ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

หุ่นยนต์ตรวจจับเส้นทาง

LINE SENSING ROBOT

โดย

ธนชัย หาญอิทธิกุล

นที วุฒิกุลวานิช

อาจารย์ที่ปรึกษา

ดร. โอพาร วงศ์วิรัตน์

อาจารย์ที่ปรึกษาร่วม

อาจารย์ อนันตพัฒน์ อนันตชัย

2/ท.
ส138ห
2550

เลขหมู่.....
เลขทะเบียน..... 04764
วัน,เดือน,ปี..... - 7 ต.ค. 2551

b.11976967.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ตรวจจับเส้นทาง

LINE SENSING ROBOT

ธนชัย หาญอิทธิกุล

นที วุฒิกุลวานิช

อาจารย์ที่ปรึกษา

ดร.โอพาร วงศ์วิรัตน์

อาจารย์ที่ปรึกษาร่วม

อาจารย์ อนันตพัฒน์ อนันตชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LINE SENSING ROBOT

TANACHAI HANIDTHIKUL

NATEE VUTHIKULVANICH

A PROJECT SUMMITTED IN PARTIAL FULLFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2/2007

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้มีไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2550

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ตรวจจับเส้นทาง

LINE SENSING ROBOT

ผู้จัดทำ

1. นาย ธนชัย หาญอิทธิกุล รหัสประจำตัว 47070013

2. นาย นที วุฒิกุลวานิช รหัสประจำตัว 47070018

.....**อาจารย์ที่ปรึกษา**

(ดร. โอฬาร วงศ์วิรัตน์)

.....**อาจารย์ที่ปรึกษาร่วม**

(อาจารย์ อนันตพัฒน์ อนันตชัย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	หุ่นยนต์ตรวจจับเส้นทาง	
นักศึกษา	นาย ธนชัย หาญอิทธิกุล	รหัสนักศึกษา 47070013
	นาย นที วุฒิกุลวานิช	รหัสนักศึกษา 47070018
ปริญญา	วิทยาศาสตรบัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2550	
อาจารย์ผู้ควบคุมโครงงาน	ดร.โอพาร วงศ์วิรัตน์	
อาจารย์ผู้ควบคุมโครงงานร่วม	อ.อนันตพัฒน์ อนันตชัย	

บทคัดย่อ

ปัจจุบัน การพัฒนาระบบฝังตัว (Embedded System) ที่เป็นมาตรฐาน ซึ่งจะช่วยให้การพัฒนาเป็นไปอย่างมีลำดับขั้นตอน และตอบสนองต่อความต้องการในการพัฒนาระบบอย่างมีทิศทางนั้น ยังไม่ได้รับการส่งเสริมเท่าที่ควร นอกจากนี้การใช้แนวทางการพัฒนาระบบเพื่อเข้ามาช่วยในการพัฒนาแอปพลิเคชั่นระบบฝังตัว ก็ยังไม่ได้ถูกนำมาประยุกต์ใช้อย่างแพร่หลายมากนัก

โครงงานฉบับนี้ จึงนำเสนอแนวทางการพัฒนาระบบเชิงวัตถุ (Object-Oriented Analysis and Design Methodology) เพื่อเข้ามาใช้ในการพัฒนาระบบฝังตัว "หุ่นยนต์ตรวจจับเส้นทาง (Line Sensing robot)" โดยใช้เครื่องมือในการพัฒนาแบบยูเอ็มแอล (UML: Unified Modeling Language) เพื่อให้การพัฒนาเป็นไปอย่างมีมาตรฐาน และช่วยในการพัฒนาโปรแกรมที่เหมาะสมสำหรับระบบฝังตัว โครงงานนี้ใช้ชุดทดลองเลโก้ (Lego) รุ่นอาร์ซีเอ็กซ์ (RCX) เป็นฮาร์ดแวร์หลัก และใช้โปรแกรมภาษาซี++ (C++) เป็นเครื่องมือในการพัฒนาหุ่นยนต์ตรวจจับเส้นทาง หลักการทำงานคือหุ่นยนต์สามารถเคลื่อนที่ผ่านสิ่งกีดขวางที่กำหนด เพื่อไปถึงจุดสิ้นสุดที่เป็นเป้าหมายได้ โดยใช้อุปกรณ์เซนเซอร์แสง (Light Sensor) ทำหน้าที่ในการตรวจจับพื้นผิวสนามแต่ละประเภท เพื่อกำหนดการเคลื่อนที่ของหุ่นยนต์ในรูปแบบต่างๆ ทั้งการเคลื่อนที่บนเส้นทางปกติ การเร่งความเร็วและชะลอความเร็วบนทางชัน การเคลื่อนที่ไปในเขตต่างๆของสนาม และการเคลื่อนที่ภายในขอบเขตสนามที่กำหนด รวมทั้งใช้อุปกรณ์เซนเซอร์สัมผัส (Touch Sensor) ทำหน้าที่ในการตรวจจับสิ่งกีดขวางระหว่างการเคลื่อนที่ไปยังเป้าหมาย เพื่อให้หุ่นยนต์สามารถหลบหลีกเมื่อมีการสัมผัสกับสิ่งกีดขวางดังกล่าวเกิดขึ้นการทำงานของหุ่นยนต์จะเสร็จสมบูรณ์เมื่อตรวจจับเจอจุดสิ้นสุดที่เป็นเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project	Line Sensing Robot
Student	Mr. Thanachai Hanidhikul Student ID. 47070013
	Mr. Natee Vuthikulvanich Student ID. 47070018
Degree	Bachelor of Science
Programme	Information Technology
Year	2007
Project Advisor	Dr. Olarn Wongwirat
Project Co-Advisor	Mr. Anuntapat Anuntachai

ABSTRACT

Recently, standard development of embedded system, which leads to the step by step development and fulfills the demand of systematic development in the right direction, has not been sufficiently promoted. Besides, the use of system development approach to improve the application development of embedded system has not been widely applied.

This project introduces an Object-Oriented Analysis and Design Methodology that is applied to the development of embedded system, i.e., "Line Sensing Robot." By using UML (Unified Modeling Language) as a tool, it can be ensured that the development can meet the standard and support the development of suitable program for embedded system. The project uses Lego model RCX as the main hardware and employs C++ language programming as a tool for the development of line sensing robot. Basically, the robot can move across the defined obstacles and can approach to the end point, as a final target, by using a light sensor. The function of light sensor is to detect individual surface of the tested field in order to define different types of movement, i.e., moving on a regular surface, increasing and reducing speed over the slop, moving through each tested field, and moving within the boundary of defined area. In addition, the robot uses touch sensors to detect the obstacles during its movement to the defined target. By this mechanism, the robot can avoid the obstacles when hitting them. The robot operation will be completed when the end point is detected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์สำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร.โอพาร วงศ์วิรัตน์ และ อาจารย์ที่ปรึกษาร่วม อาจารย์อนันตพัฒน์ อนันตชัย คอยให้ความช่วยเหลือ แนะแนวทางต่างๆ และช่วยแก้ไขปัญหาต่างๆที่เกิดขึ้น รวมถึงแนะแนวคิดต่างๆที่ได้ถ่ายทอดมา ทำให้เกิดแนวทางในการทำปริญญานิพนธ์โครงงานนี้ ตลอดทุกช่วงเวลา ทางคณะผู้จัดทำได้ทราบซึ้งและ ขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณ คณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารฯลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง รวมทั้งได้แนะแนวคิดต่างๆ ในการทำปริญญานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณ เจ้าหน้าที่ฝ่ายทะเบียนทุกๆท่านที่ได้แนะนำและอำอวยความสะดวกในการดำเนินงานต่างๆที่เกี่ยวข้องกับการทำปริญญานิพนธ์ ทั้งทางด้านโครงงานและด้านเอกสารต่างๆ

ขอขอบคุณ บิดา มารดา ญาติมิตร ที่คอยสนับสนุนและเป็นกำลังใจให้ตลอดในการทำปริญญานิพนธ์โครงงานนี้

ขอขอบคุณและขอบใจ เพื่อนๆ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารฯลาดกระบัง รุ่นที่ 2 ที่คอยให้กำลังใจและความห่วงใยกันตลอดมา ไม่ว่าเหตุการณ์จะดีหรือจะร้าย ยามทุกข์หรือยามสุขก็ตาม

ธนชัย หาญอิทธิกุล

นที วุฒิกุลวานิช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการศึกษา.....2

1.3 ทฤษฎีที่เกี่ยวข้อง.....2

1.4 ขอบเขตของโครงงาน.....3

1.5 ขั้นตอนการศึกษา.....3

1.6 ประโยชน์ที่ได้รับ.....4

บทที่ 2 ทฤษฎีบท.....5

2.1 ระบบฝังตัว.....5

2.1.1 องค์ประกอบทางด้านฮาร์ดแวร์..... 5

2.1.1.1 เอ็มพียู..... 6

2.1.1.2 หน่วยความจำ.....7

2.1.1.3 บัส..... 7

2.1.1.4 อินพุตเอาท์พุตพื้นฐาน.....8

2.1.1.5 เทคโนโลยีอินเทอร์รัพท์.....9

2.1.1.6 เทคโนโลยีดีเอ็มเอ..... 10

2.1.1.7 เทคโนโลยีแคช.....11

2.1.1.8 เทคโนโลยีหน่วยความจำเสมือน..... 12

2.1.2 องค์ประกอบทางด้านซอฟท์แวร์.....13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.1.2.1 ระบบปฏิบัติการ 13
2.1.2.2 แอพพลิเคชั่น 15
2.2 พัฒนาระบบ 15
2.2.1 รูปแบบของวงจรการพัฒนาระบบ 15
2.2.1.1 วงจรการพัฒนาระบบแบบทำซ้ำ 15
2.2.1.2 วงจรการพัฒนาระบบแบบน้ำตก 16
2.2.1.3 วงจรการพัฒนาระบบแบบน้ำตกที่เหมาะสม 16
2.2.1.4 วงจรการพัฒนาระบบแบบวิวัฒนาการ 17
2.2.1.5 วงจรการพัฒนาระบบแบบเพิ่มขึ้น 17
2.2.1.6 วงจรการพัฒนาระบบแบบก้นหอย 18
2.2.2 แนวทางการพัฒนาระบบที่เลือกใช้ 19
2.2.3 การจัดการความต้องการ 19
2.2.4 ยูเอ็มแอล 20
2.2.4.1 แผนภาพยูสเคส 21
2.2.4.2 แผนภาพคลาส 23
2.2.4.3 แผนภาพอ็อบเจกต์ 26
2.2.4.4 แผนภาพลำดับเหตุการณ์ 27
2.2.4.5 แผนภาพปฏิสัมพันธ์ 28
2.2.4.6 แผนภาพกิจกรรม 28
2.2.4.7 แผนภาพสถานะ 30
2.2.5 การตรวจสอบคุณภาพ 30
2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ 30
2.3.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ 31
2.3.1.1 อาร์ซีเอ็กซ์ 31
2.3.1.2 องค์ประกอบของอาร์ซีเอ็กซ์ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3.2 ซอฟท์แวร์ที่ใช้ในการพัฒนาระบบ..... 38
2.3.2.1 บริคโอเอส..... 38
2.3.2.2 คอมไพล์เลอร์แบบไขว้..... 38
บทที่ 3 การพัฒนาระบบหุ่นยนต์ตรวจจับเส้นทาง..... 40
3.1 ความต้องการของระบบ..... 40
3.1.1 หน้าที่ของระบบ..... 40
3.1.2 ส่วนประกอบของระบบ..... 40
3.2 การออกแบบระบบงาน..... 41
3.2.1 แผนภาพยูสเคส..... 41
3.2.2 แผนภาพกิจกรรม..... 50
3.2.3 แผนภาพคลาส..... 59
3.2.4 แผนลำดับเหตุการณ์..... 60
3.2.5 แผนภาพการปฏิสัมพันธ์..... 63
3.2.6 แผนภาพสถานะ..... 65
3.3 การออกแบบหุ่นยนต์ตรวจจับเส้นทางและสนามทดลอง..... 66
3.3.1 การออกแบบหุ่นยนต์ตรวจจับเส้นทาง..... 66
3.3.1.1 การประกอบหุ่นยนต์ตรวจจับเส้นทาง..... 68
3.3.2 การออกแบบสนามทดสอบ..... 72
3.3.2.1 การออกแบบเส้นแบ่งเขตสนาม..... 74
3.3.2.2 การออกแบบทางขึ้นทางลง..... 75
บทที่ 4 การทดลองและผลการทดลอง..... 77
4.1 การทดลองบนทางเรียบธรรมดา..... 77
4.2 การทดลองบนทางที่มีสิ่งกีดขวาง..... 80
4.3 การทดลองบนทางที่มีทางชัน..... 83
4.4 การทดลองการทำงานของหุ่นยนต์..... 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปโครงงานและข้อเสนอแนะ....90

5.1 สรุปโครงการ....90

5.2 ปัญหาและอุปสรรค....90

5.2.1 ปัญหาทางด้านฮาร์ดแวร์....90

5.2.2 ปัญหาทางด้านซอฟท์แวร์....91

5.2.3 ปัญหาด้านการเขียนโปรแกรม....91

5.3 ข้อเสนอแนะ....92

บรรณานุกรม....93

ภาคผนวก ก....95

ภาคผนวก ข....102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ประเภทของแผนภาพในยูเอ็มแอล....21

2.2 ความสัมพันธ์ของปุ่ม 4 ปุ่มบนอาร์ซีเอ็กซ์....33

2.3 ความสัมพันธ์ระหว่างช่องทางเข้าและสายส่งออก....34

2.4 หมายเลขบิตที่ควบคุมช่องทางออกแต่ละช่องบนอาร์ซีเอ็กซ์....37

2.5 ผลกระทบของสถานะบนแอกทูเอเตอร์ที่เชื่อมต่อช่องทางออก....37

3.1 Use Case Description – Separate Type of Field....42

3.2 Use Case Description – Detect EndPoint....43

3.3 Use Case Description – Control Movement....44

3.4 Use Case Description – Detect BorderLine....45

3.5 Use Case Description – Detect Impediment....46

3.6 Use Case Description – Dodge Obstacle....47

3.7 Use Case Description – Check Past Obstacle....48

3.8 Use Case Description – Calculate New Path....49

3.9 การแปลงความต้องการของระบบเป็นความต้องการในการออกแบบหุ่นยนต์....67

3.10 การแปลงความต้องการของระบบเป็นความต้องการในการออกแบบสนามทดลอง....72

3.11 การตอบสนองต่อการตรวจจับเส้นแบ่งเขตสนาม....74

3.12 รูปแบบการตรวจจับ และตอบสนองต่อทางขึ้นทางลง....76

4.1 การทดลองบนทางเรียบปกติ....77

4.2 การทดลองบนทางที่มีสิ่งกีดขวาง....80

4.3 การทดลองบนทางที่มีทางชัน....83

4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ....85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ลักษณะองค์ประกอบของไมโครคอนโทรลเลอร์ 6

2.2 ขั้นตอนในการประมวลผลของหน่วยประมวลผลไมโครคอนโทรลเลอร์ 6

2.3 การอินเทอร์รัพต์แบบระบุแอดเดรส 10

2.4 การอินเทอร์รัพต์แบบตารางเวคเตอร์ 10

2.5 การเปรียบเทียบของการไหลของข้อมูลแบบไม่ใช้ ดีเอ็มเอ และ ใช้ดีเอ็มเอ 11

2.6 การทำงานของหน่วยความจำแคชโดยเปรียบเทียบกรณีพบกับไม่พบข้อมูลในแคช 12

2.7 ระบบการทำงานของหน่วยความจำเสมือน 13

2.8 แผนภาพส่วนประกอบของซอฟต์แวร์ 13

2.9 วงจรการพัฒนาระบบแบบทำซ้ำ 16

2.10 วงจรการพัฒนาระบบแบบน้ำตก 16

2.11 วงจรการพัฒนาระบบแบบน้ำตกที่เหมาะสม 17

2.12 วงจรการพัฒนาระบบแบบวิวัฒนาการ 17

2.13 วงจรการพัฒนาระบบแบบเพิ่มขึ้น 18

2.14 วงจรการพัฒนาระบบแบบก้นหอย 18

2.15 รูปแบบวงจรการพัฒนาระบบที่เลือกใช้ 19

2.16 ขั้นตอนการหาความต้องการที่แท้จริงของระบบ 20

2.17 สัญลักษณ์ยูสเคส 22

2.18 สัญลักษณ์ผู้กระทำ 22

2.19 สัญลักษณ์ขอบเขตระบบ 23

2.20 สัญลักษณ์ของคลาสและองค์ประกอบ 24

2.21 ตัวอย่างระดับความสัมพันธ์ระหว่างคลาสเท่ากับ 1 25

2.22 ตัวอย่างระดับความสัมพันธ์ระหว่าง 2 คลาส 25

2.23 ตัวอย่างระดับความสัมพันธ์ระดับ 3 คลาส 26

2.24 ตัวอย่างแผนภาพอ็อบเจกต์ 26

2.25 สัญลักษณ์ภายในแผนภาพลำดับเหตุการณ์ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.26 ตัวอย่างแผนภาพการปฏิสัมพันธ์ 28
2.27 สัญลักษณ์แผนภาพกิจกรรมแบบทางเลือกตัดสินใจ 29
2.28 สัญลักษณ์แผนภาพกิจกรรมแบบทำงานพร้อมกัน 29
2.29 สัญลักษณ์แผนภาพกิจกรรมแบบแบ่งส่วนด้วยลู่ว่ายน้ำ 29
2.30 ตัวอย่างแผนภาพสถานะ 30
2.31 ลักษณะปุ่มต่างๆ บนตัวอาร์ซีเอ็กซ์ 33
2.32 ลักษณะช่องทางเข้าของอาร์ซีเอ็กซ์ 1, 2 และ 3 33
2.33 ชนิดของตัวรับรู้แบบต่างๆ 34
2.34 ลักษณะตัวรับ/ส่งอินฟราเรดของอาร์ซีเอ็กซ์ 35
2.35 ลักษณะหน้าจอแอลซีดีบนอาร์ซีเอ็กซ์ 35
2.36 ช่องทางออกของอาร์ซีเอ็กซ์ A, B และ C 36
3.1 แผนภาพยูสเคสระบบตรวจจับเส้นทาง 41
3.2 แผนภาพกิจกรรม Separate Type Field 50
3.3 แผนภาพกิจกรรม Detect EndPoint 51
3.4 แผนภาพกิจกรรม Control Movement 52
3.5 แผนภาพกิจกรรม Detect BorderLine 53
3.6 แผนภาพกิจกรรม Detect Impediment 54
3.7 แผนภาพกิจกรรม Dodge Obstacle 56
3.8 แผนภาพกิจกรรม Check Past Obstacle 57
3.9 แผนภาพกิจกรรม Calculate New Path 59
3.10 แผนภาพ Class Diagram ของระบบหุ่นยนต์ตรวจจับเส้นทาง 60
3.11 Sequence Diagram ของ Usecase Separate Type Field ของระบบหุ่นยนต์ตรวจจับเส้นทาง. 60
3.12 Sequence Diagram ของ Usecase Detect EndPoint ของระบบหุ่นยนต์ตรวจจับเส้นทาง 61
3.13 Sequence Diagram ของ Usecase Control Movement ของระบบหุ่นยนต์ตรวจจับเส้นทาง 61
3.14 Sequence Diagram Usecase Detect Border Line ของระบบหุ่นยนต์ตรวจจับเส้นทาง 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.15 Sequence Diagram Usecase Detect Impediment ของระบบหุ่นยนต์ตรวจจับเส้นทาง.....62

3.16 Sequence Diagram Usecase Dodge Obstacle ของระบบหุ่นยนต์ตรวจจับเส้นทาง.....63

3.17 Collaboration Diagram Usecase Separate Type Field ของระบบหุ่นยนต์ตรวจจับเส้นทาง... 63

3.18 Collaboration Diagram Usecase Detect End Point ของระบบหุ่นยนต์ตรวจจับเส้นทาง.....64

3.19 Collaboration Diagram Usecase Control Movement ของระบบหุ่นยนต์ตรวจจับเส้นทาง..... 64

3.20 Collaboration Diagram Usecase Detect Border Line ของระบบหุ่นยนต์ตรวจจับเส้นทาง..... 64

3.21 Collaboration Diagram Usecase Detect Impediment ของระบบหุ่นยนต์ตรวจจับเส้นทา..... 65

3.22 Collaboration Diagram Usecase Dodge Obstacle ของระบบหุ่นยนต์ตรวจจับเส้นทาง..... 65

3.23 Statechart Diagram Obstacle Control ของระบบหุ่นยนต์ตรวจจับเส้นทาง.....65

3.24 Statechart Diagram Dodge Control ของระบบหุ่นยนต์ตรวจจับเส้นทาง..... 66

3.25 หุ่นยนต์ตรวจจับเส้นทาง..... 66

3.26 รูปแบบเซนเซอร์แสง.....67

3.27 รูปแบบกันชน.....67

3.28 รูปแบบชุดขับเคลื่อน.....67

3.29 การประกอบฐานรองรับล้อและมอเตอร์.....68

3.30 การประกอบส่วนโครงสร้างหลัก และ มอเตอร์..... 68

3.31 การประกอบส่วนแกนล้ออิสระ..... 69

3.32 การประกอบส่วนยึดเซ็นเซอร์แสง.....69

3.33 การประกอบส่วนเซ็นเซอร์แสงเข้ากับโครงสร้างหลัก..... 70

3.34 การประกอบก้านตรวจจับสิ่งกีดขวาง.....70

3.35 การประกอบก้านตรวจจับสิ่งกีดขวางกับโครงสร้างหลัก..... 71

3.36 การประกอบอาร์ซีเอ็กซ์เข้ากับโครงสร้างหลัก และ เดินสายข้อมูลเข้าสู่อาร์ซีเอ็กซ์.....71

3.37 รูปแบบพื้นสนาม.....72

3.38 รูปแบบสิ่งกีดขวาง.....72

3.39 รูปแบบจุดสิ้นสุด.....73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.40 รูปแบบเส้นแบ่งเขต.....73

3.41 รูปแบบทางชัน.....73

3.42 การแบ่งพื้นที่และการกำหนดสีเส้นแบ่งเขตสนาม.....74

3.43 การแบ่งพื้นที่และการกำหนดสีทางขึ้นลง.....75

4.1 รูปแบบสนามการทดลองบนทางเรียบธรรมดา.....78

4.2 หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 1.....77

4.3 หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 2.....78

4.4 หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 3.....78

4.5 หุ่นยนต์เมื่อถึงพบขอบเขตสนาม.....79

4.6 หุ่นยนต์เมื่อถึงจุดสิ้นสุด.....79

4.7 การทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านซ้าย).....80

4.8 ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านซ้าย).....80

4.9 การทดลองบนทางที่มีสิ่งกีดขวาง (การชนตรงกลาง).....81

4.10 ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนด้านหน้า).....81

4.11 ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนด้านหน้า).....81

4.12 การทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านขวา).....82

4.13 ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านขวา).....82

4.14 การทดลองบนทางที่มีทางชัน (การขึ้นทางชัน).....83

4.15 ผลการทดลองบนทางที่มีทางชัน (การขึ้นทางชัน).....83

4.16 การทดลองบนทางที่มีทางชัน (การลงทางชัน).....84

4.17 ผลการทดลองบนทางที่มีทางชัน (การลงทางชัน).....84

4.18 แบบสนามที่ใช้ทดลองการทำงานของหุ่นยนต์.....85

4.19 หุ่นยนต์เคลื่อนที่เข้าสู่พื้นที่สิ่งกีดขวาง.....85

4.20 หุ่นยนต์หลบหลีกสิ่งกีดขวางระหว่างทางเดิน.....86

4.21 หุ่นยนต์ชนขอบกั้นสะพาน.....86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.22 หุ่นยนต์หลบหลีกขอบสะพาน....86
4.23 หุ่นยนต์ขึ้นสะพานด้วยความเร่ง....87
4.24 หุ่นยนต์ขึ้นถึงพื้นที่ปกติบนทางชัน....87
4.25 หุ่นยนต์ลงทางชันด้วยความหน่วง....87
4.26 หุ่นยนต์เข้าถึงพื้นที่ที่มีจุดสิ้นสุด....88
4.27 หุ่นยนต์ทำการเปลี่ยนทิศการเคลื่อนที่เพื่อไม่ให้ออกนอกสนาม....88
4.28 หุ่นยนต์เคลื่อนที่เข้าจุดสิ้นสุด....88
4.29 หุ่นยนต์หยุดการทำงานเมื่อพบกับจุดสิ้นสุด....89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีระบบฝังตัวเป็นเทคโนโลยีที่เข้ามาทำให้อุปกรณ์มีการตอบสนองกับผู้ใช้ได้อย่างชาญฉลาด มีขนาดเล็กและใช้พลังงานต่ำ เปรียบเสมือนการใช้คอมพิวเตอร์ส่วนบุคคล มาย่อลงในผลิตภัณฑ์เพื่อให้สามารถประมวลผลและตอบสนองต่อคำสั่งของผู้ใช้งานได้ โดยระบบฝังตัวนี้ใช้ทรัพยากรต่างๆเท่าที่จำเป็น ทำให้ความสามารถของเทคโนโลยีด้านนี้มีการประยุกต์ไปอย่างแพร่หลาย และมีการพัฒนาอย่างต่อเนื่อง โดยส่วนมากแล้วระบบฝังตัวจะเป็นสิ่งรอบตัวเรา ที่มีทั้งเทคโนโลยีฮาร์ดแวร์ และเทคโนโลยีซอฟต์แวร์อยู่ภายใน เช่น รถยนต์ มือถือ ทีวี ตู้เย็น เป็นต้น

ในปัจจุบันยังไม่มีการพัฒนาระบบฝังตัวอย่างเป็นมาตรฐานที่แน่ชัด ที่จะส่งผลให้การพัฒนาระบบฝังตัวเป็นไปอย่างมีศักยภาพ และมีลำดับขั้นตอนการดำเนินการที่มีประสิทธิภาพ ที่จะตอบสนองต่อความต้องการในการพัฒนาระบบอย่างมีทิศทาง อีกทั้งยังไม่ค่อยมีการใช้แนวทางการพัฒนาระบบเข้ามาช่วยในการพัฒนา ที่จะเป็นพื้นฐานในการประยุกต์ใช้งานต่อไปในอนาคต

ระบบฝังตัว ส่วนใหญ่แล้วจะมีทรัพยากรในการทำงานที่ค่อนข้างจำกัด เหมาะสำหรับการทำงานเฉพาะอย่างไม่ซับซ้อนมากนัก และอีกอย่างคือหน่วยความจำที่มีขนาดจำกัด เนื่องจากความสามารถของหน่วยประมวลผลที่ต้องมีความเหมาะสมกัน ดังนั้นการที่จะให้แนวคิดการเขียนโปรแกรมลงไปในระบบฝังตัว จึงมีข้อจำกัดเกิดขึ้นหลายอย่าง จำเป็นจะต้องเขียนโปรแกรมที่ตรงตามจุดประสงค์ โดยที่ไม่เกินกว่าหน่วยความจำ การจะทำได้นั้น จำเป็นจะต้องอาศัยแนวคิดการพัฒนาระบบเชิงวัตถุเข้ามาช่วย

ซึ่งแนวคิดที่จะนำหลักการของยูเอ็มแอล (UML: Unified Modeling Language) ที่เป็นมาตรฐานการพัฒนาระบบที่ใช้ในการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design Methodology) มาเป็นเครื่องมือที่ใช้ในการพัฒนาระบบ เพื่อให้เข้าใจถึงโครงสร้าง, องค์ประกอบ และสิ่งที่เกี่ยวข้องกับระบบ รวมไปถึงการทำงาน, ความต้องการ และความรู้เกี่ยวกับระบบทุกขั้นตอนอีกด้วย จะช่วยให้การเขียนโปรแกรมให้มีลำดับขั้นตอน และยังเน้นไปที่การนำกลับมาใช้ใหม่ซึ่งจะช่วยการเขียนโปรแกรมให้มีขนาดเล็กลง รวมไปถึงการแบ่งระบบออกเป็นส่วน ที่จะทำให้การตัดสินใจหาแนวทางในการพัฒนาระบบง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของการศึกษา

โครงงานฉบับนี้มุ่งหวังเพื่อกระตุ้น ส่งเสริมและให้ความรู้ในการพัฒนาระบบฝังตัวอย่างเป็นกระบวนการ สามารถนำไปประยุกต์ใช้ในการออกแบบผลิตภัณฑ์ต้นแบบใหม่ๆ รวมไปถึงส่งเสริมให้เกิดการพัฒนาระบบฝังตัวขั้นพื้นฐานดังนี้

1. เพื่อศึกษาแนวคิดโครงสร้างและองค์ประกอบของระบบฝังตัว และการประยุกต์ใช้งาน
2. เพื่อศึกษาแนวทางในการพัฒนาระบบฝังตัว
3. เพื่อศึกษาฮาร์ดแวร์ระบบฝังตัว (ชุดทดลองเลโก้รุ่นอาร์ซีเอ็กซ์) และการประยุกต์ใช้งาน
4. เพื่อศึกษาเครื่องมือในการพัฒนาซอฟต์แวร์ระบบฝังตัวสำหรับชุดการทดลองเลโก้รุ่นอาร์ซีเอ็กซ์
5. เพื่อศึกษาและพัฒนาแอปพลิเคชั่นสำหรับระบบฝังตัว หุ่นยนต์ตรวจจับเส้นทาง

1.3 ทฤษฎีที่เกี่ยวข้อง

ระบบฝังตัวเป็นระบบที่มีการประมวลผลภายในตัวของมันเอง และนำมาใช้ในงานเฉพาะด้าน ส่งผลทำให้อุปกรณ์ที่ระบบฝังตัวเข้าไปทำงาน มีการทำงานตามกระบวนการที่ได้โปรแกรมไว้ โดยระบบฝังตัวมีหลักการทำงานเปรียบเสมือนคอมพิวเตอร์ขนาดเล็ก ซึ่งจะมีเรียลไทม์เคอร์เนล (Real time-kernel) หรือโมดูลกลางที่ทำหน้าที่ในการให้กำเนิดฟังก์ชั่นพื้นฐานของระบบปฏิบัติการแบบเรียลไทม์ และมีดีไวซ์ไดรเวอร์หรือโมดูลที่ทำหน้าที่ในการดูแลควบคุมส่วนของอุปกรณ์อินพุตและเอาท์พุต อันสุดท้ายคือ มิดเดิลแวร์ หรือโมดูลที่ทำหน้าที่สนับสนุนฟังก์ชั่นทั่วไปของระบบปฏิบัติการ โดยโครงงานนี้ได้นำอุปกรณ์ของเลโก้ที่เป็นระบบฝังตัวมาพัฒนา ประกอบกับแนวทางในการพัฒนาระบบเชิงวัตถุที่ใช้ยูเอ็มแอล (UML) มาเป็นเครื่องมือในพัฒนา ซึ่งยูเอ็มแอลก็คือภาษารูปภาพ เข้ามาช่วยในการบรรยายภาพรวมของระบบ เพื่อให้เกิดความเข้าใจ และก่อให้เกิดแบบจำลองที่สามารถนำมาช่วยในการพัฒนาระบบฝังตัวได้ง่ายขึ้น

วิธีในการพัฒนาด้านฮาร์ดแวร์โดยปกติแล้วจะนำ ดีเอฟดี (Data Flow Diagram) เข้ามาประกอบกับ ผังงาน (Flowchart) โดยดีเอ็ฟดีจะแสดงให้เห็นถึงข้อมูลและการเคลื่อนไหวของข้อมูลในระบบ ที่จะเห็นแหล่งที่มา ที่ไปของข้อมูล แต่จะไม่เห็นในส่วนของกลุ่ม หรือชุดคำสั่งต่างๆที่อยู่ภายใน ส่วนทางด้านผังงานนั้นจะแสดงถึงขั้นตอนการทำงานทั้งหมดของระบบเพียงอย่างเดียว โดยยูเอ็มแอลนั้นจะแสดงให้เห็นถึงหน้าที่ของระบบ ขอบเขตระบบ ส่วนของขั้นตอนการทำงาน การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดต่อต่างๆ รวมไปถึงวางแผนโครงสร้างของการโปรแกรมให้มีการติดต่อ ลำดับขั้นตอนการทำงานของส่วนต่างๆอีกด้วย

1.4 ขอบเขตของโครงงาน

โครงงานฉบับนี้ได้นำเสนอการพัฒนาระบบฝังตัวโดยใช้ยูเอ็มแอลดังนี้

1. ศึกษาวิเคราะห์ และออกแบบแอปพลิเคชั่นระบบฝังตัว หุ่นยนต์ตรวจจับเส้นทาง โดยใช้กระบวนการ การพัฒนาระบบฝังตัวแบบอ็อบเจ็คโอเรี่ยนเต็ด (OO: Object Oriented)
2. ศึกษาวิเคราะห์ และออกแบบฮาร์ดแวร์ระบบฝังตัวโดยใช้ชุดทดลองเลโก้รุ่นอาร์ซีเอ็กซ์สำหรับแอปพลิเคชั่นหุ่นยนต์ตรวจจับเส้นทาง
3. ศึกษาวิเคราะห์ และออกแบบโปรแกรมสำหรับแอปพลิเคชั่นหุ่นยนต์ตรวจจับเส้นทางบนชุดทดลอง เลโก้รุ่นอาร์ซีเอ็กซ์
4. ทำการทดลอง และ วิเคราะห์ผลลัพธ์ที่ได้

1.5 ขั้นตอนของการศึกษา

โครงงานฉบับนี้ได้วางแผนขั้นตอนออกเป็น 7 ขั้นตอนดังนี้

ขั้นตอนที่ 1 ศึกษาทฤษฎีที่เกี่ยวข้อง ประกอบด้วยสถาปัตยกรรมของระบบฝังตัว แนวทางและเครื่องมือในการพัฒนาระบบฝัง ฮาร์ดแวร์ชุดทดลองเลโก้รุ่นอาร์ซีเอ็กซ์เครื่องมือในการพัฒนาโปรแกรมสำหรับชุดทดลองเลโก้รุ่นอาร์ซีเอ็กซ์

ขั้นตอนที่ 2 ศึกษาวิเคราะห์ความต้องการของระบบ

ขั้นตอนที่ 3 ออกแบบแอปพลิเคชั่นระบบฝังตัว “หุ่นยนต์ตรวจจับเส้นทาง” โดยใช้ยูเอ็มแอล (UML) เป็นเครื่องมือในการพัฒนาแอปพลิเคชั่น (Application)

ขั้นตอนที่ 4 ออกแบบฮาร์ดแวร์หุ่นยนต์ตรวจจับเส้นทาง และสนามทดลอง

ขั้นตอนที่ 5 พัฒนาโปรแกรมสำหรับควบคุมการทำงานของหุ่นยนต์ตรวจจับเส้นทาง

ขั้นตอนที่ 6 ทำการทดลองหุ่นยนต์ตรวจจับเส้นทางภายใต้ขอบเขตความต้องการที่กำหนดและวิเคราะห์ผลการทดลอง

ขั้นตอนที่ 7 สรุปผลการทดลอง ปัญหาอุปสรรค และข้อเสนอแนะในการพัฒนาหุ่นยนต์ตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่ได้รับ

1. สามารถเรียนรู้และเข้าใจเทคโนโลยีระบบฝังตัว และการประยุกต์ใช้งาน
2. สามารถเรียนรู้และเข้าใจแนวทางการออกแบบและพัฒนาแอปพลิเคชั่นระบบฝังตัวอย่างมีระบบ
3. สามารถเรียนรู้โครงสร้างฮาร์ดแวร์ และซอฟต์แวร์ระบบฝังตัวบนชุดทดลองเลโก้มายสตอร์มรุ่นอาร์ซีเอ็กซ์ (Lego Mindstorms RCX)
4. สามารถออกแบบแอปพลิเคชั่นระบบฝังตัวโดยการประยุกต์แนวทางในการพัฒนาระบบฝังตัวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีบท

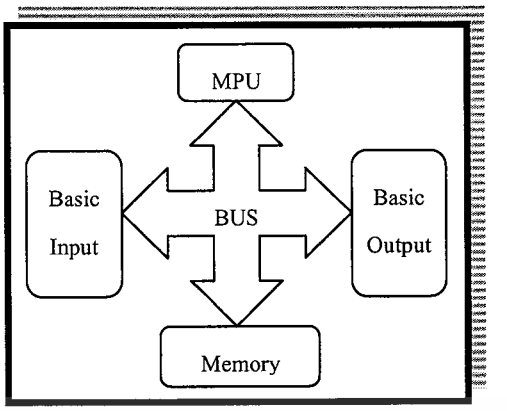
2.1 ระบบฝังตัว (Embedded System)

ระบบฝังตัว คือ ระบบที่มีไมโครชิป (Microchip) กับโปรแกรมเพื่อควบคุมไมโครชิปที่ฝังตัวอยู่ หรือมีระบบไอที (IT) ฝังอยู่ ซึ่งเปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กที่ทำงานได้เฉพาะด้านเฉพาะฟังก์ชั่นเท่านั้น และมีการตอบสนองเป็นเรียลไทม์ (Real Time) คือ สามารถตอบสนองในช่วงเวลาที่จำกัดได้ แต่ก็ขึ้นอยู่กับงานหรือฟังก์ชั่นที่ทำว่าให้ความสำคัญกับการตอบสนองมากน้อยเท่าใด ซึ่งการตอบสนองแบบเรียลไทม์มี 2 แบบ คือ ฮาร์ดเรียลไทม์(Hard - Real Time) กับซอฟท์เรียลไทม์ (Soft – Real Time) โดย ฮาร์ดเรียลไทม์(Hard - Real Time) คือ การตอบสนองที่เคร่งครัดต่อเวลา ไม่มีความยุ่นหยุ่น ส่วนซอฟท์เรียลไทม์ (Soft – Real Time) คือ การตอบสนองที่ไม่ได้เคร่งครัดเวลาในการตอบสนองมากเกินไปอาจมียืดหยุ่นได้แล้วแต่ความเหมาะสม แต่ในอาร์ซีเอ็กซ์ (RCX) นั้นเป็นการตอบสนองแบบซอฟท์เรียลไทม์ (Soft - Real Time) ระบบฝังตัวมีองค์ประกอบหลักๆ 2 องค์ประกอบคือ องค์ประกอบทางด้านฮาร์ดแวร์ (Hardware) และองค์ประกอบทางด้านซอฟต์แวร์ (Software)

2.1.1 องค์ประกอบทางด้านฮาร์ดแวร์

ส่วนสำคัญที่ถือได้ว่าเป็นหลักของระบบฝังตัวก็คือ ไมโครคอนโทรลเลอร์ (Microcontroller) ซึ่งเปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กที่ใช้ในการประมวล และการจัดการทางด้านต่างๆ ไมโครคอนโทรลเลอร์ที่ใช้ในการประมวลผล มีส่วนประกอบที่สำคัญได้แก่ เอ็มพียู (MPU) หน่วยความจำ(Memory) บัส (Bus) และอินพุต / เอาท์พุต (I/O) ดังรูปที่ 2.1 ส่วนที่เป็นเทคโนโลยีที่จำเป็นต่อระบบฝังตัวอีกส่วนหนึ่งนอกเหนือจากไมโครคอนโทรลเลอร์ก็คือ เทคโนโลยีการขัดจังหวะ (Interrupt) เทคโนโลยีดีเอ็มเอ (DMA) เทคโนโลยีแคช (Catch) และเทคโนโลยีหน่วยความจำเสมือน (Visual Memory)

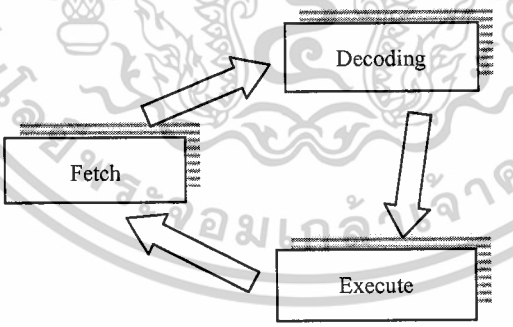
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ลักษณะองค์ประกอบของไมโครคอนโทรลเลอร์ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 89)

2.1.1.1 เอ็มพียู (MPU)

เอ็มพียู คือ หน่วยประมวลผลไมโคร (Microprocessing Unit) เปรียบเสมือนหน่วยประมวลผลกลาง (CPU) ในเครื่องคอมพิวเตอร์ มีหน้าที่ในการประมวลผลคำสั่งต่างๆ และตอบสนองต่อคำสั่งนั้นๆ โดยมีขั้นตอนในการประมวลผลคำสั่งอยู่ 3 ขั้นตอน คือ ขั้นตอนการอ่านคำสั่ง (Fetch) เป็นการอ่านคำสั่งที่เราโปรแกรมเข้ามาโดยคำสั่งนั้นจะเป็นภาษาที่เอ็มพียูเข้าใจ ขั้นตอนการตีความคำสั่ง (Decoding) เป็นการตีความคำสั่งที่ได้รับเข้ามาว่าให้ทำอะไร ขั้นตอนการทำตามคำสั่ง (Execute) เป็นการทำตามคำสั่งที่ได้รับมา จากนั้นก็รับอีกคำสั่งเข้ามาวนเวียนไปเรื่อยๆ ดังรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนในการประมวลผลของหน่วยประมวลผลไมโครคอนโทรลเลอร์ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 92)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.2 หน่วยความจำ (Memory)

หน่วยความจำ คือ ที่สำหรับเก็บรักษาข้อมูลข่าวสาร เช่น โปรแกรมและข้อมูลที่อยู่ ภายในไมโครคอมพิวเตอร์ สำหรับระบบสมองกลฝังตัวซึ่งประกอบด้วยส่วนที่เป็นฮาร์ดแวร์และซอฟท์แวร์ นั้นที่เก็บซอฟท์แวร์ก็คือ หน่วยความจำนั่นเอง หน่วยความจำที่นิยมใช้ในระบบฝังตัวมีอยู่ 2 ประเภทคือ หน่วยความจำรอม (ROM) และ หน่วยความจำแรม (RAM) หน้าที่หลักของหน่วยความจำของทั้ง 2 ประเภทคือ ใช้ในการเก็บข้อมูลและโปรแกรมต่างๆ

1. หน่วยความจำรอม (ROM)

หน่วยความจำรอม (ROM: Read Only Memory) เป็นหน่วยความจำที่ใช้สำหรับอ่านข้อมูลเมื่อมีการระบุแอดเดรส (Address) ใดๆมา โดยมีคุณลักษณะพิเศษคือ ข้อมูลนั้นจะคงอยู่ไม่หายไป แม้จะไม่มีการจ่ายไฟให้ก็ตาม (Non - Volatile) จึงมักที่จะนำรอมไปใช้ในการเก็บโปรแกรมเริ่มต้นที่ต้องถูกเอ็มพียู เข้ามาตีความทุกครั้งที่เริ่มต้นการทำงานหลังจากมีการจ่ายไฟให้กับระบบ เช่น ระบบปฏิบัติการ(Operating system) ไดรเวอร์ (driver) ของอุปกรณ์ต่างๆ ค่าตัวแปรที่ไม่มีการเปลี่ยนค่าระหว่างการทำงาน ฯลฯ โดยส่วนใหญ่แล้วรอม จะเป็นหน่วยความจำที่อ่านได้อย่างเดียว ไม่สามารถเขียนข้อมูลลงไปในรอมได้ นอกจากจะเป็นรอมประเภทพิเศษที่สามารถเขียนข้อมูลลงไปใหม่หรือเขียนทับได้ ทีหลัง

2. หน่วยความจำแรม (RAM)

หน่วยความจำแรม (RAM: Random Access Memory) เป็นหน่วยความจำที่เมื่อระบุแอดเดรสแล้วสามารถเข้าไปอ่านหรือเขียนค่าได้ แต่การใช้หน่วยความจำแรมนั้นต้องมีการจ่ายไฟเลี้ยงตลอดเวลาเพื่อไม่ให้ข้อมูลนั้นหายไป (Volatile) แต่ก็มีบางส่วนที่ใช้ในการเก็บโปรแกรมที่พัฒนาขึ้นมาใหม่เพื่อควบคุมระบบฝังตัว เช่น อาร์ซีเอ็กซ์ เป็นต้น

2.1.1.3 บัส (Bus)

บัส คือ กลุ่มของสายสัญญาณที่ใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์ภายในและภายนอกไมโครคอนโทรลเลอร์ ซึ่ง สามารถแบ่งได้ออกเป็น 2 ประเภท คือ บัสภายในและ บัสภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บัสภายใน (Internal Bus):

