สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

**Intelligent E-Logistics with RFID**

พุทธพงษ์ เชี่ยวเวช

รัชนีวรรณ บุญชัยสุข

ปก
W8925
2550

เลขหมู่.....
เลขทะเบียน 83030
วัน,เดือน,ปี 30 ก.ค. 2551

ธ 11959708
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

Intelligent E-Logistics with RFID

ผู้จัดทำ

1. นายพุทธพงษ์ เชี่ยวเวช รหัสประจำตัว 47010536
2. นางสาว รัชนีวรรณ บุญชัยสุข รหัสประจำตัว 47010620

_____ อาจารย์ที่ปรึกษา

( ผศ.ดร.วิศิษฏ์ หิรัญกิตติ )

# ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

นายพุทธพงษ์ เชี่ยวเวช 47010536
นางสาวรัชนีวรรณ บุญชัยสุข 47010620
ผศ.ดร.วิศิษฏ์ หิรัญกิตติ อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

## บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาเทคโนโลยีระบุเอกลักษณ์ด้วยความถี่วิทยุ (RFID) แล้วนำมาประยุกต์ใช้ในระบบขนส่งสิ่งของ และศึกษาการพัฒนาโปรแกรมด้วยภาษาไอรอนไพธอน (IronPython) เพื่อให้ได้ใช้ประโยชน์จากการใช้รูปแบบการเขียนโปรแกรมด้วยภาษาไพธอน (Python) ในขณะเดียวกันก็ได้ใช้ประโยชน์จากไลบรารี่ของดอทเน็ต (.NET) เพื่อติดต่อกับอุปกรณ์อาร์เอฟไอดี พร้อมทั้งทรัพยากรอื่นของดอทเน็ตด้วย การศึกษานี้ได้พัฒนาโปรแกรมจำลองการทำงานของระบบขนส่งให้อ่านแท็กอาร์เอฟไอดีที่ใช้ระบุสิ่งของ เก็บลงฐานข้อมูล ใช้อัลกอริทึมเอสตาร์ (A*) ในการค้นหาเส้นทางที่สั้นที่สุดในการขนส่ง ติดตามการขนส่งตั้งแต่สิ่งของเข้ามาในระบบจากผู้ส่งจนถึงปลายทางผู้รับผ่านเว็บแอพพลิเคชั่น บันทึกการเปลี่ยนแปลงตลอดกระบวนการขนส่ง เพื่อให้สามารถนำข้อมูลต่างๆมาวิเคราะห์เพื่อการวางแผนการส่งสิ่งของให้มีประสิทธิภาพ

**Intelligent E-Logistics with RFID**

Mr. Puttapong Chiewvech 47010536

Ms. Ratchaneewan Boonchaisuk 47010620

Asst.Prof.Dr. Visit Hirankitti Advisor

Academic Year 2007

## ABSTRACT

In this thesis we study the Radio-Frequency Identification (RFID) technology with an aim to apply the advantage of Auto-ID for logistics and also study IronPython to take the advantage of Python programming and .NET library to interface with Intermec RFID equipment, besides other .NET environment of the .NET Framework. We design the system that capable of tracking movement of items being delivered via web application starting from it entering the logistic process until it is delivered and received by a recipient, we develop this application in IronPython and use A* algorithm to find out the shortest path for transportation. The system collects the shipping data into database and records all changes that occur throughout the process so the user can analyze those data to improve the delivery planning.

# กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยความเมตตากรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.วิศิษฏ์ หิรัญกิตติ ที่เสียสละ เอื้อเฟื้อ ให้ความช่วยเหลือ สนับสนุน ทั้งด้านความรู้ อุปกรณ์การทดลอง ตลอดจนคอยให้คำชี้แนะแนวทางการแก้ปัญหาต่างๆ ให้ผู้ทำโครงงานได้ฝึกใช้ความคิด ความรู้ ความสามารถที่เรียนมา จนเกิดประสบการณ์ที่ดีในการดำรงชีวิตต่อไปได้

ขอขอบพระคุณ อาจารย์เกียรติณรงค์ ทองประเสริฐ อาจารย์ที่ปรึกษาร่วม อีกทั้งผู้ชื่นชอบการใช้ภาษาไพธอนและภาษาไอรอนไพธอนทั่วโลก ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้

ท้ายสุดนี้ คุณงามความดีและกุศลใดที่พึงบังเกิดมีจากปริญญานิพนธ์ฉบับนี้ เป็นผลมาจากความเมตตา กรุณา ของบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้ทำโครงงาน จึงขอยกคุณความดีเหล่านั้น เป็นเครื่องบูชาพระคุณ ด้วยความเคารพและสักการะยิ่ง

พุทธพงษ์ เชี่ยวเวช
รัชนีวรรณ บุญชัยสุข

# สารบัญ

หน้า


บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาของปัญหา.....1

1.2 วัตถุประสงค์ของการศึกษา.....1

1.3 ขอบเขตของงานวิจัย.....2

1.4 แผนการดำเนินงาน.....2

1.4.1 ขั้นศึกษาโครงการ.....2

1.4.2 ขั้นวางแผนออกแบบและดำเนินการ.....2

1.4.3 ขั้นทดลองหรือทดสอบประสิทธิภาพ.....2

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....4

2.1 โลจิสติกส์.....4

2.1.1 ความหมายของการจัดการโลจิสติกส์.....4

# สารบัญ (ต่อ)

หน้า

2.1.2 ประเภทของการจัดการโลจิสติกส์.....4

2.1.3 ความสำคัญของการจัดการโลจิสติกส์.....5

2.2 อาร์เอฟไอดี (RFID).....7

2.2.1 องค์ประกอบของระบบ RFID.....7

2.2.2 การทำงานของระบบ RFID.....15

2.2.3 ข้อดีข้อเสียของRFID.....17

2.2.4 มาตรฐานของ RFID.....19

2.3 ไอรอนไพธอน (IronPython).....22

2.3.1 เส้นทางการพัฒนาของภาษา IronPython.....23

2.3.2 ขอบเขตการใช้งาน.....23

2.3.3 วิธีการใช้งาน.....25

2.4 การเชื่อมต่อกับตัวอ่านอย่างง่าย.....27

2.4.1 คุณสมบัติของ BRI.....27

2.4.2 รูปแบบการเรียกใช้ BRI.....28

2.4.3 การพัฒนาโปรแกรมโดยใช้ BRI.....28

บทที่ 3 การออกแบบระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID.....33

3.1 จุดประสงค์การออกแบบ.....33

3.2 การออกแบบ.....33

# สารบัญ (ต่อ)

หน้า

3.2.1 การออกแบบระบบโลจิสติกส์.....33

3.2.2 การออกแบบคลาสไดอะแกรม.....42

บทที่ 4 การดำเนินงาน.....45

4.1 การติดต่ออุปกรณ์อาร์เอฟไอดี.....45

4.2 การจัดการฐานข้อมูล.....47

4.3 อัลกอริทึมในการค้นหาเส้นทางส่งสิ่งของ.....48

บทที่ 5 ผลการทำงานของระบบ.....50

5.1 แอพพลิเคชั่นในส่วนของสถานี.....50

5.2 แอพพลิเคชั่นในส่วนของแวร์เฮาส์.....56

5.3 แอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของ.....58

5.4 เว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ.....62

บทที่ 6 บทสรุปและวิจารณ์.....63

6.1 บทสรุปและวิจารณ์.....63

6.2 ปัญหาที่เกิดขึ้นในการพัฒนา.....64

6.3 แนวทางการพัฒนา.....64

บรรณานุกรม.....65

# สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงย่านความถี่ต่างๆของระบบ RFID และตัวอย่างการใช้งาน.....12

2.2 แสดงข้อเปรียบเทียบลักษณะของเทคโนโลยี RFID และ Barcode.....18

2.3 แสดงเงื่อนไขที่มีส่วนตัดสินใจว่า RFID จะมาแทนหรือมาเสริม Barcode.....19

2.4 แสดงการเปรียบเทียบมาตรฐานระหว่าง ISO/IEC กับ EPC.....22

2.5 แสดงคลาสในดอทเน็ตเฟรมเวิร์ก (Common .NET Assemblies and Namespaces).....24

2.6 แสดงการเขียนโค้ด Hello World ง่ายๆ เปรียบเทียบระหว่าง C# และ IronPython.....26

4.1 แสดงอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ.....45

# สารบัญรูป

รูปที่ หน้า

2.1 แสดงตัวอย่างแท็กชนิดแอ็กทีฟ.....8

2.2 แสดงตัวอย่างแท็กชนิดกึ่งพาสซีฟ.....9

2.3 แสดงตัวอย่างแท็กชนิดพาสซีฟ.....10

2.4 แสดงคลื่นความถี่ที่ใช้ในระบบ RFID.....11

2.5 แสดงตัวอย่างการใช้งานคลื่นความถี่ในย่านต่างๆ.....12

2.6 แสดงตัวอย่างเครื่องอ่านแบบติดตั้งยี่ห้อหนึ่ง.....14

2.7 แสดงตัวอย่างเครื่องอ่านแบบพกพายี่ห้อหนึ่ง.....14

2.8 แสดงการปล่อยคลื่นแม่เหล็กไฟฟ้าวิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า
Inductive Coupling) และวิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า
(Electromagnetic Propagation Coupling).....16

2.9 แสดงโครงสร้างของแท็กตามมาตรฐาน EPC-96.....21

2.10 แสดงการคอมไพล์โค้ดภาษาไพธอนภายใต้ IronPython Engine.....25

3.1 แสดงการทำงานของระบบ.....33

3.2 แสดง Use Case Diagram ส่วนควบคุมจัดการระบบขนส่ง.....35

3.3 แสดง Use Case Diagram ของการตรวจสอบและติดตามสิ่งของ.....37

3.4 แสดง Sequence diagram ของ Register delivered object.....38

# สารบัญรูป (ต่อ)

รูปที่ หน้า

3.5 แสดง Sequence diagram ของ Delivery.....39

3.6 แสดง Sequence diagram ของ Hand In.....40

3.7 แสดง Sequence diagram ของ Track the delivered object.....41

3.8 แสดง Class Diagram ของระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี.....42

3.9 แสดงตาราง Parcel.....43

3.10 แสดงตาราง Parcel Status.....43

3.11 แสดงตาราง Car.....44

3.12 แสดงตาราง Address.....44

3.13 แสดงตาราง Postcode.....44

4.1 แสดงการติดต่อระหว่าง Server และ Client (Handheld RFID Reader) ผ่าน wireless.....46

4.2 แสดงทดลองการอ่านแท็กด้วย Reader แบบ Handheld และส่งข้อมูลเข้าเครื่อง Server.....46

4.3 แสดงการทดลองอ่านแท็กด้วย Reader แบบ Fixed.....47

4.4 แสดงการใช้อัลกอริทึมเอสตาร์ในการค้นหาเส้นทางเมื่อกำหนดให้ผ่านหลายจุด.....49

5.1 แสดงฟังก์ชันการทำงานโดยรวมของระบบ.....50

5.2 แสดงแอพพลิเคชั่นในส่วนของสถานี : หน้าล็อกอินเข้าใช้งาน.....51

5.3 แสดงแอพพลิเคชั่นในส่วนของสถานี : เลือกการทำงาน.....51

5.4 แสดงแอพพลิเคชั่นในส่วนของสถานี : รับสิ่งของเข้าระบบ.....52

# สารบัญรูป (ต่อ)

รูปที่ หน้า

5.5 แสดงแอพพลิเคชั่นในส่วนของสถานี: สถานะของสิ่งของที่ได้รับภายในสถานี.....53

5.6 แสดงแอพพลิเคชั่นในส่วนของสถานี : แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ.....55

5.7 แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : รับสิ่งของเข้าแวร์เฮ้าส์.....56

5.8 แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : จัดสิ่งของขึ้นรถ.....57

5.9 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Get in .....58

5.10 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Route .....59

5.11 แสดงแผนที่ที่รถจัดส่งสิ่งของต้องวิ่งผ่าน.....60

5.12 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Send .....61

5.13 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำสถานีย่อย.....61

5.14 แสดงเว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ.....62

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาของปัญหา

การขนส่งวัตถุดิบและสิ่งของ ที่ รวดเร็ว ประหยัด และมีประสิทธิภาพ นับเป็นยุทธศาสตร์สำคัญประการหนึ่งในการพัฒนาอุตสาหกรรมและเศรษฐกิจของชาติ โดยเฉพาะอย่างยิ่งค่าใช้จ่ายในการขนส่งถือว่าเป็นต้นทุนสำคัญในการดำเนินธุรกิจแทบทุกประเภท ซึ่งการขนส่งมีทั้งรูปแบบตั้งแต่ การขนส่งวัตถุดิบ การขนส่งสิ่งของผลิตภัณฑ์ การขนส่งเอกสาร ฯลฯ แต่เมื่อพิจารณาให้ดีแล้วจะพบว่า ในรูปแบบต่างๆ เหล่านี้มีลักษณะที่คล้ายคลึงกันโดยส่วนใหญ่คือ เป็นการขนส่งสิ่งๆ หนึ่งจากจุดเริ่มต้นไปยังจุดหมาย ดังนั้นระบบขนส่งที่ดีควรสามารถรองรับการขนส่งในรูปแบบต่างๆได้โดยการปรับเปลี่ยนน้อยที่สุด นอกจากนี้การบริหารจัดการวางแผนการขนส่งที่มีประสิทธิภาพจะช่วยให้ประหยัดทรัพยากร ทั้งเวลา เชื้อเพลิง แรงงาน และท้ายที่สุดคือ ลดต้นทุนที่สำคัญในการดำเนินธุรกิจ นำไปสู่ความสามารถในการแข่งขันในตลาดการค้าเสรีในปัจจุบันและอนาคต

การพัฒนาระบบขนส่งโดยการนำเทคโนโลยีอาร์เอฟไอดีที่ใช้ระบุเอกลักษณ์มาใช้งานในการระบุถึงตัวสินค้า เพื่อมาแทนที่การใช้งานบาร์โค้ด ซึ่งจะช่วยลดเวลาในการทำงานของระบบให้น้อยลงและระบบมีความน่าเชื่อถือมากขึ้น โดยการพัฒนาระบบต้องทำให้ระบบมีความสามารถในการวางแผนการขนส่งได้อย่างมีประสิทธิภาพ สามารถค้นหาเส้นทางที่มีความเหมาะสมที่สุดในการขนส่งสินค้าได้

## 1.2 วัตถุประสงค์ของงานวิจัย

1. ศึกษาเทคโนโลยีระบุเอกลักษณ์ด้วยความถี่วิทยุ (RFID) เพื่อนำมาพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาด
2. ศึกษาภาษาไอรอนไพธอนเพื่อใช้ประโยชน์ของภาษาไพธอนและใช้ไลบรารี่ (Library) ของดอทเน็ต (.NET)
3. พัฒนาแอพพลิเคชั่นระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ อาร์เอฟไอดีด้วยภาษาไอรอนไพธอน

## 1.3 ขอบเขตของงานวิจัย

1 ศึกษาการทำงานของเทคโนโลยีอาร์เอฟไอดีและพัฒนาแอพพลิเคชั่นในการติดต่ออุปกรณ์อาร์เอฟไอดี

2 พัฒนาแอพพลิเคชั่นจำลองระบบขนส่งสินค้าโดยใช้เทคโนโลยีอาร์เอฟไอดีเพื่อระบุถึงตัวสินค้า

3 พัฒนาแอพพลิเคชั่นจำลองระบบขนส่งสินค้าโดยมีความสามารถในการค้นหาเส้นทางการขนส่งสินค้าที่มีความเหมาะสม ผู้ใช้งานสามารถตรวจสอบสถานะการขนส่งและสถานที่ที่สินค้าอยู่ ณ เวลาปัจจุบัน

## 1.4 แผนการดำเนินงาน

### 1.4.1 ขั้นศึกษาโครงการ

1 ศึกษาเทคโนโลยี อาร์เอฟไอดี และข้อมูลพื้นฐานในการใช้งานอุปกรณ์อาร์เอฟไอดี

2 ศึกษาการเขียนการใช้งานภาษาไอรอนไพธอน

3 ศึกษามิดเดิลแวร์ (Middleware) หรือเอพีไอ (API) ที่เหมาะสมกับอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

4 ศึกษาข้อมูลเกี่ยวกับระบบขนส่งสิ่งของ

### 1.4.2 ขั้นวางแผนออกแบบและดำเนินงาน

1 ออกแบบระบบฐานข้อมูลที่จะใช้ในระบบ

2 เขียนแอพพลิเคชั่นพัฒนาส่วนสื่อสารข้อมูลติดต่อฐานข้อมูลระหว่างเครื่องเซิร์ฟเวอร์ (Server) และไคลเอนต์ (Client)

3 เขียนแอพพลิเคชั่นพัฒนาในส่วนการติดต่ออุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

4 เขียนแอพพลิเคชั่นพัฒนาในส่วนควบคุมและจัดการระบบขนส่งที่เกี่ยวข้อง

5 ทำ Prototype ของระบบ

### 1.4.3 ขั้นทดสอบการทำงานของระบบ

1 ทดสอบการสื่อสารข้อมูลระหว่างเครื่องเซิร์ฟเวอร์ (Server) และไคลเอนต์ (Client)

2 ทดสอบการทำงานของแอพพลิเคชั่นในส่วนของการติดต่ออุปกรณ์อาร์เอฟไอดี

3 ทดสอบการทำงานของแอพพลิเคชั่นในส่วนของควบคุมและจัดการระบบขนส่ง

4 ค้นหาและปรับปรุงข้อผิดพลาดเพื่อพัฒนาประสิทธิภาพของระบบ

5 รายงานสรุปผลโครงการพร้อมทั้งจัดทำเอกสารประกอบ

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1 ได้ความรู้ทางด้านระบบโลจิสติกส์และเทคโนโลยีอาร์เอฟไอดี

2 ได้เรียนรู้การใช้ภาษาไอรอนไพธอนในการพัฒนาโปรแกรม โดยใช้รูปแบบการเขียนแบบภาษาไพธอน อีกทั้งยังสามารถเรียกใช้ไลบรารี่หรือทรัพยากรอื่นๆของดอทเน็ต (.NET framework) ได้

3 สามารถออกแบบพัฒนาระบบขนส่งชาญฉลาด ที่ช่วยลดต้นทุนและใช้ทรัพยากรได้อย่างมีประสิทธิภาพ

4 ได้เรียนรู้วิธีการทำงาน พร้อมทั้งวิธีการแก้ปัญหาที่เกิดขึ้น เพื่อสามารถนำไปประยุกต์ใช้ในชีวิตจริงได้

# บทที่ 2
# ทฤษฎีที่เกี่ยวข้อง

## 2.1 โลจิสติกส์

### 2.1.1 ความหมายของการจัดการโลจิสติกส์

คำว่า โลจิสติกส์ (Logistics) แปลว่า การส่งกำลังบำรุง ซึ่งความหมายในยุคศตวรรษที่ 19 ในเรื่องที่เกี่ยวกับการขนส่งสิ่งของ หมายถึง การวางแผนและบริหารจัดการเพื่อลำเลียงสิ่งของจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยเฉพาะในทางทหารและอุตสาหกรรมการผลิตสิ่งของ แต่ถ้าในเรื่องของการบริหารองค์กรที่มีสายงานมากและซับซ้อนแล้วคำว่าโลจิสติกส์จะหมายถึง การวางแผนและบริหารจัดการภารกิจที่มีความยุ่งยากซับซ้อน

เมื่อยุคสมัยเปลี่ยนไป ความหมายของ โลจิสติกส์ ก็เปลี่ยนไปตามวิธีการดำเนินธุรกิจของโลก เรียกว่า การจัดการโลจิสติกส์ธุรกิจ มีผู้ให้คำนิยามไว้มากมาย ดังเช่นตัวอย่างจาก Oak Brook, IL: Council of Logistic Management, 1993 นักวิชาการชาวออสเตรเลียกล่าวไว้ว่า

"การจัดการโลจิสติกส์ธุรกิจ คือ กระบวนการวางแผน การปฏิบัติการและควบคุม การเคลื่อนย้าย และการจัดเก็บสิ่งของอย่างมีประสิทธิภาพและประสิทธิผล รวมถึงการให้บริการและสารสนเทศที่เกี่ยวข้องตั้งแต่จุดกำเนิดจนถึงจุดการบริโภคสิ่งของ เพื่อวัตถุประสงค์ในการตอบสนองความต้องการของลูกค้า"

"Business Logistics Management is the Process of Planning, Implementing and Controlling the efficient, Effective flow and storage of Storage of Goods, Services and Related Information from Point of Origin to Point of Conforming to Customer Requirements."

### 2.1.2 ประเภทของการจัดการโลจิสติกส์

การจัดการโลจิสติกส์สามารถแบ่งเป็น 3 ประเภทใหญ่ ๆ ตามลักษณะของบทบาทหน้าที่มหภาคของประเทศได้ดังนี้

**2.1.2.1 การจัดการโลจิสติกส์ด้านการทหาร (Military Logistics Management)**หมายถึง การจัดการการจัดส่งกำลังบำรุงด้านการทหาร เช่น ยุทโธปกรณ์ ปัจจัยสี่ การรักษาพยาบาลและสารสนเทศ เพื่อมุ่งหวังชัยชนะทางทหารเป็นสำคัญ

**2.1.2.2 การจัดการลอจิสติกส์ด้านวิศวกรรม (Engineering Logistics Management)** หมายถึง การจัดการด้านการวิศวกรรมจัดส่งลำเรียง อันได้แก่การสร้าง การบูรณาการและการบำรุงรักษาสาธารณูปการ ทั้งระบบขนส่งทางบก ทางน้ำ ทางอากาศ และทางท่อ ระบบการจัดเก็บและระบบสารสนเทศ เพื่อมุ่งหวังความพร้อมในระบบการจัดเก็บและการจัดส่งลำเรียงทั้งระบบเป็นสำคัญ

**2.1.2.3 การจัดการโลจิสติกด้านธุรกิจ (Business Logistics Management)**หมายถึง การจัดการด้านการจัดเก็บและจัดส่งสิ่งของ คน สัตว์ สิ่งของ จากจุดหนึ่งไปยังจุดหนึ่งตามที่มนุษย์ต้องการ เพื่อมุ่งหวังความสำเร็จทางธุรกิจเป็นสำคัญ

**2.1.3 ความสำคัญของการจัดการโลจิสติกส์**

ปัจจุบัน การจัดการโลจิสติกส์ ต้องคำนึงถึง ต้นทุนด้านการขนส่งของประเทศ การผลิตสิ่งของหรือการบริการต่างๆ ย่อมต้องมีการติดต่อ ขนส่ง เช่น ขนส่งวัตถุดิบจากแหล่งวัตถุดิบ ไปยังโรงงานผ่านกระบวนการผลิตจนเป็นสิ่งของ จากนั้นต้องมีการขนส่งสิ่งของสู่ตลาด เพื่อกระจายให้ถึงผู้บริโภค ต้นทุนด้านการขนส่งมิได้หมายถึงเฉพาะค่าใช้จ่ายของยานพาหนะ แต่รวมถึงวิธีการบรรจุหีบห่อ ขนถ่าย และป้อนเข้าโรงงาน หากทำได้รวดเร็ว ประหยัด มีการสูญเสียน้อย นั่นย่อมหมายถึงมีต้นทุนต่ำ ในการกระจายผลผลิตสู่ตลาดและผู้บริโภคก็ต้องมีต้นทุนต่ำด้วย

ดังนั้น หน่วยงานที่เกี่ยวข้องกับการจัดการ โลจิสติกส์ จึงครอบคลุมหลายฝ่าย ตั้งแต่แหล่งวัตถุดิบ วิธีการบรรจุ ขนถ่าย กระบวนการส่ง-รับของ ผู้จัดสร้างโครงสร้างพื้นฐานด้านการขนส่งต่างๆ ทั้งระบบถนน ระบบราง ท่าเรือ ท่าอากาศยาน ศุลกากร และโรงเก็บสิ่งของ เป็นต้น ต้นทุนด้านการขนส่งจะต่ำได้ต่อเมื่อการขนถ่ายและนำส่งผลิตภัณฑ์ถึงจุดหมายโดยเร็ว สูญเสียน้อย สิ่งของถึงมือผู้รับตามเวลาโดยเร็ว ขั้นตอนกระบวนการศุลกากรทั้งนำเข้า-ส่งออก สะดวกรวดเร็ว ไม่ต้องเสียค่าเช่าโรงเก็บสิ่งของหรือตู้คอนเทนเนอร์ นานวัน ลดดอกเบี้ยของต้นทุนลงได้ด้วย

การติดต่อสื่อสาร เพื่อการสั่งซื้อทั้งวัตถุดิบและสิ่งของ ต้องสะดวกรวดเร็วและชัดเจน ระบบการถ่ายทอดส่งข้อมูลสารสนเทศ หรือ ไอที ต้องมีเครือข่ายที่เชื่อมต่อทั้งในประเทศและกับต่างประเทศอย่างทั่วถึง และเชื่อมโยงกับระบบอื่น ๆ ตั้งแต่แหล่งวัสดุ โรงงานผลิต กรมศุลกากรจนถึงผู้ซื้อ และมีมาตรฐานที่เป็นสากล เพื่อมิให้มีการผิดพลาดเกิดขึ้น ไม่สูญเสีย สามารถประหยัดต้นทุนสิ่งของได้ ประเทศต่าง ๆ ในโลกกำลังรณรงค์ด้านโลจิสติกส์ เพื่อลดต้นทุนผลผลิตของประเทศโดยเฉพาะด้านการขนส่ง

เมื่อมองภาพรวมของประเทศ คงต้องแบ่งเป็นสองส่วนด้วยกันคือในส่วนของภาครัฐที่จะต้องสนับสนุนระบบโลจิสติกส์ให้เชื่อมประสานกับระบบโลจิสติกส์ภายในของผู้ประกอบการ เพื่อการ

ประหยัดพลังงาน และลดต้นทุนในสิ่งของ สำหรับการแข่งขันในระดับนานาชาติ ในภาครัฐได้กำหนดแผนบริหารราชการแผ่นดิน (พ.ศ. 2548-2551) นโยบายปรับโครงสร้างเศรษฐกิจให้สมดุลและแข่งขันได้ โดยมีกรอบแผนแม่บทการพัฒนาระบบโลจิสติกส์ (Logistics Master Plan) ใน 6 ประเด็นหลักดังนี้

1. โครงสร้างพื้นฐาน โดยมีเป้าประสงค์
    - เพื่อลดค่าใช้จ่ายโดยรวมของการขนส่งสิ่งของและบริการ
    - ลดระยะเวลาที่ใช้ในการขนส่งสิ่งของและบริการจากต้นทางถึงปลายทาง
    - ยกระดับคุณภาพการให้บริการของท่าเรือหลัก
2. ระบบการเชื่อมโยงข้อมูลและฐานข้อมูล โดยมีเป้าประสงค์
    - ระบบข้อมูลเพื่อการนำเข้าและส่งออกของภาครัฐเชื่อมโยงเป็นระบบเดียวกัน และสามารถเชื่อมต่อกับเอกชน และผู้ใช้บริการ
    - การวางแผนและลงทุนเพื่อพัฒนาโลจิสติกส์ มีความสอดคล้องกับข้อเท็จจริงของประเทศมากยิ่งขึ้น
3. ความสามารถในการบริหารจัดการโลจิสติกส์ (Capacity Building) โดยมีเป้าประสงค์
    - ธุรกิจสามารถประยุกต์ใช้การบริหารจัดการโลจิสติกส์สมัยใหม่ได้อย่างมีประสิทธิ ภาพ
4. ธุรกิจให้บริการโลจิสติกส์ (Logistic Service Providers) โดยมีเป้าประสงค์
    - มีผู้ให้บริการโลจิสติกส์ที่มีคุณภาพ และเป็นที่ยอมรับของผู้ใช้บริการในระดับสากล
5. การเชื่อมโยงทางการค้ากับภูมิภาค โดยมีเป้าประสงค์
    - การเติบโตของการค้าระหว่างไทยกับประเทศในภูมิภาคอาเซียน (ASEAN)
6. กฎหมาย ข้อบังคับ นโยบาย และกระบวนการให้บริการของรัฐ โดยมีเป้าประสงค์
    - มีกฎหมาย ข้อบังคับ นโยบาย ที่สนับสนุนกิจกรรมโลจิสติกส์
    - บริการของหน่วยงานรัฐทางด้านโลจิสติกส์มีมาตรฐานเทียบเท่าสากล

ระบบ RFID ก็เป็นอีกเทคโนโลยีที่เข้ามามีบทบาทสำคัญในการพัฒนาระบบ Logistics ได้เป็นอย่างดี มีข้อดีหลายอย่างเมื่อเทียบกับระบบเก่าๆที่เคยใช้

## 2.2 อาร์เอฟไอดี (RFID)

อาร์เอฟไอดี(RFID) ย่อมาจากคำว่า Radio Frequency Identification เป็นระบบฉลากที่ได้ถูกพัฒนามาตั้งแต่ยุค ค.ศ. 1970s โดยที่อุปกรณ์อาร์เอฟไอดี ที่มีการประดิษฐ์ขึ้นใช้งานเป็นครั้งแรกนั้นเป็นผลงานของ León Theremin ซึ่งสร้างให้กับรัฐบาลของประเทศรัสเซียในปี ค.ศ. 1946 ซึ่งอุปกรณ์ที่สร้างขึ้นมาในเวลานั้นทำหน้าที่เป็นเครื่องมือดักจับสัญญาณ ไม่ได้ทำหน้าที่เป็นตัวระบุเอกลักษณ์อย่างที่ใช้งานกันอยู่ในปัจจุบัน

อาร์เอฟไอดีในปัจจุบันเป็นเทคโนโลยีการระบุข้อมูลที่แสดงเอกลักษณ์ของวัตถุหรือบุคคลด้วยคลื่นความถี่วิทยุที่ได้ถูกพัฒนามาในยุค ค.ศ. 1970s วัตถุประสงค์เพื่อนำไปใช้ในการบ่งชี้วัตถุในระยะไกลได้ โดยมีจุดเด่น คือ สามารถอ่านข้อมูลจากแท็ก (Tag) ได้ครั้งละหลายๆแท็ก โดยไม่จำเป็นต้องอาศัยการสัมผัส (Contact-Less) หรือต้องเห็นวัตถุนั้นๆ ก่อน และสามารถอ่านค่าได้แม้ในสภาพที่ทัศนวิสัยไม่ดี ทนต่อความเปียกชื้น แรงสั่นสะเทือน การกระทบกระแทก และสามารถจะอ่านข้อมูลได้ด้วยความเร็วสูง โดยข้อมูลจะถูกเก็บไว้ในไมโครชิปที่อยู่ในแท็ก ทำงานโดยใช้เครื่องอ่านที่สื่อสารกับแท็กด้วยคลื่นวิทยุในการอ่านและเขียนข้อมูล

### 2.2.1 องค์ประกอบของระบบอาร์เอฟไอดี

ระบบอาร์เอฟไอดีประกอบด้วย 3 ส่วนหลัก

**2.2.1.1 แท็ก (Tag) / ทรานสปอนเดอร์ (Transponder)** ซึ่งมาจากคำว่าทรานสมิตเตอร์ (Transmitter) ผสมกับคำว่าเรสปอนเดอร์ (Responder) ถ้าจะแปลให้ตรงตามศัพท์ แท็กก็จะทำหน้าที่ส่งสัญญาณหรือข้อมูลที่บันทึกอยู่ในแท็กตอบสนองไปที่ตัวอ่านข้อมูล การสื่อสารระหว่างแท็กและตัวอ่านข้อมูลจะเป็นการสื่อสารกันโดยอาศัยช่องความถี่วิทยุผ่านอากาศ

**1. องค์ประกอบของแท็ก**

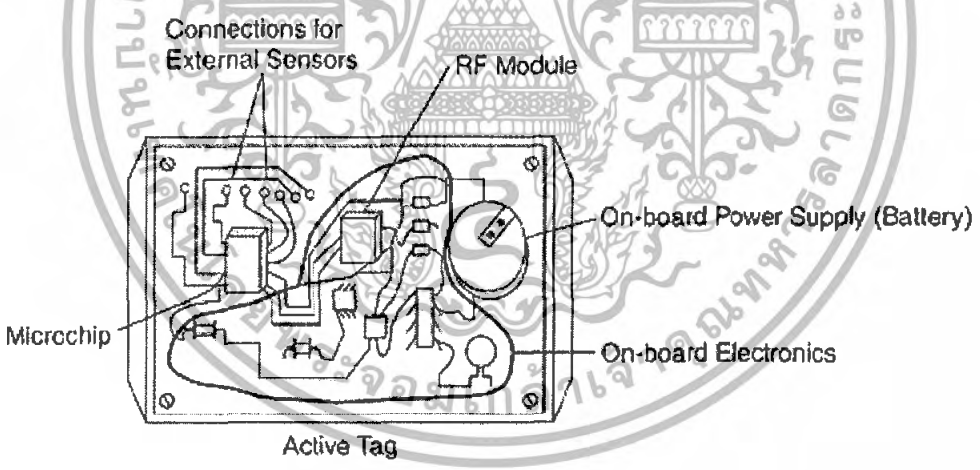
- ส่วนของไอซีซึ่งเป็นชิปสารกึ่งตัวนำ (**Semiconductor Chip**) หรือ ไมโครชิป ทำหน้าที่เก็บข้อมูลของวัตถุ
- ส่วนของขดลวดซึ่งทำหน้าที่เป็นเสาอากาศ **(Antenna)** สำหรับรับส่งสัญญาณคลื่นความถี่วิทยุและสร้างพลังงานป้อนให้ส่วนของไมโครชิป

ทั้งสองส่วนนี้จะเชื่อมต่ออยู่ด้วยกัน โดยทั่วไปตัวแท็กอาจอยู่ในรูปแบบที่เป็นกระดาษ แผ่นฟิล์ม พลาสติก มีขนาดและรูปร่างต่าง ๆ กันไป และมีหลายรูปแบบ ทั้งนี้ขึ้นอยู่กับการนำไปใช้งาน

**2. การแบ่งประเภทของแท็กตามองค์ประกอบ** ทั้งนี้เราสามารถแบ่งแท็กที่มีใช้งานกันอยู่ได้เป็น 3 ชนิดใหญ่ ๆ ได้แก่

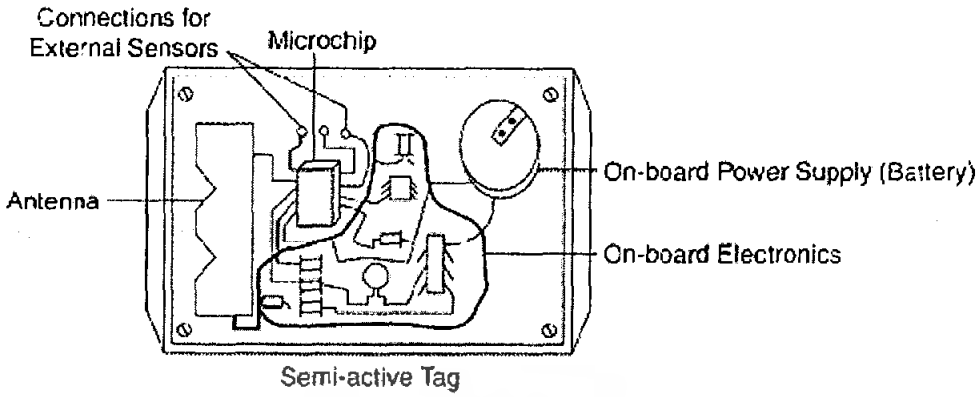
- **แท็กชนิดแอ็กทีฟ (Active Tag)** แท็กชนิดนี้จะมีแบตเตอรี่อยู่ภายในซึ่งใช้เป็นแหล่งจ่ายไฟขนาดเล็ก เพื่อป้อนพลังงานไฟฟ้าให้แท็กทำงานโดยปกติ ส่งสัญญาณแบบบรอดแคส(broadcast) ไปที่เครื่องอ่าน ได้โดยแท็กชนิดนี้มีฟังก์ชั่นการทำงานทั่วไปทั้งอ่านและเขียนข้อมูลลงในแท็กได้ และการที่ต้องใช้แบตเตอรี่จึงทำให้แท็กชนิดแอ็กทีฟมีอายุการใช้งานจำกัดตามอายุของแบตเตอรี่ โดยทั่วไปประมาณ 2-10 ปี เมื่อแบตเตอรี่หมดก็ต้องนำแท็กไปทิ้ง ไม่สามารถนำกลับมาใช้ใหม่ได้ เนื่องจากจะมีการซีล (seal) ที่ตัวแท็กจึงไม่สามารถเปลี่ยนแบตเตอรี่ได้ อย่างไรก็ตามถ้าสามารถออกแบบวงจรของแท็กให้กินกระแสไฟน้อยๆ ก็อาจจะมีอายุการใช้งานนานนับสิบปี

  แท็กชนิดแอ็กทีฟนี้จะมีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ มีกำลังส่งสูงและระยะการรับส่งข้อมูลไกลสูงเป็น 100 เมตร ซึ่งไกลกว่าแท็กชนิดพาสซีฟ นอกจากนี้ยังทำงานในบริเวณที่มีสัญญาณรบกวนได้ดี แม้แท็กชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น ราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด



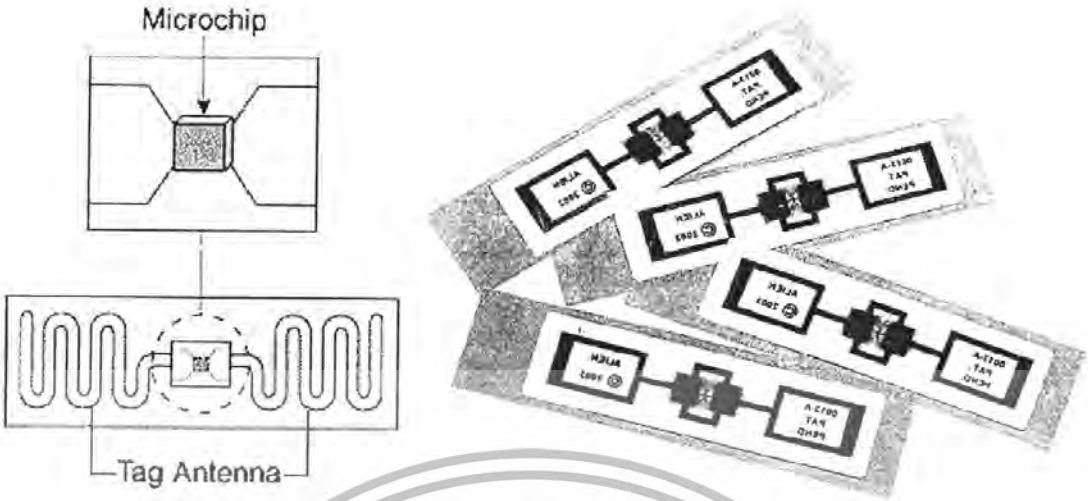

**รูปที่ 2.1** แสดงตัวอย่างแท็กชนิดแอ็กทีฟ

- **แท็กชนิดกึ่งพาสซีฟ (Semi-passive Tag)** แท็กชนิดนี้เหมือนแท็กชนิดแอ็กทีฟตรงที่มีแบตเตอรี่จ่ายไฟให้ไมโครชิป แต่ไม่ได้ส่งสัญญาณแบบบรอดแคสไปที่เครื่องอ่าน ตัวแท็กจะรอรับสัญญาณกระตุ้นให้ทำงานจากเครื่องอ่านแล้วจึงจะส่งข้อมูลกลับไปยังเครื่องอ่านเหมือนแท็กชนิดพาสซีฟ แต่ส่งได้ไกลกว่า มีความแม่นยำกว่าแท็กชนิดพาสซีฟ อีกทั้งยังมีอายุการใช้งานนานกว่าแท็กชนิดแอ็กทีฟด้วย




**รูปที่ 2.2** แสดงตัวอย่างแท็กชนิดกึ่งพาสซีฟ

- **แท็กชนิดพาสซีฟ(Passive Tag)** จะไม่มีแบตเตอรี่อยู่ภายในหรือไม่จำเป็นต้องรับแหล่งจ่ายไฟใด ๆ เพราะจะทำงานโดยอาศัยพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้าจากตัวอ่านข้อมูล (มีวงจรกำเนิดไฟฟ้าขนาดเล็กอยู่ในตัว)หรือที่เรียกว่าอุปกรณ์ Transceiver จึงทำให้แท็กชนิดพาสซีฟมีน้ำหนักเบาและเล็กกว่าแท็กชนิดแอ็กทีฟ ราคาถูกกว่า และมีอายุการใช้งานไม่จำกัด แต่ข้อเสียก็คือระยะการรับส่งข้อมูลใกล้ซึ่งสามารถส่งข้อมูลได้ไกลสุดเพียง 1.5 เมตร ซึ่งเป็นระยะการอ่านที่สั้น มีหน่วยความจำขนาดเล็กซึ่งโดยทั่วไปประมาณ 32 ถึง 128 บิต และตัวเครื่องอ่านข้อมูลจะต้องมีความไวและกำลังที่สูง นอกจากนี้แท็กชนิดพาสซีฟมักจะมีปัญหาเมื่อนำไปใช้งานในสิ่งแวดล้อมที่มีสัญญาณแม่เหล็กไฟฟ้ารบกวนสูงอีกด้วย แต่ข้อได้เปรียบในเรื่องราคาต่อหน่วยที่ต่ำกว่าแท็กชนิดแอ็กทีฟและอายุการใช้งานที่ยาวนานกว่าทำให้แท็กชนิดพาสซีฟนี้เป็นที่นิยมมากกว่า ไอซีของแท็กชนิดพาสซีฟที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็นได้ ไปจนถึงขนาดใหญ่จนสะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกัน




**รูปที่ 2.3** แสดงตัวอย่างแท็กชนิดพาสซีฟ

**3. การแบ่งประเภทของแท็กตามรูปแบบการอ่านหรือบันทึกข้อมูล**

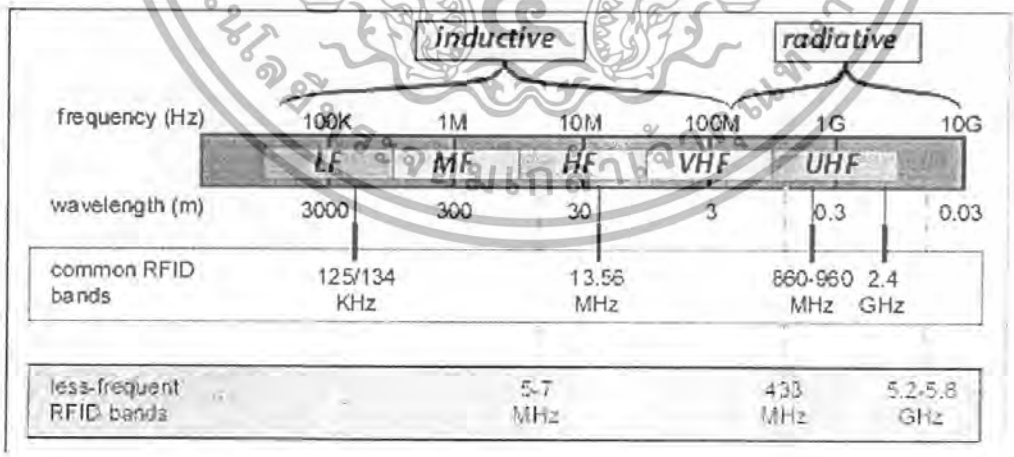
- แท็กชนิดที่สามารถถูกอ่านและเขียนข้อมูลได้หลายครั้ง (Read-Write)
- แท็กชนิดที่เขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างหลายครั้ง (Write-Once-Read-Many: WORM)
- แท็กชนิดที่อ่านได้เพียงอย่างเดียว (Read-Only)

**4. การแบ่งประเภทของแท็กตามความถี่ของการใช้งาน**

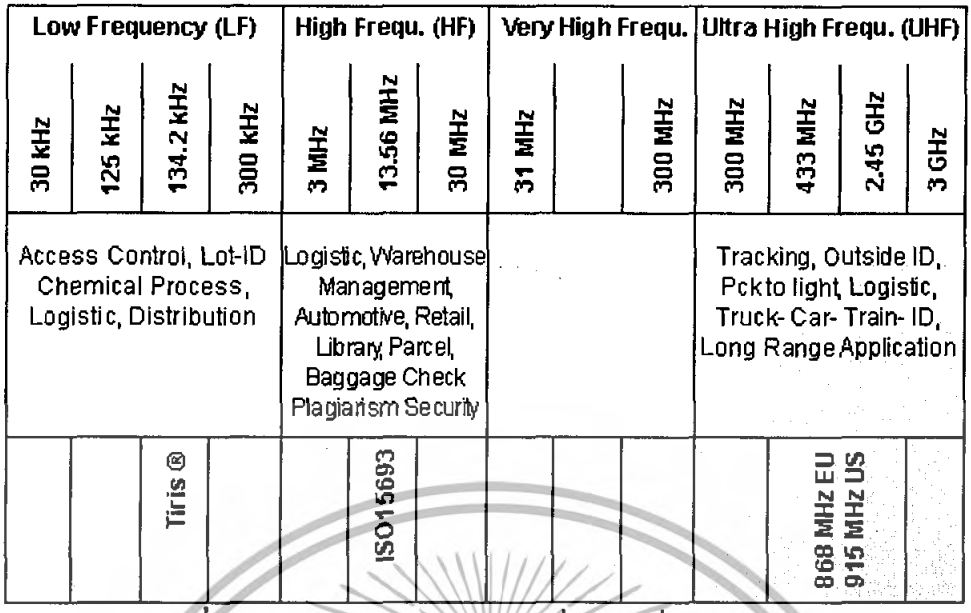
- **แท็กย่านความถี่ต่ำ (LF)** 125 - 134 กิโลเฮิรตซ์ย่านความถี่ต่ำคือมีค่า 30 – 300 KHz แต่ที่ระบบ RFID ใช้คือความถี่ในช่วง 125 – 134 KHz ซึ่งมีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านต่ำ โดยเฉพาะเมื่ออยู่ในสภาพแวดล้อมที่ไม่เหมาะสม เช่น บรรจุน้ำ โลหะ เป็นต้น แท็กที่ใช้ในย่านความถี่นี้มักเป็นแท็กชนิดพาสซีฟ แต่ก็มีใช้แท็กชนิดแอ็กทีฟบ้างขึ้นกับผู้ผลิต เพราะการใช้งานระบบอาร์เอฟไอดี ในย่านความถี่นี้มีมากมาย เป็นที่นิยมใช้กันกว้างขวาง และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก
- **แท็กย่านความถี่สูง (HF)** 13.56 เมกะเฮิรตซ์ ย่านความถี่สูงคือมีค่า 3 – 30 MHz แต่ที่ระบบอาร์เอฟไอดีใช้คือ 13.56 MHz แท็กที่ใช้ก็มักเป็นแท็กชนิดพาสซีฟเช่นกัน และแน่นอนอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านต่ำ แต่ก็เป็นที่นิยมในการใช้งานระบบอาร์เอฟไอดีในย่านความถี่สูงโดยเฉพาะในโรงพยาบาล ซึ่งไม่ต้องติดต่อกับอุปกรณ์ที่มีอยู่แล้วในระบบ และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก

ย่านความถี่ถัดไปจากนี้คือ ย่านความถี่สูงมาก (VHF) มีความถี่ในช่วง 30 – 300 MHz แต่ไม่มีระบบอาร์เอฟไอดีใดใช้งานในย่านความถี่นี้เลย

- **แท็กย่านความถี่สูงยิ่ง (UHF)** 400 - 1000 เมกะเฮิรตซ์ ย่านความถี่สูงยิ่งคือมีค่า 300 MHz – 1 GHz ระบบ อาร์เอฟไอดี ที่ใช้แท็กชนิดพาสซีฟในสหรัฐอเมริกาจะใช้ความถี่ 915 MHz ส่วนในยุโรปจะใช้ความถี่ 868 MHz แต่โดยทั่วไปจะใช้ในย่านความถี่ 315 – 433 MHz มีการใช้งานทั้งแท็กชนิดพาสซีฟและแอ็กทีฟ มีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านสูง ปัจจุบันเริ่มมีการใช้งานย่านความถี่นี้กันมากขึ้น แต่ย่านความถี่นี้ไม่เป็นที่ยอมรับทั่วโลก
- **แท็กย่านไมโครเวฟ** 2.4 กิกะเฮิรตซ์ย่านความถี่สูงยิ่งคือมีค่าตั้งแต่ 1 GHz ขึ้นไปซึ่งเป็นคลื่นไมโครเวฟ ระบบอาร์เอฟไอดีใช้งานที่ความถี่ 2.45 GHz หรือ 5.8 GHz แท็กที่ใช้ได้ก็มีทั้งแท็กชนิดกึ่งพาสซีฟและพาสซีฟ มีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านสูงมาก แต่เนื่องจากความยาวคลื่นเป็นส่วนกลับของความถี่ แท็กชนิดพาสซีฟจะทำงานในช่วงย่านความถี่ไมโครเวฟที่ต่ำที่สุด อีกทั้งยังมีขนาดเล็กมาก เพราะไมโครชิปที่อยู่ข้างในแท็กผลิตมาเล็กมาก ความถี่ที่ใช้งานมากสุดคือ 2.4 GHz ทั้งด้านอุตสาหกรรม (Industry) วิทยาศาสตร์ (Scientific) และทางการแพทย์ (Medical) รวมเรียกว่า ISM และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก




**รูปที่ 2.4** แสดงคลื่นความถี่ที่ใช้ในระบบ RFID




**รูปที่ 2.5** แสดงตัวอย่างการใช้งานคลื่นความถี่ในย่านต่างๆ

**ตารางที่ 2.1** แสดงย่านความถี่ต่างๆของระบบ RFID และตัวอย่างการใช้งาน

| ย่านความถี่ | | ระยะทาง | การใช้งาน |
|---|---|---|---|
| ย่านความถี่ต่ำ (Low Frequency :LF) | 125-134 kHz | 18 นิ้ว (~10 cm.) | ปศุสัตว์, ป้ายสินค้ากันขโมยที่อ่านในระยะใกล้, ระบบกันขโมยรถยนต์ |
| ย่านความถี่สูง (High Frequency :HF) | 13.553-13.567 MHz | 3 ฟุต (~1 m.) อ่านได้เร็ว 10-100 Tags/sec. | ห้องสมุด, สมาร์ทการ์ด, ระบบติดตามหนังสือ, ระบบปิดเปิดประตู |
| ย่านความถี่สูงยิ่ง (Ultra High Frequency :UF) | 400 – 1000 MHz (for US 433 MHz) | 10-30 ฟุต (Passive: 2-5 m.) (Active: 1-100 m.) อ่านได้เร็วมาก 100-1000 Tags/sec. | ตู้สิ่งของ, รถบรรทุก, แท่นยกสินค้า (Pallet) |
| ย่านความถี่ไมโครเวฟ (Microwave Frequency) | 2.4 GHz, 5.8 GHz | >30 ฟุต (Passive: < 1 m.) (Active: 1-15 m.) | อุปกรณ์ไร้สาย |

**2.2.1.2 ตัวอ่านข้อมูล (Reader / Interrogator)**

**1. องค์ประกอบของเครื่องอ่าน**

- **RF Interface Chip** ทำหน้าที่แปลงสัญญาณอนาลอคให้เป็นดิจิตอลในกรณีส่งข้อมูลจาก Coil ไปยัง MCU และแปลงสัญญาณดิจิตอลเป็นอนาลอค ในกรณีส่งข้อมูลจาก MCU ไปยัง Coil
- **Microcontroller** ทำหน้าที่ควบคุมการทำงานของเครื่องอ่าน เช่น ถอดรหัสสัญญาณตามรูปแบบที่เข้ารหัสมา เช่น Manchester, NRZ, Bi-phase เป็นต้น
- **Input / Output** ทำหน้าที่เป็นตัวรับข้อมูลจากภายนอก เช่น Keypad เป็นต้น และเป็นอุปกรณ์แสดงผลหรือควบคุมอุปกรณ์ที่อยู่ภายนอก เช่น LED, 7-Segment, LCD, Relay เป็นต้น
- **Graphic User Interface** ทำหน้าที่แสดงผลของข้อมูลให้อยู่ในรูปแบบที่ผู้ใช้งานสามารถเข้าใจได้ง่าย อีกทั้งยังใช้ควบคุมการทำงานของเครื่องอ่านได้อีกด้วย

**2. ประเภทของเครื่องอ่าน**

เครื่องอ่านแต่ละประเภทอาจมีชื่อเรียกแตกต่างกันไปตามลักษณะการใช้งาน หรือผู้ผลิตทางการค้าแต่ละราย แต่หลักๆ แบ่งได้ 2 ประเภท คือ

- **Stationary Reader** เครื่องอ่านแบบติดตั้ง หรือ Fixed Reader มักติดตั้งไว้ที่ผนัง กำแพง ประตูทางเข้า หรือที่ที่เหมาะสมกับระยะการอ่าน ซึ่งไม่จำเป็นต้องติดตั้งบนสิ่งที่เคลื่อนที่ไม่ได้เท่านั้น อาจจะติดตั้งบนผนังรถบรรทุกสิ่งของก็ได้ เป็นต้น โดยทั่วไปต้องอาศัยสายอากาศจากภายนอกในการอ่านข้อมูลจากแท็ก เครื่องอ่านชนิดนี้ไม่ได้ทนทานเหมือนแท็ก โดยเฉพาะอย่างหากใช้งานนอกสถานที่ต้องได้รับการดูแลรักษาเป็นอย่างดี ราคาก็ค่อนข้างถูกกว่าแบบพกพา จึงเป็นที่นิยมใช้ในปัจจุบัน โดยทั่วไปเครื่องอ่านชนิดนี้มี 2 โหมดการทำงานคือ
    - Autonomous Mode เครื่องอ่านจะอ่านแท็กอย่างต่อเนื่องในระยะการอ่านได้ของเครื่องอ่าน และบันทึกข้อมูลลงแท็กลิสต์ (tag list) ซึ่งมีเวลากำหนดว่าอยู่ในแท็กลิสต์นานแค่ไหน เรียกว่า persist time ซึ่งเครื่องควบคุมในระบบ หรือ host ที่จะอ่านข้อมูลในแท็กลิสต์สามารถอ่านได้เป็นระยะในช่วงเวลา persist time นี้เท่านั้น ไม่เช่นนั้นเครื่องอ่านจะลบข้อมูลนั้นทิ้งจากแท็กลิสต์

