ERNA - Embedded, Self-Calibrating Robotic-Arm for Gamificated Learning

Carsten Seeger, Christoph Käding, Christopher Manthey and David Neuhäuser
Institute of Computer Science
Friedrich-Schiller-University
07737 Jena, Germany
{carsten.seeger,christoph.kaeding,christopher.manthey,david.neuhaeuser}@uni-jena.de

Abstract—ERNA - Educational Robotic iNteractive Arm - is an embedded, self-calibrating robotic-system. ERNA in its recent configuration is capable to perform the game "Tower of Hanoi." School children and university students, not familiar with the principle of recursion, can play the game against the robotic-system step by step. Trying to beat the robot, the players will acquire a solution approach based on recursion to optimally play and win "Tower of Hanoi." ERNA is based on simple, low-budget components and can be rebuilt with different difficulties by high school children and univerity students. Several design challenges like mechanical tolerances, mass inertia, and switch bouncing, had to be handled to develop the robotic-system. This paper describes the design of the robotic-system and its application in detail.

Keywords: human-computer interaction, gamification, interactive learning, embedded system, self-calibration, recursion, robot, MBED, micro-controller, low budget

I. Introduction

ERNA - Educational Robotic iNteractive Arm - is an embedded, self-calibrating robotic-system to support gamificated learning [3]. It is embedded, meaning it performs without any additional, external computer control. It can autonomously solve its given task. It is self-calibrating, meaning any mechanical inaccuracy will be compensated by sensor supported absolute positioning. The robotic-system in its recent configuration is capable of playing "Tower of Hanoi" against a human player step by step.

"Tower of Hanoi" was invented 1883 by the french mathematician Edouard Lucas [5] as a challenging intelligence game for one player. At start, perforated disks of different size are sticked on three poles of equal height, see Fig. 1. At every step of the game, a single disk may be moved from one pole to another. At no time may a larger disk rest on a smaller one. The aim of the game is to use as little steps as possible to move all disk from the first to the third pole.

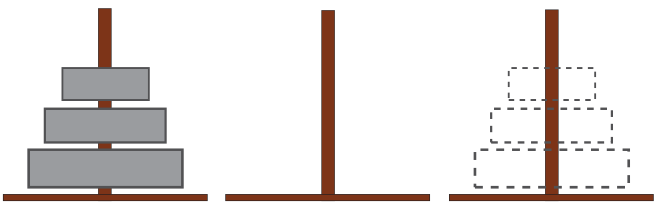

Fig. 1. Schematic of the game "Tower of Hanoi"

When playing with ERNA, the human player and the robot have their own game set up with equal amounts of disks. In every turn, the human player is first to move a disk. After completing its move, the human signals the robot to move a disk by pushing a button. When playing optimally in terms of using few disk moves, the human player will eventually beat the robot by finishing the game first. On the other hand, when the human makes any mistake, the robot will win since it plays optimally every time. To win the game continuously the human needs to find a playing strategy based on recursion. Using the robot, school children and students can be playfully introduced to the concept of recursion. This has been successfully demonstrated at the public science event "Long Night of Science" [9]. The robotic-system combines human-computer interaction and gamification.

The robot itself was also designed to be rebuilt by school children and students. The mechanical components can be assembled by both groups. The programming can be realized by older school children and students. The electrical wiring should be carried out by students. In this way, the robotic-system covers different aspects of gamification: learning with, learning from, and learning about robots [11].

II. Robotic-system

A. Specification

The scope was to design a robotic-system for solving the problem of "Tower of Hanoi".

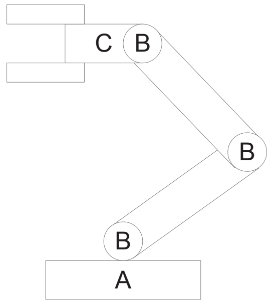

Fig. 2. Schematic of the robot: (A) electric powered wheel, (B) hinge, (C) gripper/magnet

