Bridging the Educational Gap: A Computer-Free, Cost-Effective Approach to Democratizing Robotics Learning

Rohan Jay
Princeton International School of Mathematics and Science

Abstract

This paper demonstrates a robotics kit to teach students the basics of programming and robotics. The kit's advantage lies not only in the lower cost of production but also in the fact that it does not require a computer or laptop to program, meaning students who don't have access to these can still use it. By using an IR remote for input, an attached programming guide walks users through basic movements, for-loops, and if-statements, as well as an optional bang-bang control. The kit itself is made up of a microcontroller board, gearbox DC motors, and IR sensors. The IR sensors — one on each side of the front — are utilized for the if-statements and bang-bang control. For the if-statement, a digital value is read depending on whether the IR reflectivity of the surface below exceeds a certain threshold, which is used as a Boolean value; for bang-bang control, the IR sensors are used to follow a curve using the bang-bang algorithm, variations of which are used in self-driving cars.

Introduction

One active, problem-based, team-centered approach to learning is robotics. Robotics relates well to current thinking in engineering and computing education (e.g., Striegel and Rover, 2002; IEE Problem Based Learning Initiative, 2003; Vandebona & Attard, 2002), where problem-based learning is viewed as integrative and as enhancing team-working skills. Students learn through doing; they discover knowledge and develop understanding through their own activities, guided by a teacher, rather than 'receiving' it from a teacher through instruction. Hence, hands-on robotics fits well within the constructivist view of learning (Piaget and Inhelder, 1966; Papert, 1980; Harel and Paper, 1991), which portrays learning as the acquisition or 'construction' of knowledge through observation of the effects of one's actions on the world.

In robotics, students' learning is concrete, associated with phenomena they create, observe, and interact with. Problems are open-ended, permitting many solutions and many approaches. Hence, robotics affords opportunities for learning problem-solving techniques and processes, integrates a number of domains, exposes realistic constraints and issues, and leaves room for creativity. Robotics, when supported by guidance, access to models and examples, suitable task structure, and encouragement of self-reflection, can provide a vehicle for guiding students toward an effective understanding of programming principles.

This paper details a robotics kit for low-income students that does not need a computer to program. The kit is also intended to be relatively cheap compared to most kits. In addition, there is a complementary guide that covers basics, for-loops, and if-statements, which the guide goes over using its own programming language as input. For if-statements, the kit contains photoresistors to detect the brightness level as input, returning a 1 if it exceeds a threshold and 0 otherwise. Students can either program the assembled robot with the programming language and an IR remote or, for more advanced users, with the Arduino language by overwriting the kit's interpreter. The robot also stores programs written in the former in Electrically Erasable Programmable Read-Only Memory (EEPROM) whenever they are run so that when the robot is

turned off, the data will be saved. The IR remote also has a backspace button if students make errors or changes. One other thing unique about this robot is that it has a setting to demonstrate bang-bang control, a line-following algorithm of which a variation is used in self-driving cars.

Related Work

Robotics in Education

Several other projects have had the same or similar goals. For example, [1] demonstrates a tile-based tangible programming language called "Quetzal," which can be compiled into a LEGO Mindstorm robot using a portable scanning machine. The authors of [1] note that one of the advantages of Quetzal is that students do not need to gather around a laptop, facilitating cooperation among students. Unfortunately, our kit does not have this same advantage.

[2] presents something similar to [1], except it gamifies the programming experience without a robot and uses the camera on a tablet for computer vision as opposed to a scanning machine.

[5] takes a more applied approach to robotics education by doing a pedagogical study; however, [5] is similar to our goal since it involves students who do not have previous experience in robotics. [5] is, like this paper, about a low-cost robotics kit, but also does a pedagogical study to go with it. Unlike the kit proposed in this paper, the kit in [5] is more malleable and adaptable, similar to kits like Arduino. The authors conducted tests with secondary school students and teachers using three different construction systems made out of small aluminum T-slots to connect beams — one design that uses screws, one that uses friction, and one that uses both. From the results of the evaluation, the hybrid system was the most preferred. With the hybrid system, the T-slots allow for initial adjustability of the position, and the screw system attached to a T-slot allows for easy tightening and fixing of the position.

[6] explores the use of LEGO EV3 robots in teaching Newtonian physics in an after-school STEM education program. The study involved 74 teenagers and used quantitative methods to analyze the data. The findings of the study showed that the use of LEGO EV3 robots had a positive impact on the students' understanding of Newton's second law.

