

# Synchronizing Controller Area Network and Analog Input Measurements for In-Vehicle Data Logging

## Abstract

This application note discusses the technique and importance of synchronizing data acquisition signals from sensors and data from the controller area network (CAN) for in-vehicle applications. This technique has several challenges including the multitude of different signals and sensors available, synchronization techniques, and the post-processing and analysis of the synchronized data. This application note covers recommended best practices for synchronizing data acquisition signals and vehicle network communications.

## David Ashlock

*Applications Engineer, National Instruments*

Introduction to Synchronization for In-Vehicle Data Logging .....	2
Different Types of Input Signals for Synchronization .....	3
Controller Area Network Protocol .....	3
Analog Signal.....	4
Digital Signal.....	4
Global Positioning System .....	4
Multiple Synchronization Techniques for In-Vehicle Data Logging .....	4
Software Synchronization .....	4
Hardware Synchronization .....	5
Challenges of Synchronizing “Sample Rates” .....	5
Post-Processing and Analysis of Synchronized Data .....	6
Additional Resources .....	6
Notes .....	7

## Introduction to Synchronization for In-Vehicle Data Logging

In-vehicle data logging occurs in the later stages of automotive testing. It is often used to test and verify the operation of an automotive feature, such as an electric power-assist steering system. An automotive feature is a complex operation that may require input from multiple electronic control units (ECUs), which communicate through an in-vehicle network like CAN to operate properly. A typical top-of-the-range vehicle comprises over 30 electronic systems (for example, airbags, doors, and transmission) and more than 100 sensors (pressure, temperature, parking, and so on)<sup>1</sup>. For example, a company trying to ensure a smooth driving experience might need to validate active suspension and characterize how the automobile is reacting to the road. This test requires a data acquisition (DAQ) system to acquire and log data from automotive sensors that measure vibration as well as monitor more than one ECU that is a part of this system. There are a wide range of applications for in-vehicle data logging and each requires many considerations to validate a feature, electronic control unit, or an automotive sensor.

