สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก

SMALL ROBOT MANAGEMENT AND CONTROL PROGRAM

ทรงเดช โชติกวนิชย์

ธิติพร พรพิรุณรักษ์

ภควัตร เทพวิทักษ์กิจ

เลขหมู่.....
เลขทะเบียน 43037
วัน, เดือน, ปี 26 มิ.ย. 2545

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์ และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMALL ROBOT MANAGEMENT AND CONTROL PROGRAM

SONGDEJ CHOTIGAVANICH
THITIPORN PORNPIRUNRAK
BHAKAWAT TEPWITUKSAGID

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก

SMALL ROBOT CONTROL AND MANAGEMENT PROGRAM

ชื่อนักศึกษา นายทรงเดช โชติกวนิชย์ 41056030

นายธิติพร พรพิรุณรักษ์ 41056043

นายภควัตร เทพวิทักษ์กิจ 41056076

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล

อาจารย์ วิสันต์ ตั้งวงษ์เจริญ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2544

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผู้ช่วยศาสตราจารย์สุนทร สุชาติเวชภูมิ	
กรรมการ	ผู้ช่วยศาสตราจารย์พัชรินทร์ เหมโชติ	
กรรมการ และอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
กรรมการ และอาจารย์ที่ปรึกษา	อาจารย์ วิสันต์ ตั้งวงษ์เจริญ	

(ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์ และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์ และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก

ชื่อนักศึกษา นายทรงเดช โชติกวนิชย์ 41056030

นายธิติพร พรพิรุณรักษ์ 41056043

นายภควัตร เทพวิทักษ์กิจ 41056076

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2544

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล

อาจารย์ วิสันต์ ตั้งวงษ์เจริญ

บทคัดย่อ

ในปัจจุบันนี้อุตสาหกรรมรูปแบบใหม่ ๆ เกิดขึ้นทุกวัน กิจกรรมบางอย่างมนุษย์ไม่สามารถปฏิบัติด้วยตนเองได้ เนื่องจากมีความเสี่ยงในการทำงานสูง ซึ่งอาจก่อให้เกิดอันตรายต่อชีวิตและทรัพย์สินของผู้ปฏิบัติงาน จึงมีความจำเป็นที่บุคคลเหล่านั้นจะต้องควบคุมให้เครื่องจักรเป็นผู้ปฏิบัติงานแทน แต่การนำเครื่องมืออิเลคทรอนิคส์เหล่านี้มาใช้งานย่อมมีความยุ่งยากในการเรียนรู้เป็นอย่างมาก ซึ่งแต่เดิมนั้นการทำงานกับอุปกรณ์เหล่านี้จำเป็นต้องใช้บุคลากรที่มีความรู้ความชำนาญสูง ทำให้ค่าใช้จ่ายที่ต้องใช้ในการจ้างงานเพิ่มขึ้นเป็นเงาตามตัว และบุคลากรเหล่านี้ยังมีจำนวนน้อยอยู่ ทางคณะผู้จัดทำจึงได้ทำการศึกษาและสร้างโปรแกรมเพื่อใช้ในการควบคุมการทำงานของอุปกรณ์อิเลคทรอนิคส์ในรูปแบบของหุ่นยนต์นาดเล็ก ที่มีความง่ายในการเรียนรู้และใช้งาน ผู้ใช้สามารถสั่งการและควบคุมการทำงานของหุ่นยนต์ได้โดยการเลือกคำสั่งที่สื่อโดยใช้รูปภาพ จากหน้าจอการทำงานหลักเพื่อสั่งให้อุปกรณ์เคลื่อนที่ไปตามทิศทางที่ระบุเอาไว้ ทำให้บุคลากรจากทุกระดับสามารถศึกษาและนำไปใช้งานโดยไม่เกิดความยุ่งยาก ไม่จำเป็นต้องใช้บุคลากรที่มีระดับความรู้สูงมากเป็นผู้ปฏิบัติงานทุกอย่างอีกต่อไป และเพื่อศึกษาแนวทางในการสร้างโปรแกรมต้นแบบในการอำนวยความสะดวกแก่ผู้ทำงานร่วมกับอุปกรณ์อิเลคทรอนิคส์รูปแบบต่าง ๆ พัฒนาไปใช้ในอุตสาหกรรมต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Small Robot Control and Management Program	
Students	Mr. Songdej Chotigavanich	41056030
	Mr. Thitiporn Pornpirunrak	41056043
	Mr. Bhakawat Tepwituksagid	41056076
Degree	Mathematics and Computer Science, Faculty of Science	
Department	Computer Science	
Academic Year	2001	
Special Project Advisor	Assistant Professor Teerawat Prakobphon	
	Lecture Wisan Tangwongcharoen	

ABSTRACT

New form of manufacturing is born everyday. Human being can't do some activities by themselves because of high risk during working that may damage lives and goods of workers. It is necessary that those worker have to use machines instead of doing by themselves. There are many difficulties in dealing with these electronic instruments, in the past, experts are needed to control the instruments. But there are few experts in this field and the cost of hiring them was increasing so that we did a research and created a program for controlling the electronic instruments in the form of small robot, which is easy to use. Worker can control a robot by selecting an instruction from a user interface, in order to move a robot in any specified directions. Every workers can use it without difficulty so that they will use it as a model leading to create other new model programs, as a result, it is more convenient for people who use any form of electronic instruments for developing in the use in industry.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องโปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กสามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล และอาจารย์วิสันต์ ตั้งวงษ์เจริญ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำ และเป็นที่ปรึกษาในการแก้ปัญหาต่าง ๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จไปด้วยดี รวมทั้งพี่ ๆ จากบริษัท Innovation ที่ให้คำแนะนำทางด้านอุปกรณ์หุ่นยนต์ รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญภาพ.....VIII

บทที่ 1 บทนำ.....1
1.1 ความสำคัญ และที่มาของปัญหา.....1
1.2 ความมุ่งหมาย และวัตถุประสงค์ของการศึกษา.....1
1.3 ขอบเขตของการศึกษา.....2
1.4 ขั้นตอนการดำเนินงาน.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2 ทฤษฎี และวรรณกรรมที่เกี่ยวข้อง.....4
2.1 ไมโครคอนโทรเลอร์ (Microcontroller).....4
2.1.1 การทำงานของไมโครคอนโทรเลอร์.....5
2.2 เบสิกแสตมป์ 2SX (BASIC STAMP 2SX).....6
2.2.1 ตระกูลของเบสิกแสตมป์.....7
2.2.2 คุณสมบัติของเบสิกแสตมป์.....8
2.3 การใช้งานคำสั่งของ BASIC STAMP 2.....9
2.3.1 ลักษณะพื้นฐานของภาษาพีเบสิก.....9
2.3.2 รูปแบบในการเขียนโปรแกรมพีเบสิก.....10
2.3.3 รายละเอียดชุดคำสั่งเบสิกแสตมป์ 2SX.....12
2.3.3.1 กลุ่มคำสั่งจัดการข้อมูล.....12
2.3.3.2 กลุ่มคำสั่งควบคุมลำดับการทำงาน.....12
2.3.3.3 กลุ่มคำสั่งจัดการสัญญาณดิจิตอล.....13
2.3.3.4 กลุ่มคำสั่งจัดการสัญญาณอะนาลอก.....13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.3.5 กลุ่มคำสั่งจัดการข้อมูลอนุกรม.....13
2.3.3.6 กลุ่มคำสั่งตารางข้อมูล.....14
2.3.3.7 กลุ่มคำสั่งประมวลผลทางคณิตศาสตร์.....14
2.3.3.8 กลุ่มคำสั่งประมวลผลทางลอจิก.....15
2.3.4 รายละเอียดการใช้งานแต่ละคำสั่ง.....15
2.4 การแปลงการใช้งานโปรแกรมเป็นภาษา PBASIC เพื่อการใช้งานจริง.....31
2.4.1 คุณสมบัติของเซอร์โวมอเตอร์.....32
2.4.2 การขับเซอร์โวมอเตอร์ด้วยภาษา PBASIC.....32
2.4.3 รูปแบบคำสั่งของการเขียนโปรแกรมควบคุมเซอร์โวมอเตอร์.....32
2.4.3.1 กรณีการเดินหน้า และถอยหลัง.....32
2.4.3.2 กรณีการเลี้ยวซ้าย และขวา.....34

บทที่ 3 วิธีการดำเนินการวิจัย.....36
3.1 ขั้นตอนการศึกษาโปรแกรมภาษา PBASIC.....36
3.2 ขั้นตอนการศึกษาลักษณะโครงสร้างของหุ่นยนต์ ROBO-KIT.....36
3.2.1 คุณลักษณะของหุ่นยนต์ ROBO-KIT.....37
3.2.2 ชิ้นส่วนประกอบของหุ่นยนต์ ROBO-KIT.....37
3.3 ขั้นตอนการศึกษาภาคการทำงานของหุ่นยนตร์ ROBO-KIT.....37
3.3.1 ส่วนการทำงานของไมโครคอนโทรลเลอร์.....38
3.3.2 ส่วนการทำงานของเซอร์โวมอเตอร์.....38
3.3.3 ส่วนการทำงานของไฟเลี้ยว.....39
3.3.4 ส่วนการทำงานของตัวตรวจจับแสง.....39
3.4 ขั้นตอนการศึกษาลักษณะการเคลื่อนที่ของหุ่นยนต์.....40
3.4.1 กรณีเดินหน้า และถอยหลัง.....40
3.4.2 กรณีเลี้ยวซ้าย และขวา.....40
3.4.3 กรณีหมุนซ้าย และขวา.....41
3.5 ขั้นตอนการศึกษาถึงความสัมพันธ์ระหว่างการเคลื่อนที่ของหุ่นยนต์
กับการสร้างโปรแกรมควบคุมด้วยภาษา PBASIC.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.6 ขั้นตอนการศึกษา และทำการออกแบบส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้ด้วย Visual Basic.....42
3.7 ขั้นตอนการกำหนดโครงสร้างของไฟล์ที่ใช้สำหรับใช้งานร่วมกับการทำงานของโปรแกรม.....42
3.7.1 ไฟล์ Source Code ที่ได้จากการ Compile.....43
3.7.2 ไฟล์ที่ใช้สำหรับเก็บข้อมูลเส้นทางการเคลื่อนที่ของหุ่นยนต์.....43
3.7.3 Log File.....45
3.7.4 ไฟล์กำหนดค่าของโปรแกรม (Setting File).....46
3.8 ขั้นตอนควบคุมให้โปรแกรม Visual Basic ทำการการสร้างไฟล์ที่ทำหน้าที่เป็น Source Code ของภาษา PBASIC.....47

บทที่ 4 การพัฒนาระบบงาน.....56
4.1 ความต้องการทางด้านฮาร์ดแวร์.....56
4.1.1 ความต้องการทางด้านอุปกรณ์คอมพิวเตอร์.....56
4.1.2 ความต้องการทางด้านอุปกรณ์หุ่นยนต์.....56
4.2 ความต้องการทางด้านซอฟท์แวร์.....56
4.2.1 ความต้องการทางด้านระบบปฏิบัติการ.....57
4.2.2 ความต้องการทางด้านอื่น ๆ.....57
4.3 ขั้นตอนการพัฒนาโปรแกรม.....57
4.3.1 การจัดการทางด้านโปรแกรมที่ใช้ในการพัฒนา.....57
4.3.2 การจัดการทางด้านอุปกรณ์หุ่นยนต์.....59
4.4 แผนภาพการทำงานของโปรแกรม.....61
4.4.1 หน้าจอหลัก.....62
4.4.2 หน้าจอบันทึกข้อมูล.....64
4.4.3 หน้าจอเรียกใช้งานข้อมูล.....65
4.4.4 หน้าจอ Compile โปรแกรม.....65
4.4.5 หน้าจอเครื่องมือควบคุม.....66
4.4.6 หน้าจอกำหนดค่าเส้นทาง.....66
4.4.7 หน้าจอกำหนดค่าโปรแกรม.....67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.5 ผลการทดลองใช้งานโปรแกรม.....67
4.5.1 การเดินทางเป็นเส้นตรง.....67
4.5.2 การเดินไปในทิศทางตรงข้ามกับทิศทางเดิม (ย้อนกลับ).....68
4.5.3 การหมุนไปทางซ้าย.....69
4.5.4 การหมุนไปทางขวา.....70
4.5.5 การหยุดตามเวลาที่กำหนด.....71
4.5.6 ทดสอบการชน.....72
4.5.7 การเคลื่อนที่ตามเส้นทางที่กำหนด.....73
4.6 สิ่งที่ได้จากการพัฒนาโปรแกรม.....74

บทที่ 5 สรุปผล และข้อเสนอแนะ.....75
5.1 สรุปผล.....75
5.2 ข้อเสนอแนะ.....77

ภาคผนวก ก ตัวอย่างหน้าจอในระบบ.....79

ภาคผนวก ข ทดสอบการใช้งานโปรแกรม.....88

บรรณานุกรม.....97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางแสดงคำสั่งต้องห้ามที่ไม่สามารถนำมาใช้เป็นชื่อลาเบลหรือตัวแปรได้.....10
2.2 ตารางแสดงกลุ่มคำสั่งจัดการข้อมูล.....12
2.3 ตารางแสดงกลุ่มคำสั่งควบคุมลำดับการทำงาน.....12
2.4 ตารางแสดงกลุ่มคำสั่งจัดการสัญญาณดิจิตอล.....13
2.5 ตารางแสดงกลุ่มคำสั่งจัดการสัญญาณอะนาลอก.....13
2.6 ตารางแสดงกลุ่มคำสั่งจัดการข้อมูลอนุกรม.....14
2.7 ตารางแสดงกลุ่มคำสั่งตารางข้อมูล.....14
2.8 ตารางแสดงกลุ่มคำสั่งประมวลผลทางคณิตศาสตร์.....14
2.9 ตารางแสดงกลุ่มคำสั่งประมวลผลทางลอจิก.....15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 รูปแสดงลักษณะโครงสร้างภายในตัวไมโครคอนโทรเลอร์.....5
3.1 รูปแสดงการทำงานของไมโครคอนโทรเลอร์ 38
3.2 รูปแสดงการทำงานของเซอร์โวมอเตอร์ 39
3.3 รูปแสดงการทำงานของไฟเลี้ยว (LED).....39
3.4 รูปแสดงการทำงานของตัวตรวจจับการชน.....40
3.5 รูปแสดงการเคลื่อนที่ของหุ่นยนต์.....41
3.6 Flowchart แสดงขั้นตอนการทำงานของโปรแกรมโดยรวม.....48
3.7 Flowchart แสดงขั้นตอนการรับข้อมูลเข้าโดยผู้ใช้.....49
3.8 Flowchart แสดงขั้นตอนการแสดงเส้นทางการเดินของหุ่นยนต์โดยโปรแกรม.....50
3.9 Flowchart แสดงขั้นตอนการปรับเปลี่ยนคุณสมบัติของโปรแกรม.....51
3.10 Flowchart แสดงขั้นตอนการปรับเปลี่ยนคุณสมบัติของการเคลื่อนที่ของหุ่นยนต์.....52
3.11 Flowchart แสดงขั้นตอนการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของหุ่นยนต์.....53
3.12 Flowchart แสดงขั้นตอนการ Compile เส้นทางการเคลื่อนที่ของหุ่นยนต์เป็น Source Code.....54
3.13 Flowchart แสดงขั้นตอนการบันทึก Source Code ลงไมโครคอนโทรเลอร์.....55
4.1 รูปภาพแสดงวิธีการติดตั้ง GiiSoft6.ocx.....58
4.2 รูปแสดงหน้าจอ Components ของโปรแกรม Microsoft Visual Basic.....59
4.3 รูปแสดงส่วนประกอบหลักต่าง ๆ ของโปรแกรม.....52
4.4 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า 1 เมตร.....68
4.5 รูปแสดงการวัดมุมของการเลี้ยว.....70
4.6 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์หยุดการเคลื่อนที่.....71
4.7 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์เคลื่อนที่ตามที่กำหนด.....73
A.1 รูปภาพแสดงหน้าจอหลักของโปรแกรม.....79
A.2 รูปแสดงหน้าจอเครื่องมือควบคุม.....80
A.3 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปข้างหน้า.....81
A.4 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปข้างหลัง.....81
A.5 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปด้านซ้าย.....81
A.6 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปด้านขวา.....82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.7 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หมุนขวา.....82
A.8 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หมุนขวา.....82
A.9 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หยุดการเคลื่อนที่ชั่วคราว.....82
A.10 รูปแสดงหน้าจอบันทึกข้อมูล.....83
A.11 รูปแสดงหน้าจอบันทึกข้อมูล.....84
A.12 รูปแสดงหน้าจอ Compile โปรแกรม.....85
A.13 รูปแสดงหน้าจอกำหนดค่าตัวอักษร.....86
A.14 รูปแสดงหน้าจอกำหนดค่าค่าสีพื้นหลัง.....86
B.1 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากการเรียกใช้งาน.....88
B.2 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากการซ่อนเส้น grid.....89
B.3 รูปแสดงหน้าจอหลักของโปรแกรม หลังจาก click เพื่อเลือกเส้นทาง.....90
B.4 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากบันทึกเส้นทางการเคลื่อนที่ของหุ่นยนต์.....91
B.5 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอบันทึกข้อมูล.....92
B.6 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอเรียกใช้งานข้อมูล.....93
B.7 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งาน Compile ข้อมูล.....94
B.8 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอปรับคุณสมบัติของโปรแกรม.....95
B.9 รูปแสดงหน้าจอของโปรแกรม เมื่อต้องการออกจากโปรแกรม.....96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญ และที่มาของปัญหา

ปัจจุบันอุปกรณ์อีเลคโทรนิคส์นั้นเข้ามามีบทบาทต่อชีวิตของมนุษย์ทุกคนเป็นอย่างมาไม่ว่าจะเป็นอุปกรณ์ไฟฟ้าต่าง ๆ ภายในบ้านเรือน หรือในโรงงานอุตสาหกรรม โดยแต่ก่อนนั้นการที่จะควบคุมอุปกรณ์เหล่านั้นจะใช้การควบคุมโดยตรงจากผู้ใช้ และอุปกรณ์บางชนิดก็มีความเสี่ยงต่ออันตรายสูง โดยเฉพาะอย่างยิ่งอุปกรณ์ที่ใช้ในโรงงานต่าง ๆ จำเป็นจะต้องใช้บุคลากรที่มีความสามารถ และมีความชำนาญในการใช้งานอุปกรณ์นั้น ๆ ซึ่งถ้าเกิดความผิดพลาดแล้ว อาจทำให้เกิดความเสียหายต่อทรัพย์สิน และชีวิตของผู้ใช้ได้ ในปัจจุบันจึงได้มีการพัฒนาเครื่องจักรกลต่าง ๆ เพื่อเข้ามาทำงานแทนมนุษย์ เพื่อที่จะได้ลดการเสี่ยงต่ออันตรายที่จะเกิดขึ้นได้กับมนุษย์ และเพื่อที่จะสามารถควบคุมอุปกรณ์จักรกลเหล่านั้นได้สะดวก และง่ายดายมากขึ้น จึงได้มีการนำเอาคอมพิวเตอร์เข้ามาประยุกต์เพื่อใช้ควบคุมอุปกรณ์ต่าง ๆ เหล่านั้นให้การทำงานตามต้องการ

ดังนั้นคณะผู้จัดทำจึงต้องการที่จะศึกษา และสร้างโปรแกรมที่ใช้ในการควบคุม และจัดการเกี่ยวกับการทำงานของอีเลกโทรนิกส์ โดยนำเอาอุปกรณ์พื้นฐานทางอีเลกโทรนิกส์ต่าง ๆ มาศึกษา และทำการประกอบเป็นชุดหุ่นยนต์ขนาดเล็ก เพื่อที่จะนำมาสร้างโปรแกรมต้นแบบในการควบคุมการทำงานของหุ่นยนต์ต่อไป

ลักษณะการทำงานของโปรแกรมจะเป็นการเขียนคำสั่งเพื่อไปบังคับให้หุ่นยนต์ขนาดเล็กทำงาน โดยผ่านทาง Serial Port เมื่อทำการสร้างโปรแกรมเสร็จแล้ว โดยขั้นตอนในการกำหนดให้หุ่นยนต์ทำงาน สามารถกำหนดลักษณะการทำงานของหุ่นยนต์ได้ผ่านทางโปรแกรม Interface โดยผู้ใช้สามารถกำหนด และเลือกเส้นทางเดินของหุ่นยนต์ได้

1.2 ความมุ่งหมาย และวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาการพัฒนาหุ่นยนต์ขนาดเล็กเพื่อเป็นต้นแบบในการนำมาใช้ประโยชน์ได้จริง
2. เพื่อที่จะศึกษาการทำงานของอุปกรณ์อิเลคทรอนิกส์ และการทำงานของเครื่องจักรกลต่าง ๆ
3. เพื่อที่จะศึกษาการเขียนโปรแกรมระดับต่ำ และนำไปใช้ควบคุมการทำงานของอุปกรณ์อิเลคทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของการศึกษา

1. สร้างโปรแกรมหลักสำหรับควบคุมหุ่นยนต์ผ่านทางหน้าจอมอนิเตอร์ของคอมพิวเตอร์
2. สามารถจัดการและควบคุมหุ่นยนต์ได้ ผ่านทางโปรแกรมหลักที่ทำการสร้างขึ้น
3. สามารถทำให้หุ่นยนต์เคลื่อนไหวได้ โดยใช้มอเตอร์ควบคุมล้อ
4. สามารถทำให้หุ่นยนต์ตรวจจับลักษรักการเปลี่ยนแปลงของพื้นผิวที่กำลังเคลื่อนที่อยู่ โดยตรวจจับจากตัวเซนเซอร์ที่ติดอยู่กับหุ่นยนต์
5. สามารถทำการดาว์นโหลดโปรแกรมที่ได้ทำการสร้างลงไปในตัวหุ่นยนต์ได้จากโปรแกรมหลักที่ทำการสร้างขึ้น
6. ทำการสร้างหน้าจอสำหรับทำการบังคับหุ่นยนต์ โดยออกแบบให้ง่ายต่อการใช้
7. สามารให้ผู้ใช้กำหนดเส้นทางการเคลื่อนที่ของหุ่นยนต์ได้จากหน้าจอที่สร้างขึ้นมา
8. สามารถทำให้หุ่นยนต์ส่งเสียงร้อง เมื่อมีเหตุการณ์ผิดปกติขึ้น เช่นพื้นผิวเปลี่ยนไป

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาลักษณะภายนอกของหุ่นยนต์ รวมถึงลักษณะการทำงานอย่างคร่าว ๆ
2. ศึกษาถึงองค์ประกอบ และชิ้นส่วนของหุ่นยนต์โดยรวม
3. ศึกษาถึงลักษณะของ Controller ต่าง ๆ ภายในตัวหุ่นยนต์ รวมถึงอุปกรณ์ต่อพ่วงทุกชิ้นที่ใช้ในการทำงานของหุ่นยนต์
4. ศึกษาการประกอบหุ่นยนต์ โดยพิจารณาถึงอุปกรณ์แต่ละชิ้นโดยละเอียด
5. ทำการทดลองการเคลื่อนที่ของหุ่นยนต์ โดยนำเอาโปรแกรมที่ติดมากับหุ่นยนต์มาใช้ในการทดลอง
6. ศึกษาตัวอย่างโปรแกรม และคำสั่งที่ใช้ในการควบคุมการทำงานของหุ่นยนต์โดยละเอียด
7. เริ่มทำการเขียนโปรแกรมย่อย เพื่อใช้ในการทดลองการทำงานของหุ่นยนต์
   - ทำให้หุ่นยนต์สามารถเคลื่อนที่ได้ทุกทิศในแนวราบ
   - ทำให้หุ่นยนต์สามารถตรวจจับลักษณะของพื้นผิวที่เปลี่ยนแปลงไปได้
   - ทำให้หุ่นยนต์สามารถตรวจจับหาเส้นสีบนพื้นผิวที่เคลื่อนไหวได้
8. ออกแบบส่วนการทำงานของโปรแกรมหลัก โดยออกแบบให้มีความง่ายต่อการใช้ สำหรับผู้ที่ไม่เข้าใจในภาษาระดับต่ำได้ด้วย
9. สร้างตัวต้นแบบ (Prototype) ของโปรแกรมหลักที่ใช้ควบคุมการทำงานของหุ่นยนต์
10. นำเอาโปรแกรมย่อยต่าง ๆ ที่ได้สร้างขึ้นมาแล้วใส่ในโปรแกรมหลัก และทำการเพิ่มโปรแกรมย่อยบางส่วนลงไป
11. ทำการเพิ่มเติมรายละเอียดต่าง ๆ ของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. นำเอาโปรแกรมหลักแต่ละส่วนนั้นไปทดลองกับหุ่นยนต์จริง ๆ เพื่อหาข้อผิดพลาด
13. แก้ไขข้อผิดพลาดที่เกิดขึ้น และปรับปรุงบางส่วนให้สมบูรณ์ยิ่งขึ้น

1.5 ขั้นตอนการดำเนินงาน

1. ทำให้เข้าใจการทำงานของอุปกรณ์อีเลกโทรนิกส์ต่าง ๆ
2. ได้ศึกษา และพัฒนาการเขียนโปรแกรมด้วยภาษา Assembly
3. สามารถนำเอาหุ่นยนต์ที่ได้สร้างขึ้นมาใช้เป็นตัวต้นแบบในการพัฒนา เพื่อให้เกิดประโยชน์มากขึ้นต่อไปในอนาคตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี และวรรณกรรมที่เกี่ยวข้อง

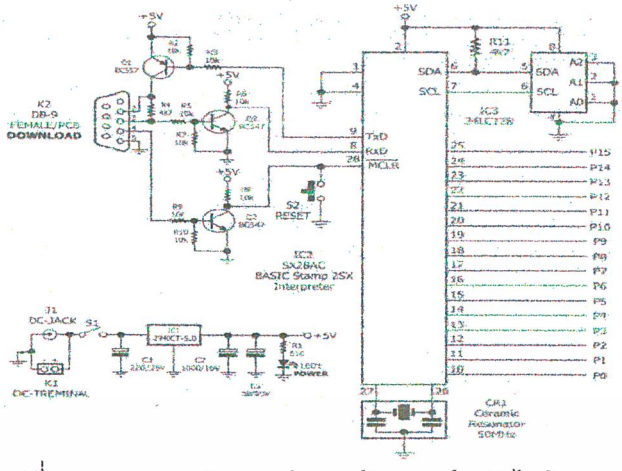
2.1 **ไมโครคอนโทรลเลอร์** (Microcontroller)

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำสองคำ คำหนึ่งคือ **ไมโคร** (micro) หมายถึง *ขนาดเล็ก* และคำว่า **คอนโทรลเลอร์** (Controller) หมายถึง *ตัวควบคุมหรืออุปกรณ์ควบคุม* ดังนั้น**ไมโครคอนโทรลเลอร์**จึงหมายถึง *อุปกรณ์ควบคุมขนาดเล็ก*

ถ้ากล่าวถึงคอมพิวเตอร์จะสามารถมองเห็นภาพรวมได้ชัดเจนว่า คอมพิวเตอร์หนึ่งเครื่องประกอบด้วย คีย์บอร์ด ,เมาส์ ,จอภาพหรือมอนิเตอร์ และส่วนสำคัญหลักคือ **หน่วยประมวลผลกลาง** หรือ ซีพียู (CPU : Central Processing Unit) ซีพียูจะสามารถติดต่อกับอุปกรณ์ต่าง ๆ ได้ต้องอาศัยคำสั่งควบคุม ซึ่งบรรจุอยู่ใน **หน่วยความจำโปรแกรม** (Program Memory) และหน่วยความจำข้อมูล (Data Memory) สำหรับการพักข้อมูลชั่วคราว โดยหน่วยความจำข้อมูลบางที่จะเรียกว่า **รีจีสเตอร์** (Register) โดยปรกติซีพียูจะไม่สามารถติดต่อกับอุปกรณ์ภายนอกได้โดยตรง ต้องอาศัยส่วนเชื่อมต่อที่เรียกว่า **พอร์ต** (Port) ซึ่งมี 2 ลักษณะด้วยกันคือ **พอร์ตอินพุต** (Input Port) ใช้สำหรับข้อมูลหรือสัญญาณเข้าจากภายนอก และ**พอร์ตเอาต์พุต** (Output Port) ใช้สำหรับส่งข้อมูล หรือสัญญาณออกไปภายนอก ถ้าพิจารณาจากเครื่องคอมพิวเตอร์ คีย์บอร์ดและเมาส์ คืออุปกรณ์อินพุตต้องต่อเข้ากับซีพียูผ่านพอร์ตอินพุต ในขณะที่จอภาพเป็นอุปกรณ์เอาต์พุตต้องต่อกับซีพียูผ่านพอร์ตเอาต์พุต เป็นต้น

สำหรับในไมโครคอนโทรลเลอร์เป็นอุปกรณ์พิเศษที่รวมเอาซีพียู ,หน่วยความจำ และพอร์ตเข้าไว้ด้วยกัน โดยบรรจุรวมกันอยู่ภายใต้ตัวถังเดียวกัน จากรูปที่ 2.1 แสดงส่วนประกอบหลักที่สำคัญของไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ที่มีขีดความสามารถสูง ๆ บางเบอร์ จะรวมเอาวงจรสร้างฐานเวลาจริง หรือเรียลไทม์คลอก (Real Time Clock : RTC) ,วงจรตรวจสอบจังหวะการทำงาน หรือวอตช์ด็อก (Watch Dog) ,วงจรตั้งเวลา หรือไทม์เมอร์ (Timer) , วงจรนับ หรือเคาน์เตอร์ (Counter) และวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC : Analog to Digital Converter) เข้าไว้ด้วย เมื่อผู้ใช้งานเข้าใจถึงการทำงานของไมโครคอลโทรลเลอร์เป็นอย่างดีแล้วก็สามารถจะพัฒนาไปใช้ไมโครคอนโทรลเลอร์เบอร์ที่มีขีดความสามารถสูงได้ไม่ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 รูปแสดงลักษณะโครงสร้างภายในตัวไมโครคอนโทรเลอร์

2.1.1 การทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะสามารถทำงานได้เมื่อจ่ายไฟเลี้ยงและต่อวงจรกำเนิดสัญญาณนาฬิกาให้แก่มัน จากนั้นซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านข้อมูลคำสั่ง แล้วทำงานตามคำสั่งที่บรรจุอยู่ในหน่วยความจำโปรแกรม

นั่นหมายความว่า ต้องมีการเขียนข้อมูลลงในหน่วยความจำโปรแกรมก่อน โดยไมโครคอนโทรลเลอร์แต่ละเบอร์จะมีรูปแบบของคำสั่งที่แตกต่างกัน ซึ่งจะต้องอาศัยกระบวนการเขียนโปรแกรม (Programming) ภาษาที่ใช้ในการเขียนโปรแกรมสามารถแบ่งได้ 2 ระดับคือ ภาษาระดับสูง (High Language) และภาษาแอสเซมบลี (Assembly Language) โดยปรกติไมโครคอนโทรลเลอร์ต้องการโปรแกรมที่เขียนด้วยภาษาแอสเซมบลี เนื่องจากสามารถทำงานได้รวดเร็วผ่านกระบวนการแปลงข้อมูลคำสั่งเป็นข้อมูลเลขฐานสิบหกเพื่อทำงานตามคำสั่งเพียง 1 ขั้นตอน คือ แปลงจากภาษาแอสเซมบลีเป็นข้อมูลเลขฐานสิบหก หรือที่เรียกว่าออปโค๊ด (Opcode) แต่ข้อเสียของการเขียนโปรแกรมด้วยภาษาแอสเซมบลีคือ ผู้เขียนต้องทำความเข้าใจในชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้น ๆ อย่างถ่องแท้ และเมื่อเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ก็ต้องทำการเรียนรู้และทำความเข้าใจชุดคำสั่งใหม่ ซึ่งอาจทำให้เสียเวลามาก และการเขียนโปรแกรมด้วยภาษาแอสเซมบลีผู้เขียนต้องมีทักษะในการเขียนโปรแกรมสูงพอสมควร และเข้าใจถึงสถาปัตยกรรมของไมโครคอนโทรลเลอร์เป็นอย่างดี

ในขณะที่การเขียนโปรแกรมด้วยภาษาระดับสูง อาทิเช่น ภาษาซี ภาษาเบสิก ต้องผ่านกระบวนการที่เรียกว่า **คอมไพล์** (Compile) เพื่อแปลงภาษาระดับสูงเหล่านั้นเป็นภาษาเครื่องหรือออปโค๊ดของไมโครคอนโทรลเลอร์เบอร์นั้น ๆ เสียก่อน และโปรแกรมที่ใช้ในการคอมไพล์นั้นเรียกว่า **คอมไพเลอร์** (Compiler) มักจะมีราคาแพง เมื่อใช้เครื่องมือทางซอฟต์แวร์นี้ ทำให้ผู้เขียนโปรแกรมอาจไม่จำเป็นต้องศึกษาสถาปัตยกรรม และชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างลึกซึ้งเท่ากับการเขียนโปรแกรมด้วยภาษาแอสเซมบลี ทั้งนี้เพราะคอมไพเลอร์จะทำหน้าที่ในส่วนนี้แทน ดังนั้นเมื่อผู้ใช้งานเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ก็เพียงแต่จัดหาโปรแกรมคอมไพเลอร์ที่เหมาะสมมาใช้งาน และศึกษาสถาปัตยกรรมของไมโครคอนโทรลเลอร์ใหม่อีกเพียงเล็กน้อยก็สามารถใช้งานได้ แต่ข้เสียของการใช้ไมโครคอนโทรลเลอร์คือ ราคาแพงมาก

ในกรณีของภาษาเบสิก (Basic Language) เป็นภาษาที่ผู้เขียนโปรแกรมให้ความสนใจและนิยมเขียนกันมาก เนื่องจากชุดคำสั่งสามารถทำความเข้าใจได้ง่าย ตรงไปตรงมา โครงสร้างของโปรแกรม (Structure) สามารถทำความเข้าใจได้ง่าย ไม่ซับซ้อนมาก ไมโครคอนโทรลเลอร์แทบทุกเบอร์จึงมักจะมีการพัฒนาด้วยตัวแปลสำหรับภาษาเบสิก แล้วบรรจุลงในหน่วยความจำโปรแกรมของตัวไมโครคอนโทรลเลอร์ในบางส่วนเรียกว่า **เบสิกอินเตอร์พรีเตอร์** (Basic Interpreter) โดยเบสิกอินเตอร์พรีเตอร์นี้ จะทำการแปลงข้อมูลคำสั่งภาษาเบสิกเป็นออปโค้ดให้แก่ซีพียูเพื่อทำงานตามคำสั่ง ด้วยกระบวนการนี้ ทำให้การใช้งานไมโครคอนโทรลเลอร์สามารถกระทำได้ภายใต้โปรแกรมภาษาเบสิก ส่งผลให้การใช้งานง่ายขึ้นอย่างมาก แต่ข้อด้อยของวิธีการนี้คือความเร็วในการทำงานของไมโครคอนโทรลเลอร์จะลดลง

จุดนี้เองคือ จุดอ่อนของการเขียนโปรแกรมด้วยภาษาเบสิก ดังนั้นผู้ผลิตไมโครคอนโทรลเลอร์จึงพยายามหาวิธีการเพื่อให้ไมโครคอนโทรลเลอร์เมื่อทำงานด้วยภาษาเบสิกแล้วมีความเร็วสูงมากขึ้น ทางแก้ไขคือ**เลือกใช้ไมโครคอนโทรลเลอร์ที่มีความเร็วในการทำงานสูงมาก ๆ และเลือกใช้ไอซีหน่วยความจำที่ใช้เก็บหน่วยความจำโปรแกรมมีค่าเวลาในการเข้าถึงข้อมูล** (Access Time) **ต่ำมาก** ในปัจจุบันสามารถพัฒนาไมโครคอนโทรลเลอร์ ให้ใช้ได้กับสัญญาณของนาฬิกาความถี่สูงกว่า 50 MHz ขึ้นไปได้แล้ว ทำให้ความเร็วในการกระทำตำสั่งภาษาเบสิกสามารถกระทำได้สูงมากถึง 10,000 คำสั่งต่อวินาทีขึ้นไปผนวกกับข้อดีของภาษาเบสิกในเรื่องความง่าย และชุดคำสั่งที่ไม่ซับซ้อนทำให้ ณ วันนี้ และอนาคต การเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์ด้วยภาษาเบสิก จะได้รับความนิยมมากขึ้น และยังส่งผลให้ผู้ที่สนใจไมโครคอนโทรลเลอร์สามารถก้าวเข้ามาทำการเรียนรู้ตลอดจนการประยุกต์ใช้งานได้ง่ายขึ้น และสะดวกขึ้น

