ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ระบบหุ่นยนต์จัดเก็บวัสดุ

STORE KEEPER ROBOT

โดย

สุวิชา จึงเจริญสุขยิ่ง

อานนท์ เจียสิริมงคล

H006062

อาจารย์ที่ปรึกษา

ผศ.ดร.โอพาร วงศ์วิรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2551

เลขหมู่.....06062.....
เลขทะเบียน.....
วัน,เดือน,ปี.....10 มี.ค. 2553.....
b.....
ใช้ประโยชน์ด้านการศึกษา

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STORE KEEPER ROBOT

SUWICHA JUENGJAREONSUKYING

ANON JEARSIRIMONGKOL

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

2/2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2009

FACULTY ON INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2551

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์จัดเก็บวัสดุ

STORE KEERPER ROBOT

ผู้จัดทำ

1. นายสุวิชา จึงเจริญสุขยิ่ง รหัสนักศึกษา 48070173
2. นายอานนท์ เจียสิริมงคล รหัสนักศึกษา 48070179

.....อาจารย์ที่ปรึกษา

(ผศ.ดร.โอพาร วงศ์วิรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์จัดเก็บวัสดุ
นักศึกษา	นายสุวิชา จึงเจริญสุขยิ่ง
	นายอานนท์ เจียสิริมงคล
รหัสนักศึกษา	48070173
	48070179
ปริญญา	วิทยาศาสตร์บัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	ผศ.ดร.โอฬาร วงศ์วิรัตน์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาและพัฒนาระบบฝังตัว "หุ่นยนต์จัดเก็บวัสดุ" โดยมีเป้าหมายเพื่อสร้างต้นแบบของหุ่นยนต์ในการช่วยอำนวยความสะดวกแก่ผู้ที่ต้องการจัดเก็บวัสดุหรือสิ่งของ ทำให้เกิดความถูกต้องแม่นยำในการจัดเก็บ โดยใช้ชุดทดลองเทคโนโลยีฝังตัว Lego MINDSTORMS รุ่น NXTในการจัดสร้างฮาร์ดแวร์ และใช้ NetBean ร่วมกับ LEJOS NXJ เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ ทฤษฎีที่ใช้ในการศึกษาและพัฒนาประกอบด้วยหลักการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object Oriented Analysis and Design) และอาศัย UML (Unified Modeling Language) เป็นเครื่องมือในการวิเคราะห์และออกแบบ ทำให้การวางแผนและดำเนินการพัฒนาเป็นไปอย่างถูกต้องและมีประสิทธิภาพ โดยหุ่นยนต์จัดเก็บวัสดุใช้เซ็นเซอร์แสงในการตรวจจับชนิดของวัสดุและทำการจำแนกชนิดของวัสดุดังกล่าว แล้วนำไปเปรียบเทียบกับชนิดของเป้าหมายในหน่วยความจำเพื่อที่จะนำวัสดุไปจัดเก็บในตำแหน่งที่ถูกต้อง นอกจากนี้หุ่นยนต์จัดเก็บวัสดุยังใช้เซ็นเซอร์แสงในการตรวจจับเส้นทางเพื่อเดินทางนำวัสดุไปจัดเก็บไว้ในตำแหน่งที่กำหนด หุ่นยนต์สามารถหยุดรอเพื่อรับคำสั่งให้ตรวจสอบหาจุดเป้าหมายที่ต้องการในเส้นทางเพื่อเดินทางไปรอรับวัสดุได้ รวมทั้งสามารถทำการจัดเรียงวัสดุในกรณีที่จุดรอรับมีหลายจุด โดยใช้การเปรียบเทียบจากข้อมูลวัสดุและเป้าหมายที่อยู่ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Store Keeper Robot
Student	Mr. Suwicha Juengjareonsukying
	Mr. Anon Jearsirimongkol
Student ID.	48070173
	48070179
Degree	Bachelor of Science
Programme	Information Technology
Academic Year	2008
Advisor	Asst.Prof.Dr. Olarn Wongwirat

ABSTRACT

This thesis is the study and development of embedded system "Store Keeper Robot." The aim of this thesis is to implement a prototype robot that correctly and accurately facilitates a user when storing the objects or items is required. The embedded technology, Lego MINDSTORMS NXT series, is employed for hardware establishment, and the LEJOS NXJ, in conjunction with NetBean programming environment, is used as a tool for software development. The theoretical background used in this thesis is an OOAD (Object-Oriented Analysis and Design) methodology and acquires a UML (Unified Modeling Language) as a tool in analysis and design process. This can provide an efficient and correct planning and development process. The store keeper robot employs a light sensor to detect and to classify the object. The result is used to compare with the target attribute in memory for storing the object in the right position. The store keeper robot also uses the light sensor to track the line to follow to the proper position as specified. The robot can wait for the commands along the line to verify the required target and to obtain the object. In addition, the robot can sort the object, when there are many store positions, by comparing the object and the target data that are stored in memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ผศ.ดร.โอพาร วงศ์วิรัตน์ อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำและข้อคิดเห็นต่างๆของการพัฒนาโครงงานมาโดยตลอด

นอกจากนี้ยังขอขอบคุณ ผศ.ดร. ธนารัตน์ ชลิดาพงศ์ อาจารย์ อนันตพัฒน์ อนันตชัย และอาจารย์ ณฐพล พันธุวงศ์ ที่กรุณาสละเวลา เป็นอาจารย์สอบโครงงาน พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์

ผู้พัฒนาโครงงานจึงขอกราบขอบพระคุณเป็นอย่างสูง มา ณ โอกาสนี้

สุวิชา จึงเจริญสุขยิ่ง

อานนท์ เจียสิริมงคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

กิตติกรรมประกาศ..... III

สารบัญ..... IV

สารบัญตาราง..... IX

สารบัญรูป..... X

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 1

1.3 หลักการและแนวคิด..... 1

1.4 ขอบเขตของโครงงาน..... 2

1.5 ขั้นตอนของการพัฒนา..... 2

1.6 ประโยชน์ที่คาดว่าจะได้รับ..... 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบฝังตัว (Embedded System)..... 3

2.1.1 องค์ประกอบทางด้านซอฟต์แวร์ (Elements of Embedded Software)..... 3

2.1.1.1 การประมวลผลแบบเรียลไทม์(Real-Time Processing)..... 3

2.1.1.2 ประเภทของซอฟต์แวร์(Software)..... 5

2.1.1.3 เรียลไทม์เคอร์เนล (Real –Time Kernel)..... 5

2.1.1.4 การอินเทอร์รัพต์เหตุการณ์ (Interrupt and Event)..... 5

2.1.1.5 มัลติโปรแกรมมิง (Multiprogramming)..... 6

2.1.1.6 วิธีการจัดการแทสค์ (Task Management Approach) 7

2.1.1.7 การเรียกระบบหรือซิสเต็มคอล (System Call) 8

2.1.1.8 การซิงโครไนเซชัน (Synchronization) 8

2.1.1.9 ดีไวซ์ไดร์เวอร์ (Device Driver) 9

2.1.1.10 มิดเดิลแวร์ (Middleware) 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.1.2 องค์ประกอบทางด้านฮาร์ดแวร์ (Elements of Embedded Hardware).... 9

2.1.2.1 เอ็มพียู (MPU) 10

2.1.2.2 หน่วยความจำ (Memory) 10

2.1.2.3 บัส(Bus) 10

2.1.2.4 เทคโนโลยีการอินเทอร์รัพต์ (Interrupt Technology) 11

2.1.2.5 เทคโนโลยีดีเอ็มเอ (DMA Technology) 12

2.1.2.6 เทคโนโลยีหน่วยความจำแคช
(Cache Memory Technology) 13

2.1.2.7 เทคโนโลยีหน่วยความจำเสมือน
(Virtual Memory Technology) 14

2.1.3 การติดต่อสื่อสารกับผู้ใช้ทั้งแบบง่ายและแบบซับซ้อน..... 14

2.1.4 แพลตฟอร์มและโครงสร้างทางสถาปัตยกรรมของหน่วย
ประมวลผลข้อมูล..... 15

2.1.5 อุปกรณ์ต่อพ่วงหรือช่องทางในการสื่อสารกับโมดูลอื่นๆ
(Peripherals) 15

2.1.6 เครื่องมือต่างๆในการพัฒนา(Tools) 15

2.1.7 Development Toolkit ที่ใช้ในการพัฒนา..... 16

2.2 วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC) 16

2.2.1 SDLC ในรูปแบบน้ำตก(Waterfall)..... 18

2.2.2 SDLC ในรูปแบบน้ำตกชนิดปรับปรุง(Adapted Waterfall)..... 18

2.2.3 SDLC ในรูปแบบวิวัฒนาการ(Evolutionary) 19

2.2.4 SDLC ในรูปแบบเพิ่มส่วน(Incremental) 19

2.2.5 SDLC ในรูปแบบวนรอบ(Spiral) 20

2.3 แนวทางปฏิบัติ (Methodologies) 20

2.3.1 Structured System Analysis and Design Methodology (SSADM) 21

2.3.2 Rapid Application Development-based Methodology (RAD) 21

2.3.3 Object-Oriented Analysis and Design Methodology (OOADM) 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4 ทฤษฎีแนวคิดของการพัฒนาระบบโดยใช้หลักการเชิงวัตถุ
(Object Oriented Analysis And Design) 22
2.4.1 การสืบทอดคุณสมบัติ(Inheritance) 23
2.4.2 ความเป็นโมดูล(Modularity) 24
2.5 โมเดลที่ใช้ในการพัฒนา - ยูเอ็มแอล โมเดล(UML Standard Model) 25
2.5.1 ยูสเคสไดอะแกรม(Use Case Diagram) 27
2.5.2 แอกทิวิตี้ไดอะแกรม(Activity Diagram) 28
2.5.3 คลาสไดอะแกรม(Class Diagram) 29
2.5.4 ซีเควนซ์ไดอะแกรม (Sequence Diagram) 30
2.6 เทคโนโลยีที่ใช้ในการพัฒนา 31
2.6.1 เทคโนโลยีทางด้านฮาร์ดแวร์ (Hardware Technology) 31
2.6.2 เทคโนโลยีทางด้านซอฟต์แวร์ (Software Technology) 35
2.6.3 เทคโนโลยีในการออกแบบฮาร์ดแวร์(Hardware Design) 37

บทที่ 3 การวิเคราะห์และออกแบบหุ่นยนต์จัดเก็บวัสดุ

3.1 การพัฒนาในส่วนของซอฟต์แวร์ (Software Development) 39
3.1.1 การวิเคราะห์ความต้องการของระบบ (Requirement Analysis) 39
3.1.1.1 หน้าที่ของระบบ 39
3.1.1.2 ส่วนประกอบเพิ่มเติม 40
3.1.2 การออกแบบซอฟต์แวร์ (Software Design) 40
3.1.2.1 แผนภาพขอบเขตของระบบ (UseCase Diagram) 40
3.1.2.2 แผนภาพกิจกรรม (Activity Diagram) 53
3.1.2.3 แผนภาพลำดับการทำงาน (Sequence Diagram) 63
3.1.2.4 แผนภาพคลาส (Class Diagram) 69
3.1.2.5 แผนภาพสถานะ (State Chart Diagram) 70
3.2 การออกแบบในส่วนของฮาร์ดแวร์ (Hardware Design) 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 การทดสอบการทำงาน

4.1 การทดสอบการทำงานของฮาร์ดแวร์.....76

4.1.1 การทดสอบมอเตอร์ (Motor Testing)..... 76

4.1.2 การทดสอบเซ็นเซอร์แสงและหน้าจอLCD

(Light Sensor&LCD Testing) 76

4.1.3 การทดสอบเซ็นเซอร์สัมผัส(Touch Sensor Testing) 76

4.1.4 การทดสอบการวัดค่าความสว่างของสี(Color Calibration) 76

4.2 การทดสอบการทำงานตามสภาพการณ์(Scenario)..... 77

4.2.1 การตรวจจับและเคลื่อนที่ตามเส้นทาง (Line Following) 77

4.2.2 การตรวจสอบการวางวัสดุ (Deploying) 78

4.2.3 การจัดเรียงวัสดุในกรณีที่จุดเก็บมีหลายส่วน (Sorting) 79

4.2.4 การรอรับคำสั่งเพื่อเดินทางหาจุดที่ต้องการจากผู้ใช้ (Order-Waiting) ... 80

บทที่ 5 สรุปผลการทดสอบและข้อเสนอแนะ

5.1 สรุปผลการทดสอบ..... 82

5.2 ประโยชน์ที่ได้รับจากการสร้างหุ่นยนต์จัดเก็บวัสดุ.....82

5.3 ปัญหาและอุปสรรค..... 82

5.4 ข้อเสนอแนะ.....83

บรรณานุกรม.....84

ภาคผนวก

โปรแกรมที่ใช้ควบคุมการทำงานของหุ่นยนต์

1. การทดสอบมอเตอร์..... 86

2. การทดสอบเซ็นเซอร์ตรวจจับความสว่างและหน้าจอ LCD..... 87

3. การหาค่าช่วงของความสว่างสี.....89

4. การทดสอบการติดตามเส้นทาง..... 93

5. การทดสอบการวางวัสดุ.....96

6. การเดินทางตามหาเส้นทางและหยุดวางของเมื่อเจอวัสดุ.....98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

7. การทดสอบการรอรับคำสั่งจากผู้ใช้..... 105

ประวัติผู้เขียน..... 112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	การขอร้องให้ตอบสนองภายในเวลาที่กำหนด	4
2.2	วิธีการพรีเอมชันตามเหตุการณ์	5
2.3	การหยุดประมวลผลตามลำดับอันเนื่องมาจากการอินเทอร์รัพต์	6
2.4	การทำพรีเอมชัน และการสลับคอนเท็กซ์	7
2.5	ตัวอย่างของบล็อกควบคุมแทสค์(TCB)	7
2.6	ฟังก์ชันของแฟลกเหตุการณ์	8
2.7	สถาปัตยกรรมพื้นฐานของระบบฝังตัว	9
2.8	ขั้นตอนการประมวลผลของหน่วยประมวลผลไมโครคอนโทรลเลอร์	10
2.9	เทคโนโลยีการอินเทอร์รัพต์	12
2.10	การเคลื่อนย้ายข้อมูลโดยวิธี DMAเทียบกับแบบธรรมดาที่ไม่มี DMA	13
2.11	การทำงานของหน่วยความจำแคช	13
2.12	ระบบการทำงานของหน่วยความจำเสมือน	14
2.13	ตัวอย่าง USB คอนเนคเตอร์ที่ใช้มาตรฐาน USBIF	15
2.14	แสดงการทำงานของคอมไพเลอร์ในภาษาจาวา	16
2.15	วงจรการพัฒนาระบบแบบ Waterfall	18
2.16	วงจรการพัฒนาระบบแบบ Adapted Waterfall	18
2.17	วงจรการพัฒนาระบบแบบ Evolutionary	19
2.18	วงจรการพัฒนาระบบแบบ Incremental	19
2.19	วงจรการพัฒนาระบบแบบ Spiral	20
2.20	อธิบายตัวอย่างการสืบทอดคุณสมบัติ	23
2.21	แสดงแผนภาพ ยูสเคสไดอะแกรมของระบบลงทะเบียน	27
2.22	ตัวอย่างแอกทิวิตี้ไดอะแกรม การลงทะเบียน	28
2.23	ตัวอย่างคลาสไดอะแกรมในระบบการซื้อของออนไลน์	29
2.24	แสดงตัวอย่างของซีเควนซ์ไดอะแกรมการทำงานของเว็บเพจ	30
2.25	หน่วยประมวลผลของ LEGO MINDSTORMS NXT	31
2.26	เซ็นเซอร์สัมผัส	33
2.27	เซ็นเซอร์เสียง	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
2.28	เซ็นเซอร์แสง	34
2.29	เซ็นเซอร์อัลตราโซนิคส์	34
2.30	เซอร์โวมอเตอร์	34
2.31	แสดงการทำงานของอินเตอร์พรีตเตอร์ในการใช้ไลบรารี	36
2.32	ตัวอย่างของ API ของ ไลบรารี LEJOS	37
2.33	ตัวอย่างโปรแกรม LDD(LEGO Digital Designer)	37
2.34	ตัวอย่างโปรแกรม LDD(2)(LEGO Digital Desginer)	38
3.1	แสดงแผนภาพยูสเคสของระบบหุ่นยนต์จัดเก็บวัสดุ	41
3.2	แสดงแผนภาพกิจกรรม AutoMode ของระบบหุ่นยนต์จัดเก็บวัสดุ	53
3.3	แสดงแผนภาพกิจกรรม WaitObject ของระบบหุ่นยนต์จัดเก็บวัสดุ	54
3.4	แสดงแผนภาพกิจกรรม Deploy ของระบบหุ่นยนต์จัดเก็บวัสดุ	55
3.5	แสดงแผนภาพกิจกรรม TrackMove ของระบบหุ่นยนต์จัดเก็บวัสดุ	56
3.6	แสดงแผนภาพกิจกรรม Bumper ของระบบหุ่นยนต์จัดเก็บวัสดุ	57
3.7	แสดงแผนภาพกิจกรรม ObjectScanner ของระบบหุ่นยนต์จัดเก็บวัสดุ	58
3.8	แสดงแผนภาพกิจกรรม TargetScanner ของระบบหุ่นยนต์จัดเก็บวัสดุ	58
3.9	แสดงแผนภาพกิจกรรม LineScanner ของระบบหุ่นยนต์จัดเก็บวัสดุ	59
3.10	แสดงแผนภาพกิจกรรม ManualMode ของระบบหุ่นยนต์จัดเก็บวัสดุ	60
3.11	แสดงแผนภาพกิจกรรม WaitOrder ของระบบหุ่นยนต์จัดเก็บวัสดุ	61
3.12	แสดงแผนภาพกิจกรรม Searcher ของระบบหุ่นยนต์จัดเก็บวัสดุ	62
3.13	แสดงแผนภาพขั้นตอนการทำงาน AutoMode ของระบบหุ่นยนต์จัดเก็บวัสดุ	64
3.14	แสดงแผนภาพขั้นตอนการทำงาน WaitObject ของระบบหุ่นยนต์จัดเก็บวัสดุ	65
3.15	แสดงแผนภาพขั้นตอนการทำงาน Deploy ของระบบหุ่นยนต์จัดเก็บวัสดุ	66
3.16	แสดงแผนภาพขั้นตอนการทำงาน TrackMove ของระบบหุ่นยนต์จัดเก็บวัสดุ	67
3.17	แสดงแผนภาพขั้นตอนการทำงาน ManualMode ของระบบหุ่นยนต์จัดเก็บวัสดุ	68
3.18	แสดงโครงสร้างของคลาสไดอะแกรมของระบบหุ่นยนต์จัดเก็บวัสดุ	69
3.19	แสดงแผนภาพสถานะ AutoModeของระบบหุ่นยนต์จัดเก็บวัสดุ	70
3.20	แสดงแผนภาพสถานะ ManualModeของระบบหุ่นยนต์จัดเก็บวัสดุ	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.21	เซ็นเซอร์ตรวจจับวัสดุ (ObjectScanner)	73
3.22	เซ็นเซอร์ตรวจจับเป้าหมาย (TargetScanner)	73
3.23	เซ็นเซอร์ตรวจจับเส้นทาง (LineTracker)	73
3.24	รูปเซ็นเซอร์ตรวจจับการชน	74
3.25	มอเตอร์ที่ใช้ในการเคลื่อนที่ (Mover)	74
3.26	มอเตอร์ที่ใช้ในการวางวัตถุ (Deployer)	74
3.27	การประกอบโมดูลทั้งหมดเข้าด้วยกัน	75
4.1	แสดงการทำงานของการเดินตามเส้นทางและเคลื่อนที่ (Track Line)	77
4.2	แสดงการวางวัสดุ (Deployer)	78
4.3	การจัดเรียงวัสดุในกรณีที่จุดเก็บมีหลายส่วน(Sorting)	79
4.4	การรอรับคำสั่งเพื่อเดินทางหาจุดที่ต้องการจากผู้ใช้(Order-Waiting)	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 เปรียบเทียบการวิเคราะห์และออกแบบระบบวิธีเดิมกับวิธีเชิงวัตถุ.....23

3.1 Use Case AutoMode Description 42

3.2 Use Case WaitObject Description 43

3.3 Use Case Deploy Description.....44

3.4 Use Case TrackMove Description.....45

3.5 Use Case Bumper Description.....46

3.6 Use Case ObjectScanner Description.....47

3.7 Use Case TargetScanner Description.....48

3.8 Use Case LineScanner Description.....49

3.9 Use Case ManualMode Description.....50

3.10 Use Case WaitOrder Description.....51

3.11 Use Case Searcher Description.....52

3.12 การออกแบบฮาร์ดแวร์ตามวัตถุประสงค์.....72

4.1 ค่าการทดสอบการวัดความสว่างของสี (Color Calibration Value) 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน การทำงานภายในโรงงานนั้น เครื่องจักรเป็นปัจจัยสำคัญที่ถูกเลือกใช้ให้ทำหน้าที่หลายอย่างแทนมนุษย์ แนวคิดในการสร้าง "หุ่นยนต์จัดเก็บวัสดุ" ได้แรงบันดาลใจมาจากการรับส่งวัสดุจากที่หนึ่งไปยังอีกที่ ภายใต้สภาพแวดล้อมการทำงานที่เป็นกิจวัตร หากใช้เครื่องจักรมาทำหน้าที่แทนมนุษย์ ทำให้เก็บประโยชน์ในด้านต่างๆมากมายเช่น ลดค่าใช้จ่ายในการจ้างแรงงาน ความสม่ำเสมอของงานที่มากขึ้น รวมทั้งประสิทธิภาพของงานที่ได้มาตรฐาน โครงงานพัฒนาหุ่นยนต์จัดเก็บวัสดุ จึงถูกคิดขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว

1.2 วัตถุประสงค์ของโครงงาน

ในโครงงานนี้มีวัตถุประสงค์เพื่อศึกษา "ระบบฝังตัวหุ่นยนต์จัดเก็บวัสดุ (Store Keeper Robot)" และประยุกต์แนวทางในการพัฒนาหุ่นยนต์ โดยใช้หลักการพัฒนาระบบเชิงวัตถุ เข้ากับระบบฝังตัว ในการจัดสร้างหุ่นยนต์จัดเก็บวัสดุ และเป็นต้นแบบในการสร้างหุ่นยนต์ที่ใช้ในงานที่ต้องการใช้การแยกแยะประเภทของวัสดุและจัดเก็บในที่ที่กำหนด

1.3 หลักการและแนวคิด

หุ่นยนต์จัดเก็บวัสดุ มีหลักการในการทำงานในเบื้องต้นคือ เมื่อเราทำการส่งวัสดุ หรือสิ่งของให้กับหุ่นยนต์ หุ่นยนต์ก็จะทำการวิเคราะห์และแยกแยะประเภทของวัสดุ ดังกล่าว และทำการนำวัสดุ ที่ได้ ไปจัดเก็บไว้ในที่ที่เหมาะสมตามที่ได้กำหนดไว้

การทำงานของหุ่นยนต์จัดเก็บวัสดุ จะใช้เซ็นเซอร์ (sensors) ต่าง ๆ เช่น เซ็นเซอร์สัมผัส เซ็นเซอร์แสง เซ็นเซอร์เสียง และเซ็นเซอร์คลื่นอัลตราโซนิกส์ (ultrasonics) เป็นต้น ที่ติดอยู่กับตัวหุ่นยนต์ในการอ่านค่าประเภทหรือชนิดของวัสดุ แล้วส่งให้กับหน่วยประมวลผลทำการตรวจวิเคราะห์ประเภทหรือชนิดของวัสดุ ดังกล่าว โดยใช้โปรแกรมที่พัฒนาขึ้น เพื่อหาว่าวัสดุ ดังกล่าวตรงกับตำแหน่ง หรือประเภทของที่เก็บส่วนใด (เช่น กำหนดให้เป็นสีเดียวกับวัสดุ เป็นต้น) จากนั้นหุ่นยนต์ก็จะนำวัสดุ นั้นไปจัดเก็บไว้ในที่ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงงาน

โครงงานฉบับนี้ได้นำเสนอการพัฒนาหุ่นยนต์จัดเก็บวัสดุให้สามารถนำวัสดุที่กำหนดไปเก็บในตำแหน่งที่ต้องการ สามารถหยุดเพื่อรอรับวัสดุชิ้นใหม่ในตำแหน่งนั้น และจัดเรียงวัสดุในกรณีที่จุดเก็บของมีหลายส่วน รวมถึงยังสามารถเลือกวัสดุที่ต้องการให้หุ่นยนต์เดินหาเพื่อรอรับวัสดุได้ โดยใช้ Lego MINDSTORMS NXT เป็นเครื่องมือในการพัฒนาด้านฮาร์ดแวร์ และ LEJOS NXJ เป็นอุปกรณ์ซึ่งใช้ภาษาจาวาเป็นพื้นฐานในการพัฒนาซอฟต์แวร์

1.5 ขั้นตอนของการพัฒนา

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของโครงงาน วัตถุประสงค์ของโครงงาน หลักการและแนวคิด ขอบเขตของโครงงาน ขั้นตอนการศึกษา และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้อง ด้านฮาร์ดแวร์ ซอฟต์แวร์ SDLC หลักการออกแบบระบบเชิงวัตถุ UML เทคโนโลยีด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้

บทที่ 3 กล่าวถึงการวิเคราะห์และออกแบบหุ่นยนต์จัดเก็บวัสดุ การออกแบบซอฟต์แวร์และฮาร์ดแวร์

บทที่ 4 กล่าวถึงการทดสอบการทำงานตามวัตถุประสงค์ที่ได้กำหนดไว้

บทที่ 5 กล่าวถึงการสรุปผลและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เพื่อเรียนรู้การประยุกต์แนวทางในการพัฒนาหุ่นยนต์ โดยใช้หลักการพัฒนาระบบเชิงวัตถุเข้ากับระบบฝังตัว ในการจัดสร้างหุ่นยนต์จัดเก็บวัสดุเพื่อเป็นต้นแบบในการสร้างหุ่นยนต์ที่ใช้ในงานที่ต้องการใช้การแยกแยะประเภทของวัสดุและจัดเก็บในที่ที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบฝังตัว (Embedded System)

ระบบฝังตัวคือระบบที่มีไมโครชิป(Microchip) กับโปรแกรมเพื่อควบคุมไม่โครชิปที่ฝังตัวอยู่ หรือมีระบบไอที(IT) ฝังอยู่ ซึ่งเปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กที่ทำงานได้เฉพาะด้านเฉพาะฟังก์ชันเท่านั้น และมีการตอบสนองเป็นเรียลไทม์ (Real Time) คือสามารถตอบสนองได้ในช่วงเวลาที่จำกัด แต่ก็ขึ้นอยู่กับงานหรือฟังก์ชันที่ทำว่าให้ความสำคัญกับการตอบสนองมากน้อยเท่าใด ซึ่งการตอบสนองแบบเรียลไทม์มี 2 แบบ คือ ฮาร์ดเรียลไทม์ (Hard – Real Time) กับ ซอฟต์เรียลไทม์ (Soft – Real Time) โดยฮาร์ดเรียลไทม์ คือการตอบสนองที่เคร่งครัดต่อเวลา ไม่มีความยืดหยุ่น ส่วนซอฟต์เรียลไทม์ คือการตอบสนองที่ไม่ได้มีความเคร่งครัดต่อเวลาในการตอบสนองมากเกินไป อาจมีความยืดหยุ่นได้แล้วแต่ความเหมาะสม โดยระบบฝังตัวนั้นมีองค์ประกอบหลักๆ 2 องค์ประกอบคือองค์ประกอบทางด้านฮาร์ดแวร์และองค์ประกอบทางด้านซอฟต์แวร์ (Japan System House Association (JASA). 2549. **เทคโนโลยีสมองกลฝังตัว Embedded Technology**. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง กรุงเทพฯ: สำนักพิมพ์ ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).)

2.1.1 องค์ประกอบทางด้านซอฟต์แวร์ (Elements of Embedded Software)

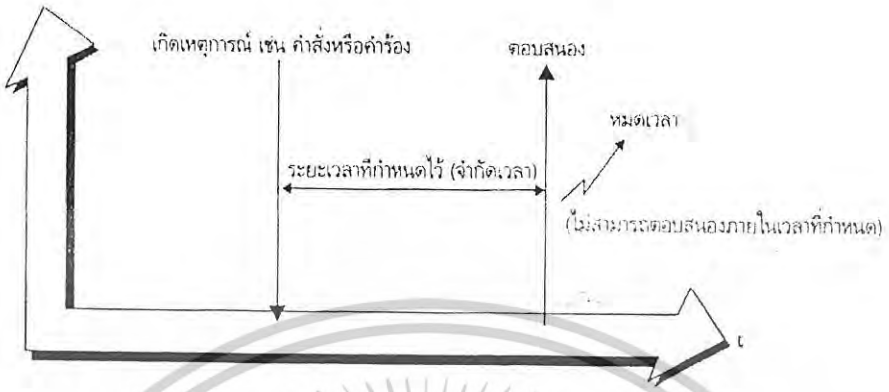
จุดประสงค์ของซอฟต์แวร์ระบบฝังตัวคือ การเพิ่มขีดความสามารถหรือฟังก์ชันการทำงานของระบบ โดยระบบที่มีไมโครชิป อยู่นั้นส่วนใหญ่จะเป็นระบบที่สามารถตอบสนองได้ทันที หรือที่เรียกว่า ระบบเรียลไทม์ (Real-Time System) ซึ่งจะรับสัญญาณจากภายนอกและจะตอบสนองภายในเวลาที่กำหนด

2.1.1.1 การประมวลผลแบบเรียลไทม์(Real-Time Processing)

โดยทั่วไประบบฝังตัวจะรับคำร้องขอประมวลผลต่างๆจากภายนอกระบบเช่น การเริ่มทำงานของอุปกรณ์ตามค่าที่ได้จากเซ็นเซอร์ (Sensor) ซึ่งส่วนใหญ่ต้องการความเป็นเรียลไทม์ หมายความว่าการประมวลผลต้องเสร็จสิ้นตามเวลาที่กำหนด และมีการตอบสนองอะไรบางอย่างกลับไป

การได้รับคำสั่งจากภายนอกโดยต้องการการตอบสนองจากระบบ เรียกสภาวะนั้นว่า เหตุการณ์ (Event) ดังรูปที่ (2.1)

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



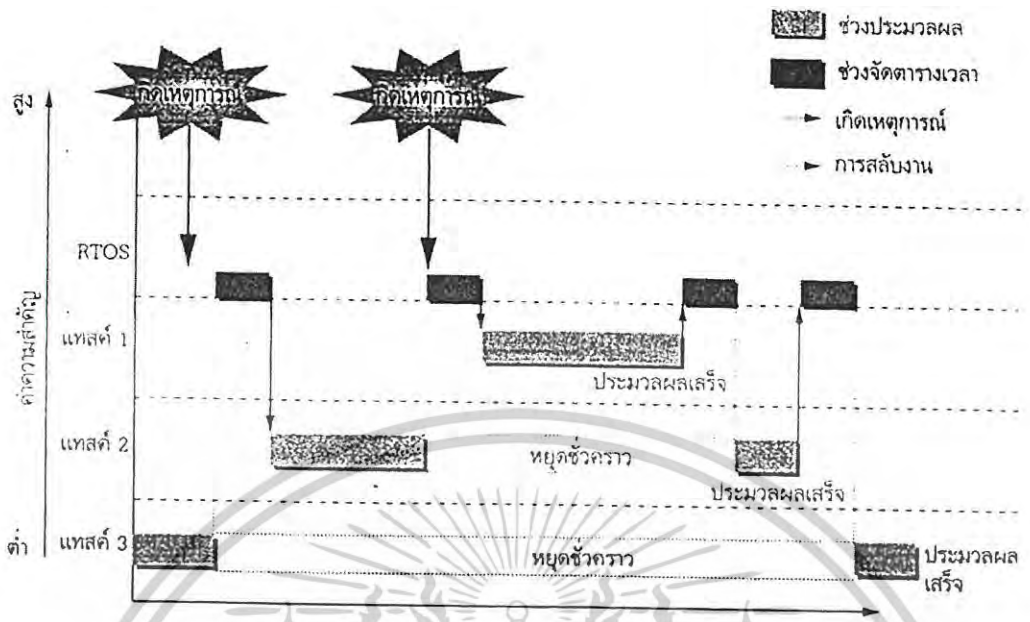
รูปที่ 2.1 การขอร้องให้ตอบสนองภายในเวลาที่กำหนด (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 16.)

ในกรณีที่เกิดเหตุการณ์หลายๆเหตุการณ์ขึ้นพร้อมกันระบบจึงต้องมีการจัดตารางงาน(Task Scheduling) ซึ่งเป็นตัวชี้วัดว่าโพรเซสใดจะถูกประมวลผลก่อน ดังนั้นจุดประสงค์หลักของระบบเรียลไทม์คือ การตอบสนองเหตุการณ์หลายๆเหตุการณ์ภายในเวลาที่กำหนด โดยสิ่งที่ให้กรอบการจัดตารางเวลางาน (Scheduling Framework) เพื่อทำให้เกิดระบบเรียลไทม์ ก็คือ ระบบปฏิบัติการเรียลไทม์ (Real-Time OS) เรียกโดยย่อว่า RTOS

โดยระบบปฏิบัติการเรียลไทม์จะให้ค่าความสำคัญหรือไพออริตีแก่การประมวลผล งานที่มีค่าความสำคัญน้อยก็จะถูกหยุดไว้ก่อน และระบบจะไปทำงานที่มีค่าความสำคัญมากกว่า ทำให้ระบบสามารถตอบสนองต่องานที่เร่งด่วนได้

การจัดตารางเวลางานแบบนี้เรียกว่า การพรีเอมชันตามเหตุการณ์(Event-Driven Preemption) (ดังรูปที่ 2.2) โดยที่ค่าความสำคัญของงานนั้นจะกำหนดตามความเร่งด่วนของงาน โดยจะห้ามไม่ให้มีการเปลี่ยนแปลงลำดับความสำคัญโดยเคอร์เนลได้ รูปแบบที่ไม่ยอมให้มีการเปลี่ยนแปลงค่าความสำคัญโดยเคอร์เนลได้นั้น เรียกว่า วิธีการจัดตารางเวลาแบบค่าความสำคัญตายตัว (Fixed Priority Scheduling Method)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 วิธีการพรีเอมชันตามเหตุการณ์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 17.)

