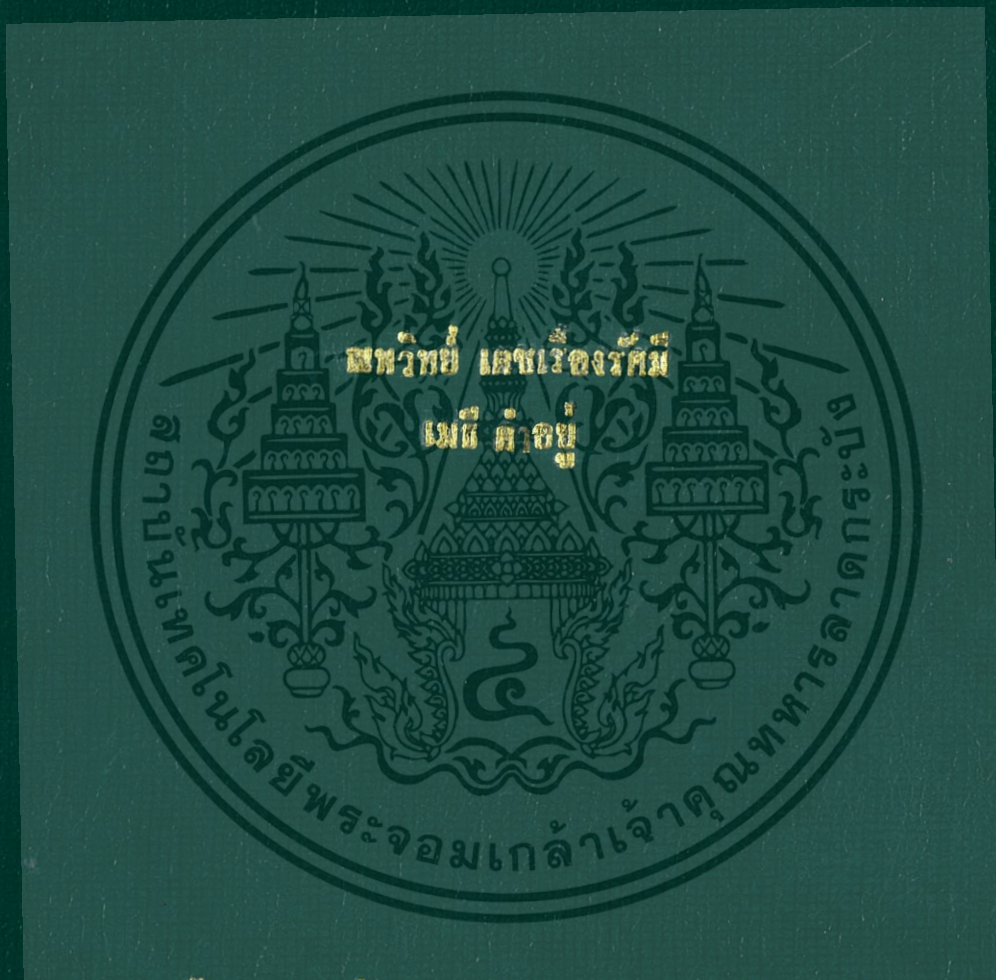


การค้นหเส้นทางอย่างมีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่  
DYNAMIC PATH PLANNING FOR MOBILE ROBOT NETWORKS



ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาคณะศึกษาศาสตร์ ศึกษาศาสตร์บัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ ๒ ปีการศึกษา ๒๕๕๘

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การค้นหาเส้นทางอย่างมีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่

DYNAMIC PATH PLANNING FOR MOBILE ROBOT NETWORKS



T146202

โดย

ณพิวิทย์ เตชเรืองรัมย์

NOPPAWIT TECHRUENGRASSAMEE

เมธี คำอยู่

MATEE KUMYOU

อาจารย์ที่ปรึกษา

ดร. ลภัส ประดิษฐ์ทัศนีย์

เลขทะเบียน 146202  
รับเดือนปี 25 ๒๕๖0

b. 12๙405๗๙  
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องสมุดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DYNAMIC PATH PLANNING FOR MOBILE ROBOT NETWORKS

**NOPPAWIT TECHRUENGRASSAMEE**

**MATEE KUMYOU**



**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2016**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเป็นงานลิขสิทธิ์ภายใต้เงื่อนไขของผู้จัดทำเอกสารฉบับนี้โดยไม่มีการรับประกันว่าเอกสารฉบับนี้จะไม่มีการแก้ไขหรือเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2558  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การค้นหาเส้นทางอย่างมีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่  
DYNAMIC PATH PLANNING FOR MOBILE ROBOT NETWORKS

ผู้จัดทำ

- |               |               |              |          |
|---------------|---------------|--------------|----------|
| 1. นายณพวิทย์ | เดชเรืองรัมย์ | รหัสนักศึกษา | 55070034 |
| 2. นายเมธี    | คำอยู่        | รหัสนักศึกษา | 55070092 |

.....อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ (ดร.ลภัส ประดิษฐ์ทัศนีย์) นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	การค้นหาเส้นทางอย่างมีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่		
นักศึกษา	นายณพวิทย์	เดชเรืองรัมย์	รหัสนักศึกษา 55070034
	นายเมธี	คำอยู่	รหัสนักศึกษา 55070092
ปริญญา	วิทยาศาสตรบัณฑิต		
สาขาวิชา	เทคโนโลยีสารสนเทศ		
ปีการศึกษา	2558		
อาจารย์ที่ปรึกษา	ดร. ลภัส ประดิษฐ์ทัศนีย์		

### บทคัดย่อ

ปฏิญานិพนธ์ฉบับนี้นำเสนอการพัฒนาอัลกอริทึมที่ใช้ในการหาเส้นทางของหุ่นยนต์รวมไปถึงการสื่อสารระหว่างหุ่นยนต์เพื่อประเมินสถานการณ์ในขณะนั้น ในการสำรวจเส้นทางของหุ่นยนต์นั้นจะใช้เซ็นเซอร์ในการตรวจจับสภาพแวดล้อมรอบตัวโดยประเมินจากสิ่งที่อยู่ข้างหน้าเพื่อนำมาประมวลผลและการหาเส้นทางที่ใช้ในการสำรวจในสภาพแวดล้อมที่ต้องการ และส่งข้อมูลสถานการณ์ปัจจุบันให้กับหุ่นยนต์ตัวอื่น เพื่อให้หุ่นยนต์ตัวอื่นใช้ประมวลผลต่อไป จากนั้นส่งข้อมูลต่อมายังหน่วยประมวลผลเพื่อดูสถานะของหุ่นยนต์แต่ละตัว ในส่วนของอัลกอริทึมจะตัดแปลงมาจากอัลกอริทึมที่มีอยู่เพื่อให้เหมาะกับสถานการณ์

<b>Project Title</b>	Dynamic Path Planning for Mobile Robot Networks		
<b>Student</b>	Mr. Noppawait	Techruengrassamee	Student ID 55070034
	Mr. Matee	Kumyou	Student ID 55070092
<b>Degree</b>	Bachelor of Science		
<b>Program</b>	Information Technology		
<b>Academic Year</b>	2016		
<b>Advisor</b>	Dr. Lapas Pradittasnee		

## ABSTRACT

This thesis will the develop algorithm of routing path protocol and the communication between robot. In case of that robot will use ultrasonic sensor for detect the obstacle in front of robot and use the ultrasonic value to simulate to environment around the robot. After that robot will send the value to the server. Another robot will use value from the server and send the robot status to the server. The algorithm is adapted from an existing algorithm to suit the situation.

## กิตติกรรมประกาศ

กิตติกรรมประกาศ ขอขอบคุณดร. ฤกษ์ ประดิษฐ์ทัศนีย์ที่เป็นผู้คิดค้นให้ความรู้เกี่ยวกับ  
โครงการนี้ ทำให้กลุ่มของเราได้มีโอกาสศึกษาการทำงานของหุ่นยนต์รวมไปถึงการพัฒนา  
อัลกอริทึมที่ใช้ในการสำรวจเส้นทางของหุ่นยนต์ อีกทั้งยังให้ความช่วยเหลือในด้านต่าง ๆ

ณพิวิทย์ เศษเรืองรัมย์

เมธิ คำอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญรูป .....	VI
บทที่ 1 บทนำ .....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของงาน.....	2
1.4 แผนการดำเนินงาน .....	2
1.5 ขั้นตอนของการศึกษา.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....	4
2.1 องค์กรประกอบฮาร์ดแวร์ .....	4
2.1.1 Lego Mindstorms NXT ver.2.....	4
2.1.2 Lego Mindstorms NXT.....	5
2.1.3 โมดูลต่างๆ .....	5
2.2 องค์กรประกอบด้านซอฟต์แวร์ .....	8
2.2.1 ภาษา Java.....	8
2.2.1.1 Lejos NXT Library .....	8
2.2.1.2 Lejos PC Library .....	9
2.2.1.3 Java Swing GUI.....	10
2.3 กระบวนการค้นหาเส้นทางของหุ่นยนต์เคลื่อนที่ .....	11
2.3.1 การสำรวจพื้นที่แบบ Offline.....	14
2.3.2 การสำรวจพื้นที่แบบ Online.....	14
2.3.3 การสำรวจพื้นที่แบบ Spanning Tree โดยใช้ Spiral-STC.....	14
2.3.4 การนำอัลกอริทึม Spiral-STC มาประยุกต์ใช้ .....	16

# สารบัญ (ต่อ)

หน้า

บทที่ 3 การวิเคราะห์และออกแบบระบบ .....	17
3.1 การวิเคราะห์การทำงานพื้นฐานของหุ่นยนต์และเซิร์ฟเวอร์ .....	17
3.1.1 การทดสอบและออกแบบคำสั่งการทำงานของหุ่นยนต์ .....	17
3.1.2 การออกแบบฝั่งเซิร์ฟเวอร์ .....	18
3.2 การวิเคราะห์และการ Implement อัลกอริทึม Spiral-STC(Spanning Tree Coverage) .	19
3.2.1 ข้อจำกัดในการนำอัลกอริทึม Spiral-STC มาใช้ในหุ่นยนต์ LEGO .....	19
3.2.2 แนวทางการแก้ไขปัญหา .....	19
3.2.3 การนำอัลกอริทึม Spiral-STC มาประยุกต์ใช้ .....	21
3.2.3.1 อัลกอริทึมการเก็บพิกัดของหุ่นยนต์ .....	21
3.2.3.2 อัลกอริทึมการตรวจสอบการเดินทางของหุ่นยนต์ .....	23
3.2.3.3 อัลกอริทึมการเดินทางสำรวจไปของหุ่นยนต์ .....	24
3.2.3.4 อัลกอริทึมการเดินทางสำรวจกลับของหุ่นยนต์ .....	26
3.3 การจำลองแผนที่จากเซ็นเซอร์อัลตราโซนิกส์ .....	27
3.4 การบันทึกข้อมูลที่ได้จากหุ่นยนต์ .....	27
3.5 การนำข้อมูลที่ได้จากหุ่นยนต์มาวิเคราะห์และบันทึกเป็น Log file .....	29
บทที่ 4 การทำงานของระบบ .....	30
4.1 การสำรวจพื้นที่ของหุ่นยนต์ .....	30
4.1.1 การเคลื่อนที่ของหุ่นยนต์ .....	30
4.1.2 การเลี้ยวของหุ่นยนต์เมื่อเจอสิ่งกีดขวาง .....	30
4.2 การส่งค่าจากเซ็นเซอร์กลับมาที่เครื่องเซิร์ฟเวอร์ .....	31
4.3 การออกแบบพื้นที่สำรวจ .....	31
บทที่ 5 สรุปผลโครงการ .....	32
5.1 สรุปผลโครงการ .....	32
บรรณานุกรม .....	35

# สารบัญรูป

รูปที่	หน้า
2.1 หุ่นยนต์ Lego Mindstorms NXT ver.2.....	4
2.2 ชุดประกอบ Lego Mindstorms NXT .....	5
2.3 NXT intelligent brick controller .....	6
2.4 มอเตอร์ขับเคลื่อน (Interactive servo motor).....	6
2.5 ตัวตรวจจับวัตถุ (Ultrasonic sensor).....	7
2.6 ตัวตรวจจับค่าแสง (Light Sensor) .....	7
2.7 ภาษา Java.....	8
2.8 การติดตั้งปลั๊กอิน LeJOS NXT Support.....	9
2.9 การสร้างโปรเจกต์ LeJOS NXT และ LeJOS PC.....	10
2.10 การใช้ Java swing GUI .....	11
2.11 การจำลองห้องที่มีสิ่งกีดขวาง.....	12
2.12 การเคลื่อนที่ของหุ่นยนต์.....	13
2.13 การเคลื่อนที่ผ่านสิ่งกีดขวางแบบ a และ b.....	13
2.14 การเลือกเส้นทาง .....	14
2.15 รูปประกอบการทำงานของอัลกอริทึม STC .....	15
2.16 แบบจำลองการเคลื่อนที่ของหุ่นยนต์.....	16
3.1 การใช้ Java swing GUI ในการจำลองแผนที่.....	18
3.2 แสดงทิศทางของหุ่นยนต์.....	19
3.3 Flowchart การหาทิศทางของหุ่นยนต์.....	20
3.4 การเก็บพิกัดของหุ่นยนต์.....	21
3.5 Flowchart การเก็บพิกัดของหุ่นยนต์.....	22
3.6 Flowchart การตรวจสอบการเดินซ้ำของหุ่นยนต์.....	23
3.7 การเลือกเส้นทางสำรวจของหุ่นยนต์.....	24
3.8 Flowchart การเดินสำรวจออกไปของหุ่นยนต์.....	25
3.9 Flowchart การเดินสำรวจกลับของหุ่นยนต์.....	26
3.10 การจำลองแผนที่จากเซ็นเซอร์อัลตราโซนิก.....	27
3.11 การบันทึกข้อมูลที่ได้จากหุ่นยนต์.....	28
3.12 การนำข้อมูลมาวิเคราะห์.....	29

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.13 การบันทึกเป็น Log file .....	29
4.1 การเคลื่อนที่สำรวจของหุ่นยนต์ .....	30
4.2 การจำลองพื้นที่จากการเคลื่อนที่ของหุ่นยนต์.....	31
5.1 สภาพแวดล้อมที่หุ่นยนต์เข้าไปสำรวจ .....	32
5.2 แบบจำลองพื้นที่ที่หุ่นยนต์เข้าไปสำรวจ.....	33
5.3 Log File.....	34



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

หุ่นยนต์เป็นสิ่งที่มนุษย์คิดค้นและสร้างขึ้นเพื่ออำนวยความสะดวกให้กับมนุษย์และใช้แทนแรงงานมนุษย์ หรือใช้ทำงานที่มนุษย์ไม่สามารถทำได้ เช่นการสำรวจสภาพแวดล้อมที่มนุษย์ไม่สามารถเข้าไปได้จึงจำเป็นต้องใช้หุ่นยนต์เข้าไปทำการสำรวจพื้นที่แทน โดยมนุษย์จะเป็นคนเขียนคำสั่งต่าง ๆ เพื่อให้หุ่นยนต์สามารถทำงานได้ตามที่ต้องการในปัจจุบันได้มีการพัฒนาหุ่นยนต์ในรูปแบบต่าง ๆ ขึ้นมามากมาย เพื่อให้เหมาะกับงานในรูปแบบต่างกันไป โดยทั่วไปแล้วหุ่นยนต์จะแบ่งได้เป็น 2 องค์ประกอบหลัก โดยองค์ประกอบแรกคือส่วนกายภาพที่เป็นชิ้นส่วนซึ่งนำมาประกอบเป็นตัวหุ่นยนต์เพื่อให้เหมาะกับงานที่หุ่นยนต์ต้องทำ และองค์ประกอบที่สองคือส่วนของโปรแกรมหรือชุดคำสั่งที่ใช้ในการสั่งงานหุ่นยนต์ ซึ่งองค์ประกอบหลังนี้มีความสำคัญมากต่อประสิทธิภาพในการทำงานของหุ่นยนต์ เพราะ อัลกอริทึมที่ทำการ โปรแกรมเข้าไปจะทำหน้าที่เสมือนสมองที่สั่งให้หุ่นยนต์ทำงานตามเป้าหมายที่มนุษย์ต้องการ และงานแต่ละประเภทก็ต้องการอัลกอริทึมที่แตกต่างกัน เมื่อหุ่นยนต์สามารถทำงานได้ตามอัลกอริทึมที่เราโปรแกรมเข้าไป หุ่นยนต์จะสามารถค้นหาเส้นทางที่ใช้ในการสำรวจได้ และอัลกอริทึมที่ใช้ยังเป็นสิ่งที่สามารถแก้ไขปัญหารวมไปถึงข้อจำกัดต่าง ๆ ในการสำรวจพื้นที่เช่น แสง สัญญาณ GPS เมื่อสามารถแก้ปัญหาเหล่านี้ได้ หุ่นยนต์จึงสามารถทำงานได้อย่างมีประสิทธิภาพ การที่หุ่นยนต์สามารถทำงานได้อย่างมีประสิทธิภาพนั้น จะทำให้เราสามารถลดระยะเวลาในการสำรวจ ประหยัดพลังงาน รวมไปถึงความถูกต้องที่ได้จากการสำรวจ ทำให้เราสามารถจำลองสภาพพื้นที่ที่หุ่นยนต์เข้าไปสำรวจได้สมบูรณ์ และยังสามารถนำแบบจำลองพื้นที่นี้ไปประโยชน์ในการสำรวจพื้นที่แบบต่าง ๆ ได้อีกด้วย