2.2 เบสิกแสตมป์ 2SX (Basic Stamp 2SX)

เบสิกแสตมป์ (Basic Stamp) คือแผงวงจรไมโครคอนโทรลเลอร์สำเร็จรูปที่บรรจุตัวแปรภาษาเบสิก หรือเบสิกอินเตอร์พรีเตอร์ (Basic Interpreter) รวมไว้ด้วยกัน สามารถใช้การเขียนโปรแกรมด้วยภาษาเบสิกควบคุมการทำงานได้ ในการพัฒนาระบบด้วยเบสิกแสตมป์ไม่จำเป็นต้องใช้เครื่องโปรแกรม (Programmer) หรือเครื่องเลียนแบบ (Emulator) แต่อย่างใด เพียงต่อสายจากคอมพิวเตอร์เข้ากับเบสิกแสตมป์เท่านั้น ก็สามารถพัฒนาโปรแกรมได้แล้ว และเหตุผลที่มีคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าแสตมป์ต่อท้ายก็เพื่อต้องการให้ทราบว่า บอร์ดไมโครคอนโทรลเลอร์ตัวนี้มีขนาดเล็กเท่ากับแสตมป์ หรือตราไปรษณียากรนั่นเอง

ในการพัฒนาโปรแกรมของเบสิกแสตมป์จะใช้ชุดคำสั่งภาษาเบสิกที่เรียกว่า **พีเบสิก** (PBASIC) ซึ่งมีด้วยกัน 36-39 คำสั่ง ทำให้การเรียนรู้ง่ายและรวดเร็ว แต่ละคำสั่งสามารถนำไปใช้ได้ทันที ไม่ต้องเขียนโปรแกรมย่อยมากมาย ในขณะที่ความเร็วในการกระทำคำสั่งของเบสิกแสตมป์ โดยเฉพาะเบสิกแสตมป์ 2SX สูงถึง 10,000 คำสั่งภาษาเบสิกต่อวินาที ทำให้เบสิกแสตมป์สามารถทำงานได้รวดเร็วเพียงพอ

2.2.1 ตระกูลของเบสิกแสตมป์

เบสิกแสตมป์ผลิตโดย Parallax Inc. ในสหรัฐอเมริกา นับถึงปัจจุบัน มีการผลิตเบสิกแสตมป์ออกมาแล้วทั้งหมด 5 รุ่น คือ

- BS1-REV.D หรือ **เบสิกแสตมป์ 1 รีวิชั่นดี** มีจำนวนอินพุต/เอาต์พุต 8 เส้น ขนาดหน่วยความจำ 256 ไบต์ บรรจุคำสั่งภาษา PBASIC-1 ได้ 75 คำสั่ง ความเร็วในการกระทำคำสั่งภาษาเบสิก 2,000 คำสั่งต่อวินาที
- BS1-IC หรือ **เบสิกแสตมป์ 1 รุ่นไอซี** มีรูปร่างเป็นไอซีแบบแถวเดี่ยว มีจำนวนขาอินพุต/เอาต์พุต ,ขนาดหน่วยความจำ และความเร็วเท่ากับรุ่น REV.D
- BS2-IC หรือ **เบสิกแสตมป์ 2 รุ่นไอซี** มีรูปร่างเหมือนและมีขนาดเท่ากับไอซีตัวถัง DIP ขนาด 24 ขา มีขาอินพุต/เอาต์พุต 16 เส้น และขาสำหรับเชื่อต่อกับคอมพิวเตอร์ 3 ขา ขนาดหน่วยความจำเท่ากับ 2 กิโลไบต์ บรรจุโปรแกรมภาษา PBASIC-2 ได้ 500 บรรทัด มีความเร็วในการประมวลผลคำสั่งภาษาเบสิก 4,000 คำสั่งต่อวินาที
- BS2SX-IC หรือ **เบสิกแสตมป์ 2 รุ่นเอสเอ็กซ์** เป็นรุ่นที่ใช้ไมโครคอนโทรลเลอร์ตระกูล SX ซึ่งมีความเร็วในการทำงานสูงกว่า BS2-IC มีจำนวนพอร์ตและขนาดตัวถังเหมือนกับ BS2-IC ขนาดของหน่วยความจำสูงถึง 16 KB สามารถบรรจุโปรแกรมภาษา PBASIC-2 ได้ 4,000 บรรทัด มีความเร็วในการประมวลผลคำสั่งภาษาเบสิก 10,000 คำสั่งต่อวินาที เหมาะสำหรับงานที่ต้องการความเร็ว และขนาดของหน่วยความจำสูง
- BS2E-IC หรือ **เบสิกแสตมป์ 2 รุ่นอี** เป็นรุ่นที่ใช้ไมโครคอนโทรลเลอร์ตระกูล SX เช่นเดียวกับ BS2SX-IC มีความเร็วในการทำงานเท่ากับ BS2-IC มีจำนวนอินพุต/เอาต์พุตและขนาดตัวถังเหมือนกับ BS2-IC ขนาดของหน่วยความจำสูงถึง 16 KB สามารถบรรจุโปรแกรมภาษา PBASIC-2 ได้ 4,000 บรรทัด มีความเร็วในการประมวลผลคำสั่งภาษาเบสิก 4,000 คำสั่งต่อวินาที เหมาะสำหรับผู้ที่ใช้งาน BS2-IC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีความต้องการหน่วยความจำที่เพิ่มมากขึ้น โดยไม่ต้องการเปลี่ยนแปลงโปรแกรมเพื่อไปใช้กับ BS2SX-IC

นอกจากนั้นทางบริษัทผู้ผลิตไมโครคอนโทรลเลอร์ยังได้ผลิตไมโครคอนโทรลเลอร์ที่บรรจุโปรแกรมแปลภาษาเบสิก หรือเบสิกแสตมป์อินเตอร์พรีเตอร์ชิปแยกจำหน่ายด้วยโดยมี 2 รุ่นคือ **เบสิกแสตมป์ 2 อินเตอร์พรีเตอร์ ซึ่งใช้ไมโครคอนโทรลเลอร์** PIC16C57 และ **เบสิกแสตมป์ 2SX อินเตอร์พรีเตอร์ ซึ่งใช้ไมโครคอนโทรลเลอร์** SX28AC ทำให้สามารถนำมาพัฒนาบอร์ดไมโครคอนโทรลเลอร์เบสิกแสตมป์ 2 และ 2SX ได้

2.2.2 **คุณสมบัติของเบสิกแสตมป์ 2SX**

จากรูปที่ 2-1 เป็นบล๊อกไดอะแกรมแสดงคุณสมบัติหลัก ๆ ของเบสิกแสตมป์ 2SX สามารถสรุปได้ดังนี้

- มีจำนวนขาอินพุต/เอาต์พุต 16 ขา (P0 – P15) และขาสื่อสารอนุกรม 2 ขา (S_{IN},S_{OUT})
- มีชุดคำสั่ง PBASIC-2 39 คำสั่ง สามารถประมวลผลทางคณิตศาสตร์ และลอจิกได้ดี
- หน่วยความจำโปรแกรม 16 กิโลไบต์ โดยแบ่งออกเป็น 8 ช่วง ๆ ละ 2 กิโลไบต์ สามาถบรรจุโปรแกรมที่แตกต่างกันได้มากถึง 8 โปรแกรม โดยจะให้ทำงานต่อเนื่อง หรือแยกจากกันก็ได้
- สามารถเขียน และลบหน่วยความจำโปรแกรมได้ 100,000 รอบ เก็บข้อมูลได้นาน 10 ปี
- มีหน่วยความจำแรม 96 ไบต์ โดยแบ่งเป็นแรมภายใน 32 ไบต์และหน่วยความจำสแครตช์แพด (scratch pad RAM) 64 ไบต์
- ความเร็วในการทำงานภาษา PBASIC-2 สูงถึง 100,000 คำสั่งต่อวินาที
- อัตราเร็วในการถ่ายทอดข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรมสูงถึง 115 กิโลบิตต่อวินาที
- ขาพอร์ตอินพุตเอาต์พุตสามารถใช้ในการกำเนิดสัญญาณพัลส์, สัญญาณ DTMF, สัญญาณ PWM และสามารถใช้ในการวัดความถี่ของสัญญาณไฟฟ้าได้โดยตรง
- ความสามารถในการจ่ายกระแสของขาพอร์ต 30 mA สำหรับกระแสซิงก์และกระแสซอร์ส และทุกขาพอร์ตสามารถกำหนดให้เป็นได้ทั้งอินพุต และเอาต์พุต
- ความสามารถในการจ่ายกระแสซอร์ส/ซิงก์สูงสุดต่อกลุ่ม 8 ขา
- สามารถขับ LED และเซอร์โวมอเตอร์ขนาดเล็กได้โดยตรง
- การโปรแกรมข้อมูลใช้การโปรแกรมในลักษณะอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความต้องการกระแสไฟฟ้า 65 mA ในขณะทำงานและ 200 uA เมื่ออยู่ในโหมดประหยัดพลังงาน

2.3 การใช้งานคำสั่งของ BASIC STAMP2

2.3.1 ลักษณะพื้นฐานของภาษาพีเบสิก

โปรแกรมที่เขียนขึ้นเพื่อใช้งานเบสิกแสตมป์ 2 จะเป็นโปรแกรมภาษาเบสิกที่เรียกว่า **พีเบสิก** (PBASIC) ซึ่งมีกฎเกณฑ์ และรูปแบบดังนี้

ค่าคงที่

คือ ค่าของตัวเลขที่ปรากฏในโปรแกรม สามารถแบ่งได้เป็น 4 รูปแบบคือ

1. **ตัวเลขฐานสิบ** (Decimal หรือ DEC) ประกอบด้วยตัวเลข 0-9 เมื่อต้องการกำหนดค่าคงที่ที่เป็นเลขฐานสิบ สามารถกำหนดได้โดยตรง โดยที่ไม่ต้องมีสัญลักษณ์หรือการระบุอื่นใดเป็นพิเศษ
2. **ตัวเลขฐานสิบหก** (Hexadciman หรือ HEX) ประกอบด้วยข้อมูล 0-F เวลาเขียนต้องนำหน้าด้วยเครื่องหมาย $ นำหน้าตัวเลขเสมอ
3. **ตัวเลขฐานสอง** (Binary หรือ BIN) ประกอบด้วยข้อมูล 0 และ 1 ในการเขียนหรืออ้างอิงต้องใส่เครื่องหมาย % นำหน้าตัวเลขเสมอ
4. **รหัสแอสกี** (ASCII) สำหรับรหัสแอสกีนี้เป็นรหัสของตัวอักษร และตัวเลขต่าง ๆ รวมถึงการเว้นวรรค การขึ้นต้นบรรทัดใหม่ ย่อหน้า ซึ่งเกิดขึ้นทุกครั้งที่มีการพิมพ์ข้อความบนโปรแกรมคอมพิวเตอร์ โดยรหัสแอสกีนี้จะมีขนาด 7 บิต นั่นคือ รหัสแอสกีสามารถใช้แทนข้อมูลตัวเลข, ตัวอักษร หรือสัญลักษณ์พิเศษที่เกิดขึ้นในการพิมพ์ทั้งหมดถึง 2^7 หรือ 128 ตัว โดยอักษรและตัวเลขแต่ละตัวจะมีค่าหรือข้อมูลประจำตัวอยู่ ในการระบุว่าข้อความหรือตัวเลขต่อไปนี้เป็นรหัสแอสกีสามารถทำได้โดยเขียนข้อมูลให้อยู่ภายในเครื่องหมายคำพูด หรือ “”

ลาเบลของแอสเดรส

ลาเบล (Label) คือการใช้ข้อความตัวอักษร หรือสัญลักษณ์ในการแทนแอดเดรสของหน่วยความจำโปรแกรมที่เบสิกแสตมป์ 2 จะต้องกระโดดไปทำงานยังตำแหน่งเหล่านั้น ทั้งนี้เพื่อหลีกเลี่ยงการระบุ address ด้วยการเขียนตัวเลข address ของหน่วยความจำโปรแกรมลงในโปรแกรม PBASIC ซึ่งทำให้ยาก และเกิดความผิดพลาดได้ง่าย

ชื่อของลาเบลสามารถให้ขึ้นต้นด้วยตัวอักษร หรือเครื่องหมาย (_) แต่จะไม่สามารถขึ้นต้นด้วยตัวเลข และชื่อของลาเบลจะต้องไม่เป็นคำสงวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อของลาเบลจะต้องตามด้วยเครื่องหมาย : แต่เมื่อมีการกระโดดไปยังลาเบลที่ต้องการ การเขียนคำสั่งเพื่อกระโดดไปนั้นชื่อลาเบลไม่จำเป็นต้องมีเครื่องหมาย : อีก

ลาเบลของค่าคงที่

เช่นเดียวกับลาเบลของแอดเดรส ผู้ใช้งานสามารถกำหนดค่าลาเบลให้กับค่าตัวแปรหรือค่าคงที่ได้ โดยลาเบลนี้ไม่จำเป็นต้องตามด้วยเครื่องหมาย (:) และสามารถกำหนดค่าลาเบลได้โดยใช้คำสั่ง “symbol”

คอมเมนต์ (Comment)

เมื่อต้องการเพิ่มข้อความอธิบายการทำงานของโปรแกรม ข้อความที่เพิ่มขึ้นมานี้จะเรียกว่า *คอมเมนต์* จะต้องขึ้นต้นด้วยเครื่องหมาย ‘ ทุกครั้ง ข้อความที่อยู่หลังเครื่องหมาย ‘ จนจบบรรทัดจะเป็นคำอธิบายทั้งหมด ซึ่งจะไม่มีผลต่อการคอมไพล์ทั้งสิ้น

2.3.2 รูปแบบในการเขียนโปรแกรมพีเบสิก

การเขียนโปรแกรมด้วยภาษาพีเบสิก มีหลักเกณฑ์เบื้องต้นอยู่ 2 ประการคือ

1. ในการเขียนโปรแกรมคำสั่ง หรือตัวแปรต่าง ๆ จะเป็นตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็กก็ได้
2. สามารถเขียนคำสั่งหลาย ๆ คำสั่งบนบรรทัดเดียวกันได้ โดยแยกคำสั่งด้วยการใช้เครื่องหมาย :

คำสั่งต้องห้าม (Reserved Word)

ในการเขียนโปรแกรมพีเบสิกจะต้องมีคำต้องห้ามที่ผู้เขียนโปรแกรมไม่สามารถกำหนด หรือใช้เป็นชื่อตัวแปรได้ เนื่องจากคำเหล่านั้นอาจเป็นชุดของคำสั่งทั้งทางตรงและคำสั่งเทียมของเบสิกแสตมป์ 2 ซึ่งจากตารางข้างล่างจะแสดงคำ หรือข้อความต้องห้าม ซึ่งผู้เขียนโปรแกรมพีเบสิกพต้องหลีกเลี่ยงไม่ใช้

ตารางที่ 2.1 ตารางแสดงคำสั่งต้องห้ามที่ไม่สามารถนำมาใช้เป็นชื่อลาเบลหรือตัวแปรได้

คำสั่งต้องห้ามที่ไม่สามารถนำมาใช้เป็นชื่อลาเบลหรือตัวแปรได้				
A	DIRC	INL	**O**	SHEX
ABS	DIRD	INPUT	OR	SHEX1..SHEX4
AND	DIRH	INS	OUT0..OUT15	SHIFTIN
ASC	DIRL	ISBIN	OUTA	SHIFTOUT
B	DIRS	ISBIN1..ISBIN16	OUTB	SIN
BELL	DTMFOUT	ISHEX	OUTB	SKIP
BKSP	**E**	ISHEX1..ISHEX4	OUTD	SLEEP
BIN	END	**L**	OUTH	SQR
BIN1..BIN14	**F**	LIGHTSON	OUTL	STEP
BIT	FOR	LOOKDOWN	OUTPUT	STOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT0..BIT15	FREQOUT	LOOKUP	OUTS	STR
BRANCH	G	LOW	P	T
BRIGHT	GET	LOWBIT	PAUSE	TAB
BUTTON	GOSUB	LOWNIB	PULSIN	THEN
BYTE	GOTO	LSBFIRST	PULSOUT	TO
C	H	LSBPOST	PUT	TOGGLE
CLS	HEX	LSBPRE	PWM	U
CON	HEX1..HEX4	M	R	UNITOFF
COS	HIGH	MAX	RANDOM	UNITON
COUNT	HIGHBIT	MIN	RCTIME	UNITSOFF
CR	HIGHNIB	MSBFIRST	READ	V
D	HOME	MSBFIRST	REP	VAR
DATA	I	MSBPOST	REV	W
DCD	IHEX	MSBPRE	REVERSE	WAIT
DEBUG	IHEX1..IHEX4	N	RUN	WAITSTR
DEC	IF	NAP	S	WORD
DEC1..DEC5	IN0..IN15	NCD	SBIN	WRITE
DIG	INA	NEXT	SBIN1..SBIN16	X
DIM	INB	NIB	SDEC	XOR
DIR0..DIR15	INC	NIB0..NIB3	SDEC1..SDEC5	XOUT
DIRA	IND	NOT	SERIN	
DIRB	INH		SESROUT	

สัญลักษณ์ของการประมวลผลทางคณิตศาสตร์

ข้อความ ตัวย่อ และสัญลักษณ์ต่าง ๆ ต่อไปนี้เป็นเครื่องหมายที่ใช้แทนการกระทำทางคณิตศาสตร์ภายในโปรแกรมภาษาพีเบสิกของเบสิกแสตมป์ 2

+	แทนการบวก
-	แทนการลบ
*	แทนการคูณ (ส่งผลลัพธ์ไบต์ด้านต่ำกลับมา)
**	แทนการคูณ (ส่งผลลัพธ์ไบต์ด้านสูงกลับมา)
/	แทนการหาร (ส่งผลลัพธ์ที่ได้จากการหารกลับมา)
//	แทนการหาร (ส่งผลลัพธ์ที่เหลือจากการหารกลับมา)
min	เก็บค่าตัวแปรที่มีค่ามากกว่า หรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
max	เก็บค่าตัวแปรที่มีค่าน้อยกว่า หรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
&	แทนการแอนด์ (AND) ทางลอจิก
|	แทนการออร์ (OR) ทางลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

^ แทนการเอ็กคลูซีฟออร์ (XOR) ทางลอจิก

&/ แทนการแอนด์-นอต (AND –NOT) หรือแนนด์ (NAND) ทางลอจิก

|/ แทนการออร์-นอต (OR-NOT) หรือนอร์ (NOR) ทางลอจิก

^/ แทนการเอ็กคลูซีฟออร์-นอต (XOR NOT) หรือเอ็กคลูซีฟนอร์ (XNOR) ทางลอจิก

2.3.3 รายละเอียดชุดคำสั่งของเบสิกแสตมป์ 2SX

คำสั่งหลักในโปรแกรมเบสิกแสตมป์ 2SX มีทั้งสิ้น 64 คำสั่ง สามารถแบ่งออกเป็นกลุ่มตามลักษณะการทำงานได้ 8 กลุ่ม ดังนี้

2.3.3.1 **กลุ่มคำสั่งจัดการข้อมูล** (data command) มี 9 คำสั่งคือ

ตารางที่ 2.2 ตารางแสดงกลุ่มคำสั่งจัดการข้อมูล

CON	ใช้กำหนดค่าคงที่
VAR	ใช้กำหนดตัวแปร
DEBUG	แสดงค่าของตัวแปรผ่านทางคอมพิวเตอร์
DATA	เก็บข้อมูลลงในหน่วยความจำอีอีพีรอม
READ	อ่านข้อมูลระดับไบต์จากหน่วยความจำอีอีพรอม
WRITE	เขียนข้อมูลระดับไบต์ไปเก็บในหน่วยความจำอีอีพรอม
RANDOM	สุ่มตัวเลข
GET	อ่านข้อมูลจากหน่วยความจำสแครตช์แพด
PUT	เก็บข้อมูลลงในหน่วยความจำสแครตช์แพด

2.3.3.2 **กลุ่มคำสั่งควบคุมลำดับการทำงาน** (flow control command) มี 12 คำสั่งคือ

ตารางที่ 2.3 ตารางแสดงกลุ่มคำสั่งควบคุมลำดับการทำงาน

FOR ... NEXT	กำหนดจำนวนรอบที่ต้องการวนลูป หรือทำงานซ้ำ
STOP	หยุดการทำงาน
END	ทำงานในโหมด SLEEP จนกว่าจะมีการจ่ายไฟ หรือต่อเบสิกแสตมป์ 2SX เข้ากับคอมพิวเตอร์
PAUSE	หน่วงเวลา 0 – 65,536 มิลลิวินาที
NAP	หยุดการทำงานในช่วงเวลาสั้น ๆ
SLEEP	ทำงานในโหมดประหยัดพลังงาน
GOTO	กระโดดไปยังแอดเดรสใด ๆ
GOSUB	กระโดดไปยังโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IF ... THEN	เปรียบเทียบเงื่อนไขก่อนกระโดด
BRANCH	กระโดดไปยังตำแหน่งที่กำหนดตามค่าของตัวแปร
RETURN	กระโดดออกจากโปรแกรมย่อย
RUN	รันโปรแกรมที่ต้องการ

2.3.3.3 **กลุ่มคำสั่งจัดการสัญญาณดิจิตอล** มี 11 คำสั่งคือ

ตารางที่ 2.4 ตารางแสดงกลุ่มคำสั่งจัดการสัญญาณดิจิตอล

INPUT	กำหนดให้ทำงานเป็นอินพุต
OUTPUT	กำหนดให้ทำงานเป็นเอาต์พุต
HIGH	ทำให้ขาเอาต์พุตเป็นลอจิก “1”
LOW	ทำให้ขาเอาต์พุตเป็นลอจิก “0”
TOGGLE	ทำให้ขาเอาต์พุตกลับสถานะลอจิก
REVERSE	ทำการสลับเปลี่ยนระหว่างขาอินพุตกับขาเอาต์พุต
PULSOUT	ส่งสัญญาณพัลส์ออก
PULSIN	วัดสัญญาณพัลส์อินพุต
COUNT	นับจำนวนไซเกิลของสัญญาณอินพุต
BUTTON	ตรวจสอบการรับค่าสวิตช์
XOUT	กำเนิดรหัสควบคุมสำหรับอุปกรณ์ต่อพ่วง X-10

2.3.3.4 **กลุ่มคำสั่งจัดการสัญญาณอะนาลอก** มี 4 คำสั่งคือ

ตารางที่ 2.5 ตารางแสดงกลุ่มคำสั่งจัดการสัญญาณอะนาลอก

PWM	สร้างสัญญาณ PWM ขนาด 8 บิตออกไปทางขาพอร์ต
RCTIME	วัดค่าเวลาการประจุและคายประจุของวงจร RC
FREQOUT	กำเนิดสัญญาณไซน์หนึ่งถึงสองความถี่ตั้งแต่ 0-82.917 kHZ
DTMFOUT	กำเนิดสัญญาณ DTMF ของระบบโทรศัพท์

2.3.3.5 **กลุ่มคำสั่งจัดการข้อมูลอนุกรม** มี 4 คำสั่งคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ตารางแสดงกลุ่มคำสั่งจัดการข้อมูลอนุกรม

SERIN	รับข้อมูลอนุกรมเข้า มีรูปแบบข้อมูลแบบ N81 หรือ E71
SEROUT	ส่งข้อมูลอนุกรมในรูปแบบ N81 หรือ E71 ออก
SHIFTIN	เลื่อนข้อมูลเขาแบบอนุกรม
SHIFTOUT	เลื่อนข้อมูลออกแบบอนุกรม

2.3.3.6 **กลุ่มคำสั่งตารางข้อมูล** มี 2 คำสั่งคือ

ตารางที่ 2.7 ตารางแสดงกลุ่มคำสั่งตารางข้อมูล

LOOKUP	เปิดตารางข้อมูล
LOOKDOWN	ค้นหาตัวเลขที่เหมือนกัน แล้วเก็บค่าในตัวแปร

2.3.3.7 **กลุ่มคำสั่งประมวลผลทางคณิตศาสตร์** มี 16 คำสั่งคือ

ตารางที่ 2.8 ตารางแสดงกลุ่มคำสั่งประมวลผลทางคณิตศาสตร์

+	คำสั่งการบวก
-	คำสั่งการลบ
*	คำสั่งการคูณ โดยส่งค่าผลลัพธ์ด้านเวิร์ดต่ำกลับมา
**	คำสั่งการคูณ โดยส่งค่าผลลัพธ์ด้านเวิร์ดสูงกลับมา
/	คำสั่งการหาร โดยส่งผลหารจำนวนเต็มที่ได้จากการหาร
//	คำสั่งการหาร โดยส่งค่าเศษที่เหลือจากการหารกลับมา
*/	คำสั่งคูณเลขทศนิยม
>>	คำสั่งเลื่อนข้อมูลไปทางขวา มีค่าเท่ากับหารด้วย 2
<<	คำสั่งเลื่อนข้อมูลไปทางซ้าย มีค่าเท่ากับคูณด้วย 2
MIN	เก็บค่าตัวแปรที่มีค่ามากกว่าหรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
MAX	เก็บค่าตัวแปรที่มีค่าน้อยกว่ากว่าหรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
ABS	ประมวลผลเฉพาะข้อมูลตัวเลขแบบไม่คิดเครื่องหมาย
SQR	ถอดรากที่สอง
SIN	คำนวณค่า SINE ทางตรีโกณมิติ
COS	คำนวณค่า COSINE ทางตรีโกณมิติ
DIG	เรียกค่าของข้อมูลในหลักที่กำหนดในรูปแบบเลขฐานสิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.8 **กลุ่มคำสั่งประมวลผลทางลอจิก** มีทั้งสิ้น 6 คำสั่งคือ

ตารางที่ 2.9 ตารางแสดงกลุ่มคำสั่งประมวลผลทางลอจิก

&	แอนด์ (AND) ทางลอจิก
|	ออร์ (OR) ทางลอจิก
^	เอ็กคลูซีพออร์ (XOR) ทางลอจิก
REV	สลับบิตข้อมูล
DCD	เลือกเขตบิตที่ต้องการของข้อมูลขนาด 16 บิต
NCD	แสดงบิตนัยสำคัญของข้อมูลขนาด 16 บิต

2.3.4 **รายละเอียดการใช้งานของแต่ละคำสั่ง**

BRANCH

รูปแบบการใช้งานคำสั่ง

```
BRANCH offset, [label0, label1,...labelN]
```

ใช้สำหรับกระโดดไปยังแอดเดรสที่กำหนดตามค่า offset (ค่า offset ต้องอยู่ภายในช่วงแอดเดรสด้วย)

offset เป็นค่าคงที่ หรือตัวแปร ที่ใช้กำหนดตำแหน่งที่จะกระโดดไปตามรายการที่กำหนดในแอดเดรส โดยค่าจะต้องอยู่ในช่วง 0-N

label เป็นชื่อลาเบลที่ใช้กำหนดตำแหน่งของแอดเดรสที่จะกระโดดไป

BUTTON

รูปแบบการใช้งานคำสั่ง

```
BUTTON pin,downstate,delay,rate,bytevariable,
targetstate,address
```

เป็นคำสั่งสำหรับอ่านค่าสวิตช์อินพุต สามารถกำหนดให้มีการป้องกันการกระเพื่อมอันเนื่องมาจากการกดสวิตช์หรือการเบาซ์ (bounce) โดยคำสั่งนี้จะกำหนดให้เบสิกแสตมป์ 2SX สร้างสัญญาณการกดสวิตช์ซ้ำ (auto-repeat) และกระโดดไปยังตำแหน่งที่ระบุถ้าปุ่มที่กดนั้นตรงกับค่าที่กำหนดไว้ที่ target state

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยวงจรสร้างสัญญาณการกดคีย์นี้สามารถให้แอกตีฟ หรือทำงานที่ลอจิก "0" หรือ "1" ก็ได้

pin เป็นค่าตัวแปร หรือคาคงที่มีค่าตั้งแต่ 0-15 เพื่อกำหนดตำแหน่งของขาพอร์ตของเบสิกแสตมป์2SX ที่ต้องการใช้งาน โดยขาที่ระบุนี้จะถูกกำหนดให้เป็นขาอินพุต

downstate เป็นค่าตัวแปร หรือค่าคงที่ มีค่าเท่ากับ "0" หรือ "1" ใช้สำหรับกำหนดสถานะลอจิกเมื่อสวิตช์ถูกกด

delay เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0-255 เพื่อใช้กำหนดระยะห่างของช่วงเวลาก่อนสร้างสัญญาณการกดสวิตช์ซ้ำ

rate เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0-255 เพื่อใช้กำหนดจำนวนของวงรอบในระหว่างการสร้างสัญญาณกดสวิตช์ซ้ำ

bytevariable เป็นตำแหน่งพื้นที่ทำงานของคำสั่ง BUTTON ค่านี้จะต้องกำหนดให้เป็น 0 ก่อนใช้คำสั่ง BUTTON ครั้งแรก

targetstate เป็นค่าตัวแปร หรือค่าคงที่ มีค่า 0 หรือ 1 เพื่อใช้กำหนดเงื่อนไขของการกดสวิตช์ ถ้าค่านี้เป็น 0 หมายถึงไม่มีการกดสวิตช์ ถ้าเป็น 1 หมายถึงสวิตช์ถูกกด

address เป็นลาเบลที่กำหนดตำแหน่งที่โปรแกรมกระโดดไป ถ้าปุ่มที่กดมีค่าตรงกับค่า targetstate

CON

รูปแบบการใช้งานคำสั่ง

```
Id CON value
```

เป็นคำสั่งเทียมใช้สำหรับกำหนดค่าคงที่ (value) ให้กับตัวแปร (id) มักปรากฏอยู่ในส่วนหัวของโปรแกรม

COUNT

รูปแบบการใช้งานคำสั่ง

```
COUNT pin, period, variable
```

ใช้นับจำนวนรอบ หรือไซเกิลสัญญาณอินพุตที่ขาพอร์ตของเบสิกแสตมป์ 2SX ตามระยะเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

pin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0 – 15 เพื่อกำหนดขอพอร์ตของเบสิกแสตมป์ 2SX

period เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 1 ถึง 65,535 มีหน่วยเป็น 0.4 มิลลิวินาทีเพื่อใช้กำหนดช่วงเวลากการนับ

DATA

รูปแบบการใช้งานคำสั่ง

```
Label DATA data_list
```

ใช้ในการกำหนดพื้นที่ หรือขอบเขตของหน่วยความจำอีอีพรอมที่ใช้เป็นหน่วยความจำข้อมูลในเบสิกแสตมป์ 2SX คำสั่ง DATA มีลักษณะคล้าย ๆ กับคำสั่ง CON หรือ VAR โดย label จะเป็นชื่อของตัวแปร ส่วน data_list คือข้อมูลที่ต้องการกำหนดลงในหน่วยความจำอีอีพรอม

DEBUG

รูปแบบการใช้งานคำสั่ง

```
DEBUG outputData{,outputData...}
```

ใช้ในการแสดงค่าตัวแปร และข้อความบนหน้าจอคอมพิวเตอร์ด้วยโปรแกรมหลักของเบสิกแสตมป์ 2SX

outputData เป็นส่วนประกอบของข้อความ, ตัวแปร, ค่าคงที่ และตัวอักษรควบคุม

คำสั่ง DEBUG 1 คำสั่งสามารถแทรกข้อความได้หลายรูปแบบ และในโปรแกรมยังสามารถแทรกคำสั่ง DEBUG ได้หลาย ๆ คำสั่งด้วย นอกจากนี้คำสั่ง DEBUG ยังมีฟังก์ชั่นพิเศษภายในอีก 6 ฟังก์ชั่น ซึ่งแทนด้วยตัวย่อดังนี้

ตัวย่อ	ค่า	ผลที่เกิดขึ้น
CLS	0	เคลียร์จอภาพ
HOME	1	กำหนดให้เคอร์เซอร์ไปอยู่ที่ตำแหน่งมุมซ้ายบนของจอภาพ
BELL	7	ขับเสียงออกทางลำโพงคอมพิวเตอร์
BKSP	8	ทำให้เกิดช่วงว่างพร้อมกับถอยหลัง 1 ตำแหน่ง
TAB	9	เลื่อนตำแหน่งเคอร์เซอร์ไปครั้งละ 8 ตำแหน่งตัวอักษร
CR	13	ขึ้นบรรทัดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

43037

DTMFOUT

รูปแบบการใช้งานคำสั่ง

```
DTMFOUT pin,{ontime,offtime,}{,tone...}
```

เป็นคำสั่งสำหรับสร้างสัญญาณสองโทนหลายความถี่หรือ DTMF(dual-tone multi-frequency) ซึ่งเป็นสัญญาณเสียงการกดปุ่มของเครื่องโทรศัพท์สมัยใหม่ หรือระบบโทน

pin เป็นค่าตัวแปร หรือค่าข้อมูลมีค่าตั้งแต่ 0 – 15 ใช้ในการระบุขาที่ใช้งาน

ontime ใช้ในการกำหนดระยะเวลาสร้างสัญญาณ DTMF มีหน่วยเป็น 0.4 มิลลิวินาที สามารถกำหนดค่าได้ตั้งแต่ 0 – 65,536

offtime ใช้กำหนดช่วงเวลาหยุดหลังจากการสร้างสัญญาณ DTMF แล้ว มีหน่วยเป็น 0.4 มิลลิวินาที สามารถกำหนดค่าได้ตั้งแต่ 0 – 65,536

tone เป็นค่าตัวแปรหรือค่าคงที่มีค่าตั้งแต่ 0 –15 ใช้สำหรับกำหนดค่าโทนของสัญญาณ DTMF ที่จะส่งออกไป

END

รูปแบบการใช้งานคำสั่ง

```
END
```

ใช้เมื่อต้องการจบโปรแกรม นำเบสิกแสตมป์ 2SX เข้าสู่โหมดประหยัดพลังงาน

FOR...NEXT

รูปแบบการใช้งานคำสั่ง

```
FOR variable = start to end
{STEP stepVal} ...NEXT
```

ใช้สำหรับสร้างลูปการทำงานของโปรแกรมซึ่งอยู่ระหว่างคำสั่ง FOR กับ NEXT การเพิ่มค่าหรือการลดค่าตัวแปร variable ขึ้นอยู่กับค่า stepVAL โปรแกรมจะวนลูปซ้ำไปเรื่อย ๆ จนกว่าค่าที่เริ่มหรือลดนั้นจะมีค่าเท่ากับ end

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

variable เป็นข้อมูลชนิดบิต, นิบเบิล (4 ไบต์), ไบต์ หรือเวิร์ดที่ใช้เป็นตัวแปรสำหรับการนับค่าการวนลูป

start เป็นค่าตัวแปร หรือค่าคงที่ที่ใช้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร

end เป็นค่าตัวแปร หรือค่าคงที่ใช้ใหรับกำหนดค่าสุดท้ายของแปร variable

stepVal เป็นจำนวนของการเพิ่มขึ้น หรือลดลงของค่าตัวแปร variable

FREQOUT

รูปแบบการใช้งานคำสั่ง

```
FREQOUT pin, duration, freq1{,freq2}
```

ใช้สำหรับกำเนิดสัญญาณไซน์หนึ่ง หรือสองสัญญาณในช่วงเวลาที่กำหนด

pin เป็นค่าตัวแปร หรือค่าคงที่มีค่าเท่ากับ 0 – 15 เพื่อกำหนดขาพอร์ตที่ใช้งาน

duration เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 1 – 65,535 ใช้กำหนดช่วงเวลาสัญญาณที่กำเนิดขึ้น มีหน่วยเป็น 0.4 มิลลิวินาที

freq เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0 – 32,767 ใช้เพื่อกำหนดค่าความถี่หน่วยเป็น 2.5 Hz ดังนั้นจึงสามารถกำเนิดสัญญาณได้ตั้งแต่ 0 – 82.917 kHz

freq2 เป็นค่าตัวแปร หรือค่าคงที่ที่ใช้เพื่อกำหนดค่าคงที่หน่วยเป็น 2.5 Hz

GET

รูปแบบการใช้งานคำสั่ง

```
GET location, variable
```

เป็นคำสั่งอ่านข้อมูลจากหน่วยความจำสแครตช์แพดแรมไปเก็บไว้ในตัวแปรที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

location เป็นค่าตัวแปร หรือค่าคงที่แสดงตำแหน่งของหน่วยความจำส แครตช์แพดแรมภายในเบสิกแสตมป์ 2 SX มีค่าตั้งแต่ 0 – 63

variable เป็นตัวแปรที่ต้องการเก็บข้อมูล

GOSUB

รูปแบบการใช้งานคำสั่ง

```
GOSUB AddressLabel
```

เป็นคำสั่งสำหรับกำหนดให้เบสิกแสตมป์ 2SX กระโดดไปทำงานยังแอดเดรสที่กำหนดโดย AddressLable

