

TELEMATICS APPLICATION FOR A FLEXIBLE MANUFACTURING SYSTEM

Dorin Popescu, Florin Manta, Adrian Moraret, Sorin Vava, Livia Popescu

*University of Craiova, Faculty of Automation, Computers and Electronics
Department of Automation and Mechatronics, Romania
dorinp@robotics.ucv.ro*

Abstract: The aim of the work was to develop telematics applications to be used in an e-teaching context. A web based application has been developed, which links a programmable logic controller (PLC) with a computer for programming via Internet. The main requirement of our application is to provide a friendly user interface that allows PLC application developers to control a Flexible Manufacturing System (FMS) from anywhere in the world via a thin client that does not need the installation of any additional software on the user side. The user has possibility to see the environment (FMS) via images from a network video camera.

Keywords: telematics, web based application, flexible manufacturing system, programmable logic controller, open-source technologies.

1. INTRODUCTION

The high cost necessary to carry out experiments (for didactical purposes) with manipulators and robots in various environments led to the development of remote facilities where the physical system, sensors and operating environment can be at a great distance.

Some researchers achieved an analysis of the current challenges in Internet tele-programming and tried to find the possible solutions under the current environment, where there are some unsolved common problems associated with this new technology such as limited bandwidth and unreliable signaling (Backes *et al.*, 2000), (Bermudez *et al.*, 2001).

Remote laboratories are laboratory experiments that run remotely via a web interface. Usually, either the student can set some parameters on the web, then a software interface converts those parameters to a form that is accepted by the local computer running the experiment or he can tele-operate an equipment (Berntzen *et al.*, 2001), (Ewald *et al.*, 2000), (Zolotova *et al.*, 2006), (Grange *et al.*, 2000).

We proposed more, namely the student can achieve a program, which can run on the PLC. The student has thorough freedom to choose what experiment wants to do. The goal is remote programming of a PLC, which controls a process station for use in tele-education. The process station is controlled through PLC and Internet connection.

Web-based tele-robotic systems have only become available on the Internet in the last decade. In the beginning they utilized a CGI interface to access the robots (Backes *et al.*, 2000), (Schilling *et al.*, 1997). With the introduction of Java and its integration into web browsers, developers could create tele-operated systems that sustained an interactive link to the robot during its execution. Many new tele-robotic systems were created that gave the user much more control and provided functionality, which under CGI could have never been possible (Amin *et al.*, 2001), (Popescu *et al.*, 2005a), (Dinulescu *et al.*, 2005), (Marin *et al.*, 2001).

The Internet lab technology offers the students the opportunity to work with sophisticated equipment, of the kind they are more likely to find in an industrial

setting, and which may be too expensive for most faculties to purchase.

This paper gives a solution for tele-programming of a certain PLC. It tries to show an aspect of higher education and training based on Information Technology. The final purpose of the remote laboratory is to allow the student to take full control of the equipment, in order to fulfill the task required by the teacher.

2. THE FLEXIBLE MANUFACTURING SYSTEM

Flexible Manufacturing System FMS-200 is a Flexible Assembly System. It enables the development of various skills associated with pneumatic, electro-pneumatic, electrical, robotic and handling automatisms, programming and programmable logic controller (PLC) technologies, industrial communications, supervision, quality control and fault diagnosis and repair.

The system comprises a flexible automation cell which carries out an assembly process involving a number of predetermined parts, with a total of 24 different possibilities. Parts are transported between the different stations by an automated transfer line with corresponding stoppers and precision lifters-positioners. Parts are mounted on pallets. The pallets incorporate a magnetic coding system using inductive detectors which allow the control system to identify the position of each pallet at any time.

The process stations function either independently of the transport system, or integrated into it. The stations are situated around the transfer line, and may be withdrawn for repositioning in a different order, moved for future extensions of work in a completely independent and self-sufficient mode.



Fig. 1. The Flexible Manufacturing System FMS-200

Each station has its own electrical panel, where the wiring system and PLC are fully visible for study, while new elements may be fitted to the panel if desired. This electrical control panel may be made entirely independent at each station for use in PLC

training. The front of each station incorporates the start, stop, and single and continuous cycle pushbuttons.

The bought system (Fig. 2) was composed by: body feed-positioning station (1), bearing handling-fitting and measurement station (2), hydraulic bearing pressing station (3), shaft handling-fitting-measurement-selection station (4), cover handling-fitting-selection station (5), station for unloading – storage - palletization of completed assembly (6), 4-metre transfer line (0).