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- มีความเข้าใจในรูปแบบในการค้นหาเส้นทางของหุ่น ในแบบต่างๆ ที่มีอยู่ในปัจจุบัน
- ศึกษาการทำงานของโปรแกรมที่ใช้ในการจำลองการทำงานของรถค้นหาเส้นทางของหุ่นยนต์
- วิเคราะห์ความต้องการสำหรับการค้นหาเส้นทางสำหรับการใช้หุ่นยนต์ในการสำรวจบริเวณที่กำหนดและสามารถพัฒนาระบบค้นหาเส้นทางที่เหมาะสมสำหรับการสำรวจพื้นที่ของห้อง 1 ห้องได้

• เพื่อให้หุ่นยนต์มีการเคลื่อนไหวอย่างมีประสิทธิภาพ เพื่อเป็นประโยชน์ในอนาคต

### 1.3 ขอบเขตของงาน

- หุ่นยนต์สามารถสำรวจพื้นที่ได้ว่าพื้นที่นั้นๆ มีสิ่งกีดขวางอะไรบ้าง
- หุ่นยนต์สามารถส่งข้อมูลเส้นทาง หรือตำแหน่งของสิ่งกีดขวางกลับมายังหน่วย

ประมวลผลได้

- หุ่นยนต์สามารถสื่อสารถึงกันได้เมื่อมีหุ่นยนต์มากกว่า 2 ขึ้นไป เมื่อหุ่นยนต์ตัวใดตัวหนึ่งเคลื่อนที่เกินระยะที่จะสื่อสารกับ หน่วยประมวลผล

### 1.4 แผนการดำเนินงาน

#### ภาคเรียนที่ 1

- ศึกษา อัลกอริทึม ของหุ่นยนต์สำรวจเดิมที่มีอยู่แล้วในปัจจุบัน
- ศึกษาการทำงานของหุ่นยนต์ที่มีว่าสามารถทำอะไรได้บ้าง
- ศึกษาการรับค่าจากเซ็นเซอร์ต่างๆ ของหุ่นยนต์
- พัฒนาให้หุ่นยนต์สามารถเคลื่อนไหวได้ตามเส้นทางที่กำหนด
- พัฒนาให้หุ่นยนต์สามารถส่งค่าที่ได้รับจากเซ็นเซอร์เมื่อเจอสิ่งกีดขวางกลับมาให้เซิร์ฟเวอร์ได้
- จำลองสภาพแวดล้อมของพื้นที่ที่หุ่นยนต์เข้าไปสำรวจ

#### ภาคเรียนที่ 2

- พัฒนาหุ่นยนต์สามารถสำรวจค้นหาเส้นทางภายในพื้นที่ที่กำหนดได้ โดยที่หุ่นยนต์เดินจากจุดเริ่มต้น ไปจนสุดทางที่เป็นไปได้แล้วจึงเดินย้อนกลับมาที่จุดเริ่มต้นอีกครั้ง
- พัฒนาให้หุ่นยนต์สามารถส่งค่าที่ได้รับจากเซ็นเซอร์เมื่อเจอสิ่งกีดขวางกลับมายังเซิร์ฟเวอร์
- สามารถสั่งงานให้หุ่นตัวที่สองเดินไปยังจุดที่กำหนด โดยอาศัยข้อมูลเส้นทางที่เรียนรู้จากหุ่นตัวแรก

### 1.5 ขั้นตอนของการศึกษา

- ศึกษา ค้นคว้าและรวบรวมข้อมูลของหุ่นยนต์ เช่น อัลกอริทึมที่มีอยู่แล้ว
- ความสามารถทางด้านกายภาพของหุ่นยนต์ เซ็นเซอร์รับค่าต่าง ๆ ฯลฯ
- วิเคราะห์ข้อมูลที่ได้
  - ออกแบบและกำหนดขอบเขตความสามารถของหุ่นยนต์
  - พัฒนาหุ่นยนต์
  - ทดสอบการทำงานของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บันทึกผลและปรับปรุงแก้ไขปัญหา
- สรุปผลและจัดทำเอกสารประกอบโครงการ

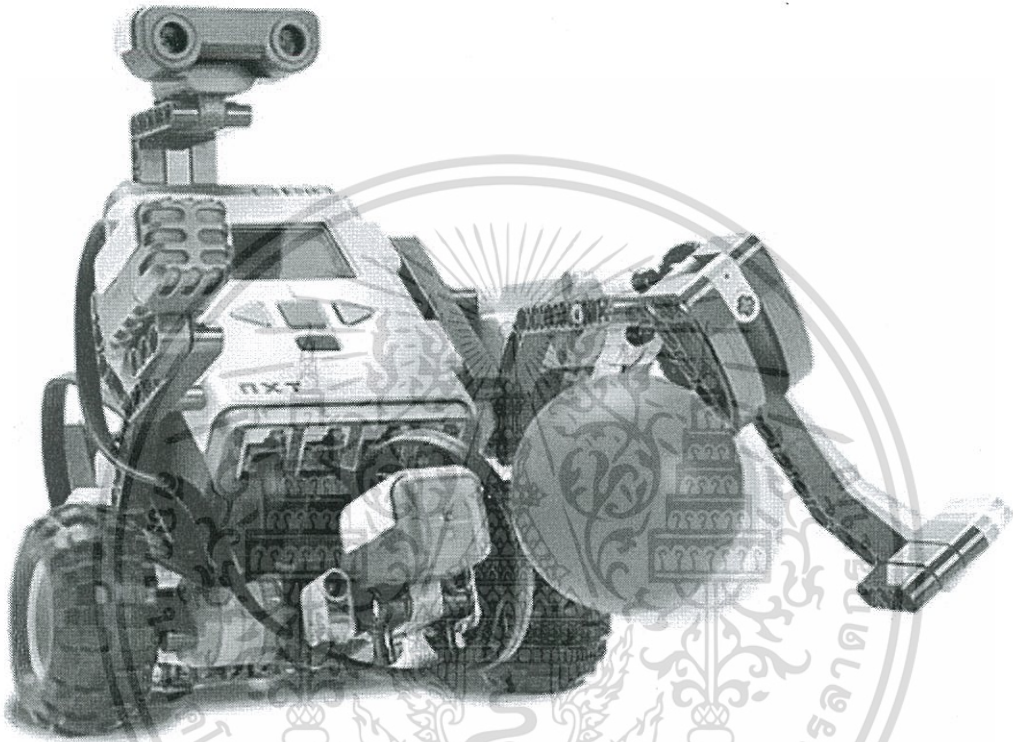


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 องค์ประกอบฮาร์ดแวร์

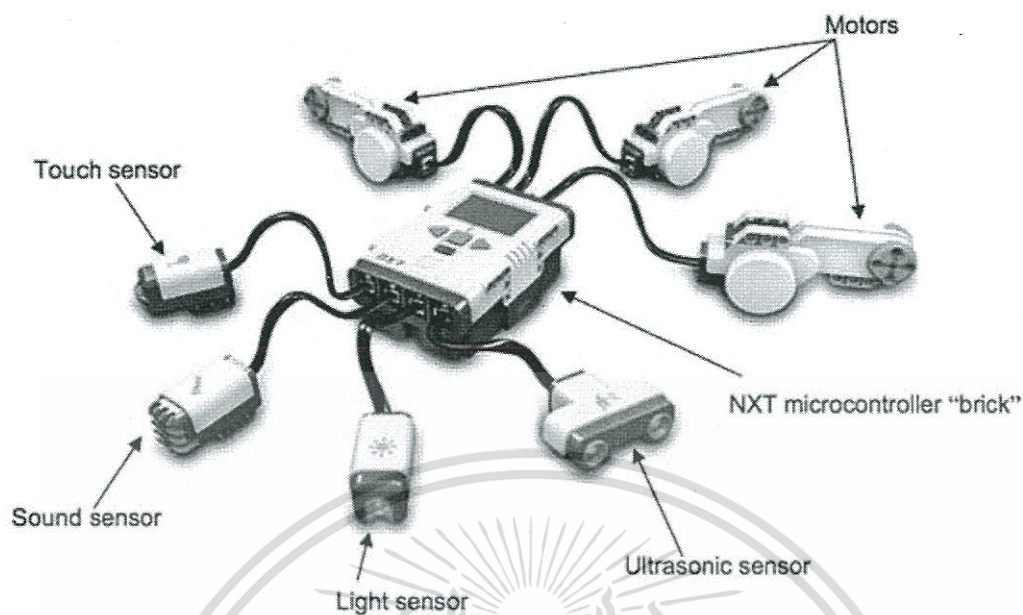
#### 2.1.1 Lego Mindstorms NXT ver.2



รูปที่ 2.1 หุ่นยนต์ Lego Mindstorms NXT ver.2  
(ที่มา: <http://krunisit.rwb.ac.th/nxt.html>)

Lego Mindstorms NXT ver.2 เป็นหุ่นยนต์ของ Lego ที่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานได้โดยโปรแกรมจะเก็บไว้ในส่วนประมวลผลของหุ่นยนต์ จากนั้นระบบจะส่งการต่อไปโมดูลต่างๆ ตามคำสั่งที่เราเขียนลงไป

## 2.1.2 LEGO MINDSTORMS NXT



รูปที่ 2.2 ชุดประกอบ Lego Mindstorms NXT

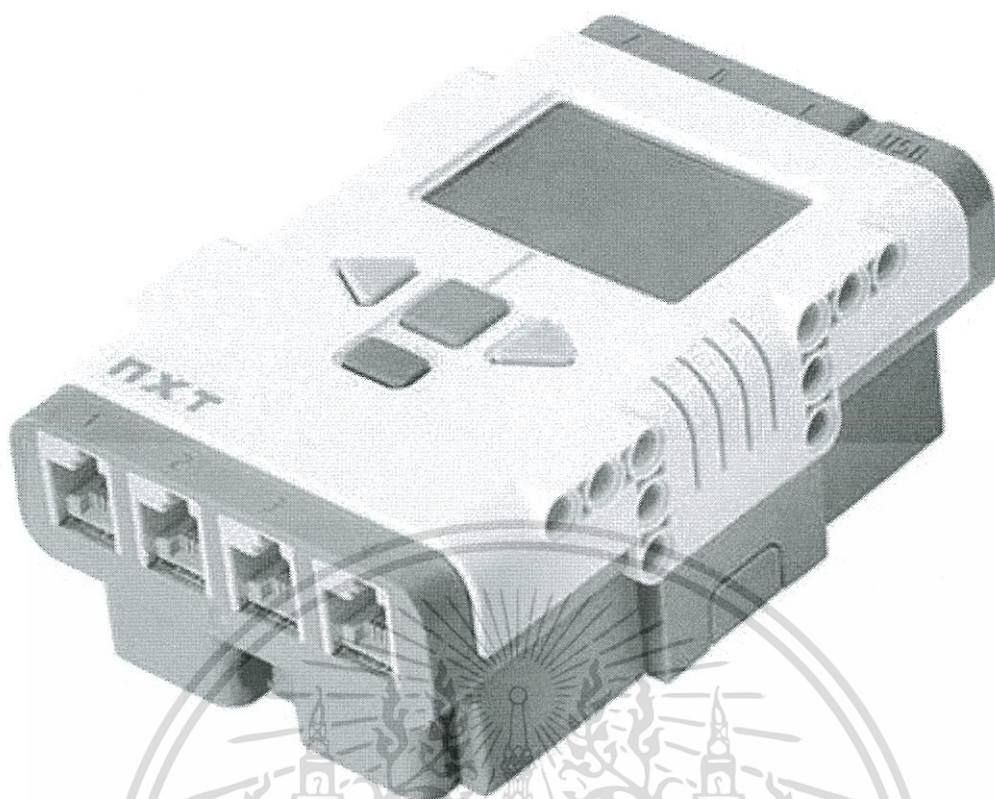
(ที่มา: <http://krunitit.rwb.ac.th/hardware.html>)

LEGO MINDSTORMS NXT เป็นชุดประกอบหุ่นยนต์ ที่ประกอบด้วยโมดูลต่างๆ เช่น ไมโครคอมพิวเตอร์ เซ็นเซอร์ มอเตอร์ และตัวต่ออื่นๆ

### 2.1.3 โมดูลต่างๆ

- คอนโทรลเลอร์อัจฉริยะ(NXT intelligent brick controller) แบบ 32 บิต มีระบบเชื่อมต่อ USB และ บลูทูธ ประกอบด้วย 4 input และ 3 output ใช้พลังงานจากถ่านขนาด AA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 NXT intelligent brick controller  
(ที่มา: <http://krunisit.rwb.ac.th/hardware.html>)

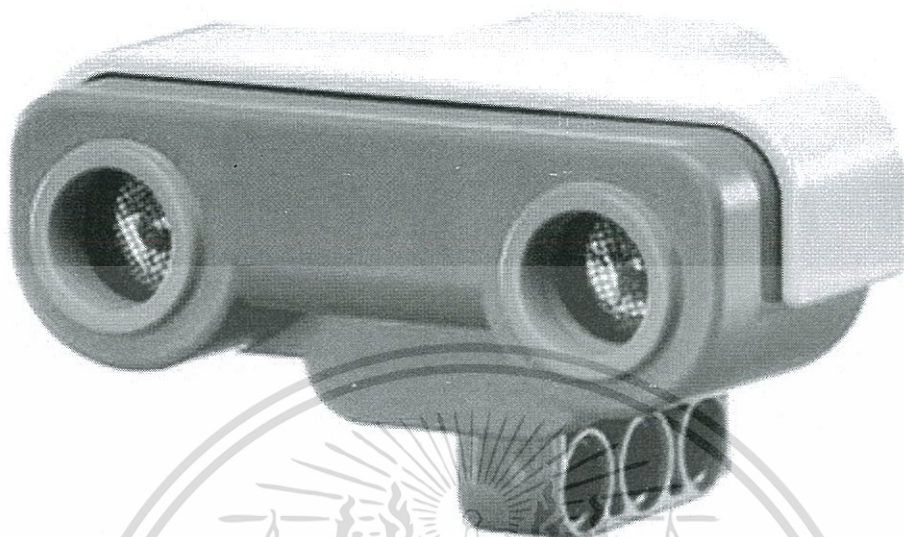
- มอเตอร์ขับเคลื่อน (Interactive servo motor) มีเซ็นเซอร์วัดรอบ ความเร็ว และระยะทาง จากนั้นส่งผลกลับมาที่คอนโทรลเลอร์



รูปที่ 2.4 มอเตอร์ขับเคลื่อน (Interactive servo motor)  
(ที่มา: <http://krunisit.rwb.ac.th/hardware.html>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

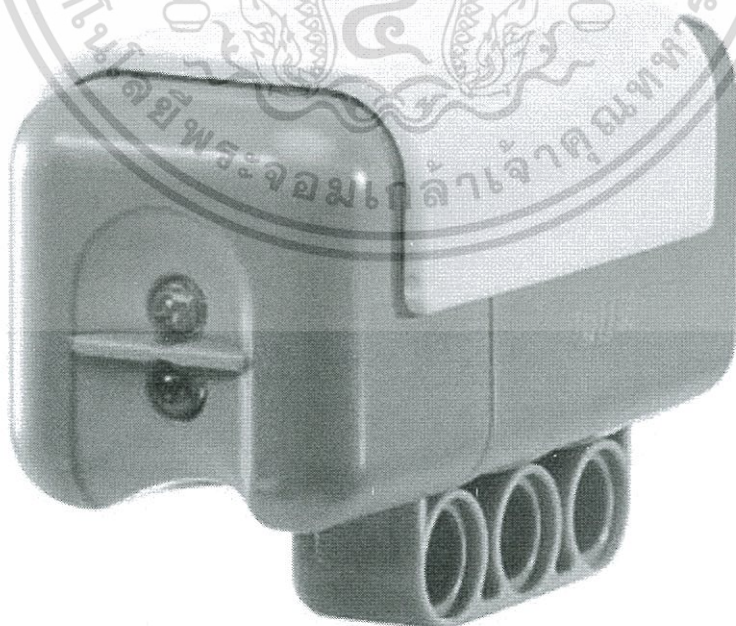
- ตัวตรวจจับวัตถุ (Ultrasonic sensor) สามารถวัดระยะห่างจากตัวเซ็นเซอร์ ถึงวัตถุที่อยู่  
ด้านหน้า



รูปที่ 2.5 ตัวตรวจจับวัตถุ (Ultrasonic sensor)

(ที่มา: <http://krunisit.rwb.ac.th/hardware.html>)

- ตัวตรวจจับค่าแสง (Light Sensor) สามารถวัดค่าความสว่างหรือมืด และวัดค่าความเข้ม  
ของแสง



รูปที่ 2.6 ตัวตรวจจับค่าแสง (Light Sensor)

(ที่มา: <http://krunisit.rwb.ac.th/hardware.html>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 องค์ประกอบด้านซอฟต์แวร์

### 2.2.1 ภาษา Java



รูปที่ 2.7 ภาษา Java

(ที่มา: <http://resource.thaicreate.com/upload/tutorial/java-what-is-java-03.jpg?v=1001>)

JAVA คือภาษาโปรแกรมเชิงวัตถุ เป็นภาษาที่พัฒนาเพื่อมาแทน C++ โดยภาษา Java จะใช้หลักการเขียนแบบ object-oriented-programming โปรแกรมจะเขียนภายใต้คลาส ภายใต้คลาสจะเก็บ method โดยเราจะใช้ Java ในการเขียนชุดคำสั่งเพื่อควบคุมการทำงานและสั่งงานหุ่นยนต์ไปสำรวจในพื้นที่นั้นๆ

#### ข้อดีของ Java

- สามารถใช้ได้กับหลาย platform ไม่ต้องนำไป compile ใหม่ให้สะดวกในการนำไปใช้
- มีการตรวจสอบในระหว่างที่ compile ทำให้เรารู้สึกเป็นปัญหาของโปรแกรม
- มี LeJOS NXT และ LeJOS PC Libraries ที่ใช้ในการเขียนคำสั่งในหุ่นยนต์และฝั่งเซิร์ฟเวอร์โดยเฉพาะ

#### 2.2.1.1 Lejos NXT Library

เป็นไลบรารีที่ใช้ในการเขียนโปรแกรมคำสั่งลงในหุ่นยนต์โดยเฉพาะ ใช้งานไม่ยาก สามารถหาคำอธิบายคำสั่งต่างๆ จากทาง [www.lejos.org](http://www.lejos.org) ได้เลย ก่อนจะใช้งานจำเป็นจะต้องติดตั้งปลั๊กอินของทาง LeJOS ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Install

### Available Software

Check the items that you wish to install.

