White Paper

Powering Smart Automation with CANopen

CANopen enables sophisticated smart automation solutions in the Industrial Internet of Things (IIoT).

Introduction

Systems used in automation include a vast number of devices, each with multiple sensors, motors, as well as other components and microcontrollers that receive and transmit information. To ensure the necessary coordination between all these components and devices, and the effective monitoring of them, a sophisticated communication system is critical.

CAN bus is a message-based protocol designed to facilitate such communication and has grown to become a preferred standard across a wide range of industries, e.g., aviation, the auto industry, and industrial automation.[1] Across these industries and applications, numerous higher-layer protocols exist, each with its specializations and industry focuses. In automation, the higher-layer protocol CANopen is one of the preferred protocols, and it has become increasingly popular to use with the growth of smart automation and the Internet of Things (IoT).[2]

However, expanding existing systems or hardware to support CANopen can be challenging, and even more so in a way that meets the needs for ruggedness in smart automation applications.

Background

CAN Bus

Controller Area Network (CAN bus) is an internal communications network initially developed by Bosch as a standard for communication between microcontrollers and devices inside automobiles. In 1993, the International Organization for Standardization adopted CAN bus as an international standard, and CAN bus has since become an increasingly popular standard in the automotive industry and beyond.[3]

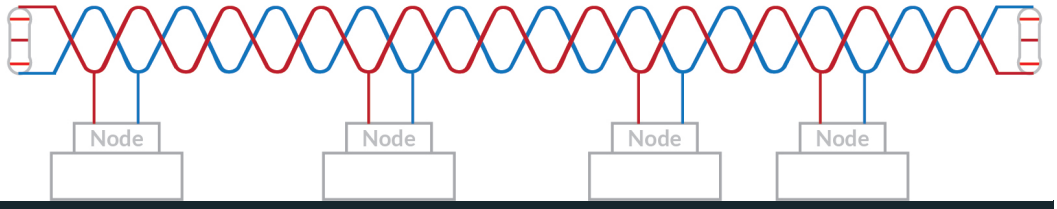

Image 1: CAN devices connected by a single twisted wire in a CAN bus system.

One of the main benefits of CAN bus is that it allows communication between devices and microcontrollers without the need for a host computer. With CAN bus, devices are connected through a single twisted wire, and signals transmitted by any device in the network are received by all other devices connected by the CAN bus.[4] Instead, identifiers within each message are used to determine the intended recipient(s) of messages transmitted through the CAN bus.

The result is an efficient system that requires little wiring, no central host computer, and that can handle a vast number of connected devices.

CAN Higher-level Protocols

CAN higher-level protocols can roughly be described as languages, each using CAN bus as a common alphabet. However, unlike with human languages, what determines which higher-level protocol to use is the intended application—with specific protocols more popular or more useful for particular applications.

There is a wide range of CAN-based higher-level protocols, many of which are specialized for specific industries or even for use by a single manufacturer. Some of the most common standardized higher-level protocols are CANopen, DeviceNet, and SAE J1939. Among the more specialized protocols are, for example, GMLAN (for General Motors), RV-C (for recreational vehicles), and CubeSat Space Protocol (for CubeSats).[5]

CAN in Automation

While CAN bus started as a standard for in-vehicle communications systems, it has since grown to become a popular standard for use in automation. In such applications, the CANopen higher-level protocol has grown to

particular prominence. CANopen is developed and supported by CAN in Automation (CiA)—an international non-profit organization for CAN users and manufacturers—through its CANopen Interest Group.[6]

Originally designed for use in motion-oriented machine control systems, the CANopen standard is now widely used within automation. For instance, it is a popular protocol to use within robotics, alongside factory conveyer belts, and throughout industrial machinery. However, CANopen is not limited to automation; it is also used within sectors such as healthcare and the automotive industry.[7]

With the growing trend of smart factories with a high degree of automation, CAN bus and the CANopen protocol are likely to only play a more significant role in the future.

