USING REAL-TIME SYSTEM TO MANAGE PRODUCTION DATA IN A DISTRIBUTED SYSTEM

Magdalena Dobrzańska, Paweł Dobrzański, Artur Łopuszyński, Łukasz Kurowski

Abstract: *The article presents the design and construction of data acquisition module, which allows read and write data counted by the data acquisition board. The acquisition was controlled using a program that was created in the QNX Momentics development environment, based on the architecture of the RLP. It involves the measurement card control directly by the read and write to memory registers. Data acquisition was performed on a computer running a hard real-time system QNX6 Neutrino. With the use of available tools QNX Neutrino RTOS was realized network communication process for managing production data in a distributed system. The article presents also industrial networks such as Profibus and Ethernet. Discusses the characteristics of distributed systems and the requirements for real-time systems.*

Keywords: *real-time system, data acquisition, distributed system*

ACM Classification Keywords: *J. Computer Applications, J.2 Physical Sciences And Engineering, J.7 Computers In Other Systems,*

Introduction

Real-time systems belong to the specific areas of modern computing. They are used in places where it is necessary to present the response to external signals. Real-time systems operating continuously since the implementation of the removal phase. They are designed to generate responses in parallel with the course and changes in the external process. They are found at almost every stage of human life, starting with simpler systems for applications such as audio equipment, the high-class nuclear power control systems, anti-aircraft weapons, serving the airport. Their use requires the timely fulfillment of critical tasks, because their failure could lead to danger for human life or significant property damage. Recently, real-time systems are becoming increasingly important and high conversion are increasing their functionality, so that they can meet the new standards of safety and security requirements. Significant impact on this have the rapid technological progress, decreasing costs and increasing capabilities of microprocessors.

The phenomenon of data acquisition presents a very important topic in real-time industrial systems. It's successful implementation allows to optimize the manufacturing process. The data obtained in the acquisition process is the main source of information for management support systems. Modern real-time systems can meet the demands posed consists in continuous operation, safety and reliability. Perform measurements, together with their development in an automatic cycle, according to the prepared algorithm. Computerization greatly reduces the time of measurements, allows them to make them in dangerous locations, difficult to access for a man. A large amount of data also allows to perform various analyzes and statistics. Real-time systems are the basis for control of complex technological processes in many industries, and in many areas of human life can play an increasingly important role over time.

At present, there has been considerable progress in measuring and control systems. Leaves from centralized systems toward distributed systems. This phenomenon is a consequence of the development of computer and electronics. Currently, the underlying elements of distributed measurement systems are the nodes that have the ability to process data and provide two-way communication with other elements forming the system. Network nodes allow direct control of objects, performing the functions of measuring, control or measuring and control, thereby, the primary source of data that must be sent in a reliable and efficient manner. Providing access to the advanced electronic systems equipped with multiple communication ports and a number of development tools, enables the distributed data processing. Distributed data processing is related to the performance of several tasks that are interdependent which affects their sequence, are also often asked time thresholds, defining the maximum response time to events occurring in the system. Ensure compliance with these requirements can use the industrial network of real-time operating system to control operations, that provides the tools and features to ensure compliance with restrictive time constraints, and efficient management of available resources and the tasks undertaken.

This article presents the commonly used network communication mechanisms used in distributed measurement and control systems, industrial networks and the principles of real-time systems in this context. In addition, the process was carried out data transmission between the node responsible for the acquisition of data to another node, which are based on client-server.

The features and types of real-time systems

Real-Time System is a computer system in which calculations are carried out in parallel with the course of an external process, they have to supervise and timely responses to the ongoing events in this process [Ułasiewicz, 2007]. It's correctness depends not only on the logical correctness of the calculations themselves, but also on the time in which the result appears. Real-time system corresponds predictably to external signals flowing in an unpredictable manner [Lal, 2003].

The time factor is present in every system. However, in some systems it is very important and in others less important.

Defines two main types of RTS systems in connection with the different effects of non-compliance with time limit:

- a rigorous real-time system (a system of hard time limits) - this is a system in which the time limit must always be met. These include control systems, missiles, nuclear power plants, aircraft, missile defense systems. Exceeding the limit of the response time could lead to danger to life or health, or in substantial property damage, and not a significant amount of time limit for answers, just the mere fact of his crossing. Hard real-time system must therefore ensure timely fulfillment of critical tasks, otherwise it becomes useless.
- Mild real-time system (a system with soft time limits) - this is a system in which time limits can sometimes be exceeded. The tasks are executed as soon as possible, but need not necessarily be completed within the specified time, because it does not cause material damage and does not threaten human health. Used in equipment such as multimedia.

In addition, another kind of RTS can be represented by systems with strong real time constraints - the delay in the response causes the result generated by the system becomes unusable, but it causes no danger to humans or equipment.