บัสภายใน เป็นสายสัญญาณเชื่อมต่ออุปกรณ์ในเอ็มพียู(MPU) เช่น รีจิสเตอร์(register) หน่วยความจำ และ พอร์ต(Ports) เป็นต้น สามารถแบ่งได้ออกเป็น 3 ประเภทหลัก คือ แอดเดรสบัส (address bus) บัสข้อมูล (data bus) และ บัสควบคุม (control bus) ในส่วนของแอดเดรสบัสจะใช้ในการระบุอุปกรณ์ และข้อมูลจะถูกแลกเปลี่ยนผ่านบัสข้อมูล เพื่อทำให้เกิดการระบุและแลกเปลี่ยนข้อมูล การกระทำทั้งหมดนี้จะถูกกำหนดผ่านบัสควบคุม

2. บัสภายนอก (External Bus):

บัสภายนอกเป็นกลุ่มของสายสัญญาณที่ใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก ต่างๆ เช่น เซ็นเซอร์(Sensor) ที่เก็บข้อมูล (Storage) เป็นต้น

2.1.1.4 อินพุตเอาท์พุตพื้นฐาน (Basic I/O)

อินพุตเอาท์พุตพื้นฐาน เป็นส่วนที่ใช้ในการแปลงข้อมูลจากสภาพแวดล้อมภายนอกให้อยู่ในรูปแบบที่ เอ็มพียู(MPU) สามารถนำไปประมวลได้รวมทั้งแปลงข้อมูลที่ประมวลได้จาก เอ็มพียู (MPU) ให้อยู่ในรูปแบบที่รับรู้เข้าใจได้โดยมนุษย์

1. ส่วนอินพุตพื้นฐาน (Basic Input)

ส่วนอินพุตพื้นฐานโดยทั่วไปเรียกว่า อินพุตพอร์ต (Input Port) เป็นส่วนที่ใช้ในการรับข้อมูลและสัญญาณต่างๆจากอุปกรณ์ภายนอกมายังไมโครคอนโทรลเลอร์ โดยข้อมูลหรือสัญญาณเหล่านี้จะอยู่ในรูปของสัญญาณไฟฟ้า ซึ่งถูกจัดการโดย เอดีคอนเวอเตอร์ (A/D Converter) ในการแปลงจากสัญญาณแอนะล็อก (Analog) ไปเป็นสัญญาณดิจิทัล (Digital) เพื่อที่จะให้เอ็มพียูนำไปประมวลผลได้ รวมไปถึงการจัดการในเรื่องของการสลับการรับสัญญาณหรือข้อมูลจากอินพุตพอร์ตแต่ละช่องสัญญาณ ในกรณีที่มีสัญญาณหรือข้อมูลต่างๆ ผ่านอินพุตพอร์ตเป็นจำนวนมาก

2. ส่วนเอาท์พุตพื้นฐาน (Basic Output)

ส่วนเอาท์พุตพื้นฐานโดยทั่วไปเรียกว่า เอาท์พุตพอร์ต (Output Port) เป็นช่องทางออกของสัญญาณหรือข้อมูลที่ได้ผ่านการประมวลผลมาจากเอ็มพียูแล้ว โดยข้อมูลหรือสัญญาณเหล่านี้จะอยู่ในรูปทางกายภาพที่เหมาะสมกับอุปกรณ์ที่นำมาเชื่อมต่อกับเอาท์พุตพอร์ต โดยใช้เอดีคอนเวอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกเช่นกัน รวมไปถึงก็มีการจัดการเรื่องของการสลับการส่งสัญญาณหรือข้อมูลไปที่เอาท์พุตพอร์ต แต่ละช่องสัญญาณ ในกรณีที่มีการติดต่อกับอุปกรณ์เอาท์พุตเป็นจำมาก เช่นเดียวกับอินพุตพอร์ต

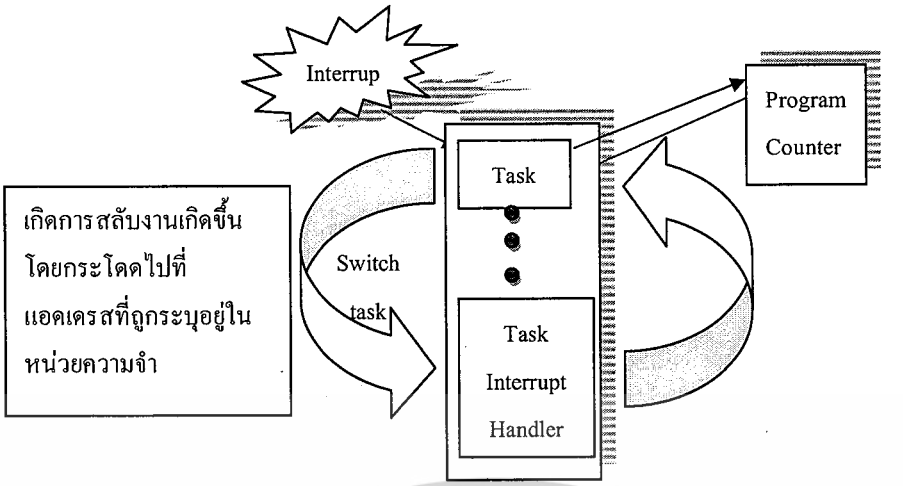
2.1.1.5 เทคโนโลยีการอินเทอร์รัพต์ (Interrupt Technology)

ดังที่กล่าวไปตั้งแต่ต้นว่าระบบฝังตัวนั้นมีการตอบสนองแบบเรียลไทม์ ซึ่งจะต้องตอบสนองหรือประมวลผลตามที่ได้วางไว้ภายในระยะเวลาที่จำกัด และต้องมีการติดต่อกับอุปกรณ์เป็นจำนวนมากทำให้ต้องใช้เทคโนโลยีการอินเทอร์รัพต์เข้ามาจัดการในส่วนนี้เพื่อทำให้ระบบมีความเป็นเรียลไทม์มากขึ้น

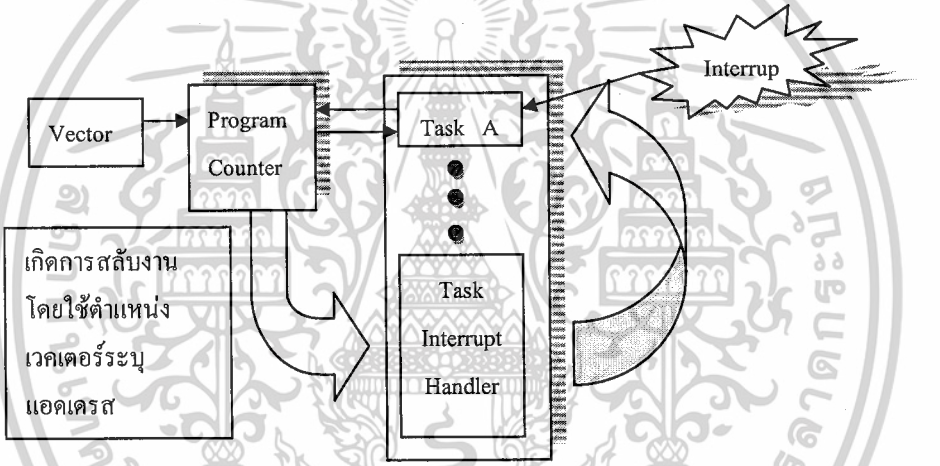
การอินเทอร์รัพต์ คือ การเปลี่ยนการทำงานจากอย่างหนึ่งเปลี่ยนเป็นอีกอย่างหนึ่งเมื่อมีเหตุการณ์มากระทบให้เกิดการอินเทอร์รัพต์เกิดขึ้น ในส่วนของการทำงานอินเทอร์รัพต์นั้นต้องมีการเตรียมการทั้งซอฟต์แวร์และฮาร์ดแวร์ ทางซอฟต์แวร์ก็คือโปรแกรม เนื่องจากอินเทอร์รัพต์ก็เป็นโปรแกรมชนิดหนึ่ง การเกิดอินเทอร์รัพต์นั้นเกิดได้หลายแบบ แต่ละแบบก็ต้องมีโปรแกรมสำหรับประมวลผลอินเทอร์รัพต์ที่แตกต่างกันไป จึงต้องมีฮาร์ดแวร์ที่ช่วยในการจัดการอินเทอร์รัพต์ ซึ่งก็คือ ตัวควบคุมการอินเทอร์รัพต์ (Interrupt Controller)

เมื่ออินเทอร์รัพต์เกิดขึ้น จะเกิดการสลับการทำงานไปยังงานที่ร้องขออินเทอร์รัพต์ ซึ่งในการสลับงานนั้นมี 2 วิธี วิธีแรกคือ การกระโดดไปยังแอดเดรสในหน่วยความจำที่ถูกระบุไว้สำหรับการประมวลผลอินเทอร์รัพต์โดยตรง วิธีที่สอง คือ การกระโดดโดยผ่าน ตารางเวคเตอร์ (Vector Table) ซึ่งตารางเวคเตอร์คือ การกำหนดพื้นที่ในหน่วยความจำไว้สำหรับรองรับการประมวลผลอินเทอร์รัพต์ เมื่อเกิดอินเทอรัพต์เกิดขึ้น ตารางเวคเตอร์นั้นจะถูกโหลดเข้าไปยังโปรแกรมเคาเตอร์ตามอินเทอร์รัพต์ที่เกิดขึ้น ทั้งนี้ทั้งนั้นทั้ง 2 วิธีจะมีการเก็บจุดที่กระทำอยู่ขณะเกิดอินเทอร์รัพต์เข้าไปในโปรแกรมเคาท์เตอร์ ก่อนการสลับงานเพื่อที่จะได้กลับมายังจุดที่ประมวลผลค้างเอาไว้หลังจากจัดการอินเทอร์รัพต์เสร็จแล้ว ดังรูปที่2.3 และ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การอินเทอร์รัพต์แบบระบุแอดเดรส (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 104)



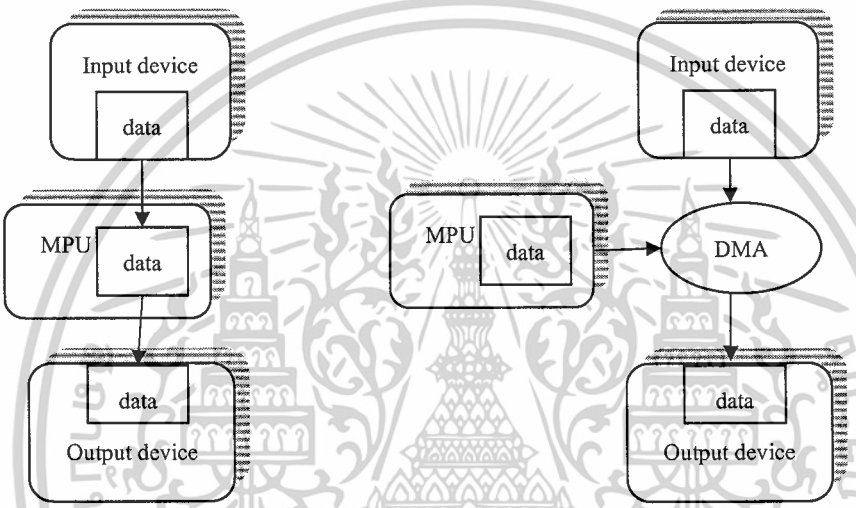
รูปที่ 2.4 การอินเทอร์รัพต์แบบตารางเวคเตอร์ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 104)

2.1.1.6 เทคโนโลยีดีเอ็มเอ (DMA Technology)

ดีเอ็มเอ (DMA: Direct Access Memory) เป็นการย้ายข้อมูลระหว่างหน่วยความจำกับอุปกรณ์หรือจะเป็นหน่วยความจำด้วยกันเอง โดยที่ไม่จำเป็นต้องผ่านการประมวลผลจากเอ็มพียู เพียงแต่ต้องมีการแจ้งให้เอ็มพียูทราบแค่นั้น (ดังรูป 2.5) ซึ่งหลังจากดีเอ็มเอได้รับทราบตำแหน่งและข้อมูลที่จะทำการย้ายจากเอ็มพียูแล้ว ดีเอ็มเอก็จะทำการย้ายข้อมูลไปในตำแหน่งที่ได้รับมาโดยที่เอ็มพียูไม่ได้ยุ่งเกี่ยวเลยหรือเรียกได้ว่าเป็นอิสระจากเอ็มพียู สำหรับการเคลื่อนย้ายแบบดีเอ็มเอนั้นจะต้องได้รับสิทธิในการครอบครองบัส รวมถึงต้องหยุดการทำงานของเอ็มพียูชั่วคราวด้วยซึ่งดีเอ็มเอนั้นจะถูกควบคุมการทำงานโดยตัวควบคุมดีเอ็มเอ (DMA Controller) จุดประสงค์ของดีเอ็มเด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อเพิ่มประสิทธิภาพให้กับระบบฝังตัวและลดภาระงานของเอ็มพียู โดยมีขึ้นตอนหลักๆ ดังนี้ ขั้นตอนแรกแจ้งให้เอ็มพียูทราบ ขั้นตอนที่สองทำการร้องขอการใช้บัสจากบัสควบคุม ขั้นตอนที่สามครอบครองบัส ขั้นตอนที่สี่ส่งข้อมูล และสุดท้ายปล่อยบัส ทั้งนี้ทั้งนั้นเนื่องจากการที่ดีเอ็มเอต้องมีการครอบครองบัส ซ้ำยังสามารถหยุดการทำงานของเอ็มพียูได้ จึงเป็นผลเสียถ้ามีการใช้ดีเอ็มเอมากเกินไป เช่น ถ้ามีการส่งข้อมูลขนาดใหญ่ จะทำให้การทำงานของเอ็มพียูช้าลง และเนื่องจากเอ็มพียูได้หยุดการทำงาน ทำให้ไม่สามารถตอบสนองอินเทอร์รัพต์ต่างๆที่ต้องการตอบสนองได้อย่างรวดเร็วได้ จึงไม่เหมาะสำหรับการนำใช้ในระบบฝังตัวที่ต้องการการตอบสนองแบบรวดเร็วได้



(ก) แบบไม่ใช้ดีเอ็มเอ (ข) แบบใช้ดีเอ็มเอ

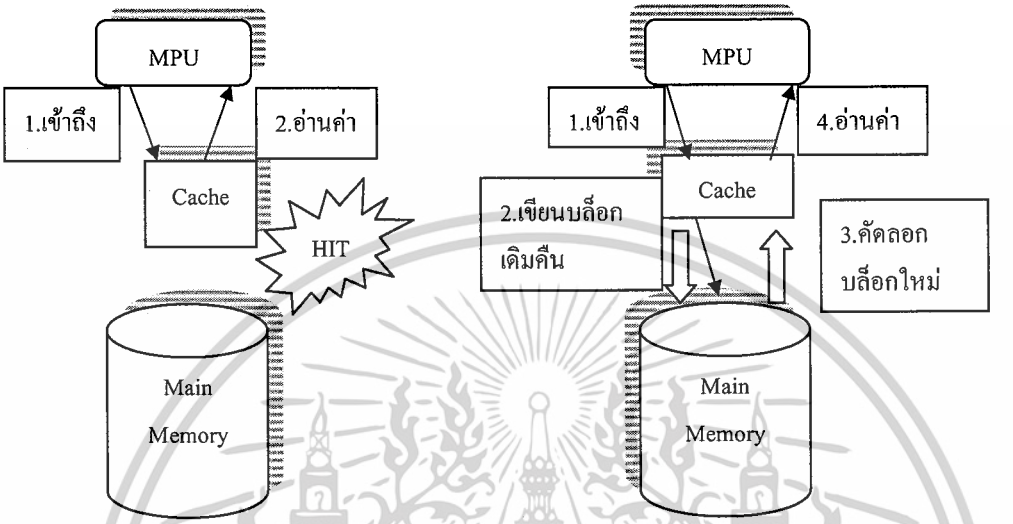
รูปที่ 2.5 การเปรียบเทียบของการไหลของข้อมูลแบบไม่ใช้ ดีเอ็มเอและใช้ดีเอ็มเอ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 104)

2.1.1.7 เทคโนโลยีหน่วยความจำแคช (Cache Memory Technology)

หน่วยความจำแคชเป็นหน่วยความจำขนาดเล็ก มีไว้เพื่อเก็บข้อมูลที่เอ็มพียูนั้นใช้งานบ่อยๆ เพื่อจุดประสงค์คือ เพิ่มความรวดเร็วในการประมวลผล เนื่องจากการที่เอ็มพียูเข้าไปหาข้อมูลที่ต้องการในการประมวลผล ในหน่วยความจำซึ่งมีขนาดใหญ่ทำให้การค้นหานั้นเป็นไปด้วยความล่าช้า การที่นำแคชมาใช้จะทำให้เอ็มพียูนั้นเข้าถึงข้อมูลได้รวดเร็วมากขึ้น การทำงานของแคชคือ การคัดลอกสำเนาข้อมูลที่เอ็มพียูใช้ในการประมวลผล สำหรับการเข้าถึงในครั้งต่อๆไป โดยหลักการมีดังนี้ เมื่อเอ็มพียูต้องการข้อมูลเพื่อใช้ในการประมวลผล จะเข้ามาค้นหาที่แคชก่อน เมื่อเจอก็จะอ่านค่าจากแคช แต่ถ้าไม่เจอ ก็ไปจะค้นหาที่หน่วยความจำหลัก เมื่อทำการค้นหาเจอ แคชก็จะทำการสำเนาข้อมูลที่เอ็มพียูใช้มาเก็บอยู่ในแคช พอเอ็มพียูต้องการใช้ข้อมูลตัวเดิมอีกก็จะมาหาที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แคช และ สามารถนำข้อมูลไปใช้ได้เลย โดยที่ไม่ต้องเสียเวลาค้นหาอีก (ดังรูป 2.6) แต่เนื่องจากแคชมีขนาดเล็ก ทำให้มีการเก็บข้อมูลจำกัด จึงมีการนำอัลกอริทึม (algorithm) มาใช้ในการเลือกว่าข้อมูลตัวไหนสมควรที่จะอยู่ต่อหรือถูกลบไป เพื่อให้ข้อมูลที่มีความต้องการมากกว่าเข้ามาอยู่แทน รวมไปถึงการอัพเดตข้อมูลให้ตรงกันกับหน่วยความจำหลักด้วย



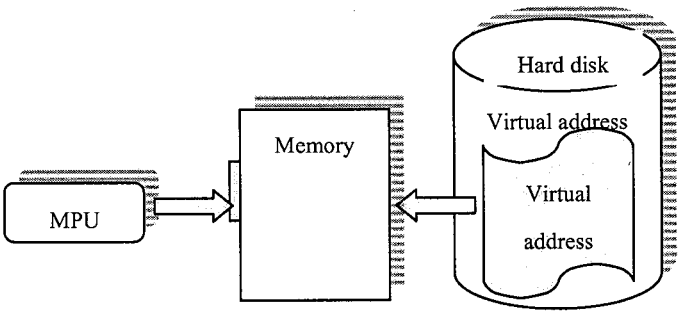
(ก) กรณีพบข้อมูลในแคช (ข) กรณีไม่พบข้อมูลในแคช

รูปที่ 2.6 การทำงานของหน่วยความจำแคชโดยเปรียบเทียบกรณีพบกับไม่พบข้อมูลในแคช(Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 108)

2.1.1.8 เทคโนโลยีหน่วยความจำเสมือน (Virtual Memory Technology)

การใช้หน่วยความจำเสมือน มีจุดประสงค์เพื่อ ทำให้โปรแกรมหรือข้อมูลที่มีขนาดใหญ่ถูกเรียกใช้งานโดยหน่วยความจำที่มีขนาดเล็กกว่าได้ ซึ่งก็คือโปรแกรมหรือข้อมูลจะถูกเก็บอยู่ในหน่วยความจำเสริม เช่น อุปกรณ์จำพวกฮาร์ดดิสก์ เป็นต้น โดยมีวิธีคือ การสร้างพื้นที่แอดเดรสเสมือนขึ้นมา โดยมีรายละเอียดการทำงานดังนี้ โปรแกรมจะอ่านคำสั่งการทำงานจากแอดเดรสในหน่วยความจำเสมือน แล้วแปลงแอดเดรสนี้ให้กลายเป็นแอดเดรสของหน่วยความจำจริง โดยผ่านฟังก์ชั่นการแปลงแอดเดรส แล้วทำการอ่านโปรแกรมจากหน่วยความจำจริง (ดังรูปที่ 2.7) แต่วิธีนี้ก็มีข้อเสียหากระบบต้องการตอบสนองในแบบเรียลไทม์ คือ ถ้าหน่วยความจำมีขนาดเล็กในขณะที่โปรแกรมที่ต้องการให้ทำงานมีขนาดใหญ่ ก็จะต้องทำการโหลดโปรแกรมจากฮาร์ดดิสก์อยู่บ่อยๆ ทำให้เสียเวลาไปมากในการเคลื่อนย้ายข้อมูลระหว่างฮาร์ดดิสก์ และ หน่วยความจำ จึงต้องใช้เทคนิคนี้อย่างระมัดระวัง

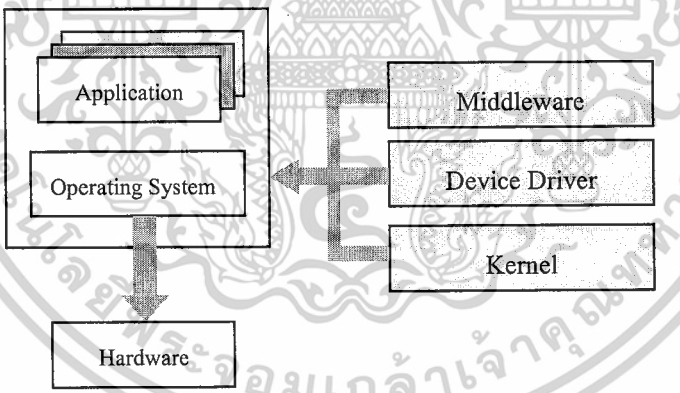
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ระบบการทำงานของหน่วยความจำเสมือน (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 111)

2.1.2 องค์ประกอบทางด้านซอฟต์แวร์

จุดประสงค์ของซอฟต์แวร์ระบบฝังตัวคือ การเพิ่มฟังก์ชั่นการทำงานของระบบฝังตัว ซึ่งรวมไปถึงการเพิ่มความสามารถของระบบฝังตัวด้วย โดยส่วนมากแล้วระบบฝังตัวส่วนใหญ่นั้นจะเป็นระบบเรียลไทม์ซึ่งเราจะเน้นตรงจุดนี้ว่ามีซอฟต์แวร์ตัวไหนบ้างที่ช่วยส่งเสริมทำให้ระบบเรียลไทม์นั้นเป็นไปได้มากที่สุด ซึ่งประกอบไปด้วย ระบบปฏิบัติการ (Operating System) ที่ประกอบไปด้วยเคอร์เนล ดีไวซ์ไดรเวอร์ และมิดเดิลแวร์ และโปรแกรมประยุกต์หรือแอพพลิเคชั่น (Application program) ดังรูปที่ 2.8



รูปที่ 2.8 แผนภาพส่วนประกอบของซอฟต์แวร์ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 18)

2.1.2.1 ระบบปฏิบัติการ

ระบบปฏิบัติการ เป็นซอฟท์แวร์ที่ใช้ร่วมกันในระบบต่างๆ และ มีหน้าที่ให้กรอบการทำงานแก่ซอฟท์แวร์อื่นๆ ในขณะที่มันทำงานอยู่ รวมทั้งใช้ในการจัดการทรัพยากรต่างๆของระบบฝังตัวซึ่งจะประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เคอร์เนล (Kernel)

เคอร์เนล คือ โมดูลกลางที่ทำหน้าที่ในการให้กำเนิดฟังก์ชั่นพื้นฐานของระบบปฏิบัติการ ซึ่งระบบฝังตัวนั้นได้ใช้เคอร์เนลที่เรียกว่า เรียลไทม์เคอร์เนล (Real time-kernel) เป็นส่วนของฟังก์ชั่นที่ทำให้เกิดการประมวลผลแบบเรียลไทม์ ซึ่งตัวมันมีหน้าที่ ควบคุมการจัดการเอ็มพียู เพื่อให้ง่ายต่อการสร้างระบบเรียลไทม์รวมไปถึงการให้กรอบการทำงานสำหรับระบบเรียลไทม์ ซึ่งมีฟังก์ชั่นต่างๆที่ใช้ดังนี้ ซึ่งประกอบไปด้วย

ฟังก์ชั่นการจัดการตารางเวลา จะใช้วิธีการจัดตารางเวลางานแบบให้ค่าความสำคัญของงานแต่ละงาน และใช้การเปลี่ยนแปลงสภาวะของงานอันเกิดจากเหตุการณ์ที่เข้ามากระทบซึ่งทำให้เกิดการเปลี่ยนแปลง เช่นการเกิดอินเทอร์รัพต์ การเริ่มงาน การหยุดงาน การรับอินพุต การส่งข้อมูลออกไปยังเอาท์พุต และ การหยุดโพรเซส(Process)

ฟังก์ชั่นการควบคุมอินเทอร์รัพต์ เป็นตัวที่ทำหน้าที่ควบคุมหรือเป็นตัวจัดการอินเทอร์รัพต์ ตามอินเทอร์รัพต์ที่เกิดขึ้น และดูแลจัดการอินเทอร์รัพต์ เช่น การบันทึกตัวจัดการอินเทอร์รัพต์ และ การตั้งค่าเวคเตอร์

ฟังก์ชั่นซิสเต็มคอล เป็นตัวที่ทำหน้าที่ให้บริการเพื่อให้งานต่างๆ ทำงานไปพร้อมกันได้ โดยจะมีฟังก์ชั่นต่างให้บริการ เช่น ฟังก์ชั่นเริ่มงาน ฟังก์ชั่นหยุดงาน ฟังก์ชั่นลบงาน ฟังก์ชั่นการติดต่อสื่อสารระหว่างงาน

2. ดีไวซ์ไดรเวอร์ (Device Driver)

ดีไวซ์ไดรเวอร์ เป็นโมดูลที่ทำให้เกิดการควบคุมอุปกรณ์อินพุตและเอาท์พุต มีหน้าที่คือซ่อนรายละเอียดของฮาร์ดแวร์ออกจากแอพพลิเคชั่น ทำให้แอพพลิเคชั่นเป็นอิสระจากการประมวลผลเพื่อควบคุมอุปกรณ์ และ ส่งผลกระทบน้อยที่สุดเมื่อมีการเปลี่ยนแปลงอุปกรณ์

3. มิดเดิลแวร์ (Middleware)

มิดเดิลแวร์ คือ เป็นโมดูลที่สนับสนุนฟังก์ชั่นทั่วไปของระบบปฏิบัติการซึ่งมีหน้าที่ให้บริการแก่แอพพลิเคชั่นผ่านทางการใช้ฟังก์ชั่นของเคอร์เนลและดีไวซ์ไดรเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 แอพพลิเคชั่น (Application)

โปรแกรมประยุกต์ หรือ แอพพลิเคชั่น นั่นเป็นโปรแกรมที่ใช้กรอบการทำงานที่ระบบปฏิบัติ การเสนอให้นั่นก็คือ เคอร์เนล, ดีไวซ์ไดรเวอร์ และ มิดเดิลแวร์ เพื่อประมวลผล ซึ่งโปรแกรมประยุกต์จะถูกสร้างให้เป็นระบบเรียลไทม์โดยการใช้ฟังก์ชั่นของเคอร์เนลที่กล่าวไว้ดังข้างต้น ด้วยการแบ่งงานออกเป็นงานย่อยหรือเรียกว่า แทสก์ (Task) แล้วทำการบันทึกตัวจัดการอินเทอร์รัพต์ของงานย่อยไว้ที่เคอร์เนล แล้วใช้ฟังก์ชั่นซิสเต็มคอลเพื่อทำให้งานย่อยเหล่านั้นสามารถทำงานร่วมกันได้

2.2 การพัฒนาระบบ

การพัฒนาระบบ (System Development Approach) หมายถึง กระบวนการทางความคิดที่จะทำให้การพัฒนาระบบมีการดำเนินการที่เป็นลำดับ ขั้นตอน เป็นมาตรฐานในการดำเนินงาน ทั้งด้านเทคนิค เอกสาร และเครื่องมือที่ใช้ ไปจนถึงดำเนินงานได้สำเร็จลุล่วงตามเป้าหมายที่ได้ตั้งไว้และอยู่ภายใต้ระยะเวลาที่กำหนด

สิ่งที่ได้จากการพัฒนาระบบ คือ ระบบงานที่ได้รับการปรับปรุงให้มีความถูกต้อง มีประสิทธิภามากขึ้น ทั้งทางด้านลำดับขั้นตอนการทำงาน และเครื่องมือต่างๆที่นำมาใช้ที่จะช่วยให้การทำงานแต่ละขั้นตอนมีประสิทธิภาพมากขึ้น ซึ่งก็คือ ตัวดำเนินการของระบบ หรือซอฟต์แวร์ของระบบงาน โดยการที่จะดำเนินโครงงานการพัฒนาระบบ “หุ่นยนต์ตรวจจับเส้นทาง” ให้มีทิศทางและมีประสิทธิภาพนั้นจำเป็นจะต้องอาศัย “แนวทางในการพัฒนาระบบ”

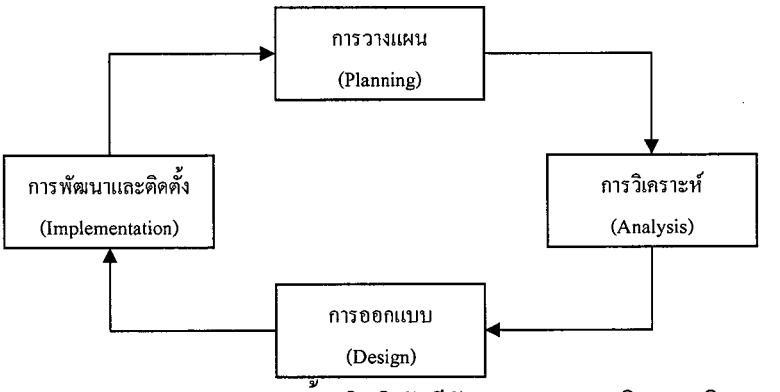
2.2.1 รูปแบบของวงจรการพัฒนาระบบ

วงจรการพัฒนาระบบ (SDLC: System Development Life Cycle) คือ กระบวนการทางความคิด (Logical Process) ในการพัฒนาเพื่อแก้ปัญหาการพัฒนาระบบ และตอบสนองความต้องการของผู้ใช้ได้ แบ่งออกเป็น 4 ระยะ (Phase) ได้แก่ ระยะวางแผน (Planning Phase) ระยะการวิเคราะห์ (Analysis Phase) ระยะการออกแบบ (Design Phase) ระยะการพัฒนาและติดตั้ง (Implementation Phase) โดยมีรูปแบบต่างๆดังนี้

2.2.1.1 วงจรการพัฒนาระบบแบบทำซ้ำ (SDLC แบบ Iterative)

วงจรการพัฒนาระบบในรูปแบบการทำซ้ำนั้น เป็นการพัฒนาที่ทำการพัฒนาระบบหลายๆรอบ โดยในแต่ละรอบจะได้สิ่งที่เราพัฒนาออกมา ดังรูปที่ 2.9

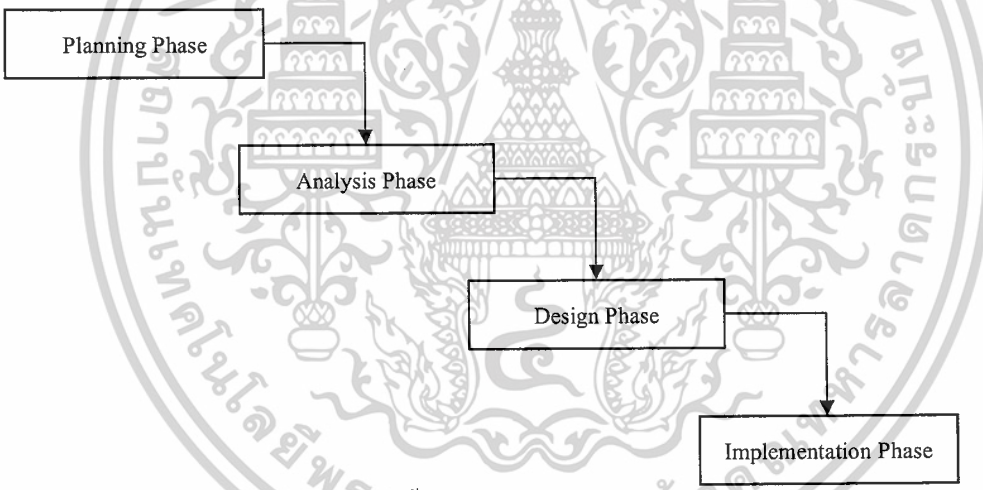
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 วงจรการพัฒนาระบบแบบทำซ้ำ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 7)

2.2.1.2 วงจรการพัฒนาระบบแบบน้ำตก (SDLC แบบ Waterfall)

เป็นการพัฒนาที่คล้ายกับน้ำตก คือ ไกลจากที่สูงลงสู่ที่ต่ำ และไม่สามารถไหลย้อนกลับได้อีก หากมีข้อผิดพลาดก่อนหน้าขั้นตอนที่ทำอยู่ จะไม่สามารถย้อนกลับมาแก้ไขได้ จึงจำเป็นจะต้องมีการวางแผนที่ดี ดังรูปที่ 2.10



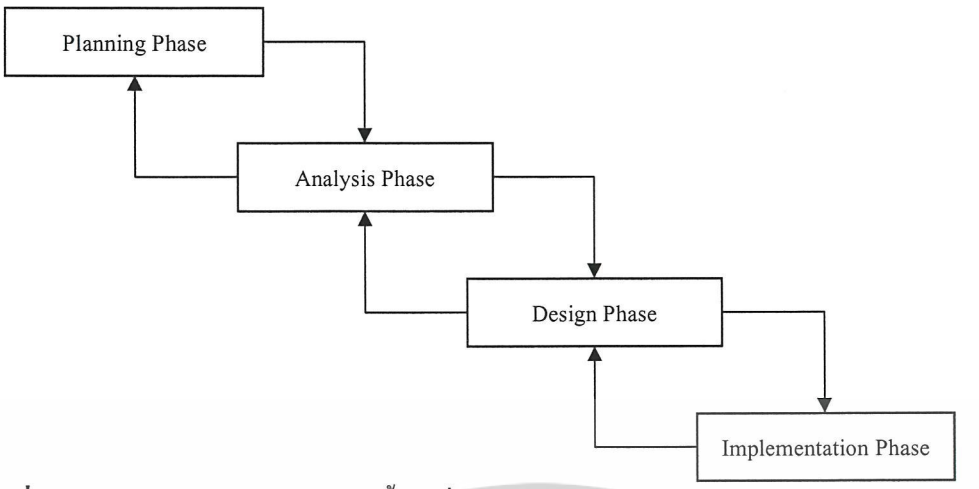
รูปที่ 2.10 วงจรการพัฒนาระบบแบบน้ำตก (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 8)

2.2.1.3 วงจรการพัฒนาระบบแบบน้ำตกที่ปรับเปลี่ยนได้ (SDLC แบบ Adapted Waterfall)

เป็นการพัฒนาที่ปรับปรุงมาจากแบบน้ำตก โดยในแต่ละขั้นตอนสามารถย้อนกลับมายังขั้นตอนก่อนหน้าเพื่อแก้ไขข้อผิดพลาด หรือสามารถย้อนกลับได้ ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

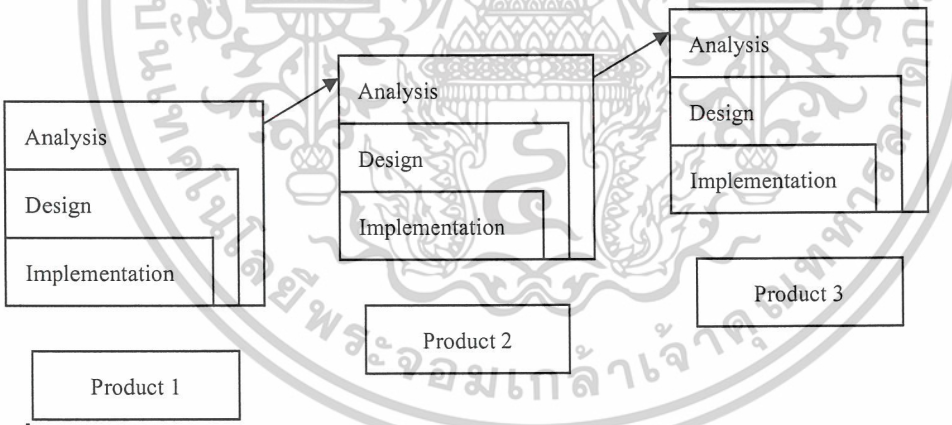
ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง



รูปที่ 2.11 วงจรการพัฒนาระบบแบบน้ำตกที่เหมาะสม (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 9)

2.2.1.4 วงจรการพัฒนาระบบแบบวิวัฒนาการ (SDLC แบบ Evolutionary)

เป็นรูปแบบที่มีแนวคิดมาจากการวิวัฒนาการ โดยจะพัฒนาระบบงานให้เสร็จในแต่ละรุ่น แล้วจึงพิจารณาว่ามีข้อดี ข้อเสียอย่างไรจึงเริ่มกระบวนการใหม่ ในรุ่นถัดมา จนกว่าจะได้ระบบที่สมบูรณ์ที่สุด สังเกตว่าการพัฒนาในแต่ละรุ่นจะไม่มีความสัมพันธ์กันเลย แม้แต่น้อย ดังรูปที่ 2.12



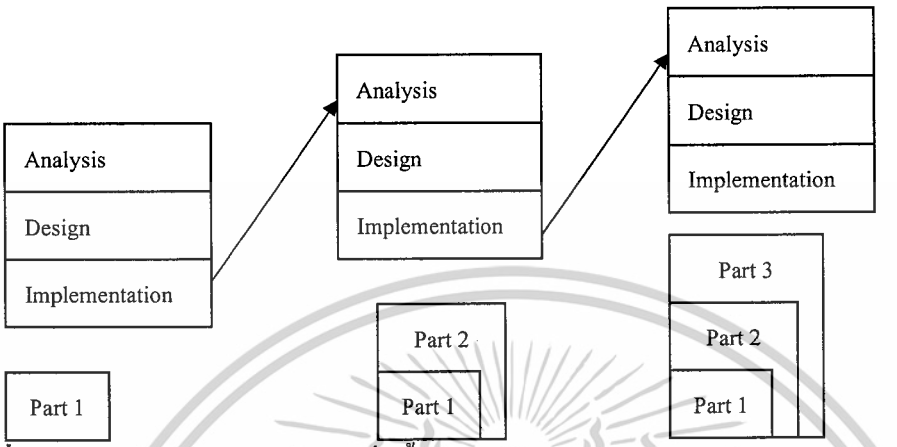
รูปที่ 2.12 วงจรการพัฒนาระบบแบบวิวัฒนาการ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 10)

2.2.1.5 วงจรการพัฒนาระบบแบบเพิ่มขึ้น (SDLC แบบ Incremental)

เป็นรูปแบบที่คล้ายกับแบบวิวัฒนาการ แต่มีข้อแตกต่างที่ตัวงาน เนื่องจากการพัฒนาขั้นแรกจะยังไม่ใช่งานที่สมบูรณ์ เป็นแค่ส่วนแรกเท่านั้นจากตัวระบบทั้งหมด จนเมื่อมีการพัฒนาใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
04764

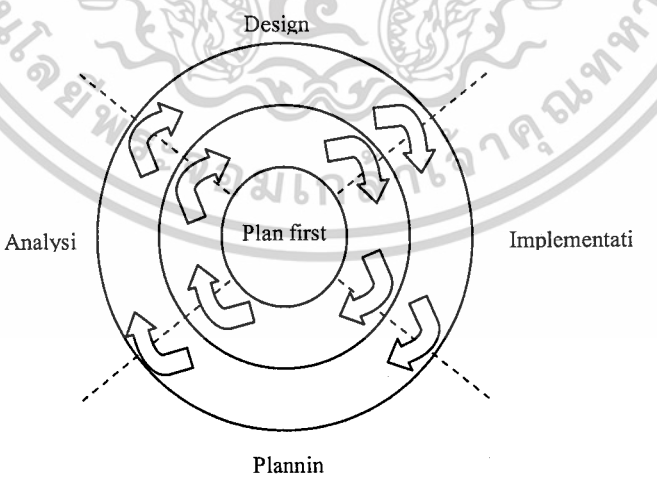
ขั้นตอนต่อไป จึงได้งานในส่วนที่ 2 เพิ่มเติมเข้าไป และจะมีการเพิ่มส่วนอื่นเข้าไปจนกลายเป็นระบบที่สมบูรณ์ ดังรูปที่ 2.13 ยังไงก็ตามยังไม่สามารถแน่ใจว่า ระบบที่เกิดขึ้นเป็นระบบที่สมบูรณ์หรือเปล่า อาจจะต้องประยุกต์ใช้กับวงจรการพัฒนาระบบรูปแบบอื่นๆด้วย



รูปที่ 2.13 วงจรการพัฒนาระบบแบบเพิ่มขึ้น (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 10)

2.2.1.6 วงจรการพัฒนาระบบแบบก้นหอย (SDLC แบบ Spiral)

เป็นการพัฒนาที่คล้ายกับก้นหอย โดยจะทำงานตามขั้นตอน และจะวนกลับมาเรื่อยๆ จนกระทั่งได้ระบบที่สมบูรณ์ การพัฒนารูปแบบนี้มีความยืดหยุ่นมากที่สุด เนื่องจากการทำงานใน 1 วงรอบนั้น ไม่จำเป็นต้องได้ระบบ หรือส่วนของระบบที่แน่นอน แต่ละขั้นตอนจะสั้นหรือยาวเท่าไรก็ได้ และอาจจะมีการข้ามขั้นตอนได้อีกด้วย ดังรูปที่ 2.14

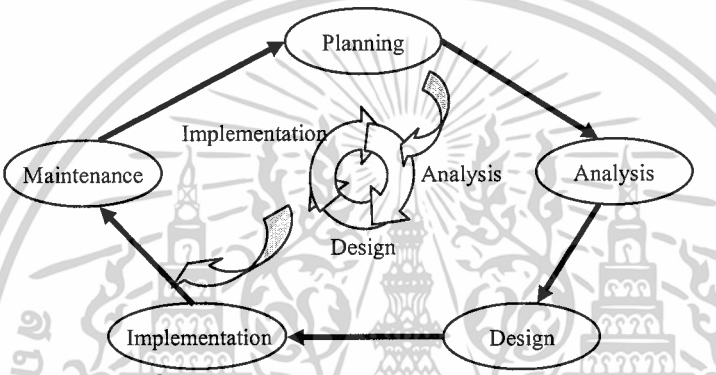


รูปที่ 2.14 วงจรการพัฒนาระบบแบบก้นหอย(กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 11)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 แนวทางการพัฒนาระบบที่เชิงวัตถุใช้

แนวทางการพัฒนาระบบที่เชิงวัตถุใช้คือ การวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design Methodology) ซึ่งเป็นการพัฒนาระบบโดยนำหลักการเชิงวัตถุที่เน้นการใช้ยูเอ็มเอล (UML: Unified Modeling Language) มาประยุกต์ใช้ร่วมกับวงจรการพัฒนาระบบ ในรูปแบบการพัฒนาแบบทำซ้ำ และเพิ่มขึ้น (Iterative and Incremental Development) โดยรูปแบบวงจรพัฒนาระบบที่ใช้ แบ่งออกเป็น 5 ขั้นตอน ได้แก่ การวางแผน (Planning) การวิเคราะห์ (Analysis) การออกแบบ (Design) การพัฒนาและติดตั้ง (Implementation) และการซ่อมบำรุง (Maintenance) ดังรูปที่ 2.15



