สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การค้นหาเส้นทางและหลบสิ่งกีดขวางสำหรับหุ่นยนต์โดยใช้การประมวลผลภาพ

Path Finding for Robots using Image Processing

T046437

โดย

นางสาวผ่องพรรณ เพียรภัทรพงศ์

นางสาวพีรดา คงชาญศิริ

เลขหมู่......
เลขทะเบียน 46437
วัน, เดือน, ปี 1 เม.ย. 2545

.b......
.i......

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การค้นหาเส้นทางและหลบสิ่งกีดขวางสำหรับหุ่นยนต์โดยการประมวลผลภาพ
TITLE	**Path Finding for Robots using Image Processing**
โดย	นางสาวผ่องพรรณ เพียรภัทรพงศ์ รหัสประจำตัว 41014282
	นางสาวพีรดา คงชาญศิริ รหัสประจำตัว 41014313
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ผศ.ดร. ปิติเขต สู้รักษา
	อาจารย์ บุญย์ชนะ ภู่ระหงษ์
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

....

(อาจารย์บุญย์ชนะ ภู่ระหงษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

....

(อาจารย์ ปิติเขต สู้รักษา)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การค้นหาเส้นทางและหลบสิ่งกีดขวางสำหรับหุ่นยนต์โดยการประมวลผลภาพ
นักศึกษา	นางสาวผ่องพรรณ เพียรภัทรพงศ์ รหัสประจำตัว 41014282
	นางสาวพีรดา คงชาญศิริ รหัสประจำตัว 41014313
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ผศ.ดร. ปิติเขต สู้รักษา
	อาจารย์ บุณย์ชนะ ภู่ระหงษ์
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

การค้นหาเส้นทางสำหรับหุ่นยนต์ให้เดินไปยังเป้าหมายที่ผ่านมานั้นมีหลากหลายวิธีการและเป็นเรื่องที่ได้รับความสนใจเป็นอย่างมาก เนื่องจากสามารถนำไปประยุกต์ใช้งานได้หลายด้าน สำหรับปริญญานิพนธ์นี้เป็นการนำทฤษฎีทางด้านการประมวลผลภาพมาใช้เนื่องจากมีความแม่นยำมากกว่า เช่นถ้าใช้เป็นเซ็นเซอร์จะไม่สามารถระบุตำแหน่งที่แน่นอนของสิ่งกีดขวางได้ หรือการใช้รังสีอัลตร้าโซนิคจะมีข้อเสียเปรียบในการแยกแยะตำแหน่งสิ่งกีดขวางหลายๆชิ้น แต่การวิเคราะห์ด้วยภาพจะหาพิกัดที่แน่นอนของวัตถุได้ เราจึงใช้การวิเคราะห์สีในภาพที่รับมาจากกล้องวิดีโอแบบเรียลไทม์และทำการปรับเฉดสีและแสงของภาพเพื่อความสะดวกในการวิเคราะห์ภาพ และเพื่อเป็นการเพิ่มประสิทธิภาพจึงได้นำอัลกอริธึมในการจัดกลุ่ม (Clustering) แบบเคมีนมาช่วยในการลดสัญญาณรบกวนในภาพ (noise) เพื่อทำการตรวจจับขอบของวัตถุ และใช้อัลกอริธึมการค้นหา (search algorithm) ในการค้นหาเส้นทางแบบคอนทัวร์เทรสซิ่ง (Contour Tracing) เพื่อหลบหลีกสิ่งกีดขวาง (Path finding) และทำการควบคุมหุ่นยนต์ด้วยระบบไร้สายให้เดินไปในทิศทางที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT TITLE	**PATH FINDING FOR ROBOTS USING IMAGE PROCESSING**
STUDENT	**MS. Pongpan Pienputtarapong No.41014282**
	MS. Peerada Kongchansiri No.41014313
ADVISORS	**Asst. Prof. Dr. Pitikhate Sooraksa**
	Mr. Boonchana Purahong
COURSE	**Bachelor of Information Engineering**
DEPARTMENT	**Information Engineering**
YEAR	**2001**

ABSTRACT

Path finding for robots to achieve goal has been implemented by many algorithms. It is considerable interesting because of its utilities in many applications. Using sensors or ultrasonic to detect obstacles are limited in specifying exactly coordinates. Image processing algorithms are used in this thesis to increase performance. The approach is based on the HSV color model to analyze the video frame at almost real-time which we can adjust the features of picture. The algorithm of K-mean clustering is derived for noise reduction for high quality of edge detection. We choose Contour Tracing search algorithm. Although it looks simple but it guarantees the result and it generates path that easy to control the robot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษาโครงงานทั้งอาจารย์บุญย์ชนะ ภู่ระหงษ์และผศ.ดร.ปิติเขต สู้รักษาเป็นอย่างสูงและขอขอบคุณอาจารย์ภุชงค์ หงษ์สุวรรณที่คอยให้คำปรึกษาในด้านแนวทางการทำงาน และที่ต้องขอบคุณมากที่สุดคือพี่เด้ยที่คอยช่วยเหลือในทุกๆด้าน

ขอขอบคุณพ่อคุณแม่และอาจารย์ทุกท่านที่ให้การสนับสนุนเสมอมา และเพื่อนๆพี่ๆที่อยู่ร่วมในการทำงานด้วยกันทุกคน

ผ่องพรรณ เพียรภัทรพงศ์

พีรดา คงชาญศิริ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	
สารบัญรูป	
สารบัญตาราง	
เอกสารอ้างอิง	
บทที่1 บทนำ	**1**
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการทำงาน	2
บทที่ 2 ทฤษฎี	**3**
2.1 ฮาร์ดแวร์ (Hardware)	3
2.1.1 คอมพิวเตอร์ส่วนบุคคล (Personal Computer)	3
2.1.2 กล้องดิจิตอล (Digital Camera)	3
2.1.3 บอร์ดทดลองรุ่น ET-AVR V1.0 (Single Board)	3
2.1.4 ชุดรับส่งคลื่นวิทยุ 49 MHz.	15
2.1.5 ตัวรถ (Car Body)	16
2.1.6 วงจรสวิตช์	16
2.1.7 วงจรรวม	17
2.1.8 การเชื่อมต่อด้วยพอร์ตอนุกรม RS-232	18
2.2 ซอฟท์แวร์ (Software)	**19**
2.2.1 โมเดลของสีชนิดเอชเอสวี (HSV Color Model)	19
2.2.2 การแบ่งกลุ่ม (Clustering) แบบเคมีน (K-mean)	21
2.2.3 การค้นหาเส้นทาง (Path Finding)	23
2.2.4 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรม (I/O Serial Port)	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบ 37
3.1 อัลกอริธึมการวิเคราะห์สีของภาพโดยใช้ทฤษฎีเอชเอสวี (HSV) 37
3.2 อัลกอริธึมในการสแกนหาตำแหน่งสิ่งกีดขวางและตำแหน่งรถ 37
3.3 อัลกอริธึมในการจัดกลุ่ม (Clustering) 37
3.4 การค้นหาเส้นทาง (Path Finding) ด้วยอัลกอริธึมคอนทัวร์เทรสซิ่ง (Contour Tracing) 38
3.5 การค้นหาเส้นทาง (Path Finding) ด้วยอัลกอริธึมคอนทัวร์ 38
เทรสซิ่ง (Contour Tracing) และเอสตาร์ (A-star)
3.6 อัลกอริธึมในการควบคุมรถ 39
บทที่ 4 การทดลองและผลลัพธ์ 46
4.1 การทดลอง 46
4.2 ผลการทดลอง 46
4.3 สรุปผลการทดลอง 52
บทที่ 5 กรณีศึกษา (Review Literature) 53
5.1 Condensation Algorithm 53
5.2 Search Algorithm 56

สารบัญภาพ

รูปที่ 2.1 แสดงโครงสร้างของบอร์ด ET-AVR V1.0 4
รูปที่ 2.2 แสดงไดอะแกรมการส่งข้อมูลแบบ UART 6
รูปที่ 2.3 แสดงไดอะแกรมการรับข้อมูลของ UART 7
รูปที่ 2.4 แสดงโครงสร้างของพอร์ต A (Pins PA0 – PA7) 15
รูปที่ 2.5 รูปวงจรรับส่ง 15
รูปที่ 2.6 ชุดเฟืองรถทามิย่า 16
รูปที่ 2.7 วงจรสวิตช์ 16
รูปที่ 2.8 วงจรภาคส่ง 17
รูปที่ 2.9 วงจรภาครับ 17
รูปที่ 2.10 รายละเอียดของคอมพอร์ต (COM1, COM2) 18
รูปที่ 2.11 แสดงกรวยซึ่งแสดงความสัมพันธ์ของสีในระบบเอชเอสวี 19
รูปที่ 2.12 แสดงภาพภาคตัดขวางของกรวยเอชเอสวี 20
รูปที่ 2.13 แสดงวงล้อของสี (Color Wheel) 20
รูปที่ 2.14 การค้นหาเส้นทางจากจุดหนึ่งไปอีกจุดหนึ่ง 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 แสดงการทำงานของอัลกอริธึมในการชนและถอย 24
รูปที่ 2.16 การค้นหาเส้นทางของวิธี Contour tracing 24
รูปที่ 2.17 เส้นทางการหลบหลีกสิ่งกีดขวาง 25
รูปที่ 2.18 เส้นทางเชื่อมต่อถึงกันเป็นเน็ตเวิร์ค 26
รูปที่ 2.19 เส้นทางที่เชื่อมโหนดโดยใช้จุดร่วมกัน(Waypoints) 26
รูปที่ 2.20 แสดงภาพโครงข่ายของกราฟทั่วไป 27
รูปที่ 2.21 แสดงการใช้ Grid เชื่อมต่อแต่ละโหนด 27
รูปที่ 2.22 แสดงภาพการแตกโหนดของ Breadth-first search 28
รูปที่ 2.23 แสดงการแตกโหนดจาดจุดเริ่มต้นและจุดเป้าหมาย 29
รูปที่ 2.24 การค้นหาเส้นทางแบบ Depth-first search 30
รูปที่ 2.25 แสดงแผนที่ระยะทาง(กิโลเมตร) และระยะกระจัดไปเมือง Bucharest 31
รูปที่ 2.26 แสดงแผนผังTreeของ A* search โดย f = g + h 31
รูปที่ 2.27 ตัวอย่างการEnable port ให้ใช้งานได้ 34
รูปที่ 2.28 ตัวอย่างการ Disable Port 34
รูปที่ 2.29 แสดงการเซ็ตค่าพร็อพเพอร์ตี้ต่างๆในการติดต่อกับพอร์ตอนุกรม 35
รูปที่ 2.30 แสดงการใช้พร็อพเพอร์ตี้อินพุทและเอาท์พุท 36
รูปที่ 3.1 แสดงการควบคุมล้อรถ 45
รูปที่ 3.2 แสดงแผนผังการทำงานของโปรแกรม 46
รูปที่ 3.3 แสดงแผนผังการทำงานของโปรแกรมส่วนค้นหาเส้นทาง (Path finding) 47
รูปที่ 3.4 แสดงแผนผังการทำงานของโปรแกรมการค้นหาเส้นทางแบบ 48
คอนทัวร์เทรสซิ่งและเอสตาร์
รูปที่ 3.5 แสดงแผนผังการทำงานของโปรแกรมส่วนตัดขอบ (Edge Detection) 49
รูปที่ 4.1 ตัวอย่างผลการทดลองที่ 1 47
รูปที่ 4.2 ตัวอย่างผลการทดลองที่ 2 48
รูปที่ 4.3 ตัวอย่างผลการทดลองที่ 3 49
รูปที่ 4.4 ตัวอย่างผลการทดลองที่ 4 50
รูปที่ 4.5 ตัวอย่างผลการทดลองที่ 5 51
รูปที่ 5.1 ตัวอย่างการติดตามการเคลื่อนไหวของมือบนพื้นหลังที่ยุ่งเหยิง 53
รูปที่ 5.2 ตัวอย่างการติดตามการเคลื่อนไหวของใบไม้บนพื้นหลังที่มีลักษณะคล้ายกัน 53
รูปที่ 5.3 ตัวอย่างการติดตามการเคลื่อนไหวของหัวเด็กบนพื้นหลังที่ยุ่งเหยิง 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 2.1 แสดงการกำหนดค่าอัตราเร็วในการรับส่ง (Baud Rate) 12

ตารางที่ 2.2 DDAn Effects on PORTA Pins 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันการศึกษาเรื่องการประมวลผลภาพ (Image Processing) ได้รับความสนใจเป็นอย่างมาก เนื่องจากสามารถนำมาใช้ประโยชน์ได้หลายด้านทั้งทางด้านอุตสาหกรรม เช่น อุตสาหกรรมการเกษตรโดยใช้ในการคัดแยกเมล็ดพันธุ์ หรือระบบรักษาความปลอดภัย เช่น การติดตั้งระบบกล้องโทรทัศน์วงจรปิดเพื่อรักษาความปลอดภัย หรือการตรวจจับความเคลื่อนไหวในธนาคาร เหล่านี้เป็นต้นโดยเฉพาะการนำทฤษฎีสีมาใช้วิเคราะห์ภาพ สามารถนำมาใช้ประโยชน์ในการตรวจจับขอบของวัตถุ (Edge Detection) คือการตัดขอบของแต่ละวัตถุในภาพเพื่อให้ได้รูปทรงของวัตถุและการระบุตำแหน่งวัตถุที่เป็นเป้าหมาย (Target) คือการค้นหาตำแหน่งของวัตถุในภาพ

ในการประยุกต์ใช้งานเราจึงนำทฤษฎีในการค้นหาเส้นทางเพื่อใช้ควบคุมหุ่นยนต์ให้เดินไปหาวัตถุเป้าหมายได้ และสามารถนำมาใช้ประโยชน์ในการควบคุมมาตรฐานอุตสาหกรรม เช่น การค้นหาวัตถุแปลกปลอมในโรงงานของเล่น โรงงานยา เป็นต้น

เทคโนโลยีที่มีอยู่ในปัจจุบัน ยังกำลังทำการศึกษากันอยู่ และมีเงื่อนไขข้อจำกัดอยู่มาก ทั้งในเรื่องของประสิทธิภาพและความเร็วในการประมวลผล รวมทั้งการพัฒนาขึ้นใช้เองในประเทศยังมีอยู่น้อย ของต่างประเทศที่มีขายก็มีราคาสูง จึงเล็งเห็นว่าการศึกษาตรงนี้จะเป็นจุดเริ่มต้นที่ดี เพื่อจะเป็นแนวทางการศึกษาต่อไป และสามารถประยุกต์ใช้ได้กับงานเฉพาะทางในอนาคตได้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาทฤษฎีในการประมวลผลภาพโดยใช้ทฤษฎีสีในการวิเคราะห์ภาพ
2. เพื่อศึกษาทฤษฎีในการตรวจจับขอบของวัตถุในภาพ(Edge Detection)
3. เพื่อศึกษาทฤษฎีในการค้นหาเส้นทางจากตำแหน่งอ้างอิงไปยังเป้าหมายได้
4. เพื่อสามารถพัฒนาซอฟท์แวร์ในการวิเคราะห์สี และปรับระดับเฉดสีของภาพให้ได้ตามต้องการ เพื่อจะสามารถระบุตำแหน่งของวัตถุใน ภาพ 2 มิติได้
5. เพื่อสามารถพัฒนาซอฟท์แวร์ในการตรวจจับขอบภาพของวัตถุ (Edge Detection)
6. เพื่อสามารถพัฒนาซอฟท์แวร์ในการค้นหาเส้นทาง (Path finding) โดยใช้อัลกอริธึมในการค้นหา (Search)
7. สามารถพัฒนาซอฟท์แวร์ที่ใช้ในการควบคุมหุ่นยนต์ให้เดินตามเส้นทางไปหาวัตถุที่เป็นเป้าหมายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

เป็นการศึกษาทฤษฎีและพัฒนาซอฟท์แวร์ (Software) ในการประมวลผลภาพ เพื่อระบุตำแหน่งของวัตถุที่มีการกำหนดสีของวัตถุและสิ่งแวดล้อมที่แน่นอน และควบคุมหุ่นยนต์ให้เคลื่อนที่ตามเส้นทางไปหาวัตถุเป้าหมายได้ สามารถแบ่งการทำงานเป็นส่วนๆได้ดังนี้

1. ส่วนที่เป็นการวิเคราะห์สีของภาพเพื่อปรับปรุงระดับเฉดสีก่อนจะตรวจจับขอบ
2. ส่วนของการลดสัญญาณรบกวนในภาพ (Noise) โดยใช้อัลกอริธึมในการจัดกลุ่มข้อมูล (Clustering) แบบเคมีน (K-mean)
3. ส่วนของการตรวจจับขอบของวัตถุ(Edge Detection) ในที่นี้จะเป็นการวนหาตำแหน่งที่เป็นขอบของภาพ และล้อมกรอบขอบของภาพที่เป็นวัตถุวงปิดที่ชัดเจน
4. ส่วนของการค้นหาเส้นทาง (Path finding) โดยใช้อัลกอริธึมในการค้นหา (Search)
5. ส่วนของซอฟท์แวร์ที่ใช้ควบคุมหุ่นยนต์ให้เคลื่อนที่ตามเส้นทางไปยังวัตถุเป้าหมายได้

โปรแกรมที่เลือกมาใช้ในการพัฒนาคือ โปรแกรม ไมโครซอฟท์วิชวลเบสิก 6 (Microsoft Visual Basic 6)

1.4 ผลที่คาดว่าจะได้รับ

1. สามารถเข้าใจทฤษฎีต่างๆและสามารถนำมาประยุกต์ใช้ในการระบุตำแหน่งและค้นหาเส้นทางได้
2. สามารถพัฒนาซอฟท์แวร์ที่ใช้ในการระบุตำแหน่งและค้นหาเส้นทางเพื่อควบคุมหุ่นยนต์ให้เคลื่อนที่ตามเส้นทางนั้นได้
3. เรียนรู้เทคโนโลยีการพัฒนาโปรแกรมโดยใช้โปรแกรม ไมโครซอฟท์วิชวลเบสิก 6 (Microsoft Visual Basic 6)

1.5 ขั้นตอนการทำงาน

1. ศึกษางานวิจัยและทฤษฎีทางด้านการประมวลผลภาพ (Image Processing) ที่เกี่ยวกับทฤษฎีสี การตรวจจับขอบของวัตถุ (Edge Detection) และการค้นหาเส้นทาง (Path finding)
2. ศึกษาโปรแกรม ไมโครซอฟท์วิชวลเบสิก 6 (Microsoft Visual Basic 6)
3. ออกแบบอัลกอริธึมในการวิเคราะห์สี ,การปรับปรุงคุณภาพของภาพด้วยการจัดกลุ่มข้อมูลแบบเคมีน,การตรวจจับขอบของวัตถุ,การค้นหาเส้นทางและควบคุมหุ่นยนต์
4. เขียนโปรแกรม
5. ทดสอบและแก้ไขโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 ฮาร์ดแวร์ (Hardware)

อุปกรณ์ที่ใช้ในโครงงานนี้ประกอบด้วย

- คอมพิวเตอร์ส่วนบุคคล 1 ชุด
- กล้องดิจิตอลที่สามารถติดต่อกับคอมพิวเตอร์ได้
- บอร์ดทดลองรุ่น ET-AVR V1.0 ซึ่งใช้ไอซีเบอร์ AT90S8535 เป็นตัวประมวลผล
- ชุดรับส่งคลื่นวิทยุ 49 MHz. ชุดเดียวกับที่ใช้ในรถกระป๋องตราเพชร
- ชุดเฟืองรถทามิย่า (Tamiya) รุ่น Twin-Motor Gearbox

<u>รายละเอียดของอุปกรณ์ที่ใช้มีดังนี้</u>

2.1.1 คอมพิวเตอร์ส่วนบุคคล (Personal Computer)

สำหรับโครงงานนี้เลือกใช้ CPU Pentium III 1 GHz. RAM 256 MB. บนระบบปฏิบัติการวินโดว์ 2000 (Windows 2000)

2.1.2 กล้องดิจิตอล (Digital Camera)

โครงงานนี้ใช้กล้องดิจิตอลรุ่น Logitech Quick Cam ความละเอียด 320 x 240 พิกเซลที่ 24 บิทสี การเชื่อมต่อแบบ USB

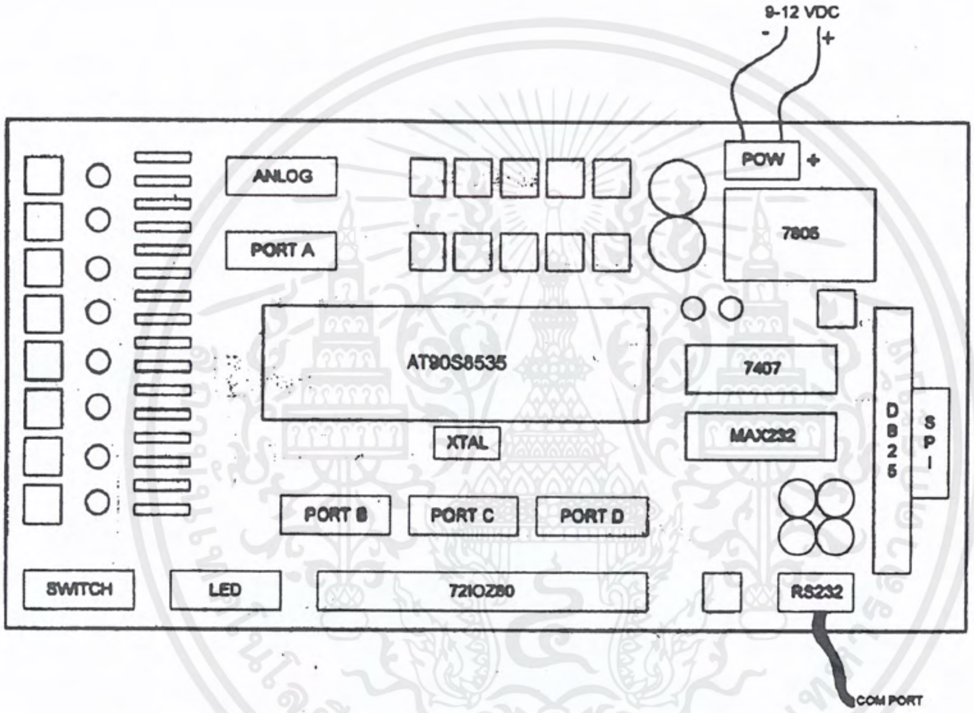
2.1.3 บอร์ดทดลองรุ่น ET-AVR V1.0 (Single Board)

2.1.3.1 คุณสมบัติบอร์ดทดลอง AVR V1.0 มีดังนี้

- พอร์ตแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog To Digital)ขนาด 10 บิท จำนวน 8 ช่องสัญญาณ
- พอร์ตแบบไตรสเตท (Tri state) ขนาด 8 บิทจำนวน 4 พอร์ต
- 34 พินบัส อินพุท/เอาท์พุท 72IOZ80
- พอร์ตอนุกรม (Serial Port RS232)
- ไทม์เมอร์ (Watch Dog Timer)
- แหล่งจ่ายไฟ
- ปุ่มรีเซ็ต
- หน่วยความจำสำหรับโปรแกรม (FLASH) 8 Kbytes
- หน่วยความจำข้อมูล (EEPROM) 512 Bytes
- หน่วยความจำข้อมูล (RAM) 512 Bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 ช่องสัญญาณ PWM
- สวิทช์อินพุทพอร์ต 8 บิท
- LED เอาท์พุท 8 บิท
- อนาล็อกอินพุท 8 ช่องสัญญาณ
- SPI โปรแกรมมิ่งบนบอร์ด
- Xtal 8 MHz.
- แหล่งจ่ายไฟ 5 โวลต์บนบอร์ด