Our Work

The crux of our project is to contribute to the democratization of robotics education. Many barriers exist for students to learn robotics. One of these barriers that we address is the need for consistent access to a computer. Our kit bypasses the need for a computer and has the added advantage of being slightly more affordable than existing kits.

Design Process Draft

Our kit is intended to be low-cost, adaptable, and educational. Given below are some of the technical specifications, key features, and the design process we adopted for our robotic kit.

Low-cost: This is important for production. Even if a non-profit uses this design, mass production will be less expensive and use less funds if the cost is lower. Also, a computer is not required.

Adaptability: This kit is meant to be able to be used by multiple skill levels, from beginners to programmers with experience in the Arduino language.

Educational: The kit is accompanied by setup guides for building the robot, wiring it, and programming it. It demonstrates concepts such as for-loops, conditionals, and sensors.

Our kit has undergone several important changes in its design, especially with regard to its programming language.

Version 1

In the original version, we had a 16x2 Liquid Crystal Display (LCD) to show the program. A single-level board accommodated the display, the motors, the Arduino microcontroller, the battery, and an ultrasonic sensor, the latter of which we intended to integrate into the programming language at a later stage. For locomotion, we had two motors in the front and a caster at the back to facilitate turning. The caster consisted of a 3D-printed hemisphere. For our programming language, we had a simple "forwards-backward-left-right" setup. "Forwards" was represented by 'F', "backwards" by 'B', etc. "Left" and "Right" meant to turn in those directions 90 degrees. For example, "FLB" would mean "forwards-left-backwards." This method of representation, while less visually appealing, has the advantage of saving display real estate.

Version 2

In the next iteration, we replaced the LCD display with a 128 by 64 Organic Light Emitting Diode (OLED) display, the reason being that the OLED display had more display real estate. Furthermore, it necessitated fewer pins, reducing the complexity of the wiring for the user. The design also evolved from a single level to a two-level system. The bottom level housed the microcontroller, motors, and caster, and the top would house the battery and OLED display. To connect the two levels, we used "walls" which also served to hold the motors in place. The walls were pre-attached to the two levels, and a tongue-and-groove joint was used to increase structural integrity. We redesigned the caster from a solid hemisphere to a hollow truncated hemisphere that had space for a marble to be used.

Version 3

In the next version, we decided to make the walls detachable and re-attachable so the user could get a bit more experience with the mechanical aspect of robotics. We inserted nuts for the user to twist screws into, thereby connecting walls to levels. To optimize space, the ultrasonic sensor was left out of this design. On the programming side, we integrated for-loops into our language. For example, "F3" would mean "go forward three times." "F3L" would mean "go forward three times, then turn left." Furthermore, we utilized EEPROM to automatically save the programs when the user switches it off — to erase the program, we provided a button for that. We also wrote the setting to go into bang-bang mode using the IR sensor module, which has two IR sensors facing downwards.

Version 4

In this version, we decided to use standoffs in lieu of walls to connect levels. This way, users can get a better look at the insides of the assembled kit. We still used the walls as motor holders but made them shorter to increase visibility. We also replaced IR sensors with photoresistors to demonstrate the concept of if-statement. We used the sensors to gauge if the floor brightness was above a threshold; if it were above, then it would be represented by "1"; else, it would be "0". This way we were able to integrate "if" into our programming language. By replacing the "erase" button with "if," we could have the following syntax

“I0 → [insert]” or “I1 → [insert]”

The “I” represents the “if”; the “0” or “1” represents the state of the floor; and the command in brackets represents the result if the truth value of the floor brightness is equal to the chosen integer. While the arrow takes up display real estate, we decided it was an appropriate trade-off to increase understanding of conditionals. Note that the body or block of the conditionals doesn’t have delimiters such as curly brackets in Java or indents to show scope; therefore, the considered command is the character immediately after the if-statement, as well as any numbers(for-loops) directly after that character. For example, “I0 → F3L” moves forward three times if the floor is dark and then turns left regardless of the state of the floor.

Figure 1 is the final set of components which we have used. These components, when assembled, result in a functional robot with the ability to respond to light conditions, remote control, and display feedback or diagnostics on the OLED screen.

