# Development of online experiments for a mobile robot via Internet

DORIN POPESCU, DAN SELISTEANU,
IONUT DINULESCU, DANIELA LAZAR
Department of Mechatronics and Automation
University of Craiova
107 Decebal Blvd., Craiova
ROMANIA

*Abstract:* - This paper presents the work for the development of a didactic mobile robot for use via Internet. It was developed a client-server application. A range of remote experiments with a mobile robot with particular emphasis on motor control, obstacle avoidance and PID control can be done by the users. The users learn the microcontroller programming, sensors use, path planning, trajectory following. The idea is for the users to be able to perform real experiments, in real time, on real equipment, but over the Internet. The remote user is connected by Internet to a dedicated computer used for controlling and monitoring the mobile robot. The user can see through a web camera the mobile robot.

*Key-Words:* - mobile robot, telematics, microcontroller, path following, Internet.

## 1 Introduction

The high cost necessary to implement experiments (for didactical purposes) with manipulators, robots and mobile robots led to the development of the tele-operating facilities, such that the physical robots, sensors, and operating environment can be remotely controlled via the Internet.

Modern Multimedia and Telematics technologies offer good potential for distance education in order to present the theory and practical experiments in an interesting and economical way [1]–[3].

A tele-robotic operation involves interaction between a human operator and a remote robotic system via communication networks. The development of tele-operated systems has gained considerable attention in recent years due to new potential applications, such as remote production monitoring, remote exploration, manipulation in inhospitable environments, tele-surgery, tele-teaching, tele-working etc.

Some researchers achieved an analysis of the current challenges in Internet tele-programming and tried to find the solutions under the current environment [1]–[5], where there are some unsolved common problems associated with this new technology such as limited bandwidth and unreliable signaling.

Remote laboratories are laboratory experiments that run remotely via a web interface. Usually, the student either can set some parameters on the web, and 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 [6]–[10].

We proposed more, namely the user can achieve a program, which can run on the microcontroller in order to control a mobile robot. The student has thorough freedom to choose what experiment wants to do. The goal is the remote programming of a mobile robot for motor control, trajectory control, and obstacle avoidance.

Web-based tele-robotic systems have only become available on the Internet in the last decade. In the beginning it was utilized a Common Gateway Interface (CGI) to access the robots [4], [10]. Once with the Java development 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 in order to gave the user much more control and provided functionality, which under CGI could have never been possible [1], [3], [4], [11]–[15].

The Internet technology offers to 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 mobile robot. It tries to show an aspect of the 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.

The rest of the paper is organized as follows. In the second section we describe the mobile robot and

architecture of telematics application. Section 3 presents the telematics application (client-server application). In section 4 we describe how the user can work with our mobile robot and the features of our tele-education application. Finally, Section 5 presents conclusions and future directions.

## 2 The online mobile robot

There are two criteria that allow establishment of a clear classification of the experimentation environments from the point of view of student use: the nature of the system (virtual or real) and the way of accessing the resources for experimental purposes (local or remote access) [1]. Our experimentation environment represents a real system with remote access. It allows access to the real equipment through the Internet. The user operates and controls, in a remote way, a real system through Internet.
One essential aspect of robotics education is the practical work necessary to give students a taste of real situations. The idea is for the students to be able to perform real experiments, in real time, on real equipment, but over the Internet.

### 2.1 The mobile robot

The mobile robot is a didactical mobile robot for tele-operating and tele-programming via the Internet. A web camera enables the user to see the robot's movements in real-time while a client-server application is used to send programs and to change data, commands with the mobile robot. The commands, programs from the server to the microcontroller are transmitted through a serial connection, and for the control of the motors is used a BS2SX Basic Stamp microcontroller. The mobile robot has two driven wheels and one or two castor wheels. It is a differential-drive robot.

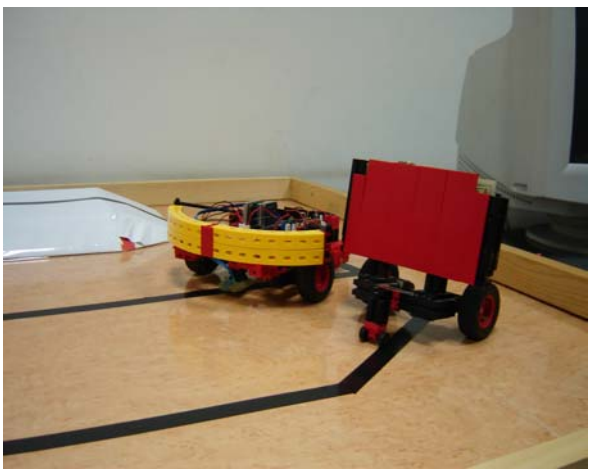

Fig. 1. The mobile robot (two variants).

