SAE TECHNICAL
PAPER SERIES

940142

ACP Ford's Audio Control Protocol

Bruce D. Emaus
Consultant to Ford Motor Co.

Reprinted from: Multiplexing and Fiberoptics (SP-1012)

SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL

**International Congress & Exposition
Detroit, Michigan
February 28-March 3, 1994**

400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. Tel: (412)776-4841 Fax:(412)776-5760

ISSN 0148-7191

Printed in USA 90-1203C/PG

940142

ACP Ford's Audio Control Protocol

Bruce D. Emaus
Consultant to Ford Motor Co.

Abstract

A new network solution has been developed and implemented by Ford Audio engineering to support several upcoming distributed audio product programs.

Organized around the seven layers of the OSI model and implemented primarily in software, ACP is based on the physical layer of SAE J1708 (same as Ford's DCL) and a portion of the SAE J2178 message structure.

ACP's UART-based protocol depends heavily on a conversation structure that utilizes message coupling to provide reliable distributed control functionality.

The use of controlled network traffic, a collision avoidance technique, and a structured message language enhance the efficiency of this new network realization.

Background

The requirement to advance toward distributed entertainment product architectures plus the desire to maintain low cost gave birth to Ford's Audio Control Protocol.

As audio system designers advanced beyond the early use of RS-232, an engineering study to evaluate several suitable network-based communication methods subsequently targeted the heavy truck SAE J1708 protocol as the most appropriate candidate.

The physical layer, an attractive low component count version of an industry standard RS-485 interface, was implemented along with the J1708 network access scheme.

Although an earlier ACP version used J1708-type messages, the message structure was replaced by the industry standard SAE J2178 message structure, the same strategy found in Ford's J1850 implementation called SCP.

Organization of ACP Network Information

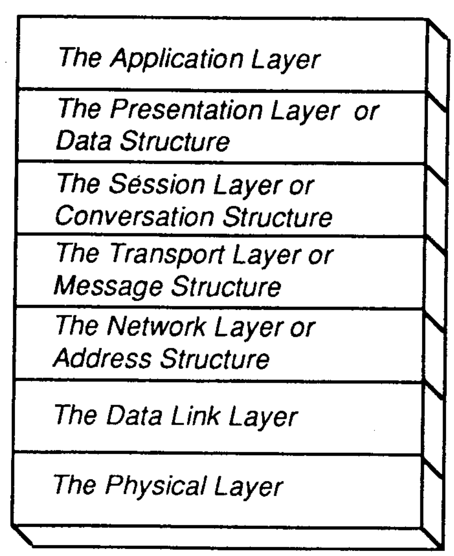
The International Standards Organization (ISO) has established an Open Systems Interconnection committee which fosters a Basic Reference Model for network architectures.

This model has organized networking into a collection of structured functions without implementation detail and has partitioned the data communication activities into seven functional layers.

The OSI model has become a worldwide standard and has gained widespread support by the computer and telecommunications industries and academia. Several earlier proprietary network implementations (i.e. MAP and TOP) have been converted toward the OSI model.

The functional design, implementation, and testing requirements for the ACP network have been strongly influenced by the OSI model. For this paper, the description of the salient features of ACP will follow the model's informational organization.

The OSI Model



The Physical Layer

The ACP Physical Layer establishes the network topology, the network interface, the electrical and mechanical connection, the form of media information, the signal representation, the network interface circuitry and its behavior, and other related characteristics.

Network Topology
Bus or Tree

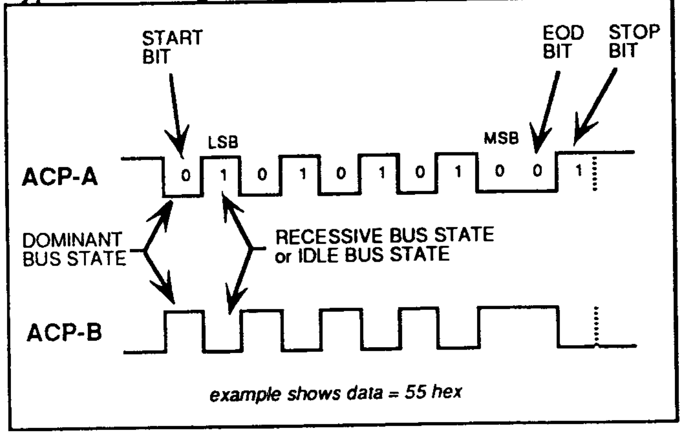
Transfer Medium
Network Wiring
  2 Wire, Twisted Pair
  Maximum Cable Length = 40 meters

ACP Bus Signal Names
  ACP-A (Bus Signal Line A) and
  ACP-B (Bus Signal Line B)

Media Information
Communication Encoding Scheme
  Serial Communications, UART based
  UART format
    1 Start Bit, 9 Data Bits, No Parity, 1 Stop Bit
    Data bit sequence is LSB to MSB
Representation of Logic "0"
  Active Media (Bus) State
Representation of Logic "1"
  Passive Media (Bus) State

Media States
Number of Media (Bus) States = 2
  1 Passive State and 1 Active State
Passive Media (Bus) State
  Bus is Passive
  All Transceiver Drivers are Tri-stated
  Signal Line ACP-A Voltage = 5 volts
  Signal Line ACP-B Voltage = 0 volts
Active Media (Bus) State
  Bus is Active
  Transceiver Drives Both Bus Lines
  Signal Line ACP-A Voltage = 0 volts
  Signal Line ACP-B Voltage = 5 volts
Idle Bus = Passive Media (Bus) State
Recessive Bus State = Passive Media (Bus) State
Dominant Bus State = Active Media (Bus) State

Typical ACP Signal Representation



example shows data = 55 hex

Media (Bus) Load Characteristics
Maximum Media (Bus) Load = 20 nodes
Media (Bus) Termination = used on every node
Network Interface Termination
  Signal Line ACP-A Pull-up Resistance = 4.7 K
  Signal Line ACP-B Pull-down Resistance = 4.7 K

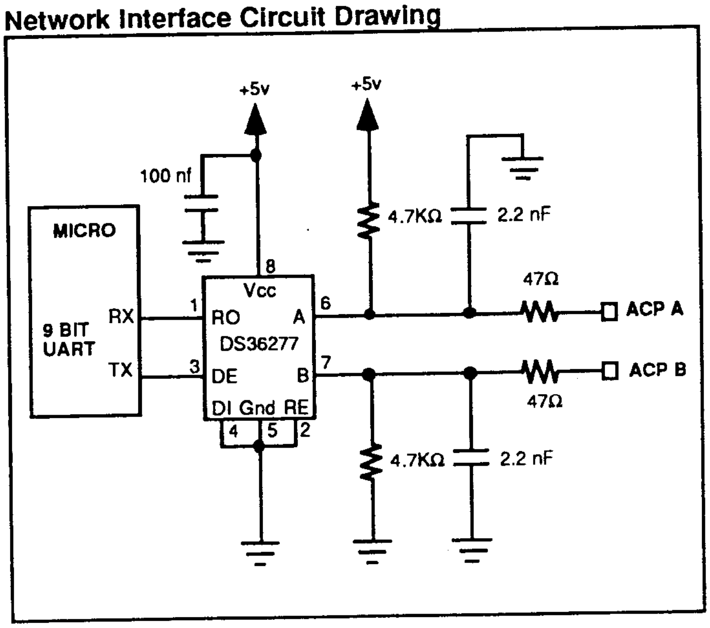
Media (Bus) Signal Characteristics
Voltage Drive Characteristics
  Differential Voltage Drive
  Standard RS-485 Drive Characteristics
  Short Circuit Limited
    To Ground
    To +12 Volts
Voltage Receive Characteristics
  Differential Voltage Receive
  Standard RS-485 Receive Characteristics
Media (Bus) Wake-up Signal = not used
Media (Bus) Ground Shift Tolerance = 7 volts

Media Signal Timing
Bit Rate = 9600 bits per second
Bit Rate Tolerance = 1 percent
Bit Time Interval (BTI) = 104 microseconds

Network Wiring Fault Tolerance
For Single Wire Faults
Survives Short to Ground = Yes
Survives Short to Battery = Yes
Survives Bus Shorted Together = Yes
Communicates With Open = No*
Communicates With Short to Ground = No*
Communicates With Short to Battery = No*
Communicates With Bus Shorted Together = No
*capability under study

Media Interface Circuitry
Circuitry Basis
  SAE J1708 or RS-485
Protection Components = optional
EMC Related Components = optional

Network Interface Circuit Drawing



The Data Link Layer

The Data Link Layer defines the network access control method, the timing of network access control, the rules of transmitting information, the rules of receiving information, and other network access related characteristics.