CANopen Devices

• Object Dictionary

CANopen devices, usually referred to as nodes, are all required to have an object dictionary—a standardized table that stores data pertinent to the node and its operation. For instance, each CANopen node's object dictionary contains an entry for its device type used for identification purposes. Other object dictionary indices may contain information such as sensor readings or process states, e.g., whether or not a signal is currently being transmitted or if a device operation is currently active.[8]

• CANopen Communication

CANopen nodes can communicate with each other according to three different architectures or communication models, namely master/slave, client/server, and producer/consumer, which in turn determine the relationship between the nodes.

In a master/slave communication model, one of the CANopen nodes sends and requests data from the "slave" nodes, which essentially just follow the instructions from their "master." In a client/server model, meanwhile, "clients" read or write data from "servers." Lastly, in a consumer/producer communication model, "producers" broadcast data to all other nodes, i.e., "consumers" of the producer's data.

Additionally, CANopen nodes communicate through different communication services, with each communication service suitable for communicating particular commands. For example, the Network Management communication service is used to communicate the desired state of CANopen nodes (e.g., turning all motors off or starting a sensor).

Another critical communication service is Heartbeat, which nodes use to regularly transmit a "heartbeat" to other nodes, thereby indicating that they are still active.[9]

- **CANopen Message Format**

Messages transmitted by CANopen nodes follow a standardized message format that makes it clear to receiving nodes which node is transmitting the message, how long the message is, as well as the actual data transmission (message). This standardization applies to all CANopen messages, no matter what communication models or communication services are used.

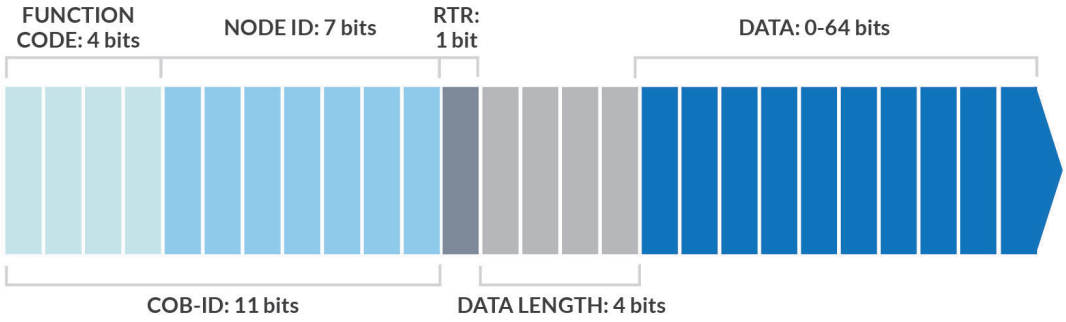

Image 2: CANopen messages follow a standardized format that identifies the source of the transmission and relay up to 64 bits of data per transmission.

The first part of a CANopen message contains an 11-bit long CAN-ID, which identifies the transmitting device, followed by a control bit. This part of the message is generally referred to as the communication object identifier or COB-ID. The next four bits tell receiving devices how long the message is (i.e., the data length) so that they can determine at which point in a transmission the message ends. The last part of the CANopen message is the actual data, which can be up to 8 bytes (or 64 bits) long. The actual length of the message depends on what type of data is transmitted.[10]

Challenges

While CANopen can seem like a very compelling system to implement in a range of applications, and perhaps most notably in smart automation, this can prove more challenging than system integrators expect. For instance, automation environments often present devices with steep requirements in terms of size and durability. Combine these requirements with the harsh environments often found in industrial environments, and these applications risk to start sounding very taxing to any new system to implement.

The difficulty and costs associated with implementing a CANopen system in automation settings—or any medium to large-scale application—also pose significant hurdles to overcome.

Application Challenges in Automation

Applications in automation can present a broad and diverse range of challenges for CANopen solutions. For instance, industrial robots are exceptionally finely tuned and space-efficient, leaving little room for interface expansion. Moreover, the speed and precision expected from such equipment require devices to meet the highest quality standards in terms of components and design.

Production lines in smart factories and Industrial Internet of Things (IIoT) applications also involve a staggering number of machines, sensors, and microcontrollers, that all need to be fully coordinated. If there is a lapse in performance or reliability localized in a single component somewhere along the production line, all production risks grinding to a halt. Thus, communication systems implemented in such applications are required to allow extreme coordination—and, therefore, performance—as well as efficient modularity in the event of a component or device failure.