รูปที่ 2.1 แสดงโครงสร้างของบอร์ด ET-AVR V1.0

ในโครงงานใช้ประโยชน์ของบอร์ดในส่วนการใช้งาน RS232 ซึ่งมีไว้เพื่อทดสอบการทำงานของการสื่อสารข้อมูลแบบอนุกรมแบบ RS232 โดยจะเป็นการติดต่อสื่อสารระหว่างบอร์ด ET-AVR V1.0 และคอมพิวเตอร์ส่วนบุคคล (PC) ผ่านทาง RS232 โดยจะต้องต่อสายสื่อสารข้อมูล RS232 ทางพอร์ต RS232 ที่เป็นคอนเนคเตอร์ 4 พินบนบอร์ด และอีกด้านหนึ่งจะต้องต่อเข้ากับคอมพอร์ตของคอมพิวเตอร์ส่วนบุคคล แล้วทำการต่อสายดาวน์โหลด (Download) จากพอร์ตพรินท์เตอร์ (Printer Port) เข้ากับบอร์ด จากนั้นจึงทำการต่อแหล่งจ่ายไฟขนาด 9-12 โวลต์เข้ากับบอร์ด แล้วจึงรันโปรแกรมลงในหน่วยความจำของ AT90S8535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 การทำงานของชิพ AT90S8535

ก) การสื่อสารอนุกรมแบบอซิงโครนัส (UART)

ไมโครคอนโทรลเลอร์ AT90S4434/8535 ถูกจัดให้มีฟังก์ชั่นในส่วนของการสื่อสารแบบอซิงโครนัสหรือ UART (Asynchronous Receiver and Transmitter) โดยได้จัดให้มีคุณสมบัติหลักๆ ดังนี้

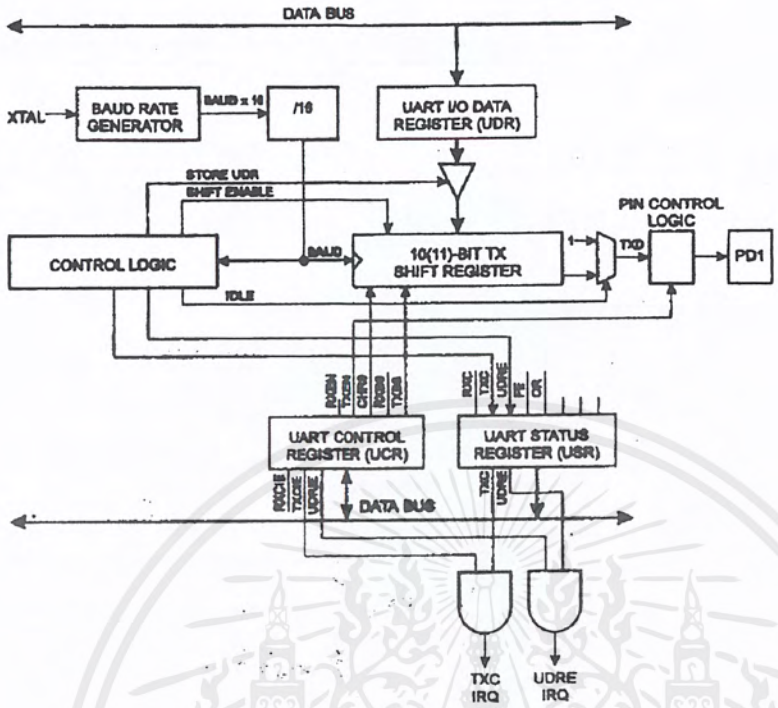
- สามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูล (Baud Rate) ของการสื่อสารได้หลายระดับ
- สามารถสื่อสารได้ในอัตราเร็วในการส่งข้อมูล ที่สูงในขณะที่ความถี่ XTAL ต่ำ
- สื่อสารข้อมูลได้ทั้ง 8 บิทและ 9 บิท
- มีส่วนของการกำจัดสัญญาณรบกวน (Noise filtering) ที่เกิดขึ้นในสายส่ง
- ตรวจจับความผิดพลาดของการสื่อสารข้อมูล ที่เกิดจากการที่ข้อมูลถูกส่งเข้ามาก่อน ยังไม่ได้ถูกอ่านออกจากบัฟเฟอร์และมีข้อมูลใหม่มาทับ (Overrun detection)
- ตรวจจับความผิดพลาดของกรอบข้อมูลผิดพลาด (Framing error detection)
- ตรวจจับความผิดพลาดของบิทเริ่มต้น (False Start Bit detection)
- จัดให้มีการแยกระบบอินเตอร์รัพท์ของการสื่อสารออกเป็น 3 ส่วน คือ อินเตอร์รัพท์ที่เกิดจากการส่งข้อมูล อินเตอร์รัพท์ที่เกิดจากการรับข้อมูล อินเตอร์รัพท์ที่เกิดจากการไม่มีข้อมูลในรีจิสเตอร์
- มีบัฟเฟอร์ในการเก็บข้อมูลที่จะส่งออกและข้อมูลที่รับ

ข) การส่งข้อมูล (Data Transmission)

การเริ่มต้นส่งข้อมูลสามารถกระทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ UDR (UART I/O Data Register) โดยข้อมูลจะถูกถ่ายโอนจากรีจิสเตอร์ UDR เข้าไปยัง Transmit Shift Register เพื่อเลื่อนข้อมูลออกไปยังสายสัญญาณ ซึ่งจะทำให้ข้อมูลที่อยู่ในรีจิสเตอร์ UDR เข้าไปยัง Transmit Shift Register มีอยู่ 2 กรณีคือ

- ถ้าข้อมูลที่ถูกเขียนลงในรีจิสเตอร์ UDR ถูกเขียนภายหลังจากที่บิท Stop ของข้อมูลที่ถูกส่งก่อนหน้านี้ถูกส่งออกไป ซึ่งจะทำให้ข้อมูลที่อยู่ในรีจิสเตอร์ UDR ถูกโหลดลงใน Transmit Shift Register โดยทันทีที่ข้อมูลนั้นถูกเขียนลงในรีจิสเตอร์ UDR
- ถ้าข้อมูลที่ถูกเขียนลงในรีจิสเตอร์ UDR ถูกเขียนที่หน้าก่อนบิท stop ของข้อมูลที่ถูกส่งก่อนหน้านี้จะถูกส่งออกไป ซึ่งจะทำให้ข้อมูลที่อยู่ใน UDR ถูกโหลดลง Transmit Shift Register ในช่วงเวลาที่บิท stop ของข้อมูลก่อนหน้านี้กำลังถูกส่งออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงไดอะแกรมการส่งข้อมูลแบบ UART

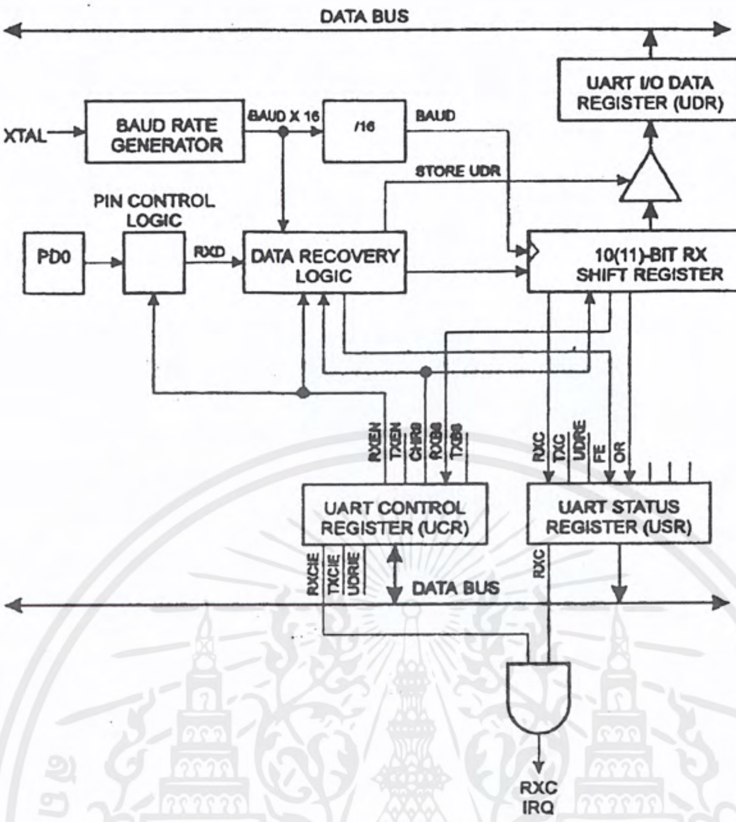
ถ้าใน Transmit Shift Register ไม่มีข้อมูลจะทำให้บิท UDRE (UART Data Register Empty) ในรีจิสเตอร์ USR (UART Status Register) จะถูกเซ็ต ซึ่งแสดงว่าผู้ใช้สามารถเขียนข้อมูลชุดใหม่ลงในรีจิสเตอร์ UDR เพื่อส่งออก โดยในขั้นตอนต่อไปถ้าข้อมูลถูกโหลดลงใน Transmit Shift Register จะทำให้บิทแรกที่จะถูกส่ง ซึ่งเป็นบิท start จะถูกเคลียร์ โดยบิทที่ 9 (ถ้าเป็นการส่งข้อมูลขนาด 10 บิท)หรือบิทที่ 10 (ถ้าเป็นการส่งข้อมูลขนาด 11 บิท) ซึ่งเป็นบิท stop จะถูกเซ็ตให้เป็น 1

ในเวลาของการเริ่มต้นส่งข้อมูลจะทำให้บิท Start ถูกส่งออกที่ขา TXD และตามด้วยข้อมูลที่เป็น LSB บิท โดยเมื่อบิท stop ถูกส่งออกจะทำให้ Transmit Shift Register โหลดข้อมูลชุดใหม่จากรีจิสเตอร์ UDR แต่ถ้าภายหลังจากที่มีการส่งบิท stop ออกไปแล้วและไม่มีข้อมูลใหม่ที่จะทำการส่ง จะทำให้บิท TX Complete Flag และ TXC ในรีจิสเตอร์ USR ถูกเซ็ต

ถ้าบิท TXEN ถูกเซ็ต จะเป็นการกำหนดให้ขา TX ทำหน้าที่ในการสื่อสารข้อมูลแบบอนุกรม แต่ถ้าบิท TXEN ถูกเคลียร์จะทำให้ขา TX ถูกกำหนดเป็นขา I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค) การรับข้อมูล (Data Reception)



รูปที่ 2.3 แสดงไดอะแกรมการรับข้อมูลของ UART

รูปที่ 2.3 แสดงบล๊อคไดอะแกรมของการรับข้อมูล ภาคหน้าของการรับข้อมูลจะเป็นส่วนที่ทำหน้าที่สุ่มข้อมูลที่เข้ามาที่ขา RXD ด้วยอัตรา 16 เท่าของอัตราเร็วในการส่งข้อมูล (Baud Rate) เมื่อสายสัญญาณถูกปล่อยให้ว่าง ส่วนรับข้อมูลจะสุ่มหาขอบขาลงของบิท start โดยสัญญาณที่สุ่มพบการเปลี่ยนแปลงจากลอจิก 1 เป็นลอจิก 0 (ขอบขาลง) ของสัญญาณในสายส่งก็จะถูกกำหนดให้เป็นการสุ่มครั้งแรก โดยสัญญาณการสุ่มที่ 8,9 และ 10 จะถูกนำมาเป็นตัวตัดสินว่าระดับสัญญาณที่ทำการสุ่มเป็นลอจิกใด ซึ่งถ้าเป็นการตรวจสอบหาบิท start และสัญญาณที่ได้จากการสุ่มครั้งที่ 8,9 และ 10 โดย 2 ใน 3 ครั้งได้ลอจิก 1 จะทำระบบรับข้อมูล เริ่มหาบิท start ใหม่ โดยถือว่าบิทที่สุ่มพบเป็นสัญญาณรบกวน แต่ถ้าบิท stop ถูกส่งเข้ามาและสัญญาณที่ได้จากการสุ่ม 2 ใน 3 ครั้ง ไม่ได้สัญญาณลอจิก 1 จะทำให้บิทความผิดพลาดของเฟรม (Framing Error : FE) ในรีจิสเตอร์ USR (UART Status Register) ถูกเซ็ต โดยก่อนที่ผู้ใช้จะทำการอ่านข้อมูลจากรีจิสเตอร์ UDR ผู้ใช้จะต้องทำการตรวจสอบบิท FE ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่รับได้จากสายส่งจะถูกส่งไปยังรีจิสเตอร์ UDR และบิท RXC ในรีจิสเตอร์ USR จะเซ็ตซึ่งรีจิสเตอร์ UDR จะถูกแบ่งออกเป็น 2 ส่วน โดยส่วนหนึ่งจะใช้ในการส่งข้อมูล และอีกส่วนหนึ่งจะใช้ในการรับข้อมูล ถ้าภายหลังจากที่ทำการอ่านรับข้อมูลเรียบร้อยแล้วและยังไม่ได้ทำการอ่านข้อมูลออกจากรีจิสเตอร์ UDR และบิท Over Run (OR) ในรีจิสเตอร์ UCR ถูกเซ็ตซึ่งแสดงว่าข้อมูลไบต์สุดท้ายที่อยู่ในรีจิสเตอร์ UDR เกิดการสูญหาย

ถ้าบิท RXEN ในรีจิสเตอร์ UCR ถูกเซ็ตเป็น 1 จะเป็นการกำหนดให้ขา RXD ทำหน้าที่เป็นขาในการรับข้อมูลแบบอนุกรม แต่ถ้าบิท RXEN ถูกเคลียร์ให้เป็น 0 จะทำให้ขา RXD ทำหน้าที่เป็นขา I/O

ง) การควบคุม UART

The UART I/O Data Register-UDR

Bit	7	6	5	4	3	2	1	0
$0C ($2C)	MSB							LSB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

โครงสร้างจะถูกแยกออกเป็น 2 ส่วนแต่ใช้ตำแหน่งที่อยู่เดียวกัน ถ้าเป็นการเขียนข้อมูลลงในรีจิสเตอร์จะเป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลที่จะส่ง แต่ถ้าเป็นการอ่านข้อมูลจากรีจิสเตอร์ จะเป็นการอ่านข้อมูลที่รับได้จาก UART

The UART Status Register - USR

Bit	7	6	5	4	3	2	1	0
$0B ($2B)	RXC	TXC	UDRE	FE	OR	-	-	-
Read/Write	R/W	R/W	R	R	R	R	R	R
Initial Value	0	0	1	0	0	0	0	0

เป็นรีจิสเตอร์ที่สามารถอ่านได้อย่างเดียว โดยข้อมูลภายในจะแสดงสถานะต่างๆของ UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Bit 7 – RXC:UART Receive Complete

  บิทนี้จะเซ็ตเป็น 1 เมื่อมีการรับข้อมูลจากพอร์ตอนุกรมและข้อมูลเก็บลงในรีจิสเตอร์ UDR เสร็จเรียบร้อย ซึ่งบิท RXC จะเซ็ตโดยไม่คำนึงถึงการตรวจสอบความผิดพลาดของกรอบข้อมูลว่าเกิดความผิดพลาดหรือไม่ โดยเมื่อบิท RXCIE ในรีจิสเตอร์ UCR และบิท RXC เซ็ตเป็น 1 จะทำให้เกิดสัญญาณอินเตอร์รัพท์เกิดขึ้น โดยเมื่อบิท RXC จะถูกเคลียร์ให้เป็น 0 เมื่อค่าในรีจิสเตอร์ UDR ถูกอ่าน

- Bit 6 – TXC:UART Transmit Complete

  บิทนี้ถูกเซ็ตเป็น 1 เมื่อข้อมูลถูกส่งออก(รวมบิท stop)ทางพอร์ตทั้งหมดและไม่มีข้อมูลชุดใหม่เขียนลงในรีจิสเตอร์ UDR ซึ่งสถานะของบิทนี้จะมีประโยชน์มากในการส่งข้อมูลอนุกรมแบบ Half Duplex

  เมื่อบิท TXCIE ในรีจิสเตอร์ UCR และบิท TXC เซ็ตเป็น 1 จะทำให้เกิดอินเตอร์รัพท์ ซึ่งบิท TXC สามารถเคลียร์ให้เป็น 0 ได้โดยการเขียนลอจิก 1 ลงในบิท TXC หรือเคลียร์โดยอัตโนมัติเมื่อ CPU ทำโปรแกรมบริการอินเตอร์รัพท์

- Bit 5 – UDRE:UART Data Register Empty

  บิทนี้จะถูกเซ็ตเป็น 1 เมื่อข้อมูลจาก Data Register ถูกเขียนลงใน Shift Register ซึ่งหมายถึงไม่มีข้อมูลใน Data Register เพื่อแสดงให้ผู้ใช้เขียนข้อมูลชุดใหม่ลงใน Data Register เพื่อส่งข้อมูลชุดต่อไป

  เมื่อบิท UDRIE และบิท UDRE เซ็ตเป็น 1 จะทำให้เกิดสัญญาณอินเตอร์รัพท์ โดยบิท UDRE จะเคลียร์เป็น 0 เมื่อเขียนข้อมูลลงในรีจิสเตอร์ UDR

- Bit 4 – FE Framing Error

  บิทนี้จะถูกเซ็ตเป็น 1 เมื่อมีการตรวจพบว่ากรอบข้อมูลมีการผิดพลาด

- Bit 3 – OR: Over run

  บิทนี้จะถูกเซ็ตเป็นเมื่อตรวจสอบได้ว่าเกิด Over run คือการที่ข้อมูลชุดใหม่เข้ามาทับข้อมูลเดิม

- Bit 2..0 – Res: Reserved bits

  บิท 0 –บิท 2 ถูกสงวนไว้ โดยปกติจะเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The UART Control Register – UCR

Bit	7	6	5	4	3	2	1	0
$0A ($2A)	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- Bit 7 – RXCIE: RX complete Interrupt Enable

เป็นบิทที่ใช้ Enable อินเตอร์รัพท์ ของการรับข้อมูลคือ เมื่อการรับข้อมูลไม่มีควาผิดพลาด จะทำให้บิท RXC เซ็ตเป็น 1 ซึ่งจะทำให้เกิดอินเตอร์รัพท์ถ้าบิท RXCIE และบิท I เซ็ตเป็น 1

- Bit 6 – TXCIE: TX Complete Interrupt Enable

เป็นบิทที่ใช้ Enable อินเตอร์รัพท์ของการส่งข้อมูล คือ เมื่อมีการส่งข้อมูลครบทุกบิทจะทำให้บิท TXC เซ็ตเป็น 1 ซึ่งจะทำให้เกิดอินเตอร์รัพท์ ถ้าบิท TXCIE และบิท I เซ็ตเป็น 1

- Bit 5 – UDRIE: UART Data Register Empty Interrupt Enable

ถ้าไม่มีข้อมูลในรีจิสเตอร์ UDR จะทำให้บิท UDRE เซ็ตเป็น 1 ซึ่งจะทำให้เกิดอินเตอร์รัพท์ ถ้าบิท UDRE และบิท I เซ็ตเป็น 1

- Bit 4 – RXEN: Receiver Enable

ใช้กำหนดให้ UART เริ่มรับข้อมูล

- Bit 3 –TXEN: Transmitter Enable

ใช้กำหนดให้ UART เริ่มส่งข้อมูล

- Bit 2 – CHR9: 9 Bit Characters

เป็นบิทที่กำหนดให้มีการรับส่งข้อมูลขนาด 9 บิท ถ้าบิทนี้เซ็ตเป็น 1 จะเป็นการกำหนดให้มีการรับส่งข้อมูลขนาด 9 บิท แต่ถ้าบิท CH9 ถูกเคลียร์เป็น 0 จะเป็นการรับส่งข้อมูลขนาด 8 บิท

- Bit 1 – RXB8: Receive Data Bit 8

บิท RXB8 จะทำหน้าที่ในการเก็บค่าบิทที่ 9 ของการรับข้อมูลขนาด 9 บิท

- Bit 0 – TXB8: Transmit Data Bit 8

บิท TXB8 จะทำหน้าที่ในการเก็บค่าบิทที่ 9 ของการส่งข้อมูลขนาด 9 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จ) การกำหนดอัตราเร็วในการรับส่งข้อมูล (Baud Rate)

$$BAUD = \frac{f_{CK}}{16(UBRR+1)}$$

- BAUD = Baud Rate
- f_{CK} = Crystal Clock Frequency
- UBRR = Content of the UART Baud Rate register , UBRR(0-255)

รีจิสเตอร์ UBRR (UART Baud Rate Register)

เป็นรีจิสเตอร์ขนาด 8 บิทที่ใช้ในการกำหนดอัตราเร็วในการรับส่งข้อมูล

Bit	7	6	5	4	3	2	1	0
\$09 (\$29)	MSB							LSB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงการกำหนดค่าอัตราเร็วในการส่ง (Baud Rate)

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBRR=	25	0.2	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	63	0.0
4800	UBRR=	12	0.2	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	31	0.0
9600	UBRR=	6	7.5	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	15	0.0
14400	UBRR=	3	7.8	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	0.0
28800	UBRR=	1	7.8	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	6.3
38400	UBRR=	1	22.9	UBRR=	2	0.0	UBRR=	2	7.8	UBRR=	3	0.0
57600	UBRR=	0	7.8	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	12.5
76800	UBRR=	0	22.9	UBRR=	1	33.3	UBRR=	1	22.9	UBRR=	1	0.0
115200	UBRR=	0	84.3	UBRR=	0	0.0	UBRR=	0	7.8	UBRR=	0	25.0

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBRR=	84	0.4	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0
4800	UBRR=	42	0.8	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0
9600	UBRR=	20	1.6	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0
14400	UBRR=	13	1.6	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0
19200	UBRR=	10	3.1	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0
28800	UBRR=	6	1.6	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0
38400	UBRR=	4	6.3	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	6.7
57600	UBRR=	3	12.5	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0
76800	UBRR=	2	12.5	UBRR=	2	0.0	UBRR=	2	7.8	UBRR=	3	6.7
115200	UBRR=	1	12.5	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	287	-
4800	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	143	0.0
9600	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	71	0.0
14400	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	47	0.0
19200	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0	UBRR=	35	0.0
28800	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	23	0.0
38400	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0	UBRR=	17	0.0
57600	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	11	0.0
76800	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	6.7	UBRR=	8	0.0
115200	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0	UBRR=	5	0.0

Maximum Baud rate to each frequency

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉ) พอร์ตอินพุตและพอร์ตเอาท์พุต

พอร์ต A

เป็นพอร์ตสองทิศทางขนาด 8 บิท โดยมีหน่วยควบคุมการทำงานต่างๆของพอร์ต คือ รีจิสเตอร์ PORTA (DATA REGISTER) อยู่ที่ตำแหน่ง $1B ($3B), รีจิสเตอร์ DDRA (DATA DIRECTION REGISTER) อยู่ที่ตำแหน่ง $1A($3A) และ PINA (PORTA INPUT PINS) อยู่ที่ตำแหน่ง $19($39) โดย PINA จะสามารถอ่านได้อย่างเดียว ไม่สามารถเขียนข้อมูลลงได้ ในขณะที่ PORTA และ DDRA สามารถอ่านและเขียนได้ โดยแต่ละขาสัญญาณของพอร์ต A สามารถกำหนดให้ความต้านทาน PULL-UP ทำงานในสภาวะแอคทีฟหรือไม่แอคทีฟก็ได้ โดยแต่ละขาสัญญาณของพอร์ต A จะสามารถรับกระแส (SINK CURRENT) ได้ 20 มิลลิแอมป์ ซึ่งถ้าขาของพอร์ตใช้งานเป็นอินพุทและภายนอกมีความต้านทาน PULL LOW ในขณะที่ความต้านทาน PULL UP ภายในแอคทีฟ จะทำให้ MCU จ่ายกระแสออกมาภายนอก

