

การพัฒนาการเขียนโปรแกรมควบคุมหุ่นยนต์สำหรับนักเรียนระดับประถมศึกษา

Development of ROS programming controlling robots for primary school students

เที่ยง เหมียตโรสง^{1*} และ ศิวัพร เหมียตโรสง²

¹ สาขาวิชาเทคโนโลยีไฟฟ้า
คณะวิศวกรรมศาสตร์และเทคโนโลยี
อุตสาหกรรม มหาวิทยาลัยราชภัฏ
เพชรบุรี
² สาขาวิชาวิทยาการคอมพิวเตอร์
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยราชภัฏเพชรบุรี
* ผู้รับผิดชอบบทความ
thiangoneone@gmail.com

Received: 15 Aug 2024
Revised: 9 Jan 2025
Accepted: 10 Mar 2025

บทคัดย่อ

ปัญหาการเรียนรู้ศาสตร์ขั้นสูงสำหรับเด็กคือองค์ความรู้เป็นนามธรรมสูง เพราะเด็กเรียนรู้ได้ดีคือสิ่งที่ป็นรูปธรรม บทความนี้นำเสนอการพัฒนาการเขียนโปรแกรมควบคุมหุ่นยนต์สำหรับนักเรียนระดับประถมศึกษา ที่เด็กสามารถเขียนโปรแกรมโดยไม่ต้องเขียนในคอมพิวเตอร์ แต่เขียนบนโต๊ะ โดยระบบประกอบด้วย 2 ส่วน คือ 1) โปรแกรมรอสที่สัมผัสและรู้สึกได้ และ 2) หุ่นยนต์เคลื่อนที่ โปรแกรมที่พัฒนาขึ้นทำให้เด็กได้แนวคิดและเข้าใจเกี่ยวกับโปรแกรมรอสและแนวคิดการเขียนโปรแกรมเชิงวัตถุเบื้องต้น จากการประเมินพบว่า เด็กสามารถควบคุมหุ่นยนต์ด้วยโปรแกรมรอส โปรแกรมที่สัมผัสและรู้สึกได้ ในการควบคุมเต้าให้เคลื่อนที่ด้วยคีย์บอร์ดคอมพิวเตอร์และแสดงการเคลื่อนที่ของเต้าบนจอคอมพิวเตอร์ได้ นอกจากนั้นเด็กยังเข้าใจคำสั่ง roscore เป็นคำสั่งเชื่อมต่อการสื่อสารระหว่างสองคำสั่งที่ควบคุมหุ่นยนต์ roscore เป็นตัวหลัก (master) หรือหัวหน้า ส่วนอีก 2 คำสั่งเป็นลูกน้อง (node) ทำให้เด็กเข้าใจแนวคิดของระบบการโปรแกรมหุ่นยนต์ด้วยโปรแกรมรอส ซึ่งเด็กยังไม่เคยเรียนรู้และใช้งานมาก่อน

คำสำคัญ: การเขียนโปรแกรม ระบบปฏิบัติการรอส หุ่นยนต์เคลื่อนที่ นักเรียนระดับประถมศึกษา

Abstract

The problem with learning advanced science for children is that the knowledge is highly abstract. Because children learn well when it comes to concrete things. This article presents the development of ROS robot programming for elementary school students. That children can write programs without having to write them on a computer but written on the table. The system consists of 2 parts 1) The ROS tangible programming and 2) mobile robots. The developed program gives children an idea and understanding of ROS programming and basic object-oriented programming concepts. From the evaluation it was found that Children can control robots with the ROS tangible programming to control the turtle's movement with a computer keyboard and display the turtle's movement on the computer screen. In addition, children also understand that the roscore command is a communication link between two commands that control the robot. Roscore is the main character (master) or leader, and the other two commands are subordinates (nodes). This makes children understand the concept of programming systems robots with the ROS programming which children have never learned and used before.

Keywords: Programming, Robot Operating System, Mobile robot, Elementary school students

1. บทนำ

การแนะนำเด็กให้รู้จักกับเทคโนโลยีขั้นสูงในช่วงหลายทศวรรษที่ผ่านมา โปรแกรมภาษาโลโก้ (logo) ถือว่าเป็นการเริ่มต้นเปิดโลกแห่งเทคโนโลยีให้กับเด็ก ต่อมาได้แนะนำ Programmable Brick ให้เด็กได้เรียนรู้เกี่ยวกับการเขียนโค้ดและฮาร์ดแวร์ เทคโนโลยีนี้จะทำให้เด็กได้เรียนรู้สิ่งใหม่ ๆ นอกจากนั้นระบบไดนามิกเป็นอีกระบบหนึ่งที่ช่วยให้เด็กได้รู้จักเรียนรู้และรับรู้ถึงพฤติกรรมของสิ่งต่าง ๆ AI เป็นศาสตร์ขั้นสูงอีกศาสตร์หนึ่งที่แนะนำให้เด็กเรียนรู้ที่จะปรับตัวและใช้งานอย่างชาญฉลาดในศาสตร์แห่งวิศวกรรมที่เป็นอัจฉริยะในยุคปัจจุบัน อินเทอร์เน็ตของสรรพสิ่ง (IoT) เป็นศาสตร์สำหรับเด็กในการเรียนรู้เพื่อเตรียมเด็กในอนาคตให้เข้าถึงโลกที่เชื่อมต่อของอุปกรณ์ผ่านอินเทอร์เน็ต และที่สำคัญศาสตร์ The Robotics Operating System (ROS) เป็นเทคโนโลยีที่เด็กจำเป็นต้องเรียนรู้เพื่อพัฒนาหุ่นยนต์ให้ทำงานร่วมกับมนุษย์ได้ดียิ่งขึ้น ช่วยให้ทุกคนทั่วโลกสามารถทำงานและมีกิจกรรมร่วมกันได้ในการพัฒนาหุ่นยนต์

