การพัฒนาโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

นาย นุกูล ทวีศักดิ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2543

ISBN 974-346-982-6

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF PROTOTYPE PROGRAM FOR DATA FILTER AND ERADICATION OF REPEATED AIRCRAFT IN THE AREA OF MULTIPLE RADARS SYSTEM

MR. NUKOOL TAVEESAK

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2000

ISBN 974-346-982-6

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง
โดย	นาย นุกูล ทวีศักดิ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์กอบกุล เตชะวณิช
อาจารย์ที่ปรึกษาร่วม	พันโทจักรพงศ์ นวลชื่น

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์วิชาญ เลิศวิภาตระกูล)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์กอบกุล เตชะวณิช)

..... อาจารย์ที่ปรึกษาร่วม
(พันโทจักรพงศ์ นวลชื่น)

..... กรรมการ
(อาจารย์ ดร. อาทิตย์ ทองทักษ์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นุกูล ทวีศักดิ์ : การพัฒนาโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง (DEVELOPMENT OF PROTOTYPE PROGRAM FOR DATA FILTER AND ERADICATION OF REPEATED AIRCRAFT IN THE AREA OF MULTIPLE RADARS SYSTEM) อ. ที่ปรึกษา : ผู้ช่วยศาสตราจารย์กอบกุล เตชะวณิช, อ. ที่ปรึกษาร่วม : พันโทจักรพงศ์ นวลชื่น , 203 หน้า ISBN 974-346-982-6

วิทยานิพนธ์นี้นำเสนอการออกแบบและการพัฒนาระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง การพัฒนาจะเริ่มจากการสร้างข้อมูลเรดาร์โดยใช้โปรโตคอลแสดงภาพสถานการณ์ทางอากาศระยะไกล (Remote Air Situation Display) ของเรดาร์ดีอาร์หนึ่งร้อยเจ็ดสิบสองเอดีวี (DR172 ADV) เป็นต้นแบบ การพัฒนาเริ่มจากการพัฒนาระบบแสดงผลทางจอภาพของเรดาร์เพื่อแสดงให้เห็นภาพทั้งการสร้างข้อมูลและการแสดงผลของข้อมูล และการพัฒนาระบบกรองข้อมูลและขจัดความซ้ำซ้อนและข้อมูลอากาศยานของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องเพื่อให้ได้ข้อมูลในรูปแบบที่ต้องการ พัฒนาระบบประเมินความถูกต้องของระบบขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

การวิจัยครั้งนี้ได้ทำการพัฒนาโปรแกรมโดยใช้หลักของมัลติโปรแกรมมิง(Multiprogramming) และใช้ภาษาจาวาและวิชาลเบสิก เป็นเครื่องมือในการพัฒนาโปรแกรม เมื่อทำการทดสอบระบบนี้ระบบทำงานได้ผลเป็นที่น่าพอใจยิ่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2543 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4171447821 : MAJOR COMPUTER SCIENCE
KEY WORD: RADAR / AIRCRAFT / ERADICATION / MULTIPROGRAMMING / PROTOCOL
NUKOOL TAVEESAK : THESIS TITLE. DEVELOPMENT OF PROTOTYPE PROGRAM FOR DATA FILTER ERADIATION OF REPEATED AIRCRAFT IN THE AREA OF MULTIPLE RADARS SYSTEM THESIS ADVISOR : KORBKUL TEJAVANIJA, THESIS COADVISOR : JAKRAPHONG NAULCHUEN, 203 pp. ISBN 974-346-982-6.

This thesis presents a design and development of a prototype program for data filtering and eradication of repeated aircrafts in the area of multiple radars system. The first step of development is the development of program that generates the radar data. By using the format of Remote Air Situation Display Protocol of DR172 ADV for generate data. The second step of development is the development of graphic user interface for input and output. The third step of development is the development of data filter and eradication of repeated aircraft in the area of multiple radars system and output for display unit. The last step of development is the development of correctness estimation for the system.

This thesis was developed by using multiprogramming method. Visual BASIC and JAVA were used for develop the program. The experimental result of this program is satisfaction.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering
Field of study Computer Science
Academic year 2000

Student's signature……..
Advisor's signature …….....
Co-advisor's signature .....…….

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ ด้วยความช่วยเหลืออย่างดียิ่ง ของผู้ช่วยศาสตราจารย์ กอบกุล เตชะวณิช อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ในการวิจัยและพัฒนามาด้วยดีตลอด

ขอขอบคุณท่านอาจารย์และเจ้าหน้าที่จากภาควิชาวิศวกรรมคอมพิวเตอร์ เจ้าหน้าที่จากหอสมุดกลางที่ได้ให้คำแนะนำและช่วยเหลือ ตลอดจน พี่ๆเพื่อนและน้องๆ ทุกคนรวมทั้งภรรยา ที่คอยช่วยเหลือและให้กำลังใจมาโดยตลอด

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา-มารดา และบุคคลในครอบครัวของผู้วิจัยซึ่งคอยให้กำลังใจแก่ผู้วิจัยเสมอจนสำเร็จการศึกษา

นุกูล ทวีศักดิ์
กุมภาพันธ์ 2541

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญ.....ช

สารบัญตาราง.....ญ

สารบัญภาพ.....ฏ

บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 แนวความคิด.....4

1.3 วัตถุประสงค์ของการวิจัย.....5

1.4 ขอบเขตของการวิจัย.....6

1.5 ขั้นตอนและวิธีดำเนินงาน.....7

1.6 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย.....8

แนวคิดและทฤษฎีที่เกี่ยวข้อง.....9

2.1 เรดาร์(RADAR).....9

2.1.1 เครื่องส่ง (TRANSMITTER).....10

2.1.2 เครื่องรับ (RECEIVER).....11

2.1.3 จานสายอากาศ (ANTENNA).....11

2.1.4 จอภาพ (DISPLAY).....12

2.2 สูตรคำนวณหาพื้นที่ร่วมของเรดาร์.....12

2.3 แนวทางในการคำนวณเวลาของเรดาร์จากการหมุนของเรดาร์.....14

2.4 แนวทางในการขจัดความซ้ำซ้อนของอากาศยาน.....15

2.5 แนวทางในการสร้างข้อมูลอากาศยาน.....16

2.6 สูตรการแปลงข้อมูลแผนที่ประเทศไทย.....17

2.7 แนวทางในการทำให้เวลาของเรดาร์แต่ละเครื่องตรงกัน.....18

2.8..โปรโตคอลการแสดงภาพสถานการณ์ทางอากาศระยะไกล (REMOTE AIR SITUATION DISPLAY PROTOCOL).....19

2.9 ข้อมูลของระบบเรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวี(DR172 ADV).....24

หน้า

การวิเคราะห์และออกแบบ 25

3.1 การออกแบบระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยาน.....27

3.1.1 รับข้อมูล 28

3.1.2 ส่งข้อมูล 29

3.1.3 การกรองข้อมูล.....29

3.1.4 ขจัดความซ้ำซ้อนของข้อมูล 29

3.1.5 การสร้างโปรโตคอล.....30

3.1.6 การส่งข้อมูลโปรโตคอล.....31

3.1.7 ตื่น / หลับครบรอบ4วินาที.....31

3.2 การออกแบบระบบสร้างข้อมูลเรดาร์และการส่งข้อมูล.....31

3.2.1 สร้างแผนที่ประเทศไทยและประเทศใกล้เคียง.....33

3.2.2 สร้างการรัศมีของเรดาร์.....34

3.2.3 สร้างเส้นทางการบิน.....34

3.2.4 สร้างข้อมูลจำลองข้อมูลโปรโตคอล.....35

3.2.5 สร้างข้อมูลแสดงจำนวนของอากาศ.....35

3.2.6 ส่งข้อมูล 35

3.3 การออกแบบระบบแสดงผลทางจอภาพ.....36

3.4 การออกแบบแฟ้มข้อมูลและตารางข้อมูล.....37

3.5 การออกแบบแสดงผลทางจอภาพ 38

3.5.1 การออกแบบแสดงผลเรดาร์ทางจอภาพ.....40

3.5.2 การออกแบบแสดงผลเส้นทางการบินทางจอภาพ.....42

3.5.3 การออกแบบแสดงผลของข้อมูลเส้นทางการบินทางจอภาพ.....44

การพัฒนาระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง.....46

4.1 เมนูหลัก 47

4.2 การสร้างข้อมูลเรดาร์.....47

4.3 การส่งข้อมูลแบบเก่า.....50

4.4 ระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานและการส่งข้อมูล 51

4.5 ระบบแสดงผลทางจอภาพระบบเก่า 53

4.6 ระบบแสดงผลทางจอภาพระบบใหม่ 55

หน้า

การทดสอบโปรแกรม 57

5.1 เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ.....57

5.2 โปรแกรมสำเร็จที่ใช้ในการทดสอบ.....57

5.3 ขั้นตอนในการทดสอบโปรแกรม 57

สรุปผลวิจัยและข้อเสนอแนะ 69

6.1 สรุปผลการวิจัย.....69

6.2 ข้อจำกัดของระบบ 71

6.3 ประโยชน์ที่คาดว่าจะได้รับ.....72

6.4 ข้อเสนอแนะ 72

รายการอ้างอิง 74

ภาคผนวก ก. 75

ภาคผนวก ข. 82

ภาคผนวก ค. 97

ภาคผนวก ง. 179

ประวัติผู้เขียนวิทยานิพนธ์ 203

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง หน้า

ตารางที่ 2.1 การส่งสัญญาณของระบบเรดาร์ 20
ตารางที่ 2.2 ข้อมูลสถานที่ตั้งของเรดาร์ 21
ตารางที่ 2.3 ข้อมูลของอากาศยาน 22
ตารางที่ 5.1 การทดสอบครั้งที่ 1 70
ตารางที่ 6.1 ผลสรุปของการทดลอง 73
ตารางที่ ก-1 ตำแหน่งของอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศไทย 78
ตารางที่ ก-2 ตำแหน่งของอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศใกล้เคียง 78
ตารางที่ ก-3 แสดงการเก็บข้อมูลของตำแหน่งเรดาร์ 78
ตารางที่ ก-4 โครงสร้างข้อมูลของอากาศยาน 79
ตารางที่ ก-5 โครงสร้างข้อมูลของอากาศยานที่ใช้ในการเรียงลำดับ 79
ตารางที่ ก-6 โครงสร้างข้อมูลรายละเอียดของอากาศยาน 80
ตารางที่ ก-7 โครงสร้างข้อมูลของอากาศยานเพื่อใช้กำหนดหมายเลขอากาศยาน 80
ตารางที่ ก-8 โครงสร้างข้อมูลโปรโตคอลของอากาศยาน 80
ตารางที่ ก-9 ข้อมูลสถานที่ตั้งของเรดาร์ 81
ตารางที่ ก-10 ข้อมูลของอากาศยาน 82
ตารางที่ ก-11 โครงสร้างข้อมูลของจำนวนอากาศยานต่อการรอบของเรดาร์ 82
ตารางที่ ก-12 โครงสร้างข้อมูลความถูกต้องของระบบ 83
ตารางที่ ง-1 การทดสอบครั้งที่ 2 186
ตารางที่ ง-2 การทดสอบครั้งที่ 3 191
ตารางที่ ง-3 การทดสอบครั้งที่ 4 196
ตารางที่ ง-4 การทดสอบครั้งที่ 5 203

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ หน้า

รูปที่ 1.1 การแจ้งเตือนการป้องกันภัยทางอากาศ 2
รูปที่ 1.2 การวางตัวของเรดาร์ทางด้านทิศตะวันตก 2
รูปที่ 1.3 อากาศยานในพื้นที่ร่วม 4
รูปที่ 1.4 การแจ้งเตือนการป้องกันภัยทางอากาศที่ได้รับการปรับปรุง 5
รูปที่ 1.5 การวางตำแหน่งของเรดาร์ 6
รูปที่ 2.1 ลักษณะPLUSEของคลื่นเรดาร์ 10
รูปที่ 2.2 ระบบการทำงานของเรดาร์ 11
รูปที่ 2.3 แสดงการที่เรดาร์ที่ไม่มีพื้นที่ร่วม 13
รูปที่ 2.4 แสดงการที่เรดาร์ที่มีพื้นที่ร่วม 13
รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างเวลาและตำแหน่งของอากาศยาน 14
รูปที่ 2.6 ขั้นตอนของการส่งสัญญาณเพื่อการแสดงผล 20
รูปที่ 3.1 ระบบเรดาร์และการแสดงผล 25
รูปที่ 3.2 ระบบการขจัดความซ้ำซ้อนและการกรองข้อมูลและการแสดงผล 26
รูปที่ 3.3 กระบวนการการส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรองข้อมูล 28
รูปที่ 3.4 การส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรอง 29
รูปที่ 3.5 การขจัดความซ้ำซ้อนของอากาศยาน 31
รูปที่ 3.6 การสร้างโปรโตคอลของอากาศยาน 32
รูปที่ 3.7 การสร้างโปรโตคอลของเรดาร์ 33
รูปที่ 3. 8 การส่งข้อมูล 37
รูปที่ 3. 9 แสดงผลทางจอภาพ 37
รูปที่ 3.10 แสดงผลเมนูหลักทางจอภาพ 40
รูปที่ 3.11 โครงสร้างของการออกแบบแสดงผลทางจอภาพ 41
รูปที่ 3.12 แสดงผลแฟ้มข้อมูลเรดาร์ทางจอภาพ 42
รูปที่ 3.13 แสดงผลการจัดการเรดาร์ทางจอภาพ 43
รูปที่ 3.14 แสดงผลย่อหรือขยายเรดาร์ทางจอภาพ 43
รูปที่ 3.15 แสดงผลการสร้างเส้นทางการบินทางจอภาพ 44
รูปที่ 3.16 แสดงผลการกำหนดส่วนของวินาทีเพื่อสร้างข้อมูลการบินทางจอภาพ 45
รูปที่ 3.17 แสดงผลย่อหรือขยายเส้นทางการบินทางจอภาพ 45
รูปที่ 3.18 แสดงผลสถิติในระบบเก่าทางจอภาพ 46

หน้า

รูปที่ 3.19 แสดงผลสถิติในระบบที่ผ่านการขจัดและการกรองข้อมูลทางจอภาพ 47
รูปที่ 3.20 แสดงผลสถิติในระบบทั้งสองระบบทางจอภาพ 47
รูปที่ 4.1 การพัฒนาโปรแกรมกรองข้อมูลและการขจัดความซ้ำซ้อน 49
รูปที่ 5.1 สร้างรัศมีของเรดาร์ 60
รูปที่ 5.2 สร้างเส้นทางการบินของอากาศยานแบบที่ 1 60
รูปที่ 5.3 สร้างเส้นทางการบินของอากาศยานแบบที่ 2 61
รูปที่ 5.4 สร้างเส้นทางการบินของอากาศยานแบบที่ 3 61
รูปที่ 5.5 รันโปรแกรมส่งกรองและขจัดความซ้ำซ้อนของข้อมูล 62
รูปที่ 5.6 การทดสอบโปรแกรมกรองข้อมูลและการขจัดความซ้ำซ้อน 63
รูปที่ ข-1 แสดงการติดตั้งโปรแกรมยังไดเรกทอรี่ c:\thesis 86
รูปที่ ข-2 แสดงผลหลังจากเลือกเมนูเรดาร์ 87
รูปที่ ข-3 แสดงลักษณะหน้าจอหลังจากเลือกเมนูสร้างเรดาร์ 88
รูปที่ ข-4 แสดงลักษณะหน้าจอของเมนูย้ายเรดาร์ 88
รูปที่ ข-5 แสดงลักษณะหน้าจอของเมนูลบเรดาร์ 89
รูปที่ ข-6 แสดงลักษณะหน้าจอของการขยายภาพตามมุมมองที่ต้องการ 90
รูปที่ ข-7 แสดงลักษณะหน้าจอของเมนูเส้นทางการบิน 90
รูปที่ ข-8 แสดงลักษณะหน้าจอของเมนูสร้างเส้นทางการบินเพื่อเลือกอากาศยานที่ต้องการ 91
รูปที่ ข-9 แสดงลักษณะหน้าจอของเมนูสร้างเส้นทางการบินของการกำหนดเส้นทางการบิน 92
รูปที่ ข-10 แสดงผลลัพธ์ที่ได้หลังจากสร้างเส้นทางการบินเรียบร้อยแล้ว 93
รูปที่ ข-11 แสดงข้อความเตือนสามารถทำการย้ายเส้นทางการบินได้ 94
รูปที่ ข-12 แสดงหน้าจอของผลลัพธ์ที่ได้หลังจากย้ายเส้นทางการบินเรียบร้อยแล้ว 94
รูปที่ ข-13 แสดงลักษณะหน้าจอของเมนูลบเส้นทางการบิน 95
รูปที่ ข-14 แสดงลักษณะหน้าจอของเมนูส่งข้อมูลแบบระบบเก่า 96
รูปที่ ข-15 แสดงลักษณะหน้าจอของเมนูส่งข้อมูลผ่านการกรองและขจัดความซับซ้อน 97
รูปที่ ข-16 แสดงลักษณะหน้าจอของเมนูส่งข้อมูล 2 ระบบ 98

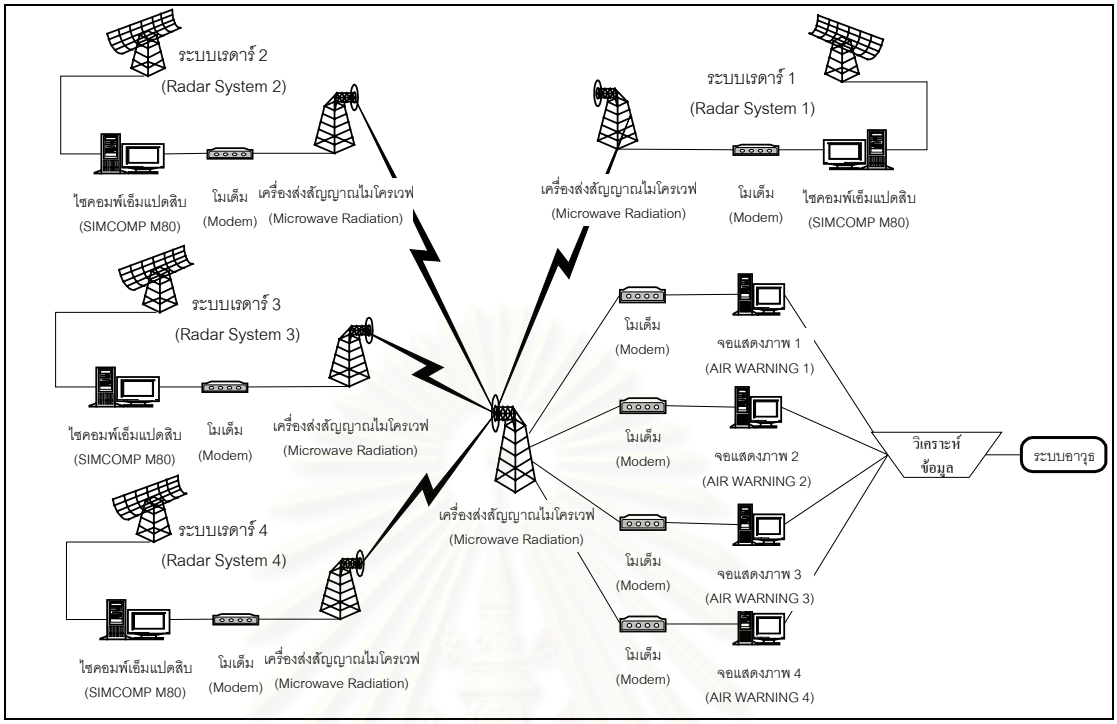
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การป้องกันภัยทางอากาศ ถือว่าเป็นภารกิจหนึ่งที่มีความสำคัญอย่างยิ่งในการป้องกันประเทศของกองทัพไทย โดยมีกองอำนวยการต่อสู้ป้องกันภัยทางอากาศกองบัญชาการทหารสูงสุดเป็นหน่วยงานหลักที่รับผิดชอบภารกิจนี้ กองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุดได้จัดซื้อเรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวี(DR172 ADV) จำนวน4เครื่องเพื่อใช้ในการตรวจจับอากาศยานของข้าศึก

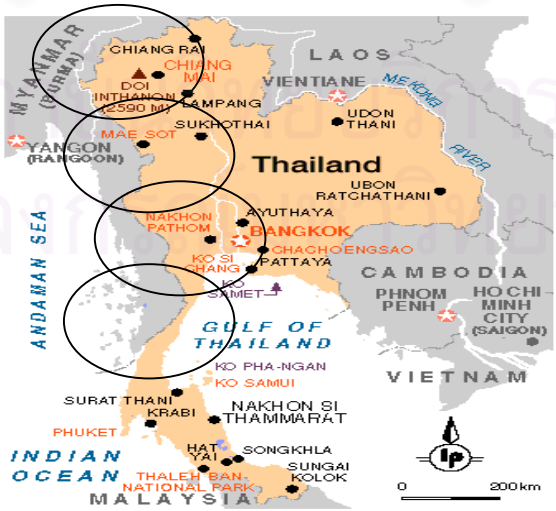
เรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวีเป็นเรดาร์ที่ทำการตรวจจับอากาศยานที่มีเพดานการบินระดับต่ำและเป็นเรดาร์ที่สามารถตรวจจับภาพแบบ 2 มิติ.เรดาร์จะมีรายละเอียดเฉพาะตำแหน่งทางภูมิศาสตร์ของอากาศยานแต่จะไม่มีรายละเอียดความสูงของอากาศยาน เรดาร์ระบบนี้จะมีข้อเด่นคือสามารถตรวจจับอากาศยานที่มีเพดานบินระดับต่ำได้มีประสิทธิภาพสูง เมื่อเรดาร์จับสัญญาณอากาศยานได้ จะส่งสัญญาณเข้าเครื่องคอมพิวเตอร์ไซคอมท์เอ็มแปดสิบ (SICOMP M80) เพื่อทำการคำนวณหาตำแหน่งทางภูมิศาสตร์ ความเร็วของอากาศยาน กำหนดฝ่ายของอากาศยาน เมื่อทำการคำนวณเรียบร้อยแล้วจะแปลงสัญญาณเพื่อส่งต่อมาทำการแสดงผลที่จอภาพแสดงผลโดยผ่านทางเครื่องส่งสัญญาณไมโครเวฟ โดยสัญญาณที่ส่งมายังจอภาพแสดงผลจะส่งมาเป็นโปรโตคอลแสดงภาพสถานการณ์ทางอากาศระยะไกล (Remote Air Situation Display) ด้วยความเร็ว 2400 บิตต่อวินาที การส่งสัญญาณออกมาโดยทุกๆ 4 วินาที การสังเกตการณ์ผลของสัญญาณเรดาร์จะมีทหารสังเกตการณ์จากจอภาพแสดงผล เมื่อพบความผิดปกติจะทำการรายงานให้ผู้บังคับบัญชาทราบเพื่อดำเนินการต่อไป เมื่อบังคับบัญชาหรือผู้ที่ได้รับการมอบหมายวิเคราะห์แล้วเห็นว่าอากาศยานข้าศึกเป็นภัยคุกคามต่อฝ่ายเรา ผู้บังคับบัญชาหรือผู้ที่ได้รับการมอบหมายจะทำการแจ้งเตือน การเข้ามาของอากาศยานข้าศึกให้แก่ระบบอาวุธ เพื่อเตรียมพร้อมในการดำเนินยุทธวิธีต่ออากาศยานข้าศึก โดยเรดาร์จะทำการส่งข้อมูลของอากาศยานมาทุกๆ 4 วินาทีดังรูปที่ 1.1

ตามแผนของกองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุด เพื่อที่จะทำให้การตรวจจับอากาศยาน ได้ผลครอบคลุมแนวรบทางแนวด้านใดด้านหนึ่งของประเทศนั้น จำเป็นจะต้องใช้เรดาร์ระบบดีอาหนึ่งร้อยเจ็ดสิบสอง เอดีวี (DR172 ADV) ที่มีเรดาร์อย่างน้อยที่สุด 4 เครื่องร่วมกันทำงาน โดยเรดาร์จะมีรัศมีของสัญญาณประมาณ 75 ไมล์ทะเลหรือประมาณ 136.5 กิโลเมตรฉะนั้นทางกองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุดจึง



รูปที่ 1.1 การแจ้งเตือนการป้องกันภัยทางอากาศ

มีโครงการที่จะจัดซื้อเรดาร์ระบบดีอาหนึ่งร้อยเจ็ดสิบสอง เอดีวี4เครื่องเพื่อที่จะสามารถครอบคลุมแนวรบทางด้านใดด้านหนึ่งได้อย่างมีประสิทธิภาพ ดังนั้นการจัดเรียงเรดาร์ทั้ง 4 เครื่องให้มีพื้นที่ซ้ำซ้อนกัน เมื่อทางกองอำนวยการต่อสู้ป้องกันภัยทางอากาศกองบัญชาการทหารสูงสุดต้องการวางแนวเรดาร์ทางด้านทิศตะวันตก ตำแหน่งของเรดาร์จะมีการจัดตำแหน่งได้ดังรูปที่ 1.2



รูปที่ 1.2 การวางตัวของเรดาร์ทางด้านทิศตะวันตก

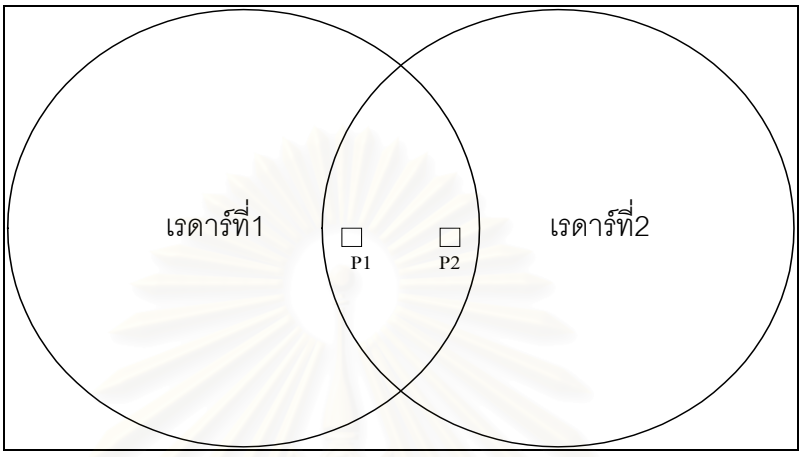
จากการที่เรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสอง เอดีวีแต่ละเครื่องจะทำงานอิสระจากกัน ดังนั้นในการปฏิบัติงานต้องใช้ทหารสังเกตการณ์ประจำจอภาพแสดงผลแต่ละเครื่อง เมื่อพบความผิดปกติจะทำการรายงานให้ผู้บังคับบัญชาทราบ เมื่อผู้บังคับบัญชาที่มีหน้าที่ทำการวิเคราะห์แล้วพบว่าอากาศยานข้าศึกมีภัยคุกคามต่อความมั่นคงของประเทศ จะทำการแจ้งเตือนการเข้ามาของอากาศยานข้าศึกนั้น ให้แก่ระบบอาวุธเพื่อเตรียมพร้อมในการดำเนินยุทธวิธีต่ออากาศยานข้าศึกต่อไป จากระบบดังกล่าวจะมีปัญหาที่ตามมาซึ่งสามารถแยกออกมาเป็นข้อๆได้ดังนี้คือ

1. เวลาที่ใช้ในการรายงานผลจะไม่ทันกับเหตุการณ์ การที่ใช้ทหารสังเกตการณ์จากจอภาพแสดงผลเมื่อพบความผิดปกติจะทำการรายงานให้ผู้บังคับบัญชาทราบ เพื่อดำเนินการต่อไป เมื่อบังคับบัญชา หรือผู้ที่ได้รับการมอบหมายวิเคราะห์แล้วเห็นว่าอากาศยานข้าศึกเป็นภัยคุกคามต่อฝ่ายเรา ผู้บังคับบัญชาหรือผู้ที่ได้รับการมอบหมายจะทำการแจ้งเตือนการเข้ามาของอากาศยานข้าศึกให้แก่ระบบอาวุธ เพื่อเตรียมพร้อมในการดำเนินยุทธวิธีต่ออากาศยานข้าศึกนั้น ต้องใช้เวลามากจนอาจจะไม่ทันต่อเหตุการณ์ เพราะอากาศยานในปัจจุบันมีความเร็วสูงและมีแนวโน้มที่จะเพิ่มขึ้นตามการณ์เวลา

2. การที่มีเรดาร์ระบบดีอาหนึ่งร้อยเจ็ดสิบสอง เอดีวี4เครื่องนั้นจะทำให้เกิดพื้นที่ร่วมของเรดาร์จะทำให้การสังเกตการณ์อากาศยานคาดเคลื่อนได้ โดยอากาศยานที่บินอยู่ในพื้นที่ร่วมของสัญญาณเรดาร์หลายเครื่องนั้นอากาศยานจะถูกตรวจจับ และแสดงภาพอยู่ในจอแสดงผลหลายเครื่อง จึงทำให้อากาศยานนั้นถูกระบุว่าเป็นหลายลำแต่ในความเป็นจริงแล้วอากาศยานนั้นอาจจะเป็นอากาศยานเดียวกันได้ โดยในทางปฏิบัติแล้วถ้าหากอากาศยานบินเข้ามาในพื้นที่ของเรดาร์ทหารจะสังเกตการณ์จากจอภาพแสดงผลประจำของตนเอง เมื่อพบความผิดปกติทหารสังเกตการณ์จะทำการรายงานให้ผู้บังคับบัญชาทราบ ข้อมูลที่ผู้บังคับบัญชาได้มานั้นอาจจะคลาดเคลื่อนจากความเป็นจริง เนื่องจากการวางตัวของเรดาร์ในลักษณะครอบคลุมพื้นที่เป็นบริเวณต่อเนื่องกัน จะมีพื้นที่บางส่วนที่สัญญาณเรดาร์มีพื้นที่ร่วมกันจะทำให้การตรวจจับอากาศยานคลาดเคลื่อนจากความเป็นจริงไปได้ ดังรูปที่ 1.3

โดยความคลาดเคลื่อนอาจมีสาเหตุมาจาก เมื่อเรดาร์ทั้งหมดเริ่มหมุนจากทางทิศเหนือและเวลาในการหมุนประมาณ 4 วินาทีต่อรอบ ทหารสังเกตการณ์จากจอภาพของเรดาร์ที่1จะเห็นภาพของอากาศยานในพื้นที่ร่วมในวินาทีที่1 ณ ตำแหน่ง P1 แต่ทหารสังเกตการณ์จากจอภาพของเรดาร์ที่ 2 จะเห็นภาพของอากาศยานเมื่อวินาทีที่ 3 ณ ตำแหน่ง P2 ทหารสังเกตการณ์จะรายงานผู้บังคับบัญชาว่ามีอากาศยาน 2 เครื่องที่ตำแหน่ง P1 และ P2 ซึ่งตามความเป็นจริงแล้วอากาศยานทั้ง 2 เป็นอากาศยานเครื่องเดียวกัน เนื่องจากเรดาร์ที่ 1 และเรดาร์ที่ 2 ตรวจพบอากาศยาน

ในเวลาที่ต่างกันโดยวินาทีที่ 1 อากาศยานปรากฏ ณ ตำแหน่ง P1 แต่วินาทีที่ 2 อากาศยานจะเคลื่อนที่ไป ณ ตำแหน่ง P2



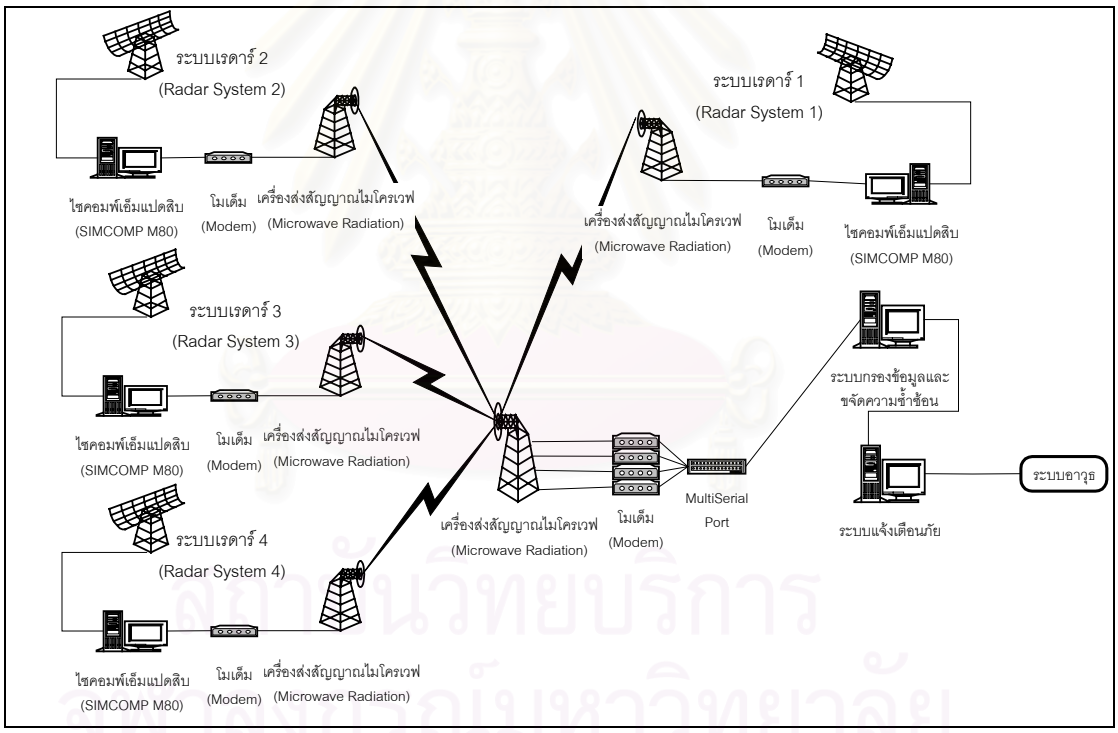
รูปที่ 1.3 อากาศยานในพื้นที่ร่วม

1.2 แนวความคิด

กองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุดได้มีนโยบายให้มีการพัฒนาระบบโปรแกรมคอมพิวเตอร์เพื่อแจ้งเตือนภัยทางอากาศ โดยโปรแกรมคอมพิวเตอร์ที่จะจัดทำขึ้นนั้นต้องสามารถรับสัญญาณจากเรดาร์ได้ทั้ง 4 เครื่อง แล้วนำข้อมูลที่ได้มาทำการวิเคราะห์เพื่อประเมินภัยคุกคามของอากาศยานที่มีต่อประเทศไทย และสามารถแจ้งเตือนภัยทางอากาศต่อหน่วยป้องกันอากาศยานเพื่อดำเนินยุทธวิธีต่ออากาศยานนั้นๆต่อไป โดยคาดหวังว่าระบบโปรแกรมคอมพิวเตอร์ที่จะทำการพัฒนาขึ้นมานี้นั้น สามารถเพิ่มประสิทธิภาพในการป้องกันภัยทางอากาศ โดยลดเวลาการติดต่อสื่อสารของหน่วยงานที่มีหน้าที่รับผิดชอบ และการส่งข้อมูลของอากาศยาน โดยระบบโปรแกรมคอมพิวเตอร์จะลดความผิดผลาดของข้อมูลเพื่อเพิ่มประสิทธิภาพในการแจ้งเตือนภัยทางอากาศ

ในการที่จะบรรลุวัตถุประสงค์ดังกล่าว จะต้องมีระบบคอมพิวเตอร์ที่สามารถรับสัญญาณจากเรดาร์ได้ทั้ง 4 เครื่อง เพื่อส่งข้อมูลที่ได้ต่อไปยังระบบแจ้งเตือนภัยทางอากาศ แต่สัญญาณที่ได้จากเรดาร์ทั้ง4เครื่องจะมีความซ้ำซ้อนของอากาศยานในพื้นที่ร่วมของสัญญาณเรดาร์ และมีข้อมูลบางส่วนที่ไม่มีความจำเป็นต่อการประเมินภัยคุกคามในระบบระบบแจ้งเตือนภัยทางอากาศ ทำให้ข้อมูลของอากาศยานคาดเคลื่อนจากความเป็นจริง และมีข้อมูลที่ไม่มีความจำเป็นต่อการประเมินภัยคุกคามในระบบระบบแจ้งเตือนภัยทางอากาศ ทำให้ระบบทำงานคาดเคลื่อนและล่าช้า

จากปัญหาดังกล่าวกองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุด ได้มีนโยบายให้มีการพัฒนาระบบโปรแกรมคอมพิวเตอร์ เพื่อกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องโดยในระบบใหม่นี้จะทำงานดังต่อไปนี้ โดยระบบเรดาร์ทำการตรวจจับอากาศยาน และส่งสัญญาณเรดาร์ที่เข้าเครื่องคอมพิวเตอร์ไซคอมท์เอ็มแปดสิบ เพื่อทำการคำนวณหาตำแหน่งทางภูมิศาสตร์ ความเร็วของอากาศยาน เมื่อทำการคำนวณเรียบร้อยแล้วจะแปลงสัญญาณ เพื่อส่งต่อมาทำการแสดงผลที่จอภาพแสดงผลโดยผ่านทางเครื่องส่งสัญญาณไมโครเวฟ เมื่อได้ข้อมูลที่ต้องการแล้วก็จะทำการส่งข้อมูลไปยังระบบกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ร่วมของอากาศยาน เมื่อได้ข้อมูลที่ต้องการแล้วระบบกรองข้อมูล และการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ร่วมของอากาศยานส่งข้อมูลที่ได้ไปยังระบบแจ้งเตือนภัยทางอากาศต่อไปดังแสดงในรูปที่ 1.4



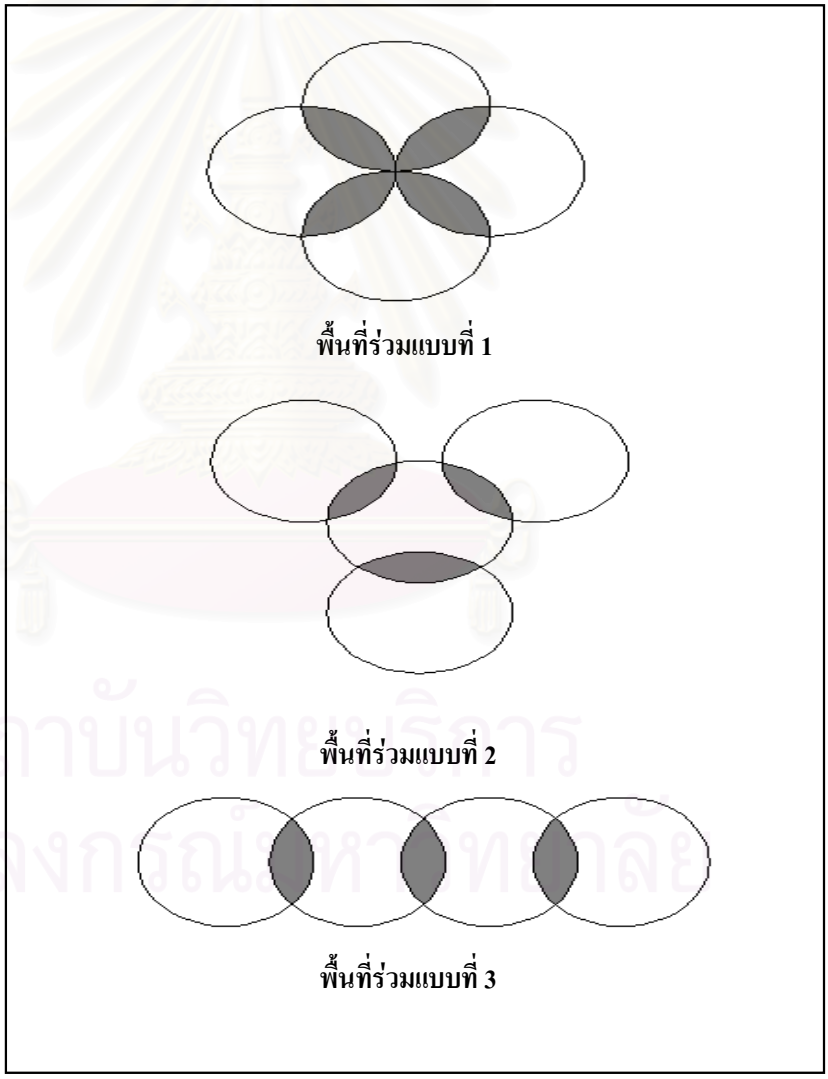
รูปที่ 1.4 การแจ้งเตือนการป้องกันภัยทางอากาศที่ได้รับการปรับปรุง

1.3 วัตถุประสงค์ของการวิจัย

สร้างซอฟต์แวร์ต้นแบบเพื่อกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

1.4 ขอบเขตของการวิจัย

1. จำลองการรับสัญญาณจากเรดาร์ระบบดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวี (DR172 ADV) จำนวนไม่เกิน 4 เครื่อง
2. ออกแบบและพัฒนาระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานของอากาศยานในพื้นที่ร่วมของเรดาร์ที่เรดาร์กวาดพบ
3. พื้นที่ร่วมที่เกิดขึ้นนั้นจะเกิดจากการซ้ำซ้อนของสัญญาณเรดาร์ระหว่างเรดาร์2 เครื่องเท่านั้นในพื้นที่เดียวกัน ตัวอย่างของพื้นร่วมที่เกิดขึ้นสามารถมีลักษณะ 3 แบบได้ดังต่อไปนี้