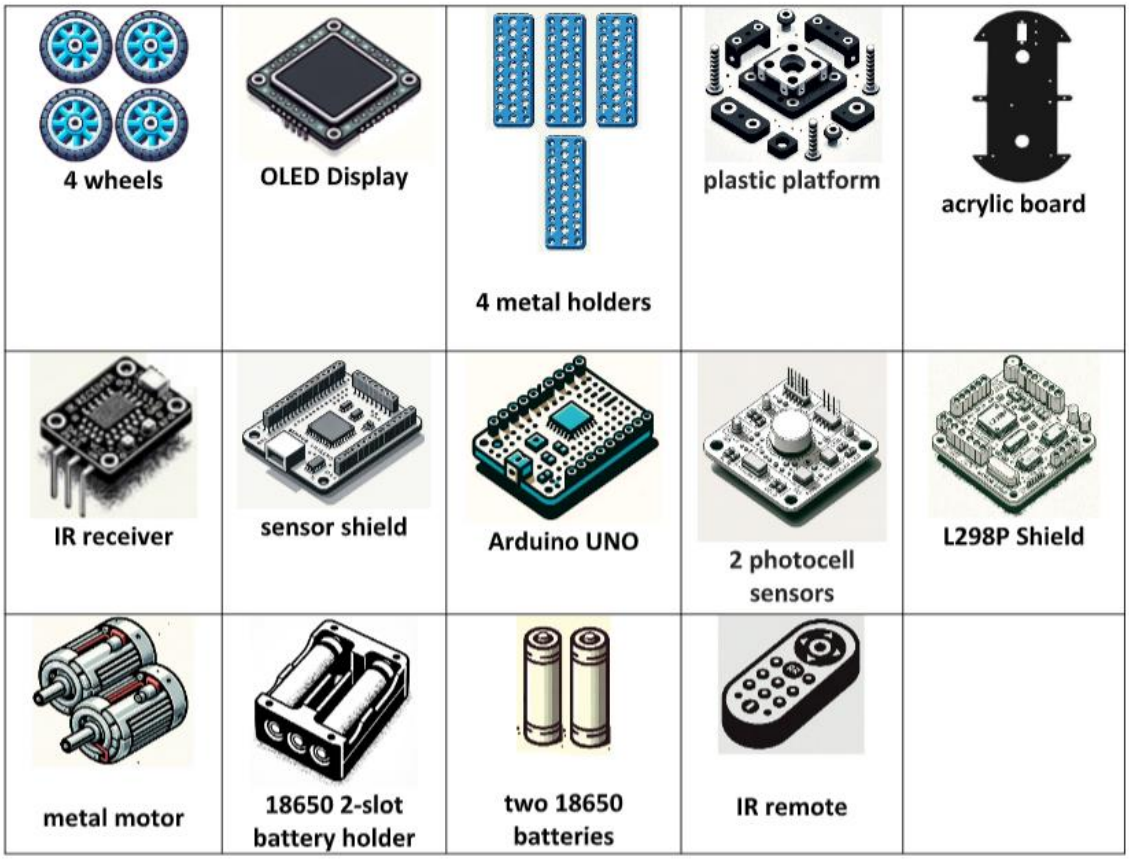

Figure 1: A list of all the components

The OLED display is 128 by 64 pixels in area. The L298P Shield has H-bridges built into it, removing the need for wiring, and purchasing of external motor controllers. The sensor shield provides additional ground and 5V pins for the photoresistors and IR receiver. Below are the details of each component.

Component	Description
4 wheels	Provide mobility to the robot, allowing it to move on surfaces.
OLED display	A screen to display information, feedback, or diagnostics from the robot.
4 metal holders	Used to hold or mount other components securely to the robot's framework.
Plastic platform	The base structure on which all other components are mounted or attached.
Acrylic board	Often used as a protective or structural layer, or to mount additional components.
IR receiver	Receives infrared signals, usually from an IR remote, allowing remote control.
Sensor shield	An interface board that simplifies connecting various sensors to microcontrollers like the Arduino.
Arduino UNO	The brain of the robot. It's a microcontroller board that processes input from sensors and controls the robot's actions.
2 photocell sensors	Detect light levels. They can be used for light-following robots or to detect changes in lighting conditions.
L298P shield	A motor driver shield that allows the Arduino to control motors, determining direction and speed.
Metal motor	Provides the necessary motion to the wheels, propelling the robot.
IR remote:	A handheld device that emits infrared signals to control the robot remotely via the IR receiver.
Two 18650 batteries	Power sources for the robot.

