ATI ACCURATE TECHNOLOGIES

Technical Paper

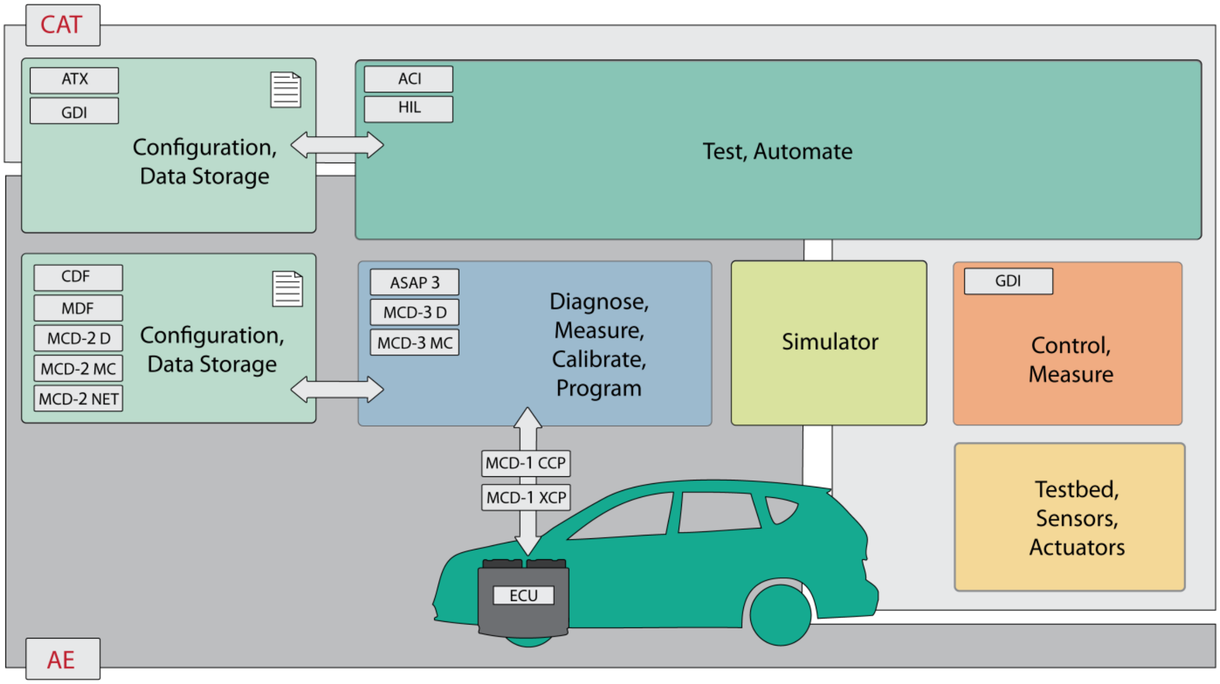
April 11, 2016

Printed for Control, Measurement & Calibration Congress

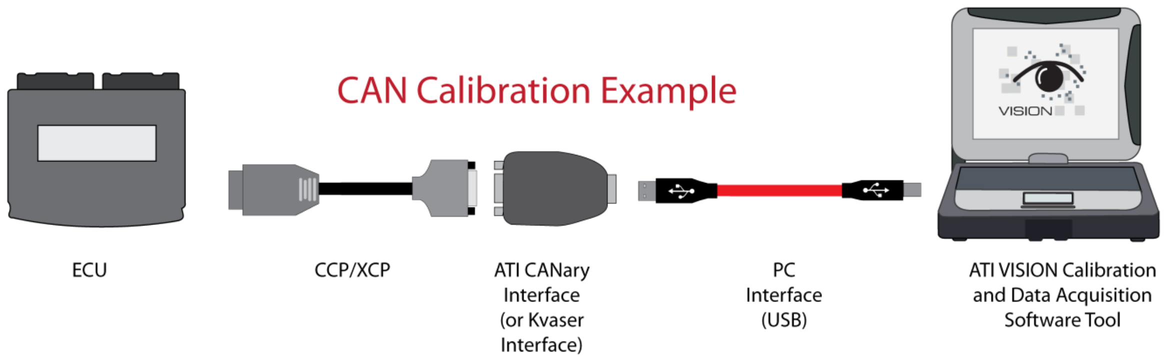
The Move from ASAM CCP to XCP Communication Protocol

Abstract

The Association for Standardization of Automation and Measuring Systems (ASAM) is an organization comprised of representatives from major OEMs and tool manufacturers in the automotive industry who work together developing technical standards that streamline electronic control development. Their goal is to enhance a development process chain that supports the interchange of data for tools used in simulation, measurement, calibration and test automation.