Work with:

Find more software by working with the

type filter text

Name	Version
> <input type="checkbox"/> 000 LeJOS NXJ Support	

Select All    Deselect All

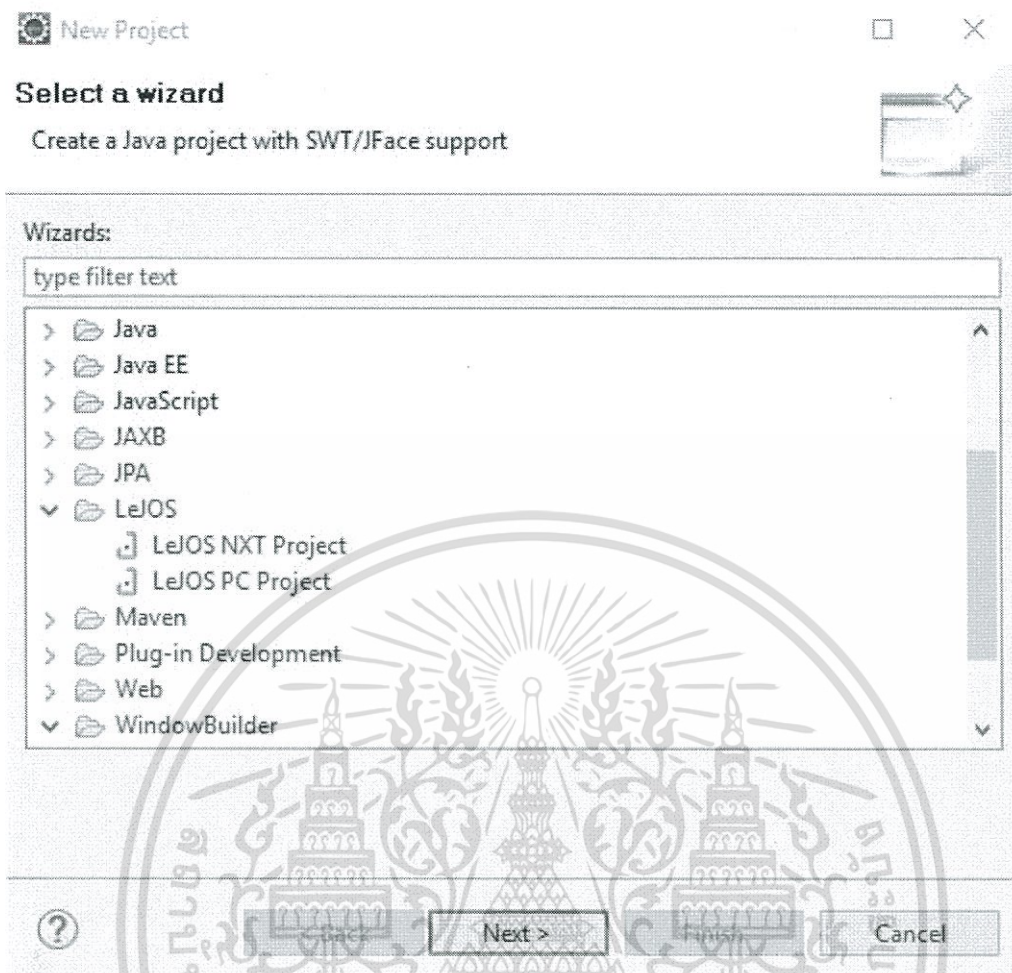
Details

### รูปที่ 2.8 การติดตั้งปลั๊กอิน LeJOS NXT Support

#### 2.2.1.2 Lejos PC Library

เป็นไลบรารีอีกตัวที่มีมาให้จากการติดตั้งปลั๊กอินของ LeJOS แต่ไลบรารีนี้จะใช้เขียนโปรแกรมทางฝั่งพีซีหรือเซิร์ฟเวอร์เพื่อติดต่อกับทางฝั่งหุ่นยนต์เพื่อใช้ในการสตรีมรับ-ส่งค่าต่างๆ ระหว่างหุ่นยนต์และเซิร์ฟเวอร์

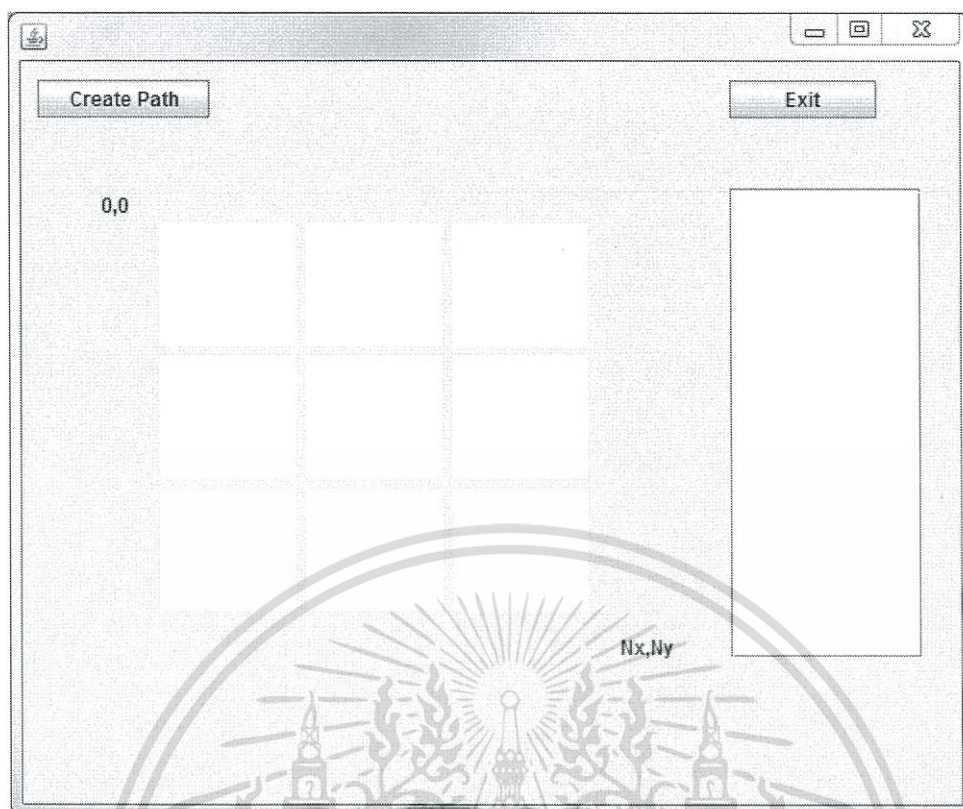
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การสร้างโปรเจกต์ LeJOS NXT และ LeJOS PC

### 2.2.1.3 Java Swing GUI

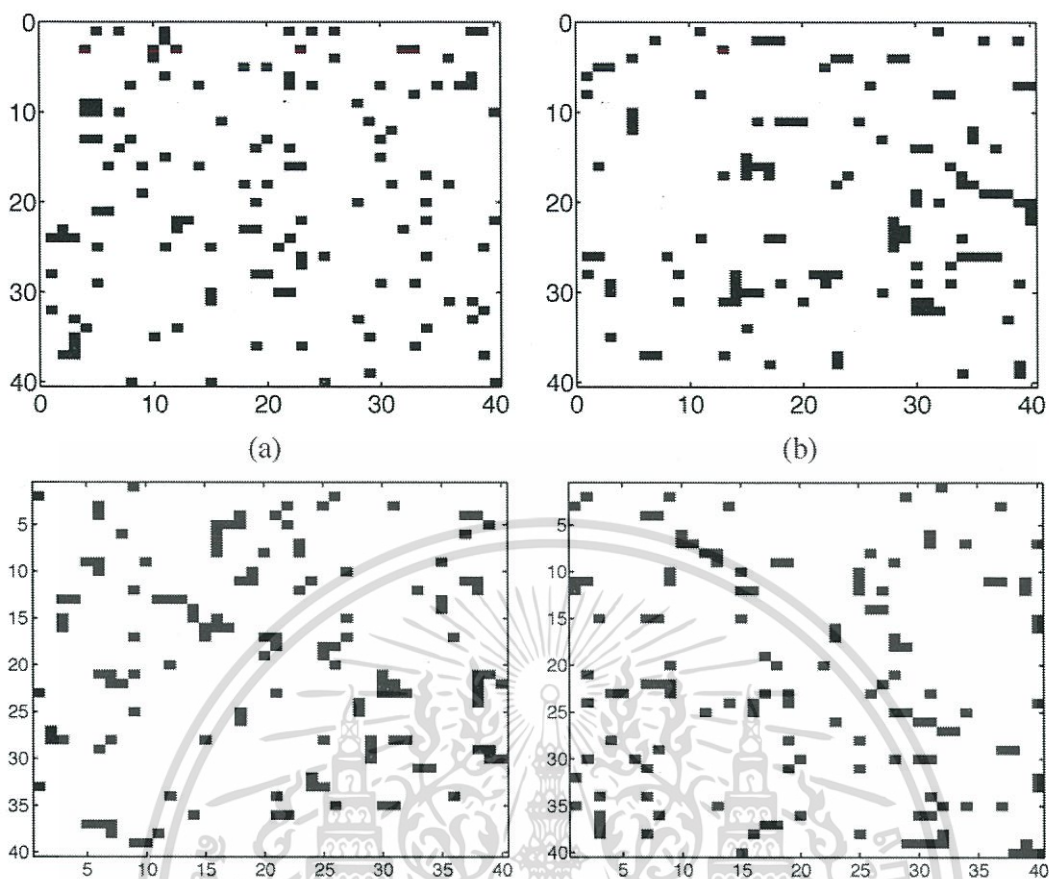
Java swing GUI คือการสร้างหน้าต่างใช้งานในรูปแบบกราฟฟิก ซึ่งจะประกอบด้วย Label, Textbox , Button รวมไปถึงการสร้าง event ต่างๆ โดย Java swing GUI นั้นยังสามารถใช้สร้างแบบจำลองสภาพแวดล้อมที่หุ่นยนต์ใช้สำรวจได้อีกด้วย



รูปที่ 2.10 การใช้Java swing GUI

### 2.3 กระบวนการค้นหาเส้นทางของหุ่นยนต์เคลื่อนที่

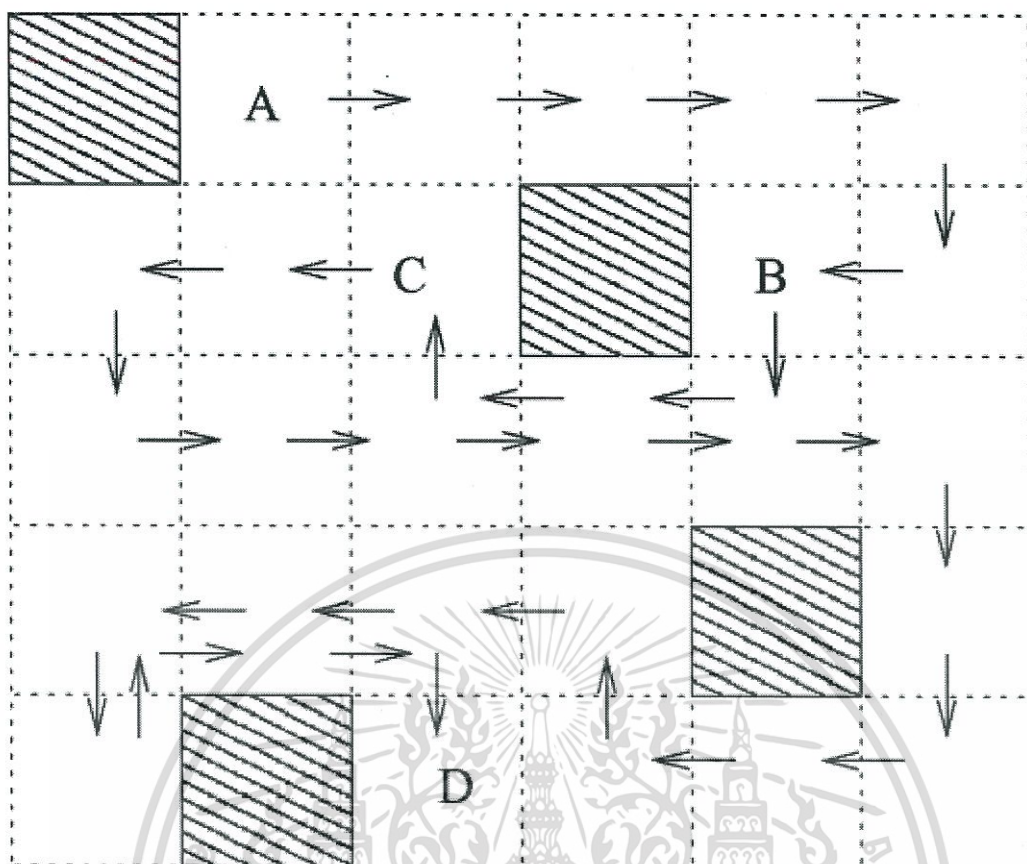
อัลกอริทึมที่ใช้จะเป็นอัลกอริทึมที่เกี่ยวข้องกับการเคลื่อนที่ในแบบจำลองห้องที่มีสิ่งกีดขวาง โดยแบบจำลองจะมีลักษณะดังภาพด้านล่าง โดยสีดำคือสิ่งกีดขวางที่จำลองขึ้น



รูปที่ 2.1 การจำลองห้องที่มีสิ่งกีดขวาง

(ที่มา: [https://engineering.purdue.edu/~ychu/publications/tr06\\_deploy.pdf](https://engineering.purdue.edu/~ychu/publications/tr06_deploy.pdf))

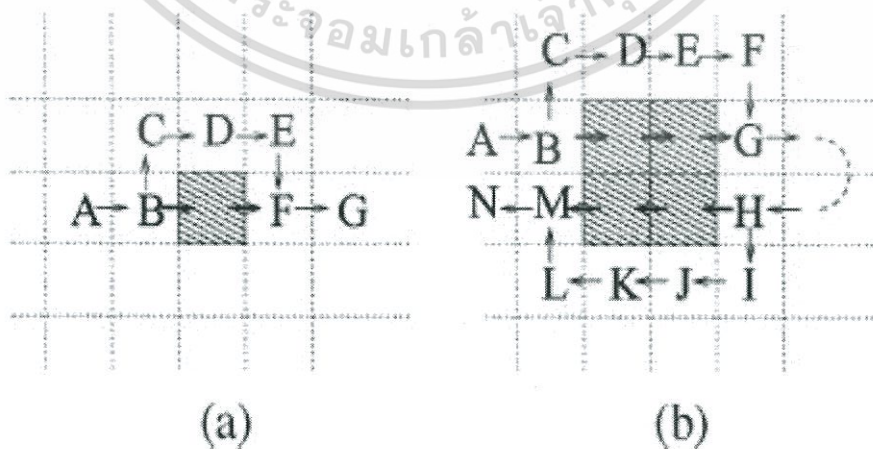
โดยสิ่งกีดขวางในแผนที่เกิดจากการสุ่มด้วยอัลกอริทึมให้มีปริมาณของสิ่งกีดขวางเป็นจำนวน 10 % ของพื้นที่ห้องทั้งหมด โดยหุ่นยนต์จะทำการเคลื่อนที่หลังจากเจอสิ่งกีดขวางและมีการเคลื่อนที่เป็นลำดับ A,B,C,D ตามลำดับ ดังรูปที่ 2.12