Real-time systems must be characterized by high reliability, ensuring that users have confidence in the services they provided. The quality system is determined by the attributes of reliability [Ułasiewicz, 2007]:

- availability - the system continuously provides its services;
- reliability - is working for a long period of time without occurrence of defects;
- security - an instance of a system failure can cause a catastrophe;

– integrity - no unauthorized changes in system are allowed;
– maintainability.

Real-Time System includes all necessary components to meet specific requirements such as: hardware (CPU, memory, peripherals, etc.), operating system and applications. Contrast, real-time operating system (RTOS) is one component of a complete system of RTS, which is usually supervises the whole system. Provide appropriate functionality that makes the whole system can meet the demands facing it.

His ability to respond to interrupts is very high, and applies expropriate scheduling strategy. It consists in the fact that at any time, may receive a process or thread that has a higher priority than currently executing. It expropriates the process and goes into the exercise, while the lower priority process is suspended. RTOS should work as planned by the user, to support different hardware platforms, be well documented and free of errors.

Acquisition of data in measuring systems

Data acquisition means the collection of electrical signals from the sensors, the measurement of their parameters, and transfer to computer for testing and processing. The signals are collected from the environment, then stored on computer and instantly analyzed to obtain relevant information and results. Physical quantities are measured by sensors equipped with a microcomputer, then the resulting signal is converted to digital form, which is sent to computer. The computer makes the analysis, visualization and documentation of the data obtained.

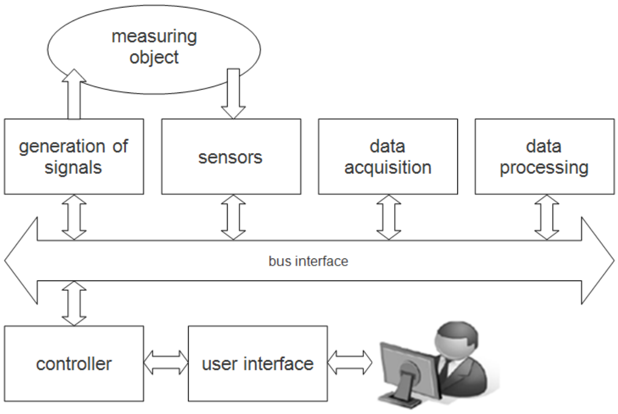

Fig. 1 Traditional measurement system [Rak, 2003]

The traditional measurement system is a set of basic elements coupled together. It contains the following function blocks (fig.1):

1. Sensors - allow receiving information from the test object, used to convert physical quantities (in example temperature, pressure) into electrical signals.
2. Data Collection - to read the signals from the sensors.
3. Data processing - processing of the measured signal.
4. Controller - responsible for controlling the entire system, and managing the flow of information. Frequently it is the role of a computer or microcontroller.
5. User Interface - provides user interaction with the measuring system.
6. Bus interface - connects the individual blocks measuring system ensuring the flow of data between them.
7. Signal Generation - responsible for generating signals in the measurement and control systems.

Successful implementation of the data acquisition process to optimize the production process in real-time industrial systems. The dynamic development of computerization linked with support for industrial processes resulting in an increase in the information provided, the processing of which is a challenge for modern systems. Computerization greatly reduces the time of measurements and allows to measure in dangerous and inaccessible to man. A large amount of data also allows to perform various statistics. Data taken from the microprocessor control systems are a major source of information for management support systems.

Transferring data in a distributed production control system

Continuous development of computer networks and increase their reliability led to the expansion of distributed systems in the control of the production process, thereby displacing centralized systems. Individual tasks can monitor a single computer or programmable logic controller (PLC), which manages real-time system, and communication between the components of the measurement system provides a communication network. The development of distributed control systems stimulate such factors as the continuous increase in the number of sensors and actuators, resulting in an increase in network cabling industry, increasing demand on communication media, security and flexibility and reliability solutions. Real-time systems, along with a computer network should ensure compliance with these requirements and be the guarantor of a certain time of the tasks related to the technological process.

Consisting of four layers of enterprise information model in a hierarchical method presents the informatics structure of the industrial system.

