given channel bandwidth for a given set of grounding topology restrictions. The current IPCA focus is to understand grounding topology requirements to enable differential transceiver and twisted pair technology to achieve a 10 Mbps baud rate, the highest communications baud rate supported by the IPCA. Based on the product performance requirement, the flexibility has been provided to operate the IPCA protocol with a baud rate as low as 625 KBps.

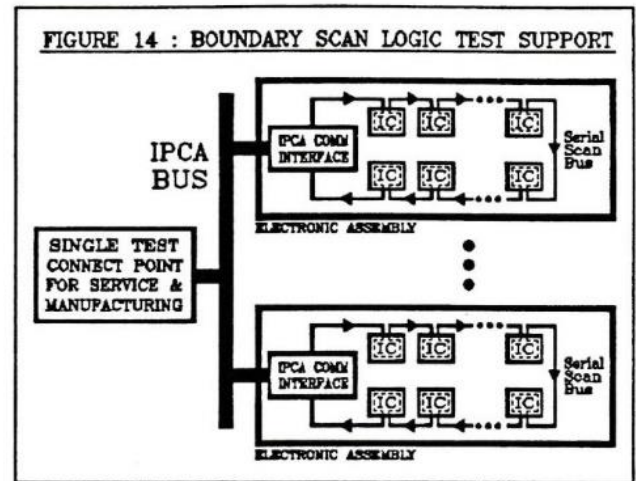


BOUNDARY SCAN LOGIC TEST SUPPORT

The high level development tools used at Pitney Bowes Inc. greatly facilitates customizing the IPCA to incorporate features which will reduce product life cycle costs. For example, the IPCA specification incorporates the capability to diagnose electronic assemblies over the network. A single IPCA node (ie. manufacturing test equipment, service laptop PC, or an internal controller) can orchestrate the testing of all IPCA nodes using Boundary Scan Logic Test Technology.

Boundary Scan Logic Testing is an emerging technology which can reduce "Bed of Nails" test requirements. It allows test equipment to attach to an electronics assembly via a dedicated connector so as to examine that assembly for shorts, open circuits, and failed IC components. By providing access to the Boundary Scan Logic over the IPCA network, the need for the dedicated connector interface is eliminated and all network nodes can be tested from a single connection to the product.

As shown in Figure 14, a serial scan bus is daisychained among a board's "Boundary Scan Compatible ICs". This scan bus begins and ends in the IPCA communication logic interface IC. While in the final product con



figuration, manufacturing test equipment can connect to the IPCA network to diagnose system electronics to the failed IC level. Installed products can be diagnosed from an internal node with built-in test capabilities or from an external service port node. This will allow field service representatives to locate product electronic failures much faster and more accurately than can be achieved today. Although the number of IC components that conform to this standard is not yet significant, it is expected to grow dramatically over the next five years.

TYPICAL PRODUCT APPLICATION

In a very general way, the communication requirements found within a motion control product can be partitioned into data processing communications and motion control communications. Data processing communications typically occur at least once per machine cycle and carry a communication latency requirement of about half a machine cycle (50 -to- 300 msec). On the other hand, motion control communications are tightly coupled to the mechanical performance of the product and typically require a communication latency of less than 5 msec.

Figure 15 provides an illustration of a motion control product broken down in this way. Here, the product's System and Motion Controllers are shown as a single entity (there is no compelling reason why they couldn't have been separate). Also shown in the figure are the multitude of point-to-point connections and junction boards commonly found between the motion controller and peripherals such as motors, sensors, solenoids, indicators, etc. Needless to say, the harnessing challenge that has resulted from the increasing application of digital servo techniques to motion control problems has advanced a strong need for innovative solutions.