18650 2-slot battery holder:	Holds the 18650 batteries in place, providing a stable power connection.
4 wheels:	Provide mobility to the robot, allowing it to move on surfaces.

Figure 2: A table with descriptions of the functionality for each component

Below is a diagram of the buttons corresponding to commands on the remote.

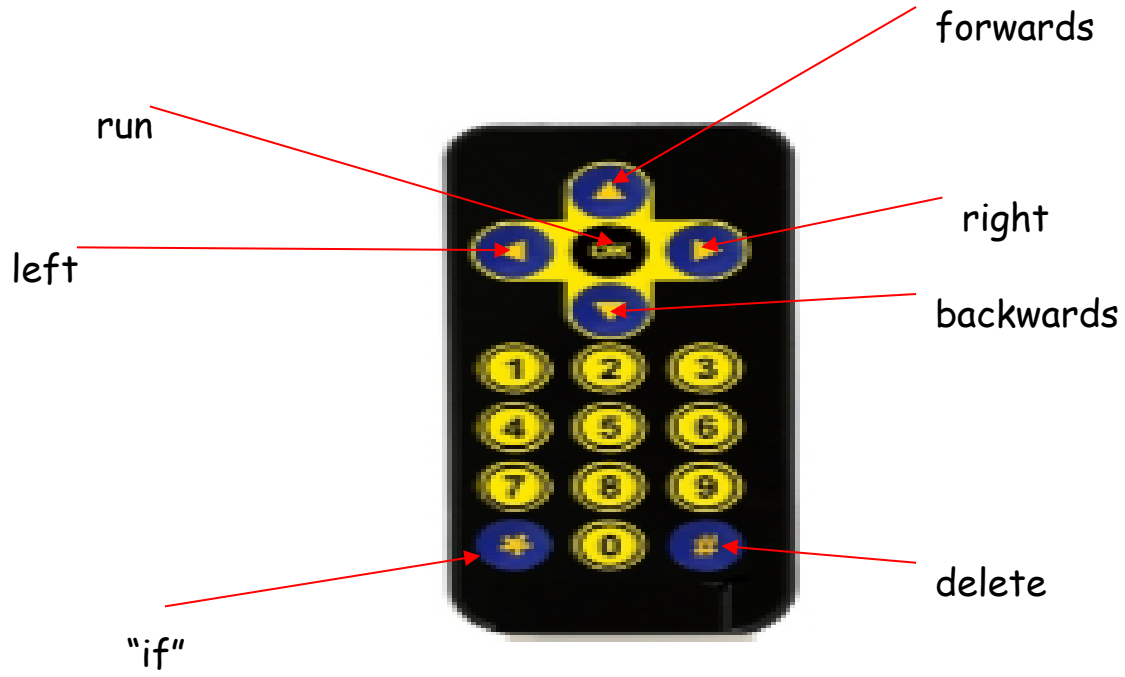

Figure 3: A picture of the remote with the buttons labeled with the commands

Below is the list of commands and their description.

Commands	**Description**
Left	This function directs the robot to rotate right in-place 90 degrees
Right	This function directs the robot to rotate right in-place 90 degrees

Backwards	This function directs the robot to move in the reverse direction from its current orientation.
Forwards	This function directs the robot to move in the forward direction from its current orientation.
If	This function makes the robot perform certain actions based on reading from the photoresistor. This is represented on the OLED display as 'I'. If this is pressed twice in a row, the robot enters bang-bang mode.
Delete	This function allows users to erase a previously entered command or instruction.

Figure 4: A table of the programming commands with their descriptions

We created assembly, wiring, and programming guides for the users, which we have attached in the Appendix as a link to a GitHub repository, which also contains our low-level code.

Future Work

Our kit has been built keeping in mind the low cost, inclusivity, and ease of use for students. Below is our vision to continue to innovate with further refinements.

Limitations: The kit has some limitations. For example, the EEPROM memory used to store the data is limited, and each bit can only be written to or changed at most 100,000 times. This limits the number of changes that can be made to a program; in the future, to combat this, program sequences can be processed by a one-to-one function that returns a unique number for each sequence; this number can be stored in a single bit, increasing effective memory.

