BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 65 (69), Numărul 1, 2019
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

SENDING AND RECEIVING DATA THROUGH INTERNET BETWEEN USER PROCESSES RUNNING ON APPLICATION SYSTEMS EQUIPPED WITH ATMEL MICROCONTROLLERS

BY

PETRUŢ DUMA*

“Gheorghe Asachi” Technical University of Iaşi,
Faculty of Electronics, Telecommunications and Information Technology

Received: February 21, 2019
Accepted for publication: March 29, 2019

Abstract. The paper describes the hardware structure of an interface that, added to various remote user processes, allows sending and receiving data blocks through Internet, between application systems with ATMEL microcontrollers. A specialized device server performs, on one hand, a serial connection between the user application and its serial interface, and, on the other hand, the connection to the computer network.

Keywords: sending/receiving data through Internet; serial connection, communications protocols; device server; serial asynchronous interface; Ethernet interface.

1. Introduction

Nowadays, any system can be remotely commanded and controlled by one or several users, using computer networks. The users seek to remotely manage, configure, monitor, diagnose, command and control various electronic devices, which allow an increased efficiency for the companies in all domains.

In the particular case of the device servers XPort form Lantronix, or from any other producer for that matter, it is generally required to connect the

* *e-mail*: pduma@etti.tuiasi.ro

user application to a microcontroller or microprocessor based system featuring a serial interface and, respectively, to connect the device's Ethernet port to this device.

Once the device's serial port is configured, it becomes a network access port, with its own IP address and all the other parameters. The user connects to the serial port through the network, from a computer running a terminal emulation software that permits to send commands/data to the application as well as to receive state information/data from the application, as if it were locally connected, directly to the application's serial interface. Thus, through the serial interface, the application can be controlled and commanded from any point of the computer network.

This paper deals with the physical connection of such a device server, in order to perform through the Internet, the transmission and, respectively, the reception of data blocks between different microcontrollers from user application systems.

A future paper will deal with reading and writing data blocks communicated through Internet by user applications, from/into a website that can be subsequently be accessed by various user categories.

2. The Architecture of the XPort XP1001000

The XP1001000 device incorporates all the features required to operate in a network, including an Ethernet 10 Base-T or 100 Base-Tx, an operating system, an Web server, e-mail based alerts, 128 bit AES data encryption etc. It is built around a network DSTni chip (Device Server Technology Network Interface) with an efficient internal microcontroller that manages a SRAM volatile memory, Flash non-volatile one and various peripheral circuits.

The block schematics of the LANTRONIX XP1001000 device is depicted in Fig. 1. The acronyms have the following meanings: DSTni LX_MC – DSTni LX microcontroller; EI – Ethernet interface; UART – Universal asynchronous interface for data reception and transmission; Osc_1, Osc_2 – clock

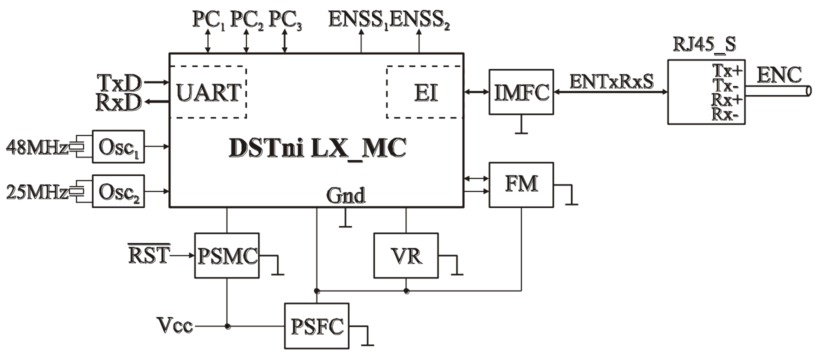

Fig. 1 – Structure of the device server XP1001000.

oscillator; IMFC – insulation and magnetic filtering circuit; RJ45_S – Ethernet RJ45 network socket; ENC – Ethernet network cable; FM – Flash memory; PSFC – power supply filtering circuit; PSMC – power supply monitoring circuit; VR – voltage regulator. The signals shown have the following meanings: ENTxRxS – Ethernet network data transmission and reception signals; $ENSS_1$ – Ethernet 10 Mbit/s network status signal; $ENSS_2$ – Ethernet 100 Mbit/s network status signal; TxD, RxD - data transmission and reception signals, from the serial asynchronous interface; PC_1, PC_2, PC_3 – programmable connections; $\overline{RST}$ – external initialization input; V_{CC}, Gnd – power supply connections.