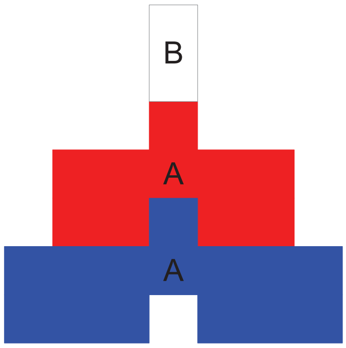

Fig. 3. Design of "Tower of Hanoi" suitable for the robot: (A) disks (B) magnet

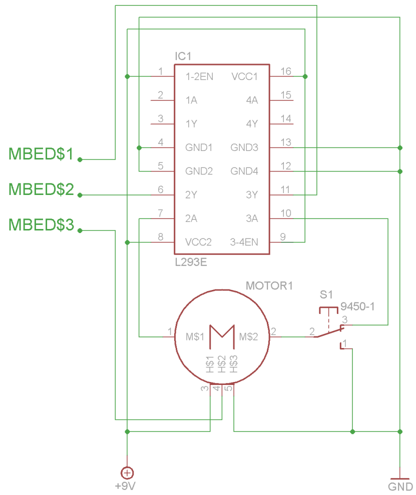

Fig. 5. Motor control wiring

To simplify this task some restrictions were applied. A stack of only three disks has to be moved. The solution algorithm inside the robot only needs a predefined static initial state of the towers. Additionally the system should be flexible by reaching arbitrary positions. For this purpose the robotic-system should be designed using hinges (see Fig. 2) which are realized using servo-motors. To energize the motors a self-designed circuit with a micro-controller is used. The mechanical environment is built with kits from fischertechnik [1].

To be able to grab a disk, two robot design alternatives arise. We can either using a magnet or a gripper, see Fig. 3. By using a magnet the original design of tower of Hanoi has to be modified. The poles have to be removed to easily grab the disks. With a gripper usage, two design options are given. The first is to grab the disk from above and the second one is to grab from the side. To reduce the amount of heavy mechanical parts, we decided to use an electrical magnet to grab the disks.

The system is supposed to be able to solve the Tower of Hanoi fully autonomously. No additional host is required. A micro-controller is used to control the robotic-system. It should be able to be programmed in a suitable language and have ports which make it possible to connect the controller to the magnet, any needed sensors, and the motors. Furthermore it should be possible to rebuild the project by school children and students.

B. Hardware development

The first approach was to develop a control to test the motor behavior. Therefore the motor was directly connected to the micro-controller. Some tests like inertia behavior and pulse count were performed.

The motor itself is a geared 9V motor using a regulated power supply. To spin the motor the power supply is activated for a specific amount of time. This is regulated with the help of an additional driver controlled by the micro-controller to regulate the power supply. To determine the exact position of the motor, it issues a PWM signal [10] to the micro-controller.

