CAN vs. RS485

White Paper

Overview

CAN (Controller Area Network) and RS485 are popular standards in fieldbus systems. As RS485 only addresses layer 1 (the physical layer) and CAN also adds layer 2 (the data link layer) in the OSI model, it is difficult to compare the two standards. Nevertheless, this paper will compare the two standards in an attempt to assist users in deciding whether or not to employ CAN in their systems.

Basics

RS485
The RS485 standard was created in 1983 providing a multi-drop communication capability for devices with at least one UART. Any device connected to the RS485 bus can communicate with any other device in half-duplex mode. Although half duplex enables a bidirectional communication, only one device can transmit at a time and the other devices have to listen.

At this, RS485 is not a protocol, it only provides the basic rules and the physical link for data exchange, enabling the transmission of serial messages using a multi-drop bus whereas the message content is completely user defined.

This also means, that e.g. the communication frame structure, the scheme for addressing other nodes, the mechanisms to avoid data collisions and other tasks have to be implemented by the developer in form of a protocol software.

CAN
The Controller Area Network (CAN) was created in the 1980s by Robert Bosch GmbH. In the beginning, CAN mainly targeted the automotive industry, but nowadays it is used in a variety of applications, like industrial automation, medical, transportation and so on.
Compared with RS485, CAN not only provides the physical media for the communication, it also provides all other mechanisms necessary for addressing data packets (messages), avoiding data collisions, detecting failures in the transmitted data, automatic repetition of disturbed messages and ensuring data consistency over all nodes in a network. Furthermore, CAN specifies the structure of the data frame, with message identifier, data and control bytes.

Like RS485, a CAN node can communicate with any other node in half-duplex mode. A message consists primarily of an identifier (ID), which represents the priority of the message with respect to collision avoidance and up to eight data bytes. It is transmitted serially over the bus. This signal pattern is encoded in non-return-to-zero (NRZ) and is sensed by all nodes.

If the bus is free, any node may begin to transmit. If two or more nodes begin sending messages at the same time, the message with the higher priority, as defined by the more dominant ID (which has more dominant bits, i.e., zeroes), will overwrite other nodes' less dominant IDs. This insures that only the dominant message is transmitted and received by all other nodes. This mechanism is referred to as priority-based bus arbitration. Messages with numerically smaller values of IDs have higher priority and are transmitted first in the event that two messages are sent simultaneously.

Comparison

RS485 is normally operated with the aid of a UART module. Communication is, therefore, carried out on the basis of individual characters which are written or read via an 8-bit register. Therefore, the structure of a telegram must be implemented in software. In contrast, CAN functions with a permanently pre-set telegram format which, in addition to the 0 to 8 databytes, also contains the control information for addressing and data consistency (CRC check). The user only has to transmit the message identifier and the payload, the rest is added by a so called CAN controller. The CAN controller may be an external chip or an internal module within a microcontroller.

Each CAN message has its own priority which is derived from the message identifier. CAN messages with higher priority can be treated preferentially by the CAN controller chip and are, therefore, transmitted before messages with a lower priority. Also, among the nodes on the CAN bus, the message with the highest priority is transmitted first.

Consequently, CAN then becomes a real-time-compatible solution. Depending on the priority of a message by virtue of its ID, the maximum latency time can be predicted for each CAN message, independently of the busload or performance of the other nodes. In contrast, RS485 cannot trigger any message collisions, therefore, collision avoidance must be guaranteed with the application software. Typically, collision avoidance in RS485 networks is achieved in master/slave relationships by the master polling all slaves one after the other. The latency time results from the number and reaction time of all nodes and is therefore much longer.

Due to message-wise arbitration, CAN enables multi-master operation without making additional precautions. In the case of RS485, this is only possible with a specific protocol, such as token ring, which has to be implemented in software.

CAN has very advanced error management. If a message is not correctly received by a node (CRC or format error), the telegram is destroyed by the recipient via an error frame and is, therefore, marked as invalid for all nodes. This action initiates an automatic response in the CAN controller to repeat the transmit process. This procedure ensures that all nodes only receive valid messages. Nodes that repeatedly transmit incorrect data or at first recognise received data as defective are automatically disconnected from participation on the CAN bus. This guarantees that defective CAN controllers or bus connections cannot permanently disrupt data traffic.

