A New Open Source 3D-Printable Mobile Robotic Platform for Education

J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno,
and M. Abderrahim

Abstract. In this paper we present the Miniskybot, our new mobile robot aimed for educational purposes, and the underlying philosophy. It has three new important features: 3D-printable on low cost reprap-like machines, fully open source (including mechanics and electronics), and designed exclusively with open source tools. The presented robotic platform allows the students not only to learn robot programming, but also to modify easily the chassis and create new custom parts. Being open source the robot can be freely modified, copied, and shared across the Internet. In addition, it is extremely cheap, being the cost almost exclusively determined by the cost of the servos, electronics and sensors.

1 Introduction

Mobile robotics is increasingly entering the curricula of many technical studies. Robotics is gaining terrain in industry and consequently more firms are recruiting candidates with experience in robot programming. For this reason, many universities are teaching robotics in their master and degrees programmes[15, 13].

A common approach when teaching robot programming is the use of simulations, in which the user can create different robot configurations with low effort. These ad-hoc robots can be also shared with other people, multiplying the number of *out-of-shell* platforms [3, 5]. Furthermore, the cost is zero, you

---

J. Gonzalez-Gomez · A. Valero-Gomez · M. Abderrahim
Robotics Lab, Carlos III University of Madrid, Spain
e-mail: `{jggomez,avgomez,mohamed}@ing.uc3m.es`

A. Prieto-Moreno
Autonomous University of Madrid, Spain
e-mail: `aprieto@uam.es`

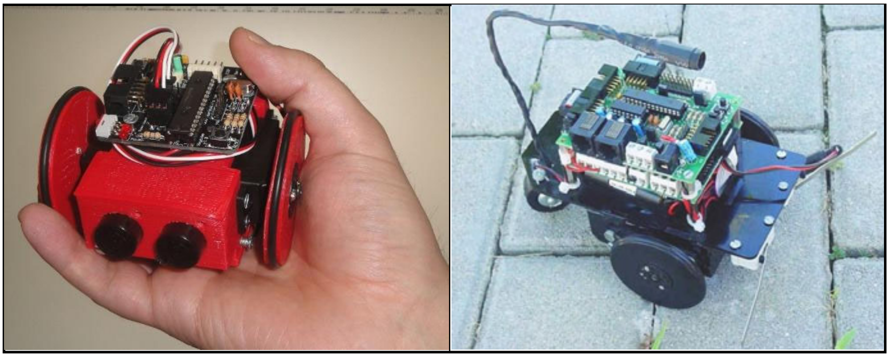

Fig. 1 Left: The new Miniskybot v1.0 robot. Right: The educational skybot robot

may have as many robots as you want, and they will never break. But this solution has one drawback: simulated robots are not like real robots. Things working on simulation may not work the same on a real platform. In addition, students will not enjoy the same testing their ideas on a real robot than on a simulated one.

Should it not be great that robots could be shared in the same way that code is shared (like in simulation)? If this could be possible researchers, professors and students could share their open source robots through the Internet, exchange ideas with other research groups, compare prototypes, test their algorithms on different configurations, evolve proposals from others..., such an idea is now possible and affordable thanks to the open source Reprap-like 3D printers[4].

This opens a new way of teaching robotics with the following advantages:

- Fast prototyping of robotic platforms.
- Low cost printing of robot parts.
- Easy reconfiguration and adaptation of the platform (evolution).
- Easy sharing of robot models among people.
- Motivation for students not only to implement algorithms on an existing platform but also to design and build new platforms.

In this paper, our new 3D printable Miniskybot robot platform is presented (shown in figure 1). It is fully open source (both the mechanical and electronics parts) and exclusively designed with open source tools (Openscad, Freecad and Kicad). The parts were furtherer printed in a Makerbot Cupcake 3D printer.

