

ภาคผนวก ก.

ส่วนเกี่ยวข้องกับฮาร์ดแวร์

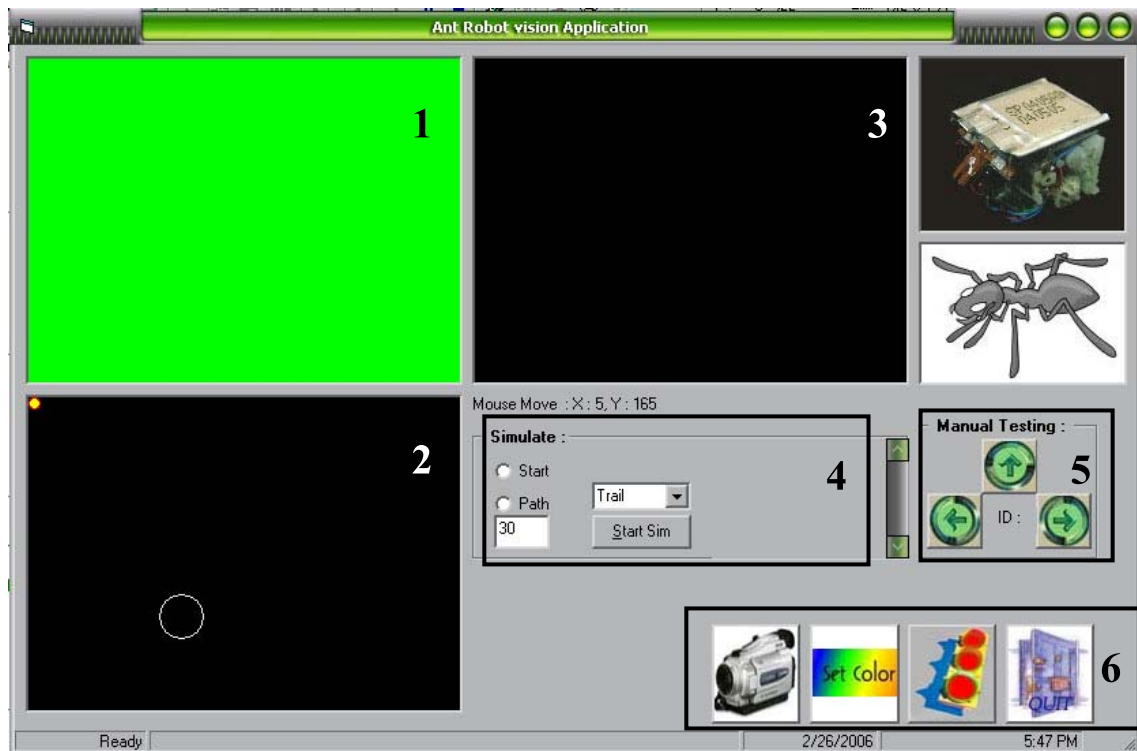
ก.1 ส่วนติดต่อผู้ใช้งานโปรแกรม

ส่วนติดต่อผู้ใช้ของโปรแกรมมีองค์ประกอบดังนี้

1. ส่วนแสดงผลสัญญาณภาพจากกล้องวิดีโอ
2. ส่วนกำหนดเส้นทางการเดินของหุ่นยนต์ตอนเริ่ม
3. ส่วนแสดงผลจากการจำลองเส้นทางการเดินของหุ่นยนต์
4. ส่วนกำหนดรูปแบบการทำงานของหุ่นยนต์
5. ส่วนควบคุมและทดสอบการทำงานของหุ่นยนต์โดยผู้ใช้
6. ส่วนของการตั้งค่าต่าง ๆ

สำหรับฟังก์ชันการใช้งานของโปรแกรม หลัก ๆ มีดังนี้

1. การประมวลผลภาพเพื่อทำการหาตำแหน่งของวัตถุเป้าหมาย และหุ่นยนต์
2. การตั้งค่าสำหรับการประมวลผลภาพ
3. การทดสอบและควบคุมจากผู้ใช้งาน
4. กำหนดรูปแบบการทำงานแบบอัตโนมัติ โดยระบบกฎพื้นฐาน
5. กำหนดเส้นทางการเดินตอนเริ่มต้น และแสดงผลจำลองการเดินของหุ่นยนต์



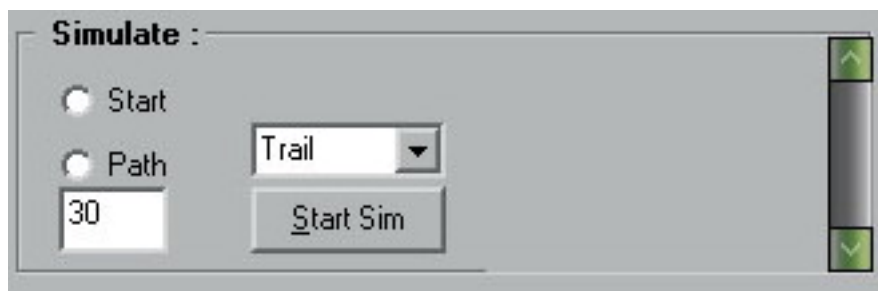
รูปที่ ก.1 หน้าจอทั้งหมดของส่วนติดต่อผู้ใช้

