Sborník vědeckých prací Vysoké školy báňské - Technické univerzity Ostrava
číslo 1, rok 2008, ročník LIV, řada strojní
článek č. 1576

Jan BABJAK[*], Tomáš KOT[**], Petr NOVÁK[***], Vladimír MOSTÝN[****]

CONTROL MODULE FOR SMALL MOBILE ROBOTS
ŘÍDÍCÍ MODUL PRO MALÉ MOBILNÍ ROBOTY

Abstract

The article 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

Článek popisuje modul soužící jako kompletní řidicí 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. 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.

[*] Ing. BABJAK Ján, Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University of Ostrava, 17.listopadu 15, Ostrava-Poruba, tel. (+420) 59 6995757, jan@babjak.cz
[**] Ing. KOT Tomáš, Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University of Ostrava, 17.listopadu 15, Ostrava-Poruba, tel. (+420) 59 6995757, tomas@kot.cz
[***] doc. Dr. Ing. NOVÁK Petr, Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University of Ostrava, 17.listopadu 15, Ostrava-Poruba, tel. (+420) 59 6993595, petr.novak@vsb.cz
[****] prof. Dr. Ing. Mostýn Vladimír , Department of Robotics, Faculty of Mechanical Engineering, VŠB-Technical University of Ostrava, 17.listopadu 15, Ostrava-Poruba, tel. (+420) 59 6994257, vladimir.mostyn@vsb.cz

2 LOWER LEVEL CONTROL SYSTEM

Low level control system is based upon microprocessor ATMEL AT Mega8. Its task is receiving commands from high level control system (personal computer), their processing and handing-over to locomotion subsystems. Next task of this control system is data acquisition from sensor subsystems and their handing-over back to high level control system. Communication between RoboControl module and high level control system is realized by common serial line (RS232). For wireless communication is used pair of wireless connection devices HandyPort.

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

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 motion. For high powered motors control, this module is equipped with I2C communication bus, to which we can connect powerful module, for example module MD03 which is able to control motors of up to 100W. Maximum number of these modules connected on I2C bus is eight which allows realizing many types of mobile applications.

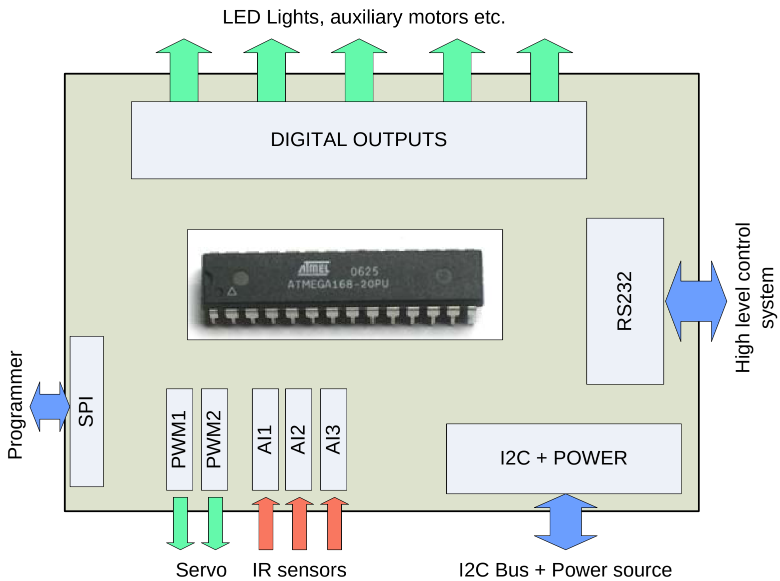

Fig. 1 Block layout of the RoboControl module

The module disposes of five digital outputs with powered switches. We can use these outputs for control of high-power LED diodes for lighting robot surroundings for example. This module also responds for data acquisitions from up to four analog signals. One of these analog signals is used for monitoring of battery condition and to other analog signals we can connect proximity detectors (we use SHARP infrared modules). After reprogramming firmware of this RoboControl module we can also use these three inputs for acquiring of digital signals.

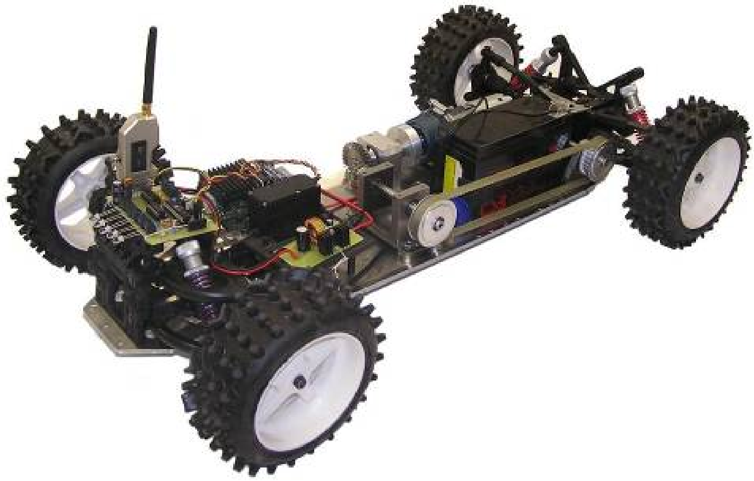

Fig. 2 RoboControl module on a mobile robot

Robust secure software elements were implemented to control module. Communication with high level control system uses error detection algorithm for checking incoming commands for eventually errors incurred during transmission to suppress unforeseeable behavior if error in command occurs. Next secure feature is watchdog function which is able to detect potential signal lost (this is indicated by changing information LED diode from green to red color) and to turn the mobile robot to a secure stand-by state.

