ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #35967

Low-Cost Open-Source Robotics for Education

Mr. Brennan Patrick Miller-Klugman, Wentworth Institute of Technology

Brennan Miller-Klugman is student at Wentworth Institute of Technology. He is currently working towards completing a B.S. in Computer Engineering.

Yali Izzo, Wentworth Institute of Technology

Computer Science student and Robotics/Game Development enthusiast.

Corey Comperchio, Wentworth Institute of Technology

Corey Comperchio is in his final year studying electrical engineering at Wentworth Institute of Technology. His previous co-ops include electrical engineering internships at Stantec in Burlington, MA and R.W. Sullivan in Charlestown, MA. He has worked closely with AutoCAD and Revit, designing and preparing contract drawings that meet the National Electrical Code. He is driven to contribute to society through engineering.

Dr. Marisha Rawlins, Wentworth Institute of Technology

Dr. Marisha Rawlins is an Assistant Professor in the Electrical and Computer Engineering Program at Wentworth Institute of Technology (WIT). Her research interests include computer architecture optimizations, embedded systems and devices used in teaching and healthcare, and methods and systems for improving teaching and learning. Dr. Rawlins received her PhD in Electrical and Computer Engineering from The University of Florida. Prior to working at WIT, she was an Assistant Professor in Computer Engineering, and the Discipline Coordinator for the BASc in Computer Engineering and the MSc in Information and Communication Technology Programmes, at The University of Trinidad and Tobago.

Low-Cost Open-Source Robotics for Education

Brennan Miller-Klugman, *Yali Izzo, Corey Comperchio, Marisha Rawlins
School of Engineering/*School of Computing and Data Science
Wentworth Institute of Technology, Boston, MA, USA
miller-klugmanb@wit.edu, izzoy@wit.edu, comperchioc@wit.edu, rawlinsm@wit.edu

Introduction

In recent years, robotics in the classroom has become a popular form of teaching STEM topics to young students. The modular and freeform nature of robotics allows students to exercise creativity while exploring concepts in STEM fields such as programming and engineering. However, one underlying issue has prevented robotics in the classroom from being more widely adopted - cost. Many of the high-end student-focused robots, like Lego's Mindstorm [1], offer features like customizability but trade a high cost for additional functionality. Other lower-cost options, like Makeblock's mBot [2] do exist but do away with many features to lower the price. The Small Education Robot (WitBot) was designed as an affordable, open-source, and modular alternative to popular educational robotics platforms. Designed for users aged 8 to 10, by using off-the-shelf components including a Raspberry Pi Zero, and several readily available sensors, the WitBot can complete many of the same tasks as much more expensive units. The design goals of the WitBot were to remain cost-effective, user-friendly, and open source. The open-source nature of the WitBot is critical in allowing students to exercise creativity, be it by adding additional components via the 3D printed chassis or coming up with new challenges to solve using the Blockly [3] visual coding interface. A provided image file can easily be burned to a Raspberry Pi compatible micro-SD card to quickly deploy new WitBots. The comprehensive user guide provides build instructions and covers software deployment. A custom lesson plan was constructed to introduce several key programming concepts such as loops and conditionals, aimed at students within the age range of 8 to 10 years old. The low-cost design goal of the WitBot was put in place to help robotics become available to more audiences. Building multiple WitBots at once lowers the price considerably due to the cost savings of buying wires, resistors, and other electrical components in bulk. While building a single WitBot is a viable option, building in bulk allows the WitBot to match the price of even the cheapest competitors while offering advantages such as utilizing a Linux-based Raspberry Pi which allows the WitBot to be controlled through any language that has a Raspberry Pi GPIO library.

Related Work

Given the increased criticality of technology in modern society, the importance of STEM is greater than ever. Using robots in education can help to minimize a lack of interest in STEM [4]. In [4] teachers stated that they believe that students treat robotics-based education as play more than learning and due to this they absorb the information better. [5] demonstrates that robotics in education can lead to a higher level of student engagement.

Products such as Lego's Mindstorm line offer modularity and customization but come at an extremely high cost, whereas products like Makeblock's mBot come in at a more realistic price but trade away some of the customizability options. The Lego Mindstorm kits offer a considerable amount of customizability allowing students to add or remove components either based on pre-defined lessons, or their own creativity. The customizability comes at an extremely high cost of USD359.99. Makeblock's mBot comes in at a much more digestible price of USD89.99 but offers considerably less customizability. The Lego Mindstorm is advertised for users ages 10 and older whereas the mBot lists children 8 and older as their primary audience. Table 1 holds data related to different educational robots along with their included sensors and price.