- Interactive Mode เครื่องอ่านจะถูกควบคุมโดยโปรแกรมที่รันบนระบบ หรือ host ที่ติดต่อกับเครื่องอ่านได้ ส่งคำสั่งให้เครื่องอ่านทำงาน ซึ่งเครื่องอ่านก็สามารถทำงานได้หลายคำสั่ง และส่งข้อมูลในแท็กลิสต์ให้ตัวควบคุมที่สั่งให้ทำงานมาได้

**รูปที่ 2.6** แสดงตัวอย่างเครื่องอ่านแบบติดตั้งยี่ห้อหนึ่ง

- **Handheld / Portable Reader** เครื่องอ่านแบบพกพา ก็ตามชื่อพกพาได้สะดวก เพราะโดยทั่วไปมีสายอากาศติดตั้งอยู่ภายในตัวเครื่อง ราคาแพงกว่าแบบติดตั้ง แต่ปัจจุบันก็เริ่มมีผู้ใช้มากขึ้น เพราะมีผู้ผลิตหลายรายแข่งขันกัน ทำให้ราคาถูกลง

**รูปที่ 2.7** แสดงตัวอย่างเครื่องอ่านแบบพกพายี่ห้อหนึ่ง

หน้าที่สำคัญของตัวอ่านข้อมูล (Reader หรือ Interrogator) ก็คือการรับข้อมูลที่ส่งมาจากแท็ก แล้วทำการตรวจสอบความผิดพลาดของข้อมูล ถอดรหัสสัญญาณข้อมูลที่ได้รับ ซึ่งกระทำโดย ไมโครคอนโทรเลอร์ อัลกอริทึมที่อยู่ในเฟิร์มแวร์ (Firmware) ของตัวไมโครคอนโทรเลอร์จะทำหน้าที่ในการส่งสัญญาณ ถอดรหัสสัญญาณที่ได้ และทำหน้าที่ติดต่อกับคอมพิวเตอร์เพื่อนำข้อมูลผ่านเข้าสู่กระบวนการต่อไป

นอกจากนี้ตัวอ่านข้อมูลที่ดีต้องมีความสามารถในการป้องกันการอ่านข้อมูลซ้ำ เช่น เมื่อมีแท็กหลายๆ อันเข้ามาอยู่ใกล้เครื่องอ่าน เมื่อแท็กมีพลังงานเพียงพอ แท็กแต่ละอันจะพยายามส่งข้อมูลของตัวเองมาที่เครื่องอ่านพร้อม ๆ กัน ทำให้เครื่องอ่านไม่สามารถแยกแยะข้อมูลที่ส่งมาได้ ซึ่งเราเรียกปรากฏการณ์นี้ว่า การชนกันของข้อมูล (Collision) จะไม่มีข้อมูลส่งถึงเครื่องอ่านเลย วิธีการแก้ไขโดยการทำการเพิ่มฟังก์ชั่นป้องกันการชนกันบนแท็กและเครื่องอ่าน (Anti-collision) ซึ่งจะมี หลายเทคนิค เช่นจัดคิวการอ่านแท็กโดยทำเป็นช่วงเวลาสั้นๆ เมื่อแท็กถูกอ่านแล้วจะไม่มีการอ่านซ้ำอีก

#### 2.2.1.3 คอนโทรลเลอร์(Controller / Host)

ระบบประยุกต์ใช้งาน รวมถึงระบบฮาร์ดแวร์และซอร์ฟแวรประยุกต์ใช้งาน หรือระบบฐานข้อมูล มักเป็นเครื่องคอมพิวเตอร์ที่ควบคุมการทำงานของระบบที่ใช้ RFID อาจจะเรียกว่า Middleware ก็ได้ ส่วนนี้ขึ้นกับระบบการใช้งานของระบบที่เกี่ยวข้อง

### 2.2.2 การทำงานของระบบอาร์เอฟไอดี

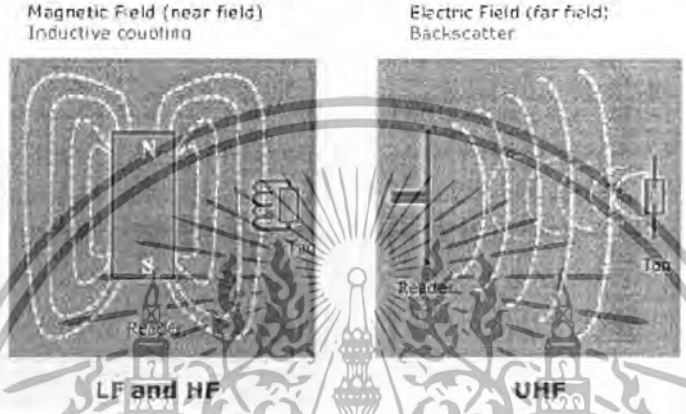
อาร์เอฟไอดีเป็นระบบที่นำเอาคลื่นวิทยุมาเป็นคลื่นพาหะเพื่อใช้ในการสื่อสารข้อมูลระหว่างอุปกรณ์สองชนิดที่เรียกว่า แท็กและตัวอ่านข้อมูลซึ่งเป็นการสื่อสารแบบไร้สายโดยต้องมีการเข้ารหัสข้อมูลดิจิตอลที่จะส่งผ่านให้อยู่ในรูปที่เหมาะสมสำหรับการส่งผ่านช่องสัญญาณซึ่งอาจมีสัญญาณรบกวนอีก จึงต้องเลือกวิธีการเข้ารหัสให้ข้อมูลมีค่าความผิดพลาดน้อยที่สุดเท่าที่เป็นไปได้ ซึ่งมีหลายวิธี เช่น NRZ, Manchester, Miller, Differential, DBP เป็นต้น แล้วต้องนำข้อมูลที่ต้องการส่งที่ผ่านการเข้ารหัสแล้วมาทำการมอดูเลต (modulation) กับคลื่นพาหะ เพื่อปรับเปลี่ยนค่าต่างๆ ของคลื่นพาหะให้ส่งออกไปได้เหมาะสมแล้วส่งออกผ่านทางสายอากาศที่อยู่ในตัวรับข้อมูล การมอดูเลตก็มีหลายวิธีอีก เช่น ASK (Amplitude Shift Keying), PSK (Phase Shift Keying), FSK (Frequency Shift Keying) เป็นต้น

**หลักการทำงานเบื้องต้นของระบบอาร์เอฟไอดี**

1. ตัวอ่านข้อมูลจะปล่อยคลื่นแม่เหล็กไฟฟ้าออกมาตลอดเวลา และคอยตรวจจับว่ามีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้าหรือไม่ หรืออีกนัยหนึ่งก็คือการคอยตรวจจับว่ามีการมอดูเลตสัญญาณเกิดขึ้นหรือไม่

การส่งคลื่นมี 2 วิธี คือ

- ในย่านความถี่ต่ำและสูง (LF และ HF) จะใช้วิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling หรือ Proximity Electromagnetic)
- ในระบบความถี่สูงยิ่งและไมโครเวฟ (UHF และ Microwave) จะใช้วิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Propagation Coupling)




**รูปที่ 2.8** แสดงการปล่อยคลื่นแม่เหล็กไฟฟ้าวิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling) และวิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Propagation Coupling)

2. เมื่อมีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้า แท็กจะได้รับพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำของคลื่นแม่เหล็กไฟฟ้าเพื่อให้แท็กเริ่มทำงาน และจะส่งข้อมูลในหน่วยความจำที่ผ่านการมอดูเลตกับคลื่นพาหะแล้วออกมาทางสายอากาศที่อยู่ภายในแท็ก

ถ้าเป็นแท็กย่านความถี่สูงยิ่ง (UHF) ก็จะส่งไปยังเครื่องอ่านแบบ backscattering

ถ้าเป็นแท็กชนิดแอ็กทีฟจะส่งสัญญาณแบบ Broadcast ออกมา แต่จะทำการส่งข้อมูลออกให้เครื่องอ่าน ก็ต่อเมื่อได้รับสัญญาณจากเครื่องอ่านและแบบเครื่องบอกตำแหน่ง หรือเบคอน (beacon) ซึ่งสัญญาณจะถูกปล่อยออกมาเป็นระยะ ๆ ตลอดเวลา

3. คลื่นพาหะที่ถูกส่งออกมาจากแท็กจะเกิดการเปลี่ยนแปลงแอมปลิจูด, ความถี่ หรือเฟส ขึ้นอยู่กับวิธีการมอดูเลต

4. ตัวอ่านข้อมูลจะตรวจจับความเปลี่ยนแปลงของคลื่นพาหะแปลงออกมาเป็นข้อมูล แล้วทำการถอดรหัสเพื่อนำข้อมูลไปใช้งานต่อไป

### 2.2.3 ข้อดีข้อเสียของอาร์เอฟไอดี

**ข้อดีของอาร์เอฟไอดี**

- ใช้บ่งชี้ลักษณะแสดงตัวเฉพาะได้(Individual Uniqueness)
- สามารถอ่านข้อมูลได้อัตโนมัติในรัศมีการรับส่งคลื่นของเครื่องอ่านโดยไม่ต้องเห็นแท็ก ไม่ต้องจัดแท็กให้หน้าตรงกับเครื่องอ่าน (Non-Line of Sight)
- สามารถอ่านข้อมูลได้ทีละหลายๆแท็ก(Simultaneous reading)
- สามารถอ่านข้อมูลได้รวดเร็ว และแม่นยำ ขจัดปัญหาการอ่านข้อมูลซ้ำ
- สามารถอ่านข้อมูลได้ระยะไกล ขึ้นกับแท็กและคลื่นที่ใช้แต่ละชนิด
- สามารถอ่านข้อมูลขณะวัตถุกำลังเคลื่อนไหวได้
- สามารถทำทั้งอ่านและเขียนข้อมูลลงแท็กได้ตลอดเวลา โดยไม่ต้องสัมผัส (Contactless)
- ความละเอียดและความจุของข้อมูลสูง
- ข้อมูลมีความปลอดภัยสูง ยากต่อการปลอมแปลงหรือลอกเลียนแบบ
- นำแท็กกลับมาใช้ใหม่ได้(Reusable)
- ทนทานต่อความเปียกชื้น แรงสั่นสะเทือน การกระทบกระแทก
- ลดความเสี่ยงต่อชีวิตคนในพื้นที่ที่อันตราย

**ข้อเสียของอาร์เอฟไอดี**

- ราคาลงทุนค่อนข้างสูง
- อาจถูกรบกวนจากคลื่นความถี่ต่ำที่ใช้งานได้ง่าย ต้องได้รับการควบคุมดูแลจากหน่วยงานที่เกี่ยวข้อง
- ข้อจำกัดกับผลิตภัณฑ์และสิ่งแวดล้อมที่รบกวนคลื่นแม่เหล็กไฟฟ้า เช่น น้ำ, โลหะ เป็นต้น
- ปัญหาด้านการละเมิดสิทธิส่วนบุคคลหากนำอาร์เอฟไอดีไปใช้อย่างไม่ระมัดระวัง
- ปัญหาด้านความปลอดภัยของข้อมูลยังมีอยู่หากผู้โจรกรรมมีความสามารถหรือผู้ใช้งานไม่มีการจัดการเรื่องความปลอดภัยไว้

83030

**การเปรียบเทียบกับรหัสแท่ง (Barcode)**

เนื่องจากอาร์เอฟไอดีเป็นเทคโนโลยีระบบบ่งชี้อัตโนมัติ (Auto-ID) ชนิดหนึ่ง ที่เข้ามามีบทบาทสำคัญพอๆกับเทคโนโลยีรหัสแท่ง (Barcode) ที่เป็นที่รู้จักและใช้งานกันมานานอย่างแพร่หลาย จึงมักเป็นข้อเปรียบเทียบกันระหว่างสองเทคโนโลยีนี้

**ตารางที่ 2.2** แสดงข้อเปรียบเทียบลักษณะของเทคโนโลยีอาร์เอฟไอดีและ Barcode

| **อาร์เอฟไอดี** | **Barcode** |
|---|---|
| อ่านได้อัตโนมัติไม่ต้องเห็นแท็ก (Non-Line of Sight) | ต้องอ่านแนวตรงกับเครื่องอ่าน ห้ามมีอะไรบังแท็ก |
| อ่านได้ทีละหลายอัน | อ่านได้ทีละอัน |
| อ่านได้ระยะไกล | ต้องแนบกับเครื่องอ่าน |
| อ่านได้รวดเร็ว | อ่านช้ากว่า |
| อ่านและเขียนข้อมูลได้ | อ่านได้อย่างเดียว |
| ความจุของข้อมูลสูง | ความจุของข้อมูลต่ำ |
| ความถูกต้องของข้อมูลสูงกว่า(99.5%) | ความถูกต้องของข้อมูลต่ำกว่า(80%) |
| ทนทาน | เสียหายง่าย |
| ความปลอดภัยของข้อมูลสูง มนุษย์อ่านแท็กโดยตรงไม่ได้ | ไม่ได้ป้องกันความปลอดภัยใดๆ แท็กอ่านออกได้โดยมนุษย์ |
| นำกลับมาใช้ใหม่ได้ | นำกลับมาใช้ใหม่ไม่ได้ |
| นำมาใช้กับบางวัตถุไม่ได้ เช่น น้ำ, โลหะ เป็นต้น | ไม่มีข้อจำกัดในเรื่องนี้ |
| ราคาแพง | ราคาถูก |

**ตารางที่ 2.3** แสดงเงื่อนไขที่มีส่วนตัดสินใจว่าอาร์เอฟไอดีจะมาแทนหรือมาเสริม Barcode

| เงื่อนไข / ข้อกำหนด | อาร์เอฟไอดี | Barcode |
|---|---|---|
| ต้นทุนในการลงทุนเริ่มแรกต่ำ | | ✓ |
| อุณหภูมิ สูง/ต่ำ | ✓ | |
| ของเหลว | | ✓ |
| ข้อจำกัดการมองเห็นไม่ได้ | ✓ | |
| ความคงทน (ขึ้นกับลักษณะการใช้งาน) | ✓ | ✓ |
| ความยืดหยุ่น (ขึ้นกับลักษณะการใช้งาน) | ✓ | ✓ |
| ความน่าเชื่อถือ / ความถูกต้อง | ✓ | ✓ |
| การเชื่อมโยงการปฏิบัติงานกับผู้ค้าหลายราย | | ✓ |
| ความสามารถในการสแกนสิ่งของหลายตัว | ✓ | |
| การนำกลับมาใช้ใหม่ | ✓ | |

### 2.2.4 มาตราฐานของอาร์เอฟไอดี

มาตรฐานของอาร์เอฟไอดีสามารถแบ่งออกเป็น 2 ส่วน ได้แก่

- **คลื่นความถี่ (RF Spectrum)** คลื่นความถี่ที่จะใช้สำหรับอุปกรณ์อาร์เอฟไอดีนี้ ค่อนข้างยากที่จะกำหนดให้เป็นมาตรฐานสากล เนื่องจากแต่ละประเทศเป็นเจ้าของและเป็นผู้ควบคุมการใช้คลื่นความถี่เอง ดังนั้น จึงขึ้นอยู่แต่ละประเทศที่จะเป็นผู้พิจารณาจัดสรรคลื่นความถี่สำหรับการใช้งานประเภทต่างๆ
- **มาตรฐานอาร์เอฟไอดี (RFID Standards)** บริษัทผู้ผลิตต่างๆ ได้พัฒนาและผลิตระบบอาร์เอฟไอดี ออกมาโดยมีมาตรฐานที่ใช้ในการส่งถ่ายข้อมูลระหว่างเครื่องอ่านกับแท็กที่แตกต่างกันออกไป ทั้งนี้ ขึ้นกับความต้องการของตลาดเป็นสำคัญ

อย่างไรก็ตาม หลายองค์กรได้ตระหนักถึงปัญหาเรื่องมาตรฐานนี้ และได้เริ่มมีการพัฒนาระบบมาตรฐานขึ้นมาทั้งในยุโรปและอเมริกา ตัวอย่างเช่น

- **หน่วยงาน ANSI's X3T6** ซึ่งประกอบไปด้วยผู้ผลิตและผู้ใช้งานระบบอาร์เอฟไอดีในอเมริกา ได้กำลังทำการพัฒนามาตรฐานของระบบอาร์เอฟไอดีที่ความถี่ 2.45 GHz ขึ้นมา

- **The International Organization for Standards : ISO** ก็ได้มีมาตรฐานเกี่ยวกับการใช้ระบบ RFID ดังนี้

  ISO 11784 และ 11785 กับงานปศุสัตว์

  ISO/IEC 15961 : Data Protocol

  ISO/IEC 19562 : Data encoding rules and logical memory function

  ISO/IEC 15963 : Unique Identification of RF Tag

  ISO/IEC 18000 : RFID Air Interface Standard

- **The Electronic Product Code (EPC Global)** เป็นองค์กรที่เกิดจากความร่วมมือระหว่าง the Uniform Code Council (UCC) และ EAN International โดยได้รับการสนับสนุนจากภาคอุตสาหกรรม ทำหน้าที่ในการกำหนดและพัฒนามาตรฐานรหัสของสิ่งของอิเลคทรอนิกส์และโครงข่าย มาตรฐาน EPC นี้เป็นมาตรฐานเปิด (Open standard) พัฒนาขึ้นโดย the Auto-ID Center ซึ่งได้รับทุนจากบริษัทขนาดใหญ่หลายบริษัทได้แก่ บริษัท Coca-Cola, Intel, Wal-Mart และ Philips Semiconductors เป็นต้น โดยมีวัตถุประสงค์เพื่อที่จะทำให้ราคาของอุปกรณ์ฮาร์ดแวร์ลดต่ำลงจนอยู่ในระดับที่สามารถนำ RFID ไปติดไว้กับสิ่งของรายชิ้นได้

ซึ่ง 2 หน่วยงานหลังนี้ (ISO และ EPC Global) เป็นหน่วยงานหลักที่กำหนดมาตรฐาน RFID มี 4 ด้านดังนี้

- มาตรฐานด้านเทคโนโลยี (Technology)
- มาตรฐานรูปแบบของข้อมูล (Data format)
- มาตรฐานวิธีการทดสอบ (Conformance)
- มาตรฐานการใช้งาน (Applications)

เนื่องจาก EPC Global ไม่ได้กำหนดเฉพาะมาตรฐานอาร์เอฟไอดี และเป็นที่นิยมใช้ในปัจจุบัน จึงขอยกตัวอย่างที่ EPC Global กำหนดมาตรฐานของแท็กอาร์เอฟไอดีแบ่งเป็นคลาส 5 คลาส ดังนี้

คลาส 0 /1 (Class 0/Class I) ใช้กับแท็กชนิดพาสซีฟ แบบอ่านได้อย่างเดียว

คลาส 2 (Class II) ใช้กับแท็กชนิดพาสซีฟแต่มี ฟังก์ชันเพิ่มเติมขึ้นมา

คลาส 3 (Class III) ใช้กับแท็กชนิดกึ่งพาสซีฟ

คลาส 4 (Class IV) ใช้กับแท็กชนิดแอ็กทีฟ ที่ติดต่อสื่อสารกันได้ทั้งแบบบอร์ดแบนด์ (board-band) และแบบเพียร์ทูเพียร์ (peer-to-peer)

คลาส 5 (Class V) เครื่องอ่านที่ใช้มาตรฐานนี้สามารถส่งไฟเลี้ยง (power) ให้แท็กในคลาส1, 2 และ 3 ได้ ส่วนแท็กในคลาส 4 และ 5 ก็สามารถติดต่อสื่อสารกันได้

EPC -96 Structure แบ่งโครงสร้างข้อมูลของแท็กคลาสต่างๆ ดังนี้

| Element | Header | EPC Manager | Object Class | Serial Number |
|---|---|---|---|---|
| Bits | 8 | 34 | 20 | 34 |
| Values[1] | 0-256 | 0-17,179,869,183 | 0-1,048,575 | 0-17,179,869,183 (inc 0-9) |

**รูปที่ 2.9** แสดงโครงสร้างของแท็กตามมาตรฐาน EPC-96

คลาส 0 มี 64 บิต

คลาส 1 มี 96 บิต

คลาส 1 Gen2 มี 128 หรือ 256 บิต

คลาส 2 มี 96 บิตเหมือนคลาส 1 แต่มีหน่วยความจำใหญ่ขึ้น และสามารถอ่านและเขียนแท็กได้

คลาส 3 มี 96 บิตเหมือนคลาส 2 แต่เป็นแท็กชนิดกึ่งพาสซีฟ

คลาส 4 มี 96 บิตเหมือนคลาส 2 แต่เป็นแท็กชนิดแอ็กทีฟ

ISO ก็เป็นองค์กรใหญ่ที่สำคัญเช่นกัน จึงมักมีการเปรียบเทียบมาตรฐานกันกับ EPC Global

**ตารางที่ 2.4** แสดงการเปรียบเทียบมาตรฐานระหว่าง ISO/IEC กับ EPC

| | ISO/IEC | EPC |
|---|---|---|
| เทคโนโลยี | ISO/IEC 18000 – RF-IC for<br>Item Management<br>Part2 -< 135 kHz<br>Part3 – 13.56 MHz<br>Part4 – 2450 MHz<br>Part6 – 860 – 960 MHz<br>Part7 – 433.92 MHz (active) | Class I-V (13.56 and UHF only)<br>Class 0/Class I: read-only passive tags<br>Class II tags : passive tags with additional<br>functionality Class III tags: semi–passive RFID<br>tags<br>Class IV tags: active tags<br>With broad-band peer-to-peer<br>Communication<br>Class V tags : Readers Can<br>Power other Class I, II and<br>III tags: Communication with<br>Classes IV and V |
| รูปแบบของข้อมูล | ISO/IEC 15418 – Application<br>Identifiers & Data Identifiers<br>ISO/IEC 15434 – Syntax<br>ISO/IEC 15459 – Transport<br>License Plate<br>ISO/IEC 15961 – Data Protocol:<br>Application Interface<br>ISO/IEC 15962 – Data Protocol:<br>Data Encoding Rules and<br>Logical Memory Functions | EPC<br>Class 0 – 64 bits<br>Class 1 – 96 bits<br>Class 1 G2 – 128/256 bits<br>Class 2 – Class 1 with larger<br>Memory and read/write<br>Class 3 – Class 2 with<br>Sensors (semi-passive)<br>Class 4 – passive tags |

## 2.3 ไอรอนไพธอน (IronPython)

IronPython เป็นภาษาที่สามารถพัฒนาโปรแกรมด้วยภาษา Python นำมาประมวลบนแพลตฟอร์ม (Platform) .NET และ Mono ได้ โดย Jim Hugunin เป็นคนสร้างภาษานี้ขึ้นมา เวอร์ชัน 1.0 (Version 1.0) ประกาศออกมาให้ใช้กันเมื่อ 5 กันยายน 2549

ภาษา IronPython ตั้งแต่เวอร์ชัน 0.6 อยู่ภายใต้มาตรฐาน Common Public License เมื่อเดือนสิงหาคม ปี 2547 ก็มีการรวมตัวกันของหลายๆ โครงการพัฒนาภาษา IronPython เนื่องจากเริ่มแรกภาษา IronPython ถูกสร้างให้เป็นส่วนหนึ่งของ Microsoft's Shared Source ผู้พัฒนาก็เลยอ้างว่าไม่

น่าจะเป็น Open source อีกต่อไปเพราะไม่ได้เริ่มสร้างจากองค์กร OSI: Open Source Initiative ก็เลย มีเวอร์ชัน 2.0 แอลฟา (2.0 alpha) ออกมาภายใต้ Microsoft Permissive License

ภาษา IronPython เขียนขึ้นมาด้วยภาษา C# เกือบทั้งหมด มีบางส่วนเป็นภาษา Python ที่เกิดอัตโนมัติจาก code generator ในภาษา Python

### 2.3.1 เส้นทางการพัฒนาของภาษา IronPython

IronPython เวอร์ชั่นล่าสุดในปัจจุบัน คือเวอร์ชัน 1.1 ซึ่งเปรียบเทียบได้กับการใช้งาน CPython 2.4.4 แต่ถึงอย่างไรก็มีข้อแตกต่างกันอยู่บ้างในการเขียนภาษา Python กับ IronPython

ส่วนเวอร์ชัน 2.0 แอลฟา (ที่ประกาศออกมาในปัจจุบัน คือ 2.0 Alpha Release 4) จะเทียบได้กับ CPython 2.5 IronPython 2.0 จะอยู่ที่ส่วนบนของ Dynamic Language Runtime: DLR ที่ประกอบด้วย Dynamic Type System และ Dynamic Language Hosting Environment ที่ถ่ายทอดมาจาก IronPython 1

DLR ถูกสร้างขึ้นมาเพื่อให้สามารถเขียนโปรแกรมภาษาพวกที่เป็นไดนามิกส์ (Dynamic Language) บน CLR: Common Language Runtime ได้สะดวก ซึ่งพวกภาษาแบบไดนามิกส์ที่กำลังพัฒนาให้รันบน DLR ได้นั้นก็มีอีกหลายภาษาเช่น IronRuby, Jscript, VBx

DLR รันอยู่ส่วนบนสุดของแก่น CLR เลยที่ครอบคลุม Microsoft Silverlight อยู่ด้วย ดังนั้น IronPython ก็เขียน Script ของ Silverlight ผ่านบราวเซอร์ browser ที่ฝั่งไคล์แอนต์ (Client-side) ได้

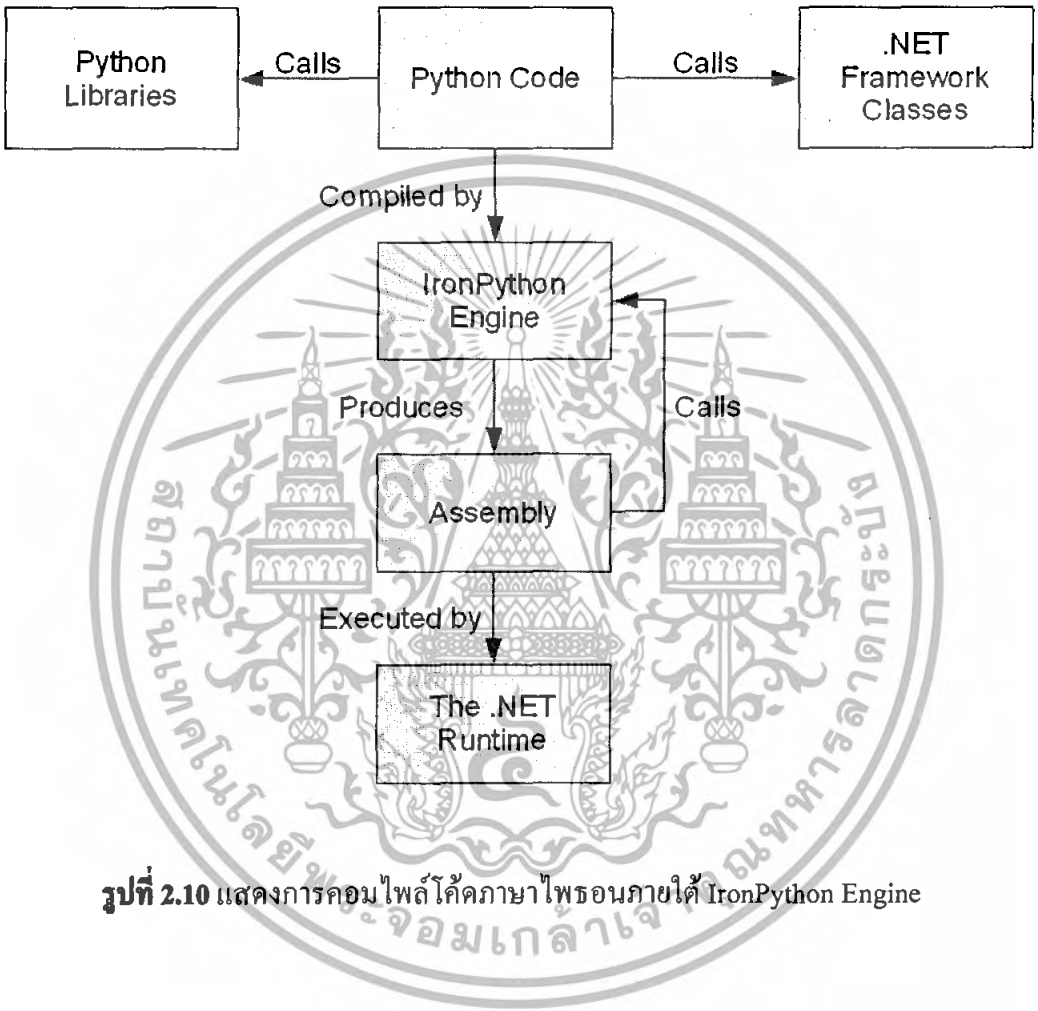
### 2.3.2 ขอบเขตการใช้งาน

ข้อดีของ IronPython ที่สำคัญอย่างหนึ่งคือ มีฟังก์ชันครอบคุมการใช้งาน .NET framework เพื่อพัฒนาโปรแกรมบน .NET ได้เหมือนพวกภาษาของ .NET เอง วิธีการง่ายๆ คือ การนำตัวแปลภาษา IronPython (IronPython Interpreter) ไปรวมกับ .NET framework ที่มีอยู่แล้ว ซึ่งนักพัฒนาก็ลองเขียน script ในภาษา IronPython ติดต่อกับออบเจกต์ (Object) ใน .NET framework แล้วก็ใช้การได้ นอกจากนี้ยังเพิ่มฟังก์ชั่นอื่นเพิ่มเติมเข้าไปใน .NET framework ได้เลยโดยไม่ต้องเปลี่ยนแปลงโค้ดโครงสร้างหลักของเก่า

**ตารางที่ 2.5** แสดงคลาสในดอทเน็ตเฟรมเวิร์ก (Common .NET Assemblies and Namespaces)

| **Assembly Name** | **Purpose** |
|---|---|
| System | Contains the base .NET types, exceptions, garbage collection classes and much more. |
| System.Data | Classes for working with databases, both high and low level. |
| System.Drawing | Provides access to the GDI+ graphics system. |
| System.Management | Provides access to Windows management information and events (WMI), useful for system administration tasks. |
| System.Environment | Allows you to access and manipulate the current environment, like command line arguments and environment variables. |
| System.Diagnostics | Interact with processes. |
| System.XML | For processing XML, including SOAP, XSL/T and more. |
| System.Web | The ASP.NET web development framework.<br>System.IO Contains classes for working with paths, files and directories. Includes classes to read and write to file systems or data-streams, synchronously or asynchronously.<br>Microsoft.Win32 Classes that wrap Win32 common dialogs and components including the registry. |
| System.Threading | Classes needed for multithreaded application development. |
| System.Text | Classes for working with strings (like StringBuilder) and the Encoding classes which can convert text to and from bytes. |
| System.Windows.Forms | Provides a rich user interface for applications. |
| System.Windows | The base namespace for WPF, the new GUI framework that is part of .NET 3.0. |
| System.ServiceModel | Contains classes, enumerations, and interfaces to build Windows Communication Foundation (WCF) service and client applications. |

IronPython ทำให้การใช้งานออบเจกต์ต่างๆ ของ .NET ได้มีประสิทธิภาพมากขึ้น เพราะเมื่ออ้างอิงถึงออบเจกต์ใดออบเจกต์หนึ่งใน .NET IronPython จะมีการใช้กลไก Reflection คือทำการอิมพอร์ท (import) ทั้ง type และ method ทั้งหมดของออบเจกต์นั้น ทำให้เราสามารถเพิ่มหรือเปลี่ยนแปลงออบเจกต์นั้นๆในรูปแบบของภาษา IronPython เองได้




**รูปที่ 2.10** แสดงการคอมไพล์โค้ดภาษาไพธอนภายใต้ IronPython Engine

### 2.3.3 วิธีการใช้งาน

สำหรับภาษาไอรอนไพธอนได้มีทางเลือกให้ผู้พัฒนาโปรแกรมได้หลายทาง เช่น ใช้ไอรอนไพธอนร่วมกับ Visual Studio 2005 ซึ่งต้องลง SDK เวอร์ชั่นล่าสุดหรือเวอร์ชั่น 4.0 ขึ้นไป อีกวิธีคือใช้อิดิเตอร์ (Editor) ของไพธอนทั่วไปที่มีอยู่แล้ว นำโค้ดมารันผ่านคอมไพเลอร์ของไอรอน-ไพธอน (IronPython Engine) เพียงแค่ดาวน์โหลด (Download) มาแล้วเรียกใช้ตอนที่จะคอมไพล์ผ่านคอมมานด์ไลน์ (Command line)

ในที่นี้เราใช้วิธีหลัง เนื่องจากมีความคุ้นเคย (เพราะ SDK ได้พัฒนาออกมาทีหลัง) แต่ก็ยังได้ใช้ประโยชน์จากวิธีแรกอยู่ กล่าวคือ เราสามารถใช้สิ่งที่ง่าย และสะดวกจาก SDK ได้เช่น การออกแบบแอพพลิเคชั่น สามารถใช้ส่วนประกอบ (Component) จากแถบเครื่องมือ (Tool box) ออกแบบได้ง่าย ไม่ได้เขียนโค้ดเองทั้งหมด แต่เนื่องจากบางคุณลักษณะที่เราต้องการไม่สามารถทำได้ด้วยวิธีตามดอทเน็ตที่คุ้นเคยธรรมดา (หรืออาจเป็นเพราะยังไม่รู้วิธีการใช้ที่ถูกต้อง) จึงใช้วิธีการแบบผสม เพื่อความสะดวก รวดเร็ว และง่ายต่อการพัฒนา

**ตารางที่ 2.6** แสดงการเขียนโค้ด Hello World ง่ายๆ เปรียบเทียบระหว่าง C# และ IronPython

| **A Small HelloWorld App in C#** | **Equivalent in Python** |
|---|---|
| using System;<br>class Hello<br>{<br>string msg;<br>public Hello()<br>{<br>this.msg = "Hello World";<br>}<br>public Hello(string msg)<br>{<br>this.msg = msg;<br>} | class Hello(object):<br>def __init__(self,<br>msg='hello world'):<br>self.msg = msg<br>def sayHello(self):<br>print self.msg<br>app = Hello ()<br>app.sayHello() |

ข้อควรระวังในการใช้ Visual Studio 2005 ที่รวมไอรอนไพธอนอยู่ด้วยนั้น ต้องระวังเรื่องชนิดของตัวแปร (Data types) การเขียนด้วยไพธอน เราไม่ต้องประกาศชนิดตัวแปรไว้ ในไอรอนไพธอนก็เช่นเดียวกัน แต่หากมีการเรียกใช้ตัวแปรบางชนิดของดอทเน็ต ต้องระมัดระวังการใช้อาจไม่เหมือนที่คุ้นเคย กล่าวคือ เราต้องรู้ชนิดของตัวแปรนั้นอย่างแท้จริง แล้วเรียกใช้แบบชื่อเต็ม (หากไม่ประกาศให้ใช้แค่ชื่อได้) เช่น ชนิดของตัวแปรนั้นเป็น null ที่รู้จักกันทั่วไป ในไพธอนจะใช้ none

หรือกรณี null ในฐานข้อมูล ดอทเน็ตยังคงใช้ null เหมือนเดิมได้ แต่ไอรอนไพธอน ต้องใช้ System.DBnull

โดยปกติดอทเน็ตจะสร้างโค้ดของแต่ละส่วนประกอบให้อัตโนมัติ แต่มีบางส่วนที่ใช้ไม่ได้ เช่น การใช้ DataGridView ในดอทเน็ตจะมีการสร้างโค้ด DataGridViewName.BeginInit() กับ DataGridViewName.EndInit() ซึ่งส่งผลให้คอมล์โค้ดไม่ได้ แต่ตอนออกแบบต้องให้มี statements นี้อยู่จึงจะไม่เกิด Error แต่ตอนรันเราก็ต้องเอา statements นี้ออก

## 2.4 การเชื่อมต่อตัวอ่านอย่างง่าย

Basic Reader Interface (BRI) พัฒนาขึ้นมาเพื่อควบคุมการทำงานของ RFID Reader โดยสร้างให้มี overhead ของโปรโตคอลน้อยที่สุด BRI เป็น emmbeded softwareในตัว RFID Reader ซึ่งตัวsoftwareนี้เป็น อินเตอร์เฟสที่เชื่อมต่อระหว่าง โฮสต์กับ RFID Manager ที่รันอยู่บน RFID Reader จุดมุ่งหมายของทำการเชื่อมต่อทำให้อุปกรณ์ที่ไม่มีความสามารถในการติดต่อ เช่นอุปกรณ์ประเภท Programmable Logic Controllers (PLC) สามารติดต่อและควบคุม RFID Reader ได้ด้วยคำสั่งง่ายๆผ่านทางอินเตอร์เฟส

BRI ยังเป็นอินเตอร์เฟสที่เชื่อมต่อไปยัง RFID Manager Core software ซึ่งเป็นส่วนที่ทำหน้าที่ในการควบคุมการทำงานของ RFID Radio Interface BRI มีสามารถปรับโหมดการทำงานได้อย่างหลากหลายและมีประสิทธิภาพสูง

### 2.4.1 คุณสมบัติของ BRI

- รองรับการทำงานผ่านคำสั่งแบบ ASCII
- มีรูปแบบของชุดคำสั่งที่เข้าใจได้ง่าย
- มาตรฐานการใช้ประกาศค่าตัวแปรสตริงและตัวเลขเหมือนในโปรแกรมมิ่งภาษา C เช่นการใช้ single quotes แทน ASCII 1 character และ double quotes แทนหลาย ASCII character
- รองรับการเชื่อมต่อหลากหลายรูปแบบ เช่น serial, Ethernet, USB และอื่นๆ
- รองรับ general purpose input and output (GPIO) interface ในการกำหนดค่าพารามิเตอร์ต่างๆในการควบคุมการทำงานของ RFID Reader โดยตัว RFID Reader ต้องรองรับการทำงานของ GPIO ด้วย

- มีชุดคำสั่งที่หลากหลายซึ่งสามารถนำไปใช้กับ RFID Reader แต่ละชนิดที่มีความแตกต่างกันได้
- สามารถสร้าง macro เพื่อกำหนดค่าพารามิเตอร์และcommand ต่างๆลงใน RFID Reader ได้เองเพื่อเพิ่มความสามารถในการทำงานของ RFID Reader

**2.4.2 รูปแบบการเรียกใช้ BRI**

โดยปกติการเรียกใช้ BRI มักจะอยู่ในรูปของการติดต่อผ่านทางอินเตอร์เฟสระหว่างคอมพิวเตอร์ โดยการเรียกใช้ BRI เพื่อติดต่อไปยัง RFID Reader สามารถทำได้ 2 รูปแบบคือ

- ติดต่อผ่านทาง serial RS-232
- ติดต่อผ่านทาง TCP connection โดยผ่านทาง port 2189

**2.4.3 การพัฒนาโปรแกรมโดยใช้ BRI**

ในการพัฒนาแอพพลิเคชั่นโดยใช้ BRI เป็นโปรโตคอลติดต่อไปยัง RFID reader สามารถทำการติดต่อได้สองรูปแบบคือ

1. พัฒนาโฮสต์แอพพลิเคชั่นให้เป็นตัวจัดการปรับแต่งค่าพารามิเตอร์ต่างๆของ RFID เองโดยมีข้อดีคือ

- Extensibility สามารถให้โฮสต์แอพพลิเคชั่นทำการปรับแต่งให้รองรับการอัพเกรดระบบ
- Scalability โฮสต์แอพพลิเคชั่นสามารถเลือกอุปกรณ์ที่จะติดต่อและโดยการระบุชื่อหรือ address ของอุปกรณ์
- Maintainability ง่ายต่อการซ่อมบำรุงฮาร์ดแวร์

2. ใช้ RFID Resource Kit ซึ่งเป็น API ที่พัฒนาขึ้นโดยบริษัท Intermec ซึ่งสามารถใช้งานร่วมกับผลิตภัณฑ์ของบริษัท Intermec ได้ทั้งหมด สามารถ download ได้ที่ www.intermec.com เลือกที่ Service &Support > Developer Library > Resource Kits. ในการพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีจะใช้ RFID Resource kit ในการพัฒนา

ในการใช้งาน RFID Resource Kit จะประกอบด้วยโมเดลในการเขียนโปรแกรมอยู่สองโมเดลคือ

- Basic Programming model เป็นอินเดอร์เฟสระดับ low-level ในการติดต่อไปยัง BRI reader ทำงานในลักษณะของ command/response ใช้งานโดยอาศัย syntax ของ BRI ซึ่งเข้าใจได้ยากกว่า โมเดลแบบ Advance การใช้งานโมเดลแบบ Basic มีคลาสในการทำงานหลักคือ BasicBRIReader มีพื้นฐานในการทำงานคือ การส่ง BRI commands ไปยัง RFID Reader และรอรับการ responses and asynchronous event จาก RFID Reader
- Advanced Programming model รูปแบบของการเขียนโปรแกรมติดต่ออาร์เอฟไอดีขั้นสูง ก็มีการเตรียมฟังก์ชันที่ดีกว่าให้ใช้งาน เพื่อจัดการกับแท็กข้อมูลที่มากกว่าโมเดลแบบ Basic ซึ่งสามารถตั้งค่าได้ทั้งที่เครื่องอ่านแท็กและการจัดการต่างๆเกี่ยวกับแท็กได้มีประสิทธิภาพ ช่วยให้ผู้เขียนโปรแกรมใช้งานได้ง่ายขึ้น ไม่ต้องยุ่งกับรูปแบบการเขียน BRI มากนัก ก็สามารถเอาข้อมูลที่อยู่ในแท็กออกมาได้โดยง่าย

ปกติเราสามารถเขียนแอพพลิเคชั่นโดยใช้แค่ BRI ซึ่งจะใช้รูปแบบการเขียนที่เข้าใจง่ายและใช้ง่ายด้วย มีสองรูปแบบใหญ่ๆ คือ การกรองแท็ก (tag-filter statements) เพื่อแยกกลุ่มหรือประเภทของแท็ก และ โครงสร้างของแท็ก (schema statements) เพื่อกำหนดจำนวน หรือตำแหน่งของข้อมูลในแท็ก (data fields) ที่จะอ่านหรือเขียน

การใช้รูปแบบนี้ จะใช้คลาส BRIReader ซึ่งจะแบ่ง BRI ทำงานเป็นขั้นตอน (Method) คือ ส่วนอ่านแท็ก (Read) และ ตรวจจับแท็ก (PollTags) แล้วแบ่งเป็นเหตุการณ์ (BRI tag events) แล้วห่อหุ้ม (encapsulate) ข้อมูลที่เกี่ยวข้องกับแต่ละแท็กที่เครื่องอ่านตรวจจับได้ คือแท็กคีย์ (Tag Key) และข้อมูลที่เรากำหนด เก็บไว้เป็นอาเรย์ (Array) ที่อยู่ในแท็ก

ในการพัฒนาระบบจะใช้โมเดลแบบ Advanced ในการติดต่อไปยัง RFID Reader คลาสในการทำงานหลักคือ BRIReader โดยมี constructor ดังนี้

```
public BRIReader(Control Owner);
public BRIReader(Control Owner, string aReaderURI);
public BRIReader(Control Owner, string aReaderURI, BRIReader.LoggerOptionsAdv LogOpts);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize);
public BRIReader(Control Owner, string aReaderURI, string aFieldSchema);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize, BRIReader.LoggerOptionsAdv LogOpts);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize, int aEventQueueSize);
public BRIReader(Control Owner, string aReaderURI, string aFieldSchema, BRIReader.LoggerOptionsAdv LogOpts);
```

ในการพัฒนาเลือกใช้ constructor ในรูปแบบที่ 3 โดยค่าพารามิเตอร์คือ

1. Control Owner คือ control ตัวที่จะทำการสร้าง BRIReader โดยปกติตัวที่เรียกจะเป็น System.Windows.Forms
2. String aReaderURI เป็นสตริงที่บอกถึงการเชื่อมต่อและที่อยู่ของ RFID Reader ถ้าเป็นการเชื่อมต่อผ่านทาง TCP สตริงจะมีค่าเป็น "TCP://nnn.nnn.nnn.nnn" โดย n แทนค่าไอพีแอดเดรสของ RFID Reader และถ้าเชื่อมต่อผ่านทาง serial RS-232 สตริงจะมีค่าเป็น "SERIAL://COM1"
3. BRIReader.LoggerOptionAdv LogOpts เป็นการเก็บ log ในการติดต่อ RFID Reader

เมื่อทำการสร้างคลาส BRIReader แล้วจะสามารถเรียกใช้ฟังก์ชันการทำงานต่างๆของ RFID ได้ผ่านทางคลาสที่สร้างไป โดยฟังก์ชันหลักที่ใช้งานคือ ReadTagsAsStringData() ซึ่งจะทำการอ่านแท็กอาร์เอฟไอดีแล้วเก็บรายละเอียดของแท็กไว้ใน BRIReader.Tags เพื่อนำไปใช้งานในการระบุถึงสินค้าแต่ละชิ้นได้

เนื่องจากในระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีต้องมีการใช้งานร่วมกับระบบอื่นๆที่ทำการพัฒนาไว้แล้วในห้องวิจัยการสื่อสารและคมนาคมชาญฉลาด (ICT Lab) ซึ่งพัฒนาไว้โดยใช้ภาษาไพธอน แต่ในการติดต่ออุปกรณ์อาร์เอฟไอดีต้องติดต่อผ่าน BRI ซึ่งพัฒนาโดยใช้ภาษา C# จึงทำการพัฒนาโปรแกรมในการติดต่ออาร์เอฟไอดีโดยเขียนเป็นอินเตอร์เฟสโดยใช้ภาษาไอรอนไพธอนซึ่งมีรายละเอียดดังนี้

```
import clr
clr.AddReference('System')
clr.AddReference('System.Windows.Forms')
clr.AddReference("Intermec.DataCollection.RFID.BasicBRI.dll")
clr.AddReference("Intermec.DataCollection.RFID.AdvancedBRI.dll")
import System
from System.Windows.Forms import *
from Intermec.DataCollection.RFID import *

class InterfaceRFID(Form):
    def __init__(self):
        #"""RFID Parameters"""
        self.m_Reader = BRIReader
        self.logOpts = BRIReader.LoggerOptionsAdv()
        self.logOpts.IDEConsoleEnable = False
        self.logOpts.LogEnable = True
        self.logOpts.LogFilePath = "C:\MySampleLog1.txt"
        self.logOpts.TimeStampEnable = True
        self.logOpts.ShowNonPrintableChars = True
        self.logOpts.MaxFileSize = 400000
        self.m_fConnected = False
        self.sCurrentTagType = "GEN2"
        self.fIP4Connected = True
```

ฟังก์ชันหลักของการทำงานในการติดต่อ RFID Reader คือ ฟังก์ชัน OpenRFIDReader() จะทำการติดต่อไปยัง RFID Reader โดยจะเก็บ log เป็นไว้ที่ C:\MySampleLog1.txtและเก็บค่าตัวแปร self.m_fConnected เป็นตัวแปร boolean เพื่อบอกว่าติดต่อ RFID Reader สำเร็จหรือไม่

```
def OpenRFIDReader(self,sURIAddress):
        self.fSuccess = True
        try:
            self.m_Reader = BRIReader(self,sURIAddress,self.logOpts)
        except:
            self.m_Reader = None
            self.fSuccess = False
        if((self.m_Reader != None) and (self.m_Reader.IsConnected)):
            self.m_fConnected = self.fSuccess
        else:
            self.m_fConnected = False
        return self.m_fConnected

def ReadTagsAsStringData(self):
        self.Data = ""
        self.fSuccess = False
        self.sSchema = None
        self.sFilter = None
        if (self.m_Reader == None):
            return False
        try:
            self.fNoUpdate = False
            while True:
                self.fTBUpdated = False
                if (self.sSchema == None):
                    self.m_Reader.DefaultFieldSchema = None
                    self.fSuccess = self.m_Reader.Read()
                else:
                    self.fSuccess =
self.m_Reader.Read(self.sFilter,self.sSchema)
                if (self.m_Reader.TagCount == 0):
                    self.Data = "No Tags detected"
                    return False
                for i in range(0,self.m_Reader.TagCount):
                    self.Data += self.m_Reader.Tags[i].ToString()
                    if i != self.m_Reader.TagCount -1 :
                        self.Data += ","
                return self.Data
                break
        except:
            self.fSuccess = False
        return self.fSuccess
```

ส่วนฟังก์ชัน ReadTagsAsString() จะทำการอ่านแท็กและเก็บค่าเป็นสตริงในตัวแปร self.Data โดยเก็บรหัสของแท็กอาร์เอฟไอดีที่อ่านได้และคั่นด้วยเครื่องหมายจุลภาค

# บทที่ 3
# การออกแบบระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

## 3.1 ความสามารถของระบบ

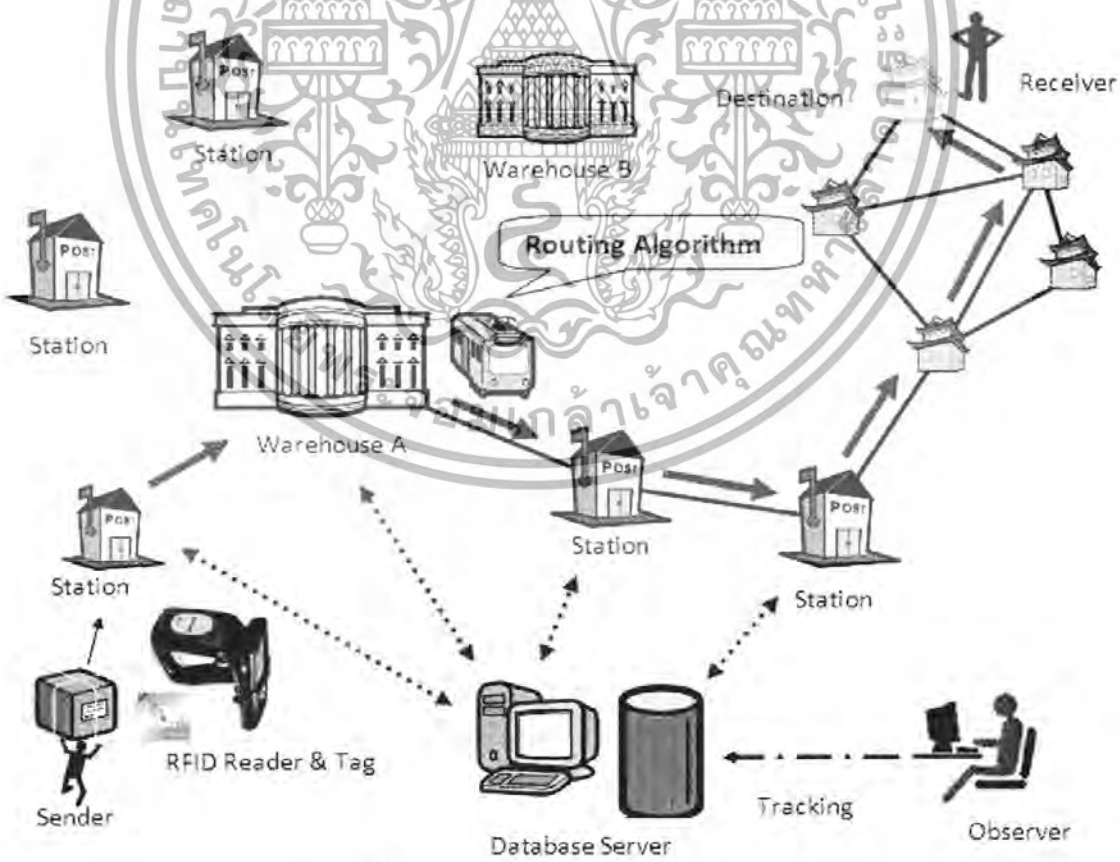
ระบบการขนส่งที่ออกแบบขึ้นมีความสามารถดังต่อไปนี้

1. ระบบขนส่งสิ่งของมีระบบฐานข้อมูลที่สามารถบันทึกกิจกรรมการขนส่ง
2. สามารถติดตามการขนส่งตั้งแต่สิ่งของเข้ามาในระบบจากผู้ส่งจนถึงปลายทางผู้รับได้
3. สามารถนำข้อมูลต่างๆ มาวิเคราะห์เพื่อการวางแผนการส่งสิ่งของให้มีประสิทธิภาพ

