

CAN data remotely accessed from anywhere in the world

Kvaser has announced an upgrade to the EtherCAN (CAN-over-Ethernet) product line with the Kvaser EtherCAN HS.

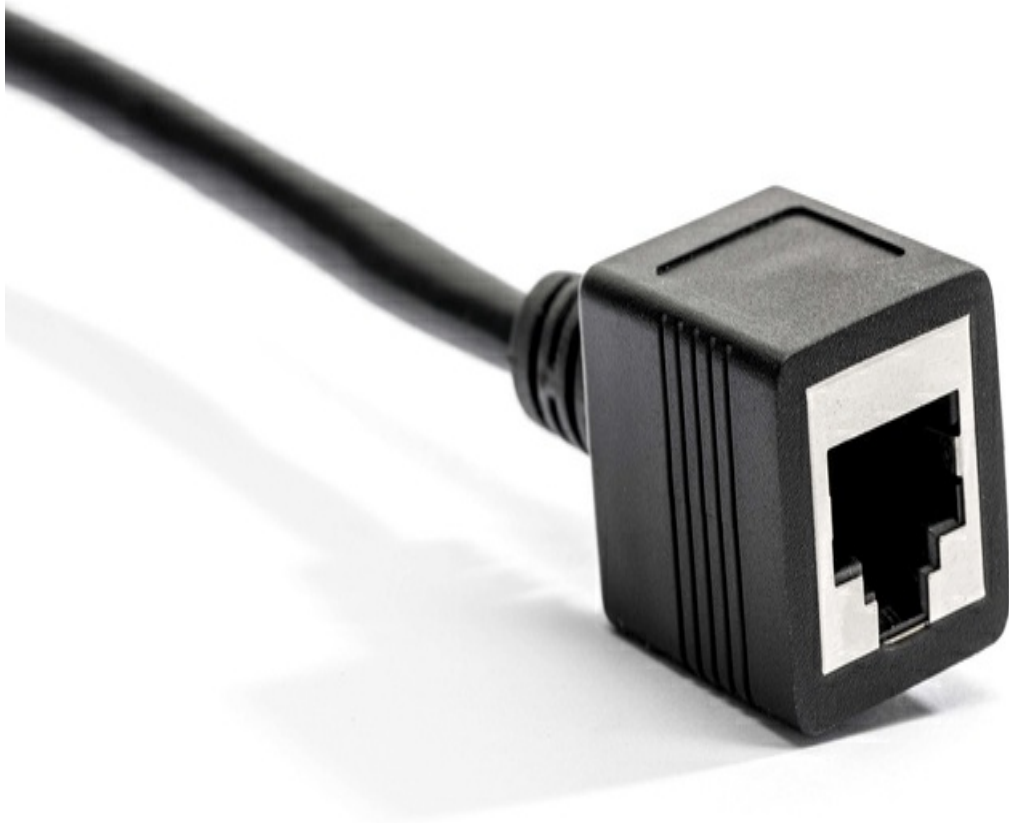


The EtherCAN HS (Photo: Kvaser)

The EtherCAN HS is a real-time Ethernet-to-CAN interface that, when linked over the Internet to an Ethernet-equipped PC, allows CAN data to be remotely accessed from anywhere in the world. Built-in Power-over-Ethernet eliminates the need for a separate power cable when users can't power the device from the CAN network.

The EtherCAN product range enables the user to implement the Internet-of-Things (IoT) concept by enabling data from any CAN product or system to be sent over a corporate network or the Cloud, using the Kvaser CANlib API. Users of the EtherCAN HS can also choose to connect to the device using the built-in Rest API for web-enabled devices, such as smartphones.

As a programmable interface, the EtherCAN HS is suited to advanced applications, such as ECU reflashing. For example, a "t" program could be created to process CAN messages locally in the device, removing the latency of WiFi and/or Ethernet.