2.1.1.2 ประเภทของซอฟต์แวร์(Software)

ประเภทโดยสังเขปของซอฟต์แวร์ระบบฝังตัวโดยสามารถแบ่งออกได้เป็นสองประเภทใหญ่ๆคือ ระบบปฏิบัติการ(Operating System) และ โปรแกรมประยุกต์ (Application Program) โดยระบบปฏิบัติการนั้นจะเป็นซอฟต์แวร์ที่มีหน้าที่ให้กรอบการทำงานแก่ซอฟต์แวร์อื่นๆ ส่วนโปรแกรมประยุกต์นั้นมีหน้าที่ใช้กรอบงานดังกล่าว ที่ระบบปฏิบัติการเสนอให้ เพื่อใช้ประมวลผลเพื่อแก้ปัญหาตามความต้องการ

2.1.1.3 เรียลไทม์เคอร์เนล (Real –Time Kernel)

เป็นส่วนสนับสนุนฟังก์ชันพื้นฐานที่จะทำให้เกิดการประมวลผลแบบเรียลไทม์ ซึ่งจะทำหน้าที่ในการควบคุมจัดการ MPU หรือหน่วยประมวลผล เพื่อให้ง่ายต่อการสร้างระบบเรียลไทม์

2.1.1.4 การอินเทอร์รัพต์เหตุการณ์ (Interrupt and Event)

การอินเทอร์รัพต์เป็นวิธีการเข้าแทรกแซงเป็นฟังก์ชันที่ฮาร์ดแวร์ต้องสนับสนุน ซึ่งมีความสำคัญและจำเป็นมากในการสร้างระบบเรียลไทม์ โดยทั่วไปการทำงานของอุปกรณ์รอบข้างของคอมพิวเตอร์ (Computer Peripherals) เช่นอุปกรณ์รับส่งข้อมูล เป็นต้น จะเป็นอิสระ กล่าวคือ MPU จะถูกเรียกประมวลผลเมื่อใดก็ได้ ดังนั้นเพื่อให้มีความพร้อมในการตอบสนองเหตุการณ์ที่อาจจะเกิดขึ้นจะต้องมีการอินเทอร์รัพต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

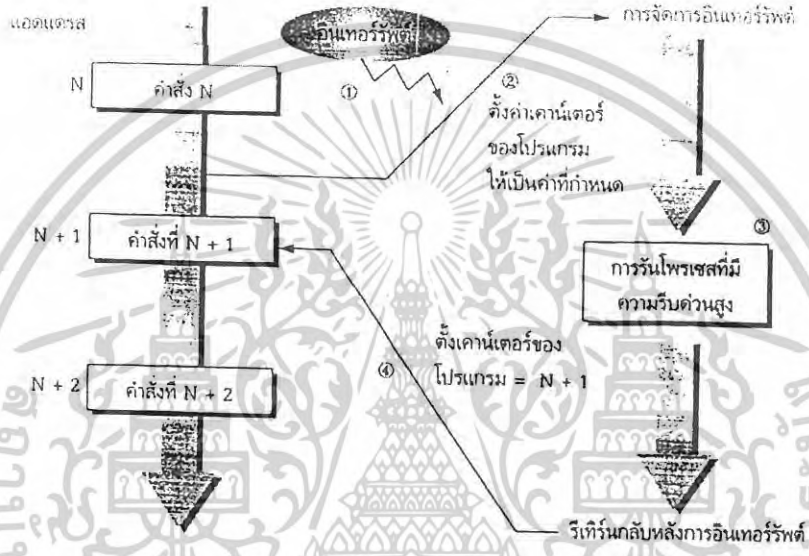
กรอบการทำงานของอินเทอร์รัพต์ มีดังต่อไปนี้ (ดังรูปที่ 2.3)

1. รับสัญญาณจากอุปกรณ์รอบข้าง

2.เมื่อถึงเวลาตามที่ตัวนับหรือเค้าน์เตอร์(Counter)ของโปรแกรมตั้งไว้ก็จะหยุดการทำงานของโปรแกรมที่รันอยู่ชั่วคราว

3. บังคับให้โปรแกรมที่ตอบสนองต่อสัญญาณที่รับจากอุปกรณ์นั้นๆ

4. กลับไปทำงานเดิมที่ถูกหยุดไว้ชั่วคราว

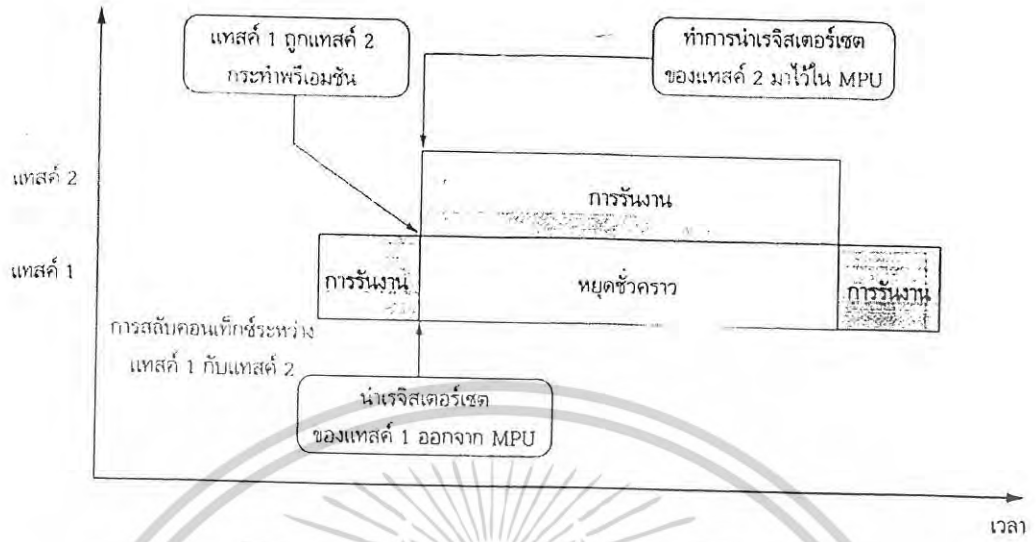


รูปที่ 2.3 การหยุดประมวลผลตามลำดับอันเนื่องมาจากการอินเทอร์รัพต์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 24.)

2.1.1.5 มัลติโปรแกรมมิง (Multiprogramming)

มัลติโปรแกรมมิง คือการที่ระบบฝังตัวรันโปรแกรมหลายๆโปรแกรมในเวลาเดียวกัน โดยการสลับไปมาระหว่างโปรแกรมตามตัวเหนี่ยวไก (Trigger) บางอย่างภายใต้สิ่งแวดล้อมของมัลติโปรแกรมมิง หน่วยของโปรแกรมที่วิ่งสลับไปมานั้นเรียกว่า แทสค์(Task) มีบางแทสค์ ที่หยุดทำงานเพราะแทสค์ตัวอื่น ปรากฏการณ์นี้เรียกว่า พรีเอมชัน (Preemption) ภายใต้สิ่งแวดล้อมมัลติโปรแกรมมิงนั้น แทสค์สามารถถูกพรีเอมชันขณะที่ทำงานอยู่ได้ โดยทั่วไปการไหลของการประมวลผล เรียกว่า คอนเท็กซ์หรือบริบท (Context) การสลับไปมาของคอนเท็กซ์ เรียกว่า "การสวิตซ์ระหว่างคอนเท็กซ์ (Context Switching)" (ดังรูปที่ 2.4) แสดงการนำรีจิสเตอร์ของ MPU ที่แทสค์ใช้อยู่ระหว่างการทำงาน หรือทำเรียกว่าคอนเท็กซ์ มาเก็บลงในสแตก (Stack) เพื่อหยุดการทำงานชั่วคราว และนำคอนเท็กซ์กลับมาจากสแตกเพื่อทำงานต่อ

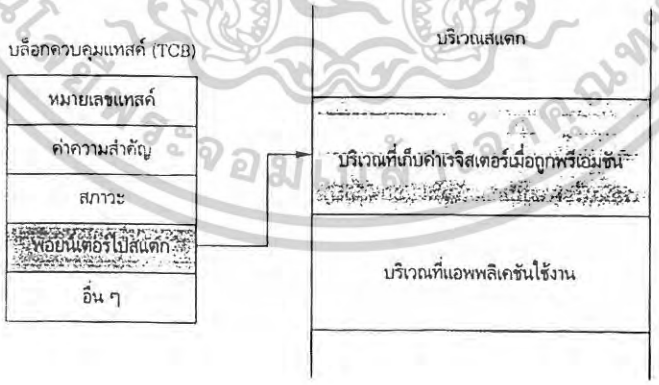
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การทำพรีเอมชัน และการสลับคอนเท็กซ์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 29.)

2.1.1.6 วิธีการจัดการแทสค์ (Task Management Approach)

เคอร์เนลจะจัดการแทสค์แต่ละตัวโดยโครงสร้างข้อมูลที่รียกว่า บล็อกควบคุมแทสก์ (Task Control Block:TCB) โดย TCB จะประกอบด้วยหมายเลขแทสค์ ค่าความสำคัญ สภาวะของแทสค์ (ดังรูปที่ 2.5)



รูปที่ 2.5 ตัวอย่างของบล็อกควบคุมแทสค์(TCB) (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 35.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.7 การเรียกระบบหรือซิสเต็มคอล (System Call)

การเรียกระบบหรือซิสเต็มคอลก็คือ เซอร์วิสที่ให้กับเคอร์เนลหรือตัวจัดการอินเทอร์รัพต์ เพื่อสร้างระบบเรียลไทม์ ซิสเต็มคอลจะให้เซอร์วิสต่างๆ เช่น การควบคุมแทสค์ การสื่อสารระหว่างแทสค์ การซิงโครไนเซชัน การควบคุมการแยกจากกัน เป็นต้น

2.1.1.8 การซิงโครไนเซชัน (Synchronization)

ในสิ่งแวดล้อมมักจะเกิดการประมวลผลที่ต้องรอสัญญาณซึ่งกันและกันระหว่างแทสค์ เรียกว่าการ ซิงโครไนเซชัน (Synchronization) วิธีการทั่วไปที่เคอร์เนลทำซิงโครไนเซชัน คือใช้แฟลกเหตุการณ์ (Event Flag) (ดังรูปที่ 2.6) แฟลกเหตุการณ์จะทำให้เกิดฟังก์ชันควบคุมแทสค์ได้โดยใช้สภาวะของแฟลก(Flag Status) เป็นตัวกำหนด สภาวะของแฟลกมีได้สองค่าคือ ON และ OFF เคอร์เนลจะมีซิสเต็มคอลในการเซตค่าแฟลกเหตุการณ์ให้เป็น ON และทำการเคลียร์ค่าแฟลกเหตุการณ์ให้เป็น OFF โดยซิสเต็มคอลนี้จะกำหนดให้แทสค์อยู่ในสภาวะรอจนกว่าแฟลกจะถูกตั้งให้เป็น ON แทสค์ที่รอเหตุการณ์บางอย่างให้เกิดขึ้นจะทำการเคลียร์ค่าแฟลกเหตุการณ์เป็น OFF ก่อนจะเปลี่ยนไปอยู่ในสภาวะรอ จนกว่าแฟลกเหตุการณ์จะถูกเซตให้เป็น ON อีกครั้ง

รูปที่ 2.6 ฟังก์ชันของแฟลกเหตุการณ์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 49.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.9 ดีไวซ์ไดร์เวอร์ (Device Driver)

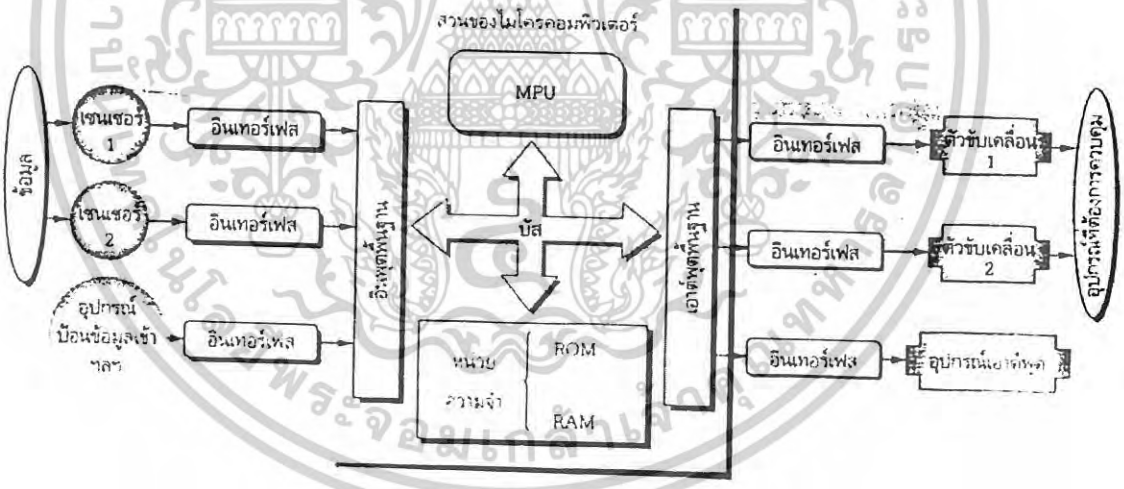
ดีไวซ์ไดรเวอร์ เป็นโมดูลที่ทำให้เกิดกสรควบคุมอุปกรณ์อินพุตและเอาท์พุต มีหน้าที่คือ ซ่อนรายละเอียดของฮาร์ดแวร์ออกจากแอพพลิเคชัน ทำให้แอพพลิเคชันเป็นอิสระจากการประมวลผลเพื่อควบคุมอุปกรณ์ และส่งผลกระทบน้อยที่สุดเมื่อมีการเปลี่ยนแปลงอุปกรณ์

2.1.1.10 มิดเดิลแวร์ (Middleware)

มิดเดิลแวร์ เป็นโมดูลที่สนับสนุนฟังก์ชันทั่วไปของระบบปฏิบัติการซึ่งมีหน้าที่ให้บริการแก่แอพพลิเคชันผ่านทางการใช้ฟังก์ชันของเคอร์เนลและดีไวซ์ไดร์เวอร์

2.1.2 องค์ประกอบทางด้านฮาร์ดแวร์ (Elements of Embedded Hardware)

ส่วนสำคัญที่ถือได้ว่าเป็นปัจจัยหลักของระบบฝังตัวคือ ไมโครคอนโทรลเลอร์ (Microcontroller) ซึ่งเปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กที่ใช้ในการประมวลผลและการจัดการทางด้านต่างๆ โดยไมโครคอนโทรลเลอร์ที่ใช้ในการประมวลผลมีส่วนประกอบที่สำคัญได้แก่ เอ็มพียู (MPU) หน่วยความจำ(Memory) บัส (Bus) และอินพุต/เอาท์พุต (I/O) (ดังรูปที่ 2.7)

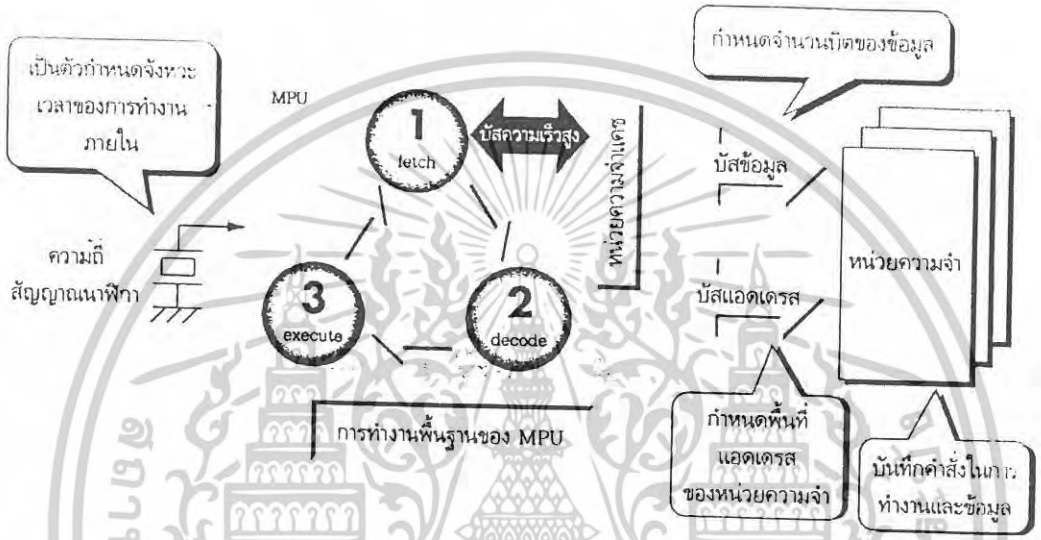


รูปที่ 2.7 สถาปัตยกรรมพื้นฐานของระบบฝังตัว (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 89.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1 เอ็มพียู (MPU)

เอ็มพียู คือหน่วยประมวลผลไมโคร (Microprocessing Unit) เปรียบเสมือนหน่วยประมวลผลกลาง (CPU) ในเครื่องคอมพิวเตอร์ มีหน้าที่ในการประมวลผลคำสั่งต่างๆและตอบสนองต่อคำสั่งนั้นๆ โดยมีขั้นตอนในการประมวลผลอยู่ 3 ขั้นตอน คือขั้นตอนการอ่านคำสั่ง การตีความคำสั่ง และขั้นตอนการทำตามคำสั่ง (ดังรูปที่ 2.8)



รูปที่ 2.8 ขั้นตอนการประมวลผลของหน่วยประมวลผลไมโครคอนโทรลเลอร์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 92.)

2.1.2.2 หน่วยความจำ (Memory)

หน่วยความจำ คือ ที่สำหรับเก็บรักษาข้อมูลข่าวสาร เช่น โปรแกรมและข้อมูลที่อยู่ภายในไมโครคอมพิวเตอร์สำหรับไมโครชิป ซึ่งประกอบด้วยส่วนที่เป็นฮาร์ดแวร์ และซอฟต์แวร์นั้น ที่เก็บซอฟต์แวร์ก็คือ หน่วยความจำนั่นเอง

2.1.2.3 บัส(Bus)

บัส คือ กลุ่มของสายสัญญาณที่ใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์ภายในและภายนอกไมโครคอนโทรลเลอร์ ซึ่ง สามารถแบ่งได้ออกเป็น 2 ประเภทคือ บัสภายในและบัสภายนอก

1. บัสภายใน(Internal Bus)

บัสภายใน เป็นสายสัญญาณเชื่อมต่ออุปกรณ์ในเอ็มพียู(MPU) เช่นรีจิสเตอร์ (Register) หน่วยความจำ และพอร์ต(Ports) เป็นต้น สามารถแบ่งออกได้เป็น 3 ประเภทหลักคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสบัส(Address Bus) บัสข้อมูล(Data Bus) และบัสควบคุม(Control Bus) ในส่วนของแอดเดรสบัสจะใช้ในการระบุอุปกรณ์ และข้อมูลการกระทำทั้งหมดนี้จะถูกกำหนดผ่านบัสควบคุม

2. บัสภายนอก(External Bus)

บัสภายนอกเป็นกลุ่มของสายสัญญาณที่ใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอกต่างๆ เช่น เซ็นเซอร์ ที่เก็บข้อมูล(Storage) เป็นต้น

อินพุตเอาท์พุตพื้นฐาน(Basic I/O)

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

1. ส่วนอินพุตพื้นฐาน (Basic Input)

ส่วนอินพุตพื้นฐาน เป็นส่วนที่ใช้ในการรับสัญญาณต่างๆจากอุปกรณ์ภายนอกมายังไมโครคอนโทรลเลอร์ โดยจะอยู่ในรูปของสัญญาณไฟฟ้า ซึ่งถูกจัดการโดนเอดีคอนเวอเตอร์ (A/D Converter) ในการแปลงสัญญาณจากแอนาล็อก ไปเป็นสัญญาณดิจิตอล (Digital) เพื่อที่จะให้เอ็มพียูนำไปประมวลผลได้

2. ส่วนเอาท์พุตพื้นฐาน (Basic Output)

ส่วนเอาท์พุตพื้นฐาน เป็นช่องทางออกของสัญญาณ หรือข้อมูลที่ได้ผ่านการประมวลผลจากเอ็มพียูแล้ว โดยข้อมูลหรือสัญญาณเหล่านี้จะอยู่ในรูปทางกายภาพที่เหมาะสมกับอุปกรณ์ที่นำมาเชื่อมต่อกับเอาท์พุตนั้นๆ

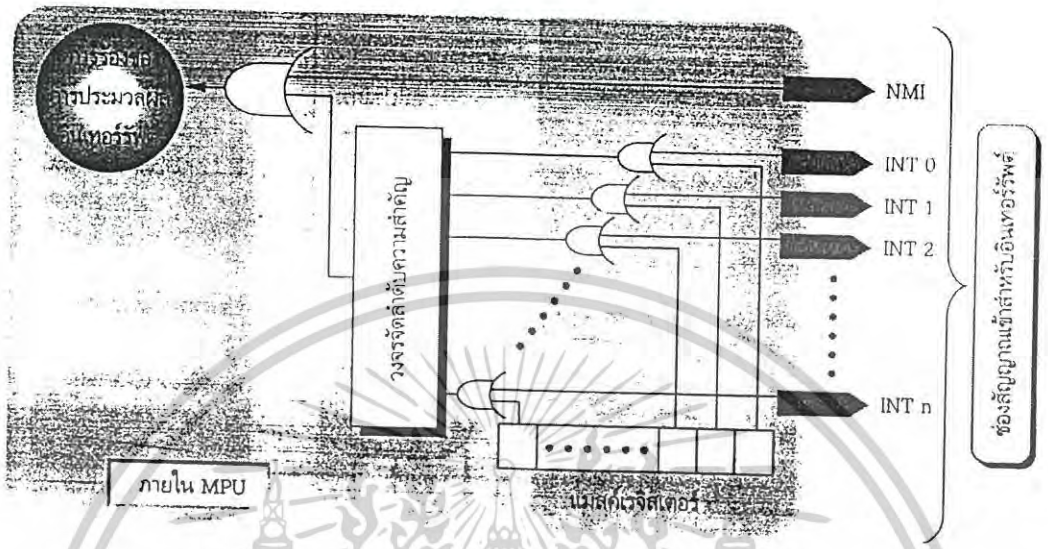
2.1.2.4 เทคโนโลยีการอินเทอร์รัพต์ (Interrupt Technology)

การอินเทอร์รัพต์ คือการเปลี่ยนการทำงานจากอย่างหนึ่งไปเป็นอีกอย่างหนึ่งเมื่อมีเหตุการณ์บางอย่างมาเป็นตัวกระตุ้น ซึ่งในส่วนของทางฮาร์ดแวร์นั้น จะมีส่วนที่ทำหน้าที่เกี่ยวกับการอินเทอร์รัพต์ดังกล่าว คือ ส่วนควบคุมการอินเทอร์รัพต์ (Interrupt Controller)

เมื่ออินเทอร์รัพต์เกิดขึ้น จะเกิดการสลับการทำงานไปยังงานที่ร้องขออินเทอร์รัพต์ ซึ่งในการสลับงานนั้นมี 2 วิธี คือ การกระโดดไปยังแอดเดรสในหน่วยความจำที่ถูกกำหนดไว้สำหรับการประมวลผลการอินเทอร์รัพต์โดยตรง วิธีที่ 2 คือการกระโดดโดยผ่านตารางเวคเตอร์ (Vector Table) ซึ่งตารางเวคเตอร์คือ การกำหนดพื้นที่ในหน่วยความจำไว้สำหรับรองรับการประมวลผลอินเทอร์รัพต์ เมื่อเกิดอินเทอร์รัพต์เกิดขึ้น ทั้งนี้ ทั้ง2 วิธีจะมีการเก็บจุดที่กระทำอยู่ก่อนหน้าที่จะเกิดการอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอร์รัพต์ เพื่อที่ว่าหลังจากกระทำการประมวลผลการอินเทอร์รัพต์เสร็จสิ้นแล้วจะกลับมาทำงานดังกล่าวต่อได้ (ดังรูปที่ 2.9)



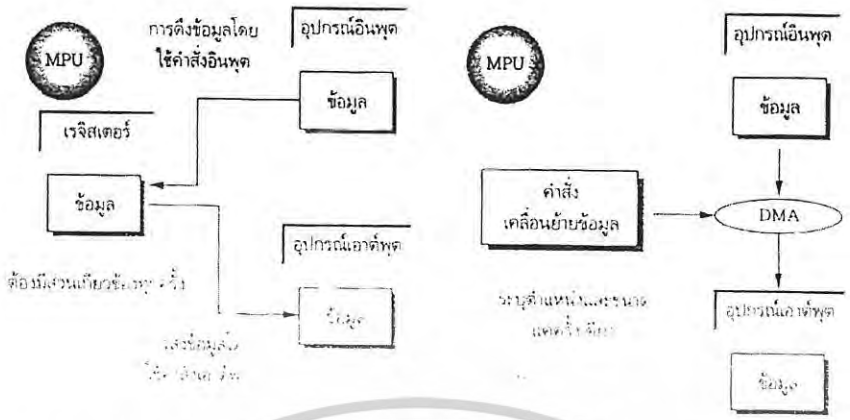
รูปที่ 2.9 เทคโนโลยีการอินเทอร์รัพต์ (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 103.)

2.1.2.5 เทคโนโลยีดีเอ็มเอ (DMA Technology)

DMA(Direct Memory Access) เป็นการย้ายข้อมูลระหว่างหน่วยความจำกับอุปกรณ์ หรือจะเป็นหน่วยความจำด้วยกันเอง โดยที่ไม่จำเป็นต้องผ่านการประมวลผลจากเอ็มพียู (ดังรูปที่ 2.10) เพียงแต่ต้องมีการแจ้งให้เอ็มพียูทราบแค่นั้น ซึ่งหลังจากดีเอ็มเอได้รับทราบตำแหน่งและข้อมูลที่จะทำการย้ายจากเอ็มพียูแล้ว ดีเอ็มเอก็จะทำการย้ายข้อมูลไปในตำแหน่งที่ได้รับมาโดยที่เอ็มพียูไม่ได้ยุ่งเกี่ยวเลย หรือเรียกได้ว่าเป็นอิสระจากเอ็มพียู สำหรับการเคลื่อนย้ายแบบดีเอ็มเอนั้นจะถูกควบคุมการทำงานโดยตัวควบคุมดีเอ็มเอ (DMA Controller) จุดประสงค์ของดีเอ็มเอ คือเพื่อเพิ่มประสิทธิภาพให้กับระบบฝังตัวและลดภาระงานของเอ็มพียู โดยมีขั้นตอนหลักๆดังนี้

1. แจ้งให้เอ็มพียูทราบ
2. ทำการร้องขอการใช้บัสจากบัสควบคุม
3. ครอบครองบัส (สามารถหยุดการทำงานของเอ็มพียูได้ จึงเป็นผลเสียถ้ามีการใช้ดีเอ็มเอมากเกินไป เช่น ถ้ามีการส่งข้อมูลขนาดใหญ่ จะทำให้การทำงานของเอ็มพียูช้าลงและเนื่องจากเอ็มพียูได้หยุดการทำงาน ทำให้ไม่สามารถตอบสนองอินเทอร์รัพต์ต่างๆที่ต้องการตอบสนองได้อย่างรวดเร็ว จึงไม่เหมาะสำหรับการนำมาใช้ในระบบฝังตัวที่ต้องการการตอบสนองแบบรวดเร็วได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

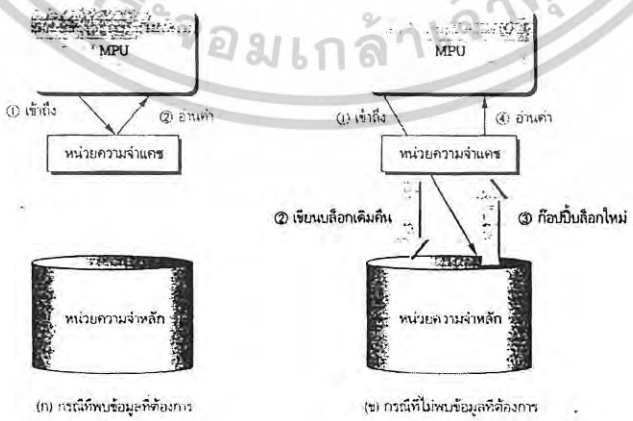


ก.การเคลื่อนย้ายข้อมูลแบบธรรมดา ข.การเคลื่อนย้ายข้อมูลแบบ DMA

รูปที่ 2.10 การเคลื่อนย้ายข้อมูลโดยวิธี DMAเทียบกับแบบธรรมดาที่ไม่มี DMA (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 105.)

2.1.2.6 เทคโนโลยีหน่วยความจำแคช (Cache Memory Technology)

หน่วยความจำแคชเป็นหน่วยความจำขนาดเล็ก มีไว้เพื่อเก็บข้อมูลที่เอ็มพียูนั้นใช้งานบ่อยๆ เพื่อจุดประสงค์คือเพื่อความรวดเร็วในการประมวลผล เนื่องจากการที่เอ็มพียูเข้าไปหาข้อมูลที่ต้องการในหน่วยความจำขนาดใหญ่ทำให้การค้นหานั้นเป็นไปด้วยความล่าช้า การที่นำแคชมาใช้จะทำให้เอ็มพียูนั้นเข้าถึงข้อมูลได้รวดเร็วมากขึ้น การทำงานของแคชคือ การคัดลอกสำเนาข้อมูลที่เอ็มพียูใช้ในการประมวลผล สำหรับการเข้าถึงในครั้งต่อๆไป โดยหลักการมีดังนี้ เมื่อเอ็มพียูต้องการข้อมูลที่ใช้ในการประมวลผลจะเข้ามาค้นหาที่แคชก่อน เมื่อเจอก็จะอ่านค่าจากแคช หากไม่เจอก็จะไปค้นหาที่หน่วยความจำหลัก เมื่อทำการค้นหาเจอ แคชก็จะทำการสำเนาข้อมูลที่เอ็มพียูใช้มาเก็บอยู่ในแคช พอเอ็มพียูต้องการใช้ข้อมูลเดิมก็จะเข้ามาค้นหาในแคชและสามารถนำข้อมูลไปใช้ได้เลย โดยที่ไม่ต้องเสียเวลาค้นหาอีก (ดังรูปที่ 2.11)

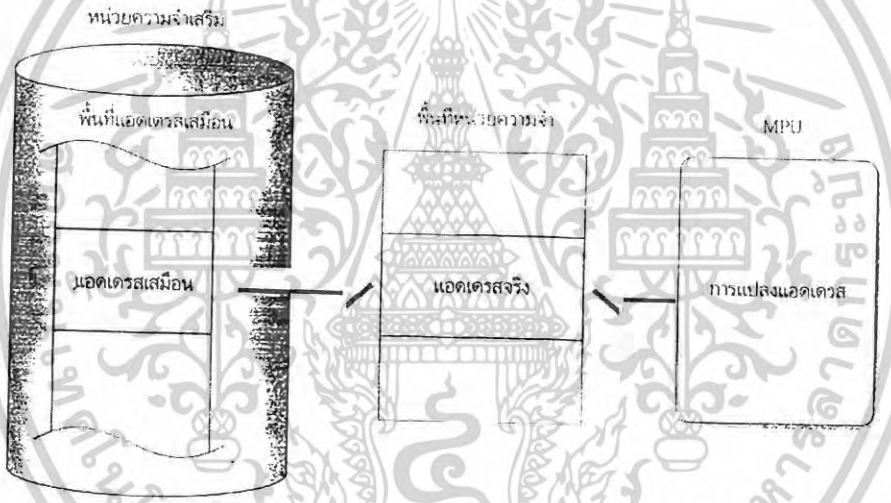


รูปที่ 2.11 การทำงานของหน่วยความจำแคช (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 108.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.7 เทคโนโลยีหน่วยความจำเสมือน(Virtual Memory Technology)

การใช้หน่วยความจำเสมือน มีจุดประสงค์คือ เพื่อทำให้โปรแกรมหรือข้อมูลที่มีขนาดใหญ่ถูกเรียกใช้งานได้โดยหน่วยความจำที่มีขนาดเล็กกว่าได้ ซึ่งก็คือโปรแกรมหรือข้อมูลนั้นจะถูกเก็บอยู่ในหน่วยความจำเสริม เช่นอุปกรณ์จำพวกฮาร์ดดิสก์ เป็นต้น โดยมีวิธีคือ การสร้างพื้นที่แอดเดรสเสมือนขึ้นมา แล้วแปลงแอดเดรสนี้ให้กลายเป็นแอดเดรสของหน่วยความจำจริง โดยผ่านฟังก์ชันการแปลงแอดเดรส แล้วทำการอ่านโปรแกรมจากหน่วยความจำจริง (ดังรูปที่ 2.12) แต่วิธีนี้ก็มีข้อเสียในระบบที่ต้องการการตอบสนองแบบเรียลไทม์ กล่าวคือถ้าหน่วยความจำมีขนาดเล็กในขณะที่โปรแกรมที่ต้องการให้ทำงานมีขนาดใหญ่ ก็จะต้องทำการโหลดโปรแกรมจากฮาร์ดดิสก์บ่อยๆ(Swapping) ทำให้เสียเวลาไปมากสำหรับการเคลื่อนย้ายข้อมูลระหว่างฮาร์ดดิสก์ และหน่วยความจำ



รูปที่ 2.12 ระบบการทำงานของหน่วยความจำเสมือน (Japan System House Association (JASA). 2549. แปลและเรียบเรียงโดย ดร.ธนารักษ์ ธีระมั่นคง: 111.)

2.1.3 การติดต่อสื่อสารกับผู้ใช้ทั้งแบบง่ายและแบบซับซ้อน (Simple System& Complex System)

การตอบสนองต่อผู้ใช้งานเพื่อตรวจดูความถูกต้องหรือการทำงานควรมีระบบที่จะสามารถสื่อสารกับผู้ใช้ เช่นอุปกรณ์ในการสั่งงานต่างๆผ่านทางตัวระบบนั้นๆเช่น ปุ่มในการกดเพื่อใส่ค่า Input หลอดไฟ LED เพื่อใช้ในการส่งสัญญาณบอกอะไรบางอย่างแก่ผู้ใช้ หรือเมนูที่แสดงผลออกทางหน้าจอ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบการติดต่อกับผู้ใช้แบบซับซ้อนสามารถควบคุมการทำงานต่างๆได้อย่างละเอียดและมีประสิทธิภาพมากยิ่งขึ้น ยกตัวอย่างเช่น ระบบสัมผัสเพื่อสั่งการผ่านทางหน้าจอได้ด้วยปลายนิ้ว หรืออุปกรณ์ที่คล้ายกัน รวมไปถึงการควบคุมและสั่งการโดยจอยสติ๊ก ต่างๆเป็นต้น

2.1.4 แพลตฟอร์มและโครงสร้างทางสถาปัตยกรรมของหน่วยประมวลผลข้อมูล (CPU Platform & Architecture Design)

ในระบบฝังตัว จำเป็นที่จะต้องมีหน่วยประมวลผลข้อมูลที่มีความแตกต่างกันในด้านของหน้าที่การนำไปใช้โดยการคำนึงถึงความเหมาะสมของงาน

2.1.5 อุปกรณ์ต่อพ่วงหรือช่องทางในการสื่อสารกับโมดูลอื่นๆ (Peripherals)

ในการสื่อสารกับโมดูลหรือคอมโพเนนต์อื่นๆในระบบนั้น จะเป็นต้องมีช่องทางในการสื่อสารเพื่อใช้ในการลำเลียงข้อมูล เช่น Serial Communication Interfaces (SCI) หรือ Universal Serial Bus (USB) เป็นต้น (ดังรูปที่ 2.13)

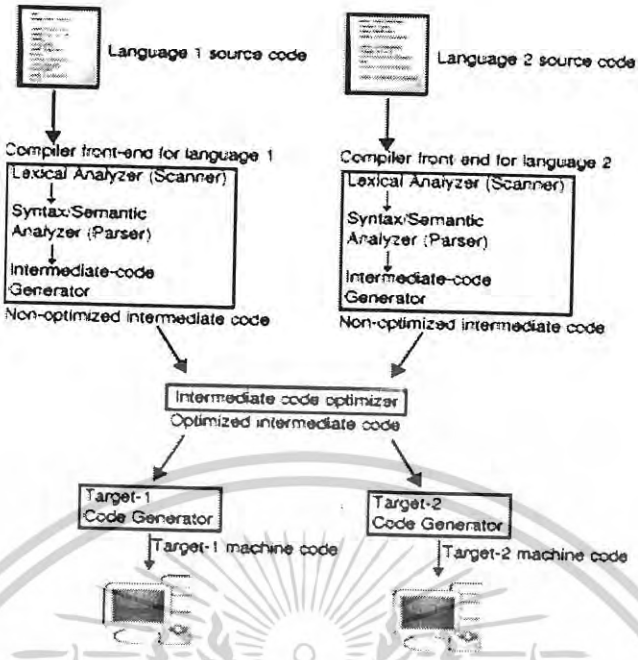
รูปที่ 2.13 ตัวอย่าง USB คอนเนคเตอร์ที่ใช้มาตรฐาน USBIF

(Thaiwikipedia. 2552: http://th.wikipedia.org/wiki/ยูเอสบี.)

