Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2008, vol. LIV,
article No. 1613

Ján BABJAK[*], Tomáš KOT[**]

DESIGN OF A CONTROL MODULE FOR MOBILE ROBOTS

NÁVRH ŘÍDICÍHO MODULU PRO MOBILNÍ ROBOTY

Abstract

This contribution describes a module serving as a complete control system for locomotory and sensory subsystems of small mobile service robots. This module is still in development on the Department of robotics at VŠB-TU Ostrava, now it is in the phase of practical testing on various wheeled mobile robots, which reveals new problems to be dealt with by modifying existing features or adding new ones. The control system consists of all necessary components; both the low level and higher level control systems and their mutual communication.

The low level control system includes all hardware and a program in a microprocessor necessary for servos or electromotors control and data acquisition from sensors. Communication with the higher level control system is made via wireless RS232. The higher level control system is programmed in Visual Basic .NET and its function is to get control inputs from user, process it and send to the lower level system and also to provide the user with all necessary informations from the robot. In the future it can also do some artificial intelligence calculations.

Abstrakt

Příspěvek popisuje modul soužící jako kompletní řídicí systém lokomočního a senzorického ústrojí malých servisních mobilních robotů. Tento modul je stále ještě ve vývoji na Katedře robototechniky na VŠB-TU Ostrava, nyní je ve fázi praktického testování na rozličných kolových mobilních robotech. Tím jsou zjišťovány problémy, které se následně odstraňují modifikacemi nebo přidáváním nových funkcí. Řídicí systém se skládá ze všech nezbytných komponentů – pro řízení vyšší a nižší úrovně a jejich vzájemnou komunikaci.

Řízení nižší úrovně obsahuje hardware a program v mikropočítači nezbytné pro řízení elektromotorů a serv a pro získávání dat ze senzorů. Komunikace s řízením vyšší úrovně je zajištěna prostřednictvím bezdrátového RS232. Řídicí systém vyšší úrovně je naprogramován ve Visual Basicu .NET a jeho funkce spočívá v získávání vstupů od uživatele, jejich zpracování a posílání do řízení nižší úrovně. Dále poskytuje uživateli zpětnou vazbu o současném stavu robota. V budoucnu může být aplikace použita i pro aplikaci umělé inteligence a podobně.

1 INTRODUCTION

Research on the Department of robotics at VŠB-TU Ostrava is focused on mobile robots. One tracked and few wheeled mobile robots were already developed and also physically built, which brought the need to make a universal control system.

Some of the mobile robots previously had RC-based remote control system deigned for manual control by an operator. This principle is unsuitable for controlling using a PC with SW

[*] Ing., Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University, 17. listopadu 15, Ostrava, tel. (+420) 597 325 757, e-mail jan@babjak.cz
[**] Ing., Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University, 17. listopadu 15, Ostrava, tel. (+420) 597 325 757, e-mail tomas.kot@vsb.cz

control algorithms. Thus it was necessary to develop a new control system allowing control from a PC.

The control system should be able to be used on various mobile robot undercarriages without large modifications, to gain more time for the main goal of the research. The designed control system consists of two parts – a higher level control system dealing with user inputs or robot's artificial intelligence and a lower level control system working with drives and sensors.

2 HIGHER LEVEL CONTROL SYSTEM

The higher level control system has been programmed in the Microsoft Visual Basic .NET environment and runs on a personal computer or a notebook. Its task is to react on user's inputs, process them and create corresponding control codes. These codes are then sent to the lower level control system on the mobile robot. It also provides all necessary visual and text informations about the current state of the robot.

The control system is being developed with emphasis on its universality, which means the ability of controlling various mobile robots without the need of source code modifications. The current version of the system can be used (without any hardware or software alterations) to simultaneously control three drives, five binary outputs and read values of four analog or binary inputs (sensors). Another aspect of the application is modularity.

Communication with the lower level control system located on the mobile robot is handled by our own purpose-built .NET Framework component called RoboControl, compiled to a dynamic-link library. This component can be placed on an application form in the Visual Studio editor and then its properties, methods and events can be used easily.

Besides this fundamental component, the control system uses also a class enabling the application to retrieve data from input devices and also a class for network communication. Both these classes are also compiled to separate dynamic-link libraries and have general use.

