Educational robotic platform ”Zero2Nine” for autonomous navigation and tracking based on imaging sensor systems

A. Linz, A. Ruckelshausen
University of Applied Sciences Osnabrück, Germany

Abstract

Robotic platforms are more and more used at engineering educational institutes for teaching various theoretical and practical subjects, such as electronics, computer science, mechatronics, sensor technologies, image processing or artificial intelligence. However, the development of the mechanical platforms as well as the integration of sensors and system technology often limits the experience of combining theoretical concepts with practical experiences. In this paper a commercially available platform is combined with different smart imaging sensor systems and a corresponding system technology. The resulting educational robotic platform “Zero2Nine” allows navigation, tracking and detection applications based on individual imaging devices or sensor fusion algorithms. The name of the robot is related to an application example, the detection of numbers in outdoor environments. The modular system, the robustness and a short getting-started period are important aspects.

As a vehicle the commercially available Fraunhofer VolksBot RT4 platform is applied, which is equipped with 2 Maxon-motors and a motor controller. As imaging sensor systems a color smart camera, a laser scanner and a 3D-Time-of-flight camera are integrated, OpenCV as a computer vision library can be used. The system architecture is based on a PC with Linux as operation system and Ethernet with a standard TCP/IP protocol, thereby implementing a high flexibility with respect to the integration of additional components.

The technology is used for teaching, laboratory experiments and in particular for student projects in the Master and Bachelor modules, such as “Imaging sensor systems”, “Sensor systems”, “Optoelectronics” and “Seminar Mechatronics”. The robot is a helpful facility, especially for “advanced lab”, where the students work more intensive on a subject e.g. a smart cam or a laser range finder. It has also been a powerful starting tool for student groups participating robot competitions (example: “International Field Robot Event”). The students are highly motivated and this educational option has a potential with respect to increasing knowledge and practical experience together with team working. Moreover the robot is demonstrated at fairs, examples are “Agritechnica” or “IdeenExpo”.

Keywords

Autonomous mobile robotics, Imaging Sensor Systems, University Education

1 INTRODUCTION

Ever more, robots will be present in our daily live. First commercial products are lawn movers or vacuum cleaners (irobot, 2011). Service robots like PR2 (Meeussen et al. 2010) or field robots like BoniRob (Ruckelshausen et al. 2009) are in a development state, but not too far away from a product. At universities robotic platforms are introduced for teaching interdisciplinary aspects of mechanics, electronic and computer sciences. There are a lot of robot platforms with its advantages and disadvantages on the market available. Low cost robots like Lego Mindstorm (Lew et al. 2010) or Asuro (asurowiki, 2011) from the DLR can be afford from the students to get first experience. They are also used in robot competitions for children and pupils. A robot build for outdoor use like optoMaizer (Klose et. al 2006) is more expensive, unique and has fixed soft- and hardware on board. However, VolksBot is an expensive but high expandable and robust platform. That is the reason why the University of Applied Sciences Osnabrück purchased 4 of them for different labs in order to get a technically interdisciplinary approach to autonomous robots. The labs have their focus on Embedded System Engineering, Computer Engineering, Mechatronic System Engineering and Sensor Systems. The focus of this paper is on a Volksbot equipped with optoelectronic sensor systems.

2 CONCEPT

The concept of the mobile robot is to keep the system as modular and flexible as possible. This concerns software, hardware and mechanical construction likewise. Flexibility is given by Ethernet (TCP/IP) as bus system. Sensors, intelligent sensor systems, actuators, controller boards and PCs are interacting on this bus. It is possible to connect to each of this part from "outside" via the WLAN-Bridge or an Ethernet cable. Sensors and actuators can be plugged or unplugged without an influence on the whole system. An example for high flexibility is, that the software can run somewhere in a virtual machine (i.e. Virtual Box with Linux), which connects via LAN or WLAN to all sensors and actuators and controls the robot. The same software can be run on a microcontroller board or a PC, which is physically installed on the robot.