รูปที่ 1.5 การวางตำแหน่งของเรดาร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. ส่งข้อมูลที่ได้รับการกรองและขจัดความซ้ำซ้อนของอากาศยานไปยังระบบอาวุธโดยข้อมูลที่ทำการส่งมีรายละเอียดคือ
    4.1. จำนวนของข้อมูล แสดงจำนวนอากาศยานทั้งหมดหลังจากการขจัดความซ้ำซ้อน
    4.2. ตัวเลขอ้างอิงเส้นทางการบินที่ของอากาศยานที่ตรวจพบ
    4.3. ตำแหน่งพิกัดเส้นรุ้ง มีหน่วยเป็น1/8ของไมล์ทะเล
    4.4. ตำแหน่งพิกัดเส้นแวง มีหน่วยเป็น1/8ของไมล์ทะเล
    4.5. ความเร็วทางแกนX มีหน่วยเป็น1/8ของไมล์ทะเลต่อชั่วโมง
    4.6. ความเร็วทางแกนY มีหน่วยเป็น1/8ของไมล์ทะเลต่อชั่วโมง
    4.7. ลำดับของเส้นทางการบินโดยจะมีตัวเลขตั้งแต่ 001 ถึง 999
    4.8. หลักฐานของเส้นทางการบิน(Track Identification)
    4.9. เวลาของระบบ(System Time)
5. การพัฒนาจะทำโดยภาษาระดับสูง (High level language)
6. ระบบปฏิบัติการ วินโดว์ (Windows)
7. ข้อมูลมาจากการจำลองขึ้น

1.5 ขั้นตอนและวิธีดำเนินงาน

ในการพัฒนาระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องมีขั้นตอนการทำงานตามตารางเวลาการพัฒนาดังนี้

1. ศึกษาและวิเคราะห์ข้อมูลของสัญญาณเรดาร์
2. ออกแบบและพัฒนาระบบการจำลองการสร้างสัญญาณเรดาร์
3. ออกแบบและพัฒนาระบบการแสดงผลทางจอภาพด้วยภาพกราฟิกที่รับข้อมูลเรดาร์โดยตรงและข้อมูลที่ผ่านการกรองและขจัดความซ้ำซ้อน
4. พัฒนาระบบการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง
5. ทดสอบระบบระบบการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง
6. ประเมินและสรุปผลการทำงานของระบบการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

1.6 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1. ทำให้กองทัพบกได้โปรแกรมต้นแบบเพื่อกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง
2. เป็นแนวทางในการพัฒนาโปรแกรมกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานจากในพื้นที่ร่วมของระบบเรดาร์หลายเครื่องที่กวาดในรูป 3 มิติ
3. ข้อมูลที่ได้ผ่านการกรองและขจัดความซ้ำซ้อนของอากาศยานแล้วสามารถนำไปใช้เป็นข้อมูลของระบบแจ้งเตือนภัยทางอากาศ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 เรดาร์(RADAR) [1,2]

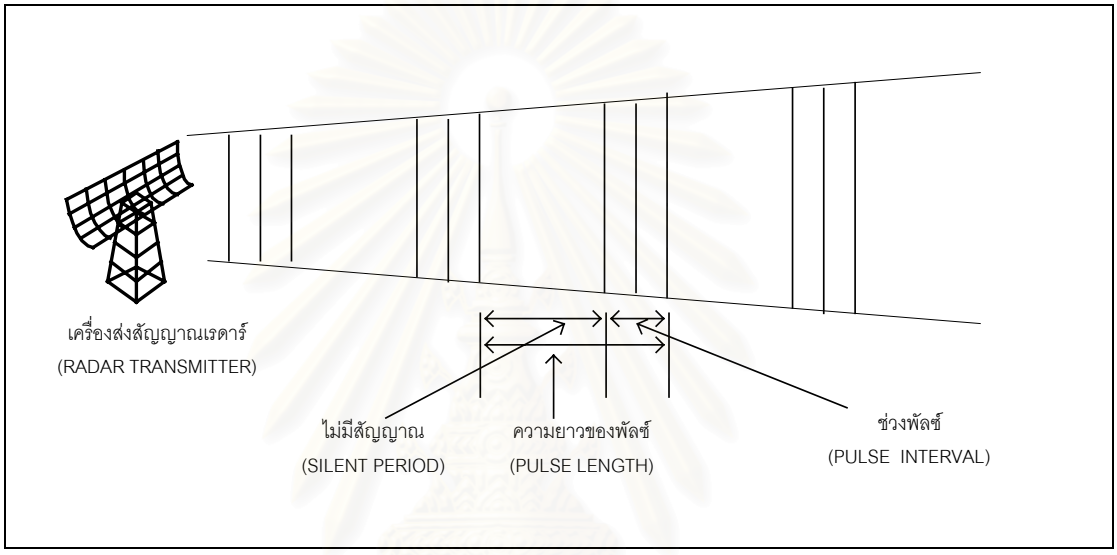
เรดาร์ได้รับการพัฒนามายังต่อเนื่องตั้งแต่ปี ค.ศ.1925 เป็นต้นมาจนถึงปัจจุบัน โดยการนำหลักการจากคุณสมบัติของคลื่นแม่เหล็กไฟฟ้าซึ่งสะท้อนกลับเมื่อกระทบวัตถุ ได้มีการพัฒนาเรดาร์เพื่อใช้ในการตรวจจับเครื่องบิน หรือค้นหาเรือ การพัฒนาเรดาร์ได้มีการพัฒนาอย่างจริงจังเพื่อใช้ในการสงครามเมื่อเกิดสงครามโลกครั้งที่สองขึ้น เรดาร์เป็นเครื่องมือที่มีประสิทธิภาพมากที่สุดอย่างหนึ่งที่ใช้ในการตรวจจับวัตถุและใช้ทางด้านอุตุนิยมวิทยา เรดาร์สามารถแสดงพื้นที่ ขนาด ทิศทาง ระยะห่าง และทิศทางการเคลื่อนที่ได้อย่างต่อเนื่องบนจอภาพ คลื่นแม่เหล็กไฟฟ้าซึ่งเรดาร์ตรวจจับอากาศยานส่งออกไปนั้นจะเป็นระบบพัลซ์ (PLUSE) โดยมีความยาวเป็นไมโครวินาที (10^{-6} วินาที) คลื่นแม่เหล็กไฟฟ้านี้จะถูกส่งออกไปโดยจานสายอากาศเมื่อกระทบกับเป้าหมาย คลื่นแม่เหล็กไฟฟ้าบางส่วนจะสะท้อนกลับมายังจานสายอากาศ ในขณะเดียวกันจานสายอากาศจะเป็นเครื่องรับสัญญาณที่สะท้อนกลับนั้น และสัญญาณจะได้รับการขยาย ดัดแปลง และแสดงเป็นภาพในจอแสดงผลของเรดาร์

คลื่นแม่เหล็กไฟฟ้าที่ส่งออกไปจากเรดาร์ตรวจอากาศจะเป็นระบบพัลซ์โดยมี2ลักษณะคือเป็นคลื่นแม่เหล็กไฟฟ้าที่ติดต่อกัน (COHERENT) หรือเป็นคลื่นแม่เหล็กไฟฟ้าที่ไม่ติดต่อกัน (NON-COHERENT) ในชนิดติดต่อกันคลื่นแม่เหล็กไฟฟ้า คลื่นแม่เหล็กไฟฟ้าที่ไม่ติดต่อกันหรือเรียกอีกชื่อหนึ่งว่าคลื่นเรดาร์นั้นจะถูกใช้เป็นคลื่นเปรียบเทียบกับคลื่นที่สะท้อนกลับมาที่รับได้จากจานสายอากาศเนื่องจากคลื่นที่สะท้อนกลับมาจะมีลักษณะเปลี่ยนแปลงไปสัมพันธ์กับการเคลื่อนที่ของเป้าหมาย (TARGET)โดยคุณสมบัติของความถี่แบบดอปเปลอร์ (DOPPLER FREQUENCY / SHIFT) หลักการนี้ได้นำมาสร้างเรดาร์แบบดอปเปอลร์ (DOPPER RADAR) สามารถตรวจหาหรือตรวจวัดได้ ทิศทางในการเคลื่อนที่ของวัตถุ การหาความเร็วของวัตถุในระบบไม่ติดต่อกันซึ่งการตรวจจับเป้าหมายและการกำหนดระยะห่างจะกระทำโดยการคำนวนจากช่วงเวลาที่คลื่นใช้ในการเดินทางไป-กลับ ทิศทางกำหนดจากทิศของคลื่นสะท้อนกลับ และลักษณะของเป้าหมายกำหนดจากลักษณะของคลื่นสะท้อนกลับ

ส่วนประกอบและการทำงานของเรดาร์นั้นเรดาร์จะทำงานคล้ายคลึงกับการทำงานของวิทยุ กล่าวคือประกอบด้วย เครื่องส่ง (TRANSMITTER) ทำการส่งคลื่นแม่เหล็กไฟฟ้าในช่วงของคลื่นวิทยุหรือคลื่นพาหะ (CARRIER WAVE) ออกไปสู่เครื่องรับ (RECEIVER) เครื่องรับจะรับคลื่นวิทยุแล้วแปลงเป็นคลื่นไฟฟ้าหรือคลื่นเสียงโดยจะทำงานตลอดเวลา แต่ในระบบพัลซ์จะไม่ทำการรับส่งตลอดเวลาเนื่องจากคลื่นเรดาร์ที่ส่งออกไปจะกระทบเป้าหมาย และสะท้อนกลับมาสู่เครื่อง

รับตลอดเวลา ทำให้ไม่สามารถวัดเวลาที่คลื่นเรดาร์ใช้ในการเดินทางจึงไม่สามารถทราบระยะทางและลักษณะของเป้าหมาย นอกจากนี้คลื่นเรดาร์ส่งออกไปยังอาจถูกรบกวนอยู่ตลอดเวลา

เรดาร์ในระบบพัลซ์จะส่งคลื่นแม่เหล็กไฟฟ้าหรือคลื่นเรดาร์นี้ออกไปเป็นจังหวะ หรือช่วงเรียกว่าพัลซ์ดังรูปที่ 2.1 การกำหนดความยาวของพัลซ์ได้เป็นสองแบบคือกำหนดแบบความยาว (L) หน่วยเป็นเมตร หรือกำหนดเป็นเวลา (T) หน่วยเป็นวินาที



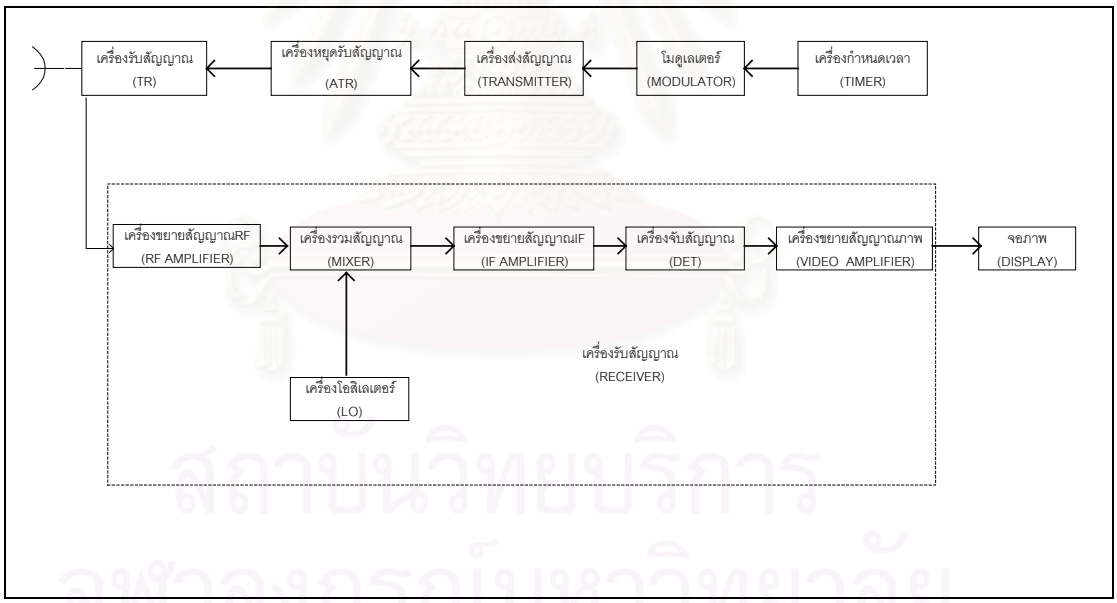
รูปที่ 2.1 ลักษณะPLUSEของคลื่นเรดาร์

2.1.1 **เครื่องส่ง** (TRANSMITTER)

รูปที่2.1 รูปการทำงานของเรดาร์ระบบพัลซ์โดยเริ่มจากเครื่องกำหนดเวลา (TIME OR TRIGGER GENERATOR) จะเป็นเครื่องกำหนดความถี่ของพัลซ์ โดยควบคุมการทำงานของโมดูเลเตอร์ (MODULATOR) ให้ส่งพลังงานไฟฟ้ากำลังสูง (HIGH VOLTAGE PULSE) ไปยังหลอดแมกนิตรอน (MAGNETRON) จะผลิตคลื่นแม่เหล็กไฟฟ้าในช่วงความถี่คลื่นวิทยุแล้วส่งความถี่คลื่นวิทยุแบบพัลซ์ (RADIO FREQUENCY ตัวย่อ RF) ไปยังเครื่องรับแบบเอทีอาร์ (ANTITRANSMIT RECEIVER ตัวย่อ ATR) แล้วจึงส่งสัญญาณต่อไปยังเครื่องรับสัญญาณ (TRANSMIT RECEIVER ตัวย่อ TR) ก่อนที่เครื่องรับสัญญาณจะส่งสัญญาณไปยังจานสายอากาศ (ANTENNA) ในขณะทำการส่งคลื่นเรดาร์อยู่นี้เครื่องรับสัญญาณจะทำหน้าที่ตัดวงจรในส่วนของเครื่องรับสัญญาณ เพื่อป้องกันความเสียหายที่จะเกิดขึ้นแก่รับสัญญาณ

2.1.2 **เครื่องรับ** (RECEIVER)

เมื่อคลื่นเรดาร์กระทบเป้าหมาย คลื่นบางส่วนสะท้อนกลับมาแหล่งกำเนิดของคลื่นและคลื่นที่สะท้อนกลับมานี้ถูกรับไว้โดยจานสายอากาศเดิม ซึ่งในขณะนั้นจะต่อวงจรอยู่กับเครื่องส่งสัญญาณและตัดวงจรจากเครื่องส่งชั่วขณะหนึ่งเพื่อทำการรับสัญญาณ และสัญญาณจะได้รับการขยายสัญญาณโดยหลอดขยายทีอาอาเอฟ (TR RF APLIFIER) ในเครื่องรับ ต่อจากนั้นจะนำสัญญาณที่ได้รับขยายสัญญาณแล้วผสมกับสัญญาณจากเครื่องโอสิเลเตอร์ (LOCAL OSILLATOR) โดยเครื่องโอสิเลเตอร์ทำหน้าที่แปลงสัญญาณนี้ให้เป็นความถี่ที่เป็นกลาง (INTERMEDIATE FREQUENCY ตัวย่อ IF) และนำสัญญาณที่ได้รับการแล้วผสมกับสัญญาณจากเครื่องโอสิเลเตอร์ส่งไปยังเครื่องขยายสัญญาณ โดยเครื่องขยายสัญญาณ แล้วส่งสัญญาณต่อไปยังเครื่องจับสัญญาณ (DETECTOR) เพื่อดัดแปลงสัญญาณความถี่ที่เป็นกลางให้เป็นสัญญาณภาพดังรูปที่ 2.2 โดยมีเครื่องขยายสัญญาณภาพทำการแปลงสัญญาณแล้วส่งไปยังจอภาพเพื่อแสดงผลต่อไป



รูปที่ 2.2 ระบบการทำงานของเรดาร์

2.1.3 **จานสายอากาศ** (ANTENNA)

ทำหน้าที่ส่งคลื่นเรดาร์ออกไปในบรรยากาศและรับคลื่นสะท้อนกลับ จานสายอากาศทั่วไปจะมีรูปร่างเป็นครึ่งวงกลม (PARABPLOID) จึงสามารถรวมพลังงานของคลื่นเรดาร์ให้อยู่ใน

ลักษณะแคบๆ และส่งคลื่นออกไปในบรรยากาศตามทิศทางที่กำหนด

2.1.4 **จอภาพ** (DISPLAY)

จอภาพของเรดาร์จะมีอยู่หลายชนิดจอภาพแต่ละชนิดจะมีส่วนประกอบที่ใกล้เคียงกัน คือ ประกอบด้วยหลอดภาพ (CATHODE RAY TUBE ตัวย่อ CRT) ซึ่งทำหน้าที่ผลิตอิเลคตรอนตามลักษณะสัญญาณที่ได้รับแล้วยิงไปยังจอหน้าปัดเรืองแสง (SCEEN) จอภาพนั้นจะเคลือบด้วยสารเรืองแสง 2 ชั้น ชั้นแรกเคลือบสารซึ่งเรืองแสงได้นานไม่เกิน 0.1 วินาที ชั้นสองที่เป็นผิวนอกเคลือบสารซึ่งเรืองแสงได้นานหลายวินาทีทำให้เกิดภาพขึ้นบนจอภาพเป็นเวลาสั้นๆ ในเรดาร์เครื่องหนึ่งอาจจะมีจอภาพจอเดียวหรือหลายจอประกอบกัน ขึ้นอยู่กับจุดมุ่งหมายและความต้องการในการใช้งาน

คลื่นเรดาร์ (RADAR WAVE) จากการที่เรดาร์รับสัญญาณสะท้อนกลับของคลื่นเรดาร์เมื่อคลื่นเรดาร์กระทบเป้าหมาย คลื่นเรดาร์เป็นคลื่นแม่เหล็กชนิดหนึ่งโดยคุณสมบัติแล้วพลังงานคลื่นแม่เหล็กไฟฟ้าเป็นพลังงานที่รวมระหว่างสนามไฟฟ้าและสนามแม่เหล็กตั้งฉากซึ่งกันและกัน และเคลื่อนที่ไปตามทิศทางของสนามไฟฟ้า โดยทั่วไปคลื่นเรดาร์เคลื่อนที่ในลักษณะลีเนียร์โพลาไร (LINERLY POLARIZED) ไม่ว่าจะเป็นทางแนวนอนหรือแนวตั้ง ทำให้เกิดการสั่นสะเทือนเป็นลูกคลื่น (SINUSOIDAL WAVE) เคลื่อนที่ไปในแนวระดับเดียว เมื่อจานสายอากาศสามารถส่งคลื่นออกไปในรูปของวงกลมหรือวงรีได้ คลื่นวงกลมจะเกิดจากคลื่นสองคลื่นที่มีความสูงเท่ากัน ส่วนคลื่นที่แผ่เป็นวงรีนั้นจะเกิดจากคลื่นทั้งสองจะมีความสูงไม่เท่ากัน

ความยาวของคลื่นเรดาร์ (WAVELENGTH OR RADAR FREQUENCY BANDS) เรดาร์โดยทั่วๆไปจะมีความถี่ของคลื่นในช่วง 220 MHz ถึง 35 GHz แต่จะมีเรดาร์บางชนิดเช่น เรดาร์คลื่นฟ้า (SKYWAVERADAR) มีความถี่ของคลื่นต่ำมากที่ 4-5 MHz และเรดาร์คลื่นพื้นดิน (GROUNDWAVE RADAR) มีความถี่ของคลื่นต่ำแค่เพียง 2 MHz ส่วนเรดาร์มิลิเมตร (MILLIMETER RADAR) มีความถี่ของคลื่นสูงถึง 94 GHz

คลื่นเรดาร์ที่มีความยาวคลื่นสั้นจะมีความสามารถในการแยกแยะ (RES0LLUSION) เป้าหมายได้ดีกว่า แต่จะมีข้อเสียคือคลื่นเรดาร์ที่มีความยาวคลื่นสั้นเมื่อผ่านเข้าไปในชั้นบรรยากาศจะมีการลดลงของพลังงานมากกว่าคลื่นเรดาร์ที่มีความยาวคลื่นยาวกว่า ดังนั้นการเลือกความยาวคลื่นเรดาร์จึงต้องเป็นไปตามจุดประสงค์ของการใช้งาน

2.2 **สูตรคำนวณหาพื้นที่ร่วมของเรดาร์** [3]

การคำนวณหาพื้นที่ร่วมของเรดาร์นั้น สามารถทำได้โดยการคำนวณหาระยะห่างระหว่างจุดศูนย์กลางของเรดาร์แต่ละเครื่อง โดยถ้าระยะห่างของเรดาร์คู่นั้นน้อยกว่ารัศมีของเรดาร์เครื่องที่

1รวมกับรัศมีของเรดาร์เครื่องที่2แสดงว่าเรดาร์คู่นั้นมีพื้นที่ร่วม สูตรของการคำนวณจะเป็นสูตรการหาด้านที่ 3 ของสามเหลี่ยมโดยมีสูตรดังต่อไปนี้

สูตรการหาด้านที่ 3 ของสามเหลี่ยม $R=\sqrt{x^2+y^2}$

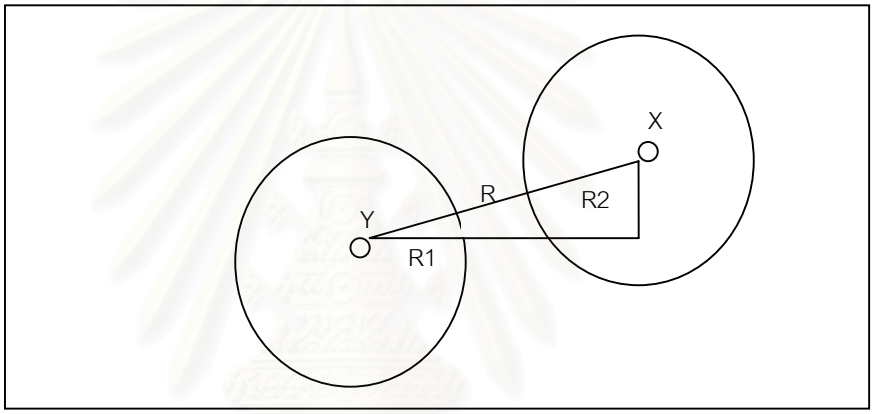
โดย R = ด้านที่ 3 ของสามเหลี่ยม

X = ด้านที่ 1 ของสามเหลี่ยม

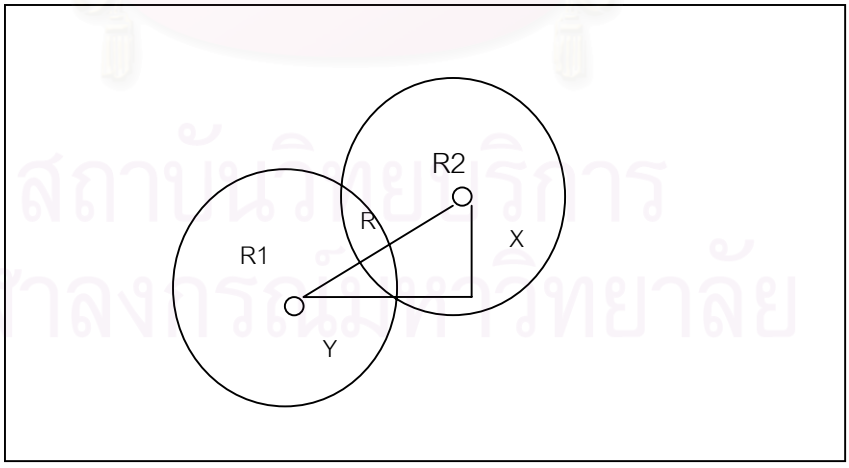
Y = ด้านที่ 2 ของสามเหลี่ยม

ผลที่ได้ถ้า R > R1+R2 => เรดาร์ทั้งสองเครื่องไม่มีพื้นที่พื้นที่ร่วมกันแสดงในรูปที่ 2.3 และ

ผลที่ได้ถ้า R < R1+R2 => เรดาร์ทั้งสองเครื่องมีพื้นที่พื้นที่ร่วมกันแสดงในรูปที่ 2.4



รูปที่ 2.3 แสดงการที่เรดาร์ที่ไม่มีพื้นที่ร่วม



รูปที่ 2.4 แสดงการที่เรดาร์ที่มีพื้นที่ร่วม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.3 แนวทางในการคำนวณเวลาของเรดาร์จากการหมุนของเรดาร์ [3]

จากข้อมูลเรดาร์หมุนหนึ่งรอบใช้เวลาประมาณ 4 วินาที และรัศมีของเรดาร์ 75 ไมล์ทะเล ดังนั้นเราสามารถนำเอาข้อมูลที่เรามีมาทำการคำนวณหาเวลาโดยประมาณที่เรดาร์ตรวจอากาศยานพบ โดยการคำนวณหามุมองศาของอากาศยานจากตำแหน่งทางแกน X และแกน Y เมื่อได้มุมองศาแล้วจะนำมาคำนวณให้เวลาเป็นวินาที การหมุนหนึ่งรอบของเรดาร์จะเป็นมุม 360 องศา และใช้เวลาประมาณ 4 วินาที ดังนั้นเราจึงสามารถคำนวณหาเวลาที่พบอากาศยานได้ โดยมีสูตรดังต่อไปนี้คือ

สูตรการหามุมของเรดาร์ $\text{Tan}(\theta) = \frac{X}{Y}$

โดย X = ระยะทางของเส้นรุ้ง

Y = ระยะทางของเส้นแวง

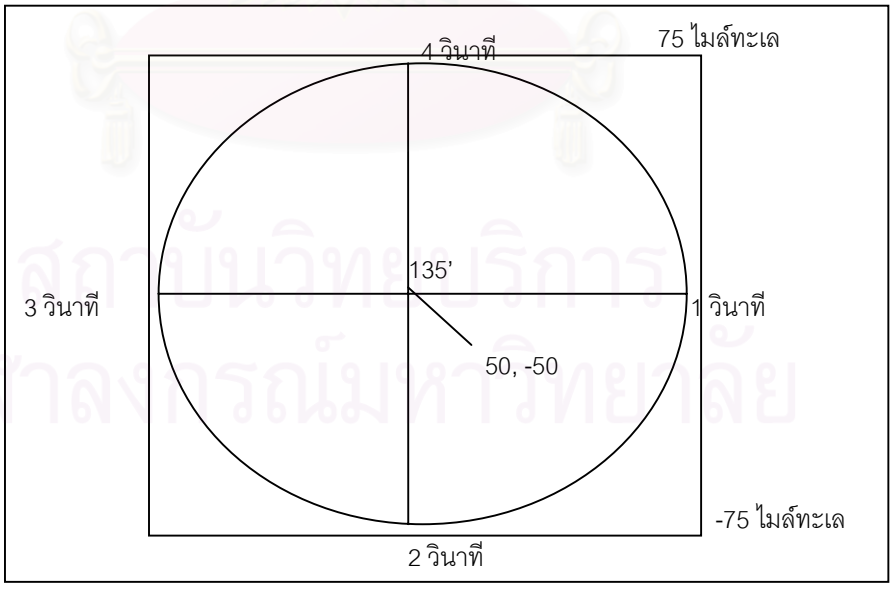
ตัวอย่างจากการสมมุติ ถ้า X = 50 NM, Y = -50 NM

$$\text{Tan}(\theta) = \frac{50}{-50}$$

$$\text{Tan}(\theta) = -1$$

$$\theta = 135 \text{ องศา}$$

เมื่อเราทราบมุมเราจึงสามารถคำนวณเวลาได้คือ 135÷360×4 = 1.5 วินาที จากรูปที่ 2.5



รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างเวลาและตำแหน่งของอากาศยาน

2.4 แนวทางในการขจัดความซ้ำซ้อนของอากาศยาน [3]

การขจัดความซ้ำซ้อนของอากาศยานนั้นเราจะทำการเมื่ออากาศยานผ่านเข้ามาในพื้นที่ร่วมของสัญญาณเรดาร์เท่านั้น โดยอาศัยจากสูตรการคำนวณหาระยะทางของอากาศยานจากจุดศูนย์กลางของเรดาร์ถ้าระยะห่างจากจุดศูนย์กลางของเรดาร์เครื่องที่ 1 และเรดาร์เครื่องที่ 2 น้อยกว่ารัศมีของเรดาร์แสดงว่าอากาศยานอยู่ในพื้นที่ร่วมของสัญญาณเรดาร์ ดังนั้นเราจึงต้องทำการขจัดความซ้ำซ้อนของอากาศยาน โดยการคำนวณหาความแตกต่างของเวลาที่ได้จากการคำนวณจากตำแหน่งของอากาศยาน เพื่อนำมาใช้ในการคำนวณหาตำแหน่งที่น่าจะเป็นไปได้ของอากาศยาน โดยมีสูตรดังต่อไปนี้

สูตรการหาช่วงเวลาของเวลา $\triangle T = T1 - T2$

โดย $\triangle T$ = ช่วงเวลาของเวลา

T1 = เวลา ของเรดาร์ที่ 1

T2 = เวลาของเรดาร์ที่ 2

เมื่อได้ความแตกต่างของเวลาแล้วจะนำมาแทนค่าสูตรเพื่อหาตำแหน่งที่น่าจะเป็นของอากาศยาน โดยจะทำการคำนวณจากเรดาร์เครื่องที่ 1 มีสูตรดังต่อไปนี้

สูตรการหาตำแหน่ง $Px = P1x + (Sx \times \triangle T)$

โดย Px = ตำแหน่งที่น่าจะเป็นของอากาศยานทางแกนX

P1x = ตำแหน่งเริ่มต้นของอากาศยานทางแกนX

Sx = ความเร็วของอากาศยานทางแกนX

$\triangle T$ = ช่วงเวลาของเวลา

และสูตรความเร็วทางแกนYโดยมีสูตรดังต่อไปนี้

สูตรการหาตำแหน่ง $P = P1y + (Sy \times \triangle T)$

โดย Py = ตำแหน่งที่น่าจะเป็นของอากาศยานทางแกนY

P1y = ตำแหน่งเริ่มต้นของอากาศยานทางแกนY

Sy = ความเร็วของอากาศยานทางแกนY

$\triangle T$ = ช่วงเวลาของเวลา

เมื่อนำตำแหน่งที่น่าจะเป็นของอากาศยานทางแกนXและแกนYของเรดาร์เครื่องที่ 1 มาเปรียบเทียบกับ ตำเหน่งของเรดาร์เครื่องที่ 2 ถ้าค่าเท่ากันหรือใกล้เคียงกัน สามารถสรุปได้ว่าอากาศยานทั้งสองเป็นอากาศยานเดียวกัน

2.5 แนวทางในการสร้างข้อมูลอากาศยาน [3]

การสร้างข้อมูลอากาศยานในขั้นแรกนั้นต้องทำการสร้างรัศมีของเรดาร์ เมื่อได้รัศมีของเรดาร์แล้วต้องทำการคำนวณหาตำแหน่งของอากาศยานว่าอยู่ในรัศมีของเรดาร์หรือไม่ โดยสูตรดังต่อไปนี้

สูตรการหาด้านที่3ของสามเหลี่ยม $R = \sqrt{x^2 + y^2}$

โดย R = ด้านที่ 3 ของสามเหลี่ยม

X = ด้านที่ 1 ของสามเหลี่ยม

Y = ด้านที่ 2 ของสามเหลี่ยม

ผลที่ได้ถ้า R < 4500 => อากาศยานอยู่ในรัศมีของเรดาร์ และ

ผลที่ได้ถ้า R > 4500 => อากาศยานไม่อยู่ในรัศมีของเรดาร์

ในการสร้างข้อมูลจำลองข้อมูลของเรดาร์นั้นต้องทำการ คำนวณเวลาของอากาศยานจากข้อมูลของอากาศยานตำแหน่งของอากาศยาน การคำนวณความเร็วของอากาศยานมีหน่วยเป็นพิลิปดาต่อวินาที การคำนวณสามารถนำสูตรการแปลงหน่วยมาใช้ได้ดังต่อไปนี้คือ

โดย ความเร็ว 1 ไมล์ทะเล = 1.852 กิโลเมตร

1 ชั่งโมง = 3600 วินาที

ไมล์ทะเล = 6,076 ฟุต หรือระยะหนึ่งลิปดาแห่งเส้นรอบวงของโลก

สูตรการคำนวณหาความเร็วของอากาศยานสามารถคำนวณได้ในสูตรดังต่อไปนี้

ความเร็วของอากาศยานต่อวินาที $S = AS \div KMR \div 3600$

โดย S = ความเร็วของอากาศยานต่อวินาที

AS = ความเร็วของอากาศยาน

KMR = ความเร็ว 1 ไมล์ทะเลแปลงเป็นกิโลเมตร

เพื่อที่จะได้ความเร็วที่มีหน่วยย่อยลงสามารถหาได้จากสูตรที่ 8 ความเร็วของอากาศยานต่อส่วนของวินาที

สูตรความเร็วของอากาศยานต่อส่วนของวินาที $STS = AS \div KMR \div 3600 \div TSS$

โดย STS = ความเร็วของอากาศยานต่อส่วนของวินาที

AS = ความเร็วของอากาศยาน

KMR = ความเร็ว 1 ไมล์ทะเลแปลงเป็นกิโลเมตร

TSS = ส่วนของวินาที

การคำนวณมุมของอากาศยานโดยในการหมุนหนึ่งรอบของเรดาร์แบ่งมุม 360 องศา ในหนึ่งรอบการหมุนของเรดาร์จะใช้เวลา 4 วินาทีโดยประมาณ เพื่อเพิ่มความแม่นยำของข้อมูลเวลาควรใช้อยู่ที่ 1 ใน 200 ส่วนของวินาที จากสูตรดังต่อไปนี้

สูตรมุมของเรดาร์ต่อเวลา 1 ใน 200 ส่วนของวินาที Radius = R ÷ (S × TSS)

โดย Radius = มุมของเรดาร์ต่อเวลา 1 ใน 200 ส่วนของวินาที

R = มุมของเรดาร์

S = การหมุนของเรดาร์1รอบจะใช้เวลา 4 วินาที

TSS = ส่วนของวินาที

การคำนวณตำแหน่งของเรดาร์ที่ตรวจอากาศยานพบ โดยการคำนวณตำแหน่งของอากาศยานทั้งเส้นรุ้งและเส้นแวงสูตรดังต่อไปนี้

สูตรตำแหน่งเส้นแวง Px1 = Px0 + (STS × RN × COS(θ))

โดย Px1 = ตำแหน่งเส้นแวงเดิม

Px0 = ตำแหน่งเส้นแวงใหม่

STS = ความเร็วของอากาศยานต่อส่วนของวินาที

RN = รอบที่ของเรดาร์

θ = มุมของอากาศยานกับศูนย์กลางของเรดาร์

สูตรตำแหน่งเส้นรุ้ง Py1 = Py0 + (STS × RN × SIN(θ))

โดย Py1 = ตำแหน่งเส้นรุ้งเดิม

Py0 = ตำแหน่งเส้นรุ้งใหม่

STS = ความเร็วของอากาศยานต่อส่วนของวินาที

RN = รอบที่ของเรดาร์

θ = มุมของอากาศยานกับศูนย์กลางของเรดาร์

2.6 สูตรการแปลงข้อมูลแผนที่ประเทศไทย

แผนที่ประเทศไทยจะประกอบด้วยตัวอักษร4ตัวตามด้วยตัวเลข4ตัวเช่น UGMF4448 ในการคำนวณหาตำแหน่งของประเทศไทยโดยตำแหน่งแรกจะขึ้นต้นด้วย U หรือ V และตำแหน่งที่สองจะขึ้นต้นด้วย U หรือ V ตำแหน่งที่สามและตำแหน่งที่สี่จะเป็นตัวอักษรในภาษาอังกฤษ ส่วนตำแหน่งที่ห้าถึงตำแหน่งที่แปดจะเป็นตัวเลขโดยจะมีสูตรในการคำนวณตำแหน่งทางภูมิศาสตร์ได้ดังต่อไปนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. ตำแหน่งที่หนึ่งขึ้นต้นด้วย U ค่าเริ่มต้นของเส้นแวงจะเป็น90องศา
2. ตำแหน่งที่หนึ่งขึ้นต้นด้วย V ค่าเริ่มต้นของเส้นแวงจะเป็น105องศา
3. ตำแหน่งที่สามเป็น A B C D E F G หรือ Hตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ (ASCII)ของ(ตัวอักษรตำแหน่งที่สาม)ลบด้วย65รวมกับค่าเริ่มต้นของเส้นแวง
4. ตำแหน่งที่สามเป็นJ K L M หรือ N ตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ของ(ตัวอักษรตำแหน่งที่สาม) ลบด้วย66รวมกับค่าเริ่มต้นของเส้นแวง
5. ตำแหน่งที่สามเป็นP Q หรือ R ตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ของ (ตัวอักษรตำแหน่งที่สาม) ลบด้วย67รวมกับค่าเริ่มต้นของเส้นแวง
6. ตำแหน่งของเส้นแวงมีหน่วยเป็นลิปดา = ตำแหน่งของเส้นแวง × 60 + ตำแหน่งที่ห้าและตำแหน่งที่หก
7. ตำแหน่งที่หนึ่งขึ้นต้นด้วยGค่าเริ่มต้นของเส้นรุ้งจะเป็น0องศา
8. ตำแหน่งที่หนึ่งขึ้นต้นด้วย V ค่าเริ่มต้นของเส้นรุ้งจะเป็น15องศา
9. ตำแหน่งที่สามเป็น A B C D E F G หรือ H ตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ของ (ตัวอักษรตำแหน่งที่สาม) ลบด้วย65รวมกับค่าเริ่มต้นของเส้นรุ้ง
10. ตำแหน่งที่สามเป็น J K L M หรือ N ตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ของ (ตัวอักษรตำแหน่งที่สาม) ลบด้วย66รวมกับค่าเริ่มต้นของเส้นรุ้ง
11. ตำแหน่งที่สามเป็น P Q หรือ R ตำแหน่งของเส้นรุ้งจะเท่ากับค่าแอสกี้ของ (ตัวอักษรตำแหน่งที่สาม) ลบด้วย67รวมกับค่าเริ่มต้นของเส้นรุ้ง
12. ตำแหน่งของเส้นรุ้งมีหน่วยเป็นลิปดา = ตำแหน่งของเส้นรุ้ง × 60 + ตำแหน่งที่เจ็ดและตำแหน่งที่แปด

ตัวอย่างการคำนวณตำแหน่งทางภูมิศาสตร์จากตัวอย่าง UGMF4448 ในการคำนวณหาตำแหน่งของเส้นแวงจะเริ่มจากการคำนวณตำแหน่งแรกมีค่าเป็นUเท่ากับ 90 องศา ตำแหน่งที่ 3 มีค่าเป็นMจะเท่ากับ 77 – 66 หรือเท่ากับ11องศา ตำแหน่งของเส้นแวงมีหน่วยเป็นลิปดา = (101 × 60) + 44 = 6104 ลิปดา ในการคำนวณหาตำแหน่งของเส้นรุ้งจะเริ่มจากการคำนวณตำแหน่งที่ 2 มีค่าเป็น G จะเท่ากับ0องศาตำแหน่งที่4มีค่าเป็น F เท่ากับ 70 – 65 หรือเท่ากับ 5 องศา ตำแหน่งของเส้นรุ้งมีหน่วยเป็นลิปดา = 5 × 60 + 48 = 348 ลิปดา ตำแหน่งทางภูมิศาสตร์ UGMF4448 จะเท่ากับ เส้นแวงที่ 6104 เส้นรุ้งที่ 348

2.7 แนวทางในการทำให้เวลาของเรดาร์แต่ละเครื่องตรงกัน [4]

จากการที่เรดาร์ที่แต่ละเครื่องมีเวลาไม่ตรงกันนั้นจะทำการขจัดความซ้ำซ้อนของอากาศ

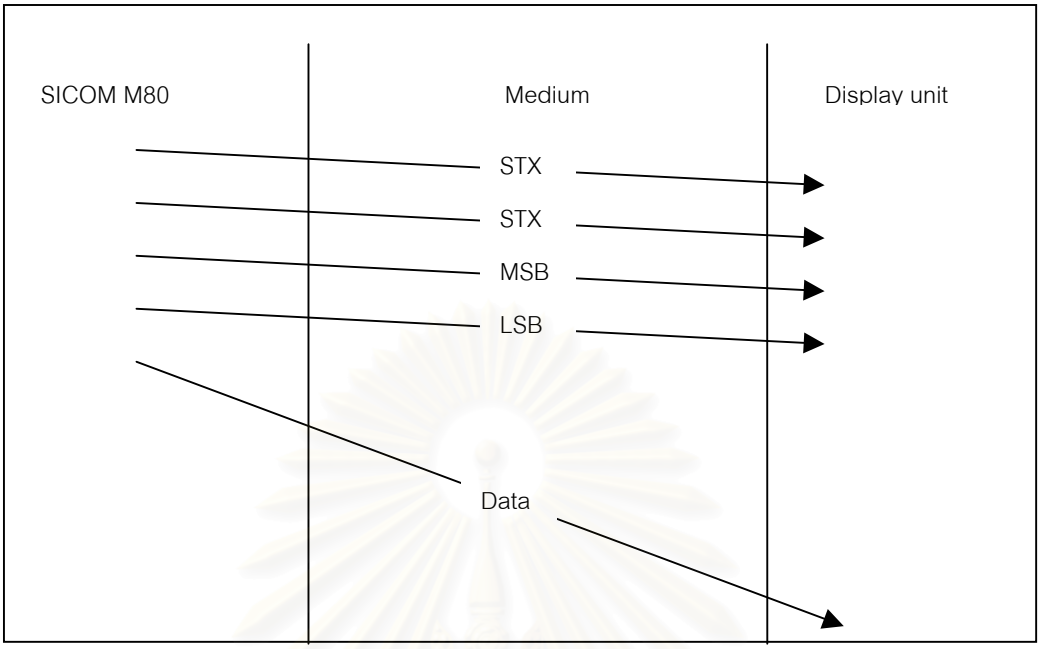
ยานได้ไม่ถูกต้อง แนวทางในการทำให้เวลาของเรดาร์แต่ละเครื่องตรงกันมีหลายแนวทางแต่ละวิธีมีความยากง่ายต่างกันไป แต่ที่เราจะนำมาใช้ในการทำให้เวลาของเรดาร์แต่ละเครื่องตรงกันนั้นจะใช้เวลามาตรฐาน (STANDARD-TIME) ในการอ้างอิงการใช้เวลาจากสถานีวิทยุที่กระจายเสียงเวลามาตรฐาน (RADIO CLOCK) เมื่อคอมพิวเตอร์และเรดาร์ทุกเครื่องมีเวลาเป็นมาตรฐานเดียวกันแล้ว เราจะสามารถทำงานได้ตามปรกติและจะต้องมีการทำการให้เวลาของเรดาร์แต่ละเครื่องตรงกันอยู่เสมอ โดยจะต้องตั้งเวลาตามเวลามาตรฐานทุกเดือนหรือตามเห็นสมควร