2.1.6 เครื่องมือต่างๆในการพัฒนา(Tools)

ในการพัฒนาระบบฝังตัวนั้น จำเป็นที่จะต้องมีเครื่องมือต่างๆเช่น คอมไพเลอร์(Compiler)ที่ทำหน้าที่เป็นตัวช่วยแปลภาษาจากซอร์สโค้ดที่เป็นเท็กซ์ไฟล์(SourceCode)ไปเป็นทาร์เก็ตโค้ด(TargetCode) หรือโค้ดที่จำเป็นในการนำไปใช้งานต่อได้(Executable Program) ซึ่งโดยส่วนมากจะอยู่ในรูปของการแปลงโค้ดจากภาษาโปรแกรมมิ่งชั้นสูงไปสู่ชั้นล่าง (High-Level Program To Low-Level Program) เช่นในภาษาจาวา คอมไพเลอร์จะแปลงจาก ซอร์สไฟล์(นามสกุล .java)ไปสู่ไบท์โค้ด(นามสกุล .class) เป็นต้น เช่นเดียวกันในภาษาระดับล่างก็จำเป็นต้องมีการแปลงในลักษณะเช่นนี้โดยส่วนใหญ่จะเรียกตัวแปลงว่า แอสเซมเบลอร์(Assembler) (K.Ratchadaporn. 2549. **JAVA PROGRAMMING**: www.cs.su.ac.th/~kanawong/courses/517511/slide/JAVA%20PROGRAMMING%201.ppt) (ดังรูปที่ 2.14)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงการทำงานของคอมไพเลอร์ในภาษาจาวา

(Englishwikipedia. 2552: http://en.wikipedia.org/wiki/Compiler.)

2.1.7 Development Toolkit ที่ใช้ในการพัฒนา

ในการพัฒนาระบบแน่นอนว่าต้องมีภาษาในการเขียนเพื่อควบคุมส่วนประกอบต่างๆในระบบฝังตัวให้ทำงานเป็นไปตามที่ต้องการได้อย่างถูกต้อง ในการเลือกใช้ภาษานั้นควรพิจารณาจากจุดเด่นต่างๆเช่นความเหมาะสมของระบบกับงานหรือ การสนับสนุนของภาษานั้นๆกับงานเพื่อให้การพัฒนาเป็นไปโดยราบรื่นที่สุด ยกตัวอย่างเช่นภาษาจาวาเป็นภาษาที่โดดเด่นในด้านของการสนับสนุนการพัฒนาระบบแบบหลักการเชิงวัตถุ ซึ่งเหมาะสมกับทฤษฎีการพัฒนาระบบในครั้งนี้ จึงได้มีการเลือกใช้ภาษาจาวา นอกจากนี้ ภาษาจาวายังเป็นที่นิยมค่อนข้างแพร่หลาย จึงทำให้มีตัวเลือกในการค้นคว้าค่อนข้างมากเนื่องจากมีแหล่งค้นคว้าข้อมูลอยู่มากมาย

2.2 วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC)

วงจรการพัฒนาระบบ คือกระบวนในการพัฒนาระบบสารสนเทศ เพื่อแก้ปัญหาทางธุรกิจและตอบสนองความต้องการของผู้ใช้ได้ โดยภายในวงจรนั้นจะแบ่งกระบวนการพัฒนาออกเป็นกลุ่มงานหลักๆดังนี้

(Administrator. 2550. **การพัฒนาSoftware(System Development Life Cycle):** http://takasila.coe.psu.ac.th/~s4810168/SDLC%20%C7%A7%A8%C3%AA%D5%C7%D4%B5%A2%CD%A7%A1%D2%C3%BE%D1%B2%B9%D2%20Software.doc.)

- การกำหนดปัญหา(Problem Definition) หรือ การเลือกสิ่งที่จะนำมาพัฒนาระบบงาน (Project Identification and Selection) นับ ว่าเป็นขั้นตอนแรกในวงจรของการพัฒนา ขั้นตอนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

มักจะเกิดขึ้นอย่างเป็นทางการ จากการประชุมของฝ่ายบริหาร เพื่อที่จะค้นหาวิธีการทำงานที่มีประสิทธิภาพ และมุ่งหวังที่จะใช้แทนวิธีการทำงานแบบเดิม ปรับปรุงวิธีการทำงานหรือเพื่อสร้างรูปแบบบริการแบบใหม่

- การวิเคราะห์ปัญหา(Analysis) เมื่อ ผ่านขั้นตอนการการกำหนด หรือ เลือกโครงการที่จะทำการพัฒนาแล้ว ขั้นตอนต่อไปก็จะต้องนำเอาสิ่งที่ได้จากขั้นตอนแรกมาทำการวิเคราะห์ โดยนักวิเคราะห์ระบบจะต้องทำการ วิเคราะห์ระบบ ในขั้นตอนนี้เป็นขั้นตอนที่มีความสำคัญมาก และไม่ควรทำอย่างรีบเร่ง เนื่องจากโครงการพัฒนาจำนวนมากที่ประสบความล้มเหลวเพราะการวิเคราะห์ และออกแบบที่ไม่ถูกต้อง

- การออกแบบ(Design) จะเป็นการนำเอาสิ่งที่ได้จากการวิเคราะห์ มาออกแบบเป็นระบบงาน สำหรับการพัฒนาในขั้นตอนถัดไป เช่น การออกแบบฟอร์ม รายงาน ไดอะล็อก (Dialogue) ส่วนติดต่อกับผู้ใช้(Interface) ไฟล์และฐานข้อมูล(File&Database) โปรแกรมและการออกแบบโพรเซส(Program & Process Design)เป็นต้น

- การพัฒนาระบบงาน(Development) หรือ การสร้างระบบงานจริง ขั้นตอนนี้จะเป็นขั้นตอนที่นำเอาสิ่งที่ได้จากการออกแบบระบบมาทำการเขียนโปรแกรมหรือ สร้างตัวระบบงานขึ้นมาใช้งานจริง ผู้ที่มีบทบาทสูงในขั้นตอนนี้คือผู้พัฒนาโปรแกรม

- การทดสอบ(Testing) การทดสอบระบบจะเป็นการตรวจสอบความถูกต้องของระบบงานที่ถูกสร้างขึ้นมาว่าตรงตามกับความต้องการจริงๆ หรือไม่ การทดสอบ จะมีด้วยกันหลายระดับกล่าวคือ

1. การทดสอบในระดับโมดูล(Unit Test) เป็นการทดสอบการทำงานโดยแยกเป็นส่วนย่อยๆ ในแต่ละ module

2. การทดสอบการทำงานร่วมกัน(Integrate test) จะนำเอา โมดูลย่อยๆ มาทำการทดสอบการทำงานเป็นกระบวนการร่วมกัน

3. การทดสอบระดับระบบ(System test) เป็นการทดสอบโดยนำเอาโปรแกรมย่อยมาทดสอบการทำงานร่วมกันทั้งระบบ

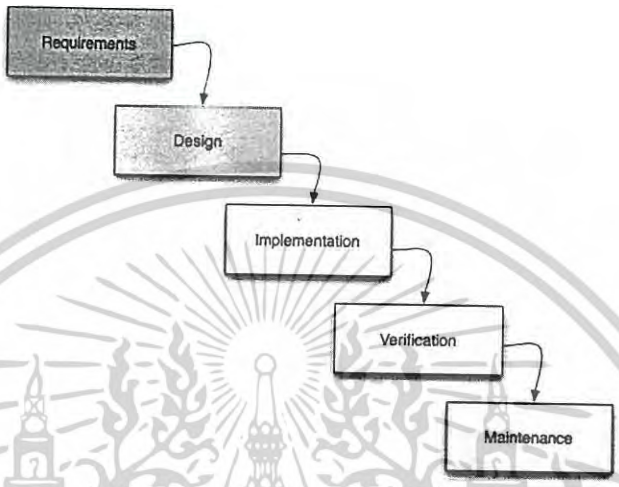
4. การทดสอบขั้นส่งมอบ(Acceptance test) เป็นการทดสอบขั้นสุดท้าย โดยผู้ใช้งาน เป็นผู้ทำการทดสอบ

ปัจจุบันมีรูปแบบของวงจรการพัฒนาระบบแตกแขนงออกไปมากมายดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
06062

2.2.1 SDLC ในรูปแบบน้ำตก(Waterfall Model)

การพัฒนาระบบงานด้วยหลักการนี้เปรียบเสมือนน้ำตก เมื่อทำขั้นตอนหนึ่งแล้วจะไม่สามารถย้อนกลับมาที่ขั้นตอนก่อนหน้าได้อีก หากมีข้อผิดพลาดจะไม่สามารถย้อนมาแก้ไขได้ (ดังรูปที่ 2.15)

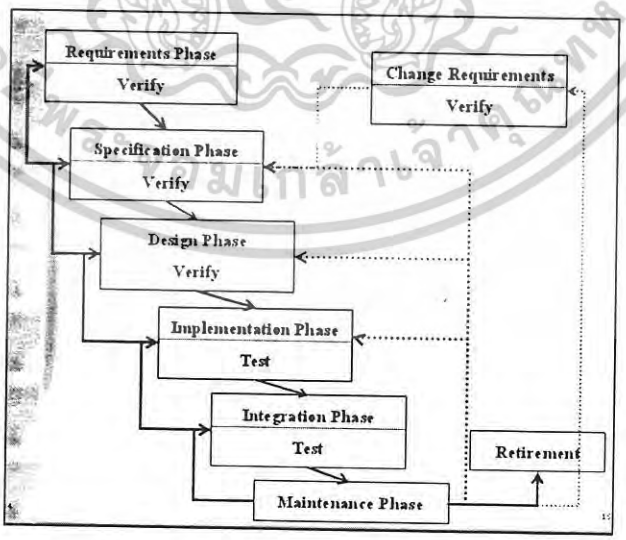


รูปที่ 2.15 วงจรการพัฒนาระบบแบบน้ำตก

(Thaiwikipedia. 2552: http://th.wikipedia.org/wiki/ทฤษฎีแบบจำลองน้ำตก)

2.2.2 SDLC ในรูปแบบน้ำตกชนิดปรับปรุง(Adapted Waterfall Model)

เป็นรูปแบบในการพัฒนาระบบงานที่ปรับปรุงมาจากแบบน้ำตกโดยแต่ละขั้นตอนสามารถกลับมาแก้ไขข้อผิดพลาดในขั้นตอนก่อนหน้าหรือย้อนกลับข้ามขั้นได้ (ดังรูปที่ 2.16)

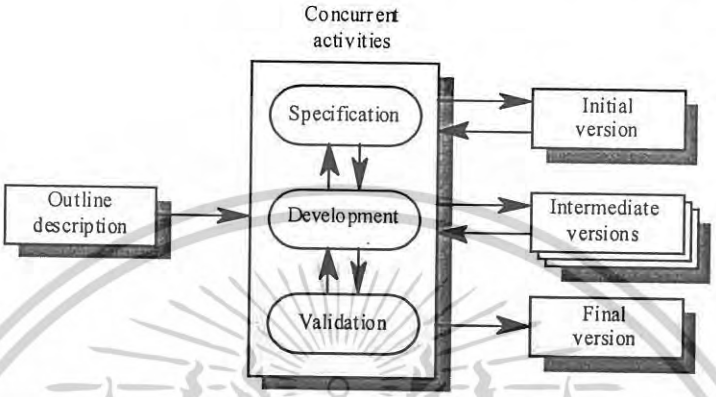


รูปที่ 2.16 วงจรการพัฒนาระบบแบบน้ำตกชนิดปรับปรุง (ICT IBM PC. 2552: www.cs.tu.ac.th/classes/471/undergradcs456-/public/AdditionalRead/IntroProcess.doc,)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 SDLC ในรูปแบบวิวัฒนาการ(Evolutionary Model)

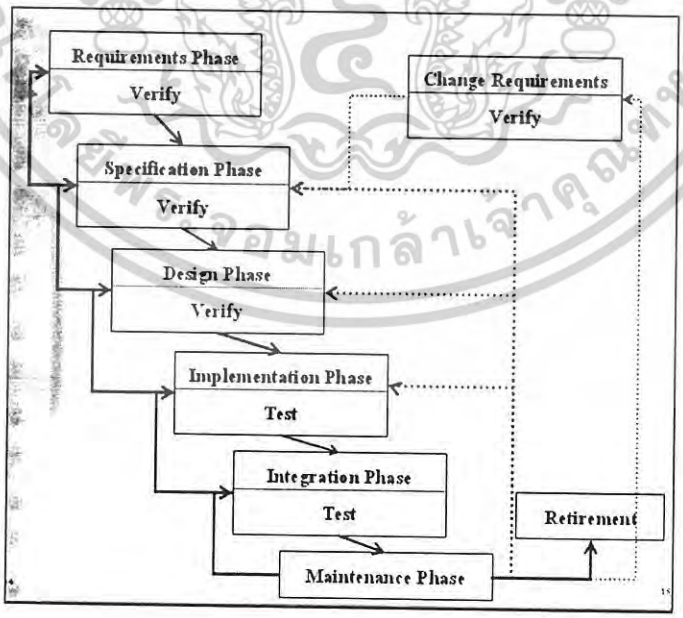
เริ่มต้นจากการกำหนดความต้องการเบื้องต้น แล้วทำการพัฒนาในเชิงวิวัฒน์ (Evolutionary) ซึ่งเป็นการปรับปรุงให้ระบบมีความสมบูรณ์มากขึ้นเรื่อยๆ จนกว่าระบบที่พัฒนานั้นจะสามารถใช้งานได้จริงและตรงกับความต้องการของผู้ใช้ (Prototyping) (ดังรูปที่ 2.17)



รูปที่ 2.17 วงจรการพัฒนาระบบแบบวิวัฒนาการ (วุฒิพงษ์ เรือนทอง. 2552: learners.in.th/file/wuttipongr/DevMethodology.ppt.)

2.2.4 SDLC ในรูปแบบส่วนเพิ่ม(Incremental Model)

การพัฒนาระบบงานในรูปแบบนี้ จะมีลักษณะคล้ายคลึงกับแบบวิวัฒนาการ แต่มีข้อแตกต่างที่ ผลิตภัณฑ์ ซึ่งการพัฒนาขึ้นครั้งแรกนั้นไม่ใช่ ผลิตภัณฑ์ ที่สมบูรณ์ จึงพัฒนาเพิ่มเติมในขั้นต่อๆไปจนได้ ผลิตภัณฑ์ ที่สมบูรณ์ (ดังรูปที่ 2.18)



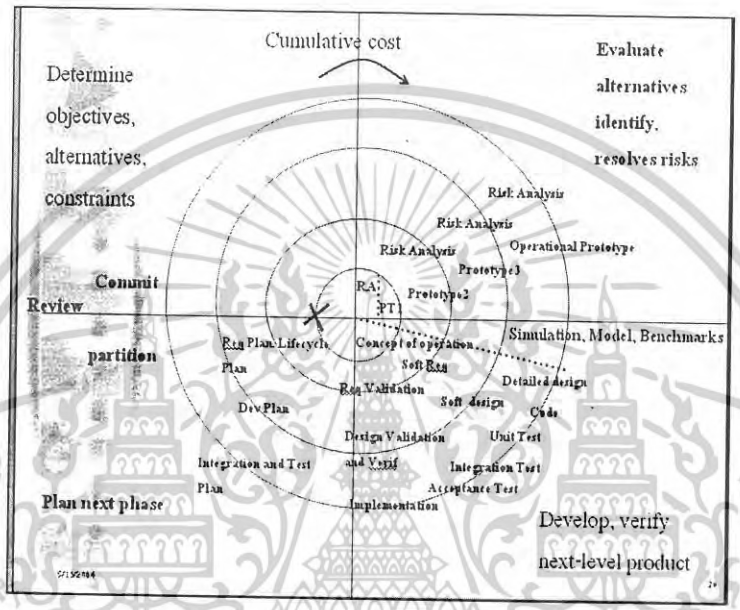
รูปที่ 2.18 วงจรการพัฒนาระบบแบบส่วนเพิ่ม (ICT IBM PC. 2552: www.cs.tu.ac.th/classes/471/undergradcs456-/public/AdditionalRead/IntroProcess.doc.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 SDLC ในรูปแบบวนรอบ(Spiral Model)

มีลักษณะเป็นวงจรวิเคราะห์ – ออกแบบ – พัฒนา – ทดสอบ และจะวนกลับมาในแนวทางเดิม จนกว่าจะได้ Product ที่สมบูรณ์ การพัฒนาในรูปแบบนี้มีความยืดหยุ่นมากที่สุด (ดังรูปที่ 2.19)

จากกระบวนการในการพัฒนาระบบ จะต้องมีวิธีการหรือแนวทางที่จะนำกระบวนการนั้นมาลงมือปฏิบัติเพื่อให้การพัฒนาระบบเพื่อให้เกิด ผลสำเร็จจนกลายเป็นระบบที่สามารถใช้งานได้อย่างมีประสิทธิภาพ วิธีการดังกล่าวเรียกว่า แนวทางปฏิบัติ



รูปที่ 2.19 วงจรการพัฒนาระบบแบบวนรอบ (ICT IBM PC. 2552: www.cs.tu.ac.th/classes/471/undergradcs456-/public/AdditionalRead/IntroProcess.doc.)

2.3 แนวทางปฏิบัติ (Methodologies)

แนวทางปฏิบัติ คือ วิธีการหรือแนวทางที่จะนำกระบวนการพัฒนาระบบสารสนเทศมาปฏิบัติจริงจนกลาย เป็นระบบสารสนเทศที่สามารถใช้งานได้ โดยมีการระบุถึงขั้นตอนในการปฏิบัติเพื่อใช้พัฒนาระบบในวงจรการพัฒนาระบบ

ใน ปัจจุบันมี แนวทางปฏิบัติ ที่ใช้ในการพัฒนาระบบใน SDLC มากมายแตกต่างกันไป ทั้งนี้ก็เพื่อให้สามารถนำไปประยุกต์ใช้ให้เหมาะสมกับระยะเวลาและงบประมาณใน การพัฒนาระบบของแต่ละองค์กร จึงเกิด แนวทางปฏิบัติหลากหลายวิธีการ ยกตัวอย่าง แนวทางปฏิบัติที่นิยมใช้กันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Structured System Analysis and Design Methodology (SSADM)

แนวทางในการพัฒนาระบบสารสนเทศ แบบ Structured System Analysis and Design Methodology (SSADM) จะใช้แบบจำลองที่เป็นแผนภาพเพื่ออธิบายขั้นตอนการทำงานและข้อมูลทั้งหมดของระบบ โดยเรียกวิธีการใช้แผนภาพเพื่อจำลองขั้นตอนการทำงานหลักของระบบว่า "Process-Center Approach" และเรียกวิธีการใช้แผนภาพเพื่อจำลองข้อมูลของระบบว่า "Data-Center Approach" ข้อดี ของ SSADM คือ สามารถรวบรวมความต้องการจากผู้ใช้ได้เป็นระยะเวลานานก่อนที่จะเริ่มเขียน โปรแกรม และการเปลี่ยนแปลงความต้องการมีน้อย เนื่องจากก่อนที่จะถึงขั้นตอนการเขียนโปรแกรม ข้อมูลต่าง ๆ ที่วิเคราะห์มานั้นจะต้องได้รับการอนุมัติเห็นชอบจากเจ้าของระบบก่อน จึงจะสามารถเข้าสู่ขั้นตอนการเขียนโปรแกรมได้ หมายความว่าข้อมูลทุกอย่างที่วิเคราะห์และออกแบบมานั้นจะต้องตรงตามความต้อง การของผู้ใช้และเจ้าของระบบมากที่สุดนั่นเอง และข้อเสียของการพัฒนา คือใช้เวลานานในขั้นตอนการวิเคราะห์และออกแบบระบบ และการออกแบบต่าง ๆ จะร่างลงบนกระดาษ ซึ่งผู้ใช้หรือเจ้าของระบบไม่สามารถทดลองใช้งานได้ จึงอาจจะทำให้ไม่ทราบถึงปัญหาที่เกิดขึ้น ส่งผลให้อาจเกิดปัญหาในระหว่างขั้นตอนการเขียนโปรแกรมได้

2.3.2 Rapid Application Development-based Methodology (RAD)

Rapid Application Development-based Methodology เป็น แนวทางปฏิบัติแนวใหม่ที่พัฒนาขึ้นในช่วงปี ค.ศ.1990 เพื่อแก้ไขจุดอ่อนของ แนวทางปฏิบัติแบบโครงสร้าง ด้วยการปรับระยะในวงจรการพัฒนาระบบ ให้มีขั้นตอนการทำงานที่รวบรัดมากขึ้น มีการเลือกใช้เครื่องมือ และเทคนิคต่าง ๆ เพื่อช่วยให้การพัฒนาระบบนั้นดำเนินการได้อย่างรวดเร็ว อีกทั้งผู้ใช้ระบบยังสามารถทดลองใช้โปรแกรมต้นแบบเพื่อบอกนักวิเคราะห์ระบบ ได้ว่า ระบบที่ออกแบบมานั้นถูกต้องหรือไม่ และมีข้อผิดพลาดใดเกิดขึ้นบ้าง

2.3.3 Object-Oriented Analysis and Design Methodology (OOADM)

จาก แนวทางปฏิบัติทั้งหมดในกลุ่มของ RAD Methodology ที่กล่าวมาข้างต้นนั้น เป็นการดัดแปลงวิธีการในการพัฒนาระบบจาก แนวทางปฏิบัติแบบ Structured System Analysis and Design กล่าวคือ วิเคราะห์และออกแบบเพื่อแก้ปัญหาของระบบงานด้วยการอ้างอิงกับขั้นตอนการทำงาน (Process-Centered Approach) และอ้างอิงกับข้อมูลทั้งหมดของระบบ (Data-Centered Approach) แต่ทั้ง Process-Centered Approach และ Data-Centered Approach มีความสำคัญใกล้เคียงกันมาก ทำให้นักวิเคราะห์ระบบและทีมงานพัฒนาระบบเกิดความสับสนในการเลือกที่จะอ้างอิงด้วยวิธีการใด จึงทำให้เกิด "Object-Oriented Analysis and Design Methodology" เพื่อพัฒนาระบบในวงจร SDLC ด้วยการวิเคราะห์และออกแบบเพื่อแก้ปัญหาระบบงาน โดยมองปัญหาเหล่านั้นให้เป็นวัตถุที่ประกอบไปด้วยขั้นตอนการทำงาน รวมกับข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปฏิบัติตามขั้นตอนต่าง ๆ ของ แนวทางปฏิบัติเพื่อพัฒนาระบบใน SDLC นั้นแต่ละแนวทางปฏิบัติมีการใช้แบบจำลอง เครื่องมือ และเทคนิคที่แตกต่างกันไป เพื่อช่วยให้การดำเนินการในแต่ละขั้นตอนสะดวกยิ่งขึ้น และสามารถรองรับระบบงานที่มีความซับซ้อนได้

แบบจำลอง (Model) คือ สัญลักษณ์ที่ใช้ในการจำลองข้อเท็จจริงต่าง ๆ ที่เกิดขึ้นในระบบ ไม่ว่าจะเป็นแบบจำลองข้อมูล (Data Model) หรือขั้นตอนการทำงานของระบบ (Process Model)

ตัวอย่างแบบจำลอง

- Flow Chart
- Data Flow Diagram (DFD)
- Entity Relationship Diagram (ERD)
- Structure Chart
- Use Case Diagram
- Class Diagram
- Sequence Diagram
- PERT Chart
- Gantt Chart
- Organization Hierarchy Chart

เครื่องมือในการพัฒนาระบบ (Tools) คือ ซอฟต์แวร์ที่ช่วยสร้างหรือวาดแบบจำลองชนิดต่าง ๆ ตรวจสอบความถูกต้องของแบบจำลอง และยังช่วยสร้างรายงานและแบบฟอร์มประกอบการพัฒนา

2.4 ทฤษฎีแนวคิดของการพัฒนาระบบโดยใช้หลักการเชิงวัตถุ (Object Oriented Analysis And Design)

ในปัจจุบันการพัฒนาระบบ ไม่ได้หยุดอยู่แค่การทำระบบหนึ่งๆและปล่อยให้จบไปโดยที่ไม่ได้มีการพัฒนาต่อยอดหรือนำแนวคิดมาประยุกต์ใช้ต่อ เพราะเหตุนี้ ทำให้การพัฒนาระบบเชิงวัตถุได้เข้ามามีบทบาทมากและได้รับความนิยมอย่างมากในโลกปัจจุบัน เพราะเหตุนี้จึงได้มีหลักการที่จะนำแนวความคิดเชิงวัตถุดังกล่าวมาเพื่อที่จะใช้สร้างระบบฝังตัวที่มีความแตกต่างจากระบบฝังตัวแบบเดิมโดยมีแนวคิดเชิงวัตถุเข้ามาเกี่ยวข้องและสามารถพัฒนาต่อยอดในอนาคตได้เป็นหลัก (ดร.อดิศร ณ อุบล. 2548. **UML**: learners.in.th/file/apealex/UML.ppt)
(โปรดดูตารางที่ 2.1 ประกอบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 เปรียบเทียบการวิเคราะห์และออกแบบระบบวิธีเดิมกับวิธีเชิงวัตถุ

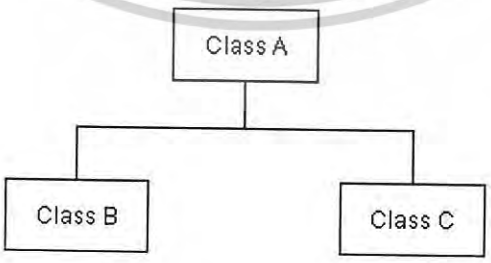
(ดร.อดิศร ณ อุบล. 2548. **UML**: learners.in.th/file/apealex/UML.ppt)

วิธีเดิม	**วิธีเชิงวัตถุ**
เริ่มต้นจากการวิเคราะห์เอกสารผลลัพธ์ และการทำงานของระบบงานเดิม	เริ่มต้นการวิเคราะห์จากออปเจ็กต์ ที่ สามารถเห็นได้ชัดเจน
แตกการทำงานออกเป็นหน่วยย่อยๆ	แบ่งกลุ่มของออปเจ็กต์ ตามคุณลักษณะ
องค์ประกอบต่างๆของระบบเช่นการประมวลผล การออกรายงาน การคำนวณ จะเกี่ยวพันกัน การเปลี่ยนแปลงจะกระทบซึ่งกันและกัน	แต่ละออปเจ็กต์ เป็นอิสระต่อกัน การเปลี่ยนแปลงจะไม่กระทบกัน
การปรับเปลี่ยนระบบต้องแก้ไขโค้ด	การปรับเปลี่ยนระบบทำได้โดยการเปลี่ยนแอตทริบิวต์, ฟังก์ชัน ของ ออปเจ็กต์
เครื่องมือ ที่สนับสนุนมีน้อยลง	เครื่องมือ ที่สนับสนุนมีมากขึ้น

ในโครงการหุ่นยนต์จัดเก็บวัสดุนั้นประโยชน์ของการพัฒนาระบบแบบหลักการเชิงวัตถุที่สามารถนำมาประยุกต์ใช้และเห็นว่าเกี่ยวข้องโดยตรงมีดังต่อไปนี้

2.4.1 การสืบทอดคุณสมบัติ(Inheritance)

การสืบทอดคุณสมบัติเป็นจุดเด่นอย่างหนึ่งของการพัฒนาระบบเชิงวัตถุที่นำมาใช้ในการพัฒนาระบบหุ่นยนต์จัดเก็บวัสดุ โดยสามารถสร้างคลาสหลักขึ้นมา 1 คลาสแล้วกำหนดให้มีคุณสมบัติต่างๆ ที่จำเป็นต้องมี แล้วจึงสร้างคลาสอีกคลาสหนึ่งขึ้นมาเพื่อรับคุณสมบัติทั้งหมดจากคลาสหลัก และเพิ่มคุณสมบัติอื่นเข้าไปอีกในคลาสนี้ โดยเรียกคลาสหลักว่าคลาสแม่(Superclass) และเรียกคลาสย่อยว่า คลาสลูก(Subclass) (ดังรูปที่ 2.20)



รูปที่ 2.20 อธิบายตัวอย่างการสืบทอดคุณสมบัติ

(Mr.POP. 2550: http://www.webthaidd.com/java/webthaidd_article_263_1.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักของการสืบทอดคุณสมบัติจะทำให้ความสัมพันธ์ระหว่างออปเจ็กต์มีความชัดเจนยิ่งขึ้นกล่าวคือถ้ามีความสัมพันธ์ที่ชัดเจนมากขึ้นเท่าใด จะส่งผลให้การออกแบบระบบงานง่ายขึ้น

ผู้ออกแบบระบบงานเชิงวัตถุสามารถออกแบบระบบงานขนาดใหญ่ได้โดยการอาศัย ออปเจ็กต์ที่มีการนิยามไว้ก่อนหรือที่มีผู้อื่นทำการออกแบบไว้ก่อนแล้ว ซึ่งเป็นที่มาของการนำกลับมาใช้ใหม่ (Reusability)

ข้อดีของการสืบทอดคุณสมบัติที่สามารถนำไปประยุกต์ใช้ในการพัฒนาระบบอื่นๆต่อไป

1. ทำให้มีโครงสร้างที่เป็นระบบ สามารถปรับเปลี่ยนได้ง่าย

แน่นอนว่าในการพัฒนาระบบเชิงวัตถุนั้นข้อดีอย่างแรกคือการทำให้ทุกๆอย่างมีความเป็นระบบและโครงสร้างที่สามารถปรับเปลี่ยนเพื่อนำไปใช้งานหรือปรับแก้ต่อไปในอนาคตได้ง่าย

2. ลดเวลาในการพัฒนาระบบ

ยกตัวอย่างเช่นการเขียนโค้ดเมื่อทำการนำระบบที่มีอยู่แต่แรกไปพัฒนาต่อ เนื่องจากใช้การสืบทอดคุณสมบัติทำให้ช่วยย่นระยะเวลาในการพัฒนาระบบ (สร้างใหม่หรือเขียนใหม่)ลงไปได้เป็นอย่างดี อีกทั้งยังทำให้โค้ดโดยไม่รวมเยอะจนเกินไปในกรณีที่ทำการคัดลอกฟังก์ชันมา

3. ลดค่าใช้จ่ายในการพัฒนาระบบ

หากไม่มีการนำระบบที่มีการพัฒนามาเพื่อต่อยอดในการพัฒนาต่อไปโดยใช้หลักการของการนำกลับมาใช้ใหม่แล้ว ทำให้ระบบต้องสร้างฟังก์ชันที่จำเป็นในการพัฒนาใหม่แทบทุกอย่างทำให้ต้องเสียค่าใช้จ่ายในการพัฒนาเป็นอย่างมาก ในกรณีนี้ยกตัวอย่างได้ชัดเจนคือ พวกไลบรารี (Library) หรือ API(Application Programming Interface) ที่มีผู้เห็นความสำคัญของการนำกลับมาใช้ใหม่ช่วยกันพัฒนาต่อเนื่องมาให้ผู้ใช้งานในรุ่นต่อๆมาได้มีความสะดวกในการพัฒนานั่นเอง

2.4.2 ความเป็นโมดูล(Modularity)

ความหมายดั้งเดิมของการโปรแกรมเชิงโครงสร้างโมดูล คือการออกแบบโปรแกรมที่ประกอบไปด้วยโปรแกรมย่อย ซึ่งเป็นส่วนประกอยย่อยๆ หลายๆ ส่วนนำมาปะติดปะต่อเข้าด้วยกันนอกจากนี้ ด้วยมาตรฐานการเรียกใช้ฟังก์ชันนั้นไม่มีข้อจำกัด (เรียกได้อิสระ) การเขียนโปรแกรมที่ไม่เหมาะสม จะทำให้การเข้าถึงตัวแปรและฟังก์ชันต่างๆ กระทำได้จากหลายทาง และเมื่อแก้ไขโปรแกรมบางส่วนอาจจะทำให้เกิดปัญหาต่อโปรแกรมตั้งระบบ (เกิดบักขึ้นจากการแก้ไขโปรแกรม)

ด้วยประโยชน์ของด้านการนำกลับมาใช้ใหม่ของโมดูลาริตี้ทำให้ ระบบที่ออกแบบโดยหลักการพัฒนาเชิงวัตถุนั้นมีข้อได้เปรียบอย่างชัดเจนยิ่งขึ้นหากถ้าระบบนั้นมีจุดประสงค์เพื่อให้มีการพัฒนาต่อในอนาคตนั้นเอง ด้วยความได้เปรียบของความง่ายในการเข้าใจระบบและพัฒนาต่อทำให้การเลือกแนวทางการพัฒนาเชิงวัตถุดูมีความสำคัญมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์จัดเก็บวัสดุ ที่ออกแบบโดยอิงทฤษฎีการออกแบบระบบเชิงวัตถุนี้ก็เช่นเดียวกัน ยกตัวอย่างเช่นการพัฒนาระบบหุ่นยนต์ตามปกตินั้นจะใช้วิธีเขียนโค้ดแบบโพรซีเยอร์ กล่าวคือการเขียนโค้ดเป็นลำดับไล่ลงมา เมื่อเกิดปัญหาหรือต้องการที่จะปรับแก้ระบบหรือแม้แต่จะศึกษาข้อมูลบางส่วนเพื่อที่จะนำไปพัฒนาต่อมีความยุ่งยากในการค้นหาส่วนที่ต้องการจากโค้ดทั้งหมดดังกล่าว แต่เมื่อใช้หลักการพัฒนาระบบแบบเชิงวัตถุที่มีความเป็นโมดูลาริตี้นั้น จะให้ทุกองค์ประกอบที่ควรจะอยู่ด้วยกันสามารถรวมกันเป็นคอมโพเนนต์หรือฟังก์ชันเดียวกันซึ่งแน่นอนว่ามีความง่ายต่อผู้ที่จะมาศึกษาต่อ แต่ต้องทำใจยอมรับว่าเป็นการยากในการพัฒนาระบบดังกล่าวเช่นเดียวกัน

