ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

หุ่นยนต์ติดตามเส้นทาง

LINE TRACE ROBOT

H004778

โดย

วัชรินทร์ จันทภา

นลัทพร เมืองบัว

อาจารย์ที่ปรึกษา

ดร. โอพาร วงศ์วิรัตน์

อาจารย์ที่ปรึกษาร่วม

อาจารย์อนันตพัฒน์ อนันตชัย

เลขหมู่.....
เลขทะเบียน.....04778.....
วัน,เดือน,ปี.....8 ต.ค. 2551.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LINE TRACE ROBOT

WATCHARIN JANTHAPHA

NALATPORN MUANGBOA

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2008

FACULTY ON INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2550

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ติดตามเส้นทาง

LINE TRACE ROBOT

ผู้จัดทำ

1. นางสาววัชรินทร์ จันทภา รหัสประจำตัว 47070129
2. นางสาวนลัทพร เมืองบัว รหัสประจำตัว 47070135

..... อาจารย์ที่ปรึกษา

(ดร.โอฬาร วงศ์วิรัตน์)

..... อาจารย์ที่ปรึกษาร่วม

(อาจารย์อนันตพัฒน์ อนันตชัย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	หุ่นยนต์ติดตามเส้นทาง	
นักศึกษา	นางสาววัชรินทร์ จันทภา	รหัสนักศึกษา 47070129
	นางสาวนลัทพร เมืองบัว	รหัสนักศึกษา 47070135
ปริญญา	วิทยาศาสตรบัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2550	
อาจารย์ที่ปรึกษา	ดร.โอฬาร วงศ์วิรัตน์	
อาจารย์ที่ปรึกษาร่วม	อาจารย์ อนันตพัฒน์ อนันตชัย	

บทคัดย่อ

ปัจจุบันเทคโนโลยีระบบฝังตัว (Embedded System) ได้เข้ามามีบทบาทสำคัญในการดำเนินชีวิตประจำวันเป็นอย่างมาก โดยมีการประยุกต์ใช้งานในอุปกรณ์และเครื่องใช้ต่างๆ เพื่อช่วยเพิ่มประสิทธิภาพในการทำงานและอำนวยความสะดวกในการใช้งาน รวมทั้งเพิ่มมูลค่าให้กับผลิตภัณฑ์มากขึ้น ปัจจุบันการพัฒนาแอปพลิเคชั่นระบบฝังตัวยังขาดการศึกษาและพัฒนาอย่างเป็นระบบ ทำให้การพัฒนางานด้านนี้เป็นไปอย่างล่าช้า ไม่สมบูรณ์ และขาดรูปแบบการพัฒนาอย่างเป็นขั้นตอนที่เหมาะสม

ดังนั้นโครงงานฉบับนี้จึงได้นำเสนอการประยุกต์แนวทางการพัฒนาระบบโดยใช้แนวทางการวิเคราะห์และออกแบบระบบเชิงวัตถุด้วยยูเอ็มแอล (Object Oriented Approach with UML) เพื่อนำมาพัฒนาแอปพลิเคชันระบบฝังตัวอย่างเหมาะสม โดยแอปพลิเคชั่นที่พัฒนาขึ้น คือ หุ่นยนต์ติดตามเส้นทาง (Line trace robot) ซึ่งการทำงานของหุ่นยนต์ติดตามเส้นทางจะใช้การอ่านข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซนเซอร์ (Sensor) และนำข้อมูลดังกล่าวมาประมวลผล เพื่อให้หุ่นยนต์สามรถทำงานได้อย่างถูกต้องตามที่กำหนด คือสามารถเดินตามเส้นทางในรูปแบบต่าง ๆ เลือกเส้นทาง ปรับระดับความเร็ว และเคลื่อนที่ไปยังเป้าหมายที่กำหนดได้ การพัฒนาแอปพลิเคชั่นหุ่นยนต์ติดตามเส้นทางนี้ ได้ใช้อุปกรณ์เซนเซอร์ (Sensor) แอคทูเอเตอร์ (Actuator) และหน่วยประมวลผลแบบฝังตัว (Embedded processor) บนชุดทดลองเลโก้ (Lego) รุ่นอาร์ซีเอ็กซ์ (RCX) เป็นฮาร์ดแวร์หลัก และทำการพัฒนาโปรแกรมเพื่อควบคุมการทำงานของหุ่นยนต์ด้วยภาษาC++ ซึ่งเป็นภาษาโปรแกรมเชิงวัตถุที่สอดคล้องกับแนวทางในการพัฒนาระบบที่นำเสนอดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Line trace robot		
Student	Miss Watcharin Janthapha	Student ID.	47070129
	Miss Nalatporn Muangbua	Student ID.	47070135
Degree	Bachelor of Science		
Programme	Information Technology		
Academic Year	2007		
Advisor	Dr. Olarn Wongwirat		
Co-Advisor	Mr. Anuntapat Anuntachai		

ABSTRACT

In the present, embedded system technology plays an important role in daily life. The embedded technology is applied into devices and equipment for improving operation efficiency and convenient, including increases value added to products. However, an application development of embedded systems stills lack of systematical study, currently. It causes the development in this area that becomes slow and incomplete, and misses appropriate formal step in development.

Therefore, this project presents the novel approach that applies a system development into the embedded system by acquiring an object-oriented system analysis and design using UML. The application developed in the project is a line trace robot. The line trace robot operation is to read data from environment via sensors. Then, it manipulates the data to command the robot to perform correct operations, such as tracing and following multiple line patterns, selecting lines, adjusting speeds, and moving toward the specified target. The sensors, actuators, and embedded processor used, as a main hardware, in the system come from Lego RCX. The program development to control robot operations uses C++, which is a programming language that conforms to the object oriented development approach as previously mentioned.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยการช่วยเหลือและสนับสนุนจากบุคคลหลายท่านจึงใคร่ขอแสดงความขอบคุณดังต่อไปนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ดร.โอพาร วงศ์วิรัตน์ ในการให้ความช่วยเหลือ ให้ความรู้และคำแนะนำที่ดีในทุก ๆ ด้าน ตลอดจนปรับปรุงแก้ไขจนปริญญานิพนธ์เล่มนี้สำเร็จสมบูรณ์เป็นอย่างดี

ขอขอบพระคุณอาจารย์ที่ปรึกษาร่วม อ.อนันตพัฒน์ อนันตชัย ที่คอยช่วยเหลือและให้คำปรึกษาในทุก ๆ ด้าน

ขอขอบพระคุณอาจารย์บุญประเสริฐ สุรักษ์รัตนสกุล ที่ให้คำแนะนำในเรื่องการวิเคราะห์และออกแบบระบบเชิงวัตถุด้วยยูเอ็มแอล รวมถึงขอขอบพระคุณคณาจารย์ทุกท่านที่คอยประสิทธิ์ประสาทความรู้และให้ประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณนายภาสกร จินานุกูลวงษ์ ที่คอยช่วยเหลือและให้คำแนะนำที่ดีในทุก ๆ ด้านเสมอมา รวมถึงขอขอบคุณนายเอกชัย หาญอาษา นายวัณรพ อาจคำพันธ์ และนายพัชรินทร์ สุทธิปัญโญ ที่คอยช่วยเหลือและให้คำปรึกษาทั้งในด้านการออกแบบสนามที่ใช้ในการทดลอง รวมถึงการให้คำแนะนำในการทำรายงานและการทำงานเป็นอย่างดี

ขอบคุณเพื่อนๆ พี่ๆ และน้องๆ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รวมถึงเพื่อน ๆ ทุกคนที่คอยให้กำลังใจและช่วยเหลือกันเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ครอบครัวและญาติพี่น้องทุกท่านที่เป็นแรงผลักดัน เป็นกำลังใจ และให้การสนับสนุนแก่ข้าพเจ้าในทุกด้าน คุณค่าและประโยชน์อันพึงมาจากโครงงานฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

วัชรินทร์ จันทภา
นลัทพร เมืองบัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ.. III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์.....1

1.3 ขอบเขตของโครงงาน.....1

1.4 ขั้นตอนการพัฒนาโครงงาน 2

1.5 ประโยชน์ที่ได้รับ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3

2.1 ระบบฝังตัว.....3

2.1.1 นิยาม 3

2.1.2 โครงสร้างและสถาปัตยกรรมของระบบฝังตัว.....4

2.1.3 ส่วนประกอบทางด้านฮาร์ดแวร์ 4

2.1.3.1 เอ็มพียู.....4

2.1.3.2 หน่วยความจำ.....5

2.1.3.3 บัส 6

2.1.3.4 อุปกรณ์อินพุตและเอาต์พุตพื้นฐาน 7

2.1.4 ส่วนประกอบของซอฟต์แวร์ 7

2.1.4.1 ระบบปฏิบัติการ.....8

2.1.4.2 โปรแกรมประยุกต์ 9

2.1.5 เทคโนโลยีที่เกี่ยวข้องกับระบบฝังตัว 9

2.1.5.1. การประมวลผลแบบเรียลไทม์ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.1.5.2 เทคโนโลยีอินเทอร์รัพต์ 11
2.1.5.3 เทคโนโลยีดีเอ็มเอ 11
2.1.5.4 เทคโนโลยีหน่วยความจำแคช 12
2.1.5.5 เทคโนโลยีหน่วยความจำเสมือน 13
2.1.5.6 เทคโนโลยีประหยัดพลังงาน 13
2.2 แนวทางการพัฒนาระบบ 13
2.2.1 วงจรในการพัฒนาระบบสารสนเทศ 14
2.2.2 โมเดลการพัฒนาระบบ 15
2.2.3 วิธีการพัฒนาระบบสารสนเทศ 18
2.2.3.1 การวิเคราะห์และออกแบบระบบเชิงโครงสร้าง 18
2.2.3.2 การวิเคราะห์และออกแบบระบบเชิงวัตถุ 18
2.2.4 ยูเอ็มแอล 20
2.2.5 องค์ประกอบของยูเอ็มแอล 21
2.2.6 ไดอะแกรมต่าง ๆ ของยูเอ็มแอล 23
2.2.6.1 ยูสเคสไดอะแกรม 23
2.26.2 แอคทิวิตี้ไดอะแกรม 24
2.26.3 คลาสไดอะแกรม 25
2.26.4 อ็อบเจ็คไดอะแกรม 27
2.26.5 ซีเควนไดอะแกรม 27
2.2.6.6 คอลแลบอเรชั่นไดอะแกรม 28
2.26.7 สเตตชาร์ตไดอะแกรม 29
2.2.6.8 คอมโพเนนต์ไดอะแกรม 30
2.2.6.9 ดีพลอยเมนต์ไดอะแกรม 31
2.3 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ 31
2.3.1 อาร์ซีเอ็ก (RCX) 31
2.3.2 โครงสร้างของอาร์ซีเอ็ก 32
2.3.3 ส่วนประกอบของอาร์ซีเอ็ก 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การพัฒนาระบบหุ่นยนต์ติดตามเส้นทาง 38

3.1 ความต้องการของระบบ 38

3.1.1 ความต้องการหลักของระบบ (Functional Requirement) 38

3.1.2 เงื่อนไขและสภาพแวดล้อมที่ใช้ในการพัฒนาระบบ 38

3.2 การวิเคราะห์และออกแบบระบบ 40

3.2.1 การวิเคราะห์ระบบ 40

3.2.2 การออกแบบซอฟต์แวร์ 41

3.2.2.1 ยูสเคสไดอะแกรม 41

3.2.2.2 แผนภาพกิจกรรม (Activity Diagram) 51

3.2.2.3 คลาสไดอะแกรม 63

3.2.2.4 อ็อบเจ็คไดอะแกรม 65

3.2.2.5 ซีเควนไดอะแกรม 74

3.2.2.6 สเตทชาร์ตไดอะแกรม 77

3.2.3 การออกแบบฮาร์ดแวร์ 78

3.2.3.1 แอปพลิเคชั่นระบบหุ่นยนต์ติดตามเส้นทาง 78

3.2.3.2 แนวคิดของการออกแบบฮาร์ดแวร์ 79

3.2.4 การออกแบบสภาพแวดล้อมที่ใช้ในการทดลอง 81

บทที่ 4 การทดลองและผลการทดลอง 84

4.1 การทดลองการทำงานของเซนเซอร์แสง 84

4.2 การทดลองการเคลื่อนที่ของหุ่นยนต์ 86

4.2.1 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางตรง 86

4.2.2 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางโค้ง 88

4.2.3 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางชัน 89

4.2.4 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางลัด 91

4.2.5 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม 92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 บทสรุปและข้อเสนอแนะ 95

5.1 สรุปผลโครงการ 95

5.2 ปัญหาและอุปสรรค 96

5.2.1 ปัญหาทางด้านฮาร์ดแวร์ 96

5.2.2 ปัญหาทางด้านซอฟต์แวร์ 97

5.3 ข้อเสนอแนะ 97

บรรณานุกรม 98

ภาคผนวก 99

ประวัติผู้เขียน 127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงส่วนประกอบของ Use Case Diagram 21

2.2 แสดงสัญลักษณ์ Activities Diagram....22

2.3 สัญลักษณ์แสดง Relationship ของคลาสไดอะแกรม....23

2.4 แสดงการเชื่อมต่อปุ่มกับสายอินพุต ทั้ง 4 สายของ H8/3292 24

2.5 แสดงโหมดการทำงาน 3 โหมดและตำแหน่งที่อยู่ของ อินพุต/เอาท์พุต 26

2.6 แสดงตัวแปลงสัญญาณ อนาล็อกเป็นดิจิตอล (A/D converter)....28

3.1 ตารางการวิเคราะห์และออกแบบระบบ....40

3.2 แสดงคำอธิบายยูสเคส การร้องขอข้อมูล (RequestData)....43

3.3 แสดงคำอธิบายยูสเคส การติดตามเส้นทางปกติ (TraceLine)....44

3.4 แสดงคำอธิบายยูสเคส การติดตามเส้นทางลัด (TraceShortcut)....45

3.5 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางตรง (DirectLine)....46

3.6 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางโค้ง (CurveLine)....47

3.7 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางลาดชัน (SlopeLine) 48

3.8 แสดงคำอธิบายยูสเคส เลือกเส้นทาง (SelectLine)....49

3.9 แสดงคำอธิบายยูสเคส ควบคุมความเร็ว (ControlSpeed)....50

3.10 แสดง CRC Card ของคลาส Sensor 66

3.11 แสดง CRC Card ของคลาส LightSensor....67

3.12 แสดง CRC Card ของคลาส TraceControl....68

3.13 แสดง CRC Card ของคลาส Trace_Line....69

3.14 แสดง CRC Card ของคลาส Trace_ShortCut....70

3.15 แสดง CRC Card ของคลาส Motor 71

3.16 แสดง CRC Card ของคลาส DrivingMotor....72

3.17 แสดง CRC Card ของคลาส SteeringMotor....73

3.18 แสดงการออกแบบส่วนของการรับค่าข้อมูล....79

3.19 แสดงการออกแบบส่วนของสเตียริงมอเตอร์ 80

3.20 แสดงการออกแบบส่วนของไดร์วิงมอเตอร์ 81

3.21 สภาพแวดล้อมที่ใช้ในการทดลอง....82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง(ต่อ)

ตารางที่ หน้า

4.1 แสดงวิธีการทดลองการทำงานของเซนเซอร์แสง 84

4.2 แสดงค่าสีที่ได้จากการทดลองการรับค่าแสงจากเซนเซอร์แสง.....85

4.3 แสดงค่าสีเมื่อทำการแปลงเป็นเลขฐานสิบ.....85

4.4 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางตรง.....87

4.5 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางโค้ง.....88

4.6 แสดงการทดลองและผลการทหลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางชัน.....90

4.7 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางลัด.....91

4.8 แสดงการทดลองการและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม.....92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างโดยสังเขปของระบบฝังตัว 3
2.2 สถาปัตยกรรมพื้นฐานทางด้านฮาร์ดแวร์ของระบบฝังตัว 4
2.3 การทำงานของเอ็มพียู 5
2.4 การแบ่งประเภทหน่วยความจำ 6
2.5 ส่วนประกอบของซอฟต์แวร์ 7
2.6 ฟังก์ชันการทำงานหลักของเคอร์เนล 8
2.7 โครงสร้างของดีไวซ์ไดรเวอร์ 9
2.8 การตอบสนองแบบเรียลไทม์ 10
2.9 การเคลื่อนย้ายข้อมูลด้วยวิธีการดีเอ็มเอ 12
2.10 การทำงานของหน่วยความจำแคช 12
2.11 ระบบหน่วยความจำเสมือน 13
2.12 วงจรในการพัฒนาระบบสารสนเทศ 15
2.13 แบบจำลองน้ำตกแบบดั้งเดิมและแบบจำลองน้ำตกแบบปรับปรุง 16
2.14 แบบจำลองแบบก้นหอย 16
2.15 แบบจำลองแบบเพิ่มขึ้น 17
2.16 Object-Oriented approach 19
2.17 ตัวอย่าง Polymorphism 19
2.18 ตัวอย่างสัญลักษณ์ต่าง ๆ ที่ใช้ในไดอะแกรม 21
2.19 ประเภทของไดอะแกรม 23
2.20 สัญลักษณ์ที่ใช้ใน Class diagram 25
2.21 multiplicity 26
2.22 สัญลักษณ์ Object Diagram 27
2.23 แสดงส่วนประกอบของซีเควนไดอะแกรม 28
2.24 ตัวอย่าง Collaboration Diagram 29
2.25 แสดงสัญลักษณ์ของ State chart Diagram 29
2.26 แสดงสัญลักษณ์การจำลองกิจกรรมของ State 30
2.27 สัญลักษณ์ใน Component Diagram 30
2.28 สัญลักษณ์ของ Deployment Diagram 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.29 ส่วนประกอบของ RCX 32
2.30 แสดงส่วนประกอบและโครงสร้างของ RCX 32
2.31 แสดงปุ่มควบคุมการทำงานของ RCX 34
2.32 เซนเซอร์สัมผัส (Touch Sensor) 34
2.33 เซนเซอร์แสง (Light Sensor) 35
2.34 เซนเซอร์จับการหมุน (Rotation Sensor) 35
2.35 มอเตอร์ 35
2.36 เครื่องส่งสัญญาณด้วยแสงอินฟราเรด 36
3.1 แสดงสภาพถนนที่เป็นเงื่อนไขของระบบ 39
3.2 แสดงสภาพถนนที่เป็นเส้นทางลัด 39
3.3 แสดง Use Case Diagram 41
3.4 แสดงแผนภาพกิจกรรมภาพรวมระบบ 51
3.5 แสดงแผนภาพกิจกรรม ร้องขอข้อมูล (RequestData) 52
3.6 แสดงแผนภาพกิจกรรม การติดตามเส้นทางปกติ (TraceLine) 54
3.7 แสดงแผนภาพกิจกรรม การติดตามเส้นทางลัด (TraceShortCut) 55
3.8 แสดงแผนภาพกิจกรรม เคลื่อนที่ไปในทางตรง (DirectLine) 56
3.9 แสดงแผนภาพกิจกรรม เคลื่อนที่ไปในทางโค้ง (CurveLine) 57
3.10 แสดงแผนภาพกิจกรรม เคลื่อนที่ไปในทางลาดชัน (SlopeLine) 59
3.11 แสดงแผนภาพกิจกรรม เลือกเส้นทาง (SelectLine) 61
3.12 แสดงแผนภาพกิจกรรม ควบคุมความเร็ว (ControlSpeed) 62
3.13 แสดงแผนภาพ Class diagram 63
3.14 แสดงแผนภาพ Object Diagram 65
3.15 แสดงแผนภาพ Sequence Diagram ของ RequestData 74
3.16 แสดงแผนภาพ Sequence Diagram ของ TraceLine 75
3.17 แสดงแผนภาพ Sequence Diagram ของ TraceShortCut 76
3.18 แสดง Statechart Diagram 77
3.19 แสดงส่วนประกอบต่างๆของหุ่นยนต์ติดตามเส้นทาง 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.20 แสดงส่วนประกอบทางด้านข้างของหุ่นยนต์ติดตามเส้นทาง..... 78

3.21 แสดงการติดตั้งส่วนของมอเตอร์ 79

3.22 แสดงส่วนของสเตียริงมอเตอร์ (Steering motor) 80

3.23 ส่วนประกอบในส่วนของไดร์วิงมอเตอร์ (Driving Motor) 81

3.24 สภาพแวดล้อมที่ใช้ในการทดลอง 83

4.1 แสดงการทดลองการรับค่าความเข้มสีต่างๆจากเซนเซอร์แสง 86

4.2 การทดลองการเคลื่อนที่ของหุ่นยนต์บนเส้นทางตรงระยะทาง 3 เมตร 87

4.3 การทดลองการเคลื่อนที่ของหุ่นยนต์บน เส้นทางตรงระยะทาง 0.75 เมตร 87

4.4 การทดลองการเคลื่อนที่บนเส้นโค้ง 90 องศา..... 88

4.5 การทดลองการเคลื่อนที่บนเส้นโค้ง แบบต่อเนื่อง..... 89

4.6 การทดลองการเคลื่อนที่ขึ้นทางชัน 90

4.7 การทดลองการเคลื่อนที่บนเส้นทางลัด 91

4.8 ทดลองการเคลื่อนที่บนเส้นทางตรงในสภาพสนามรวม 93

4.9 ทดลองการเคลื่อนที่บนเส้นทางโค้งในสภาพสนามรวม 93

4.10 ทดลองการเคลื่อนที่บนเส้นทางลัดในสภาพสนามรวม 94

4.11 ทดลองการเคลื่อนที่บนเส้นทางชันในสภาพสนามรวม 94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันเทคโนโลยีของระบบฝังตัว (Embedded System) ได้เข้ามามีบทบาทสำคัญในการดำเนินชีวิตประจำวันเป็นอย่างมาก โดยมีการประยุกต์ใช้งานในด้าน รถยนต์ อุปกรณ์ควบคุมทางอุตสาหกรรม อุปกรณ์เครื่องมือด้านการแพทย์ อุปกรณ์เครือข่าย เครื่องใช้ไฟฟ้าและอุปกรณ์สำนักงาน ทำให้เกิดความสะดวกสบาย ช่วยเพิ่มประสิทธิภาพการทำงานและความหลากหลายของการใช้งาน รวมไปถึงการเพิ่มมูลค่าให้กับผลิตภัณฑ์มากขึ้น ในปัจจุบันการพัฒนาแอปพลิเคชั่นระบบฝังตัวยังขาดการศึกษาและพัฒนาอย่างเป็นระบบ ทำให้การพัฒนางานด้านนี้เป็นไปอย่างล่าช้าไม่สมบูรณ์แบบและขาดรูปแบบการพัฒนาอย่างเป็นขั้นตอนที่เหมาะสม

ดังนั้นโครงงานนี้จึงเป็นการประยุกต์แนวทางการพัฒนาระบบ เพื่อนำมาพัฒนาและสร้างแอปพลิเคชั่นระบบฝังตัวขึ้น โดยใช้เครื่องมือต่างๆ เข้ามาช่วยเพื่อให้การพัฒนาแอปพลิเคชั่นเป็นไปอย่างมีระบบ โดยแอปพลิเคชั่นที่พัฒนาขึ้น คือ หุ่นยนต์ติดตามเส้นทาง (Line trace robot) หลักการทำงานของหุ่นยนต์ติดตามเส้นทางจะทำการอ่านค่าข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซ็นเซอร์ (Sensor) และนำข้อมูลดังกล่าวมาประมวลผลเพื่อให้ทำงานได้ถูกต้องตามที่กำหนด คือ สามารถติดตามเส้นทางทั้งทางตรง ทางโค้ง ทางชัน ทางลัด สามารถเลือกเส้นทาง ปรับระดับความเร็วในการเคลื่อนที่และเคลื่อนที่ไปยังเป้าหมายที่กำหนดได้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาองค์ประกอบและโครงสร้างของระบบฝังตัว และการประยุกต์ใช้งาน
2. เพื่อศึกษาแนวทางและเครื่องมือในการพัฒนาระบบฝังตัว
3. เพื่อออกแบบและพัฒนาแอปพลิเคชั่นระบบฝังตัว หุ่นยนต์ติดตามเส้นทาง

1.3 ขอบเขตของโครงงาน

ปริญญานิพนธ์นี้เป็นการศึกษาและพัฒนาแอปพลิเคชั่นระบบฝังตัว โดยประยุกต์แนวทางในการพัฒนาระบบฝังตัว ฮาร์ดแวร์(Hardware) และซอฟต์แวร์ (Software) และทำการออกแบบแอปพลิเคชั่นหุ่นยนต์ติดตามเส้นทาง รวมทั้งทำการเขียนโปรแกรมเพื่อสั่งงานให้ทำงานได้ตามที่ต้องการ รวมถึงทำการทดสอบและสรุปผลที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการพัฒนาโครงงาน

1. ศึกษาระบบฝังตัว (Embedded Systems) และแนวทางในการประยุกต์ใช้งาน
2. ศึกษากระบวนการพัฒนาระบบเพื่อเป็นแนวทางในการพัฒนาแอปพลิเคชั่นระบบฝังตัวที่สร้างขึ้น
3. ศึกษาส่วนประกอบต่างๆและหลักการทำงานของอุปกรณ์ชุดทดลอง Lego รุ่น RCX เพื่อใช้พัฒนาแอปพลิเคชั่นระบบฝังตัวที่สร้างขึ้น คือ หุ่นยนต์ติดตามเส้นทาง
4. ศึกษาการเขียนโปรแกรมเพื่อสั่งงานอุปกรณ์ เซ็นเซอร์ (Sensor) และแอคทูเอเตอร์ (Actuator) บนชุดทดลอง Lego รุ่น RCX
5. ศึกษาและวิเคราะห์ความต้องการของระบบ
6. วิเคราะห์และออกแบบโปรแกรมแอปพลิเคชั่นระบบฝังตัวสำหรับหุ่นยนต์ติดตามเส้นทางโดยใช้อัลกอลิทึมที่เหมาะสม
7. ทดสอบการทำงานของแอปพลิเคชั่นระบบฝังตัวที่สร้างขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเรียนรู้แนวทางการพัฒนาระบบและประยุกต์ใช้งานระบบฝังตัวได้
2. สามารถพัฒนาระบบฝังตัวโดยใช้เครื่องมือจากแนวทางการพัฒนาระบบได้อย่างเหมาะสม
3. สามารถทำการออกแบบและพัฒนาแอปพลิเคชั่นระบบฝังตัวได้
4. สามารถเรียนรู้หลักการทำงานและสั่งงานอุปกรณ์ต่างๆ ด้วยโปรแกรมภาษาคอมพิวเตอร์
5. สามารถนำไปประยุกต์ใช้งานและเป็นแนวทางในการพัฒนาแอปพลิเคชั่นระบบฝังตัวอื่น ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีบทที่เกี่ยวข้อง

2.1 ระบบฝังตัว

2.1.1 นิยาม

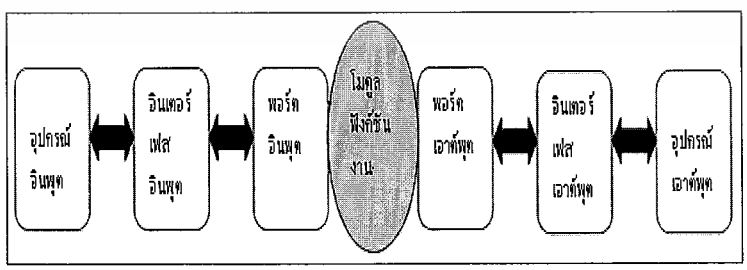
ระบบฝังตัว (Embedded Systems) คือระบบที่มีไมโครชิปหรือโปรแกรมเพื่อควบคุมการทำงานฝังตัวอยู่ เช่น อุปกรณ์อัตโนมัติต่างๆ เครื่องใช้ไฟฟ้า เครื่องเล่นอิเล็กทรอนิกส์ หุ่นยนต์ รวมถึงอุปกรณ์ที่ควบคุมการทำงานต่าง ๆ ของรถยนต์ เป็นต้น

ลักษณะเฉพาะของระบบฝังตัวนั้น เป็นระบบที่มีการทำงานร่วมกันของฮาร์ดแวร์และซอฟต์แวร์ มีความหลากหลายทั้งฟังก์ชันการทำงาน ส่วนประกอบ และเทคโนโลยีที่นำมาใช้ซึ่งแตกต่างกันไปในแต่ละอุปกรณ์ นอกจากนี้ยังมีการประมวลผลแบบเรียลไทม์ (Real-time) ซึ่งเป็นการประมวลผลและตอบสนองต่อเหตุการณ์ที่เกิดขึ้นพร้อม ๆ กันภายในเวลาที่กำหนดได้

2.1.2 โครงสร้างของระบบฝังตัว

โดยทั่วไประบบฝังตัวจะประกอบขึ้นจากทรัพยากรระบบ 2 ส่วนคือฮาร์ดแวร์ (Hardware) และซอฟต์แวร์(Software) ได้แก่ หน่วยประมวลผล (Microcontroller) หน่วยความจำ (Memory) อุปกรณ์อินพุต/เอาต์พุต(Input/Output Devices) โปรแกรมข้อมูล เป็นต้น นอกจากนี้อาจมีทรัพยากรด้านเนตเวิร์ค (Network) ซึ่งเชื่อมโยงอุปกรณ์ต่าง ๆ เข้าด้วยกัน

การทำงานของระบบฝังตัวส่วนใหญ่จะมีอุปกรณ์อินพุต/เอาต์พุต(Input/Output Devices) ทำหน้าที่ติดต่อกับสิ่งแวดล้อมภายนอก ข้อมูลจะถูกนำเข้าและส่งออกผ่านทางอินเตอร์เฟสเข้าสู่พอร์ตอินพุต/เอาต์พุต โดยมีโมดูลฟังก์ชันงานทำหน้าที่ในการประมวลผลการทำงานต่าง ๆ โดยที่อุปกรณ์ที่เป็นอินเตอร์เฟซอินพุต/เอาต์พุตและโมดูลฟังก์ชันงานนี้จะถูกควบคุมด้วยส่วนที่เป็นซอฟต์แวร์ของระบบฝังตัว ดังรูปที่ 2.1

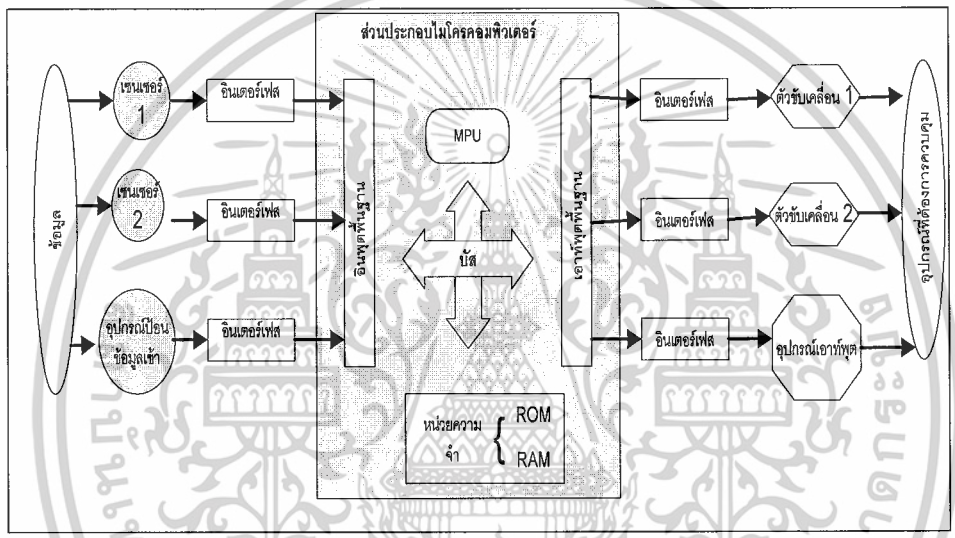


รูปที่ 2.1 โครงสร้างโดยสังเขปของระบบฝังตัว (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 89)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ส่วนประกอบทางด้านฮาร์ดแวร์

ส่วนประกอบทางด้านฮาร์ดแวร์ของระบบฝังตัวจะแตกต่างกันไปตามการประยุกต์ใช้งานแต่จะประกอบไปด้วยโครงสร้างพื้นฐานเช่นเดียวกัน คือ ส่วนไมโครคอมพิวเตอร์ (Microcomputer) ซึ่งภายในประกอบไปด้วย หน่วยประมวลผลหรือเอ็มพียู (MPU: micro-processing unit) หน่วยความจำ (Memory) พอร์ตอินพุต/เอาต์พุตพื้นฐาน (basic I/O) ซึ่งจะติดต่อกันผ่านทางบัส (BUS) และมีการเชื่อมต่อไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านทางอินเตอร์เฟซ (Interface) ซึ่งทำหน้าที่ในการแปลงสัญญาณให้อยู่ในรูปแบบที่เหมาะสมต่อการใช้งาน และการทำงานของเอ็มพียู เช่น การปรับจังหวะเวลา การจัดการเหตุการณ์ที่เข้ามาจากหลายช่องสัญญาณ เป็นต้น ดังรูปที่ 2.2



รูปที่ 2.2 สถาปัตยกรรมพื้นฐานทางด้านฮาร์ดแวร์ของระบบฝังตัว(Japan System House Association. 2549. แปลและเรียบเรียง โดย ดร.ธนารัตน์ ธีระมั่นคง : 90)

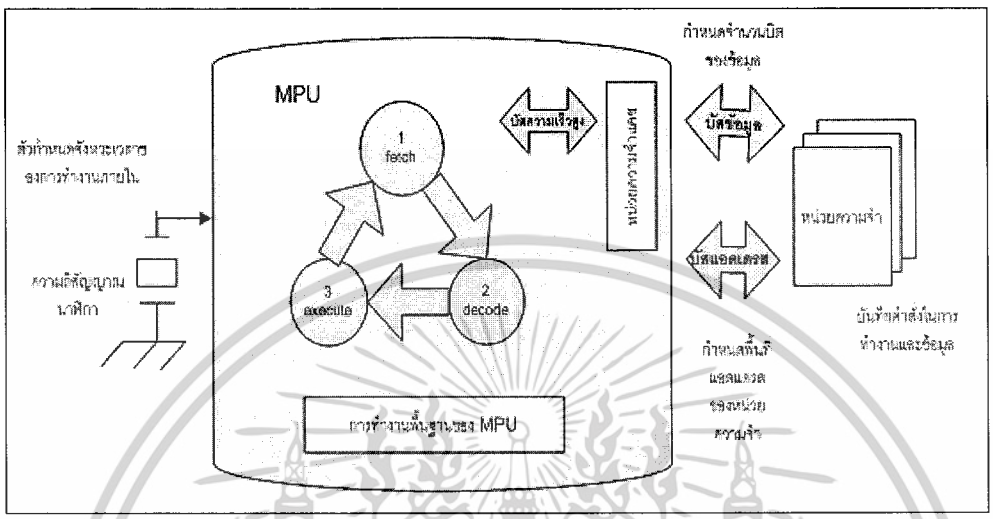
2.1.3.1 เอ็มพียู

เอ็มพียูเป็นหัวใจหลักในการทำงานของไมโครคอมพิวเตอร์ ทำหน้าที่ในการประมวลผลคำสั่งและควบคุมการทำงานต่าง ๆ โดยปกติแล้วเอ็มพียูจะมีการทำงานโดยเริ่มจากการอ่านคำสั่ง (Fetch) แล้วทำการตีความคำสั่ง (Decode) จากนั้นจึงทำตามคำสั่ง (Execute) นอกจากนี้ยังอาจมีการทำงานแบบไปป์ไลน์ (Pipeline) คือ การประมวลผลหลาย ๆ คำสั่งไปพร้อมกันดังรูปที่ 2.3 โดยเอ็มพียูสามารถแบ่งโครงสร้างภายในได้เป็น 2 ประเภทหลัก ๆ คือ CISE และ RISC

1. CISC (Complex Instruction Set Computer) เป็นเอ็มพียูที่ออกแบบมาให้มีคำสั่งในการทำงานที่ซับซ้อนจำนวนมาก เพื่อให้สามารถตอบสนองกับความต้องการโปรแกรมฟังก์ชันต่าง ๆ มีความซับซ้อนสูง ความถี่ของสัญญาณนาฬิกามีหลายลูก และสามารถเข้าถึงหน่วยความจำได้โดยตรงไม่ต้องผ่านรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. RISC (Reduced Instruction Set Computing) ถูกออกแบบมาให้มีคำสั่งจำนวนน้อย โครงสร้างไม่ซับซ้อน ใช้คำสั่งง่ายๆ เพื่อให้ประมวลผลได้อย่างรวดเร็ว สามารถใช้กับสัญญาณนาฬิกาที่มีความถี่สูงได้ การทำงานมักถูกออกแบบมาให้ใช้รีจิสเตอร์ภายใน



รูปที่ 2.3 การทำงานของเอ็มพียู (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 93)

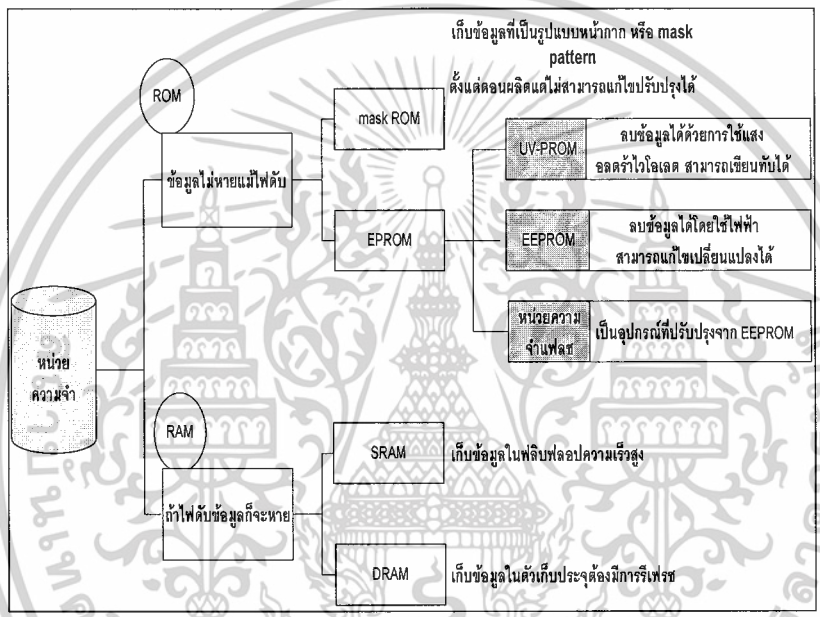
2.1.3.2 หน่วยความจำ

หน่วยความจำเป็นส่วนสำหรับเก็บคำสั่งและข้อมูลต่าง ๆ โดยจะเก็บข้อมูลในรูปตัวเลขฐานสอง ซึ่งก็คือสัญญาณทางไฟฟ้านั่นเอง สำหรับการเก็บข้อมูลจะเก็บรวมกันเป็นกลุ่มบิตข้อมูล เช่น 8 บิต รวมกันเป็น 1 ไบต์ และมีการกำหนดตำแหน่งที่อยู่ข้อมูล (Address) เพื่อใช้อ้างถึงในการเขียนหรืออ่านข้อมูลนั้น หน่วยความจำแบ่งได้เป็น 2 ประเภท คือ หน่วยความจำรอม (ROM: Read Only Memory) และแรม (RAM: Random Access Memory) ดังรูปที่ 2.4

1. รอม เป็นหน่วยความจำที่มีคุณสมบัติพิเศษ คือ ข้อมูลจะไม่หายไปไหนแม้ไม่มีไฟเลี้ยง (non-volatile) ดังนั้นรอมจึงเป็นที่เก็บโปรแกรมเริ่มต้นที่จะถูกเอ็มพียูอ่านเข้าไปเพื่อตีความคำสั่งขณะเริ่มต้นการทำงานหลังจากที่มีการจ่ายไฟเข้าระบบ ซึ่งโปรแกรมนี้จะถูกเรียกทำงานก่อนเพื่อโหลดโปรแกรม โดยทั่วไปรอมแบ่งได้เป็น 2 ประเภทหลัก ๆ คือ mask ROM ซึ่งเป็นรอมที่ถูกเขียนค่าไว้เริ่มต้นตั้งแต่ขั้นตอนการผลิต และ EPROM(Erasable Programmable Read-Only Memory) เป็นรอมที่จะเขียนข้อมูลเข้าภายหลังหรือสามารถลบได้ นอกจากนี้ EPROM ยังสามารถแบ่งได้เป็นประเภทต่าง ๆ ได้แก่ UV-PROM เป็นรอมที่สามารถลบข้อมูลได้ด้วยการฉายแสงอัลตราไวโอเลตหรือยูวี (UV) และ EEPROM เป็นรอมที่สามารถลบหรือเขียนข้อมูลเข้าไปใหม่ด้วยไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แรมเป็นหน่วยความจำที่สามารถเขียนหรืออ่านได้ แต่ข้อมูลจะหายเมื่อไม่มีการจ่ายไฟ (Volatile) เอ็มพียูจะทำงานโดยเก็บหรือเขียนทับค่าของตัวแปรที่ใช้งานหรือเป็นบัฟเฟอร์สำหรับตัวแปรหรือข้อมูลต่างๆ ซึ่งขนาดจะต้องไม่เล็กไปกว่าขนาดที่จำเป็นสำหรับการใช้งาน โดยทั่วไปแรมแบ่งได้เป็น 2 ประเภท คือ SRAM (Static Random Access Memory) และ DRAM (Dynamic Random Access Memory) โดยที่ SRAM เป็นแรมที่ประกอบเป็นวงจรฟลิปฟลอปทำงานด้วยความเร็วสูง นิยมนำมาเป็นหน่วยความจำแบบแคช(Cache memory) ส่วน DRAM เป็นแรมที่มีการเก็บข้อมูลที่ต้องมีการรีเฟรช(Refresh) เพื่อให้ข้อมูลยังคงอยู่ สามารถสร้างให้เป็นหน่วยความจำที่มีขนาดใหญ่ได้ แต่ความเร็วในการเข้าถึงจะช้ากว่าแบบ SRAM



รูปที่ 2.4 การแบ่งประเภทหน่วยความจำยู (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 94)

2.1.3.3 บัส

บัส(Bus) คือ เส้นทางการติดต่อสื่อสารระหว่างข้อมูล สามารถแบ่งตามตำแหน่งการใช้งานได้เป็น2 ประเภทหลัก ๆ ได้แก่ บัสภายใน(Internal Bus) เป็นกลุ่มของสัญญาณไมโครคอนโทรลเลอร์ที่ใช้ร่วมกันสำหรับส่งผ่านข้อมูลระหว่างเอ็มพียู หน่วยความจำ อินพุต/เอาต์พุต และบัสภายนอก (External Bus) เป็นกลุ่มของสายสัญญาณที่ใช้ส่งข้อมูลระหว่างไมโครคอมพิวเตอร์และอุปกรณ์ภายนอกต่าง ๆ บัสแบ่งได้เป็น 3 ประเภทหลัก ๆ ได้แก่

1. แอดเดรสบัส (Address bus) ใช้สำหรับระบุตำแหน่งในการรับส่งข้อมูล
2. ดาต้าบัส (Data bus) ใช้สำหรับแลกเปลี่ยนข้อมูล
3. คอลโทรลบัส (Control bus) เป็นตัวกำหนดและควบคุม แอดเดรสบัสและดาต้าบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.4 อุปกรณ์อินพุตและเอาต์พุตพื้นฐาน

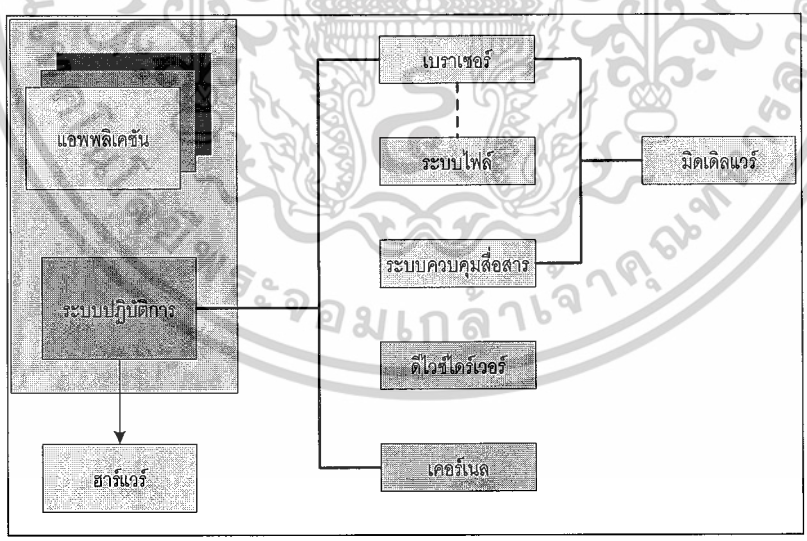
ในระบบฝังตัวนั้นเอ็มพียูจะมีการรับส่งข้อมูลจากภายนอกเข้ามาประมวลผลก่อนที่จะส่งออกไปภายนอกอีกครั้ง ดังนั้นจะต้องมีการแลกเปลี่ยนข้อมูลระหว่างเอ็มพียูกับอุปกรณ์ภายนอกผ่านทางอินเตอร์เฟส(I/O interface) ซึ่งอุปกรณ์ภายนอกประกอบด้วยอุปกรณ์อินพุตและอุปกรณ์เอาต์พุต

อุปกรณ์อินพุตพื้นฐาน เป็นส่วนรับข้อมูลและสัญญาณต่างๆ ของไมโครคอมพิวเตอร์ โดยจะส่งข้อมูลในรูปของข้อมูลดิจิตอลให้กับเอ็มพียูเพื่อนำไปประมวลผล โดยทั่วไปเรียกว่า อินพุตพอร์ต โดยผ่านอุปกรณ์หรือวงจรไฟฟ้าเพื่อทำหน้าที่แปลงสัญญาณให้อยู่ในรูปแบบที่เหมาะสมเรียกว่า วงจรอินเตอร์เฟส (Interface circuit) อุปกรณ์อินพุตเช่น เซนเซอร์

อุปกรณ์เอาต์พุตพื้นฐานเป็นช่องทางออกของสัญญาณที่เกิดจากการประมวลผลของเอ็มพียู โดยจะมีการแปลงข้อมูลแบบดิจิตอลให้เป็นสัญญาณทางกายภาพที่เหมาะสมเรียกว่าเอาต์พุตพอร์ต ซึ่งเป็นส่วนสำคัญในการเชื่อมการทำงานของเอ็มพียูกับอุปกรณ์ภายนอก อุปกรณ์เอาต์พุตได้แก่ ตัวขับเคลื่อน (Actuator) เช่นมอเตอร์ จอแสดงผล เป็นต้น

2.1.4 ส่วนประกอบของซอฟต์แวร์

ส่วนประกอบของซอฟต์แวร์โดยทั่วไปแบ่งได้เป็น 2 ประเภทใหญ่ ๆ คือ ระบบปฏิบัติการ (Operating system) และโปรแกรมประยุกต์ (application program) ดังรูปที่ 2.5



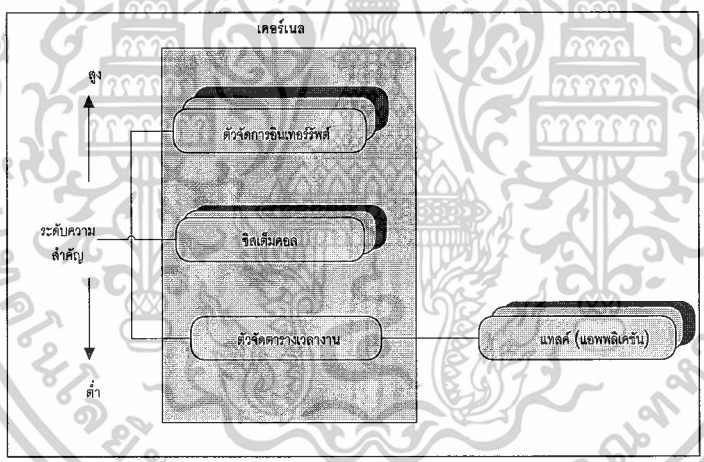
รูปที่ 2.5 ส่วนประกอบของซอฟต์แวร์(Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 95)