Table 1 includes the external connections of the XP1001000 device, used for interfacing with the user application.

Table 1
External Connections of the Device XPort

Pin	Signal	Sense	Signal function
1	Gnd	–	Ground
2	Vcc	–	Power
3	$\overline{RST}$	In	External reset input
4	TxD	Out	Transmission data serial output
5	RxD	In	Reception data serial input
6	PC_1	In/Out	Programmable data input/output
7	PC_2	In/Out	Programmable data input/output
8	PC_3	In/Out	Programmable data input/output

The XPort module is commanded and controlled by the DSTni LX microcontroller in order to perform data transmission and reception through Ethernet network, but also for communicating with the user application through the serial asynchronous interface. The Ethernet interface implements the required layers of the OSI reference model in order to perform the two-way communication between the user processes. The signals for transmitting and receiving data through the network (Table 2) are provided to the RJ45 socket through an insulation and magnetic filtering circuit.

The serial asynchronous interface communicates with the user application in order to acquire analogic and/or digital signals from various sensors and transducers, but also to transmit specific commands. This interface allows for a transfer rate between 300 bit/s and 230 kbit/s.

Two quartz crystal driven oscillators, on 48 MHz and 25 MHz frequencies, provide all the necessary clock signals required for the module's operation.

The non-volatile Flash internal memory is loaded with different application management programs and with the corresponding web pages. The various data acquired is saved in real-time in a database in the Flash memory that can be subsequently accessed by users. For large data volumes, external Flash memory cards can be added to the application system, having storage capacities in the GB range, in order to allow saving the data locally; these memory cards are managed by a serial SPI interface.

Table 2

Connections of the Network Socket RJ45

Pin	Signal	Sense	Signal function
1	Tx+	Out	Differential Ethernet Transmit Data+
2	Tx–	Out	Differential Ethernet Transmit Data–
3	Rx+	In	Differential Ethernet Receive Data+
4	–	–	–
5	–	–	–
6	Rx–	In	Differential Ethernet Receive Data–
7	–	–	–
8	–	–	–
Shield	–	–	Connected to ground

The XPort device is powered by a DC voltage source of +3.3 V, that is filtered by an internal circuit. If the power voltage drops under +3 V, then a power supply monitoring circuit triggers a module initialization and maintains it until the power voltage rises over the threshold value; this monitoring circuit can also be initialized by the user with an external reset signal ($\overline{\text{RST}}$).

An internal voltage regulator of +2.5 V ensures the power for the device core that performs the data processing in microcontroller DSTni LX.

The internal structure of microcontroller DSTni LX, used for managing the XPort device, is shown in Fig. 2; the notes have the following meanings: DSTni LX_MC – microcontroller for Device Server Technology Network Interface LANTRONIX; 186_MPC – 186 microprocessor core; EU – execution unit; ALU – arithmetic and logic unit; GR – general registries; EU.C – execution unit command; BIU – bus interfacing unit; Q – instruction queue; SR – segment registries; S – summer; Bus.C – bus command; MEM – memory; BROM – boot read only memory; SRAM – static random access memory; PC/I – peripheral circuits/interfaces; UART – Universal asynchronous interface for data reception and transmission; PIH – priority interrupt handling interface; PIO – parallel input/output ports interface; C/T – counter/timer interface; EI – Ethernet interface; DMA – direct memory access interface; SPI – serial peripheral interface; WDT – watchdog timer interface; IB – internal bus.

The internal bus of microcontroller DSTni LX is used for connecting all the resources and consists of the data bus, the address bus and the command bus. The XPort device microcontroller includes the core of the I186 microprocessor that has a 16 bit data bus and a 20 bit address bus.

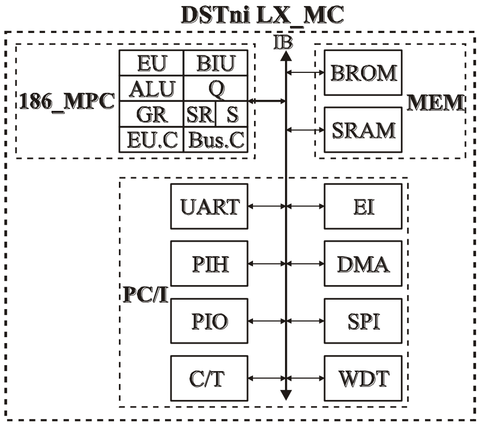

Fig. 2 – Structure of the microcontroller DSTni LX.