Access Control Method (reference 3)
Uses Distributed Transmission Control
Supports Carrier Sense Multiple Access
 with Collision Avoidance (CSMA/CA)
Carrier Sensing Technique
 Uses Detection of Idle State
 or Idle Detection

Access Control Information
Minimum Message Size = 6 Message Bytes
Maximum Message Size = 12 Message Bytes

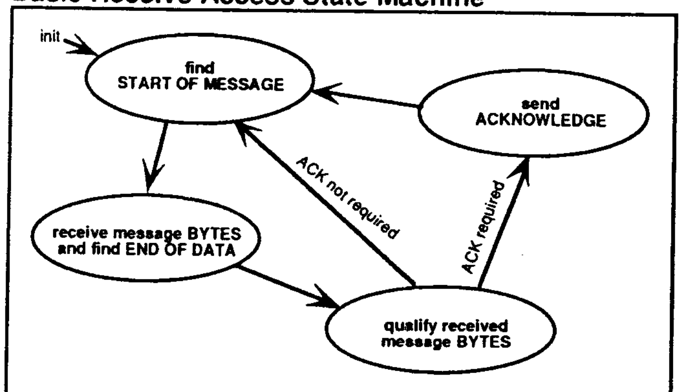
Access Control Semaphores
Start of Message = Minimum Idle Interval (14 BTI)
End of Data = 9th data bit equal to "1"
End of Message = semaphore not used

Receive Access Control
Receive Synchronization Basis
 Idle Detection
Receive Synchronization Reference
 Last Received Stop Bit
 or Upon Power-Up Initialization
 or Network Re-initialization
Receive Idle Interval determines Start of Message = Yes
Maximum Intra-byte Gap = 11 BTI
Supports Receive Acknowledgment = Yes

Receive Access Control States
ACP Receive Access Control States
 Ready / Not Ready for Receive Message
 UART Receive Buffer Ready
 Start of Message
 End of Data
 End of Message
 Keep Message Byte
 Discard Message Byte

Receive Buffer
Adjustable Receive Message Buffer = Yes

Basic Receive Access State Machine



normal receive process only

Transmit Access Control
Transmit Synchronization Basis
 Idle Detection
Transmit Synchronization Reference
 Last Received Stop Bit
 or Upon Power-Up Initialization
 or Network Re-initialization

Transmit Access Priority
Transmit Idle Interval establishes Message Priority = Yes
Elements of the Priority Scheme
 Used to Avoid Collisions = Yes
 Number of Message Priority Levels = 4
 Shift to Random Priority If Collision Detected = Yes
 Number of Random Priority Levels = 4

Transmit Idle States
Number of ACP Transmit Idle States = 6
ACP Transmit Idle States
 Transmit Access Not Allowed
 Transmit Idle State: Priority 0
 Transmit Idle State: Priority 1
 Transmit Idle State: Priority 2
 Transmit Idle State: Priority 3
 Transmit Access Absolutely Allowed

Transmit Idle Intervals
Number of Transmit Idle Intervals = 4
 Prioritized Transmit Wait Interval 0 = 19 BTI
 Prioritized Transmit Wait Interval 1 = 22 BTI
 Prioritized Transmit Wait Interval 2 = 25 BTI
 Prioritized Transmit Wait Interval 3 = 28 BTI

Collision Detection and Avoidance
Supports Collision Detection = Yes
Collision Detection Method = Transmit UART Buffer Not
 Equal to Receive UART Buffer

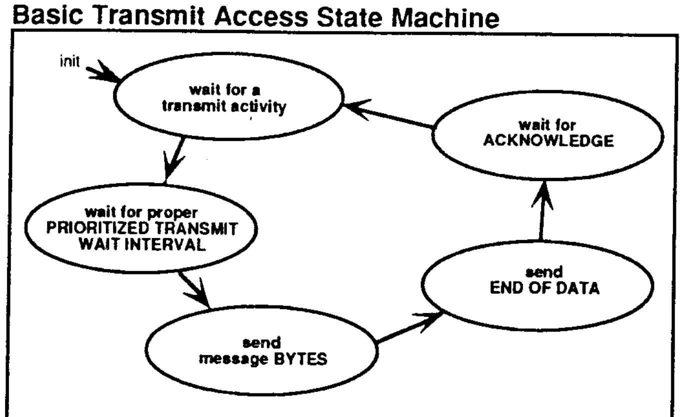
Retry Transmission After Collision Strategy = Yes
Maximum Number of Transmission Retries = 5

Supports Collision Avoidance = Yes

Collision Avoidance Method
 Check Receive Pin Before Starting Transmit

Transmit Buffer
Adjustable Transmit Message Buffer = Yes

Basic Transmit Access State Machine



normal transmit process only

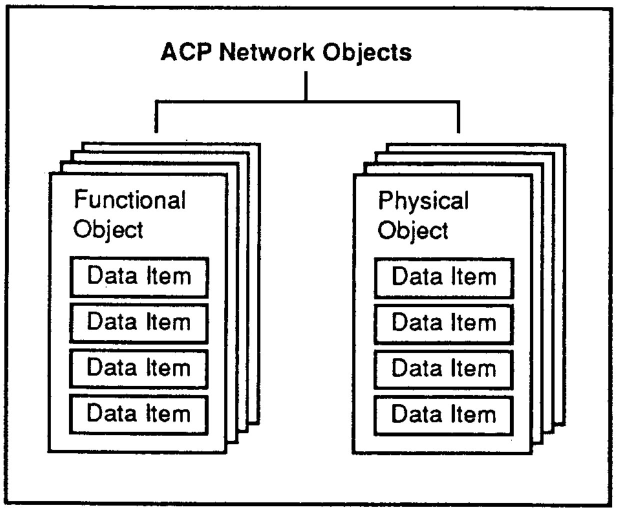
The Network Layer - The Address Structure

The Network Layer defines the method of addressing network entities. The ACP Address Structure is based on SAE J2178 and supports both functional and physical objects. The address structure is essentially the method of pointing at data contained within the system.

ACP Network Address Hierarchy
First Level of Address Hierarchy
  Network Objects
Second Level of Address Hierarchy
  Data Items

ACP Network Object Types
Types of Supported Network Objects
  Functional
  Physical

Diagram of the ACP Address Structure



Address Size Information
Maximum Number of Addressable Objects
  128 Functional Network Objects
  128 Physical Network Objects
Maximum Number of Data Items
  64 Data Items Within a Single Functional Object
  64 Data Items Within a Single Physical Object

Address - Data Item Relationship Rules

1. For all messages, the ACP address always points at the location of the designated data item. This data item location is called the Data Home. See the Presentation Layer for more information related to the Data Home concept.

2. For the ACP network, the Data Home or that object which contains the data item is always responsible for the transmission of the Message ACKNOWLEDGE.

Address Information Within ACP Message

Location of Address Type
  Bit 2 of Message TYPE Byte
    For Functional Address Bit 2 = 0
    For Physical Address Bit 2 = 1

Location of Network Object Address
  Upper 7 bits of Message TARGET Byte
  Name of Network Object Address
    Primary ID

Location of Data Item Address
  Lower 6 bits of Message DATA (1) Byte
  Name of Data Item Address
    Secondary ID

The Transport Layer - The Message Structure

The Transport Layer defines reliable message transfer, the rules of proper message transfer, the use of the checksum as a message qualifier, and other message related characteristics. The ACP Message Structure defines all supported message types and the definition of message bytes.

