Mechanics and Mechanical Engineering
Vol. 13, No. 2 (2009) 33–54
© Technical University of Lodz

System Linux in Steering of Robots for Detecting Concentrations of Gasses

Józef Giergiel and Dariusz Szybicki
Department of Applied Mechanics and Robotics,
Rzeszow University of Technology

Received (1 August 2009)
Revised (25 August 2009)
Accepted (8 November 2009)

As part of the work was built on Linux system, operating in real time, graphical user interface AmigoBot mobile robot control. The application written in C++ language has the ability to control important parameters of wireless robot, which allows easy and intuitive movement and enables the construction of graphical environment maps. At this work was also carried out the solution of equipment in the electronic work AmigoBot measurement module of harmful gasses, controlled by the robot control application. AmigoBot powered by the measurement module acquires the characteristics of the inspection robot. Is able to reach the destination because of the high levels of dangerous gasses, inaccessible and dangerous for humans.

Keywords: Linux, robots, graphical interface

1. Introduction

Mobile robots are independent devices performing the given movement without the outside support. They are usually equipped with sets of sensors, the manipulator allows them to perform the task in question, and many other devices allowing the state set the environment [1]. These by design should be autonomous devices i.e. such like, which almost nothing is limited for example, control or power cables. Wheeled mobile robots can be used to implement an autonomous guided transport in the field of industry and economic life. Often used in surveillance systems, to penetrate and identify unknown dangerous area. Ever wider field of applications of such robots become the task of reviewing and analyzing the environment, robots reach wherever human action is not possible or significantly impeded.

Equipped with mobile robots in sets of sensors and the sensors are able to perform roles such as inspection and provide e.g. a mobile platform for measuring gas concentrations as it is the solution presented in this work.

For controlling mobile robots are increasingly often used graphical interfaces that control the vestments of work and allow for easy management functions of the robot.

Construction and operation of control applications require advanced development tools, stability, reliability and work in real time. All of these meet the expectations of the Linux operating system, which is an open programming environment and activities, having a lot of unique solutions of commercial advantages.

2. Examples of robots running under control of the Linux and real-time systems

- **Robot Hallu II.** This robot is controlled by single chip computer Geode with the installed there system Linux.

Figure 1 Robot Hallu II with the control cabin [19]

It has a very complicated drive system. Just to mention that in each of his eight legs installed seven engines, giving a total of huge, for this type of work, the number of degrees of freedom – 56, each equipped with an optical encoder or potentiometric sensor bending angle of the pond. A special cabin is used for steering (Fig. 1), which is large enough to fit a human inside. On the half–spherical screen with a diameter of 150 cm is displayed image from the robot camera, and two complicated manipulators serve are use for steering. Manipulators are additionally equipped with force–feedback system that resists, in situations where certain movements are prohibited - for example when the robot is close to the edge of his labyrinth, it isn't possible to steer him so that came up on the wall and damaged [19].

- **NASA Mars Landers.** Another interesting example of the application, is to use the real time VxWorks operating system like Linux, belonging to the family of Unix systems in the Pathfinder mission.

Figure 2 Mars Landers [20]

It was a NASA space probe designed to examinations of Mars. The probe consisted of two parts: the Lander and sixwheeled vehicle named Sojourner. This was the first in the history of Mars exploration vehicle, controlled from the earth moving along the surface of another planet. Pathfinder sent 16 thousand pictures from the Lander and 550 images from the vehicle.

After the success of the Pathfinder project (Fig. 2, on the left), real–time operating system VxWorks again found himself on board Landers MER (Mars Exploration Rovers Fig. 2, to the right) [20].

- **Humanoid robot companies Fujitsu's HOAP.** It is a humanoid, measuring half a meter height and weighing 7 kg, the robot control system running RTLinux [21].

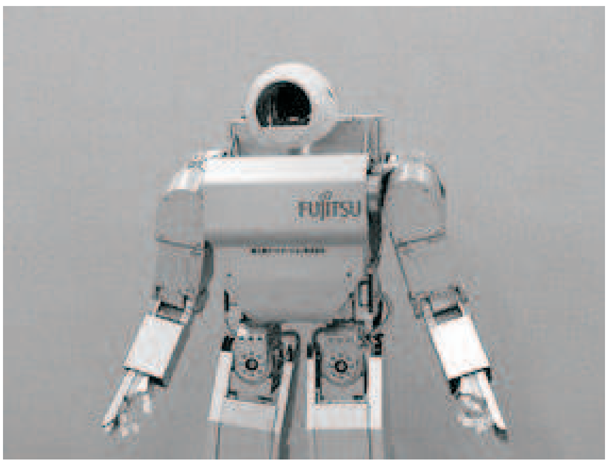

Figure 3 HOAP Robot [21]

The heart of a robot HOAP–2 is a processor Pentium III 700 MHz. Name of the robot: HOAP stands from Humanoid Open Architecture Platform. The device is equipped with a USB port, through which exchanges data with the computer. Optionally, you can choose a machine with wireless control, carried out by means of WLAN 802.11b. Two–handed and two–legged robot (Fig. 3) can move his head, hips and hands.

3. Linux system as a standard for robotics

The Linux system is widely used in robotics, also in steering, control, software development and simulation of all robots types. This is linked with similar roots and history of the development both robotic and Linux. Ob River of field are a field of experiments, the domain of enthusiasts seeking to optimize trying new solutions and seeking to improve and modernize the subject of its interest. The first robots were considered a kind of extravagance of its creators, a strange mechanism with few practical uses. Similarly was with the system Linux, it was written by a group of enthusiasts who want to develop their skills, improve existing solutions and make them available to all concerned. Time showed that both of these areas have developed in a way that simply amazing astonishing speed up expansion of its creators. Robots have gradually become more common, occur in virtually every field of industry and enter into more and more services and entertainment segment. Similarly, the system for Linux enthusiasts has become a powerful, extensive, offering ample opportunities and a system designed for home users, supercomputers, mobile solutions and industry.

The reasons why Linux may be defined as a standard and a system that can dominate the robotics solutions:

- already submitted a short common roots and history;
- associated with an earlier point of the same origin and development environment;
- open–source Linux operating system;
- the availability, or incurred low cost and legality of the scheme;
- high stability and reliability of Linux;
- functionality, the number of available software and libraries;
- the possibility of Linux as a real-time system.