2.8..โปรโตคอลการแสดงภาพสถานการณ์ทางอากาศระยะไกล (REMOTE AIR SITUATION DISPLAY PROTOCOL) [5]

การส่งสัญญาณของโปรโตคอลการแสดงภาพสถานการณ์ทางอากาศระยะไกลนั้น จะส่งออกมาทางช่องอนุกรม (serial port) ด้วยความเร็ว 2400 บิตต่อวินาที และ สัญญาณจะเป็นแบบอสมวาร (asynchronous) โดยสัญญาณจะส่งออกมาเป็นวงจรโดยใช้เวลาในแต่ละวงจร 4 วินาทีโดยสัญญาณจะขึ้นอยู่กับการหมุนของเรดาร์ การส่งสัญญาณจะส่งออกมามากที่สุดที่ 8,400 บิตและในสัญญาณที่ส่งออกมานี้จะเป็นข้อมูล 840 ไบต์ การส่งสัญญาณจะเป็นแบบทางเดียว เมื่อเรดาร์ตรวจจับอากาศยานได้เรดาร์จะส่งสัญญาณมายังเครื่องไซคอมพ์เอ็มแปดสิบ (SICOMP M80) เพื่อทำการประมวลผลเมื่อเครื่องไซคอมพ์เอ็มแปดสิบ ทำการประมวลผลเสร็จเรียบร้อยแล้วจะส่งสัญญาณต่อไปยังเครื่องแสดงผล

การส่งข้อมูลจะแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกคือหัวของสัญญาณในส่วนนี้ประกอบด้วยข้อมูลที่จะบอกว่ามีการเริ่มส่งข้อมูลจะเริ่มต้นโดยการส่ง ASCII#2 (STX) และความยาวของข้อมูล ส่วนที่สองเป็นข้อมูลของเรดาร์ซึ่งจะเป็นสถานที่ตั้งของเรดาร์หรือข้อมูลของอากาศยาน รูปแบบของการส่งสัญญาณนั้นโดยทั่วๆไปสามารถแสดงออกได้ดังในรูปที่ 2.6 จะแสดงถึงการส่งสัญญาณซึ่งมองในลักษณะของเวลาส่วนตารางที่ 2.1 แสดงเขตของข้อมูล

จากตารางที่ 2.1 แสดงถึงการส่งสัญญาณสถานที่ตั้งของเรดาร์โดยข้อมูลจะแบ่งออกเป็นสองส่วนคือ ส่วนแรกจะเป็นส่วนของส่วนหัวของข้อมูล (Header) จะประกอบด้วย 2 ส่วนย่อยคือ ส่วนย่อยแรกจะเป็นส่วนที่แสดงว่ามีการเริ่มต้นส่งสัญญาณ ส่วนย่อยที่สองจะเป็นส่วนที่แสดงความยาวของข้อมูลในการส่งครั้งนั้นๆ ส่วนที่สองจะเป็นสถานที่ตั้งของเรดาร์ โดยสถานที่ตั้งของเรดาร์จะเป็นตำแหน่งที่ตั้งทางภูมิศาสตร์ของเรดาร์



รูปที่ 2.6 ขั้นตอนของการส่งสัญญาณเพื่อการแสดงผล

เขตข้อมูล	ข้อความภายใน (content)	ความหมาย
1	STX(ASCII 2)	ส่วนหัวของสัญญาณ
2	STX(ASCII 2)	ส่วนหัวของสัญญาณ
3	Block length(MSB)	ความยาวของสัญญาณ
4	Block length(LSB)	ความยาวของสัญญาณ
5	ข้อมูล	ข้อมูลอากาศยานที่ 1
ถึง	ข้อมูล	ข้อมูลอากาศยานที่ ..
N	ข้อมูล	ข้อมูลอากาศยานที่ n

ตารางที่ 2.1 การส่งสัญญาณของระบบเรดาร์

เขตข้อมูล	ข้อความภายใน(content)	ความหมาย
1	STX	ส่วนหัวของสัญญาณ
2	STX	ส่วนหัวของสัญญาณ
3	Block length	ความยาวของสัญญาณ

เขตข้อมูล	ข้อความภายใน(content)	ความหมาย
4	Block length	ความยาวของสัญญาณ
5	Message – identification	หลักฐานของสัญญาณ
6	Number of symbols	หมายเลขของเรดาร์
7	X-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นรุ้ง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
8	X-position (LSB)	
9	Y-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นแวง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
10	Y-position (LSB)	
11	Symbol label	เครื่องหมายของเรดาร์
12	Present not used	ไม่ใช้
13	Minutes(6 bit) | seconds (2 MSB)	ตำแหน่งและทิศทางของเรดาร์เส้นรุ้งมีหน่วยเป็นองศา ลิปดา พิลิปดา
14	Seconds (4 LSB) | direction (4 bit)	
15	Degree of latitude (MSB)	
16	Degree of latitude (LSB)	
17	Minutes(6 bit) | seconds (2 MSB)	ตำแหน่งและทิศทางของเรดาร์เส้นแวงมีหน่วยเป็นองศา ลิปดา พิลิปดา
18	Seconds (4 LSB) | direction (4 bit)	
19	Degree of longtitude (MSB)	
20	Degree of longtitude (LSB)	

ตารางที่ 2.2 ข้อมูลสถานที่ตั้งของเรดาร์

ตารางที่ 2.2 แสดงถึงการส่งสัญญาณข้อมูลของเรดาร์โดยข้อมูลจะแบ่งออกเป็นสองส่วน คือส่วนแรกจะเป็นส่วนของส่วนหัวของข้อมูล (Header) เพื่อแสดงถึงการเริ่มต้นส่งสัญญาณและขนาดของข้อมูลในการส่งข้อมูลครั้งนี้ ส่วนที่สองจะเป็นข้อมูลของเรดาร์เช่น หมายเลขของเรดาร์ เครื่องหมายของเรดาร์ และตำแหน่งพิกัดสถานที่ตั้งของเรดาร์

ตารางที่ 2.3 ข้อมูลของอากาศยาน

เขตข้อมูล	**ข้อความภายใน(content)**	**ความหมาย**
1	STX	ส่วนหัวของสัญญาณ
2	STX	ส่วนหัวของสัญญาณ
3	Block length	ความยาวของสัญญาณ
4	Block length	ความยาวของสัญญาณ
5	n-time the data block of a track, maximum 50 tracks message – identification	จำนวนของอากาศยาน
6	Track reference number	ตัวเลขอ้างอิงเส้นทางการบิน
7	X-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นรุ้ง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
8	X-position (LSB)	
9	Y-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นแวง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
10	Y-position (LSB)	
11	X-speed (MSB)	ความเร็วของอากาศยานเส้นรุ้ง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
12	X-speed (LSB)	
13	Y-speed (MSB)	ความเร็วของอากาศยานเส้นแวง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
14	Y-speed (LSB)	
15	Track number (MSB)	ตัวเลขของเส้นทางการบิน
16	Track number (LSB)	
17	Track - Identification	หลักฐานของเส้นทางการบิน
18	Radar – site – identification	หมายเลขเรดาร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 2.3 แสดงถึงการส่งสัญญาณข้อมูลของเส้นทางการบินของอากาศยานโดยข้อมูลจะแบ่งออกเป็นสองส่วนคือส่วนแรกจะเป็นส่วนของส่วนหัวของข้อมูล (Header) เพื่อแสดงถึงการเริ่มต้นส่งสัญญาณ ส่วนที่สองจะเป็นข้อมูลของเส้นทางการบินของอากาศยาน จะประกอบด้วยตัวเลขอ้างอิงเส้นทางการบิน ตำแหน่งของอากาศยานทางภูมิศาสตร์ และความเร็วของอากาศยานที่มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล

รายละเอียดของข้อมูลข้อมูลของอากาศยาน

ตัวเลขอ้างอิงเส้นทางการบิน (Track reference number)	1 ถึง 50	ไบต์
ตำแหน่ง(Position)	–2048 ถึง +2047(ไม่ถูกนำมาใช้)	
ความเร็ว(Speed)	0 ถึง 2000 ไมล์ทะเล(Nautical miles)	

หลักฐานของเส้นทางการบิน (Track –Identification)

หลักฐานของเส้นทางการบิน (Track – Identification)	ประเภท (class)	เครื่องหมาย (sign)
มิตร(Friendly)	F	F
เครื่องบินสกัดกั้น(Interceptor)	F	I
ศัตรู(Hostile)	H	H
สัญญาณถูกรบกวน(Lamentation)	H	J
เป้าหมาย(Target)	H	T
ไม่ทราบสถานะ(Unknown)	H	U
สนใจเป็นพิเศษ(Kilo (Special Interest))	F	K
รังสีเอกซ์(XRAY (Unknown))	H	X
คอยคำสั่ง(Pending)	F	P
ฉุกเฉิน(Emergency)	F	E
ไม่มีการเคลื่อนไหว(Zombie)	F	Z
ตัวเลขของเส้นทางการบิน (Track numbers)	001 ถึง 999	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.9 ข้อมูลของระบบเรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวี(DR172 ADV)

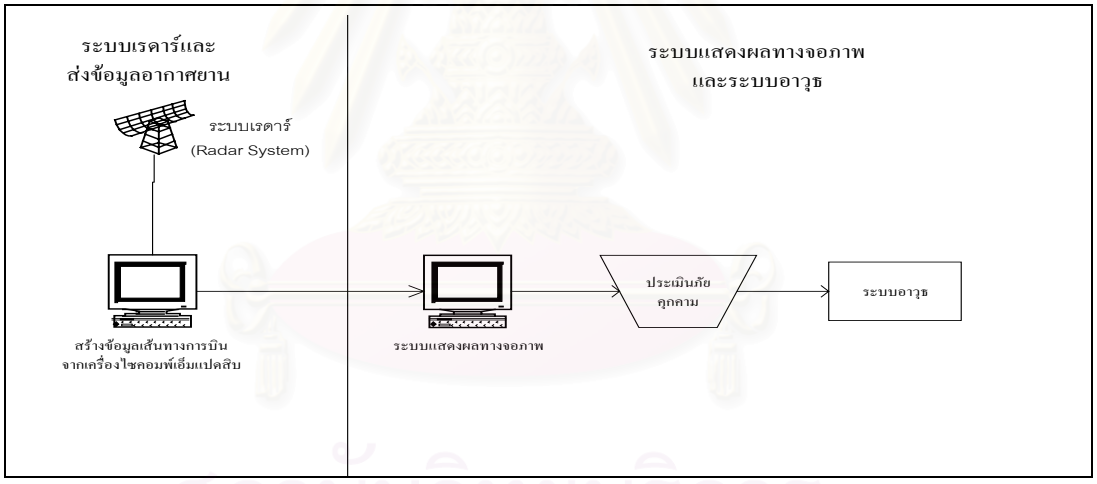
มีคุณสมบัติที่สำคัญคือ

1 ย่านความถี่	แบนด์(1215 ถึง 1365 MHz)	
2 โหมดการทำงาน	Diversity หรือ Single Channel, MTI	
3 PRF	1.5 และ 1 kHz	
4 ระยะการทำงานของเรดาร์	1 ถึง 75 ไมล์ทะเล(1 ไมล์ทะเล = 6076 ฟุต หรือ 1 ลิปดา)	
5 Blind speed zone	1180 ถึง 1319 ก.ม./ชม.	
6 ระบบ ECCM	Constant False Alarm Rate,Amplitude versus azimuth,	
	Pulse Length Discriminator,	
	Interpulse suppression	
7 IFF	โหมด 1, 2, 3/A และ C	
8 Power supply	220/380 Vac 3 เฟส ความถี่ 50 Hz $\pm$ 5 %	
9 ขนาดของเรดาร์	Antenna shelter	5065 x 2673 x 2590 ม.ม.
	Operator shelter	6450 x 2421(2600) x 2438 ม.ม.
	Power generator	2500 x 1385 x 2250 ม.ม.
	Transport device	5000 x 2750 x 1500 ม.ม.
10 การหมุนของเรดาร์	การหมุน 1 รอบเรดาร์จะใช้เวลา 4 $\pm$ 0.25 วินาที	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การวิเคราะห์และออกแบบ

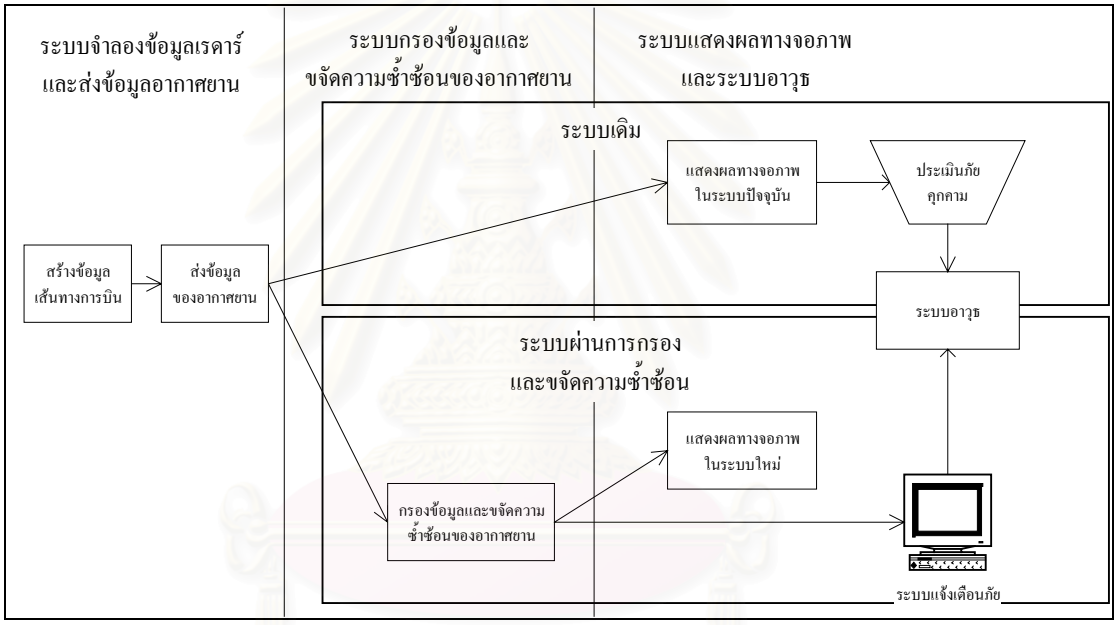
การสร้างโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง การทำงานของระบบเรดาร์ดีอาหนึ่งร้อยเจ็ดสิบสองเอดีวีนั้นจะเริ่มจากเมื่อเรดาร์ตรวจจับอากาศยานได้ เรดาร์จะส่งข้อมูลมายังเครื่องไซคอมพ์เอ็มแปดสิบเพื่อทำการประมวลผล โดยเครื่องไซคอมพ์เอ็มแปดสิบจะนำข้อมูลของอากาศยานที่ตรวจจับได้นั้นมาเปรียบเทียบกับแผนการบินของอากาศยานและชนิดของอากาศยานฝ่ายเรา เพื่อกำหนดฝ่ายโดยการกำหนดฝ่ายจะแบ่งออกเป็น 2 ฝ่ายคือ อากาศยานฝ่ายเราแทนสัญญาลักษณ์ด้วย F และอากาศยานฝ่ายข้าศึกแทนสัญญาลักษณ์ด้วย H และคำนวณตำแหน่งและความเร็วของอากาศยาน เมื่อเครื่องไซคอมพ์เอ็มแปดสิบประมวลผลข้อมูลเรียบร้อยแล้วจะส่งข้อมูลในรูปแบบของโปรโตคอล การแสดงผลระยะไกลเพื่อมาแสดงผลทางจอภาพ ดังแสดงในรูปที่ 3.1 เพื่อใช้ในการประเมินภัยคุกคามเมื่อเห็นว่าอากาศมีภัยคุกคามระบบอาวุธจะดำเนินการทางยุทธวิธีต่อไป



รูปที่ 3.1 ระบบเรดาร์และการแสดงผล

ในการทำการวิจัยครั้งนี้ผู้วิจัยไม่สามารถนำข้อมูลจริงของเรดาร์มาใช้ในการวิจัยได้ จึงต้องทำการสร้างข้อมูลจำลองมาเพื่อใช้ในการวิจัย ในการสร้างข้อมูลนั้นผู้ทำการวิจัยได้สร้างข้อมูลจำลอง โดยนำโปรโตคอลการแสดงภาพสถานการทางอากาศระยะไกลมาใช้เป็นต้นแบบในการสร้างข้อมูลจำลองเพื่อความสะดวกและผลที่ถูกต้องในการทำการวิจัย ผู้ทำการวิจัยได้ออกแบบการสร้างข้อมูลจำลองขึ้นโดยสามารถกระทำได้ด้วยการใช้กราฟฟิกเพื่อสร้างข้อมูล และข้อมูลที่ได้จากการการสร้างขึ้นมานี้นั้นจะถูกใช้เป็นข้อมูลของระบบโปรแกรมต้นแบบเพื่อการกรองและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง เพื่อให้เห็นภาพของอากาศยาน

ต้องมีการสร้างโปรแกรมการแสดงผลทางจอภาพแบบกราฟฟิก การที่เห็นภาพของอากาศยานเป็นแบบกราฟฟิกทำให้เราเห็นปัญหาที่เกิดของระบบที่ใช้ในปัจจุบัน และการแก้ปัญหาของระบบปัจจุบันหลังจากการนำโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องมาใช้แล้ว เพื่อให้เห็นประสิทธิภาพของระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง ต้องมีการสร้างระบบการประมวลผลความถูกต้องของระบบ โดยความถูกต้องจะประเมินออกมาในรูปของเปอร์เซ็นต์ความถูกต้องของระบบสะสมเพื่อแสดงให้เห็นการทำงานโดยภาพรวมของระบบนับจากการเริ่มต้นการทำงานว่าระบบมีความถูกต้องรวมเป็นเท่าไร โดยในรูปที่ 3.2 จะแสดงแนวคิด



รูปที่ 3.2 ระบบการขจัดความซ้ำซ้อนและการกรองข้อมูลและการแสดงผล

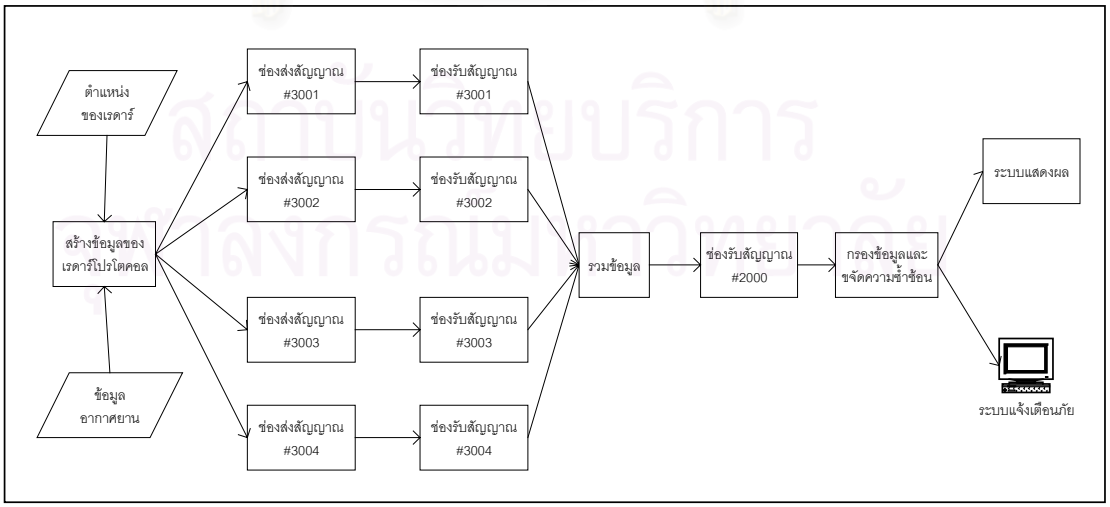
ในการสร้างระบบจำลองข้อมูลเส้นทางการบิน และการแสดงผลทางจอภาพเพื่อใช้ในโปรแกรมต้นแบบเพื่อการกรองข้อมูล และขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องนั้นสามารถแบ่งออกเป็น 5 ส่วน

1. การออกแบบระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยาน
2. การออกแบบระบบสร้างข้อมูลเรดาร์และการส่งข้อมูล
3. การออกแบบระบบแสดงผลทางจอภาพ
4. การออกแบบแฟ้มข้อมูลและตารางข้อมูล

5. การออกแบบแสดงผลทางจอภาพ

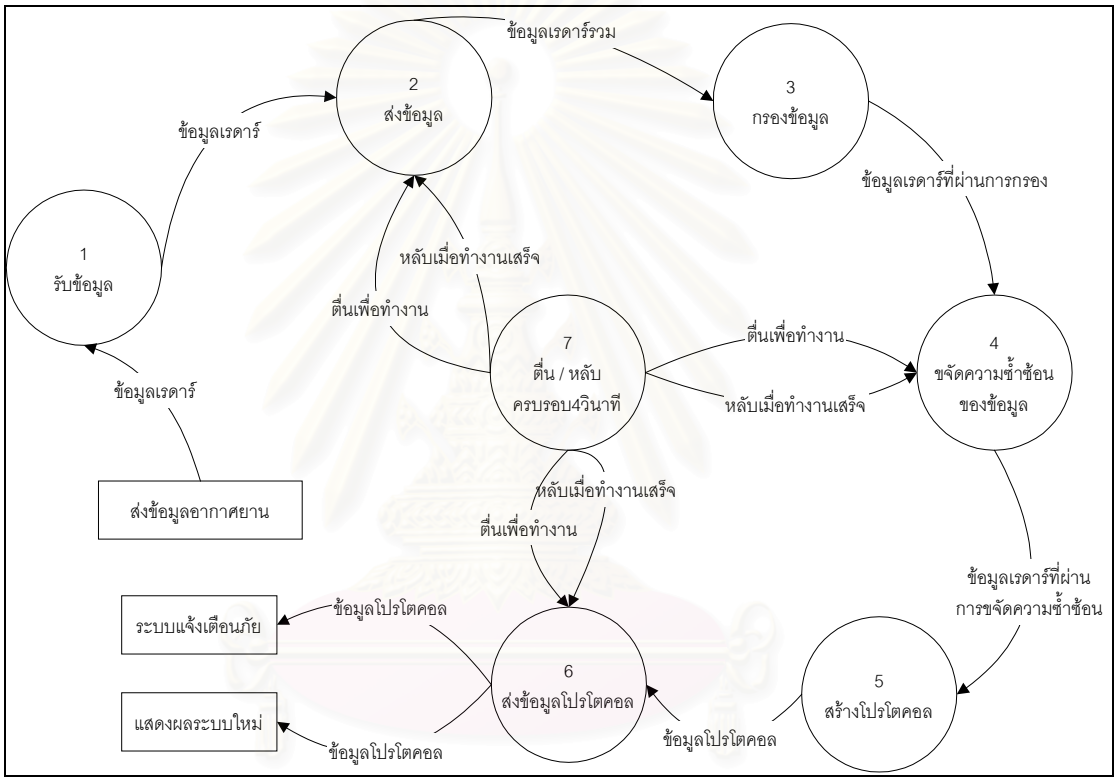
3.1 การออกแบบระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยาน

การทำงานของระบบจะเริ่มจากการที่มีเรดาร์ทั้งหมด 4 เครื่องต้องส่งผลของการตรวจจับอากาศยานมายังหน่วยกลางเพื่อทำการประมวลผล ในส่วนของส่วนกลางนั้นต้องมีอุปกรณ์พิเศษคือช่องอนุกรมหลายช่อง (Multiserial Port) เพื่อทำการรับข้อมูลจากเรดาร์ทั้ง 4 เพื่อนำมาประมวลผลในเครื่องๆเดียว เนื่องจากผู้ทำการวิจัยไม่สามารถจัดหาอุปกรณ์นี้ได้ การแก้ปัญหาในขั้นตอนนี้คืออาศัยการออกแบบโดยใช้หลักการของมัลติโปรแกรมมิ่ง (Multiprogramming) เพื่อการจำลองการทำงานของช่องอนุกรมหลายช่อง (Multiserial Port) เมื่อมีการรับข้อมูลจากหลายช่องทาง ช่องรับส่งสัญญาณ (Socket Port) สร้างขึ้นเพื่อทำหน้าที่ของช่องอนุกรม เพื่อให้สามารถทั้งรับและส่งสัญญาณของข้อมูลเรดาร์ได้หลายเครื่องสามารถใช้หลักการของการสร้างเทร็ด (Thread) ในมัลติโปรแกรมมิ่งได้ โดยเริ่มต้นที่การเปิดช่องส่งสัญญาณหมายเลข 3001 ถึง 3004 (หมายเหตุ:หมายเลขของช่องรับและส่งสัญญาณทั้งหมดในการทำวิจัยครั้งนี้นั้น ผู้ทำการวิจัยได้กำหนดขึ้นโดยเลือกช่องสัญญาณที่ไม่มีการใช้งาน) เพื่อทำการรับและส่งสัญญาณของเรดาร์ โดยหมายเลขของช่องส่งสัญญาณจะเพิ่มทีละ 1 เพิ่มตามจำนวนของเรดาร์ในการรับสัญญาณนั้นจะรอสัญญาณอยู่4วินาทีเพื่อให้สัญญาณมาให้ครบ และเปิดช่องรับสัญญาณหมายเลข 2000 เพื่อทำการรับข้อมูลของเรดาร์ทุกๆเครื่องหลังจากรอครบทุก 4 วินาทีแล้ว ระบบจะนำข้อมูลจากช่องรับสัญญาณหมายเลข 2000 ไปกรองและขจัดความซ้ำซ้อนก่อนที่จะส่งต่อไปยังระบบแสดงผลต่อไป โครงสร้างของโปรแกรมสามารถแสดงได้ในรูปที่ 3.3



รูปที่ 3.3 กระบวนการการส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรองข้อมูล

ในรูปที่ 3.3 กระบวนการการส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรองข้อมูล แสดงการทำงานของการขจัดความซ้ำซ้อนแบ่งออกเป็นกระบวนการ (Processes) โดยเริ่มจากสร้างกระบวนการเพื่อส่งข้อมูลและรับข้อมูลเท่ากับจำนวนของเรดาร์ กระบวนการจะทำการรับและส่งข้อมูลทุกๆ 4 วินาที เมื่อกระบวนการรับข้อมูลแล้วจะส่งข้อมูลเพื่อทำการรวมจะรอเพื่อทำการรับข้อมูลจนครบ 4 วินาที แล้วจึงส่งข้อมูลที่ได้รับการรวมข้อมูลไปยังกระบวนการเพื่อขจัดความซ้ำซ้อนและแสดงผลต่อไป



รูปที่ 3.4 การส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรอง

จากรูปที่ 3.4 การส่งข้อมูลที่ผ่านขจัดความซ้ำซ้อนและการกรองสามารถแบ่งการทำงานออกเป็นส่วนย่อยๆ 7 ส่วนด้วยกันคือ

3.1.1 **รับข้อมูล**

เพื่อรับข้อมูลเส้นทางการบินและตำแหน่งที่ตั้งของเรดาร์จากระบบส่งข้อมูลเรดาร์ ต้องสร้างเทร็ดขึ้นเท่ากับจำนวนของเรดาร์เท่ากับจำนวนของเรดาร์ เพื่อเปิดช่องรับสัญญาณข้อมูล

จะเริ่มจากหมายเลขที่ 3001 3002 3003 และ 3004 เมื่อรับข้อมูลแล้วจะส่งข้อมูลไปยังกระบวนการส่งข้อมูล

3.1.2 **ส่งข้อมูล**

เพื่อส่งข้อมูลเส้นทางการบินและตำแหน่งที่ตั้งของเรดาร์จากระบบส่งข้อมูลเรดาร์ ต้องสร้างเทร็ดขึ้นเท่ากับจำนวนของเรดาร์ เพื่อข้อมูลจากช่องรับสัญญาณข้อมูลจะเริ่มจากหมายเลขที่ 3001 3002 3003 และ 3004 และเก็บข้อมูลไว้รอจนครบ 4 วินาที่แล้วแต่ละเทร็ดจะส่งข้อมูลไปยังช่องรับสัญญาณที่ 2000 เพื่อทำการรวมข้อมูลและนำไปใช้ต่อไป เมื่อทำงานเสร็จแล้วเทร็ดจะหลับโดยใช้เวลาเท่ากับ 4 วินาทีลบด้วยเวลาในการทำงาน

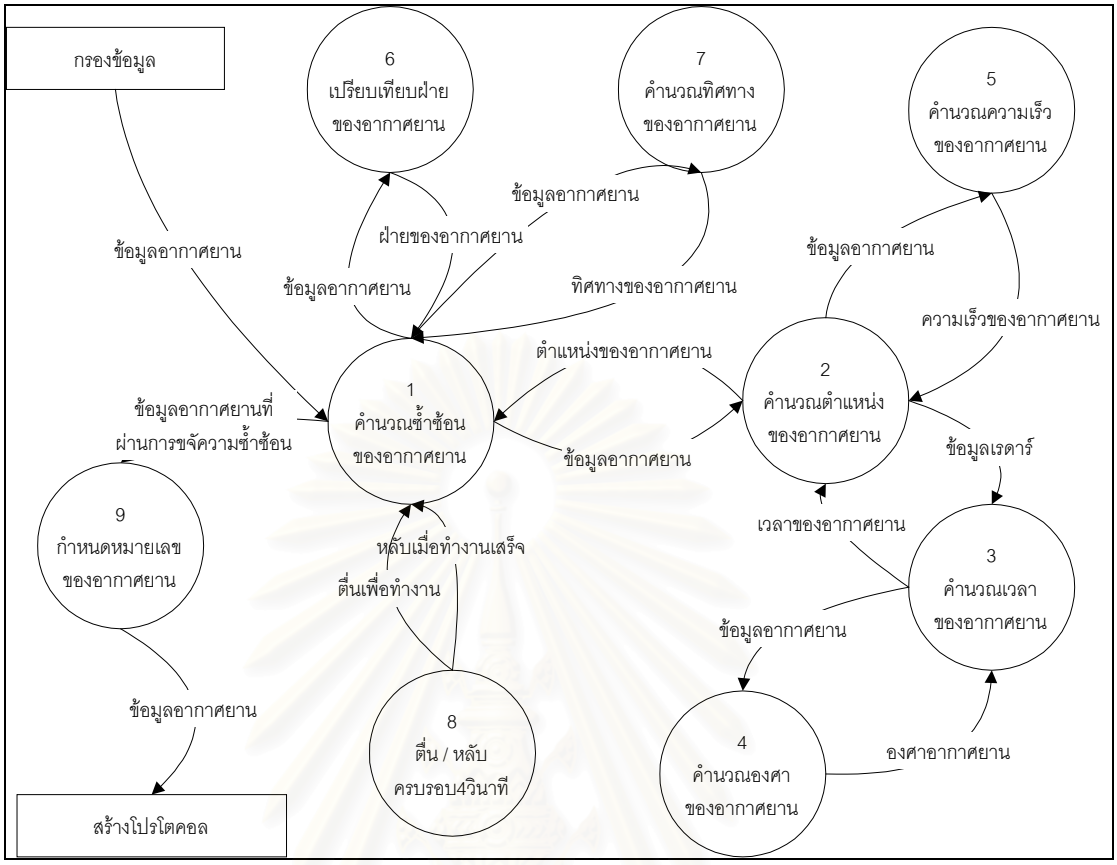
3.1.3 **การกรองข้อมูล**

ข้อมูลที่ไม่จำเป็นในการแจ้งเตือนภัยจะถูกตัดออกเช่น ความยาวของสัญญาณ ตัวเลขของเส้นทางการบินและหมายเลข

3.1.4 **ขจัดความซ้ำซ้อนของข้อมูล**

ในการทำการกรองและขจัดความซ้ำซ้อนของอากาศยาน ถ้าข้อมูลเก่าและข้อมูลใหม่รวมกันผลที่ได้ออกมาจะผิดพลาด ฉนั้นจึงต้องมีการป้องกันข้อมูลเก่าและข้อมูลใหม่ไม่ให้รวมกันโดยการใช้หลักผู้บริโภคและผู้ผลิต (Consumer-Producer) หลักการทำงานของผู้บริโภคและผู้ผลิตคือ ขณะที่มีการใช้ข้อมูลต้องไม่มีการสร้างข้อมูลได้จนกว่าการใช้ข้อมูลได้ทำเสร็จสิ้นแล้วจึงจะสามารถสร้างข้อมูลได้ และเมื่อมีการสร้างข้อมูลต้องรอให้สร้างข้อมูลจะเสร็จเรียบร้อยก่อนจึงสามารถนำข้อมูลไปใช้ได้

เมื่อรับข้อมูลเส้นทางการบินและตำแหน่งที่ตั้งของเรดาร์จากช่องรับสัญญาณมูลหมายเลข 3001 ถึงหมายเลข 3004 ก่อนที่ข้อมูลจะถูกส่งไปเพื่อบันทึกเก็บไว้ในหน่วยความจำนั้นต้องทำการทดสอบว่าในขณะนั้นมีการอ่านข้อมูลหรือไม่ ถ้ามีการอ่านข้อมูลต้องรอจนกว่าการอ่านข้อมูลจะเสร็จสิ้นเสียก่อนจึงบันทึกเก็บไว้ในหน่วยความจำได้ ในทางตรงข้ามกันการอ่านนั้นต้องทำการทดสอบว่าในขณะนั้นมีบันทึกเก็บไว้ในหน่วยความจำหรือไม่ ถ้ามีการบันทึกเก็บไว้ในหน่วยความจำต้องรอจนกว่าบันทึกเก็บไว้ในหน่วยความจำจะเสร็จเรียบร้อยก่อน จึงสามารถอ่านข้อมูลได้ ในทุกๆ 4 วินาทีข้อมูลจะถูกส่งผ่านไปยังช่องรับสัญญาณ 2000 เพื่อทำการกรองและขจัดความซ้ำซ้อน เมื่อทำงานเสร็จแล้วเทร็ดจะหลับโดยใช้เวลาเท่ากับ 4 วินาทีลบด้วยเวลาในการทำงาน สามารถแสดงได้ดังรูปที่ 3.5

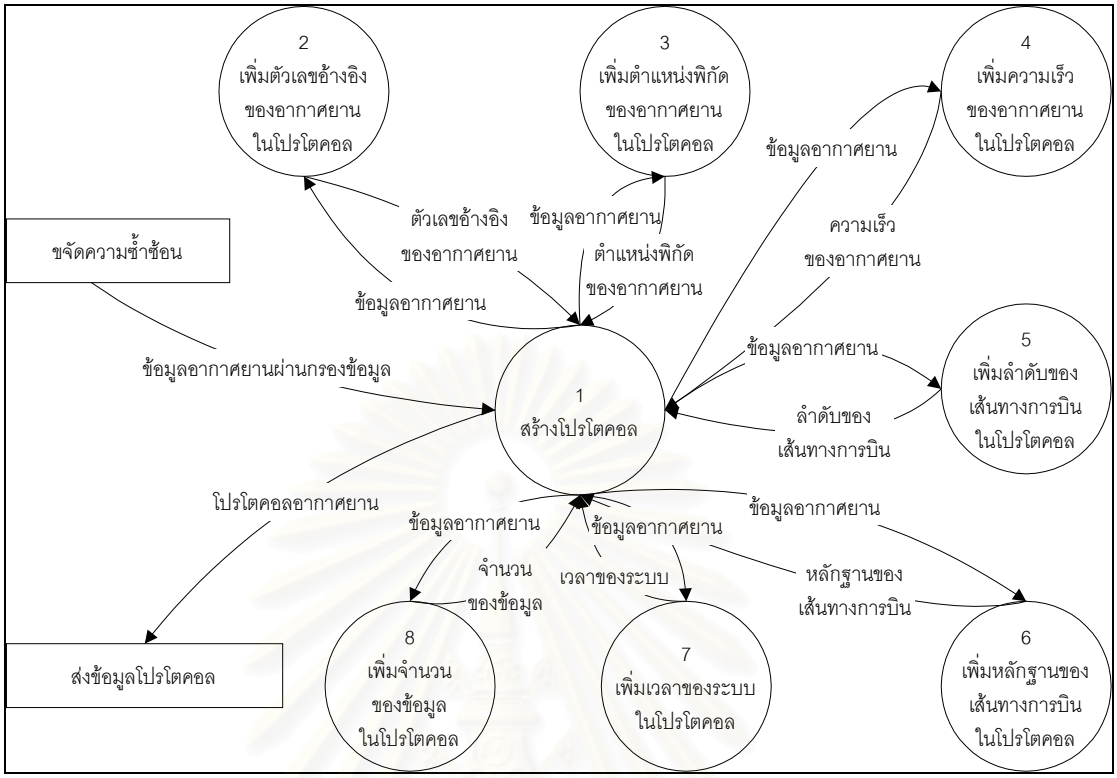


รูปที่ 3.5 การขจัดความซ้ำซ้อนของอากาศยาน

แนวทางในการขจัดความซ้ำซ้อนของข้อมูลนั้นจะเปรียบเทียบข้อมูล 4 ชนิดของอากาศยาน คือ ทิศทางของอากาศยาน ความเร็วของอากาศยาน ฝ่ายของอากาศยาน และตำแหน่งของอากาศยาน ถ้าอากาศยานคู่ใดๆมีคุณสมบัติครบทั้ง 4 ประการโปรแกรมก็จะสรุปว่าอากาศยานนั้นเป็นอากาศยานเดียวกัน โปรแกรมจะกำหนดหมายเลขของอากาศยานให้เป็นหมายเลขเดียวกัน ถ้าเป็นอากาศยานเครื่องใหม่ก็จะกำหนดหมายเลขอากาศยานใหม่ให้อากาศยาน

3.1.5 **การสร้างโปรโตคอล**

เมื่อข้อมูลครบตามที่ต้องการแล้ว ต้องสร้างข้อมูลตามรูปแบบของข้อมูลโปรโตคอลการแสดงภาพสถานการณ์ทางอากาศระยะไกล โดยข้อมูลจะประกอบด้วย 10 ข้อมูลโดยรายละเอียดของข้อมูลสามารถดูได้จากตารางที่ ก-8 การออกแบบโปรแกรมสามารถแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 การสร้างโปรโตคอลของอากาศยาน

3.1.6 การส่งข้อมูลโปรโตคอล

เมื่อข้อมูลครบตามที่ต้องการแล้วต้องสร้างข้อมูลตามรูปแบบของข้อมูลโปรโตคอลการแสดงภาพสถานการทางอากาศระยะไกลเพื่อส่งไปแสดงผลทางจอภาพต่อไปโดยผ่านทางช่องส่งสัญญาณหมายเลข 4000 และส่งข้อมูลผ่านทางช่องส่งสัญญาณหมายเลข 400 เพื่อเป็นข้อมูลของระบบแจ้งเตือนภัย โดยลักษระของโปรโตคอลแสดงในตารางที่ ก-9 และ ตารางที่ ก-10

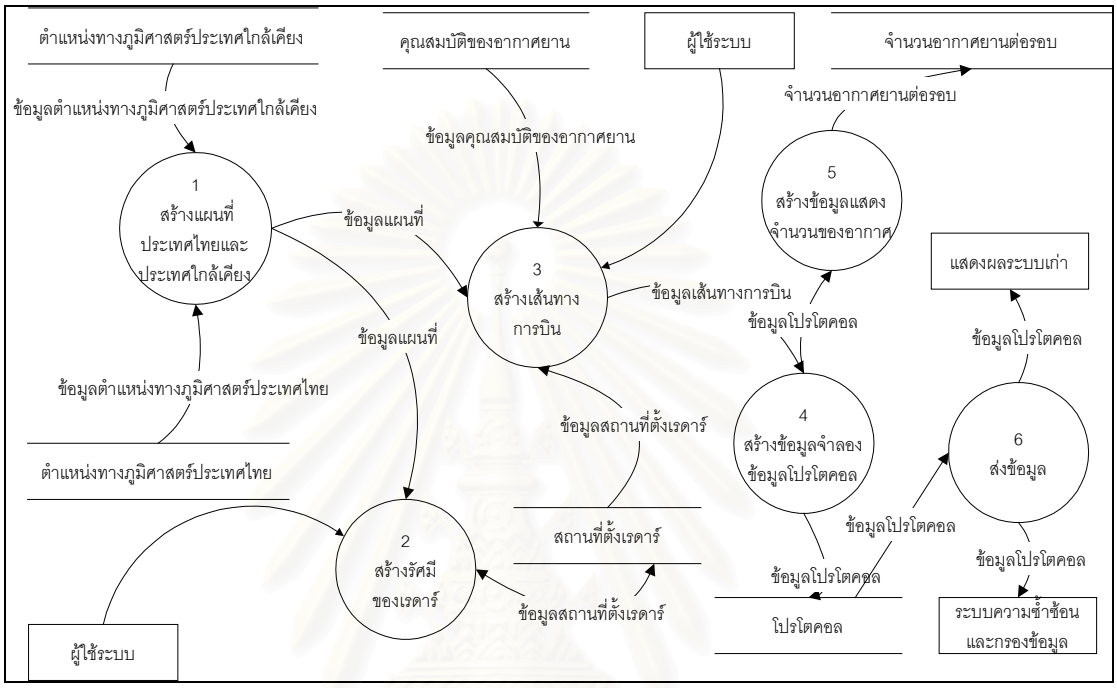
3.1.7 ตื่น / หลับครบรอบ4วินาที

เมื่อระบบทำงานเสร็จเวลาที่แตกต่างระหว่างเวลาที่ใช้ในการทำงานจริงและเวลา4วินาทีจะเวลาที่เทร็ดหลับ และเมื่อครบรอบของ4วินาทีเทร็ดต้องตื่นมาเพื่อทำงานต่อไป