Starting from the lowest it includes the following levels:

I. Control in real time
II. Visualization and supervision over the production process
III. Tracing the production and material, optimization
IV. ERP/MRP

Control in real time includes equipment, whose task is to gather information about the process and control it. The first level provides the link between man and machines and technological devices. The most commonly used devices in this layer are microcomputers, programmable logic controllers, remote transmission devices and multifunctional controllers and local workstations. Visualization and supervision over the production process, the second level is used to configure measurement and control devices and monitoring of the technological process. To accomplish these tasks requires a dedicated application programs. The main function of this layer is to provide an interface between human and machine (MMI - Man Machine Interface). Thanks to him it is possible to influence product quality and the quantity of waste materials and energy. Levels outlined above are closely related to each other and their functions are sometimes intertwined. Production and material tracking, and optimization are the third layer of the model. It is the interface between layers I and II and ERP systems. Its main tasks include: visualization and monitoring of production, creation of documentation and reports, quality management, controlling the flow of materials and means of production. Provides the ability to modify the production plan on an ongoing basis depending on the circumstances arising. The introduction of level III results in takeover by him of the functions performed by the lower levels. Fourth level systems include information systems planning and management of the production process. Enable management resources such as procurement, finance, costing, planning, forecasting and optimization of the process in terms of improving quality and reducing costs. The boundaries of the above-mentioned layers are contractual and depend on the implementation of the system. Therefore, certain features may be implemented at different levels. Communication between the different layers can be realized with the industrial networks and computer networks and their integration provides lower costs and improves management of the company and the production process.

The smooth operation of the measuring system and it meets the requirements assigned to it is possible by providing communication between the various components of the system. Communication is not only data transfer but also transfer timing of commands and time synchronization operations. To the process of communication is needed: hardware combination and adaptation of the system and for the efficient management is the responsibility of the software. Interfaces are called standardized communication systems, which combine elements of the measuring system. Mechanical, electrical and functional system and cables, connectors, controllers and software create an interface system. Communication path between the elements of the measuring system called the bus interface, which consists of a set of rails. As we understand the set of bus-rail lines designed to transmit specific information [Tumański 2007].

There are two basic ways transmission of measuring data, they are: serial transmission and parallel transmission. In the serial transmission bits are sent sequentially on a continuous basis, bit by bit at the specified frequency, while the parallel transmission bits are transmitted sequences of defined length, word for word. The way of data transmission is the basis for division of the measuring network, which can use a serial interface or parallel interface.

Overview of selected methods of data transmission in industrial automation

Requirements for industrial networks are different from those which are placed before the local networks used in offices, public facilities, universities and schools. This is largely due to the need to work out the network in widely different circumstances. The basic requirement of industrial networks is the timeliness of communication, which is the guarantee of the proper conduct of the controlled process. These networks, in contrast to local, provide connectivity between sensors, controllers and actuators.

In addition, industrial networks should ensure high data transfer efficiency, which are usually short communications, high reliability and safety. In the measurement and control system, we can distinguish three groups of networks:

- sensorbus, includes sensors and actuators
- devicebus associated with level control
- filedbus on the department level

Each of these groups can be assigned to the individual, the previously discussed levels, from the first, showing the structure of the information system. The OSI model, which describes the structure of communication network consists of the following seven layers: physical, data link, network, transport, session, presentation and application. The measuring and control systems most often used as three of the seven layers are: physical layer, data link layer and application layer. This is due to the fact that the network requires good communication parameters, reduce costs and simplify the communication structure.

Profibus among the most popular industrial communication solutions. They are often used in measuring and control systems, in which we can distinguish the level of sensors and actuators, the level of production and the level of faculty. At the level of production, data is transmitted periodically. Devices occurring at this level are: I/O modules, valves, transducers, and pre-setting engines. Profibus includes recommendations and standards for the three layers of distributed systems, these are the applications layer, communication layer and physical layer. In the communication layer, we can distinguish three protocols: PROFIBUS-DP (Decentralized Periphery), PROFIBUS-PA (Process Automation) occurring at the level of sensors and actuators, and PROFIBUS-FMS (Fieldbus Message Specification) is used to communicate with these devices. PROFIBUS-DP protocol is used in systems with distributed actuators. It is characterized by a high speed and resistance to interference transmission and low costs and high efficiency. This is the most widely used protocol. PRFOIBUS-FMS protocol is now supplanted by TCP/IP. Profibus system sizes depend on the technology specified in the physical layer. It is recommended to use standard transmission lines such as RS-485, IEC 1158-2 and fiber. Profibus is a set of standards and regulations, which aim is to organize the production management both in the sphere of hardware and software, thus providing a common communication protocols, and compatibility and interchangeability of components. The communication structure of Profibus systems is a type of a master - slave.

Ethernet allows for the creation in the industry of a uniform platform for data exchange between computer systems and industrial automation systems. Has a high compatibility, versatility and flexibility of configuration. Allows connection of many thousands of devices with varied functionality, while providing them with direct communication. The continuous increase in transfer speeds and low costs of setting up a network, meant that Ethernet was applied in industrial networks. Figure 2 shows how the various devices can connect to an Ethernet network and to what extent covered the whole of manufacturing company. Forms the basis of Ethernet TCP/IP has four layers are: the data link layer, network layer, transport layer, application layer. Based on the Ethernet physical layer, numerous protocols used in the industry. Here are the most popular ones: Modbus/TCP, EtherNet/IP, Ethernet Powerlink, Profinet, EtherCAT, SERCOS III. Occurrence of so many varieties of Ethernet in the industry due to the need to preserve compatibility with older techniques, networking and with networks that enable real-time job. TCP is not designed to work in real time.