2 Motive and Problem Statement

Among the commercial educational platforms we can find a great variety of opportunities, starting with the well known Lego Robot, and going through the Meccano Robot, the RoboRobot robotic kit[1], or the OWI Robot Arm Edge[2]. These products are quite extended in the educational environment, they are affordable, and easy to use. They usually come with associated software, which allows users to interface with the robot, having access to sensors and actuators, program them, and so forth. These platforms have been present for some years now in the educational environment. In [10] the authors demonstrated the idea of a children's league for RoboCup, using robots constructed and programmed with the LEGO MindStorms kit to play soccer. Since then, RoboCupJunior has evolved into an international event where teams of young students build robots to compete in one of three challenges: soccer, rescue and dance [11, 14]. Goldmand et al.[6] presented an educational robotics curriculum to enhance teaching of standard physics and math topics to middle and early high school students. This project was also centered around the Lego MindStorm.

The major disadvantage of these platforms is that they are close. The users can hardly adapt them to their necessities, and instead, they must adapt to them. The reconfiguration of the platform may be a great advantage in order to be able to deploy all the initiative of the researchers, professors or students. The Lego MindStorm inherits the "build-it-yourself" of the Lego traditional toys, but users are constrained to use the sensors provided by the manufacturer, as well as the development software. An effort could be done to work around this limitation, but this goes beyond the original design of the platform. A work trying to meet the "open source" and the "non-free" directions is done by O'Hara et al[12].

Ad-hoc mini-robots have been built by research groups or university spin-offs mainly for educational purposes. These solutions overcome the limitations of the commercial robots, providing cheaper and more adapted solutions. Efforts have been done with the intention of developing effective and low-cost robots for education and home use, designed and built to fit the particular requirements of a teaching programme. Examples are those of IntelliBrain-Bot [3], Martin F. Schlogl's robots[4], the TankBot[5], the Trikebot [8] among many others.

This had been also our way of teaching robotics during many years, with our Skybot[6] platform (shown in figure 1). In our courses, the students build the Skybot from scratch and then program it. Sometimes they are so motivated that they propose wonderful modifications to the robot design. Even though some modifications are known beforehand that will not work well, we would like the student to discover it by himself. But in any case, it is not

---

[1] http://roborobo.koreasme.com/educational-robot-kit.html

[2] http://www.owiroboticarmedge.com/

[3] http://www.ridgesoft.com/intellibrainbot/intellibrainbot.htm

[4] http://www.mfs-online.at/robotics.htm

[5] http://profmason.com/?p=320

[6] http://goo.gl/MdRJs

possible to implement these modifications during the course due to the time it takes to the manufacturer to build the parts. At the end we had to keep the platform, or in the best case, change it for the next course with new students.

To summarize, the classical way of teaching robotics must focus, by necessity, on the programming of the robotic agent given a particular platform. Even if only this can be quite challenging and inspiring, with our current proposal of open source printable robots, the teaching programme must not be focused any more *only* on the robotic agent, but it may also include its mechanical design. Beginning with a basic platform, like the Miniskybot, students can be guided through the design and programming process. In this way, they may discover the tight relation between hardware and software, and how each of them can and must, adapt to the other requirements in order to achieve a precise task. They may learn that a particular mechanical design suits better a precise task, test different alternatives, and so forth. And something that is hardly considered in robotic programmes, students may learn that a change in the mechanical design could solve a problem better, faster, and more robustly than a software solution.

3 On Low-Cost 3D Printers

Bradshaw et. al [2] have recently made a study on low-cost 3D printing. They briefly run through the history of 3D printing, beginning in the late 1970s. These more than thirty years have driven to affordable 3D printers for individuals[1], and allow them to print complex engineering parts entirely automatically from design files that it is straightforward to share over the Internet. While open source software development has been studied extensively, relatively little is known about the viability of the same development model for a physical object design. 3D printers are offering new possibilities of sharing physical objects. As they can be defined using code, researchers can share their own parts, evolve them and "build" them straight forward using 3D printers. This allows for a decentralized community to independently produce physical parts based on digital designs that are shared via the Internet. Apart from improving the device, dedicated infrastructures were developed by user innovators. As Bruijn shows in his master thesis [4], a considerable improvement of hardware are proposed by people sharing parts and having access to 3D printers. This hardware modifications are relatively easy for others to replicate. As it has been the case with software for many years, currently, there are also on-line repositories of parts, where people can download and upload their designs[7].