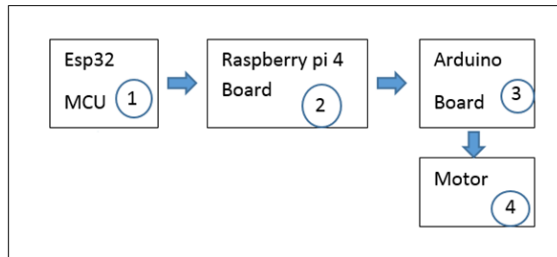
ปัญหาการสอนให้เด็กเรียนรู้ในระดับประถมศึกษาคือการเรียนรู้สิ่งที่เป็นนามธรรม ด้วยเหตุที่เด็กวัยนี้จะเรียนรู้ได้ดีในสิ่งที่เป็นรูปธรรม การเรียนรู้ศาสตร์ขั้นสูงส่วนมากจะเป็นองค์ความรู้ที่เป็นนามธรรมขั้นสูง ทำให้เกิดความยากขึ้นไปอีกหลายเท่าตัวสำหรับเด็กในวัยนี้ ดังนั้นที่ผ่านมาการเรียนรู้ด้านเทคโนโลยีและสิ่งที่เป็นนามธรรมสูงได้มีการพัฒนาการเรียนรู้มาตลอดในปี 1980 Papert ได้ค้นพบรูปแบบการเรียนรู้ทางคณิตศาสตร์ขณะเล่นกับเกียร์ (Gears) โดยค้นพบเกี่ยวกับการเรียนรู้ว่าสิ่งต่าง ๆ จะเป็นเรื่องง่ายหากสามารถเรียนรู้และซึมซับมันในรูปแบบของตนเอง เกียร์เป็นวัตถุแห่งการคิดสำหรับการพัฒนาการคิดเชิงคณิตศาสตร์และความเข้าใจสำหรับตนเอง Papert ได้แนะนำการเขียนโปรแกรมให้เด็กได้เรียนรู้สภาพแวดล้อมของโลก [1] ซึ่งถือเป็นจุดเริ่มต้นในการเปิดโลกคอมพิวเตอร์ให้กับเด็ก Randy et al ได้จัดทำเทคโนโลยีใหม่เรียกว่า Programmable Brick ที่ทำให้เกิดกิจกรรมต่าง ๆ เพื่อเข้าใจเทคโนโลยี และสำรวจว่าเทคโนโลยีใหม่นี้อาจเปิดโอกาสการเรียนรู้ใหม่ๆ ให้กับเด็กได้อย่างไร [2] Zukerman เสนอ System Blocks แสดงพฤติกรรมของระบบ (โดยใช้แสง เสียง และกราฟ) เพื่อสนับสนุนรูปแบบการเรียนรู้ที่หลากหลาย และการสำรวจกระบวนการแบบไดนามิกที่มี

ความสนุกสนาน โดยแนะนำการใช้ System Blocks เป็นแพลตฟอร์มการสร้างแบบจำลองและการจำลองที่สามารถให้โอกาสเด็กนักเรียนในการเผชิญหน้ากับความเข้าใจผิดเกี่ยวกับพฤติกรรมแบบไดนามิก [3] Williams ออกแบบ PopBots ชุดเครื่องมือที่พัฒนาขึ้นสำหรับเด็กอายุ 4-6 ปีเพื่อสำรวจและเรียนรู้เกี่ยวกับ AI ซึ่งพบว่าเด็กเล็กสามารถเข้าใจแนวคิด AI ได้ [4] งานวิจัยของ Strawhacker เกี่ยวกับ CRISPEE ซึ่งเป็นเทคโนโลยีที่สัมผัสและรู้สึกได้แบบใหม่เพื่อแนะนำแนวคิดพื้นฐานของวิศวกรรมชีวภาพให้กับเด็กด้าน DNA ที่มีส่วนผสมร่วมกันระหว่างแนวคิดด้านวิศวกรรมศาสตร์และวิทยาการคอมพิวเตอร์ [5] ศิวพรและเที่ยง พัฒนาสมาร์ตฟาร์มโดยใช้อินเทอร์เน็ตของสรรพสิ่งในการเกษตรโดยการเขียนโปรแกรมที่สัมผัสและรู้สึกได้สำหรับเด็กพบว่าเด็กสามารถเข้าใจแนวคิดฟาร์มอัจฉริยะโดยใช้อินเทอร์เน็ตของสรรพสิ่งในการเกษตรและอัลกอริทึมของการเขียนโปรแกรม [6]