รูปที่ 2.12 การเคลื่อนที่ของหุ่นยนต์

(ที่มา: [https://engineering.purdue.edu/~ychu/publications/tr06\\_deploy.pdf](https://engineering.purdue.edu/~ychu/publications/tr06_deploy.pdf))

นอกจากเรื่องการกำหนดสิ่งกีดขวางในแผนที่ซึ่งต้องคำนึงถึงการสำรวจบริเวณรอบๆ สิ่งกีดขวางเพื่อให้การสำรวจของหุ่นยนต์มีประสิทธิภาพสูงสุดและครอบคลุมพื้นที่ทั้งหมด โดยการเคลื่อนที่ผ่านสิ่งกีดขวางจะแบ่งออกเป็น 2 ประเภทดังรูปที่ 2.13



รูปที่ 2.13 การเคลื่อนที่ผ่านสิ่งกีดขวางแบบ a และ b

(ที่มา: [https://engineering.purdue.edu/~ychu/publications/tr06\\_deploy.pdf](https://engineering.purdue.edu/~ychu/publications/tr06_deploy.pdf))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาด์เห็นาเบไซประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การทำงานให้ครอบคลุมนั้น หุ่นยนต์ทำงานตามขั้นตอน เริ่มที่เซลล์ปัจจุบันหุ่นยนต์เลือกเส้นทาง การเดินโดยการเลือกเมกาเซลล์ใหม่ที่สามารถเคลื่อนที่ไปได้ในรูปแบบทวนเข็มนาฬิกาจากนั้น Spanning tree ใหม่จะแผ่ขยายต่อไปยัง เมกาเซลล์ใหม่ๆ อัลกอริทึมนี้เป็นแบบการทำงานซ้ำไปมา การทำงานของอัลกอริทึมนี้จะหยุดก็ต่อเมื่อไม่มีเซลล์ใหม่เหลืออยู่แล้ว (เมกาเซลล์ที่ถือว่าไม่ใช่ เซลล์ใหม่คือหุ่นยนต์ได้ทำการเคลื่อนที่ผ่านเซลล์เล็กด้านในหมดแล้ว) นี่คือผลของการทำงานซ้ำๆ หุ่นยนต์จะเคลื่อนที่ไปตาม Spanning tree จนสุดปลายด้านหนึ่งของ Spanning tree จะสังเกตได้ว่า เมื่อหุ่นยนต์ทำการสำรวจเสร็จสมบูรณ์หุ่นยนต์จะกลับมาอยู่ที่เซลล์เริ่มต้น แล้วทำการจัดเก็บข้อมูล ในทางกลับกันหุ่นยนต์จะไม่ทำการเคลื่อนที่ไปซ้ำเซลล์เล็กที่เคยเดินผ่านมาแล้ว

ขั้นตอนการสำรวจ

- กำหนดให้  $s$  คือเซลล์เริ่มต้น
- $x$  คือเซลล์ปัจจุบัน
- $w$  คือเซลล์ที่อยู่ก่อนหน้าในแบบ spanning tree
- ฟังก์ชัน STC เป็นฟังก์ชันแบบเรียกใช้งานซ้ำ
- ขั้นตอนที่ 1 กำหนดให้เซลล์ปัจจุบันคือ  $x$
- ขั้นตอนที่ 2 เมื่อเซลล์  $x$  ยังมีเซลล์เพื่อนบ้านที่สามารถไปได้
  - ขั้นตอนที่ 2.1 หาเซลล์เพื่อนบ้านที่อยู่ในด้านทวนเข็มนาฬิกาจาก  $w$  คือเซลล์  $y$
  - ขั้นตอนที่ 2.2 สร้าง spanning tree จาก  $x$  ไป  $y$
  - ขั้นตอนที่ 2.3 เคลื่อนที่จากเซลล์  $x$  ไปหา  $y$
  - ขั้นตอนที่ 2.4 รันอัลกอริทึม STC ไปเรื่อยๆ
- เมื่อจบ loop
- ขั้นตอนที่ 3 ถ้า  $x$  เป็นคนละเซลล์กับ  $s$  ถอยหลัง  $x$  ไปยังช่องเล็กของ  $w$
- ขั้นตอนที่ 4 จบการทำงาน (หุ่นยนต์กลับมายังจุดเริ่มต้น)

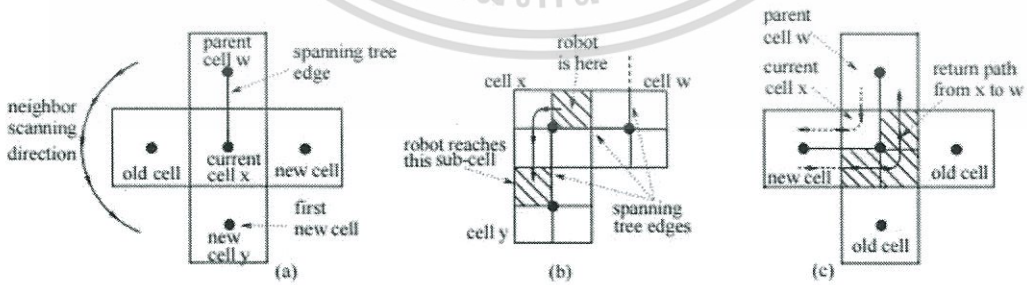


Figure 3: (a) Counterclockwise scanning of four neighbors. (b) A move from  $x$  to a new cell  $y$ . (c) A return from  $x$  to a parent cell  $w$ .

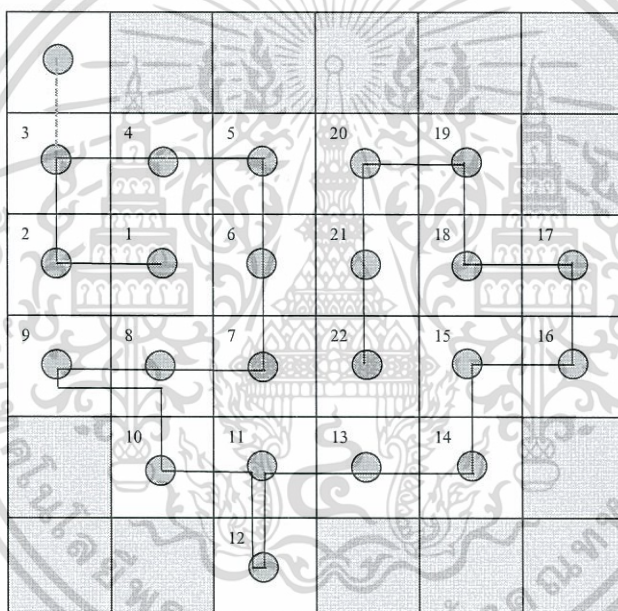
รูปที่ 2.15 รูปประกอบการทำงานของอัลกอริทึม STC

(ที่มา: Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.4 การนำอัลกอริทึม Spiral-STC มาประยุกต์ใช้

เริ่มทำการสำรวจโดยการแบ่งพื้นที่สำรวจเป็นตาราง โดย 1 ช่องของตารางจะมีขนาดเท่ากับขนาดของหุ่นยนต์ หากเซลล์นั้นมีขนาดพื้นที่น้อยกว่าที่หุ่นจะสามารถเข้าไปได้ จะถือว่าเป็นสิ่งกีดขวาง จากจุดเริ่มต้น จะแบ่งการสำรวจออกเป็น 2 ขา คือขาไปและขากลับ การสำรวจของหุ่นยนต์ขาไปจะเลือกเส้นทางการเดินในรูปแบบทวนเข็มนาฬิกา วนเป็นลักษณะก้นหอย หุ่นยนต์จะทำการบันทึกข้อมูลในทุกๆ เซลล์ที่เดินไป โดยเซลล์ไหนที่หุ่นยนต์ได้ทำการเดินสำรวจแล้ว จะไม่เดินซ้ำอีกเป็นครั้งที่สอง ยกเว้นว่าหุ่นยนต์จำเป็นต้องเดินย้อนกลับเนื่องจากไม่สามารถไปข้างหน้าต่อไปได้ (เช่น จุดที่ 9 และจุดที่ 12) หากทำการสำรวจจนไม่มีทางที่สามารถไปต่อไปได้แล้ว หุ่นยนต์จะทำการเดินย้อนกลับเส้นทางเดิม พร้อมกับหาเส้นทางใหม่ที่ยังไม่ได้ทำการสำรวจ หากหุ่นยนต์เดินย้อนกลับมาถึงจุดเริ่มต้น จะถือว่าเสร็จสิ้นการสำรวจพื้นที่นั้น ๆ



รูปที่ 2.16 แบบจำลองการเคลื่อนที่ของหุ่นยนต์

### บทที่ 3

## การวิเคราะห์และออกแบบระบบ

### 3.1 การวิเคราะห์การทำงานพื้นฐานของหุ่นยนต์และเซิร์ฟเวอร์

#### 3.1.1 การทดสอบและออกแบบคำสั่งการทำงานของหุ่นยนต์

- ทดสอบการสั่งงานหุ่นยนต์

การสั่งการหุ่นยนต์นั้นจะทำการส่งผ่านระบบบลูทูธ โดยที่เซิร์ฟเวอร์จะส่งคำสั่งไปยังหุ่นยนต์เพื่อให้มอเตอร์ต่างๆ ของหุ่นยนต์ทำงานได้ตามที่ต้องการ เช่น เซิร์ฟเวอร์สั่งให้มอเตอร์ B และมอเตอร์ C ซึ่งเป็นมอเตอร์ล้อทั้งสองข้างของหุ่นยนต์หมุนไปข้างหน้า หุ่นยนต์ก็จะเคลื่อนที่ไปข้างหน้าตามที่เซิร์ฟเวอร์สั่ง

- การทดสอบเซ็นเซอร์อัลตราโซนิกส์

เซ็นเซอร์อัลตราโซนิกส์คือเซ็นเซอร์ที่ใช้ตรวจจับสิ่งกีดขวาง โดยมีหลักการทำงานคือการส่งคลื่นความถี่สูงออกไปทางด้านหน้า หากมีสิ่งกีดขวาง คลื่นจะสะท้อนกลับมายังตัวเซ็นเซอร์ โดยเซ็นเซอร์จะแปลงระยะเวลาที่คลื่นสะท้อนกลับออกมาเป็นระยะห่างระหว่างเซ็นเซอร์กับสิ่งกีดขวาง มีค่าตั้งแต่ 0 – 255

เราได้ทำการทดลองหาค่าที่เหมาะสมที่สุดที่บ่งบอกว่าด้านหน้าของหุ่นมีสิ่งกีดขวาง ค่าที่ได้คือ 35 หน่วย เนื่องจากระยะห่างเท่านี้จะมีขนาดเท่ากับตัวหุ่นยนต์พอดี หากน้อยกว่าหรือเท่ากับ 35 หน่วย จะถือว่าสิ่งกีดขวางทันที เพราะหุ่นยนต์จะไม่สามารถเดินเข้าไปสำรวจได้

- การบังคับเลี้ยวของหุ่นยนต์

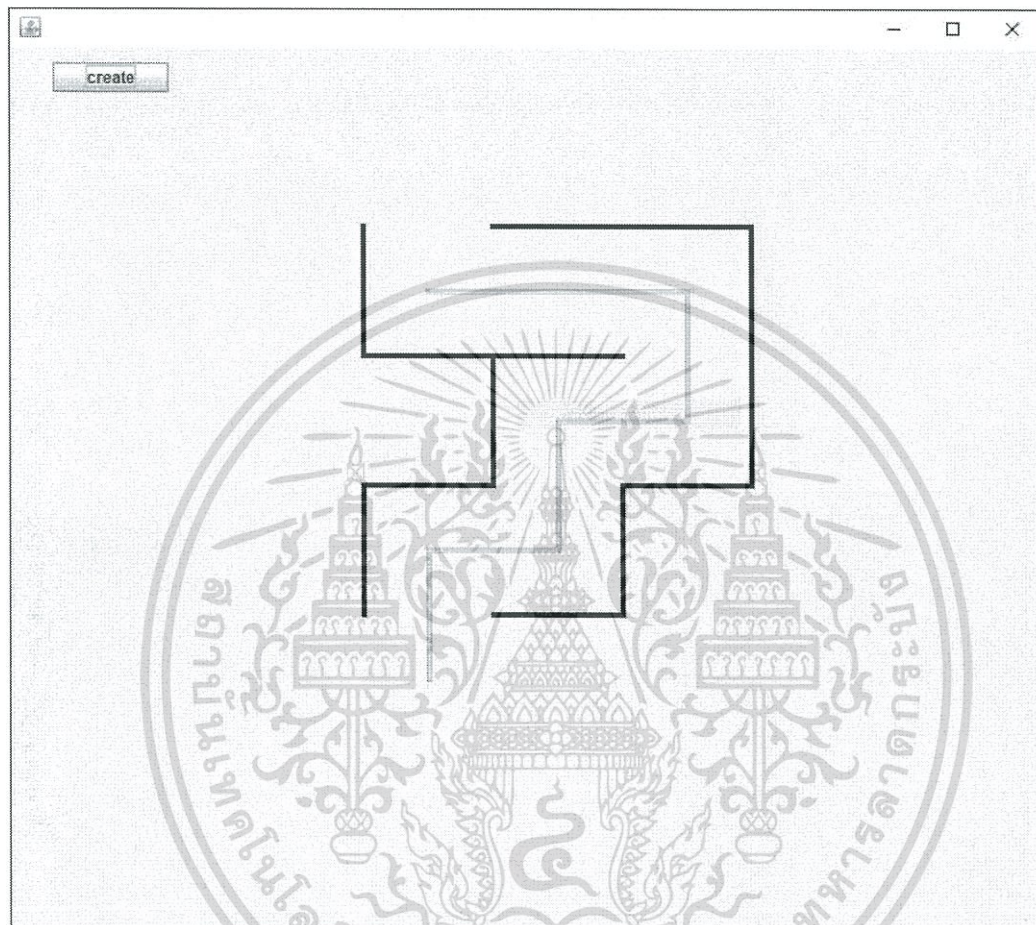
การบังคับเลี้ยวของหุ่นยนต์ทำได้โดยการสั่งให้มอเตอร์ล้อซ้าย และมอเตอร์ล้อขวาหมุนสลับกันทำมุม 90 องศา เช่น หากต้องการเลี้ยวขวา เซิร์ฟเวอร์ก็จะสั่งให้มอเตอร์ล้อซ้ายหมุนไปทางด้านหน้า สั่งให้มอเตอร์ล้อขวาหมุนไปทางด้านหลัง หุ่นยนต์ก็จะสามารถเลี้ยวไปทางด้านขวาได้

- การเคลื่อนที่ในระยะทางที่กำหนด

เนื่องจากหุ่นยนต์เคลื่อนที่เป็นเซลล์ แต่ละเซลล์มีขนาดเท่ากับตัวหุ่นยนต์พอดี เราจึงต้องทดลองหาระยะทางที่หุ่นยนต์เดิน 1 ครั้ง เท่ากับ 1 เซลล์ โดยจับการจับเวลา หากมอเตอร์หมุนด้วยความเร็วคงที่ ต้องหมุนเป็นระยะเวลาสั้นเท่าใดหุ่นยนต์ก็จะเคลื่อนที่เป็นระยะทางตามที่ต้องการ

### 3.1.2 การออกแบบผังเซิร์ฟเวอร์

การออกแบบโปรแกรมผังเซิร์ฟเวอร์โดยจะออกแบบให้เซิร์ฟเวอร์รับค่าพิกัด และข้อมูลของแต่ละพิกัดที่ได้จากการสำรวจมาจากหุ่นยนต์ นำมาสร้างเป็นแผนที่จำลองโดยการรับค่าจากหุ่นยนต์นั้นจะใช้ LeJOS PC Library ส่วนการสร้างแผนที่จำลองจะใช้ JAVA Swing GUI



รูปที่ 3.1 การใช้ Java swing GUI ในการจำลองแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การวิเคราะห์และการ Implement อัลกอริทึม Spiral-STC (Spanning Tree Coverage)

#### 3.2.1 ข้อจำกัดในการนำอัลกอริทึม Spiral-STC มาใช้ในหุ่นยนต์ LEGO

- ไม่มี GPS

ในการสำรวจของหุ่นยนต์จำเป็นต้องรู้ตำแหน่งปัจจุบันของหุ่นยนต์เพื่อที่จะบันทึกเป็นข้อมูลว่า ณ ตำแหน่งนั้นๆ หุ่นยนต์เจอสิ่งกีดขวางหรือไม่

- ไม่รู้ทิศทางของหุ่นยนต์

การค้นหาเส้นทางของอัลกอริทึม Spiral-STC จำเป็นต้องรู้ทิศทางการหันของหุ่นยนต์เพื่อหาเซลล์ที่จะต้องเดินต่อไป หากไม่รู้ทิศทางของหุ่นยนต์ก็ไม่สามารถหาเส้นทางไปต่อได้

