SqueakBot : a Pedagogical Robotic Platform

Serge Stinckwich
GREYC, CNRS, Université de Caen
Serge.Stinckwich@info.unicaen.fr

Séverin Lemaignan
Planète Sciences
skadge@gmail.com

Samir Saidani
Planète Sciences
Samir.Saidani@info.unicaen.fr

1 Introduction

Since 1962, a french association named Planète Sciences proposes to young people experimental scientific and technical activities, within the context of their leisure and school time, with the support of well-known scientific and industrial organizations. Each year, more than 50 000 young people are touched, with the assistance of 500 scientific clubs and the participation of 1000 organizers and trainers specialized in the supervision of holidays centers, workshops, projects of educational actions, exposures, ... Planète Sciences developed a whole innovating teaching around robotics: from the design of robots by children to their programming. This activity is well-known in France because Planète Sciences is also at the origin of the french robotic competition broadcasted by television.

Many teaching approaches using robots have already be proposed by many universities courses among the world. The general difficulty of the majority of these projects is that it requires on one hand the learning of a computer programming language and in the other hand sufficient knowledge to build mechanical and electronical parts of a robot. This is usually a too difficult task for young people. Kids spend too much time on the physical aspect of the robot than the design of the embedded intelligence. This could be disappointing for young people who quickly wish to see results of their own work.

The University of Caen established a collaboration with Planète Sciences to answer some of these difficulties. We produced a first prototype of a pedagogical robotic platform named SqueakBot. This platform on the one hand is based on modular control components that are plugged to the actuators and sensors of a real robot, and on the other hand is programmed with the help of Squeak and EToys. Finally we developed first bricks of a robotic simulation system which allow the virtual design of robot in order to accelerate the effective construction of a robot.

2 SqueakBot Platform

Planète Sciences developed an in-house whole set of electronic components making it possible to control real robots. These components, which we will describe more in detail below, appeared as electronic boxes, which were greatly simplified, so that they can be easily handled by young people. They provide inputs and outputs making it possible to connect them on side to a computer which is used as remote brain for the robots and the other side with sensors (photoresistors or simple contact) or actuators. LEDs on the front of the cases indicates clearly the state of these modules.

The programming of these components is carried out in MSWLogo[1], a free and open-source version of Logo. The essential advantage of Logo is that it's a very simple programming language to learn for children. Nevertheless, this choice suffers of many drawbacks: MSWLogo only works on Microsoft Windows platform, despite his simplicity, Logo requires to learn a syntax (a more visual approach is more desirable), lastly LOGO remains primarily a functional language et is somewhat too much turtle-oriented. These constraints led us to replace LOGO by a software solution based on Squeak EToys. Squeak is a freely available implementation of the Smalltalk programming language that provides a rich set of multimedia bricks (images, sound, video, 3D), a fully integrated development environment and a great constructivist environnement for children called EToys.

2.1 Electronic control modules

2.1.1 Overview

The electronic modules of control, which were designed by Planète Sciences, are the nervous system of the robot. They were designed to be as simple as possible, to allow children to focus on the choice of actuators and sensors and on the programmation of the robot. A parallel port adaptor is also provided, a kind of simple "adaptor pipe". Its role is to connect the parallel port cable to a phone-style cable and to provide some voltage security in case of mistakes. The figure 1 describes how the different components are plugged into each other. The adaptor (MOEBUS) is plugged into the computer and a control module (SMEC) into this adaptor. Electrical motors and sensors are plugged into control modules. If needed,