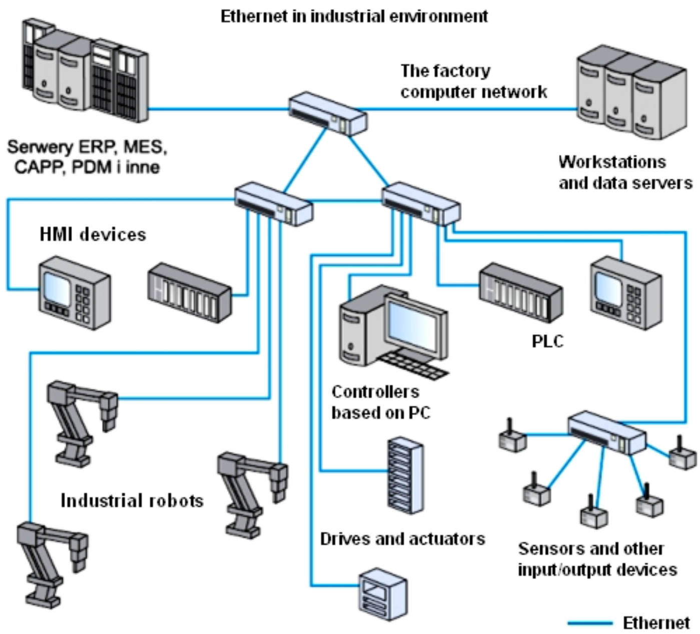

Fig. 2. Ethernet in industrial environments. Source [Automatyka B2B].

Wireless networks are used in a situation where because of the difficult physical conditions cannot use a serial cable or a data source object is in motion. Through the use of wireless technology, especially at lower levels of the hierarchy network pyramid can facilitate the work directly on the production line by using the portable control panels and remote measurement, particularly in endangered areas or hard to reach places. Wireless communication is a new trend in network industries. Frequently it is based on standard Ethernet, 802.11x (0,1, b, g), which is commonly used in office and public networks. Despite its universality, this standard is not fully suitable for use in demanding industrial applications, because does not provide an adequate level of data security and is not compatible with the communication interfaces of industrial automation equipment. They have been developing new communication protocols for industrial networks based on Ethernet for example, WirelessHART

(IEEE802.15.4). The popular and rapidly developing technology is Bluetooth, which is seen as an extension of the functionality of wireless networks.

Distributed systems in the control of production

Distributed control system is responsible for visualization and control of the production process having a common database. Individual components of the system communicate with each other using a specific communication protocol, so that they can work independently. The main objectives in creating a distributed system is to connect users and resources, transparency, openness, fault tolerance and scalability. Connecting users and resources means that the principal objective of the system is to facilitate access to remote resources and sharing them with other users in a controlled manner. This facilitates communication and exchange of information in the system. Transparency involves concealing the fact of the physical dispersion processes and resources. There are several types of transparency. Transparency of access is related to the differences caused by hiding data representation format and how to access resources. Location transparency is hiding information about the physical location of the resource. Transparency migration allows the transfer of resource to a different location without having to change how to communicate with him, and the change in location of the resource while using it to ensure transparency movement.

Transparency of multiplication is related to the of hiding information about the presence of multiple copies of the resource. The independent use at the same time by different users with the same resource allows distributed systems, and the concealment of this fact corresponds to the transparency of concurrency. It is important to split the resource remains in a consistent state. The hiding of failure is related to transparency failures. Information on whether the resource is in memory or in persistent storage hides transparent persistence. Openness is another goal of distributed systems. This property is associated with the ability to expand the system by adding external devices, memory, adding new communication protocols and resource sharing services without affecting existing services. The openness is related to the ability to work, which involves the cooperation of system components from different manufacturers due to the existence of a common standard. Features of a distributed system of openness aimed to ensure that such a system extensible and allow flexibility in the addition of different parts. Fault tolerance is associated with the possibility of completing the task started in spite of a hardware failure. For this purpose, the redundancy of hardware, which consists of allocating redundant hardware to perform the same tasks. In the event of failure of any of the equipment of the other takes over its task. Scalability is the last mentioned features a distributed system. It provides a possibility of increasing the scale of the system and software without the need for changes in them [Tanenbaum 2006].

