## Chapter 1
# A New Open Source 3D-printable Mobile Robotic Platform for Education

J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, 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.

## 1.1 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[13, 11].

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

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.

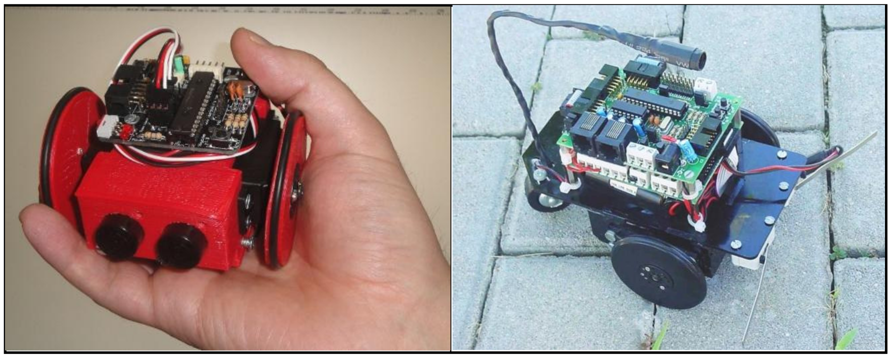

**Fig. 1.1** Left: The new Miniskybot v1.0 robot. Right: The educational skybot robot

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 1.1). 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.

## 1.2 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 [8] 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 [9, 12]. 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[10].

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 IntelliBrainBot [3], Martin F. Schlogl's robots[4], the TankBot[5], the Trikebot [7] among many others.

This had been also our way of teaching robotics during many years, with our Skybot[6] platform (shown in figure 1.1). In our courses, the students

[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

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 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.

## 1.3 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]

## 1.4 The Miniskybot mobile robot platform

The new Miniskybot robotic platform 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...).

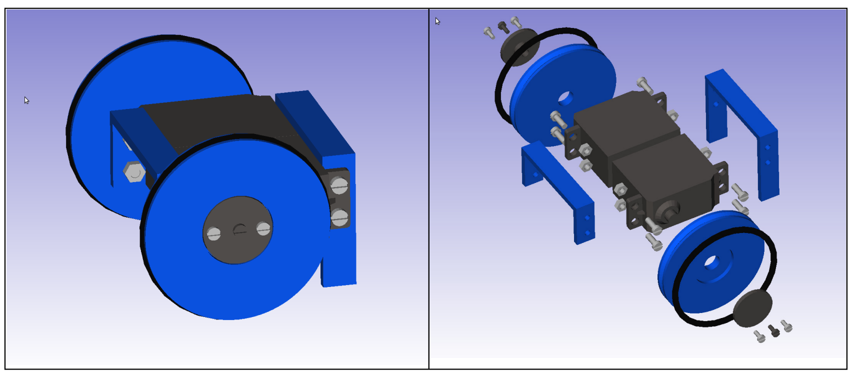

**Fig. 1.2** Miniskybot. Minimal version

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 first is the minimal robot chassis shown in figure 1.2 (It is the "hello world" chassis). 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 minimal robot chassis 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.

The version 1.0 chassis is an evolution of the previous design (figure 1.3). It consist of nine printable parts: the front, the rear, two wheels, the battery

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

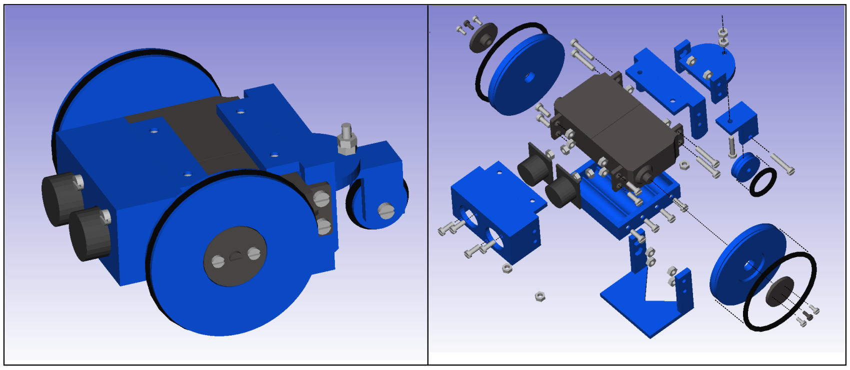

**Fig. 1.3** Miniskybot Robot. Version 1.0

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[8] 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.