## 3.2 การออกแบบ

### 3.2.1 การออกแบบระบบโลจิสติกส์

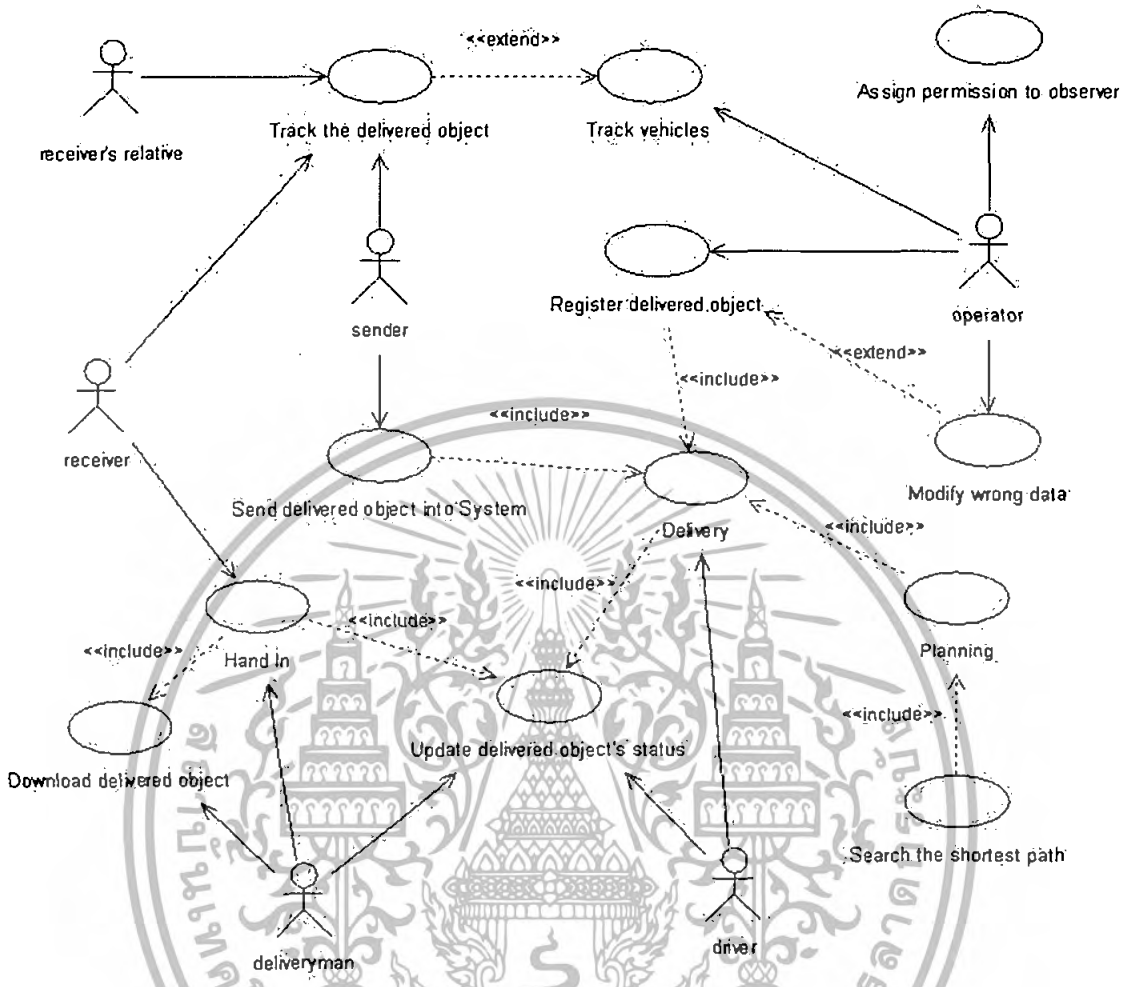
ในการพัฒนาระบบโลจิสติกส์สามารถแบ่งได้เป็นสองส่วนคือ คือ ส่วนคลังจัดเก็บสิ่งของและส่วนการขนส่งสิ่งของ ซึ่งในที่นี้จะพัฒนาระบบโดยเน้นไปในส่วนการขนส่งสิ่งของเท่านั้น




**รูปที่ 3.1** แสดงการทำงานของระบบ

ในระบบที่ออกแบบขึ้นนี้ จะควบคุมตั้งแต่เมื่อสิ่งของเข้าสู่ระบบ โดยประเภทของการขนส่งสิ่งของมี 2 ประเภท คือ แบบธรรมดา และ แบบด่วนพิเศษ โดยสิ่งของทั้งหมดที่เข้ามาในระบบแล้วจะต้องติดแท็กอาร์เอฟไอดีทุกชิ้น เพื่อใช้ประโยชน์จากเทคโนโลยี Auto-ID ซึ่งข้อมูลรายละเอียดเกี่ยวกับสิ่งของที่จำเป็นในการขนส่งจะถูกบันทึกลงฐานข้อมูลทั้งหมด ในที่นี้จะถือว่า สิ่งของเริ่มส่งจากสถานีศูนย์กลาง ดูแลจัดการกระจายการขนส่งไปที่สถานีย่อย (ไม่ได้ดูแลเรื่องการขนสิ่งของกลับเข้าศูนย์กลาง) โดยสถานีย่อยจะแบ่งเขตการรับผิดชอบการขนส่งไปถึงที่อยู่ปลายทางของผู้รับ ศูนย์กลางจัดการเรื่องเส้นทางการขนส่งโดยใช้อัลกอริทึมเอสตาร์ (A* Algorithm) เพื่อหาเส้นทางที่สั้นที่สุด เมื่อรถขนส่งมาส่งสิ่งของที่สถานีย่อยใดๆ จะทำการอัพเดทข้อมูลลงฐานข้อมูลด้วย จากสถานีไปส่งที่ที่อยู่ของผู้รับก็ใช้วิธีการหาเส้นทางเช่นแบบเดียวกัน เมื่อส่งสิ่งของถึงมือผู้รับแล้ว ผู้รับต้องรับรองการรับสิ่งของนั้นด้วย ทั้งนี้ระหว่างการขนส่งผู้ที่เกี่ยวข้องกับสิ่งของนั้นๆ จะมีไอดีที่สามารถใช้ในการติดตามสิ่งของได้ผ่านเว็บแอพพลิเคชั่น แล้วผู้ขนส่งต้องอัพเดทข้อมูลการรับสิ่งของนั้นด้วย จึงจะถือว่าสิ้นสุดการขนส่งสิ่งของนั้นๆ

ในส่วนของการออกแบบ มองภาพรวมของระบบขนส่งสิ่งของในชีวิตจริงแล้วเขียน Use Case Diagram ได้ดังนี้




**รูปที่ 3.2** แสดง Use Case Diagram ส่วนควบคุมจัดการระบบขนส่ง

ในระบบมีผู้เกี่ยวข้อง (Actor) หลัก 6 ส่วนคือ

1 พนักงาน (Operator) มีหน้าที่รับสิ่งของเข้าระบบแล้วระบุให้สิ่งของมีเอกลักษณ์ไม่ซ้ำกัน กรอกรายละเอียดเกี่ยวกับสิ่งของที่เกี่ยวข้องทั้งหมดลงระบบฐานข้อมูล สามารถปรับเปลี่ยนข้อมูลเกี่ยวกับสิ่งของที่ไม่ถูกต้องก่อนเริ่มการขนส่งได้ สามารถติดตามการขนส่งสิ่งของที่เกี่ยวข้องกับสถานีนั้นๆได้ตลอดเส้นทางทั้งหมด มีสิทธิ์ในการอนุญาตสิทธิ์การติดตามสถานะการขนส่งสิ่งของให้แก่ผู้ที่เกี่ยวข้องทุกฝ่าย นอกจากนี้พนักงานยังสามารถนำข้อมูลที่บันทึกไว้มาวิเคราะห์หรือใช้งานอย่างอื่นได้อีกด้วย เพื่อพัฒนาปรับปรุงประสิทธิภาพของระบบต่อไป

2 ผู้ส่ง (Sender) คือบุคคลหรือหน่วยงานที่มีความต้องการส่งสิ่งของใดๆ เข้ามาในระบบ เพื่อใช้บริการของระบบในการส่งของให้ถึงมือผู้รับ สามารถตรวจสอบสถานะการนำส่งได้ตลอดเวลาที่ของยังอยู่ในระบบ

3 ผู้รับ (Receiver) คือบุคคลหรือหน่วยงานที่เป็นผู้มีสิทธิ์รับสิ่งของที่ผู้ส่งต้องการส่งให้ หากผู้รับได้รับสิ่งของถูกต้องแม่นยำหรือไม่ ก็มีผลต่อประสิทธิภาพของระบบขนส่ง

4 ผู้ขนส่ง (Deliveryman) มีหน้าที่ส่งสิ่งของจนถึงมือผู้รับในระบบ รายงานผลการขนส่งจนถึงมือผู้รับ

5 ผู้เกี่ยวข้องกับผู้รับ (Receiver's relative) คือบุคคลหรือหน่วยงานที่มีความเกี่ยวข้องกับผู้รับสิ่งของ สามารถตรวจสอบและติดตามสถานะการส่งของสิ่งของได้

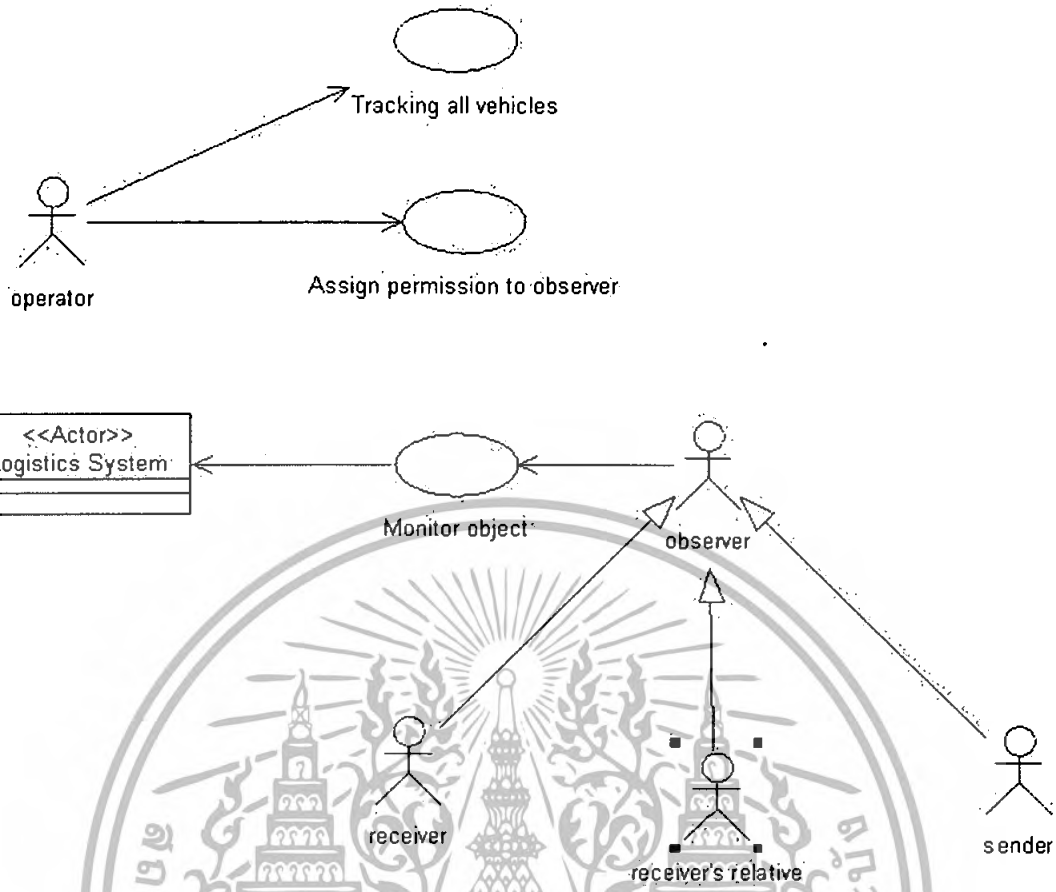
6 พนักงานขับรถ (Driver) ทำหน้าที่เป็นคนดำเนินการตามที่ระบบคำนวณเส้นทางในการขนส่งจากการใช้อัลกอริทึมเอสตาร์ในการแก้ปัญหา เมื่อส่งสิ่งของเข้าที่สถานีย่อยใดๆ ผู้ขับรถขนส่งต้องอัพเดทข้อมูลลงฐานข้อมูลทุกครั้ง

ส่วนควบคุมจัดการระบบขนส่งเริ่มตั้งแต่มีผู้ส่งต้องการส่งสิ่งของถึงผู้รับ นำสิ่งของนั้นมาเจ้าหน้าที่ก็เป็นคนดำเนินการติดแท็กอาร์เอฟไอดี แล้วกรอกรายละเอียดที่จำเป็นในการขนส่ง เช่น ชื่อ ที่อยู่ของผู้รับ น้ำหนักและประเภทของการขนส่ง เพื่อคำนวณค่าบริการการขนส่ง แล้วเมื่อสิ้นสุดการรับข้อมูล เจ้าหน้าที่ต้องให้หมายเลขที่ใช้ในการติดตามการขนส่งสิ่งของแก่ผู้ส่ง ซึ่งผู้ส่งจะแจ้งหมายเลขให้แก่ผู้รับ หรือผู้ที่เกี่ยวข้องทราบอย่างไรนั้น ขึ้นกับผู้ส่งเอง ผู้ที่ทราบหมายเลขนี้สามารถดูสถานะการขนส่งได้ตั้งแต่ต้นจนจบผ่านเว็บแอพพลิเคชั่น

เจ้าหน้าที่สามารถแก้ไขข้อมูลที่พบว่าผิดพลาดได้ หากสิ่งของนั้นยังอยู่ในสถานีที่รับผิดชอบ นอกจากนี้เจ้าหน้าที่ยังมีสิทธิ์ในการกำหนดสิทธิ์การติดตามการขนส่งแก่บุคคลที่เกี่ยวข้องได้ และยังสามารถติดตามการขนส่งสิ่งของที่เข้าสถานีที่รับผิดชอบได้ทั้งหมด เจ้าหน้าที่ที่ศูนย์กลางสามารถติดตามรถขนส่งได้ด้วย

ผู้ขับรถขนส่งต้องทราบเส้นทางการขนส่งจากการใช้อัลกอริทึมเอสตาร์ในการแก้ปัญหาเมื่อส่งสิ่งของเข้าที่สถานีย่อยใดๆ ผู้ขับรถขนส่งต้องอัพเดทข้อมูลลงฐานข้อมูลทุกครั้ง

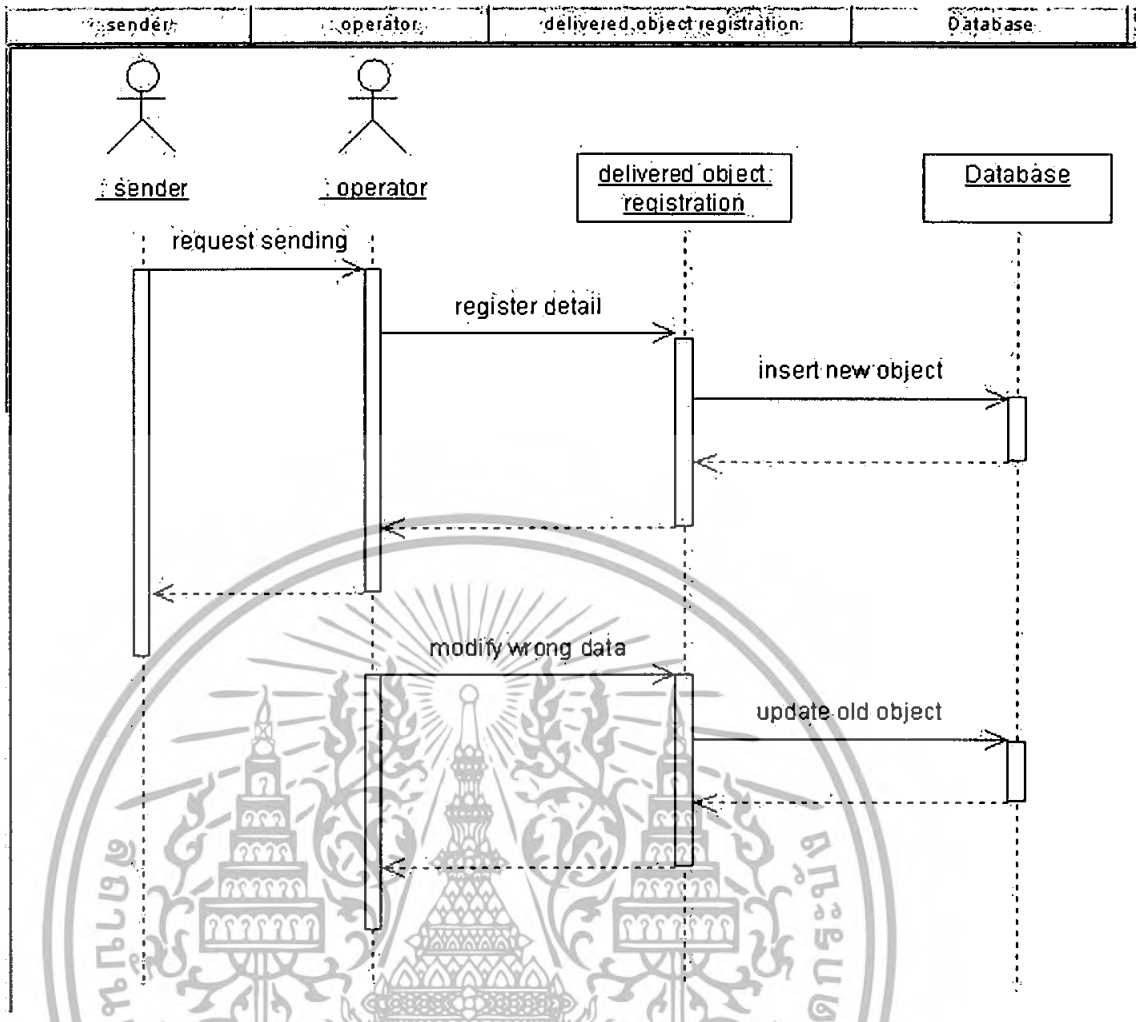
เมื่อผู้ขนส่งจะส่งสิ่งของจากสถานีย่อยไปถึงมือผู้รับก็ต้องใช้อัลกอริทึมเอสตาร์ในการหาเส้นทางสั้นที่สุด นำสิ่งของส่งถึงมือผู้รับ แล้วรายงานผลการรับสิ่งของ อัพเดทลงฐานข้อมูล




**รูปที่ 3.3** แสดง Use Case Diagram ของการตรวจสอบและติดตามสิ่งของ

การตรวจสอบและติดตามสิ่งของนั้น เจ้าหน้าที่สามารถกำหนดสิทธิ์ในการติดตามสิ่งของให้แก่ผู้ที่ต้องการตรวจสอบทั้งหมดได้ ไม่ว่าจะเป็นผู้ส่ง ผู้รับ ผู้ขนส่ง ผู้ขับรถขนส่ง หรือผู้ที่เกี่ยวข้องกับสิ่งของนั้นๆ อีกทั้งยังสามารถตรวจสอบสถานะของรถขนส่งได้ด้วย

ผู้ที่ต้องการตรวจสอบสิ่งของต้องทราบหมายเลข หรือมีสิทธิ์การเข้าถึงข้อมูลจากเจ้าหน้าที่ ดูสถานะการขนส่งที่ดูแลโดยระบบ




**รูปที่ 3.4** แสดง Sequence diagram ของ Register delivered object

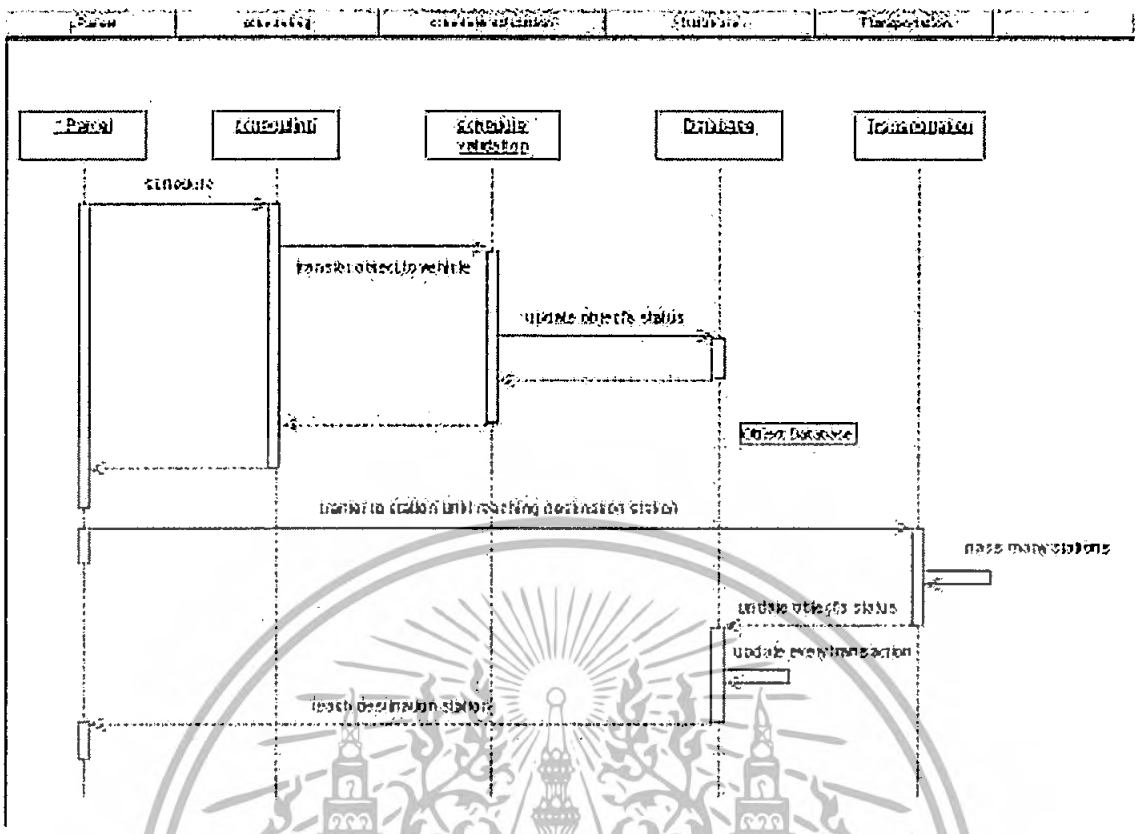
1 เมื่อผู้ส่งต้องการส่งสิ่งของ ก็บอกรายละเอียดแก่พนักงาน

2 พนักงานกรอกรายละเอียดทั้งหมดที่ต้องการใช้ในระบบขนส่ง พร้อมทั้งติดแท็ก RFID ให้เรียบร้อย

3 เพิ่มข้อมูลลงไปในฐานข้อมูลพร้อมทั้งให้หมายเลขในการติดตามสิ่งของแก่ผู้ส่งด้วย

4 หากมีการพบว่าข้อมูลผิดพลาดก่อนเริ่มการขนส่ง พนักงานสามารถแก้ไขข้อมูลได้

5 เมื่อมีการแก้ไขข้อมูล ต้องอัพเดทฐานข้อมูลทุกครั้ง



รูปที่ 3.5 แสดง Sequence diagram ของ Delivery

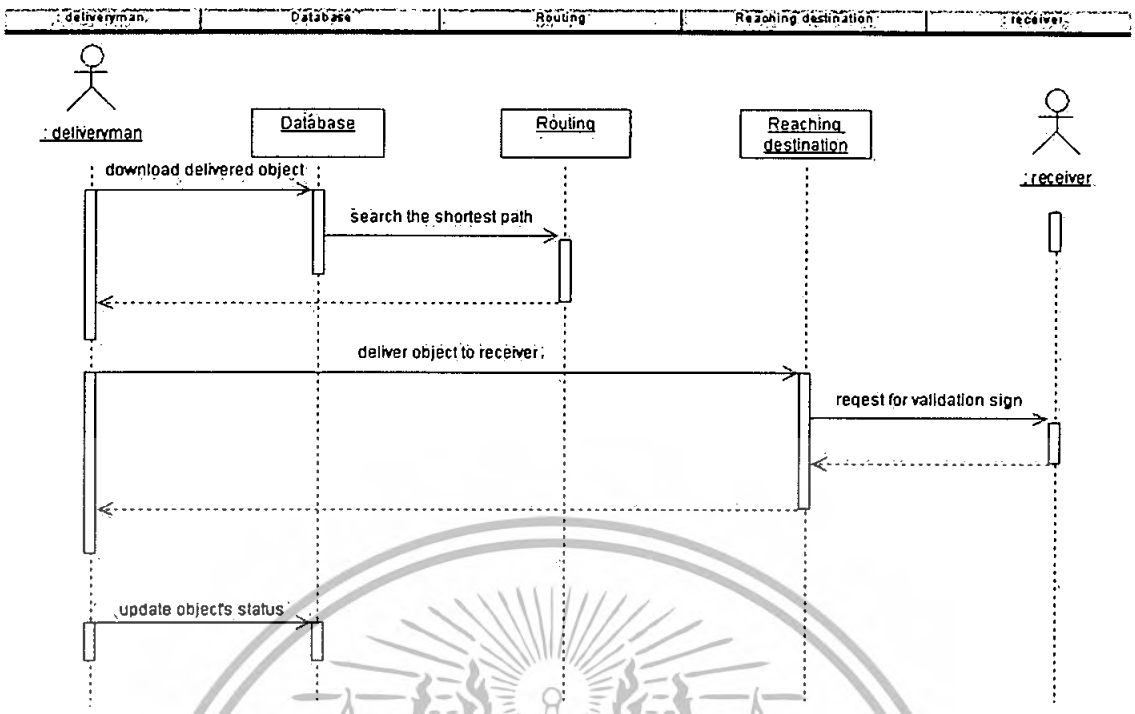
1 เมื่อจะขนส่งสิ่งของต้องจัดการเรียงลำดับการขนส่งให้เหมาะสม พร้อมทั้งจัดการว่าจะขนโดยรถคันไหน

2 ขนของขึ้นรถตามที่จัดการไว้

3 ตรวจสอบความถูกต้องตามที่จัดไว้ แล้วอัพเดทฐานข้อมูลด้วย

4 ขับรถขนส่งไปตามเส้นทางที่รับผิดชอบ ขนส่งสิ่งของให้สถานีที่รับผิดชอบให้ถูกต้อง

5 อัพเดทฐานข้อมูลทุกครั้งที่สิ่งของถึงสถานีปลายทางที่รับผิดชอบ



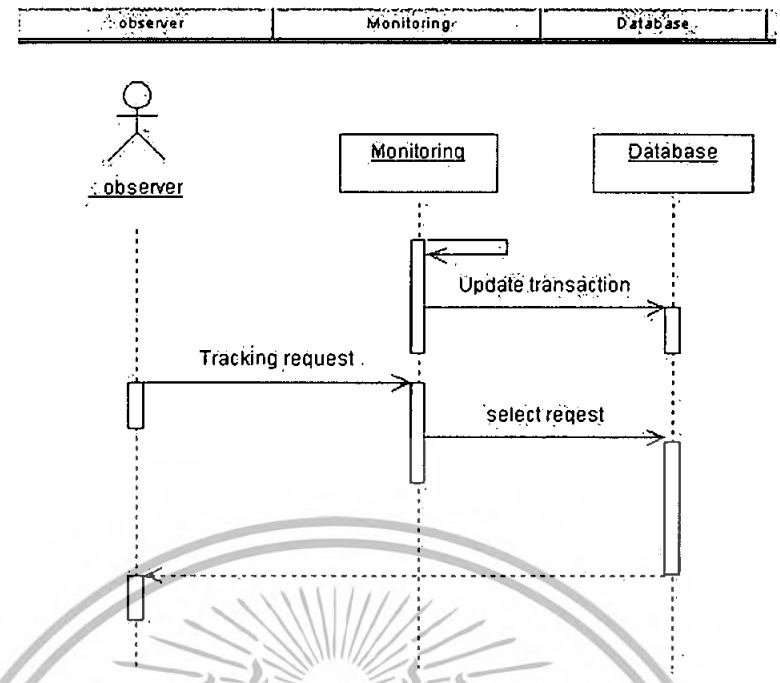

รูปที่ 3.6 แสดง Sequence diagram ของ Hand In

1 เมื่อผู้ขนส่งจะดูรายการที่ต้องขนส่งที่อยู่ในสถานีที่รับผิดชอบ ก็เลือกรายการที่ต้องขนส่ง

2 ทำการหาเส้นทางที่สั้นที่สุดในการขนส่งให้ผู้ขนส่ง

3 ผู้ขนส่งนำสิ่งของไปส่งจนถึงมือผู้รับตามที่กำหนด แล้วให้ผู้รับเซ็นรับสิ่งของให้เรียบร้อย

4 ผู้ขนส่งต้องมารายงานผลการขนส่ง อัพเดทในฐานข้อมูลทั้งหมด




รูปที่ 3.7 แสดง Sequence diagram ของ Track the delivered object

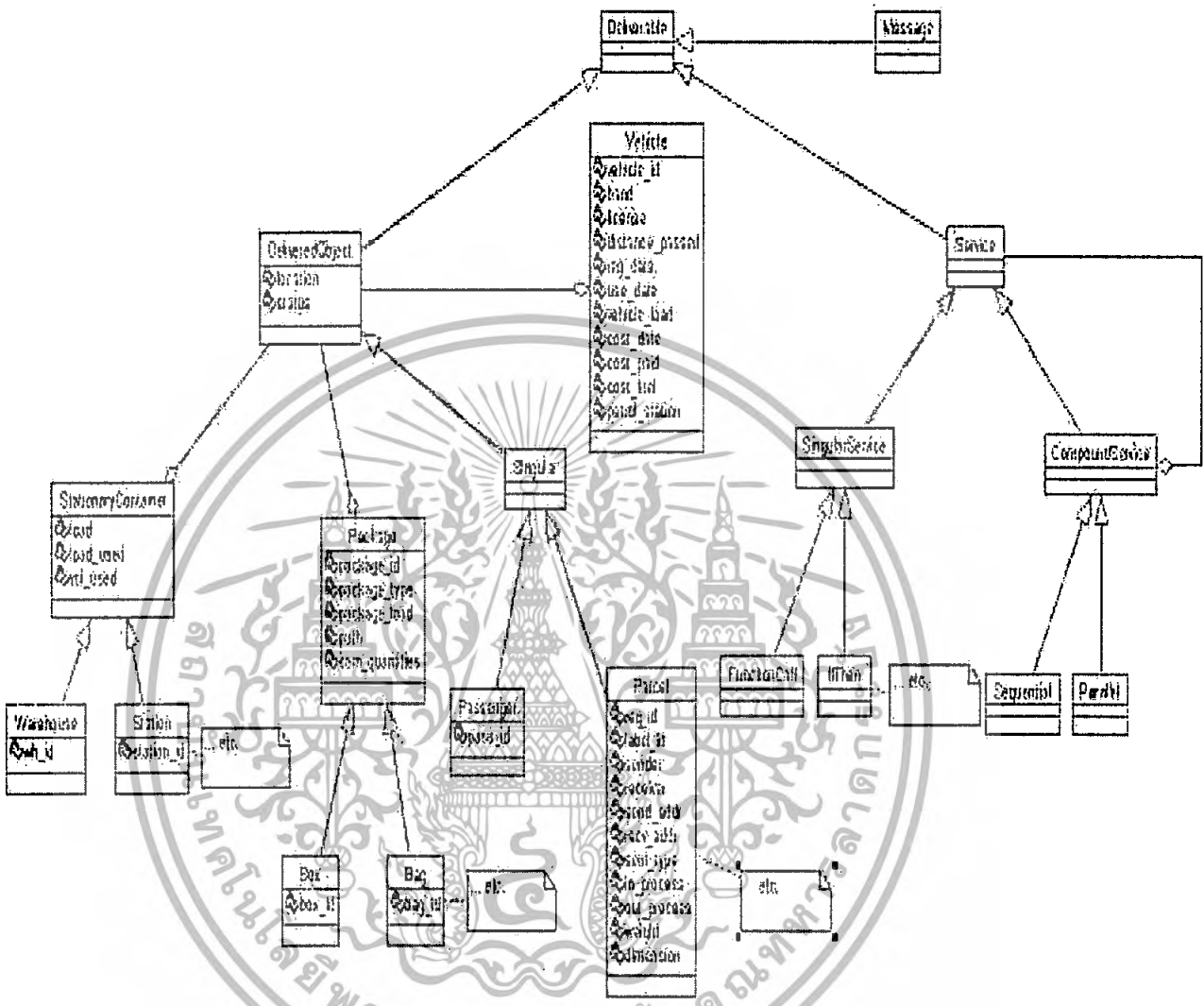
1 เมื่อมีการเปลี่ยนแปลงสถานะการขนส่ง ต้องมีการอัพเดทข้อมูลทุกครั้ง

2 เมื่อผู้ติดตามต้องการรู้สถานะการขนส่ง ต้องมีหมายเลขที่ออกให้เพื่อใช้ในการติดตาม

3 นำหมายเลขนั้นเพื่อตรวจสอบข้อมูลในฐานข้อมูล

### 3.2.2 การออกแบบคลาสไดอะแกรม

Class Diagram ที่ออกแบบแสดงได้ดังรูป




รูปที่ 3.8 แสดง Class Diagram ของระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี

สิ่งที่ขนส่งได้ในระบบเรียกว่า Deliverable ซึ่งเป็นได้ทั้งที่เป็นวัตถุ (DeliveredObject) และการบริการต่างๆ (Service) ในที่นี้จะเน้นที่ DeliveredObject ซึ่งเป็นวัตถุเดี่ยว (Singular) แบ่งเป็นคน (Passenger) และสิ่งของ (Parcel) โดยวัตถุที่ขนส่งได้นั้นจะถูกบรรจุอยู่ได้หลายสิ่งทั้งที่เคลื่อนที่ได้ (Vehicle) และเคลื่อนที่ไม่ได้(StationaryContainer) อีกทั้งวัตถุที่สามารถเป็นสิ่งของที่ถูกขนส่งแต่สามารถบรรจุสิ่งของอื่นได้ (Package)

เราได้ใช้ PostgreSQL เป็น DBMS ในการพัฒนาฐานข้อมูลและนำไปออกแบบตารางฐานข้อมูลดังนี้

**ตาราง Parcel** เก็บข้อมูลของสิ่งของที่รับเข้ามา

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Reg_ID | text | NOT NULL | | Alter Drop | Primary key of Parcel table incresingly |
| Label_ID | text | NOT NULL | | Alter Drop | Tracking code or RFID Tag |
| Sender | text | NOT NULL | | Alter Drop | |
| Send_Addr | text | | | Alter Drop | |
| Receiver | text | NOT NULL | | Alter Drop | |
| Recv_Addr | text | NOT NULL | | Alter Drop | |
| Sent_Type | text | NOT NULL | | Alter Drop | |
| Weight | integer | NOT NULL | | Alter Drop | |
| Dimension | text | | | Alter Drop | width x length x height |
| Postcode | integer | NOT NULL | | Alter Drop | |
| Comment | text | | | Alter Drop | |
| Send_Email | text | | | Alter Drop | |
| Send_Tel | text | | | Alter Drop | |
| Recv_Email | text | | | Alter Drop | |
| Recv_Tel | text | | | Alter Drop | |

**รูปที่ 3.9** แสดงตาราง Parcel

**ตาราง Parcel_Status** เก็บข้อมูลของสิ่งของที่ทำการจัดส่งซึ่งข้อมูลมีการเปลี่ยนแปลงไปตามกระบวนการจัดส่ง

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Reg_ID | text | NOT NULL | | Alter Drop | fk from Parcel |
| Status | text | NOT NULL | | Alter Drop | |
| In_Process | timestamp with time zone | NOT NULL | | Alter Drop | date & time |
| Out_Process | timestamp with time zone | | | Alter Drop | date & time |
| Staff | text | | | Alter Drop | |
| Package | text | | | Alter Drop | |
| Location | integer | NOT NULL | | Alter Drop | |
| Car_ID | integer | | | Alter Drop | |

**รูปที่ 3.10** แสดงตาราง Parcel_Status

**ตาราง Car** เก็บข้อมูลของรถที่ทำหน้าที่จัดส่งสิ่งของ

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Car_ID | integer | NOT NULL | | Alter Drop | |
| Load | integer | | | Alter Drop | |
| Car_Status | text | | | Alter Drop | |
| Route | text | | | Alter Drop | |
| Arrive | text | | | Alter Drop | |

**รูปที่ 3.11** แสดงตาราง Car

**ตาราง Address** เก็บข้อมูลของที่อยู่ปลายทางที่จะทำการจัดส่งสิ่งของ

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Seq | integer | NOT NULL | | Alter Drop | |
| Home_number | text | | | Alter Drop | |
| Address | text | | | Alter Drop | |
| Postcode | integer | | | Alter Drop | |
| x | text | | | Alter Drop | |
| y | text | | | Alter Drop | |

**รูปที่ 3.12** แสดงตาราง Address

**ตาราง Postcode** เก็บข้อมูลรหัสไปรษณีย์ของแต่ละเขต

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Postcode | integer | NOT NULL | | Alter Drop | |
| Province | text | | | Alter Drop | |
| District | text | | | Alter Drop | |

**รูปที่ 3.13** แสดงตาราง Postcode

# บทที่ 4
# การดำเนินงาน

## 4.1 การติดต่อกับอุปกรณ์อาร์เอฟไอดี

ในการพัฒนาระบบโปรแกรมนี้ใช้ผลิตภัณฑ์ของ Intermec ดังนี้

**ตารางที่ 4.1** แสดงอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

| ผลิตภัณฑ์ | ข้อมูลจำเพาะ |
|---|---|
| 700 Series Color Mobile Computer : 761B | Microprocessor : Intel® XScale™ Processor, 400 MHz<br>RAM Memory : 128MB<br>Flash ROM: 64 MB; includes ROM folder for application storage<br>Operating System : Windows Mobile® 2003 |
| Handheld Readers : IP4 | Feature:<br>• Multi-protocol radio for worldwide use<br>• Available for UHF frequency bands worldwide<br>• Infrared data connection to 700 Color |
| Fixed Reader : IP61 | Feature:<br>• Intel® Celeron® processor and ample storage to run complex RFID applications<br>• Hosts applications written in Java®, Java Script or C# .Net<br>• "Store and forward" capabilities ensure data won't be lost<br>• Directly monitors and controls presence detectors and signal lights<br>• Localized workbench to load, edit and run Java Script directly on the reader<br>• Based on EPC global certified radio<br>• Available in 865MHz, 869MHz or 915MHz frequency bands |

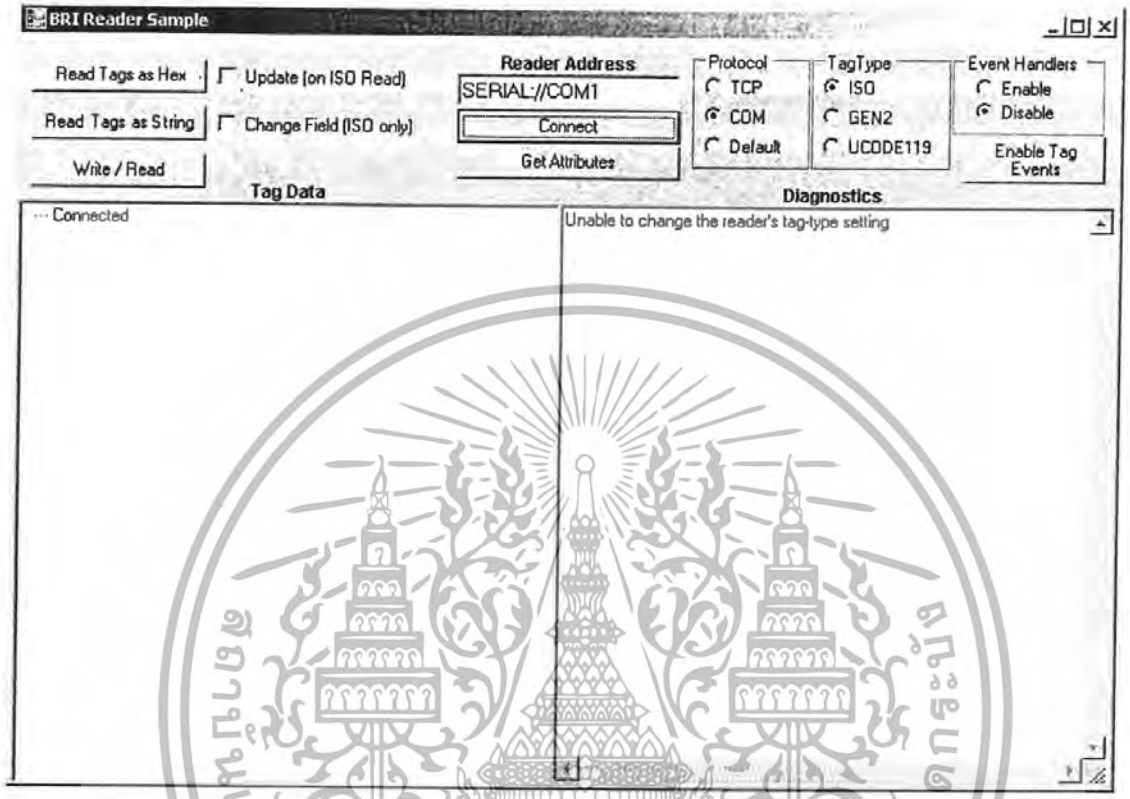
ในการติดต่อกับ RFID Reader ซึ่งในการติดต่อกับ RFID Reader นี้แบ่งการทำงานได้ 2 รูปแบบคือ

1. RFID Reader แบบ Handheld (IP4) ในการติดต่อ Reader แบบนี้จะทำงานในลักษณะ Client-Server โดยตัว Reader จะทำหน้าที่เป็น Client ส่งข้อมูลของแท็กที่อ่านได้ไปที่ Server ซึ่ง Server และ Client นั้นติดต่อกันผ่าน wireless

**รูปที่ 4.1** แสดงการติดต่อระหว่าง Server และ Client (Handheld RFID Reader) ผ่าน wireless

**รูปที่ 4.2** แสดงทดลองการอ่านแท็กด้วย Reader แบบ Handheld และส่งข้อมูลเข้าเครื่อง Server

2. RFID Reader แบบ Fixed (IF61) การติดต่อ Reader จะติดต่อผ่านทางพอร์ตอนุกรม (Serial port)




**รูปที่ 4.3** แสดงการทดลองอ่านแท็กด้วย Reader แบบ Fixed

## 4.2 การจัดการฐานข้อมูล PostgreSQL

ในการพัฒนาระบบได้พัฒนาโมดูลในการติดต่อฐานข้อมูล PostgreSQL 8.1 โดยใช้ PostgreSQL OLE DB Provider เป็นตัวติดต่อซึ่งเขียนเป็นฟังก์ชันให้เรียกใช้งานในภาษาไอรอนไพธอน โดยมีฟังก์ชันในการติดต่อไปยังเซิร์ฟเวอร์ฐานข้อมูลที่เครื่อง 161.246.5.68

```
import clr
import System
clr.AddReference("System.Data")
from System.Data import *
from System.Data.OleDb import *

conString = """Provider=PostgreSQL OLE DB
    Provider;Password=ictlab;User ID=postgres;Data
    Source=161.246.5.68;Location=t1"""
con = OleDbConnection(conString)
```

ฟังก์ชันหลักสำหรับการใช้งานในการติดต่อกับฐานข้อมูล คือ Select , Insert , Delete และ Update มีรายละเอียดดังนี้

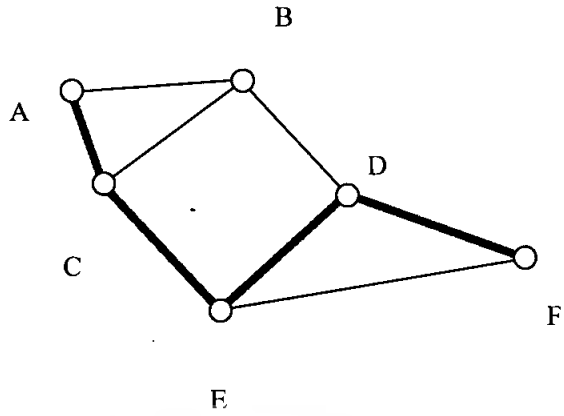
```
def SelectDT(tableName,colName = "*",condition="",dtName = '0')
def InsertRow(tableName,vals)
def InsertColumn(tableName,colName,vals)
def DeleteDT(tableName,condition)
def UpdateDT(tableName,colName,vals,condition)

# tableName : String name of table in DBMS
# colName   : String of columns name separate with comma
            : "id,name,addr"
            ->default "*" every columns
# condition : String of every selected conditions (optional)
            : "where id = 1 order by id"
            ->default is no condition
# dtName    : String name of DataTable
            ->defaule "0"
# vals      : List of string values of each columns separate by comma
              values order by columns of table
            : ['\'string'\','integer']
```

## 4.3 อัลกอริทึมในการค้นหาเส้นทางส่งสิ่งของ

ในการค้นหาเส้นทางส่งสิ่งของจะใช้อัลกอริทึมเอสตาร์มาใช้ในการค้นหาเส้นทางที่สั้นที่สุดในการขนส่งสิ่งของ ซึ่งมีอัลกอริทึมดังนี้

โดยทั่วไปเอสตาร์จะหาเส้นทางที่สั้นที่สุดจากจุด 2 จุด จุดต้นไปยังจุดปลาย แต่ในระบบนี้ต้องการกำหนดให้สถานีย่อยที่ต้องนำส่งสิ่งของเป็นจุดที่ต้องผ่านหลายจุด ซึ่งในการค้นหาแบบมีจุดผ่านหลายจุดนั้น ไม่เพียงแค่จุดสุดท้ายที่ตรวจสอบจะต้องเป็นจุดเป้าหมายเท่านั้น ในเส้นทางที่เราตรวจสอบอยู่จะต้องเป็นเส้นทางที่ผ่านจุดที่ต้องการด้วย ซึ่งเราสามารถอธิบายด้วยทฤษฎีเซตทางคณิตศาสตร์ได้ คือเราจะเสร็จสิ้นการค้นหาก็ต่อเมื่อ เซตของจุดหมาย(เซตของจุดที่ผ่านรวมกับจุดหมาย)เป็นสับเซตของเส้นทางที่ตรวจสอบอยู่ในขณะนั้น และจุดสุดท้ายที่ตรวจสอบเป็นจุดเป้าหมาย

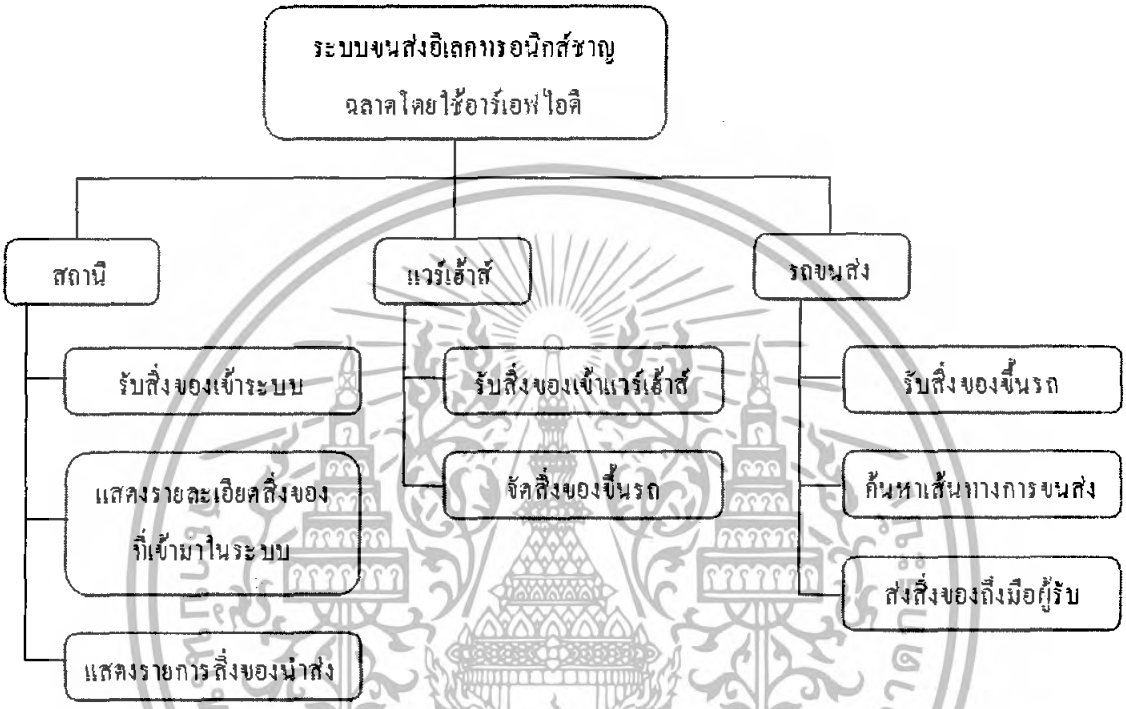



**รูปที่ 4.4** แสดงการใช้อัลกอริทึมเอสตาร์ในการค้นหาเส้นเมื่อกำหนดให้ผ่านหลายจุด

จากรูปข้างต้น เราต้องการเดินทางจากจุด A ไปยังจุด F โดยกำหนดให้ผ่านจุด E และจุด D ด้วย เราจะแทนข้อมูลในรูปของเซตได้ คือให้ S แทนเซตของจุดเริ่มต้น ให้ G คือเซตของจุดเป้าหมาย และ P แทนเซตของจุดเส้นทางที่ตรวจสอบ ดังนี้ S = {A},G={E,D,F}และ P={A,C,E,D,F} จะเห็นได้ว่าเซต G เป็นสับเซตของเซต P และจุดสุดท้ายที่ตรวจสอบเป็นจุดเป้าหมาย

# บทที่ 5
# ผลการทำงานของระบบ

ระบบที่เราได้ออกแบบมีฟังก์ชันการทำงานดังนี้



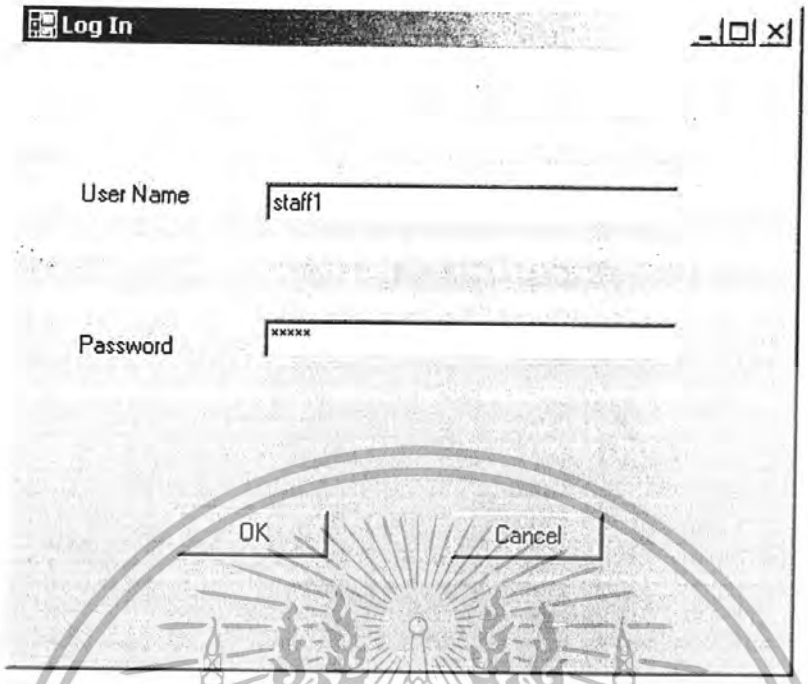

**รูปที่ 5.1** แสดงฟังก์ชันการทำงานโดยรวมของระบบ

เมื่อพัฒนาเป็นแอพพลิเคชั่นตามระบบที่ออกแบบไว้ สามารถแบ่งแอพพลิเคชั่นการทำงานต่างๆได้ดังนี้

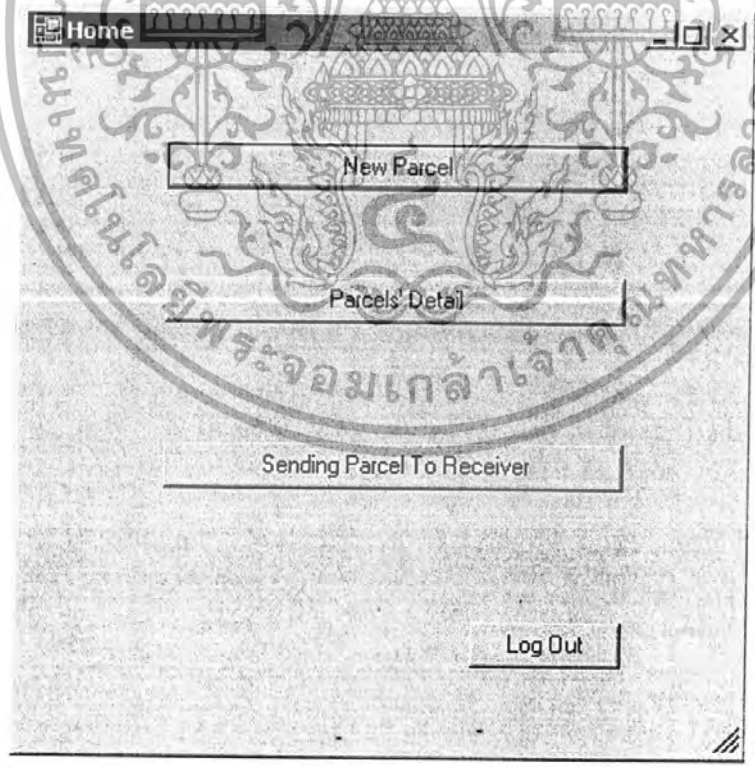
## 5.1 แอพพลิเคชั่นในส่วนของสถานี

แอพพลิเคชั่นในส่วนของสถานีเป็นแอพพลิเคชั่นที่มีประจำอยู่ในทุกสถานีย่อยที่ทำหน้าที่ในการรับสิ่งของจากผู้ใช้งานเข้ามาในระบบ และทำการจัดส่งสิ่งของไปยังผู้รับ ผู้ใช้งานแอพพลิเคชั่นในส่วนนี้คือเจ้าหน้าที่ที่ประจำแต่ละสถานี

ในหน้าแรกของแอพพลิเคชั่นจะเป็นหน้าจอการล็อกอินเพื่อเข้ามาใช้งานในระบบ ใช้งานโดยการกรอก Username และ Password ลงไปเพื่อทำการล็อกอินเข้าสู่ระบบ




