BeBot: A Modular Mobile Miniature Robot Platform Supporting Hardware Reconfiguration and Multi-standard Communication

Stefan Herbrechtsmeier[1], Ulf Witkowski[2], and Ulrich Rückert[1]

[1] System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn, Paderborn, Germany
hbmeier@hni.upb.de
[2] Electronics and Circuit Technology, South Westphalia University of Applied Sciences, Soest, Germany
witkowski@fh-swf.de

Abstract. Mobile robots become more and more important in current research and education. Especially small 'on the table' experiments attract interest, because they need no additional or special laboratory equipments. In this context platforms are desirable which are small, simple to access and relatively easy to program. An additional powerful information processing unit is advantageous to simplify the implementation of algorithm and the porting of software from desktop computers to the robot platform. In this paper we present a new versatile miniature robot that can be ideally used for research and education. The small size of the robot of about 9 cm edge length, its robust drive and its modular structure make the robot a general device for single and multi-robot experiments executed 'on the table'. For programming and evaluation the robot can be wirelessly connected via Bluetooth or WiFi. The operating system of the robot is based on the standard Linux kernel and the GNU C standard library. A player/stage model eases software development and testing.

1 Introduction

Mobile robots are more and more in the focus of current research, with almost every major university having one or more labs that focus on mobile robot research. Autonomous robot research is now gaining a broader base by spreading from a few well resourced laboratories into many small university laboratories and even to the hobbyist work bench. This will undoubtedly accelerate the advancement of the field. Alongside, the number of primary and secondary school students participating in robot competitions has increased enormously, reflecting the high educational and entertainment value of low cost autonomous mobile robots. Small robots are also becoming increasingly useful as a test bed for animal behavioural research and for small scale prototyping of larger engineering systems. We at the Heinz Nixdorf Institute are using mobile miniature robots in different applications in various fields like path planning, area explorations,

J.-H. Kim et al. (Eds.): FIRA 2009, CCIS 44, pp. 346–356, 2009.

map building and localization using SLAMs, also in multi-robot experiments like robot swarming, ad hoc networking, robot soccer, etc. Therefore we need a powerful but also small robot that should be compact in size, suitable for various environments even with slightly rough surfaces, and offers extendibility to other modules. All these factors lead to the design of the BeBot miniature robot.

In the following sections, the mini robot BeBot is introduced in detail. Section 2 explains the platform itself, i.e. chassis and electronic hardware architecture. Section 3 focuses on the software environment. Special features of the robot, these are its communication framework and dynamic reconfiguration option, are presented in section 4. Section 5 give some current applications were the BeBot is used. The last section finish the paper with a conclusion.

2 Platform

The miniature robot platform BeBot (figure 1) has been developed at the Heinz Nixdorf Institute, University of Paderborn. It has a size of approximately 9 x $9cm^2$ and a height of about 5cm.

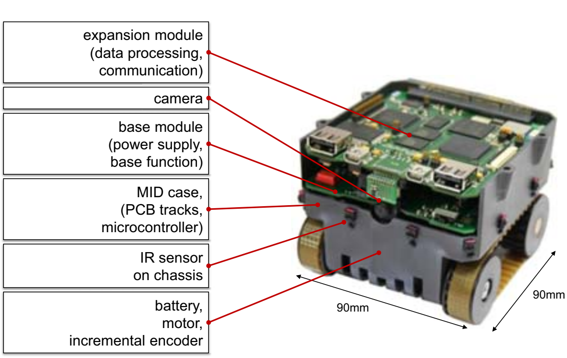

Fig. 1. BeBot mini robot (fully equipped)

