2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

AERobot: An Affordable One-Robot-Per-Student System for Early Robotics Education

Michael Rubenstein[1] Bo Cimino[2] Radhika Nagpal[3] Justin Werfel[4]

***Abstract*— There is a widely recognized need for improved STEM education and increased technological literacy. Robots represent a promising educational tool with potentially large impact, due to their broad appeal and wide relevance; however, many existing educational robot platforms have cost as a barrier to widespread use. Here we present AERobot, a simple low-cost robot that can be easily used for introductory programming and robotics teaching, starting from a primary or middle school level. The hardware is open-source and can be built for ∼$10 per robot, making it possible for each student to have (and keep) their own robot, while still encompassing a rich sensor suite enabling a variety of activities. A free, open-source graphical programming environment allows students without previous programming experience to command the robot. We report on the results of three sessions of a one-week pilot course held in the summer of 2014 by STEM summer camp i2 Camp.**

I. Introduction

Education in the fields of science, technology, engineering, and math (STEM) has received recent emphasis as a key priority, calling for increased engagement and impact in P-12 (pre-kindergarten through high school) classrooms [1]. Computational thinking in particular has long been identified as a critical foundational skill for modern education [2], [3]. The growing ubiquity of computers and related devices has made them attractive as an interactive and flexible tool in a variety of educational contexts; however, one drawback is the imbalance in engagement among student sub-populations, computers empirically having much stronger appeal to "traditional" demographics [4].

Robots, by contrast to computers and traditional programming courses, exert a universal appeal, and are frequently cited as a particularly promising teaching tool to engage underrepresented demographics [5], [6]. One obstacle to widespread adoption of robots in classrooms is cost: many robots used in educational contexts cost hundreds or even thousands of dollars each [7], [8] (Table I). Robotics is also frequently seen as an advanced subject, best encountered first in college, while early exposure can have maximum impact, before preferences and societal influences on a student are more established [4].

Here we present the AERobot (Affordable Education Robot), a low-cost introductory robotic platform that can be used at the primary or middle school level. Key advantages of the AERobot include:

- Its *low cost* (∼$10 each, if produced in quantities of 1000) makes it accessible for classrooms in a wide range of socioeconomic settings. Each student can have—and keep—their own robot, increasing feelings of ownership, and reducing cases where some students dominate a team activity and others are left out.
- At the same time, it has a *rich sensor suite* compared to other very low-cost robots, enabling a wide variety of activities.
- The one-robot-per-student framework and physical design lends itself to related activities like artistic decoration of the robots, further contributing to feelings of ownership and engagement and broadening the appeal.

AERobot was motivated by the African Robotics Network (AFRON) Ultra Affordable Educational Robot Project, whose goal is the creation of educational robot systems an order of magnitude less expensive than existing options. AFRON awarded the AERobot first place in the software category, and second place each in the hardware and curriculum categories, in its 2014 Design Challenge [9].

Section III presents the hardware design; section IV summarizes the graphical software interface. In section V we discuss the results of a one-week pilot course conducted in 2014 with rising fifth through eighth graders through STEM summer camp i2 Camp (http://i2camp.org/).

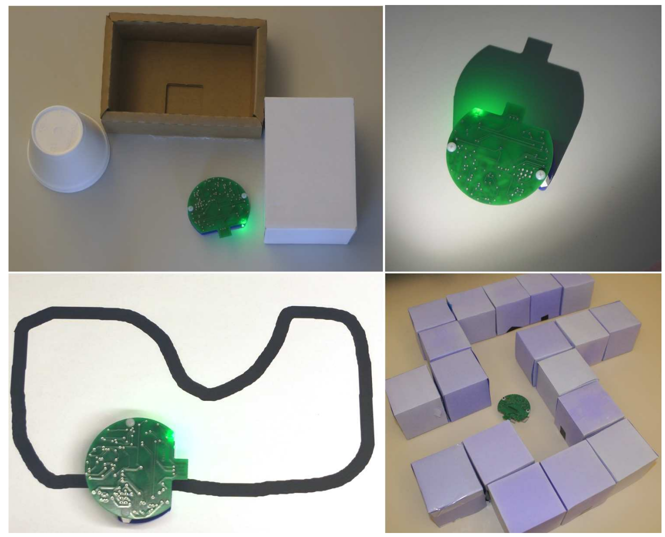

Fig. 1. Examples of behaviors the AERobot can be programmed to perform. Clockwise from bottom left: line-following, wall-following, phototaxis, maze-solving.