ACP Messages Types
Supports Functional Broadcast Messages = Yes
Supports Physical Broadcast Message = Yes
Supports Node-to-Node Message Type = No
Supports Function Read Message Type = No

ACP Message Content Information
Message Contains Address Information = Yes
Message Contains Command Information = Yes
Message Contains Data = Dependent on Command
Message Contains Start of Message Semaphore = No
Message Contains End of Data Semaphore = Yes
Message Contains End of Message Semaphore = No
Message Contains Message Priority Information = Yes
Message Contains Checksum = Yes

Broadcast Message Format

1st Message Byte = PRIORITY/TYPE
2nd Message Byte = TARGET
3rd Message Byte = SOURCE
4th Message Byte = DATA(1)
Optional Message Bytes = DATA(2) ... DATA(7)
2nd to Last Message Byte = CHECKSUM
Last Message Byte = ACKNOWLEDGE

Example of an ACP Message

Typical ACP Message with no data item content

PTYPE	TARGET	SOURCE	DATA(1)	CS	ACK

PRIORITY/TYPE Byte Content

PRIORITY/TYPE (or PTYPE) is the first byte of the ACP Message and contains the following elements.

PTYPE contains the Message Priority which indicates the level of priority for the message. Message Priority, is primarily designed to serve the bit-wise arbitration requirements of J1850. However, ACP cannot perform bit-wise arbitration and therefore, the transmission of this information on the network is not important. As indicated in the Data Link Layer, actual ACP message priority is established by coordinating the access to the network using a set of priority-related Bus Idle Time Periods.

PTYPE contains the Message Type to indicate the message format. While J2178 provides a variety of different message types, ACP only utilizes a single message type, that is, the broadcast message.

PTYPE contains the Address Mode to indicate the addressing method associated with the message. ACP provides two different address modes, functional addressing and physical addressing.

Bit Content of PRIORITY/TYPE

Bit 7 = 0 Always
Bit 6 = 1 Required By Vehicle Systems Group
Bit 5,4 = Message Priority
  00 = Highest
  01 = Medium High
  10 = Medium Low
  11 = Low
Bit 3 = 0 Always
Bit 2 = Addressing Mode
  0 = Functional Addressing
  1 = Physical Addressing
Bits1,0 = Message Type
  00 = not supported
  01 = Broadcast Type (if bit 2 = 0)
  10 = not supported
  11 = Broadcast Type (if bit 2 = 1)

TARGET Byte Content

TARGET is the second byte of the ACP Message and contains the following elements.
Primary ID
W Bit

Using the upper-most 7 bits, the Primary ID represents the most significant portion of the network member's address. Supporting either functional or physical addressing, the Primary ID points at the network object or entity which contains the designated data item. Used in association with the Secondary ID, a complete network address is formed that directly points at a designated data item. The Secondary ID, the remaining portion of the network address, is located in the first data byte of the message.

Using bit 0, the W Bit represents a single piece of information that directly relates to a desired message operator or command. The W Bit makes the following data operator distinctions. The W Bit is set to a zero to indicate the "Set or Load" command or set to a one to indicate either a "Request" or "Report" operation. Refer to Structured Message Language Commands. This bit carries no addressing information.

Bit Content of TARGET

Bit 7,6,5,4,3,2,1 = Primary ID
Bit 0 = W = Command/Status Bit
  0 = Set Or Load
  1 = Request Or Report

SOURCE Byte Content

For ACP, SOURCE is the third byte of the ACP Message and contains the address of physical network module which sent the message.

Bit Content of SOURCE

Bit 7 = 0
Bits 6,5,4,3,2,1,0 = Physical Address

DATA(1) Byte Content

DATA(1) is the J2178 designated name of the fourth byte of the ACP Message. It contains the following elements.
Secondary ID
Q Bit
C Bit

Using the lower 6 bits, the Secondary ID represents the least significant portion of the network address. The Primary ID points at the network object while the Secondary ID points at a unique data item located inside the network object.

The Q Bit , located in bit 7, and the C Bit, located in bit 6, combine to represent additional message operator or command information. Together, the Q , C, and W Bits form all data operator distinctions. Refer to Structured Message Language Commands for additional information.

Bit Content of DATA(1)

Bit 7 = Q = Parameter/Quantity Bit
  0 = means Off or Disable or False or Decrement
  1 = means On or Enable or True or Increment
Bit 6 = C = Command Type Bit
  0 = means Load Command
  1 = means Modify Command
Bit 5,4,3,2,1,0 = Secondary ID

DATA(2) ... DATA(7) Byte Content

DATA(2), DATA(3), DATA(4), DATA(5), DATA(6), & DATA(7) are the J2178 designated names of the fifth through tenth bytes of the ACP Message. These message data bytes contain application information and are transmitted only for two parameter message types; the Load Message and the Report Message.

The number of message data bytes transmitted is not fixed at six bytes, the actual size depends on the application's representation of the data item. If one byte is required, one byte is used; if two bytes are required, then two bytes are used. For example many application data items are defined as a single byte and thus, only DATA(2) is used.

Refer to the ACP Data Structure for more information concerning data items and their content.

CHECKSUM Byte Content

CHECKSUM (or CS) is the designated single byte that immediately follows the message data byte(s) of the ACP Message. This byte is used to qualify the integrity of a message and is always added to a message to increase the probability of transmitting (or receiving) a correct ACP message transfer.

Calculated using binary arithmetic, the eight bit checksum is computed by adding all the following message bytes.

Type
Target
Source
Data(1)
and if used ...
Data(2), Data(3), Data(4), Data(5), Data(6), Data(7)

Checksum Computation

CHECKSUM = TYPE + TARGET + SOURCE + DATA

A properly addressed receiver will compute a Checksum for any incoming message and compare this calculated value to the transmitted Checksum that it receives. If these two values match, the entire message content is accepted and assumed to be correct. If these two values do not match, the entire message content is discarded.

End Of Data Semaphore

For ACP, an End of Data Semaphore is included within the message. End Of Data is indicated by the 9th data bit being set to a one.

The End of Data Semaphore only occurs in the message byte called CHECKSUM.

End Of Data signifies that all of the following message bytes have been transmitted or received.

Type
Target
Source
Data(1)
and if used ...
Data(2), Data(3), Data(4), Data(5), Data(6), Data(7)
plus
Checksum

Implementation of End Of Data required that all other message data bytes with the exception of Checksum have their corresponding 9th data bits set to a zero.

ACKNOWLEDGE Byte Content

ACKNOWLEDGE is the last designated ACP message byte used to signify the successful reception of a message.

The object which successfully receives the message and sends the Acknowledge is named the acknowledger. The acknowledger is established as being that resident network entity that has been addressed by the Primary ID of the TARGET byte.

Acknowledgment is indicated only after the following sequence of message transfer events has occurred.

1. The specified network object that is addressed by the Primary ID of the TARGET byte is resident on the network.

2. The received message meets all of the data link receiving rules and possesses no detected receiver error.

3. The computed Checksum for the received message is equal to the received Checksum.

Once all three conditions have been satisfied, the acknowledger will transmit the Acknowledge byte immediately following the received Checksum. For ACP, the Acknowledge byte content is always equal to 06 hex.

The Valid ACP Message

While the acknowledgment sequence serves as an important portion of the transfer process, all message transport qualification rules must be satisfied before a receiver can assume that a valid message transfer has occurred.

Message Transport Qualification Rules

A Valid Message Must...

1. Begin With a Proper Idle Interval to indicate Start of Message (14 BTI)

2. Satisfy All the Data Link Access Rules

3. Include a Correct Type Byte (Type 7 or Type 1)

4. Include a Resident Network Object Address (Proper Primary ID)

5. Include a Correct Command (Valid QCW bits)

6. Include a Resident Data Item Address (Proper Secondary ID)

7. Include a Correct Checksum

8. Include the Correct Data Size

9. Include a Proper End of Data Semaphore (CHECK bit 9 = 1)

The Session Layer - The Conversation Structure

The Session Layer establishes the structure of network conversation or message transactions, the meaning of conversation and its rules, organization of message dialog, the timing and synchronization of conversation, the rules for using networks for either closed-loop or open-loop functions, and other dialog related characteristics.

