CAN Newsletter Online

CAN NEWSLETTER MAGAZINE

CAN protocol enhancement

This article describes the enhanced CAN protocol called CAN-HG and the features of the IC circuitry from Canis that implement it.

This article originally appeared in the September issue of the CAN Newsletter magazine 2018. This is just an excerpt.

Carrier frame signals at the TX pin (Photo: Canis)

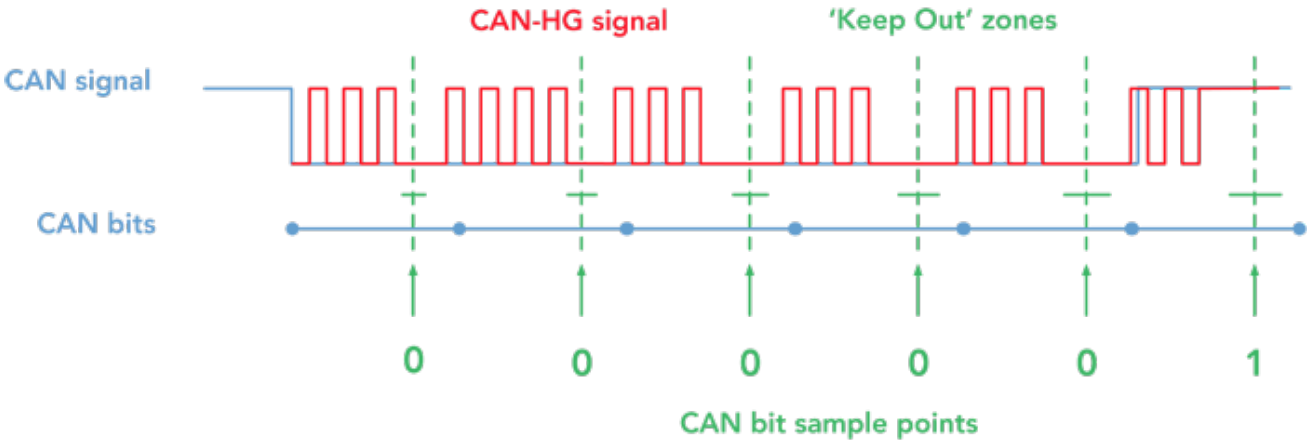
CAN-HG has been designed to meet two important requirements: Increasing the bandwidth and guarding the network. There is a third requirement – and this requirement is the most important: the protocol enhancement must be completely compatible and interoperable with Classical CAN. It must run on existing wiring with existing nodes using legacy CAN controllers in legacy micro-controllers. Any approach that requires every node on a bus to be re-developed for new micro-controllers is infeasible: it must be possible to freely mix Classical CAN and CAN-HG on the same network segment.

Increasing the throughput

The bit time in CAN is setaccording to the electrical characteristics of the physical CAN network and the dominant factor is the propagation time across the bus: CAN arbitration requires that there is sufficient time for all nodes to signal a bit and for the signal to have reached all other nodes before the line is sampled. After arbitration has been decided this propagation time constraint no longer applies and shorter bit times could be adopted. But a new frame format, in which there is a switch to faster signaling causes compatibility problems: legacy CAN controllers cannot follow the new frame format and would typically generate error frames and drive the transmitter into bus-off state. CANHG solves this problem with the notion of a carrier frame. A carrier frame is a CAN frame with a payload of 8 byte fixed to 30 00 00 00 00 00 00 00_{16}. After bit stuffing this results in the following bit pattern:

1000**0**011100000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**0000**1**.

The underlined bits are the DLC field and those in bold are stuff bits. The others are the payload bits in the 8-byte CAN data field. The digital signal to the CAN transceiver (i.e. the TX pin from a CAN controller) is shown in Figure 1. In a carrier frame are fourteen intervals of five dominant bits (i.e. 00000) followed by a recessive bit. The proposed CAN-HG protocol adds short-duration bits – called Fast Bits – within these intervals.



A 6-bit CAN interval beginning with a falling edge, recessive-to-dominant (Photo:Canis)

The Fast Bits are placed so that all CAN controllers receiving the frame (including the transmitter) see only the original signal. This is illustrated with a simplified example in Figure 2. It shows a 6-bit interval beginning with a falling edge from a recessive bit (i.e. 1) and ending at the end of the final recessive bit, with the vertical arrows showing the sample points for each bit. The fast bits overwrite the original CAN signal but return to the original signal value around the sample point.

In ordinary circumstances it would not be possible to put arbitrary edges within a CAN bit: a falling edge initiates the re-synchronization process, which adjusts the sample point and this would then cause later bits to be misread, leading to a CAN error frame being raised and a failed frame transmission. But with CAN-HG the format of the carrier frame and the placement of fast bits are designed to exploit a feature of the re-synchronization: the first falling edge at the beginning of the interval is a re-synchronization point. All connected CAN controllers will perform sample point adjustments to offset from when this edge is detected. But further falling edges within the same bit (up to the sample point) will not result in another re-synchronization: this is prohibited by the Classical CAN protocol.

If you want to continue reading this article, you can download the PDF of Dr. Ken Tindell. Or you download the full magazine.

cw