All these mentioned features make Linux is one of the most widely used systems in all kinds of issues in robotics, control, with inspection of devices and measuring apparatus. Rapid advances in robotics and the growing role of software and ever more certain and unwavering position of Linux allow say that it remains the standard robotics and both these fields will be developing faster and faster.

4. Real–time systems and robots

Control and monitoring of the parameters of specific objects such as robots, both mobile and manipulators will require some specific features of the control system such as speed of response and precision. In the robots due to the presence of movable segments, the sometimes rapid shifts controlled, slow reaction control system could result in the wrong position or even referred to a loss of control over the object. In moving robots a very important issue is the fact that for them it is necessary to constantly control in numerical integration defining the position of the entire robot or its elements [2]. When the moving object stretches of time between the measurements of the speed of the wheels are too long, increasing the degree of inaccuracy of numerical integration, which causes the big differences between the actual location of the robot, and its position taken into account the control algorithm. This means that for this type of robots that between one procedure call and the next steering the robot can travel a long way too, and for example, hit a barrier. Way to ensure fast, accurate, high quality control of robots is to use real–time systems, both types hard, soft, and having the necessary properties of solutions based on Linux.

By advanced essential steering unnecessary becomes support complex mathematical functions such as PID algorithms or fast Fourier transform, but to have the proper libraries, which the vast majority of the RTOS (real–time operating systems) doesn't hold [22]. The solution for this problem may be or either time-consuming and require expertise to create their own algorithms or use the library systems and open source. In such an area open up opportunities for Linux and other free software, because most of the problems and the necessary algorithms have already been developed and tested. Only needed to be find, modify and use, for which usually allow the license and the authors.

Real time systems capable of delivering high quality and unchanging, both the control and processing of measurement results. These systems should have fixed response time to stop and set a maximum tolerable delay in the implementation of subsequent requests, this value should be the order of several to tens of microseconds.

Even Linux isn't and has never been a hard real–time system [18]. It was designed in such a way that the average performance of each application was as best. The standard Linux kernel doesn't have the possibility of real–time, available in hard–type systems, such as the determined response time and the suspension microsecond [10]. The reason of this state is that the Linux operating system has the architecture of the universal destination. It was built in such a way as to maintain high functionality and reliability of Unix, further development improving network technologies and the way of using [12, 13].

In order to make real–time Linux system requires the removal of the unnecessary items, such as unnecessary modules of the nucleus, programs, files. It is possible to change the kernel–raising the response time and causing that the system is picking up of the real–time. For this type of modification can include, among others:

- change of the frequency of task-switching in the system;
- modifying the processes of dispossession;
- the use of mechanisms standard POSIX 1.b, 1.c.

Until recently, in order to reduce time delays in the Linux operating system, so to make it a real–time system, it was necessary to install patches low delays. Currently kernels in the 2.6.10 version and newer offer a very good performance and allow to get low latency without any additional activities [18]. Also include a compilation of available options so far only by the installation of patches.

5. Mobile robot to measure the concentrations

An example of a mobile robot which can explore a potentially dangerous environment and transmit the measurement results is presented in this work AmigoBot. Thanks to equipping it with the electronic measuring module has gained a unique set of sensors already possess complementary sensors and enabling to better analyze the environment in which to move.

Graphical user interface operational control and written in a Linux environment allows to remotely control the important functions of the robot and receive feedback on the state of the environment of his work.

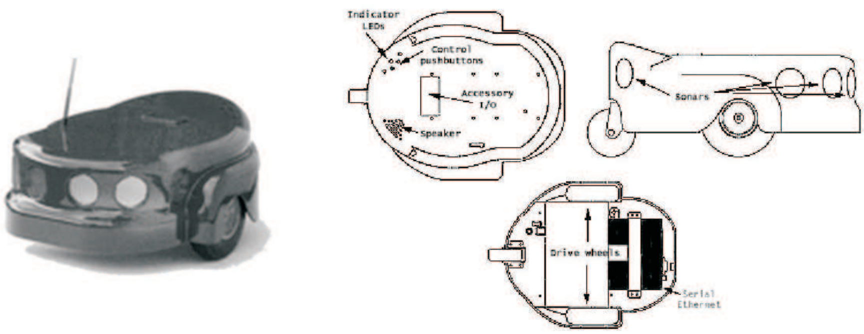

Figure 4 AmigoBot [23]

Despite its inconspicuous appearance of the device can be described as a fully functional, three–wheel, mobile robot. It is equipped with a whole set of sensors and control software Amigos, making the intelligent machine capable of moving in the indeterminate environment, record environmental changes and respond to them appropriately.

Robot is able to because all the sensors, safely move from one place to another without the aid of the operator, while avoiding any obstacles and precisely at the same time realizing their position. AmigoBot is designed for indoor–type homes, schools, hospitals, offices and research laboratories. Currently, the widest field of application of these robots is to teach, they are used as an educational platform for mobile robotics at universities and colleges around the world.

The dynamics of the robot

Description of the dynamics of the actual construction starts from a physical model and mathematical model describing the phenomena of interest to us. Mathematical model can be used for various purposes, such as analysis and synthesis, modification, control, identification, optimization, etc. In modeling the dynamics of wheeled mobile robots are often used in the Lagrange equations of the second kind of multipliers based on them or Maggie equation. Dynamic equations of motion of wheeled mobile robots can be used to solve the simple task and inverse dynamics. In setting straights dynamics it is possible to appoint the parameters associated with the move, while the opposite may be appointed task forces and moments acting on the robot.

In describing the dynamics of wheeled mobile robots can use different mathematical formalisms. One of these equations are the Lagrange multipliers of the second kind for notholonomic systems [2, 3, 4, 5, 6, 15, 17]:

$$\frac{d}{dt}\left(\frac{\partial E}{\partial \dot{q}}\right)^T - \left(\frac{\partial E}{\partial q}\right)^T = Q + J^T\left(q\right)\lambda \tag{1}$$

where:
q – vector of generalized coordinates
$E = E(q, \dot{q})$ – kinetic energy of the system
Q – vector of generalized forces
$J(q)$ – Jacobian equations arising from bonds
λ – vector of Lagrange multipliers

Received dynamics equations can be difficult to resolve, for this purpose should apply the certain transformation allowing e.g. for decoupling multipliers from driving moments appearing in these equations. To do this, write the equation (1) in the form of a matrix [2, 4, 5, 15]:

$$M(q)\ddot{q} + C(q,\dot{q})q = B(q)\tau + J^T(q)\lambda \tag{2}$$

where:
M – inertia matrix
C – matrix of centrifugal forces and Coriolis
B – matrix coefficients of forces and moments
τ – vector forces and moments

