Highly Adaptable Hardware Architecture for Scientific and Industrial Mobile Robots

Matthias Merten*† and Horst-Michael Gross†
*MetraLabs GmbH, Germany
Email: matthias.merten@metralabs.com
†Neuroinformatics and Cognitive Robotics Lab
Ilmenau University of Technology, Ilmenau, Germany
Email: horst-michael.gross@tu-ilmenau.de

***Abstract*— The growing field of mobile robotics produces various types of robot systems for different applications. However, the development of new robots often requires the design of new hardware systems which is time consuming and susceptible to errors. Here, we present a modular and highly flexible hardware architecture that allows to simplify the adaptation of the robot to a new task and reduces the time for the development of a new robot system. This hardware architecture was developed according to industrial standards and is applicable to industrial and to research mobile robot systems. This paper describes the underlying demands to the hardware, the technical solutions and the designed modules. Moreover, we show the successful integration of the new architecture in three different robot systems (indoor and outdoor)[1].**

I. INTRODUCTION

The usage of mobile robot systems in industrial applications, research projects and end-user-products has produced numerous technical solutions. Independent statistical evaluations (e.g., [1]) predict a further increase of mobile robot usage in the next years. Through the progress of technology and algorithm development, robot systems are capable of solving more complex tasks, so new market segments for applications of mobile robot systems appear. The success of a new robot system will be influenced not only by the price of the platform, but also by the quality and the time-to-market of the system. Consequently, the hardware of a mobile robot, the mechanical and electrical components, must be flexible to meet different requirements and to allow a fast adjustment to novel tasks. The re-usability would make the development faster and would reduce the failure rate of new units.

Up to now, several hardware architectures were developed to solve these demands. For example, the design of the K9 and K10 rover at the NASA Ames Research Center [2] has used combinations of off-the-shelf hardware and custom designed hardware to realize an extensible architecture. Moreover, ARMAR III and LAURON IV robot systems [3] use a modular hardware inspired by modular software and computer architectures.

However, the re-use of the hardware in a robot system and the extension with additional components are still very restricted in current solutions. The hardware architectures are highly specialized and adapted to the final application, so the expansion with additional hardware modules is complicated. In particular, this is a huge disadvantage for scientific research projects. The solutions of modular hardware architectures, mentioned above, provide a good approach, but they are still not flexible enough and have only small numbers of different functionalities.

Therefore, we developed a modular hardware architecture, where the requirements were defined by industrial demands and the needs of the scientists in the field of mobile robot research. The hardware was designed under the consideration of industrial standards. The functioning of our highly adaptable hardware architecture was tested in three different robot platforms: the SCITOS G5 (Fig. 1), a research and industrial platform; the SCITOS A5 (Fig. 6), a mobile shopping assistant; and MILVA (Fig. 7), an outdoor research platform.

In the following section, we give an introduction into the requirements of the new hardware architecture. In section III, we will present the technical solution, which includes the communication between the modules and the power supply as well as some essential hardware modules. Finally, we will describe the successful implementation of the hardware architecture in three robot applications.

Fig. 1. Mobile robot platform SCITOS G5

---

[1] This work was partially supported by the TAB Grant #2006FE0154 to Ilmenau University of Technology and #2006FE0153 to MetraLabs GmbH Ilmenau.

978–1–4244–1676–9/08/$25.00 © 2008 IEEE

II. REQUIREMENTS

Three application fields defined the technical requirements for the new hardware architecture. First, the application in end-user products, like the shopping-assistant [4], [5], developed in the SERvice-ROboter-KONzeption (SERROKON) project. Therefore, necessary criteria were a reliable functionality over a working period of over five years, the possibility of an off-the-shelf production process and an acceptable price. On the other hand, the integration of the hardware in various research projects required a high flexibility for the adaptation to many different tasks, a reasonable software access to all standard and additional hardware modules and the preparation of communication interfaces for the connection of user-defined add-ons. Finally, the implementation of the hardware architecture in industrial products required a development oriented on various industrial directives.