The software on the robot starts a server, which exchanges most data in plain ASCII code. If the connection to the server is established, the client will be able to control the robot or read out sensor data with some simple commands. The easiest way is to connect for example with "PuTTy" (a Telnet client, *Figure 4*) in raw mode and type in a command. A more sophisticated way is to write a program in a high level programming language like JAVA, C#, Perl etc. or use tools like MATLAB or LabVIEW to exchange data and control the robot.

3 HARDWARE

The robotic platform VolksBot is a construction kit developed and distributed by the Fraunhofer Institute IAIS (Intelligent Analyse- und Informationssysteme; volksbot, 2011). The platform is especially constructed for "Physical Rapid Prototyping" of mobile service robots. The modular "Item-system" and 2 Maxon DC motors with 150W makes it extremely robust and reliable. It is designed for outdoor and indoor use respectively.

There are several types of the platform available. Here the 4-wheel version "RT 4" with 2 motors and differential steering (*Figure 1*) is be used. Short datasheet of the Robot is:

- Dimension: 540 x 500 x 260 mm (l x w x h)
- Wight: 13kg (without PC and accumulators)
- 260 x 85 mm wheels
- Gear ratio: 43:1
- Maximum velocity: 2.2 m/s
- Maximum load: 40kg

The basic set of the RT 4 includes a 3-channel motor controller with a RS232- and a CAN-Bus Interface. The RS232-Interface is well documented, so that a self-developed software interface for driving and steering can be written. The motor and the other electronic parts are powered by 2 12V/7.5Ah lead batteries.

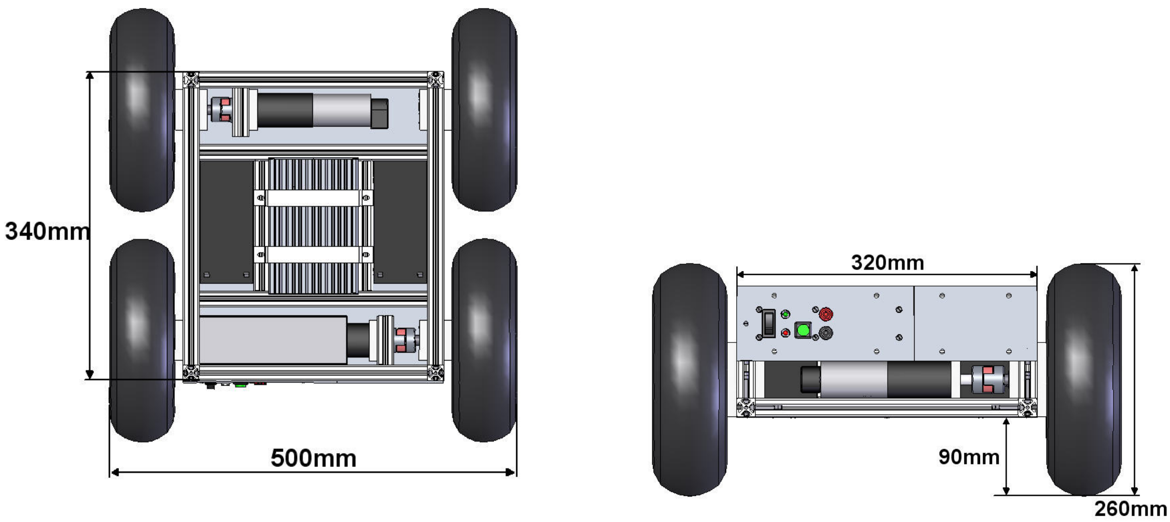


Figure 1: Volksbot® RT 4 from the top and from behind. (volksbot, 2011)

Recently installed sensor systems are:

- The **laser scanner** LMS100 from Sick. It is a very robust and reliable range finder which can scan a maximum distance of 32m in a range of 270° with a resolution of 0.5°. The sample rate is 50Hz.
- The **Time of Flight 3D camera** pmd3D from IFM has a resolution of 64 x 50 and gives back the distance up to 6.5m, and the intensity for every pixel.
- The **smart camera** leanXCam from SCS-Vision has a resolution of 752x512 and is equipped with a 500MHz Blackfin processor on board. There are 2 digital in- and outputs and an RS232 interface, which can be used for debugging purposes. Maximum frame rate of the camera is 30fps. The OpenCV library can be used for image processing.
- The **smart camera** NI1742 from National Instruments is a more professional camera system for industrial applications, equipped with a 533MHz PowerPC from Motorola. The resolution is 640x480. It has a maximum frame rate of 60fps.