Next upsetting the vector of coordinates generalized to the form:

$$q = [q_1,\, q_2]^T \quad q \in R^n, \quad q_1 \in R^m, \quad q_2 \in R^{n-m} \tag{3}$$

It is possible then to write leveling bonds as follows:

$$[J_1(q), J_2(q)]\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} = 0, \quad \det J_1(q) \neq 0 \tag{4}$$

Vector q_2 should be chosen in such a way that its size corresponded to the quantity of degrees of freedom:

$$\dot{q} = \begin{bmatrix} J_{12}(q)\\ I_{n-m}\end{bmatrix}\dot{q}_2 = T(q)\dot{q}_2 \tag{5}$$

$$\ddot{q} = \dot{T}(q)\dot{q}_2 + T(q)\ddot{q}_2$$

where:
$J_{12} = -J_1^{-1}(q)J_2(q)$
I_{n-m} – matrix unit
After completing this operation received decoupling dynamic equations of motion in the form of [2, 3, 5, 7, 8, 15]:

$$\begin{aligned} M_{12}(q)\ddot{q} + C_{12}(q,\dot{q})\dot{q}_2 = B_(q)\tau + J_1^T(q)\lambda \\ M_{22}(q_2)\ddot{q} + C_{22}(q_2,\dot{q}_2)\dot{q}_2 = B_2(q_2)\tau \end{aligned} \tag{6}$$

Equation (6) provide an analysis of simple tasks and inverse dynamics. The issue of dynamic modeling is extremely important for the proper solution to this type of traffic control systems.

Another way to describe the dynamics of wheeled mobile robots is to use Maggie equations, which are based on Lagrange equations. Application of equations Maggie bypass procedure allows decoupling of multiplying factors driving moments, as the number of generalized coordinates in the equations Maggie is exactly equal to the number of degrees of freedom. So if driving moments are interesting then applying Maggie equations is more beneficial. Form of these equations is as follows [2, 9]:

$$\sum_{j=1}^{n} C_{ij}\left[\frac{d}{dt}\left(\frac{\partial E}{\partial \dot{q}_j}\right) - \left(\frac{\partial E}{\partial q_j}\right)\right] = \Theta_i \quad i = 1...s \tag{7}$$

where s determines the number of independent parameters of the generalized coordinates q_j (j=1..n) in an amount equal to the number degrees of freedom of the arrangement. Thus, generalized speeds recorded as follows:

$$\dot{q}_j = \sum_{i=1}^{s} C_{ij}\dot{e}_i + G_j \tag{8}$$

In equation (8) we are calling sizes $\dot{e}_i$ characteristics or cinematic parameters of the arrangement in generalized coordinates. Maggie right sides of equations (7) are the coefficients of variations δe_i in the expression prepared to work the system and the external forces are defined as [2, 9]:

$$\sum_{i=1}^{s} \Theta_i \delta e_i = \sum_{i=1}^{s} \delta e_i \sum_{i=1}^{n} C_{ij} Q_j \tag{9}$$

In matrix notation equation (7) recorded as follows:

$$\sum_{j=1}^{n} C_{ij} L_j = \Theta_i \quad i = 1...s \tag{10}$$

where:

$$L = M(q)\ddot{q} + C(q,\dot{q})\dot{q} \tag{11}$$

Maggie equation (7) as Lagrange equations can be used to analyze the task simple and inverse dynamics. Taking into account the problem of control, it appears that the use of equations Maggie is better. Formalisms math presented above also apply in the case of all other wheeled mobile robots.

Having dynamic equations of motion can determine the moments drive, which are extremely interesting because notholonomic control these systems. On the other hand, knowing the time moments of gas mileage, and with the disposal of the dynamic equations of motion, kinematic parameters can be determined model, namely displacement, velocity and acceleration, which can be used to identify the model to describe mobile robot.

Big roles in the modeling process m.r.k. plays way of a dynamic description these systems If exclusively driving moments are interesting in many cases it is easier to use Maggie equations, which are based on Lagrange equations, because isn't necessary to apply the transformation decoupling multiplies moments as is the case with the description of the dynamics Lagrange equations the II type with multipliers.

Additionally they stated, that in case of 2–wheeled mobile robot supporting the influence of the self–adjusting wheel to the mathematical model is relatively small, however the complexity of the calculations rather indicates that it is possible to calculate the mathematical models to take the impact of not-including the self–adjusting propping wheels.

6. The graphical system of controlling the mobile robot

Graphical user interface controls the robot is one of the ways to provide information resulting from the interaction (understood as the exchange of control information) from being put to the robot controller. The principle of the interface based on a drawing and use widgets, i.e. the basic elements of a graphical user interface (GUI), such as slider, window, button or edit field. Used in the application widgets are used to transfer, in this case by the program to the robot, data and user commands.

Allow visualization of the screen data on the state of the program and controlled by the robot (output). With the construction of the desktop environment the user is able convenient and easy communication with an application using keyboard and mouse.

As the environment of the structure and the work of the graphic interface of steering a system was used Linux. The reasons for which it is a Linux system very well is suited for the construction of a robot control interface AmigoBot:

- Linux is characterized by high stability and reliability, and all aspects of control of robot control software need to be tested, free of mistakes and tested in a possibly many existing solutions;
- described later in this work the manufacturer's collection Aria robot, available in C++ language (very good implementation of that language in the system);
- chosen way to communicate with the robot, which is built for Unix systems (like Linux), TCP/IP and very well supported by the wireless system broadcasts a Wi–Fi;
- opportunity to build in this system convenient, providing a large amount of information, not restricted license GUI (Graphical User Interface, i.e. graphical

environment presenting graphical information and allows the user to interact with);

- available for Linux, mobile robots simulator MobileSim;
- properties of Linux as a real–time required for the control and registration of the mobile robot parameters.

The mechanism of communication: robot – a computer with the system Linux

Figure 5 Elements of the mechanism of communication robot – computer

There are many possibilities to ensure communication between Linux and the mobile robot. Both the system chosen and the robot AmigoBot, which is the object of control, equipped with a wide range of available methods and means of communication. Creator of Linux as one of the foundations and the priorities set themselves to communicate, either through a wired medium and as well as wireless. It is equipped with almost all transmission protocols (including TCP/IP protocol that is used to connect the robot) and device drivers needed for communication (such as Wi–Fi cards used for wireless transmission). AmigoBot also allows you to communicate with him by radio (available modules are Wi–Fi or radio modems), or using a wire serial transmission [16].

