Arduino-based Educational Robotics with Android Mobile Application

Surasak Artyatha* and Wansuree Massagram

Department of Computer Science and Information Technology, Faculty of Science,
Naresuan University, Phitsanulok, Thailand

Abstract

The purpose of this project is to develop an effective tool based on mobile computing and tangible technology to assist children's computational thinking skills. By employing physical objects---in this case, A robot car, people, particularly school students, are encouraged to learn computer science concepts without being intimidated by text-based complicated syntax. For mobile computing, Android OS was selected over iOS for wider accessibility. The application contains various puzzles with a set of solvable commands in blocks. The commands from the mobile application are sent to the Arduino-controlled robot via Bluetooth connection. The combination of mobile-application and tangible robot cars could increase the children's interest and inspire them to learn more about computer science and computational thinking

Keywords: Arducation Bot; Education; Android; Arduino; Coding; Computer Science; Computational Thinking

INTRODUCTION

The first step in the global efforts to prepare children in today's fast changing world is to equip them with knowledge in computer science. Ministries of education worldwide have added computer programming and computation thinking in their curriculums. Computational thinking is a systematic problem solving, using computer science concepts like abstraction and decomposition. Such skills are essential for the digital age in today's modern world (Sye, Y. L., Joyce, H. L. K. ,2014, pp.51-61). However, most schools still do not have access to enough personal computers in order for schools to be able to offer coding as a class.

Phetsrikarn (Phetsrikran, T., Massagram, W., & Harfield, A.,2017) proposed a combination of mobile iOS application and robot car to promote computational thinking skills in children, and demonstrated its feasibility in a cross-cultural setting in (Phetsrikran, T., Massagram, W., Phoka, T., & Harfield, 2018 A.,pp.1-6). This work extends Phetsrikarn's initiatives by creating a mobile application for Android OS. The robot in this work was also modified to improve its performance with a

* ผู้ประสานงาน (Corresponding Author)
e-mail: surasaka59@email.nu.ac.th

proportional–integral–derivative (PID) controller to ensure a more robust locomotion of the robot. Users could still enjoy solving the puzzles by writing short snippets of code to achieve pre-specified goals using a visual programming language similar to Blocky and Logo. Such drag-and-drop interface has been known to be more user-friendly than text-based programming.

The organization of this paper is as follows: Section 2 presents technical background and related research, Section 3 illustrates the design and implementation, Section 4 shows the results of the study, and Section 5 presents this study's conclusion.

TECHNICAL BACKGROUND

The robot in this study was inspired by two main robots: Arducation Bot and the Arduino Line Follower Robot.

Arducation bot (Phetsrikran, T., Massagram, W., & Harfield, A.,2017), (Phetsrikran, T., Massagram, W., Phoka, T., & Harfield, 2018 A.,pp.1-6). [2], [3] is a collaboration between educational robotics and mobile application. It was designed to be a teaching platform to introduce computational thinking concepts to children. But instead of building an iOS application with Swift for the mobile tablet, Android Kotlin is used.

Arduino Line Follower Robot (Anonymous, 2019) [4] is the robot that moves along the black line. It is one of the first robots for students and beginners who want to have a hands-on experience with robotics. The principle of line following is straightforward. By using two IR sensors, which consist of IR LEDs and photodiodes, placing in a reflective way i.e. side – by – side, whenever they come into proximity of a reflective surface, the light emitted by the IR LEDs will be detected by the photodiode. This technique is commonly used for the movement of the robot.

The robot and the application in this project were aimed to teach computational thinking. This is a required skill for a problem-solving process that includes some characteristics and dispositions. The goal of the project is that children can acquire, through solving the puzzles, these following computational thinking skills: decomposition, pattern recognition, abstraction, and algorithm design.

DESIGN AND IMPLEMENTATION

This section presents the design and implementation of both hardware and software development.

A. Design

i) System Architecture:

The system architecture consist of two parts: hardware and software. As illustrated in fig. 1, the system would start when both devices are connected via Bluetooth.