All Sensors have an Ethernet interface. The motor controller and the display are connected over a device server (*Figure 2*), which converts TCP/IP to RS232 and back. There are 2 ports left for other devices i.e. GPS receiver.

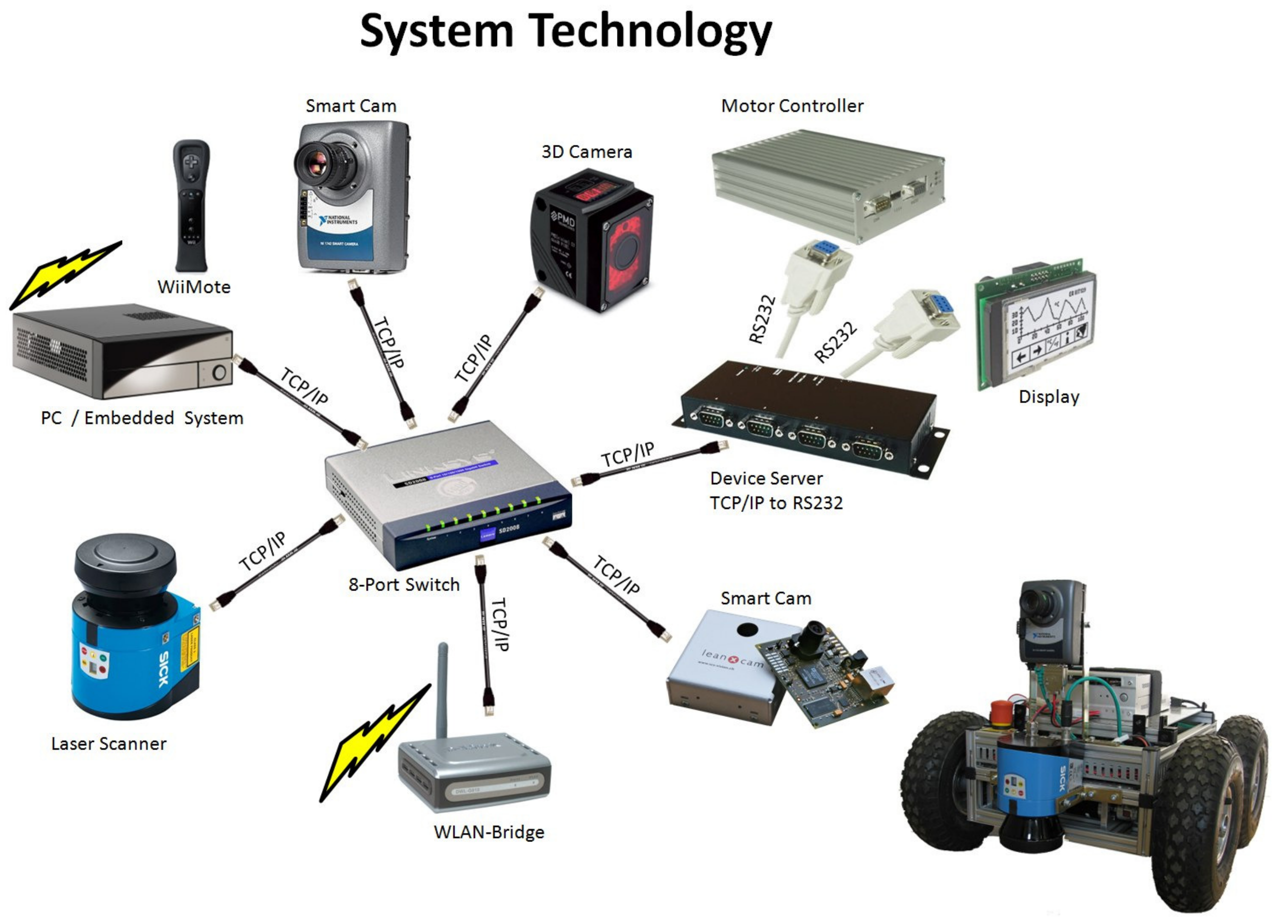


Figure 2: System Technology

4 SOFTWARE

The software is split in 2 parts, the software part running on the robot (or somewhere in the network) and the software part connecting over the server (described later) to the robot. The first part includes all drivers for sensors and actuators. The second part is used for external visualization and processing of sensor data.