รูปที่ ก.1 แสดงส่วนประกอบทั้งหมดของโปรแกรม ส่วนปุ่มบังคับทิศทางสำหรับทดสอบ
เพื่อทำการตรวจสอบความพร้อมของหุ่นยนต์ด้วยตนเอง แสดงดังรูปที่ ก.2



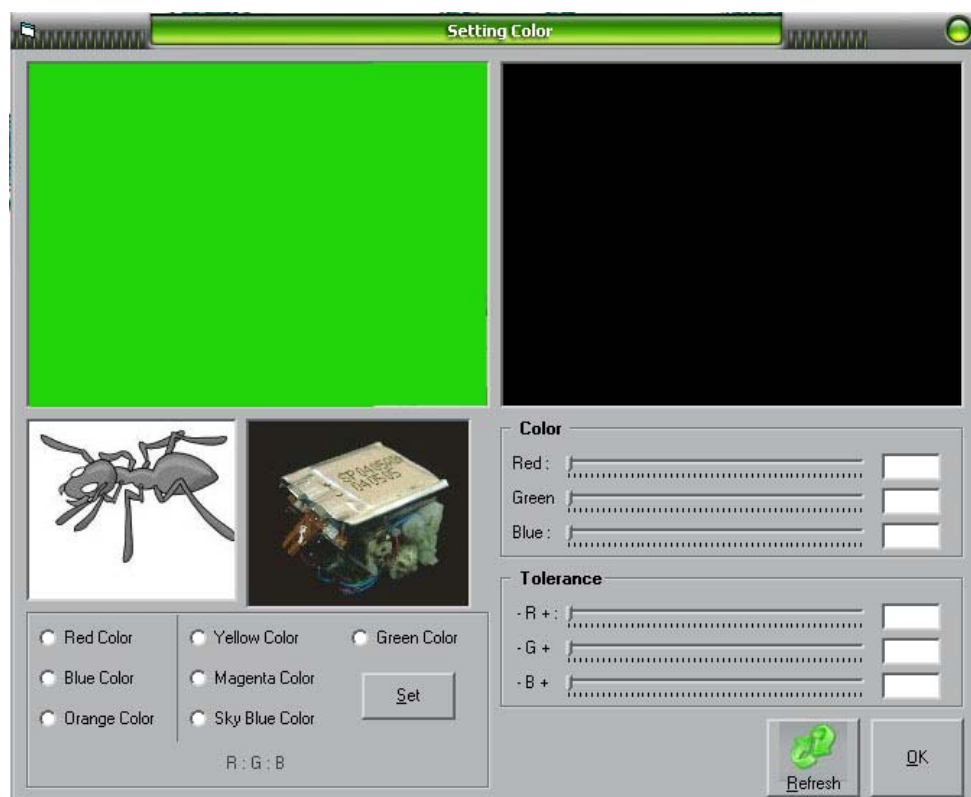
รูปที่ ก.2 ปุ่มสำหรับทดสอบการทำงานการเคลื่อนที่ของหุ่นยนต์ด้วยตนเอง

รูปที่ ก.3 เป็นหน้าจอสำหรับติดต่อของผู้ใช้สำหรับจำลองการกำหนดรูปแบบพฤติกรรม
การเคลื่อนที่ของหุ่นยนต์ รูปแบบของการเคลื่อนที่นี้สามารถกำหนดเส้นทางระหว่างเป้าหมายที่
ต้องการกับจุดเริ่มต้นได้ และดูเส้นทางการเคลื่อนที่ที่ปรากฏที่จอ จะเห็นเส้นทางการเคลื่อนที่
จนกระทั่งท้ายสุดได้เป็นระยะขจัดระหว่างเป้าหมายและจุดเริ่มต้น



รูปที่ ก.3 ปุ่มสำหรับจำลองการกำหนดรูปแบบการเคลื่อนที่ของหุ่นยนต์

รูปที่ ก.4 เป็นหน้าจอการปรับแต่งค่าสีเพื่อการประมวลผลภาพ ใช้เพื่อแก้ไขความผิดเพี้ยนทางสีในเบื้องต้น ที่อาจเกิดขึ้นเมื่อรับสัญญาณภาพจากกล้องเข้ามาแล้ว