#### 3.2.2 แนวทางการแก้ไขปัญหา

- การแก้ไขปัญหา GPS

คำนวณตำแหน่งปัจจุบันของหุ่นโดยการกำหนดพิกัด  $x, y$  ให้กับหุ่น โดยจะอธิบายลักษณะการทำงานหัวข้อ 3.2.3 ต่อไป

- การแก้ไขปัญหาการหาทิศทางของหุ่นยนต์

เพิ่มอัลกอริทึมการหาทิศทางของหุ่นยนต์ โดยการหาทิศทางของหุ่นยนต์เริ่มจากกำหนดค่าตัวเลขแทนทิศของหุ่นยนต์จากจุดเริ่มต้น กำหนดให้ทิศทางด้านหน้า ขวา หลัง ซ้ายของหุ่นยนต์มีค่าเท่ากับ 1 2 3 และ 4 ตามลำดับ แทนด้วยตัวแปร  $drt$  หากหุ่นยนต์หมุนขวา ค่า  $drt$  ก็จะเพิ่มขึ้น หากเพิ่มเกิน 4 ก็จะวนกลับมาที่ 1 ใหม่ หากหุ่นยนต์หมุนซ้าย ค่าของ  $drt$  ก็จะลดลง หากลดลงน้อยกว่า 1 ก็จะวนกลับไป 4 ใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเชิงวิชาการเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Flowchart การหาทิศทางของหุ่นยนต์

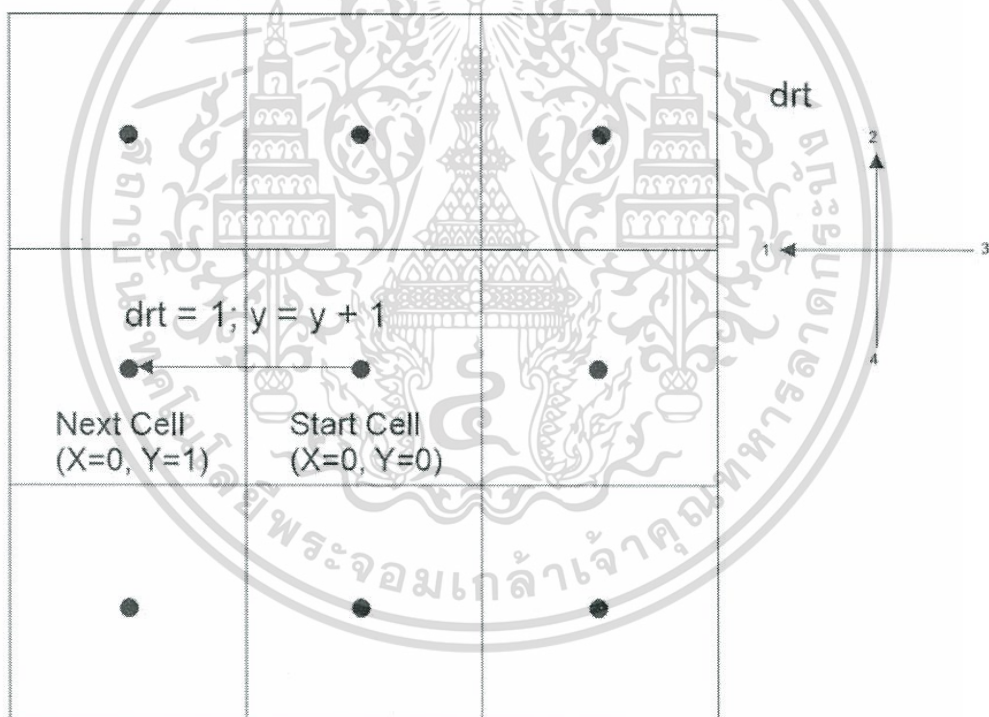
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 การนำอัลกอริทึม Spiral-STC มาประยุกต์ใช้

การหาค้นหาเส้นทางแบบ Spiral-STC ได้มีการแบ่งเซลล์ออกเป็นสองขนาดโดยเซลล์ขนาดเล็กจะมีขนาดเท่ากับตัวหุ่นยนต์และเซลล์ใหญ่มีขนาดเท่ากับ 4 เซลล์เล็กรวมกันเรียกว่า เมกาเซลล์ แต่การแบ่งเซลล์ออกเป็นสองขนาดมีความซับซ้อนและยุ่งยากมาก เราจึงเลือกแบ่งเซลล์ออกเป็นขนาดเดียวคือขนาดที่เท่ากับตัวหุ่นยนต์ โดยมีอัลกอริทึมต่างๆ ในการค้นหาเส้นทางสำรวจ ดังนี้

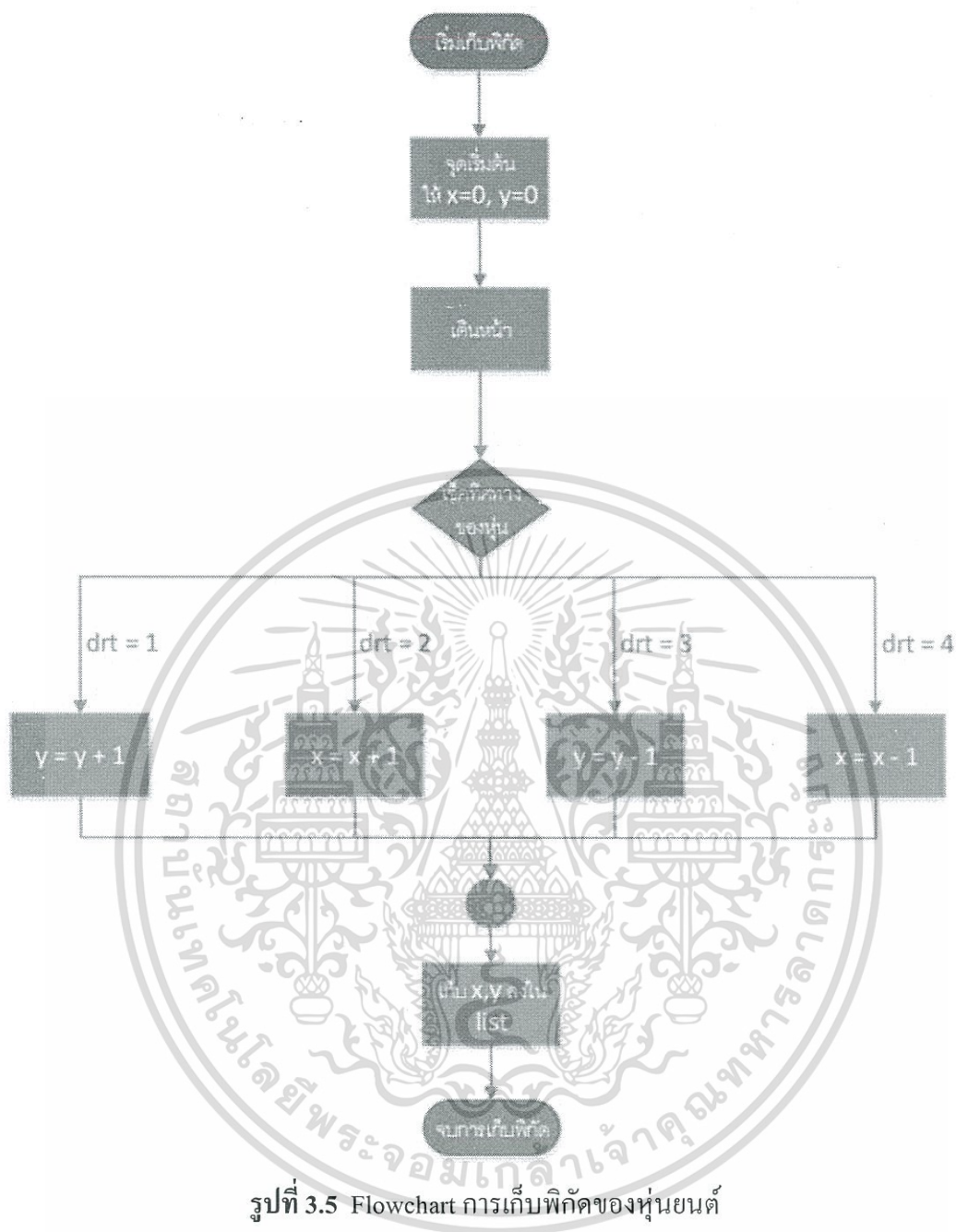
#### 3.2.3.1 อัลกอริทึมการเก็บพิกัดของหุ่นยนต์

การเก็บพิกัดของหุ่นยนต์จะเริ่มจากการกำหนดให้จุดเริ่มต้นมีค่าเท่ากับ  $x = 0, y = 0$  ทุกครั้งที่หุ่นทำการเดิน จะเรียกใช้ฟังก์ชันการหาทิศทางของหุ่นยนต์ว่ากำลังหันหน้าไปทางไหน ก็จะทำการเพิ่มหรือลด  $x, y$  ตามด้านที่หุ่นยนต์เดินไปแล้วทำการเก็บค่า  $x, y$  ลงใน List



รูปที่ 3.4 การเก็บพิกัดของหุ่นยนต์

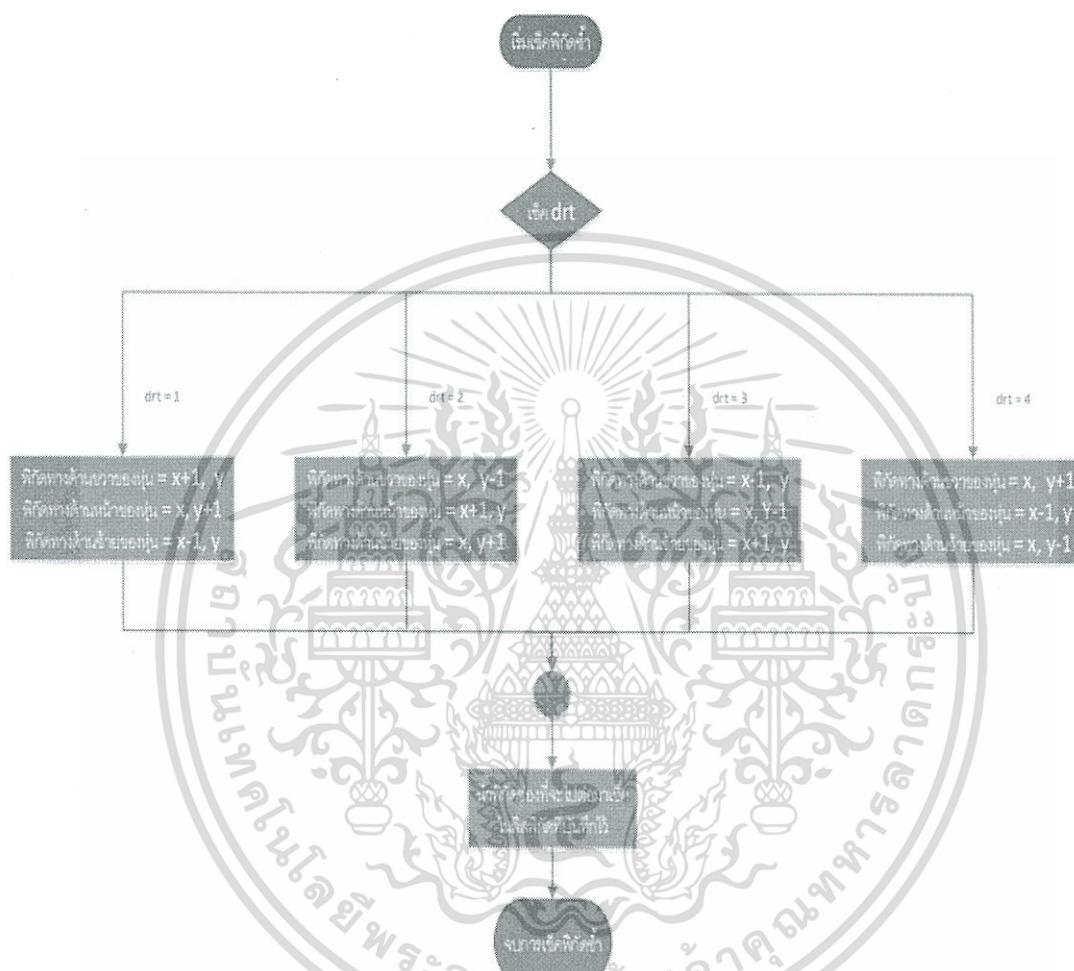
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.2 อัลกอริทึมการตรวจสอบการเดินซ้ำของหุ่นยนต์

หุ่นยนต์จะเริ่มทำการตรวจสอบเซลล์ที่จะเดินต่อไปว่าเคยสำรวจมาหรือยัง โดยเริ่มจากการเรียกใช้ฟังก์ชันหาทิศทางของหุ่นยนต์ เมื่อทราบทิศทางที่กำลังหันหน้าไปก็จะนำเอาพิกัดของเซลล์ที่จะเดินต่อไปเช็คในลิสต์ที่บันทึกพิกัดที่หุ่นยนต์เดินสำรวจมาแล้วทั้งหมดว่าซ้ำกันหรือไม่

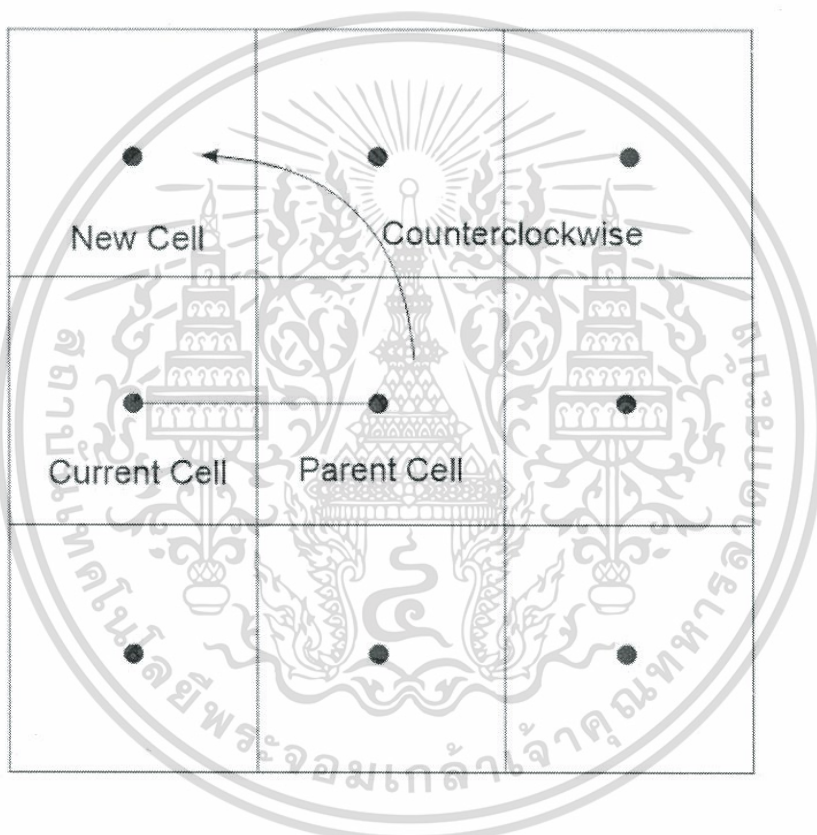


รูปที่ 3.6 Flowchart การตรวจสอบการเดินซ้ำของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.3 อัลกอริทึมการเดินทางสำรวจขาไปของหุ่นยนต์

หุ่นยนต์จะเริ่มทำการสำรวจเส้นทางขาไปโดยเรียกใช้ฟังก์ชันเก็บพิกัดตำแหน่งปัจจุบันลงในลิสต์ จากนั้นทำการตรวจจับสิ่งกีดขวางทางด้านขวา ด้านหน้า และด้านซ้าย หากระยะที่ตรวจจับได้จากทางด้านขวามีค่ามากกว่า 40 ก็จะนำเอาพิกัดของเซลล์ทางด้านขวามาตรวจสอบการเดินทางซ้ำของหุ่นยนต์ หากเซลล์นั้นหุ่นยนต์ยังไม่เคยไปสำรวจมาก่อน หุ่นยนต์ก็จะเลือกเซลล์นั้นเป็นเส้นทางที่จะเดินต่อไป แต่ถ้าหากระยะที่ตรวจจับได้จากทางด้านขวามีค่าน้อยกว่า 40 หรือเซลล์ทางด้านขวาเคยไปทำการสำรวจมาแล้ว ก็จะเลือกทางเดินทางด้านอื่นต่อไป โดยการเลือกเส้นทางของหุ่นยนต์จะเลือกเดินทางด้านขวา ด้านหน้า และด้านซ้ายตามลำดับเป็นรูปแบบทวนเข็มนาฬิกา