One main issue of the architecture is the concept of the power supply. The power consumption of the electronic components should allow operating times over ten hours. Thus, an *Embedded PC* with an optimized use of energy should be integrated. The electronic modules like motor controller, external power supplies or sensor modules must have a small power consumption and the possibility to be turned off completely by software to save energy. Another aspect is the input voltage of the electronic modules, which should range between $+9.0V$ and $+30.0V$. This requirement is necessary to allow a stable operation in $12V$ and $24V$ systems. So, even high current loads caused by interference voltage peaks (e.g., produced by the motor controller) can be tolerated irrespective of the accumulator charging state. The accumulator must be usable in stressful modes to allow cycle charges as well as high current discharges.

The main requirement on the communication concept was a standard interface combined with a verified communication protocol. Our goal was to apply a standard protocol tested on a long run to avoid implementation problems. This standard protocol would also allow the connection of hardware modules of various third-party manufacturer. A small 8-Bit processor should be able to manage the physical interface and the communication protocol in order to save energy and production costs.

Finally, the developed hardware architecture must follow the obligations of the electromagnetic compatibility (EMC) and the electromagnetic influence (EMI) for industrial applications. For all end-user products, functionality and safety features must fulfill the requirements of the German Technical Inspection Agency (TÜV).

III. REALIZATION

This section will describe the technical realization of the new hardware architecture under the consideration of the discussed requirements. To outline the implemented solutions, we start with the description of the realized hardware modules. Then, we will illustrate technical basics like communication, power supply and security mechanisms in more detail.

A. Modules

During the development of this hardware architecture, different modules have been developed to adapt the architecture to various tasks. Fig. 4 shows an overview of the developed modules, which were used in the three example implementations. The modules can be classified into two groups. The first group contains modules that have to be placed in special construction groups because of functional or producible reasons. Theses devices are adapted in size and shape to the assigned position in a robot platform, e.g., the *Charger*, the *DisplayModule* or the *RobotHead* module (Fig. 3).

The second group includes modules that are designed to fit in a slot-based electronic case. This concept, which is oriented on the industrial $19''$-technology, allows a very flexible combination and expansion and reduces the need of wiring-based connections between the modules. Consequently, the main control units of a robot system, e.g., the *MotorController*, the *SensorModule* (Fig. 2) and External Power Supplies (*EBC-Modules*) are placed in the electronic case.

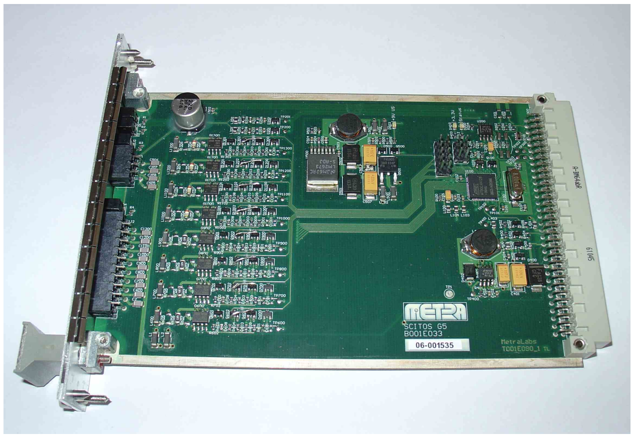

Fig. 2. PCB of the *SensorModule*

So far, the following modules were developed and used in several robot applications:

- *Charger* - This module is responsible for the control of the charging and discharging process of the accumulators, it turns the robot on and off and handles security tasks to protect the robot system from external failures.
- *PowerModule* - This module contains indicator elements for the display of different system states and filters the incoming power and signal lines. It does not contain any intelligent components, like a microcontroller or a FPGA.
- *MotorConroller* - This - system specific - module controls the drive systems of a mobile robot platform. It calculates the odometry of the robot and analyzes security-relevant sensor systems like collision sensors and emergency stop buttons.
- *SensorModule* - This unit communicates with external sensor modules like the ultrasonic sensors and provides different analog and digital inputs and outputs.
- *JoystickModule* - This module contains an interface for the connection of an analog joystick for the control of

the robot platform. Furthermore, it has an external service port for debugging and configuration.