The system is modular and may be extended, allowing future incorporation of other process stations according to user needs. Due to FMS-200 is a flexible system we increased the possibilities of this system by development of new station (7 in Fig. 2).

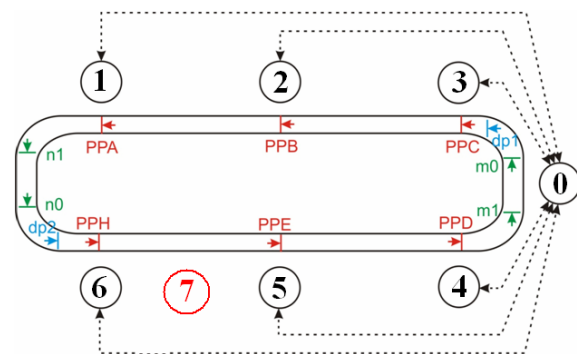


Fig. 2.

This screwing station (Fig. 3) allows that a magnetic screw-driver to assembly the components through 4 screws.



Fig. 3.

The standard layout includes a PLC, Mitsubishi model FX2N-32MR. The FX2N-32MR PLC has 16 inputs and 16 outputs, timed and programmed interrupts, internal PID, high-speed counter and 0.08 µs instruction time. It incorporates an RS-422 port for programming, and an RS-485 port for communication with other PLCs or PCs. The PLCs which govern the cell stations communicate with each other via a network.

3. COMMUNICATION FORMAT

The FX Series Mitsubishi PLCs allows 5 types of communication:

- 1) N:N network
- 2) Parallel link
- 3) Computer link (Data transfer using dedicated protocol)
- 4) No protocol communication (Data transfer using RS instruction)
- 5) Optional programming port

The communication format decides the communication setting (data length, parity, and baud rate, etc.) between computer link and no protocol communication (RS instruction). The communication format can be set using the special data register D8120 in a programmable controller. The setting of D8120 is done in accordance with external equipment used.

The PLC has some special auxiliary relays, for example:

- M8121 = data transmission delayed (RS instruction);
- M8122 = data transmission flag (RS instruction);
- M8123 = finished receiving flag (RS instruction);
- M8126 = global flag (computer link);
- M8128 = on-demand error flag (computer link).

The PLC has some special data registers, for example:

Special Data Registers	Description
D8120	Communication format (RS instruction, computer link)
D8121	Station No. setting (computer link)
D8122	Number of remaining data to be transmitted (RS instruction)
D8123	Number of receive data (RS instruction)
D8124	Data header <Initial value: STX> (RS instruction)
D8125	Data terminator <Initial value: ETX> (RS instruction)
D8127	On-demand head device register (computer link)
D8128	On-demand data length register (computer link)
D8129	Data network Time-out timer value (RS instruction, computer link)

The special data register D8120 has: 1 bit for data length, 2 bits for parity, 1 bit as stop bit, 4 bits for baud rate, 1 bit for header, 1 bit for terminator, 3 bits for control line, 1 bit for sum check, 1 bit for protocol, 1 bit for transmission control protocol.

When setting the content shown on the left, it is performed programming as follows"

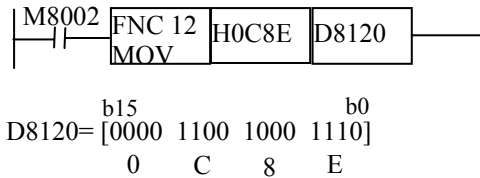


Table 1. D8120 register (example)

Data length	7 bits
Parity	Even
Stop bit	2 bits
Baud rate	9600 bps
Protocol	No protocol
Header	No used
Terminator	No used
Control line	Normal mode 1

The data flow by link when computer reads data from PLC is shown in figure 4. We needed to know the data flow between computer and PLC, the format of instructions, other information about communication between them. For this we had to „listen” the transmission between computer and PLC.

We had to respect some rules as:

- conditions under which the transmission sequence is initialized. The transmission sequence of the PLC is initialized during the following conditions: when the power is on; when data communication is completed normally; when control code EOT (end of transmission) or CL (clear) is received; when a NAK (no acknowledgment) control is received; after the time-out check time has elapsed.
- Occurrence of framing error at the computer.
- NAK response from the programmable controller.
- Command transmission from computer

In order to „listen” the transmission of the PLC program between computer and PLC we wrote a C program which runs on another computer. On the left side of the figure 5, data bytes (in hexa) transmitted by computer to PLC are shown. On the right side of the figure 5, data bytes (in hexa) transmitted by PLC to computer are shown. These data are the response of the PLC with regard to data send by computer.