Data acquisition module

Developed data acquisition module includes the following: NI 6601 data acquisition board, encoder, connector block, wiring necessary to connect the card with encoder, and the software controlling the operation of the card. Card is placed in the computer running the system QNX6 Neutrino, and also is connected with the connector block tape. The connector cables that are plugged in the analog signals generated by the encoder, which in turn are read and converted by the software-controlled data acquisition card. Connector block simplifies the connection of two different interfaces, data acquisition card and the encoder.

NI Measurement 6601 card was plugged into the PCI bus on the motherboard computer running QNX. The card comes with an NI-DAQmx driver. During the driver installation program is also installed Meauserement & Automation Explorer (MAX), which automatically detects all installed equipment and NI software installed.

If possible is recommended to use the driver, because it provides the greatest benefits of the equipment. NI DAQmx driver could not be used, however, because the Ni 6601 measuring card is not compatible with the QNX Neutrino system. In a standard situation, for example, under Windows or Linux, NI-DAQmx driver and other software supplied with it could be used, so that further work would simplify considerably.

When it is not possible to use an existing driver, you must use the architecture of RLP (Register Level Programming), which is used to control the device directly through reading and writing to memory registers without calling a function provided by the driver. To register a data acquisition board in the QNX Neutrino operating system is necessary to know about:

- Vendor ID (vendor ID device) - this number is identical for all devices NI (National Instruments) to be incorporated into PCI/PXI. This number can be found at the manufacturer's data acquisition card (www.ni.com), is it: Vendor ID = 0x1093.
- Device ID (Product ID device) - allows to identify a specific model. This number can be found in the NI documentation (ID = 0x2C60).

Using these two numbers, the system can identify the NI 6601 data acquisition card in system. Software controls the operation of data acquisition card, allowing to read and convert the pulses sent to the encoder. Software made in the QNX Momentics development environment, using the architecture of the Register Level Programming, available from National Instruments (NI). This architecture provides several functions to access measurement cards, on systems that are not supported by the hardware manufacturer, and facilitates communication between the hardware and the system.

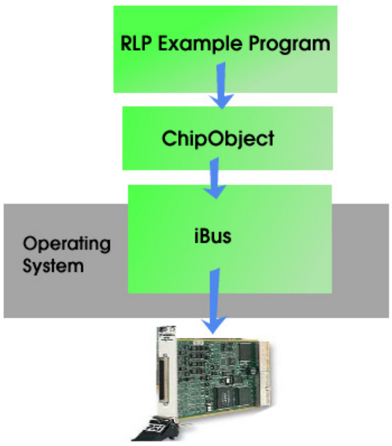

Fig. 3 Three-layer model of programming the memory registers [MHDDK]

No possibility to use the driver supplied with the data acquisition board forced its configuration under the control functions to read and write directly to the memory registers. The program configures the data acquisition card by setting the appropriate registers for its proper operation, and controls the data acquisition process.

In order to facilitate work on the created program or driver, the NI company divided RLP architecture into three layers, defined as osiBus, RLP ChipObject, RLP Example Program.

Figure 3 shows the three-layer model of RLP. OsiBus layer provides a simple interface between the PCI/PXI and running programs, also provides a simple I/O functions performed by ChipObject the memory registers. The supplied interface is designed in such a way as to be able to work with any operating system.

ChipObject an abstract reflection of Logic chip devices used, and thus consists of several registers and bit fields, using to their definition of logical names. This helps to avoid numerous errors and easy to use equipment on many different operating systems using the RLP. It is used for configuration and programming of the device by using the methods defined in C++ classes and functions osiBus layer. It may happen that the device will contain more than one chip is responsible for control and configuration, then cannot avoid a situation where one counter is controlled by two systems, since it could lead to damage. ChipObject can prevent this situation by assigning to each chips separate address spaces [MHDDK]. The final element to facilitate the programming of the memory registers are RLP examples. They include the implementation of specific operations, which makes it possible to perform a specific device. Using for that purpose the basic functions provided by the previously described RLP layer. Knowing the construction of memory registers user can easily adapt the chosen example to suit your needs.

The acquisition software, first searches the data acquisition card in system. Then it initializes the interface MITE, in order to connect to the PCI bus. These activities allow access to the data acquisition card from the system. Then the program uses the registers function sets up the data acquisition board (counter mode, the value of the initial count, the internal clock of the card (frequency 20 MHz), and turns the counter). After configuring the card program performs the data acquisition process, which are displayed and saved to a file in real time.

Measuring card is to look at a bus address returned by the program ("PXI0::8::INSTR"). It is used for this function bus = acquireBoard("PXI0::8::INSTR"), which returns both the base addresses of the measuring card (BAR0, BAR1). Address BAR0 refers to the memory address register MITE, while bar1 to registers of the data acquisition card (TIO). Based on the addresses returned is the address space that allows access to the data acquisition card registers. Described function returns only addresses BAR0, BAR1. Functions are performed on registers address space address BAR1, and therefore it is necessary to provide the address BAR1 and to highlight its address space. On the basis of the returned basic addresses interface is initialized using the MITE function initMite(bus). It aims to make available, the connection address bar1 plate measuring board and gain access to the registers of data acquisition board.

