

การศึกษาเปรียบเทียบวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ ด้วยแพลตฟอร์ม Arduino

A Comparison Study on Suitable Methods for Developing an Arduino based Autonomous Mobile Robots

นฤพนธ์ พนาวงศ์*, เอกสิทธิ์ สิริธิตสมาน, ภัคจิรา ศิริโส

Naruepon Panawong*, Akkasit Sittisaman, Pakjira Sirisom

ภาควิชาวิทยาศาสตร์ประยุกต์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครสวรรค์

Department of Applied Science, Faculty of Science and Technology, Nakhon Sawan Rajabhat University

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อ 1) ศึกษาวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ 2) ออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino 3) เปรียบเทียบประสิทธิภาพและหาวิธีที่เหมาะสมสำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ โดยแบ่งวิธีการดำเนินงานวิจัยออกเป็น 2 ส่วน คือ 1) การออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ ผู้วิจัยใช้แพลตฟอร์ม Arduino ประกอบไปด้วย บอร์ดอิเล็กทรอนิกส์ ESP32 ในการประมวลผลคำสั่งต่าง ๆ โมดูล HC-SR04 อัลตราโซนิกเป็นอุปกรณ์วัดระยะทางเพื่อตรวจจับสิ่งกีดขวาง IR Infrared เป็นอุปกรณ์ตรวจจับเส้นทางในการเคลื่อนที่ของหุ่นยนต์ และโมดูล L298N เป็นตัวขับเคลื่อนหุ่นยนต์ทำให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งต่าง ๆ ได้ และใช้ภาษา C สำหรับ Arduino ในการเขียนโปรแกรมตามอัลกอริทึมการค้นหาเส้นทาง 2) การพัฒนาอัลกอริทึมที่ใช้สำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ โดยใช้ 4 อัลกอริทึมที่นิยมนำมาใช้งานค้นหาเส้นทาง ได้แก่ คือ อัลกอริทึมการค้นหาในแนวกว้างก่อน อัลกอริทึมการค้นหาเชิงลึกก่อน อัลกอริทึมการค้นหาแบบดีที่สุดในสุดก่อน และอัลกอริทึมการค้นหาแบบเอ-สตาร์ จากการทดสอบการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติพบว่าอัลกอริทึมที่ใช้เวลาในการเดินทางเฉลี่ยน้อยที่สุด คือ อัลกอริทึมการค้นหาแบบเอ-สตาร์ ใช้เวลาเฉลี่ยเท่ากับ 39.12 วินาที ส่วนอัลกอริทึมที่ใช้เวลาในการเดินทางเฉลี่ยมากที่สุด คือ อัลกอริทึมการค้นหาเชิงลึกก่อน ใช้เวลาเฉลี่ยเท่ากับ 48.64 วินาที ซึ่งทุกอัลกอริทึมใช้ระยะทางในการเดินทางเท่ากัน คือ 10 บล็อก

คำสำคัญ: หุ่นยนต์ อัลกอริทึม ภาษา C สำหรับ Arduino

* Corresponding author : naruepon.p@nsru.ac.th

Abstract

This research aims to study the appropriateness methods for developing an autonomous mobile robot, design and develop an autonomous mobile robot based on Arduino platform, and last but not least, to compare and search for suitable robot movement algorithms. The research methodology is composed of two parts. First, the design and development of an autonomous mobile robot using the Arduino platform. The components of the proposed robot are ESP32 used for processing tasks, a HC-SR04 ultrasonic sensor module used for measuring the distance between the robot and obstacles, IR Infrared used for path detection while the robot is moving, and L298N drive module for controlling the movement of the proposed robot. The algorithm of path finding for the proposed robot are developed in C for Arduino. The second part is the development of path search algorithm for the autonomous mobile robot using the four popular path search algorithms which are Breadth First Search, Depth First Search, Best First Search, and A* Search. The experiments on the movement of the autonomous mobile robot shows that the A* Search algorithm has the shortest duration of 39.12 second while the Depth First Search has the longest duration of 48.64 second. Each algorithm moves the autonomous mobile robot for 10 blocks.

Keywords: Robot, Algorithm, C for Arduino

1. บทนำ

ปัจจุบันมีการใช้และพัฒนาหุ่นยนต์ที่ทำงานแบบอัตโนมัติในรูปแบบต่าง ๆ มากมาย เช่น หุ่นยนต์สำรวจ หุ่นยนต์เสิร์ฟอาหาร หุ่นยนต์ทำความสะอาดพื้น เป็นต้น ซึ่งสอดคล้องกับการพัฒนาประเทศในยุคประเทศไทย 4.0 รวมถึงสอดคล้องกับการพัฒนาการศึกษาของประเทศที่ให้เด็ก ๆ ได้เรียนวิชาวิทยาการคำนวณ ตั้งแต่ระดับประถมเพื่อเป็นพื้นฐานในการสร้างนวัตกรรมที่ใช้เทคโนโลยีเป็นฐาน ทั้งนี้การนำรูปแบบการสอน Coding ด้วยการให้หุ่นยนต์จะช่วยกระตุ้นการเรียนรู้ของผู้เรียนได้ดี อีกทั้งมีการจัดแข่งขันหุ่นยนต์รูปแบบต่าง ๆ ทั้งแบบบังคับด้วยรีโมท หรือ ทำภารกิจแบบอัตโนมัติ สำหรับการเดินทางไม่ว่าจะเป็นในรูปแบบของการท่องเที่ยว หรือ การขนส่งสินค้า หากไม่มีการวางแผนการเดินทางแล้ว อาจต้องใช้เวลาเดินทางมากกว่าปกติเพื่อให้ไปถึงจุดหมาย นอกจากเวลาที่เสียไปแล้วยังสิ้นเปลืองน้ำมันมากขึ้นเช่นกัน หากเป็นภาคเอกชนนั้นถือว่าเกิดความเสียหายต่อต้นทุนในการขนส่งสินค้าอย่างมาก สำหรับอัลกอริทึมที่ถูกใช้ในการวางแผนการเดินทางหรือขนส่งสินค้า หรือ ในเกมส์อย่างแพคแมน เช่น อัลกอริทึมฮิวริสติก อัลกอริทึมเอ-สตาร์ อัลกอริทึมการค้นหาในแนวกว้างก่อน อัลกอริทึมการค้นหาเชิงลึกก่อน อัลกอริทึมการค้นหาแบบดีที่สุดในตอนต้น เป็นต้น ล้วนเป็นอัลกอริทึมที่สามารถนำมาประยุกต์ใช้กับงานวิจัยนี้ได้