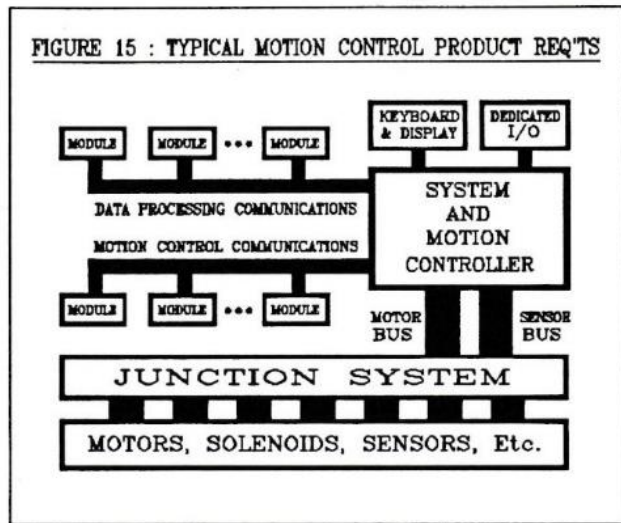
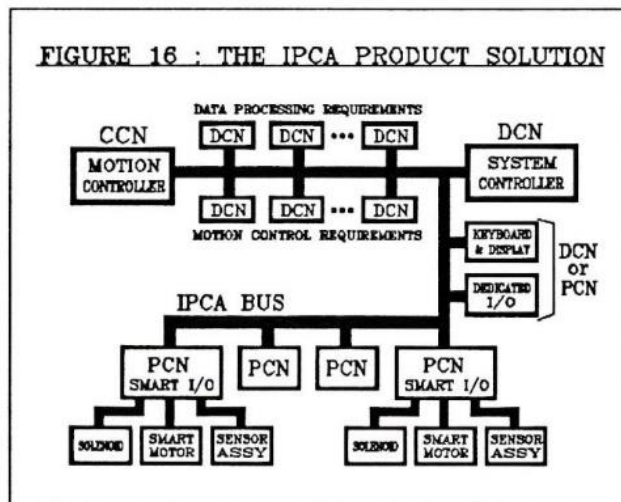


Figure 16 shows the impact of basing this generalized motion control product control system on the IPCA. Advantages of the IPCA implementation include :

1. That the data harnessing system is greatly simplified.
2. Modules all share a common network with a communications latency generally less than 2 msec.
3. The high modularity level achieved will promote reuse of modules and designs in future products.
4. Once FCC compliance becomes predictably achievable for the IPCA network, the IPCA impact of reducing the number of connections into each module down to one should reduce electronics assembly complexity in achieving system level FCC compliance.



CONCLUSION

The motion control industry appears to be focusing on low volume, custom applications. With the high mark-up afforded by these value added applications, there has been little incentive to reduce motion control costs. As a result, existing motion control technology costs have prohibited its use in high volume, commercial product offerings. The IPCA presents an excellent opportunity to reduce motion control costs in existing applications. Moreover, its requirements have been geared to enable a growth in higher volume, commercial products; thereby promoting a broader and stronger motion control industry. Longer term, it is anticipated that the emergence of low cost motion control technology offerings based on a modular control architecture will act in concert with maturing artificial intelligence technologies to stimulate a renaissance in the domestic robotics industry.

We appreciate your interest in our communications architecture and welcome proposals for the commercialization and proliferation of the technology by way of standardization and the further development of motion control products (such as boards, tools, and chip sets).

ACKNOWLEDGMENT

The author would like to thank David K. Lee, David W. Riley, and Frederick W. Ryan, Jr. for their valuable contributions in the development of the IPCA and also William Berson, John M. Gomes, James L. Harman, and Robert H. Whisker for their ardent support.

REFERENCES

1. Chris Bardwell-Jones, SSD Corporation, Reston, Virginia, "SSD LINK; The New Fiber Optic DCS Standard for Industrial Drives & Process Controls", presented at the 3/19/91 Boston Motion Control Conference.
2. Steven Cortese, Rexroth Corp., Indramat Div., "SERCOS - Real Time Communication Between CNC's and Digital Drives", 3/19/91 Boston Motion Control Conference Proceedings.
3. Thomas Bucella, Teknic Inc, Rochester, NY, "Using the Trajectory Regulator Protocol", 3/19/91 Boston Motion Control Conference Proceedings.
4. "IBM Synchronous Data Link Control", General Info., IBM Form GA27-3093.
5. IEEE Standard 1149.1 - 1990, "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE Inc., 345 East 47th Street, NY, NY, 10017.