The microcontroller we will use in this project is a special-purpose device designed to make life as simple as possible. The device is called a "BASIC Stamp" and is created by a company called Parallax. A BASIC Stamp is a PIC microcontroller that has been customized to understand the BASIC programming language. The use of the BASIC language makes it extremely easy to create software for the controller. The microcontroller chip can be purchased on a small carrier board that accepts a 9-volt battery, and the user can program it by plugging it into one of the ports on your desktop computer.
The mobile robot uses FischerTechnik blocks. The students build two variants of the mobile robot (Fig.1) taking into account the stability and flexibility required for the robot. We designed an electronic board representing the intelligence of the robot. Two infrared sensors and Hall sensors are the present sensors on the mobile robot, but more sensors can be added. An LCD display can be added too.
The control board structure (Fig. 2) has the microcontroller, two motors' controllers, power connector, serial connector, sensors' connectors etc.

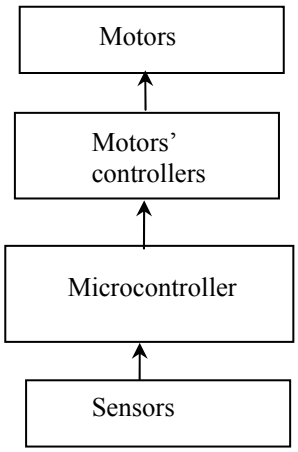


Fig. 2. The main components of the mobile robot.

The motors are controlled using a Motor Mind B motor controller from Solutions Cubed. The user' program send commands to the motor controllers and they will run the motors.
Teleoperation of the mobile robot really allows users to take the lab home, because the system is correctly built and it is possible to experiment from any computer connected to the Internet, on demand.
The remote user is connected by Internet to a dedicated web server on the computer used for controlling and monitoring the mobile robot. On the dedicated website the user can download the client application and learn about our infrastructure or he can use the e-mail in order to receive the client application.

### 2.2 Application architecture

The goal of this project in the remote programming area is to develop the system by combining web technology with capabilities of microcontrollers and mobile robots. 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 [1], [5].

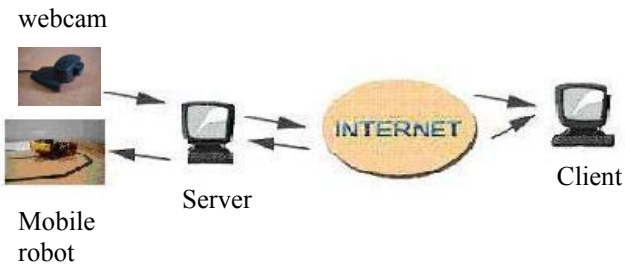


Fig. 3. The application architecture.

The telematics system is composed of mobile robot, USB webcam, computer for server application and website (Fig. 3).

## 3 Telematics implementation

The developed telematics application is a client-server application. This first version of our implementation uses an application client-server that it consisted of two programs: the server runs on the local computer (connected to the mobile robot) whereas the client runs on any Internet-linked computer.

Required main features of the client-server application were:

- Chat functionality
- Editor for BasicStamp (several features from the original editor) with syntax checker
- File transfer to the server (source code) and download to the microcontroller
- Video acquisition
- Online video transmission

### 3.1 Implementation of the server

The communication is based on the TCP-IP protocol; socket connection.

A message consists in a header (indicating the message type) and the message itself. When a message arrives, it is first analyzed, decomposed in main elements and specific actions follow. Based on the coding convention, the message type is identified and its body represents the sent data (useful part).

The codes are established by means of constants with specific meanings:

```
Public Const c_sSeparator          As String = ":-:"
Public Const c_sMsgServerStopped   As String = "ServerStopped"   ' informs the client that the server has stopped
Public Const c_sMsgClientName      As String = "MyName"          ' client name
Public Const c_sMsgDisconnect      As String = "PlsDisconnect"   ' disconnect request
Public Const c_sMsgSendFile        As String = "SendingFile"     ' intention of sending the file
Public Const c_sMsgBeginFile       As String = "BeginFile"       ' begin sending the file
Public Const c_sMsgEndFile         As String = "EndFile"         ' end file
```

Once server started, it begins listening for connection requests.