พอร์ต A จะถูกนำไปใช้งานอีกอย่างหนึ่งคือ เป็นขาสัญญาณของอินพุทของวงจรแปลงอนาล็อกเป็นดิจิตอล โดยถ้าบางขาของพอร์ต A ถูกใช้งานเป็นขาเอาท์พุตของพอร์ต ในขณะที่บางขาของพอร์ต A ถูกใช้เป็นขาอินพุตของอนาล็อกเป็นดิจิตอล กำลัง Conversion สัญญาณ ซึ่งอาจจะทำให้การทำการแปลงสัญญาณถูกขัดจังหวะ จะมีผลให้ผลการแปลงสัญญาณผิดพลาด

การทำงานในโหมดประหยัดพลังงานจะทำให้ SCHMITT TRIGGER ของส่วนอินพุตถูกตัดออก

The Port A Data Register – PORT A

Bit	7	6	5	4	3	2	1	0
$1B($3B)	POARTA7	POARTA6	POARTA5	POARTA4	POARTA3	POARTA2	POARTA1	POARTA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Port A Input Pins Address - PINA

Bit	7	6	5	4	3	2	1	0
\$19(\$39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0
Read/Write	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

PINA ไม่ใช่รีจิสเตอร์ ซึ่งตำแหน่งที่กำหนดจะเป็นตำแหน่งของแต่ละขาของพอร์ต โดยเมื่ออ่านพอร์ต A จะเป็นการอ่านค่าที่ LATCH ส่วนการอ่านค่า PINA จะเป็นการอ่านค่าจริงๆของขาพอร์ต

การใช้งานพอร์ต A เป็นขาอินพุตเอาท์พุต

บิท DDAn ในรีจิสเตอร์ DDRA จะเป็นบิทที่ใช้ในการกำหนดทิศทางของแต่ละพอร์ต ถ้าบิท DDAn ถูกเซ็ตเป็น 1 จะทำให้ขาของพอร์ตนั้นเป็นเอาท์พุต แต่เมื่อ DDAn ถูกเคลียร์เป็น 0 จะทำให้ขานั้นถูกกำหนดเป็นอินพุต ซึ่งสภาวะของขาต่าง ๆ แสดงดังตารางที่ 2.2

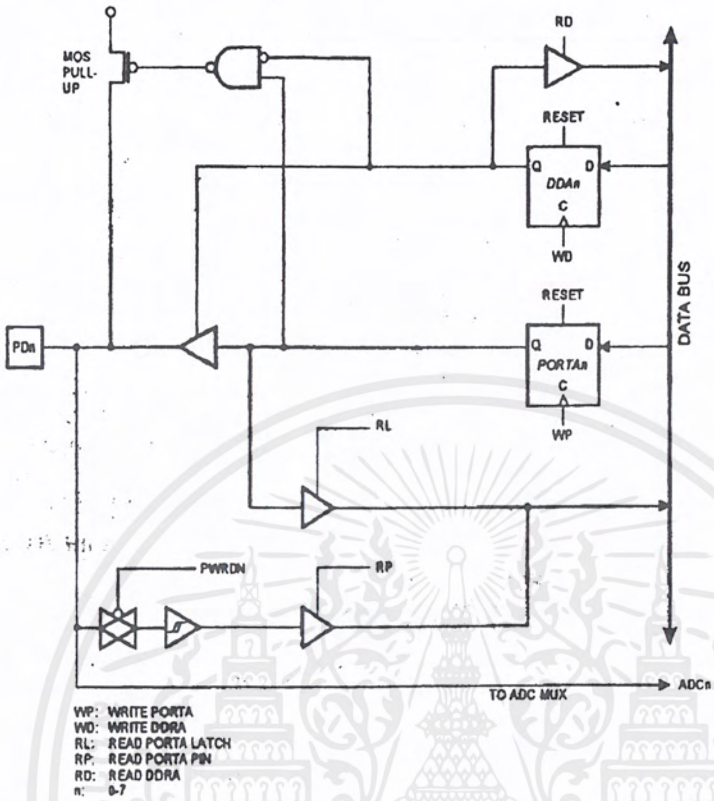
DDAn	PORTA	I/O	Pull Up	Comment
0	0	Input	No	Tri-State (Hi-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

ตารางที่ 2.2 DDAn Effects on PORTA Pins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>โครงสร้างของพอร์ต A</u>

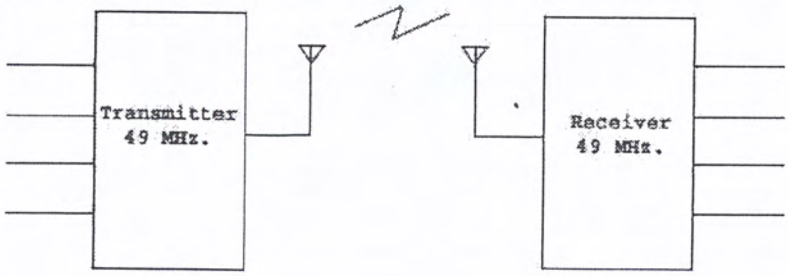
ทุกขาทั้งหมดของพอร์ต A จะมีลักษณะโครงสร้างดังรูปที่ 2.4



รูปที่ 2.4 แสดงโครงสร้างของพอร์ต A (Pins PA0 – PA7)

2.1.4 ชุดรับส่งคลื่นวิทยุ 49 MHz.

ในโครงงานนี้ใช้ชุดรับส่งคลื่นวิทยุในการสื่อสารระหว่างโปรแกรมคอมพิวเตอร์กับตัวรถซึ่งเป็นชุดรับส่งแบบ 4 ช่องสัญญาณอิสระต่อกัน จึงทำให้สามารถควบคุมการทำงานมอเตอร์ได้ 2 ตัวในเวลาเดียวกัน และเนื่องจากความพิเศษของวงจรนี้ที่รวมวงจรไดร์ฟมอเตอร์และวงจรรับสัญญาณคลื่นวิทยุไว้ด้วยกัน จึงทำให้ไม่ต้องสร้างวงจรไดร์ฟมอเตอร์ขึ้นมาใหม่



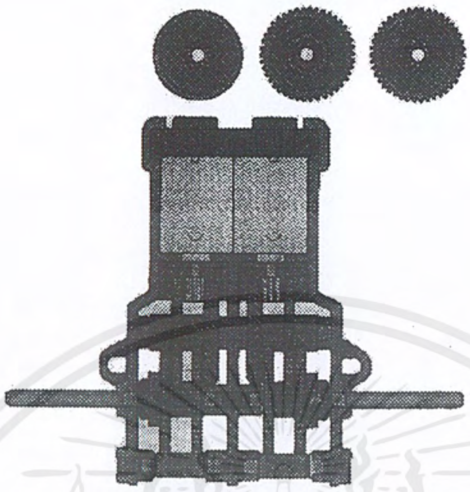
รูปที่ 2.5 รูปวงจรรับส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ตัวรถ (Car Body)

โครงสร้างของตัวรถประกอบด้วย

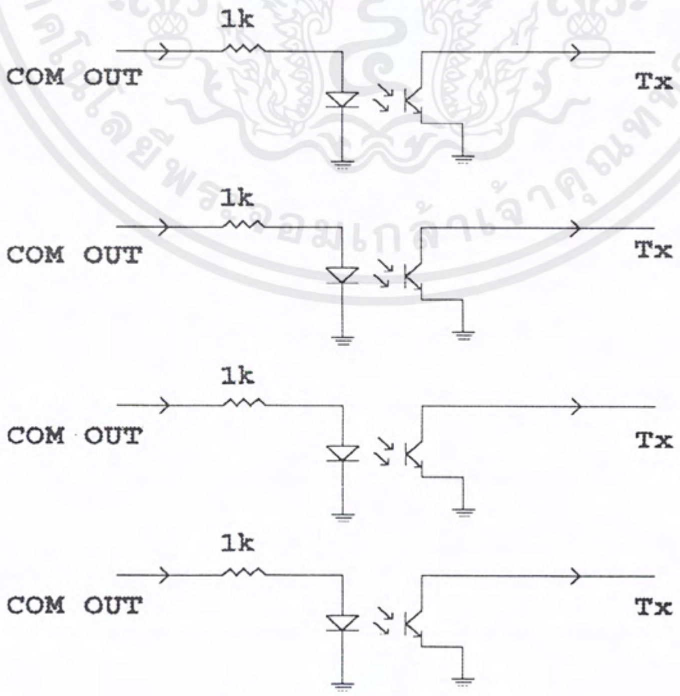
- ชุดเฟืองรถทามิย่า อัตราทด 203:1
- ชุดล้อ 2 ล้อ



รูปที่ 2.6 ชุดเฟืองรถทามิย่า

2.1.6 วงจรสวิตช์

เนื่องจากสัญญาณที่ได้จากบอร์ด AVR จะมีความต่างศักย์ของเอาท์พุต 0-5 โวลต์ แต่วงจรรับส่งคลื่นวิทยุใช้ความศักย์อินพุตที่ 0-3 โวลต์ จึงต้องสร้างวงจรสวิตช์ซึ่งทำงานที่ 5 โวลต์เพื่อควบคุมสวิตช์ 3 โวลต์เพื่อให้ทั้งสองวงจรทำงานร่วมกันได้



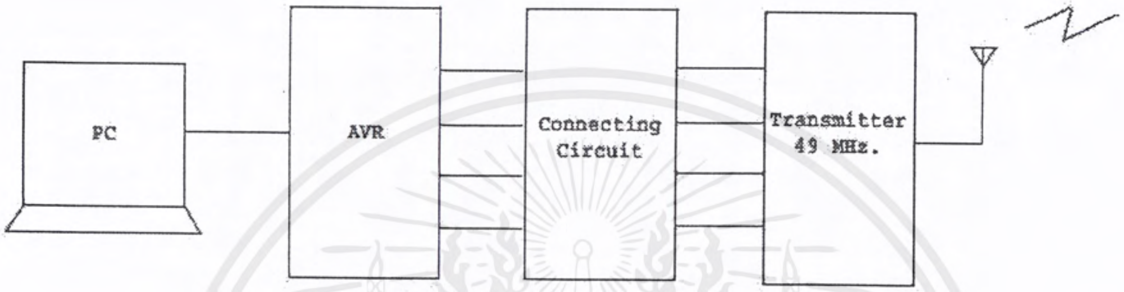
รูปที่ 2.7 วงจรสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

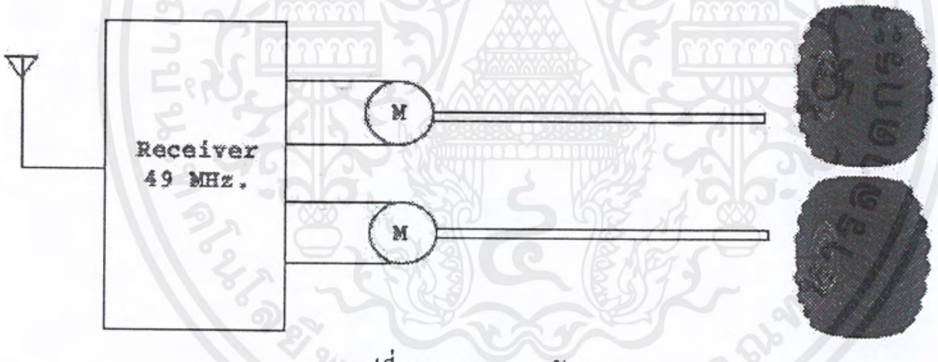
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.1.7 วงจรรวม

เมื่อคอมพิวเตอร์ประมวลผลภาพเรียบร้อยแล้ว จะส่งสัญญาณควบคุมทางพอร์ตอนุกรม RS-232 (COM1, COM2) ผ่านไปยังบอร์ด AVR ซึ่งจะทำการแปลงให้เป็นสัญญาณขนาน 4 บิทผ่านไปยังวงจรสวิตช์เพื่อแปลงสัญญาณลอจิก 0-5 โวลต์เป็นสวิตช์ไปควบคุมวงจรส่งให้ส่งสัญญาณคลื่นวิทยุไปยังตัวรับ ทางด้านตัวรับจะนำสัญญาณที่ได้ไปควบคุมล้อต่อไป



รูปที่ 2.8 วงจรภาคส่ง

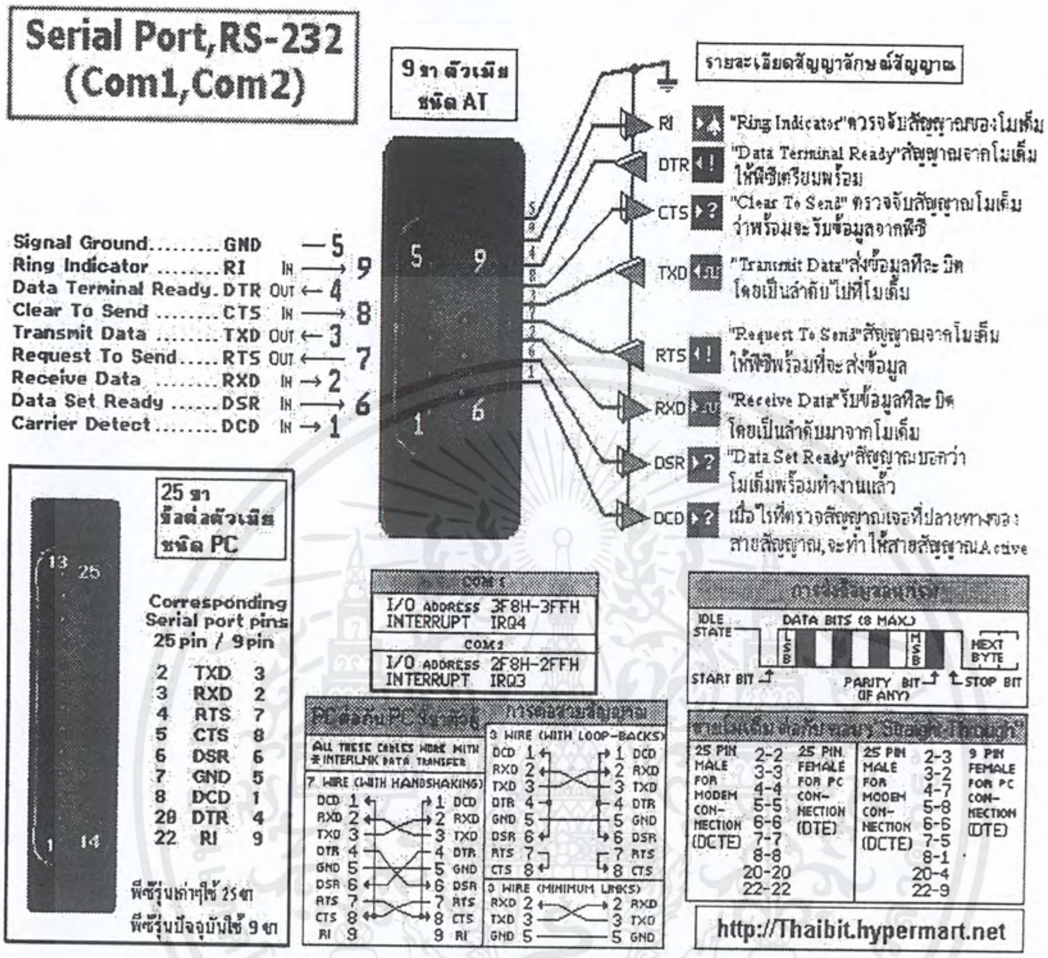


รูปที่ 2.9 วงจรภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46437

2.1.8 การเชื่อมต่อด้วยพอร์ตอนุกรม RS-232



รูปที่ 2.10 รายละเอียดของคอมพอร์ต (COM1, COM2)

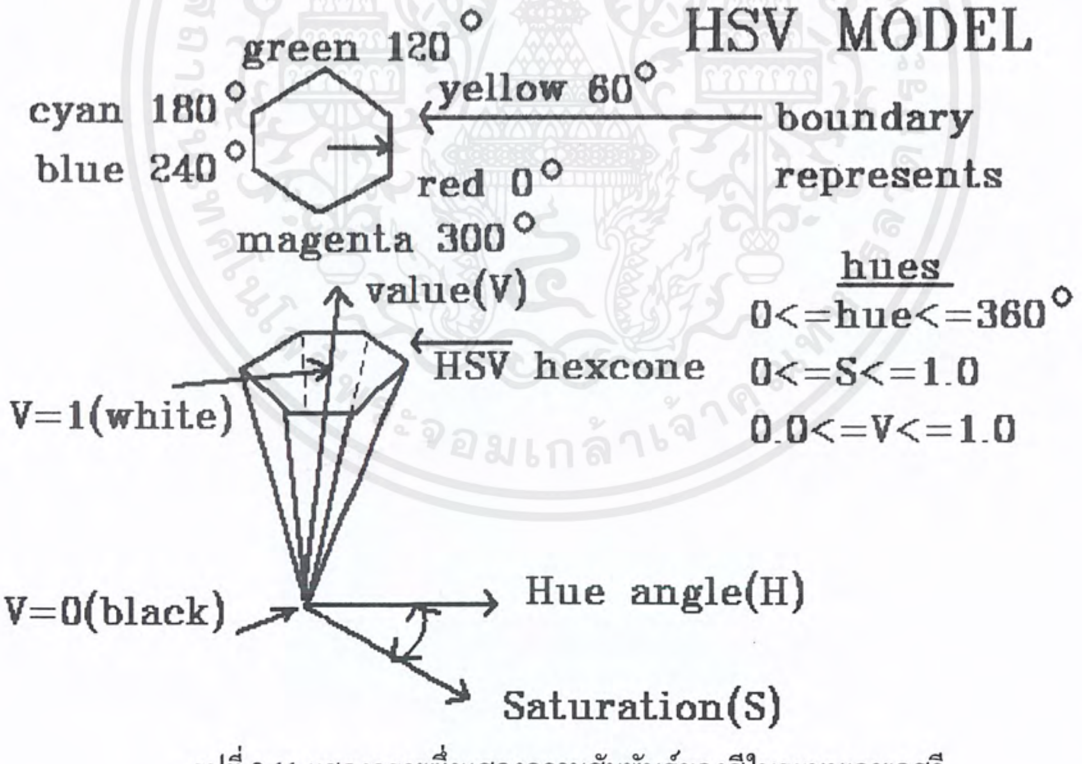
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ซอฟท์แวร์ (Software)

2.2.1 โมเดลของสีชนิดเอชเอสวี (HSV Color Model)

HSV Model (Hue, Saturation, Value) เป็นโมเดลที่แสดงคุณลักษณะของสีได้ดีกว่าโมเดลอาร์จีบี (RGB Model) เราสามารถเลือกสีโดยการกำหนดค่า HUE แล้วทำการปรับความเข้มของสีดำและขาว พารามิเตอร์ที่สนใจมีอยู่ 3 พารามิเตอร์ด้วยกัน ได้แก่ Hue, Saturation และ Value การปรับค่า Saturation นั้นจะเป็นการเพิ่มหรือลดความเข้มของสีขาว และการปรับค่า Value จะเป็นการเพิ่มหรือลดความเข้มของสีดำ

โมเดลของเอชเอสวีนี้ได้มาจากรูปแสดงโมเดลของอาร์จีบี โดยถ้ามองจากแกนของเกรย์ (Gray Diagonal) จะเห็นเป็นรูปหกเหลี่ยมซึ่งก็คือโคนหกเหลี่ยมซึ่งแสดงเอชเอสวีโมเดลนั่นเอง ค่า HUE ก็คือค่ามุมที่ทำในแนวแกนตั้ง(Vertical Axis) โดยสีแดง (Red)ทำมุม 0 องศา สีเหลือง(Yellow) ทำมุม 60 องศา สีเขียว(Green)ทำมุม 120 องศา สีน้ำทะเล(Cyan)ทำมุม 180 องศา สีฟ้า(Blue)ทำมุม 240 องศาและ สีบานเย็น(Magenta)ทำมุม 300 องศา



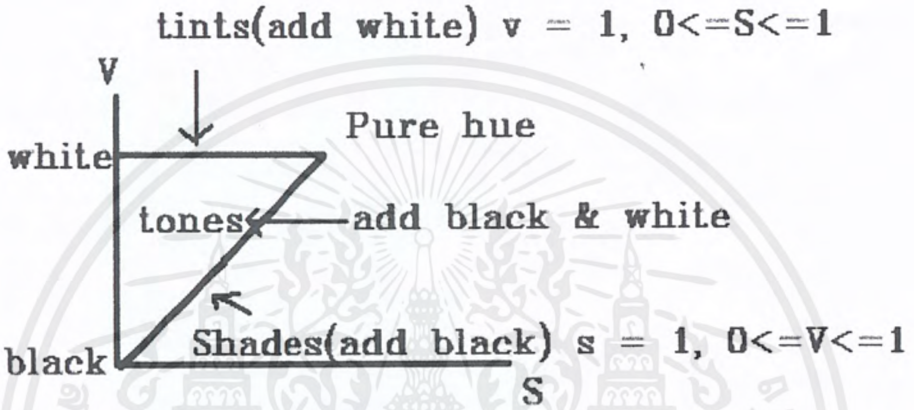
รูปที่ 2.11 แสดงกรวยซึ่งแสดงความสัมพันธ์ของสีในระบบเอชเอสวี

ค่าแซททูเรชั่น (Saturation) จะมีค่าตั้งแต่ 0.0 ถึง 1.0 และเป็นอัตราส่วนกับความบริสุทธิ์ของ HUE นั้นๆ โดยจะมีค่ามากที่สุดที่ S = 1 และที่ S = 0 จะเป็นเกรย์สเกล (Gray Scale) ดังนั้นจะเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

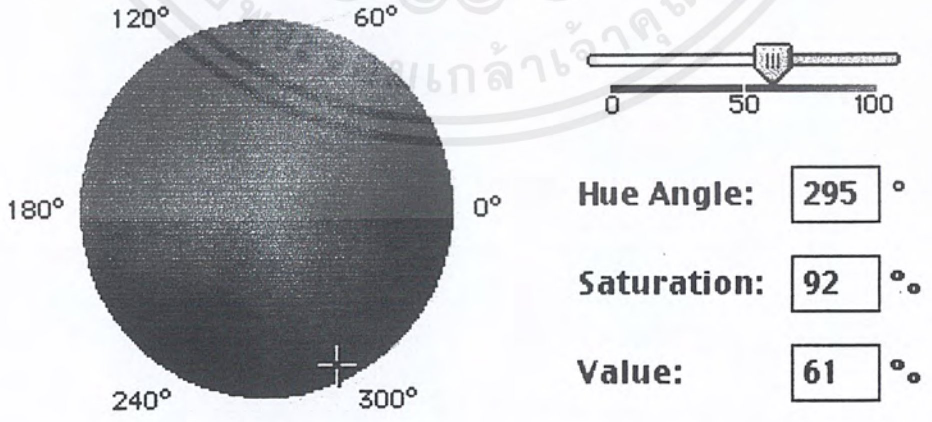
ว่าแกนของ RGB จะสัมพันธ์กับค่า V ใน HSV และสีตรงข้าม (Complementary) ได้แก่ Red +Cyan, Blue+ Yellow, Green+ Magenta คือแกนที่มีทิศตรงกันข้ามนั่นเอง การเลือกสีทำได้โดยการเลือกค่า Hue บริสุทธิ์ (ที่ S = V = 1) โดยลดค่า S เพื่อเพิ่มสีขาว และลดค่า V เพื่อเพิ่มสีดำ

ตัวอย่างเช่นหลักการผสมสีของศิลปินก็คือการเริ่มต้นด้วยแม่สี (Pure Hue) แล้วจึงค่อยๆเติมสีดำหรือขาว โดยถ้ายิ่งเพิ่มสีดำ ก็จะได้สีที่มีเฉดเข้ม หรือถ้าเพิ่มสีขาวก็จะได้สีเฉดอ่อน ถ้าเพิ่มทั้ง 2 อย่างก็จะได้สีที่มีโทนสีแตกต่างกันไป