2.1.4.1 ระบบปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปฏิบัติการเป็นส่วนของซอฟต์แวร์ระบบที่ทำหน้าที่เป็นตัวกลางระหว่างฮาร์ดแวร์และโปรแกรมประยุกต์ทั่วไป บางครั้งอาจเป็นเฟิร์มแวร์ก็ได้ มีหน้าที่หลัก ๆ คือการจัดสรรทรัพยากรในระบบเพื่อให้บริการซอฟต์แวร์ประยุกต์ ควบคุมการรับส่งและจัดเก็บข้อมูล จัดสรรพื้นที่หน่วยความจำ รวมทั้งทำหน้าที่จัดสรรเวลาในการประมวลผล ระบบปฏิบัติการประกอบด้วยเคอร์เนล (Kernel) มิดเดิลแวร์ (Middleware) และ ดีไวซ์ไดร์เวอร์ (Devices Driver)

เคอร์เนล เป็นโมดูลกลางที่ทำหน้าที่ในการให้กำเนิดฟังก์ชันพื้นฐานของระบบปฏิบัติการให้กำเนิดฟังก์ชันของการจัดตารางงานแบบขับเคลื่อนด้วยเหตุการณ์ เรียกว่า เรียลไทม์เคอร์เนล(Real-time kernel) หรือเรียลไทม์โอเอส (Real-time Operating System) ซึ่งเป็นส่วนสนับสนุนฟังก์ชันพื้นฐาน ที่ทำให้เกิดการประมวลผลแบบเรียลไทม์ การทำงานหลักของเคอร์เนลจะประกอบไปด้วยฟังก์ชันการจัดตารางงานทำหน้าที่ในการจัดการเหตุการณ์ที่เกิดขึ้นพร้อมกันหลาย ๆ เหตุการณ์ ฟังก์ชันควบคุมการอินเทอร์รัพต์ (Interrupt) ทำหน้าที่จัดการและดูแลอินเตอร์รัพต์ที่เกิดขึ้นตามแต่ละประเภท และฟังก์ชันซิสเต็มคอล (System call) ทำหน้าที่จัดการการเรียกใช้บริการจากระบบ ดังรูปที่ 2.6



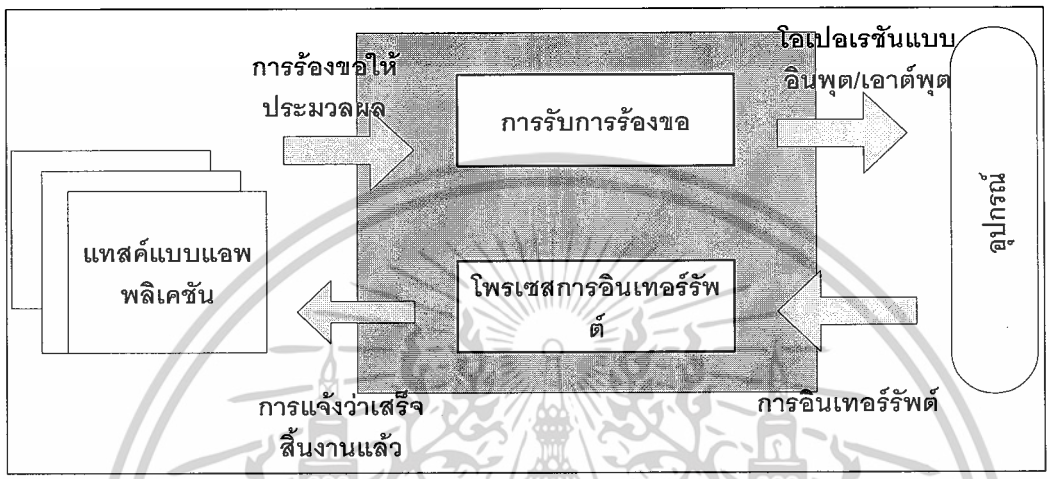
รูปที่ 2.6 ฟังก์ชันการทำงานหลักของเคอร์เนล (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 96)

มิดเดิลแวร์ เป็นโมดูลที่สนับสนุนฟังก์ชันทั่วไปของระบบ ทำหน้าที่เป็นตัวกลาง เป็นตัวประสานบริการ(Service) ต่างๆ ให้กับแอพพลิเคชั่นโดยผ่านทางเคอร์เนลที่ควบคุมการทำงานของระบบปฏิบัติการ ทำให้ใช้งานแอพพลิเคชั่นต่างๆ ได้สะดวกขึ้น ซึ่งอาจประกอบด้วยไลบรารี่ (Library) เช่นระบบไฟล์ ฟังก์ชันควบคุมการสื่อสาร เป็นต้น

ดีไวซ์ไดร์เวอร์ เป็นส่วนควบคุมอุปกรณ์อินพุต/เอาต์พุต เพื่อให้เกิดฟังก์ชั่นในการรับส่งข้อมูลกับโปรแกรมประยุกต์ บางครั้งเรียกว่าระบบควบคุมอินพุตเอาต์พุต (Input & Output Control System - IOCS) ระบบปฏิบัติการโดยทั่วไปดีไวซ์ไดร์เวอร์จะเป็นส่วนหนึ่งของระบบปฏิบัติการเลย

เอกสารนี้เป็นเอกสารที่สงวนสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าเป็นซอฟต์แวร์ระบบฝังตัวนั้นจะแตกต่างกันไปตามผลิตภัณฑ์ และการใช้งานอุปกรณ์ ดังนั้นจึงไม่สามารถใช้ฟังก์ชันของดีไวซ์ไดร์เวอร์แบบเฉพาะเจาะจงลงไปในระบบปฏิบัติการได้ ส่วนใหญ่จะมีเพียงดีไวซ์ไดรเวอร์พื้นฐานหลัก ๆ เช่น ตัวจับเวลา (timer) อินเตอร์เฟสชนิดซีเรียล (Serial I/F) เป็นต้น สำหรับดีไวซ์ไดรเวอร์อื่น ๆ นั้นอาจจะต้องหาจากซอฟต์แวร์อื่นที่มีอยู่ หรืออาจต้องพัฒนาขึ้นมาเอง โดยดีไวซ์ไดร์เวอร์ มีโครงสร้างดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างของดีไวซ์ไดรเวอร์ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 97)

2.1.4.2 โปรแกรมประยุกต์

โปรแกรมประยุกต์เป็นซอฟต์แวร์ที่ใช้กับงานด้านต่างคือ ๆ ตามความต้องการของผู้ใช้ที่สามารถนำมาใช้ประโยชน์ได้โดยตรง ทำให้การทำงานได้สะดวกขึ้น หรืออาจเป็นซอฟต์แวร์ใช้งานเฉพาะด้านซึ่งผู้ใช้เป็นผู้พัฒนาขึ้นเองเพื่อให้เหมาะสมกับสภาพการทำงานของตน สามารถแบ่งซอฟต์แวร์ประยุกต์ออกเป็นสองกลุ่มคือ ซอฟต์แวร์สำเร็จรูป คือ โปรแกรมที่ถูกสร้างหรือเขียนขึ้นมาเสร็จเรียบร้อยแล้ว พร้อมที่จะนำไปใช้งานต่างๆได้ทันที เช่น Microsoft Office เป็นต้น และซอฟต์แวร์ที่ใช้งานเฉพาะด้าน เช่น ซอฟต์แวร์ประมวลผลคำ ซอฟต์แวร์ระบบบัญชี เป็นต้น

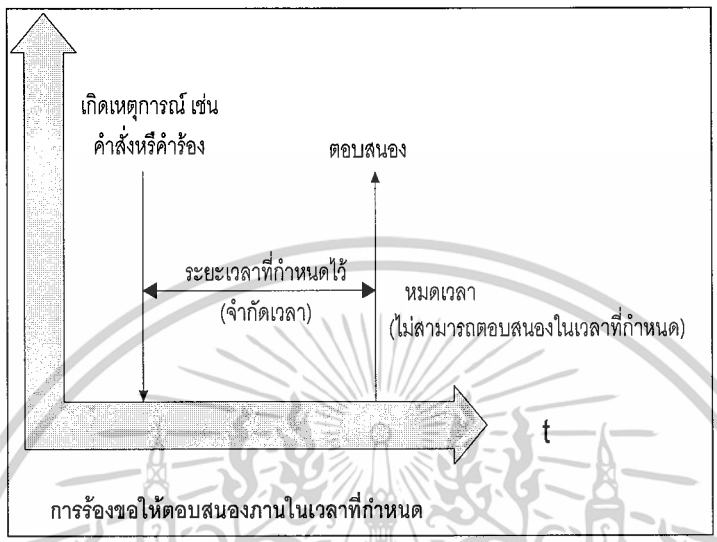
2.1.5 เทคโนโลยีที่เกี่ยวข้องกับระบบฝังตัว

เทคโนโลยีที่เป็นส่วนสนับสนุนการทำงานภายในระบบฝังตัวที่มีเอ็มพียูเป็นศูนย์กลางในการคำนวณและประมวลผลทั้งหมด ได้แก่

2.1.5.1 การประมวลผลแบบเรียลไทม์ (Real-time Processing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฝังตัวส่วนใหญ่จะมีการประมวลผลแบบเรียลไทม์ หรือการตอบสนองแบบทันที โดยเมื่อได้คำร้องขอการประมวลผลจากภายนอก จะต้องสามารถประมวลผลหรือตอบสนองต่อเหตุการณ์ต่าง ๆ ภายในเวลาที่กำหนด ดังรูปที่ 2.8



รูปที่ 2.8 การตอบสนองแบบเรียลไทม์(Japan System House Association. 2549. แปลและเรียบเรียง โดย ดร.ธนารัตน์ ธีระมั่นคง : 98)

การตอบสนองต่อเหตุการณ์ต่าง ๆ นั้น จะมีกรอบการจัดตารางเวลางาน (Task scheduling) ทำหน้าที่ในการประมวลผลเหตุการณ์ที่เกิดขึ้นพร้อม ๆ กัน โดยมีการจัดลำดับความสำคัญในการตอบสนองต่อเหตุการณ์ก่อนหลังได้ เรียกว่า การกำหนดค่าความสำคัญหรือไพรออริตี้ (Priority) หากเหตุการณ์ใดไม่ถูกตอบสนองภายในเวลาที่กำหนดจะเรียกว่า หมดเวลาหรือไทม์เอาต์ (Time out) หรือเลยเวลาหรือไทม์โอเวอร์ (Time over) ลักษณะการทำงานแบบนี้ทำให้เกิดระบบปฏิบัติการแบบเรียลไทม์หรือเรียลไทม์โอเอส (Real-time OS) เรียกย่อ ๆ ว่า RTOS ขึ้น

ระบบเรียลไทม์ สามารถแบ่งเป็น 2 ประเภท คือ ซอฟต์เรียลไทม์ (Soft Real-time) และฮาร์ดเรียลไทม์ (Hard Real-time)

1. ซอฟต์เรียลไทม์ เป็นระบบเรียลไทม์ที่ยอมให้เกิดความคลาดเคลื่อนของเวลาได้ แต่ไม่ได้ส่งผลกระทบต่อผู้ใช้ทั่วไป เช่น การบอกเวลาเข้าออกของเครื่องบิน สามารถคลาดเคลื่นได้น้อยกว่า 3 นาที ซึ่งเวลาที่คลาดเคลื่อนดังกล่าวไม่ส่งผลกระทบกับผู้โดยสารเลย หรือเป็นระบบที่สามารถรอให้งานอื่นเสร็จก่อนที่จะทำงานได้

2. ฮาร์ดเรียลไทม์ เป็นระบบเรียลไทม์ที่ต้องการความแม่นยำของเวลาสูงมากๆ คือไม่ต้องการให้เกิดความคลาดเคลื่อนของเวลาแม้แต่นิดเดียว ตัวอย่างเช่น การทำงานของอุปกรณ์อิเล็กทรอนิคค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วามเร็วสูงสองชิ้นที่มีการทำ Interface ติดต่อสื่อสารกัน หากอุปกรณ์ใดอุปกรณ์หนึ่งมีระบบเวลาที่ผิดพลาดแม้แต่เพียงนิดเดียว การติดต่อสื่อสารของอุปกรณ์สองชิ้นนั้นก็อาจจะคุยกันไม่ได้เลยก็ได้

2.1.5.2 เทคโนโลยีอินเทอร์รัพต์

เทคโนโลยีอินเทอร์รัพต์ (Interrupt technology) เป็นการเปลี่ยนการทำงานจากโปรแกรมหนึ่งซึ่งทำงานอยู่จนกระทั่งมีเหตุการณ์เกิดขึ้น และเหตุการณ์นั้นทำให้เกิดการเปลี่ยนการทำงานจากโปรแกรมเดิมไปอีกโปรแกรมหนึ่ง กรอบการทำงานของอินเทอร์รัพต์ จะมีกรอบการทำงาน ดังนี้

1. รับสัญญาณจากอุปกรณ์รอบข้าง
2. ทำตามกำหนดเวลาที่ตัวนับหรือเคาน์เตอร์ (Counter) ของโปรแกรมที่ตั้งไว้
3. รันโปรแกรมที่ตอบสนองต่อเหตุการณ์นั้น
4. กลับเข้าสู่โปรแกรมเดิมที่ถูกหยุดไว้ชั่วคราว

ขั้นตอนการอินเทอร์รัพต์นี้มักจะแตกต่างกันแต่ละเอ็มพียู ซึ่งเอ็มพียูจะมีการตรวจสอบการยอมรับอินเทอร์รัพต์ การตรวจสอบระดับของการอินเทอร์รัพต์ และการอ้างอิงค่าการอนุญาตในการอินเทอร์รัพต์ของเอ็มพียู ซึ่งการกำหนดค่าต่างๆ ของการอินเตอร์รัพต์จะถูกติดตั้งไว้ในอุปกรณ์ฮาร์ดแวร์แต่ละชิ้น

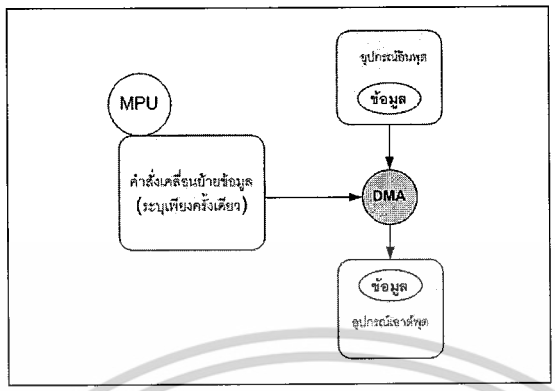
การอินเทอร์รัพต์แบ่งได้เป็น 2 ประเภทคือ อินเทอร์รัพต์ภายใน(Internal Interrupt) และ อินเทอร์รัพต์ภายนอก (External Interrupt)โดยอินเทอร์รัพต์ภายในประกอบด้วย อินเทอร์รัพต์คำสั่ง (command interrupt) และอินเตอร์รัพต์ยกเว้น (exception interrupt) ซึ่งจะเกิดขึ้นเมื่อพบเหตุการณ์ไม่ปกติหรือมีการใช้ฟังก์ชันเคอร์เนล ส่วนอินเทอร์รัพต์ภายนอกนั้นจะประกอบไปด้วยอินเทอร์รัพต์จากความผิดปกติ (abnormal interrupt) และอินเทอร์รัพต์จากอินพุต/เอาต์พุต นอกจากนี้ในการประมวลผลอาจมีการอินเทอร์รัพต์เกิดขึ้นซ้อนกันได้ เรียกว่า การอินเทอร์รัพต์แบบหลายชั้น (multiplex interruption) ดังนั้นจึงมีตัวจัดการอินเทอร์รัพต์ (interrupt handle) เข้ามาช่วยจัดการการทำงาน การควบคุมการอินเทอร์รัพต์ที่เกิดขึ้นแบบหลายชั้นนี้

2.1.5.3 เทคโนโลยีดีเอ็มเอ

เทคโนโลยีดีเอ็มเอ(DMA technology) เป็นการเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำและอุปกรณ์อินพุตเอาท์พุต หรือระหว่างหน่วยความจำด้วยกัน โดยใช้ฮาร์ดแวร์ควบคุมและไม่จำเป็นต้องใช้การประมวลผลด้วยโปรแกรมของเอ็มพียูดังรูปที่ 2.9 โดยอุปกรณ์ฮาร์ดแวร์ที่ทำหน้าที่เคลื่อนย้ายข้อมูลโดยตรงจากหน่วยความจำโดยไม่ผ่านเอ็มพียู เรียกว่าอุปกรณ์ช่องสัญญาณดีเอ็มเอ (DMA channel) เมื่อเป็นไอซี(IC) จะเรียกว่าดีเอ็มเอคอนโทรลเลอร์ (DMA controller)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีนี้จะช่วยให้สามารถเข้าถึงข้อมูลได้โดยตรง ช่วยลดเวลาในการประมวลผลของเอ็มพียูทำให้ระบบรวดเร็วมากขึ้น

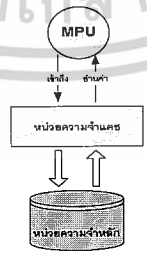


รูปที่ 2.9 การเคลื่อนย้ายข้อมูลด้วยวิธีการดีเอ็มเอ (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 99)

2.1.5.4 เทคโนโลยีหน่วยความจำแคช

เทคโนโลยีหน่วยความจำแคช (Cache memory technology) เป็นการติดตั้งหน่วยความจำที่มีขนาดเล็กแต่มีความเร็วสูงไว้ระหว่างเอ็มพียูกับหน่วยความจำหลัก โดยจะทำการคัดลอกข้อมูลจากหน่วยความจำหลักในบริเวณที่ใกล้เคียงกับที่เอ็มพียูเข้าถึงครั้งต่อไป เอ็มพียูจะอ่านข้อมูลจากหน่วยความจำนี้แทนการเข้าถึงหน่วยความจำหลักโดยตรง ซึ่งจะช่วยเพิ่มความเร็วของการเข้าถึงข้อมูลได้ดังรูปที่ 2.10 การทำงานของหน่วยความจำแคชจะมีรายละเอียดดังนี้

1. เอ็มพียูทำการอ่านข้อมูลจากหน่วยความจำแคช
2. เมื่อพบข้อมูลที่ต้องการจะทำการประมวลผลข้อมูลนั้น
3. หากไม่พบข้อมูลที่ต้องการจะคัดลอกข้อมูลจากหน่วยความจำหลักมาไว้ในหน่วยความจำแคช และใช้งานข้อมูลจากหน่วยความจำแคชแทน



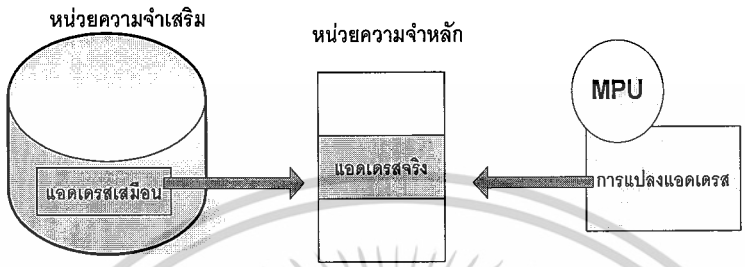
รูปที่ 2.10 การทำงานของหน่วยความจำแคช

2.1.5.5 เทคโนโลยีหน่วยความจำเสมือน

เทคโนโลยีหน่วยความจำเสมือน (Virtual memory technology) เป็นเทคโนโลยีที่ช่วยให้หน่วยความจำที่มีขนาดใหญ่ถูกเรียกใช้งานโดยระบบที่มีหน่วยความจำขนาดเล็กได้ เช่น โปรแกรมที่ถูกเก็บในฮาร์ดดิสก์ เป็นต้น เมื่อเอ็มพียูทำงานข้อมูลที่ถูกเก็บในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสริมหรือหน่วยความจำภายนอกนั้นจะถูกเรียกมาเก็บในหน่วยความจำหลัก โดยจะใช้หลักการในการสร้างแอดเดรสเสมือนที่มีขนาดใหญ่กว่าพื้นที่แอดเดรสจริงของระบบ จากนั้นทำการอ่านคำสั่งจากแอดเดรสในหน่วยความจำเสมือนแล้วแปลงแอดเดรสนี้ให้กลายเป็นแอดเดรสของหน่วยความจำจริง โดยผ่านฟังก์ชันการแปลงแอดเดรส แล้วจึงทำการอ่านข้อมูลเข้ามาประมวลผลในเอ็มพียู ดังรูปที่ 2.11



รูปที่ 2.11 ระบบหน่วยความจำเสมือน(Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง :101)

2.1.5.6 เทคโนโลยีประหยัดพลังงาน

เทคโนโลยีประหยัดพลังงาน (Energy saving technology) เป็นเทคโนโลยีที่เป็นลักษณะเฉพาะในระบบฝังตัว ชิปบางตัวจึงอาจออกแบบให้มีความสามารถพิเศษเฉพาะตัวอื่น ๆ เช่น ใช้แรงดันไฟฟ้าต่ำ ซึ่งฟังก์ชันลักษณะนี้มีประโยชน์มากกับระบบฝังตัวแบบพกพา ซึ่งต้องพึ่งพาพลังงานจากแบตเตอรี่

2.2 แนวทางการพัฒนาระบบ

เนื่องจากในปัจจุบันซอฟต์แวร์ถูกพัฒนาและมีการนำมาใช้เพิ่มขึ้นอย่างรวดเร็ว จนเป็นเทคโนโลยีที่มีความสำคัญในด้านต่าง ๆ จึงมีการคิดค้นแนวทางการพัฒนาตลอดจนเครื่องมือ(Tool) ต่างๆ มาใช้ในการพัฒนา เพื่อให้ได้ซอฟต์แวร์ที่มีคุณภาพรวมถึงลดปัญหาต่าง ๆ ในการพัฒนาได้

สำหรับระบบฝังตัวก็เช่นเดียวกันเนื่องจากค่อนข้างมีความซับซ้อน มีความหลากหลายในด้านฟังก์ชันการทำงานและมีความต้องการในการใช้งานที่ต่างกัน จึงจำเป็นต้องมีการใช้เทคโนโลยีทางด้านสถาปัตยกรรมฮาร์ดแวร์และการออกแบบซอฟแวร์ร่วมกัน ดังนั้นจึงต้องเลือกใช้วิธีการตลอดจนเครื่องมือต่าง ๆ อย่างเหมาะสม และเหมาะกับแนวโน้มในปัจจุบัน เช่น แนวคิดเชิงวัตถุ (Object-Oriented) การประยุกต์ใช้เครื่องมือ (tool) สมัยใหม่ อย่างเช่นยูเอ็มแอล เป็นต้น

2.2.1 วงจรในการพัฒนาระบบสารสนเทศ

โดยทั่วไประบบสารสนเทศต่าง ๆ จะมีวงจรของการพัฒนาระบบเรียกว่าวงจรการพัฒนาระบบ(System Development Lift Cycle: SDLC) คือกระบวนการในการพัฒนาระบบสารสนเทศเพื่อแก้ปัญหาต่าง ๆ และตอบสนองความต้องการของผู้ใช้ ดังรูปที่ 2.12 ซึ่งขั้นตอนหลัก ๆ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1: การกำหนดปัญหา (Identification)

เป็นการกำหนดปัญหาและขอบเขตของระบบที่ต้องการพัฒนา โดยทำการรวบรวมข้อมูลที่จำเป็น วิเคราะห์ความเป็นไปได้ วิเคราะห์ปัญหาและแนวทางในการแก้ปัญหา

ขั้นตอนที่ 2: การวางแผน (Planning)

เป็นการวางแผนในการพัฒนา โดยทำการวางแผนด้านทรัพยากรด้านต่าง ๆ การระบุขอบเขตของงาน ระยะเวลา เป็นต้น

ขั้นตอนที่ 3: การวิเคราะห์ (Analysis)

เป็นการวิเคราะห์ระบบ โดยนำข้อมูลที่มีมาทำการวิเคราะห์ปัญหาและหาวิธีการในการแก้ปัญหา โดยใช้เครื่องมือ (tools) ต่าง ๆ มาใช้เพื่อให้เข้าใจการทำงานมากขึ้นเช่น DFDs , UML

ขั้นตอนที่4 : การออกแบบ (Design)

เป็นการออกแบบระบบทั้งในส่วนที่เป็นโครงสร้างโปรแกรม หน้าจอเพื่อใช้ติดต่อกับUser การออกแบบรายงานที่ได้จากระบบ การออกแบบฐานข้อมูล เป็นต้น

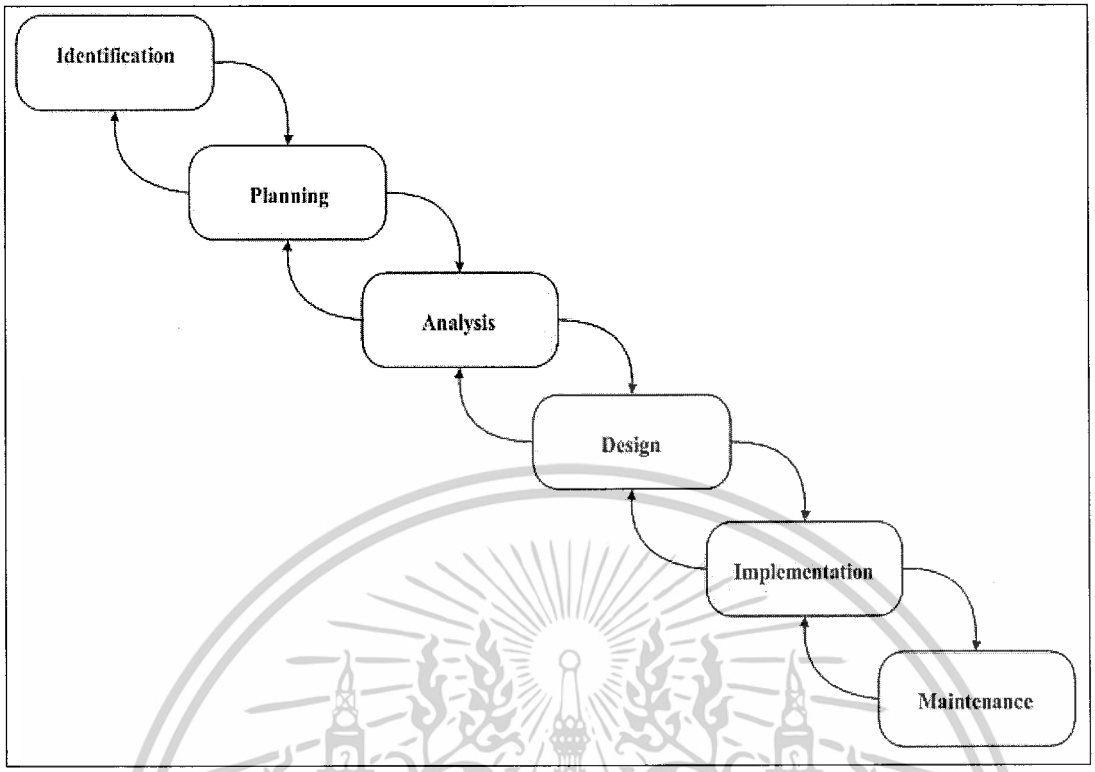
ขั้นตอนที่ 5: การจัดทำและสร้าง (Implementation)

เป็นการจัดทำและสร้างระบบ โดยทำการเขียนโปรแกรมตามขั้นตอนการทำงานที่ได้ออกแบบไว้ จากนั้นทำการทดสอบ รวมถึงการจัดทำเอกสารที่เกี่ยวข้องกับระบบและโปรแกรมที่พัฒนาขึ้นแล้วทำการติดตั้งระบบที่ได้พัฒนาขึ้นให้ผู้ใช้สามารถใช้งานได้จริง

ขั้นตอนที่ 6: การดูแลบำรุงรักษา (Maintenance)

เป็นการดูแลบำรุงรักษาระบบ เช่นการแก้ไขข้อบกพร่องหลังการใช้งานจริง การอบรมการใช้โปรแกรม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



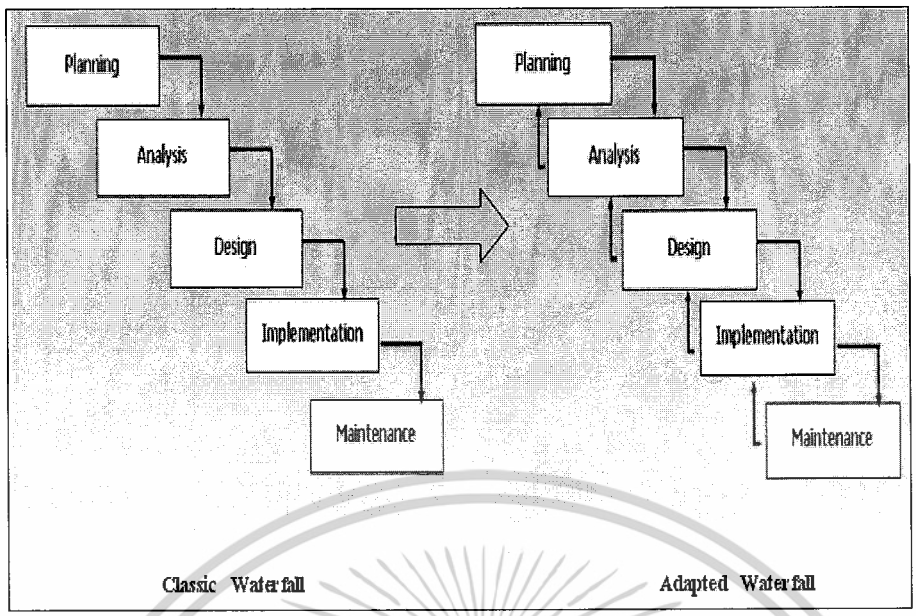
รูปที่2.12 วงจรในการพัฒนาระบบสารสนเทศ (อ. อดิศร ณ อุบล. 2550. http://learners.in.th/file/apealex/UML.ppt)

2.2.2 โมเดลการพัฒนาระบบ

โมเดลการพัฒนาระบบ (Software process models) ใช้เป็นตัวชี้ถึงกิจกรรมหลัก (Key Activities) ในการพัฒนาซอฟต์แวร์รวมถึงความสัมพันธ์ของส่วนอื่น ๆ แต่ละโมเดลจะระบุรายละเอียดและวิธีปฏิบัติของแต่ละกิจกรรมไว้ การใช้โมเดลในการพัฒนาระบบ จะช่วยให้การพัฒนาเป็นไปอย่างถูกต้องและรวดเร็วมากขึ้น และช่วยลดความเสี่ยงที่อาจเกิดขึ้นได้ ปัจจุบันได้มีการเสนอโมเดลในการพัฒนาที่หลากหลาย เช่น แบบจำลองน้ำตก (Water Fall Model) แบบจำลองแบบก้นหอย (Spiral Model) แบบจำลองแบบค่อยเป็นค่อยไป (Incremental Model) แบบจำลองต้นแบบ (Prototyping Model) เป็นต้น แต่ที่นิยมใช้กันมากในปัจจุบันคือแบบจำลองเชิงวัตถุ (Object-Oriented model) ซึ่งเป็นโมเดลที่ใช้แนวคิดเชิงวัตถุในการพัฒนา

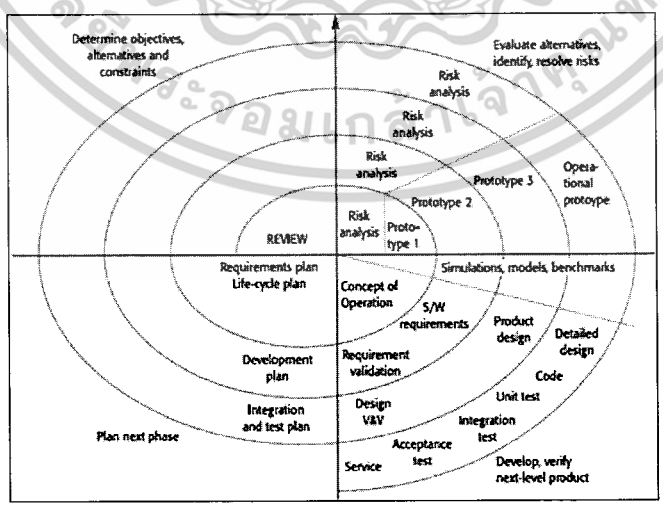
1. แบบจำลองน้ำตก เป็นโมเดลที่มีการใช้งานมานานมาก โมเดลนี้จะมีการดำเนินตามขั้นตอนอย่างเรียงลำดับจากบนลงล่างเปรียบเสมือนน้ำตก คือทำไปทีละขั้น แต่ละขั้นตอนต้องทำเสร็จก่อนจึงทำขั้นตอนถัดไปได้ เรียกโมเดลลักษณะนี้ว่าแบบจำลองน้ำตกแบบดั้งเดิม (Classic Waterfall Model) ต่อมาได้มีการปรับปรุงให้มีการย้อนการทำงานกลับไปยังขั้นตอนก่อนหน้าได้ ซึ่งเรียกว่าแบบจำลองน้ำตกแบบปรับปรุง (Adapted Waterfall Model) ดังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แบบจำลองน้ำตกแบบดั้งเดิมและแบบจำลองน้ำตกแบบปรับปรุง
(อ. อดิศร ณ อุบล.2550. http://learners.in.th/file/apealex/UML.ppt)

2. แบบจำลองแบบก้นหอย (Spiral Model) พัฒนาโดย Boehm ในค.ศ 1988 มีลักษณะการทำงานแบบวนรอบ แต่ละรอบจะประกอบด้วยขั้นตอนต่าง ๆ ดังนี้ การวางแผน (Planning) การวิเคราะห์ความเสี่ยง (Risk Analysis) การพัฒนา (Develop / Engineering) และการประเมิน (Evaluation) ข้อดีของการทำงานลักษณะนี้คือช่วยลดความเสี่ยง เนื่องจากในการทำงานแต่ละรอบจะได้ต้นแบบออกมา และตรวจสอบกับความต้องการได้ เหมาะกับงานที่มีความเสี่ยงสูง หรือเปลี่ยนแปลงความต้องการบ่อย โดยแบบจำลองแบบก้นหอยจะมีลักษณะดังรูปที่ 2.14

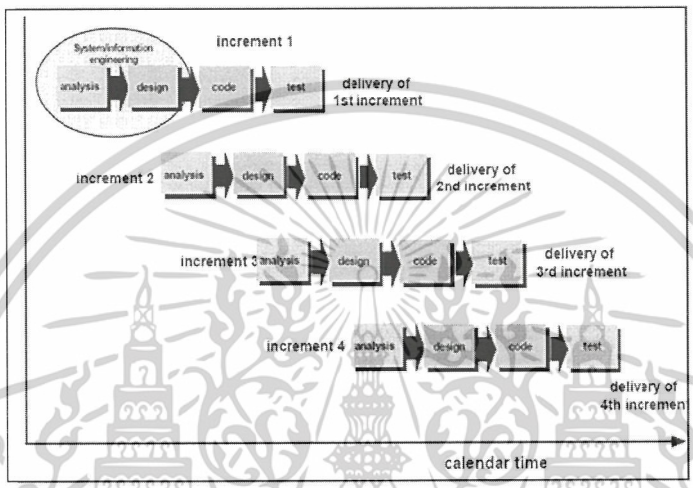


รูปที่ 2.14 แบบจำลองแบบก้นหอย (อ. อดิศร ณ อุบล. 2550.
http://learners.in.th/file/apealex/UML.ppt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

3. แบบจำลองแบบเพิ่มขึ้น (Incremental Model) มีรูปแบบการทำหลายรอบ โดยจะแบ่งงานออกเป็นงานย่อยๆ และเรียกงานย่อยๆ นั้นว่า Increment แล้วจึงค่อยพัฒนางานให้สำเร็จทีละ Increment ไป โดยเราจะทำการพัฒนางานที่เป็นแกนของโครงการก่อน แล้วค่อยพัฒนา Increment ย่อยอื่นๆ ข้อดีของโมเดลนี้คือทำให้สามารถส่งมอบงานได้รวดเร็ว สามารถส่งงานได้เป็นส่วน ๆ และลดความเสี่ยงที่อาจเกิดขึ้นจากความเปลี่ยนแปลงได้ โดยแบบจำลองแบบเพิ่มขึ้นจะมีลักษณะการแบ่งงานดังรูปที่ 2.15



รูปที่2.15 แบบจำลองแบบเพิ่มขึ้น(อ. อดิศร ณ อุบล. 2550. http://learners.in.th/file/apealex/UML.ppt)

4. แบบจำลองต้นแบบ (Prototyping Model) เป็นการทำต้นแบบ (Prototype) ของซอฟต์แวร์ออกมาก่อนซึ่งต้นแบบนี้จะมีรูปร่างที่เสมือนจริง ทำให้สามารถทราบได้ว่างานที่พัฒนาตรงกับความต้องการหรือไม่ วิธีนี้เหมาะกับงานที่เปลี่ยนแปลงบ่อย ช่วยลดความเสี่ยงและช่วยให้พัฒนาระบบที่ตรงกับความต้องการ สำหรับการพัฒนาตัวต้นแบบสามารถทำได้หลายลักษณะ เช่นตัวต้นแบบเฉพาะหน้าจอ (Prototype the Screens only) ตัวต้นแบบเฉพาะการประมวลผล (Prototype the Processing Procedures Only) เป็นต้น

5. แบบจำลองเชิงวัตถุ (Object Oriented Model) ใช้หลักการของแนวคิดเชิงวัตถุ ข้อดีของโมเดลนี้จะช่วยในด้านของการนำกลับมาใช้ใหม่ (Reuse) ช่วยให้การพัฒนาเป็นไปได้อย่างรวดเร็ว หากมีการเปลี่ยนแปลงแก้ไขก็สามารถกระทำได้ง่าย การนำแนวคิดเชิงวัตถุจะใช้โมเดลหรือแบบจำลองเชิงวัตถุที่มีภาษามาตรฐานในการอธิบายการทำงานหรือเป็นเครื่องมือ(Tool) ที่ช่วยในการวิเคราะห์และออกแบบระบบที่เรียกว่า ยูเอ็มแอล (Unified Modeling Language :UML) โดยยูเอ็มแอลเป็นแผนภาพที่ใช้ในการอธิบายระบบและการทำงาน ทำให้มองเห็นภาพระบบได้ชัดเจนขึ้น ช่วยลดขั้นตอนในการออกแบบและพัฒนา โดยที่ไม่ต้องยึดติดกับภาษาใดภาษาหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
04778

2.2.3 วิธีการพัฒนาระบบสารสนเทศ

วิธีการพัฒนาระบบสารสนเทศมี 2 แบบ คือ เชิงโครงสร้าง (Structure Analysis and Design - SAD) และ เชิงวัตถุ (Object-Oriented Analysis and Design - OOAD)

2.2.3.1 การวิเคราะห์และออกแบบระบบเชิงโครงสร้าง

การวิเคราะห์และออกแบบระบบเชิงโครงสร้าง เป็นวิธีการแบบดั้งเดิมที่เน้นความสำคัญของข้อมูลเป็นหลัก โดยจะพิจารณากระบวนการเปลี่ยนแปลงข้อมูลเพื่อนำไปแสดงผลหรือจัดเก็บข้อมูล โดยมีกระบวนการในการประมวลผลข้อมูลพื้นฐาน โดยเฉพาะอย่างยิ่งในงานด้านธุรกิจ ซึ่งจะเป็นการทำงานลักษณะการจัดการแฟ้มข้อมูล (File Processing)

การพัฒนาโปรแกรมแบบโครงสร้างจะมีลักษณะเป็นบน-ล่าง (TOP/DOWN) การเปลี่ยนแปลงแก้ไขข้อมูลใดๆจะกระทำได้แต่ค่อนข้างยากเนื่องจากติดยึดกับโครงสร้างข้อมูลหนึ่งๆ ซึ่งหากมีการเปลี่ยนแปลงใด ๆ จะต้องกลับไปปรับปรุงจากขั้นตอนแรกใหม่หมด เป็นผลให้เกิดความล่าช้า เพิ่มเวลาและค่าใช้จ่ายในการพัฒนาเป็นอย่างยิ่ง เครื่องมือ (tool) ต่าง ๆ ที่ใช้ในการวิเคราะห์และออกแบบระบบแบบโครงสร้างได้แก่ Data Flow Diagrams (DFDs) Context Diagram Structure Chart System Flowchart เป็นต้น

2.2.3.2 การวิเคราะห์และออกแบบระบบเชิงวัตถุ

การวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design: OOAD) เป็นแนวคิดการพัฒนาแบบใหม่ เพื่อมุ่งหวังในเรื่องของการนำส่วนประกอบต่างๆ ของระบบกลับมาใช้ใหม่ การปรับปรุงในด้านของประสิทธิภาพของระบบ สนับสนุนการพัฒนาระบบที่ซับซ้อนและสิ่งสำคัญคือการวิเคราะห์และออกแบบระบบในเชิงวัตถุจะทำให้ได้ปริมาณของงานที่เพิ่มมากขึ้นในระยะของการพัฒนาที่เท่าเดิม สำหรับระบบฝังตัวก็เช่นเดียวกันได้มีการประยุกต์แนวคิดเชิงวัตถุมาใช้งานและมีแนวโน้มที่จะเพิ่มขึ้นอย่างรวดเร็ว

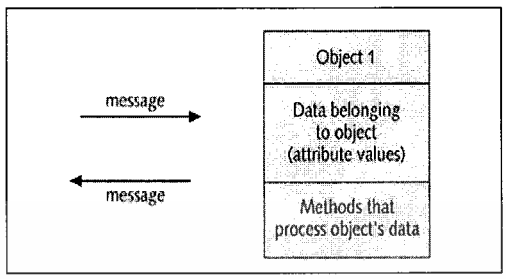
แนวคิดเชิงวัตถุ (Object Oriented) หมายถึง การใช้อ็อบเจ็ค (Object) เป็นตัวหลักในการพิจารณาความเป็นจริงต่างๆ ที่เกิดขึ้นในโลก โดยมองทุกสิ่งในโลกเป็นวัตถุและมองกิจกรรมที่เกิดขึ้นในโลกนี้เกิดจากความสัมพันธ์และปฏิสัมพันธ์ระหว่างวัตถุ อ็อบเจ็กต์ใช้เป็นตัวแทนสิ่งที่จับต้องได้ (Tangible) และจับต้องไม่ได้ (Intangible) เช่นสิ่งของ คน สถานที่ เหตุการณ์ เป็นต้น

วัตถุแต่ละอันจะประกอบไปด้วยคุณลักษณะ(Attribute) หน้าที่การทำงาน (Operation) และความสัมพันธ์ (Relationship) โดยมีการการสื่อสารระหว่างอ็อบเจ็กต์ เรียกว่าเมจเสจ (Message) ซึ่งการแสดงความสัมพันธ์และการทำงานของอ็อบเจ็คในรูปแบบของคลาส (Class)

คลาส คือการนิยามกลุ่มของคุณสมบัติและกลุ่มของพฤติกรรมขึ้น เพื่อกำหนดโครงสร้างให้แก่อ็อบเจ็คของคลาสนั้น ๆ เมื่ออ็อบเจ็คถูกสร้างจากคลาสดังกล่าว ก็จะทำการสร้างโครงสร้างทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคุณสมบัติและพฤติกรรมมาให้แก่อ็อบเจ็คนั้น ๆ กล่าวได้ว่าคลาสคือแม่แบบของอ็อบเจ็คนั่นเอง โดยกระบวนการเกิดอ็อบเจ็คจากคลาสนั่นจะเรียกว่าการอินสแตนซ์ (Instance) ซึ่งคลาสจะประกอบด้วยส่วนต่างๆดังรูปที่ 2.16

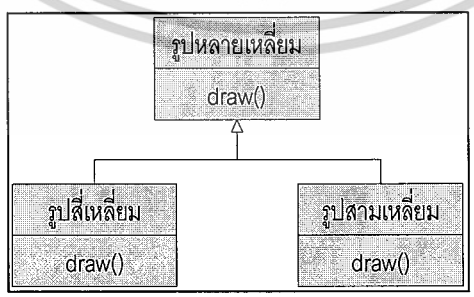


รูปที่ 2.16 Object-Oriented approach (อ. อดิศร ณ อุบล. 2550. http://learners.in.th/file/apealex/UML.ppt)

หลักการของแนวคิดเชิงวัตถุจะมีคุณสมบัติเด่น3แบบ ได้แก่ การมีได้หลายรูปแบบ (Polymorphism) การปกปิดข้อมูล (Encapsulation) การสืบทอด (Inheritance)

1. การมีได้หลายรูปแบบ คือ การที่ออบเจ็คที่ต่างกันมีปฏิกิริยาตอบสนองต่อ Function/ Message หนึ่งๆในวิธีที่ต่างกัน ในเชิงโปรแกรมมิ่งจะเป็นการที่ตัวแปรออบเจ็กต์ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปแบบไปจากคลาสเดิมได้ หลักการมีได้หลายรูปแบบ ช่วยให้สามารถนำโค้ด (code) กลับมาใช้ใหม่ ได้ เนื่องจากสามารถกำหนดชุดคำสั่งทั่วไป และมอบหน้าที่รายละเอียดของการนำไปใช้แก่อ็อบเจ็คที่เกี่ยวข้องจัดการ

ตัวอย่างเช่น อ็อบเจ็ครูปสี่เหลี่ยมกับอ็อบเจ็ครูปสามเหลี่ยมที่ได้รับการสืบทอดคุณสมบัติมาจากอ็อบเจ็ครูปหลายเหลี่ยมทั้งคู่อาจมีโอเปอเรชั่นที่มีชื่อว่า Draw() เหมือนกันแต่เมื่อมีการเรียกใช้โอเปอเรชั่นดังกล่าวของออบเจ็กต์รูปสี่เหลี่ยมจะมีการวาดรูปสี่เหลี่ยม ซึ่งต่างกับการเรียกโอเปอเรชั่น draw() ของรูปสามเหลี่ยมซึ่งจะเป็นการวาดรูปสามเหลี่ยม ดังรูปที่ 2.17



รูปที่ 2.17 ตัวอย่าง Polymorphism (อ. อดิศร ณ อุบล. 2550. http://learners.in.th/file/apealex/UML.ppt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การปกปิดข้อมูล หมายถึงการรวบรวมโอเปอเรชั่น (Operation) และแอตทริบิวต์ (Attribute) เข้าเป็นหน่วยเดียวกันเพื่อให้แอตทริบิวต์ (Attribute) ถูกเปลี่ยนแปลงได้อย่างเหมาะสมโดยผ่านโอเปอเรชั่น การเขียนโปรแกรมในลักษณะนี้จะมีการซ่อนข้อมูลที่ต้องการควบคุมความถูกต้องของข้อมูลไว้ (Information Hiding) และบังคับให้อ็อบเจ็คอื่นเข้าถึงข้อมูลที่ซ่อนไว้ผ่านทางอินเตอร์เฟส (Interface) ที่เตรียมไว้ ซึ่งเป็นกลไกในการปกป้องกันข้อมูลของอ็อบเจ็ค

