BUS SYSTEMS

Klaus Bender, Rolf Birkhofer, and Markus Bregulla
Institute for Information Technology in Mechanical Engineering (itm), Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany

Peter Wenzel
Profibus International, Haid-und-Neu-Strasse 7, 76131 Karlsruhe, Germany

Keywords: Industrial Automation, Fieldbus, Data Transmission, Parallel Bus, Serial Bus, Protocol, Service, Industrial Network, Network Access Methods, Network Management

Contents

Summary

This chapter presents an overview of the principles, implementations and deployment areas of bus systems.

At first a general discussion about the requirements that led to the development of communication systems is given in Section 1. It will be demonstrated, that in the communication domain from a user perspective not only the technical specifications are necessary to understand but also the ideas of *openness*, *conformity* and *interoperability*. This can be explained with an inherent characteristic of the need for communication systems: Communication nearly always requires the integration of components and implementations of different manufacturers. In order to allow such integration an agreement upon the interfaces is necessary. Within Section 1, also some generic principles are presented which finally leads to the differentiation between parallel and serial bus systems. According to this classification the following sections are arranged.

The Section 2 deals with techniques of parallel bus systems, which are often employed within components (e. g. *Personal Computers*). In contrary, serial bus systems are used between components mainly due to the fact that the wiring costs are dramatically lower than in parallel systems. The special mechanisms of serial bus transmission will be presented in Section 3.

1. General Reflections

So far, automation systems of all branches are built up by a variety of heterogeneous peripheral components. The characteristics of computing and peripheral components vary depending on the required tasks. Starting from the enterprise control layer down to the actuator/sensor layer various computer systems, system controllers and field devices are used. The overall function of an automation system is successful as long as all components fulfill their partial functions in time. For example, in running automation processes *orders* are transmitted to the actuators according to defaults from control computers. For this they need information about the condition of the automation process which is delivered by the sensors. In addition, conditions and other information about the automation system can be transmitted to the leading computer where it is observed and analyzed by users and affected by its interventions.

In the past these controllers and leading computers were central units in the automation process that were built up by a single efficient processing unit which has controlled the whole process or an autonomous partial process (monolithic arithmetic unit).

The attainable degree of automation was in such systems, compared to the current, relatively slight and the so structured systems comparatively inflexible, i.e. high expenditures had to be made if changes had to be carried out in the automation processes, e.g. by a product changeover.

The progress in electronics, communication and information technology caused the entry of a new architecture of such automation systems.

On one hand, computers became increasingly efficient over the years (e.g. faster processors, enhanced memory, more efficient algorithms) and on the other hand new knowledge lead to progresses in methods, procedures and architecture concepts.

Today it is possible that heterogeneously manufactured computers can interact and exchange data under real time conditions (within the time borders given by a respective process) by distributing arithmetic tasks so that more complex tasks within an essentially smaller period can be mastered.

As a result automation systems were transformed into distributed systems in which automation processes are managed at several places at the same time whereby involved components are synchronized with each other and required information is exchanged according to a partial task.

1.1. Bus Systems within Automation

The information between computers is transferred via the so-called bus systems. Such a bus system consists of:

- Some wires or bundle that connect at least two components for the purpose of data or message exchange.
- Specified services and protocols as well as structures that regulate the data exchange.
- Component interfaces.