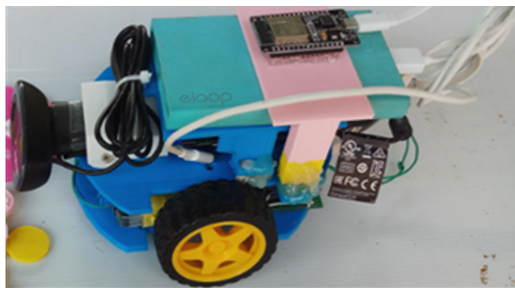
สำหรับงานด้านการโปรแกรม ROS ได้พัฒนาโปรแกรมสามแพลตฟอร์ม ได้แก่ เท็กซ์โหมด (text mode) เป็นการโปรแกรมด้วยข้อความ [7] ไอคอนิกโหมด (iconic mode) เป็นการโปรแกรมด้วยการใช้ไอคอนในการเขียนโปรแกรมเช่นโปรแกรม ciracore เป็นโปรแกรมที่พัฒนาโดยคนไทย [8] และโปรแกรม รอสวิซลสำหรับโปรแกรมหุ่นยนต์ [9] และการเชื่อมต่อที่สัมผัสและรู้สึกได้โหมด (tangible mode) [10-11]

จากงานวิจัยที่ผ่านมาในอดีตจะเห็นว่ามีความพยายามที่จะนำศาสตร์ในระดับสูงมา让孩子ได้เรียนรู้ด้วยแนวคิดที่ว่า การเรียนรู้สิ่งใหม่ด้วยวิธีการใหม่ โดยเน้นไปที่องค์ความรู้ที่จับต้องและสัมผัสได้ นอกจากนั้นยังมีงานวิจัยที่พยายามที่จะนำระบบปฏิบัติการของหุ่นยนต์มาใช้งานในรูปแบบที่ง่ายขึ้นในดั่งในแพลตฟอร์มทั้งสามข้างต้น โดยเฉพาะโหมด การเชื่อมต่อที่สัมผัสและรู้สึกได้แต่การใช้งานยังคงพัฒนาให้ผู้ใช้งานที่เป็นผู้ใหญ่ บทความนี้จึงคิดค้นพัฒนาหาวิธีการที่จะให้เด็กได้เรียนรู้ระบบปฏิบัติการของหุ่นยนต์ ที่เด็กสามารถเรียนรู้และเข้าใจง่ายโดยพัฒนาโปรแกรมรอส (ROS) ซึ่งเป็นการเขียนโปรแกรมที่สัมผัสและรู้สึกได้ในการควบคุมหุ่นยนต์เคลื่อนที่สำหรับนักเรียนระดับประถมศึกษา โดยในการออกแบบอ้างอิงลักษณะโปรแกรมที่สัมผัสและรู้สึกได้จากเรื่องราวของนิทานกระต่ายกับเต่า

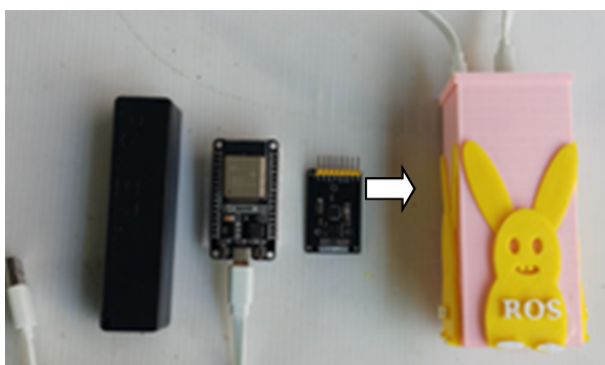
เช่น คำสั่ง “roscore” จะส่งรหัส 49 ไปยังหุ่นยนต์ โดยรหัส ID 49 เป็นคำสั่งในการเชื่อมต่อสื่อสารระบบ รอส เป็นต้น



รูปที่ 3 แผนภาพบล็อกการทำงานของหุ่นยนต์



รูปที่ 4 หุ่นยนต์



รูปที่ 5 ชุดอ่านคำสั่ง รอส เพื่อส่งไปยังหุ่นยนต์

3.2.2 การออกแบบหุ่นยนต์

หุ่นยนต์ประกอบด้วย 1) ส่วนประมวลผลใช้ ESP32MCU เป็นบอร์ดสำหรับรับรหัสโค้ดคำสั่งรอส จากชุดอ่านคำสั่ง โดยการส่งคำสั่งจะดำเนินการผ่านระบบ ESPNow 2) บอร์ด Raspberry pi 4 เป็นบอร์ดที่มีระบบปฏิบัติการควบคุมหุ่นยนต์รอส สำหรับรับคำสั่งแต่ละคำสั่ง ซึ่งทำงานโดยโปรแกรมภาษาไพทอน เพื่อประมวลผลแพ็กเก็ตคำสั่งระบบรอสแต่ละคำสั่ง และรอคำสั่ง

ถัดไป คำสั่งรหัสจะถูกส่งผ่านระบบพอร์ต RS232 เพื่อเปิดใช้งานแพ็กเก็ตของรอส 3) บอร์ด Arduino ใช้เพื่อรับคำสั่งควบคุมมอเตอร์หุ่นยนต์ผ่านพอร์ต RS232 จากบอร์ด Raspberry pi 4) มอเตอร์ควบคุมหุ่นยนต์ แผนภาพบล็อกการทำงานและหุ่นยนต์ ดังรูปที่ 3 และรูปที่ 4 การทำงานในการรับคำสั่ง เมื่อหุ่นยนต์ได้รับคำสั่งจากชุดคำสั่งจะทำให้หุ่นยนต์สายไปมาอยู่กับที่ เพื่อแสดงให้ทราบว่าได้รับคำสั่งแล้ว ถ้าไม่ได้รับคำสั่งหรือคำสั่งไม่สมบูรณ์หุ่นยนต์จะไม่สายไปมา