รูปที่ 2.15 รูปแบบวงจรการพัฒนาระบบที่เลือกใช้

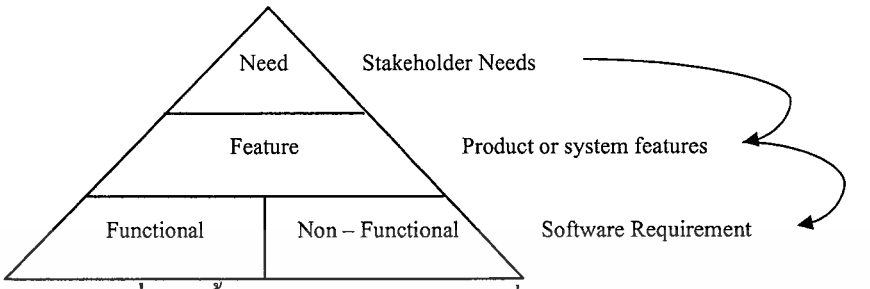
จากรูปที่ 2.15 จะเห็นได้ว่าจะแบ่งออกเป็น 2 ส่วนคือ วงรอบด้านนอก และวงรอบด้านใน โดยวงรอบด้านนอกนั้นคือ วงรอบของวงจรการพัฒนาระบบที่มีลักษณะของการทำซ้ำ ซึ่งเมื่อพัฒนาระบบได้ในชุดแรก และใช้งานระดับหนึ่ง ก็สามารถเริ่มต้นพัฒนาระบบในชุดต่อไปได้ วงรอบด้านในนั้นคือ ในการพัฒนาแต่ละชุดของวงรอบด้านนอก จะมี 3 ขั้นตอนที่ต้องวนรอบทำซ้ำและทำเพิ่ม ได้แก่ การวิเคราะห์ การออกแบบ การพัฒนาและติดตั้ง โดยเมื่อเริ่มต้นพัฒนาในแต่ละชุด หลังจากที่สิ้นสุดการดำเนินงานในขั้นตอนการวางแผนพัฒนาระบบแล้ว ก็จะเข้าสู่การวนรอบทำซ้ำและทำเพิ่ม (ต้องมีการวางแผนจำนวนรอบที่จะทำซ้ำและสิ่งที่จะทำเพิ่มด้วย) ตามจำนวนรอบที่กำหนดไว้ ในแต่ละรอบด้านในจะต้องทำการเพิ่มฟังก์ชันต่างๆจนครบ และได้ระบบงานชุดนั้นมา

2.2.3 การจัดการความต้องการ (Manage Requirement)

เป็นการจัดการความต้องการของระบบ เพื่อให้ได้มาซึ่งความต้องการของระบบที่แท้จริง (Software Requirement) เริ่มที่การสำรวจความต้องการของระบบ คือ การที่ทีมงานต้องไปเก็บรวบรวมข้อเท็จจริง และสารสนเทศข้อมูลต่างๆ มาศึกษาหาปัญหา นำมาวิเคราะห์ เพื่อหาความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการสูงสุดของผู้เกี่ยวข้องก่อนที่จะนำความต้องการเหล่านั้นมาทำให้เกิดความต้องการของระบบ ดังรูปที่ 2.16



รูปที่ 2.16 ขั้นตอนการหาความต้องการที่แท้จริงของระบบ

ขั้นตอนในการสำรวจความต้องการที่แท้จริงของระบบนั้น จากการเก็บรวบรวมข้อมูลต่างๆ นำมาศึกษา และวิเคราะห์หาปัญหาที่เกิดขึ้น ได้มาซึ่งปัญหาหลักที่เราจะนำไปแก้ไข ต่อมาเราจะเข้าไปดูถึงความต้องการระดับสูง หรือความต้องการของผู้เกี่ยวข้อง (Stakeholder Needs) ซึ่งเป็นความต้องการระดับแรกที่ได้มาจากผู้ที่มีส่วนเกี่ยวข้องกับระบบนี้ หลังจากนั้นต้องหาคุณลักษณะของผลิตภัณฑ์หรือระบบ (Product or system features) ที่จะมารองรับต่อความต้องการของผู้เกี่ยวข้อง หลังจากที่ได้คุณลักษณะมาแล้วเราจะต้องนำคุณลักษณะที่ได้มานั้น ทำเป็นความต้องการของระบบที่แท้จริง หรือเป็นความต้องการที่เข้ามาตอบสนองกับระบบ โดยแบ่งออกเป็นหน้าที่ของระบบ (Functional) ที่จำเป็นต้องทำ และส่วนประกอบของระบบ (Non – Functional, Supplementary) ที่จะเป็นต้องมี

2.2.3 ยูเอ็มแอล(UML: Unified Modeling Language)

ยูเอ็มแอลคือภาษารูปภาพหรือสัญลักษณ์ (Graphical Language) ที่ใช้ถ่ายทอดระบบออกมาเป็นแผนภาพ ประกอบไปด้วยรูปภาพหรือสัญลักษณ์มากมายตามรูปแบบของการสร้าง หรือยูเอ็มแอลก็คือภาษาสำหรับสร้างแบบจำลองของระบบ ที่พัฒนาเพื่อใช้ในการวิเคราะห์และออกแบบเชิงวัตถุโดยเฉพาะ

แผนภาพคือส่วนที่ทำหน้าที่รวบรวมนำสัญลักษณ์ต่างๆเข้าไว้ที่เดียวกันภายใต้ข้อกำหนดที่มี โดยภาษายูเอ็มแอลมีการแบ่งแผนภาพออกเป็น 2 กลุ่ม ดังตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ประเภทของแผนภาพในยูเอ็มแอล

Diagrams
1. Structural Diagram
- Class Diagram
- Object Diagram
- Component Diagram
- Deployment Diagram
2. Behavioral Diagram
- Use Case Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
-Activity Diagram

กลุ่มแผนภาพเชิงโครงสร้าง (Structural Diagram) เป็นกลุ่มแผนภาพที่แสดงให้เห็นโครงสร้างเชิงสถิติของระบบโดยจะไม่มีการเปลี่ยนแปลงหรือเคลื่อนไหวเมื่อมีเหตุการณ์ใดๆ เกิดขึ้น

กลุ่มแผนภาพเชิงพฤติกรรม (Behavioral Diagram) เป็นกลุ่มแผนภาพที่แสดงให้เห็นโครงสร้างเชิงกิจกรรมของระบบโดยจะแสดงให้เห็นพฤติกรรมของระบบที่มีการเปลี่ยนแปลงไปเมื่อมีเหตุการณ์ใดๆเกิดขึ้น และแสดงให้เห็นถึงการดำเนินการในหน้าที่บางอย่างของระบบ

การพัฒนาระบบไม่ได้นำแผนภาพคอมโพเนนต์ (Component Diagram) และแผนภาพดีพลอยเม้นต์ (Deployment Diagram) มาใช้โดยแผนภาพอื่นๆ 7 แผนภาพมีคุณลักษณะที่แตกต่างกันดังนี้

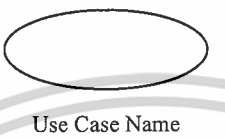
2.2.3.1 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคส เป็นแผนภาพที่จะบ่งบอกให้ทราบว่าระบบมีการทำงาน หรือมีหน้าที่อะไรบ้าง โดยนำมาจากการเก็บรวบรวมความต้องการที่แท้จริงของระบบ จะได้มาจากความต้องการของระบบที่แท้จริงในส่วนของหน้าที่ของระบบ ทำให้เห็นถึงขอบเขตหน้าที่ของระบบได้ชัดเจน โดยมีส่วนประกอบที่สำคัญ 4 ส่วนคือ ยูสเคส (Use Case) ผู้กระทำ (Actor) ขอบเขตระบบ (System Boundary) ความสัมพันธ์ (Relationship)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ยูสเคส

ยูสเคส คือ หน้าที่ที่ระบบต้องกระทำ (Functionality) ซึ่งได้มาจากการหาความต้องการที่แท้จริงแล้วได้หน้าที่ของระบบ โดยระบบดังกล่าวจะต้องเป็นระบบที่เราสนใจ ปกติแล้วจะเป็นผู้ที่ทำให้ระบบต้องมีหน้าที่ต่างๆเกิดขึ้นเพื่อตอบสนองต่อผู้กระทำ หรือบางหน้าที่ก็ต้องการข้อมูลบางอย่างจากผู้กระทำ โดยให้ใช้บรรทัดฐานที่ว่า สิ่งที่ทำหน้าที่ดำเนินกิจกรรมของระบบ หรือทำให้เกิดผลลัพธ์ต่างๆ ขึ้นในระบบ ให้ถือว่าสิ่งนั้นอยู่ในระบบ เรียกว่า "ยูสเคส" ดังรูปที่ 2.17



รูปที่ 2.17 สัญลักษณ์ยูสเคส (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 174)

2. ผู้กระทำ

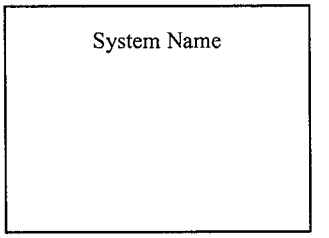
ผู้กระทำ คือผู้ที่เกี่ยวข้องกับระบบ โดยเป็นผู้คาดหวังผลลัพธ์จากระบบ ทำหน้าที่ผลักดันให้เกิดกิจกรรมของระบบ ทำหน้าที่ควบคุมดูแลกิจกรรมของระบบ หรือสัมผัสระบบโดยตรง อาจจะเป็นได้ทั้งบุคคล กลุ่มบุคคล หรือหน่วยงานที่อยู่นอกระบบ หรืออุปกรณ์ ระบบอื่นๆ เป็นต้น ดังรูปที่ 2.18 ผู้กระทำนั้นแบ่งได้เป็น 2 ประเภท ได้แก่ ประเภทแรกคือ ผู้กระทำหลัก (Primary Actor) คือผู้กระทำที่สัมผัสกับระบบโดยตรง เช่น ผู้ควบคุมดูแลกิจกรรมของระบบ หรือผู้ที่ได้รับประโยชน์จากระบบโดยตรง ประเภทที่สองคือ ผู้กระทำร่วม (Stakeholder Actor) คือผู้กระทำที่เป็นผู้ให้ข้อมูลข่าวสาร หรือผู้กระตุ้นให้เกิดยูสเคส หรือผู้ที่คาดหวังผลลัพธ์จากระบบ

รูปที่ 2.18 สัญลักษณ์ผู้กระทำ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 174)

3. ขอบเขตระบบ

ขอบเขตระบบ คือเส้นแบ่งที่กั้นระหว่างระบบ กับผู้กระทำต่อระบบ มีสัญลักษณ์เป็นสี่เหลี่ยม มีชื่อระบบกำกับไว้ด้านบนดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 สัญลักษณ์ขอบเขตระบบ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 174)

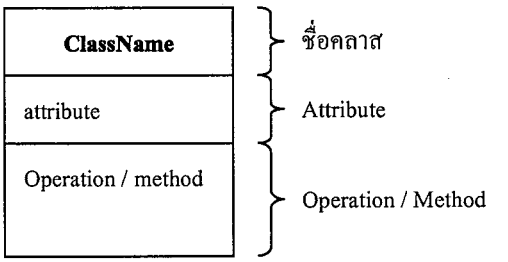
4. ความสัมพันธ์

ความสัมพันธ์ คือ เส้นที่เชื่อมต่อระหว่างผู้กระทำ กับยูสเคสที่มีปฏิสัมพันธ์กัน โดยใช้เส้นตรงมีหัวลูกศร และเส้นที่เชื่อมต่อระหว่างยูสเคส กับยูสเคสที่มีความสัมพันธ์กัน โดยใช้เส้นประมีหัวลูกศรมีด้วยกัน 2 แบบ คือ แบบที่หนึ่ง ความสัมพันธ์แบบขยาย (Extend Relationship) เกิดในกรณีที่ยูสเคสดำเนินกิจกรรมของตนเองไปตามปกติแต่อาจจะมีเงื่อนไข หรือสิ่งกระตุ้นบางอย่างที่ส่งผลให้ไปเรียกยูสเคสที่ทำหน้าที่กระตุ้น โดยลูกศรชี้จากยูสเคสที่ทำหน้าที่กระตุ้นไปหายูสเคสหลัก (อาจจะมีชื่อกำกับลูกศร <<Extend>>) แบบที่สองความสัมพันธ์แบบเรียกใช้ (Include Relationship) เกิดในกรณีที่ยูสเคสหนึ่งไปเรียกหรือดึงกิจกรรมของอีกยูสเคสหนึ่งมาใช้เพื่อให้กิจกรรมนั้นเกิดขึ้นในยูสเคสของตนเอง โดยลูกศรชี้จากยูสเคสหลักที่ไปเรียกไปยังยูสเคสที่ถูกเรียกใช้ (อาจจะมีชื่อกำกับลูกศร <<Include>>)

2.2.3.2 แผนภาพคลาส (Class Diagram)

แผนภาพคลาส คือแผนภาพที่แสดงให้เห็นโครงสร้างทางด้านคุณลักษณะของระบบ และโครงสร้างทางด้านการประมวลผลรวมอยู่ด้วยกันกับข้อมูลด้วย องค์ประกอบที่สำคัญของแผนภาพคลาสคือ “คลาส” ซึ่งเป็นสิ่งที่อธิบายแนวคิดของกลุ่มวัตถุ (Object) โดยการอธิบายคุณสมบัติ และพฤติกรรมของกลุ่มวัตถุ ซึ่งคลาสมีองค์ประกอบ 3 ส่วน คือ ชื่อคลาส (Class Name) คุณสมบัติ (Attribute) และการดำเนินการ (Operation/Method) ดังรูปที่2.20 และ ความสัมพันธ์ระหว่างคลาส 3 รูปแบบ คือ 1) ความสัมพันธ์ในคลาสเดียว (Unary Relationship) 2) ความสัมพันธ์ระหว่าง 2 คลาส (Binary Relationship) และ 3) ความสัมพันธ์ระหว่าง 3 คลาส (Ternary Relationship)ดังรูปที่ 2.21, 2.22, 2.23 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 สัญลักษณ์ของคลาสและองค์ประกอบ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 198)

1. ชื่อคลาส

ชื่อคลาส ต้องเป็นคำนามขึ้นต้นด้วยตัวพิมพ์ใหญ่ทุกคำ

2. คุณสมบัติ

1. เครื่องหมายสิทธิการเข้าถึง (Visibility) ได้แก่ " + " (Public) เข้าถึงได้ทั้งหมด " - " (Private) เข้าถึงได้เฉพาะคลาสนั้นๆ " # " (Protect) เข้าถึงได้เฉพาะคลาสที่สืบทอดกันและไม่มีเครื่องหมาย (Default) เข้าถึงได้เฉพาะในแพคเกจ(package) เดียวกัน
2. ชื่อคุณลักษณะ หรือชื่อของตัวแปรที่ใช้ในคลาสนั้นๆ
3. เครื่องหมายมหัพภาคคู่ (:) ไว้แบ่งระหว่าง ชื่อ และ ชนิดของคุณลักษณะ
4. ชนิดของคุณลักษณะ หรือประเภทของตัวแปรนั้น (Type) จะอยู่ต่อจากเครื่องหมายมหัพภาคคู่ ขึ้นต้นด้วยตัวพิมพ์ใหญ่ แบ่งเป็น 2 ชนิด ได้แก่ แบบพื้นฐาน (Primitive Type) หมายถึง คุณสมบัติของคุณลักษณะ แบบคลาส (Class Type) หมายถึง คุณลักษณะของคลาสหนึ่งอาจจะมีประเภทหรือชนิดเป็นคลาสอื่น

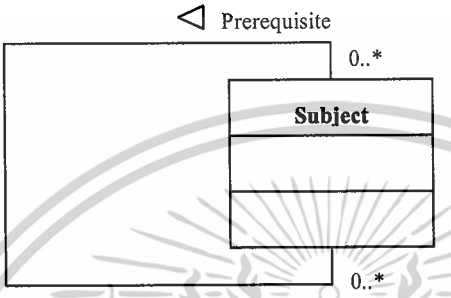
3. การดำเนินการ

การดำเนินการ คือ บริการที่คลาสต้องมี เพื่อให้คลาสอื่นๆเรียกใช้บริการได้ และยังอาจหมายถึงการกระทำที่มีผลกระทบโดยตรงต่อคุณลักษณะของคลาส เช่นเดียวกับคุณลักษณะ คือจะต้องถูกกำกับด้วยเครื่องหมายสิทธิการเข้าถึง ที่แตกต่างก็คือชื่อ เพราะชื่อซ้ำกันได้แต่มีได้ต่อเมื่อกรณีที่หนึ่ง จำนวนของตัวแปร (Parameter) แตกต่างกัน ไม่สนว่าชนิดของตัวแปร (Type) คืออะไร โดยไม่สนใจว่าชนิดของค่าที่ส่งกลับ (Return Type) เหมือนหรือต่างกัน กรณีที่สอง จำนวนตัวแปรเท่ากัน แต่ชนิดของตัวแปรแตกต่างกันโดยไม่สนใจว่าชนิดของค่าที่ส่งกลับ (Return Type) เหมือนหรือต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ความสัมพันธ์ในคลาสเดียว

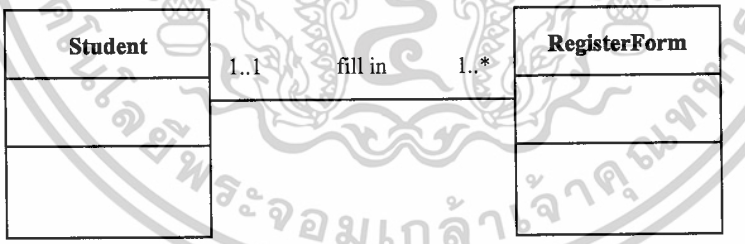
ความสัมพันธ์ในคลาสเดียว เป็นความสัมพันธ์ที่เกิดขึ้นภายในคลาสเดียว หรือคลาสมีความสัมพันธ์กับตัวเอง (เรียกได้อีกอย่างหนึ่งว่า " Recursive Relationship ") เกิดในกรณีที่คุณลักษณะของคลาสนั้นสามารถสร้างความสัมพันธ์กับอีกคุณลักษณะหนึ่งภายในคลาสเดียวกัน ดังตัวอย่างในรูปที่ 2.21



รูปที่ 2.21 ตัวอย่างระดับความสัมพันธ์ระหว่างคลาสเท่ากับ 1(กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 204)

5. ความสัมพันธ์ระหว่าง 2 คลาส

ความสัมพันธ์ระหว่าง 2 คลาส มีคลาสที่ความสัมพันธ์ระหว่างกันจำนวน 2 คลาส เห็นได้ทั่วไป ดังรูปที่ 2.22

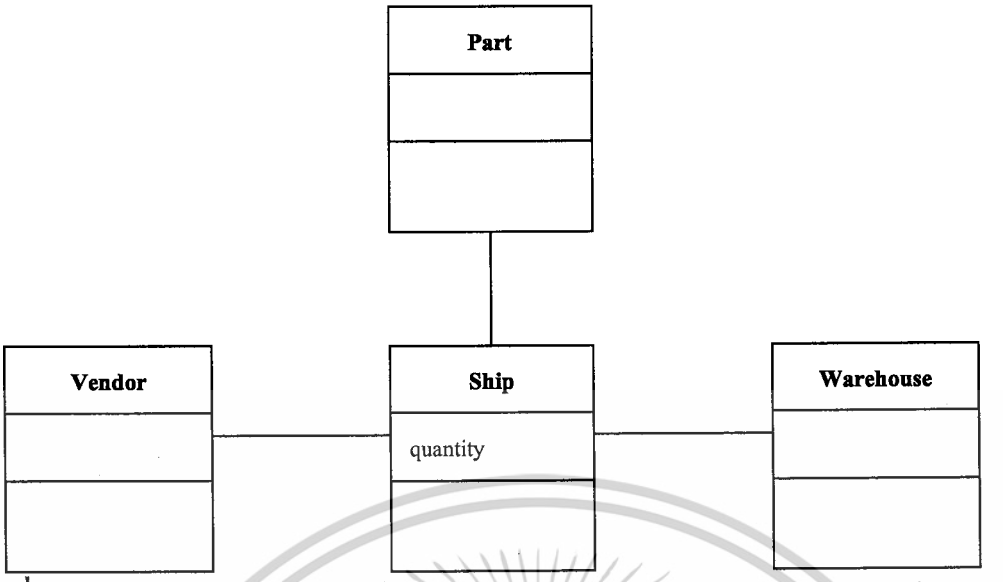


รูปที่ 2.22 ตัวอย่างระดับความสัมพันธ์ระหว่าง 2 คลาส กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 204)

6. ความสัมพันธ์ระหว่าง 3 คลาส

ความสัมพันธ์ระหว่าง 3 คลาส เป็นความสัมพันธ์ที่จะมีการสร้างคลาสหนึ่งมาเป็นตัวกลางระหว่างความสัมพันธ์ระหว่าง 3 คลาสที่ไม่สามารถใส่ความสัมพันธ์ได้ ดังรูปที่ 2.23

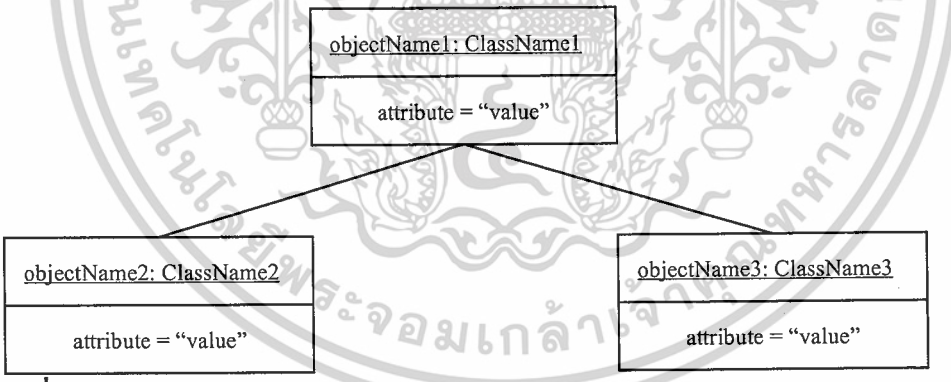
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 ตัวอย่างระดับความสัมพันธ์ระดับ 3 คลาส (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 205)

2.2.3.3 แผนภาพอ็อบเจกต์ (Object Diagram)

แผนภาพอ็อบเจกต์ เป็นแผนภาพที่ใช้ในการแสดงกลุ่มของวัตถุ (Object) และความสัมพันธ์ระหว่างวัตถุที่เกิดขึ้นในคลาสต่างๆของแผนภาพคลาส ดังรูปที่ 2.24



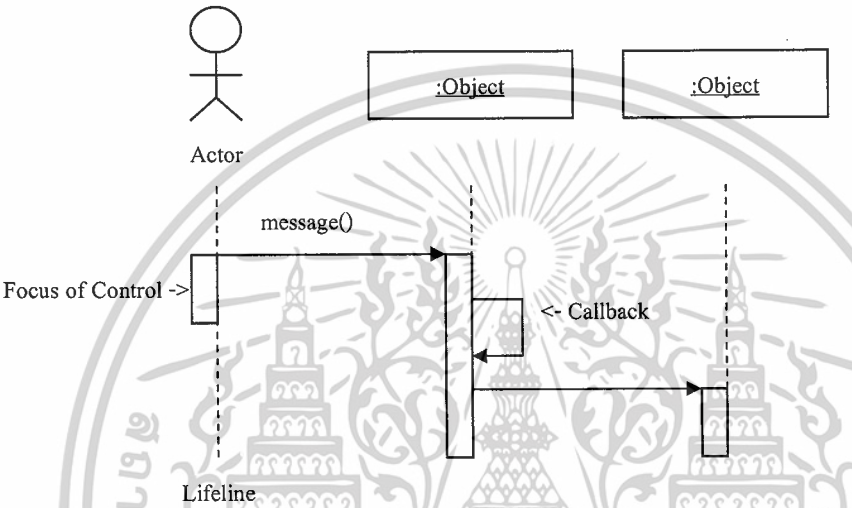
รูปที่ 2.24 ตัวอย่างแผนภาพอ็อบเจกต์ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 92)

จะเห็นได้ว่าสัญลักษณ์ของอ็อบเจกต์นั้นคล้ายกับคลาสเพียงแต่ต้องเริ่มต้นด้วยชื่ออ็อบเจกต์ และต้องขีดเส้นใต้ชื่ออ็อบเจกต์กับคลาสนั้นๆด้วย ส่วนการเชื่อมต่ออื่นๆระหว่างอ็อบเจกต์นั้น ต้องดูจากแผนภาพคลาส แล้วนำมาประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.4 แผนภาพลำดับเหตุการณ์ (Sequence Diagram)

แผนภาพลำดับเหตุการณ์ เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจกต์ของคลาส ตามลำดับเวลา ให้ทราบว่ามีการส่งสาร (Message) หากันอย่างไร สารใดเกิดก่อน สารใดเกิดหลัง โดยจะนำแผนภาพยูสเคสมาเป็นพื้นฐานในการสร้าง โดยจำนวนของแผนภาพลำดับเหตุการณ์จะเท่ากับจำนวนของยูสเคสหลัก แผนภาพลำดับเหตุการณ์มีสัญลักษณ์ต่างๆดังรูปที่ 2.25



รูปที่ 2.25 สัญลักษณ์ภายในแผนภาพลำดับเหตุการณ์ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 93)

ส่วนประกอบของแผนภาพลำดับเหตการณ์

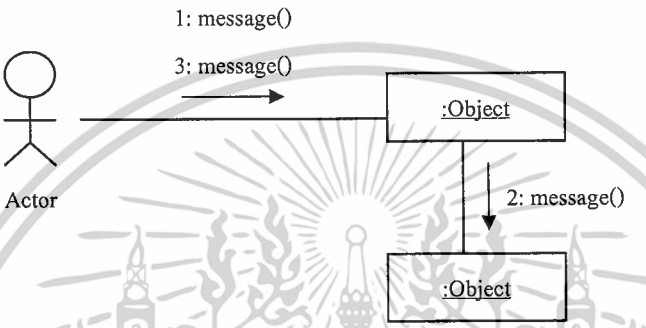
1. ผู้กระทำ เป็นผู้กระทำเดียวดันกับ ผู้กระทำที่มีปฏิสัมพันธ์กับยูสเคส ในแผนภาพยูสเคส
2. อ็อบเจกต์ เป็นอ็อบเจกต์ที่ทำหน้าที่ตอบสนองต่อผู้กระทำ
3. เส้นชีวิต (Lifeline) แสดงชีวิตของอ็อบเจกต์ หรือคลาสนั้นๆ
4. กิจกรรมที่ดูแล (Focus of Control / Activation) แสดงจุดเริ่มต้น และสิ้นสุดของการทำงานในแต่ละกิจกรรม ของแต่ละอ็อบเจกต์นั้นๆ
5. สาร เป็นคำสั่งหรือฟังก์ชันที่อ็อบเจกต์หนึ่งส่งให้อีกอ็อบเจกต์หนึ่ง ไม่ว่าส่งไปหรือส่งกลับ
6. เรียกกลับ (Callback / Self Delegation) เป็นการประมวลผล และคืนค่าที่ได้ให้กลับอ็อบเจกต์เดียวกัน หรือตัวเอง

โดยแผนภาพลำดับเหตุการณ์นั้นจะช่วยให้ทราบว่าคลาสใดมีการดำเนินการใดบ้าง จากการส่ง-รับสารกันระหว่างอ็อบเจกต์ และทำให้เห็นผู้กระทำปฏิบัติ กับระบบปฏิบัติแยกกันชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.5 แผนภาพการปฏิสัมพันธ์ (Collaboration Diagram)

แผนภาพปฏิสัมพันธ์ เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) กันระหว่างอ็อบเจกต์เช่นเดียวกับแผนภาพลำดับเหตุการณ์ โดยจะไม่เน้นที่การแสดงลำดับการส่งสาร แต่จะแสดงในส่วนของลักษณะการทำงานชัดเจน สัญลักษณ์ต่างๆจะเหมือนกับแผนภาพลำดับเหตุการณ์ ส่วนการส่งสารไปมาระหว่างอ็อบเจกต์นั้นไม่เน้นเรื่องลำดับเวลา ดังนั้นไม่ต้องเรียงตามลำดับเวลาก็ได้ ดังรูปที่ 2.26



รูปที่ 2.26 ตัวอย่างแผนภาพการปฏิสัมพันธ์ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 269)

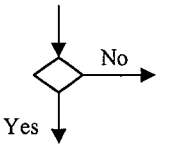
2.2.3.6 แผนภาพกิจกรรม (Activity Diagram)

แผนภาพกิจกรรม เป็นแผนภาพที่แสดงให้เห็นถึงลำดับการดำเนินกิจกรรม (Activity) จากกิจกรรมหนึ่งไปยังอีกกิจกรรมหนึ่งที่เกิดจากการทำงานของอ็อบเจ็กต์ภายในระบบ ซึ่งจะแตกต่างกับแผนภาพลำดับเหตุการณ์ตรงที่ แผนภาพกิจกรรมนั้นสามารถแสดงให้เห็นถึงกิจกรรมที่มีอยู่ เมื่อเกิดข้อผิดพลาดได้ ในขณะที่แผนภาพลำดับเหตุการณ์จะแสดงเพียงกิจกรรมเมื่อไม่มีข้อผิดพลาดเกิดขึ้นเท่านั้น ลักษณะของแผนภาพกิจกรรมนั้นจะคล้ายกับผังงาน (Flowchart) แผนภาพกิจกรรมนั้นจะแสดงถึงลำดับของกิจกรรมได้ 3 แบบ คือ

1. แบบทางเลือกตัดสินใจ

เป็นรูปแบบการทำงานเพื่อกิจกรรมที่ต้องมีการแบ่งออกเป็น 2 กรณีให้เลือก โดยลากลูกศรไปหาข้าวหลามตัด แล้วก็ลากไปยังแต่ละทางที่เลือก ดังรูปที่ 2.27

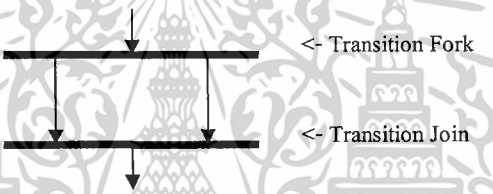
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 สัญลักษณ์แผนภาพกิจกรรม แบบทางเลือกตัดสินใจ(กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 98)

2. แบบทำงานพร้อมกัน

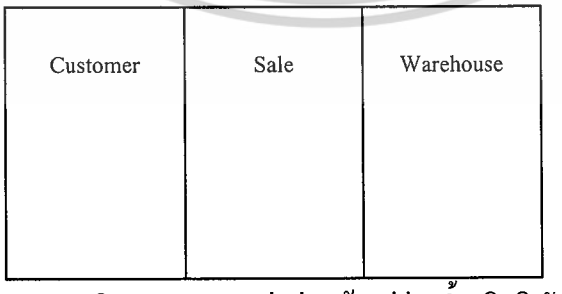
เป็นรูปแบบการทำงานเพื่อกิจกรรมที่มีการทำงานพร้อมกันจะมีจุดเปลี่ยนแยก (Transition Fork) โดยลากลูกศรเข้าสู่เส้นตรงที่ขวางแนวนอนก่อนที่จะลากลูกศรแยกไปยังกิจกรรมที่ต้องทำพร้อมกัน เมื่อเสร็จสิ้นการทำงานพร้อมกันของแต่ละกิจกรรมแล้วก็จะมีจุดเปลี่ยนรวม (Transition join) ซึ่งจะนำลูกศรจากกิจกรรมที่แยกกันลากเข้าสู่เส้นตรงขวางแนวนอนแล้วจึงลากลูกศรรวมไปกิจกรรมอื่นต่อไป ดังรูปที่ 2.28



รูปที่ 2.28 สัญลักษณ์แผนภาพกิจกรรม แบบทำงานพร้อมกัน (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 99)

3. แบบแบ่งส่วนด้วยลู่ว่ายน้ำ (Swimlanes)

แบบแบ่งส่วนด้วยลู่ว่ายน้ำ เป็นการแบ่งส่วนการทำงานออกเป็นหลายส่วนที่เกี่ยวข้องกันใน 1 กิจกรรม โดยแบ่งเป็นแนวตั้งเป็นลู่สระแต่ละส่วน ซึ่งอาจจะเป็นได้ทั้งบุคคล หน่วยงาน หรือสถานที่ ดังรูปที่ 2.29

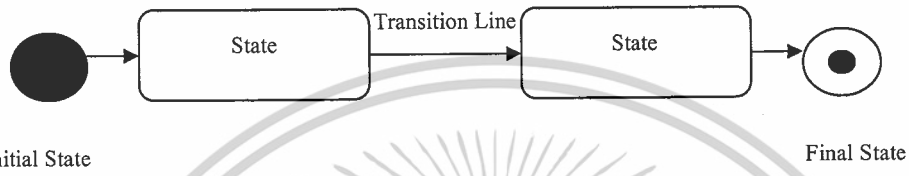


รูปที่ 2.29 สัญลักษณ์แผนภาพกิจกรรม แบบแบ่งส่วนด้วยลู่ว่ายน้ำ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 100)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.7 แผนภาพสถานะ (Statechart Diagram)

แผนภาพสถานะ เป็นแผนภาพที่แสดงให้เห็นถึงสถานะ และการเปลี่ยนสถานะของเหตุการณ์ในอ็อบเจกต์ 1 ช่วง (1 Sequence) เพื่อใช้ในการพิจารณาสถานะของอ็อบเจกต์นั้นๆได้ดียิ่งขึ้น เหมาะสำหรับการอธิบายสถานะของอ็อบเจกต์ที่มีสถานะจำนวนมาก และอธิบายสถานะของอ็อบเจกต์ที่ถูกใช้งานข้ามยูสเคสหลายยูสเคส โดยสัญลักษณ์จะเหมือนกับแผนภาพกิจกรรม (คล้าย Flowchart แต่เป็นแนวนอน) ดังรูปที่ 2.30



รูปที่ 2.30 ตัวอย่างแผนภาพสถานะ (กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548 : 96)

2.2.4 การตรวจสอบคุณภาพ (Verify Quality)

การตรวจสอบคุณภาพเป็นการวัดระบบที่เราได้พัฒนาขึ้นมา ว่ามีรายละเอียดความสามารถตามความต้องการหรือไม่ โดยระบบที่เราพัฒนาจะต้องมีหน้าที่ของระบบ (Functional: สิ่งที่ระบบจำเป็นต้องทำ) ครบตามความต้องการที่แท้จริงเป็นอันดับแรก ส่วนการวัดคุณภาพจะดูจากในส่วนของส่วนประกอบของระบบ (Non – Functional, Supplementary: สิ่งที่ระบบจำเป็นต้องมี) ว่าเรามีการพัฒนาระบบให้รองรับกับความต้องการของส่วนประกอบระบบหรือไม่ ซึ่งจะมีการกำหนดเงื่อนไขต่างๆในการทดสอบ ออกแบบการทดลอง ทดสอบ และวิเคราะห์ผลที่ได้ ซึ่งจะมีการนำเสนอในบทที่ 4 ต่อไป

2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ

โครงงานนี้ทดลองใช้ชุดหุ่นยนต์เลโก้ มายสตอร์มรุ่นอาร์ซีเอ็กซ์(Lego Mindstorm RCX) มาเป็นเครื่องมือทางฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ ซึ่งในอาร์ซีเอ็กซ์นั้นสามารถนำมาประกอบเป็นหุ่นยนต์ได้หลายรูปแบบตามจินตนาการ และมีระบบฝังตัวคือ มีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานภายในอาร์ซีเอ็กซ์นั้น ด้านซอฟท์แวร์ได้ใช้ บริคโอเอส(Brick OS) ซึ่งเปรียบเสมือนเป็นคลังชุดคำสั่งพื้นฐานของอาร์ซีเอ็กซ์ในการโปรแกรม ร่วมกับโปรแกรม Cygwin ในการที่จะเป็นคอมไพล์เลอร์แบบไขว้ (Cross Compiler) ในการแปลงโปรแกรมที่เราเขียนในภาษาขั้นสูง ไปเป็นภาษาเครื่องที่อาร์ซีเอ็กซ์เข้าใจและทำงานตามที่โปรแกรมไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ

2.3.1.1 อาร์ซีเอ็กซ์ (RCX)

อาร์ซีเอ็กซ์ คือ อุปกรณ์หลักของเลโก้ (LEGO) รุ่น มายด์สตอร์ม (Mindstorms) ที่ใช้เป็นเอ็มพียู(MPU) โดยมี ไมโครคอนโทรลเลอร์ รุ่น H8 / 3292 ซึ่งใช้หน่วยประมวลผลกลาง (CPU) รุ่น H8 / 300 เป็นหลักในการประมวลผลและควบคุมอุปกรณ์ต่างๆ ที่นำมาติดต่อสื่อสารกับอาร์ซีเอ็กซ์ซึ่ง ไมโครคอนโทรลเลอร์ H8 / 3292 ประกอบไปด้วยองค์ประกอบ 3 ส่วนหลักๆ คือ หน่วยประมวลผลกลาง (CPU) หน่วยความจำ (Memory) อินพุต / เอาท์พุต (Input / Output) ซึ่งองค์ประกอบเหล่านี้มีการติดต่อโดยผ่านบัสแอดเดรส (Address Bus) บัสข้อมูล (Data Bus) และบัสควบคุม (Control Bus)

1. หน่วยประมวลผลกลางรุ่น H8 / 300

หน่วยประมวลผลกลางรุ่น H8 / 300 นั้นมีความถี่ในการประมวลผล 16 เมกะเฮิร์ต(MHz) เป็นหน่วยประมวลผลกลาง ที่มีสถาปัตยกรรมทั่วๆไป คือ ประกอบไปด้วยเรจิสเตอร์ (Register) มีฟังก์ชั่นในการทำกระบวนการทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร ฯลฯ และ กระบวนการทางตรรกะ เช่น AND OR NOR NOT XOR การเลื่อนบิต (Shift Bit) ฯลฯ

2. หน่วยความจำ

หน่วยความจำบน H8 / 3292 นั้นประกอบไปด้วย รอม (หน่วยความจำอ่านอย่างเดียว) ขนาด 16 กิโลไบต์ (KByte) ซึ่งใช้ในการเก็บตัวขับ (Driver) ของอุปกรณ์ต่างๆในอาร์ซีเอ็กซ์รวมไปถึงเฟิร์มแวร์ (Firmware) ซึ่งใช้ในการแปลคำสั่งที่เราโปรแกรมเข้าไป และแรม (หน่วยความจำเข้าถึงโดยสุ่ม) ขนาด 32 กิโลไบต์ ซึ่งเป็นที่อยู่ของโปรแกรมที่เรานำเข้าไปประมวลผล กับชุดคำสั่งต่างๆที่ใช้ในการทำงานโปรแกรม

3. อินพุต / เอาท์พุต

อินพุต / เอาท์พุตของ H8/3292 นั้นมี 3 ประเภท คือ

ส่วนต่อประสานติดต่อสื่อสารอนุกรม (Serial Communication Interface) ใช้ในการควบคุมอุปกรณ์ ตัวรับตัวส่งการติดต่อสื่อสารอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปลงแอนะล็อกเป็นดิจิตอล (Analog-to-Digital Converter) ใช้ในการแปลงสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัลโดยวิธีการชักตัวอย่าง (Sampling) ออกมาเป็นข้อมูลขนาด 10 บิต และยังสามารถทำหน้าที่เป็นตัวควบคุมอุปกรณ์สำหรับช่องการสื่อสาร 8 ช่องทาง

อินพุตพอร์ต/เอาท์พุตพอร์ต (Input / Output Port) อินพุตพอร์ต/เอาท์พุตพอร์ต ถูกใช้เป็นส่วนต่อประสาน (Interface) กับอุปกรณ์ที่มาเชื่อมต่อกับอาร์ซีเอ็กซ์

บน H8/3292 นั้นมีกลไกในการขัดจังหวะ (Interrupt) สำหรับ การขัดจังหวะภายนอก และการขัดจังหวะภายใน ซึ่งถูกควบคุมการทำงานโดย ตัวควบคุมการขัดจังหวะ (Interrupt Control) ซึ่งการขัดจังหวะภายในนั้นถูกสร้างขึ้นมาโดยอุปกรณ์ อินพุต / เอาท์พุต แล้วแต่เหตุการณ์ เช่น ตัวจับเวลาเกิดการล้น (Timer Overflow) การขัดจังหวะนั้นจะทำผ่าน เวคเตอร์การขัดจังหวะ (Interrupt Vector) ของแต่ละเหตุการณ์นั้นๆที่เกิดขึ้นโดย เวคเตอร์การขัดจังหวะ นั้นจะทำหน้าที่ส่งการขัดจังหวะที่เกิดขึ้น ไปที่ตัวจัดการการขัดจังหวะ (Interrupt Handler) เพื่อที่จะจัดการกับการขัดจังหวะนั้นๆ ส่วนการขัดจังหวะภายนอกถูกสร้างขึ้นมาจากสัญญาณภายนอกโดยผ่านทางพิน (pin) ของไมโครคอนโทรลเลอร์

อุปกรณ์อินพุต / เอาท์พุต ประกอบไปด้วย 2 ส่วน คือ ดีไวซ์คอนโทรลเลอร์ (Device Controller) และอุปกรณ์อินพุต / เอาท์พุต เอง โดยดีไวซ์คอนโทรลเลอร์ จะทำหน้าที่ควบคุมการติดต่อสื่อสารของแต่ละตัวอุปกรณ์ และ ควบคุมการขัดจังหวะ

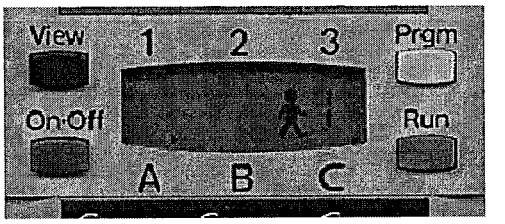
2.3.1.3 องค์ประกอบของอาร์ซีเอ็กซ์

องค์ประกอบของอาร์ซีเอ็กซ์ประกอบไปด้วย 6 องค์ประกอบหลักๆ คือ ปุ่ม, อินพุตพอร์ต (Input Port) และเซ็นเซอร์ (Sensor) ตัวรับ/ส่งอินฟราเรด (Infrared Transmitter / Receiver) หน้าจอแอลซีดี (จอภาพผลึกเหลว จอผลึกเหลว) ลำโพง เอาท์พุตพอร์ต (Output Port) และ แอกทูเอเตอร์ (Actuator)

1. ปุ่ม

ปุ่มบนอาร์ซีเอ็กซ์จะมีอยู่ 4 ปุ่มด้วยกันคือ “View, On-Off, Prgm, Run” ค่าบิตที่ได้รับจากการอ่านเป็น 0 เมื่อปุ่มถูกกดและเป็น 1 เมื่อปุ่มถูกปล่อย สำหรับค่าบิตในแต่ละปุ่มสามารถเข้าถึงได้โดยตรงจากอุปกรณ์รีจิสเตอร์โดยการจัดแจงบิต (Bit Manipulation) ดังรูปที่ 2.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 ลักษณะปุ่มต่างๆ บนตัวอาร์ซีเอ็กซ์ (Ole Caprani. 2549 . http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

ปุ่ม "View" มีไว้สำหรับดูสถานะต่างของอาร์ซีเอ็กซ์โดยผ่านทางหน้าจอแอลซีดี

ปุ่ม "On-Off" มีไว้สำหรับการปิด / เปิด อาร์ซีเอ็กซ์

ปุ่ม "Prgm" มีไว้สำหรับการอัพโหลดโปรแกรมเข้ามาในอาร์ซีเอ็กซ์

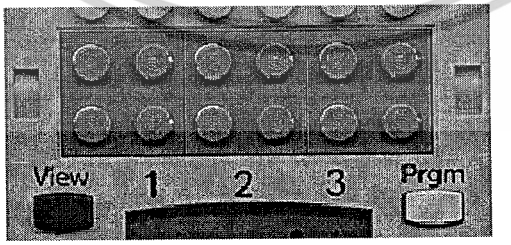
ปุ่ม "Run" มีไว้สำหรับดำเนินงานโปรแกรมที่เราโปรแกรมเข้าไป รวมไปถึงการหยุดลูปของโปรแกรม

ปุ่มทั้ง 4 ปุ่มนี้ได้ถูกเชื่อมต่อกับสายช่องทางเข้า (Input Line) ทั้ง 4 สายของช่องทางเข้า / ออกปุ่ม "Run" และปุ่ม "On-Off" นั้นสามารถเริ่มต้นด้วยการขัดจังหวะภายนอกได้ รายละเอียดสามารถดูได้จากตารางที่ 2.2

ตารางที่ 2.2 ความสัมพันธ์ของปุ่ม 4 ปุ่ม บนอาร์ซีเอ็กซ์

ปุ่ม	ช่องทางเข้า	ที่อยู่	ตำแหน่งบิต	การขัดจังหวะ
Run	port 4	0xffb7	bit 2	IRQ 0
On-Off	port 4	0xffb7	bit 1	IRQ 1
View	port 7	0xffbe	bit 6	no interrupt
Prgm	port 7	0xffbe	bit 7	no interrupt