Due to the fact that the available AmigoBot robots are equipped with wireless communication modules Wi–Fi operating in the 2.4GHz frequency these are exactly this standard of wireless LAN, and more specifically one of its varieties using 802.11b protocol, was chosen as the most in drawing up the optimal solution [16].

7. Characteristic elements communication of the mechanism.

- **WiBox**

  This is a serial–port server, allowing for quick and easy connection of equipment to the asynchronous radio network at 802.11b.

  LANTRONIX WiBox allows any asynchronous device, equipped with RS–232, RS–422 or RS–485 interface, could be connected to an Ethernet network, and thus to the WAN or the Internet [16, 24].

Figure 6 WiBox company LANTRONIX [24]

- **PC with the system Linux**

  Installed a Linux system on which the software was created, and that its represented the working environment and communication applications of the robot is Ubuntu 7.10 Gutsy Gibbon with kernel version 2.6.22–14. Wireless communication with the server serial WiBox provides onboard Wi–Fi.

- **ARIA**

  This is the Object class library (SDK) C++ language, allowing developers to access high and low–level software and functionality of robots MobileRobots company and platforms ActivMedia [23]. It provides a wide range of useful tools in the configuration, control and monitoring of parameters mobile robots such as p3dx, amigo–sh powerbot, p2de, p3atiw–sh.

8. Building a graphical interface of controlling the robot in the environment Linux

The steering interface built at this work is based on the KDE environment, which it beside is GNOME one of most popular for Unix systems.

KDE uses the graphical Qt library as well as from the object Kparts system (subsystem components in KDE) [25].

- **Qt graphics library**

  Robot control interface was built using the Qt programming library. This environment is a set of portable libraries and tools dedicated to languages, C++ and Java, but also to support the programming languages such as Python, Perl, Ruby, PHP, Ada, Pascal, and several others. The basic components of Qt are classes whose task is to build a graphical interface computer program.

  Multiplatform Qt allows to distribute and run applications developed with it, including of course built in this work the control interface for many types of operating systems and embedded platforms.

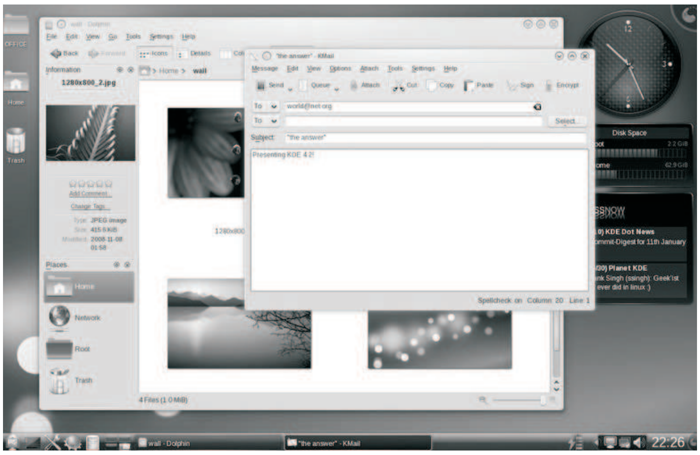

Figure 7 KDE Environment [25]

- **Integrating Qt with the interface Aria**

  In order to build a graphical interface for controlling the robot AmigoBot it was necessary to integrate Qt responsible for the graphical interface and Aria providing the necessary tools and functions of a control and inspection work.

  A key element in collaboration Qt and Aria was a mechanism for Singleton pattern, substantial programming in C++ on Linux. This mechanism is one way to use global variables, that is obvious to read and write the entire program.

  By applying this solution anywhere in the slot, for example supports the call button could be a robot control functions (such as variables (). Robot.getX () – function that returns the robot to move the x–axis).

9. Managing robot interface based on Qt4

Mobile robot control interface AmigoBot was built in C++ programming, using the already presented Aria and Qt4 libraries.

It is a graphic type program which consists of several different elements, the main window, child windows, a large number of different types of widgets and graphics files. All elements of the exchange and process information between themselves, with the program through a keyboard and mouse, and most importantly communicate with the mobile robot in real time. Communication of individual elements of the program the robot is a two–way, information is transmitted in real time, with the program and work to AmigoBota graphics program. Control of mobile robot, by using all the elements presented here, takes place in real time and with feedback control.

The most important and essential element of the application main window is called Robot. In the main the majority of controls and widgets responsible for communications, command and control parameters of the robot and connected to the unit of measurement.

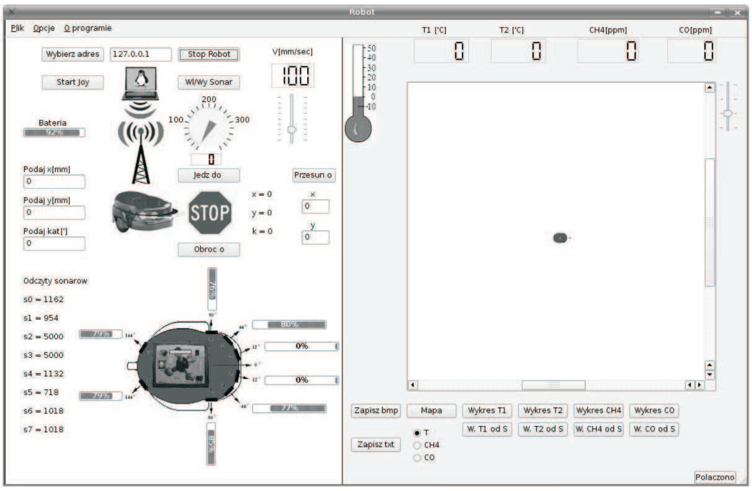

Figure 8 The main window of the program

The most important elements of the main window are:

- the main menu of the program allowing to close the program, providing access to settings and display information about the program;
- The main elements of controlling the robot:
  - a button *Wl/Wy Sonar;*
  - a button *Start Joy* – controlling the robot with the help of the joystick;
  - a button *Go to* – transferring the robot to a set position;
  - a button *Rotate* – rotation the robot angle specified length in field *Enter the angle*;
  - a button *Move to* – shift relative to the value entered;
  - a slider that sets the robot's current speed set point;
  - the speedometer round, presenting a translation robot's current speed;
  - the text labels $x = 0, y = 0, k = 0$, are displayed in real–time position of the robot figures in mm and rotation angle in degrees.