3.2.3 การออกแบบ RFID อ่านคำสั่งควบคุมหุ่นยนต์

การส่งคำสั่งรหัส ID ไปยังหุ่นยนต์ ประกอบด้วยบอร์ด ESP32 และโมดูล RFID RC522 รุ่นมินิ RFID 13.56 MHZ โดยโปรแกรมในโหมด ESPNow_Master slave สำหรับส่งข้อมูลรหัส RFID ไปยังหุ่นยนต์ โดยอุปกรณ์ทั้งหมดจะถูกบรรจุในกล่องสี่เหลี่ยมรูปกระต่ายสีเหลืองมีตัวอักษรที่ท้องของกระต่ายคำว่า “ROS” ดังรูปที่ 5

3.3 การโปรแกรมหุ่นยนต์ รอส

การในการควบคุมหุ่นยนต์ จะให้เด็กเรียนรู้โปรแกรมหุ่นยนต์รอส ภายใต้การดูแลของผู้ใหญ่ให้การแนะนำ โดยใช้ชุดอ่านคำสั่ง RFID เตะบนหลังเต่า เมื่อนำชุดอ่านคำสั่งเตะที่หลังเต่า RFID จะอ่านค่ารหัส ID ส่งรหัสคำสั่งไปยังตัวหุ่นยนต์ โดยเต่า 1 ตัว คือแพ็กเก็ตคำสั่งรอส 1 แพ็กเก็ต แสดงดังรูปที่ 6



รูปที่ 6 การอ่านคำสั่งแต่ละคำสั่ง

แนวคิดการรันแพ็กเก็ต Turtlesim ในโหมด Turtlesim_node เป็นการทดสอบการทำงานของระบบรอส โดยคำสั่งที่ประกอบด้วย “roscore” คือคำสั่งที่เริ่มต้นแต่ละโหมดเพื่อสื่อสาร

ระหว่างกันคือกลุ่มคำสั่ง ROS-Communication จากนั้นคำสั่ง “roslaunch turtlesim” คำสั่งนี้จะรันโนโหนดชื่อ roscore turtlesim_node ในแพ็คเกจ Turtlesim คือกลุ่มคำสั่งการ simulator ส่วนคำสั่ง “roslaunch turtle_teleop_key” เป็นคำสั่งกลุ่ม ROS-Behaviours คือคำสั่งการควบคุมการเคลื่อนที่ของเต่าด้วยการแสดงผลการเคลื่อนที่ของเต่าด้วยการบังคับควบคุมที่แป้นพิมพ์คอมพิวเตอร์ แสดงดังรูปที่ 7



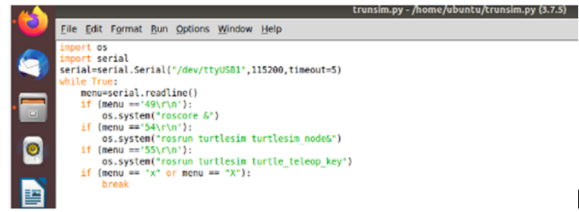
รูปที่ 7 คำสั่ง turtlesim_node

4. การทดสอบการใช้งาน

การทดสอบจะทดสอบกับเด็กในระดับชั้นประถมศึกษาปีที่ 4 ถึงชั้นประถมศึกษาปีที่ 6 ในการทดสอบระดับชั้นละ 3 คน เด็กแต่ละระดับไม่มีพื้นฐานการเขียนโปรแกรมมาก่อน การโปรแกรมประกอบด้วยการเรียนรู้ 3 ระดับคือ ระดับพื้นฐาน เป็นการโปรแกรมหุ่นยนต์เต่าบนจอคอมพิวเตอร์ด้วยคีย์บอร์ด ระดับกลาง การโปรแกรมหุ่นยนต์เคลื่อนที่ด้วยคีย์บอร์ด และระดับสูง การโปรแกรมหุ่นยนต์ตรวจจับสี

4.1 การโปรแกรมหุ่นยนต์เต่าด้วยคีย์บอร์ด

การโปรแกรมหุ่นยนต์เต่าด้วยคีย์บอร์ด ได้แก่ คำสั่ง roscore > roslaunch turtlesim> roslaunch turtle_teleop_key การทำงานทั้งหมดประมวลผลด้วย โปรแกรมภาษาไพทอนแสดงรูปที่ 8 และลักษณะการอ่านโปรแกรมแต่ละคำสั่งแสดงรูปที่ 9 โดยที่หุ่นยนต์จะรอรับคำสั่งจากพอร์ต rs232 เมื่อพบรหัสคำสั่งคือ รหัส 49 จะเรียกใช้คำสั่ง "roscore" เมื่อพบรหัส 54 จะเรียกใช้คำสั่ง "roslaunch turtlesim" และเมื่อพบรหัส 55 จะเรียกใช้คำสั่ง "roslaunch turtle_teleop_key" เป็นต้น