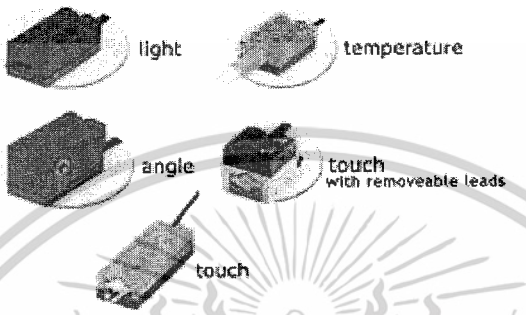
2. อินพุตพอร์ตและเซ็นเซอร์



รูปที่ 2.32 ลักษณะอินพุตพอร์ตของอาร์ซีเอ็กซ์ 1, 2 และ 3 (Ole Caprani. 2549 . http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็นเซอร์ในชุดทดลองนั้นมีด้วยกัน 2 ประเภทคือ เซ็นเซอร์แบบแพสซีฟ (Passive Sensor) ได้แก่ เซ็นเซอร์สัมผัส (Touch Sensor) กับ เซ็นเซอร์อุณหภูมิ (Temperature Sensor) และเซ็นเซอร์แบบแอ็กทีฟ (Active Sensor) ได้แก่ เซ็นเซอร์แสง (Light Sensor) กับ เซ็นเซอร์การหมุน (Rotation Sensor or Angle Sensor) จากมุมมองของการเขียนโปรแกรม ตัวรับรู้ทั้ง 2 ประเภทนี้มีลักษณะที่แตกต่างกันในการรับค่า



รูปที่ 2.33 ชนิดของเซ็นเซอร์แบบต่างๆ (Ole Caprani. 2549 .
http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

อินพุตพอร์ตของอาร์ซีเอ็กซ์ 3 ช่องทางถูกเชื่อมต่อกับช่องสัญญาณของตัวแปลงสัญญาณแอนะ-ล็อกไปเป็นดิจิทัล และส่งออกมาทางสายส่งออก (Output Line) ของตัวรับตัวส่งที่ 6 ซึ่งรายละเอียดสามารถดูได้จากตารางที่ 2.3

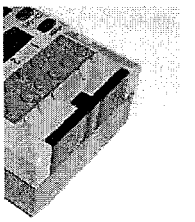
ตารางที่ 2.3 ความสัมพันธ์ระหว่างช่องทางเข้า และสายส่งออก

Input Port	**A / D Channel**	**Output Line**
1	Analog input pin 2 (AN2)	Port 6, bit 2
2	Analog input pin 1 (AN1)	Port 6, bit 1
3	Analog input pin 0 (AN0)	Port 6, bit 0

ค่าของเซ็นเซอร์ทั้ง 2 ประเภทที่ได้อ่านค่าเข้ามานั้น จะผ่านตัวแปลงสัญญาณแอนะล็อกเป็นดิ-จิทัลเพื่อเริ่มต้นการแปลง (Conversion) และแสดงค่าที่ได้หลังการแปลง ผลลัพธ์ที่ถูกแปลงจะมีค่าข้อมูลขนาด 10 บิตเพื่อใช้ในการประมวลผลค่าได้รับมาจากเซ็นเซอร์ที่ตัวประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตัวรับ / ส่ง อินฟราเรด

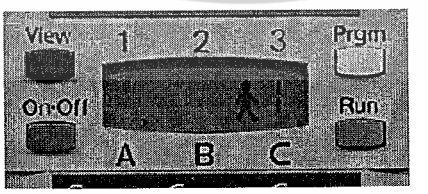


รูปที่ 2.34 ลักษณะตัวรับ / ส่ง อินฟราเรด ของอาร์ซีเอ็กซ์(Ole Caprani. 2549 . http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

อาร์ซีเอ็กซ์นั้นใช้แสงอินฟราเรดเพื่อติดต่อสื่อสารกับคอมพิวเตอร์หรืออาร์ซีเอ็กซ์ตัวอื่นๆ ตัวรับ / ตัวส่งอินฟราเรดนั้น ได้ถูกเชื่อมต่อกับส่วนต่อประสานการติดต่อสื่อสารแบบอนุกรม (Serial Communication Interface : SCI) ซึ่งทำหน้าที่เป็นตัวควบคุมอุปกรณ์สำหรับการติดต่อสื่อสารอินฟราเรด (Serial Infrared Communication) ดังรูปที่ 2.34 แสงอินฟราเรดนั้นเปิดและปิดด้วยความถี่ 38.5 กิโลเฮิร์ต(KHz) จะถูกใช้เป็นตัวขนส่งกระแสของบิต โดยค่าบิต 0 แทนช่วงเวลาตอนที่มีแสงอินฟราเรด (มีการส่งข้อมูล) ส่วนค่าบิต 1 แทนช่วงเวลาตอนที่ไม่มีแสงอินฟราเรด (ไม่มีการส่งข้อมูล)

คลื่นความถี่ของตัวขนส่งถูกกำหนดโดย(สิ่ง)ออก (Output) ของตัวจับเวลา (Timer) 1 การส่งข้อมูลโดยอินฟราเรดนั้นมีอยู่ด้วยกัน 2 โหมดคือโหมดพิสัยใกล้ (Short Range Mode) สำหรับติดต่อสื่อสารกับเครื่องคอมพิวเตอร์และ โหมดพิสัยไกล(Long Range Mode) สำหรับแสงอินฟราเรดที่ส่งออกมาไม่เพียงแต่ถูกรับโดยตัวรับอินฟราเรดของเครื่องคอมพิวเตอร์แล้ว แต่ยังถูกรับโดยตัวรับอินฟราเรดบนอาร์ซีเอ็กซ์อีกด้วย การขนส่งสามารถเกิดขึ้นพร้อมกันได้ทั้งสองทางหรือมากกว่านั้น เพราะฉะนั้นการชนกัน (collision) ของข้อมูลก็สามารถเกิดขึ้นได้ง่ายแต่ก็สามารถตรวจสอบได้โดยการตรวจสอบของกระแสบิตรับ (Inspection of The Received Bit Stream)

4. หน้าจอแอลซีดี



รูปที่ 2.35 ลักษณะหน้าจอแอลซีดีบนอาร์ซีเอ็กซ์(Ole Caprani. 2549 . http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

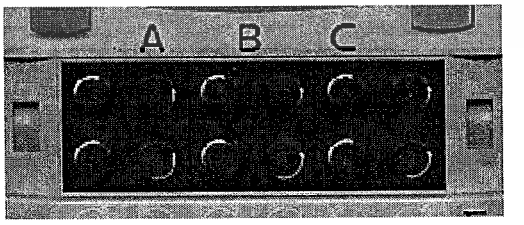
อาร์ซีเอ็กซ์มีหน้าจอแอลซีดี ซึ่งมีจำนวน 43 เซกเมนต์ (Segments) ซึ่งเซกเมนต์หนึ่งๆนั้นจะเป็นรูปภาพหรือสัญลักษณ์ เช่น จุด ลูกศร หรือขาของตัวคนตัวเล็กๆ ดังรูปที่ 2.35 เซกเมนต์แต่ละเซกเมนต์สามารถควบคุมเฉพาะตัวได้ โดยการเปิดหรือปิดตัวมันเองโดยตรงเลย แต่จะต้องผ่านตัวควบคุมหน้าจอแอลซีดี ซึ่งเป็นตัวที่ทำหน้าจัดการหน้าจอแอลซีดีให้สอดคล้องกับข้อมูลที่ได้รับเข้ามา พื้นที่ของหน่วยความจำจะมีแอลซีดีบัฟเฟอร์ (LCD Buffer) โดย 1 บิต สำหรับ1 เซกเมนต์ สำหรับรูทีน (Routine) ตัวจัดการอาร์ซีเอ็กซ์ (RCX Executive Routine) นั้นจะเขียนข้อมูลเกี่ยวกับการแสดงผลบนหน้าจอโดยใช้บัฟเฟอร์ และจะเรียกไปยังรูทีนตัวจัดการอาร์ซีเอ็กซ์ โดยจะใช้โปรโตคอลในการปรับ (ให้เป็นปัจจุบัน) หน้าจอ ดังนั้นเซกเมนต์จะปรากฏ หรือซ่อนก็จะสอดคล้องกับค่าบิตที่แท้จริงในแอลซีดีบัฟเฟอร์

5. ลำโพง

ลำโพงบนตัวอาร์ซีเอ็กซ์ถูกสร้างขึ้นมาเพื่อแสดงเสียงเพื่อ บอกสถานะการทำงานของอาร์ซีเอ็กซ์ เช่น เสียงปี๊บ (Beep) ลากยาวจะเกิดต่อเมื่อมีความผิดปกติเกิดขึ้นที่ตัวอาร์ซีเอ็กซ์ ลำโพงนั้นสามารถถูกควบคุมโดยผ่านช่องทางเข้า / ออกที่ 6 บิตที่ 4 และดาต้าไดเร็กรีจิสเตอร์ (Data direction register) ของช่องทางเข้า / ออกที่ 6 จะถูกตั้งค่าก่อนที่ลำโพงจะทำงานได้

6. เอาท์พุตพอร์ตและแอกทูเอเตอร์

อาร์ซีเอ็กซ์มีเอาท์พุตพอร์ต 3 ช่องซึ่งมีอักษรเอ (A) บี (B) และ ซี (C) กำกับดังรูปที่2.36 แอกทูเอเตอร์ชนิดต่างๆสามารถถูกเชื่อมต่อกับเอาท์พุตพอร์ต แอกทูเอเตอร์ของเลโก้มายด์สตรอมคือ มอเตอร์ หลอดไฟ และแหล่งกำเนิดเสียง ซึ่งเอาท์พุตพอร์ตแต่ละตัวสามารถขับเคลื่อนแอกทูเอเตอร์มากกว่า 1 ตัว เช่น ถ้าต่อมอเตอร์ และหลอดไฟ ไว้ที่ช่องเอาท์พุตพอร์ต เดียวกัน เมื่อมอเตอร์ทำงานหลอดไฟก็จะสว่างด้วย แต่เมื่อต่อกรณีเช่นนี้ จะไม่สามารถควบคุมมอเตอร์ และหลอดไฟให้ทำงานแยกกันได้ ซึ่งการต่อแบบนี้ส่วนใหญ่จะใช้ในการตรวจสอบการทำงานของอุปกรณ์ว่าเกิดการทำงานขึ้นจริงหรือไม่



รูปที่ 2.36 ช่องทางออกของอาร์ซีเอ็กซ์ A, B และ C (Ole Caprani. 2549 . http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุตพอร์ตนั้นถูกขับเคลื่อนโดยแบตเตอรี่ขนาด 9 โวลต์ที่อยู่ภายในตัวอาร์ซีเอ็กซ์ สถานะของกระแสไฟฟ้าที่ไหลผ่านแอกทูเอเตอร์ จะถูกควบคุมโดยข้อมูลที่เก็บอยู่ภายในรีจิสเตอร์อุปกรณ์เอาท์พุตพอร์ตของอาร์ซีเอ็กซ์ดีไวซ์เรจิสเตอร์ (RCX Output Ports Device Register) ซึ่งแต่เอาท์พุตพอร์ตจะถูกควบคุมโดยบิต 2 ตัวใน 8 บิตดีไวซ์เรจิสเตอร์ (8 Bit Device Register) ดังตารางที่ 2.4

ตารางที่ 2.4 หมายเลขบิตที่ควบคุมเอาท์พุตพอร์ตแต่ละช่องบนอาร์ซีเอ็กซ์

Output Port	**A**	**B**	**C**
Bits	bit 7 and 6	bit 3 and 2	bit 1 and 0

ในการควบคุมสถานะของกระแสไฟที่ไหลผ่านแอกทูเอเตอร์จะมี 4 สถานะ ที่เกิดจากบิตที่ควบคุม 2 บิต ค่า 01 และ 10 จะยอมให้กระแสไหลไปในคนละทิศทาง เพื่อการควบคุมการไหลของกระแส ส่วนค่า 00 นั้นจะไม่เชื่อมต่อแอกทูเอเตอร์กับตัวส่งพลังงาน และสุดท้ายค่า 11 นั้นจะเชื่อมต่อแอกทูเอเตอร์กับตัวส่งพลังงาน

การเรียกรูทีนของเอาท์พุตพอร์ตเอ จะตั้งค่าให้ตรงกับ 1 ใน 4 สถานะ (เรียกว่า Float OnPos OnNeg และ Brake) ซึ่งขึ้นอยู่กับค่าที่แท้จริงของสถานะพารามิเตอร์ (Parameter State)

ตารางที่ 2.5 ผลกระทบของสถานะบนแอกทูเอเตอร์ที่เชื่อมต่อเอาท์พุตพอร์ต

Output Bits	**Motor**	**Lamp**	**Flashlight**	**Sound Element**
00(float)	หมุน	ไม่มีแสง	ไม่มีไฟกระพริบ	ไม่มีเสียง
01(OnPos)	เลี้ยวไปในทิศทางแรก	มีแสง	มีไฟกระพริบ	มีเสียงแบบแรก
10(OnNeg)	เลี้ยวไปในทิศทางที่สอง	มีแสง	มีไฟกระพริบ	มีเสียงแบบที่สอง
11(Break)	หยุดหมุน	ไม่มีแสง	ไม่มีไฟกระพริบ	ไม่มีเสียง

เมื่อพิจารณาถึงมอเตอร์ที่ขับเคลื่อนโดยตัวรับรู้ที่เชื่อมต่อกับเอาท์พุตพอร์ตเอและซี เมื่อเรียก (Call) สถานะกระแสไฟ 4 สถานะ (Four Current States) สำหรับ “Float” “OnPos” OnNeg” และ “Break” ที่กล่าวมาแล้วสถานะ “OnPos” กับ “OnNeg” จะทำให้มอเตอร์หมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา ซึ่งขึ้นอยู่กับว่าสายไฟถูกต่อเข้ากับเอาท์พุตพอร์ต และเซ็นเซอร์อย่างไรและมีการโปรแกรมไว้อย่างไร การเปลี่ยนจากสถานะที่มีกำลังไฟฟ้า (Powered State) “OnPos” กับ “OnNeg” ไปยังสถานะที่ไม่มีไฟฟ้า “Float” หรือ “Break” มีผลกระทบที่แตกต่างกันมากต่อการหมุนของมอเตอร์ การเปลี่ยนเป็น “Break” จะบังคับให้มอเตอร์หยุดหมุนทันทีแต่ “Float” ยอมให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์หมุนได้อิสระ ถ้าสถานะบนเอาท์พุตพอร์ตมีการเปลี่ยนแปลงเกิดขึ้นอย่างรวดเร็วเพียงพอ เช่น “Float” “OnPos” “OnNeg” “Break” ด้วยระยะเวลา 1 มิลลิวินาที ต่อ 1 สถานะ จะส่งผลกระทบต่อหลอดไฟ คือหลอดไฟจะไม่สว่างเท่ากับหลอดไฟที่ทำงานปกติ โหมดควบคุมแอกทูเอเตอร์นี้ถูกเรียกว่า ตัวกล้ำสัญญาณพลัสไวด์ (PWM: Plus Width Modulator) แต่ละสถานะที่เปิดอยู่ถูกเรียกว่า พลัส (Plus) และระยะเวลาของพลัส ถูกเรียกว่าไวด์ (Width) ของพลัส พลังงานที่เคลื่อนที่ไปยังแอกทูเอเตอร์สามารถควบคุมโดยพลัสไวด์ (Plus Width) และรูปแบบของพลัส (รูปแบบของสัญญาณพัลส์) ในการนำอุปกรณ์ควบคุมช่องทางออก PWM ไปใช้หรือนำไปประยุกต์ เราสามารถใช้ตัวจับเวลา T0 ของ H8 / 3292 เพื่อจะกำหนดตัวจับเวลาการขัดจังหวะ (Timer Interrupts) ด้วยระยะเวลา เช่น 1 มิลลิวินาที การขัดจังหวะนี้สามารถถูกนำมาใช้เพื่อเรียก รูทีนภาษาซี (C - Routine) ซึ่งกำหนดรูปแบบปกติของสถานะที่เปลี่ยนแปลงไปของช่องทางออก ซึ่งสามารถถูกขับเคลื่อนที่ระดับพลังงานที่แตกต่างกัน 16 ระดับ ถูกกำหนดโดยรูปแบบพัลส์ที่แตกต่าง 16 รูปแบบ

2.3.2 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

ซอฟท์แวร์ที่ใช้ในการพัฒนาระบบหุ่นยนต์ตรวจจับเส้นทาง คือ บริคโอเอส(BrickOS) และซิกวิน(Cygwin)

2.3.2.1 บริคโอเอส (BrickOS)

บริคโอเอส เป็นระบบปฏิบัติการของระบบฝังตัว ที่ได้เตรียมการให้สามารถใช้ทรัพยากรต่างๆของชุดเลโก้มายด์สตรอมได้โดยใช้ภาษาซี (C) หรือ ภาษาซีพลัสพลัส (C++) ซึ่งเป็นภาษาที่นิยมใช้ในการเขียนชุดคำสั่งลงไปยัง เลโก้มายด์สตรอมรุ่นอาร์ซีเอ็กซ์ (RCX) ซึ่งบริคโอเอสเปรียบเสมือนคลัง (โปรแกรม Library) ซึ่งเป็นแหล่งเก็บรวบรวมโปรแกรมของคอมพิวเตอร์ที่มีชุดคำสั่งพื้นฐานอยู่ภายใน ทั้งนี้การที่เราเขียนโปรแกรมขึ้นมาเพื่อควบคุมอาร์ซีเอ็กซ์ เมื่อเราคอมไพล์โปรแกรมแล้ว เราไม่สามารถอัพโหลดโปรแกรมที่เราเขียนลงไปในอาร์ซีเอ็กซ์ได้โดยตรงเนื่องจากส่วนที่เราโปรแกรมเข้าไปและสิ่งที่อาร์ซีเอ็กซ์สามารถรับรู้ได้นั้น เป็นข้อมูลคนละรูปแบบกัน จึงมีการใช้ คอมไพล์เลอร์แบบไขว้ (Cross-Compiler) เข้ามาช่วยโดยจะกล่าวถึงในหัวข้อถัดไป

2.3.2.2 คอมไพล์เลอร์แบบไขว้(Cross-Complier)

คอมไพล์เลอร์แบบไขว้นั้น เป็นคอมไพล์เลอร์ชนิดหนึ่งที่ทำหน้าที่แปลงระบบปฏิบัติการที่เราโปรแกรมเข้าไปให้อยู่ในรูปแบบที่ระบบฝังตัวนั้นสามารถเข้าใจได้ ในที่นี้คือการแปลงระบบปฏิบัติการที่เราเขียนขึ้นมาจากภาษาซี หรือ ภาษาซีพลัสพลัส ให้อยู่ในรูปภาษาเครื่องที่อาร์ซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็กซ์นั้นเข้าใจ และทำงานตามที่โปรแกรมไว้ได้ ในส่วนของคอมไพล์เลอร์แบบไขว้ที่ใช้ในการพัฒนาอาร์ซีเอ็กซ์นั้น ส่วนมากจะนิยมใช้ จีซีซี (GCC) หรือ จีพลัสพลัส (G++) สำหรับในโครงงานนี้จะเลือกใช้ ซิกวิน(Cygwin) ในการเป็นคอมไพล์แบบไขว้

1. ซิกวิน (Cygwin)

ซิกวิน(Cygwin) คือระบบปฏิบัติการลีนุกซ์ (Linux) ที่ถูกพัฒนามาสำหรับระบบปฏิบัติการวินโดว์ ซึ่งประกอบไปด้วย 2 ส่วนส่วนแรก คือ DLL เป็นตัวที่เหมือนเป็น ชั้นของเอพีไอของลีนุกซ์ ซึ่งได้เตรียมฟังก์ชั่นต่างๆของลีนุกซ์เอาไว้ ส่วนที่ 2 คือ รูปแบบเครื่องมือต่างๆ ซึ่งเป็นเครื่องมือที่มีลักษณะรูปร่าง และความรู้สึกแบบลีนุกซ์ แต่ ซิกวิน(Cygwin) นั้นโดยปกติทั่วไปไม่ใช่ตัวที่ดำเนินการโปรแกรมของลีนุกซ์บนวินโดว์ ระบบปฏิบัติการจำเป็นต้องทำการสร้างมันขึ้นมาใหม่จากการเขียนโปรแกรมขึ้นมารองรับ ถ้าต้องการให้มันดำเนินการอยู่บนวินโดว์ และซิกวิน(Cygwin) ไม่ใช่ตัวพิเศษที่จะทำให้โปรแกรมบนวินโดว์รู้จักฟังก์ชั่นของยูนิกซ์ จำเป็นจะต้องทำการเขียนโปรแกรมขึ้นมารองรับอีกเช่นกัน ถ้าต้องการใช้ฟังก์ชั่นของซิกวิน(Cygwin) ซึ่งในการที่จะติดต่อกับอาร์ซีเอ็กซ์เพื่อทำการอัพโหลด (Upload) ข้อมูลเข้าไปในตัวอาร์ซีเอ็กซ์นั้นได้ผ่านตัวรับ/ ส่ง อินฟราเรดหรือที่เรียกว่า “อินฟราเรดทาวเวอร์ (Infrared Tower)” ซึ่งมีมาพร้อมกับทดลองชุดเลโก้ มายสตอร์มโดยซอฟท์แวร์ที่ช่วยในการอัพโหลดคือ ซิกวิน(Cygwin) อีกเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การพัฒนาหุ่นยนต์ตรวจจับเส้นทาง

ในบทนี้จะอธิบายถึงการประยุกต์กระบวนการพัฒนาระบบเพื่อใช้ในการพัฒนาหุ่นยนต์ตรวจจับเส้นทาง โดยดำเนินการตามขั้นตอนต่างๆ ที่ได้ศึกษาและอธิบายในส่วนของการพัฒนาระบบ โดยเริ่มจากขั้นตอนการศึกษาปัญหา หรือความต้องการของระบบ แล้วทำการวิเคราะห์และออกแบบ กระบวนการทำงานของระบบกับฮาร์ดแวร์ที่ใช้ พร้อมทั้งจัดสร้างโปรแกรมสำหรับระบบงานจริง สำหรับระบบหลักที่จะต้องพัฒนาใช้ในหุ่นยนต์ คือ ระบบตรวจจับเส้นทาง โดยมีรายละเอียดการพัฒนาดังนี้

3.1 ความต้องการของระบบ

การแบ่งความต้องการของระบบนั้น จะแบ่งหลักๆได้ 2 ประเภท คือ หน้าที่ของระบบ (Functional) ที่ระบบจำเป็นจะต้องทำ และส่วนประกอบของระบบ (Non – Functional Supplementary) ที่ระบบจำเป็นต้องมีได้ดังนี้

3.1.1 หน้าที่ของระบบ

ระบบตรวจจับเส้นทางนั้น มีการทำงานบนฮาร์ดแวร์ที่เป็นลักษณะของหุ่นยนต์ ดังนั้นหน้าที่การทำงานส่วนใหญ่แล้วจะเป็นการทำงานที่ระบบจะตอบสนองกับตัวฮาร์ดแวร์ ดังนี้

1. หุ่นยนต์ต้องสามารถเดินจากจุดเริ่มต้นไปจนถึงจุดสิ้นสุดที่กำหนดได้
2. ระบบจะสามารถเลือกเส้นทางผ่านสิ่งกีดขวางระหว่างการเดินทางไปยังจุดสิ้นสุดได้
3. หุ่นยนต์ต้องไม่เดินออกนอกขอบเขตสนามทดสอบ
4. หุ่นยนต์ต้องเดินผ่านทางต่างระดับได้
5. เมื่อขึ้นทางต่างระดับระบบควรจะปรับมอเตอร์ให้เหมาะสมไม่ให้หุ่นยนต์หงายหลังได้
6. เมื่อลงทางต่างระดับระบบควรจะปรับมอเตอร์ให้เหมาะสมไม่ให้หุ่นยนต์ลื่นไถลได้

3.1.2 ส่วนประกอบของระบบ

ระบบตรวจจับเส้นทางนั้น มีการทำงานบนฮาร์ดแวร์ที่เป็นลักษณะของหุ่นยนต์ ดังนั้นส่วนประกอบของระบบ จึงจะเป็นในเรื่องของเงื่อนไขที่สามารถตรวจสอบได้ดังนี้

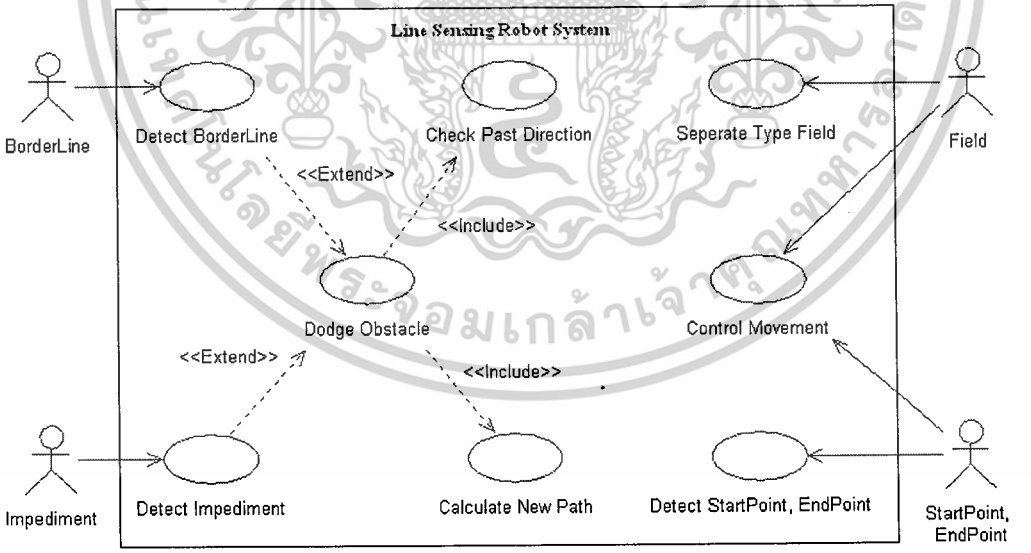
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ต้องมีรูปแบบจุดสิ้นสุดที่ตรวจจับได้แน่นอน
2. สิ่งกีดขวางต้องมีความสูงมากกว่าระดับการตรวจจับของเซ็นเซอร์สัมผัส
3. สิ่งกีดขวางต้องมีน้ำหนักพอที่จะทำให้หุ่นยนต์เคลื่อนที่ผ่านไปไม่ได้
4. พื้นผิวลาดชันต้องมีมุมองศาที่ไม่มากกว่ากำลังที่มอเตอร์รับได้ และมีขอบกั้นเพื่อกันหุ่นยนต์หล่นจากทางลาดชัน
5. ต้องสามารถปรับเปลี่ยนสิ่งกีดขวาง อุปสรรค จุดเริ่มต้น และจุดสิ้นสุดสำหรับการทดสอบได้
6. ต้องมีรูปแบบเซ็นเซอร์สัมผัสที่รองรับการตรวจสอบสิ่งกีดขวาง
7. ต้องมีขอบเขตของสนามทดสอบชัดเจน
8. ต้องมีการแบ่งส่วนของพื้นที่สนามชัดเจน
9. สีของพื้นผิวทางต่างระดับทั้งทางขึ้นและทางลง รวมไปถึงทางธรรมดาต้องต่างกัน

3.2 การออกแบบระบบงาน

3.2.1 แผนภาพยูสเคส

เป็นแผนภาพที่แสดงถึงขอบเขตของระบบ ที่จะแสดงให้เห็นถึงหน้าที่ของระบบที่กระทำ แสดงดังรูปที่ 3.1 และแสดงรายละเอียดของยูสเคส ในตารางที่ 3.1 – 3.8



รูปที่ 3.1 แผนภาพยูสเคสระบบตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 Use Case Description – Separate Type of Field

Use Case Name: Separate Type of Field	**ID:** 01	**Importance Level:** Medium
Primary Actor: Field	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ตรวจสอบค่าแสงที่ได้รับ และประมวลผลประเภทพื้นสนามทดสอบ		
Trigger: เมื่อเซ็นเซอร์แสงเริ่มทำงาน **Type:**		
Relationships: **Association:** Field **Include:** **Extend:** **Generalization**		
Normal Flow of Events: 1. รับค่าแสงจากพื้นผิว 2. ตรวจสอบประเภทของค่าแสงที่ได้รับ 3. ส่งข้อมูลประเภทพื้นผิวไปที่ตัวควบคุม		
Sub Flows: S-1: 1. เซ็นเซอร์แสงตรวจจับค่าแสงจากพื้นผิว 2. แปลงค่าแสงจากฐาน 16 เป็นฐาน 10 S-2 1. เปรียบเทียบค่าแสงกับค่าของพื้นผิวแต่ละประเภทที่กำหนดไว้ 2. เก็บค่าประเภทพื้นผิวที่ตรวจสอบ		
Alternative /Exception Flows:		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 Use Case Description – Detect EndPoint

Use Case Name: Detect EndPoint	**ID:** 02	**Importance Level:** High
Primary Actor: EndPoint	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ตรวจสอบค่าแสงที่ได้รับ และประมวลผลจุดสิ้นสุด		
Trigger: เมื่อเซ็นเซอร์แสงเริ่มทำงาน **Type:**		
Relationships: **Association:** EndPoint **Include:** **Extend:** **Generalization**		
Normal Flow of Events: 1. รับค่าแสงจากพื้นผิว 2. ตรวจสอบจุดสิ้นสุดที่ได้รับ 3. ส่งข้อมูลจุดสิ้นสุดไปที่ตัวควบคุม		
Sub Flows: S-1: 1. เซ็นเซอร์แสงตรวจจับค่าแสงจากพื้นผิว 2. แปลงค่าแสงจากฐาน 16 เป็นฐาน 10 S-2 1. เปรียบเทียบค่าแสงกับค่าของจุดสิ้นสุดที่กำหนดไว้ 2. เก็บค่าจุดสิ้นสุด		
Alternative /Exception Flows:		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Use Case Description – Control Movement

Use Case Name: Control Movement	**ID:** 03	**Importance Level:** Low
Primary Actor: Field, EndPoint	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ควบคุมรูปแบบการทำงานของมอเตอร์ทั้ง 2 ตัว		
Trigger: เมื่อมีการส่งค่าข้อมูลประเภทการเคลื่อนที่เข้ามา **Type:**		
Relationships: **Association:** Field, EndPoint **Include:** **Extend:** **Generalization**		
Normal Flow of Events: 1. รับค่าแสงจากพื้นผิว 2. เลือกประเภทการตอบสนอง 3. ควบคุมการทำงานของมอเตอร์		
Sub Flows: S-1: 1. เซ็นเซอร์แสงตรวจจับค่าแสงจากพื้นผิว 2. แปลงค่าแสงจากฐาน 16 เป็นฐาน 10 S-2: 1. เปรียบเทียบค่าแสงกับประเภทการเคลื่อนที่ตามช่วงสีที่กำหนดไว้ 2. ส่งข้อมูลประเภทการตอบสนอง S-3: 1. กำหนดทิศทางการหมุนของมอเตอร์ 2. กำหนดความเร็วในการหมุนของมอเตอร์		
Alternative /Exception Flows: 1.a กรณีเจอจุดสิ้นสุดจะหยุดการทำงานของมอเตอร์เลย		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 Use Case Description – Detect BorderLine

Use Case Name: Detect BorderLine	**ID:** 04	**Importance Level:** Medium
Primary Actor: BorderLine	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ตรวจสอบค่าแสงที่ได้รับ และประมวลผลประเภทเส้นขอบเขตสนาม		
Trigger: เมื่อเซ็นเซอร์แสงเริ่มทำงาน **Type:**		
Relationships: **Association:** BorderLine **Include:** - **Extend:** Dodge Obstacle **Generalization:** -		
Normal Flow of Events: 1. รับค่าแสงจากพื้นผิว 2. ตรวจสอบประเภทของเส้นขอบเขตที่ได้รับ 3. ส่งข้อมูลเส้นขอบเขตไปที่ตัวควบคุม		
Sub Flows: S-1: 1. เซ็นเซอร์แสงตรวจจับค่าแสงจากพื้นผิว 2. แปลงค่าแสงจากฐาน 16 เป็นฐาน 10 S-2 1. เปรียบเทียบค่าแสงกับค่าของเส้นขอบเขตที่กำหนดไว้		
Alternative /Exception Flows: 1.a ในกรณีที่เจอเส้นขอบเขตจะส่งการทำงานไปยัง Dodge Obstacle		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 Use Case Description – Detect Impediment

Use Case Name: Detect Impediment	**ID:** 05	**Importance Level:** Medium
Primary Actor: Impediment	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ตรวจสอบสิ่งกีดขวาง		
Trigger: เมื่อเซ็นเซอร์สัมผัสเริ่มทำงาน **Type:**		
Relationships: **Association:** Impediment **Include:** - **Extend:** Dodge Obstacle **Generalization:** -		
Normal Flow of Events: 1. รับค่าการชนจากเซ็นเซอร์สัมผัส 2. ตรวจสอบประเภทของสิ่งกีดขวางที่ได้รับ 3. ส่งข้อมูลสิ่งกีดขวางไปที่ตัวควบคุม		
Sub Flows: S-1: 1. เซ็นเซอร์แสงตรวจจับค่าการชนจากเซ็นเซอร์สัมผัส 2. แปลงค่าการชนจากฐาน 16 เป็นฐาน 10 S-2 1. เปรียบเทียบค่าการชนกับค่าของสิ่งกีดขวางที่กำหนดไว้		
Alternative /Exception Flows: 1.a ในกรณีที่เจอสิ่งกีดขวางจะส่งการทำงานไปยัง Dodge Obstacle		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 Use Case Description -- Dodge Obstacle

Use Case Name: Dodge Obstacle	**ID:** 06	**Importance Level:** High
Primary Actor: -	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: เลือกวิธีการหลบหลีกอุปสรรค		
Trigger: เมื่อมีการส่งค่าของอุปสรรคเข้ามา **Type:**		
Relationships: **Association:** Detect BorderLine, Detect Impediment **Include:** Check Past Obstacle, Calculate New Path **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. รับค่าเป้าหมายอุปสรรค และอุปสรรคที่ผ่านมา 2. ตรวจสอบอุปสรรคล่าสุดจากอุปสรรคที่ผ่านมา 3. ส่งค่าอุปสรรคล่าสุดไปคำนวณหาเส้นทางใหม่ 4. เลือกประเภทการเคลื่อนที่ตอบสนองทิศทางใหม่ที่ได้รับ		
Sub Flows: S-2: 1. ส่งการทำงานไปตรวจสอบรูปแบบอุปสรรคล่าสุดจาก Check Past Obstacle 2. รับค่ารูปแบบอุปสรรคล่าสุดจาก Check Past Obstacle S-3 1. ส่งการทำงานไปหาเส้นทางใหม่จาก Calculate New Path 2. รับค่าเส้นทางใหม่จาก Calculate New Path S-4 1. เปรียบเทียบทิศทางใหม่กับประเภทการเคลื่อนที่ 2. ส่งค่าข้อมูลประเภทการเคลื่อนที่		
Alternative /Exception Flows:		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Use Case Description – Check Past Obstacle

Use Case Name: Check Past Obstacle	**ID:** 07	**Importance Level:** High
Primary Actor: -	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: ตรวจสอบทิศทางปัจจุบันจากทิศทางที่ผ่านมา		
Trigger: เมื่อมีการเรียกการทำงานจาก Dodge Obstacle **Type:**		
Relationships: **Association: -** **Include:** Dodge Obstacle **Extend: -** **Generalization: -**		
Normal Flow of Events: 1. คำนวณอุปสรรคล่าสุด		
Sub Flows: S-1: 1. รับค่าประเภทการเคลื่อนที่ที่ผ่านมา กับอุปสรรคที่ผ่านมา 2. ตรวจสอบทิศทางจากการเคลื่อนที่และอุปสรรคที่ผ่านมา กับอุปสรรคเป้าหมาย 3. คำนวณทิศทางสุดท้ายที่เกิดขึ้น 4. ส่งค่าทิศทางสุดท้ายไปยัง Dodge Obstacle		
Alternative /Exception Flows:		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 Use Case Description – Calculate New Path

Use Case Name: Calculate New Path	**ID:** 08	**Importance Level:** High
Primary Actor: -	**Use Case Type:**	
Stakeholders and Interests: -		
Brief Description: คำนวณหาทิศทางใหม่		
Trigger: เมื่อมีการเรียกการทำงานจาก Dodge Obstacle **Type:**		
Relationships: **Association: -** **Include:** Dodge Obstacle **Extend: -** **Generalization: -**		
Normal Flow of Events: 1. คำนวณหาเส้นทางใหม่		
Sub Flows: S-1: 1. รับค่าทิศทางสุดท้าย 2. ตรวจสอบทิศทางจุดสิ้นสุด 3. เปรียบเทียบทิศทางสุดท้ายกับทิศทางจุดสิ้นสุดคำนวณหาทิศทางที่เป็นไปได้ 4. ส่งค่าทิศทางใหม่ไปยัง Dodge Obstacle		
Alternative /Exception Flows:		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

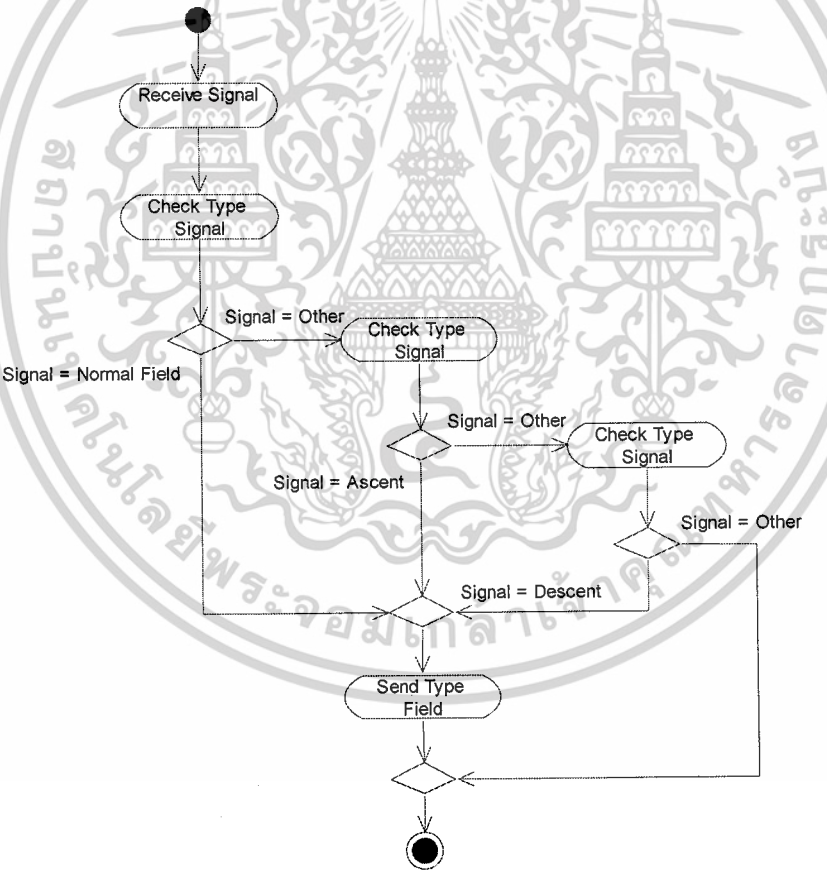
3.2.2 แผนภาพกิจกรรม (Activity Diagram)

เป็นแผนภาพที่แสดงถึงกิจกรรมของระบบ ที่จะแสดงให้เห็นถึงกิจกรรมภายในการทำงานของระบบแต่ละอย่าง แสดงรายละเอียดของแผนภาพกิจกรรม ดังรูปที่ 3.2 - 3.9

กิจกรรม Separate Type Field เริ่มต้นด้วยการรับค่าสัญญาณที่ได้จากการตรวจจับแสงของเซนเซอร์แสง หลังจากนั้นจะเป็นการตรวจสอบประเภทของสัญญาณที่ได้รับมา ถ้าค่าสัญญาณอยู่ในประเภทพื้นผิวที่กำหนดไว้ จะทำการส่งประเภทของพื้นผิวให้ระบบจดจำดังรูปที่ 3.2

<u>ตัวอย่างโปรแกรม</u>

```
ls2      = (unsigned long int)l2.get();     // รับค่าสัญญาณ
if(ls2==38 || ls2==39 || ls2==40) {         // เช็คค่าพื้นผิวธรรมดา
        x=5;    // ส่งประเภทของพื้นผิวให้ระบบจดจำ
}
```



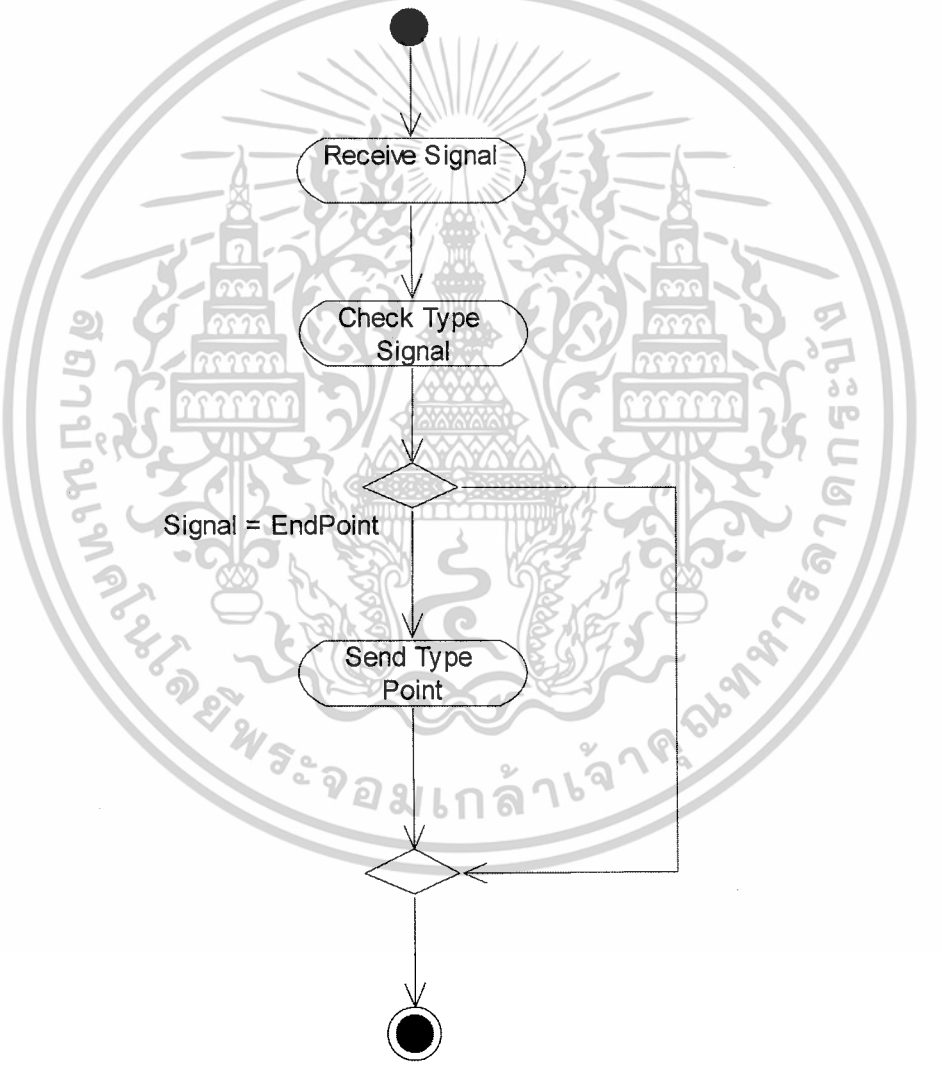
รูปที่ 3.2 แผนภาพกิจกรรม Separate Type Field

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Detect EndPoint เริ่มต้นด้วยการรับค่าสัญญาณที่ได้จากการตรวจจับแสงของเซนเซอร์แสง หลังจากนั้นจะเป็นการตรวจสอบประเภทของสัญญาณที่ได้รับมา ถ้าค่าสัญญาณอยู่ในประเภทจุดสิ้นสุดที่กำหนดไว้ จะทำการส่งค่าจุดสิ้นสุดให้ระบบจดจำ ดังรูปที่ 3.3

<u>ตัวอย่างโปรแกรม</u>

```
ls2        = (unsigned long int)l2.get();      // รับค่าสัญญาณ
if(ls2==48 || ls2==49) {                  // เช็คค่าพื้นผิวจุดสิ้นสุด
        x=8;     // ส่งประเภทของจุดสิ้นสุดให้ระบบจดจำ
}
```