The Message Coupling or Transaction

Most conversation on the ACP network involves the use of a pair of messages. One example of this is the request message that is followed by a responding report message. These two messages represent a message coupling or transaction. Employing message transactions is an integral part of closed-loop control processing over the ACP network.

Conversation Structure Capability

Supports Closed-Loop Control (With Feedback) = Yes
Supports Open-Loop Control (No Feedback) = Yes
Supports Message Coupling = Yes

Conversation Synchronization

Primary Event that Starts Conversation
  Human Caused Event
Secondary Event that Starts Conversation
  Status Information Request
Event that Stops or Ends a Conversation
  Report Message

Two Classes of Conversation

Conversation With Communication Exchange
  Coupled Conversation
Conversation Without Communication Exchange
  Uncoupled Conversation

Starting Conversation

A conversation starts when a system event requires a network transmission. System event types include either a human response event or a timed event.

Stopping Conversation

Once all distributed system processing is complete, the conversation may end. A normal conversation ends with a report message.

Re-synchronizing Conversation

If any extraneous event occurs that inhibits the completion of a conversation, some mechanism must be employed to re-synchronize the conversation. Re-synchronizing techniques include both the required coupling of messages and checking the expected conversation timing .

The Coupled Conversation

Coupled Conversation is considered as a normal communication exchange. It is network dialog which is expected and carries both informational and timing rules.

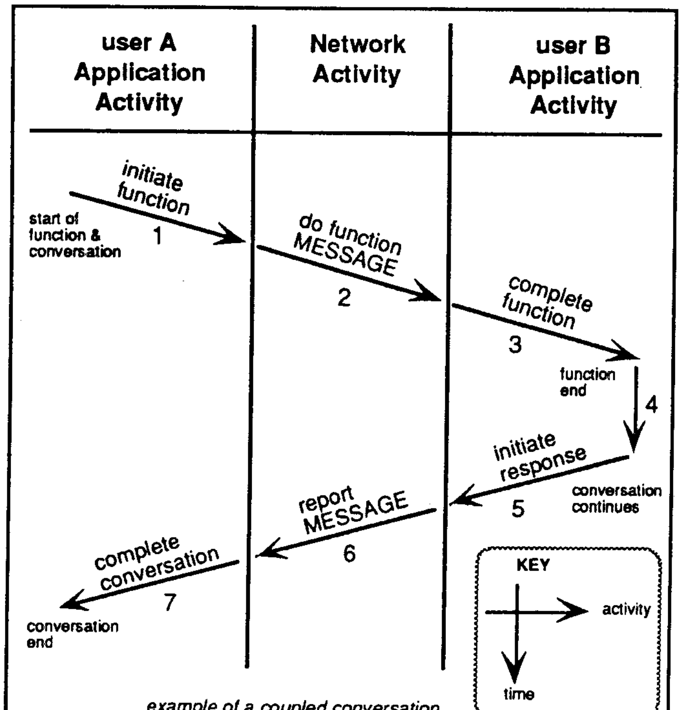
Example of a Coupled Conversation

Because of some system or module event, a conversation begins...

1. Network member, user "A", detects a "function initiating event" and initiates a call to its network transmitter to prepare for the transmission of Message 1.
2. This first message is placed onto the network by user "A" and is simultaneously received by another network member, user "B".
3. User "B", initiates a call to the network receiver to move the received message into its Application software.
4. User "B" interprets the message and completes the function. ("function concluding event")
5. Since user "B" must respond to complete the message coupling, it initiates a call to its network transmitter to prepare for the transmission of Message 2.
6. This responding message is placed onto the network by user "B" and is simultaneously received by network member, user "A".
7. User "A", initiates a call to the network receiver to move the received message into its Application software, interprets, and processes this responding message.

... and the conversation ends.

Graphical Representation



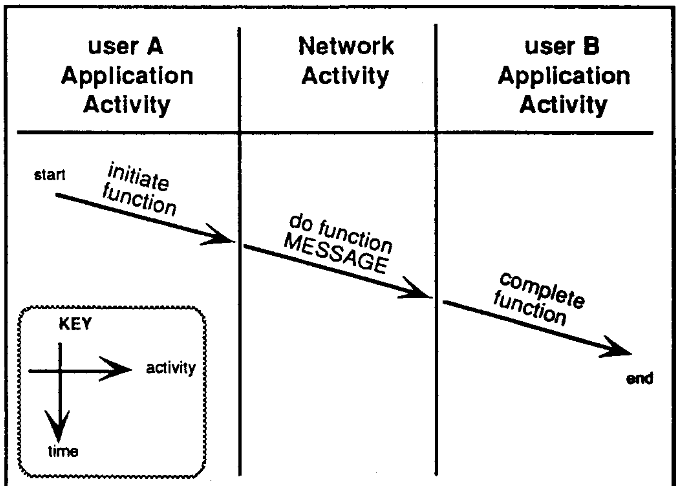
example of a coupled conversation

This pictorial representation, similar to a network service diagram, can be used to show the timed system and network events related to a conversation. The diagram shows the movement of information through various system layers. The vertical axis indicates time and the horizontal axis indicates activity. Each activity is identified as an arrow.

Uncoupled Conversation

Uncoupled Conversation may be used where open-loop functionality is acceptable. This acceptable conversational form is labeled as Proper Uncoupled Conversation.

Example of a Proper Uncoupled Conversation



typical "do function" message (shown with no coupling)

In some cases Uncoupled Conversation is caused by a conversational fault condition. This unacceptable and unexpected conversational form is labeled as Improper Uncoupled Conversation.

Proper Uncoupled Conversation

Proper Uncoupled Conversation is defined as having three types.

1. Condition Report
2. Returned Error Report
3. Spontaneous Error Report

Condition Report

The Condition Report, a type of Proper Uncoupled Conversation, is defined as an expected report of information usually based on a system event or a periodic required event.

No message exchange occurs.

Returned Error Report

The Returned Error Report, another type of Proper Uncoupled Conversation, is defined as an unexpected conversation. For this case, a network member transmits a command, but the receiving network member replies with an "Error Report" rather than the expected normal response. A Returned Error Report is usually invoked because a request cannot be performed.

Causes for Returned Error Report include:

1. The transmitting network member requests a non-existence operation.
2. The receiving network member cannot perform the message.
3. The receiving network member has a condition or fault that interferes with the requested operation.

Spontaneous Error Report

Similar to the Condition Report, the Spontaneous Error Report, a type of Proper Uncoupled Conversation, is defined as an unexpected report of error information usually invoked by a detected condition.

No message exchange occurs.

Improper Uncoupled Conversation

Improper Uncoupled Conversation is defined as having two types.

1. No Response
2. No Report

No Response

No Response, a type of Improper Uncoupled Conversation, is defined as a conversation fault, where a network member talks, but no other network member ever replies.

Causes for No Response are numerous and include:

1. The talking network member is not on the network.
2. The expected receiving network member is not on the network.
3. The network media (wiring) is broken.
4. The talking network member's physical layer has a fault.
5. The expected receiving network member's physical layer has a fault.
6. An excessive amount of noise is present on the network.
7. The expected receiving network member has a microcomputer fault.

No Report

No Report, another type of Improper Uncoupled Conversation, is defined as a conversation fault, where a network member talks and receives an acknowledge, but no expected report ever occurs.

Causes for No Report include:

1. The talking network member is only transmitting a "Report" message.
2. The expected receiving network member does not understand the message.
3. The talking network member gets no response within some time period.
4. An excessive amount of noise is present on the network.
5. Some fault has occurred immediately following the original transmission.

ACP Message Couples

The following ACP messages are classified as suitable for feedback or closed-loop control.

For each initiating message, its coupled response message is listed.

Initiating Message	*Response Message*
Set State	*Report State*
Reset State	*Report State*
Request State	*Report State*
Load Parameter	*Report Parameter*
Increment Parameter	*Report Parameter*
Decrement Parameter	*Report Parameter*
Request Parameter	*Report Parameter*

Note - The data types of state and parameter are described in the Data Structure section.