3.2 การออกแบบระบบสร้างข้อมูลเรดาร์และการส่งข้อมูล

การออกแบบระบบสร้างข้อมูลเรดาร์และการส่งข้อมูลมีความสำคัญมากต่อการทำวิจัยครั้งนี้ เพราะในการทำการวิจัยครั้งนี้นั้นผู้ทำการวิจัยไม่สามารถนำข้อมูลของเรดาร์มาแสดงได้ จึง

ต้องทำการสร้างข้อมูลจำลองขึ้นมาเพื่อใช้ในการทำการวิจัยโดยอาศัยโปรโตคอลของเรดาร์เป็นต้นแบบในการจำลองข้อมูล แนวคิดในการออกแบบการสร้างข้อมูลการบินเพื่อใช้เป็นข้อมูลในการขจัดความซ้ำซ้อนและการกรองสามารถแบ่งออกเป็น 6 ส่วนสามารถแสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 การสร้างโปรโตคอลของเรดาร์

การออกแบบระบบสร้างข้อมูลเรดาร์และการส่งข้อมูลมีประกอบหลัก6ส่วนด้วยกันคือ

1. สร้างแผนที่ประเทศไทยและประเทศใกล้เคียง
2. สร้างการรัศมีของเรดาร์
3. สร้างเส้นทางการบิน
4. สร้างข้อมูลจำลองข้อมูลโปรโตคอล
5. สร้างข้อมูลแสดงจำนวนของอากาศ
6. ส่งข้อมูล

โดยรายละเอียดสามารถแสดงได้ดังต่อไปนี้

3.2.1 สร้างแผนที่ประเทศไทยและประเทศใกล้เคียง

การสร้างแผนที่โปรแกรมจะต้องอ่านแสดงข้อมูลจากแฟ้มข้อความ (Text file) เพื่อใช้เป็นข้อมูลในการสร้างแผนที่ประเทศไทย และประเทศใกล้เคียง(ข้อมูลและความหมายเป็นข้อมูลที่ใช้ในกองอำนวยการต่อสู้ป้องกันภัยทางอากาศ กองบัญชาการทหารสูงสุด) โดยแยกตามเส้นรุ้งและเส้นแวงตามตำแหน่งทางภูมิศาสตร์ เพื่อใช้ในการอ้างสถานที่ตั้งของเรดาร์ตามตำแหน่งทางภูมิศาสตร์ ตัวอย่างของข้อมูลจะเป็นดังในตัวดังนี้ UGMF4448 กำหนดเส้นทางการบินของอากาศยาน และสร้างค่าข้อมูลการบิน โดยรายละเอียดสามารถแบ่งออกได้ดังต่อไปนี้ ข้อมูลแผนที่ประเทศไทยจะมีความยาว8ตัวอักษรต่อโดยอักษรตำแหน่งที่1 และที่3 แทนองศาของเส้นแวง อักษรตำแหน่งที่5 และที่ 6 แทนลิปดาของเส้นแวง.อักษรตำแหน่งที่ 2 และที่ 4 แทนองศาของเส้นรุ้ง อักษรตำแหน่งที่7 และที่ 8 แทนลิปดาของเส้นรุ้ง ที่โดยสามารถแสดงในตารางที่ ก-1

ข้อมูลแผนที่ประเทศใกล้เคียงจะมีความยาว14 ตัวอักษรต่อตัวอย่างของข้อมูลจะเป็นดังในตัวดังนี้ 100911 0984042 โดยอักษรตำแหน่งที่1 ถึงตำแหน่งที่ 6 แทนตำแหน่งทางภูมิศาสตร์ของเส้นแวง อักษรตำแหน่งที่8 ถึงตำแหน่งที่ 14 แทนตำแหน่งทางภูมิศาสตร์ของเส้นรุ้ง ตำแหน่งที่ 7 จะเป็นตัวขอบเขตแบ่งระหว่างเส้นแวง โดยสามารถแสดงในตารางที่ ก-2

การออกแบบโปรแกรมจะเน้นที่การสร้างแผนที่ประเทศไทย โดยให้ประเทศไทยตั้งอยู่ตรงกลางของแผนที่ เริ่มจากการอ่านข้อมูลจากแฟ้มข้อความที่ตั้งทางภูมิศาสตร์ของประเทศไทยแล้วนำมาวาดเป็นแผนที่ประเทศไทย และประเทศใกล้เคียงโดยขอบเขตของแผนที่จะอยู่ที่เส้นรุ้งที่95องศาตะวันออกถึง 111 องศาตะวันออก และเส้นแวงที่ 5 องศาเหนือถึงเส้นแวงที่ 21 องศาเหนือพิกัด 0950000E0050000N1110000E0210000N เพื่อให้ง่ายต่อการใช้งานโปรแกรมแสดงผลการเคลื่อนที่ของเมาส์เป็นตำแหน่งทางภูมิศาสตร์ในระบบแผนที่ได้ เมื่อเมาส์มีการเคลื่อนที่จะส่งค่าการตำแหน่งของเมาส์ทางด้านแนวตั้งและแนวนอน เมื่อระบบได้รับตำแหน่งของเมาส์ระบบจะนำค่ามาเปรียบเทียบกับตำแหน่งทางภูมิศาสตร์ของแผนที่แล้วแปลงค่าออกมาเป็นค่าของเส้นรุ้งและเส้นแวง เมื่อระบบได้รับค่าการย่อหรือขยายของแผนที่ประเทศไทยและประเทศใกล้เคียง ระบบจะทำการเลือกพื้นที่ของแผนที่จะทำการย่อ หรือขยายแล้วนำค่าที่ได้รับมาคำนวณแผนที่ที่ได้รับการย่อขยายแล้ว จึงทำการวาดแผนที่ประเทศไทยและประเทศใกล้เคียง

3.2.2 **สร้างการรัศมีของเรดาร์**

โปรแกรมต้องสร้างเรดาร์ให้มีขนาดเท่ากับขนาดของรัศมีของเรดาร์คือ 75 ไมล์ทะเล โดยขนาดของเรดาร์ต้องเหมาะสมกับขนาดของแผนที่ โดยใช้ตำแหน่งของเมาส์ที่ได้รับการเลือกเป็นจุดศูนย์กลางของเรดาร์ ในการย้ายเรดาร์โปรแกรมต้องได้รับตำแหน่งของเมาส์เพื่อกำหนดเรดาร์ที่จะทำการย้ายและรับตำแหน่งของเมาส์เพื่อกำหนดตำแหน่งใหม่ของเรดาร์

การบันทึกข้อมูลของเรดาร์โปรแกรมจะต้องบันทึกจำนวนของเรดาร์ และตำแหน่งทางภูมิศาสตร์ของเรดาร์แต่ละเครื่องลงบนแฟ้มข้อความตามชื่อที่สามารถกำหนดได้ ข้อมูลตำแหน่งของเรดาร์ที่จะทำการบันทึกจะมีโครงสร้างดังตารางที่ ก-3

การเปิดแฟ้มข้อความของเรดาร์ เมื่อเปิดแฟ้มข้อความต้องอ่านข้อมูลเก็บไว้ในหน่วยความจำเพื่อนำมาวาดเป็นภาพรัศมีของเรดาร์ ตามอัตราส่วนการย่อขยายของระบบแผนที่ประเทศไทย และประเทศใกล้เคียง

เมื่อมีการย่อหรือขยายแผนโปรมแกรมต้องทำการคำนวณรัศมีของเรดาร์ที่มีขนาดย่อหรือขยายตามขนาดของแผนที่ในอัตราส่วนเดียว

3.2.3 **สร้างเส้นทางการบิน**

โปรแกรมสามารถสร้างรัศมีของเรดาร์โดยการเปิดข้อมูลจากแฟ้มข้อมูลเพื่อใช้ในการสร้างรัศมีของเรดาร์ การกำหนดเส้นทางการบินนั้นต้องสามารถเลือกชนิดของอากาศยาน ความเร็วของอากาศยาน โดยในการเลือกต้องมีภาพและความเร็วของอากาศยานแสดงให้เห็นรวมทั้งต้องสามารถกำหนดฝ่ายของอากาศยานได้ โปรแกรมจะอ่านข้อมูลภาพและความเร็วของอากาศยานเพื่อแสดงผลทางจอภาพ การสร้างเส้นทางการบินต้องแสดงให้เห็นเส้นทางการบินวาดลงบนแผนที่ประเทศไทย และประเทศใกล้เคียงเพื่อให้เห็นเป็นภาพกราฟิก เส้นทางการบินของอากาศยานที่วาดบนแผนที่ประเทศไทยและประเทศใกล้เคียง ต้องสามารถการแก้ไข หรือลบเส้นทางการบินเส้นทางการบินได้ เมื่อมีแก้ไขเส้นทางการบินโปรแกรมต้องวาดเส้นทางการบินตามที่ได้รับการแก้ไขนั้น ข้อมูลของอากาศยานและเรดาร์สามารถบันทึกบนแฟ้มข้อความโดยมีโครงสร้างของข้อมูลดังตารางที่ ก-4 การบันทึกข้อมูลเส้นทางต้องสามารถกำหนดชื่อได้ การเปิดข้อมูลเส้นทางโปรแกรม จะนำข้อมูลมาสร้างเป็นเส้นทางการบินบนแผนที่ประเทศไทยและประเทศใกล้เคียงเพื่อให้เห็นเป็นภาพกราฟฟิก

3.2.4 **สร้างข้อมูลจำลองข้อมูลโปรโตคอล**

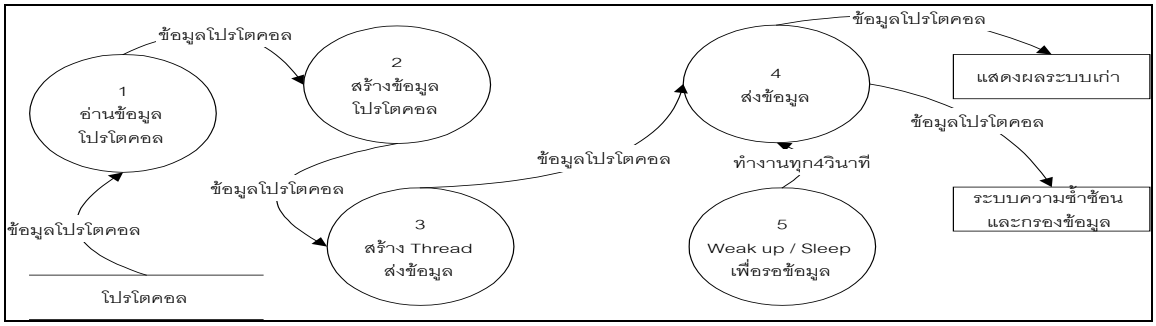
การสร้างข้อมูลจำลองข้อมูลของเรดาร์นั้นต้องทำการ คำนวณเวลาของอากาศยานจากข้อมูลของอากาศยานตำแหน่งของอากาศ โดยคำนวณได้จากความเร็วของอากาศยานต่อส่วนของวินาที การคำนวณมุมของอากาศยานโดยในการหมุนหนึ่งรอบของเรดาร์ อากาศ โดยคำนวณได้จากมุมของเรดาร์ต่อเวลา 1 ใน 200 ส่วนของวินาที คำนวณมุมของอากาศยานกับเรดาร์เพื่อนำมาเปรียบเทียบกับมุมของเรดาร์ ถ้ามุมเท่ากันแสดงว่าเรดาร์ตรวจจับอากาศยานได้ที่มุมและเวลาที่คำนวณได้ โดยคำนวณได้จากตำแหน่งเส้นแวงและตำแหน่งเส้นรุ้ง โดยข้อมูลของเรดาร์และมุมจะเก็บเป็นแฟ้มข้อความโดยมีโครงสร้างดังตารางที่ ก-5 และชนิดความเร็ว และหมายเลขของอากาศยานจะถูกแยกออกมาเก็บเป็นแฟ้มข้อความโดยมีโครงสร้างดังตารางที่ ก-6 เพื่อที่จะสร้างข้อมูลจากการสร้างเส้นทางการบินจะต้องมีการสำรองข้อมูลในไมโครซอฟท์แอคเซ็สส์ เพื่อใช้ความสามารถในการเรียงลำดับของข้อมูลโครงสร้างของข้อมูลสามารถแสดงในตารางที่ ก-7 เมื่อทำการกำหนดหมายเลขอากาศยานเรียบร้อยแล้วต้องสร้างแฟ้มข้อมูลเพื่อนำไปใช้ในการขจัดความซ้ำซ้อน โครงสร้างของข้อมูลสามารถแสดงในตารางที่ ก-8

3.2.5 **สร้างข้อมูลแสดงจำนวนของอากาศ**

เพื่อตรวจจำนวนของอากาศยานจากระบบที่ทำการสร้างข้อมูลกับข้อมูลของอากาศยานที่ได้รับเพื่อการแสดงออกทางจอแสดงภาพ การแสดงจะแสดงจำนวนของอากาศยานของแต่ละเรดาร์และแสดงความถูกต้องของอากาศยานทั้งหมดเป็นเปอร์เซ็นต์ โดยข้อมูลที่ได้จากการคำนวณทั้งหมดต้องแสดงบนจอภาพ ข้อมูลจำนวนอากาศยานทั้งหมดมีโครงสร้างดังตารางที่ ก-11 และความถูกต้องระบบดังตารางที่ ก-12

3.2.6 **ส่งข้อมูล**

การส่งข้อมูลต้องทำการจำลองการส่งข้อมูลของเรดาร์ การออกแบบสร้างมัลติโปรแกรมมิ่ง (Multiprogramming) โดยการสร้างเทร็ด (Thread) เพื่อการจำลองการส่งสัญญาณการส่งสัญญาณของเรดาร์และเรดาร์จะทำการส่งสัญญาณทุกๆ4วินาที โดยเริ่มต้นจากการอ่านข้อมูลโปรโตคอลแล้วสร้างโปรโตคอลของเรดาร์ สร้างช่องส่งสัญญาณเท่ากับจำนวนของเรดาร์แล้วทำการส่งสัญญาณ เมื่อได้ส่งสัญญาณโปรโตคอลของเรดาร์เรียบร้อยแล้วหลับ (Sleep) เพื่อให้ครบ4วินาที เมื่อครบ4วินาทีโปรแกรมย่อยจะตื่นเพื่อทำการส่งสัญญาณรอบต่อไป ดังสามารถแสดงได้ในรูปที่ 3.8

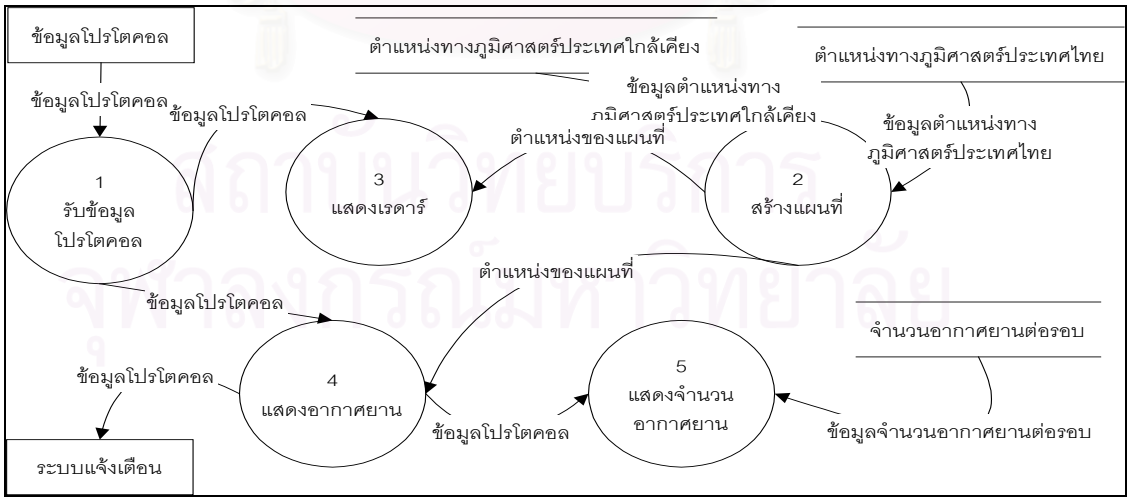


รูปที่ 3. 8 การส่งข้อมูล

3.3 การออกแบบระบบแสดงผลทางจอภาพ

การแสดงผลจะต้องแสดงสถานที่ตั้งของเรดาร์และแสดงการของเคลื่อนที่ของอากาศยาน โดยการแสดงอากาศยานจะต้องมีความแตกต่างระหว่างฝ่ายเราและฝ่ายข้าศึก โดยการใช้สีแดง และ H ตามด้วยหมายเลขสามหลัก (001-999) แทนฝ่ายข้าศึกและใช้สีเขียวและ F ตามด้วยหมายเลขสามหลัก (001-999) แทนฝ่ายเรา เมื่อได้รับข้อมูลจากช่องรับสัญญาณ (Socket Port)

โปรแกรมต้องทำการคำนวณตำแหน่ง และฝ่ายของอากาศยานก่อนที่จะทำ การวาดต้องทำการวาดทุกๆ 4 วินาที เมื่อมีการย่อหรือขยายขนาดของแผนที่จะต้องมีการทำการปรับตำแหน่งของอากาศยานก่อนที่จะทำการวาด และต้องแสดงจำนวนของอากาศยานต่อการหมุนของเรดาร์ ดังแสดงในรูปที่ 3.9



รูปที่ 3. 9 แสดงผลทางจอภาพ

3.4 การออกแบบแฟ้มข้อมูลและตารางข้อมูล

จากแนวคิดการทำงานของระบบสามารถออกแบบแฟ้มข้อมูลและคำอธิบายได้ดังต่อไปนี้

1 แฟ้มข้อมูลThaibder.GEOเป็นแฟ้มข้อมูลเก็บตำแหน่งของอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศไทยโดยข้อมูลจะเก็บตำแหน่งที่ตั้งทางภูมิศาสตร์ของประเทศไทยอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) ตัวอย่างของข้อมูลเช่นUGNG0515 โดยสามารถมีสูตรแปลงมาเป็นตำแหน่งที่ตั้งทางภูมิศาสตร์ เป็นเส้นรุ้งและเส้นแวง มีหน่วยเป็นลิปดา โดยมีรายละเอียดดังตารางที่ ก-1

2 แฟ้มข้อมูลNeighbor.GEOเป็นแฟ้มข้อมูลเก็บอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศใกล้เคียงอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) ตัวอย่างของข้อมูลเช่น จะอยู่ในตัวอย่างต่อไปนี้ 095826 0983434 ตำแหน่งที่1-6 จะแทนเส้นรุ้งที่ 9 องศา 58 ลิปดา 26 พิลิปดา และรายละเอียดดังตำแหน่งที่8-14แทนเส้นแวงที่ 98 องศา 34 ลิปดา 34 พิลิปดา โดยมีรายละเอียดดังตารางที่ ก-2

3 แฟ้มข้อมูลของเรดาร์ชื่อของแฟ้มข้อมูลเรดาร์ตามที่กำหนดและตามด้วย.RAD เป็นแฟ้มข้อมูลเก็บจำนวนของเรดาร์และตำแหน่งทางภูมิศาสตร์ของเรดาร์ เก็บอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-3

4 แฟ้มข้อมูลของโปรโตคอลชื่อของแฟ้มข้อมูลโปรโตคอลตามที่กำหนดและตามด้วย.TCK เป็นแฟ้มข้อมูลเก็บโปรโตคอลของเรดาร์และตำแหน่งของเรดาร์ เก็บอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) โดยในส่วนแรกจะเก็บจำนวนและสถานที่ตั้งของเรดาร์และตามด้วยโปรโตคอลของเรดาร์รายละเอียดดังตารางที่ ก-4

5 แฟ้มข้อมูลของอากาศยานชื่อของแฟ้มข้อมูลอากาศยานจะเป็นTrackDataXตามที่กำหนดและตามด้วย.FLT เป็นแฟ้มข้อมูลเก็บโปรโตคอลของเรดาร์และตำแหน่งของเรดาร์ เก็บอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-5

6 แฟ้มข้อมูลของรายละเอียดของอากาศยานชื่อของแฟ้มข้อมูลรายละเอียดของอากาศยานจะเป็นTrackDataInfo.Inf เป็นแฟ้มข้อมูลเก็บโปรโตคอลของเรดาร์และตำแหน่งของเรดาร์ เก็บอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-6

7 ดาตาเบสTrack.MDBเก็บข้อมูลของอากาศยานที่ได้รับการจัดเรียงตามเวลาและเรดาร์รายละเอียดดังตารางที่ ก-7

8 แฟ้มข้อมูลของโปรโตคอลของอากาศยานชื่อของแฟ้มข้อมูลโปรโตคอลของอากาศยานจะเป็น ProtocolRadarX.PCT เป็นแฟ้มข้อมูลเก็บโปรโตคอลของเรดาร์และ

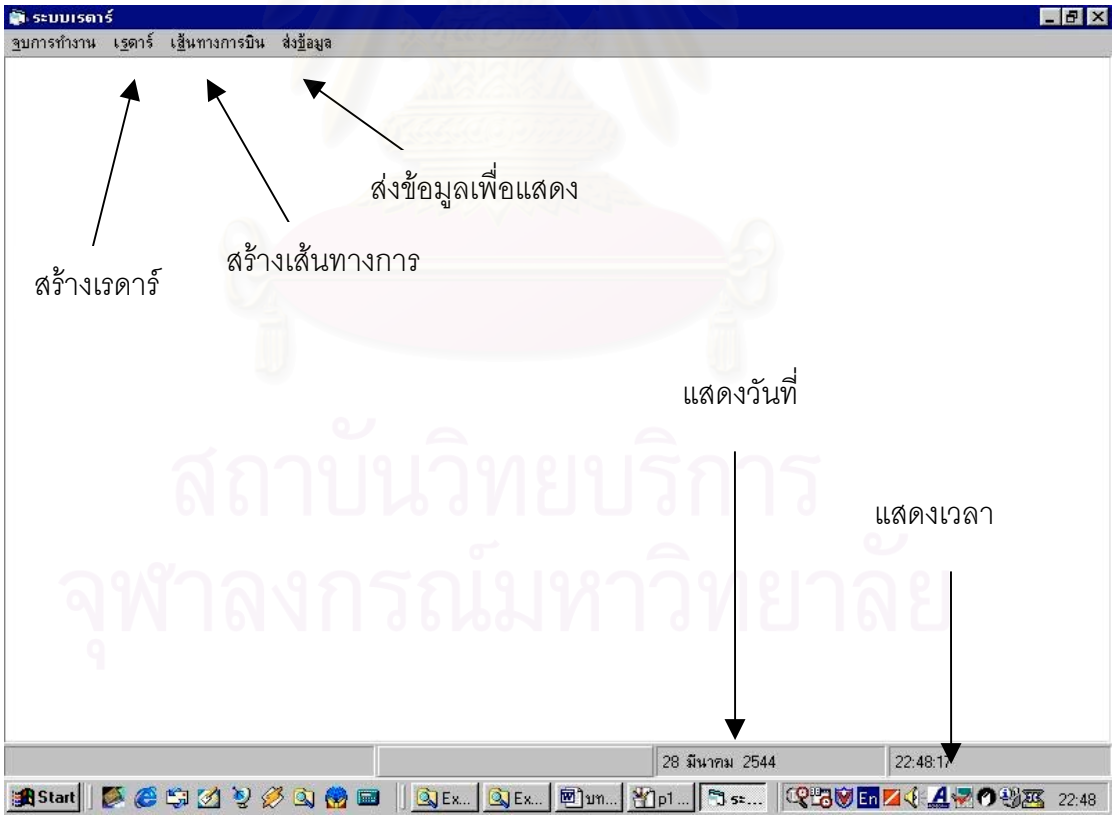
ตำแหน่งของเรดาร์ เก็บอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-8

9 แฟ้มข้อมูลTotalAC.TOTเป็นแฟ้มข้อสถิติความถูกต้องของจำนวนเรดาร์อยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-9

10 แฟ้มข้อมูลBothX.TXTเป็นแฟ้มข้อสถิติความถูกต้องของของการกรองและขจัดความซ้ำซ้อนอยู่ในรูปแบบของฟรีฟอร์แมท (Free Format) รายละเอียดดังตารางที่ ก-10

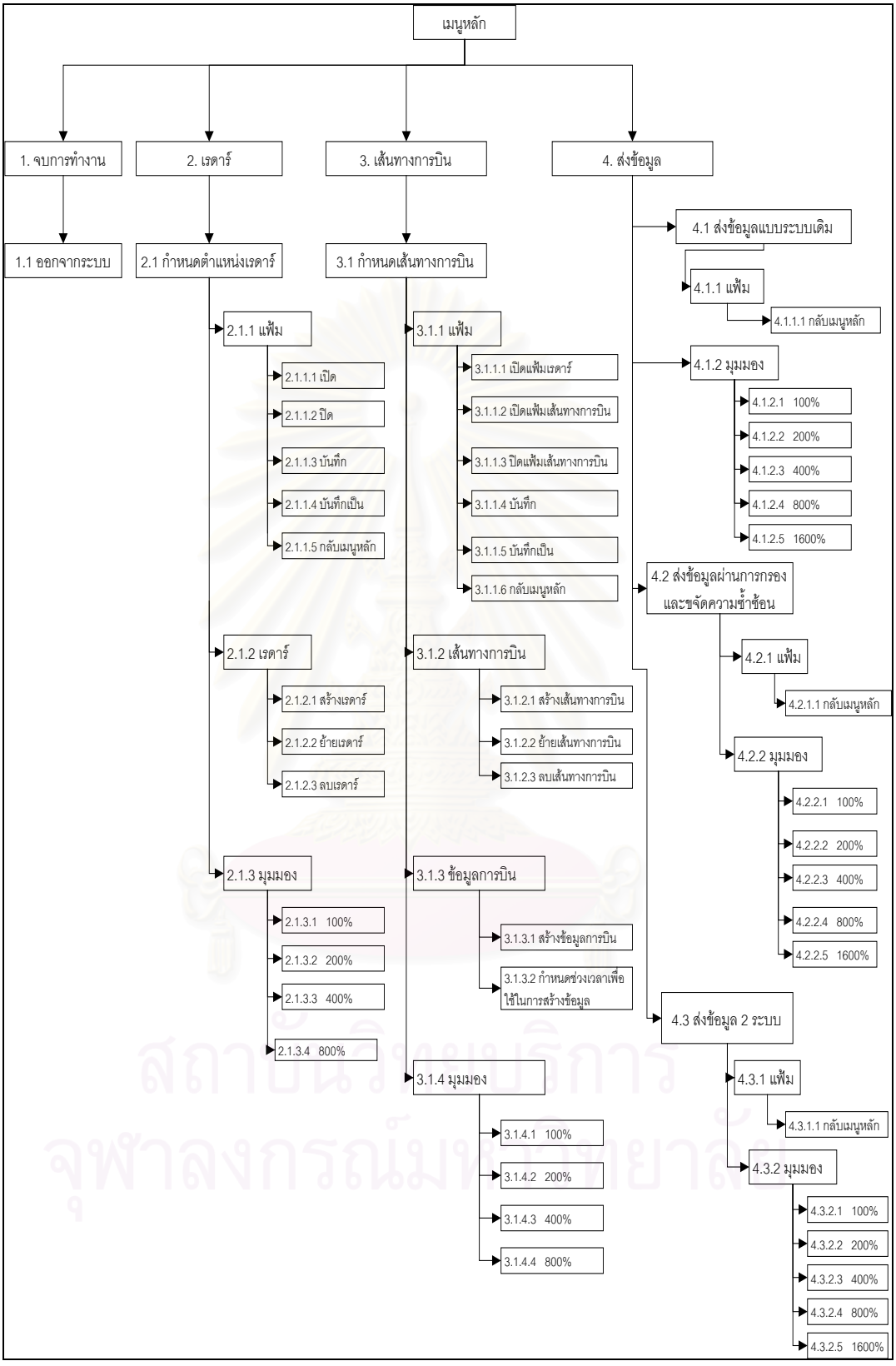
3.5 การออกแบบแสดงผลทางจอภาพ

การออกแบบหน้าจอของระบบการออกแบบการสร้างข้อมูลการบินและแสดงผลทางจอภาพ เพื่อใช้ในการสร้างข้อมูลเส้นทางการบิน และการแสดงผลของเรดาร์โดยการรับข้อมูลมาของอากาศยานเพื่อแสดงผลข้อมูลแบบการแสดงผลกราฟิก โดยการเริ่มต้นจะต้องเริ่มต้นจากโปรแกรมหลักก่อนเพื่อที่จะแสดงเมนูรวมของระบบ ดังแสดงในรูปที่ 3.10 ภาพรวมของการของการออกแบบแสดงผลทางจอภาพสามารถแสดงได้ในรูปที่ 3.11



รูปที่ 3.10 แสดงผลเมนูหลักทางจอภาพ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

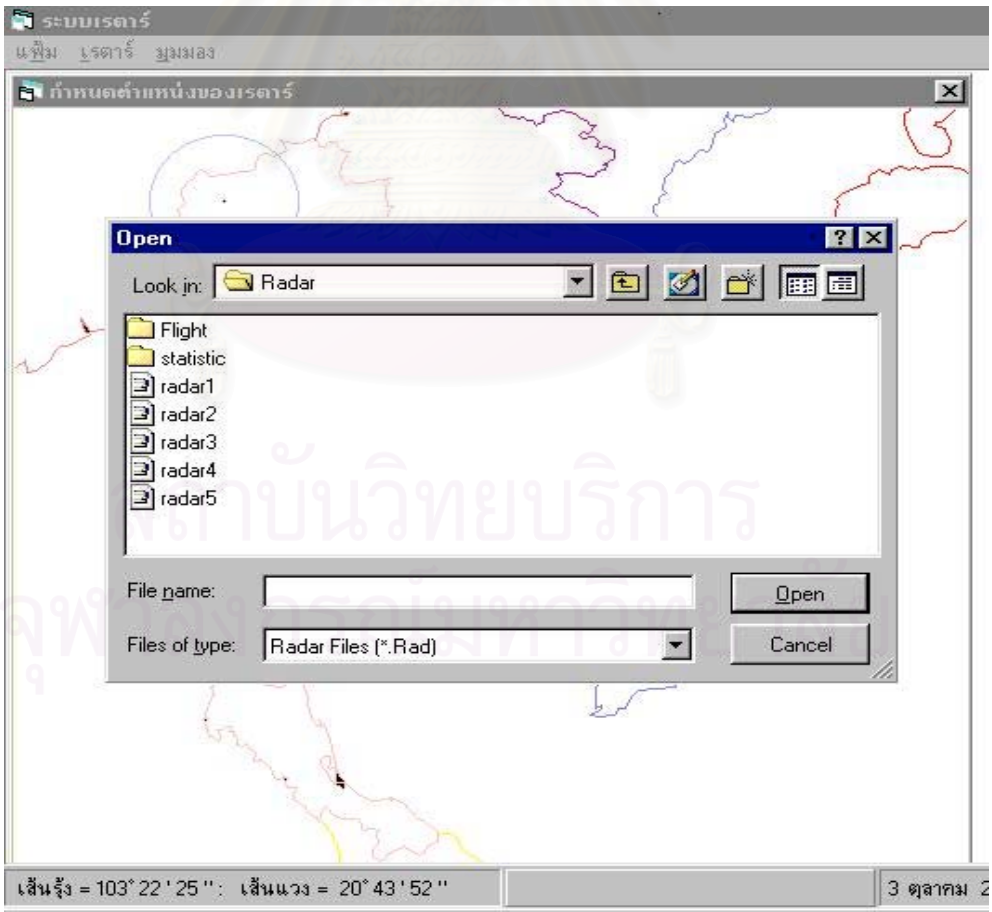


รูปที่ 3.11 โครงสร้างของการออกแบบแสดงผลทางจอภาพ

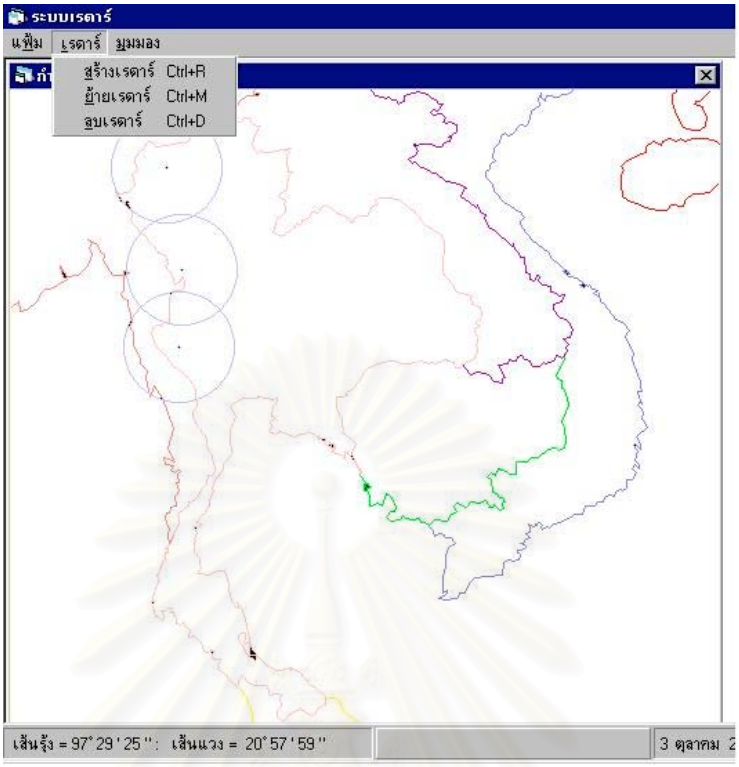
3.5.1 การออกแบบแสดงผลเรดาร์ทางจอภาพ

หน้าที่ของโปรแกรมนี้สร้างเรดาร์บนแผนที่ประเทศไทยและประเทศใกล้เคียง และเมื่อมีการเคลื่อนย้ายเมาส์ต้องแสดงตำแหน่งทางภูมิศาสตร์ของเมาส์ โดยจะสามารถแบ่งออกเป็น 3 ส่วนประกอบหลักคือ

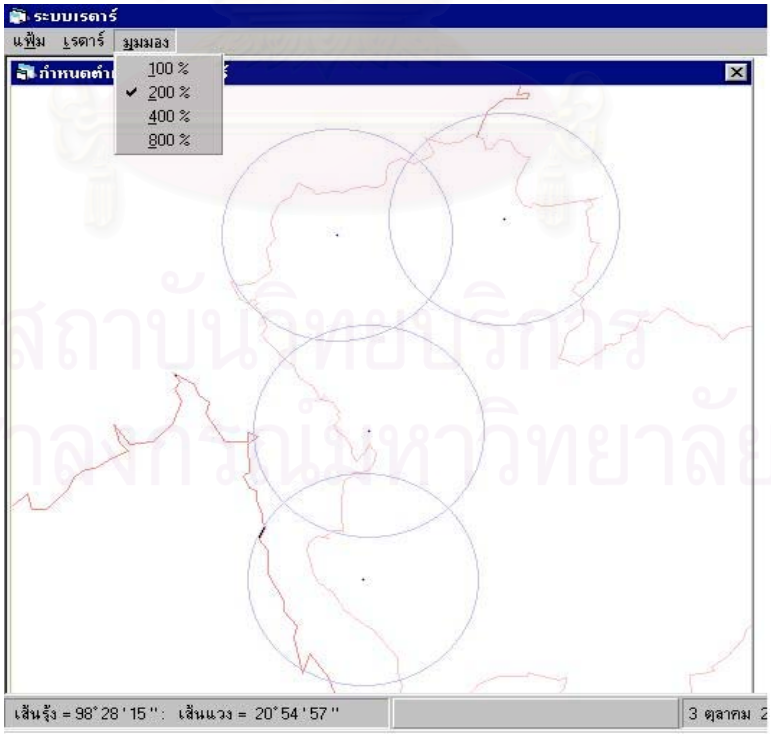
1. แฟ้มข้อมูลสามารถบันทึกข้อมูลของเรดาร์ลงบนแฟ้มข้อมูล อ่านข้อมูลจากจากแฟ้มข้อมูลเพื่อมาแสดงเป็นตำแหน่งที่ตั้งของเรดาร์ ในการบันทึกผู้ใช้สามารถกำหนดชื่อตามที่ต้องการได้ เมื่ออ่านข้อมูลและต้องแสดงตำแหน่งที่ตั้งของเรดาร์ ดังแสดงได้ในรูปที่ 3.12
2. เรดาร์สามารถสร้างเรดาร์ ลบเรดาร์และเคลื่อนย้ายโดยการใช้เมาส์แสดงได้ในรูปที่ 3.13
3. มุมสามารถย่อยหรือขยายแผนที่ประเทศไทยและเรดาร์ โดยเรดาร์จะย่อหรือขยายตามอัตราส่วนของประเทศไทยตามไปด้วยดังแสดงได้ในรูปที่ 3.14



รูปที่ 3.12 แสดงผลแฟ้มข้อมูลเรดาร์ทางจอภาพ



รูปที่ 3.13 แสดงผลการจัดการเรดาร์ทางจอภาพ

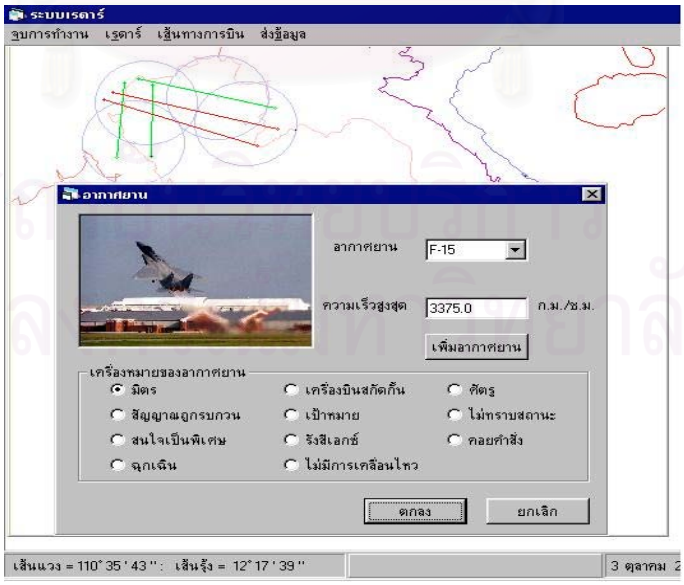


รูปที่ 3.14 แสดงผลย่อหรือขยายเรดาร์ทางจอภาพ

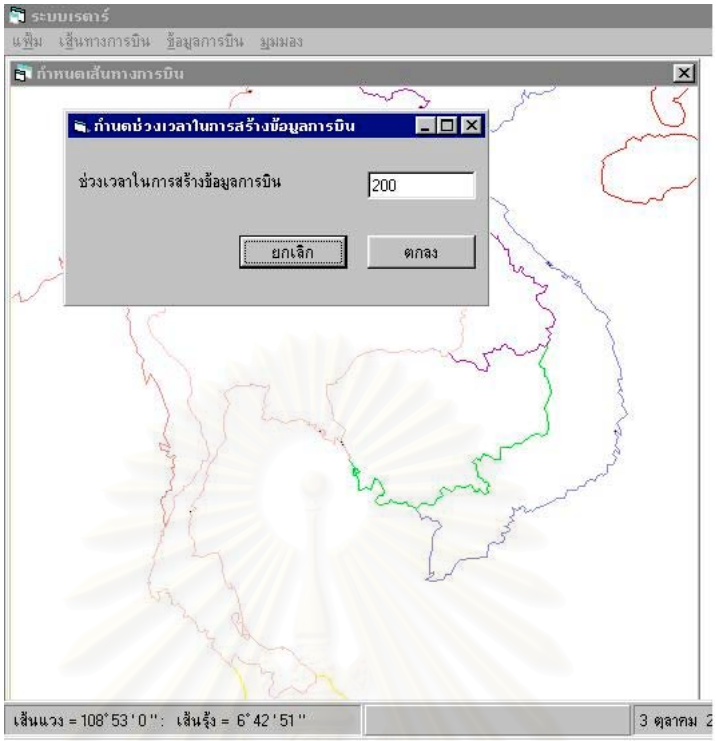
3.5.2 การออกแบบแสดงผลเส้นทางการบินทางจอภาพ

หน้าที่ของโปรแกรมนี้สร้างเส้นทางการบินบนแผนที่ประเทศไทย และประเทศใกล้เคียง โดยแสดงเส้นทางการบินแบบกราฟฟิกและสร้างข้อมูลการบินเพื่อนำไปใช้ต่อไปได้ โดยจะสามารถแบ่งออกเป็น 4 ส่วนประกอบหลักคือ