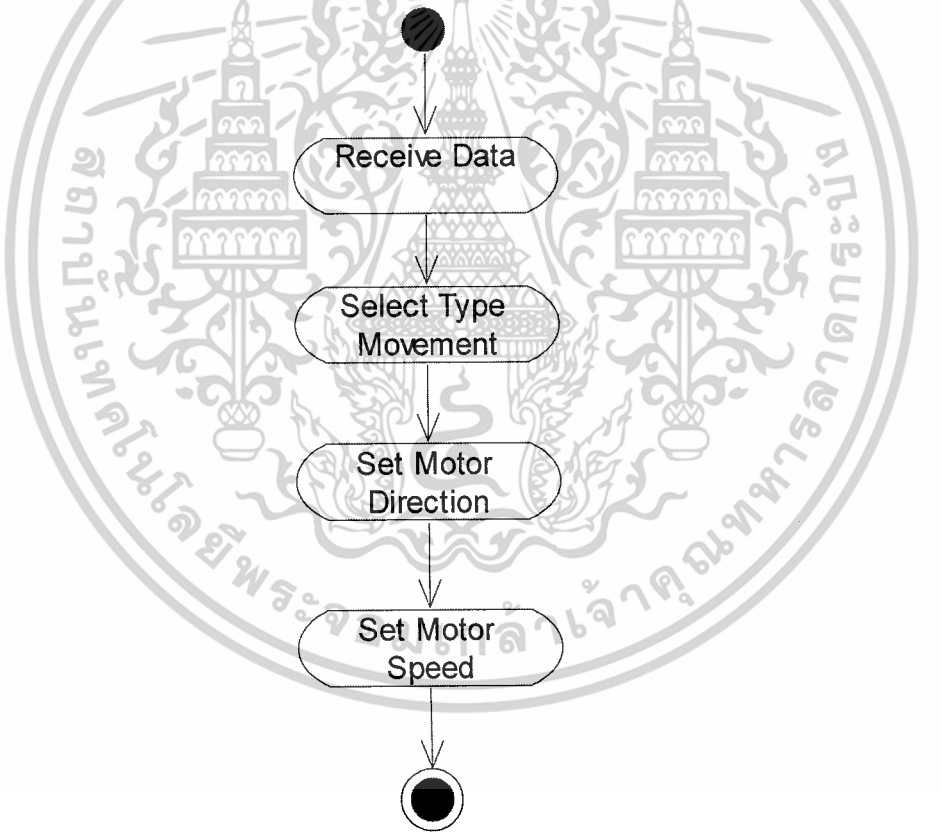
รูปที่ 3.3 แผนภาพกิจกรรม Detect EndPoint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Control Movement เริ่มต้นด้วยการรับค่าข้อมูลที่ได้จากการตรวจจับแสงของเซนเซอร์แสง หลังจากนั้นจะเป็นการเลือกประเภทการเคลื่อนที่ และจะทำการตั้งค่าทิศทาง และความแรงของมอเตอร์ ดังรูปที่ 3.4

ตัวอย่างโปรแกรม

```
void chooseCaseLightSensor(int j,int timeSleepL){        // รับค่าข้อมูล
        switch(j){                                       // เลือกประเภทการเคลื่อนที่
                case 5:
                        p->moveForward(p1,p2);           // ปรับทิศทาง และความเร็ว
        }
}
```



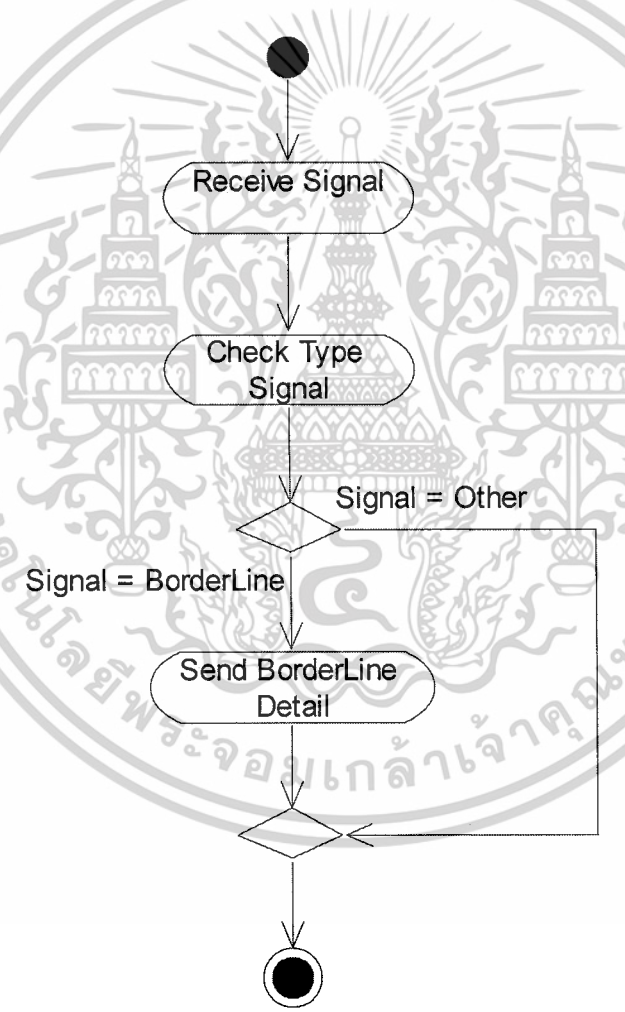
รูปที่ 3.4 แผนภาพกิจกรรม Control Movement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Detect BorderLine เริ่มต้นด้วยการรับค่าสัญญาณที่ได้จากการตรวจจับแสงของเซนเซอร์แสง หลังจากนั้นจะเป็นการตรวจสอบประเภทของสัญญาณที่ได้รับมา ถ้าค่าสัญญาณอยู่ในประเภทเส้นขอบเขตที่กำหนดไว้ จะทำการส่งค่าเส้นขอบเขตให้ระบบจดจำดังรูปที่ 3.5

ตัวอย่างโปรแกรม

```
ls2        = (unsigned long int)l2.get();      // รับค่าสัญญาณ
if (ls2==31 || ls2==32 || ls2==33) {           // เช็คค่าเส้นขอบเขต
        x=4;     // ส่งประเภทของเส้นขอบเขตให้ระบบจดจำ
}
```



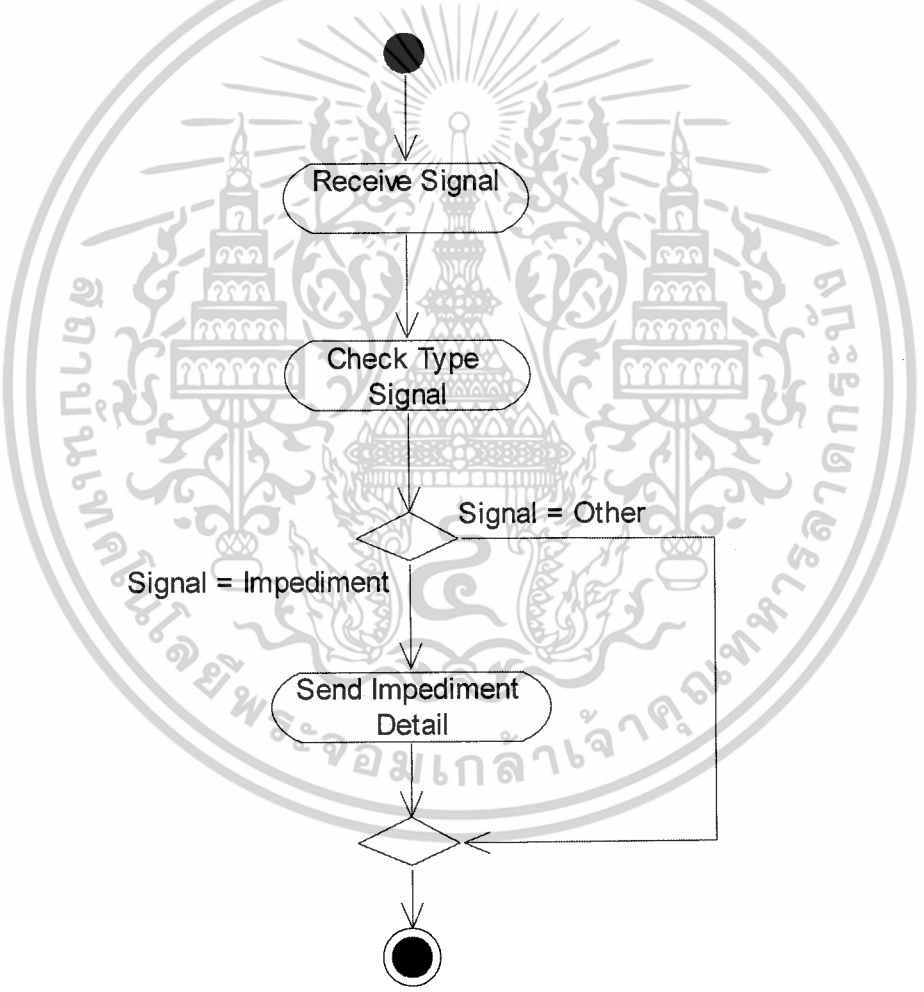
รูปที่ 3.5 แผนภาพกิจกรรม Detect BorderLine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Detect Impediment เริ่มต้นด้วยการรับค่าสัญญาณที่ได้จากการตรวจจับการชนของเซนเซอร์สัมผัส หลังจากนั้นจะเป็นการตรวจสอบประเภทของสัญญาณที่ได้รับมา ถ้าค่าสัญญาณอยู่ ในประเภทการชนที่กำหนดไว้ จะทำการส่งประเภทของการชนให้ระบบจดจำดังรูปที่ 3.6

<u>ตัวอย่างโปรแกรม</u>

```
ts1 = (unsigned long int)t1.get();          // รับค่าสัญญาณ
if(ts1<=61446 && ts3>=61446){               // เช็คค่าการสัมผัสด้านซ้าย
        w=1;    // ส่งประเภทการชนให้ระบบจดจำ
}
```



รูปที่ 3.6 แผนภาพกิจกรรม Detect Impediment

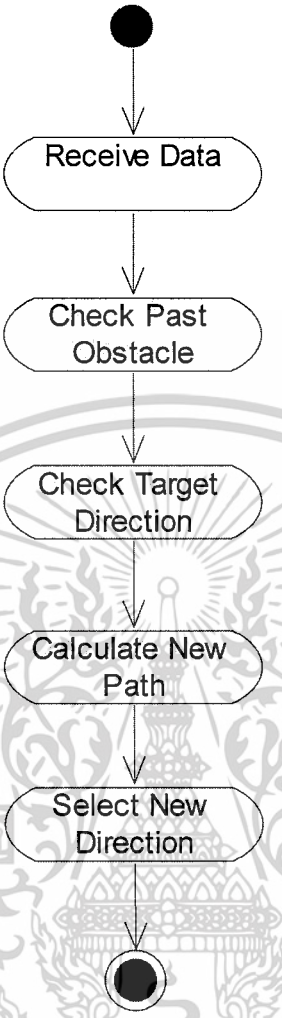
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Dodge Obstacle เริ่มต้นด้วยการรับค่าข้อมูลที่ได้จากการตรวจจับว่าต้องมีการหลบหลีก หลังจากนั้นจะดึงค่าของอุปสรรคเก่า เป็นการตรวจสอบอุปสรรคปัจจุบัน และจะทำการตรวจสอบทิศทางปัจจุบันที่หุ่นยนต์อยู่ และจะทำการคำนวนหาเส้นทางใหม่ พร้อมทั้งเลือกทิศทางใหม่จากเส้นทางที่หาได้ดังรูปที่ 3.7

ตัวอย่างโปรแกรม

```
if(x!=6 && x!=7){        // เช็คอุปสรรคเก่า
        if(z==0)         // เช็คข้อมูล
                z=1;     // อุปสรรคใหม่
        else if(z==1)
                z=0;
}
if(z==0){        // เช็คทิศทางเป้าหมาย
        if(x==6){                // คำนวนหาทิศทางใหม่
                count = 50;
         }else{
                x=6;             // เลือกทิศทางใหม่
                count = 0;
         }
}else if(z==1){ //Check Downer = 7
         if(x==7){
                count = 50;
         }else{
                x=7;
                count = 0;
         }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แผนภาพกิจกรรม Dodge Obstacle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Check Past Obstacle เริ่มต้นด้วยการรับค่าข้อมูลจากการส่งมาเช็คอุปสรรคเก่าเกิดขึ้น หลังจากนั้นจะทำการเปรียบเทียบกับเป้าหมายอุปสรรคที่กำหนดไว้ ถ้าใช่ก็จะส่งข้อมูลอุปสรรคนั้นๆไป ดังรูปที่ 3.8

ตัวอย่างโปรแกรม

```
if(x==4){                    // เปรียบเทียบเป้าหมายอุปสรรค
        count = 50;
}else{
        x=4;                 // ส่งค่าอุปสรรคปัจจุบัน
        count = ;
}
```

รูปที่ 3.8 แผนภาพกิจกรรม Check Past Obstacle

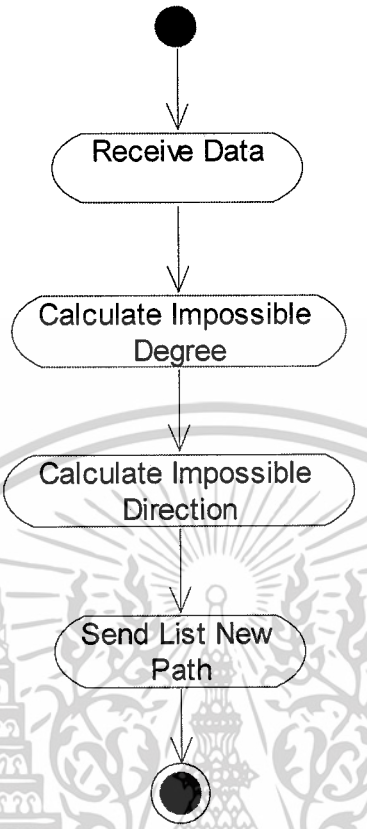
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรม Calculate New Path เริ่มต้นด้วยการรับค่าข้อมูลจากการส่งมาคำนวนหาทิศทางใหม่ โดยจะทำการคำนวนหาองศา และทิศทางที่เป็นไปและจะส่งข้อมูลอุปสรรคนั้นๆไป ดังรูปที่ 3.9

<u>ตัวอย่างโปรแกรม</u>

```
if(ts1<=61446 && ts3>=61446){          // โดนชนด้านซ้ายของหุ่นยนต์
        w=1;            // เลือกทิศทางที่เป็นไปได้
        r->chooseCaseTouchSensor(w,count);
}else if(ts1>=61446 && ts3<=61446){    // โดนชนด้านขวาของหุ่นยนต์
        w=2;
       r->chooseCaseTouchSensor(w,count);
}
void chooseCaseTouchSensor(int sideT,int timeSleepT){
        switch(sideT){
                case 1: // Left
                        q->brakeWheel(q2);
                        q->moveBack(q1,q2);
                        q->turnRight(q1,q2);
                case 2: // Right
                        q->brakeWheel(q2);
                        q->moveBack(q1,q2);
                        q->turnLeft(q1,q2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

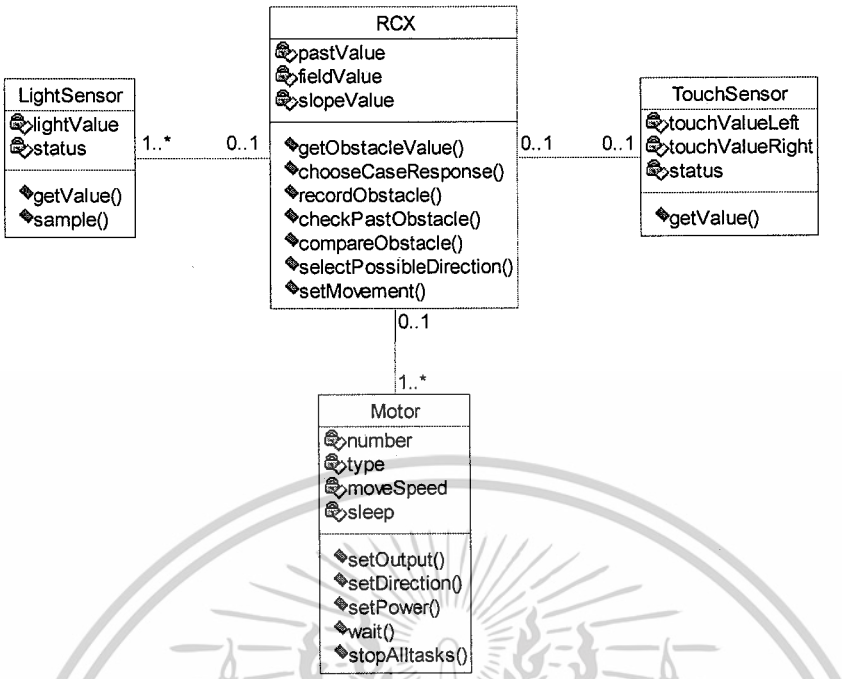


รูปที่ 3.9 แผนภาพกิจกรรม Calculate New Path

3.2.3 แผนภาพคลาส (Class Diagram)

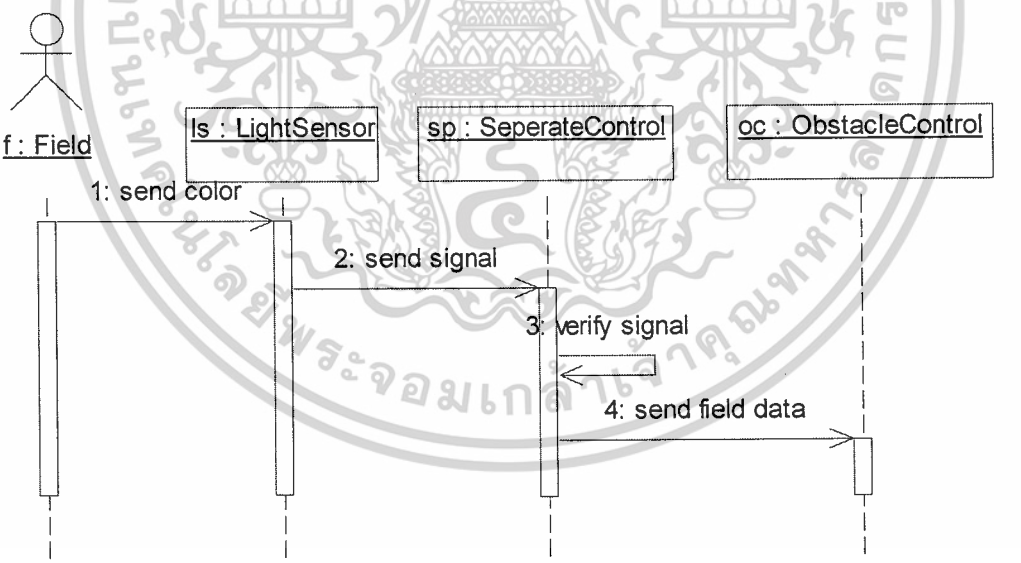
เป็นแผนภาพที่แสดงให้เห็นโครงสร้างทางด้านคุณลักษณะ และโครงสร้างทางด้านการประมวลผล ซึ่งจะได้รูปแบบทั้งหมด 4 คลาส โดยคลาส LightSensor และ TouchSensor เป็น Boundary Class ที่แทนรูปแบบของการตรวจจับจากอุปสรรคต่างๆ โดยการตรวจจับแต่ละอย่างนั้นมีรูปแบบที่คล้ายคลึงกันจึงถูกแทนด้วยเซนเซอร์แสงที่เป็นส่วนประกอบของหุ่นยนต์ที่มีหน้าที่ตรวจจับ ส่วนคลาส Motor เป็นคลาสที่แทนรูปแบบการเคลื่อนที่ต่างๆ ที่ถูกแทนด้วยมอเตอร์ที่เป็นอุปกรณ์ขับเคลื่อนของหุ่นยนต์ และสุดท้ายคลาส RCX เป็นคลาสที่มีหน้าที่ควบคุมการทำงานต่างๆ โดยตัวประมวลผลจะถูกเก็บอยู่ภายในส่วน RCX ของหุ่นยนต์ หน้าที่ต่างๆภายในคลาสจะเป็นฟังก์ชั่นการทำงานต่างๆ ที่อยู่ภายในโปรแกรม ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



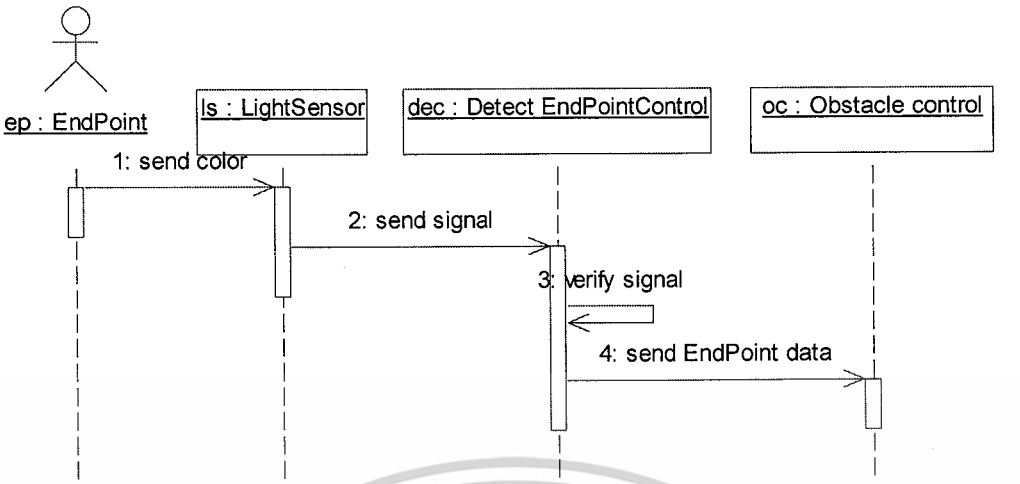
รูปที่ 3.10 แผนภาพ Class Diagram ของระบบหุ่นยนต์ตรวจจับเส้นทาง

3.2.4 แผนภาพลำดับเหตุการณ์ (Sequence Diagram)

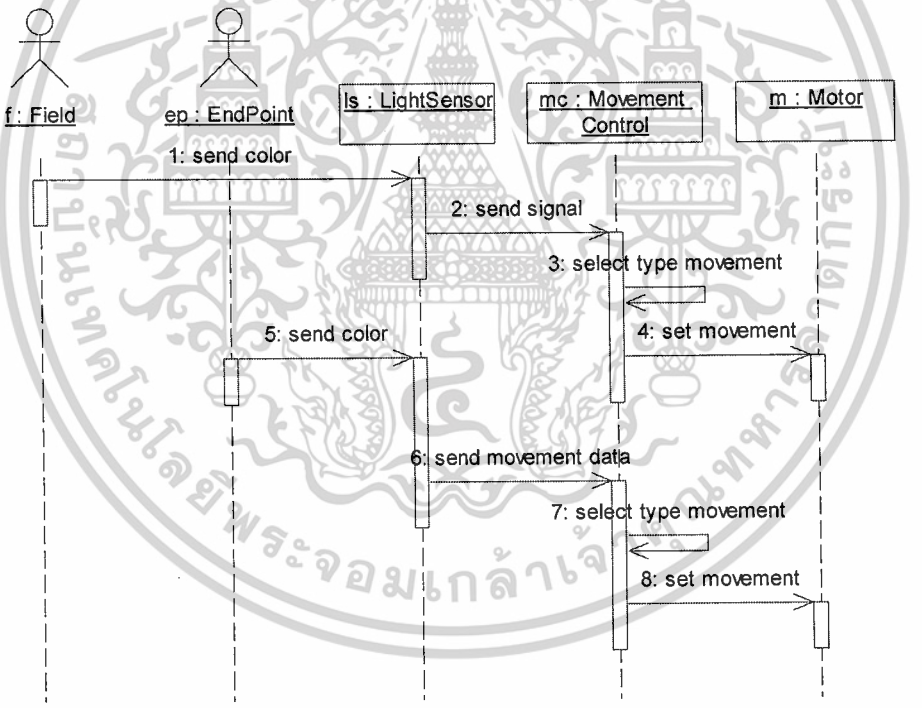


รูปที่ 3.11 Sequence Diagram ของ usecase Separate Type Field ของระบบหุ่นยนต์ตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

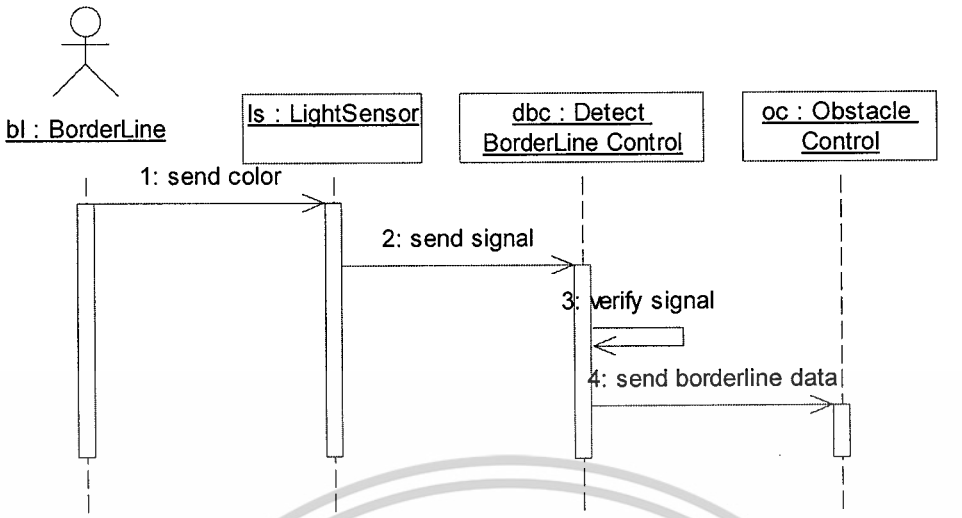


รูปที่ 3.12 Sequence Diagram ของ Usecase Detect EndPoint ของระบบหุ่นยนต์ตรวจจับเส้นทาง

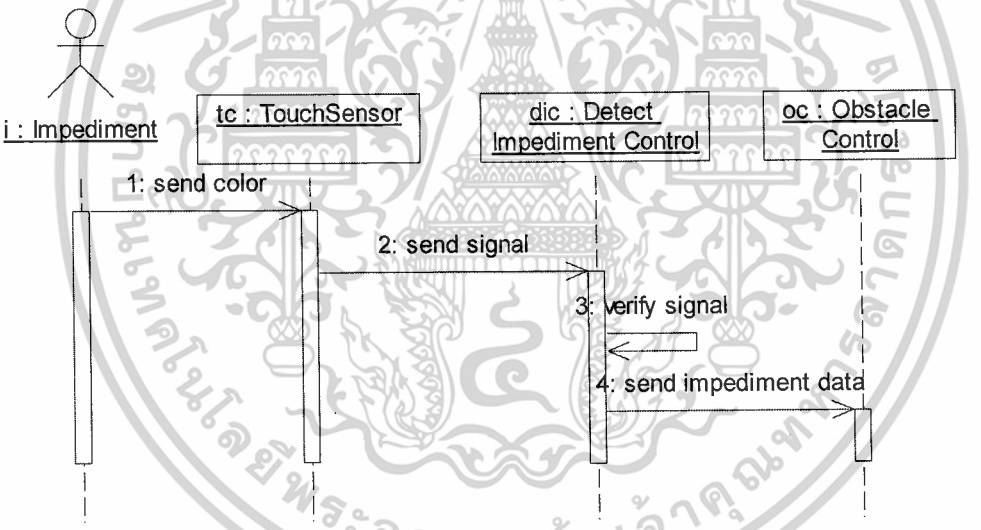


รูปที่ 3.13 Sequence Diagram ของ Usecase Control Movement ของระบบหุ่นยนต์ตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

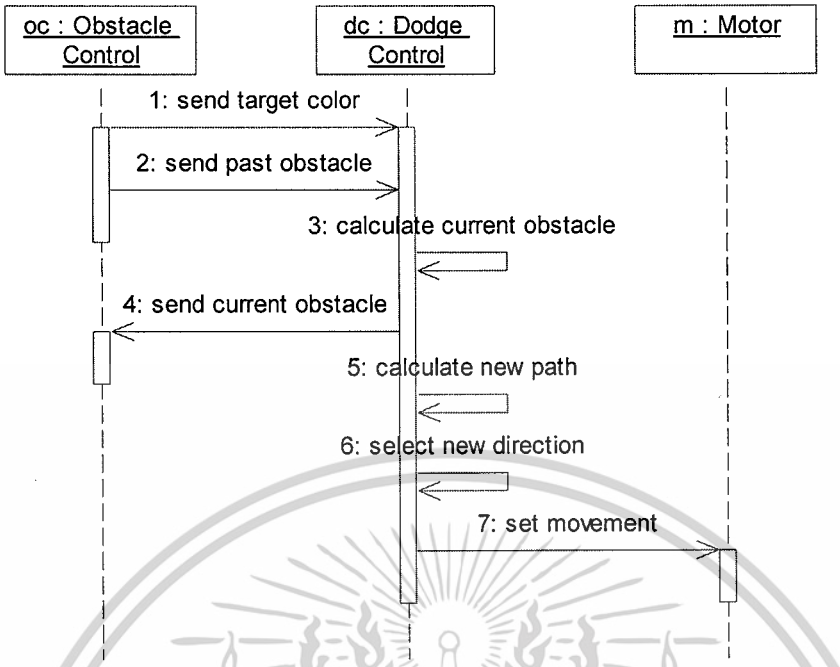


รูปที่ 3.14 Sequence Diagram Usecase Detect Border Line ของระบบหุ่นยนต์ตรวจจับเส้นทาง



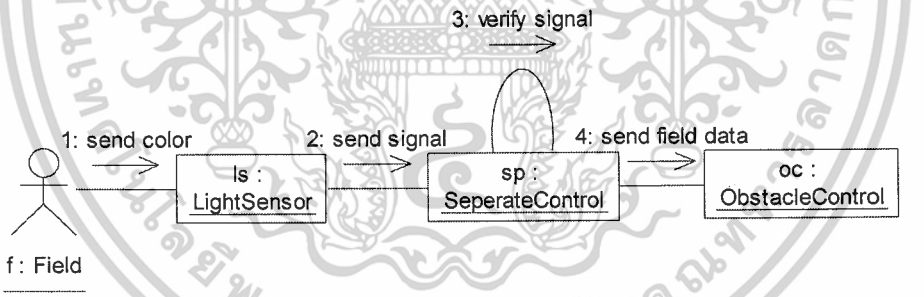
รูปที่ 3.15 Sequence Diagram Usecase Detect Impediment ของระบบหุ่นยนต์ตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



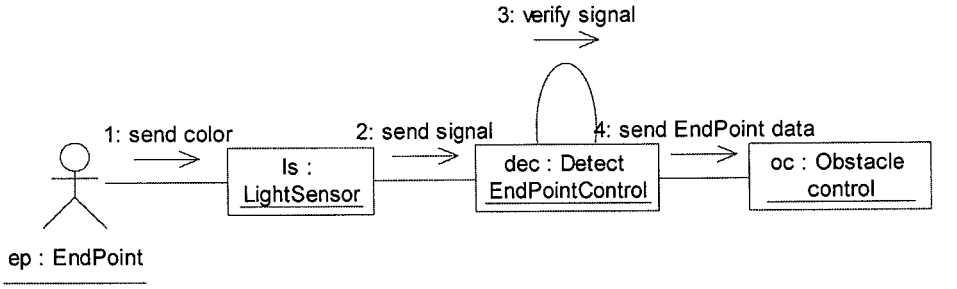
รูปที่ 3.16 Sequence Diagram Usecase Dodge Obstacle ของระบบหุ่นยนต์ตรวจจับเส้นทาง

3.2.5 แผนภาพการปฏิสัมพันธ์ (Collaboration Diagram)

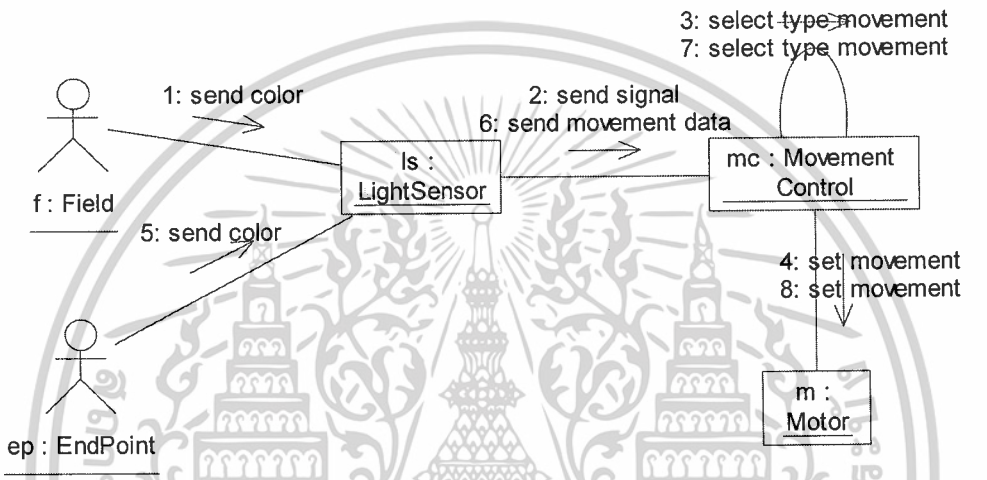


รูปที่ 3.17 Collaboration Diagram Usecase Separate Type Field ของระบบหุ่นยนต์ตรวจจับเส้นทาง

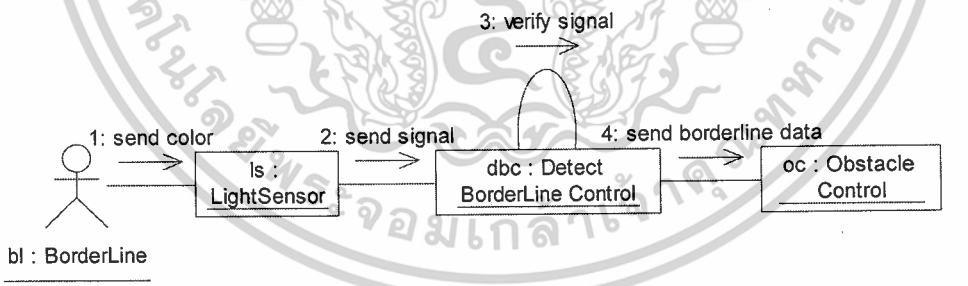
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 Collaboration Diagram Usecase Detect End Point ของระบบหุ่นยนต์ตรวจจับเส้นทาง

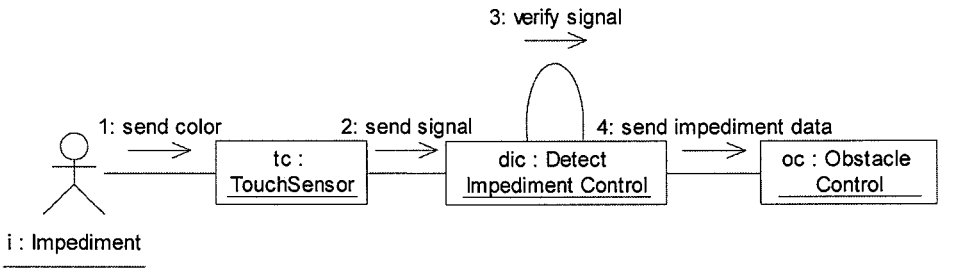


รูปที่ 3.19 Collaboration Diagram Usecase Control Movement ของระบบหุ่นยนต์ตรวจจับเส้นทาง

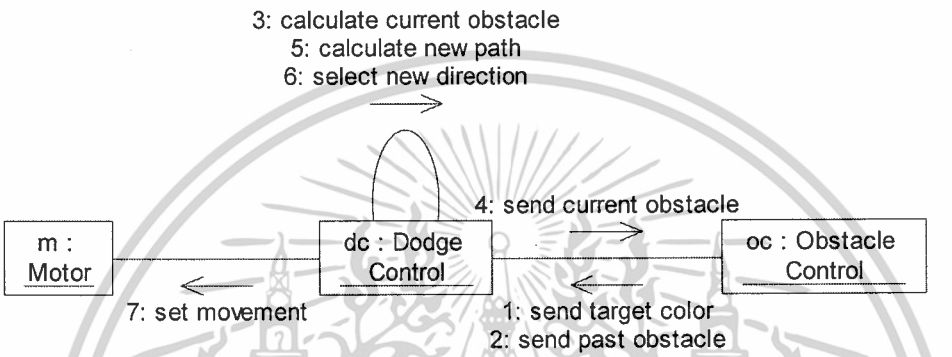


รูปที่ 3.20 Collaboration Diagram Usecase Detect Border Line ของระบบหุ่นยนต์ตรวจจับเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

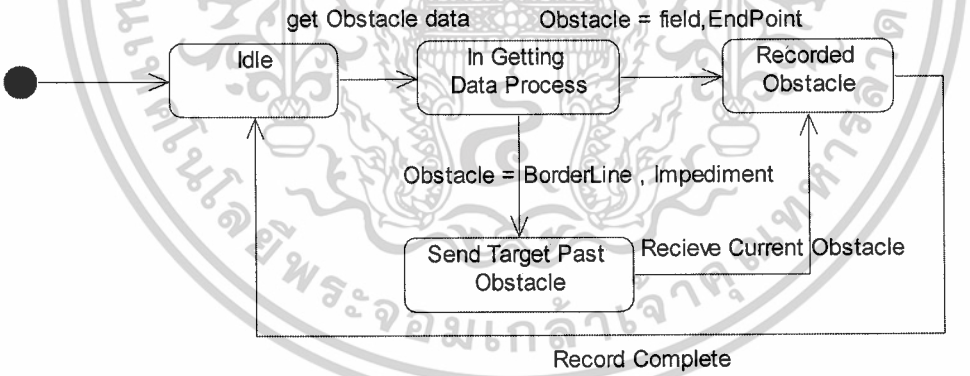


รูปที่ 3.21 Collaboration Diagram Usecase Detect Impediment ของระบบหุ่นยนต์ตรวจจับเส้นทาง



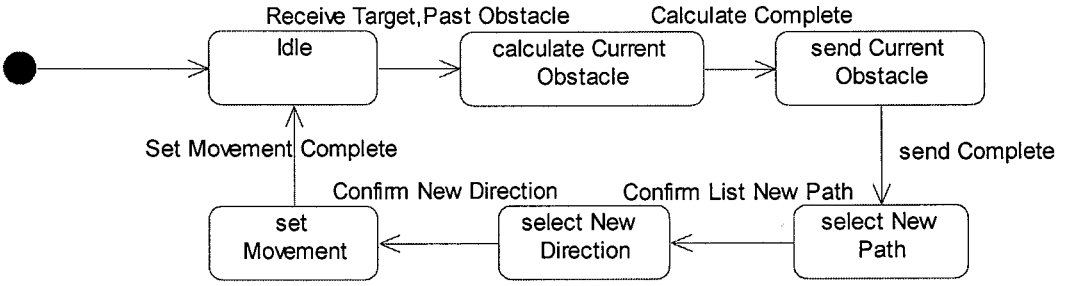
รูปที่ 3.22 Collaboration Diagram Usecase Dodge Obstacle ของระบบหุ่นยนต์ตรวจจับเส้นทาง

3.2.6 แผนภาพสถานะ (Statechart Diagram)



รูปที่ 3.23 Statechart Diagram Obstacle Control ของระบบหุ่นยนต์ตรวจจับเส้นทาง

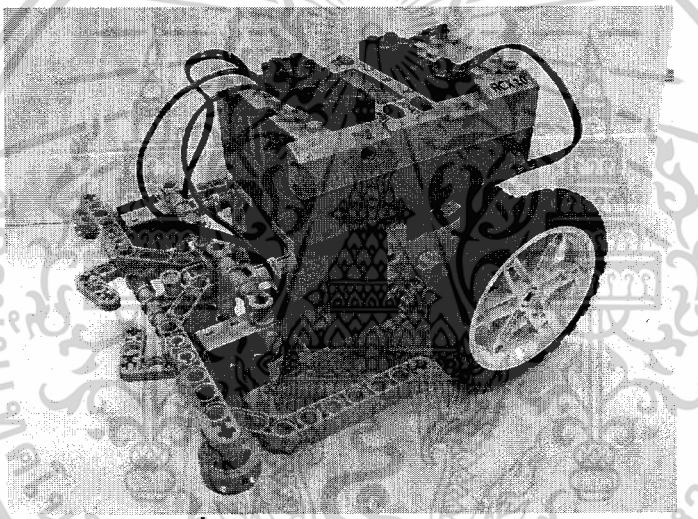
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 Statechart Diagram Dodge Control ของระบบหุ่นยนต์ตรวจจับเส้นทาง

3.3 การออกแบบหุ่นยนต์ตรวจจับเส้นทางและสนามทดลอง

3.3.1 การออกแบบหุ่นยนต์ตรวจจับเส้นทาง

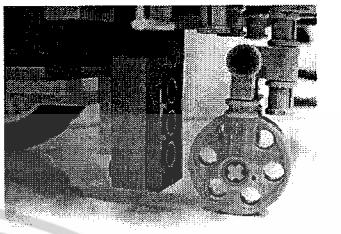
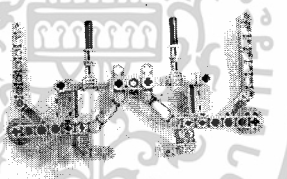
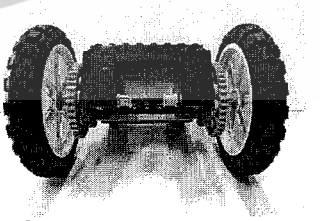


รูปที่ 3.25 หุ่นยนต์ตรวจจับเส้นทาง

จากความต้องการของระบบและหน้าที่ของระบบ ส่งผลทำให้เกิดการออกแบบหุ่นยนต์ตรวจจับเส้นทาง ซึ่งการแปลงความต้องการของระบบ เป็นความต้องการในการออกแบบหุ่นยนต์สามารถแยกออกได้ดังตารางที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

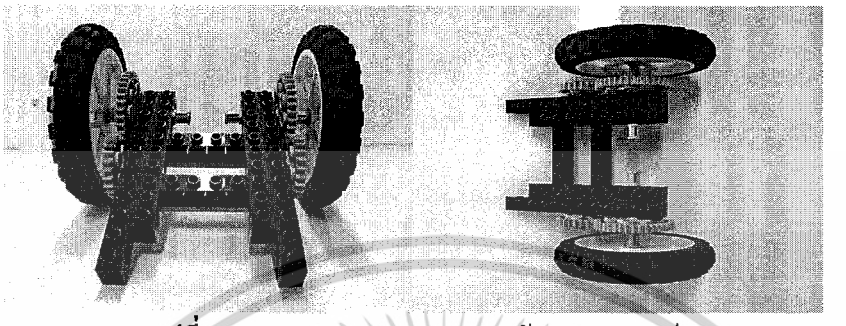
ตารางที่ 3.9 การแปลงความต้องการของระบบเป็นความต้องการในการออกแบบหุ่นยนต์

หน้าที่ของระบบ (หัวข้อ 3.1.1) **ข้อ 3** หุ่นยนต์ต้องไม่เดินออกนอกเขตสนาม **ข้อ 5** เมื่อขึ้นลงต่างระดับระบบควรจะปรับมอเตอร์ให้เหมาะสมไม่ให้หุ่นยนต์หงายหลังได้ **ข้อ 6** เมื่อลงทางต่างระดับระบบควรจะปรับมอเตอร์ให้เหมาะสมไม่ให้หุ่นยนต์ลื่นไถลได้	**การออกแบบหุ่นยนต์ตรวจจับเส้นทาง** ใช้เซนเซอร์แสง 1 ตัว ในการแยกแยะและตรวจจับความแตกต่างของพื้นผิว ,จุดสิ้นสุดและขอบเขตสนามทดสอบ  **รูปที่ 3.26** รูปแบบเซนเซอร์แสง
หน้าที่ของระบบ (หัวข้อ 3.1.1) **ข้อ 2** ระบบต้องเลือกเส้นทางผ่านสิ่งกีดขวางระหว่างเดินทางไปยังจุดสิ้นสุดได้ **ส่วนประกอบของระบบ (หัวข้อ 3.1.2)** **ข้อ 6** ต้องมีรูปแบบเซนเซอร์สัมผัสที่รองรับการตรวจสอบสิ่งกีดขวาง	**การออกแบบหุ่นยนต์ตรวจจับเส้นทาง** - ใช้เซนเซอร์สัมผัส 2 ตัวในการตรวจจับ ทั้งซ้ายและขวา โดยวางด้านหน้าของหุ่นยนต์  **รูปที่ 3.27** รูปแบบกันชน
หน้าที่ของระบบ (หัวข้อ 3.1.1) **ข้อ 4** หุ่นยนต์ต้องเดินผ่านต่างระดับได้	**การออกแบบหุ่นยนต์ตรวจจับเส้นทาง** - ใช้มอเตอร์ 2 ตัวในการควบคุมทิศทางและความเร็วของหุ่นยนต์  **รูปที่ 3.28** รูปแบบชุดขับเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