This list is not meant to imply that each initiating message is guaranteed to be coupled with a responding message. Only the system design strategy document that specifies a closed-loop requirement will determine when coupling is actually implemented.

Messages Without Coupling

The following messages are classified as Messages Without Coupling and not suitable for feedback or closed-loop control.

Initiating Message	*Response Message*
Report State	*no response*
Report Parameter	*no response*

Conversation Rules

ACP Network Conversation Rules are required rules that pertain to the orderly exchange of information.

Rule 1: Human Response Conversation is First, Status Conversation is Last
Rule 2: Guarantee Closed Loop Functionality With a Message Coupling
Rule 3: Answer if Properly Asked
Rule 4: Do Not Misuse the Network By Transmitting Too Fast or Too Often
Rule 5: Use Coupled Conversation With a Timer
Rule 6: For the Conversational Fault called No Response, Use an Appropriate Re-attempt Conversation Period
Rule 7: For the Conversational Fault called No Report, Use an Appropriate Re-attempt Conversation Period
Rule 8: Properly Handle a Returned Error Report
Rule 9: Properly Handle a Spontaneous Error Report

The Presentation Layer - The Data Structure

The Presentation Layer or Data Structure defines the organization of data, the formats of data, and operations on data for the ACP network. The ACP Data Structure is essentially a data based organization of network data.

As described earlier in the address structure, ACP Network Objects include both Functional and Physical Objects. The data items found within these two object types are bound by the rules of the Data Structure.

ACP Network Data Users

For the ACP network, there are three important network data users.

1. Normal Operation
2. Diagnostics
3. Manufacturing

ACP Network Data Items

A data item is a specified piece of information which could be a bit, a byte, or a group of bytes. Any individual Normal Operation, Diagnostic, or Manufacturing data requirement will be supported by a data item. Data items are grouped by object type, located inside assigned objects, and addressed by a portion of the ACP message. Each required data item is named, sized, located, content defined, documented, and published in the associated object's Data and Message Set.

Network Relationships With Data Items

The meaning of network data is heavily influenced by the relationship between a network object and a network data item. The relational distinction of whether you are the home of the data or a controller of the data or just a user of the data is significant. Three data item relationship terms are defined.

1. the Data Home
2. the Data Controller
3. the Data User

The Data Home is that network object which contains the data. A data item can only reside in a single network object. The network address of the data home is designated by a specific network object address (the Primary ID) and a specific data item address (the Secondary ID) that points at the data item located within the object.

The Data Controller is the network object which has the relational authority to change the data content of a designated data item. More than one network member may be authorized as a controller of data. This is especially true when a diagnostic or manufacturing tester is employed.

The Data User is a network object which is a user of a designated data item. Any network object may listen to the data and in order to do so must possess knowledge of the data item's location.

General ACP Network Data Operations

To support the requirement of accessing and manipulating network data items ACP defines three general, but essential, network operators or commands.

1. LOAD
2. REQUEST
3. REPORT

These commands, with some variation and extension to the LOAD command, allow network users the total ability to function within the distributed product environment.

Four variations or extensions to the LOAD operator are defined.

1. SET
2. RESET
3. INCREMENT
4. DECREMENT

Based on the type of data and actual system requirements, each designated data item may or may not support all available data operations.

ACP Data Representation - Format and Operations of ACP Data by Type

The ACP network supports three types of data.

1. the binary state
2. the parameter
3. the file

The Data Type - State

Referred to as "state", the binary state is data represented as a single bit. For each data item that is defined as a state, the following is specified:

1. the state's assigned name
2. the state's assigned secondary address
3. the definition of the state's data content
    a. for set
    b. for reset
    c. for report

Note - An ACP message data size equal to 0 (zero) indicates that the data representation is "binary state" data.

State Data Operations

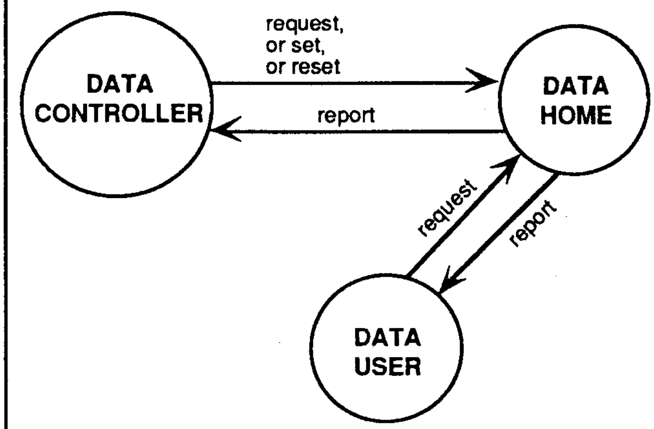
The entire set of data operations for "binary state" data are as follows:

1. Set State
2. Reset State
3. Request State
4. Report State

Note that the operators of "set" and "reset" are variations of the "load" command.

Depending on the data relationship or other system implementation constraints, some data operations may or may not be allowed.

Data Relationships for State



data relationship for state only

The Data Type - Parameter

Parameter data, the most common form of network data, is data represented in byte form which may be sized from one to six bytes.

For each data item that is defined as a parameter, the following is specified:

1. the parameter's assigned name
2. the parameter's assigned secondary address
3. the size of the parameter's data
4. the definition of the parameter's data content
    a. for load
    b. for increment (if applicable)
    c. for decrement (if applicable)
    d. for report

Parameter data may or may not include content values with units. For those parameter-type data items that use units, the following additional rules apply.

1. define the units
2. define the valid range of data values
3. use SAE data content values (when recommended and appropriate)

Parameter Data Operations

The entire set of data operations for "parameter" data are as follows:

1. Load Parameter
2. Increment Parameter
3. Decrement Parameter
4. Request Parameter
5. Report Parameter

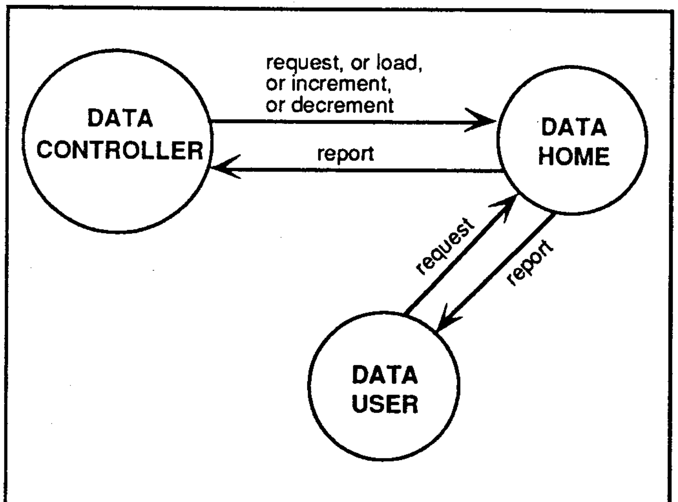
The operators of "increment" and "decrement" are extensions of the "load" command. These two additional parameter commands allow data modification without the burden of communicating data content.

Depending on the data relationship or other constraints, some data operations may or may not be allowed.

Increment and Decrement may be used under the following conditions:

1. The Data Home supports these commands
2. The Data Item's content employs data with units

Data Relationships for Parameter



data relationship for parameter only

The Data Type - File

ACP's use of the broadcast message places a restriction on the amount of network data that may be transmitted within a single message, that is, data must be sized as 0, 1, 2, 3, 4, 5, or 6 bytes. This established limitation presents no difficulty for the vast majority of defined application-based data.

However ACP data requirements sometime exceed the 6 byte limitation of parameter data and therefore, ACP has supported the concept of file data. Based on an extension of parameter data, file data is data presented as an ordered list of information to allow application-based data representations of any size.

File data requires several support features.

1. file data is organized into blocks or records of 4 bytes (parameter data)
2. each block or record is addressable
3. a pointing method is used to allow positioning through the information (file)

The Organization of ACP Network Data

The entire data organization for each designed object, whether functional or physical, is documented as a composite "data and message set".

