Lollybot: Where Candy, Gaming, and Educational Robotics Collide

Thomas Tilley[1]

Abstract— **Lollybot is a cheap mobile educational robot made from a game controller that features lollipops as part of its design. Students can construct the robot themselves using simple tools and it was originally built for the African Robotics Network (AFRON) $10 Robot Design Challenge in 2012 where it won the tethered robot category. This paper discusses the design of Lollybot's hardware and software as well as the robot's educational impact since the competition.**

I. INTRODUCTION

Lollybot is an educational mobile robot designed to be built by high school or university students using simple tools and readily available parts for US $8.96. The body of the robot is made from a plug-and-play Universal Serial Bus (USB) game controller with wheels made from scavenged plastic bottle lids, paper-clip suspension, and a tail skid made from a wire coat hanger. The robot's design builds upon previous work by the author using hacked game controllers as cheap interfaces for novel gaming experiences [1], [2]. These earlier projects laid the groundwork for both the robots hardware and software with some existing code developed for these projects being re-used in Lollybot's software.

Lollybot was originally built as an entry in the 2012 African Robotics Network (AFRON) $10 Robot Design Challenge where it won the tethered robot category [3], [4]. The aim of the AFRON $10 Robot Design Challenge was to make robotics more accessible to African students by lowering the cost of robots. The competition encouraged both professional robot builders and hobbyists to build an educational robot for a target price of only US$10. Robots were submitted in three different categories and the designs had to include at least one type of sensor so the robot could interact with its environment.

Lollybot's design includes two different types of sensors. The first type consists of a pair of light detectors mounted beneath the robot that can be used to implement line following behavior. The second is a pair of spherical Chupa Chup lollipops that act as simple bump sensors and also give the robot an appealing crab-like appearance as shown in Fig. 1. The robot was originally called "Suckerbot" because the English translation of the Spanish "Chupa Chup" approximates to "suck suck". "Sucker" is also an informal name for a lollipop in the United States, however, the word also carries some negative connotations so the name was changed to "Lollybot" after the competition.

[1]Thomas Tilley is with Department of Information Technology, International College, Payap University, Chiang Mai, Thailand `thomas_t@payap.ac.th`

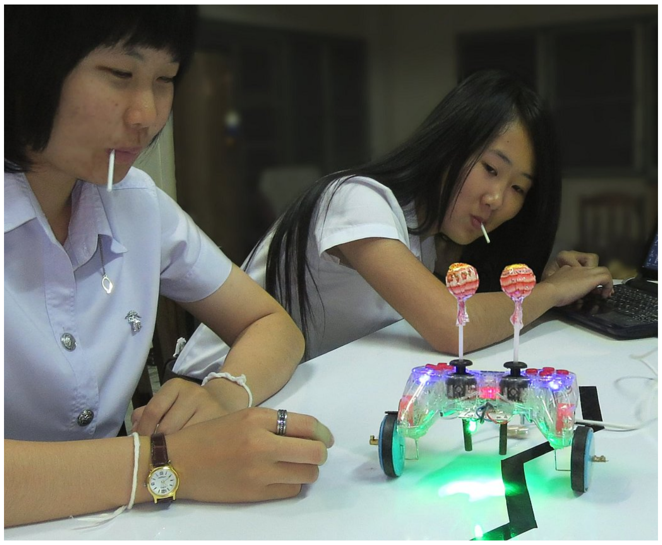

Fig. 1. Thai university students with Lollybot. A high resolution version of this image is available on-line at: `http://www.tomtilley.net/projects/lollybot/icra2015/`

The next section of the paper provides more information on the design of Lollybot's hardware before details of the software are presented in Section III. Section IV then discusses the robot's educational impact since the competition and current work and future directions for Lollybot are described in Section V. Finally, Section VI concludes the paper.

II. HARDWARE

In keeping with the goals of the $10 Robot Design Challenge, Lollybot was designed so that high-school or university level students should be able to be build the robot themselves using readily available parts and tools. The following sections describe the implementation of Lollybot's wheels and sensors as well as some limitations of the current design.

A. Locomotion

Modern game controllers often incorporate "rumble" motors with offset counterweights to provide simple force feedback during game play. The USB game controller used in Lollybot's design includes two 5 volt DC motors which are independently controllable with 255 levels of power. If the motors' counterweights are removed and the sides of the controller are cut off then the exposed motor shafts can be used to drive two wheels providing locomotion.