[1] http://www.softronix.com/

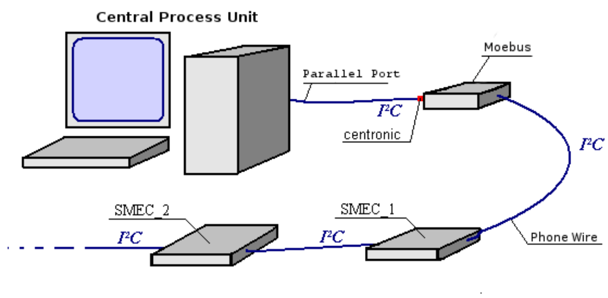

Figure 1. The robot with its brain and its nervous system

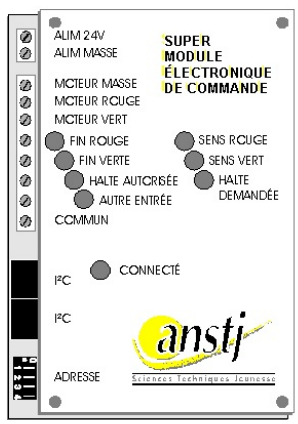

Figure 2. SMEC module

more sensors and actuators input/output are obtained by cascading additional modules (up to eight control modules).

2.1.2 SMEC and I2C protocol

SMEC (Super Module for Electronic Command[2], figure 2) is an advanced module for young people beyond 13. It allows the control of an electric motor, by using the parallel port of a computer, thanks to the MOEBus module (see below). Several SMECs can be connected on the same robot and their control is done by means of a dedicated serial link that use the I2C[3] communication protocol developped by Philips company. The main advantage of I2C is that the computer can receive information provide by sensors of the robot : the program can react directly to his environment.

The protocol used for communication is the I2C[4] protocol developped by Philips company. Through this protocol, the computer can send an information command to the motors and receive information provided by sensors of the robot : the program can react directly to its environment. There is two main protocols for robotic and domotic applications : the I2C bus and the CAN bus. The CAN bus is a much more complicated protocol than the I2C one, and it's designed to be very robust. But the simplicity of the I2C protocol is better suited to use in a pedagogical environment.

2.1.3 MOEBUS

The MOEBUS module (MOdule Électronique de contrôle du Bus I2C[5]) is the essential complement of SMECs when one works with a computer. This module can control the others modules by adapting the parallel port to an I2C bus. It also integrates a power supply for all the modules on I2C network.

In order to control a robot, you have to connect a MOEBUS module to the parallel port of your computer and each sensors and actuators are connected to a set of SMEC modules. The communications between the various modules are carried out thanks to the I2C protocol.

2.2 Squeak Implementation

All the infrastructure explained above (SMEC and MOEBUS) have been used for around ten years with the LOGO language as the main programming language. Although LOGO is a great functional language, we find in Squeak the opportunity to explore how children could handle robots programmation with a fully object-language like Squeak. The Squeak syntax is rather simple to understand for the young, and the multimedia capacities as well as the platform openess allows the expression of creativity at all level. The possibility to share the software and to run squeak in different platform is crucial when working with children : it gives them the right to share with their friends their own work regardless of the underlying operating system.

This project started in 2002 when we proposed to students to develop a Squeak interface suitable for the young. A lot of work have been done during three years by different people, mainly students, involved in the project. Here is a summarize of what has been achieved:

2.2.1 SqueakBot Low-level support

A Squeak plugin written in C was carried out to manage the low-level aspect dealing with parallel port communication with the Planète Sciences modules. This plugin use the Squeak FFI[6] interface. Several Squeak classes represents each components of the system : **CommunicationHandler** handles in

[2] Super Module Électronique de Commande in french
[3] Inter-Integrated Circuits
[4] Inter-Integrated Circuits
[5] I2C Electronic Control Module
[6] Foreign Function Interface

an uniform manner all the communication, **I2CHandler** deals with the I2C protocol, **I2CInterface** defines the abstract interface of an adaptor making it possible to exchange I2C data on various ports (parallel port, USB, ...), **Moebus**, **Smec**, **ParallelPort** represent the real devices.

2.2.2 Remote control with EToys

The control and command modules (MOEBUS, SMEC) are represented on the computer screen by means of the EToys mechanims so their behavior is easily scriptable (see figure 2). We also use the Morphic Wrapper interface to facilitate the dialogue with the modules.