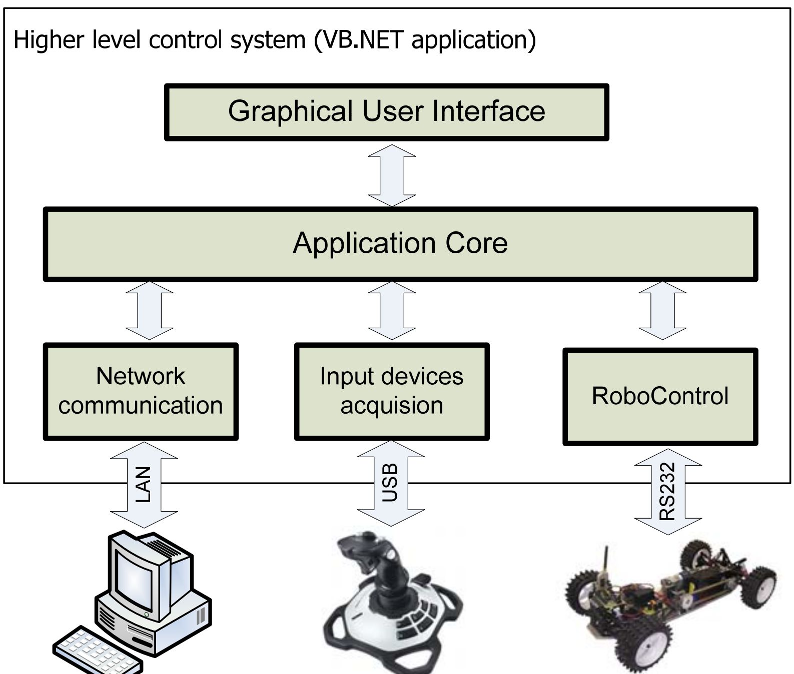

Fig. 1 Block diagram of the higher level control system components

The higher level control system application allows the robot to be operated by any common gaming input device (joystick, gamepad, wheel etc.) compatible with Microsoft DirectX. All the code required to access the DirectInput API (part of the DirectX library), that means to query for available devices, connect to one, set necessary properties and then continuously read its data, was programmed in Visual C++ at the lowest possible level and compiled as a set of functions and structures to a DLL library, to achieve maximal compatibility (this library can be used for example also in MATLAB). For easier utilization in Visual Basic, the content of this library was encapsulated to a class, which also further extends the functionality. Thus the input device can be handled simply by setting various properties (axes filters and dead zones etc.), reading outputs (device name and capabilities etc.), calling methods and responding to events (buttons pressed, axis value changed etc.) of an instance of this class.

The library contains also functions required to work with the Force Feedback effects of gaming devices that supports them. These force and vibration effects can be used in the future versions of the control system to provide a physical feedback from the robot, allowing the user to feel for example strengthening joystick vibrations when the robot is approaching an obstacle.

The application can run in three network modes:

- as a stand-alone application,
- as a server application,
- as a client application.

When the application is in the stand-alone mode, it performs all the necessary tasks – getting inputs form a user, sending control codes to the mobile robot via RS232, reading values of sensors sent by the robot (also via RS232) and displaying corresponding informations on the screen. The RS232 communication is made as wireless.

In the server application mode, user is not working directly with this application. The application gets inputs from a client application via network communication using UDP protocol and the same way it returns values of sensors located on the robot. This mode is suitable in cases when the robot is equipped with a small PC (embedded PC) and the server application is running on it. The server application communicates with the lower level control system via wired RS232, while the network communication between the server and client is wireless, using any common technology for wireless network transmission (for example wi-fi).

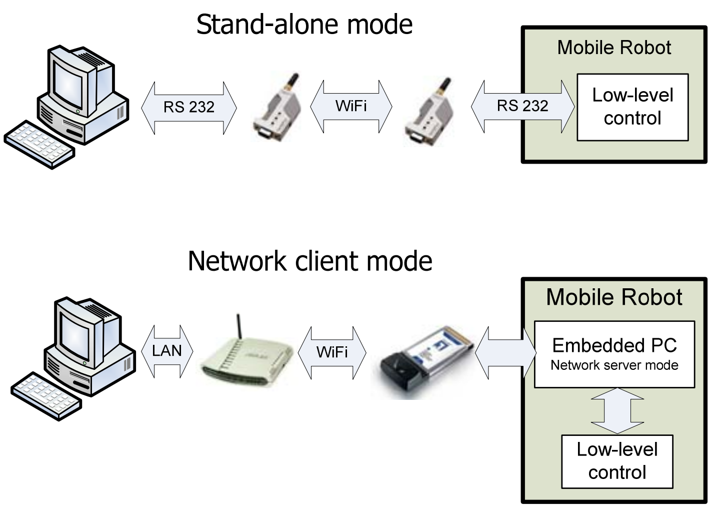

Fig. 2 Schematic representation of relationships between the application modes

The third mode is network client running on user's computer and communicating with the server application described above.