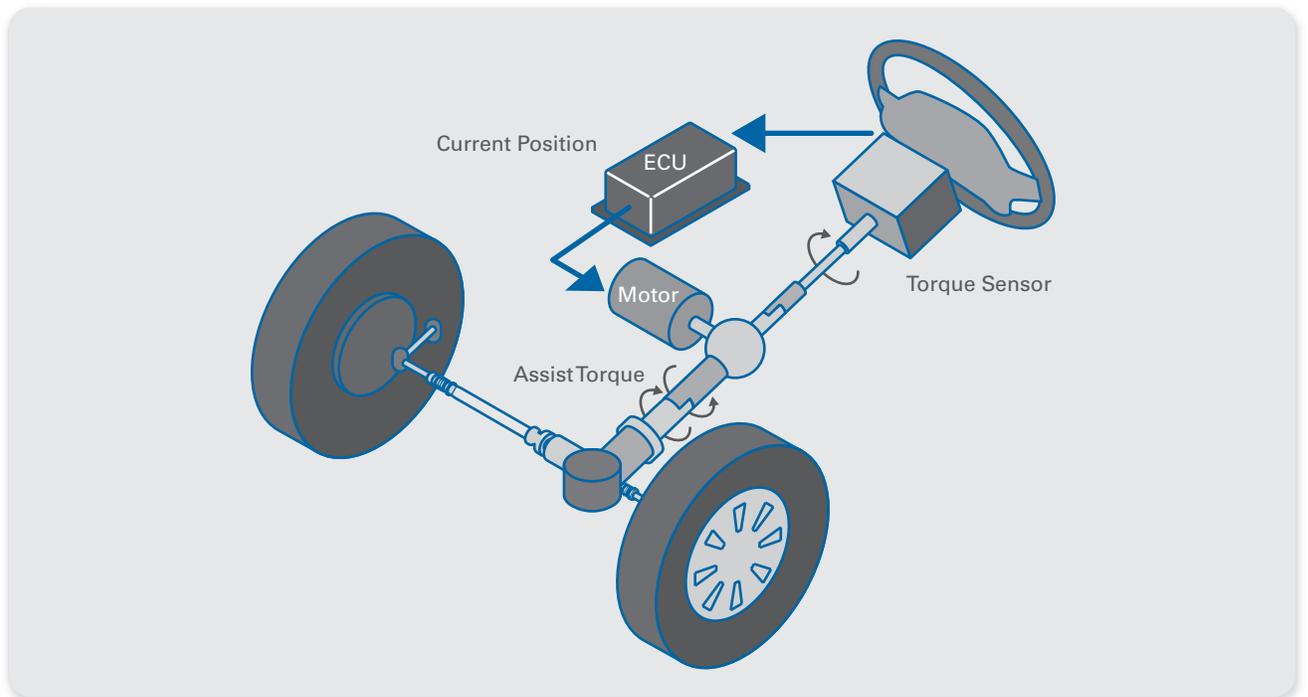


Figure 1. This is a diagram of an electric power-assist steering system, a feature in automobiles that acquires analog input measurements in order to determine the appropriate amount of motor voltage to be applied.

As the electronic complexity of vehicles has increased so too has the importance of automotive sensors and network communication. This has placed an added importance on the incorporation of in-vehicle networks like CAN in the testing phase. The ability to synchronize CAN transmissions with these other elements is an important characteristic of a data acquisition system to achieve repeatable measurement results. To validate these systems, you need to understand everything that happens at each instance, which makes synchronization necessary. For example, to validate hydraulic actuated active suspension, the data received from the vibration sensors needs to correspond to the appropriate CAN message to ensure the hydraulic servos are applying the correct counter forces. With synchronization of these signals and messages, an engineer can validate hydraulic actuated active suspension with confidence, because the data from a number of different sources is now correlated appropriately.

There are several considerations when synchronizing data acquisition signals and CAN transmissions. These can include the different types of input signals and sensors for automotive applications, the differences between hardware and software synchronization, and the post-processing and analysis of synchronized data. In this application note, review the best techniques to perform synchronization of data acquisition signals and CAN transmissions for in-vehicle data logging. The end goal is an accurate representation of a feature, an ECU, and an automotive sensor's performance.

This application note uses NI cDAQ-9174, NI 9201, and NI 9862 C Series hardware for the synchronization and data acquisition.

## Different Types of Input Signals for Synchronization

In a modern-day automobile, multiple types of signals are often involved. Depending on the in-vehicle data-logging application, different signals—such as CAN, voltage, current, pressure, PWM, GPS, and so on—may be required. To continue the previous example, for an active suspension system, there are analog sensors that measure the vibration of the automobile and CAN transmissions relay commands to the hydraulic servos to ensure a smooth ride. In addition, digital signals alert users when active suspension is engaged and GPS data could be used to correlate locations with the amount of hydraulic servo action. To discuss the challenges associated with logging this data, you first need to have a deeper understanding for the different signals that an engineer would encounter for these applications.

### Controller Area Network Protocol

Some electronic control units form independent subsystems, but communications between other systems are essential. A subsystem may need to control actuators or receive feedback from sensors and distribute this information to another system. The CAN protocol was created to be a reliable communication method to transmit and receive data from different ECUs. While CAN connects ECUs, smart sensors and actuators can now also connect and communicate directly on the bus like an ECU would. A CAN message contains an identifier, a native timestamp, and up to 8 bytes of data. A CAN message can either be transmitted at a regular interval or be event driven. This is an important difference between a CAN message and analog signals, which are sampled at a regular interval. Figure 2 demonstrates an electric power-assist steering system in which the CAN message is sent to control the motor based on the torque signal and vehicle speed received.