Table 1. Market Comparison: Current Educational Robotics Platforms with Price and Included Sensors

Product Name	Cost (In USD)	Included Sensors
Makeblock mBot	89.99	ultrasonic sensor, line finder
Lego® Mindstorm	359.99	touch, color, IR
Cubelets Discovery Set	149.00	light, IR
Edison V2	64.90	sound, IR, light, line finder
Root rt0	129.99	touch, bumpers, light, gyroscope, accelerometer

Methodology

Overview

Since the WitBot is aimed primarily at teaching robotics to students aged 8 to 10, it was critical to create lessons that contain a level of difficulty that both captures the student's attention, while also not being so hard to solve that they lead to frustration. Components were chosen to complete tasks such as obstacle avoidance and following a maze. These tasks were chosen as they included core programming concepts like conditionals and loops while also not inhibiting the student's creativity.

Components

Given the plethora of readily available sensors and motors, picking the right parts was a difficult task. For the most part, components were picked based on price and reviews, however, alternatives for any of the specified hardware could be used to offer differences in performance and total cost.

Determining what sensors to use was based on market research and were picked to fit within the original design specifications. An ultrasonic sensor was used to offer students a way to explore obstacle detection and avoidance whereas a line sensor was implemented to allow students to complete tasks like mazes. Potentially, more sensors like additional line sensors, temperature sensors, or even a lidar could be added to make the SERs capabilities more robust.

The WitBot was tested using two different motor drivers, the L298N and the SN754410 but ended up utilizing a L298N in the final design. This choice was made because the L298N has screw-in terminals and a labeled PCB making installation and setup much easier. This driver was paired with two Adafruit TT-Motors which were chosen over a normal DC motor due to the gearbox enabling better performance, while also being easier to mount on the chassis.

Chassis Overview

The chassis design of the WitBot was 3D printed to make it easy for interested parties to modify and add new components to the chassis. The design was created in Fusion360, shown in Figure 1, and was printed on a Prusa Mini.

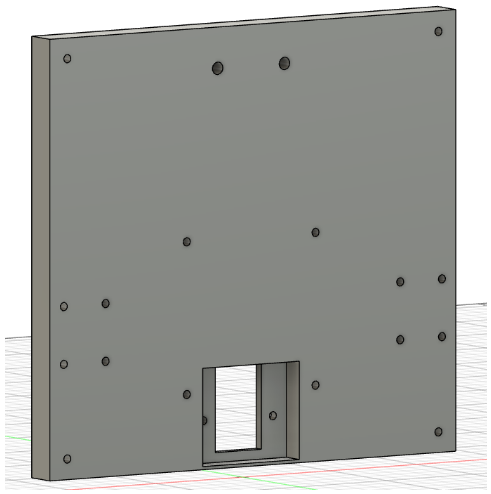

Figure 1. Chassis Design: Fusion360 model for the bottom plate of the WitBot chassis

The WitBot is aimed at being easily assembled by someone who is somewhat technologically literate, and a comprehensive guide on how to put together the robot is available in the documentation. Figure 2 represents the assembly of all the hardware components necessary for the WitBot.

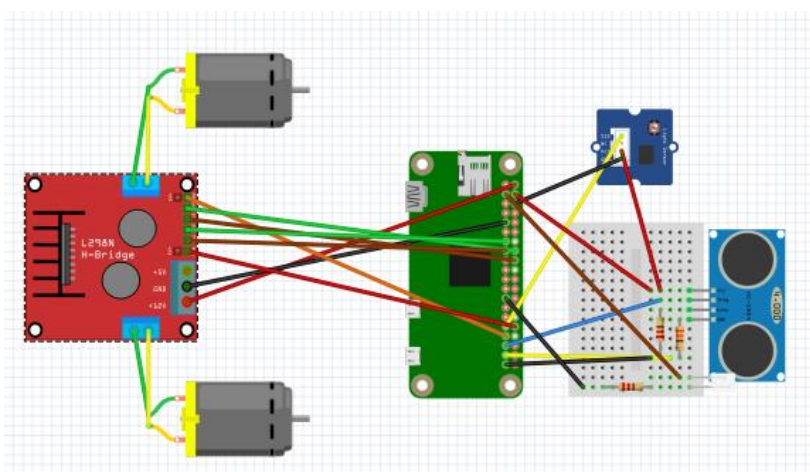

Figure 2. Hardware Diagram: Fritzing diagram of the SERs hardware assembly

Software Design

It can be difficult to learn programming by printing "hello world" on a screen. Due to this, many existing robotics platforms for education integrate a visual coding tool to help young students see how their code interacts with the robot. Using Blockly, a visual coding platform created by Google, the WitBot offers visual programming. Using drag and drop blocks, users can have the WitBot complete tasks like following a maze or stopping before hitting an obstacle. The Blockly panel is available via a web interface created in Flask which can be seen in Figure 3.