The application is of single-thread type, as only one client can connect to the server and operate on the robot.

The user interface of the server is presented in Fig. 4.

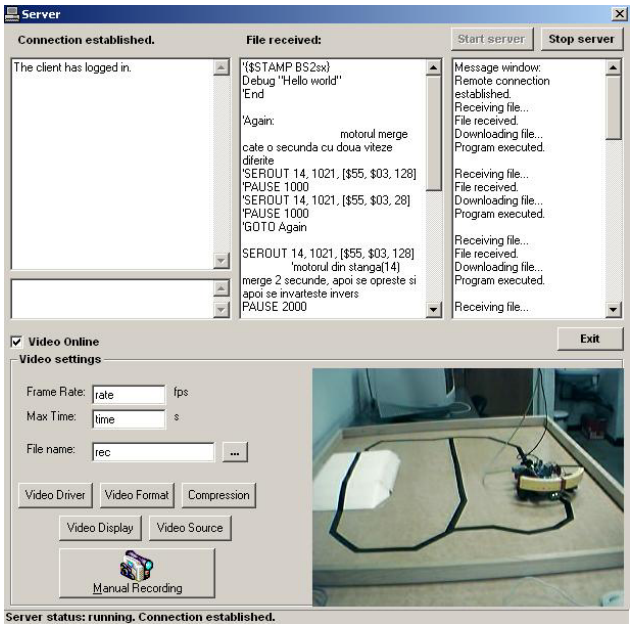


Fig. 4. The server interface.

### 3.2. Implementation of the client

The client application attempts to connect to the server; after the connection is established, the communication goes on in the same framework of exchange of messages (with the above mentioned codes).

An editor window allows writing the BasicStamp program. After the syntax check, the program may be sent and executed on the server.

The user interface is similar that of the server (Fig. 5).

All the functions for syntax check are grouped, in the spirit of OOP programming and having in view the further improvements, in a class named clsCheck.

The check is made analyzing line by line the file source code. Semicolons separate the instructions lines, so the split is realized taking in considerations this separator. The public method of the class is the *Check* method.

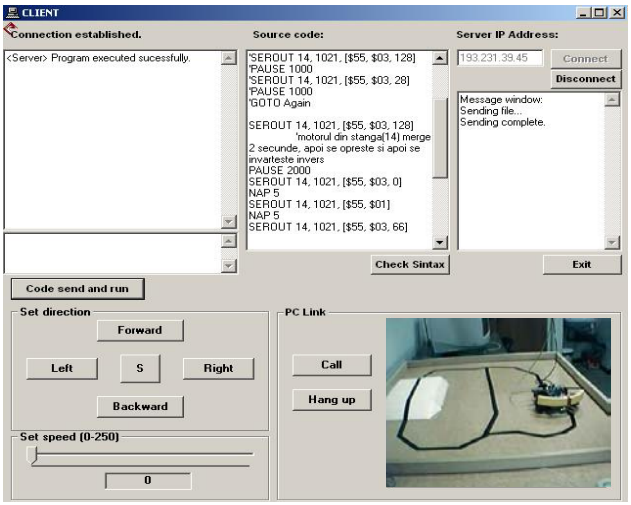


Fig. 5. The client (user) interface.

For the video feedback there were considered two options. The first was to send a live video feed (streaming) concurrently with the experiment. This test done by the authors proved that the existing connection did not allow always a reliable good-quality live video feed.

The second solution, chosen by the authors was that of recording the video to the server and sending the resulting file to the client when the experiment was over. In this way the received video quality did not depend on the variations of the Internet connection speed, and the client could analyze (by reviewing) the received video.

The user opened the client application and wrote the source code of the program. On the client application it was also implemented the compiler so the user could check the syntax of the program at any time. When he was ready he connected to the server and his connection was accepted (if there were no other users active). He could choose some parameters regarding the video frame rate and the duration of the experiment based on how fast the process was and how much the program had to run. Then he sent his program to the server.

On the server application, once the code was received, it was compiled into the machine code. If the compiler encountered any errors, an error message was sent to the client specifying the line number containing the error. If the code was compiled successfully then the execution was started.

Also the web camera started recording using the frame rate specified by the client and the video compressor selected by the server supervisor depending on the available ones. Once the time allocated (specified) for the experiment was over, the video recording is stopped and the video was sent to the client.

Another possibility of the user is to tele-operate the mobile robot. He has possibility to move the robot forward, backward, left and right, or to stop. He can also choose the speed (Fig. 5).

