Introduction to Higher-Level Protocols

Version 1.0

Application Note AN-AND-1-160

Restrictions	Public Document
Abstract	This application notes gives an overview of the major features of the most widely used high-level protocols in the automotive industry.

Table of Contents

1 Overview

Modern automotive motion, convenience and safety applications are implemented over distributed electrical architectures that include a growing number of electronic control units (ECUs) communicating via one or more communication protocols over network buses. In addition to using the CAN protocol for automotive communication, there are several other major protocols in use today that are competing for the automotive market. This application note introduces these current protocols.

2 Current High-Level Automotive Protocols

The following table gives an overview of the major protocols in use today in the automotive industry. This list does not claim to be complete. Other protocols may be added to this document as needed.

Protocol	Home	Used For
CANopen	CAN in Automation (CiA)	Automotive, factory automation, off-road vehicles, maritime electronics, medical equipment, and railways
CAN Calibration Protocol (CCP)	Association for Standardization of Automation and Measuring Systems (ASAM)	Automotive, heavy truck and bus
DeviceNet	Allen Bradley, Open DeviceNet Vendor Association (ODVA)	industrial devices (such as limit switches, photoelectric sensors, valve manifolds, motor starters, process sensors, etc.)
ISO 9141 (K-Line)	International Organization for Standardization (ISO)	Road vehicles and diagnostic testers
J 1939	Society of Automotive Engineers (SAE)	Off-highway machines in applications such as construction, material handling, and forestry machines
Keyword Protocol 2000 (KWP 2000)	International Organization for Standardization (ISO)	European vehicles, diagnostic testers
NMEA 2000	National Marine Electronics Association	Marine vessels, maritime applications
XCP (General (X) CCP)	Association for Standardization of Automation and Measuring Systems (ASAM)	Automotive

Table 1 – High-Level Automotive Protocols

2.1 CANopen

CANopen was originally developed as a standardized embedded network with high flexible configuration capabilities. In 1995, the CANopen specification was turned over to CAN in Automation (CiA) international users and manufacturers group. Originally, the CANopen communication profile was based on the CAN Application Layer (CAL) protocol, an implementation of the OSI Application Layer. It works on top of the CAN Data Link Layer, which is normally implemented as hardware.

CANopen is a seven-layer protocol based on CAN that was originally developed for factory automation applications and industrial control systems. It is a device and communication standard for all seven OSI layers.

The CANopen profile family specifies standardized communication mechanisms and device functionalities. The family of specifications includes different device profiles as well as frameworks for specific applications. The profile family consists of a communication profile (CiA DS-301), the Framework for Programmable Devices (CiA DS-302) and a set of device profiles (CiA DS-4xx). Customer-specific profiles are also allowed. The Master Source Code provides all functions necessary for a CANopen Master and so allows the user to design applications with CANopen-conformant master functionality.

CANopen networks are also used in off-road vehicles, maritime electronics, medical equipment, and railways. It supports the interoperability of different devices, device synchronization, cyclic and event-driven data transfer and synchronous reading or setting of inputs, outputs and parameters. CANopen has modular capability, covering everything from simple to complex devices.

CANopen is now owned and maintained by CAN in Automation (CiA), the “International Users and Manufacturers Group”. It is open, vendor-independent, and can be implemented without a license fee.

For more information, see www.can-cia.org .

2.2 CCP

The CAN Calibration Protocol, or CCP, is a protocol for communicating between a target processor and a host machine via the Controller Area Network (CAN) protocol. A calibration tool (such as CANape) running on the host can communicate with the target, allowing remote signal monitoring and parameter tuning.

The interface defines methods to handle module calibration, measurement data acquisition, and flash programming activities. The module developer's needs decide whether the complete CCP interface or a portion of it is supported by the tool or implemented in the ECU. CCP places no limitations on the choice of physical layer or the system-selected communication bit rate.