The central processing unit (CPU) consists of the execution unit and the bus interface unit; the two units work almost independently and simultaneously, in order to increase the processing speed. The execution unit obtains the instructions codes from the BIU or awaits until the instruction queue is empty, then performs instructions, arithmetical and logical operations, works with 8-bit or 16-bit data and addresses, updates the flags and provides addresses and data for the BIU.

The bus interface unit executes all the bus cycles at the request of the EU or for filling the instruction queue with codes. This unit performs fetch cycles in the intervals where EU does not request the bus. If the instruction queue is full, then inactive bus cycles are executed.

The non-volatile boot memory has a 2 KB capacity, containing by default the routines and functions required for loading user application programs and for implementing communication protocols.

The volatile memory is static and has a 256 KB capacity. This memory is used for storing and executing user programs, but also as temporary storage for data about to be saved into the flash memory database.

The Ethernet interface permits network data transfer with either 10 Mbit/s or 100 Mbit/s rates and implements the required levels from the OSI reference model in order to allow the application development based on various communication protocols.

The data circulating through the serial asynchronous interface are assembled in a format consisting of a START bit defined as logical 0, then the data bits starting from the less significant and finishing with the most significant bit (data can be 7 or 8 bits long), followed optionally by a parity bit (odd or even) and, finally, by one or two STOP bits defined as logical 1. The data flow control can be made through software, by opening and closing the communication channel (Xon/Xoff) or through the control signals of the RS232 interface, consisting of Data Terminal Ready ($\overline{\mathrm{DTR}}$), Data Set Ready ($\overline{\mathrm{DSR}}$), Request To Send ($\overline{\mathrm{RTS}}$), Clear To Send ($\overline{\mathrm{CTS}}$) and other signals. These modem control signals are obtained by software programming input/output lines of the parallel ports.

In order to gain access to the Internet, this XPort device integrates a TCP/UP network protocol stack, a completely developed operating system, an embedded Web server for implementing various user applications and several other features. The internally implemented communications protocol portfolio, based on functions and subroutines, is presented in Fig. 3. The notes have the following meaning: IP – Internet Protocol; ICMP – Internet Control Message Protocol; TCP – Transmission Control Protocol; UDP – User Datagram Protocol; SNMP – Simple Network Management Protocol; DHCP – Dynamic Host Configuration Protocol; SMTP – Simple Mail Transfer Protocol; AES – Advanced Encryption Standard (also known as Rijndael); TFTP – Trivial File Transfer Protocol; OEM – Original Equipment Manufacturer.

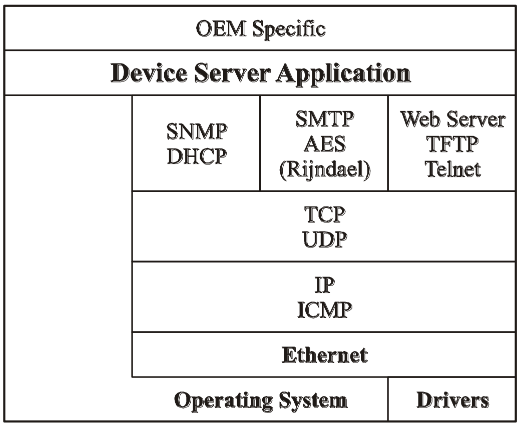

Fig. 3 – Communications protocols implemented by XPort.

The XPort device uses Internet protocols for network communications, addressing, routing and data block manipulation, while the transmission control protocol makes sure that the data is not lost, is not received twice and that it reaches its rightful destination unaltered. Other protocols from the TCP/IP stack are used for network communications, network management, serial port connection, firmware update, e-mail messaging, typical data communication applications, for instance those in which various devices interact with each other without maintaining a point to point connection, for updating Web pages etc.

3. Interfacing XP1001000 Device

The command and control of the XP1001000 device, used in the initial stage for checking and testing the application, is made with a development system equipped with a microcontroller from ATMEL family. The microcontroller used in this case is AT89S8253, powered at a DC voltage of +5 V. The structure of the interface to this device, performing the physical connection through Ethernet between various remote parts of the user application, is shown in Fig. 4.