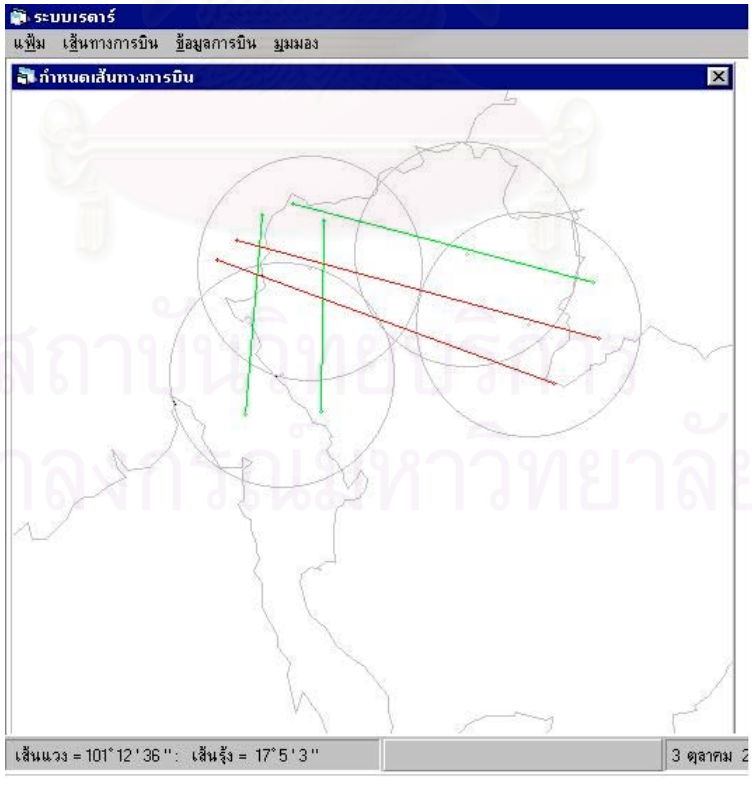
1. แฟ้มข้อสามารถบันทึกข้อมูลของเส้นทางการบินลงบนแฟ้มข้อมูล อ่านข้อมูลจาก จากแฟ้มข้อมูลเพื่อมาแสดงเป็นตำแหน่งที่ตั้งของเรดาร์และเส้นทางการบิน ในการบันทึกผู้ใช้สามารถกำหนดชื่อตามที่ต้องการได้ และแสดงเส้นทางการบินแบบกราฟฟิกเมื่อมีการอ่านจากแฟ้มข้อความ
2. เส้นทางการบินสามารถสร้างเส้นทางการบิน ลบเส้นทางการบิน และเคลื่อนย้ายเส้นทางการบิน การใช้งานสามารถสร้างเส้นทางการบิน ลบเส้นทางการบิน และเคลื่อนย้ายเส้นทางการบิน โดยการใช้เมาส์ดังแสดงได้ในรูปที่ 3.15
3. สร้างข้อมูลการบิน ทำหน้าที่สร้างเส้นทางการบินและในระหว่างที่ทำการสร้างเส้นทางการบินต้องแสดงสถานะของการสร้างเส้นทางการบินให้ทราบ ในการสร้างเส้นทางการบินสามารถเปลี่ยนส่วนของวินาทีในการสร้างข้อมูลการบินได้ โดยการใช้เมาส์แสดงได้ในรูปที่ 3.16
4. มุมมอง ทำหน้าที่ย่อหรือขยายแผนที่ประเทศไทยโดยเรดาร์และเส้นทางการบินจะย่อหรือขยายตามอัตราส่วนของประเทศไทยตามไปด้วยดังแสดงได้ในรูปที่ 3.17



รูปที่ 3.15 แสดงผลการสร้างเส้นทางการบินทางจอภาพ



รูปที่ 3.16 แสดงผลการกำหนดส่วนของวินาทีเพื่อสร้างข้อมูลการบินทางจอภาพ

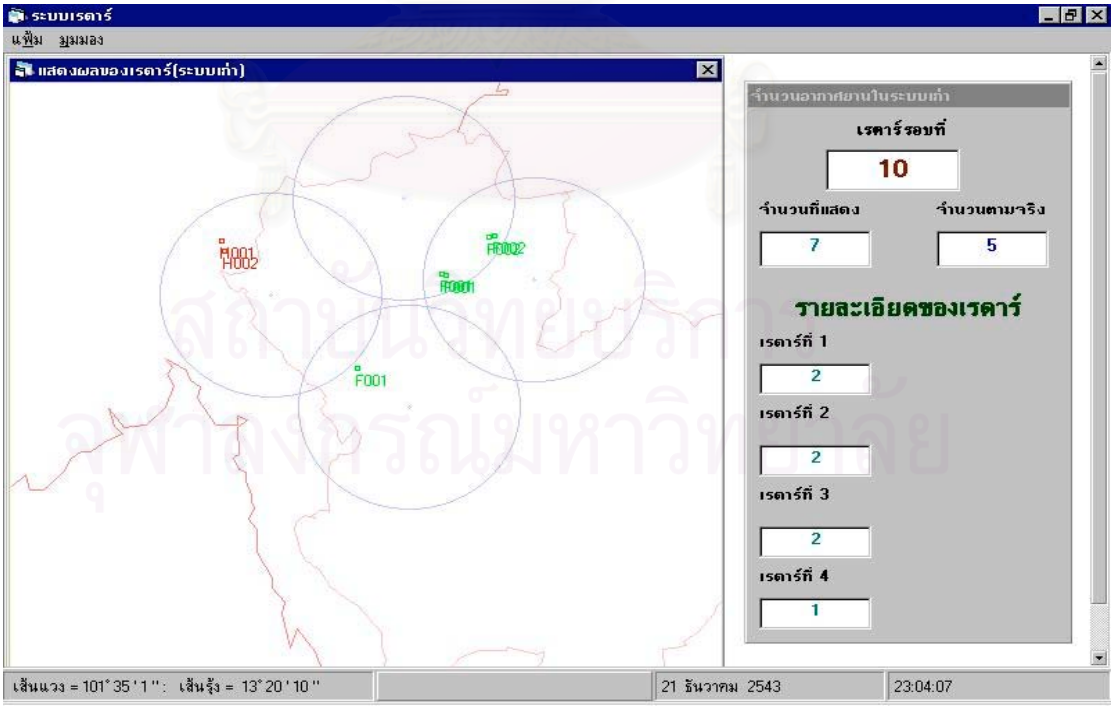


รูปที่ 3.17 แสดงผลย่อหรือขยายเส้นทางการบินทางจอภาพ

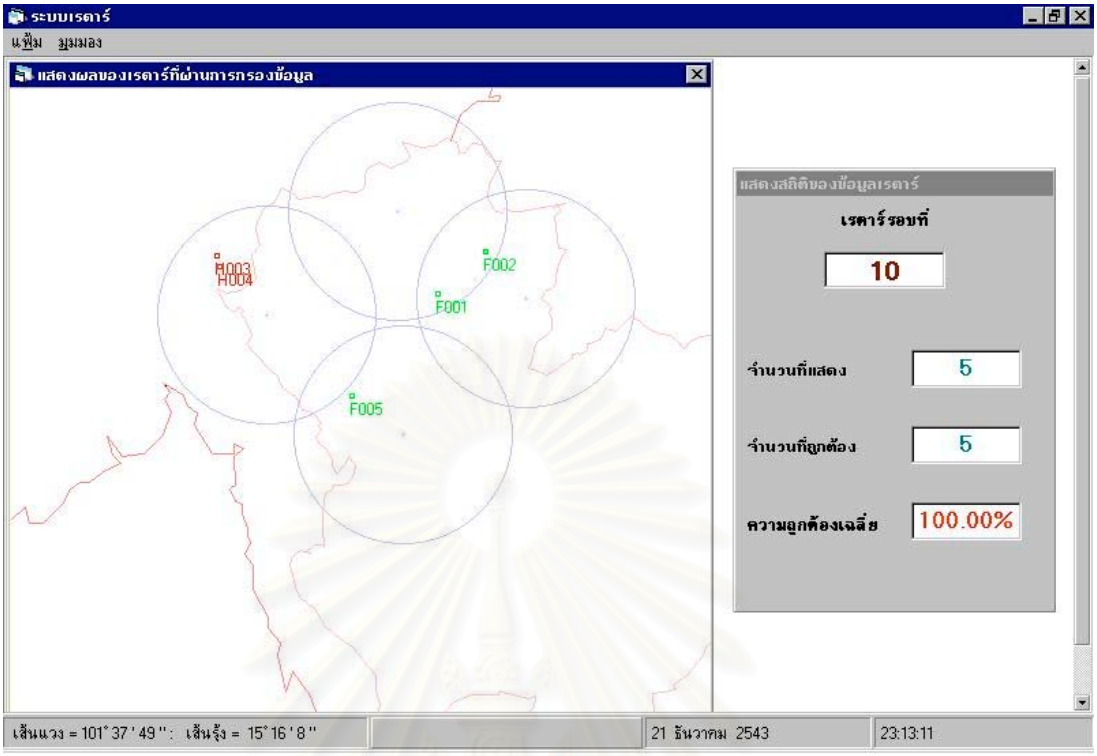
3.5.3 การออกแบบแสดงผลของข้อมูลเส้นทางการบินทางจอภาพ

หน้าที่ของโปรแกรมนี้แสดงผลเส้นทางการบินบนแผนที่ประเทศไทยและประเทศใกล้เคียง โดยจะสามารถแบ่งออกเป็น 3 ส่วนประกอบหลักคือ

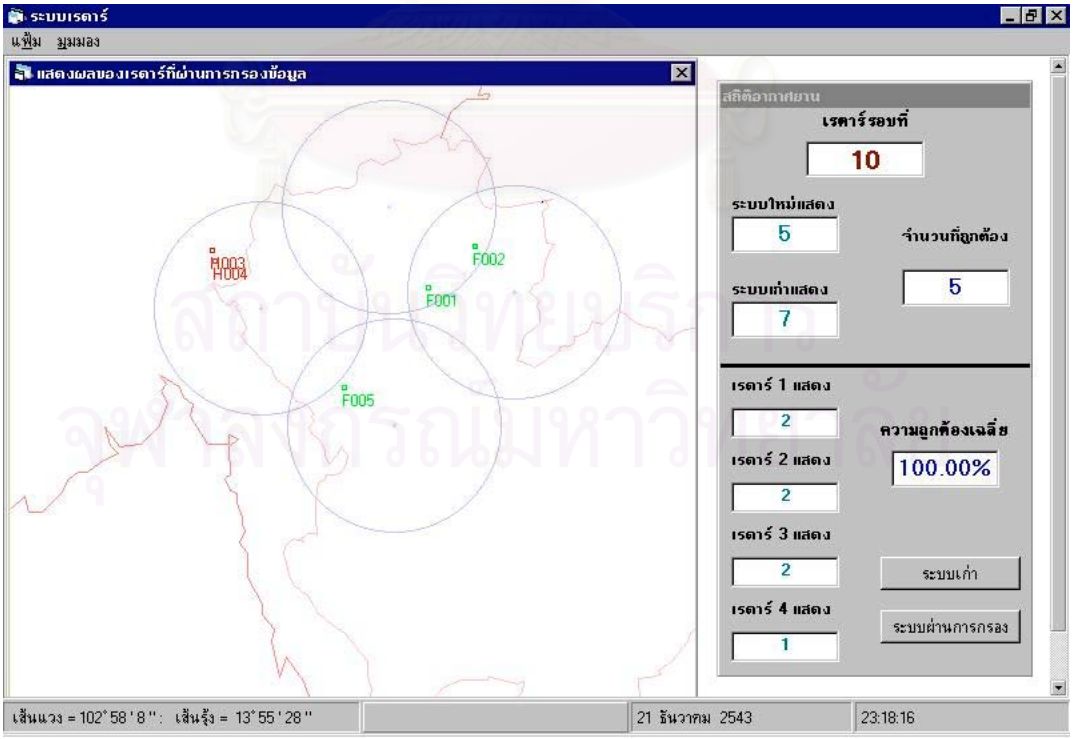
1. สามารถแสดงผลเส้นทางการบินบนแผนที่ประเทศไทยและประเทศใกล้เคียงในระบบเก่าโดยต้องแสดงการเคลื่อนไหวของอากาศยานทุกๆ 4 วินาทีและแสดงสถิติของอากาศยานได้ตามเรดาร์ และต้องสามารถย่อและขยายแผนที่ได้ดังแสดงได้ในรูปที่ 3.18
2. สามารถแสดงผลเส้นทางการบินบนแผนที่ประเทศไทยและประเทศใกล้เคียงในระบบใหม่โดยต้องแสดงการเคลื่อนไหวของอากาศยานทุกๆ 4 วินาทีและแสดงสถิติความถูกต้องของอากาศยานเมื่อผ่านการกรองและขจัดความซ้ำซ้อนของข้อมูล และต้องสามารถย่อและขยายแผนที่ได้ดังแสดงได้ในรูปที่ 3.19
3. สามารถแสดงผลเส้นทางการบินบนแผนที่ประเทศไทยและประเทศใกล้เคียงในระบบเก่าและระบบใหม่พร้อมกันโดยต้องแสดงการเคลื่อนไหวของอากาศยานทุกๆ 4 วินาทีและแสดงสถิติของอากาศยานได้ตามเรดาร์ ต้องสามารเลือกที่จะดูระบบเก่าหรือระบบใหม่ได้ และต้องสามารถย่อและขยายแผนที่ได้ดังแสดงได้ในรูปที่ 3.20



รูปที่ 3.18 แสดงผลสถิติในระบบเก่าทางจอภาพ



รูปที่ 3.19 แสดงผลสถิติในระบบที่ผ่านการขจัดและการกรองข้อมูลทางจอภาพ



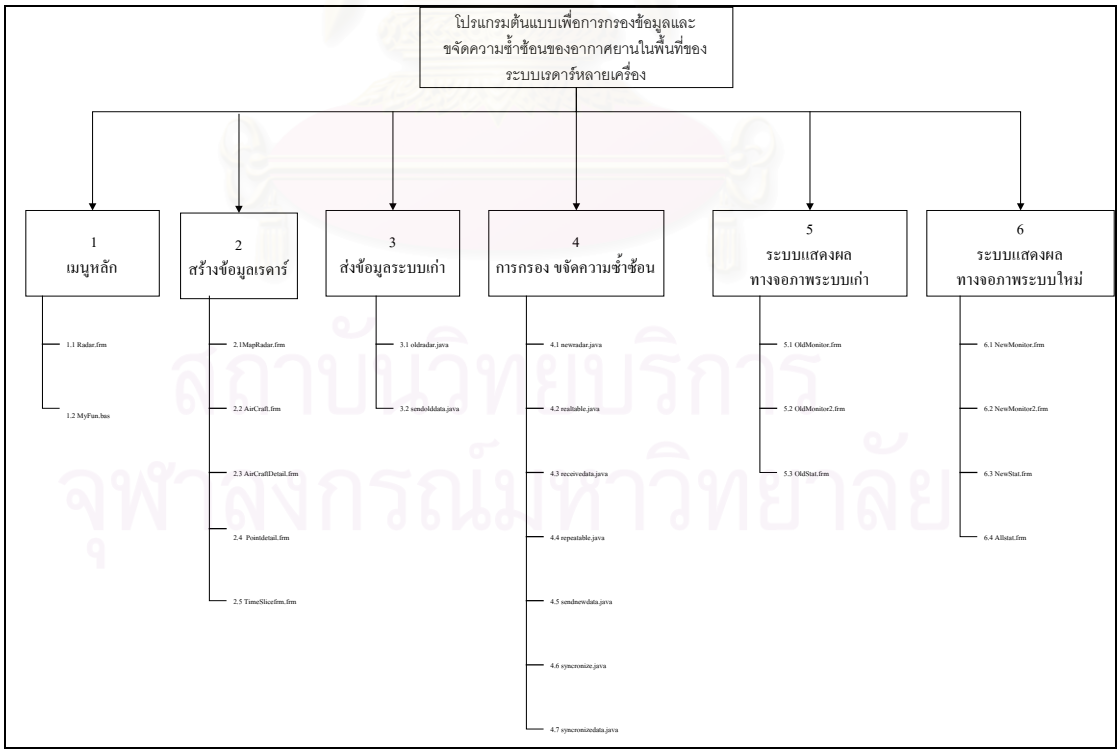
รูปที่ 3.20 แสดงผลสถิติในระบบทั้งสองระบบทางจอภาพ

บทที่ 4
การพัฒนาระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

ในการพัฒนาระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องผู้วิจัยได้จัดทำโปรแกรมหลักเพื่อทำงานในส่วนต่างๆดังนี้

1. เมนูหลัก
2. การสร้างข้อมูลเรดาร์
3. การส่งข้อมูลแบบเก่า
4. ระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานและการส่งข้อมูล
5. ระบบแสดงผลทางจอภาพระบบเก่า
6. ระบบแสดงผลทางจอภาพระบบใหม่

รายละเอียดและขั้นตอนการพัฒนาระบบโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องแสดงได้ดังรูปที่ 4.1 และรายละเอียดของโปรแกรมบางส่วนที่สำคัญจะปรากฏในภาคผนวก ค.



รูปที่ 4.1 การพัฒนาโปรแกรมกรองข้อมูลและการขจัดความซ้ำซ้อน

4.1 เมนูหลัก

โปรแกรมเมนูหลักทำหน้าที่แสดงโปรแกรมย่อยทั้งหมดของระบบ จะประกอบด้วยโปรแกรมย่อย 2 โปรแกรมคือ

1.1 Radar.frm แสดงเมนูหลัก ประกอบด้วยส่วนต่างๆคือ

1.1.1 RadarPosition_Click	โปรแกรมย่อยเรียกโปรแกรมสร้างเรดาร์
1.1.2 TrackPlan_Click	โปรแกรมย่อยเรียกโปรแกรมสร้างเส้นทางการบิน
1.1.3 SendDataOld_Click	โปรแกรมย่อยเรียกโปรแกรมแสดงเส้นทางการบินแบบเก่า
1.1.4 SendDataNew_Click	โปรแกรมย่อยเรียกโปรแกรมแสดงเส้นทางการบินที่ผ่านการกรองและขจัดความซ้ำซ้อนของข้อมูล
1.1.5 Bothsystem_Click	เรียกโปรแกรมเส้นการบินทั้งสองระบบ

1.2 MyFun.bas เก็บค่าตัวแปรต่างๆที่ใช้ในระบบ

4.2 การสร้างข้อมูลเรดาร์

โปรแกรมเริ่มต้นจากการวาดแผนที่ประเทศไทย จะอ่านข้อมูลจากแฟ้มข้อมูลตำแหน่งทางภูมิศาสตร์ของประเทศไทย และประเทศใกล้เคียงมาสร้างเป็นแผนที่ เมื่อมีการสร้างเรดาร์ ลบเรดาร์หรือย้ายเรดาร์ รวมทั้งการสร้างเส้นทางการบิน สร้างโปรโตคอล และสร้างสถิติการบิน โปรแกรมที่ถูกเรียกใช้ในโปรแกรมต่างดังมีรายละเอียดต่อไปนี้

2.1 MapRadar.frm สร้างแผนที่ในระบบเรดาร์โดยอ่านข้อมูลตำแหน่งทางภูมิศาสตร์ของประเทศไทยและประเทศใกล้เคียง เพื่อแสดงเป็นแผนที่และแสดงตำแหน่งของเมาส์ตามตำแหน่งทางภูมิศาสตร์ สร้างและปรับปรุงตำแหน่งของเรดาร์

2.1.1 FileOpen_Click	โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์และแสดงตำแหน่งของเรดาร์
2.1.2 FileClose_Click	โปรแกรมย่อยปิดข้อมูลเรดาร์และแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
2.1.3 FileSave_Click	โปรแกรมย่อยบันทึกแฟ้มข้อมูลเรดาร์เก็บตำแหน่งของเรดาร์

2.1.4 FileSavaAs_Click โปรแกรมย่อยบันทึกแฟ้มข้อมูลเรดาร์เก็บตำแหน่งของเรดาร์โดยสามารถกำหนดชื่อได้ตามที่ต้องการ

2.1.5 PoltNeighborBorder โปรแกรมย่อยแสดงแผนที่ประเทศใกล้เคียง

2.1.6 PoltThaiBorder โปรแกรมย่อยแสดงแผนที่ประเทศไทย

2.1.7 InRadarArea โปรแกรมย่อยเพื่อทดสอบว่าจุดที่กำหนดอยู่ในรัศมีของเรดาร์หรือไม่

2.1.8 RadarCreate_Click โปรแกรมย่อยสร้างรัศมีของเรดาร์จากจุดศูนย์กลาง

2.1.9 RadarMove_Click โปรแกรมย่อยย้ายรัศมีของเรดาร์โดยการ

- ทดสอบว่าจุดที่ต้องการย้ายอยู่ในรัศมีของเรดาร์เครื่องไหน
- รับตำแหน่งจุดศูนย์กลางของตำแหน่งใหม่
- ย้ายรัศมีของเรดาร์ไปยังจุดที่ต้องการ
- แสดงรัศมีของเรดาร์ทั้งหมดที่ได้รับการปรับปรุง

2.1.10 RadarDelete_Click โปรแกรมย่อยลบรัศมีของเรดาร์โดยการ

- ทดสอบว่าจุดที่ต้องการลบอยู่ในรัศมีของเรดาร์เครื่องไหน
- ลบรัศมีของเรดาร์ไปยังจุดที่ต้องการ
- แสดงรัศมีของเรดาร์ทั้งหมดที่ได้รับการปรับปรุง

2.1.11 ViewModify1_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด100%

2.1.12 ViewModify2_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด200%

2.1.13 ViewModify4_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด400%

2.1.14 ViewModify8_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด800%

2.2 AirCraft.frm สร้างเส้นทางการบิน สร้างโปรโตคอลเส้นทางการบิน เปิดและบันทึกเส้นทางการบินบนแฟ้มข้อความ

2.2.1 FileRadarOpen_Click โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์และแสดงตำแหน่งของเรดาร์

2.2.2 FileAirCraftOpen_Click โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์ เส้นทางการบิน และแสดงตำแหน่งของเรดาร์

2.2.3 FileAirCraftClose_Click โปรแกรมย่อยปิดแฟ้มข้อมูลเรดาร์ เส้นทางการบิน แสดง

	ตำแหน่งของเรดาร์และแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
2.2.4 FileSave_Click	โปรแกรมย่อยบันทึกแฟ้มข้อของข้อมูลเรดาร์ และเส้นทางการบินของอากาศยาน
2.2.5 FileSaveAs_Click	โปรแกรมย่อยบันทึกแฟ้มของข้อมูลเรดาร์ และเส้นทางการบินของอากาศยานโดยสามารถกำหนดชื่อได้ตามที่ต้องการ
2.2.6 TrackCreate_Click	โปรแกรมย่อยสร้างเส้นทางการบินโดยการ - รับชนิดของอากาศยานที่ได้รับการเลือก - รับฝ่ายของอากาศยานที่ได้รับการเลือก - รับความเร็วของอากาศยาน - รับจุดเส้นทางการบินของอากาศยาน - แสดงเส้นทางการบินของอากาศยาน
2.2.7 TrackMove_Click	โปรแกรมย่อยย้ายเส้นทางการบินโดยการ - ค้นหาตำแหน่งของเส้นทางการบินที่ต้องการย้าย - รับตำแหน่งของอากาศยานใหม่ - แสดงเส้นทางการบินของอากาศยานที่ได้รับการปรับปรุง
2.2.8 TrackDelete_Click	โปรแกรมย่อยลบเส้นทางการบินโดยการ - ค้นหาตำแหน่งของเส้นทางการบินที่ต้องการลบ - ลบเส้นทางการบิน - แสดงเส้นทางเส้นบินของอากาศยานที่ได้รับการปรับปรุง
2.2.9 TrackDataCreate_Click	โปรแกรมย่อยสร้างข้อมูลของเรดาร์จากข้อมูลเส้นทางการบินและสถานที่ตั้งของเรดาร์
2.2.10 CreateTimeSlice_Click	โปรแกรมย่อยรับค่าส่วนของวินาทีเพื่อใช้ในการสร้างข้อมูลของเรดาร์
2.2.11 ViewModify1_Click	โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด100%
2.2.12 ViewModify2_Click	โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด200%

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2.13 ViewModify4_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด400%

2.2.14 ViewModify8_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด800%

2.3 AirCraftDetail.frm แสดงและเก็บรายละเอียดของอากาศยาน เครื่องหมายของอากาศยาน

2.3.1 Combo1_Click โปรแกรมย่อยแสดงภาพและความเร็วสูงสุดของอากาศยาน

2.3.2 Command2_Click โปรแกรมย่อยเก็บค่าความเร็วและฝ่ายของอากาศ

2.3.3 Command3_Click โปรแกรมย่อยยกเลิกการเลือกอากาศยาน

2.4 Pointdetail.frm แสดงและเก็บตำแหน่งของเส้นทางการบินที่ได้กำหนด

2.4.1 Command1_Click โปรแกรมย่อยรับค่าตำแหน่งของเส้นทางการบิน

2.4.2 Command2_Click โปรแกรมย่อยยกเลิกตำแหน่งของเส้นทางการบิน

2.4.3 Command3_Click โปรแกรมย่อยจบการกำหนดเส้นทางการบิน

2.5 TimeSlicefrm.frm รับข้อมูลช่วงของเวลาในส่วนของวินาที

4.3 การส่งข้อมูลแบบเก่า

โปรแกรมเริ่มต้นจากการอ่านข้อมูลจากแฟ้มข้อมูลโปรโตคอลแล้วสร้างข้อมูลเรดาร์เพื่อส่งไปแสดงผล ยังโปรแกรมแสดงผล ดังมีรายละเอียดต่อไปนี้

3.1 oldradar.java สร้างข้อมูลเส้นทางการบินและเรดาร์ในระบบเก่า

3.1.1 readpositionradar โปรแกรมย่อยอ่านตำแหน่งของเรดาร์จากแฟ้มข้อมูล

3.1.2 readnumberradar โปรแกรมย่อยอ่านจำนวนของเรดาร์จากแฟ้มข้อมูล

3.1.3 openfile โปรแกรมย่อยเปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

3.1.4 closeFile โปรแกรมย่อยปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

3.1.5 oldradar ส่งข้อมูลที่ตั้งของเรดาร์

3.1.6 main โปรแกรมหลักเรียก class oldradar

3.2 sendolddata.java ส่งข้อมูลไปยังจอแสดงผลระบบเก่าโดยผ่านทางช่องส่งสัญญาณ

3.2.1 senddata	ส่งข้อมูลผ่านช่องส่งสัญญาณ
3.2.2 clientSendData	โปรแกรมย่อยเตรียมข้อมูลเป็นกลุ่มข้อมูล(package)เพื่อทำการส่งข้อมูล
3.2.3 openfile	โปรแกรมย่อยเปิดแฟ้มข้อมูลของเรดาร์
3.2.4 closeFile	โปรแกรมย่อยปิดแฟ้มข้อมูลของเรดาร์
3.2.5 openport	โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล
3.2.6 readprotocol	โปรแกรมย่อยอ่านข้อมูลของเรดาร์จากแฟ้มข้อมูลของเรดาร์
3.2.7 readprotocollong	โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นตัวเลข
3.2.8 readprotocolstring	โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นอักขระ
3.2.9 run	โปรแกรมวงวนไม่จุดจบ(infinite loop)เพื่อส่งข้อมูล

4.4 ระบบกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานและการส่งข้อมูล

โปรแกรมเริ่มต้นจากการอ่านข้อมูลจากแฟ้มข้อโปรโตคอลเพื่อทำการส่งต่อไปยังช่องรับสัญญาณ เพื่อขจัดความซ้ำซ้อนและกรองข้อมูลหลังจากขจัดความซ้ำซ้อนและกรองข้อมูลแล้วระบบจะสร้างข้อมูลเรดาร์เพื่อส่งไปแสดงผลโดยช่องส่งสัญญาณ ยังโปรแกรมแสดงผล ดังมีรายละเอียดต่อไปนี้

4.1 newradar.java อ่านข้อมูลเรดาร์ในระบบใหม่ เริ่มเปิดข้อมูลจากแฟ้มของเรดาร์เพื่ออ่านหมายเลขและตำแหน่งของเรดาร์

4.1.1 readpositionradar	โปรแกรมย่อยอ่านตำแหน่งของเรดาร์จากแฟ้มข้อมูล
4.1.2 readnumberradar	โปรแกรมย่อยอ่านจำนวนของเรดาร์จากแฟ้มข้อมูล
4.1.3 openfile	โปรแกรมย่อยเปิดแฟ้มข้อมูลที่ตั้งของเรดาร์
4.1.4 closeFile	โปรแกรมย่อยปิดแฟ้มข้อมูลที่ตั้งของเรดาร์
4.1.5 newradar	ส่งข้อมูลที่ตั้งของเรดาร์
4.1.6 Main	โปรแกรมหลักเรียก class newradar

4.2 realtable.java เก็บข้อมูลของอากาศยานข้อมูลเพื่อการนำไปใช้ในการขจัดความซ้ำซ้อน

4.3 receivedata.java รับข้อมูลจากช่องรับสัญญาณ

4.3.1 receivedata	รับข้อมูลจากช่องรับสัญญาณ
4.3.2 ClientSendData	โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ
4.3.3 openport	โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล
4.3.4 waitforpacket	โปรแกรมย่อยรับข้อมูลมาเก็บไว้เพื่อทำการส่งต่อ
4.3.5 run	โปรแกรมวงวนไม่จุดจบเพื่อรับข้อมูล

4.4 repeatable.java เก็บข้อมูลอากาศยานที่ซ้ำซ้อน

4.4.1 repeatedtable	เพื่อเก็บข้อมูล
4.4.2 addrepeat	โปรแกรมย่อยเพิ่มข้อมูลในตาราง
4.4.3 clear	โปรแกรมย่อยลบข้อมูลจากตาราง

4.5 senddata.java ส่งข้อมูลที่ผ่านการขจัดความซ้ำซ้อนและกรองข้อมูลไปยังจอแสดงผลระบบใหม่

4.5.1 senddata	ส่งข้อมูลผ่านช่องส่งสัญญาณ
4.5.2 ClientSendData	โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ
4.5.3 openfile	โปรแกรมย่อยเปิดแฟ้มข้อมูลของเรดาร์
4.5.4 closeFile	โปรแกรมย่อยปิดแฟ้มข้อมูลของเรดาร์
4.5.5 openport	โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล
4.5.6 readprotocol	โปรแกรมย่อยอ่านข้อมูลจากแฟ้มข้อมูล
4.5.7 readprotocollong	โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นตัวเลข
4.5.8 readprotocolstring	โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นอักขระ
4.5.9 run	โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

4.6 syncronize.java เตรียมข้อมูลเพื่อการขจัดความซ้ำซ้อน

4.6.1 syncronize	เตรียมข้อมูลเพื่อการขจัดความซ้ำซ้อน
4.6.2 openport	โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล
4.6.3 storedata	โปรแกรมย่อยเก็บข้อมูลของเรดาร์
4.6.4 storeempty	โปรแกรมย่อยลบข้อมูลทั้งหมด
4.6.5 waitforpacket	โปรแกรมย่อยรับข้อมูลมาเก็บไว้เพื่อทำการส่งต่อ

4.6.6 run	โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

4.7 syncronizedata.java ขจัดความซ้ำซ้อนของข้อมูลเรดาร์

4.7.1 syncronizedata	ขจัดความซ้ำซ้อนของข้อมูลเรดาร์
4.7.2 adddata	โปรแกรมย่อยเก็บข้อมูลของเรดาร์
4.7.3 assignid	โปรแกรมย่อยกำหนดหมายเลขของอากาศยาน
4.7.4 ClientSendData	โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ
4.7.5 findclass	โปรแกรมย่อยเปรียบเทียบชนิดของอากาศยาน
4.7.6 finddirection	โปรแกรมย่อยเปรียบเทียบทิศทางของอากาศยาน
4.7.7 findposition	โปรแกรมย่อยเปรียบเทียบตำแหน่งของอากาศยาน
4.7.8 findradius	โปรแกรมย่อยคำนวณตำแหน่งของอากาศยาน
4.7.9 findrepeat	โปรแกรมย่อยหาความซ้ำซ้อนของอากาศยาน
4.7.10 findspeed	โปรแกรมย่อยคำนวณความเร็วของอากาศยาน
4.7.11 findtime	โปรแกรมย่อยคำนวณความเวลาของอากาศยาน
4.7.12 inradius	โปรแกรมย่อยคำนวณตำแหน่งของอากาศยานว่าอยู่ในรัศมีของเรดาร์หรือไม่
4.7.13 makeprotocoldata	โปรแกรมย่อยสร้างโปรโตคอลของเรดาร์
4.7.14 missingdata	โปรแกรมย่อยปรับค่าการซ้ำซ้อนของอากาศยาน
4.7.15 openport	โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล
4.7.16 sendradarposition	โปรแกรมย่อยส่งตำแหน่งของเรดาร์
4.7.17 run	โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

4.5 ระบบแสดงผลทางจอภาพระบบเก่า

รับข้อมูลอากาศยานและแสดงผลของอากาศยานบนแผนที่ประเทศไทยและประเทศใกล้เคียงเพื่อสร้างเป็นแผนที่เมื่อไม่มีการกรองข้อมูล โปรแกรมที่ถูกเรียกใช้ในโปรแกรมต่างๆดังมีรายละเอียดต่อไปนี้

5.1 OldMonitor.frm สร้างแผนที่ในระบบแสดงผลแบบเก่า

5.1.1 Form_Load	โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์
5.1.2 Timer1_Timer	โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน

5.1.3 PlotMap	โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
5.1.4 DrawFlight	โปรแกรมย่อยแสดงอากาศยานเริ่มต้น
5.1.5 ReDrawFlight	โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง
5.1.6 DrawRadarNuber	โปรแกรมย่อยแสดงรัศมีของเรดาร์
5.1.7 ReDrawRadar	โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง
5.1.8 GetRadarACPosition	โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน
5.1.9 RedrawF	โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

5.2 OldMonitor2.frm สร้างแผนที่ในระบบแสดงผลแบบเก่าที่ทำงานร่วมกับระบบใหม่

5.2.1 Form_Load	โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์
5.2.2 Timer1_Timer	โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน
5.2.3 PlotMap	โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
5.2.4 DrawFlight	โปรแกรมย่อยแสดงอากาศยานเริ่มต้น
5.2.5 ReDrawFlight	โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง
5.2.6 DrawRadarNuber	โปรแกรมย่อยแสดงรัศมีของเรดาร์
5.2.7 ReDrawRadar	โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง
5.2.8 GetRadarACPosition	โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน
5.2.9 RedrawF	โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

5.3 OldStat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงในระบบเก่า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.6 ระบบแสดงผลทางจอภาพระบบใหม่

รับข้อมูลอากาศยานและแสดงผลของอากาศยานบนการแผนที่ประเทศไทยและประเทศใกล้เคียงเพื่อสร้างเป็นแผนที่เมื่อมีการกรองข้อมูล โปรแกรมที่ถูกเรียกใช้ในโปรแกรมต่างๆดังมีรายละเอียดต่อไปนี้

6.1 NewMonitor.frm สร้างแผนที่ในระบบแสดงผลเมื่อมีการขจัดความซ้ำซ้อน

6.1.1 Form_Load	โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์
6.1.2 Timer1_Timer	โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน
6.1.3 PlotMap	โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
6.1.4 DrawFlight	โปรแกรมย่อยแสดงอากาศยานเริ่มต้น
6.1.5 ReDrawFlight	โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง
6.1.6 DrawRadarNuber	โปรแกรมย่อยแสดงรัศมีของเรดาร์
6.1.7 ReDrawRadar	โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง
6.1.8 GetRadarACPosition	โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน
6.1.9 RedrawF	โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

6.2 NewMonitor2.frm สร้างแผนที่ในระบบแสดงผลที่ทำงานร่วมกับระบบเก่า

6.2.1 Form_Load	โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์
6.2.2 Timer1_Timer	โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน
6.2.3 PlotMap	โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง
6.2.4 DrawFlight	โปรแกรมย่อยแสดงอากาศยานเริ่มต้น
6.2.5 ReDrawFlight	โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง
6.2.6 DrawRadarNuber	โปรแกรมย่อยแสดงรัศมีของเรดาร์
6.2.7 ReDrawRadar	โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง
6.2.8 GetRadarACPosition	โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน
6.2.9 RedrawF	โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.3 NewStat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงในระบบใหม่

6.4 Allstat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงทั้งสองระบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
การทดสอบโปรแกรม

ในบทนี้จะกล่าวถึงการทดสอบโปรแกรมการพัฒนาโปรแกรมต้นแบบเพื่อกรองข้อมูล และการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง การทดสอบจะเริ่มจากสร้างข้อมูลเรดาร์จำลองเพื่อใช้ในการทดสอบ โดยข้อมูลได้ดังกล่าวจะถูกส่งไปให้ระบบกรองและขจัดความซ้ำซ้อนของพื้นที่ร่วมของระบบเรดาร์หลายตัวทำการกรอง และขจัดความซ้ำซ้อนแล้วจึงนำข้อมูลส่งไปยังระบบแสดงผลทางจอภาพ และข้อมูลอีกส่วนหนึ่งจะถูกส่งไปยังระบบเก่าเพื่อให้เห็นถึงปัญหาของระบบคือมีการซ้ำซ้อนของอากาศยาน ส่วนข้อมูลที่ผ่านการกรองและขจัดความซ้ำซ้อนจะต้องมาคำนวณหาความถูกต้องของระบบ

5.1 เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ

เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบต้องมีคุณสมบัติดังต่อไปนี้

1. เครื่องคอมพิวเตอร์ซึ่งมีซีพียูเพนเทียมสาม 500 เมกะเฮิร์ต(Pentium III 500 MHz)
2. หน่วยความจำหลัก 128 เมกะ ไบต์
3. ฮาร์ดดิสก์ขนาด 13 กิกะไบต์
4. ตัวปรับเครือข่าย(Network Adapter)
5. ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 98 (Windows 98)

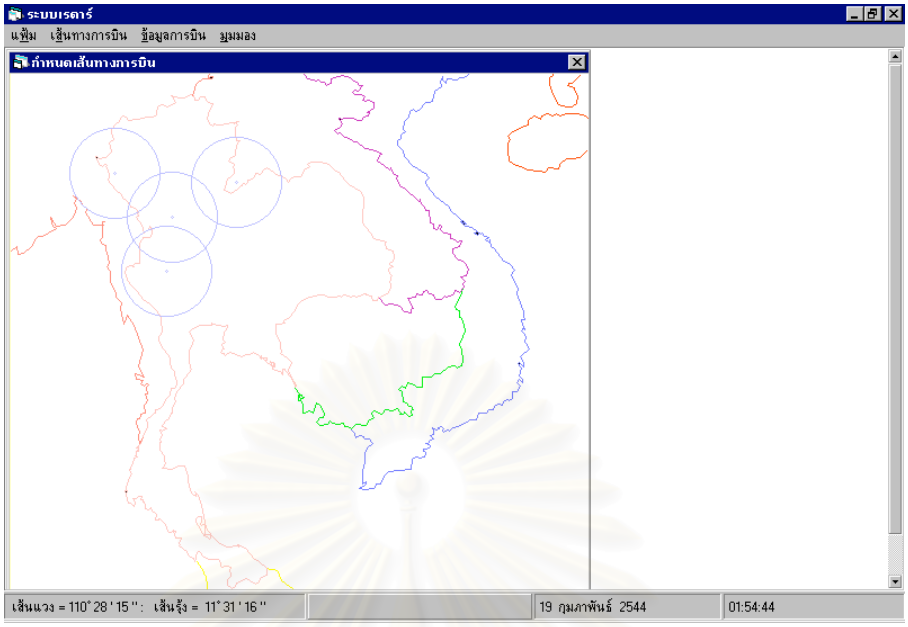
5.2 โปรแกรมสำเร็จที่ใช้ในการทดสอบ

1. โปรแกรมไมโครซอฟท์วิชวลเบสิก เวอร์ชัน 5 ขึ้นไป
2. โปรแกรมเจบิวต์เดอร์ เวอร์ชัน 2.0 ขึ้นไป
3. โปรแกรมไมโครซอฟท์แอกเซส
4. โปรแกรมเรดาร์ (radar.exe)
5. แฟ้มข้อมูลของเจบิวต์เดอร์ โครงการ(radar.jpr)