CCP was developed during the early 1990’s and is used in applications in the automotive industry. It was adopted and enhanced by a European task force called ASAP (German for "Working Group For Standardizing Application Systems"), founded by German companies, and enhanced with optional functions. The ASAP standards effort has been renamed into a new organization called ASAM (Association for Standardization of Automation and Measuring Systems). The new organization has expanded the scope of the original ASAP interfaces beyond measurement and calibration to include diagnostics.

The CAN Calibration Protocol is primarily used as a monitor program which uses a standard protocol rather than a company-specific proprietary protocol.

CCP uses a specific conversation or dialog with only two CAN identifiers for message transfers. The tool (or Master) always initiates the conversation with a single CAN message. The “Master” controls data transfers. Once this message has been received, the ECU (or Slave) is then responsible for responding with a single CAN message.

CCP can support both a single point-to-point connection or a networked connection to an entire distributed system, making any combination of calibration, measurement data acquisition, and flash programming activities possible for a single module or any number of modules across a CAN network.

More information about the ASAM-MCD standard can be found at www.asam.de/new/standards .

2.3 DeviceNet

DeviceNet is a communications link, based on the serial bus standard for automotive networking known as Controller Area Networking (CAN), to connect industrial devices (such as limit switches, photoelectric sensors, valve manifolds, motor starters, process sensors, bar code readers, variable frequency drives, panel displays and operator interfaces) via a single network. It provides interchangeability of "like" components from multiple vendors.

DeviceNet allows the interchangeability of simple devices while making it possible to interconnect more complex devices. In addition to reading the state of discrete devices, DeviceNet provides the capability of reporting temperatures, reading the load current in a motor starter, changing the deceleration rate of drives, or counting the number of packages that have passed on a conveyor in the previous hour.

DeviceNet adopts the so-called object modeling approach, i.e. all information is structured in different objects. Services (such as Get and Set) can be applied to these objects to extract or change this information.

Four basic objects are required to handle information exchange:

Identity object. Identification information about a device, such as vendor ID, Device Profile, Revision etc., is stored in this object. Users can identify a particular object by remotely accessing this object.

Message Router. This object handles the explicit messages received by routing it to the proper destination objects.

DeviceNet Object. This object stores all DeviceNet related information, e.g. MAC ID and baud rate.

Connection Object. This object handles module connection, such as Explicit Messaging or I/O Messaging.

Each object has its own parameters called attributes (such as vendor ID) which govern the behavior of a device.

DeviceNet specifications have been developed by the Open DeviceNet Vendor Association (ODVA) and are internationally standardized.

More information about DeviceNet can be found at www.odva.org and at www.ab.com/catalogs/b113/comm .

2.4 ISO 9141 (K-Line)

The ISO 9141 standard specifies the requirements for setting up the interchange of digital information between an on board ECU and a suitable diagnostic tester to facilitate inspection, test, diagnosis and adjustment of vehicles, systems and ECU's.

The ECU must have one (K) or two (K and L) communication connections. Connecting lines K or L from one or more ECUs together results in a bus system. Line K is defined as the line that provides information in a serial digital form from the ECU to the diagnostic tester. Line K may also be bi-directional, in which case it may carry commands or data from the diagnostic tester to the ECU. Line K may also be used to initialize the serial communication. Line L is a unidirectional line from the diagnostic tester to the ECU. It may be used to initialize the serial communication and/or to carry commands and/or data. The standard K-line is ISO 9141 compatible for baud rates up to 250 kBaud.

ISO 9141-2 describes a subset of ISO 9141:1989. It specifies the requirements for setting-up the interchange of digital information between on-board emission-related electronic control units of road vehicles and the SAE OBD II scan tool as specified in SAE J1978. It is limited to vehicles with nominal 12 V supply voltage.

More information about K-Line can be found at www.iso.ch/iso/en .