รูปที่ 8 การประมวลผลคำสั่งใน โปรแกรมภาษา ไพทอน



รูปที่ 9 การอ่านคำสั่งแต่ละคำสั่ง

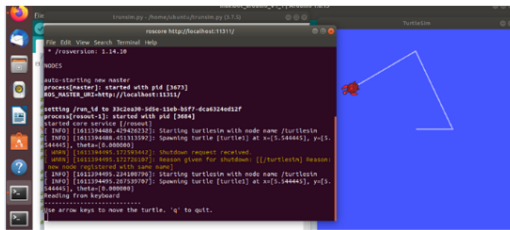
ผลการทดสอบการทำงาน เมื่อรันโปรแกรมภาษาไพทอน คำสั่งที่ใช้ประกอบด้วย roscore > roslaunch turtlesim> roslaunch turtle_teleop_key เสร็จสิ้นลง เด็กสามารถใช้แป้นพิมพ์คีย์บอร์ดควบคุมเต่าให้เคลื่อนที่ได้ตามต้องการ การทดลองนี้เป็นการทดสอบว่า เด็กสามารถควบคุมหุ่นยนต์ด้วยโปรแกรมรอส โปรแกรมที่สัมผัสและรู้สึกได้ ในการควบคุมเต่าให้เคลื่อนที่ด้วยคีย์บอร์ดคอมพิวเตอร์และแสดงผลการเคลื่อนที่ของเต่าบนจอคอมพิวเตอร์ได้ดังรูปที่ 10 อีกทั้งเป็นการทดสอบการทำงานของระบบรอสในเบื้องต้น ซึ่งทั้ง 3 คำสั่งเป็นชุดคำสั่งแรกที่ให้เด็กได้สัมผัสกับการโปรแกรมรอส เบื้องต้น

4.2 การโปรแกรมหุ่นยนต์ด้วยคีย์บอร์ด

หุ่นยนต์เคลื่อนที่ควบคุมด้วย รอสโปรแกรมที่สัมผัสและรู้สึกได้ด้วยแป้นพิมพ์คีย์บอร์ด การทดลองนี้เป็นการควบคุมหุ่นยนต์ด้วยแป้นพิมพ์คีย์บอร์ดด้วยคำสั่ง 3 คำสั่ง คือ : roscore> roslaunch portUSB> roslaunch teleop ดังรูปที่ 11

การทำงานของโปรแกรมแต่ละคำสั่ง แสดงด้วย โปรแกรมภาษาไพทอนดังรูปที่ 12 คำสั่งแรก คือ “roscore” โดยรหัสคำสั่งคือ 49 คำสั่งนี้เป็นการเรียกใช้คำสั่ง “roscore” คำสั่งที่ 2 คือ “roslaunch portUSB” รหัสคำสั่ง คือ 50 คำสั่งนี้จะเป็นการเปิด

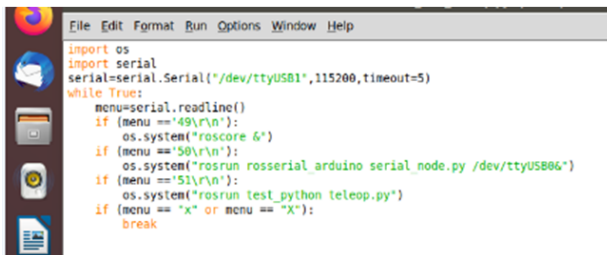
พอร์ต USB สำหรับการรับและส่งค่าในระบบควบคุมหุ่นยนต์ คำสั่งที่ 3 คำสั่ง: “roscat teleop” รหัสคำสั่งคือ 51 คำสั่งจะทำการเชื่อมต่อระบบของหุ่นยนต์กับการควบคุมระยะไกล ทั้ง 3 คำสั่งเป็นการให้เด็กเห็นการโปรแกรมหุ่นยนต์เคลื่อนที่ได้โดยใช้คำสั่งเพียง 3 คำสั่ง จากการทดลองให้เด็กควบคุมหุ่นยนต์ด้วยคีย์บอร์ดขนาดเล็ก พบว่าเด็กสามารถโปรแกรมทั้ง 3 คำสั่งได้ และใช้คีย์บอร์ดควบคุมหุ่นยนต์ได้



รูปที่ 10 การควบคุมเต่าให้เคลื่อนที่ด้วยคีย์บอร์ด



รูปที่ 11 คำสั่งการควบคุมหุ่นยนต์ด้วยแป้นพิมพ์



รูปที่ 12 โปรแกรมภาษาไพทอนควบคุม

นอกจากให้เด็กได้โปรแกรมคำสั่งแต่ละคำสั่งแล้ว ยังให้เด็กได้ฝึกแก้ปัญหาติดโปรแกรม โดยการนำคำสั่งบางคำสั่งออกจากระบบการโปรแกรม เช่นคำสั่ง roscore ซึ่งพบว่าเด็กสามารถ

เรียนรู้และเข้าใจการทำงานของระบบบรอดได้ ซึ่งคำสั่ง roscore เป็นคำสั่ง มาสเตอร์ จะต้องมีการสั่งนี้ระบบถึงจะทำงานได้

4.3 การโปรแกรมหุ่นยนต์ตรวจจับสี

การทดสอบจะทดสอบการใช้คำสั่งตรวจจับสี คำสั่งทั้งหมดประกอบด้วย 5 คำสั่ง ประกอบด้วย roscore >roscat portUSB> roscatcmd> roscatteleop> roscatdetect_color โดยที่คำสั่ง roscore จะเป็นคำสั่ง master ส่วนอีกที่เหลืออีก 4 คำสั่งจะเป็นคำสั่ง node ดังรูปที่ 13