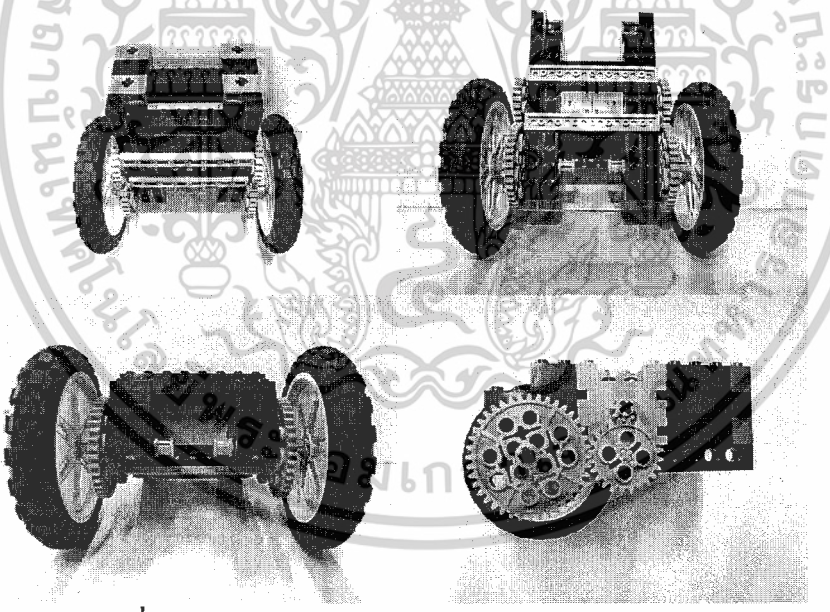
3.3.1.1 การประกอบหุ่นยนต์ตรวจจับเส้นทาง

1. สร้างฐานรองรับล้อและมอเตอร์ ซึ่งการออกแบบฐานนั้นได้รองรับมอเตอร์ 2 ตัวเพื่อใช้ในการบังคับทิศทางอย่างอิสระต่อกันทั้งล้อทางซ้ายและล้อทางขวา ดังรูปที่ 3.29



รูปที่ 3.29 การประกอบฐานรองรับล้อและมอเตอร์

2. ทำการประกอบมอเตอร์เข้ากับฐานล้อ และ ประกอบส่วนที่ยึดโครงสร้างหลักโดยเน้นการประกอบที่แข็งแรงโดยใช้อุปกรณ์น้อยที่สุด ดังรูปที่ 3.30

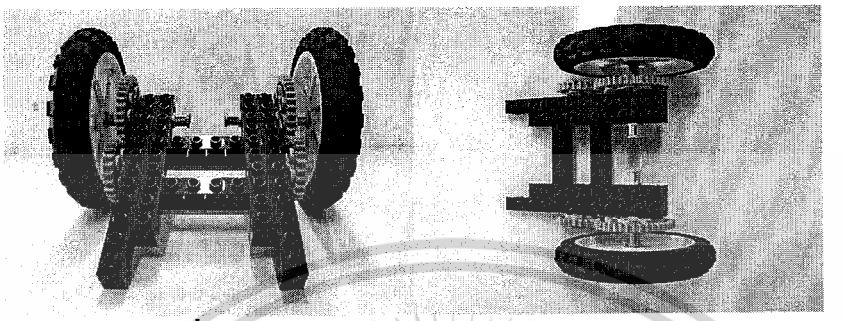


รูปที่ 3.30 การประกอบส่วนโครงสร้างหลัก และ มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

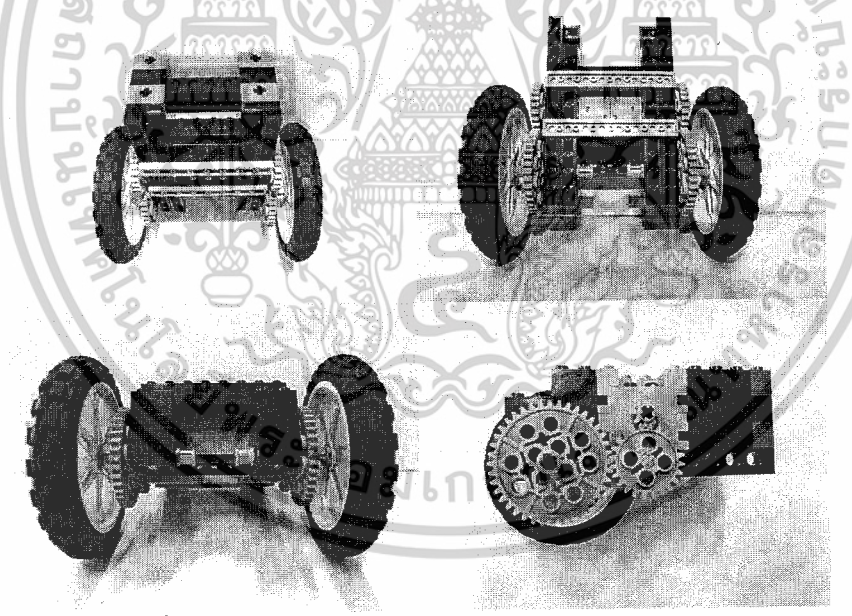
3.3.1.1 การประกอบหุ่นยนต์ตรวจจับเส้นทาง

1. สร้างฐานรองรับล้อและมอเตอร์ ซึ่งการออกแบบฐานนั้นได้รองรับมอเตอร์ 2 ตัวเพื่อใช้ในการบังคับทิศทางอย่างอิสระต่อกันทั้งล้อทางซ้ายและล้อทางขวา ดังรูปที่ 3.29



รูปที่ 3.29 การประกอบฐานรองรับล้อและมอเตอร์

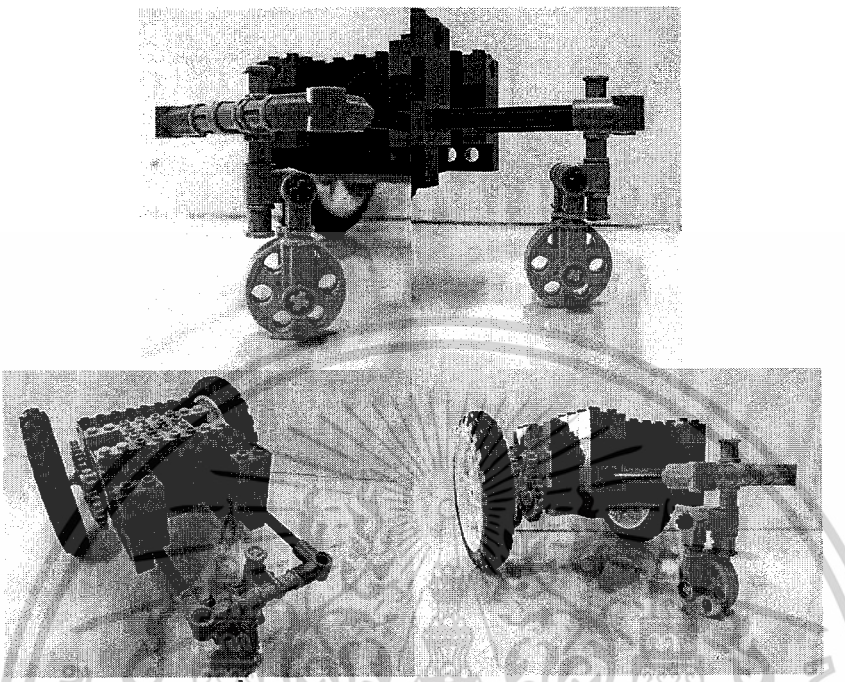
2. ทำการประกอบมอเตอร์เข้ากับฐานล้อ และ ประกอบส่วนที่ยึดโครงสร้างหลักโดยเน้นการประกอบที่แข็งแรงโดยใช้อุปกรณ์น้อยที่สุด ดังรูปที่ 3.30



รูปที่ 3.30 การประกอบส่วนโครงสร้างหลัก และ มอเตอร์

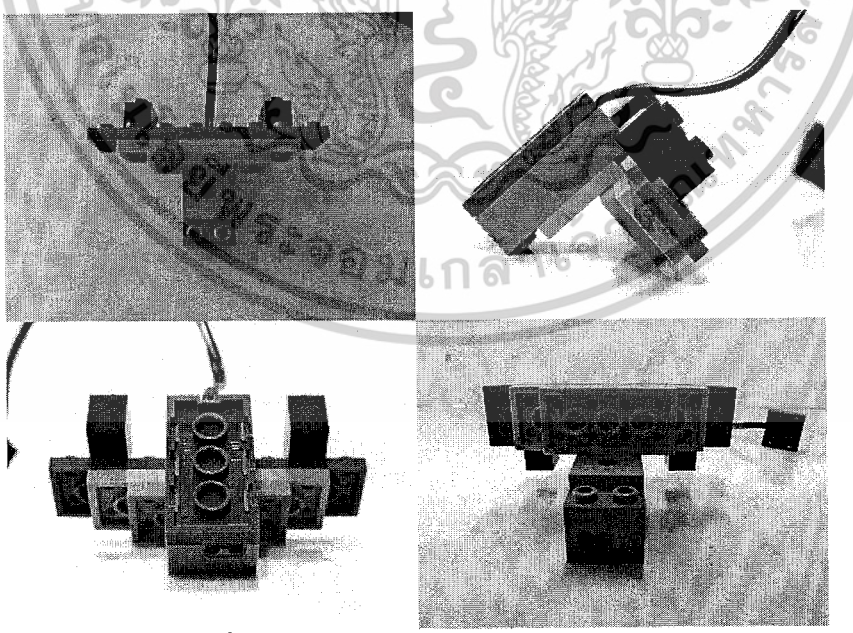
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ประกอบแกนล้ออิสระเพื่อพยุงตัวหุ่นยนต์ เข้ากับโครงสร้างหลักของหุ่นยนต์ ดังรูปที่ 3.31



รูปที่ 3.31 การประกอบส่วนแกนล้ออิสระ

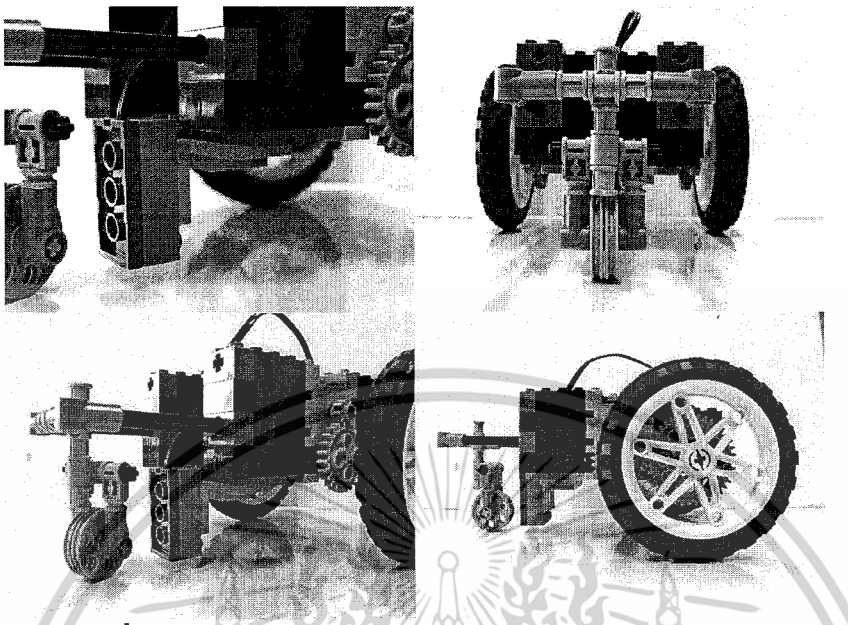
4. ประกอบตัวยึดเซ็นเซอร์แสง ในการยึดเซ็นเซอร์แสงกับโครงสร้างหลักดังรูปที่ 3.32



รูปที่ 3.32 การประกอบส่วนยึดเซ็นเซอร์แสง

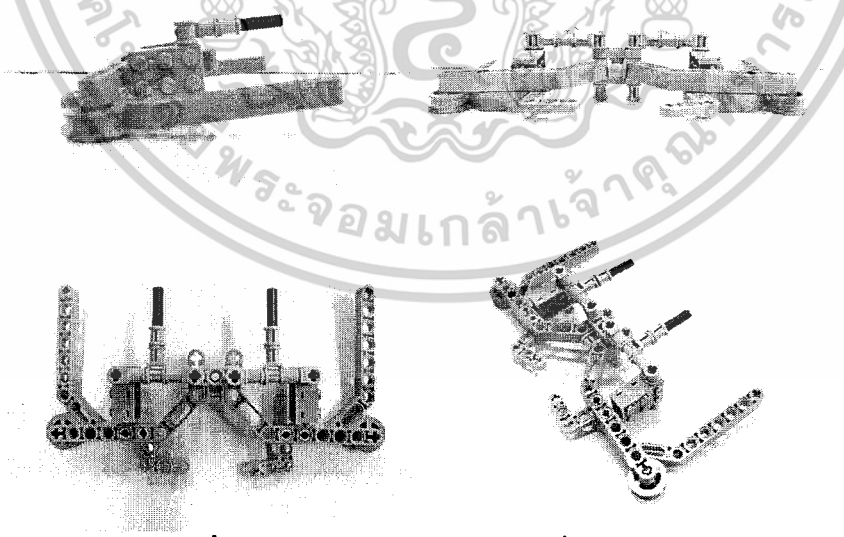
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการประกอบเซ็นเซอร์แสงเข้ากับโครงสร้างหลัก ดังรูปที่ 3.33



รูปที่ 3.33 การประกอบส่วนเซ็นเซอร์แสงเข้ากับโครงสร้างหลัก

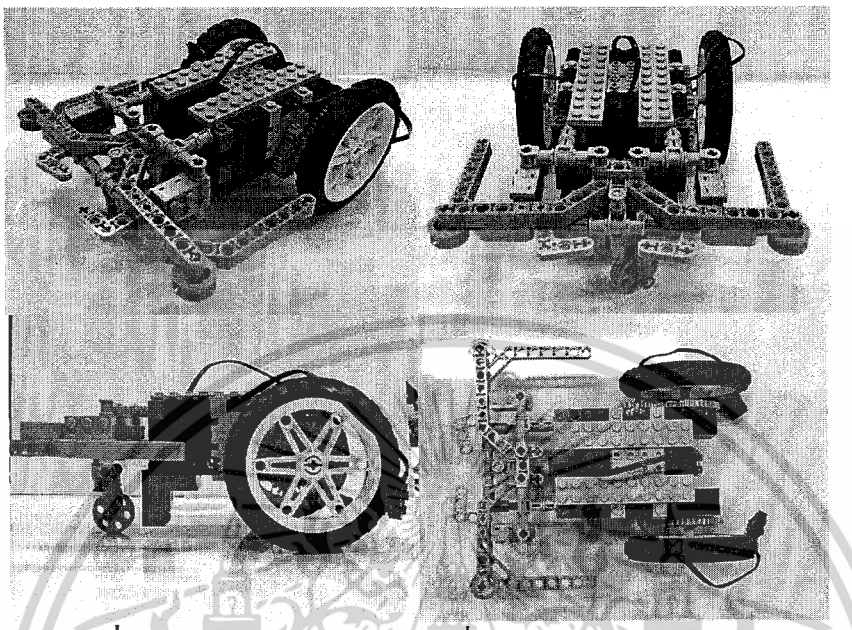
6. ทำการประกอบก้านตรวจจับสิ่งกีดขวางโดยใช้เซ็นเซอร์สัมผัสเป็นแกนหลักทั้งทางด้านซ้ายและขวา ดังรูปที่ 3.34



รูปที่ 3.34 การประกอบก้านตรวจจับสิ่งกีดขวาง

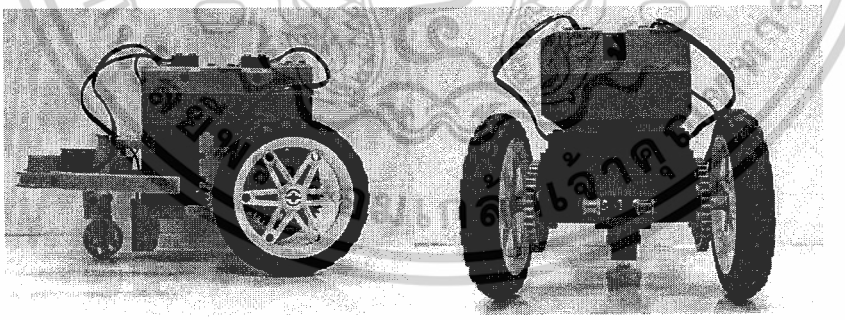
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการประกอบก้านตรวจจับสิ่งกีดขวางเข้ากับโครงสร้างหลัก และ ทำการใส่แกนฐานรองรับตัวอาร์ซีเอ็กซ์ ดังรูปที่ 3.35



รูปที่ 3.35 การประกอบก้านตรวจจับสิ่งกีดขวางกับโครงสร้างหลัก

8. ทำการประกอบตัวอาร์ซีเอ็กซ์เข้ากับโครงสร้างหลักที่เตรียมมาจนถึงขั้นสุดท้าย และทำการเดินสายส่งข้อมูลจากมอเตอร เซ็นเซอร์แสง และ เซ็นเซอร์สัมผัสเข้ากับตัวอาร์ซีเอ็กซ์ ดังรูป 3.36



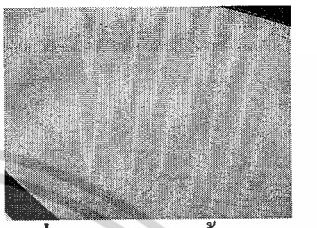
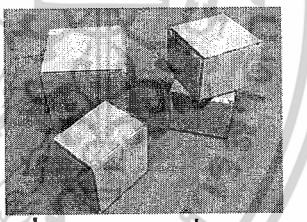
รูปที่ 3.36 การประกอบอาร์ซีเอ็กซ์เข้ากับโครงสร้างหลัก และ เดินสายข้อมูลเข้าสู่อาร์ซีเอ็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การออกแบบสนามทดลอง

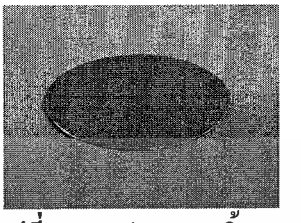
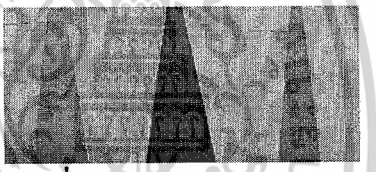
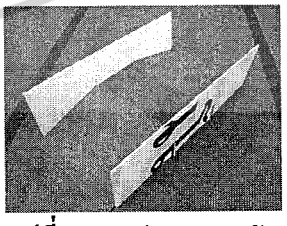
จากความต้องการของระบบ แปลงเป็นรูปแบบในการออกแบบสนามได้ ดังตารางที่ 3.10

ตารางที่ 3.10 การแปลงความต้องการของระบบเป็นความต้องการในการออกแบบสนามทดลอง

ส่วนประกอบของระบบ(หัวข้อ 3.1.2)	การออกแบบสนาม
ข้อ 9 สีของพื้นผิวต่างระดับทั้งทางขึ้นทางและทางลง รวมไปถึงทางธรรมดาต้องต่างกัน	- พื้นสนามทางธรรมดาใช้สีน้ำเงิน  **รูปที่ 3.37** รูปแบบพื้นสนาม
ส่วนประกอบของระบบ(หัวข้อ 3.1.2) **ข้อ 2** สิ่งกีดขวางต้องมีความสูงมากกว่าระดับการตรวจจับของเซนเซอร์สัมผัส **ข้อ 3** สิ่งกีดขวางต้องมีน้ำหนักพอที่จะทำให้หุ่นยนต์เคลื่อนที่ผ่านไปไม่ได้ **ข้อ 5** ต้องสามารถปรับเปลี่ยนสิ่งกีดขวางอุปสรรค, จุดเริ่มต้น และจุดสิ้นสุดสำหรับการทดสอบได้	**การออกแบบสนาม** -สิ่งกีดขวางมีลักษณะ เป็นทรงลูกบาศก์สี่เหลี่ยม 2 ขนาด ถ่วงน้ำหนักโดยใช้ดินน้ำมัน และ สามารถเคลื่อนย้ายได้  **รูปที่ 3.38** รูปแบบสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 การแปลงความต้องการของระบบเป็นความต้องการในการออกแบบสนามทดลอง(ต่อ)

ส่วนประกอบของระบบ(หัวข้อ 3.1.2) **ข้อ 1** ต้องมีรูปแบบจุดสิ้นสุดที่ตรวจจับได้แน่นอน **ข้อ 5** ต้องสามารถปรับเปลี่ยนสิ่งกีดขวางอุปสรรค, จุดเริ่มต้น และจุดสิ้นสุดสำหรับการทดสอบได้	**การออกแบบสนาม** -จุดสิ้นสุดมีลักษณะเป็นวงกลม สีแดงเข้มสามารถเคลื่อนย้ายได้  **รูปที่ 3.39** รูปแบบจุดสิ้นสุด
หน้าที่ของระบบ(หัวข้อ 3.1.1) **ข้อ 3** หุ่นยนต์ต้องไม่เดินออกนอกเขตสนามทดสอบ **ส่วนประกอบของระบบ(หัวข้อ 3.1.2)** **ข้อ 7** ต้องมีขอบเขตสนามทดสอบชัดเจน **ข้อ 8** ต้องมีการแบ่งส่วนของพื้นที่สนามชัดเจน	**การออกแบบสนาม** - ขอบเขตสนามใช้เส้นสีดำ - ส่วนแบ่งเขตใช้เส้นสีแดงอ่อนและเส้นสีเขียวเข้ม  **รูปที่ 3.40** รูปแบบเส้นแบ่งเขต
ส่วนประกอบของระบบ(หัวข้อ 3.1.2) **ข้อ 4** พื้นผิวทางลาดชันต้องมีมุมองศาที่ไม่มากกว่ากำลังที่มอเตอร์รับได้ และมีขอบกั้นเพื่อกันหุ่นยนต์หล่นจากทางลาดชัน **ข้อ 9** สีของพื้นผิวทางต่างระดับทั้งทางขึ้นทางลง รวมถึงทางธรรมดาต้องต่างกัน	**การออกแบบสนาม** - ทางลาดชันทั้งสองด้านมีขนาด 15 องศา และ พื้นทางชันสีเขียวอ่อน ด้านข้างมีขอบกั้นกันหุ่นยนต์ตกขอบ  **รูปที่ 3.41** รูปแบบทางชัน

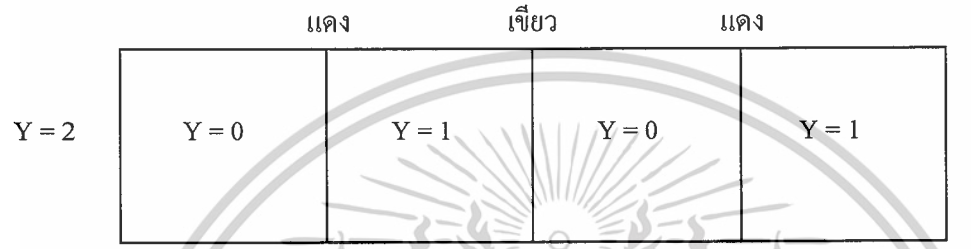
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.1 การออกแบบเส้นแบ่งเขตสนาม

1. แนวคิด

แบ่งพื้นที่สนามเพื่อกำหนดการเคลื่อนที่ของหุ่นยนต์ให้รับรู้ทิศทางในการเคลื่อนที่เข้าหาจุดสิ้นสุด

2. รูปแบบการแบ่งพื้นที่ และการกำหนดสี



รูปที่ 3.42 การแบ่งพื้นที่และการกำหนดสีเส้นแบ่งเขตสนาม

การกำหนดอัลกอริทึมจะมี 2 สี เพื่อเป็นการกำหนดลำดับของพื้นสนามให้หุ่นยนต์แยกแยะออกว่า ควรจะผ่านไปหรือไม่ โดยจะกำหนดเรียงลำดับสีดังรูปที่ 3.42 ซึ่งเริ่มแรกจะกำหนด Y = 2 และให้ Y = 0 เมื่อมีการเริ่มโปรแกรม ในกรณีที่พื้นที่คือ Y = 0 ถ้าเจอเส้นสีแดง คือเจอพื้นที่ต่อไปให้มีการเคลื่อนที่ผ่านไปเหมือนเป็นพื้นสนามธรรมดา และจะกำหนดพื้นที่เป็น Y = 1 แต่ถ้าเจอเส้นสีเขียว คือพื้นที่ที่เคยผ่านมาแล้ว จะมีการตอบสนองเหมือนเจอเส้นขอบเขตสนาม (เส้นสีดำ) ไม่ให้ผ่านกลับ และในกรณีที่พื้นที่เป็น Y = 1 จะกลับกัน ถ้าเจอเส้นสีเขียว คือเจอพื้นที่ต่อไปให้มีการเคลื่อนที่ผ่านไป และจะกำหนดพื้นที่เป็น Y = 0 แต่ถ้าเจอเส้นสีแดงจะตอบสนองเหมือนเส้นขอบเขตสนาม โดยแบ่งเป็นกรณีต่างๆดังตารางที่ 3.11

ตารางที่ 3.11 การตอบสนองต่อการตรวจจับเส้นแบ่งเขตสนาม

พื้นที่	เส้นแบ่งเขต	รูปแบบการตอบสนอง
Y = 0	สีแดง	พื้นสนามธรรมดา
Y = 0	สีเขียว	เส้นขอบเขตสนาม
Y = 1	สีแดง	เส้นขอบเขตสนาม
Y = 1	สีเขียว	พื้นสนามธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 การออกแบบทางขึ้นทางลง

1. แนวคิด

แบ่งพื้นที่เป็น 3 ส่วน คือทางขึ้น, ทางธรรมดา และทางลง เพื่อกำหนดการค้นหาและรูปแบบการตอบสนอง

2. รูปแบบการแบ่งพื้นที่ และการกำหนดสี

รูปที่ 3.43 การแบ่งพื้นที่และการกำหนดสีทางขึ้นลง

การกำหนดอัลกอริทึม จะใช้เพียงสีเดียวบ่งบอกว่าเป็นทางชัน แต่บริเวณที่เป็นพื้นเรียบข้างบน จะเป็นสีน้ำเงินเหมือนพื้นสนามธรรมดาเพื่อแยก ทางขึ้น – ทางลง และเพื่อกำหนดความเร็ว ดังรูปที่ 3.43 การทำงานของหุ่นยนต์จะมีการจดจำพื้นผิวที่จับได้ โดยจะจดจำที่ค่า X ซึ่งจะกำหนดให้ X = 6 คือทางขึ้น หรือ X = 7 คือทางลง เมื่อมีการเจอทางชัน (สีเขียวอ่อน) และจะมีการกำหนดค่า Z ไว้สำหรับแบ่งแยกทางขึ้นทางลงและตรวจสอบสถานะการเคลื่อนที่ โดยทุกครั้งที่มีการเจอสีเขียวอ่อนจะทำการสลับค่า Z จาก 0 เป็น 1 หรือ 1 เป็น 0 ค่าเริ่มต้นคือ 1 หลังจากนั้นมีการตรวจเช็ค ว่าเจอสีเขียวอ่อน และค่าพื้นที่เก่าที่เก็บไว้ไม่ใช่ทั้งทางขึ้น (X = 6) และทั้งทางลง (X = 7) จะทำการสลับค่า Z หมายถึง มีการขึ้น หรือลงอีกครั้ง และจะทำการเช็คจากค่า Z ว่า ถ้า Z = 0 หมายถึงพื้นผิวเป็นแบบทางขึ้น ให้ทำการบันทึกเก็บค่า X = 6ในการส่งต่อการตอบสนองต่อไป ถ้า Z =1 จะบันทึกเก็บค่า X = 7 ไปส่งต่อการตอบสนองดัง ตารางที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 รูปแบบการตรวจจับ และตอบสนองต่อทางขึ้นทางลง

ค่าการตรวจสอบ	ค่าพื้นที่เก่า	การตอบสนอง
Z = 0	X != 6, 7	เปลี่ยนค่า Z = 1 แสดงว่าเริ่มเจอทางลง ส่งการตอบสนอง และกำหนดค่า X = 6 เพื่อแสดงว่าเป็นทางลง
Z = 0	X = 6	ไม่เปลี่ยนค่าใดๆ แสดงว่ายังอยู่ในบริเวณทางขึ้น
Z = 0	X = 7	แสดงว่าเจอทางขึ้นให้ทำการเปลี่ยนค่า X = 6 และส่งการตอบสนองต่อไป
Z = 1	X != 6, 7	เปลี่ยนค่า Z = 0 แสดงว่าเริ่มเจอทางขึ้น ส่งการตอบสนอง และกำหนดค่า X = 6 เพื่อแสดงว่าเป็นทางขึ้น
Z = 1	X = 6	แสดงว่าเจอทางลงให้ทำการเปลี่ยนค่า X = 6 และส่งการตอบสนองต่อไป
Z = 1	X = 7	ไม่เปลี่ยนค่าใดๆ แสดงว่ายังอยู่ในบริเวณทางขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและวิเคราะห์ผล

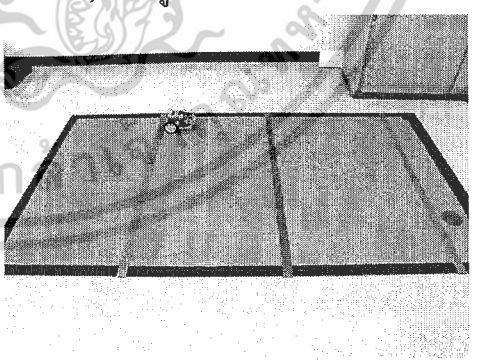
การทดลองหุ่นยนต์ตรวจจับเส้นทาง ทดลองเพื่อที่จะตอบสนองวัตถุประสงค์ของโครงงานหุ่นยนต์ตรวจจับเส้นทาง หัวข้อการทดลองได้จำแนกเป็นกลุ่มความต้องการตามหน้าที่ของระบบ (หัวข้อ 3.1.1) และ ส่วนประกอบของระบบ (หัวข้อ 3.1.2) ได้ การทดลองบนทางเรียบปกติ การทดลองบนทางที่มีสิ่งกีดขวาง การทดลองบนทางที่มีทางชัน และ การทดลองการทำงานของหุ่นยนต์ทั้งระบบ

4.1 การทดลองบนทางเรียบปกติ

วัตถุประสงค์ เพื่อทำการดูผลการตอบสนองของหุ่นยนต์ เมื่อเจอพื้นสนาม เส้นขอบเขตสนามเส้นแบ่งสนาม และ จุดสิ้นสุด

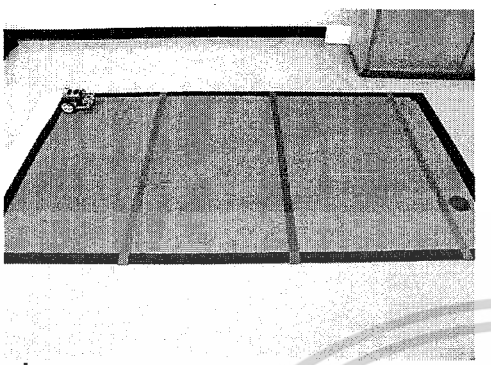
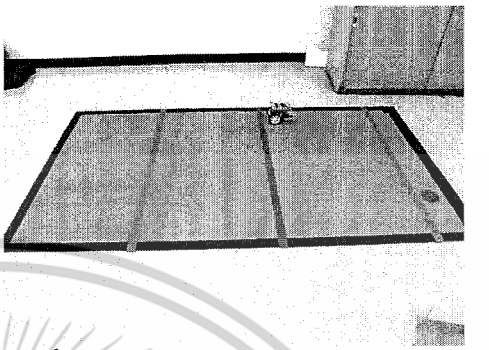
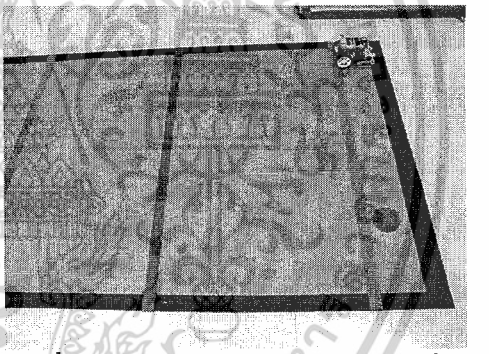
สมมติฐาน ถ้าเซนเซอร์แสงสามารถแยกแยะความเข้มของพื้นสนามได้ หุ่นยนต์จึงสามารถเดินผ่านเส้นแบ่งสนาม โดยไม่ย้อนกลับมาที่เส้นแบ่งเดิมอีก หุ่นยนต์ไม่เดินออกนอกสนาม และ หุ่นยนต์จะหยุดการทำงานเมื่อพบกับจุดสิ้นสุด

ตารางที่ 4.1 การทดลองบนทางเรียบปกติ

วิธีการทดลอง	**ผลการทดลอง**
1.นำหุ่นยนต์ไว้ที่ฝั่งซ้ายของสนาม และ นำจุดสิ้นสุดไว้ที่ฝั่งขวาของสนาม รวมถึงมีเส้นแบ่งเขตสนามสามเส้นสองสี โดยเรียงเส้นสีแบ่งเขตสนาม ดังนี้ สีแดงอ่อน-> สีเขียวเข้ม-> สีแดงอ่อน จากซ้ายไปขวา ซึ่งระยะของพื้นที่ของเส้นแบ่งเขตต้องเท่ากัน และ เหลือพื้นที่ ที่จะวางจุดสิ้นสุด ให้มีพื้นที่น้อยที่สุดดังรูปที่ 4.1	1. หุ่นยนต์สามารถเดินผ่านเส้นแบ่งเขตส่วนแรก (สีแดงอ่อน)ได้ดังรูปที่ 4.2  **รูปที่ 4.2** หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 1

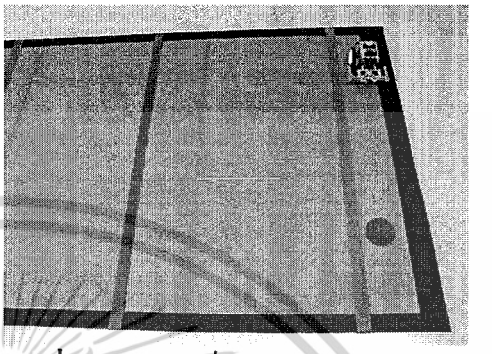
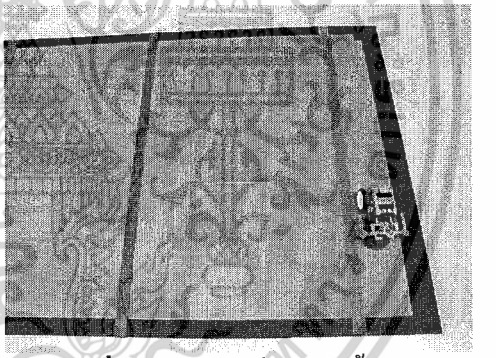
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การทดลองบนทางเรียบปกติ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
 **รูปที่ 4.1** รูปแบบสนามการทดลองบนทางเรียบธรรมดา	2. หุ่นยนต์สามารถเดินผ่านเส้นแบ่งเขตส่วนที่สอง(สีเขียวเข้ม)ได้ดังรูป 4.3  **รูปที่ 4.3** หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 2 3. หุ่นยนต์สามารถเดินผ่านเส้นแบ่งเขตส่วนที่สาม(สีแดงอ่อน)ได้ดังรูปที่ 4.4  **รูปที่ 4.4** หุ่นยนต์ผ่านเส้นแบ่งเขตส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การทดลองบนทางเรียบปกติ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
	4. เมื่อหุ่นยนต์ได้พบกับเส้นขอบเขตสนาม(สีดำ) จะทำการเปลี่ยนทิศทางการเคลื่อนที่ไปทางขวา เพื่อรักษาให้หุ่นยนต์อยู่ในสนามดังรูปที่ 4.5  **รูปที่ 4.5** หุ่นยนต์เมื่อถึงพบขอบเขตสนาม 5. หุ่นยนต์เดินไปเรื่อยๆจนพบจุดสิ้นสุด ก็เป็นอันหยุดการทำงานดังรูปที่ 4.6  **รูปที่ 4.6** หุ่นยนต์เมื่อถึงจุดสิ้นสุด

วิเคราะห์ รูปแบบการทำงานของหุ่นยนต์มีการเคลื่อนไหวผ่านเส้นแบ่งเขตต่างๆได้ตามที่ได้ตั้งการตอบสนองไว้ เมื่อเจอกับเส้นขอบเขตจึงหลบหลีกไปในทิศทางขวา และเจอกับจุดสิ้นสุด จึงหยุดการทำงานตามคาดหมาย

สรุป แสงในสภาพแวดล้อมค่อนข้างที่จะมีผลมาก เนื่องจากเซนเซอร์แสงรับค่าความเข้มของสีพื้นสนามแต่ละครั้งไม่เท่ากัน ในสภาวะแสงที่ไม่เท่ากัน ส่งผลทำให้เกิดปัญหาทางด้านซอฟท์แวร์ตามมา

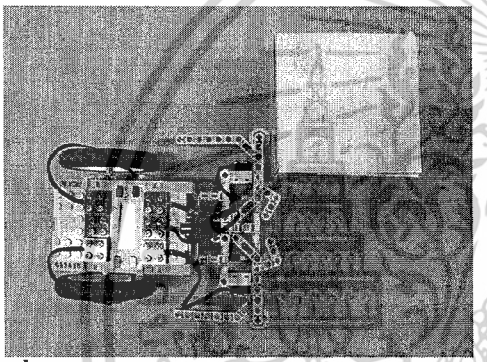
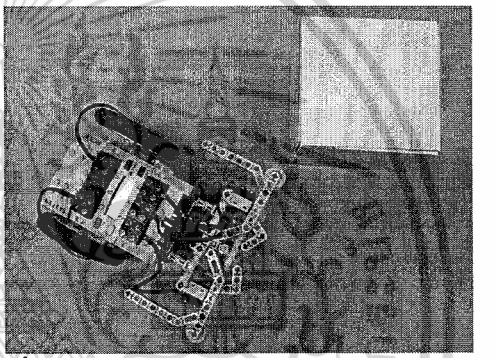
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองบนทางที่มีสิ่งกีดขวาง

วัตถุประสงค์ เพื่อดูการตอบสนองของหุ่นยนต์เมื่อกันชนของหุ่นยนต์ได้กระทบกับสิ่งกีดขวางในมุมการชนที่ต่างกัน

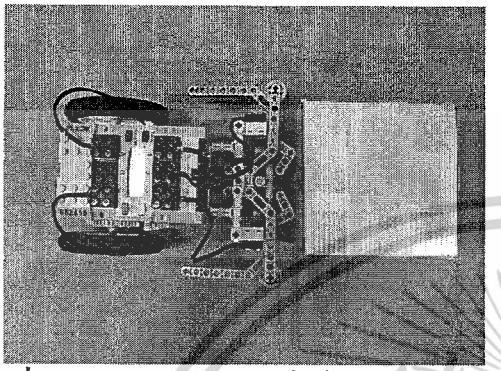
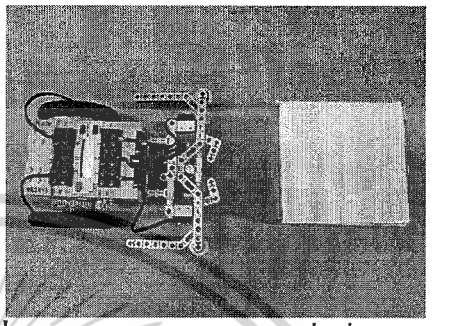
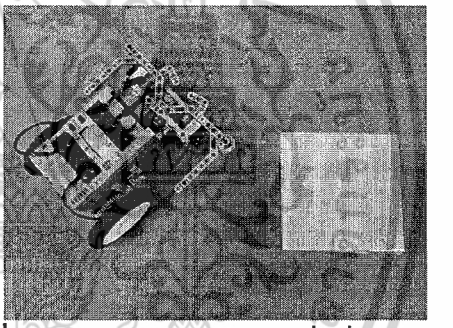
สมมติฐาน ถ้ากันชนได้รับการกระทบ แล้ว เซนเซอร์ถูกกด หุ่นยนต์จะสามารถเคลื่อนที่หลีกเลี่ยงโดย เปลี่ยนทิศการเคลื่อนที่ ไปในทิศทางอื่นที่ไม่ใช้ทิศทางเดิม

ตารางที่ 4.2 การทดลองบนทางที่มีสิ่งกีดขวาง

วิธีการทดลอง	**ผลการทดลอง**
1. ทดลองการชนทางด้านซ้ายของหุ่นยนต์โดยการ นำสิ่งกีดขวางไปวางในตำแหน่งทางด้านซ้าย ของหุ่นยนต์ดังรูปที่ 4.7  **รูปที่ 4.7** การทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านซ้าย)	1. หุ่นยนต์ทำการเปลี่ยนทิศทางการเคลื่อนที่ไปในทางขวา เพื่อหลบหลีกสิ่งกีดขวางดังรูปที่ 4.8  **รูปที่ 4.8** ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านซ้าย)

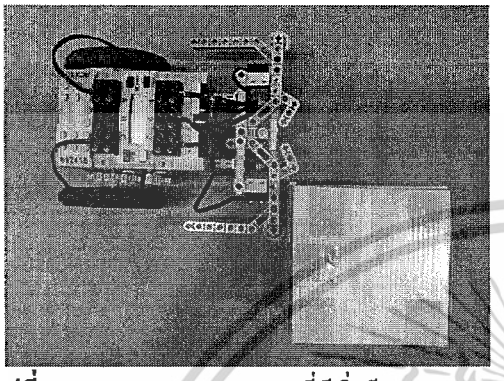
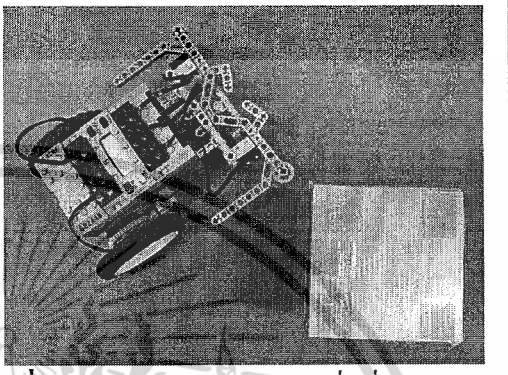
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การทดลองบนทางที่มีสิ่งกีดขวาง (ต่อ)

วิธีการทดลอง	ผลการทดลอง
2. ทดลองการชนด้านหน้าตรงของหุ่นยนต์โดยการนำสิ่งกีดขวางไปวางในตำแหน่งทางด้านหน้าของหุ่นยนต์ดังรูปที่ 4.9  **รูปที่ 4.9** การทดลองบนทางที่มีสิ่งกีดขวาง (การชนตรงกลาง)	2. หุ่นยนต์ทำการเคลื่อนที่ถอยหลังก่อนจากนั้นค่อยเปลี่ยนทิศทางการเคลื่อนที่ไปในทางซ้ายเพื่อหลบหลีกสิ่งกีดขวางดังรูปที่ 4.10-4.11  **รูปที่ 4.10** ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนด้านหน้า)  **รูปที่ 4.11** ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนด้านหน้า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การทดลองบนทางที่มีสิ่งกีดขวาง (ต่อ)

วิธีการทดลอง	ผลการทดลอง
3. ทดลองการชนด้านขวา โดยนำสิ่งกีดขวางไปวางในตำแหน่งในตำแหน่งทางด้านขวาของหุ่นยนต์ดังรูปที่ 4.12  **รูปที่ 4.12** การทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านขวา)	3. หุ่นยนต์ทำการเปลี่ยนทิศทางการเคลื่อนที่ไปในทางซ้าย เพื่อหลบหลีกสิ่งกีดขวางดังรูปที่ 4.13  **รูปที่ 4.13** ผลการทดลองบนทางที่มีสิ่งกีดขวาง (การชนทางด้านขวา)