**รูปที่ 5.2** แสดงแอพพลิเคชั่นในส่วนของสถานี : หน้าล็อกอินเข้าใช้งาน

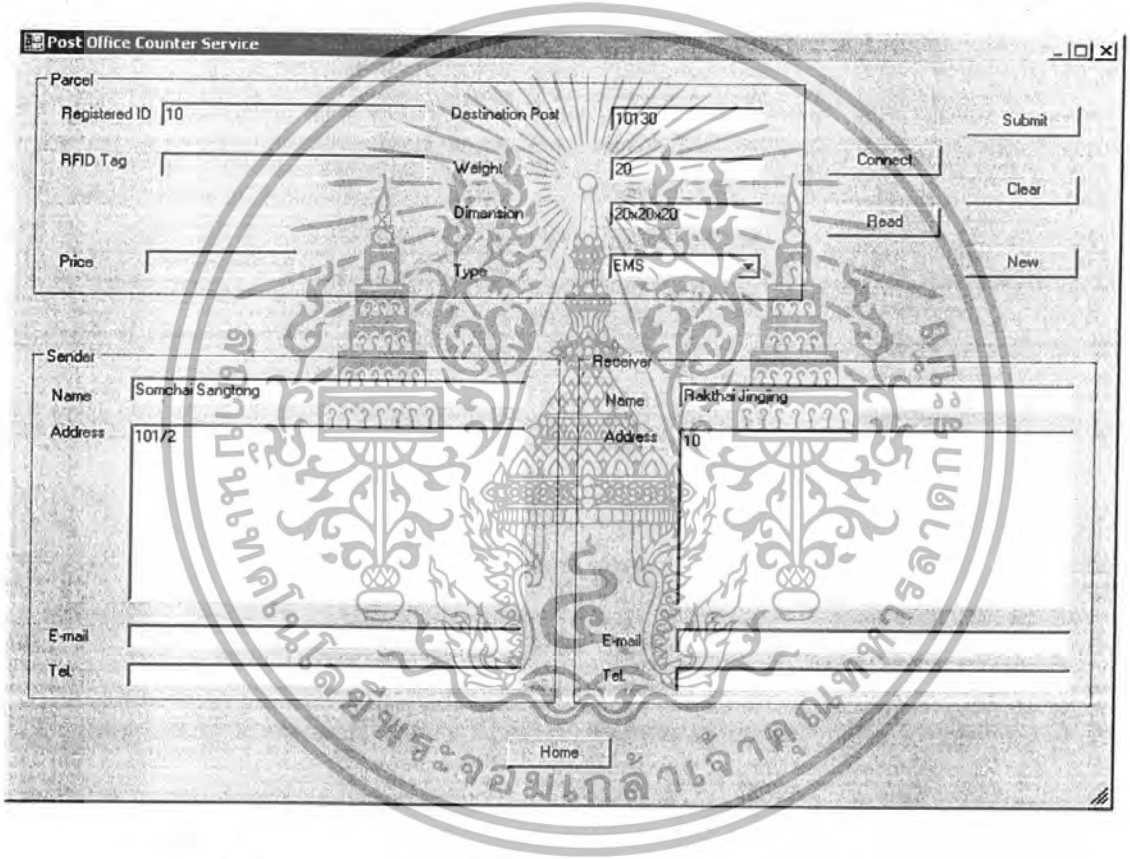



**รูปที่ 5.3** แสดงแอพพลิเคชั่นในส่วนของสถานี : เลือกการทำงาน

เมื่อผ่านหน้าล็อกอินเข้ามาจะเข้าสู่หน้าจอที่แสดงโหมดการทำงานหลักของแอพพลิเคชั่นในส่วนของสถานีซึ่งมีการทำงาน 3 ส่วนประกอบด้วย

1. รับสิ่งของเข้าระบบ
2. แสดงสถานะของสิ่งของที่ได้รับภายในสถานีนั้นๆ
3. แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ

**1. รับสิ่งของเข้าระบบ**




**รูปที่ 5.4** แสดงแอพพลิเคชั่นในส่วนของสถานี : รับสิ่งของเข้าระบบ

ในการทำงานของแอพพลิเคชั่นในส่วนของสถานี หน้ารับสิ่งของเข้าระบบเป็นส่วนที่ทำหน้าที่เก็บรายละเอียดต่างๆของสิ่งของเพื่อนำสิ่งของเข้ามาในระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีโดย หน้าจอจอรับสิ่งของเข้าระบบมีรายละเอียดดังนี้

1. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่ในช่อง RFID Tag

2. ปุ่ม Read อ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลเป็นรหัสของแท็กอาร์เอฟไอดีที่อ่านได้ในช่อง RFID Tag

3. ปุ่ม Summit เก็บรายละเอียดของสิ่งของลงในฐานข้อมูล ข้อมูลของสิ่งของจะถูกบันทึกลงในฐานข้อมูลโดยมีสถานะของการส่งสิ่งของเป็น 'Get in system'

4. ปุ่ม Clear ลบรายละเอียดทั้งหมดที่กรอกภายในหน้าจอรับสิ่งของนี้

5. ปุ่ม New เริ่มต้นรับสิ่งของชิ้นใหม่

6. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ

7. Textbox Registered ID เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นรหัสที่สร้างขึ้นมาเพื่อใช้ในระบบ แทนที่จะใช้รหัสของแท็กอาร์เอฟไอดีตรงๆ เพื่อความปลอดภัยของระบบ

8. Textbox RFID Tag เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นรหัสที่อ่านได้จากแท็กอาร์เอฟไอดี

9. Textbox Price เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นราคาค่าบริการในการจัดส่งสิ่งของโดยคำนวนจากน้ำหนักของสิ่งของและประเภทของการส่งสิ่งของ

10. Textbox Destination Post เจ้าหน้าที่กรอกรหัสไปรษณีย์ของสถานีย่อยที่อยู่ของผู้รับปลายทาง

11. Textbox Weight เจ้าหน้าที่กรอกน้ำหนักของสิ่งของ

12. Textbox Dimension เจ้าหน้าที่กรอกปริมาตรของสิ่งของ

13. ComboBox Type เจ้าหน้าที่เลือกชนิดของการจัดส่งสิ่งของ (EMS หรือ Ordinary)

14. Textbox Sender Name เจ้าหน้าที่กรอกชื่อ นามสกุลของผู้ส่ง

15. Textbox Sender Address เจ้าหน้าที่กรอกรายละเอียดที่อยู่ของผู้ส่ง

16. Textbox Sender E-mail เจ้าหน้าที่กรอก E-mail ของผู้ส่ง

17. Textbox Sender Tel. เจ้าหน้าที่กรอกเบอร์โทรศัพท์ของผู้ส่ง

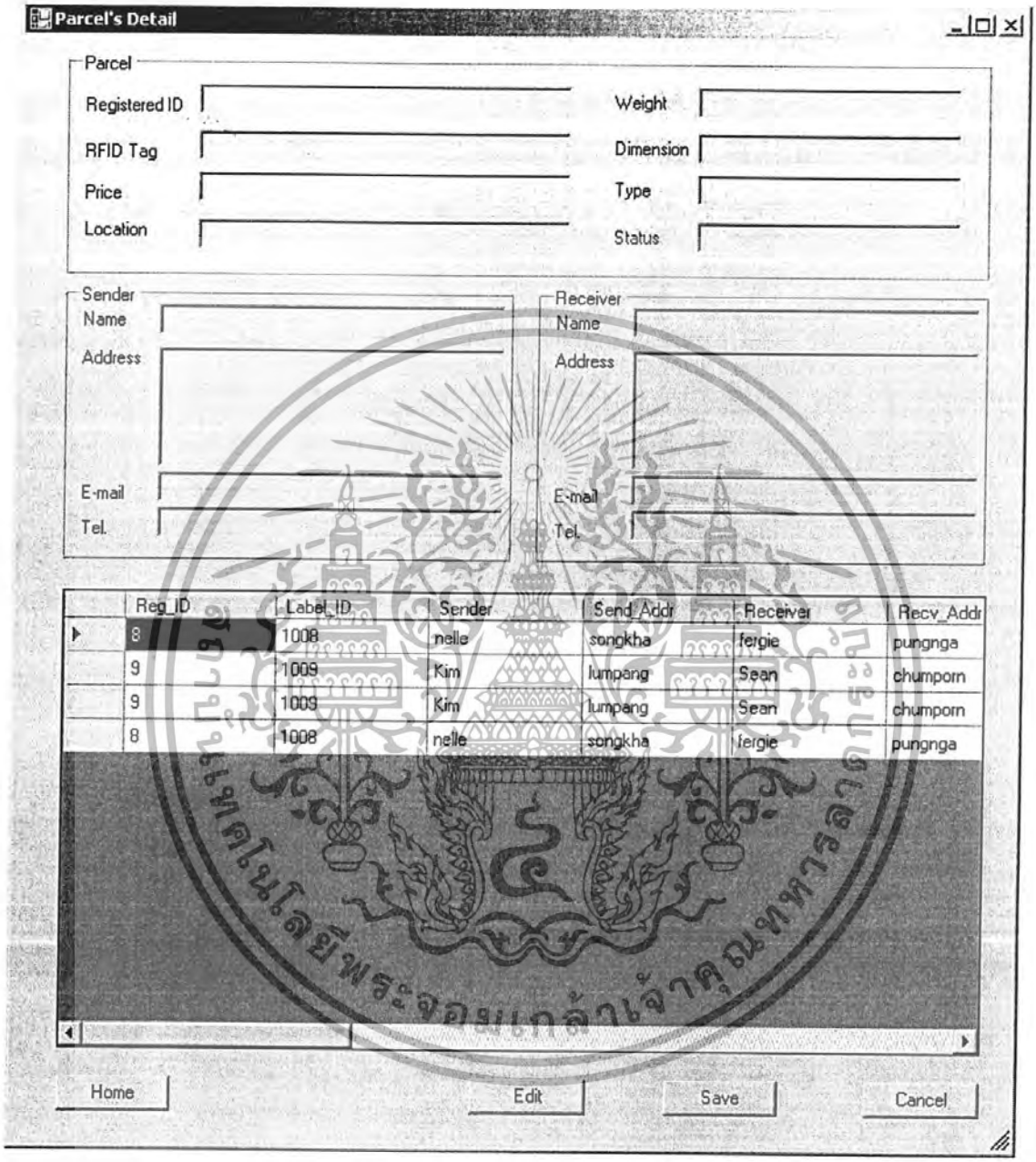
18. Textbox Receiver Name เจ้าหน้าที่กรอกชื่อ นามสกุลของผู้รับ

19. Textbox Receiver Address เจ้าหน้าที่กรอกรายละเอียดที่อยู่ของผู้รับ

20. Textbox Receiver E-mail เจ้าหน้าที่กรอก E-mail ของผู้รับ

21. Textbox Receiver Tel. เจ้าหน้าที่กรอกเบอร์โทรศัพท์ของผู้รับ

**2. แสดงสถานะของสิ่งของที่ได้รับภายในสถานีนั้นๆ**




**รูปที่ 5.5** แสดงแอพพลิเคชั่นในส่วนของสถานี: สถานะของสิ่งของที่ได้รับภายในสถานี

หน้าแสดงรายละเอียดของสิ่งของที่รับของมาภายในสถานีนั้นๆโดยจะแสดงสถานะล่าสุดของสิ่งของชิ้นนั้นๆด้วยว่ามีสถานะการส่งสถานะอะไรและสิ่งของชิ้นนั้นอยู่ที่ไหน หน้าแสดงรายละเอียดของสิ่งของมีรายละเอียดดังนี้

1. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ

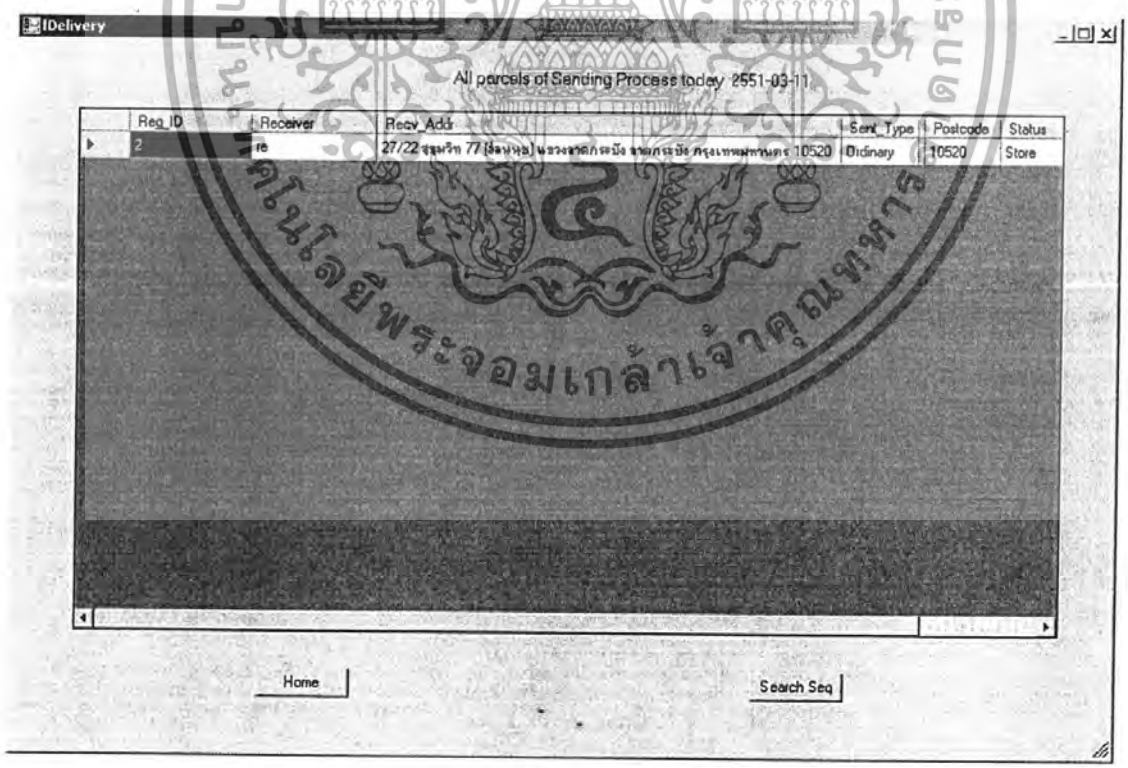
2. ปุ่ม Edit เจ้าหน้าที่สามารถกดเลือกสิ่งของที่ต้องการทำการแก้ไขโดยเลือกจากตารางที่แสดงผล (DataGridView) สิ่งของที่ถูกเลือกจะแสดงรายละเอียดใน Textbox ด้านบน จากนั้นกดปุ่ม Edit เพื่อเข้าไปทำการแก้ไขสิ่งของชิ้นนั้นๆ ปุ่ม Edit จะเปลี่ยนเป็นปุ่ม OK กดปุ่ม OK เมื่อทำการแก้ไขสิ่งของเสร็จเรียบร้อย สิ่งของที่ถูกแก้ไขจะแสดงรายละเอียดใหม่ในตาราง

3. ปุ่ม Save เจ้าหน้าที่กดเมื่อต้องการทำการบันทึกข้อมูลสิ่งของที่มีการแก้ไขลงในฐานข้อมูล

4. ปุ่ม Cancel กดเพื่อยกเลิกการเปลี่ยนแปลงใดๆ

5. DataGridview แสดงรายละเอียดของสิ่งของที่ได้รับจากผู้ส่งภายในสถานีนั้นๆ ซึ่งจะแสดงรายละเอียดและสถานะการส่งสิ่งของล่าสุดของสิ่งของชิ้นนั้นๆด้วย เจ้าหน้าที่สามารถเลือกกดที่สิ่งของเพื่อแก้ไขข้อมูลของสิ่งของได้ โดยจะแสดงรายละเอียดของสิ่งของและให้ทำการแก้ไขได้ในส่วนTextbox ด้านบน แต่การแก้ไขจะจำกัดเฉพาะสิ่งของที่ยังอยู่ภายในสถานีและยังไม่ถูกส่งต่อเข้ากระบวนการอื่นๆเท่านั้น

### 3. แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ




**รูปที่ 5.6** แสดงแอพพลิเคชั่นในส่วนของสถานี : แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ

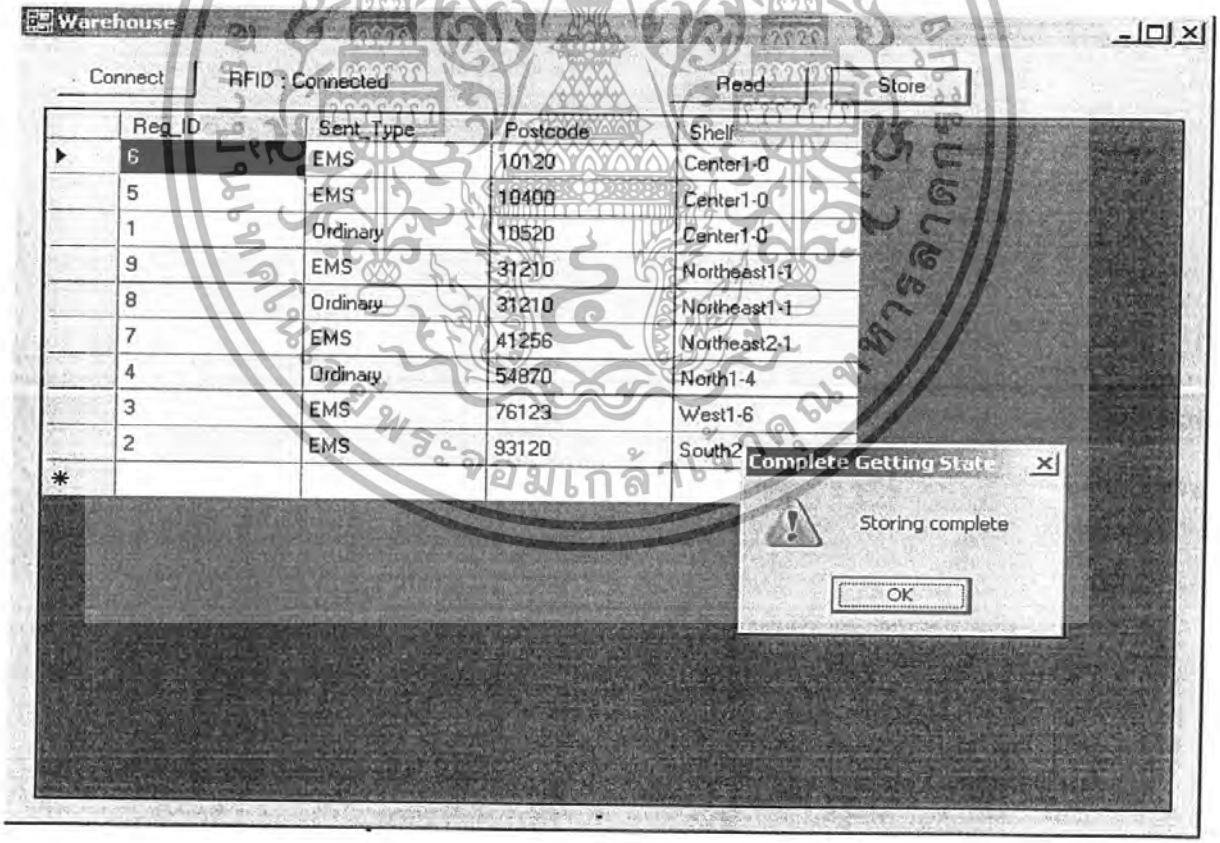
แอพพลิเคชั่นหน้าแสดงรายการส่งสิ่งของไปถึงมือผู้รับนี้ มีเพื่อแสดงรายละเอียดของสิ่งของที่ต้องนำไปส่งถึงมือผู้รับ มีรายละเอียดดังนี้

1. DataGridView แสดงรายละเอียดของสิ่งของที่ส่งถึงสถานีปลายทางนั้น
2. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ
3. ปุ่ม Search Seq. เจ้าหน้าที่กดเพื่อดูลำดับการส่งสิ่งของตามความเหมาะสมที่โปรแกรมคำนวณให้

## 5.2 แอพพลิเคชั่นในส่วนของแวร์เฮาส์

แอพพลิเคชั่นในส่วนของแวร์เฮ้าส์จะแบ่งการทำงานเป็น 2 ส่วนคือ

1. **รับสิ่งของเข้าแวร์เฮ้าส์** แอพพลิเคชั่นใช้สำหรับการอัพเดทข้อมูล และแบ่งจัดเก็บสิ่งของในแวร์เฮ้าส์โดยแบ่งตามภูมิภาค และรหัสไปรษณีย์เพื่อความสะดวกต่อการจัดส่งในขั้นถัดไป สิ่งของที่เข้ามาเก็บในแวร์เฮ้าส์จะมีสถานะของการส่งสิ่งของเป็น 'Store'




**รูปที่ 5.7** แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : รับสิ่งของเข้าแวร์เฮ้าส์

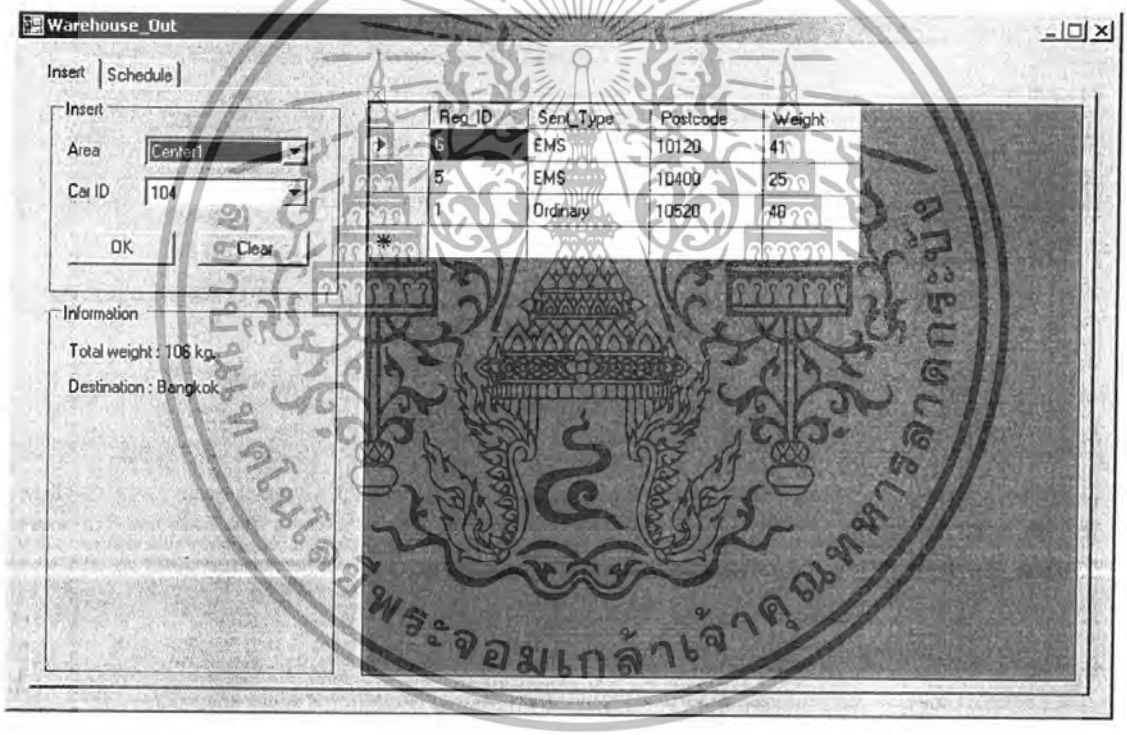
แอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ ที่ทำหน้าที่รับสิ่งของเข้า มีรายละเอียดดังนี้

1. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่

2. ปุ่ม Read อ่านแท็กอาร์เอฟไอดีจากนั้นจะนำรหัสที่อ่านได้จากแท็กอาร์เอฟไอดีไปดึงข้อมูลในฐานข้อมูลมาแสดงว่ามีสิ่งของอะไรบ้างที่เข้ามาในขั้นตอนการรับของเข้าแวร์เฮ้าส์

3. ปุ่ม Store อัพเดทข้อมูลว่าสิ่งของที่อ่านแท็กอาร์เอฟไอดีได้รับเข้ามาเก็บในแวร์เฮ้าส์แล้วโดยจะมีแบ่งการเก็บสิ่งของไปตามชั้นเก็บของตามภูมิภาคและรหัสไปรษณีย์

**2. จัดสิ่งของขึ้นรถขนส่ง** เป็นขั้นตอนที่จะนำสิ่งของขึ้นรถเพื่อส่งสิ่งของไปสถานีหลักของแต่ละภูมิภาค เพื่อทำการจัดส่งในขั้นถัดไป




**รูปที่ 5.8** แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : จัดสิ่งของขึ้นรถ

รายละเอียดของแอพพลิเคชั่นในส่วนของการจัดสิ่งของขึ้นรถ มีดังนี้

1. ComboBox Area เจ้าหน้าที่ทำการเลือกภูมิภาคในการส่ง เมื่อเลือกภูมิภาคแล้ว DataGridView ด้านซ้ายจะแสดงข้อมูลของสิ่งของที่ต้องจัดส่งไปที่ภูมิภาคนั้นๆทั้งหมด โดยจะเรียงจากชนิดของการขนส่งโดยให้การส่งแบบ EMS มีความสำคัญสูงที่สุด จากนั้นจึงเรียงลำดับจากเวลาที่สิ่งของเข้ามาในระบบ ด้านซ้ายจะเป็นส่วนแสดงข้อมูลว่าน้ำหนักรวมของสิ่งของเป็นเท่าไหร่และปลายทางในการขนส่งอยู่ที่จังหวัดไหน

2. ComboBox Car_ID แสดงเบอร์รถขนส่งที่มีสถานะพร้อมจะส่งได้ กำหนดให้รถสามารถรองรับสิ่งของได้น้ำหนัก 1000 กิโลกรัม ไม่จำกัดปริมาตรไม่การขนส่ง

3. ปุ่ม OK เพื่อทำการจัดสิ่งของที่จะทำการขนส่งให้กับรถขนส่งตามเบอร์รถที่เลือกไป

4. ปุ่ม Clear เพื่อเริ่มต้นการจัดสิ่งของขึ้นรถใหม่

## 5.3 แอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของ

รถจัดส่งมี 2 ประเภท คือ รถประจำแวร์เฮาส์ และรถประจำสถานีย่อย หน้าที่โดยทั่วไปเหมือนกัน จะแตกต่างกันเล็กน้อย

รถประจำแวร์เฮาส์ รถจัดส่งสิ่งของจะเริ่มต้นการทำงานที่แวร์เฮ้าส์โดยการรับสิ่งของเข้ามาในรถจากนั้นจะทำการจัดส่งไปยังสถานีหลักของแต่ละภูมิภาคและจะกระจายไปยังสถานีของแต่ละจังหวัด ไปถึงสถานีย่อยปลายทางของสิ่งของในรถนั้นๆ

**รูปที่ 5.9** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Get in

รายละเอียดของแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของมีดังนี้

1. ComboBox Car_ID เพื่อเลือกว่ารถที่ต้องการจัดการเป็นเบอร์อะไร

2. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่

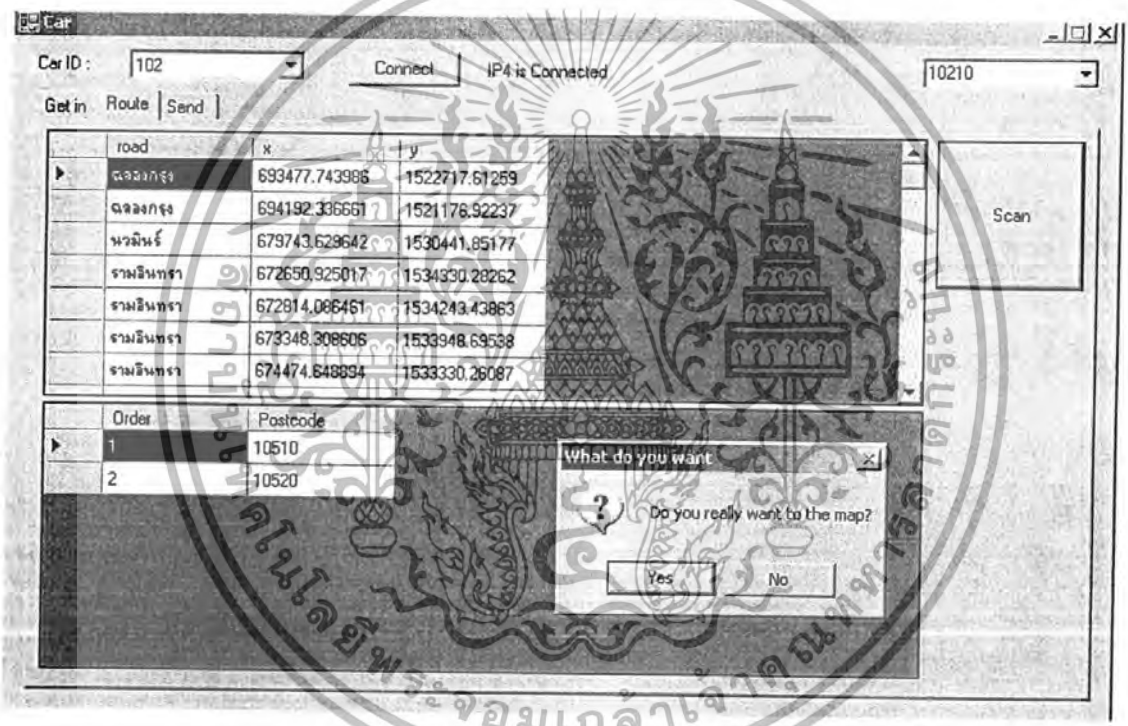
3. ปุ่ม Scan อ่านแท็กอาร์เอฟไอดีจากนั้นจะนำรหัสที่อ่านได้จากแท็กอาร์เอฟไอดีไปดึงข้อมูล

ในฐานข้อมูลมาแสดงว่ามีสิ่งของอะไรบ้างที่อ่านได้และจะแสดงผลที่ DataGridView โดยจะแสดงสถานะของสิ่งของแต่ละแถวตามสีซึ่งมีความหมายดังนี้

- สีน้ำเงิน เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจพบ
- สีเหลือง เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจ**ไม่**พบ
- สีแดง **ไม่**เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจพบ

4. ปุ่ม Update ทำการอัพเดทสถานะของสิ่งของ ถ้านำสิ่งของเข้ารถจอัพเดทสถานะของสิ่งของเป็น 'Sending' แต่ถ้านำของออกจากรถจะอัพเดทสถานะเป็น 'Store'

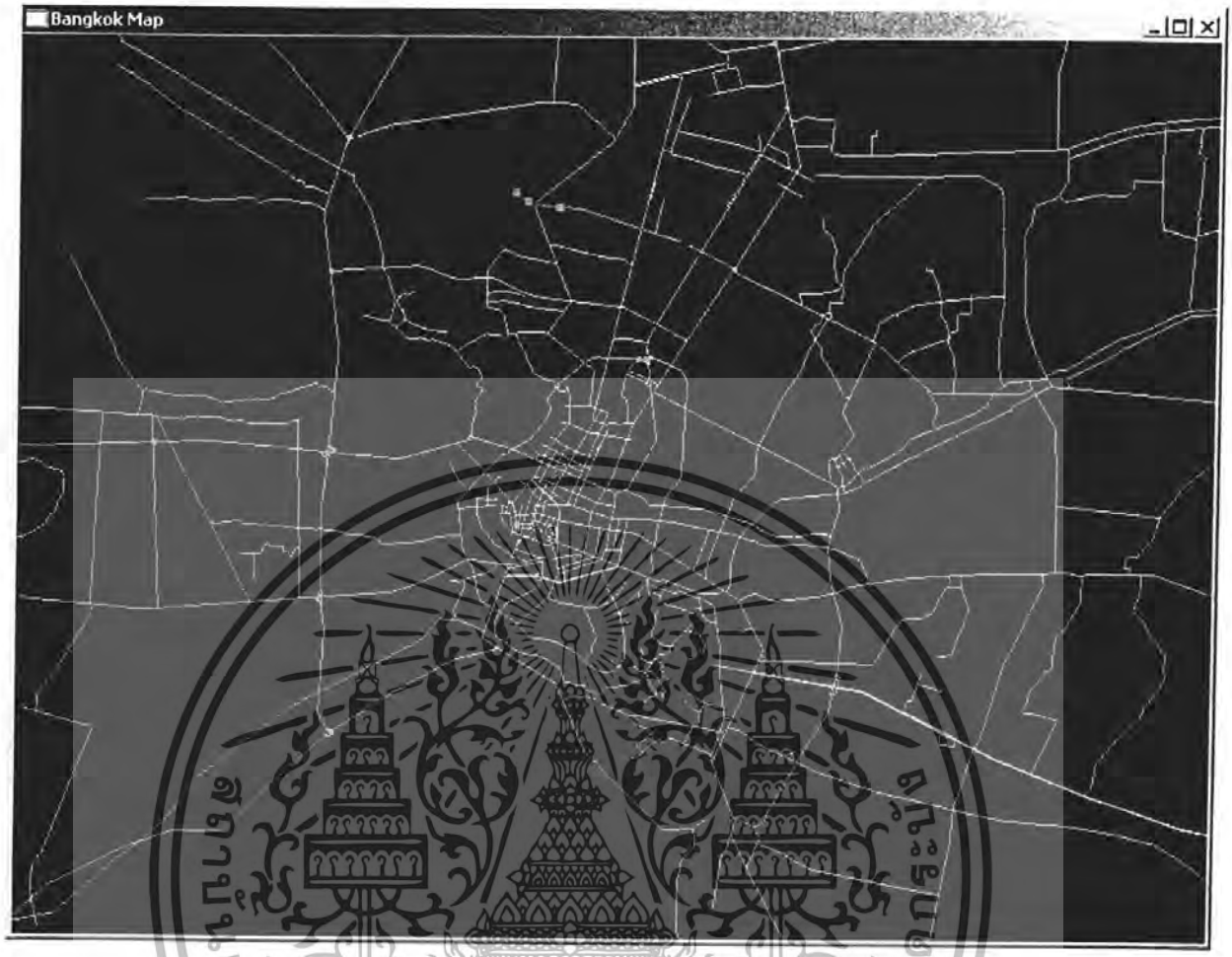
5. ComboBox เพื่อบอกว่าปัจจุบันรถอยู่ ณ สถานีไหน




**รูปที่ 5.10** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Route

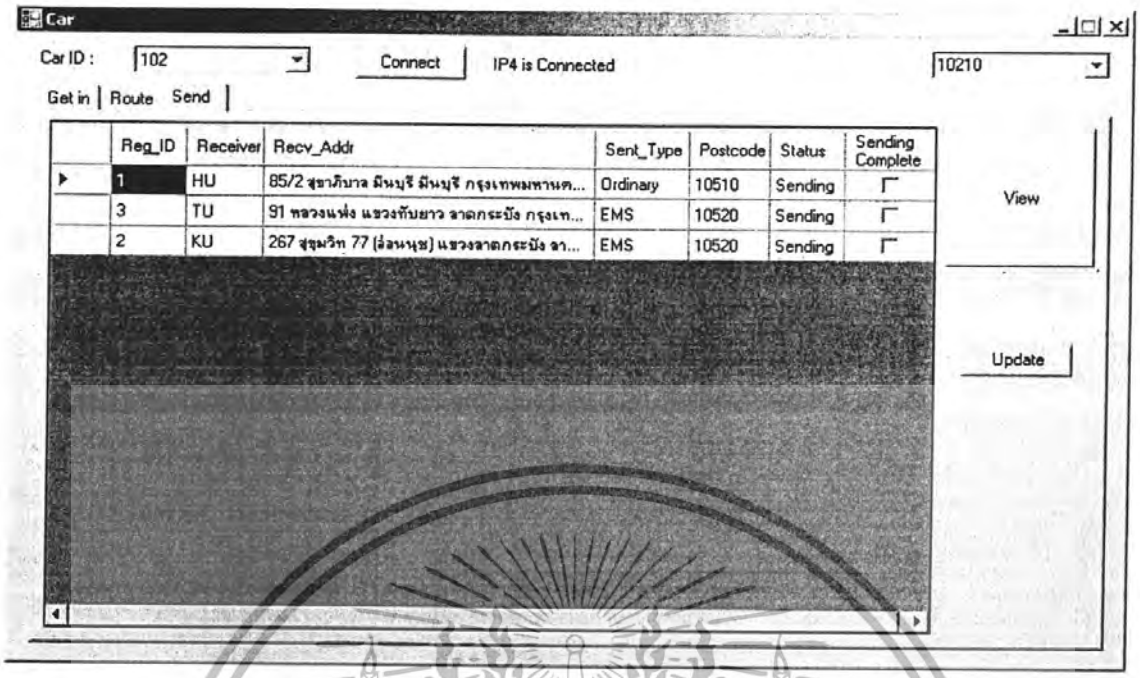
เมื่อขนสิ่งของขึ้นรถเรียบร้อยแล้วก็จัดการเรื่องเส้นทางการขนส่งที่แท็บ Route โดยจะมีปุ่ม Scan ให้กดแสดงผลลัพธ์ใน DataGridView 2 อันคือ อันบนจะแสดงเส้นทางเป็นชื่อถนนและพิกัดตามลำดับที่ได้จากการค้นหาเส้นทางผ่านอัลกอริทึม ส่วนอันล่างจะแสดงลำดับว่าต้องไปที่สถานีย่อยใดก่อนหลังตามลำดับ

นอกจากนี้ยังมีหน้าต่างย่อย popup ออกมาเพื่อให้เลือกว่าต้องการแสดงแผนที่การเดินทางหรือไม่ โดยเส้นทางเป็นแผนที่กรุงเทพฯ ถนนวาดแทนด้วยเส้นสีเขียว จุดเริ่มต้นจะแทนด้วยสีน้ำเงิน จุดต่อไปแทนด้วยสีส้ม และจุดสุดท้ายแทนด้วยสีชมพู



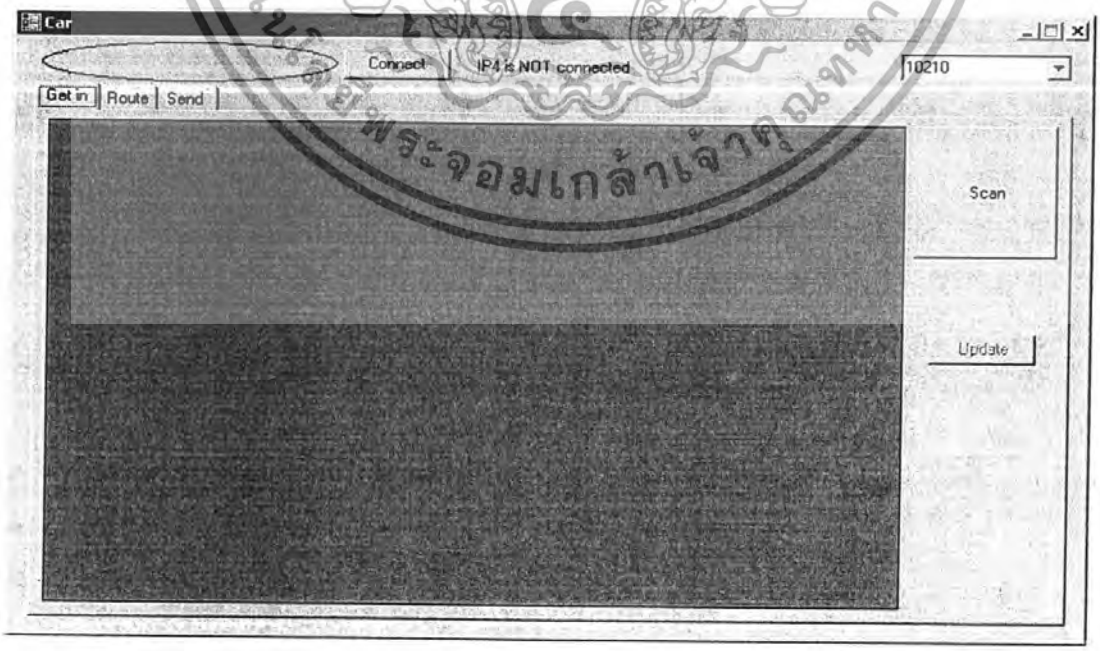

**รูปที่ 5.11** แสดงแผนที่ที่รถจัดส่งสิ่งของต้องวิ่งผ่าน

ระหว่างการขนส่งในแท็บ Send มีปุ่ม View ให้กด เพื่อแสดงลำดับของสิ่งของที่ต้องส่งไปสถานีย่อยใดๆใน DatagridView ภายในคอลัมน์สุดท้ายจะมีช่อง CheckBox ให้เลือกว่าทำการส่งเรียบร้อยแล้วหรือยัง เมื่อถึงสถานีย่อยที่ต้องไปส่ง โดยต้องเลือกรหัสไปรษณีย์ของสถานีย่อยนั้นตรง DropDown มุมบนขวาก่อน จึงกดปุ่ม Update เพื่อบันทึกข้อมูลลงฐานข้อมูลว่าของถึงสถานีย่อยปลายทางแล้ว สถานะของสิ่งของจะเป็น “Store” อยู่ที่สถานีย่อยนั้น




**รูปที่ 5.12** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Send

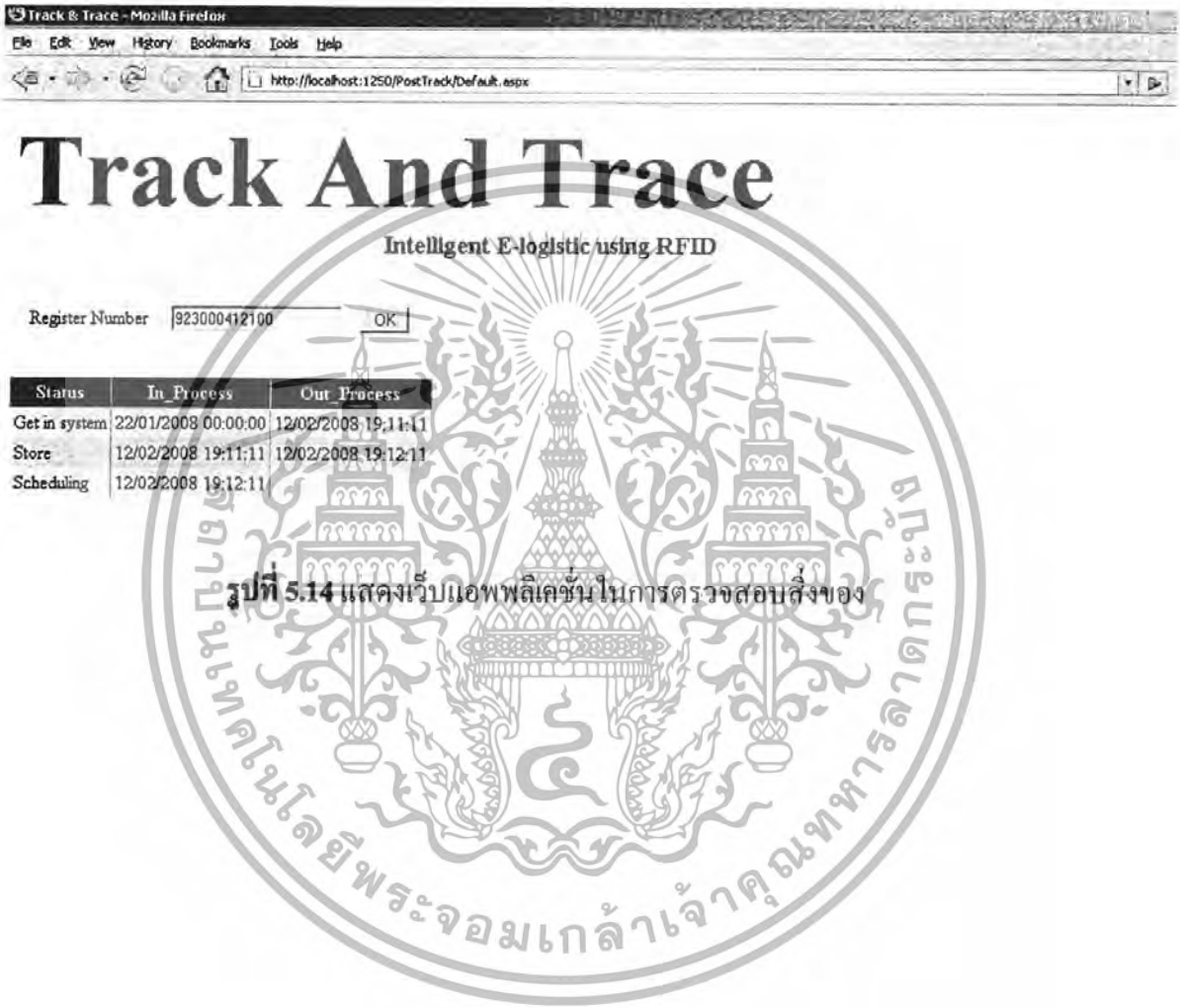
รถประจำสถานีย่อย รถจัดส่งสิ่งของจะเริ่มต้นการทำงานที่สถานีย่อยนั้นๆ โดยใช้รหัสไปรษณีย์บ่งบอก จะมีแต่การส่งออกของไปยังที่อยู่ผู้รับเท่านั้น หน้าต่างแอพพลิเคชั่นก็จะแตกต่างจากของรถประจำแวร์เฮาส์เล็กน้อย คือ ไม่ต้องมีส่วน ComboBox สำหรับเลือกรถจัดส่ง




**รูปที่ 5.13** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำสถานีย่อย

## 5.4 เว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ

ผู้ใช้งานระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีสามารถทำการตรวจสอบสถานะของสิ่งของได้ว่าอยู่ขั้นตอนส่งในขั้นใดโดยใช้บริการของเว็บแอพพลิเคชั่น และกรอกของรหัสสิ่งของลงไป เว็บจะทำการแสดงรายละเอียดของการส่งสิ่งของชิ้นนั้นๆขึ้นมา




**รูปที่ 5.14** แสดงเว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ

# บทที่ 6
# บทสรุปและวิจารณ์

## 6.1 บทสรุปและวิจารณ์

ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี เป็นระบบที่พัฒนาขึ้นมาโดยมีจุดประสงค์หลักคือ ลดเวลาและค่าใช้จ่ายในการขนส่งสิ่งของจนถึงมือผู้บริโภค ซึ่งระบบมีการนำเทคโนโลยีอาร์เอฟไอดีมาพัฒนาเพื่อที่จะนำมาแทนที่ระบบเก่าที่ใช้บาร์โค้ด ทำให้ระบบสามารถทำงานได้รวดเร็วขึ้นเนื่องจากสามารถระบุถึงตัวสิ่งของแต่ละชิ้นได้อย่างรวดเร็วและแม่นยำมากกว่าเดิม ทั้งยังส่งผลให้ระบบมีความน่าเชื่อมากขึ้นด้วย เนื่องจากแท็กอาร์เอฟไอดีนั้นมีคุณสมบัติที่ยากต่อการปลอมแปลง ระบบยังมีการติดตามสิ่งของโดยให้บริการเป็นเว็บแอพพลิเคชั่นให้ผู้ใช้งานสามารถตรวจสอบและติดตามการขนส่งสิ่งของได้

นอกจากการนำเทคโนโลยีอาร์เอฟไอดีมาใช้ในระบบแล้วยังมีการพัฒนาระบบให้มีความสามารถสูงขึ้นโดยการนำอัลกอริทึมเอสตาร์มาใช้เพื่อค้นหาเส้นทางในการขนส่งสิ่งของให้ถึงปลายทาง ทำให้ระบบสามารถค้นหาเส้นทางที่สั้นที่สุดในการขนส่งสิ่งของได้ ซึ่งส่งผลให้เวลาในการขนส่งสิ่งของลดลงไปอีก ทั้งนี้เส้นทางการขนส่งที่สั้นที่สุดอาจไม่ใช่เส้นทางที่ใช้เวลาในการขนส่งน้อยที่สุด ดังนั้นจึงควรมีการเก็บข้อมูลของการขนส่งสิ่งของในแต่ละครั้ง เพื่อนำมาเป็นข้อมูลในการตัดสินใจและเปรียบเทียบการเลือกเส้นทางในการขนส่งครั้งต่อๆไป

## 6.2 ปัญหาที่เกิดขึ้นในการพัฒนา

- ในการพัฒนาระบบมีการใช้ DBMS หลายตัวเช่น PostgreSQL Oracle MySQL จึงต้องศึกษาการใช้งานฐานข้อมูลแต่ละตัวเพื่อนำมาพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี
- เนื่องจากภาษาไอรอนไพธอนยังไม่เป็นที่นิยมนัก หรือยังไม่ค่อยรู้จัก จึงต้องศึกษาค้นคว้าโดยอาศัยประสบการณ์การเขียนภาษาไพธอนและดอทเน็ต ซึ่งมีบางสิ่งแต่กต่างจากวิธีที่คุ้นเคย ต้องอาศัยการสอบถามพูดคุยผ่านทางจดหมายอิเลคทรอนิกส์กับผู้ที่ใช้เป็นประจำหรือมีประสบการณ์ที่คล้ายๆกันหรือมากกว่า หรือต้องสร้างความคุ้นเคยในใช้ระยะเวลาหนึ่งก็จะสามารถใช้งานได้ดี

## 6.3 แนวทางการพัฒนาต่อ

- เลือกแบ่งกลุ่มเขตการจัดส่งสิ่งของ เพื่อลดจำนวนสิ่งของในการค้นหาเส้นทางการขนส่งสิ่งของให้น้อยลงเพื่อให้สอดคล้องกับอัลกอริทึมในการค้นหาเส้นทาง
- ปรับปรุงอัลกอริทึมในการค้นหาเส้นทางการขนส่งสิ่งของให้สามารถค้นหาเส้นทางที่รองรับสิ่งของได้จำนวนมากยิ่งขึ้น
- เพิ่มข้อมูลในฐานข้อมูลในส่วนของตาราง Address เพื่อรองรับสถานที่ในการส่งให้มากยิ่งขึ้น

# บรรณานุกรม


กัลยา อุดมวิทิต, อภิญญา กมลสุข และคณะ. 2548. หน้า 6. ใน **รายงานเบื้องต้นสถานภาพของอุตสาหกรรม RFID ในประเทศไทย.** ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

ขวัญชนก วิริยกุลโอภาศ. 2549. "RFID." หน้า 6 – 13. ใน **รายงานการศึกษารายวิชาเทคโนโลยีสำหรับบรรณารักษศาสตร์และสารนิเทศศาสตร์.** กรุงเทพฯ : คณะอักษรศาสตร์ จุฬาลงกรณมหาวิทยาลัย.

ทวีศักดิ์ กออนันตกูล. 2548. **เทคโนโลยี RFID กับผลกระทบต่อประเทศไทย.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

นระ คมนามูล. 2550. "**ความหมาย *โลจิสติกส์* ในด้านการขนส่ง".** [Online]. Available : http://www.tpa.or.th/writer/read_this_book_topic.php?passTo=f241731bd26874e0a13071bd26874e0a13071a66b4f289d&pageid=1&bookID=589&read=true&count=true.

วัชรากร หนูทอง. ม.ป.ป. **RFID หนึ่งในเทคโนโลยีที่น่าจับตามอง.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

วัชรากร หนูทอง และ อนุกูล น้อยไม้. ม.ป.ป. **RFID หนึ่งในเทคโนโลยีที่น่าจับตามอง.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

ศูนย์พัฒนาธุรกิจออกแบบวงจรรวม. 2548. **รู้จักกับเทคโนโลยีอาร์เอฟไอดี.** ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

สุชาติ กิจธนะเสรี. 2549. **Logistics RFID.** สถาบันคอมพิวเตอร์.

Brooks Automation. **"RFID Basics".** [Online]. Available : http://www.ready4rfid.com/rfid_index.php?menu=rfid_frequencies&spr=e.

Daniel, M. Dobkin. 2005. **"The RF in RFID".** [Online]. Available : http://www.enigmatic-consulting.com/Communications_articles/RFID/RFID_frequencies.html

Hunt, V. Daniel, Puglia, Albert and Puglia, Mike. 2007. **RFID-A GUIDE TO RADIO FREQUENCY IDENTIFICATION.** New Jersey : John Wiley & Sons.

IBM Press. 2005. **"RFID Sourcebook".** [Online]. Available : http://wireless.itworld.com/4985/051004_book_rfidsourcebook/page_1.html

Identify Limited. 2549. **RFID in Logistics & Supply Chain.** [สไลด์]. นนทบุรี : Identify Limited.

Smith Sukamith. 2004. "RFID for Smart Logistics" ใน **Logistics Asia 2004 Conference**

Smith Sukamith. 2007. **RFID Technology & Applications.** [สไลด์]. ปทุมธานี : สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ.