Graphical user interface of the application is being developed together with the functionality of the whole control system. The current state is shown on Figure 3.

It shows the chosen application mode, informations about active input device (name, axes values, buttons), serial port status, visual representation of the robot undercarriage, battery condition and also feedback from one IR proximity sensor in the form of a distance in cm. Further it offers controls for input device axes and buttons assignment etc. Many of these informations are displayed mainly for testing and debugging purposes.

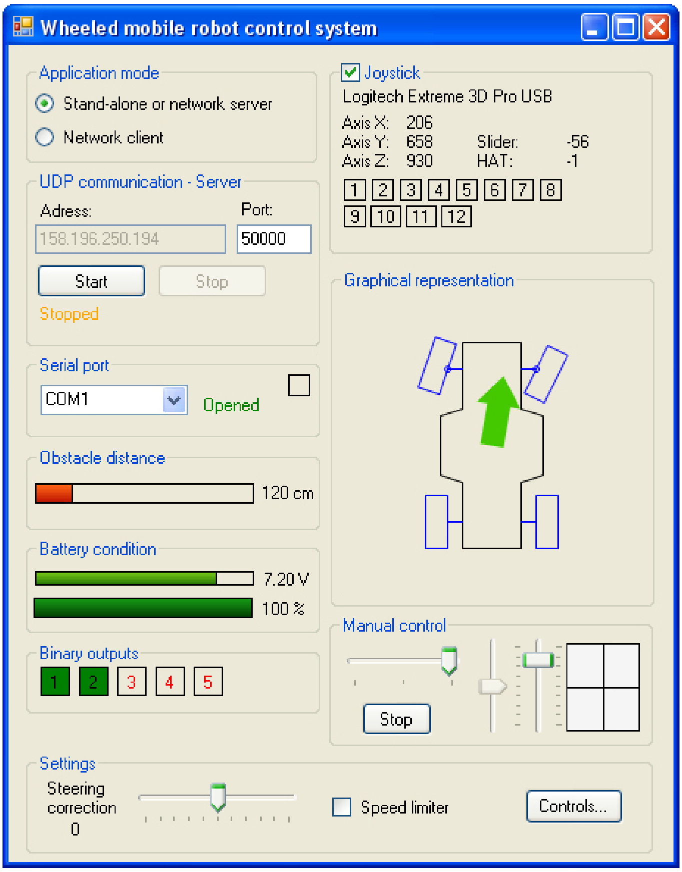

Fig. 3 Graphical user interface of the control application

RoboControl component is a non-visual control (it is added to the component tray in Visual Studio form editor). Its properties can be set both in the design environment and during run-time. These properties include serial port settings (port name, baudrate) and values specifying limits for individual servomotors.

During the control process itself, the control applications is passing the RoboControl component just desired values for individual drives in general range of -100% to +100% and desired states of binary outputs (on/off) and is reading current values of analog inputs of the lower level control system (sensors). This way, an abstraction is achieved, because for the control application it is not important what kinds of sensors and drives are used on the mobile robot. The RoboControl component arranges translation between the abstract (general) values and specific values for the particular device (based on set ranges) and also transmission via RS232.

3 LOW LEVEL CONTROL SYSTEM (ROBOCONTROL MODULE)

The low level control system is based on the microprocessor ATMEL AT Mega8 [4]. Its task is to receive commands from the high level control system (personal computer), process them and hand-over to locomotion subsystems. Another task of this control system is data acquisition from sensor subsystems and their sending forth to the high level control system. Communication between the RoboControl module and the high level control system is realized by a common serial line (RS232), a pair of wireless connection devices HandyPort is used for wireless communication.

The task mentioned previously is solved by a processor board (RoboControl module) which was developed and made especially for this purpose. For design of a printed circuit board, CAD system EAGLE (free version) was used. The parts of this board are: processor ATMEL, device used for digital outputs switching and device providing communication with high level control system.