- *EBC-Module* - This unit produces different voltage levels for the connection of external loads, e.g., laser range finders or camera systems. It provides switchable and protected outputs for the unregulated voltage level of the accumulators ($2 \times 4A$), a regulated $+12V$ output ($2 \times 2.5A$) and a regulated $+5V$ output ($2 \times 2.5A$). The turn-off current of every output channel is software-configurable. A feature of this module is the slot-based node-ID. This allows the usage of more than one *EBC-Module* to expand the available supply outputs.
- *US-Module* - Every US-Module controls an ultrasonic sensor. The modules are connected by an I2C-Bus to the *SensorModule*.
- *Backplane* - This unit distributes power and signal lines to slot-bases modules. It is the second module without intelligent circuits.
- *InterfaceModule* - This module provides the interface to the *Embedded PC* and controls its functionality. Furthermore, it prepares internal interfaces of the PC for an external access.
- *DisplayModule* - This module allows the connection of a TFT-display to the robot system. Moreover, it contains a multimedia interface for video conference applications.
- *RobotHead* - This device was developed for the control of a robot head. It is able to control up to six servo motors, three stepper motors and one linear motor. It contains an interface to an optical indicator PCB, a temperature sensor and a brightness sensor.
- *StatusDisplay* - This user-accessible module allows for the setting of important system parameter, e.g., turning off/on of the EBC-Ports or the reset of the robot's odometry and the control of system parameter, e.g., accumulator voltage, charging state.
- *ACS-Charger* - This additional charger device was developed to be usable with an external autonomous charging station.
- *RFID-Reader* - This unit allows the reading of RFID-Tags.

The *Embedded PC* is the control center of the hardware architecture. Although, a communication between different modules takes place, the internal PC is responsible for the basic control of the robot. We decided to use an industrial mini-ITX based motherboard with a low power consumption and a high computation power, which perfectly fits to the requirements of the architecture. To minimize the cabling complexity, the *Embedded PC* is placed inside the electronic case and connected to the *InterfaceModule*.

After the boot-up of the robot system and the start of the *Embedded PC* a basis software package checks the available devices and creates an image of the current system configuration. Accordingly, this software package enables available features of the hardware architecture for the access by the user-built software of the *Embedded PC*. If a module is hot-plugged in or out, the base software of the PC will communicate the changes to the user-built software.

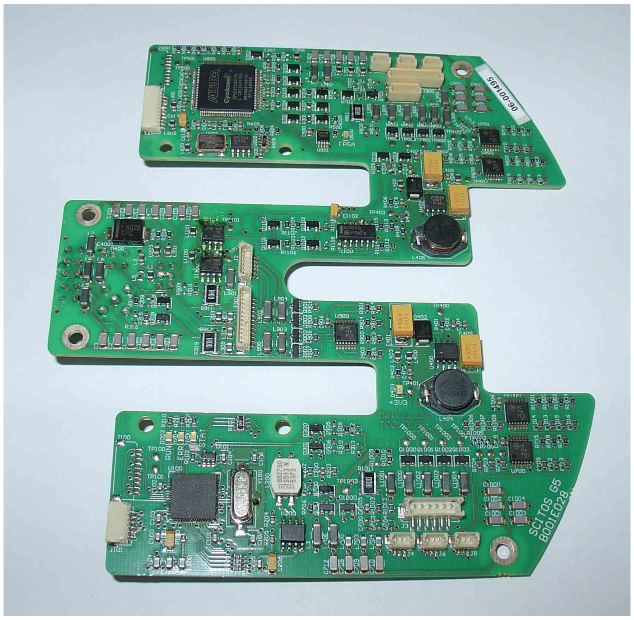

Fig. 3. PCB of the *RobotHead* module, mounted at the neck position of the robot head

B. Communication

The main communication network in our new hardware architecture is based on the Controller Area Network (CAN) by the company Bosch (Germany). This physical bus was developed in the year 1983 and is used in many industrial and automotive products. Nowadays, many stand-alone CAN controller circuits as well as CAN controllers in combination with standard microcontrollers are available. In our implementations, we use mainly the AT90CAN128 microcontroller (Atmel, USA) because this chip is already equipped with a CAN controller and has a suitable cost-performance ratio.

In addition to the CAN-bus, other communication networks are available in our system. We included an I2C-bus for the communication between the ultrasonic sensors to save space and costs. For high speed communication between the *Embedded PC* and external displays, a Low Voltage Differential Signaling (LVDS) interfaces is available. Furthermore, standard interfaces like RS232, Ethernet, USB or IrDA are applicable for add-ons.