รูปที่ 2.12 แสดงภาพภาคตัดขวางของกรวยเอชเอสวี

ตามนุษย์โดยปกติสามารถแยกแยะความแตกต่างได้ 128 Hue 130 Tints (ระดับของ Saturation) และ 16 ถึง 23 เฉด ดังนั้นมนุษย์จึงสามารถแยกความแตกต่างของสีได้เท่ากับ 128 x 130 x 23 = 380,000 สี



รูปที่ 2.13 แสดงวงล้อของสี (Color Wheel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การแบ่งกลุ่ม (Clustering) แบบเคมีน (K-means)

เป็นอัลกอริธึมที่ใช้ในการแบ่งกลุ่มของข้อมูล ใช้หลักการหาค่ากลาง (mean) ของกลุ่ม (cluster) นั้นเพื่อเป็นตัวแทนของกลุ่มข้อมูล มีขั้นตอนดังนี้

ขั้นที่1 ทำการกำหนดก่อนว่าจะแบ่งข้อมูลเป็นกี่กลุ่ม และ ทำการสุ่ม ค่าจุดศูนย์กลางเริ่มต้นให้เท่าจำนวนกลุ่มที่ต้องการ โดยแต่ละจุดต้องไม่ซ้ำกัน

ขั้นที่ 2 นำข้อมูลทุกชุดมาคำนวณหาระยะห่างจากจุดศูนย์กลางทุกจุด โดยคิดจากผลต่างยกกำลังสอง ตามสูตร

$$E_k^2 = \sum_{i=1}^{n_k}(x_i^{(k)} - m^{(k)})^T(x_i^{(k)} - m^{(k)})$$

E: error (distance)

x: ข้อมูล

m: จุดศูนย์กลาง

k: จำนวนกลุ่ม

และเปรียบเทียบหาระยะห่างน้อยที่สุด เพื่อหาว่าข้อมูลควรอยู่ในกลุ่มไหน ทำจนครบทุกชุดของข้อมูล เช่น ถ้าต้องการแบ่งข้อมูลเป็น 4 กลุ่ม ให้กำหนดจุดศูนย์กลาง 4 จุด แล้วนำข้อมูลแต่ละชุดมาหาระยะห่างจากจุดศูนย์กลาง ข้อมูลชุดที่ 1 (2, 1, 0, 1)

เทียบกับจุดศูนย์กลาง (0, 1, 1, 1) $e^2 = (2-0)^2 + (1-1)^2 + (0-1)^2 + (1-1)^2 = 5$

เทียบกับจุดศูนย์กลาง (0, 1, 1, 0) $e^2 = (2-0)^2 + (1-1)^2 + (0-1)^2 + (1-0)^2 = 6$

เทียบกับจุดศูนย์กลาง (0, 1, 0, 1) $e^2 = (2-0)^2 + (1-1)^2 + (0-0)^2 + (1-1)^2 = 4$

เทียบกับจุดศูนย์กลาง (0, 1, 0, 0) $e^2 = (2-0)^2 + (1-1)^2 + (0-0)^2 + (1-0)^2 = 5$

ระยะห่างน้อยที่สุด คือ 4 ข้อมูลควรอยู่ในกลุ่ม 3 ที่มีจุดศูนย์กลาง (0, 1, 0, 1) ทำจนครบทุกชุดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่3 คำนวณหาค่าจุดศูนย์กลางของกลุ่มใหม่ เนื่องจากมีการเปลี่ยนแปลงของข้อมูลในกลุ่ม โดยการคำนวณหาค่าจุดศูนย์กลางเหมือนการคำนวณหาค่ากลาง (mean value)

$$m^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i^{(k)}$$

m: ค่า center

n: จำนวนข้อมูล

x: ข้อมูล

เช่น ถ้าแบ่งข้อมูลเป็น 4 กลุ่ม โดยกลุ่มที่ 1 มี (0,1,1,1) (0,0,1,1) (0,1,01) (1,1,1,1) ค่าจุดศูนย์กลางใหม่จะเป็น

Class $0+0+0+1/3 = 0.33$

Age $1+0+1+1/3 = 1$

Sex $1+1+0+1/3 = 1$

Survive $1+1+1+1/3 = 1.33$

ทำจนได้จุดศูนย์กลางใหม่จนครบทุกกลุ่ม

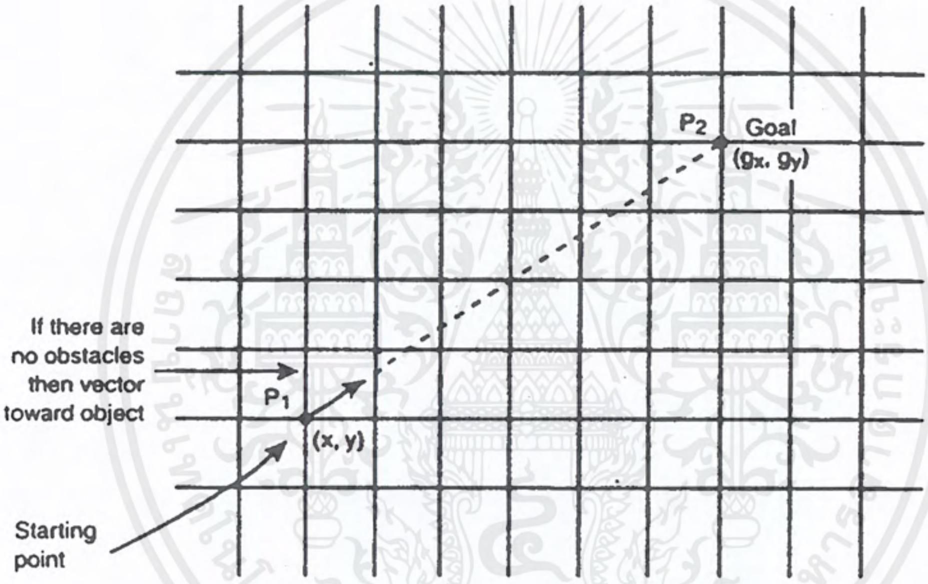
ขั้นที่ 4 ทำตาม ขั้นที่และขั้นที่3 อีกครั้ง โดยนำข้อมูลทั้งหมดมาเทียบหาระยะห่างจากจุดศูนย์กลางใหม่เพื่อหาค่าที่น้อยที่สุด และคำนวณหาจุดศูนย์กลางแต่ละกลุ่มใหม่ ทำจนค่าจุดศูนย์กลางคงที่หรือจนกว่าจะพอใจ

ขั้นที่5 ปรับจำนวนกลุ่มให้เหมาะสมโดยอาจรวมกลุ่ม หรือแยกกลุ่มที่มีอยู่ หรือตัดกลุ่มเล็กๆทิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การค้นหาเส้นทาง (Path Finding)

Path finding คือการคำนวณและวิเคราะห์หาเส้นทางที่จะเดินจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยจะพยายามหาเส้นทางที่สามารถไปถึงเป้าหมายได้โดยใช้เวลาหรือระยะที่สั้นที่สุด ในรูปที่ 12.17 ถ้าเป็นกรณีที่ไม่มีสิ่งกีดขวางบนเส้นทางเดินเลย ก็จะสร้างเวคเตอร์ทางเดินจากจุดเริ่มต้นไปยังเป้าหมายตรงๆเลย แต่หากเป็นกรณีที่มีวัตถุมากีดขวางทางเดินก็จะต้องมีอัลกอริธึมในการค้นหาเส้นทางเดินเพื่อที่จะหลบหลีกสิ่งกีดขวางและเดินทางไปถึงเป้าหมายให้ได้ ซึ่งจะทำให้กระบวนการคิดคำนวณมีความซับซ้อนมากยิ่งขึ้น



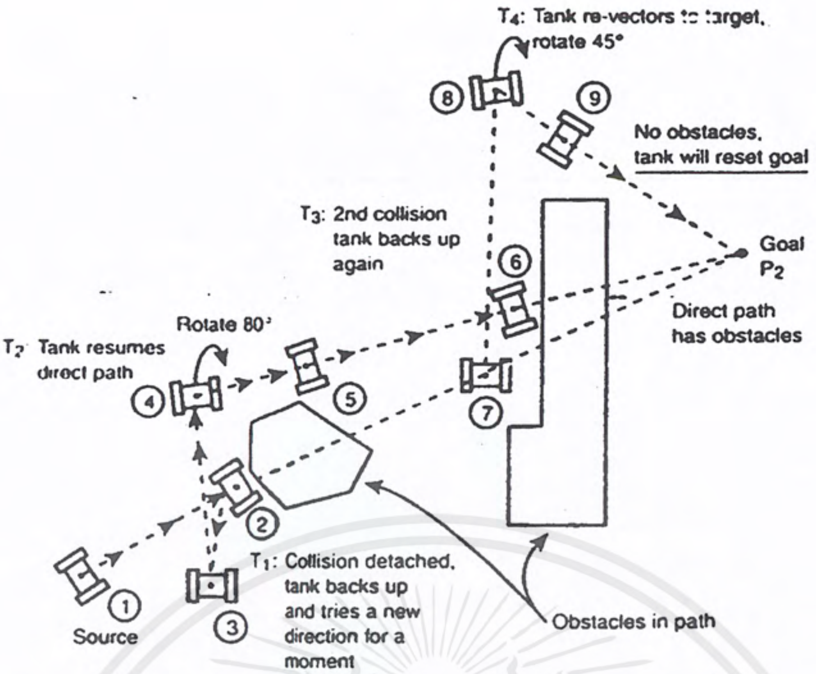
รุปที่ 2.14 การค้นหาเส้นทางจากจุดหนึ่งไปอีกจุดหนึ่ง

วิธีในการคำนวณหาเส้นทางการเดินมีอยู่หลายวิธีด้วยกัน ได้แก่

- Trial and Error

เป็นวิธีการสุ่มเดินไปเรื่อยๆ โดยวิธีนี้จะเหมาะกับสภาพแวดล้อมที่มีสิ่งกีดขวางขนาดไม่ใหญ่นัก และส่วนใหญ่เป็นพื้นที่ที่มีความโค้งหรือนูนออก กระบวนการคิดจะตัดสินใจให้เดินไปเรื่อยๆ เมื่อชนเข้ากับสิ่งกีดขวางจะทำการถอยหลังกลับมาแล้วทำการหมุนซ้ายหรือขวาไปเท่ากับมุมที่กำหนด เช่น 45-90 องศา หลังจากนั้นจึงเดินหน้าต่อไปอีกในระยะที่ตั้งไว้แล้วทำการค้นหาเป้าหมายอีกครั้งหนึ่ง (Retarget) และทำการทดลองเดินไปข้างหน้าอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

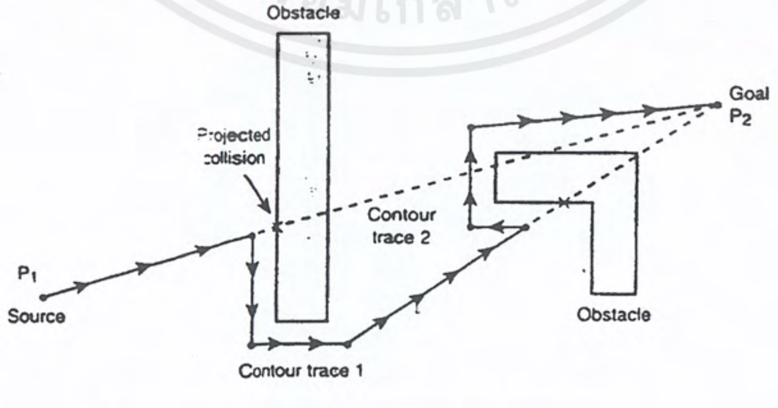


รูปที่ 2.15 แสดงการทำงานของอัลกอริธึมในการชนและถอย

อย่างไรก็ตามวิธีนี้ไม่ได้ใช้วิธีคิดคำนวณที่ฉลาดนัก แต่สามารถใช้ค้นหาเส้นทางได้จริง เพราะมีการสุ่มเลือกทางเพื่อหลบสิ่งกีดขวาง จะทดลองเดินไปในแต่ละเส้นทางจนหาทางไปยังเป้าหมายได้ในที่สุด

- Contour Tracing

วิธีนี้ใช้หลักการหาเส้นทางเดินอ้อมเพื่อหลบหลีกสิ่งกีดขวางจากขอบเขตนอก (Contour) ของวัตถุที่ขวางบนเส้นทางที่จะเดิน โดยขอบนอกของวัตถุจะห่างจากขอบของวัตถุจริงเป็นระยะหนึ่งซึ่งจะทำให้ไม่ไปชนกับวัตถุ และจะทำการตรวจสอบการซ้อนทับของสิ่งกีดขวางบนเส้นทางเดินเป็นระยะๆ ถ้าไม่มีก็จะทำการสร้างเวคเตอร์ทางเดินตรงไปยังเป้าหมายเลย



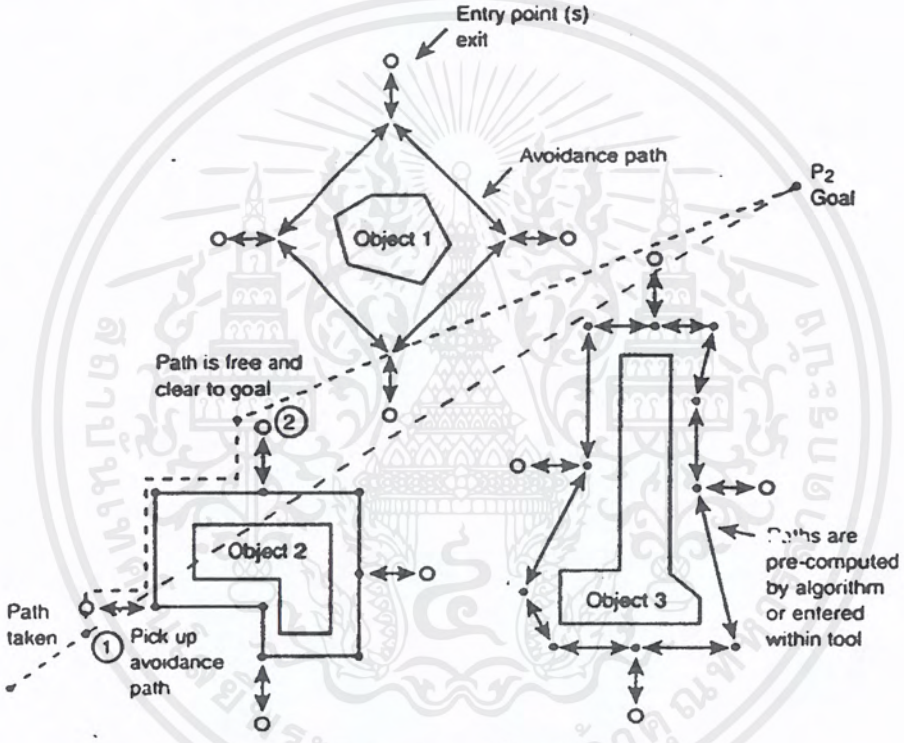
รูปที่ 2.16 การค้นหาเส้นทางของวิธี Contour tracing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธี Contour tracing นี้ยังไม่เป็นวิธีที่ดีที่สุด เพราะใช้วิธีเดินอ้อมสิ่งกีดขวางแทนที่จะใช้เส้นทางที่เดินไปยังเป้าหมายโดยตรงซึ่งจะใช้ระยะที่สั้นที่สุด ดังนั้นจึงควรลองใช้วิธี Trial and Error ก่อนแล้วถ้าไม่สามารถหาเส้นทางที่จะไปยังเป้าหมายได้แล้วจึงค่อยมาใช้วิธีนี้

- Collision Avoidance Tracks

แนวคิดของวิธีคือจะทำการสร้างทางเดินรอบๆวัตถุซึ่งประกอบด้วยชุดของจุดและเวคเตอร์ของเส้นทางโดยจะคำนวณหาจากระยะทางที่สั้นที่สุด เมื่อตัวหุ่นเดินไปจะถามไปยังตัวกระบวนการหาเส้นทาง (Path finder) ว่าจะต้องเดินไปทางใดเพื่อให้ได้เส้นทางที่สามารถหลบหลีกสิ่งกีดขวางได้ในระยะทางที่สั้นที่สุด ซึ่งความจริงอาจมีมากกว่าหนึ่งเส้นทางก็ได้

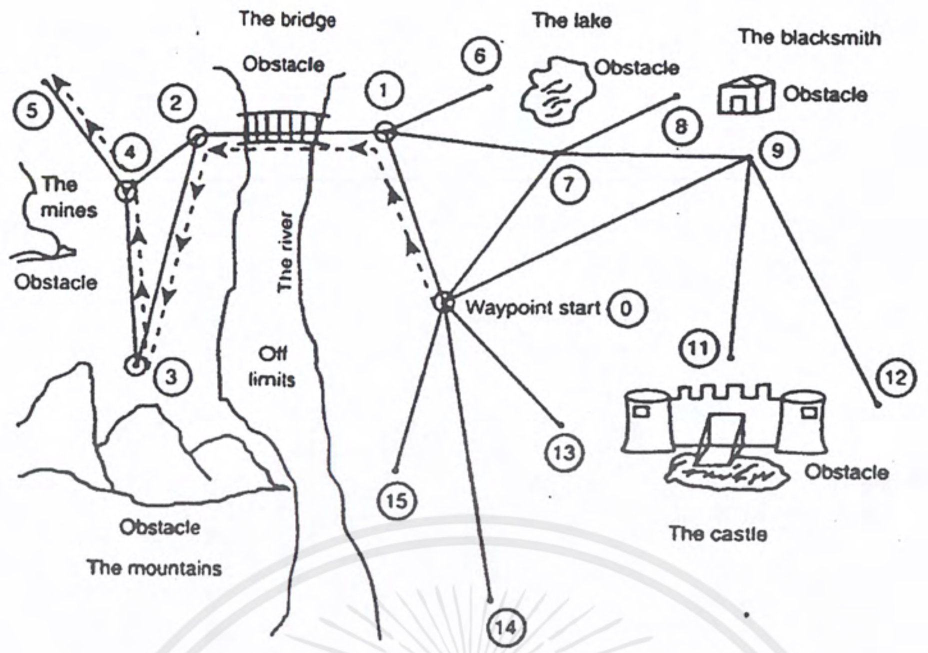


รูปที่ 2.17 เส้นทางการหลบหลีกสิ่งกีดขวาง

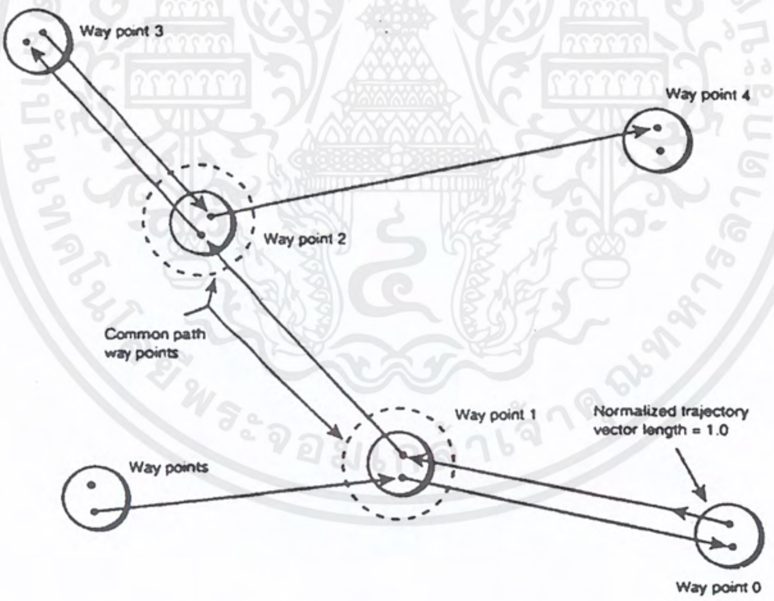
- Waypoint path finding

วิธีนี้จะทำการสร้างเส้นทางการเดิน โดยจะกำหนดจุดหรือโหนด(Node) ที่สนใจและเชื่อมแต่ละจุดด้วยเวคเตอร์ซึ่งแทนเส้นทางการเดินกันเป็นเน็ตเวิร์ค วิธีนี้จะรับประกันได้ว่าจะสามารถไปถึงเป้าหมายได้อย่างแน่นอน และที่สำคัญวิธีจะให้เส้นทางที่ไปยังจุดสำคัญๆ ได้โดยการกำหนดจุดหรือโหนดที่ตำแหน่งที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 เส้นทางเชื่อมต่อถึงกันเป็นเน็ตเวิร์ค

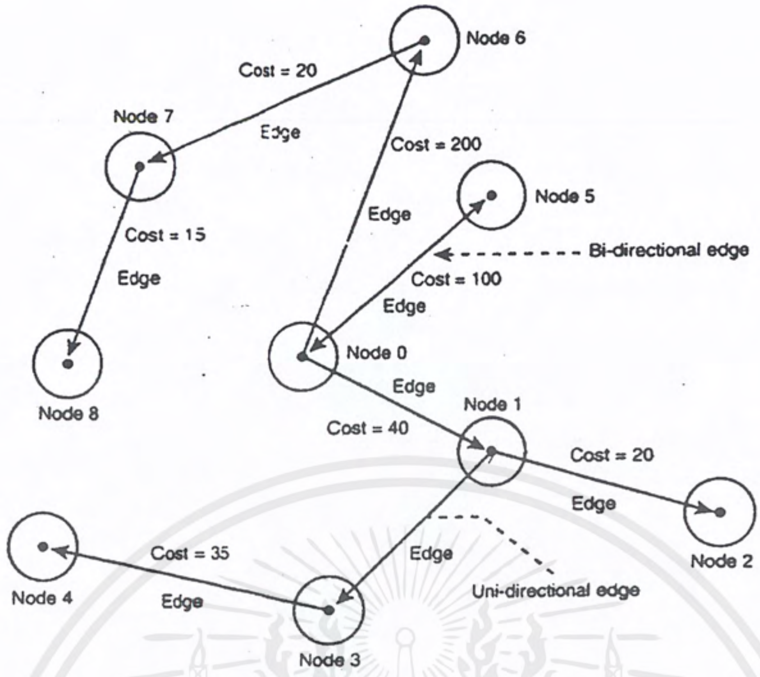


รูปที่ 2.19 เส้นทางที่เชื่อมโหนดโดยใช้จุดร่วมกัน(Waypoints)

- Robust Pathfinding

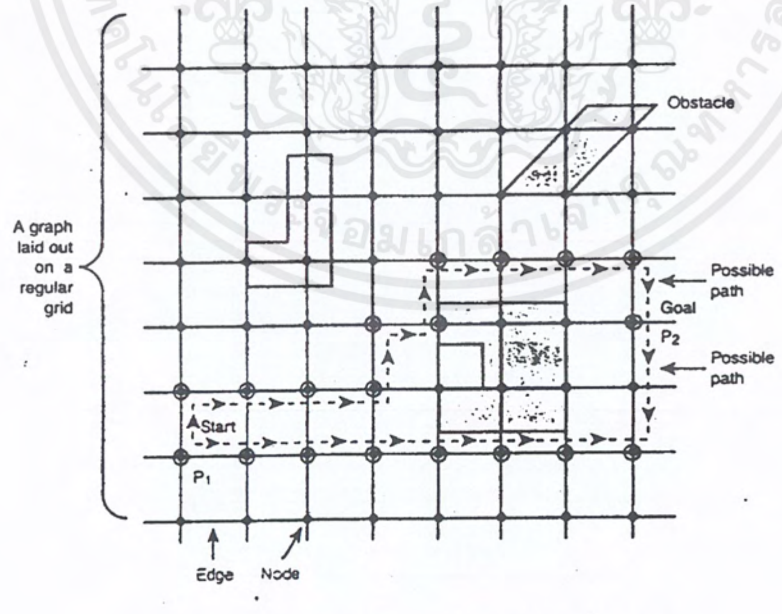
อัลกอริธึมในการค้นหาเส้นทางมีอยู่มากมายหลายวิธี และไม่ว่าจะเป็นวิธีใดก็มีรากฐานความคิดอยู่บนโครงสร้างของกราฟ ซึ่งจะประกอบด้วยโหนดและเส้นเชื่อมต่อโหนด โดยแต่ละโหนดจะไปถึงกันได้ และมักมีการคำนวณค่าสูญเสีย (Cost) ในแต่ละเส้นทางด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงภาพโครงข่ายของกราฟทั่วไป