รูปที่ ก.4 หน้าจอการปรับแต่งค่าสีเพื่อการประมวลผลภาพ

ก.2 การออกแบบหุ่นยนต์

เนื่องจากวิทยานิพนธ์นี้ต้องการหุ่นยนต์ขนาดเล็กที่สามารถปฏิบัติงานได้สะดวก และมีราคาต่ำ เมื่อพิจารณาแล้วขนาดเล็กที่สุดเท่าที่สามารถหาอุปกรณ์ได้ไม่ยากนักและตรงตามงบประมาณที่คาดการณ์ไว้ ทำให้ได้หุ่นยนต์ขนาดกว้างสองเซนติเมตร ยาวสองเซนติเมตร และสูงสองเซนติเมตรโดยประมาณ สื่อสารโดยใช้ชุดวงจรสื่อสารอินฟราเรด มีแผงวงจรควบคุมหุ่นยนต์และวงจรขับเคลื่อนมอเตอร์ขนาดเล็ก

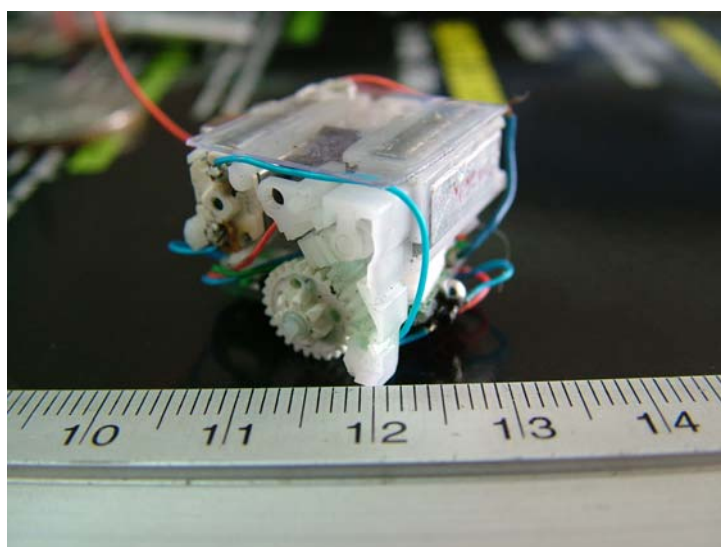
โดยปกติมอเตอร์ชนิดเดียวกันควรมีคุณสมบัติและประสิทธิภาพที่เหมือนกัน แต่ในความเป็นจริง การหามอเตอร์ที่มีประสิทธิภาพเท่ากันทุกด้านเป็นเรื่องยาก แม้เป็นมอเตอร์ใหม่จากโรงงานก็ตาม หุ่นยนต์ที่ใช้ในวิทยานิพนธ์นี้ต้องการมอเตอร์ที่มีความเหมาะสม คือ มีขนาดเล็กและมีแรงบิดพอเพียงสามารถขับเคลื่อนตัวหุ่นได้ดี แต่มอเตอร์ขนาดเล็กส่วนใหญ่มีความเร็วรอบสูงแต่ไม่มีแรงบิดพอ จึงแก้ไขโดยการประกอบชุดเฟืองทดเพื่อเพิ่มแรงบิดให้มอเตอร์ เพื่อให้มีแรงขับเคลื่อนตัวหุ่นยนต์ ชุดขับเคลื่อนหุ่นยนต์เลือกใช้มอเตอร์ที่อยู่ในชุดขับเคลื่อนถาดของเครื่องเล่นซีดี เป็นมอเตอร์ที่ผ่านการใช้งานแล้วเพื่อประหยัดต้นทุน โดยพยายามเลือกมอเตอร์ที่มีความเร็ว

รอบใกล้เคียงกันมากที่สุด ผลการทดสอบแสดงในภาคผนวก ข แต่เนื่องจากมอเตอร์ที่เลือกมาส่วนมากเมื่อจ่ายแรงดันเข้าไปในมอเตอร์ในระดับที่เท่ากัน มอเตอร์หมุนด้วยความเร็วที่ไม่เท่ากัน จึงอาจทำให้หุ่นยนต์เคลื่อนที่ไปไม่ตรงตำแหน่งตามที่ต้องการ จึงทำการแก้ไขข้อบกพร่องส่วนนี้ โดยแก้ไขร่วมกันทั้ง 2 ทางคือทางวงจรและทางโปรแกรมการจัดการ รูปของมอเตอร์ที่ใช้แสดงในรูปที่ ก.5