## 4 User guide

A user (student) might find himself in any location, with a computer and Internet. He will access the website of the laboratory where he/she will find how can work with our mobile robot. The user could do it in the following way: he could ask us by e-mail for the client application. He receives this client application and the period of time when he can work with the mobile robot. He has to read the documentation about mobile robot and microcontroller. At fixed time the user can begin to write his program in the programming language of the microcontroller. He could connect to the server and after acceptance he could send his user program. The user program for mobile robot runs and the user could observe the movements and operation of the mobile robot by means of a web-cam (Fig. 5). If there were delays the user could ask to record a movie (avi file) with the movements of the mobile robot and after a cycle of movements could download the movie and see it.

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 microcontroller in order to:

- control the motors of the mobile robot;
- control the trajectory of the mobile robot with obstacle avoidance.

After this, the user uploads his program and the results can be viewed on screen. The experiments have video streams available for visual monitoring.
The main requirement of this application is to provide a friendly user interface that allows application developers to control the devices from anywhere in the world via a thin client.
A motor can be set to run at a speed from 0 to 255.
Commands are sent to the motor controller using the SEROUT command. Here is the prototype:
SEROUT *motorpi*n, 1021, [$55, $03, *spee*d]
The *motorpin* value depends for right or left motor and the *speed* value is 0 to 255. The other numbers specify the serial port settings and motor controller protocol.
When a motor speed command is given, the motor will stay at that speed until the program sends a new command. If the user wants to stop the motor, he has to send a command with speed 0.
The direction can be changed using the SEROUT command with the same pins as the motor speed pins. The motors can also be braked. This is useful to stop the robot immediately. Braking is done using the LOW and HIGH commands.
The all telematics laboratory infrastructure is presented in Fig. 6.

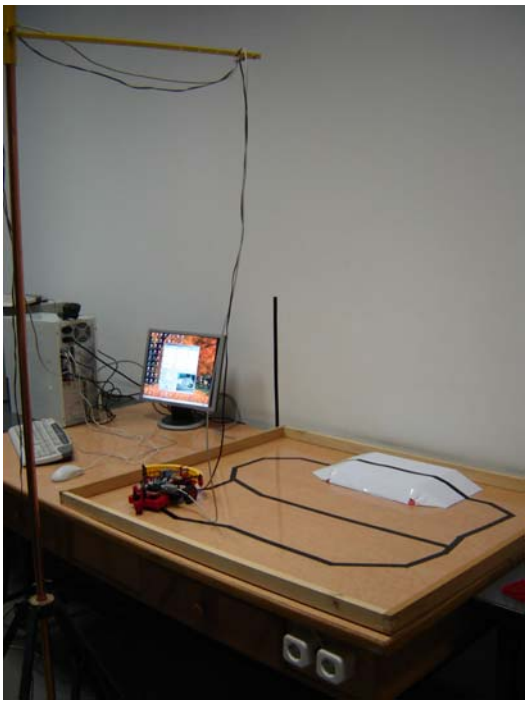

Fig. 6. The telematics laboratory infrastructure.

Many on-line robots allow tele-operating facilities. Our first goal was to offer more - tele-programming. But later we implemented the tele-operating, too. With a dedicated interface the student can move the robot through its environment. The user can choose the same speed for both motors of wheels or he can choose different speeds for them. The mobile robot can be moved forward, backward, left and right using the dedicated buttons.

## 5 Conclusions and future work

A user with a computer and Internet can work with our telematics application for a mobile robot. The user can write programs for the mobile robot in order to solve some problems such as: trajectory following, obstacle avoidance, path planning, and PID control (by tele-programming of the microcontroller).
Tele-education technology implements a distributed education environment independent of time, space, and individuals.
Some advantages of our mobile robot tele-education application are:

- advanced training system
- better presentation of complex material by using multimedia elements
- facilities for studying independently of location
- self learning for work
- appropriateness of the subject to remote operation
- tele-education allows continuing education with minimum costs
- tele-control of remote device and application
- testing robot with regard to the accuracy in positioning
- PID control
- remote surveillance (through webcam)

The disadvantage of client-server application is the need for user to have the client application. From this point of view it is more difficult to work with this type of application.
A requirement for telematics implementations 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 technologies that considerably reduced the implementation costs while accomplishing the other project requirements. Now, we begin to work to a web based application.
Thus, Java (for client) and Visual Basic 6 (for server) were chosen for the application development [16]–[19]. In order to speed up the development process and eliminate software bugs, additional Java open source libraries and Application Programming Interfaces (APIs) will also be used.