Initial experiments showed that the motors had insufficient power to move the controller with wheels mounted directly on the motor shafts so some mechanism to increase torque would be required. One approach would be to use gears but buying gears would add to the cost of the robot. The contest also allowed for the use of scavenged parts, provided they could be sourced by students in an African context, and while a DVD player or CD-ROM drive would be a possible source of cheap gears, it was unclear if students would have free access to old players/drives. Furthermore, to find matching sets of gears for both motors it was likely that students would also need two players/drives per robot. The inclusion of gears in the design would also raise the level of manufacturing precision required beyond the ability of most students with access to simple tools.

After many experiments a simple design was found that used wheels made from scavenged plastic lids which were mounted underneath the controller using suspension made from large paper-clips. The edges of the wheels are driven via direct contact with the motor shaft and the increase in torque comes from the difference between the small diameter of the motor shaft and the comparatively larger diameter of the wheel. A rubber band cut from an old bicycle tire is stretched over the wheel to increase friction between the wheel and the drive shaft. The paper-clip suspension also provides a small amount of spring force to keep the wheel pressed against the shaft and this helps compensate for student-made wheels where the center of rotation is slightly off-center.

B. Bump Sensors

Chupa Chup lollipops are essentially a small mass mounted on the end of a plastic rod. By drilling a small hole into each of the analog joysticks (or "thumbsticks") a lollipop can be used to effectively extend the length of the sticks so that they become simple inertia or "bump" sensors. Even with the additional weight of an 11 gram Chupa Chup the springs in the thumbsticks are still able to return to their center position after an impact.

The thumbsticks have two axes and the potentiometer on each axis is connected to an 8-bit analog-to-digital converter (ADC) which gives a deflection range of approximately 127 steps (255/2) in each of the four compass directions. Currently Lollybot only uses a single axis to detect bumps as joystick movements which exceed an adjustable threshold, however, both the X and Y axes are read by the software and it should be possible to compute an approximate collision vector to give a rough indication of the direction of impact.

C. Line Sensors

The two line sensors mounted underneath Lollybot each consist of a light source and a light detector. The light source is a green LED with a current limiting resistor chosen to work with the 5V DC power supplied via the game controller's USB cable. The light detecting half of the sensor is a light dependent resistor (LDR) in series with a 20k ohm resistor that together form a simple voltage divider network. The resistor is mounted inside a small piece of plastic drinking straw to help shield it from stray light. This simple LDR and resistor network are connected to the game controller in place of one of the thumbstick potentiometers. As the level of reflected light beneath the robot varies it acts as if someone is simply moving the thumbstick and the value can be read with 8-bit precision. The two line sensors replace both the X and Y axes on one of the thumbsticks so although there are two lollipops only one actually works as a bump sensor and the other one is purely aesthetic. The circuit containing the two line detectors is built on a literal printed circuit board made from paper and cardboard to help keep the cost and construction difficulty low.

D. Limitations

Lollybot is a tethered robot that relies on the USB cable for both power and communication. Unfortunately, as the robot moves this tether can become tangled over time which shortens its length or the cable may obstruct the wheels. The USB cable on the game controller used for Lollybot is 1.7 m long but by using an extension cable and suspending the point where the two cables meet about 1 m off the ground this gives the robot an effective radius of 1.2 m while minimizing tangling and obstruction.

Another limitation is that Lollybot's motors are not very powerful and while the current design does work, a more robust and reliable wheel drive system is desirable. Using larger wheels to increase torque, as discussed above, reduces the overall speed of the robot. The bump sensor can still detect if the robot has been impacted by a large external force but it is no longer fast enough to detect impacts with the environment resulting from its own movement.

III. SOFTWARE

Lollybot's software reports telemetry from all of the the game controller's buttons and thumbsticks as well as providing control of the two motors. It includes simple line following and bump modes, and students can also steer the robot directly from the keyboard [5].

The original version of the software only runs on Windows-based operating systems. It was written in Object Pascal using the Delphi development environment and re-used code developed in earlier projects. Unfortunately, there are now only commercial versions of Delphi available so students wanting to modify Lollybot's software would need to purchase the development environment or obtain a copy of Delphi illegally.

In mid-2013 a second version of the software was written using HTML 5, JavaScript, jQuery, and Node.js [6]. A Node.js server provides communication with the robot while the telemetry data and control code are presented as a web page that runs in an HTML 5 compatible browser. This client/server architecture means that the robot can now be controlled from a mobile device. In addition, all of the software used is free, open source, and cross-platform so all students need to be able to start modifying Lollybot's code is a text editor. The disadvantage of this version is that the

Fig. 2. Lollybots built by students in Thailand (left) and Ghana (right).

installation of the software is now more complex than simply running the Windows executable of the original software.

IV. EDUCATIONAL IMPACT

A number of Lollybot building workshops have now been held with students in Thailand and Ghana. In March 2013 at Chiang Mai University in Thailand workshops were held on three days where groups of nine students worked in parallel to build a Lollybot in only two and a half hours. See Fig. 2. In September 2013 at Payap University in Thailand students worked in pairs spending two hours per week to build Lollybots over a three week period.