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.

The electronic board is also a minimal design with only the necessary components for controlling the robot. It includes an 8-bit pic16f876a microcontroller, 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 1.4).

An electric connection diagram is shown in figure 1.5, 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

[8] http://openscad.org

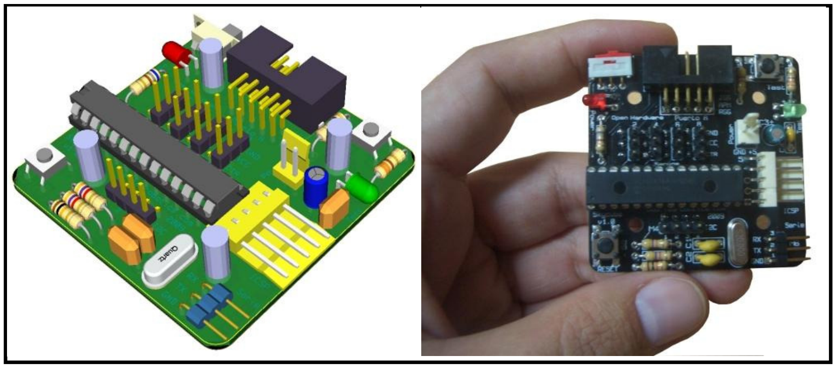

**Fig. 1.4** Electronics. Skycube board

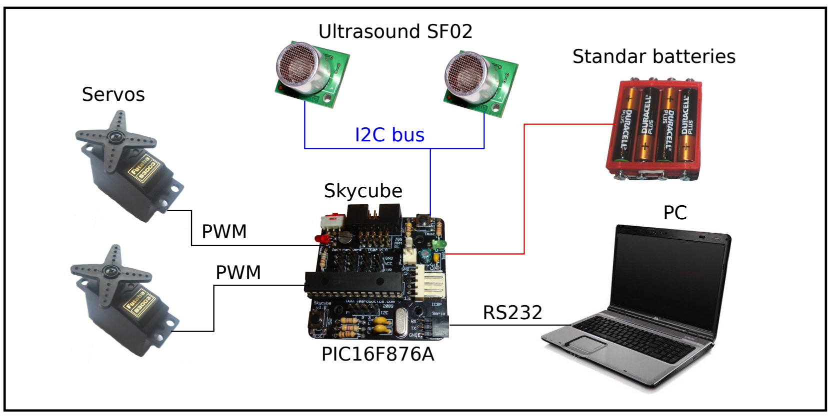


**Fig. 1.5** Electronic diagram

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.

## 1.5 Results

The Miniskybot robot has been successfully printed on a Makerbot Cupcake 3D printer in ABS plastic (Acrylonitrile Butadiene Styrene). The machine is

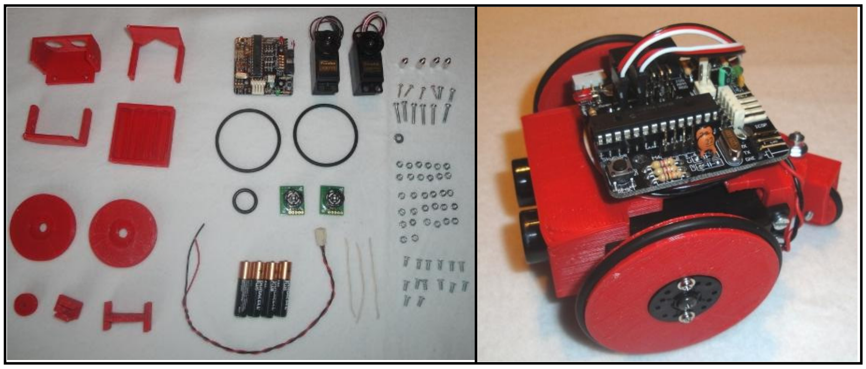

**Fig. 1.6** The MiniSkybot robot printed in red ABS

equipped with the latest 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 1.6 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 1.1. It can be seen that the cost of the chassis (the printable parts) is marginal: less than 1€. Therefore, the robot cost is only determined by the cost of the electronics, motors and sensors.

| 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€** |

**Table 1.1** Printing time and cost of the MiniSkybot v1.0 robot

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 1.7 a group of six Miniskybots is shown, printed in different colors.