3. การสืบทอด เป็นการสืบทอดคุณสมบัติของอ็อบเจ็ค หรือการที่คลาสหนึ่งสืบทอดคุณสมบัติและพฤติกรรมจากอีกคลาสหนึ่ง โดยเรียกคลาสที่สืบทอดมาว่าคลาสลูก (Subclass) และคลาสที่ถ่ายทอดเรียกว่าคลาสแม่ (Super class) ดังนั้นคลาสลูกจะมีองค์ประกอบคือคุณลักษณะ และพฤติกรรมทุกอย่างที่คลาสแม่มี นั่นคือคลาสลูกสามารถทำหรือแสดงทุกสิ่งทุกอย่างที่คลาสแม่ ทำได้ นอกจากนั้นแล้วอาจมีความสามารถมากกว่าคลาสแม่ได้ด้วย เช่นมีข้อมูลคุณลักษณะเพิ่มขึ้น มีพฤติกรรมในการทำงานมากยิ่งขึ้น หรือมีการปรับปรุงพฤติกรรมให้ทำงานได้ดีขึ้น ข้อดีของการสืบทอดคุณสมบัติจะทำให้พัฒนาโครงสร้างอย่างเป็นระบบ ปรับเปลี่ยนได้ง่าย และลดเวลาในการพัฒนา นอกจากคุณสมบัติเด่นดังกล่าวแล้วแนวคิดเชิงวัตถุยังมีข้อดีอื่นอีก เช่น การนำกลับมาใช้ใหม่ (reuse) ความเป็นอิสระต่อการนำไปใช้และการปรับเปลี่ยนเวอร์ชัน (modularity) ฯลฯ

2.2.4 ยูเอ็มแอล

ยูเอ็มแอล(UML :Unified Modeling Language) เป็นภาษาสัญลักษณ์รูปภาพ (Graphical Language) สำหรับใช้ในการสร้างโมเดลเชิงวัตถุ ใช้เพื่อถ่ายทอดความคิดที่มีต่อระบบออกมาเป็นแผนภาพ ซึ่งประกอบไปด้วยรูปภาพหรือสัญลักษณ์ตามกฎในการสร้างแผนภาพนั้นยูเอ็มแอลเป็นภาษาสำหรับสร้างแบบจำลองของระบบ " ที่ได้รับการพัฒนาขึ้นมาเพื่อใช้ในการวิเคราะห์และออกแบบระบบเชิงวัตถุ โดยจะมีคุณสมบัติของการวนซ้ำและการเพิ่มเติมในแต่ละรอบการทำงาน (Incremental Iterative Approach) เพื่อขยายต่อเติมฟังก์ชันของระบบจนกว่าจะได้ระบบที่สมบูรณ์

ข้อดีของยูเอ็มแอล

1. เป็นภาษารูปภาพมาตรฐาน สามารถสื่อความหมายและสร้างความเข้าใจได้ทุกขั้นตอนของการพัฒนา
2. สามารถนำเสนอและสนับสนุนหลักการเชิงวัตถุได้อย่างชัดเจน ทำให้นักพัฒนาระบบสามารถทำความเข้าใจกับปัญหาและค้นพบวิธีแก้ไขได้อย่างรวดเร็วและง่ายยิ่งขึ้น
3. ไม่ผูกติดกับภาษาโปรแกรมภาษาใดภาษาหนึ่ง สามารถถูกแปลงไปเป็นระบบจริงที่ถูกสร้างขึ้นด้วยภาษาโปรแกรมเชิงวัตถุใด ๆ ก็ได้
4. เป็นภาษาที่ง่ายต่อการทำความเข้าใจ สามารถถูกแปลงเป็นภาษาที่ใช้ในการสร้างระบบขึ้นจริงได้อย่างอัตโนมัติ จึงเป็นการช่วยลดภาระด้านเวลา และค่าใช้จ่ายได้

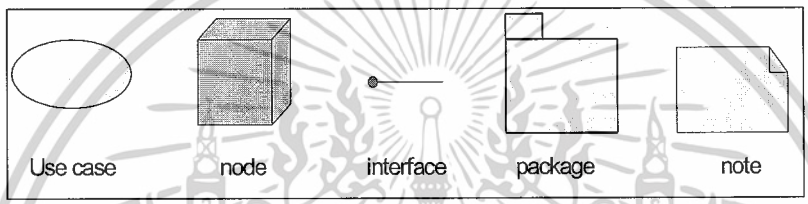
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สนับสนุนการขยายปรับปรุงระบบหรือการนำกลับมาใช้ใหม่

2.2.5 องค์ประกอบของยูเอ็มแอล

ยูเอ็มแอลประกอบไปด้วย 3 ส่วนใหญ่ ๆ คือ

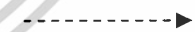
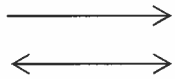
1. สัญลักษณ์พื้นฐาน เป็นสัญลักษณ์ที่ถูกใช้งานในการสร้างไดอะแกรมต่าง ๆ ของยูเอ็มแอล แบ่งเป็นประเภทต่าง ๆ ได้แก่
   - หมวดโครงสร้าง ได้แก่ สัญลักษณ์ยูสเคส อินเทอร์เฟซ โหนด (node) เป็นต้น
   - หมวดพฤติกรรม ได้แก่ อินเตอร์แอ็กชั่น สเตตแมชชีน เป็นต้น
   - หมวดการจัดกลุ่มใช้รวบรวมองค์ประกอบต่างๆ เช่นแพ็กเกจ (package) เป็นต้น
   - หมวดคำอธิบายประกอบ ได้แก่ โน๊ต (note) เป็นต้น



รูปที่ 2.18 ตัวอย่างสัญลักษณ์ต่าง ๆ ที่ใช้ในไดอะแกรม

2. ความสัมพันธ์ (Relationship) ที่ใช้ในยูเอ็มแอล ประกอบไปด้วยรูปแบบความสัมพันธ์ต่าง ๆ ดังสรุปได้จากตารางต่อไปนี้

ตารางที่ 2.1 รูปแบบความสัมพันธ์ที่ใช้ในยูเอ็มแอล

ความสัมพันธ์	คำอธิบาย	สัญลักษณ์
ความสัมพันธ์แบบพึ่งพา (Dependency Relationship)	เป็นความสัมพันธ์แบบขึ้นอยู่ต่อกัน คือเมื่อเกิดการเปลี่ยนแปลงที่ส่วนหนึ่งแล้วจะส่งผลกระทบถึงอีกส่วน	
ความสัมพันธ์แบบเกี่ยวพัน (Association Relationship)	เป็นความสัมพันธ์ระหว่างกันของสองอ็อบเจ็ค เช่นการรับ-ส่งข้อมูลระหว่างกัน อาจเป็นความสัมพันธ์แบบทางเดียวหรือสองทาง	
ความสัมพันธ์แบบสืบทอด (Generalization Relationship)	เป็นความสัมพันธ์ในลักษณะที่มีการสืบทอดคุณสมบัติของอ็อบเจ็ค	

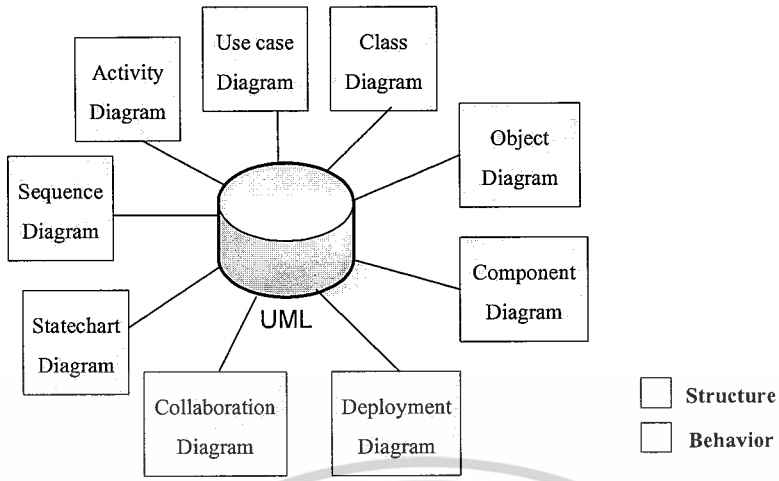
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไดอะแกรมต่าง ๆ ในยูเอ็มแอลประกอบไปด้วยไดอะแกรมต่างๆ สามารถสรุปการใช้งาน ดังตารางที่ 2.2 และสามารถแบ่งได้เป็นสองประเภทหลัก ๆ ได้แก่ เชิงโครงสร้าง (Structure diagram) และเชิงพฤติกรรม (Behavior diagram) ดังรูปที่ 2.19

ตารางที่ 2.2 ไดอะแกรมต่าง ๆ ของยูเอ็มแอลและการใช้งาน

ไดอะแกรม	การใช้งาน
ยูสเคสไดอะแกรม (Use case Diagram)	แสดงฟังก์ชันการทำงานของระบบ โดยพิจารณาจากความสัมพันธ์ของระบบและสิ่งที่อยู่นอกระบบด้วย
แอคทิวิตี้ไดอะแกรม (activity diagram)	แสดงลำดับกิจกรรมในระบบ ใช้หลักการเดียวกับโฟลว์ชาร์ต (Flow chart)
คลาสไดอะแกรม (Class diagram)	แสดงคลาสต่าง ๆ ที่จำเป็นในระบบ และแสดงความสัมพันธ์ระหว่างคลาส
อ็อบเจ็คไดอะแกรม (object diagram)	ความสัมพันธ์ระหว่างอ็อบเจ็ค ซึ่งเป็นอินสแตนซ์ (instance) ของคลาส
ซีเควนไดอะแกรม (sequence diagram)	แสดงปฏิสัมพันธ์ระหว่างอ็อบเจ็ค โดยดูจากมุมมองของเวลา (time) และดูลำดับความสัมพันธ์ระหว่างการทำงานร่วมกัน
คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram)	แสดงปฏิสัมพันธ์ระหว่างออปเจกต์โดยดูจากมุมมองของสเปซ (space) หรือตามลักษณะการทำงาน
สเตตชาร์ตไดอะแกรม (state chart diagram)	แสดงถึงสถานะของอ็อบเจ็คในระหว่างการทำงาน หรือแสดงการเปลี่ยนสถานะของอ็อบเจ็คเมื่อมีเหตุกาณ์มากระตุ้น
คอมโพเนนต์ไดอะแกรม (Component Diagram)	แสดงคอมโพเนนต์หรือส่วนประกอบย่อยของซอฟต์แวร์ต่างๆ ที่มีในระบบ
ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)	ใช้แสดงการติดตั้งใช้งานส่วนประกอบต่างๆ ของระบบทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



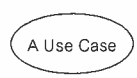
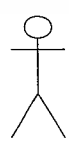
รูปที่ 2.19 ประเภทของไดอะแกรม (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. 23)

2.2.6 ไดอะแกรมต่าง ๆ ของยูเอ็มแอล

2.2.6.1 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรม เป็นแบบจำลองเพื่ออธิบายหน้าที่ของระบบ โดยพิจารณาจากมุมมองของผู้ใช้ระบบซึ่งจะเกิดขึ้นในขั้นตอนแรกๆ ของการพัฒนาระบบได้แก่ ขั้นตอนการวิเคราะห์ความต้องการของผู้ใช้และใช้เป็นพื้นฐานการสร้างแบบจำลองอื่นโดยที่ยูสเคสไดอะแกรมเป็นไดอะแกรมที่ทำหน้าที่ช่วยในการพิจารณาความต้องการของระบบ เพื่อแสดงถึงขอบเขตของการทำงาน

ตารางที่ 2.3 ส่วนประกอบของยูสเคสไดอะแกรม

องค์ประกอบ	คำอธิบาย	สัญลักษณ์
1. Use Case	คือหน้าที่แต่ละหน้าที่ที่ระบบจะต้องปฏิบัติ แต่ละยูสเคสแทนชุดของทรานแซคชั่น (transactions) ที่ระบบทำงานโต้ตอบกับ ผู้ใช้งาน หรือระบบอื่นๆ ภายนอก	
2. Actor	คือ ผู้ที่กระทำกับระบบ อาจเป็นผู้ที่ทำการรับและส่งข้อมูลหรือแลกเปลี่ยนข้อมูลกับระบบ ได้แก่ คน อุปกรณ์ต่างๆ เช่นเซ็นเซอร์ แอคทูเอเตอร์ หรือเป็นระบบอื่นๆ เป็นต้น โดยกำหนดตามบทบาทหน้าที่ที่กระทำต่อระบบ	
3. System Boundary	เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ ใช้สี่เหลี่ยมเป็นสัญลักษณ์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

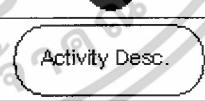
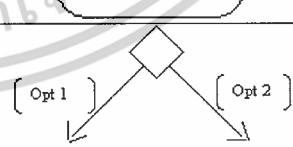
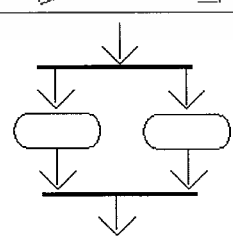
ตารางที่ 2.3 (ต่อ) ส่วนประกอบของยูสเคสไดอะแกรม

องค์ประกอบ	คำอธิบาย	สัญลักษณ์
4. Relationship	คือ ความสัมพันธ์ที่มีในยูสเคสไดอะแกรม - Association เป็นความสัมพันธ์ที่มีการติดต่อสื่อสารกัน ทั้งการรับ และส่ง messages ให้แก่กันและกัน - Generalization อธิบายความสัมพันธ์ ระหว่าง actors เป็นการถ่ายทอดคุณสมบัติกัน - Extends เป็นความสัมพันธ์ในกรณีที่ Use Case หนึ่งๆ ขยาย (extends) Use Case อื่น โดยการเพิ่มการกระทำ - Includes/Uses เป็นความสัมพันธ์ในกรณีที่ Use Case หนึ่งๆ เรียกใช้ (uses) Use Case อื่น ที่พิจารณาให้เป็นส่วนหนึ่งของ Use Case นั้นๆ	association generalization <<extend>> <<include>>

2.2.6.2 แอคทิวิตี้ไดอะแกรม

แอคทิวิตี้ไดอะแกรม เป็นแผนภาพที่แสดงให้เห็นลำดับการดำเนินกิจกรรม (Activity) จากกิจกรรมหนึ่งไปกิจกรรมหนึ่งภายในระบบที่เกิดจากการทำงาน ลักษณะของแผนภาพจะคล้ายโฟว์ชาร์ต (Flow Chart) โดยจะแสดงให้เห็นลำดับของกิจกรรมต่างๆ ที่เกิดขึ้นของระบบ

ตาราง 2.4 สัญลักษณ์ Activities Diagram

ส่วนประกอบ	Symbol
Initial Activity : จุดเริ่มต้น	
Activity: กิจกรรมที่เกิดขึ้น	
Decisions: ทางเลือกการตัดสินใจ	
Concurrent Activities: การทำงานพร้อมกัน	
Final Activity: จุดสิ้นสุด	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6.3 คลาสไดอะแกรม

คลาสไดอะแกรม เป็นแบบจำลองที่สร้างขึ้นเพื่ออธิบายโครงสร้างของระบบ โดยใช้ในการแสดงกลุ่มของคลาส โครงสร้างของคลาส อินเตอร์เฟส (Interface) และแสดงความสัมพันธ์ (Relationship) ระหว่างคลาส

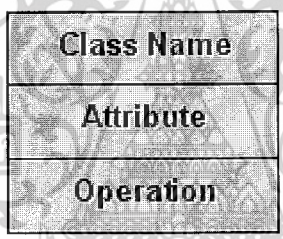
คลาส คือกลุ่มของอ็อบเจ็คที่มีโครงสร้างพื้นฐานเดียวกันคลาส เปรียบเสมือนพิมพ์เขียวของอ็อบเจ็ค โดยที่คลาสจะไม่สามารถทำงานได้ ถือเป็นนามธรรม(Abstract) ที่ใช้ในการนิยามอ็อบเจ็ค แต่อ็อบเจ็คสามารถทำงานได้ การทำงานของอ็อบเจ็คจะเป็นไปตามคุณสมบัติที่กำหนดไว้ในคลาส และอ็อบเจ็คทุกตัวต้องอยู่ในคลาสโดยสัญลักษณ์ที่ใช้ในคลาสไดอะแกรมดังรูปที่ 2.20

สัญลักษณ์ Class ประกอบด้วย

Class Name คือ ชื่อของ Class

Attributes คือ คุณลักษณะของ Class

Operations คือ กิจกรรมที่สามารถกระทำกับ Object นั้นๆ



รูปที่ 2.20 สัญลักษณ์ที่ใช้ในคลาสไดอะแกรม (อ. อดิศร ณ อุบล. 2550. http://learners.in.th/file/apealex/UML.ppt)

ความสามารถในการมองเห็น (Visibility) หรือการเข้าถึงของคลาส มีอยู่ 3 รูปแบบ คือ

1. สัญลักษณ์ (+) แสดงความเป็น Public หมายความว่า (Attribute) และเมดธอด (Method) นั้นสามารถมองเห็นได้จากอ็อบเจ็กต์อื่น
2. สัญลักษณ์ (-) แสดงความเป็นPrivate หมายความว่า แอตทิบิวต์และเมดธอดนั้นสามารถมองเห็นได้เฉพาะภายในอ็อบเจ็กต์เดียวกัน หรือเรียกใช้ภายในอ็อบเจ็คเดียวกันเท่านั้น
3. สัญลักษณ์ (#) แสดงความเป็น Protected หมายความว่าแอตทิบิวต์หรือเมดธอดนี้จะยอมให้อ็อบเจ็คภายในแพ็คเกจ (Package) เท่านั้นที่จะสามารถเข้าถึงแอตทิบิวต์หรือเมดธอดนี้ได้ ซึ่งแพ็คเกจคือกลุ่มของคลาสที่ได้รวมไว้ด้วยกันเป็นกลุ่มๆ โดยผู้ออกแบบระบบหรือผู้พัฒนาระบบ

ความสัมพันธ์ (Relationship) ภายในคลาสไดอะแกรมมี 4 แบบ ดังนี้

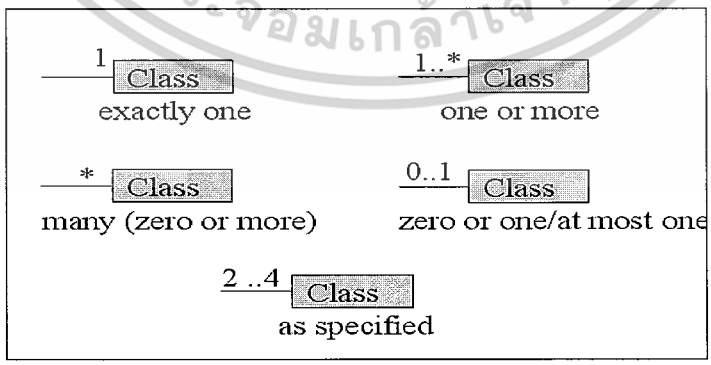
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Association จะแสดงความสัมพันธ์ระหว่างคลาส จะมีได้ทั้งทางเดียว และ 2 ทาง
2. Aggregation เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ระหว่าง Whole และ Parts ของมัน คือ Wholeจะประกอบไปด้วย parts ต่างๆ ของมัน การดงอยู่ของparts จะต้องขึ้นกับ Whole หรือจะเรียกความสัมพันธ์แบบนี้ว่า Whole-part
3. Dependency ใช้แสดงความสัมพันธ์กันระหว่างคลาส 2 คลาส ในแง่ที่คลาสหนึ่งเรียกใช้บริการของอีกคลาสหนึ่ง
4. Generalization แสดงความสัมพันธ์ระหว่างคลาสกับคลาสในแง่ที่คลาสหนึ่งถ่ายทอดคุณสมบัติและโครงสร้างจากอีกคลาสหนึ่ง

ตาราง 2.5 สัญลักษณ์แสดงความสัมพันธ์ในคลาสไดอะแกรม

Relationship	สัญลักษณ์
Association	
Aggregation	
Dependency	
Generalization	

การพิจารณาจำนวนอินสแตนซ์(Instances) ของคลาสหรือที่เรียกว่ามัลติพิซิตี้ (Multiplicity) หมายถึง การพิจารณาจำนวนอินสแตนซ์หรือจำนวนการเกิดอ็อบเจ็คจากคลาสหนึ่งซึ่งเชื่อมโยงกับคลาสที่เกี่ยวข้อง สามารถแสดงรูปแบบของการเขียนสัญลักษณ์ได้รูปที่ 2.21



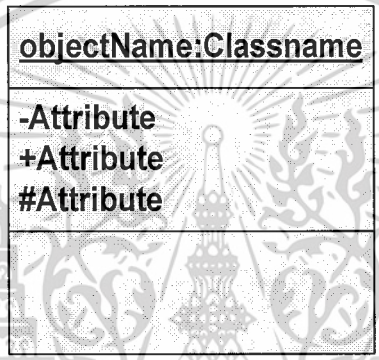
รูปที่ 2.21 multiplicity (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548: 25)

2.2.6.4 อ็อบเจ็คไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ็อบเจ็คไดอะแกรมเป็นแผนภาพที่ใช้แสดงกลุ่มของออบเจ็กต์และความสัมพันธ์ (Relationship) ระหว่างออบเจ็กต์ที่เกิดขึ้นในคลาสต่างๆ ของคลาสไดอะแกรม โดยจะประกอบด้วย อ็อบเจ็คและความสัมพันธ์ระหว่างอ็อบเจ็ค โดยจะแสดงแต่ละอินสแตนซ์ของแต่ละคลาสหรือแสดงจำนวนการเกิดอ็อบเจ็คจากคลาสที่มีในระบบ

สัญลักษณ์ของอ็อบเจ็กนั้นคล้ายกับสัญลักษณ์ของคลาส เพียงแต่การเขียนชื่ออ็อบเจ็กนั้นจะต้องเริ่มต้นด้วยชื่อของออบเจ็กต์ (คำแรกขึ้นต้นด้วยอักษรตัวพิมพ์เล็กส่วนคำต่อไปขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่) ตามด้วยเครื่องหมาย (:) และชื่อคลาส (ชื่อคลาสจะต้องขึ้นต้นด้วยตัวอักษรตัวพิมพ์ใหญ่) โดยจะต้องขีดเส้นใต้ให้กับชื่อออบเจ็กต์และชื่อคลาสนั้นด้วย พร้อมทั้งให้ระบุค่าของแต่ละแอททริบิวต์ไว้ด้วย ดังรูปที่ 2.22



รูปที่ 2.22 สัญลักษณ์อ็อบเจ็คไดอะแกรม (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548: 32)

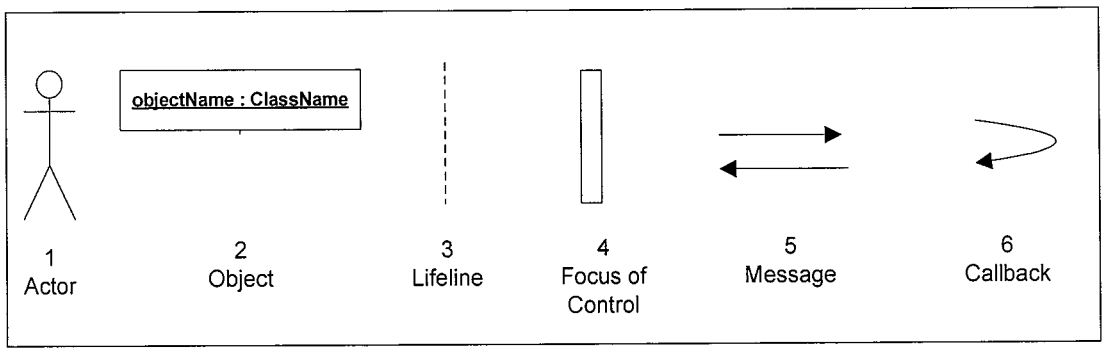
2.2.6.5 ซีเควนไดอะแกรม

ซีเควนไดอะแกรมเป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างออบเจ็กต์โดยมีการส่งเมจเสจระหว่างออบเจ็กต์ตามลำดับของเวลาที่เกิดเหตุการณ์ โดยจะแสดงถึงขั้นตอนการทำงานและลำดับของการสื่อสารระหว่างอ็อบเจ็คที่ตอบโต้กัน

ส่วนประกอบของ ซีเควนไดอะแกรม

- Actor คือ ผู้กระทำต่อระบบ
- Object คือ ออบเจ็กต์ที่ต้องทำหน้าที่
- Lifeline คือ เส้นแสดงชีวิตของออบเจ็กต์หรือคลาส
- Focus of Control / Activation คือ จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรม
- Message คือ คำสั่งหรือฟังก์ชันที่คลาสหนึ่งส่งให้อีกคลาสหนึ่ง
- Callback คือ การประมวลผลและคืนค่าที่ได้ภายในออบเจ็กต์เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงส่วนประกอบของซีเควนไดอะแกรม (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548: 32)

ในการติดต่อสื่อสารระหว่างอ็อบเจ็คต์จะทำการส่งเมจเสจ(Message) ถึงกันระหว่างอ็อบเจ็คต์แต่ละตัวหรืออาจส่งกลับมาหาตัวเองก็ได้ แบ่งการติดต่อเป็น 5 แบบ ดังแสดงในตารางที่ 2.6

ตาราง 2.6 Message

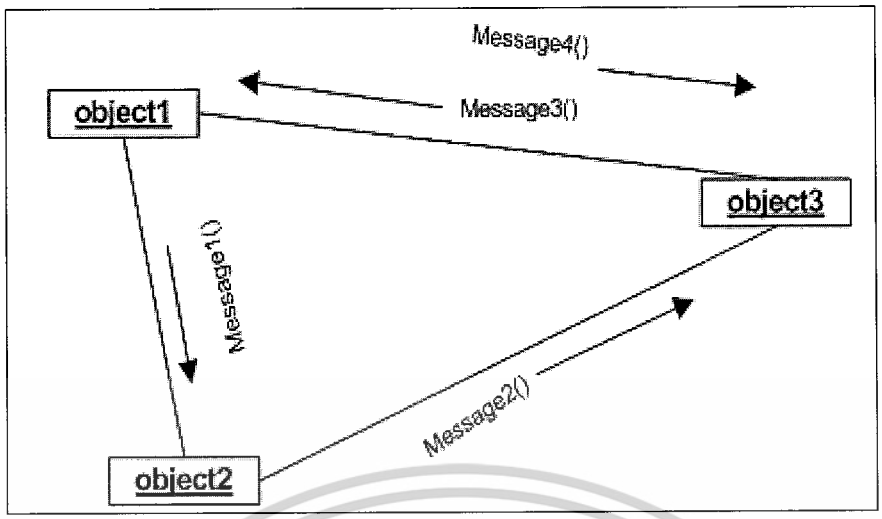
ประเภท	คำอธิบาย
Call	Message ที่ Sender เรียกใช้ Method ของ Receiver
Return	Message ที่ใช้ส่งข้อมูลที่ถูกร้องขอกลับไปยัง Sender
Send	การส่งสัญญาณเพื่อบอกหรือกระตุ้น object อื่น
Create	Message ที่ทำให้เกิดการสร้าง object ของ class ขึ้น
Destroy	Message ที่ส่งออกไปเพื่อให้ object นั้นทำลายตัวเอง

2.2.6.6 คอลแลบอเรชั่นไดอะแกรม

คอลแลบอเรชั่นไดอะแกรม ทำหน้าที่เช่นเดียวกับซีเควนไดอะแกรมแต่จะต่างกันที่ คอลแลบอเรชั่นจะเน้นการแสดงถึงความสัมพันธ์ระหว่างอ็อบเจ็กต์ตามลักษณะการทำงาน (Control by Organization)

สัญลักษณ์ที่ใช้ในไดอะแกรมจะใช้รูปสี่เหลี่ยมผืนผ้าแทนอ็อบเจ็กต์แต่ละอ็อบเจ็กต์ที่มีปฏิสัมพันธ์กันจะมีเส้นตรงเชื่อมโยงอยู่ และมีเมจเสจที่อ็อบเจ็กต์ส่งถึงกันโดยจะมีตัวเลขกำกับเพื่อบอกถึงลำดับของการเกิดเหตุการณ์ และมีลูกศรเป็นตัวแสดงทิศทางการส่งเมจเสจ ดังแสดงในรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



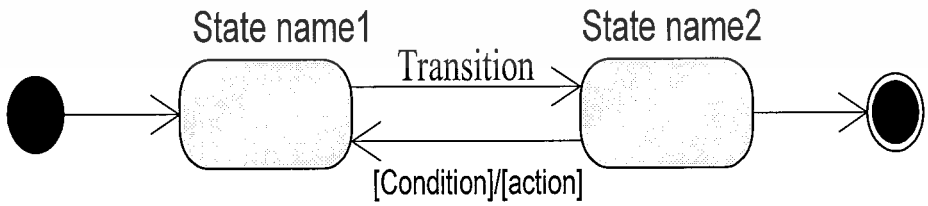
รูปที่ 2.24 ตัวอย่าง Collaboration Diagram (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. 37)

2.2.6.7 สเตตชาร์ตไดอะแกรม

สเตตชาร์ตไดอะแกรม เป็นแผนภาพที่แสดงให้เห็นพฤติกรรมของออบเจ็กต์ โดยประกอบด้วย Stateต่างๆ ของ Object และเหตุการณ์ต่างๆ ที่ทำให้สถานะของ Object เปลี่ยนแปลง จะเน้นที่การแสดงให้เห็นสถานะ (State) การเปลี่ยนสถานะ (Transition) ที่มีต่อเหตุการณ์ (Event) ที่เกิดขึ้นในช่วงชีวิตของออบเจ็กต์ 1 ช่วง (1 Sequence)

สัญลักษณ์สเตตชาร์ตไดอะแกรม ประกอบไปด้วย

- Initial State คือ จุดเริ่มต้นการเปลี่ยนสถานะ
- Final State คือ จุดสิ้นสุดของการเปลี่ยนสถานะ
- Transition คือ เส้นกระตุ้นให้เปลี่ยนสถานะ
- State คือ สถานะของออบเจ็กต์

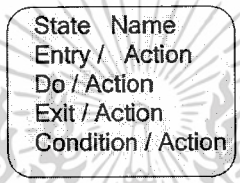


รูปที่ 2.25 แสดงสัญลักษณ์ของ State chart Diagram (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. 38)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองกิจกรรมภายใน State ใช้เพื่อระบุรายละเอียดในการทำงานของ State ต่าง ๆ แบ่งได้เป็น

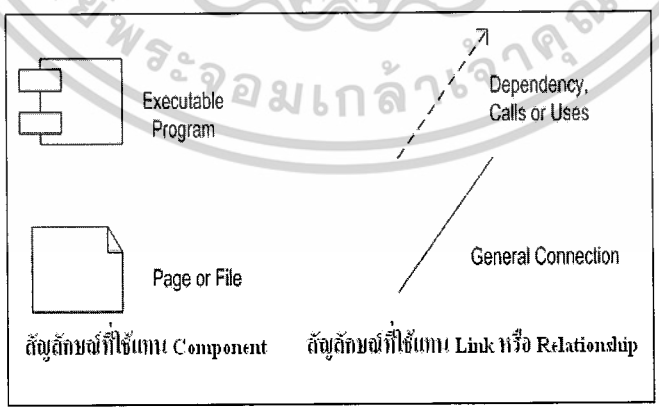
- Entry/action หมายถึง เมื่อเข้ามายัง state นี้ให้ทำกิจกรรม (action)
  เช่น Entry/count=0 หมายถึง เมื่อเข้ามายัง state ให้ค่า count เป็น 0
- Do/action หมายถึง หลังจากเข้ามายัง state นี้แล้ว หากไม่มีเงื่อนไขอื่นใด ให้ทำกิจกรรม
  เช่น do/count:=count+1 หมายถึง เมื่อเข้ามายัง state นี้ให้เพิ่มค่า count ทีละ 1
- Exit/action หมายถึง ขณะที่จะออกจาก state นี้ให้ทำกิจกรรม action
  เช่น Exit/show "Good Bye" หมายถึง หากออก state นี้ให้แสดงข้อความ "Good Bye"
- Condition/action หมายถึง ขณะที่ยังอยู่ใน state นี้เมื่อเกิดเงื่อนไขใด ๆ ที่กำหนด โดยให้ทำกิจกรรม เช่น every 2 seconds/phone ring หมายถึง ทุก ๆ 2 วินาทีให้เสียงดัง 1 ครั้ง



รูปที่ 2.26 แสดงสัญลักษณ์การจำลองกิจกรรมของ State (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ ลมกล่อม. 2548. 38)

2.2.6.8 คอมโพเนนต์ไดอะแกรม

คอมโพเนนต์ไดอะแกรม เป็นไดอะแกรมที่แสดงโครงสร้างทางกายภาพของโปรแกรม ประกอบด้วยส่วนประกอบต่างๆ ที่เรียกว่า "Component" ซึ่งหมายถึงส่วนประกอบย่อยของซอฟต์แวร์ของระบบทั้งหมดซึ่งอยู่ในรูปต่างๆ เช่นไบนารี (Binary) เท็กซ์ (text) เป็นต้น ดังรูปที่ 2.27

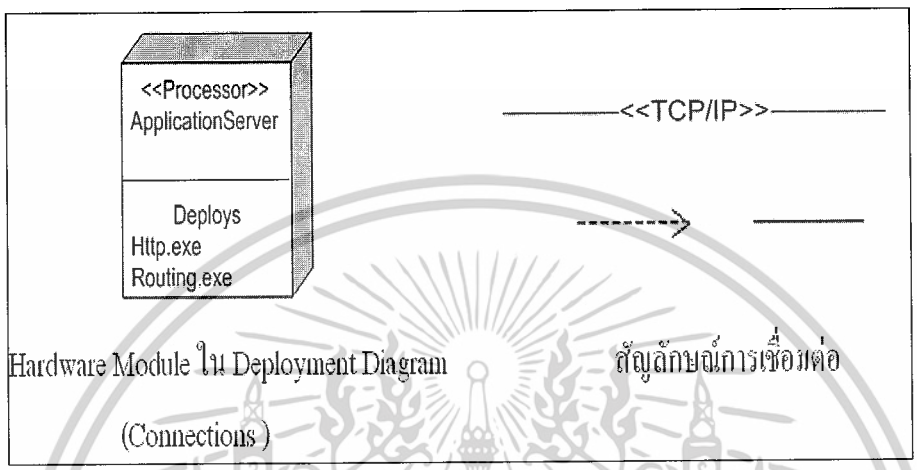


รูปที่2.27 สัญลักษณ์ในคอมโพเนนต์ไดอะแกรม (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ ลมกล่อม. 2548. 39)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6.9 ดีพลอยเมนต์ไดอะแกรม

ดีพลอยเมนต์ไดอะแกรม เป็นแผนภาพแสดงโครงสร้างทางด้านฮาร์แวร์ของระบบงาน ส่วนใหญ่จะใช้ร่วมกับComponent Diagram โดยการแสดงการเชื่อมต่อระบบ สามารถทำการแสดงระบบสถาปัตยกรรมของ Hardware/Software ตลอดจนความสัมพันธ์ระหว่างกัน



รูป 2.28 สัญลักษณ์ของ Deployment Diagram (กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ ลมกล่อม. 2548. 39)

2.3 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

2.3.1 อาร์ซีเอ็ก (RCX)

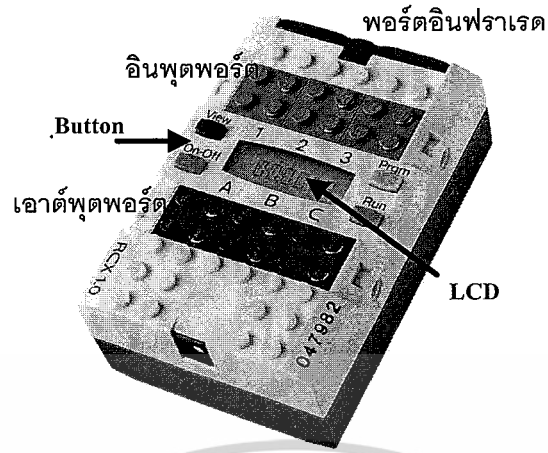
อาร์ซีเอ็ก คือ ตัวควบคุมโปรแกรมที่ใช้ในการควบคุมหุ่นยนต์เลโก้มายสตรอม (LEGO MINDSTORMS Robots: Robotics Invention System) หรือเป็นองค์ประกอบหลักของระบบมายสตรอม (mind storms) ซึ่งเป็นระบบฝังตัว ที่ใช้ไมโครคอนโทรลเลอร์ (microcontroller) ของฮิตาชิ รุ่น H8/3932 (Hitachi H8/3932) เป็นหน่วยประมวลผลกลางควบคุมการทำงาน โดยใช้หน่วยประมวลผลหลัก (CPU) รุ่น H8/300 ดังรูปที่ 2.29

คุณสมบัติของอาร์ซีเอ็ก

1. สามารถเก็บบันทึกข้อมูลได้5 โปรแกรม
2. สามารถเก็บคำสั่งได้ประมาณ 1,500 คำสั่ง
3. พอร์ตอินพุตและพอร์ตเอาท์พุต อย่างละ 3 พอร์ต
4. อาร์ซีเอ็กสามารถส่งถ่ายข้อมูลระหว่างอาร์ซีเอ็กกับอาร์ซีเอ็กได้

* หมายเหตุ ใช้แบตเตอรี่ถ่าน AA 6 ก้อน (9 v)

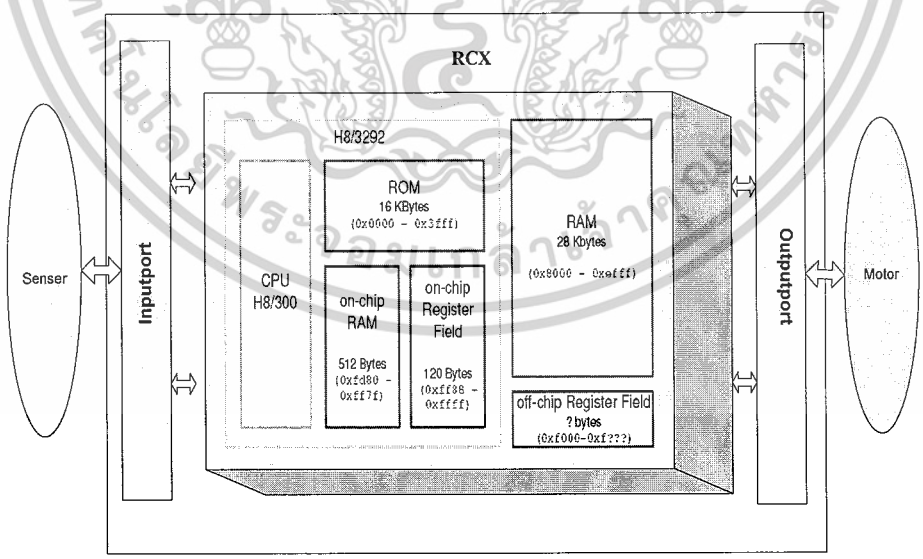
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 ส่วนประกอบของ อาร์ซีเอ็ก (http://legolab.daimi.au.dk/DigitalControl.dir /RCX/Manual.dir/RCXManual)

2.3.2 โครงสร้างของอาร์ซีเอ็ก

โครงสร้างของอาร์ซีเอ็กซึ่งมีไมโครคอนโทรลเลอร์ H8/3292 เป็นหน่วยประมวลผลภายในประกอบด้วยหน่วยประมวลผลกลางของฮิตาชิรุ่น H8/300 หน่วยความจำและพอร์ตอินพุตและเอาต์พุตสำหรับการเชื่อมต่อกับอุปกรณ์อื่น เช่นเซนเซอร์ มอเตอร์ และแอคทูเอเตอร์ต่าง ๆเป็นต้น สามารถประมวลผลคำสั่งจากเครื่องคอมพิวเตอร์ผ่านทางการเชื่อมต่อแบบอินฟราเรด มีพอร์ตอินพุตและเอาต์พุต นอกจากนี้ยังประกอบด้วย ปุ่มกด ลำโพง หน้าจอแสดงผล (LCD) เป็นต้น ดังรูปที่ 2.30



รูปที่ 2.30 แสดงส่วนประกอบและโครงสร้างของ RCX (Japan System House Association. 2549. แปลและเรียบเรียงโดย ดร.ธนารัตน์ ธีระมั่นคง : 95)

ภายในไมโครคอนโทรลเลอร์ H8/3292 ซึ่งประกอบด้วยส่วนหลักๆ 3 ส่วนที่ทำงานร่วมกัน ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หน่วยประมวลผล (CPU) เป็นหน่วยประมวลผลของฮิตาชิรุ่น H8 / 300 ทำงานที่ความถี่ 16 เมกะเฮิร์ต(MHz) เป็นหน่วยประมวลผลกลาง ที่มีสถาปัตยกรรมทั่วไป คือ ประกอบไปด้วยกระบวนการทางคณิตศาสตร์และตรรกกะ และประกอบไปด้วย เรจิสเตอร์ (Register) เพื่อช่วยในการประมวลผลและทำงาน

2. หน่วยความจำ ประกอบไปด้วยหน่วยความจำแรม (random access memory : RAM) ขนาด 32 กิโลไบต์ ซึ่งใช้ในการเก็บข้อมูลและที่อยู่ของชุดคำสั่งของโปรแกรมที่ใช้ในการประมวลผล และประกอบไปด้วยหน่วยความจำรอม (Read Only Memory: ROM) ขนาด16 กิโลไบต์ซึ่งเป็นที่อยู่ของ ไดรเวอร์ (driver)ที่ใช้ในการทำงานในขณะที่เริ่มต้นใช้งาน RCX ซึ่งส่วนของไดรเวอร์ จะถูกแบ่งเป็นส่วนของเฟิร์มแวร์ส่วนหนึ่งซึ่งทำหน้าที่ในการประมวลผลคำสั่งที่โปรแกรมเข้าไป

3. อินพุตและเอาต์พุตพอร์ต เป็นส่วนต่อประสาน (Interface) กับอุปกรณ์ที่มาเชื่อมต่อกับอาร์ซีเอ็กซ์ เช่น มอเตอร์ เซนเซอร์ และแอคทูเอเตอร์ เป็นต้น ประกอบไปด้วยส่วนต่อประสานติดต่อ สื่อสารอนุกรม (Serial Communication Interface) ใช้ในการควบคุมอุปกรณ์ ตัวรับตัวส่งการติดต่อสื่อสารอนุกรม ตัวแปลงแอนะล็อกเป็นดิจิตอล(Analog-to-Digital Converter) ใช้ในการแปลงสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัลเพื่อใช้ในการประมวลผลข้อมูล และมีกลไกสำหรับขัดจังหวะ (Interrupt) ทั้งภายในและภายนอก ซึ่งถูกควบคุมการทำงานโดย ตัวควบคุมการขัดจังหวะ (Interrupt Control) นอกจากนียังประกอบไปด้วยส่วนของดีไวซ์คอนโทรลเลอร์ (Device Controller) ซึ่งทำหน้าที่ควบคุมการติดต่อสื่อสารของแต่ละตัวอุปกรณ์ และควบคุมการขัดจังหวะ

2.3.3 ส่วนประกอบของอาร์ซีเอ็กซ์

ส่วนประกอบของอาร์ซีเอ็กซ์ประกอบด้วยส่วนของหน้าจอแสดงผล ปุ่มควบคุมการทำงาน พอร์ตอินพุตและพอร์ตเอาต์พุตเพื่อเชื่อมต่อกับอุปกรณ์ภายนอก

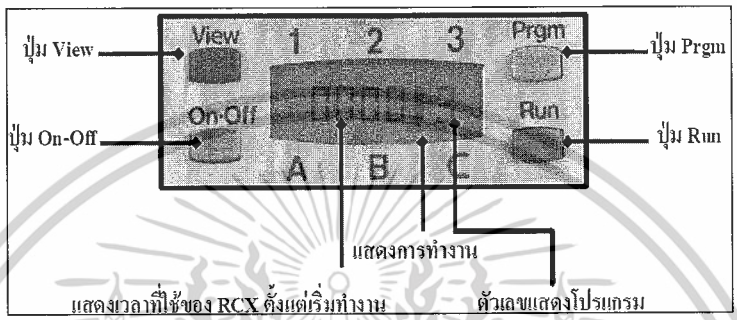
1. จอแสดงผล เป็นภาพสัญลักษณ์รูปคน และหมายเลขที่แสดงถึงโปรแกรมที่ใช้อยู่ สัญลักษณ์นี้จะปรากฏขึ้นเมื่ออาร์ซีเอ็กเปิดใช้งานอยู่ ถ้ากดปุ่ม Run รูปสัญลักษณ์คนจะเปลี่ยนเป็นรูปคนกำลังเดิน แสดงว่าขณะนี้อาร์ซีเอ็ก กำลังทำงานตามคำสั่งอยู่

2. ปุ่มควบคุมการทำงานประกอบด้วย 4 ปุ่ม โดยทั้ง 4 ปุ่ม จะเชื่อมต่อกับสายอินพุต/เอาท์พุตพอร์ต ได้แก่

- ปุ่ม On-Off (ปุ่มสีแดง) กดเพื่อเปิด – ปิดตัวอาร์ซีเอ็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่ม View (ปุ่มสีดำ) ทำหน้าที่เลือกและดูการทำงานของพอร์ตอินพุตและเอาท์พุต
- ปุ่ม Prgm (ปุ่มสีเทา) ใช้เลือกโปรแกรมที่ต้องการจะใช้งาน (สามารถเก็บได้5 โปรแกรม แต่โปรแกรมที่1 และ 2 จะถูกล็อคไว้เป็นเฟิร์มแวร์)
- ปุ่ม Run (ปุ่มสีเขียว) เป็นปุ่มใช้เริ่มการทำงานตามที่ได้ตั้งโปรแกรมไว้



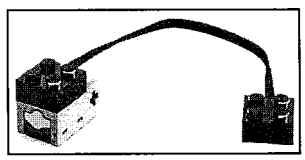
รูปที่ 2.31 แสดงปุ่มควบคุมการทำงานของ RCX (http://legolab.daimi.au.dk/DigitalControl.dir /RCX/Manual.dir/RCXManual)

3. พอร์ตต่าง ๆ บนอาร์ซีเอ็กซ์ ประกอบไปด้วย พอร์ตอินพุตและพอร์ตเอาต์พุต

- พอร์ตอินพุต (Input Port) ใช้เชื่อมต่อเซนเซอร์ต่าง ๆ เช่น เซนเซอร์แสงเป็นต้นโดยใช้พอร์ตที่เป็นตัวเลขคือ 1,2 และ 3 ที่ใช้เป็นตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล
- พอร์ตเอาพุต (Output Port) ใช้เชื่อมต่อกับอุปกรณ์แสดงผล เช่น มอเตอร์หรือหลอดไฟ เป็นต้น จะใช้พอร์ตที่เป็นตัวอักษรคือ A, B และ C ใช้แปลงสัญญาณดิจิตอล เป็นอนาล็อก

4. เซนเซอร์ (Sensor) เป็นอุปกรณ์ในการตรวจจับสัญญาณ เพื่อนำข้อมูลดังกล่าวไปประมวลผล เปรียบเสมือนระบบประสาทของมนุษย์ ซึ่งประกอบด้วยเซนเซอร์ต่าง ๆ ดังนี้

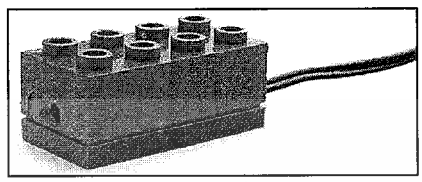
เซนเซอร์สัมผัส (Touch Sensor) เป็นอุปกรณ์ที่ใช้ในการตรวจจับการชนถ้ามีการสัมผัสหรือชนกับตัวเซนเซอร์ จะส่งค่ากลับไปยัง อาร์ซีเอ็ก (คือมีค่าเป็น 1) แต่ถ้าไม่มีการชนหรือสัมผัสกับตัวเซนเซอร์ ก็จะไม่ส่งค่าไปยังอาร์ซีเอ็ก (คือมีค่าเป็น 0)