**"Logistic System".** [Online]. Available : http://www.jb-hg.com/elibrary/logistic.htm#important

**"Radio-frequency identification".** [Online]. Available : http://en.wikipedia.org/wiki/Radio-frequency_identification.

**"RFID Technology".** [Online]. Available : http://www.speakingeye.com/index.php?lay=show&ac=article&Id=401359&Ntype=2.

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

**Intelligent E-Logistics with RFID**

พุทธพงษ์ เชี่ยวเวช

รัชนีวรรณ บุญชัยสุข

ปก
W8325
2550

เลขหมู่.....
เลขทะเบียน 83030
วัน,เดือน,ปี 30 ก.ค. 2551

b 11959708
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

# ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

นายพุทธพงษ์ เชี่ยวเวช 47010536
นางสาวรัชนีวรรณ บุญชัยสุข 47010620
ผศ.ดร.วิศิษฏ์ หิรัญกิตติ อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

## บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาเทคโนโลยีระบุเอกลักษณ์ด้วยความถี่วิทยุ (RFID) แล้วนำมาประยุกต์ใช้ในระบบขนส่งสิ่งของ และศึกษาการพัฒนาโปรแกรมด้วยภาษาไอรอนไพธอน (IronPython) เพื่อให้ได้ใช้ประโยชน์จากการใช้รูปแบบการเขียนโปรแกรมด้วยภาษาไพธอน (Python) ในขณะเดียวกันก็ได้ใช้ประโยชน์จากไลบรารี่ของดอทเน็ต (.NET) เพื่อติดต่อกับอุปกรณ์อาร์เอฟไอดี พร้อมทั้งทรัพยากรอื่นของดอทเน็ตด้วย การศึกษานี้ได้พัฒนาโปรแกรมจำลองการทำงานของระบบขนส่งให้อ่านแท็กอาร์เอฟไอดีที่ใช้ระบุสิ่งของ เก็บลงฐานข้อมูล ใช้อัลกอริทึมเอสตาร์ (A*) ในการค้นหาเส้นทางที่สั้นที่สุดในการขนส่ง ติดตามการขนส่งตั้งแต่สิ่งของเข้ามาในระบบจากผู้ส่งจนถึงปลายทางผู้รับผ่านเว็บแอพพลิเคชั่น บันทึกการเปลี่ยนแปลงตลอดกระบวนการขนส่ง เพื่อให้สามารถนำข้อมูลต่างๆมาวิเคราะห์เพื่อการวางแผนการส่งสิ่งของให้มีประสิทธิภาพ

**Intelligent E-Logistics with RFID**

Mr. Puttapong Chiewvech 47010536

Ms. Ratchaneewan Boonchaisuk 47010620

Asst.Prof.Dr. Visit Hirankitti Advisor

Academic Year 2007

## ABSTRACT

In this thesis we study the Radio-Frequency Identification (RFID) technology with an aim to apply the advantage of Auto-ID for logistics and also study IronPython to take the advantage of Python programming and .NET library to interface with Intermec RFID equipment, besides other .NET environment of the .NET Framework. We design the system that capable of tracking movement of items being delivered via web application starting from it entering the logistic process until it is delivered and received by a recipient, we develop this application in IronPython and use A* algorithm to find out the shortest path for transportation. The system collects the shipping data into database and records all changes that occur throughout the process so the user can analyze those data to improve the delivery planning.

# กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยความเมตตากรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.วิศิษฏ์ หิรัญกิตติ ที่เสียสละ เอื้อเฟื้อ ให้ความช่วยเหลือ สนับสนุน ทั้งด้านความรู้ อุปกรณ์การทดลอง ตลอดจนคอยให้คำชี้แนะแนวทางการแก้ปัญหาต่างๆ ให้ผู้ทำโครงงานได้ฝึกใช้ความคิด ความรู้ ความสามารถที่เรียนมา จนเกิดประสบการณ์ที่ดีในการดำรงชีวิตต่อไปได้

ขอขอบพระคุณ อาจารย์เกียรติณรงค์ ทองประเสริฐ อาจารย์ที่ปรึกษาร่วม อีกทั้งผู้ชื่นชอบการใช้ภาษาไพธอนและภาษาไอรอนไพธอนทั่วโลก ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้

ท้ายสุดนี้ คุณงามความดีและกุศลใดที่พึงบังเกิดมีจากปริญญานิพนธ์ฉบับนี้ เป็นผลมาจากความเมตตา กรุณา ของบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้ทำโครงงาน จึงขอยกคุณความดีเหล่านั้น เป็นเครื่องบูชาพระคุณ ด้วยความเคารพและสักการะยิ่ง

พุทธพงษ์ เชี่ยวเวช
รัชนีวรรณ บุญชัยสุข

# สารบัญ

หน้า


บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาของปัญหา.....1

1.2 วัตถุประสงค์ของการศึกษา.....1

1.3 ขอบเขตของงานวิจัย.....2

1.4 แผนการดำเนินงาน.....2

1.4.1 ขั้นศึกษาโครงการ.....2

1.4.2 ขั้นวางแผนออกแบบและดำเนินการ.....2

1.4.3 ขั้นทดลองหรือทดสอบประสิทธิภาพ.....2

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....4

2.1 โลจิสติกส์.....4

2.1.1 ความหมายของการจัดการโลจิสติกส์.....4

# สารบัญ (ต่อ)

หน้า

2.1.2 ประเภทของการจัดการโลจิสติกส์.....4

2.1.3 ความสำคัญของการจัดการโลจิสติกส์.....5

2.2 อาร์เอฟไอดี (RFID).....7

2.2.1 องค์ประกอบของระบบ RFID.....7

2.2.2 การทำงานของระบบ RFID.....15

2.2.3 ข้อดีข้อเสียของRFID.....17

2.2.4 มาตรฐานของ RFID.....19

2.3 ไอรอนไพธอน (IronPython).....22

2.3.1 เส้นทางการพัฒนาของภาษา IronPython.....23

2.3.2 ขอบเขตการใช้งาน.....23

2.3.3 วิธีการใช้งาน.....25

2.4 การเชื่อมต่อกับตัวอ่านอย่างง่าย.....27

2.4.1 คุณสมบัติของ BRI.....27

2.4.2 รูปแบบการเรียกใช้ BRI.....28

2.4.3 การพัฒนาโปรแกรมโดยใช้ BRI.....28

บทที่ 3 การออกแบบระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID.....33

3.1 จุดประสงค์การออกแบบ.....33

3.2 การออกแบบ.....33

# สารบัญ (ต่อ)

หน้า

3.2.1 การออกแบบระบบโลจิสติกส์.....33

3.2.2 การออกแบบคลาสไดอะแกรม.....42

บทที่ 4 การดำเนินงาน.....45

4.1 การติดต่ออุปกรณ์อาร์เอฟไอดี.....45

4.2 การจัดการฐานข้อมูล.....47

4.3 อัลกอริทึมในการค้นหาเส้นทางส่งสิ่งของ.....48

บทที่ 5 ผลการทำงานของระบบ.....50

5.1 แอพพลิเคชั่นในส่วนของสถานี.....50

5.2 แอพพลิเคชั่นในส่วนของแวร์เฮาส์.....56

5.3 แอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของ.....58

5.4 เว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ.....62

บทที่ 6 บทสรุปและวิจารณ์.....63

6.1 บทสรุปและวิจารณ์.....63

6.2 ปัญหาที่เกิดขึ้นในการพัฒนา.....64

6.3 แนวทางการพัฒนา.....64

บรรณานุกรม.....65

# สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงย่านความถี่ต่างๆของระบบ RFID และตัวอย่างการใช้งาน.....12

2.2 แสดงข้อเปรียบเทียบลักษณะของเทคโนโลยี RFID และ Barcode.....18

2.3 แสดงเงื่อนไขที่มีส่วนตัดสินใจว่า RFID จะมาแทนหรือมาเสริม Barcode.....19

2.4 แสดงการเปรียบเทียบมาตรฐานระหว่าง ISO/IEC กับ EPC.....22

2.5 แสดงคลาสในดอทเน็ตเฟรมเวิร์ก (Common .NET Assemblies and Namespaces).....24

2.6 แสดงการเขียนโค้ด Hello World ง่ายๆ เปรียบเทียบระหว่าง C# และ IronPython.....26

4.1 แสดงอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ.....45

# สารบัญรูป

รูปที่ หน้า

2.1 แสดงตัวอย่างแท็กชนิดแอ็กทีฟ.....8

2.2 แสดงตัวอย่างแท็กชนิดกึ่งพาสซีฟ.....9

2.3 แสดงตัวอย่างแท็กชนิดพาสซีฟ.....10

2.4 แสดงคลื่นความถี่ที่ใช้ในระบบ RFID.....11

2.5 แสดงตัวอย่างการใช้งานคลื่นความถี่ในย่านต่างๆ.....12

2.6 แสดงตัวอย่างเครื่องอ่านแบบติดตั้งยี่ห้อหนึ่ง.....14

2.7 แสดงตัวอย่างเครื่องอ่านแบบพกพายี่ห้อหนึ่ง.....14

2.8 แสดงการปล่อยคลื่นแม่เหล็กไฟฟ้าวิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า
Inductive Coupling) และวิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า
(Electromagnetic Propagation Coupling).....16

2.9 แสดงโครงสร้างของแท็กตามมาตรฐาน EPC-96.....21

2.10 แสดงการคอมไพล์โค้ดภาษาไพธอนภายใต้ IronPython Engine.....25

3.1 แสดงการทำงานของระบบ.....33

3.2 แสดง Use Case Diagram ส่วนควบคุมจัดการระบบขนส่ง.....35

3.3 แสดง Use Case Diagram ของการตรวจสอบและติดตามสิ่งของ.....37

3.4 แสดง Sequence diagram ของ Register delivered object.....38

# สารบัญรูป (ต่อ)

รูปที่ หน้า

3.5 แสดง Sequence diagram ของ Delivery.....39

3.6 แสดง Sequence diagram ของ Hand In.....40

3.7 แสดง Sequence diagram ของ Track the delivered object.....41

3.8 แสดง Class Diagram ของระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี.....42

3.9 แสดงตาราง Parcel.....43

3.10 แสดงตาราง Parcel Status.....43

3.11 แสดงตาราง Car.....44

3.12 แสดงตาราง Address.....44

3.13 แสดงตาราง Postcode.....44

4.1 แสดงการติดต่อระหว่าง Server และ Client (Handheld RFID Reader) ผ่าน wireless.....46

4.2 แสดงทดลองการอ่านแท็กด้วย Reader แบบ Handheld และส่งข้อมูลเข้าเครื่อง Server.....46

4.3 แสดงการทดลองอ่านแท็กด้วย Reader แบบ Fixed.....47

4.4 แสดงการใช้อัลกอริทึมเอสตาร์ในการค้นหาเส้นทางเมื่อกำหนดให้ผ่านหลายจุด.....49

5.1 แสดงฟังก์ชันการทำงานโดยรวมของระบบ.....50

5.2 แสดงแอพพลิเคชั่นในส่วนของสถานี : หน้าล็อกอินเข้าใช้งาน.....51

5.3 แสดงแอพพลิเคชั่นในส่วนของสถานี : เลือกการทำงาน.....51

5.4 แสดงแอพพลิเคชั่นในส่วนของสถานี : รับสิ่งของเข้าระบบ.....52

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

Intelligent E-Logistics with RFID

ผู้จัดทำ

1. นายพุทธพงษ์ เชี่ยวเวช รหัสประจำตัว 47010536
2. นางสาว รัชนีวรรณ บุญชัยสุข รหัสประจำตัว 47010620

_____ อาจารย์ที่ปรึกษา

( ผศ.ดร.วิศิษฏ์ หิรัญกิตติ )

# สารบัญรูป (ต่อ)

รูปที่ หน้า

5.5 แสดงแอพพลิเคชั่นในส่วนของสถานี: สถานะของสิ่งของที่ได้รับภายในสถานี.....53

5.6 แสดงแอพพลิเคชั่นในส่วนของสถานี : แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ.....55

5.7 แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : รับสิ่งของเข้าแวร์เฮ้าส์.....56

5.8 แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : จัดสิ่งของขึ้นรถ.....57

5.9 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Get in .....58

5.10 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Route .....59

5.11 แสดงแผนที่ที่รถจัดส่งสิ่งของต้องวิ่งผ่าน.....60

5.12 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Send .....61

5.13 แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำสถานีย่อย.....61

5.14 แสดงเว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ.....62

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาของปัญหา

การขนส่งวัตถุดิบและสิ่งของ ที่ รวดเร็ว ประหยัด และมีประสิทธิภาพ นับเป็นยุทธศาสตร์สำคัญประการหนึ่งในการพัฒนาอุตสาหกรรมและเศรษฐกิจของชาติ โดยเฉพาะอย่างยิ่งค่าใช้จ่ายในการขนส่งถือว่าเป็นต้นทุนสำคัญในการดำเนินธุรกิจแทบทุกประเภท ซึ่งการขนส่งมีทั้งรูปแบบตั้งแต่ การขนส่งวัตถุดิบ การขนส่งสิ่งของผลิตภัณฑ์ การขนส่งเอกสาร ฯลฯ แต่เมื่อพิจารณาให้ดีแล้วจะพบว่า ในรูปแบบต่างๆ เหล่านี้มีลักษณะที่คล้ายคลึงกันโดยส่วนใหญ่คือ เป็นการขนส่งสิ่งๆ หนึ่งจากจุดเริ่มต้นไปยังจุดหมาย ดังนั้นระบบขนส่งที่ดีควรสามารถรองรับการขนส่งในรูปแบบต่างๆได้โดยการปรับเปลี่ยนน้อยที่สุด นอกจากนี้การบริหารจัดการวางแผนการขนส่งที่มีประสิทธิภาพจะช่วยให้ประหยัดทรัพยากร ทั้งเวลา เชื้อเพลิง แรงงาน และท้ายที่สุดคือ ลดต้นทุนที่สำคัญในการดำเนินธุรกิจ นำไปสู่ความสามารถในการแข่งขันในตลาดการค้าเสรีในปัจจุบันและอนาคต

การพัฒนาระบบขนส่งโดยการนำเทคโนโลยีอาร์เอฟไอดีที่ใช้ระบุเอกลักษณ์มาใช้งานในการระบุถึงตัวสินค้า เพื่อมาแทนที่การใช้งานบาร์โค้ด ซึ่งจะช่วยลดเวลาในการทำงานของระบบให้น้อยลงและระบบมีความน่าเชื่อถือมากขึ้น โดยการพัฒนาระบบต้องทำให้ระบบมีความสามารถในการวางแผนการขนส่งได้อย่างมีประสิทธิภาพ สามารถค้นหาเส้นทางที่มีความเหมาะสมที่สุดในการขนส่งสินค้าได้

## 1.2 วัตถุประสงค์ของงานวิจัย

1. ศึกษาเทคโนโลยีระบุเอกลักษณ์ด้วยความถี่วิทยุ (RFID) เพื่อนำมาพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาด
2. ศึกษาภาษาไอรอนไพธอนเพื่อใช้ประโยชน์ของภาษาไพธอนและใช้ไลบรารี่ (Library) ของดอทเน็ต (.NET)
3. พัฒนาแอพพลิเคชั่นระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ อาร์เอฟไอดีด้วยภาษาไอรอนไพธอน

## 1.3 ขอบเขตของงานวิจัย

1 ศึกษาการทำงานของเทคโนโลยีอาร์เอฟไอดีและพัฒนาแอพพลิเคชั่นในการติดต่ออุปกรณ์อาร์เอฟไอดี

2 พัฒนาแอพพลิเคชั่นจำลองระบบขนส่งสินค้าโดยใช้เทคโนโลยีอาร์เอฟไอดีเพื่อระบุถึงตัวสินค้า

3 พัฒนาแอพพลิเคชั่นจำลองระบบขนส่งสินค้าโดยมีความสามารถในการค้นหาเส้นทางการขนส่งสินค้าที่มีความเหมาะสม ผู้ใช้งานสามารถตรวจสอบสถานะการขนส่งและสถานที่ที่สินค้าอยู่ ณ เวลาปัจจุบัน

## 1.4 แผนการดำเนินงาน

### 1.4.1 ขั้นศึกษาโครงการ

1 ศึกษาเทคโนโลยี อาร์เอฟไอดี และข้อมูลพื้นฐานในการใช้งานอุปกรณ์อาร์เอฟไอดี

2 ศึกษาการเขียนการใช้งานภาษาไอรอนไพธอน

3 ศึกษามิดเดิลแวร์ (Middleware) หรือเอพีไอ (API) ที่เหมาะสมกับอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

4 ศึกษาข้อมูลเกี่ยวกับระบบขนส่งสิ่งของ

### 1.4.2 ขั้นวางแผนออกแบบและดำเนินงาน

1 ออกแบบระบบฐานข้อมูลที่จะใช้ในระบบ

2 เขียนแอพพลิเคชั่นพัฒนาส่วนสื่อสารข้อมูลติดต่อฐานข้อมูลระหว่างเครื่องเซิร์ฟเวอร์ (Server) และไคลเอนต์ (Client)

3 เขียนแอพพลิเคชั่นพัฒนาในส่วนการติดต่ออุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

4 เขียนแอพพลิเคชั่นพัฒนาในส่วนควบคุมและจัดการระบบขนส่งที่เกี่ยวข้อง

5 ทำ Prototype ของระบบ

### 1.4.3 ขั้นทดสอบการทำงานของระบบ

1 ทดสอบการสื่อสารข้อมูลระหว่างเครื่องเซิร์ฟเวอร์ (Server) และไคลเอนต์ (Client)

2 ทดสอบการทำงานของแอพพลิเคชั่นในส่วนของการติดต่ออุปกรณ์อาร์เอฟไอดี

3 ทดสอบการทำงานของแอพพลิเคชั่นในส่วนของควบคุมและจัดการระบบขนส่ง

4 ค้นหาและปรับปรุงข้อผิดพลาดเพื่อพัฒนาประสิทธิภาพของระบบ

5 รายงานสรุปผลโครงการพร้อมทั้งจัดทำเอกสารประกอบ

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1 ได้ความรู้ทางด้านระบบโลจิสติกส์และเทคโนโลยีอาร์เอฟไอดี

2 ได้เรียนรู้การใช้ภาษาไอรอนไพธอนในการพัฒนาโปรแกรม โดยใช้รูปแบบการเขียนแบบภาษาไพธอน อีกทั้งยังสามารถเรียกใช้ไลบรารี่หรือทรัพยากรอื่นๆของดอทเน็ต (.NET framework) ได้

3 สามารถออกแบบพัฒนาระบบขนส่งชาญฉลาด ที่ช่วยลดต้นทุนและใช้ทรัพยากรได้อย่างมีประสิทธิภาพ

4 ได้เรียนรู้วิธีการทำงาน พร้อมทั้งวิธีการแก้ปัญหาที่เกิดขึ้น เพื่อสามารถนำไปประยุกต์ใช้ในชีวิตจริงได้

# บทที่ 2
# ทฤษฎีที่เกี่ยวข้อง

## 2.1 โลจิสติกส์

### 2.1.1 ความหมายของการจัดการโลจิสติกส์

คำว่า โลจิสติกส์ (Logistics) แปลว่า การส่งกำลังบำรุง ซึ่งความหมายในยุคศตวรรษที่ 19 ในเรื่องที่เกี่ยวกับการขนส่งสิ่งของ หมายถึง การวางแผนและบริหารจัดการเพื่อลำเลียงสิ่งของจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยเฉพาะในทางทหารและอุตสาหกรรมการผลิตสิ่งของ แต่ถ้าในเรื่องของการบริหารองค์กรที่มีสายงานมากและซับซ้อนแล้วคำว่าโลจิสติกส์จะหมายถึง การวางแผนและบริหารจัดการภารกิจที่มีความยุ่งยากซับซ้อน

เมื่อยุคสมัยเปลี่ยนไป ความหมายของ โลจิสติกส์ ก็เปลี่ยนไปตามวิธีการดำเนินธุรกิจของโลก เรียกว่า การจัดการโลจิสติกส์ธุรกิจ มีผู้ให้คำนิยามไว้มากมาย ดังเช่นตัวอย่างจาก Oak Brook, IL: Council of Logistic Management, 1993 นักวิชาการชาวออสเตรเลียกล่าวไว้ว่า

"การจัดการโลจิสติกส์ธุรกิจ คือ กระบวนการวางแผน การปฏิบัติการและควบคุม การเคลื่อนย้าย และการจัดเก็บสิ่งของอย่างมีประสิทธิภาพและประสิทธิผล รวมถึงการให้บริการและสารสนเทศที่เกี่ยวข้องตั้งแต่จุดกำเนิดจนถึงจุดการบริโภคสิ่งของ เพื่อวัตถุประสงค์ในการตอบสนองความต้องการของลูกค้า"

"Business Logistics Management is the Process of Planning, Implementing and Controlling the efficient, Effective flow and storage of Storage of Goods, Services and Related Information from Point of Origin to Point of Conforming to Customer Requirements."

### 2.1.2 ประเภทของการจัดการโลจิสติกส์

การจัดการโลจิสติกส์สามารถแบ่งเป็น 3 ประเภทใหญ่ ๆ ตามลักษณะของบทบาทหน้าที่มหภาคของประเทศได้ดังนี้

**2.1.2.1 การจัดการโลจิสติกส์ด้านการทหาร (Military Logistics Management)**หมายถึง การจัดการการจัดส่งกำลังบำรุงด้านการทหาร เช่น ยุทโธปกรณ์ ปัจจัยสี่ การรักษาพยาบาลและสารสนเทศ เพื่อมุ่งหวังชัยชนะทางทหารเป็นสำคัญ

**2.1.2.2 การจัดการลอจิสติกส์ด้านวิศวกรรม (Engineering Logistics Management)** หมายถึง การจัดการด้านการวิศวกรรมจัดส่งลำเรียง อันได้แก่การสร้าง การบูรณาการและการบำรุงรักษาสาธารณูปการ ทั้งระบบขนส่งทางบก ทางน้ำ ทางอากาศ และทางท่อ ระบบการจัดเก็บและระบบสารสนเทศ เพื่อมุ่งหวังความพร้อมในระบบการจัดเก็บและการจัดส่งลำเรียงทั้งระบบเป็นสำคัญ

**2.1.2.3 การจัดการโลจิสติกด้านธุรกิจ (Business Logistics Management)**หมายถึง การจัดการด้านการจัดเก็บและจัดส่งสิ่งของ คน สัตว์ สิ่งของ จากจุดหนึ่งไปยังจุดหนึ่งตามที่มนุษย์ต้องการ เพื่อมุ่งหวังความสำเร็จทางธุรกิจเป็นสำคัญ

**2.1.3 ความสำคัญของการจัดการโลจิสติกส์**

ปัจจุบัน การจัดการโลจิสติกส์ ต้องคำนึงถึง ต้นทุนด้านการขนส่งของประเทศ การผลิตสิ่งของหรือการบริการต่างๆ ย่อมต้องมีการติดต่อ ขนส่ง เช่น ขนส่งวัตถุดิบจากแหล่งวัตถุดิบ ไปยังโรงงานผ่านกระบวนการผลิตจนเป็นสิ่งของ จากนั้นต้องมีการขนส่งสิ่งของสู่ตลาด เพื่อกระจายให้ถึงผู้บริโภค ต้นทุนด้านการขนส่งมิได้หมายถึงเฉพาะค่าใช้จ่ายของยานพาหนะ แต่รวมถึงวิธีการบรรจุหีบห่อ ขนถ่าย และป้อนเข้าโรงงาน หากทำได้รวดเร็ว ประหยัด มีการสูญเสียน้อย นั่นย่อมหมายถึงมีต้นทุนต่ำ ในการกระจายผลผลิตสู่ตลาดและผู้บริโภคก็ต้องมีต้นทุนต่ำด้วย

ดังนั้น หน่วยงานที่เกี่ยวข้องกับการจัดการ โลจิสติกส์ จึงครอบคลุมหลายฝ่าย ตั้งแต่แหล่งวัตถุดิบ วิธีการบรรจุ ขนถ่าย กระบวนการส่ง-รับของ ผู้จัดสร้างโครงสร้างพื้นฐานด้านการขนส่งต่างๆ ทั้งระบบถนน ระบบราง ท่าเรือ ท่าอากาศยาน ศุลกากร และโรงเก็บสิ่งของ เป็นต้น ต้นทุนด้านการขนส่งจะต่ำได้ต่อเมื่อการขนถ่ายและนำส่งผลิตภัณฑ์ถึงจุดหมายโดยเร็ว สูญเสียน้อย สิ่งของถึงมือผู้รับตามเวลาโดยเร็ว ขั้นตอนกระบวนการศุลกากรทั้งนำเข้า-ส่งออก สะดวกรวดเร็ว ไม่ต้องเสียค่าเช่าโรงเก็บสิ่งของหรือตู้คอนเทนเนอร์ นานวัน ลดดอกเบี้ยของต้นทุนลงได้ด้วย

การติดต่อสื่อสาร เพื่อการสั่งซื้อทั้งวัตถุดิบและสิ่งของ ต้องสะดวกรวดเร็วและชัดเจน ระบบการถ่ายทอดส่งข้อมูลสารสนเทศ หรือ ไอที ต้องมีเครือข่ายที่เชื่อมต่อทั้งในประเทศและกับต่างประเทศอย่างทั่วถึง และเชื่อมโยงกับระบบอื่น ๆ ตั้งแต่แหล่งวัสดุ โรงงานผลิต กรมศุลกากรจนถึงผู้ซื้อ และมีมาตรฐานที่เป็นสากล เพื่อมิให้มีการผิดพลาดเกิดขึ้น ไม่สูญเสีย สามารถประหยัดต้นทุนสิ่งของได้ ประเทศต่าง ๆ ในโลกกำลังรณรงค์ด้านโลจิสติกส์ เพื่อลดต้นทุนผลผลิตของประเทศโดยเฉพาะด้านการขนส่ง

เมื่อมองภาพรวมของประเทศ คงต้องแบ่งเป็นสองส่วนด้วยกันคือในส่วนของภาครัฐที่จะต้องสนับสนุนระบบโลจิสติกส์ให้เชื่อมประสานกับระบบโลจิสติกส์ภายในของผู้ประกอบการ เพื่อการ

ประหยัดพลังงาน และลดต้นทุนในสิ่งของ สำหรับการแข่งขันในระดับนานาชาติ ในภาครัฐได้กำหนดแผนบริหารราชการแผ่นดิน (พ.ศ. 2548-2551) นโยบายปรับโครงสร้างเศรษฐกิจให้สมดุลและแข่งขันได้ โดยมีกรอบแผนแม่บทการพัฒนาระบบโลจิสติกส์ (Logistics Master Plan) ใน 6 ประเด็นหลักดังนี้

1. โครงสร้างพื้นฐาน โดยมีเป้าประสงค์
    - เพื่อลดค่าใช้จ่ายโดยรวมของการขนส่งสิ่งของและบริการ
    - ลดระยะเวลาที่ใช้ในการขนส่งสิ่งของและบริการจากต้นทางถึงปลายทาง
    - ยกระดับคุณภาพการให้บริการของท่าเรือหลัก
2. ระบบการเชื่อมโยงข้อมูลและฐานข้อมูล โดยมีเป้าประสงค์
    - ระบบข้อมูลเพื่อการนำเข้าและส่งออกของภาครัฐเชื่อมโยงเป็นระบบเดียวกัน และสามารถเชื่อมต่อกับเอกชน และผู้ใช้บริการ
    - การวางแผนและลงทุนเพื่อพัฒนาโลจิสติกส์ มีความสอดคล้องกับข้อเท็จจริงของประเทศมากยิ่งขึ้น
3. ความสามารถในการบริหารจัดการโลจิสติกส์ (Capacity Building) โดยมีเป้าประสงค์
    - ธุรกิจสามารถประยุกต์ใช้การบริหารจัดการโลจิสติกส์สมัยใหม่ได้อย่างมีประสิทธิ ภาพ
4. ธุรกิจให้บริการโลจิสติกส์ (Logistic Service Providers) โดยมีเป้าประสงค์
    - มีผู้ให้บริการโลจิสติกส์ที่มีคุณภาพ และเป็นที่ยอมรับของผู้ใช้บริการในระดับสากล
5. การเชื่อมโยงทางการค้ากับภูมิภาค โดยมีเป้าประสงค์
    - การเติบโตของการค้าระหว่างไทยกับประเทศในภูมิภาคอาเซียน (ASEAN)
6. กฎหมาย ข้อบังคับ นโยบาย และกระบวนการให้บริการของรัฐ โดยมีเป้าประสงค์
    - มีกฎหมาย ข้อบังคับ นโยบาย ที่สนับสนุนกิจกรรมโลจิสติกส์
    - บริการของหน่วยงานรัฐทางด้านโลจิสติกส์มีมาตรฐานเทียบเท่าสากล

ระบบ RFID ก็เป็นอีกเทคโนโลยีที่เข้ามามีบทบาทสำคัญในการพัฒนาระบบ Logistics ได้เป็นอย่างดี มีข้อดีหลายอย่างเมื่อเทียบกับระบบเก่าๆที่เคยใช้

## 2.2 อาร์เอฟไอดี (RFID)

อาร์เอฟไอดี(RFID) ย่อมาจากคำว่า Radio Frequency Identification เป็นระบบฉลากที่ได้ถูกพัฒนามาตั้งแต่ยุค ค.ศ. 1970s โดยที่อุปกรณ์อาร์เอฟไอดี ที่มีการประดิษฐ์ขึ้นใช้งานเป็นครั้งแรกนั้นเป็นผลงานของ León Theremin ซึ่งสร้างให้กับรัฐบาลของประเทศรัสเซียในปี ค.ศ. 1946 ซึ่งอุปกรณ์ที่สร้างขึ้นมาในเวลานั้นทำหน้าที่เป็นเครื่องมือดักจับสัญญาณ ไม่ได้ทำหน้าที่เป็นตัวระบุเอกลักษณ์อย่างที่ใช้งานกันอยู่ในปัจจุบัน

อาร์เอฟไอดีในปัจจุบันเป็นเทคโนโลยีการระบุข้อมูลที่แสดงเอกลักษณ์ของวัตถุหรือบุคคลด้วยคลื่นความถี่วิทยุที่ได้ถูกพัฒนามาในยุค ค.ศ. 1970s วัตถุประสงค์เพื่อนำไปใช้ในการบ่งชี้วัตถุในระยะไกลได้ โดยมีจุดเด่น คือ สามารถอ่านข้อมูลจากแท็ก (Tag) ได้ครั้งละหลายๆแท็ก โดยไม่จำเป็นต้องอาศัยการสัมผัส (Contact-Less) หรือต้องเห็นวัตถุนั้นๆ ก่อน และสามารถอ่านค่าได้แม้ในสภาพที่ทัศนวิสัยไม่ดี ทนต่อความเปียกชื้น แรงสั่นสะเทือน การกระทบกระแทก และสามารถจะอ่านข้อมูลได้ด้วยความเร็วสูง โดยข้อมูลจะถูกเก็บไว้ในไมโครชิปที่อยู่ในแท็ก ทำงานโดยใช้เครื่องอ่านที่สื่อสารกับแท็กด้วยคลื่นวิทยุในการอ่านและเขียนข้อมูล

### 2.2.1 องค์ประกอบของระบบอาร์เอฟไอดี

ระบบอาร์เอฟไอดีประกอบด้วย 3 ส่วนหลัก

**2.2.1.1 แท็ก (Tag) / ทรานสปอนเดอร์ (Transponder)** ซึ่งมาจากคำว่าทรานสมิตเตอร์ (Transmitter) ผสมกับคำว่าเรสปอนเดอร์ (Responder) ถ้าจะแปลให้ตรงตามศัพท์ แท็กก็จะทำหน้าที่ส่งสัญญาณหรือข้อมูลที่บันทึกอยู่ในแท็กตอบสนองไปที่ตัวอ่านข้อมูล การสื่อสารระหว่างแท็กและตัวอ่านข้อมูลจะเป็นการสื่อสารกันโดยอาศัยช่องความถี่วิทยุผ่านอากาศ

**1. องค์ประกอบของแท็ก**

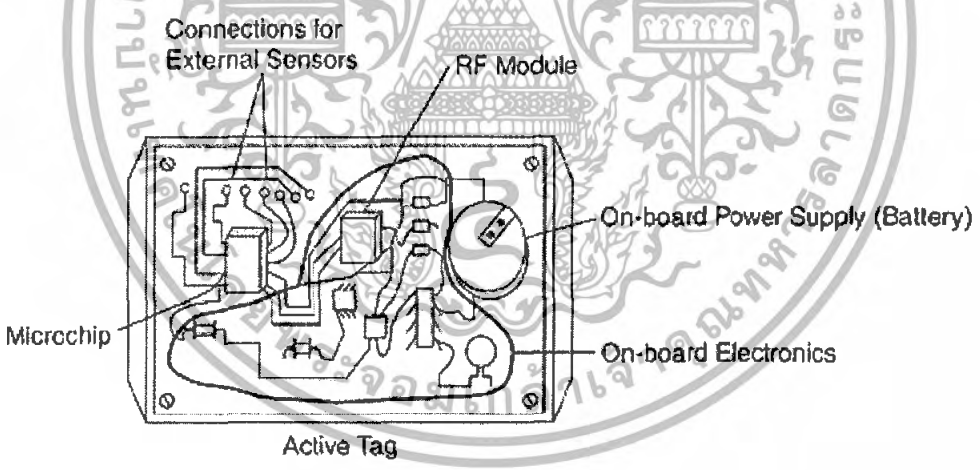
- ส่วนของไอซีซึ่งเป็นชิปสารกึ่งตัวนำ (**Semiconductor Chip**) หรือ ไมโครชิป ทำหน้าที่เก็บข้อมูลของวัตถุ
- ส่วนของขดลวดซึ่งทำหน้าที่เป็นเสาอากาศ **(Antenna)** สำหรับรับส่งสัญญาณคลื่นความถี่วิทยุและสร้างพลังงานป้อนให้ส่วนของไมโครชิป

ทั้งสองส่วนนี้จะเชื่อมต่ออยู่ด้วยกัน โดยทั่วไปตัวแท็กอาจอยู่ในรูปแบบที่เป็นกระดาษ แผ่นฟิล์ม พลาสติก มีขนาดและรูปร่างต่าง ๆ กันไป และมีหลายรูปแบบ ทั้งนี้ขึ้นอยู่กับการนำไปใช้งาน

**2. การแบ่งประเภทของแท็กตามองค์ประกอบ** ทั้งนี้เราสามารถแบ่งแท็กที่มีใช้งานกันอยู่ได้เป็น 3 ชนิดใหญ่ ๆ ได้แก่

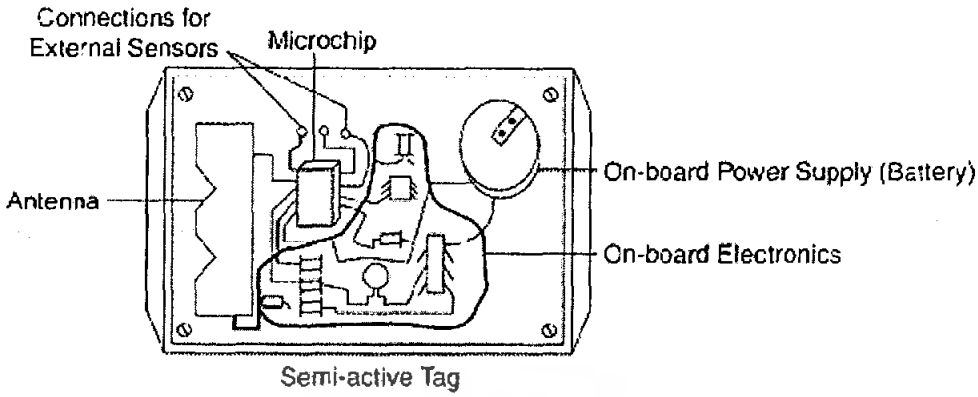
- **แท็กชนิดแอ็กทีฟ (Active Tag)** แท็กชนิดนี้จะมีแบตเตอรี่อยู่ภายในซึ่งใช้เป็นแหล่งจ่ายไฟขนาดเล็ก เพื่อป้อนพลังงานไฟฟ้าให้แท็กทำงานโดยปกติ ส่งสัญญาณแบบบรอดแคส(broadcast) ไปที่เครื่องอ่าน ได้โดยแท็กชนิดนี้มีฟังก์ชั่นการทำงานทั่วไปทั้งอ่านและเขียนข้อมูลลงในแท็กได้ และการที่ต้องใช้แบตเตอรี่จึงทำให้แท็กชนิดแอ็กทีฟมีอายุการใช้งานจำกัดตามอายุของแบตเตอรี่ โดยทั่วไปประมาณ 2-10 ปี เมื่อแบตเตอรี่หมดก็ต้องนำแท็กไปทิ้ง ไม่สามารถนำกลับมาใช้ใหม่ได้ เนื่องจากจะมีการซีล (seal) ที่ตัวแท็กจึงไม่สามารถเปลี่ยนแบตเตอรี่ได้ อย่างไรก็ตามถ้าสามารถออกแบบวงจรของแท็กให้กินกระแสไฟน้อยๆ ก็อาจจะมีอายุการใช้งานนานนับสิบปี

  แท็กชนิดแอ็กทีฟนี้จะมีหน่วยความจำภายในขนาดใหญ่ได้ถึง 1 เมกะไบต์ มีกำลังส่งสูงและระยะการรับส่งข้อมูลไกลสูงเป็น 100 เมตร ซึ่งไกลกว่าแท็กชนิดพาสซีฟ นอกจากนี้ยังทำงานในบริเวณที่มีสัญญาณรบกวนได้ดี แม้แท็กชนิดนี้จะมีข้อดีอยู่หลายข้อแต่ก็มีข้อเสียอยู่ด้วยเหมือนกัน เช่น ราคาต่อหน่วยแพง มีขนาดค่อนข้างใหญ่ และมีระยะเวลาในการทำงานที่จำกัด



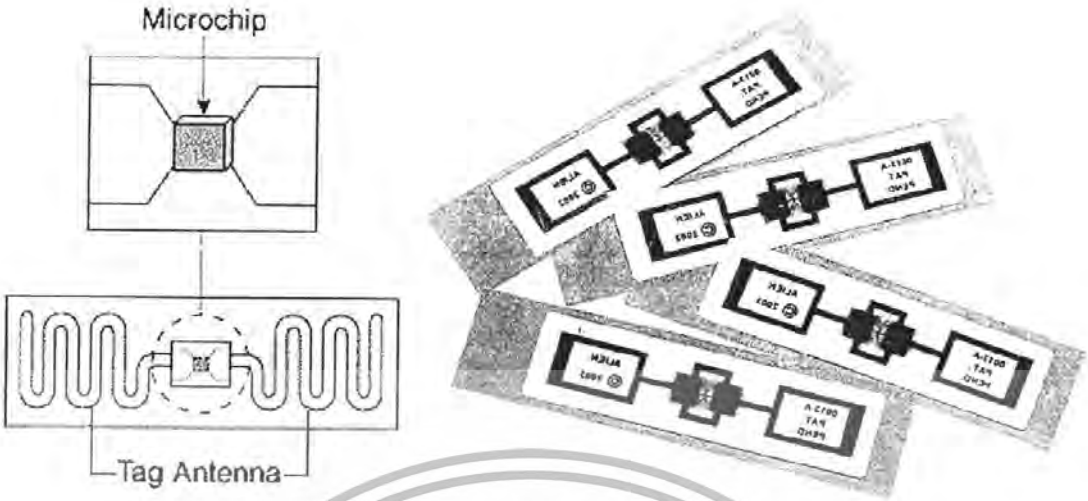

**รูปที่ 2.1** แสดงตัวอย่างแท็กชนิดแอ็กทีฟ

- **แท็กชนิดกึ่งพาสซีฟ (Semi-passive Tag)** แท็กชนิดนี้เหมือนแท็กชนิดแอ็กทีฟตรงที่มีแบตเตอรี่จ่ายไฟให้ไมโครชิป แต่ไม่ได้ส่งสัญญาณแบบบรอดแคสไปที่เครื่องอ่าน ตัวแท็กจะรอรับสัญญาณกระตุ้นให้ทำงานจากเครื่องอ่านแล้วจึงจะส่งข้อมูลกลับไปยังเครื่องอ่านเหมือนแท็กชนิดพาสซีฟ แต่ส่งได้ไกลกว่า มีความแม่นยำกว่าแท็กชนิดพาสซีฟ อีกทั้งยังมีอายุการใช้งานนานกว่าแท็กชนิดแอ็กทีฟด้วย




**รูปที่ 2.2** แสดงตัวอย่างแท็กชนิดกึ่งพาสซีฟ

- **แท็กชนิดพาสซีฟ(Passive Tag)** จะไม่มีแบตเตอรี่อยู่ภายในหรือไม่จำเป็นต้องรับแหล่งจ่ายไฟใด ๆ เพราะจะทำงานโดยอาศัยพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้าจากตัวอ่านข้อมูล (มีวงจรกำเนิดไฟฟ้าขนาดเล็กอยู่ในตัว)หรือที่เรียกว่าอุปกรณ์ Transceiver จึงทำให้แท็กชนิดพาสซีฟมีน้ำหนักเบาและเล็กกว่าแท็กชนิดแอ็กทีฟ ราคาถูกกว่า และมีอายุการใช้งานไม่จำกัด แต่ข้อเสียก็คือระยะการรับส่งข้อมูลใกล้ซึ่งสามารถส่งข้อมูลได้ไกลสุดเพียง 1.5 เมตร ซึ่งเป็นระยะการอ่านที่สั้น มีหน่วยความจำขนาดเล็กซึ่งโดยทั่วไปประมาณ 32 ถึง 128 บิต และตัวเครื่องอ่านข้อมูลจะต้องมีความไวและกำลังที่สูง นอกจากนี้แท็กชนิดพาสซีฟมักจะมีปัญหาเมื่อนำไปใช้งานในสิ่งแวดล้อมที่มีสัญญาณแม่เหล็กไฟฟ้ารบกวนสูงอีกด้วย แต่ข้อได้เปรียบในเรื่องราคาต่อหน่วยที่ต่ำกว่าแท็กชนิดแอ็กทีฟและอายุการใช้งานที่ยาวนานกว่าทำให้แท็กชนิดพาสซีฟนี้เป็นที่นิยมมากกว่า ไอซีของแท็กชนิดพาสซีฟที่มีการผลิตออกมาจะมีทั้งขนาดและรูปร่างเป็นได้ตั้งแต่แท่งหรือแผ่นขนาดเล็กจนแทบไม่สามารถมองเห็นได้ ไปจนถึงขนาดใหญ่จนสะดุดตา ซึ่งต่างก็มีความเหมาะสมกับชนิดงานที่แตกต่างกัน




**รูปที่ 2.3** แสดงตัวอย่างแท็กชนิดพาสซีฟ

**3. การแบ่งประเภทของแท็กตามรูปแบบการอ่านหรือบันทึกข้อมูล**

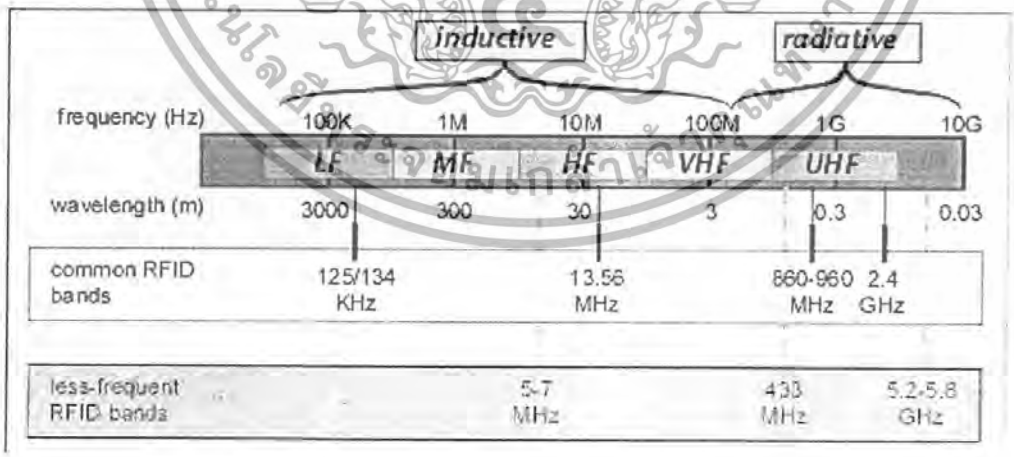
- แท็กชนิดที่สามารถถูกอ่านและเขียนข้อมูลได้หลายครั้ง (Read-Write)
- แท็กชนิดที่เขียนได้เพียงครั้งเดียวเท่านั้นแต่อ่านได้อย่างหลายครั้ง (Write-Once-Read-Many: WORM)
- แท็กชนิดที่อ่านได้เพียงอย่างเดียว (Read-Only)

**4. การแบ่งประเภทของแท็กตามความถี่ของการใช้งาน**

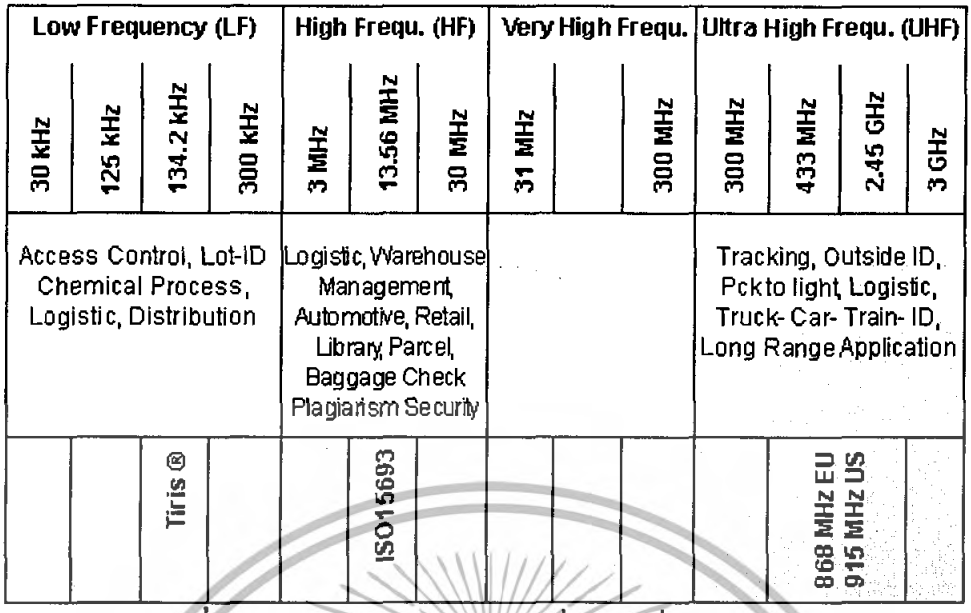
- **แท็กย่านความถี่ต่ำ (LF)** 125 - 134 กิโลเฮิรตซ์ย่านความถี่ต่ำคือมีค่า 30 – 300 KHz แต่ที่ระบบ RFID ใช้คือความถี่ในช่วง 125 – 134 KHz ซึ่งมีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านต่ำ โดยเฉพาะเมื่ออยู่ในสภาพแวดล้อมที่ไม่เหมาะสม เช่น บรรจุน้ำ โลหะ เป็นต้น แท็กที่ใช้ในย่านความถี่นี้มักเป็นแท็กชนิดพาสซีฟ แต่ก็มีใช้แท็กชนิดแอ็กทีฟบ้างขึ้นกับผู้ผลิต เพราะการใช้งานระบบอาร์เอฟไอดี ในย่านความถี่นี้มีมากมาย เป็นที่นิยมใช้กันกว้างขวาง และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก
- **แท็กย่านความถี่สูง (HF)** 13.56 เมกะเฮิรตซ์ ย่านความถี่สูงคือมีค่า 3 – 30 MHz แต่ที่ระบบอาร์เอฟไอดีใช้คือ 13.56 MHz แท็กที่ใช้ก็มักเป็นแท็กชนิดพาสซีฟเช่นกัน และแน่นอนอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านต่ำ แต่ก็เป็นที่นิยมในการใช้งานระบบอาร์เอฟไอดีในย่านความถี่สูงโดยเฉพาะในโรงพยาบาล ซึ่งไม่ต้องติดต่อกับอุปกรณ์ที่มีอยู่แล้วในระบบ และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก

ย่านความถี่ถัดไปจากนี้คือ ย่านความถี่สูงมาก (VHF) มีความถี่ในช่วง 30 – 300 MHz แต่ไม่มีระบบอาร์เอฟไอดีใดใช้งานในย่านความถี่นี้เลย

- **แท็กย่านความถี่สูงยิ่ง (UHF)** 400 - 1000 เมกะเฮิรตซ์ ย่านความถี่สูงยิ่งคือมีค่า 300 MHz – 1 GHz ระบบ อาร์เอฟไอดี ที่ใช้แท็กชนิดพาสซีฟในสหรัฐอเมริกาจะใช้ความถี่ 915 MHz ส่วนในยุโรปจะใช้ความถี่ 868 MHz แต่โดยทั่วไปจะใช้ในย่านความถี่ 315 – 433 MHz มีการใช้งานทั้งแท็กชนิดพาสซีฟและแอ็กทีฟ มีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านสูง ปัจจุบันเริ่มมีการใช้งานย่านความถี่นี้กันมากขึ้น แต่ย่านความถี่นี้ไม่เป็นที่ยอมรับทั่วโลก
- **แท็กย่านไมโครเวฟ** 2.4 กิกะเฮิรตซ์ย่านความถี่สูงยิ่งคือมีค่าตั้งแต่ 1 GHz ขึ้นไปซึ่งเป็นคลื่นไมโครเวฟ ระบบอาร์เอฟไอดีใช้งานที่ความถี่ 2.45 GHz หรือ 5.8 GHz แท็กที่ใช้ได้ก็มีทั้งแท็กชนิดกึ่งพาสซีฟและพาสซีฟ มีอัตราการรับส่งข้อมูลระหว่างแท็กและเครื่องอ่านสูงมาก แต่เนื่องจากความยาวคลื่นเป็นส่วนกลับของความถี่ แท็กชนิดพาสซีฟจะทำงานในช่วงย่านความถี่ไมโครเวฟที่ต่ำที่สุด อีกทั้งยังมีขนาดเล็กมาก เพราะไมโครชิปที่อยู่ข้างในแท็กผลิตมาเล็กมาก ความถี่ที่ใช้งานมากสุดคือ 2.4 GHz ทั้งด้านอุตสาหกรรม (Industry) วิทยาศาสตร์ (Scientific) และทางการแพทย์ (Medical) รวมเรียกว่า ISM และย่านความถี่นี้เป็นที่ยอมรับทั่วโลก




**รูปที่ 2.4** แสดงคลื่นความถี่ที่ใช้ในระบบ RFID




**รูปที่ 2.5** แสดงตัวอย่างการใช้งานคลื่นความถี่ในย่านต่างๆ

**ตารางที่ 2.1** แสดงย่านความถี่ต่างๆของระบบ RFID และตัวอย่างการใช้งาน

| ย่านความถี่ | | ระยะทาง | การใช้งาน |
|---|---|---|---|
| ย่านความถี่ต่ำ (Low Frequency :LF) | 125-134 kHz | 18 นิ้ว (~10 cm.) | ปศุสัตว์, ป้ายสินค้ากันขโมยที่อ่านในระยะใกล้, ระบบกันขโมยรถยนต์ |
| ย่านความถี่สูง (High Frequency :HF) | 13.553-13.567 MHz | 3 ฟุต (~1 m.) อ่านได้เร็ว 10-100 Tags/sec. | ห้องสมุด, สมาร์ทการ์ด, ระบบติดตามหนังสือ, ระบบปิดเปิดประตู |
| ย่านความถี่สูงยิ่ง (Ultra High Frequency :UF) | 400 – 1000 MHz (for US 433 MHz) | 10-30 ฟุต (Passive: 2-5 m.) (Active: 1-100 m.) อ่านได้เร็วมาก 100-1000 Tags/sec. | ตู้สิ่งของ, รถบรรทุก, แท่นยกสินค้า (Pallet) |
| ย่านความถี่ไมโครเวฟ (Microwave Frequency) | 2.4 GHz, 5.8 GHz | >30 ฟุต (Passive: < 1 m.) (Active: 1-15 m.) | อุปกรณ์ไร้สาย |

**2.2.1.2 ตัวอ่านข้อมูล (Reader / Interrogator)**

**1. องค์ประกอบของเครื่องอ่าน**

- **RF Interface Chip** ทำหน้าที่แปลงสัญญาณอนาลอคให้เป็นดิจิตอลในกรณีส่งข้อมูลจาก Coil ไปยัง MCU และแปลงสัญญาณดิจิตอลเป็นอนาลอค ในกรณีส่งข้อมูลจาก MCU ไปยัง Coil
- **Microcontroller** ทำหน้าที่ควบคุมการทำงานของเครื่องอ่าน เช่น ถอดรหัสสัญญาณตามรูปแบบที่เข้ารหัสมา เช่น Manchester, NRZ, Bi-phase เป็นต้น
- **Input / Output** ทำหน้าที่เป็นตัวรับข้อมูลจากภายนอก เช่น Keypad เป็นต้น และเป็นอุปกรณ์แสดงผลหรือควบคุมอุปกรณ์ที่อยู่ภายนอก เช่น LED, 7-Segment, LCD, Relay เป็นต้น
- **Graphic User Interface** ทำหน้าที่แสดงผลของข้อมูลให้อยู่ในรูปแบบที่ผู้ใช้งานสามารถเข้าใจได้ง่าย อีกทั้งยังใช้ควบคุมการทำงานของเครื่องอ่านได้อีกด้วย

**2. ประเภทของเครื่องอ่าน**

เครื่องอ่านแต่ละประเภทอาจมีชื่อเรียกแตกต่างกันไปตามลักษณะการใช้งาน หรือผู้ผลิตทางการค้าแต่ละราย แต่หลักๆ แบ่งได้ 2 ประเภท คือ

- **Stationary Reader** เครื่องอ่านแบบติดตั้ง หรือ Fixed Reader มักติดตั้งไว้ที่ผนัง กำแพง ประตูทางเข้า หรือที่ที่เหมาะสมกับระยะการอ่าน ซึ่งไม่จำเป็นต้องติดตั้งบนสิ่งที่เคลื่อนที่ไม่ได้เท่านั้น อาจจะติดตั้งบนผนังรถบรรทุกสิ่งของก็ได้ เป็นต้น โดยทั่วไปต้องอาศัยสายอากาศจากภายนอกในการอ่านข้อมูลจากแท็ก เครื่องอ่านชนิดนี้ไม่ได้ทนทานเหมือนแท็ก โดยเฉพาะอย่างหากใช้งานนอกสถานที่ต้องได้รับการดูแลรักษาเป็นอย่างดี ราคาก็ค่อนข้างถูกกว่าแบบพกพา จึงเป็นที่นิยมใช้ในปัจจุบัน โดยทั่วไปเครื่องอ่านชนิดนี้มี 2 โหมดการทำงานคือ
    - Autonomous Mode เครื่องอ่านจะอ่านแท็กอย่างต่อเนื่องในระยะการอ่านได้ของเครื่องอ่าน และบันทึกข้อมูลลงแท็กลิสต์ (tag list) ซึ่งมีเวลากำหนดว่าอยู่ในแท็กลิสต์นานแค่ไหน เรียกว่า persist time ซึ่งเครื่องควบคุมในระบบ หรือ host ที่จะอ่านข้อมูลในแท็กลิสต์สามารถอ่านได้เป็นระยะในช่วงเวลา persist time นี้เท่านั้น ไม่เช่นนั้นเครื่องอ่านจะลบข้อมูลนั้นทิ้งจากแท็กลิสต์