AddressLabel เป็นแหน่งลาเบลระบุตำแหน่งที่จะกระโดดไป

GOTO

รูปแบบการใช้งานคำสั่ง

```
GOTO AddressLabel
```

เป็นคำสั่งสำหรับกำหนดให้เบสิกแสตมป์ 2SX กระโดดไปทำงานยังแอดเดรสที่กำหนดโดย AddressLable

AddressLabel เป็นแหน่งลาเบลระบุตำแหน่งที่ปลายทาง

HIGH

รูปแบบการใช้งานคำสั่ง

```
HIGH pin
```

ใช้ในการกำหนดให้ขาพอร์ตเอาต์พุตมีลอจิกเป็น "1"

pin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0 – 15 เพื่อกำหนดขาพอร์ตของเบสิกแสตมป์ 2SX ที่ต้องการใช้งาน

IF...THEN

รูปแบบการใช้งานคำสั่ง

```
IF condition THEN AddressLabel
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการตรวจสอบเงื่อนไข และถ้าเงื่อนไขเป็นจริง กระโดดไปยังแอดเดรสที่ระบุไว้ใน AddressLabel

condition เป็นการกำหนดเงื่อนไข สามารถกำหนดเป็นสมการ แล้วทำการตรวจสอบสมการนี้ได้ว่าเป็นจริง หรือเป็นเท็จ

AddressLabel เป็นค่าลาเบลที่กำหนดตำแหน่งที่โปรแกรมจะกระโดดไปถ้าเงื่อนไขที่ตรวจสอบเป็นจริง

INPUT

รูปแบบการใช้งานคำสั่ง

```
INPUT pin
```

ใช้กำหนดให้ขาพอร์ตที่ต้องการเป็นอินพุต

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

LOOKDOWN

รูปแบบการใช้งานคำสั่ง

```
LOOKDOWN value,{Comparisonop,}
[value0, value1,...valueN],resultVariable
```

เป็นคำสั่งที่ใช้สำหรับเก็บค่าตัวชี้ของตัวเลขที่นำมาเปรียบเทียบแล้วผลการเปรียบเทียบเป็นจริง

value เป็นค่าตัวแปร หรือค่าคงที่สำหรับเปรียบเทียบกับค่าที่อยู่ใน value0...valueN

ComparisonOP เป็นการกำหนดเงื่อนไขในการเปรียบเทียบเพิ่มเติม ซึ่งจะใช้หรือไม่ก็ได้

value0, value1... สร้างบัญชีรายชื่อของค่าข้อมูลโดยค่าที่กำหนดนี้เป็นค่าตัวแปรหรือค่าคงที่ก็ได้ มีขนาดสูงสุด 16 บิต

ResultVariable เป็นตัวแปรสำหรับเก็บค่าตัวชี้ ถ้าหากการเปรียบเทียบพบค่าที่ตรงกัน

LOOKUP

รูปแบบการใช้งานคำสั่ง

```
LOOKUP index,
[value0, value1,...valueN],resultVariable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งที่ใช้ในการเปิดตารางข้อมูล โดยคำสั่ง LOOKUP จะมองหาค่าที่ชี้โดย index และนำค่าที่ได้ไปเก็บใน ResultVariable ถ้าค่า index มีค่ามากกว่าค่าที่อยู่ในรายการของ Value ค่า ResultVariable จะไม่มีการเปลี่ยนแปลง

index เป็นค่าตัวแปร หรือค่าคงที่เพื่อกำหนดเป็นตัวชี้ค่าตัวเลขจากรายการ

value0, value1... สร้างบัญชีรายชื่อของค่าข้อมูลโดยค่าที่กำหนดนี้เป็นค่าตัวแปรหรือค่าคงที่ก็ได้ มีขนาดสูงสุด 16 บิต

ResultVariable เป็นตัวแปรสำหรับเก็บค่าตัวชี้ ถ้าหากการเปรียบเทียบพบค่าที่ตรงกัน

LOW

รูปแบบการใช้งานคำสั่ง

LOW pin

ใช้ในการกำหนดให้ขาพอร์ตที่เป็นเอาต์พุตมีลอจิกเป็น "0"

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

NAP

รูปแบบการใช้งานคำสั่ง

NAP period

เป็นคำสั่งสำหรับกำหนดให้เบสิกแสตมป์ 2SX เข้าสู่โหมดประหยัดพลังงานในช่วงเวลาสั้น ๆ

period เป็นค่าตัวแปร หรือค่าคงที่สำหรับกำหนดคาบเวลาที่เข้าสู่โหมดประหยัดพลังงาน

OUTPUT

รูปแบบการใช้งานคำสั่ง

OUTPUT pin

กำหนดให้ขาพอร์ตที่ต้องการติดต่อเป็นเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

PAUSE

รูปแบบการใช้งานคำสั่ง

```
PAUSE milliseconds
```

คำสั่งนี้จะหยุดการทำงานของโปรแกรมชั่วคราวตามช่วงเวลาที่ระบุใน milliseconds

milliseconds เป็นค่าตัวแปร หรือค่าคงที่เพื่อใช้กำหนดค่าของช่วงเวลาหยุดพัก (pause) มีหน่วยเป็นมิลลิวินาที ค่า pause สามารถกำหนดช่วงเวลาได้นานที่สุดเท่ากับ 65,535 มิลลิวินาที

PULSIN

รูปแบบการใช้งานคำสั่ง

```
PULSIN pin, state, resultVariable
```

ใช้วัดสัญญาณพัลส์ที่ป้อนเข้ามายังขาพอร์ตของเบสิกแสตมป์ 2SX ทุก ๆ 0.8 ไมโคร วินาที

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการใช้สำหรับวัดจำนวนพัลส์

state เป็นค่าตัวแปร หรือค่าคงที่ มีค่าเท่ากับ 0 หรือ 1 เพื่อใช้กำหนดค่าลอจิกเริ่มต้นในการวัด ถ้าค่า state = 0 จะเป็นการกำหนดให้นับเมื่อพัลส์เปลี่ยนจากลอจิก “1” เป็น “0” ถ้าค่า state = 1 จะเป็นการกำหนดให้นับเมื่อพัลส์เปลี่ยนจากลอจิก “0” เป็น “1”

resultVariable เป็นค่าตัวแปร หรือค่าคงที่ใช้เก็บจำนวนพัลส์ที่วัดได้

PULSOUT

รูปแบบการใช้งานคำสั่ง

```
PULSOUT pin, time
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งพัลส์ลอจิกสูงที่มีคาบเวลาตั้งแต่ 0.8 ไมโครวินาทีถึง 52 มิลลิวินาที ออกไปทางขาพอร์ตที่กำหนด

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

time เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0 – 65,535 ใช้กำหนดคาบเวลาของพัลส์มีค่าหน่วยละ 0.8 ไมโครวินาที

PUT

รูปแบบการใช้งานคำสั่ง

```
PUT location, value
```

เป็นคำสั่งสำหรับนำข้อมูลในตัวแปร หรือค่าคงที่ไปเก็บไว้ในหน่วยความจำสแครตช์แพดแรม

location เป็นตัวแปร หรือค่าคงที่แสดงตำแหน่งของหน่วยความจำสแรตช์แพดแรมภายในเบสิกแสตมป์ 2SX มีค่าตั้งแต่ 0 – 62

value เป็นตัวแปร หรือค่าคงที่ที่ต้องการเก็บไว้ในหน่วยความจำสแครตช์แพดแรม

PWM

รูปแบบการใช้งานคำสั่ง

```
PWM pin, duty, cycles
```

ใช้สำหรับเปลี่ยนข้อมูลดิจิตอลเป็นค่าสัญญาณอะนาลอกผ่านการมอดูเลชั่นทางความกว้างของพัลส์ หรือ PWM (Pulse-width Mudulation)

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

duty เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0 – 255 ใช้กำหนดระดับสัญญาณ PWM ความละเอียด 8 บิต

cycles เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0 – 65,535 ใช้กำหนดค่าคาบเวลาโดยประมาณของสัญญาณ PWM หน่วยเป็น 0.4 มิลลิวินาที

RANDOM

รูปแบบการใช้งานคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RANDOM variable

เป็นคำสั่งจำลองการสุ่มตัวเลข

variable เป็นตัวแปรสำหรับกำหนดจำนวนสูงสุดของการสุ่มตัวเลข ต้องกำหนดเป็นข้อมูลแบบไบต์ (byte) หรือแบบเวิร์ด (word)

RCTIME

รูปแบบการใช้งานคำสั่ง

RCTIME pin, state, resultVariable

เป็นคำสั่งสำหรับนับค่าของเวลาที่ขาพอร์ตยังคงสถานะเดิมอยู่ ใช้ในการวัดค่าการเก็บประจุ และคายประจุของตัวเก็บประจุต้านทานในวงจร RC

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

state เป็นค่าตัวแปร หรือค่าคงที่มีค่าเท่ากับ 1 หรือ 0 เพื่อกำหนดสภาวะของขาเมื่อกระทำคำสั่ง RCTIME

resultVariable เป็นค่าตัวแปร หรือค่าคงที่ มีค่าเท่ากับ 0 ถึง 65,535 สำหรับเก็บค่าคาบเวลามีค่าหน่วยละ 0.8 ไมโครวินาที

READ

รูปแบบการใช้งานคำสั่ง

READ location, variable

ใช้ในการอ่านค่าของหน่วยความจำอีอีพรอม และเก็บค่าไว้ในตัวแปรที่กำหนด

location เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 0 – 2047 เพื่อใช้กำหนดตำแหน่งแอดเดรสที่ใช้ในการอ่านหน่วยความจำอีอีพรอม

variable เป็นตัวแปรสำหรับที่เก็บข้อมูลที่อ่านได้จากหน่วยความจำอีอีพรอม มีค่าตั้งแต่ 0 – 255

RETURN

รูปแบบการใช้งานคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RETURN

เป็นคำสั่งสำหรับกำหนดให้เบสิกแสตมป์ 2SX กระโดกลับไปทำงานยังโปรแกรมหลักหลังจากการทำงานที่โปรแกรมย่อยเสร็จสิ้น

REVERSE

รูปแบบการใช้งานคำสั่ง

REVERSE pin

ใช้เปลี่ยนการทำงานของขาพอร์ตจากอินพุตเป็นเอาต์พุต หรือจากเอาต์พุตเป็นอินพุต

pin เป็นค่าคงที่ หรือค่าตัวแปรมีค่าอยู่ระหว่าง 0 – 15 ใช้เลือกขาพอร์ตเบสิกแสตมป์ 2SX ที่ต้องการ

RUN

รูปแบบการใช้งานคำสั่ง

RUN program

เป็นคำสั่งให้เบสิกแสตมป์ 2SX ไปทำงานยังโปรแกรมที่กำหนด

program เป็นตัวแปร หรือค่าคงที่สำหรับกำหนดตำแหน่งของโปรแกรมที่ต้องการให้ เบสิกแสตมป์ 2SX ทำงาน มีค่าตั้งแต่ 0 – 7

SERIN

รูปแบบการใช้งานคำสั่ง

SERIN rpin{\fpin},baudmode,{plabel,}{timeout,tlabel,}{inputData}

เป็นคำสั่งที่ใช้ในการรับข้อมูลอนุกรมแบบอะซิงโครนัส โดยเบสิกแสตมป์ 2SX สามารถรับข้อมูลอนุกรมด้วยอัตรา 608 ถึง 115,200 บิตต่อวินาที

rpin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0 – 16 เพื่อกำหนดขาพอร์ตสำหรับรับข้อมูลอินพุตอนุกรม เมื่อต้องการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

fpin เป็นตัวเลือกเพิ่มเติม โดยจะเป็นตัวแปรหรือค่าคงที่ตั้งแต่ 0 – 15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX มาเป็นขาควบคุมการไหลของข้อมูล

baudmode เป็นค่าคัวแปร หรือค่าคงที่ขนาด 16 บิตเพื่อกำหนดคุณสมบัติต่าง ๆ ของการสื่อสารข้อมูลแบบอนุกรม โดยข้อมูล 13 บิตด้านต่ำใช้สำหรับการกำหนดค่าอัตราเร็วในการถ่ายทอดข้อมูล หรืออัตราบอดหรือบอดเรต

plabel เป็นตัวเลือกเพิ่มเติมสำหรับกำหนดลาเบลที่โปรแกรมจะกระโดดไปในกรณีเกิดข้อผิดพลาดที่การตรวจสอบพาริตี้

timeout เป็นตัวเลือกเพิ่มเติม เป็นตัวแปรหรือค่าคงที่ 0 – 65,535 เพื่อกำหนดเวลาในการรอรับข้อมูล

tlabel เป็นตัวเลือกเพิ่มเติมสำหรับกำหนดตำแหน่งลาเบลที่โปรแกรมจะกระโดดไปเมื่อเกิด timeout หรือไม่มีการรับข้อมูลเขามาเมื่อถึงเวลาที่กำหนดโดยตัวแปร timeout

inputData เป็นตัวแปรสำหรับเก็บข้อมูลที่รับเข้ามา

SEROUT

รูปแบบการใช้งานคำสั่ง

```
SEROUT tpin, baudmode,{pace,}[outputData]
SEROUT   tpin\fpin,baudmode,{timeout,tlabel,}
[outputData]
```

เป็นคำสั่งสำหรับส่งข้อมูลอนุกรมแบบอะซิงโครนัส สามารถส่งข้อมูลอนุกรมได้ที่ความเร็ว 608 ถึง 115,200 บิตต่อวินาที

fpin เป็นตัวเลือกเพิ่มเติม โดยจะเป็นตัวแปรหรือค่าคงที่ตั้งแต่ 0 – 15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX มาเป็นขาควบคุมการไหลของข้อมูล

baudmode เป็นค่าคัวแปร หรือค่าคงที่ขนาด 16 บิตเพื่อกำหนดคุณสมบัติต่าง ๆ ของการสื่อสารข้อมูลแบบอนุกรม โดยข้อมูล 13 บิตด้านต่ำใช้สำหรับการกำหนดค่าอัตราเร็วในการถ่ายทอดข้อมูล หรืออัตราบอดหรือบอดเรต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pace เป็นตัวเลือกเพิ่มเติมอาจเป็นตัวแปร หรือค่าคงที่ 0 – 65,535 ใช้ในการกำหนดค่าเวลาที่หยุดรอระหว่างการรับส่งข้อมูล มีหน่วยเป็นมิลลิวินาที

outputData เป็นตัวแปรหรือค่าคงที่ของข้อมูลที่ใช้ในการส่ง

fpin เป็นตัวเลือกเพิ่มเติมมีค่าตั้งแต่ 0 – 15 เพื่อเลือกขาพอร์ตของเบสิกแสตมป์ 2 SX ที่จะใช้เป็นขาควบคุมการไหลของข้อมูล

timeout เป็นตัวเลือกเพิ่มเติม เป็นตัวแปรหรือค่าคงที่ 0 – 65,535 เพื่อกำหนดเวลาในการรอรับข้อมูล

tlabel เป็นตัวเลือกเพิ่มเติมสำหรับกำหนดตำแหน่งลาเบลที่โปรแกรมจะกระโดดไปเมื่อเกิด timeout หรือไม่มีการรับข้อมูลเขามาเมื่อถึงเวลาที่กำหนดโดยตัวแปร timeout

SHIFTIN

รูปแบบการใช้งานคำสั่ง

```
SHIFTIN   dpin,cpin,mode,[result{\bits}{,result
{\bits}...}]
```

เป็นการอ่านค่าข้อมูลจากอุปกรณ์ที่มีการสื่อสารอนุกรมแบบซิงโครนัส

dpin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0-15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX สำหรับข้อมูลแบบซิงโครนัส หลังจากทำคำสั่งเรียบร้อยแล้ว ขานี้จะอยู่ในสถานะอินพุต

cpin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0-15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX สำหรับเชื่อต่อกับขาอินพุตสัญญาณนาฬิกาของอุปกรณ์ซิงโครนัสที่ต้องการติดต่อด้วย

mode มีค่าอยู่ระหว่าง 0-3 เพื่อกำหนดรูปแบบการส่งสัญญาณนาฬิกา

result เป็นตัวแปรบิต, นิปเปิล, ไบต์ หรือเวิร์ดที่ใช้เก็บข้อมูลที่รับเข้ามา

bits ใช้กำหนดจำนวนบิตอินพุตที่รับเข้ามา มีค่า 1-16 ถ้าไม่มีการกำหนด ค่าปกติจะเท่ากับ 8

SHIFTOUT

รูปแบบการใช้งานคำสั่ง

```
SHIFTIN   dpin,cpin,mode,[data{\bits}{,data
{\bits}...}]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้เลื่อนข้อมูลออกไปยังอุปกรณ์สื่อสารข้อมูลอนุกรมแบบซิงโครนัส

dpin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0-15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX สำหรับข้อมูลแบบซิงโครนัส หลังจากทำคำสั่งเรียบร้อยแล้ว ขานี้จะอยู่ในสถานะอินพุต

cpin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0-15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX สำหรับเชื่อต่อกับขาอินพุตสัญญาณนาฬิกาของอุปกรณ์ซิงโครนัสที่ต้องการติดต่อด้วย

mode มีค่าอยู่ระหว่าง 0-3 เพื่อกำหนดรูปแบบการส่งสัญญาณนาฬิกา

mode เป็นค่าข้อมูลที่ต้องการส่งออกไป

bits ใช้กำหนดจำนวนบิตอินพุตที่รับเข้ามา มีค่า 1-16 ถ้าไม่มีการกำหนด ค่าปกติจะเท่ากับ 8

SLEEP

รูปแบบการใช้งานคำสั่ง

SLEEP seconds

กำหนดให้เบสิกแสตมป์ 2SX เข้าสู่โหมดประหยัดพลังงาน ด้วยระยะเวลาตามที่กำหนด มีหน่วยเป็นวินาที

seconds เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 1-65,535 ใช้กำหนดคาบเวลาในการเข้าสู่โหมดประหยัดพลังงาน มีหน่วยเป็นวินาที

STOP

รูปแบบการใช้งานคำสั่ง

STOP

หยุดการทำงานของโปรแกรม

SLEEP

รูปแบบการใช้งานคำสั่ง

SLEEP seconds

กำหนดให้เบสิกแสตมป์ 2SX เข้าสู่โหมดประหยัดพลังงาน ด้วยระยะเวลาตามที่กำหนด มีหน่วยเป็นวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

seconds เป็นค่าตัวแปร หรือค่าคงที่ มีค่าตั้งแต่ 1-65,535 ใช้กำหนดคาบเวลาในการเข้าสู่โหมดประหยัดพลังงาน มีหน่วยเป็นวินาที

TOGGLE

รูปแบบการใช้งานคำสั่ง

```
TOGGLE
```

เปลี่ยนสถานะลอจิกของขาพอร์ต

pin เป็นค่าตัวแปร หรือค่าคงที่มีค่าตั้งแต่ 0-15 ใช้เลือกขาพอร์ตของเบสิกแสตมป์ 2SX ที่ต้องการใช้งาน

VAR

รูปแบบการใช้งานคำสั่ง

```
Symbol VAR size
```

เป็นคำสั่งเทียมใช้สำหรับกำหนดขนาดหรือชนิดของตัวแปร

symbol เป็นชื่อของตัวแปรที่ต้องการกำหนด จะต้องขึ้นต้นด้วยตัวอักษรเสมอ

size เป็นชื่อของขนาดของข้อมูลที่ต้องการสามารถกำหนดได้ 4 ขนาดคือ บิต(bit), นิบเบิล, หรือนิบ(nib: 4 bit), ไบต์(byte : 8bit) และเวิร์ด(word : 16 bit)

WRITE

รูปแบบการใช้งานคำสั่ง

```
WRITE address,byte
```

เขียนข้อมูล 1 ไบต์ลงในหน่วยความจำข้อมูลอีอีพรอม

address เป็นตัวแปรหรือค่าคงที่ ใช้กำหนดแอดเดรสของหน่วความจำอีอีพรอม มีค่าตั้งแต่ 0-2047

byte เป็นค่าข้อมูลที่ต้องการเขียน

XOUT

รูปแบบการใช้งานคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
XOUT   mpin,zpin,[house\keyOrCommand{\cycles}{,house\keyOrCommand
{\cycles}...}]
```

ใช้ส่งคำสั่งควบคุมเครื่องใช้ไฟฟ้าผ่านสายไฟฟ้าสลับตามมาตรฐาน X-10 จะต้องส่งคำสั่งผ่านอุปกรณ์เชื่อมต่อสายไฟฟ้าสลับ

mpin เป็นขาพอร์ตของเบสิกแสตมป์ 2SX ที่ใช้สำหรับส่งข้อมูลคำสั่ง เมื่อกระทำคำสั่งนี้ ขาพอร์ตจะทำงานเป็นเอาต์พุต

zpin เป็นขาพอร์ตของเบสิกแสตมป์ 2SX ใช้สำหรับรับสัญญาณจุดชนวนที่ศูนย์จากอุปกรณ์เชื่อมต่อสายไฟฟ้าสลับ เพื่อกระตุ้นเครื่องใช้ไฟฟ้าให้ทำงานที่มุมศูนย์พอดี เมื่อกระทำคำสั่งนี้ขาพอร์ตจะทำงานเป็นอินพุต

house เป็นรหัสของอุปกรณ์ไฟฟ้าที่ใช้มาตรฐานการควบคุมแบบ X-10 มีค่าตั้งแต่ 0-15 ใช้แทนตัวอักษร A ถึง P

keyOrCommand เป็นข้อมูลคำสั่ง

cycles เป็นส่งเพิ่มเติมใช้ในกรณีที่ต้องการกำหนดรอบของการส่งรหัสหรือข้อมูคำสั่ง หากไม่มีการกำหนดค่า cycles เบสิกแสตมป์ 2SX จะกำหนดให้มีค่าเท่ากับ 2

2.4 การแปลงการใช้งานโปรแกรมเป็นภาษา PBASIC เพื่อการใช้งานจริง

ในการเคลื่อนที่ของหุ่นยนต์จะทำการสั่งงานผ่านทางเบสิกแสตมป์ 2SX ซึ่งจะทำการเขียนโปรแกรมเพื่อควบคุมการทำงานจะทำการเขียนโปรแกรมโดยใช้ภาษา PBASIC ซึ่งคุณสมบัติของเบสิกแสตมป์กับภาษา PBASIC ได้กล่าวไว้ดังกล่าวข้างต้นแล้ว ซึ่งจะสรุปการเคลื่อนที่ของหุ่นยนต์ได้ดังนี้

- การเคลื่อนที่เดินหน้า
- การเคลื่อนที่ถอยหลัง
- การเคลื่อนที่ไปทางซ้าย
- การเคลื่อนที่ไปทางขวา

โดยการเคลื่อนที่ประเภทต่าง ๆ นั้นจะมีโปรแกรมการทำงานที่แตกต่างกัน โดยที่การนำไปใช้งานจริงนั้นอาจจะมีการเคลื่อนที่หลายประเภทในการเคลื่อนที่แต่ละครั้ง ดังนั้นในตัว source code หนึ่ง ๆ สามารถจะทำการกำหนดการเคลื่อนที่ที่แตกต่างกันได้

ในการออกแบบโปรแกรมจะทำการกำหนดการเคลื่อนที่แต่ละแบบเป็น module แล้วเมื่อผู้ใช้ทำการกำหนดเส้นทางที่แตกต่างกัน ก็จะทำการสร้างโปรแกรมที่สอดคล้องกับเส้นทางเหล่านั้นโดยการเปลี่ยนแปลงข้อมูลบางส่วนจาก source code ที่กำหนดไว้เริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเคลื่อนที่ของหุ่นยนต์ หุ่นยนต์จะสามารถเคลื่อนที่ได้จากเซอร์โวมอเตอร์ซึ่งเป็นจุดสำคัญในการทำให้หุ่นยนต์สามารถเคลื่อนไหวได้ โดยคุณสมบัติของเซอร์โวมอเตอร์จะมีดังนี้คือ

2.4.1 คุณสมบัติของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ (servo motor) เป็นมอเตอร์ไฟตรงแบบหนึ่งที่ทำงานด้วยสัญญาณพัลส์ มีสายต่อ 3 เส้นคือ *ไฟเลี้ยง กราวด์ และสายรับสัญญาณ* ภายในเซอร์โวมอเตอร์จะมีวงจรอีเลกทรอนิกส์บรรจุอยู่เพื่อควบคุมการทำงาน การควบคุมความเร็วจะมาจากลักษณะของสัญญาณพัลส์ที่ป้อนเข้าไปยังสายรับสัญญาณ

แกนหมุนของเซอร์โวมอเตอร์จะไม่สามารถหมุนได้รอบตัวครบ 360 องศาเหมือนกับมอเตอร์ไฟตรง เนื่องจากข้อจำกัดทางด้านกลไกภายใน ซึ่งโดยปกติแล้วเซอร์โวมอเตอร์สามารถหมุนได้ประมาณ 180 องศา

การขับให้เซอร์โวมอเตอร์หมุนจะต้องใช้ขบวนของสัญญาณพัลส์ที่มีคาบเวลารวมประมาณ 12 มิลลิวินาที โดยต้องมีช่วงเวลาที่เป็นพัลส์บวก 1-2 มิลลิวินาที และเป็นช่วงพัลส์ลบประมาณ 10 มิลลิวินาที โดยถ้าหากพัลส์บวกมีความกว้าง 1 มิลลิวินาที แกนหมุนของเซอร์โวมอเตอร์จะหมุนไปทางซ้าย ในขณะที่ถ้าหากพัลส์บวกมีความกว้าง 1.5 มิลลิวินาที แกนหมุนของมอเตอร์จะอยู่ตำแหน่งกึ่งกลาง และสุดท้ายหากพัลส์บวกมีความกว้าง 2 มิลลิวินาที มอเตอร์จะอยู่ในตำแหน่งขวา

2.4.2 การขับเซอร์โวมอเตอร์ด้วยภาษา PBASIC

การขับเซอร์โวมอเตอร์สามารถทำได้โดยการสร้างโปรแกรมไปทำการควบคุมการหมุนของเซอร์โวมอเตอร์ สำหรับเบสิกแสตมป์ 2SX สามารถทำได้โดยต่อขาพอร์ตของเบสิกแสตมป์ 2SX เข้ากับสายสัญญาณของเซอร์โวมอเตอร์ แล้วทำการเขียนโปรแกรมควบคุมในการส่งพัลส์เพื่อควบคุมการหมุนของเซอร์โวมอเตอร์ โดยใช้คำสั่ง PULSOUT โดยพัลส์ที่เกิดขึ้นจากคำสั่งนี้จะมีความกว้างในหน่วย 0.8 ไมโครวินาที

2.4.3 รูปแบบคำสั่งของการเขียนโปรแกรมควบคุมเซอร์โวมอเตอร์

2.4.3.1 กรณีการเดินหน้า และถอยหลัง

ในกรณีต้องการให้หุ่นยนต์เดินหน้า ต้องส่งสัญญาณไปยังเซอร์โวมอเตอร์ซ้ายให้หมุนในลักษณะทวนเข็มนาฬิกา ส่วนมอเตอร์ขวาให้หมุนตามเข็มนาฬิกา

ในกรณีต้องการให้หุ่นยนต์ถอยหลัง ต้องส่งสัญญาณไปยังทิศทางตรงกันข้ามกับการเดินหน้า นั่นคือมอเตอร์ซ้ายสั่งให้หมุนตามเข็มนาฬิกา ส่วนมอเตอร์ขวาหมุนทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเขียนเป็นโปรแกรมได้ดังนี้

```
'Forward And Backward movement routine
LEFT_SERVO      CON 15
RIGHT_SERVO     CON 3
LOOP            VAR WORD

FORWARD:  FOR LOOP=1 TO 100
                PULSOUT LEFT_SERVO,1875+SPEED
                PULSOUT RIGHT_SERVO,1875-SPEED
          PAUSE 20
          NEXT
          PAUSE 1000

BACKWARD: FOR LOOP=1 TO 100
                PULSOUT LEFT_SERVO,1875-SPEED
                PULSOUT RIGHT_SERVO,1875+SPEED
                PAUSE 20
          NEXT
          PAUSE 1000
```

ซึ่งผู้ที่ทำการสร้างโปรแกรมจะสามารถคำนวณการระยะทางการเคลื่อนที่ หรือระยะเวลาที่หุ่นยนต์ใช้ในการเคลื่อนที่ได้ โดยขั้นตอนของโปรแกรมจะทำการสร้างขบวนของสัญญาณพัลส์จำนวน 100 ลูกสำหรับขับเซอร์โวมอเตอร์ในการขับเซอร์โวมอเตอร์ไปข้างหน้า และจะทำการสร้างสัญญาณพัลส์ 100 ลูกในการขับเซอร์โวมอเตอร์ไปข้างหลัง

การคำนวณหาพัลส์เวลาที่ใช้ในการขับเซอร์โวมอเตอร์จะสามารถทำการคำนวณได้ จากคำสั่ง

```
PULSOUT LEFT_SERVO,SPEED
```

ซึ่งค่า SPEED จะบอกถึงระยะเวลาที่จะให้เซอร์โวมอเตอร์เคลื่อนที่ โดยค่า SPEED จะมีหน่วยเป็น 0.8 ไมโครวินาที ซึ่งสามารถทำเป็นหน่วยมิลลิวินาทีได้จากการคำนวณดังนี้

$$SPEED * 0.8 * 10^{-6}$$

สำหรับค่า 1875 เป็นค่าที่ทำให้มอเตอร์หยุดหมุน ดังนั้นหากต้องการให้มอเตอร์หมุนตามเข็มนาฬิกา ต้องกำหนดให้เป็นค่าน้อยกว่า 1875 และถ้าต้องการให้หมุนทวนเข็มนาฬิกา ต้องกำหนดให้ค่ามากกว่า 1875

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมจะทำการคำนวณทั้งหมด 100 รอบ จึงทำให้หน่วยที่ได้จากการคำนวณเป็นวินาที (t = SPEED * 0.8 * 10^{-3}) และจากการวัดรัศมีของวงล้อของหุ่นยนต์ ได้ข้อมูลจากการวัดออกมาคือ 2.8 เซนติเมตร เพราะฉะนั้นในหนึ่งรอบการหมุนของเซอร์โวมอเตอร์ จะเคลื่อนที่ไปได้เป็นระยะทาง

```
2*Pi* R     = 2 * 3.1428 * 0.028
            = 0.1759968
```

เพราะฉะนั้น ในหนึ่งรอบการหมุนของเซอร์โวมอเตอร์จะเคลื่อนที่ได้ระยะทาง 0.1759968 เมตร หรือประมาณ 17.6 เซนติเมตร

สมมติให้อัตราเร็วในการเคลื่อนที่ของหุ่นยนต์เท่ากับ u m/s

ดังนั้นจากสูตรของการเคลื่อนที่ในแนวราบคือ

$$s = ut$$

จึงสรุปได้ว่า เวลาในการเคลื่อนที่ 1 เมตร คือ

$$t = s/u$$

$$t = 1/u$$

ดังนั้นจึงต้องทำให้มอเตอร์หมุนไปทั้งสิ้นเป็นเวลา t เพื่อให้เคลื่อนที่ 1 เมตร จะต้องใช้ SPEED ในการโปรแกรมข้อมูลเป็น

$$speed = t/(0.8 * 10^{-3})$$

$$speed = 1/(u*(0.8 * 10^{-3}))$$

$$speed = 1250/u$$

และทำการแทนค่าลงไปในโปรแกรม ก็จะได้ระยะทางในการเคลื่อนที่ 1 เมตร

2.4.3.2 กรณีการเลี้ยวซ้าย และขวา

ในกรณีต้องการให้หุ่นยนต์เลี้ยวซ้ายหรือขวา ทำได้โดยกำหนดให้มอเตอร์ตัวหนึ่งหยุดหมุน ในกรณีที่ต้องการเลี้ยวซ้ายเดินหน้า ต้องสั่งให้มอเตอร์ซ้ายหยุด และสั่งให้มอเตอร์ขวาหมุนตามเข็มนาฬิกา

ในกรณีต้องการให้หุ่นยนต์เลี้ยวซ้ายถอยหลัง ต้องสั่งให้มอเตอร์ซ้ายหยุด และสั่งให้มอเตอร์ขวาหมุนทวนเข็มนาฬิกา

ในกรณีต้องการให้หุ่นยนต์เลี้ยวขวาเดินหน้า ต้องสั่งให้มอเตอร์ขวาหยุด และสั่งให้มอเตอร์ซ้ายหมุนทวนเข็มนาฬิกา

ในกรณีต้องการให้หุ่นยนต์เลี้ยวขวาถอยหลัง ต้องสั่งให้มอเตอร์ซ้ายหยุด และสั่งให้มอเตอร์ขวาหมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเขียนเป็นโปรแกรมได้ดังนี้

```
LEFT_SERVO      CON 15
RIGHT_SERVO     CON 3
LOOP            VAR WORD

LEFT_FW:  FOR LOOP=1 TO 100
                PULSOUT RIGHT_SERVO,1875-SPEED
                PAUSE 20
          NEXT
          PAUSE 1000

LEFT_BW:  FOR LOOP=1 TO 100
                PULSOUT RIGHT_SERVO,1875+SPEED
                PAUSE 20
          NEXT
          PAUSE 1000

RIGHT_FW: FOR LOOP=1 TO 100
                PULSOUT LEFT_SERVO,1875+SPEED
                PAUSE 20
          NEXT
          PAUSE 1000