ปัจจุบันมีงานวิจัยเกี่ยวกับการเพิ่มประสิทธิภาพการจัดเส้นทางรถขนส่งสินค้าโดยใช้อัลกอริทึมต่าง ๆ เพื่อช่วยลดต้นทุนทางพลังงาน หรือ แก้ปัญหาการขนส่งที่ล่าช้าด้วยการใช้เส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุด รวมถึงการใช้หุ่นยนต์มาประยุกต์ใช้ในการขนส่ง หรือ จำลองการเคลื่อนที่แบบไร้คนขับ ซึ่งมีงานวิจัยที่ผู้วิจัยได้ให้แนวคิดและสามารถนำไปประยุกต์ใช้กับงานวิจัยในครั้งนี้ได้ เช่น Alli et al. (2018) ได้พัฒนาหุ่นยนต์หลบหลีกสิ่งกีดขวางที่ใช้แพลตฟอร์ม Arduino สำหรับยานพาหนะไร้คนขับ ประกอบด้วย 1) ไมโครคอนโทรลเลอร์ Arduino Uno เป็นตัวควบคุมการทำงานของหุ่นยนต์ โดยเขียนโปรแกรมด้วยภาษา C++ 2) เซนเซอร์ IR และใช้เซนเซอร์อัลตราโซนิกในการตรวจจับสิ่งกีดขวางและติดตั้งไว้ด้านหน้าของหุ่นยนต์ 3) DC Motor เป็นตัวขับเคลื่อนล้อหรือขาของหุ่นยนต์ และ 4) แหล่งจ่ายไฟ ที่ทำหน้าที่จ่ายกระแสไฟให้กับไมโครคอนโทรลเลอร์ จากการทดลองพบว่าระบบมีประสิทธิภาพ 85% รวมถึง Pavithra and Subramanya (2018) ได้อธิบายการออกแบบและพัฒนาหุ่นยนต์หลบสิ่งกีดขวางบนแพลตฟอร์ม Arduino โดยใช้เซนเซอร์อัลตราโซนิกสำหรับตรวจจับสิ่งกีดขวาง และใช้ไมโครคอนโทรลเลอร์ ATmega328 ควบคุมการทำงานของหุ่นยนต์ ซึ่งผลการทดลองพบว่าหุ่นยนต์สามารถเคลื่อนที่ได้อย่างอัตโนมัติ เมื่อพบสิ่งกีดขวางในระยะ 30 เซนติเมตรก็สามารถเคลื่อนที่ไปทางอื่นได้ จากนั้น Ginting and Sembiring (2019). ได้แสดงการเปรียบเทียบการค้นหาไฟล์บนระบบปฏิบัติการวินโดวส์ ระหว่างอัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) กับอัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) จากการทดลองพบว่าอัลกอริทึมการค้นหาในแนวกว้างก่อนจะทำงานได้ดีสำหรับการค้นหาแนวกว้างและเหมาะกับการค้นหาไฟล์ อีกทั้ง Lim and Ng. (2021). ได้ทดลองสร้างหุ่นยนต์ค้นหาเส้นทางด้วยอัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) แบบเดินตามเส้น โดยใช้แพลตฟอร์ม Arduino ในการออกแบบหุ่นยนต์ประกอบไปด้วยบอร์ด Uno R3 มอเตอร์เกียร์ โมดูล L298N เพื่อควบคุมการทำงานของมอเตอร์เกียร์ และติดตั้งเซนเซอร์ IR ไว้ด้านหน้าของหุ่นยนต์เรียงกัน 5 ตำแหน่ง เพื่อใช้ตรวจจับเส้นขาว/ดำ และใช้ภาษา C เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์ ซึ่งผลการทดลองพบว่าหุ่นยนต์สามารถค้นหาเส้นทางที่เหมาะสมโดยการจัดเส้นทางที่ไม่ต้องการได้ และสามารถแก้ปัญหาการค้นหาเส้นทางที่แตกต่างกันได้เป็นอย่างดี รวมถึงยังสามารถเรียนรู้ และจดจำเส้นทางเพื่อกลับไปยังจุดเริ่มต้นและกลับสู่เป้าหมายได้สำเร็จ อีกทั้ง Hii et al. (2021). ได้ทำสร้างหุ่นยนต์บนแพลตฟอร์ม Arduino เพื่อใช้ในการค้นหาเส้นทางในเขาวงกตโดยใช้อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาด้านซ้ายก่อน (Left-Hand Rule) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) เพื่อใช้ในการวางแผนการเดินทางออกจากเขาวงกตทั้งหมด 3 ขนาด คือ 5x5, 10x10 และ 15x15 จากการทดลองพบว่าอัลกอริทึมการค้นหาแบบเอ-สตาร์ ใช้เวลาน้อยที่สุด ส่วนอัลกอริทึมการค้นหาในแนวกว้างก่อนและอัลกอริทึมการค้นหาแบบเอ-สตาร์ ได้ระยะทางที่สั้นที่สุด สอดคล้องกับงานวิจัยของ Irawan et al. (2021). หุ่นยนต์ทำความสะอาดอัตโนมัติ โดยใช้บอร์ด Arduino UNO ควบคุมการทำงานของหุ่นยนต์ ใช้เซนเซอร์อัลตราโซนิกสำหรับตรวจจับสิ่งกีดขวาง ใช้โมดูล L298 ควบคุมการทำงานของมอเตอร์ ซึ่งหุ่นยนต์สามารถทำงานได้ตรงตามวัตถุประสงค์ของงานวิจัยได้เป็นอย่างดี รวมถึงงานวิจัยของ Navare et al. (2022) ที่ใช้แพลตฟอร์ม Arduino พัฒนาหุ่นยนต์ตรวจจับระเบิด โดยใช้ Arduino MEGA (ATmega 2560) เป็นบอร์ดควบคุมการทำงานของหุ่นยนต์ ใช้ L293D

เป็นโมดูลควบคุมการทำงานของมอเตอร์ ใช้เซนเซอร์ตรวจสอบโลหะ และกล้องไร้สายเพื่อตรวจสอบว่าเป็นระเบิดหรือไม่ และใช้ buzzer ส่งเสียงแจ้งเตือน โดยหุ่นยนต์สามารถควบคุมการทำงานจากระยะไกลผ่านเครื่องคอมพิวเตอร์ จากการทดลองพบว่าหุ่นยนต์สามารถตรวจจับวัตถุที่เป็นโลหะได้และสามารถถ่ายภาพวัตถุส่งกลับมาให้ผู้ควบคุมสามารถพิจารณาวัตถุนั้นเป็นระเบิดหรือไม่ นอกจากนี้งานวิจัยของน้ำฝน พาพันธ์ และภัทรานิษฐ์ แก้วประดิษฐ์ (2563) ได้ประยุกต์ใช้อัลกอริทึมแบบประหยัดเพื่อจัดเส้นทางที่เหมาะสมและเพิ่มประสิทธิภาพในการขนส่งในแต่ละรอบการขนส่งให้กับโรงงานเม็ดพลาสติกแห่งหนึ่ง โดยการคำนวณจากระยะเวลาในการขนส่ง ใช้ข้อมูลย้อนหลัง 1 เดือน เพื่อให้ได้เส้นทางใหม่ที่สั้นที่สุดในรอบการขนส่ง โดยมีเริ่มจากการสร้างคำตอบเริ่มต้น จากนั้นคำนวณหาค่าความประหยัด จัดลำดับค่าระยะทางประหยัดจากค่ามากที่สุดไปยังค่าน้อยที่สุด โดยจะคำนวณหาค่าประหยัดของแต่ละคู่ลูกค้าและนำค่าที่สูงที่สุดมาเป็นเกณฑ์ในการตัดสินใจเลือกคู่ลูกค้าที่จะทำการส่งและหลังจากการใช้อัลกอริทึมแบบประหยัด ผลการทดลองพบว่า ระยะทางที่ได้จากการจัดเส้นทางแบบใหม่ ลดลงจากเดิมคิดเป็น 34.16% สามารถลดระยะทางจาก 3,180.60 กิโลเมตร เหลือเพียง 2,094.10 กิโลเมตร