5.3 ขั้นตอนในการทดสอบโปรแกรม

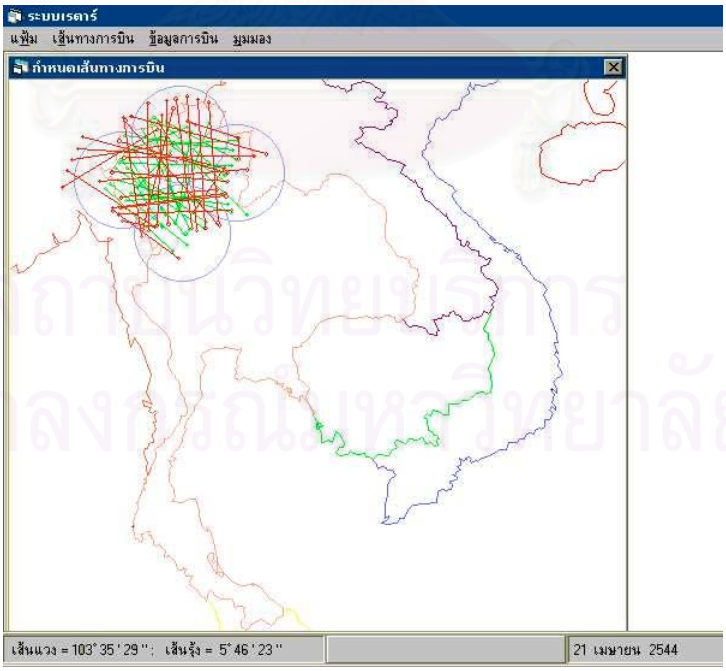
1. รันโปรแกรม Radar.exe โปรแกรมจะแสดงหน้าเมนูหลัก เลือกตัวเลือกเรดาร์เพื่อสร้างตำแหน่งของเรดาร์ที่จะใช้ในการทดสอบ
2. สร้างตำแหน่งของเรดาร์อย่างน้อย 3 แบบการสร้างตำแหน่งที่กำหนดในข้อจำกัดของวิทยานิพนธ์ การสร้างรัศมีของเรดาร์ต้องให้เรดาร์มีพื้นที่ซ้อนทับกันไม่เกิน2เครื่องและสร้างเรดาร์ 4 เครื่องในการทดลองแต่ละครั้ง ดังแสดงในรูปที่ 5.1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

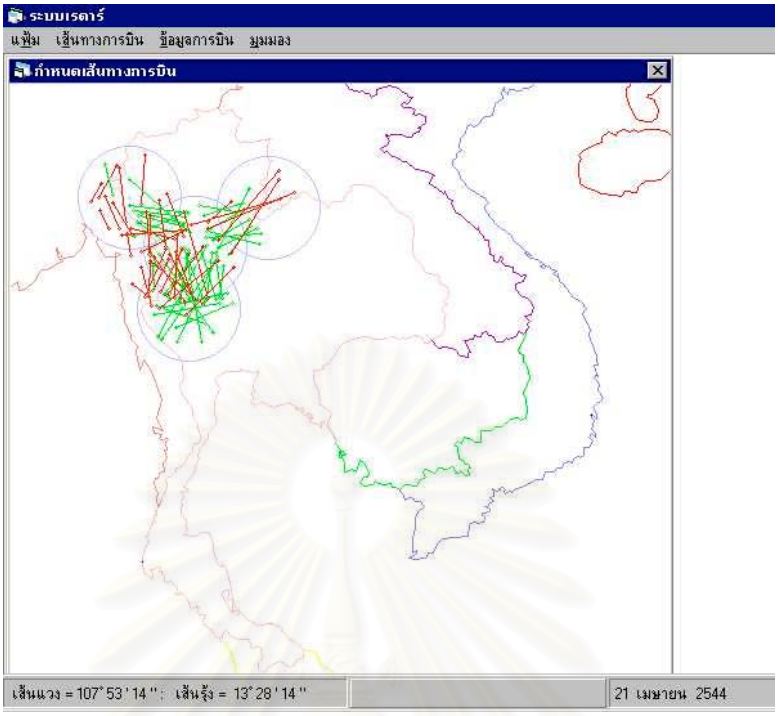


รูปที่ 5.1 สร้างรัศมีของเรดาร์

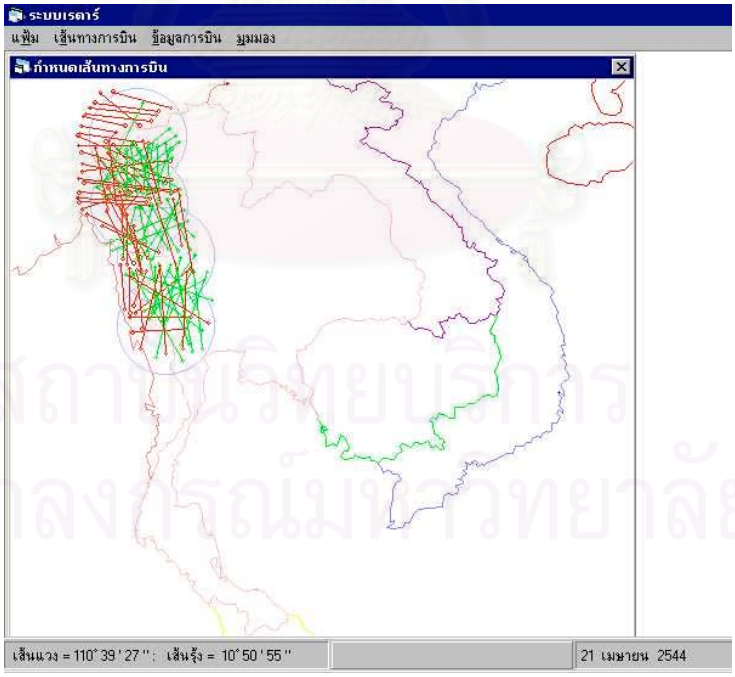
3. เลือกชนิดของอากาศยานและเลือกเส้นทางการบินเพื่อสร้างเส้นทางการบิน
4. สร้างเส้นทางการบินของอากาศยานอย่างน้อยที่สุด 4 เส้นทางแต่ไม่เกิน 200 เส้นทาง โดยเลือกชนิดและความเร็วของเครื่องบินแตกต่างกันไป และเวลาในการทำการบินแตกต่างกันไปตามดังแสดงได้ดังรูปที่ 5.2 5.3 และ 5.4 แสดงการทดสอบทั้ง 3 แบบ



รูปที่ 5.2 สร้างเส้นทางการบินของอากาศยานแบบที่ 1



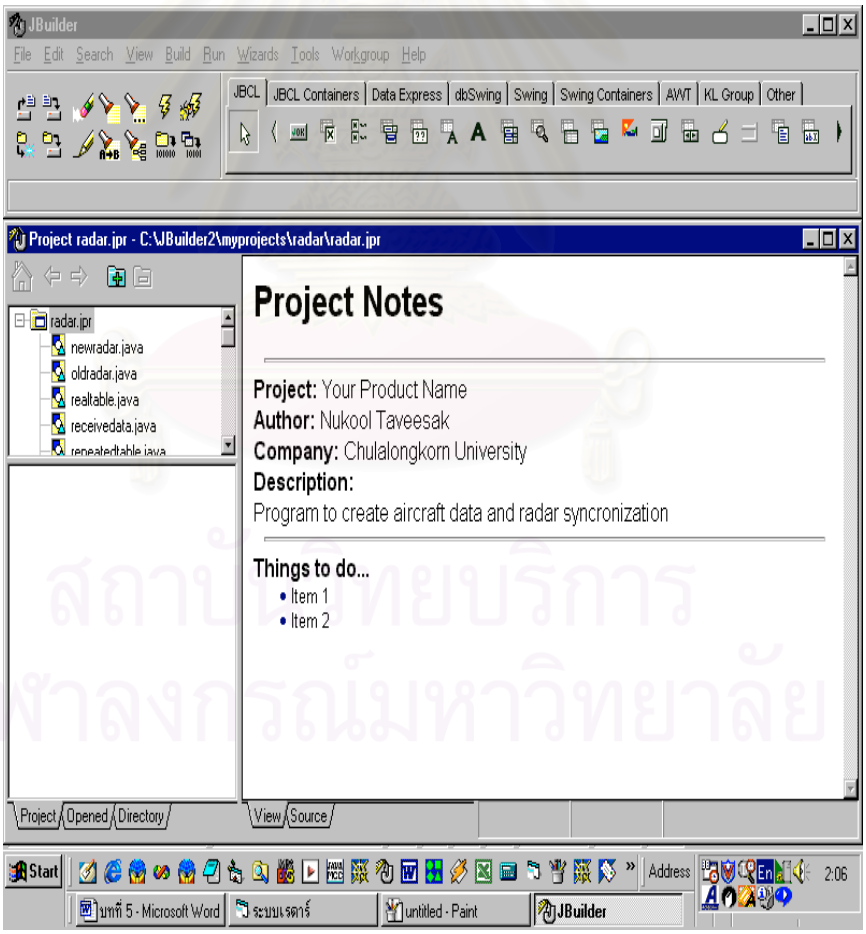
รูปที่ 5.3 สร้างเส้นทางการบินของอากาศยานแบบที่ 2



รูปที่ 5.4 สร้างเส้นทางการบินของอากาศยานแบบที่ 3

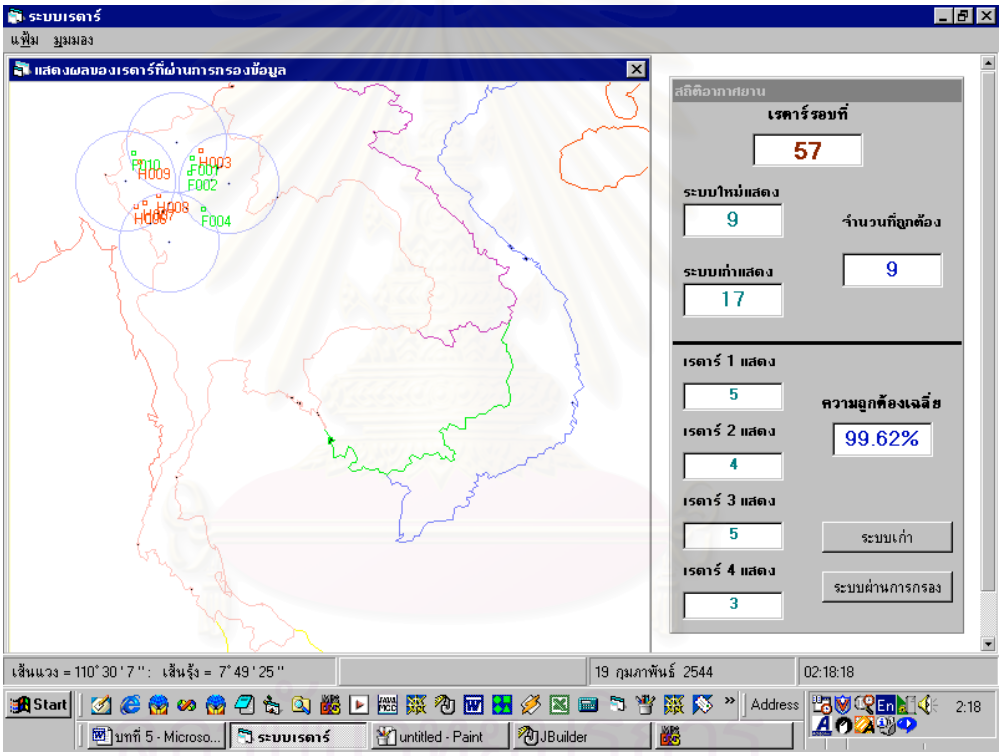
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. สร้างข้อมูลการบินเมื่อได้สร้างเส้นทางการบินที่ต้องการจะทดสอบแล้วจึงทำการจำลองข้อมูลโปรโตคอลของเรดาร์
6. เลือกตัวเลือกส่งข้อมูลผ่าน 2 ระบบเพื่อทำการทดสอบ
7. ทำการรันโปรแกรมจาวาโดยเปิดโปรเจ็ค radar.jpr ดังสามารถแสดงได้ในรูปที่ 5.5
8. ถ้าต้องการเห็นผลระบบเก่า เลือกตัวเลือกระบบเก่าเพื่อให้เห็นว่าถ้าไม่ผ่านระบบการกรองข้อมูลแล้วผลที่ได้จากระบบเก่าจะมีความคลาดเคลื่อนและยากต่อการวิเคราะห์ประเมินภัยคุกคาม
9. ถ้าต้องการเห็นผลผ่านระบบการกรองและขจัดความซ้ำซ้อน เลือกตัวเลือกระบบผ่านการกรอง ส่งข้อมูลผ่านระบบการกรองและขจัดความซ้ำซ้อน เพื่อให้เห็นว่าหลังจากมีการขจัดความซ้ำซ้อนออกไปแล้ว ดังแสดงได้ดังรูปที่ 5.6



รูปที่ 5.5 รันโปรแกรมส่งกรองและขจัดความซ้ำซ้อนของข้อมูล

10. ทำการทดลองซ้ำ 30 ครั้งของเส้นทางการบินที่ต่างกันโดยข้อมูลของการทดลองครั้งที่ 1 ตารางที่ 5.1 โดยตารางจะแสดงรอบที่หมุนของเรดาร์ จำนวนอากาศยานของเรดาร์แต่ละเครื่อง จำนวนอากาศยานที่ความผิดพลาดต่อรอบ จำนวนอากาศยานที่เรดาร์ในระบบเก่าตรวจจับได้รวม จำนวนอากาศยานที่เป็นจริง ความถูกต้องของระบบ
11. เก็บรวบรวมสถิติและแสดงผลสถิติจากการทดลอง 30 ครั้ง เพื่อคำนวณหาความถูกต้องสะสมเฉลี่ยโดยรายละเอียดของการทดลองครั้งที่ 2-5 สามารถแสดงได้ในภาคผนวก ง.



รูปที่ 5.6 การทดสอบโปรแกรมกรองข้อมูลและการขจัดความซ้ำซ้อน

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์รอบที่	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศยานสะสม	ความถูกต้อง (%)
	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม		
0	3	1	+5	0	+9	+8	9	1	1	9	88.89
1	3	1	5	0	9	+9	9	0	1	18	94.44
2	3	1	5	0	9	9	9	0	1	27	96.3
3	3	1	5	0	9	9	9	0	1	36	97.22
4	3	1	5	0	9	9	9	0	1	45	97.78
5	3	1	5	0	9	9	9	0	1	54	98.15

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
6	3	1	5	0	9	9	9	0	1	63	98.41
7	3	1	5	0	9	9	9	0	1	72	98.61
8	3	1	5	0	9	9	9	0	1	81	98.77
9	3	1	5	0	9	9	9	0	1	90	98.89
10	3	1	5	0	9	9	9	0	1	99	98.99
11	3	1	5	0	9	9	9	0	1	108	99.07
12	3	1	5	0	9	9	9	0	1	117	99.15
13	3	1	5	0	9	9	9	0	1	126	99.21
14	3	1	5	0	9	9	9	0	1	135	99.26
15	3	1	5	0	9	9	9	0	1	144	99.31
16	3	1	5	0	9	9	9	0	1	153	99.35
17	3	1	5	0	9	9	9	0	1	162	99.38
18	3	1	5	0	9	9	9	0	1	171	99.42
19	3	1	5	0	9	9	9	0	1	180	99.44
20	3	1	5	0	9	9	9	0	1	189	99.47
21	3	+2	5	0	+10	9	9	0	1	198	99.49
22	3	2	5	0	10	9	9	0	1	207	99.52
23	3	2	5	0	10	9	9	0	1	216	99.54
24	3	2	5	0	10	9	9	0	1	225	99.56
25	3	2	5	0	10	9	9	0	1	234	99.57
26	3	2	5	0	10	9	9	0	1	243	99.59
27	3	2	-4	0	-9	9	9	0	1	252	99.6
28	+4	2	+5	0	+11	9	9	0	1	261	99.62
29	4	2	5	0	11	9	9	0	1	270	99.63
30	4	2	5	0	11	9	9	0	1	279	99.64
31	4	2	5	0	11	9	9	0	1	288	99.65
32	4	2	5	0	11	9	9	0	1	297	99.66
33	4	2	5	0	11	9	9	0	1	306	99.67
34	4	2	5	0	11	9	9	0	1	315	99.68
35	4	+3	5	0	+12	9	9	0	1	324	99.69
36	+5	3	5	0	+13	9	9	0	1	333	99.7
37	5	3	5	0	13	9	9	0	1	342	99.71
38	5	3	5	0	13	9	9	0	1	351	99.72
39	5	3	5	0	13	9	9	0	1	360	99.72
40	5	+4	5	0	+14	9	9	0	1	369	99.73
41	5	4	5	0	14	9	9	0	1	378	99.74
42	5	4	5	0	14	9	9	0	1	387	99.74
43	5	4	5	0	14	9	9	0	1	396	99.75
44	5	4	5	0	14	9	9	0	1	405	99.75
45	5	4	5	0	14	9	9	0	1	414	99.76
46	5	4	5	0	14	9	9	0	1	423	99.76
47	5	4	5	0	14	9	9	0	1	432	99.77
48	5	4	5	0	14	9	9	0	1	441	99.77
49	5	4	5	0	14	9	9	0	1	450	99.78
50	5	4	5	+1	+15	9	9	0	1	459	99.78
51	5	4	5	1	15	9	9	0	1	468	99.79
52	5	4	5	1	15	9	9	0	1	477	99.79
53	5	4	5	1	15	9	9	0	1	486	99.79

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
54	5	4	5	1	15	9	9	0	1	495	99.8
55	5	4	5	1	15	9	9	0	1	504	99.8
56	5	4	5	1	15	9	9	0	1	513	99.81
57	5	4	5	+3	+17	9	9	0	1	522	99.81
58	5	4	5	3	17	9	9	0	1	531	99.81
59	5	4	5	3	17	9	9	0	1	540	99.81
60	5	4	5	+4	+18	9	9	0	1	549	99.82
61	5	4	5	4	18	9	9	0	1	558	99.82
62	5	4	5	4	18	9	9	0	1	567	99.82
63	5	4	5	4	18	9	9	0	1	576	99.83
64	5	4	5	4	18	9	9	0	1	585	99.83
65	5	4	5	4	18	9	9	0	1	594	99.83
66	5	4	5	4	18	9	9	0	1	603	99.83
67	5	4	5	4	18	9	9	0	1	612	99.84
68	5	4	5	4	18	9	9	0	1	621	99.84
69	5	4	5	4	18	9	9	0	1	630	99.84
70	5	4	5	4	18	9	9	0	1	639	99.84
71	5	4	5	4	18	9	9	0	1	648	99.85
72	5	4	5	4	18	9	9	0	1	657	99.85
73	5	4	5	4	18	9	9	0	1	666	99.85
74	5	4	5	4	18	9	9	0	1	675	99.85
75	5	4	5	4	18	9	9	0	1	684	99.85
76	5	4	5	4	18	9	9	0	1	693	99.86
77	5	4	5	4	18	9	9	0	1	702	99.86
78	5	4	5	4	18	9	9	0	1	711	99.86
79	5	4	5	4	18	9	9	0	1	720	99.86
80	5	4	5	4	18	9	9	0	1	729	99.86
81	5	4	5	4	18	9	9	0	1	738	99.86
82	5	4	5	4	18	9	9	0	1	747	99.87
83	5	4	5	4	18	9	9	0	1	756	99.87
84	5	4	5	4	18	9	9	0	1	765	99.87
85	5	4	5	4	18	9	9	0	1	774	99.87
86	5	4	5	4	18	9	9	0	1	783	99.87
87	5	4	5	4	18	9	9	0	1	792	99.87
88	5	4	5	4	18	9	9	0	1	801	99.88
89	5	4	5	4	18	9	9	0	1	810	99.88
90	5	4	5	4	18	9	9	0	1	819	99.88
91	5	4	5	4	18	9	9	0	1	828	99.88
92	5	4	5	4	18	9	9	0	1	837	99.88
93	5	4	5	4	18	9	9	0	1	846	99.88
94	5	4	5	4	18	9	9	0	1	855	99.88
95	5	4	5	4	18	9	9	0	1	864	99.88
96	5	4	5	4	18	9	9	0	1	873	99.89
97	5	4	5	4	18	9	9	0	1	882	99.89
98	5	4	5	-3	-17	9	9	0	1	891	99.89
99	5	4	5	+4	+18	9	9	0	1	900	99.89
100	5	4	5	4	18	9	9	0	1	909	99.89
101	5	-3	5	4	-17	9	9	0	1	918	99.89

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
102	5	3	5	4	17	9	9	0	1	927	99.89
103	5	3	5	4	17	9	9	0	1	936	99.89
104	5	3	5	4	17	9	9	0	1	945	99.89
105	5	3	-4	4	-16	9	9	0	1	954	99.9
106	-4	3	4	4	-15	9	9	0	1	963	99.9
107	4	3	4	4	15	9	9	0	1	972	99.9
108	4	3	4	4	15	9	9	0	1	981	99.9
109	4	3	4	4	15	9	9	0	1	990	99.9
110	4	3	4	4	15	9	9	0	1	999	99.9
111	4	3	4	4	15	9	9	0	1	1,008	99.9
112	4	3	4	4	15	9	9	0	1	1,017	99.9
113	4	3	4	4	15	9	9	0	1	1,026	99.9
114	4	3	4	4	15	9	9	0	1	1,035	99.9
115	4	3	4	4	15	9	9	0	1	1,044	99.9
116	4	3	4	4	15	9	9	0	1	1,053	99.91
117	4	3	4	4	15	9	9	0	1	1,062	99.91
118	4	3	4	4	15	9	9	0	1	1,071	99.91
119	4	3	4	4	15	9	9	0	1	1,080	99.91
120	4	3	4	4	15	9	9	0	1	1,089	99.91
121	4	3	4	4	15	9	9	0	1	1,098	99.91
122	4	3	4	4	15	9	9	0	1	1,107	99.91
123	4	3	4	4	15	9	9	0	1	1,116	99.91
124	4	3	4	4	15	9	9	0	1	1,125	99.91
125	4	3	4	4	15	9	9	0	1	1,134	99.91
126	4	3	4	4	15	9	9	0	1	1,143	99.91
127	4	3	4	4	15	9	9	0	1	1,152	99.91
128	4	3	4	4	15	9	9	0	1	1,161	99.91
129	4	3	-3	4	-14	9	9	0	1	1,170	99.91
130	4	3	3	4	14	9	9	0	1	1,179	99.92
131	4	3	3	4	14	9	9	0	1	1,188	99.92
132	4	3	3	4	14	9	9	0	1	1,197	99.92
133	4	3	-2	4	-13	9	9	0	1	1,206	99.92
134	4	3	2	4	13	9	9	0	1	1,215	99.92
135	4	3	2	4	13	9	9	0	1	1,224	99.92
136	4	3	2	4	13	9	9	0	1	1,233	99.92
137	4	3	2	4	13	9	9	0	1	1,242	99.92
138	4	3	2	4	13	9	9	0	1	1,251	99.92
139	4	3	2	4	13	9	9	0	1	1,260	99.92
140	4	3	2	4	13	9	9	0	1	1,269	99.92
141	4	3	2	4	13	9	9	0	1	1,278	99.92
142	4	3	2	4	13	9	9	0	1	1,287	99.92
143	4	3	2	4	13	9	9	0	1	1,296	99.92
144	-3	3	2	4	-12	9	9	0	1	1,305	99.92
145	3	3	2	4	12	9	9	0	1	1,314	99.92
146	3	3	2	4	12	9	9	0	1	1,323	99.92
147	3	3	2	4	12	9	9	0	1	1,332	99.92
148	3	3	2	4	12	9	9	0	1	1,341	99.93
149	3	3	2	4	12	9	9	0	1	1,350	99.93

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
150	3	3	2	4	12	9	9	0	1	1,359	99.93
151	3	3	2	4	12	9	9	0	1	1,368	99.93
152	3	-2	2	4	-11	9	9	0	1	1,377	99.93
153	3	2	2	4	11	-8	9	1	2	1,386	99.86
154	3	2	2	4	11	8	8	0	2	1,394	99.86
155	3	2	2	4	11	8	8	0	2	1,402	99.86
156	3	2	2	4	11	8	8	0	2	1,410	99.86
157	3	2	2	4	11	8	8	0	2	1,418	99.86
158	3	2	-1	4	-10	8	8	0	2	1,426	99.86
159	3	2	1	4	10	8	8	0	2	1,434	99.86
160	3	+3	1	4	+11	8	8	0	2	1,442	99.86
161	3	3	1	4	11	8	8	0	2	1,450	99.86
162	3	3	1	4	11	8	8	0	2	1,458	99.86
163	3	3	1	-3	-10	8	8	0	2	1,466	99.86
164	3	3	1	3	10	-7	8	1	3	1,474	99.8
165	3	3	1	3	10	7	7	0	3	1,481	99.8
166	3	3	1	3	10	7	7	0	3	1,488	99.8
167	3	3	-0	3	+9	7	7	0	3	1,495	99.8
168	3	3	0	3	9	7	7	0	3	1,502	99.8
169	3	3	0	3	9	7	7	0	3	1,509	99.8
170	3	3	0	3	9	7	7	0	3	1,516	99.8
171	3	3	0	3	9	7	7	0	3	1,523	99.8
172	3	3	0	3	9	7	7	0	3	1,530	99.8
173	3	3	0	3	9	7	7	0	3	1,537	99.8
174	3	3	0	3	9	7	7	0	3	1,544	99.81
175	3	3	0	3	9	7	7	0	3	1,551	99.81
176	3	3	0	3	9	7	7	0	3	1,558	99.81
177	3	-2	0	3	-8	7	7	0	3	1,565	99.81
178	3	2	0	3	8	-6	7	1	4	1,572	99.75
179	3	2	0	3	8	6	7	1	5	1,579	99.68
180	3	2	0	-1	-6	6	6	0	5	1,585	99.68
181	-2	2	0	1	-5	-5	5	0	5	1,590	99.69
182	2	2	0	1	5	-4	5	1	6	1,595	99.62
183	2	2	0	1	5	4	4	0	6	1,599	99.62
184	2	2	0	1	5	4	4	0	6	1,603	99.63
185	2	2	0	1	5	4	4	0	6	1,607	99.63
186	2	2	0	1	5	4	4	0	6	1,611	99.63
187	2	2	0	1	5	4	4	0	6	1,615	99.63
188	2	2	0	1	5	4	4	0	6	1,619	99.63
189	2	2	0	1	5	4	4	0	6	1,623	99.63
190	2	2	0	1	5	4	4	0	6	1,627	99.63
191	2	2	0	1	5	4	4	0	6	1,631	99.63
192	-1	2	0	1	-4	-3	4	1	7	1,635	99.57
193	1	2	0	1	4	3	3	0	7	1,638	99.57
194	1	2	0	1	4	3	3	0	7	1,641	99.57
195	1	2	0	1	4	3	3	0	7	1,644	99.57
196	1	2	0	1	4	3	3	0	7	1,647	99.57
197	1	2	0	1	4	3	3	0	7	1,650	99.58

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
198	1	2	0	1	4	3	3	0	7	1,653	99.58
199	1	2	0	1	4	3	3	0	7	1,656	99.58
200	1	2	0	1	4	3	3	0	7	1,659	99.58
201	1	2	0	1	4	3	3	0	7	1,662	99.58
202	1	2	0	1	4	3	3	0	7	1,665	99.58
203	1	2	0	1	4	3	3	0	7	1,668	99.58
204	-0	2	0	1	-3	3	3	0	7	1,671	99.58
205	0	2	0	1	3	-2	3	1	8	1,674	99.52
206	0	2	0	1	3	2	3	1	9	1,677	99.46
207	0	2	0	1	3	2	2	0	9	1,679	99.46
208	0	2	0	1	3	2	2	0	9	1,681	99.46
209	0	2	0	1	3	2	2	0	9	1,683	99.47
210	0	2	0	1	3	2	2	0	9	1,685	99.47
211	0	2	0	1	3	2	2	0	9	1,687	99.47
212	0	2	0	1	3	2	2	0	9	1,689	99.47
213	0	-1	0	1	-2	2	2	0	9	1,691	99.47
214	0	1	0	1	2	-1	2	1	10	1,693	99.41
215	0	1	0	1	2	1	1	0	10	1,694	99.41
216	0	1	0	1	2	1	1	0	10	1,695	99.41
217	0	1	0	1	2	1	1	0	10	1,696	99.41
218	0	1	0	1	2	1	1	0	10	1,697	99.41
219	0	1	0	1	2	1	1	0	10	1,698	99.41
220	0	1	0	1	2	1	1	0	10	1,699	99.41
221	0	1	0	1	2	1	1	0	10	1,700	99.41
222	0	1	0	1	2	1	1	0	10	1,701	99.41
223	0	1	0	1	2	1	1	0	10	1,702	99.41
224	0	1	0	1	2	1	1	0	10	1,703	99.41
225	0	1	0	1	2	1	1	0	10	1,704	99.41
226	0	1	0	1	2	1	1	0	10	1,705	99.41
227	0	1	0	1	2	1	1	0	10	1,706	99.41
228	0	1	0	1	2	1	1	0	10	1,707	99.41
229	0	1	0	1	2	1	1	0	10	1,708	99.41
230	0	1	0	1	2	1	1	0	10	1,709	99.41
231	0	1	0	1	2	1	1	0	10	1,710	99.42
232	0	1	0	1	2	1	1	0	10	1,711	99.42
233	0	1	0	1	2	1	1	0	10	1,712	99.42
234	0	1	0	1	2	1	1	0	10	1,713	99.42
235	0	1	0	1	2	1	1	0	10	1,714	99.42
236	0	1	0	1	2	1	1	0	10	1,715	99.42
237	0	1	0	1	2	1	1	0	10	1,716	99.42
238	0	1	0	1	2	1	1	0	10	1,717	99.42
239	0	1	0	1	2	1	1	0	10	1,718	99.42
240	0	1	0	1	2	1	1	0	10	1,719	99.42
241	0	1	0	1	2	1	1	0	10	1,720	99.42
242	0	1	0	1	2	1	1	0	10	1,721	99.42
243	0	1	0	1	2	1	1	0	10	1,722	99.42
244	0	1	0	1	2	1	1	0	10	1,723	99.42
245	0	1	0	1	2	1	1	0	10	1,724	99.42

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
246	0	1	0	1	2	1	1	0	10	1,725	99.42
247	0	1	0	1	2	1	1	0	10	1,726	99.42
248	0	1	0	1	2	1	1	0	10	1,727	99.42
249	0	1	0	1	2	1	1	0	10	1,728	99.42
250	0	1	0	1	2	1	1	0	10	1,729	99.42
251	0	1	0	1	2	1	1	0	10	1,730	99.42
252	0	1	0	1	2	1	1	0	10	1,731	99.42
253	0	1	0	1	2	1	1	0	10	1,732	99.42
254	0	1	0	1	2	1	1	0	10	1,733	99.42
255	0	1	0	1	2	1	1	0	10	1,734	99.42
256	0	1	0	-0	-1	1	1	0	10	1,735	99.42
257	0	1	0	0	1	1	1	0	10	1,736	99.42
258	0	1	0	0	1	1	1	0	10	1,737	99.42
259	0	1	0	0	1	1	1	0	10	1,738	99.42
260	0	1	0	0	1	1	1	0	10	1,739	99.42
261	0	1	0	0	1	1	1	0	10	1,740	99.43
262	0	1	0	0	1	1	1	0	10	1,741	99.43
263	0	1	0	0	1	1	1	0	10	1,742	99.43
264	0	1	0	0	1	1	1	0	10	1,743	99.43
265	0	1	0	0	1	1	1	0	10	1,744	99.43
266	0	1	0	0	1	1	1	0	10	1,745	99.43
267	0	1	0	0	1	1	1	0	10	1,746	99.43
268	0	1	0	0	1	1	1	0	10	1,747	99.43
269	0	1	0	0	1	1	1	0	10	1,748	99.43
270	0	1	0	0	1	1	1	0	10	1,749	99.43
271	0	1	0	0	1	1	1	0	10	1,750	99.43
272	0	1	0	0	1	1	1	0	10	1,751	99.43
273	0	1	0	0	1	1	1	0	10	1,752	99.43
274	0	1	0	0	1	1	1	0	10	1,753	99.43
275	0	1	0	0	1	1	1	0	10	1,754	99.43
276	0	1	0	0	1	1	1	0	10	1,755	99.43
277	0	1	0	0	1	1	1	0	10	1,756	99.43
278	0	1	0	0	1	1	1	0	10	1,757	99.43
279	0	1	0	0	1	1	1	0	10	1,758	99.43
280	0	1	0	0	1	1	1	0	10	1,759	99.43
281	0	1	0	0	1	1	1	0	10	1,760	99.43
282	0	1	0	0	1	1	1	0	10	1,761	99.43
283	0	1	0	0	1	1	1	0	10	1,762	99.43
284	0	1	0	0	1	1	1	0	10	1,763	99.43
285	0	1	0	0	1	1	1	0	10	1,764	99.43
286	0	1	0	0	1	1	1	0	10	1,765	99.43
287	0	1	0	0	1	1	1	0	10	1,766	99.43
288	0	1	0	0	1	1	1	0	10	1,767	99.43
289	0	1	0	0	1	1	1	0	10	1,768	99.43
290	0	1	0	0	1	1	1	0	10	1,769	99.43
291	0	1	0	0	1	1	1	0	10	1,770	99.44
292	0	1	0	0	1	1	1	0	10	1,771	99.44
293	0	1	0	0	1	1	1	0	10	1,772	99.44

ตารางที่ 5.1 แสดงจำนวนอากาศยานต่อรอบของการทดสอบครั้งที่ 1

เรดาร์ รอบที่	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศยานสะสม	ความถูกต้อง (%)
	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม		
294	0	1	0	0	1	1	1	0	10	1,773	99.44
295	0	1	0	0	1	1	1	0	10	1,774	99.44
296	0	1	0	0	1	1	1	0	10	1,775	99.44
297	0	1	0	0	1	1	1	0	10	1,776	99.44
298	0	1	0	0	1	1	1	0	10	1,777	99.44
299	0	1	0	0	1	1	1	0	10	1,778	99.44
300	0	1	0	0	1	1	1	0	10	1,779	99.44
301	0	1	0	0	1	1	1	0	10	1,780	99.44
302	0	1	0	0	1	1	1	0	10	1,781	99.44
303	0	1	0	0	1	1	1	0	10	1,782	99.44
304	0	1	0	0	1	1	1	0	10	1,783	99.44
305	0	1	0	0	1	1	1	0	10	1,784	99.44
306	0	1	0	0	1	1	1	0	10	1,785	99.44
307	0	1	0	0	1	1	1	0	10	1,786	99.44
308	0	1	0	0	1	1	1	0	10	1,787	99.44
309	0	1	0	0	1	1	1	0	10	1,788	99.44
310	0	1	0	0	1	1	1	0	10	1,789	99.44
311	0	1	0	0	1	1	1	0	10	1,790	99.44
312	0	1	0	0	1	1	1	0	10	1,791	99.44
313	0	1	0	0	1	1	1	0	10	1,792	99.44
314	0	1	0	0	1	1	1	0	10	1,793	99.44
315	0	-0	0	0	-0	-0	1	1	11	1,794	99.39

หมายเหตุ : + หมายถึงมีอากาศยานเข้ามาในระบบ

- หมายถึงมีอากาศยานออกจากระบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
สรุปผลวิจัยและข้อเสนอแนะ

วิทยานิพนธ์ได้ดำเนินการออกแบบ และพัฒนาซอฟต์แวร์เพื่อกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง โดยมีผลสรุปดังต่อไปนี้

6.1 สรุปผลการวิจัย

ในการทำวิทยานิพนธ์ครั้งนั้น เนื่องจากข้อมูลของเรดาร์เป็นความลับของทางราชการ ดังนั้นผู้ทำการวิจัยได้สร้างโปรแกรมเพื่อสร้างข้อมูลของเรดาร์ เพื่อให้ได้โปรโตคอลใกล้เคียงกับโปรโตคอลของเรดาร์ให้มากที่สุด ผู้ทำการวิจัยได้อาศัยเอกสารโปรโตคอลของเรดาร์เป็นต้นแบบสร้างข้อมูลโปรโตคอลเส้นทางการบินของเรดาร์ ผู้ทำการวิจัยได้ออกแบบการสร้างเส้นทางการบินเป็นแบบกราฟฟิกซึ่งจะทำให้ผู้ใช้สามารถสร้างเส้นทางการบินสะดวกขึ้น ในการทำการวิจัยครั้งต้องใช้เครื่องคอมพิวเตอร์ในการทำการจำลองการข้อมูลจากเรดาร์ 4 เครื่องและอุปกรณ์ช่องอนุกรมหลายช่อง (Multiserial Port) และคอมพิวเตอร์เพื่อใช้ในการแสดงผลอีก 1 เครื่อง ในการทำการวิจัยผู้ทำการวิจัยได้จำลองการทำงานของเครื่องคอมพิวเตอร์หลายเครื่อง โดยอาศัยหลักการของมัลติโปรแกรมมิงที่แต่ละโปรแกรมจะทำงานอิสระจากกันเสมือนการทำงานของเครื่องคอมพิวเตอร์หลายเครื่อง และใช้ช่องรับสัญญาณของเครื่องคอมพิวเตอร์แทนอุปกรณ์ช่องอนุกรมหลายช่อง ทำให้สามารถทำการทดลองครั้งนี้โดยใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวได้

ในการทดสอบผู้ทำวิจัยได้สร้างตำแหน่งของเรดาร์ 3 แบบและในแต่ละแบบได้สร้างเส้นทางการบินของอากาศยานแตกต่างกันออกไปทั้งชนิดของอากาศยานและทิศทาง การสร้างเส้นทางการบินจะผู้วิจัยได้สร้างเส้นทางการบินในเรดาร์แต่ละแบบ 10 ครั้งที่มีรูปแตกต่างกันออกไป รวมเส้นทางการบินทั้งหมดเป็น 30 ครั้ง เมื่อได้เส้นทางการบินแล้วจึงบันทึกเป็นแฟ้มข้อมูลของเส้นทางการบิน จากเส้นทางการบินที่สร้างไว้นั้นได้นำมาสร้างเป็นโปรโตคอลของเรดาร์ โดยทำการจำลองข้อมูลตามโปรโตคอลของเรดาร์ ทำให้สามารถควบคุมเส้นทางการบินและชนิดของอากาศยานดังนั้นการคำนวณหาค่าความถูกต้องของระบบจึงสามารถกระทำได้ ซึ่งไม่สามารถกระทำได้ในการใช้ข้อมูลจริง เมื่อสร้างข้อมูลเรียบร้อยแล้วได้ทำการส่งข้อมูลทำการส่งผ่านช่องรับส่งสัญญาณ โดยให้โปรแกรมย่อยทำการติดต่อกันโดยผ่านทางช่องรับส่งสัญญาณ การส่งสัญญาณสามารถทำได้ในเครื่องคอมพิวเตอร์เครื่องเดียวกันหรือต่างเครื่องกันก็สามารถทำได้ ในกรณีที่ต้องการส่งข้อมูลหรือต่างเครื่องกันก็สามารถทำได้ โดยการเปลี่ยนชื่อของเครื่องคอมพิวเตอร์ให้เป็นชื่อของเครื่องที่เราต้องการจะส่งข้อมูลเท่านั้น ข้อมูลก็จะถูกส่งไปยังเครื่องที่เราต้องการ

จากการทดสอบการทำงานเราสามารถคำนวณความถูกต้องของระบบ โดยการทดลองทั้งหมด 30 ครั้ง จากผลการทดสอบรวมของการทำการทดสอบทั้งหมดแสดงได้ในตารางที่ 6.1 โดยมีข้อมูลของผลของการวิจัยดังนี้คือ

1. เรดาร์หมุนทั้งหมด 5,714 รอบ
2. สร้างเส้นทางการบินของอากาศยาน 1,240 เส้นทาง
3. อากาศยานสะสมที่ผ่านระบบทั้งหมด 114,035 เครื่อง
4. อากาศยานในระบบเดิมผิดพลาดสะสมทั้งหมด 35,514 เครื่อง
5. อากาศยานในระบบผ่านการกรองและขจัดความซ้ำซ้อนผิดพลาดสะสมทั้งหมด 2,307 เครื่อง

สรุปผลของการทำการวิจัยความถูกต้องคิดเป็นเปอร์เซ็นต์ 97.98 % แต่ถ้าเป็นในระบบเดิมความถูกต้องคิดเป็นเปอร์เซ็นต์เพียงแค่ 68.86% ผลของการทดลองครั้งที่ 2 ถึง 5 จะอยู่ในภาคผนวก ง. ตารางที่ ง-2 ถึงตารางที่ ง-5

ตารางที่ 6.1 ผลสรุปของการทดลอง

ครั้งที่	จำนวนรอบ	จำนวนอากาศยาน	อากาศยานสะสม	ผิดพลาดสะสม (ระบบเก่า)	ความถูกต้องระบบเก่า(%)	ผิดพลาดสะสม (ระบบใหม่)	ความถูกต้องระบบใหม่(%)
1	317	9	1,796	920	48.78	13	99.28
2	221	15	2,455	1,033	57.92	43	98.25
3	241	31	4,788	1,792	62.57	98	97.95
4	242	44	4,788	1,792	62.57	98	97.95
5	291	58	8,978	3,550	60.46	178	98.02
6	242	4	4,788	1,806	62.28	96	97.97
7	151	13	1,092	257	76.47	31	97.16
8	150	18	1,384	301	78.25	28	97.98
9	242	27	4,787	1,806	62.27	95	98.02
10	151	48	3,988	827	79.26	73	98.17
11	119	8	680	211	68.97	9	98.68
12	202	19	2,219	698	68.54	16	99.28
13	228	47	4,538	1,230	72.90	83	98.17
14	228	57	5,731	1,641	71.37	115	97.99
15	228	76	7,758	1,882	75.74	134	98.27
16	84	64	3,056	843	72.41	57	98.13

ครั้งที่	จำนวนรอบ	จำนวนอากาศยาน	อากาศยานสะสม	ผิดพลาดสะสม (ระบบเก่า)	ความถูกต้องระบบเก่า(%)	ผิดพลาดสะสม (ระบบใหม่)	ความถูกต้องระบบใหม่(%)
17	84	74	3,597	1,056	70.64	58	98.39
18	103	82	4,232	1,105	73.89	121	97.14
19	109	92	5,134	1,324	74.21	192	96.26
20	220	52	634	121	80.91	13	97.95
21	268	14	2,006	465	76.82	27	98.65
22	184	31	3,367	1,609	52.21	63	98.13
23	185	43	4,406	1,301	70.47	81	98.16
24	185	55	4,932	1,502	69.55	101	97.95
25	185	63	5,499	1,649	70.01	112	97.96
26	185	69	5,897	1,752	70.29	141	97.61
27	185	73	6,133	1,805	70.57	125	97.96
28	186	16	1,782	444	75.08	29	98.37
29	149	18	1,698	379	77.68	40	97.63
30	149	20	1,892	413	78.17	37	98.04
รวม	5,714	1,240	114,035	35,514	68.86	2,307	97.98

6.2 ข้อจำกัดของระบบ

1. ข้อจำกัดของเรดาร์นั้นเกิดสาเหตุ 2 ประการคือ

   1. เมื่ออากาศยานบินข้ามจากเส้นสิ้นสุดของสัญญาณการหมุน มายังเส้นเริ่มต้นของสัญญาณการหมุนของเรดาร์นั้น จะเกิดการคลาดเคลื่อนในการตรวจจับอากาศยานของเรดาร์ได้

   2. เมื่อทำการสร้างเส้นทางการบินที่ใกล้เคียงกับจุดศูนย์กลางของเรดาร์เกินไปจะทำให้การสร้างข้อมูลของเรดาร์คลาดเคลื่อนได้ และการขจัดความซ้ำซ้อนจะไดรับผลกระทบโดยตรงจากการที่ข้อมูลของเรดาร์คลาดเคลื่อน

2. ในการจำลองเส้นทางของอากาศยาน ถ้าจุดเริ่มต้นของอากาศยานอยู่ในรัศมีของเรดาร์ใด ๆ ระบบจะตรวจพบตำแหน่งอากาศยานดังทำให้มีการนับจำนวนอากาศยาน แต่ข้อมูลของโปรโตคอลยังไม่มีข้อมูลของทิศทางและความเร็วดังนั้นระบบจำลองเส้นทางของ

อากาศยานจึงยังไม่นับอากาศยานเครื่องนั้น และในทำนองเดียวกัน ถ้าอากาศยานสิ้นสุดภายในรัศมีของเรดาร์ระบบจำลองเส้นทางของอากาศยานจึงยังไม่นับอากาศยานเครื่องนั้น ดังนั้นการคำนวณจำนวนของอากาศยานจึงเกิดข้อผิดพลาดได้

3. การขจัดความซ้ำซ้อนนั้นสามารถทำได้เมื่อเรดาร์มีพื้นที่ร่วมไม่เกิน 2 เครื่อง

6.3 ประโยชน์ที่คาดว่าจะได้รับ

1. โปรแกรมต้นแบบเพื่อการขจัดความซ้ำซ้อนและกรองข้อมูลของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง
2. เป็นโปรแกรมต้นแบบและแนวทางในการพัฒนาระบบขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง และเป็นต้นแบบในการพัฒนาระบบขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องที่กวาดในรูป 3 มิติต่อไป
3. ข้อมูลที่ได้ผ่านการกรองและขจัดความซ้ำซ้อนของอากาศยานแล้วสามารถนำไปใช้เป็นข้อมูลของระบบแจ้งเตือนภัยทางอากาศ
4. เป็นเครื่องมือในการสร้างเส้นทางการบินจำลองของอากาศยาน

6.4 ข้อเสนอแนะ

การทำการพัฒนาระบบโปรแกรมต้นแบบเพื่อขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่องได้ทำการพัฒนาจากข้อมูลจำลอง เพื่อจะได้นำไปใช้งานจริงต้องมีการแก้ไขดังต่อไปนี้

1. เปลี่ยนรับข้อมูลโปรโตคอลโดยการอ่านข้อมูลจากแฟ้มข้อมูลมาเป็นการอ่านข้อมูลจากช่องอนุกรม (Serial Port) การแก้ไขต้องทำดังต่อไปนี้ดังต่อไปนี้
   1.1 โปรแกรมnewradar.javaในส่วนของopenfileให้การอ่านข้อมูลจากช่องอนุกรม
   2.1 โปรแกรมoldradar.javaในส่วนของopenfileให้การอ่านข้อมูลจากช่องอนุกรม
   3.1 โปรแกรมsendnewdata.javaในส่วนของopenfileให้การอ่านข้อมูลจากช่องอนุกรม
   4.1 โปรแกรมsendolddata.javaในส่วนของopenfileให้การอ่านข้อมูลจากช่องอนุกรม
2. เปลี่ยนการส่งข้อมูลโปรโตคอลเพื่อไปยังระบบแจ้งเตือนภัย ต้องเปลี่อนชื่อของเครื่องคอมพิวเตอร์ในโปรแกรมให้เป็นเครื่องคอมพิวเตอร์ที่เราต้องการ โดยการเปลี่ยนโปรแกรม syncronizedata.java ในส่วนของ WarningSendData ให้เป็นเครื่องคอมพิวเตอร์ที่เราต้องการ

3. ในการนำไปใช้งานจริงนั้นเรดาร์แต่ละเครื่องอาจจะหมุนใช้เวลาไม่เท่ากัน ดังนั้นควรใช้เวลาเฉลี่ยของเรดาร์ทั้งหมดแทน 4 วินาที
4. ในการนำไปใช้งานจริงนั้นเรดาร์แต่ละเครื่องอาจเริ่มต้นการหมุนในทิศที่ต่างกัน ดังนั้นควรหามุมการหมุนเริ่มต้นของเรดาร์ แล้วคำนวณหาเวลาที่แตกต่างกันของเรดาร์แต่ละเรดาร์เพื่อทำการปรับให้เรดาร์เริ่มต้นในทิศทางดียวกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1 Merrill I. Skolnik. Introduction to radar systems. Second Edition. Singapore : McGRAW-HILL BOOK COMPANY, 1981.