Next feature of this module is connector for in-build processor programming (ISP - In System Programming interface) which enables debugging and testing control application of RoboControl module. This application is programmed at free version of development environment BASCOM AVR. Powering of module is realized by BEC interface which provides installed model speed controller. We can also power this module using external 5V power source.

3 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.

Structure of the control system

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.

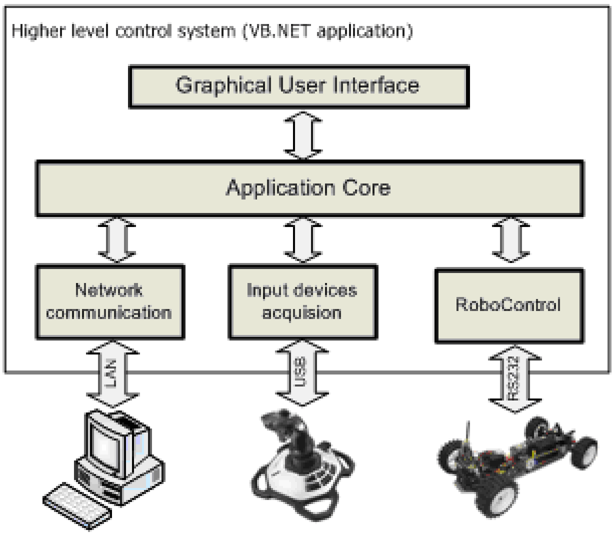

Fig. 3 Block diagram of the higher level control system components

Input devices

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.

Network communication

The application can run in three 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).

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

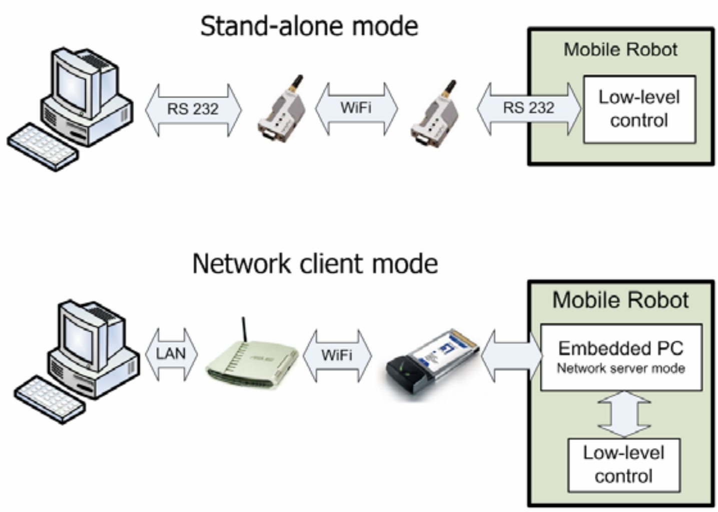

Fig. 4 Schematic representation of relationships between the application modes

Graphical user interface

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

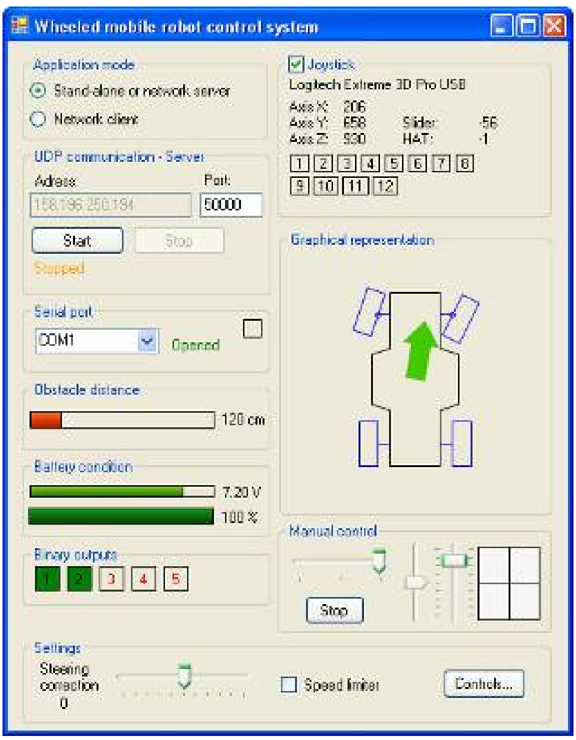

Fig. 5 Graphical user interface of the control application

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.

Communication with the lower level control system

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.

4 CONCLUSION

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.

This article was compiled as part of projects FT-TA3/014, supported by the Fund for University Development from the Ministry of Industry and Trade.

5 REFERENCES

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

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

[3] BABJAK, J.: *Řízení mobilního robotu neuronovou sítí*, (Diploma thesis), Ostrava: katedra robototechniky, VŠB-TU Ostrava, 2006, 76 s.

[4] ATMEL: Procesor ATMega8 – Datasheet. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf

[5] MAXON: On-line catalogue A-Max. Available: http://www.maxonmotor.com/A-max.asp

[6] ST MICROELECTRONICS: L298 – Dual Full-Bridge Driver – Datasheet. Available: http://www.st.com/stonline/products/literature/ds/1773.pdf

Rewiewer:
ing. Roman Dudek, ESP, Insisa, a.s., Ostrava
ing. Marcel Boščík, ABB, Robotics Division, Ostrava-Hrabová