The software on the robot is written in C++ using the GNU C++ compiler. It runs on a 900MHz Mini-ITX with an Ubuntu Linux version 10.04. The development environment is installed on a virtual machine with the same Ubuntu version as on the robot. Thus, it is possible to compile the program on the VM, copy it to the PC on the robot and run it there. Connection from the VM to the robot can be established over VNC or SSH. Because the hardware communicates with TCP/IP, there is no difference if the program is running in the VM or on the robot (as long as the physical connection is fast enough). This makes software developing and debugging much easier.

The concept of the software is that every driver has its own class and runs in its own thread. Data is exchanging over a special class called "*SensorData.cpp*". After a thread is started, it tries to connect to the hardware, initializes it with settings out of the class "*Settings.cpp*" and writes data from the sensor to the data class. The "*WorkingClass.cpp*" class then takes the needed data from the data class, process it and sends drive commands, with the help of the class "*VMC.cpp*", to the motor controller.
The class "*ToolBox.cpp*" supports some functions for reading and converting sensor data and controlling the robot, i.e. "turn 120°" or "drive forward with 25% of maximum speed". *Figure 3* shows all classes used in the program "*SensorServer*".

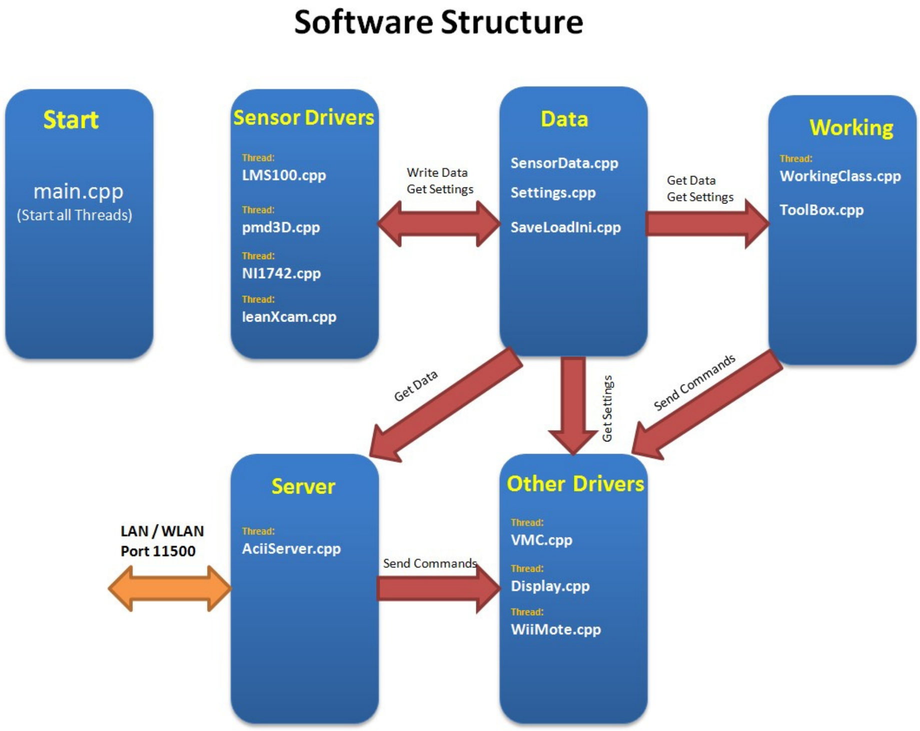


Figure 3: Software Structure

The class "*AsciiServer.cpp*" is also running in a thread and has the special function to communicate over a TCP/IP connection with external applications. The thread opens a port and waits for connection. A connected client can send commands in plain ASCII code. If the command is ok, the server returns an acknowledge "ACK". Otherwise it sends back a "no acknowledge" ("NCK"). *Figure 4* shows a PuTTy-Connection to the server with some commands which can be used by the client.