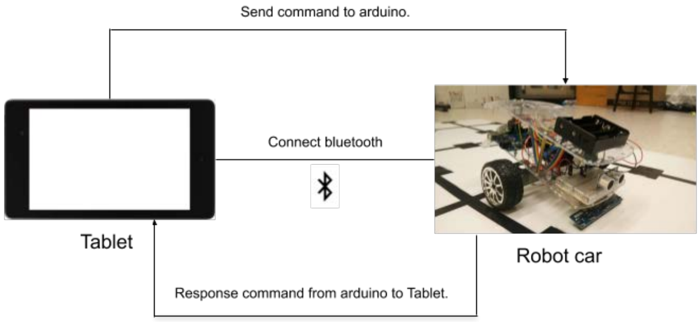


Fig. 1. System architecture

ii) System Analysis:

The main work in this project was to build a reliable communication between Android application and Arduino. The system flow was designed as shown in fig. 2. The flowchart diagram explains how the Android app and the Arduino robot car interacts. The use case diagram explaining how a user interacts with the system is shown in fig. 3.

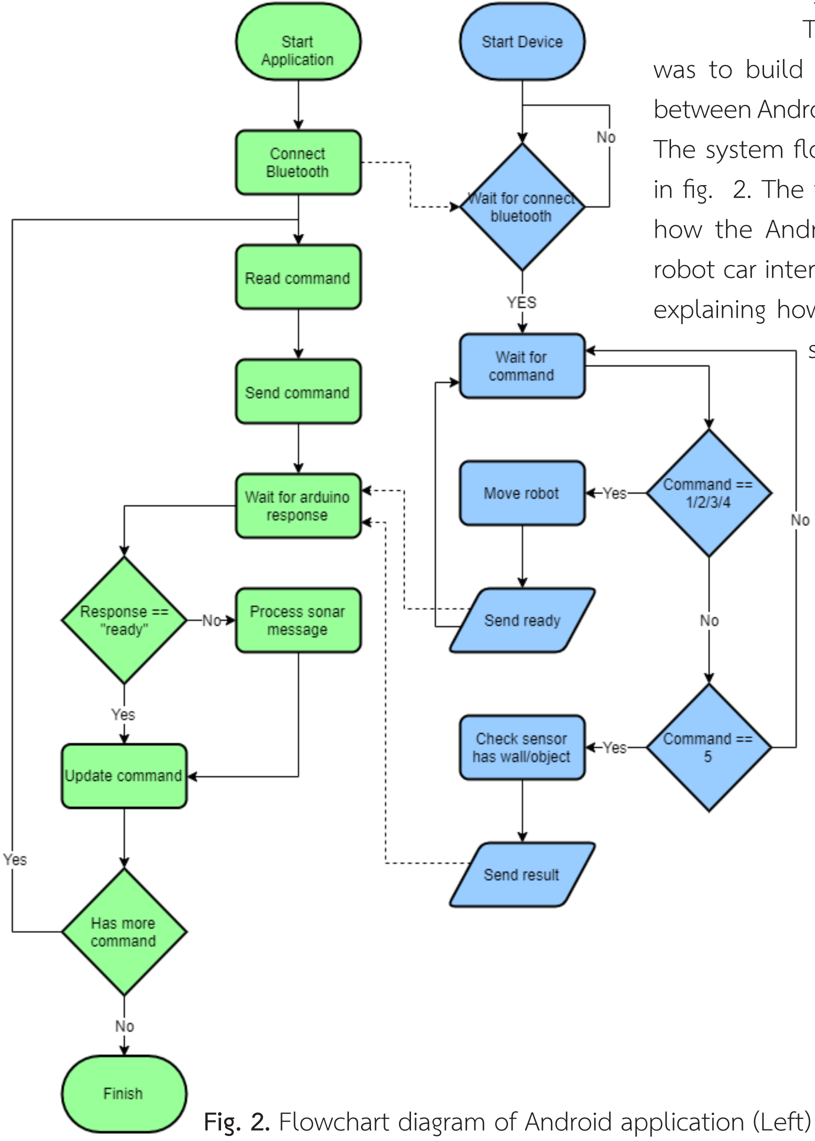


Fig. 2. Flowchart diagram of Android application (Left) and Arduino Process (Right)