In figure 2 four of the most important open source 3D-printers are shown. The origin of these kind of printers was the reprap project[8] [9] started by Adrian Bowyer in 2004. The aim of this project was to develop an open-source

---

[7] http://www.thingiverse.com

[8] http://reprap.org/wiki/Main_Page

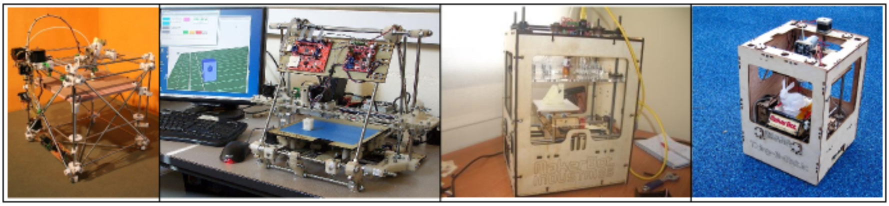

Fig. 2 Pictures of some open source 3D printers. From left to the right: RepRap Darwin, the first generation (May, 2007); Reprap Mendel (Sep, 2009) the second generation; Makerbot Cupcake (April, 2009), the first commercial open-source 3D printer; Makerbot Thing-o-Matic (Sep, 2010), second version

self-replicating machine. In May 2007 the first prototype, called Darwin was finished and some days later, in May 29th the first replication was achieved. Since then, the reprap community (original reprap machines and derived designs) has been growing exponentially[4]. The current estimated population is around 4500 machines. The second reprap generation, called Mendel, was finished in September 2009. Some of the main advantages of the Mendel printers over Darwin are bigger print area, better axis efficiency, simpler assembly, cheaper, lighter and more portable.

Initially, both Darwin and Mendel were not designed for the general public but for people with some technical background. As the reprap project was open-source, small companies were created to start shelling these 3D printers, as well as derived designs. The first company was Makerbot Industries[9], who shipped a first batch of their Cupcake CNC in April 2009. By the end of 2009 they had shipped nearly 500 complete kits. After operating for a year they had sold about 1000 kits in April 2010. Their latest design is the thing-o-matic printer, announced in September 2010. It is really easy to build and use, and their cost is around 950€.

Currently, at the System Engineering and Automatic Department of Carlos III University of Madrid we have one thing-o-matic available for the students, shown in figure 3. It was fully assembled by the students. Anyone has free access to it so that they can print whatever designs they want. Our main goal is to stimulate their imagination and enhance their creativity.

In addition we have started a project, called “Clone wars”[10], in which a group of students are building their own reprap printers from the scratch. All the parts are being printed in our thing-o-matic, which has been named MADRE (that means mother in spanish). We have chosen the Prusa Mendel model as the design to build, because it is very well documented and it is rather easy to assemble. In figure 3(on the right) the first prototype is shown. In total the students are building 20 of them.

---

[9] http://www.makerbot.com/

[10] http://asrob.uc3m.es/index.php/Proyecto:_Clone_wars

Fig. 3 Our open source 3D printers at Carlos III University of Madrid. On the left: the Makerbot Thing-o-matic, called MADRE. On the right: a Prusa Mendel prototype, being built by the students

4 The Miniskybot Mobile Robot Platform

4.1 Introduction

The new Miniskybot robotic platform[11] is open source: all the mechanical and electronic design has been released with a copy-left license. Furthermore, only open source software tools have been employed. This is important because in doing so it is guaranteed that anyone will be able to read, understand and modify the design files without license issues and using their preferred computer platform (Linux, Mac, BSD, Windows...).

The Miniskybot is a differential drive robot composed of printable parts and two modified (hacked) hobby servos. It has been designed so that it can be printed on open source reprap-like 3D-printer. Two mechanical designs have been developed: the minimal version and the 1.0.

4.2 Minimal Version

The first prototype developed was a minimal robot chassis. The idea was to design a printable robot with the minimal parts, a kind of “hello world” robot. It is shown in figure 4. It consist of only four printable parts: the front, the rear and two wheels. They are all attached to the servos by means of M3 bolts and nuts. Standard O-rings are used as wheel tires. For making the robot stable, the rear part has two support legs that slide across the floor. Therefore this prototype is only valid for moving on smooth flat surfaces. The goal of this first design was to show the students a minimal fully working mobile robot for stimulating their minds. They were encouraged to improve this initial design.