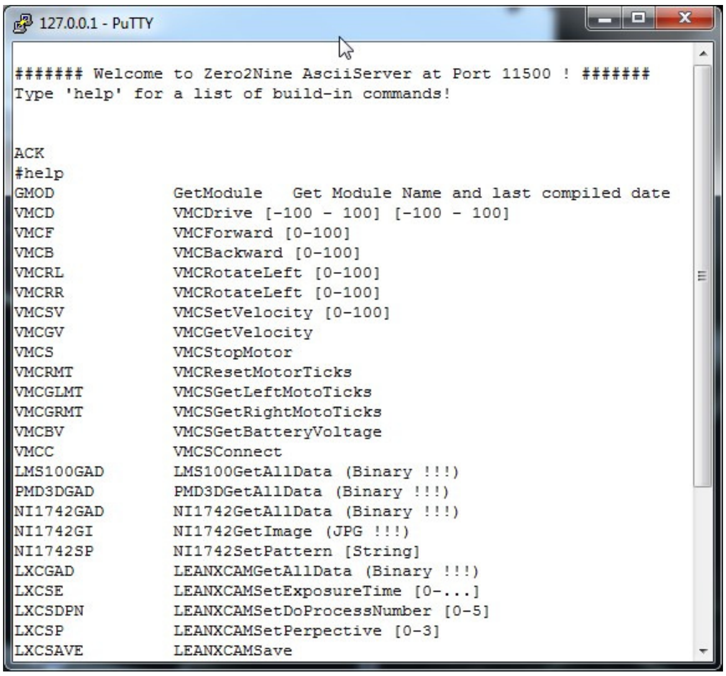
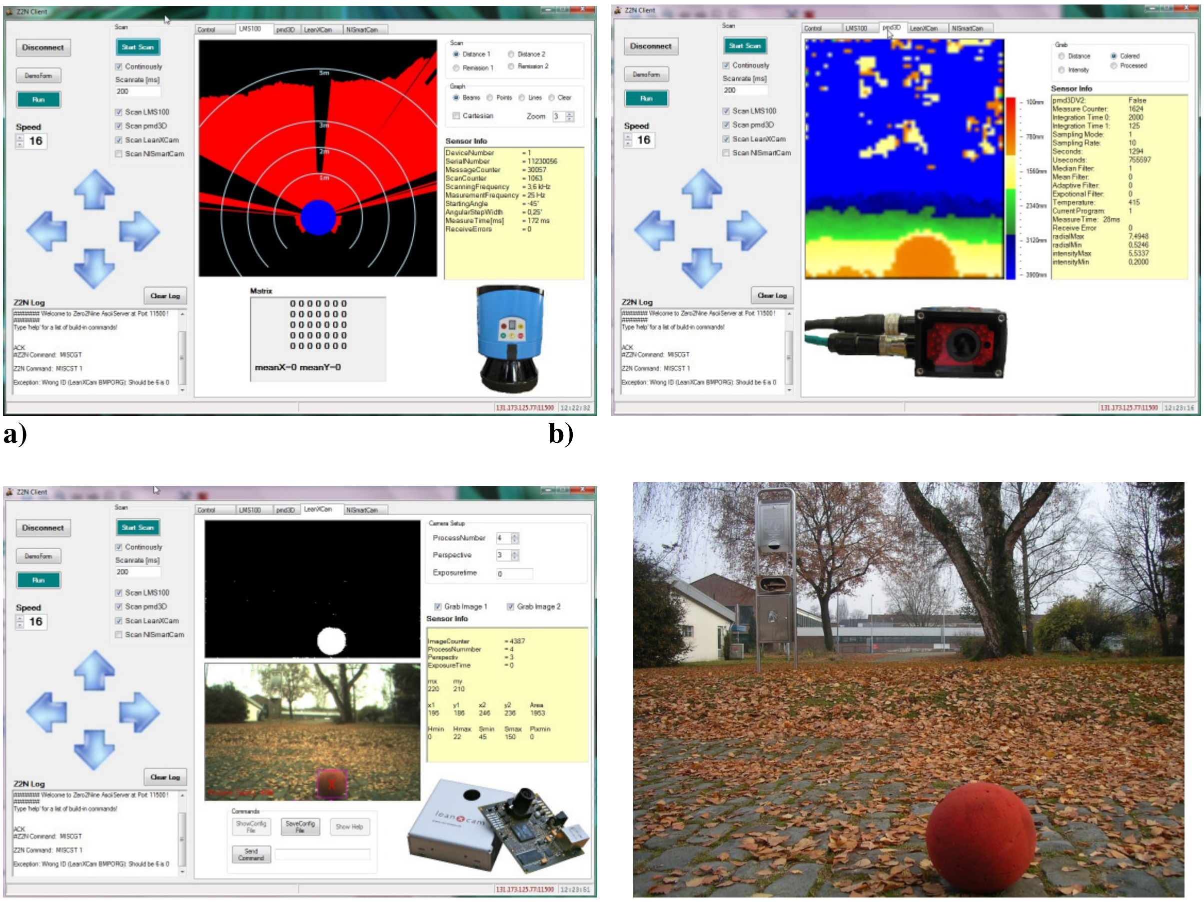