To ensure a safe communication between the different modules, we decided on the communication protocol CANopen [6], [7]. This protocol is known from industrial machines for the transfer of process information and is equipped with several useful features like synchronization, error indication and software update mechanisms. The maximum communication speed is defined by the specification of the CAN-bus with up to $1MBit/s$. This transfer rate is adequate for the transfer of actuator commands, sensor information and control sequences. The underlying software structure is shown in Fig. 5. The interface to the user application is defined by an object dictionary where the process information that should

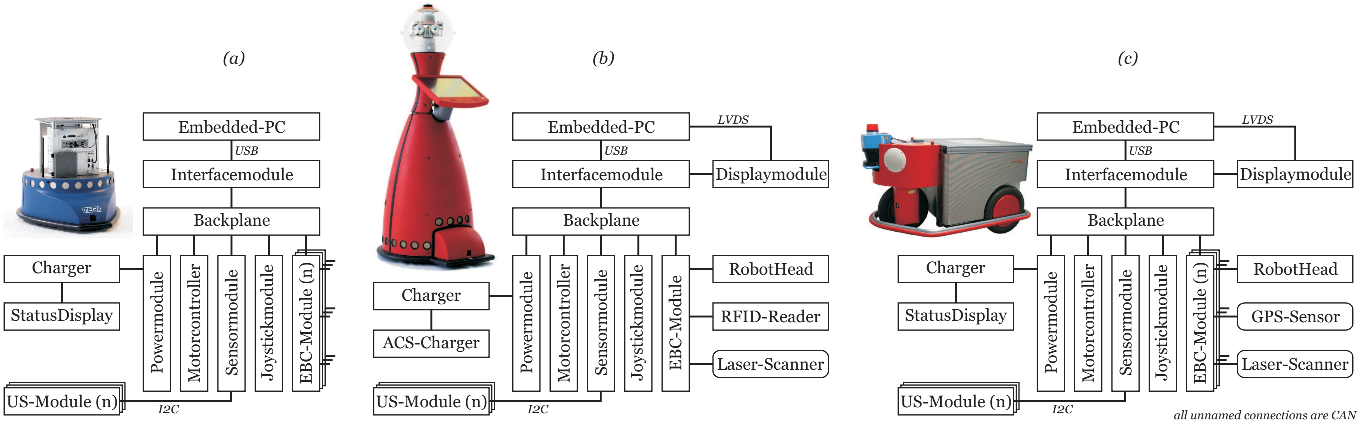

Fig. 4. The developed hardware architecture in the three example implementations; (a) shows the basic configuration in the research platform SCITOS G5, (b) shows the most complex implementation in the robot system SCITOS A5 for the application of a shopping-assistant robot system and (c) shows the configuration in the outdoor research platform MILVA. The units in the round borders are power supplied sensor systems and not a part of the hardware architecture.

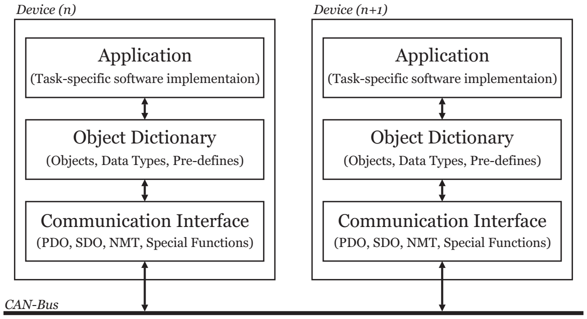

Fig. 5. Embedded software architecture of CANopen devices

be transfered is defined and accessible by the user application and the CANopen stack.