ข้อด้อยของการโปรแกรมเชิงโครงสร้างโมดูลคือ เมื่อโปรแกรมมีความซับซ้อนมากขึ้น ความยุ่งยากของโครงสร้างก็ทวีมากขึ้นเช่นกัน ทำให้การออกแบบโปรแกรมที่มีขนาดใหญ่ มีหน้าที่การทำงานซับซ้อน เป็นไปด้วยความยากลำบากยิ่งขึ้นได้

ใจความของการพัฒนาระบบเชิงวัตถุนั้น มีเพื่อให้สำหรับผู้สนใจที่จะพัฒนาระบบต่อจากนี้มีความเข้าใจหรือสามารถปรับแก้โดยการนำฟังก์ชันหรือคอมโพเนนต์ต่างๆที่มีอยู่ในระบบหรือแม้แต่แนวคิดทางอัลกอริทึม และนำไปประยุกต์เข้ากับระบบของตนเองได้โดยง่าย

อันที่จริงแล้วคุณสมบัติของการพัฒนาระบบเชิงวัตถุนั้นยังมีอีกมากมายหลายหัวข้อ อาทิ เช่น การปกปิดไม่ให้เห็นสภาวะภายในหรือการทำงานภายใน(Encapsulation) การพ้องรูป, ภาวะการณ์ที่มีหลากหลายรูปแบบ(Polymorphism) แต่เนื่องจากหัวข้อหรือประเด็นดังกล่าวไม่ได้มีความสอดคล้องกับโครงการมากนัก จึงไม่ขอกล่าวถึง

2.5 โมเดลที่ใช้ในการพัฒนา - ยูเอ็มแอล โมเดล(UML Standard Model)

ยูเอ็มแอลคือ โมเดลมาตรฐานที่ใช้หลักการออกแบบระบบเชิงวัตถุ OOP(Object oriented programming) รูปแบบของภาษา ยูเอ็มแอลจะมีสัญลักษณ์(Notation) ที่นำไปใช้ใน โมเดลต่างๆ UMLจะมีข้อกำหนดกฎระเบียบต่างๆ ในการโปรแกรม โดยกฎระเบียบต่างๆ จะมีความหมายต่อการเขียนโปรแกรม ดังนั้นการใช้ ยูเอ็มแอลจะต้องทราบความหมายของ สัญลักษณ์ต่างๆ เช่น Generalize, association dependency class และ package สิ่งเหล่านี้มีความจำเป็นอย่างยิ่งต่อการตีความของการวิเคราะห์และออกแบบระบบ ก่อนนำไปใช้ในระบบงานจริง ในปัจจุบันมีเครื่องมือมากมายที่สามารถแปลง โมเดลยูเอ็มแอลเป็น โค้ดภาษาต่างๆ ยกตัวอย่าง เช่น ภาษาจาวา(Java), พาวเวอร์บิลเดอร์(Power Builder) และ วิชวลเบสิค(Visual Basic) เป็นต้น

(www.itmelody.com. 2552: http://www.itmelody.com/tu/uml1.htm)

จุดประสงค์ของการนำยูเอ็มแอลโมเดลมาใช้ในการพัฒนาระบบเชิงวัตถุนี้ แน่นอนว่าเพื่อสนับสนุนวิธีคิดเชิงวัตถุเป็นหลักตามทฤษฎีทั้งหลายที่ได้กล่าวไปข้างต้น และความสามารถของโมเดลยูเอ็มแอลนั้นยังสามารถสรุปโดยรวมว่าสามารถเป็นโมเดลที่ออกแบบวางแผนการทำงานของระบบได้เป็นอย่างดีจากผู้ที่ได้เคยลองศึกษาหรือใช้งานมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยูเอ็มแอล ถูกนำมาใช้ในการพัฒนาซอฟต์แวร์อย่างกว้างขวาง และทำให้การทำงานมีคุณภาพ ดังนี้

1. ช่วยลดระยะเวลาในการพัฒนาระบบงาน(Shortest Development life cycle)
2. ช่วยเพิ่มความสามารถในการทำงาน(Increase productivity)
3. ช่วยเพิ่มคุณภาพของระบบงาน (Improve software quality)
4. รองรับระบบงานเดิม(Support legacy system)
5. ช่วยในการสื่อสารระหว่างทีมผู้พัฒนาระบบงาน(Improve team connectivity)

สาเหตุที่เลือกใช้ยูเอ็มแอล

1. ยูเอ็มแอลสามารถแสดงส่วนประกอบในการสร้างโปรเจคในรูปของแนวคิดเชิงวัตถุ
2. เชื่อมแนวคิดกับการออกแบบระบบโดยใช้โมเดลเชิงวัตถุได้
3. ง่ายต่อการทำความเข้าใจและสามารถแปลงเป็นโค้ดโปรแกรมได้

ในการวางแผนงานหรือแบบจำลองของระบบโดยใช้โมเดลยูเอ็มแอลนั้น จำเป็นจะต้องจำแนกแต่ละสโคป(Scope)หรือมุมมองของการออกแบบออกเป็นหลายส่วนตามไดอะแกรมที่ยูเอ็มแอลได้วางไว้ ซึ่งการจะเลือกใช้ไดอะแกรมใดบ้างนั้นขึ้นอยู่กับจุดประสงค์ของผู้ออกแบบและพัฒนา รวมไปถึงผู้ใช้งานต่อไป เป็นสำคัญ โดยไดอะแกรมที่เห็นว่ามีความสำคัญ มีดังต่อไปนี้

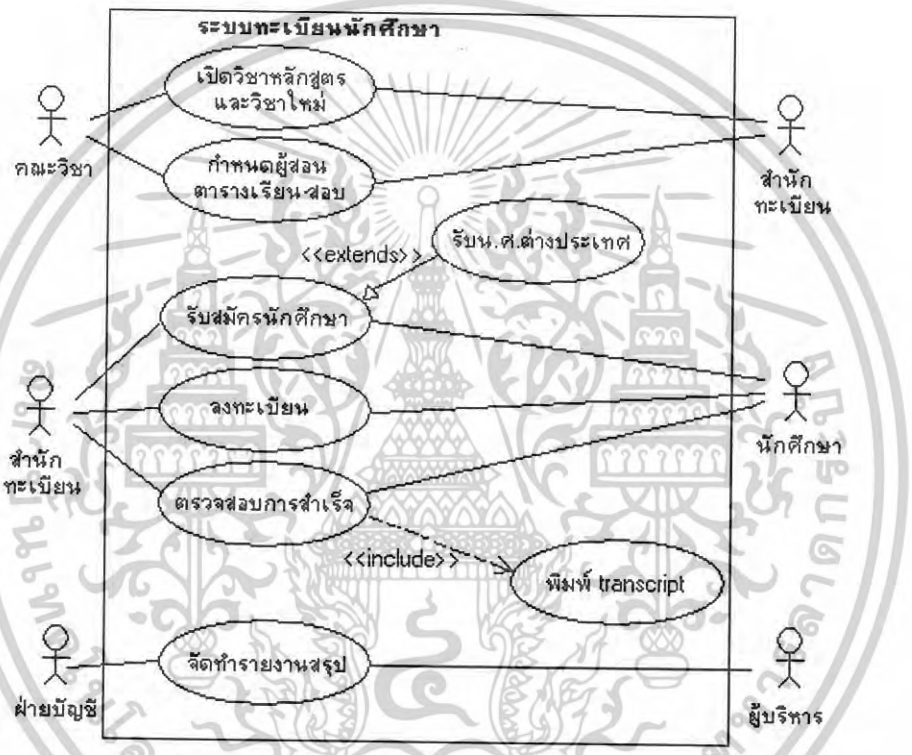
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 แผนภาพขอบเขตของระบบ(Use Case Diagram)

ยูสเคสไดอะแกรมเป็นแผนภาพที่ใช้ที่แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน ยูสเคสไดอะแกรมประกอบด้วย (ดังรูปที่ 2.21)

- แอ็คเตอร์(Actor) คือ ผู้ที่กระทำกับระบบ อาจเป็นผู้ที่ทำการส่งข้อมูล, รับข้อมูล หรือแลกเปลี่ยนข้อมูลกับระบบนั้นๆ เช่น ลูกค้ากับระบบสั่งซื้อสินค้าทางโทรศัพท์

- ยูสเคส คือ หน้าที่หรืองานต่างๆในระบบ เช่น การเช็คสต็อค การสั่งซื้อสินค้า เป็นต้น - รีเลชันชิพ(Relationship) คือ ความสัมพันธ์ระหว่าง ยูสเคส กับ แอ็คเตอร์



รูปที่ 2.21 แสดงยูสเคสไดอะแกรม ของระบบลงทะเบียน

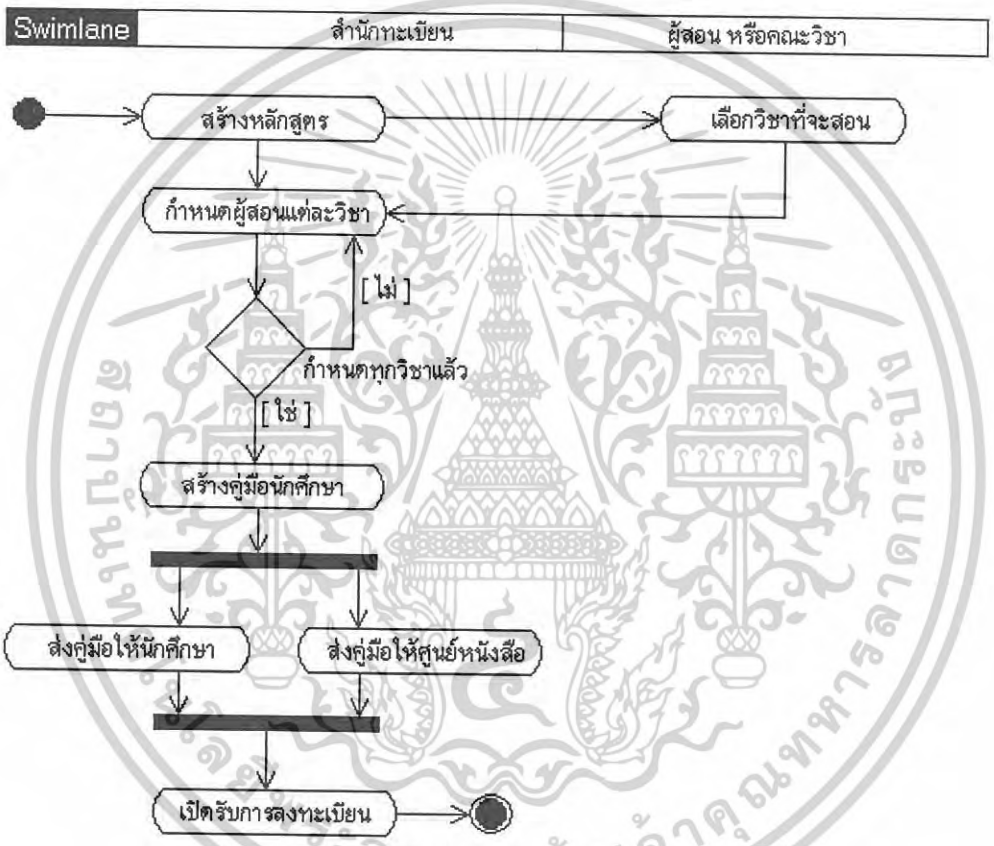
(thaiall.com. 2551: http://www.thaiall.com/uml/use_case02.gif.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 แผนภาพกิจกรรม(Activity Diagram)

แอกทิวิตี้ไดอะแกรมเป็นแผนภาพที่ใช้ที่แสดงขั้นตอนการทำงานของ ยูสเคส (เช่นเดียวกับซีเควสซ์ไดอะแกรม) แต่จะเน้นไปที่งานย่อยของวัตถุ โดยจะมีกระบวนการทำงานคล้ายกับ โฟลวชาร์ต(Flowchart) (ดังรูปที่ 2.22)

แอกทิวิตี้ไดอะแกรมบางครั้งมีลักษณะคล้าย สวิมเลน(Swimlane) โดยจะแบ่งกลุ่มกิจกรรมที่เกิดขึ้นเป็นช่อง โดยกำกับแต่ละช่องด้วยชื่อของออปเจ็กต์ แต่ละ สวิมเลนแสดงถึงกิจกรรมที่เกิดขึ้นกับ ออปเจ็กต์นั้นๆ



รูปที่ 2.22 ตัวอย่างแอกทิวิตี้ไดอะแกรม การลงทะเบียน

(thaiall.com. 2551: http://www.thaiall.com/uml/act_regist1.gif.)

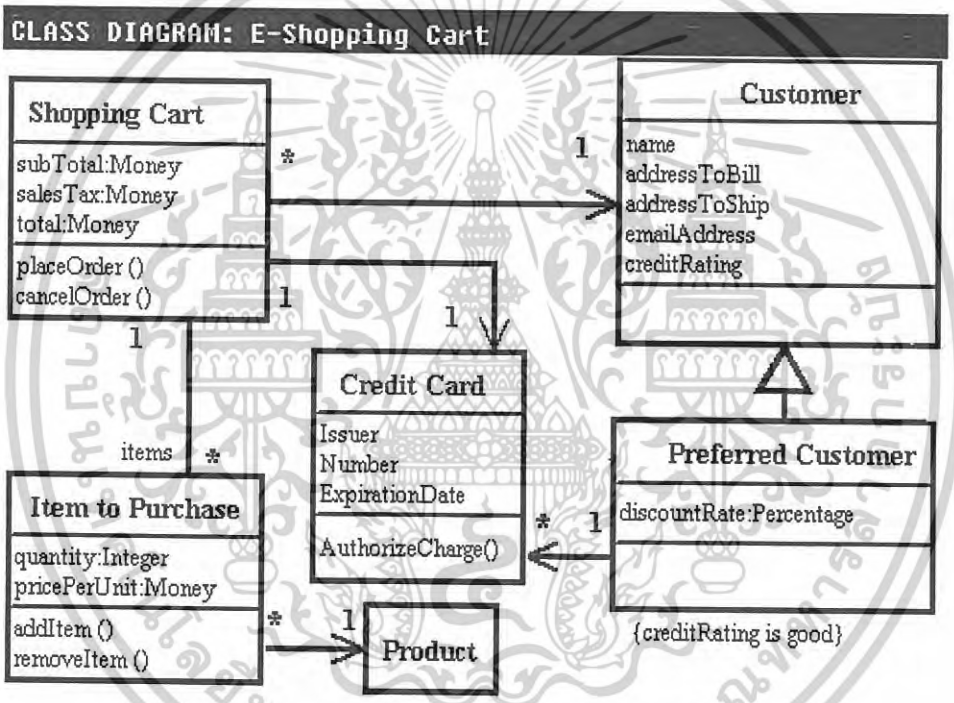
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 แผนภาพคลาส(Class Diagram)

คลาสไดอะแกรม คือ แผนภาพที่ใช้แสดง คลาสและ ความสัมพันธ์ระหว่าง คลาสของระบบที่สนใจ (Problem Domain) เช่น ในระบบจัดซื้อ คลาสที่เกี่ยวข้องคือ ผู้ผลิต, พนักงานจัดซื้อ, ใบสั่งซื้อ, ใบเสนอราคา, ใบเสร็จรับเงิน เป็นต้น ดังรูปที่ (2.23)

สัญลักษณ์ คลาสประกอบด้วย

- ชื่อของคลาส(Class Name)
- คุณลักษณะของคลาส(Attributes)
- การกระทำ(Operations หรือ Methods) คือ กิจกรรมที่สามารถกระทำกับออปเจ็กต์ นั้นๆ



รูปที่ 2.23 ตัวอย่างคลาสไดอะแกรมในระบบการซื้อของออนไลน์

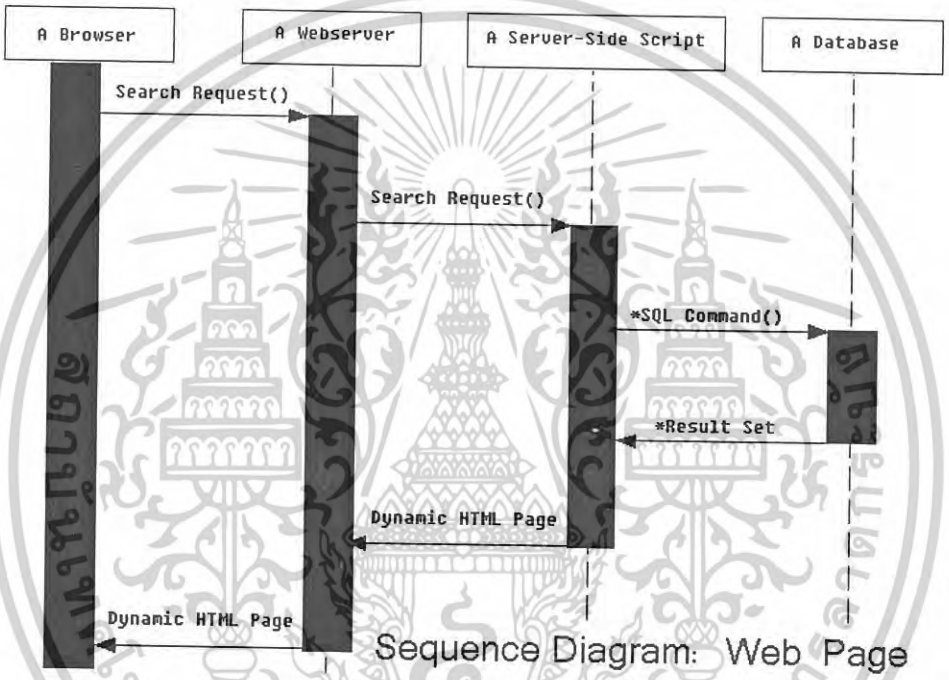
(thaiall.com. 2551: http://www.thaiall.com/uml/class01.gif.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 แผนภาพลำดับการทำงาน(Sequence Diagram)

ซีเควนซ์ไดอะแกรม เป็นแผนภาพที่ใช้อธิบายการทำงานของยูสเคสเพื่อแสดงถึงขั้นตอนการทำงานและลำดับของการสื่อสารระหว่างออปเจ็กต์ที่ตอบโต้กัน
ดังรูปที่ (2.24)

ซีเควนซ์ไดอะแกรม จะแสดงอยู่ในรูปแบบ 2 มิติ โดยเส้นประแนวตั้ง (Lifeline) จะนำเสนอในด้านเวลา ส่วนเส้นแนวนอน (Message) จะนำเสนอเกี่ยวกับการโต้ตอบกันระหว่าง ออปเจ็กต์หรือคลาสต่างๆ



รูปที่ 2.24 แสดงตัวอย่างของซีเควนซ์ไดอะแกรมการทำงานของเว็บเพจ

(thaiall.com. 2551: http://www.thaiall.com/uml/sequence01.gif.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เทคโนโลยีที่ใช้ในการพัฒนา

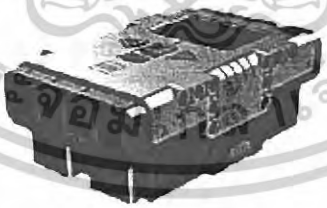
ในส่วนนี้จะกล่าวถึงเทคโนโลยีต่างๆที่ใช้ในการพัฒนาระบบ หุ่นยนต์จัดเก็บวัสดุ ซึ่งแบ่งออกเป็นสองส่วนคือส่วนของเทคโนโลยีด้านฮาร์ดแวร์ และ เทคโนโลยีด้านซอฟต์แวร์ที่นำมาใช้ประกอบการพัฒนา

2.6.1 เทคโนโลยีทางด้านฮาร์ดแวร์ (Hardware Technology)

ในการพัฒนาด้านฮาร์ดแวร์นั้นเทคโนโลยีที่จะนำมาใช้เพื่อพัฒนาหุ่นยนต์จัดเก็บวัสดุคือชุดทดลองเทคโนโลยีฝังตัว LEGO MINDSTORMS รุ่น NXT โดยเหตุผลที่เลือกใช้ผลิตภัณฑ์ของ LEGO เป็นเพราะมีความสอดคล้องกับเนื้อหาโดยรวมของโครงการ และตัวผลิตภัณฑ์มีความยืดหยุ่นต่อการประกอบหรือสร้างตัวหุ่นเป็นอย่างดี อีกทั้งยังในเรื่องขององค์ประกอบอื่นๆที่ทาง LEGO และหน่วยงานที่เกี่ยวข้องได้จัดหาไว้สำหรับนักพัฒนาโปรแกรม ไม่ว่าจะเป็นเรื่องของภาษาที่ใช้สนับสนุน อุปกรณ์ที่เกี่ยวข้อง หรือแม้แต่เครื่องมือที่ช่วยในการออกแบบ(Design Tools)ที่เหมาะสม ด้วยเหตุนี้ ชุดทดลองเทคโนโลยีฝังตัว LEGO MINDSTORMS รุ่น NXT จึงตรงกับความต้องการเบื้องต้นของโครงการ หุ่นยนต์จัดเก็บวัสดุ ที่ได้กล่าวมาข้างต้น ซึ่งในกรณีของผู้สนใจจะนำไปพัฒนาต่อ จำเป็นจะต้องคำนึงถึงองค์ประกอบทั้งหลายที่มีความสัมพันธ์และสอดคล้องกัน

ส่วนประกอบทางด้านฮาร์ดแวร์ของ LEGO MINDSTORMS NXT มีดังต่อไปนี้
(LEGO Group. 2552: http://mindstorms.LEGO.com/eng/Overview/default.aspx.)

หน่วยประมวลผลของ LEGO MINDSTORMS NXT



รูปที่ 2.25 หน่วยประมวลผลของ LEGO MINDSTORMS NXT
(LEGO Group. 2552: http://mindstorms.LEGO.com/eng/Overview/The_NXT.aspx.)

เปรียบเสมือนสมองหรือหัวใจหลักของหุ่นยนต์(ดังรูปที่ 2.25) NXT มีหน้าที่ทั้งทางด้านการประมวลผลและการควบคุม ประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตมอเตอร์(Motor ports) ช่องทางที่ใช้เชื่อมต่อกับโมดูลมอเตอร์

พอร์ตเซ็นเซอร์(Sensor ports) ช่องทางที่ใช้เชื่อมต่อกับโมดูลกลุ่มเซนเซอร์

พอร์ตยูเอสบี(USB port) ช่องทางที่ใช้เชื่อมต่อกับคอมพิวเตอร์

ลำโพง(Loudspeaker) เป็นอุปกรณ์ที่ทำหน้าที่ตอบสนองการทำงานของโปรแกรมด้วยเสียงเมื่อเริ่มการรันโปรแกรม

ปุ่มกดเพื่อรับอินพุต(NXT Buttons)

ปุ่มสีส้ม: เปิด,ตกลง,เริ่มการทำงาน

ปุ่มลูกศรสีเทาอ่อน:ใช้ในการเลือกเมนูต่างๆในตัวNXT

ปุ่มสีเทาเข้ม:ยกเลิก,กลับ

หน้าจอแสดงผล(NXT Display)

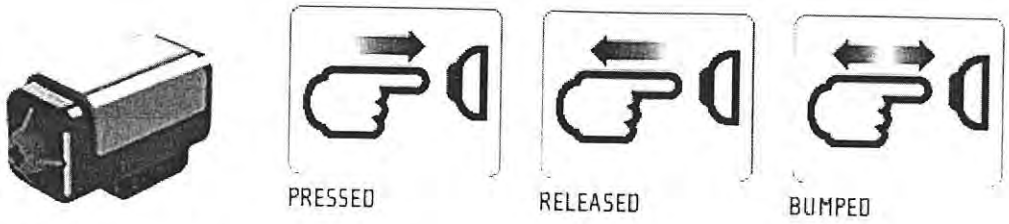
แสดงผลรายการต่างๆรวมประกอบด้วยการทำงานของโปรแกรมและเมนูที่ทำการเลือก ฯลฯ

รายละเอียดทางเทคนิค(Technical specifications)

- ไมโครคอนโทรลเลอร์ 32 บิต ตระกูล ARM7
- 256 กิโลไบต์ หน่วยความจำแฟลช(แบบไม่มีไฟเลี้ยง), 64 กิโลไบต์ RAM
- ไมโครคอนโทรลเลอร์ 8 บิต ตระกูล AVR
- 4 กิโลไบต์ หน่วยความจำแฟลช(แบบไม่มีไฟเลี้ยง), 512 กิโลไบต์ RAM
- บลูทูธ เวอร์ชัน 2.0
- พอร์ตยูเอสบี 12เมกกะบิต/วินาที
- 4 อินพุตพอร์ต , 6 สายเคเบิลสำหรับเชื่อมต่อ รองรับมาตรฐาน IEC 61158
- 3 เอาท์พุตพอร์ต, 6 สายเคเบิลสำหรับเชื่อมต่อ
- หน้าจอแอลซีดี ขนาด 100 x 64 พิกเซล
- ลำโพง – ความถี่เสียง 8 เฮิร์ต . ช่องสัญญาณเสียงขนาด 8 บิต และ ความละเอียดของเสียงอยู่ที่ 2-16 เฮิร์ต
- แหล่งพลังงาน 6 AA แบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็นเซอร์สัมผัส(Touch Sensor)



รูปที่ 2.26 เซ็นเซอร์สัมผัส

(LEGO Group. 2552: http://mindstorms.LEGO.com/eng/Overview/Touch_Sensor.aspx.)

เป็นเซ็นเซอร์ ที่ถูกออกแบบมาเพื่อให้รองรับการใช้งานประเภทสัมผัส หรือ กด เพื่อนำไปประมวลผลในการทำงานตามที่ได้กำหนดไว้ ต่อไป (ดังรูปที่ 2.26)

เซ็นเซอร์เสียง (Sound Sensor)

รูปที่ 2.27 เซ็นเซอร์เสียง

(LEGO Group. 2552: http://mindstorms.LEGO.com/eng/Overview/Sound_Sensor.aspx.)

เป็นเซ็นเซอร์ที่ถูกออกแบบมาเพื่อให้รองรับการใช้งานประเภทเสียง หรือการรับคลื่นความถี่เสียงโดยจะมี 2 โหมด หลัก คือ

DBA (detecting adjusted decibels) คือจะสามารถปรับแต่งการรับคลื่นเสียงให้อยู่ในรูปของ เสียงที่เราต้องการ ได้เช่น เสียงของมนุษย์หรือสัตว์ได้ยินในระดับหนึ่ง

DB (detecting standard [unadjusted] decibels) จะมีความแตกต่างกับ DBA คือจะสามารถรับความถี่เชิงปริมาณเพียงอย่างเดียว โดยวัดจากค่าเดซิเบล ตามความจริง นั่นหมายความว่าเสียงที่รับ อาจจะไม่อยู่ในคลื่นความถี่ที่สิ่งมีชีวิตเช่น คนหรือสัตว์ได้ยินก็เป็นได้
(ดังรูปที่ 2.27)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็นเซอร์แสง (Light Sensor)

รูปที่ 2.28 เซ็นเซอร์แสง

(LEGO Group. 2552: http://mindstorms.LEGO.com/eng/Overview/Light_Sensor.aspx.)

เป็นเซ็นเซอร์ที่ถูกออกแบบมาเพื่อใช้ในการแยกแยะการมองเห็น(Vision)ของNXT โดยจะแยกแยะออกผ่านทาง แสงของ ภาพที่เห็น (ดังรูปที่2.28) โดยรูปทางด้านขวามือจะแสดงให้เห็นถึงสีที่หุ่นยนต์สามารถรับรู้ได้

เซ็นเซอร์อัลตราโซนิคส์ (Ultrasonic Sensor)

รูปที่ 2.29 เซ็นเซอร์อัลตราโซนิคส์ (LEGO Group. 2552:

http://mindstorms.LEGO.com/eng/Overview/Ultrasonic_Sensor.aspx.)

เป็น เซ็นเซอร์ที่ใช้ในเกี่ยวข้องกับการมองเห็น คล้ายกับเซ็นเซอร์แสง แต่จะเป็นการจำแนกโดยระยะทางของวัตถุ ที่อยู่ห่างจากตัวเซ็นเซอร์โดยพิสัยของระยะจะรับได้ถึงประมาณ
0 - 225 เซนติเมตร (ดังรูปที่ 2.29)

เซอร์โวมอเตอร์ (Servo Motor)

รูปที่ 2.30 เซอร์โวมอเตอร์ (LEGO Group. 2552:

http://mindstorms.LEGO.com/eng/Overview/interactive_Servo_Motors.aspx.)

ทำหน้าที่เป็นมอเตอร์ในการหมุนซึ่งเป็น ปัจจัยสำคัญในการเคลื่อนที่โดยล้อของหุ่นยนต์ รวมไปถึงการคลื่นไหวในบางส่วน ซึ่ง เซอร์โวมอเตอร์นี้จะรวมไปถึงเซ็นเซอร์ควบคุมการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในตัวมันเองด้วย ทำให้สามารถที่จะรับ สถานะองศาของระนาบมอเตอร์ ณ เวลานั้นๆได้ ดังรูปที่ (2.30)

เนื่องจากระบบฝังตัวที่ใช้ในการทำโครงการในครั้งนี้เป็นระบบฝังตัวแบบสำเร็จรูป (ซึ่งรายละเอียดนั้นจะได้กล่าวถึงในส่วนของเทคโนโลยีที่เกี่ยวข้อง) ที่มีการพัฒนามาก่อนหน้านี้ จึงมีความได้เปรียบในเรื่องของความเสถียรของระบบ รวมถึงประสิทธิภาพในการทำงานต่างๆที่ได้รับการรับรองมาตรฐาน การนำระบบนี้ไปพัฒนาต่อในอนาคต จึงควรคำนึงถึงการเลือกใช้หรือการพัฒนาในส่วนของระบบฝังตัวให้มีความสอดคล้องกับระบบนั้นๆด้วย

ชุดทดลองเทคโนโลยีฝังตัว LEGO Mindstorms NXT ที่ถูกเลือกมาใช้งานในระบบนี้ ถูกจัดว่าเป็นระบบฝังตัวขนาดเล็ก เพราะประกอบด้วยหน่วยประมวลผลขนาดเล็กที่ใช้ควบคุมอุปกรณ์อิเล็กทรอนิคส์ โดยมีพื้นฐานการทำงานที่ครบถ้วน เนื่องจากประกอบไปด้วยหน่วยประมวลผลหน่วยควบคุมการทำงานการรับและส่งออกข้อมูลและหน่วยความจำแบบแฟลช รวมไปถึงเฟิร์มแวร์(Firmware) ที่เปรียบเสมือน ระบบปฏิบัติการ ของระบบแบบฝังตัวนั่นเอง

โดยในรูปแบบองค์ประกอบพื้นฐานของNXT ที่จะนำมาพัฒนาระบบของหุ่นยนต์จัดเก็บวัสดุนั้น จะมีคุณลักษณะที่เป็นโมดูลของส่วนประกอบต่างๆที่แยกออกจากกันโดยแบ่งตามหน้าที่การทำงานของแต่ส่วนอย่างชัดเจน ทำให้มีจุดเด่นในเรื่องของความแยกส่วนกันอย่างชัดเจน อันจะนำไปสู่การวิเคราะห์และออกแบบ(แยกแยะและแจกแจงระบบ) ออกจากกันเพื่อนำไปพัฒนาต่อในอนาคตโดยง่าย

2.6.2 เทคโนโลยีทางด้านซอฟต์แวร์ (Software Technology)

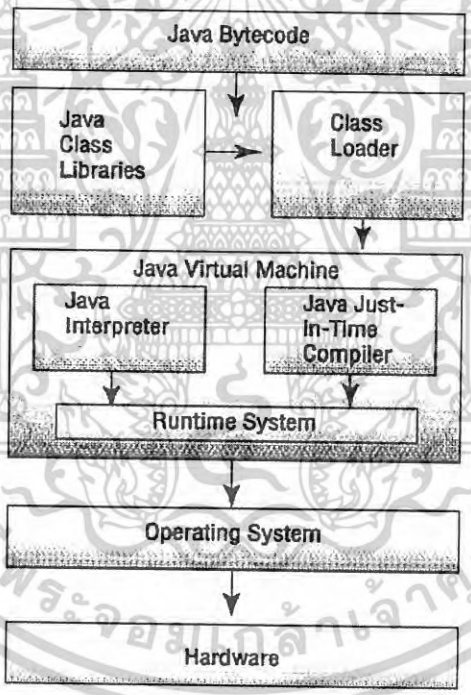
ในการพัฒนาหุ่นยนต์จัดเก็บวัสดุนั้น เทคโนโลยีด้านซอฟต์แวร์ที่นำมาใช้พัฒนาประกอบไปด้วย ไลบรารี LEJOS NXJ สำหรับการควบคุมชุดทดลองเทคโนโลยีฝังตัว LEGO MINDSTORMS รุ่น NXT และ Java Development Toolkit สำหรับเป็นเครื่องมือในการเขียนโปรแกรมเพื่อพัฒนาหุ่นยนต์

LEJOS(Java For LEGO MINDSTORMS) นับเป็นอีกเทคโนโลยีด้านภาษาโปรแกรมมิงที่ LEGO เลือกให้เข้ามามีบทบาททางด้านการพัฒนาโปรแกรมเพื่อควบคุม Mindstorms NXT ทั้งนี้ LEJOS ได้เข้ามามีส่วนร่วมตั้งแต่ Mindstorms RCX แล้ว และได้พัฒนาต่อเนื่องกันเรื่อยมาจนมาถึง NXJ เวอร์ชัน นี้ ซึ่งได้เลือกใช้ในโครงการนี้ เพราะภาษาจาวาเป็นพื้นฐานทางด้านการเขียนโปรแกรมเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบของไลบรารีของ LEJOS ดังรูปที่ (2.31-2.32)

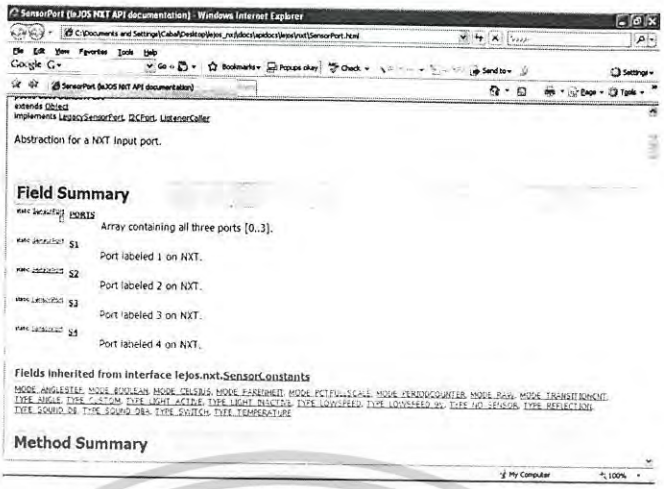
- สนับสนุนการเขียนโปรแกรมโดยใช้หลักการเชิงวัตถุ
- ความสามารถทางด้านการจัดการเกี่ยวกับการจองงานไม่ให้มีการจองทรัพยากรที่ซ้ำซ้อนกันหรือการจองแบบไม่ให้เกิดประโยชน์
- รองรับการใช้งานอาร์เรย์(ฟังก์ชันการเก็บค่าข้อมูลแบบ1ตัวแปรแต่เก็บได้หลายค่า)หลายมิติ
- สนับสนุนการใช้งานฟังก์ชันแบบเรียกตัวเอง(Recursive Function)
- การแสดงข้อผิดพลาดที่เกิดขึ้นขณะคอมไพล์โปรแกรมหรือรันโปรแกรม
- มีการใช้ ตัวแปรชนิดข้อมูล Java พื้นฐาน เช่น Float long String
- มีการใช้งาน java.lang java.util และ java.io ซึ่งเป็นคลาส พื้นฐาน ที่ผู้ใช้จาวาคุ้นเคย
- API ที่ประกอบด้วยการควบคุมการทำงานและคำสั่งพื้นฐาน



รูปที่ 2.31แสดงการทำงานของอินเตอร์พรีตเตอร์ในการใช้ไลบรารี

(coffeycountyks.org. 2552: http://www.coffeycountyks.org/Terms/2461HTML-30.gif.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

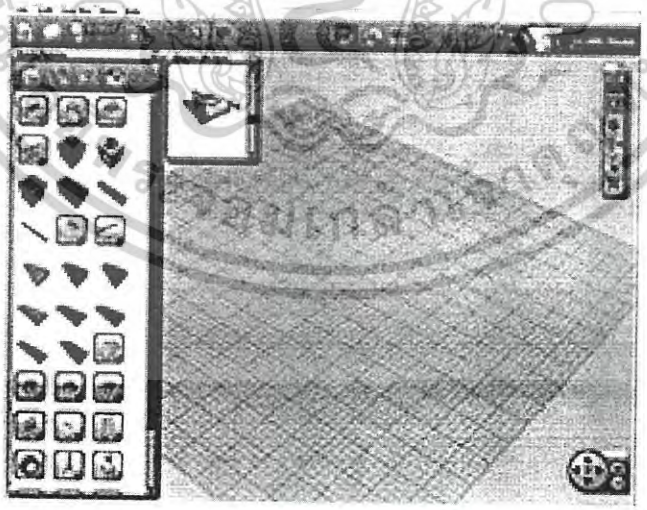


รูปที่ 2.32 ตัวอย่างของ API ของ ไลบรารี LEJOS

(LEJOS NXJ. 2551: http://lejos.sourceforge.net/nxt/nxj/api/index.html.)