Figure 4: Connection to the "AsciiServer" with PuTTy.

A client can be a command line tool like PuTTy, it can also be a program written in a high level programming language. Operation system and programming language are of no relevance as long as it is possible to establish a TCP/IP connection. Sensor data returned from the server are binary. A data packet starts with *BinaryDataPacketID*, *BinaryDataPacketSize* and *BinaryDataPacketFingerprint* followed by a C-Structure including different types of data. This makes transmitting of the data more easy and secure. However the C-Structure must be known very well, to extract the data in the right way. There are some helpful clients written in C# which are used for development and demonstration. Every sensor has its own client. The program "Z2N Client" is a collection of all clients together.



a) b) c) d)

Figure 5: Z2N Client with the visualization of a) the laser scanner, b) 3D-camera, c) leanXcam with color tracking and d) the original sight taken by a photo camera.

In *Figure 5* a demonstration of the robot driving outdoor and tracking a red ball is shown. The laser scanner and also the 3D-camera give back the distance of the ball and other obstacles in the environment. The leanXCam tracks the color and returns the position of the detected object to the client. If one of the blue arrows will be pressed, the client sends a drive command to the robot i.e. turn left.

5 EDUCATIONAL ASPECTS

5.1 Lectures and laboratory experiments

The robot is used in lectures for explaining sensor systems and sensor fusion. It is helpful to get a "feeling" what a sensor can "see" and where are the limits of it. It is also employed in laboratory experiments, where students can concentrate on one sensor system, sensor fusion or navigation of a mechanical platform. The software structure allows the students a very short getting-started period. The only requirement is to have experience with C++ and maybe other high level programming languages.

In the special format called “advanced lab”, students just work out a normal lab experiment e.g. using some function of openCV (an image processing library) on the leanXcam. Second step is to work out a more advanced exercise within the same experiment, given by the supervisors. As a third part the students than conceive an experiment selected by themselves. For example, this could be a tracking experiment with the leanXcam on the robot.

The “advanced lab” results in a higher motivation and a deeper understanding of the subject to the students. An essential part of the exercise is the presentation and discussion of the experimental results. So, the students can get an insight view to the other experiments in the lab. Table 1 gives an overview of modules where the platform is used.

Modules	**Courses**	**Learning objectives**
Optoelectronics	Bachelor Electrical Engineering	Understanding optical sensor systems, integration in mobile platforms
Imaging Sensor Systems	Master Automation Systems	Image processing, understanding image sensor systems, programming, integration in mobile platforms
Seminar Mechatronic	Master Mechatronic System Engineering	Understanding sensor systems, navigation, programming, integration in mobile platforms
Sensor Systems	Master Mechatronic System Engineering	Understanding sensor systems, programming, integration in mobile platforms
Student projects	Bachelor and Master Courses	Topic defined by the supervisors, by students or proposed by companies

Table 1: Modules and Learning objectives

5.2 Student projects

Student projects are running over the whole term with small teams of typically three people. Thus, more complex issues like navigation through rows, object recognition or SLAM (Simultaneous localization and mapping) techniques can be worked on. Presentation and discussion of the results with the other students and the supervisor is also content of the project.