---

[11] http://www.thingiverse.com/thing:7989

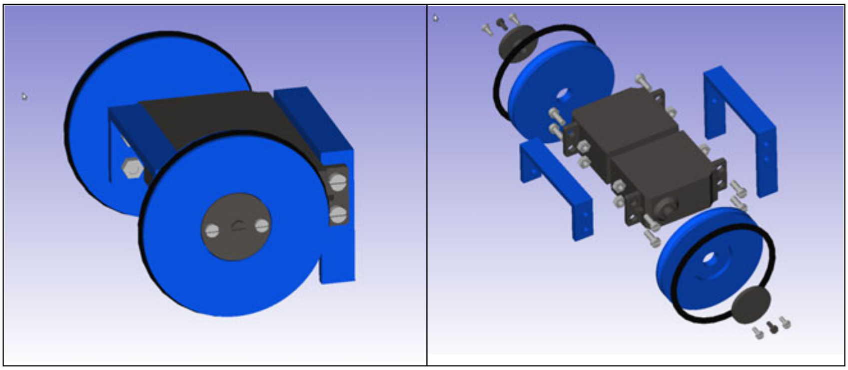

Fig. 4 Miniskybot. Minimal version

4.3 Miniskybot 1.0

The version 1.0 chassis is an evolution of the previous design (figure 5). It consist of nine printable parts: the front, the rear, two wheels, the battery compartment, the battery holder and the castor wheel. An important feature is that the parts have been parameterized, just changing some parameters new parts are obtained. For example the battery compartment is automatically changed if the parameter battery type is set from AAA to AA. In this case a new compartment capable of holding AA batteries (instead AAA) is generated.

The parametric feature is possible thanks to the open source Openscad[12] software used for designing the pieces. The parts themselves are not graphical meshes but scripts that determine how they are built by primitive geometric forms. When these scripts are "compiled" the graphical part is generated and rendered on the screen, and later exported as an STL file for 3D printing.

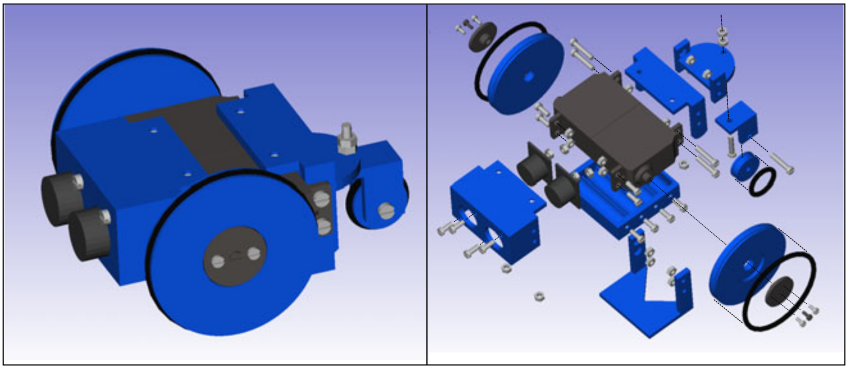

Fig. 5 Miniskybot Robot. Version 1.0

---

[12] http://openscad.org

This approach is very flexible because the parts are ASCII scripts that can be easily shared through Internet, stored in repositories and so forth. Therefore the mechanical designs can be modified, used, and printed easily by different people around the world.

4.4 Electronics and Sensors

The Miniskybot's electronics is the Skycube board[13]. It was previously designed for fitting into the Y1 modules for controlling the modular robots used for research purposes[7]. It is a minimal design with only the necessary components for controlling the robot. It includes an 8-bit pic16f876a micro-controller, headers for connecting the servos, an I2C bus for the sensors, serial connection to the PC, a test led and a switch for powering the circuit (figure 6).