[1]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA mrubenst@seas.harvard.edu
[2]South Dakota State University, Brookings, SD bo.cimino@jacks.sdstate.edu
[3]School of Engineering and Applied Sciences and Wyss Institute, Harvard University, Cambridge, MA rad@seas.harvard.edu
[4]Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, MA justin.werfel@wyss.harvard.edu

978-1-4799-6922-7/15/$31.00 ©2015 IEEE

Robot	Proximity sensing	Bump sensing (physical)	Light sensing	Line sensing	Accelerometer	Gyroscope	Speaker	Microphone	Camera	LEDs	Extendable design	Target audience	Cost (USD)
Khepera III [8]	✓		✓	✓						✓	✓	College+	3120 [1]
e-puck [10]	✓				✓		✓	✓	✓	✓	✓	College	1017 [1]
Scribbler 2 [11]	✓		✓	✓			✓	✓		✓	✓	Ages 14+	130 [1]
Scribbler 2 + IPRE Fluke2 board [11]	✓	✓	✓	✓			✓	✓	✓	✓	✓	College	229 [1]
Thymio II [6]	✓			✓	✓		✓	✓		✓		Ages 4-18	150 [1]
LEGO Mindstorms EV3 [12]	✓	✓	✓	✓		✓	✓			*	✓	Ages 10+	350 [1]
Kilobot [13]			✓							✓		College	148 [1] / 14.05 [2]
r-one [7]	✓	✓	✓		✓	✓	✓			✓		High school+	247 [2]
Lollybot [9]		✓		✓								Elementary school+	8.96 [2]
MIT SEG [9]				✓							✓	High school	20.27 [2]
Aerobot	✓		✓	✓						✓		Ages 10+	10.70 [3]

*LCD display

[1]Retail
[2]Parts only
[3]Parts+assembly

TABLE I
EXAMPLES OF ROBOT PLATFORMS USED FOR EDUCATION, WITH SUMMARY OF CAPABILITIES, INTENDED AUDIENCE, AND COST

II. RELATED WORK

Many robot platforms have been designed or used for education (Table I). In some cases, cost on the order of US$1000+ per robot [8], [10] presents an obstacle to widespread adoption and use. Such robots are typically used by students at the college or more advanced levels. Lower-cost robots, on the order of US$150+ each, have been developed to make their use more widely accessible, and to enable use by younger students [6], [7], [11]. The most widely used robot system in pre-college education, LEGO Mindstorms [12], focuses emphasis on building and mechanical design, frequently at the expense of programming and related topics. While these latter robots are more broadly affordable to many classrooms, cost still limits their use in some settings, and typically necessitates groups of students sharing a single robot; the vision of a personal educational robot has typically not been feasible below the college level [11]. More recently, very-low-cost platforms, on the order of US$10 each, have been developed with the goal of making very widespread use of classroom robots a real possibility [9]. A limitation of most such robots is heavily restricted sensing and actuation capabilities [9], [13], limiting their pedagogical usefulness. The AERobot's design intent is to provide a wide set of capabilities, to lend itself to use for many classic introductory robotics activities, while keeping the cost at a fraction of that of comparable robots.

Table I shows characteristics of several educational robots.

III. HARDWARE

AERobot's design goals involved three chief criteria: (1) to be very low cost, to maximally allow widespread adoption; (2) to support a wide variety of behaviors, for maximal flexibility and breadth in classroom activities; (3) to be robust and easy to operate, to best support use in middle school classrooms. The first two criteria are generally conflicting—e.g., increasing the robot's capabilities typically involves adding sensors or motors and thus increasing cost—so the design needed to find a balance. We specified several functions typical for educational robots that the robot needed to be able to perform or emulate: line sensing, bump detection, distance sensing from obstacles, and directional light sensing. These functions would support having students program the robot to perform nontrivial actions like line-following, phototaxis, wall-following, and maze solving (Fig. 1; accompanying video). We also set a cost goal of approximately $10 per robot for large production volumes [9].