5.3 Bachelor and master thesis

Another field of application are bachelor and master thesis, where students evaluate new sensor systems i.e. Kinect-Camera from Microsoft. Because the camera has an USB-interface, the sensor software can run on a PC or a notebook, which navigates the robot over the “ASCII-Server”. *Figures 6* shows an example of a bachelor thesis, where a student programmed a 3D-ToF-Camera from PMD with an USB-interface. For evaluation and development he used MATLAB running on a notebook. MATLAB processes the image data, establishes a TCP/IP connection to the robot and sent the commands for the navigation in plain ASCII-code.

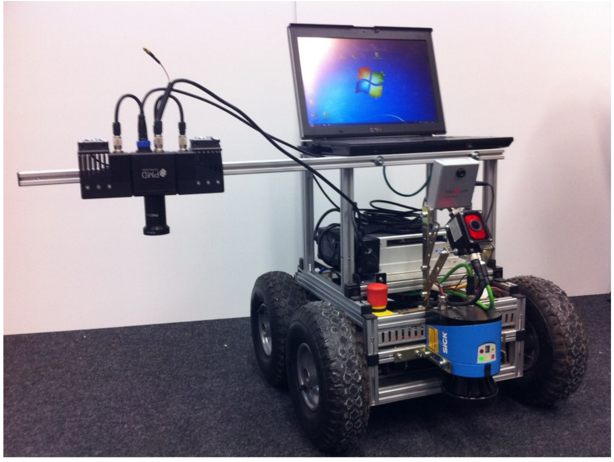


Figure 6: Zero2Nine equipped with a 3D-ToF-camera on the left side. The camera is used for soil properties detection.

5.4 Student competitions

Students use the system as a powerful starting tool for participating at robot competitions. Examples are the yearly International Field Robot Event (Mueller et al. 2006). The robots have to navigate between maize rows as fast and smooth as possible. Moreover, other tasks such as weed control are included. The development of robots and algorithms for outdoor applications results in high-level practical experiences for the students. Teamwork is an important aspect in the competition teams. They have to share different tasks like programming a sensor, preparing mechanical parts or create posters for public relations. There is also a team captain with the function of a supervisor. This is a perfect preparation for professional live. Moreover the students document their work in proceeding of the event (example: Feldkämper et al. 2010).

a) **b)**

Figure 7: a) Field robot Team "optomaize Prime" at the Field Robot Event 2010 in Braunschweig, Germany, b) Zero2Nine" at the Ideen Expo 2011 in Hannover, Germany

5.5 Fairs and demonstrations

The robot ran on several fairs like Agritechnica 2009 (go.amazone, 2011) or Ideen Expo 2011 (hs-osnabrueck, 2011) (Figure 7b) in Hannover for 7 and 10 days respectively. This showed the robustness of the system, because there haven`t been any mentionable hardware or software problems. The integrated Wii-control attracts children (and grownups) because they can control and interact with the robot. If there is an obstacle in front of the robot, they getting a vibration feedback. This is a good starting point to explain the sensors and the algorithm.
There are a lot of guidance tours in the lab for school classes or other research institutes, where Zero2Nine is used as a demonstration tool. Especially the school classes are exited and have a lot of question to the function of the robot.

6 CONCLUSION AND OUTLOOK

The students have a robot platform for evaluating, understanding and programming imaging sensor systems. There is a fast learning curve and a high motivation, because they can test and see the results immediately. The platform is well tested in a lot of projects and at robot competitions. Of course a short introduction to soft- and hardware is needed, but usually there is no more need for support.
A very simple and robust algorithm for driving around with avoiding collision and the visualisation of the sensor data makes Zero2Nine a nice tool for demonstrations in the lab and on fairs. People are getting a better understanding how an autonomous system can navigate by "itself".
Next development step is to integrate additionally ROS (Robot Operation System, ros, 2011). ROS is a widely distributed open source operation system especially for autonomous mobile robots. A lot of driver for different sensors and motor controllers and also more complex software like Kalman filter or SLAM-algorithm just exist. ROS seems to become an important tool for robot development.
Simulation is another topic. First experiences have been made in the lab with Microsoft Robotic Studio (Tsukor et al. 2011). Integration of the platform in MRST (microsoft, 2011) or GAZEBO (playerstage, 2011) is one of the next steps.