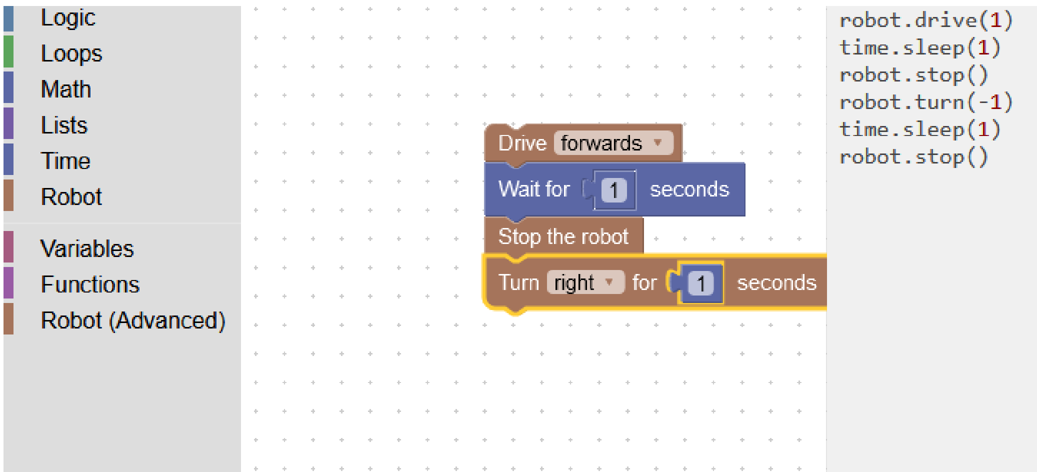

Figure 3. Software Interface: Example of a simple program written in the Blockly interface

Additional Blocks can easily be added to the SERs open-source codebase to accommodate components not supported by default. Alternatively, a more advanced user could SSH into the Raspberry Pi controlling the WitBot to write custom code to control the robot with any language that has an accompanying Raspberry Pi GPIO module.

Although the SERs build guide recommends which General Purpose Input Output (GPIO) pins to use, the assembler can choose to use whichever of the GPIO pins they would like and can adjust which pins were chosen in a config file present in the WitBot repository. A settings page was also created to allow customization like trim settings to ensure that the WitBot is moving straight, as well as a slider to limit the maximum speed of the WitBot.

Results

As shown in Figure 4, the WitBot was successfully assembled and tested.

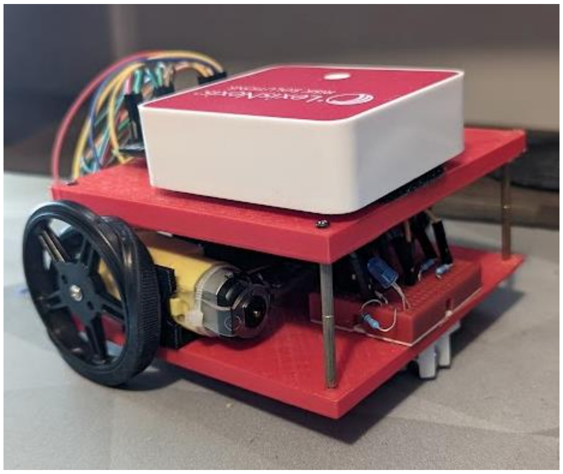

Figure 4. Completed Bot: Image of an assembled WitBot

Utilizing the Blockly interface shown in Figure 3 the WitBot can successfully complete tasks including following a line and stopping before an obstacle. Additionally, several lessons designed for students were written to complete predefined objectives. These lessons along with a brief description are outlined in Table 2.

Table 2. Lesson Outline: Titles and descriptions of lessons

Lesson Name	Lesson Description
Controller	Provide a general overview of the options provided through the SERs web interface.
Blockly Overview	Provide a general explanation of Blockly and use it to blink the onboard LED.
Conditionals	Explain the concept of conditionals and use them to stop the WitBot if the ultrasonic sensor reads that it is too close to an object.
Loops	Explain the concept of loops and use them to repeat actions like driving the bot forward if the ultrasonic sensor does not see an obstacle.
Line Sensor Introduction	Introduce the line sensor and use it to follow a straight line.
Following a Non-Straight Line	Using loops and conditionals, make the WitBot follow a line into a turn.

The WitBot's price decreases greatly depending on how many bots are being constructed since many of the components such as screws and wires come in large packs suited to constructing multiple units. Table 3 shows the price breakdown of the WitBot.