Several key points in the design contribute to achieving these goals. The printed circuit board (PCB) doubles as the main robot chassis, reducing cost and complexity. All components are mounted on a single side of the PCB, and can be automatically attached with a pick-and-place machine. All remaining assembly steps can be quickly and easily performed by the student without needing tools. Vibration motors rather than wheels for locomotion reduce cost. The robot uses purely optical sensors, greatly reducing cost; the lack of moving parts for sensing also increases reliability and robustness to dirty and dusty environments, important for

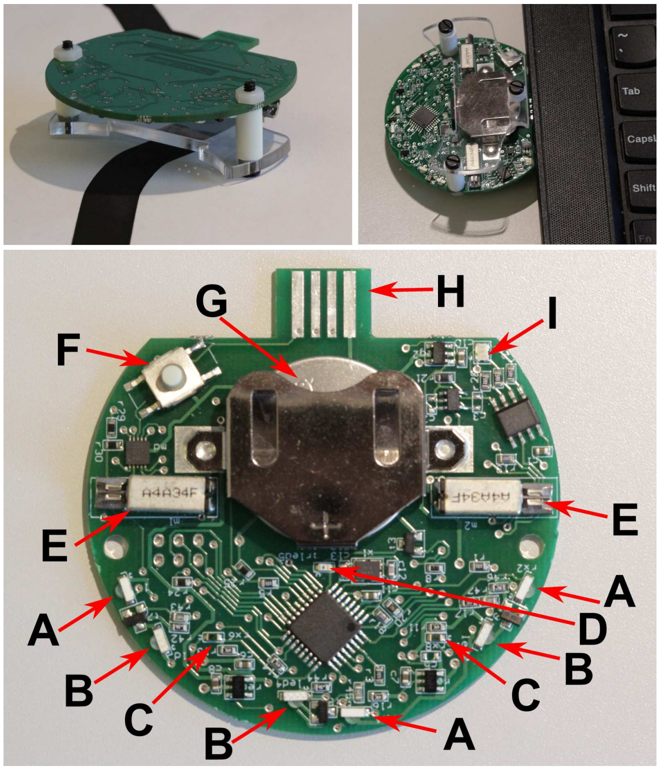

Fig. 2. (Top left) AERobot in nominal orientation following a line on paper. (Top right) AERobot plugged into a USB port for charging or programming. (Bottom) Bottom view of AERobot circuit board (plastic foot removed) showing (A) outward-facing IR/ambient light receivers, (B) outward-facing IR transmitters, (C) downward-facing IR sensors, (D) downward-facing IR transmitter, (E) vibration motors, (F) power switch, (G) lithium-ion rechargeable battery, (H) USB connector, (I) RGB LED.

classroom use by younger students. A USB interface built directly into the PCB eliminates the cost of, and the risk of losing or needing to replace, separate cables, adapters, or other hardware otherwise needed to program and charge the robot.

A. Capabilities

Each robot is controlled with a built-in 8-bit microcontroller, and powered by a rechargeable lithium-ion battery allowing for two or more hours of use between recharges. The robot's inputs are three pairs of retro-reflective infrared (IR) sensors facing outward (Fig. 2A,B), and one downward-facing IR emitter and two receivers (Fig. 2C,D). Its outputs are one RGB LED (Fig. 2I) and two vibration motors (Fig. 2E).

