A Human-Friendly Way of Programming Robots

Francis wyffels, Karel Bruneel, Peter Bertels, Michiel D'Haene, Wim Heirman and Tim Waegeman
Electronics and Information Systems Department, Ghent University — Belgium

***Abstract*— Embedded systems are becoming a more important part of our lives. Students enjoy them on a daily basis, but they cannot imagine they will ever be able to build such systems themselves. We believe that this is due to the dull bottom-up approach that is taken by our science education system. By the time students are allowed to do something fun and interesting with the knowledge they have acquired, they were bored to death by endless hours of abstract science and math classes. We believe that introducing robotics early on in their curriculum can spark their curiosity and revitalize science education.**

I. Introduction

In "A robot in every home" [1], in the January edition of Scientific America, Bill Gates predicts that the next hot field will be robotics. This requires that today's youngsters get acquainted with robotics — or technical subjects in general — early on. This idea is, in our experience, not yet adapted by the majority of Belgian schools. On the other hand, the toys industry answered this need by providing robot building kits, of which Lego Mindstorms NXT is well known among Belgian youth. These kits are both fun and educational. However, the step towards real robots is still substantial, due to the large level of abstraction. Other kits exist, but are often too complex and aim for the already technically minded. Hence, one could say that there lacks an integrated methodology with which programming can be taught in an enjoyable way, from high abstraction levels down to lower levels, throughout the multiple layers of education. From our experience, we know that robots can be of great help in realizing this goal [2].

In this work, we present an approach for teaching and motivating students from college and higher education levels. We propose a three-fold way of programming embedded systems and the use of a real robotics platform. By combining hardware and software, we hope that students and educators who are currently hesitating due to lack of knowledge, can be motivated by our approach.

II. From Graphics to Text

Our main goal is to make microcontrollers available to everyone. Since textual programming is often a step too far for beginners, we developed Dwengo Blocks,[1] a freely available web application with which microcontrollers can be graphically programmed and simulated. The main advantage of graphical programming is that the program's control flow is explicitly represented. This makes it easier for students to understand their program. Once the program is built and tested in the simulator, students can download the compiled program and run it on a real Dwengo board, which makes the experience real and exciting, see Figure 1.

To enable a smooth transition from graphical to textual programming, the graphical program is automatically translated into easily readable C code. Due to the one-to-one mapping of the graphical building blocks onto C constructs, we are able to visualize which code is being executed. Showing the graphical program next to the textual program during simulation, makes it easier for students to understand how their program is represented in textual form.

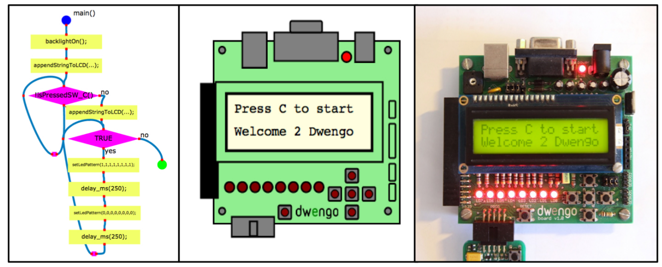

Fig. 1. Example in Dwengo Blocks: from graph to real hardware.

III. From Theory to Reality

Students tend to get motivated when they are asked to solve real world problems, and more so when these problems involve robots. Therefore, we developed a robotic starters kit based on the Dwengo board. Its tracked wheels and multiple sensors can be programmed to interact with its environment, turning the robot into for instance a simple line following robot or an advanced mace solving robot. Students hereby have to deal with real world problems such as noisy sensor information and dynamic events. To raise the excitement even more, robot competitions can be organized in class.

We tested this approach with 37 high school students (16–18 y.o.). The students were tasked to program a line following robot in C using the Dwengo library. Because of time constraints in a 2 hour workshop, and their lack of existing programming skills (these are hardly integrated in their courses), we divided the problem into four challenges: (1) driving the robot in a square pattern, (2) adding blinking lights, (3) reading sensors, and (4) following a line. Some sample code was provided as well a robot platform for every two students. After only two hours, more than 50% of the robots fulfilled all four tasks with flying colors, and, more importantly, all students had a great time doing so!

IV. Conclusions

In this work we present a novel way of motivating young people for technical education. We believe that by using our three-fold approach combining programming and robotics, programming can be made accessible for everyone.

References

[1] B. Gates. A robot in every home. *Scientific American*, 296:58–65, 2007.
[2] F. wyffels, K. Bruneel, P-J. Kindermans, M. D'Haene, P. Woestyn, P. Bertels, and B. Schrauwen. Robot competitions trick students into learning. In *Proceedings of the 2nd international conference on robotics in education*, 2011.

All authors are (co)founders of Dwengo (*http://www.dwengo.org*), a non-profit organization making embedded device programming available to everyone by providing the necessary tools to educators, students and hobbyists.

[1]You can try out Dwengo Blocks at *http://blocks.dwengo.org*