To transfer data over the CAN-bus, different transfer types are defined by the CANopen specification. First, the Network Management (NMT) services are used to set the actual working state of the modules and to initiate the reset or the re-initialization procedures. Furthermore, predefined communication objects are available to show emergencies or error states to other bus members, to synchronize different modules or to set data timestamps. Nevertheless, the most important parts in the communication of CANopen are the Service Data Object (SDO) and the Process Data Object (PDO). The SDO transfer is used to read and write variables in the object dictionary. The access to the objects is based on the address of the device, an index pointer and a subindex pointer to the dictionary. Since the information overhead is high, this transfer is usually applied for slow-changing or constant data values like configuration parameters (e.g., PID parameter) or slow sensor and actuator values (e.g., temperature values). The PDO transfer is optimized for a fast data transfer. Consequently, only the pure information is transfered in this mode and every data package has its own CAN-ID. Unlike the SDO transfer, the parameter of the data packages (ID, size, data composition) must be defined in the data source and the data sinks.

The access to different modules (nodes) is based on device IDs. The specification of CANopen allows up to 127 different nodes on one physical bus, which is sufficient for most applications. Although modules node-IDs can be set dynamically, we defined fixed IDs for singular modules and ID-areas for repetitive modules like external power supplies. This allows for an easier system configuration without any restrictions for the use in mobile robot systems.

C. *Power Supply*

Our new hardware architecture was designed to work with $12V$ and with $24V$ robot systems. To ensure a stable operation even at low voltage levels of the accumulators, all the modules are able to handle input voltages from $+6.0V$ up to $+40.0V$ and are accessible in this range over the CAN-bus. Thus, the system parameters and error states can be read even in low voltage modes. The necessary power levels are generated on each module separately. This local power supply concept minimizes the interactions between different modules because generated voltage peaks or noise (e.g., caused by the *MotorController*) will be eliminated by the power converter of the other modules before the disturbance can reach sensible circuits.

To save energy and to increase the working time of the system, almost all modules can be turned off by the software. Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFET) are placed in the lines of the power supply. The start-up of the robot initiates the *Charger* module, which activates a global power-up signal - like the ignition key in a car by starting the engine - and enables all MOSFETs on the modules. The module-specific software holds then the local power-up signal active. If a module is not longer needed, it can be completely deactivated by a SDO access. To wakeup the off-state modules, the *Charger* can re-activate the global power-up signal for a defined time period.

In addition to the control of the global power-up signal, the *Charger* has other tasks. This module controls the charging process of the accumulators and must avoid a deep discharge of the accumulators. For this reason, the charger is measuring the voltage levels all the time. If a critical voltage level is reached, the *Charger* will send PDO messages with the information that the robot has to be turned off immediately. If nothing happens, the *Charger* sends a SDO message after a configured time to the *InterfaceModule*, which will turn off the *Embedded PC* by using the connection of the power button. After a second configured time, the *Charger* will turn off all electronic modules in the robot system. Finally, the *Charger* will be turned off by a security circuitry and the robot will go in a zero consumption state to protect the accumulators.

To guarantee a long live time of the accumulators, the battery technology must be robust against cycle charging and high current discharges. We decided on lead-acid accumulators because this battery-technology is well qualified for these operation modes.

D. *Safety Mechanism*

Another main aspect of our hardware architecture helps to avoid application and adaptation mistakes of the platform by researchers and engineers and to avoid dangerous situation in applications with walk-in customers.

The implementation of the robot's power supply and the realization of the drive system are the two relevant aspects for a secure architecture. Both parts will be checked in detail by testing facilities like the German TÜV.

The user must not be able to access high voltage levels that are connected during the charging process of the robot. This includes considerations by the design of charger modules, the wiring and the mechanical realization of the platform. A quality aspect of the power supply is the protection of all accessible power lines against stressful use, like over-voltage, reverse powering and short-circuit.

The safety state of the driving system comprises a stop of the platform with applied brakes. The *MotorController* must enter this state as soon as any problem occurs. Therefore, this safety controlling device includes precautions for the detection of critical issues, e.g., broken wires to motors or collision sensors, disconnected plugs, stall modes of the wheels or burned fuses. These precaution functionalities are based on additional circuitries for a reliable detection. A software implementation would not be acceptable by industrial standards. Furthermore, the setting of the speed commands via a PDO access contains a time stamp. If data packages will not arrive in a defined time window or the information is not up to date, the *MotorController* will also stop the robot platform.

IV. EXAMPLE IMPLEMENTATIONS

In the following section, we present the successful adaptation to three different robot systems. We will describe the implementation in the robot base SCITOS G5 (Fig. 1), which was developed for the usage by research groups, the robot system SCITOS A5 (Fig. 6), developed as a mobile shopping-assistant system and the outdoor robot platform MILVA (Fig. 7) used in outdoor research projects.