รูปที่ 2.32 เซนเซอร์สัมผัส (http://legolab.daimi.au.dk/DigitalControl.dir)

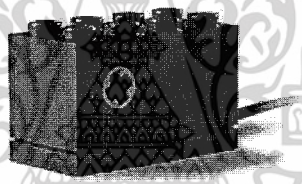
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์แสง (Light Sensor) เป็นอุปกรณ์ที่ใช้ในการตรวจจับการสะท้อนของแสงและการดูดกลืนแสง และใช้วัดความเข้มของแสงที่กระทบพื้นขึ้นมา ดังนั้นเราสามารถกำหนดโปรแกรมหรือพฤติกรรมของหุ่นยนต์ได้จากแสงด้วยค่าที่อ่านได้จะขึ้นกับความสามารถในการสะท้อนแสงของวัตถุที่ตรวจจับ ดังนั้นจะมีค่า 0-100



รูปที่ 2.33 เซนเซอร์แสง (http://legolab.daimi.au.dk/DigitalControl.dir)

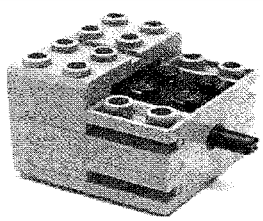
เซนเซอร์จับการหมุน (Rotation Sensor) เป็นอุปกรณ์ที่ใช้ในการวัดระยะทางการเคลื่อนที่ของหุ่นยนต์โดยการตรวจจับที่เพลาหมุนของล้อหุ่นยนต์สามารถประยุกต์ใช้ในการวัดระยะทางได้ วัดความเร็วลมได้ จากการนับสัญญาณที่ส่งมา



รูปที่ 2.34 เซนเซอร์จับการหมุน (http://legolab.daimi.au.dk/DigitalControl.dir)

5. แอกทูเอเตอร์ คืออุปกรณ์ที่เป็นส่วนที่ใช้แสดงผลของระบบว่ามีการทำงานหรือเอาท์พุตของระบบ พอร์ตเอาท์พุตแต่ละตัวสามารถขับเคลื่อนแอกทูเอเตอร์มากกว่า 1 ตัวได้ แอกทูเอเตอร์ ประกอบด้วย3 ชนิด ดังนี้

- มอเตอร์ (Motor) เป็นอุปกรณ์ที่ใช้ในการขับเคลื่อน ซึ่งเป็นมอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ทำให้เกิดการเคลื่อนที่ใน 2 ทิศทาง ในลักษณะทวนเข็มนาฬิกาและตามเข็มนาฬิกา



รูปที่ 2.35 มอเตอร์(http://legolab.daimi.au.dk/DigitalControl.dir)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

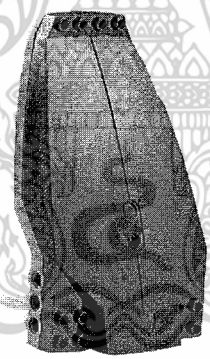
- หลอดไฟ เป็นอุปกรณ์การแสดงผลของระบบโดยการแสดงออกด้วยการแสดงแสงออกมาทางหลอดไฟ เพื่อบอกสถานะ หรือผลการทำงานในขณะนั้น

- ลำโพง เป็นอุปกรณ์การแสดงผลของระบบโดยการแสดงเสียงออกมาทางลำโพงเพื่อบอกสถานะหรือผลการทำงานในขณะนั้น

6. เครื่องส่งสัญญาณด้วยแสงอินฟราเรด (Infrared Transmitter: IR) เป็นเครื่องส่งสัญญาณด้วยแสงอินฟราเรดที่ใช้สำหรับส่งข้อมูลจากเครื่องคอมพิวเตอร์กับอาร์ซีเอ็ก โดย เครื่องส่งสัญญาณด้วยแสงอินฟราเรด (IR) จะถูกติดตั้งเข้ากับเครื่องคอมพิวเตอร์ที่พอร์ตสื่อสาร USB โดยขณะส่งข้อมูลต้องหันทางด้านหน้าของอาร์ซีเอ็ก (ต้องเปิดเครื่อง อาร์ซีเอ็ก ทุกครั้งที่ทำการดาวน์โหลด) เข้าหาด้านหน้าของ โดยเครื่องส่งสัญญาณด้วยแสงอินฟราเรด และจะมีสัญญาณไฟ (LED) สีเขียวแสดงสถานะ การทำงาน

อุปกรณ์ภาคส่งอินฟราเรด ทำหน้าที่เป็นตัวควบคุมอุปกรณ์สำหรับการสื่อสารใน อาร์ซีเอ็ก แสงอินฟราเรดเปิดและปิดด้วยความถี่ 38.5 กิโลเฮิร์ต จะถูกใช้เป็นตัวส่งกระแสของบิต

- ค่าบิต 0 แทนช่วงเวลา (Time Interval) ตอนที่มีแสงอินฟาเรด มีการส่งข้อมูล
- ค่าบิต 1 แทนช่วงเวลา (Time Interval) ตอนที่ ไม่มีการส่งข้อมูล



รูปที่ 2.36 เครื่องส่งสัญญาณด้วยแสงอินฟราเรด (http://legolab.daimi.au.dk/DigitalControl.dir)

2.3.4 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

การแปลงโปรแกรมเป็นภาษาสำหรับ RCX จะต้องแปลงจากภาษาที่เขียนให้เป็นภาษาที่ ตัวประมวลผลกลาง (CPU H8/300) เข้าใจก่อน ซึ่งวิธีที่ใช้ในการแปลงนั้นจะต้องมีตัวช่วยในการแปลงข้อมูลภาษา ให้อยู่ในรูปแบบที่หน่วยประมวลผลเข้าใจได้ และส่งคำร้องขอไปยังหน่วยประมวลผลโปรแกรมของอาร์ซีเอ็ก(RCX executive)เพื่อการดาวน์โหลดและเริ่มโปรแกรม ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ ประกอบไปด้วยบริคโอเอส (Bricks) และโปรแกรมซิกวิน (Cygwin)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บริคโอเอส (BrickOS) เป็นชนิดของระบบปฏิบัติ (Operating System) ชนิดหนึ่ง ที่มีแนวคิดของระบบปฏิบัติการแบบเรียลไทม์ และมี หน่วยประมวลผลกลาง (Central processing unit :CPU) ซึ่งจะทำการประมวลผลบนสภาพแวดล้อมของอาซีเอ็ก(Robotics Command Explorer) บริคโอเอสพัฒนามาจากระบบปฏิบัติการที่ชื่อว่า legOS เป็นระบบปฏิบัติการที่ให้ใช้งานกันแบบเปิด (open source) ประกอบด้วยชุดคำสั่งพื้นฐานที่ใช้ในการควบคุมอาร์ซีเอ็กซ์อยู่ภายใน สามารถพัฒนาด้วยโปรแกรมภาษาซี หรือ ภาษาซีพลัสพลัส (C++) ได้และสามารถพัฒนาได้ในหลายระบบปฏิบัติการ ทั้งลินุกซ์ (Linux) วินโดว์ (Windows) เป็นต้น ซึ่งการทำงานที่ต่างระบบปฏิบัติการนี้จะต้องมีคอมไพเลอร์แบบไขว้ (cross-compiler) มาช่วยในการแปลงข้อมูลให้อยู่ในรูปแบบที่อาร์ซีเอ็กซ์เข้าใจได้

2. โปรแกรมซิกวิน คือชุดโปรแกรมสร้างสภาพแวดล้อมลินุกซ์บนวินโดวส์ ประกอบด้วย cygwin1.dll และเครื่องมือที่จำเป็น เพื่อจำลองสภาพแวดล้อมให้เหมือนลีนุกซ์ ซึ่งโปรแกรมนี้จะช่วยคอมไพล์โปรแกรมที่ถูกพัฒนาให้คอมไพล์บนลินุกซ์สามารถถูกแปลบนวินโดวน์ได้ในภายหลัง โดยจะขึ้น bash shell ให้สามารถใช้งานได้เหมือนทำงานกับเครื่องที่เป็นยูนิกซ์ (UNIX) ทั่วไป ซึ่งโปรแกรมนี้จะใช้สำหรับการคอมไพล์โปรแกรมลงบน RCX ได้เช่นเดียวกัน

3. เครื่องมือ (Tools) ที่ใช้ในการพัฒนาระบบ คือ ภาษาซีพลัสพลัส (C++ programming language) ซึ่งเป็นภาษาโปรแกรมคอมพิวเตอร์อเนกประสงค์ มีโครงสร้างภาษาที่มีการจัดการข้อมูลแบบสแตติก (Statically typed) และสนันสนุนการเขียนโปรแกรมที่หลากหลาย ได้แก่ การโปรแกรมเชิงกระบวนคำสั่ง การโปรแกรมเชิงวัตถุ เป็นต้น

ภาษาซีพลัสพลัสเป็นภาษาโปรแกรมที่นิยมมากภาษาหนึ่งนับตั้งแต่ช่วงทศวรรษ 1990 ซึ่งเกิดขึ้นเมื่อปี ค.ศ 1980 .ภาษา C++ ถูกพัฒนามาจากภาษา C โดยนาย Bjarne Stroustrup ที่ห้องวิจัย Bell Lab บริษัท AT&T ในช่วงนั้นเป็นการเพิ่มคลาสให้กับภาษา C เรียกว่า C with classes และใช้เฉพาะหน่วยงานภายในเท่านั้น การพัฒนาภาษา C++ นั้นจะทำร่วมกับภาษา Simula ซึ่งเป็นภาษาเชิงวัตถุที่เริ่มต้นในช่วงนั้น และในปี 1983 ได้ถูกตั้งชื่อขึ้นเป็นภาษา C++ หลังจากนั้นภาษา C++ ได้ถูกพัฒนาและเพิ่มคุณสมบัติเด่นมากขึ้นเรื่อย ๆ ได้แก่ เวอร์ชวลฟังก์ชัน การโอเวอร์โหลดโอเปอเรเตอร์ การสืบทอดหลายสาย เท็มเพลต และการจัดการเอ็กเซ็พชัน มาตรฐานของภาษาซีพลัสพลัสได้รับการรับรองในปีค.ศ. 1998 เป็นมาตรฐาน ISO/IEC 14882:1998 เวอร์ชันล่าสุดคือเวอร์ชันในปี ค.ศ.2003 ซึ่งเป็นมาตรฐาน ISO/IEC 14882:2003 ในปัจจุบันมาตรฐานของภาษาในเวอร์ชันใหม่ (รู้จักกันในชื่อ C++0x) กำลังอยู่ในขั้นพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การพัฒนาระบบหุ่นยนต์ติดตามเส้นทาง

ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบและพัฒนาระบบหุ่นยนต์ติดตามเส้นทาง (Line trace robot) โดยเริ่มจากการเขียนรายละเอียดความต้องการของระบบแล้วทำการวิเคราะห์และออกแบบกระบวนการทำงานของระบบงานจริงทั้งในส่วนของซอฟต์แวร์ (Software) และฮาร์ดแวร์ (hardware)

3.1 ความต้องการของระบบ

หุ่นยนต์ติดตามเส้นทาง (Line trace robot) จะทำงานโดยการอ่านค่าข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซนเซอร์ (Sensor) และนำข้อมูลดังกล่าวมาประมวลผลเพื่อให้สามารถทำงานได้อย่างถูกต้องตามที่ต้องการ คือสามารถติดตามเส้นทางและเคลื่อนที่ไปตามสภาพแวดล้อมที่กำหนดได้ ซึ่งสามารถสรุปเป็นความต้องการของระบบได้ดังนี้

3.1.1. ความต้องการหลักของระบบ (Functional Requirement)

1. หุ่นยนต์สามารถรับข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซนเซอร์ซึ่งประกอบด้วยเซนเซอร์แสงจำนวน 1 ตัว และนำข้อมูลดังกล่าวมาประมวลผลได้
2. หุ่นยนต์สามารถเคลื่อนที่โดยสั่งงานผ่านมอเตอร์จำนวน 2 ตัว ซึ่งประกอบด้วยสเตียริงมอเตอร์ (Steering Motors) ทำหน้าที่ในการควบคุมทิศทางของการเคลื่อนที่และไดร์วิงมอเตอร์ (Driving Motors) ทำหน้าที่ในการขับเคลื่อนหุ่นยนต์ให้สามารถทำงานได้ตามที่ต้องการ
3. หุ่นยนต์สามารถเคลื่อนที่ไปตามสภาพแวดล้อมที่กำหนดได้ ประกอบด้วยสภาพเส้นทางที่เป็นเส้นทางตรง เส้นทางโค้ง เส้นทางชันและเส้นทางลัด (Shortcut) ได้
4. หุ่นยนต์สามารถตัดสินใจเลือกเส้นทาง โดยสามารถตัดสินใจเลือกเส้นทางที่เป็นเส้นทางปกติ เส้นทางลัด (Short cut) เส้นทางชัน และเคลื่อนที่ไปตามเส้นทางนั้นได้
5. หุ่นยนต์สามารถควบคุมระดับความเร็วตามลักษณะเส้นทางที่เคลื่อนที่ไปได้

3.1.2 เงื่อนไขและสภาพแวดล้อมที่ใช้ในการพัฒนาระบบ

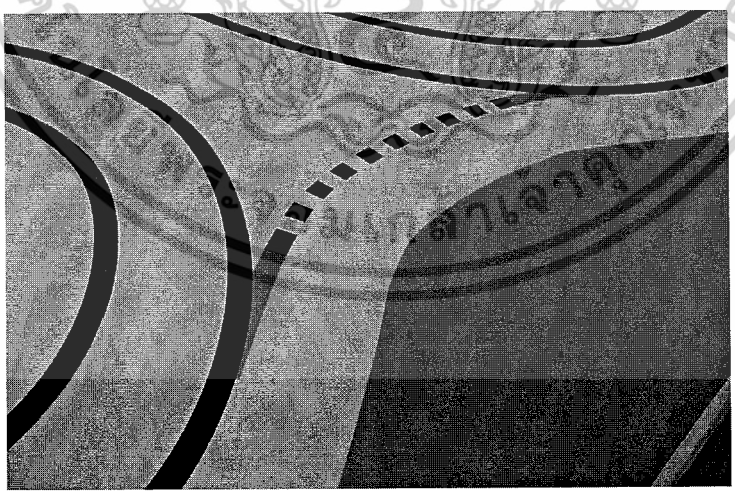
ระบบสามารถทำงานได้โดยมีสภาพแวดล้อมในการพัฒนาดังนี้

1. เส้นทึบสีดำ แสดงสภาพถนนในรูปแบบปกติ ขนาดความกว้าง 34 มิลลิเมตร
2. เส้น Short Cut แสดงสภาพถนนที่เป็นทางลัด (Short Cut) โดยมีขนาดความกว้างของเส้นสีขาวและสีดำคือ 34 มิลลิเมตร และระยะห่างหรือความยาวของสีดำในแต่ละช่วงคือ 20 มิลลิเมตรและความยาวของสีขาวในแต่ละช่วงคือ 10 มิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ลักษณะสภาพถนนแบบเส้นปกติจะมีสภาพดังนี้ คือ
   - เส้นทางตรง
   - เส้นทางโค้ง
   - เส้นทางลาดชัน
4. ลักษณะสภาพถนนแบบเส้นทางลัดจะมีสภาพดังนี้ คือ
   - เส้นทางตรง
   - เส้นทางโค้ง

รูปที่ 3.1 แสดงสภาพถนนที่เป็นเงื่อนไขของทั้งระบบ



รูปที่ 3.2 แสดงสภาพถนนที่เป็นเส้นทางลัด (Short Cut)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การวิเคราะห์และออกแบบระบบ

จากความต้องการของระบบดังกล่าวสามารถนำมาวิเคราะห์และออกแบบระบบได้ดังนี้

3.2.1 การวิเคราะห์ระบบ

ตารางที่ 3.1 การวิเคราะห์ความต้องการและฟังก์ชันการทำงานของระบบ

ความต้องการของระบบ	ยูสเคสที่รองรับการทำงาน	แอคเตอร์
1. ระบบสามารถอ่านข้อมูลจากสภาพแวดล้อม โดยผ่านทางอุปกรณ์เซนเซอร์แสงจำนวน 1 ตัว		
2. ระบบสามารถเคลื่อนที่โดยสั่งงานผ่านมอเตอร์เพื่อให้เคลื่อนที่ไปตามสภาพแวดล้อมที่ต้องการได้	 	
3. ระบบสามารถเคลื่อนที่ไปตามสภาพแวดล้อมที่กำหนดได้ ซึ่งประกอบด้วยสภาพเส้นทางที่เป็นเส้นทางตรง เส้นทางโค้งและเส้นทางชันได้	 	
4. ระบบสามารถตัดสินใจเลือกเส้นทางโดยสามารถตัดสินใจเลือกเส้นทางที่เป็นทางปกติ เส้นทางลัด (Short cut) และเส้นทางชัน ตามที่กำหนดและเคลื่อนที่ไปตามเส้นทางนั้นได้		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

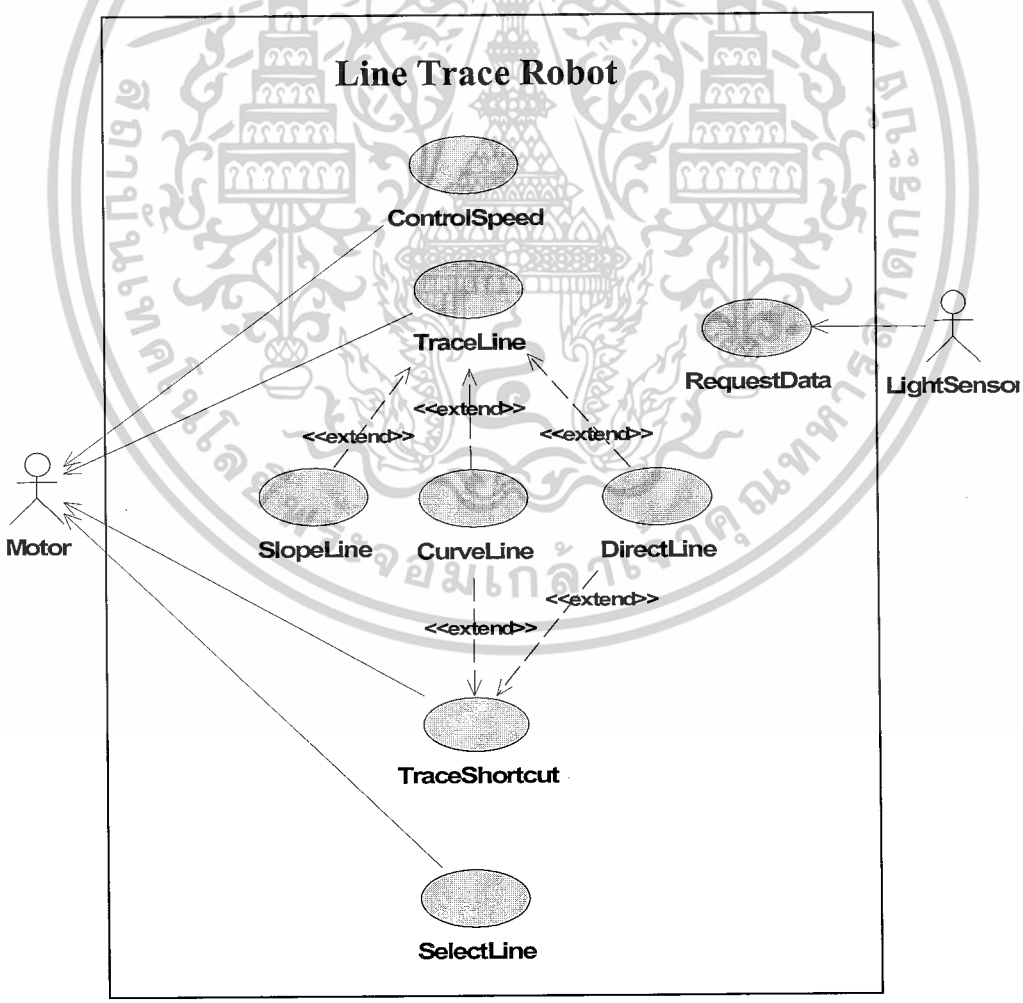
ตารางที่ 3.1 (ต่อ) การวิเคราะห์ความต้องการและฟังก์ชันการทำงานของระบบ(ต่อ)

ความต้องการของระบบ	ยูสเคสที่รองรับการทำงาน	แอคเตอร์
5. ระบบสามารถควบคุมความเร็วตามลักษณะเส้นทางที่เคลื่อนที่ไปตามแต่ละสภาพเส้นทางได้	ControlSpeed	Motor

3.2.2 การออกแบบซอฟต์แวร์

3.2.2.1 ยูสเคสไดอะแกรม

จากความต้องการหลักของระบบสามารถนำมาออกแบบเป็นแผนภาพยูสเคสไดอะแกรม(Use case Diagram) ได้ดังนี้



รูปที่ 3.3 แสดงยูสเคสไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพยูสเคสไดอะแกรมดังกล่าว สามารถอธิบายได้ดังนี้ ระบบหุ่นยนต์ติดตามเส้นทางจะมีการรับข้อมูลจากสภาพแวดล้อม ซึ่งแสดงโดยยูสเคส RequestData โดยมีแอคเตอร์ (Actor) ที่มีปฏิสัมพันธ์กับยูสเคสนี้คือเซนเซอร์แสง (LightSensor) ซึ่งเป็นแอคเตอร์ที่มีการให้ข้อมูลกับระบบ

เมื่อยูสเคส RequestData ทำการรับข้อมูลและประมวลผลข้อมูลดังกล่าวเพื่อใช้สำหรับการทำงานต่อไป โดยที่ระบบจะต้องมีการตัดสินใจเลือกเส้นทาง ซึ่งแสดงโดยยูสเคส SelectLine เพื่อตรวจสอบเงื่อนไขของระบบจากค่าที่รับเข้ามา ได้แก่การตรวจสอบเส้นทางปกติ เส้นทางลัด และเส้นทางชัน หากมีการเคลื่อนที่ในเส้นทางปกติ ซึ่งแสดงโดยยูสเคส TraceLine ระบบจะเคลื่อนที่โดยการติดตามเส้นที่มีลักษณะสีดำต่อเนื่อง โดยจะประกอบด้วยลักษณะของการเคลื่อนที่ที่เป็นลักษณะของเส้นทางตรง เส้นทางโค้ง และเส้นทางชัน ซึ่งแสดงโดยยูสเคส DirectLine CurveLine และ SlopeLine ตามลำดับ ซึ่งยูสเคสเหล่านี้จะมีความสัมพันธ์กับยูสเคส TraceLine ในรูปแบบของการ Extend

สำหรับการเคลื่อนที่ในเส้นทางลัด (Shortcut) แสดงโดยยูสเคส TraceShortCut ซึ่งเป็นการเคลื่อนที่ในลักษณะที่ต่างไปจากเส้นทางปกติ โดยยูสเคสนี้จะมีการ Extend ยูสเคส DirectLine และ CurveLine ซึ่งเป็นเงื่อนไขการทำงานสำหรับเส้นทางลัด นอกจากนี้ยังมียูสเคส ControlSpeed ซึ่งเป็นยูสเคสที่ใช้ในการปรับระดับความเร็วตามสภาพแวดล้อมที่ระบบเคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายยูสเคส (Use case specification)

ตารางที่ 3.2 แสดงคำอธิบายยูสเคสร้องขอข้อมูล (RequestData)

Line trace robot		
Use Case Name : ร้องขอข้อมูล (RequestData)	ID: 01	Importance Level: High
Primary Actor : เซนเซอร์แสง (Light Sensor)	Use Case Type : Function	
Stakeholders : -		
Precondition : ทำการเปิดสวิตซ์เพื่อให้ระบบเริ่มทำงาน		
Post-condition: : -		
Brief Description : ระบบรับข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซนเซอร์แสง (Light Sensor)		
Trigger : เซนเซอร์แสง (Light Sensor)อ่านข้อมูล		
Relationships : Association : เซนเซอร์แสง (Light Sensor)		
Normal Flow of Events : 1. ระบบเริ่มทำงานโดยการเปิดสวิตซ์ 2. ระบบทำการเก็บค่าข้อมูลจากสภาพแวดล้อมจริงเพื่อใช้ในการประมวลผลโดยใช้ปุ่ม PROGRAM ในการกดบันทึกค่าแต่ละครั้ง ซึ่งประกอบด้วยค่าข้อมูลสีของเส้นสีดำ ค่าข้อมูลพื้นสนามสีขาว ค่าข้อมูลสำหรับการตัดสินใจเลือกเส้นทางชัน และค่าข้อมูลสำหรับการตัดสินใจเลือกเส้นทางลัด (Shortcut) 3. ระบบเริ่มเคลื่อนที่โดยการกดปุ่ม RUN 4. เซนเซอร์ (Sensor) อ่านค่าข้อมูลในขณะที่ระบบทำการเคลื่อนที่ 5. ระบบทำการตรวจสอบค่าข้อมูลที่รับเข้ามา		
SubFlows : -		
Alternate/Exceptional Flows : 5.a ระบบทำการตรวจสอบค่าข้อมูลที่รับเข้ามา 5.a.1 ถ้าอ่านค่าสีตรงตามที่ค่าที่กำหนดไว้ให้ทำการติดตาม (Trace) จากค่าสีนั้น 5.a.2 ถ้าอ่านค่าสีไม่ได้ตามที่ต้องการให้ทำการเคลื่อนที่และอ่านค่าต่อไป 5.a.3 ถ้าอ่านค่าที่เป็นเงื่อนไขการตัดสินใจหรือการทำงาน ได้แก่ เส้นทางชันและเส้นทางลัดให้เตรียมปรับลักษณะการเคลื่อนที่และความเร็วที่ใช้		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงคำอธิบายยูสเคสติดตามเส้นทางปกติ (TraceLine)

Line trace robot		
Use Case Name : ติดตามเส้นทางปกติ(TraceLine)	ID: 02	Importance Level: High
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : -		
Post-condition: : -		
Brief Description : ระบบทำการติดตามเส้นทางปกติ		
Trigger : เซนเซอร์แสง (Light Sensor)อ่านข้อมูลได้เป็นเส้นสีปกติ (เส้นสีดำ)		
Relationships : Association : มอเตอร์ (motor) Extend : DirectLine ,SlopeLine, CurveLine		
Normal Flow of Events : 1. ระบบรับค่าเข้ามาเป็นเส้นสีปกติ 2. ระบบทำการเคลื่อนที่ในลักษณะของเส้นปกติ 3. ระบบทำการตรวจสอบสภาพเส้นทาง ได้แก่ เส้นทางตรง ทางโค้งและทางชัน 4. ปรับระดับความเร็วและควบคุมมอเตอร์ให้เคลื่อนที่ตามประเภทเส้นทาง 5. ระบบทำการติดตามเส้นทางปกติ และรับค่าเส้นสีปกติไปเรื่อย ๆ จนกว่าจะมีการรับค่าที่ระบุว่าเป็นเส้นทางทางลัด		
SubFlows : -		
Alternate/Exceptional Flows : 3.a. ระบบรับค่าที่ไม่ใช่เส้นทางตรง 3.a.1 หากค่าที่รับเข้ามาระบุว่าเป็นทางชันระบบจะต้องเตรียมปรับระดับความเร็วและควบคุมมอเตอร์สำหรับทางลาดชัน 3.a.2 หากมีการเคลื่อนที่ในลักษณะของสภาพเส้นทางโค้ง ระบบจะต้องทำการปรับระดับความเร็วและควบคุมมอเตอร์ให้เคลื่อนที่ไปตามลักษณะเส้นทางได้ 4.a. ระบบเคลื่อนที่ผ่านทางแยกที่มีเส้นทางลัด ระบบจะต้องทำการเปลี่ยนทิศทางให้เคลื่อนที่ไปตามเส้นทางลัด (Shortcut)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 แสดงคำอธิบายยูสเคสติดตามเส้นทางลัด (TraceShortcut)

Line trace robot		
Use Case Name : ติดตามเส้นทางลัด (TraceShortcut)	ID: 03	Importance Level: High
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : ระบบมีการตัดสินใจเลือกเส้นทาง		
Post-condition: : -		
Brief Description : ระบบทำการติดตามเส้นทางลัด (TraceShortcut) จนกว่าจะกลับสู่เส้นทางปกติ		
Trigger : เซนเซอร์แสง (Light Sensor)รับค่าข้อมูลที่เป็นเส้นทางลัด (Shortcut)		
Relationships : Association : มอเตอร์ (motor) Extend : DirectLine ,SlopeLine, CurveLine		
Normal Flow of Events : 1. ระบบรับค่าเข้ามาโดยระบุว่าเป็นเส้นทางลัด (Shortcut) 2. ระบบทำการเปลี่ยนลักษณะของการเคลื่อนที่เป็นการเคลื่อนที่ในเส้นทางลัด 3. ระบบทำการตรวจสอบสภาพเส้นทาง ได้แก่เส้นทางตรง และทางโค้ง 4. ปรับระดับความเร็วและควบคุมมอเตอร์ให้เคลื่อนที่ในลักษณะของเส้นทางลัด 5. ระบบทำการติดตามเส้นทางลัด (Shortcut) และรับไปเรื่อย ๆ จนกว่าจะมีการตรวจสอบและเคลื่อนที่กลับสู่เส้นทางปกติ		
SubFlows : -		
Alternate/Exceptional Flows : 5. a ระบบเคลื่อนที่ผ่านทางแยกที่มีการตัดสินใจเลือกเส้นทางอีกครั้ง ระบบจะต้องทำการเปลี่ยนทิศทางให้เคลื่อนที่ไปตามเส้นทางปกติ		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางตรง (DirectLine)

Line trace robot		
Use Case Name : เคลื่อนที่ไปในทางตรง(DirectLine)	ID: 04	Importance Level: medium
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : -		
Post-condition: : -		
Brief Description : เคลื่อนที่ไปตามเส้นทางตรง		
Trigger : เซนเซอร์แสง (Light Sensor) อ่านข้อมูลตามปกติ		
Relationships : Association : มอเตอร์ (motor) Include : - Extend : - Generalization : -		
Normal Flow of Events : 1. ระบบรับข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสง 2. ตรวจสอบค่าข้อมูลหากเป็นค่าเส้นทางตรง ให้เคลื่อนที่ไปตามเส้นที่กำหนด 3. ทำการปรับระดับความเร็วไปตามเส้นทาง และควบคุมมอเตอร์ในลักษณะปกติ 4. รับค่าข้อมูลและเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่ไปในสภาพเส้นทางที่ต่างจากเดิม		
SubFlows : -		
Alternate/Exceptional Flows : 3.a. ระบบทำการปรับความเร็วเพิ่มขึ้นในขณะที่เคลื่อนที่ไปในเส้นทางตรง 4.a. หากระบบทำการเคลื่อนที่แล้วมีค่าที่รับเข้ามาต่างไปจากเดิม ระบบจะต้องมีการตรวจสอบว่าเส้นทางที่เปลี่ยนไปมีลักษณะเช่นไร หรือมีการเลือกเส้นทางหรือไม่ และเตรียมปรับระดับความเร็วและทิศทางให้เคลื่อนที่ไปในสภาพเส้นทางนั้นได้		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางโค้ง (CurveLine)

Line trace robot		
Use Case Name : เคลื่อนที่ไปในทางโค้ง(CurveLine)	ID: 05	Importance Level: medium
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : -		
Post-condition: : -		
Brief Description : เคลื่อนที่ไปตามทางโค้ง(CurveLine)		
Trigger : เซนเซอร์แสง (Light Sensor) อ่านข้อมูลได้ค่าที่ถูกกำหนดว่าเส้นทางเป็นทางโค้ง		
Relationships : Association : มอเตอร์ (motor)		
Normal Flow of Events : 1. ระบบรับข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสง 2. ตรวจสอบค่าข้อมูลเป็นค่าเบี่ยงเบนไปจากทิศทางที่เป็นเส้นทางตรง 3. ระบบทำการตรวจสอบทิศทางการเคลื่อนที่ในลักษณะที่เป็นทางโค้ง 4. ทำการปรับลดระดับความเร็วไปตามเส้นทาง และควบคุมมอเตอร์ตามลักษณะเส้นทางที่เป็นทางโค้ง 5. รับค่าข้อมูลและเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่กลับสู่เส้นทางปกติหรือจนกว่าจะมีการเคลื่อนที่ในทิศทางอื่นที่ต่างออกไป		
SubFlows : -		
Alternate/Exceptional Flows : 2. a. ตรวจสอบค่าข้อมูลเป็นค่าเบี่ยงเบนไปจากทิศทางที่เป็นเส้นทางตรง 2.a.1 หากมีการเคลื่อนที่ออกนอกเส้นทางที่กำหนด ให้ทำการเคลื่อนที่เพื่อค้นหาและกลับเข้าสู่เส้นทางเดิม จากนั้นทำการเคลื่อนที่ต่อไปตามเส้นทางเดิม		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 แสดงคำอธิบายยูสเคส เคลื่อนที่ไปในทางลาดชัน (SlopeLine)

Line trace robot		
Use Case Name : เคลื่อนที่ไปในทางลาดชัน (SlopeLine)	ID: 06	Importance Level: medium
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : รับค่าข้อมูลที่ระบุว่าเป็นทางชัน		
Post-condition: : -		
Brief Description : เคลื่อนที่ไปตามทางชัน (SlopeLine)		
Trigger : เซนเซอร์แสง (Light Sensor) อ่านข้อมูลได้ค่าที่ถูกกำหนดว่าเส้นทางเป็นทางชัน		
Relationships : Association : มอเตอร์ (motor)		
Normal Flow of Events : 1. ระบบรับข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสง 2. ตรวจสอบค่าข้อมูลเป็นที่ถูกกำหนดว่าเส้นทางมีลักษณะเป็นเส้นทางชัน 3. ทำการปรับระดับความเร็วไปตามเส้นทาง และควบคุมมอเตอร์ตามลักษณะทิศทาง - ทางชันในทิศทางลาดขึ้น ให้ทำการปรับระดับความเร็วขึ้น - ทางชันในทิศทางลาดลง ให้ทำการลดระดับความเร็วลง 4. รับค่าข้อมูลและเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่กลับสู่เส้นทางปกติ		
SubFlows : -		
Alternate/Exceptional Flows :-		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 แสดงคำอธิบายยูสเคส เลือกเส้นทาง (SelectLine)

Line trace robot		
Use Case Name : เลือกเส้นทาง (SelectLine)	ID: 07	Importance Level: High
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : -		
Post-condition: : -		
Brief Description : ตัดสินใจเลือกเส้นทาง (SelectLine) ที่จะเคลื่อนที่ได้		
Trigger : เซนเซอร์แสง (Light Sensor)ได้รับข้อมูลสภาพเส้นทางที่ต้องมีการตัดสินใจเลือกลักษณะเส้นทางแบบเส้นทางลัดหรือแบบเส้นทางปกติ		
Relationships : Association : มอเตอร์ (motor)		
Normal Flow of Events : 1. ระบบรับข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสง 2. ตรวจสอบว่ามีเงื่อนไขของการตัดสินใจในการเลือกเส้นทางหรือไม่ 3. ทำการประมวลผลข้อมูลและตัดสินใจเลือกทิศทางที่จะไป 4. ปรับความเร็วและทิศทางของมอเตอร์ไปตามเส้นทางที่เลือก		
SubFlows : -		
Alternate/Exceptional Flows : 3.a. ระบบตัดสินใจเลือกเส้นทาง 3.a.1 หากกำหนดให้ตัดสินใจเลือกเส้นทางปกติ ให้เคลื่อนที่ตามปกติ 3.a.2 หากกำหนดให้ตัดสินใจเลือกเส้นทางลัด ให้เตรียมปรับระดับความเร็วและรูปแบบการเคลื่อนตามลักษณะของเส้นทางลัด		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

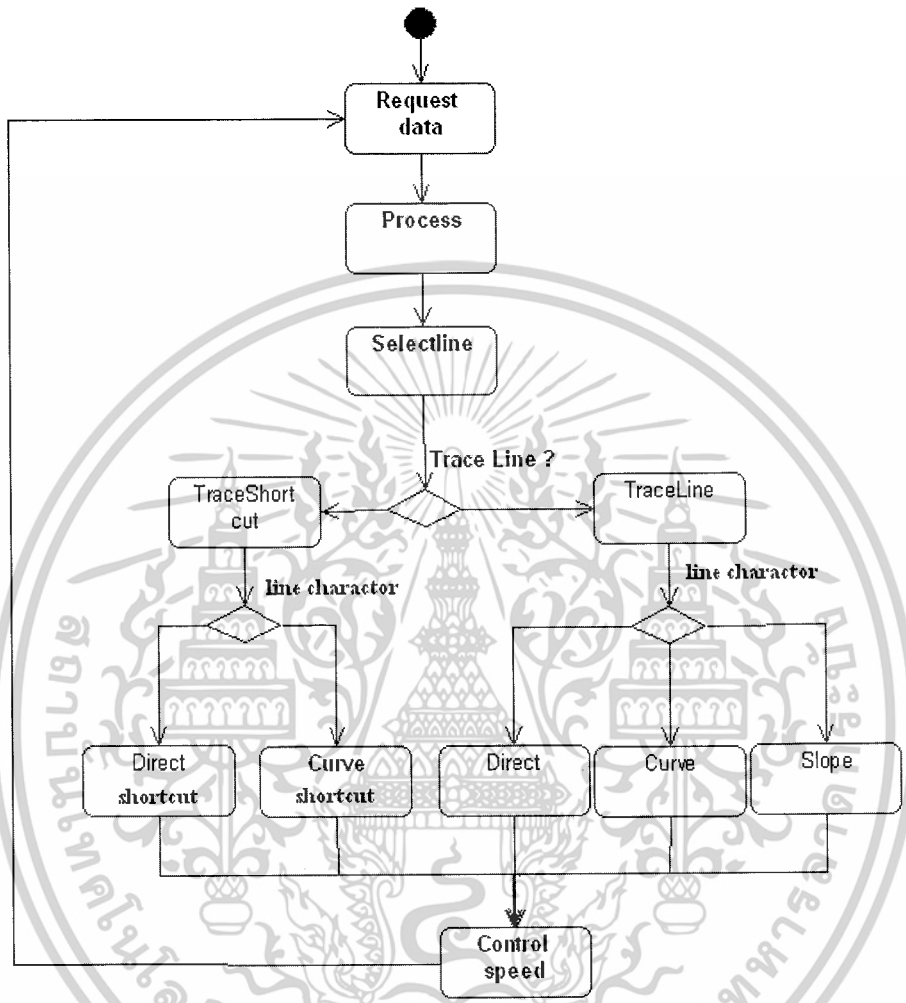
ตารางที่ 3.9 แสดงคำอธิบายยูสเคสควบคุมความเร็ว (ControlSpeed)

Line trace robot		
Use Case Name : ควบคุมความเร็ว (ControlSpeed)	ID: 08	Importance Level: High
Primary Actor : มอเตอร์ (motor)	Use Case Type : Function	
Stakeholders : เซนเซอร์แสง (Light Sensor)		
Precondition : -		
Post-condition: : -		
Brief Description : ควบคุมความเร็วมอเตอร์ตามลักษณะการเคลื่อนที่		
Trigger :		
Relationships : Association : มอเตอร์ (motor)		
Normal Flow of Events : 1. ระบบทำการตรวจสอบลักษณะเส้นทางและทิศทางของการเคลื่อนที่ ได้แก่ เส้นทางตรง เส้นทางโค้ง และเส้นทางลาดชัน 2. ปรับระดับความเร็วตามลักษณะของการเคลื่อนที่ 3. ส่งคำสั่งให้มอเตอร์ทำงานตามความเร็วและทิศทางการเคลื่อนที่นั้น		
SubFlows : -		
Alternate/Exceptional Flows :		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 แผนภาพกิจกรรม (Activity Diagram)

แผนภาพกิจกรรมภาพรวมระบบ

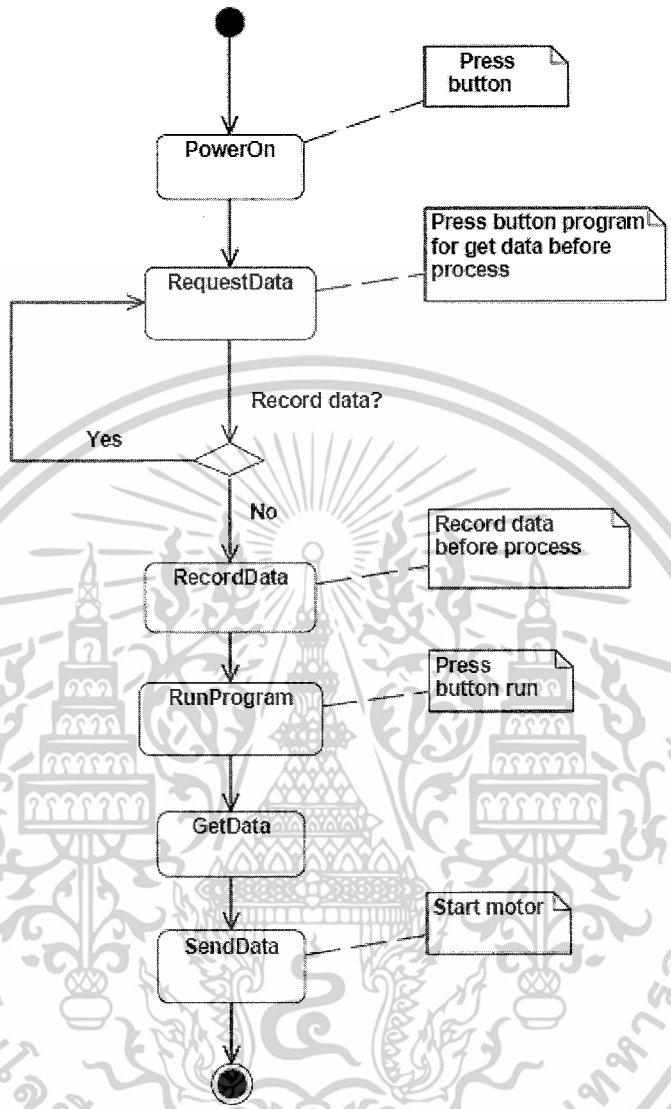


รูปที่ 3.4 แสดงแผนภาพกิจกรรมภาพรวมระบบ

จากแผนภาพกิจกรรมภาพรวมของทั้งระบบ สามารถอธิบายได้ดังนี้ เมื่อหุ่นยนต์เริ่มทำงาน จะมีการรับค่าข้อมูลเข้ามาประมวลผลและทำการตรวจสอบค่าที่ได้รับว่าตรงตามเงื่อนไขใดหรือไม่เพื่อปรับระดับความเร็วและทิศทางการเคลื่อนที่ได้ โดยเมื่อข้อมูลเข้ามาจะผ่านการประมวลผลข้อมูลดังกล่าวแล้วทำการตัดสินใจเลือกเส้นทางว่าเส้นทางนั้นเป็นเส้นทางปกติหรือเส้นทางลัด หากมีการเคลื่อนที่ในเส้นทางปกติ จะมีการเคลื่อนที่ไปในสภาพเส้นทางตรง ทางโค้ง และทางชัน ระบบจะต้องสามารถปรับรูปแบบการเคลื่อนที่ที่เหมาะสมได้ และหากมีการเคลื่อนที่ในลักษณะของเส้นทางลัดจะมีการเคลื่อนที่ไปในสภาพเส้นทางตรงและทางโค้ง ระบบจะต้องมีการปรับระดับความเร็วและรูปแบบในการเคลื่อนที่ให้เหมาะกับสภาพเส้นทางนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมร้องขอข้อมูล (RequestData)



รูปที่ 3.5 แสดงแผนภาพกิจกรรมการร้องขอข้อมูล (RequestData)

จากแผนภาพกิจกรรมการร้องขอข้อมูล สามารถอธิบายได้ดังนี้ เมื่อมีการกดปุ่มเปิดหรือรีเซตเครื่องแล้ว จะมีการร้องขอข้อมูลจากภายนอกเข้ามาและตรวจสอบข้อมูลที่รับเข้ามาว่าเคยถูกบันทึกไว้หรือยัง ถ้ายังให้ทำการกดปุ่ม Program เพื่อบันทึกค่าข้อมูลใหม่ไว้สำหรับประมวลผล คือระบบจะใช้การเก็บค่าแสง ณ สภาพแวดล้อมปัจจุบันมาใช้ในการโปรแกรมเพื่อสั่งงานระบบในสภาพแวดล้อมปัจจุบัน แต่ถ้ากรณีที่ข้อมูลเคยถูกบันทึกแล้วระบบจะทำการร้องขอข้อมูลเข้ามาใหม่เพื่อบันทึก และจะทำการรันโปรแกรมจากข้อมูลที่เคยบันทึกไว้โดยกดปุ่ม run ซึ่งระบบจะเริ่มทำงานโดยรับค่าใหม่เข้ามาเรื่อยๆเพื่อประมวลผลและนำผลที่ได้ไปควบคุมการเคลื่อนที่ของมอเตอร์

ตัวอย่างโปรแกรมการเก็บค่าจากสภาพแวดล้อมจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