At Ashesi University College in Ghana students have built Lollybots at workshops that were held in May and August 2013. The August workshop was part of the Ashesi Robotics eXperience (ARX) where students spent most of the week working with Lego Mindstorms kits as well as a day building Lollybots [7]. In both Thailand and Ghana the students said they were surprised that they could actually build a robot themselves using everyday items.

In addition to workshops there have also been a number of Lollybot-based undergraduate student projects. A student dissertation by Neequaye at Ashesi University College details the implementation of "Lollybot V2" using a Raspberry Pi and battery pack to remove the limitation of the USB cable length as well as implementing a drag-and-drop programming interface for the HTML5 and JavaScript version of the software [8]. "Everybot " is a senior thesis project at Drexel University in Pennsylvania that re-uses Lollybot's software and extends the hardware [9]. Barnes' design substitutes geared motors for the controller's original vibration motors and also adds an infrared sensor.

Lollybot has also been presented to a range of people outside of academia. In June 2013 the author spoke about the implementation of Lollybot's software to the Thai software development and hacking community at BarCamp Chiang Mai 6. A talk about the robot's development was also presented at TEDx Chiang Mai in September 2013.

V. CURRENT WORK & FUTURE DIRECTIONS

The HTML 5 and JavaScript version of the software discussed in Section III uses a Node.js server to communicate with Lollybot because the client web-page is inside the browser's security sandbox. However, the chrome.usb API now makes it possible to communicate with USB devices directly from the browser and work is underway to create a standalone version of Lollybot's software. This code should simplify installation for students while still being cross-platform, however, the trade-off will be that some mobile devices may no longer be supported.

Some work has also been done to explore the addition of new sensors to Lollybot that could be used to teach students new concepts from physics, computing, and biology. For example, initial experiments show that it should be possible to add a simple audio digitizer to the robot using a readily available electret microphone element. Data from the game controller is sampled at 125 Hz and by patching the microphone into one of the thumbstick ADCs it should be possible to create a simple 8-bit digitizer. Although the sensor would only be able to handle very low frequencies (with a theoretical maximum of 62.5 Hz) it could still be used to demonstrate concepts like quantization, sampling, aliasing, and Nyquist's theorem.

It may also be possible to create a simple heart-rate sensor by re-using the existing line sensors with a finger-tip placed between the light source and the light detector. Alternatively, a simple sensor that uses a cheap piezo element attached to a fingertip could be used to demonstrate heart rate monitoring and biofeedback.

VI. CONCLUSION

Lollybot is a cheap mobile educational robot that incorporates line detectors and a bump sensor for only US $8.96. The design is based around a USB game controller, features two Chupa Chup lollipops, and can be built by students from readily available parts using simple tools. Lollybot was originally built as an entry for the 2012 AFRON $10 Robot Design Challenge where it won the tethered robot category. Since then the robot has been built by students at workshops in Thailand and Ghana.

Additional photographs can be found on-line at `http://www.tomtilley.net/projects/lollybot/icra2015/`.

REFERENCES

[1] T. Tilley, "The Asian Mule in Cyberspace: Building Game Controllers from Locally Appropriate Materials," in *Computer Games, Multimedia & Allied Technology (CGAT'09)*. Research Publishing Services, 2009, pp. 344–351.

[2] ——, "Game Controller Hacking for Fun and Profit," in *Annual International Conference on Computer Science Education: Innovation & Technology (CSEIT 2010)*. Research Publishing Services, 2010, pp. 53–58.

[3] G. A. Korsah and K. Goldberg, "The African Robotics Network and the 10 Dollar Robot Design Challenge," *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp. 116–118, 2013.

[4] ——, "IEEE RAS and the African Robotics Network: The 2014 Ultra-Affordable Educational Robot Challenge," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 13–14, 2013.

[5] (2012) Suckerbot: Telemetry and code for the Suckerbot DIY robot. [Online]. Available: https://code.google.com/p/suckerbot/

[6] (2013) Lollybot: JavaScript + HTML5 version of the telemetry and control software for the Lollybot robot. [Online]. Available: https://code.google.com/p/lollybot/

[7] W. Hussein. (2013) ARX|2013. [Online]. Available: https://code.google.com/p/lollybot/

[8] G. S. N. Neequaye, "An inexpensive, programmable, mobile robotic system for education in Africa," Dissertation, Ashesi University College, April 2014.

[9] R. Barnes. (2014) Everybot: Robotics for everyone. [Online]. Available: http://www.everybot.org