A. *SCITOS G5*

The design of the SCITOS G5 platform focused on the flexible use for different research purposes. Consequently, the included electronic modules must be adaptable to various tasks, like sensor based navigation research, human-robot-interaction design, manipulator based applications or simple transportation tasks.

To achieve the highest flexibility, this platform is equipped with the base configuration of the new hardware architecture (Fig. 4a). Depending on the requirements of the application, necessary modules can be added for the final task. Moreover, an easy access to all provided interfaces of the hardware modules and the *Embedded PC* is possible. The availability of multiple *EBC-Modules* and the inputs and outputs of the *SensorModule* give additional ways to individualize the platform by adding user-specific devices.

Up to now, different research groups are working effectively with the modular concept of the robot base (e.g., [8]). In addition to the described extension modules, user-specific devices like camera systems, displays or interaction devices were added to the platform. Furthermore, developments of mobile industrial handling robots and transportation systems have started based on this platform.

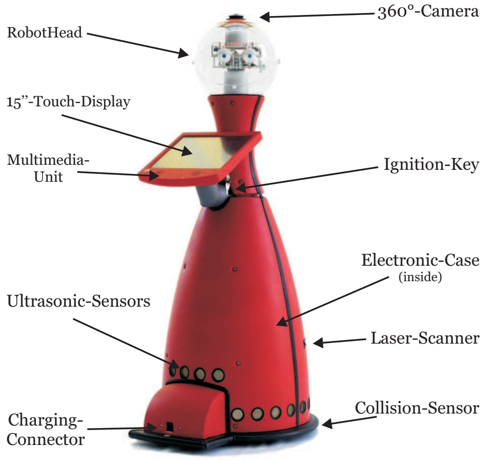

Fig. 6. Mobile robot system SCITOS A5. The size of the platform is $150 \times 74 \times 62 cm^3$ (HxLxW). The robot can drive up to $1.4 m/s$ and has a weight of approximately $75 kg$. The maximum working time with one battery charge is up to twelve hours.

B. *SCITOS A5*

The mobile robot system SCITOS A5 was developed as an interactive service robot [5], [9]. The characteristics of this systems can be summarized by a higher number of sensor systems (laser range finder, vision sensors), a higher computation power for the guidance of the robot's behavior and more interactive elements. The resulting configuration of the hardware architecture is shown in Fig. 4b. As a result of the difficult task of this system, the architecture is more complex and an *Embedded PC* with an Intel Core Duo processor $2.0 GHz$ is used.

The developed application shows that the hardware architecture is able to handle the needs of high-end-applications. Although the concept is still flexible and modular, the resulting production costs are suitable for this application. The power consumption of the hardware in combination with the power safe functionalities allow working times between ten and twelve hours without additional charging cycles. This follows the defined requirements. To further extend the working time, the additional charging module (*ACS-Charger*) is implemented.

C. MILVA

The third implemented solution is the mobile outdoor robot platform MILVA. This robot base was developed by mecos Robotics AG (Zurich, Switzerland) in the year 1994. The original hardware architecture was not able to manage modern research tasks and had to be exchanged.

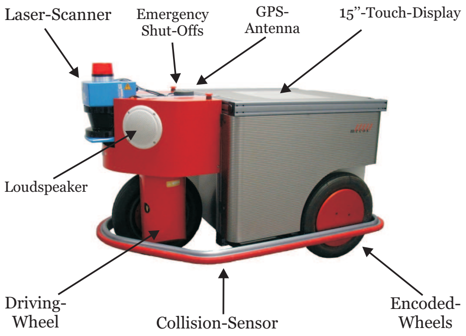

Fig. 7. Outdoor mobile robot system MILVA. The platform has a size of approximately $70 \times 110 \times 70 cm^3$ (HxLxW) and a weight of $119kg$.