Table 3. Price Breakdown: Parts list and cost breakdown of building 10 WitBots in USD

Name	Quantity	Price
Raspberry Pi Zero W with Headers	10	$140.00
Grove Line Finder	10	$35.10
Ultrasonic Rangefinder	10	$35.6
Battery Bank	10	$100
L298N Motor Driver	10	$24.975
Micro SD Card	10	$42.99
T-T Motor	20	$53.2
Ball Caster	10	$17.60
Wheels	20	$50.0
Jumper Wires	3 Packs	$17.97
Resistors	1 Pack	4.99
Screw Kit	1	14.99
Total Cost		537.415
Cost Of Each Individual Bot		53.7415

Due to the modular nature of the WitBot any of these components could easily be swapped out depending on availability and cost restrictions. A specific model for the battery bank was not listed as it is recommended to use whatever the cheapest and most readily available model is. Additionally, the wheels and ball caster can be 3D printed to cut down on the total cost. Compared to other similar popular products on the market, the WitBot remains very affordable as can be seen in Figure 5.

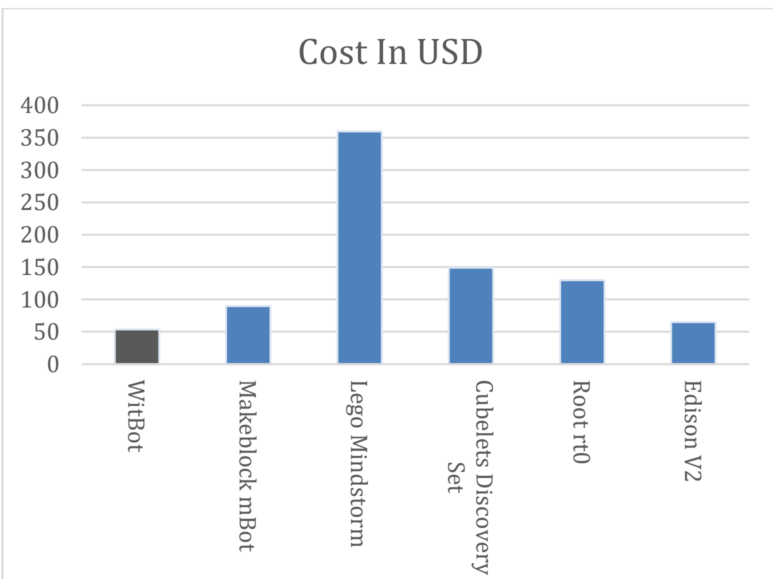

Figure 5. Cost Analysis: Cost of currently available products in comparison to the WitBot (when assembling 10) in USD

The cost of educational robots varies greatly from manufacturer to manufacturer due to variance in componentry, software design, and modularity. Despite this wide range of prices, the WitBot is still able to come in at a low cost compared to many of the main products on the market while still retaining functionality such as including multiple sensors like a line finder and an ultrasonic sensor as well as integrating a visual coding interface.

Conclusion

This paper presented a small, affordable, open-source educational robot built using off-the-shelf parts including readily available motors and sensors. A visual interface created using Blockly was implemented to allow students to experience and apply programming concepts like loop and conditionals in a tactile way. A lesson plan was created to help guide students through the core programming concepts while also leaving room for creativity and exploration. A Raspberry Pi Zero was used for scalability allowing more advanced users to create code in any language sporting a GPIO library. A cost analysis proves that the WitBot can contend with much more expensive robots in terms of features while offering a low price and open-source codebase and design. In the future, additional software and features could be added to improve the SERs capabilities. Allowing multiple SERs to communicate, and complete common tasks could be implemented. Adding more sensors, such as a light sensor or microphone on the WitBot could allow for additional capabilities like stopping when hearing a sound.

References

[1] “Mindstorms Robot Inventor”..Lego. https://www.lego.com/en-us/product/robot-inventor-51515 (accessed January 14, 2022).

[2] “Makeblock mBot STEM Education Coding Robot Kit for Scratch, Arduino C”. Makeblock. https://store.makeblock.com/products/diy-coding-robot-kits-mbot (accessed January 14, 2022).

[3] “Blockly”. Google. https://developers.google.com/blockly (accessed January 14, 2022).

[4] A. Khanlari, "Effects of educational robots on learning STEM and on students' attitude toward STEM," in Dec 2013, Available: https://ieeexplore.ieee.org/document/6908304. DOI: 10.1109/ICEED.2013.6908304.

[5] V. Chaudhary, P. Sureka and A. Sureka, "An Experience Report on Teaching Programming and Computational Thinking to Elementary Level Children using Lego Robotics Education Kit," IEEE 8th International Conference on Technology for Education, 2016. DOI: 10.1109/T4E.2016.15.

[6] "Cubelets Discovery Set”. Cubelets. https://modrobotics.com/cubelets-discovery-set/ (accessed January 14, 2022).

[7] “Edison Programmable Robot - Ideal for school classroom education”. Edison. https://meetedison.com/ (accessed January 14, 2022).

[8] "Introducing the Root robots for coding, discovery, and play". iRobot. https://www.irobot.com/Root (accessed January 14, 2022).