จากปัญหาที่ได้กล่าวมาและการทบทวนวรรณกรรมจากงานวิจัยข้างต้น ผู้วิจัยจึงได้ทำการออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติเพื่อจำลองการเดินทางด้วยวิธีการที่เหมาะสมในด้านเวลาและระยะทาง ซึ่งวิธีการที่เหมาะสมก็คืออัลกอริทึมต่าง ๆ ที่จะมาช่วยให้หุ่นยนต์คิดและวิเคราะห์การเดินทางจากจุดเริ่มต้นไปยังจุดหมายได้ โดยงานวิจัยนี้จะใช้แพลตฟอร์ม Arduino มาสร้างต้นแบบหุ่นยนต์เคลื่อนที่แบบอัตโนมัติและใช้ภาษา C สำหรับ Arduino สำหรับเขียนโปรแกรมลงบนบอร์ดอิเล็กทรอนิกส์ จากนั้นจะทำการเปรียบเทียบในแต่ละอัลกอริทึมเพื่อหาวิธีการที่เหมาะสมและมีประสิทธิภาพ เพื่อนำไปประยุกต์ใช้สำหรับการขนส่งสินค้าหรือวางแผนการเดินทางท่องเที่ยว

2. วัตถุประสงค์การวิจัย

- 2.1 เพื่อศึกษาวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ
- 2.2 เพื่อออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino
- 2.3 เพื่อเปรียบเทียบประสิทธิภาพและหาวิธีที่เหมาะสมสำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ

3. วิธีดำเนินการวิจัย

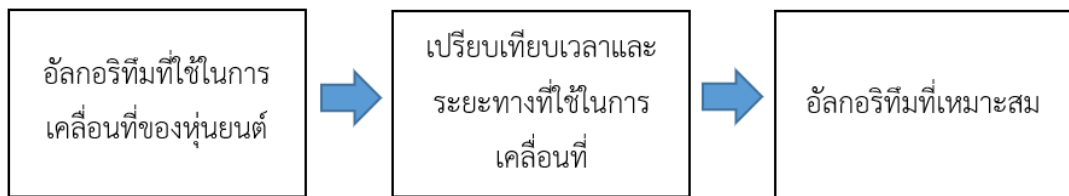
งานวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการวิจัยแบ่งเป็น 2 ส่วน คือ การออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ และการพัฒนาอัลกอริทึมที่ใช้สำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 การออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ

ในหัวข้อนี้ผู้วิจัยได้ออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ ซึ่งมีรายละเอียดดังต่อไปนี้

แนวคิดการศึกษาของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ

ในหัวข้อนี้ผู้วิจัยได้ออกแบบแนวคิดการศึกษาของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino ดังภาพที่ 1



ภาพที่ 1 แนวคิดการศึกษาของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

จากภาพที่ 1 แสดงแนวคิดการศึกษาของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino โดยมีขั้นตอนการทำงานดังรายละเอียดต่อไปนี้

1. หุ่นยนต์จะเคลื่อนที่แบบอัตโนมัติโดยใช้ 4 อัลกอริทึมที่นิยมนำมาใช้งานค้นหาเส้นทาง คือ อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) ซึ่งในการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัตินั้นจะมีโมดูล HC-SR04 อัลตราโซนิกในการวัดระยะทางเพื่อตรวจสอบสิ่งกีดขวาง โดยหากมีสิ่งกีดขวางหุ่นยนต์จะหยุดและไม่เดินไปชนวัตถุนั้น จากนั้นจะคำนวณเส้นทางใหม่และเดินทางต่อไปให้อยู่ในขอบเขตด้วยการใช้ IR Infrared ในการตรวจจับเส้นทางในการเคลื่อนที่ของหุ่นยนต์ ทั้งนี้เพื่อให้หุ่นยนต์เดินให้ตรงตามเส้นทาง

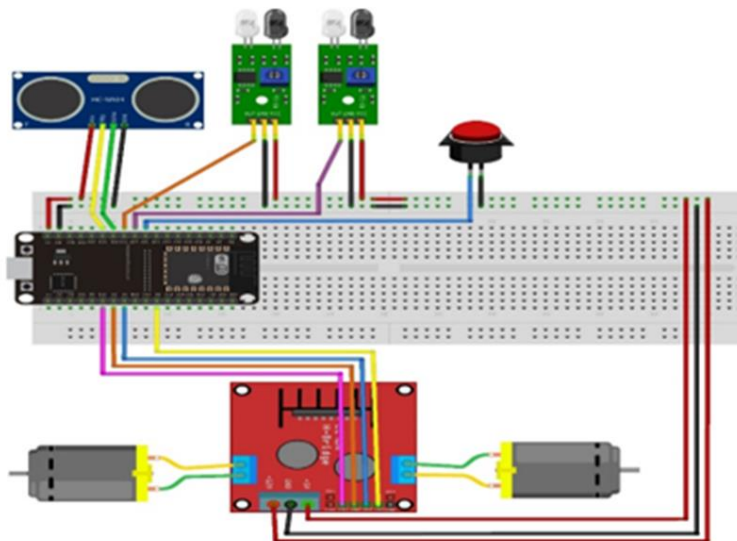
2. นำเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์และระยะทางมาเปรียบเทียบเพื่อหาอัลกอริทึมที่เหมาะสมสำหรับการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

การออกแบบการต่อวงจรอิเล็กทรอนิกส์

ในงานวิจัยนี้คณะผู้วิจัยใช้โปรแกรม Fritzing ซึ่งเป็นโปรแกรม Open Source ที่ใช้ในการวางแผนการต่อวงจรบนแผงอิเล็กทรอนิกส์หรือบอร์ดต่าง ๆ แบบเสมือนจริงมาออกแบบการต่อวงจรอิเล็กทรอนิกส์ของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino ประกอบด้วยบอร์ดอิเล็กทรอนิกส์ โมดูล HC-SR04 (เซนเซอร์อัลตราโซนิก) โมดูล IR infrared (ตรวจจับเส้นขาวดำ) และปุ่มสั่งให้หุ่นยนต์เคลื่อนที่ ดังภาพที่ 2 ซึ่งสามารถอธิบายขั้นตอนการต่อวงจรระหว่างบอร์ด ESP32 กับโมดูล HC-SR04 โมดูล IR infrared จำนวน 2 ตัว โมดูล L298N และปุ่มกด Switch ได้ดังนี้