Each of the three pairs of outward-facing IR sensors contains a transmitter (LED) and a broad-band phototransistor sensitive to IR and visible light. One pair faces forward; the other two are at 45° to the left and right. These sensors serve multiple purposes: (1) They detect the presence of and distance to obstacles in the environment. Working in conjunction with its IR LED, each phototransistor can detect the amount of IR light reflected back from obstacles in the environment, using the reflected light intensity to approximate distance to the obstacle in that direction. In classroom settings, this allows the robot to detect obstacles as far as 8 cm away. (2) They emulate bump detection. Using a reflected-intensity threshold corresponding to very short distances lets the robot report that it has "contacted" an object, without adding separate mechanical bump sensors to the hardware (increasing cost and fragility). (3) They detect the amount of incoming ambient light in each direction, using the phototransistors alone with the transmitters off. This capacity lets the robot orient toward a bright external light source.

The downward-facing IR sensors act to detect the position of a dark line beneath the robot. The LED in the center reflects IR light off the surface below to the two receivers symmetrically offset from the robot's sagittal (left/right) axis. By measuring the amount of light received, the reflectivity of the point on the table below each receiver can be approximated; comparing the values for the two receivers allows the robot to determine on which side a line is located.

For both the obstacle distance sensors and the line sensors, the returned value is a function of the ambient light levels and the obstacle distance or table reflectance, respectively. A high-pass filter is used to remove this dependence on ambient light. To avoid the cost of implementing a high-pass filter in hardware, we approximated one in software. This is done by sampling the sensor value before the transmitter is turned on, and subtracting this value from the sensor value when the transmitter is on. This procedure adequately removes the effect ambient light has on these sensors in practice.

The output LED color can have each channel (R/G/B) independently set to one of 256 intensities.

For locomotion, AERobot makes use of two vibration motors, as commonly found in cell phones, to create slip-stick locomotion [14]. These allow it to move on a flat, smooth surface, such as a table, sheet of paper, or whiteboard. Each motor is independently driven by a pulse-width modulation (PWM)-controlled H-bridge, allowing control over the speed and direction of each motor. When both motors spin in the same direction, the robot moves forward or backward; asymmetric motor activation can let the robot turn in place. Compared to a more traditional geared DC motor with a wheel attached, these vibration motors are considerably lower cost and require no assembly, but have the drawback of being slower and harder to control.

The difficulty in controlling robot movement with these vibration motors is due to the lack of odometry, and the fact that the AERobot hardware provides no direct control over motor speed, only average motor voltage. Due to high part variability of the motors and driving electronics, a given PWM signal can create a wide range of motor speeds on different robots. These range from zero speed due to the applied voltage being unable to overcome internal motor static friction, to a speed to which causes the motors to spin fast enough to temporarily lift the robot off the table (bringing it outside of the slip-stick operating regime), making it uncontrollable. To correct for this variability, we created a program which allows a student to calibrate the motor speeds to achieve the desired robot motion. This calibration program is loaded onto the robot, and the student touches the bump sensors to adjust the motor speeds until the robot moves smoothly in the appropriate direction. Once the student is satisfied with the calibrated motor speeds, they are saved in the robot's non-volatile EEPROM, and are used to set the motor speeds whenever a motor command is used in a student program. This calibration allows for effective motor function despite the variability in components and on a wide range of possible surfaces.

B. Operations

In addition to the robot functional capabilities used for programs, it is important to consider how robot operations such as programming and recharging are handled. A major feature of AERobot is that its circuit board has a built-in USB interface that allows it to directly plug into any computer with a USB standard type "A" port. Using this USB connection, it can recharge its lithium-ion battery using the USB power bus and be reprogrammed using the USB data with the use of a VUSB-based bootloader (http://www.obdev.at/products/vusb/usbasploader.html), all without requiring any additional hardware beyond a standard computer—avoiding the need for an external programmer and charger, which could easily double the system cost. Building the USB interface directly into the circuit board also obviates the additional costs of a USB connector and cable, which would increase cost by an estimated 10%. Moreover, by bypassing the need for such external components, the fully self-contained design avoids the attendant risk of losing them, perhaps especially important for a classroom robot intended for younger grades.