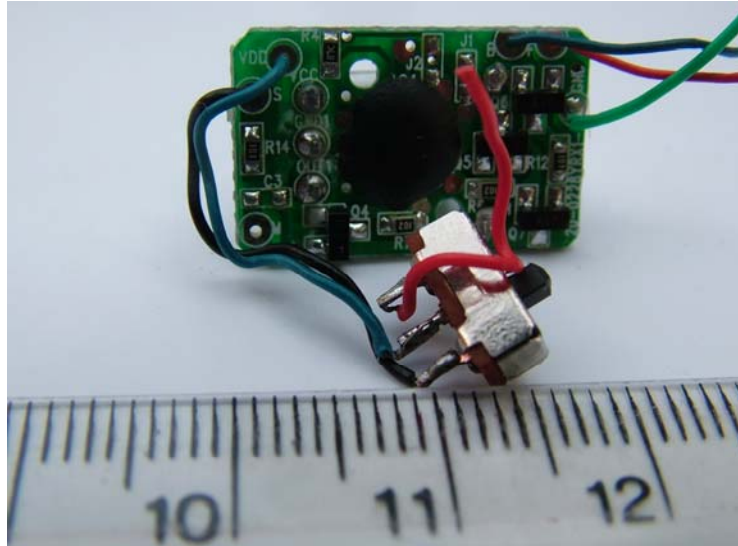
รูปที่ ก.5 มอเตอร์กระแสตรงที่เลือกใช้งาน

เมื่อเลือกมอเตอร์ที่เหมาะสมทั้งขนาด คุณภาพ และราคาได้แล้ว จึงนำมาจัดรูปแบบและประกอบขึ้นเป็นตัวหุ่นเพื่อให้ได้ขนาดของตัวหุ่นตามที่ต้องการ ดังรูปที่ ก.6



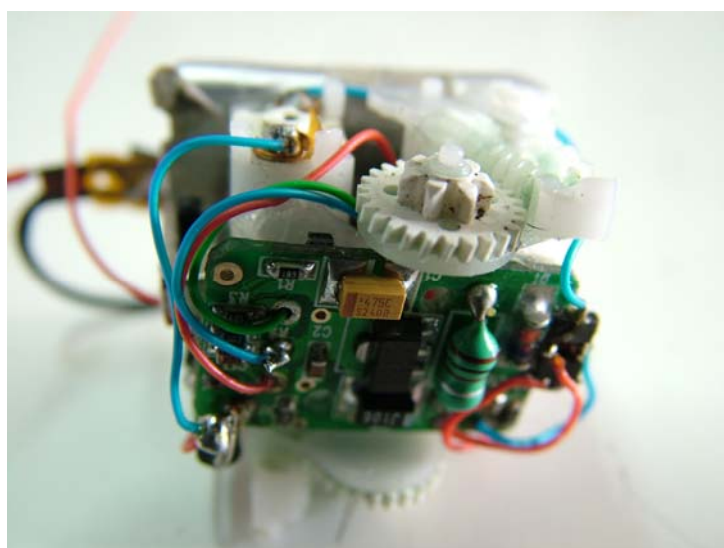
รูปที่ ก.6 หุ่นยนต์หลังจากประกอบมอเตอร์แล้ว

จากชุดวงจรในรูปที่ ก.7 ประกอบด้วยชุดวงจรควบคุมเป็นแบบวงจรรวม มีวงจรขับมอเตอร์จักรเป็นลักษณะวงจรบริดจ์ ใช้ทรานซิสเตอร์เป็นตัวขับ 4 ตัวรวมอยู่ด้วย



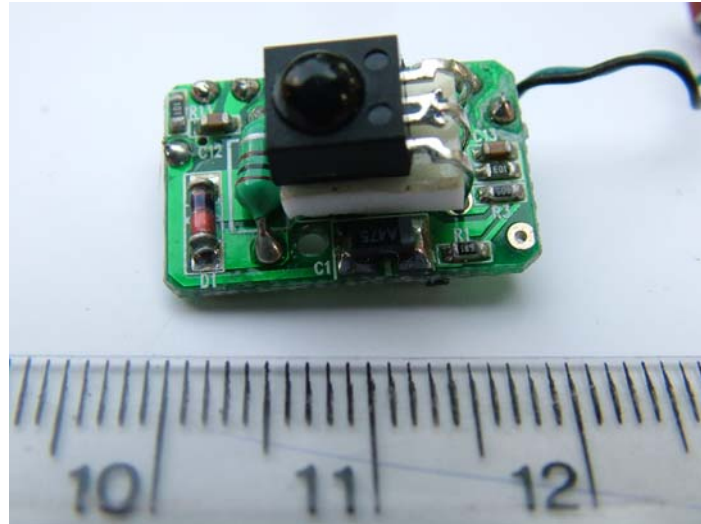
รูปที่ ก.7 แผงวงจรควบคุมและรับส่งสัญญาณ พร้อมวงจรขับมอเตอร์

แผงวงจรควบคุมถูกวางไว้ข้างใต้มอเตอร์เนื่องจากเมื่อทำการจัดตำแหน่งมอเตอร์แล้วมีพื้นที่ว่างพอดี การเคลื่อนที่ใช้มอเตอร์ขับเฟืองตัวหนอนคู่ชุดเฟืองตามให้เป็นล้อขับเคลื่อนตัวหุ่น ทั้งมอเตอร์ข้างซ้ายและข้างขวา ดังรูปที่ ก.8