หรืออาจมองกราฟเป็นการเชื่อมต่อแต่ละจุดที่มีระยะห่างเท่าๆกัน โดยการสร้างตาข่าย (Grid) และในแต่ละช่อง (Cell) ซึ่งอยู่ติดกับอีก 8 ช่องข้างเคียง การคิดค่าสูญเสีย (Cost) ก็คือค่าระยะห่างนั่นเองดังรูป



รูปที่ 2.21 แสดงการใช้ Grid เชื่อมต่อแต่ละโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

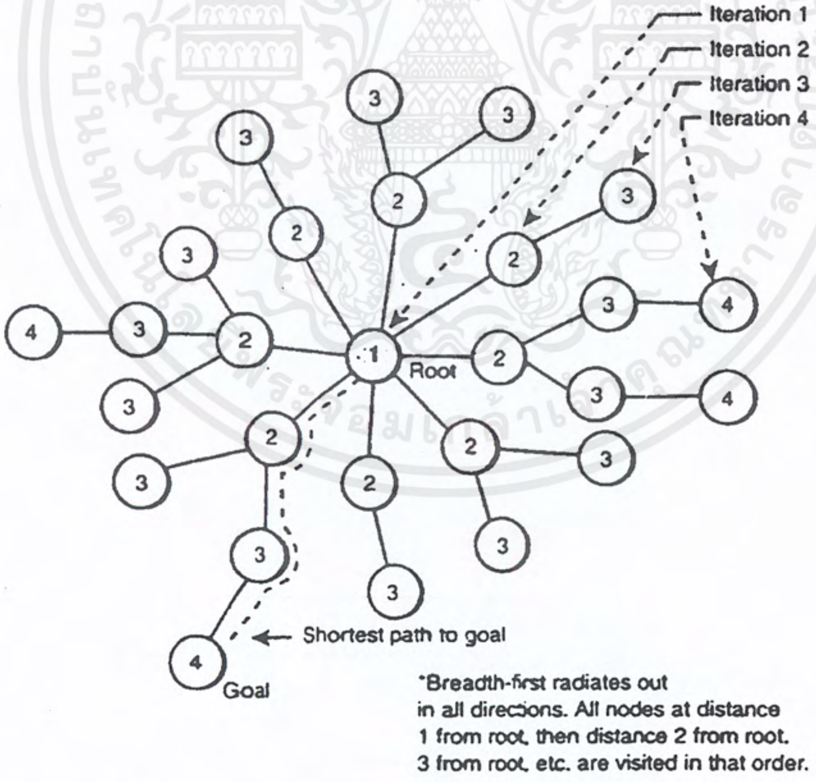
เมื่อสามารถสร้างกราฟที่เชื่อมต่อแต่ละจุดเข้าด้วยกันแล้วก็สามารถค้นหาเส้นทางจากจุดหนึ่งไปยังอีกจุดหนึ่งได้โดยมีค่าสูญเสียน้อยที่สุด เพราะในกราฟนั้นจะไม่รวมวัตถุที่เป็นสิ่งกีดขวางเข้าไปด้วย

อัลกอริธึมในการค้นหาเส้นทางที่นิยมใช้ในระบบคอมพิวเตอร์ได้แก่

- Breadth-first search
- Bidirectional breadth-first search
- Depth-first search
- Dijkstra's search
- A*search

Breadth-First search

วิธีนี้จะกระจายเส้นทางไปยังทุกๆจุดรอบด้านพร้อมๆกันเป็นระยะหนึ่งหน่วย สองหน่วย และสามหน่วยไปเรื่อยๆ จะแผ่กระจายออกไปเป็นวงกลม แบบนี้จะแตกโหนดออกไปอย่างหยาบๆ โดยไม่คำนึงถึงทิศทางที่จะไปถึงเป้าหมายเลย

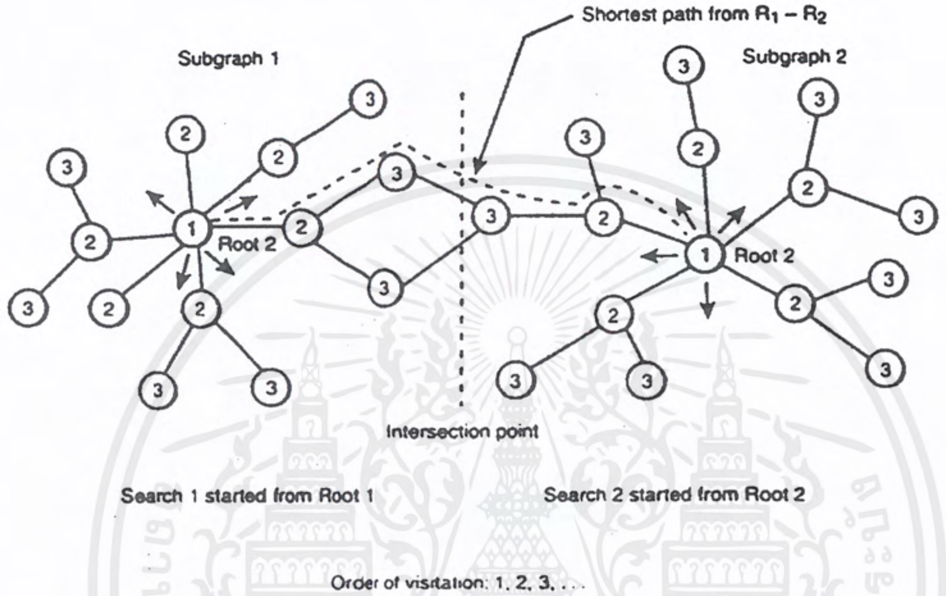


รูปที่ 2.22 แสดงภาพการแตกโหนดของ Breadth-first search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bidirectional Breadth-First search

คล้ายกับวิธี Breadth-First search แต่จะทำการแตกโหนดจาก 2 จุดพร้อมกันซึ่งจุดหนึ่งคือจุดเริ่มต้น และอีกจุดหนึ่งคือเป้าหมายดังรูปที่ 12.26

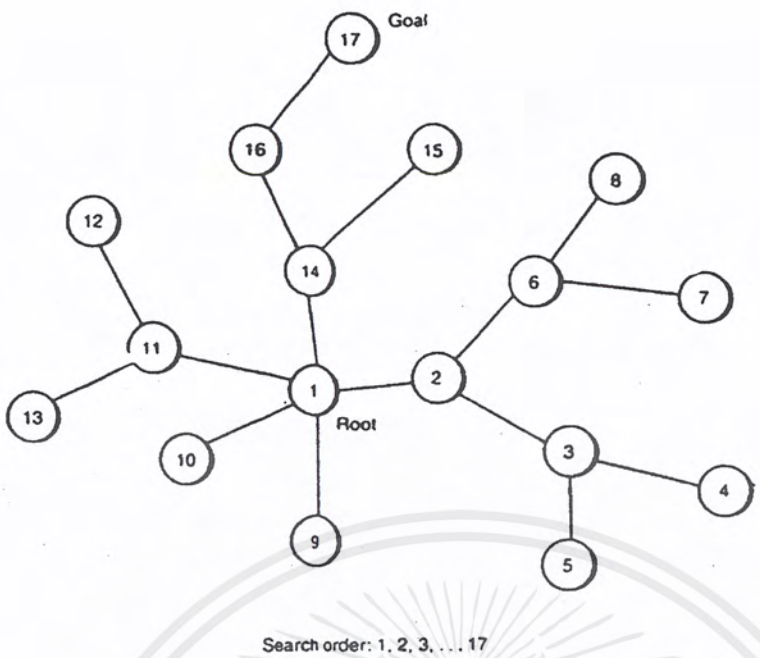


รูปที่ 2.23 แสดงการแตกโหนดจาดจุดเริ่มต้นและจุดเป้าหมาย

Depth-First search

เป็นวิธีกลับกันกับ Breadth-first search เพราะ depth-first search จะทำการค้นหาไปในทิศทางเดียวไปจะกว่าจะไม่สามารถแตกโหนดไปได้อีกหรือจนกว่าจะพบเป้าหมาย แล้วจึงจะลองเส้นทางใหม่ไปเรื่อยๆ วิธีนี้จะเกิดปัญหาถ้ากำหนดความลึกของการค้นหาไม่ดี เพราะอาจทำให้ค้นหาหลงทางได้ จึงจำเป็นต้องกำหนดค่าความลึกของการค้นหา (Depth limit) คือถ้าค้าหาไปถึงความลึกที่กำหนดแล้วยังไม่เจอเป้าหมาย จะต้องเปลี่ยนเส้นทางไปเส้นทางอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 การค้นหาเส้นทางแบบ Depth-first search

Dijkstra's search

ใช้หลักการเดียวกันกับ Minimum Spanning Tree คือจะคำนวณค่าสูญเสีย (Cost) ในทุกการแตกโหนดออกไปยังจุดถัดไป และจะเลือกแตกโหนดไปยังโหนดที่มี cost ต่ำที่สุด

A* search

การsearch มีการคำนวณcostที่จะไปถึงเป้าหมาย เรียกว่า "Greedy search" ซึ่งจะช่วยลดcostได้ ลักษณะคล้ายๆกับ Depth First search จะมุ่งไปในเส้นทางเดียวแล้วจะวกกลับมาเมื่อเจอทางตัน (dead end) ส่วนการsearchที่มีการคิดcostของเส้นทางที่จะไป เรียกว่า "Uniform Cost search" ทั้งสองแบบจะมีข้อดีที่แตกต่างกันไปดังนั้น A* search จึงรวมคุณสมบัติของทั้งสองวิธีเข้าด้วยกัน โดยรวมสมการcostเป็น

$$f(n) = g(n) + h(n)$$

f (n) : costรวมของเส้นทางที่จะไปnode n

g (n): costจากnodeแรกถึงnode n

h (n): costจากnode n ถึงเป้าหมาย

จะทำการเลือกเส้นทางที่มีค่าCostรวมน้อยที่สุด ตามปกติแล้วค่า h (n) ฟังก์ชันจะมีค่าน้อยกว่าความเป็นจริงเสมอ เรียกว่า "Admissible Heuristic" ซึ่งจะทำให้ค่า f (n) ไม่มีทางเกินจริง A*search ช่วยแก้จุดอ่อนของ Greedy searchที่ว่าถ้าเส้นทางนั้นใกล้เป้าหมายก็จริงแต่ถ้าเดินไปแล้วอาจจะมีระยะทางรวมมากกว่าเส้นทางอื่นก็ได้ ส่วนจุดอ่อนของ Uniform cost search คือเส้นทางที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเลือกไปนั้นอาจมีcostน้อยแต่อาจจะไกลจากเป้าหมายมากกว่าก็ได้ เมื่อ A*รวมทั้ง2คุณสมบัติเข้าด้วยกันทำให้มีคุณภาพดีขึ้น

<u>ตัวอย่าง</u>

ต้องการเดินทางจากเมือง Arad ไปเมือง Shibu ด้วยระยะทางที่สั้นที่สุด

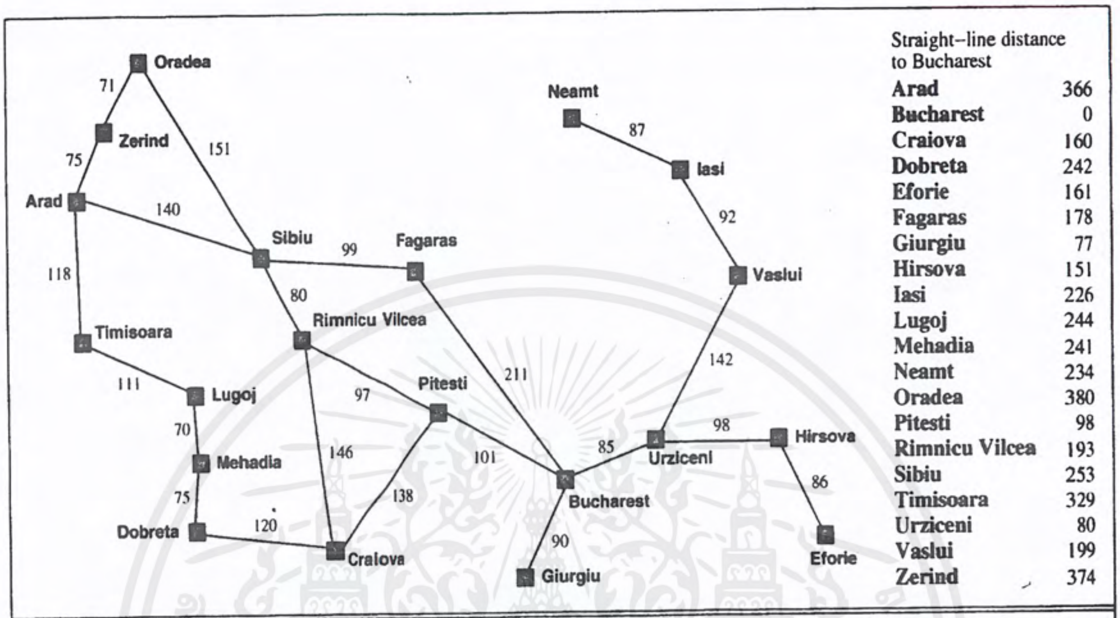
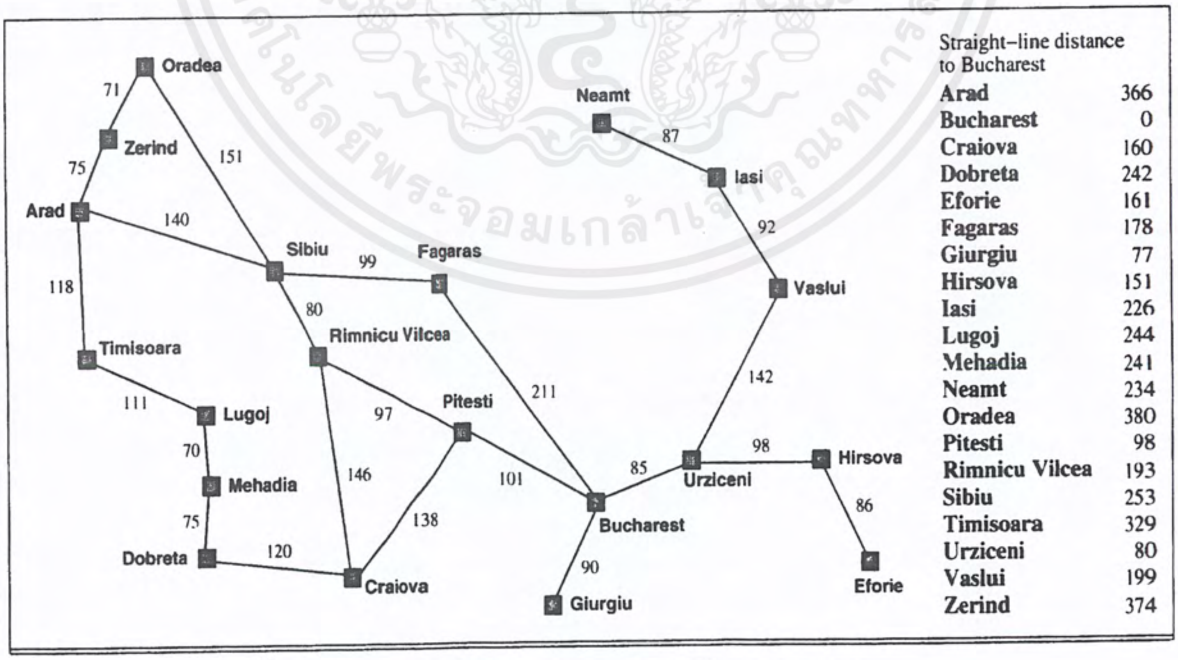


Figure 4.2 Map of Romania with road distances in km, and straight-line distances to Bucharest.

รูปที่ 2.25 แสดงแผนที่ระยะทาง(กิโลเมตร) และระยะกระจัดไปเมือง Bucharest



รูปที่ 2.26 แสดงแผนผังTreeของ A* search โดย f = g + h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเมือง Arad จะเลือกไปเมือง Sibui เพราะมีค่าcostรวมน้อยที่สุด และจากเมือง Sibui จะเลือกไปเมือง Rimnicu เพราะcostรวมน้อยที่สุดเช่นกัน จากเมือง Rimnicu จะเลือกไปเมือง Pitesti และจาก Pitesti จะไปสู่เป้าหมายคือเมือง Bucharest

2.2.4 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรม (I/O Serial Port)

ในโปรแกรมวิชวลเบสิกสามารถทำการติดต่อกับพอร์ตอนุกรมได้โดยใช้ VB Control ที่ชื่อว่า **MSComm** ซึ่งจะต้องทำการ add component ก่อนโดยเข้าไปที่เมนู Project → Components แล้วเลือก control ที่ชื่อว่า Microsoft Comm Control 6 เสร็จแล้วจะปรากฏ icon รูปโทรศัพท์บน tool box ให้เลือกแล้วเอามาวางบนฟอร์ม

<u>การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ</u>

2.2.4.1 การติดต่อแบบอินเตอร์รัพต์

ขบวนการอินเตอร์รัพท์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2.2.4.2 การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า"Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในVBเราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพต์ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive, comEvSend ถ้าเป็นการติดต่อสื่อสารแบบโมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.3 องค์ประกอบในการใช้ MSComm

การตั้งค่าติดต่อกับพอร์ต

- Comport คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1, Com2)
- Setting คือ เราต้องกำหนดอัตรา Baud, Parity, Data (จำนวนบิต), Stop ตัวอย่าง 1200, n, 8, 1
- Handshaking คือ เราจะกำหนดได้ 4 แบบ comNone, comXonXoff, ComRTS, comTRSXonXoff

2.2.4.4 การใช้ Buffer ในการรับส่งข้อมูล

- InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา
-OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป
- Rthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- Sthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- Inputlen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- EOFEnable คือ การที่บอกว่าสิ้นสุดของไฟล์ (EOF) End of File

2.2.4.5 ด้านฮาร์ดแวร์

- ParityReplace คือ ค่าของคาแรกเตอร์ที่จะแทนในเมื่อเกิด Parity Error
- NullDiscard คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- RTSEnable คือ ทำให้มีสัญญาณ RTS (Request To Send)
- DTSEnable คือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

2.2.4.5 การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ต

1. Property ชื่อ Comport คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน ดังตัวอย่างการเขียนโค๊ด

<u>ตัวอย่าง</u> MSComm1.CommPort = 2 (ในที่นี่เลือกจะใช้ Com2 อยู่ที่ด้านหลังเครื่องคอมฯ)

2. Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings = "Baud (อัตราการรับส่งข้อมูล), Parity (ถ้าไม่ใช้ใส่ N, จำนวนบิทข้อมูล, บิทสต๊อบ"

<u>ตัวอย่าง</u> MSComm1.Settings = "9600, N, 8, 1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

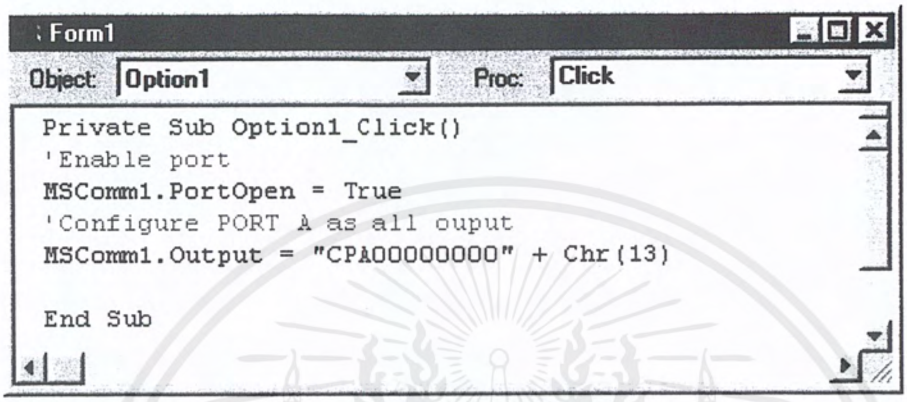
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Property ชื่อ InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์

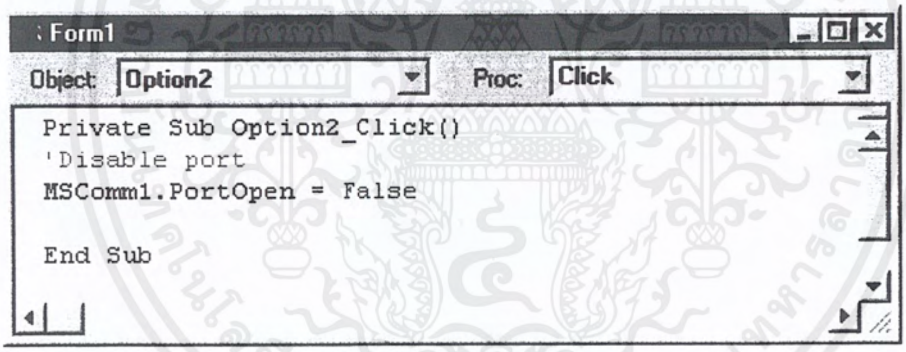
<u>ตัวอย่าง</u> MSComm1.InputLen = 1

4. Property ชื่อ Port Open คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False

<u>ตัวอย่าง</u> MSComm1.PortOpen = True



รูปที่ 2.27 ตัวอย่างการEnable port ให้ใช้งานได้

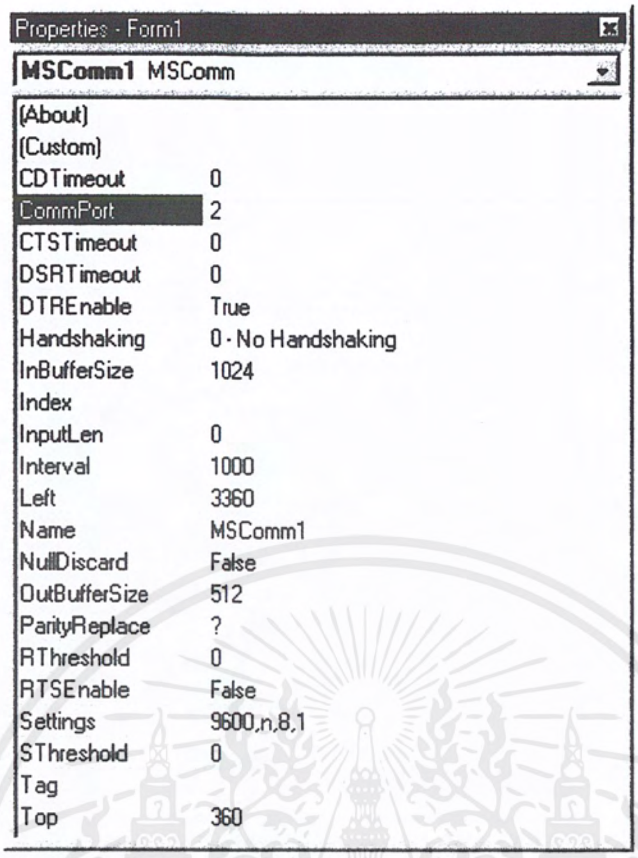


รูปที่ 2.28 ตัวอย่างการ Disable Port

5. Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล (Comport) จะทำให้เกิด CommEvent ใน OnComm Event

<u>ตัวอย่าง</u> MSComm1.Rthreshold =1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 แสดงการเซ็ตค่าพร็อพเพอร์ตี้ต่างๆ ในการติดต่อกับพอร์ตอนุกรม