C. Assembly

Besides the cost of robot components, the cost of assembly is also a critical, but often overlooked, factor in overall robot cost. Accordingly, all AERobot electronics are designed to use only commonly available surface-mount components, including the motors and battery case, which can all be placed using an automatic pick-and-place machine, drastically reducing assembly costs. The only exception to this is the plastic "foot", which is a laser-cut piece of acrylic, which can be produced in a highly efficient and automated fashion. Additionally, all components are mounted on a single side of the circuit board, cutting assembly cost in half when compared to those with components on both sides. All remaining assembly steps (inserting the battery, and attaching the foot to the circuit board with plastic screws, washers, and standoffs) are very simple, require no tools, and can be done by the student in a few minutes.

The hardware details for AERobot, including circuit board design and a parts list, are freely available at https://sites.google.com/site/affordableeducationrobot under an open-source license. The cost breakdown for the robot hardware is shown in Table II. All part prices are assuming volume purchase of 1000 robots using common part distributors. We expect the robot cost would be greatly reduced if parts are acquired directly from manufacturers.

Parts	Cost (USD)
Motors	2.60
Battery	1.50
Microcontroller	1.30
PCB and assembly (estimated)	1.67
Various electronics	3.26
Mechanical components	0.37
Total	10.70

TABLE II
PART COST PER ROBOT FOR 1000 ROBOTS

IV. SOFTWARE

For greatest ease of use by students with no previous programming experience, we provide a graphical programming language (Fig. 3). This software is a modification of the open-source package Minibloq (http://blog.minibloq.org/), modified to support AERobot-specific hardware and functions. Graphical programming languages allow novice programmers to focus on logic without getting stuck on less fundamental issues like syntax. The student uses a point-and-click interface to manipulate program elements called blocks, corresponding to standard programming constructs like loops and conditionals as well as robot sensing and action abilities.

Blocks are available corresponding to each of the robot's capabilities and functions: distance returned by each of the three obstacle sensors; boolean value for each of the three bump sensors; intensity of ambient light detected by each of the three light sensors; position of a dark line underneath the robot (left, right, center, or none); move forward or backward, turn left or right, or stop moving; set the LED to one of eight colors or off.

After writing a program on the computer, the student transfers it to the robot by plugging the robot into the computer and clicking a button within Minibloq that loads the active program onto the robot. The robot contains a USB bootloader which allows Minibloq to directly program a robot plugged into the USB port.

Minibloq has a feature that regulates its syntax: only relevant blocks will appear as choices according to their context in the program. For example, after clicking on a "move" block, only the "forward", "backward", "left", "right", and "stop" blocks are next available; sensor-value blocks are available in conjunction with value-comparison blocks, but not at the beginning of a new command. This feature simplifies the interface and prevents syntax errors.

The Minibloq environment lends itself to preparing students for text-based programming if they or instructors desire. A button toggles visibility of C code which is automatically and dynamically generated by Minibloq corresponding to the graphical program the student is writing. This feature helps the student become familiar with C syntax. Additionally, the graphical program layout imitates that of a text-based program: e.g., indentation of blocks follows good programming practice, and the order of blocks in variable assignment statements matches that in most text-

Fig. 3. Graphical programming language, based on Minibloq (minibloq.org). (A) A simple example program. The robot will turn its LED red and back up if it detects an obstacle too close to its center distance sensor; otherwise it will turn its LED green and move forward. (B) The action pane, giving categories of blocks that can be used to start a new statement: while loops, for loops, conditionals, delays, LED-color commands, motor commands, variable initialization and variable assignment (both floating-point and Boolean), comments. (C) A block starting a statement provides a context-specific selection of blocks to continue the statement: e.g., a motor block can be followed by a block specifying forward, left, or backward movement. (D–E) Different blocks specify the type of sensor (D: bump, distance, ambient light, line) and its location (E: left, center, right).

based languages. More advanced users can currently create programs for the AERobot directly in the Ardunio development environment, or using C; in future work we intend to add a more user-friendly environment for intermediate users to program the AERobot in C or Python.

V. Pilot Program

In July and August 2014, STEM summer camp i2 Camp (http://i2camp.org/) ran three sessions of a one-week pilot course "BugBots: Programming Mini-Robots" (Fig. 4) which we developed using the AERobot hardware and software.