Specifically for calibration and measurement throughout all phases of development, ASAM MCD-1 defines several standards for the connection of an electronic control unit (ECU) to a measurement or data logging device. The focus of this paper is the background of the MCD-1 standards: CAN Calibration Protocol (CCP) and Universal Measurement and Calibration Protocol (XCP) (). The differences between the standards and why moving to the new XCP standard is the preferred approach for future controller communication protocol choices will be discussed.



From Proprietary to CCP

The first version of the ASAM MCD-1 CCP standard, CAN Calibration Protocol, was released in 1995. It initiated the move from proprietary, company-specific calibration schemes to the idea of a common standard protocol among calibration and data acquisition tools to enable the use of a variety of CAN devices from multiple vendors. Also known as "ASAP1," it is described on the ASAM website as “the definition of a communication protocol between master and slave controllers on a CAN 2.0B network, a network of CAN devices that uses 11-bit or 29-bit identifiers.” More specifically, the standard is concerned with moving acquired data continuously from and calibration data to the slave devices, as well as control of that data.

Although CCP is used throughout the industry and still the most prevalent, the standard was considered finished in 1999 and is no longer updated. In the early 2000s, the introduction of new communication networks added to ECUs and debugger interfaces incorporated into many microcontrollers offered faster calibration interfaces than CCP could provide. Therefore, the ASAM organization focused work on the next generation standard protocol, XCP.

CCP to XCP

The ASAM MCD-1 XCP standard titled "The Universal Measurement and Calibration Protocol Family" is an evolution of CCP. The XCP standard was established in 2003 and is based on years of experience garnered from CCP implementations. The purpose of the standard was to reduce the high demand of ECU resources and maximize data transmission over the communication network. The next generation XCP standard is described on the ASAM website as "the definition of a bus-independent communication protocol between master and slave controllers." The protocol definitions have expanded to include synchronous data acquisition and stimulation, read/write access on calibration data, memory page management, flash programming and further optional features. Transport layer specifications (the X of XCP) are defined for CAN, Ethernet (TCP/IP & UDP/IP), FlexRay, USB and SPI/SCI.

Many CCP implementations are migrating to XCP in Europe and Asia for calibration, data acquisition, and ECU flashing. XCP integration to ECUs is readily supported by the major tool chain suppliers, vehicle OEMS and ECU/microprocessor suppliers, and within hardware and software solutions that include standalone data logging devices. Capitalizing the on software tools support for XCP, it is also being used as the communication interface for external data acquisition hardware and other devices as direct communication link to those software tools.

What is extra in XCP?

To create a stronger more advanced standard protocol, the working committee collected input from the industry major players, including users. XCP provides depth of implementation along with extremely beneficial new features making the protocol faster and more effective.

- *Transport Layer Independent*
    - XCP is no longer limited to CAN. As other communication and debugger interfaces appeared on ECUs and microcontrollers, the need to leverage faster existing hardware was need. It was also critical to offer the variety of performance options to satisfy a wider range of tools. Virtually independent of the transport layer of communications, XCP supports CAN, Ethernet (TCP/IP & UDP/IP), FlexRay, USB and SPI/SCI. This

independence allows a software provider to write one XCP driver with the adoption of the driver to another transport layer requiring only small changes.

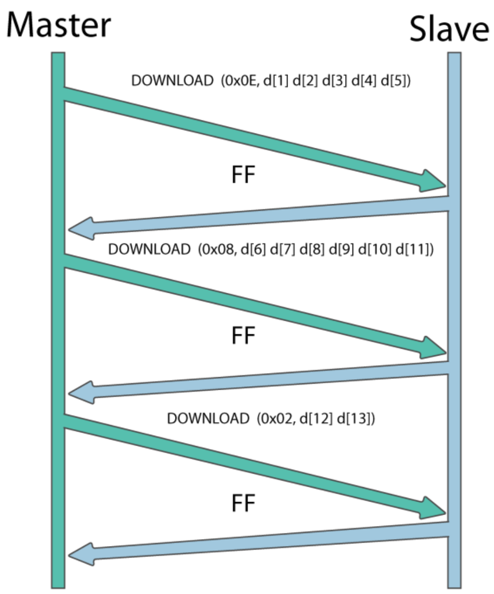
- *Higher Efficiency*

  The original CCP standard was relatively loosely defined. Parts of the standards that were not fully specified or ambiguous were interpreted differently by implementers causing the evolution of proprietary variants. For tool suppliers, making a single implementation is a challenge. The XCP protocol provides a more robust standard that leaves less to interpretation.