Classes of ACP Data Items

In general, data items are either specific or reserved.

A "specific" data item is created, named, and assigned to an object. The data item's content is defined by either the Normal Operation subsystem designer, the Diagnostics subsystem designer, or the Manufacturing subsystem designer.

A "reserved" data item is created, named, and assigned to an object with content definition established by the ACP Systems Group.

The Data and Message Set Documents

For the ACP Network, two types of Data and Message Set documents are created, developed, written, published, and maintained by the ACP Systems Group.

1. the Functional Data and Message Set
2. the End Item (Physical) Data and Message Set

The Functional Data and Message Set

The Functional Data and Message Set contains the following information.

1. the functional object's name and
2. the assigned functional Primary ID
3. a list of all functional data items
4. For each data item, the following is specified:
   a. the assigned name
   b. the assigned Secondary ID
   c. the data type
   d. the definition of data content, for load and report

End Item Data and Message Set

The End Item Data and Message Set contains the following information.

1. the end item name
2. the assigned physical primary address
3. a list of all physical data items
   a. Manufacturing-type Data Items
   b. Diagnostic-type Data Items
   c. Physical Reserved Data Items
4. For each data item, the following is specified:
   a. the assigned name
   b. the assigned secondary address
   c. the data type
   d. the definition of data content, for load and report

Data Structure Faults

Because problems will arise if ACP network data structure requirements are not met, the management of data structure faults is a network requirement.

Four different types of data structure faults are defined.

1. Non-existent Data Item
2. Unsupported Data Item
3. Wrong Sized Data Item
4. Invalid Data Item Content

Non-existent Data Item

A Non-existent Data Item is defined as a data item that does not exist. No secondary address or data item address has been assigned nor has any data item definition ever been established.

Unsupported Data Item

An Unsupported Data Item is defined as a data item that is not supported by one network object but is supported by another network object. This condition commonly occurs for configuration dependent data items.

Wrong Sized Data Item

A Wrong Sized Data Item is defined as a data item that is placed onto the network which uses data size different than that defined in the associated Data Set. This condition may be complicated by the additional consideration of whether the data size is configuration or version dependent.

Invalid Data Item Content

Invalid Data Item Content is defined as incorrect content for a specific data item. This means that the data item's value is outside the expected range of valid values.

The Application Layer

The Application Layer defines the rules of interfacing with the network, the initialization and maintenance of upper-level network function, and other application/network related characteristics.

Selecting the viewpoint of the application, the ACP Network interface supports the network activities of transmitting, receiving, and supervision. These activities have been organized into three components.

1. the Network Transmitter
2. the Network Receiver
3. the Network Supervisor

Network Transmitter

The Network Transmitter supports the operation of transmitting messages. The Network Transmitter includes the following elements.

1. The procedure of calling the transmitter
2. The management of transmit status
3. The development and construction of transmit message tables

Network Receiver

The Network Receiver supports the operation of receiving and the management of receive status and receive message tables.

The Network Receiver supports the operation of receiving messages. The Network Receiver includes the following elements.

1. The procedure of calling the receiver
2. The management of receiver status
3. The development and construction of receive message tables

The Network Supervisor

Network supervision activities include initializing network software and hardware, powering up the network (if required), powering down the network (if required), maintaining continued and proper network functionality, managing network fault conditions.

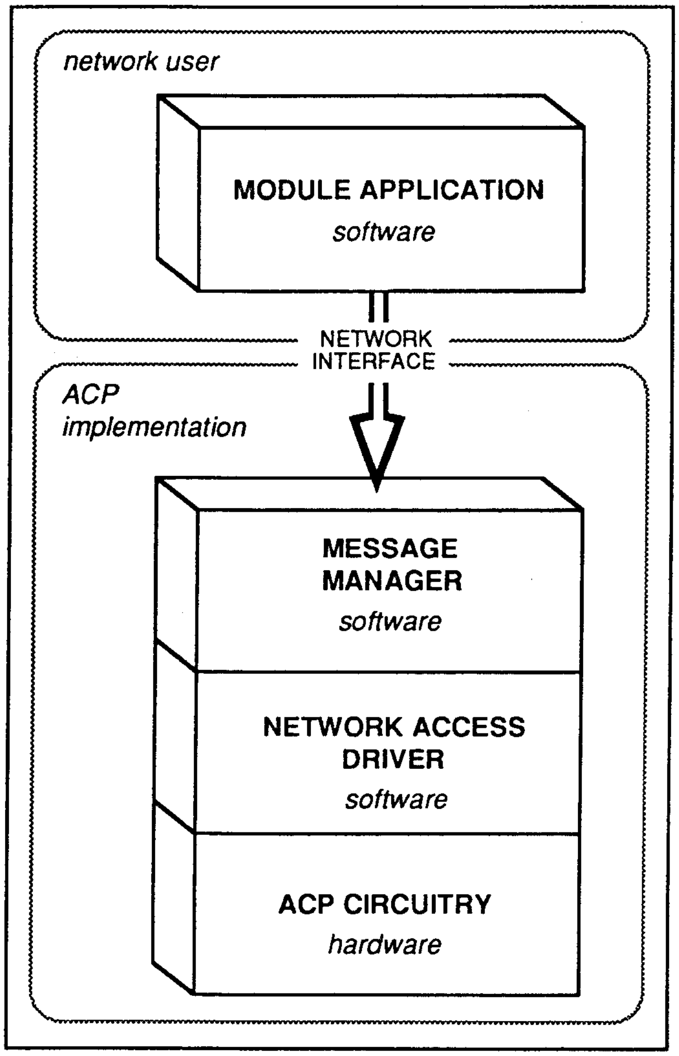
Network Supervision Tasks

- Maintain Network Status
- Network Initialization
- Network Power-Up
- Network Power-Down
- Network Watchdog Timer
- Network Fault Management

ACP Implementation

From the network user's point of view, a network solution could be designed as a so-called "network product", a combinational plug-in software module and plug-in hardware module. This was the systems approach taken by ACP network designers. With ACP, the module designer and application programmer need not become network experts in order to use the network.

To implement this approach, conversion of the ACP design requirements involved constructing ACP as a complete package of hardware, software, and information (data and message sets).



ACP Hardware Implementation

Starting with a micro controller that includes a UART, module designers implement the ACP hardware by adding a common RS-485-type transceiver chip with several passive components. Depending on the electrical environment, optional EMI or transient protection components may be required.

ACP Software Implementation

ACP Network software consists of two integrated software modules; a network access driver and a message manager. These software modules are then integrated with the module application's functional and physical message sets. This combination plus a set of application interface rules places the typical ACP user on the network.

Pre-Tested ACP Software

The ACP software activity has developed, tested, and released several ACP software implementations for several microcomputer platforms. Written in both assembly language and C, these software modules are maintained and periodically upgraded .

ACP Information Implementation - ACP Data & Message Sets

Using the design requirements of the ACP Data Structure, data and message sets have been developed, published, and maintained to support the system and module development of ACP-based products. The following list indicates several of the documented data and message sets.

Functional Data & Message Sets
- Audio Control
- Tuner
- Amplifier
- Audio Button
- Compact Disc
- Climate Control

Physical Data & Message Sets
- Front Control Unit
- Rear Control Unit
- CD Player
- Navigation
- Remote Climate Control

ACP Network Testing

ACP Network Testing is that collection of network-related tests that each module must pass in order to gain membership to the network. This verification process is organized around the seven-layer OSI model and validates hardware, software, information, and strategy conformance.

ACP Tests include:
1. Physical Layer Testing
2. Data Link Layer Testing
3. Address Structure Testing
4. Message Structure Testing
5. Conversation Structure Testing
6. Data Structure Testing
7. Application Interface Testing

Those ACP implementations that pass all testing requirements are released and are subsequently qualified as members on the network.

As an example of this layered approach, let us discuss the Data Structure Testing. The purpose of data structure testing is two-fold; first, to determine the adequacy of an implemented data structure, and second, to verify that no data structure fatal errors are present in any final released function or subsystem. The testing emphasis is on data; that is data items and content.