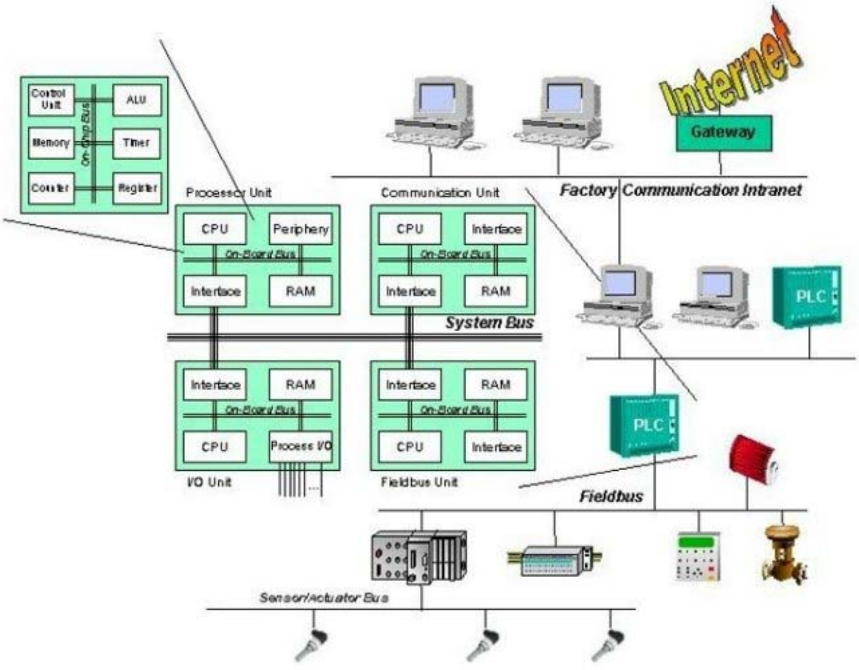

Figure 1: Employment Domains of Bus Systems

The development of various automation system architectures from a centrally oriented towards a decentralized structure followed several steps. Today, these systems are decentralized at such a high level that even components with a direct interface to the process enclose an increasing degree of local arithmetic performance, so that already a pre-processing of the process data is possible. Such a distributed hierarchical automation process can be synchronized and managed only with the support of suitable and optimized communication systems that connect different systems. Fig. 1 shows a typical system architecture and the variety of used communication systems distinguished into two sorts of bus systems - parallel and serial.

The parallel bus systems connect primarily computer units and clusters under the notion of computing busses. In Automation such a computer can be, e. g. a modularly constructed *Programmable Logic Controller* (PLC), a *Numerical Control* (NC), a *Robot Control* (RC), a *Personal Computer* (PC) or a database server.

The serial bus systems connect these computers and peripheral automation system components such as field devices, sensors or actuators, together in a network. They are known as Local Area Networks (LAN) whereas the notion LAN is still frequently understood as part of the office communication. In production areas the notions control bus, field bus and sensor-actor-bus have established themselves depending on technical process intentions. However, there is no precise distinction between both areas.

An essential distinction between parallel computer busses and serial LAN’s lies in the fact that parallel computer busses consist of a larger number of specialized control lines which transmit a defined number of data bits containing data, addressing information and control signals for the data transportation in parallel. LAN’s as well as field busses transmit an address and data information package using a time-serial signal stream on the bus wires.

A specific feature of parallel busses is that larger data sets within very short intervals can be transferred. However, this is limited to a short distance (few meters). Certainly, the serial data busses need more time for the data communication, however, they are able to transmit data over larger distances (up to some kilometers).

Communication systems frequently found in automation areas are system busses that connect components of a modular constructed computer and different types of LAN’s - particularly field bus systems. The different type of LAN’s underlie similar principles which can represent different complexity and variants, depending on various conditions of the application.

The *on-chip* and *on-board*-busses process according to identical principles like system busses since they simply differ in the geometric expansion. The so called WAN’s (Wide Area Networks) process similar to LAN’s. However, in certain areas they are imposing other principles because they are structured in a different way. WAN’s from a topological viewpoint are usually built up as a free network and can be identified by a larger size. Frequently the general designation network is applied here instead of the notion bus system.

The message exchange between the net components takes place via interfaces. The interfaces enclose a physical part (electric and mechanical interface) and protocols that regulate the exchange message flows.

Therefore, a bus system represents an interface definition and encloses an exchange module that displays a partner specific part of interface functionality. As long as the "understanding" between participants has to be precise an interface must be specified carefully.

This applies first of all when a so-called "open" system should be used. A bus system is named open when its specification is generally accessible at full extent (e. g. like an international standard or a published company standard).