Selecting and operation of the various program options are presented visually to the user, by changing the appropriate symbols in the window next to the robot (Fig. 9, to the right).

- the main setting (Fig. 9, on the left) containing the necessary, relevant working parameters on the only program and the controlled robot (in order to choose them Options > Settings);

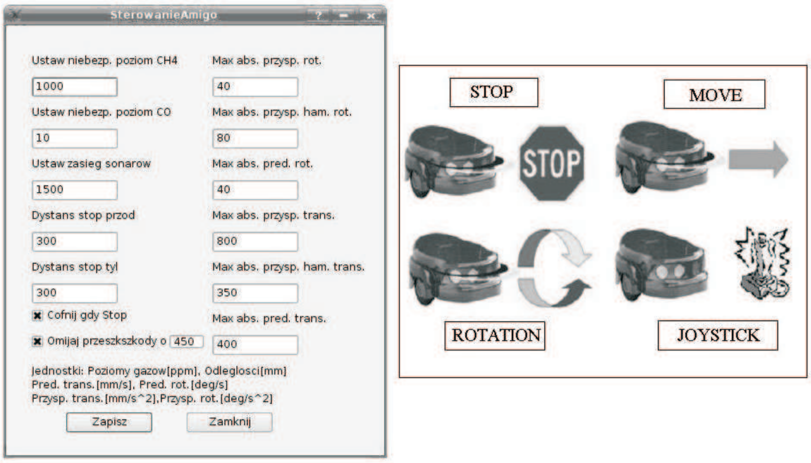

Figure 9 General Settings (on the left), visualization of robot behavior (to the right)

- elements showing the measurements made by sonar (Fig. 10)

  Distances to obstacles measured by the sonar are presented in real time in two ways, through text boxes display the current distance measured in mm and graphically on the strips arranged in a range of locations corresponding to each sonar's.

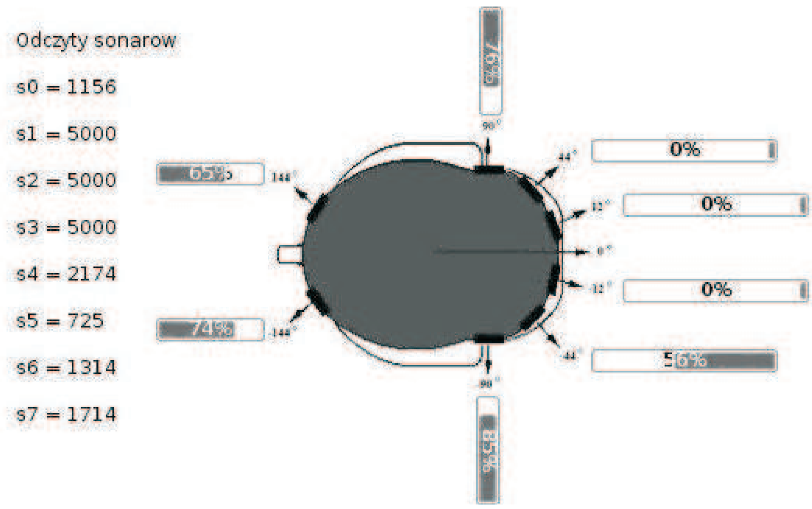

Figure 10 Presentation of data from the robot's sonar

- the map showing the robot and its location (Fig. 11);

  It presents an environment based on measurements of ultrasonic sensors. It shows real–time distribution of temperature and dangerous gasses on the basis of measurements made by the electronic measuring module.

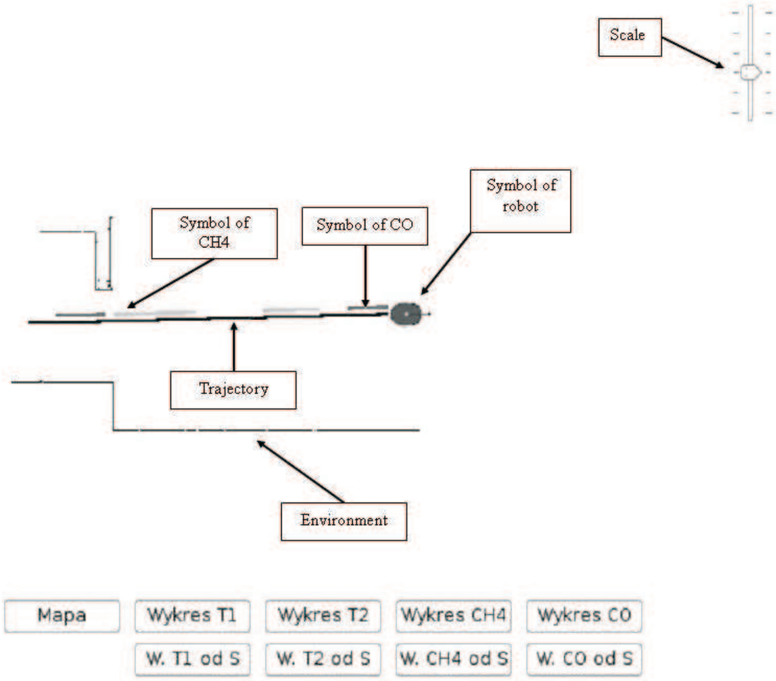

Figure 11 Map of the environment along with a description of the individual elements

- elements of the program that displays information from the electronic module measuring:
  - a thermometer – it shows graphically the temperature measured by the sensor T1 measuring module;

  The temperature of the sensor T2 is presented on the map by color of surroundings drawn on the basis of ultrasonic sensors. This is a very simplistic and not accurate type of the thermal map, but it gives some information about the distribution of temperatures surrounded by the robot.

- a digital display – presenting numerical values of temperature, methane and carbon monoxide (Fig. 12);

Figure 12 Digital displays data from the measuring module

In case of exceeding the dangerous, specific value in placing the program, by the appropriate display a symbol of the warning will appear and in the place of detecting surroundings on the map an appropriate line will be drawn (green for methane and brown for the carbon monoxide Fig. 13).

Figure 13 Digital displays and symbols of the dangerous level of gasses

- elements of the program a recording to external files;
- specify the program allow the movement of the mouse target robot (Fig. 14).