- Interactive Mode เครื่องอ่านจะถูกควบคุมโดยโปรแกรมที่รันบนระบบ หรือ host ที่ติดต่อกับเครื่องอ่านได้ ส่งคำสั่งให้เครื่องอ่านทำงาน ซึ่งเครื่องอ่านก็สามารถทำงานได้หลายคำสั่ง และส่งข้อมูลในแท็กลิสต์ให้ตัวควบคุมที่สั่งให้ทำงานมาได้

**รูปที่ 2.6** แสดงตัวอย่างเครื่องอ่านแบบติดตั้งยี่ห้อหนึ่ง

- **Handheld / Portable Reader** เครื่องอ่านแบบพกพา ก็ตามชื่อพกพาได้สะดวก เพราะโดยทั่วไปมีสายอากาศติดตั้งอยู่ภายในตัวเครื่อง ราคาแพงกว่าแบบติดตั้ง แต่ปัจจุบันก็เริ่มมีผู้ใช้มากขึ้น เพราะมีผู้ผลิตหลายรายแข่งขันกัน ทำให้ราคาถูกลง

**รูปที่ 2.7** แสดงตัวอย่างเครื่องอ่านแบบพกพายี่ห้อหนึ่ง

หน้าที่สำคัญของตัวอ่านข้อมูล (Reader หรือ Interrogator) ก็คือการรับข้อมูลที่ส่งมาจากแท็ก แล้วทำการตรวจสอบความผิดพลาดของข้อมูล ถอดรหัสสัญญาณข้อมูลที่ได้รับ ซึ่งกระทำโดย ไมโครคอนโทรเลอร์ อัลกอริทึมที่อยู่ในเฟิร์มแวร์ (Firmware) ของตัวไมโครคอนโทรเลอร์จะทำหน้าที่ในการส่งสัญญาณ ถอดรหัสสัญญาณที่ได้ และทำหน้าที่ติดต่อกับคอมพิวเตอร์เพื่อนำข้อมูลผ่านเข้าสู่กระบวนการต่อไป

นอกจากนี้ตัวอ่านข้อมูลที่ดีต้องมีความสามารถในการป้องกันการอ่านข้อมูลซ้ำ เช่น เมื่อมีแท็กหลายๆ อันเข้ามาอยู่ใกล้เครื่องอ่าน เมื่อแท็กมีพลังงานเพียงพอ แท็กแต่ละอันจะพยายามส่งข้อมูลของตัวเองมาที่เครื่องอ่านพร้อม ๆ กัน ทำให้เครื่องอ่านไม่สามารถแยกแยะข้อมูลที่ส่งมาได้ ซึ่งเราเรียกปรากฏการณ์นี้ว่า การชนกันของข้อมูล (Collision) จะไม่มีข้อมูลส่งถึงเครื่องอ่านเลย วิธีการแก้ไขโดยการทำการเพิ่มฟังก์ชั่นป้องกันการชนกันบนแท็กและเครื่องอ่าน (Anti-collision) ซึ่งจะมี หลายเทคนิค เช่นจัดคิวการอ่านแท็กโดยทำเป็นช่วงเวลาสั้นๆ เมื่อแท็กถูกอ่านแล้วจะไม่มีการอ่านซ้ำอีก

#### 2.2.1.3 คอนโทรลเลอร์(Controller / Host)

ระบบประยุกต์ใช้งาน รวมถึงระบบฮาร์ดแวร์และซอร์ฟแวรประยุกต์ใช้งาน หรือระบบฐานข้อมูล มักเป็นเครื่องคอมพิวเตอร์ที่ควบคุมการทำงานของระบบที่ใช้ RFID อาจจะเรียกว่า Middleware ก็ได้ ส่วนนี้ขึ้นกับระบบการใช้งานของระบบที่เกี่ยวข้อง

### 2.2.2 การทำงานของระบบอาร์เอฟไอดี

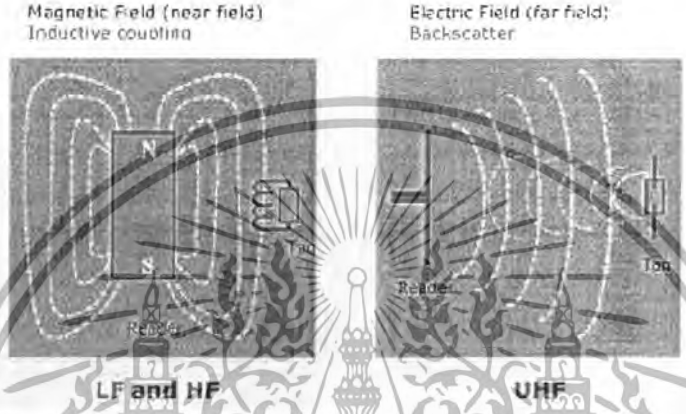
อาร์เอฟไอดีเป็นระบบที่นำเอาคลื่นวิทยุมาเป็นคลื่นพาหะเพื่อใช้ในการสื่อสารข้อมูลระหว่างอุปกรณ์สองชนิดที่เรียกว่า แท็กและตัวอ่านข้อมูลซึ่งเป็นการสื่อสารแบบไร้สายโดยต้องมีการเข้ารหัสข้อมูลดิจิตอลที่จะส่งผ่านให้อยู่ในรูปที่เหมาะสมสำหรับการส่งผ่านช่องสัญญาณซึ่งอาจมีสัญญาณรบกวนอีก จึงต้องเลือกวิธีการเข้ารหัสให้ข้อมูลมีค่าความผิดพลาดน้อยที่สุดเท่าที่เป็นไปได้ ซึ่งมีหลายวิธี เช่น NRZ, Manchester, Miller, Differential, DBP เป็นต้น แล้วต้องนำข้อมูลที่ต้องการส่งที่ผ่านการเข้ารหัสแล้วมาทำการมอดูเลต (modulation) กับคลื่นพาหะ เพื่อปรับเปลี่ยนค่าต่างๆ ของคลื่นพาหะให้ส่งออกไปได้เหมาะสมแล้วส่งออกผ่านทางสายอากาศที่อยู่ในตัวรับข้อมูล การมอดูเลตก็มีหลายวิธีอีก เช่น ASK (Amplitude Shift Keying), PSK (Phase Shift Keying), FSK (Frequency Shift Keying) เป็นต้น

**หลักการทำงานเบื้องต้นของระบบอาร์เอฟไอดี**

1. ตัวอ่านข้อมูลจะปล่อยคลื่นแม่เหล็กไฟฟ้าออกมาตลอดเวลา และคอยตรวจจับว่ามีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้าหรือไม่ หรืออีกนัยหนึ่งก็คือการคอยตรวจจับว่ามีการมอดูเลตสัญญาณเกิดขึ้นหรือไม่

การส่งคลื่นมี 2 วิธี คือ

- ในย่านความถี่ต่ำและสูง (LF และ HF) จะใช้วิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling หรือ Proximity Electromagnetic)
- ในระบบความถี่สูงยิ่งและไมโครเวฟ (UHF และ Microwave) จะใช้วิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Propagation Coupling)




**รูปที่ 2.8** แสดงการปล่อยคลื่นแม่เหล็กไฟฟ้าวิธีเหนี่ยวนำคลื่นแม่เหล็กไฟฟ้า (Inductive Coupling) และวิธีการแผ่กระจายคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Propagation Coupling)

2. เมื่อมีแท็กเข้ามาอยู่ในบริเวณสนามแม่เหล็กไฟฟ้า แท็กจะได้รับพลังงานไฟฟ้าที่เกิดจากการเหนี่ยวนำของคลื่นแม่เหล็กไฟฟ้าเพื่อให้แท็กเริ่มทำงาน และจะส่งข้อมูลในหน่วยความจำที่ผ่านการมอดูเลตกับคลื่นพาหะแล้วออกมาทางสายอากาศที่อยู่ภายในแท็ก

ถ้าเป็นแท็กย่านความถี่สูงยิ่ง (UHF) ก็จะส่งไปยังเครื่องอ่านแบบ backscattering

ถ้าเป็นแท็กชนิดแอ็กทีฟจะส่งสัญญาณแบบ Broadcast ออกมา แต่จะทำการส่งข้อมูลออกให้เครื่องอ่าน ก็ต่อเมื่อได้รับสัญญาณจากเครื่องอ่านและแบบเครื่องบอกตำแหน่ง หรือเบคอน (beacon) ซึ่งสัญญาณจะถูกปล่อยออกมาเป็นระยะ ๆ ตลอดเวลา

3. คลื่นพาหะที่ถูกส่งออกมาจากแท็กจะเกิดการเปลี่ยนแปลงแอมปลิจูด, ความถี่ หรือเฟส ขึ้นอยู่กับวิธีการมอดูเลต

4. ตัวอ่านข้อมูลจะตรวจจับความเปลี่ยนแปลงของคลื่นพาหะแปลงออกมาเป็นข้อมูล แล้วทำการถอดรหัสเพื่อนำข้อมูลไปใช้งานต่อไป

### 2.2.3 ข้อดีข้อเสียของอาร์เอฟไอดี

**ข้อดีของอาร์เอฟไอดี**

- ใช้บ่งชี้ลักษณะแสดงตัวเฉพาะได้(Individual Uniqueness)
- สามารถอ่านข้อมูลได้อัตโนมัติในรัศมีการรับส่งคลื่นของเครื่องอ่านโดยไม่ต้องเห็นแท็ก ไม่ต้องจัดแท็กให้หน้าตรงกับเครื่องอ่าน (Non-Line of Sight)
- สามารถอ่านข้อมูลได้ทีละหลายๆแท็ก(Simultaneous reading)
- สามารถอ่านข้อมูลได้รวดเร็ว และแม่นยำ ขจัดปัญหาการอ่านข้อมูลซ้ำ
- สามารถอ่านข้อมูลได้ระยะไกล ขึ้นกับแท็กและคลื่นที่ใช้แต่ละชนิด
- สามารถอ่านข้อมูลขณะวัตถุกำลังเคลื่อนไหวได้
- สามารถทำทั้งอ่านและเขียนข้อมูลลงแท็กได้ตลอดเวลา โดยไม่ต้องสัมผัส (Contactless)
- ความละเอียดและความจุของข้อมูลสูง
- ข้อมูลมีความปลอดภัยสูง ยากต่อการปลอมแปลงหรือลอกเลียนแบบ
- นำแท็กกลับมาใช้ใหม่ได้(Reusable)
- ทนทานต่อความเปียกชื้น แรงสั่นสะเทือน การกระทบกระแทก
- ลดความเสี่ยงต่อชีวิตคนในพื้นที่ที่อันตราย

**ข้อเสียของอาร์เอฟไอดี**

- ราคาลงทุนค่อนข้างสูง
- อาจถูกรบกวนจากคลื่นความถี่ต่ำที่ใช้งานได้ง่าย ต้องได้รับการควบคุมดูแลจากหน่วยงานที่เกี่ยวข้อง
- ข้อจำกัดกับผลิตภัณฑ์และสิ่งแวดล้อมที่รบกวนคลื่นแม่เหล็กไฟฟ้า เช่น น้ำ, โลหะ เป็นต้น
- ปัญหาด้านการละเมิดสิทธิส่วนบุคคลหากนำอาร์เอฟไอดีไปใช้อย่างไม่ระมัดระวัง
- ปัญหาด้านความปลอดภัยของข้อมูลยังมีอยู่หากผู้โจรกรรมมีความสามารถหรือผู้ใช้งานไม่มีการจัดการเรื่องความปลอดภัยไว้

83030

**การเปรียบเทียบกับรหัสแท่ง (Barcode)**

เนื่องจากอาร์เอฟไอดีเป็นเทคโนโลยีระบบบ่งชี้อัตโนมัติ (Auto-ID) ชนิดหนึ่ง ที่เข้ามามีบทบาทสำคัญพอๆกับเทคโนโลยีรหัสแท่ง (Barcode) ที่เป็นที่รู้จักและใช้งานกันมานานอย่างแพร่หลาย จึงมักเป็นข้อเปรียบเทียบกันระหว่างสองเทคโนโลยีนี้

**ตารางที่ 2.2** แสดงข้อเปรียบเทียบลักษณะของเทคโนโลยีอาร์เอฟไอดีและ Barcode

| **อาร์เอฟไอดี** | **Barcode** |
|---|---|
| อ่านได้อัตโนมัติไม่ต้องเห็นแท็ก (Non-Line of Sight) | ต้องอ่านแนวตรงกับเครื่องอ่าน ห้ามมีอะไรบังแท็ก |
| อ่านได้ทีละหลายอัน | อ่านได้ทีละอัน |
| อ่านได้ระยะไกล | ต้องแนบกับเครื่องอ่าน |
| อ่านได้รวดเร็ว | อ่านช้ากว่า |
| อ่านและเขียนข้อมูลได้ | อ่านได้อย่างเดียว |
| ความจุของข้อมูลสูง | ความจุของข้อมูลต่ำ |
| ความถูกต้องของข้อมูลสูงกว่า(99.5%) | ความถูกต้องของข้อมูลต่ำกว่า(80%) |
| ทนทาน | เสียหายง่าย |
| ความปลอดภัยของข้อมูลสูง มนุษย์อ่านแท็กโดยตรงไม่ได้ | ไม่ได้ป้องกันความปลอดภัยใดๆ แท็กอ่านออกได้โดยมนุษย์ |
| นำกลับมาใช้ใหม่ได้ | นำกลับมาใช้ใหม่ไม่ได้ |
| นำมาใช้กับบางวัตถุไม่ได้ เช่น น้ำ, โลหะ เป็นต้น | ไม่มีข้อจำกัดในเรื่องนี้ |
| ราคาแพง | ราคาถูก |

**ตารางที่ 2.3** แสดงเงื่อนไขที่มีส่วนตัดสินใจว่าอาร์เอฟไอดีจะมาแทนหรือมาเสริม Barcode

| เงื่อนไข / ข้อกำหนด | อาร์เอฟไอดี | Barcode |
|---|---|---|
| ต้นทุนในการลงทุนเริ่มแรกต่ำ | | ✓ |
| อุณหภูมิ สูง/ต่ำ | ✓ | |
| ของเหลว | | ✓ |
| ข้อจำกัดการมองเห็นไม่ได้ | ✓ | |
| ความคงทน (ขึ้นกับลักษณะการใช้งาน) | ✓ | ✓ |
| ความยืดหยุ่น (ขึ้นกับลักษณะการใช้งาน) | ✓ | ✓ |
| ความน่าเชื่อถือ / ความถูกต้อง | ✓ | ✓ |
| การเชื่อมโยงการปฏิบัติงานกับผู้ค้าหลายราย | | ✓ |
| ความสามารถในการสแกนสิ่งของหลายตัว | ✓ | |
| การนำกลับมาใช้ใหม่ | ✓ | |

### 2.2.4 มาตราฐานของอาร์เอฟไอดี

มาตรฐานของอาร์เอฟไอดีสามารถแบ่งออกเป็น 2 ส่วน ได้แก่

- **คลื่นความถี่ (RF Spectrum)** คลื่นความถี่ที่จะใช้สำหรับอุปกรณ์อาร์เอฟไอดีนี้ ค่อนข้างยากที่จะกำหนดให้เป็นมาตรฐานสากล เนื่องจากแต่ละประเทศเป็นเจ้าของและเป็นผู้ควบคุมการใช้คลื่นความถี่เอง ดังนั้น จึงขึ้นอยู่แต่ละประเทศที่จะเป็นผู้พิจารณาจัดสรรคลื่นความถี่สำหรับการใช้งานประเภทต่างๆ
- **มาตรฐานอาร์เอฟไอดี (RFID Standards)** บริษัทผู้ผลิตต่างๆ ได้พัฒนาและผลิตระบบอาร์เอฟไอดี ออกมาโดยมีมาตรฐานที่ใช้ในการส่งถ่ายข้อมูลระหว่างเครื่องอ่านกับแท็กที่แตกต่างกันออกไป ทั้งนี้ ขึ้นกับความต้องการของตลาดเป็นสำคัญ

อย่างไรก็ตาม หลายองค์กรได้ตระหนักถึงปัญหาเรื่องมาตรฐานนี้ และได้เริ่มมีการพัฒนาระบบมาตรฐานขึ้นมาทั้งในยุโรปและอเมริกา ตัวอย่างเช่น

- **หน่วยงาน ANSI's X3T6** ซึ่งประกอบไปด้วยผู้ผลิตและผู้ใช้งานระบบอาร์เอฟไอดีในอเมริกา ได้กำลังทำการพัฒนามาตรฐานของระบบอาร์เอฟไอดีที่ความถี่ 2.45 GHz ขึ้นมา

- **The International Organization for Standards : ISO** ก็ได้มีมาตรฐานเกี่ยวกับการใช้ระบบ RFID ดังนี้

  ISO 11784 และ 11785 กับงานปศุสัตว์

  ISO/IEC 15961 : Data Protocol

  ISO/IEC 19562 : Data encoding rules and logical memory function

  ISO/IEC 15963 : Unique Identification of RF Tag

  ISO/IEC 18000 : RFID Air Interface Standard

- **The Electronic Product Code (EPC Global)** เป็นองค์กรที่เกิดจากความร่วมมือระหว่าง the Uniform Code Council (UCC) และ EAN International โดยได้รับการสนับสนุนจากภาคอุตสาหกรรม ทำหน้าที่ในการกำหนดและพัฒนามาตรฐานรหัสของสิ่งของอิเลคทรอนิกส์และโครงข่าย มาตรฐาน EPC นี้เป็นมาตรฐานเปิด (Open standard) พัฒนาขึ้นโดย the Auto-ID Center ซึ่งได้รับทุนจากบริษัทขนาดใหญ่หลายบริษัทได้แก่ บริษัท Coca-Cola, Intel, Wal-Mart และ Philips Semiconductors เป็นต้น โดยมีวัตถุประสงค์เพื่อที่จะทำให้ราคาของอุปกรณ์ฮาร์ดแวร์ลดต่ำลงจนอยู่ในระดับที่สามารถนำ RFID ไปติดไว้กับสิ่งของรายชิ้นได้

ซึ่ง 2 หน่วยงานหลังนี้ (ISO และ EPC Global) เป็นหน่วยงานหลักที่กำหนดมาตรฐาน RFID มี 4 ด้านดังนี้

- มาตรฐานด้านเทคโนโลยี (Technology)
- มาตรฐานรูปแบบของข้อมูล (Data format)
- มาตรฐานวิธีการทดสอบ (Conformance)
- มาตรฐานการใช้งาน (Applications)

เนื่องจาก EPC Global ไม่ได้กำหนดเฉพาะมาตรฐานอาร์เอฟไอดี และเป็นที่นิยมใช้ในปัจจุบัน จึงขอยกตัวอย่างที่ EPC Global กำหนดมาตรฐานของแท็กอาร์เอฟไอดีแบ่งเป็นคลาส 5 คลาส ดังนี้

คลาส 0 /1 (Class 0/Class I) ใช้กับแท็กชนิดพาสซีฟ แบบอ่านได้อย่างเดียว

คลาส 2 (Class II) ใช้กับแท็กชนิดพาสซีฟแต่มี ฟังก์ชันเพิ่มเติมขึ้นมา

คลาส 3 (Class III) ใช้กับแท็กชนิดกึ่งพาสซีฟ

คลาส 4 (Class IV) ใช้กับแท็กชนิดแอ็กทีฟ ที่ติดต่อสื่อสารกันได้ทั้งแบบบอร์ดแบนด์ (board-band) และแบบเพียร์ทูเพียร์ (peer-to-peer)

คลาส 5 (Class V) เครื่องอ่านที่ใช้มาตรฐานนี้สามารถส่งไฟเลี้ยง (power) ให้แท็กในคลาส1, 2 และ 3 ได้ ส่วนแท็กในคลาส 4 และ 5 ก็สามารถติดต่อสื่อสารกันได้

EPC -96 Structure แบ่งโครงสร้างข้อมูลของแท็กคลาสต่างๆ ดังนี้

| Element | Header | EPC Manager | Object Class | Serial Number |
|---|---|---|---|---|
| Bits | 8 | 34 | 20 | 34 |
| Values[1] | 0-256 | 0-17,179,869,183 | 0-1,048,575 | 0-17,179,869,183 (inc 0-9) |

**รูปที่ 2.9** แสดงโครงสร้างของแท็กตามมาตรฐาน EPC-96

คลาส 0 มี 64 บิต

คลาส 1 มี 96 บิต

คลาส 1 Gen2 มี 128 หรือ 256 บิต

คลาส 2 มี 96 บิตเหมือนคลาส 1 แต่มีหน่วยความจำใหญ่ขึ้น และสามารถอ่านและเขียนแท็กได้

คลาส 3 มี 96 บิตเหมือนคลาส 2 แต่เป็นแท็กชนิดกึ่งพาสซีฟ

คลาส 4 มี 96 บิตเหมือนคลาส 2 แต่เป็นแท็กชนิดแอ็กทีฟ

ISO ก็เป็นองค์กรใหญ่ที่สำคัญเช่นกัน จึงมักมีการเปรียบเทียบมาตรฐานกันกับ EPC Global

**ตารางที่ 2.4** แสดงการเปรียบเทียบมาตรฐานระหว่าง ISO/IEC กับ EPC

| | ISO/IEC | EPC |
|---|---|---|
| เทคโนโลยี | ISO/IEC 18000 – RF-IC for<br>Item Management<br>Part2 -< 135 kHz<br>Part3 – 13.56 MHz<br>Part4 – 2450 MHz<br>Part6 – 860 – 960 MHz<br>Part7 – 433.92 MHz (active) | Class I-V (13.56 and UHF only)<br>Class 0/Class I: read-only passive tags<br>Class II tags : passive tags with additional<br>functionality Class III tags: semi–passive RFID<br>tags<br>Class IV tags: active tags<br>With broad-band peer-to-peer<br>Communication<br>Class V tags : Readers Can<br>Power other Class I, II and<br>III tags: Communication with<br>Classes IV and V |
| รูปแบบของข้อมูล | ISO/IEC 15418 – Application<br>Identifiers & Data Identifiers<br>ISO/IEC 15434 – Syntax<br>ISO/IEC 15459 – Transport<br>License Plate<br>ISO/IEC 15961 – Data Protocol:<br>Application Interface<br>ISO/IEC 15962 – Data Protocol:<br>Data Encoding Rules and<br>Logical Memory Functions | EPC<br>Class 0 – 64 bits<br>Class 1 – 96 bits<br>Class 1 G2 – 128/256 bits<br>Class 2 – Class 1 with larger<br>Memory and read/write<br>Class 3 – Class 2 with<br>Sensors (semi-passive)<br>Class 4 – passive tags |

## 2.3 ไอรอนไพธอน (IronPython)

IronPython เป็นภาษาที่สามารถพัฒนาโปรแกรมด้วยภาษา Python นำมาประมวลบนแพลตฟอร์ม (Platform) .NET และ Mono ได้ โดย Jim Hugunin เป็นคนสร้างภาษานี้ขึ้นมา เวอร์ชัน 1.0 (Version 1.0) ประกาศออกมาให้ใช้กันเมื่อ 5 กันยายน 2549

ภาษา IronPython ตั้งแต่เวอร์ชัน 0.6 อยู่ภายใต้มาตรฐาน Common Public License เมื่อเดือนสิงหาคม ปี 2547 ก็มีการรวมตัวกันของหลายๆ โครงการพัฒนาภาษา IronPython เนื่องจากเริ่มแรกภาษา IronPython ถูกสร้างให้เป็นส่วนหนึ่งของ Microsoft's Shared Source ผู้พัฒนาก็เลยอ้างว่าไม่

น่าจะเป็น Open source อีกต่อไปเพราะไม่ได้เริ่มสร้างจากองค์กร OSI: Open Source Initiative ก็เลย มีเวอร์ชัน 2.0 แอลฟา (2.0 alpha) ออกมาภายใต้ Microsoft Permissive License

ภาษา IronPython เขียนขึ้นมาด้วยภาษา C# เกือบทั้งหมด มีบางส่วนเป็นภาษา Python ที่เกิดอัตโนมัติจาก code generator ในภาษา Python

### 2.3.1 เส้นทางการพัฒนาของภาษา IronPython

IronPython เวอร์ชั่นล่าสุดในปัจจุบัน คือเวอร์ชัน 1.1 ซึ่งเปรียบเทียบได้กับการใช้งาน CPython 2.4.4 แต่ถึงอย่างไรก็มีข้อแตกต่างกันอยู่บ้างในการเขียนภาษา Python กับ IronPython

ส่วนเวอร์ชัน 2.0 แอลฟา (ที่ประกาศออกมาในปัจจุบัน คือ 2.0 Alpha Release 4) จะเทียบได้กับ CPython 2.5 IronPython 2.0 จะอยู่ที่ส่วนบนของ Dynamic Language Runtime: DLR ที่ประกอบด้วย Dynamic Type System และ Dynamic Language Hosting Environment ที่ถ่ายทอดมาจาก IronPython 1

DLR ถูกสร้างขึ้นมาเพื่อให้สามารถเขียนโปรแกรมภาษาพวกที่เป็นไดนามิกส์ (Dynamic Language) บน CLR: Common Language Runtime ได้สะดวก ซึ่งพวกภาษาแบบไดนามิกส์ที่กำลังพัฒนาให้รันบน DLR ได้นั้นก็มีอีกหลายภาษาเช่น IronRuby, Jscript, VBx

DLR รันอยู่ส่วนบนสุดของแก่น CLR เลยที่ครอบคลุม Microsoft Silverlight อยู่ด้วย ดังนั้น IronPython ก็เขียน Script ของ Silverlight ผ่านบราวเซอร์ browser ที่ฝั่งไคล์แอนต์ (Client-side) ได้

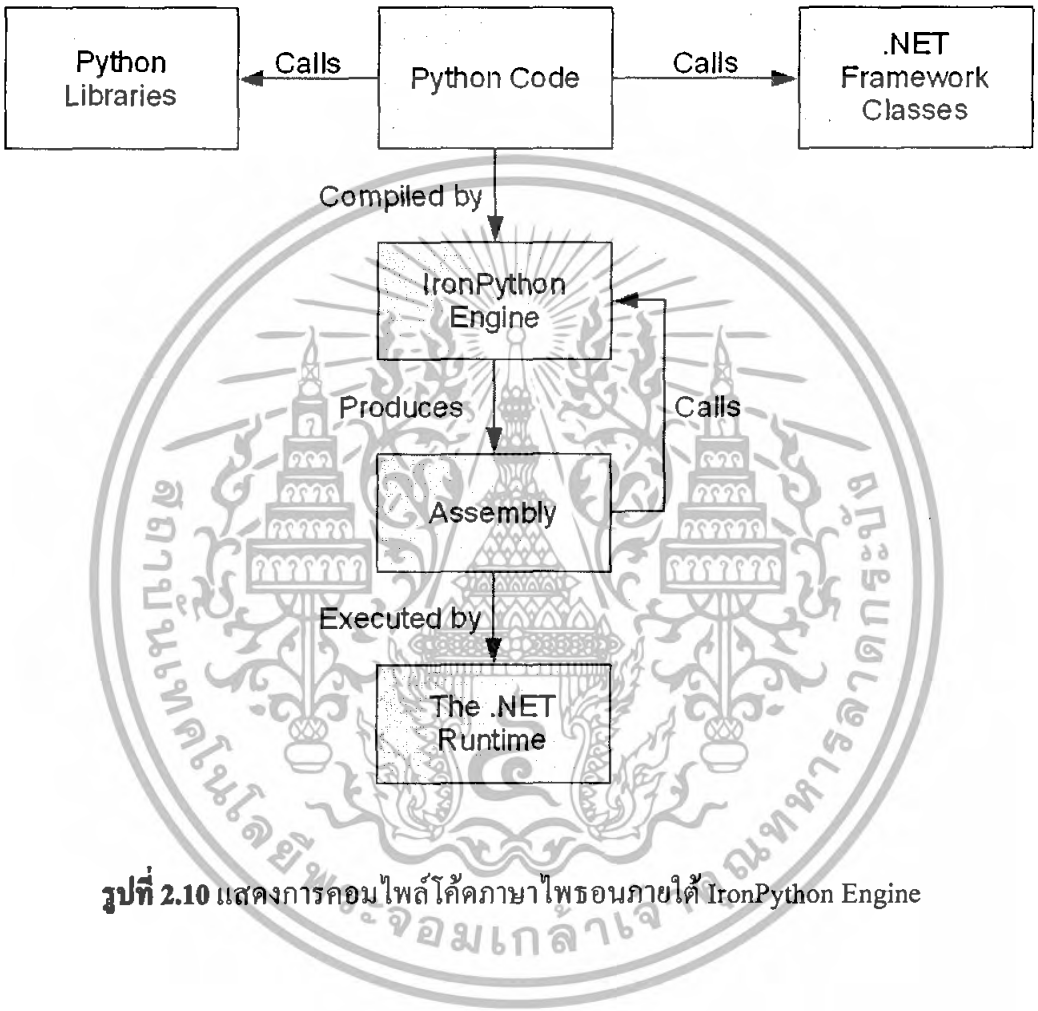
### 2.3.2 ขอบเขตการใช้งาน

ข้อดีของ IronPython ที่สำคัญอย่างหนึ่งคือ มีฟังก์ชันครอบคุมการใช้งาน .NET framework เพื่อพัฒนาโปรแกรมบน .NET ได้เหมือนพวกภาษาของ .NET เอง วิธีการง่ายๆ คือ การนำตัวแปลภาษา IronPython (IronPython Interpreter) ไปรวมกับ .NET framework ที่มีอยู่แล้ว ซึ่งนักพัฒนาก็ลองเขียน script ในภาษา IronPython ติดต่อกับออบเจกต์ (Object) ใน .NET framework แล้วก็ใช้การได้ นอกจากนี้ยังเพิ่มฟังก์ชั่นอื่นเพิ่มเติมเข้าไปใน .NET framework ได้เลยโดยไม่ต้องเปลี่ยนแปลงโค้ดโครงสร้างหลักของเก่า

**ตารางที่ 2.5** แสดงคลาสในดอทเน็ตเฟรมเวิร์ก (Common .NET Assemblies and Namespaces)

| **Assembly Name** | **Purpose** |
|---|---|
| System | Contains the base .NET types, exceptions, garbage collection classes and much more. |
| System.Data | Classes for working with databases, both high and low level. |
| System.Drawing | Provides access to the GDI+ graphics system. |
| System.Management | Provides access to Windows management information and events (WMI), useful for system administration tasks. |
| System.Environment | Allows you to access and manipulate the current environment, like command line arguments and environment variables. |
| System.Diagnostics | Interact with processes. |
| System.XML | For processing XML, including SOAP, XSL/T and more. |
| System.Web | The ASP.NET web development framework.<br>System.IO Contains classes for working with paths, files and directories. Includes classes to read and write to file systems or data-streams, synchronously or asynchronously.<br>Microsoft.Win32 Classes that wrap Win32 common dialogs and components including the registry. |
| System.Threading | Classes needed for multithreaded application development. |
| System.Text | Classes for working with strings (like StringBuilder) and the Encoding classes which can convert text to and from bytes. |
| System.Windows.Forms | Provides a rich user interface for applications. |
| System.Windows | The base namespace for WPF, the new GUI framework that is part of .NET 3.0. |
| System.ServiceModel | Contains classes, enumerations, and interfaces to build Windows Communication Foundation (WCF) service and client applications. |

IronPython ทำให้การใช้งานออบเจกต์ต่างๆ ของ .NET ได้มีประสิทธิภาพมากขึ้น เพราะเมื่ออ้างอิงถึงออบเจกต์ใดออบเจกต์หนึ่งใน .NET IronPython จะมีการใช้กลไก Reflection คือทำการอิมพอร์ท (import) ทั้ง type และ method ทั้งหมดของออบเจกต์นั้น ทำให้เราสามารถเพิ่มหรือเปลี่ยนแปลงออบเจกต์นั้นๆในรูปแบบของภาษา IronPython เองได้




**รูปที่ 2.10** แสดงการคอมไพล์โค้ดภาษาไพธอนภายใต้ IronPython Engine

### 2.3.3 วิธีการใช้งาน

สำหรับภาษาไอรอนไพธอนได้มีทางเลือกให้ผู้พัฒนาโปรแกรมได้หลายทาง เช่น ใช้ไอรอนไพธอนร่วมกับ Visual Studio 2005 ซึ่งต้องลง SDK เวอร์ชั่นล่าสุดหรือเวอร์ชั่น 4.0 ขึ้นไป อีกวิธีคือใช้อิดิเตอร์ (Editor) ของไพธอนทั่วไปที่มีอยู่แล้ว นำโค้ดมารันผ่านคอมไพเลอร์ของไอรอน-ไพธอน (IronPython Engine) เพียงแค่ดาวน์โหลด (Download) มาแล้วเรียกใช้ตอนที่จะคอมไพล์ผ่านคอมมานด์ไลน์ (Command line)

ในที่นี้เราใช้วิธีหลัง เนื่องจากมีความคุ้นเคย (เพราะ SDK ได้พัฒนาออกมาทีหลัง) แต่ก็ยังได้ใช้ประโยชน์จากวิธีแรกอยู่ กล่าวคือ เราสามารถใช้สิ่งที่ง่าย และสะดวกจาก SDK ได้เช่น การออกแบบแอพพลิเคชั่น สามารถใช้ส่วนประกอบ (Component) จากแถบเครื่องมือ (Tool box) ออกแบบได้ง่าย ไม่ได้เขียนโค้ดเองทั้งหมด แต่เนื่องจากบางคุณลักษณะที่เราต้องการไม่สามารถทำได้ด้วยวิธีตามดอทเน็ตที่คุ้นเคยธรรมดา (หรืออาจเป็นเพราะยังไม่รู้วิธีการใช้ที่ถูกต้อง) จึงใช้วิธีการแบบผสม เพื่อความสะดวก รวดเร็ว และง่ายต่อการพัฒนา

**ตารางที่ 2.6** แสดงการเขียนโค้ด Hello World ง่ายๆ เปรียบเทียบระหว่าง C# และ IronPython

| **A Small HelloWorld App in C#** | **Equivalent in Python** |
|---|---|
| using System;<br>class Hello<br>{<br>string msg;<br>public Hello()<br>{<br>this.msg = "Hello World";<br>}<br>public Hello(string msg)<br>{<br>this.msg = msg;<br>} | class Hello(object):<br>def __init__(self,<br>msg='hello world'):<br>self.msg = msg<br>def sayHello(self):<br>print self.msg<br>app = Hello ()<br>app.sayHello() |

ข้อควรระวังในการใช้ Visual Studio 2005 ที่รวมไอรอนไพธอนอยู่ด้วยนั้น ต้องระวังเรื่องชนิดของตัวแปร (Data types) การเขียนด้วยไพธอน เราไม่ต้องประกาศชนิดตัวแปรไว้ ในไอรอนไพธอนก็เช่นเดียวกัน แต่หากมีการเรียกใช้ตัวแปรบางชนิดของดอทเน็ต ต้องระมัดระวังการใช้อาจไม่เหมือนที่คุ้นเคย กล่าวคือ เราต้องรู้ชนิดของตัวแปรนั้นอย่างแท้จริง แล้วเรียกใช้แบบชื่อเต็ม (หากไม่ประกาศให้ใช้แค่ชื่อได้) เช่น ชนิดของตัวแปรนั้นเป็น null ที่รู้จักกันทั่วไป ในไพธอนจะใช้ none

หรือกรณี null ในฐานข้อมูล ดอทเน็ตยังคงใช้ null เหมือนเดิมได้ แต่ไอรอนไพธอน ต้องใช้ System.DBnull

โดยปกติดอทเน็ตจะสร้างโค้ดของแต่ละส่วนประกอบให้อัตโนมัติ แต่มีบางส่วนที่ใช้ไม่ได้ เช่น การใช้ DataGridView ในดอทเน็ตจะมีการสร้างโค้ด DataGridViewName.BeginInit() กับ DataGridViewName.EndInit() ซึ่งส่งผลให้คอมล์โค้ดไม่ได้ แต่ตอนออกแบบต้องให้มี statements นี้อยู่จึงจะไม่เกิด Error แต่ตอนรันเราก็ต้องเอา statements นี้ออก

## 2.4 การเชื่อมต่อตัวอ่านอย่างง่าย

Basic Reader Interface (BRI) พัฒนาขึ้นมาเพื่อควบคุมการทำงานของ RFID Reader โดยสร้างให้มี overhead ของโปรโตคอลน้อยที่สุด BRI เป็น emmbeded softwareในตัว RFID Reader ซึ่งตัวsoftwareนี้เป็น อินเตอร์เฟสที่เชื่อมต่อระหว่าง โฮสต์กับ RFID Manager ที่รันอยู่บน RFID Reader จุดมุ่งหมายของทำการเชื่อมต่อทำให้อุปกรณ์ที่ไม่มีความสามารถในการติดต่อ เช่นอุปกรณ์ประเภท Programmable Logic Controllers (PLC) สามารติดต่อและควบคุม RFID Reader ได้ด้วยคำสั่งง่ายๆผ่านทางอินเตอร์เฟส

BRI ยังเป็นอินเตอร์เฟสที่เชื่อมต่อไปยัง RFID Manager Core software ซึ่งเป็นส่วนที่ทำหน้าที่ในการควบคุมการทำงานของ RFID Radio Interface BRI มีสามารถปรับโหมดการทำงานได้อย่างหลากหลายและมีประสิทธิภาพสูง

### 2.4.1 คุณสมบัติของ BRI

- รองรับการทำงานผ่านคำสั่งแบบ ASCII
- มีรูปแบบของชุดคำสั่งที่เข้าใจได้ง่าย
- มาตรฐานการใช้ประกาศค่าตัวแปรสตริงและตัวเลขเหมือนในโปรแกรมมิ่งภาษา C เช่นการใช้ single quotes แทน ASCII 1 character และ double quotes แทนหลาย ASCII character
- รองรับการเชื่อมต่อหลากหลายรูปแบบ เช่น serial, Ethernet, USB และอื่นๆ
- รองรับ general purpose input and output (GPIO) interface ในการกำหนดค่าพารามิเตอร์ต่างๆในการควบคุมการทำงานของ RFID Reader โดยตัว RFID Reader ต้องรองรับการทำงานของ GPIO ด้วย

- มีชุดคำสั่งที่หลากหลายซึ่งสามารถนำไปใช้กับ RFID Reader แต่ละชนิดที่มีความแตกต่างกันได้
- สามารถสร้าง macro เพื่อกำหนดค่าพารามิเตอร์และcommand ต่างๆลงใน RFID Reader ได้เองเพื่อเพิ่มความสามารถในการทำงานของ RFID Reader

**2.4.2 รูปแบบการเรียกใช้ BRI**

โดยปกติการเรียกใช้ BRI มักจะอยู่ในรูปของการติดต่อผ่านทางอินเตอร์เฟสระหว่างคอมพิวเตอร์ โดยการเรียกใช้ BRI เพื่อติดต่อไปยัง RFID Reader สามารถทำได้ 2 รูปแบบคือ

- ติดต่อผ่านทาง serial RS-232
- ติดต่อผ่านทาง TCP connection โดยผ่านทาง port 2189

**2.4.3 การพัฒนาโปรแกรมโดยใช้ BRI**

ในการพัฒนาแอพพลิเคชั่นโดยใช้ BRI เป็นโปรโตคอลติดต่อไปยัง RFID reader สามารถทำการติดต่อได้สองรูปแบบคือ

1. พัฒนาโฮสต์แอพพลิเคชั่นให้เป็นตัวจัดการปรับแต่งค่าพารามิเตอร์ต่างๆของ RFID เองโดยมีข้อดีคือ

- Extensibility สามารถให้โฮสต์แอพพลิเคชั่นทำการปรับแต่งให้รองรับการอัพเกรดระบบ
- Scalability โฮสต์แอพพลิเคชั่นสามารถเลือกอุปกรณ์ที่จะติดต่อและโดยการระบุชื่อหรือ address ของอุปกรณ์
- Maintainability ง่ายต่อการซ่อมบำรุงฮาร์ดแวร์

2. ใช้ RFID Resource Kit ซึ่งเป็น API ที่พัฒนาขึ้นโดยบริษัท Intermec ซึ่งสามารถใช้งานร่วมกับผลิตภัณฑ์ของบริษัท Intermec ได้ทั้งหมด สามารถ download ได้ที่ www.intermec.com เลือกที่ Service &Support > Developer Library > Resource Kits. ในการพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีจะใช้ RFID Resource kit ในการพัฒนา

ในการใช้งาน RFID Resource Kit จะประกอบด้วยโมเดลในการเขียนโปรแกรมอยู่สองโมเดลคือ

- Basic Programming model เป็นอินเดอร์เฟสระดับ low-level ในการติดต่อไปยัง BRI reader ทำงานในลักษณะของ command/response ใช้งานโดยอาศัย syntax ของ BRI ซึ่งเข้าใจได้ยากกว่า โมเดลแบบ Advance การใช้งานโมเดลแบบ Basic มีคลาสในการทำงานหลักคือ BasicBRIReader มีพื้นฐานในการทำงานคือ การส่ง BRI commands ไปยัง RFID Reader และรอรับการ responses and asynchronous event จาก RFID Reader
- Advanced Programming model รูปแบบของการเขียนโปรแกรมติดต่ออาร์เอฟไอดีขั้นสูง ก็มีการเตรียมฟังก์ชันที่ดีกว่าให้ใช้งาน เพื่อจัดการกับแท็กข้อมูลที่มากกว่าโมเดลแบบ Basic ซึ่งสามารถตั้งค่าได้ทั้งที่เครื่องอ่านแท็กและการจัดการต่างๆเกี่ยวกับแท็กได้มีประสิทธิภาพ ช่วยให้ผู้เขียนโปรแกรมใช้งานได้ง่ายขึ้น ไม่ต้องยุ่งกับรูปแบบการเขียน BRI มากนัก ก็สามารถเอาข้อมูลที่อยู่ในแท็กออกมาได้โดยง่าย

ปกติเราสามารถเขียนแอพพลิเคชั่นโดยใช้แค่ BRI ซึ่งจะใช้รูปแบบการเขียนที่เข้าใจง่ายและใช้ง่ายด้วย มีสองรูปแบบใหญ่ๆ คือ การกรองแท็ก (tag-filter statements) เพื่อแยกกลุ่มหรือประเภทของแท็ก และ โครงสร้างของแท็ก (schema statements) เพื่อกำหนดจำนวน หรือตำแหน่งของข้อมูลในแท็ก (data fields) ที่จะอ่านหรือเขียน

การใช้รูปแบบนี้ จะใช้คลาส BRIReader ซึ่งจะแบ่ง BRI ทำงานเป็นขั้นตอน (Method) คือ ส่วนอ่านแท็ก (Read) และ ตรวจจับแท็ก (PollTags) แล้วแบ่งเป็นเหตุการณ์ (BRI tag events) แล้วห่อหุ้ม (encapsulate) ข้อมูลที่เกี่ยวข้องกับแต่ละแท็กที่เครื่องอ่านตรวจจับได้ คือแท็กคีย์ (Tag Key) และข้อมูลที่เรากำหนด เก็บไว้เป็นอาเรย์ (Array) ที่อยู่ในแท็ก

ในการพัฒนาระบบจะใช้โมเดลแบบ Advanced ในการติดต่อไปยัง RFID Reader คลาสในการทำงานหลักคือ BRIReader โดยมี constructor ดังนี้

```
public BRIReader(Control Owner);
public BRIReader(Control Owner, string aReaderURI);
public BRIReader(Control Owner, string aReaderURI, BRIReader.LoggerOptionsAdv LogOpts);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize);
public BRIReader(Control Owner, string aReaderURI, string aFieldSchema);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize, BRIReader.LoggerOptionsAdv LogOpts);
public BRIReader(Control Owner, string aReaderURI, int aReadBufferSize, int aEventQueueSize);
public BRIReader(Control Owner, string aReaderURI, string aFieldSchema, BRIReader.LoggerOptionsAdv LogOpts);
```

ในการพัฒนาเลือกใช้ constructor ในรูปแบบที่ 3 โดยค่าพารามิเตอร์คือ

1. Control Owner คือ control ตัวที่จะทำการสร้าง BRIReader โดยปกติตัวที่เรียกจะเป็น System.Windows.Forms
2. String aReaderURI เป็นสตริงที่บอกถึงการเชื่อมต่อและที่อยู่ของ RFID Reader ถ้าเป็นการเชื่อมต่อผ่านทาง TCP สตริงจะมีค่าเป็น "TCP://nnn.nnn.nnn.nnn" โดย n แทนค่าไอพีแอดเดรสของ RFID Reader และถ้าเชื่อมต่อผ่านทาง serial RS-232 สตริงจะมีค่าเป็น "SERIAL://COM1"
3. BRIReader.LoggerOptionAdv LogOpts เป็นการเก็บ log ในการติดต่อ RFID Reader

เมื่อทำการสร้างคลาส BRIReader แล้วจะสามารถเรียกใช้ฟังก์ชันการทำงานต่างๆของ RFID ได้ผ่านทางคลาสที่สร้างไป โดยฟังก์ชันหลักที่ใช้งานคือ ReadTagsAsStringData() ซึ่งจะทำการอ่านแท็กอาร์เอฟไอดีแล้วเก็บรายละเอียดของแท็กไว้ใน BRIReader.Tags เพื่อนำไปใช้งานในการระบุถึงสินค้าแต่ละชิ้นได้

เนื่องจากในระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีต้องมีการใช้งานร่วมกับระบบอื่นๆที่ทำการพัฒนาไว้แล้วในห้องวิจัยการสื่อสารและคมนาคมชาญฉลาด (ICT Lab) ซึ่งพัฒนาไว้โดยใช้ภาษาไพธอน แต่ในการติดต่ออุปกรณ์อาร์เอฟไอดีต้องติดต่อผ่าน BRI ซึ่งพัฒนาโดยใช้ภาษา C# จึงทำการพัฒนาโปรแกรมในการติดต่ออาร์เอฟไอดีโดยเขียนเป็นอินเตอร์เฟสโดยใช้ภาษาไอรอนไพธอนซึ่งมีรายละเอียดดังนี้

```
import clr
clr.AddReference('System')
clr.AddReference('System.Windows.Forms')
clr.AddReference("Intermec.DataCollection.RFID.BasicBRI.dll")
clr.AddReference("Intermec.DataCollection.RFID.AdvancedBRI.dll")
import System
from System.Windows.Forms import *
from Intermec.DataCollection.RFID import *

class InterfaceRFID(Form):
    def __init__(self):
        #"""RFID Parameters"""
        self.m_Reader = BRIReader
        self.logOpts = BRIReader.LoggerOptionsAdv()
        self.logOpts.IDEConsoleEnable = False
        self.logOpts.LogEnable = True
        self.logOpts.LogFilePath = "C:\MySampleLog1.txt"
        self.logOpts.TimeStampEnable = True
        self.logOpts.ShowNonPrintableChars = True
        self.logOpts.MaxFileSize = 400000
        self.m_fConnected = False
        self.sCurrentTagType = "GEN2"
        self.fIP4Connected = True
```

ฟังก์ชันหลักของการทำงานในการติดต่อ RFID Reader คือ ฟังก์ชัน OpenRFIDReader() จะทำการติดต่อไปยัง RFID Reader โดยจะเก็บ log เป็นไว้ที่ C:\MySampleLog1.txtและเก็บค่าตัวแปร self.m_fConnected เป็นตัวแปร boolean เพื่อบอกว่าติดต่อ RFID Reader สำเร็จหรือไม่

```
def OpenRFIDReader(self,sURIAddress):
        self.fSuccess = True
        try:
            self.m_Reader = BRIReader(self,sURIAddress,self.logOpts)
        except:
            self.m_Reader = None
            self.fSuccess = False
        if((self.m_Reader != None) and (self.m_Reader.IsConnected)):
            self.m_fConnected = self.fSuccess
        else:
            self.m_fConnected = False
        return self.m_fConnected

def ReadTagsAsStringData(self):
        self.Data = ""
        self.fSuccess = False
        self.sSchema = None
        self.sFilter = None
        if (self.m_Reader == None):
            return False
        try:
            self.fNoUpdate = False
            while True:
                self.fTBUpdated = False
                if (self.sSchema == None):
                    self.m_Reader.DefaultFieldSchema = None
                    self.fSuccess = self.m_Reader.Read()
                else:
                    self.fSuccess =
self.m_Reader.Read(self.sFilter,self.sSchema)
                if (self.m_Reader.TagCount == 0):
                    self.Data = "No Tags detected"
                    return False
                for i in range(0,self.m_Reader.TagCount):
                    self.Data += self.m_Reader.Tags[i].ToString()
                    if i != self.m_Reader.TagCount -1 :
                        self.Data += ","
                return self.Data
                break
        except:
            self.fSuccess = False
        return self.fSuccess
```

ส่วนฟังก์ชัน ReadTagsAsString() จะทำการอ่านแท็กและเก็บค่าเป็นสตริงในตัวแปร self.Data โดยเก็บรหัสของแท็กอาร์เอฟไอดีที่อ่านได้และคั่นด้วยเครื่องหมายจุลภาค

# บทที่ 3
# การออกแบบระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้ RFID

## 3.1 ความสามารถของระบบ

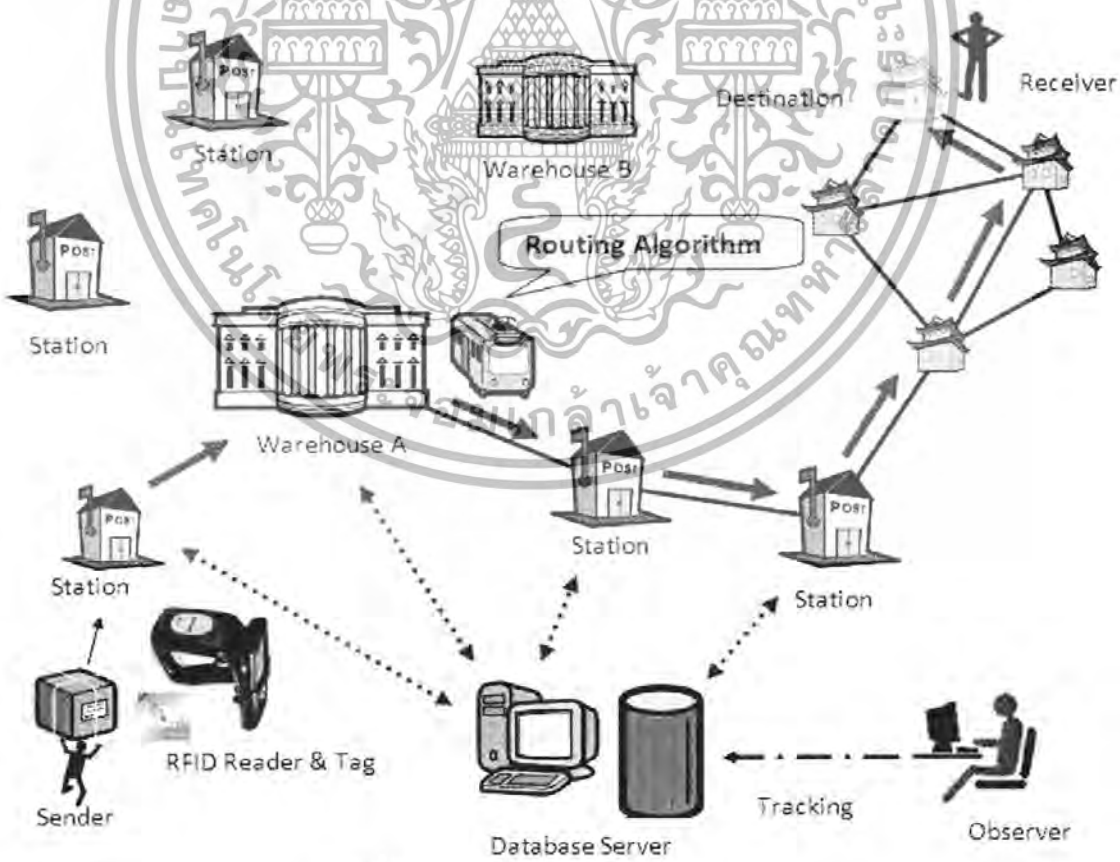
ระบบการขนส่งที่ออกแบบขึ้นมีความสามารถดังต่อไปนี้

1. ระบบขนส่งสิ่งของมีระบบฐานข้อมูลที่สามารถบันทึกกิจกรรมการขนส่ง
2. สามารถติดตามการขนส่งตั้งแต่สิ่งของเข้ามาในระบบจากผู้ส่งจนถึงปลายทางผู้รับได้
3. สามารถนำข้อมูลต่างๆ มาวิเคราะห์เพื่อการวางแผนการส่งสิ่งของให้มีประสิทธิภาพ