Since large numbers of distributed teams concurrently and simultaneously develop products to co-operate in a single system a precise definition of the processes and structures of a particular interface is a precondition for the interoperability between components.

During the last years the principle of system openness was put through in several business areas sensitive to a successful interface functionality such as the communication technology sector.

Communication technology offers various technical and economical advantages. A large variety and flexibility distinguishes open systems, because they can be composed from a range of different manufacturers.

At the same time functional identical products can be substituted (e.g. because of a lower price). The following chapter considers the main issues of open and parallel system busses and serial busses sensitive for process data.

However, since other types of communication concepts follow similar principles compared to the concepts presented above an extensive fragmentation of all communication technology principles is not intended here.

1.2. Basic Functions

The task of a bus system is to perform the exchange of messages between at least two computers. In this light, exchange conditions play an important role. Smaller or larger sets of data can be transferred under timeliness conditions required from a system in order to guarantee the correct application functionality. Hence, the errorless transmission of messages is equally important.

Since these conditions are, at least in some cases, of contradictory nature, exchange mechanisms are optimized according to dominant conditions. This however led to the variety to be found in the domain of bus systems. However, a bundle of uniform basic functions can be extracted which fulfil the task.

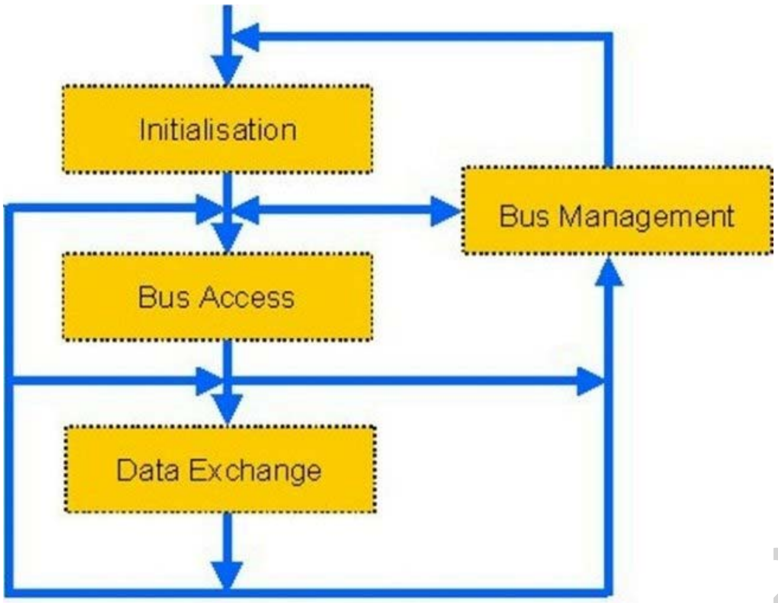

Figure 2: Generic State Machine of a Bus System

The uniform basic functions can be clustered into the four groups *Initialization*, *Bus arbitration*, *Data exchange* and *Bus Management*. Fig. 2 displays the possible transitions between functions of these groups.

Initialization functions occur at the start-up of a system (cold start), at a system reset e.g. after a detected error or in case of system re-configuration. Here, the terms module, bus and system initialization have to be distinguished. In distributed systems with several potential bus users it is necessary to guarantee that two participants do not transfer data on a shared medium at the same moment in time. This is guaranteed by the bus access function (arbitration). This function determines the participant which gets the right to start up data transmission for a defined period of time.

The mechanisms for data exchange form the core of parallel computer bus specifications. For doing this, different writing and reading mechanisms are available for transmitting small or large sets of data as well as specific messages. The bus management serves as a monitoring feature of the described functions and of the general condition of the participants as well as the bus itself. This includes the bus system configuration as well as functions relevant to the enterprise such as voltage monitoring.

The sequences of message exchange (bus arbitration and data transmission) and the supporting functions (initialization and management) are regulated by protocols. Beside the physical mechanisms and electrical signals (e. g. the bit coding) a protocol lays down the logical processes as there are sequences and time dependencies of single protocol steps as well as formats of data and control information.