Data Structure Testing is a collection of related tests directly focused at determining the quality of a given set of network data and is required for each physical node and functional object. The testing process examines the object for each of the following potential data faults.
1. Non-existent Data Item
2. Unsupported Data Item
3. Wrong Sized Data Item
4. Invalid Data Item Content

Performed on a single network node or object basis, data structure testing occurs in the test laboratory or other suitable location, is part of the normal testing process of Module Network DV, and is performed by the ACP Test Group. Upon satisfactory completion of all Data Structure Testing, the ACP Test Group will sign-off on the designated network node or object.

ACP Development and Test Tools

The development of network tools was an essential part of the development of ACP. Several ACP Tools are available and the ACP Tool development has been focused on meeting a wide range of tool user requirements.

ACP Tool Users
- ACP Network Designers
- Module Designers
- Vehicle System Designers
- Module Manufacturers
- Diagnostic Designers

ACP Network Tools support vehicle sub-system checkout, verification of ACP design implementations, bus monitoring, data logging, module development, manufacturing, and diagnostics.

ACP Tool Capabilities
- English-based Message Representation
- Message Reception
- Message Transmission
- Address Qualifiers
- Single Message Editing and Generation
- Multiple Message Scripting
- Message Script Files
- Vehicle Data Logging
- Message Time Stamping

Major ACP Design Considerations

Structured Message Language

Network communications may be considered as conversation using sentences (messages) which contain words (message bytes). To form a better understanding of the meaning of a message, it is appropriate to define the words that make up these sentences such that we may examine the forms of sentences.

For any given network message, there are 3 types of words associated with a message. These words are **address, command**, and **data**. What we want to do implies the **command**. Whom we want to tell about is the **address**. Any additional information we wish to convey is the **data**.

Address, command, and **data** are the important words; they alone convey the meaning of the message. Based on these types of word classes, messages may be organized into a generic set of sentences. This entire organization is called Structured Message Language and is designed with the goal of making network use independent of the actual message structure. This is a very desirable feature. With such a language, system designers need only to learn a set of sentences and need not pay attention to the details of the actual network MESSAGE structure.

Let's examine what the systems designer is interested in doing with the network. The distributed function designer wants to:

1. Do Something
2. Request Information
3. Report Information

Each of these basic sentences is directly related to an action and is classified by **command**. These three types of sentence classes based on **command** are expanded and renamed into single word descriptors. Note that, "To Do Something" has been expanded into SET, CLEAR, LOAD, INCREMENT, and DECREMENT.

This set of commands is sufficient to perform all network functions or distributed actions.

Types of SML Commands

SET	To Set a State On
CLEAR	To Clear a State
LOAD	To Load a Value
INCREMENT	To Increase a Value
DECREMENT	To Decrease a Value
REQUEST	Ask Information About Something
REPORT	Answer with the Requested Information

Address points at the thing which we are trying to communicate.

Types of SML Addresses

STATE
PARAMETER
FILE

Those remote entities or things that are defined as single bits are defined as type STATE.

Those remote data entities or things that are larger than STATE (a single bit) and not requiring a form like FILE (an ordered list) are defined as type PARAMETER. Those remote data collections larger than parameter are defined as type FILE.

The word class **data** is related to the information that pertains to the thing whom we are trying to communicate with or about. **Data** directly influences the content of STATE, PARAMETER or FILE.

Combining all of these word classes, we can define a set of generalized sentence structures that will serve the needs of network conversation.

Structured Message Language Conventions

Basic SML Construct Format
Command(Address,Data)

SML State Constructs
Set State (StateAddress)
Clear State (StateAddress)
Request State (StateAddress)
Report State (StateAddress,StateData)

SML Parameter Constructs
Load Parameter (ParameterAddress,ParameterData)
Increment Parameter (ParameterAddress)
Decrement Parameter (ParameterAddress)
Request Parameter (ParameterAddress)
Report Parameter (ParameterAddress,ParameterData)

SML File Constructs
Load File (FileAddress,FileData)
Increment File (FileAddress)
Decrement File (FileAddress)
Request File (FileAddress)
Report File (FileAddress,FileData)

This simple and comprehensive set of structured network sentences allows network users the total ability to operate distributed functions and to converse with remote modules.

Advantages of Structured Message Language

With Structured Message Language, system designers need only to learn one general purpose set of message constructs. Similar to the consistency provided by the computer industry's standardization of assembly languages, SML places the emphasis on a reusable and generalized sentence-based structure. It allows system designers a considerable amount of independence from the details and complexities of the industry-wide variance in network message organizations.

Implementation of SML on ACP

Currently, ACP supports all SML State and Parameter commands. SML has also been applied to some ACP tools to allow system designers the ability to read messages in English.

Using Conversation For Control

All control systems, whether networking is or is not used, require some form of feedback to maintain proper operation. However, once a distributed system architecture is selected, any required feedback technique will use either a separate return path (typically a wire) or some form of data communications.

In a networked control system, the initiator or originator of a control command must be informed that the intended function has been performed by the receiving entity. Without such confirmation or feedback, the desired function is not under control.

Proper control functionality carries the requirement of response time. When using distributed product architectures, network feedback must deliver adequate time performance for every customer and system demand. To deliver such time performance, the selected network must provide reasonable message speed and properly manage network message traffic.

Some control functions may require a response time that exceeds the capability of the selected network. For these cases, a separate feedback wire may be required or a faster network may be appropriate.

It is also quite important, that network feedback is not necessary established by employing the acknowledgment process. An acknowledgment signifies that a message has been received by someone, it does not indicate that the command has been reliably completed.

When using a network for feedback control, it is important that some form of confirmation must be returned in order to guarantee reliable control. This need for confirmation implies that a transmitted message must be coupled with a return message.

Message coupling is the conversational technique used on ACP to deliver reliable control performance.

For each type of control function or command, there may be a unique form of confirmation. One or more REPORT messages may be required to perform a given control function.

Some commands or information messages may not even require a message transaction. This may depend on whether the desired function requires closed-loop control.

System Latency and ACP

For system designers to properly apply a network, the composite latency of the distributed system must be characterized as well as the timing aspects of the network.

Many network solutions are compared by the speed at which the data moves across the network wires without taking into consideration the additional amount of microcomputer processing time required to do all of the other related distributed system tasks.

Using the system point of view, a distributed function timing characteristic that includes the network timing can be developed and used to quantify the performance of the overall distributed system.

The Distributed Function Time

Confined to the boundaries of the electrical system and not including the mechanical system, the Distributed Function Time is defined as that time duration between the following two events.

1. The Function Initiating Event
2. The Function Concluding Event

Between these events, a distributed system must coordinate an elaborate sequence of application and network activities that will completely process the entire function from its inception to its completion.

Computation of Distributed Function Time

The Distributed Function Time (DF) is equal to the sum of the following combination of application and network timing terms:

1. the Transmit Latency time (TL)
2. the Communication Time (CT)
3. the Receive Latency time (RL)

or as the time equation:

$$DF = TL + CT + RL$$

The **Transmit Latency** time (TL) is that time duration between the beginning of the Function Initiating Event to the time at which the network transmission process begins. This time is directly proportional to the internal micro controller program organization and timing. Because this time is not a network timing characteristic, the requirement is established by the system designers.

Including both Idle Time, Message Time, and a possible collision-caused Message Retry Time, the **Communication Time** (CT) is defined as the total time duration that is required of a transmitting station to gain access of the network and to transmit a message.

The Idle Time or Idle Interval is a specific time duration of the idle condition employed to determine either readiness for message transmission or reception.

The Message Time is that time duration between the beginning of the Start Bit of the first message byte to the end of the Stop Bit of the Acknowledge. The Message Time is directly proportional to the Message Data Length.

If a collision occurs, a subsequent Message Retry Time period will follow and consume additional time. The Message Retry Time is that time duration between the beginning of the Start Bit of the first re-try message byte to the end of the Stop Bit of the final and completed message Acknowledge. The Message Retry Time is a composite time period that includes all additional Idle Times and all additional Message Times to complete the retry.