วิเคราะห์ รูปแบบการชนต่างของหุ่นยนต์จะได้รับการตอบสนองในการหลบหลีกไปในทิศทางที่เหมาะสม ท่ามีการชนบริเวณด้านซ้าย จะหลบหลีกไปในทิศทางขวา ท่ามีการชนบริเวณด้านขวา จะหลบหลีกไปในทิศทางซ้าย ถ้าชนด้านหน้าจะถอยหลังและหลบหลีกไปในทิศทางใดทิศทางหนึ่ง ซึ่งเป็นไปตามการตอบสนองที่ได้ตั้งไว้

สรุป น้ำหนักของสิ่งกีดขวางมีผลอย่างมากต่อการชน ถ้าน้ำหนักของสิ่งกีดขวางน้อยเกินไปหุ่นยนต์จะดันสิ่งกีดขวางแทนการชน ด้วยความแรงของมอเตอร์นำพาไป และการตอบสนองต่อการชน กับรูปแบบการตอบสนองเป็นไปอย่างรวดเร็ว ถูกต้องตามการสัมผัสที่เกิดขึ้น

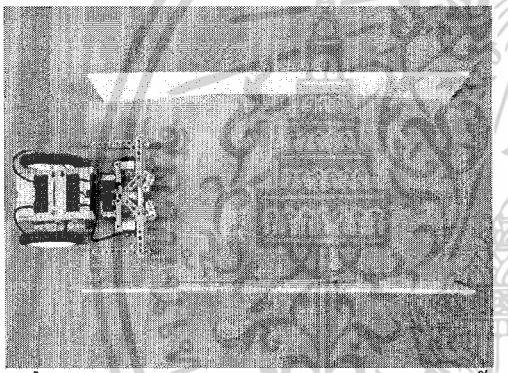
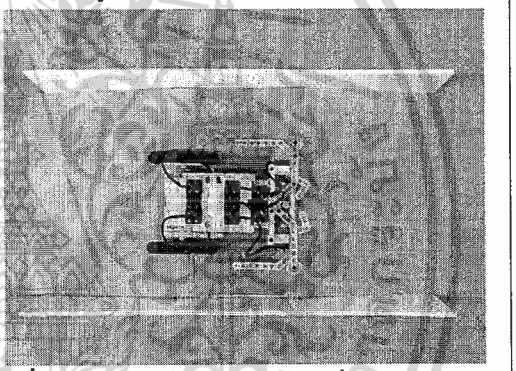
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองบนทางที่มีทางชัน

วัตถุประสงค์ เพื่อดูการตอบสนองของหุ่นยนต์เมื่อหุ่นยนต์เคลื่อนบนทางชันในแนวขาขึ้นและขาลง

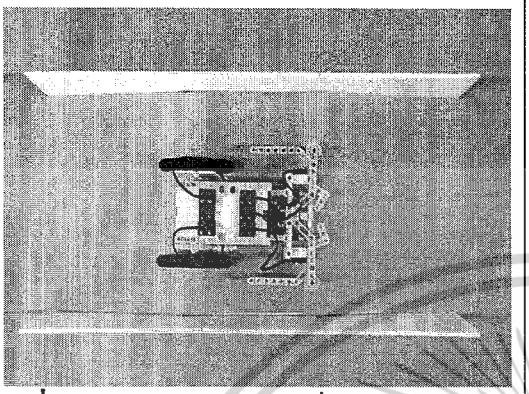
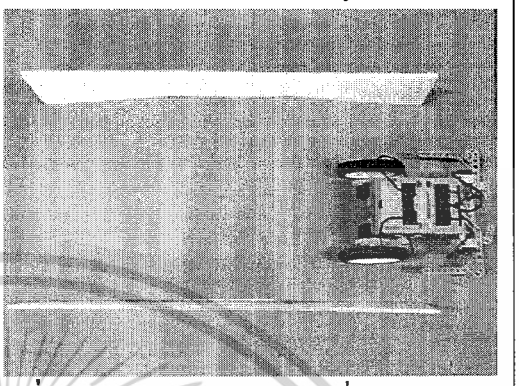
สมมติฐาน ถ้าหุ่นยนต์มีแหล่งพลังงานที่สมบูรณ์ และ องศาความชัน สอดคล้องกับเร็วในการเคลื่อนที่ของหุ่นยนต์แล้ว หุ่นยนต์สามารถขึ้นความชันด้วยความเร่งเพื่อรักษาความเร็ว และลงทางชันด้วยความหน่วงเพื่อรักษาความเสถียรของตัวหุ่นยนต์ได้

ตารางที่ 4.3 การทดลองบนทางที่มีทางชัน

วิธีการทดลอง	**ผลการทดลอง**
1. ทดลองการขึ้นทางชัน โดยนำหุ่นยนต์มาวางแนวเดียวกับทางชันดังรูปที่ 4.14  **รูปที่ 4.14** การทดลองบนทางที่มีทางชัน (การขึ้นทางชัน)	1. หุ่นยนต์สามารถขึ้นทางชันด้วนความเร่ง โดยไม่คว่ำ ดังรูปที่ 4.15  **รูปที่ 4.15** ผลการทดลองบนทางที่มีทางชัน (การขึ้นทางชัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 การทดลองบนทางที่มีทางชัน (ต่อ)

วิธีการทดลอง	**ผลการทดลอง**
2. ทดลองการลงทางชัน โดยนำหุ่นยนต์มาวางแนวเดียวกับทางลงดังรูปที่ 4.16  **รูปที่ 4.16** การทดลองบนทางที่มีทางชัน (การลงทางชัน)	2. หุ่นยนต์สามารถ ลงทางชันด้วยความหน่วงของมอเตอร์ และไม่ไถลลงมาดังรูปที่ 4.17  **รูปที่ 4.17** ผลการทดลองบนทางที่มีทางชัน (การลงทางชัน)

วิเคราะห์ การเร่งจากการขึ้น กับการหน่วงจากการลงทางชันนั้น จะขึ้นอยู่กับทิศทางการขึ้น และแรงที่ได้รับจากพื้นผิวก่อนหน้านั้น รวมไปถึงค่าพลังงานในการส่งแรงให้กับมอเตอร์ให้เพียงพอต่อค่ามุมองศาของทางชัน

สรุป แหล่งพลังงานของหุ่นยนต์มีผลอย่างมากถึงมากที่สุดต่อการเคลื่อนที่ขึ้นทางชันเนื่องจากหุ่นยนต์ไม่สามารถขึ้นทางชันได้ถ้าแหล่งพลังงานมีพลังงานไม่เพียงพอต่อการแจกจ่ายไปยังมอเตอร์ทั้ง 2 ตัวและองศาความชันมีผลต่อความสามารถในการเคลื่อนที่ขึ้นของหุ่นยนต์ถ้ามีความชันมากเกินไปหุ่นยนต์จะไม่สามารถขึ้นทางชันได้ และ จะเกินการฝืนของมอเตอร์ทำให้มอเตอร์ได้รับความเสียหายได้และการเคลื่อนที่ลงทางชันหุ่นยนต์จะคว่ำ ทำให้อุปกรณ์ต่างๆ ของหุ่นยนต์เสียหายได้

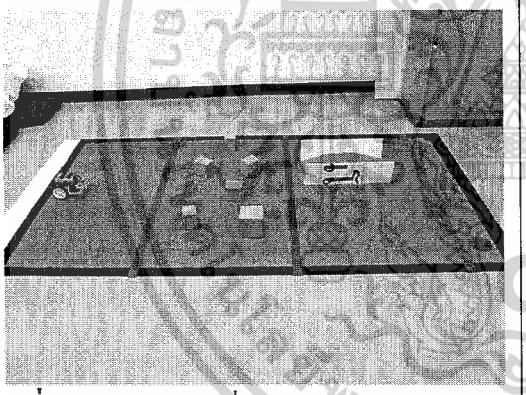
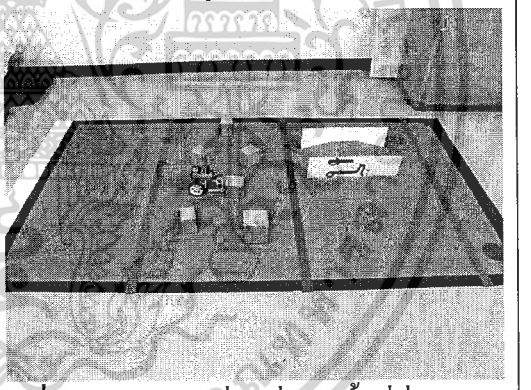
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ

วัตถุประสงค์ เพื่อทดสอบการทำงานของหุ่นยนต์ตามการออกแบบ และ ความต้องการที่กำหนดทั้งระบบ

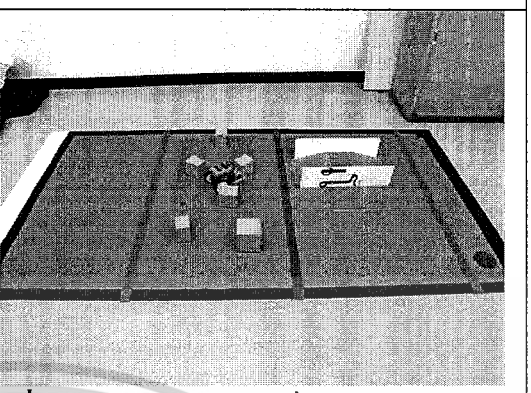
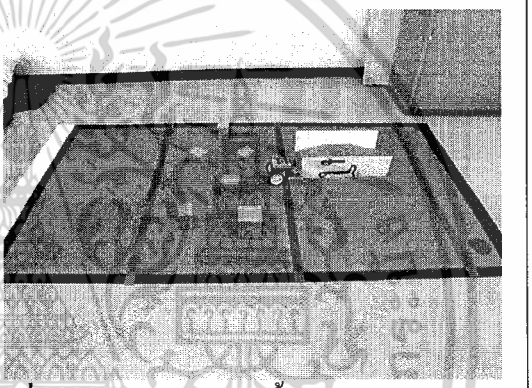
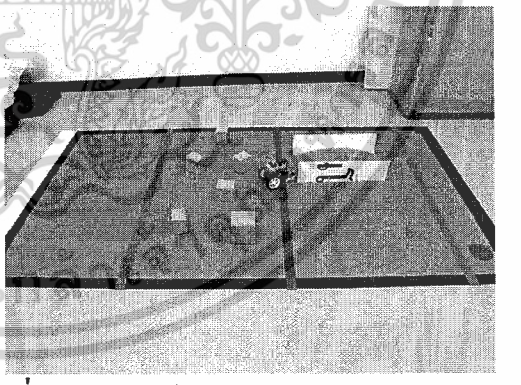
สมมติฐาน ถ้าการออกแบบเป็นไปตามความต้องการที่กำหนดแล้ว หุ่นยนต์จะสามารถไปผ่านอุปสรรคไปถึงจุดสิ้นสุดที่กำหนดไว้ได้

ตารางที่ 4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ

วิธีการทดลอง	ผลการทดลอง
1.วางจุดเริ่มต้นของหุ่นยนต์ในส่วนของพื้นที่ซ้ายสุดของสนามและวางจุดสิ้นสุดไว้ในสวนของพื้นที่ขวาสุดของสนาม โดยให้พื้นที่ส่วนของหุ่นยนต์เริ่มเป็นพื้นที่ว่างเปล่า ถัดมาเป็นพื้นที่ที่มีแต่สิ่งกีดขวาง และ ถัดมาเป็นพื้นที่ ที่มีทางต่างระดับ สุดท้ายจะเป็นพื้นที่ที่มีจุดสิ้นสุดอยู่ ดังรูป 4.18	หุ่นยนต์สามารถเดินจากจุดเริ่มต้นไปยังจุดสิ้นสุดที่กำหนดไว้ได้ โดยหุ่นยนต์สามารถหลบหลีกสิ่งกีดขวางในส่วนแรกได้เป็นอย่างดี พอเข้ามาในพื้นที่ส่วนที่ 2 หุ่นยนต์สามารถที่จะจัดการตัวเองโดยเร่งมอเตอร์เพื่อขึ้นทางชัน และสามารถที่จะหน่วงมอเตอร์เพื่อลดความเร็วในการลงทางชันได้ดังรูปที่ 4.19-4.29
 **รูปที่ 4.18** แบบสนามที่ใช้ทดลองการทำงานของหุ่นยนต์	 **รูปที่ 4.19** หุ่นยนต์เคลื่อนที่เข้าสู่พื้นที่สิ่งกีดขวาง

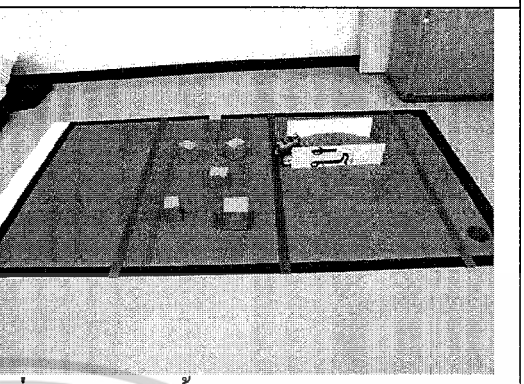
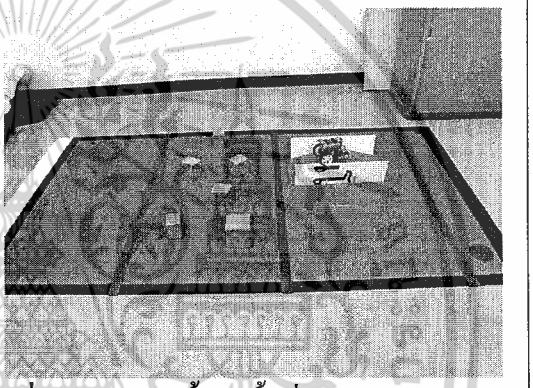
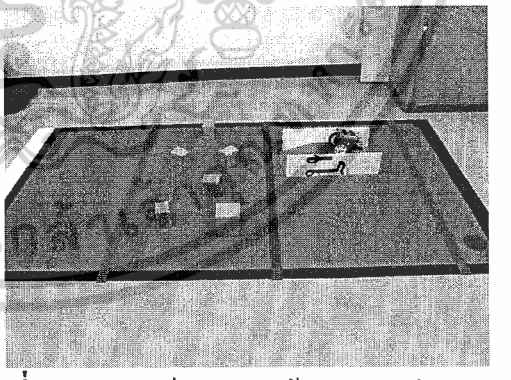
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.20** หุ่นยนต์หลบหลีกสิ่งกีดขวางระหว่างทางเดิน  **รูปที่ 4.21** หุ่นยนต์ชนขอบกั้นสะพาน  **รูปที่ 4.22** หุ่นยนต์หลบหลีกขอบสะพาน

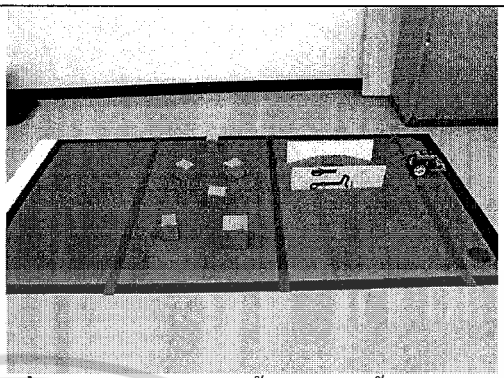
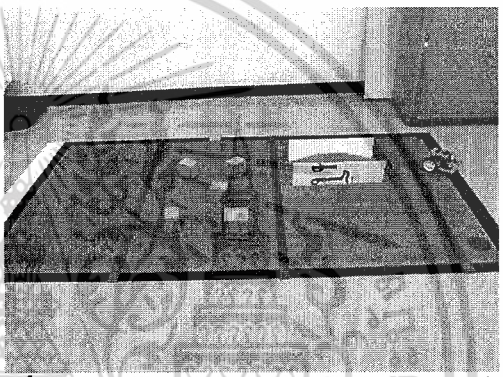
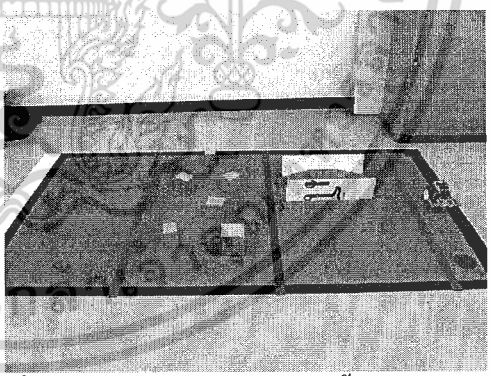
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.23** หุ่นยนต์ขึ้นสะพานด้วยความเร่ง  **รูปที่ 4.24** หุ่นยนต์ขึ้นถึงพื้นที่ปกติบนทางชัน  **รูปที่ 4.25** หุ่นยนต์ลงทางชันด้วยความหน่วง

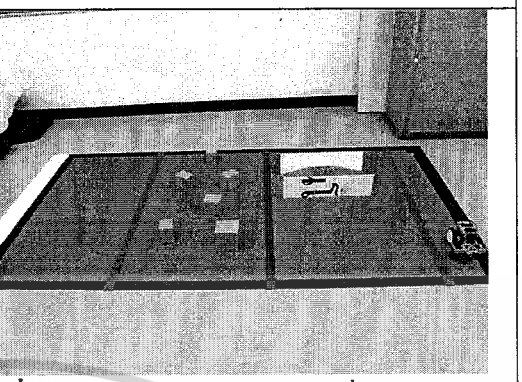
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.26** หุ่นยนต์เข้าถึงพื้นที่ที่มีจุดสิ้นสุด  **รูปที่ 4.27** หุ่นยนต์ทำการเปลี่ยนทิศการเคลื่อนที่เพื่อไม่ให้ออกนอกสนาม  **รูปที่ 4.28** หุ่นยนต์เคลื่อนที่เข้าจุดสิ้นสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดลองการทำงานของหุ่นยนต์ทั้งระบบ (ต่อ)

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.29** หุ่นยนต์หยุดการทำงานเมื่อพบกับจุดสิ้นสุด

วิเคราะห์ การตอบสนองที่ได้รับจากการตรวจจับของหุ่นยนต์นั้นรวดเร็วทันที เป็นการตอบสนองแบบเรียลไทม์ (Real-Time) โดยรูปแบบการตอบสนองจะเป็นไปตามที่ตั้งไว้ แต่จะมีความคลาดเคลื่อนอันเนื่องมาจากการตรวจจับที่ไม่ดีพอ ทำให้มีการตอบสนองที่ผิดเพี้ยนไปบ้าง

สรุป ผลการทดลองที่ได้เป็นไปตามจุดประสงค์ในการสร้างระบบหุ่นยนต์ตรวจจับเส้นทาง แต่ยังขาดในเรื่องความกระชับของเวลาในการเดินจากจุดเริ่มต้นไปยังจุดสิ้นสุด ความแม่นยำในการตอบสนองการตรวจจับต่างๆ และกรณีค่าที่อ่านได้จากเซนเซอร์แสงเพี้ยนทำให้กระบวนการคิดตอบสนองของหุ่นยนต์ผิดไปจากปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปโครงงานและข้อเสนอแนะ

5.1 สรุปผลโครงการ

โครงงานระบบหุ่นยนต์ตรวจจับเส้นทาง เป็นโครงงานที่จัดทำมาเพื่อพัฒนาหุ่นยนต์โดยใช้เทคโนโลยีระบบฝังตัวนำมาประมวลผลและตอบสนองคำสั่งของผู้ใช้ รูปแบบการพัฒนาได้นำเอาแนวทางการพัฒนาเชิงวัตถุที่ใช้ยูเอ็มแอลมาเป็นเครื่องมือในการพัฒนา ที่จะนำเสนอแนวคิด การวางแผน การปฏิบัติงานให้เห็นในรูปแบบที่ง่ายต่อการเข้าใจของผู้ใช้ ในการอธิบายภาพรวมของระบบ และลงลึกไปถึงส่วนของขอบเขตขั้นตอนการทำงาน การติดต่อของกลุ่มคำสั่งต่าง รวมไปถึงโครงสร้างของโปรแกรมที่มีลำดับขั้นตอนการติดต่ออีกด้วย ขอบเขตการพัฒนาหุ่นยนต์จะอยู่ภายใต้สนามที่กำหนด โดยหุ่นยนต์สามารถตรวจจับสิ่งแวดล้อมเพื่อการเดินไปหาจุดสิ้นสุดที่เป็นเป้าหมายได้

หุ่นยนต์ตรวจจับเส้นทางที่ได้จัดทำขึ้นในโครงงานนี้ มีความสามารถในการตรวจจับอุปสรรคต่างๆที่อยู่ภายในสนาม ทั้งการสัมผัสจากเซนเซอร์สัมผัส แสงของพื้นผิวจากเซนเซอร์แสง และจะทำการตอบสนองต่อการตรวจจับนั้นๆในทันที จนสามารถเดินไปถึงจุดสิ้นสุดได้ แต่ความสามารถในการตรวจจับของหุ่นยนต์ยังไม่แน่นอนทั้งหมด ทำให้เกิดการผิดพลาดในการเคลื่อนที่ในบางครั้ง

โครงงานนี้บรรลุวัตถุประสงค์ในการเป็นต้นแบบของการออกแบบผลิตภัณฑ์ที่มีระบบฝังตัวด้วยความริเริ่มสร้างสรรค์ และสามารถนำไปประยุกต์ประยุกต์ใช้ผลิตภัณฑ์ให้เป็นต้นแบบของผลิตภัณฑ์ใหม่ๆอาทิเช่น ไม้เท้าคนตาบอด รถขนส่งอัตโนมัติ ฯลฯ รวมไปถึงส่งเสริมให้เกิดการพัฒนาระบบฝังตัวทั้งทางด้านฮาร์ดแวร์ และซอฟแวร์ สำหรับหุ่นยนต์ตรวจจับเส้นทาง

5.2 ปัญหาและอุปสรรค

5.2.1 ปัญหาทางด้านฮาร์ดแวร์

ปัญหาที่เกิดจากเซ็นเซอร์แสง เกิดเนื่องจากค่าที่เซ็นเซอร์แสงอ่านได้เป็นค่าความเข้มของแสงในรูปแบบเกรย์สเกล (Gray scale) ซึ่งระยะห่างของรูปแบบการตรวจจับระหว่างเซ็นเซอร์กับพื้นที่ทดลองมีผลต่อค่าที่ได้จากการตรวจจับ ยิ่งระยะห่างระหว่างเซ็นเซอร์กับพื้นทดลองมีมากจะทำให้มีผลให้จับค่าแสงได้มากขึ้น ซึ่งค่าที่ได้จะเพี้ยนไปจากเดิม ส่งผลทำให้กระบวนการคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดสินใจของหุ่นยนต์ไม่แน่นอนได้ จากปัญหาเรื่องระยะห่าง ยังมีปัญหาเรื่องความสว่างที่เกิดขึ้นเวลาตรวจจับอีกด้วย เนื่องจากสภาพแวดล้อมของแสง รวมไปถึงสถานที่ทดสอบ มีมุมองศาความสว่างที่ไม่เท่ากัน ก็จะทำให้เกิดปัญหาเดียวกับปัญหาระยะห่างอีกด้วย

ปัญหาเซ็นเซอร์สัมผัส ค่าที่ตรวจจับได้จากเซ็นเซอร์สัมผัสระหว่างการทำงานของหุ่นยนต์ทั้งสองฝั่ง แตกต่างจากการทดสอบเซ็นเซอร์สัมผัสเพียงตัวเดียว ซึ่งค่าที่ได้รับมีค่าไม่แน่นอนและ มีช่วงของค่าที่แตกต่างกันในการกดเซ็นเซอร์สัมผัสในแต่ละครั้ง ทำให้เกิดการสับสนกับการกำหนด ค่าของการสัมผัสจากเซ็นเซอร์สัมผัส แต่จากผลการทดลองทำให้สามารถแก้ไขปัญหาได้เนื่องจากค่าที่บันทึกจากการทดลองที่เซ็นเซอร์สัมผัสไม่ถูกกด จะมีค่าเป็นฐาน 16 มากกว่า F000 เสมอ แต่สูงสุดเพียง 4 หลัก ส่วนค่าที่ได้จากการสัมผัสนั้น จะมีค่ากว้างกว่าคือ จะมีค่าเป็นฐาน 16 น้อยกว่า F000 เสมอ จึงสามารถแบ่งกรณีจากช่วงของค่าที่ตรวจสอบรับเข้ามา

ปัญหาเรื่องกระดาษสี ซึ่งสีที่ได้มาทำการทดลองนั้นสีเดียวกันแต่มีความเข้มของสีไม่เหมือนกันซึ่งอาจจะเกิดจากกระบวนการผลิตของกระดาษ ทำให้กระบวนการตรวจสอบค่าจากเซ็นเซอร์แสง ไม่เป็นไปตามที่กำหนดไว้ ทำให้ในการซื้อกระดาษสีในแต่ละครั้งต้องมีการนำเซ็นเซอร์แสงไปวัดค่าด้วยทุกครั้งแม้ว่าจะซื้อสีเดียวกันก็ตาม

5.2.2 ปัญหาทางด้านซอฟท์แวร์

ทุกครั้งที่มีการถอดแบตเตอรี่ออกจากตัวอาร์ซีเอ็กซ์ (RCX) หรือหยุดการทำงานเป็นระยะเวลาหนึ่ง จะทำให้เฟิร์มแวร์บริคโอเอส (BrickOS) นั้นหายไป ทำให้ต้องมีการโหลดเฟิร์มแวร์บริคโอเอสลงหุ่นยนต์ใหม่ทุกครั้ง รวมไปถึงการโหลดโปรแกรมลงหุ่นยนต์นั้นจำเป็นจะต้องอาศัยเฟิร์มแวร์บริคโอเอสเท่านั้น ดังนั้นถ้าเกิดการข้ามขั้นตอน หรือตัวข้อมูลซอฟต์แวร์ภายในถูกลบ หรือหายไป จะทำให้ไม่สามารถลงโปรแกรมได้ ทุกครั้งที่มีการเริ่มทำงานจำเป็นจะต้องเช็คเฟิร์มแวร์บริคโอเอสภายในตัวหุ่นยนต์ก่อนเสมอ ถ้าตัวเฟิร์มแวร์บริคโอเอสยังไม่ได้ลงใหม่ หายไป หรือถ้ามีปัญหาอื่นๆ ต้องทำการลงใหม่อีกครั้ง

5.2.3 ปัญหาด้านการเขียนโปรแกรม

การเขียนโปรแกรมกับหุ่นยนต์อาร์ซีเอ็กซ์นั้น เป็นการเขียนโดยใช้ภาษา C++ แต่ในการมาประยุกต์กับกระบวนการพัฒนาแบบ OOP (Object-Oriented-Programming) นั้น รูปแบบการเขียน Abstract ที่จะให้มีการสืบทอดทำได้บากเนื่องจากส่วนประกอบของระบบ หรือสิ่งตอบรับที่มีต่อระบบจะเป็นรูปธรรม จึงทำให้มีจำนวนน้อยไม่สามารถมีการสืบทอดได้ เพราะจะเป็นการสืบทอดที่ไร้ประโยชน์ และในการเขียนโปรแกรมจำเป็นจะต้องมีการไล่โปรแกรมสำหรับการเรียกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามลำดับบนลงล่าง ที่ไม่เหมือนกับการใช้จาวา เมธอดไหน หรือคลาสใดที่มีการถูกเรียกใช้ จะต้องมีการรันผ่านจากบริเวณข้างบนมาก่อน หรือตัวโค๊ดที่ถูกเรียกจำเป็นจะต้องอยู่บริเวณด้านบนของโค๊ดที่ได้ทำการเรียกใช้ ไม่เช่นนั้นจะคอมไพล์ไม่ผ่าน

ด้านหุ่นยนต์เมื่อได้ทำการเริ่มรันโปรแกรมนั้นจะมีการทำงานที่อยู่นอกเหนือจากตัวโค๊ดที่เราได้กำหนดเข้าไป คือส่วนของเซ็นเซอร์แสงจะมีการไล่ค่าแสงเข้ามาก่อน 1 รอบก่อนเริ่มทำงาน ดังนั้นการกำหนดตัวแปรต่างๆ รวมไปคือข้อกำหนดในโปรแกรมต้องมีการคิดครอบคลุมเผื่อด้วย

5.3 ข้อเสนอแนะ

ในการทำงานต่างๆในกระบวนการที่มีฮาร์ดแวร์เข้ามาร่วมด้วย ควรจะมีการทดสอบจำนวนมาก เพื่อให้รับทราบถึงข้อผิดพลาดต่างๆที่อาจจะเกิดขึ้นได้ ที่เป็นความผิดพลาดที่ไม่ได้เกิดจากผู้พัฒนา แต่เป็นข้อผิดพลาดที่เกิดจากสิ่งแวดล้อม และส่วนต่างๆที่นอกเหนือจากความรู้ของผู้พัฒนา จากการทดสอบต่างๆที่ได้กำหนดไว้ ทำให้ทราบถึงข้อผิดพลาดต่างๆ และการใช้งานที่เหมาะสมของฟังก์ชั่นการทำงานที่มีอยู่ในตัวหุ่นยนต์อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. **คัมภีร์ การวิเคราะห์และออกแบบระบบเชิงวัตถุด้วย UML.** 1. กรุงเทพฯ: เคทีพี

กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548. **คัมภีร์การพัมนาระบบเชิงวัตถุด้วย UML และ Java.**กรุงเทพฯ : เคทีพี

พนิดา พานิชกุล. 2548. **Object-Oriented ฉบับพื้นฐาน**. กรุงเทพฯ : เคทีพี

appsofttech.com. **Robotics kits**. [online].Available : http://www.appsofttech.com/robot/products/Team%20CHallenge%20Set.shtml

appsofttech.com. **The RCX**. [online].Available : http://www.appsofttech.com/robot/mindstorms/index.shtml

blogninja.com. **brickOS C++ Documentation v0.9.0.** [online].Available : http://blogninja.com/doc/brickos/html/html-c++/main.html

brickos.sourceforge.net. **brickOS 0.2.6.10 Instructions for installing on Win9x/NT/2000/XP** [online].Available : http://brickos.sourceforge.net/docs/INSTALL-cygwin.html

brickos.sourceforge.net. **brickOS C++ Documentation v0.2.6.10.6.** [online].Available : http://brickos.sourceforge.net/docs/APIs/html-c++/

cygwin.com. NameOfWeb. **GNU + Cygnus + Windows = cygwin.** [online].Available : http://www.cygwin.com/

David C. Chen. **BrickOS Command Reference v2.0-0.2.6.10**. [online].Available : http://brickos.sourceforge.net/docs/CommandRef.html

David C. Chen. **LegOS Command Reference 1.0 + epsilon**. [online].Available : http://www.it.uu.se/edu/course/homepage/realtid/H04/ass2/legos_command_reference.html

Japan System House Association. 2549. **เทคโนโลยีสมองกลฝังตัว**. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง. 1. กรุงเทพฯ:สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

Ole Caprani. **RCX Manual**. [online].Available : http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม(ต่อ)

Russel Nelson. **LEGO® MINDSTORMS(tm) Internals**. [online].Available : http://www.crynwr.com/lego-robotics/

Todd S. Lehman and Suzanne D. Rich. **Welcome to the brickOS home page!** [online].Available : http://brickos.sourceforge.net/

www.cs.brown.edu. **Mechanical control of the RCX**. [online].Available : http://www.cs.brown.edu/courses/cs148/old/2004fall/brickOS/HOWTO/control.html

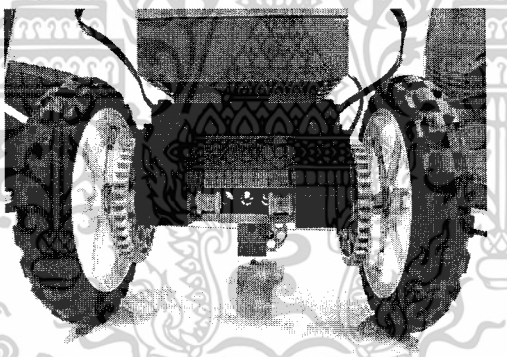
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
การทดสอบอุปกรณ์

ในการทดสอบอุปกรณ์นั้น มีจุดประสงค์เพื่อทดลองการทำงานของแต่ละอุปกรณ์ และหาค่าที่ได้จากอุปกรณ์แต่ละชนิด การทดลองนั้นได้อ้างอิงกับโปรแกรมตรวจจับเส้นทางที่เขียนขึ้นเพื่อทดสอบอุปกรณ์เหล่านี้ซึ่งมีการทดลองใหญ่ๆ อยู่ 3 แบบ คือ

1.1 การทดสอบมอเตอร์ขับเคลื่อน

จุดประสงค์ในการทดสอบมอเตอร์ขับเคลื่อนทั้งสองฝั่งนั้นดังรูปที่ ก.1 เพื่อตรวจสอบดูว่าในการใช้คำสั่งโปรแกรมนั้น คำสั่ง " fwd " , " rev " , " brake " และ " off " มอเตอร์จะหมุนไปทางไหน และ หุ่นยนต์จะมีการเคลื่อนที่อย่างไร และ ทดลองความเร็วของมอเตอร์จากการที่ได้โปรแกรมค่าเข้าไปตั้งแต่ 0-255



รูปที่ ก.1 มอเตอร์ขับเคลื่อน

การทดลองโปรแกรมตรวจจับเส้นทางการใช้คำสั่งทั้ง 6 รูปแบบจะได้ผลดังตารางที่ ก.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 แสดงความสัมพันธ์ระหว่างคำสั่งที่โปรแกรมลงไปในมอเตอร์ขับเคลื่อนกับผลที่ได้รับ

มอเตอร์ฝั่งซ้าย	มอเตอร์ฝั่งขวา	ผลที่ได้รับ
“fwd”	“fwd”	หมุนไปทางขวา
“fwd”	“rev”	เคลื่อนที่ไปข้างหน้า
“rev”	“fwd”	เคลื่อนที่ไปข้างหลัง
“rev”	“rev”	หมุนไปทางซ้าย
“brake”	“brake”	หยุดการเคลื่อนที่
“off”	“off”	เคลื่อนที่อิสระ

การทดลองความเร็วของมอเตอร์ในโดยการใส่ค่าความเร็วเข้าไปในโปรแกรม จะได้ผลดังตารางที่ ก.2

ตารางที่ ก.2 แสดงความสัมพันธ์ระหว่างค่าของความเร็วที่ได้โปรแกรมเข้าไปกับผลที่ได้รับ

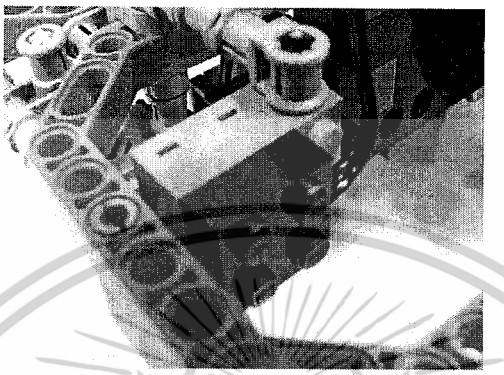
ค่าเร็วที่กำหนด	ผลที่ได้
“0”	มอเตอร์ไม่มีการทำงาน
“1”	มอเตอร์นั้นหมุนด้วยความเร็วที่ช้าที่สุด
“128”	มอเตอร์หมุนด้วยความเร็วปานกลาง
“255”	มอเตอร์หมุนด้วยความเร็วที่เร็วที่สุด

ในการปรับความเร็วของมอเตอร์นั้นต้องทำให้สัมพันธ์กับน้ำหนักของหุ่นยนต์ด้วย เนื่องจากหุ่นยนต์นั้นก็มีน้ำหนักที่มากทำให้การใช้ความเร็วที่ต่ำมากจะส่งผลทำให้หุ่นยนต์ไม่สามารถเคลื่อนที่ได้ ซึ่งจะทำให้มอเตอร์นั้นฝืนกำลัง อาจจะทำให้มอเตอร์ได้รับความเสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 การทดสอบเซ็นเซอร์สัมผัส

จุดประสงค์ของการทดสอบเซ็นเซอร์สัมผัสดังรูปที่ ก.2 คือ การดูค่าที่เซ็นเซอร์สัมผัสได้ส่งมายังหุ่นยนต์ใน สถานะที่มันถูกทำให้แอคทีฟ (ถูกกด) และไม่แอคทีฟ (ไม่ถูกกด)



รูปที่ ก.2 เซ็นเซอร์สัมผัส

การทดสอบเบื้องต้นจะได้ว่า ค่าที่ได้รับจากเซ็นเซอร์ของทั้ง พอร์ต1 และ 3 นั้นมีค่าประจำตัวของแต่ละพอร์ต ซึ่งค่าที่ได้รับแสดงดังตารางที่ ก.3

ตารางที่ ก.3 แสดงความสัมพันธ์ระหว่างค่าที่ได้กับพอร์ตที่เซ็นเซอร์ติดต่อ

กรณีการสัมผัส	ค่าจากเซ็นเซอร์สัมผัสฝั่งซ้าย (Hex)	ค่าจากเซ็นเซอร์สัมผัสฝั่งขวา (Hex)
ไม่มีการสัมผัส	FF00 FF40 FF80 FFC0 ... ค่าที่ได้ > F000 ทั้งหมด	FF00 FEC0 FDC0 FD80 ... ค่าที่ได้ > F000 ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 แสดงความสัมพันธ์ระหว่างค่าที่ได้กับพอร์ตที่เซ็นเซอร์ติดต่อตารางที่ (ต่อ)

กรณีการสัมผัส	ค่าจากเซ็นเซอร์สัมผัสฝั่งซ้าย (Hex)	ค่าจากเซ็นเซอร์สัมผัสฝั่งขวา (Hex)
มีการสัมผัส เซ็นเซอร์ฝั่งซ้าย	0840 0880 0940 0900 ... ค่าที่ได้ < F000 ทั้งหมด	FE80 FD80 FDC0 FF80 ... ค่าที่ได้ > F000 ทั้งหมด
มีการสัมผัส เซ็นเซอร์ฝั่งขวา	FE80 FD80 FDC0 FF80 ... ค่าที่ได้ > F000 ทั้งหมด	0940 21C0 1540 17C0 ... ค่าที่ได้ < F000 ทั้งหมด
มีการสัมผัส เซ็นเซอร์ทั้งสองฝั่ง	0940 44F0 7740 84C0 ... ค่าที่ได้ < F000 ทั้งหมด	08C0 0F00 17C0 A1C0 ... ค่าที่ได้ < F000 ทั้งหมด

การทดลองตามโปรแกรมตรวจจับเส้นทาง จะได้ผลดังตารางที่ ก.4

ตารางที่ ก.4 แสดงความสัมพันธ์ระหว่างพอร์ตเซ็นเซอร์กับผลที่ได้รับจากการที่เซ็นเซอร์ถูก/ไม่ถูกแอคทีฟ

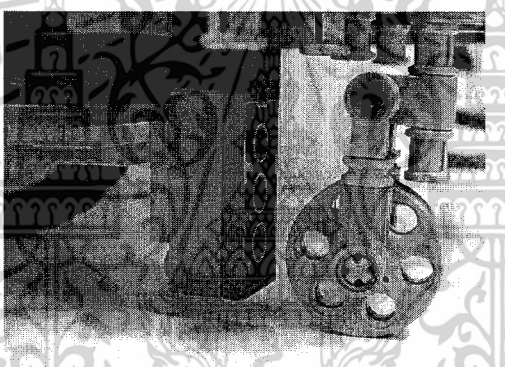
เซ็นเซอร์ที่ต่อกับพอร์ต	รูปแบบการตอบสนองที่ตั้งไว้	รูปแบบการตอบสนองที่ได้รับ
1	เคลื่อนที่ไปข้างหน้า	เคลื่อนที่ไปข้างหน้า
3	เคลื่อนที่ไปข้างหลัง	เคลื่อนที่ไปข้างหลัง
1 ,3	หมุนซ้ายแล้วหมุนขวา	หมุนซ้ายแล้วหมุนขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบเซ็นเซอร์สัมผัสนั้นค่าที่ได้รับจากเซ็นเซอร์สัมผัสเมื่อไม่มีการแอคทีฟ (ไม่ถูกกด) นั้นมีค่าไม่คงที่ขึ้นอยู่กับพอร์ตที่เชื่อมต่อ แต่ค่าที่ได้จะมากกว่าเลขฐานสิบหก F000 เสมอ ซึ่งแต่ละพอร์ตจะมีค่าไม่เหมือนกัน และเมื่อเซ็นเซอร์สัมผัสแอคทีฟ (ถูกกด) ค่าที่ได้นั้นมีค่าไม่นิ่ง (ค่าที่ได้เป็นคนละค่ากับตอนที่ไม่ถูกกด) เป็นอย่างมากจนไม่สามารถบันทึกการทดลองเป็นค่าที่แน่นอนได้ แต่ค่าที่ได้จะน้อยกว่าเลขฐานสิบหก F000 เสมอ

1.3 การทดสอบเซ็นเซอร์แสง

จุดประสงค์ของการทดสอบเซ็นเซอร์แสงดังรูปที่ ก.3 คือ เพื่อดูค่าที่เซ็นเซอร์แสงอ่านได้ ที่ความเข้มสีของสีและวัสดุต่างๆ เพื่อนำมาใช้ในการแยกแยะสี ส่งผลทำให้เกิดการตอบสนองของหุ่นยนต์ที่แตกต่างกันในแต่ละความเข้มของสี



รูปที่ ก.3 เซ็นเซอร์แสง

การทดสอบเบื้องต้นจะได้ว่าค่าของความเข้มของแสงที่อาร์ซีเอ็กซ์ได้แสดงผลออกมาทางหน้าจอจะเป็นเลขฐาน 16 ซึ่งในการทดลองนั้นได้ทำการทดลองกับสีทั้งหมด 7 สี ซึ่งเป็นสีที่ใช้ในสนามทดลอง ผลการทดลองที่ได้เป็นดังตารางที่ ก.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.5 แสดง ค่าที่ได้อ่านได้จากเซ็นเซอร์แสงตามสีที่ได้ทดลอง

ลำดับ	สีและชนิด	ค่าที่อาร์ซีเอ็กซ์อ่านได้ (Hex)	ค่าที่แปลงเป็นเลขฐาน 10 (Dec)
1	สีดำ(ขอบสนาม)	0020-0021	32-33
2	สีเขียวเข้ม(เส้นแบ่งเขต)	0024-0025	36-37
3	สีน้ำเงิน(พื้นสนาม)	0027-0028	39-40
4	สีเขียวอ่อน(ทางขึ้น-ลง)	0029-002A	41-42
5	สีแดงอ่อน(เส้นแบ่งเขต)	002D-002E	45-46
6	สีแดงเข้ม(จุดสิ้นสุด)	0030-0031	48-49

การทดลองตามโปรแกรมตรวจจับเส้นทาง จะได้ผลดังตารางที่ ก.6

ตารางที่ ก.6 แสดงความสัมพันธ์ระหว่างสีที่ใช้ในการทดลองโปรแกรมหุ่นยนต์ตรวจจับเส้นทางกับผลที่ได้

สีที่ทดสอบ	รูปแบบการตอบสนองที่ตั้งไว้	รูปแบบการตอบสนองที่ได้รับ
สีขาวฟิวเจอร์บอร์ด	เคลื่อนที่ไปข้างหน้า	เคลื่อนที่ไปข้างหน้า
สีน้ำเงิน(พื้นสนาม)	เคลื่อนที่ไปข้างหลัง	เคลื่อนที่ไปข้างหลัง
สีแดงเข้ม(จุดสิ้นสุด)	หมุนซ้ายแล้วหมุนขวา	หมุนซ้ายแล้วหมุนขวา
สีแดงอ่อน(เส้นแบ่งเขต)	หมุนขวาแล้วหมุนซ้าย	หมุนขวาแล้วหมุนซ้าย
สีเขียวเข้ม(เส้นแบ่งเขต)	หมุนซ้ายแล้วเคลื่อนที่ไปข้างหน้า	หมุนซ้ายแล้วเคลื่อนที่ไปข้างหน้า
สีเขียวอ่อน(ทางขึ้น-ลง)	หมุนขวาแล้วเคลื่อนที่ไปข้างหน้า	หมุนขวาแล้วเคลื่อนที่ไปข้างหน้า
สีดำ(ขอบสนาม)	หมุนซ้ายแล้วเคลื่อนที่ไปข้างหลัง	หมุนซ้ายแล้วเคลื่อนที่ไปข้างหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่เซ็นเซอร์แสงอ่านได้ในแต่ละความเข้มสีนั้นไม่นิ่ง แต่จะอยู่ในช่วงที่ได้ทำการทดสอบจากอาร์ซีเอ็กซ์ไว้ ดังตารางที่ ก.5 เนื่องจาก ไม่สามารถกำหนดสภาวะแวดล้อมได้คงที่เช่น แสงไฟรบกวน เงาของตัวหุ่นยนต์ และ ระนาบของเซ็นเซอร์ที่ไม่ได้อยู่ในแนวระนาบอย่างต่อเนื่อง ทำให้การโปรแกรมทดสอบเซ็นเซอร์แสงนั้นต้องมีการหน่วงค่าที่ได้รับเข้ามาเพื่อที่จะทำให้ค่านั้นคงที่พอที่จะสามารถนำไปใช้ได้ หรือ มีการสุ่มชักตัวอย่างขึ้นมาจากการเฉลี่ยค่าที่อ่านมาได้ทั้งหมดจากความเข้มสีนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
ขั้นตอนการติดตั้งบริคโอเอสบนวินโดว์เอ็กซ์พี