รูปที่ 13 คำสั่งควบคุมการตรวจจับสี



รูปที่ 14 การตรวจจับสีของหุ่นยนต์

การทดสอบเมื่อเด็กโปรแกรมคำสั่งเสร็จแล้วทั้ง 5 คำสั่งจะนำเต่าที่มีสีต่างๆ มาทดสอบกับหุ่นยนต์ โดยให้หุ่นยนต์ติดตามรูปเต่าที่มีสีตามลักษณะที่โปรแกรมไว้ รูปที่ 14 ให้หุ่นยนต์ติดตามเต่าสีส้ม จากการทดลองพบว่า เด็กสามารถโปรแกรมหุ่นยนต์ได้ และทำการทดสอบให้หุ่นยนต์ติดตามเต่าสีส้มพบว่าหุ่นยนต์สามารถติดตามหุ่นยนต์เต่าสีส้มได้เช่นกัน โดยถ้าเป็นสีอื่นหุ่นยนต์จะไม่ติดตาม

5. ผลการทดสอบและการวิจารณ์ผล

การประเมินการเขียนโปรแกรมรอส สำหรับการควบคุมหุ่นยนต์ สำหรับนักเรียนระดับประถมศึกษา ทำการทดสอบใน 2 ลักษณะ คือ ทดสอบด้านฮาร์ดแวร์และซอฟต์แวร์ และ ทดสอบการเรียนรู้ของเด็ก

5.1 ผลการทดสอบด้านฮาร์ดแวร์และซอฟต์แวร์

การทดสอบฮาร์ดแวร์และซอฟต์แวร์ พบว่าเครื่องอ่าน รหัส RFID สามารถอ่านข้อมูลได้อย่างสมบูรณ์โดยการทดสอบการอ่าน 10 ครั้ง พบว่าระบบสามารถอ่านค่าได้ 10 ครั้งโดยไม่มีข้อผิดพลาด สำหรับหุ่นยนต์สามารถรับคำสั่งจากชุดส่งคำสั่ง ได้ทุกครั้งไม่ผิดพลาดเช่นกัน โดยเมื่อหุ่นยนต์รับรหัสคำสั่งได้แล้วจะหมุนตัวซ้ายไปมาได้ทุกครั้ง ซึ่งเป็นการแสดงให้เห็นว่าหุ่นยนต์ได้รับคำสั่งที่แต่ละบิตตัวได้ถูกส่งไปที่หุ่นยนต์สมบูรณ์แล้ว เมื่อคำสั่งแต่ละคำสั่งถูกส่งไปที่หุ่นยนต์แล้ว ทำการทดสอบการทำงานของตัวหุ่นยนต์ พบว่าหุ่นยนต์สามารถทำงานได้ตามคำสั่งที่ส่งไปยังหุ่นยนต์ได้ไม่มีความผิดพลาด

5.2 ผลการทดสอบการเรียนรู้ของเด็ก

การทดสอบกับเด็กเพื่อให้เด็กสามารถสร้างความรู้เกี่ยวกับโปรแกรมรอส ตามพื้นฐานทฤษฎีการเรียนรู้คอนสตรัคชันนิซึม ในการทดลองผู้วิจัยได้ช่วยเหลือเด็กในการรันโปรแกรมที่เด็กไม่สามารถรันได้ เช่น โปรแกรมภาษาไพทอน โปรแกรม รอส จากแนวคิดการทดสอบกรณีศึกษาการทดสอบระบบ รอส ด้วยคำสั่ง `roslaunch turtlesim turtlesim3.launch` และคำสั่ง `roslaunch turtlesim turtlesim3_teleop_key.launch` พบว่าเด็กสามารถโปรแกรมทั้ง 3 คำสั่งโดยใช้กระดุมแต่ละบิตตัวได้ เด็กสามารถใช้แป้นพิมพ์เพื่อควบคุมตัวให้เคลื่อนไหวยบนจอคอมพิวเตอร์ได้ แต่เด็กยังไม่เข้าใจความหมายของทั้งสามคำสั่ง สิ่งที่ได้คือ เด็กมีความสนุกสนานและตื่นตัวในการควบคุมตัวบนหน้าจอคอมพิวเตอร์ให้เคลื่อนที่ไปมาและแย่งชิงกันควบคุมตัวโดยการแย่งกดคีย์บอร์ด เมื่อทดสอบการแก้ปัญหาการติดกั ผู้วิจัยได้นำคำสั่ง `roscat /turtle1/cmd_vel` ออกจากระบบการโปรแกรม มีเพียง คำสั่ง `roslaunch turtlesim turtlesim3.launch` เท่านั้น ซึ่งเมื่อใช้คีย์ควบคุมตัวจะไม่สามารถควบคุมตัวให้เคลื่อนที่ได้ เมื่อนำคำสั่ง `roscat /turtle1/cmd_vel` กลับมาโปรแกรมอีกครั้งให้คำสั่ง 3 คำสั่งเช่นเดิม พบว่าตัวสามารถใช้คีย์บอร์ดควบคุมการทำงานได้ เป็นการแสดงให้เห็นว่า การทำงานของ

ระบบจำต้องประกอบด้วยคำสั่ง `roscat` เด็กได้จับต้องคำสั่งรอสที่เป็นรูปเต๋าและชุดอ่านคำสั่งที่เป็นรูปกระดุม ได้เรียนรู้การเรียงลำดับคำสั่งด้วยสองมือ การอ่านคำสั่งด้วยชุดอ่านคำสั่งด้วยมือ ทำให้เด็กเกิดแนวคิดและเข้าใจการโปรแกรมหุ่นยนต์มีการโปรแกรมเรียงลำดับในลักษณะนี้ ตามแนวคิดทฤษฎีการเรียนรู้คอนสตรัคชันนิซึม