After initializing the interface MITE main function is responsible for configuring and controlling measurement board (test(bus)). At the beginning of the function returns the address space of address BAR1 shared by the function initMite(bus), which provides registers of the measurement board. It is used for this function CardSpace = bus-> createAddressSpace (kPCI_BAR1). Function test(bus) consists of two main parts. The first of these sets contains the functions responsible for the configuration registers and control the measurement board: set the timer, the value of the initial count, counter mode, the gate signals, the internal clock (sampling frequency), and turns the counter. After setting registers and turn counter is done the second part of the function test(bus), which is responsible for the data acquisition process. The data is sent from the encoder. A measure of the encoder are two separate channels, the signals A and B of the measuring board (fig.4). Using the two signals (A and B) are offset by 90°, to distinguish the direction of rotation of the encoder.

The use of two offset signals is also used when determining the measuring board read the signals from the encoder and to add the pulse, and when subtract. In addition, there is yet another signal that serves as a marker to indicate when it was made a full turn. Set in the X1 mode (fig. 5) means that the values are counted in 0-2000 (with a value of 2000 will be made full rotation encoder).

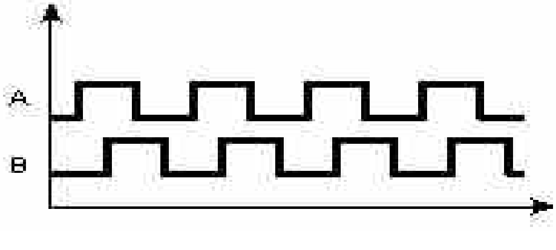

Fig.4. Signals offset by 90 °

Counter mode (set in the first part of the function test(bus)) can be changed using the register function: board-> G0_Counting_Mode_Register.setG0_Encoder_Counting_Mode (1); Changing the counting is done by changing the numbers passed as a parameter to the function:

- 1 - X1 mode
- 2 - X2 mode
- 3 - X4 mode

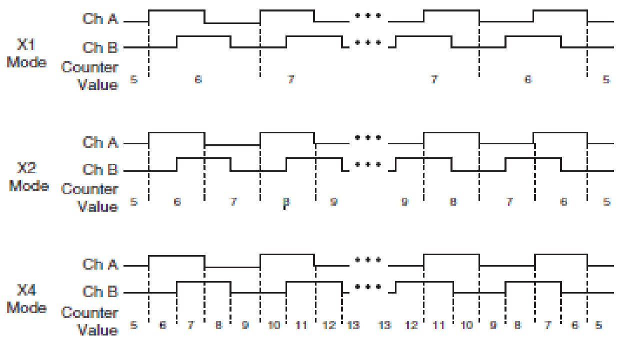

Fig. 5 Available counting modes [DAQ NI, 2002].

The second part of the function test (bus) performs the data acquisition process. Measure board compares the described signals transmitted from the encoder, and then using a timer, count them according to the mode of counting X1 shown in Figure 5. Counter values are displayed in real time on the console and stored in a file. Using a loop is set to the read counter value data which are read by the function:

board->G0_Save_Registers.readRegister()

After making number of counts the acquisition process was completed. It is still disconnecting the board from the bus, and releases the resources assigned to it. Counter disconnected by using the the board-> G0_Command_Register. writeG0_Disarm (1), system resources allocated to the start of the function test(bus) released by: bus->destroyAddressSpace(cardSpace)

The release at the end of base addresses (BAR) returned by the function acquireBoard ("PXI0 :: 8 :: INSTR") at the beginning of the acquisition, using releaseBoard (bus).

The data obtained in the acquisition process specify the position of the encoder at a time. Set in program counter mode X1 means that pulses are added in the range 0-2000, depending on the relative positions of the signals, according to the diagram counting mode. If the direction of rotation to the left encoder data is reduced in the range of 0 - (-2000). The data obtained in the acquisition process can be converted based on the algorithms, it allows the determination of position, displacement, distance, speed. The encoder can be combined with additional mechanics, allowing you to perform various measurements. If the direction of encoder rotation is to the left, data is reduced in the range of 0 - (-2000). The data obtained in the acquisition process can be converted based on the algorithms, it allows the determination of position, displacement, distance, speed. The encoder can be combined with additional mechanics, allowing to perform various measurements.

Today, encoders are used widely in industry. They are used for the implementation of various measurements, they are able to obtain specific information regarding such trading performed by the element in the machine or work piece to be treated. The data obtained in the acquisition process also allow you to improve the manufacturing process for industrial machinery.