Content: The course was intended as an introduction to programming and robotics for middle school students with little or no previous experience with either. The overall framework treated robots from the perspective of artificial insects (as implied by the title). This context allowed activities to be introduced in the language of different "behaviors", which together with the sensors supported not only classic robotics activities (e.g., bump-and-turn, line-following, wall-following, light-following, maze solving) and the basic programming and robotics concepts that supported them (e.g., variables, conditionals, loops, debugging, open- vs. closed-loop control) but also more traditionally advanced concepts and philosophical explorations (e.g., behavior-based robotics, Braitenberg vehicles [15]; what is a robot? Could an insect be considered a robot? What is intelligence?). The framework and low cost also lent themselves to students decorating the robots as a key activity (Fig. 5), an element of personalization and artistic expression that proved especially popular.

Participants: A total of 41 students participated in the three sessions. The first two sessions (17 rising 5th and 6th graders; 14 rising 7th and 8th graders) were held as part of a free program supported by the Hayden Foundation, in which students were assigned to this and another course without a choice of their activities. Thus the experience from these sessions may be more representative of use of these robots and material in a "typical" classroom setting as a STEM activity. Additionally, participants in these sessions were from lower-income, inner-city families, chosen as otherwise unlikely to participate in such a summer course. The third session (10 rising 5th and 6th graders) was held with students who signed up for this topic due to personal interest, and thus may be more representative of self-selected groups in usual robotics activities like after-school programs. Below we refer to these cohorts as the "general" and "self-selected" groups, respectively.

Students filled out pre- and post-course surveys about their previous experience and their evaluations of the week. Not all surveys were turned in complete; we report below on available data.

Participants were ages 10–15, with 22 boys and 19 girls (15 boys/16 girls in the general group, 7 boys/3 girls in the self-selected group) (Fig. 6). The majority of students in both groups self-reported no previous experience with robotics (6 out of 24 responses in the general group reported previous experience, 3 out of 8 in the self-selected group). Many in the general group reported previous experience with programming (14 out of 24), fewer in the self-selected group (2 out of 8).

Evaluations: With the student post-course surveys, we hoped to gather data on student responses to the course and what they learned, feedback on the most effective aspects, and suggestions for improvements to both the course and the robot. We anticipated that especially popular aspects would be decorating the robots and getting to keep them at the end of the week, and that points of criticism would be the need for motor calibration and the robots' slow movement speed.

The course-specific post-course survey included several statements with which students indicated their agreement on a four-point Likert scale (Table III). Students indicated overall satisfaction with and enjoyment of the course, and increased understanding of and interest in both programming and robotics. Of particular note was the very strong positive response to being able to keep the robot.

The post-survey also asked "What do you wish the robot would do that it can't—what would you add if you could?" These suggestions were less directly useful for identifying potential hardware improvements than we had hoped (Fig. 7C)—the most common requests (10 out of 23) were to have the robot fly, levitate, jump, or flip. Only two students asked for the robot to move faster.

Fig. 4. Students participating in the AERobot-based i2 Camp pilot courses.

Fig. 5. Examples of robots decorated by the students.

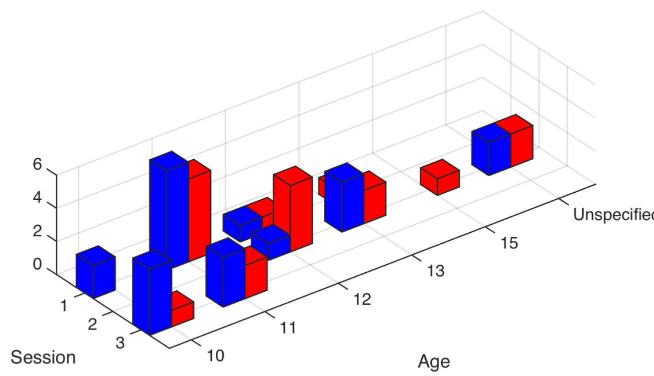

Fig. 6. Demographics of the three pilot sessions: number of boys (blue) and girls (red) of each age.

