The Herd of Educational Robotic Devices (HERD): Promoting Cooperation in Robotics Education

Andreas Bihlmaier[1], Michael Vollert[1], Heinz Wörn[1]

[1]Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics and Robotics (IAR), Intelligent Process Control and Robotics Lab (IPR)
http://herd-project.org
{andreas.bihlmaier,woern}@kit.edu

Abstract. Teaching robotics in a high school environment can not only benefit the students because they acquire technical skills, they can also learn a lot about purposeful cooperation. We developed simple, small and cheap *swarm* robots designated for high school education that allow to broaden robotics experiments in interesting ways. The paper describes design considerations, their implementation in hardware and software, additional tools important in the high school setting and presents qualitative results from evaluation of the prototype swarm in different schools and different school classes.

Keywords: HERD, Swarm Robotics, High School, Teaching with Robots.

1 Introduction

Working hands-on with robotics arouses most high school students' interest to a much greater extent than pure computer science lessons – at least this is what we experienced in our initial evaluations (cf. Section 4) of the robotics platform presented here. The interdisciplinary field of robotics addresses a diverse set of interests and skills, from the more theoretical STEM disciplines, mathematics, algorithmics, electrical and mechanical engineering, to the rather hands-on work found in programming and making. The important initial difference in student interest seems to be related to the immediate experience of real-world consequences that result from changing some numeric value in the robot's code. Without getting into the discussion about embodiment, it is simply a more holistic experience when a real, physical robot moves about instead of some pixels on a computer screen. However, most (high) schools have to cope with scarce resources and either they cannot afford or do not have the room available to work with the kind of robots used in higher education institutions.

We developed the Herd of Educational Robotic Devices (HERD), simple and small *swarm* robots with extended capabilities, which find themselves in an affordable price segment at a designated price of 50 EUR [1]. A general inspiration for the HERD platform was the Wanda robot [2] used in academic research, however, with a much higher price tag. Our design of hardware, software and supplementary tools was carried out with a high school teaching objective and hence happened in close

Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics &
5th International Conference Robotics in Education
Padova (Italy) July 18, 2014
ISBN 978-88-95872-06-3
pp. 110-117

consultation with high school teachers. We implemented the resulting design guidelines in the form of 15 HERD robots (cf. Figure 1) together with an application programming interface (API), which is adjusted to beginners in programming, yet at the same time capable of being used by more advanced students. In combination with a custom Live System (DVD or USB stick), the HERD robots were used in different schools in grades 8 to 11. The students' programming skills ranged from students who never had an introduction to programming up to ones with one or two years of Java programming experience.

Fig. 1. A swarm of prototype HERD robots is shown. Each robot has a size of 8x12x5 cm. All robots within direct line of sight are able to communicate by infrared senders and receivers.

The following section will detail the considerations that we arrived at during development of our HERD robotics platform and that are much more general than the specific - open hardware and open software - prototype implementation of the robots, described in Section 3. Having described the technical side of teaching with robots, Section 4 summarizes our experiences and that of the students who used the HERD platform either as part of a regular computer science course or during school project days. In the final section, we discuss findings and consequences of the school evaluations and provide an outlook about current developments and the future of HERD.

2 Design Considerations

There are three basic parts to a robotics platform for education: robot hardware, robot software and supporting tools. Each part has specific requirements due to the boundary conditions of high schools: students as users, teachers as advisors and administrators and schools as budget and staff limited customers.

2.1 Hardware

Capabilities. Robots have to physically interact with the world in some way. The simplest way in terms of hardware and control requirements is that the robot is mobile, i.e. it is at least able to drive on a flat surface such as a table.
In order to interact with the world in a more interesting manner, a robot requires sensors to incorporate some relevant states of the world into its actions. Due to the fact that most schools do not have room for a dedicated robotics lab, the robot's world should only consist of elements that can easily be set up and taken down for course hours. At the same time, the sensors should be cheap, their readings easy to interpret, intuitive to act on from a student programmer's point of view and still allow the implementation of interesting robot behavior.
It should be possible for the robots to interact in order to facilitate cooperation between the students and to enable interesting experiments without a robot arena. We postulate the equipment of the robots with means of swarm communication, i.e. contactless, spatially local data exchange capabilities, to meet this requirement.
Extensibility by simple hardware additions such as a better distance sensor, a display or custom electronics should be provided through an extension port, similar to Arduino [7] shields.