RIGHT_BW: FOR LOOP=1 TO 100
                PULSOUT LEFTT_SERVO,1875-SPEED
                PAUSE 20
          NEXT
          PAUSE 1000
```

ซึ่งระยะทางในการเลี้ยวจะขึ้นอยู่กับจำนวน LOOP และ SPEED ที่กำหนดไว้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการวิจัย

จากการวิจัยโครงงาน โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก ซึ่งมีขั้นตอนการวิจัย และการดำเนินงาน แบ่งได้เป็น 5 ส่วนดังนี้

3.1 ขั้นตอนการศึกษาโปรแกรมภาษา PBASIC เพื่อการใช้งาน โดยจะศึกษาถึงโครงสร้างของภาษา เพื่อที่จะนำมาใช้ในการบังคับหุ่นยนต์ด้วย Basic Stamp 2SX

3.2 ขั้นตอนการศึกษาลักษณะโครงสร้างของหุ่นยนต์ ROBO-KIT

3.3 ขั้นตอนการศึกษาภาคการทำงานของหุ่นยนต์ ROBO-KIT

3.4 ขั้นตอนการศึกษาลักษณะการเคลื่อนที่ของหุ่นยนต์

3.5 ขั้นตอนการศึกษาถึงความสัมพันธ์ระหว่างการเคลื่อนที่ของหุ่นยนต์ กับการสร้างโปรแกรมควบคุมด้วยภาษา PBASIC

3.6 ขั้นตอนการศึกษา และทำการออกแบบส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้ ด้วย Visual Basic โดยจะให้ผู้ใช้สามารถทำการกำหนดเส้นทางการเคลื่อนที่ของหุ่นยนต์ได้

3.7 ขั้นตอนควบคุมให้โปรแกรม Visual Basic ทำการการสร้างไฟล์ที่ทำหน้าที่เป็น Source Code ของภาษา PBASIC เพื่อนำไปใช้งานกับหุ่นยนต์ได้จริง

3.1 ขั้นตอนการศึกษาโปรแกรมภาษา PBASIC

ในขั้นตอนนี้จะทำการศึกษาถึงโครงสร้างของภาษา PBASIC โดยจะทำการศึกษาถึงลักษณะและรูปแบบของการเขียนโปรแกรม ในการศึกษาจะเริ่มทำการศึกษาจากคำสั่งต่าง ๆ ที่ใช้ในการควบคุมการทำงาน รวมถึงจะทำการศึกษาถึงรูปแบบการใช้งานคำสั่งต่าง ๆ เหล่านั้น เพื่อที่จะสามารถนำมาใช้งานจริงได้ โดยเมื่อทำการศึกษาภาษา PBASIC เสร็จสิ้นแล้ว จะทำการศึกษาถึงรูปแบบการใช้งาน เพื่อที่จะนำเอาภาษา PBASIC ที่ทำการศึกษานั้นมาใช้บังคับหุ่นยนต์ได้จริง

หมายเหตุ : สำหรับโครงสร้าง และรูปแบบของภาษา PBASIC นั้นได้ทำการแสดงไว้ในบทที่ 2 แล้ว

3.2 ขั้นตอนการศึกษาลักษณะโครงสร้างของหุ่นยนต์ ROBO-KIT

ในขั้นตอนนี้จะทำการศึกษาถึงโครงสร้าง และส่วนประกอบของหุ่นยนต์ ROBO-KIT จะทำการศึกษาจากคู่มือที่มาพร้อมกับตัว ROBO-KIT และจะทำการศึกษาถึงส่วนประกอบแต่ละชิ้นส่วนโดยละเอียด ซึ่งจากการศึกษา สามารถสรุปโครงสร้างของหุ่นยนต์ ROBO-KIT ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 **คุณลักษณะของหุ่นยนต์** ROBO-KIT

- ควบคุมด้วยไมโครคอนโทรเลอร์ BASICSTAMP 2SX
- ขนาดหน่วยความจำสำหรับเก็บข้อมูล 16 กิโลไบต์ แบ่งเป็น 8 ช่วง ช่วงละ 2 กิโลไบต์
- เชื่อมต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมเพื่อรับข้อมูลควบคุมการทำงาน และแสดงผลการทำงาน
- ขับเคลื่อนด้วยเซอร์โวมอเตอร์ขนาด 6 V
- ใช้ไฟเลี้ยง 2 ชุดแยกกันระหว่างแผงวงจรควบคุม (+6V) และเซอร์โวมอเตอร์ (+6V)
- มีแผงต่อวงจรขนาด 2.5x3.5 นิ้ว จึงทำให้สามารถสร้างวงจรเพื่อเชื่อมต่อกับอุปกรณ์ตรวจจับได้หลายชิ้น
- มีแผงวงจรตรวจจับวัตถุแบบใช้แสงทำให้หุ่นยนต์สามารถเดินตามเส้น หรือเคลื่อนที่หลบหลีกวัตถุ ตลอดจนเคลื่อนที่ในพื้นที่ที่กำหนด

และหลังจากการศึกษาโครงสร้างของหุ่นยนต์ สามารถที่จะจำแนกชิ้นส่วนหลัก ๆ ของหุ่นยนต์ได้ดังนี้

3.2.2 **ชิ้นส่วนประกอบของหุ่นยนต์** ROBO-KIT

- แผงวงจรหลัก ROBOT-KIT พร้อมแผงต่อวงจรขนาด 2.5x3.5 นิ้ว
- โครงตัวถังซึ่งทำด้วยพลาสติก
- ล้อข้างพร้อมด้วยยางหัวแหวน 2 ชุด
- เซอร์โวมอเตอร์ 6V 2 ชุด
- กระบะถ่านขนาด AA 4 ก้อนพร้อมสายต่อ
- กระบะถ่านขนาด AA 4 ก้อนพร้อมปลั๊ก
- แผงวงจรตรวจจับวัตถุ 2 ชุด
- สายดาวน์โหลด
- สกรูพร้อมเสารองขนาดต่าง ๆ

3.3 ขั้นตอนการศึกษาภาคการทำงานของหุ่นยนต์ ROBO-KIT

ในการทำงานของหุ่นยนต์ ROBO-KIT สามารถแบ่งโครงสร้างการทำงานออกได้เป็นส่วนย่อย ๆ ต่าง ๆ ได้ดังนี้คือ

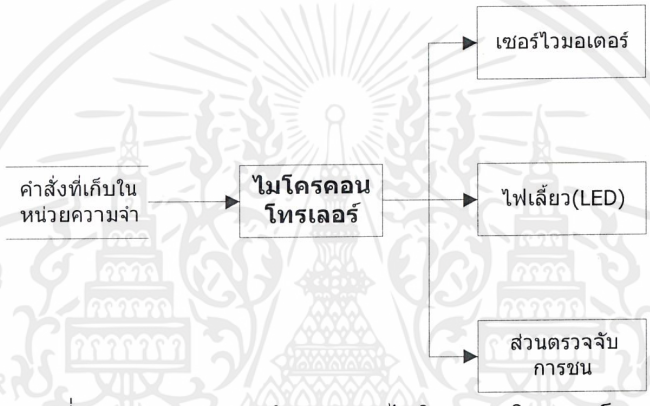
- ส่วนการทำงานของไมโครคอนโทรเลอร์
- ส่วนการทำงานของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนการทำงานของไฟเลี้ยว
- ส่วนการทำงานของตัวตรวจจับแสง

3.3.1 ส่วนการทำงานของไมโครคอนโทรเลอร์

ไมโครคอนโทรเลอร์ถือว่าเป็นหัวใจหลักของหุ่นยนต์ ROBO-KIT โดยจะทำหน้าที่ประมวลผลคำสั่งต่าง ๆ ที่ถูกบันทึกอยู่ในหน่วยความจำ และทำให้หุ่นยนต์สามารถเคลื่อนที่ได้ตามคำสั่งเหล่านั้น ไมโครคอนโทรเลอร์มีลักษณะการทำงานที่สำคัญคือ จะนำเอาคำสั่งที่เก็บอยู่ในหน่วยความจำมาทำการแปลงเป็นสัญญาณไฟฟ้า และทำการส่งสัญญาณไฟฟ้าเหล่านั้นไปยังส่วนต่าง ๆ ตามพอร์ตที่กำหนดไว้ ทำให้หุ่นยนต์สามารถที่จะเคลื่อนที่หรือทำงานต่าง ๆ ได้



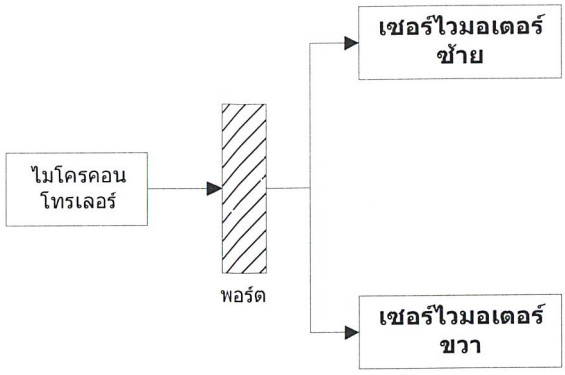
รูปที่ 3.1 รูปแสดงการทำงานของไมโครคอนโทรเลอร์

3.3.2 ส่วนการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์เป็นส่วนที่ทำให้หุ่นยนต์สามารถที่จะเคลื่อนที่ได้ ไม่ว่าจะเป็นการเคลื่อนที่เป็นเส้นตรง หรือจะเคลื่อนที่แบบหมุนเป็นองศาต่าง ๆ ก็ตาม ซึ่งลักษณะของการเคลื่อนที่ต่าง ๆ ของหุ่นยนต์จะอธิบายอยู่ในส่วนที่ 3.4

การทำงานของเซอร์โวมอเตอร์จะทำงานโดยรับเอาข้อมูลมาจากไมโครคอนโทรเลอร์ผ่านทาง Port ที่กำหนดไว้ สิ่งที่ไมโครคอนโทรเลอร์ส่งไปยังเซอร์ไวมอเตอร์จะเป็นสัญญาณไฟฟ้า ซึ่งรอบการหมุนของเซอร์โวมอเตอร์จะขึ้นอยู่กับแรงดันไฟฟ้าที่รับเข้ามา

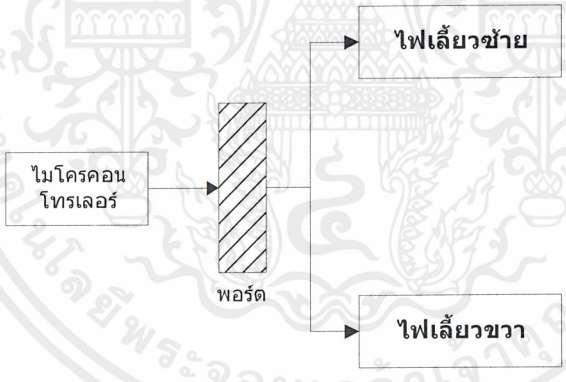
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 รูปแสดงการทำงานของเซอร์โวมอเตอร์

3.3.3 ส่วนการทำงานของไฟเลี้ยว

ไฟเลี้ยวจะมีลักษณะการทำงานคือ เมื่อหุ่นยนต์มีการเลี้ยวซ้าย หรือเลี้ยวขวา ไมโครคอนโทรเลอร์จะทำการส่งสัญญาณไปยังไฟเลี้ยวแต่ละด้านที่หุ่นยนต์เลี้ยว ทำให้ไฟเลี้ยว (LED) ด้านนั้น ๆ มีติดไฟ

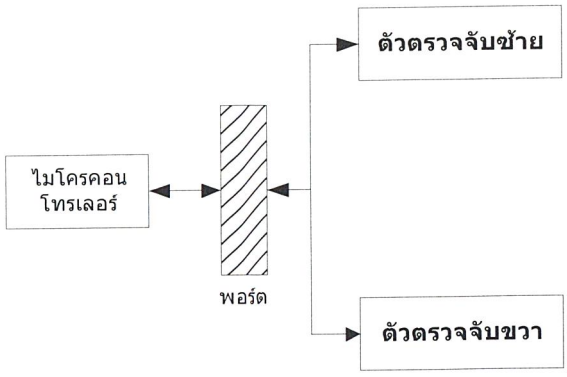


รูปที่ 3.3 รูปแสดงการทำงานของไฟเลี้ยว (LED)

3.3.4 ส่วนการทำงานของตัวตรวจจับแสง

การทำงานของตัวตรวจจับแสงจะมีลักษณะคล้ายกับเซอร์โวมอเตอร์ และไฟเลี้ยวคือรอรับสัญญาณจากไมโครคอนโทรเลอร์ แต่จะมีข้อแตกต่างคือจะมีการส่งสัญญาณกลับไปยังไมโครคอนโทรเลอร์ด้วย เมื่อมีการตรวจจับการชนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 รูปแสดงการทำงานของตัวตรวจจับการชน

3.4 ขั้นตอนการศึกษาลักษณะการเคลื่อนที่ของหุ่นยนต์

ในการศึกษาถึงลักษณะของการเคลื่อนที่ของหุ่นยนต์ จะสามารถจำแนกลักษณะการเคลื่อนที่หลัก ๆ ของหุ่นยนต์ ได้ดังนี้

- กรณีเดินหน้า และถอยหลัง
- กรณีเลี้ยวซ้าย และขวา
- กรณีหมุนซ้าย และขวา

ซึ่งการเคลื่อนที่แต่ละกรณี มีรายละเอียดดังนี้

3.4.1 กรณีเดินหน้า และถอยหลัง

ในกรณีที่ต้องการให้หุ่นยนต์เดินหน้า ต้องส่งสัญญาณไปยังเซอร์โวมอเตอร์ซ้ายให้หมุนในลักษณะทวนเข็มนาฬิกา ส่วนมอเตอร์ขวาให้หมุนตามเข็มนาฬิกา

ในกรณีถอยหลัง ต้องส่งสัญญาณให้มอเตอร์หมุนไปในทิศทางตรงข้ามกับการเดินหน้า คือมอเตอร์ซ้ายสั่งให้หมุนตามเข็มนาฬิกา ส่วนมอเตอร์ขวาหมุนทวนเข็มนาฬิกา

3.4.2 กรณีเลี้ยวซ้าย และขวา

ในกรณีที่ต้องการให้หุ่นยนต์เลี้ยวซ้าย หรือขวา ทำได้โดยกำหนดให้มอเตอร์ตัวหนึ่งหยุดหมุน ในกรณีที่ต้องการเลี้ยวซ้ายเดินหน้า ต้องสั่งให้มอเตอร์ซ้ายหยุด และสั่งให้มอเตอร์ขวาหมุนตามเข็มนาฬิกา

ในกรณีที่ต้องการให้หุ่นยนต์เลี้ยวซ้ายถอยหลัง ต้องสั่งให้มอเตอร์ซ้ายหยุด และมอเตอร์ขวาหมุนทวนเข็มนาฬิกา

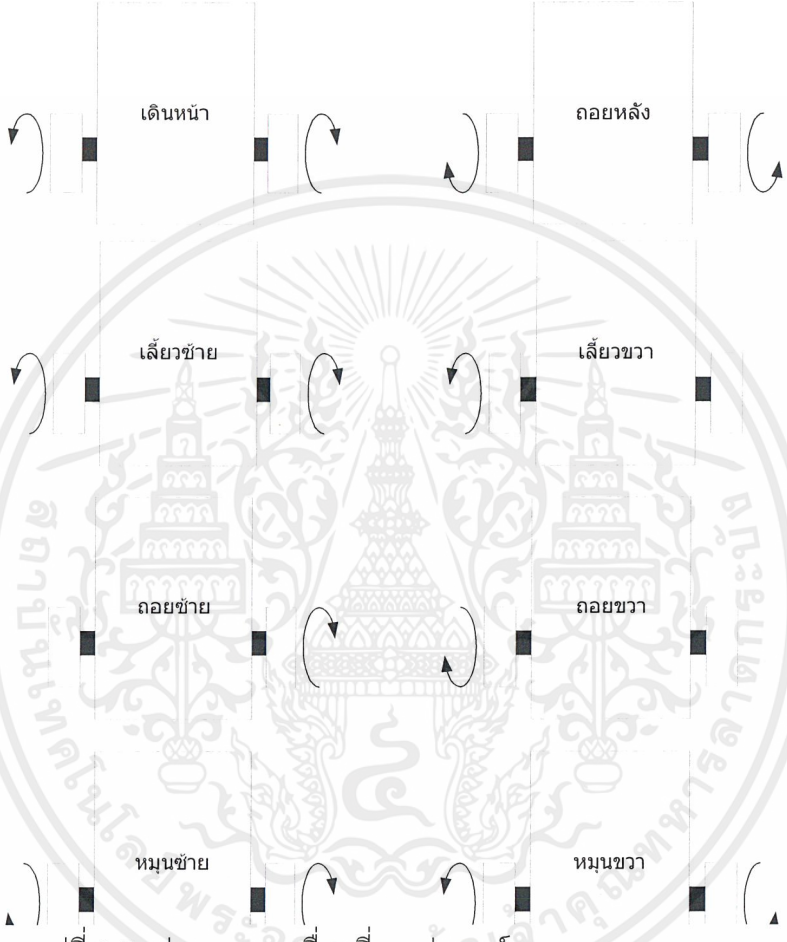
ในกรณีที่ต้องการให้หุ่นยนต์เลี้ยวขวาเดินหน้า ต้องสั่งให้มอเตอร์ขวาหยุด และสั่งให้มอเตอร์ซ้ายหมุนทวนเข็มนาฬิกา

ในกรณีที่ต้องการให้หุ่นยนต์เลี้ยวขวาถอยหลัง ต้องสั่งให้มอเตอร์ซ้ายหยุด และสั่งให้มอเตอร์ขวาหมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 กรณีหมุนซ้าย และขวา

ในกรณีที่ต้องการให้หุ่นยนต์หมุนซ้าย หรือขวาทำได้โดยกำหนดให้มอเตอร์ซ้ายหมุนถอยหลัง และมอเตอร์ขวาหมุนเดินหน้า จะเป็นการทำให้หุ่นยนต์หมุนตัวไปทางซ้ายในกรณีที่ต้องการให้หุ่นยนต์หมุนขวาต้องทำให้มอเตอร์ซ้ายหมุนเดินหน้า และมอเตอร์ขวาหมุนถอยหลัง



รูปที่ 3.5 รูปแสดงการเคลื่อนที่ของหุ่นยนต์

3.5 ขั้นตอนการศึกษาถึงความสัมพันธ์ระหว่างการเคลื่อนที่ของหุ่นยนต์ กับการสร้างโปรแกรมควบคุมด้วยภาษา PBASIC

ในขั้นตอนนี้จะทำการศึกษาถึงคำสั่งต่าง ๆ ที่ทำให้หุ่นยนต์เคลื่อนที่ไปตามเส้นทางที่ต้องการได้ โดยจะเป็นการกำหนดเส้นทางการเคลื่อนที่ของหุ่นยนต์ผ่านทาง Source Code ของภาษา PBASIC เช่นในกรณีที่ต้องการกำหนดให้หุ่นยนต์เดินหน้า และถอยหลังสามารถเขียนเป็นโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'Forward And Backward movement routine
LEFT_SERVO      CON 15
RIGHT_SERVO     CON 3
LOOP            VAR WORD
SPEED           VAR 250

FORWARD:  FOR LOOP=1 TO 100
                PULSOUT LEFT_SERVO,1875+SPEED
                PULSOUT RIGHT_SERVO,1875-SPEED
          PAUSE 20
          NEXT
          PAUSE 1000

BACKWARD: FOR LOOP=1 TO 100
                PULSOUT LEFT_SERVO,1875-SPEED
                PULSOUT RIGHT_SERVO,1875+SPEED
                PAUSE 20
          NEXT
          PAUSE 1000
```

ตัวอย่างที่ 3.1 ตัวอย่างการเขียนโปรแกรมควบคุมให้หุ่นยนต์เดินหน้า และถอยหลัง

ซึ่งเมื่อผู้ใช้ทำการสร้างโปรแกรมดังตัวอย่าง และทำการบันทึกโปรแกรมลงไปใน Microcontroller เบสิกแสตมป์ 2SX ผ่านทางโปรแกรม BASICSTAMP EDITOR แล้ว จะสามารถทำให้หุ่นยนต์เคลื่อนที่เดินหน้า และถอยหลังได้

3.6 ขั้นตอนการศึกษา และทำการออกแบบส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้ ด้วย Visual Basic

ในขั้นตอนนี้จะทำการศึกษาการทำงานของโปรแกรม Visual Basic โดยที่จะศึกษาถึงส่วนที่จะนำมาใช้ในการสร้างส่วนติดต่อกับผู้ใช้ (User Interface) โดยจะเน้นการทำงานทางด้านกราฟฟิก เพื่อที่จะให้ผู้ใช้สามารถที่จะสร้างเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยผ่านรูป หรือสัญลักษณ์รูปภาพได้ ซึ่งเมื่อผู้ใช้ทำการกำหนดเส้นทางเสร็จสิ้นแล้ว ผู้ใช้จะสามารถแปลงรูปแบบการเคลื่อนที่ที่สร้างเป็นรูปแบบ Source Code ภาษา PBASIC ได้

หลังจากทำการศึกษาการทำงานของโปรแกรม Visual Basic แล้ว จะทำการออกแบบส่วนที่ผู้ใช้สามารถใช้งานได้จริง และนำมาออกแบบเป็นต้นแบบ (Prototype) เพื่อเป็นแนวทางในการสร้างโปรแกรมต่อไปได้

3.7 ขั้นตอนการกำหนดโครงสร้างของไฟล์ที่ใช้สำหรับใช้งานร่วมกับการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กจะมีการใช้งานไฟล์ร่วมกับโปรแกรมเพื่อใช้ในการบันทึกข้อมูลต่าง ๆ โดยมีไฟล์ที่ใช้งานทั้งหมด 4 ประเภทคือ

3.7.1 ไฟล์ Source Code ที่ได้จากการ Compile

เป็นไฟล์ที่ได้จากการบันทึกไฟล์เมื่อผู้ใช้ทำการ Compile เส้นทางการเคลื่อนที่ของหุ่นยนต์เรียบร้อยแล้ว โดยไฟล์ที่ได้จะมีโครงสร้างของภาษา PBASIC เป็นหลักซึ่งจะมีนามสกุลของไฟล์เป็น .BSX ผู้ใช้สามารถนำเอาไฟล์นี้ไปทำการบันทึกลงไมโครคอนโทรเลอร์ ผ่านโปรแกรมเบสิกแสตมป์ได้โดยตรง

3.7.2 ไฟล์ที่ใช้สำหรับเก็บข้อมูลเส้นทางการเคลื่อนที่ของหุ่นยนต์

เป็นไฟล์ที่ใช้สำหรับเก็บข้อมูลของเส้นทางการเคลื่อนที่ต่าง ๆ ของหุ่นยนต์เมื่อผู้ใช้ทำการบันทึกเส้นทางการเคลื่อนที่ โดยจะมีโครงสร้างของไฟล์เป็นประเภท Text File และมีนามสกุลของไฟล์ (File Type) เป็น .SRC ย่อมาจาก Small Robot Control program ซึ่งในแต่ละส่วนจะแยกกออกจากกันด้วยสัญลักษณ์คือ “**********” ส่วนประกอบของไฟล์จะมีดังนี้คือ

3.7.2.1 Header File

เป็นส่วนที่ใช้สำหรับเก็บคุณสมบัติการเคลื่อนที่ของหุ่นยนต์ซึ่งได้แก่

- จำนวนเส้นทางทั้งหมด
- การตรวจสอบการชน
- เปิดไฟเลี้ยวซ้าย
- เปิดไฟเลี้ยวขวา
- วนซ้ำเส้นทาง

ซึ่งจำนวนเส้นทางทั้งหมด จะทำการเก็บเป็นตัวเลขจำนวนเต็ม ซึ่งจะใช้ในการตรวจสอบความผิดพลาดที่อาจเกิดจากการอ่านข้อมูลของไฟล์ ส่วนที่เหลือจะทำการเก็บเป็นค่า “True” หรือ “False” เพื่อเป็นการบอกว่าจะต้องการให้ตรวจสอบในส่วนต่าง ๆ หรือไม่

3.7.2.2 Path เส้นทาง

เป็นส่วนที่ใช้สำหรับเก็บรายละเอียดเส้นทางการเคลื่อนที่ของหุ่นยนต์ เป็นส่วนประกอบหลักของไฟล์นี้ โดยการเก็บเส้นทางจะใช้สัญลักษณ์เป็นตัวอักษรภาษาอังกฤษแทนประเภทการเคลื่อนที่กับค่ากำกับการเคลื่อนที่ของการเคลื่อนที่แต่ละครั้ง รายละเอียดของสัญลักษณ์ต่าง ๆ มีดังนี้คือ

- a หมายถึงการเคลื่อนที่ไปข้างหน้า (เมตร)
- b หมายถึงการเคลื่อนที่ไปข้างหลัง (เมตร)

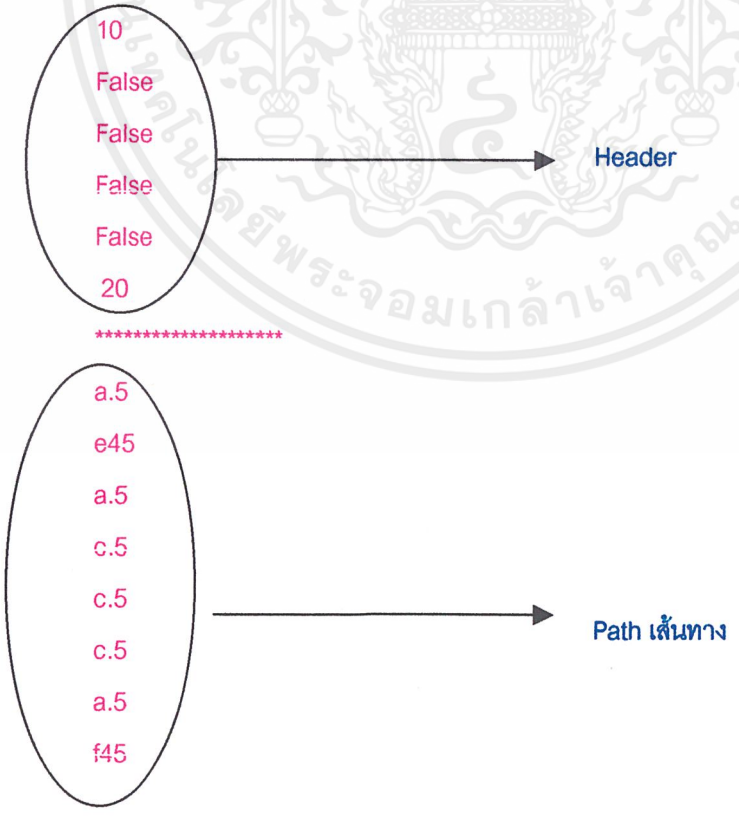
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- c หมายถึงการเคลื่อนที่ไปทางซ้าย (เมตร)
- d หมายถึงการเคลื่อนที่ไปทางขวา (เมตร)
- e หมายถึงการเคลื่อนที่แบบหมุนซ้าย (องศา)
- f หมายถึงการเคลื่อนที่แบบหมุนขวา (องศา)
- g หมายถึงการหยุดการเคลื่อนที่ (วินาที)

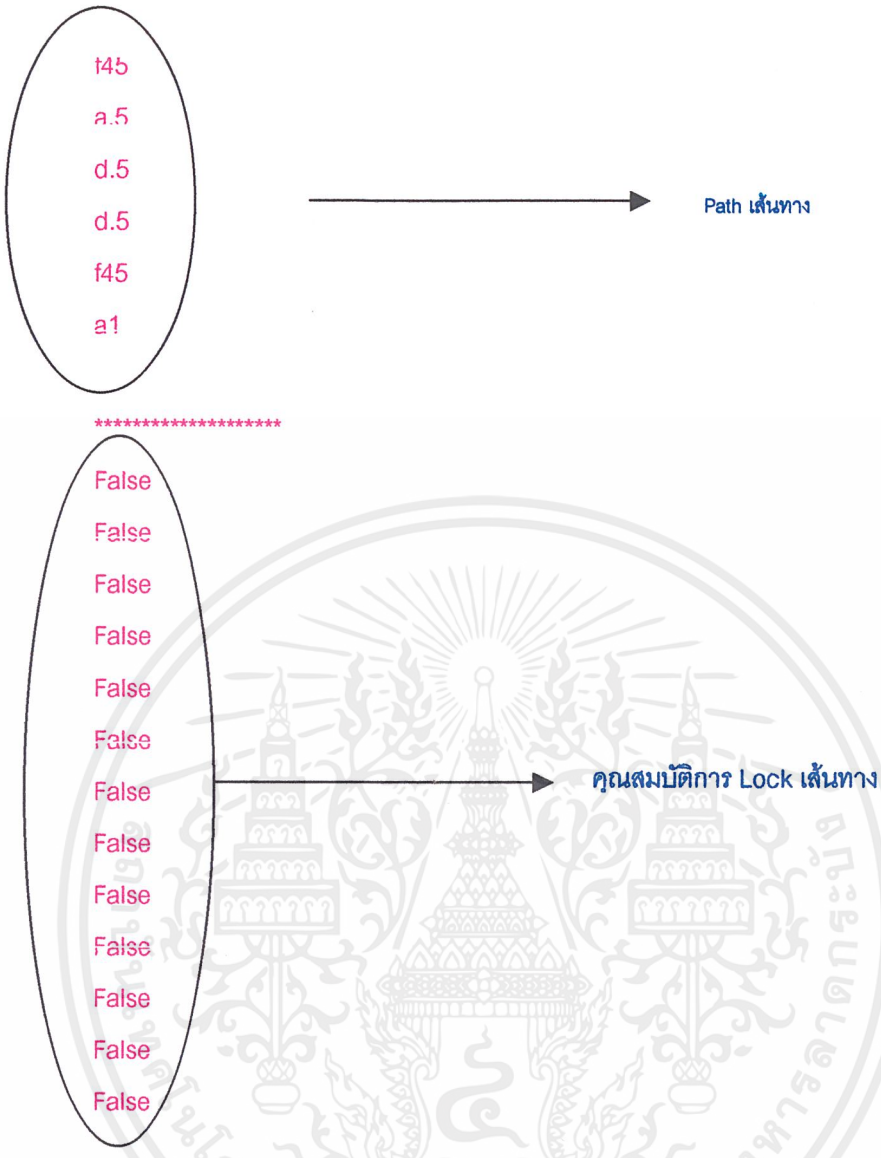
3.7.2.3 คุณสมบัติการ Lock เส้นทาง

คุณสมบัติการ Lock เส้นทางจะใช้สำหรับให้ผู้ใช้ Lock เส้นทางบางเส้นทางของหุ่นยนต์ เพื่อป้องกันข้อผิดพลาดที่เกิดจากการลบเส้นทางบางเส้นทางออกไป จะทำให้ไม่สามารถกลับมาแก้ไขได้ ในการเก็บข้อมูลการ Lock เส้นทางนั้นจะเก็บเป็นค่า “True” หรือ “False” ซึ่งถ้าค่าเป็น True จะหมายถึงเส้นทางนั้น Lock และถ้าเป็น False จะหมายถึงเส้นทางนั้นไม่ได้ Lock ในการเก็บค่าการ Lock นั้นจะต้องเก็บเท่ากับจำนวนเส้นทางทั้งหมดของหุ่นยนต์

ตัวอย่างไฟล์ SRC : Butterfly.src



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3.7.3 Log File

เป็นไฟล์ที่ใช้สำหรับเก็บประวัติการทำงานของผู้ใช้ โดยจะทำการแสดงรายละเอียดของการใช้งานโปรแกรม เพื่อที่จะสามารถบอกประวัติการทำงานของผู้ใช้ได้ และเมื่อเกิดข้อผิดพลาดของโปรแกรม จะสามารถหาข้อผิดพลาดของโปรแกรมได้ง่ายยิ่งขึ้นส่วนประกอบของ Log File จะประกอบไปด้วย วันที่ทำการใช้งาน ,เวลาที่ทำการใช้งาน ,รายละเอียดการใช้งานโปรแกรม สำหรับ Log File จะมีนามสกุลเป็น .dat โดยผู้ใช้สามารถกำหนด Path ของ Log File ได้จากส่วน Preference ของโปรแกรม

ตัวอย่างไฟล์ Log : Robot.Log

At 9/2/2545 22:05:33 : Fmain Load

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
**********************************************
At 9/2/2545 22:05:54 : Fmain Load
At 9/2/2545 22:05:57 : เดินหน้า 1 เมตร
At 9/2/2545 22:05:59 : เดินหน้า 1 เมตร
At 9/2/2545 22:06:00 : เดินหน้า 1 เมตร
At 9/2/2545 22:06:00 : เดินหน้า 1 เมตร
At 9/2/2545 22:06:01 : เดินหน้า 1 เมตร
At 9/2/2545 22:06:02 : เดินหน้า 1 เมตร