## 3.2 การออกแบบ

### 3.2.1 การออกแบบระบบโลจิสติกส์

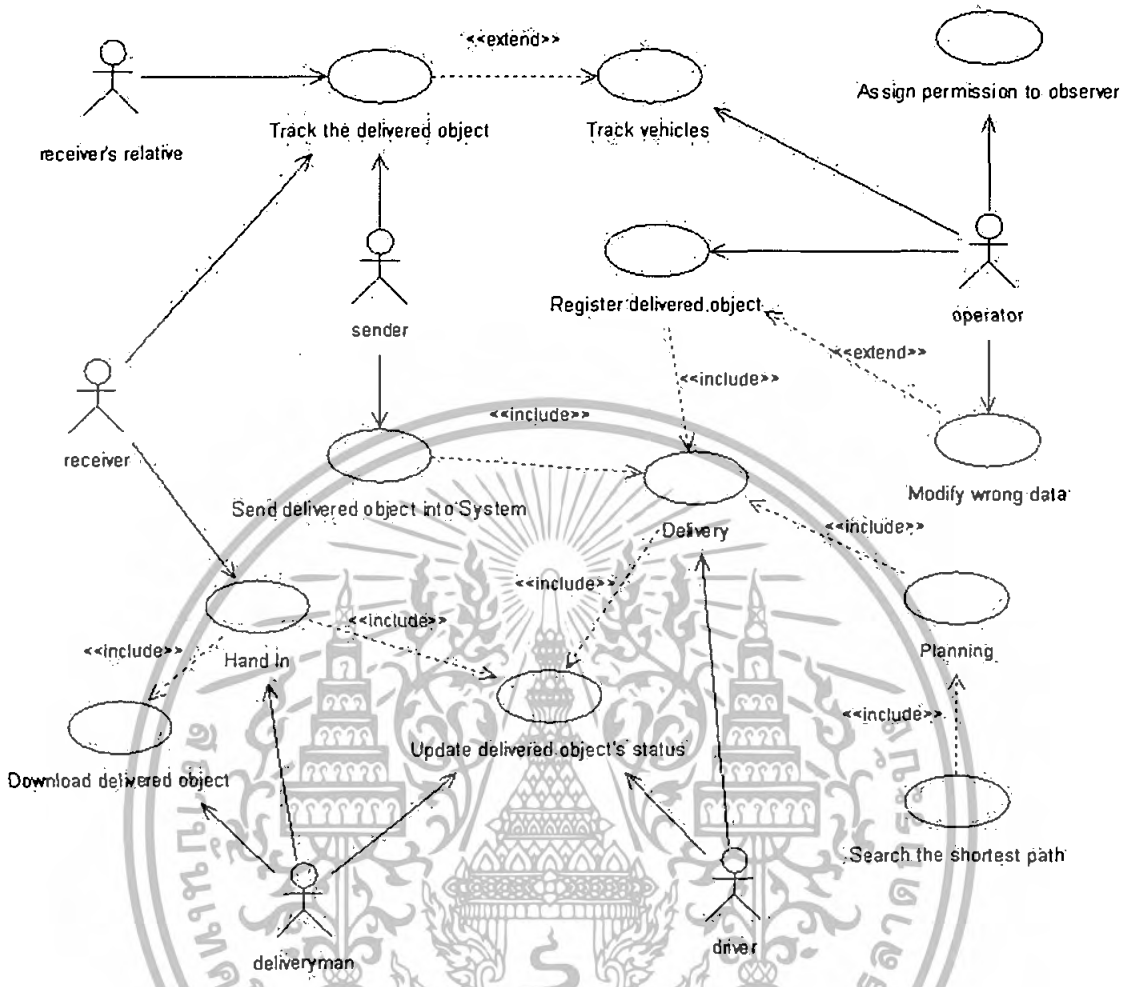
ในการพัฒนาระบบโลจิสติกส์สามารถแบ่งได้เป็นสองส่วนคือ คือ ส่วนคลังจัดเก็บสิ่งของและส่วนการขนส่งสิ่งของ ซึ่งในที่นี้จะพัฒนาระบบโดยเน้นไปในส่วนการขนส่งสิ่งของเท่านั้น




**รูปที่ 3.1** แสดงการทำงานของระบบ

ในระบบที่ออกแบบขึ้นนี้ จะควบคุมตั้งแต่เมื่อสิ่งของเข้าสู่ระบบ โดยประเภทของการขนส่งสิ่งของมี 2 ประเภท คือ แบบธรรมดา และ แบบด่วนพิเศษ โดยสิ่งของทั้งหมดที่เข้ามาในระบบแล้วจะต้องติดแท็กอาร์เอฟไอดีทุกชิ้น เพื่อใช้ประโยชน์จากเทคโนโลยี Auto-ID ซึ่งข้อมูลรายละเอียดเกี่ยวกับสิ่งของที่จำเป็นในการขนส่งจะถูกบันทึกลงฐานข้อมูลทั้งหมด ในที่นี้จะถือว่า สิ่งของเริ่มส่งจากสถานีศูนย์กลาง ดูแลจัดการกระจายการขนส่งไปที่สถานีย่อย (ไม่ได้ดูแลเรื่องการขนสิ่งของกลับเข้าศูนย์กลาง) โดยสถานีย่อยจะแบ่งเขตการรับผิดชอบการขนส่งไปถึงที่อยู่ปลายทางของผู้รับ ศูนย์กลางจัดการเรื่องเส้นทางการขนส่งโดยใช้อัลกอริทึมเอสตาร์ (A* Algorithm) เพื่อหาเส้นทางที่สั้นที่สุด เมื่อรถขนส่งมาส่งสิ่งของที่สถานีย่อยใดๆ จะทำการอัพเดทข้อมูลลงฐานข้อมูลด้วย จากสถานีไปส่งที่ที่อยู่ของผู้รับก็ใช้วิธีการหาเส้นทางเช่นแบบเดียวกัน เมื่อส่งสิ่งของถึงมือผู้รับแล้ว ผู้รับต้องรับรองการรับสิ่งของนั้นด้วย ทั้งนี้ระหว่างการขนส่งผู้ที่เกี่ยวข้องกับสิ่งของนั้นๆ จะมีไอดีที่สามารถใช้ในการติดตามสิ่งของได้ผ่านเว็บแอพพลิเคชั่น แล้วผู้ขนส่งต้องอัพเดทข้อมูลการรับสิ่งของนั้นด้วย จึงจะถือว่าสิ้นสุดการขนส่งสิ่งของนั้นๆ

ในส่วนของการออกแบบ มองภาพรวมของระบบขนส่งสิ่งของในชีวิตจริงแล้วเขียน Use Case Diagram ได้ดังนี้




**รูปที่ 3.2** แสดง Use Case Diagram ส่วนควบคุมจัดการระบบขนส่ง

ในระบบมีผู้เกี่ยวข้อง (Actor) หลัก 6 ส่วนคือ

1 พนักงาน (Operator) มีหน้าที่รับสิ่งของเข้าระบบแล้วระบุให้สิ่งของมีเอกลักษณ์ไม่ซ้ำกัน กรอกรายละเอียดเกี่ยวกับสิ่งของที่เกี่ยวข้องทั้งหมดลงระบบฐานข้อมูล สามารถปรับเปลี่ยนข้อมูลเกี่ยวกับสิ่งของที่ไม่ถูกต้องก่อนเริ่มการขนส่งได้ สามารถติดตามการขนส่งสิ่งของที่เกี่ยวข้องกับสถานีนั้นๆได้ตลอดเส้นทางทั้งหมด มีสิทธิ์ในการอนุญาตสิทธิ์การติดตามสถานะการขนส่งสิ่งของให้แก่ผู้ที่เกี่ยวข้องทุกฝ่าย นอกจากนี้พนักงานยังสามารถนำข้อมูลที่บันทึกไว้มาวิเคราะห์หรือใช้งานอย่างอื่นได้อีกด้วย เพื่อพัฒนาปรับปรุงประสิทธิภาพของระบบต่อไป

2 ผู้ส่ง (Sender) คือบุคคลหรือหน่วยงานที่มีความต้องการส่งสิ่งของใดๆ เข้ามาในระบบ เพื่อใช้บริการของระบบในการส่งของให้ถึงมือผู้รับ สามารถตรวจสอบสถานะการนำส่งได้ตลอดเวลาที่ของยังอยู่ในระบบ

3 ผู้รับ (Receiver) คือบุคคลหรือหน่วยงานที่เป็นผู้มีสิทธิ์รับสิ่งของที่ผู้ส่งต้องการส่งให้ หากผู้รับได้รับสิ่งของถูกต้องแม่นยำหรือไม่ ก็มีผลต่อประสิทธิภาพของระบบขนส่ง

4 ผู้ขนส่ง (Deliveryman) มีหน้าที่ส่งสิ่งของจนถึงมือผู้รับในระบบ รายงานผลการขนส่งจนถึงมือผู้รับ

5 ผู้เกี่ยวข้องกับผู้รับ (Receiver's relative) คือบุคคลหรือหน่วยงานที่มีความเกี่ยวข้องกับผู้รับสิ่งของ สามารถตรวจสอบและติดตามสถานะการส่งของสิ่งของได้

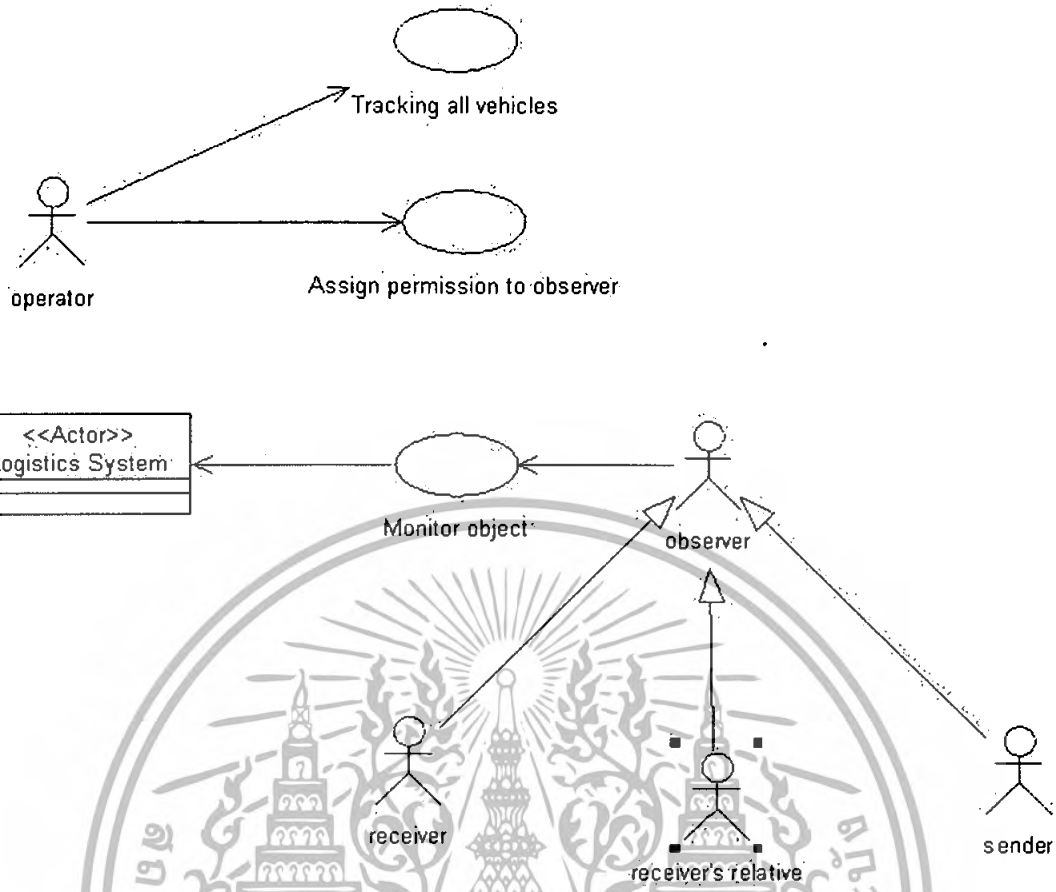
6 พนักงานขับรถ (Driver) ทำหน้าที่เป็นคนดำเนินการตามที่ระบบคำนวณเส้นทางในการขนส่งจากการใช้อัลกอริทึมเอสตาร์ในการแก้ปัญหา เมื่อส่งสิ่งของเข้าที่สถานีย่อยใดๆ ผู้ขับรถขนส่งต้องอัพเดทข้อมูลลงฐานข้อมูลทุกครั้ง

ส่วนควบคุมจัดการระบบขนส่งเริ่มตั้งแต่มีผู้ส่งต้องการส่งสิ่งของถึงผู้รับ นำสิ่งของนั้นมาเจ้าหน้าที่ก็เป็นคนดำเนินการติดแท็กอาร์เอฟไอดี แล้วกรอกรายละเอียดที่จำเป็นในการขนส่ง เช่น ชื่อ ที่อยู่ของผู้รับ น้ำหนักและประเภทของการขนส่ง เพื่อคำนวณค่าบริการการขนส่ง แล้วเมื่อสิ้นสุดการรับข้อมูล เจ้าหน้าที่ต้องให้หมายเลขที่ใช้ในการติดตามการขนส่งสิ่งของแก่ผู้ส่ง ซึ่งผู้ส่งจะแจ้งหมายเลขให้แก่ผู้รับ หรือผู้ที่เกี่ยวข้องทราบอย่างไรนั้น ขึ้นกับผู้ส่งเอง ผู้ที่ทราบหมายเลขนี้สามารถดูสถานะการขนส่งได้ตั้งแต่ต้นจนจบผ่านเว็บแอพพลิเคชั่น

เจ้าหน้าที่สามารถแก้ไขข้อมูลที่พบว่าผิดพลาดได้ หากสิ่งของนั้นยังอยู่ในสถานีที่รับผิดชอบ นอกจากนี้เจ้าหน้าที่ยังมีสิทธิ์ในการกำหนดสิทธิ์การติดตามการขนส่งแก่บุคคลที่เกี่ยวข้องได้ และยังสามารถติดตามการขนส่งสิ่งของที่เข้าสถานีที่รับผิดชอบได้ทั้งหมด เจ้าหน้าที่ที่ศูนย์กลางสามารถติดตามรถขนส่งได้ด้วย

ผู้ขับรถขนส่งต้องทราบเส้นทางการขนส่งจากการใช้อัลกอริทึมเอสตาร์ในการแก้ปัญหาเมื่อส่งสิ่งของเข้าที่สถานีย่อยใดๆ ผู้ขับรถขนส่งต้องอัพเดทข้อมูลลงฐานข้อมูลทุกครั้ง

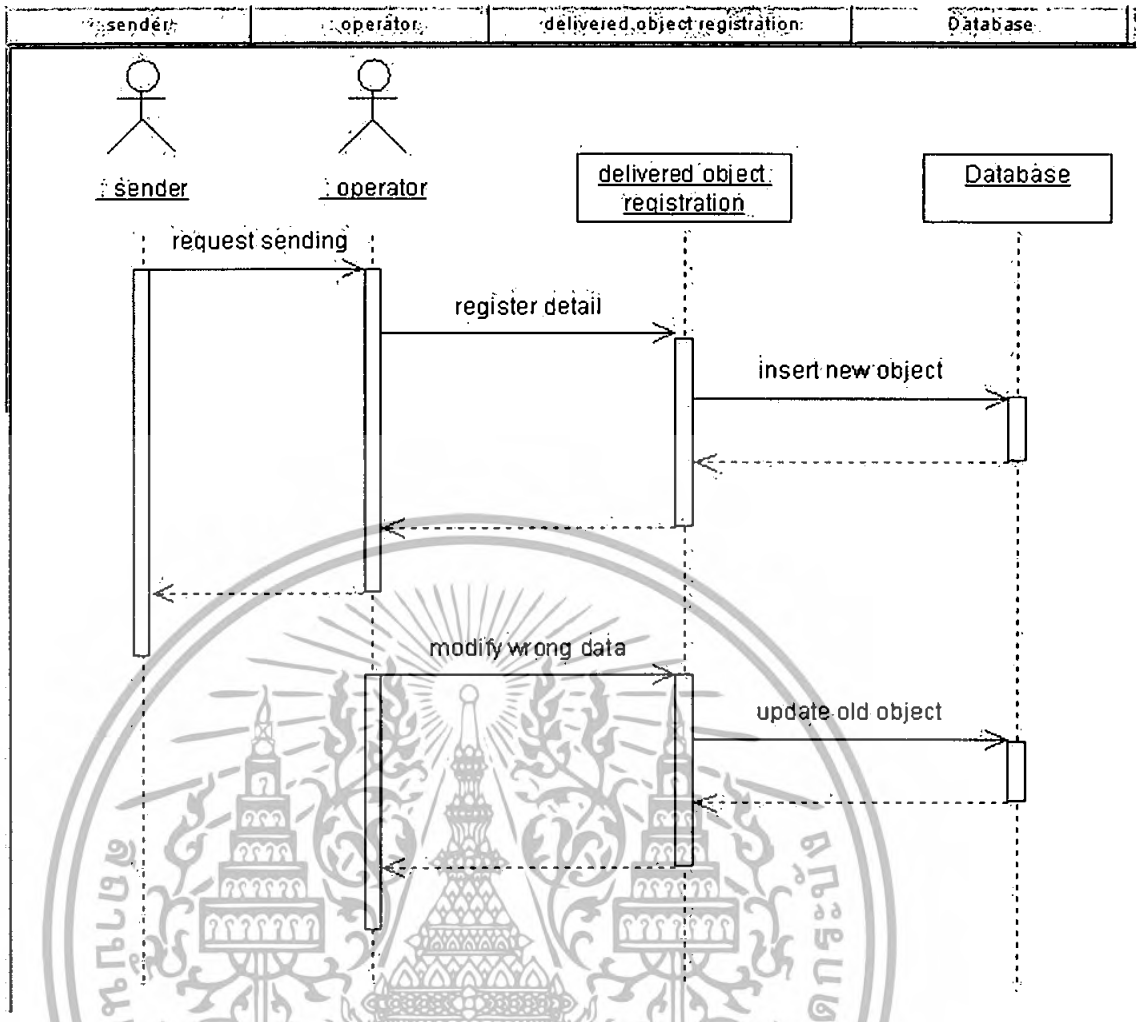
เมื่อผู้ขนส่งจะส่งสิ่งของจากสถานีย่อยไปถึงมือผู้รับก็ต้องใช้อัลกอริทึมเอสตาร์ในการหาเส้นทางสั้นที่สุด นำสิ่งของส่งถึงมือผู้รับ แล้วรายงานผลการรับสิ่งของ อัพเดทลงฐานข้อมูล




**รูปที่ 3.3** แสดง Use Case Diagram ของการตรวจสอบและติดตามสิ่งของ

การตรวจสอบและติดตามสิ่งของนั้น เจ้าหน้าที่สามารถกำหนดสิทธิ์ในการติดตามสิ่งของให้แก่ผู้ที่ต้องการตรวจสอบทั้งหมดได้ ไม่ว่าจะเป็นผู้ส่ง ผู้รับ ผู้ขนส่ง ผู้ขับรถขนส่ง หรือผู้ที่เกี่ยวข้องกับสิ่งของนั้นๆ อีกทั้งยังสามารถตรวจสอบสถานะของรถขนส่งได้ด้วย

ผู้ที่ต้องการตรวจสอบสิ่งของต้องทราบหมายเลข หรือมีสิทธิ์การเข้าถึงข้อมูลจากเจ้าหน้าที่ ดูสถานะการขนส่งที่ดูแลโดยระบบ




**รูปที่ 3.4** แสดง Sequence diagram ของ Register delivered object

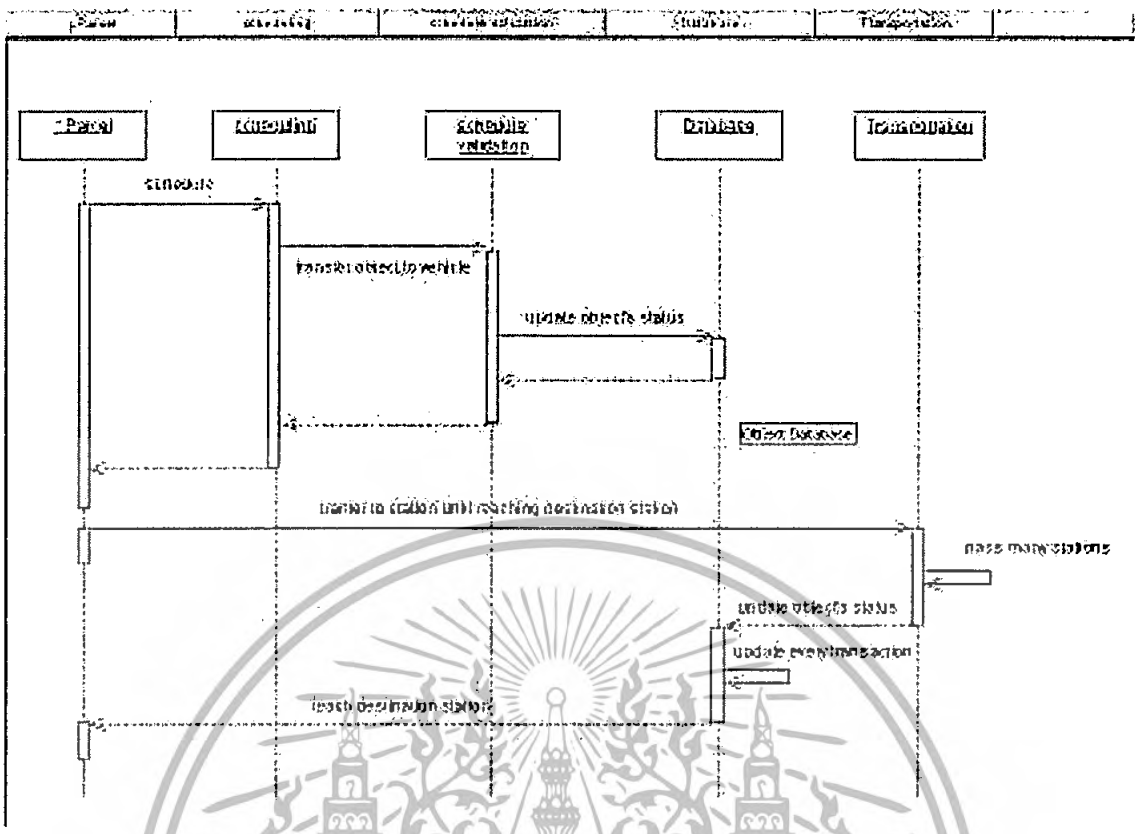
1 เมื่อผู้ส่งต้องการส่งสิ่งของ ก็บอกรายละเอียดแก่พนักงาน

2 พนักงานกรอกรายละเอียดทั้งหมดที่ต้องการใช้ในระบบขนส่ง พร้อมทั้งติดแท็ก RFID ให้เรียบร้อย

3 เพิ่มข้อมูลลงไปในฐานข้อมูลพร้อมทั้งให้หมายเลขในการติดตามสิ่งของแก่ผู้ส่งด้วย

4 หากมีการพบว่าข้อมูลผิดพลาดก่อนเริ่มการขนส่ง พนักงานสามารถแก้ไขข้อมูลได้

5 เมื่อมีการแก้ไขข้อมูล ต้องอัพเดทฐานข้อมูลทุกครั้ง



รูปที่ 3.5 แสดง Sequence diagram ของ Delivery

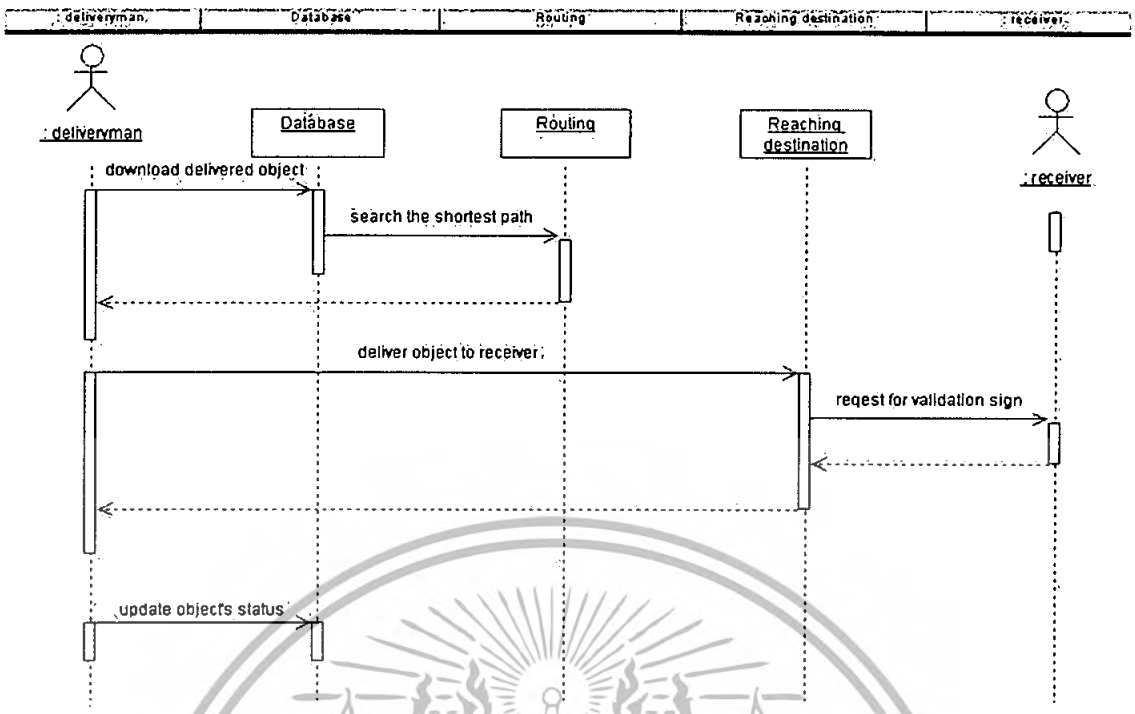
1 เมื่อจะขนส่งสิ่งของต้องจัดการเรียงลำดับการขนส่งให้เหมาะสม พร้อมทั้งจัดการว่าจะขนโดยรถคันไหน

2 ขนของขึ้นรถตามที่จัดการไว้

3 ตรวจสอบความถูกต้องตามที่จัดไว้ แล้วอัพเดทฐานข้อมูลด้วย

4 ขับรถขนส่งไปตามเส้นทางที่รับผิดชอบ ขนส่งสิ่งของให้สถานีที่รับผิดชอบให้ถูกต้อง

5 อัพเดทฐานข้อมูลทุกครั้งที่สิ่งของถึงสถานีปลายทางที่รับผิดชอบ



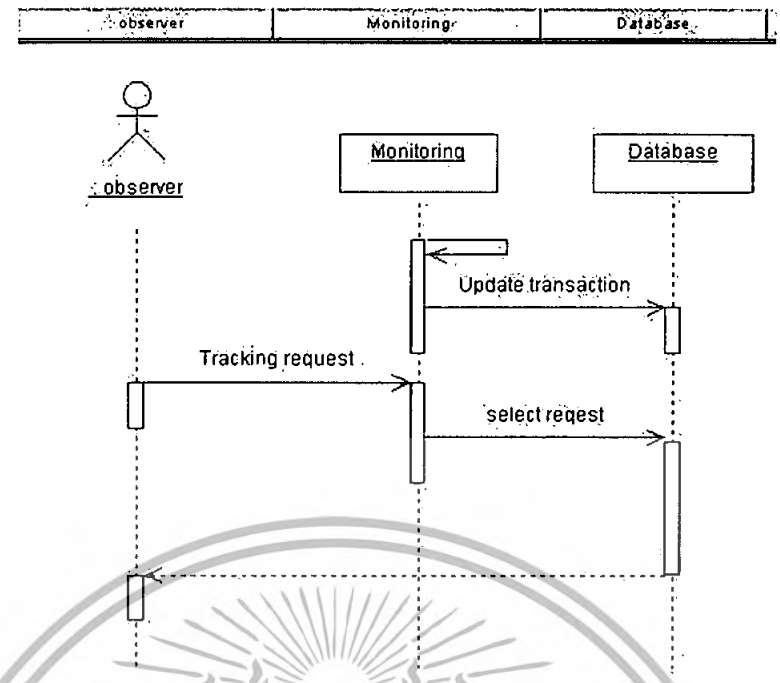

รูปที่ 3.6 แสดง Sequence diagram ของ Hand In

1 เมื่อผู้ขนส่งจะดูรายการที่ต้องขนส่งที่อยู่ในสถานีที่รับผิดชอบ ก็เลือกรายการที่ต้องขนส่ง

2 ทำการหาเส้นทางที่สั้นที่สุดในการขนส่งให้ผู้ขนส่ง

3 ผู้ขนส่งนำสิ่งของไปส่งจนถึงมือผู้รับตามที่กำหนด แล้วให้ผู้รับเซ็นรับสิ่งของให้เรียบร้อย

4 ผู้ขนส่งต้องมารายงานผลการขนส่ง อัพเดทในฐานข้อมูลทั้งหมด




รูปที่ 3.7 แสดง Sequence diagram ของ Track the delivered object

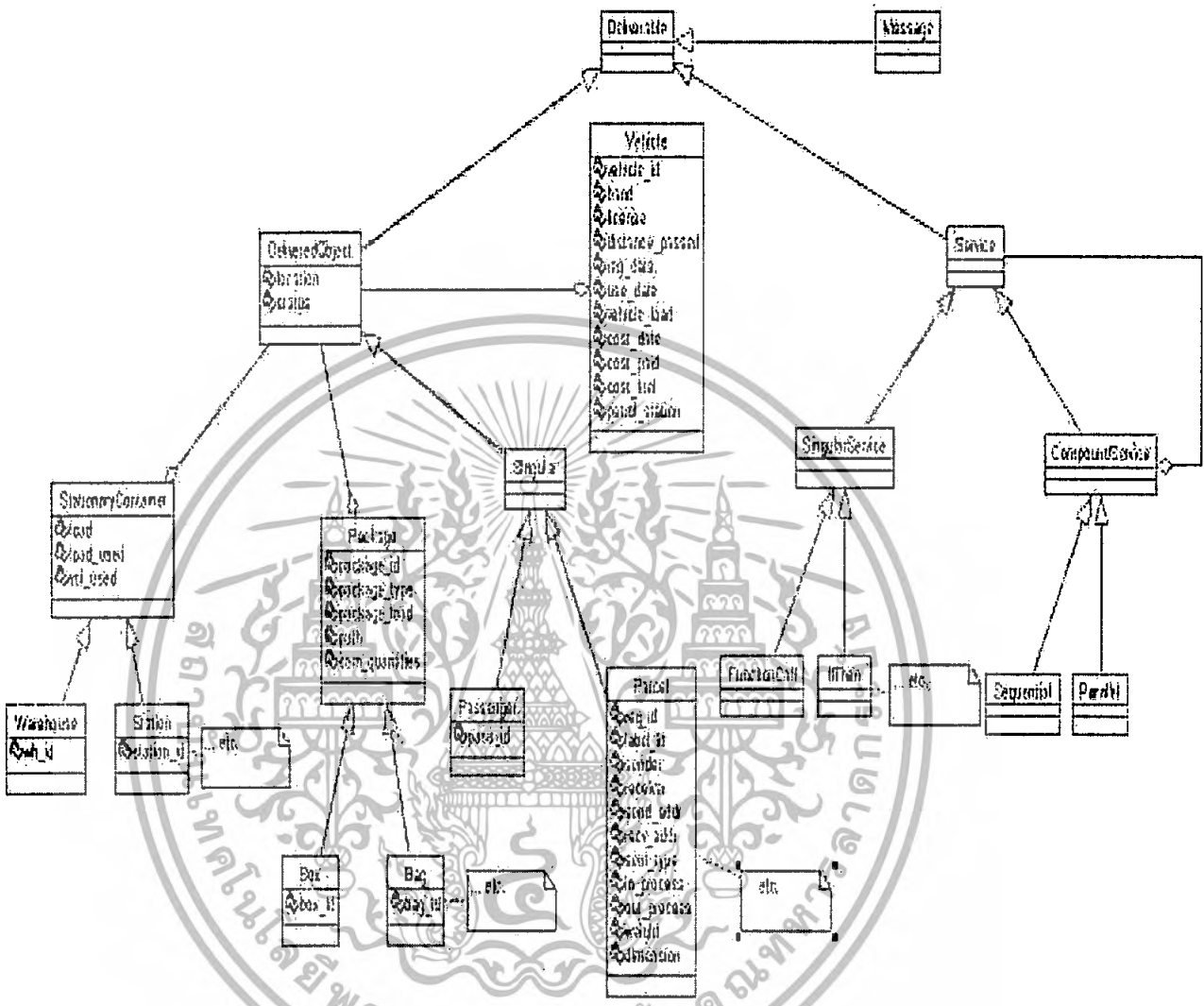
1 เมื่อมีการเปลี่ยนแปลงสถานะการขนส่ง ต้องมีการอัพเดทข้อมูลทุกครั้ง

2 เมื่อผู้ติดตามต้องการรู้สถานะการขนส่ง ต้องมีหมายเลขที่ออกให้เพื่อใช้ในการติดตาม

3 นำหมายเลขนั้นเพื่อตรวจสอบข้อมูลในฐานข้อมูล

### 3.2.2 การออกแบบคลาสไดอะแกรม

Class Diagram ที่ออกแบบแสดงได้ดังรูป




รูปที่ 3.8 แสดง Class Diagram ของระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี

สิ่งที่ขนส่งได้ในระบบเรียกว่า Deliverable ซึ่งเป็นได้ทั้งที่เป็นวัตถุ (DeliveredObject) และการบริการต่างๆ (Service) ในที่นี้จะเน้นที่ DeliveredObject ซึ่งเป็นวัตถุเดี่ยว (Singular) แบ่งเป็นคน (Passenger) และสิ่งของ (Parcel) โดยวัตถุที่ขนส่งได้นั้นจะถูกบรรจุอยู่ได้หลายสิ่งทั้งที่เคลื่อนที่ได้ (Vehicle) และเคลื่อนที่ไม่ได้(StationaryContainer) อีกทั้งวัตถุที่สามารถเป็นสิ่งของที่ถูกขนส่งแต่สามารถบรรจุสิ่งของอื่นได้ (Package)

เราได้ใช้ PostgreSQL เป็น DBMS ในการพัฒนาฐานข้อมูลและนำไปออกแบบตารางฐานข้อมูลดังนี้

**ตาราง Parcel** เก็บข้อมูลของสิ่งของที่รับเข้ามา

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Reg_ID | text | NOT NULL | | Alter Drop | Primary key of Parcel table incresingly |
| Label_ID | text | NOT NULL | | Alter Drop | Tracking code or RFID Tag |
| Sender | text | NOT NULL | | Alter Drop | |
| Send_Addr | text | | | Alter Drop | |
| Receiver | text | NOT NULL | | Alter Drop | |
| Recv_Addr | text | NOT NULL | | Alter Drop | |
| Sent_Type | text | NOT NULL | | Alter Drop | |
| Weight | integer | NOT NULL | | Alter Drop | |
| Dimension | text | | | Alter Drop | width x length x height |
| Postcode | integer | NOT NULL | | Alter Drop | |
| Comment | text | | | Alter Drop | |
| Send_Email | text | | | Alter Drop | |
| Send_Tel | text | | | Alter Drop | |
| Recv_Email | text | | | Alter Drop | |
| Recv_Tel | text | | | Alter Drop | |

**รูปที่ 3.9** แสดงตาราง Parcel

**ตาราง Parcel_Status** เก็บข้อมูลของสิ่งของที่ทำการจัดส่งซึ่งข้อมูลมีการเปลี่ยนแปลงไปตามกระบวนการจัดส่ง

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Reg_ID | text | NOT NULL | | Alter Drop | fk from Parcel |
| Status | text | NOT NULL | | Alter Drop | |
| In_Process | timestamp with time zone | NOT NULL | | Alter Drop | date & time |
| Out_Process | timestamp with time zone | | | Alter Drop | date & time |
| Staff | text | | | Alter Drop | |
| Package | text | | | Alter Drop | |
| Location | integer | NOT NULL | | Alter Drop | |
| Car_ID | integer | | | Alter Drop | |

**รูปที่ 3.10** แสดงตาราง Parcel_Status

**ตาราง Car** เก็บข้อมูลของรถที่ทำหน้าที่จัดส่งสิ่งของ

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Car_ID | integer | NOT NULL | | Alter Drop | |
| Load | integer | | | Alter Drop | |
| Car_Status | text | | | Alter Drop | |
| Route | text | | | Alter Drop | |
| Arrive | text | | | Alter Drop | |

**รูปที่ 3.11** แสดงตาราง Car

**ตาราง Address** เก็บข้อมูลของที่อยู่ปลายทางที่จะทำการจัดส่งสิ่งของ

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Seq | integer | NOT NULL | | Alter Drop | |
| Home_number | text | | | Alter Drop | |
| Address | text | | | Alter Drop | |
| Postcode | integer | | | Alter Drop | |
| x | text | | | Alter Drop | |
| y | text | | | Alter Drop | |

**รูปที่ 3.12** แสดงตาราง Address

**ตาราง Postcode** เก็บข้อมูลรหัสไปรษณีย์ของแต่ละเขต

| Column | Type | Not Null | Default | Actions | Comment |
|---|---|---|---|---|---|
| Postcode | integer | NOT NULL | | Alter Drop | |
| Province | text | | | Alter Drop | |
| District | text | | | Alter Drop | |

**รูปที่ 3.13** แสดงตาราง Postcode

# บทที่ 4
# การดำเนินงาน

## 4.1 การติดต่อกับอุปกรณ์อาร์เอฟไอดี

ในการพัฒนาระบบโปรแกรมนี้ใช้ผลิตภัณฑ์ของ Intermec ดังนี้

**ตารางที่ 4.1** แสดงอุปกรณ์อาร์เอฟไอดีที่ใช้ในระบบ

| ผลิตภัณฑ์ | ข้อมูลจำเพาะ |
|---|---|
| 700 Series Color Mobile Computer : 761B | Microprocessor : Intel® XScale™ Processor, 400 MHz<br>RAM Memory : 128MB<br>Flash ROM: 64 MB; includes ROM folder for application storage<br>Operating System : Windows Mobile® 2003 |
| Handheld Readers : IP4 | Feature:<br>• Multi-protocol radio for worldwide use<br>• Available for UHF frequency bands worldwide<br>• Infrared data connection to 700 Color |
| Fixed Reader : IP61 | Feature:<br>• Intel® Celeron® processor and ample storage to run complex RFID applications<br>• Hosts applications written in Java®, Java Script or C# .Net<br>• "Store and forward" capabilities ensure data won't be lost<br>• Directly monitors and controls presence detectors and signal lights<br>• Localized workbench to load, edit and run Java Script directly on the reader<br>• Based on EPC global certified radio<br>• Available in 865MHz, 869MHz or 915MHz frequency bands |

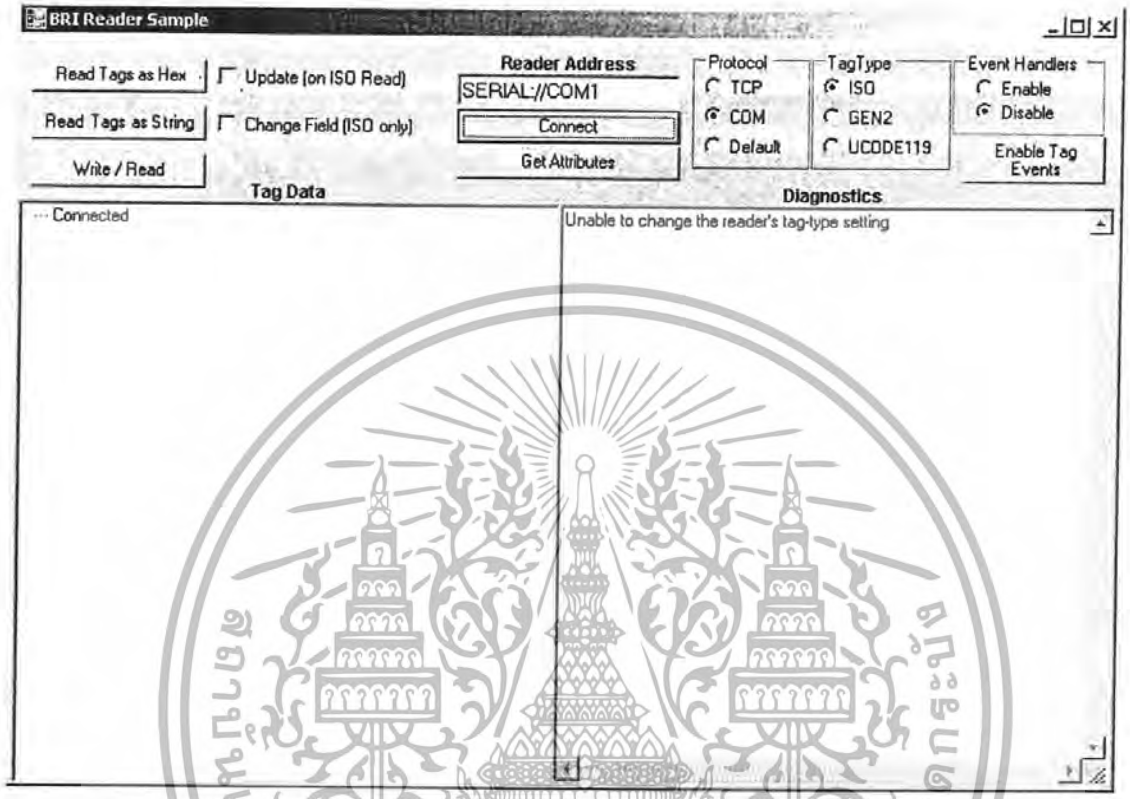
ในการติดต่อกับ RFID Reader ซึ่งในการติดต่อกับ RFID Reader นี้แบ่งการทำงานได้ 2 รูปแบบคือ

1. RFID Reader แบบ Handheld (IP4) ในการติดต่อ Reader แบบนี้จะทำงานในลักษณะ Client-Server โดยตัว Reader จะทำหน้าที่เป็น Client ส่งข้อมูลของแท็กที่อ่านได้ไปที่ Server ซึ่ง Server และ Client นั้นติดต่อกันผ่าน wireless

**รูปที่ 4.1** แสดงการติดต่อระหว่าง Server และ Client (Handheld RFID Reader) ผ่าน wireless

**รูปที่ 4.2** แสดงทดลองการอ่านแท็กด้วย Reader แบบ Handheld และส่งข้อมูลเข้าเครื่อง Server

2. RFID Reader แบบ Fixed (IF61) การติดต่อ Reader จะติดต่อผ่านทางพอร์ตอนุกรม (Serial port)




**รูปที่ 4.3** แสดงการทดลองอ่านแท็กด้วย Reader แบบ Fixed

## 4.2 การจัดการฐานข้อมูล PostgreSQL

ในการพัฒนาระบบได้พัฒนาโมดูลในการติดต่อฐานข้อมูล PostgreSQL 8.1 โดยใช้ PostgreSQL OLE DB Provider เป็นตัวติดต่อซึ่งเขียนเป็นฟังก์ชันให้เรียกใช้งานในภาษาไอรอนไพธอน โดยมีฟังก์ชันในการติดต่อไปยังเซิร์ฟเวอร์ฐานข้อมูลที่เครื่อง 161.246.5.68

```
import clr
import System
clr.AddReference("System.Data")
from System.Data import *
from System.Data.OleDb import *

conString = """Provider=PostgreSQL OLE DB
    Provider;Password=ictlab;User ID=postgres;Data
    Source=161.246.5.68;Location=t1"""
con = OleDbConnection(conString)
```

ฟังก์ชันหลักสำหรับการใช้งานในการติดต่อกับฐานข้อมูล คือ Select , Insert , Delete และ Update มีรายละเอียดดังนี้

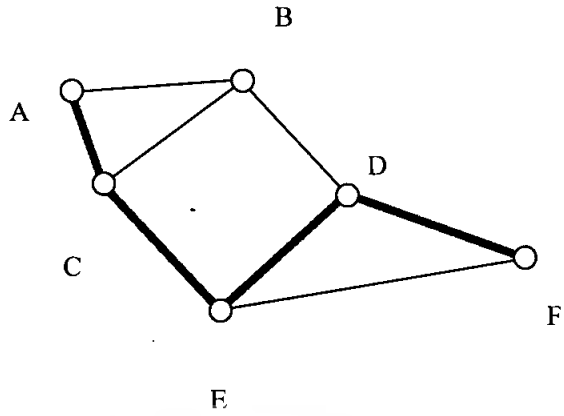
```
def SelectDT(tableName,colName = "*",condition="",dtName = '0')
def InsertRow(tableName,vals)
def InsertColumn(tableName,colName,vals)
def DeleteDT(tableName,condition)
def UpdateDT(tableName,colName,vals,condition)

# tableName : String name of table in DBMS
# colName   : String of columns name separate with comma
            : "id,name,addr"
            ->default "*" every columns
# condition : String of every selected conditions (optional)
            : "where id = 1 order by id"
            ->default is no condition
# dtName    : String name of DataTable
            ->defaule "0"
# vals      : List of string values of each columns separate by comma
              values order by columns of table
            : ['\'string'\','integer']
```

## 4.3 อัลกอริทึมในการค้นหาเส้นทางส่งสิ่งของ

ในการค้นหาเส้นทางส่งสิ่งของจะใช้อัลกอริทึมเอสตาร์มาใช้ในการค้นหาเส้นทางที่สั้นที่สุดในการขนส่งสิ่งของ ซึ่งมีอัลกอริทึมดังนี้

โดยทั่วไปเอสตาร์จะหาเส้นทางที่สั้นที่สุดจากจุด 2 จุด จุดต้นไปยังจุดปลาย แต่ในระบบนี้ต้องการกำหนดให้สถานีย่อยที่ต้องนำส่งสิ่งของเป็นจุดที่ต้องผ่านหลายจุด ซึ่งในการค้นหาแบบมีจุดผ่านหลายจุดนั้น ไม่เพียงแค่จุดสุดท้ายที่ตรวจสอบจะต้องเป็นจุดเป้าหมายเท่านั้น ในเส้นทางที่เราตรวจสอบอยู่จะต้องเป็นเส้นทางที่ผ่านจุดที่ต้องการด้วย ซึ่งเราสามารถอธิบายด้วยทฤษฎีเซตทางคณิตศาสตร์ได้ คือเราจะเสร็จสิ้นการค้นหาก็ต่อเมื่อ เซตของจุดหมาย(เซตของจุดที่ผ่านรวมกับจุดหมาย)เป็นสับเซตของเส้นทางที่ตรวจสอบอยู่ในขณะนั้น และจุดสุดท้ายที่ตรวจสอบเป็นจุดเป้าหมาย

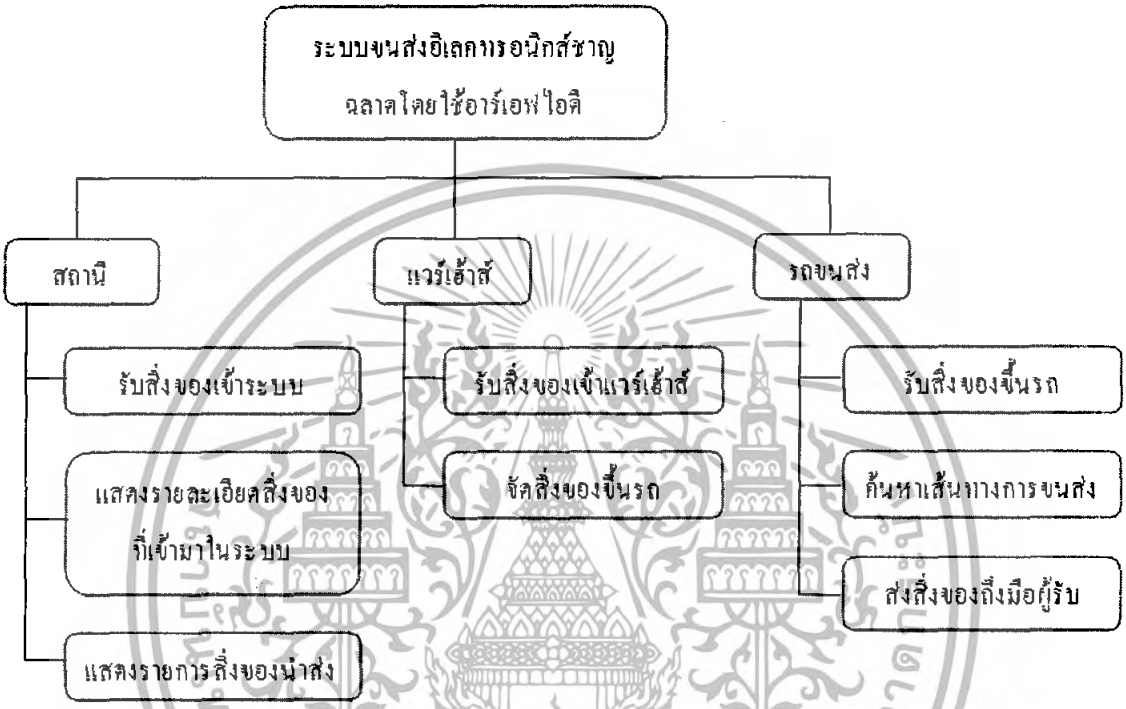



**รูปที่ 4.4** แสดงการใช้อัลกอริทึมเอสตาร์ในการค้นหาเส้นเมื่อกำหนดให้ผ่านหลายจุด

จากรูปข้างต้น เราต้องการเดินทางจากจุด A ไปยังจุด F โดยกำหนดให้ผ่านจุด E และจุด D ด้วย เราจะแทนข้อมูลในรูปของเซตได้ คือให้ S แทนเซตของจุดเริ่มต้น ให้ G คือเซตของจุดเป้าหมาย และ P แทนเซตของจุดเส้นทางที่ตรวจสอบ ดังนี้ S = {A},G={E,D,F}และ P={A,C,E,D,F} จะเห็นได้ว่าเซต G เป็นสับเซตของเซต P และจุดสุดท้ายที่ตรวจสอบเป็นจุดเป้าหมาย

# บทที่ 5
# ผลการทำงานของระบบ

ระบบที่เราได้ออกแบบมีฟังก์ชันการทำงานดังนี้



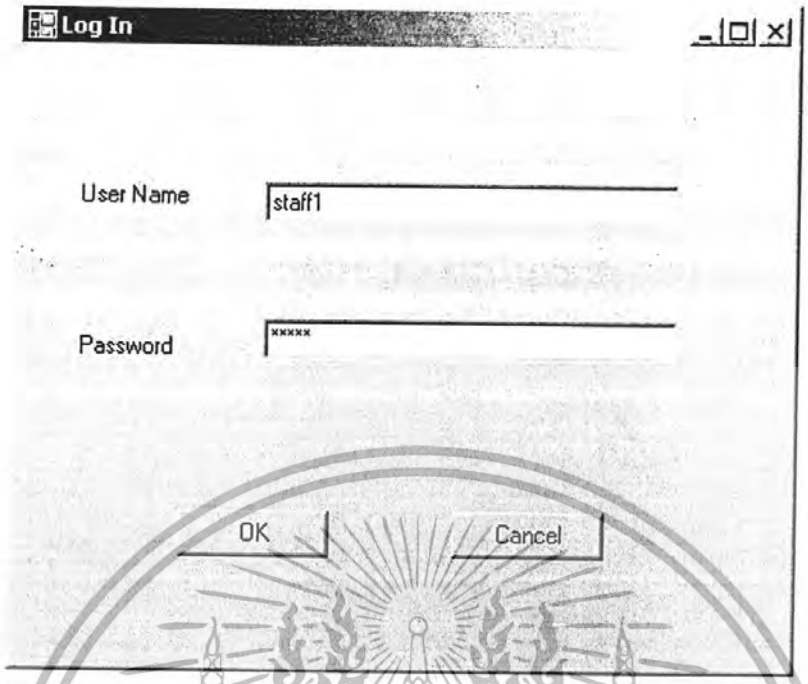

**รูปที่ 5.1** แสดงฟังก์ชันการทำงานโดยรวมของระบบ

เมื่อพัฒนาเป็นแอพพลิเคชั่นตามระบบที่ออกแบบไว้ สามารถแบ่งแอพพลิเคชั่นการทำงานต่างๆได้ดังนี้

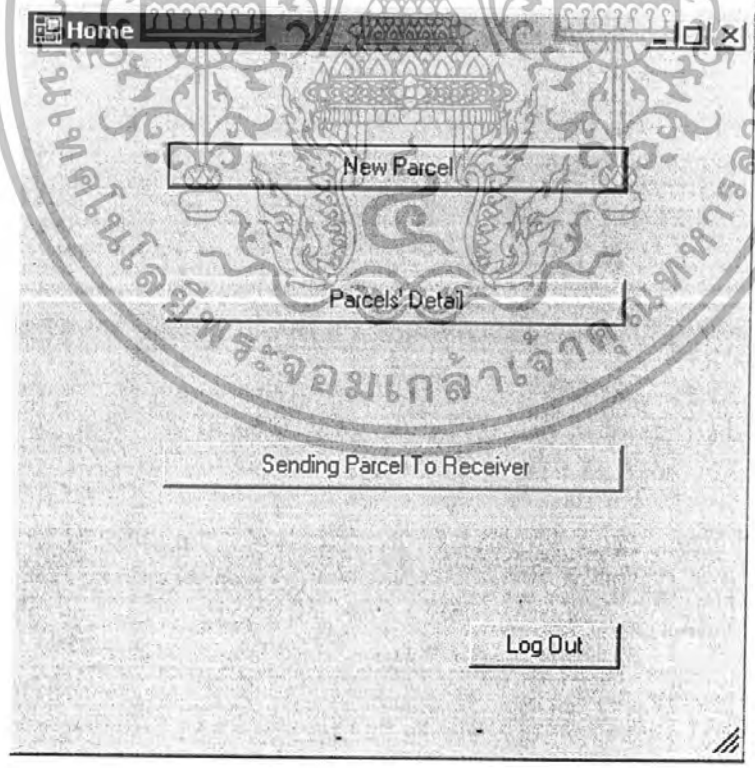
## 5.1 แอพพลิเคชั่นในส่วนของสถานี

แอพพลิเคชั่นในส่วนของสถานีเป็นแอพพลิเคชั่นที่มีประจำอยู่ในทุกสถานีย่อยที่ทำหน้าที่ในการรับสิ่งของจากผู้ใช้งานเข้ามาในระบบ และทำการจัดส่งสิ่งของไปยังผู้รับ ผู้ใช้งานแอพพลิเคชั่นในส่วนนี้คือเจ้าหน้าที่ที่ประจำแต่ละสถานี

ในหน้าแรกของแอพพลิเคชั่นจะเป็นหน้าจอการล็อกอินเพื่อเข้ามาใช้งานในระบบ ใช้งานโดยการกรอก Username และ Password ลงไปเพื่อทำการล็อกอินเข้าสู่ระบบ