2.6.3 เทคโนโลยีในการออกแบบฮาร์ดแวร์(Hardware Design)

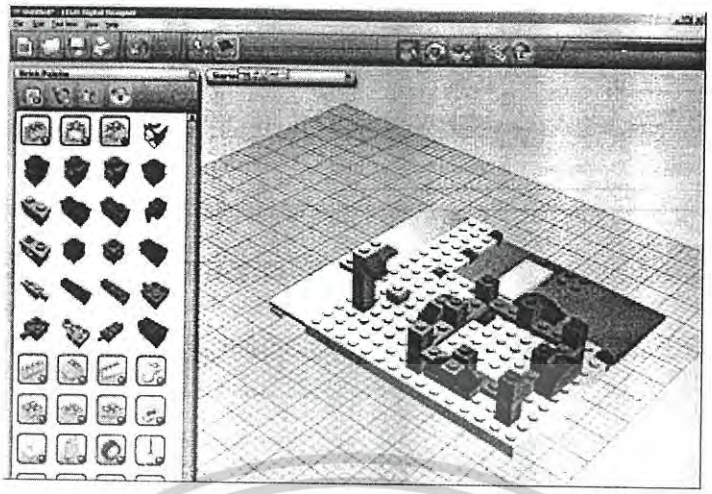
ในการออกแบบตัวหุ่นยนต์หรือการประกอบตัวหุ่นเพื่อนำไปพัฒนาระบบต่อไปนั้น ได้มีการเลือกใช้ผลิตภัณฑ์ซอฟต์แวร์สำหรับออกแบบตัวหุ่นยนต์โดยใช้โปรแกรมจำลองการออกแบบของทาง LEGO ที่ใช้ชื่อว่า "LEGO Digital Designer Version 2" (รูปที่ 2.33-2.34)มาช่วยในการออกแบบการประกอบตัวหุ่นเบื้องต้น เพื่อความสะดวกในการวางแผนหรือแม้แต่การเลือกใช้วัสดุอุปกรณ์ต่างๆและยังมีความได้เปรียบมากในเรื่องความรวดเร็วของการออกแบบ



รูปที่ 2.33 ตัวอย่างโปรแกรม LDD(LEGO Digital Designer)

(LEGO Digital Designer. 2552: http://ldd.lego.com/.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 ตัวอย่างโปรแกรม LDD(2)(LEGO Digital Desginer)
(LEGO Digital Designer. 2551: http://ldd.lego.com/.)

โดยตัวโปรแกรมมีตัวช่วยในการออกแบบ(Design Tools)มากมาย อาทิเช่น การเชื่อมต่อระหว่างชิ้นส่วนมีไกด์นาวิเกเตอร์คอยนำทางสำหรับผู้เริ่มต้น หรือมีพาเลตต์(Palette) ของส่วนประกอบใหม่ที่เพิ่มเข้ามาในเวอร์ชันใหม่ ซึ่งจะทำให้เพิ่มประสิทธิภาพในการประกอบหรือโมเดลแบบจำลองในการประกอบขึ้นพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบหุ่นยนต์จัดเก็บวัสดุ

จากวัตถุประสงค์ของโครงการที่ได้กล่าวถึงในบทที่ 1 ในบทนี้จะอธิบายถึงกระบวนการพัฒนาระบบ ตามทฤษฎีเกี่ยวกับขั้นตอนการออกแบบและพัฒนาระบบหุ่นยนต์จัดเก็บวัสดุโดยจะแบ่งออกเป็นสองส่วนคือซอฟต์แวร์และฮาร์ดแวร์ ซึ่งฮาร์ดแวร์ที่จะออกแบบสามารถทำตามขั้นตอนการทำงานต่างๆที่ได้มีการวางแผนไว้ในส่วนของการพัฒนาระบบ โดยเริ่มจากขั้นตอนการจัดการปัญหา หรือขั้นตอนของระบบ นักวิเคราะห์และออกแบบกระบวนการทำงานของระบบกับฮาร์ดแวร์ที่จะมารองรับ

3.1 การพัฒนาในส่วนของซอฟต์แวร์ (Software Development)

ในส่วนของการพัฒนาหุ่นยนต์ด้านซอฟต์แวร์นั้นจะนำเสนอในรูปแบบของทฤษฎีการพัฒนาระบบเชิงวัตถุโดยมี 4 ขั้นตอนคือ วิเคราะห์-ออกแบบ-พัฒนา-ทดสอบ ซึ่งจะมีแผนภาพแสดงการทำงานของหุ่นยนต์โดยใช้แผนภาพยูเอ็มแอลในการนำเสนอ ประกอบไปด้วย แผนภาพยูสเคส (Use Case Diagram) แผนภาพกิจกรรม(Activity Diagram) แผนภาพลำดับการทำงาน(Sequence Diagram) แผนภาพคลาส(Class Diagram) และแผนภาพสถานะ
(State Chart Diagram)

3.1.1 การวิเคราะห์ความต้องการของระบบ (Requirement Analysis)

การแบ่งความต้องการของระบบจะแบ่งหลักๆได้สองประเภท คือ หน้าที่ของระบบ (Functional) ซึ่งระบบจะต้องมีเพื่อให้ทำงานได้ และส่วนประกอบเพิ่มเติม(Supplementary) ที่จะทำให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ ดังนี้

3.1.1.1 หน้าที่ของระบบ

ระบบหุ่นยนต์จัดเก็บวัสดุ มีการทำงานบนฮาร์ดแวร์ที่เป็นลักษณะของหุ่นยนต์ ดังนั้นหน้าที่การทำส่วนใหญ่แล้วจะเป็นการทำงานที่ระบบตอบสนองกับฮาร์ดแวร์ ดังนี้

1. หุ่นยนต์จะต้องสามารถแยกแยะชนิดของวัสดุที่ได้รับ
2. หุ่นยนต์ต้องสามารถเดินจากจุดเริ่มต้นไปยังจุดสิ้นสุดได้อย่างถูกต้อง
3. หุ่นยนต์จะต้องไม่เดินออกนอกเส้นทางที่กำหนด
4. หุ่นยนต์จะต้องสามารถแยกแยะตำแหน่งและชนิดของจุดปล่อยวางสำหรับวางวัสดุ
5. หุ่นยนต์จะต้องสามารถวางวัสดุในจุดปล่อยวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 ส่วนประกอบเพิ่มเติม

หุ่นยนต์จัดเก็บวัสดุนั้น มีการทำงานบนฮาร์ดแวร์ที่เป็นลักษณะของหุ่นยนต์ ดังนั้น ส่วนประกอบของระบบจึงจะเป็นเรื่องของเงื่อนไขเพิ่มเติมที่สามารถเพิ่มประสิทธิภาพการทำงานในสภาพแวดล้อมการใช้งานจริง ดังนี้

- หุ่นยนต์สามารถหยุดเพื่อรอรับคำสั่งในการหาวัสดุที่ต้องการ
- หุ่นยนต์สามารถเคลื่อนที่ตามเส้นทางเพื่อหาวัสดุที่ต้องการ
- หุ่นยนต์สามารถจัดเรียงวัสดุในกรณีที่จุดวางมีหลายส่วน(พื้นที่ในแต่ละส่วนจำกัด)

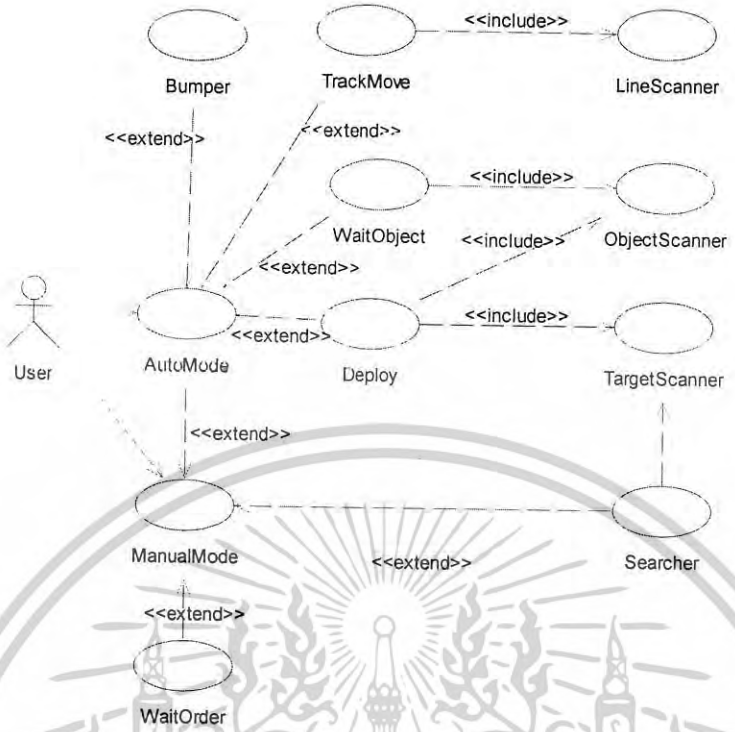
3.1.2 การออกแบบซอฟต์แวร์ (Software Design)

ในการออกแบบ จะอธิบายโดยใช้แผนภาพยูเอ็มแอลในการอธิบาย ซึ่งได้แก่ แผนภาพยูสเคส(Use Case Diagram) แผนภาพกิจกรรม(Activity Diagram) แผนภาพลำดับการทำงาน(Sequence Diagram) แผนภาพโครงสร้างคุณลักษณะ(Class Diagram) และ แผนภาพสถานะการทำงาน(State Chart Diagram)

3.1.2.1 แผนภาพขอบเขตของระบบ (UseCase Diagram)

เป็นแผนภาพที่แสดงถึงขอบเขตของระบบซึ่งจะบอกให้เห็นถึงหน้าที่ของระบบที่สามารถกระทำได้ โดยมี ออโตโหมดยูสเคส คือการทำงานหลักของระบบ(การนำวัสดุไปจัดเก็บ) ซึ่งจะมียูสเคสอื่นๆ เป็นองค์ประกอบย่อยเพื่อให้ระบบสามารถทำงานได้ตรงตามเป้าหมายที่กำหนดไว้ โดยมี ผู้เกี่ยวข้องกับระบบเป็นสิ่งที่อยู่ในสภาพแวดล้อม ซึ่งเป็นสิ่งที่ให้ข้อมูลกับระบบในการตัดสินใจการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงแผนภาพยูสเคสของระบบหุ่นยนต์จัดเก็บวัสดุ

ยูสเคสออโตโหมด(Automode)จะเป็นโหมดที่รวมเอายูสเคสอื่นๆเข้าไว้ด้วยกันในการทำงาน โดยจะมีการติดตามเส้นทาง(TrackMove) การรอรับวัสดุ(WaitObject) การวางวัสดุเมื่อพบเจอเป้าหมาย(Deploy)และการตรวจสอบการชน(Bumper) โดยการติดตามเส้นทางจะมีการเรียกใช้ยูสเคสการตรวจสอบเส้นทาง(LineScanner) การรอรับวัสดุมีการเรียกใช้ยูสเคสการตรวจสอบวัสดุ (ObjectScanner) การวางวัสดุเมื่อพบเจอเป้าหมาย(Deploy) มีการเรียกใช้ยูสเคสการตรวจสอบวัสดุและยูสเคสการตรวจสอบเป้าหมาย(TargetScanner) เพื่อเปรียบเทียบกัน โดยมีโหมดอีกรูปแบบหนึ่งคือมานวลโหมด(ManualMode)คือโหมดที่ไว้สำหรับรองรับคำสั่งจากผู้ใช้งานในกรณีที่ต้องการให้หุ่นยนต์เดินหาเป้าหมายที่ผู้ใช้เลือก ซึ่งมานวลโหมดจะใช้คุณสมบัติการสืบทอด (Inheritance) มาจากยูสเคสออโตโหมดและมีการเรียกใช้ยูสเคสเพิ่มอีก 2 ยูสเคสคือการรอรับคำสั่ง (WaitOrder)และการค้นหาเป้าหมายที่ตรงกับคำสั่ง(Searcher) (ดังรูปที่ 3.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถอธิบายการทำงานของแต่ละการทำงานของผู้ใช้ระบบ ได้ดังตารางที่ 3.1 – 3.11

ตารางที่ 3.1 Use Case AutoMode Description

Use case Name : AutoMode	**ID:** 01	**Importance Level :** Medium
Primary Actor : User	Use Case Type :	
Brief Description เป็นโหมดการทำงานมาตรฐานของหุ่นยนต์ โดยจะเรียกใช้งานระบบ		
Related Use Cases **Association :** **Include :** **Extend :** WaitObject , TrackMove , Deploy , Bumper **Generalization :**		
Normal flow 1. รอรับคำสั่งจากผู้ใช้ 2. ทำการเดินตามเส้นทาง 3. พบMarkerจะทำการตรวจสอบความหมายของMarker 4. ถ้าเป็นจุดปล่อยวัสดุที่ถูกต้อง จะทำการปล่อยวัสดุ 5. ตรวจสอบว่ามียังมีวัสดุอยู่อีกหรือไม่ 6. จบการทำงาน		
Sub flow 1. เมื่อได้รับคำสั่ง จึงจะเริ่มทำขั้นต่อไป 3. ถ้าMarkerกับวัสดุที่ได้รับไม่สัมพันธ์กัน จะไม่ทำอะไร และกลับไปทำ2ใหม่ 5. ถ้ามีวัสดุอยู่อีก จะกลับไปทำ2ใหม่		
Alternate/Exception Flow : 1. Bumperตรวจจับได้ว่ามีสิ่งกีดขวางอยู่ตรงหน้า 2. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 Use Case WaitObject Description

Use case Name : WaitObject	**ID:** 02	**Importance Level :** Medium
Primary Actor : User	Use Case Type :	
Brief Description ใช้พักการทำงานของหุ่นยนต์ จนกว่าจะได้รับวัสดุ		
Related Use Cases **Association :** **Include : ObjectScanner** **Extend :** **Generalization :**		
Normal flow 1. รอรับวัสดุ 2. เมื่อได้รับวัสดุจะทำการเปลี่ยนโหมดการทำงานเป็นโหมดปกติ		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Use Case Deploy Description

Use case Name : Deploy	**ID:** 03	**Importance Level :** High
Primary Actor : User	Use Case Type :	
Brief Description ใช้ในการควบคุม การตรวจสอบความตรงกันของMarkerและวัสดุ เพื่อใช้ในการปล่อยวัสดุ		
Related Use Cases **Association :** **Include :** ObjectScanner , TargetScanner **Extend :** **Generalization :**		
Normal flow 1. เมื่อพบMarker จะทำการอ่านค่าของMarkerและวัสดุที่ถืออยู่ 2. ถ้าเป็นจุดปล่อยที่ถูกต้อง จะทำการปล่อยวัสดุ		
Sub flow 2. ถ้าไม่ถูกต้อง จะเดินผ่านไปโดยไม่ทำอะไร		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 Use Case TrackMove Description

Use case Name : TrackMove	**ID:** 04	**Importance Level :** High
Primary Actor : User	Use Case Type :	
Brief Description ทำการตรวจสอบค่าของพื้น เพื่อใช้ในการควบคุมล้อ ให้วิ่งไปตามเส้นทางที่วางไว้		
Related Use Cases **Association :** **Include : LineScanner** **Extend :** **Generalization :**		
Normal flow 1. ตรวจสอบสีของพื้น 2. ทำการเคลื่อนที่ตามสีของพื้นที่พบ		
Sub flow 2a ถ้าอยู่บนเส้น จะเลี้ยวขวา 2b ถ้าอยู่กลางระหว่างเส้นและพื้นสนาม จะวิ่งเป็นเส้นตรง 2c ถ้าอยู่บนพื้นสนาม จะเลี้ยวซ้าย		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน 2. Bumperตรวจจับได้ว่ามีสิ่งกีดขวางอยู่ตรงหน้า		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 Use Case Bumper Description

Use case Name : Bumper	**ID:** 05	**Importance Level :** Low
Primary Actor : User	Use Case Type :	
Brief Description เป็นกันชนของตัวหุ่นยนต์ ใช้สำหรับหยุดการทำงานของหุ่นยนต์เมื่อสำผัสสิ่งกีดขวาง		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. เมื่อเกิดการชน 2. สั่งหยุดการเคลื่อนที่ของหุ่นยนต์ทันที และส่งสัญญาณเตือน 3. เมื่อสิ่งกีดขวางถูกนำออกไป 4. สั่งการทำงานทุกอย่างเป็นปกติ		
Sub flow		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 Use Case ObjectScanner Description

Use case Name : ObjectScanner	**ID:** 06	**Importance Level :** High
Primary Actor : User	Use Case Type :	
Brief Description ใช้ในอ่านค่าสีของวัสดุ และทำการตีความหมายค่าของสี เพื่อให้นำมาคำนวณทาง Logic ได้		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. ทำการอ่านค่าของสี 2. แปลค่าของสี จากค่าช่วงแกว่งของตัวเลข เป็นตัวหนังสือ		
Sub flow		
Alternate/Exception Flow :		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Use Case TargetScanner Description

Use case Name : TargerScanner	**ID:** 07	**Importance Level :** High
Primary Actor : User	Use Case Type :	
Brief Description ใช้ในอ่านค่าสีของMarker และทำการตีความหมายค่าของสี เพื่อให้นำมาคำนวณทาง Logic ได้		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. ทำการอ่านค่าของสี 2. แปลค่าของสี จากค่าช่วงแกว่งของตัวเลข เป็นตัวหนังสือ		
Sub flow		
Alternate/Exception Flow :		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 Use Case LineScanner Description

Use case Name : LineScanner	**ID:** 08	**Importance Level :** High
Primary Actor : User	Use Case Type :	
Brief Description ใช้ในอ่านค่าสีของLine และทำการตีความหมายค่าของสี เพื่อให้นำมาคำนวณทาง Logic ได้		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. ทำการอ่านค่าของสี 2. แปลค่าของสี จากค่าช่วงแกว่งของตัวเลข เป็นตัวหนังสือ		
Sub flow		
Alternate/Exception Flow :		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 Use Case ManualMode Description

Use case Name : ManualMode	**ID:** 09	**Importance Level :** Medium
Primary Actor : User	Use Case Type :	
Brief Description เป็นการทำงานคล้ายๆกับAutoModeแต่ผู้ใช้งานสามารถสั่งให้หุ่นยนต์ค้นหาMarkerที่ต้องการ เพื่อรอรับวัสดุจากผู้ใช้งานที่อยู่ที่Markerนั้นๆ เพื่อรับวัสดุสำหรับทำงานในAutoModeต่อไป		
Related Use Cases **Association :** **Include :** **Extend :** WaitOrder ,Searcher , AutoMode **Generalization :**		
Normal flow 1. ทำการรอรับคำสั่งจากผู้ใช้งาน 2. เดินตามเส้นทาง เพื่อค้นหาMarkerที่ผู้ใช้งานสั่ง 3. เมื่อเจอMarkerที่ถูกต้องตามที่สั่งไว้ จะหยุดรอรับวัสดุ และทำงานเช่นเดียวกับAutoMode		
Sub flow 3. ถ้าMarkerไม่ถูกต้อง จะทำการค้นหาMarkerอื่นๆต่อไป		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน 2. Bumperตรวจจับได้ว่ามีสิ่งกีดขวางอยู่ตรงหน้า		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 Use Case WaitOrder Description

Use case Name : WaitOrder	**ID:** 10	**Importance Level :** Medium
Primary Actor : User	Use Case Type :	
Brief Description ใช้สำหรับรับคำสั่งจากผู้ใช้โดยตรงผ่านทางเมนู โดยผู้ใช้จะทำการป้อนข้อมูลเป็นชื่อของMarkerที่ต้องการ เพื่อเป็นข้อมูลให้กับระบบ		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. ผู้ใช้กดปุ่มเปลี่ยนคำสั่ง 2. แสดงชื่อMarkerผ่านทางหน้าจอ 3. ผู้ใช้กดปุ่มENTER 4. ทำการบันทึกชื่อMarkerที่ผู้ใช้เลือก		
Sub flow 3. ผู้ใช้สามารถกดปุ่มเปลี่ยนคำสั่งได้หลายครั้ง ก่อนจะกดENTER		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 Use Case Searcher Description

Use case Name : Searcher	**ID:** 11	**Importance Level :** Medium
Primary Actor : User	Use Case Type :	
Brief Description ใช้สำหรับตรวจสอบMarkerที่ต้องกับข้อมูลที่มี และจะทำการหยุดรอคำสั่งอื่นๆต่อไป		
Related Use Cases **Association :** **Include :** **Extend :** **Generalization :**		
Normal flow 1. ตรวจพบMarkerที่ถูกต้อง 2. ทำการหยุดการเคลื่อนที่		
Sub flow 1. ถ้าไม่ถูกต้อง จะทำการหาMarkerอื่นๆต่อไป		
Alternate/Exception Flow : 1. กดปุ่มEscapeบนตัวหุ่นยนต์ เพื่อหยุดการทำงาน		

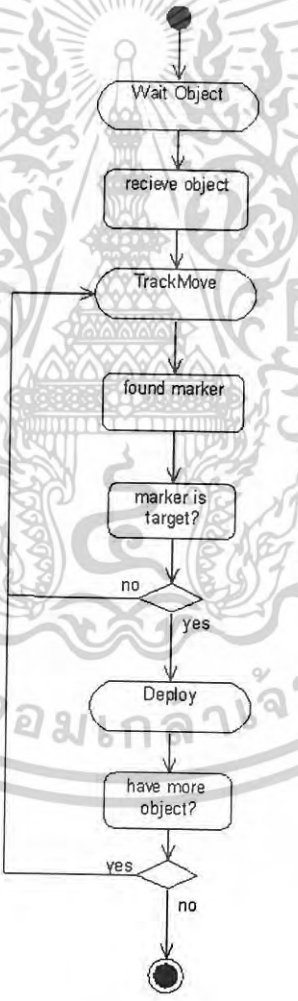
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 แผนภาพกิจกรรม (Activity Diagram)

เป็นแผนภาพที่แสดงถึงกิจกรรมของระบบ ซึ่งจะแสดงให้เห็นถึงกิจกรรมภายในการทำงาน ของระบบแต่ละอย่าง แสดงรายละเอียดของแผนภาพกิจกรรม

แผนภาพกิจกรรมออโตโหมด (AutoMode Activity Diagram)

เป็นกิจกรรมที่รวมกิจกรรมย่อยอื่นๆเข้ามาทำงานซึ่งเป็นตัวจัดลำดับการทำงานของกิจกรรมในระบบโดยเริ่มจากการรับวัสดุและทำการตรวจสอบ แล้วจึงทำการตรวจสอบเส้นทางเพื่อทำการเคลื่อนที่โดยระหว่างนี้ หากตรวจพบจุดปล่อยวางวัสดุจะทำการตรวจสอบจุดวางวัสดุนั้นเพื่อทำการวางจนกว่าจะพบจุดสิ้นสุดของการทำงาน (ดังรูปที่ 3.2)

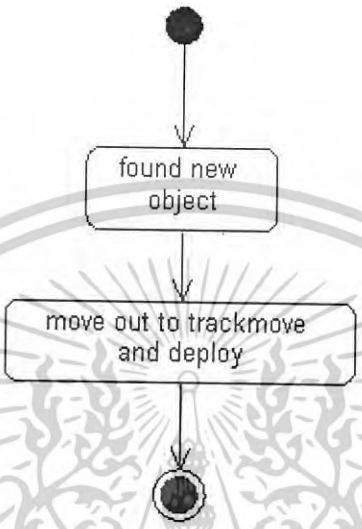


รูปที่ 3.2 แสดงแผนภาพกิจกรรมออโตโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการรอรับวัสดุ (WaitObject Activity Diagram)

เป็นกิจกรรมที่ใช้ในการเชื่อมต่อระหว่างการรอรับวัสดุเมื่อได้รับวัสดุแล้วก็จะทำการโอนหน้าที่ไปให้กับการติดตามเส้นทางและวางวัสดุ (ดังรูปที่ 3.3)

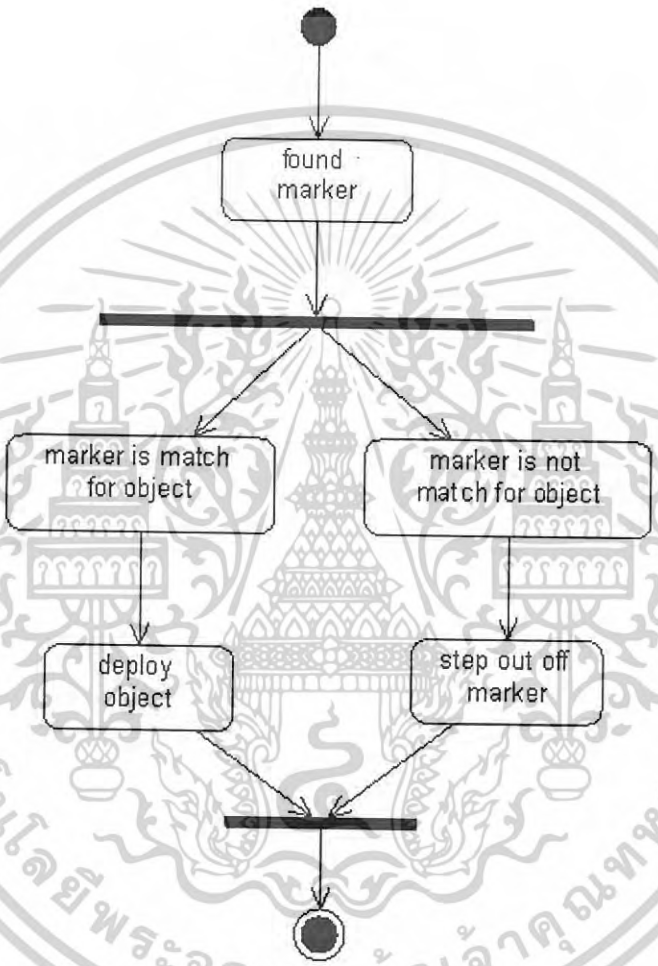


รูปที่3.3 แสดงแผนภาพกิจกรรมการรอรับวัสดุของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการวางวัสดุ (Deploy Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับกรณีที่เซ็นเซอร์ตรวจจับเป้าหมายพบการเปลี่ยนแปลงของสีที่ไม่ใช่สีพื้น จะทำการตรวจว่าเป็นสีของเป้าหมายที่ต้องการหรือไม่ถ้าใช่ก็จะทำการวางวัสดุ แต่หากไม่ใช่เป้าหมายที่ต้องการ จะทำการเดินออกจากสีของเป้าหมายดังกล่าวที (ดังรูปที่ 3.4)

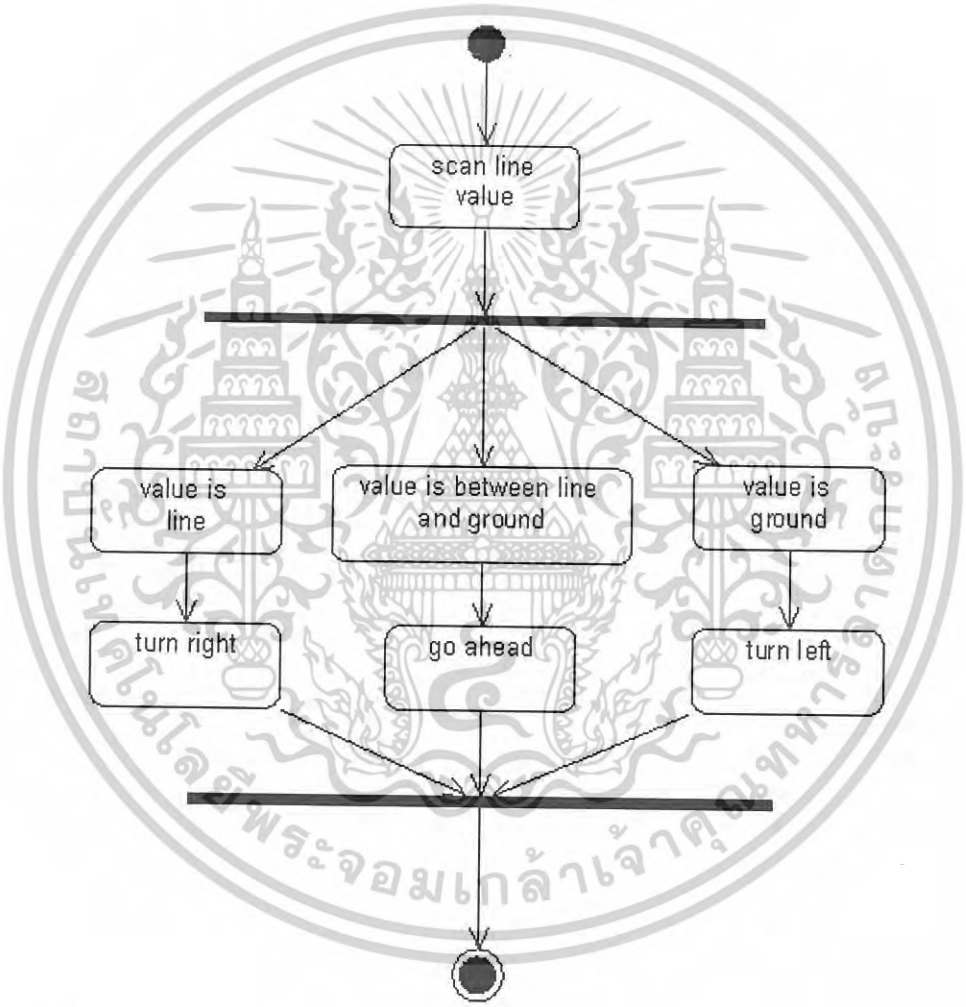


รูปที่ 3.4 แสดงแผนภาพกิจกรรมการวางวัสดุของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการติดตามเส้นทาง (TrackMove Acitvity Diagram)

เป็นกิจกรรมที่ใช้สำหรับการตรวจจับเส้นทางการเดินทางของหุ่นยนต์ โดยจะทำการตรวจสอบสีของเส้นทางเก็บเป็นข้อมูลดิจิตอลให้กับระบบ ซึ่งในสภาวะงาน เราจะใช้เป็นเส้นคู่สีตัดกันอย่างชัดเจน (ในที่นี้กำหนดให้เป็นสีของเส้นกับสีพื้นสนาม)จะทำการตรวจสอบค่าความสว่างของเส้นที่เซ็นเซอร์ตรวจจับเส้นทางตรวจจับได้ และทำการเปรียบเทียบหากว่าเป็นสีของเส้นสนามจะทำการเลี้ยวขวา แต่หากเป็นสีของพื้นสนามจะทำการเลี้ยวซ้าย (ดังรูปที่ 3.5)

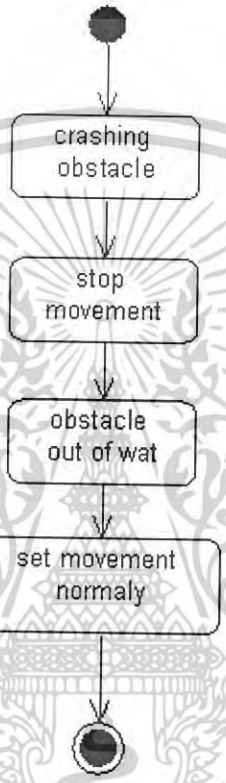


รูปที่ 3.5 แสดงแผนภาพกิจกรรมการติดตามเส้นทางของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการตรวจสอบการชน (Bumper Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับตรวจสอบการชนกันของตัวหุ่นยนต์กับวัสดุอื่นๆโดยเซ็นเซอร์สัมผัส โดยหากเกิดการชน เซ็นเซอร์สัมผัสที่อยู่ด้านหน้าของตัวหุ่นจะทำการรับรู้จะทำการหยุดการเคลื่อนที่ เมื่อมีการเคลื่อนย้ายวัสดุที่ขวางอยู่ออกก็จะทำการเดินตามปกติ (ดังรูปที่ 3.6)

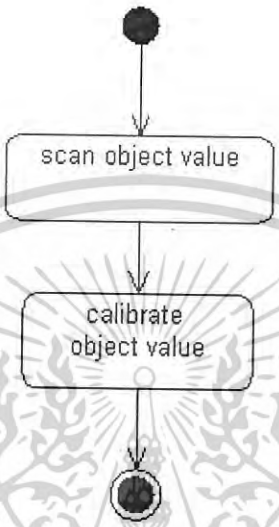


รูปที่ 3.6 แสดงแผนภาพกิจกรรมการตรวจสอบการชนของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการตรวจสอบชนิดวัสดุ (ObjectScanner Activity Diagram)

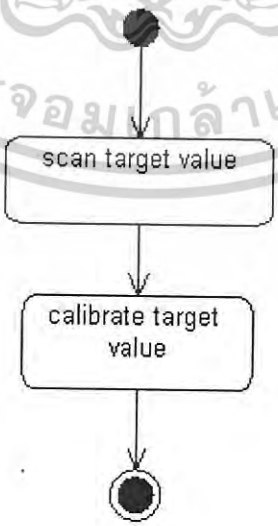
เป็นกิจกรรมที่ใช้สำหรับตรวจสอบค่าความสว่างของวัสดุโดยจะนำค่าที่ได้จากเซ็นเซอร์ตรวจสอบวัสดุมาทำการตรวจกับค่าที่ทำการวัดได้ว่าอยู่ในช่วงใดและกำหนดเป็นสีใด (ดังรูปที่ 3.7)