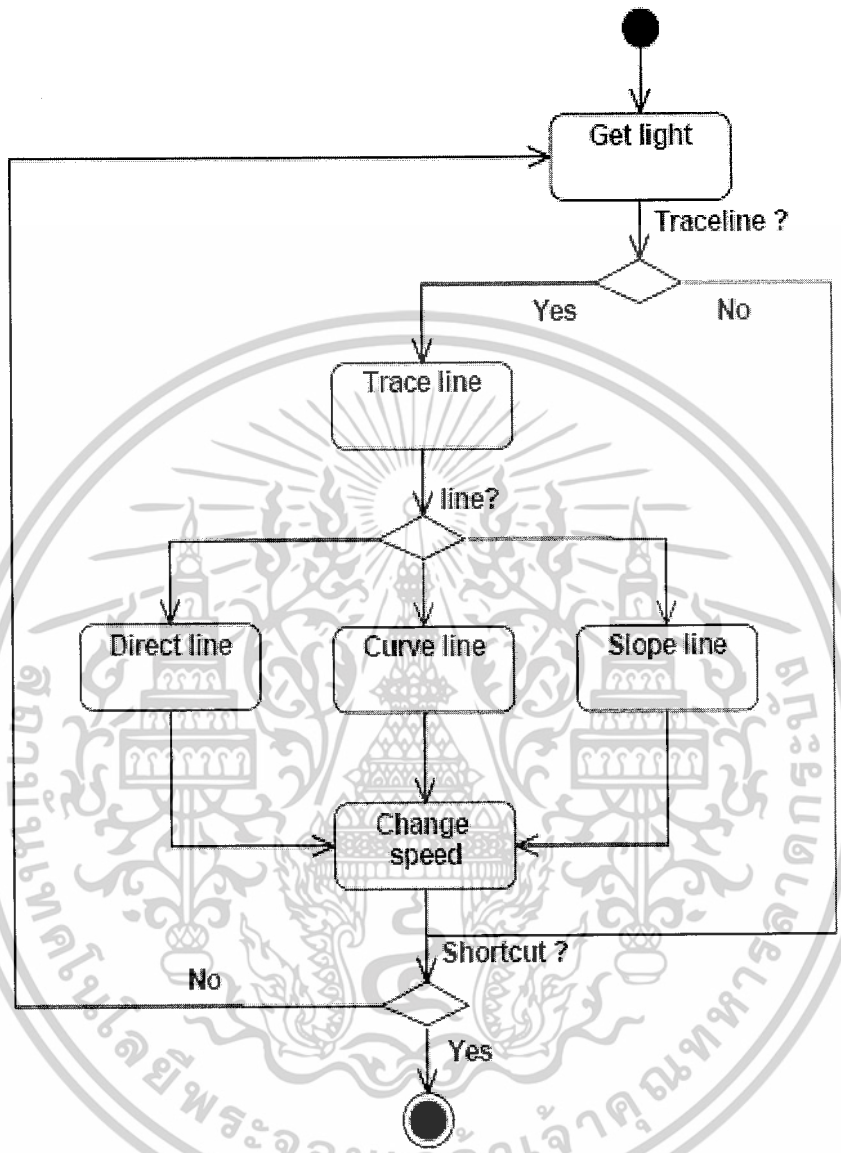
โปรแกรมนี้จะทำการเก็บค่าสีขาวและสีดำจากสภาพแวดล้อมจริง โดยที่การเก็บค่าสีแต่ละครั้งสามารถทำได้โดยการกดปุ่ม PROGRAM ซึ่งจะทำการบันทึกค่าสีแต่ละสีเก็บไว้ในตัวแปรที่ตั้งขึ้นมา สำหรับคำสั่งที่ใช้สำหรับเก็บค่าสีคือ sample() ซึ่งอาจใช้คำสั่ง get() แทนก็ได้ ซึ่งค่าที่ได้จะแสดงออกมาเป็นค่าเลขฐานสิบหก ซึ่งโปรแกรมนี้จะทำการแสดงค่าข้อมูลออกมาเป็นเลขฐานสิบ โดยการสร้างฟังก์ชันเพื่อแสดงข้อมูลดังกล่าวคือ ฟังก์ชัน cputw10

```
cputs("Whit");
sleep(1);
while(!PRESSED(dbutton(),BUTTON_PROGRAM)){
        white = ls->sample();
        cputw10(white);
}
dsound_system(DSOUND_BEEP);
cputs("Blak");
sleep(1);
while(!PRESSED(dbutton(),BUTTON_PROGRAM)){
        black = ls->sample();
        cputw10(black);
}
dsound_system(DSOUND_BEEP);

// begin running
while (1)
{
        if(PRESSED(dbutton(),BUTTON_RUN))  break;
        lightLevel=ls->sample();
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการติดตามเส้นทางปกติ (TraceLine)



รูปที่ 3.6 แสดงแผนภาพกิจกรรม การติดตามเส้นทางปกติ (TraceLine)

จากแผนภาพกิจกรรมการติดตามเส้นทางปกติ สามารถอธิบายได้ดังนี้ คือ ระบบรับค่าแสงเข้ามาแล้วทำการตรวจสอบค่าแสงว่าเป็นลักษณะของเส้นปกติหรือไม่ ถ้าเป็นเส้นปกติให้เคลื่อนที่แบบเส้นปกติและจะมีการตรวจสอบสภาพเส้นทางที่เป็นทางตรง ทางโค้งและทางชันและจะมีการปรับระดับความเร็วที่ใช้ในการควบคุมตามมอเตอร์ให้เหมาะสำหรับการเคลื่อนที่ในแต่ละลักษณะ โดยระบบจะทำการติดตามเส้นทางปกติ และรับค่าเส้นสีปกติไปจนกว่าจะมี การรับค่าที่ระบุว่าเป็นเส้นทางทางลัด

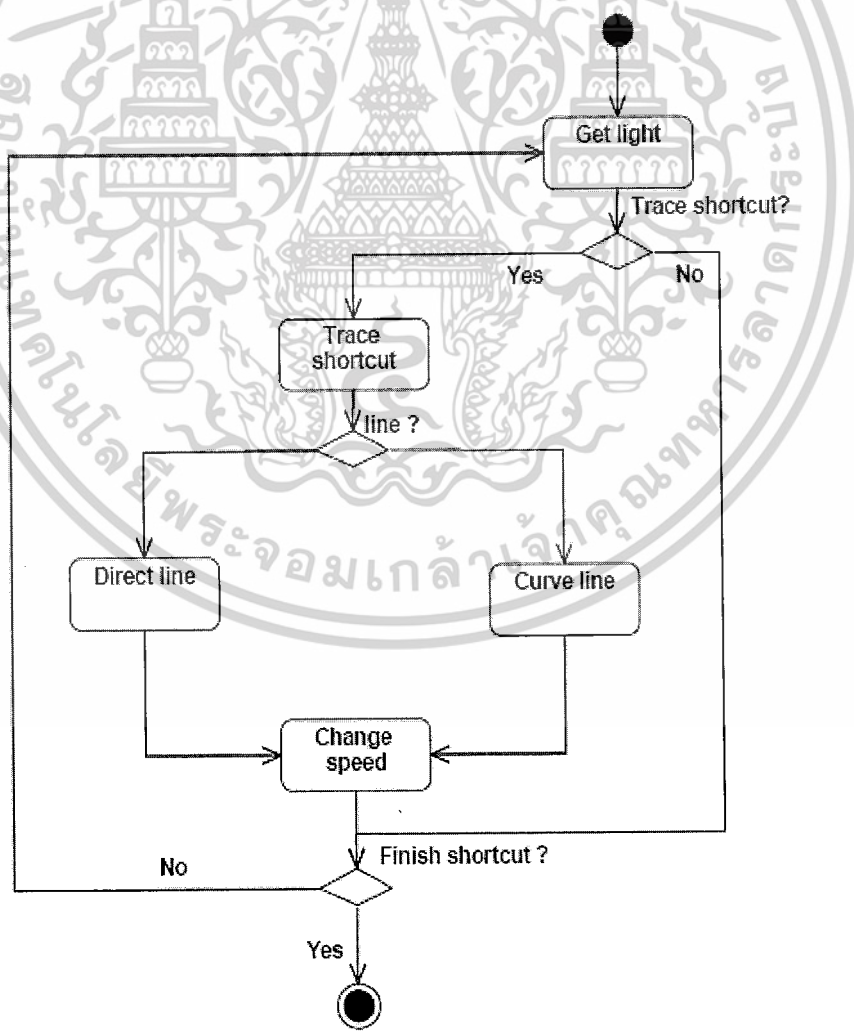
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมการตัดสินใจเลือกเส้นทาง

โปรแกรมนี้จะทำการตรวจสอบค่าสีเพื่อทำการตัดสินใจเลือกการเคลื่อนที่

```
if( lightLevel >= (white+3) )//slope
        driving_slope();
else if(lightLevel < black-3)//shortcut
        driving_shortcut();
else // normal line
        driving_line(speed);
```

แผนภาพกิจกรรม การติดตามเส้นทางลัด (TraceShortCut)

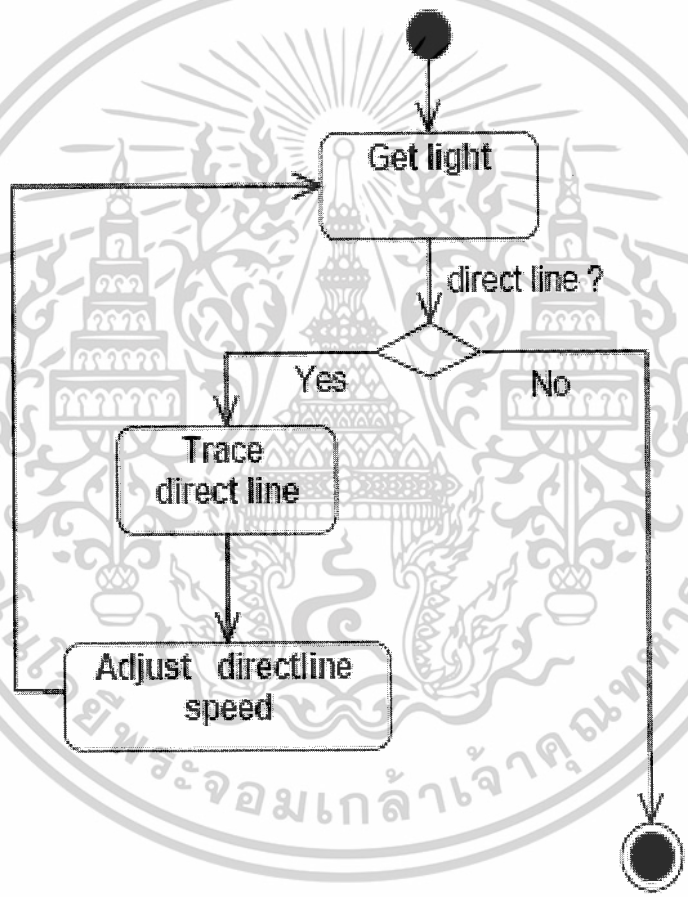


รูปที่ 3.7 แสดงแผนภาพกิจกรรม การติดตามเส้นทางลัด (TraceShortCut)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพกิจกรรมการติดตามเส้นทางลัด สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าแสงเข้ามาระบบจะมีการตรวจสอบค่าว่าเป็นลักษณะเส้นทางลัดหรือไม่ ถ้าเป็นเส้นทางลัดระบบจะมีการเคลื่อนที่ในลักษณะที่เหมาะสมกับเส้นทางลัดโดยระบบจะทำการตรวจสอบสภาพเส้นทางอีกรอบว่ามีลักษณะใด ได้แก่เส้นทางตรง และทางโค้งจากนั้นปรับระดับความเร็วที่ใช้ควบคุมมอเตอร์ให้เหมาะสมกับการเคลื่อนที่ในลักษณะของเส้นทางลัดซึ่งระบบจะทำการติดตามเส้นทางลัดไปเรื่อย ๆ จนกว่าจะมีการตรวจสอบพบเส้นทางปกติ

แผนภาพกิจกรรมเคลื่อนที่ไปในทางตรง (DirectLine)



รูปที่ 3.8 แสดงแผนภาพกิจกรรม เคลื่อนที่ไปในทางตรง (DirectLine)

จากแผนภาพกิจกรรมการติดตามเส้นทางตรง สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าแสงเข้ามาระบบจะตรวจสอบค่าข้อมูลที่รับเข้ามานั้นหากเป็นค่าเส้นทางตรงระบบจะเคลื่อนที่ไปตามเส้นที่กำหนดโดยทำการปรับระดับความเร็วของมอเตอร์ให้เหมาะสมกับลักษณะเส้นทางตรง จากนั้นจะเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่ไปในสภาพเส้นทางที่ต่างจากเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

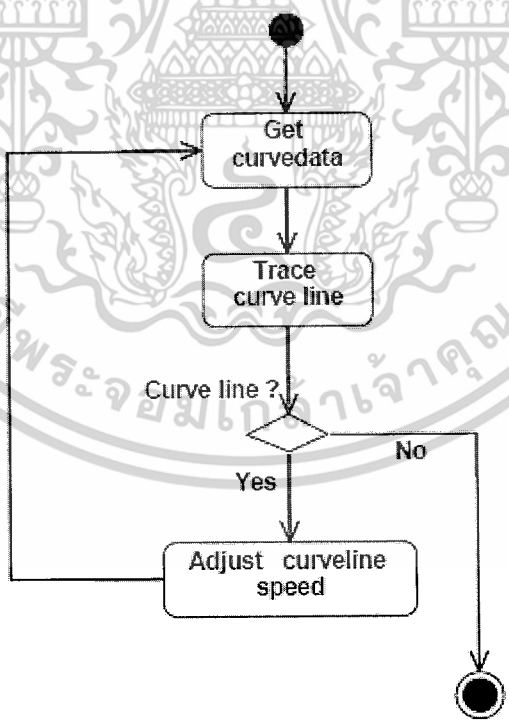
ตัวอย่างโปรแกรมสำหรับการเคลื่อนที่ในแนวเส้นทางตรง

สำหรับโปรแกรมนี้จะทำการเคลื่อนที่โดยการตรวจสอบระหว่างค่าสีขาวและสีดำ โดยเมื่อทำการเก็บค่าข้อมูลเพื่อตรวจสอบค่าสีขาวและดำแล้ว จะทำการควบคุมสเตียริงมอเตอร์และไดร์วิงมอเตอร์ โดยที่ใช้คำสั่งดังต่อไปนี้คือ

```
//สร้างอ็อบเจ็คมอเตอร์จากคลาสมอเตอร์เพื่อเรียกใช้งาน
        sm = new Motor(Motor::A);       //steering
        dm = new Motor(Motor::C);       //driving

        dm->forward(100);  //สั่งงานให้ไดร์วิงมอร์เตอร์เคลื่อนที่ไปข้างหน้าด้วยความเร็ว 100
        sm->reverse(40);  //  สั่งงานให้สเตียริงมอร์เตอร์เคลื่อนที่ไปทางซ้ายด้วยความเร็ว 40
```

แผนภาพกิจกรรม เคลื่อนที่ไปในทางโค้ง (CurveLine)



รูปที่ 3.9 แสดงแผนภาพกิจกรรม เคลื่อนที่ไปในทางโค้ง (CurveLine)

แผนภาพกิจกรรมการเคลื่อนที่ไปในทางโค้ง สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสงเข้ามา และตรวจสอบค่าข้อมูลเป็นค่าเบี่ยงเบนไปจากทิศทางที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเส้นทางตรง ระบบจะทำการตรวจสอบทิศทางการเคลื่อนที่ในลักษณะที่เป็นทางโค้งและทำการปรับลดระดับความเร็วไปตามเส้นทาง และควบคุมมอเตอร์ตามลักษณะ เส้นทางที่เป็นทางโค้ง และจะมีการรับค่าข้อมูลและเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่กลับสู่เส้นทางปกติหรือจนกว่าจะมีการเคลื่อนที่ในทิศทางอื่นที่ต่างออกไป

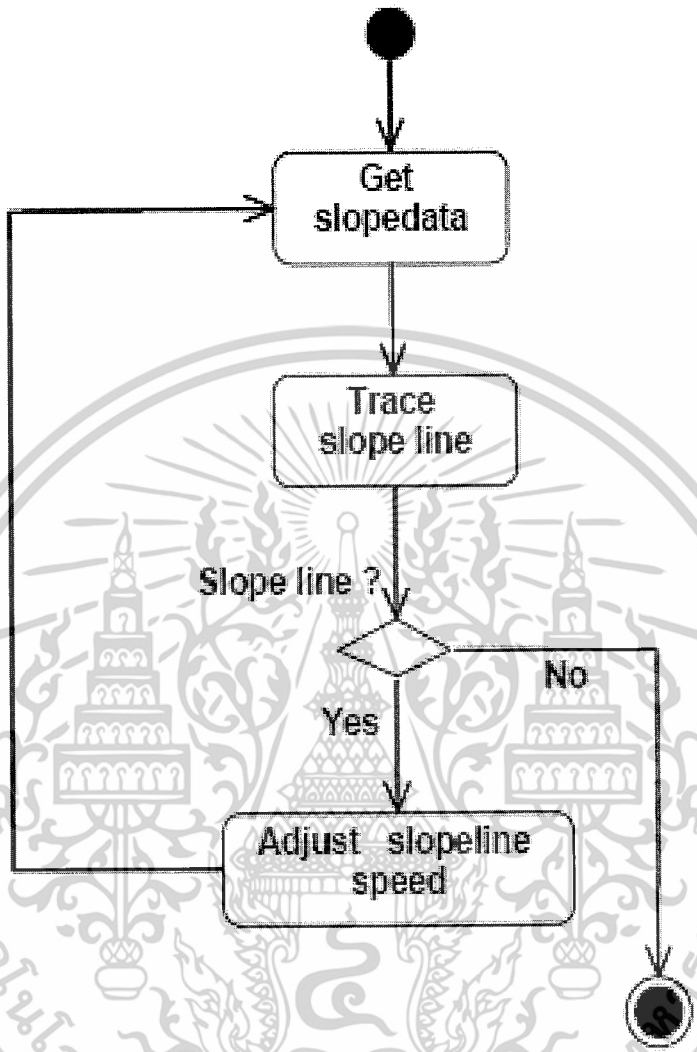
ตัวอย่างโปรแกรมเพื่อสั่งงานให้เคลื่อนที่ไปในเส้นทางโค้ง

โปรแกรมนี้จะทำการตรวจสอบค่าเมื่ออยู่ภายในบริเวณสีขาวเกินกว่าที่กำหนด จะทำการเรียกฟังก์ชันการเคลื่อนที่สำหรับเส้นทางโค้ง โดยจะทำการลดระดับความเร็วลง เมื่อตรวจสอบค่าและยังพบค่าสีขาวจะทำการเคลื่อนที่ถอยหลังเพื่อค้นหาเส้น

```
if (lightLevel > (black_white+2)) // now in white
        {
        dm->forward(limit_speed-10); //low speed
        sm->forward(righ+10);  //move to white_line --> forward steering more turning
radius
                count_w++;
                if(count_w>=10)  //10  count_w->check in white and turn on curve
                {
                        dsound_system(DSOUND_BEEP);
                        sm->forward(righ+15); //  forward steering more turning radius
                        dm->forward(-30); //to back OR no speed dm->forward(stop)
                        msleep(100);
                        while(lightLevel >(black_white+5))  //check in white
                                lightLevel= ls->sample();
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมเคลื่อนที่ไปในทางลาดชัน (SlopeLine)



รูปที่ 3.10 แสดงแผนภาพกิจกรรมเคลื่อนที่ไปในทางลาดชัน (Slope Line)

แผนภาพกิจกรรมการเคลื่อนที่ไปในทางลาดชัน สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสงเข้ามา ระบบจะตรวจสอบค่าข้อมูลว่าถูกกำหนดว่าเส้นทางมีลักษณะเป็นเส้นทางชันหรือไม่ ถ้าเป็นลักษณะเป็นเส้นทางชันจะทำการปรับระดับความเร็วไปตามเส้นทาง และควบคุมมอเตอร์ตามลักษณะทิศทาง

- ทางชันในทิศทางลาดขึ้น ให้ทำการปรับระดับความเร็วขึ้น
- ทางชันในทิศทางลาดลง ให้ทำการลดระดับความเร็วลง

และจะรับค่าข้อมูลและเคลื่อนที่ไปตามเส้นทางจนกว่าจะมีการเคลื่อนที่กลับสู่เส้นทางปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

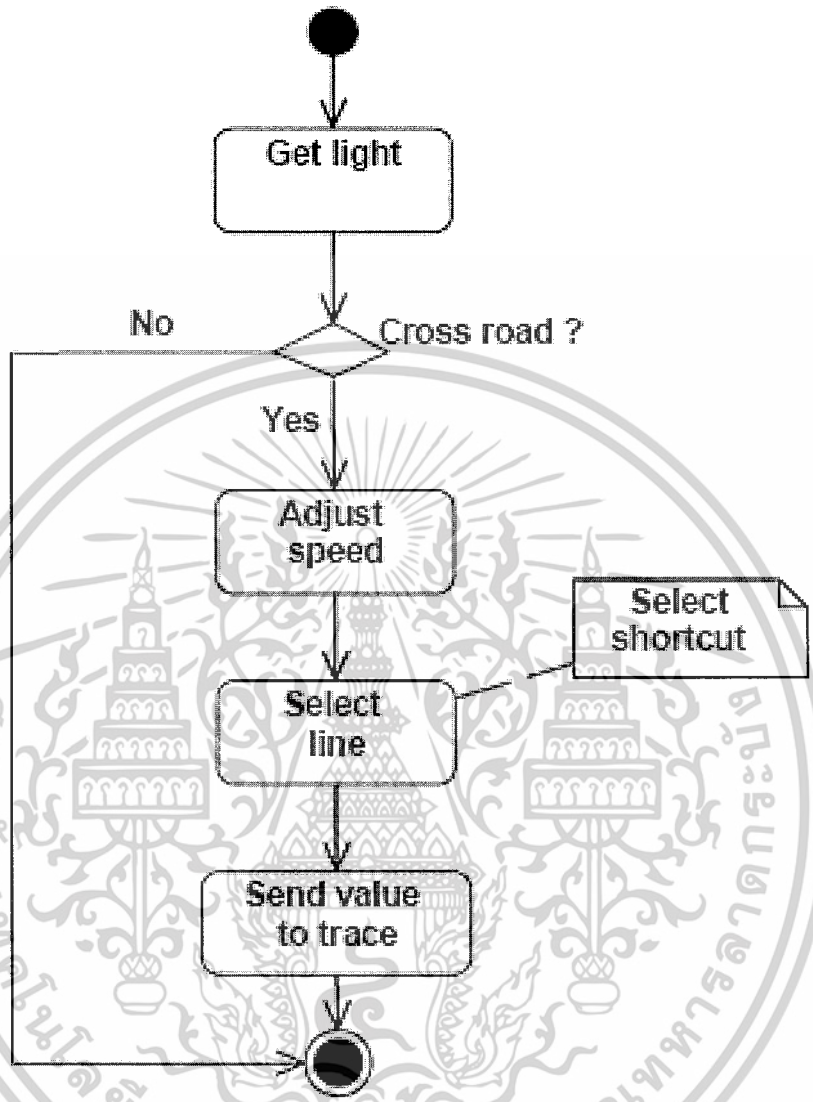
ตัวอย่างโปรแกรมสำหรับการเคลื่อนที่ไปในเส้นทางชัน

โปรแกรมนี้จะทำการตรวจสอบค่าสีที่แสดงว่าเป็นเส้นทางชัน โดยที่เมื่อเจอค่าสีที่กำหนดไว้ครั้งแรก จะทำการเร่งความเร็วเนื่องจากแสดงว่าเป็นเส้นทางลาดขึ้น เมื่อเจอค่าสีครั้งที่สองจะทำการลดความเร็วหรือกำหนดให้หุ่นยนต์เคลื่อนที่โดยไม่กำหนดความเร็ว หรือกำหนดให้หุ่นยนต์เคลื่อนที่โดยการเบรคเพื่อลดความเร็วของการเคลื่อนที่ในทิศทางลาดลง ในที่นี้จะใช้ค่า flag เป็นตัวนับการเปลี่ยนแปลง

```
if( lightLevel >= (white+3) )  // check detect slope
        while(flag_slope)
        {
                if(flag_slope==1)          driving_line(slope_speed); // move up
                else if (flag_slope==2) // move on center slope
                {
                        driving_line(slope_speed-90);
                }
                else  //move down
                {
                        sm->reverse(lef+10);
                        dm->forward(stop);  // no speed  OR to back [ dm->forward(-40)]
                        sleep(1);
                        flag_slope=0;
                }
                lightLevel= ls->sample();
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรม เลือกเส้นทาง (SelectLine)

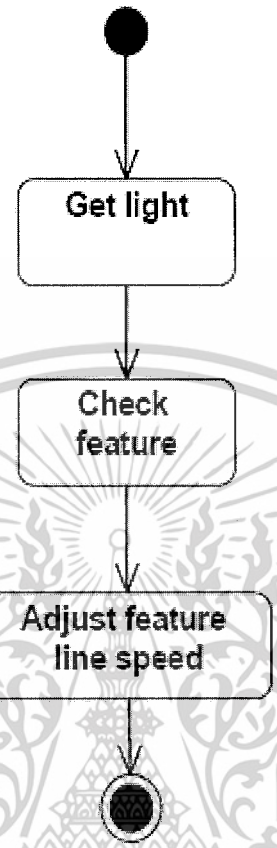


รูปที่ 3.11 แสดงแผนภาพกิจกรรม เลือกเส้นทาง (SelectLine)

แผนภาพกิจกรรม เลือกเส้นทาง สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสงเข้ามา และมีการตรวจสอบว่ามีเงื่อนไขของการตัดสินใจในการเลือกเส้นทางหรือไม่ ถ้ามีจะทำการประมวลผลข้อมูลและตัดสินใจเลือกทิศทางที่จะเคลื่อนที่โดยมีการปรับความเร็วและทิศทางของมอเตอร์ไปตามเส้นทางที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรม ควบคุมความเร็ว (ControlSpeed)



รูปที่ 3.12 แสดงแผนภาพกิจกรรม ควบคุมความเร็ว (ControlSpeed)

แผนภาพกิจกรรม ควบคุมความเร็ว สามารถอธิบายได้ดังนี้ คือ เมื่อระบบมีการรับค่าข้อมูลเส้นทางเข้ามาจากเซนเซอร์แสงเข้ามา ระบบจะทำการตรวจสอบลักษณะเส้นทางและทิศทางของการเคลื่อนที่ ได้แก่ เส้นทางตรง เส้นทางโค้ง และเส้นทางลาดชัน จากนั้นจะปรับระดับความเร็วตามลักษณะของการเคลื่อนที่ส่งคำสั่งให้มอเตอร์ทำงานตามความเร็วและทิศทางการเคลื่อนที่นั้น

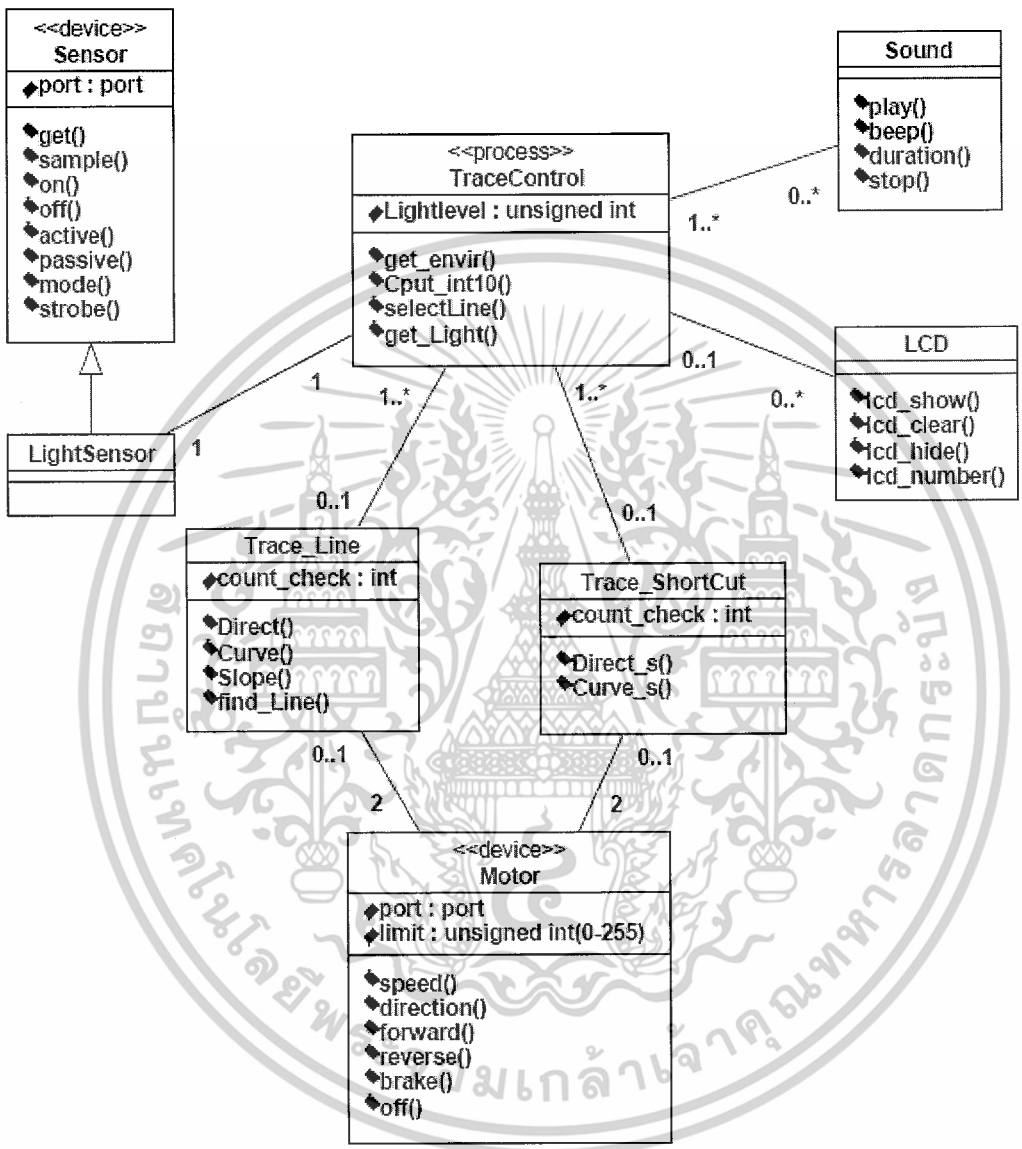
ตัวอย่างโปรแกรมสำหรับการควบคุมความเร็วประกอบไปด้วยคำสั่งดังต่อไปนี้

```
dm->forward(stop);
dm->forward(stop);
dm -> brake();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.3 คลาสไดอะแกรม

จากการวิเคราะห์ความต้องการและการทำงานของระบบ สามารถนำข้อมูลที่ออกแบบไว้มาเขียนแผนภาพคลาสไดอะแกรม (Class Diagram) ได้ดังนี้



รูปที่ 3.13 แสดงคลาสไดอะแกรมของระบบหุ่นยนต์ติดตามเส้นทาง

จากคลาสไดอะแกรมซึ่งแสดงคลาสต่าง ๆ ที่มีในสามารถอธิบายได้ดังนี้ คลาส TraceControl ทำหน้าที่ในการรับค่าข้อมูลและประมวลผลเพื่อตัดสินใจเลือกให้หุ่นยนต์เคลื่อนที่ไปตามลักษณะที่ต้องการ โดยจะมีการรับค่าข้อมูลจากสภาพแวดล้อมจริงก่อนที่จะเริ่มทำงานหรือรันโปรแกรม โดยจะเก็บค่าแสงจากเงื่อนไขต่าง ๆ ของการทำงานของระบบเพื่อใช้ในการประมวลผลจริงตอนที่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มทำการติดตามเส้นทาง นอกจากนี้ยังประกอบไปด้วยฟังก์ชันที่ใช้ในการแสดงค่าเลขฐานสิบ และฟังก์ชั่นการตัดสินใจเลือกเส้นทาง โดยคลาสนี้จะมีความสัมพันธ์กับคลาส LightSensor ซึ่งจะใช้ LightSensor จำนวนหนึ่งตัว ซึ่งจะนำข้อมูลที่ได้จากคลาส LightSensor เข้ามาทำการประมวลผล โดยคลาส LightSensor เป็นคลาสที่ถูกเตรียมไว้ให้เรียกใช้งานได้เลยจากบริคโอเอส โดยจะมีการสืบทอดคุณสมบัติมาจากคลาส Sensor ซึ่งเป็นคลาสที่อยู่ในบริคโอเอสเช่นเดียวกัน

นอกจากนี้คลาส TraceControl ยังมีความสัมพันธ์กับคลาส Sound และ คลาส LCD ซึ่งเป็นคลาสที่ใช้ในการแสดงผลทางเสียงและแสดงผลทางหน้าจอของอาร์ซีเอ็กซ์ ซึ่งทั้งสองคลาสเป็นคลาสที่ถูกจัดเตรียมไว้ในบริคโอเอสเช่นเดียวกัน เมื่อคลาส TraceControl ทำการประมวลผลจะส่งข้อมูลระหว่างกันกับคลาสอีกสองคลาสคือคลาส Trace_Line และ คลาส Trace_ShortCut

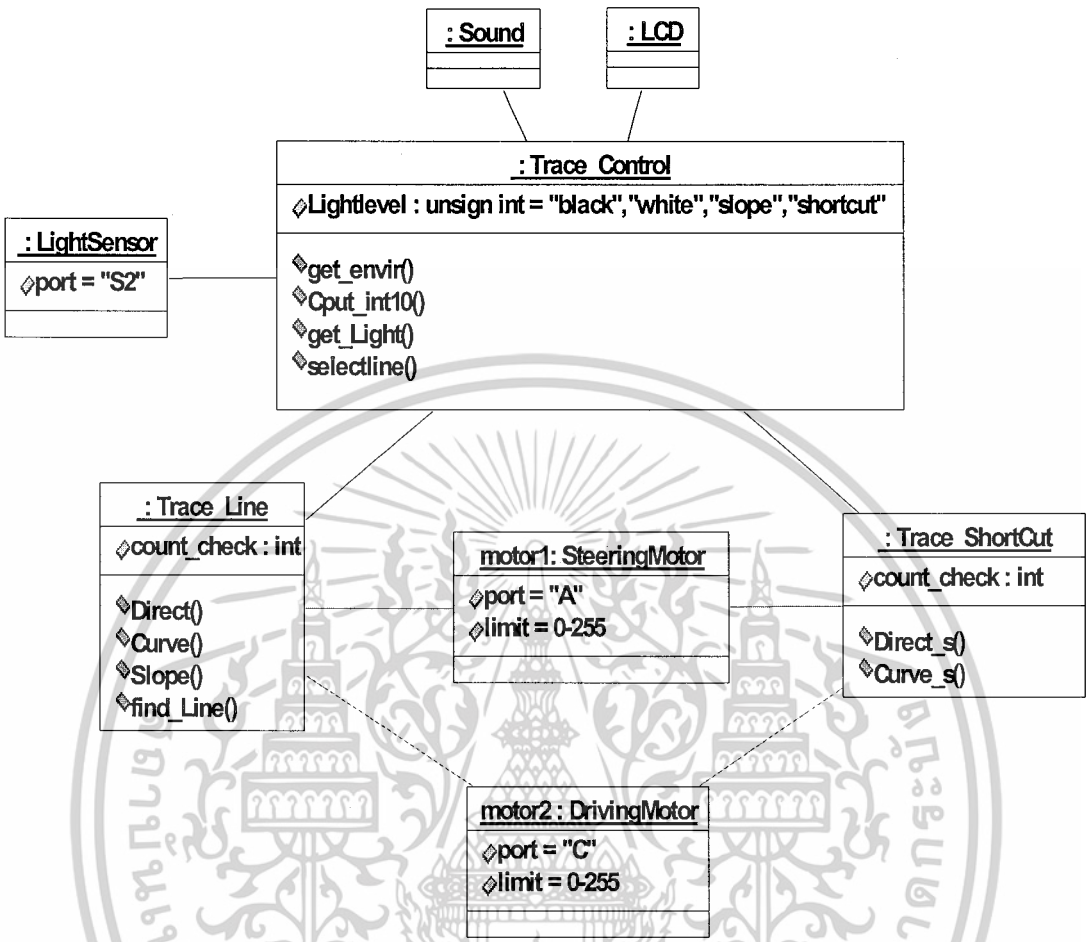
คลาส Trace_Line ทำหน้าที่ในการเคลื่อนที่ไปในเส้นทางปกติ ซึ่งจะประกอบไปด้วยการเคลื่อนที่ในลักษณะที่เป็นเส้นทางตรง เส้นทางโค้งและเส้นทางชัน นอกจากนี้ยังมีฟังก์ชันที่รองรับการทำงานในกรณีที่ระบบเคลื่อนที่หลุดออกจากเส้นทางอีกด้วย

คลาส Trace_ShortCut เป็นคลาสที่ทำหน้าที่ในการเคลื่อนที่ในลักษณะเส้นทางลัด (ShortCut) ซึ่งจะทำการเคลื่อนที่ในลักษณะเส้นทางที่เป็นเส้นทางตรงและทางโค้ง โดยที่เส้นทางลัดจะมีการควบคุมมอเตอร์และการปรับระดับความเร็วด้วยวิธีการที่ต่างจากเส้นทางปกติ

เมื่อทั้งคลาส Trace_Line และ คลาส Trace_ShortCut จะมีความสัมพันธ์กับคลาส Motor ซึ่งทำหน้าที่ในการควบคุมมอเตอร์ ซึ่งในที่นี้จะใช้มอเตอร์จำนวนสองตัวด้วยกันดังแสดงในอ็อบเจ็คไดอะแกรมในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.4 อ็อบเจ็คไดอะแกรม



รูปที่ 3.14 แสดงแผนภาพ Object Diagram

จากอ็อบเจ็คไดอะแกรมสามารถอธิบายได้ดังนี้ เมื่อระบบเกิดการทำงานจะทำการสร้างอ็อบเจ็คจากคลาสที่ได้ออกแบบไว้ โดยจะใช้เซนเซอร์แสงจำนวนหนึ่งตัวซึ่งจะทำการต่อกับพอร์ตสอง และใช้มอเตอร์จำนวนสองตัวคือ SteeringMotor ซึ่งทำการต่อกับพอร์ต A และ Driving Motor ซึ่งทำการต่อกับพอร์ต C ส่วนคลาสอื่น ๆ มีหน้าที่การทำงานดังที่อธิบายในหัวข้อก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายคลาส

คำอธิบายคลาสสามารถอธิบายโดย CRC Card ได้ดังนี้

ตารางที่ 3.10 แสดง CRC Card ของคลาส Sensor

Front:

Class Name: **Sensor**	ID: 1	Type:
Description:		Associated Use Cases: RequestData
Responsibilities - get - sample - on - off - active - passive - mode - strobe		Collaborators - TraceControl

Back:

Attributes: port : port
Relationships: Generalization (a-kind-of): Superclass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 แสดง CRC Card ของคลาส LightSensor

Front:

Class Name: **LightSensor**	ID: 2	Type:
Description: ใช้รับค่าข้อมูลจากสภาพแวดล้อม		Associated Use Cases: RequestData
Responsibilities - get - sample - on - off - active - passive - mode - strobe		Collaborators - TraceControl

Back:

Attributes: port : port
Relationships: Generalization (a-kind-of): Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 แสดง CRC Card ของคลาส TraceControl

Front:

Class Name: **TraceControl**	ID: 3	Type:
Description: ใช้ในการรับค่าข้อมูลและประมวลผลเพื่อตัดสินใจเลือกให้เคลื่อนที่ตามลักษณะที่ต้องการ		Associated Use Cases: ControlSpeed
Responsibilities - get_envir - Cput_int10 - selectLine - get_Light		Collaborators - LightSensor - TraceLine - TraceShortCut - LCD - Sound

Back:

Attributes: value : unsigned int port : port
Relationships: Generalization (a-kind-of):

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.13 แสดง CRC Card ของคลาส Trace_Line

Front:

Class Name: **Trace_Line**	ID: 5	Type:
Description: ใช้ในการควบคุมการเดินแบบเส้นทางปกติ		Associated Use Cases: TraceLine
Responsibilities - Direct - Curve - Slope		Collaborators - TraceControl - Motor

Back:

Attributes: valueLine : int
Relationships: Generalization (a-kind-of): Trace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 แสดง CRC Card ของคลาส Trace_ShortCut

Front:

Class Name: **Trace_ShortCut**	ID: 6	Type:
Description: ใช้ในการควบคุมการเดินแบบเส้นทางลัด		Associated Use Cases: TraceShortCut
Responsibilities - Direct_s - Curve_s		Collaborators - TraceControl - Motor

Back:

Attributes: valueShortCut : int
Relationships: Generalization (a-kind-of): Trace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.15 แสดง CRC Card ของคลาส Motor

Front:

Class Name: **Motor**	ID: 8	Type:
Description: ใช้ขับเคลื่อนระบบ		Associated Use Cases: ControlSpeed, TraceLine, TraceShortCut, SelectLine
Responsibilities - speed - direction - forward - reverse - brake - off		Collaborators - Trace_Line - Trace_ShortCut

Back:

Attributes: port : port limit : unsigned int(0-255)
Relationships: Generalization (a-kind-of):

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.16 แสดง CRC Card ของคลาส DrivingMotor

Front:

Class Name: **DrivingMotor**	ID: 9	Type:
Description: ใช้ขับเคลื่อนหุ่นยนต์		Associated Use Cases: ControlSpeed, TraceLine, TraceShortCut, SelectLine
Responsibilities - speed - direction - forward - reverse - brake - off		Collaborators - Trace_Line - Trace_ShortCut

Back:

Attributes: port : port limit : unsigned int(0-255)
Relationships: Generalization (a-kind-of): Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 แสดง CRC Card ของคลาส SteeringMotor

Front:

Class Name: **SteeringMotor**	ID: 10	Type:
Description: ใช้ควบคุมทิศทางหุ่นยนต์		Associated Use Cases: ControlSpeed, TraceLine, TraceShortCut, SelectLine
Responsibilities - speed - direction - forward - reverse - brake - off		Collaborators - Trace_Line - Trace_ShortCut

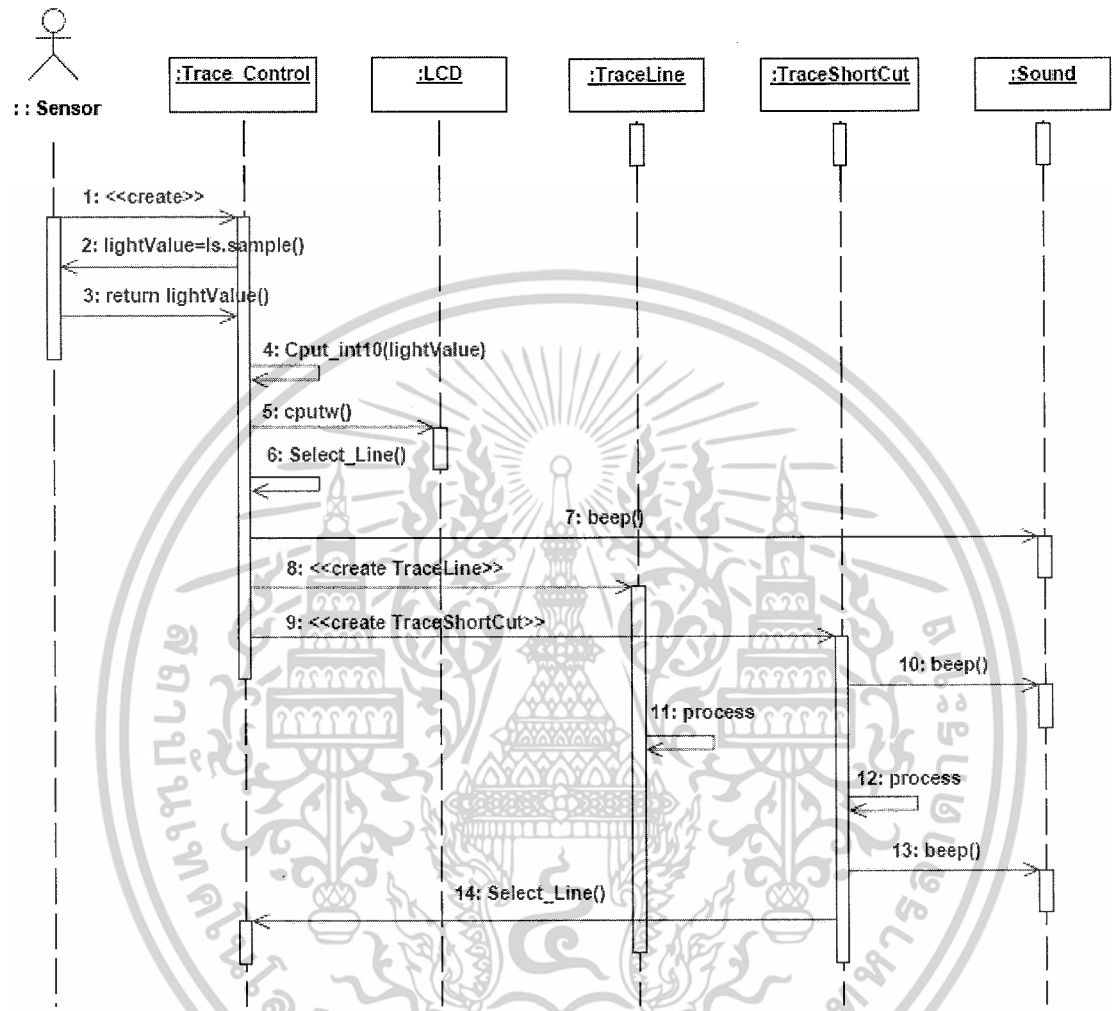
Back:

Attributes: port : port limit : unsigned int(0-255)
Relationships: Generalization (a-kind-of): Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.5 ซีเควนไดอะแกรม

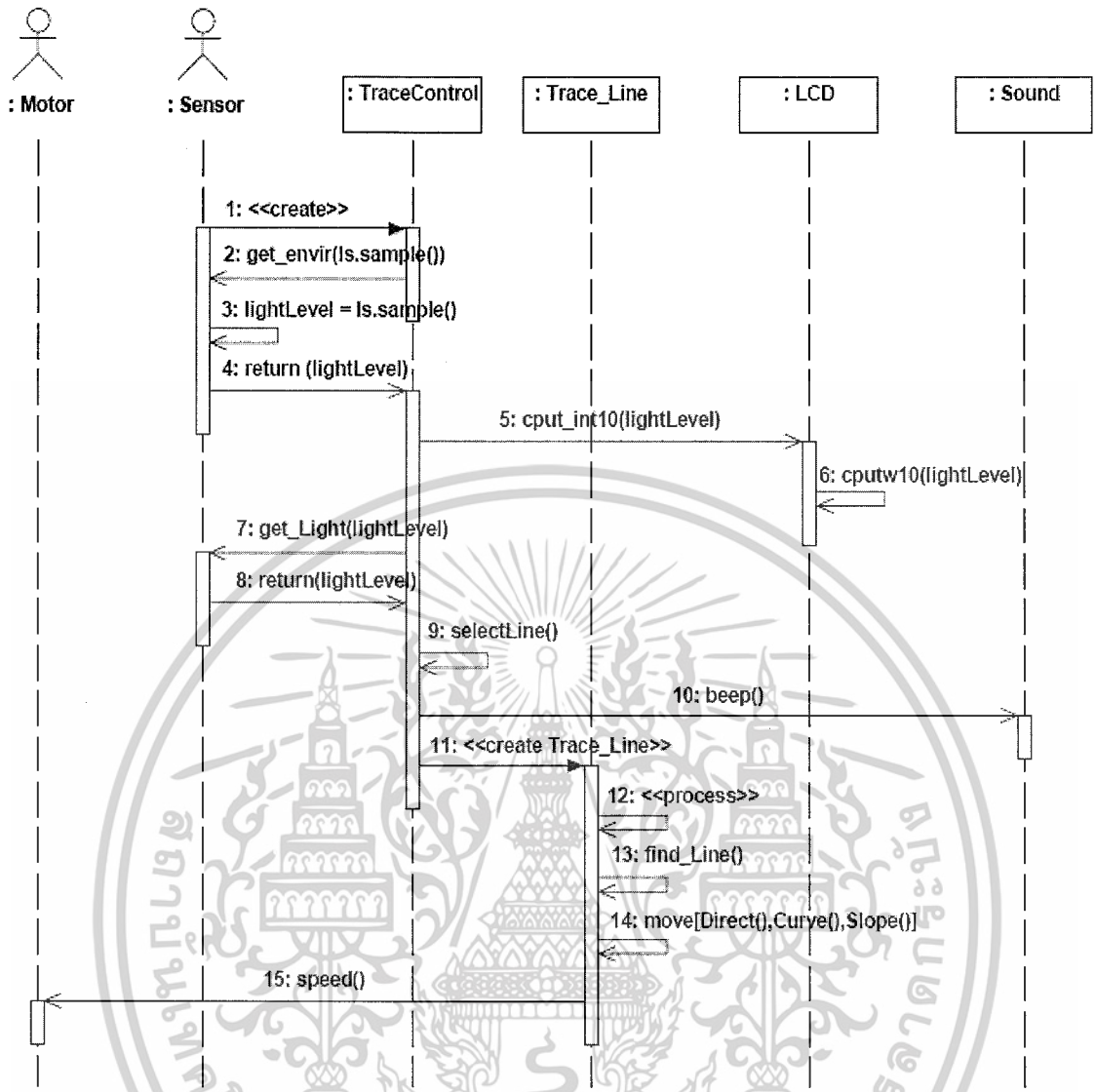
จากการวิเคราะห์ระบบงาน และการออกแบบคลาสไดอะแกรม (Class Diagram) สามารถอธิบายองค์ประกอบและความสัมพันธ์ได้ดังนี้