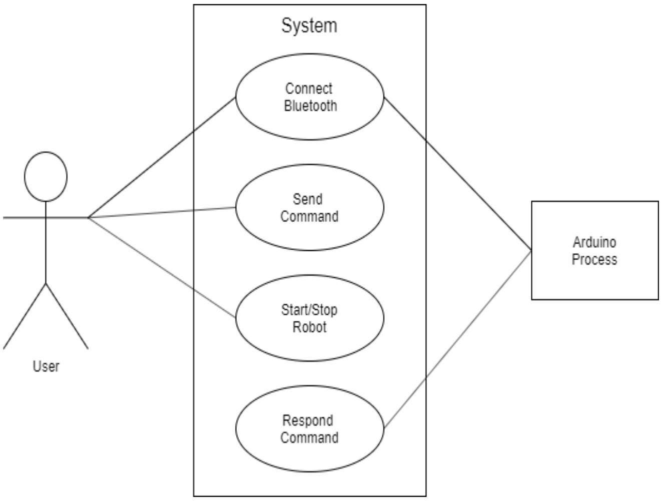


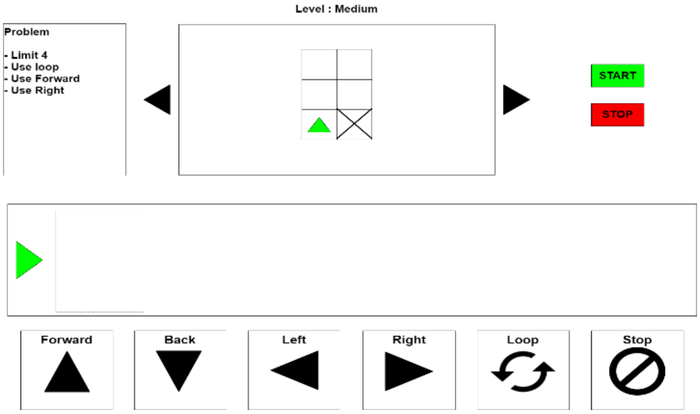
Fig. 3. Use case diagram

B. Implementation

i) InitialDesign:

The project was planned for both hardware and software. Since the project was aimed at children, UI/UX played an important role in the design. For hardware, a ladybug was the inspiration for the robot car. Hence, biomimicry was followed. The major hardware components are the microcontroller, the input sensors, and the output drive system. If the microcontroller is the brain, the sonar sensor would be the insect's eyes, the infrared line sensors would be the antenna, and the wheels would be its legs.

For software, its main functionality is to communicate with the robot car via Bluetooth connection. Fig. 4 shows the initial design of the puzzle page. The user must drag the command into the commands box to make the robot move. The entered commands are passed through the Arduino board to allow the robot to move according to the instructions.



Level : Medium

Problem

- Limit 4
- Use loop
- Use Forward
- Use Right

Fig. 4. The initial puzzle design for the mobileapplication

ii) Final Design:

The project has gone through many iterations of work. The final iteration is described as follows.

To improve the control feedback of the robot car, A proportional–integral–derivative (PID) controller (O'Dwyer, A., 2009) [5] was selected for this work. PID calculates the errors derived from the process variables and the desired value. In this project, the values from the IR sensor were used as input to find the error values to adjust the suitable motor speed while moving forward. PID has three values which are Kp, Ki, and Kd. Kp is proportional to the errors. If Kp value is set too high, the robot will vibrate and unstable. However, if it is too low, the robot will be slow to respond. Ki is the integral sum of errors over time. When combined with Kp, it will reach the desired point and reduce the remaining errors. However, the Ki is a response from the previous cumulative error which may result in diverting the desired point to another direction. The last variable, Kd, is the rate of change of errors. Kd is a slow rate of change to reach the desired level. It is used to reduce the overshoot caused by Ki values and improve stability of control. Nonetheless, Kd will be amplified very quickly to disturb the Kp value and can make the control unstable. The PID are calculated from the following equations and their corresponding algorithm implemented is shown in fig. 5.