- ขา VCC ของบอร์ดต่อเข้ากับขา VCC ของโมดูล HC-SR04
- ขา GND ของบอร์ดต่อเข้ากับขา GND ของโมดูล HC-SR04
- ขา 13 ของบอร์ดต่อเข้ากับขา Trig ของโมดูล HC-SR04
- ขา 12 ของบอร์ดต่อเข้ากับขา Echo ของโมดูล HC-SR04
- ขา VCC ของบอร์ดต่อเข้ากับขา VCC ของโมดูล IR infrared ตัวที่ 1
- ขา GND ของบอร์ดต่อเข้ากับขา GND ของโมดูล IR infrared ตัวที่ 1
- ขา 14 ของบอร์ดต่อเข้ากับขา OUT ของโมดูล IR infrared ตัวที่ 1 ขา

- VCC ของบอร์ดต่อเข้ากับขา VCC ของโมดูล IR infrared ตัวที่ 2
- ขา GND ของบอร์ดต่อเข้ากับขา GND ของโมดูล IR infrared ตัวที่ 2
- ขา 27 ของบอร์ดต่อเข้ากับขา OUT ของโมดูล IR infrared ตัวที่ 2
- ขา VCC ของบอร์ดต่อเข้ากับขา 12v ของโมดูล L298N
- ขา GND ของบอร์ดต่อเข้ากับขา GND ของโมดูล L298N
- ขา VCC ของบอร์ดต่อเข้ากับขา 5v ของโมดูล L298N
- ขา 15 ของบอร์ดต่อเข้ากับขา A1 ของโมดูล L298N
- ขา 2 ของบอร์ดต่อเข้ากับขา A2 ของโมดูล L298N
- ขา 4 ของบอร์ดต่อเข้ากับขา B1 ของโมดูล L298N
- ขา 5 ของบอร์ดต่อเข้ากับขา B2 ของโมดูล L298N
- ขา 26 ของบอร์ดต่อเข้ากับขา บวก (+) ของปุ่มกด Switch
- ขา GND ของบอร์ดต่อเข้ากับขา GND ของปุ่มกด Switch



ภาพที่ 2 การต่อวงจรอิเล็กทรอนิกส์ของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

การออกแบบโครงสร้างของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

ในหัวข้อนี้ผู้วิจัยได้ออกแบบโครงสร้างของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino ดัง

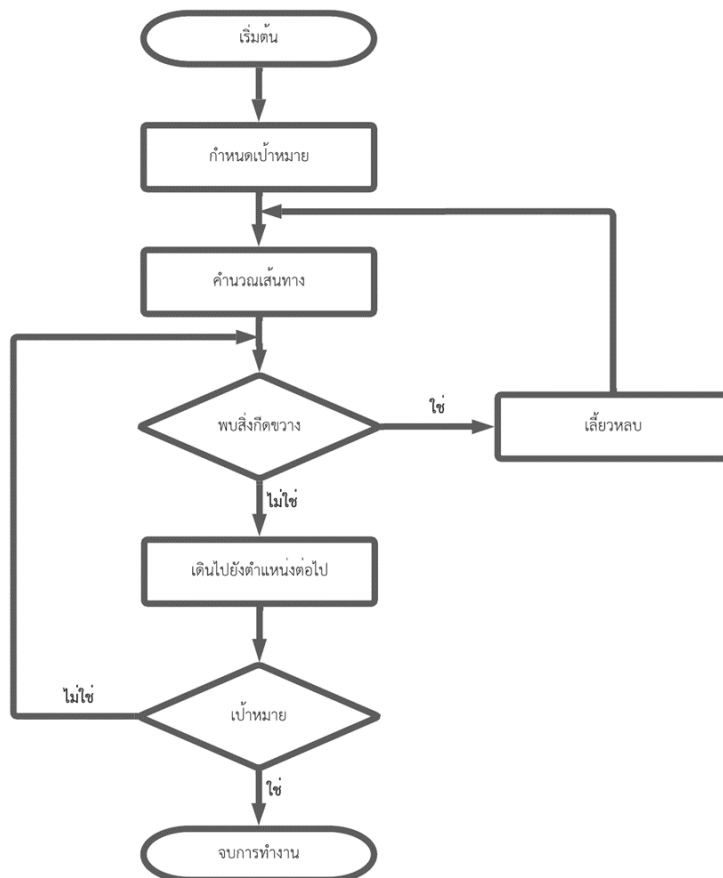
ภาพที่ 3



ภาพที่ 3 การออกแบบโครงสร้างของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

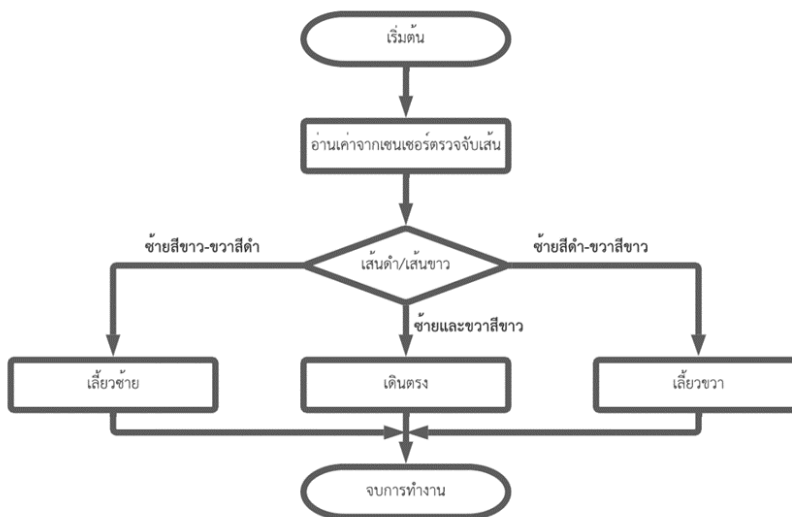
แผนผังการทำงานของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

ในหัวข้อนี้ผู้วิจัยได้อธิบายการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino โดยแสดงเป็นแผนผังการทำงาน ดังภาพที่ 4



ภาพที่ 4 แผนผังการทำงานของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino

จากภาพที่ 4 เริ่มต้นจากการกำหนดเป้าหมายให้หุ่นยนต์ จากนั้นโปรแกรมจะทำการคำนวณเส้นทางการเดินทางจากจุดเริ่มต้นไปยังตำแหน่งเป้าหมาย เมื่อพบสิ่งกีดขวางหุ่นยนต์จะเคลื่อนที่ด้วยการหลบ ซึ่งจะหลบซ้ายหรือหลบขวานั้นขึ้นอยู่กับตำแหน่งปัจจุบันของหุ่นยนต์และแผนที่เส้นทาง โดยโปรแกรมจะทำการคำนวณเส้นทางจากตำแหน่งปัจจุบันไปยังตำแหน่งเป้าหมายใหม่ กรณีไม่พบสิ่งกีดขวางหุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งต่อไป ดังแสดงการทำงานตามภาพที่ 5 และเมื่อพบว่าตำแหน่งนั้นเป็นเป้าหมายหุ่นยนต์จะหยุดการเคลื่อนที่และจบการทำงาน



ภาพที่ 5 แผนผังการทำงานของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติให้อยู่ตรงตามตำแหน่ง