I had fun in this course.	3.6 ± 0.5
I liked using the Aerobot.	3.6 ± 0.5
I'm glad I get to keep my robot.	3.8 ± 0.4
I plan to keep playing with my robot at home.	3.4 ± 0.7
After taking this course, I now have a better understanding of programming.	3.3 ± 0.6
I'm now more interested in programming.	3.1 ± 0.8
I now have a better understanding of robotics.	3.5 ± 0.7
I'm now more interested in robotics.	3.4 ± 0.7

TABLE III
OVERALL PARTICIPANT RESPONSES TO VARIOUS QUESTIONS, ON A FOUR-POINT LIKERT SCALE (1–4: "NO!/NOT REALLY/YES/ABSOLUTELY!")

Additionally, all i2 Camp students completed a standard post-course evaluation including several elements:

- A response to the question "How do you feel about this course now that you've taken it?" on a four-point Likert scale (1–4: "I Really didn't like it/It was OK/I liked it/I loved it!").

The overall average rating was 3.5 ± 0.7. No statistically significant differences were found between ratings by general (mean 3.5) and self-selected (mean 3.4) groups, or between boys (mean 3.4) and girls (mean 4.0).

- Free responses to queries about their favorite and least favorite parts of the course.

Multiple students named taking the robot home as their favorite thing about the course. Others cited the activity of decorating the robot (Fig. 7A). The most common element cited as least favorite (3 out of 31 responses) was the act of calibrating the robots' motors (see discussion in §VI below). The most common response (8 out of 31) was to write that they had no least favorite part (Fig. 7B).

- Free responses to the question "Room for improvement: What would you take away, change or add to make this course better?".

The most common response (9 out of 31) was that nothing should be changed about the course (Fig. 7D). Two other students requested that the course be longer.

- A response to the question "After this course are you more or less interested in a career in Science, Technology, Engineering, or Math?".

One student checked "Less Interested", 14 "The Same", 13 students "More Interested", 1 "I don't know". No statistically significant differences were found between responses by general and self-selected groups, or between boys and girls.

- Free responses to the question "Is there anything else you'd like to tell us about your experience this week at i2 Camp?"

Responses were overwhelmingly positive (Fig. 7E).

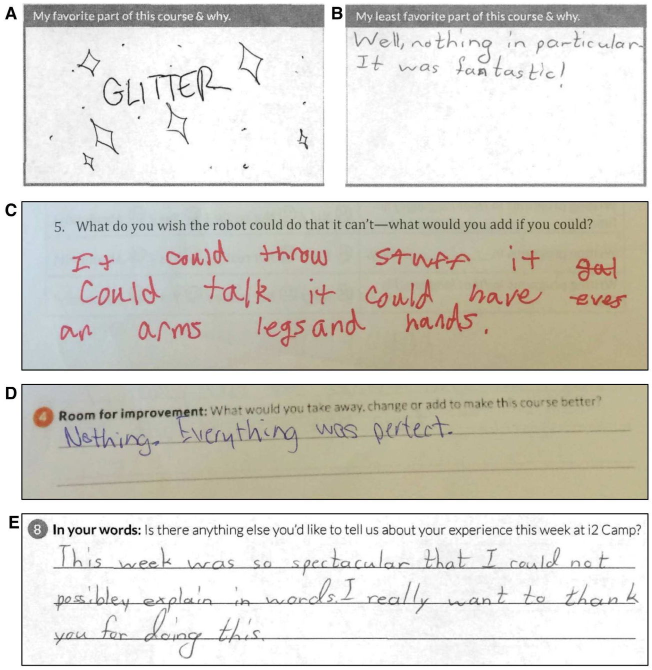

Fig. 7. A few of our favorite responses to surveys. (A) Decorating the robots was a popular element. (B) Many students could think of nothing negative to say about the course. (C) Suggestions for changes to the robot were generally more enthusiastic than realistic (for a low-cost, one-per-student robot). (D) A plurality of students requested that no changes be made to the course. (E) The overall response to the course by participants was extremely positive.