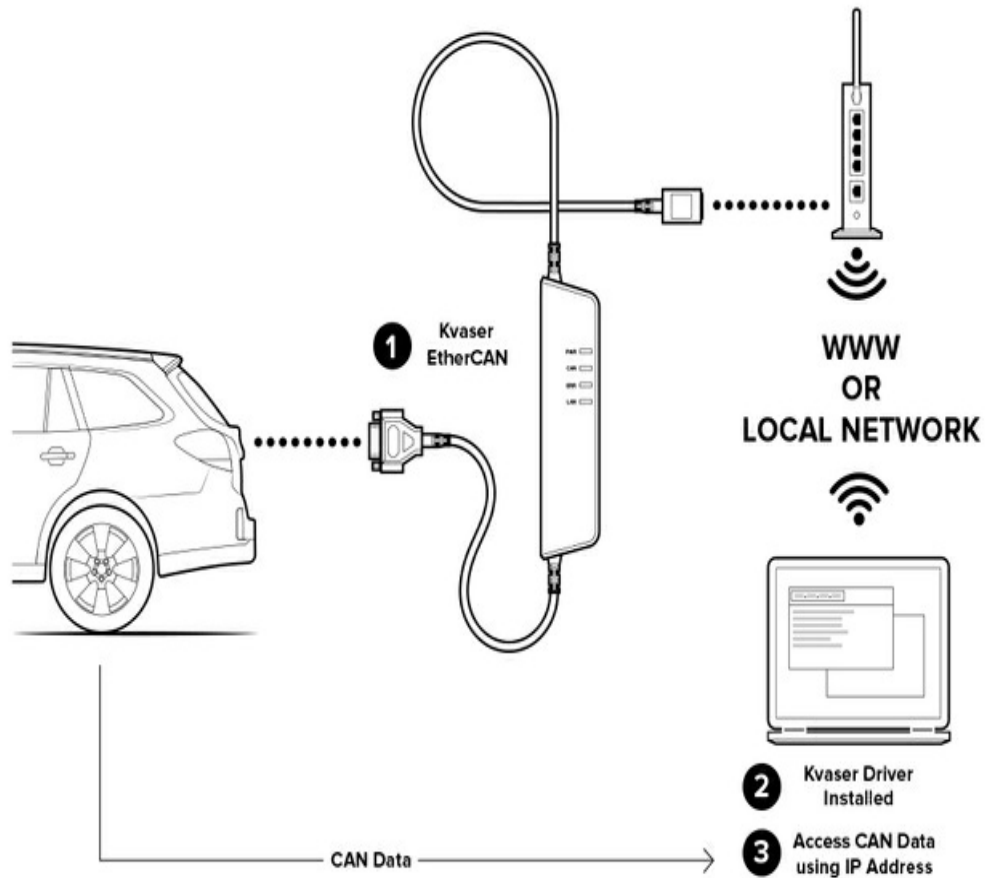


The EtherCAN HS in detail (Photo: Kvaser)

Why use a programmable interface?

WiFi or Ethernet latency is a topic for example in advanced application such as ECU reflashing, where fast round-trip responses are needed. Latency can manifest at the PC, when sending a CAN message, and when the PC is reading a return response. Whilst latency can be handled in software when it is a known value; unpredictable sources of latency are a problem. WiFi or Ethernet latency is unpredictable because random delays are created.

Some CAN interfaces are better suited than others at handling this problem. If latency is an issue for an application, a programmable interface is the answer. For example, the EtherCAN HS users can implement the CAN protocol in a t program (to create in Kvaser's CANlib SDK, which is free to download) and run it as a script locally in the device, thus avoiding any impact from WiFi or Ethernet delays as the message timestamp comes from the interface itself.



EtherCAN illustration (Photo: Kvaser)

This approach could also be advantageous to users of the company's Rest API. A program can be written to process the CAN messages locally on the interface, so that just the required information is sent to the host PC. The alternative of transmitting all messages over Rest is not that efficient because of the overhead of http and the json coding.

When using a CAN interface to transmit a Diagnostic Protocol command, the program typically expects a message response within a certain period before it 'timesouts'. Without programming the CAN message flow, the program could timeout most of the time. Programming a Kvaser interface to avoid this situation isn't as complex as it sounds, due to Kvaser's software resources and example files, and it could be much less time-consuming than repeated timeouts. And if users don't want to create a program for their interface, they can get in touch with the company to find a technical partner that can create one for them. The product is available for sale immediately.

[CW](#)