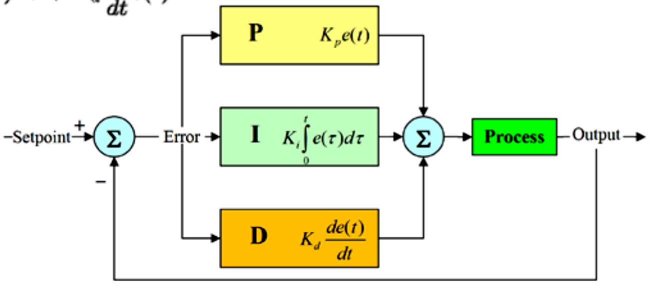
$$P_{\text{out}} = K_p\, e(t)$$

$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau$$

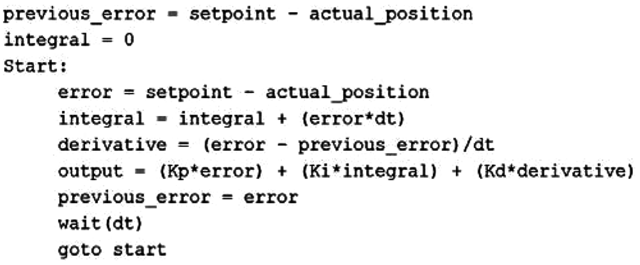
$$D_{\text{out}} = K_d \frac{d}{dt} e(t)$$

$$\mathrm{MV(t)} = P_{\text{out}} + I_{\text{out}} + D_{\text{out}}$$

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)$$



(a).



```
previous_error = setpoint - actual_position
integral = 0
Start:
     error = setpoint - actual_position
     integral = integral + (error*dt)
     derivative = (error - previous_error)/dt
     output = (Kp*error) + (Ki*integral) + (Kd*derivative)
     previous_error = error
     wait(dt)
     goto start
```

Fig. 5. PID controller. (a) feedback control block diagram and (b) algorithm

Once the control algorithm was finalized, the final hardware was specified and sourced. The robot car's hardware includes an Arduino Mega R3, an HC-05 Bluetooth module, a 5-way infrared line tracking, an HC-SR04 ultrasonic sensor, a motor encoder, an L298N motor driver, and two motors. The components were assembled as shown in fig. 6. The circuit diagram of the design, as illustrated in fig. 7, shows the red and black wires as the power source for all components. The other elements were connected accordingly and installed on the plastic chassis with wheels.

A simple Android program was written to test the Bluetooth communication between the Arduino and the Android application. Once the Bluetooth communication was verified to work properly, more functions were added to the Android application.

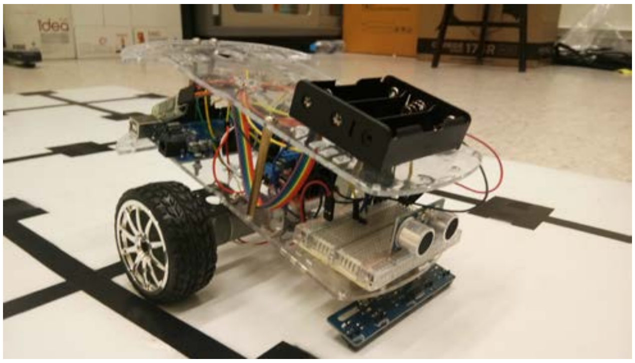


Fig. 6. The robot car

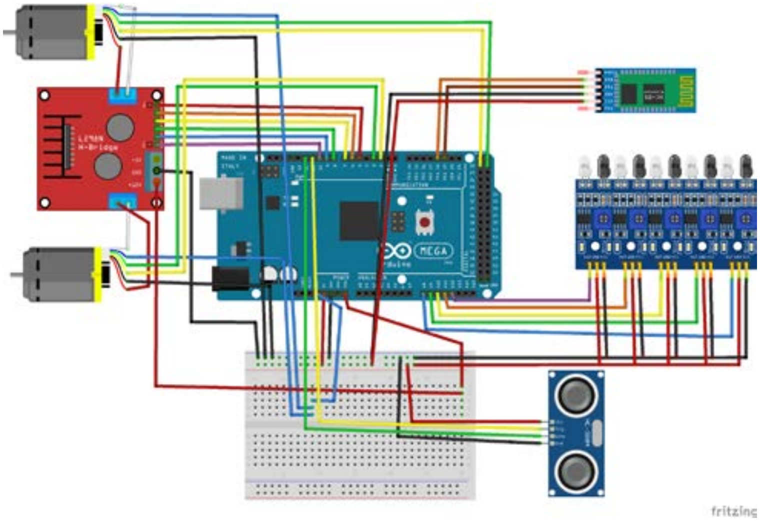


Fig. 7. The robot car's circuit diagram.