VI. Conclusions and Future Work

We believe that AERobot constitutes a useful platform for early STEM education. Its design strikes a balance between cost and capabilities, enabling it to be an effective tool. Its low cost allows its use in a wide range of socioeconomic settings, and enables a one-robot-per-student framework allowing personalization and increasing student feelings of ownership and engagement with their robots. It can be used effectively by younger middle school students, helping to make hands-on robotics experience available to younger ages than currently typical. Pilot classes have demonstrated success with the target age group, ability to increase student interest and understanding, and appeal to both boys and girls and to both general and self-selected audiences.

The most important issue for improvement, as identified by the pilot experience (teacher as well as student comments), is motor calibration. Some students found this step inconvenient and frustrating; however, a poorly calibrated robot moves unpredictably, a still more frustrating experience. Improving the usability of the software calibration routine could significantly improve the experience for these students. A possible hardware-based approach to eliminating this issue might be to add an accelerometer (∼$1), which could sense vibrations from motors and use this feedback for automatic self-calibration or ongoing adjustment of motor speed.

Other issues for software improvements include an integrated environment to program the robot using text-based languages like Python or C directly, and ports to a wider range of computer operating systems. Access to lower-level functionality of the robot could also be a useful pedagogical tool for more advanced users.

We are currently working to address these issues and to extend the use of AERobot to programs more broadly nationwide. We hope that AERobot and/or future robots based on its design (hardware and software both open-source) will help students, especially those with fewer opportunities, to develop interest in STEM topics and gain hands-on experience in robotics.

Acknowledgment

We thank i2 Camp for supporting “BugBots” among their courses, and Jerry Shaw and Chris Roche for teaching the pilot courses. This work was supported by the Wyss Institute for Biologically Inspired Engineering.

References

[1] U.S. Department of Education, “Science, technology, engineering and math: education for global leadership,” http://www.ed.gov/stem, 2014.
[2] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980.
[3] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.
[4] J. Cooper, “The digital divide: the special case of gender,” *Journal of Computer Assisted Learning*, vol. 22, pp. 320–334, 2006.
[5] I. A. Arroyo, A. A. Arroyo, and E. M. Schwartz, “The integration of robotics in grade school to promote interest in math and science,” in *Florida Conference on Recent Advances in Robotics*, 2003.
[6] F. Riedo, P. Rétornaz, L. Bergeron, N. Nyffeler, and F. Mondada, “A two years informal learning experience using the thymio robot,” in *Advances in Autonomous Mini Robots*, 2012, pp. 37–48.
[7] J. McLurkin, A. Chou, W. Li, M. John, C. Licato, N. Okeke, J. Rykowski, S. Kim, J. Shen, N. Chen, Q. Kaseman, A. Becker, L. Langford, and J. Hunt, “r-one: An advanced, low-cost multi-robot system,” in *IROS*, 2013.
[8] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, “A review of robotics kits for tertiary education,” in *Proc. 3rd Intl. Wksp. Teaching Robotics & Teaching with Robotics*, 2012, pp. 153–162.
[9] African Robotics Network, “The AFRON “10 dollar robot” design challenge,” http://robotics-africa.org/afron-design-challenges.html.
[10] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, “The e-puck, a robot designed for education in engineering,” in *Proc. 9th Conference on Autonomous Robot Systems and Competitions*, 2009, pp. 59–65.
[11] T. Balch, J. Summet, D. Blank, D. Kumar, M. Guzdial, K. O’Hara, D. Walker, M. Sweat, G. Gupta, S. Tansley, J. Jackson, M. Gupta, M. N. Muhammad, S. Prashad, N. Eilbert, and A. Gavin, “Designing personal robots for education: hardware, software, and curriculum,” *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 5–9, 2008.
[12] M. Rollins, *Beginning LEGO MINDSTORMS EV3*. Apress, 2014.
[13] M. Rubenstein, C. Ahler, and R. Nagpal, “Kilobot: A low cost scalable robot system for collective behaviors,” in *ICRA*, 2012.
[14] P. Vartholomeos and E. Papadopoulos, “Analysis, design and control of a planar micro-robot driven by two centripetal-force actuators,” in *ICRA*, 2006.
[15] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. Cambridge: MIT Press, 1984.