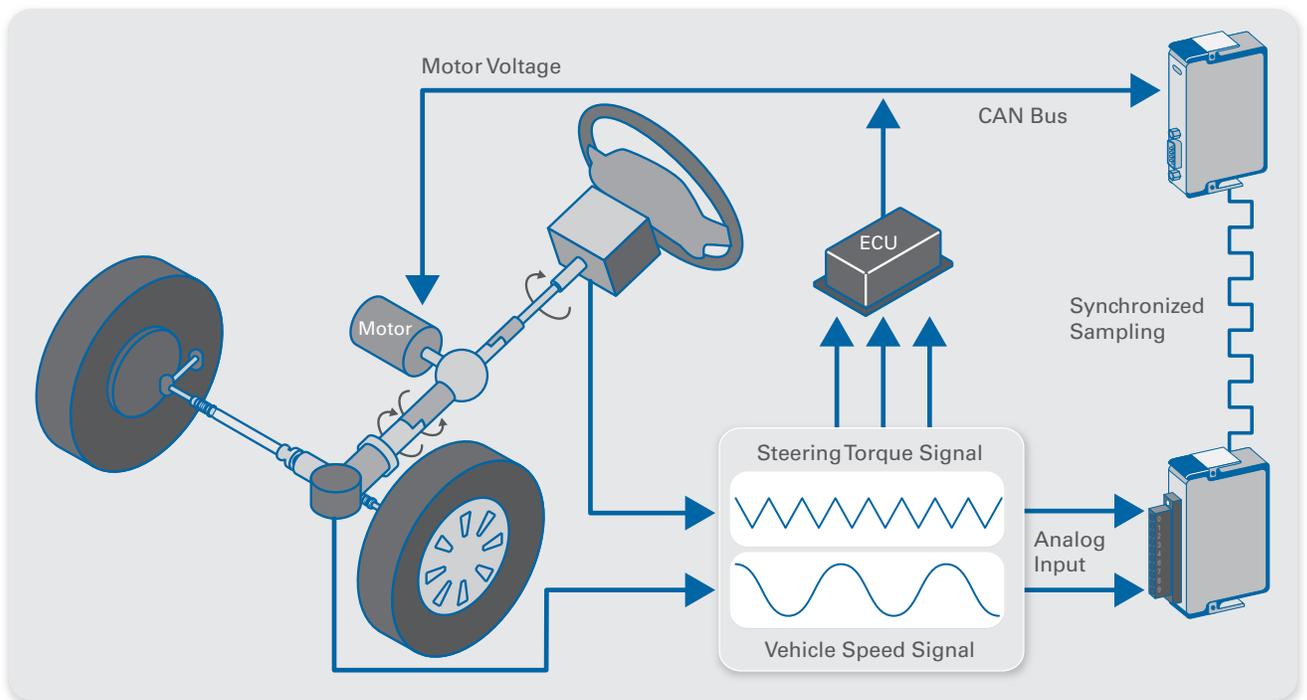


Figure 2. The system samples the steering torque and vehicle speed that are being delivered to the ECU. A CAN message is sent to alter the motor voltage based on the steering torque and vehicle speed received. The engineer can log CAN and DAQ messages simultaneously to determine if the desired motor output is being produced.

## Analog Signal

For in-vehicle data-logging applications, you can't validate a feature with only CAN communication. You need to gather data from different sources, such as analog sensors. Depending on what type of analog signal you are measuring, you can use a wide range of sensors (O<sub>2</sub>, ECT, MAF, vehicle speed, crankshaft position, and so on). Each sensor requires a different sampling rate. For example, it is not necessary to sample the vehicle speed sensor at 100 kHz because the speed for an automobile typically won't alter dramatically in a second. However, with a crankshaft position sensor, it may be necessary to sample at 100 kHz because of the possible instantaneous change and importance of this sensor data. You need to find the right automotive sensors based on the environmental and signal considerations and sampling rate. In Figure 2, the analog sensors measure torque and vehicle speed and when these sensors reach a specified threshold, a CAN message is sent to alter the motor voltage.

## Digital Signal

Digital signals can also be vital to certain systems. For in-vehicle data-logging applications, the typical digital signal will be measured or distributed as a high or low, PWM, or counter signal. In an automobile, the user wants to know when certain systems are operational, whether these systems are of high importance or not. For example, when you change the airflow, an LED is illuminated to show what setting the air conditioner is on. Another example of high importance is in an electric vehicle where a PWM signal is used to control motor output, which is critical to the operation of the vehicle.