Then we decoded the transmission protocol, the instruction code and the operands code, and wrote a

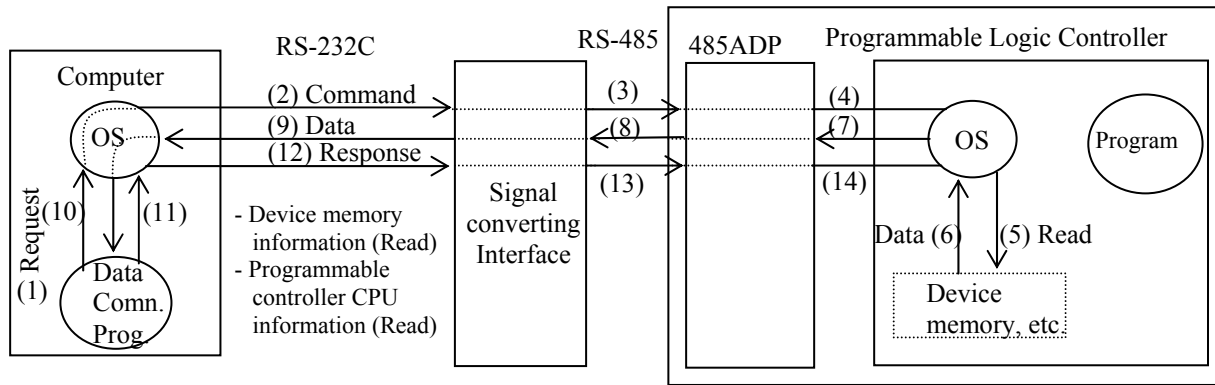


Fig. 4.

program for communication between Mitsubishi FX2N PLC and computer.

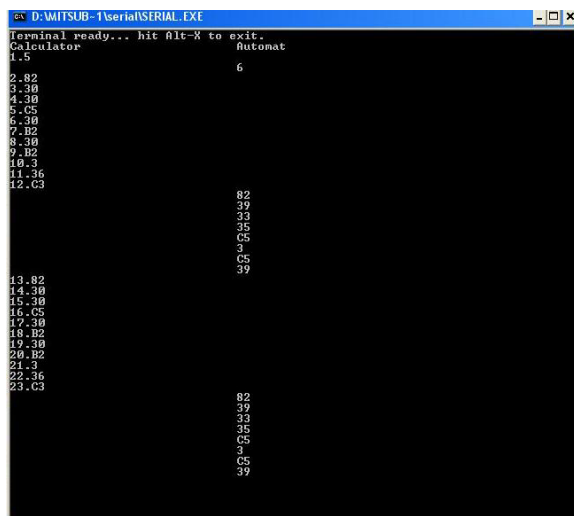


Fig. 5.

The program was written in Java and used to test the transmission (sending/receiving) of data (in hexa) from computer to PLC.

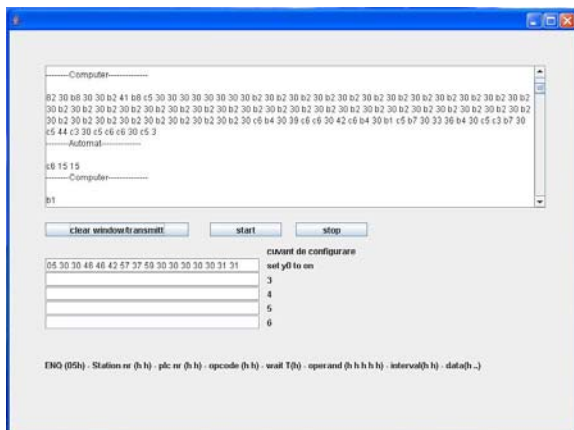


Fig. 6.

The main TextArea (Fig. 6) contains the data sending by the computer and the responses of the PLC. The Start/Stop buttons represent remote start and remote stop of the transmission. The Textfields from the

bottom of the window show bytes (in hexa) flow which can be send by the computer.

4. TELEMATICS APPLICATION

The goal of this project in the remote programming area is to discover and develop the system by combining network technology with capabilities of PLCs and process stations. The use of Internet technology for remote programming application offers the advantage of low-cost deployment. There is no longer a requirement for expensive purpose built equipment at each operator's location. Almost every computer connected to the Internet can be used to control a tele-operable device. The downside is the limitation of varying bandwidth and unpredictable time delays. These Internet features should be defined and considered before designing an efficient remote programming system. Besides that, several functional requirements should also be defined before designing any tele-operable system (Popescu *et al.*, 2005b).

This web-based application allows users to perform own experiments remotely from another computer. Using a standard web browser and a connection to the Internet, the user can write his program for the PLC.