The main function of the Android application is to receive the user's commands and send them to control the robot car while keeping track of the user's progress and engaging the users.

To keep young children engaged, colorful and approachable graphics were used. Block programming is intuitive for kids. The following description contains the step-by-step instructions of the program workflow.

When the user enters the application, the first page displays the homepage of the Arducation Bot application as shown in fig. 8.

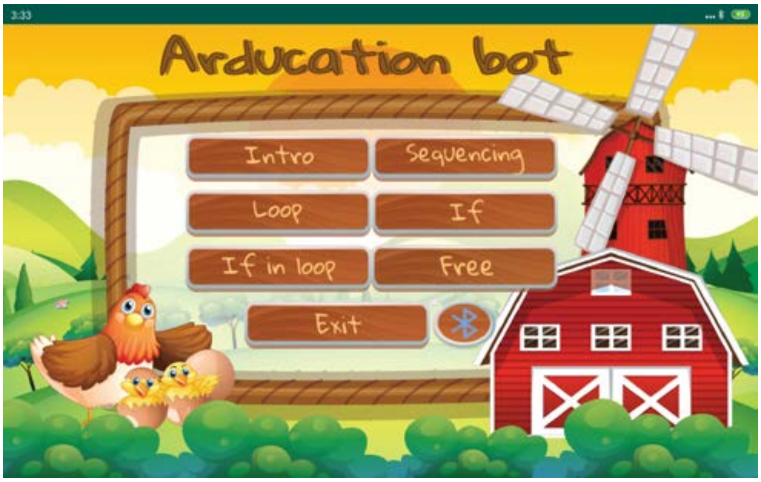
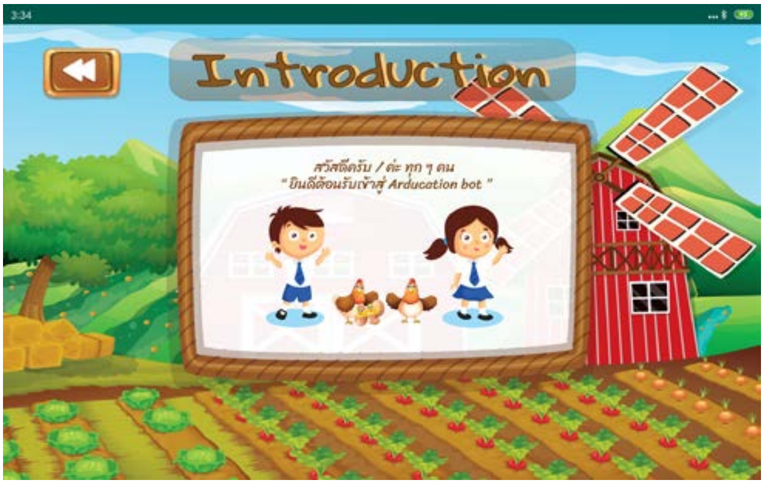


Fig. 8. The application homepage

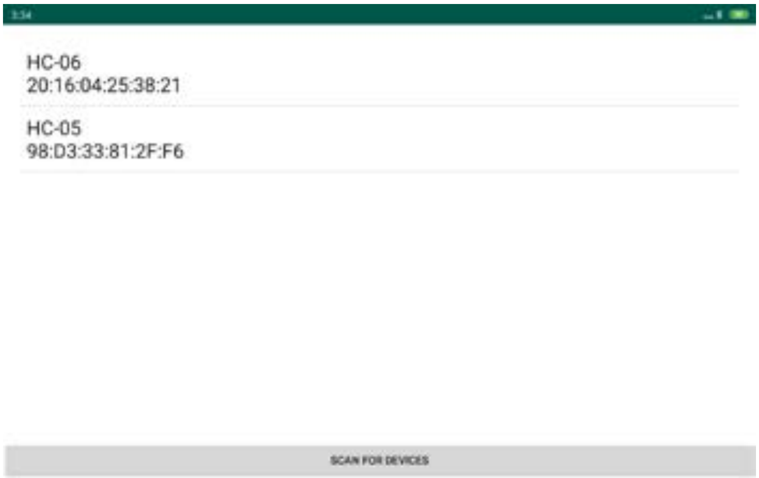
If the user selects the "Intro" button, the application takes the user to fig. 9. The introduction page contains basic instructions for the application.