Production data management module in a distributed system

The data transfer process for monitoring and controlling the production process is closely linked with systems of data collection and processing. These elements together form the measurement system, which can be a single position, e.g. the measurement system in an industrial laboratory, or may take the form of a distributed, while its range covers many areas and the distance between the devices is greater than the length of the interface cable. The basic elements of the measuring system is a computer or dedicated microprocessor controller, whose job is to control the transmission of data and their processing and archiving. Often specialized measurement systems are controlled using a computer operating system, which is the basis for launching programs, which enable the realization of tasks measuring system including the presentation of processed data. Distributed real-time system implemented in this article consists of two nodes, one acting as the position used for data acquisition, the other to a server to which the read data is sent (and which data can be visualized). These nodes are connected by an Ethernet network. For communication between nodes the Qnet protocol is used. In addition, the network is connected to the computer with the QNX Momentics development platform. Network nodes are controlled by the real-time operating system QNX Neutrino while application development is done on a remote host running on Windows. The position used for data acquisition consists of: data acquisition card from National Instruments PCI6601, PCI slot connected to the computer, dedicated to the card, data cable and connector R6868 CB-68LP, interface for connecting a source of signals. As the source signal encoder is used. Figure 6 shows a simplified diagram of a distributed measurement system using the previously described elements.

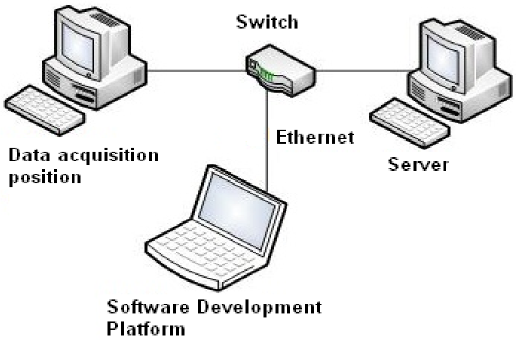

Fig. 6 Diagram of a distributed data acquisition system with software development platform.

In order to implement interprocess communication, provide the following parameters: the node identifiers and a process known as NID and PID and the channel known as CHID. The applied method of communication works very well when communicating processes run on one node. In a distributed system is the difficulty of the processes running on different nodes to provide access to information about the required IDs. The solution to this problem is to use the services of the global name server known as GNS (Eng - Global Names Service). It allows to connect to the server using for this purpose its name. In this paper, the communication between a client process running on a separate node, responsible for data acquisition and transmission and the receiving server process read from the data acquisition card is made using the GNS. To do this, follow these steps:

- check the operation of the network Qnet, whose correct configuration is required to use the GNS
- start the node used to data acquisition gns program in client
- start the server program gns running in server mode
- on the server, run a program that will receive the measurement data
- on the node for data acquisition, run a program that supports the operation of the measuring card and whose operation has been extended with features for sending messages.

Figure 7 shows the sequence to run the various server and client programs, and illustrates how it is sending the message and when processes are locked and unlocked.

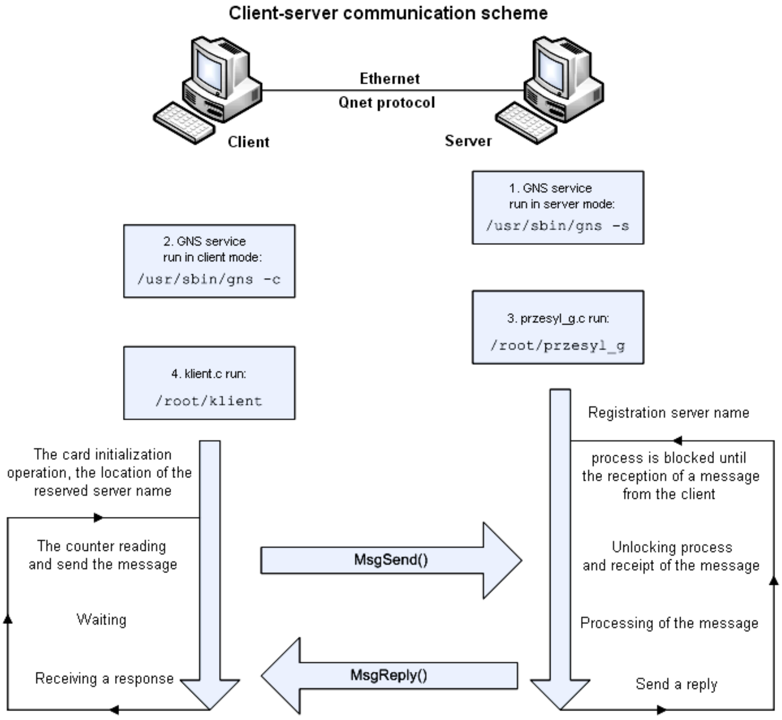

Fig. 7. Diagram of client-server communication.