**รูปที่ 5.2** แสดงแอพพลิเคชั่นในส่วนของสถานี : หน้าล็อกอินเข้าใช้งาน

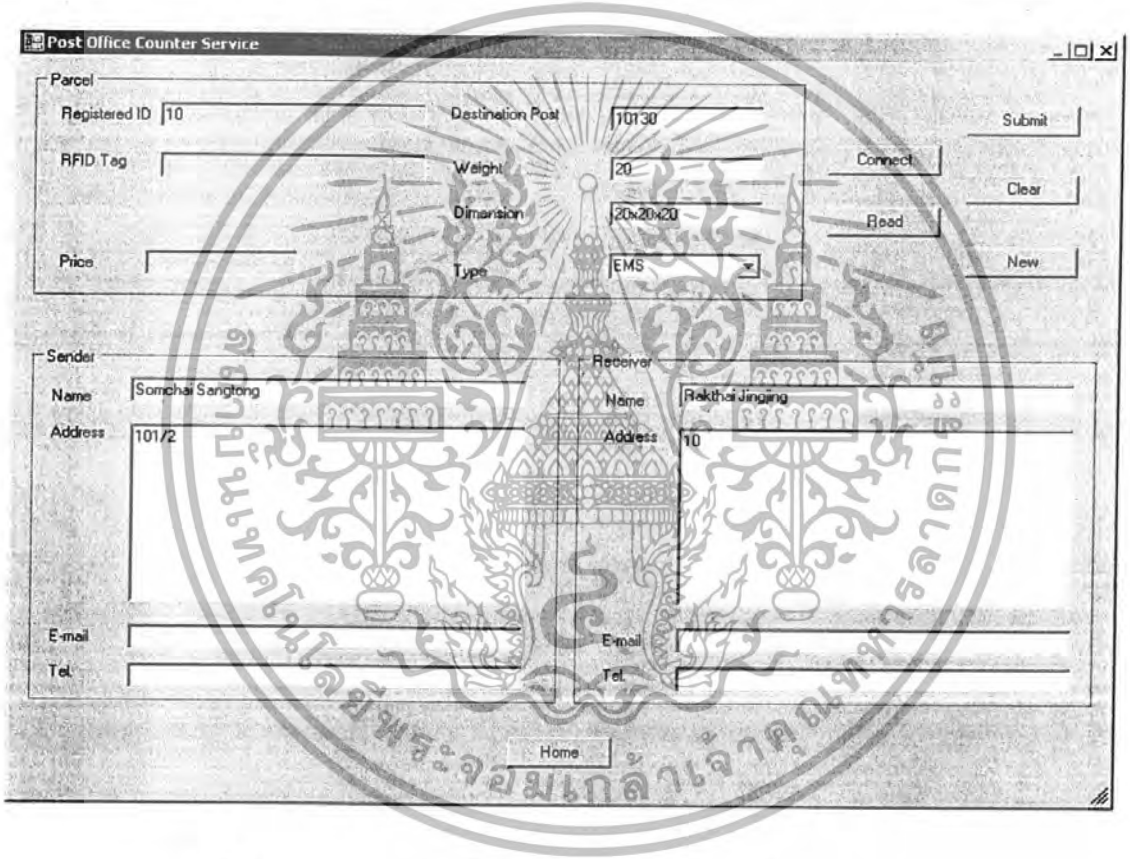



**รูปที่ 5.3** แสดงแอพพลิเคชั่นในส่วนของสถานี : เลือกการทำงาน

เมื่อผ่านหน้าล็อกอินเข้ามาจะเข้าสู่หน้าจอที่แสดงโหมดการทำงานหลักของแอพพลิเคชั่นในส่วนของสถานีซึ่งมีการทำงาน 3 ส่วนประกอบด้วย

1. รับสิ่งของเข้าระบบ
2. แสดงสถานะของสิ่งของที่ได้รับภายในสถานีนั้นๆ
3. แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ

**1. รับสิ่งของเข้าระบบ**




**รูปที่ 5.4** แสดงแอพพลิเคชั่นในส่วนของสถานี : รับสิ่งของเข้าระบบ

ในการทำงานของแอพพลิเคชั่นในส่วนของสถานี หน้ารับสิ่งของเข้าระบบเป็นส่วนที่ทำหน้าที่เก็บรายละเอียดต่างๆของสิ่งของเพื่อนำสิ่งของเข้ามาในระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีโดย หน้าจอจอรับสิ่งของเข้าระบบมีรายละเอียดดังนี้

1. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่ในช่อง RFID Tag

2. ปุ่ม Read อ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลเป็นรหัสของแท็กอาร์เอฟไอดีที่อ่านได้ในช่อง RFID Tag

3. ปุ่ม Summit เก็บรายละเอียดของสิ่งของลงในฐานข้อมูล ข้อมูลของสิ่งของจะถูกบันทึกลงในฐานข้อมูลโดยมีสถานะของการส่งสิ่งของเป็น 'Get in system'

4. ปุ่ม Clear ลบรายละเอียดทั้งหมดที่กรอกภายในหน้าจอรับสิ่งของนี้

5. ปุ่ม New เริ่มต้นรับสิ่งของชิ้นใหม่

6. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ

7. Textbox Registered ID เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นรหัสที่สร้างขึ้นมาเพื่อใช้ในระบบ แทนที่จะใช้รหัสของแท็กอาร์เอฟไอดีตรงๆ เพื่อความปลอดภัยของระบบ

8. Textbox RFID Tag เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นรหัสที่อ่านได้จากแท็กอาร์เอฟไอดี

9. Textbox Price เจ้าหน้าที่ไม่สามารถกรอกเองได้ เป็นราคาค่าบริการในการจัดส่งสิ่งของโดยคำนวนจากน้ำหนักของสิ่งของและประเภทของการส่งสิ่งของ

10. Textbox Destination Post เจ้าหน้าที่กรอกรหัสไปรษณีย์ของสถานีย่อยที่อยู่ของผู้รับปลายทาง

11. Textbox Weight เจ้าหน้าที่กรอกน้ำหนักของสิ่งของ

12. Textbox Dimension เจ้าหน้าที่กรอกปริมาตรของสิ่งของ

13. ComboBox Type เจ้าหน้าที่เลือกชนิดของการจัดส่งสิ่งของ (EMS หรือ Ordinary)

14. Textbox Sender Name เจ้าหน้าที่กรอกชื่อ นามสกุลของผู้ส่ง

15. Textbox Sender Address เจ้าหน้าที่กรอกรายละเอียดที่อยู่ของผู้ส่ง

16. Textbox Sender E-mail เจ้าหน้าที่กรอก E-mail ของผู้ส่ง

17. Textbox Sender Tel. เจ้าหน้าที่กรอกเบอร์โทรศัพท์ของผู้ส่ง

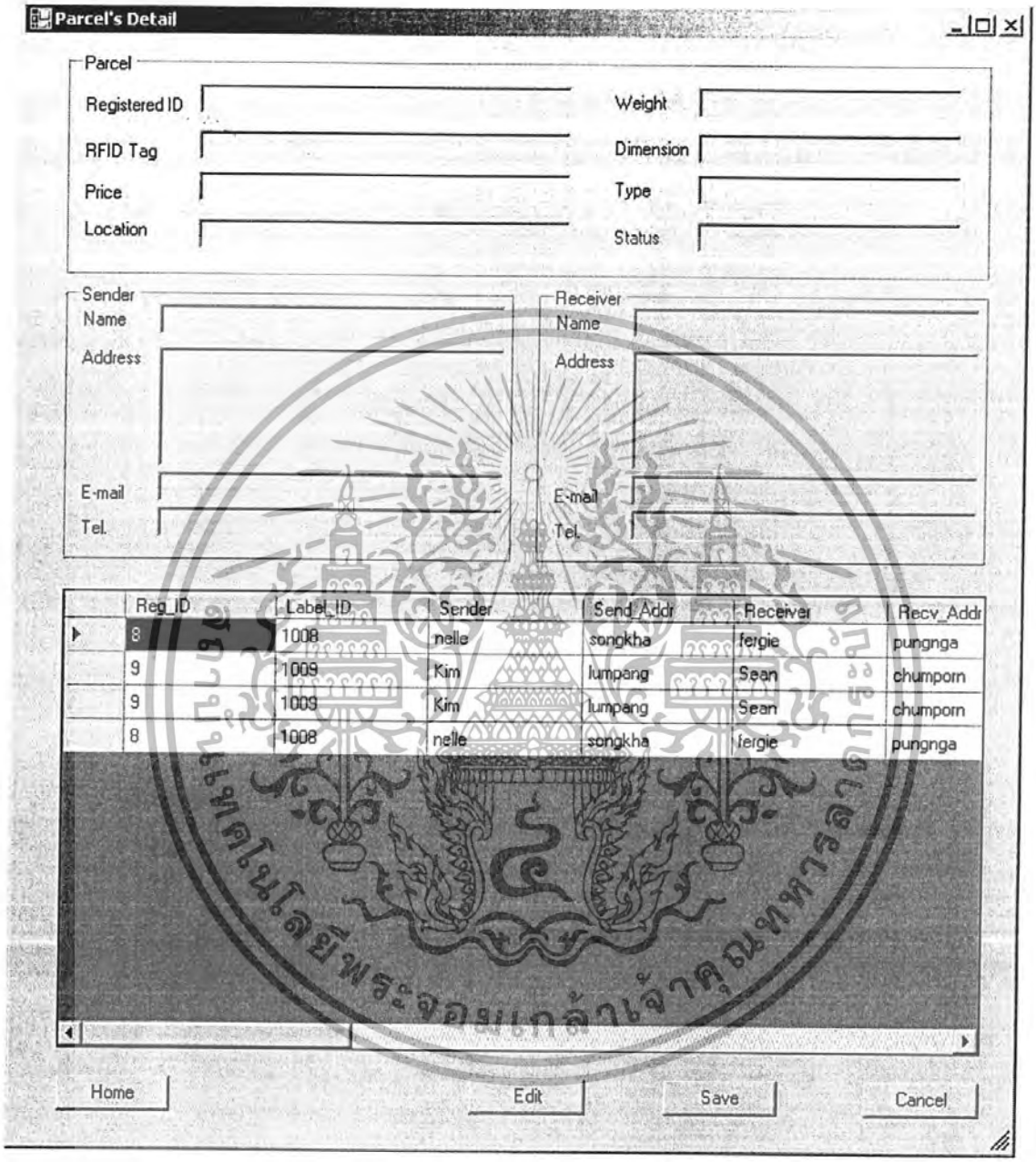
18. Textbox Receiver Name เจ้าหน้าที่กรอกชื่อ นามสกุลของผู้รับ

19. Textbox Receiver Address เจ้าหน้าที่กรอกรายละเอียดที่อยู่ของผู้รับ

20. Textbox Receiver E-mail เจ้าหน้าที่กรอก E-mail ของผู้รับ

21. Textbox Receiver Tel. เจ้าหน้าที่กรอกเบอร์โทรศัพท์ของผู้รับ

**2. แสดงสถานะของสิ่งของที่ได้รับภายในสถานีนั้นๆ**




**รูปที่ 5.5** แสดงแอพพลิเคชั่นในส่วนของสถานี: สถานะของสิ่งของที่ได้รับภายในสถานี

หน้าแสดงรายละเอียดของสิ่งของที่รับของมาภายในสถานีนั้นๆโดยจะแสดงสถานะล่าสุดของสิ่งของชิ้นนั้นๆด้วยว่ามีสถานะการส่งสถานะอะไรและสิ่งของชิ้นนั้นอยู่ที่ไหน หน้าแสดงรายละเอียดของสิ่งของมีรายละเอียดดังนี้

1. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ

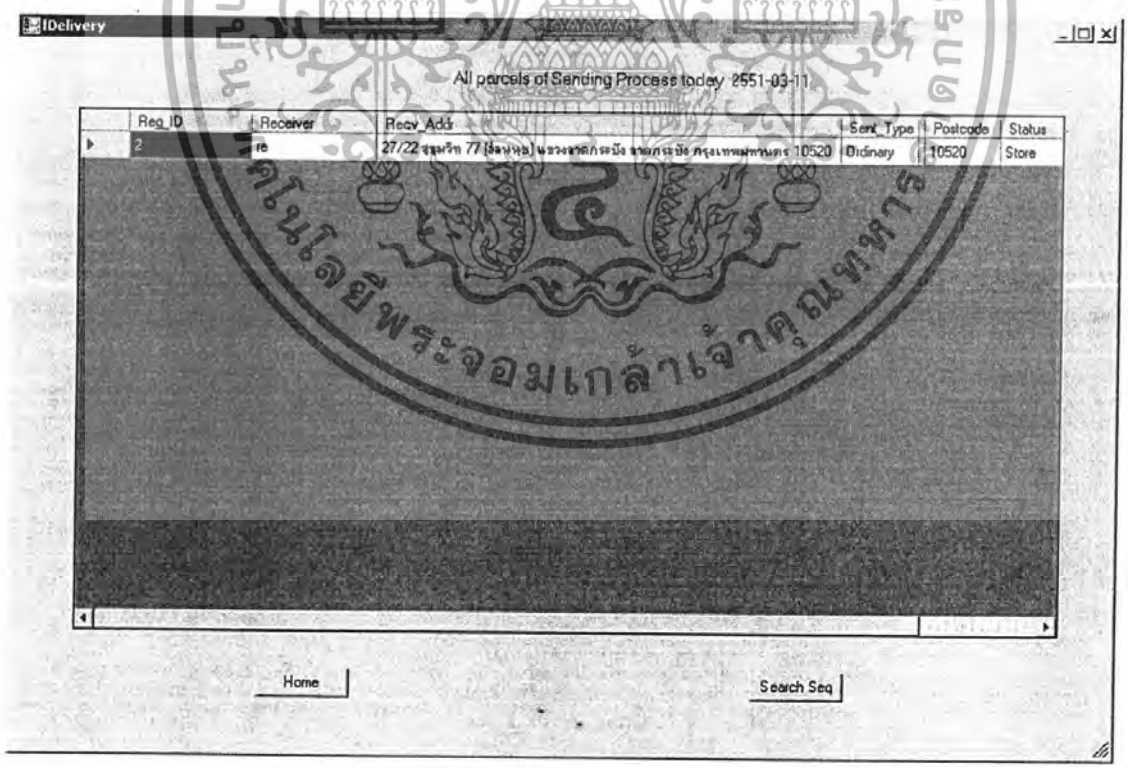
2. ปุ่ม Edit เจ้าหน้าที่สามารถกดเลือกสิ่งของที่ต้องการทำการแก้ไขโดยเลือกจากตารางที่แสดงผล (DataGridView) สิ่งของที่ถูกเลือกจะแสดงรายละเอียดใน Textbox ด้านบน จากนั้นกดปุ่ม Edit เพื่อเข้าไปทำการแก้ไขสิ่งของชิ้นนั้นๆ ปุ่ม Edit จะเปลี่ยนเป็นปุ่ม OK กดปุ่ม OK เมื่อทำการแก้ไขสิ่งของเสร็จเรียบร้อย สิ่งของที่ถูกแก้ไขจะแสดงรายละเอียดใหม่ในตาราง

3. ปุ่ม Save เจ้าหน้าที่กดเมื่อต้องการทำการบันทึกข้อมูลสิ่งของที่มีการแก้ไขลงในฐานข้อมูล

4. ปุ่ม Cancel กดเพื่อยกเลิกการเปลี่ยนแปลงใดๆ

5. DataGridview แสดงรายละเอียดของสิ่งของที่ได้รับจากผู้ส่งภายในสถานีนั้นๆ ซึ่งจะแสดงรายละเอียดและสถานะการส่งสิ่งของล่าสุดของสิ่งของชิ้นนั้นๆด้วย เจ้าหน้าที่สามารถเลือกกดที่สิ่งของเพื่อแก้ไขข้อมูลของสิ่งของได้ โดยจะแสดงรายละเอียดของสิ่งของและให้ทำการแก้ไขได้ในส่วนTextbox ด้านบน แต่การแก้ไขจะจำกัดเฉพาะสิ่งของที่ยังอยู่ภายในสถานีและยังไม่ถูกส่งต่อเข้ากระบวนการอื่นๆเท่านั้น

### 3. แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ




**รูปที่ 5.6** แสดงแอพพลิเคชั่นในส่วนของสถานี : แสดงรายการสิ่งของที่ต้องนำไปส่งให้ผู้รับ

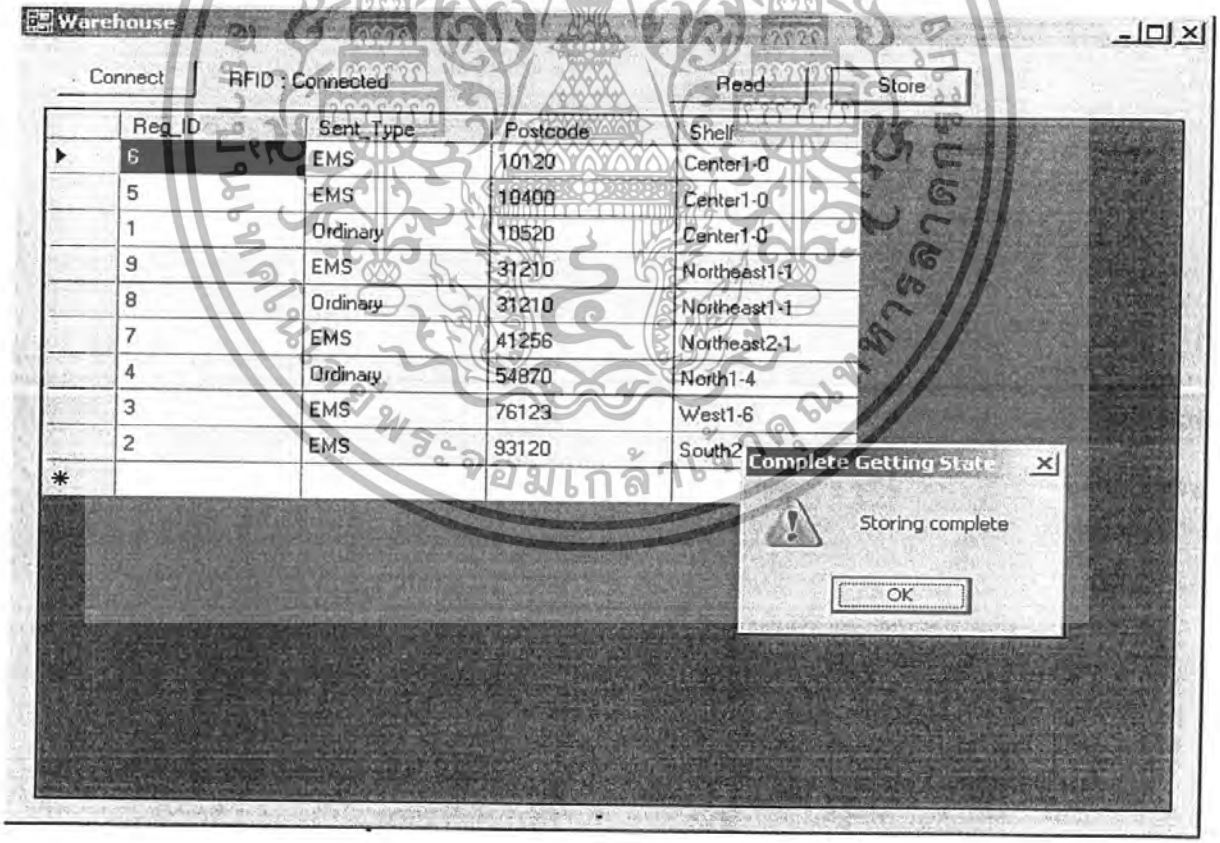
แอพพลิเคชั่นหน้าแสดงรายการส่งสิ่งของไปถึงมือผู้รับนี้ มีเพื่อแสดงรายละเอียดของสิ่งของที่ต้องนำไปส่งถึงมือผู้รับ มีรายละเอียดดังนี้

1. DataGridView แสดงรายละเอียดของสิ่งของที่ส่งถึงสถานีปลายทางนั้น
2. ปุ่ม Home กลับไปหน้าจอหลักเพื่อเลือกการทำงานอื่นๆ
3. ปุ่ม Search Seq. เจ้าหน้าที่กดเพื่อดูลำดับการส่งสิ่งของตามความเหมาะสมที่โปรแกรมคำนวณให้

## 5.2 แอพพลิเคชั่นในส่วนของแวร์เฮาส์

แอพพลิเคชั่นในส่วนของแวร์เฮ้าส์จะแบ่งการทำงานเป็น 2 ส่วนคือ

1. **รับสิ่งของเข้าแวร์เฮ้าส์** แอพพลิเคชั่นใช้สำหรับการอัพเดทข้อมูล และแบ่งจัดเก็บสิ่งของในแวร์เฮ้าส์โดยแบ่งตามภูมิภาค และรหัสไปรษณีย์เพื่อความสะดวกต่อการจัดส่งในขั้นถัดไป สิ่งของที่เข้ามาเก็บในแวร์เฮ้าส์จะมีสถานะของการส่งสิ่งของเป็น 'Store'




**รูปที่ 5.7** แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : รับสิ่งของเข้าแวร์เฮ้าส์

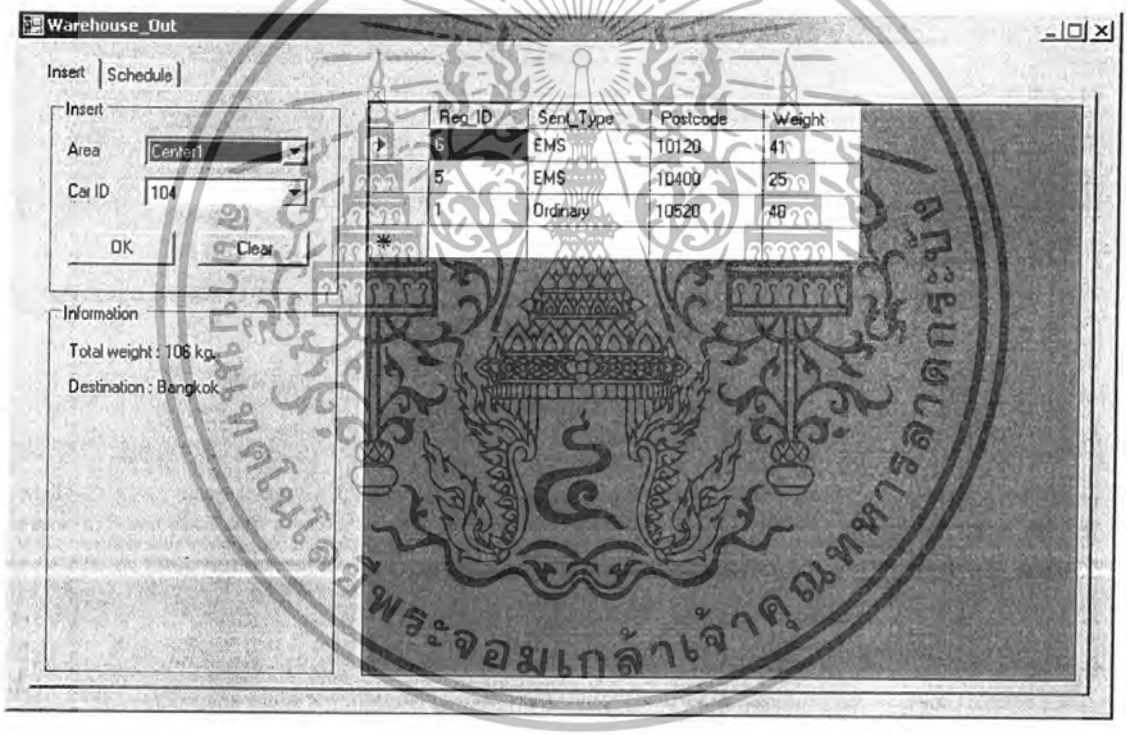
แอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ ที่ทำหน้าที่รับสิ่งของเข้า มีรายละเอียดดังนี้

1. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่

2. ปุ่ม Read อ่านแท็กอาร์เอฟไอดีจากนั้นจะนำรหัสที่อ่านได้จากแท็กอาร์เอฟไอดีไปดึงข้อมูลในฐานข้อมูลมาแสดงว่ามีสิ่งของอะไรบ้างที่เข้ามาในขั้นตอนการรับของเข้าแวร์เฮ้าส์

3. ปุ่ม Store อัพเดทข้อมูลว่าสิ่งของที่อ่านแท็กอาร์เอฟไอดีได้รับเข้ามาเก็บในแวร์เฮ้าส์แล้วโดยจะมีแบ่งการเก็บสิ่งของไปตามชั้นเก็บของตามภูมิภาคและรหัสไปรษณีย์

**2. จัดสิ่งของขึ้นรถขนส่ง** เป็นขั้นตอนที่จะนำสิ่งของขึ้นรถเพื่อส่งสิ่งของไปสถานีหลักของแต่ละภูมิภาค เพื่อทำการจัดส่งในขั้นถัดไป




**รูปที่ 5.8** แสดงแอพพลิเคชั่นในส่วนของแวร์เฮ้าส์ : จัดสิ่งของขึ้นรถ

รายละเอียดของแอพพลิเคชั่นในส่วนของการจัดสิ่งของขึ้นรถ มีดังนี้

1. ComboBox Area เจ้าหน้าที่ทำการเลือกภูมิภาคในการส่ง เมื่อเลือกภูมิภาคแล้ว DataGridView ด้านซ้ายจะแสดงข้อมูลของสิ่งของที่ต้องจัดส่งไปที่ภูมิภาคนั้นๆทั้งหมด โดยจะเรียงจากชนิดของการขนส่งโดยให้การส่งแบบ EMS มีความสำคัญสูงที่สุด จากนั้นจึงเรียงลำดับจากเวลาที่สิ่งของเข้ามาในระบบ ด้านซ้ายจะเป็นส่วนแสดงข้อมูลว่าน้ำหนักรวมของสิ่งของเป็นเท่าไหร่และปลายทางในการขนส่งอยู่ที่จังหวัดไหน

2. ComboBox Car_ID แสดงเบอร์รถขนส่งที่มีสถานะพร้อมจะส่งได้ กำหนดให้รถสามารถรองรับสิ่งของได้น้ำหนัก 1000 กิโลกรัม ไม่จำกัดปริมาตรไม่การขนส่ง

3. ปุ่ม OK เพื่อทำการจัดสิ่งของที่จะทำการขนส่งให้กับรถขนส่งตามเบอร์รถที่เลือกไป

4. ปุ่ม Clear เพื่อเริ่มต้นการจัดสิ่งของขึ้นรถใหม่

## 5.3 แอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของ

รถจัดส่งมี 2 ประเภท คือ รถประจำแวร์เฮาส์ และรถประจำสถานีย่อย หน้าที่โดยทั่วไปเหมือนกัน จะแตกต่างกันเล็กน้อย

รถประจำแวร์เฮาส์ รถจัดส่งสิ่งของจะเริ่มต้นการทำงานที่แวร์เฮ้าส์โดยการรับสิ่งของเข้ามาในรถจากนั้นจะทำการจัดส่งไปยังสถานีหลักของแต่ละภูมิภาคและจะกระจายไปยังสถานีของแต่ละจังหวัด ไปถึงสถานีย่อยปลายทางของสิ่งของในรถนั้นๆ

**รูปที่ 5.9** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Get in

รายละเอียดของแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของมีดังนี้

1. ComboBox Car_ID เพื่อเลือกว่ารถที่ต้องการจัดการเป็นเบอร์อะไร

2. ปุ่ม Connect ทำการติดต่อไปยังตัวอ่านแท็กอาร์เอฟไอดีโดยจะแสดงผลการติดต่อว่าสำเร็จหรือไม่

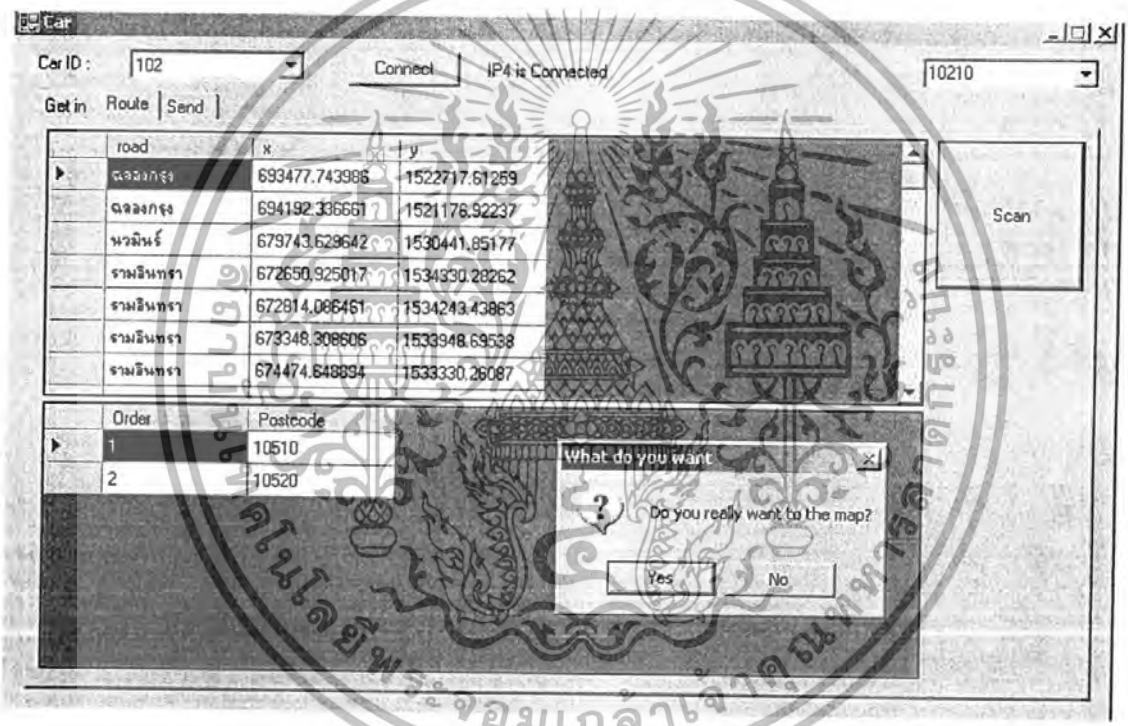
3. ปุ่ม Scan อ่านแท็กอาร์เอฟไอดีจากนั้นจะนำรหัสที่อ่านได้จากแท็กอาร์เอฟไอดีไปดึงข้อมูล

ในฐานข้อมูลมาแสดงว่ามีสิ่งของอะไรบ้างที่อ่านได้และจะแสดงผลที่ DataGridView โดยจะแสดงสถานะของสิ่งของแต่ละแถวตามสีซึ่งมีความหมายดังนี้

- สีน้ำเงิน เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจพบ
- สีเหลือง เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจ**ไม่**พบ
- สีแดง **ไม่**เป็นสิ่งของที่ต้องนำขึ้นรถและอ่านแท็กอาร์เอฟไอดีแล้วตรวจพบ

4. ปุ่ม Update ทำการอัพเดทสถานะของสิ่งของ ถ้านำสิ่งของเข้ารถจอัพเดทสถานะของสิ่งของเป็น 'Sending' แต่ถ้านำของออกจากรถจะอัพเดทสถานะเป็น 'Store'

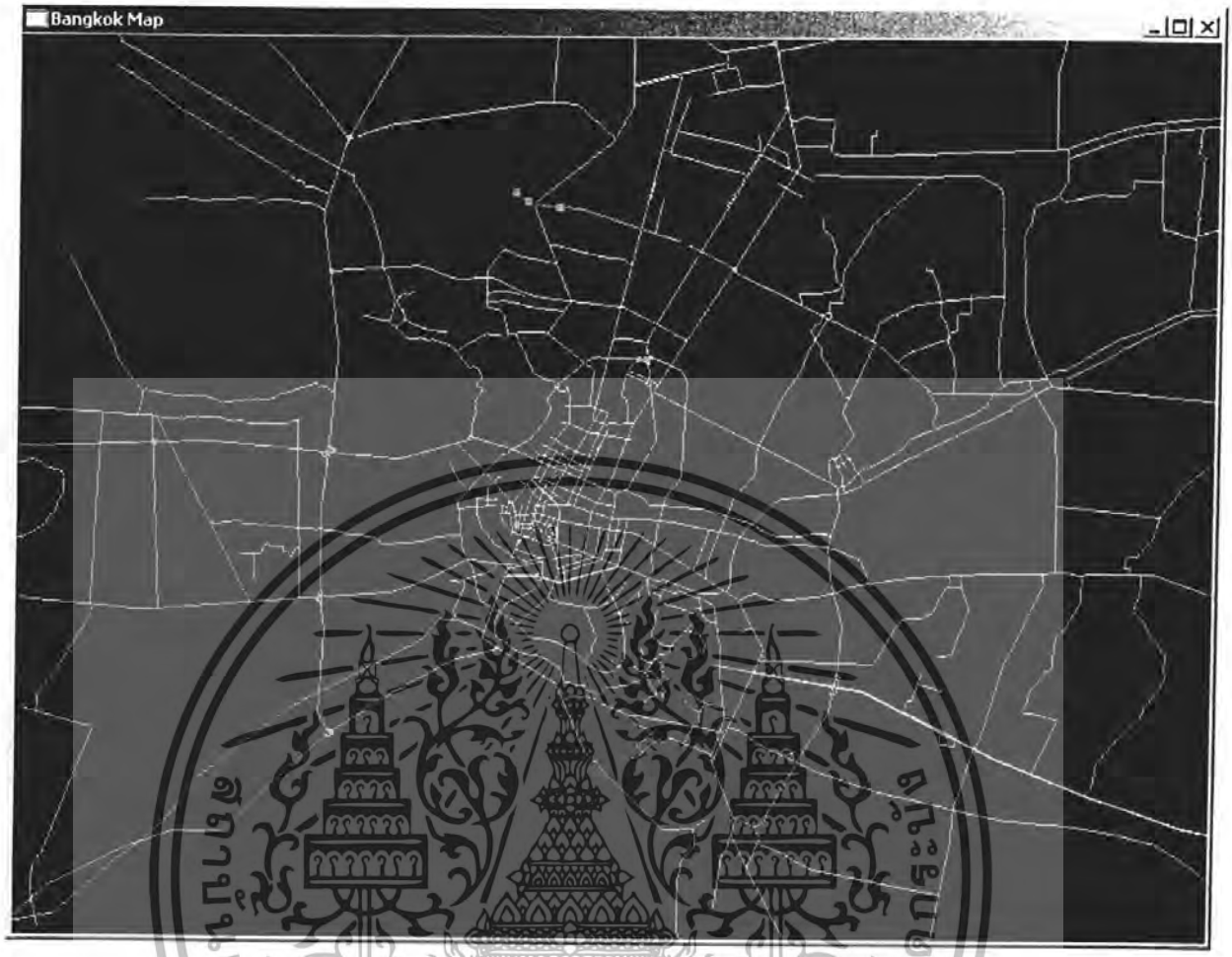
5. ComboBox เพื่อบอกว่าปัจจุบันรถอยู่ ณ สถานีไหน




**รูปที่ 5.10** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Route

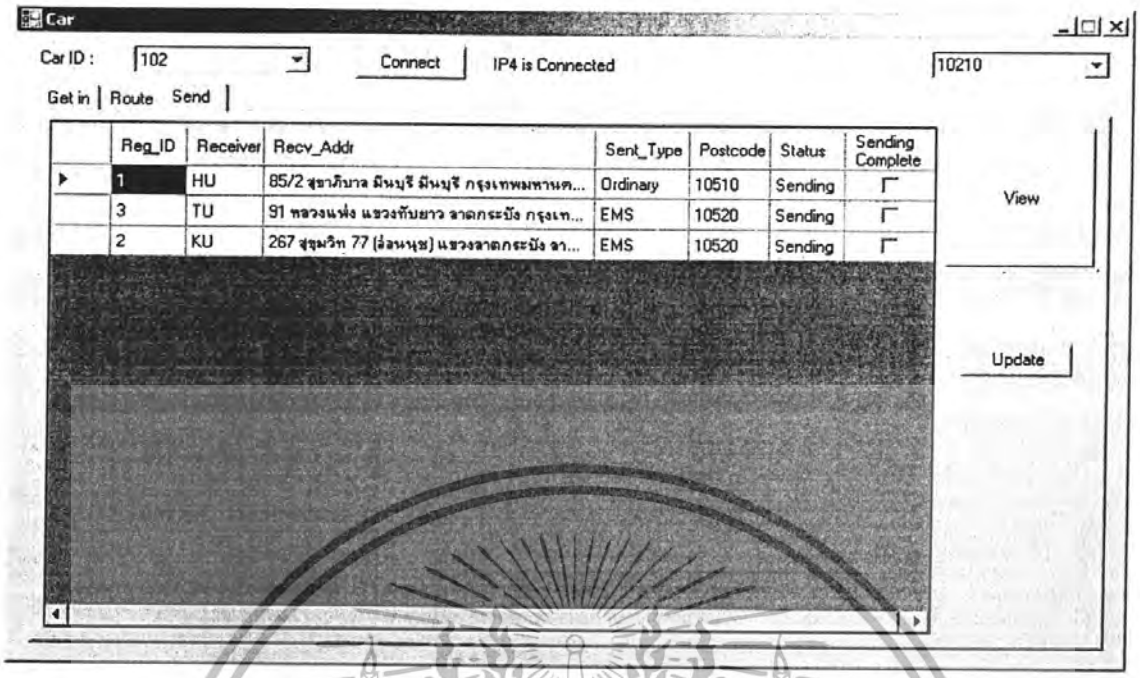
เมื่อขนสิ่งของขึ้นรถเรียบร้อยแล้วก็จัดการเรื่องเส้นทางการขนส่งที่แท็บ Route โดยจะมีปุ่ม Scan ให้กดแสดงผลลัพธ์ใน DataGridView 2 อันคือ อันบนจะแสดงเส้นทางเป็นชื่อถนนและพิกัดตามลำดับที่ได้จากการค้นหาเส้นทางผ่านอัลกอริทึม ส่วนอันล่างจะแสดงลำดับว่าต้องไปที่สถานีย่อยใดก่อนหลังตามลำดับ

นอกจากนี้ยังมีหน้าต่างย่อย popup ออกมาเพื่อให้เลือกว่าต้องการแสดงแผนที่การเดินทางหรือไม่ โดยเส้นทางเป็นแผนที่กรุงเทพฯ ถนนวาดแทนด้วยเส้นสีเขียว จุดเริ่มต้นจะแทนด้วยสีน้ำเงิน จุดต่อไปแทนด้วยสีส้ม และจุดสุดท้ายแทนด้วยสีชมพู



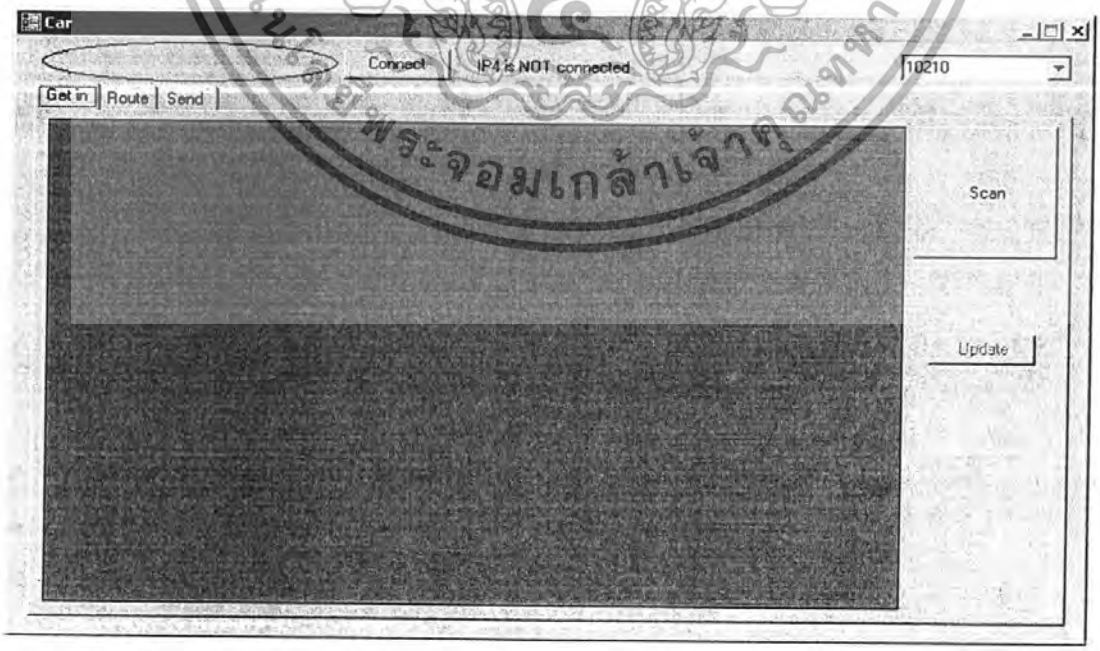

**รูปที่ 5.11** แสดงแผนที่ที่รถจัดส่งสิ่งของต้องวิ่งผ่าน

ระหว่างการขนส่งในแท็บ Send มีปุ่ม View ให้กด เพื่อแสดงลำดับของสิ่งของที่ต้องส่งไปสถานีย่อยใดๆใน DatagridView ภายในคอลัมน์สุดท้ายจะมีช่อง CheckBox ให้เลือกว่าทำการส่งเรียบร้อยแล้วหรือยัง เมื่อถึงสถานีย่อยที่ต้องไปส่ง โดยต้องเลือกรหัสไปรษณีย์ของสถานีย่อยนั้นตรง DropDown มุมบนขวาก่อน จึงกดปุ่ม Update เพื่อบันทึกข้อมูลลงฐานข้อมูลว่าของถึงสถานีย่อยปลายทางแล้ว สถานะของสิ่งของจะเป็น “Store” อยู่ที่สถานีย่อยนั้น




**รูปที่ 5.12** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำแวร์เฮาส์: แท็บ Send

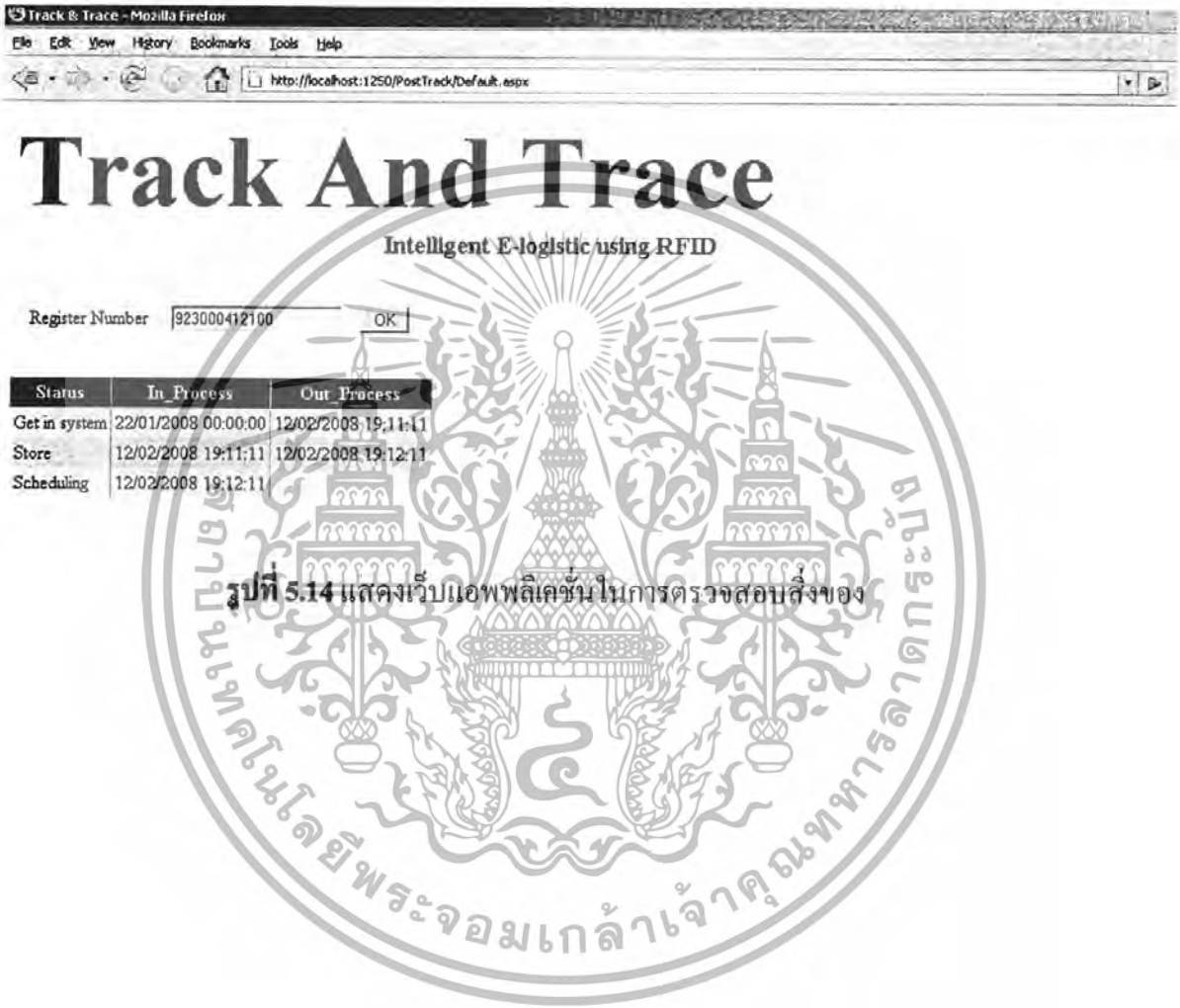
รถประจำสถานีย่อย รถจัดส่งสิ่งของจะเริ่มต้นการทำงานที่สถานีย่อยนั้นๆ โดยใช้รหัสไปรษณีย์บ่งบอก จะมีแต่การส่งออกของไปยังที่อยู่ผู้รับเท่านั้น หน้าต่างแอพพลิเคชั่นก็จะแตกต่างจากของรถประจำแวร์เฮาส์เล็กน้อย คือ ไม่ต้องมีส่วน ComboBox สำหรับเลือกรถจัดส่ง




**รูปที่ 5.13** แสดงแอพพลิเคชั่นในส่วนของรถจัดส่งสิ่งของประจำสถานีย่อย

## 5.4 เว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ

ผู้ใช้งานระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดีสามารถทำการตรวจสอบสถานะของสิ่งของได้ว่าอยู่ขั้นตอนส่งในขั้นใดโดยใช้บริการของเว็บแอพพลิเคชั่น และกรอกของรหัสสิ่งของลงไป เว็บจะทำการแสดงรายละเอียดของการส่งสิ่งของชิ้นนั้นๆขึ้นมา




**รูปที่ 5.14** แสดงเว็บแอพพลิเคชั่นในการตรวจสอบสิ่งของ

# บทที่ 6
# บทสรุปและวิจารณ์

## 6.1 บทสรุปและวิจารณ์

ระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี เป็นระบบที่พัฒนาขึ้นมาโดยมีจุดประสงค์หลักคือ ลดเวลาและค่าใช้จ่ายในการขนส่งสิ่งของจนถึงมือผู้บริโภค ซึ่งระบบมีการนำเทคโนโลยีอาร์เอฟไอดีมาพัฒนาเพื่อที่จะนำมาแทนที่ระบบเก่าที่ใช้บาร์โค้ด ทำให้ระบบสามารถทำงานได้รวดเร็วขึ้นเนื่องจากสามารถระบุถึงตัวสิ่งของแต่ละชิ้นได้อย่างรวดเร็วและแม่นยำมากกว่าเดิม ทั้งยังส่งผลให้ระบบมีความน่าเชื่อมากขึ้นด้วย เนื่องจากแท็กอาร์เอฟไอดีนั้นมีคุณสมบัติที่ยากต่อการปลอมแปลง ระบบยังมีการติดตามสิ่งของโดยให้บริการเป็นเว็บแอพพลิเคชั่นให้ผู้ใช้งานสามารถตรวจสอบและติดตามการขนส่งสิ่งของได้

นอกจากการนำเทคโนโลยีอาร์เอฟไอดีมาใช้ในระบบแล้วยังมีการพัฒนาระบบให้มีความสามารถสูงขึ้นโดยการนำอัลกอริทึมเอสตาร์มาใช้เพื่อค้นหาเส้นทางในการขนส่งสิ่งของให้ถึงปลายทาง ทำให้ระบบสามารถค้นหาเส้นทางที่สั้นที่สุดในการขนส่งสิ่งของได้ ซึ่งส่งผลให้เวลาในการขนส่งสิ่งของลดลงไปอีก ทั้งนี้เส้นทางการขนส่งที่สั้นที่สุดอาจไม่ใช่เส้นทางที่ใช้เวลาในการขนส่งน้อยที่สุด ดังนั้นจึงควรมีการเก็บข้อมูลของการขนส่งสิ่งของในแต่ละครั้ง เพื่อนำมาเป็นข้อมูลในการตัดสินใจและเปรียบเทียบการเลือกเส้นทางในการขนส่งครั้งต่อๆไป

## 6.2 ปัญหาที่เกิดขึ้นในการพัฒนา

- ในการพัฒนาระบบมีการใช้ DBMS หลายตัวเช่น PostgreSQL Oracle MySQL จึงต้องศึกษาการใช้งานฐานข้อมูลแต่ละตัวเพื่อนำมาพัฒนาระบบขนส่งอิเลคทรอนิกส์ชาญฉลาดโดยใช้อาร์เอฟไอดี
- เนื่องจากภาษาไอรอนไพธอนยังไม่เป็นที่นิยมนัก หรือยังไม่ค่อยรู้จัก จึงต้องศึกษาค้นคว้าโดยอาศัยประสบการณ์การเขียนภาษาไพธอนและดอทเน็ต ซึ่งมีบางสิ่งแต่กต่างจากวิธีที่คุ้นเคย ต้องอาศัยการสอบถามพูดคุยผ่านทางจดหมายอิเลคทรอนิกส์กับผู้ที่ใช้เป็นประจำหรือมีประสบการณ์ที่คล้ายๆกันหรือมากกว่า หรือต้องสร้างความคุ้นเคยในใช้ระยะเวลาหนึ่งก็จะสามารถใช้งานได้ดี

## 6.3 แนวทางการพัฒนาต่อ

- เลือกแบ่งกลุ่มเขตการจัดส่งสิ่งของ เพื่อลดจำนวนสิ่งของในการค้นหาเส้นทางการขนส่งสิ่งของให้น้อยลงเพื่อให้สอดคล้องกับอัลกอริทึมในการค้นหาเส้นทาง
- ปรับปรุงอัลกอริทึมในการค้นหาเส้นทางการขนส่งสิ่งของให้สามารถค้นหาเส้นทางที่รองรับสิ่งของได้จำนวนมากยิ่งขึ้น
- เพิ่มข้อมูลในฐานข้อมูลในส่วนของตาราง Address เพื่อรองรับสถานที่ในการส่งให้มากยิ่งขึ้น

# บรรณานุกรม


กัลยา อุดมวิทิต, อภิญญา กมลสุข และคณะ. 2548. หน้า 6. ใน **รายงานเบื้องต้นสถานภาพของอุตสาหกรรม RFID ในประเทศไทย.** ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

ขวัญชนก วิริยกุลโอภาศ. 2549. "RFID." หน้า 6 – 13. ใน **รายงานการศึกษารายวิชาเทคโนโลยีสำหรับบรรณารักษศาสตร์และสารนิเทศศาสตร์.** กรุงเทพฯ : คณะอักษรศาสตร์ จุฬาลงกรณมหาวิทยาลัย.

ทวีศักดิ์ กออนันตกูล. 2548. **เทคโนโลยี RFID กับผลกระทบต่อประเทศไทย.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

นระ คมนามูล. 2550. "**ความหมาย *โลจิสติกส์* ในด้านการขนส่ง".** [Online]. Available : http://www.tpa.or.th/writer/read_this_book_topic.php?passTo=f241731bd26874e0a13071bd26874e0a13071a66b4f289d&pageid=1&bookID=589&read=true&count=true.

วัชรากร หนูทอง. ม.ป.ป. **RFID หนึ่งในเทคโนโลยีที่น่าจับตามอง.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

วัชรากร หนูทอง และ อนุกูล น้อยไม้. ม.ป.ป. **RFID หนึ่งในเทคโนโลยีที่น่าจับตามอง.** [สไลด์]. ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

ศูนย์พัฒนาธุรกิจออกแบบวงจรรวม. 2548. **รู้จักกับเทคโนโลยีอาร์เอฟไอดี.** ปทุมธานี : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ.

สุชาติ กิจธนะเสรี. 2549. **Logistics RFID.** สถาบันคอมพิวเตอร์.

Brooks Automation. **"RFID Basics".** [Online]. Available : http://www.ready4rfid.com/rfid_index.php?menu=rfid_frequencies&spr=e.

Daniel, M. Dobkin. 2005. **"The RF in RFID".** [Online]. Available : http://www.enigmatic-consulting.com/Communications_articles/RFID/RFID_frequencies.html

Hunt, V. Daniel, Puglia, Albert and Puglia, Mike. 2007. **RFID-A GUIDE TO RADIO FREQUENCY IDENTIFICATION.** New Jersey : John Wiley & Sons.

IBM Press. 2005. **"RFID Sourcebook".** [Online]. Available : http://wireless.itworld.com/4985/051004_book_rfidsourcebook/page_1.html

Identify Limited. 2549. **RFID in Logistics & Supply Chain.** [สไลด์]. นนทบุรี : Identify Limited.

Smith Sukamith. 2004. "RFID for Smart Logistics" ใน **Logistics Asia 2004 Conference**

Smith Sukamith. 2007. **RFID Technology & Applications.** [สไลด์]. ปทุมธานี : สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ.

**"Logistic System".** [Online]. Available : http://www.jb-hg.com/elibrary/logistic.htm#important

**"Radio-frequency identification".** [Online]. Available : http://en.wikipedia.org/wiki/Radio-frequency_identification.

**"RFID Technology".** [Online]. Available : http://www.speakingeye.com/index.php?lay=show&ac=article&Id=401359&Ntype=2.