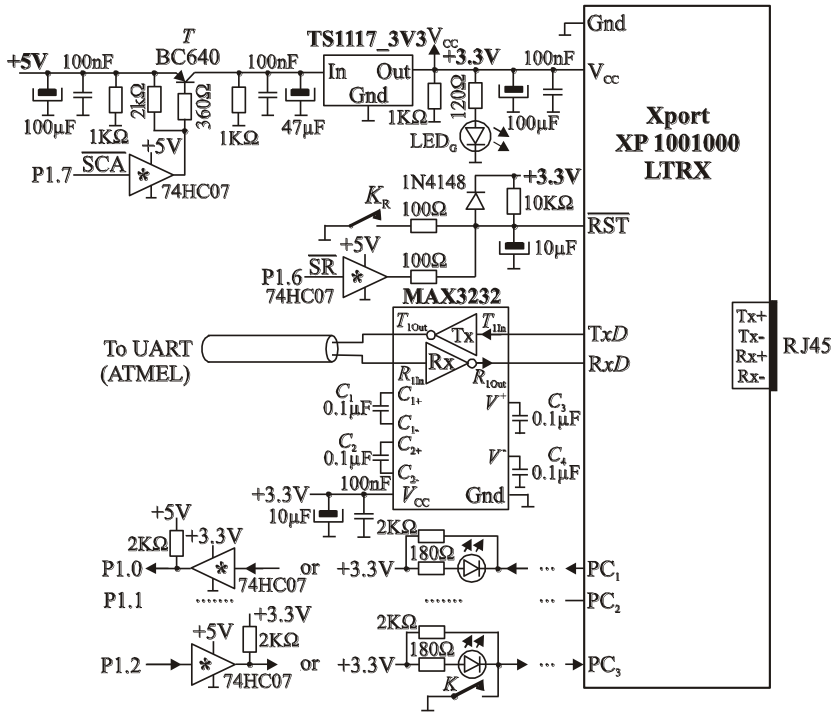

Fig. 4 – Interface with device server XP1001000.

The XP1001000 device can be powered within the voltage range from +3.1 V up to +3.5 V (+3.3 V recommended). The interface built in practice uses an XPort powered at +3.3 V, provided by the integrated voltage regulator TS1117_3V3. This regulator has a compact structure, requires only external filtering capacitors (100 nF, 47 μF) and is controlled through software. The power is commanded using an open drain buffer (74HC07) and a BC640 transistor (*T*) that switches a current in the range of hundreds of mA. The signal $\overline{\mathrm{SCA}}$ for power command is active on logical 0 and is software provided on line P1.7 of the ATMEL microcontroller. When the application communicates through the Ethernet network at an 100 Mbit/s rate, the XPort requires a nominal current of 210 mA. After the device is powered, a few hundred ms must pass for its initialization, in order for it to be able to communicate with the application microcontroller.

The initialization input ($\overline{\mathrm{RST}}$) of the XPort allows for resetting the device in one of the following situations: when powered on, on user action or by the application microcontroller through software. The duration of the initialization process is at least 200 ms. An external RC group (10 KΩ şi 10 μF) is connected at input $\overline{\text{RST}}$ for the automatic initialization when the system is powered on, along with a switch K_R, that permits the manual initialization when flipped by the user, and a open drain buffer (74HC07) in order to apply a software reset ($\overline{\text{SR}}$) transmitted on line P1.6 of the application microcontroller.

The communication between the XPort and the serial interface of microcontroller ATMEL is based on the MAX3232 circuit, from the communication interfaces family EIA/TIA-232, that is powered by the +3.3V DC voltage source. This circuit contains, basically: two change pump DC-DC voltage converters, two RS232 line receptors and two RS232 line transmitters. One of the internal converters connected to C_1 and C_3 capacitors delivers at output V+ the +5.5 V voltage, while the other internal converter with C_2 and C_4 delivers at output V – the –5.5 V. The transmitter inverts the logical level and converts it to the standard EIA/TIA-232, while the receiver converts accordingly the signal received from the line in logical levels and in the end inverts it.

The programmable input/output lines of the XPort device (PC_1, PC_2, PC_3) can be used for signalling on LEDs various indicators/internal status information of the user process or for allowing the user to insert various commands/operation modes or to configure the application using switches. These lines can be managed by the ATMEL microcontroller to send/receive various commands/data through open drain buffers.

The final user application is commanded and controlled by an application system equipped with an ATMEL microcontroller powered at +3.3 V, while the serial interface driver and the open drain buffers are no longer necessary.

4. Conclusions

The hardware structure described above and built in practice is simple and requires a minimal number of external components. This interface is based on Lantronix XP1001000 device, that requires to its connections a controlled voltage source of +3.3 V, an initialization circuit, a circuit consisting of line RS232 receptors and transmitters and a command microsystem equipped with an ATMEL family microcontroller.

The structure is used for sending and receiving, through Internet, data blocks from the user application systems. The ATMEL microcontroller also manages the user application process requiring data connections between various components of the application system, remotely distributed.

The XPort device offers the possibility to build intelligent interfaces, that feature the advantages of Internet communications and that may be used in various innovative applications in every field.