**********************************************
At 9/2/2545 22:06:07 : Fmain Load
At 9/2/2545 22:06:10 : เดินหน้า .1 เมตร
At 9/2/2545 22:06:11 : เดินหน้า .2 เมตร
At 9/2/2545 22:06:12 : เดินหน้า .3 เมตร
At 9/2/2545 22:06:13 : เดินหน้า .4 เมตร
At 9/2/2545 22:06:13 : เดินหน้า .5 เมตร
At 9/2/2545 22:06:14 : เดินหน้า .6 เมตร
At 9/2/2545 22:06:17 : เดินหน้า .7 เมตร
At 9/2/2545 22:06:18 : เดินหน้า .8 เมตร
At 9/2/2545 22:06:19 : เดินหน้า .9 เมตร
At 9/2/2545 22:06:20 : เดินหน้า 1 เมตร
At 9/2/2545 22:06:23 : เดินหน้า 2 เมตร
At 9/2/2545 22:06:42 : ลบ Record จาก History Record ที่ 7 9/2/2545 22:06:42
At 9/2/2545 22:06:46 : ลบ Record จาก History Record ที่ 6 9/2/2545 22:06:46
At 9/2/2545 22:06:50 : ลบ Record จาก History Record ที่ 6 9/2/2545 22:06:50
At 9/2/2545 22:06:56 : ลบ Record จาก History ตั้งแต่ Record ที่ 5 9/2/2545 22:06:56
At 9/2/2545 22:07:01 : Lock Record ที่ 3 9/2/2545 22:07:01
At 9/2/2545 22:07:10 : ลบ Record จาก History Record ที่ 4 9/2/2545 22:07:10
```

3.7.4 ไฟล์กำหนดค่าของโปรแกรม (Setting File)

เป็นไฟล์ที่ใช้สำหรับเก็บค่าที่ผู้ใช้กำหนดสำหรับการใช้งานโปรแกรม โดยจะประกอบไปด้วยสองส่วนค่า

- ค่า Path ที่ใช้สำหรับเก็บ Log File โดยเมื่อมีการบันทึกการใช้งานลงใน Log File จะทำการบันทึกเก็บไว้ตาม Path ที่กำหนดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User_Defined_Log_Path\Robot.Log

- ค่า Path ที่ใช้สำหรับเก็บโปรแกรม BasicStamp ผู้ใช้สามารถที่จะทำการเรียกใช้งานโปรแกรม BasicStamp ได้ผ่านทางเมนูของโปรแกรมหลัก โดยผู้ใช้จำเป็นต้องทำการกำหนด Path ของโปรแกรม BasicStamp ให้ถูกต้องก่อนจึงจะสามารถใช้งานได้

User_Defined_BasicStamp_Path\BasicStamp.exe

สำหรับการเก็บ Setting File นั้นจะทำการเก็บที่ Application_Path\Setting โดยใช้ File ที่ชื่อว่า Setting.Dat สำหรับการเก็บข้อมูล

ตัวอย่างของ Setting File

D:\MoMo\Robot New\log ----------> Log File Path

D:\MoMo\Robot New\Stamp2 ----------> BasicStamp Path

3.8 ขั้นตอนควบคุมให้โปรแกรม Visual Basic ทำการการสร้างไฟล์ที่ทำหน้าที่เป็น Source Code ของภาษา PBASIC

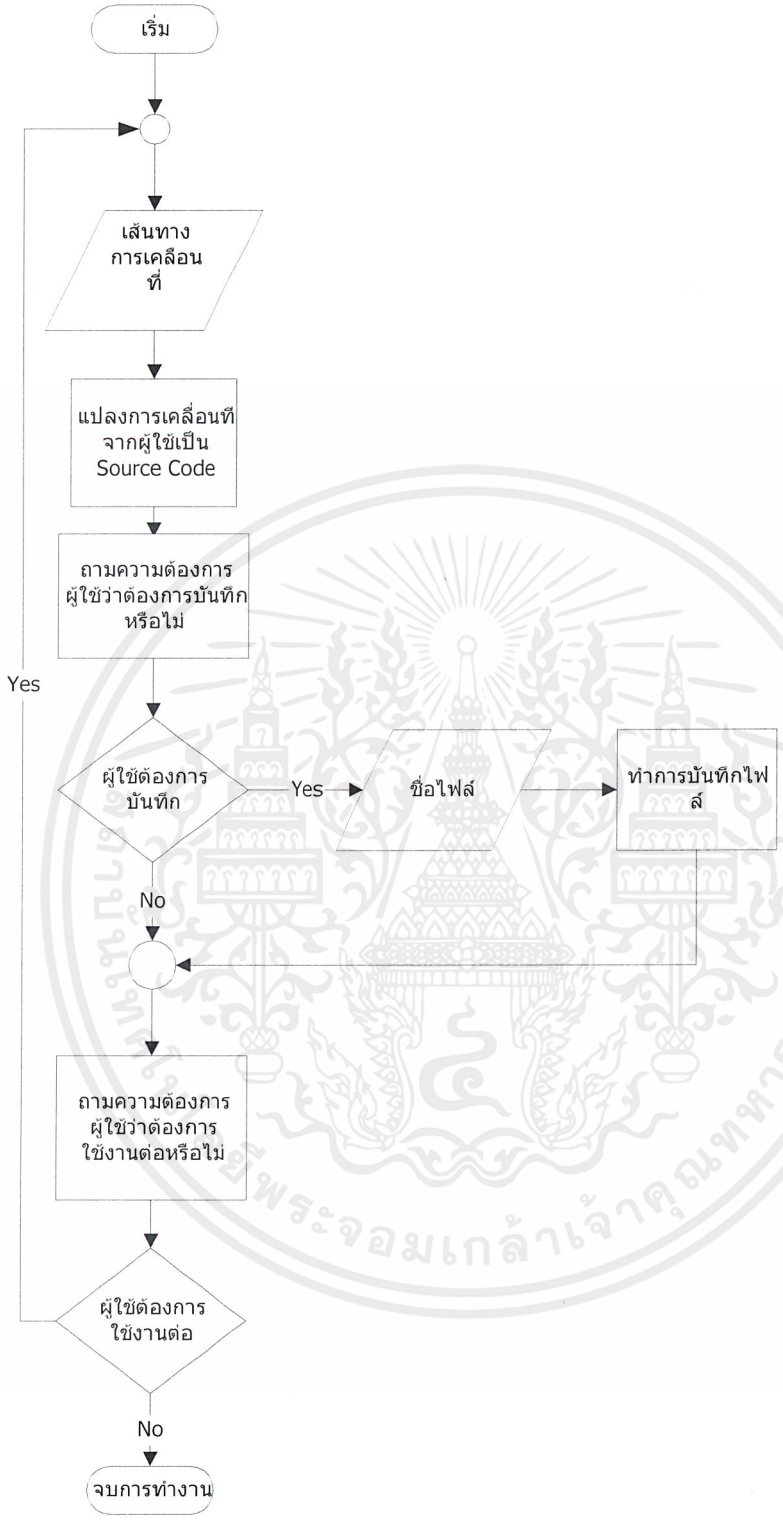
ในขั้นตอนนี้ เมื่อทำการแปลงเส้นทางการเคลื่อนที่ของหุ่นยนต์เป็น Source Code ได้เรียบร้อยแล้ว จะทำการบันทึก Source Code ของเส้นทางนั้นเป็น File ใช้งาน เพื่อที่จะสามารถนำมาใช้อีกภายหลังได้ โดยที่ไม่จำเป็นต้องทำการสร้างเส้นทางการเคลื่อนที่ของหุ่นยนต์ใหม่

โดยในการบันทึกเป็น File นั้นจะต้องบันทึกเป็น File ที่มีนามสกุลเป็น .BSX และลักษณะภายในของ File จะเป็น Text File โดยจะมีข้อมูลเป็น Source Code ภาษา PBASIC

โดยสรุปแล้วในการรับข้อมูลเข้า และการแปลงเป็น File ข้อมูลนั้น จะมีขั้นตอนการทำงานดังนี้คือ

1. ขั้นตอนการรับข้อมูลเข้าโดยผู้ใช้
2. ขั้นตอนการแสดงเส้นทางการเดินของหุ่นยนต์โดยโปรแกรม
3. ขั้นตอนการปรับเปลี่ยนคุณสมบัติของโปรแกรม
4. ขั้นตอนการปรับเปลี่ยนคุณสมบัติของการเคลื่อนที่ของหุ่นยนต์
5. ขั้นตอนการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของหุ่นยนต์
6. ขั้นตอนการ Compile เส้นทางการเคลื่อนที่ของหุ่นยนต์เป็น Source Code
7. ขั้นตอนการบันทึก Source Code ลงไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

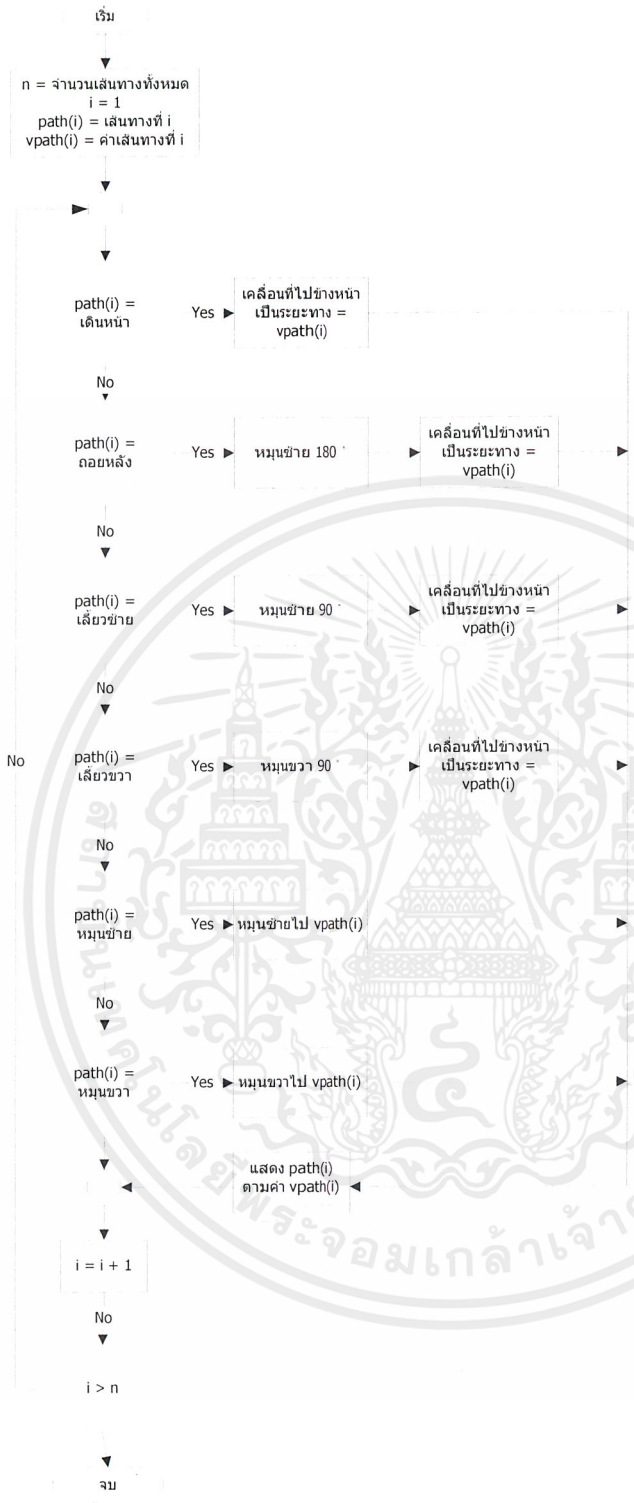


รูปที่ 3.6 Flowchart แสดงขั้นตอนการทำงานของโปรแกรมโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

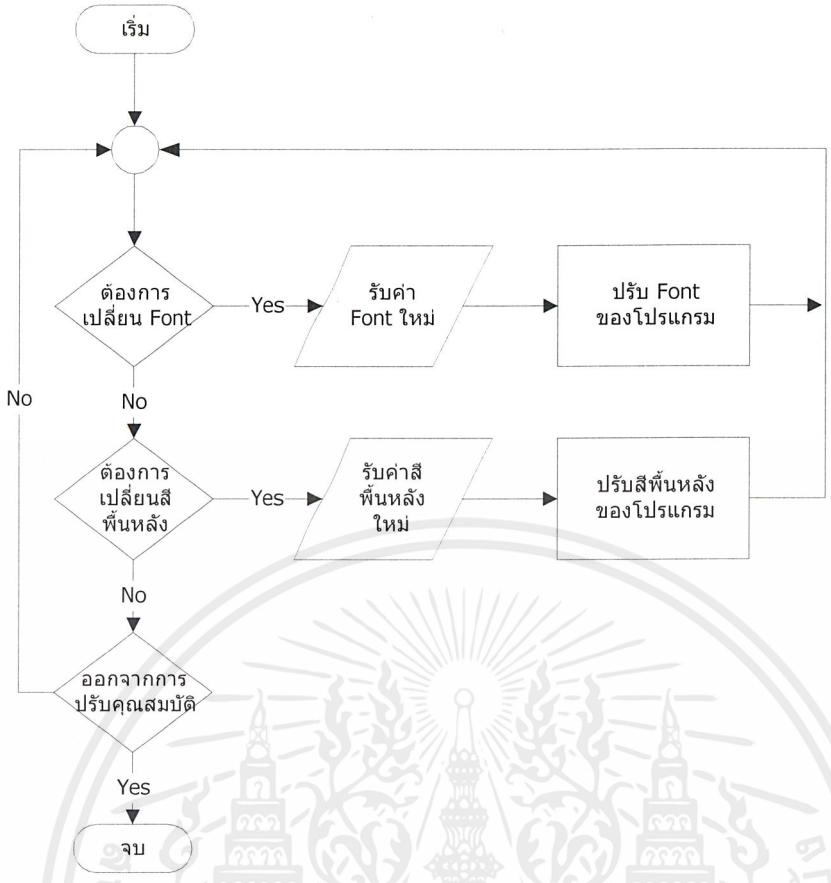
รูปที่ 3.7 Flowchart แสดงขั้นตอนการรับข้อมูลเข้าโดยผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



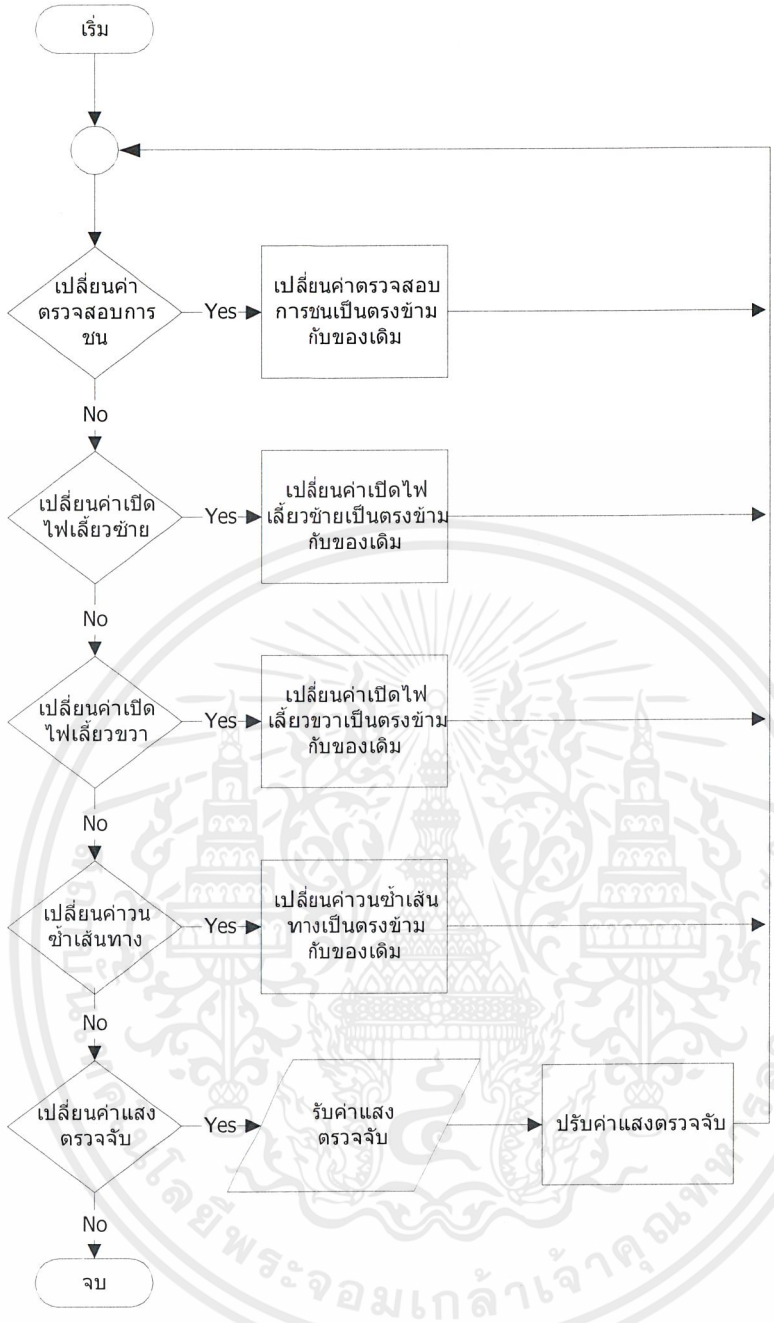
รูปที่ 3.8 Flowchart แสดงขั้นตอนการแสดงเส้นทางการเดินของหุ่นยนต์โดยโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 Flowchart แสดงขั้นตอนการปรับเปลี่ยนคุณสมบัติของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

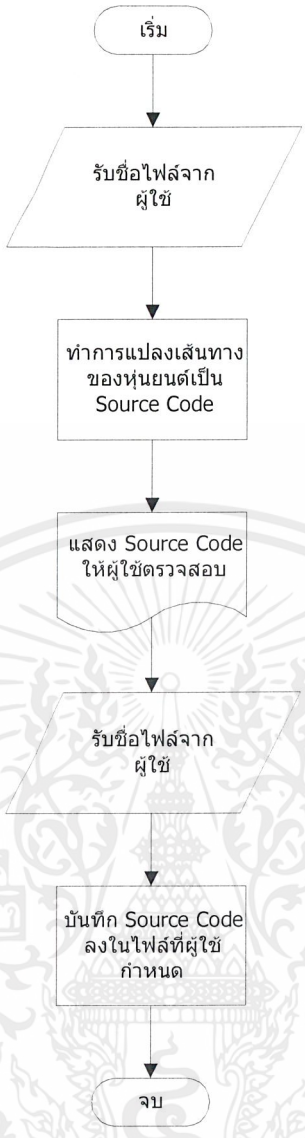


รูปที่ 3.10 Flowchart แสดงขั้นตอนการปรับเปลี่ยนคุณสมบัติของการเคลื่อนที่ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

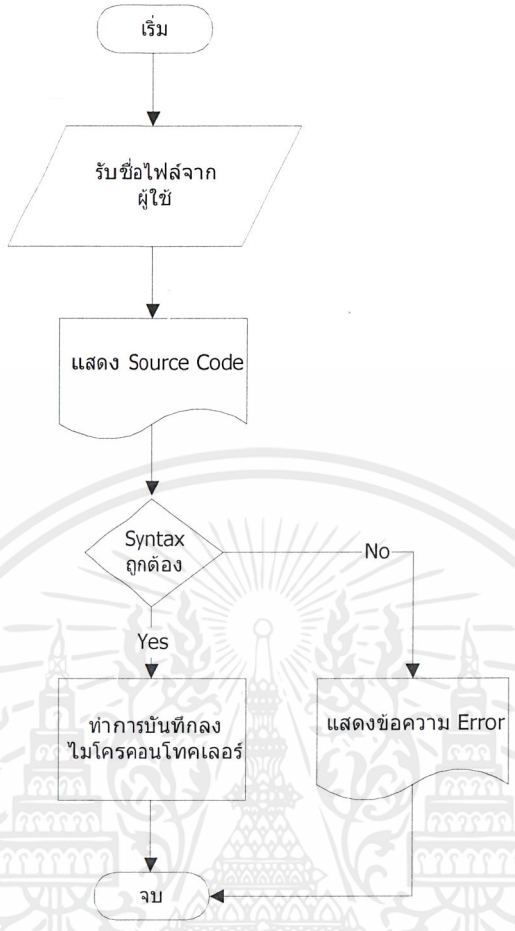
รูปที่ 3.11 Flowchart แสดงขั้นตอนการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ขั้นตอนการ Compile เส้นทางการเคลื่อนที่ของหุ่นยนต์เป็น Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Flowchart แสดงขั้นตอนการบันทึก Source Code ลงไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาระบบงาน

4.1 **ความต้องการทางด้านฮาร์ดแวร์** (Hardware Requirement)

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก มีความต้องการทางด้านการใช้อุปกรณ์หลายด้าน ประกอบด้วยความต้องการทางด้านอุปกรณ์คอมพิวเตอร์ และความต้องการทางด้านอุปกรณ์หุ่นยนต์ โดยสามารถจำแนกได้ดังนี้

4.1.1 ความต้องการทางด้านอุปกรณ์คอมพิวเตอร์

โปรแกรมจัดการและควบคุมหุ่นยนต์ขนาดเล็กมีความต้องการทางด้านอุปกรณ์คอมพิวเตอร์ต่าง ๆ ดังนี้

- เครื่องคอมพิวเตอร์ที่ใช้งานระบบปฏิบัติการ Windows 98 ได้เป็นอย่างต่ำ
- อุปกรณ์แสดงผล (Output) ได้แก่ จอมอนิเตอร์ (Monitor)
- อุปกรณ์สำหรับรับข้อมูลเข้า(Input) ได้แก่ คีย์บอร์ด (Keyboard) ,เมาส์ (Mouse)
- พื้นที่ว่างสำหรับใช้ในการบันทึกข้อมูลใน Harddisk อย่างน้อย 1 Gb
- พอร์ตสำหรับต่อสายสัญญาณ RS-232 (Serial Port)

4.1.2 ความต้องการทางด้านอุปกรณ์หุ่นยนต์

สำหรับหุ่นยนต์ที่ใช้ในการปฏิบัติการ จะใช้หุ่นยนต์ Robo-Kit ที่มีไมโครคอนโทรเลอร์ควบคุมรุ่น BasicStamp IISX โดยหุ่นยนต์จะประกอบไปด้วยอุปกรณ์หลักต่าง ๆ ดังนี้คือ

- แผงวงจรหลัก Robo-Kit พร้อมแผงวงจรต่อขนาด 2.5 x 3.5 นิ้ว 1 ชุด
- สายเชื่อมต่อคอมพิวเตอร์ (สายดาว์นโหลด) 1 ชุด
- โครงตัวถังพลาสติก 1 ชุด
- ยางวงแหวน พร้อมล้อข้าง 2 ชุด
- เซอร์โวมอเตอร์ 6V 2 ชุด
- กระบะถ่านขนาด AA 4 ก้อนพร้อมสายต่อ 1 ชุด
- กระบะถ่านขนาด AA 4 ก้อนพร้อมปลั๊ก 1 ชุด

4.2 **ความต้องการทางด้านซอฟท์แวร์** (Software Requirement)

ความต้องการทางด้านซอฟท์แวร์ของโปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก สามารถแบ่งออกเป็นส่วนต่าง ๆ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ความต้องการทางด้านระบบปฏิบัติการ

โปรแกรมจัดการและควบคุมหุ่นยนต์ขนาดเล็กจะต้องทำการประมวลผลภายใต้ระบบปฏิบัติการที่เป็นตระกูล Microsoft Windows ที่มี version ไม่ต่ำกว่า Windows 98

4.2.2 ความต้องการทางด้านอื่น ๆ

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก ได้ทำการพัฒนาจากโปรแกรม Microsoft Visual Basic 6.0 ซึ่งทำหน้าที่เป็นคอมไพล์เลอร์ของโปรแกรมและโปรแกรม BasicStamp ซึ่งทำหน้าที่เป็นโปรแกรมที่ทำการบันทึกข้อมูลลงไมโครคอนโทรเลอร์ BasicStamp IISX

4.3 ขั้นตอนการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมจะทำการแบ่งขั้นตอนออกเป็นสองส่วนใหญ่ ๆ คือส่วนแรกสำหรับจัดการทางด้านโปรแกรมที่ใช้ในการพัฒนา และส่วนที่สองสำหรับจัดการทางด้านอุปกรณ์หุ่นยนต์ซึ่งจะทำการประกอบหุ่นยนต์ให้สามารถนำมาใช้งานได้จริง

4.3.1 การจัดการทางด้านโปรแกรมที่ใช้ในการพัฒนา

4.3.1.1 ในขั้นตอนแรก ผู้ที่ทำการพัฒนาจะต้องทำการติดตั้งระบบปฏิบัติการ Windows 98 ก่อน รวมถึงจัดการติดตั้ง Driver ของอุปกรณ์เสริมต่าง ๆ เพื่อให้โปรแกรมสามารถทำงานได้ถูกต้อง

4.3.1.2 หลังจากทำการติดตั้งระบบปฏิบัติการ Windows 98 เรียบร้อยแล้ว จะต้องทำการติดตั้งโปรแกรม Microsoft Visual Basic 6.0 โดยสามารถหาได้จากแผ่นโปรแกรม Microsoft Visual Studio 6.0 ของบริษัทไมโครซอฟท์ และเลือก package ที่ชื่อว่า Microsoft Visual Basic 6.0 ในขณะติดตั้ง

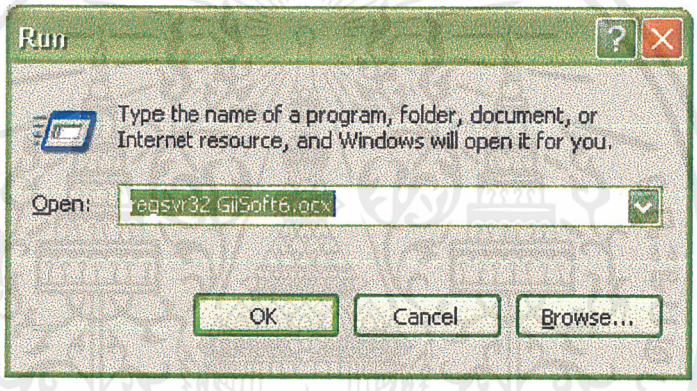
4.3.1.3 หลังจากติดตั้งเสร็จแล้ว ผู้ใช้ควรที่จะทำการ update โปรแกรม Microsoft Visual Studio 6.0 โดยทำการติดตั้ง Microsoft Visual Studio 6.0 Service Pack 5 เพื่อทำการแก้ไขข้อผิดพลาดของโปรแกรม Microsoft Visual Basic โดยขั้นตอนการติดตั้งสามารถทำได้ดังนี้

- ทำการ download โปรแกรม Microsoft Visual Studio 6.0 Service Pack 5 ได้จากเวปไซท์ของบริษัท Microsoft (www.microsoft.com)
- หลังจากที่ทำการ download เสร็จสิ้นแล้ว ให้ทำการติดตั้งโปรแกรม โดยรันไฟล์ setupsp5.exe ที่ได้รับมา
- หลังจากนั้นโปรแกรมจะทำการ update ส่วนต่าง ๆ ของ Microsoft Visual Studio 6.0 โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.4 เมื่อทำการ update เสร็จเรียบร้อยแล้ว จะต้องทำการเพิ่มส่วนของ ocx ที่ใช้ในการพัฒนาโปรแกรม ทำให้สามารถพัฒนาโปรแกรมได้สะดวกมากขึ้น โดยส่วนของโปรแกรมนี้มีชื่อว่า GiiSoft สามารถหา download ได้จาก http://vbstreet.hypermart.net และเมื่อ download มาเรียบร้อยแล้ว จะต้องทำการ register เข้าไปยังหน่วยความจำของระบบปฏิบัติการ โดยสามารถทำได้ดังนี้

- ทำการ extract zip file ไปยัง directory "c:\windows\system"
- คลิ๊กที่ปุ่ม start ทางด้านซ้ายมือของหน้าจอ
- ทำการเลือกคำสั่ง run
- พิมพ์คำสั่งตามนี้คือ "regsvr32 Giisoft6.ocx"



รูปที่ 4.1 รูปภาพแสดงวิธีการติดตั้ง GiiSoft6.ocx

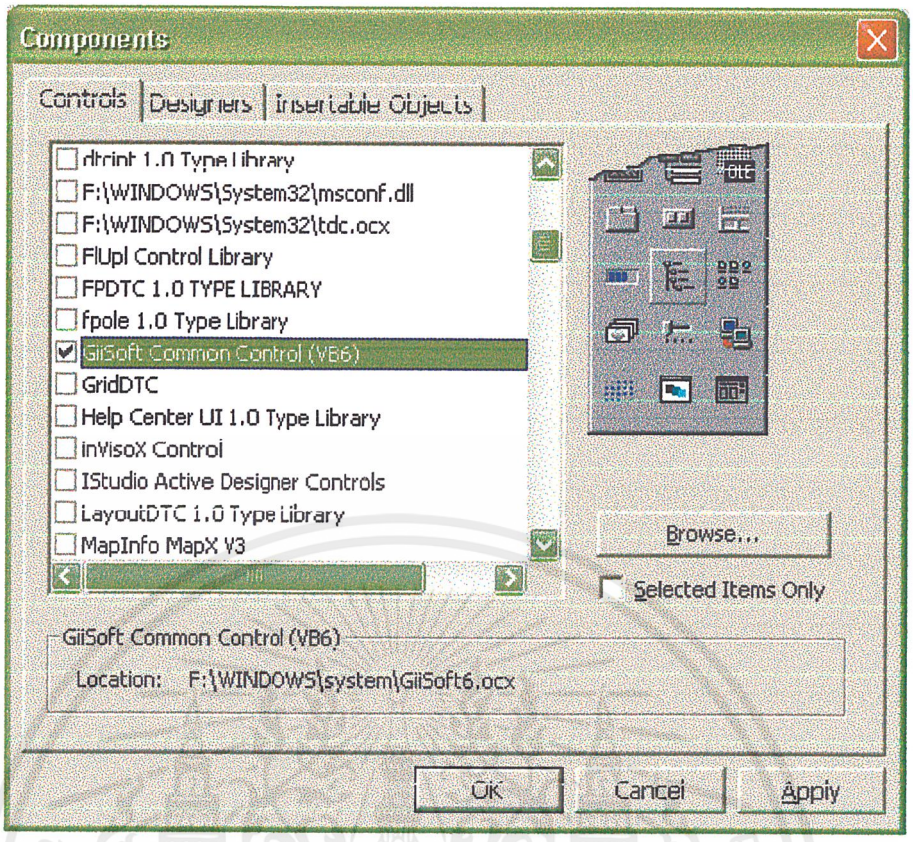
4.3.1.5 หลังจากนั้นผู้พัฒนาจะต้องทำการเรียกใช้ componet โดย

4.3.1.5.1 เปิดโปรแกรม Microsoft Visual Basic

4.3.1.5.2 Click ที่ปุ่ม Project --> Components หรือกด Ctrl + T

4.3.1.5.3 โปรแกรมจะทำการแสดงหน้าจอ Components

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 รูปแสดงหน้าจอ Components ของโปรแกรม Microsoft Visual Basic

4.3.1.5.3 ทำการเลือก Componet ที่ชื่อว่า GiiSoft Common Control (VB6)

4.3.1.5.4 Click ที่ปุ่ม OK เพื่อยืนยัน

4.3.1.6 เมื่อทำการติดตั้งแล้ว ผู้พัฒนาจะต้องทำการติดตั้งโปรแกรม BasicStamp เพื่อที่จะสามารถบันทึกข้อมูลลงไมโครคอนโทรเลอร์ได้

4.3.1.7 หลังจากทำการติดตั้งทุกอย่างเสร็จสิ้นแล้ว ผู้พัฒนาสามารถเริ่มทำการพัฒนาโปรแกรมได้

4.3.2 การจัดการทางด้านอุปกรณ์หุ่นยนต์

เมื่อผู้พัฒนาได้ทำการติดตั้งส่วนของโปรแกรมที่ใช้ในการพัฒนาเรียบร้อยแล้ว ขั้นตอนต่อไป ผู้พัฒนาจะต้องทำการติดตั้งหุ่นยนต์ที่ใช้ในการพัฒนา โดยขั้นตอนต่าง ๆ ดังนี้

ติดตั้งเสารอง และกระบะถ่าน

4.3.2.1 นำเสารอง 20 มม. ยึดเข้ากับโครงหุ่นยนต์ตามตำแหน่งที่กำหนดด้วยสกรูเกลียวปล่อยทั้ง 4 มุม

4.3.2.2 ติดตั้งกระบะถ่านแบบสายต่อเข้ากับส่วนหลังของโครงหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.3 นำกระบะถ่าน 4 ก้อนแบบสายปลั๊กเข้ากับโครงหุ่นยนต์ทางด้านบน โดยติดตั้งให้ชิดมาทางด้านหน้า ยึดด้วยสกรูหัวเทเปอร์ 2 ตัว

ติดตั้งเซอร์โวมอเตอร์ และล้อข้าง

4.3.2.4 นำเซอร์โวมอเตอร์ที่ดัดแปลงแล้วติดตั้งเข้ากับโครงหุ่นยนต์ โดยใช้สกรูขนาด 3x8 มม. โดยติดตั้งทีละตัว จนเรียบร้อย

4.3.2.5 นำยางวงแหวนใส่เข้ากับล้อข้างให้เรียบร้อยทั้งสองล้อ

4.3.2.6 นำล้อข้างที่ใส่ยางแล้ว เสียบเข้าไปในแกนหมุนของเซอร์โวมอเตอร์

4.3.2.7 ใช้สกรูที่ถอดออกจากก้านหมุนในขั้นตอนการดัดแปลงเซอร์โวมอเตอร์ ขันยึดเข้ากับเซอร์โวมอเตอร์

ติดตั้งส่วนประกอบที่เหลือ

4.3.2.8 นำแบตเตอรี่ AA ขนาด 1.5 V 8 ก้อนติดตั้งลงบนกระบะถ่านทั้งสองตัว ตัวละ 4 ก้อน

4.3.2.9 นำแผงวงจรหลักวางบนเสารองทั้ง 4 ตัวแล้วยึดด้วยสกรูเกลียวปล่อย

4.3.2.10 ต่อสายจากเซอร์โวมอเตอร์ และจากกระบะถ่านแบบสายต่อเข้ากับแผงวงจร

การปรับแต่งเซอร์โวมอเตอร์

โดยปกติแล้วเซอร์โวมอเตอร์จะสามารถหมุนได้ตั้งแต่ 0-120 องศา นั่นคือไม่สามารถหมุนได้รอบตัว แต่ในการสร้างหุ่นยนต์ Robo-Kit จะต้องทำให้เซอร์โวมอเตอร์สามารถหมุนได้ 0-360 องศา หรือหมุนได้รอบตัวอย่างไม่ติดขัด ดังนั้นผู้พัฒนาจะต้องทำการดัดแปลง และปรับแต่งเซอร์โวมอเตอร์ ซึ่งสามารถทำได้ดังนี้

4.3.2.11 นำเซอร์โวมอเตอร์ทั้งสองตัว มาทำการกำหนดก่อนว่าต้องการให้ตัวใดเป็นมอเตอร์ด้านซ้าย หรือด้านขวา

4.3.2.12 ใช้ไขควงสี่แฉกถอนสกรูที่ก้านหมุนของมอเตอร์ออก แล้วเก็บสกรูที่ยึดก้านหมุนไว้

4.3.2.13 คว่ำมอเตอร์ลง แล้วใช้ไขควงสี่แฉกถอนสกรูที่ด้านล่างออก

4.3.2.14 เปิดฝาด้านบนของมอเตอร์ออก จะเห็นเฟืองวางซ้อนกัน

4.3.2.15 ถอดเฟืองตัวกลางออก แล้วเก็บไว้ เพราะประกอบกลับคืนภายหลัง

4.3.2.16 ถอดเฟืองขับเคลื่อนซึ่งเป็นเฟืองตัวใหญ่สุดออก

4.3.2.17 ทำการตัดเดือย (Stop tab) ของเฟืองขับเคลื่อน ต้องระมัดระวัง โดยต้องตัดเพียงเฟืองหยุดเท่านั้น อย่าตัดถูกแกนเฟืองเด็ดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.18 เมื่อตัดเดือยหยุดแล้ว ให้คว่ำเฟืองขับเคลื่อนลง ทำการถอดแผ่นขับ (drive plate) ออกจากเฟืองขับ แล้วเก็บเฟืองขับเคลื่อนไว้ เพราะต้องใช้ประกอบกลับคืนภายหลัง