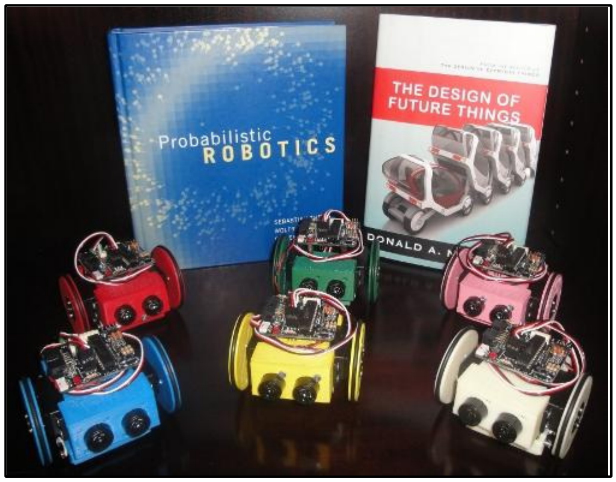


**Fig. 1.7** A group of six MiniSkybot robots (v1.0) in different colors

## 1.6 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 sensors, wheels, whegs and so on.

## 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 contributed to the evolution of the Miniskybot robot. Special thanks to Mrs. Olalla Bravo for the design of the printable castor wheel.

## References

1. Adrian Bowyer. The Self-replicating Rapid Prototyper, Manufacturing for the Masses. In *8th National Conference on Rapid Design, Prototyping & Manufacturing*, June 2007.
2. Simon Bradshaw, Adrian Bowyer, and Patrick Haufe. The Intellectual Property Implications of Low-Cost 3D Printing. *SCRIPTed 5*, 2010.
3. Stefano Carpin, Michael Lewis, Jijun Wang, Stephen Balakirsky, and Chris Scrapper. Usarsim: a robot simulator for research and education. In *2007 IEEE International Conference on Robotics and Automation, ICRA 2007*, pages 1400–1405, 2007.
4. Erik de Bruijn. On the viability of the open source development model for the design of physical objects. Lessons learned from the RepRap project, November 2010.
5. Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *11th International Conference on Advanced Robotics (ICAR 2003), Portugal*, pages 317–323, June 2003.
6. Rachel Goldman, Amy Eguchi, and Elizabeth Sklar. Using educational robotics to engage inner-city students with technology. In *Proceedings of the 6th international conference on Learning sciences*, ICLS '04, pages 214–221. International Society of the Learning Sciences, 2004.
7. T. Hsiu, S. Richards, A. Bhave, A. Perez-Bergquist, and I. Nourbakhsh. Designing a low-cost, expressive educational robot. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2404 – 2409 vol.3, 2003.
8. Henrik Hautop Lund and Luigi Pagliarini. Robot soccer with lego mindstorms. In *RoboCup-98: Robot Soccer World Cup II*, pages 141–151. Springer, 1998.
9. Henrik Hautop Lund and Luigi Pagliarini. Robocup jr. with lego mindstorms. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA*, pages 813–819. IEEE, 2000.
10. Keith J. O'Hara and Jennifer S. Kay. Investigating open source software and educational robotics. *J. Comput. Small Coll.*, 18:8–16, February 2003.
11. K.S. Rawat and G.H. Massiha. A hands-on laboratory based approach to undergraduate robotics education. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1370 – 1374 Vol.2, 2004.
12. Elizabeth Sklar, Amy Eguchi, and Jeffrey Johnson. Robocupjunior: Learning with educational robotics. In *RoboCup 2002: Robot Soccer World Cup VI*, volume 2752 of *Lecture Notes in Computer Science*, pages 238–253. Springer, 2002.
13. Igor M. Verner, Shlomo Waks, and Eli Kolberg. Educational robotics: An insight into systems engineering. *European Journal of Engineering Education*, 24(2):201, 1999.