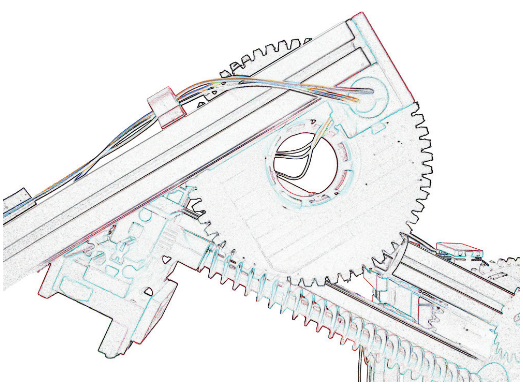

Fig. 4. Hinge with cogwheel and screw

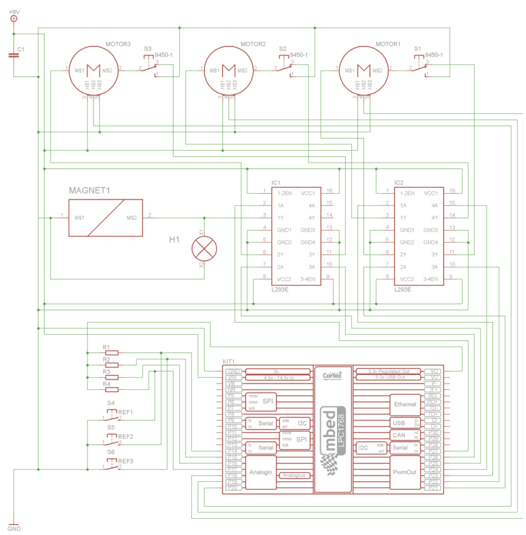

Fig. 6. Robot wiring plan

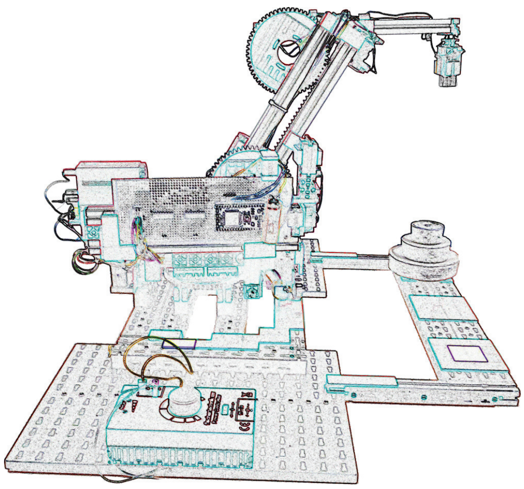

Fig. 7. Fully assembled robotic-system

For exact results it is necessary to define threshold values for low and high outputs of the PWM pulse.

During tests some unwanted behavior was noticed, for example inertia with and without load and mechanical blocking. Inertia handling and blocking detection were realized on the software side and will be discussed later.

The next step was to develop a hinge design. Important decision indicators were inertia prevention and the value of load. Therefore a design was used, were the gears which are responsible to move the segments of the arm, are not directly connected to the driving motor gear. An axis with screw and a cogwheel is located between these two gears. This has several advantages. For example, the load which is caused by the weight of the construction does not act in the direction of the rotation of the gears. So there is no need to spend some energy to hold the robotic-system in place. It locks itself and protects the motor mechanics. With the hinge design in Fig. 4, a first system with three degrees of freedom was built.

The micro-controller uses 5 V supply voltage. A driver element is needed (see Fig. 5) to use higher voltages driving the motors with more power. Using such an electronic driver, the motors can rotate forward and backward with high force.

The three motors and the magnet are connected via the drivers to the micro-controller. Several pull-up resistors and a capacitor are used to guarantee the stability of the motor driving voltage. At last the calibration switches are linked to the micro-controller to read the sensor activation on reaching calibration positions. Fig. 6 shows all parts connected together. The fully assembled robotic-system is shown in Fig. 7.

C. Software development

At first we implemented a blocking system to prevent damaging the motors while testing. The blocking system uses the fact that if a motor blocks at some point no PWM signal restores back to the micro-controller. If the motor gets input by the power supply but no PWM signal is received for a defined amount of time, a blocking is detected and the motor was triggered to stop performing.

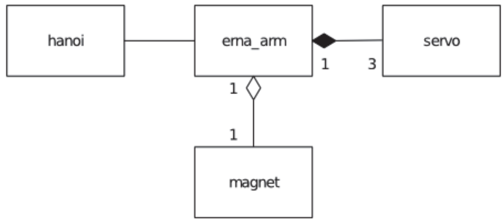

Fig. 8. Software classes and their relation

After creating the first versatile version of ERNA, software was needed to perform movement. A micro-controller MBED LPC 1768 [6] [7] programmable in C++ was used in addition with L293DNE drivers by Texas Instruments [2]. The Software itself is class-based with four classes called motor, magnet, erna_arm and hanoi. These classes represent the objects used by the system and behave like shown in the UML pattern in Fig. 8.

The most important class is the hanoi class which uses the erna_arm class to perform movement to positions calculated by a standard recursive algorithm to solve the “Tower of Hanoi” [4]. The erna_arm class uses three motor objects and the magnet object to move itself to a given position. It uses a radial coordinate system which gains position feedback from the motors and behaves like shown in Fig. 9.

Each of the three motors gets a specific position on its own circle. To move from an absolute position to another the motor class calculates the relative movement the motors need to perform. The magnet and motor classes are the low level classes interacting with the hardware.

To use absolute positions a null position has to be defined. For each motor a mechanical interrupt switch was installed on a defined null position. To reach this position all motors move in direction of their mechanical interrupt switch until they hit them. The mechanical interrupt switch itself breaks only the connection used for the direction towards the switch, the

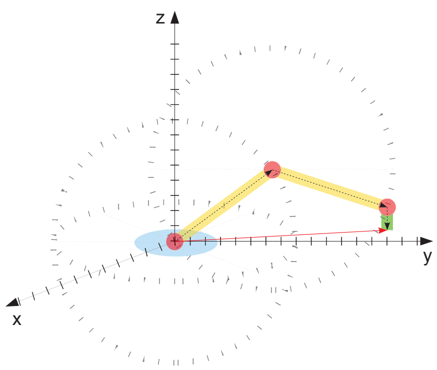

Fig. 9. Radial coordinate system: arm segments marked yellow, hinges red, magnet green.