2 วิภา รุ่งดิลกโรจน์. เรดาร์ตรวจอากาศและการวิเคราะห์ภาพสัญญาณ. กรุงเทพมหานคร : ฝ่ายวิจัยอุตุนิยมวิทยา กองการศึกษาและวิจัย เอกสารวิชาการเลขที่ 551.501.81.-01-2535, 2534.

3 M.A. Munem and D.J. Foulis. Calculus with Analytic Geometry. New York, USA: Worth Publishers, Inc, 1988.

4 D.L.Mills. RFC956 Algorithms for synchronizing network clocks. USA: NEWWORK WORKING GROUP REQUESTS FOR COMMENTS: 956, 1985.

5 Kautzsch. Technical Specification for Remote Situation Display. Germany: Document of radar DR172 ADV protocol material number 10272835, 1997.

6 Harvey Deitel and Pual Deitel. JAVA Multimedia Cyber Classroom. Second Edition. Jercy, USA: Prentice Hall PTR, 1998.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โครงสร้างตารางข้อมูล

ตารางที่ ก-1 ตำแหน่งของอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศไทย

ตำแหน่งของอักษร	ความหมายของข้อมูล
1	องศาของเส้นแวง
2	องศาของเส้นรุ้ง
3	องศาของเส้นแวง
4	องศาของเส้นรุ้ง
5	ลิปดาของเส้นแวง
6	ลิปดาของเส้นแวง
7	ลิปดาของเส้นรุ้ง
8	ลิปดาของเส้นรุ้ง

ตารางที่ ก-2 ตำแหน่งของอักษรและตำแหน่งทางภูมิศาสตร์ของประเทศใกล้เคียง

ตำแหน่งของอักษร	ความหมายของข้อมูล
1-6	ตำแหน่งทางภูมิศาสตร์ของเส้นแวง
7	ตัวแบ่งขอบเขตของเส้นรุ้งและเส้นแวง
8-14	ตำแหน่งทางภูมิศาสตร์ของเส้นรุ้ง

ตารางที่ ก-3 แสดงการเก็บข้อมูลของตำแหน่งเรดาร์

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	จำนวนเรดาร์
2	เส้นแวง (หน่วยเป็นพิลิปดา) เรดาร์ 1
3	เส้นรุ้ง (หน่วยเป็นพิลิปดา) เรดาร์ 1
.	.
n	เส้นแวง (หน่วยเป็นพิลิปดา) เรดาร์ n
n	เส้นรุ้ง (หน่วยเป็นพิลิปดา) เรดาร์ n

ตารางที่ ก-4 โครงสร้างข้อมูลของอากาศยาน

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	หมายเลขของเรดาร์
2	ตำแหน่งที่ของเส้นทางการบิน
3	ชนิดของอากาศยาน
4	ความเร็วของอากาศยาน
5	ตำแหน่งทางภูมิศาสตร์ของเส้นแวง(หน่วยเป็นพิลิปดา)
6	ตำแหน่งทางภูมิศาสตร์ของเส้นรุ้ง(หน่วยเป็นพิลิปดา)
7	ฝ่ายของอากาศยาน
8	เครื่องหมายของอากาศยาน

ตารางที่ ก-5 โครงสร้างข้อมูลของอากาศยานที่ใช้ในการเรียงลำดับ

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	หมายเลขของเรดาร์
2	รอบที่ของเรดาร์
3	ค่ามุมของอากาศยาน
4	ตำแหน่งทางภูมิศาสตร์ของเส้นแวง(หน่วยเป็นพิลิปดา)
5	ตำแหน่งทางภูมิศาสตร์ของเส้นรุ้ง(หน่วยเป็นพิลิปดา)
6	ความเร็วทางแกนเส้นแวง(1/8หน่วยเป็นพิลิปดา)
7	ความเร็วทางแกนเส้นรุ้ง(1/8หน่วยเป็นพิลิปดา)

ตารางที่ ก-6 โครงสร้างข้อมูลรายละเอียดของอากาศยาน

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	หมายเลขของอากาศยาน
2	ชนิดของอากาศยาน
3	ฝ่ายของอากาศยาน
4	เครื่องหมายของอากาศยาน

ตารางที่ ก-7 โครงสร้างข้อมูลของอากาศยานเพื่อใช้กำหนดหมายเลขอากาศยาน

ขอบเขตข้อมูลที่	ชนิดของข้อมูล	ความหมายของข้อมูล
1	ตัวเลข	หมายเลขของเรดาร์
2	ตัวเลข	ตำแหน่งที่ของเส้นทางการบิน
3	ตัวเลข	เวลาของเรดาร์
4	ตัวเลข	ตำแหน่งทางภูมิศาสตร์ของเส้นแวง(หน่วยเป็นพิลิปดา)
5	ตัวเลข	ตำแหน่งทางภูมิศาสตร์ของเส้นรุ้ง(หน่วยเป็นพิลิปดา)
6	ตัวเลข	ความเร็วทางแกนเส้นแวง(1/8หน่วยเป็นพิลิปดา)
7	ตัวเลข	ความเร็วทางแกนเส้นรุ้ง(1/8หน่วยเป็นพิลิปดา)

ตารางที่ ก-8 โครงสร้างข้อมูลโปรโตคอลของอากาศยาน

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	จำนวนของข้อมูล แสดงจำนวนอากาศยานทั้งหมดหลังจากการขจัดความซ้ำซ้อน
2	ตัวเลขอ้างอิงเส้นทางการบินที่ของอากาศยานที่ตรวจพบ
3	ตำแหน่งพิกัดเส้นรุ้ง มีหน่วยเป็น1/8ของไมล์ทะเล
4	ตำแหน่งพิกัดเส้นแวง มีหน่วยเป็น1/8ของไมล์ทะเล
5	ความเร็วทางแกนX มีหน่วยเป็น1/8ของไมล์ทะเลต่อชั่วโมง
6	ความเร็วทางแกนY มีหน่วยเป็น1/8ของไมล์ทะเลต่อชั่วโมง
7	ลำดับของเส้นทางการบินโดยจะมีตัวเลขตั้งแต่ 001 ถึง 999
8	หลักฐานของเส้นทางการบิน(Track Identification)
9	เวลาของระบบ(System Time)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ก-9 ข้อมูลสถานที่ตั้งของเรดาร์

เขตข้อมูล	ข้อความภายใน(content)	ความหมาย
1	STX	ส่วนหัวของสัญญาณ
2	STX	ส่วนหัวของสัญญาณ
3	Block length	ความยาวของสัญญาณ
4	Block length	ความยาวของสัญญาณ
6	Number of symbols	หมายเลขของเรดาร์
7	X-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นรุ้ง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
8	X-position (LSB)	
9	Y-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นแวง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
10	Y-position (LSB)	

ตารางที่ ก-10 ข้อมูลของอากาศยาน

เขตข้อมูล	ข้อความภายใน(content)	ความหมาย
1	STX	ส่วนหัวของสัญญาณ
2	STX	ส่วนหัวของสัญญาณ
3	Block length	ความยาวของสัญญาณ
4	Block length	ความยาวของสัญญาณ
5	n-time the data block of a track, maximum 50 tracks message – identification	จำนวนของอากาศยาน
6	Track reference number	ตัวเลขอ้างอิงเส้นทางการบิน
7	X-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นรุ้ง มีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
8	X-position (LSB)	

เขตข้อมูล	ข้อความภายใน(content)	ความหมาย
9	Y-position (MSB)	ตำแหน่งของเรดาร์ตามเส้นแวงมีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
10	Y-position (LSB)	
11	X-speed (MSB)	ความเร็วของอากาศยานเส้นรุ้งมีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
12	X-speed (LSB)	
13	Y-speed (MSB)	ความเร็วของอากาศยานเส้นแวงมีหน่วยเป็น 1 ใน 8 ของไมล์ทะเล
14	Y-speed (LSB)	
15	Track number (MSB)	ตัวเลขของเส้นทางการบิน
16	Track number (LSB)	
17	System Time	เวลาของระบบ
18	System Time	
19	System Time	
20	System Time	
21	System Time	
22	System Time	

ตารางที่ ก-11 โครงสร้างข้อมูลของจำนวนอากาศยานต่อการรอบของเรดาร์

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	รอบที่หมุนของเรดาร์
2	จำนวนอากาศยาน

ตารางที่ ก-12 โครงสร้างข้อมูลความถูกต้องของระบบ

ขอบเขตข้อมูลที่	ความหมายของข้อมูล
1	รอบที่หมุนของเรดาร์
2	จำนวนอากาศยานของเรดาร์ที่ 1
3	จำนวนอากาศยานของเรดาร์ที่ 2
4	จำนวนอากาศยานของเรดาร์ที่ 3
5	จำนวนอากาศยานของเรดาร์ที่ 4
6	จำนวนอากาศยานรวมของเรดาร์ทั้ง 4 เครื่องของระบบเดิม
7	จำนวนอากาศยานรวมของเรดาร์ระบบใหม่
8	จำนวนอากาศยานรวมของเรดาร์ที่เป็นจริง
9	จำนวนอากาศยานที่ผิดพลาดต่อรอบ
10	จำนวนอากาศยานที่ผิดพลาดสะสม
11	จำนวนอากาศยานสะสม
12	ความถูกต้องของระบบคิดเป็นเปอร์เซ็นต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การใช้งานโปรแกรม

ก่อนที่จะใช้งานระบบการพัฒนาโปรแกรมต้นแบบเพื่อกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง ผู้ควบคุมระบบจะต้องมีการติดตั้งส่วนประกอบโปรแกรมต่างๆ เพื่อจะทำให้โปรแกรมสามารถทำงานผ่านระบบได้

สิ่งที่ต้องเตรียมก่อนการใช้งานคือ การจัดหาโปรแกรมไมโครซอฟท์วิชวลเบสิก (Microsoft Visual BASIC) เป็นเครื่องมือในการพัฒนา โดยใช้เป็นโปรแกรมในการสร้างหน้าจอเมนหลักในการนำระบบต่างๆมาไว้ยังที่เดียวกัน เมื่อมีการเลือกเมนูจากโปรแกรมหลักก็จะทำการเรียกโปรแกรมที่ทำการเลือกมาดำเนินงาน โปรแกรมไมโครซอฟท์แอกเซส (Microsoft Access) เพื่อใช้เป็นโปรแกรมจัดเก็บข้อมูลเพื่อใช้ในการดำเนินงาน โปรแกรมเจ บิวต์เดอร์ (J Builder) เวอร์ชัน 2.0x เพื่อใช้เป็นโปรแแกรมขจัดความซ้ำซ้อนของข้อมูลและเพื่อใช้ส่งข้อมูลที่ผ่านการคำนวณแล้วไปยังโปรแกรมหลักเพื่อใช้ในการประมวลผลต่อไป

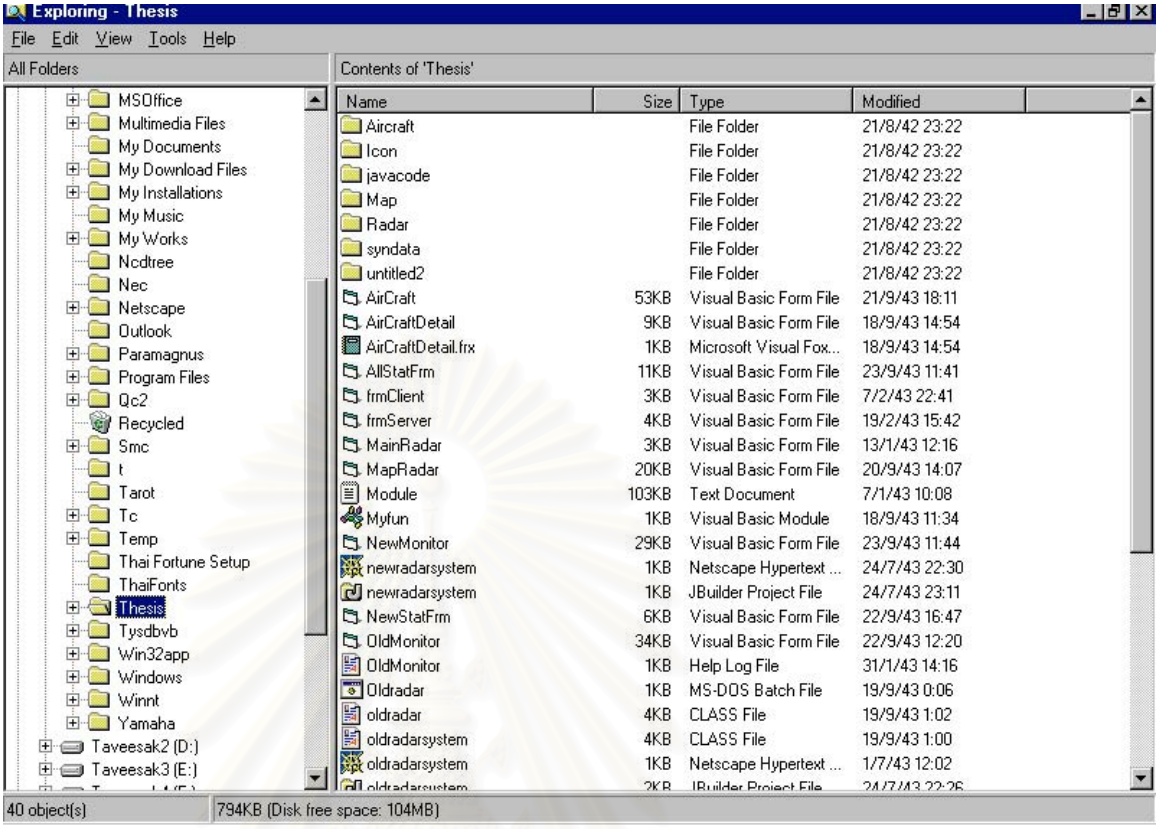
สิ่งที่ต้องเตรียมก่อนใช้งานโปรแกรม

ระบบที่จะใช้กับโปรแกรมควรเป็นดังนี้

1. ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 98 (Windows 98) ซึ่งมีซีพียูเพนเทียมสาม 500 เมกะเฮิร์ต (Pentium III 500 MHz) และ ตัวปรับเครือข่าย(Network Adapter)
2. โปรแกรมไมโครซอฟท์วิชวลเบสิก เวอร์ชัน 5 ขึ้นไป
3. โปรแกรมเจ บิวต์เดอร์ เวอร์ชัน 2.0 ขึ้นไป
4. โปรแกรมไมโครซอฟท์แอกเซส เพื่อใช้จัดเก็บข้อมูลเพื่อใช้ในการประมวลผล

การติดตั้งโปรแกรมระบบการพัฒนาโปรแกรมต้นแบบเพื่อกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

การติดตั้งโปรแกรมนั้นเนื่องจากการใช้งานจะต้องมีการเตรียมการติดตั้งระบบต่างๆ ไว้ในไดเรกทอรี่ c:\thesis เนื่องจากโปรแกรมมีการเรียกใช้ยังไดเรกทอรี่ดังกล่าว นอกจากนั้นยังต้องติดตั้งโปรแกรมวิชวลเบสิก โปรแกรมไมโครซอฟท์แอกเซส และโปรแกรมเจบิวต์เดอร์ เพื่อใช้ประมวลผล



รูปที่ ข-1 แสดงการติดตั้งโปรแกรมยังไดเรกทอรี่ c:\thesis

การใช้งานโปรแกรม

จากระบบ ประกอบด้วยการทำงาน 3 ส่วนหลักๆ ดังนี้คือ

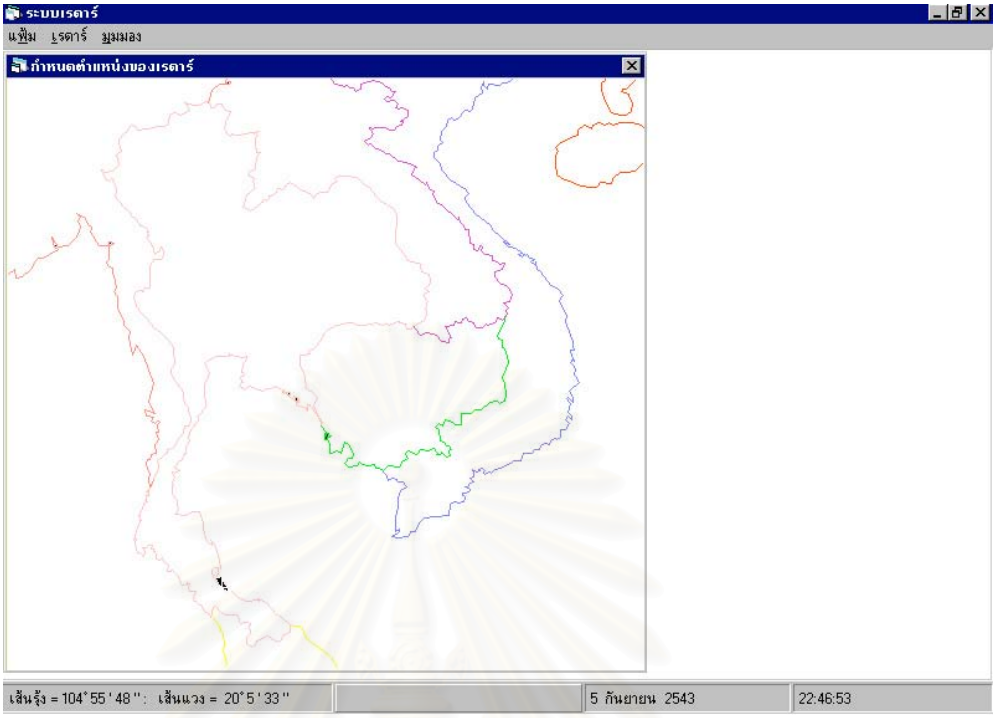
เรดาร์ เพื่อใช้กำหนดตำแหน่งเรดาร์

เส้นทางการบิน เพื่อใช้กำหนดเส้นทางการบิน

ส่งข้อมูล เพื่อใช้ส่งข้อมูลและแสดงผล

1. **เรดาร์**

ประกอบด้วยเมนูย่อยคือ กำหนดตำแหน่งของเรดาร์ หลังจากที่เลือกการทำงานในส่วนนี้ โปรแกรมจะแสดงผลของดังรูปที่ ข-2 ลักษณะการทำงานในส่วนนี้ใช้กำหนดตำแหน่งต่างๆ ในแผนที่



รูปที่ ข-2 แสดงผลหลังจากเลือกเมนูเรดาร์

ทางด้านล่างของหน้าจอนี้จะมีข้อมูลเกี่ยวกับพิกัดของแผนที่ตามที่ Pointer ได้ชี้ไปยังตำแหน่งนั้น เป็นเส้นรุ้งและเส้นแวง , วันและเวลา โดยการทำงานในแต่ละส่วนสามารถอธิบายได้ดังนี้ จากรูปประกอบด้วยการทำงาน 3 ส่วนคือ

1.1 แฟ้ม ประกอบด้วย

1.1.1 เปิด ใช้สำหรับเปิดข้อมูลที่ได้มีการบันทึกไว้จากแฟ้มข้อมูล

1.1.2 ปิด ใช้สำหรับปิดแผนที่ที่ได้ทำงานอยู่ โดยไม่มีการบันทึก

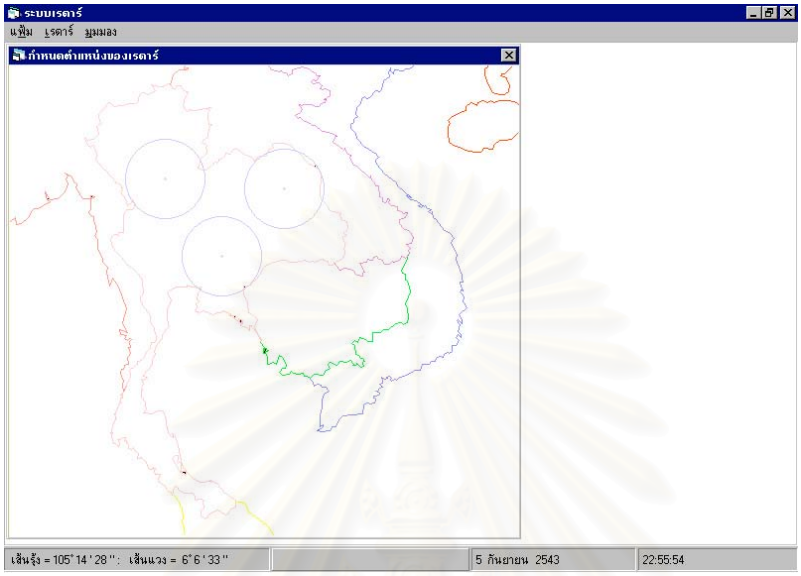
1.1.3 บันทึก ใช้ในกรณีที่ต้องการบันทึกข้อมูลต่างๆ ที่มีอยู่บนแผนที่ลงแฟ้มข้อมูล

1.1.4 บันทึกเป็น ใช้ในกรณีที่ต้องการบันทึกข้อมูลต่างๆ ลงเป็นอีกชื่อหนึ่ง ในกรณีที่ได้มีการตั้งชื่อไว้เรียบร้อยแล้ว

1.1.5 กลับไปโปรแกรมหลัก เพื่อกลับเข้าไปในหน้าจอของโปรแกรมหลักเพื่อทำงานในส่วนอื่นถัดไป

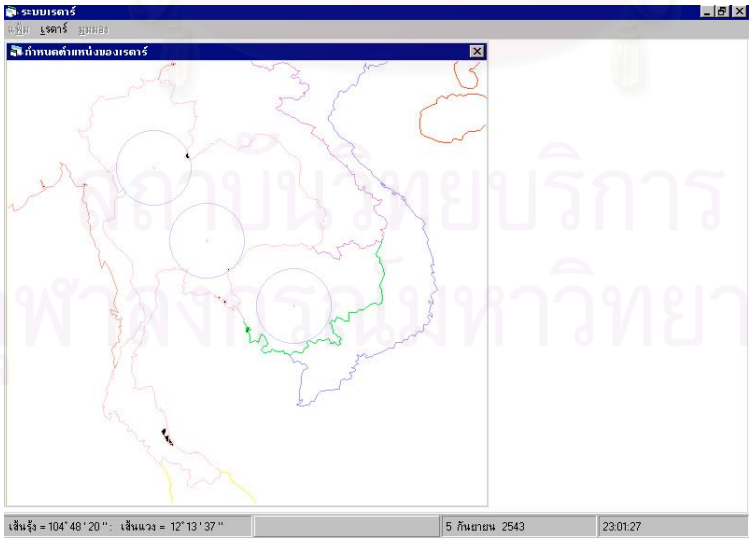
1.2 เรดาร์

1.2.1 สร้างเรดาร์ เพื่อใช้กำหนดพิกัดของเรดาร์บนแผนที่ให้มีจุดศูนย์กลางอยู่ ณ ตำแหน่งใด ลักษณะดังรูป ข-3



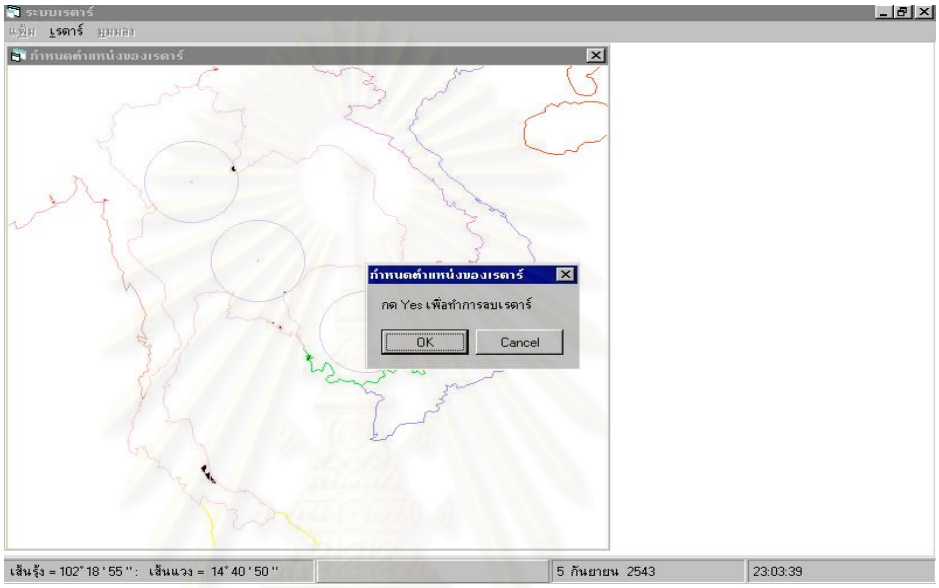
รูปที่ ข-3 แสดงลักษณะหน้าจอหลังจากเลือกเมนูสร้างเรดาร์

1.2.2 ย้ายเรดาร์ เพื่อใช้ย้ายพิกัดของตำแหน่งเรดาร์ไปไว้ในพิกัดใหม่ โดยสามารถใช้ Mouse เลือกไปยังบริเวณรัศมีของเรดาร์หลังจากนั้นกด Mouseค้างลากMouse ไปไว้ยังพิกัดที่ต้องการ ดังรูปที่ ข-4



รูปที่ ข-4 แสดงลักษณะหน้าจอของเมนูย้ายเรดาร์

1.2.3 ลบเรดาร์ เพื่อใช้ในกรณีที่ต้องการลบตำแหน่งเรดาร์ ณ พิกัดนั้นๆ ออกไป ลักษณะการใช้งานโดยเลือกเมนูลบเรดาร์ หลังจากนั้นให้นำ Mouse ไปเลือก ณ บริเวณใดบริเวณหนึ่งในรัศมีของจุดศูนย์กลางแล้วกด Mouse หลังจากนั้นจะขึ้นข้อความเพื่อยืนยันการลบข้อมูล ดังรูป ข-5ในกรณีที่ต้องการยืนยันการลบให้เลือก OK แต่ถ้ายังไม่ต้องการยืนยันการลบข้อมูลให้กด Cancel ลักษณะดังรูป



รูปที่ ข-5 แสดงลักษณะหน้าจอของเมนูลบเรดาร์

1.3 มุมมอง เพื่อใช้ขยายภาพของแผนที่ตามมุมมองที่ต้องการ มีให้เลือกดังนี้

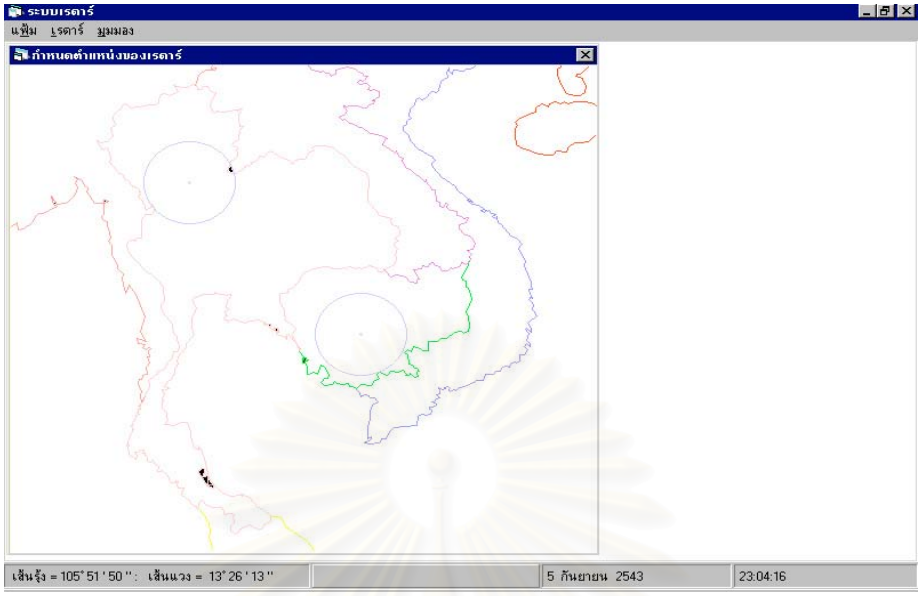
1.3.1 100 %

1.3.2 200 %

1.3.3 400 %

1.3.4 800 %

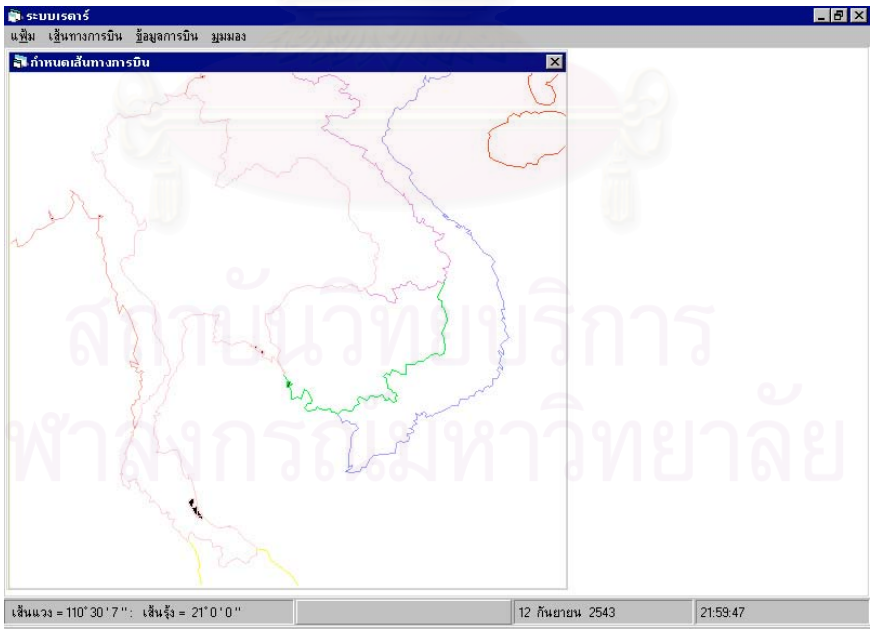
จากรูปที่ ข-6 เป็นมุมมองที่ 100 % แต่ในกรณีที่เลือกข้ออื่นๆ นั้นเนื่องจากหน้าจอคอมพิวเตอร์ให้เป็นมุมมองที่ 100 % แล้ว ดังนั้นการขยายภาพจึงต้องกำหนดพิกัดที่สนใจ ในกรณีที่พิกัดที่สนใจอยู่บริเวณกึ่งกลาง สามารถดึงข้อมูลแผนที่โดยยึดจุดที่สนใจที่ได้เลือกเป็นจุดศูนย์กลางใหม่ของแผนที่ แต่หากจุดที่สนใจอยู่บริเวณริมแผนที่ จะนำภาพของขอบแผนที่เป็นขอบริมของจุดสนใจใหม่



รูปที่ ข-6 แสดงลักษณะหน้าจอของการขยายภาพตามมุมมองที่ต้องการ

2. **เส้นทางการบิน**

เป็นส่วนที่ใช้กำหนดเส้นทางการบิน ประกอบด้วยเมนูย่อยคือ กำหนดเส้นทางการบิน หลังจากที่ได้เลือกการทำงานในส่วนนี้ โปรแกรมจะแสดงผล ดังรูปที่ ข-7 ดังนี้คือ



รูปที่ ข-7 แสดงลักษณะหน้าจอของเมนูเส้นทางการบิน

ในส่วนของการกำหนดเส้นทางการบินจะประกอบด้วย

2.1 แฟ้ม

2.1.1 เปิดแฟ้มเรดาร์ สำหรับค้นหาแฟ้มที่ได้มีการบันทึกไว้ก่อนหน้านี้

2.1.2 เปิดแฟ้มเส้นทางการบิน สำหรับค้นหาแฟ้มเส้นทางการบินที่ได้มีการบันทึกไว้ก่อนหน้านี้

2.1.3 ปิดแฟ้มเส้นทางการบิน สำหรับปิดแฟ้มเส้นทางการบินที่คุณต้องการ

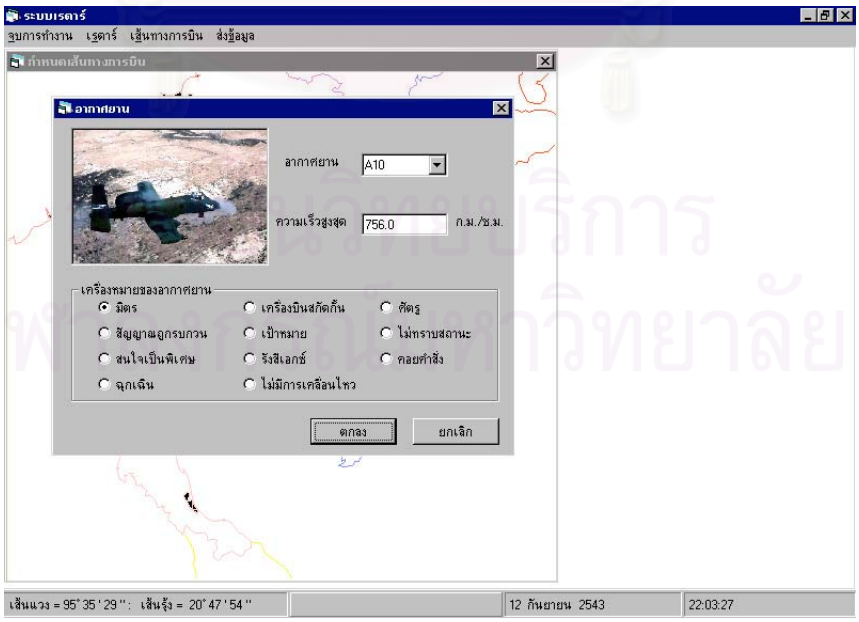
2.1.4 บันทึก ใช้สำหรับบันทึกรายละเอียดของการเปลี่ยนแปลงลงแฟ้มเส้นทางการบินลงชื่อแฟ้มเดิม

2.1.5 บันทึกเป็น ใช้สำหรับบันทึกรายละเอียดของการเปลี่ยนแปลงลงแฟ้มเส้นทางการบินเป็นชื่อใหม่

2.1.6 กลับไปโปรแกรมหลัก

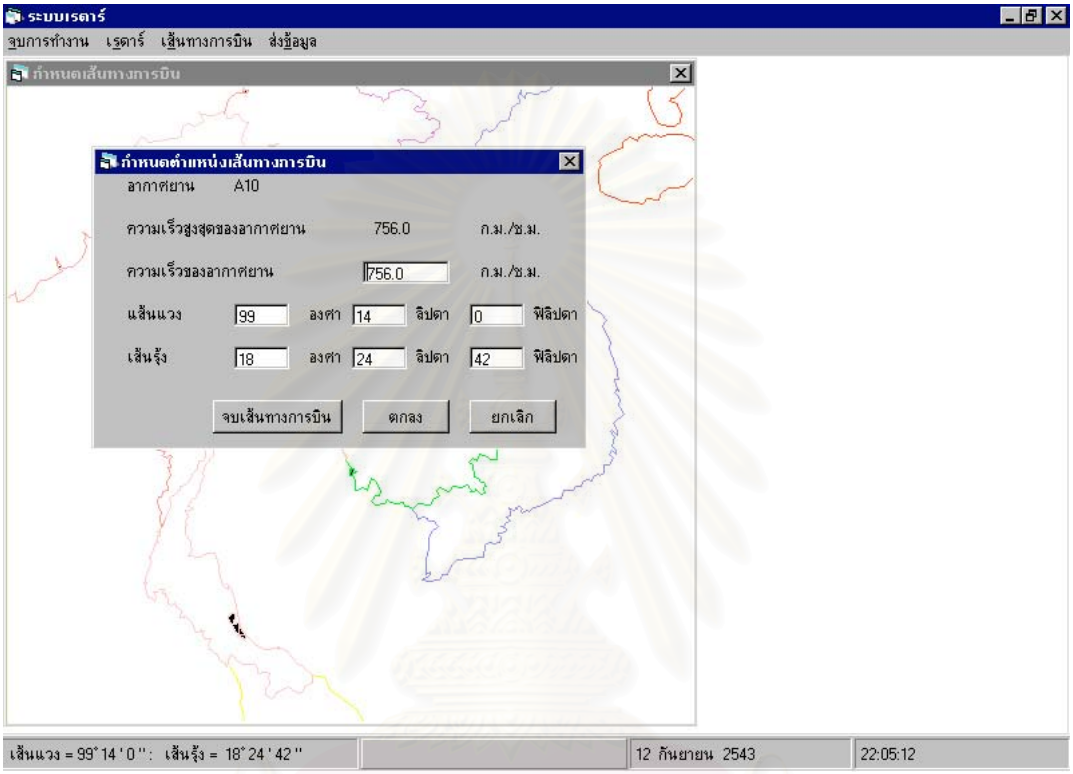
2.2 เส้นทางการบิน

2.2.1 สร้างเส้นทางการบิน เพื่อใช้สร้างเส้นทางการบิน หลังจากที่เลือกการทำงานในส่วนนี้แล้ว โปรแกรมจะแสดงผลโดยให้เลือกลักษณะของอากาศยาน ความเร็วที่ใช้ และเครื่องหมายของอากาศยาน มีลักษณะดังรูปที่ ข-8