## Global Positioning System

Another common type of data in vehicles is global positioning system (GPS). GPS is a radio frequency encoded signal that provides time, position, and velocity information by means of a network of triangulation satellites<sup>2</sup>. A vehicle uses GPS for many different reasons, such as tracking, network management, and monitoring user information. GPS information is vitally important for fleet recording to ensure that the location of testing is recorded when it comes to the post-processing and analysis stage of testing. For an ambulance network management, when someone has to be rushed to the hospital, the response time is vitally important. One interesting solution is to use a GPS-based fleet control system. The main targets are to know the exact location and the state (on duty or off duty) of all its ambulances at all times and to plan the ideal route to arrive at the suitable medical center from the help call place<sup>3</sup>.

## Multiple Synchronization Techniques for In-Vehicle Data Logging

You can synchronize CAN and sensor signals for in-vehicle data logging through two ways: software or hardware. To ensure the most accurate synchronization, you must take into account clock drift and start and stop latency by using hardware synchronization over software synchronization with the available Real-Time System Integration (RTSI) bus or onboard clocks. If the data is synchronized via hardware, the analog channel data will correspond with the appropriate CAN message because there is no concern of clock drift or latency.

### Software Synchronization

You can design an application to use software synchronization for sensor signals and CAN because both signals contain a timestamp. However, using software synchronization, there is no guarantee on the accuracy of the timestamps themselves because of clock drift and start and stop latency<sup>4</sup>.

#### Clock Drift

Clock oscillators are built with inherent frequency instability or jitter, therefore over time any two unsynchronized clocks drift relative to each other. For many validation applications, this amount of clock drift is unacceptable, especially in the cases of fleet recording due to the extended time of testing. If the two clocks on a DAQ device and CAN interface have drifted, you can't ensure that the samples obtained from each are correlated.

#### Start and Stop Latency

Using software synchronization, the control of functions starting and stopping is dependent on software latency. Since function calls must be serialized in a program, there will be some latency between the start of the DAQ and the start of the CAN acquisition. This latency results in an unpredictable amount of error between what DAQ considers sample 0 and what CAN considers transmission 0. The result is that the DAQ and the CAN information

are not correlated in time, and thus cannot be accurately displayed together without additional data manipulation. Software synchronization is also at the mercy of the operating system. If someone is using a general-purpose operating system, the latency could continually grow since the operating system is not deterministic.

### Hardware Synchronization

With hardware synchronization, the clock and trigger signals are physically connected; therefore, your application can achieve higher precision and is less vulnerable to clock signal problems. However, you want to ensure the hardware platform that you choose can use all measurements. Otherwise, you have to use software to try and synchronize your measurements, which is not ideal. For example, on a platform that performs hardware synchronization, such as the NI CompactDAQ platform, all measurements can be combined in a single task to easily synchronize all channels to the same clock<sup>5</sup>. All of the ports in the API (NI-XNET) for the [NI 9862 CAN communication module](#) share a common timebase with other data acquisition measurements. As shown in Figure 3, hardware synchronization uses the same clock to synchronize the CAN messages and sensor measurements. Software synchronization would use multiple clocks, which would not necessarily guarantee accurate synchronization.