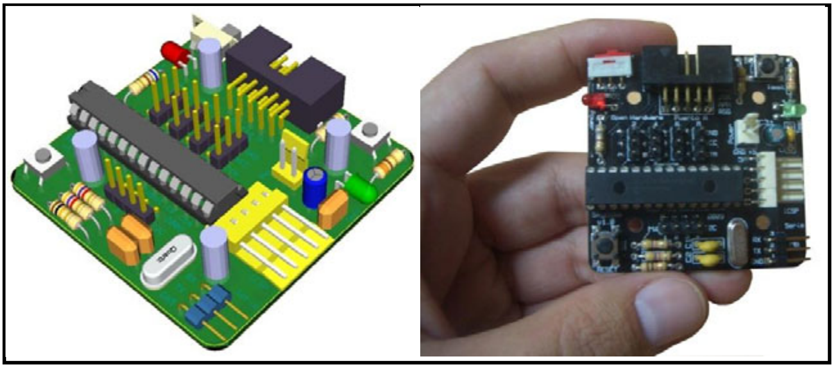

Fig. 6 Electronics. Skycube board

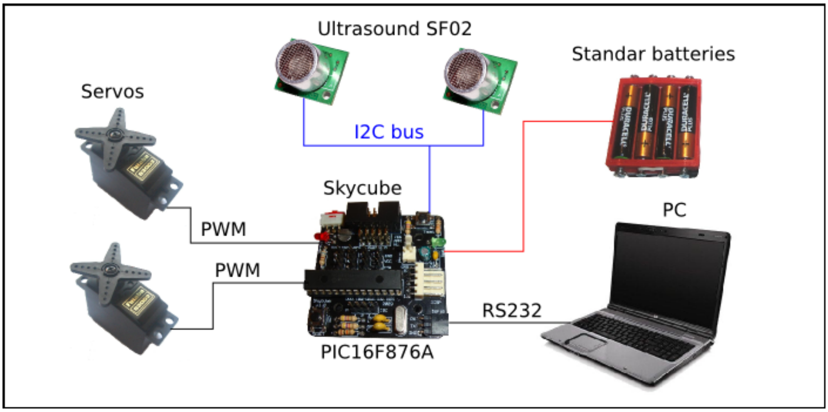

Fig. 7 Electronic diagram

---

[13] http://www.iearobotics.com/wiki/index.php?title=Skycube

An electric connection diagram is shown in figure 7, where the servos are connected directly to the board. The speed is set by means of two PWM signals. The two ultrasound sensors in the robot's front are connected thought the I2C bus. Robot version 1.0 have two ultrasound sensors, but as they are connected to the I2C bus, more sensors can be easily added. For the power supply four AAA type standard batteries are used. The board can be connected to the PC by a serial RS232 connection for downloading the firmware. The PCB has been designed with the open source Kicad tool.

The robot is programed in C language using the open source SDCC cross compiler and the binary files are downloaded into the board by means of a serial cable. Previously a bootloader firmware needs to be burned in the flash memory by means of the ICSP connector. Loading the firmware this way the students do not need to use any programming hardware but just a simple cable. Also, the download is done very fast, where it takes only a few seconds to complete the whole process.

5 Derived Designs from Miniskybot

In contrast to our previous Skybot robot which remained unmodified for many years, the MiniSkybot has inspired the imagination of the students which have developed new designs in record time. There were two main reason for this motivation, according to the students: 1) Full access to the Miniskybot "source code", 2) Being able to turn their thoughts on real physical objects very fast, thanks to the 3D printer. The former let the students to fully understand a real robot and realize that it is not so difficult to design the mechanical parts. Instead of starting from the scratch, they just simply start modifying the Miniskybot parts. The latter is related to the strong feeling of happiness and power that the students have when they see their designs become a reality.

In the following sections two new derived design are presented, fully created by second year undergraduate engineering students with no special knowledge on mechanics.