The command program that was designed consists of a set of functions and subroutines that permit the serial asynchronous and the Ethernet interfaces configuration in order to transmit and to receive data blocks between the user process microcontrollers through the computer network. At this stage, it is implemented a relatively simple user process, that requires few command data and that provides data acquired from sensors and transducers to be communicated through the Ethernet network.

REFERENCES

Aghion C., Ursaru O., *Aplicaţii practice ale microcontrolerelor*, Edit. PIM, Iaşi, 2009.

Aghion C., Ursaru O., *Informatică aplicată. Introducere în microcontrolere*, Edit. PIM, Iaşi, 2015.

Balan R., *Microcontrolere. Structură şi aplicaţii*, Edit. Todescu, Cluj-Napoca, 2002.

DragomirF., Dragomir O.E., *Programarea în limbaj de asamblare a microcontrolerelor*, Edit. Matrix Rom, Bucureşti, 2013.

Drăgan F., *Protocoale de comunicaţie*, Edit. UT Press, Cluj-Napoca, 2008.

Duma P., *Microcontrolere în telecomunicaţii,* Edit. TEHNOPRESS, Iaşi, 2007.

Duma P., *Microcontrolerul INTEL 8051. Aplicaţii*, Edit. „TEHNOPRESS", Iaşi, 2004.

Găitan V., Popa V., Tănase A., *Arhitectura reţelelor industriale locale*, Edit. Matrix Rom, Bucureşti, 2004.

Ionescu V.M., *Reţele de calculatoare. Aplicaţii*, Edit. Universităţii din Piteşti, 2015.

Ionescu V.M., Sima I., Sofron L., *Aplicaţii software pentru protocoale de comunicaţie*, Edit. Matrix Rom, Bucureşti, 2008.

Liţă I., Bănică L., *Protocoale de comunicaţie în Internet*, Edit. Matrix Rom, Bucureşti, 2007.

Pentiuc Şt.Gh., *Elemente de programarea aplicaţiilor pe Internet*, Edit. Mediamira, Cluj-Napoca, 2001.

Petreuş D., Muntean G., Juhos Z., Palaghiţa N., *Aplicaţii cu Microcontrolere din Familia 8051*, Edit. Mediamira, Cluj-Napoca, 2005.

Potorac A.D., *Transmiterea informaţiei în reţelele de calculatoare*, Edit. Matrix Rom, Bucureşti, 2009.

*** ATMEL, *CMOS Memory*, Data Book, 2004.

*** ATMEL, *CMOS Flash Memory*, Data Book, 2004.

*** ATMEL, *Family Microcontroller*, Data Book, 1998.

*** ATMEL, *AT89S8253 Microcontroller*, Data Sheet, 2005.

*** LANTRONIX, *XPort Embedded Device Server*, Data Sheet, 2003.

*** LANTRONIX, *XPort Technical Data*, Data Sheet, 2003.

*** LANTRONIX, *XPort User Manual and Development Kit*, Data Book, 2003.

*** LANTRONIX, *Comm Port Redirector Guide*, Data Sheet, 2003.

*** LANTRONIX, *XPort Sample Code and Solutions*, Multi-Family Dwelling Environmental Control and Monitoring, Data Sheet, 2003.

*** MAXIM, *Multichannel RS232 Drivers/Receivers*, MAX3232 Data Sheet, 2007.

*** Taiwan Semiconductor, *Low Dropout Positive Voltage Regulator*, TS1117_3V3 Data Sheet, 2003.

*** Texas Instruments, *MOS Memory Commercial and Military*, Data Book, 1995.

*** Texas Instruments, *Digital Logic*, Data Book, 2007.

*** TOSHIBA, *MOS Memory Products*, Data Book, 1998.

TRANSMITEREA ŞI RECEPŢIONAREA DATELOR PRIN INTERNET ÎNTRE PROCESELE UTILIZATOR CARE RULEAZĂ PE SISTEMELE DE APLICAŢIE CU MICROCONTROLER ATMEL

(Rezumat)

Lucrarea descrie structura hardware a unei interfeţe care adăugată diferitelor procese utilizator de aplicaţie aflate la distanţă, permite transmiterea şi recepţionarea blocurilor de date prin reţeaua Internet între sistemele de aplicaţie cu microcontroler ATMEL. Un device server specializat asigură realizarea, pe de o parte, a unei conexiuni seriale între aplicaţia utilizator cu interfaţa serială a acestuia şi pe de altă parte, asigură realizarea unei conexiuni de comunicare cu reţeaua de calculatoare