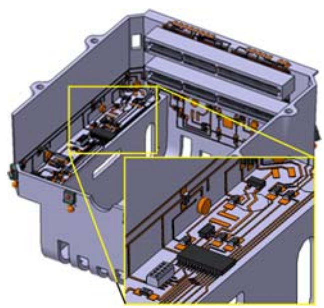

Fig. 2. MID chassis of the BeBot robot

The chassis uses MID (molded interconnect device) technology and has traces directly on the surface [1] which offers new possibilities for the synergistic integration of mechanics and electronics. Figure 2 depicts the integration of electronic components on the plastic chassis. A microcontroller on the left and right side is used to control the infrared (IR) senders. Also the received IR signals are processed by the microcontroller and the digitized data is sent via I^2C link to the others processors used within the robot. The MID technology allows the assembly of electrical components directly on the device. This technique is used for mounting 12 infrared sensors and two microcontrollers, several transistors and resistors for preprocessing directly on the robot chassis. The actuation consists of a chain drive. Together with two 2W dc gear motors with built-in encoders the robot offers robust motion even on slightly rough ground. The complete system is supplied by a 3.7V / 3900mAh lithium-ion accumulator that allows a runtime of approximately 4 hours for a full equipped robot.

The robot uses a modular concept for information processing and has two slots for extension board. The lower board (base module in figure 1) implements basis functions like motor control and power supply. An ARM 7 based microcontroller allows low level behavior realization. The module also contains a three axis acceleration sensor, a yaw rate gyroscope and a sensor for battery monitoring. A possible application for the gyroscope and the 3D accelerometer are local navigation algorithms. Because of the high accuracy and low drift of these sensors they are able to provide lower errors than an odometry based on wheel encoders.

The upper slot (expansion module in figure 1) provides a more powerful information processing and wireless communication. It is equipped with a low power 520MHz processor, 64MB main and flash memory. An FPGA (field programmable gate array) enables the use of reconfiguration on hardware level. This allows the computation of complex algorithms through the use of dynamic coprocessors. The integrated wireless communication standards ZigBee, Bluetooth and WLAN offer communication with various bandwidth and power consumption. The board provides a variety of additional interfaces, like USB, MMC / SD-card, audio, LCD and camera.

A new version of the expansion module optimized in terms of size and power consumption is in development. This supports different techniques for energy saving like dynamic frequency and voltage scaling as well as dynamic power down of non-used hardware components including RF processing. Main device of the new expansion module is Texas Instruments (TI) new OMAP 3 processor. This high-performance applications processor consists of a 600MHz ARM Cortex-A8 processor with NEON SIMD coprocessor. It supports dynamic branch prediction and has a comprehensive power and clock management, which enables high-performance, low-power operation via TI's SmartReflex adaptive voltage control. It offers more than 1200Dhrystone MIPS with maximal power consumption from less than 2W for the whole chip. The processor is connected to 512MB NAND Flash and 256MB mobile low power DDR SDRAM. It is equipped with the wireless communication standards Bluetooth and WiFi. Both have external antennas for better signal qualities and support power down to disable not needed communication devices. A coexistence solution ensures simultaneous operation of Bluetooth and WiFi. Both communication devices have a peak power consumption of less than 1W during continuous transmit over Bluetooth and WiFi. Additionally, the supported wired communication standards I^2C, SPI, UART and high speed USB allows variable expansion of the system. Via these interfaces other communication devices like ZigBee, Sub-1 GHz communication or UWB hardware or other components like sensors and actuators can be easily connected to the system enabling the robot meeting several demands. The main interface to a computer is a USB device interface.

From the information processing point a view the robot is an embedded system providing distributed processing. Besides the main processor (Marvel PXA270) three additional microcontrollers are available for distributed processing. For data exchange, the processor and microcontroller are connected by an I^{2}C link. Closely coupled to the main processor an FPGA device (Xilinx XC3S1600E) has been integrated. It is connected to the memory bus enabling high bandwidth data exchange between FPGA, processor, and the memory devices.

The BeBot can be used in mainly two hardware configurations, not considering mechatronic extensions like a gripper or transporter at this point. In the minimal configuration, shown in figure 3, the robot is able to perform simple experiments. In this configuration the motors can be controlled - a speed controller based on integrated wheel encoders is already implemented. The microcontroller of the base board can get sensor data from the microcontrollers mounted on the chassis via the I^{2}C link. Possible behaviors in this configuration are simple exploration strategies or behaviors known as Braitenberg behaviors [2]. Additionally, the robot can be remote controlled by using a Bluetooth wireless link. Control commands can be sent by a PDA as shown in figure 4. A feedback is given from the robot by sending the data from the infrared sensors to the PDA.