รูปที่ 3.7 แสดงแผนภาพกิจกรรมการตรวจสอบชนิดวัสดุของระบบหุ่นยนต์จัดเก็บวัสดุ

แผนภาพกิจกรรมการตรวจสอบชนิดเป้าหมาย (TargetScanner Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับตรวจสอบค่าความสว่างของเป้าหมายโดยจะนำค่าที่ได้จากเซ็นเซอร์ตรวจสอบเป้าหมายมาทำการตรวจกับค่าที่ทำการวัดได้ว่าอยู่ในช่วงใดและกำหนดเป็นสีใด (ดังรูปที่ 3.8)

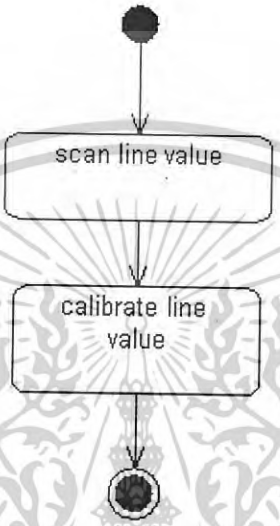


รูปที่ 3.8 แสดงแผนภาพกิจกรรมการตรวจสอบชนิดเป้าหมายของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการตรวจสอบชนิดเส้นทาง (LineScanner Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับตรวจสอบค่าความสว่างของเป้าหมายโดยจะนำค่าที่ได้จากเซ็นเซอร์ตรวจสอบเป้าหมายมาทำการตรวจกับค่าที่ทำการวัดได้ว่าอยู่ในช่วงใดและกำหนดเป็นสีใด (ดังรูปที่ 3.9)

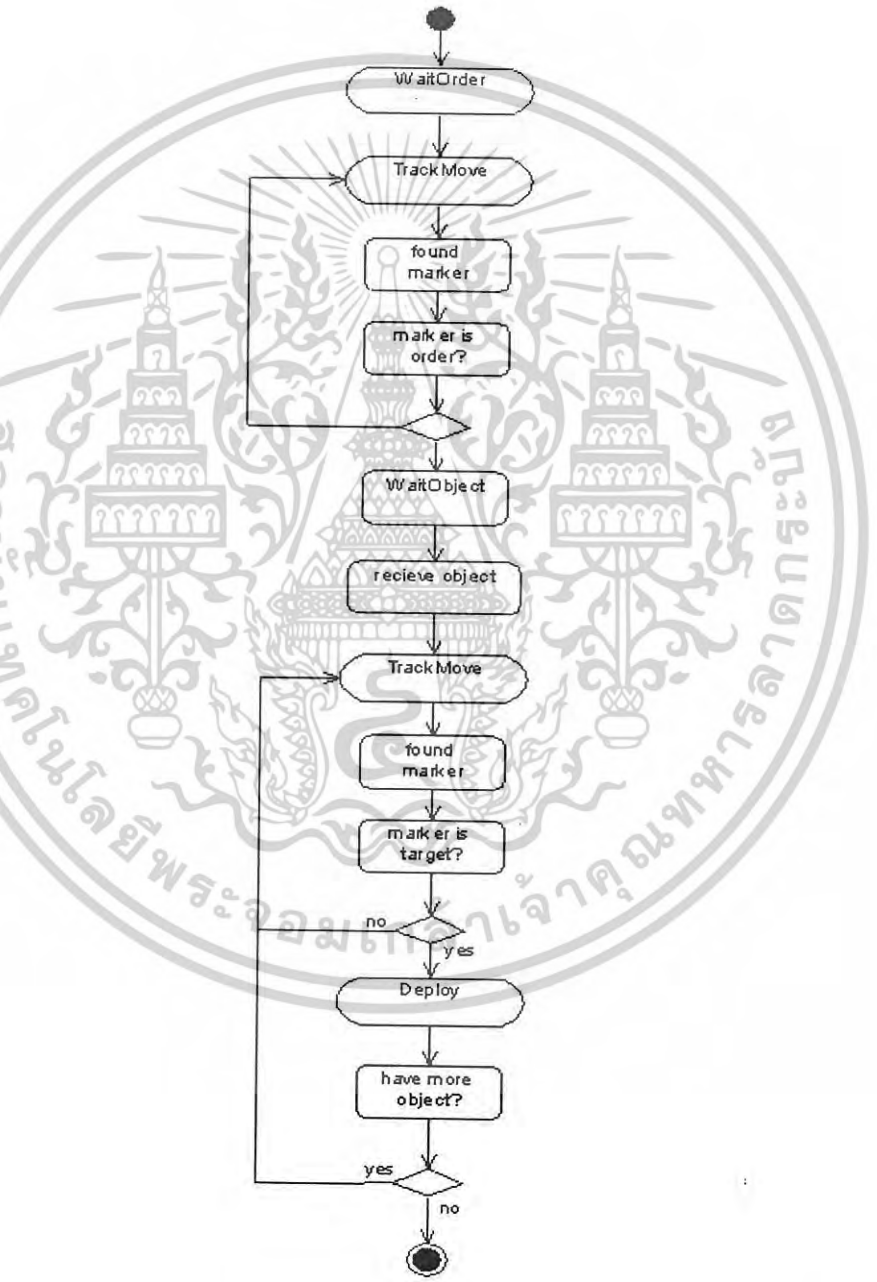


รูปที่ 3.9 แสดงแผนภาพกิจกรรมการตรวจสอบชนิดเส้นทางของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมมานวลโหมด (ManualMode Activity Diagram)

เป็นกิจกรรมที่เรียกใช้การรอรับคำสั่งจากผู้ใช้งานในการเลือกเป้าหมายที่ต้องการจะให้หุ่นยนต์ออกเดินหา โดยในกิจกรรมนี้จะทำการสืบทอดคุณสมบัติมาจากออโตโหมด แต่จะมีการเพิ่มการรอรับคำสั่งและการเปรียบเทียบค่าของเป้าหมายกับค่าของคำสั่งที่ได้รับ ซึ่งแตกต่างจากออโตโหมด (ดังรูปที่ 3.10)

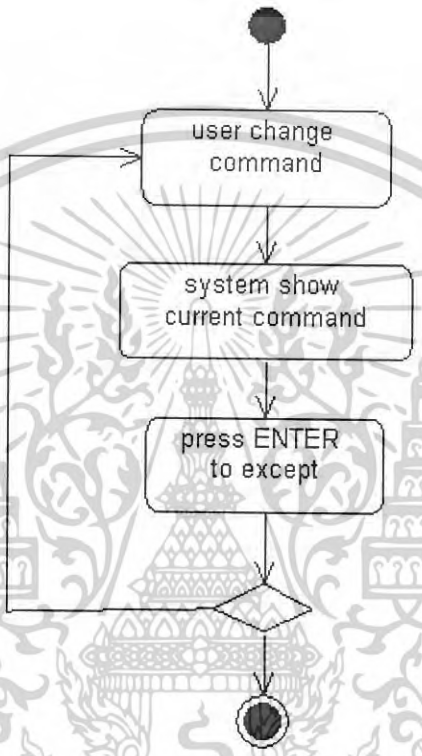


รูปที่ 3.10 แสดงแผนภาพกิจกรรมมานวลโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการรอรับคำสั่ง (WaitOrder Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับการรอรับคำสั่งจากผู้ใช้งาน ซึ่งกิจกรรมนี้ถูกเรียกใช้โดยกิจกรรมมานวลโหมด โดยจะแสดงผลเป็นตัวเลือกสีของเป้าหมายที่ผู้ใช้ต้องการและเมื่อทำการกดตกลงก็จะทำการนำค่านั้นไปดำเนินการต่อไป (ดังรูปที่ 3.11)

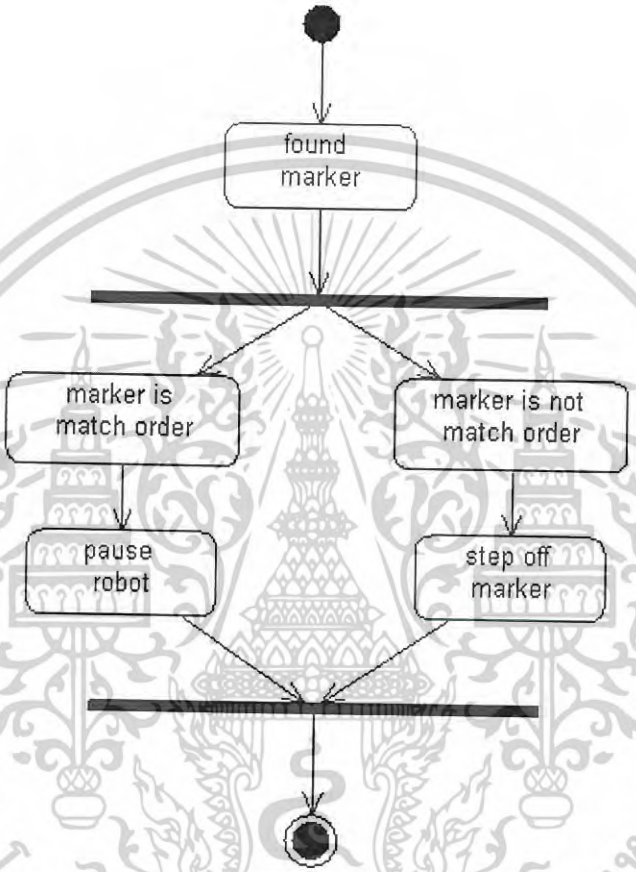


รูปที่ 3.11 แสดงแผนภาพกิจกรรมการรอรับคำสั่งของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพกิจกรรมการค้นหาเป้าหมาย (Searcher Activity Diagram)

เป็นกิจกรรมที่ใช้สำหรับการเปรียบเทียบค่าของเป้าหมายกับค่าของคำสั่งที่ผู้ใช้งานต้องการ ซึ่งถูกเก็บอยู่ในหน่วยความจำ หากมีค่าตรงกันก็จะทำการหยุดรอรับวัสดุ แต่หากไม่ตรงก็จะทำการเดินออกจากเป้าหมายนั้น (ดังรูปที่ 3.12)



รูปที่ 3.12 แสดงแผนภาพกิจกรรมการค้นหาเป้าหมายของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

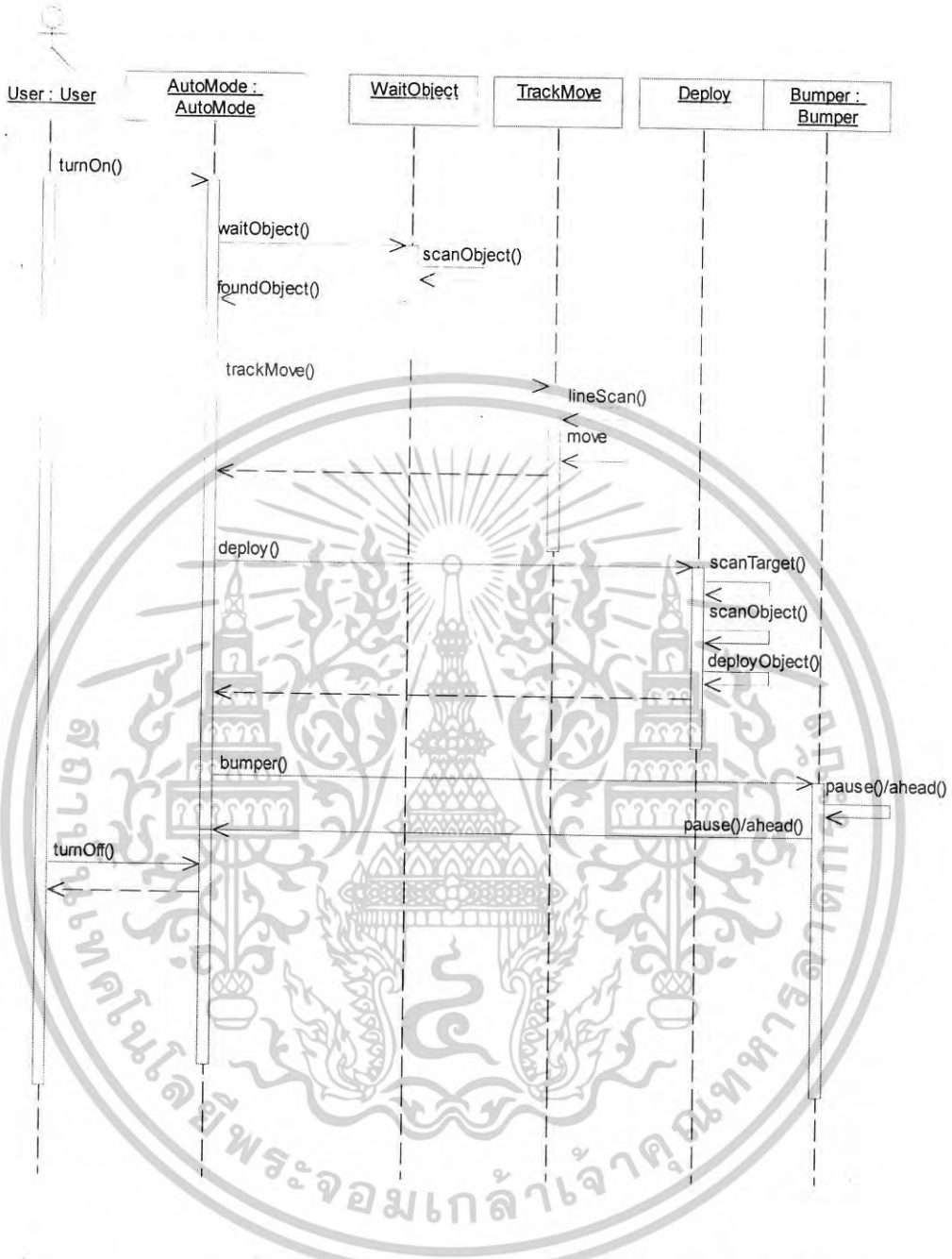
3.1.2.3 แผนภาพลำดับการทำงาน (Sequence Diagram)

คือแผนภาพที่ใช้แสดงการทำงานของระบบซึ่งแสดงให้เห็นถึงลำดับขั้นตอนการส่งเมสเสจภายในระบบโดยจะแสดงรายละเอียดของแผนภาพ

แผนภาพลำดับการทำงานออโตโหมด (AutoMode Sequence Diagram)

ขั้นตอนการงานของออโตโหมดคือเมื่อผู้ใช้งานทำการเปิดเครื่อง คลาสออโตโหมดจะทำการเรียกใช้ฟังก์ชันการการรอวัสดุ(WaitObject) โดยจะทำการสแกนวัสดุอยู่ตลอดเวลา เมื่อเจอวัสดุจะทำการส่งค่าวัสดุนั้นกลับมาให้คลาสออโตโหมด และคลาสออโตโหมดจะทำการเรียกใช้ฟังก์ชันการติดตามเส้นทาง(TrackMove)โดยฟังก์ชันนี้จะทำการสแกนเส้นทางและทำการเคลื่อนที่ตามเส้นทางและเมื่อทำการตอบสนองการสั่งงานแล้วจะทำการส่งสัญญาณกับมา ออโตโหมดจะทำการเรียกใช้ฟังก์ชันการวางวัสดุ(Deploy) ซึ่งจะมีการทำงานอยู่ 3 อย่างคือการสแกนเป้าหมาย สแกนวัสดุ ทำการเปรียบเทียบระหว่างเป้าหมายกับวัสดุหากมีค่าตรงกันก็จะทำการวางวัสดุและส่งสัญญาณกลับมา ออโตโหมดจะทำการเรียกใช้ฟังก์ชันตรวจสอบการชน(Bumper)โดยจะทำการตรวจว่าตัวหุ่นยนต์มีการชนกับวัสดุอื่นหรือไม่บนเส้นทางหากชนก็จะทำการหยุด และเมื่อมีการเคลื่อนย้ายออกนอกเส้นทางแล้วก็จะทำการเดินต่อไป จนกว่าจะทำการปิดเครื่อง(ดังรูปที่ 3.13)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

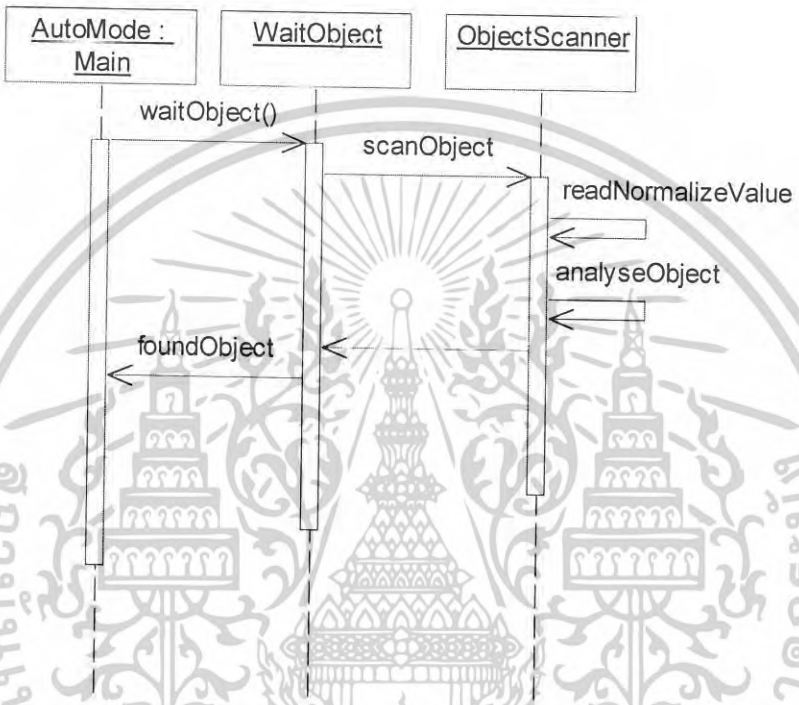


รูปที่ 3.13 แสดงแผนภาพขั้นตอนการทำงานออโตโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพลำดับการทำงานการรอรับวัสดุ (WaitObject Sequence Diagram)

ขั้นตอนการทำงานของการรอรับวัสดุ(WaitObject) คือจะมีกรถูกเรียกใช้โดยออโตโหมด ซึ่งฟังก์ชัน WaitObject จะทำการเรียกใช้ฟังก์ชัน ObjectScanner คือการสแกนค่าของวัสดุด้วยเมธอด readNormalizeValue และทำการตรวจสอบค่าที่ได้ว่าอยู่ในช่วงใดจากนั้นจึงส่งค่ากลับให้แก่ฟังก์ชัน WaitObject และออโตโหมดตามลำดับ (ดังรูปที่ 3.14)

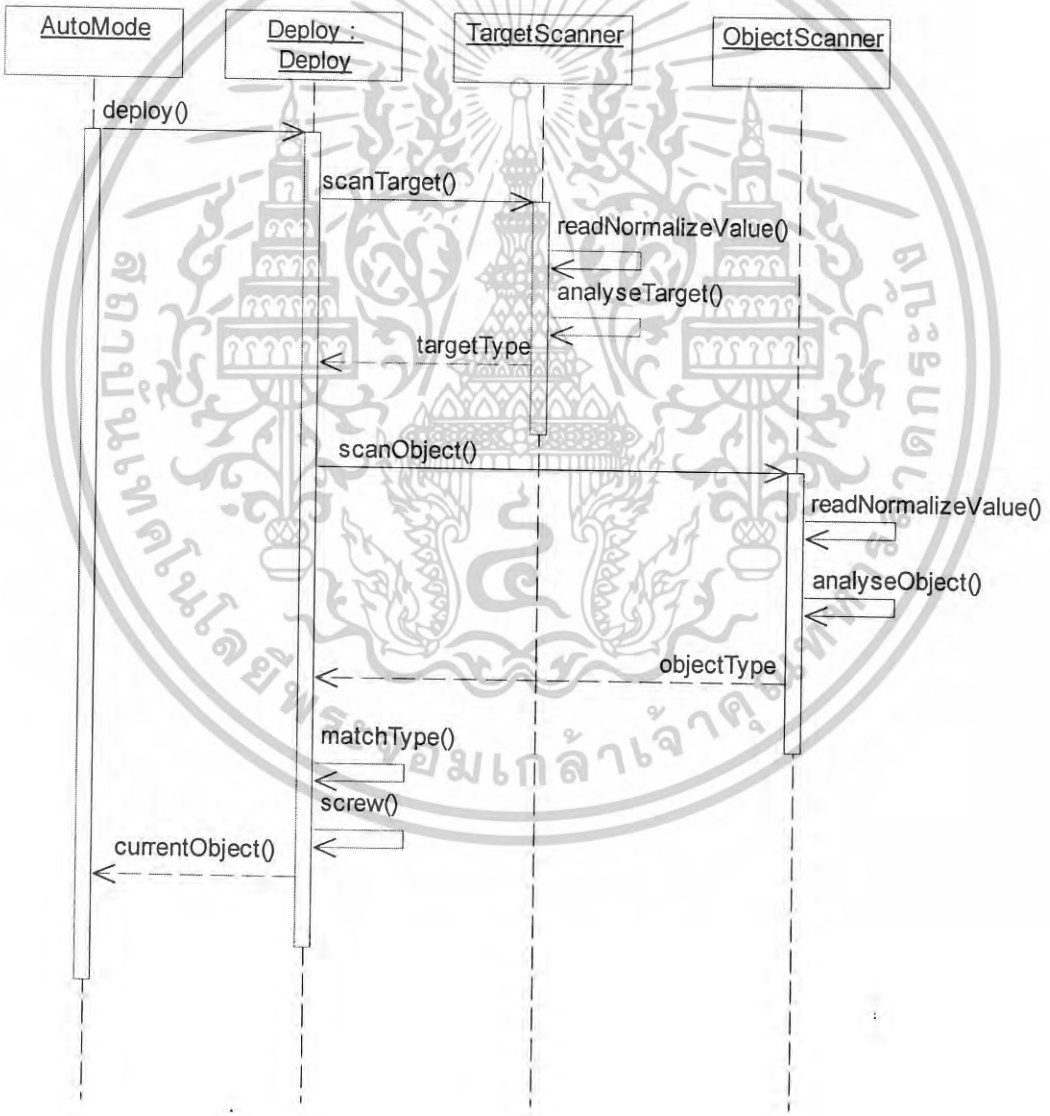


รูปที่ 3.14 แสดงแผนภาพขั้นตอนการทำงานการรอรับวัสดุของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพลำดับการทำงานการวางวัสดุ (Deploy Sequence Diagram)

แสดงถึงขั้นตอนการทำงานของการวางวัสดุ โดยเมื่อออโตโหมดทำการเรียกใช้ฟังก์ชันการวางวัสดุ(Deploy) ก็จะทำการเรียกฟังก์ชัน TargetScanner และ ObjectScanner โดย TargetScanner จะทำการเรียกเมธอด readNormalizeValue และทำการตรวจสอบค่าของเป้าหมายที่ได้ว่าอยู่ในช่วงใดจากนั้นจึงส่งค่ากลับให้แก่ฟังก์ชัน Deploy ส่วนObjectScannerก็เช่นเดียวกัน คือจะทำการตรวจสอบค่าของวัสดุว่าอยู่ในช่วงใดจากนั้นจึงส่งค่ากลับให้แก่ฟังก์ชัน Deploy เมื่อฟังก์ชัน Deploy ได้ค่าทั้งสองมาก็จะทำการเปรียบเทียบว่าตรงกันหรือไม่หากตรงกันก็จะทำการปล่อยวางวัสดุและตอบสนองการทำงานกลับสู่ออโตโหมดต่อไป (ดังรูปที่ 3.15)

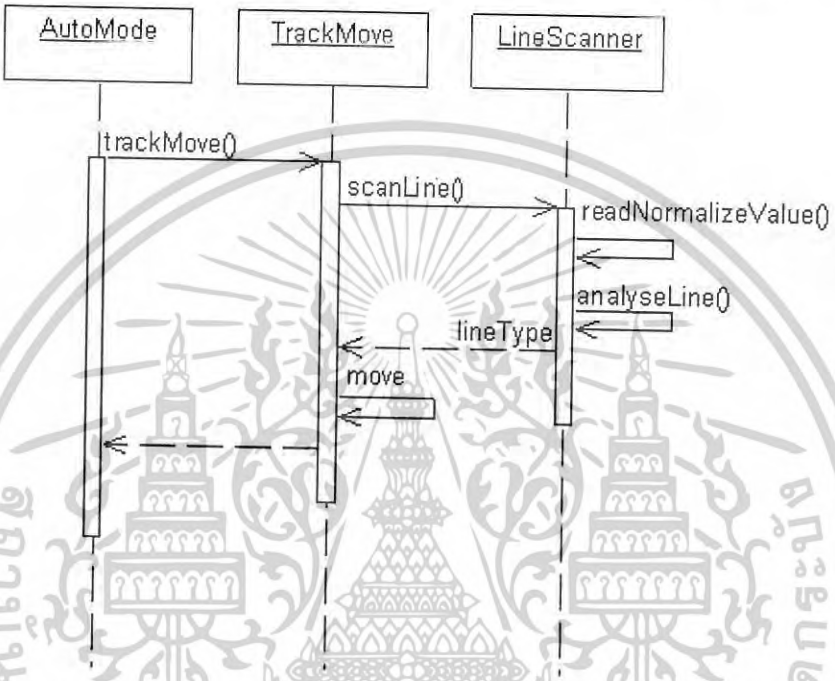


รูปที่ 3.15 แสดงแผนภาพขั้นตอนการทำงานการวางวัสดุของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพลำดับการทำงานการติดตามเส้นทาง (TrackMove Sequence Diagram)

แสดงถึงขั้นตอนการทำงานของการติดตามเส้นทาง โดยฟังก์ชันการติดตามเส้นทางจะถูกเรียกใช้โดยคลาสออโตโหมด และฟังก์ชัน TrackMove จะทำการเรียกฟังก์ชันLineScanner คือกาสแกนค่าของเส้นทางด้วยเมธอด readNormalizeValue และทำการตรวจสอบค่าที่ได้ว่าอยู่ในช่วงใด จากนั้นจึงส่งค่ากลับให้แก่ฟังก์ชัน TrackMove และ ออโตโหมด ตามลำดับ (ดังรูปที่ 3.16)

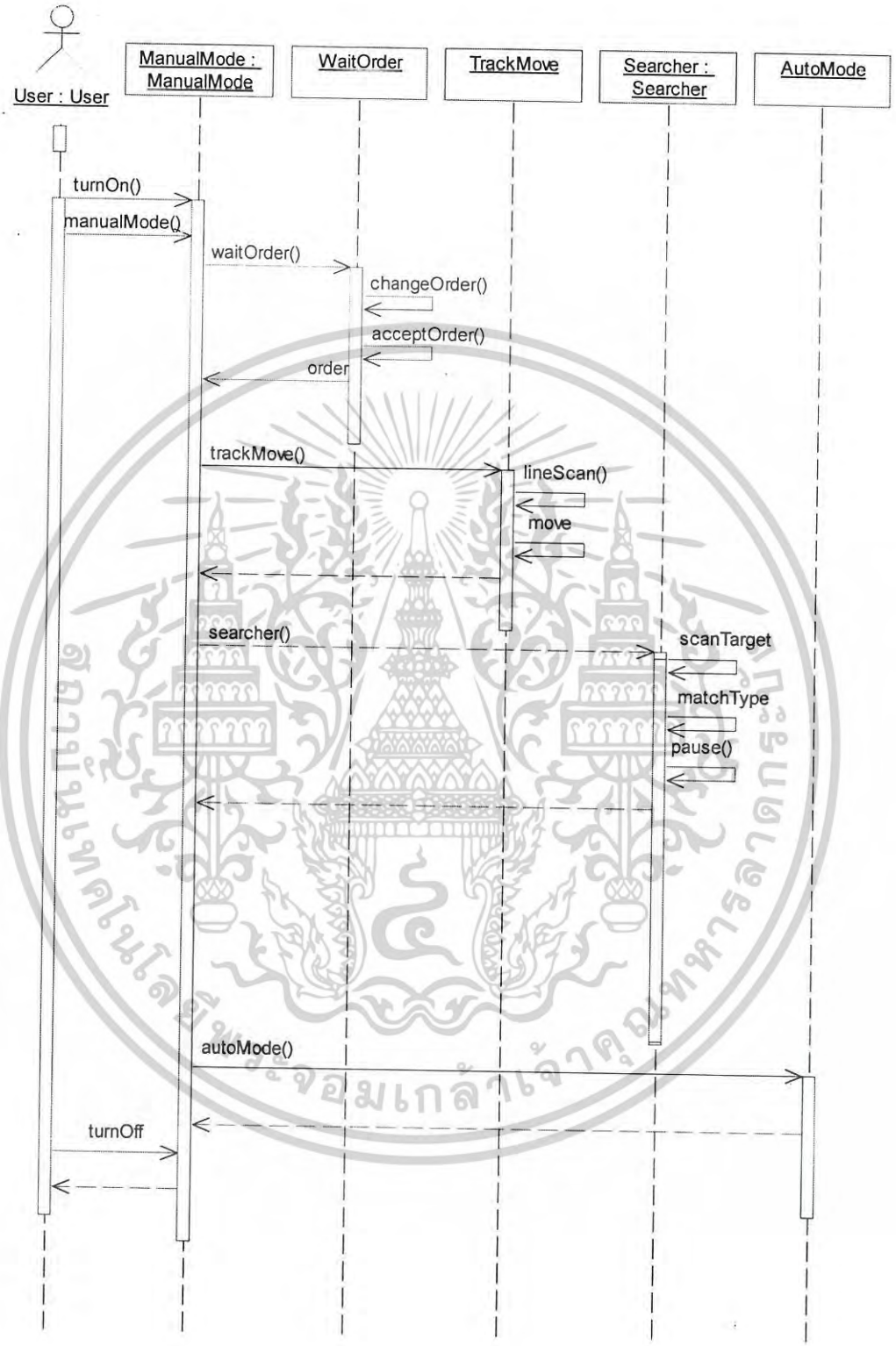


รูปที่ 3.16 แสดงแผนภาพขั้นตอนการทำงานการติดตามเส้นทางของระบบหุ่นยนต์จัดเก็บวัสดุ

แผนภาพลำดับการทำงานมานวลโหมด (ManualMode Sequence Diagram)

แสดงถึงขั้นตอนการทำงานของคลาสมานวลโหมด(ManualMode) โดยเริ่มแรกทำการเปิดเครื่องคลาสมานวลโหมดจะทำการเรียกฟังก์ชัน รอรับคำสั่ง(Waitorder)ซึ่งจะเป็นการให้ผู้ใช้งานทำการเลือกเป้าหมายที่ต้องการจากหน้าจอของหุ่นยนต์และเมื่อกดตกลง ข้อมูลที่ถูกเลือกจะถูกส่งกลับมายังคลาสมานวลโหมด ขั้นตอนต่อไปคลาสมานวลโหมดจะทำการเรียกฟังกชันการติดตามเส้นทาง(TrackMove)ให้ทำการสแกนเส้นและเดินตามเส้น และคลาสมานวลโหมดจะทำการเรียกใช้ฟังก์ชัน Searcher เพื่อทำการสแกนเป้าหมายและเปรียบเทียบกับค่าของเป้าหมายที่ถูกเลือกโดยผู้ใช้งานซึ่งถูกเก็บในหน่วยความจำ ซึ่งหากตรงกันก็จะทำการหยุดรอรับวัสดุ หลังจากนั้นการทำงานก็จะไปสู่โหมดของออโตโหมดคือรับส่งวัสดุตามปกติ (ดังรูปที่ 3.17)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

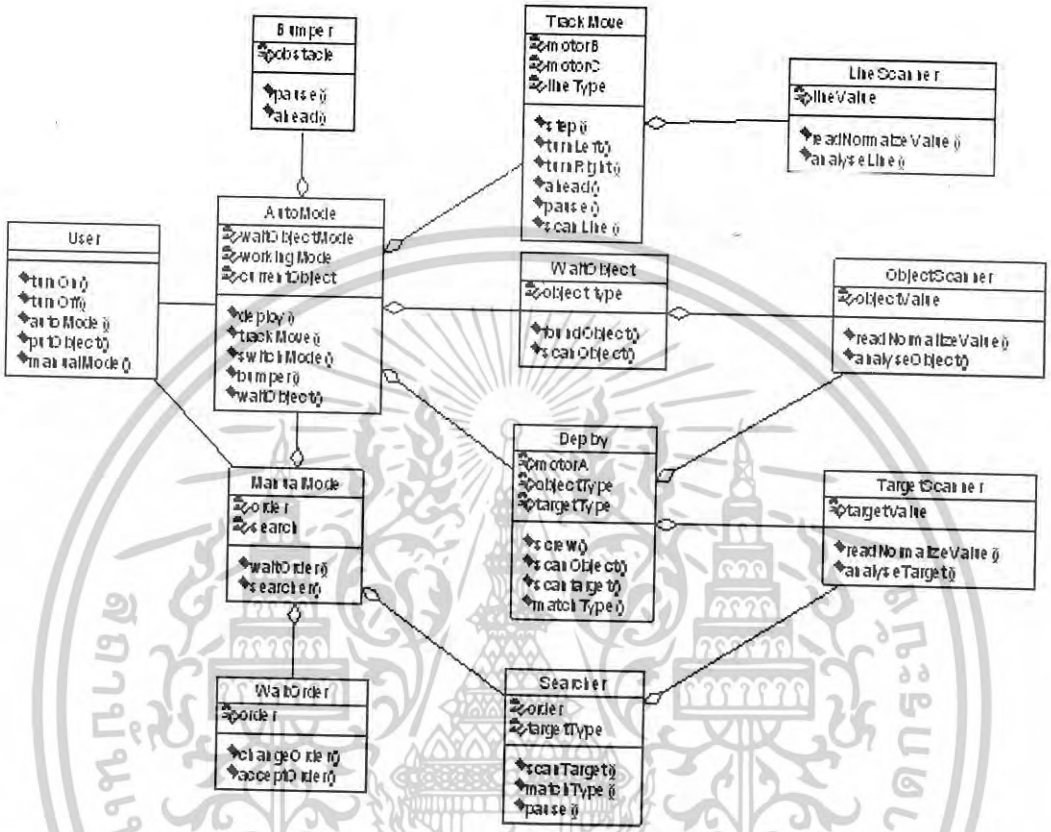


รูปที่ 3.17 แสดงแผนภาพขั้นตอนการทำงานมานวลโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.4 แผนภาพคลาส (Class Diagram)

จากแผนภาพการทำงานของผู้ใช้ระบบ ทำให้เห็นส่วนการทำงานของระบบทั้งหมด สามารถนำมาสร้างเป็นแผนภาพคลาส เพื่อใช้ในขั้นตอนการพัฒนาระบบ