4.3.2.19 ทำการต่อสายของเซอร์โวมอเตอร์เข้ากับแผงวงจรหลัก Robo-Kit

4.3.2.20 เชื่อมแผงวงจรหลัก Robo-Kit เข้ากับคอมพิวเตอร์ เปิดโปรแกรมเบสิกแสตมป์เอดิเตอร์ แล้วเขียนโปรแกรมเพื่อสั่งให้เซอร์โวมอเตอร์หยุดทำงาน

4.3.2.21 จ่ายไฟให้กับแผงวงจรหลักแล้วทำการดาว์นโหลดโปรแกรมลงแผงวงจรหลัก

4.3.2.22 จ่ายไฟให้กับเซอร์โวมอเตอร์ โดยใช้แบตเตอรี่ 1.5 V 4 ก้อนบรรจุลงในกระบะถ่าน แล้วต่อสายของกระบะถ่านลงบนแผงวงจร เนื่องจากยังไม่ได้ปรับแต่งเซอร์ไวมอเตอร์ดังนั้นจะเห็นเฟืองของมอเตอร์เริ่มหมุน

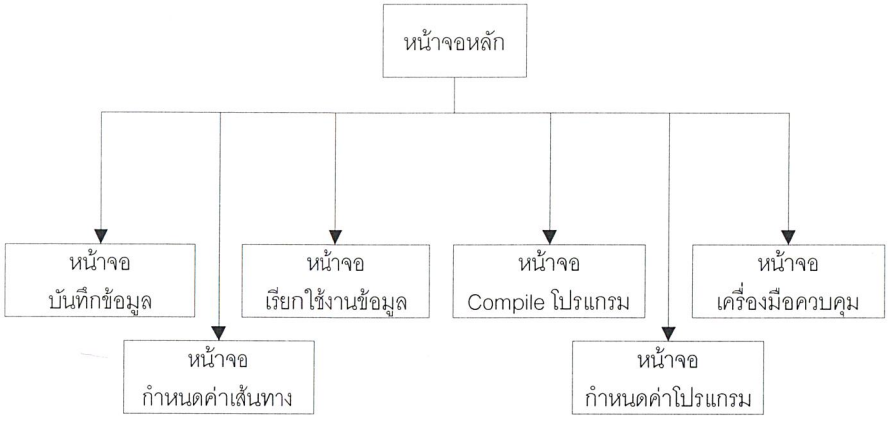
4.3.2.23 ใช้คีมปากจิ้งจกหมุนแกนของตัวต้านทานปรับค่า จนกระทั่งเฟืองทั้งหมดหยุดหมุน นั่นคือมอเตอร์หยุดหมุนด้วย จากนั้นปลดไฟเลี้ยงมอเตอร์ และแผงวงจรหลักออก ในการปรับแต่งเซอร์โวมอเตอร์นั้นถ้าต้องการให้ได้ค่าที่ถูกต้องอย่างละเอียด จำเป็นต้องใช้อุปกรณ์ในการปรับแต่งที่เหมาะสม เพราะการตรวจสอบว่ามอเตอร์หยุดหมุนอย่างแท้จริงหรือไม่นั้น ไม่สามารถมองได้ด้วยตาเปล่า จากการปรับแต่งด้วยตนเองนั้นพบว่า เมื่อดูจากภายนอกแม้ว่ามอเตอร์หยุดหมุนแล้ว แต่ยังคงได้ยินเสียงมอเตอร์เคลื่อนไหวอยู่ และเมื่อนำไปใช้งานปรากฏว่าการเคลื่อนที่ของหุ่นยนตร์ยังไม่ถูกต้อง

เมื่อผู้พัฒนาได้ทำการติดตั้งทุกอย่างเรียบร้อยแล้ว ขั้นตอนต่อไปจึงเป็นการพัฒนาโปรแกรม โดยในการพัฒนาโปรแกรมนั้น จะต้องทำการสร้างโปรแกรม พร้อมกับทดสอบเพื่อค้นหาข้อผิดพลาดของโปรแกรม ทำการปรับแต่งค่าต่าง ๆ ที่ใช้ รวมถึงตรวจสอบขั้นตอนการทำงานของโปรแกรม ให้สามารถนำไปใช้งานกับหุ่นยนต์ได้จริง

4.4 แผนภาพการทำงานของโปรแกรม

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก สามารถแบ่งขั้นตอนการทำงานออกเป็นส่วนต่าง ๆ กันได้ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 รูปแสดงส่วนประกอบหลักต่าง ๆ ของโปรแกรม

โดยที่หน้าจอหลักต่าง ๆ ของโปรแกรม มีคุณสมบัติการทำงานดังนี้คือ

4.4.1 หน้าจอหลัก

ภายในหน้าจอหลัก จะเป็นหน้าจอที่ผู้ใช้ทำการใช้งานโปรแกรม โดยผู้ใช้จะทำการกำหนดลักษณะเส้นทาง รวมถึงลักษณะการเดินทางของหุ่นยนต์ โดยภายในหน้าจอหลัก จะมีส่วนประกอบหลัก ๆ ดังนี้คือ

4.4.1.1 Menu Bar จะเป็นส่วนที่ให้ผู้ใช้งานควบคุมการใช้งานของโปรแกรม โดยผู้ใช้โปรแกรมสามารถที่จะเลือก และปรับแต่งลักษณะของโปรแกรมได้ ผ่านทาง Menu Bar นี้ โดยภายใน Menu Bar จะมีคำสั่งย่อยคือ

4.4.1.1.1 File : เป็น Menu ที่รวบรวมการจัดการเกี่ยวกับ input และ output ของโปรแกรม รวมถึงการออกจากโปรแกรม และการเริ่มต้นการใช้งานโปรแกรมใหม่

4.4.1.1.1.1 New : เป็นคำสั่งสำหรับการเริ่มต้นการใช้งานโปรแกรมใหม่ โดยโปรแกรมจะทำการ refresh สถานะทุกอย่างที่ผู้ใช้งานทำก่อนหน้ากลับไปเป็นสถานะเริ่มต้น

4.4.1.1.1.2 Save : เป็นคำสั่งสำหรับการบันทึกข้อมูลเส้นทางการเดินทางของหุ่นยนต์ รวมถึงคุณสมบัติต่าง ๆ ที่ผู้ใช้กำหนด โดยคำสั่งนี้จะทำการเรียกใช้งาน **หน้าจอบันทึกข้อมูล**

4.4.1.1.1.3 Open : เป็นคำสั่งสำหรับการเรียกใช้งานเส้นทางการเดินทางของหุ่นยนต์ ที่ผู้ใช้เคยบันทึกไว้แล้ว โดยคำสั่งนี้จะทำการเรียกใช้งาน **หน้าจอเรียกใช้งานข้อมูล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1.1.1.4 Exit : เป็นคำสั่งสำหรับออกจากโปรแกรม เมื่อผู้ใช้ไม่ต้องการใช้งานโปรแกรมแล้ว

4.4.1.1.2 View : เป็น Menu ที่จัดการเกี่ยวกับลักษณะมุมมองของผู้ใช้ โดยผู้ใช้สามารถที่จะกำหนดได้ว่าต้องการให้หน้าจอประเภทใดแสดงหรือไม่แสดง

4.4.1.1.2.1 ToolBar : เป็นคำสั่งสำหรับเรียกให้แสดง หรือปิด**หน้าจอเครื่องมือควบคุม** โดยที่ถ้ามีเครื่องหมาย √ อยู่ข้างหน้า จะหมายถึงหน้าจอแสดงเครื่องมือควบคุมถูกแสดงอยู่ และถ้าผู้ใช้งาน Click ที่คำสั่ง ToolBar จะเป็นการซ่อนหน้าจอเครื่องมือควบคุม

4.4.1.1.3 Properties : เป็น Menu ที่จัดการเกี่ยวกับคุณลักษณะของโปรแกรม ไม่ว่าจะเป็นลักษณะของสีพื้นหลัง การแสดงเส้น grid รวมถึงแบบตัวอักษรที่ใช้

4.4.1.1.3.1 Grid : เป็นคำสั่งที่ใช้สำหรับแสดง หรือซ่อนเส้น grid บนหน้าจอการทำงานหลัก

4.4.1.1.3.2 Preference : เป็นคำสั่งสำหรับแสดง**หน้าจอกำหนดค่าโปรแกรม** สำหรับปรับเปลี่ยนค่าคุณสมบัติต่าง ๆ ของโปรแกรม

4.4.1.1.4 Compile : เป็น Menu ที่จัดการเกี่ยวกับการแปลงเส้นทางที่ผู้ใช้กำหนดให้เป็น Code ของภาษา PBASIC

4.4.1.1.4.1 Compile : เป็นคำสั่งที่ใช้สำหรับแสดง**หน้าจอCompileโปรแกรม** เพื่อใช้ในการแปลงเส้นทางที่ผู้ใช้กำหนด ให้เป็น Code ของภาษา PBASIC

4.4.1.2 MainPic : Picture Box

เป็นส่วนสำหรับแสดงเส้นทางที่ผู้ใช้กำหนด รวมถึงแสดงเส้น grid ที่ช่วยทำให้ผู้ใช้สามารถกำหนดเส้นทางต่าง ๆ ได้ง่ายขึ้น

4.4.1.3 ScrollBar

เป็นส่วนสำหรับให้ผู้ใช้สามารถเลื่อนมุมมองใน Picture Box ทำให้ผู้ใช้สามารถที่สร้างเส้นทางได้ในระยะทางที่ยาวมากขึ้นทั้งทางแนวตั้ง และแนวนอน

4.4.1.4 GridAtt : MsFlexGrid

เป็นส่วนที่ทำการแสดงถึงคุณสมบัติของโปรแกรมต่าง ๆ ที่ผู้ใช้กำหนด โดยภายในส่วนนี้จะประกอบไปด้วยรายการย่อย ๆ ทั้งหมด 6 รายการ ได้แก่

- ชื่อไฟล์
- จำนวนเส้นทาง
- การตรวจสอบการชน
- เปิดไฟเลี้ยวซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เปิดไฟเลี้ยวขวา
- วนซ้ำเส้นทาง

โดยที่รายการ การตรวจสอบการชน ,เปิดไฟเลี้ยวซ้าย ,เปิดไฟเลี้ยวขวา ,วนซ้ำเส้นทาง จะเก็บคุณสมบัติเป็นแบบ True หรือ False ผู้ใช้สามารถที่เปลี่ยนแปลงค่าได้ โดยการ Double Click ที่ตำแหน่งของค่าที่ต้องการเปลี่ยน

4.4.1.5 GridHistory : MsFlexGrid

เป็นส่วนที่ทำการแสดงถึงรายการเส้นทางทั้งหมดที่ผู้ใช้ได้กำหนดไว้ โดยจะแสดงรายการทั้งหมดที่ผู้ใช้ได้ทำไปตามลำดับ ผู้ใช้สามารถเปลี่ยนแปลงรายการเหล่านี้ได้

4.4.1.6 Toolbar1 : ToolBar

เป็นส่วนที่ใช้สำหรับการจัดการกับค่าเส้นทางที่ผู้ใช้กำหนด โดยมีเมนูย่อยทั้งหมด 3 รายการ คือ

- **ลบรายการเดียว** ใช้สำหรับลบรายการเส้นทางที่ผู้ใช้เลือกจาก GridHistory
- **ลบหลายรายการ** ใช้สำหรับลบรายการเส้นทางตั้งแต่เส้นทางที่ผู้ใช้เลือก
- **ล๊อครายการ** ใช้สำหรับล๊อครายการที่ผู้ใช้เลือก เพื่อป้องกันความผิดพลาดจากการลบเส้นทางโดยไม่ได้ตั้งใจจากผู้ใช้

4.4.2 หน้าจอบันทึกข้อมูล

หน้าจอบันทึกข้อมูล เป็นหน้าจอสำหรับการบันทึกเส้นทางการเดินทางของหุ่นยนต์ ที่ผู้ใช้ได้กำหนดขึ้น รวมถึงคุณสมบัติต่าง ๆ ของโปรแกรมที่ผู้ใช้กำหนด ให้เป็นไฟล์ เพื่อที่ผู้ใช้จะสามารถเรียกใช้เส้นทางที่บันทึกไว้ได้ในภายหลัง

โดยภายในหน้าจอบันทึกข้อมูล จะประกอบไปด้วยส่วนประกอบย่อยต่าง ๆ ดังนี้

4.4.2.1 SaveDrive : DriveListBox เป็นส่วนที่ใช้สำหรับการกำหนด drive ที่ผู้ใช้ต้องการบันทึกไฟล์

4.4.2.2 SaveDir : DirListBox เป็นส่วนที่ใช้สำหรับการกำหนด directory หรือ path ที่ผู้ใช้ต้องการบันทึกไฟล์

4.4.2.3 SaveFileBox : FileListBox เป็นส่วนที่ใช้สำหรับแสดงรายการไฟล์ที่มีอยู่ใน directory ที่กำหนดอยู่ใน SaveDir

4.4.2.4 TxtFileName : TextBox เป็นส่วนที่ใช้สำหรับกำหนดชื่อไฟล์ที่ผู้ใช้ต้องการบันทึกไฟล์

4.4.2.5 BtnNewFolder : Button เป็นส่วนที่ใช้สำหรับกำหนด directory ใหม่

4.4.2.6 Bcancel : Button เป็นส่วนที่ใช้สำหรับยกเลิกการบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2.7 BSave : Button เป็นส่วนที่ใช้สำหรับบันทึกข้อมูล

4.4.3 หน้าจอเรียกใช้งานข้อมูล

เป็นส่วนที่ใช้สำหรับเรียกใช้งานข้อมูลที่ผู้ใช้เคยบันทึกไว้แล้ว โดยในการเรียกใช้งานโปรแกรมจะทำการนำเอาเส้นทาง รวมถึงคุณสมบัติต่าง ๆ ที่ผู้ใช้กำหนดไว้มาแสดงทั้งหมด

โดยภายในหน้าจอเรียกใช้งานข้อมูล จะประกอบไปด้วยส่วนประกอบย่อยต่าง ๆ ดังนี้

4.4.3.1 OpenDrive : DriveListBox เป็นส่วนที่ใช้สำหรับการกำหนด drive ที่ผู้ใช้ต้องการเรียกใช้งานไฟล์

4.4.3.2 OpenDir : DirListBox เป็นส่วนที่ใช้สำหรับการกำหนด directory หรือ path ที่ผู้ใช้ต้องการเรียกใช้งานไฟล์

4.4.3.3 OpenFileBox : FileListBox เป็นส่วนที่ใช้สำหรับแสดงรายการไฟล์ที่มีอยู่ใน directory ที่กำหนดอยู่ใน OpenDir

4.4.3.4 TxtFileName : TextBox เป็นส่วนที่ใช้สำหรับกำหนดชื่อไฟล์ที่ผู้ใช้ต้องการเรียกใช้งาน

4.4.3.5 CmbFType : ComboBox เป็นส่วนสำหรับการกำหนดประเภทของไฟล์ที่ผู้ใช้ต้องการเรียกใช้งาน

4.4.3.1 Bcancel : Button เป็นส่วนที่ใช้สำหรับยกเลิกการเรียกใช้งาน

4.4.3.6 BOpen : Button เป็นส่วนที่ใช้สำหรับเรียกใช้งานไฟล์ที่ผู้ใช้ต้องการ

4.4.4 หน้าจอ Compile โปรแกรม

เป็นส่วนที่ใช้งานสำหรับการบันทึก Source Code ภาษา PBASIC ที่ถูก Compile แล้วไปเป็นไฟล์ที่สามารถนำไปบันทึกลงไมโครคอนโทรเลอร์ได้

ภายในหน้าจอ Compile โปรแกรม จะประกอบไปด้วยส่วนประกอบย่อย ๆ ดังนี้คือ

4.4.4.1 LstCode : ListBox เป็นส่วนที่ใช้สำหรับการแสดง source code ภาษา PBASIC ที่ Compile ตามเส้นทางที่ผู้ใช้กำหนด โดยไฟล์ที่จะทำการ compile จะมีข้อมูลภายในเช่นเดียวกับที่แสดงอยู่ในส่วนนี้

4.4.4.2 FileBwser : FolderBrowser เป็นส่วนที่ใช้สำหรับกำหนด path ปลายทางที่จะเก็บไฟล์ที่ได้จากการ compile

4.4.4.3 TxtFileName : TextBox เป็นส่วนที่ใช้สำหรับกำหนดชื่อไฟล์ที่จะเป็น target file ในการบันทึก

4.4.4.4 BCancel : Button เป็นส่วนที่ใช้สำหรับยกเลิกการ compile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4.5 CmdCompile : Button เป็นส่วนที่ใช้สำหรับการบันทึกไฟล์ที่ได้รับการ compile แล้ว

4.4.5 หน้าจอเครื่องมือควบคุม

เป็นส่วนที่ใช้สำหรับการกำหนดทิศทางเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยภายในหน้าจอนี้จะประกอบไปด้วยส่วนควบคุมลักษณะการเดินทางทั้งหมด 7 เส้นทางคือ

4.4.5.1 ToolBar1 : ToolBar เป็นส่วนที่ทำการรวบรวมเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยภายใน ToolBar1 จะประกอบไปด้วยลักษณะเส้นทางดังนี้

4.4.5.1.1 Forward กำหนดให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า

4.4.5.1.2 Backward กำหนดให้หุ่นยนต์เคลื่อนที่ไปข้างหลัง

4.4.5.1.3 Turn Left กำหนดให้หุ่นยนต์เคลื่อนที่ไปทางด้านซ้าย

4.4.5.1.4 Turn Right กำหนดให้หุ่นยนต์เคลื่อนที่ไปทางด้านขวา

4.4.5.1.5 Rotate Left กำหนดให้หุ่นยนต์หมุนทางด้านซ้าย

4.4.5.1.6 Rotate Right กำหนดให้หุ่นยนต์หมุนทางด้านขวา

4.4.6 หน้าจอกำหนดค่าเส้นทาง

เป็นหน้าจอที่ใช้สำหรับกำหนดค่าที่ใช้ในการเคลื่อนที่ โดยค่าที่กำหนดนี้จะสัมพันธ์กับทิศทางที่ผู้ใช้เลือกในหน้าจอเครื่องมือควบคุม ภายในหน้าจอนี้จะประกอบไปด้วยส่วนประกอบหลักคือ

4.4.6.1 TxtMoveFor : TextBox เป็นส่วนที่ใช้สำหรับกำหนดระยะทางในการเคลื่อนที่ไปข้างหน้า

4.4.6.2 TxtMoveBack : TextBox เป็นส่วนที่ใช้สำหรับกำหนดระยะทางในการเคลื่อนที่ไปข้างหลัง

4.4.6.3 TxtMoveLeft : TextBox เป็นส่วนที่ใช้สำหรับกำหนดระยะทางในการเคลื่อนที่ไปทางด้านซ้าย

4.4.6.4 TxtMoveRight : TextBox เป็นส่วนที่ใช้สำหรับกำหนดระยะทางในการเคลื่อนที่ไปทางด้านขวา

4.4.6.5 TxtRoLeft : TextBox เป็นส่วนที่ใช้สำหรับกำหนดค่าองศาที่หมุนไปทางซ้าย

4.4.6.6 TxtRoRight : TextBox เป็นส่วนที่ใช้สำหรับกำหนดค่าองศาที่หมุนไปทางขวา

4.4.6.7 TxtPause : TextBox เป็นส่วนที่ใช้สำหรับกำหนดระยะเวลาในการหยุดการเดินทางของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.6.8 CmdArgu : Button เป็นส่วนที่ใช้สำหรับยืนยันการกำหนดตามที่ผู้ใช้กำหนด

4.4.7 หน้าจอกำหนดค่าโปรแกรม

เป็นหน้าจอที่ใช้สำหรับกำหนดค่าส่วนต่าง ๆ ของโปรแกรมที่ไม่เกี่ยวข้องกับเส้นทางการเดินทางของหุ่นยนต์ โดยผู้ใช้สามารถกำหนดค่าได้สองส่วนคือ สีพื้นหลัง และรูปแบบตัวอักษร

ภายในหน้าจอนี้ ประกอบไปด้วยส่วนต่าง ๆ ดังนี้คือ

4.4.7.1 List1 : ListBox เป็นส่วนที่ใช้ในการแสดงประเภทรูปแบบที่ผู้ใช้สามารถกำหนดได้ คือ Font และ BackGround

4.4.7.2 FontBws : FontBrowser เป็นส่วนที่ใช้ในการเลือก Font ที่ผู้ใช้สามารถเลือกได้

4.4.7.3 ColorBoard : PictureBox เป็นส่วนที่ใช้ในการแสดงสีที่ผู้ใช้สามารถเลือกได้

4.4.7.4 Scroll(0) : HScrollBar เป็นส่วนที่ใช้กำหนดอัตราส่วนของสีแดง

4.4.7.5 Scroll(1) : HScrollBar เป็นส่วนที่ใช้กำหนดอัตราส่วนของสีเขียว

4.4.7.6 Scroll(2) : HScrollBar เป็นส่วนที่ใช้กำหนดอัตราส่วนของสีฟ้า

4.4.7.7 Command3 : Button เป็นส่วนที่ใช้สำหรับยืนยันการปรับปรุงคุณสมบัติของโปรแกรม

4.4.7.8 Command2 : Button เป็นส่วนที่ใช้สำหรับปรับปรุงการแก้ไขของโปรแกรม

4.4.7.9 Command1 : Button เป็นส่วนที่ใช้สำหรับยกเลิกการแก้ไขของโปรแกรม

4.5 ผลการทดลองใช้งานโปรแกรม

เพื่อตรวจสอบความถูกต้องในการทำงานของโปรแกรมและหุ่นยนตร์ว่าทำงานได้ตรงตามที่ได้กำหนดไว้หรือไม่ จึงต้องมีการทดสอบการทำงานของโปรแกรมเพื่อตรวจสอบความถูกต้องและทำการปรับเปลี่ยนค่าคงที่ต่าง ๆ เพื่อหาค่าที่เหมาะสมกับสภาพแวดล้อมในการทำงานจริง