Future Work/Next Steps: While our kit is cheaper to produce than the selling prices of most robotics kits and does not require a computer, there is still scope for improvement in the costs. Currently, the main cost arises from the motor shield and, to a lesser extent, the sensor shield.

On the memory storage side, wear-leveling can be incorporated into the EEPROM storage in order to increase the lifetime. Wear-leveling is a method of spreading out the load of storage among the bits. Instead of rewriting new data to a bit, the data is written to another pristine bit. This way, although more bits get written to and worn out, the overall rate of wearing out is slower.

Further programming concepts such as variables and while-loops can be integrated, and analog input from the sensors can be used to incorporate inequalities into the conditionals. These are things that can be improved on in future work.

Conclusion

In conclusion, the development of this low-cost robotics kit signifies a pivotal step toward making robotics and programming education more accessible, especially for students who may not have regular access to computers or laptops. Throughout the progression of this paper, we have outlined the

design and functionality of a comprehensive robotics kit that enables students to grasp the fundamentals of programming and robotics in an interactive and hands-on manner.

The kit's uniqueness lies in its affordability and independence from computer-based programming, breaking down significant barriers to entry in the field of robotics education. By utilizing an IR remote for input and providing an integrated programming guide, the kit facilitates a user-friendly environment for students to learn basic programming structures such as movements, loops, and conditionals. The use of photoresistors and IR sensors introduces the learners to practical applications of sensors, fostering an understanding of their functionality in robotics.

Reflecting upon the detailed comparison with related works in robotics education, it is evident that while there are several innovative approaches in the field, our kit carves out its own niche by emphasizing cost-effectiveness, simplicity, and the ability to function independently of a computer. This not only makes it a viable option for low-income demographics but also promotes inclusivity and equal opportunity in STEM education.

As we consider the future trajectory of this project, it is imperative to address the outlined limitations and explore avenues for further improvement and innovation. From addressing the constraints of EEPROM memory to reducing the kit's cost further, there lies a plethora of opportunities to refine and enhance the kit's functionality and reach. The introduction of advanced programming concepts and the incorporation of analog input from sensors stand out as promising areas for future development.

Acknowledgements

The author would like to thank JaCoya Thompson from Northwestern University for her invaluable guidance and mentorship in the writing of this paper and gathering of resources.

References

1. Horn, M.S. & Jacob, R.J.K. (2007) *Designing Tangible Programming Languages For Classroom Use*, ACM Digital Library, https://dl.acm.org/doi/10.1145/1226969.1227003
2. Hu, F. & Zekelman, A. & Horn, M.S. Judd F. *Strawbies: Explorations in Tangible Programming* (2015), ACM Digital Library, https://dl.acm.org/doi/10.1145/2771839.2771866
3. Saar, L. & Liang, H. & Wang, A. & McDannald A. & Rodriguez, E. & Takeuchi, I. & Kusne A.G. *A Low-Cost Robot Science Kit for Education with Symbolic Regression for Hypothesis Discovery and Validation*, Springer Link, https://link.springer.com/article/10.1557/s43577-022-00430-2#:~:text=03%20November%202022-,The%20LEGOLAS%20Kit%3A%20A%20low-cost%20robot%20science%20kit%20for,for%20hypothesis%20discovery%20and%20validation
4. Horn, M.S. & Banerjee, A. & West, M. & Pinkard, N. & Pratt, A. *TunePad: Engaging Learners at the Intersection of Music and Code*, International Society of the Learning Sciences, https://repository.isls.org//handle/1/6320
5. Vandevelde, C. & Wyffels, F. & Ciocci M. & Vanderborght, B. & Saldien J. *Design and evaluation of a DIY construction system for educational robot kits*, International Journal of Technology and Design Education

6. Addido, J. & Borowczak A.C. & Walwema G.B. *Teaching Newtonian physics with LEGO EV3 robots: An integrated STEM approach*, EURASIA Journal of Mathematics, Science, and Technology Education, https://doi.org/10.29333/ejmste/13232
7. Afari, E. & Khine M.S., *Robotics as an Educational Tool: Impact of LEGO Mindstorms*, International Journal of Information and Education Technology, 10.18178/ijiet.2017.7.6.908

Appendix:

Github Link: https://github.com/roha913/Bridging-the-Educational-Gap-A-Cost-Effective-Computer-Free-Approach-to-Democratizing-Robotics-Le.git