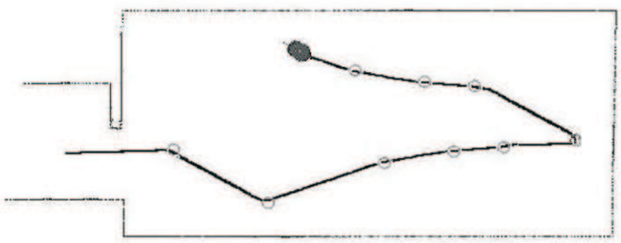

Figure 14 Map of the migration robot to points indicated by mouse

10. Electronic measurement module

An electronic measurement module is a device, whose purpose is to measure the physical size dangerous to man as well as those that require constant monitoring.

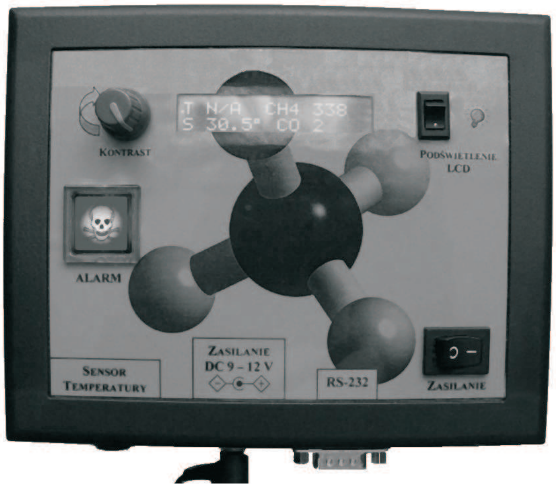

Figure 15 Electronic measurement module [14]

It was built in the thesis of the items [14] literature (in the work presented here one of the tasks was the development of software and hardware combination measuring module of the robot and the development of algorithms for data retrieval and presentation of measurement data in the control). The device was equipped robot AmigoBot for which it was built a graphical user interface for Linux. By combining the module and the robot, the first gained mobility, the possibility of measurement in a changing world and the wireless transfer of measurement (via a modem Wi–Fi robot). Robot has gained unique and complementary set of sensors already possess sensors and enabling to better analyze the environment in which to move. A very important benefit from the cooperation of the two devices is the ability to remotely measure the size dangerous for a human, the concentration of methane and carbon monoxide supplemented by temperature.

- **Method of measuring the communication module with the robot and the graphical user interface running in Linux system.**

  Electronic measurement module is connected to the serial port AmigoBot through an AUX1 located a motherboard of the robot (Fig. 17).

  The way of connecting and communication standards used communication standards are shown in Fig. 16. Mechanism of the data transmission to the control scheme is quite complicated and the algorithm for receiving, processing and interpretation of results require the use of multiple tools for C++ function libraries Aria and Qt. Some methods of the presentation of data from the measuring unit were introduced earlier, they were digital displays thermal map, a graphical thermometer. They have been presented previously, as they represent a very integral and important part of the graphical user interface. They allow steering and controlling the robot, with electronic measuring module connected with him.

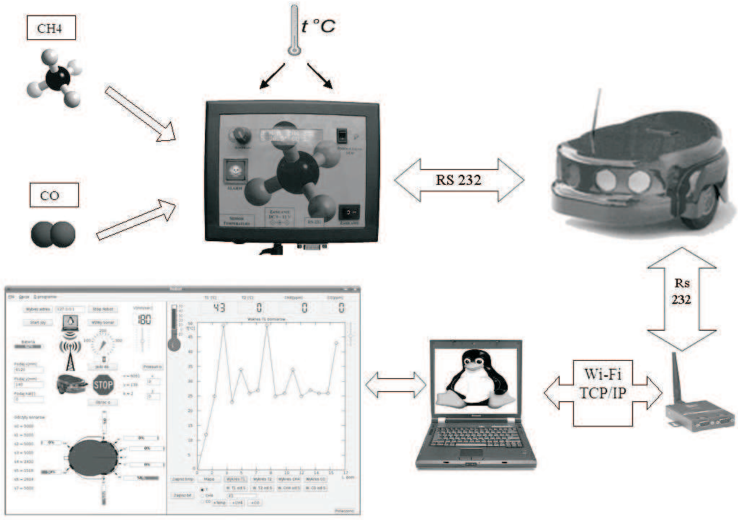

Figure 16 Communication mechanism of the measurement module with the control interface

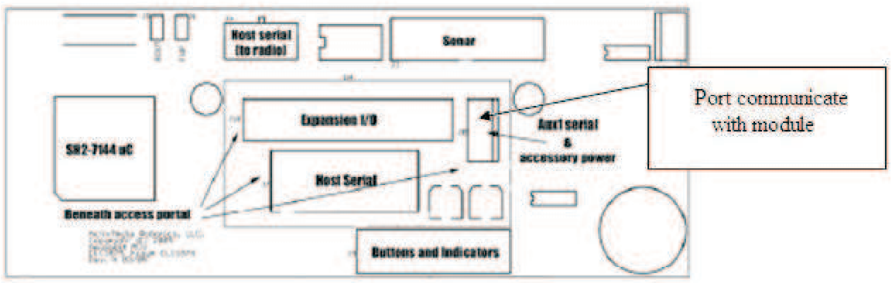

Figure 17 Motherboard controller of the AmigoBot robot [23]

On Fig. 18 and Fig. 19 was presented feature, which allows to generate the main window different graphs values measured by the measurement module and show them the way it functions in the robot traveled.

11. Verification of the developed solution

Verification obtained solutions, consisting of the combination mobile robot and the module AmigoBot circular measuring gasses, was carried out in a robotics laboratory of the Department of Applied Mechanics and Robotics, Technical University of Rzeszow. Testing and research facility was available in the lab, already shown a mobile 3–wheeled AmigoBot.