Manufacturing and Costs. Because we cannot directly influence the volume of produced robots to profit from bulk production costs, the key to affordability is robot design. Simple mechanics with a minimum amount of custom parts allow to keep the robot cost low and they can even increase robot reliability – something absent simply cannot break. If the electronic printed circuit boards (PCBs) can also serve as (part of) the mechanics, costs can be further reduced. All electronic components should be readily available and remain available for at least a few years. Nevertheless, owing to hardware interdependencies, it is beneficial to stay behind the state of the art components. A new high fidelity sensor requires a newer, more powerful microcontroller, which in turn often requires finer PCBs and a higher capacity battery that might increase the total weight of the robot, again necessitating adjustments such as larger motors.
The robot cost drops significantly, if it can serve multiple purposes and the school can buy the robot instead of some other hardware. For example, if the robot's microcontroller can be used for non-robotics tasks, such teaching assembler programming.

2.2 Software

API. Depending on the specific robotics aspect to be taught and the students' specific prior knowledge, the robot's API has to provide different levels of abstraction and require different subsets of the programming language syntax. If the lesson is about basic programming constructs, the robot API functions should be useable as black box one-liners. On the other hand, if more advanced students program a swarm search and gather algorithm, the API should provide advanced language features such as pointers.

Bootloader. To render swarm experiments in school lessons realistic, it must be possible to program all robots simultaneously without manually attaching each one to a computer. One possibility is to utilize the robots' means of swarm communication together with a write protected bootloader to wirelessly broadcast code from one robot to all others.

2.3 Live System

Most high school IT infrastructures are rather diverse and it is often difficult for a teacher to install and maintain additional software. Therefore a live system, on a DVD or USB stick, significantly eases teaching robotics in a school context. At the beginning of the lesson all computers are rebooted to the live system, robotics programs are written, complied and transferred using the live system's IDE, afterwards all computers are rebooted to their installed system.

3 Implementation and HERD Prototype Swarm

This section describes how we implemented the general design considerations in form of the HERD robots.

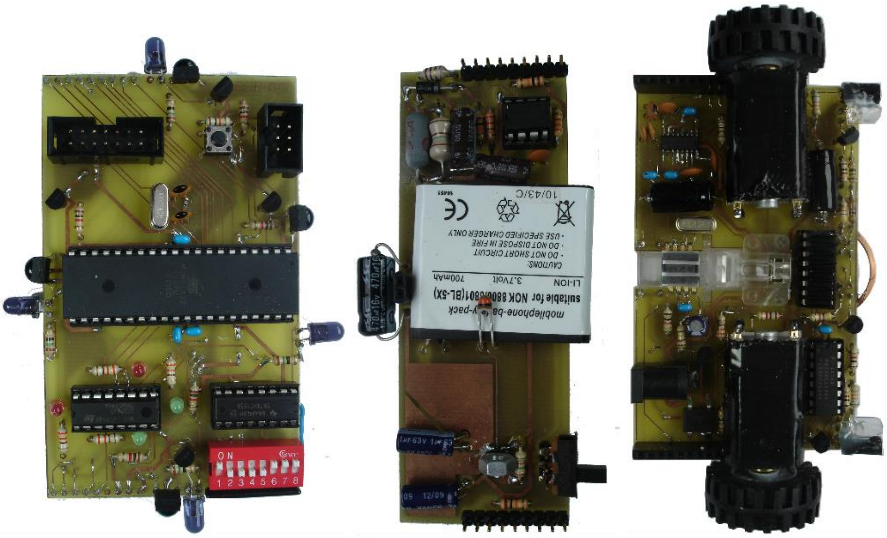

Fig. 2. The three PCBs which make up the hardware of a single HERD robot are shown. From left to right the front side of the top, middle and bottom PCB is pictured.

Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics &
5th International Conference Robotics in Education
Padova (Italy) July 18, 2014
ISBN 978-88-95872-06-3
pp. 110-117

3.1 Hardware

Each HERD robot consists of three printed circuit boards (PCBs), shown in Figure 2, which fit together on a single PCB eurocard (160x100 mm). They are cut out, populated and assembled through pin headers, which provide the mechanical and electronic connection (cf. Figure 3). The robot functionality is distributed to the top, middle and bottom boards, which will be described in the following. Only the most important technical properties are provided here, all low level details, e.g. port expansion, multiplexing and electronics design are left out.[1] All layouts are available as open hardware and can be modified with the open-source EDA suite KiCad [5].

Top. The top board contains an ATMega32 microcontroller, which controls all high and low level tasks. There are DIP switches to encode the robots swarm ID, to start the swarm bootloader and for user specific purposes. Four user controllable LEDs provide feedback about the robot state. One infrared (IR) sender and transmitter on each robot side provides local, line of sight based swarm communication.

Middle. The middle PCB contains all power related components. A Li-Ion battery as the ones found in (old) mobile phones is used together with a battery charger and a step-up converter to provide a regulated 5 V power source.

Bottom. The bottom board holds the differential drive together with all necessary electrical components. In order to provide more accurate movement, an optical mouse sensor is used in addition, measuring the robot's traveled distance. Furthermore, there are three different kind of sensors integrated: First, a simple distance sensor. While the robot is moving, blue LEDs are flashed and the reflected amount of light is measured. The intensity is proportional to the distance to collision objects. Second, two reflective sensors measure the ground reflectance. These measurements allow the robot to detect an abyss and stop before plunging down. The sensor can also differentiate between a white tabletop and a black line marked on top of it. Third, a RFID sensor is integrated - by far the most modern chip on the robot - which can detect RFID labels attached to objects or the table surface. This sensor is important because it allows to reliably recognize objects and positions, through the IDs stored in the RFID tags.

Extensions. An extension port is provided on the top board. Currently, there are already a few simple extension boards available such as an LC display, a serial port, multiple LED and segment displays. Since the extension port provides power and directly exposes a few microcontroller pins, custom extensions are easy to implement – both hardware and software wise.

[1] All the technical details can be found at http://herd-project.org/wiki/Hardware.

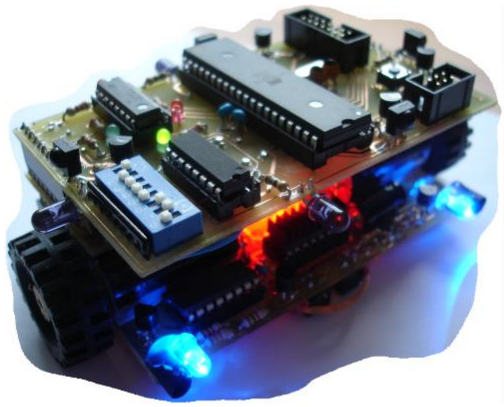

Fig. 3. The image above shows an assembled HERD robot. The three PCBs shown in Figure 2 are mechanically and electronically connected by pin headers. The blue LEDs are used for collision detection and avoidance. The mouse sensor used for position sensing is the source of the red glow.

3.2 Software

API. The API for the HERD robots uses the C programming language. We deem it is beneficial to use a “real” programming language instead of a domain-specific language (DSL) even for beginners. However, for student beginners the API does not show its C nature – apart from the syntax:

```
#include <all.h>

int main(void) {

    init_all(); /* initialize devices */

    while(1) {
        led_set(LEFT); /* turn left led on */
        delay_ms(1000); /* sleep for one second */
        led_set(RIGHT);
        delay_ms(1000);
    }
    return 0;
}
```

The reason for our assessment is that the transition from playfully exploring robotics to practicing more complex robotics should be smooth and not require world-switching, e.g. from GUI to textual programming.