2.5 J 1939

The Society of Automotive Engineers (SAE) Truck and Bus Control and Communications Subcommittee has developed a family of standards concerning the design and use of devices that transmit electronic signals and control information among vehicle components. SAE J1939 is a high-speed, Class C type communications network designed to support real-time closed loop control functions between electronic control devices that are physically distributed throughout the vehicle. It is used for off-highway machines in applications such as construction, material handling, and forestry machines.

J1939 is structured into several parts based on the ISO Open Systems Interconnect (OSI) Model. The OSI Model defines seven layers of communication, each performing different functions.

J1939 was developed to use CAN protocol physical layers, and much of the standard CAN data-link layer. Every message includes a 29-bit identifier which defines the message priority, who sent it, and what data is contained within ft. Collisions are avoided due to the arbitration process that occurs while the identifier is transmitted using non-destructive arbitration. This permits high priority messages to get through with low latency (delay) times because there is equal access on the network for any device.

While there is a J1939 document allocated to each layer, not all of them are explicitly defined within J1939.

Most messages on J1939 are broadcast to the entire network membership. This lets any device use the data without requiring additional request messages. When a message must be directed to a particular device, a specific destination address can be included in the message identifier.

To send a particular data item, a message must be constructed by first referencing the Data Content and Message table within appropriate J1939 documents. This will detail the Data Content value to use, the message update (transmission) rate, and default priority. Since related data items are typically packed together in a message to reduce overhead, it will also define the data field format. Messages that contain more than eight bytes of data can be sent as multi-packet messages.

The current data rate of J1939 is 250 Kbps. A typical message containing 8 data bytes is 128 bits long excluding bits used for bit stuffing) which is approximately 0.5 ms. The shortest message is 64 bits long.

More information about J1939 can be found at www.sae.org/servlets/index .

2.6 KWP 2000

KWP 2000 is the common name for ISO 14230, a protocol that is used mostly on vehicles manufactured and sold in Europe. It is also commonly referred to as "Keyword Protocol 2000".

The ISO 14230 protocol is very similar to the earlier ISO 9141 protocol. The physical implementation is identical, but it uses a different data format, as well as an optional "fast" initialization sequence. Much like ISO 9141, the vehicle must still be "initialized" to get diagnostic data from the vehicle's ECU(s). There are, however, two types of initialization. One type is essentially unchanged, and still uses the same 5-baud initialization sequence. The other type of initialization is known as "fast initialization". This is achieved by sending a 25 Millisecond pulse, then sending an initialization request at the normal 10.7 Kbaud rate.

KWP2000 operations primarily describe communication between an off-board tester and the on-board network. Before any communication between the tester unit and the vehicle network can take place, the communications link must be established through an initialization process. (This is a logical operation, not a physical one.) The KWP2000 specification calls out three different methods of initialization. Each is described in the ISO specification.

Another important and unique aspect of operations in KWP2000 mode is that it supports messages with a data field length of 255 bytes.

Key Word Protocol 2000 is covered under ISO specification 14230 and can be obtained from both SAE (www.sae.org) or ANSI (www.ansi.org).

2.7 NMEA 2000

The NMEA 2000 standard, developed by the National Marine Electronics Association Standards Committee Working Group in the late 1990’s, contains the requirements for the minimum implementation of a serial-data communications network to interconnect marine electronic equipment on board vessels. Equipment designed to this standard will have the ability to share data, including commands and status, with other compatible equipment over a single signaling channel.

Decisions that affect the operation of the ship and involve safety might include navigation, power generation, engines and machinery, fire alarm and control, etc. The International Electrotechnical Commission (IEC) standard IEC 61162-4 is addressing network requirements. Based on a Norwegian initiative called MiTS (Maritime Information Technology Standard), it is an Ethernet-based system, designed for operation up to 100 megabits/second.