Fig. 3. Robot in minimal configuration equipped with base board

Fig. 4. Remote control of the robot via a PDA

Figure 1 depicts the BeBot robot fully equipped with hardware modules, both PCBs are inserted. In this configuration, the robot's hardware architecture corresponds to the architecture presented above. Powerful processing devices (processor and FPGA) are available for the implementation of complex algorithms. The advantage of using an FPGA device on this platform is discussed in the section 4.2.

3 Software Environment

The software environment of the robot is a Linux operating system. It consists of a modified Linux kernel 2.6.26, the GNU C standard library and the device manager udev. The standard Unix tools were provided by the software application

BusyBox. This combines tiny versions of many common Unix utilities in a single small executable. The software building is done via OpenRobotix [3]. This is an extension of the OpenEmbedded development environment which allows the creation of a fully usable Linux operating system. It generates cross-compile software packages and images for the embedded target. The existing software branch was extended to contain the robot specific information, patches and additional software like the Player network server and drivers for the robot hardware.

The Player project is a language and platform independent robot control system. It consists of the Player client/server model and the platform simulators Stage and Gazebo. Player implements a Player server and a Player library which is used to build the Player client. Stage is a 2D robot simulator with a player interface. Together they allow the platform and real / simulation independent robot programming.

The interface of the OS to the robot's hardware is divided into two parts. The first part is a kernel driver. This implements the low level I^2C communication and provides user space hardware control over virtual files in the Linux sys file system. Above, a Player driver uses these files and fulfills the second part. It implements the player driver class, makes number conversions and adds some additional information like sensor position and robot sizes. The whole system allows the controlling of the robot through the player interfaces. Additionally, a robot model for the Stage simulator allows the simulation of experiments with the same software interface.

The WiFi communication is directly supported through the Linux kernel and so supports all standard communication protocols. The Bluetooth communication is implemented by the BlueZ protocol stack and supports all standard Bluetooth protocols like RFCOMM and BNEP. Additionally, all Linux and platform independent or arm compatible protocol implantations can be ported to the robot platform. One example is the ad hoc wireless mesh routing daemon OLSRD. This implements the optimized link state routing protocol and allows mesh routing on any network device.

4 Special Features

The robot is supporting special features like wireless communication via several communication standards for synergetic combination. Additionally, partially dynamic reconfiguration is supported to maximize hardware utilization and to cope with restricted resources of the mobile robot. Both features are supported by the robot's operating system to ease program development.

4.1 Wireless Communication

Three types of wireless communication are supported by the mini robot BeBot. Bluetooth and ZigBee are directly integrated onto the robot's extension board. Communication via WiFi is realized by connecting a WLAN device to a USB connector. The implemented communication standards differ in network size, radio range, data rate and power consumption. Wireless LAN is suited to high data rate and high range communication at the cost of high power consumption. Bluetooth has a lower data rate and transmission range but in turn significantly lower power consumption. ZigBee is highly scalable with even lower power consumption but with a trade off for lower data rates. One major challenge in wireless ad hoc networks, particularly in mobile ad hoc networks, is the design of efficient routing algorithms. All integrated communication devices are supported by the Linux operating system running on the robot. It is possible to directly access the communication devices. But in order to ease the access to the communication devices, a network abstraction layer and a communication framework on top have been integrated.

4.2 Dynamic Reconfiguration

The robot supports dynamic reconfiguration of its hardware during runtime [4]. Reconfiguration capabilities are provides by the FPGA device. Several types of reconfigurations are supported. At startup of the robot, the FPGA is programmed by the processor with the contents of the Flash memory. This allows loading different hardware configurations for the FPGA at power-on of the robot to perform build-in routines e.g. self-test, demonstration mode etc. The FPGA is capable of dynamical reconfiguration, so that parts of the hardware design on the FPGA can be exchanged on demand by new modules, kept in Flash, SDRAM or received wirelessly by Bluetooth or WiFi via a network link.