Bootloader. To program a robot, the robot must be connected to the development computer and the program is flashed onto the microcontroller. Afterwards it is possible to position all other HERD robots around the programed one, toggle a switch on each robot and the already programmed robot will transfer its code to all other robots in parallel via IR communication.

3.3 Live System

The live system (cf. Figure 4) can be booted from a DVD or USB stick and does not require or allow accessing the computer's hard drive. All software required to work with the HERD robots is provided and can be automatically updated.

Fig. 4. A screenshot of the booted HERD live system, which includes a customized Eclipse IDE, robot programming tools, an update mechanism and all required documentation.

Technically, the system is based on the Debian Live Systems project [8], running Eclipse as IDE, avr-gcc as cross compiler, avrdude as AVR programmer and git as version control system and for automatic updating.

4 Initial Evaluations

We took the 15 HERD prototypes to several schools in the vicinity of Karlsruhe, Germany, and evaluated them with students in grades 8 to 11. The students had different levels of programming experience ranging from no previous experience at all to two years of regular Java programming lessons. A few students had already used Lego Mindstorms [4] during school project days. Each pair of students had access to a computer running the HERD live system and one HERD robot.
Qualitatively, there are three main points worth stressing: First, all teachers reported that their students were eager to start using the robots and kept their motivation throughout the course material. Second, the beginners were able to learn the basic programming control structures – statements, function parameterization, loops and conditions - and utilize their knowledge to complete the exercises that required these. Third, the more advanced students quickly ran through all the single robot exercises –

apart from the line following one, but went on spending the rest of the lesson, if possible even staying extra time, to work on the swarm robot programs.
Overall, we got very positive feedback from students and their teachers. The students' feedback was that the robotics lessons were more interesting than "normal" computer science classes. The teachers emphasized the ease of use through the live system and that working with the HERD robots was not only a fun activity, but also brought the curriculum forward.

5 Discussion and Outlook

Most of the HERD development occured in 2011 and the evaluations continued in 2012, unfortunately it stagnated in 2013 because of lack of time on our side. We try to pick up the development again and attract support from other people working at the junction between (high school) education and robotics.
The near-term roadmap is as follows: 1) Substitute all through-hole components by SMD parts to bring down manufacturing costs - this part is already close to completion. 2) Improve the differential drive by either using stronger motors or adding a cheap gearbox. 3) Add a preconfigured robotics simulation environment, such as the one developed for the Wanda robot [3] or something based on Gazebo [6].
Yet, the biggest remaining challenge is to go from HERD prototypes to HERD as a product that can be ordered by interested schools.

Acknowledgments. This research has been supported by the *Scholarship for Outstanding Students Computer Science Karlsruhe (Begabtenstiftung Informatik Karlsruhe)* and the authors thank for the generous scholarship financing hardware costs.

References

1. The HERD project website, http://herd-project.org/
2. Kettler, A., Szymanski, M., Liedke, J., Wörn, H.: Introducing Wanda - A new robot for research, education, and Arts. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 4181--4186, 18-22 Oct. 2010
3. Kettler, A., Szymansik, M., Wörn, H.: The Wanda Robot and Its Development System for Swarm Algorithms. In: Rückert, U., Joaquin, S., Felix, W.: Advances in Autonomous Mini Robots, pp. 133-146. Springer, Berlin Heidelberg (2012)
4. The official Lego Mindstorms website, http://mindstorms.lego.com/
5. The KiCad website, http://www.kicad-pcb.org/
6. The Gazebo simulator website, http://gazebosim.org/
7. The official Arduino website, http://www.arduino.cc/
8. The Debian Live Systems website, http://live.debian.net/