In this way the only software needed on the user side (for tele-programming of the microcontroller of the mobile robot) will be a conventional web browser that can run Java applets.

The application will be achieved using Java and Visual Basic and contains two modules: the client (Java) and the server (VB). The client application will be an applet hosted by website department. The server application will be hosted by a computer connected to the mobile robot via serial connection.

Some basic features of our experimentation environment are:

- access to our remote environment was carried out exclusively via the Internet. A web browser and our client application are the only necessary software tools that students need when undertaking a practical experiment;
- the web camera window allows users to see the movements of the mobile robot in almost real-time;
- our remote mobile robot promotes students' interest and motivation when they obtain real and coherent responses. The system reacts in almost real time to any action from the student;
- There is no time restriction

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

*References:*


[1] J. de Lafontaine, and C.A. Brunet (eds.), *Proceedings 2nd International Workshop on Tele-Education in Engineering Using Virtual Laboratories*, Sherbrooke, Canada, 2002.

[2] S. Dormido (ed.), Internet Based Control Education, *Proceedings of the IFAC Workshop on Internet Based Control Education*, Madrid, 2001.

[3] K. Schilling, and H. Roth (eds.), *Telematics Applications in Automation and Robotics*, Pergamon Press, 2001.

[4] P. Backes, K Tso, J. Norris, G. Tharp, J. Slostad, R. Bonitz and K. All, "Internet based operations for the Mars polar lander mission", in *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, 2000, pp. 2025-2032.

[5] G. Belmonte Bermudez and M.A. Perez Sanchez, "Robots Tele-programming", in *1st Workshop on Robotics Education and Training, RET 2001*, Weingarten, Germany, 2001, pp. 19-24.

[6] R. Berntzen, J.O. Strandman, T. Fjeldly and M. Shur, "Advanced Solutions for Performing Real Experiments over the Internet", in *Int. Conference on Engineering Education*, Oslo, 2001, pp. 21-26.

[7] H. Ewald and G.F. Page, "Performing Experiments by Remote Control Using the Internet", in *Global J. of Engineering Education*, Vol. 4, No. 3, 2000.

[8] D. Popescu and K. Schilling, "TOM – A Remote Laboratory for Mobile Robots Education", in *Proc. 7th International Conference on Intelligent Engineering Systems*, Assiut, Egipt, 2003, pp. 447-453.

[9] S. Grange, T. Fong and C. Baur, "Effective vehicle teleoperation on the world wide web", in *Proc. IEEE International Conference on Robotics & Automation*, 2000, pp. 2007-2012.

[10] K. Schilling, "Virtual Laboratories for Mobile Robots", in *Proceedings IFAC Workshop on Internet Based Control Education*, Madrid, 2001, pp. 115-120.

[11] S. Amin, M. Zakaria, N. Majid, L. Siong, L. Horvath and J. Tar, "Internet-based telerobotics: UTM's experience and future direction", in *Proc. The 10th International Conference on Advanced Robotics, ICAR 2001*, Budapest, Hungary, 2001, pp. 313-319.

[12] D. Popescu, I. Dinulescu, L. Popescu, "Tele-programming of Programmable Logic Controllers via Internet", in *WSEAS Trans. on Advances in Engineering Education*, 2, Vol.2, 2005, pp. 118-126.

[13] I. Dinulescu, D. Popescu, G. Terejanu, "Web based Telematics Application using Open-Source technologies", in *Proc. 12th Int. Symposium on Systems Theory, SINTES12*, Craiova, 2005, pp. 267-272.

[14] R. Marin and P. Sanz, "The UJI Telerobotic Training System", in *1st Workshop on Robotics Education and Training*, RET 2001, Weingarten, Germany, 2001, pp. 25-30.

[15] D. Popescu, G. Terejanu, I. Dinulescu, M. Leoveanu, "PLC based Telemanipulation via Internet", in *Proc. 14th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, Bucharest, 2005, pp.441-447.

[16] D. Flanagan, *Java Script – The Definitive Guide*, O'Reilly, 2001.

[17] M. Hall, *Core Servlets and Java Server Pages*, Prentice Hall, 2000.

[18] B. McLaughlin, *Java and XML*, O'Reilly, 2000.

[19] Microsoft Visual Basic 6.0 – Programmer's Guide, Ed. Microsoft Press.