สวัสดีครับ / ค่ะ ทุก ๆ คน
" ยินดีต้อนรับเข้าสู่ Arducation bot "

Fig. 9. The introduction page

Before using this application, the Bluetooth connection must be properly established. The Bluetooth symbol can be seen on the home page in fig. 8. The user can select it to pair and connect with the available Bluetooth, as shown in fig. 10. For an Android OS tablet, this should allow the robot car to pair its Bluetooth with the tablet's before proceeding with the application. The user can select the Bluetooth that has already been paired to connect Android and Arduino.



3:34

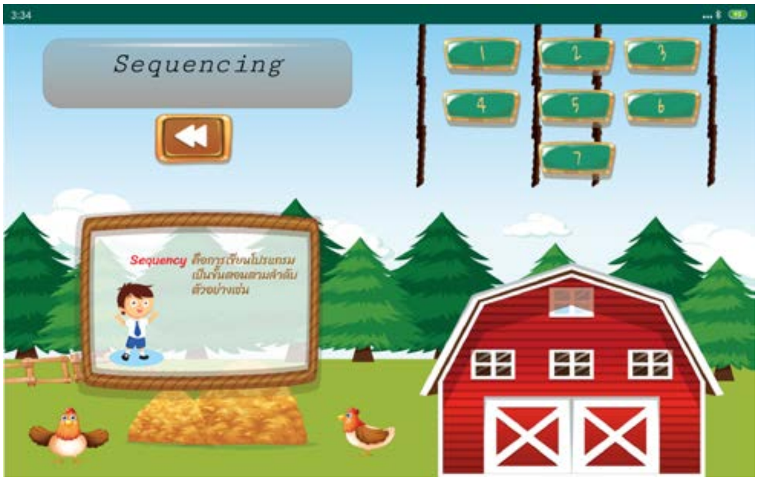
HC-06
20:16:04:25:38:21

HC-05
98:D3:33:81:2F:F6

SCAN FOR DEVICES

Fig. 10. Bluetooth connection menu

The user can now choose to solve the problem starting from Sequencing, Loop, If, If in Loop, and Free (as in freestyle). Fig. 11 shows the screen capture from Sequencing. There are seven puzzles to teach the basic algorithm to solve the problems sequentially. Each topic also contains a simple and short description in Thai for the user to understand.



Sequency คือการเขียนโปรแกรม เป็นขั้นตอนตามลำดับ ตัวอย่างเช่น

Fig. 11. Sequencing menu

The image in the middle of fig. 12 contains an example problem for sequencing. The workspace, seen in the same figure, waits for the commands to be dragged and dropped into the field. The user can solve the puzzle on this page by dragging and dropping the commands she thinks appropriate.

Below the workspace are various commands the user can select from. The number of commands required to solve each problem is shown in the workspace. For example, the particular puzzle in fig. 12 requires two commands to move the robot forward two blocks. The right of the page are menu buttons. The Run button will send all commands in the workspace to the robot car to move according to the command. The robot will operate one by one and stop waiting for the next command. The Stop button will stop the robot from working immediately. The Trash button is a button for clearing all commands in the workspace.

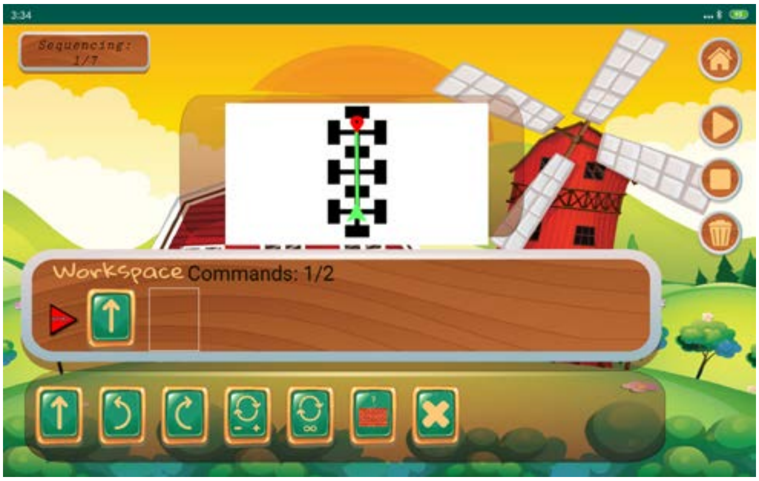


Fig. 12. Workspace in a puzzle page.