รูปที่ 3.15 แสดงแผนภาพ Sequence Diagram ของ RequestData

จากแผนภาพซีเควนไดอะแกรมของการรับข้อมูลและเลือกเส้นทางสามารถอธิบายได้ดังนี้ คือ ระบบจะทำการรับค่าข้อมูลจากสภาพแวดล้อมเข้ามาประมวลผลที่คลาส Trace_Control และทำการแสดงค่าเป็นเลขฐานสิบที่หน้าจอ โดยเรียกใช้ฟังก์ชันของการแสดงผลจากคลาส LCD จากนั้นหากมีการตัดสินใจเลือกเส้นทางระบบจะแสดงเสียงโดยเรียกคำสั่งจากคลาส Sound เพื่อบอกว่าระบบกำลังมีการตัดสินใจเลือกเส้นทาง เมื่อระบบทำการตัดสินใจเลือกเส้นทางแล้ว หากเป็นเส้นทางปกติระบบจะเคลื่อนที่ต่อไป แต่หากเป็นเส้นทางลัดระบบจะแสดงเสียงก่อนเข้าสู่สภาพเส้นทางที่เป็นทางลัดและหลังจากออกจากสภาพเส้นทางที่เป็นทางลัด จากนั้นก็จะกลับไปทำงานยังคลาส Trace_Control ต่อไป

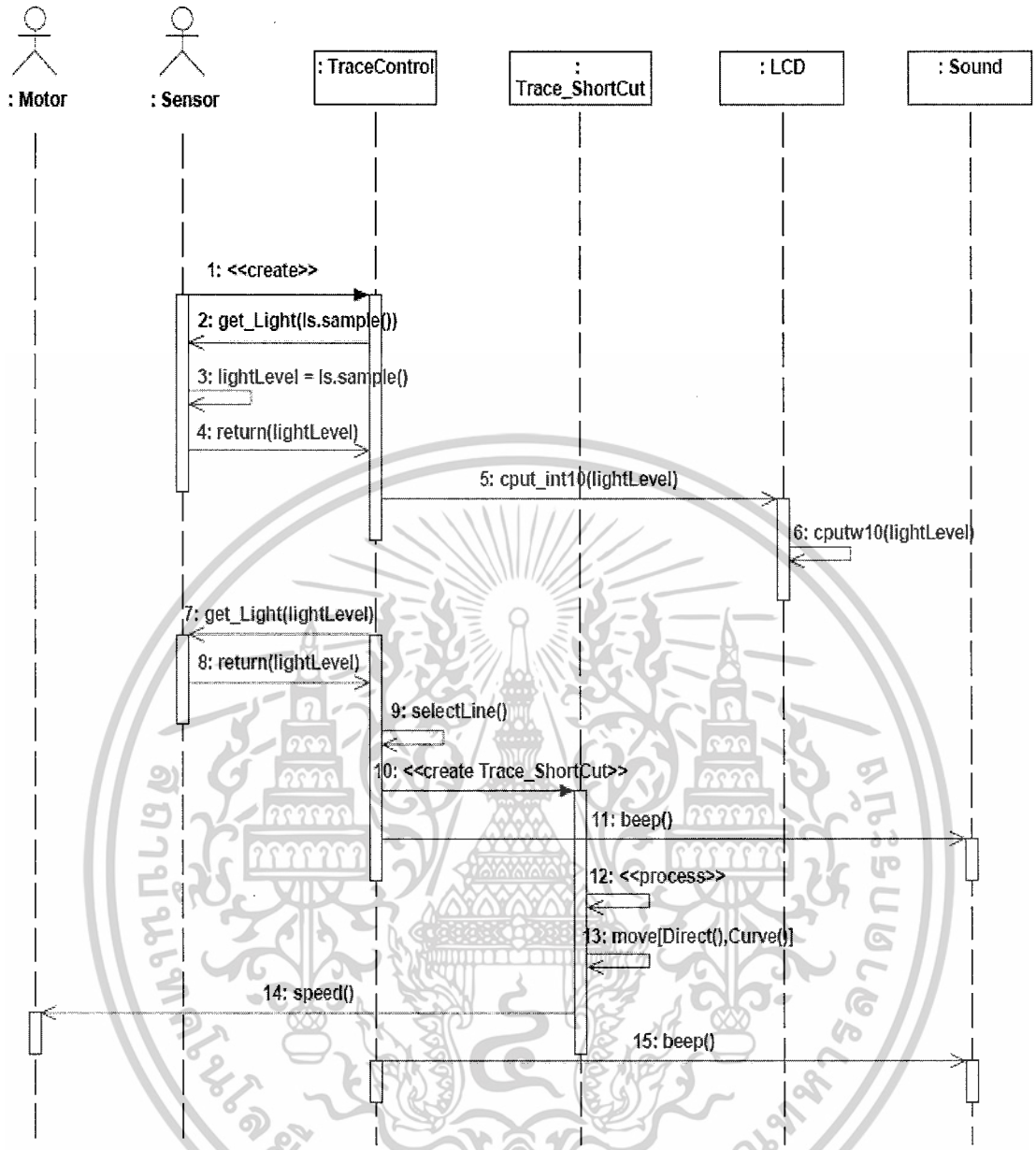
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงแผนภาพ Sequence Diagram ของ TraceLine

จากแผนภาพซีเควนไดอะแกรมของการติดตามเส้นทาง สามารถอธิบายได้ดังนี้ คือระบบจะทำการอ่านค่าข้อมูลจากสภาพแวดล้อมเข้ามาเก็บเพื่อใช้ในการประมวลก่อนเริ่มทำการรันโปรแกรมให้หุ่นยนต์เคลื่อนที่ โดยทุกครั้งที่รับค่าจะแสดงค่าออกทางหน้าจอ จากนั้นเมื่อระบบเริ่มทำการเคลื่อนที่จะมีการรับข้อมูลจากสภาพแวดล้อมเข้าไปประมวลผลตลอดเวลาที่มีการเคลื่อนที่ ซึ่งหากเป็นการเคลื่อนที่ในเส้นทางปกติจะมีการตรวจสอบค่าข้อมูลที่รับเข้ามาและตัดสินใจเลือกเส้นทางได้ โดยการทำงานในลักษณะของการเคลื่อนที่ในเส้นทางปกติจะทำการเคลื่อนที่ไปในเส้นทางตรง ทางโค้ง และทางชันได้ นอกจากนี้หากมีการเคลื่อนที่หลุดนอกเส้นทางที่กำหนด ระบบจะต้องทำการค้นหาเพื่อกลับสู่เส้นทางปกติ โดยที่ทุกการทำงานจะต้องมีการสั่งงานมอเตอร์เพื่อกำหนดความเร็วและทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

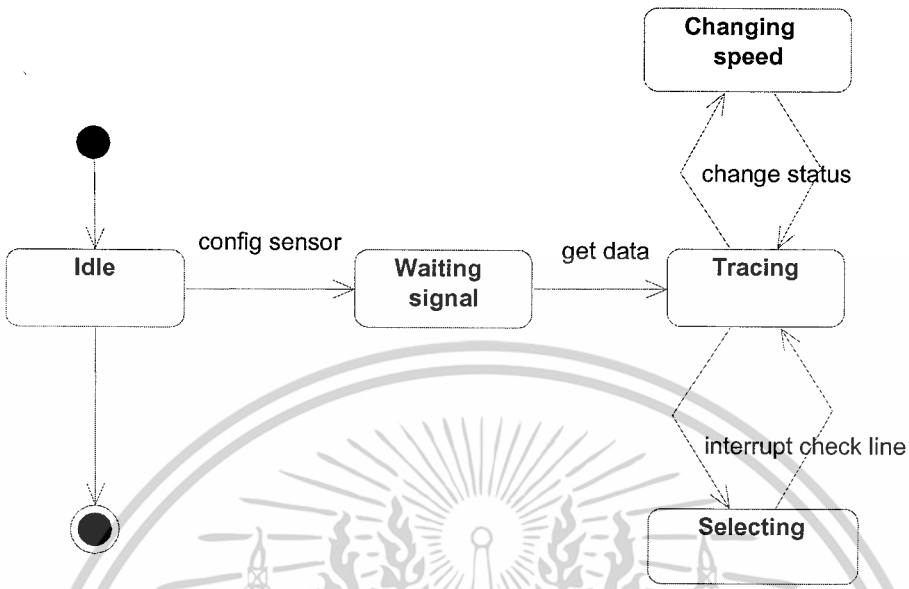


รูปที่ 3.17 แสดงแผนภาพ Sequence Diagram ของ TraceShortCut

จากแผนภาพซีเควนไดอะแกรมของการติดตามเส้นทางลัดสามารถอธิบายได้เช่นเดียวกับแผนภาพซีเควนไดอะแกรมของการติดตามเส้นทาง แต่หากมีการตัดสินใจเลือกเส้นทางระบบจะแสงเสียงสัญญาณว่ามีเงื่อนไขเพื่อให้ระบบทำการตัดสินใจเลือกทิศทางในการเคลื่อนที่ ซึ่งหากระบบทำการเคลื่อนไปในเส้นทางลัด ระบบจะแสดงเสียงสัญญาณในขณะที่อยู่ในเส้นทางลัดจนกว่าจะมีการออกจากเส้นทางลัดเพื่อเข้าสู่เส้นทางปกติ โดยจะสั่งงานมอเตอร์ให้เคลื่อนที่ด้วยความเร็วและทิศทางที่เหมาะสมกับลักษณะของเส้นทางลัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.6 สเตทชาร์ตไดอะแกรม



รูปที่ 3.18 แสดง State chart Diagram

จากสเตทชาร์ตไดอะแกรมสามารถอธิบายการทำงานได้ดังนี้ เมื่อเริ่มต้นที่ Initial State โดยมีการกดปุ่ม RUN ที่อาร์ซีเอ็กซ์ หุ่นยนต์จะเข้าสู่สภาวะหยุดนิ่ง (Idle) จนเมื่อเซนเซอร์เริ่มการทำงาน ระบบจะเข้าสู่สถานะ WaitingSignal เพื่อรอค่าข้อมูลจากเซนเซอร์แสง

เมื่อเซนเซอร์แสงทำการอ่านข้อมูลเข้ามาโดยมีเหตุการณ์ที่มากระตุ้นการเปลี่ยนสถานะคือ เซนเซอร์มีการอ่านข้อมูล (get data) เมื่อมีข้อมูลเข้าระบบจะทำการเปลี่ยนสถานะเป็นสถานะของการติดตามเส้นทาง โดยสถานะของการติดตามเส้นทางจะมีการเปลี่ยนแปลงสถานะไปสู่สถานะอื่นคือ

เมื่อมีเหตุการณ์หรือการรับค่าข้อมูลที่ระบุว่าเป็นเส้นทางที่ต่างไปจากลักษณะเส้นทางปกติ คือ กรณีที่เป็นเส้นทางโค้ง ทางชัน จะมีการเปลี่ยนสถานะไปสู่สถานะของการเปลี่ยนแปลงความเร็ว จนกว่าจะมีการเคลื่อนที่กลับสู่เส้นทางปกติซึ่งจะเปลี่ยนสถานะกลับสู่สถานะเดิม

เมื่อมีเหตุการณ์หรือการรับค่าข้อมูลที่ต้องตัดสินใจในการเลือกเส้นทาง จะเปลี่ยนแปลงสถานะไปยังสถานะที่ต้องตัดสินใจเพื่อเตรียมปรับรูปแบบของการเคลื่อนที่ไปตามลักษณะเส้นทางที่ต้องการได้

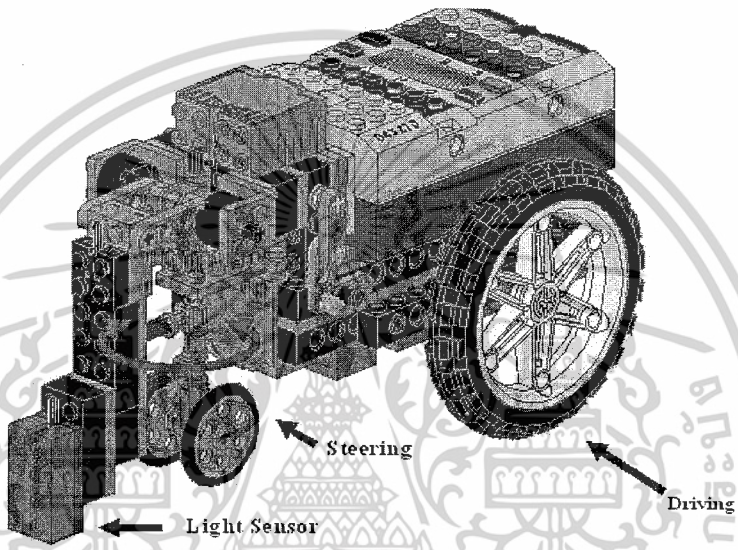
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบฮาร์ดแวร์

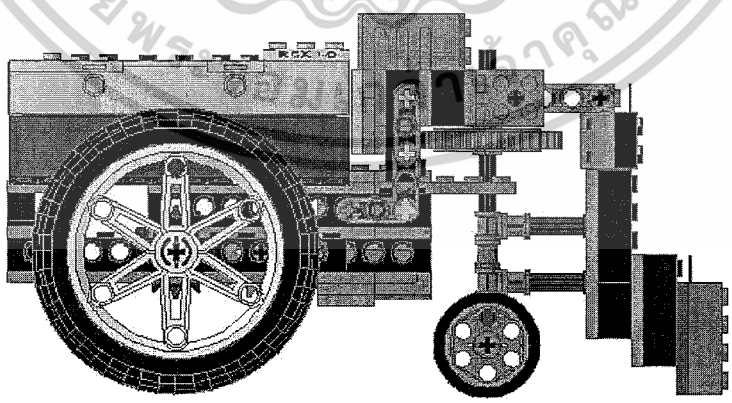
สำหรับการออกแบบและพัฒนาหุ่นยนต์ติดตามเส้นทาง (Line trace robot) จะใช้อุปกรณ์ทดลองชุดเลโก้มายด์สตอม(LEGO Mindstorms) รุ่นอาร์ซีเอ็ก (RCX) ซึ่งเป็นฮาร์ดแวร์ระบบฝังตัวในการทดลอง

3.2.3.1 แอปพลิเคชั่นระบบหุ่นยนต์ติดตามเส้นทาง

หุ่นยนต์ติดตามเส้นทางจะมีลักษณะดังแสดงในรูปที่ 3.19 และรูปที่ 3.20



รูปที่ 3.19 แสดงส่วนประกอบของหุ่นยนต์ติดตามเส้นทาง



รูปที่ 3.20 แสดงส่วนประกอบทางด้านข้างของหุ่นยนต์ติดตามเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

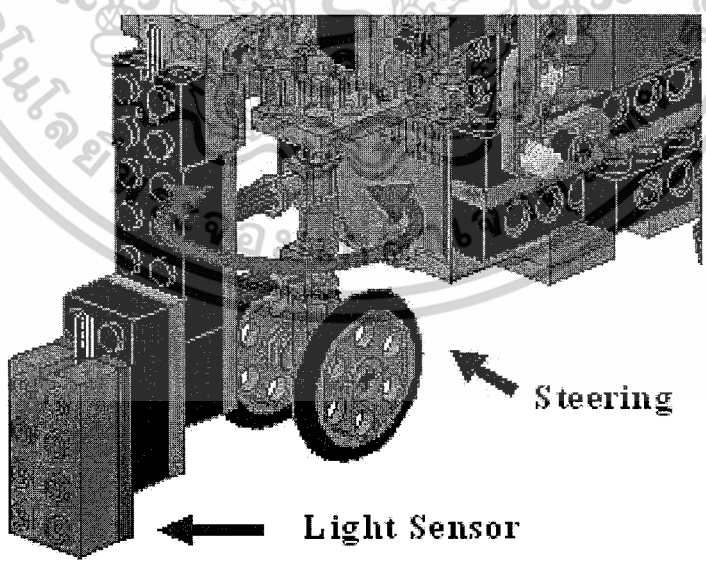
3.2.3.2 แนวคิดของการออกแบบฮาร์ดแวร์

เนื่องจากการทำงานของหุ่นยนต์ติดตามเส้นทางจะมีลักษณะของการทำงาน โดยจะมีการอ่านค่าข้อมูลเข้ามาจากสภาพแวดล้อมเข้ามาประมวลผล เพื่อให้สามารถติดตามเส้นทางปกติซึ่งเป็นเส้นทึบสีดำ และเส้นทาง short cut ที่มีลักษณะเป็นเส้นทึบสีขาวและดำสลับกัน โดยจะต้องสามารถเคลื่อนที่ผ่านลักษณะเส้นทางที่เป็นเส้นทางตรง ทางโค้ง และทางชันได้ จากลักษณะการทำงานของหุ่นยนต์ดังกล่าวสามารถนำมาออกแบบและสร้างแอพพลิเคชั่นที่ใช้ในการทำงานได้ดังนี้

1. แสดงการออกแบบส่วนของการรับค่าข้อมูล

ตารางที่ 3.18 แสดงการออกแบบส่วนของการรับค่าข้อมูล

ความต้องการของระบบ	การออกแบบหุ่นยนต์
การรับค่าข้อมูลเข้ามาเพื่อประมวลผล ระบบจะทำการรับค่าข้อมูลผ่านทางอุปกรณ์เซนเซอร์ ซึ่งจะใช้เซนเซอร์แสง(Light Sensor)จำนวน 1 ตัว ทำหน้าที่เปรียบเสมือนสายตาคนในการรับค่าวัตถุหรือแสงเข้ามา	สำหรับการออกแบบหุ่นยนต์ในส่วนของการรับค่าแสง จะใช้เซนเซอร์แสง(Light Sensor) ซึ่งจะถูกติดอยู่ด้านหน้าสุดของหุ่นยนต์เพื่อเป็นตัวตรวจจับว่าหุ่นยนต์จะต้องเคลื่อนที่ไปในทิศทางใด โดยจะต้องติดเซนเซอร์ให้อยู่ต่ำให้ใกล้กับค่าที่รับเข้ามามากที่สุดเพื่อลดปัญหาเรื่องการสะท้อนของแสง



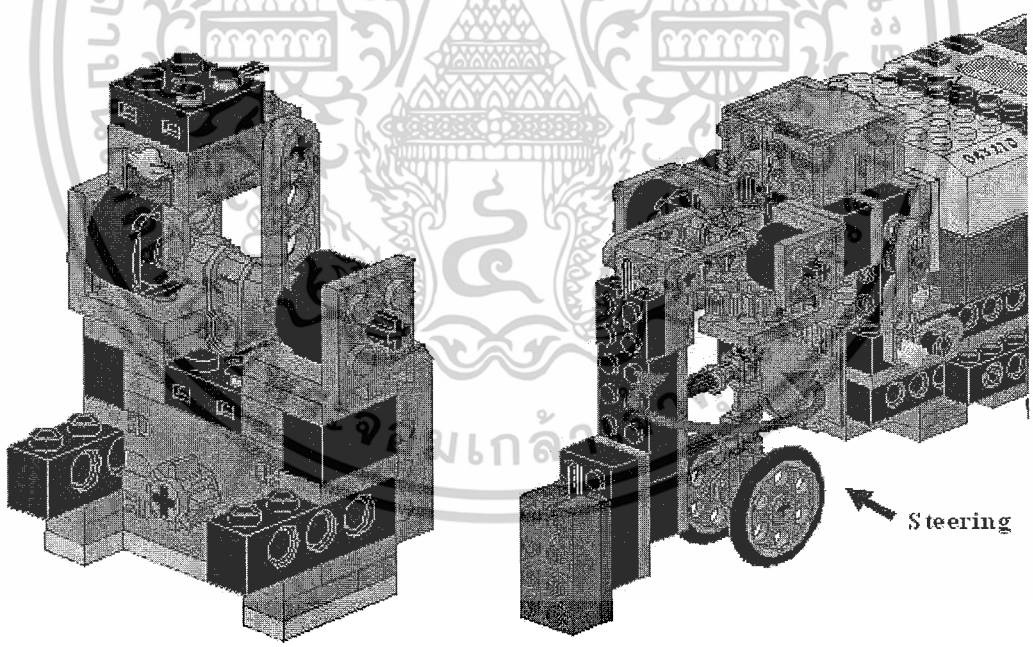
รูป 3.21 แสดงการติดตั้งส่วนของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของสเตียริงมอเตอร์ (Steering motor)

ตารางที่ 3.19 แสดงการออกแบบส่วนของสเตียริงมอเตอร์

ความต้องการของระบบ	การออกแบบหุ่นยนต์
ระบบสามารถเคลื่อนที่ไปตามลักษณะของเส้นทางแต่ละเส้นทางได้ ส่วนของสเตียริงมอเตอร์จะทำหน้าที่ในการควบคุมทิศทางของการเคลื่อนที่ (คล้ายการบังคับพวงมาลัยในรถยนต์) เพื่อทำการติดตามเส้นทางดังกล่าวได้	ส่วนของสเตียริงจะประกอบด้วยมอเตอร์ 1 ตัวที่ใช้ในการหมุนเพื่อควบคุมทิศทาง ซึ่งจะถูกเชื่อมต่อกับเซนเซอร์แสงเพื่อทำงานร่วมกันในการควบคุมและค้นหาเส้นทาง โดยสเตียริงจะทำหน้าที่ในการหมุนล้อด้านหน้าในลักษณะซ้าย-ขวา และมีเซนเซอร์แสงอ่านค่าจากสภาพแวดล้อมเข้าไปให้หน่วยประมวลผลใช้ประมวลผล



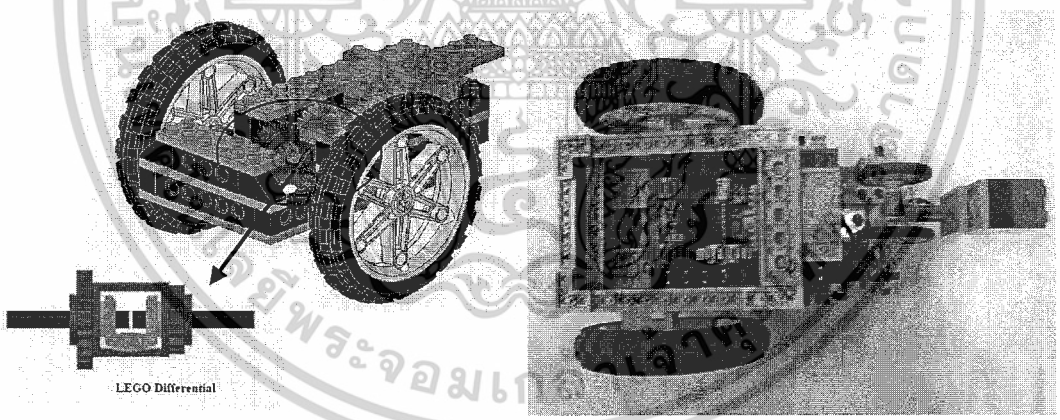
รูป 3.22 แสดงส่วนของสเตียริงมอเตอร์ (Steering motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนของไดร์วิงมอเตอร์ (Driving Motor)

ตารางที่ 3.20 แสดงการออกแบบส่วนของไดร์วิงมอเตอร์

ความต้องการของระบบ	การออกแบบหุ่นยนต์
ระบบจะต้องทำติดตามเส้นทางและเคลื่อนที่ไปด้วยความเร็วที่เหมาะสม รวมถึงสามารถปรับความเร็วตามสภาพแวดล้อมที่ต่างกันได้	ส่วนของไดร์วิงมอเตอร์ ทำหน้าที่ในการขับเคลื่อนหุ่นยนต์ ประกอบด้วยมอเตอร์ 1 ตัวที่ถูกประกอบเข้ากับเพลาและเพื่อใช้เชื่อมต่อเข้ากับล้อด้านหลัง 2 ตัว สำหรับใช้ในการขับเคลื่อนล้อทั้งสองข้างในลักษณะเดินหน้าและถอยหลังและใช้ limited slip diffrential ช่วยในการควบคุมล้อทั้งสองข้างที่ถูกสั่งงานผ่านมอเตอร์นี้ โดยหุ่นยนต์จะต้องมีการควบคุมความเร็วในการเคลื่อนที่ได้อย่างเหมาะสม



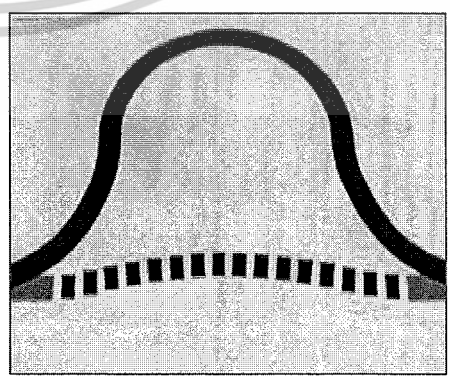
รูป 3.23 ส่วนประกอบในส่วนของไดร์วิงมอเตอร์ (Driving Motor)

3.2.4 การออกแบบสภาพแวดล้อมที่ใช้ในการทดลอง

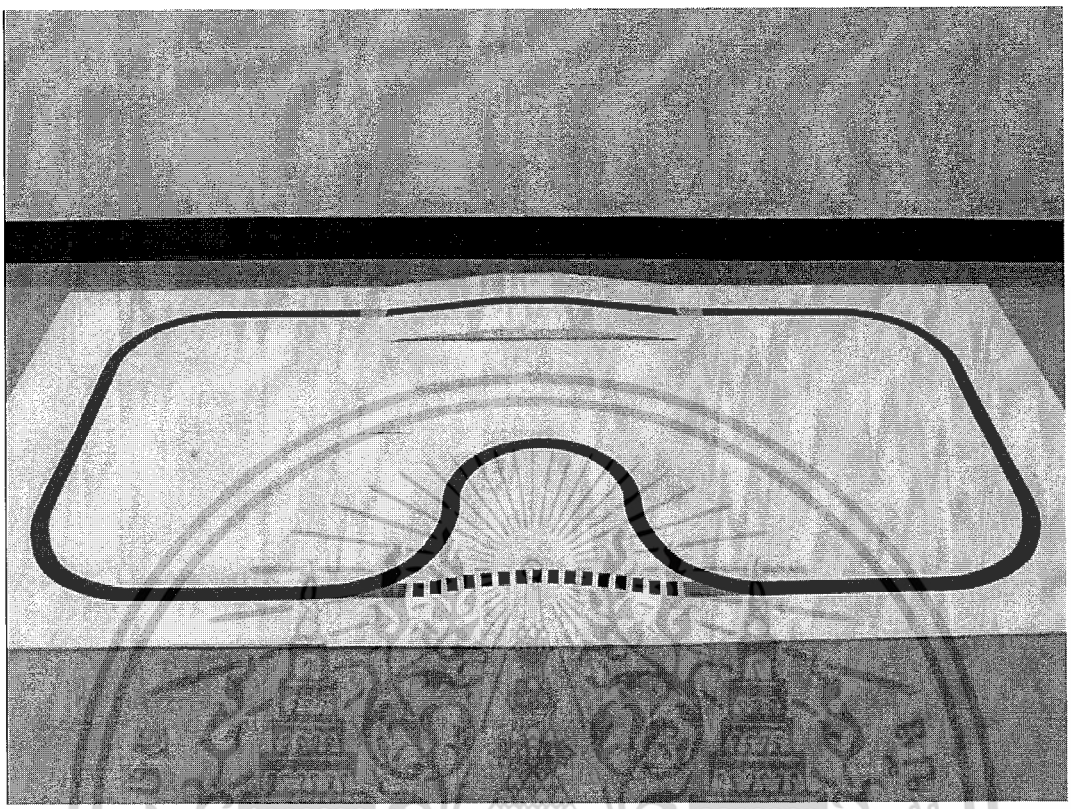
จากความต้องการของระบบซึ่งระบบจะต้องสามารถเคลื่อนที่ไปในเส้นทางลักษณะที่แตกต่างกันและสามารถตัดสินใจเพื่อเลือกเส้นทาง ได้แก่ สภาพสนามที่ประกอบด้วยเส้นทางตรง เส้นทางโค้ง เส้นทางชัน และเส้นทางลัด สามารถนำมาสร้างเป็นสภาพแวดล้อมที่จะใช้ในการทดลองได้ดังแสดงในตารางที่ 3.13 และรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.21 สภาพแวดล้อมที่ใช้ในการทดลอง

สภาพแวดล้อม	รูปแสดงสภาพแวดล้อม
เส้นทางตรง	
เส้นทางโค้ง	
เส้นทางชัน	
เส้นทางลัด (Shortcut)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.24 สภาพแวดล้อมที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองผลและผลการทดลอง

จากการวิเคราะห์และออกแบบระบบในบทที่ 3 ทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์ ในบทนี้จะเป็นการทดลองและผลการทดลองตามที่ได้ออกแบบไว้ ซึ่งแยกย่อยการทดลองและสรุปผลการทดลองแต่ละการทดลองได้ดังนี้

4.1 การทดลองการทำงานของเซนเซอร์แสง

ในการประมวลผลการทำงานของหุ่นยนต์จะมีการรับค่าข้อมูลแสงผ่านทางเซนเซอร์แสง ซึ่งในการทำงานจะใช้เซนเซอร์แสงจำนวน 1 ตัว ดังนั้นการประมวลผลการวัดค่าแสงจึงมีส่วนสำคัญต่อการทำงานของระบบในการตรวจสอบค่าสีของเส้นทางที่ต้องเคลื่อนที่ไปตามที่กำหนด การทดลองนี้จะแสดงให้เห็นผลของการรับค่าสีจากค่าสีที่แตกต่างกัน จากสภาพแวดล้อมที่ต่างกัน จากนั้นจะทำการเปรียบเทียบค่าสีที่ได้ เพื่อใช้ในการทดสอบโปรแกรมต่อไป

วัตถุประสงค์ เพื่อทดสอบการทำงานของเซนเซอร์แสงที่ระดับความเข้มแสงและสภาพแวดล้อมที่แตกต่างกัน

สมมติฐาน เซนเซอร์จะวัดค่าแสงได้ถูกต้องแม่นยำเมื่อวัตถุที่ใช้ทดลองและสภาพแวดล้อมที่ใช้ทดลองมีสภาวะการสะท้อนแสงปกติ

ตารางที่ 4.1 แสดงวิธีการทดลองการทำงานของเซนเซอร์แสง

วิธีการทดลอง
1. ทดลองโดยใช้คำสั่ง get() 2. ทดลองโดยใช้คำสั่ง sample() 3. ทดลองโดยใช้คำสั่ง sample(x,y) หมายเหตุ - ค่าค่าพารามิเตอร์ x กับ y ให้ทดลองเปลี่ยนค่าพารามิเตอร์ x กับ y หลายๆค่าที่แตกต่างกัน - ทดลองโดยใช้กระดาษสีปกติ กระดาษสติ๊กเกอร์สีและเทปกาวสี โดยจะทดสอบกับระดับแสงที่ต่างกันสามระดับดังในตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

จากการทดลองให้ผลการทดลองจะพบว่าการรับค่าโดยตรงด้วยคำสั่ง get() หรือรับค่าเฉลี่ยด้วยคำสั่ง sample(x,y) โดยที่ทดสอบการเปลี่ยนค่าพารามิเตอร์ x ไม่มีผลต่อระดับค่าความเข้มสีที่ได้แต่ชนิดของกระดาษมีผลต่อระดับค่าความเข้มสีดังนี้

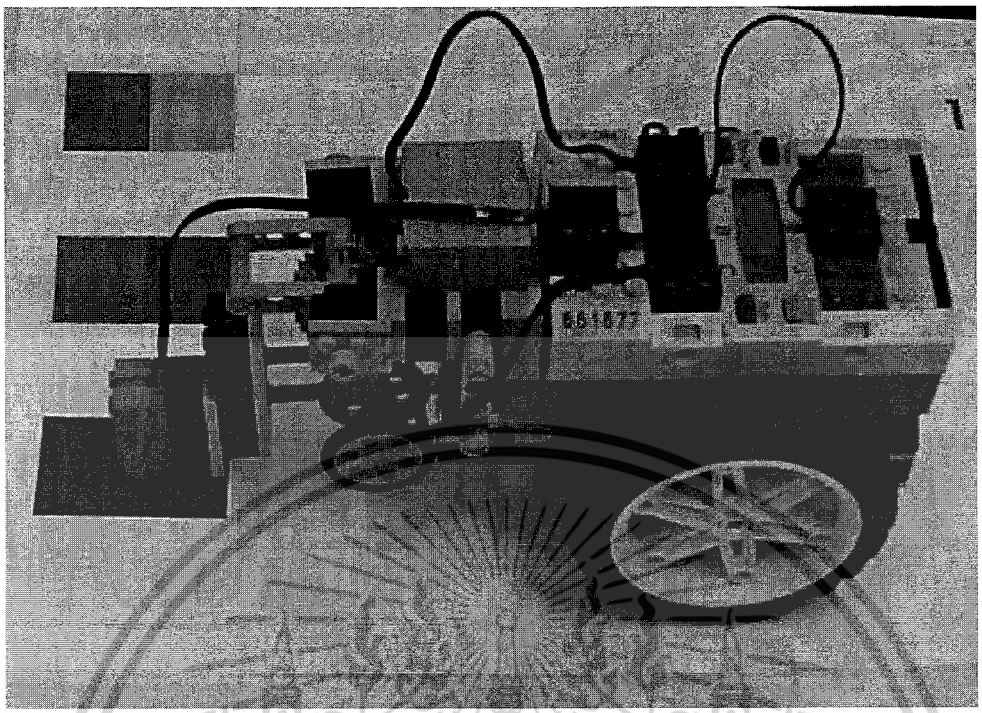
ตารางที่ 4.2 แสดงค่าสีที่ได้จากการทดลองการรับค่าแสงจากเซนเซอร์แสง

ชนิดของกระดาษ			ระดับค่าความเข้มของสีที่ระดับความสว่างที่แตกต่างกัน		
กระดาษปกติ	สติ๊กเกอร์สี	เทปกาวสี	สว่างน้อย	สว่างปานกลาง	สว่างมาก
		แดง	0x30	0x30	0x30
		ม่วง	0x2b	0x2b	0x2b
		เขียว	0x22	0x2b	0x22
		เหลือง	0x30-0x31	0x31	0x30-0x31
		ดำ	0x23	0x23	0x24
		น้ำเงิน	0x29	0x2a-0x2b	0x28-0x29
ดำ			0x24	0x22	0x25
เขียว			0x2b	0x2b	0x2b
น้ำเงิน			0x28-0x29	0x29	0x2a
ส้ม			0x31	0x31	0x30-0x31
ชมพู			0x2b	0x31	0x31
	ดำ		0x25	0x24	0x23

ตารางที่ 4.3 แสดงค่าสีเมื่อทำการแปลงเป็นเลขฐานสิบ

สีที่ใช้	ค่าจริงที่เก็บได้(Hex)	ค่าเลขฐานสิบ(%)
แดง	0x30	49
ม่วง	0x2b	43-44
เขียว	0x22	44
เหลือง	0x30-0x31	49
ดำ	0x24	00
น้ำเงิน	0x28-0x29	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงการทดลองการรับค่าความเข้มสีต่างๆจากเซนเซอร์แสง

จากค่าที่ได้จะพบว่าชนิดของกระดาษและความเข้มของระดับแสงมีผลต่อการรับค่าของเซนเซอร์ทำให้การแยกสีของเซนเซอร์อาจมีความผิดพลาดได้

4.2 การทดลองการเคลื่อนที่ของหุ่นยนต์

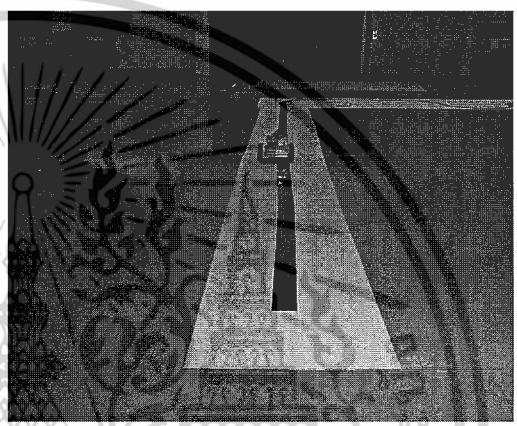
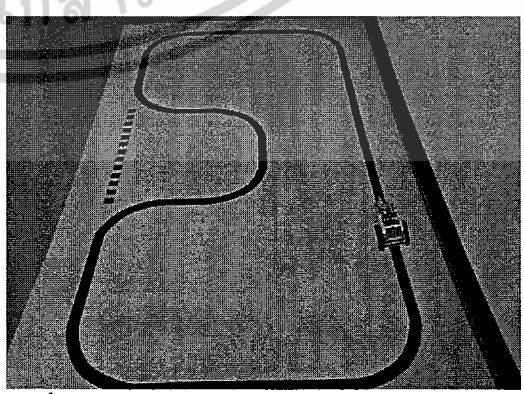
4.2.1 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางตรง

วัตถุประสงค์ เพื่อทดสอบการเคลื่อนที่ของหุ่นยนต์ ในลักษณะของเส้นทางที่เป็นแบบเส้นทางตรง

สมมติฐาน หุ่นยนต์สามารถเคลื่อนที่ตามเส้นตรงได้ถูกต้องและรวดเร็วโดยใช้ความเร็วในการเคลื่อนที่ของหุ่นยนต์ที่เหมาะสม และเซนเซอร์แสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางตรง

วิธีการทดลอง	ผลการทดลอง
1. ทดลองบนเส้นตรงที่มีระยะทาง 3 เมตร 1.1 ทดลองกำหนดระดับความเร็วของไดร์วิงมอเตอร์ระดับต่างๆ 1.2 กำหนดระดับความเร็วของสเตียริงมอเตอร์ระดับต่างๆ 1.3 สังเกตลักษณะและเวลาที่หุ่นยนต์เคลื่อนที่	1. ผลการทดลองบนเส้นตรงที่มีระยะทาง 3 เมตร หุ่นยนต์สามารถเดินตามเส้นได้ถูกต้องและรวดเร็วเมื่อใช้ระดับความเร็วของไดร์วิงมอเตอร์ที่ระดับ 85 รอบ/นาที และสเตียริงมอเตอร์ที่ระดับ 20 รอบ/นาที  **รูปที่ 4.2** การทดลองการเคลื่อนที่ของหุ่นยนต์บนเส้นทางตรงระยะทาง 3 เมตร
2. ทดลองบนเส้นตรงที่มีระยะทาง 0.75 เมตร ทดลองลักษณะเดียวกับการเคลื่อนที่บนเส้นตรงที่มีระยะทาง 3 เมตร <u>หมายเหตุ</u> ต้องมีการควบคุมแสงในสภาพแวดล้อมที่ใช้ ทดลองในระดับเดียวกัน	2. ผลการทดลองบนเส้นตรงที่มีระยะทาง 0.75 เมตร หุ่นยนต์สามารถเดินตามเส้นได้ถูกต้องและรวดเร็วเมื่อใช้ระดับความเร็วของไดร์วิงมอเตอร์ที่ระดับ 90 รอบ/นาที และสเตียริงมอเตอร์ที่ระดับ 20 รอบ/นาที  **รูปที่ 4.3** การทดลองการเคลื่อนที่ของหุ่นยนต์บน เส้นทางตรงระยะทาง 0.75 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ผลการทดลอง

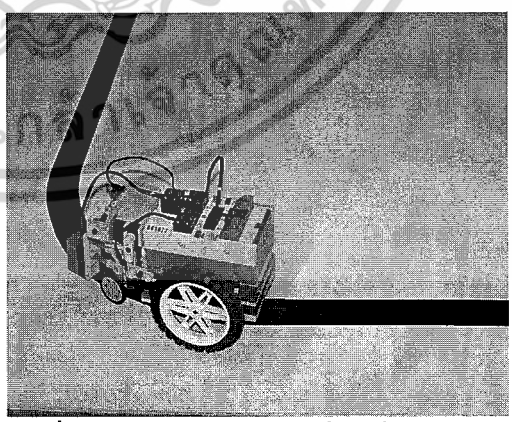
ระดับความเร็วที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางตรงในระยะทางที่แตกต่างกันจะใช้ระดับความเร็วที่เหมาะสมในการเคลื่อนที่แตกต่างกันเพราะความเร็วในการเคลื่อนที่ของหุ่นยนต์มีผลต่อความเร็วในการรับค่าจากสภาพแวดล้อมไปประมวลผล

4.2.2 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางโค้ง

วัตถุประสงค์ เพื่อทดสอบการเคลื่อนที่ของหุ่นยนต์ ในลักษณะของเส้นทางที่เป็นแบบเส้นทางโค้ง

สมมติฐาน หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางโค้งได้ถูกต้องและรวดเร็วโดยใช้ความเร็วในการเคลื่อนที่ของหุ่นยนต์ที่เหมาะสม และเซนเซอร์แสงสามารถแยกแยะสีของพื้นสนามและสีของเส้นสนามได้ถูกต้อง

ตารางที่ 4.5 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางโค้ง

วิธีการทดลอง	ผลการทดลอง
1. ทดลองบนเส้นโค้ง 90 องศา 1.1 ทดลองกำหนดระดับความเร็วของไดร์วิงมอเตอร์ ระดับต่างๆ 1.2 กำหนดระดับความเร็วของสเตียริงมอเตอร์ ระดับต่างๆ 1.3 สังเกตลักษณะและเวลาที่หุ่นยนต์เคลื่อนที่	1. ผลทดลองบนเส้นโค้ง 90 องศา หุ่นยนต์สามารถเดินตามเส้นได้ถูกต้องและรวดเร็วเมื่อใช้ระดับความเร็วของไดร์วิงมอเตอร์ที่ระดับ 85 รอบ/นาที และสเตียริงมอเตอร์ที่ระดับ 20 รอบ/นาที  **รูปที่ 4.4** การทดลองการเคลื่อนที่บนเส้นโค้ง90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 (ต่อ) แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางโค้ง

วิธีการทดลอง	ผลการทดลอง
2. ทดลองบนเส้นโค้งแบบต่อเนื่อง ทดลองลักษณะเดียวกับการเคลื่อนที่บนเส้นโค้ง 90 องศา	2. ผลทดลองบนเส้นโค้งแบบต่อเนื่อง หุ่นยนต์สามารถเดินตามเส้นได้ถูกต้องและรวดเร็วเมื่อใช้ระดับความเร็วของไดร์วิงมอเตอร์ที่ระดับ 90 รอบ/นาที และสเตียริงมอเตอร์ที่ระดับ20 รอบ/นาที **รูปที่ 4.5** การทดลองการเคลื่อนที่บนเส้นโค้ง แบบต่อเนื่อง

วิเคราะห์ผลการทดลอง

1. การเคลื่อนที่ของหุ่นยนต์บนเส้นทางโค้ง 90 องศา มีผลทำให้การเคลื่อนที่ของหุ่นยนต์ช้ากว่าการเคลื่อนที่ในเส้นตรงและทางโค้งปกติ

2. การเคลื่อนที่ของหุ่นยนต์บนเส้นทางโค้งปกติมีผลทำให้การเคลื่อนที่ของหุ่นยนต์ช้ากว่าการเคลื่อนที่ในเส้นตรงแต่เร็วกว่าทางโค้ง 90 องศา

เนื่องจากการเคลื่อนที่ของหุ่นยนต์บนเส้นทางโค้ง 90 องศาและทางโค้งปกติ สเตียริงมอเตอร์มีการหมุนในมุมที่กว้างขึ้นจึงมีผลให้การเคลื่อนที่ของหุ่นยนต์ช้าลง

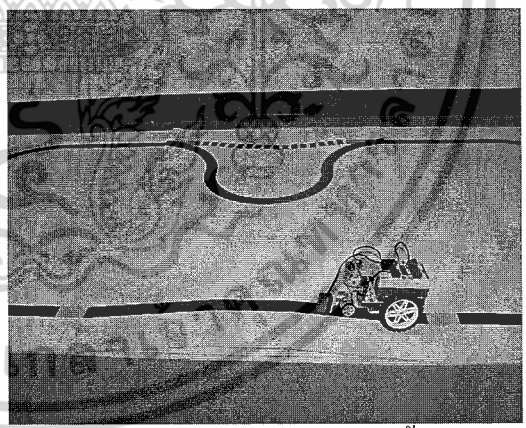
4.2.3 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางชัน

วัตถุประสงค์ เพื่อทดสอบการเคลื่อนที่ของหุ่นยนต์ ในลักษณะของเส้นทางที่เป็นแบบเป็นเส้นทางชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติฐาน หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางชันโดยไม่เสียการควบคุมด้วยความเร็วที่เหมาะสม

ตารางที่ 4.6 แสดงการทดลองและผลการทหลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางชัน

วิธีการทดลอง	ผลการทดลอง
ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางชัน 1. ทดลองกำหนดระดับความเร็วของไดร์วิงมอเตอร์ ระดับต่างๆ 2. ทดลองกำหนดระดับความเร็วของสเตียริงมอเตอร์ ระดับต่างๆ 3. สังเกตลักษณะและเวลาที่หุ่นยนต์เคลื่อนที่	1. หุ่นยนต์สามารถเคลื่อนที่ขึ้นตามเส้นทางชันได้ถูกต้องรวดเร็วและไม่เสียการควบคุมที่ระดับความเร็วของไดร์วิงมอเตอร์ที่ 150 รอบ/นาที และระดับความเร็วของสเตียริงมอเตอร์ที่ 20 รอบ/นาที 2. หุ่นยนต์สามารถเคลื่อนที่ลงตามเส้นทางชันได้ถูกต้องและไม่เสียการควบคุมที่ระดับความเร็วของไดร์วิงมอเตอร์ที่ 40 รอบ/นาที และระดับความเร็วของสเตียริงมอเตอร์ที่ 20 รอบ/นาที 

รูปที่ 4.6 การทดลองการเคลื่อนที่ขึ้นทางชัน

วิเคราะห์ผลการทดลอง

การกำหนดความเร็วของไดร์วิงมอเตอร์ในการเคลื่อนที่ในลักษณะของพื้นสนามที่สภาพแวดล้อมที่แตกต่างกัน มีผลโดยตรงต่อการเคลื่อนที่หุ่นยนต์ซึ่งทำให้หุ่นยนต์สามารถทำงานได้ตามการควบคุมที่กำหนดไว้

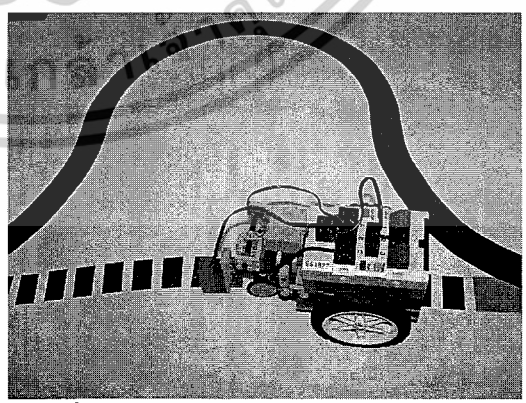
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางลัด

วัตถุประสงค์ เพื่อทดลองการตัดสินใจและเลือกเส้นทางในการเคลื่อนที่ของหุ่นยนต์ในลักษณะของเส้นทางที่มีลักษณะเป็นเส้นทางลัดได้ด้วยความเร็วที่เหมาะสม

สมมติฐาน หุ่นยนต์สามารถตัดสินใจและเลือกเส้นทางในการเคลื่อนที่ตามเส้นทางลัดได้โดยไม่เสียการควบคุมด้วยความเร็วที่เหมาะสม

ตารางที่ 4.7 แสดงการทดลองและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางลัด

วิธีการทดลอง	ผลการทดลอง
1. ทดลองการตัดสินใจเลือกเส้นทางในการเคลื่อนที่ตามเส้นทางลัดของหุ่นยนต์ด้วยการกำหนดให้หุ่นยนต์เลือกเส้นทางลัดเมื่อหุ่นยนต์ตรวจจับสภาพสนามที่มีระดับความเข้มของค่าสีที่สูงกว่าระดับความเข้มของค่าสีที่เป็นสีขาว 2. ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเป็นเส้นทางลัดด้วยระดับความเร็วที่ใช้ในการเคลื่อนที่ที่แตกต่างกัน 2.1 ทดลองกำหนดระดับความเร็วของไดร์วิงมอเตอร์ ระดับต่างๆ 2.2 ทดลองกำหนดระดับความเร็วของสเตียริงมอเตอร์ ระดับต่างๆ	1. ผลการทดลองการตัดสินใจเลือกเส้นทางในการเคลื่อนที่ตามเส้นทางลัดของหุ่นยนต์ พบว่าหุ่นยนต์สามารถตัดสินใจเลือกเส้นทางได้ถูกต้องเมื่อหุ่นยนต์ตรวจพบเส้นสีขาวสะท้อนแสง 2. ผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามที่มีลักษณะเส้นทางลัด หุ่นยนต์สามารถเคลื่อนที่ได้ถูกต้องและรวดเร็วเมื่อหุ่นยนต์เคลื่อนที่ด้วยระดับความเร็วของไดร์วิงมอเตอร์ 85 รอบ/นาที และสเตียริงมอเตอร์ 20 รอบ/นาที  **รูปที่ 4.7** การทดลองการเคลื่อนที่บนเส้นทางลัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ผลการทดลอง

ระดับความเข้มของค่าสีที่เซนเซอร์รับมาประมวลเพื่อสั่งงานมอเตอร์และระดับความเร็วที่เคลื่อนที่ของหุ่นยนต์มีผลต่อการตัดสินใจเลือกเส้นทางลัดที่ถูกต้องของหุ่นยนต์

4.2.5 ทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม

วัตถุประสงค์ เพื่อทดสอบการเคลื่อนที่ของหุ่นยนต์ในทุกสภาพสนามที่ต่อเนื่องกัน ได้แก่ สภาพสนามที่เป็นเส้นทางตรง ทางโค้ง ทางลัดและทางลาดชันได้โดยไม่เสียการควบคุมด้วยความเร็วที่เหมาะสม

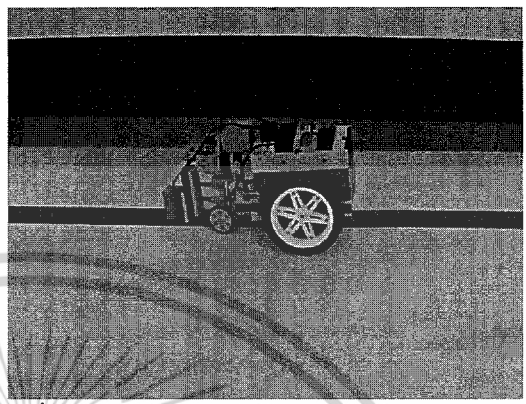
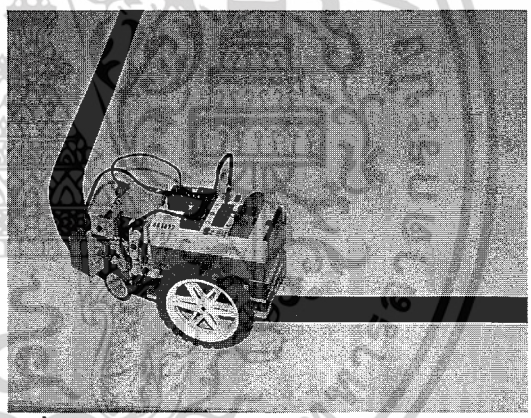
สมมติฐาน หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางในทุกสภาพสนามได้ตามที่กำหนดโดยไม่เสียการควบคุมเมื่อใช้ความเร็วในการเคลื่อนที่ที่เหมาะสม

ตารางที่ 4.8 แสดงการทดลองการและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม

วิธีการทดลอง	ผลการทดลอง
1. ทดลองการเคลื่อนที่แบบต่อเนื่องของหุ่นยนต์ในทุกสภาพสนามได้แก่ เส้นทางตรง ทางโค้ง ทางลัดและเส้นทางชันแบบต่อเนื่อง โดยทำการควบคุมความเร็วที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ในแต่ละเงื่อนไขที่ระดับความเร็วระดับเดียวตลอดเส้นทาง 2. ทดลองการเคลื่อนที่แบบต่อเนื่องของหุ่นยนต์ในทุกสภาพสนามได้แก่ เส้นทางตรง ทางโค้ง ทางลัดและเส้นทางชันแบบต่อเนื่อง โดยทำการควบคุมระดับความเร็วที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ในแต่ละเงื่อนไขที่ระดับความเร็วแตกต่างกันตามสภาพสนาม	หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางได้รวดเร็วและไม่เสียการควบคุมเมื่อหุ่นยนต์ 1. ใช้ความเร็ว 90 รอบ/นาที ในการเคลื่อนที่ในเส้นทางตรง 2. ใช้ความเร็ว 85 รอบ/นาที ในการเคลื่อนที่ในเส้นทางโค้ง 3. ใช้ความเร็ว 85 รอบ/นาที ในการเคลื่อนที่ในเส้นทางลัด 4. ใช้ความเร็ว 150 รอบ/นาที ในการเคลื่อนที่ขึ้นทางลาดชัน และใช้ความเร็ว 40 รอบ/นาที ในการเคลื่อนที่ลงทางลาดชัน <u>หมายเหตุ</u> ความเร็วในการเคลื่อนที่ของหุ่นยนต์จะต้องปรับเปลี่ยนไปตามเงื่อนไขของสภาพสนามหุ่นยนต์จึงจะสามารถเคลื่อนที่ได้ต่อเนื่องในทุกสภาพสนาม

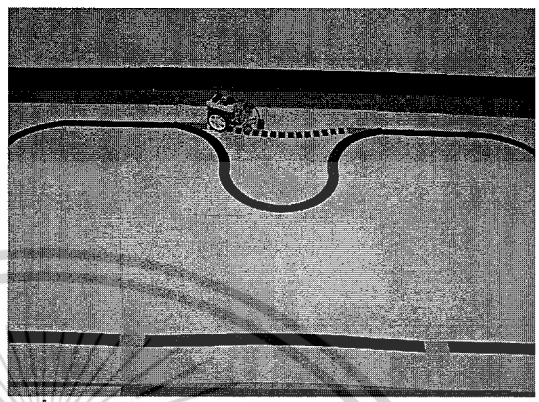
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 (ต่อ) แสดงการทดลองการและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.8** ทดลองการเคลื่อนที่บนเส้นทางตรงในสภาพสนามรวม  **รูปที่ 4.9** ทดลองการเคลื่อนที่บนเส้นทางโค้งในสภาพสนามรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 (ต่อ) แสดงการทดลองการและผลการทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพสนามรวม

วิธีการทดลอง	ผลการทดลอง
	 **รูปที่ 4.10** ทดลองการเคลื่อนที่บนเส้นทางลัดในสภาพสนามรวม **รูปที่ 4.11** ทดลองการเคลื่อนที่บนเส้นทางชันในสภาพสนามรวม

วิเคราะห์ผลการทดลอง

ระดับความเข้มของค่าสีที่เซนเซอร์แสงรับมาประมวลเพื่อสั่งงานมอเตอร์และระดับความเร็วที่เคลื่อนที่ของหุ่นยนต์ในแต่ลักษณะของสภาพสนามมีผลต่อการทรงตัวและตัดสินใจเลือกเส้นทางที่ถูกต้องของหุ่นยนต์ โดยหุ่นยนต์ไม่สามารถเคลื่อนที่ในทุกสภาพสนามที่ระดับความเร็วระดับเดียวตลอดเส้นทางได้ เนื่องจากสภาพสนามในแต่ละลักษณะแตกต่างกัน ดังนั้นหุ่นยนต์จะต้องมีการปรับเปลี่ยนระดับความเร็วให้เหมาะสมตามสภาพสนามที่กำลังเคลื่อนที่อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลโครงงาน

ในการศึกษาวิธีการออกแบบและพัฒนาระบบหุ่นยนต์ติดตามเส้นทาง(Line trace robot) เพื่อศึกษาและพัฒนาแอปพลิเคชั่นระบบฝังตัว โดยศึกษาจากแนวทางในการพัฒนาระบบฝังตัว ฮาร์ดแวร์(Hardware) และซอฟต์แวร์ (Software) และทำการออกแบบแอปพลิเคชั่นหุ่นยนต์ติดตามเส้นทาง จากแนวทางการพัฒนาระบบที่ศึกษามา รวมทั้งทำการเขียนโปรแกรมเพื่อสั่งงานให้ทำงานได้ตรงตามที่ต้องการรวมถึงทำการทดสอบและสรุปผลที่ได้ โดยหุ่นยนต์ติดตามเส้นทางจะมีหลักการทำงานคือจะทำการอ่านค่าข้อมูลจากสภาพแวดล้อมผ่านทางอุปกรณ์เซ็นเซอร์ (Sensor) และนำข้อมูลดังกล่าวมาประมวลผลเพื่อให้ทำงานได้ถูกต้องตามที่กำหนด คือ สามารถติดตามเส้นทางเส้นทาง เลือกเส้นทาง ปรับระดับความเร็วในการเคลื่อนที่และเคลื่อนที่ไปยังเป้าหมายที่กำหนดได้

จากการศึกษาและทดลองทำโครงงานนี้ทำให้สามารถเรียนรู้แนวทางในการพัฒนาระบบและประยุกต์ใช้งานระบบฝังตัวได้ โดยสามารถพัฒนาระบบฝังตัวโดยใช้เครื่องมือจากแนวทางในการพัฒนาระบบอย่างเหมาะสม และสามารถนำมาออกแบบและพัฒนาแอพพลิเคชั่นระบบฝังตัวได้ ซึ่งจะช่วยให้การพัฒนางานในด้านระบบฝังตัวมีรูปแบบและขั้นตอนในการพัฒนาอย่างเป็นระบบยิ่งขึ้น และการประยุกต์ใช้เครื่องมือ (tool) ที่ทันสมัยและเหมาะสมกับการพัฒนาระบบงานในปัจจุบัน จะทำให้การพัฒนางานด้านนี้เป็นไปได้อย่างรวดเร็ว ซึ่งในปัจจุบันและอนาคตระบบฝังตัวจะมีบทบาทมากยิ่งขึ้น โครงงานนี้จึงเป็นอีกแนวทางที่ดีที่ช่วยให้การพัฒนางานด้านนี้เป็นไปอย่างเป็นระบบ เหมาะสม และสมบูรณ์ยิ่งขึ้น สามารถที่จะประยุกต์ใช้กับแอพพลิเคชั่นได้หลากหลาย ช่วยลดค่าใช้จ่ายในการพัฒนาและเพิ่มขีดความสามารถให้กับงานด้านระบบฝังตัวมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและอุปสรรค

5.2.1 ปัญหาทางด้านฮาร์ดแวร์

ปัญหาและอุปสรรคที่เกิดจากทางด้านฮาร์ดแวร์ ได้แก่

1. ปัญหาการรับค่าแสงจากเซ็นเซอร์แสง เนื่องจากการทำงานของหุ่นยนต์จะต้องมีการรับค่าข้อมูลเพื่อใช้ในการประมวลผลเป็นส่วนใหญ่ แต่เซ็นเซอร์แสงที่ใช้ในการพัฒนาเป็นเซ็นเซอร์แสงที่มีการรับค่าโดยการอ่านค่าความเข้มแสง ดังนั้นในที่ที่สภาพแวดล้อมต่างกันย่อมทำให้ค่าความเข้มแสงต่างกัน และในช่วงเวลาที่ต่าง ๆ หรือตำแหน่งในการวางเซ็นเซอร์แสงในระยะที่ต่างกันในสภาพแวดล้อมเดียวกันก็ย่อมให้ค่าความเข้มแสงที่ต่างกัน ดังนั้นจึงเป็นอุปสรรคต่อการกำหนดค่าตัวแปรต่าง ๆ ที่ต้องใช้ค่าจากความเข้มแสงในการเปรียบเทียบ หรือทำให้เกิดปัญหาความไม่แน่นอนในการทำงานของโปรแกรม ที่อาจทำให้โปรแกรมทำงานผิดพลาดเนื่องจากอ่านค่าแสงที่ผิดเพี้ยนไปจากเดิม

นอกจากนี้ยังส่งผลถึงการสร้างสนามทดลองที่ใช้ในการทดสอบระบบ ซึ่งต้องทำการปรับแก้สนามและเลือกใช้อุปกรณ์ในการสร้างสนามทดลองที่เหมาะสมและให้ค่าความเข้มแสงที่ชัดเจนที่สุดเพื่อลดปัญหาในเรื่องของความเข้มแสงที่ต่างกันด้วย

ดังนั้นในการใช้งานเซนเซอร์แสงอาจจำเป็นต้องติดตั้งเซนเซอร์แสงไว้ให้ใกล้กับสิ่งที่ต้องการเก็บค่ามากที่สุด หรืออาจใช้กระดาษทำการปิดรอบบริเวณส่วนของการรับแสงของเซนเซอร์เพื่อลดการรบกวนจากแสงภายนอก

2. โครงสร้างของการออกแบบส่วนของไดร์วิงมอเตอร์ (Driving motor) ทำให้กำลังมอเตอร์ต่างกันได้ โดยการออกแบบการทดรอบมอเตอร์ด้วยเฟืองที่มีลักษณะแตกต่างกันและจำนวนที่ต่างกัน ซึ่งจะทำให้ได้แรงบิด (Torque) ของมอเตอร์ที่แตกต่างกัน ดังนั้นจึงมีผลต่อการเคลื่อนที่ในลักษณะที่ต้องอาศัยการปรับความเร็วในการเคลื่อนที่ เช่น การเคลื่อนที่ขึ้นทางลาดชัน ซึ่งจำเป็นต้องออกแบบส่วนของไดร์วิงให้เหมาะสมและมีกำลังมอเตอร์มากพอที่จะเคลื่อนที่ไปในลักษณะเส้นทางดังกล่าวได้

3. ปัญหาด้านพลังงานที่ใช้ในการทำงานของระบบ หุ่นยนต์ติดตามเส้นทางจะทำงานโดยอาศัยพลังงานจากแบตเตอรี่โดยใช้แบตเตอรี่ขนาด 9 โวลต์ ซึ่งระดับพลังงานของแบตเตอรี่จะมีผลต่อการขับเคลื่อนมอเตอร์ของหุ่นยนต์ ซึ่งระดับพลังงานที่น้อยหรือต่ำอาจส่งผลให้การทำงานผิดพลาดได้ เช่น ความเร็วของการหมุนมอเตอร์ที่เซตค่าไว้ที่ระดับหนึ่งอาจทำงานได้ไม่ตรงตามที่ต้องการได้หรือได้ความเร็วที่ต่ำเนื่องจากแบตเตอรี่อ่อนกำลัง และต้องทำการเปลี่ยนแบตเตอรี่ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งต้องทำการดาวน์โหลดโปรแกรมลงไปใหม่เช่นกัน ซึ่งอาจทำการตรวจสอบพลังงานของแบตเตอรี่ในขณะที่หุ่นยนต์ทำการเคลื่อนที่ เพื่อปรับความเร็วให้เหมาะสมกับการทำงาน

5.2.2 ปัญหาทางด้านซอฟต์แวร์

ปัญหาทางด้านซอฟต์แวร์ของโปรแกรมได้แก่ปัญหาในด้านการคอมไพล์และดาวน์โหลดโปรแกรมลงอาร์ซีเอ็กซ์ผ่านอินฟราเรดไออาร์ทาวเวอร์ โดยทุกครั้งที่มีการถอดแบตเตอรี่จะต้องทำการดาวน์โหลดเฟิร์มแวร์และโปรแกรมลงไปใหม่ทุกครั้งหุ่นยนต์จึงจะสามารถทำงานได้ เนื่องจากข้อมูลที่เก็บไว้จะหายไป ซึ่งทำให้การทำงานเกิดความยุ่งยากและต้องทำการตรวจสอบก่อนทุกครั้งว่ามีโปรแกรมอยู่หรือไม่

นอกจากนี้ยังมีปัญหาในขั้นตอนของการติดตั้งโปรแกรมที่ใช้ในการดาวน์โหลดและคอมไพล์โปรแกรม ซึ่งได้แก่โปรแกรมซิกวิน (Cygwin) ซึ่งเป็นครอส-คอมไพเลอร์ในการคอมไพล์โปรแกรมภาษาซีพลัสพลัสให้อาร์ซีเอ็กซ์เข้าใจ และต้องทำการดาวน์โหลดเฟิร์มแวร์ ซึ่งอาจเกิดปัญหาของการดาวน์โหลดเฟิร์มแวร์ซ้อนทับกันหรือใช้เฟิร์มแวร์ที่ต่างกัน จะทำให้เกิดข้อผิดพลาดในการคอมไพล์โปรแกรมและดาวน์โหลดโปรแกรมได้

5.3 ข้อเสนอแนะ

สำหรับข้อเสนอแนะและแนวทางในการพัฒนาหุ่นยนต์ติดตามเส้นทาง สามารถนำไปพัฒนาแอพพลิเคชั่นให้มีความสามารถที่หลากหลายมากยิ่งขึ้น เช่นสามารถตรวจจับสิ่งกีดขวางได้ สามารถตรวจจับสัญญาณเช่นสัญญาณไฟจราจร เพื่อช่วยให้สามารถนำไปประยุกต์ใช้งานจริงกับรถยนต์ เพื่อให้สามารถขับเคลื่อนได้อย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. **การวิเคราะห์และออกแบบระบบเชิงวัตถุด้วย UML**. กรุงเทพฯ : สำนักพิมพ์ เคทีพี.

ประยงค์ อู่ประสิทธิ์วงศ์. 2549. **หลักการเขียนโปรแกรมและแก้ปัญหาด้วยภาษา C++**. กรุงเทพฯ : โรงพิมพ์โสภณการพิมพ์.

JAPAN SYSTEM HOUSE ASOCIATION (JASA). **เทคโนโลยีสมองกลฝังตัว**. แปลโดย ดร.ธนา รักษ์ ธีระมั่นคง. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท.

Joe Nagata. **ประดิษฐ์หุ่นยนต์ LEGO MINDSTORMS**. แปลโดย นันท์ชญา มหาขันธ์. กรุงเทพฯ : นานมีบุ๊คส์พับลิเคชั่นส์.

Brian W. Kernighan, Dennis H. Ritchie. "**RCX**." [Online]. Available : http://legolab.daimi.au.dk/DigitalControl.dir/RCX/Manual.dir/RCXManual.html. 2006

Embedded.com Links. "**Embedded System.**" [Online]. Available : http://www.embedded.com. 2007.

SOURCEFORCE.NET. **"C++ Language API reference."** [Online]. Available : http://brickos.sourceforge.net/docs/APIs/html-c++/. 2002.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ขั้นตอนการติดตั้ง BrickOS บน Windows XP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้ง BrickOS บน Windows XP

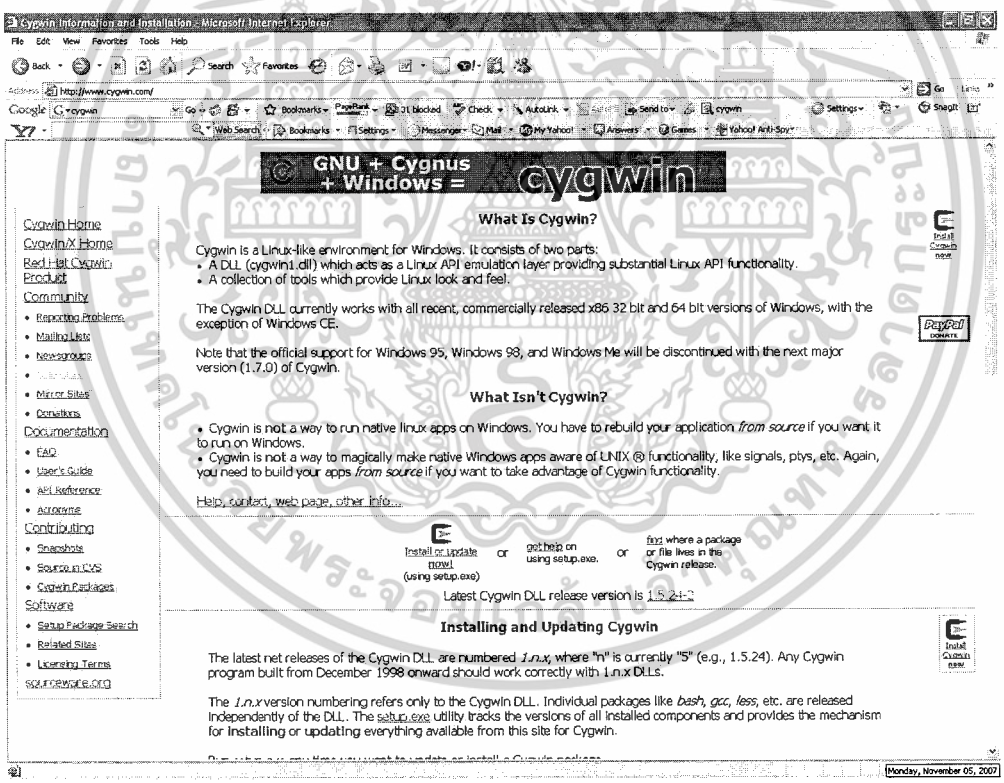
1. ติดตั้งโปรแกรม Cygwin

โปรแกรม Cygwin เป็นโปรแกรมที่มีสภาพแวดล้อมการทำงานเหมือนลินุกซ์ แต่ใช้สำหรับระบบปฏิบัติการวินโดว์ โดยมีส่วนประกอบที่สำคัญ 2 ส่วนคือ

- DLL ซึ่งทำหน้าที่เป็นตัวจำลอง ฟังก์ชั่น API ของลินุกซ์ต่าง ๆ ให้สามารถใช้งานได้ภายใต้สภาพแวดล้อมของระบบปฏิบัติการวินโดว์ และ
- ส่วนของเครื่องมือต่าง ๆ ที่สนับสนุนการทำงานภายใต้รูปแบบของลินุกซ์บนวินโดว์

1.1 ขั้นตอนการติดตั้งโปรแกรม Cygwin

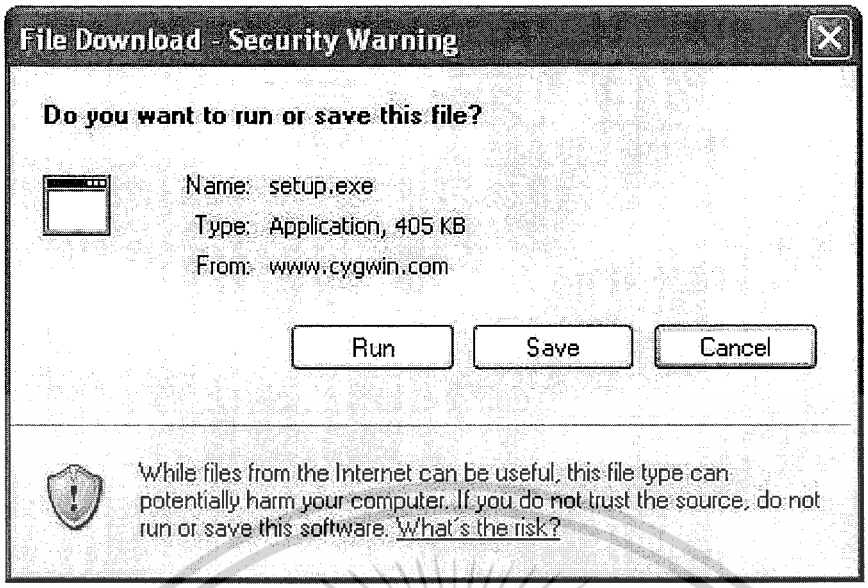
1.1.1 เข้าไปที่เว็ปไซต์ cygwin (http://www.cygwin.com) เพื่อทำการดาวโหลดโปรแกรมมาติดตั้ง ดังรูปที่ ก.1



รูปที่ ก.1 แสดงเว็ปไซต์ cygwin

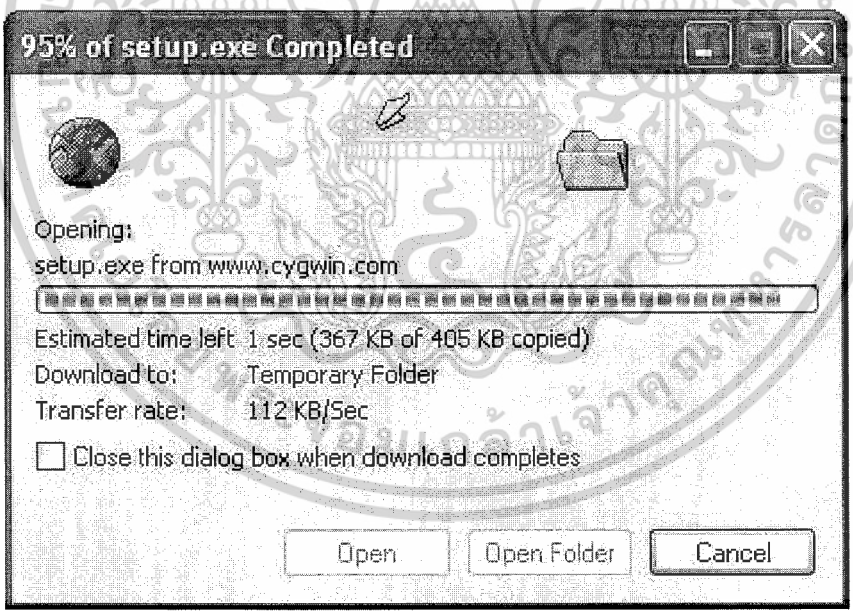
1.1.2 ให้กดปุ่มไอคอนInstall or update now! เพื่อทำการรันโปรแกรม โดยให้กดปุ่ม Run ดังรูปที่ ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 แสดงหน้าต่างการรันโปรแกรม

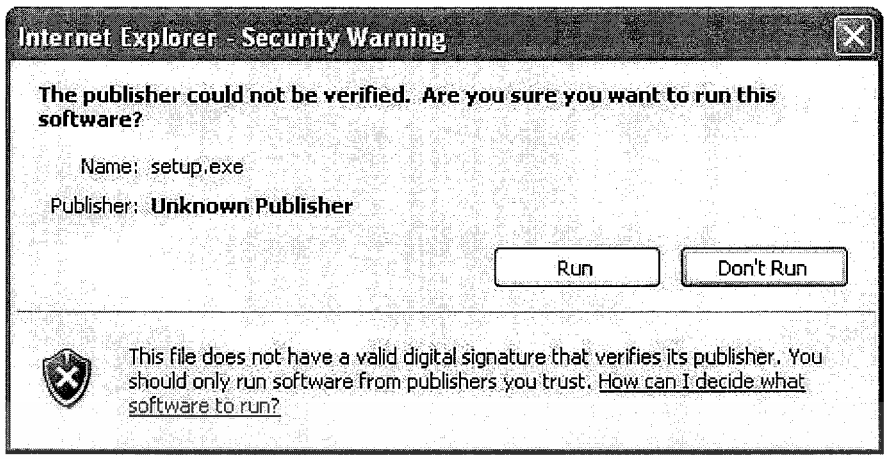
1.1.3 เครื่องจะทำการดาวโหลดไฟล์ที่จำเป็นสำหรับโปรแกรม setup.exe มาเก็บไว้บนเครื่อง



รูปที่ ก.3 แสดงหน้าต่างการดาวโหลดไฟล์

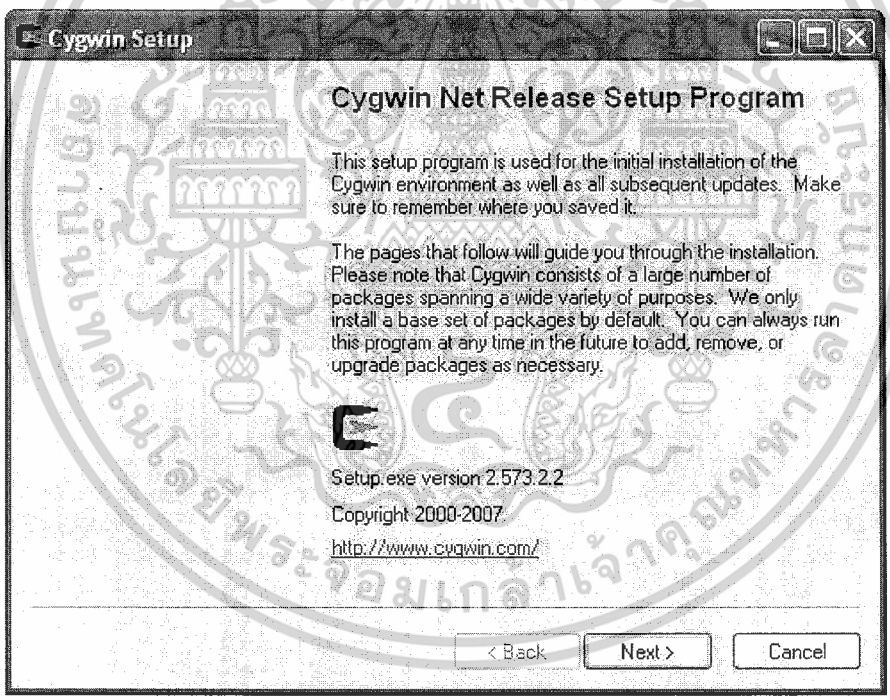
1.1.4 เมื่อดาวโหลดเสร็จเรียบร้อยให้ทำการกดปุ่ม Run เพื่อทำการรันโปรแกรม ดังรูปที่ ก.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 แสดงหน้าต่างการดาวโหลดไฟล์เสร็จเรียบร้อย

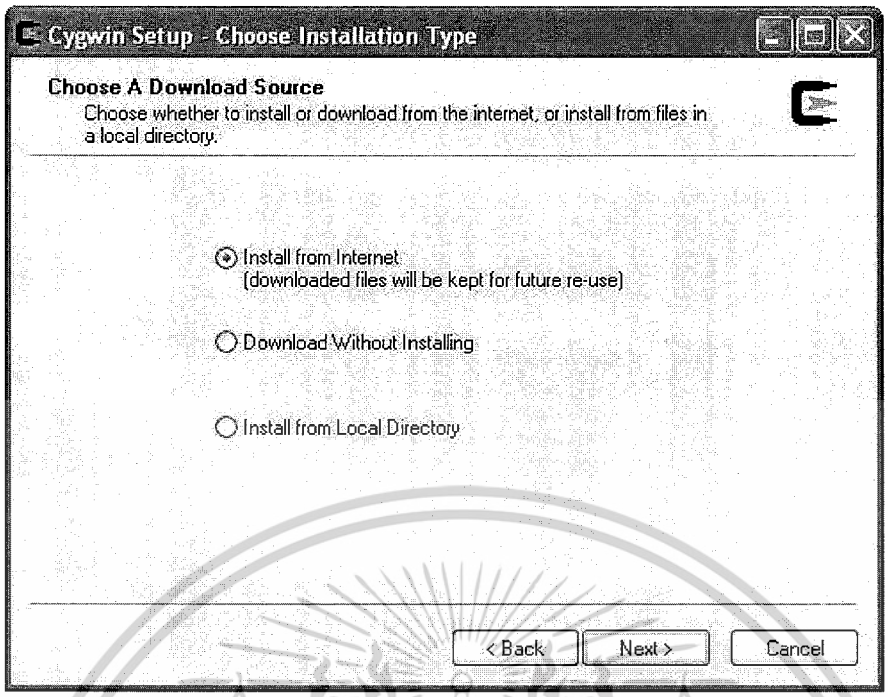
1.1.5 จากนั้นคลิ๊กปุ่ม Next เพื่อให้โปรแกรมทำการติดตั้งโปรแกรม Cygwin รวมทั้งสภาพแวดล้อมในการใช้งาน



รูปที่ ก.5 แสดงหน้าต่างการติดตั้งโปรแกรม Cygwin

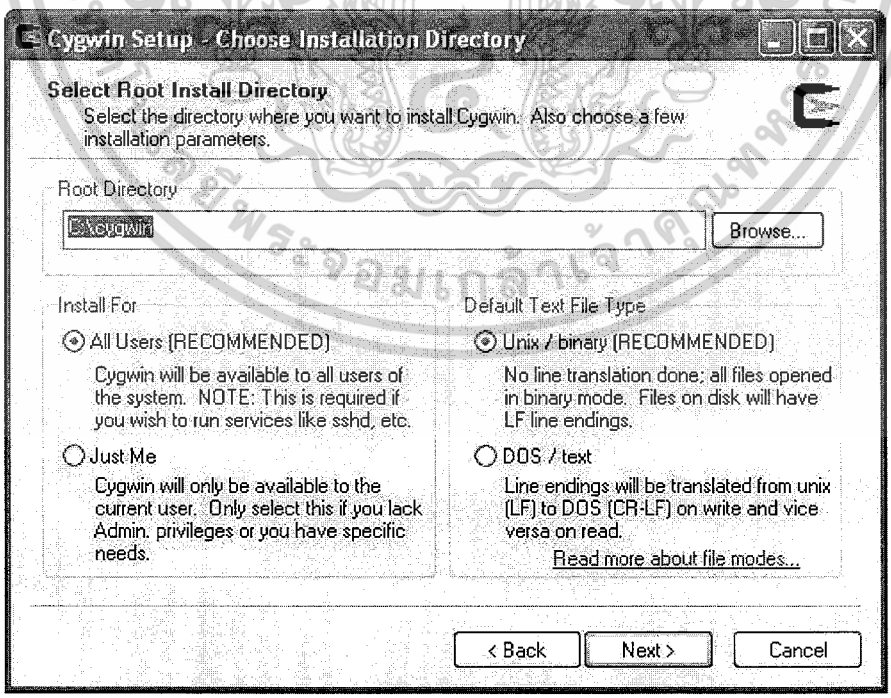
1.1.6 จากนั้นให้เลือกรูปแบบการติดตั้งไปที่ Install from Internet แล้วคลิ๊กปุ่ม Next ดังรูปที่ ก.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.6 แสดงหน้าต่างการติดตั้งโปรแกรม Cygwin ขั้นที่ 2

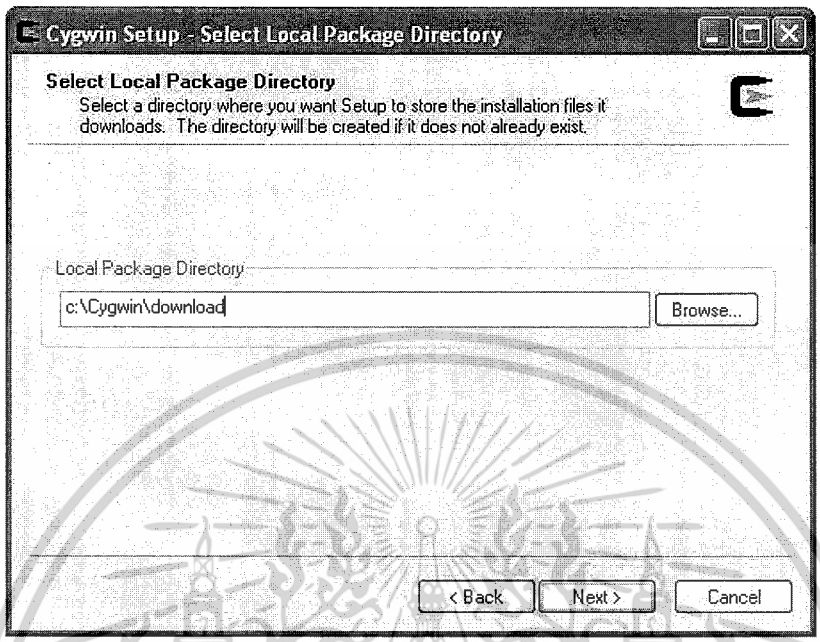
1.1.7 เสร็จแล้วกำหนดไดเร็กทอรี่หลัก (Root directory) ที่จะทำการติดตั้งโปรแกรม โดยเลือกรูปแบบการติดตั้งสำหรับใช้งาน (Install for) และรูปแบบของตัวอักษร (Default Text File Type) ดังรูป จากนั้นคลิ๊กปุ่ม Next



รูปที่ ก.7 แสดงหน้าต่างการกำหนดไดเร็กทอรี่หลักที่จะทำการติดตั้งโปรแกรม

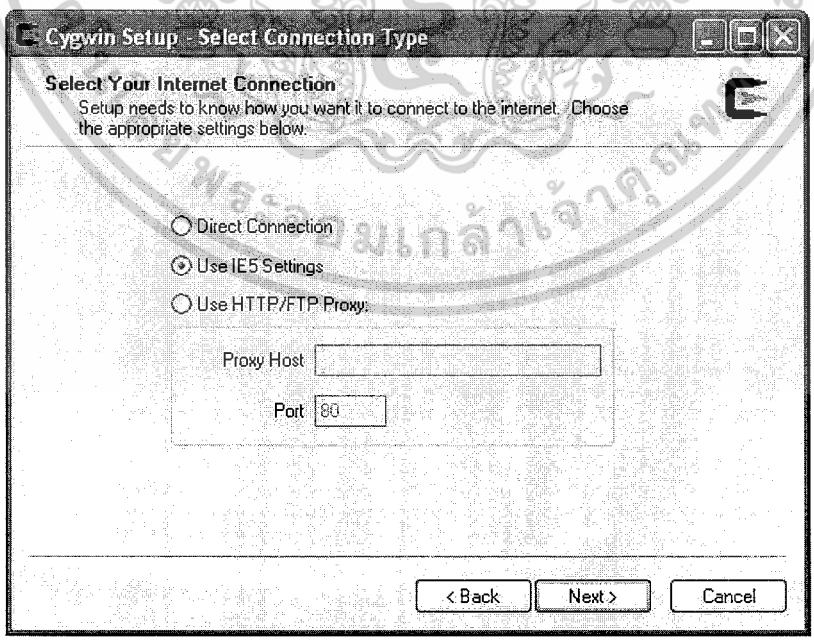
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.8 ทำการกำหนดไดเร็กทอรี่ที่จะทำการเก็บไฟล์ที่เกี่ยวข้องในการติดตั้ง(ซึ่งจะถูกดาวโหลดมาเก็บบนเครื่องดังรูป) จากนั้นคลิ๊ก Next



รูปที่ ก.8 แสดงหน้าต่างการกำหนดไดเร็กทอรี่ที่จะทำการเก็บไฟล์ที่เกี่ยวข้องในการติดตั้ง

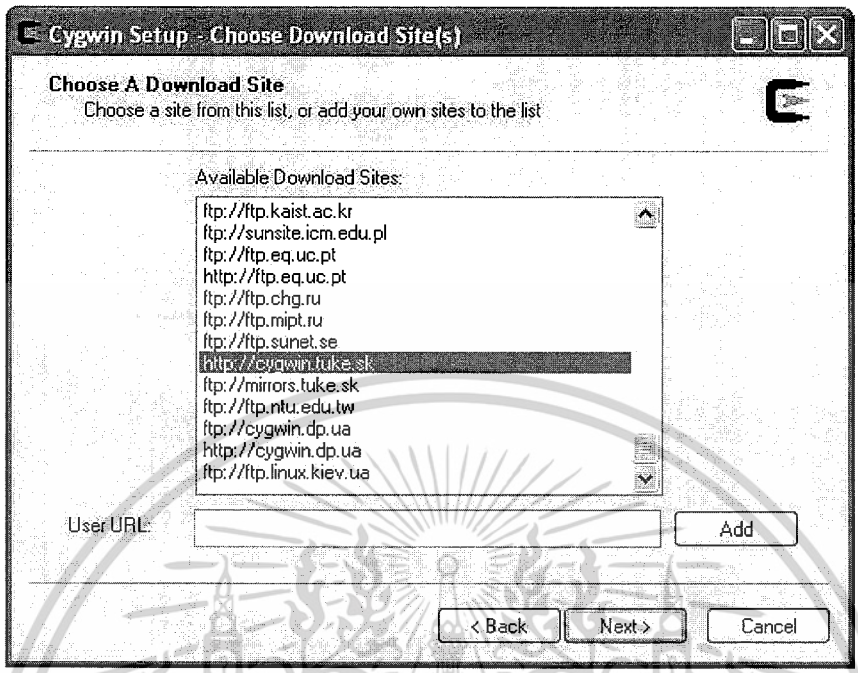
1.1.9 เลือกรูปแบบการเชื่อมต่อกับอินเตอร์เน็ทที่จะใช้ในการติดตั้ง แล้วคลิ๊กปุ่ม Next ดังรูปที่ ก.9



รูปที่ ก.9 แสดงหน้าต่างการเลือกรูปแบบการเชื่อมต่อกับอินเตอร์เน็ท

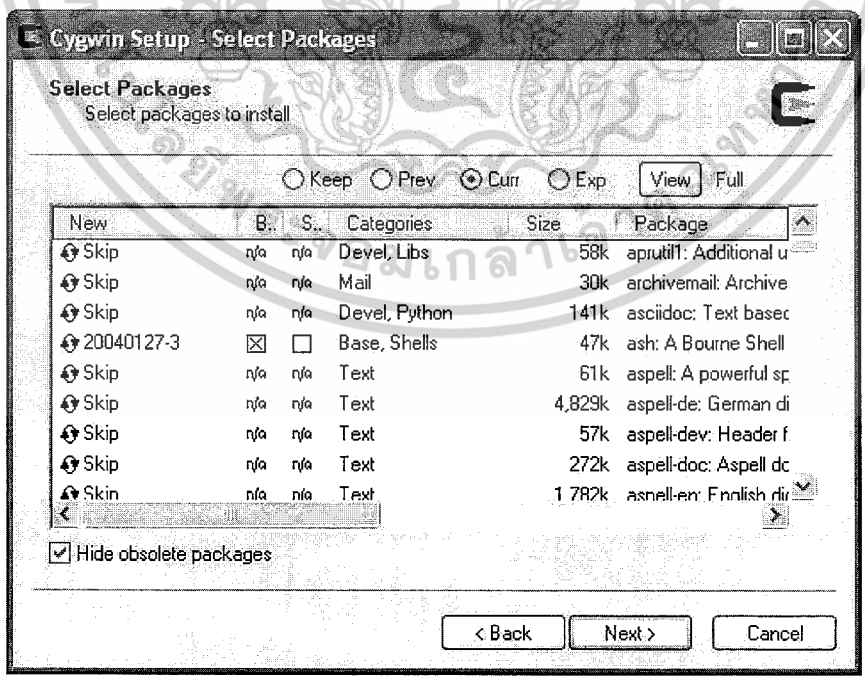
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.10 ทำการเลือกเว็ปไซท์ (Mirror sites) ที่จะทำการดาวโหลดโปรแกรม แล้วคลิ๊กปุ่ม Next



รูปที่ ก.10 แสดงหน้าต่างการเลือกเว็ปไซท์ที่จะทำการดาวโหลดโปรแกรม

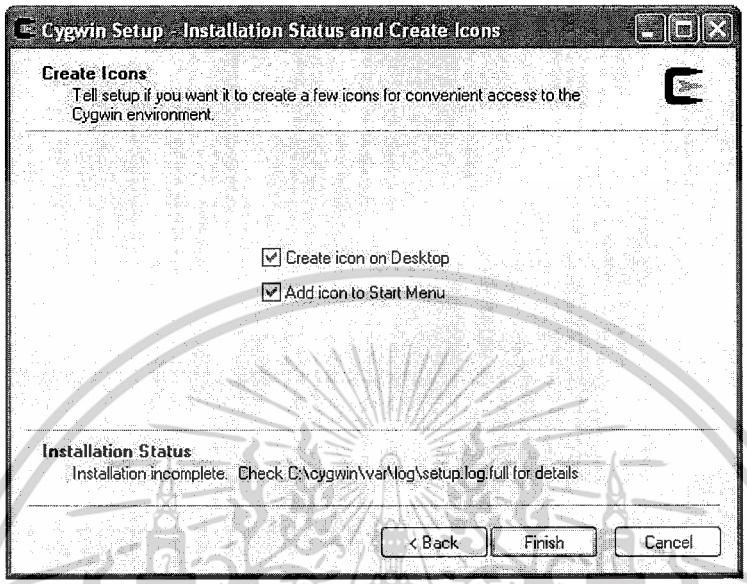
1.1.11 เครื่องจะทำการดาวโหลดโปรแกรมจนเสร็จ จากนั้นจะปรากฏหน้าต่างให้เลือกแพ็คเกจ (Packages) ที่จะทำการติดตั้ง ดังรูปที่ ก.11 จากนั้นให้คลิ๊กปุ่ม Next



รูปที่ ก.11 แสดงหน้าต่างแพ็คเกจ (Packages) ที่ต้องเลือกเพื่อทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.12 เครื่องจะทำการดาวโหลดแพ็คเกจที่เลือกมาทำการติดตั้งจนเสร็จ (ประมาณ 2-3 นาที ขึ้นกับความเร็ว) จากนั้นให้เลือกว่าจะสร้างไอคอนบน Desktop หรือใน Start menu แล้วคลิ๊กปุ่ม Finish เพื่อออกจากการติดตั้ง ดังรูปที่ ก.12



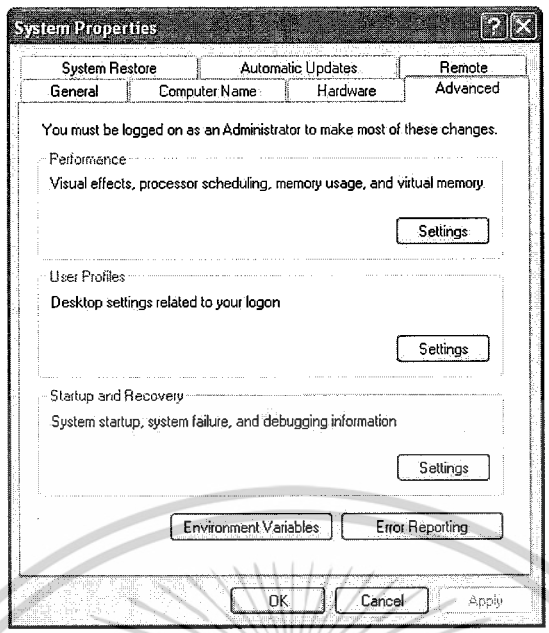
รูปที่ ก.12 แสดงหน้าต่างการเลือกไอคอนบน Desktop

หากการติดตั้งสมบูรณ์ จะปรากฏหน้าต่างการติดตั้งเสร็จสิ้นสมบูรณ์ (Install completed) ขึ้น ให้คลิ๊กปุ่ม Ok

1.1.13 ให้ตรวจสอบไดเร็กทอรี่ c:\cygwin\bin และ c:\cygwin\var ว่ามีการจัดสร้างและมีไฟล์อยู่หรือไม่ (หากการติดตั้งไม่สมบูรณ์ จะไม่มีไฟล์ใด ๆ ในไดเร็กทอรี่ดังกล่าว)

1.1.14 ทำการกำหนดตัวแปร (variable) และเส้นทาง (path) ของโปรแกรม cygwin ให้กับระบบปฏิบัติการวินโดว์ โดยเลือกที่ไอคอน My computer แล้วคลิ๊กเม้าท์ปุ่มขวา เลือก Properties จะปรากฏหน้าต่าง System Properties ดังรูปที่ ก.13