ในการทดลองอีกรูปแบบหนึ่งคือการควบคุมหุ่นยนต์ด้วยคีย์บอร์ด ผู้วิจัยได้ให้เด็กโปรแกรมเพียงสองคำสั่ง คือ คำสั่ง `roslaunch turtlesim turtlesim3.launch` และคำสั่ง `roslaunch turtlesim turtlesim3_teleop_key.launch` โดยไม่มีคำสั่ง `roscat` เด็กพบว่าหุ่นยนต์ไม่สามารถควบคุมด้วยแป้นพิมพ์ได้ ในครั้งนี้เด็กถามหาคำสั่ง `roscat` แล้วทำการโปรแกรมใหม่ โดยเพิ่มโปรแกรมคำสั่ง `roscat` หุ่นยนต์ก็สามารถทำงานได้ ผู้วิจัยได้อธิบายให้เด็กเข้าใจคำสั่ง `roscat` เป็นคำสั่งเชื่อมต่อการสื่อสารระหว่างสองคำสั่งที่ควบคุมหุ่นยนต์ `roscat` เป็นตัวหลัก (master) หรือหัวหน้า ส่วนอีก 2 คำสั่งเป็นลูกน้อง (node) ทำให้เด็กเข้าใจแนวคิดของระบบการโปรแกรมหุ่นยนต์ด้วยโปรแกรมรอส ที่มีการโปรแกรมในลักษณะของ มาสเตอร์ และ โหนด

ส่วนการทดสอบการตรวจจับสีเป็นการทดสอบคำสั่งที่มีความซับซ้อนยิ่งขึ้น โดยโปรแกรมคำสั่ง 5 คำสั่งดังนี้ `roscat > roslaunch turtlesim turtlesim3.launch > roslaunch turtlesim turtlesim3_teleop_key.launch > roslaunch turtlesim turtlesim3_detect_color.launch` การทดลองนี้เป็นการทดสอบการเรียนรู้การเขียนโปรแกรมที่ทำให้หุ่นยนต์สามารถตรวจจับสีวัตถุโดยใช้ความสามารถของโปรแกรม รอส เพื่อให้เด็กได้เรียนรู้การโปรแกรมที่ซับซ้อนยิ่งขึ้น ในการทดลองนี้จะทำให้หุ่นยนต์ตรวจจับวัตถุที่เป็นสีส้มเพียงสีเดียวและติดตามวัตถุสีส้ม ถ้าเป็นสีอื่นหุ่นยนต์จะไม่ติดตาม จากคำสั่ง 5 คำสั่ง เด็กสามารถโปรแกรมโน้ตทั้ง 4 คำสั่งโดยสามารถโปรแกรมโน้ตโน้ตก่อนหรือหลังก็ได้ ไม่จำเป็นต้องเรียงลำดับ ซึ่งในการโปรแกรมเด็กพยายามที่จะโปรแกรมแบบเรียงลำดับก่อนหลังจากการทดลองที่ผ่านมา แต่ในการทดลองนี้ มีเด็กที่โปรแกรมผิดพลาด แต่หุ่นยนต์สามารถทำงานได้ เด็กได้สอบถามผู้วิจัยว่าทำไมไม่ต้องเรียงลำดับคำสั่ง หุ่นยนต์ก็ทำงานได้ ปรากฏการณ์นี้คือการสร้างมโนทัศน์ให้กับเด็กในการเรียนรู้การเขียนโปรแกรมเชิงวัตถุ ทำให้เด็กเรียนรู้การโปรแกรมในรูปแบบนี้ว่าไม่จำเป็นต้องโปรแกรมแบบเรียงลำดับก็ได้ และถือเอาคำสั่ง 1 คำสั่งเป็นวัตถุซึ่งแนวคิดนี้คล้ายกับการ

เขียนโปรแกรมเชิงวัตถุ OOP (Object Oriented Programming) โดยให้มองทุกสิ่งทุกอย่างเป็นวัตถุ โปรแกรมคือ กลุ่มของวัตถุที่สื่อสารกันได้ [14] จากการทดลองทำให้เด็กได้เรียนรู้การเขียนโปรแกรมอีกรูปแบบหนึ่ง โดยใช้โปรแกรมรอสที่สัมผัสและรู้สึกได้ ถ้าใช้โปรแกรม รอส ที่เป็นการโปรแกรมในโหมดเท็กซ์ เด็กจะไม่สามารถโปรแกรมหุ่นยนต์ตั้งในการทดลองได้ เนื่องจากโปรแกรมมีความซับซ้อนและเป็นนามธรรมสูง

ข้อแนะนำ ในการทดลองผู้วิจัยได้ใช้โปรแกรมนี้กับเด็กในระดับประถมศึกษาปีที่ 1 ถึงชั้นประถมศึกษาปีที่ 3 พบว่า เด็กในช่วงชั้นวัยนี้ไม่สามารถโปรแกรมและเข้าใจได้ จึงเลื่อนระดับไป ช่วงชั้นต่อไปคือ ระดับช่วงชั้นประถมศึกษาปีที่ 4 ถึง ชั้นประถมศึกษาปีที่ 6 ซึ่งพบว่าสามารถโปรแกรมและเข้าใจได้ แสดงให้เห็นว่า โปรแกรมที่พัฒนาขึ้นสามารถใช้ได้กับเด็กในระดับช่วงชั้นตั้งแต่ระดับประถมศึกษาปีที่ 4 ขึ้นไปจากผลการทดลองใช้งาน ควรใช้งานทดลองกับเด็กในระดับชั้นประถมศึกษาที่ 4 ขึ้นไป