5.1 Caterpillator

The Miniskybot robot uses two drive wheels for moving. Two students wonder if it was possible to design a robot with tracks instead of wheels. Inspired by this chain an pinions design in thingiverse[14], Olalla Bravo and Daniel Gomez decided to create the first printable track for mobile robots. After some initial failed tries, they succeed in building a parametric track[15]. The beauty of this

---

[14] http://www.thingiverse.com/thing:5656

[15] http://www.thingiverse.com/thing:7209

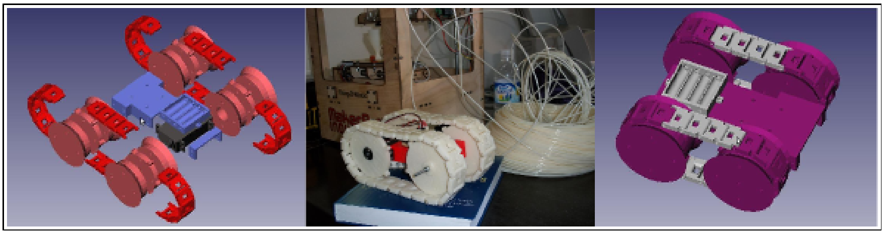

Fig. 8 The caterpillator robot

design was not only its functionality but its property of being parametric. Just changing some parameters, different tracks can be obtained, as well as the necessary pinions. In addition, 3mm plastic spool was used as pins for the links. Therefore no special screws and nuts were necessary.

The latest version is shown in figure 8. It is also available in thingiverse[16], along with some videos showing how it moves.

5.2 UniTrack and F-Track

A different approach was taken by Jon Goitia. He focused on designing robots with articulated tracks. The first design was Unitrack[17], shown in figure 9 (on the left and in the middle). It is an autonomous track driven by a hacked Futaba 3003 servo (the same servo used for the Miniskybot). It consist of two wheels attached to the servo and five standard o-rings used as tracks. Another o-ring is used as the transmission system between the servo and one wheel. Unitrack is also parametric, therefore the wheel's diameter and number of o-rings can be easily changed. This innovative design was for one month the first most popular thing on thingiverse, which is not easy to achieve (currently there are more than ten thousand things!).

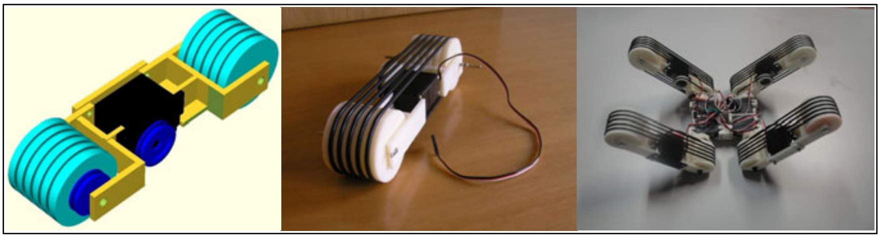

Fig. 9 Unitrack (left) and F-track (right) robots

---

[16] http://www.thingiverse.com/thing:8559

[17] http://www.thingiverse.com/thing:7640

Once Unitrack was fully functional, the F-track robot was created, shown in figure 9 (on the right). It consist of four articulated independent Unitracks joined to a body. This design is an example on how the creativity emerges from some students when they are stimulated.

6 A New Design Paradigm: Evolutionary Robots

Our new robotic platform combines two important features. On one hand it is open hardware, so that anyone can study, modify and distribute the robot. On the other hand the robot is printable making it very easy for the people to materialize it. The result is that anyone in the wold with access to Internet and to an open-source 3D printer can copy the robot, improve it or create derived design.

These features allow the emergence of a new design paradigm in robotics: Evolutionary robots. The robots can now be evolved by the community in the same way the open source movement creates and maintain in a distributed way new software applications, such as the Linux kernel, gnu tools, wikipedia, firefox and so forth. Now it is possible to bring these ideas into the robotics world.

In the previous section we have shown the derived robots created by a group of local students from the Miniskybot robot. It is difficult to imagine and foreseeing the wonderful robots that can be developed by thousand of people around the world collaborating together.

With the Miniskybot robot we have planted the seed. We have already gotten some indications of the potential of this idea: some weeks after the Miniskybot were published on thingiverse, at least three derived design were built. The first was printed by people from Makerbot at the RoboFest 2011 in Baltimore, the second at the FUBAR hacklab space in New Jersey. They are using Roboduino as electronics. The third one was built by CW kreimer[18] for teaching robotics at the Pittsburgh boy scout high tech camp.