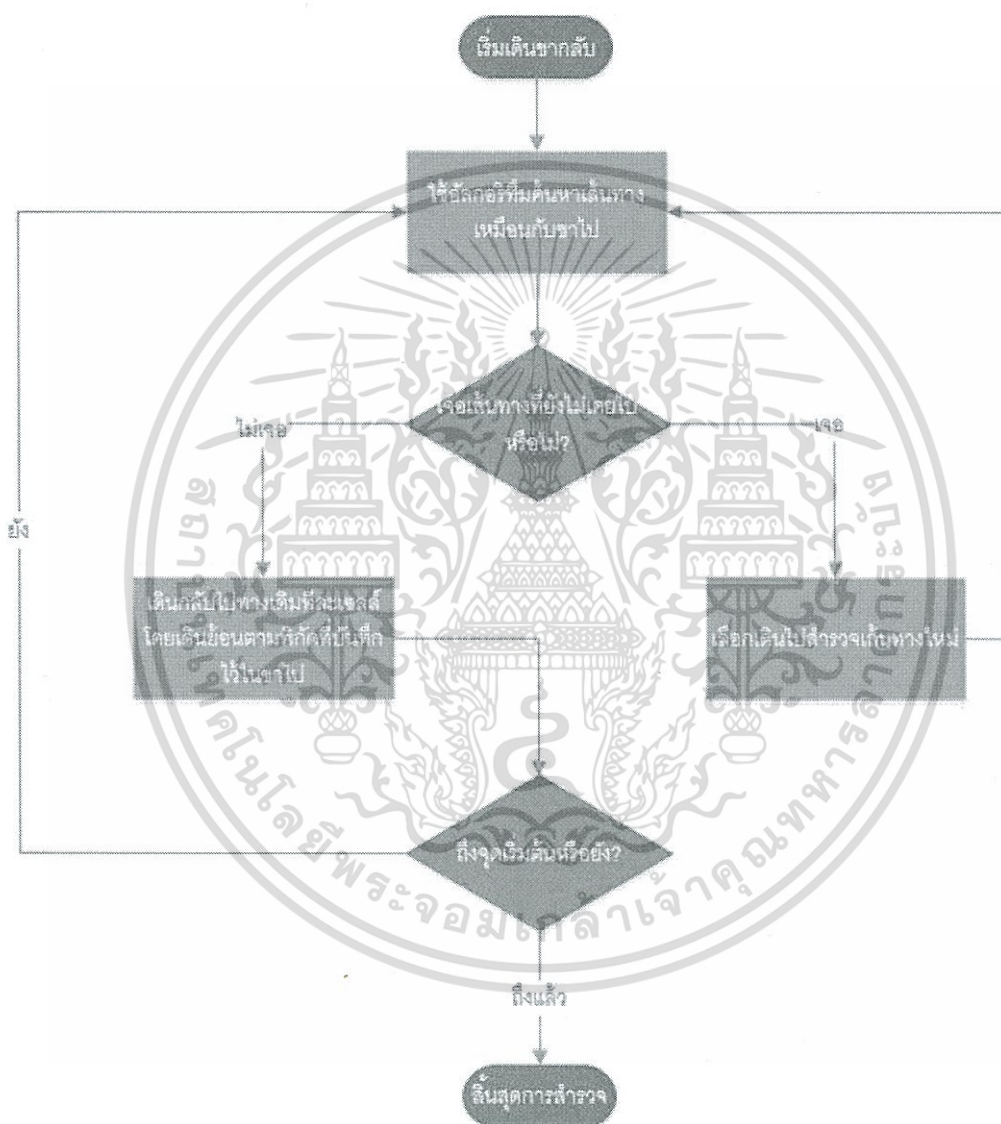
รูปที่ 3.7 การเลือกเส้นทางสำรวจของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 3.2.3.4 อัลกอริทึมการเดินสำรวจจากลับของหุ่นยนต์

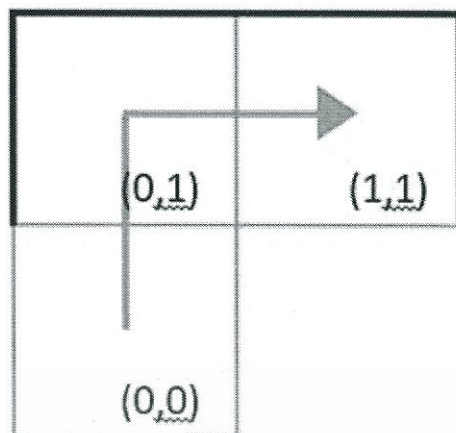
เมื่อหุ่นยนต์เดินสำรวจจนไม่มีเส้นทางให้เลือกเดินต่อ หุ่นยนต์จะเริ่มเดินย้อนกลับตามเส้นทางเดิมที่เดินมาที่ละเซลล์ ระหว่างการเดินกลับก็จะค้นหาเส้นทางที่ยังไม่เคยเดินไปสำรวจ หากเจอเส้นทางใหม่ก็จะเลือกไปสำรวจตามเส้นทางใหม่โดยใช้อัลกอริทึมเดียวกับการเดินสำรวจมา หากไม่เจอเส้นทางใหม่แล้วก็จะด้านย้อนกลับมาจนถึงจุดเริ่มต้นซึ่งถือว่าสิ้นสุดการสำรวจ



รูปที่ 3.9 Flowchart การเดินสำรวจจากลับของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การจำลองแผนที่ยากจากเซ็นเซอร์อัลตราโซนิกส์



รูปที่ 3.10 การจำลองแผนที่ยากจากเซ็นเซอร์อัลตราโซนิกส์

เส้นสีดำ คือสิ่งกีดขวางในแผนที่โดยค่าจากเซ็นเซอร์อัลตราโซนิก  $< 40$

เส้นสีน้ำเงิน คือเส้นแบ่งเซลล์ที่ใช้ในการสำรวจโดยค่าจากเซ็นเซอร์อัลตราโซนิก  $> 40$

เส้นสีเขียว คือเส้นแสดงการเดินของหุ่นยนต์ในเซลล์นั้นๆ

#### ตัวอย่างการจำลองแผนที่

เซลล์เริ่มต้นคือเซลล์  $(0,0)$  โดยหุ่นยนต์จะมีทิศเริ่มต้นเป็น 1 เมื่อหุ่นยนต์ทำการเคลื่อนที่ไปยังเซลล์  $(0,1)$  เป็นการเคลื่อนที่ไปด้านหน้า ทำให้ทิศของหุ่นยนต์ยังมีค่าเป็น 1 เราจึงวาดเส้นการเคลื่อนที่ของหุ่นยนต์เป็นเส้นตรงไปข้างหน้า เมื่อหุ่นยนต์เคลื่อนที่ไปยังเซลล์  $(0,1)$  หุ่นยนต์ทำการสำรวจ พบว่าด้านซ้ายและด้านหน้าของหุ่นยนต์มีสิ่งกีดขวางคือเส้นสีดำโดยทิศของหุ่นยนต์เป็น 1 สิ่งกีดขวางที่พบจึงอยู่ในทิศทางเดียวกับหุ่นยนต์

เมื่อหุ่นยนต์เคลื่อนที่ไปยังเซลล์  $(1,1)$  ซึ่งอยู่ทางด้านขวาทำให้หุ่นยนต์ต้องเลี้ยวขวาทิศของหุ่นยนต์จึงกลายเป็น 2 เราจึงวาดเส้นการเคลื่อนที่ของหุ่นยนต์ไปด้านขวาจากเส้นเดิม จากนั้นหุ่นยนต์ทำการสำรวจพบว่าสิ่งกีดขวางด้านหน้าและด้านซ้ายของหุ่นยนต์ แต่ทิศของหุ่นยนต์มีการเปลี่ยนแปลงจาก 1 เป็น 2 เราจึงวาดเส้นสีดำเส้นบนและเส้นขวา โดยเส้นบนคือสิ่งกีดขวางด้านซ้าย เส้นขวาแทนสิ่งกีดขวางด้านหน้า

### 3.4 การบันทึกข้อมูลที่ได้จากหุ่นยนต์

เซิร์ฟเวอร์จะทำการบันทึกข้อมูลที่หุ่นยนต์ส่งกลับมาจากการสำรวจเป็นข้อมูลดิบ ข้อมูลต่างๆ ที่หุ่นยนต์ส่งกลับมาเซิร์ฟเวอร์ได้แก่ พิกัด, ทิศทางหุ่นยนต์, ระยะของเซ็นเซอร์อัลตราโซนิกส์ที่ตรวจจับได้ทั้งด้านหน้า ขวา ซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

I am the FLASH - Notepad
File Edit Format View Help
(0,1)
direction = 1
right distance = 255
left distance = 50
front distance = 21
(1,1)
direction = 2
right distance = 139
left distance = 22
front distance = 53
(1,0)
direction = 3
right distance = 111
left distance = 22
front distance = 112
(1,-1)
direction = 3
right distance = 122
left distance = 23
front distance = 81
(0,-1)
direction = 4
right distance = 77
left distance = 255