รูปที่ 3.18 แสดงโครงสร้างของคลาสไดอะแกรมของระบบหุ่นยนต์จัดเก็บวัสดุ

จากแผนภาพคลาสไดอะแกรมจะสามารถแบ่งประเภทของคลาสออกได้เป็น 4 เลเยอร์คือ ส่วนของผู้ใช้งานคือคลาส User เป็นเลเยอร์ที่ใหญ่ที่สุด ถัดมาคือโหมดแบ่งออกเป็นสองโหมดคือ ออโตโหมด(AutoMode) และมานวลโหมด(ManualMode) เป็นเลเยอร์ที่สองซึ่งออโตโหมดจะมีความสัมพันธ์กับคลาสการติดตามเส้นทาง(TrackMove) การวางวัสดุ(Deployer) การรอรับวัสดุ (WaitObject)และการตรวจสอบการชน(Bumper) ส่วนของมานวลโหมดจะติดต่อกับคลาสการรอรับคำสั่ง(WaitOrder) และการค้นหาเป้าหมาย(Searcher) รวมถึงการติดตามเส้นทางและการวางวัสดุด้วย คลาสในเลเยอร์สุดท้ายคือคลาส การสแกนเส้นทาง(LineScanner) การสแกนวัสดุ (ObjectScanner) และการสแกนเป้าหมาย(TargetScanner) ซึ่งจะถูกเรียกใช้โดยคลาสในเลเยอร์ที่สามคือ TrackMove WaitObject Deploy และ Searcher (ดังรูปที่ 3.18)

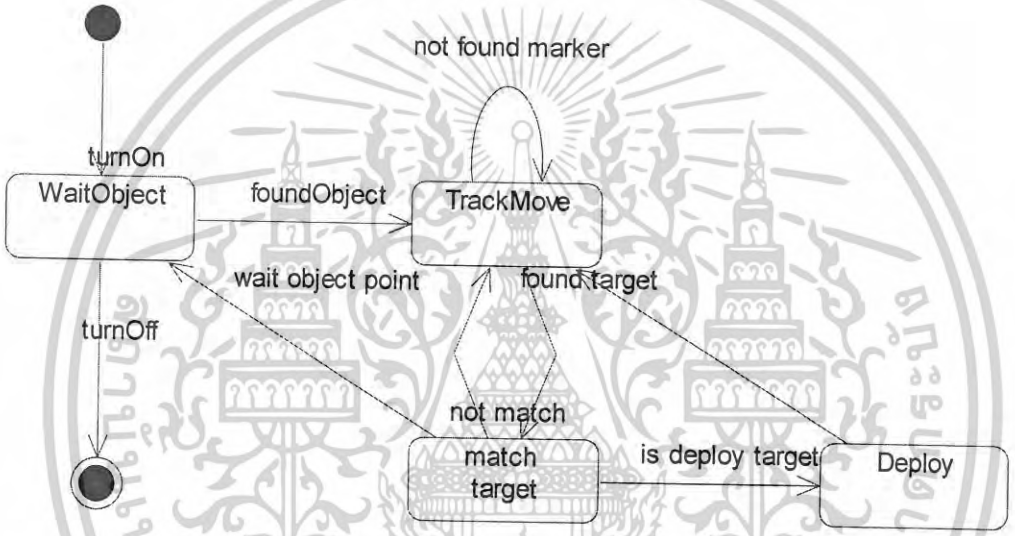
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.5 แผนภาพสถานะ (State Chart Diagram)

เป็นแผนภาพที่ใช้แสดงสถานการณ์ทำงานของออปเจ็กต์ต่างๆในระบบ มีดังนี้

แผนภาพสถานะออโตโหมด (AutoMode StateChart Diagram)

จากแผนภาพ แสดงถึงลำดับการทำงานจากสถานะตั้งต้น ที่รอรับวัสดุอยู่ เมื่อได้รับวัสดุ จะทำการวิเคราะห์ชนิดของวัสดุ แล้วจึงไปเข้าสถานะของการหาเส้นทางและเคลื่อนที่ จนกว่าจะจบการทำงาน ซึ่งในระหว่างนี้ หากพบว่าเจอจุดวาง จะไปเข้าสู่สถานะการตรวจสอบวัสดุและการวางวัสดุ แล้วก็จะกลับสู่การค้นหาเส้นทางและการเคลื่อนที่ต่อไป (ดังรูปที่ 3.19)



รูปที่ 3.19 แสดงแผนภาพสถานะออโตโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพสถานะมานวลโหมด (ManualMode StateChart Diagram)

จากแผนภาพ แสดงถึงลำดับการทำงานจากสถานะตั้งต้น ที่รอรับคำสั่งอยู่ เมื่อได้รับคำสั่งจากผู้ใช้ จึงไปเข้าสถานะของการหาเส้นทางและเคลื่อนที่ จนกว่าจะจบการทำงาน ซึ่งในระหว่างนี้หากพบว่าเจอจุดวาง จะไปเข้าสู่สถานการณ์ตรวจสอบคำสั่งที่ได้รับกับเป้าหมายและทำการวางวัสดุ แล้วก็จะกลับสู่การค้นหาเส้นทางและการเคลื่อนที่ต่อไป (ดังรูปที่ 3.20)

รูปที่ 3.20 แสดงแผนภาพสถานะมานวลโหมดของระบบหุ่นยนต์จัดเก็บวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

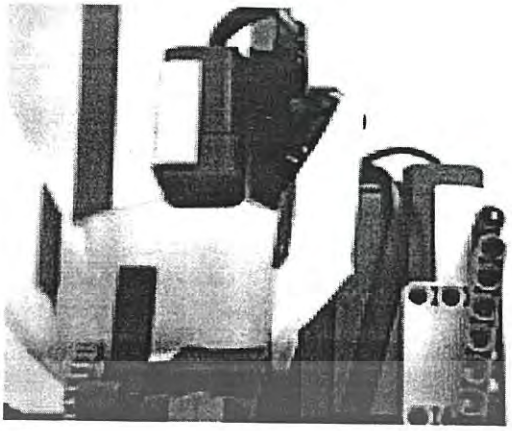
3.2 การออกแบบในส่วนของฮาร์ดแวร์ (Hardware Design)

ในส่วนของการออกแบบฮาร์ดแวร์ ได้มีรูปแบบการประกอบ และแนวคิดในการออกแบบในแต่ละโมดูลตามวัตถุประสงค์ซึ่งได้มาจากการวิเคราะห์ความต้องการ ดังนี้

ตารางที่ 3.12 การออกแบบฮาร์ดแวร์ตามวัตถุประสงค์

วัตถุประสงค์	การออกแบบ
เซ็นเซอร์ตรวจจับวัสดุ (ดังรูปที่ 3.21)	ออกแบบให้อยู่ติดกับภาชนะที่รับวัสดุ (อยู่ส่วนบนของตัวหุ่นยนต์)
เซ็นเซอร์ตรวจจับเป้าหมาย (ดังรูปที่ 3.22)	ออกแบบให้อยู่ข้างลำตัวหุ่นยนต์ใกล้กับพื้น เพื่อให้สามารถเห็นมาร์คเกอร์เป้าหมาย
เซ็นเซอร์ตรวจจับเส้นทาง (ดังรูปที่ 3.23)	ออกแบบให้อยู่ตรงกลางระหว่างล้อทั้งสองข้าง เพื่อให้สามารถตรวจจับเส้นทางและเคลื่อนที่ได้ตามอัลกอริทึม
เซ็นเซอร์ตรวจจับการชน (ดังรูปที่ 3.24)	ออกแบบให้อยู่ด้านหน้าของตัวหุ่นยนต์
มอเตอร์ที่ใช้ในการเคลื่อนที่ (ดังรูปที่ 3.25)	ออกแบบให้อยู่ด้านหน้าสุดของตัวหุ่นยนต์ เพราะเป็นจุดที่เหมาะที่สุด สำหรับการเคลื่อนที่ไปข้างหน้าและการเลี้ยว(ไม่เหมาะสำหรับการวิ่งถอยหลัง)โดยจะมีล้อพยุงอยู่ในส่วนท้ายของตัวหุ่นยนต์เพื่อรักษาระดับการตั้งตรงขนานกับพื้น
มอเตอร์ที่ใช้ในการวางวัสดุ (ดังรูปที่ 3.26)	ออกแบบให้อยู่ใต้ภาชนะสำหรับบรรทุกวัสดุ โดยยึดกับส่วนปลายสุดของภาชนะ เพื่อให้เมื่อทำการปล่อยวางวัสดุ จะสามารถวางได้โดยไม่ขัดกับส่วนประกอบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

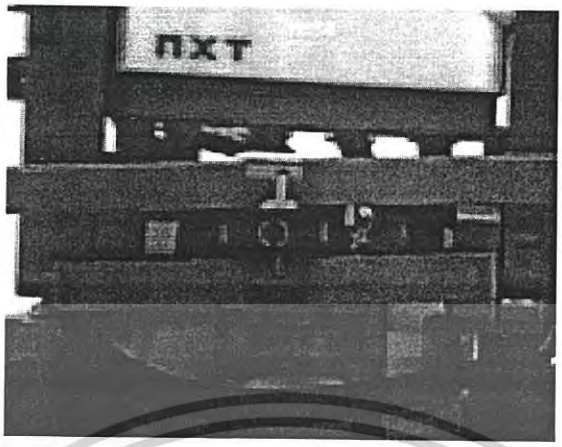


รูปที่ 3.21 เซ็นเซอร์ตรวจจับวัสดุ (ObjectScanner)

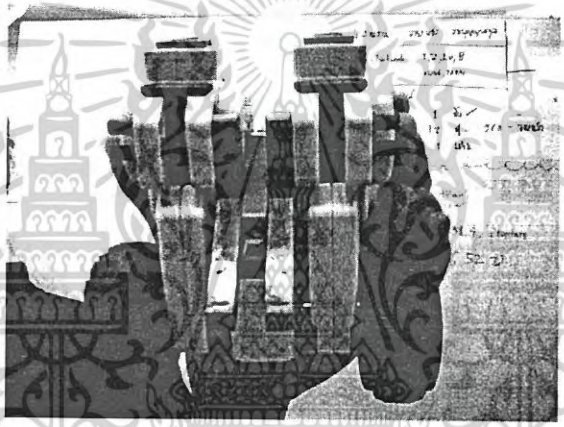
รูปที่ 3.22 เซ็นเซอร์ตรวจจับเป้าหมาย (TargetScanner)

รูปที่ 3.23 เซ็นเซอร์ตรวจจับเส้นทาง (LineTracker)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 รูปเซ็นเซอร์ตรวจจับการชน

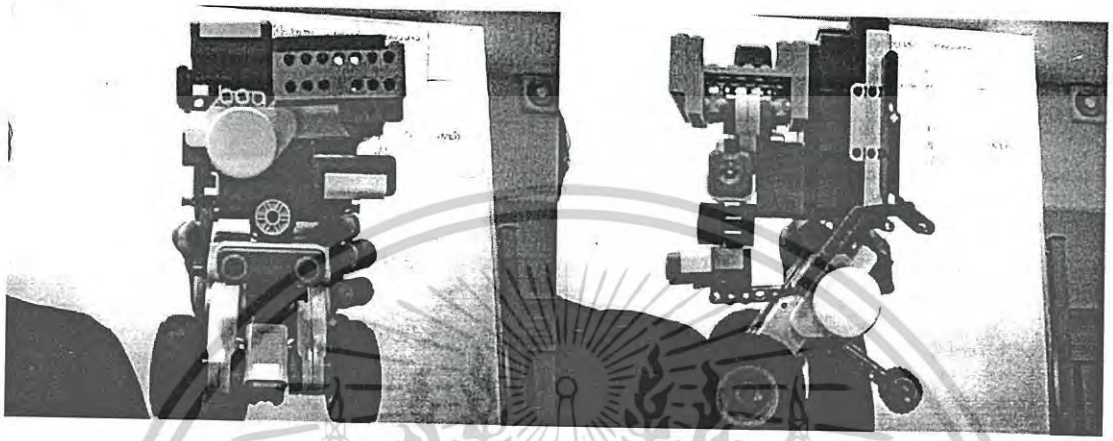


รูปที่ 3.25 มอเตอร์ที่ใช้ในการเคลื่อนที่ (Mover)

รูปที่ 3.26 มอเตอร์ที่ใช้ในการวางวัสดุ (Deployer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกอบโมดูลต่างๆเข้าด้วยกัน จะแสดงให้เห็นถึงตัวหุ่นที่ครบถ้วนตามองค์ประกอบดังกล่าว ออกแบบโดยคำนึงถึงลักษณะตัวหุ่นยนต์ที่มีรูปแบบเป็นทรงสูง เพื่อลดปัญหาการเคลื่อนไหวที่อาจไม่สะดวกในพื้นที่แคบ อีกทั้งยังทำให้วางวัสดุได้ง่ายขึ้น
(ดังรูปที่ 3.27)



รูปที่ 3.27 การประกอบโมดูลทั้งหมดเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดสอบการทำงาน

ในบทนี้จะกล่าวถึงองค์ประกอบต่างๆที่ใช้ในการทดลอง รวมถึงผลการทดลองที่บันทึกได้จากการ ซึ่งองค์ประกอบของการทดลองจะประกอบไปด้วย สนามที่ใช้ , กลไกควบคุมระบบทางด้านฮาร์ดแวร์(Mechanism) และ โปรแกรมที่ใช้ควบคุม รวมไปถึงองค์ประกอบอื่นๆ ที่เป็นสภาพแวดล้อมที่เกี่ยวข้องกับระบบ เช่น วัสดุ , อุปกรณ์ตัวรับวัสดุ หรือ เส้นทาง โดยจะนำเสนอเป็นสองส่วนคือส่วนของการทดสอบการทำงานของเซ็นเซอร์(อินพุต) และอุปกรณ์แสดงเอาท์พุต เช่น มอเตอร์ หน้าจอแอลซีดีเป็นต้น ส่วนที่สองคือการทดสอบตามสภาพการณ์หรือการทำงานตามวัตถุประสงค์ต่างๆ

4.1 การทดสอบการทำงานของฮาร์ดแวร์

ในหัวข้อนี้จะทำการทดสอบการทำงานของฮาร์ดแวร์ต่างๆ ทั้งอินพุตและเอาท์พุต เช่น เซ็นเซอร์ มอเตอร์ หรือ หน้าจอแอลซีดี โดยการทดสอบจะทำการเขียนโปรแกรมเพื่อหาความสัมพันธ์ที่ได้จากการทดสอบ

4.1.1 การทดสอบมอเตอร์ (Motor Testing)

จะใช้คลาส Motor ในการทดสอบการขับเอาท์พุตให้กับมอเตอร์ของNXT โดยภายในคลาสจะมีการใช้คำสั่งต่างๆเช่น การกำหนดความเร็วของมอเตอร์ การหมุนด้วยการกำหนดค่าองศา การสั่งให้มอเตอร์ทำการหมุนไปข้างหน้าและหลัง เป็นต้น

4.1.2 การทดสอบเซ็นเซอร์แสงและหน้าจอแอลซีดี (Light Sensor&LCD Testing)

จะใช้คลาส LightValue ในการทดสอบการรับอินพุตจากเซ็นเซอรตรวจจับค่าความสว่างของแสงแล้วนำไปแสดงผลที่หน้าจอแอลซีดี เพื่อตรวจสอบค่าที่ได้จากเซ็นเซอร์และนำไปใช้งานทดสอบโดยการให้แสงที่มีค่าแตกต่างกันแก่เซ็นเซอร์แสงแล้วสังเกตค่าความแตกต่างที่เกิดขึ้น

4.1.3 การทดสอบเซ็นเซอร์สัมผัส(Touch Sensor Testing)

ใช้การทดสอบการรับรู้ของเซ็นเซอร์ตรวจจับการสัมผัสโดยการ กดที่เซ็นเซอร์สัมผัสแล้วให้แสดงผลออกมาเป็นเสียงในกรณีที่เซ็นเซอร์รับรู้

4.1.4 การทดสอบการวัดค่าความสว่างของสี(Color Calibration)

ใช้คลาส ValueAnalysis ที่ถูกสร้างขึ้นเพื่อใช้ในการวัดช่วงค่าความสว่างของแต่ละสีที่เซ็นเซอร์มองเห็น โดยจะใช้หลักการของการวัดผลกระทบจากการแกว่งของค่าดังกล่าว
ที่เกิดจากแสง (ตารางที่ 4.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าการทดสอบการวัดความสว่างของสี (Color Calibration Value)

	Lowerbound	Upperbound	
Black(Floor)	238	282	***Lowest
Violet	290	329	
Blue	343	381	
Green	340	374	
Yellow	434	454	***Highest
Orange	419	437	
Red	390	418	

4.2 การทดสอบการทำงานตามสภาพการณ์ (Scenario)

ในหัวข้อนี้จะทำการทดสอบการทำงานของหุ่นยนต์ตามวัตถุประสงค์ ซึ่งได้แก่ การตรวจจับและเคลื่อนที่ตามเส้นทาง (Line Following) การตรวจสอบการวางวัสดุ (Deploying) การจัดเรียงวัสดุในกรณีที่จุดเก็บมีหลายส่วน (แต่ละส่วนมีพื้นที่จำกัด)(Sorting) การรอรับคำสั่งเพื่อเดินทางหาจุดที่ต้องการจากผู้ใช้ (Order-Waiting)

4.2.1 การตรวจจับและเคลื่อนที่ตามเส้นทาง (Line Following)

รูปที่ 4.1 แสดงการทำงานของการ Track Line และเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์เพื่อทดสอบระบบการตรวจสอบเส้นทางที่โดยใช้อัลกอริทึม และสภาพแวดล้อมดังต่อไปนี้

- รูปแบบของเส้น(สนาม)(ดังรูปที่ 4.1)
- สีของเส้นทางที่จะใช้ในการทดสอบ ประกอบด้วย สี ดำ(พื้นสนาม) และ สีฟ้า(เส้น)

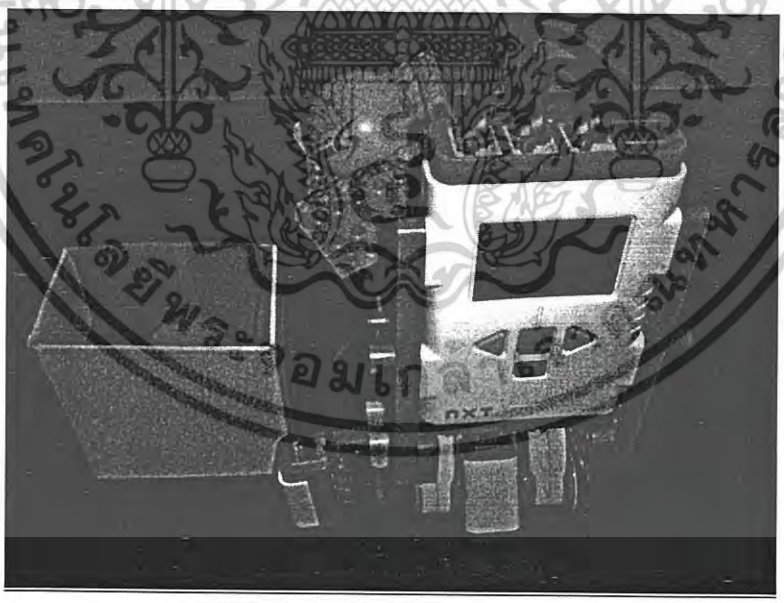
เป็นการกำหนดเงื่อนไขให้กับการเคลื่อนที่ของหุ่นยนต์โดยจะกำหนดรูปแบบการควบคุมอยู่ 3 รูปแบบคือการเลี้ยวซ้าย (ล้อซ้ายหยุด – ล้อขวาหมุน) เลี้ยวขวา (ล้อซ้ายหมุน – ล้อขวาหยุด) และเดินตรง (ล้อหมุนทั้งคู่)

โดยจะมีการตรวจสอบเงื่อนไขจากค่าของ 1 ที่รับได้จาก LightSensor ซึ่งพิจารณาเป็นช่วงของความสว่างแสงที่รับได้ จึงจะให้ผลที่แตกต่างกันในการเดิน

ผลการทดสอบ

หุ่นยนต์จะทำการเคลื่อนที่ตามรอยต่อระหว่างเส้นสนามด้านนอกกับพื้นสนาม ทั้งการเคลื่อนที่แบบตามเข็มหรือทวนเข็ม และการกำหนดความเร็วของมอเตอร์ที่ใช้แทนล้อทั้งสองข้างให้มีความเร็วที่ไม่เท่ากันจะสามารถควบคุมการตีวงในการเลี้ยวของตัวหุ่นยนต์ให้แคบหรือกว้างได้

4.2.2 การตรวจสอบการวางวัสดุ (Deploying)



รูปที่ 4.2 แสดงการวางวัสดุ (Deployer)

จุดประสงค์ เพื่อตรวจสอบการวางวัสดุที่หุ่นยนต์มีลงในตำแหน่งที่ถูกต้อง โดยใช้การตรวจสอบค่าสีของวัสดุ กับ ค่าสีของเป้าหมาย(Marker) ณ จุดวาง ซึ่งใช้อัลกอริทึมและสภาพแวดล้อมดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สีที่ใช้แทนจุดรับวัสดุ มี 2 สี
  - สีส้ม
  - สีฟ้า
- ใช้อุปกรณ์เซ็นเซอร์ตรวจจับค่าความสว่างของแสง 2 ตัว
  - ตรวจจับค่าความสว่างของวัสดุ
  - ตรวจจับค่าความสว่างของเป้าหมาย
- เซอร์โวมอเตอร์ในการขับวัสดุ 1 ตัว

ติดตั้งอยู่ ณ จุดที่รับวัสดุ

ผลการทดสอบ

หุ่นยนต์ตรวจสอบค่าสีที่ได้จากการกำหนดช่วงค่าสีทั้งหมดที่ใช้ในการทดสอบแล้วใส่ค่าไว้ในตัวแปรชนิด String แล้วทำการเปรียบเทียบค่าระหว่างวัสดุที่มีอยู่กับค่าของมาร์คเกอร์เป้าหมาย(จุดวาง) หากมีค่าเท่ากัน(ตัวแปรมีค่าเท่ากัน)ก็จะทำการวางวัสดุโดยการหมุนมอเตอร์ที่ใช้ทำการวางวัสดุไปเป็นจำนวน 90 องศา

4.2.3 การจัดเรียงวัสดุในกรณีที่จุดเก็บมีหลายส่วน (Sorting)

รูปที่ 4.3 การจัดเรียงวัสดุในกรณีที่จุดเก็บมีหลายส่วน

ในกรณีที่จุดวางวัสดุแบ่งออกเป็นส่วนย่อยๆหลายส่วนและและแต่ละส่วนมีพื้นที่จำกัด ยกตัวอย่างเช่น จุดวางวัสดุเป็นกล่องหรือรางที่แบ่งเซ็กชัน(Section) เอาไว้อย่างชัดเจน แต่ละเซ็กชันสามารถเก็บวัสดุได้เพียงชิ้นเดียว ในการวางวัสดุจึงต้องเพิ่มส่วนของการเก็บค่าที่อ้างอิงถึงจุดวางแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

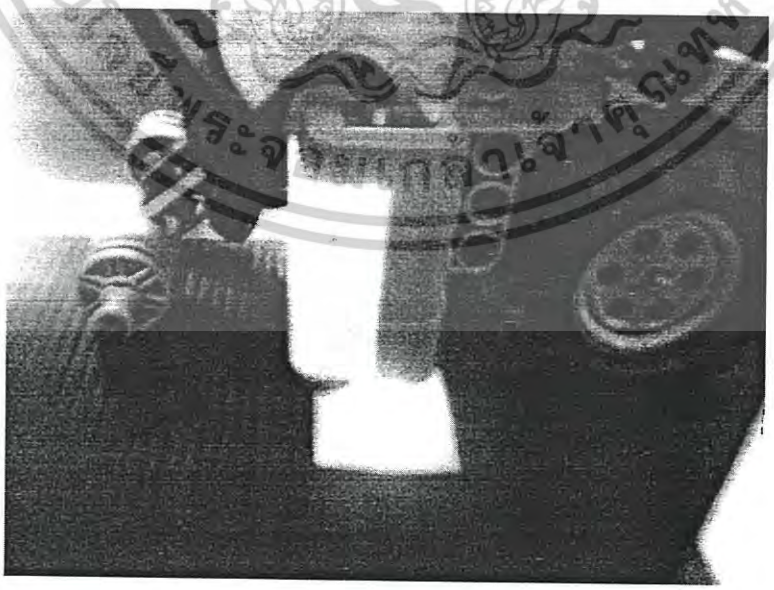
ละจุด จำนวนครั้งที่สามารถวางได้(จำนวนเซ็กชัน) และจำนวนครั้งที่วางไปแล้วในแต่ละเซ็กชัน โดยมีสภาพแวดล้อมดังนี้

- สีที่ใช้แทนจุดรับวัสดุ มี 2 สี
  - สีส้ม (กำหนดเป็นจุดวางที่มีจำนวนกล่อง 2 กล่อง)
  - สีฟ้า
- ใช้อุปกรณ์เซ็นเซอร์ตรวจจับค่าความสว่างของแสง 2 ตัว
  - ตรวจจับค่าความสว่างของวัสดุ
  - ตรวจจับค่าความสว่างของเป้าหมาย
- เซอร์โวมอเตอร์ในการขับวัสดุ 1 ตัว ติดตั้งอยู่ ณ จุดที่รับวัสดุ
- เซอร์โวมอเตอร์ในการขับเคลื่อนล้อ 2 ตัว

ผลการทดสอบ

หุ่นยนต์จะทำการตรวจสอบ ค่าที่อ้างอิงถึงจุดวางแต่ละจุด จำนวนครั้งที่สามารถวางได้ (จำนวนกล่อง) และจำนวนครั้งที่วางไปแล้วในแต่ละกล่อง ซึ่งทั้งหมดถูกเก็บอยู่ในหน่วยความจำ ซึ่งหากพบว่ามีการวางวัสดุที่จุดเดิม จะทำการขยับล้อไปเล็กน้อยเพื่อให้ทำการวางวัสดุพอดีกับตำแหน่งของกล่องถัดไป และเมื่อจำนวนของวัสดุถึงขีดจำกัดที่จุดวางสามารถรับได้ หุ่นยนต์ก็จะไม่ทำการวางวัสดุและส่งเสียงเตือน

4.2.4 การรอรับคำสั่งเพื่อเดินทางหาจุดที่ต้องการจากผู้ใช้ (Order-Waiting)



รูปที่ 4.4 การรอรับคำสั่งเพื่อเดินทางหาจุดที่ต้องการจากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบนี้ หุ่นยนต์สามารถรอรับคำสั่ง โดยผู้ใช้งานจะทำการป้อนอินพุตคือสีของจุดวางวัสดุที่ต้องการเพื่อให้หุ่นยนต์สามารถเดินทางหาจุดดังกล่าวและหยุดเพื่อรอรับวัสดุต่อไป โดยจะใช้สภาพแวดล้อมดังนี้

- สีที่ผู้ใช้งานสามารถเลือกให้หุ่นยนต์เดินหา 2 สี
  - สีส้ม
  - สีฟ้า
- ใช้อุปกรณ์เซ็นเซอร์ตรวจจับค่าความสว่างของแสง 2 ตัว
  - ตรวจจับค่าความสว่างของเส้นทาง
  - ตรวจจับค่าความสว่างของเป้าหมาย
- เซอร์โวมอเตอร์ในการขับเคลื่อนล้อ 2 ตัว

ผลการทดสอบ

เริ่มแรกหุ่นยนต์จะทำการหยุดรอรับอินพุตซึ่งเป็นค่าสีที่ต้องการ(แสดงผลบนจอ LCD)จากผู้ใช้งาน และเมื่อผู้ใช้งานทำการป้อนค่าผ่านทางปุ่มบน NXT หุ่นยนต์ก็จะทำการเดินทางเพื่อหาจุดเป้าหมาย เมื่อเจอจุดเป้าหมายหุ่นยนต์ก็จะทำการหยุดเพื่อรอรับวัสดุและทำงานหลักคือการจัดเก็บวัสดุต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดสอบและข้อเสนอแนะ

5.1 สรุปผลการทดสอบ

การพัฒนาหุ่นยนต์จัดเก็บวัสดุ ใช้ภาษาจาวาในการเขียนโปรแกรม โดยมีไลบรารีของ LEJOS NXJ และใช้ฮาร์ดแวร์ของ ชุดทดลองเทคโนโลยีฝังตัวLego MINDSTORMS NXT นั้นช่วยให้สามารถจัดเก็บวัสดุภายในสภาพแวดล้อมของการทำงานภายในโรงงานได้โดยการแยกแยะประเภทของวัสดุ เซ็นเซอร์แยกประเภทของเป้าหมาย และ เซ็นเซอร์ในการตรวจจับการชน รวมถึงมอเตอร์ที่ใช้ในการขับเคลื่อนล้อในการเคลื่อนที่และในการวางวัสดุ

หุ่นยนต์จัดเก็บวัสดุสามารถตอบสนองความต้องการของผู้ใช้ได้โดยมีรูปแบบการทำงานที่สามารถรอรับคำสั่งในการเดินหาจุดเป้าหมายที่ต้องการเพื่อรอรับวัสดุได้ และสามารถจัดเรียงวัสดุในกรณีที่จุดวางวัสดุมีหลายส่วน

5.2 ประโยชน์ที่ได้รับจากการสร้างหุ่นยนต์จัดเก็บวัสดุ

1. ทำให้เกิดการเรียนรู้เกี่ยวกับการสร้างระบบฝังตัวหุ่นยนต์จัดเก็บวัสดุ (Store Keeper Robot)" และประยุกต์แนวทางในการพัฒนาหุ่นยนต์ โดยใช้หลักการพัฒนาระบบเชิงวัตถุ เข้ากับระบบฝังตัว

2. ได้ต้นแบบในการสร้างหุ่นยนต์ที่ใช้ในงานที่ต้องการใช้การแยกแยะประเภทของวัสดุและจัดเก็บในที่ที่กำหนด

5.3 ปัญหาและอุปสรรค

ปัญหาที่เกิดขึ้นในการทดลองส่งผลกระทบต่อวิธีการดำเนินการในแต่ละขั้นตอน ให้มีรูปแบบการพัฒนาโครงงานที่จำเป็นต้องปรับเปลี่ยนแนวคิดและวิธีการแก้ปัญหาอยู่ตลอด ปัญหาที่เกิดขึ้นมีดังต่อไปนี้

- เซ็นเซอร์แสง มีปัญหาในเรื่องของการเก็บค่าที่ เป็นตัวเลข ซึ่งทำให้การวิเคราะห์สีเป็นไปได้ค่อนข้างยาก

- ผลกระทบจากปัจจัยแวดล้อมภายนอก คือ ระดับของแสง ที่มีผลต่อความเข้มของแสงที่เซ็นเซอร์รับได้ อีกช่วงของแสงที่รับได้นั้นมีอยู่อย่างจำกัด และเมื่อต้องมีการนำมาใช้งานในการจำแนกวัตถุ จึงเกิดปัญหาที่เรียกว่า "ปัญหาการจัดสรรช่วงของความเข้มแสง" เกิดจากการที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์เซ็นเซอร์ เพียงตัวเดียวในการทำหน้าที่เป็นทั้ง ตรวจจับเส้นทางและตรวจจับเป้าหมายพร้อมๆกัน ทำให้เกิดปัญหาการซ้ำซ้อนของสีและเกิดอาการรวนของระบบเมื่อแสดงผล ทำให้ต้องปรับแก้ทางกลไกฮาร์ดแวร์

- ปัญหากลไกทางฮาร์ดแวร์ที่มีความยืดหยุ่นน้อยทำให้ต้องปรับเปลี่ยนรูปแบบของโครงสร้างอยู่ตลอด

- ปัญหาการไม่สอดคล้องกันของค่าที่รับได้ในแต่ละเซ็นเซอร์ทั้งๆที่เกิดจากวัตถุสีเดียวกัน เนื่องมาจากระดับความสูงที่แตกต่างกันทำให้ได้ค่าที่แตกต่างกัน

5.4 ข้อเสนอแนะ

ผลกระทบจากสภาพแวดล้อมภายนอก มีผลโดยตรงกับการทำงานของเซ็นเซอร์ ซึ่งทำให้เกิดปัญหาตามมามากมาย เพราะฉะนั้น ในการนำแนวคิดไปพัฒนาต่อจึงควรเลือกเซ็นเซอร์ที่มีประสิทธิภาพ เพื่อลดความคลาดเคลื่อนดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

อดิศร ณ อุบล. 2548. **UML.** [Online]. Available:
learners.in.th/file/apealex/UML.ppt.

วุฒิพงษ์ เรือนทอง. 2552. **Evolution.** [Online]. Available:
learners.in.th/file/wuttipongr/DevMethodology.ppt.

Administrator. 2550. **การพัฒนาSoftware(System Development Life Cycle).** [Online].Available:
http://takasila.coe.psu.ac.th/~s4810168/SDLC%20%C7%A7%A8%C3%AA%D5%C7%D4%B5%A2%CD%A7%A1%D2%C3%BE%D1%B2%B9%D2%20Software.doc.

coffeycountyks.org. 2555. **Java Interpreter.** [Online]. Available:
http://www.coffeycountyks.org/Terms/2461HTML-30.gif.

en.wikipedia. 2551. **Compiler**. [Online]. Available:
http://en.wikipedia.org/wiki/Compiler.

Japan System House Association (JASA). 2549. **เทคโนโลยีสมองกลฝังตัว Embedded Technology**: สำนักพิมพ์ ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

K.Ratchadaporn. 2549. **JAVA PROGRAMMING.** [Online]. Available:
www.cs.su.ac.th/~kanawong/courses/517511/slide/JAVA%20PROGRAMMING%2.

LEJOS NXJ. 2551. **API.** [Online]. Available:
http://lejos.sourceforge.net/nxt/nxj/api/index.html.

Lego Digital Designer. 2551. **Intro.** [Online]. Available:
http://ldd.lego.com/.