We adapted our hardware architecture to the remaining parts of MILVA: the drive system, the incremental encoders and the closed bumper. Compared to the SCITOS systems, two changes were necessary to install the hardware into this outdoor platform. First, the capacity of the accumulators had to increase from $912Wh$ to $1560Wh$ because of the higher energy consumption of the drive system. Consequently, the *Charger* has to work with different parameters for charging and discharging, which is configurable via a SDO access to the corresponding values in the *Chargers* object dictionary. The second change was necessary because the driving system is based on two $48V$-DC-motors. An additional DC/DC-Converter for the generation of an $48V$ power supply and a modified motor controller had to be developed. Furthermore, in this realization the *SensorModule* is used to read the states of the incremental encoders and to calculate the odometry of the robot. This information is send by a PDO to the *Motor-Controller* to allow the computation of the PID-algorithms.

With the described realization, we were able to show that the hardware architecture was easily adaptable to a completely different robot system.

V. CONCLUSIONS AND FUTURE WORKS

The requirements for the development of a highly adaptable hardware architecture were defined by different application areas. In consideration of these requirements, we developed a flexible architecture and designed the required hardware modules. We implemented the system in three different robot platforms for the use in various applications. We were able to show the successful realization of the architecture in the practical use in robot systems for indoor and outdoor mobile robot research [10], [11], industrial transportation systems and a shopping assistant. We verified the functionality of the modules and passed the verification process of the German TÜV.

In the future, new modules will be developed to open up new applications or to adapt the hardware architecture to other robot platforms. The increasing production of the SCITOS G5 and the SCITOS A5 platforms will reduce the production costs of the modules and will make these modules to off-the-shelf products.

VI. ACKNOWLEDGMENT

This work was done in strong cooperation between the company MetraLabs GmbH and the Neuroinformatics and Cognitive Robotics Lab of the Ilmenau Technical University. Especially, we thank Christian Martin and Andreas Bley from MetraLabs GmbH and Hans-Joachim Böhme, Christof Schröter, Steffen Müller, and Alexander König from the Ilmenau Technical University for the support by the formulation of the requirements, the enriching discussions and the assistance during the verification of the functionalities. The project was funded by the Thuringian Ministry Of Science And Culture (TMWFK) and the Thuringian Ministry of Economy, Technology, and Labour (TMWTA) in Germany.

REFERENCES

[1] R. D. Schraft, M. Haegele, and A. Breckweg, Eds., *World Robotics 2006*, 2006.
[2] E. Park, L. Kobayashi, and S. Y. Lee, "Extensible hardware architecture for mobile robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'05)*, April 2005, pp. 3095–3100.
[3] K. Regenstein, T. Kerscher, C. Birkenhofer, T. Asfour, J. Zllner, and R. Dillmann, "A modular approach for controlling mobile robots," in *Proc. of CLAWAR2007, 10th International Conference on Climbing and Walking Robots*, May 2007.
[4] (2007, August) Neuroinformatics and Cognitive Robotics Lab, Ilmenau Technical University. [Online]. Available: http://wcms1.rz.tu-ilmenau.de/fakia/SERROKON-H.3538.0.html?&L=1
[5] H.-M. Gross and H.-J. Boehme, "Perses - a vision-based interactive mobile shopping assistant," in *Proc. IEEE Int. Conference on Systems, Man and Cybernetics (IEEE-SMC 2000)*, 2000, pp. 80–85.
[6] H. Zeltwanger, *CANopen*, 2007.
[7] (2007, August) CAN in Automation (CiA). [Online]. Available: http://www.canopen.org
[8] L. Spacek and C. Burbridge, "Instantaneous robot self-localisation and motion estimation with omnidirectional vision," *Journal of Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 667–674, 2007.
[9] C. Schroeter and H.-M. Gross, "Memory-efficient gridmaps in rao-blackwellized particle filters for SLAM using sonar range sensors," in *Proc. 3rd European Conference on Mobile Robots (ECMR'07)*, 2007, pp. 138–143.
[10] M. Schenderlein, K. Debes, A. Koenig, and H.-M. Gross, "Appearance-based visual localisation in outdoor environments with an omnidirectional camera," in *Proc. 52nd Int. Scientific Colloquium Ilmenau*, 2007.
[11] S. Mueller, E. Schaffernicht, A. Scheidig, H.-J. Boehme, and H.-M. Gross, "Are you still following me?" in *Proc. 3rd European Conference on Mobile Robots (ECMR'07)*, 2007, pp. 211–216.