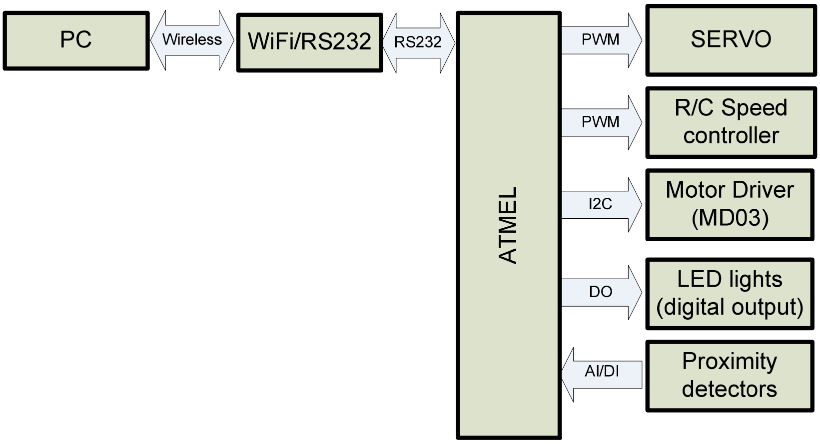

Fig. 1 Block layout of the RoboControl module

This processor module is able to control up to two servos or two model motor speed controllers or any combinations of these devices. In our concrete case this module controls one servo for mobile robot steering and one speed regulator for control of forward and backward robot movement. For controlling of high powered motors, this module is equipped with I2C communication bus, which can be used to connect a powerful controller – for example the module MD03 which is able to control motors of up to 100 W. The maximum number of these controllers connected to a single I2C bus is eight – so it is possible to realize a variety of mobile applications.

The module disposes of five digital outputs with powered switches. We can use these outputs for example to control high-power LED diodes for lighting of the robot's surroundings. This module also acquires data from up to four analog signals. One of these analog signals is used for monitoring of battery condition and the remaining can be used for proximity detectors (we use the SHARP infrared modules). The inputs can nevertheless be reconfigured to work with digital signals.

Robust and secure software elements were implemented to the control module. Communication with the high level control system uses error detection algorithm to check the incoming commands for potential errors occurred during transmission to prevent unforeseeable behaviour if an error occurs. Another precautionary feature is a watchdog function which is able to detect potential signal lost (this is also indicated by a red LED diode) and to turn the mobile robot to a safety stand-by state.

The module also features a connector for in-build processor programming (ISP - In System Programming interface) which enables debugging and testing of the RoboControl module control application. The application has been programmed using a free version of the development environment BASCOM AVR [3]. Powering of the module is realized by a BEC interface provided by the installed model speed controller. We can also power this module using an external 5V power source.

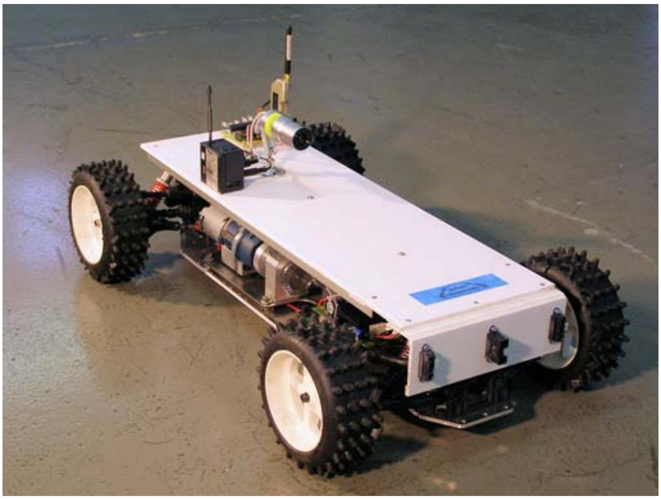

Fig. 5 A mobile robot with the RoboControl module

4 CONCLUSION

The designed control module can be used to control almost any type of small mobile robots. Due to the software interface the control logic is separated and independent on the used undercarriage hardware. Using any of the Microsoft Visual Studio programming languages with the developed control component it is possible to easily control reconnaissance robots created on the department.

All the components of the control system are being tested nowadays, to verify their functionality and compatibility. This shows new problems, which are then solved by modifications or by adding new features. The system is intended to be used on all mobile robots designed on the department in the future, first mainly for research purposes and in the end even for practical application.

5 REFERENCES

[1] NOVÁK, P.: *Mobilní roboty – pohony, senzory, řízení*. Praha : Nakladatelství BEN – technická literatura, 2005. 250s. ISBN 80-7300-141-1.

[2] NOVÁK, P.: *Mikropočítačové řídicí systémy*. 1. vyd. Ostrava : VŠB-TUO, 2002. ISBN 80-248-0291-8.

[3] VÁŇA, V.: *Mikrokontroléry Atmel AVR – Programování v jazyce Bascom. Praha :* BEN, 2004. 144 s. ISBN 80-7300-115-2.

[4] MATOUŠEK, D.: *Práce s mikrokontroléry ATMEL AVR*. Praha : BEN, 2003. 375 s. ISBN 80-7300-088-1

Reviewers:

doc. Ing. Jaroslav Šeminský, PhD., Technical University of Košice

dr inż. Janusz Pluta, AGH University of Science and Technology