7 Results

The Miniskybot robot has been successfully printed on a Makerbot Cupcake 3D printer in ABS plastic (Acrylonitrile Butadiene Styrene). The machine is equipped with the MK5 extruder and a heated build platform. It is very affordable with a total cost of 680€.

All the parts have been printed without raft. The software used was Replicator-G 0023 with Skeinforge 35. In figure 10 a red prototype is shown, along with all the parts needed for assembling the robot.

The total printing time is 2 hours and 50 minutes and the total robot cost is around 57€, as shown in table 1. It can be seen that the cost of the chassis

---

[18] http://www.youtube.com/watch?v=2EqvuPXYKf0

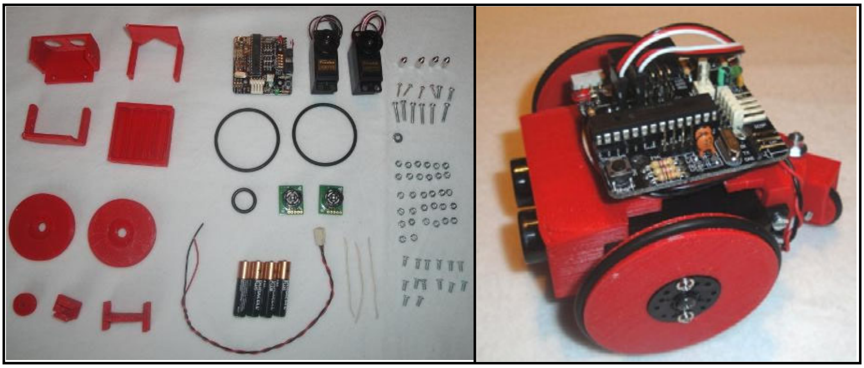

Fig. 10 The MiniSkybot robot printed in red ABS

(the printable parts) is marginal: less than 1€. Therefore, the robot cost is only determined by the cost of the electronics, motors and sensors.

Although this kind of 3D printers are not meant for production but for prototyping, they can be used for building small series of robots for giving courses on robotics to small groups. Given that every 3h the parts for a new robot are built and if the machine is working without interruption, 8 robot chassis per day can be printed. In figure 11 a group of six Miniskybots is shown, printed in different colors.

Table 1 Printing time and cost of the MiniSkybot v1.0 robot

Parts	**Printable**	**Printing time (min)**	**Cost (€)**
Wheels	yes	2x24	2x0.05
Battery compartment	yes	30	0.07
Front	yes	30	0.07
Rear	yes	16	0.04
Battery holder	yes	14	0.03
Castor wheel part 3	yes	12	0.03
Castor wheel part 2	yes	6	0.01
Castor wheel part 1	yes	4	0.01
Wheel O-rings	no	———	2x0.5
Castor Wheel O-ring	no	——-	0.4
SRF02 ultrasound sensor	no	——	11.8
Skycube board	no	——	20
Servo Futaba 3003	no	—	2x9
4 AAA batteries	no	—	2.5
Nuts and bolts	no	—-	2.5
Total:		**170 min (2h, 50min)**	**56.6€**