After this, the user uploads his program and the results can be viewed on screen. The experiments have video streams available for visual monitoring.

4.1 Performance requirements and technologies

The main requirement of this application is to provide a friendly user interface that allows PLC application developers to control the devices from anywhere in the world via a thin client that does not need the installation of any additional software on the user side.

Another requirement is related to the costs that are involved during the project development (e.g. development tools, hardware etc.).

After considering several alternatives, it has been opted for web based open-source technologies that considerably reduced the implementation costs while accomplishing the other project requirements. Thus, Java were chosen for the application development (Flanagan, 2001), (Hall, 2000), (Kurniawan, 2002), (McLaughlin, 2000). In order to speed up the development process and eliminate software bugs, additional open source libraries and APIs have also been used. The role of each one within the whole system will be explained in the next paragraphs.

4.2 Application architecture

The general system structure is described in Fig. 7. The only software needed on the user side is a conventional web browser that can run JavaScript scripts and Java applets. This application has been tested and optimized to work on the most popular web browsers including Internet Explorer and FireFox, while other ones are currently being tested.

The Java server application is responsible for the application administration module, that involves users management, communication with PLC, PLC program editor etc.

An AXIS web camera is used for sending live video of the process to the user's browser. This camera model includes a built-in WEB server running under Embedded Linux that transmits the video stream either via an ActiveX control or a Java applet. While the ActiveX transmission method gives better results, the Java applet is a good choice for browsers that don't support ActiveX technology.

4.3 Implementation details

Application is achieved using Java and contain two modules: the client and the server. The client application is an applet hosted by website

department. The server application is hosted by a computer connected to Mitsubishi PLC via RS232 connection. The server application uses COMM API for connection with PLC.

The client GUI is presented in Fig. 8. The client application offers buttons and an editor window for instruction list.

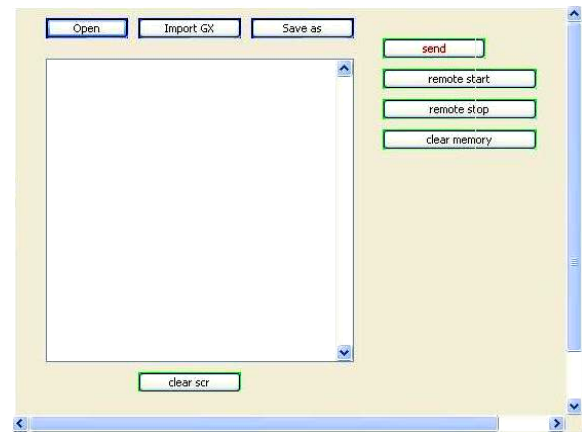


Fig. 8.

The editor window permits to write the Mitsubishi PLC program as instruction list. A syntax checking system was developed. If the user writes a wrong instruction mnemonic or a wrong variable, he will receive an error message.

The error message window indicates the number of line where it is the error and the wrong instruction mnemonic or the wrong variable. In addition, in the editor window the wrong instruction mnemonic or the wrong variable will be marked by red colour. In this way it will be more easier for user to correct the program.

The Java server application is responsible for the application administration module, that involves

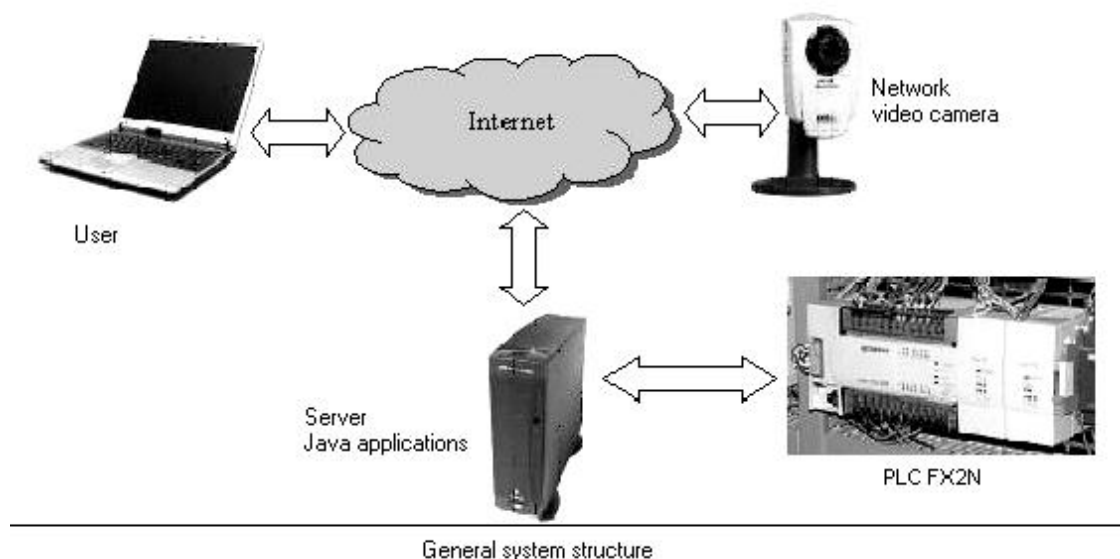


Fig. 7.