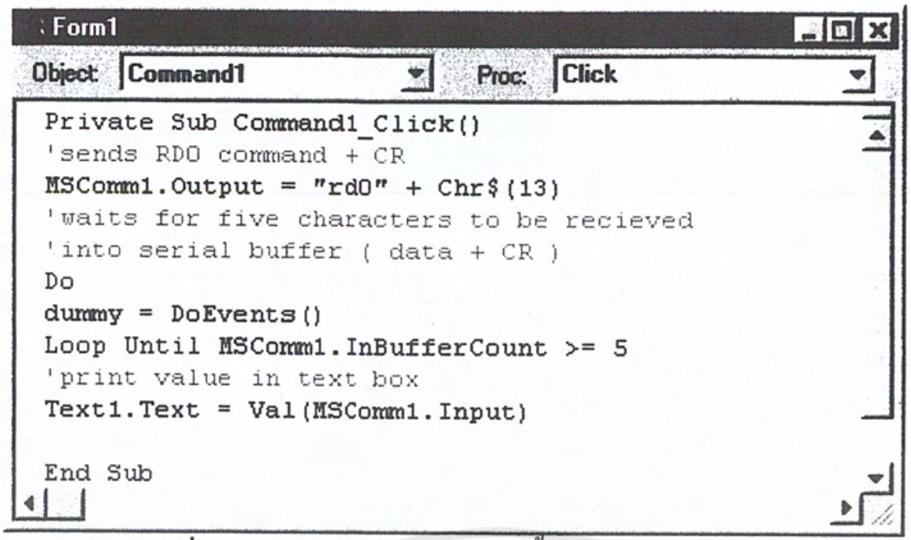
2.2.4.6 วิธีของการรับส่งข้อมูลจาก Serial Port

ด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

Output = ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input = เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS232 นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 แสดงการใช้พร็อพเพอร์ตี้อินพุทและเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

จากทฤษฎีทั้งหมดที่ศึกษามานำมาสร้างเป็นอัลกอริธึมที่ใช้ในการสร้างเส้นทางไปยังวัตถุเป้าหมาย โดยหลีกเลี่ยงสิ่งกีดขวางและเป็นเส้นทางที่เหมาะสมที่สุดหรือระยะทาง(Cost) น้อยที่สุด จากนั้นก็สร้างอัลกอริธึมในการควบคุมรถให้เดินทางไปยังเป้าหมาย และส่งคำสั่งในการควบคุมรถให้เดินทางไปในทิศทางที่ต้องการ ดังนั้นจึงสามารถแบ่งส่วนของซอฟท์แวร์เป็น 2 ส่วนดังนี้

3.1 อัลกอริธึมการวิเคราะห์สีของภาพโดยใช้ทฤษฎีเอชเอสวี (HSV)

เมื่อโปรแกรมทำการรับภาพเฟรมวิดีโอมาจากกล้องดิจิตอลจะทำการคำนวณหาค่าพารามิเตอร์สีอาร์จีบีของแต่ละพิกเซล แล้วนำมาแปลงเป็นค่าเอชเอสวี (HSV) โดยใช้อัลกอริธึมที่ใช้ในการแปลง

3.2 อัลกอริธึมในการสแกนหาตำแหน่งสิ่งกีดขวางและตำแหน่งรถ

เมื่อได้ค่าเอชเอสวีแล้วจะสามารถเลือกสแกนเฉพาะสีที่ต้องการ ได้ ในที่นี้จะกำหนดให้สีน้ำเงินเป็นสีของสิ่งกีดขวาง สีแดงและเขียวเป็นสีของหัวรถ ท้ายรถ

1) การสแกนจะสแกนหาค่ามุมฮิวที่อยู่ในช่วงสีน้ำเงินแล้วทำการเก็บจุดทุกจุดที่เป็นสีน้ำเงินไว้ เพื่อมาทำการจัดกลุ่ม (Clustering) ด้วยอัลกอริธึมเคมีน (K-mean)
2) สแกนหาค่ามุมฮิวที่อยู่ในช่วงสีแดงและสีเขียว แล้วนำพิกัดจุดที่ได้ไปหาจุดศูนย์กลางของทั้งสีแดงและสีเขียว นำจุดศูนย์กลางทั้งสองไปหาความชันของรถ ซึ่งการสแกนตำแหน่งของรถจะสแกนจากทุกเฟรมภาพที่รับมาด้วยอัตราเร็ว(refresh) ทุกๆ 100(μSec) (10 ภาพต่อวินาที)

3.3 อัลกอริธึมในการจัดกลุ่ม (Clustering)

พิกัดจุดทุกจุดที่เป็นสีน้ำเงินไว้ นำมาทำการจัดกลุ่ม (Clustering) ด้วยอัลกอริธึมเคมีน (K-mean) ออกเป็น25กลุ่ม โดยแบ่งกลุ่มตามตำแหน่งพิกัด แล้วทำการตัดกลุ่มที่มีสมาชิกน้อยๆออก เพราะถือว่าเป็นสัญญาณรบกวน (noise) ในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การค้นหาเส้นทาง (Path Finding) ด้วยอัลกอริธึมคอนทัวร์เทรสซิ่ง (Contour Tracing)

จากพิกัดจุดสีน้ำเงินของสิ่งกีดขวางและพิกัดจุดศูนย์กลางของรถและเป้าหมายนำมาพล็อตลงในภาพ

1. สแกนทั้งภาพเพื่อทำการติดตามขอบ(Edge detection) แล้วเก็บพิกัดขอบของสิ่งกีดขวางนั้นไว้
2. นำพิกัดจุดศูนย์กลางรถ และรับอินพุทของพิกัดเป้าหมาย นำมาสร้างสมการเส้นตรงซึ่งเป็นเส้นทางที่รถจะเดินไป
3. นำพิกัดขอบของสิ่งกีดขวางแต่ละชิ้นมาแทนในสมการเส้นตรงเพื่อตรวจสอบว่าสิ่งกีดขวางนั้นอยู่บนแนวเส้นตรงหรือไม่ ถ้าไม่มีสิ่งกีดขวางบนแนวเส้นทางเลย ให้กำหนดจุดเป้าหมายเป็นเวย์พอยนท์ที่จะให้รถเดินไป
4. ถ้ามีสิ่งกีดขวางอยู่บนเส้นทางให้หาพิกัดของสิ่งกีดขวางที่อยู่ใกล้ที่สุดก่อน แล้วกำหนดพิกัดบนด้านของสิ่งกีดขวางนั้นเป็นเวย์พอยนท์ถัดไปที่จะให้รถเดินไป
5. เลือกจุดมุมที่อยู่บนด้านเดียวกันกับจุดเวย์พอยนท์ก่อนหน้านี้ ว่าที่มุมใหนจะใกล้กับเป้าหมายมากกว่า แล้วทำการกำหนดจุดมุมนั้นให้เป็นเวย์พอยนท์ถัดไป
6. สร้างสมการเส้นตรงที่เป็นเส้นทางจากจุดมุมนั้นไปยังเป้าหมาย
7. นำพิกัดขอบของสิ่งกีดขวางแต่ละชิ้นมาแทนในสมการเส้นตรงเพื่อตรวจสอบว่ามีสิ่งกีดขวางชิ้นอื่นอยู่บนเส้นทางหรือไม่
    - ถ้าไม่มีสิ่งกีดขวางอยู่บนเส้นทางเลยให้กำหนดเป้าหมายเป็นจุดเวย์พอยนท์ถัดไป
    - ถ้ามีสิ่งกีดขวางอยู่บนเส้นทางให้ตรวจดูว่าเป็นสิ่งกีดขวางชิ้นเดิมหรือไม่
8. ถ้าสิ่งกีดขวางชิ้นใหม่อยู่บนเส้นทางกลับไปคำนวณสร้างเส้นทางในข้อ 2 โดยคราวนี้กำหนดพิกัดของรถเป็นตำแหน่งจุดเวย์พอยนท์ล่าสุด
9. ถ้าเส้นทางผ่านสิ่งกีดขวางชิ้นเดิมให้กำหนดมุมถัดไปที่อยู่บนด้านเดียวกันกับจุดเวย์พอยนท์ก่อนหน้านี้เป็นจุดเวย์พอยนท์ถัดไป แล้วกลับไปสร้างเส้นทางในข้อ 2 โดยคราวนี้กำหนดพิกัดของรถเป็นจุดเวย์พอยนท์ล่าสุด

3.5 การค้นหาเส้นทาง (Path Finding) ด้วยอัลกอริธึมคอนทัวร์เทรสซิ่ง (Contour Tracing) และเอสตาร์ (A-star)

1. หลังจากทำการติดตามขอบ(Edge detection) ได้พิกัดขอบของสิ่งกีดขวาง นำไปหาค่ามากสุด น้อยสุดของขอบ
2. หาพิกัดมุมของสิ่งกีดขวาง 4 มุม
3. สร้างเส้นทางกระจัด(Straight line distance) จากพิกัดรถถึงพิกัดเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการตรวจสอบว่ามีสิ่งกีดขวางบนเส้นทางหรือไม่โดยแทนค่าพิกัดในสมการเส้นตรง
5. ถ้าไม่มีสิ่งกีดขวางบนเส้นทางให้สร้างโหนดจากพิกัดโหนด(โหนดแรกเป็นพิกัดของรถ) แล้วเก็บค่าพิกัดนั้นเป็นเส้นทาง
6. ถ้ามีสิ่งกีดขวางบนเส้นทางให้สร้างโหนดจากพิกัดรถไปยังพิกัดมุมของสิ่งกีดขวาง
7. ตรวจสอบว่าสร้างโหนดได้หรือไม่
8. ถ้าทางตันไม่สามารถสร้างโหนดได้ ให้ทำการถอยหลังไปหนึ่งระดับ
9. ถ้าทางไม่ตันทำการคำนวณค่าคอสท์ของทุกเส้นทาง ตามสมการ

$$f(n) = g(n) + h(n)$$

f(n) : costรวมของเส้นทางที่จะไปnode n

g(n): costจากnodeแรกถึงnode n

h(n): costจากnode n ถึงเป้าหมาย

10. เลือกโหนดที่มีค่าคอสท์น้อยที่สุด กำหนดโหนดนั้นเป็นเส้นทางที่จะเดินไป
11. ตรวจสอบว่าโหนดนั้นเป็นพิกัดเป้าหมายหรือยัง
12. ถ้ายังไม่ใช่เป้าหมายให้กลับไปทำข้อที่ 3
13. ถ้าถึงเป้าหมายแล้ว จบโปรแกรม

3.6 อัลกอริธึมในการควบคุมรถ

โปรแกรมจะทำการสั่งให้รถเดินไปตามตำแหน่งเวย์พอยนท์ที่ได้จากการค้นหาเส้นทางโดยจะเปรียบเทียบตำแหน่งของรถว่าอยู่ในบริเวณที่เป็นเวย์พอยนท์หรือยัง ถ้ายังก็จะตรวจสอบว่าทิศทางของรถมุ่งหน้าไปยังเวย์พอยนท์นั้นแล้วหรือไม่ โดยการเปรียบเทียบจะคำนวณหาค่าความชัน (Slope) โดยถ้าความชันของรถผิดพลาดไปไม่เกิน 5 องศา ก็จะสั่งให้รถเดินหน้าไปเรื่อยๆ แต่ถ้าความชันของรถนั้นหลุดไปจากช่วงที่ต้องการจะสั่งให้รถหยุดและทำการหมุนให้รถมีทิศทางตรงกับเวย์พอยนท์นั้น

เมื่อรถเดินทางไปถึงบริเวณที่เป็นเวย์พอยนท์แล้วจะตรวจสอบว่าตำแหน่งนั้นเป็นเวย์พอยนท์สุดท้ายแล้วหรือยัง ถ้าใช่(เป็นตำแหน่งเป้าหมายสุดท้าย) ก็จะสั่งควบคุมให้รถหยุด แต่ถ้ายังไม่ใช่เวย์พอยนท์สุดท้ายก็จะกำหนดเวย์พอยนท์ถัดไปเพื่อควบคุมให้รถเดินทางต่อไป

การควบคุมรถทำได้โดยการส่งรหัสไปสั่งล้อให้หมุนไปข้างหน้าหรือถอยหลัง ซึ่งจะควบคุมแยกกัน 2 ล้อโดยที่

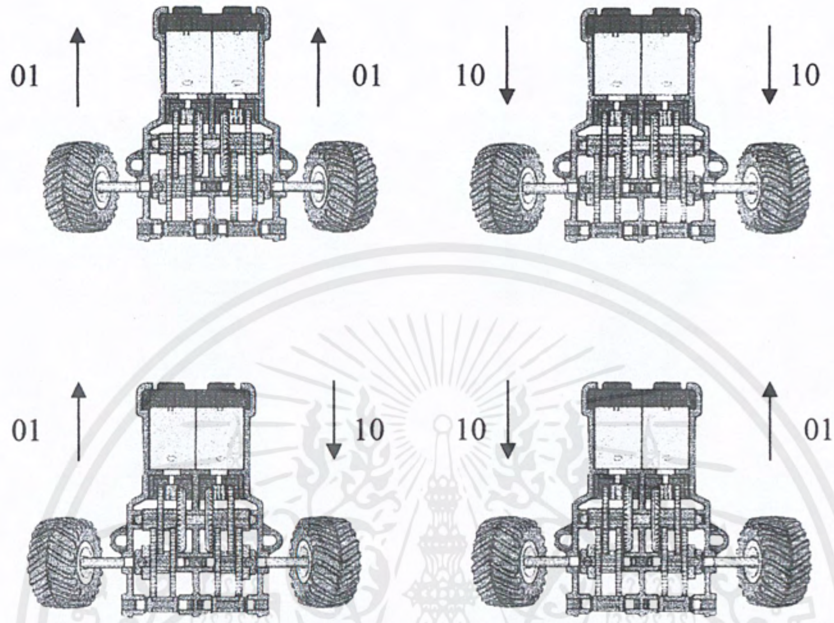
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งรหัส 0101 → เดินหน้า

ส่งรหัส 1010 → ถอยหลัง

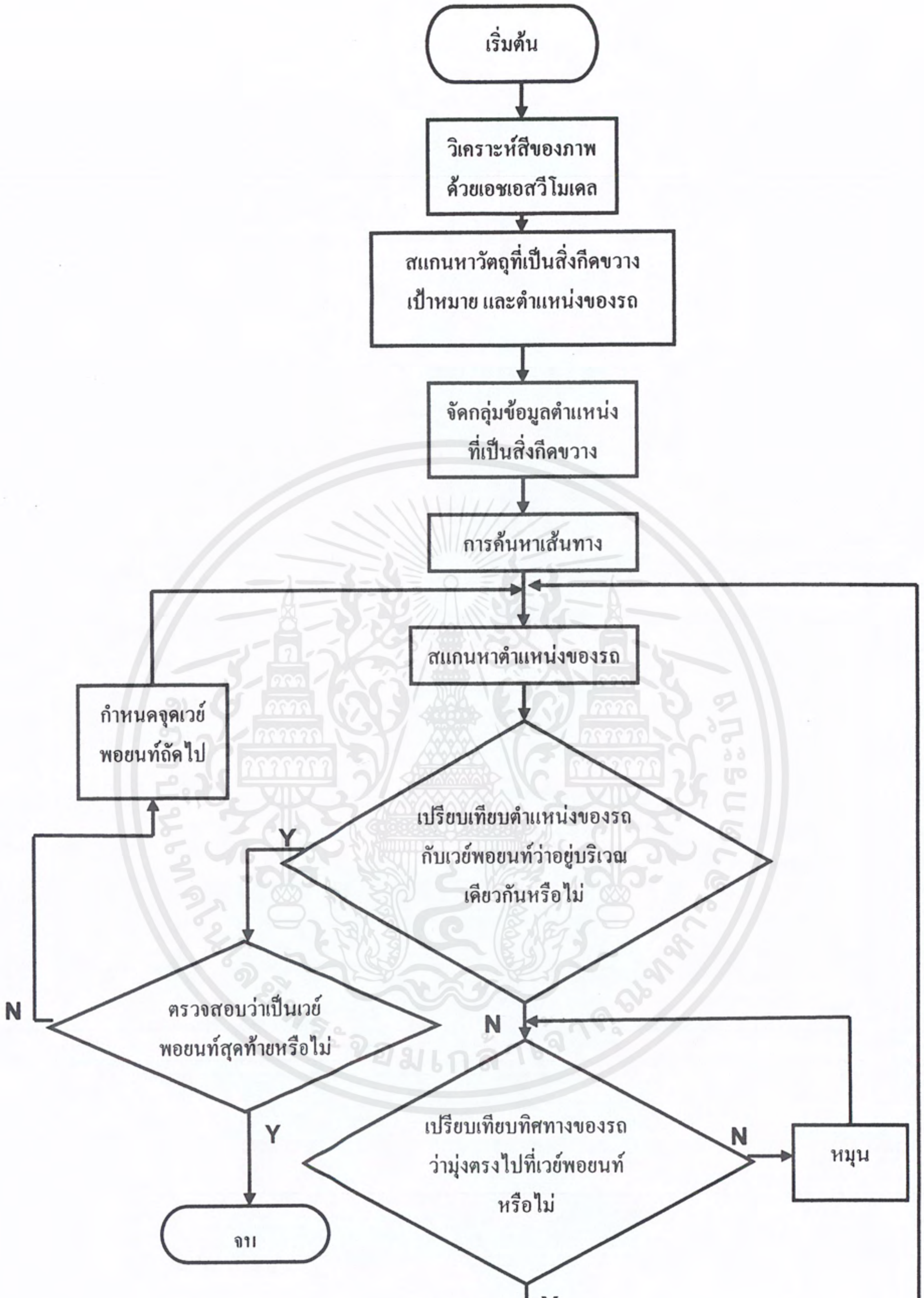
ส่งรหัส 0110 → เลี้ยวขวา

ส่งรหัส 1001 → เลี้ยวซ้าย



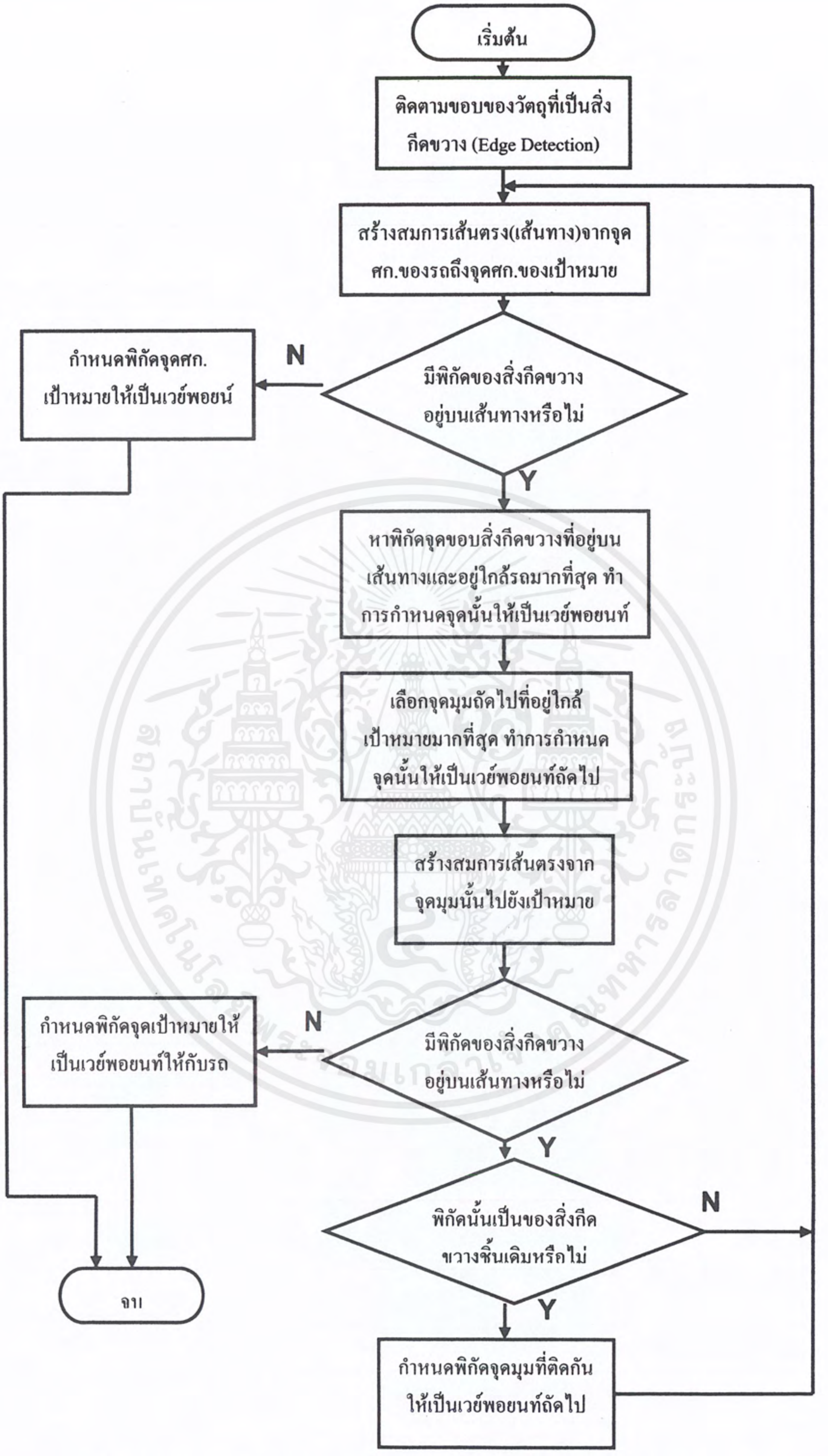
รูปที่ 3.1 แสดงการควบคุมล้อรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงแผนผังการทำงานของโปรแกรม

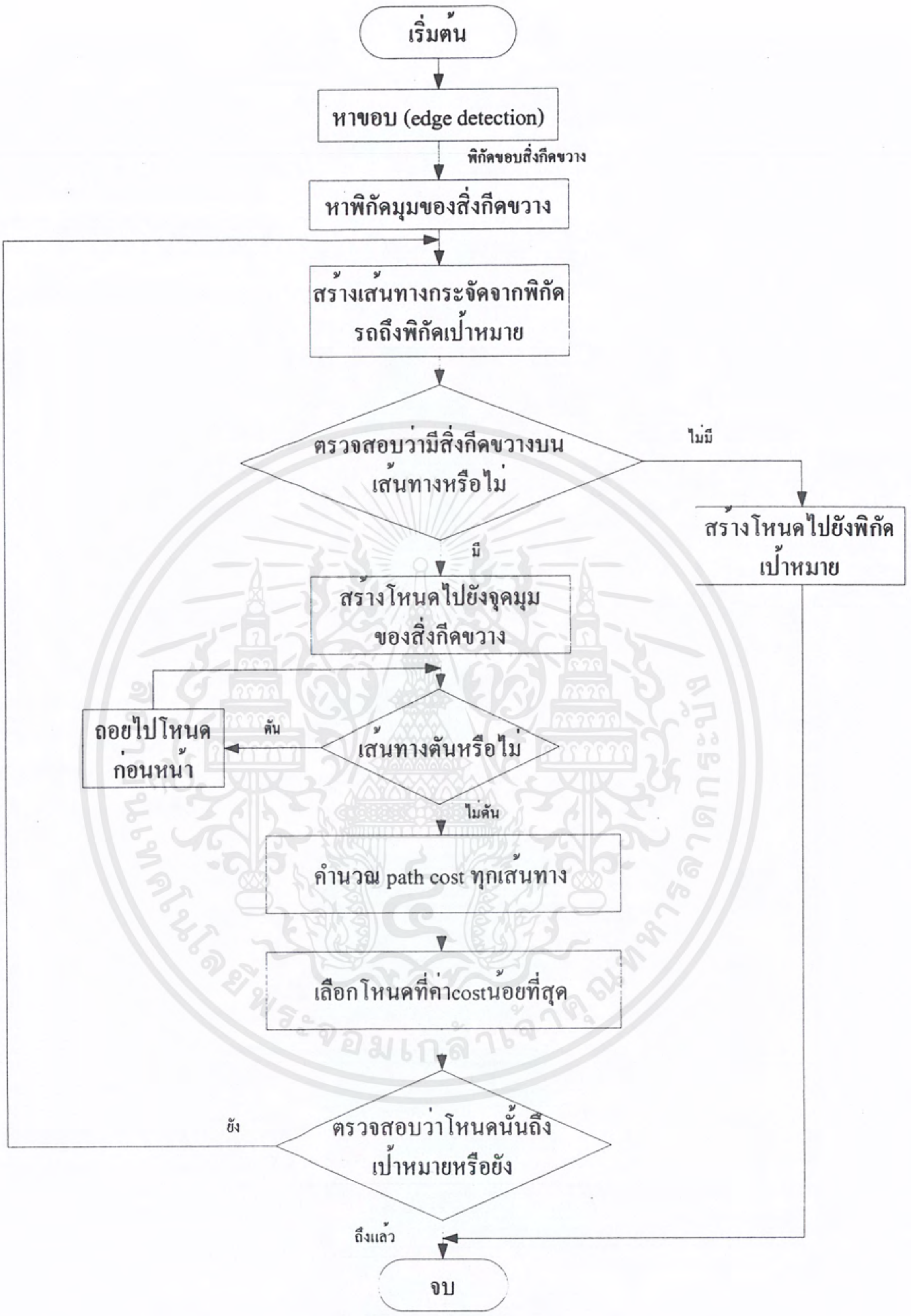
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงแผนผังการทำงานของโปรแกรมส่วนค้นหาเส้นทาง (Path finding)