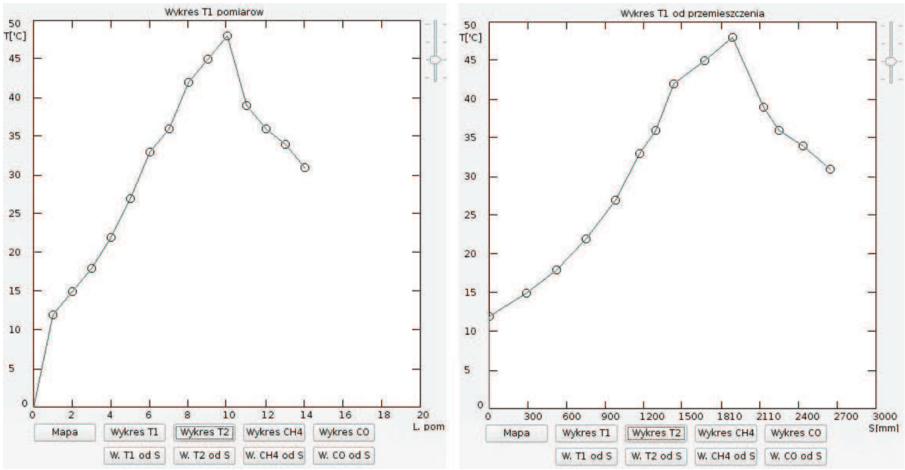

Figure 18 Graph of temperature sensor T1 on the left and a graph of temperature changes on the distance the robot to the right traveled

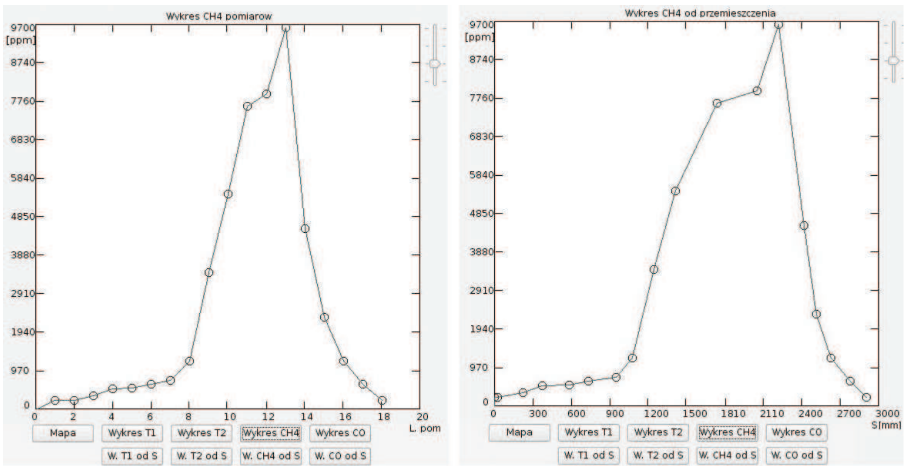

Figure 19 Graph of changes of methane concentration on the left and the graph changes in the concentration of the distance the robot to the right traveled

This robot was presented with pre–electronic measurement module of harmful gasses. In the laboratory has been built right track measurement allows control of test solutions in real–time detection of harmful gasses such as methane and carbon monoxide and temperature in the working environment. Prepared for measuring the track led to the presentation and verify the resulting solution.

In a picture surroundings visible on the Fig. 21 a map built during transfer the robot along the measuring path is shown. It was established on the basis of measurements made by the robot's sonar.