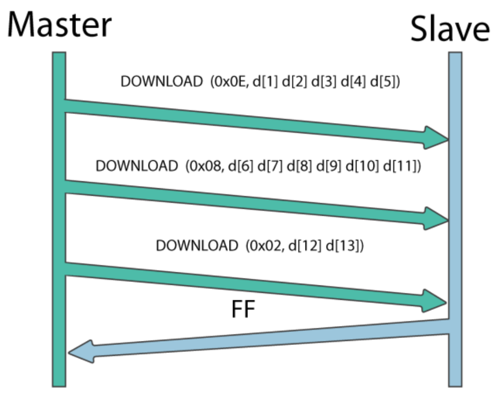
- *More Throughput*

  The XCP protocol builds upon the standard CCP command structure by adding additional optional commands to enhance previously time consuming operations for calibration (download commands) and flashing (programming commands). This makes the commands more efficient and thus increases the throughput of data between the tool and the ECU.

  For example, when transmitting consecutive data elements for the DOWNLOAD command, the added DOWNLOAD_NEXT is significantly more efficient.



DOWNLOAD command



DOWNLOAD_NEXT command

- *Support of Several Synchronous DAQ list modes*

  When it comes to measurement methods, both CCP and XCP can use asynchronous data collection, i.e. polling, to acquire values of measurement parameters. Likewise, XCP and CCP both support time and event driven synchronous data collection methods as well. Traditional, or static DAQ lists, have predefined sizes and specified rates that are supported by both CCP and XCP. The DAQ lists and ODT tables are permanently defined in the ECU code and are described in the A2L description file (ASAM MCD-2 MC) for the tool to interpret.

  Expanding CCP, XCP now supports predefined DAQ lists. This flavor of DAQ lists is entirely fixed as to what values are measured and at what rate the values are broadcasted. This method is practically never used in ECUs (due to its restrictiveness) but is popular with external measurement systems, like analog, thermocouple, and digital signal readers.

  Furthermore, XCP now offers dynamic DAQ lists. Unlike static DAQ lists, this method allows the tool to dynamically allocate the size and structure for desired DAQ lists, meaning that they can grow or shrink on demand to improve overall efficiency. When the ECU is queried, it informs the master of its DAQ memory capacity and lets the tool have more control over how this memory is used. This is another example of an intelligent XCP and tool interaction.

- *Introduction of STIM Data Transmission*

  XCP offers a Stimulation (STIM) mode which is the opposite of data acquisition. STIM mode is similar to DAQ mode, but the data flows synchronously in the opposite direction. The data packets from the measurement and calibration tool are buffered and saved back to ECU memory in a task synchronized way. The DAQ command requests a value of a variable whereas the STIM command sets a proposed value of a variable. For example, if a STIM command sets the oil temperature variable to 45^{o}C, then the ECU can elect to use that temperature rather than the temperature read from the sensor.

  Note that the XCP driver, the ECU code, and the measurement and calibration tool all must support STIM functionality for bypassing to function. The benefit can be described as a light form of rapid prototyping. "Hooks" to these variables are actually added to the ECU code for the purpose of altering them at a later date. This feature can be used to implement a bypass, where parts of the ECU's control algorithms are calculated outside the ECU by an external bypassing system.

- *Auto-detection of Slaves*
  Another intelligent feature of XCP is the auto-detection of slaves. The XCP protocol allows the Master to poll the Slave about its protocol-specific properties. A number of commands are available for this. The XCP master requests information from a newly detected slave and the slave provides information regarding its supported communication options, available resources, and any DAQ limitations.

Conclusion

XCP has gained momentum in automotive and in other applications that use embedded controllers such as rail systems, combustion chain saws and other applications are implementing XCP. Its success has also been marked by its adoption as the communication interface to non-ECU hardware, such as DAQ modules. In addition, XCP has become a more viable alternative to the new calibration interfaces, such as the debugger ports of microprocessors. Other standards have embraced XCP by integrating the protocol into the AUTOSAR basic software architecture.

With the industry acceptance of the XCP standard, customers are benefiting by gaining more freedom to purchase the equipment that works with their application that best fits their requirements and cost. In the spirit of expanding compatibility, Accurate Technologies' has supported ASAM on both CCP and XCP standards committees and offers embedded CCP and XCP drivers at no cost to its customers. ATI will continue to work on the XCP standard as it becomes more efficient and offers further features that will continue to make it attractive for streamlining controller development by expanding choices for customers, lowering costs and improving performance.

For additional questions, contact ATI support at support@accuratetechnologies.com.