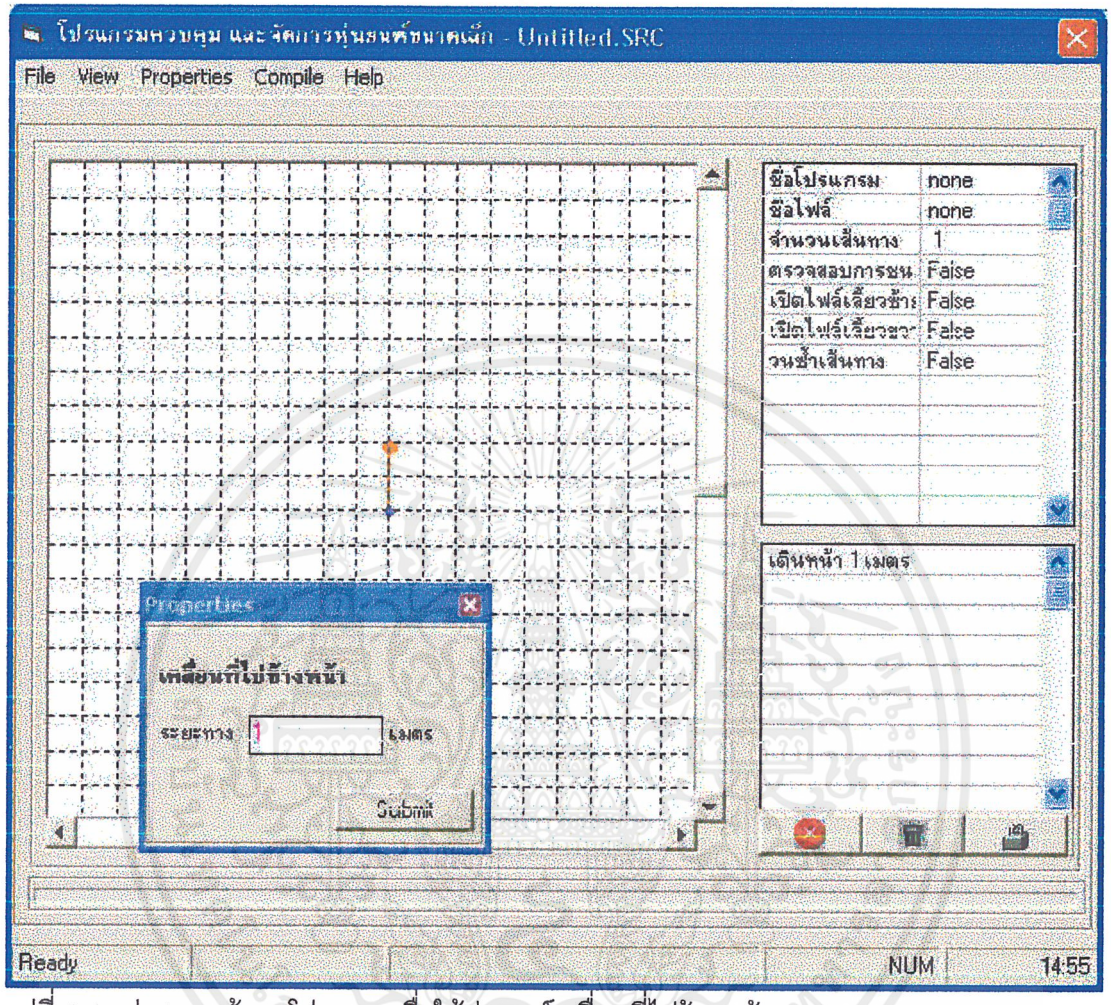
การทดลองครั้งแรก ๆ จะใช้ค่าคงที่ที่ทางโรงงานระบุมาให้เพื่อตรวจสอบว่าสามารถทำงานได้ตามที่ผู้พัฒนาต้องการหรือไม่ โดยจะทำการทดลองตามฟังก์ชันการทำงานหลัก ๆ ที่มีในโปรแกรมดังต่อไปนี้

4.5.1 การเดินทางเป็นเส้นตรง

แนวความคิดที่ใช้ในการเคลื่อนที่ไปข้างหน้าคือ ให้เซอร์โวสองตัวหมุนไปพร้อมกันในทิศที่สวนทางกันในระยะเวลาที่กำหนด เพื่อให้หุ่นยนต์เคลื่อนที่ได้ตามระยะทางที่ต้องการ โดยเวลาที่ใช้ในการหมุนเซอร์โวจะเป็นค่าที่จะต้องกำหนดให้กับค่าขอบเขตที่ใช้ในการวนลูป จากการคำนวนค่าที่ใช้ในการเคลื่อนที่ไปข้างหน้า 1 เมตรจะมีค่าเท่ากับ 307 ทำการกำหนดค่านี้ให้กับโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทดลอง - กำหนดจุดกึ่งกลางของล้อด้านหน้าให้เป็นจุดเริ่มต้น จากนั้นสั่งให้เดินทางไปด้านหน้าหนึ่งเมตร กำหนดให้จุดกึ่งกลางล้อด้านหน้าเป็นจุดสุดท้าย แล้ววัดระยะทางจากจุดเริ่มต้นมายังจุดสุดท้าย พร้อมทั้งบันทึกผล



รูปที่ 4.4 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า 1 เมตร

เส้นทางการเคลื่อนที่ - เดินทางไปด้านหน้าเป็นระยะทาง 1 เมตร

ผลการทดลอง – หุ่นยนต์เคลื่อนที่ไปข้างหน้าได้ตามที่ระบุ แต่มีการกระตุกเมื่อพบพื้นผิวที่ไม่เรียบ ทำให้ทิศทางเบี่ยงเบนไปบ้างเล็กน้อย

ครั้งที่ 1 ระยะทางที่วัดได้ 96 เซนติเมตร

ครั้งที่ 2 ระยะทางที่วัดได้ 94 เซนติเมตร

ครั้งที่ 3 ระยะทางที่วัดได้ 96 เซนติเมตร

4.5.2 การเดินไปในทิศทางตรงข้ามกับทิศทางเดิม (ย้อนกลับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทดลอง - กำหนดจุดกึ่งกลางของล้อด้านหน้าเป็นจุดเริ่มต้น จากนั้นหุ่นยนต์จะหมุนตัวไปทางด้านซ้าย 180 องศา โดยใช้ล้อด้านซ้ายเป็นจุดหมุน เมื่อหมุนครบรอบแล้วจะทำให้จุดกึ่งกลางของล้อด้านหน้าห่างจากจุดเริ่มต้นเป็นระยะเท่ากับขนาดของตัวหุ่นยนต์ (19 เซนติเมตร ให้รวมค่านี้เข้ากับระยะทางการเคลื่อนที่ด้วย เนื่องจากเป็นระยะที่ห่างจากจุดเริ่มต้นเดิม) จากนั้นหุ่นยนต์จะเดินทางไปด้านหน้าอีก 81 เซนติเมตร กำหนดจุดกึ่งกลางล้อด้านหน้าเป็นจุดสุดท้าย แล้ววัดระยะทางพร้อมบันทึกผล

เส้นทางการเคลื่อนที่ - เดินทางไปด้านหลังเป็นระยะทาง 1 เมตร

ผลการทดลอง – ในการทดลองครั้งที่ 1 และ 2 ระหว่างการหมุนหุ่นยนตร์เกิดสะดุด เนื่องจากพื้นผิวที่ไม่เรียบ ทำให้องศาในการหมุนไม่ครบ 180 องศา ทำให้การวัดระยะทางเกิดการเบี่ยงเบนไปด้วย

ครั้งที่ 1 ระยะทางที่วัดได้ 92 เซนติเมตร

ครั้งที่ 2 ระยะทางที่วัดได้ 91 เซนติเมตร

ครั้งที่ 3 ระยะทางที่วัดได้ 97.5 เซนติเมตร

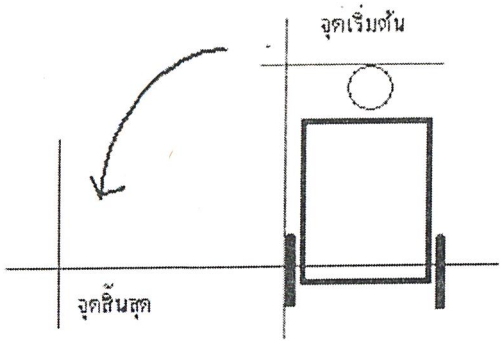
การทำงานในส่วนของการเคลื่อนที่จะมีลักษณะเดียวกับการเคลื่อนที่ไปด้านหน้า แต่เมื่อต้องเปลี่ยนทิศทางจึงมีการหมุนเข้ามาเกี่ยวข้องด้วย ดังนั้นการทดลองครั้งนี้ความผิดพลาดที่เกิดจากองศาของการหมุนก็มีผลทำให้ผลการทดลองที่ได้คลาดเคลื่อนไปด้วย

4.5.3 การหมุนไปทางซ้าย

แนวความคิดที่ใช้ในการเขียนโปรแกรมเพื่อให้หมุน จะใช้การวนรอบการทำงานของเซอร์โว ไปในทิศทางที่ต้องการตามระยะเวลาที่คำนวณได้ โดยใช้ FOR ลูป ค่าที่มีผลกับการทำงานคือขอบเขตของลูป ถ้าตั้งให้มีค่ามาก เซอร์โวก็จะหมุนเป็นระยะเวลานานทำให้องศาการหมุนมาก ดังนั้นจากการคำนวณค่าที่ใช้เมื่อต้องการหมุนเป็นระยะ 90 องศาคือ 78

วิธีการทดลอง - กำหนดจุดเริ่มต้นโดยลากเส้นตรงขนานกับด้านหน้าและด้านข้างของตัวหุ่นยนต์ (ใช้จุดกึ่งกลางด้านหน้าของหุ่นยนต์และจุดกึ่งกลางของล้อหลังด้านซ้ายตามลำดับเป็นหลัก) แล้วใช้จุดตัดที่เกิดจากเส้นตรงสองเส้นดังกล่าว เป็นจุดเริ่มต้น และใช้จุดกึ่งกลางของล้อหลังด้านซ้ายเป็นจุดหมุน จากนั้นให้ตัวหุ่นยนต์หมุนไปทางด้านซ้ายทำมุม 90 องศา ทำการกำหนดจุดสุดท้ายโดยใช้จุดตัดกันของเส้นตรงขนานกับด้านหน้าและด้านข้างของตัวหุ่นยนต์ (ดังที่กล่าวไว้ข้างต้น) แล้วทำการวัดมุมที่เกิดจากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 รูปแสดงการวัดมุมของการเลี้ยว

ผลการทดลอง - ในการหมุนทั้งสามครั้ง เมื่อเซอร์โวหยุดหมุนแล้วยังคงเกิดแรงเหวี่ยงที่ทำให้องศาการหมุนเพิ่มขึ้นเล็กน้อยทุก ๆ ครั้ง

ครั้งที่ 1 มุมที่วัดได้ 95 องศา

ครั้งที่ 2 มุมที่วัดได้ 95 องศา

ครั้งที่ 3 มุมที่วัดได้ 97 องศา

4.5.4 การหมุนไปทางขวา

แนวความคิดที่ใช้เหมือนกับการหมุนไปทางซ้าย โดยค่าที่ใช้เป็นขอบเขตในการหมุน 90 องศาคือ 78 เหมือนกัน แต่ในการหมุนไปทางขวาจะเป็นการกำหนดให้กับการทำงานของเซอร์โวทางด้านซ้ายมือ

วิธีการทดลอง - ทำการทดลองในลักษณะเดียวกับการหมุนไปทางซ้าย กำหนดจุดเริ่มต้นโดยลากเส้นตรงขนานกับด้านหน้าและด้านข้างของตัวหุ่นยนต์ (ใช้จุดกึ่งกลางด้านหน้าของหุ่นยนต์และจุดกึ่งกลางของล้อหลังด้านขวาตามลำดับ เป็นหลัก) แล้วใช้จุดตัดที่เกิดจากเส้นตรงสองเส้นดังกล่าว เป็นจุดเริ่มต้น และใช้จุดกึ่งกลางของล้อหลังด้านขวาเป็นจุดหมุน จากนั้นให้ตัวหุ่นยนต์หมุนไปทางด้านขวาทำมุม 90 องศา ทำการกำหนดจุดสุดท้ายโดยใช้จุดตัดกันของเส้นตรงขนานกับด้านหน้าและด้านข้างของตัวหุ่นยนต์ แล้วทำการวัดมุมที่เกิดขึ้นจากการทดลอง

ผลการทดลอง - ในการหมุนทั้ง 3 ครั้งหุ่นยนตร์ยังคงเคลื่อนที่ไปอีกเล็กน้อยแม้ว่ามอเตอร์จะหยุดหมุนแล้ว

ครั้งที่ 1 มุมที่วัดได้ 92 องศา

ครั้งที่ 2 มุมที่วัดได้ 97 องศา

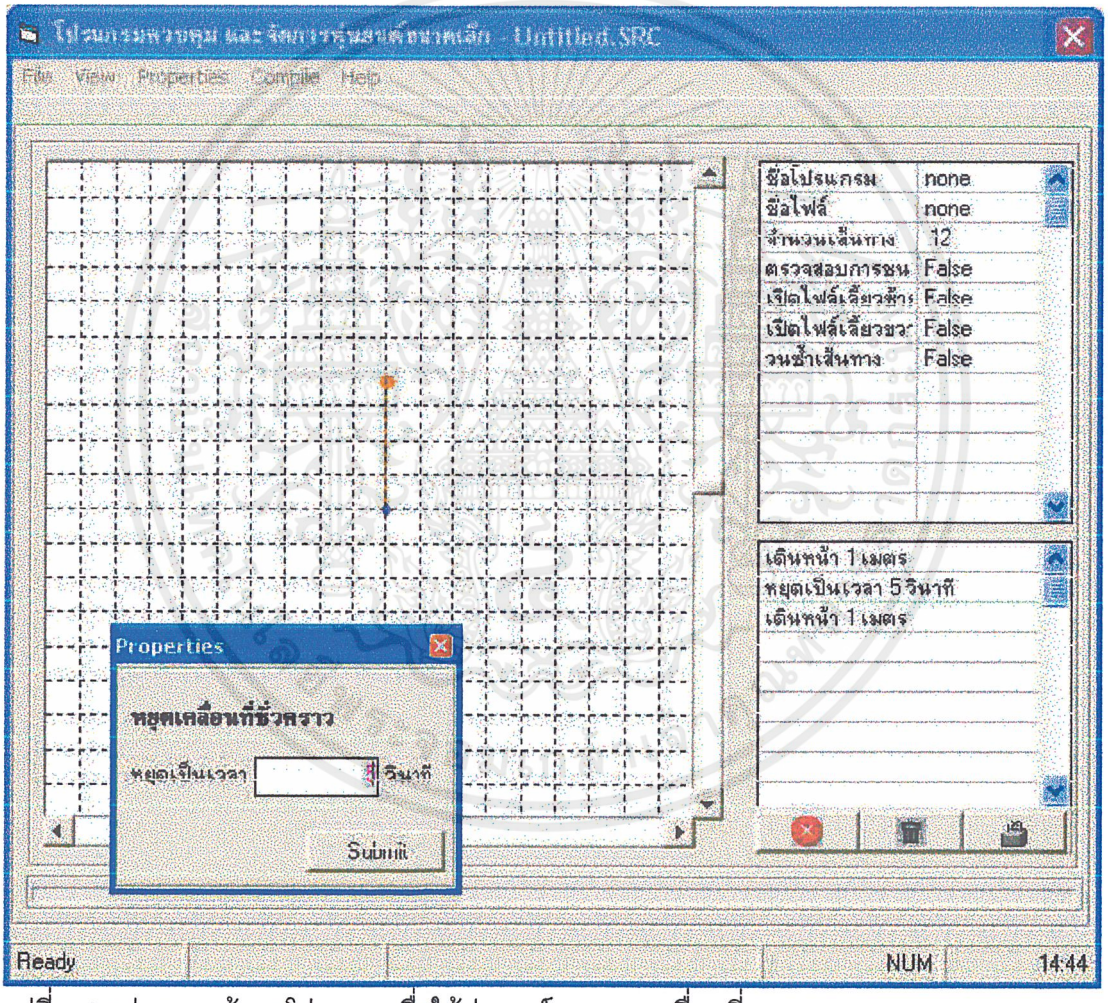
ครั้งที่ 3 มุมที่วัดได้ 95 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.5 การหยุดตามเวลาที่กำหนด

การหยุดจะทำงานโดยการกำหนดค่า PAUSE ตามด้วยค่าคงที่ค่าหนึ่งที่กำหนดให้เป็นระยะเวลาที่ใช้ในการหยุด จากการคำนวนในการหยุดการทำงานเป็นเวลาหนึ่งวินาที จะต้องกำหนดค่าคงที่นั้นเป็น 1000 เช่น ต้องการหยุดเป็นเวลา 5 วินาทีค่าคงที่ที่ใช้คือ 5000

วิธีการทดลอง - สั่งให้หุ่นยนต์เดินไปทางด้านหน้าเป็นระยะทาง 1 เมตร แล้วหยุดเป็นเวลา 5 วินาที จากนั้นจะเดินไปทางด้านหน้าอีก 1 เมตร ทำการจับเวลาตั้งแต่เริ่มต้นหยุดจนหุ่นยนตร์เคลื่อนที่อีกครั้ง



รูปที่ 4.6 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์หยุดการเคลื่อนที่

ผลการทดลอง – จับเวลาการหยุดได้ดังนี้

ครั้งที่ 1 เวลาที่วัดได้ 5.09 วินาที

ครั้งที่ 2 เวลาที่วัดได้ 5.24 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่ 3 เวลาที่วัดได้ 5.11 วินาที

4.5.6 ทดสอบการชน

เนื่องจากแสงจากสภาพแวดล้อมมีผลต่อการทำงานของตัวต้านทางแปรค่าตามแสงที่ติดตั้งไว้ทางด้านหน้าของหุ่นยนต์ ดังนั้นจึงทำการทดลองเพื่อหาค่าที่เหมาะสมสำหรับสภาพแวดล้อมตัวอย่างสองลักษณะ คือ แสงจากภายในห้อง และแสงธรรมชาติจากภายนอก เพื่อทำการปรับค่าให้กับตัวแปรที่ใช้ภายในโปรแกรม โดยตัวแปรที่มีความสำคัญในการทดลองนี้คือ CRASH_VALUE ซึ่งเป็นตัวแปรที่ใช้เป็นขอบเขตในการระบุว่าเกิดการชนขึ้นหรือยัง ถ้ามีค่ามากจะทำให้ความละเอียดในการตรวจจับการชนสูง ถ้ามีค่าน้อยจะทำให้มีความละเอียดในการตรวจจับการชนต่ำเมื่อพบว่ามีวัตถุขวางเส้นทางการเคลื่อนที่ หุ่นยนต์จะหยุดการทำงานทันที

วิธีการทดลอง - ทำการทดลองในสภาพแวดล้อมสองลักษณะคือ ในห้องปิดและภายนอก โดยกำหนดค่าเริ่มต้นของตัวแปร CRASH_VALUE ในการทดลองทั้งสองครั้งเป็น 100 ซึ่งเป็นค่าเริ่มต้นมาตรฐานที่ได้จากหนังสือคู่มือของหุ่นยนต์ จากนั้นจึงทำการปรับเปลี่ยนจนได้ค่า CRASH_VALUE ที่เหมาะสม

ผลการทดลอง

- ภายในห้องปิด

CRASH_VALUE = 100 เมื่อเปิดสวิตช์หุ่นยนต์ขยับไปทางด้านหน้าเพียงเล็กน้อยก่อนที่จะหยุดการทำงานทันที

CRASH_VALUE = 75 ขยับไปด้านหน้าเล็กน้อยก่อนหยุดการทำงาน

CRASH_VALUE = 50 เคลื่อนที่ไปทางด้านหน้าเรื่อย ๆ เมื่อเปิดสวิทช์ หยุดการเคลื่อนทีทันทีเมื่อนำเอาวัตถุไปขวางตัวตรวจจับทางซ้าย หรือทางขวา

- ภายนอกห้อง

CRASH_VALUE = 100 เมื่อเปิดสวิตช์หุ่นยนต์ขยับไปทางด้านหน้าเพียงเล็กน้อยก่อนที่จะหยุดการทำงานทันที

CRASH_VALUE = 75 ขยับไปด้านหน้าเล็กน้อยก่อนหยุดการทำงาน

CRASH_VALUE = 50 เคลื่อนที่ไปทางด้านหน้าเรื่อย ๆ หยุดการเคลื่อนที่เมื่อนำเอาวัตถุไปขวางตัวตรวจจับในระยะที่ใกล้มาก

CRASH_VALUE = 65 เคลื่อนที่ไปทางด้านหน้าเรื่อย ๆ เมื่อเปิดสวิทช์ หยุดการเคลื่อนทีทันทีเมื่อนำเอาวัตถุไปขวางตัวตรวจจับ

เมื่อดูจากผลการทดลองจะเห็นว่าเมื่ออยู่ภายนอกแสงจากรอบด้านมีผลทำให้ค่าที่ใช้ในการตรวจจับการชนเปลี่ยนไป เนื่องจากมีความสว่างมากขึ้นทำให้ตัวแปรที่ใช้ในการวัดค่าแสง

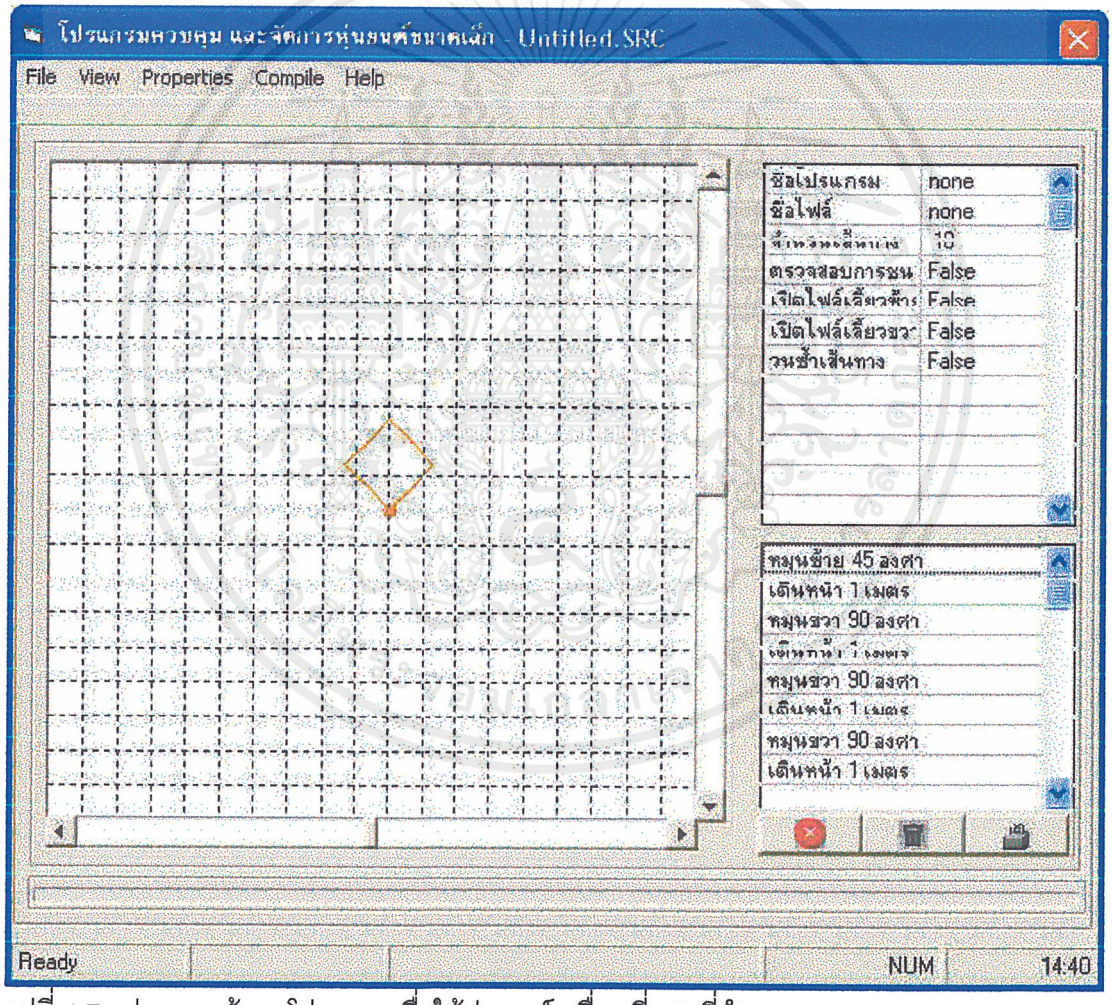
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีค่ามากขึ้นตามไปด้วย ดังนั้นค่า CRASH_VALUE ที่ใช้จึงเพิ่มขึ้นตาม

4.5.7 การเคลื่อนที่ตามเส้นทางที่กำหนด

กำหนดเส้นทางการเคลื่อนที่ให้กับหุ่นยนต์เพื่อตรวจสอบดูการทำงานโดยรวม เมื่อมีทั้งการเดินทางเป็นเส้นตรงและการหมุนรวมอยู่ด้วยกัน ถ้าการเคลื่อนที่ไม่คลาดเคลื่อน จุดเริ่มต้นกับจุดสิ้นสุดควรเป็นจุดเดียวกัน

วิธีการทดลอง - ให้จุดกึ่งกลางของล้อด้านหน้าเป็นจุดเริ่มต้น กำหนดเส้นทางให้กับหุ่นยนต์ในการเคลื่อนที่เมื่อเคลื่อนที่เสร็จสิ้นทำการวัดจุดกึ่งกลางของล้อหน้าเพื่อให้เป็นจุดสิ้นสุด แล้ววัดระยะห่างจากจุดเริ่มต้นจนถึงจุดสิ้นสุด



รูปที่ 4.7 รูปแสดงหน้าจอโปรแกรมเมื่อให้หุ่นยนต์เคลื่อนที่ตามที่กำหนด

เส้นทางการเคลื่อนที่ - มีลักษณะเป็นรูปสี่เหลี่ยมด้านเท่า โดยทำการเคลื่อนที่ตามลำดับดังนี้ หมุนซ้าย 45 องศา เคลื่อนที่ไปข้างหน้า 1 เมตร, หมุนขวา 90 องศา เคลื่อนที่ไปข้างหน้า 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตร หมุนขวา 90 องศา เคลื่อนที่ไปข้างหน้า 1 เมตร และสุดท้ายหมุนขวา 90 องศา เคลื่อนที่ไปข้างหน้า 1 เมตร

ผลการทดลอง – จากการสังเกตการเคลื่อนที่ของหุ่นยนตร์ ทั้งการเคลื่อนที่เป็นเส้นตรงและการหมุนมีความคลาดเคลื่อนที่มองเห็นได้ด้วยตา ทำให้จุดสุดท้ายเบี่ยงเบนไปจากจุดเริ่มต้นอย่างเห็นได้ชัด

ครั้งที่ 1 ระยะห่างที่วัดได้ 13 เซนติเมตร

ครั้งที่ 2 ระยะห่างที่วัดได้ 16 เซนติเมตร

ครั้งที่ 3 ระยะห่างที่วัดได้ 14 เซนติเมตร

4.6 สิ่งที่ได้จากการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็ก ทำให้ผู้พัฒนาได้ทราบถึงระบบ และวิธีการทำงานของหุ่นยนต์ รวมถึงกลไกในการควบคุมต่าง ๆ ซึ่งทำให้มีความเข้าใจวิธีการทำงานของหุ่นยนต์มากขึ้น รวมถึงการพัฒนาหุ่นยนต์นี้ทำให้ผู้ใช้สามารถที่จะเรียนรู้ถึงภาษา PBASIC ซึ่งเป็นภาษาที่ใช้ในการควบคุมไมโครคอนโทรเลอร์ BASICSTAMP IISX อีกด้วย

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กเป็นโปรแกรมที่ทำให้ผู้ใช้งานสามารถทำการใช้งานหุ่นยนต์ Robo-Kit ได้สะดวกมากขึ้น รวมถึงผู้ใช้งานสามารถที่จะกำหนดค่าต่าง ๆ ของหุ่นยนต์ได้โดยไม่จำเป็นต้องเรียนรู้ภาษา PBASIC และโครงสร้างการทำงานของหุ่นยนต์ในรายละเอียด ผู้สนใจสามารถทำการปรับปรุงและพัฒนาโปรแกรมต่อเพื่อให้การทำงานของโปรแกรมสอดคล้องกับการนำไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผล และข้อเสนอแนะ

5.1 สรุปผล

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กได้รับการพัฒนาขึ้นมาโดยมีวัตถุประสงค์เพื่อที่จะทำให้ผู้ที่ต้องการใช้งานหุ่นยนต์ เพื่อที่จะทำการกำหนดเส้นทางการเคลื่อนที่ รวมถึงกำหนดคุณสมบัติการเคลื่อนที่ของหุ่นยนต์นั้นสามารถทำได้โดยง่าย และไม่จำเป็นต้องเรียนรู้ถึงภาษาระดับต่ำ และโครงสร้างของหุ่นยนต์

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กได้รับการพัฒนาขึ้นโดยอาศัยการศึกษาถึงโครงสร้างการทำงานของหุ่นยนต์ องค์ประกอบทางกายภาพของหุ่นยนต์ รวมถึงภาษาที่ใช้ในการควบคุมหุ่นยนต์ ซึ่งจำเป็นต้องใช้เวลาในการเรียนรู้ถึงโครงสร้างต่าง ๆ จึงจะสามารถทำการพัฒนาโปรแกรมได้อย่างมีประสิทธิภาพ

ในการพัฒนาโปรแกรมจะทำการพัฒนาโดยอาศัยข้อมูลจากลักษณะความต้องการใช้งานของผู้ใช้ โดยจะทำการพิจารณาถึงรูปแบบ และขั้นตอนการทำงานของหุ่นยนต์ควบคู่ไปกับความสามารถของหุ่นยนต์ ซึ่งผู้พัฒนาจะทำการกำหนดกิจกรรม และขั้นตอนที่ผู้ใช้สามารถใช้งานโปรแกรมได้ และหลังจากนั้นจึงทำการพัฒนาโปรแกรม ซึ่งจะทำให้ได้โปรแกรมที่ตรงกับความต้องการของผู้ใช้งานมากที่สุด

โปรแกรมจัดการ และควบคุมหุ่นยนต์ขนาดเล็กจะมีฟังก์ชั่นการทำงานต่าง ๆ ซึ่งผู้ใช้สามารถเข้าใจได้โดยง่าย เพราะใช้สัญลักษณ์ต่าง ๆ เป็นรูปภาพ และคำพูดต่าง ๆ ที่สื่อความหมายกับผู้ใช้โดยตรง และจะมีการป้องกันความผิดพลาดที่เกิดจากการกรอกข้อมูลผิดพลาด โดยโปรแกรมจะทำการเตือนผู้ใช้งาน เมื่อผู้ใช้กรอกข้อมูลที่ผิดพลาด

หลังจากการพัฒนาโปรแกรมในแต่ละส่วน เช่นส่วนของการเคลื่อนที่ในทิศทางต่าง ๆ แล้ว ผู้พัฒนาจะทำการทดลองใช้งานโปรแกรมควบคู่กับหุ่นยนต์ โดยจะทำการวัดค่าคลาดเคลื่อนที่ได้จากการทดลอง และนำค่าต่าง ๆ มาใช้ในการปรับปรุงโปรแกรม ซึ่งจะทำให้ได้ค่าที่ตรงกับความเป็นจริงมากที่สุด หลังจากการทดลองใช้งานโปรแกรมแล้ว ทำให้สามารถสรุปถึงข้อจำกัดต่าง ๆ ทั้งของโปรแกรม และของตัวหุ่นยนต์ได้ดังนี้คือ