Modern FPGAs are heterogeneous architectures constituted by programmable functional blocks and embedded application specific integrated hardware (e.g., embedded processors, SRAM memory, dedicated multipliers) interconnected by a reconfigurable network. The configuration and interconnection of the internal resources determine the functionality of the implemented design. This configuration is provided by a bitstream file, which is loaded at start-up. Some FPGAs can be partially reconfigured during run time. A partial bitstream, targeting a specific area of the FPGA is loaded while the rest of the FPGA can still operate without interruption. This process is known as dynamic reconfiguration, which can be used to enhance the resource-utilization of an FPGA by time-sharing logical resources among different designs (event-driven reconfiguration) or by time-multiplexing a design requiring a bigger amount of resources than available (virtual hardware) [5]. The event driven concept is explained in the following paragraph.

Given the resource limitations of FPGAs, it is not possible to realize a large number of algorithms using only static designs. Furthermore, it would be a waste of resources to implement these algorithms statically if they are not needed the whole time. Event-driven dynamic reconfiguration can be modeled as finite state machine where every state represents a different configuration of the hardware. Mutually exclusive configurations (e.g., designs that are not needed at the same time) time-share the same hardware slot on the dynamically reconfigurable area, where several slots can coexist (e.g., there are several non-mutually exclusive designs). Event-driven dynamic reconfiguration makes it possible to adapt, during run time, the behavior of the system without wasting resources.

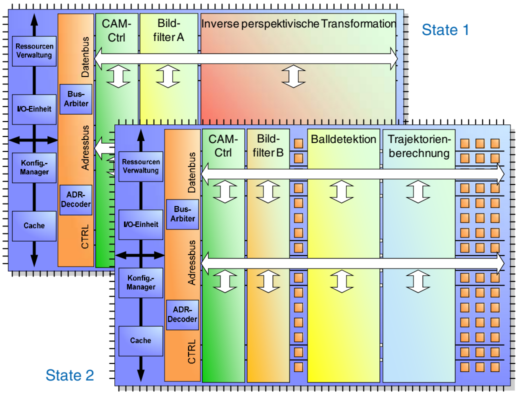

Fig. 5. Dynamic reconfiguration from state 1 to state 2

An example of the event-driven dynamic reconfiguration is shown in figure 5. Two simple states have been defined. The considered scenario is from robot soccer where the robot has the task to orientate itself in the soccer pitch, to detect the ball and to calculate trajectories to push the ball into the desired direction. The perception is done by 2D CMOS color camera. In state one, the FPGA is configured with a camera controller for image capturing, one image filter algorithm and a complex unit for calculating the inverse perspective transformation [6]. This calculation is done to get the robot's position based on the lines in the pitch. After the calculation of the robot's positions has been done, the FPGA is reconfigured, which means, that another filter is loaded and algorithms for ball detection and calculation of robot's trajectories are instantiated. This is state two in figure 5. If it's again necessary to calculate an updated position of the robot, the configuration is switched to state 1. By doing this reconfiguration, complex algorithms can be parallel executed in hardware with good resource usage.

5 Applications

The robot has been successfully used in education of students and research projects in the field of robotics. In the following paragraphs a student project is presented as well as the EU funded GUARDIANS research project. Furthermore, the robot BeBot is used as a platform in the SFB614 "Self-Optimizing Concepts and Structures in Mechanical Engineering".

5.1 Mechatronic Seminar

The aim of this student project was the development and implementation of a control strategy that makes it possible for a group of robots to drive collision free and as fast as possible through a gate from one side of a field to the other side, see figure 6. The challenge was to design an overall concept to solve the whole

Fig. 6. Student project: robots have to change the ends as fast as possible passing the narrow gate

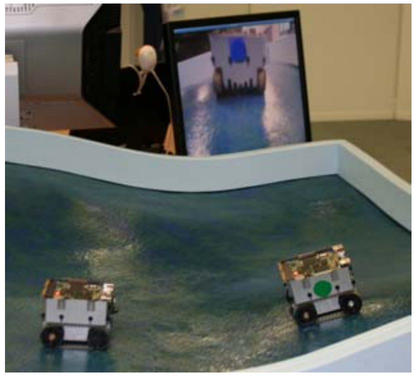

Fig. 7. Student project: robots have to follow each other based on color recognition

problem and to split this in several tasks. The tasks were trajectory planning, model building of the robot, controller design, implementation and start-up of the whole system to verify the developed model and the control strategy.