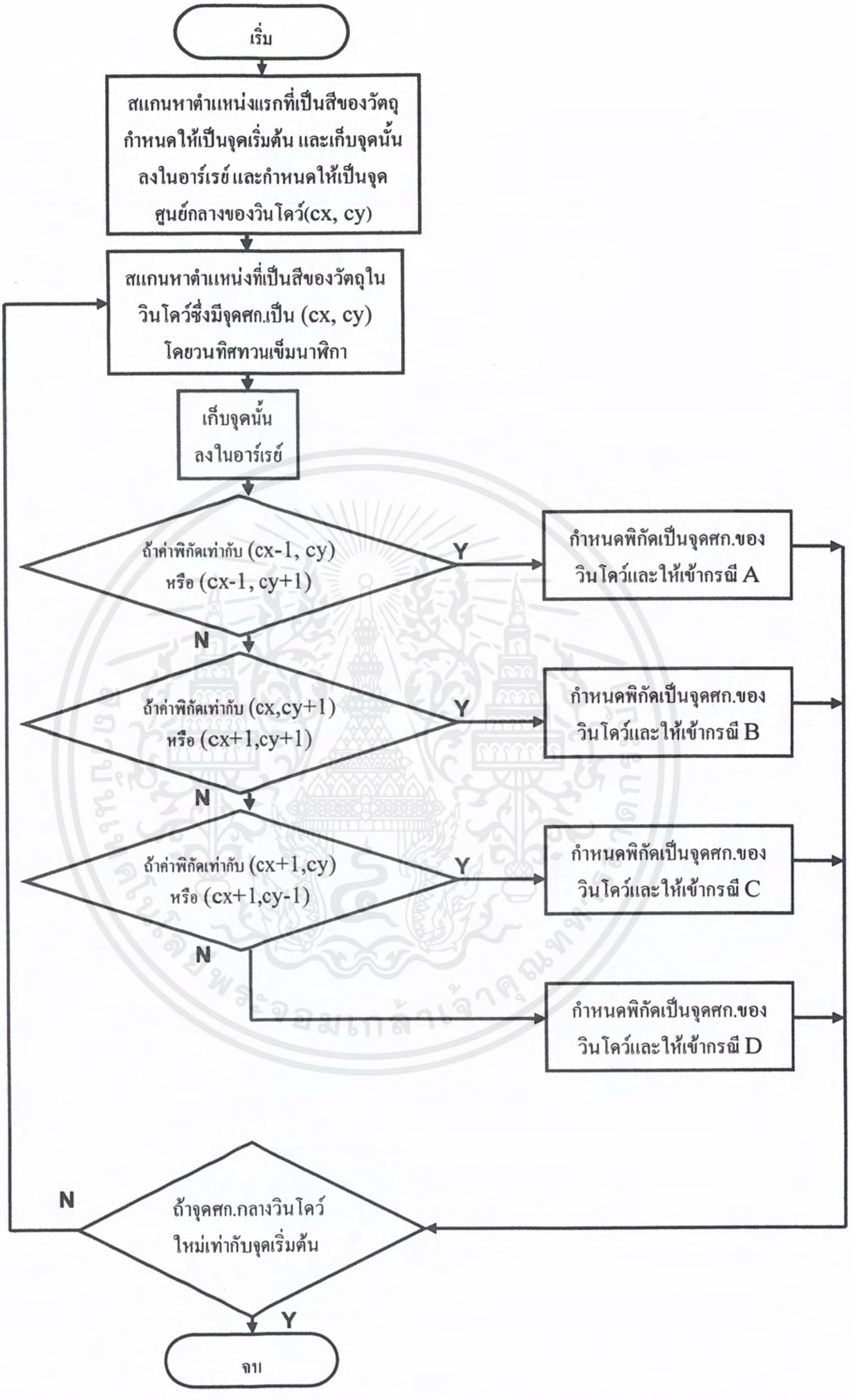
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงแผนผังการทำงานของโปรแกรมการค้นหาเส้นทางแบบคอนทัวร์เทรสซิ่งและเอสตาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงแผนผังการทำงานของโปรแกรมส่วนตัดขอบ (Edge Detection)

เอกสารนี้เป็นเอกสารที่ใช้ในการศึกษาเพื่อการศึกษาเท่านั้น ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ การทำงานในส่วนตัดหาขอบ (Edge Detection) คือจะทำการสร้างวินโดว์ขนาด 3 x 3 พิกเซลซึ่งมีจุดศูนย์กลางที่พิกัด (cx, cy) แล้วเลื่อนวินโดว์ไปเพื่อสแกนหาตำแหน่งที่เป็นส่วนของวัตถุ โดยการวนในวินโดว์นั้นจะมีทิศทวนเข็มนาฬิกาโดยเริ่มที่ตำแหน่งต่างๆตามกรณีที่เข้าจนครบทั้งวินโดว์

โดย กรณี A จะเริ่มสแกนที่ตำแหน่งทางด้านบนซ้าย (1)
กรณี B จะเริ่มสแกนที่ตำแหน่งทางด้านล่างซ้าย (3)
กรณี C จะเริ่มสแกนที่ตำแหน่งทางด้านล่างขวา (5)
กรณี D จะเริ่มสแกนที่ตำแหน่งทางด้านบนขวา (7)

1	8	7
2	CX,CY	6
3	4	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลลัพธ์

4.1 การทดลอง

1. จัดสนามจำลองขนาดประมาณ 90 x 120 ตารางเซนติเมตร และติดตั้งกล้องที่ความสูงจากพื้นสนาม 1.5 เมตร โดยจัดตำแหน่งกล้องให้จับภาพจากด้านบน (Top view)

2. รับภาพเฟรมวิดีโอจากกล้องดิจิตอลแล้วมาคำนวณค่าพารามิเตอร์ต่างๆเพื่อปรับเฉดสีและความเข้มแสงที่ต้องการ และนำข้อมูลตำแหน่งของวัตถุที่เป็นสิ่งกีดขวางมาทำการจัดกลุ่มข้อมูลเพื่อลดสัญญาณรบกวน

3. เมื่อได้ตำแหน่งของสิ่งกีดขวาง จะมาทำการค้นหาเส้นทางเพื่อหลบหลีกสิ่งกีดขวางเหล่านั้นแล้วสั่งควบคุมให้รถเคลื่อนที่ไปตามเส้นทางไปยังเป้าหมาย

4.2 ผลการทดลอง

เฟรมที่ 1 แสดงภาพที่ได้จากกล้องดิจิตอล

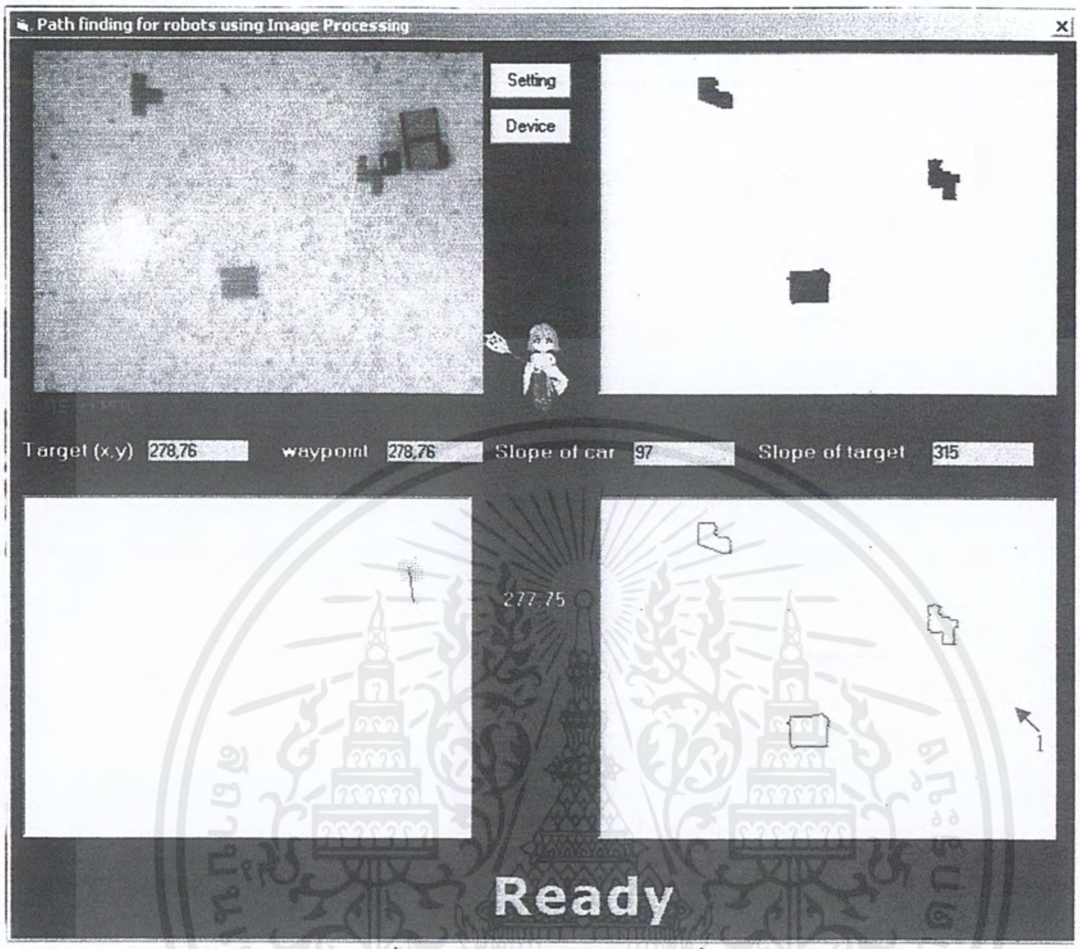
เฟรมที่ 2 แสดงผลการจัดกลุ่มข้อมูลพิกัดของสิ่งกีดขวาง

เฟรมที่ 3 แสดงสถานะของรถ โดยสีแดงหมายถึงหัวรถ และสีเขียวหมายถึงท้ายรถ เส้นสีดำหมายถึงเส้นทางไปยังจุดเป้าหมายสุดท้าย เส้นสีฟ้าหมายถึงเส้นทางที่จะไปยังจุดเวย์พอยนท์

เฟรมที่ 4 แสดงผลการค้นหาเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1ผลการทดลองที่ 1

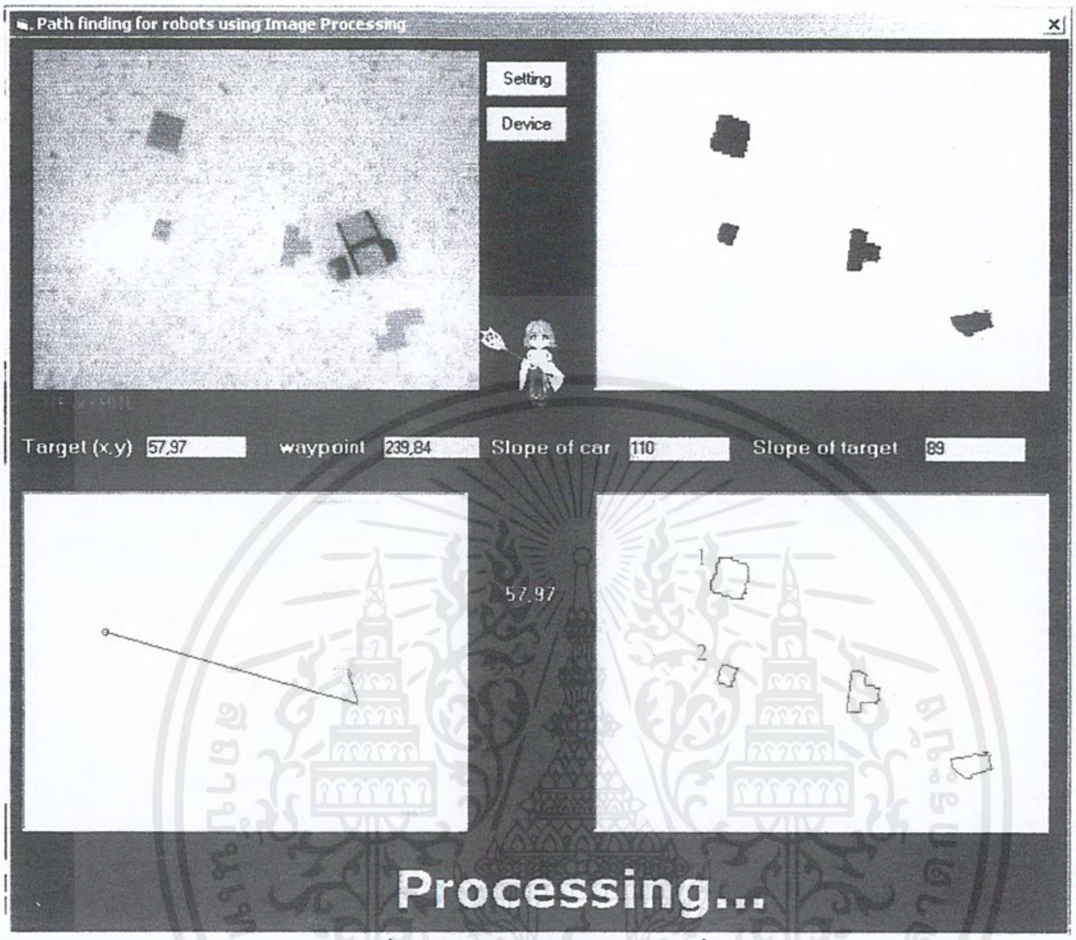


รูปที่ 4.1ตัวอย่างผลการทดลองที่ 1

เมื่อรับภาพจากเฟรมที่ 1 แล้วทำการจัดกลุ่มข้อมูลแล้วได้วัตถุสิ่งกีดขวาง 3 ชิ้นทำการติดตามขอบเพื่อจะนำมาหาจุดมุมของสิ่งกีดขวาง และสร้างเส้นทางไปบนจุดมุมของสิ่งกีดขวาง โดยเลือกเส้นทางที่มีค่าคอสท์น้อยที่สุด จากภาพผลการทดลองนี้จากจุดเริ่มต้นเมื่อแตกไปยังโหนดถัดไปจะเห็นว่า โหนดที่ 1 มีค่าคอสท์น้อยที่สุดก็จะสร้างเส้นทางไปยังโหนดนั้น และจะนำไปควบคุมรถให้เดินไปตามเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองที่ 2

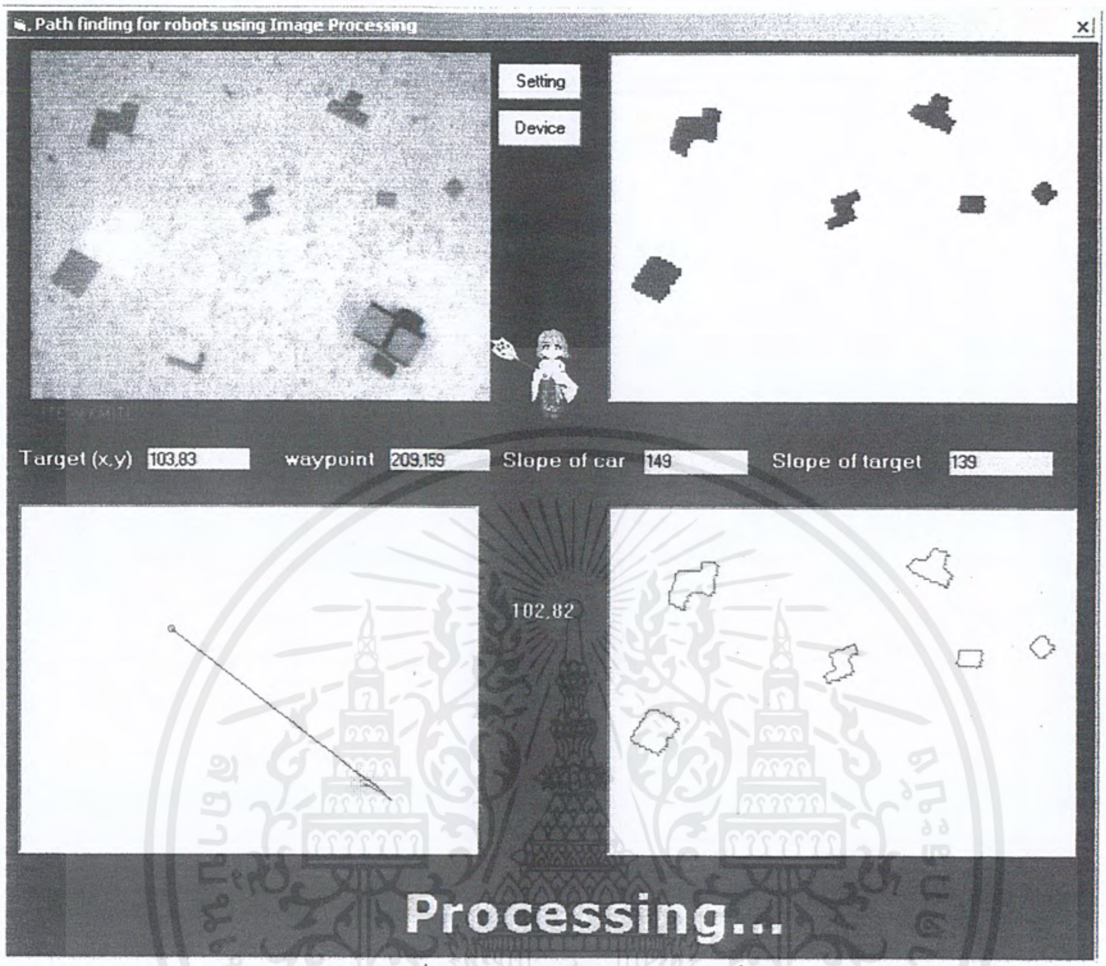


รูปที่ 4.2 ตัวอย่างผลการทดลองที่ 2

จากเฟรมที่ 4 จะเห็นว่าเส้นทางจะอ้อมเพราะว่าวัตถุชิ้นที่ 1 และ 2 อยู่ใกล้กันเกินไปทำให้รถไม่สามารถเดินผ่านไปได้ จึงต้องใช้เส้นทางอ้อมมาทางโหนดที่ 4 ในเฟรมที่ 3 แสดงสถานะของรถกำลังหมุนให้ทิศทางไปตรงกับเส้นสีฟ้าซึ่งเป็นจุดเวย์พอยนท์ถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองที่ 3

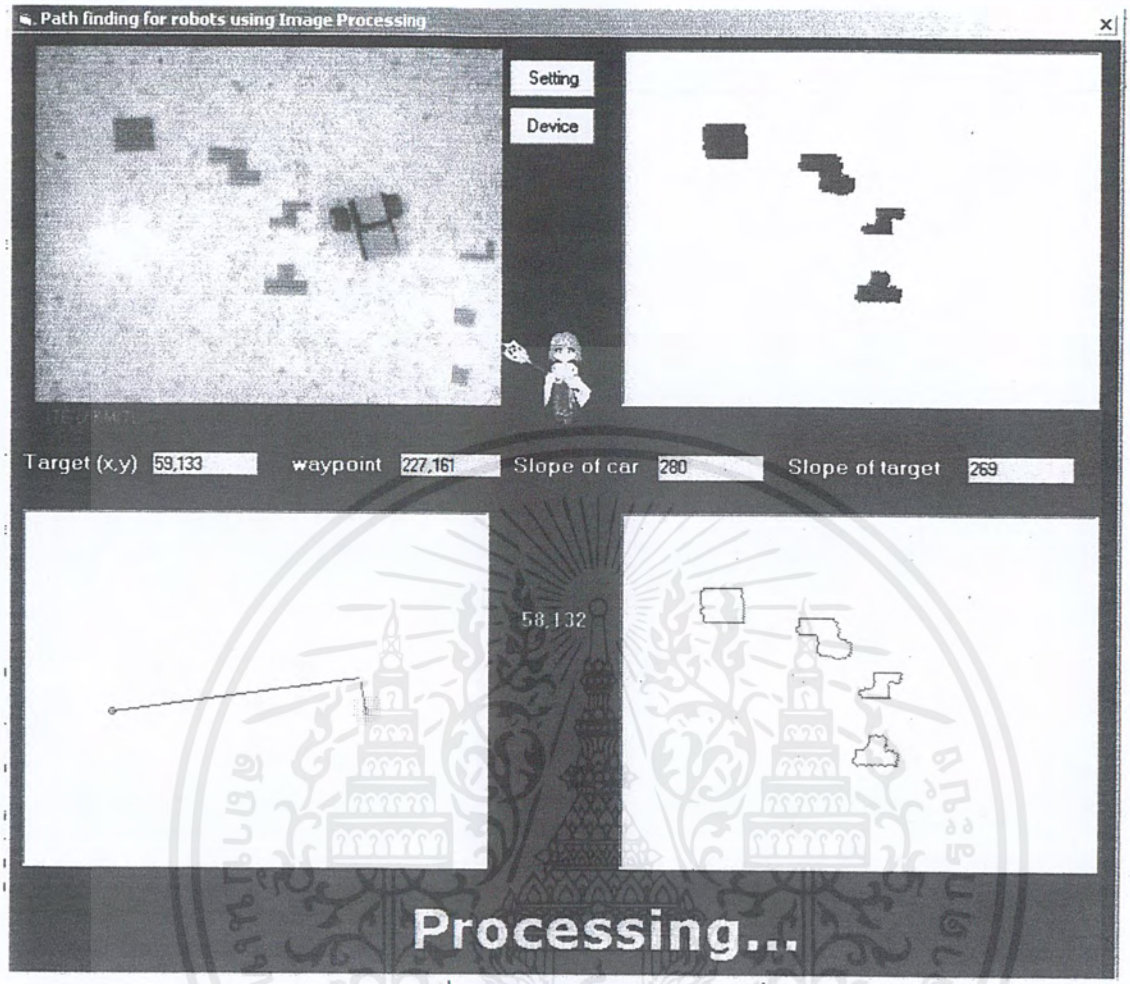


รูปที่ 4.3 ตัวอย่างผลการทดลองที่ 3

เมื่อเพิ่มจำนวนสิ่งกีดขวางให้มากขึ้น ก็ยังสามารถสร้างเส้นทางหลบสิ่งกีดขวางไปได้ จากเฟรมที่3 จะเห็นสถานะของรถปรับทิศจนตรงแล้วกำลังจะเริ่มเดินไปยังเวย์พอยนท์ถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ผลการทดลองที่ 4