These measures taken collectively (brief messages, differential transmission, error detection and troubleshooting, withdrawal of defective nodes) make CAN a very robust, secure, reliable network which is why CAN is employed in many critical or safety-related applications in vehicles, ships, elevators, medical devices, aircraft and industrial plants.

Almost every microcontroller family today offer versions with an integrated CAN controller that is virtually free. RS485, which typically works with an internal UART, is also free. This means that there are really no cost advantages or disadvantages when employing CAN or RS485. The costs for the transceiver modules for CAN are even lower than for RS485 as illustrated in the summary comparison table below.

Due to the full implementation of the CAN protocol in hardware, the load on the microcontroller system is reduced (only one interrupt per message). With RS485 there is typically one interrupt per received character, in addition to processing of the protocol in software (including data consistency mechanisms), which can lead to a considerable CPU load for RS485. Depending on the protocol and data transmission rate, as well as requirements of the application RS485 requires more powerful and, therefore, more expensive CPUs. Even already lower transmission rates can have a considerable impact on the CPU performance and consequently on the cost.

CAN controllers are capable of filtering out receive messages based on the message identifier. The CPU load can thus be further reduced, as only those messages that are relevant for the node are received.

Further arguments in favor of CAN
By using CAN, the UART interface of the microcontroller remains free and can be used for debugging or other purposes (printf).

Due to the high degree of standardization with CAN, PC/CAN interfaces (USB, PCIe, ...) including software drivers are available at low cost and thus enable easy access to CAN for PC applications.

Many CAN analysis, simulation, configuration and other tools are available for PCs, which provide access from layer 1 to layer 7 and thus simplify development and troubleshooting.

Various layer 7 protocols such as CANopen, DeviceNet and J1939 are available from many vendors for CAN. This enables the designer to use proven layer 7 software for almost all available microcontrollers leading to reduced risk, faster time to market and lower development costs.

I/O nodes, sensors and drives are available on the market for CANopen and DeviceNet layer 7 protocols. In addition, these protocols are available with pre-defined device standards that guarantee, by virtue of compliance with the standard, that all nodes operating on the CAN bus will communicate together. An important consideration in upgrading from RS485 to the more robust CAN standard is the ability to use existing RS485 wiring and cable harnesses provided that they are not too long and do not exceed the limits of the CAN standard. Even in such cases, low cost CAN repeaters or bridges can be used to extend the length of the CAN network.

Summary

Feature	RS485	CAN
Required interface	UART	CAN controller
Supported ISO model layers	Physical layer	Physical layer and data link layer
Detection of data collisions	Not implemented	Yes, CSMA/CR
Maximum transmission rate	10 Mbit/s (up to 12 m)	1 Mbit/s (up to 50 m)
Maximum bus length	1200 m (at 100 kbit/s)	1600 m (at 50 kbit/s)
Supported Bus arbitration principles	Master/Slave or Token Ring	Multimaster and all principles, which can be derived from that like Master/Slave or Token Ring
Maximum data amount per frame	Unlimited	8 bytes
Bus transceiver costs	~ 70 ¢	~ 35 ¢
Examples of popular protocols	Modbus RS485, Profibus	CANopen, DeviceNet, J1939

For new developments, in which real-time behavior, stability of communication, failure safety and availability of proven layer 7 communication protocol stacks and tools play a role, CAN stands alone. In addition, implementation with CAN now costs less than implementation using RS485.

For pure master/slave systems, for which the above criteria do not play a role but which address a very high number of nodes, and/or where large data quantities have to be transmitted, which may be limited by CAN communication, an RS485 implementation can offer advantages.

List of Sources

[1] http://en.wikipedia.org/wiki/OSI_model
[2] http://en.wikipedia.org/wiki/Controller_area_network
[3] http://en.wikipedia.org/wiki/RS-485