6. บทสรุป

การพัฒนาการเขียนโปรแกรมรอสควบคุมหุ่นยนต์สำหรับนักเรียนระดับประถมศึกษา การออกแบบโปรแกรมรอสที่สัมผัสและรู้สึกได้ ประกอบด้วย 2 ส่วน คือ 1) โปรแกรมรอสที่สัมผัสและรู้สึกได้ประกอบด้วยคำสั่ง 7 คำสั่งประกอบด้วย “roscore” “roscorportUSB” “roscuncmd” “roscunteleop roscundetect_color” “roscunTurtlesim” และ “roscun turtle_teleop_key” 2) หุ่นยนต์ประกอบด้วยบอร์ด Raspberry Pi 4 โมดูล ESP32 และบอร์ด Arduino ส่วนซอฟต์แวร์ประกอบด้วยโปรแกรม python โปรแกรม Arduino IDE และ โปรแกรม รอส (ROS) การทดลองใช้งานกับเด็กในระดับชั้นประถมศึกษาปีที่ 4 ถึงชั้นประถมศึกษาปีที่ 6 ในการทดสอบระดับชั้นละ 3 คน ทำการทดสอบ 3 กรณี คือ 1) การโปรแกรมหุ่นยนต์ด้วยคีย์บอร์ด พบว่าเด็กสามารถโปรแกรมหุ่นยนต์ควบคุมบนจอคอมพิวเตอร์ได้ 2) การโปรแกรมหุ่นยนต์เคลื่อนที่ด้วยคีย์บอร์ด พบว่าเด็กสามารถโปรแกรมหุ่นยนต์เคลื่อนที่ด้วยโปรแกรมรอสที่สัมผัสและรู้สึกได้ 3) การโปรแกรมหุ่นยนต์ตรวจจับสี พบว่าเด็กสามารถโปรแกรมหุ่นยนต์เคลื่อนที่ที่สามารถตรวจจับสีได้

7. กิตติกรรมประกาศ

ขอขอบคุณมหาวิทยาลัยราชภัฏเพชรบุรีและโปรแกรมเทคโนโลยีไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเพชรบุรี และคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเพชรบุรี

8. เอกสารอ้างอิง

- [1] S. Papert. *Mindstorms: Children Computers and Powerful Ideas*, Cambridge, Massachusetts: Perseus Publishing, 1980.
- [2] Y. Kafai, and M. Resnick. *Constructionism in Practice Designing Thinking and Learning ;j'in a Digital World*, Lawrence Erlbaum Associates Publishing, 1996.
- [3] O. Zuckerman. *System Blocks: Learning about Systems Conceptsthrough Hands-on Modeling and Simulation*. M.S. Thesis, Massachusetts Institute of Technology, 2004.
- [4] R. Williams. *PopBots Leveraging Social Robots to Aid Preschool Children’s Artificial Intelligence Education*. M. S. Thesis, Massachusetts Institute of Technology, 2016.
- [5] A. Strawhacker, *Biodesign Education in Early Childhood: A Design-Research Study of the Tangible CRISPEE Technology and Learning Intervention*, Doctoral dissertation, Tufts University, Medford, MA, 2020.
- [6] S. Meadthaisong and T. Meadthaisong, "Smart Farming Using Internet of Thing (IoT) in Agriculture by Tangible Programming for Children". *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. 24-27 June. Phuket Thailand : pp. 611-614, 2020.

- [7] Open Robotics. (20 January 2024). *ROS-Robot Operating System*. [Online] Available: <https://www.ros.org/>
- [8] Cirarobotics (28 January 2024). *Ciracore*. [Online] Available: <https://www.cira-ai.com/>
- [9] Karaca, Mustafa and Uğur Yayan. "ROS Based Visual Programming Tool for Mobile Robot Education and Applications," *ACM Robotics*., pp. 1-10, 2020. doi.org/10.48550/arXiv.2011.13706
- [10] Y. S. Sefidgar, P. Agarwal and M. Cakmak, "Situating Tangible Robot Programming". *ACM/IEEE International Conference on Human-Robot Interaction*. 6-9 March. Vienna Austria : pp. 473-482, 2017.
- [11] Yasaman S. Sefidgar, Thomas Weng, Heather Harvey, Sarah Elliott, and Maya Cakmak, "RobotIST: Interactive Situated Tangible Robot Programming". *Conference Paper, Proceedings of Symposium on Spatial User Interaction*. 13-14 October. Berlin Germany : pp. 141-149, 2018.
- [12] H.Ishii, & B.Ullmer, "Tangible Bits: Towards Seamless Interfaces between People Bit and Atoms". *Proceedings of ACM Conference on Human Factors & Computing Systems*. 22-27 March. Atlanta Georgia USA : pp. 234-241, 1997.
- [13] M. Resnick and B. Silverman, "Some reflections on designing construction kits for kids". *Proceedings of the conference on Interaction design and children*. 8-10 June. New York USA : pp. 117-122, 2005.
- [14] Alan Kays. (30 January 2024). *Definition of Object Oriented*. [Online] Available: <https://wiki.c2.com/?AlanKaysDefinitionOfObjectOriented>.