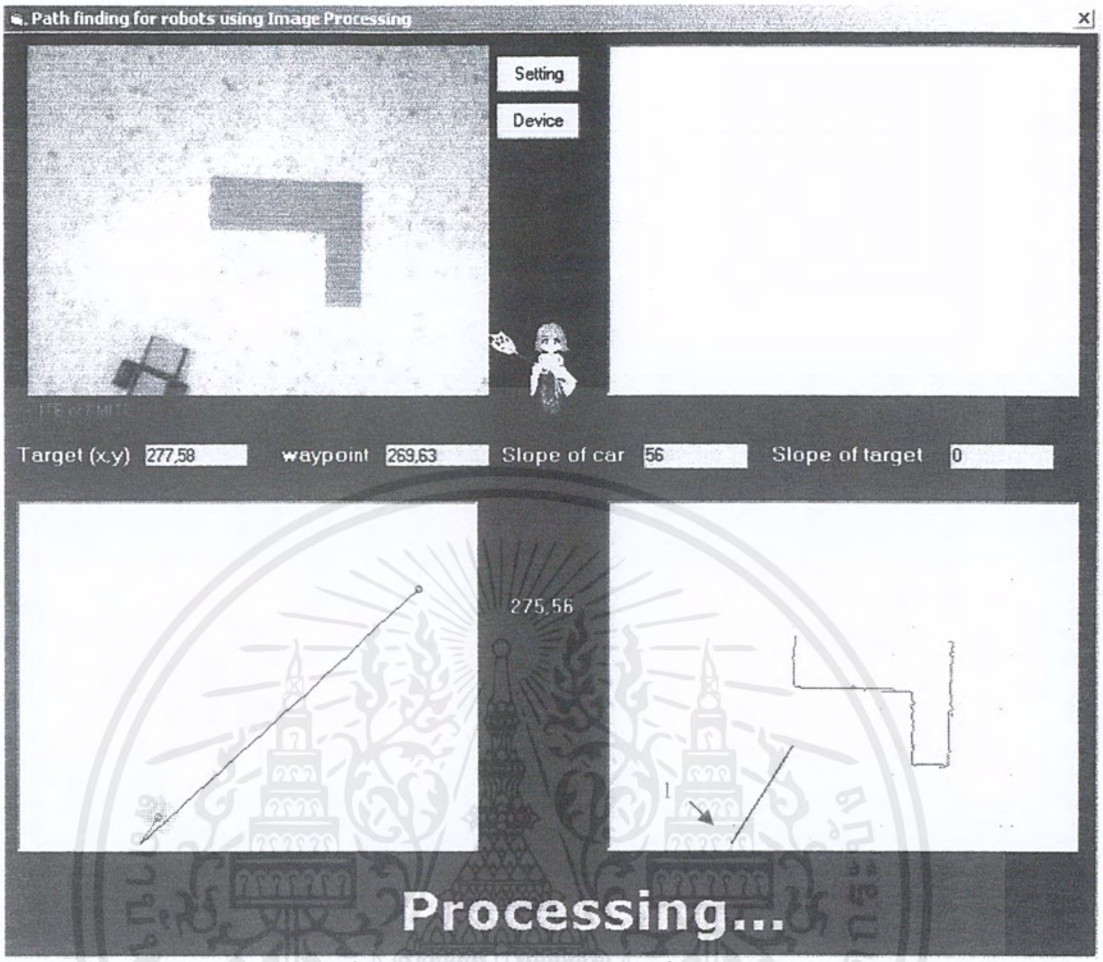


รูปที่ 4.4 ตัวอย่างผลการทดลองที่ 4

จากเฟรมที่ 2 ผลจากการจัดกลุ่มข้อมูลทำให้วัตถุสิ่งกีดขวางทางด้านขอบขวามือหายไปเนื่องจากเป็นชิ้นเล็กๆ และจากเฟรมที่ 4 เส้นทางเดินอ้อมไปทางด้านล่างเพราะไม่สามารถเดินผ่านไปทางช่องแคบด้านบนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 ผลการทดลองที่ 5



รูปที่ 4.5 ตัวอย่างผลการทดลองที่ 5

จากเฟรมที่ 4 จะเห็นได้ว่าเริ่มแรกจะสร้างทางเดินไปตามแนวสีดำแต่เมื่อแตกโหนดแล้วไปชนกับส่วนที่เป็นสิ่งกีดขวาง ทำให้ไปต่อไม่ได้จึงต้องย้อนกลับมาที่จุด 1 และเลือกแตกโหนดไปอีกทางหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปผลการทดลอง

การนำทฤษฎีสีมาใช้ในการวิเคราะห์ภาพสามารถแยกแยะขอบของวัตถุที่ต้องการแต่ยังมีสัญญาณรบกวนที่ไม่สามารถกำจัดได้บ้าง เนื่องจากสภาวะสิ่งแวดล้อมของความเข้มแสงไม่เหมาะสม ทั้งนี้ขึ้นอยู่กับจุดประสงค์ของการนำไปใช้งาน

วิธีการค้นหาเส้นทางแบบคอนทัวร์เทรสซึ่งจะได้เส้นทางที่อ้อมสิ่งกีดขวางโดยยึดให้อยู่ในแนวของเส้นทางกระจัดจากจุดนั้นไปยังเป้าหมายมากที่สุด เหมาะสภาพแวดล้อมที่มีสิ่งกีดขวางไม่มาก เนื่องจากถ้ามีสิ่งกีดขวางหลายชิ้นจะทำการสร้างทางเดินที่อ้อมวัตถุทำให้เสียเวลาแทนที่จะเลือกไปทางอื่นที่มีสิ่งกีดขวางน้อยกว่าและวิธีนี้จะไม่คำนึงถึงระยะทางที่จะเดินไปด้วย ซึ่งทำให้ไม่ได้เส้นทางที่ดีที่สุด ส่วนในแบบเอสตาร์นั้นจะมีการคำนึงถึงค่าคอสท์ที่ด้วย เมื่อนำทั้งสองวิธีมารวมกัน จะทำให้ได้เส้นทางหลบหลีกสิ่งกีดขวางและเป็นเส้นทางที่ค่าคอสท์น้อยด้วย

ข้อจำกัดทางด้านฮาร์ดแวร์ทำให้ไม่สามารถควบคุมทิศทางได้อย่างแม่นยำ เมื่อรถเดินออกจากเส้นทางทำให้เกิดความผิดพลาด รวมทั้งปัญหาของการสั่งควบคุมยังไม่เป็นเรียลไทม์ เนื่องจากการส่งข้อมูลแบบไร้สายทำให้ข้อมูลเกิดความล่าช้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

กรณีศึกษา (Review Literature)

5.1 Condensation Algorithm

การติดตามการเคลื่อนไหวของวัตถุบนภาพพื้นหลังที่ยุ่งเหยิงนั้นเป็นเรื่องยากมาก การใช้อัลกอริธึมความน่าจะเป็นเป็นสิ่งสำคัญ ที่ผ่านมาอัลกอริธึมคาลมานฟิลเตอร์มีขีดจำกัดในเรื่องการกระจายความน่าจะเป็น จึงพัฒนาโมเดลใหม่ที่เรียกว่า คอนเดนท์เซชั่น อัลกอริธึม(Condensation Algorithm: Conditional Density Propagation) ซึ่งใช้โมเดลของการเคลื่อนไหวแบบนอนลีเนียร์ ที่ซับซ้อนกว่าในคาลมานฟิลเตอร์ เราได้ใช้การผสมผสานของแทรคเกอร์แบบไม่ต่อเนื่องในโครงงานนี้ ซึ่งเป็นการสับไปมาระหว่างหลายๆ โมเดลการเคลื่อนไหวที่กลับไปกลับมาตามเมตริกซ์การถ่ายทอดที่ไม่ต่อเนื่อง รวมทั้งการใช้เทคนิคทางสถิติของการสุ่มตัวอย่างที่สำคัญ มันเป็นไปได้ที่ใช้ในคอนเดนท์เซชั่น เป็นแทรคเกอร์ที่ทำงานแบบเรียลไทม์ จากตัวอย่างเป็นการติดตามการเคลื่อนไหวของมือบนภาพพื้นหลังที่ยุ่งเหยิง และการติดตามการเคลื่อนไหวของใบไม้บนภาพพื้นหลังที่มีลักษณะคล้ายๆกัน และเป็นภาพของหัวเด็กที่กำลังกระโดดเล่นบนพื้นหลังที่ยุ่งเหยิงอีกเช่นกัน

รูปที่ 5.1 ตัวอย่างการติดตามการเคลื่อนไหวของมือบนพื้นหลังที่ยุ่งเหยิง

รูปที่ 5.2 ตัวอย่างการติดตามการเคลื่อนไหวของใบไม้บนพื้นหลังที่มีลักษณะคล้ายกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.3 ตัวอย่างการติดตามการเคลื่อนไหวของหัวเด็กบนพื้นหลังที่ยุ่งเหยิง

5.1.1 Mixed-state Condensation tracker with automatic model-switching

Michael Isard and Andrew Blake Proc 6th Int.Conf. Computer Vision, 107-112 (1998)

<u>บทคัดย่อ</u>

การแสดงรูปแบบการเคลื่อนไหวด้วยคอมพิวเตอร์วิชั่นกำลังได้รับความสนใจเป็นอย่างมาก งานวิจัยนี้เป็นการพัฒนามีความสำคัญของทฤษฎีการสุ่มตัวอย่าง ซึ่งเป็นการสับเปลี่ยนอย่างอัตโนมัติระหว่างรูปแบบการเคลื่อนไหวหลากหลายรูปแบบการเคลื่อนไหวที่เป็นส่วนของการติดตาม เบย์เซียน มิกซ์เสตท เฟรมเวอร์ค (Bayesian Mixed-state framework) ได้นำมาอธิบายในซึ่งทฤษฎีนี้เป็นส่วนสำคัญในการปรับปรุงประสิทธิภาพการติดตามบนสิ่งแวดล้อมที่ยุ่งเหยิง ส่วนที่เกี่ยวข้องกับการจดจำท่าทางถูกนำมาใช้ ซึ่งสามารถติดตามการเคลื่อนไหวอย่างเป็นธรรมชาติได้

5.1.2 Visual Motion Analysis by Probabilistic Propagation of Conditional Density

ผลงานของ *Michael Isard D.Phil. Thesis, Oxford University, 1998*

<u>บทคัดย่อ</u>

วิทยานิพนธ์นี้สร้างโครงงานสำหรับการติดตามการเคลื่อนไหวของเส้นโค้งบนภาพที่ยุ่งเหยิง โดยใช้อัลกอริธึม เบเซียน แรนดอม แซมพลิง (Bayesian random-sampling algorithm) มีแนวความคิดมาจากสถิติ ทฤษฎีการควบคุม และคอมพิวเตอร์วิชั่น ปัญหาก็คือกาติดตามขอบและคุณลักษณะของวัตถุด้านหน้า เหมือนเส้นโค้งที่เคลื่อนในภาพยุ่งเหยิง และความต้องการที่จะทำให้ใกล้กับความเร็วของเฟรมวิดีโอ อัลกอริธึมคอนเดนท์เซชั่นได้ถูกคิดค้นขึ้นมาจากนักวิจัยหลายๆคนใช้หลักของสถิติ และซิกแนลโพรเซสซิ่ง (Signal Processing) วิทยานิพนธ์นี้แสดงถึงการกรองแบบคอนเดนท์เซชั่นด้วยผลงานของอัลกอริธึมพื้นฐาน และแสดงการประมาณภาพการเคลื่อนไหวโดยการสาธิตการติดตามประสิทธิภาพสูงในสิ่งแวดล้อมที่ยุ่งเหยิง ทั้งที่เป็นรูปแบบอัลกอริธึมอย่างง่ายและใช้รูปแบบการเคลื่อนไหวแบบนอนลีเนียร์ซึ่งเป็นการรวมคุณสมบัติของดิสครีต ฮิดเด็น มาร์คอฟ โมเดล (Discrete Hidden Marcov Models) ด้วยคอนทินิวอัส ออโต้ รีเกรสสีฟ โพรเสส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมชั่น โมเดล (Continuous Auto-Regressive Process motion model) ที่ใช้ในคาลมานฟิลเตอร์ การผสมผสานระหว่างดิสครีท คอนทินิวอัส โมเดลทำให้ผลลัพธ์ของการกรองแบบคอนเดนท์เซชั่นมีความแม่นยำมากขึ้น

5.1.3 Condensation: Unifying low-level and high-level tracking in a stochastic framework

ผลงานของ *Michael Isard and Andrew Blake* Proc 5th European Conf. Computer Vision, Vol. 1 893-908

<u>บทคัดย่อ</u>

งานวิจัยการติดตามการเคลื่อนไหวแบ่งเป็นสองรูปแบบคือ แบบระดับต่ำซึ่งเร็วแต่ให้ข้อมูลง่ายๆ แต่แบบระดับสูงซึ่งใช้ติดตามในรูปแบบที่ซับซ้อนในหลายระนาบ แต่อาจจะความเร็วด้อยกว่า การทำแบบเรียลไทม์ด้วยระบบชั้นดีในที่ยุ่งเหยิงจะทำงานแย่มาก วิทยานิพนธ์นี้แสดงเทคนิคใหมที่รวมระบบชั้นแย่กับชั้นดีเข้าด้วยกัน ใช้เทคนิคทางสถิติในการสุ่มตัวอย่างด้วยคอนเดนท์เซชั่นอัลกอริธึม เรียกว่าไอคอนเดนท์เซชั่น (I condensation)

5.1.4 A Smoothing Filter for Condensation

ผลงานของ *Michael Isard and Andrew Blake* Proc 5th European Conf. Computer Vision, Vol. 1 767-781, 1998

<u>บทคัดย่อ</u>

คอนเดนท์เซชั่นที่กล่าวมาเป็นส่วนหนึ่งของอัลกอริธึมการกรอง ซึ่งแสดงสถานการณ์ติดตามวัตถุโดยใช้การกระจายของความน่าจะเป็น ความยุ่งเหยิงอาจทำให้การกระจายตัวแยกเป็นหลายๆพีคซึ่งแต่ละอันแสดงถึงแต่ละวัตถุ เมื่อการวัดกลายเป็นความกำกวมอีกครั้ง ไม่มีพีคใหนแสดงถึงตำแหน่งวัตถุที่แท้จริง การปรับความเรียบ (Smoothing) ในหัวข้อนี้เป็นเทคนิคทางสถิติของการกำหนดเงื่อนไขของสถณะการกระจายตัวการวัดทั้งในอดีตและอนาคต หลังจากการปรับพีคความยุ่งเหยิงถูกลดลง อันนี้จะใช้อัลกอริธึมสองอย่างในการปรับผลลัพธ์ของคอนเดนท์เซชั่นเทคนิคนี้ประยุกต์มาจากงานของ คีตากาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.5 Condensation – conditional density propagation for visual tracking

ผลงานของ *Michael Isard and Andrew Blake* Int. j. computer Vision, 29, 1, 5-28, (1998)

<u>บทคัดย่อ</u>

การติดตามการเคลื่อนไหวในความหนาแน่นที่ยุ่งเหยิงเป็นสิ่งที่ท้าทายมาก คาลมานฟิลเตอร์นั้นยังไม่เพียงพอเพราะมันใช้หลักของเกาส์เซียนที่ไม่สามารถแสดงผลการวิเคราะห์หลายๆ อย่างในเวลาเดียวกันได้ คอนเดนท์เซชั่นใช้ "Factor Sampling" ซึ่งก่อนหน้านี้ใช้กับภาพนิ่ง ที่ซึ่งการกระจายความน่าจะเป็นแสดงด้วยกลุ่มที่สร้างขึ้นอย่างกระจัดกระจาย คอนเดนท์เซชั่นใช้การเรียนรู้รูปแบบที่เปลี่ยนแปลงตลอดเวลา รวมทั้งการสังเกตภาพ เพื่อสามารถจัดการกระจายของกลุ่ม ผลลัพธ์ที่ได้เป็นการติดตามการเคลื่อนไหวได้ดี โดยไม่จำเป็นต้องใช้ทฤษฎีที่ซับซ้อน อัลกอริธึมนี้สามารถทำงานได้ใกล้เรียลไทม์มาก

5.2 Search Algorithm

5.2.1 A* Search Demonstration

ผลงานของ *James Macgill - Center for Computational Geography.*

A* Search เป็นอัลกอริธึมในการค้นหาชนิดหนึ่งซึ่งสามารถใช้แก้ปัญหาบางอย่างได้โดยการจำลองสถานการณ์เป็นเหตุการณ์เริ่มต้น (Initial State) โดยแต่ละแอ็คชั่นก็จะทำให้สเตทเปลี่ยนไป และถ้าสเตทเปลี่ยนเป็นสเตทที่เราต้องการ นั่นคือเราสามารถแก้ปัญหานั้นได้แล้ว

ในโปรแกรมสาธิตนี้จะอนุญาตให้เราสามารถสร้างสิ่งแวดล้อมเริ่มต้นและกำหนดจุดเริ่มต้นและเป้าหมาย แล้วทำการรันโปรแกรมซึ่งใช้อัลกอริธึมในการค้นหา และแสดงผลเส้นทาง (Path) ผลลัพธ์ที่ได้ออกมาพร้อมทั้งแสดงค่า cost ของเส้นทางนั้นด้วย ซึ่งอัลกอริธึ่มในการค้นหานั้นมี 3 แบบด้วยกันคือ Classic A*, Dijkstra's algorithm และ Fudge

- Classic A* เป็นการอิมพลิเมนท์โดยใช้ทฤษฎีของ A*
- Dijkstra's algorithm เป็นวิธีที่คล้ายกับ A*
- Fudge ใกล้เคียงกับ A* แต่ต่างกันตรงที่ใช้ค่า heuristic ไม่เหมือนกันโดยจะคำนึงถึงระยะทางตรงมากกว่า

5.2.2 Heuristic for A*

เมื่อใช้อัลกอริธึมอย่างเอสตาร์ (A*) จะต้องเลือกฮิวริสติค (Heuristic) ที่เหมาะสม ในที่นี้จะพูดถึงฮิวริสติคสำหรับกริดแมพ (Grid map) ถ้าจะใช้กับรูปหลายด้านอื่นอาจต้องใช้ฮิวริสติคอื่น ฮิวริสติคจะต้องเป็นฟังก์ชันต่อเนื่องเพื่อจะกระโดดจากจากช่องหนึ่งของแผนที่ไปยังช่องข้างเคียง ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอสตาร์จะมีประสิทธิภาพเมื่อใช้ระยะทางที่การสูญเสีย (cost) น้อยที่สุดจากตำแหน่งนั้นไปถึงเป้าหมาย แต่บางครั้งเป้าหมายก็ต่างกันขึ้นกับการนำไปใช้งานด้วย ชนิดของฮิวริสติคมีดังนี้

5.2.2.1 แมนฮัตตัน ดิสแตนซ์ (Manhattan Distance)

เป็นฮิวริสติคมาตรฐาน ซึ่งดูจากฟังก์ชันการสูญเสีย(cost) แล้วเลือกค่าการสูญเสียที่น้อยที่สุดเพื่อจะเดินไปเส้นทางนั้น

h(A) = (abs(A.x-goal.x) + abs(A.y-goal.y))

5.2.2.2 ไดแอกกอนัล (Diagonal Distance)

ถ้าแผนที่เป็นแบบหลายทิศทาง ดังนั้นต้องใช้แมนฮัตตันที่เป็น 4 ทิศทางตะวันออก และ 4 ทิศทางเหนือ หรือแทนด้วย 4 ทิศทางตะวันออกเฉียงเหนือ ฟังก์ชันที่ใช้กับไดแอกกอนัลคือ

h(A) = max(abs(A.x-goal.x), abs(A.y-goal.y))

5.2.2.3 สเตรทไลน์ดิสแตนซ์

ถ้าสามารถเคลื่อนไปได้ทุกทิศทาง ดังนั้นสามารถใช้ระยะกระจัดโดยตรง

h(A) = sqrt((A.x-goal.x)^2 + (A.y-goal.y)^2)

5.2.2.4 การคาดเดาคุณภาพของเส้นทาง (Guessing path quality)

แทนที่จะดูค่าการสูญเสีย (cost) ของระยะทางอย่างเดียวก็พิจารณาสิ่งกีดขวางด้วย โดยเพิ่มค่าการสูญเสียของเส้นทางเมื่อเจอสิ่งกีดขวาง เริ่มจากการสร้างแผนที่ตารางหลายเหลี่ยมโดยประมาณ เมื่อลากเส้นทางจากตำแหน่งอ้างอิงไปยังเป้าหมาย สแกนหาสิ่งกีดขวาง และทำการลากเส้นล้อมรอบ แล้วความยาวของเส้นจะเป็นค่าฮิวริสติค

5.2.2.5 เบรคกิ้ง ไท(Breaking ties)

ปัญหาหนึ่งที่จะพบคือเมื่อหลายๆเส้นทางคือค่าฮิวริสติคเท่ากัน ดังนั้นต้องเลือกเพียงหนึ่ง เราสามารถเพิ่มไทค์เบรกเกอร์ (Tied-breaker) เข้าไปกับฮิวริสติค โค้ดนี้ได้มาจากการครอสโพรดัคท์ระหว่างเวคเตอร์จุดเริ่มต้นถึงเป้าหมาย และเวกเตอร์ตำแหน่ง ณ ขณะนั้นถึงเป้าหมาย ถ้าเวกเตอร์ทั้งสองไม่อยู่ในแนวเดียวกัน ค่าผลลัพธ์ครอสโพรดัคท์จะสูง เมื่อคูณกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าฮิวริสติคผลลัพธ์จะสูงขึ้น ซึ่งแบบนี้จะปรับปรุงประสิทธิภาพให้ดีขึ้นเพราะผ่านสิ่งกีดขวางน้อยลง

```
double dx1 = currentX - goalX;
double dy1 = currentY - goalY;
double dx2 = startX - goalX;
double dy2 = startY - goalY;
cross = dx1*dy2 - dx2*dy1;
if( cross<0 ) cross = -cross;
... add cross*0.001 to the heuristic ...
```

5.2.2.6 การค้นหาพื้นที่

เมื่อต้องการค้นหาจุดที่ใกล้เป้าหมาย สามารถสร้างฮิวริสติค h'(x) ที่เป็นค่าที่น้อยที่สุดของ h1(x), h2(x), h3(x) โดยค่าเหล่านี้เป็นฮิวริสติคของจุดใกล้ๆ ทางที่เร็วกว่าคือให้เอสตาร์ค้นหาจุดศูนย์กลางของเป้าหมายและอาจจะหยุดเมื่อถึงจุดข้างเคียงก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

“HSV Color Model” G. Scott Owen, owen@siggraph.org

http://www.siggraph.org/education/materials/HyperGraph/color/colorhs.htm

“Web Graphics and presentation: Web color”

http://www.projectcool.com/developer/gzone/color.htm

"The RGB Color Calculator. Web page color resources."

http://www.webadv.com/color_convert.htm

“Amit’s Thoughts on Path finding” Amit J. Patel

http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

“Data Mining”: Concepts and Techniques by Jiawei Han and Micheline Kamber

“Artificial Intelligence: A Modern Approach” by Stuart J. Russell, Peter Norvig

“Tricks of the Windows Game Programming Gurus” Andre LaMothe's nya är HÄR Part III Hardcore Game Programming, Chapter 12 Making Silicon Think with Artificial Intelligence page.713

“การเขียนโปรแกรมติดต่อ Serial Port”

http://thaiio.com/Hardware/SerialPort.htm

http://thaiio.com/Programing/VBSerial.htm

http://www.ontrak.net/visual.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้