ISBN: 978-1-61804-231-6

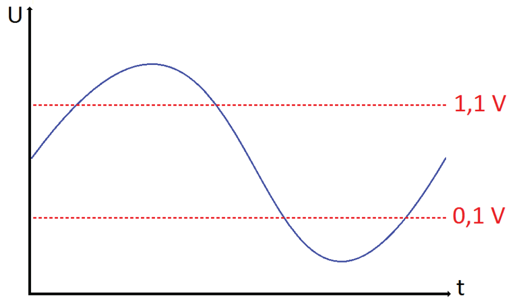

Fig. 10. PWM-signal thresholds

direction away from the switch stays connected. If a motor hits the switch it directly stops movement without the need of a blocking protection or software control. This was needed to differentiate the null position from a blocking position. On the first start of the micro-controller an initialization procedure moves the arm segments of ERNA to these null positions.

1) Evaluation and control of PWM-signals: The PWM-signals are sampled analog and evaluated by the software using pre-defined threshold values, see Fig. 10.

This design choice was made to reduce the complexity of the circuit. A software routine reads the input of the PWM-signal in cyclic time periods. Discretized voltage values are used to reconstruct the PWM-signal. The sampling frequency is 200 times higher than the maximum frequency of the motor and reconstructs the PWM-signal without loss. This sampling frequency fulfills the Nyquist-Shannon [8] theorem.

Another fact to consider is the inertia of a moving motor. The position has to be adjusted if an inertia movement is registered. To quantify the value of the inertia movement the PWM values between deactivating the supply voltage of a motor and the stopping of the PWM-signals are used.

To minimize inertia movement behavior a PWM control is needed. While using a DC motor a PWM controlled voltage equals to a lower direct current. The duty cycle is the relation between the low and high current time of the PWM-signal. For example a 0.5 duty cycle reduces the motor speed by half compared to a 1.0 duty cycle. This was used to reduce motor speed when coming closer to the target position. Four duty cycle levels where defined, the initial duty cycle and three levels downgrading the motor speed with empirical defined values.

2) Testing: Firstly we tested the motor behavior with and without weight load. The result was as expected, the motor moves without load with higher error rate as with heavy load.

Secondly we tested movement with compensation against inertia. This results in an unusable movement. The movement error is higher than the possible compensation.

The most important test is the precision test. We performed 50 movement iterations with and without PWM control. ERNA has to move between two points to show the deferment of the positions caused by repeated movements. This test shows a significant position deferment due to high tolerances of the mechanical parts. As a result, the robotic-system will not be capable of solving the problem of ”Tower of Hanoi” without any form of self-calibration.

3) Self-Calibration: To calibrate the robotic-system three sensors called reference switches were added on specific positions. These positions are physically fixed. If a motor activates the sensor the calculated position will be compared with the physical position of the sensor. If they don’t match the calculated position will be overwritten with the sensor position. In Fig. 11 a schematic of the calibration positions is shown. When moving between positions I and II, II and III, I and III, I and IV, II and V, III and VI a sensor is triggered and corrects the calculated position. Due to mechanical dimensions of the sensor, the actual physical position when activating the sensor differs when moving from I to II or from II to I, as well as for the other positions respectively. To compensate the physical dimension of the sensor two positions per switch are used in combination with the actual direction of the motor. The switches are debounced on software side. The hard-coded positions of the switches are empirical detected.

4) Hanoi algorithm: A standard recursive C++ implementation is used for the Hanoi algorithm [4]. This version was