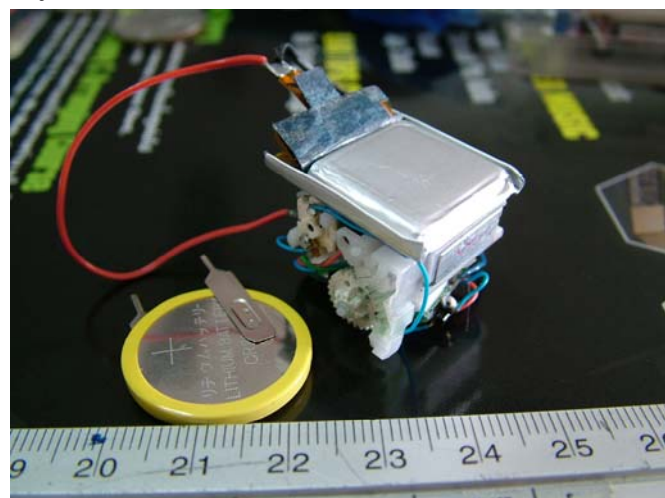
รูปที่ ก.8 แผงวงจรควบคุมหุ่นยนต์หลังจากติดตั้งแล้ว

ชุดวงจรรับสัญญาณควบคุมอินฟราเรดดังรูปที่ ก.9 ใช้สำหรับรับสัญญาณจากชุดส่งสัญญาณขนาดเล็กที่ต่ออยู่กับเครื่องคอมพิวเตอร์ มีขนาดกว้าง 2 เซนติเมตร และยาว 2 เซนติเมตร สามารถติดต่อสื่อสารได้ระยะทางไม่เกินหนึ่งเมตร โดยประมาณ ซึ่งมีความเหมาะสมเนื่องจากมีขนาดเล็กและน้ำหนักเบา



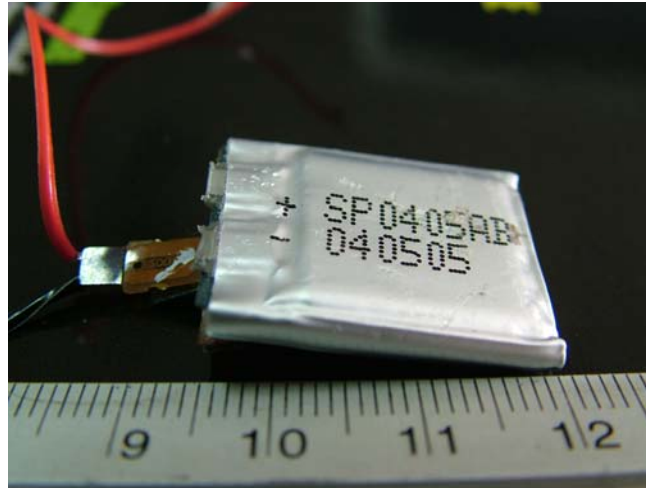
รูปที่ ก.9 ชุดเครื่องรับอินฟราเรด

จากการเลือกแบตเตอรี่โดยการเปรียบเทียบคุณสมบัติต่าง ๆ พบว่าแบตเตอรี่ที่มีขนาดและพลังงานเพียงพอเหมาะสม มีอยู่ 2 ชนิด คือ ชนิดลิเทียม (Li) ซึ่งมีข้อจำกัดคือ ไม่สามารถนำมาประจุใหม่ได้ แต่มีราคาถูก และชนิด ลิเทียมโพลิเมอร์ (Li-Po) ที่สามารถนำกลับมาประจุใหม่ได้ แต่ต้องใช้เครื่องประจุเฉพาะ รูปที่ ก.10 แสดงลักษณะแบตเตอรี่ทั้ง 2 ประเภท



รูปที่ ก.10 แบตเตอรี่ชนิดลิเทียมและลิเทียมโพลิเมอร์

เนื่องจากต้องการให้หุ่นยนต์มีขนาดเล็กและน้ำหนักเบา การเลือกใช้แหล่งพลังงาน จำเป็นต้องเลือกที่มีขนาดเล็ก จึงเลือกใช้แบตเตอรี่ชนิดลิเทียมโพลิเมอร์ ซึ่งมีขนาดกว้างและยาว ประมาณสองเซนติเมตร หนาเพียงสามมิลลิเมตรเท่านั้น และยังสามารถนำกลับมาประจุใหม่ได้อีก ด้วย ดังแสดงในรูปที่ ก.11