TESTING AND RESULTS

A. Hardware Testing

The sensors were first tested to ensure that the output data of the sensors worked as expected. A motor encoder was initially selected as the input feedback to control the movement of the robot car. Using the value of encoder motor to control the movement is difficult because the rotation of each motor must be measured for the calculation. It was difficult for both motors to spin and stop at the same time. The IR sensor was chosen instead a PID controller. The values from the IR sensor were used to calculate the errors straightforwardly. The values of K_p, K_i, and K_d could be adjusted to suit the robot car.

This project uses the Arduino IDE as an environment to write the programs for the robot car in C-like Arduino language. There was a problem with the Bluetooth communication because initially, an HM-10 Bluetooth module was used. However, the HM-10 was designed for iOS and thus had to be replaced with the HC-05 module. The communication was tested by sending data from an Android device to show in Serial Monitor on Arduino IDE and receiving the data back to show that the prototyping communication worked.

B. Software Testing

The iOS application and the robot car [2] and [3] that this work was inspired by and improved upon was tested on more than one hundred students. For the initial phase of this work, the Android version and the improved robot car with the PID controller were tested with a much smaller target group of 13 students. They solved all the puzzles and completed the survey. Although the number of participants was small and statistically insignificant, the summary of the questionnaire and the collected feedback are shown in Table 1.

Table 1. Summary of the questionnaire (5 being the most significant)

Questions	$\bar{X}$
Software	
1. How satisfied are you with the speed of the application?	4.15
2. How satisfied are you with the overall menu?	4.31
3. Do you think it’s easy to use apps with the robot car?	4.15
4. Do you like the application design?	4.15
Hardware	
1. How satisfied are you with this robot car ease of use?	3.92
2. How satisfied are you with the movement of the robot car?	3.92
3. Do you like the robot car’s design?	4.08
Overall	
How much fun do you have with the app and robot cars?	4.38
Feedback	
Add more diverse puzzles in the application.	

Most users were satisfied with the overall performance of the Android application and the robot car as a pedagogical tool to teach computational thinking. However, the feedback shows that despite improving with the PID controller, the locomotion of the robot car could still be enhanced, perhaps with more sensory data.

Currently, there are only twenty puzzles in the application to keep the play short. However, for the students to receive more benefits, more practice problems should be added along with instructional video clips.

CONCLUSION

The research objectives of designing and implementing a robot car to be controlled with an Android application on a tablet was met. PID controller was implemented to improve the performance of the robot car locomotion. The mobile application, inspired by the previous iOS version, was written in Android for more universal use. The users were satisfied with the outcomes of the project. The future work includes using more sensory data for PID feedback control loop, more puzzles, and instructional videoclips

ACKNOWLEDGMENT

The authors would like to thank the Naresuan University Maker Club for the sponsored equipment, Dr. Antony Harfield for his advice on debugging and developing the mobile application, and finally, Assistant Professor Dr. Thanathorn Phoka, Mr. Titipan Phetchsrichan, and Mr. Kritsana Kumphet for their technical and moral support.

REFERENCE

Sye, Y. L., Joyce, H. L. K. (2014, September). Review on teaching and learning of computational thinking through programming: What is next for K-12?, 2014 Computers in Human Behavior (pp. 51- 61).

Phetsrikran, T., Massagram, W., & Harfield, A. (2017). First steps in teaching computational thinking through mobile technology and robotics. Asian International Journal of Social Sciences, 17(3), 37-52. https://doi.org/10.29139/aijss.20170303

Phetsrikran, T., Massagram, W., Phoka, T., & Harfield, A. (2018, November). A Feasibility Study of Arducation Bot: An Educational Robotics and Mobile Application Kit for Computational Thinking Skills. In *2018 22nd International Computer Science and Engineering Conference (ICSEC)* (pp. 1-6). IEEE.

How To Make Arduino Line Follower Robot?, October 10, 2019 Retrived from https://www.electronicshub.org/arduino-line-follower-robot/

O'Dwyer, A. (2009). *Handbook of PI and PID controller tuning rules.* Imperial college press.