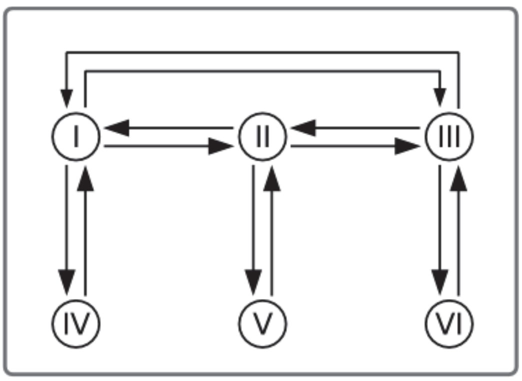

Fig. 11. Gripper positions used for calibration

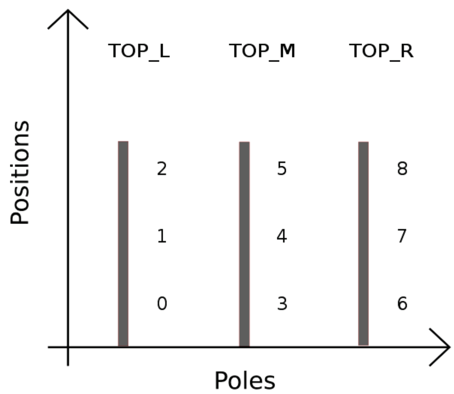

Fig. 12. Gripper position encoding

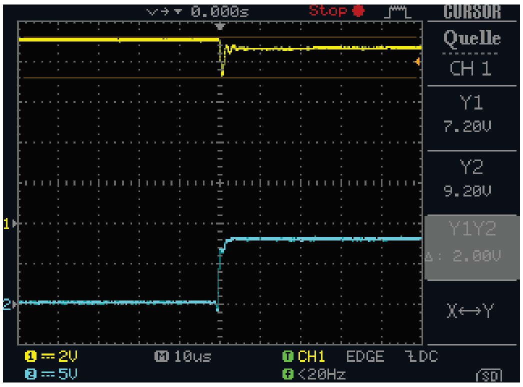

Fig. 13. Example of a voltage drops. Channel 1 (yellow): supply voltage dropping 2V down. Channel 2 (blue): motor voltage on starting induced by the robot arms mass inertia.

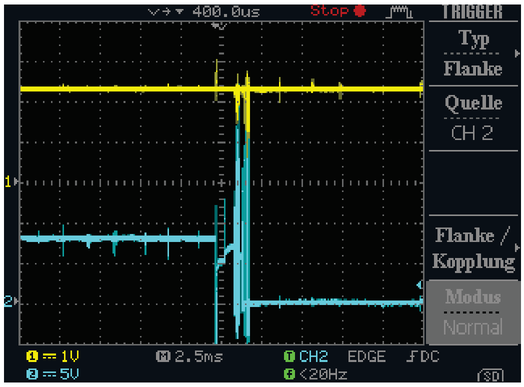

Fig. 14. Induced voltage when stopping a motor: channel 1 (yellow) signal of the switch, channel 2 (blue) drops on motor voltage

modified by adding a stack-based data structure representing the poles of the game. By using three different integer stacks, the information about position and height could be saved in this data structure. An advantage using this data structure is that pushing and pulling from a stack has the same behavior as pushing or pulling a disk from a pole in the real game. The positions are coded like shown in Fig. 12.

This encoding allows to calculate the top position by position modulo 3. For the left top position this value is lower then 1.0 for middle one between 1.0 and 2.0 and for the right one higher than 2.0 . A solve sequence is generated through a function using the origin and destination position of the calculated movement and saved in a FIFO data structure.

D. Solving the problem of "Tower of Hanoi"

As described before the major class is the hanoi class using the robotic-arm class to perform movement. Furthermore it triggers the hanoi algorithm with the standard parameters which start at the left pole with all disks and end at the right pole with all disks and uses three disks in total. This generates the solve sequence dynamically and could also generate solve sequences for all valid intermediate positions. After generating the solve sequence a loop grabs two elements of the FIFO data structure and creates a movement where the first element is the position of the disk which should be moved to the position represented by the second element. This additionally needs to use the top positions of every position in between. These positions can easily be calculated as described in II-C4. One loop iteration uses four positions: first the top position of the original position of the disk, then the original position itself, third the top position of the destination of the disk and at last the destination position itself.

III. Design Challenges

The most time consuming challenge is to ensure the precision of the whole system, jeopardized by the high tolerances of the cogwheels. Furthermore the motors state their movement by PWM-signals but can not be controlled with PWM input. The motors are powered by a constant current. Constant current can result in inertia which has to be calculated additionally.

Voltage drops are another problem as shown in Figure 13. The motors need a high voltage. The time delay of the power source to provide high voltage causes voltage drops when the motors start to move [10]. This can result in dysfunctions of the system. Additional capacitors charged during normal movement and discharged during movement starts are a suitable solution. Induced voltage can also occur when a motor stops movement, see Fig. 14.