รูปที่ ก.11 ขนาดของแบตเตอรี่ชนิดลิเทียมโพลิเมอร์

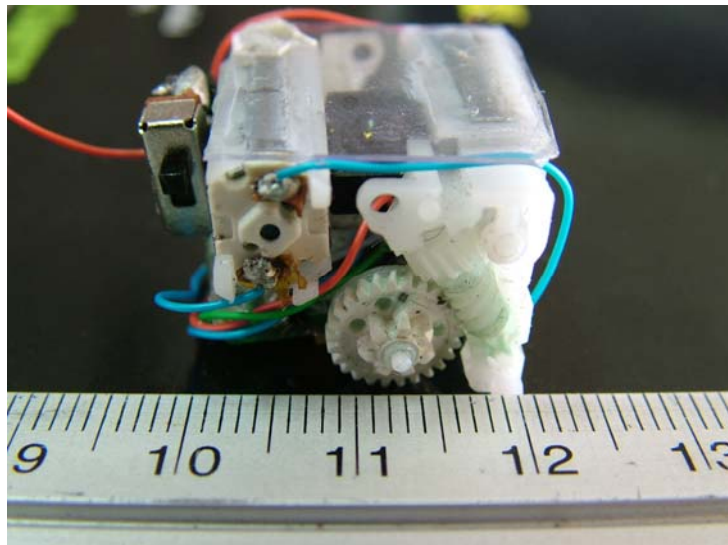
แต่เนื่องจากแบตเตอรี่แบบนี้มีกรรมวิธีการประจุแบตเตอรี่เฉพาะ จึงจำเป็นต้องใช้เครื่อง ประจุเฉพาะของแบตเตอรี่ ดังแสดงในรูปที่ ก.12



รูปที่ ก.12 เครื่องประจุแบตเตอรี่ชนิดลิเทียมโพลิเมอร์

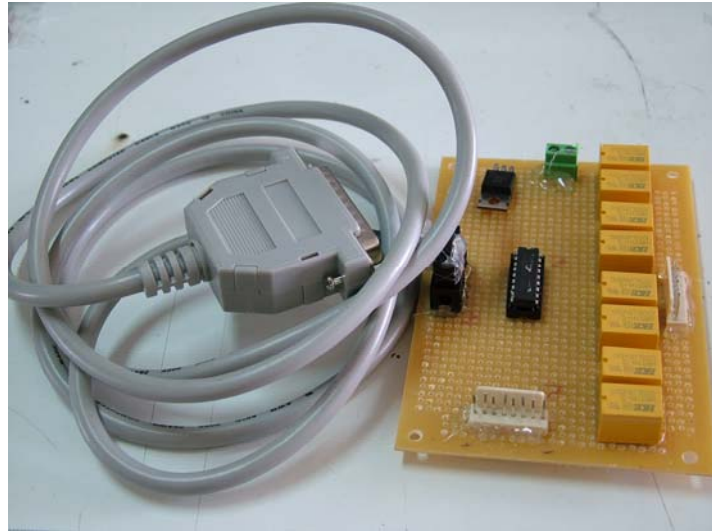
เนื่องจากมอเตอร์ที่ใช้มีแรงบิดไม่มากนัก จึงต้องออกแบบให้หุ่นยนต์มีน้ำหนักไม่มาก การประกอบชิ้นส่วนเข้าด้วยกันจึงไม่ใช่แผ่นโลหะ เพราะทำให้หุ่นยนต์มีขนาดใหญ่และน้ำหนักมาก ด้วยเหตุนี้จึงเลือกใช้แผ่นพลาสติกซึ่งมีขนาดบางและน้ำหนักเบา ใช้กาวร้อนเป็นตัวประสานให้ชิ้นส่วนต่าง ๆ ติดกัน

รูปที่ ก.13 เป็นรูปหุ่นยนต์ที่เสร็จสมบูรณ์แล้วด้านข้างจะเห็นมอเตอร์ขับเคลื่อนทั้งสองตัววางประกบกันอยู่และชุดรับสัญญาณอินฟราเรดติดตั้งอยู่ตำแหน่งตรงกลางระหว่างมอเตอร์ทั้งสองตัว



รูปที่ ก.13 โครงสร้างหุ่นยนต์ด้านข้าง

การควบคุมหุ่นยนต์แต่ละตัวจากคอมพิวเตอร์หลัก โดยชุดส่งสัญญาณดังรูปที่ ก.14 รับสัญญาณควบคุมหุ่นยนต์แต่ละตัวจากคอมพิวเตอร์ผ่านพอร์ตแบบขนาน สามารถควบคุมหุ่นยนต์ได้พร้อมกันมากที่สุด 4 ตัว

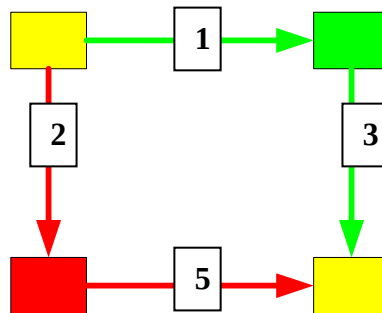


รูปที่ ก.14 ชุดเชื่อมต่อระหว่างคอมพิวเตอร์และชุดเครื่องส่งสัญญาณ

ก.3 การค้นหาเส้นทาง (Path finding)