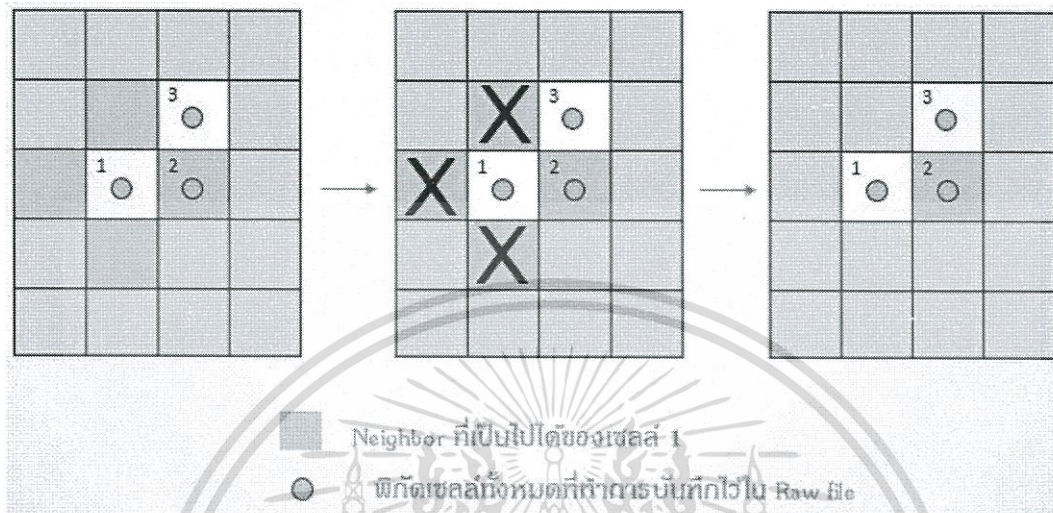
```

รูปที่ 3.11 การบันทึกข้อมูลที่ได้จากหุ่นยนต์

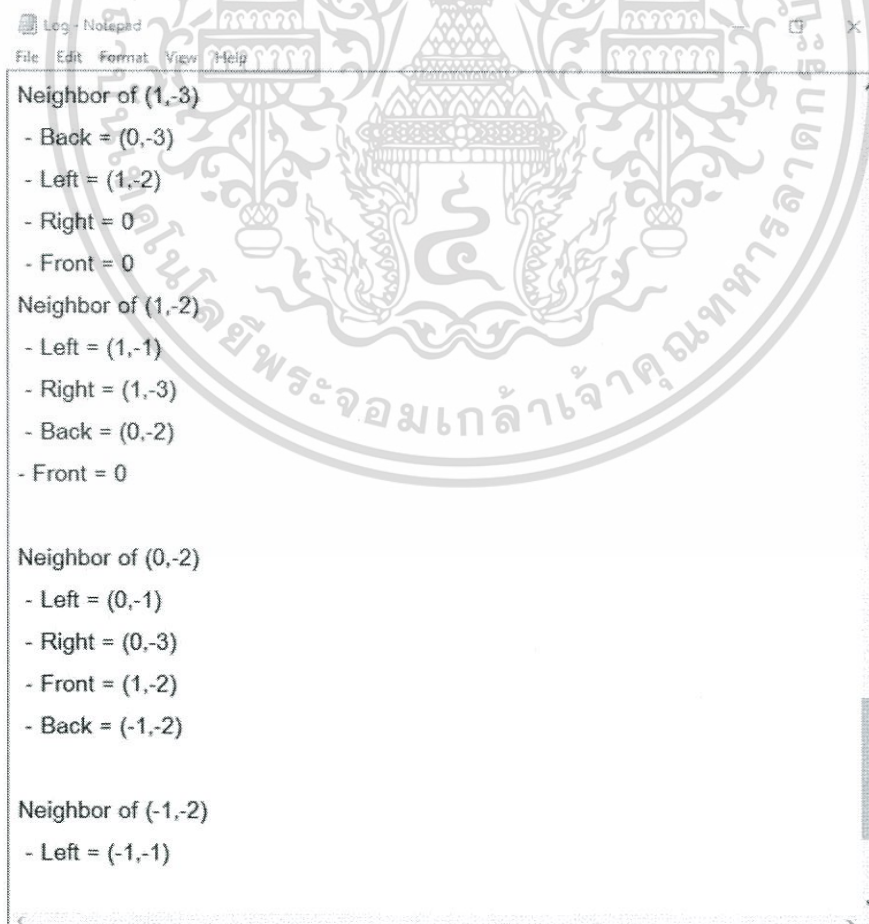
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การนำข้อมูลที่ได้จากหุ่นยนต์มาวิเคราะห์และบันทึกเป็น Log file

เซิร์ฟเวอร์จะทำการนำพิกัดทั้งหมดที่หุ่นยนต์สำรวจได้มาสมมติหา Neighbor Cell ที่เป็นไปได้ทั้งหมดของแต่ละพิกัด แล้วนำพิกัด Neighbor Cell ที่สมมติขึ้น มาเปรียบเทียบกับพิกัดที่หุ่นยนต์สำรวจได้ จะได้เป็นข้อมูลของแต่ละเซลล์ว่าเซลล์นั้นๆ ทั้ง 4 ด้านติดกับเซลล์อะไรบ้าง



รูปที่ 3.12 การนำข้อมูลมาวิเคราะห์



รูปที่ 3.13 การบันทึกเป็น Log file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทำงานของระบบ

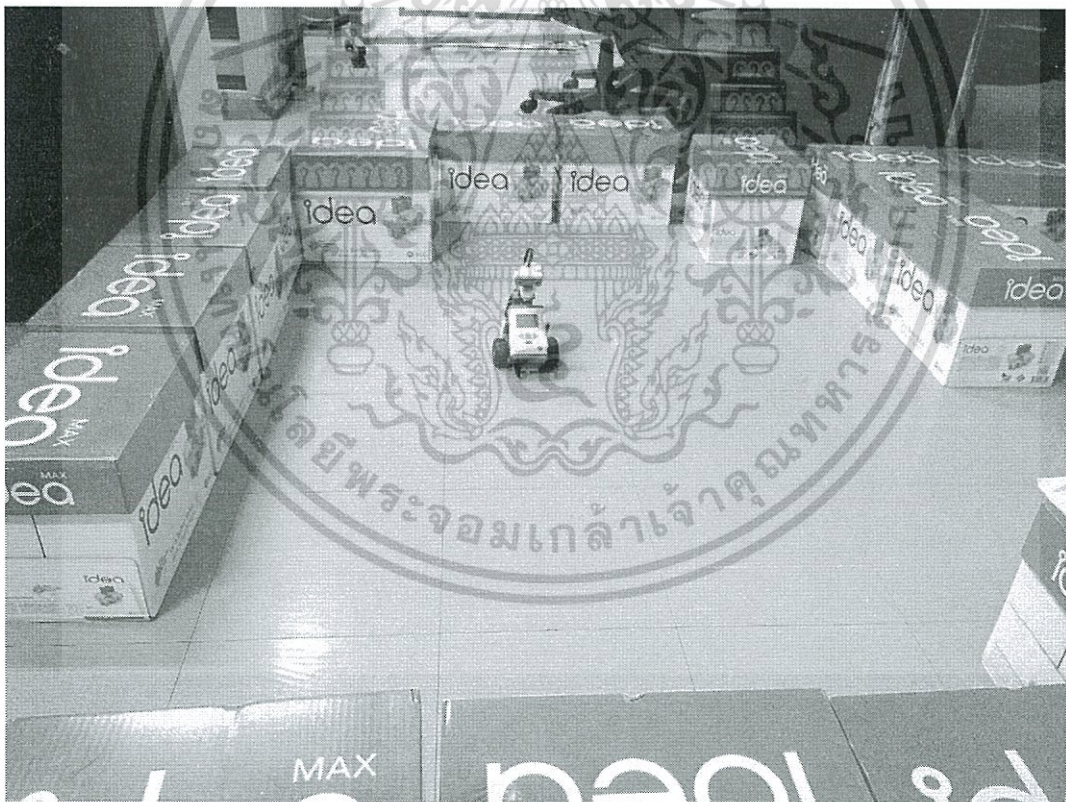
#### 4.1 การสำรวจพื้นที่ของหุ่นยนต์

##### 4.1.1 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์ จะเคลื่อนที่เป็นแบบตาราง เคลื่อนที่ทีละ 1 เซลล์ แต่ละเซลล์จะมีขนาดเท่ากับตัวหุ่นยนต์ หุ่นยนต์จะเลือกเส้นทางการเคลื่อนที่แบบทวนเข็มนาฬิกาเป็นลักษณะก้นหอย และจะไม่เดินซ้ำกับเซลล์ที่เคยสำรวจแล้ว

##### 4.1.2 การเลี้ยวของหุ่นยนต์เมื่อเจอสิ่งกีดขวาง

เมื่อหุ่นยนต์เจอสิ่งกีดขวางหรือไม่มีเส้นทางที่สามารถเลือกเดินต่อไปได้แล้ว หุ่นยนต์จะทำการกลับหลังหันเดินย้อนกลับมายังเซลล์ก่อนหน้า ก่อนที่จะเลือกหาเส้นทางเดินต่อไป



รูปที่ 4.1 การเคลื่อนที่สำรวจของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

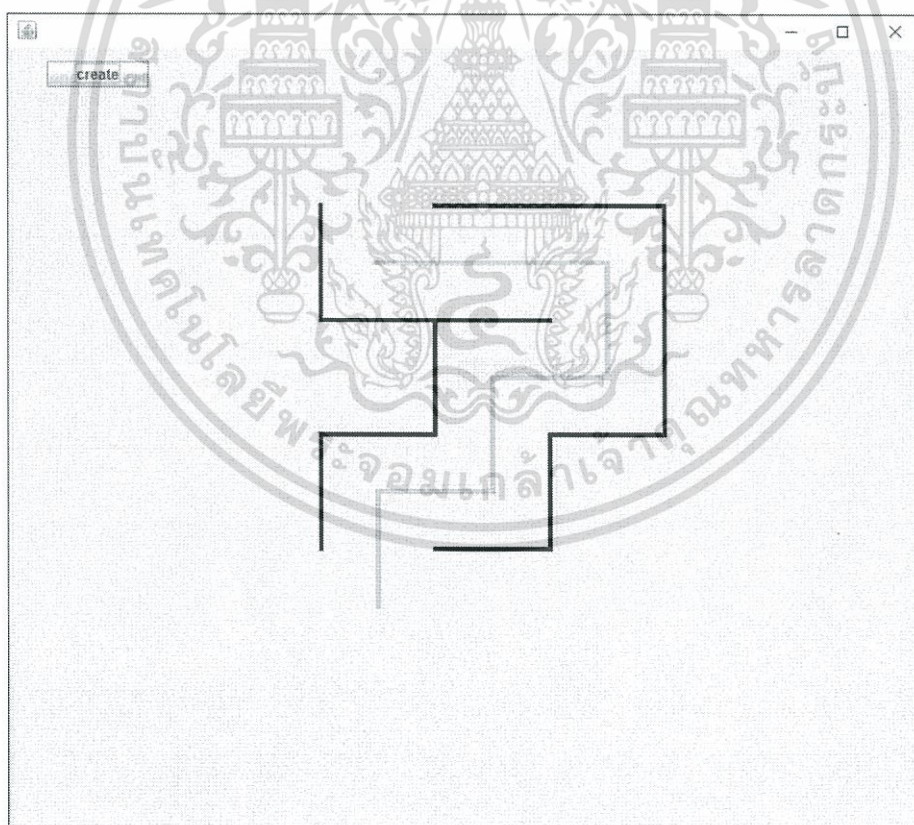
## 4.2 การส่งค่าจากเซ็นเซอร์กลับมาที่เครื่องเซิร์ฟเวอร์

หุ่นยนต์จะทำการส่งค่าที่ตรวจจับได้จากเซ็นเซอร์อัลตราโซนิกส์กลับมาที่เซิร์ฟเวอร์ ซึ่งจะทำการส่งค่าผ่านทางบลูทูธ โดยจะมีการสร้างการเชื่อมต่อระหว่างเซิร์ฟเวอร์กับหุ่นยนต์ก่อนจะเริ่มสำรวจเส้นทาง ซึ่งหุ่นยนต์จะใช้ช่องทางบลูทูธนี้ในการส่งข้อมูลจากเซ็นเซอร์กลับมาที่เซิร์ฟเวอร์ และเซิร์ฟเวอร์จะนำข้อมูลจากเซ็นเซอร์นี้ไปใช้ในการออกแบบพื้นที่ต่อไป

## 4.3 การออกแบบพื้นที่สำรวจ

การออกแบบพื้นที่สำรวจรวมไปถึงสภาพแวดล้อมในพื้นที่ที่สำรวจ โดยจะใช้ค่าที่ได้จากเซ็นเซอร์อัลตราโซนิกส์ของหุ่นยนต์ มาใช้ในการออกแบบพื้นที่ เมื่อหุ่นยนต์เคลื่อนที่ไปในทิศทางใด เครื่องเซิร์ฟเวอร์ก็จะทำการวาดเส้นสีเขียวขึ้นเพื่อแสดงเส้นทางการเคลื่อนที่ของหุ่นยนต์ และวาดเส้นสีดำเมื่อตรวจจับวัตถุหรือสิ่งกีดขวางได้จากเซ็นเซอร์อัลตราโซนิกส์

การจำลองพื้นที่สำรวจนั้นจะให้เส้นสีเขียวแทนการเคลื่อนที่ของหุ่นยนต์ และเส้นสีดำแทนสภาพแวดล้อมที่หุ่นยนต์เคลื่อนที่ผ่านดังรูปที่ 4.2



รูปที่ 4.2 การจำลองพื้นที่จากการเคลื่อนที่ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลโครงการ

### 5.1 สรุปผลโครงการ

โครงการนี้เป็นโครงการศึกษาและพัฒนาอัลกอริทึมในการสำรวจพื้นที่ของหุ่นยนต์ Lego Mindstorms NXT ver.2 รวมไปถึงการจำลองสภาพแวดล้อมของพื้นที่ที่หุ่นยนต์เข้าไปทำการสำรวจในรูปแบบตารางเพื่อใช้ในการสำรวจพื้นที่ที่ต้องการโดยโครงการถูกแบ่งออกเป็น 3 ส่วนคือ

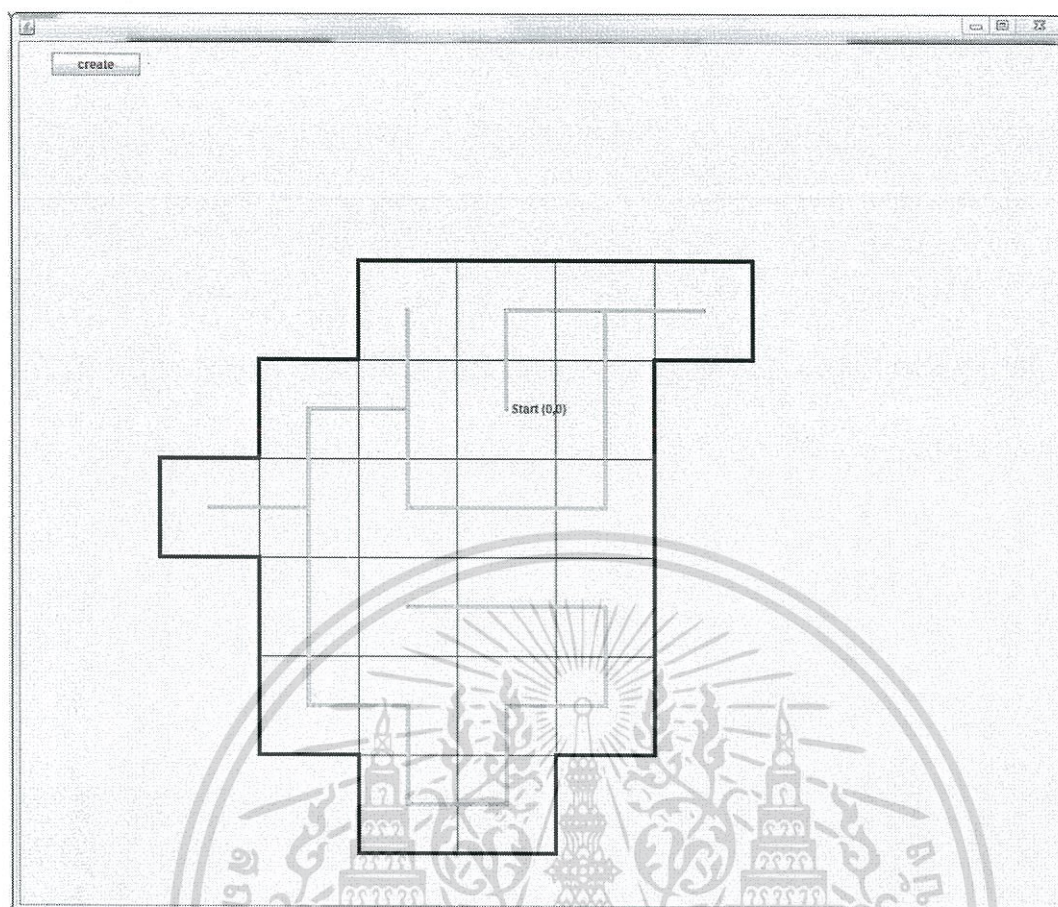
1) การสำรวจพื้นที่ของหุ่นยนต์โดยใช้อัลกอริทึมที่เขียนขึ้นโดยใช้ภาษา Java เพื่อใช้ในการสำรวจพื้นที่ดังรูป 5.1 หลังจากที่หุ่นยนต์ทำการสำรวจพื้นที่เสร็จแล้วจะทำการส่งข้อมูลที่ได้จากการสำรวจมายังหน่วยประมวลผลกลางเพื่อใช้ในการจำลองสภาพพื้นที่ที่หุ่นยนต์เข้าไปทำการสำรวจ



รูปที่ 5.1 สภาพแวดล้อมที่หุ่นยนต์เข้าไปสำรวจ

2) การสร้างแบบจำลองพื้นที่โดยใช้ Java swing GUI ในการสร้างแบบจำลอง โดยโครงสร้างของแบบจำลองถูกสร้างโดยใช้ค่าที่ได้จากการสำรวจของหุ่นยนต์โดยจำได้แบบจำลองดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แบบจำลองพื้นที่ที่หุ่นยนต์เข้าไปสำรวจ

3) นำข้อมูลจากการสำรวจของหุ่นยนต์ มาเก็บเป็น Log File เพื่อใช้เป็นข้อมูลประกอบกับแบบจำลองพื้นที่ที่หุ่นยนต์เข้าไปสำรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Yongguo Mei, Yung-Hsiang Lu, Y. Charlie Hu, C. S. Geotge Lee. **Deployment of Mobile Robots With Energy and Timing Constraints** Reading : School of Electronic. & Computer. Eng., Purdue University, West Lafayette, IN, USA June 2006
- [2] D. Auguelov, D. Koller, E. Parker, and S. Thrun, **Detecting and modeling doors with mobile robots.** in Proc. IEEE Int. Conf. Robot. Autom., 2004, pp. 3777–3784.
- [3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, **Sensor deployment strategy for detection of targets traversing a region.** Mobile Netw. Appl., vol. 8, no. 4, pp. 453–461, 2003.
- [4] A. Howard, M. J. Mataric, and G. S. Sukhatme, **An incremental selfdeployment algorithm for mobile sensor networks.** Auton. Robots, vol. 13, no. 2, pp. 113–126, 2002.
- [5] B. C. Kuo and J. Tal, Eds., **DC Motors and Control Systems.** New York: SRL Publishing, 1978.
- [6] C. J. Taylor and D. J. Kriegman, **Vision-based motion planning and exploration algorithms for mobile robots.** IEEE Trans. Robot. Autom., vol. 14, no. 3, pp. 417–426, Jun. 1998.
- [7] Y. Zou and K. Chakrabarty, **Sensor deployment and target localization based on virtual forces.** in Proc. INFOCOM, 2003, pp. 1293–1303.

# การค้นหาเส้นทางอย่างมีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่

ณพวิทย์ เตชเรืองรัมย์, เมธี คำอยู่, และ ลภัส ประดิษฐ์ทัศนีย์

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

noppawitmax@gmail.com, matee1993.k@gmail.com

## บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้นำเสนอการพัฒนาอัลกอริทึมที่ใช้ในการหาเส้นทางของหุ่นยนต์รวมไปถึงการสื่อสารระหว่างหุ่นยนต์เพื่อประเมินสถานการณ์ในขณะนั้น ในการสำรวจเส้นทางของหุ่นยนต์นั้นจะใช้เซ็นเซอร์ในการตรวจจับสภาพแวดล้อมรอบตัวโดยประเมินจากสิ่งที่อยู่ข้างหน้า เพื่อนำมาประมวลผลและการหาเส้นทางที่ใช้ในการสำรวจในสภาพแวดล้อมที่ต้องการ และส่งข้อมูลสถานการณ์ปัจจุบันให้กับหุ่นยนต์ตัวอื่น เพื่อให้หุ่นยนต์ตัวอื่นใช้ประมวลผลต่อไป จากนั้นส่งข้อมูลต่อมายังหน่วยประมวลผลเพื่อดูสถานะของหุ่นยนต์แต่ละตัว ในส่วนของอัลกอริทึมจะตัดแปลงมาจากอัลกอริทึมที่มีอยู่เพื่อให้เหมาะกับสถานการณ์

**คำสำคัญ** – หุ่นยนต์ (robot) , อัลกอริทึม(Algorithm), เส้นทาง(Path)

## 1. บทนำ

หุ่นยนต์เป็นสิ่งที่มนุษย์คิดค้นและสร้างขึ้นเพื่ออำนวยความสะดวกให้กับมนุษย์และใช้แทนแรงงานมนุษย์หรือใช้ทำงานที่มนุษย์ไม่สามารถทำได้ เช่นการสำรวจสภาพแวดล้อมที่มนุษย์ไม่สามารถเข้าไปได้จึงจำเป็นต้องใช้หุ่นยนต์เข้าไปทำการสำรวจพื้นที่แทน โดยมนุษย์จะเป็นคนเขียนคำสั่งต่างๆเพื่อให้หุ่นยนต์สามารถทำงานได้ตามที่ต้องการในปัจจุบันได้มีการพัฒนาหุ่นยนต์ในรูปแบบต่างๆขึ้นมามากมาย เพื่อให้เหมาะกับงานในรูปแบบต่างๆไป โดยทั่วไปแล้วหุ่นยนต์จะแบ่งได้เป็น 2 องค์ประกอบหลัก โดยองค์ประกอบแรกคือส่วนกายภาพที่เป็นชิ้นส่วนซึ่งนำมาประกอบเป็นตัวหุ่นยนต์เพื่อให้เหมาะกับงานที่หุ่นยนต์ต้องทำ และองค์ประกอบที่สองคือส่วนของโปรแกรมหรือชุดคำสั่งที่ใช้ในการสั่งงานหุ่นยนต์ ซึ่งองค์ประกอบเหล่านี้มีความสำคัญมากต่อประสิทธิภาพในการทำงานของหุ่นยนต์ เพราะ อัลกอริทึมที่ทำการโปรแกรมเข้าไปจะทำหน้าที่เสมือนสมองที่สั่งให้หุ่นยนต์ทำงานตามเป้าหมายที่มนุษย์ต้องการ และงานแต่ละประเภทก็ต้องการอัลกอริทึมที่แตกต่างกัน เมื่อหุ่นยนต์สามารถทำงานได้ตามอัลกอริทึมที่เราโปรแกรมเข้าไปหุ่นยนต์จะสามารถค้นหาเส้นทางที่ใช้ในการสำรวจได้ และอัลกอริทึมที่ใช้อย่างมีประสิทธิภาพก็ช่วยแก้ปัญหา รวมไปถึงข้อจำกัดต่างๆในการสำรวจพื้นที่เช่น แสง สัญญาณ GPS เมื่อสามารถแก้ปัญหาเหล่านี้ได้ หุ่นยนต์จึงสามารถทำงานได้อย่างมีประสิทธิภาพ การที่หุ่นยนต์สามารถทำงานได้

อย่างมีประสิทธิภาพนั้น จะทำให้เราสามารถลดระยะเวลาในการสำรวจ ประหยัดพลังงาน รวมไปถึงความถูกต้องที่ได้จากการสำรวจ ทำให้เราสามารถจำลองสภาพพื้นที่ที่หุ่นยนต์เข้าไปสำรวจได้สมบูรณ์ และยังสามารถนำแบบจำลองพื้นที่นี้ไปประโยชน์ในการสำรวจพื้นที่แบบต่างๆได้อีกด้วย

## 2. ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

### 2.1 Lego Mindstorms NXT ver.2



รูปที่ 1. หุ่นยนต์ Lego Mindstorms NXT ver.2

Lego Mindstorms NXT ver.2 เป็น หุ่นยนต์ของ Lego ที่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานได้โดยโปรแกรมจะเก็บไว้ในส่วนประมวลผลของหุ่นยนต์ จากนั้นระบบจะสั่งการต่อไปโมดูลต่างๆ ตามคำสั่งที่เราเขียนลงไป

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ภาษา Java



### รูปที่ 2. ภาษา Java

JAVA คือภาษาโปรแกรมเชิงวัตถุ เป็นภาษาที่พัฒนาเพื่อมาแทน C++ โดยภาษา Java จะใช้หลักการเขียนแบบ object-oriented-programming โปรแกรมจะเขียนภายใต้คลาส ภายใต้คลาสจะเก็บ method โดยเราจะใช้ Java ในการเขียนชุดคำสั่งเพื่อควบคุมการทำงานและสั่งงานหุ่นยนต์ไปสำรวจในพื้นที่นั้นๆ