Mr.POP. 2550. **ตัวอย่างการสืบทอดคุณสมบัติ.** [Online]. Available:
http://www.webthaidd.com/java/webthaidd_article_263_1.html.

th.wikipedia. 2551. **USB.** [Online]. Available:
http://th.wikipedia.org/wiki/ยูเอสบี.

th.wikipedia. 2551. **ทฤษฎีแบบจำลองน้ำตก.** [Online]. Available:
http://th.wikipedia.org/wiki/ทฤษฎีแบบจำลองน้ำตก.

www.itmelody.com. 2552. **UML Model**. [Online].Available:
http://www.itmelody.com/tu/uml1.htm.

www.cs.tu.ac.th. 2551. **Process Model.** [Online]. Available:
www.cs.tu.ac.th/classes/471/undergradcs456-/public/AdditionalRead/IntroProcess.doc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมที่ใช้ควบคุมการทำงานของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การทดสอบมอเตอร์

คลาส Motor ถูกสร้างขึ้นเพื่อใช้ในการทดสอบการขับ output ให้กับมอเตอร์ของ NXT

```
import lejos.nxt.*;
public class Motor {
  public static void main(String[] args) throws Exception {
    Motor.C.setSpeed(50);
    Motor.B.setSpeed(50);
    Motor.A.setSpeed(50);
    while (true){
      if (Button.ENTER.isPressed()) {
        break;}
      else {
    Motor.A.backward();
    Motor.B.forward();
    Motor.C.forward();
      }
    }
  }
}
```

อธิบายโปรแกรม

- บรรทัดแรก ทำการ import package library เพื่อการควบคุม NXT นั้นจำเป็นต้องมีการเรียกใช้ Class ใน library ดังกล่าว ด้วยคำสั่ง import lejos.nxt.*;
- ต่อมา Method main ในคลาส motor ทำหน้าที่ ดังต่อไปนี้
  - กำหนดค่าเริ่มต้นให้กับความเร็วของ มอเตอร์ทั้งสามคือ A ,B และ C ด้วยคำสั่ง Motor.[X].setSpeed([Parameter]) ; โดย
    X คือ ชื่อมอเตอร์ใดๆ
    Parameter คือ ค่าความเร็วที่ต้องการกำหนดให้
  - ถัดมาเป็น Loop While ที่ต้องใช้ Loop เพราะต้องการให้มีการการวนรอรับคำสั่งและกระทำคำสั่งดังกล่าวไปเรื่อยๆ จนกว่าจะมีการกดปุ่มใดๆ จึงจะหยุดการทำงานใน Loop โดยจะใช้คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(       if (Button.ENTER.isPressed()) {
  break; }
)
```

- เมื่อไม่มีการกดปุ่มใดๆ จะทำการให้มอเตอร์ทั้งสาม ลองหมุน ไปข้างหน้าและหลัง เพื่อตรวจสอบความถูกต้องของการดำเนินการทดสอบโดยใช้คำสั่ง

Motor.[X].backward(); เมื่อต้องการให้มอเตอร์ดังกล่าวหมุนไปข้างหลัง

Motor.[X].forward(); เมื่อต้องการให้มอเตอร์ดังกล่าวหมุนไปข้างหน้า

เมื่อ X คือ ชื่อมอเตอร์ใดๆ

2. การทดสอบเซ็นเซอร์ตรวจจับความสว่างและหน้าจอ LCD

คลาส LightValue ถูกสร้างขึ้นเพื่อใช้ในการทดสอบการรับ input จาก Light Sensor แล้วนำไปแสดงผลที่จอ LCD ของ NXT

```
import lejos.nxt.*;
public class LightValue {
  public static int l1 = 0;
  public static int l2 = 0;
  public static int l3 = 0;
  public static void main(String[] args) throws Exception {
    LightSensor light1 = new LightSensor(SensorPort.S1);
    LightSensor light2 = new LightSensor(SensorPort.S2);
    LightSensor light3 = new LightSensor(SensorPort.S3);
    while (true) {
      if (Button.ENTER.isPressed()) {
        break;
      } else {
        l1 = light1.readNormalizedValue();
        l2 = light2.readNormalizedValue();
        l3 = light3.readNormalizedValue();
        LCD.drawString("Tracker", 1, 1);
        LCD.drawString("Target", 1, 3);
        LCD.drawString("Object", 1, 5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCD.drawInt(l1, 10, 1);
            LCD.drawInt(l2, 10, 3);
            LCD.drawInt(l3, 10, 5);
        }
    }
  }
}
```

อธิบายโปรแกรม

- บรรทัดแรก ทำการ import package library เพื่อการควบคุม NXT นั้นจำเป็นต้องมีการเรียกใช้ Class ใน library ดังกล่าว ด้วยคำสั่ง import lejos.nxt.*;
- ต่อมา ภายในคลาส LightValue ได้ทำการกำหนดตัวแปรไว้ 3 ตัวเป็นตัวแปร Integer แบบ Static ในกรณีนี้เพื่อให้ทุก Object สามารถเข้าถึงและใช้งานตัวแปรเดียวกัน โดยมีค่าเริ่มต้นเป็น 0
- ภายใน Method main ทำการสร้าง Object ของคลาส LightSensor ขึ้นมา 3 Object โดยให้ชื่อ Object ตาม Port ดังนี้

light1	=	Port1
light2	=	Port2
light3	=	Port3

โดยรูปแบบของการสร้าง Object LightSensor จะเป็นดังนี้

```
LightSensor X = new LightSensor(SensorPort.S[Y]);
```

โดยที่ X คือ ชื่อของ Object ที่จะมารองรับ

Y คือ หมายเลข Port Number (1,2,3,…)

- ถัดมาเป็น Loop While ที่ต้องใช้ Loop เพราะต้องการให้มีการการวนรอรับคำสั่งและกระทำคำสั่งดังกล่าวไปเรื่อยๆ จนกว่าจะมีการกดปุ่มใดๆ จึงจะหยุดการทำงานใน Loop โดยจะใช้คำสั่ง

```
(       if (Button.ENTER.isPressed()) {
 break; }
)
```

- และหากไม่มีการกดปุ่มใดๆเพื่อหยุดการทำงาน จะมาทำคำสั่งต่อไปนี้
- ให้ตัวแปร l1 , l2 , l3 ที่ได้ทำการกำหนดไว้ตอนเริ่มต้น เป็นตัวรับค่าของการใช้ Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

readNormalizedValue() ซึ่ง Method นี้จะทำการ Return ค่าความสว่างของแสงกลับมาเป็นตัวแปร Integer ที่มีช่วงระหว่าง145 (ความมืด) – 890 (แสงอาทิตย์) ของคลาส LightSensor ที่อ้างอิงโดย Object light1 , light2 และ light3 ตามลำดับ โดยคำสั่งจะเป็นรูปแบบดังต่อไปนี้

[X] = [Y].readNormalizedValue(); โดย

X คือ ตัวแปรที่จะมารับค่าที่มีการ Return กลับมาจากการเรียกใช้ Method readNormalizedValue() ของ Object ที่สร้างขึ้น(light1 , light2 , light3)

Y คือ Object ของคลาส LightSensor (light1 , light2 , light3)

- ทำการแสดงผลค่าตัวแปร l1 ,l2 ,l3 เพื่อดูค่าความสว่างของแสงที่ NXT รับรู้ จาก LightSensor โดยการใช้คำสั่งดังต่อไปนี้

- LCD.drawString("[String]", X, Y);

Method drawString นี้เป็น Method ของ Class LCD ที่ใช้ ในการแสดงผล Array ของตัวอักษรที่ต้องการแสดงผล โดย

String คือ ตัวอักษรที่ต้องการให้แสดงผล ต้องอยู่ภายใต้เครื่องหมาย “ ”

X คือ พิกัดแกน X ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

Y คือ พิกัดแกน Y ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

- LCD.drawInt([Integer], X, Y);

Method drawInt นี้เป็น Method ของ Class LCD ที่ใช้ ในการแสดงผลค่าตัวเลขประเภท Integer ที่ต้องการแสดงผล โดย

Integer คือ ตัวเลขที่ต้องการให้แสดงผล

X คือ พิกัดแกน X ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

Y คือ พิกัดแกน Y ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การหาค่าช่วงของความสว่างสี

คลาส ValueAnalysis ถูกสร้างขึ้นเพื่อใช้ในการทดสอบ การวัดค่าของความสว่างของสี โดยจะใช้หลักการวัดผลกระทบจากการแกว่งของค่าดังกล่าว ที่เกิดจากแสงในแต่ละพื้นที่ที่ต้องการใช้งาน

	Lowerbound	Upperbound	
Black(Floor)	238	282	***Lowest
Violet	290	329	
Blue	343	381	
Green	340	374	
Yellow	434	454	***Highest
Orange	419	437	
Red	390	418	

ตารางที่ 4.3 แสดงการบันทึกค่าที่ได้จากการทำ ValueAnalysis

บันทึกผลในรูปของตารางที่เก็บค่าความแกว่งของแต่ละสี(Lowerbound-Upperbound) และค่าต่ำสุด - สูงสุดของสีทั้งหมด(Lowest-Highest)

```
import lejos.nxt.*;
public class ValueAnalysis {
  public static void main(String[] args) throws Exception {
    LightSensor light1 = new LightSensor(SensorPort.S1);
    LightSensor light2 = new LightSensor(SensorPort.S2);
    LightSensor light3 = new LightSensor(SensorPort.S3);
    int l1,l2,l3;
    int c1 = 0,c2 = 0,c3 = 0;
    int f1 = 800,f2 = 800,f3 = 800;
    Motor.C.setSpeed(50);
    Motor.B.setSpeed(50);
    while (true) {
      if (Button.ENTER.isPressed()) {
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        else {
          l1 = light1.readNormalizedValue();
          l2 = light2.readNormalizedValue();
          l3 = light3.readNormalizedValue();
          if (l1 > c1) c1 = l1;
          if (l1 < f1) f1 = l1;
          if (l2 > c2) c2 = l2;
          if (l2 < f2) f2 = l2;
          if (l3 > c3)c3 = l3;
          if (l3 < f3) f3 = l3;
          LCD.drawString("Mov", 3, 0);
          LCD.drawString("Tar", 7, 0);
          LCD.drawString("Obj", 11, 0);
          LCD.drawString("C", 0, 2);
          LCD.drawString("L", 0, 4);
          LCD.drawString("F", 0, 6);
          LCD.drawInt(c1, 3, 2);
          LCD.drawInt(l1, 3, 4);
          LCD.drawInt(f1, 3, 6);
          LCD.drawInt(c2, 7, 2);
          LCD.drawInt(l2, 7, 4);
          LCD.drawInt(f2, 7, 6);
          LCD.drawInt(c3, 11, 2);
          LCD.drawInt(l3, 11, 4);
          LCD.drawInt(f3, 11, 6);
          Motor.C.forward();
          Motor.B.backward();
        }
      }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายโปรแกรม

- บรรทัดแรก ทำการ import package library เพื่อการควบคุม NXT นั้นจำเป็นต้องมีการเรียกใช้ Class ใน library ดังกล่าว ด้วยคำสั่ง import lejos.nxt.*;
- Method main ภายในคลาส ValueAnalysis กระทำการดังต่อไปนี้
- สร้าง Object ของคลาส LightSensor ด้วยคำสั่ง

  LightSensor [ObjectName] = new LightSensor(SensorPort.S[PortNumber]);
- กำหนดค่าตัวแปร l1 , l2 , l3 เป็นตัวแปรประเภท Integer เพื่อรองรับค่าที่ Return กลับมาของ Object ของคลาส LightSensor
- กำหนดค่าตัวแปร Dummy ของ ค่าความสว่างสูงสุด (ค่าเพดาน : Ceiling) เพื่อใช้ในการเปรียบเทียบกับค่าที่ Return กลับมาจาก Method readNormalizedValue() และทำการเก็บค่าในตัวแปร c1 , c2 , c3 ตามลำดับ แยกตาม Port แต่ละ Port (LightSensor แต่ละตัว)
- กำหนดค่าตัวแปร Dummy ของ ค่าความสว่างต่ำสุด (ค่าพื้น : Floor) เพื่อใช้ในการเปรียบเทียบกับค่าที่ Return กลับมาจาก Method readNormalizedValue() และทำการเก็บค่าในตัวแปร f1 , f2 , f3 ตามลำดับ แยกตาม Port แต่ละ Port (LightSensor แต่ละตัว)
- ขั้นตอนนี้เป็นการตรวจสอบค่าความสว่างที่ต่ำสุดและสูงสุดที่รับได้ เพื่อเป็นการหาค่าช่วงของความสว่าง เพื่อใช้ในการทดสอบต่อไป โดยหลักการคือ หาก ค่าความสว่างที่รับได้(l)มีค่ามากกว่าค่าความสว่างต่ำสุด (f) จะทำการแทนที่ ค่าของค่าความสว่างต่ำสุด (f) ด้วย ค่าความสว่างที่รับได้(l) ด้วยคำสั่ง

```
if (l[Number] < f[Number])      {
f[Number]  = l[Number] ;
}
```

โดยในท้ายที่สุด ค่าของค่าความสว่างต่ำสุด (f) ก็จะเป็นค่าที่ต่ำที่สุดที่รับได้ เป็นเหตุผลที่ต้องกำหนดค่าเริ่มต้นของค่าความสว่างต่ำสุด (f) ด้วยค่าที่มากที่สุด เพื่อให้ค่าเปลี่ยนไปได้เมื่อทำการเปรียบเทียบกับค่าความสว่างที่รับได้ในการเปรียบเทียบครั้งแรก

และหาก ค่าความสว่างที่รับได้(l)มีค่ามากกว่าค่าความสว่างสูงสุด (c) จะทำการแทนที่ ค่าของค่าความสว่างต่ำสุด (c) ด้วย ค่าความสว่างที่รับได้(l) ด้วยคำสั่ง

```
if (l[Number] > c[Number])      {
c[Number]  = l[Number] ;
}
```

โดยในท้ายที่สุด ค่าของค่าความสว่างสูงสุด (c) ก็จะเป็นค่าที่สูงที่สุดที่รับได้ เป็นเหตุผลที่ต้องกำหนดค่าเริ่มต้นของค่าความสว่างสูงสุด (c) ด้วยค่าที่น้อยที่สุด เพื่อให้ค่าเปลี่ยนไปได้เมื่อทำการเปรียบเทียบกับค่าความสว่างที่รับได้ในการเปรียบเทียบครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการใช้คำสั่ง LCD.drawString("[String]", X, Y) และ LCD.drawInt([Integer], X, Y); เพื่อในการแสดงผล ค่าของความสว่างต่ำสุด (f : Floor) ค่าของความสว่างที่ Return กลับมา (l : light) และ ค่าของความสว่างสูงสุด (c : Ceiling)
- ทำการสั่งให้ NXT หมุนตัวเพื่อเก็บค่าความสว่างของแสงด้วยการขยับล้อตรงข้ามกัน เช่น ให้ล้อซ้ายหมุนไปข้างหน้า ล้อขวาหมุนไปข้างหลัง

โดยใช้คำสั่ง

```
Motor.[LeftMotor].forward();
Motor.[RightMotor].backward();
```

4. การทดสอบการติดตามเส้นทาง

คลาส TrackMove ถูกสร้างขึ้นมาเพื่อใช้ในการทดลองการ รับค่าความสว่างของเส้นทาง และเดินตามเส้นทางโดยมีการทำงานร่วมกันระหว่าง input ที่ได้รับจาก Light Sensor กับ output ผ่านมอเตอร์

```
import lejos.nxt.*;
public class TrackMove {
  public static void main(String[] args) throws Exception {
    LightSensor light = new LightSensor(SensorPort.S1);
    int l = 0;
    Motor.C.setSpeed(200);
    Motor.B.setSpeed(200);
      while (true) {
        if (Button.ENTER.isPressed()) {
          break;
        } else {
          LCD.drawInt(light.readNormalizedValue(), 4, 0, 1);
          l = light.readNormalizedValue();
          if (l >= 230 && l < 290) {
            Motor.C.stop();
            Motor.B.backward();
          } else if (l >= 330 && l < 380) {
            Motor.C.backward();
            Motor.B.backward();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            } else if (l >= 400 && l < 460) {
              Motor.C.backward();
              Motor.B.stop();
            }
          }
        }
      }
}
```

อธิบายโปรแกรม

- บรรทัดแรก ทำการ import package library เพื่อการควบคุม NXT นั้นจำเป็นต้องมีการเรียกใช้ Class ใน library ดังกล่าว ด้วยคำสั่ง import lejos.nxt.*;
- Method main ภายในคลาส TrackMove กระทำการดังต่อไปนี้
- สร้าง Object ของคลาส LightSensor ด้วยคำสั่ง

  LightSensor [ObjectName] = new LightSensor(SensorPort.S[PortNumber]);
- กำหนดค่าตัวแปร l เป็นตัวแปรประเภท Integer เพื่อรองรับค่าที่ Return กลับมาของ Object ของคลาส LightSensor
- กำหนดค่าเริ่มต้นให้กับความเร็วของ มอเตอร์ทั้งสองคือ B และ C (ล้อซ้ายและขวา)

  ด้วยคำสั่ง Motor.[X].setSpeed([Parameter]) ; โดย

  X คือ ชื่อมอเตอร์ใดๆ

  Parameter คือ ค่าความเร็วที่ต้องการกำหนดให้
- ถัดมาเป็น Loop While ที่ต้องใช้ Loop เพราะต้องการให้มีการการวนรอรับคำสั่งและกระทำคำสั่งดังกล่าวไปเรื่อยๆ จนกว่าจะมีการกดปุ่มใดๆ จึงจะหยุดการทำงานใน Loop โดยจะใช้คำสั่ง

  (if (Button.ENTER.isPressed()) {
  break; }
)
- ทำการแสดงผลค่าตัวแปร l เพื่อดูค่าความสว่างของแสงที่ NXT รับรู้ จาก LightSensor โดยการใช้คำสั่งดังต่อไปนี้
- LCD.drawInt([Integer], X, Y);

  Method drawInt นี้เป็น Method ของ Class LCD ที่ใช้ ในการแสดงผลค่าตัวเลขประเภท Integer ที่ต้องการแสดงผล โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Integer คือ ตัวเลขที่ต้องการให้แสดงผล

X คือ พิกัดแกน X ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

Y คือ พิกัดแกน Y ของหน้าจอ LCD(NXT Monitor) ที่ต้องการให้เริ่มแสดงผล

- กำหนดให้ตัวแปร l ที่เป็นประเภท Integer ทำการรับค่าที่ Return มาจาก Method readNormalizeValue() ของ Object
- ถัดไปเป็นการกำหนดเงื่อนไขให้กับการเคลื่อนที่ของ NXT โดยจะกำหนดรูปแบบการควบคุมอยู่ 3 รูปแบบคือการเลี้ยวซ้าย (ล้อซ้ายหยุด – ล้อขวาหมุน) เลี้ยวขวา (ล้อซ้ายหมุน – ล้อขวาหยุด) และเดินตรง (ล้อหมุนทั้งคู่)

โดยจะมีการตรวจสอบเงื่อนไขจากค่าของ l ที่รับได้จาก LightSensor ซึ่งพิจารณาเป็นช่วงของความสว่างแสงที่รับได้ จึงจะให้ผลที่แตกต่างกันในการเดิน

โดยคำสั่งคือ

```
if (l >= [LowerBound_Range1] && l < [UpperBound_Range1]) {
Motor.C.stop();                                    // เลี้ยวซ้าย
Motor.B.backward();
} else if (l >= [LowerBound_Range2] && l < [UpperBound_Range2]) {
Motor.C.backward();                                // ตรงไปข้างหน้า
Motor.B.backward();
} else if (l >= [LowerBound_Range3] && l < [UpperBound_Range3]) {
Motor.C.backward();                                // เลี้ยวขวา
Motor.B.stop();
}
```

โดยที่ LowerBound และ UpperBound คือค่าต่ำสุดและสูงสุดของช่วงความสว่างของแสงที่หาได้จาก คลาส ValueAnalysis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การทดสอบการวางวัสดุ

คลาส Deployer เป็นคลาสที่ถูกสร้างขึ้นเพื่อใช้ในการทดสอบการตรวจจับเป้าหมาย(ที่รับวัสดุ) เมื่อพบเป้าหมายที่ตรงกับวัสดุที่บรรทุก NXT จะทำการวางวัสดุลงในเป้าหมายดังกล่าว

```
public static String objcolor;
public static String tarcolor;
public class Deployer {
public static void main(String[] args) throws Exception {
scanValue();
checkobj();
checktar();
if (objcolor.equals(tarcolor)) {
        deploy();
}
public static void checkobj() {
if (l3 >= 321 && l3 <= 348) {
objcolor = "violet";
} else if (l3 >= 383 && l3 <= 410) {
objcolor = "blue";
} else if (l3 >= 477 && l3 <= 489) {
objcolor = "orange";
}
}
public static void checktar() {
if (l1 >= 224 && l1 <= 257) {
tarcolor = "black";
} else if (l1 >= 258 && l1 <= 265) {
tarcolor = "white";
} else if (l1 >= 266 && l1 <= 299) {
tarcolor = "violet";
} else if (l1 >= 340 && l1 <= 381) {
tarcolor = "blue";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    } else if (l1 >= 419 && l1 <= 437) {
      tarcolor = "orange";
    }
  public static void scanValue() {
    l1 = light1.readNormalizedValue(); // ใช้เป็น Tracker
    l2 = light2.readNormalizedValue(); // ใช้เป็น TargetScaner
    l3 = light3.readNormalizedValue(); // ใช้เป็น ObjectScaner
    LCD.drawString("Tracker", 1, 1);
    LCD.drawString("Target", 1, 3);
    LCD.drawString("Object", 1, 5);
    LCD.drawInt(l1, 10, 1);
    LCD.drawInt(l2, 10, 3);
    LCD.drawInt(l3, 10, 5);
  }
}
```

อธิบายโปรแกรม

- บรรทัดแรก ทำการ import package library เพื่อการควบคุม NXT นั้นจำเป็นต้องมีการเรียกใช้ Class ใน library ดังกล่าว ด้วยคำสั่ง import lejos.nxt.*;
- คลาส Deployer มีการกระทำดังต่อไปนี้
- กำหนดค่าตัวแปร tarcolor และ objcolor เป็นตัวแปรประเภท String เพื่อรองรับค่าที่กำหนดให้ในการสแกนแต่ละครั้ง
- Method main ภายในคลาส Deployer
- สร้าง Object ของคลาส LightSensor ด้วยคำสั่ง

  LightSensor [ObjectName] = new LightSensor(SensorPort.S[PortNumber]);
- กำหนดค่าเริ่มต้นให้กับความเร็วของ มอเตอร์ที่ทำหน้าที่เป็นตัว Deployer คือ มอเตอร์ A

  ด้วยคำสั่ง Motor.[X].setSpeed([Parameter]) ; โดย

  X คือ ชื่อมอเตอร์ใดๆ

  Parameter คือ ค่าความเร็วที่ต้องการกำหนดให้
- ถัดมาเป็น Loop While ที่ต้องใช้ Loop เพราะต้องการให้มีการการวนรอรับคำสั่งและกระทำคำสั่งดังกล่าวไปเรื่อยๆ จนกว่าจะมีการกดปุ่มใดๆ จึงจะหยุดการทำงานใน Loop โดยจะใช้คำสั่ง

  (if (Button.ENTER.isPressed()) {

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break; }
        )
```

- ใน Loop กำหนดให้ตัวแปร tarcolor ที่เป็นประเภท Integer ทำการรับค่าที่ Return มาจาก Method readNormalizeValue ของ Object
- เรียกทำงาน Method main() ซึ่งใน Method จะประกอบไปด้วยขั้นตอนดังนี้

```
- scanValue();      // การเรียก เมธอด ที่ใช้ในการอัพเดตค่าสีให้กับทุกตัวแปร
  checkobj();       // การเช็คสีของวัสดุ
  checktar();       // การเช็คสีของเป้าหมาย
  if (objcolor.equals(tarcolor)) {    // ถ้าหากว่ามีค่าเท่ากันก็จะให้ไปทำเมธอด deploy
          deploy();
  }
- public static void deploy() {   // เมธอด deploy ทำการหมุนมอเตอร์ A  90 องศา
  pause();
  Motor.A.rotate(-90);
  ahead();
  }
```

6. การเดินทางตามหาเส้นทางและหยุดวางของเมื่อเจอวัสดุ

เป็นการนำอัลกอริทึมและโค้ดของโปรแกรมการค้นหาเส้นทางและหยุดวางวัสดุ มาประกอบกัน และยังเพิ่มส่วนของการจัดเรียงวัสดุในกรณีที่มีการวางซ้ำซ้อน

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
import lejos.nxt.*;
import lejos.nxt.Sound;
/**
 *
 * @author Peridot
 */
public class AutoMode {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ประกาศตัวแปรทั้งหมดไว้ด้านนอกเพื่อการเข้าถึงได้อย่างง่าย

    public static String objcolor = "a";
    public static String tarcolor = "b";
    public static String tracolor = "c";
    public static String color = "default";
    public static int l1 = 0;
    public static int l2 = 0;
    public static int l3 = 0;
    public static int orange = 1;
    public static int violet = 1;
    public static int blue = 1;
    public static int i = 1;
    public static LightSensor light1 = new LightSensor(SensorPort.S1);
    public static LightSensor light2 = new LightSensor(SensorPort.S2);
    public static LightSensor light3 = new LightSensor(SensorPort.S3);
    public static TouchSensor touch = new TouchSensor(SensorPort.S4);
    public static void main(String[] args) throws Exception {
//      เมธอดตั้งค่าที่ใช้ครั้งเดียว เอาไว้นอกloop while
      Motor.A.setSpeed(100);
      Motor.B.setSpeed(100);
      Motor.C.setSpeed(100);
// ในwhileจะใช้แค่if ไม่มีelse เป็นการเชคเงื่อนไขเพียงแค่นั้น
      while (true) {
        if (Button.ESCAPE.isPressed()) {
          break;
        }
        while (true) {
          if (Button.ESCAPE.isPressed()) {
            break;
          }
          if (touch.isPressed()) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pause();
            Sound.beep();
          } else {
            scanValue();
            if (tarcolor.equals("red  ")) {
              pause();
              Sound.beepSequence();
              step2();
              break;
            }
            if (!(tarcolor.equals("target"))) {
              step2();
            }
            if (tarcolor.equals(objcolor)) {
              Sound.beep();
              updateColor();
              if (i > 2) {
                Sound.beepSequence();
              } else {
                deploy();
              }
            }
            trackMove();
          }
        }
        while (true) {
          if (Button.ESCAPE.isPressed()) {
            break;
          }
          scanValue();
          if (!(objcolor.equals("object"))) {
            Sound.beepSequence();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                step2();
                step2();
                break;
            }
        }
    }
}
//checkobj ใช้สำหรับตีความค่าความสว่างของวัสดุให้เป็นชื่อของสี เพื่อสะดวกในการเปรียบเทียบ
  public static void checkObject() {

    if (l1 >= 370 && l1 <= 396) {
      objcolor = "blue  ";
    } else if (l1 >= 405 && l1 <= 475) {
      objcolor = "orange";
    } else if (l1 >= 316 && l1 <= 360){
      objcolor = "object";
    }
  }
  public static void checkTarget() {
    if (l3 >= 350 && l3 <= 420) {
      tarcolor = "blue  ";
    } else if (l3 >= 470 && l3 <= 500) {
      tarcolor = "orange";
    } else if (l3 >= 440 && l3 <= 460) {
      tarcolor = "red   ";
    } else {
      tarcolor = "target";
    }
  }
//checktrack ใช้สำหรับตีความค่าความสว่างของเส้นให้เป็นชื่อของสี เพื่อสะดวกในการเปรียบเทียบ
  public static void checkTrack() {
    if (l2 < 310) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tracolor = "black";
        } else if (l2 >= 310 && l2 <= 330) {
            tracolor = "space";
        } else if (l2 > 330) {
            tracolor = "blue";
        }
    }
  //pause ใช้สำหรับหยุดการหมุนของล้อทั้งสอง
    public static void pause() {
        Motor.B.stop();
        Motor.C.stop();
    }
  //ahead ใช้สำหรับตั้งค่าMotorให้เดินตรงไปข้างหน้า
    public static void ahead() {
        Motor.C.setSpeed(100);
        Motor.B.setSpeed(100);
        Motor.B.backward();
        Motor.C.backward();
    }
  //turnLeft ใช้สำหรับตั้งค่าMotorให้เลี้ยวไปทางซ้าย
    public static void turnLeft() {
        Motor.C.setSpeed(10);
        Motor.B.setSpeed(100);
        Motor.C.forward();
        Motor.B.backward();
    }
  //turnRight ใช้สำหรับตั้งค่าMotorให้เลี้ยวไปทางขวา
    public static void turnRight() {
        Motor.C.setSpeed(100);
        Motor.B.setSpeed(10);
        Motor.C.backward();
        Motor.B.forward();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
// scanValue ใช้สำหรับเก็บค่าความสว่างจากLightSensorทุกตัว และมีการแสดงที่เก็บได้ทางLCD
    public static void scanValue() {
        l1 = light1.readNormalizedValue(); //ObjectScanner
        l2 = light2.readNormalizedValue(); //Tracker
        l3 = light3.readNormalizedValue(); //TargetScanner
        LCD.drawString("Obj", 0, 1);
        LCD.drawString("Tra", 0, 3);
        LCD.drawString("Tar", 0, 5);
        LCD.drawInt(l1, 4, 1);
        LCD.drawInt(l2, 4, 3);
        LCD.drawInt(l3, 4, 5);
        checkObject();
        checkTrack();
        checkTarget();
        LCD.drawString(objcolor, 8, 1);
        LCD.drawString(tracolor, 8, 3);
        LCD.drawString(tarcolor, 8, 5);
    }
// trackMove ทำการรับค่าที่แปลงจากTargetScannerมาเป็นตัวบ่งบอกว่าจะเคลื่อนที่ไปทางไหน
    public static void trackMove() {
        {
            if (tracolor.equals("black")) {
                turnLeft();
            } else if (tracolor.equals("blue")) {
                turnRight();
            } else {
                ahead();
            }
        }
    }
    //deploy ใช้สำหรับปล่อยวัสดุที่เก็บไว้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public static void deploy() {
        Sound.twoBeeps();
        step();
        Motor.A.rotate(-90, false);
    }
//step ใช้ในแนวคิดเรื่องการใช้เซนเซอร์ตัวเดียวทำหลายหน้าที่
    public static void step() {
        pause();
        Motor.B.rotate(-150 * i, true);
        Motor.C.rotate(-150 * i, false);
        scanValue(); // ใช้เป็น ตรวจสอบเมื่อเข้าสู่เป้าหมาย
    }
    public static void step2() {
        pause();
        Motor.B.rotate(-20, true);
        Motor.C.rotate(-20, false);
        scanValue(); // ใช้เป็น ตรวจสอบเมื่อเข้าสู่เป้าหมาย
    }
    private static void updateColor() {
        if (objcolor.equals("violet")) {
            i = violet;
            violet++;
        } else if (objcolor.equals("orange")) {
            i = orange;
            orange++;
        } else if (objcolor.equals("blue")) {
            i = blue;
            blue++;
        }
        scanValue();//ใช้ตรวจสอบว่าอยู่นอกเป้าหมายหรือไม่
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การทดสอบการรอรับคำสั่งจากผู้ใช้

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Peridot
 */
import lejos.nxt.*;
public class ManualMode {
   public static String tracolor = "c";
   public static String index = "orange";
   public static String objcolor = "a";
   public static String tarcolor = "b";
   public static String color = "default";
   public static int l1 = 0;
   public static int l2 = 0;
   public static int l3 = 0;
   public static int orange = 1;
   public static int violet = 1;
   public static int blue = 1;
   public static int i = 1;
   public static LightSensor light1 = new LightSensor(SensorPort.S1);
   public static LightSensor light2 = new LightSensor(SensorPort.S2);
   public static LightSensor light3 = new LightSensor(SensorPort.S3);
   public static void main(String[] args) throws Exception {
      while (true) {
         if (Button.ESCAPE.isPressed()) {
            break;
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD.drawString("Outer", 2, 4);
while (true) {
  LCD.drawString("Loop 1", 8, 1);
  if (Button.ENTER.isPressed()) {
    break;
  }
  if (Button.RIGHT.isPressed()) {
    changeColor();
  }
  if (Button.LEFT.isPressed()) {
    changeColor();
  }
  if (Button.ESCAPE.isPressed()) {
    break;
  }
  LCD.drawString(index, 2, 4);
}
while (true) {
  LCD.drawString("Loop 2", 8, 1);
  if (Button.ESCAPE.isPressed()) {
    break;
  }
  scanValue();
  if (!(tarcolor.equals("target"))) {
    step2();
  }
  if (tarcolor.equals(index)) {
    pause();
    Sound.beep();

    break;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        trackMove();
      }
      while (true) {
        LCD.drawString("Loop 3", 8, 1);
        if (Button.ESCAPE.isPressed()) {
          break;
        }
        scanValue();
        if (!(objcolor.equals("object"))) {
          Sound.beepSequence();
          step2();
          step2();
          break;
        }
      }
      while (true) {
        LCD.drawString("Loop 4", 2, 7);
        if (Button.ESCAPE.isPressed()) {
          break;
        }
        scanValue();
        if (!(tarcolor.equals("target"))) {
          step2();
        }
        if (tarcolor.equals(objcolor)) {
          Sound.beep();
          updateColor();
          if (i > 3) {
            Sound.beepSequence();
          } else {
            deploy();
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
          }
          trackMove();
        }
      }
    }
    private static void changeColor() {
      if (index.equals("orange")) {
        index = "blue  ";
      } else if (index.equals("blue  ")) {
        index = "violet";
      } else if (index.equals("violet")) {
        index = "orange";
      }
    }
    public static void trackMove() {
      {
        if (tracolor.equals("black")) {
          turnLeft();
        } else if (tracolor.equals("blue")) {
          turnRight();
        } else {
          ahead();
        }
      }
    }
    public static void ahead() {
      Motor.C.setSpeed(100);
      Motor.B.setSpeed(100);
      Motor.B.backward();
      Motor.C.backward();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    static void turnLeft() {
        Motor.C.setSpeed(10);
        Motor.B.setSpeed(100);
        Motor.C.forward();
        Motor.B.backward();
    }
//turnRight ใช้สำหรับตั้งค่าMotorให้เลี้ยวไปทางขวา
    public static void turnRight() {
        Motor.C.setSpeed(100);
        Motor.B.setSpeed(10);
        Motor.C.backward();
        Motor.B.forward();
    }
    public static void pause() {
        Motor.B.stop();
        Motor.C.stop();
    }
    public static void scanValue() {
        l1 = light1.readNormalizedValue(); //ObjectScanner
        l2 = light2.readNormalizedValue(); //Tracker
        l3 = light3.readNormalizedValue(); //TargetScanner
        LCD.drawString("Obj", 0, 1);
        LCD.drawString("Tra", 0, 3);
        LCD.drawString("Tar", 0, 5);
        LCD.drawInt(l1, 4, 1);
        LCD.drawInt(l2, 4, 3);
        LCD.drawInt(l3, 4, 5);
        checkObject();
        checkTrack();
        checkTarget();
        LCD.drawString(objcolor, 8, 1);
        LCD.drawString(tracolor, 8, 3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    LCD.drawString(tarcolor, 8, 5);
  }
  public static void checkTrack() {
    if (l2 < 310) {
      tracolor = "black";
    } else if (l2 >= 310 && l2 <= 330) {
      tracolor = "space";
    } else if (l2 > 330) {
      tracolor = "blue";
    }
  }
  public static void checkTarget() {
    if (l3 >= 350 && l3 <= 420) {
      tarcolor = "blue  ";
    } else if (l3 >= 470 && l3 <= 500) {
      tarcolor = "orange";
    } else if (l3 >= 440 && l3 <= 460) {
      tarcolor = "red   ";
    } else {
      tarcolor = "target";
    }
  }
  public static void checkObject() {
    if (l1 >= 370 && l1 <= 396) {
      objcolor = "blue  ";
    } else if (l1 >= 405 && l1 <= 475) {
      objcolor = "orange";
    } else if (l1 >= 316 && l1 <= 360){
      objcolor = "object";
    }
  }
  public static void step() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน สุวิชา จึงเจริญสุขยิ่ง

วันเดือนปีเกิด 2 สิงหาคม 2528

สถานที่เกิด สุพรรณบุรี

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนนวมินทราชินูทิศปีการศึกษา 2541

ศึกษาระดับปริญญาตรี วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณหารลาดกระบัง

ปีการศึกษา 2548

ชื่อผู้เขียน อานนท์ เจียสิริมงคล

วันเดือนปีเกิด 21 มิถุนายน 2530

สถานที่เกิด กรุงเทพฯ

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนปทุมคงคา ปีการศึกษา 2542

ศึกษาระดับปริญญาตรี วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้