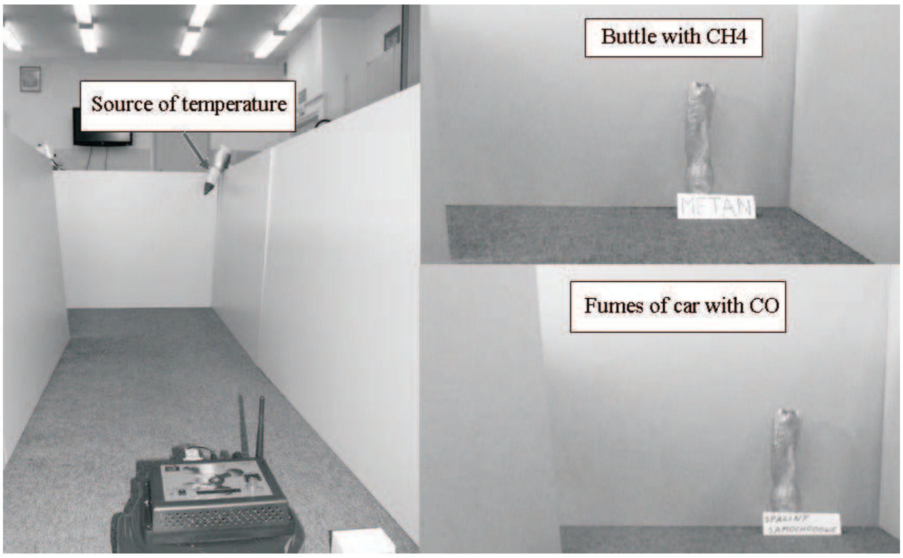

Figure 20 Sources of heat, methane and carbon monoxide used for research

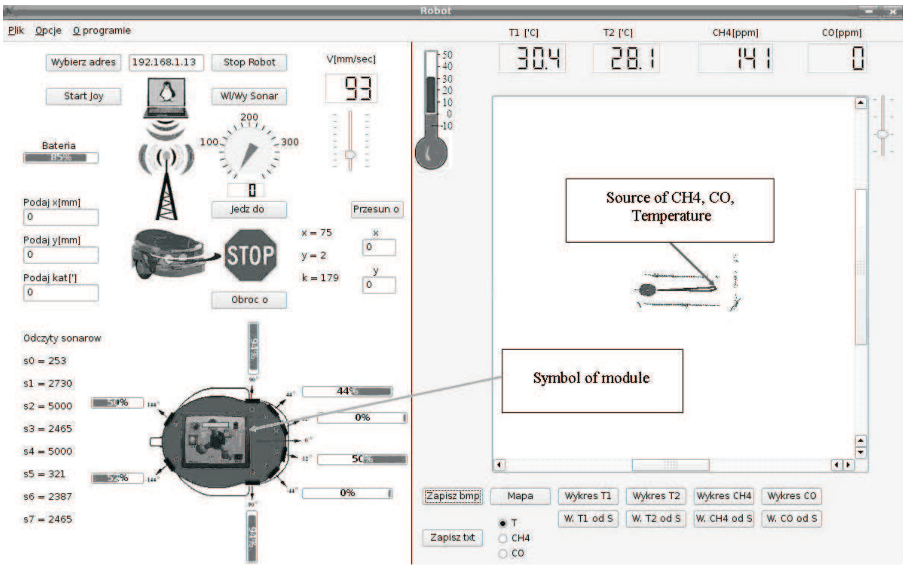

Figure 21 Map of the environment obtained in control

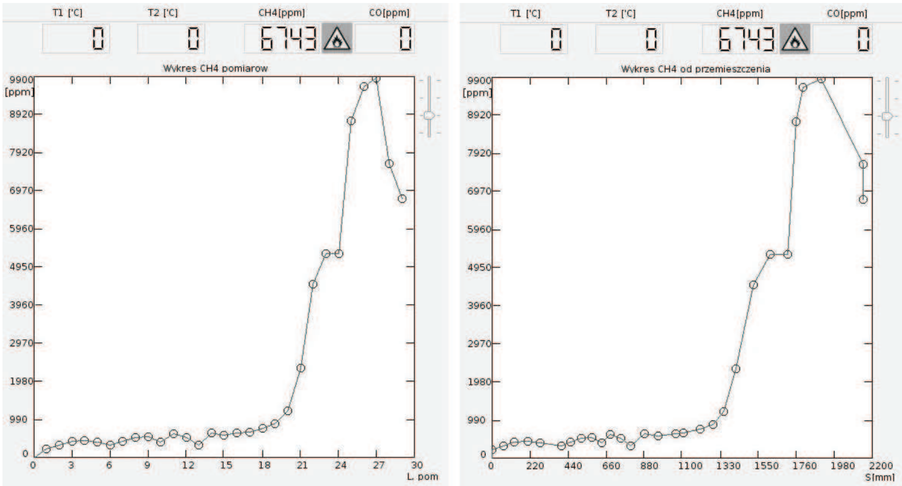

Figure 22 Graphs constructed in control

The accuracy of mapping the environment is not ideal due to the nature and methods of measurement, but allows you to assess the nature and distance of obstacles exist on the robot way. Graphs changes of parameters incurred on the basis of measurements performed by the measurement module are visible while in Fig. 22 Graphs presented with a map showing the environment allow for a thorough and comprehensive evaluation of the environment in real time by the operator a graphical user–interface controls.

12. Summation

As part of the work a mobile platform for measuring the concentration of harmful gases controlled by a graphical application for Linux was build. Presented here was wheeled mobile robot AmigoBot, its characteristics and advantages, described mechanism of cooperation with an electronic measuring module.

Written in C++ program allows to control and control in the real time robot wheeled, with connected to the electronic module, measuring harmful gasses. Measurement system which was a combination of a client computer running Linux, mobile robot and the measurement module. It allows for a modern and advanced testing potentially of dangerous operating environment of the robot. Undoubted benefits of the resulting solution is the mobility, the ability to measure the levels of gasses in a changing surroundings, wireless data transfer. They are complemented by the strengths of the program, such as environment mapping, charting, and several options for controlled movement of the robot and record the data received.

Application system Linux, as construction and environmental control program allows for the creation of a fully legal (free tools and open source code), satisfying the conditions of the control in real time, intuitive and convenient to use the application.

References

[1] **Buratowski T. and Kurc K.**: Podstawy Robotyki i Mechatroniki. Wykłady Prof. Józefa Giergiela, Część 1, Wprowadzenie do Robotyki, Kraków, AGH Katedra Robotyki i Dynamiki Maszyn, **2004**.

[2] **Giergiel M.J., Hendzel Z. and Żylski W.**: Modelowanie i sterowanie mobilnych robotów kołowych, *PWN*, Warszawa, **2002**.

[3] **Giergiel J., Hendzel Z. and Żylski W.**: Zagadnienia dynamiki mobilnych robotów kołowych, *Przegląd Mechaniczny,* Zeszyt 13, 1999.

[4] **Gutkowski R. and Swietlicki W.A.**: Dynamika i drgania układów o skończonej liczbie stopni swobody, *PWN*, Warszawa, **1986**.

[5] **Żylski W.**: Kinematyka i dynamika mobilnych robotów kołowych, *Oficyna Wydawnicza Politechniki Rzeszowskiej*, Rzeszów, **1996**.

[6] **Buratowski T.**: Modelowanie kinematyki mobilnych robotów kołowych przy zastosowaniu procesora symbolicznego Maple^{TM}, Praca magisterska, Kraków, **1999**.

[7] **Giergiel J. and Żylski W.**: Wybrane zagadnienia kinematyki i dynamiki mobilnych robotów kołowych, Materiały: Warsztaty mobilnych robotów, *IPPT PAN*, Warszawa, **1997**.

[8] **Giergiel J. and Żylski W.**: Kinematyka i dynamika dwukołowego mobilnego robota, *Materiały XVI Ogólnopolskiej Konferencji Naukowo–Dydaktycznej Teorii Maszyn i Mechanizmów*, Rzeszów, **1998**.

[9] **Żylski W. and Buratowski T.**: Przekształcenia symboliczne w modelowaniu dynamiki mobilnych robotów kołowych, *Zeszyty Naukowe Katedry Mechaniki Stosowanej*, Zeszyt 15, **2001**.

[10] **Lal K., Rak T. and Orkisz K.**: RTlinux system czasu rzeczywistego, Wydawnictwo *Helion*, **2003**.

[11] **Newman D.**: Linux Robotics Programming Smarter Robots, *TAB Electronics Technician Library*, **2005**.

[12] **Parker T.**: Linux. Księga eksperta, Wydawnictwo *Helion*, **1999**.

[13] **Stutz M.**: Linux. Najlepsze przepisy, Wydawnictwo *Helion*, **2005**.

[14] **Turoń W.**: Moduł pomiarowy – eksploracja niebezpiecznego otoczenia, Praca dyplomowa, promotor Prof. Józef Giergiel, Rzeszów **2009**.

[15] **Giergiel M. and Małka P.**: Modelowanie kinematyki i dynamiki minirobota mobilnego, *XLV Sympozjum modelowanie w mechanice*, Wisła – **2006**.

[16] **Giergiel M. and Małka P.**: Mechatroniczne projektowanie mobilnego minirobota kołowego, Projektowanie mechatroniczne – zagadnienia wybrane, Kraków, s. 76-84, **2005**.

[17] **Giergiel J., Giergiel M. and Małka P.**: Identification m.r.k, *Mechanics and Mechanical Engineering,* Łódź, **2006**.

[18] Magazyn Linux+ wyd. Listopad 2008

[19] http://www.nauka.gildia.pl

[20] http://gazeta–it.pl

[21] http://www.asimo.pl

[22] http://www.elektronikab2b.pl

[23] http://robots.mobilerobots.com

[24] http://www.tradex.pl

[25] http://pl.wikipedia.org/wiki/KDE