Data messages are transmitted as a series of data frames, each with error checking and confirmed frame delivery. Data frames contain (in addition to control and error-checking bits) an 8-byte data field and a 29-bit identification field that sets message priority and identifies the data message, the source, and the destination. Typical data includes discrete parameters such as position latitude and longitude, GPS status values, steering commands to autopilots, finite parameter lists such as waypoints, and moderately sized blocks of data such as electronic chart database updates.

The standard defines all of the pertinent layers of the International Standards Organization Open Systems Interconnect (ISO/OSI) model, from the Application Layer to the Physical Layer, necessary to implement the required NMEA 2000 network functions.

The components of an NMEA 2000 network are:

- Physical Layer. Fully defined by the standard, including signaling voltages, cables, and connectors.
- Data Link Layer. Defined by ISO 11783-3 with additional requirements specified by the standard.
- Network Layer. To be defined in future versions of the standard.
- Network Management. Defined by ISO 11783-5 with additional requirements specified by the standard.
- Application Layer. Fully defined by the standard and includes a provision for manufacturer’s proprietary messages.

More information about the NMEA 2000 standard can be found at www.nmea.org/publications/standards .

2.8 XCP

The ASAM working group, a European organization whose members include several automobile manufacturers and suppliers, defines interfaces for the description and integration for micro-controller-based open-loop and closed-loop control systems. It developed *X*CP based on the established CAN Calibration Protocol (CCP). XCP offers all services needed to record ECU-internal run-time variables and to calibrate the control algorithms by modifying parameters.

XCP is a protocol used for measurement and calibration systems. XCP was created to fill a need for a media-independent universal measurement and calibration protocol. The “X” signifies a generic protocol independent of the medium used.

XCP works on CAN, Ethernet (UDP and TCP/IP) and on SCI. It is planned to be used in Flexray, TTCAN, USB and Firewire.

The XCP Packet Layer describes how the XCP Protocol is converted to a specific medium.It has a calibration data page initialization and switching, online memory calibration (read / write access), synchronous data acquisition, synchronous data stimulation and flash programming for the ECU development process.

Communication modes can be Standard, Block or Interspersed. With the Standard Communication Model, the Slave responds to each request packet by sending the corresponding response packet or an error packet. The Master waits for a response before sending the next packet. The Block Transfer Communication Model is used to increase the throughput of memory uploads, downloads and flash programming.

Communication with XCP is via two reserved XCP messages. Dialog is accomplished via data objects called packets, which contain from 8 to 255 bytes of data. Most *X*CP dialog is Master/Slave. The Tool acts as the Master, and initiates the dialog by sending a request packet to the ECU. The Master request packet is received by the ECU (the Slave), which will send a response packet. Minimal packet size needed for the transfer medium is 8 bytes (CAN).

There are two Basic Packet Types: a packet for transferring generic control commands (CTO), used for carrying out protocol commands (CMD), and a packet for transferring synchronous data (DTO).Applications for XCP include obtaining real-time ECU information via basic read and write functions, accessing ECU parameters in real time, real-time adjustment of ECU process algorithms (calibration), in-system or in-vehicle evaluation of design concepts, evaluation of engineering design modifications, in-system or in-vehicle flash programming and emulation beyond the lab bench.

Data throughput (upload, download, flashing and measurement data) with CAN 1MBit: 50K bytes/sec, with Ethernet 10MBit: 800K bytes/sec, USB 1.x: 1M bytes/sec.

For more information, contact ASAM at www.asam.net.

3 Contacts

Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
Germany
Tel.: +49 711-80670-0
Fax: +49 711-80670-111
Email: info@vector-informatik.de

Vector CANtech, Inc.
39500 Orchard Hill Pl., Ste 550
Novi, MI 48375
Tel: (248) 449-9290
Fax: (248) 449-9704
Email: info@vector-cantech.com

VecScan AB
Lindholmspiren 5
SE-417 56 Göteborg
Sweden
Tel: +46 (0) 31 764 76 00
Fax: +46 (0) 31 764 76 19
Email: info@vecscan.com