Difficulty of Interface Expansion

The complexity of the systems used in smart factories is one of the key challenges for any new communications system to address. With a vast number of different devices, each with their customized and intricate features and hardware layouts, the interface expansion necessary to implement a system like CANopen can be daunting—if not simply impossible.

If each device in such systems is required to be redesigned to support CANopen, the expense, time, and difficulty associated with such an undertaking make it close to an impossible task, or at the very least one that is difficult to defend from a financial point of view.

Another compounding factor is the risk that customized designs to allow CANopen support present a considerable challenge to system operators in the event of a hardware failure. Without sufficient modularity, any such hardware failure risks requiring that replacement parts undergo an extensive redesign process, and thereby risking lengthy downtime and significant loss of revenue.

Environmental Challenges

Hardware and components used in applications in automation often have to withstand considerable stresses and hostile environmental factors for long periods. Therefore, it is crucial that all components are rugged and feature technologies that help offset the harmful effects of their harsh operating environments.

Among the factors that present the most significant risks in automation environments include extreme temperatures, exposure to shocks and vibrations, heightened risk for power surges, as well as damage from static electricity.

However, the diversity of automation applications' operating environments means that CANopen system implementation is sometimes only possible with a significant degree of customization to account for unique environmental factors. For example, in some applications, such as in the oil and mining industries, exposure to hazardous pollution can prove an important challenge to hardware integrity.

Solutions

CAN Bus Expansion Cards

The hardware requirements of the implementation of a CAN bus system (e.g., one that uses the CANopen or SAE J1939 higher-level protocols) can be addressed cost-effectively and with great flexibility by leveraging CAN bus expansion cards. CAN bus expansion cards that use standard input connectors such as mPCIe, 5-pin header, and M.2 and common interfaces such as USB and PCI-Express offer a high degree of flexibility—allowing them to be used in a diverse range of automation devices and equipment. Thus, system operators do not have to redesign devices or their hardware to accommodate for CAN bus support. Instead, CAN expansion cards can easily be plugged into existing expansion ports to provide the desired functionality.

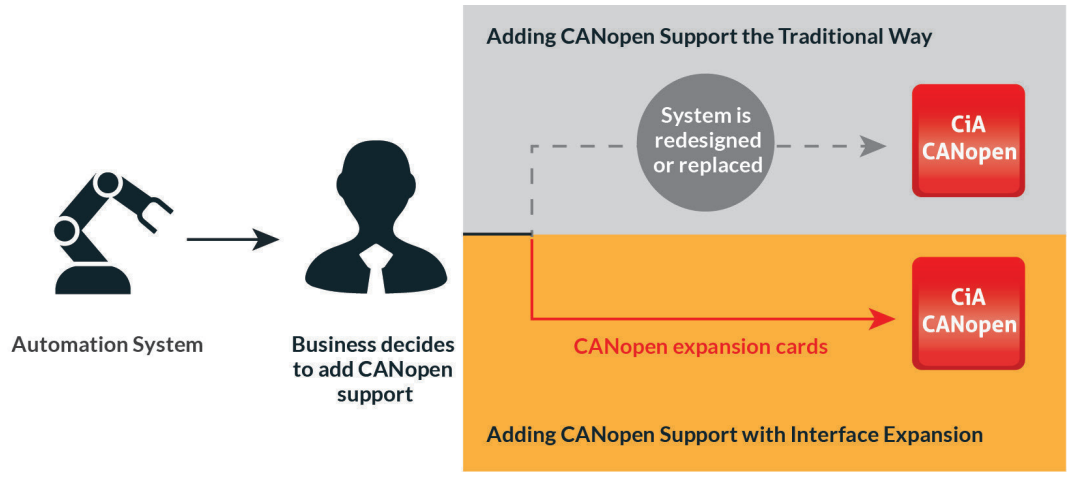

Image 3: CAN bus expansion cards make it easy to add CANopen support to existing systems.