GNS Running in server mode and client is possible when the user has administrator privileges. The GNS in server mode is responsible for maintaining the database of registered trademarks, and supports sent by the client process requests the name and location makes the connection. The gns when working in client mode transmit request location to the server and creates a connection to it. It is also possible run programs gns automatically at boot time by placing it in the INI file system. To start data transfer, run the appropriate program on the server would not otherwise be registered with the name of the server process and client program after running reports an error. Similarly to the server process data read from the encoder are saved to a file. The method of communication uses a simple interprocess communication mechanism using the GNS name administrator making it possible relatively easily create distributed measurement and control system.

Conclusion

Informatics company model can be illustrated by dividing it into four levels, ranging from the lowest level of sensors and control level, the level of faculty associated with the tracking of production and materials through to the level that includes planning and management systems using ERP/MRP. Now the aim is to integrate measurement and control systems with systems operating at higher levels of the presented model. This trend is caused by the desire to reduce the operating costs of the company and increase its efficiency. The basis of integration of the various levels of company information model is to ensure the proper flow of data and enable agile information processing. Therefore, there are many different interfaces based on the serial, parallel or wireless data transmission systems which design and functional parameters are strictly determined by the standard. Mentioned interfaces are used for industrial networking its various organizational levels. Profibus is used in measuring and control systems, whose working conditions are monitored continuously in real time. Ethernet network are used in the enterprise-level planning and management. May be noted that Ethernet-based networks provide coverage lower levels of the present business model, it is a consequence of the emergence of market measurement and control devices, equipped with an Ethernet connector and adjusting these networks to meet the requirements posed in front of them real-time systems, and whose primary requirement is a timely response to changes in the controlled process. Companies typically provide coverage area in which to perform measurement and control processes, as well as management and planning processes, often carried out by individual network nodes, which work independently allow access to available resources, therefore such a network can be described as a distributed network. Real-time systems can be operated under the control of real time operating systems, which due to special design of the kernel and providing a number of tools to greatly facilitate the fulfillment of the requirements for control and measuring systems. System deployed to the process of sending data in this paper, is a system QNX Neutrino, which provides the user with a number of functions to support interprocess communication, which is characterized in that the communication between processes both in the area of one network node and between processes located on remote from each other nodes is possible to achieve using the same methods. Stage of the work involved in preparing the position for data acquisition showed that the major and how important it is to use measuring devices and operating system are compatible, which translates into effective use of the opportunities offered by the device to the system. Presented by the authors of the process of data transfer between two nodes includes simple, available in QNX Neutrino interprocess communication mechanisms. Thus, there is further development of the proposal to solve the problem or its improvements for more efficient flow of information in the network and a detailed analysis of the data transmission process using available in QNX Neutrino, mechanisms for meeting the requirements from real-time systems.

Bibliography

[Ułasiewicz, 2007] J.Ułasiewicz. Systemy Czasu Rzeczywistego QNX6 Neutrino, Wydawnictwo BTC, Warszawa, 2007.

[DAQ NI, 2002] DAQ NI 660X Register-Level Programmer Manual, National Instrument Corporation, 2002.

[Rak, 2003] R.J.Rak. Wirtualny przyrząd pomiarowy. Realne narzędzie współczesnej metrologii, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2003.

[Lal, 2003] K.Lal, T.Rak, K.Orkisz. "RTLinux – system czasu rzeczywistego", HELION, 2003.

[NI] National Instruments, http://ni.com/.

[MHDDK] Measurement Hardware Driver Development Kit, Architecture RLP.

[Tumański 2007] S.Tumański. Technika pomiarowa, WNT, Warszawa 2007. s 319-379.

[Tanenbaum 2006] A.S.Tanenbaum. Maarten van Steen: Systemy rozproszone. Zasady i paradygmaty, WNT, Warszawa 2006, s2-57.

[Automatyka B2B] Automatyka B2B. Ethernet przemysłowy. http://automatykab2b.pl/tematmiesiaca/1535-ethernet-przemyslowy

Authors' Information

Magdalena Dobrzańska, PhD, Eng*– Rzeszow University of Technology; Department of Computer Science, ul. W. Pola 2 Rzeszow 35-959, Poland; e-mail: md@prz.edu.pl*

Paweł Dobrzański, PhD, Eng*– Rzeszow University of Technology; Department of Computer Engineering in Management, ul. W. Pola 2 Rzeszow 35-959, Poland; e-mail: pd@prz.edu.pl*

Artur Łopuszyński, Eng*– Student Rzeszow University of Technology, Poland*

Łukasz Kurowski, Eng*– Student Rzeszow University of Technology, Poland*