ในกรณีที่เป็นข้อจำกัดของโปรแกรมที่ได้ทำการพัฒนาขึ้น

- โปรแกรมที่ทำการพัฒนาขึ้นมา สามารถนำไปใช้กับหุ่นยนต์ Robo-Kit โดยต้องมีคุณสมบัติตรงกับที่กำหนดไว้ในบทที่ 4 เท่านั้น ไม่สามารถนำมาใช้กับหุ่นยนต์อื่น ๆ หรือที่มีคุณสมบัติต่างกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมสามารถที่จะกำหนดเส้นทางของหุ่นยนต์ได้ในพื้นระนาบเดียวกันเท่านั้น ไม่สามารถที่จะนำไปใช้กับพื้นที่อยู่ต่างระดับกันได้
- โปรแกรมไม่สามารถทำการบันทึก Source Code ลงไมโครคอนโทรเลอร์ได้โดยตรง จะต้องทำการบันทึกผ่านโปรแกรม BasicStamp เท่านั้น
- ค่าคงที่ต่าง ๆ ที่ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ จะต้องทำการกำหนดภายในตัว Code ของโปรแกรมเท่านั้น ผู้ใช้ไม่สามารถกำหนดค่าต่าง ๆ เหล่านั้นได้จากหน้าจอการใช้งานโปรแกรม
- User Interface ที่ได้ทำการพัฒนาขึ้นยังไม่สมบูรณ์มากนัก ในบางส่วนยังทำให้ผู้ใช้สับสนในการใช้งานได้

ในส่วนของข้อจำกัดของหุ่นยนต์ สามารถสรุปออกมาได้ดังนี้คือ

- ในการพัฒนาหุ่นยนต์จะต้องทำการดัดแปลงหุ่นยนต์ก่อนจึงจะสามารถใช้งานได้จริง ซึ่งในขั้นตอนการประกอบ และดัดแปลงนี้จำเป็นต้องอาศัยความชำนาญ มิเช่นนั้นอาจทำให้หุ่นยนต์ไม่สามารถเคลื่อนที่ได้
- สภาพแวดล้อมต่าง ๆ มีผลต่อการเคลื่อนที่ของหุ่นยนต์ ซึ่งไม่สามารถทำให้หุ่นยนต์สามารถเคลื่อนที่ได้อย่างถูกต้องในทุกสภาพแวดล้อมได้ โดยสามารถสรุปสภาพแวดล้อมที่มีผลต่อการเคลื่อนที่ของหุ่นยนต์ได้ดังนี้คือ สภาพพื้นผิวที่เป็นจุดหมุนของล้อ ถ้าพื้นผิดลื่น หรือว่าฝืดเกินไป จะทำให้เกิดการไถลในการหมุนแต่ละครั้ง ทำให้ได้องศาการหมุนไม่ตรงกับที่กำหนด และสภาพความเข้มของแสดงซึ่งมีผลกระทบต่อค่าที่ใช้ในการตรวจจับการชน
- สำหรับพลังงานที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ จะมาจากเพียง battery เท่านั้น ไม่สามารถใช้พลังงานจากแหล่งอื่นได้ ซึ่งทำให้เป็นการสิ้นเปลือง

การพัฒนาโปรแกรมดังกล่าวทำให้ผู้ที่ทำการพัฒนาได้รับประโยชน์ต่าง ๆ มากมาย ได้แก่ ความรู้ถึงโครงสร้าง และส่วนประกอบต่าง ๆ ของหุ่นยนต์ ,ความรู้ถึงภาษาที่ใช้ในการควบคุมหุ่นยนต์ ซึ่งทำให้เข้าใจถึงโครงสร้าง และการทำงานของหุ่นยนต์รวมถึงภาษาระดับต่ำมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

หลังจากการทดลองใช้งานโปรแกรม และได้ทำการสรุปผลถึงข้อผิดพลาดรวมถึงข้อจำกัดต่าง ๆ ที่เกิดขึ้นแล้ว คณะผู้พัฒนาโปรแกรมได้ทำการพิจารณา และสามารถสรุปผลออกมาเป็นข้อเสนอแนะได้ดังนี้ คือ

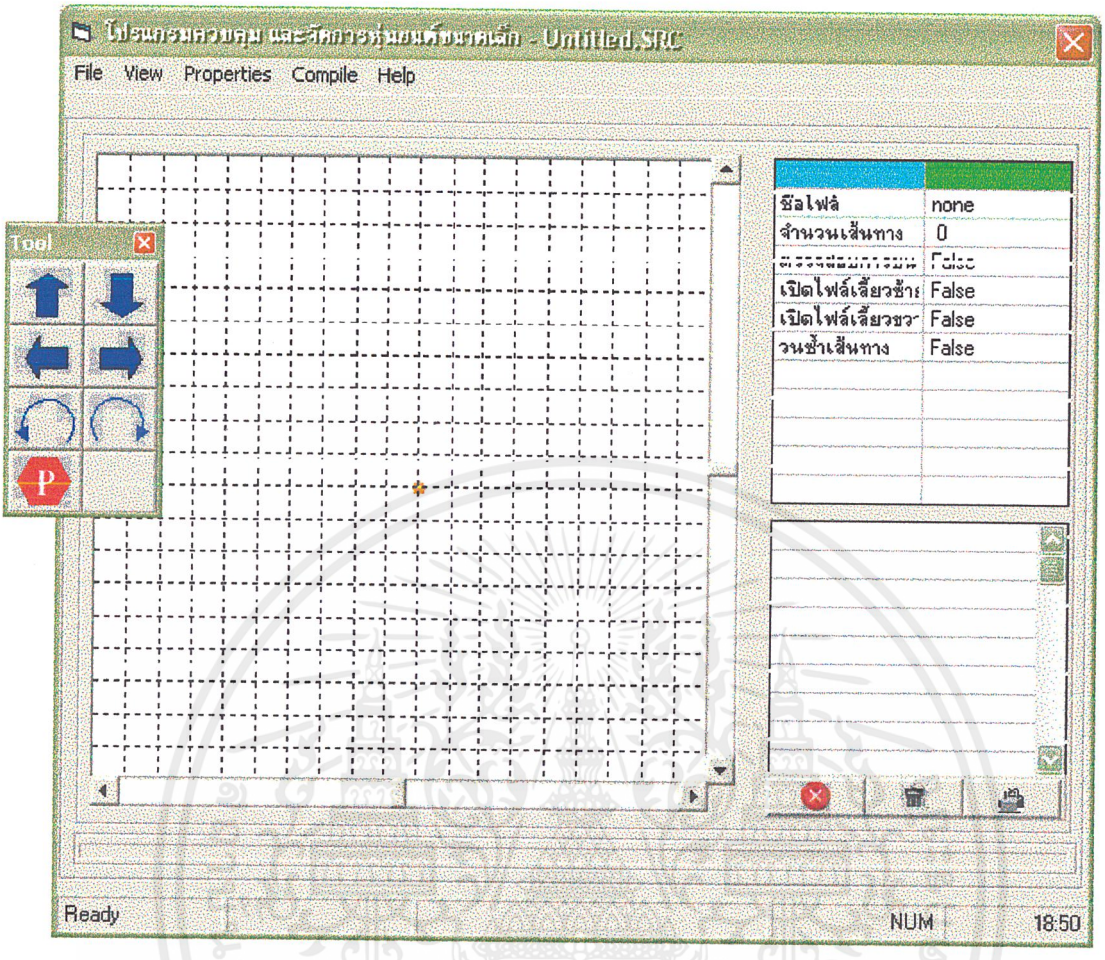
- ผู้ใช้โปรแกรมควรที่จะจัดระนาบของการเคลื่อนที่ให้เป็นระนาบเดียวกันทั้งหมด
- ผู้พัฒนาในขั้นต่อไป ควรที่จะสามารถทำให้โปรแกรมสามารถที่จะบันทึกข้อมูลลงไมโครคอนโทรเลอร์ได้โดยตรง
- ควรที่จะให้ผู้ใช้สามารถที่จะกำหนดค่าคงที่ต่าง ๆ ได้แก่ ค่าที่ใช้เปรียบเทียบองศาการหมุน ค่าที่ใช้เปรียบเทียบระยะทางการเคลื่อนที่ของหุ่นยนต์ เพื่อที่ผู้ใช้ไม่จำเป็นต้องไปปรับเปลี่ยนในตัวโปรแกรม
- ควรที่จะทำการปรับปรุงหน้าจอ User Interface ในส่วนต่าง ๆ ให้มีพื้นที่ใช้งานมากขึ้น โดยทำการตัดในส่วนที่ไม่จำเป็นทิ้งไป
- ควรที่จะทำการปรับปรุงโครงสร้างของตัวหุ่นยนต์ให้มีรูปแบบการทำงานที่มากขึ้น เพื่อเพิ่มประสิทธิภาพการทำงานของหุ่นยนต์
- ควรที่จะทำการปรับสภาพของยางล้อให้สามารถเคลื่อนที่ได้สมบูรณ์ในทุกสภาพผิว
- ควรที่จะสามารถบังคับการเคลื่อนที่จากในระยะไกลในลักษณะ online ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
ตัวอย่างหน้าจอในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 หน้าจอหลักของโปรแกรม



รูปที่ A.1 รูปภาพแสดงหน้าจอหลักของโปรแกรม

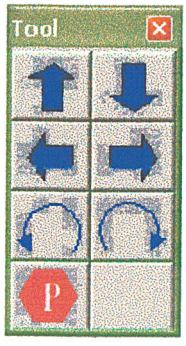
ภายในหน้าจอหลัก จะเป็นหน้าจอที่ผู้ใช้ทำการใช้งานโปรแกรม โดยผู้ใช้จะทำการกำหนดลักษณะเส้นทาง รวมถึงลักษณะการเดินทางของหุ่นยนต์ โดยภายในหน้าจอหลัก โดยส่วนตรงกลางจะเป็นส่วนที่ใช้สำหรับการแสดงเส้นทางการเคลื่อนที่ของหุ่นยนต์

ตารางทางด้านมุมขวาบน จะใช้สำหรับแสดงคุณสมบัติต่าง ๆ ของการเคลื่อนที่ของหุ่นยนต์ รวมถึงแสดงชื่อไฟล์ที่บันทึก รวมถึงเมื่อเรียกใช้งานไฟล์เก่าด้วย

ตารางทางด้านมุมขวาล่าง ใช้สำหรับแสดงรายการเส้นทางการเคลื่อนที่ของหุ่นยนต์ทั้งหมด ซึ่งสามารถลบ หรือล๊อคเส้นทางได้ ผ่านทางปุ่ม 3 ปุ่มข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าจอเครื่องมือควบคุม



รูปที่ A.2 รูปแสดงหน้าจอเครื่องมือควบคุม

เป็นส่วนที่ใช้สำหรับการกำหนดทิศทางเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยภายในหน้าจอนี้จะประกอบไปด้วยส่วนควบคุมลักษณะการเดินทางทั้งหมด 7 เส้นทางได้แก่

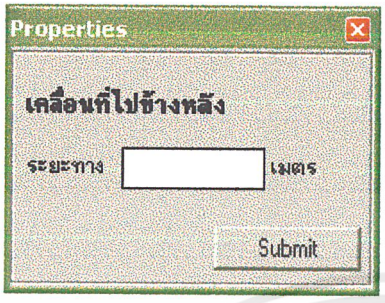
- เคลื่อนที่ไปข้างหน้า
- เคลื่อนที่ไปข้างหลัง
- เคลื่อนที่ไปทางซ้าย
- เคลื่อนที่ไปทางขวา
- หมุนซ้าย
- หมุนขวา
- หยุดการเคลื่อนที่ชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หน้าจอกำหนดค่าเส้นทาง

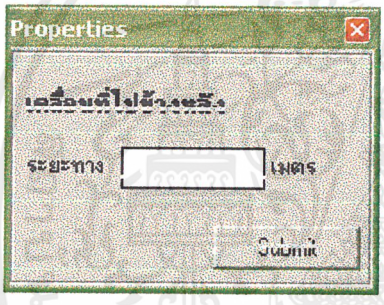
หน้าจอนี้ จะสามารถแบ่งออกเป็นหน้าจอย่อย ๆ ได้ทั้งหมด 7 หน้าจอ ตามประเภทการใช้งาน ได้แก่

3.1 ในกรณีที่ผู้ใช้ต้องการเคลื่อนที่ไปข้างหน้า



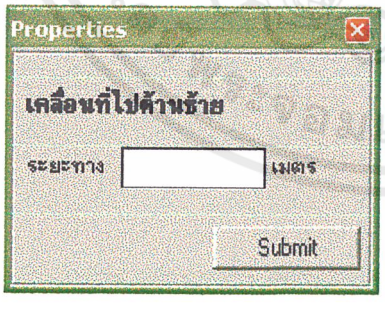
รูปที่ A.3 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปข้างหน้า

3.2 ในกรณีที่ผู้ใช้ต้องการเคลื่อนที่ไปข้างหลัง



รูปที่ A.4 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปข้างหลัง

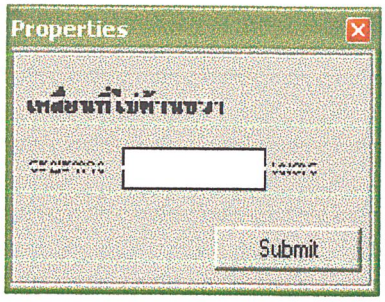
3.3 ในกรณีที่ผู้ใช้ต้องการเคลื่อนที่ไปด้านซ้าย



รูปที่ A.5 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปด้านซ้าย

3.4 ในกรณีที่ผู้ใช้ต้องการเคลื่อนที่ไปด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

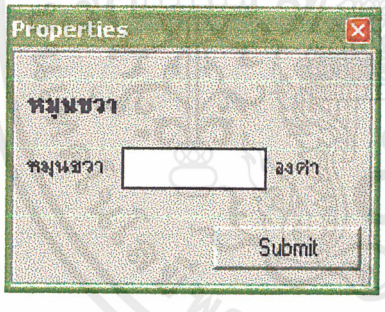


รูปที่ A.6 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่เคลื่อนที่ไปด้านขวา

3.5 ในกรณีที่ผู้ใช้ต้องการหมุนขวา

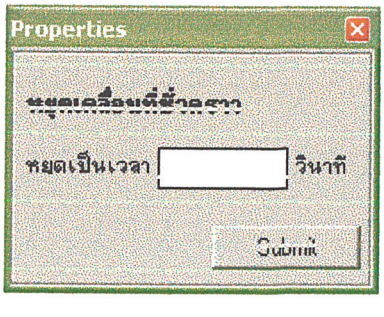
รูปที่ A.7 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หมุนขวา

3.6 ในกรณีที่ผู้ใช้ต้องการหมุนขวา



รูปที่ A.8 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หมุนขวา

3.7 ในกรณีที่ผู้ใช้ต้องการหยุดการเคลื่อนที่ชั่วคราว

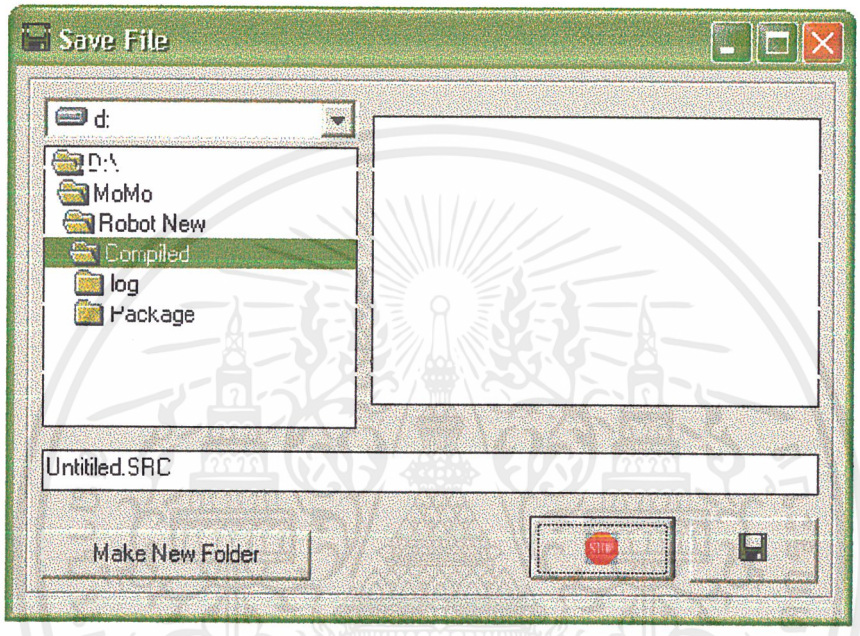


รูปที่ A.9 รูปภาพแสดงหน้าจอกำหนดค่าเส้นทางในกรณีที่หยุดการเคลื่อนที่ชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอกำหนดค่าเส้นทางเป็นหน้าจอที่ใช้สำหรับกำหนดค่าที่ใช้ในการเคลื่อนที่ โดยค่าที่กำหนดนี้จะสัมพันธ์กับทิศทางที่ผู้ใช้เลือกในหน้าจอเครื่องมือควบคุม

4. หน้าจอบันทึกข้อมูล

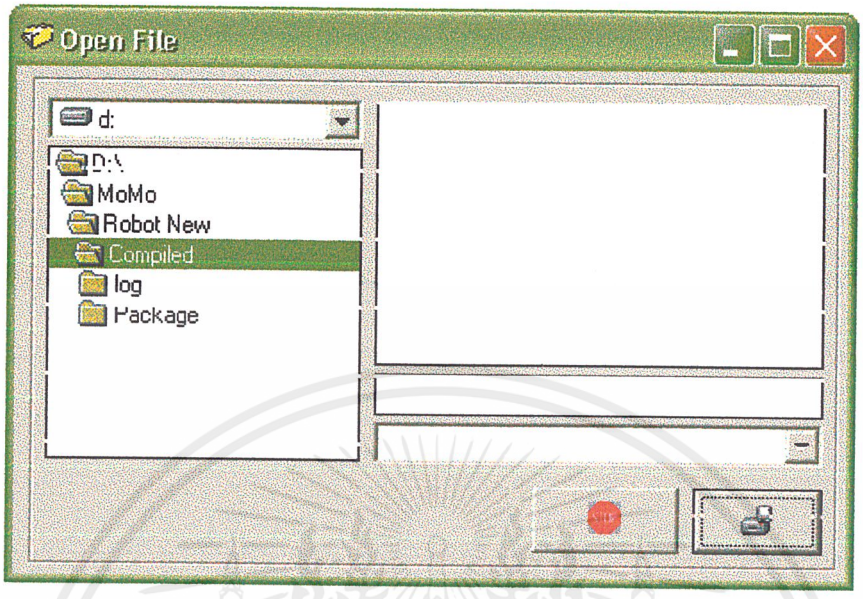


รูปที่ A.10 รูปแสดงหน้าจอบันทึกข้อมูล

หน้าจอบันทึกข้อมูล เป็นหน้าจอสำหรับการบันทึกเส้นทางการเดินทางของหุ่นยนต์ ที่ผู้ใช้ได้กำหนดขึ้น รวมถึงคุณสมบัติต่าง ๆ ของโปรแกรมที่ผู้ใช้กำหนด ให้เป็นไฟล์ เพื่อที่ผู้ใช้จะสามารถเรียกใช้เส้นทางที่บันทึกไว้ได้ในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หน้าจอเรียกใช้งานข้อมูล

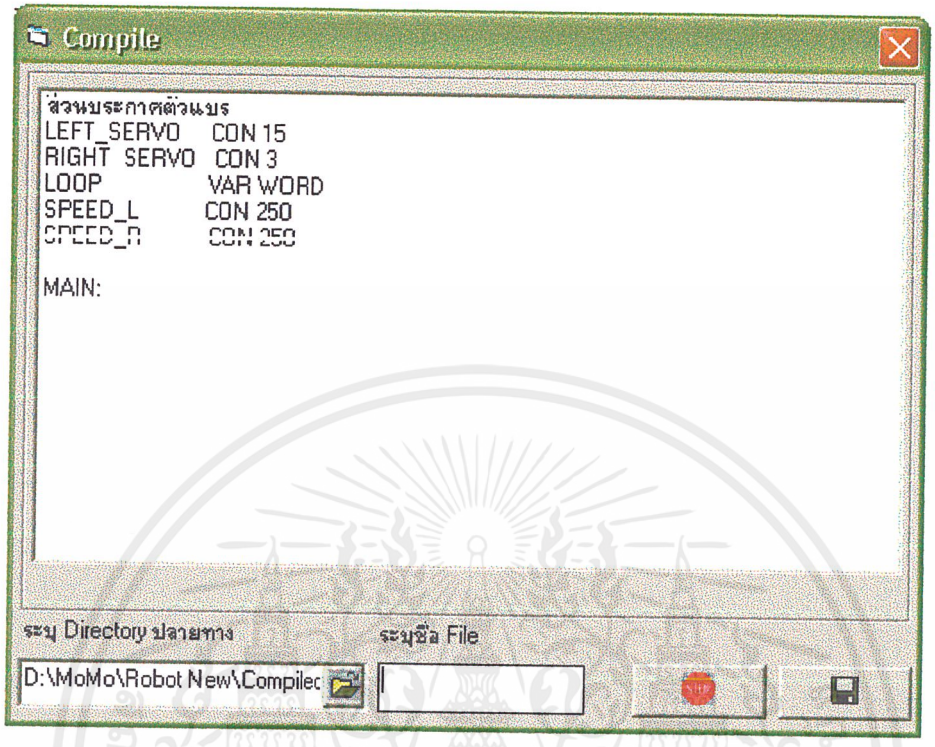


รูปที่ A.11 รูปแสดงหน้าจอบันทึกข้อมูล

เป็นส่วนที่ใช้สำหรับเรียกใช้งานข้อมูลที่ผู้ใช้เคยบันทึกไว้แล้ว โดยในการเรียกใช้งานโปรแกรมจะทำการนำเอาเส้นทาง รวมถึงคุณสมบัติต่าง ๆ ที่ผู้ใช้กำหนดไว้มาแสดงทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หน้าจอ Compile โปรแกรม



รูปที่ A.12 รูปแสดงหน้าจอ Compile โปรแกรม

เป็นส่วนที่ใช้งานสำหรับการบันทึก Source Code ภาษา PBASIC ที่ถูก Compile แล้วไปเป็นไฟล์ที่สามารถนำไปบันทึกลงไมโครคอนโทรเลอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. หน้าจอกำหนดค่าโปรแกรม

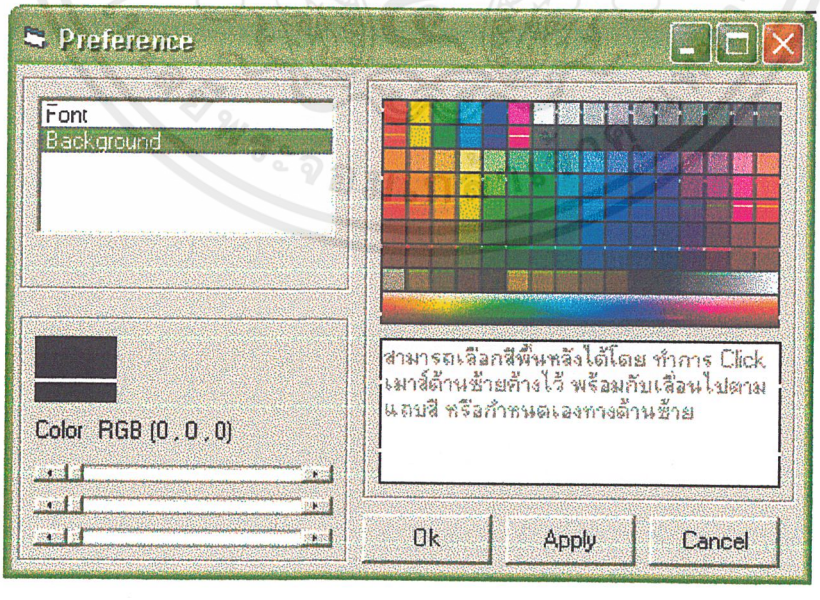
ในหน้าจอกำหนดค่าโปรแกรมจะประกอบไปด้วยหน้าย่อย ๆ 2 ส่วนคือ

- กรณีที่ผู้ใช้ต้องการปรับค่าตัวอักษร
- กรณีที่ผู้ใช้ต้องการปรับค่าสีพื้นหลัง

7.1 กรณีที่ผู้ใช้ต้องการปรับค่าตัวอักษร

รูปที่ A.13 รูปแสดงหน้าจอกำหนดค่าตัวอักษร

7.2 กรณีที่ผู้ใช้ต้องการปรับค่าสีพื้นหลัง



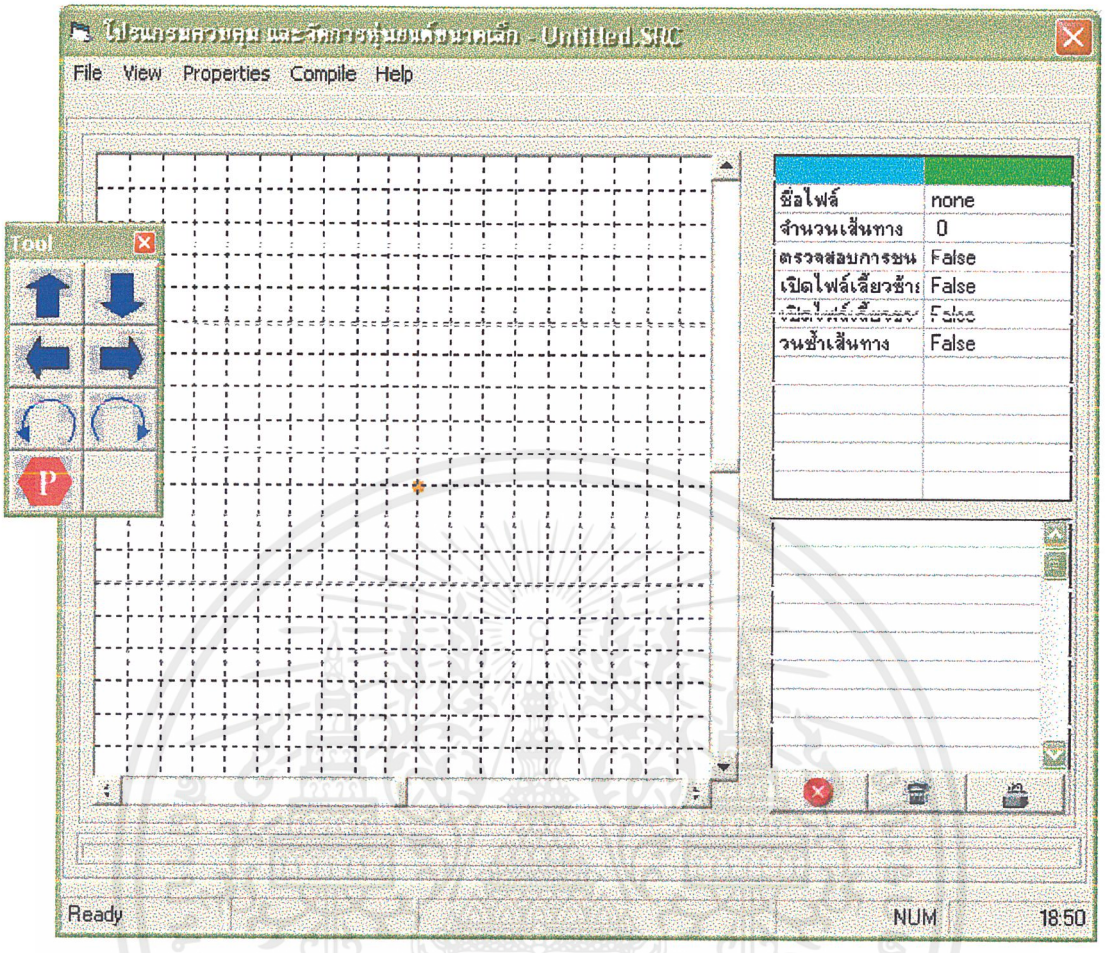
รูปที่ A.14 รูปแสดงหน้าจอกำหนดค่าค่าสีพื้นหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
ทดสอบการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

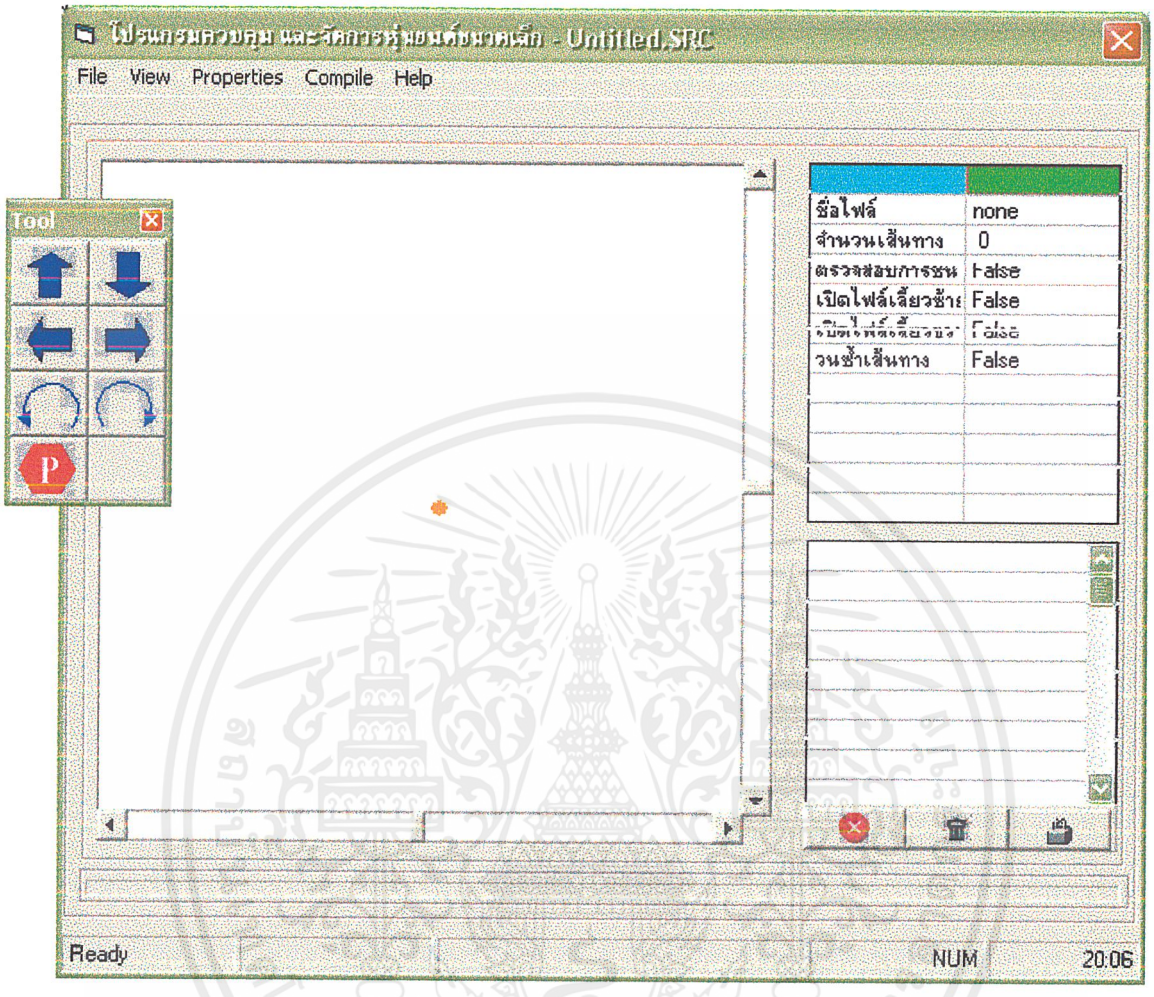
1. เมื่อผู้ใช้ทำการเรียกใช้โปรแกรม โปรแกรมจะทำการแสดงหน้าจอหลักขึ้นมาก่อน



รูปที่ B.1 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากการเรียกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

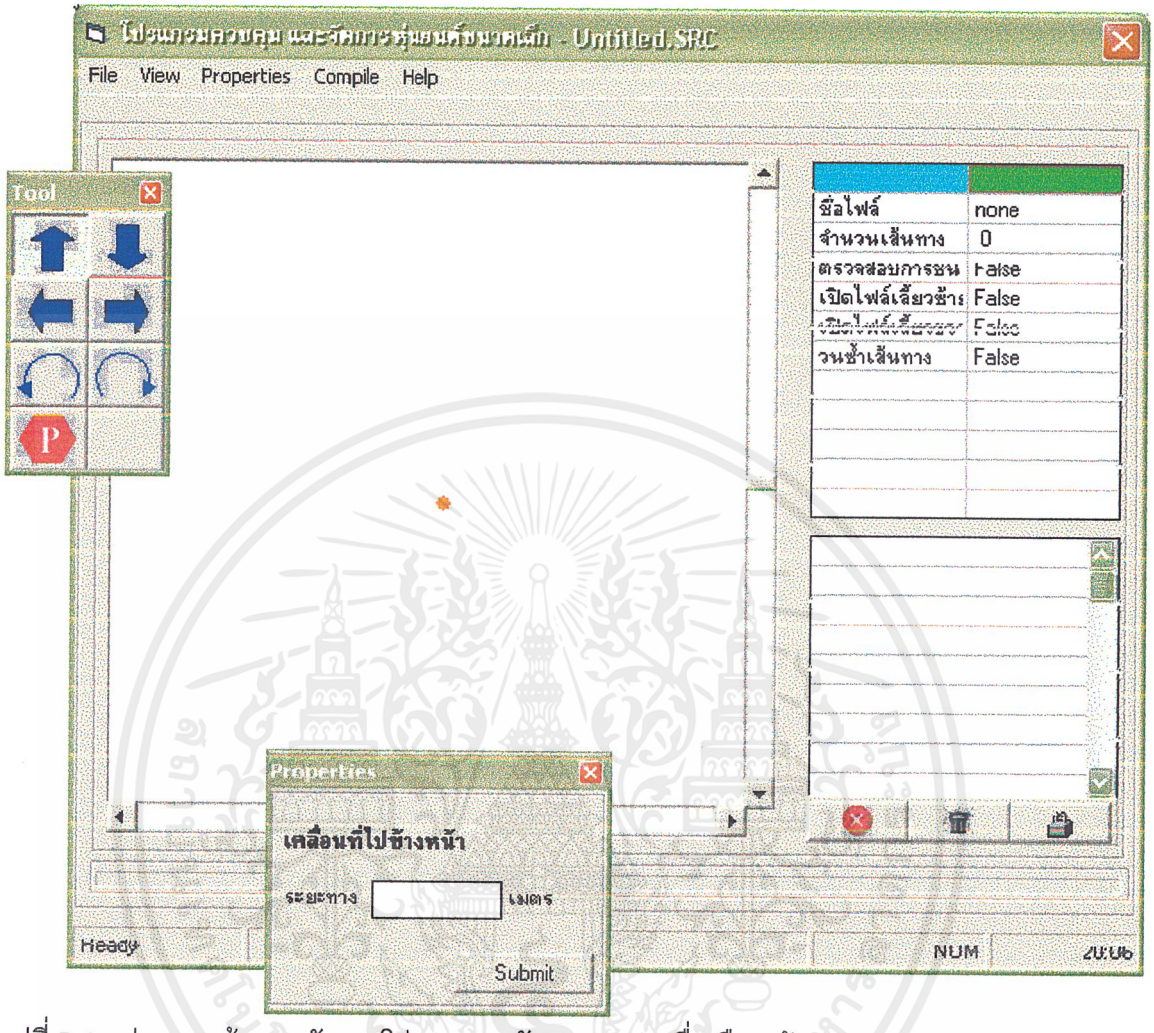
2. เมื่อผู้ใช้ทำการ Click ที่เมนู Properties --> Grid โปรแกรมจะทำการซ่อนเส้น grid ในหน้าจอหลัก



รูปที่ B.2 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากการซ่อนเส้น grid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

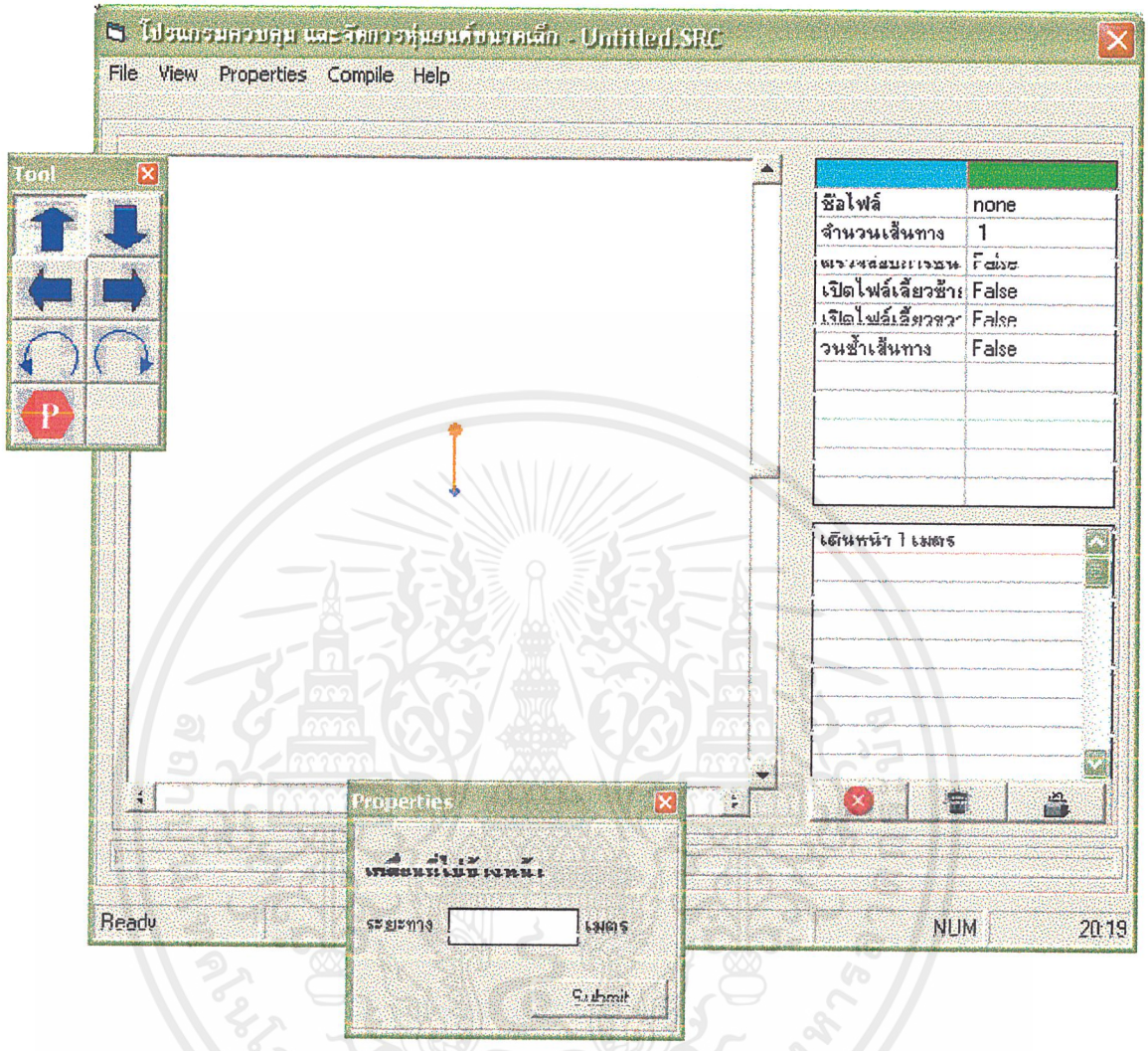
3. ผู้ใช้ทำการ click ที่ปุ่มเคลื่อนที่ไปข้างหน้า โปรแกรมจะทำการแสดงหน้าจอกำหนดค่าเส้นทางสำหรับการเคลื่อนที่ไปข้างหน้า



รูปที่ B.3 รูปแสดงหน้าจอหลักของโปรแกรม หลังจาก click เพื่อเลือกเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ผู้ใช้ทำการใส่ค่าระยะทางที่ต้องการให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า โดยใส่ค่าระยะทางเป็น 1 และทำการ click ที่ปุ่ม submit หรือกด Enter

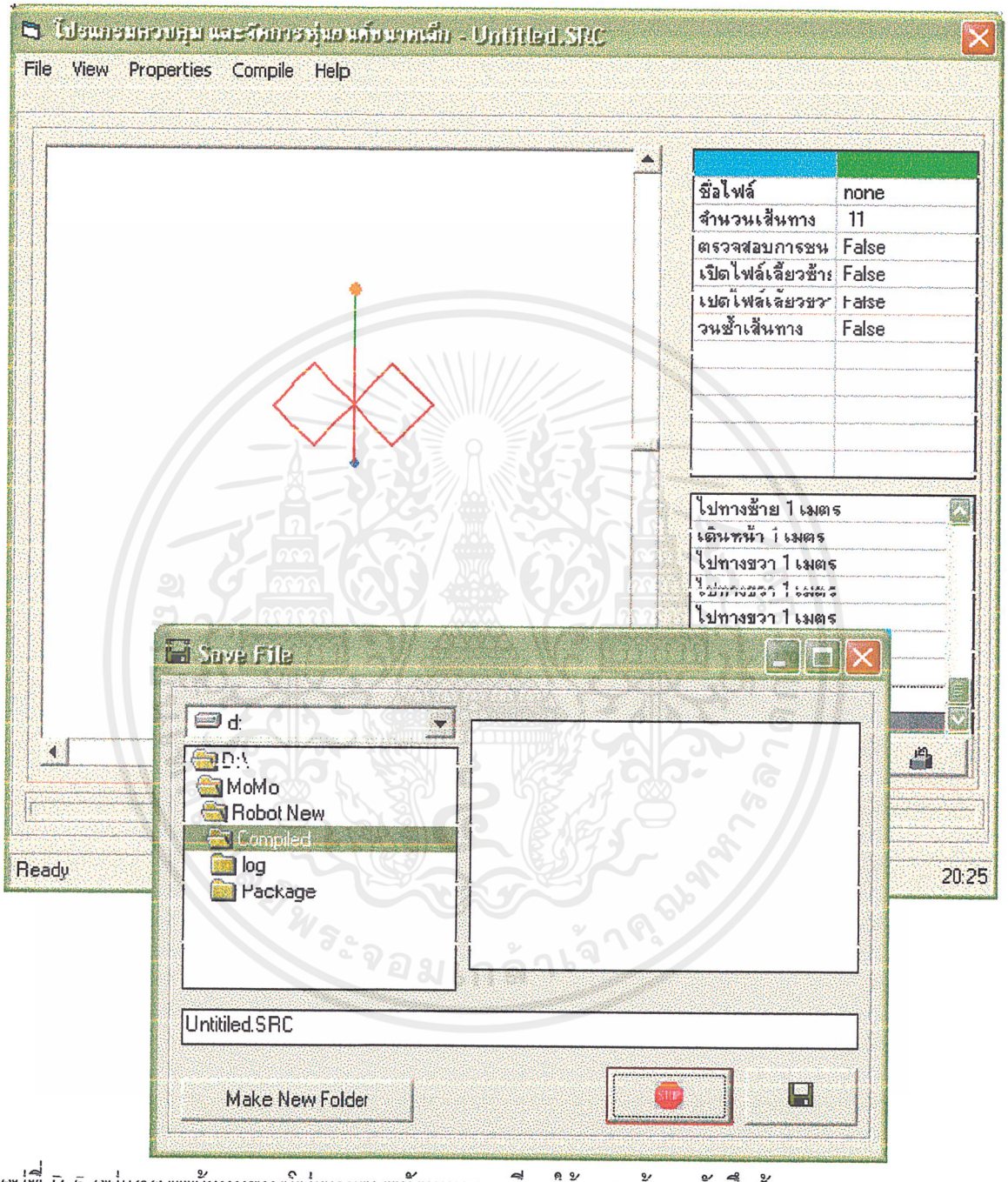


รูปที่ B.4 รูปแสดงหน้าจอหลักของโปรแกรม หลังจากบันทึกเส้นทางการเคลื่อนที่ของหุ่นยนต์

สังเกตได้ว่า ส่วนที่มีการเปลี่ยนแปลงคือ จำนวนเส้นทางจะเพิ่มขึ้นอีก 1 เส้นทาง และตารางด้านล่างจะมีการบันทึกเส้นทางการเดินทางเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

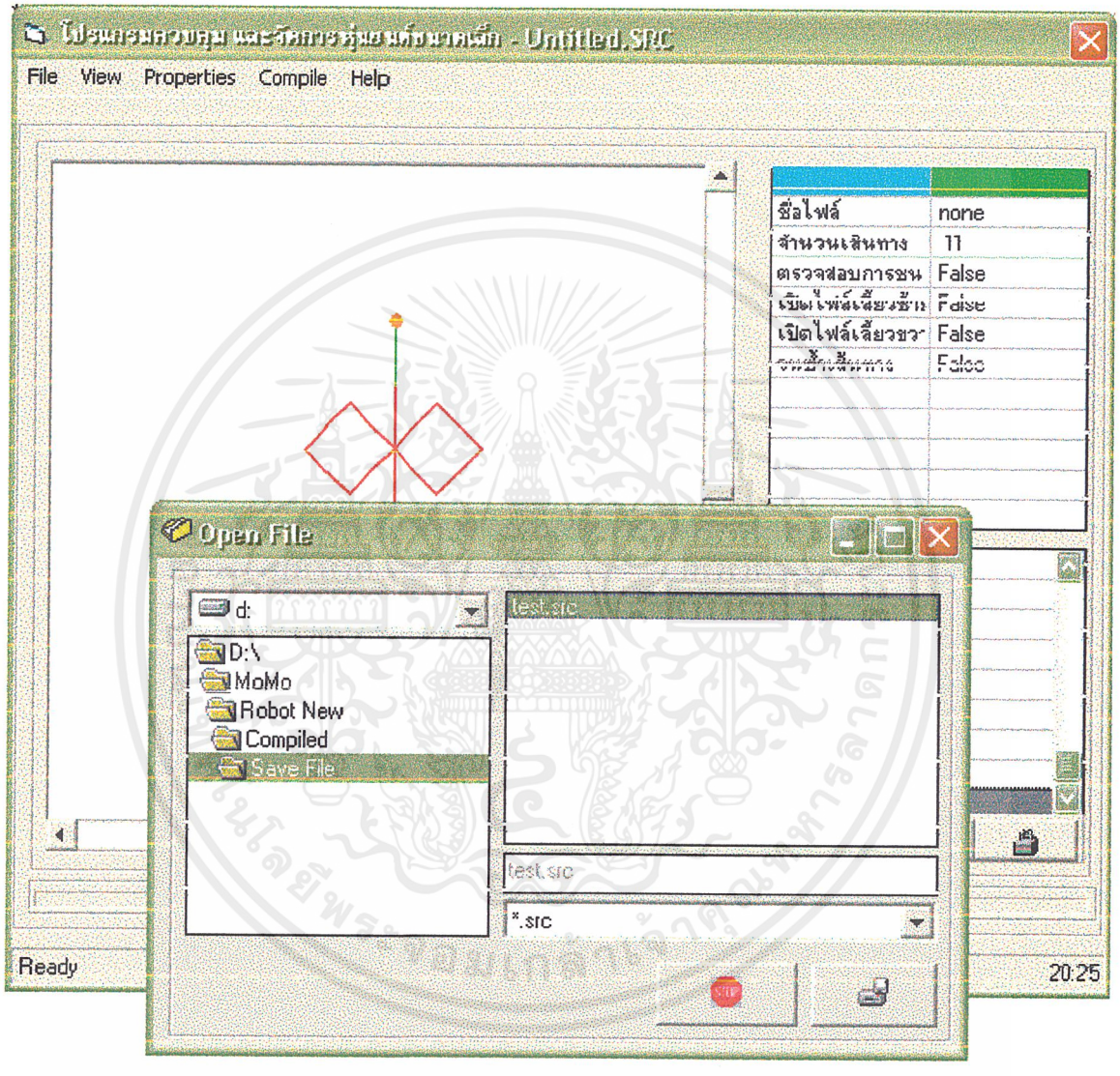
5. หลังจากที่ผู้ใช้โปรแกรม ทำการกำหนดเส้นทางเรียบร้อยแล้ว ในกรณีที่ผู้ใช้ต้องการบันทึกเส้นทางการเคลื่อนที่ของโปรแกรม สามารถทำได้โดยเลือกที่เมนู File --> Save หรือกด Ctrl + s โปรแกรมจะทำการแสดงหน้าจอบันทึกเส้นทางขึ้นมา



รูปที่ B.5 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

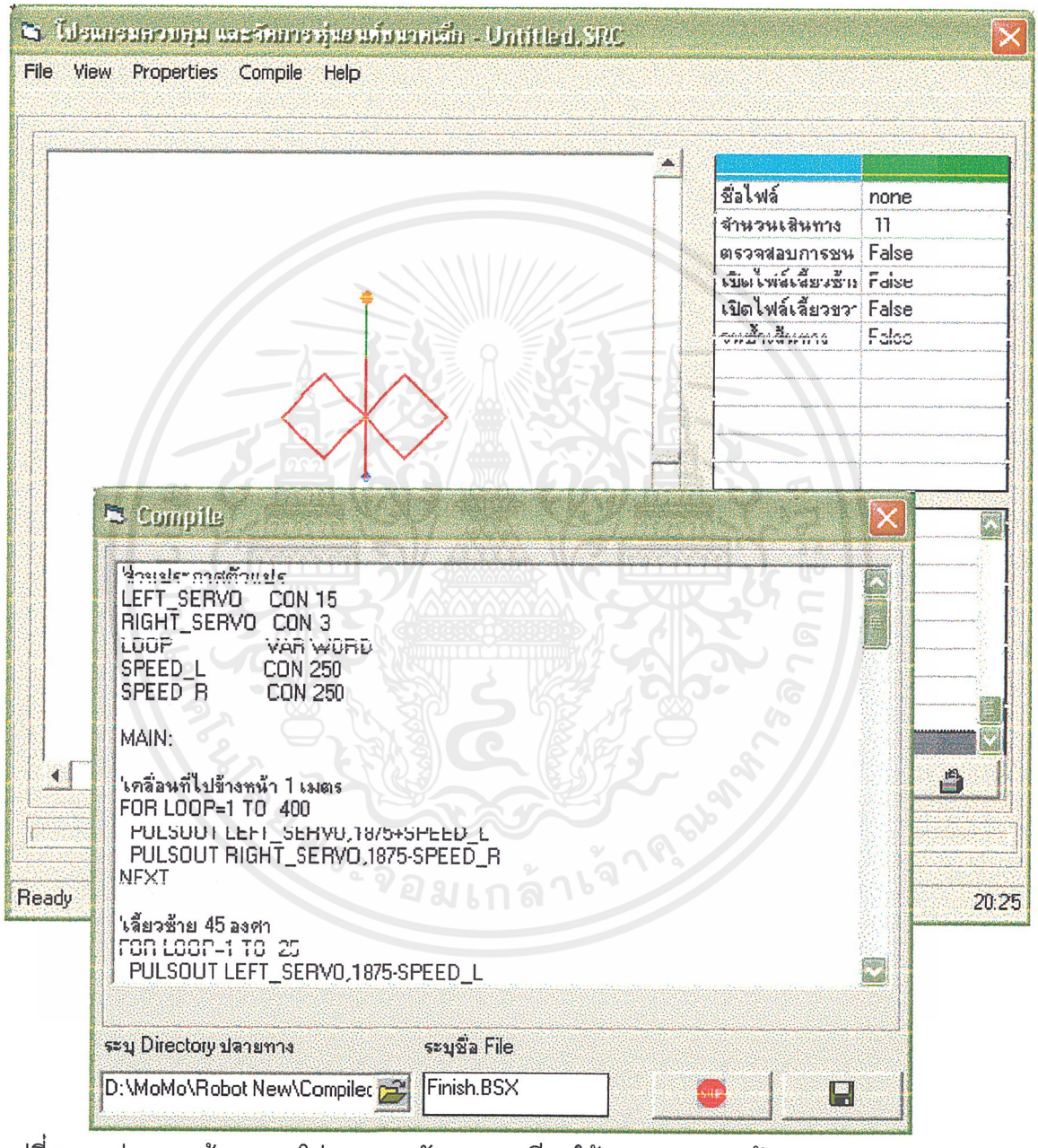
6. จากนั้นผู้ใช้ทำการบันทึกไฟล์ใน path ที่ต้องการ และในกรณีที่ผู้ใช้ต้องการเรียกใช้ไฟล์นั้นผู้ใช้สามารถทำได้โดยการเลือกที่ File --> Open หลังจากนั้นโปรแกรมจะทำการแสดงหน้าจอเรียกใช้งานข้อมูลขึ้นมา ผู้ใช้จะทำการเลือกไฟล์ที่ต้องการเรียกใช้งาน และกดที่ปุ่มเรียกใช้งาน โปรแกรมจะทำการเรียกใช้งานไฟล์ที่ต้องการ



รูปที่ B.6 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอเรียกใช้งานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

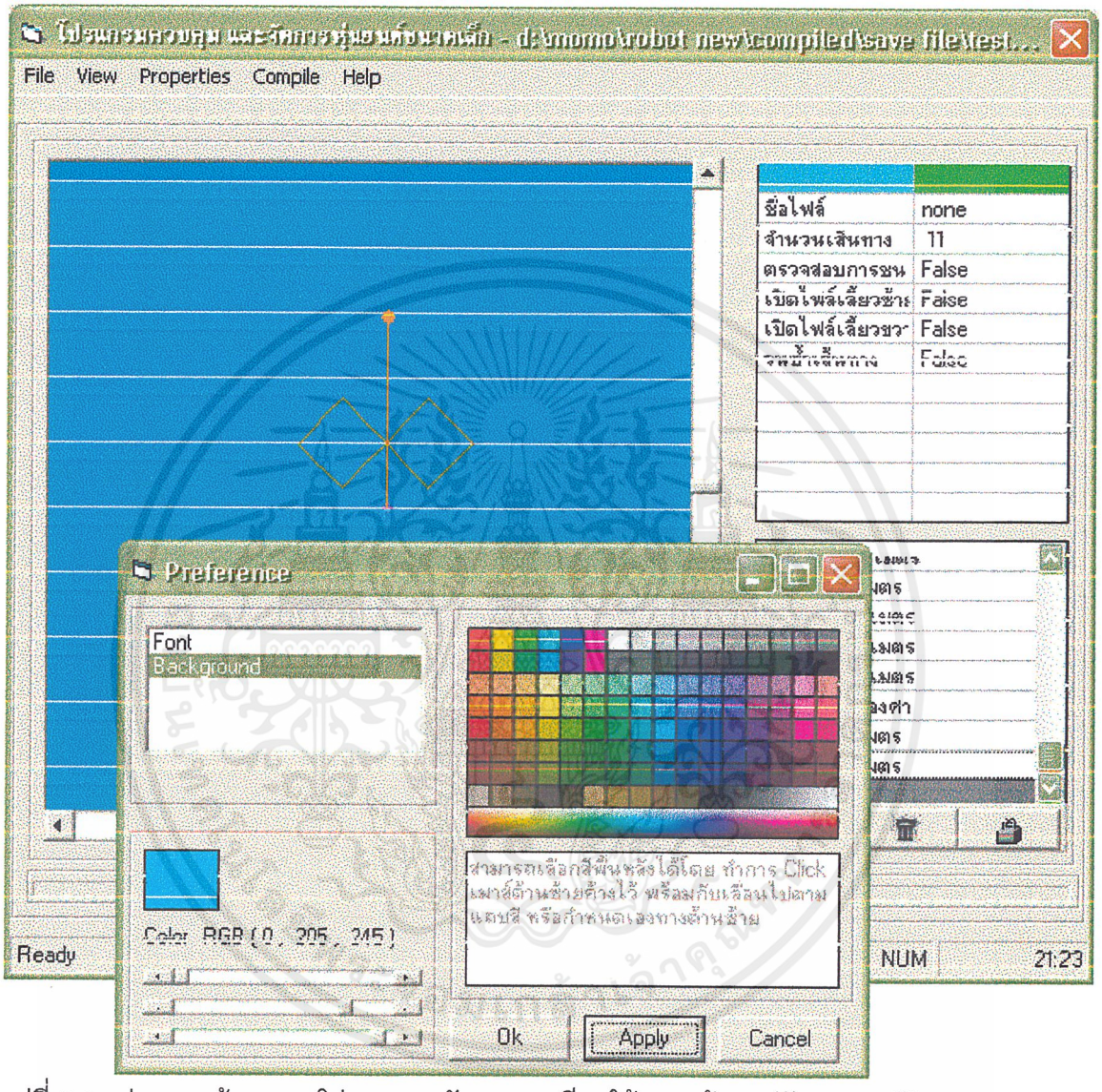
7. ในกรณีที่ผู้ใช้จะทำการ compile โปรแกรม ผู้ใช้จะทำการเลือกที่เมนู Compile --> Compile หรือกด F5 โปรแกรมจะทำการแสดงหน้าจอ Compile เส้นทางขึ้นมา ผู้ใช้จะทำการกำหนด path ที่ต้องการบันทึก หลังจากนั้นจะทำการกดบันทึก ผู้ใช้ก็จะได้ไฟล์ที่บันทึก source code ของภาษา PBASIC ที่สามารถนำไปบันทึกลงหุ่นยนต์ได้



รูปที่ B.7 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งาน Compile ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

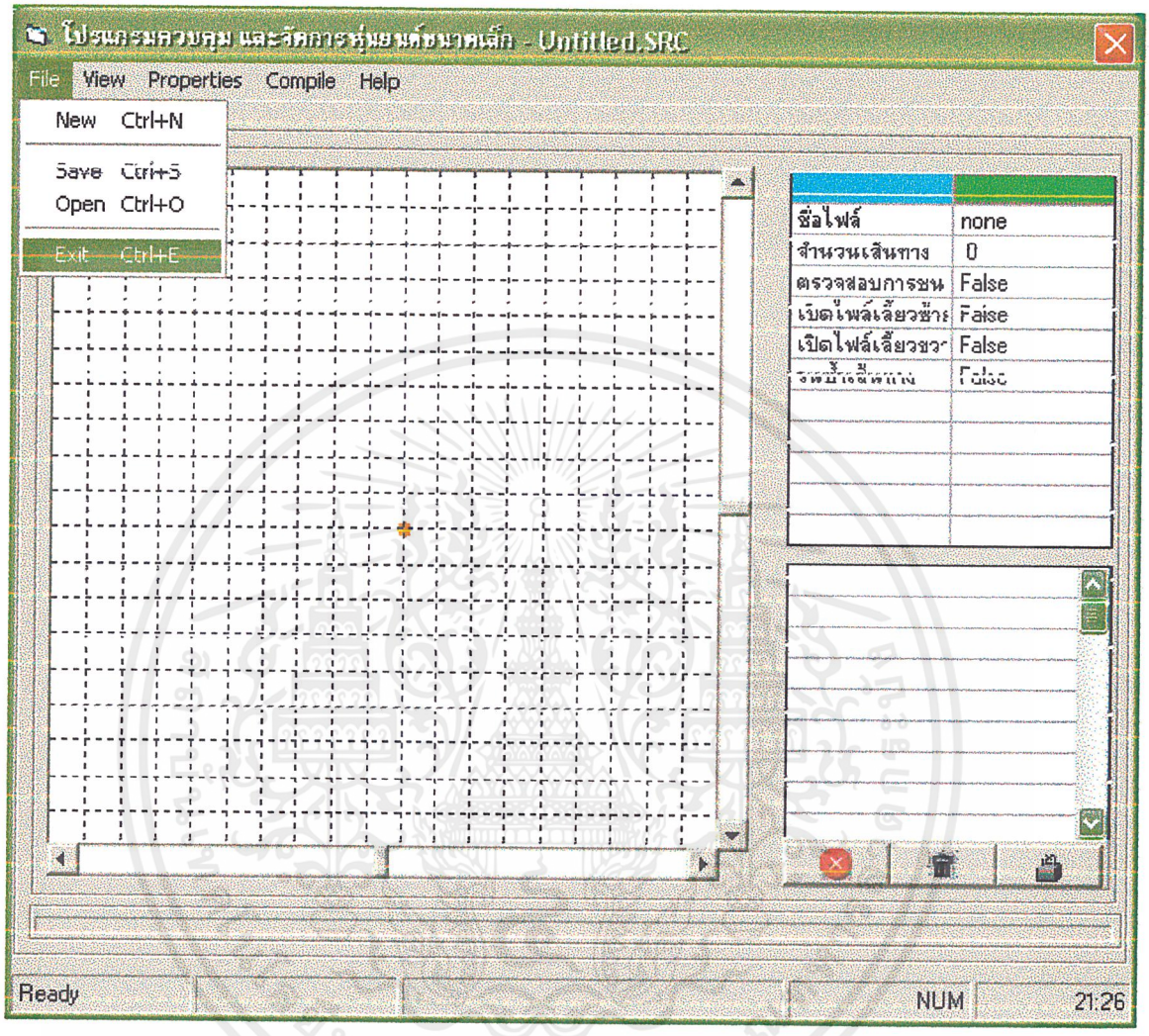
8. ในกรณีที่ผู้ใช้ต้องการปรับปรุงคุณสมบัติของโปรแกรม สามารถทำได้โดย Click ที่ปุ่ม Properties --> Preference โปรแกรมจะทำการแสดงหน้าจอกำหนดค่าโปรแกรม ผู้ใช้สามารถที่จะปรับเปลี่ยนคุณสมบัติของโปรแกรมได้จากหน้าจอนี้



รูปที่ B.8 รูปแสดงหน้าจอของโปรแกรม หลังจากการเรียกใช้งานหน้าจอปรับคุณสมบัติของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เมื่อผู้ใช้ต้องการออกจากโปรแกรม สามารถทำได้โดย Click ที่เมนู File --> Exit หรือกด Ctrl+x จะเป็นการปิดโปรแกรมทั้งหมด



รูปที่ B.9 รูปแสดงหน้าจอของโปรแกรม เมื่อต้องการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และจำลอง ครูอุตสาหะ. 2543. **VisualBasic 6 ฉบับโปรแกรมเมอร์**. พิมพ์ครั้งที่ 7. กรุงเทพฯ : บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด

คู่มือการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก : Innovative Experiment Co.,Ltd.

ณภัทร บุญสมบูรณ์. 2545. **ลึกแต่ไม่ลับกับ Visual Basic**. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท ซีเอ็ดยูเคชัน จำกัด (มหาชน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้