For the majority of distributed functions performed on the ACP Network, the typical ACP Communication Time is approximately 10 milliseconds.

The **Receive Latency** Time (RL) is that time duration between the beginning of the call to the Network

Receiver to the time of the Function Concluding Event. This time is directly proportional to the internal micro controller program organization and timing. Because this time is not a network timing characteristic, the requirement is established by the system designers.

ACP Network Traffic

Two types of distributed system events directly cause ACP network traffic.

Human Events Status Reports

Network Traffic Caused by Human Events

Human Events are detectable actions initiated by humans that require some form of interactive processing (usually a combination of hardware and software).

As an example, a driver who presses the button to turn on the distributed audio system is causing a Human Event. This event is recognized by the audio Front Control Unit, moved into its local micro controller, and interpreted. If the audio system was off, then a network message will be generated to activate both the amplifier and the tuner located in the Rear Control Unit.

For ACP, each human function operation may results in a network message. If pressing a button as fast as five times a second occurred and such events generated network activity, then a conversation speed of 5 conversations per second would be required. This human speed limit is one of the self-limiting properties of ACP network traffic.

Only a finite number of humans may interact with the available operating functions and such interaction seldom occurs simultaneously. This also is another of the self-limiting properties of network traffic.

Network Traffic Caused by Status Reports

Status Reports are controllable messages initiated by system events or periodic timed functions that require some form of distributed processing and thus result in network traffic.

As an example, an operating audio system playing a Compact Disc could experience a CD fault. Upon detection of this fault, the CD Player would generate a Spontaneous Error Report to indicate the condition.

Status Reports can be designed to be controllable. Such controls were employed to enhance the efficiency of ACP network traffic.

By placing limits on the total number and frequency of Status Reports, the major portion of ACP network traffic is primarily a function of Human Events. Status Reports have been moved to a lower priority to allow the higher priority Human Events immediate access to the network.

This scheme of efficiently manipulating the network access process coupled with the self-limiting nature of human generated messages has produced an overall traffic controlled ACP network.

Network Access Collisions

For the ACP-based distributed audio system, collisions seldom occur. Although this is partially due to the low volume of network traffic, the ACP access method is not random, it is synchronized.

All ACP nodes follow a specific set of synchronization rules before accessing the media. This method has profoundly lowered the number of collisions that would be usually expected had a random access CSMA/CD type network been employed.

An added measure of collision avoidance is included which uses a "check just before transmitting" technique.

ACP Network Performance

With its limited speed and UART-based simplicity, the performance of this Class A -type network has met all original design goals.

1. Low Cost 2. Low System Error Rate 3. 100% Distributed Control Functionality 4. Controlled Network Traffic

Summary

ACP is a complete, functional network solution which supports the requirements of Ford's Audio product development.

The design, implementation, documenting, and testing of the ACP network was influenced and made simpler by adherence to the OSI model.

The design of a conversation structure provides reliable communication and use of a structured message language provided an efficient method of organizing network operations.

The performance of the ACP network appears quite capable of supporting the vast majority of human response-type data found within the driver's area of the vehicle. ACP offers the opportunity to apply this Class-A type network solution into other applications.

ACP Development Acknowledgments

Akram Mufid - ACP Testing
Bob Gehrke - ACP Tool Development
Brad Ballard - ACP Tool Development
Cliff Merz - ACP Node Controller
Dave Sziraki - Network Software Development
Jeff Golden - Physical Layer Development
John Cotner - Founder of ACP

John Konstantine - ACP Physical Layer
Lynn Lenhart - Data & Message Sets & Documentation
Mark Burek - ACP System Support
Mike Burke - ACP Software & Documentation Support
Max Chapman - Early ACP Software
Nick Sarafopoulos - ACP Diagnostics
Rich Berry - State Machines & Object Oriented Design
Ron Brombach - ACP Vehicle System Group

References

1. "Serial Data Communications Between Microcomputer Systems in Heavy Duty Vehicle Applications - SAE J1708, June 1987

2. "Information Processing Systems - Open Systems Interconnection - Basic Reference Model", International Standard ISO 7498

3. James Martin, Local Area Networks, Prentice Hall, 1989

4. Grady Booch, Object Oriented Design, Benjamin/Cummings Publishing Company, 1991

5. Gary Dickson and Alan Lloyd, Open Systems Interconnection, Prentice Hall, 1992

6. Adrian Tang and Sophia Scoggins, Open Networking with OSI, Prentice Hall, 1992

7. Bijendra N. Jain and Ashok K. Agrawala, Open Systems Interconnection, McGraw Hill, 1993

8. Mark B. Abbott and Larry L. Peterson, "A Language-Based Approach to Protocol Implementation", Association of Computing Machinery, ACM 0-89791-526-7, 1992

Ford ACP

The following informations are taken from Andy Hammonds Yampp3/USB MP3 player firmware with ACP protocol support.
Download

Ford ACP is a network protocol used by the Head Unit to communicate with and control audio devices such as the Ford 6 disc CD Changer and the Nokia integrated cell phone or Ford Telematics units.

It is based on RS485 with 9 bit character data at 9600 baud.

A MAX-481 low power RS485 transceiver will work as interface between a serial USART and ACP bus.

```
+-+  +---------+  +-+
|  1  2  3  4  5  6 |
| [] [] []   [] [] |
|                   |
|  7  8  9 10 11 12 |
| [] [] [] [] [] [] |
+-------------------+
```

Pin Function
1 ACP +
2 ACP Shield
3 GND
4 n/c
5 Audio Left +
6 Audio Right +

7 ACP -
8 CDENABLE
9 +12V Power (unfused)
10 Audio Shield
11 Audio Left -
12 Audio Right +

You will need an AMP plug to connect to the head unit.
AMP Multilock Series 40 cable connector housing with 12 pins or sockets.

The CDENABLE line is 0V when the radio is off and +10V when it is on and can be used as a standby switch for the yampp.
It is not a power supply and can't drive a relay directly.

Communication

* a delay of 1642us (16 Bit times) will indicate a start of new message
* the 9th bit in a byte must be set in the last byte of message to indicate the end of message
* Acknowledge is given with 0x06

Byte 0 - Medium/Priority, should be 0x71

Byte 1 - Changer functional address, should be 0x9A or 0x9B

Byte 2 - Head unit address, 0x80 on receive, 0x82 on transmit

Byte 3 - Command control byte

- 0xE0 - Handshake 1, byte 4 should be 0x04
- 0xFC - Handshake 2, byte 4 must be the same for transmit and receive
- 0xC8 - Handshake 3, byte 4 must be the same for transmit and receive
- 0xFF - Current disc status in byte 4
    - Byte 4 - 0x00 Disk OK
    - Byte 4 - 0x01 No disc in current slot

        - Byte 4 - 0x02 No disc at all
        - Byte 4 - 0x03 Check current disk
        - Byte 4 - 0x04 Check all disc
- 0xC2 and 0xD0 - Change or request current disc
        - Byte 1 - 0x9A - command to change disc
        - Byte 1 - not 0x9A - request current disc
        - Byte 4 - disc number
- 0xC1 - Control command
        - Byte 4
            - Bit 0 - Fast search
            - Bit 1
            - Bit 3
            - Bit 4 - change Random status
            - Bit 5 - change Loudness status
            - Bit 6 - change Play/Stop status
            - Bit 7
        - Send back byte 4 with actual mode
- 0xC3 - Next track
        - Byte 4 - Track number
- 0x43 - Previous track
        - Byte 4 - Track number

The last byte in all message is a checksum of all previous bytes. Simply add all bytes of message to calculate the checksum.

Message examples
To display current play time, disc and track number:

Byte 0	1	2	3	4	5	6	7	8
0x71	0x9B	0x82	0xD0	Disc No	Track No	Minutes	Seconds	Checksum

No disc message:

Byte 0	1	2	3	4	5
0x71	0x9B	0x82	0xFF	0x01	Checksum

All informations are given without guarantee. Please mail for update or change requests.

Ford Audio Control Protocol (ACP) Specification

Simon Fisher source code package for ACP monitor/logger/interface