จากภาพที่ 5 หุ่นยนต์จะทำการอ่านค่าเซนเซอร์ตรวจจับเส้นทั้งด้านซ้ายและด้านขวา โดยถ้าเซนเซอร์ตรวจจับเส้นด้านซ้ายและขวามีค่าได้สีขาวจะให้หุ่นยนต์เดินตรงไป ถ้าเซนเซอร์ตรวจจับเส้นด้านซ้ายอ่านค่าได้สีขาวและเซนเซอร์ตรวจจับเส้นขวามีค่าได้สีดำจะให้หุ่นยนต์เลี้ยวซ้าย และถ้าเซนเซอร์ตรวจจับเส้นด้านซ้ายอ่านค่าได้สีดำและเซนเซอร์ตรวจจับเส้นขวามีค่าได้สีขาวจะให้หุ่นยนต์เลี้ยวขวา

3.2 การพัฒนาอัลกอริทึมที่ใช้สำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ

สำหรับการพัฒนาอัลกอริทึมที่ใช้สำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัตินั้น ผู้วิจัยใช้ 4 อัลกอริทึมที่นิยมนำมาใช้ค้นหาเส้นทาง คือ อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) โดยใช้ภาษา C สำหรับ Arduino พัฒนาอัลกอริทึมดังกล่าว ซึ่งชูพันธุ์ รัตนโกศา (2559) ได้อธิบายการทำงานของแต่ละอัลกอริทึมไว้ดังรายละเอียดต่อไปนี้

อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search)

อัลกอริทึมการค้นหาในแนวกว้างก่อนนี้สถานะทุกตัวที่อยู่ในระดับเดียวกันของต้นไม้หรือกราฟ จะถูกตรวจสอบก่อนสถานะที่อยู่ในระดับถัดไป วิธีการของการค้นหาแบบนี้ทำโดยเริ่มจากการสร้างสถานะลูกของสถานะเริ่มต้นก่อนแล้วตรวจสอบว่ามีสถานะใดที่เป็นสถานะเป้าหมายหรือไม่ ถ้าหากว่ามีก็ถือว่าการค้นหาเป็นอันสิ้นสุด ถ้าไม่มีก็จะสร้างสถานะลูกของสถานะเหล่านั้นต่อไปอีก และทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะพบสถานะเป้าหมายหรือจนไม่สามารถสร้างสถานะลูกใหม่ได้อีก ซึ่งอัลกอริทึมที่ใช้กับการจัดการเทคนิคการค้นหาประเภทนี้คือ เทคนิคแถวคอย (Queue)

อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search)

อัลกอริทึมการค้นหาเชิงลึกก่อนจะสร้างสถานะในแนวลึกทางด้านมุมซ้ายล่างก่อน ถ้าสถานะตามแนวดิ่งถูกสร้างหรือกระจายจนหมดและยังไม่ได้คำตอบ ก็จะไล่กลับขึ้นด้านบนเพื่อหาเส้นทางอื่นที่จะเป็นไปได้ ซึ่งอัลกอริทึมที่ใช้กับการจัดการเทคนิคการค้นหาประเภทนี้คือ เทคนิคสแต็ก (Stack)

อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search)

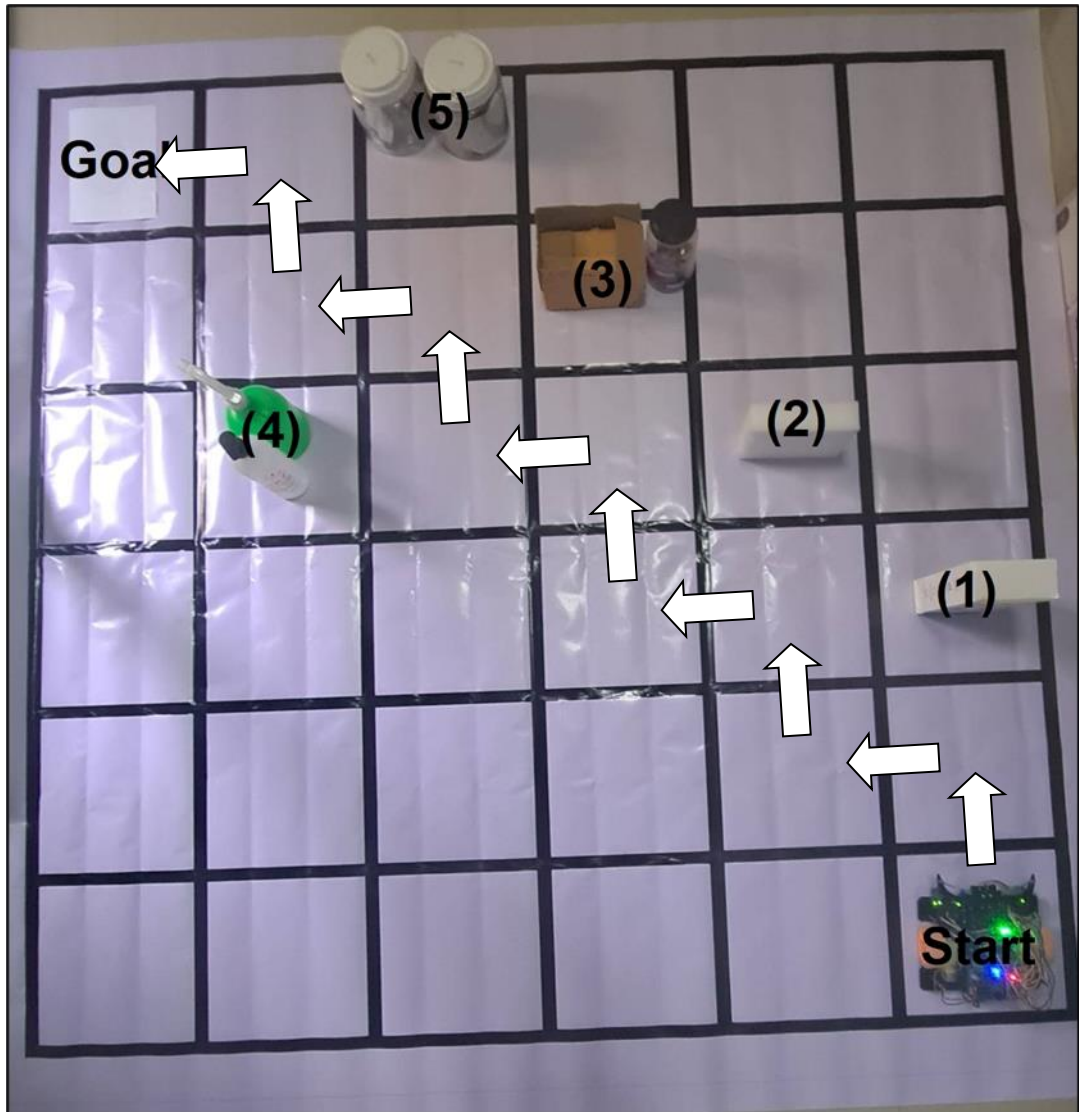
อัลกอริทึมการค้นหาแบบดีที่สุดก่อนจะใช้ฟังก์ชัน Evaluation เพื่อให้ค้นหาเป้าหมายได้เร็วขึ้น ซึ่งจะพิจารณาเฉพาะกับเส้นทางที่มองเห็นในขณะนั้นเท่านั้น และเลือกเส้นทางที่ใช้ทรัพยากรจากตำแหน่ง ณ สถานะนั้นจนถึงเป้าหมายให้น้อยที่สุด ดังนั้นฟังก์ชัน Evaluation ที่ได้จะพิจารณาจากค่าฮิวริสติกเพียงอย่างเดียว โดยค่าฮิวริสติกนั้นใช้เพื่อบ่งบอกสถานะของปัญหาว่ามีโอกาสเข้าใกล้เป้าหมายมากน้อยเพียงใด หากค่าฮิวริสติกมีค่าที่ดี (กำหนดให้ค่าฮิวริสติกยิ่งน้อยยิ่งดี) แสดงว่าสถานะของปัญหานั้นใกล้ถึงเป้าหมายมากขึ้น ดังนั้นการค้นหาก็จะพิจารณาจากสถานะที่มีค่าฮิวริสติกน้อย จนกว่าจะพบสถานะเป้าหมายที่ต้องการ (ค่าฮิวริสติกเป็น 0) ซึ่งเส้นทางที่เดินทางที่ได้จากการค้นหาด้วยเทคนิคนี้จะไม่ได้เส้นทางที่ใช้ระยะทางที่สั้นที่สุด แต่ช่วยทำให้การค้นหาเส้นทางไปสู่เป้าหมายได้เร็วขึ้น เนื่องจากมีการใช้งานฟังก์ชันฮิวริสติกที่ช่วยแนะนำสถานะที่ควรสำรวจก่อน แต่อย่างไรก็ตามการค้นหาด้วยเทคนิคนี้อาจไม่ค้นพบเส้นทางไปสู่สถานะเป้าหมายได้ทุกครั้ง โดยมีโอกาสที่จะค้นหาเส้นทางที่ซ้ำซ้อนจนเกิดเป็นวงเวียนไม่มีที่สิ้นสุดได้

อัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search)

อัลกอริทึมการค้นหาแบบเอ-สตาร์เป็นวิธีการค้นหาที่ถูกพัฒนาขึ้นมาเพื่อแก้ไขปัญหาที่เกิดขึ้นในอัลกอริทึมการค้นหาแบบดีที่สุดก่อน โดยการนำข้อมูลเข้ามาพิจารณาเพิ่มเติม คือ ข้อมูลทรัพยากรที่ใช้ตั้งแต่ตำแหน่งเริ่มต้นจนถึงตำแหน่งที่กำลังพิจารณาขณะนั้น โดยหลักการค้นหาก็จะพิจารณาสถานะที่เป็นโหนดในแต่ละสถานะในขณะนั้น ว่าสถานะใดที่มีค่าฟังก์ชัน Evaluation ดีที่สุด และทำเช่นนี้จนกว่าจะพบเส้นทางไปยังสถานะเป้าหมายที่สั้นที่สุด ซึ่งอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) นั้นสามารถแก้ไขปัญหาวงเวียนไม่มีที่สิ้นสุดของอัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) ได้อีกด้วย

4. ผลการวิจัย

ในการวิจัยครั้งนี้ โดยผู้วิจัยได้ทดสอบการทำงานของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแผนที่การเดินทางเป็นตารางแบบบล็อก 6x6 ขนาด 10x10 นิ้ว โดยกำหนดเริ่มต้นของหุ่นยนต์ (Start) กำหนดเป้าหมายที่หุ่นยนต์ต้องเดินทางไปถึง (Goal) และสิ่งกีดขวาง โดยในแต่ละอัลกอริทึมจะทดสอบ 4 ครั้ง รวมทั้งหมด 16 ครั้ง เริ่มจากวางสิ่งกีดขวางจำนวน 2 จุด ไปจนถึงวางสิ่งกีดขวางจำนวน 5 จุด ตามหมายเลข (1), (2), (3), (4) และ (5) ซึ่ง และทุกครั้งที่ทดสอบจะอยู่ในสภาพแวดล้อมเดียวกัน ดังภาพที่ 6 ซึ่งมีรายละเอียดดังต่อไปนี้



ภาพที่ 6 กำหนดตำแหน่งเริ่มต้น ตำแหน่งเป้าหมาย ตำแหน่งสิ่งกีดขวางและเส้นทางการเดินทาง

จากภาพที่ 6 แสดงตำแหน่งเริ่มต้น ตำแหน่งเป้าหมาย ตำแหน่งสิ่งกีดขวางและเส้นทางการเดินทางของหุ่นยนต์กรณีมีสิ่งกีดขวางจำนวน 5 จุด ซึ่งหุ่นยนต์ใช้ระยะทางในการเดินทางทั้งหมด 10 บล็อก ดังนี้

1. เดินตรงไปตามลูกศร เมื่อพบสิ่งกีดขวาง (1) เลี้ยวซ้าย
2. ตรงไป แล้วเลี้ยวขวา
3. ตรงไป พบสิ่งกีดขวาง (2) เลี้ยวซ้าย
4. ตรงไป แล้วเลี้ยวขวา
5. ตรงไป พบสิ่งกีดขวาง (3) เลี้ยวซ้าย
6. ตรงไป พบสิ่งกีดขวาง (4) เลี้ยวขวา
7. ตรงไป พบสิ่งกีดขวาง (5) เลี้ยวซ้าย
8. ตรงไป เลี้ยวขวา
9. ตรงไป เลี้ยวซ้าย
10. ตรงไป พบเป้าหมาย หยุดการเดินทาง

ในกรณีที่มีสิ่งกีดขวางจำนวน 2 จุด (จุดที่ 1 และจุดที่ 2) เมื่อพบสิ่งกีดขวางในจุดที่ 2 จะเลี้ยวซ้ายตรงไปเลี้ยวขวาจากนั้นจะตรงไปจนบล็อกสุดท้ายของสนามแล้วเลี้ยวซ้ายและตรงไปจนพบเป้าหมาย นับระยะทางได้ทั้งหมดจำนวน 10 บล็อก เช่นกัน ซึ่งกรณีมีสิ่งกีดขวางจำนวน 3 และ 4 จุดก็เช่นเดียวกัน

สรุปผลการทดสอบการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติ

ในงานวิจัยนี้ผู้วิจัยได้ทำการทดสอบการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติ โดยทำการเปรียบเทียบแต่ละวิธีในด้านเวลาและระยะทางที่ใช้ในการเคลื่อนที่จากจุดเริ่มต้นไปยังเป้าหมาย ซึ่งงานวิจัยนี้ได้ใช้ 4 อัลกอริทึมมาเปรียบเทียบกัน ได้แก่ อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) ดังแสดงผลการทดสอบตามตารางที่ 1 และตารางที่ 2 อีกทั้งผู้วิจัยได้นำข้อมูลในตารางมาสร้างเป็นแผนภูมิเพื่อแสดงการเปรียบเทียบการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติในด้านของเวลา ดังภาพที่ 7 และในด้านของการสำรวจเส้นทางเพื่อตัดสินใจในการเคลื่อนที่ของหุ่นยนต์ ดังภาพที่ 8