users management, communication with PLC, PLC status, etc. Only one user can be connected to server application (it means to Mitsubishi PLC).

The server translates the program from text to machine-code instructions and then it will send the program to Mitsubishi PLC at the transmission rate of 9600 bps.

A user with a computer and Internet has to work in the following way after he navigate through our website: he has to ask permission to work with our PLC; he will receive an ID and a password; he will connect to our server and he can read or download our documentation about system (PLC - process station); he has to answer to some questions and then he can write a user program with the original language of the Mitsubishi PLC; the application points out the syntactic errors; he can send his user program. The user program for PLC will be run and the user can observe the movements and operation of the process station by means of a network video camera (Fig. 9).



Fig. 9.

5. CONCLUSION

This project is currently being used at University of Craiova (in the PLC class). Such remote laboratory experiment method enables the student to use laboratory equipment, which is not usually available to the students.

The application has been tested inside the faculty where a computer has been connected to Internet that the user had used to program the PLC in order to control the process station which was in another laboratory.

Students can watch the process station via the application. For the majority of remote experiments via Internet in mechatronics, the student can teleoperate a robot or set some parameters on the web and then he can watch the robot movements. We performed more, namely the student can write a

program, which can run on the PLC in order to control the process station as he wants. The application is provided with friendly GUI. The exercises help students to understand and compare different process control methods using a PLC.

REFERENCES

- Amin, S., M. Zakaria, N. Majid, L. Siong, L. Horvath and J. Tar (2001). Internet-based telerobotics: UTM's experience and future direction. *The 10th International Conference on Advanced Robotics, ICAR 2001*, pp. 313-319.
- Backes, P., K. Tso, J. Norris, G. Tharp, J. Slostad, R. Bonitz and K. All (2000). Internet based operations for the Mars polar lander mission. *Proc. of the 2000 IEEE International Conference on Robotics & Automation*, pp. 2025-2032.
- Belmonte Bermudez, G., and M.A. Perez Sanchez (2001). Robots Tele-programming, *1st Workshop on Robotics Education and Training*, pp. 19-24.
- Berntzen, R., J.O. Strandman, T. Fjeldly and M. Shur (2001). Advanced Solutions for Performing Real Experiments over the Internet, *Int. Conference on Engineering Education*, pp. 21-26.
- Dinulescu, I., D. Popescu, G. Terejanu (2005). Web based Telematics Application using Open-Source technologies. *12th Int. Symposium on Systems Theory*, pp. 267-272.
- Ewald, H., and G.F. Page (2000). Performing Experiments by Remote Control Using the Internet, *Global J. of Engineering. Education*, Vol. 4, No. 3.
- Flanagan, D. (2001). *Java Script – The Definitive Guide*, O'Reilly.
- Grange, S., T. Fong and C. Baur (2000). Effective vehicle teleoperation on the world wide web, *IEEE International Conference on Robotics & Automation*, pp. 2007-2012.
- Hall, M. (2000). *Core Servlets and Java Server Pages*, Prentice Hall.
- Kurniawan, B. (2002). *Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions*, New Riders Publishing.
- Marin, R., and P. Sanz (2001). The UJI Telerobotic Training System, *1st Workshop on Robotics Education and Training*, Germany, pp. 25-30.
- McLaughlin, B. (2000). *Java and XML*, O'Reilly.
- Popescu, D., I. Dinulescu, L. Popescu (2005a). Tele-programming of Programmable Logic Controllers via Internet. *WSEAS Trans. on Advances in Engineering Education*, 2, Vol.2, pp. 118-126.
- Popescu, D., G. Terejanu, I. Dinulescu, M. Leoveanu (2005b). PLC based Telem Manipulation via Internet, *14th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, pp. 441-447.
- Zolotova, I., J. Ligus, A. Jadlovska (2006). Remote and Virtual Lab – CyberVirtLab. *Innovation in Education for Electrical and Information Engineering*, Ed. Universitaria, pp. 339-342.