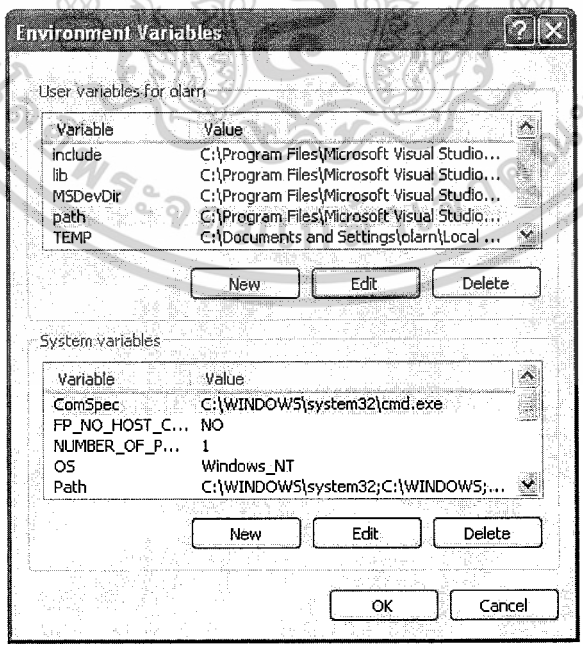
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.13 แสดงหน้าต่างการตัวแปรและเส้นทางของโปรแกรม cygwin

จากนั้น เลือกไปที่แทป (Tab) Advanced แล้วคลิ๊กปุ่มตัวแปรสภาพแวดล้อม (environment variables)

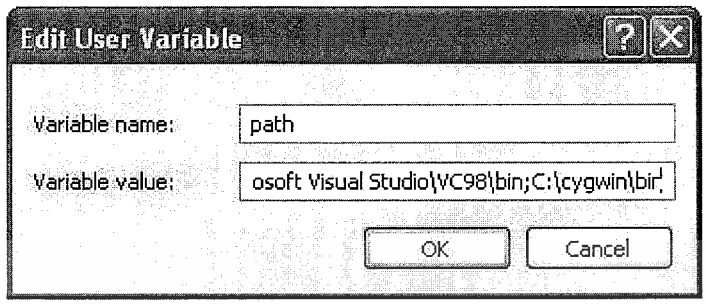
1.1.15 ทำการเลือกตัวแปร (Variable) เส้นทาง (Path) ในช่องตัวแปรสำหรับผู้ใช้ (User variables) แล้วทำการคลิ๊กปุ่ม Edit เพื่อทำการเพิ่มค่าตัวแปร ดังรูป (หากยังไม่มีตัวแปร Path ให้คลิ๊กที่ปุ่ม New เพื่อสร้างขึ้นใหม่)



รูปที่ ก.14 แสดงหน้าต่างการเพิ่มค่าตัวแปรในช่องตัวแปรสำหรับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.16 ทำการเพิ่มค่าตัวแปร (Variable value) ภายใต้ตัวแปรชื่อ Path โดยพิมพ์ค่า ;c:\cygwin\bin ต่อท้ายในช่องค่าตัวแปร (Variable value) ดังรูป แล้วคลิ๊ก OK



รูปที่ ก.15 แสดงหน้าต่างการเพิ่มค่าตัวแปรภายใต้ตัวแปรชื่อ Path

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

2. ติดตั้งโปรแกรม BrickOS

โปรแกรม BrickOS เป็นระบบปฏิบัติการฝังตัวแบบเปิดเผย (open source embedded OS) รวมทั้งกำหนดสภาพแวดล้อมการโปรแกรมสำหรับชุดอุปกรณ์ Lego Mindstorms Robotics Kits ด้วยภาษา C และ C++ เพื่อให้ผู้ใช้งานสามารถโปรแกรมชุดอุปกรณ์ Lego ดังกล่าวได้ด้วยตนเอง (แทนที่จะใช้ภาษาโปรแกรมมาตรฐานของ Lego)

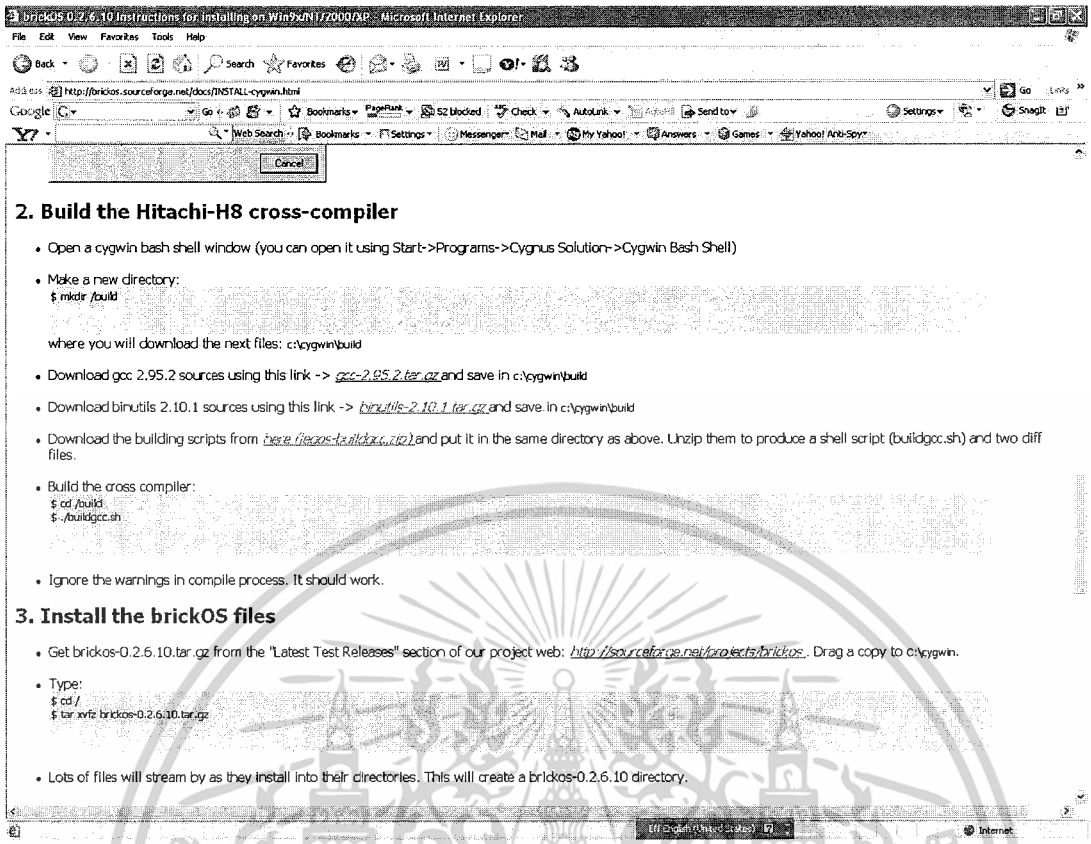
โปรแกรม BrickOS ได้รับการพัฒนาขึ้นภายใต้สภาพแวดล้อมแบบลินุกซ์ แต่สามารถทำงานได้บนระบบปฏิบัติการวินโดว์ (โดยความช่วยเหลือจากโปรแกรม Cygwin ข้างต้น) ในการใช้งาน BrickOS จะต้องมีการติดตั้งคอมไพเลอร์ของฮิตาชิ (Hitachi H8 cross compiler assembler) และตัวลิงก์ (Linker) ที่เกี่ยวข้อง ดังนี้

2.1 ขั้นตอนการการติดตั้งโปรแกรม BrickOS

เข้าไปที่เว็ปไซด์ http://birckos.sourceforge.net/docs/INSTALL-cygwin.html ดังรูป เพื่อเตรียมการดาวโหลดไฟล์ต่าง ๆ สำหรับการติดตั้งดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.16 แสดงหน้าต่างเว็ปไซต์ที่ใช้ดาวโหลดไฟล์ต่าง ๆ สำหรับการติดตั้ง

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

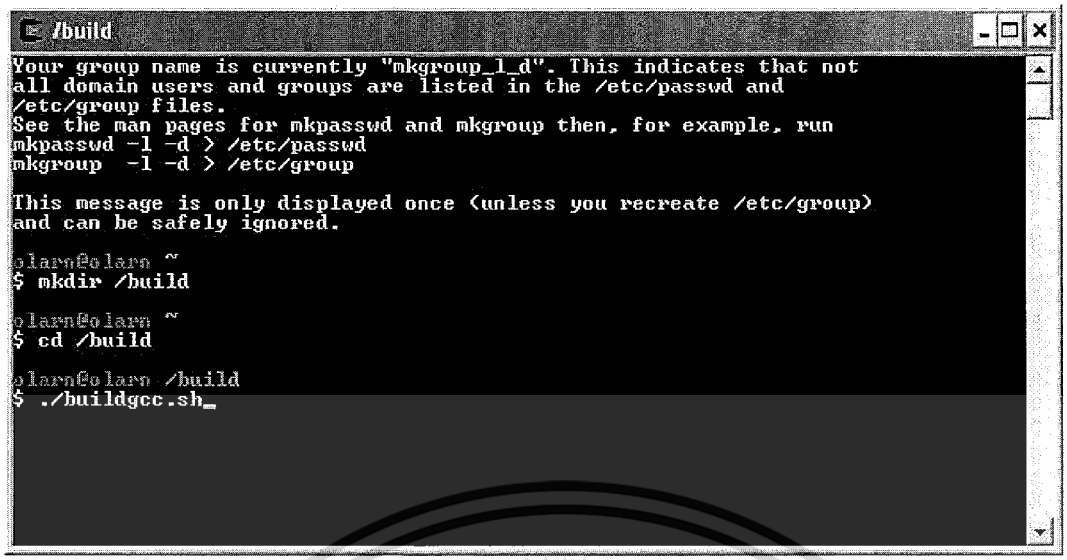
จากนั้นทำการ unzip ไฟล์ legos-buildgcc.zip เพื่อจะให้ได้ไฟล์ดังต่อไปนี้

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

รันโปรแกรม cygwin จากนั้นทำการรันโปรแกรม buildgcc.sh เพื่อสร้างสภาพแวดล้อมของตัวคอมไพเลอร์ (cross compiler environment) โดยพิมพ์คำสั่งต่าง ๆ ดังนี้

- cd /build
- ./buildgcc.sh

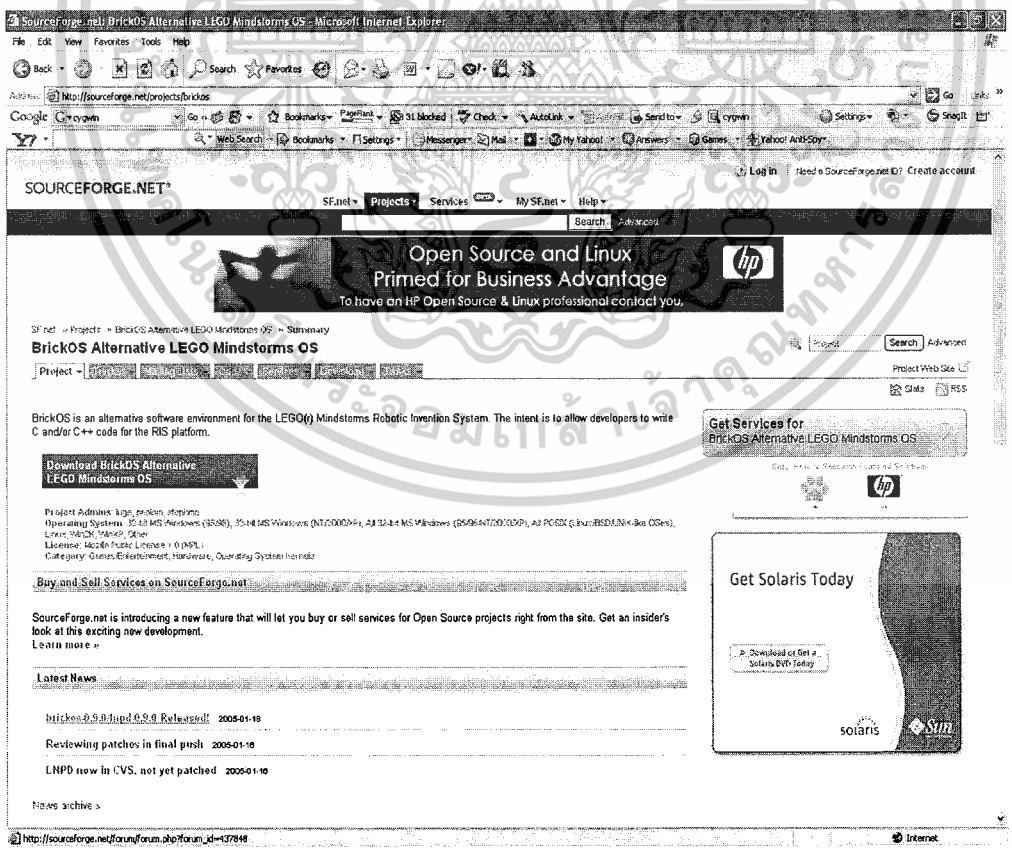
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.17 แสดงการรันโปรแกรม buildgcc.sh เพื่อสร้างสภาพแวดล้อมของตัวคอมไพเลอร์

เข้าไปที่เว็ปไซต์ http://sourceforge.net/projects/birckos ดังรูป เพื่อทำการดาวโหลดไฟล์ต่อไปนี้มาเก็บไว้ในไดเร็กทอรี่ C:\cygwin

- brickos-0.9.0.tar.gz



รูปที่ ก.18 แสดงเว็ปไซต์ที่ใช้ในการดาวโหลดไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสร็จแล้วกลับไปที่โปรแกรม cygwin ทำการอันซิปไฟล์ที่ดาวโหลดมาโดยพิมพ์คำสั่งดังนี้

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

รูปที่ ก.19 แสดงการอันซิปไฟล์ที่ดาวน์โหลดมา

เมื่อทำการอันซิปไฟล์ในขั้นตอน 2.6 เสร็จเรียบร้อยแล้ว ให้ทำการกำหนดสภาพแวดล้อม (configure) สำหรับการคอมไพล์โปรแกรมเพื่อให้ หุ่นยนต์เคลื่อนไหวได้ โดยพิมพ์คำสั่งต่อไปนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.20 แสดงการกำหนดสภาพแวดล้อมสำหรับการคอมไพล์โปรแกรม

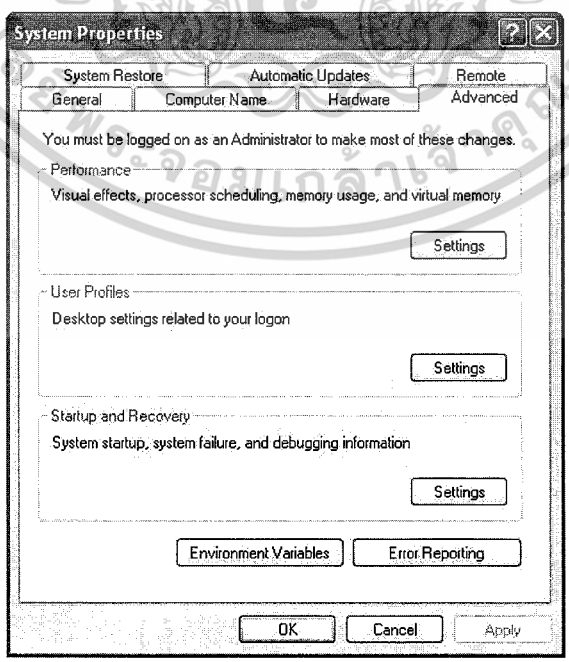
หากไม่มีข่าวสารข้อผิดพลาด (Error message) ปรากฏ หมายถึงการติดตั้งเสร็จสมบูรณ์

3. กำหนดตัวแปรให้กับ IR Tower

หลังจากติดตั้งโปรแกรมตัวขับอุปกรณ์ (Device driver) ของ IR Tower เสร็จแล้ว ให้ทำการตรวจสอบตัวแปรสภาพแวดล้อมภายใต้ระบบปฏิบัติการวินโดว์ ว่ามีการกำหนดค่าไว้แล้วหรือไม่ ดังนี้

3.1 ขั้นตอนการกำหนดตัวแปรให้กับ IR Tower

1. เลือกที่ไอคอน My computer บน Desktop แล้วคลิ๊กเม๊าท์ปุ่มขวา เลือก Properties จะปรากฏหน้าต่าง System Properties ดังรูปที่ ก.21

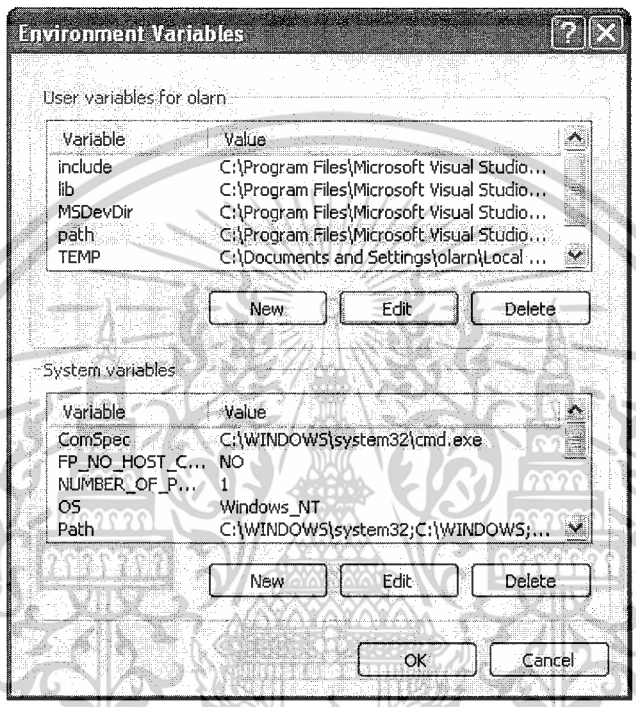


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.21 แสดงหน้าต่าง System Properties

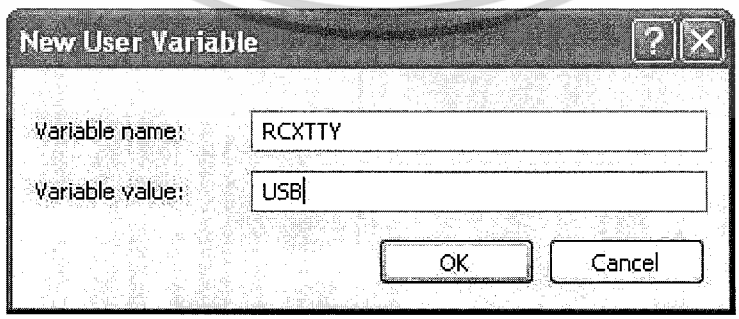
จากนั้น เลือกไปที่แทป (Tab) Advanced แล้วคลิ๊กปุ่มตัวแปรสภาพแวดล้อม (environment variables)

2. ตรวจสอบตัวแปร RCXTTY ว่ามีการสร้างขึ้นไว้หรือไม่ หากยังไม่มี ให้คลิ๊กที่ปุ่ม New เพื่อสร้างขึ้นใหม่ ดังรูปที่ ก.22



รูปที่ ก.22 แสดงหน้าต่างการตรวจสอบตัวแปร RCXTTY

3. ทำการเพิ่มตัวแปร (Variable name) ชื่อ RCXTTY โดยพิมพ์ค่าตัวแปร (Variable value) ชื่อ USB แล้วคลิ๊ก OK ดังรูปที่ ก.23



รูปที่ ก.23 แสดงหน้าต่างการเพิ่มตัวแปรชื่อ RCXTTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การถ่ายโอนเฟิร์มแวร์ (firmware) BrickOS ลงบน RCX

หลังจากติดตั้ง BrickOS เสร็จแล้ว ก็จะปรากฏไดเร็กทอรี่ย่อยคือ c:\cygwin\brickOS ปรากฏ จากนั้นให้ทำการถ่ายโอนเฟิร์มแวร์ของ birckOS ลงบน Lego RCX ตามขั้นตอนดังนี้

4.1 เชื่อมต่อ IR Tower เข้ากับพอร์ทอนุกรม (serial port) เช่น COM1 เป็นต้น ของเครื่องคอมพิวเตอร์ที่ได้ติดตั้งโปรแกรม cygwin และ brickOS รวมทั้งโปรแกรมตัวขับอุปกรณ์ (Device driver) ของ IR Tower ข้างต้น

4.2 เปิดสวิทช์ Power on บนตัวอาร์ซีเอ็กซ์เพื่อให้มันเริ่มทำงาน และจัดวางอุปกรณ์โดยหันพอร์ทอินฟาเรดบนตัวอาร์ซีเอ็กซ์ให้ตรงกับ IR Tower

4.3 รันโปรแกรม cygwin จากนั้นพิมพ์คำสั่งดังนี้ลงไป

/brickOS/util/firmdl3 –s /brickOS/boot/brickOS.srec

เครื่องจะทำการถ่ายโอนเฟิร์มแวร์ brickOS ลงบนอุปกรณ์อาร์ซีเอ็กซ์โดยจะแสดงสัดส่วน % การถ่ายโอนปรากฏบนหน้าจอจนครบ 100% ก็จะเสร็จสิ้น ดังรูปที่ ก.24

Transferring “/brickOS/boot/brickOS.srec” to RCX …

15%

รูปที่ ก.24 แสดงหน้าต่างการถ่ายโอนเฟิร์มแวร์ brickOS ลงบนอุปกรณ์อาร์ซีเอ็กซ์

<u>หมายเหตุ</u>

- หากมีแสงรบกวนแสงอินฟาเรด ขณะทำการถ่ายโอนข้อมูล อาจส่งผลต่อความผิดพลาดในการทำงาน ให้ใช้อุปกรณ์ทึบแสง เช่น กล่องทึบ เป็นต้น มาทำการคลอบอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RCX และไออาร์ทาวเวอร์ เพื่อไม่ให้แสงผ่านเข้าไปตรงช่วงระหว่างพอร์ทอินฟาเรดกับไออาร์ทาวเวอร์

- เมื่อการถ่ายโอนเฟิร์มแวร์ brickOS ลงบนตัวอุปกรณ์ RCX เสร็จเรียบร้อย ก็ถือว่าการติดตั้งเสร็จสมบูรณ์ เมื่อมีการโหลดโปรแกรมที่ต้องการลงบนตัว RCX ตัวอุปกรณ์ RCX ก็พร้อมที่จะทำงาน

หมายเหตุ

- การถ่ายโอนเฟิร์มแวร์ brickOS ที่ติดตั้งบนอาร์ซีเอ็กซ์ จะกระทำเพียงครั้งเดียวเท่านั้น

- หากเฟิร์มแวร์ดังกล่าวได้รับการติดตั้งบน RCX เรียบร้อยแล้ว ก็จะปรากฏข่าวสารข้อผิดพลาดดังนี้เกิดขึ้น

  - /brickOS/util/firmdl3: delete firmware failed

- หากต้องการลบเฟิร์มแวร์ brickOS ออกจากอาร์ซีเอ็กซ์ให้ถอดแบตเตอรี่ออกจากตัว RCX แล้วกดปุ่ม on/off

5. การคอมไพล์ซอร์ชโค้ด

ก่อนการงานซอร์ชโค้ดที่เขียนขึ้น ต้องมีการคอมไพล์ซอร์ชโค้ดก่อนเพื่อตรวจสอบข้อผิดพลาดของซอร์ชโค้ด โดยจะใช้คำสั่งดังต่อไปนี้

5.1 ใช้คำสั่งต่อไปนี้ เพื่อเข้าสู่ไดเรคทรอรีที่เก็บซอร์ชโค้ดที่ต้องการคอมไพล์

- cd /brickOS
- cd demo

5.2 ใช้คำสั่งต่อไปนี้เพื่อคอมไพล์ซอร์ชโค้ด

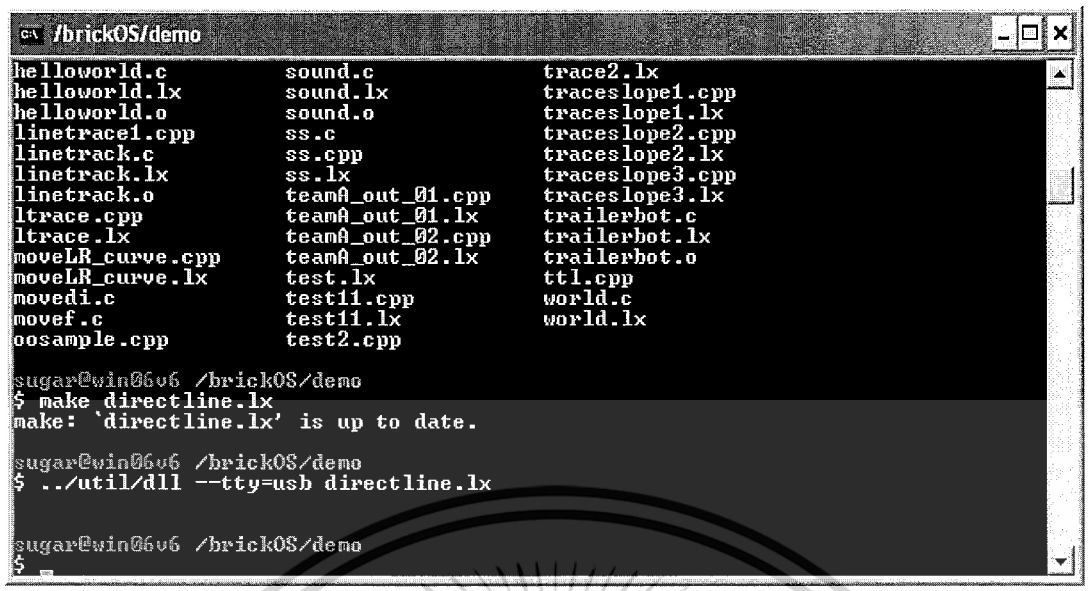
- make file_name.lx

ถ้าคอมไพล์ซอร์ชโค้ดผ่านจะไม่ปรากฏผลการผิดพลาดขึ้นดังรูปด้านล่าง

หมายเหตุ

หากมีการคอมไพล์ซอร์ชโค้ดไปแล้ว และยังไม่มีการแก้ไขใหม่ จะปรากฏข้อความว่า make : ‘file_name.lx’ is up to date. ดังรูปที่ ก.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.25 แสดงหน้าต่างการคอมไพล์ซอร์ซโค้ด

6. การดาวน์โหลดโปรแกรมลง RCX

หุ่นยนต์จะทำงานได้นั้น จะต้องมีการดาวน์โหลดโปรแกรมลงบน RCX ก่อนด้วยคำสั่งดังต่อไปนี้

- ../util/dll –tty=usb file_name.lx

ถ้าดาวน์โหลดโปรแกรมลง RCX สำเร็จ จะปรากฏข้อความดังต่อไปนี้

รูปที่ ก.26 แสดงหน้าต่างการถ้าดาวน์โหลดโปรแกรมลง RCX ที่สำเร็จ

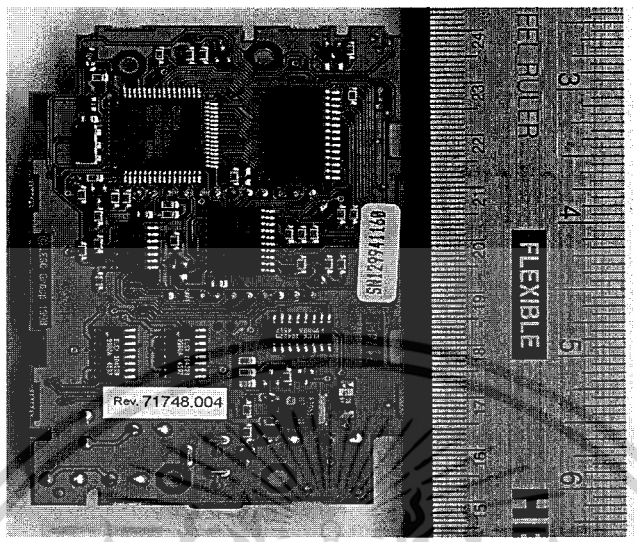
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

LEGO RCX Hitachi H8/3292

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LEGO RCX Hitachi H8/3292



รูปที่ ข.1 แสดงไมโครคอนโทรลเลอร์ Hitachi H8/3292

ส่วนประกอบของ LEGO RCX

- โมดูลควบคุมการโปรแกรมฮาร์ดแวร์
- อินเตอร์เฟสอินพุตและเอาท์พุต ได้แก่
  - เซนเซอร์
  - แอ็คทูเอเตอร์
- ไมโครคอนโทรลเลอร์ Hitachi H8/3292

อุปกรณ์อินพุตและเอาท์พุตของ LEGO RCX

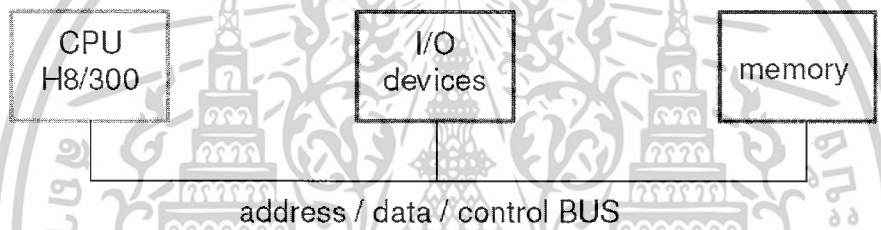
- อุปกร์อินพุต
  - พอร์ตการเชื่อมต่อเซนเซอร์ 3 พอร์ต
  - ปุ่ม 4 ปุ่ม
  - หน้าจอแสดงระดับแบตเตอรี่
  - ไทม์เมอร์
- อุปกรณ์เอาท์พุต
  - พอร์ตการเชื่อมต่อแอ็คทูเอเตอร์ 3 พอร์ต
  - LCD
  - สปีคเกอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

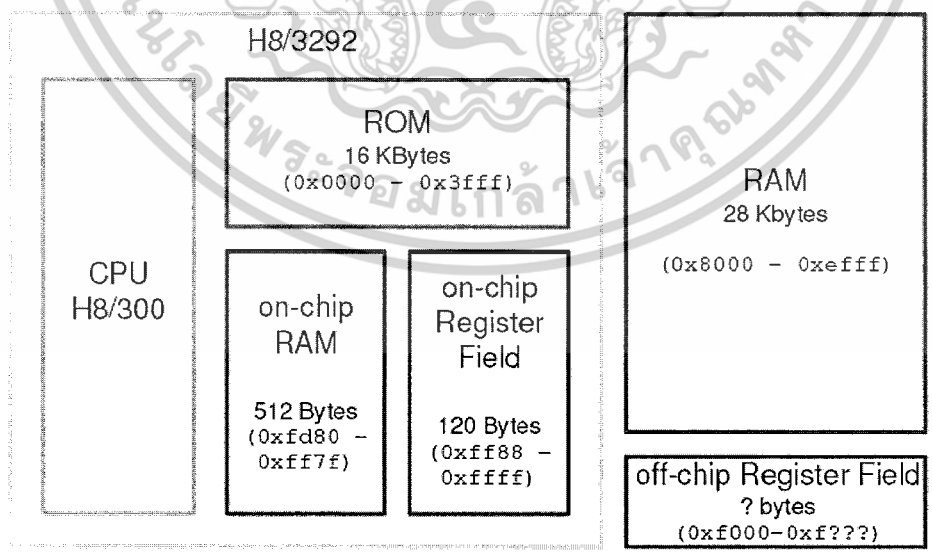
- **อุปกรณ์ส่วนติดต่อสองทิศทาง**
    - พอร์ตอินฟราเรด

หน่วยประมวลผลกลางของ RCX.

- ใช้ H8/300 เป็นหน่วยประมวลผลกลาง
- ใช้บิตข้อมูล 8 บิต
- ใช้บิตที่อยู่ 16 บิต
- 8 x 16 บิต เป็นรีจีสเตอร์ทั่วไป(r0 - r7)
    - r0 => function return
    - r7 => stack pointer
- สัญญาณนาฬิกา ความถี่ 16 เมกะเฮิร์ต



รูปที่ ข.2 แสดงการติดต่อระหว่างหน่วยประมวลผลกลาง อุปกรณ์อินพุตเอาท์พุตและหน่วยความจำ



รูปที่ ข.3 แสดงบล็อกไดอะแกรมของ Hitachi H8/3292

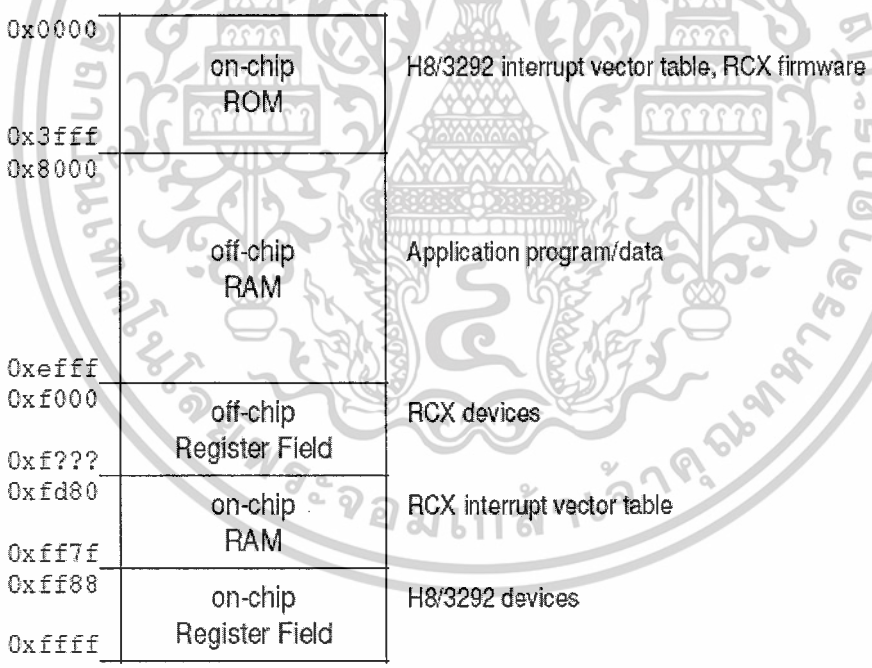
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดที่อยู่ของ Hitachi H8/300

- Register direct: rn
- Register indirect: @rn
- Register indirect with 16-bit displacement:@(d:16,rn)
- Register indirect with post-increment: @rn+
- Register indirect with pre-increment: @-rn
- Absolute address (8 or 16 bits): @aa:8, @aa:16
- Immediate (8 or 16-bit data): #aa:8, #aa:16
- PC-Relative (8-bit displacement): @(d:pc)
- Memory indirect: @@aa:8

โครงสร้างหน่วยความจำของ RCX

หน่วยประมวลผลย่อย โหมดที่ 2 (ควบคุมรีจิสเตอรที่ตำแหน่ง 0xffc5)



รูปที่ ข.4 แสดงโครงสร้างหน่วยความจำของ RCX

การอินเตอร์รัพท์แบบเร่งด่วนของ RCX

- ตารางการอินเทอร์รัพท์ ของ H8/3292
  - เก็บที่ตำแหน่ง 0x0000-0x0049 (บนรอมใน RCX)
  - RCX ROM vectors redirect interrupts to the on-chip RAM interrupt table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Decreasing priority-on-chip RAM interrupt table

- ตารางการอินเทอร์รัพท์ บนชิพ

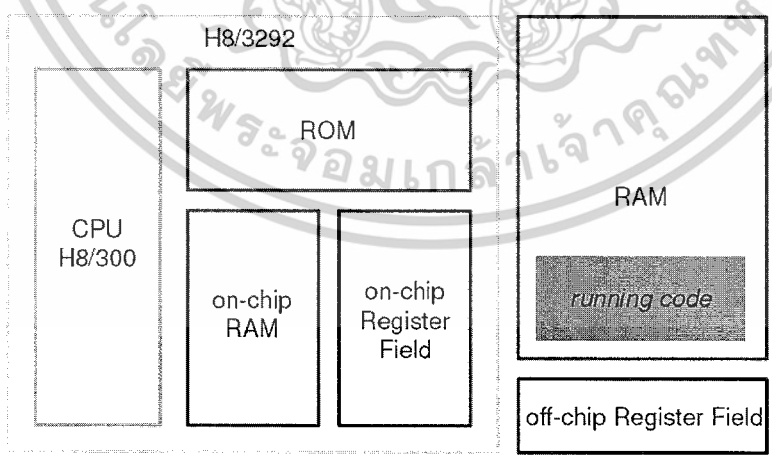
- เก็บที่ตำแหน่ง 0xfd80-0xfdbf
- ใช้พอยท์เตอร์ชี้ตำแหน่งที่จะทำการอินเตอร์รัพ

-Masking

- Globally (except NMI) CCR bit 7
- Individually through the off-chip register field

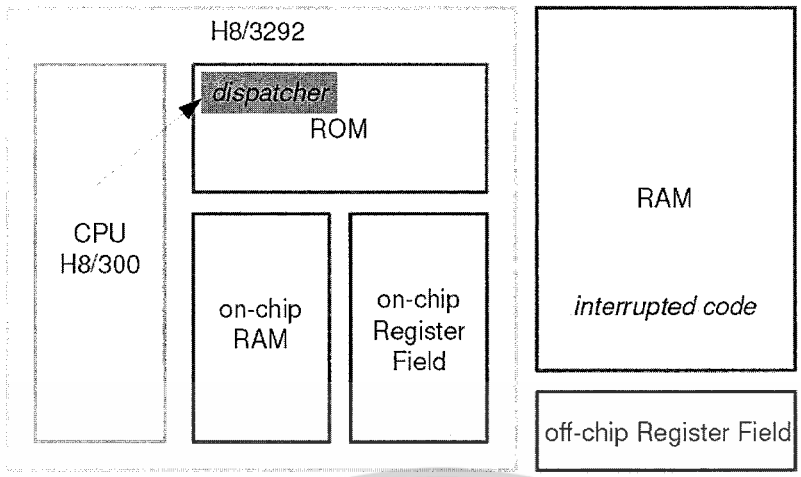
ตารางที่ ข.1 แสดงการอินเตอร์รัพท์เวกเตอร์ของ RCX

Vector	Source
0	reset
1 - 2	reserved
3	NMI
4 - 6	IRQs
7 - 11	reserved
12 - 18	16-bit timer
19 - 21	8-bit timer 0
22 - 24	8-bit timer 1
25 - 26	reserved
27 - 30	serial
31 - 34	reserved
35	ADI
36	WOVF

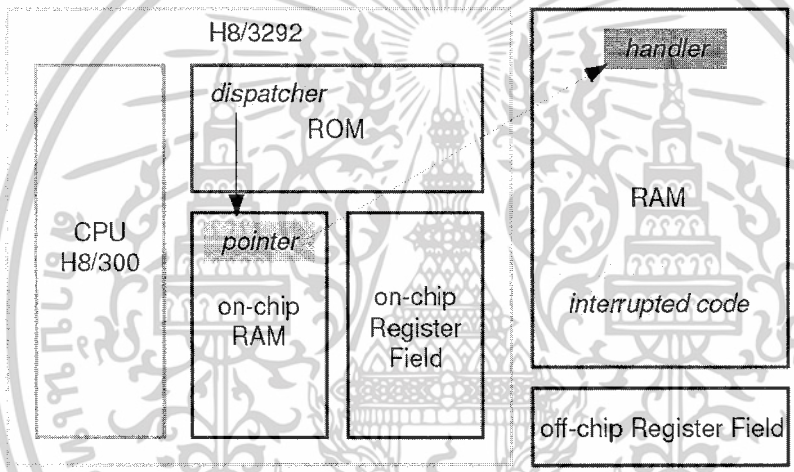


รูปที่ ข.5 แสดงสถานะที่กำลังทำงานของ RCX

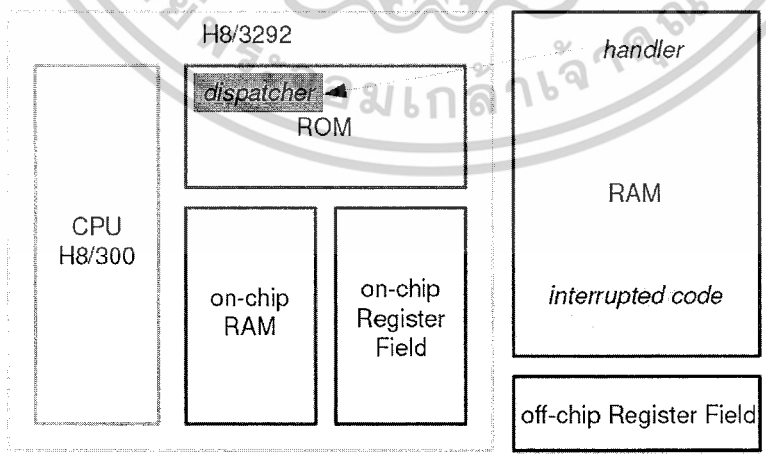
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 แสดงการอินเตอร์รัพท์แบบเร่งด่วนขั้นที่ 1 บน RCX

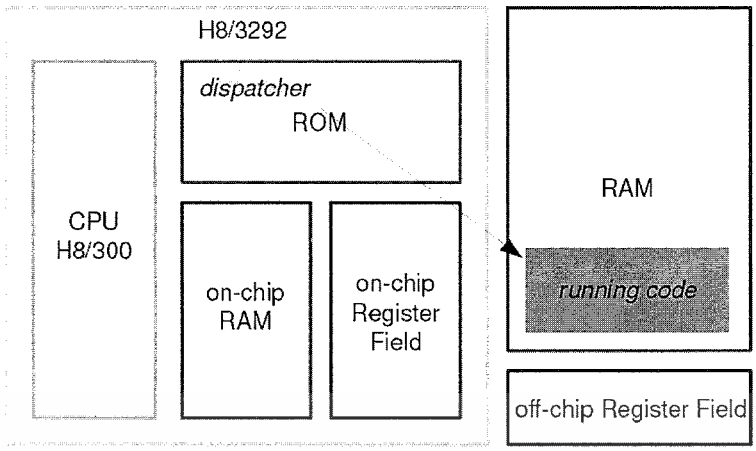


รูปที่ ข.7 แสดงการอินเตอร์รัพท์แบบเร่งด่วนขั้นที่ 2 บน RCX



รูปที่ ข.8 แสดงการอินเตอร์รัพท์แบบเร่งด่วนขั้นที่ 3 บน RCX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 แสดงการอินเตอร์รัพท์แบบเร่งด่วนขั้นที่ 4 บน RCX

อินพุตและเอาท์พุตของ Hitachi H8/3292

- พอร์ตอินพุตและเอาท์พุต 7 พอร์ต
  - พอร์ต 1 - พอร์ต 4, พอร์ต 6 : มี 8 บิต เป็นพอร์ตอินพุตและเอาท์พุต
  - พอร์ต 5 มี 3 บิต เป็นพอร์ตอินพุตและเอาท์พุต
  - พอร์ต 7 มี 8 บิต เป็นพอร์ตอินพุต

-Each comprised of up to three registers

- DDR (P1-P6): data direction (input/output)
- DR (P1-P7): data
- PCR (P1-P3): pull-up control

- โหมดที่ใช้ปฏิบัติการ 3 โหมด
  - โหมด I : I/O address space, on-chip ROM disabled
  - โหมด II : I/O address space, on-chip ROM enabled
  - โหมด III : อินพุตเอาท์พุตทั่วไป

ตารางที่ ข.2 แสดงอินพุตและเอาท์พุตของ Hitachi H8/3292

Port	Type	Mode I	Mode II	Mode III	DDR	DR	PCR
1	8-bit, I/O	16-bit I/O address (lsb)		GPIO	0xffb0	0xffb2	0xffac
2	8-bit, I/O	16-bit I/O address (msb)		GPIO	0xffb1	0xffb3	0xffad
3	8-bit, I/O	8-bit I/O data		GPIO	0xffb4	0xffb6	0xffae
4	8-bit, I/O	bus state (IRQ/WAIT/RD/WR/CLOCK/AD)		GPIO	0xffb5	0xffb7	–
5	3-bit, I/O	serial port			0xffb8	0xffba	–
6	8-bit, I/O	timer control			0xffb9	0xffbb	–
7	8-bit, I	A/D converter			–	0xffbe	–

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล
  - ควบคุมสถานะการเปลี่ยนแปลงเริ่มต้นของรีจิสเตอร์ที่หน้าจอ
  - สามารถเกิดการอินเทอร์รัพได้

ตารางที่ ข.3 แสดงตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล

ADCSR at 0xffe8

Bit	Name	Abbreviation	Function
7	A/D End Flag	ADF	1 = end of conversion(must be cleared)
6	A/D End Interrupt Enable	ADIE	0 = disabled; 1= enabled
5	A/D Start	ADST	0 = stop; 1 = start
4	Scan Mode	SCAN	0 = single; 1= scan
3	Clock Select	CKS	0 = slow; 1= fast
2 - 0	Channel Select	CH2 - CH0	000 CH0, 001 CH1, 010 CH2, 011 CH3

- ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล
  - 3 ช่องทาง ขนาด 10 บิต (ADDR[A|B|C)

ตารางที่ ข.4 แสดง 3 ช่องทาง ขนาด 10 บิต

ADDR

Channel	Data Reg.	Addr.
AN0	A	0xffe0
AN1	B	0xffe2
AN2	C	0xffe4

พอร์ตอินพุตและเอาท์พุตโหมด 2 ของ H8/3292

- ปุ่ม
  - ปุ่มที่ใช้ควบคุมการทำงานที่เชื่อมต่อกับพอร์ต 4 และ 7 ของ H8 มี 4 ปุ่ม
  - ปุ่ม Run และ OnOff ถูกกำหนดให้เป็นปุ่มอินเตอร์รัพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.5 แสดงพอร์ตอินพุตและเอาท์พุตโหมด 2 ของ H8/3292

RCX Button	H8 Port/Bit	H8 IRQ
Run	4/2	0
OnOff	4/1	1
View	7/6	-
Prgm	7/7	-

- เซนเซอร์

  - มี 3 พอร์ต คือ พอร์ต 1, 2, 3 ใช้เชื่อมต่อกับพอร์ต 7 โดยผ่านตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล
  - เซนเซอร์สัมผัสและเซนเซอร์วัดอุณหภูมิเป็นเซนเซอร์แบบ Passive คือไม่สามารถกำเนิดสัญญาณเองได้
  - เซนเซอร์แสงและเซนเซอร์จับการหมุนเป็นเซนเซอร์แบบ Active คือสามารถกำเนิดสัญญาณเองได้
  - เซนเซอร์สามารถเชื่อมต่อกับ RCX ได้โดยตรงที่พอร์ต 6

ตารางที่ ข.6 แสดงรายละเอียดอินพุตที่เป็นเซนเซอร์ของ RCX

RCX Sensor	H8 P7 bit (A/D ch)	H8 P6 bit (activate)
0	AN2	2
1	AN1	1
2	AN0	0

การควบคุมการทำงานของ RCX

- เมื่อมีการเปิดเครื่องหรือกดปุ่มรีเซตที่ RCX H8/300 จะอินเตอร์รัพเวคเตอร์อันดับหนึ่งและจะเริ่มทำงานตามคำสั่งที่ตำแหน่ง 0x0000 ซึ่งเป็นตำแหน่งแรกในการเริ่มต้นการทำงาน
- ขั้นตอนการทำงานใน RCX
  - ทำการอัพโหลดโปรแกรมผ่านทาง พอร์ตอินฟราเรด
  - สร้างสภาพแวดล้อมในการโปรแกรมการทำงานให้เหมาะกับการเชื่อมต่ออุปกรณ์เซนเซอร์และแอคทูเอเตอร์โดยใช้อินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน นางสาววัชรินทร์ จันทภา

วันเดือนปีเกิด 29 มีนาคม 2528

สถานที่เกิด หนองคาย

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนสตรีราชินูทิศ ปีการศึกษา 2544

ศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

ชื่อผู้เขียน นางสาวนลัทพร เมืองบัว

วันเดือนปีเกิด 1 กรกฎาคม 2528

สถานที่เกิด ชลบุรี

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนชลราษฎรอำรุง ปีการศึกษา 2544

ศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้