Furthermore, definitions of mechanical properties e. g. connectors, layouts of printed circuit boards, quality and number of transmission wires are required. The components which include all specified rules are compatible in such a manner that they can be brought together to a functional computer or automation system. Such components meet the demands of conformity with respect to a particular bus system specification and therefore have in principle the ability to interoperate, i.e. to the error free co-operation with other conformal components.

The fulfillment of these characteristics implies that the specified protocol is complete and consistent. How well such a bus system can fulfill the tasks of an automation system lies in other features, for example in sufficient capacity, real time ability and avoidance of deadlocks.

1.3. ISO/OSI Layer Model

Busses are communication systems and in general structured according to the ISO/OSI reference model. This model was developed on the basic experiences concerning various communication systems of different nature.

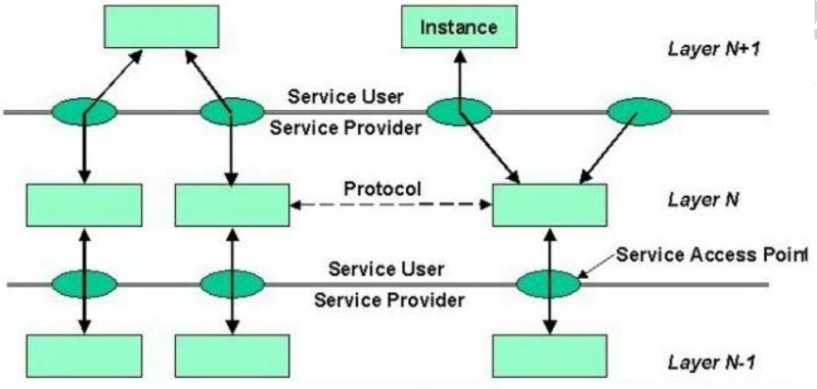

Figure 3: Services and Protocols

Generically, the ISO/OSI reference model is based on structural principles, layer arrangement and the service concept. Within the seven ISO/OSI layers different functions of communication systems are distinguished. From the physically most concrete layer up to the most abstract layer they are classified in bit transmission (or physical transmission), security, mediation, transportation, communication control, presentation and application.

Each of these layers contains the service concept as a central paradigm which defines an abstract interface between two neighboring layers. (See Fig. 3). Every service is produced by a protocol. The specified protocols of a layer build up the specification of a perceived layer. Protocols specify communication processes between communicating participants (instances) on identical layers (horizontal communication). The protocols support the kind of services which are offered by an (N-1)-layer instance to the service provider. According to the ISO/OSI concept, interaction occur by claiming services between the service user and service provider (vertical communication). These services are handled by *service primitives*. A service primitive is an abstract and invisible interaction element that is independent from the sort of the service provided. The ISO/OSI model knows four service primitives:

- Request
- Indication
- Response
- Confirmation

In accordance with the ISO/OSI-model there are two fundamental service types, the confirmed service (See Fig. 6 for a more detailed example) and the unconfirmed service, where the latter are using only the primitives request and indication.

Communication systems, that are specified according to the principles of the layer architecture can be implemented on a modular basis. This supports system openness, allows flexible and optimized realization (in hardware and software), and meets the target of software reuse.

1.4. Implementation Aspects

Parallel computer busses represent a way of information transmission within an automation system component, equipped with a microprocessor or controller. In contrast, serial computer busses in general deliver means for information transmission between automation system components and communication with external computer equipment.

Because the global task is comparable in both cases, common characteristics at the modeling level as well as general functionality exist. Differences rather lie in the development of these functions that are caused by different applications.

With the goal of data exchange between system components, they must be physically and functionally connected. This requires a medium enabling data exchange on the physical level as well as process specifications with unambiguous syntax and public semantics represented by hardware and/or software to regulate the data exchange. The different interface aspects concerning parallel and serial busses can be found in the following chapters.

-
-
-

Bibliography

Bender, K. (1993). *Profibus: The fieldbus for industrial automation*, Englewood Cliffs: Prentice Hall.