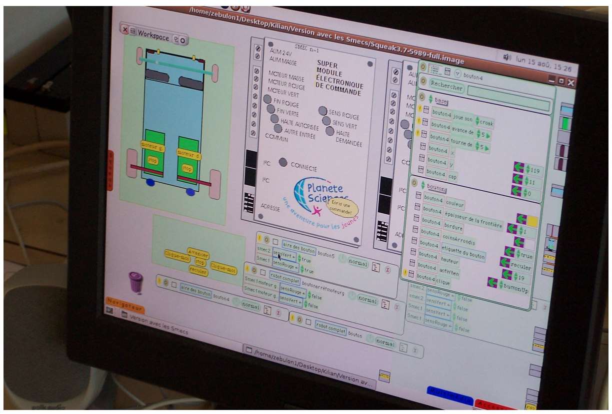

Figure 3. SqueakBot in action

3 SqueakBot Experimentations with kids

SqueakBot was successfully tested with teenagers in August 2005 during a summer camp organized by Planète Sciences. Planète Sciences proposes since more than 30 years scientific summer camps. They target at kids from 7 to 18 years old and are a mix of scientific and leisure activities, coached by young scientists. Depending on the age of kids, scientific activities take between two and six hours a day and the camps last two to three weeks. At the beginning, kids are asked to choose their main scientific activity they will develop during the camp through a team project. We tested SqueakBot in August 2005 with 9 kids who chose robotics for their projects. They were 13 to 15 years old, and, except one who had already attended a robotic summer camp, they were with little to no knowledge in robotics and even in technical activites for some of them.

Robots are made of raw material like plywood, PVC, aluminium. Parts are generally assembled with hangnail, screws or glue. Children have moreover at their disposal different kind of DC motors and a variety of small electrical and electronical devices like lights, LEDs, photoresistors, different kind of switches and micro-switches, etc. During a robotics' sensibilization phase, each kid started a programming project with Squeak. The next step, for children who chose robotics, was to define what kind of robot they wanted to build. During the August 2005 summer camp, we wanted to avoid as much as possible too classical 4-wheels rovers and kids made four teams working on four different and original projects: a **robot for Martian exploration**, with 6 independent wheels (figure 4), a **walking robot** aiming to test some hypothesis on how animals walk (figure 6), an **autonomous buggy** able to follow a path (kind of Braitenberg vehicle) and a **caterpillar robot** based on servo-motors (figure 5).

The conception of these robots started for every projects by concept schemas, plans and then a dummy model made of [pasteboard (carton, à vérifier)]. These models were proofs of concept and helped to discover mistakes for the definitive realizations. They also allowed teams to split between "programmers" and "mechanical engineers". Thus, some of the children could start experimenting with programming in the early stage of the project. This first phase lasted about five days of fifteen for the whole project.

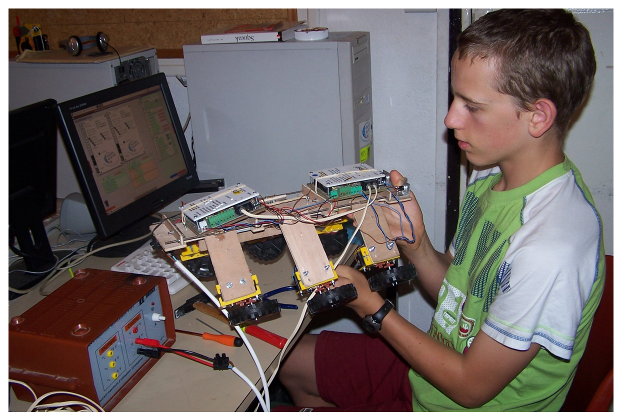

Figure 4. Robot for Martian exploration with two SMEC and SqueakBot

3.1 Robotic projects

3.1.1 Robot for Martian exploration

This project (figure 4) was lead by a team of three children. They wanted to build a vehicle able to progress on irregular soils, i.e. with several independant wheels. They chose to build a robot with six motors bound to independant arms. Two SMEC were used, one for the left motors, one for the right ones. Besides they put sensors in front of and behind the robot to programmaticaly add behaviours like going backwards when a wall was hit. The programming part hasn't been fully completed during the summer camp becouse of a time issue (the mechanical part was quite complex), but SqueakBot was successfully used to do tests on the pasteboard model.