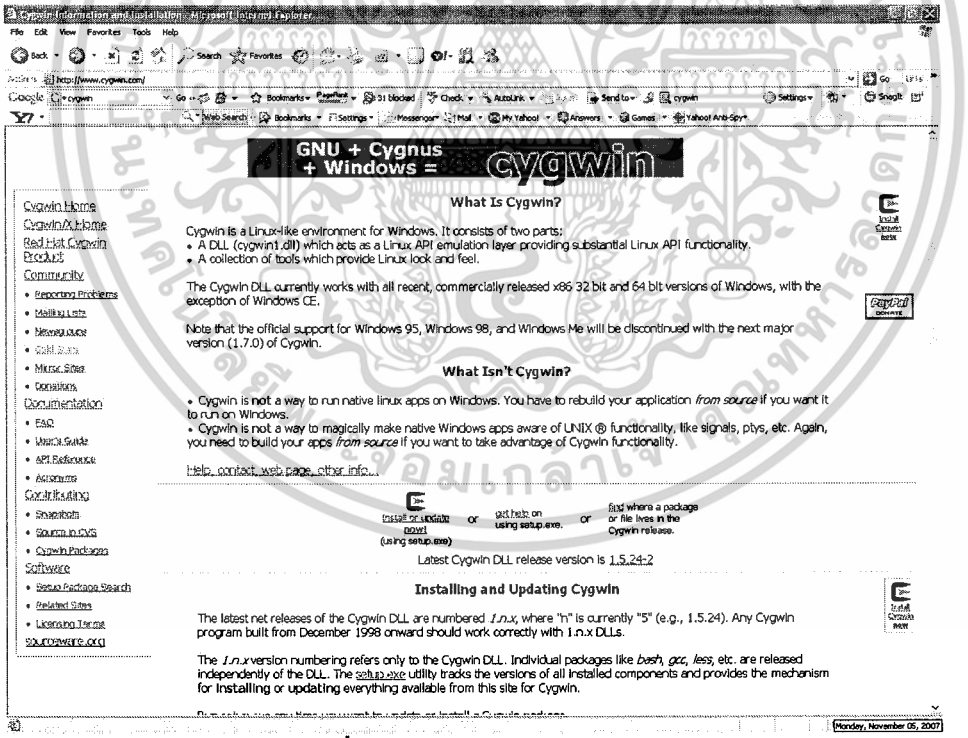
1. ติดตั้งโปรแกรมซิกวิน (Cygwin)

โปรแกรมซิกวิน(Cygwin) เป็นโปรแกรมที่มีสภาพแวดล้อมการทำงานเหมือนลีนุกซ์ (Linux) แต่ใช้สำหรับระบบปฏิบัติการวินโดว์ (Windows) โดยมีส่วนประกอบที่สำคัญ 2 ส่วนคือ

- DLL ซึ่งทำหน้าที่เป็นตัวจำลอง (emulator) ฟังก์ชั่นเอพีไอ (API) ของลีนุกซ์ต่างๆ ให้สามารถใช้งานได้ภายใต้สภาพแวดล้อมของระบบปฏิบัติการวินโดว์
- ส่วนของเครื่องมือต่าง ๆ ที่สนับสนุนการทำงานภายใต้รูปแบบของลีนุกซ์บนวินโดว์

การติดตั้งโปรแกรมซิกวิน(Cygwin) ให้ทำตามขั้นตอนต่อไปนี้

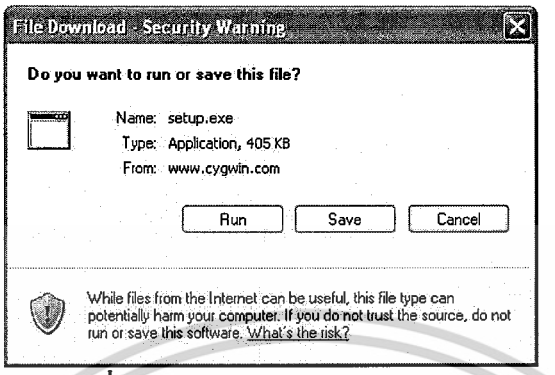
1.1 เข้าไปที่เว็ปไซต์ซิกวิน(Cygwin) (http://www.cygwin.com) เพื่อทำการดาวโหลดโปรแกรมมาติดตั้ง ดังรูปที่ ข.1



รูปที่ ข.1 หน้าเวบไซต์ของซิกวิน

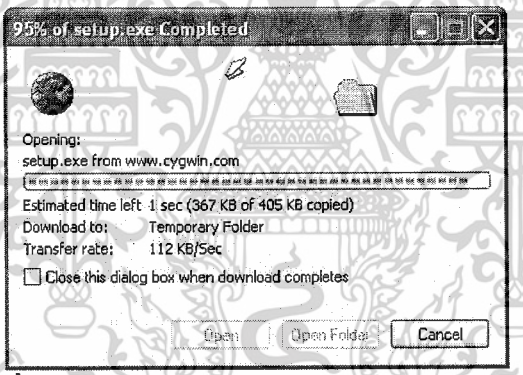
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ให้คลิ๊กที่ไอคอน “ Install or update now!” เพื่อทำการรันโปรแกรมเซ็ตอัพ (Setup) โดยให้คลิ๊กปุ่ม “Run” ดังรูปที่ ข.2



รูปที่ ข.2 หน้าต่างของโปรแกรมเซ็ตอัพ

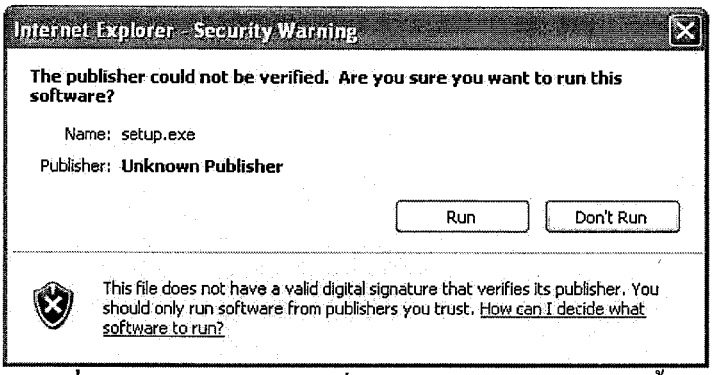
1.3 เครื่องจะทำการดาวโหลดไฟล์ที่จำเป็นสำหรับโปรแกรม “setup.exe” มาเก็บไว้บนเครื่อง ดังรูปที่ ข.3



รูปที่ ข.3 หน้าต่างการดาวโหลดไฟล์สำหรับโปรแกรม

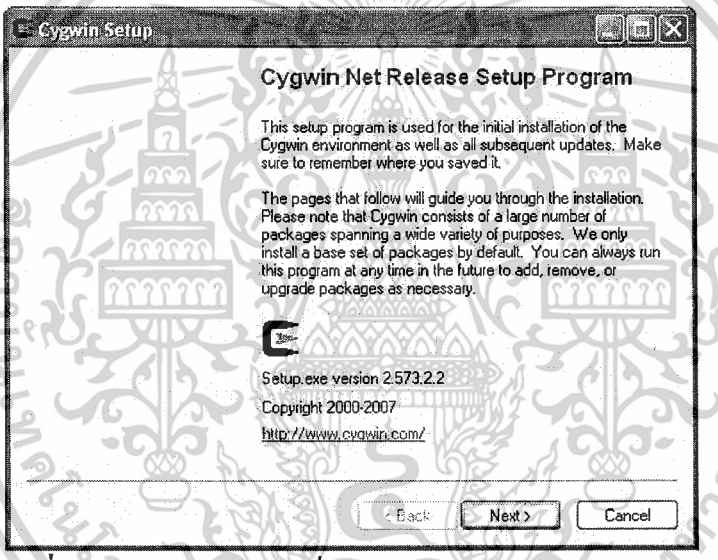
1.4 เมื่อดาวโหลดเสร็จเรียบร้อย ให้ทำการคลิ๊กปุ่ม “Run” เพื่อทำการรันโปรแกรมดังรูปที่ ข.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 หน้าต่างโปรแกรมเมื่อทำการดาวโหลดไฟล์เสร็จสิ้น

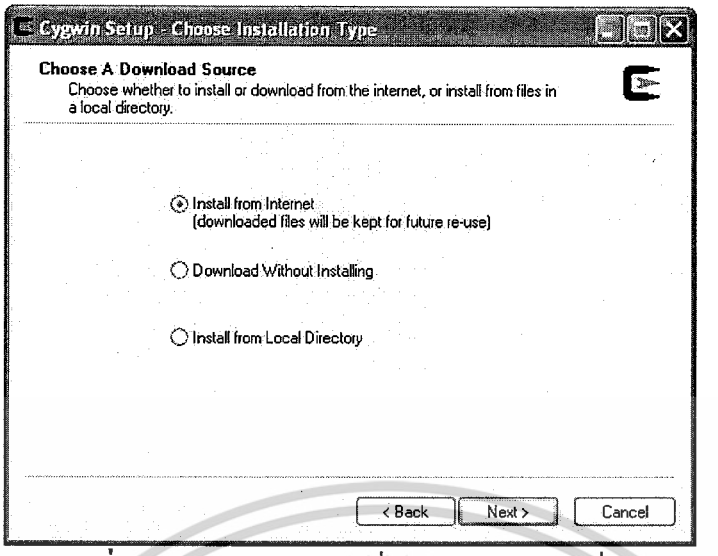
1.5 จากนั้นคลิ๊กปุ่ม "Next" เพื่อให้โปรแกรมเซ็ตอัพ (Setup) ทำการติดตั้งโปรแกรมซิกวิน (Cygwin) รวมทั้งสภาพแวดล้อมในการใช้งาน ดังรูปที่ ข.5



รูปที่ ข.5 หน้าต่างโปรแกรมเมื่อต้องการทำการลงโปรแกรมซิกวิน

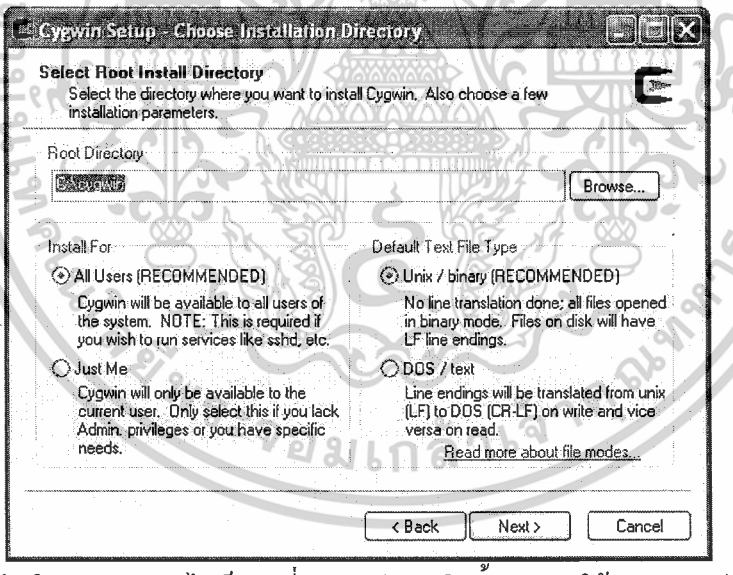
1.6 จากนั้นให้เลือกรูปแบบการติดตั้งไปที่ "Install from Internet" แล้วคลิ๊กปุ่ม "Next" ดังรูปที่ ข.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 หน้าต่างโปรแกรมเพื่อทำการเลือกแหล่งที่มา

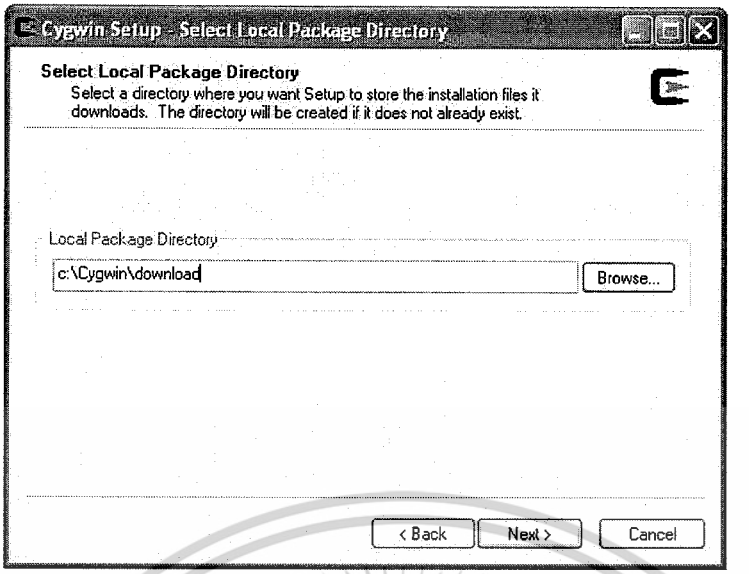
1.7 เสร็จแล้วกำหนดไดเร็กทอรี่หลัก (Root directory) ที่จะทำการติดตั้งโปรแกรม โดยเลือกรูปแบบการติดตั้งสำหรับใช้งาน (Install for) และรูปแบบของตัวอักษร (Default Text File Type) ดังรูป จากนั้นคลิ๊กปุ่ม "Next" ดังรูปที่ ข.7



รูปที่ ข.7 หน้าต่างในการกำหนดไดเร็กทอรี่หลัก รูปแบบติดตั้งสำหรับใช้งานและรูปแบบตัวอักษร

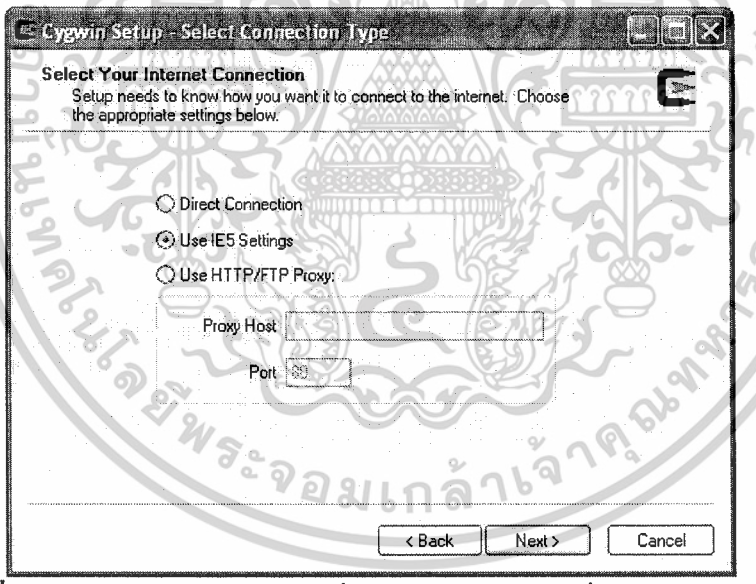
1.8 ทำการกำหนดไดเร็กทอรี่ที่จะทำการเก็บไฟล์ที่เกี่ยวข้องในการติดตั้ง (ซึ่งจะถูกดาวโหลดมาเก็บบนเครื่องดังรูป) จากนั้นคลิ๊ก "Next" ดังรูปที่ ข.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.8 หน้าต่างกำหนดไดเร็กทอรี่ที่จะทำการเก็บไฟล์ที่เกี่ยวข้องในการติดตั้ง

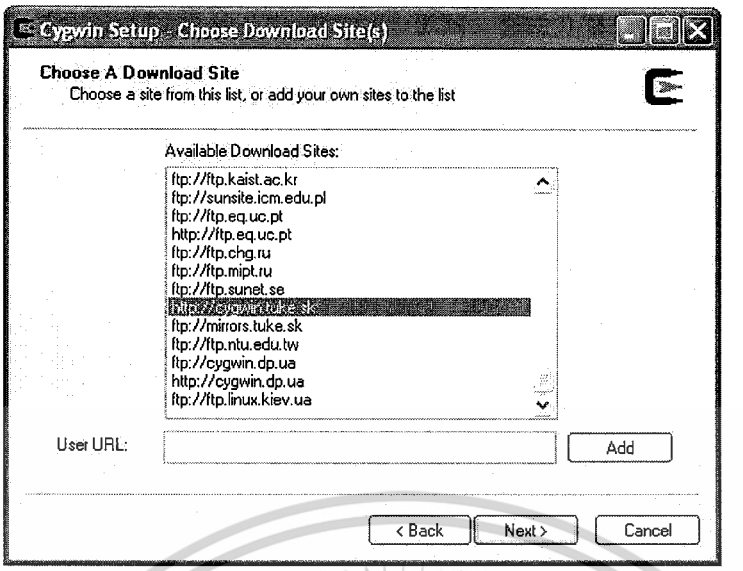
1.9 เลือกรูปแบบการเชื่อมต่อกับอินเตอร์เน็ทที่จะใช้ในการติดตั้ง แล้วคลิ๊กปุ่ม "Next" ดังรูปที่ ข.9



รูปที่ ข.9 หน้าต่างเลือกรูปแบบการเชื่อมต่อกับอินเตอร์เน็ทที่จะใช้ในการติดตั้ง

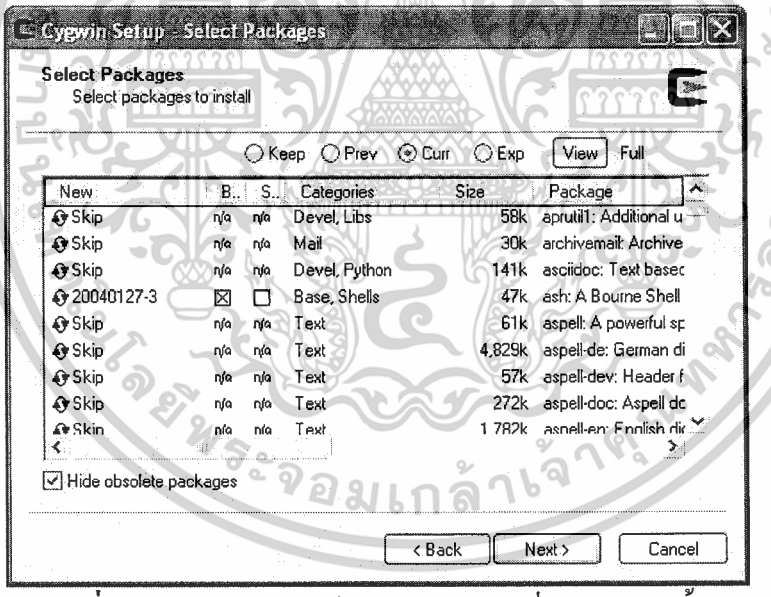
1.10 ทำการเลือกเวบไซต์ (Mirror sites) ที่จะทำการดาวโหลดโปรแกรม แล้วคลิ๊กปุ่ม "Next" ดังรูปที่ ข.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.10 หน้าต่างเลือกเวบไซต์ (Mirror sites) ที่จะทำการดาวโหลดโปรแกรม

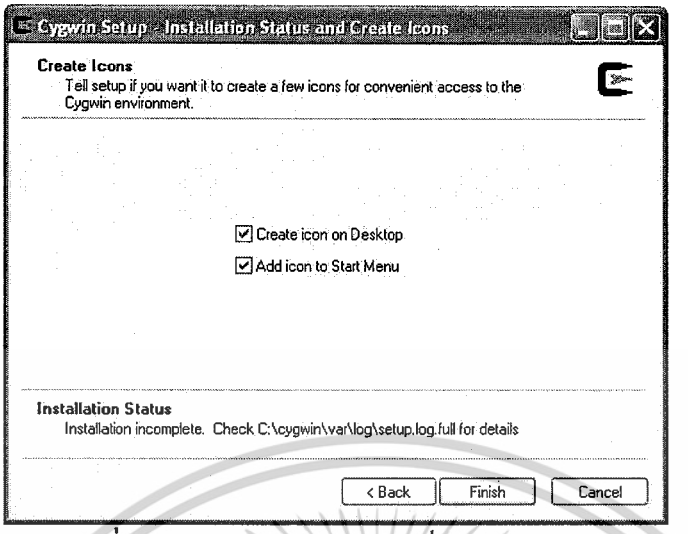
1.11 เครื่องจะทำการดาวโหลดโปรแกรมจนเสร็จ จากนั้นจะปรากฏหน้าต่างให้เลือกแพ็คเกจ (Packages) ที่จะทำการติดตั้ง ดังรูปที่ ข.11 จากนั้นให้คลิ๊กปุ่ม "Next"



รูปที่ ข.11 หน้าต่างเลือกแพ็คเกจ (Packages) ที่จะทำการติดตั้ง

1.12 เครื่องจะทำการดาวโหลดแพ็คเกจที่เลือกมาทำการติดตั้งจนเสร็จ (ประมาณ 2-3 นาที ขึ้นกับความเร็ว) จากนั้นให้เลือกว่าจะสร้างไอคอนบนเดสก์ท็อป (Desktop) หรือในเมนูเริ่มต้น (Start menu) แล้วคลิ๊กปุ่ม "Finish" เพื่อออกจากการติดตั้ง ดังรูปที่ ข.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



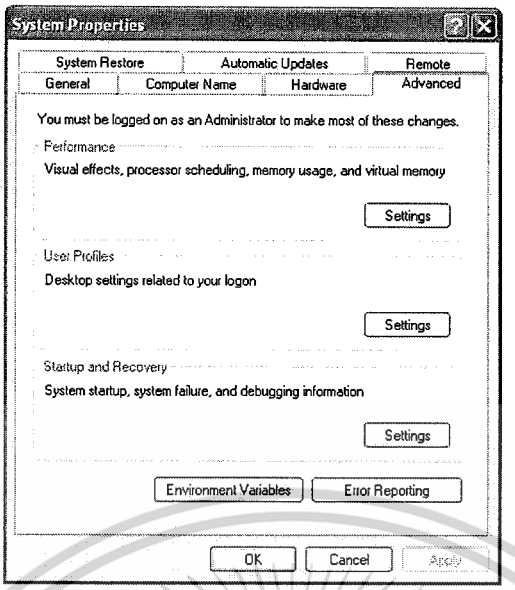
รูปที่ ข.12 หน้าต่างให้เลือกแหล่งที่จะสร้างไอคอน

หากการติดตั้งสมบูรณ์ จะปรากฏหน้าต่างการติดตั้งเสร็จสิ้นสมบูรณ์ (Install completed) ขึ้นให้คลิ๊กปุ่ม "Ok"

1.13 ให้ตรวจสอบไดเร็กทอรี่ "c:\cygwin\bin" และ "c:\cygwin\var" ว่ามีการจัดสร้างและมีไฟล์อยู่หรือไม่ (หากการติดตั้งไม่สมบูรณ์ จะไม่มีไฟล์ใดๆ ในไดเร็กทอรี่ดังกล่าว)

1.14 ทำการกำหนดตัวแปร (variable) และเส้นทาง (path) ของโปรแกรมซิกวิน(cygwin) ให้กับระบบปฏิบัติการวินโดว์ โดยเลือกที่ไอคอน "My computer" แล้วคลิ๊กเม้าท์ปุ่มขวา เลือก "Properties" จะปรากฏหน้าต่าง "System Properties" ดังรูปที่ ข.13

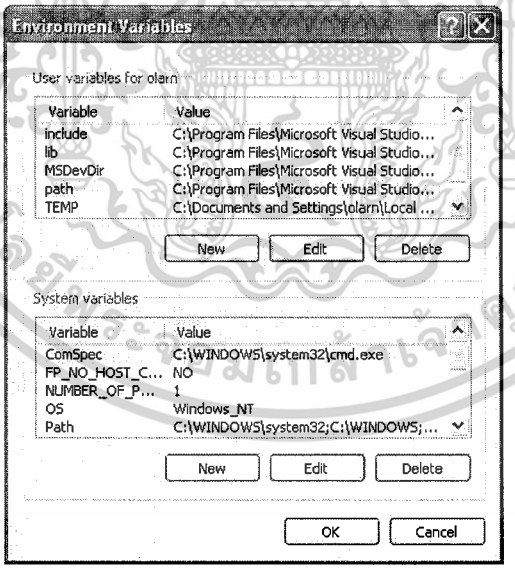
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.13 หน้าต่างคุณลักษณะระบบ (System Properties)

จากนั้น เลือกไปที่แทป (Tab) Advanced แล้วคลิ๊กปุ่มตัวแปรสภาพแวดล้อม (environment variables)

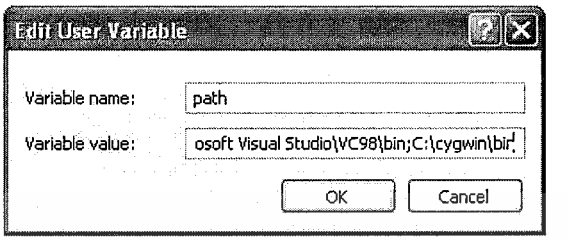
1.15 ทำการเลือกตัวแปร (Variable) เส้นทาง (Path) ในช่องตัวแปรสำหรับผู้ใช้ (User Variables) แล้วทำการคลิ๊กปุ่ม “Edit” เพื่อทำการเพิ่มค่าตัวแปร (หากยังไม่มีตัวแปร “Path” ให้คลิ๊กที่ปุ่ม “New” เพื่อสร้างขึ้นใหม่) ดังรูปที่ ข.14



รูปที่ ข.14 หน้าต่างตัวแปรสภาพแวดล้อม (environment variables)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.16 ทำการเพิ่มค่าตัวแปร (Variable value) ภายใต้ตัวแปรชื่อ Path โดยพิมพ์ค่า ";c:\cygwin\bin" ต่อท้ายในช่องค่าตัวแปร (Variable value) ดังรูปที่ ข.15 แล้วคลิ๊ก "OK"



รูปที่ ข.15 หน้าต่างจัดการตัวแปรสำหรับผู้ใช้ (Edit User Variables) ตัวแปร "path"

หมายเหตุ หากเป็นการสร้างตัวแปรใหม่ (New) ให้พิมพ์ชื่อตัวแปร "Path" ลงไปในช่องชื่อตัวแปร (Variable name) และพิมพ์เฉพาะค่า "c:\cygwin\bin" ลงไปในช่องค่าตัวแปร (Variable value) แล้วคลิ๊ก "OK"

2. ติดตั้งโปรแกรมบริคโอเอส (BrickOS)

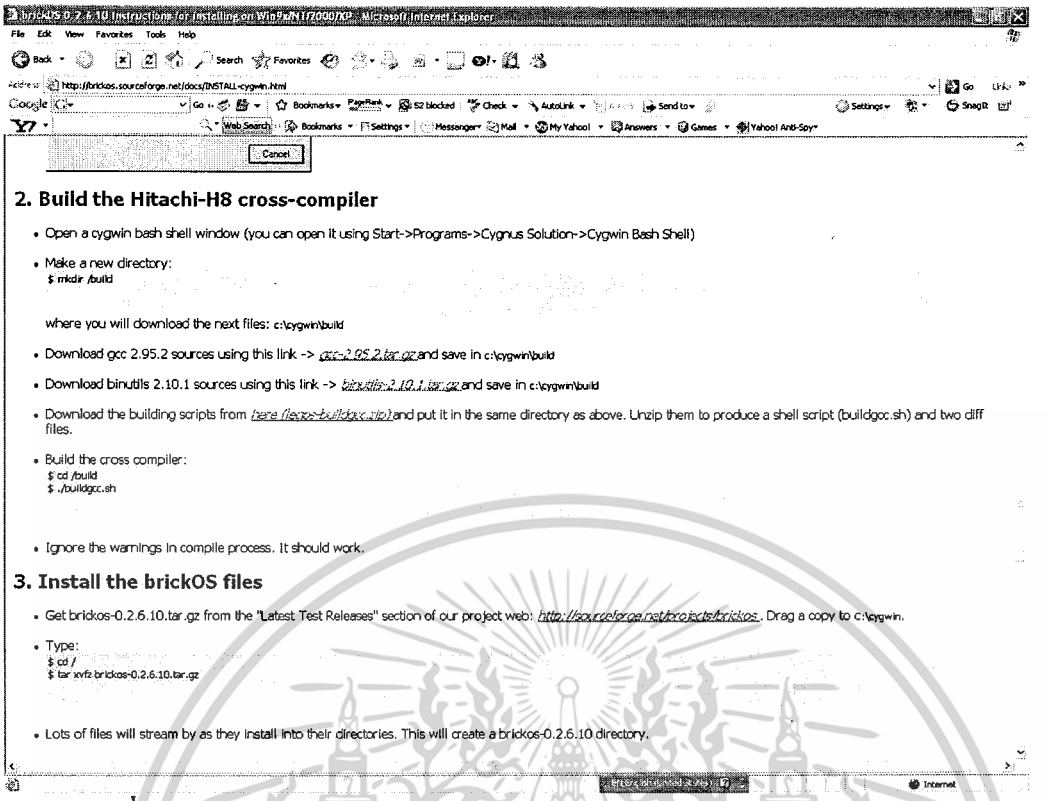
โปรแกรมบริคโอเอส (BrickOS) เป็นระบบปฏิบัติการฝังตัวแบบเปิดเผย (open source embedded OS) รวมทั้งกำหนดสภาพแวดล้อมการโปรแกรมสำหรับชุดอุปกรณ์เลโก้มายด์สตอร์มรุ่นอาร์ซีเอ็กซ์ (Lego Mindstorms Robotics Kits) ด้วยภาษา C และ C++ เพื่อให้ผู้ใช้งานสามารถโปรแกรมชุดอุปกรณ์เลโก้ ดังกล่าวได้ด้วยตนเอง (แทนที่จะใช้ภาษาโปรแกรมมาตรฐานของเลโก้)

โปรแกรมบริคโอเอส (BrickOS) ได้รับการพัฒนาขึ้นภายใต้สภาพแวดล้อมแบบลีนุกซ์ (Linux) แต่สามารถทำงานได้บนระบบปฏิบัติการวินโดว์ (โดยความช่วยเหลือจากโปรแกรมซิกวินข้างต้น) ในการใช้งานบริคโอเอส (BrickOS) จะต้องมีการติดตั้งคอมไพเลอร์ของฮิตาชิ (Hitachi H8 cross compiler assembler) และตัวลิงก์ (Linker) ที่เกี่ยวข้อง ดังนี้

1.17 เข้าไปที่เวบไซต์ http://birckos.sourceforge.net/docs/INSTALL-cygwin.html ดังรูปที่ ข.16 เพื่อเตรียมการดาวโหลดไฟล์ต่าง ๆ สำหรับการติดตั้งดังนี้

- gcc-2.95.2.tar.gz
- binutils-2.10.1.tar.gz
- legos-buildgcc.zip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2. Build the Hitachi-H8 cross-compiler

- Open a cygwin bash shell window (you can open it using Start->Programs->Cygnus Solution->Cygwin Bash Shell)
- Make a new directory:
  $ mkdir /build

  where you will download the next files: c:\cygwin\build
- Download gcc 2.95.2 sources using this link -> gcc-2.95.2.tar.gz and save in c:\cygwin\build
- Download binutils 2.10.1 sources using this link -> binutils-2.10.1.tar.gz and save in c:\cygwin\build
- Download the building scripts from here (legos-buildgcc.zip) and put it in the same directory as above. Unzip them to produce a shell script (buildgcc.sh) and two diff files.
- Build the cross compiler:
  $ cd /build
  $ ./buildgcc.sh

- Ignore the warnings in compile process. It should work.

3. Install the brickOS files

- Get brickos-0.2.6.10.tar.gz from the "Latest Test Releases" section of our project web: http://sourceforge.net/projects/brickos/. Drag a copy to c:\cygwin.
- Type:
  $ cd /
  $ tar xvfz brickos-0.2.6.10.tar.gz

- Lots of files will stream by as they install into their directories. This will create a brickos-0.2.6.10 directory.

รูปที่ ข.16 หน้าเวบไซต์ http://birckos.sourceforge.net/docs/INSTALL-cygwin.html

1.18 ทำการสร้างไดเร็กทอรี่ย่อย (sub directory) "\build" ภายใต้ "C:\cygwin" จากนั้นให้ทำการดาวโหลดไฟล์ต่าง ๆ ในข้อ 2.1 มาเก็บไว้

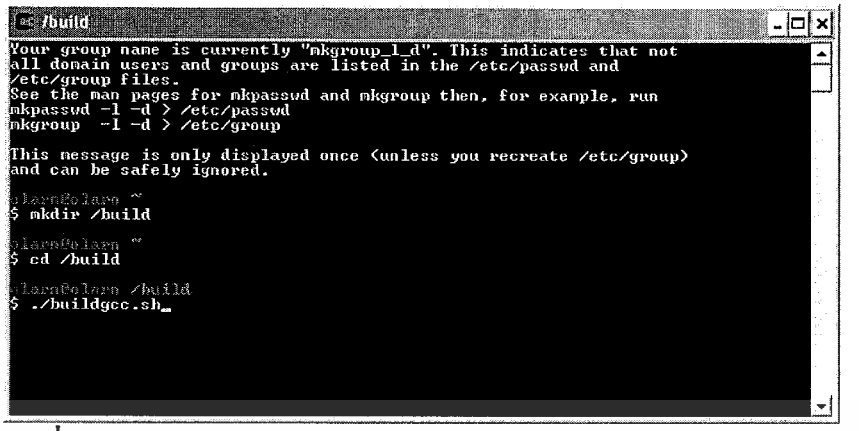
1.19 จากนั้นทำการแตกไฟล์ (unzip) ไฟล์ "legos-buildgcc.zip" เพื่อจะให้ได้ไฟล์ดังต่อไปนี้

- gcc-2.95.2-rcx-1.diff
- gcc-2.95.2-rcx-2.diff
- buildgcc.sh

1.20 รันโปรแกรมซิกวิน "cygwin" จากนั้นทำการรันโปรแกรม "buildgcc.sh" ดังรูปที่ ข.17 เพื่อสร้างสภาพแวดล้อมของตัวคอมไพเลอร์ (cross compiler environment) โดยพิมพ์คำสั่งต่าง ๆ ดังนี้

- cd /build
- ./buildgcc.sh

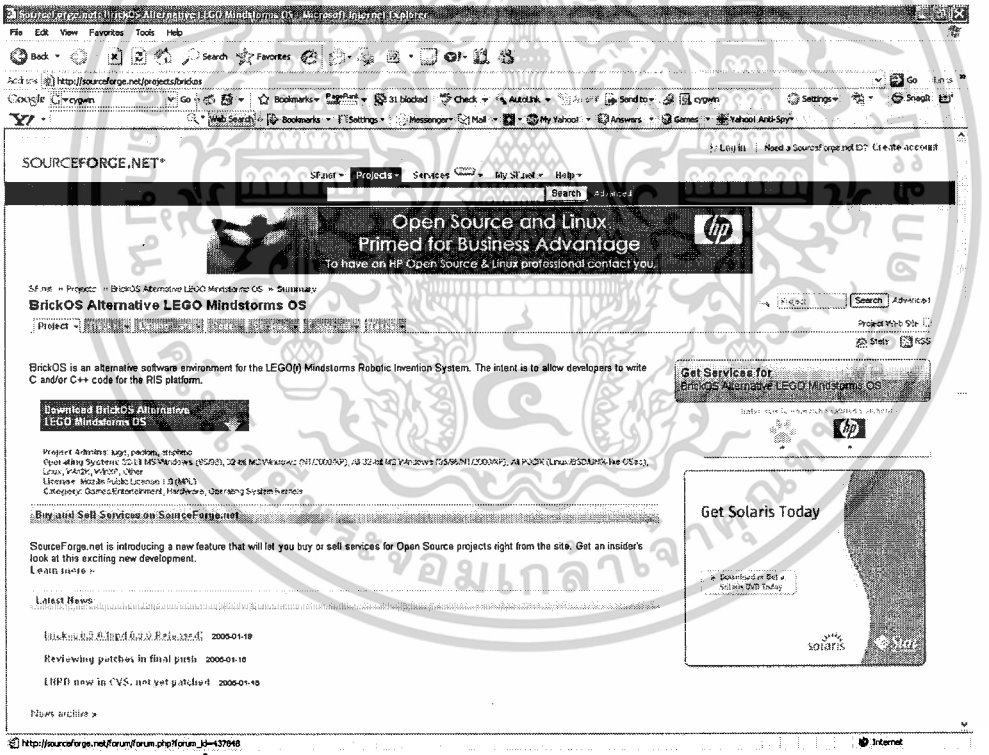
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.17 หน้าต่างของโปรแกรมซิกวินพร้อมการรันโปรแกรม “buildgcc.sh”

1.21 เข้าไปที่เว็ปไซต์ http://sourceforge.net/projects/birckos ดังรูป ข.18 เพื่อทำการดาวโหลดไฟล์ต่อไปนี้มาเก็บไว้ในไดเร็กทอรี่ “C:\cygwin”

- brickos-0.9.0.tar.gz



รูปที่ ข.18 หน้าเวบไซต์ http://sourceforge.net/projects/birckos

1.22 เสร็จแล้วกลับไปที่โปรแกรมซิกวิน (cygwin) ทำการแตกไฟล์ (unzip) ไฟล์ที่ดาวโหลดดังรูปที่ ข.19 มาจากขั้นตอน 2.5 โดยพิมพ์คำสั่งดังนี้

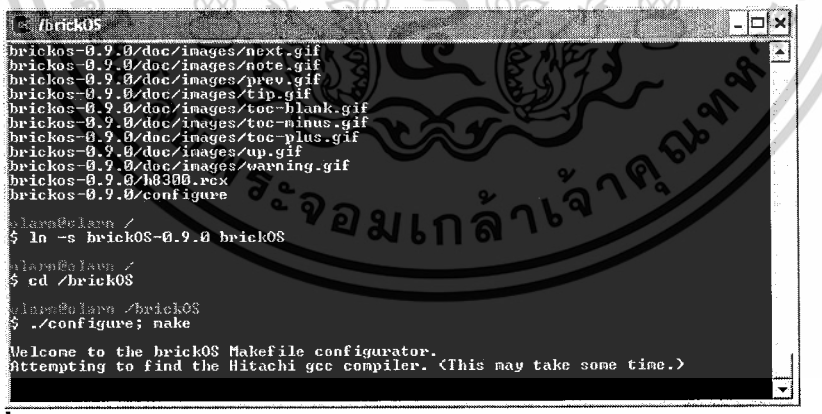
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- cd /
- tar xvfz brickos-0.9.0.tar.gz

รูปที่ ข.19 หน้าต่างโปรแกรมซิกวินพร้อมการแตกไฟล์ “brickos-0.9.0.tar.gz”

1.23 เมื่อทำการแตกไฟล์ (unzip) ไฟล์ในขั้นตอน 2.6 เสร็จเรียบร้อยแล้ว ให้ทำการกำหนดสภาพแวดล้อม (configure) สำหรับการคอมไพล์โปรแกรมเพื่อให้หุ่นยนต์เคลื่อนไหวได้ โดยพิมพ์คำสั่งต่อไปนี้ ดังรูปที่ ข.20

- ln –s brickOS-0.9.0 brickOS // สร้างลิ้งก์ให้กับ brickOS
- cd /birckOS
- ./configure // กำหนดค่าต่างสำหรับการคอมไพล์
- make // สร้างไฟล์สำหรับใช้คอมไพล์



รูปที่ ข.20 หน้าต่างโปรแกรมซิกวินพร้อมทำการกำหนดสภาพแวดล้อม (configure) หากไม่มีข่าวสารข้อผิดพลาด (Error message) ปรากฏ หมายถึงการติดตั้งเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

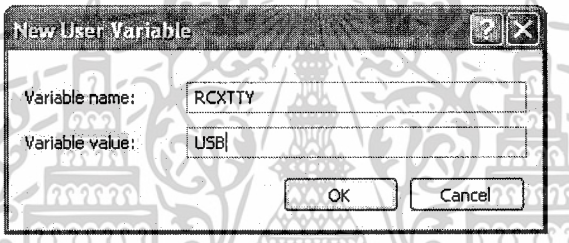
3. กำหนดตัวแปรให้กับ IR Tower

หลังจากติดตั้งโปรแกรมตัวขับอุปกรณ์ (Device driver) ของอินฟราเรดทาวเวอร์ (IR Tower) เสร็จแล้ว ให้ทำการตรวจสอบตัวแปรสภาพแวดล้อมภายใต้ระบบปฏิบัติการวินโดว์ ว่ามีการกำหนดค่าไว้แล้วหรือไม่ดังนี้

1.24 เลือกที่ไอคอน "My computer" บนเดสก์ท็อป Desktop แล้วคลิ๊กเม้าท์ปุ่มขวา เลือก "Properties" จะปรากฏหน้าต่าง "System Properties" ดังรูปที่ ข.13จากนั้น เลือกไปที่แทป (Tab) Advanced แล้วคลิ๊กปุ่มตัวแปรสภาพแวดล้อม(environment variables)

1.25 ตรวจสอบตัวแปร "RCXTTY" ว่ามีการสร้างขึ้นไว้หรือไม่ (คล้ายกับข้อ 1.15) หากยังไม่มี ให้คลิ๊กที่ปุ่ม "New" เพื่อสร้างขึ้นใหม่ ดังรูป ข.14

1.26 ทำการเพิ่มตัวแปร (Variable name) ชื่อ "RCXTTY" โดยพิมพ์ค่าตัวแปร (Variable value) ชื่อ "USB" แล้วคลิ๊ก "OK" ดังรูปที่ ข.21



รูปที่ ข.21 หน้าต่างจัดการตัวแปรสำหรับผู้ใช้ (Edit User Variables) ตัวแปร "RCXTTY"

4. การถ่ายโอนเฟิร์มแวร์ (firmware) บริคโอเอส (BrickOS) ลงบนอาร์วีเอ็กซ์ (RCX)

หลังจากติดตั้งบริคโอเอส (BrickOS) เสร็จแล้ว ก็จะปรากฏไดเร็กทอรี่ย่อย (sub directory) คือ "c:\cygwin\brickOS" ปรากฏ จากนั้นให้ทำการถ่ายโอน (transfer) เฟิร์มแวร์ (firmware) ของบริคโอเอส (BrickOS) ลงบนเลโก้อาร์ซีเอ็กซ์ (Lego RCX) ตามขั้นตอนดังนี้

1.27 เชื่อมต่ออินฟราเรดทาวเวอร์ (IR Tower) เข้ากับพอร์ทอนุกรม (serial port) เช่น "COM1" เป็นต้น ของเครื่องคอมพิวเตอร์ที่ได้ติดตั้งโปรแกรมซิกวิน (cygwin) และบริคโอเอส (BrickOS) รวมทั้งโปรแกรมตัวขับอุปกรณ์ (Device driver) ของอินฟราเรดทาวเวอร์ (IR Tower) ข้างต้น

1.28 เปิดสวิทซ์บนตัวอาร์ซีเอ็กซ์ (RCX) เพื่อให้มันเริ่มทำงาน และจัดวางอุปกรณ์โดยหันพอร์ทอินฟาเรด (IR port) บนตัวอาร์ซีเอ็กซ์ (RCX) ให้ตรงกับอินฟราเรดทาวเวอร์ (IR Tower)

1.29 รันโปรแกรมซิกวิน (cygwin) จากนั้นพิมพ์คำสั่งดังนี้ลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน นาย ธนชัย หาญอิทธิกุล

วันเดือนปีเกิด 14 กรกฎาคม 2529

สถานที่เกิด กรุงเทพมหานคร

ประวัติการศึกษา

- ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาพัฒนาการ ปีการศึกษา 2544
- ศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

ชื่อผู้เขียน นาย นที วุฒิกุลวานิช

วันเดือนปีเกิด 19 มีนาคม 2529

สถานที่เกิด กรุงเทพมหานคร

ประวัติการศึกษา

- ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาพัฒนาการ ปีการศึกษา 2544
- ศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้