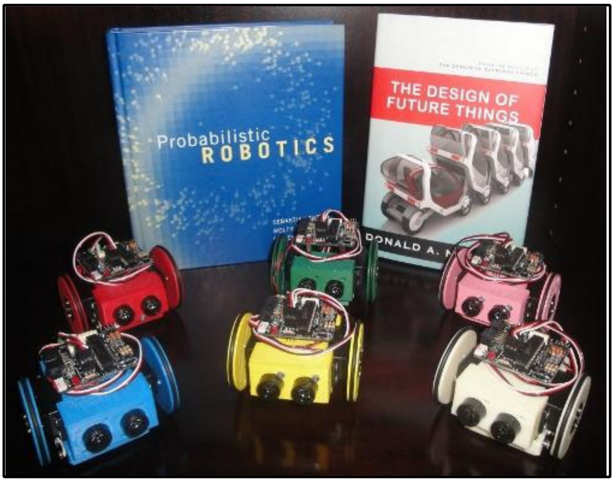

Fig. 11 A group of six MiniSkybot robots (v1.0) in different colors

8 Conclusion and Future Work

Using the latest open source 3D-printers a new printable robotic platform has been designed, built and tested. Our results confirm the viability of these new printable robots. They offer new important features for educational purposes. First, they are very flexible where the students can design new custom pieces easily which are printed and tested very fast. Therefore the robot can be mechanically evolved during the courses. Second, the robot can be thoroughly studied, modified, copied and distributed by anyone. This way the robot can evolute not only in our university but also around the world. This feature is enhanced by the fact that the mechanical parts are Openscad ASCII scripts, like any other software. Consequently, they behave like open source software and can be distributed and shared in a similar way. Finally, the total cost is very low, depending almost exclusively on the servos, electronics and sensors. The Miniskybot v1.0 costs 57€ and the printing time is around 3h, which means that eight robot chassis can be printed per day.

As a future work we are planning to continue evolving the robot in collaboration with our students, designing new parts for adding more sensors as well as creating new derived robots. Currently we are working on a new electronic board, with the same size than the Skycube but compatible with the Arduino software that is becoming more and more popular. In addition, we are developing a new idea on how to design mechanical parts using object oriented programing. We have called it as *object oriented mechanics language* (OOML).

Acknowledgements. We would like to thank Prof. Luis Moreno, head of the department, for supporting this educational initiative. Also we would like to thank all the students who have been involved in building our 3D printer MADRE. Special thanks to: Olalla Bravo, Jon Goitia and Daniel Gomez for their great contributions to the Miniskybot Robot.

References

1. Bowyer, A.: The Self-replicating Rapid Prototyper, Manufacturing for the Masses. In: 8th National Conference on Rapid Design, Prototyping & Manufacturing (June 2007)
2. Bradshaw, S., Bowyer, A., Haufe, P.: The Intellectual Property Implications of Low-Cost 3D Printing. SCRIPTed 5 (2010)
3. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: Usarsim: a robot simulator for research and education. In: 2007 IEEE International Conference on Robotics and Automation, ICRA 2007, pp. 1400–1405 (2007)
4. de Bruijn, E.: On the viability of the open source development model for the design of physical objects. Lessons Learned from the Reprap Project (November 2010)
5. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: 11th International Conference on Advanced Robotics, ICAR 2003, Portugal, pp. 317–323 (June 2003)
6. Goldman, R., Eguchi, A., Sklar, E.: Using educational robotics to engage inner-city students with technology. In: Proceedings of the 6th International Conference on Learning Sciences, ICLS 2004, pp. 214–221. International Society of the Learning Sciences (2004)
7. Gonzalez-Gomez, J.: Modular robotics and Locomotion: application to limbless robots. PhD thesis (December 2008)
8. Hsiu, T., Richards, S., Bhave, A., Perez-Bergquist, A., Nourbakhsh, I.: Designing a low-cost, expressive educational robot. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003, vol. 3, pp. 2404–2409 (2003)
9. Jones, R., Haufe, P., Sells, E., Iravani, P., Olliver, V., Palmer, C., Bowyer, A.: RepRap-the replicating rapid prototyper. Robotica 29(01), 177–191 (2011)
10. Hautop Lund, H., Pagliarini, L.: Robot Soccer with LEGO Mindstorms. In: Asada, M., Kitano, H. (eds.) RoboCup 1998. LNCS (LNAI), vol. 1604, pp. 141–151. Springer, Heidelberg (1999)
11. Lund, H.H., Pagliarini, L.: Robocup jr. with lego mindstorms. In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, pp. 813–819. IEEE, San Francisco (2000)
12. Keith, J., O'Hara, K.J., Kay, J.S.: Investigating open source software and educational robotics. J. Comput. Small Coll. 18, 8–16 (2003)
13. Rawat, K.S., Massiha, G.H.: A hands-on laboratory based approach to undergraduate robotics education. In: Proceedings of 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 2, pp. 1370–1374 (2004)
14. Sklar, E., Eguchi, A., Johnson, J.: RoboCupJunior: Learning with Educational Robotics. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 238–253. Springer, Heidelberg (2003)
15. Verner, I.M., Waks, S., Kolberg, E.: Educational robotics: An insight into systems engineering. European Journal of Engineering Education 24(2), 201 (1999)