Birkhofer, R. (2001). *XML for Automation Devices- A Multi-Schema Approach*. Proceedings of the XMLeurope 2001 Conference; May, 21st-25th 2001, Berlin.

Black, U. D. (1989). *Data Networks, concepts, theory and practice*, New Jersey: Prentice-Hall Inc.

CENELEC (1996). *General Purpose Field Communication Systems. CEN CENELEC standard EN 50170.*

Gustavson, D. (1984). *Computer Busses – A Tutorial*, published in: IEEE Micro, Aug. 1984, pp. 7-22.

Kent, A. (1995*). Encyclopedia of Microcomputers : Strategies in the Microprocessor Industry to Tcp/Ip Internetworking : Concepts, Architecture, Protocols, and Tools) Vol. 17*, Marcel Dekker.

Profibus Nutzerorganisation (1998). *Profibus-DP Extensions to EN 50170 (DPV1), Version 2.0.*

Shanley, T.; Anderson, D. (1999). *PCI System Architecture*, Reading, Menlo Park: Addison Wesley.

Biographical Sketches

Prof. Dr.-Ing Klaus Bender, after completing his studies in Electrical Engineering between 1964-1969 in the University of Karlsruhe, specialized in the field of Case Regulation and Control Techniques, with Prof. Otto Föllinger in 1973. In 1974 he worked in the Institute for Data Handling in Ruhr University of Bochum in the new field of Micro Computers. He was invited to participate in the Faculty for Information of the University of Karlsruhe where he remained active until 1992. Together with six colleagues he founded in 1984 the Research Centre for Information FZI in the University of Karlsruhe and remained on the board of management until 1992.

In 1992 Prof. Dr.-Ing. Klaus Bender followed the invitation of the Technical University of Munich to head the Institute of Information Technology in Mechanical Engineering. In the Research Centre for Information in the University of Karlsruhe, he remained associated as Director in the field of Micro Computer research MRT. Prof. Dr.-Ing. Klaus Bender is since 1985 a member of the board of management and advisor to the VDI/VDE Society for Measurement and Automation (GMA) and is furthermore the Curator of the Fraunhofer Scientific Society in the Institute for Graphical Data Handling in the University of Darmstadt, since 1987. He was decisive in the development of German Field Bus Techniques such as PROFIBUS and ASI, and is a member of the board of management of the PROFIBUS User's Organisation since 1989 as well as Interkama since 1996.

Dr. Rolf Birkhofer studied electrical engineering at the Technical University Karlsruhe, Germany. He subsequently worked in the field of industrial control and communication at the "Institute for Information Technique in Mechanical Engineering" at the Technical University of Munich, Germany. In 2001, he concluded his research work about model based description technologies for field devices with the graduation to the doctorate degree. Since 2002 he is working for the companies Endress+Hauser,Reinach(CH) resp. CodeWrights, Karlsruhe(D) in the domain of device description and integration technologies.

Prof. Dr.-Ing. Markus Bregulla, born 1963 in Silesia, has studied Automation and Computer Science on the Silesian Technical University in Gliwice. After graduating as a digital electronic engineer, he has started working as a software and hardware developer of automation systems and as a trainer for developer of controls and systems. 1995, he left the industry and began his career as a scientific assistant on the Technical University of Munich. There, he earned the title of doctor in faculty of mechanical science. Now, he is a professor for Automation and computer science on the University of Applied Sciences in Ingolstadt.

Dr. Peter Wenzel studied electrical engineering at the University of Karlsruhe. After two years activity within ABB in the area of quality assurance, he changed to the Research and Computer Science Center Karlsruhe (FZI) and completed his thesis. His main field activity was the handling of research & development projects, and later management of projects in the areas development of field bus interfaces, test systems and tools for field bus systems. Dr. Wenzel was responsible for the start-up and operation of the Test and Certification Center for PROFIBUS products at the FZI. As the department manager, he was also responsible for all activities within the research area of microcomputer technology. Since the beginning 2001, Dr. Wenzel is the acting managing director at PROFIBUS International.