REFERENCES

Feldkämper, N., Hufendiek, H., Jahn, H., Kampmeyer, C., Kerssen, S., Lemke, C., Mosler, D., Rave, P., Simon, S. Tilneac, M., Westerhoff, J. Wunder, E.: **"***Team Optimaizer, Proceedings of the 8th Field Robot Event 2010"*, urn:nbn:de:gbv:084-11100508057, p. 185-201, 2010,
URL: http://www.digibib.tu-bs.de/?docid=00041345.

Griepentrog, H.-W., Ruckelshausen, A., Jörgensen, R.N., Lund, I.: *Autonomous systems for plant protection, in „Precision Crop Protection – The Challenge and Use of Heterogeneity*", pp. 323-334 (Editors Oerke,E.-C., Gerhards, R., Menz, G., Sikora, R.A.), Springer, 2010.

Klose, R., Linz, A., Ruckelshausen, A.: *"Robotic Platform for Remote Control and Teleservice in Agicultural Environments"*, Proceedings of CIGR EurAgEng/VDI-MEG, 2006.

Lew, M.W., Horton, T.B., Sherriff, M.S.:*"Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course*", 23rd IEEE Conference on Software Engineering Education and Training (CSEE&T), 2010.

Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., Marder-Eppstein, E., Muja, M., Eruhimov, V., Foote, T., Hsu, J., Bogdan Rusu, R., Marthi, B., Bradski, G., Konolige, K., Gerkey, B., Berger E.:*"Autonomous Door Opening and Plugging In with a Personal Robot*", IEEE International Conference on Robotics and Automation (ICRA), 2010.

Mueller,J., Walther,S., Boettinger,S., Ruckelshausen,A., van Henten,E.J.: *„Field Robot design contest - a showcase in Agricultural Engineering education"*; in: Kataoka T, Noguchi N, Murase H, editors. 3rd IFAC International Workshop on Bio-Robotics, Information Technology and Intelligent Control for Bioproduction Systems 9-10, September 2006. Sapporo (Japan); p. 276-281.

Ruckelshausen, A., Biber, P., Dorna, M., Gremmes, H., Klose, R., Linz, A., Rahe, R., Resch, R., Thiel, M., Trautz, D., Weiss, U.:"*Boni Rob: an autonomous field robot platform for individual plant phenotyping"***,**Joint International Agricultural Conference, 2009.

Tsukor, V.; Klose, R.; Ruckelshausen, A.: *"Multi-sensor simulation method for outdoor plant phenotyping based on autonomous field robots"*, Proceedings CIGR 2011, to be published.

http://www.irobot.com/de/store.aspx, last accessed on November 15, 2011
http://www.asurowiki.de/pmwiki/pmwiki.php, last accessed on November 15, 2011
http://www.volksbot.de/rt4-de.php, last accessed on November 15, 2011
http://www.hs-osnabrueck.de/722+M5ea081d3d65.html, last accessed on November 15, 2011
http://go.amazone.de/index.php?news=26, last accessed on November 15, 2011
http://www.ros.org/wiki/, last accessed on November 15, 2011
http://www.microsoft.com/robotics/ , last accessed on November 15, 2011
http://playerstage.sourceforge.net/ , last accessed on November 15, 2011

Contact:
Dipl.-Ing. (FH) / B.Eng., Andreas Linz
Faculty of Engineering and Computer Science, University of Applied Sciences Osnabrück
Albrechtstr. 30, D-49076 Osnabrück, Germany
phone: +49 (0541) 969-2978
fax: +49 (0541) 969-2235
Email: a.linz@hs-osnabrueck.de