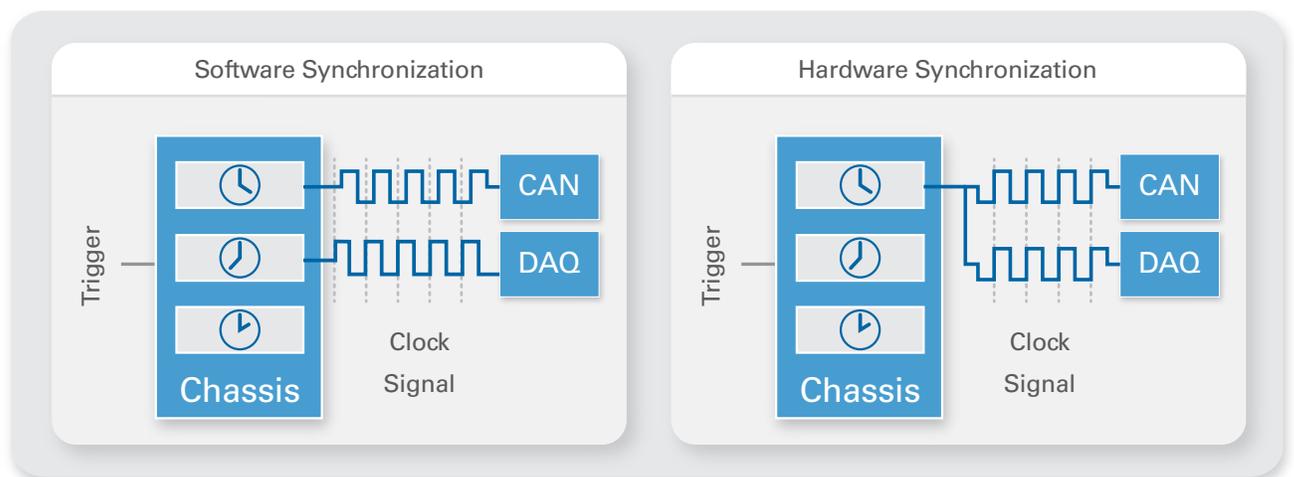


Figure 3. For software synchronization, the measurement platform would use two different timebases subjecting the sampling to timing errors such as clock drift and skew. For hardware synchronization, such as on the NI CompactDAQ platform, the chassis would use the same timebase eliminating timing errors.

### Challenges of Synchronizing “Sample Rates”

With the multitudes of sensors available for in-vehicle data-logging applications, choosing the correct sampling and streaming rate can be a challenge. Typically, based on the sampling rate needed, sensors can be classified into one of three categories: slow (OAT sensor, WTS sensor, and so on), medium (vehicle speed sensor, O2 sensor, mass flow sensor, and so on), or fast (crankshaft position sensor and so on). Choosing the right sample rate(s) for your application will depend on the types of automotive sensors being acquired.

Streaming rate is how much data the connection between your data acquisition sensors, CAN traffic, and data storage tool can handle. To calculate how much data bandwidth you need for your application, use the following equation.

$$\text{Data rate (Mbit/s)} = 2 \times \text{maximum frequency (Hz)} \times \text{bits per sample (bits)}^6$$

Note that data acquisition systems almost always acquire data in either 2 bytes per sample or 4 bytes per sample whereas CAN messages can vary in size but are at max 10 bytes per message.

There are a few options for communication buses. Each offers its own advantages and disadvantages, which you should weigh for your particular setup. The most common bus types are USB for direct connection to a PC or Ethernet for network and wireless connectivity.



## Notes

- [1] S. Prosser, "Automotive sensors: past, present and future," *Journal of Physics: Conference Series*, vol. 76, no. 1, pp. 1-6, 2007.
- [2] National Instruments, "Timing and Synchronization Systems," [Online]. Available: <http://www.ni.com/white-paper/9882/en>. [Accessed 4 April 2013].
- [3] LF. Zarazaga, P. Alvarez, J. Guillo, R. Lopez, J. Valino and P. Muro-Medrano, "Use cases of vehicle location systems based on distributed real-time GPS data," University of Zaragoza, [Online]. Available: [http://iaaa.cps.unizar.es/curriculum/09-Otras-Publicaciones-Congresos/cong\\_2000\\_TELEGEO\\_Use.pdf](http://iaaa.cps.unizar.es/curriculum/09-Otras-Publicaciones-Congresos/cong_2000_TELEGEO_Use.pdf). [Accessed 31 March 2013].
- [4] National Instruments, "Synchronizing CAN and Analog Signals for Device Validation," [Online]. Available: <http://www.ni.com/white-paper/4204/en>. [Accessed 1 March 2013].
- [5] National Instruments, "NI C Series Embedded Measurement and Logging Systems," [Online]. Available: <http://www.ni.com/white-paper/14095/en>. [Accessed 3 April 2013].
- [6] National Instruments, "Using Acoustic Beamforming for Pass-By Noise Source Detection," [Online]. Available: <http://www.ni.com/pdf/app-note/acoustic-beamforming.pdf>. [Accessed 5 April 2013].