A sensor problem is the bouncing behavior of switches shown in Fig. 15. This reduces the precision while generating random tolerances. Using a high sampling frequency reduces the effect but can not totally circumvent it. Another solution is using hardware debounced switches.

To program the MBED micro-controller in C++ an online compiler given by the company MBED has to be used (www.mbed.org/compiler). The IDE is not very user friendly and does not support C++ in its full usability. For example exception handling was not supported. This results in an unnecessary complicated software development process.

IV. Summary

A. Conclusion

ERNA - Educational Robotic iNteractive Arm - is an embedded, self-calibrating robotic-system. It is suitable to playfully teach school children and students the concept of recursion by competitively playing the game "Tower of

Fig. 15. Schematic of switch bouncing

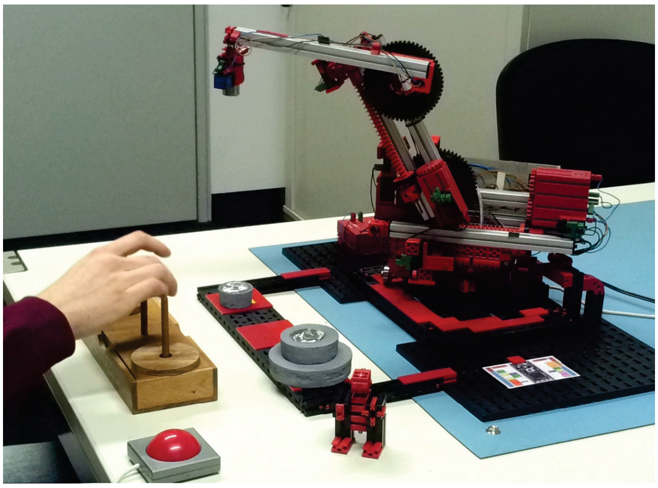

Fig. 16. Playing with ERNA

Hanoi", see Fig. 16. The robotic-arm has been successfully demonstrated at the public science event "Long Night of Science" [9].

ERNA is made of low-budget, simple components and can be rebuilt in class. Design challenges arising from using simple components have been solved successfully.

B. Future work

1) Multi-Threading architecture: The Robotic system does not support parallelized movement of the hinges. This can be achieved by using a more sophisticated micro-controller capable of multi-threading. Moving all hinges in parallel will reduce the time consumed by the robot to perform one step of the game.

2) Camera system: The system can be extended with a video camera to acquire and validate the initial state of the Hanoi disks to improve human-computer interaction.

3) Redesign: To avoid high hardware tolerances the system should be redesigned using better and in most times cheaper hardware. This will result in a much better precise and fault tolerant system. Maybe no calibration system is needed anymore, if the precise increases enough to minimize the differences between positions over a high amount of iterations.

REFERENCES

[1] fischertechnik. www.fischertechnik.de. Jan 31st, 2014.

[2] Texas Instruments. L293DNE, http://www.ti.com/lit/ds/symlink/l293d.pdf. Jan 31st, 2014.

[3] Alexandru Iosup and Dick Epema. An experience report on using gamification in technical higher education. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 27–32, New York, NY, USA, 2014. ACM.

[4] Doina Logofătu. *Algorithmen und Problemlösungen mit C++: von der Diskreten Mathematik zum fertigen Programm-Lern-und Arbeitsbuch für Informatiker und Mathematiker*. Springer DE, 2006.

[5] Edouard Lucas and N. Claus (Of Siam). The Tower Of Hanoi, Authentic Brain Teaser Of The Anamites, A Game Brought Back From Tonkina. 1883.

[6] MBED. LPC 1768 datasheet, http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf. Jan 31st, 2014.

[7] MBED. LPC 17xx User Manual, http://www.nxp.com/documents/user_manual/UM10360.pdf. Jan 31st, 2014.

[8] Harry Nyquist. Certain topics in telegraph transmission theory. *Proceedings of the IEEE*, 90(2):p. 617–644, 2002. ISSN 0018-9219.

[9] Long Night of Science. www.sternstunden-jena.de. Nov 29th, 2013.

[10] PWM-Ansteuerung. www.mikrocontroller.net/articles/Motoransteuerung_mit_PWM. Apr 20th, 2013.

[11] N. Shin and Sangah Kim. Learning about, from, and with robots: Students' perspectives. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 1040–1045, Aug 2007.