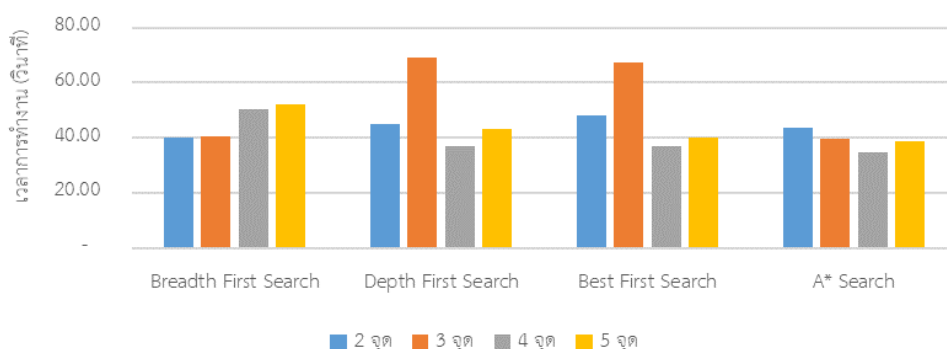
ตารางที่ 1 เปรียบเทียบเวลาการทำงานของแต่ละอัลกอริทึม (หน่วย:วินาที)

สิ่งกีดขวาง	Breadth First Search	Depth First Search	Best First Search	A* Search
2 จุด	39.89	45.12	48.02	43.65
3 จุด	40.71	69.18	67.33	39.54
4 จุด	50.44	36.92	36.85	34.55
5 จุด	52.15	43.34	40.10	38.75
เฉลี่ย	45.80	48.64	48.08	39.12

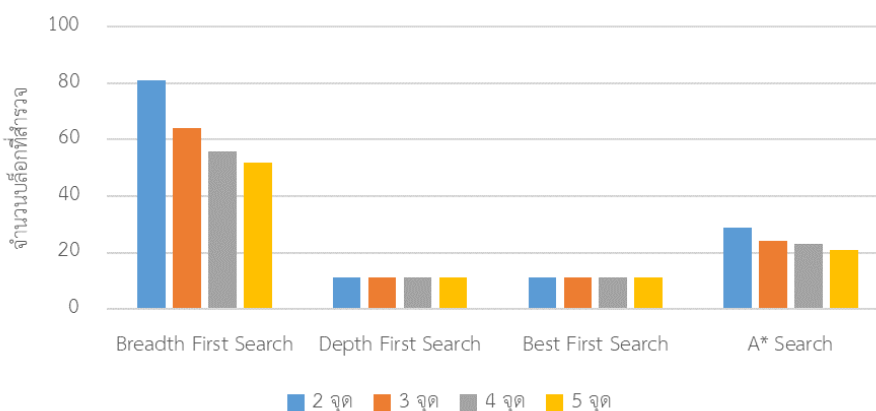
ตารางที่ 2 เปรียบเทียบระยะทางการเดินทางของหุ่นยนต์ในแต่ละอัลกอริทึม (หน่วย:บล็อก)

สิ่งกีดขวาง	Breadth First Search	Depth First Search	Best First Search	A* Search
2 จุด	10	10	10	10
3 จุด	10	10	10	10
4 จุด	10	10	10	10
5 จุด	10	10	10	10

จากตารางที่ 1 และตารางที่ 2 จะเห็นว่าผลการทดสอบหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino อัลกอริทึมที่ใช้เวลาในการเดินทางเหลือน้อยที่สุด คือ อัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) ใช้เวลาเฉลี่ยเท่ากับ 39.12 วินาที ส่วนอัลกอริทึมที่ใช้เวลาในการเดินทางเฉลี่ยมากที่สุด คือ อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) ใช้เวลาเฉลี่ยเท่ากับ 48.64 วินาที ซึ่งทุกอัลกอริทึมใช้ระยะทางในการเดินทางเท่ากัน (10 บล็อก)



ภาพที่ 7 แผนภูมิแสดงการเปรียบเทียบการเคลื่อนที่ของหุ่นยนต์ในด้านของเวลา



ภาพที่ 8 แผนภูมิแสดงการเปรียบเทียบการเคลื่อนที่ของหุ่นยนต์ในด้านการสำรวจเส้นทาง

5. อภิปรายผลการวิจัยและข้อเสนอแนะ

งานวิจัยการศึกษาวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino ผู้วิจัยได้ดำเนินงานวิจัยดังกล่าวเพื่อให้เกิดประโยชน์ผู้ที่สนใจเกี่ยวกับการนำเทคโนโลยีมาช่วยในการหาขนส่งสินค้า หรือ วางแผนการเดินทาง โดยแบ่งการอภิปรายผลการวิจัยออกเป็น 3 ส่วนตามวัตถุประสงค์ของการวิจัย คือ

1. ศึกษาวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ โดยผู้วิจัยได้ศึกษาวิธีการเคลื่อนที่ของหุ่นยนต์แบบอัตโนมัติจากงานวิจัยพบว่ามีการใช้อัลกอริทึมสำหรับการวางแผนการเดินทาง การขนส่งสินค้า การค้นหาเส้นทางในเขาวงกต การค้นหาเส้นทางในเกมส์ แม้กระทั่งการค้นหาไฟล์บนระบบปฏิบัติการวินโดวส์ ซึ่งการวิจัยครั้งนี้ผู้วิจัยใช้ 4 อัลกอริทึมที่นิยมนำมาใช้งานค้นหาเส้นทาง ได้แก่ อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) มาศึกษาวิธีการเดินทางของหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ ซึ่งพบว่าผลการศึกษาพบว่าเป็นไปตามวัตถุประสงค์ของงานวิจัยดังรายละเอียดที่จะกล่าวในหัวข้อถัดไป รวมถึงพบว่ามีประสิทธิภาพครั้งนี้มีข้อดีทำให้สามารถนำอัลกอริทึมที่มีประสิทธิภาพดีที่สุดในยุคนี้ใช้กับงานวิจัยที่เกี่ยวข้องกับการค้นหาเส้นทางได้เป็นอย่างดี

2. ออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino โดยผู้วิจัยใช้แพลตฟอร์ม Arduino ประกอบไปด้วย บอร์ดอิเล็กทรอนิกส์ตระกูล Arduino รุ่น ESP32 ในการประมวลผลคำสั่งต่าง ๆ โมดูล HC-SR04 อัลตราโซนิกเป็นอุปกรณ์วัดระยะทางเพื่อตรวจจับสิ่งกีดขวาง IR Infrared เป็นอุปกรณ์ตรวจจับเส้นทางในการเคลื่อนที่ของหุ่นยนต์ และโมดูล L298N เป็นตัวขับเคลื่อนหุ่นยนต์ทำให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งต่าง ๆ ได้ และใช้ภาษา C สำหรับ Arduino ในการเขียนโปรแกรมตามอัลกอริทึมการค้นหาเส้นทางเดินทางเพื่อให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเป้าหมายได้ ซึ่งจากการทดสอบพบว่าหุ่นยนต์สามารถเคลื่อนที่แบบอัตโนมัติไปยังตำแหน่งเป้าหมายและสามารถหลบสิ่งกีดขวางได้อย่างถูกต้อง