The robot experiments were performed on our teleworkbench platform [7]. This platform allows the recording of multi robot experiments together with a live position tracking of the robots. The robots can ask for their current positions when acting on the teleworkbench. Based on this position data the robots have organized to drive from one side to the other. The group of students has successfully solved this multi robot problem.

5.2 Image Processing Project

Another educational project in Heinz Nixdorf Institute is to teach students how to use and program miniature robots. In this relationship a student project was to build a robot platoon follow each other based on image processing. One robot equipped with color markers drives randomly or based on Braitenberg behaviors throw a separate area. A second robot tries to follow the first one. Therefore it uses color recognition to find the color markers and by controlling the size and position of the color blob in the image the robot can control its distance to the leading robot. The challenge is to keep this behavior even on rough underground, see figure 7.

5.3 Research Project Guardians

Main disaster scenario covered by the GUARDIANS project (funded by Sixth Framework Programme of the European Union, no. 045269) is a large industrial warehouse on fire, where black smoke may fill large space of the warehouse that makes it very difficult for the firefighters to orientate in the building and

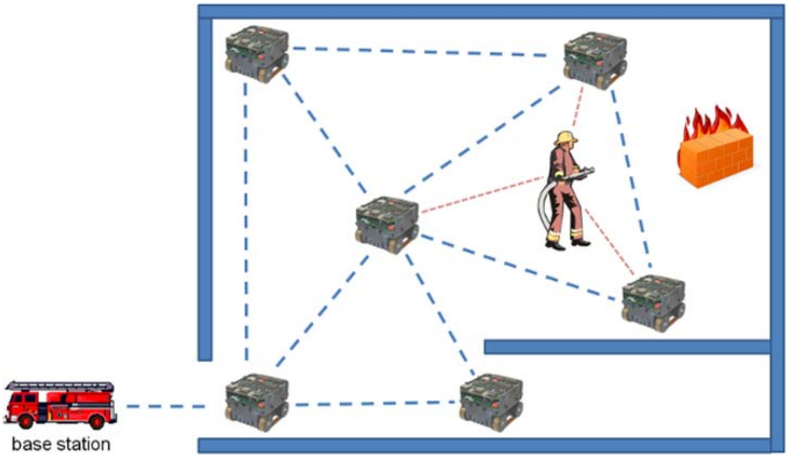

Fig. 8. Outline of a supporting communication network for fire fighters as proposed in the GUARDIANS project

thus limits the action space of the firefighters. During such mission, the robots navigate the site autonomously and serve as a guide for a human squad-leader in finding the target location or in avoiding dangerous locations or objects. They connect to a wireless ad hoc network and forward data to the squad-leader and the control station.

The ad-hoc network, which is actually a chain of robots equipped with wireless communication modules, is self-optimizing, adapts to connection failures by modifying its connections from local up to central connections [8, 9]. The autonomous swarm operates in communicative and non-communicative mode. In communicative mode, automatic service discovery is applied: the robots find peers to help them. The wireless network as depicted in figure 8 also enables the robots to support a human squad-leader operating within close range. In the

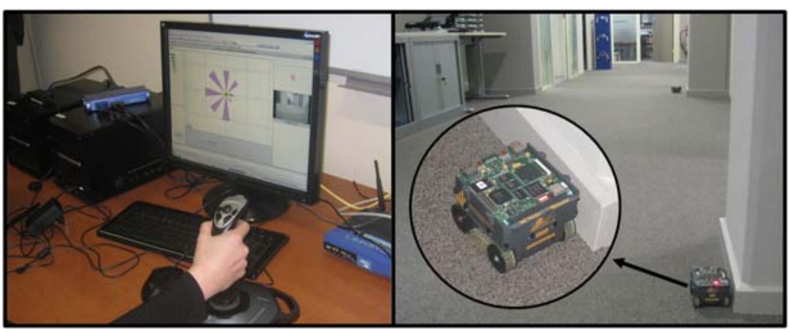

Fig. 9. Mission monitoring and remote control of selected robots via the base station in the mobile ad hoc network

case of loosing network signals, the robot swarm can still be functioning with non-communicative mode and continue serving the fire fighters.