3.1.2 The walking robot

The **walking robot** (figure 6) was a project aiming at the study of walk by quadripedes. Two children were involved on this project, made of four rigid legs controlled by four SMEC. Each leg could move in both ways, with end-of-course sensors restricting the movement amplitude. Kids started by collecting some documentation on quadripeds walk and once a rough version of the robot was achived, they sketched different type of walk and started to implement it through SqueakBot.

Without knies on the legs, it has been eventually hard to get a fluid movement, but the children really got in touch with the underlying issues of walk (synchronization, dynamic balance...).

3.2 Project based on the ASPIC interface

We suggested the last project, a **caterpillar robot** to the children, as a way to discover servomotors. One girl (actually without any technical background) started on this. This last robot (figure 5) was build around four servomotors, controlled by one ASPIC board. The child build four similar modules (a servomotor bound to a small board of wood) which were then assembled. The relative movement of each modules to others was something surprisingly hard to intuite and even to understand for the child, and the ASPIC's extension board, which allows to slowly and manually turn each servomotor individually, was helpful to decompose the whole movement.

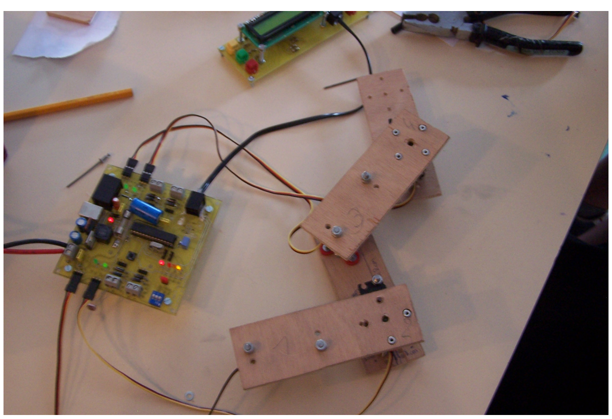

Figure 5. The caterpillar robot with the ASPIC interface

The second part of this project was its programming with SqueakBot. It was quite easy to set static positions (like on figure 5: on this picture, each servo is set to stand a heading around 45°). It was although harder to get the caterpillar actually move. The child tried to figure out what kind of movement each separate module should have. It leads us to introduce the sinusoïdal function which we set up in SqueakBot to control to servomotors heading. By adding a small phase difference to each servomotors, we got a nice wave-motion. And "Suzy the caterpillar" actually went forward.

4 Conclusion

This first test of SqueakBot in a real situation was globally very satisfying. Squeak is an appealing environment for programming where children are relatively autonomous, and SMEC and ASPIC control via SqueakBot was easy. This first release need although some polishing. The interface remains sometimes buggy, and was too slow for a smooth usage with the ASPIC interface.

Compared to other approaches based on Python or Java, Squeak is a good alternative to program and control robots remotely. Indeed, two levels of programming are offered. With the visual one (the Etoys level) you could start simple robotic activities with children beyond 13, and then it is possible to use gradually a real programming language (Smalltalk) still in the same environment.

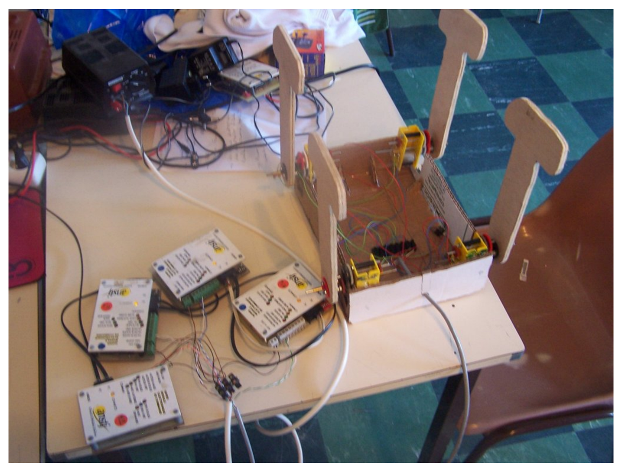

Figure 6. Walking robot with four SMEC

The project code is release under a MIT licence and freely available on the SqueakSource web site here : http://www.squeaksource.com/SqueakBot.html