3. เปรียบเทียบประสิทธิภาพและหาวิธีที่เหมาะสมสำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ โดยผู้วิจัยได้พัฒนาอัลกอริทึมที่ใช้สำหรับหุ่นยนต์เคลื่อนที่แบบอัตโนมัติ 4 อัลกอริทึม คือ อัลกอริทึมการค้นหาในแนวกว้างก่อน (Breadth First Search) อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) อัลกอริทึมการค้นหาแบบดีที่สุดก่อน (Best First Search) และอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) จากการทดสอบการทำงานในส่วนนี้พบว่าอัลกอริทึมที่ใช้เวลาในการเดินทางเฉลี่ยน้อยที่สุด คือ อัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) ใช้เวลาเฉลี่ยเท่ากับ 39.12 วินาที ส่วนอัลกอริทึมที่ใช้เวลาในการเดินทางเฉลี่ยมากที่สุด คือ อัลกอริทึมการค้นหาเชิงลึกก่อน (Depth First Search) ใช้เวลาเฉลี่ยเท่ากับ 48.64 วินาที ซึ่งทุกอัลกอริทึมใช้ระยะทางในการเดินทางเท่ากัน คือ 10 บล็อก ซึ่งสอดคล้องกับงานวิจัยของ Hii et al. (2021). ที่พบว่าอัลกอริทึมการค้นหาแบบเอ-สตาร์ (A* Search) ใช้เวลาน้อยที่สุดและเดินทางสั้นที่สุด ทั้งนี้เนื่องจากอัลกอริทึม

เอ-สตาร์ (A* Search) เป็นอัลกอริทึมที่ถูกปรับปรุงให้มีการทำงานที่มีประสิทธิภาพและนิยมใช้กับการค้นหาเส้นทางระหว่างจุด (Hart et al., 1968)

แต่อย่างไรก็ตามยังพบว่ามีประเด็นสำคัญ 2 ส่วน ที่อาจส่งผลกระทบต่อการศึกษาในครั้งนี้ คือ

1. การจัดวางสนามต้องให้สนามราบเรียบ ไม่มีรอยพับ หากมีรอยพับมากหุ่นยนต์จะไม่สามารถเคลื่อนที่ไปได้ หรือ เคลื่อนที่ไม่ตรงเส้นทาง ซึ่งจะทำให้การเดินทางของหุ่นยนต์ไม่ตรงตำแหน่งเป้าหมาย
2. เนื่องจากหุ่นยนต์ใช้ IR Infrared เป็นอุปกรณ์ตรวจจับเส้นทางในการเคลื่อนที่ของหุ่นยนต์ หากแสงสว่างไม่คงที่ จะทำให้หุ่นยนต์ไม่สามารถจับเส้นทางได้อย่างถูกต้องเป็นผลให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเป้าหมายไม่ถูกต้อง ดังนั้นควรใช้สถานที่วางสนามทดสอบที่มีแสงสว่างคงที่ตลอดการทดสอบการทำงานเพื่อให้หุ่นยนต์สามารถเคลื่อนที่ไปยังตำแหน่งเป้าหมายได้อย่างถูกต้อง

แม้ว่าการศึกษาวิธีที่เหมาะสมสำหรับพัฒนาหุ่นยนต์เคลื่อนที่แบบอัตโนมัติด้วยแพลตฟอร์ม Arduino จะได้ผลลัพธ์ตามวัตถุประสงค์ที่ตั้งไว้แล้วก็ตาม แต่การทดสอบใช้สนามขนาดเล็กเท่านั้น ซึ่งในอนาคตผู้วิจัยจะทำการทดสอบในสนามที่มีขนาดแตกต่างและมีความหลากหลายมากกว่านี้ รวมถึงการออกแบบภารกิจให้มีความแตกต่างกันออกไป เพื่อทำให้ได้วิธีที่เหมาะสมมากที่สุดและทำให้ผลการทดสอบมีความแม่นยำมากขึ้น อีกทั้งอาจใช้ระบบปฏิบัติการหุ่นยนต์ (ROS: Robot Operating System) มาประมวลผลแทนการใช้ภาษา C สำหรับ Arduino ซึ่งจะทำให้หุ่นยนต์เคลื่อนที่ได้อย่างมีประสิทธิภาพยิ่งขึ้น

6. กิตติกรรมประกาศ

ขอขอบคุณ คณะวิทยาศาสตร์และเทคโนโลยีมหาวิทยาลัยราชภัฏนครสวรรค์ที่สนับสนุนทุนวิจัยในปีงบประมาณ 2566

7. เอกสารอ้างอิง

- ชูพันธุ์ รัตนโกคา. (2559). *เอกสารคำสอน วิชา ความรู้เบื้องต้นทางปัญญาประดิษฐ์*. ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- น้ำฝน พาพันธ์ และ ภัทธนาธิษฐ์ แก้วประดิษฐ์. (2563). *การจัดเส้นทางการขนส่งสินค้าโดยวิธีอัลกอริทึมแบบกรณีศึกษา : โรงงานเม็ดพลาสติก [โครงการวิศวกรรม บัณฑิต]*. มหาวิทยาลัยธุรกิจบัณฑิต.
- Alli, K. S., Onibonoje, M. O., Oluwole, A., Ogunlade, M. A., Mmonyi, A. C., Ayamolowo, O., & Dada, S. O. (2018). Development of an Arduino-based Obstacle Avoidance Robotic System for an Unmanned Vehicle. *Journal of Engineering and Applied Science*, 13(3), 886-892.
- Ginting, S. E., & Sembiring, A. S. (2019). Comparison of Breadth First Search (BFS) and Depth-First Search (DFS) Methods on File Search in Structure Directory Windows. *Jurnal Teknologi Komputer*, 13(1), 26-31.

- Hart, P. E., Nilsson, N. J., & Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In IEEE, *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107. <https://doi.org/10.1109/TSSC.1968.300136>
- Hii, J. Y., Lee, J., & Chuah, Y. (2021). Optimization and Simulation of a Navigation Robot in Mazes. *International Journal of Engineering Research & Technology*, 10(5), 222-227. <https://doi.org/10.17577/IJERTV10IS050110>
- Irawan, Y., Yudie, M., Ordila, R., & Diandra, R. (2021). Automatic Floor Cleaning Robot Using Arduino and Ultrasonic Sensor. *Journal of Robotics and Control (JRC)*. 2(4), 240-243. <https://doi.org/10.18196/jrc.2485>
- Lim, T.H., & Ng, P. L. (2021). Evaluating Recursive Backtracking Depth-First Search Algorithm in Unknown Search Space for Self-learning Path Finding Robot. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 356, 531-543. https://doi.org/10.1007/978-3-030-69066-3_47
- Navare, D. S., Kapde, Y. R., Maurya S., Pardeshi, D. B., & William, P. (2022). Robotic Bomb Detection and Disposal: Application using Arduino. In IEEE, *International Conference on Communication and Electronics Systems (ICCES 2022)*. (pp. 479–483). <https://doi.org/10.1109/ICCES54183.2022.9836011>
- Pavithra, A. C., & Subramanya, G. V. (2018). Obstacle Avoidance Robot Using Arduino. *International Journal of Engineering Research & Technology*, 6(13), 1-4.