Several robot platforms are used in this project, some for the down-scaled scenarios like Khepera III and the BeBot robot, others for the real scenarios provided by the Spanish partner Robotnik. As an example for the BeBot usage in the project, figure 9 depicts the BeBot used to build the mobile ad hoc network. The robots manage a TCP/IP based communication over different communication standards and enable the mission monitor and control in the base station.

6 Conclusion

We have introduced a new powerful and versatile mini robot optimized for small scale experiments supporting student project as well as research work. The miniature robot BeBot offers powerful information processing including dynamic reconfiguration via an integrated FPGA. Different wireless communication techniques enable comfortable wireless access to the robot. Via the Linux operating system the setup and maintenance of large communication networks is supported. A model of the robot using the Player/Stage framework merges simulations and real experiments. Due to its modular architecture the robot can be easily extended by new sensors and mechatronic modules to realize heterogeneous groups of robots. The robot has been successfully used in student projects as well as in research projects.

Acknowledgment

The development of the BeBot mini robot platform (MID chassis) is carried out in cooperation with the department of Computer Integrated Manufacturing (Prof. J. Gausemeier) of the Heinz Nixdorf Institute and the Corporate Technology Department of Siemens AG, Berlin. This work was supported by the Sixth Framework Program of the European Union as part of the GUARDIANS project (no. 045269, www.guardians-project.eu) and by the German Collaborative Research Center 614 - Self-Optimizing Concepts and Structures in Mechanical Engineering (SFB614, www.sfb614.de).

References

1. Kaiser, I., Kaulmann, T., Gausemeier, J., Witkowski, U.: Miniaturization of autonomous robot by the new technology molded interconnect devices (mid). In: Proceedings of the 4th International AMiRE Symposium, Buenos Aires (2007)
2. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge (1984)
3. OpenRobotix: Openembedded based open source linux distribution for mini robots, http://openrobotix.berlios.de/

4. Rana, V., Santambrogio, M., Sciuto, D., Kettelhoit, B., Köster, M., Porrmann, M., Rückert, U.: Partial dynamic reconfiguration in a multi-fpga clustered architecture based on linux. In: Proceedings of the 21th International Parallel and Distributed Processing Symposium (IPDPS 2007); Reconfigurable Architectures Workshop (RAW), Long Beach, California, USA (2007)
5. Paiz, C., Chinapirom, T., Witkowski, U., Porrmann, M.: Dynamically reconfigurable hardware for autonomous mini-robots. In: The 32nd Annual Conference of the IEEE Industrial Electronics Society (2006)
6. Witkowski, U., Chinapirom, T., Rückert, U.: Self-orientation of soccer robots on soccer pitch by identifying pitch lines. In: Proceedings of FIRA RoboWorld Congress, Dortmund, Germany, pp. 13–18 (2006)
7. Tanoto, A., Witkowski, U., Rückert, U.: Teleworkbench: A teleoperated platform for multi-robot experiments. In: Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005), Awara-Spa, Fukui, JAPAN (2005)
8. Witkowski, U., El Habbal, M.A.M., Herbrechtsmeier, S., Tanoto, A., Penders, J., Alboul, L., Gazi, V.: Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios. In: Proceedings of IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE 2008), Benicassim, Spain (2008)
9. Witkowski, U., El Habbal, M.A.M., Herbrechtsmeier, S., Penders, J., Alboul, L., Motard, E., Gancet, J.: Mobile ad-hoc communication in highly dynamic environment optimized with respect to robustness, size and power efficiency. In: Proceedings of the International Workshop on Robotics for risky interventions and Environmental Surveillance (RISE 2009), Brussels, Belgium (2009)