### 2.3 การสำรวจพื้นที่แบบ Spanning tree โดยใช้ Spiral-STC

Gabriely และ Rimon ได้เสนออัลกอริทึม Spiral-STC คือการเคลื่อนที่แบบเส้นของหุ่นยนต์โดยแบบพื้นที่การทำงานเป็นตารางตามแบบ Spiral path โดยระบบ Spiral path จะสร้างในรูปแบบ Spanning tree โดยเป็นส่วนหนึ่งของตารางที่หุ่นยนต์สร้างขึ้นโดยใช้เซ็นเซอร์หุ่นยนต์สามารถจะทำงานได้ครอบคลุมพื้นที่ทุกเซลล์ในการทำงานครั้งเดียวและครอบคลุมทุกเส้นทาง Gabriely และ Rimon ได้ตรวจสอบแบบจำลองของพวกเขา อัลกอริทึมมีการทำงานดังนี้ แบ่งเซลล์ออกเป็น 2 ขนาด. เซลล์ใหญ่(เมกาเซลล์) จะแบ่งออกเป็น 4 เซลล์เล็กโดยมีขนาดเท่ากับขนาดของหุ่นยนต์ โดยการทำงานให้ครอบคลุมนั้น หุ่นยนต์ทำงานตามขั้นตอน เริ่มที่เซลล์ปัจจุบัน หุ่นยนต์เลือกเส้นทางการเดินโดยการเลือกเมกาเซลล์ใหม่ที่สามารถเคลื่อนที่ไปได้ในรูปแบบทวนเข็มนาฬิกาจากนั้น Spanning tree ใหม่จะแผ่ขยายต่อไปยัง เมกาเซลล์ใหม่ๆ อัลกอริทึมนี้เป็นแบบการทำงานซ้ำไปมาการทำงานของอัลกอริทึมนี้จะหยุดก็ต่อเมื่อไม่มีเซลล์ใหม่เหลืออยู่แล้ว (เมกาเซลล์ที่ถือว่าไม่ใช่เซลล์ใหม่คือหุ่นยนต์ได้ทำการเคลื่อนที่ผ่านเซลล์เล็กด้านในหมดแล้ว) นี่คือผลของการทำงานซ้ำๆ หุ่นยนต์จะเคลื่อนที่ไปตาม Spanning tree จนสุดปลายด้านหนึ่งของ Spanning tree จะสังเกตได้ว่าเมื่อหุ่นยนต์ทำการสำรวจเสร็จสมบูรณ์หุ่นยนต์จะกลับมาอยู่ที่เซลล์

เริ่มต้น แล้วทำการจัดเก็บข้อมูล ในทางกลับกันหุ่นยนต์จะไม่ทำการเคลื่อนที่ไปซ้ำเซลล์เล็กที่เคยเดินผ่านมาแล้ว ขั้นตอนการสำรวจ กำหนดให้

- s คือเซลล์เริ่มต้น
- x คือเซลล์ปัจจุบัน
- w คือเซลล์ที่อยู่ก่อนหน้าในแบบ spanning tree
- ฟังก์ชัน STC เป็นฟังก์ชันแบบเรียกใช้งานซ้ำ

- ขั้นตอนที่ 1 กำหนดให้เซลล์ปัจจุบันคือ x
- ขั้นตอนที่ 2 เมื่อเซลล์ x ยังมีเซลล์เพื่อนบ้านที่สามารถไปได้
- ขั้นตอนที่ 2.1 หาเซลล์เพื่อนบ้านที่อยู่ในด้านทวนเข็มนาฬิกาจาก w คือเซลล์ y
- ขั้นตอนที่ 2.2 สร้าง spanning tree จาก x ไป y
- ขั้นตอนที่ 2.3 เคลื่อนที่จากเซลล์ x ไปหา y
- ขั้นตอนที่ 2.4 รีเซ็ตอัลกอริทึม STC ไปเรื่อย ๆ เมื่อจบ loop
- ขั้นตอนที่ 3 ถ้า x เป็นคนละเซลล์กับ S ถอยหลัง x ไปยังช่องเล็กของ w
- ขั้นตอนที่ 4 จบการทำงาน (หุ่นยนต์กลับมาอย่างจุดเริ่มต้น)

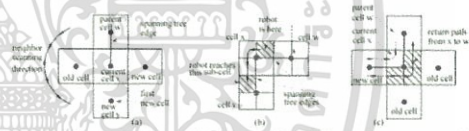


Figure 3: (a) Counter-clockwise scan of four neighbors. (b) A move from z to a new cell y. (c) A return from z to a parent cell w.

### รูปที่ 3. รูปประกอบการทำงานของอัลกอริทึม STC บทที่ 3 การวิเคราะห์และออกแบบระบบ

#### 3.1 อัลกอริทึมการเดินสำรวจขาไปของหุ่นยนต์

หุ่นยนต์จะเริ่มทำการสำรวจเส้นทางขาไปโดยเรียกใช้ฟังก์ชันเก็บพิกัดตำแหน่งปัจจุบันลงในลิสต์ จากนั้นทำการตรวจจับสิ่งกีดขวางทางด้านขวา ด้านหน้า และด้านซ้าย หากระยะที่ตรวจจับได้จากทางด้านขวามีค่ามากกว่า 40 ก็จะนำเอาพิกัดของเซลล์ทางด้านขวามาตรวจสอบการเดินซ้ำของหุ่นยนต์ หากเซลล์นั้นหุ่นยนต์ยังไม่เคยไปสำรวจมาก่อน หุ่นยนต์ก็จะเลือกเซลล์นั้นเป็นเส้นทางที่จะเดินต่อไป แต่ถ้าหากระยะที่ตรวจจับได้จากทางด้านขวามีค่าน้อยกว่า 40 หรือเซลล์ทางด้านขวาเคยไปทำการสำรวจมาแล้ว ก็จะเลือกทางเดินทางด้านอื่นต่อไป โดยการเลือกเส้นทางของหุ่นยนต์จะเลือกเดินทางด้านขวา หน้า และซ้ายตามลำดับเป็นรูปแบบทวนเข็มนาฬิกา

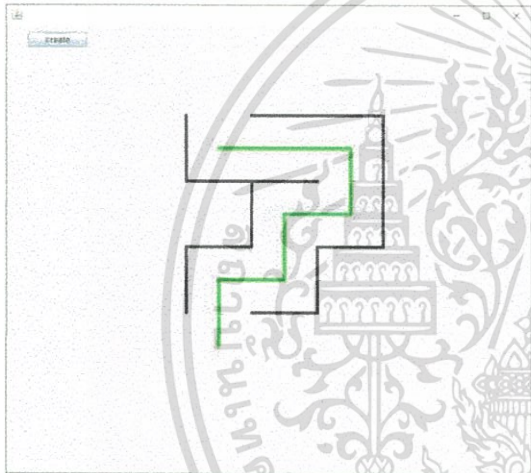
#### 3.2 อัลกอริทึมการเดินสำรวจขากลับของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อหุ่นยนต์เดินสำรวจจนไม่มีเส้นทางให้เลือกเดินต่อ หุ่นยนต์จะเริ่มเดินย้อนกลับตามเส้นทางเดิมที่เดินมาถึงเซลล์ ระหว่างการเดินกลับก็จะค้นหาเส้นทางที่ยังไม่เคยเดินไปสำรวจ หากเจอเส้นทางใหม่ก็จะเลือกไปสำรวจตามเส้นทางใหม่โดยใช้อัลกอริทึมเดียวกับการเดินสำรวจมา หากไม่เจอเส้นทางใหม่แล้วก็จะด้านย้อนกลับมาจนถึงจุดเริ่มต้นซึ่งถือว่าสิ้นสุดการสำรวจ

### 3.3 การออกแบบฝั่งเซิร์ฟเวอร์

การออกแบบโปรแกรมฝั่งเซิร์ฟเวอร์โดยจะออกแบบให้เซิร์ฟเวอร์รับค่าพิกัด และข้อมูลของแต่ละพิกัดที่ได้จากการสำรวจมาจากหุ่นยนต์ นำมาสร้างเป็นแผนที่จำลองโดยการรับค่าจากหุ่นยนต์นั้นจะใช้ LeJOS PC Library ส่วนการสร้างแผนที่จำลองจะใช้ JAVA Swing GUI



รูปที่ 4. การใช้ Java swing GUI ในการจำลองแผนที่

## บทที่ 4 การนำระบบไปใช้งาน

### 4.1 การสำรวจพื้นที่ของหุ่นยนต์

#### 4.1.1 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์ จะเคลื่อนที่เป็นแบบตาราง เคลื่อนที่ทีละ 1 เซลล์ แต่ละเซลล์จะมีขนาดเท่ากับตัวหุ่นยนต์ หุ่นยนต์จะเลือกเส้นทางการเคลื่อนที่แบบทวนเข็มนาฬิกาว่าเป็นลักษณะกันหอย และจะไม่เดินซ้ำกับเซลล์ที่เคยสำรวจแล้ว

#### 4.1.2 การเลี้ยวของหุ่นยนต์เมื่อเจอสิ่งกีดขวาง

เมื่อหุ่นยนต์เจอสิ่งกีดขวางหรือไม่มีเส้นทางที่สามารถเลือกเดินต่อไปได้แล้วหุ่นยนต์จะทำการกลับหลังหันเดินย้อนกลับมายังเซลล์ก่อนหน้าก่อนที่จะเลือกหาเส้นทางเดินต่อไป



รูปที่ 5. การเคลื่อนที่สำรวจของหุ่นยนต์

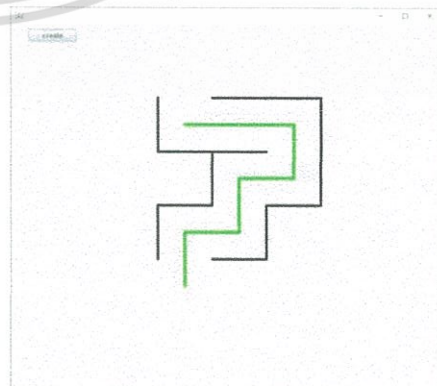
### 4.2 การส่งค่าจากเซ็นเซอร์กลับมาที่เครื่องเซิร์ฟเวอร์

หุ่นยนต์จะทำการส่งค่าที่ตรวจจับได้จากเซ็นเซอร์อัลตราโซนิกส่งกลับมาที่เซิร์ฟเวอร์ ซึ่งจะทำให้การส่งค่าผ่านทางบลูทูธ โดยจะมีการสร้างการเชื่อมต่อระหว่างเซิร์ฟเวอร์กับหุ่นยนต์ก่อนจะเริ่มสำรวจเส้นทาง ซึ่งหุ่นยนต์จะใช้ช่องทางบลูทูธนี้ในการส่งข้อมูลจากเซ็นเซอร์กลับมาที่เซิร์ฟเวอร์ และเซิร์ฟเวอร์จะนำข้อมูลจากเซ็นเซอร์นี้ไปใช้ในการออกแบบพื้นที่ต่อไป

### 4.3 การออกแบบพื้นที่สำรวจ

การออกแบบพื้นที่สำรวจรวมไปถึงสภาพแวดล้อมในพื้นที่ที่สำรวจ โดยจะใช้ค่าที่ได้จากเซ็นเซอร์อัลตราโซนิกของหุ่นยนต์ มาใช้ในการออกแบบพื้นที่ เมื่อหุ่นยนต์เคลื่อนที่ไปในทิศทางใด เครื่องเซิร์ฟเวอร์ก็จะทำการวาดเส้นสีเขียวขึ้นเพื่อแสดงเส้นทางการเคลื่อนที่ของหุ่นยนต์ และวาดเส้นสีดำเมื่อตรวจจับวัตถุหรือสิ่งกีดขวางได้จากเซ็นเซอร์อัลตราโซนิก

การจำลองพื้นที่สำรวจนั้นจะให้เส้นสีเขียวแทนการเคลื่อนที่ของหุ่นยนต์ และเส้นสีดำแทนสภาพแวดล้อมที่หุ่นยนต์เคลื่อนที่ผ่านดังรูปที่ 5



รูปที่ 6. การจำลองพื้นที่จากการเคลื่อนที่ของหุ่นยนต์

## บทที่ 5 สรุปผลโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 5.1 สรุปผลโครงการ ญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

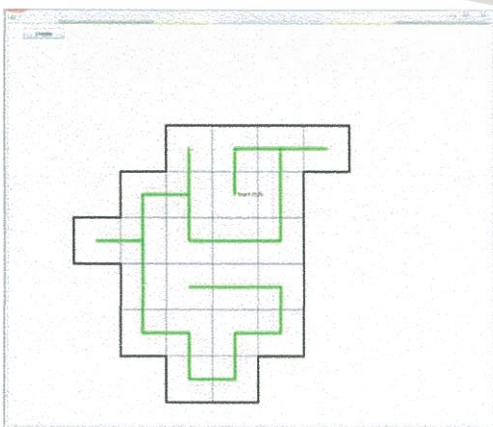
โครงการนี้เป็นโครงการศึกษาและพัฒนา อัลกอริทึมในการสำรวจพื้นที่ของหุ่นยนต์ Lego Mindstorms NXT ver.2 รวมไปถึงการจำลอง สภาพแวดล้อมของพื้นที่ที่หุ่นยนต์เข้าไปทำการสำรวจ ในรูปแบบตารางเพื่อใช้ในการสำรวจพื้นที่ที่ต้องการโดย โครงการถูกแบ่ง ออกเป็น 2 ส่วนคือ

1) การสำรวจพื้นที่ของหุ่นยนต์โดยใช้อัลกอริทึม ที่เขียนขึ้นโดยใช้ภาษา Java เพื่อใช้ในการสำรวจพื้นที่ดังรูป 5.1 หลังจากที่หุ่นยนต์ทำการสำรวจพื้นที่เสร็จแล้วจะทำการส่งข้อมูลที่ได้จากการสำรวจมายังหน่วยประมวลผล กลางเพื่อใช้ในการจำลองสภาพพื้นที่ที่หุ่นยนต์เข้าไปทำการสำรวจ



รูปที่ 7. สภาพแวดล้อมที่หุ่นยนต์เข้าไปสำรวจ

2) การสร้างแบบจำลองพื้นที่โดยใช้ Java swing GUI ในการสร้างแบบจำลอง โดยโครงสร้างของแบบจำลอง ถูกสร้างโดยใช้ค่าที่ได้จากการสำรวจของหุ่นยนต์โดยจำได้ แบบจำลองดังรูปที่ 7



รูปที่ 8. แบบจำลองพื้นที่ที่หุ่นยนต์เข้าไปสำรวจ

## 6. กิตติกรรมประกาศ

ขอขอบคุณดร. ลภัส ประดิษฐ์ทัศนีย์ ที่เป็นผู้คิดค้นให้ ความรู้เกี่ยวกับโครงการนี้ ทำให้กลุ่มของเราได้มีโอกาส ศึกษาการทำงานของหุ่นยนต์ รวมไปถึงการพัฒนา อัลกอริทึมที่ใช้ในการสำรวจเส้นทางของหุ่นยนต์ อีกทั้ง ยังให้ความช่วยเหลือในด้านต่าง ๆ

## เอกสารอ้างอิง

- [1] Yongguo Mei, Yung-Hsiang Lu, Y. Charlie Hu, C. S. Geotge Lee. **Deployment of Mobile Robots With Energy and Timing Constraints** Reading : School of Electronic. & Computer. Eng., Purdue University, West Lafayette, IN, USA June 2006
- [2] D. Auguelov, D. Koller, E. Parker, and S. Thrun, **Detecting and modeling doors with mobile robots.** in Proc. IEEE Int. Conf. Robot. Autom., 2004, pp. 3777–3784.
- [3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, **Sensor deployment strategy for detection of targets traversing a region.** Mobile Netw. Appl., vol. 8, no. 4, pp. 453–461, 2003.
- [4] A. Howard, M. J. Mataric, and G. S. Sukhatme, **An incremental selfdeployment algorithm for mobile sensor networks.** Auton. Robots, vol. 13, no. 2, pp. 113–126, 2002.
- [5] B. C. Kuo and J. Tal, Eds., **DC Motors and Control Systems.** New York: SRL Publishing, 1978.
- [6] C. J. Taylor and D. J. Kriegman, **Vision-based motion planning and exploration algorithms for mobile robots.** IEEE Trans. Robot. Autom., vol. 14, no. 3, pp. 417–426, Jun. 1998.
- [7] Y. Zou and K. Chakrabarty, **Sensor deployment and target localization based on virtual forces.** in Proc. INFOCOM, 2003, pp. 1293–1303.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้