ก.3.1 การค้นหาด้วยวิธี Dijkstra

การค้นหาด้วยวิธี Dijkstra [8] เป็นวิธีการคำนวณหาเส้นทางซึ่งใช้รูปแบบเดียวกับวิธีหาเส้นทางแบบ OSPF (Open Shortest Path First) ที่ใช้ในเครือข่ายคอมพิวเตอร์ จากรูปที่ ก.15 สมมุติว่าจะทำการหาเส้นทางจาก R1 ไปยัง R4 ซึ่งเห็นได้ว่าสามารถเดินได้ 2 ทาง คือ ผ่าน R2 กับ R3 แต่หลังจากคำนวณแล้ว พบว่าเส้นทางผ่าน R2 เป็นระยะทางที่สั้นกว่าในการเดินทาง ดังจะได้กล่าวต่อไป



รูปที่ ก.15 เส้นทางการเดินทางจากจุด R1 ไป R2

ขั้นตอนการค้นหาเส้นทางแบบ Dijkstra

ก.3.1.1 สร้างเมตริกเส้นทางของเส้นทางดังรูปที่ ก.15

$$\begin{bmatrix} \infty & 1 & 2 & \infty \\ 1 & \infty & \infty & 3 \\ 2 & \infty & \infty & 5 \\ \infty & 3 & 5 & \infty \end{bmatrix}$$

R1

รูปที่ ก.15 เมตริกของเส้นทาง

ก.3.1.2 กำหนดค่าเริ่มต้นต่าง ๆ เพื่อใช้ในการคำนวณหาเส้นทาง โดยสมมุติให้เริ่มต้นจาก R1 ก่อน ตามตารางที่ ก.1

R3

ตารางที่ ก.1 การกำหนดค่าเริ่มต้น

Node	R1	R2	R3	R4
I()	0	∞	∞	∞
Permanent (Y/N)	Yes	No	No	No

จากตารางที่ ก.1 เห็นได้ว่าที่จุด R1 ค่าของเส้นทางเป็น 0 ค่า Permanent ซึ่งแสดงการยืนยันการเดินทาง เป็น yes เมื่อเริ่มนำค่าจากเมตริกในแถวที่ 1 ตำแหน่งที่ R2 ซึ่งมีค่าเท่ากับ 1 มาบวกกับค่าเริ่มต้นที่เป็นค่า 0 แล้ว ได้ค่าในตำแหน่งที่ R2 เป็น 1 ค่า ทำเช่นนี้ไปเรื่อย ๆ จนครบ ถึง R4 ดังเกิดได้ว่าค่าที่น้อยที่สุดอยู่ที่ R2 จึงใส่ ค่า permanent เป็น yes

ก.3.1.3 ทำเหมือนข้อ ก.3.1.2 แต่เปลี่ยนตำแหน่ง เป็น R2, R3, R4 ไปเรื่อย ๆ จนครบ จะได้ค่าตามตารางในรูปที่ ก.2 ดังนี้

ตารางที่ ก.2 ผลลัพธ์เมื่อทำการวิเคราะห์แล้ว

Node	R1	R2	R3	R4
I()	0	1	2	4
Permanent (Y/N)	Yes	Yes	Yes	Yes

จากตารางที่ ก.2 สรุปได้ว่า ระยะทางที่สั้นที่สุดคือจาก R1 ไปทาง R2 ซึ่งมีค่าเป็น 1 จาก R1 ไป R3 มีค่าเป็น 2 และจาก R1 ไป R4 มีค่าเป็น 4

ก.3.2 การค้นหาเส้นทางด้วยวิธี Breadth-First

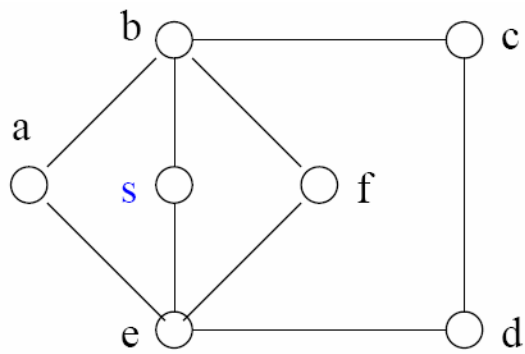
การค้นหาเส้นทางด้วยวิธี Breadth-First [9] เป็นวิธีการค้นหาเส้นทางอีกวิธีหนึ่ง โดยทำการหาจากฟังก์ชันภาพ $G = (V;E)$ ให้ระยะเป็น $d[v]$ จาก S ไป v และฟังก์ชัน $pred[v]$ โดยทำการเก็บค่าไว้ใน 2 อาร์เรย์นี้ และมี อาร์เรย์เพิ่มมาอีก 1 อาร์เรย์ เพื่อเก็บข้อมูลที่เป็นสี กำหนด ให้