Interface and Software Support

With such CAN bus expansion cards, there are many important considerations for system integrators to take into account. On a more fundamental level, system integrators need to make sure that the expansion cards support their desired higher-level protocols, e.g., CANopen and additional protocols such as SAE J1939 if they desire added flexibility. It is also important to ensure that the expansion cards can be implemented in one's software environment, whether it is a Linux-based or Windows-based system, and whether it is running on an ARM or x86 architecture.

However, only ensuring basic compatibility hardware and software compatibility fails to take other critical factors into account that can have a substantial impact on the time and costs associated with implementing CANopen support in a system. The customizations and programming that are necessary to ensure that the new CANopen system functions at a satisfactory level often require significant resources and risk causing significant delays.

Therefore, it is paramount to select expansion cards that provide adequate software and after-sales support. For instance, comprehensive APIs can significantly reduce the need for time-consuming and costly customizations necessary to guarantee a full system integration. Moreover, testing utilities and sample code can help system integrators verify that their systems are fully functional and running according to specification and user expectation. In the event of unexpected difficulties, particularly complex systems, or insufficient in-house expertise, comprehensive and dependable after-sales service and customization can also prove invaluable to meet deadlines and stay within budget.

Purpose-designed Hardware

CAN bus expansion cards are not only required to support the right connectors and interfaces, but they must also fulfill the specific hardware requirements of their applications. First and foremost, they need to physically fit into the device or equipment they are intended to be used in, which can be a considerable challenge due to the often limited space in devices used in automation. In some circumstances, the expansion cards may even require customized designs to fit the exact requirements of the devices they are to be installed in.

System integrators also need to make a thorough assessment of the intended operating environment for such expansion cards to determine what environmental factors may pose a risk to hardware integrity and longevity. Any interface expansion cards must incorporate technologies or design solutions that account for these environmental and operational risk factors to a satisfactory degree. For instance, many automation environments

may require expansion cards that can withstand extreme temperatures. In equipment or devices subject to shocks and vibrations, expansion cards may require to be fitted with mounting holes to maintain a sufficiently secure connection to its host device.

By taking these risk factors and environmental challenges into consideration when procuring CAN bus expansion cards, businesses can ensure a successful implementation of a CANopen system in their automation applications that provides the desired performance without risking downtime and costs associated with hardware failure.

Conclusion

CANopen presents businesses with an excellent opportunity to improve manufacturing processes with a high-performing communication system optimized for the smart automation applications of the future. While the interface expansion to accommodate CANopen can seem prohibitively difficult and expensive, this challenge can be successfully mitigated by leveraging CAN bus expansion cards that offer high performance, extensive software support, and the ruggedness required by automation applications. Because of their modularity and durability, such expansion cards provide a simple and cost-efficient method to implement CANopen support in any automation system.

References

1. https://www.polytechnichub.com/applications-controller-area-network-can-bus/
2. https://www.maximintegrated.com/en/design/blog/can-bus-continues-its-grand-popularity.html
3. https://www.iso.org/standard/20380.html
4. https://www.innodisk.com/Download_file?5D99A0E7C762CE71BD312D6E814156AF5F973DFFB0F066810050784EAA6049B736067CDA23521E3D92791FCF8D9355CF8163CA55F00214B74324161FD2D174E05B88B110BCEBA75C3ACEDA3CB6A45E3ECC82CEF6EF46ED08
5. https://en.wikipedia.org/wiki/CAN_bus#CAN-based_higher-layer_protocols
6. https://www.can-cia.org/about-us/
7. https://www.can-cia.org/canopen
8. https://canopen.readthedocs.io/en/latest/od.html
9. https://www.csselectronics.com/screen/page/canopen-tutorial-simple-intro/language/en
10. https://www.ni.com/zh-tw/innovations/white-papers/13/the-basics-of-canopen.html

Innodisk Corporation

5F., NO. 237, Sec. 1, Datong Rd., Xizhi Dist., New Tapei City, 221, Taiwan
Tel : +886-2-7703-3000
Fax : +886-2-7703-3555
E-Mail : sales@innodisk.com
Website : www.innodisk.com