White (W, “undiscovered”)

Gray (G, “discovered” but not “processed”)

Black (B, “discovered” and “processed”)

จากนั้นเริ่มทำการหาเส้นทางด้วยกระบวนการต่อไปนี้



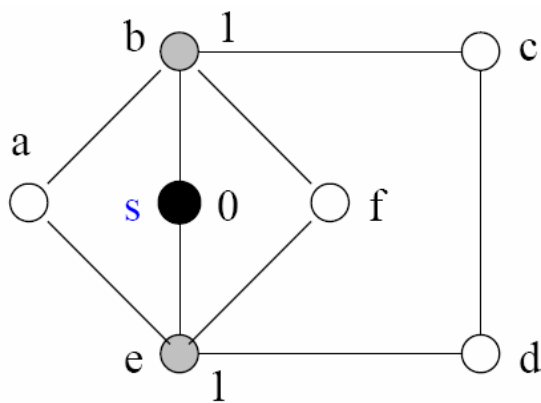
รูปที่ ก.16 เส้นทางที่ต้องการหา

ก.3.2.1 สร้างตารางโดยกำหนดค่าเริ่มต้นดังตารางที่ ก.3

ตารางที่ ก.3 ค่าเริ่มต้นของอาร์เรย์ต่าง ๆ

Vertex (u)	s	a	B	c	d	e	F
color[u]	W	W	W	W	W	W	W
d[u]	∞	∞	∞	∞	∞	∞	∞
pred[u]	Null	Null	Null	Null	Null	Null	Null

ก.3.2.2 ทำการเปลี่ยนแปลงค่าโดยใช้ค่าจากตำแหน่งใกล้เคียง เริ่มจากตำแหน่ง s ตามรูปที่ ก.17



รูปที่ ก.17 การเปลี่ยนแปลงค่าจากตำแหน่งใกล้เคียง

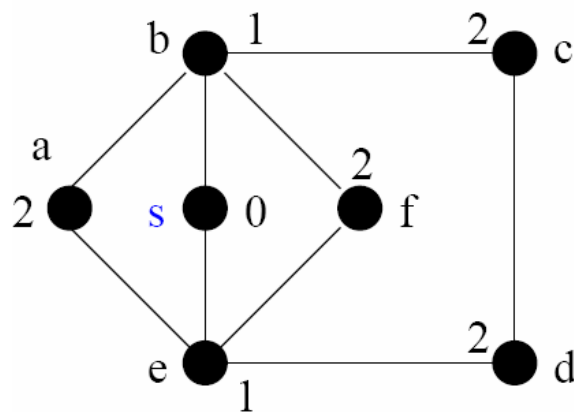
ได้ค่าสี b, e เปลี่ยนเป็นสีเทาหมายความว่า ตำแหน่งได้ถูกพบแล้วแต่ยังไม่ได้ทำการเริ่มกระบวนการในตำแหน่งต่อไป ทำการเก็บค่าระยะทางลงไปในอาร์เรย์ระยะทางตามตารางที่ ก.4 ดังต่อไปนี้

ตารางที่ ก.4 อาร์เรย์ค่าระยะทาง

Vertex (u)	s	a	B	c	d	e	F
color[u]	B	W	G	W	W	G	W
d[u]	0	∞	1	∞	∞	1	∞
pred[u]	Null	Null	S	Null	Null	s	Null

จากตารางที่ 2.4 ได้ค่าของตาราง เปลี่ยนจากตารางที่ ก.3 โดยอาร์เรย์ที่เก็บค่า pred[u] ที่ช่องของ b, e เป็นค่า s แสดงว่าขั้นตอนก่อนหน้านี้ ได้มาจากโหนด s

ก.3.2.3 ทำเหมือนในข้อที่ ก.3.2.2 ต่อเนื่องไปจนสุดท้ายแล้วได้ออกมาดังรูปที่ ก.18 และตารางที่ ก.5



รูปที่ ก.18 ผลที่ได้จากการหาเส้นทาง

ตารางที่ ก.5 ค่าการเก็บระยะทางขั้นสุดท้าย

Vertex (u)	s	a	B	c	d	e	F
color[u]	B	B	B	B	B	B	B
d[u]	0	2	1	2	2	1	2
pred[u]	Null	b	S	b	e	s	B

จากวิธีการที่กล่าวมาข้างต้น สามารถหาได้เพียงระยะทางที่สั้นที่สุดเท่านั้น จึงจำเป็นต้องหา
 เวกเตอร์ $G = (V;E)$ เพื่อใช้ทำออกมาเป็นภาพของเส้นทางต่อไป