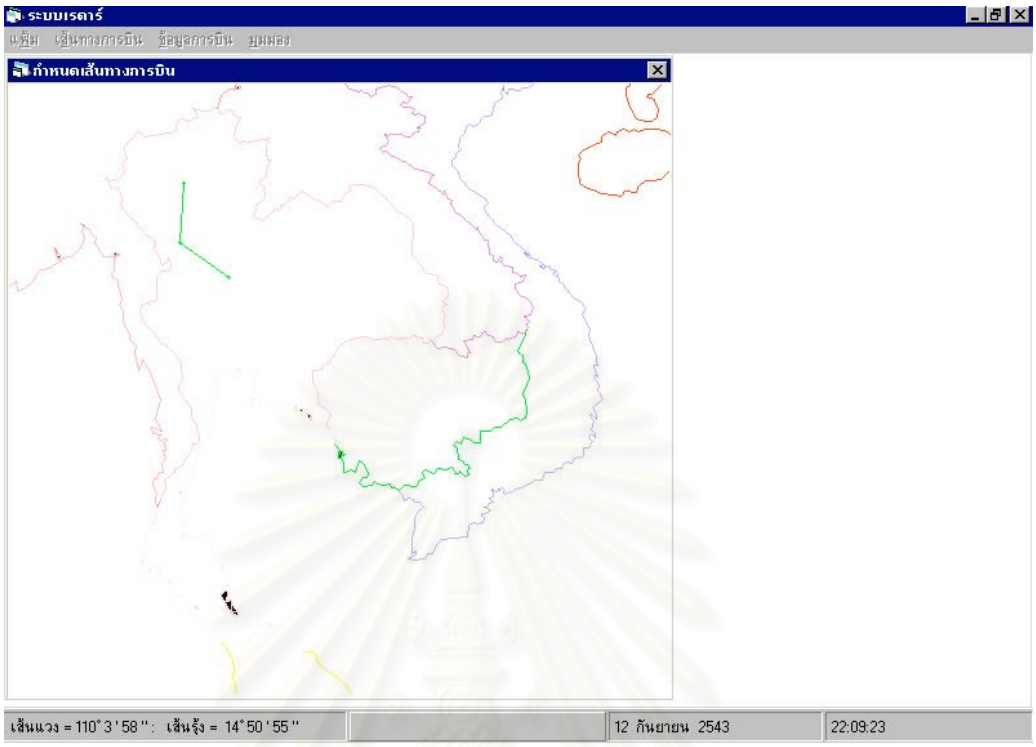
รูปที่ ข-8 แสดงลักษณะหน้าจอของเมนูสร้างเส้นทางการบินเพื่อเลือกอากาศยานที่ต้องการ

หลังจากที่เลือกข้อมูลในส่วนนี้ได้แล้ว โปรแกรมจะให้กำหนดเส้นทางการบิน โดยผู้ใช้สามารถนำเมาส์ไปเลือก ณ จุดใดๆ ในแผนที่ที่ต้องการ หลังจากเลือกโดยการคลิกเมาส์แล้ว โปรแกรมจะขึ้นมูลในส่วนถัดไป เพื่อให้ผู้ใช้สามารถกำหนดพิกัดของเส้นทางการบินดังกล่าว ลักษณะดังรูปที่ ข-9



รูปที่ ข-9 แสดงลักษณะหน้าจอของเมนูสร้างเส้นทางการบินของการกำหนดเส้นทางการบิน

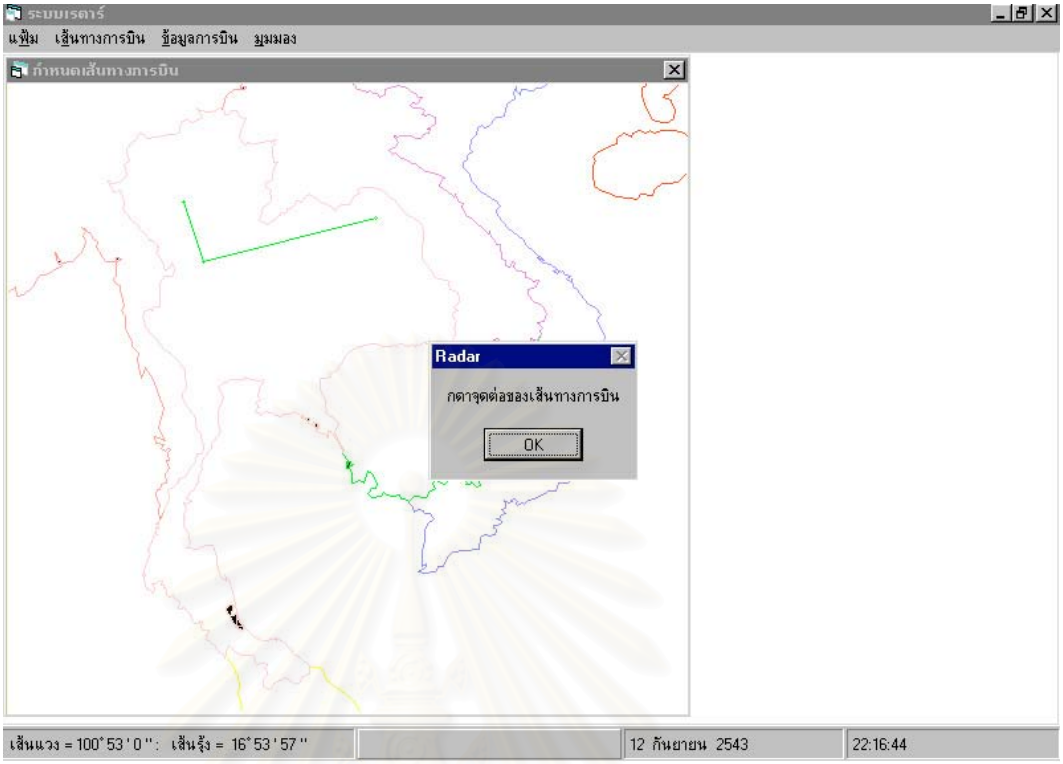
หลังจากที่กำหนดเส้นทางการบินได้แล้ว ในกรณีที่ต้องการจบเส้นทางการบิน จากรูปที่ ข-9 ให้เลือก จบเส้นทางการบิน หากผู้ใช้ต้องการกำหนดเส้นทางการบินต่อไปให้เลือก ตกลง หรือ ผู้ใช้ต้องการยกเลิกเส้นทางการบินดังกล่าวให้เลือก ยกเลิก เส้นทางการบินที่ได้จะมีลักษณะดังรูปที่ ข-10



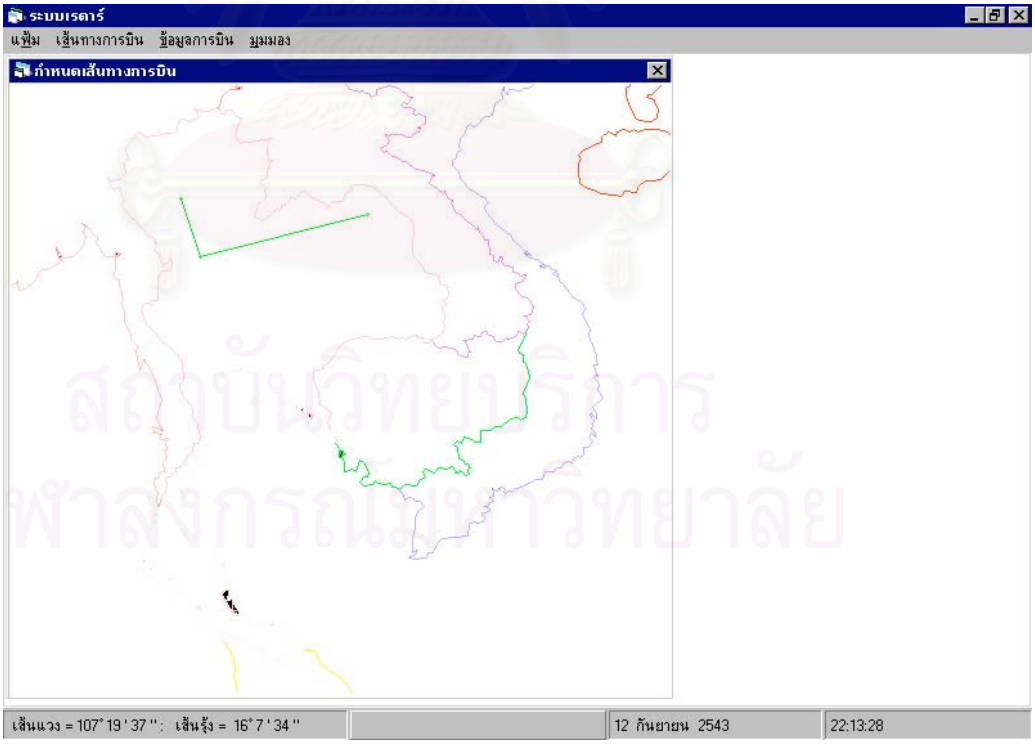
รูปที่ ข-10 แสดงผลลัพธ์ที่ได้หลังจากสร้างเส้นทางการบินเรียบร้อยแล้ว

จากรูปที่ ข-10 เส้นเขียวที่ได้คือเส้นทางการบินที่ได้สร้างขึ้นจากพิกัดที่เลือกไว้

2.3 ย้ายเส้นทางการบิน เพื่อใช้ในกรณีที่ต้องการย้ายเส้นทางการบินที่ได้สร้างขึ้น โดยลักษณะการทำการหลังจากเลือกเมนูในข้อนี้ ผู้ใช้จะต้องนำเมาส์ไปเลือกยังบริเวณที่เป็นจุดพิกัดที่ได้สร้างไว้ตั้งแต่แรกลักษณะของการย้ายทำได้โดย กดเมาส์ค้างไว้ แล้วเลื่อนจุดพิกัดใหม่ไปยังบริเวณที่ต้องการ ในกรณีที่ ผู้ใช้นำเมาส์ไปเลือกบริเวณเส้น โปรแกรมจะขึ้นข้อความเตือนว่าไม่สามารถทำงานได้ให้นำเมาส์ไปเลือกยังจุดพิกัด ดังรูปที่ ข-11 หลังจากที่ได้ทำการย้ายเรียบร้อยแล้ว จะมีลักษณะดังรูปที่ ข-12

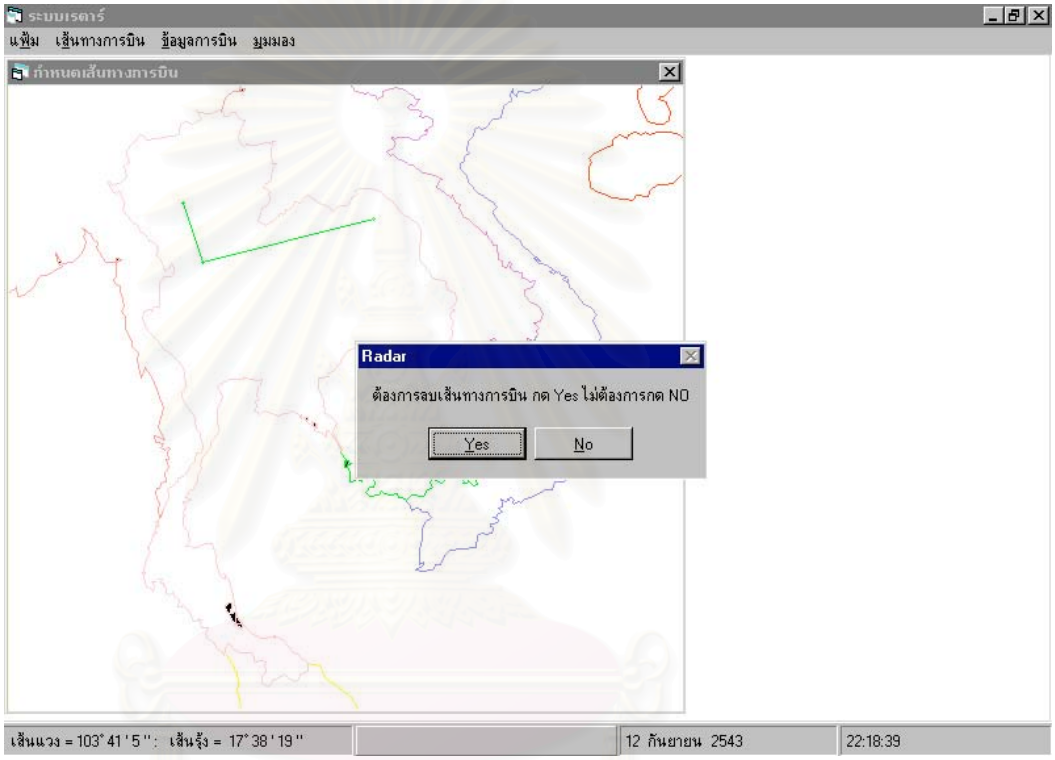


รูปที่ ข-11 แสดงข้อความเตือนสามารถทำการย้ายเส้นทางการบินได้



รูปที่ ข-12 แสดงหน้าจอของผลลัพธ์ที่ได้หลังจากย้ายเส้นทางการบินเรียบร้อยแล้ว

2.3.1 ลบเส้นทางการบิน ลักษณะการทำงานคล้ายกับการย้ายเส้นทางการบิน โดยจะต้องนำเมาส์ไปเลือกยังจุดพิกัดแต่ถ้านำเมาส์ไปเลือกยังบริเวณเส้นสีเขียว ก็จะขึ้นข้อความเตือนว่าไม่สามารถเลือกบริเวณดังกล่าวได้ หลังจากที่ได้เลือกจุดพิกัดที่ต้องการลบแล้วนั้น โปรแกรมจะถามเพื่อยืนยันคำตอบอีกครั้ง ลักษณะดังรูปที่ ข-13 ถ้าผู้ใช้ต้องการยืนยันให้เลือก Yes แต่หากไม่ต้องการให้เลือก No



รูปที่ ข-13 แสดงลักษณะหน้าจอของเมนูลบเส้นทางการบิน

2.4 ข้อมูลการบิน

2.4.1 สร้างข้อมูลการบิน

2.4.2 กำหนดช่วงเวลาเพื่อใช้ในการสร้างข้อมูล

2.5 มุมมอง เพื่อใช้ขยายภาพของแผนที่ตามมุมมองที่ต้องการลักษณะโดยรวมของมุมมองได้อธิบายไปแล้วข้างต้น จากเมนูดังกล่าวมีให้เลือกดังนี้

2.5.1 100 %

2.5.2 200 %

2.5.3 400 %

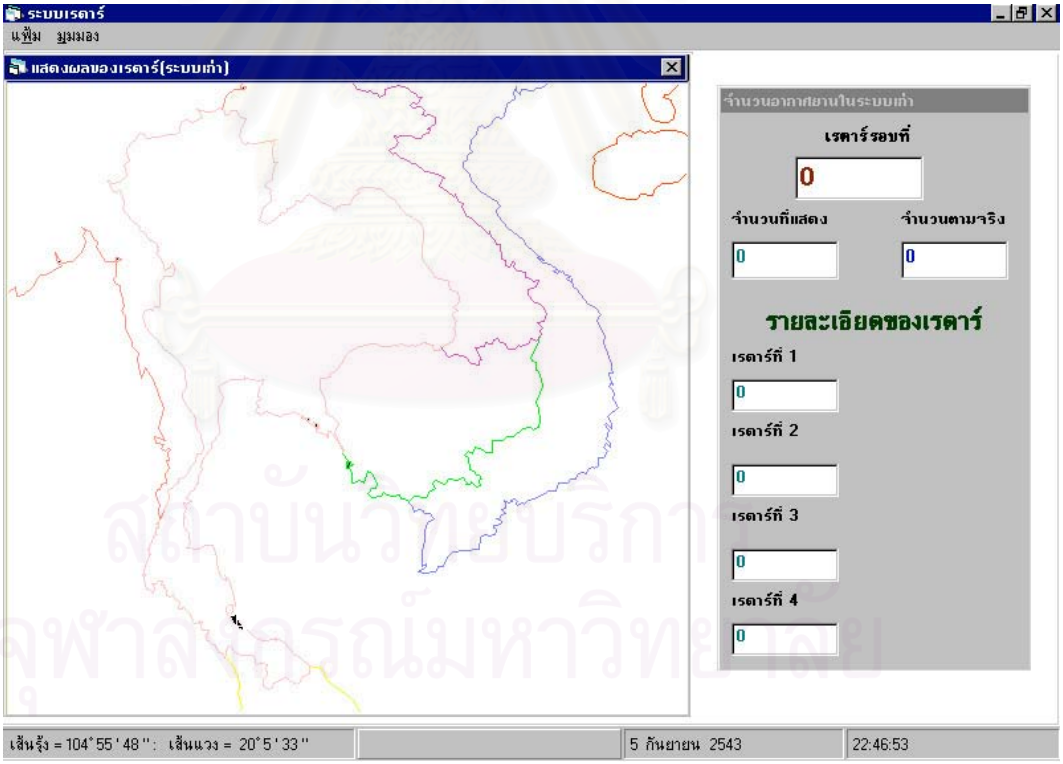
2.5.4 800 %

3. ส่งข้อมูล

ในส่วนนี้แบ่งการทำงานออกเป็น 3 ส่วนคือ 1. ส่งข้อมูลแบบระบบเดิม 2. ส่งข้อมูลผ่านการกรองและขจัดความซับซ้อน 3. ส่งข้อมูล 2 ระบบ สามารถอธิบายการทำงานในแต่ละส่วนได้ดังนี้

3.1 ส่งข้อมูลแบบระบบเดิม

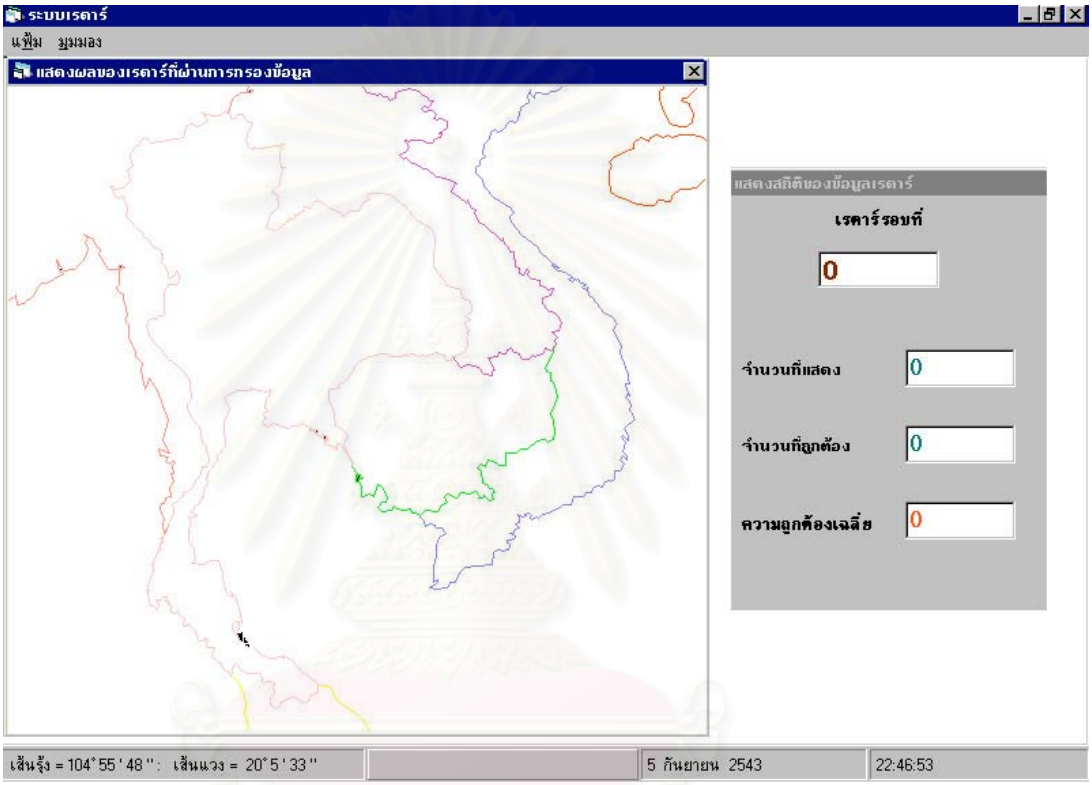
การส่งข้อมูลแบบระบบเดิมนี้เป็นการแสดงผลของสัญญาณเรดาร์ที่ได้รับจากเรดาร์ทั้ง 4 เครื่อง โดยการแสดงสัญญาณจะแสดงโดยตรง ซึ่งมิได้มีการประมวลผล ซึ่งผลที่ได้ออกมาอาจไม่ตรงกับความเป็นจริงเนื่องจาก มีอากาศยานบางเครื่องจะปรากฏอยู่หลายเรดาร์ จากการทำงานในส่วนนี้สามารถแสดงได้ดังรูปต่อไปนี้



รูปที่ ข-14 แสดงลักษณะหน้าจอของเมนูส่งข้อมูลแบบระบบเก่า

3.2 ส่งข้อมูลผ่านการกรองและขจัดความซับซ้อน

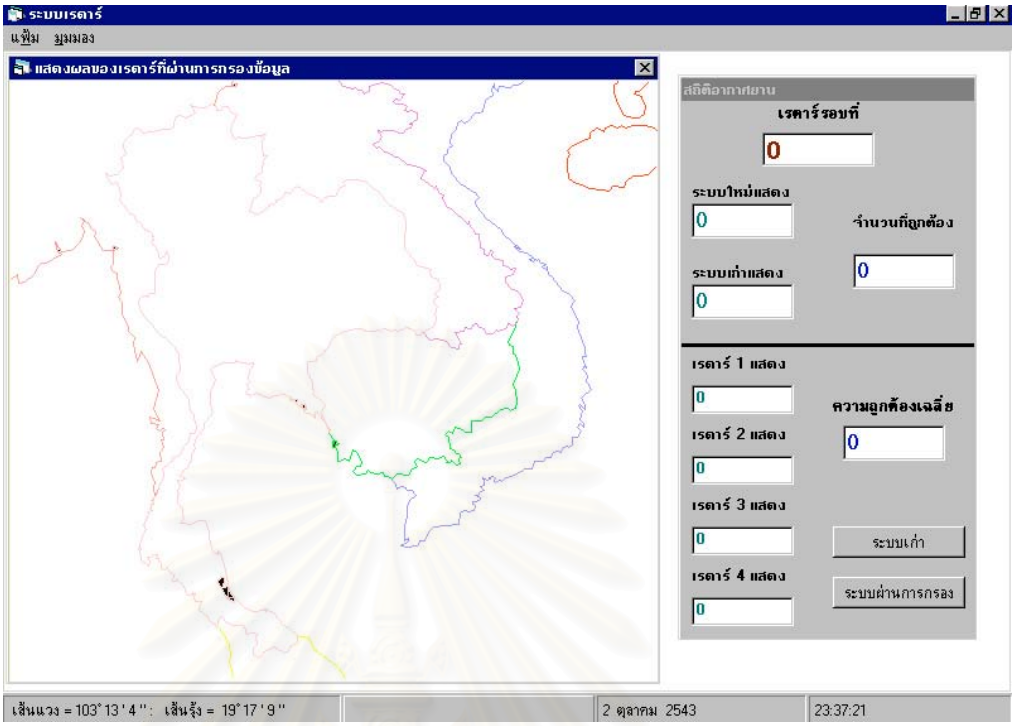
เป็นการส่งสัญญาณเรดาร์ที่ได้รับจากเรดาร์ทั้ง 4 เครื่อง แต่ได้รับการกรองและขจัดความซับซ้อน ซึ่งผลที่ได้จะทำให้อากาศที่ซ้ำซ้อนจะถูกขจัดออกไป จากการทำงานในส่วนนี้สามารถแสดงได้ดังรูปต่อไปนี้



รูปที่ ข-15 แสดงลักษณะหน้าจอของเมนูส่งข้อมูลผ่านการกรองและขจัดความซับซ้อน

3.3 ส่งข้อมูล 2 ระบบ

เป็นการผนวกทั้งสองระบบเข้าด้วยกันเพื่อทำให้เห็นภาพความแตกต่างระหว่างระบบเดิมและระบบที่ได้ผ่านการกรองและขจัดความซับซ้อน ลักษณะหน้าจอดังรูปที่ ข-16



รูปที่ ข-16 แสดงลักษณะหน้าจอของเมนูส่งข้อมูล 2 ระบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดแฟ้มรหัสต้นฉบับการสร้างโปรแกรมต้นแบบเพื่อการกรองข้อมูลและขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

1. การพัฒนาโปรแกรมเมนูหลัก

1.1 Radar.frm แสดงเมนูหลัก ประกอบด้วยส่วนต่างๆคือ

1.1.1 RadarPosition_Click โปรแกรมย่อยเรียกโปรแกรมสร้างเรดาร์

```
' Create radar
Private Sub RadarPosition_Click()
MapRadar.Show
End Sub
```

1.1.2 TrackPlan_Click โปรแกรมย่อยเรียกโปรแกรมสร้างเส้นทางการบิน

```
'Create flight track
Private Sub TrackPlan_Click()
AirCraft.Show
End Sub
```

1.1.3 SendDataOld_Click โปรแกรมย่อยเรียกโปรแกรมแสดงเส้นทางการบินแบบเก่า

```
'Send data in old system
Private Sub SendDataOld_Click()
OldMonitor.Show
OldStat.Show
End Sub
```

1.1.4 SendDataNew_Click โปรแกรมย่อยเรียกโปรแกรมแสดงเส้นทางการบินที่ผ่านการกรองและขจัดความซ้ำซ้อนของข้อมูล

```
'Send data in new system
Private Sub SendDataNew_Click()
Dim RetVal
NewMonitor.Show
NewStat.Show
End Sub
```

1.1.5 Bothsystem_Click เรียกโปรแกรมเส้นการบินทั้งสองระบบ

```
' Display New nad Old system
Private Sub Bothsystem_Click()
OldMonitor2.Show
NewMonitor2.Show
AllStat.Show
NewMonitor2.SetFocus
End Sub
```

1.2 MyFun.bas เก็บค่าตัวแปรต่างๆที่ใช้ในระบบ

```
Type ACData
ACType As String
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
ACSpeed As Double
Lon As Long
Lat As Long
Class As String
Sign  As String
End Type

Type TrackData
MaxTrack As Integer
TrackNo As Integer
Lon As Long
Lat As Long
Sign  As String
End Type

Global Const Mile2Km = 1.852 ' convert mike to kilometres
Global Const MInterval = 60 ' convert mile to of Min
Global Const RadiusRadar = 4500 ' The radius of radar is 75 miles = 75 * 60
Global Const RadarTime = 4  ' Radar rotate 4 second per 1 rotate
Global Deg As Long, Min As Long, Sec As Long
Global TrackInfo(1 To 200, 1 To 32) As ACData
Global RadarTheta(1 To 500) As Integer
Global TrackDraw(1 To 8, 1 To 200) As TrackData
Global TrackDrawNew(1 To 8, 1 To 200) As TrackData
'This Function used to calculate Second
Function Second(Deg, Min, Sec) As Double
Second = (Deg * 3600) + (Min * 60) + Sec
End Function
```

2.1 MapRadar.frm สร้างแผนที่ในระบบเรดาร์โดยอ่านข้อมูลตำแหน่งทางภูมิศาสตร์ของประเทศไทยและประเทศใกล้เคียง เพื่อแสดงเป็นแผนที่และแสดงตำแหน่งของเมาส์ตามเหน่งทางภูมิศาสตร์ สร้างและปรับปรุงตำแหน่งของเรดาร์

2.1.1 FileOpen_Click โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์และแสดงตำแหน่งของเรดาร์

```
Private Sub FileOpen_Click()
Dim i
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Rad)|*.Rad"
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the Open dialog box
CommonDialog1.ShowOpen
Open CommonDialog1.FileName For Input As #2
Line Input #2, InputText
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
RadarCount = Val(InputText)
Do While Not EOF(2)
Line Input #2, InputText
RadarPosition(i, 1) = Val(InputText)
Line Input #2, InputText
RadarPosition(i, 2) = Val(InputText)
i = i + 1
Loop
Close #2
PlotMap
ReDrawRadar
Exit Sub

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.1.2 FileClose_Click โปรแกรมย่อยปิดข้อมูลเรดาร์และแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
‘close radar file and update map
Private Sub FileClose_Click()
RadarCount = 1
PlotMap
End Sub
```

2.1.3 FileSave_Click โปรแกรมย่อยบันทึกแฟ้มข้อมูลเรดาร์เก็บตำแหน่งของเรดาร์

```
Private Sub FileSave_Click()
Dim i
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Rad)|*.Rad"
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the Save dialog box
CommonDialog1.ShowSave
Open CommonDialog1.FileName For Output As #2
Write #2, RadarCount
Do While (i < RadarCount)
Write #2, RadarPosition(i, 1)
Write #2, RadarPosition(i, 2)
i = i + 1
Loop
Close #2
' Display name of selected file
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Exit Sub

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.1.4 FileSavaAs_Click

โปรแกรมย่อยบันทึกแฟ้มข้อมูลเรดาร์เก็บตำแหน่งของเรดาร์โดยสามารถกำหนดชื่อได้ตามที่ต้องการ

```
Private Sub FileSavaAs_Click()
Dim i
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Rad)|*.Rad"
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the SaveAs dialog box
CommonDialog1.DialogTitle = "Save As"
CommonDialog1.ShowSave
Open CommonDialog1.FileName For Output As #2
Write #2, RadarCount
Do While (i < RadarCount)
Write #2, RadarPosition(i, 1)
Write #2, RadarPosition(i, 2)
i = i + 1
Loop
Close #2
Exit Sub

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.1.5 PoltNeighborBorder

โปรแกรมย่อยแสดงแผนที่ประเทศใกล้เคียง

```
Public Sub PoltNeighborBorder()
Dim Dx, Mx, Sx, My, Sy, MyScX, MyScY
Dim x, Y As Long
Dim SLon, SLat
Dim Tmpx, Tmpy
Dim GEO
NTmp = 0
MyScX = SCX
MyScY = SCY
Open App.Path & "\Map\Neighbor.GEO" For Input As #1
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
ForeColor = &H8080FF
Do While Not EOF(1)
GEO = ""
Line Input #1, TextLine
GEO = TextLine
'Calculate Upper-Lower
Dx = Val(Mid(GEO, 8, 3))
Mx = Val(Mid(GEO, 11, 2))
Sx = Val(Mid(GEO, 13, 2))
Dy = Val(Mid(GEO, 1, 2))
My = Val(Mid(GEO, 3, 2))
Sy = Val(Mid(GEO, 5, 2))
x = ((Second(Dx, Mx, Sx)) - (Second(BLLonDeg, BLLonMin, BLLonSec))) * MyScX
Y = (Second(URLatDeg, URLatMin, URLatSec) - Second(Dy, My, Sy)) * MyScY
Select Case NTmp
Case 0, 71
Case 92
ForeColor = &HFFFF&
Case 102
Case 111
ForeColor = &HFF00&
Case 196
ForeColor = &HC000C0
Case 342
ForeColor = &HFF8080
Case 523
ForeColor = &HFF&
Case 570
Case 602
ForeColor = &HC0FFFF
Case Else
Line (Tmpx, Tmpy)-(x, Y)
End Select
Tmpx = x
Tmpy = Y
NTmp = NTmp + 1
Loop
Close #1
End Sub
```

2.1.6 PoltThaiBorder โปรแกรมย่อยแสดงแผนที่ประเทศไทย

```
Public Sub PoltThaiBorder()
Dim DLat, MLat, DLon, MLon
Dim SLon, SLat
Dim Tmpx, Tmpy, MyScX, MyScY As Double
Dim x, Y As Long
Tmpx = 0
MyScX = SCX
MyScY = SCY
ForeColor = &HC0C0FF
Open "c:\thesis\map\Thaibder.geo" For Input As #1

Do While Not EOF(1)
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Georef = ""
Line Input #1, TextLine
Georef = TextLine
Select Case Mid(Georef, 1, 1)
Case "U"
DLon = 90
Case "V"
DLon = 105
End Select
Select Case Mid(Georef, 3, 1)
Case "A", "B", "C", "D", "E", "F", "G", "H"
DLon = DLon + Asc(Mid(Georef, 3, 1)) - 65
Case "J", "K", "L", "M", "N"
DLon = DLon + Asc(Mid(Georef, 3, 1)) - 66
Case "P", "Q", "R"
DLon = DLon + Asc(Mid(Georef, 3, 1)) - 67
End Select
MLon = Val(Mid(Georef, 5, 2))
SLon = (DLon * 3600) + (MLon * 60)
Select Case Mid(Georef, 2, 1)
Case "G"
DLat = 0
Case "H"
DLat = 15
End Select
Select Case Mid(Georef, 4, 1)
Case "A", "B", "C", "D", "E", "F", "G", "H"
DLat = DLat + Asc(Mid(Georef, 4, 1)) - 65
Case "J", "K", "L", "M", "N"
DLat = DLat + Asc(Mid(Georef, 4, 1)) - 66
Case "P", "Q", "R"
DLat = DLat + Asc(Mid(Georef, 4, 1)) - 67
End Select
MLat = Val(Mid(Georef, 7, 2))
SLat = (DLat * 3600) + (MLat * 60)
x = (SLon - Second(BLLonDeg, BLLonMin, BLLonSec)) * MyScX
Y = (Second(URLatDeg, URLatMin, URLatSec) - SLat) * MyScY
If Tmpx = 0 Then
Else
Line (x, Y)-(Tmpx, Tmpy)
End If
Tmpx = x
Tmpy = Y
Loop
Close #1
End Sub
```

2.1.7 InRadarArea โปรแกรมย่อยเพื่อทดสอบว่าจุดที่กำหนดอยู่ในรัศมีของเรดาร์หรือไม่

```
Private Function InRadarArea(x As Double, Y As Double) As Boolean
Dim i
i = 1
Do While (i < RadarCount)
If (Sqr(((RadarPosition(i, 1) - x) ^ 2) + ((RadarPosition(i, 2) - Y) ^ 2)) < 4500) Then
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
InRadarArea = True
FoundNo = i
i = RadarCount
Else
InRadarArea = False
End If
i = i + 1
Loop
End Function
```

2.1.8 RadarCreate_Click โปรแกรมย่อยสร้างรัศมีของเรดาร์จากจุดศูนย์กลาง

```
Private Sub RadarCreate_Click()
MousePointer = 2
File.Enabled = False
View.Enabled = False
MyMode = "ADDRADAR"
End Sub
```

2.1.9 RadarMove_Click โปรแกรมย่อยย้ายรัศมีของเรดาร์

```
Private Sub RadarMove_Click()
File.Enabled = False
View.Enabled = False
MouseIcon = LoadPicture(App.Path & "\icon\move.ico")
MousePointer = 99
MyMode = "MOVE"
End Sub
```

2.1.10 RadarDelete_Click โปรแกรมย่อยลบรัศมีของเรดาร์

```
Private Sub RadarDelete_Click()
File.Enabled = False
View.Enabled = False
'MouseIcon = LoadPicture("C:\thesis\icon\delete.ico")
'MousePointer = 99
MyMode = "DELETE"
End Sub
```

2.1.11 ViewModify1_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด100%

```
Private Sub ViewModify1_Click()
MyScale = 1
FindBL_URGEO ("0950000E0050000N1110000E0210000N")
PlotMap
ClearModify
ViewModify1.Checked = True
ReDrawRadar
End Sub
```

2.1.12 ViewModify2_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด200%

```
Private Sub ViewModify2_Click()
MyScale = 2
MyMode = "MODIFY"
```

```
ModifyClick = True
MouseIcon = LoadPicture(App.Path & "\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Radar.Enabled = False
ClearModify
ViewModify2.Checked = True
End Sub
```

2.1.13 ViewModify4_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด400%

```
Private Sub ViewModify4_Click()
MyScale = 4
MyMode = "MODIFY"
ModifyClick = True
MouseIcon = LoadPicture(App.Path & "\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Radar.Enabled = False
ClearModify
ViewModify4.Checked = True
End Sub
```

2.1.14 ViewModify8_Click โปรแกรมย่อยปรับขนาดของเรดาร์และแผนที่ขนาด800%

```
Private Sub ViewModify8_Click()
MyScale = 8
ModifyClick = True
MyMode = "MODIFY"
MouseIcon = LoadPicture(App.Path & "\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Radar.Enabled = False
ClearModify
ViewModify8.Checked = True
End Sub
```

2.2 AirCraft.frm สร้างเส้นทางการบิน สร้างโปรโตคอลเส้นทางการบิน เปิดและบันทึกเส้นทางการบินบนแฟ้มข้อความ

2.2.1 FileRadarOpen_Click โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์และแสดงตำแหน่งของเรดาร์

```
Private Sub FileRadarOpen_Click()
Dim i
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Rad)|*.Rad"
```

```
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the Open dialog box
CommonDialog1.ShowOpen
Open CommonDialog1.FileName For Input As #2
RadarName = CommonDialog1.FileName
Line Input #2, InputText
RadarCount = Val(InputText)
Do While Not EOF(2)
Line Input #2, InputText
RadarPosition(i, 1) = Val(InputText)
Line Input #2, InputText
RadarPosition(i, 2) = Val(InputText)
i = i + 1
Loop
Close #2
PlotMap
ReDrawRadar
ReDrawFlight
Exit Sub

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.2.2 FileAirCraftOpen_Click โปรแกรมย่อยเปิดแฟ้มข้อมูลเรดาร์ เส้นทางการบิน และแสดงตำแหน่งของเรดาร์

```
Private Sub FileAirCraftOpen_Click()
Dim StrLine As String
Dim i, j
i = 1
Do While (i <= 200)
j = 1
Do While (j <= 32)
TrackInfo(i, j).ACType = "X"
j = j + 1
Loop
i = i + 1
Loop
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Tck)|*.Tck"
' Specify default filter
CommonDialog1.FilterIndex = 2
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
' Display the Open dialog box
CommonDialog1.ShowOpen
Open CommonDialog1.FileName For Input As #2
Line Input #2, InputText
RadarCount = Val(InputText)
Do While (i < RadarCount)
Line Input #2, InputText
RadarPosition(i, 1) = Val(InputText)
Line Input #2, InputText
RadarPosition(i, 2) = Val(InputText)
i = i + 1
Loop
Do While Not EOF(2)
' Aircraft Number
Line Input #2, InputText
i = Val(InputText)
' Track Number
Line Input #2, InputText
j = Val(InputText)
' Aircraft Type
Line Input #2, InputText
TrackInfo(i, j).ACType = Mid(InputText, 2, (Len(InputText) - 2))
Line Input #2, InputText
TrackInfo(i, j).ACSpeed = Val(InputText)
Line Input #2, InputText
TrackInfo(i, j).Lon = Val(InputText)
Line Input #2, InputText
TrackInfo(i, j).Lat = Val(InputText)
Line Input #2, InputText
TrackInfo(i, j).Class = Mid(InputText, 2, 1)
Line Input #2, InputText
TrackInfo(i, j).Sign = Mid(InputText, 2, 1)
Loop
Close #2
TrackDataCount = i
PlotMap
ReDrawRadar
ReDrawFlight
Exit Sub

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.2.3 FileAirCraftClose_Click โปรแกรมย่อยปิดแฟ้มข้อมูลเรดาร์ เส้นทางการบิน แสดงตำแหน่งของเรดาร์และแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
Private Sub FileAirCraftClose_Click()
Dim i, j
RadarCount = 1
PlotMap
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
ReDrawRadar
i = 1
Do While (i <= 200)
j = 1
Do While (j <= 32)
TrackInfo(i, j).ACType = "X"
j = j + 1
Loop
i = i + 1
Loop
End Sub
```

2.2.4 FileSave_Click โปรแกรมย่อยบันทึกแฟ้มข้อของข้อมูลเรดาร์ และเส้นทางการ

```
Private Sub FileSave_Click()
Dim i, j
If (RadarCount < 2 Or TrackDataCount < 2) Then
MsgBox ("ต้องสร■งเส้นทางการบินและเรดาร์ก่อนที่จะบันทึก")
Else
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Tck)|*.Tck"
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the Save dialog box
CommonDialog1.ShowSave
Open CommonDialog1.FileName For Output As #2
Write #2, RadarCount
Do While (i < RadarCount)
Write #2, RadarPosition(i, 1)
Write #2, RadarPosition(i, 2)
i = i + 1
Loop
i = 1
j = 1
Do While TrackInfo(i, j).ACType <> "X"
Do While TrackInfo(i, j).ACType <> "X"
' Radar number
Write #2, i
' Position number
Write #2, j
' Aircraft Type
Write #2, TrackInfo(i, j).ACType
' Aircraft Speed
Write #2, TrackInfo(i, j).ACSpeed
' Position Lontitude
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Write #2, TrackInfo(i, j).Lon
' Position Latitude
Write #2, TrackInfo(i, j).Lat
' Aircraft Class
Write #2, TrackInfo(i, j).Class
' Aircraft Sign
Write #2, TrackInfo(i, j).Sign
j = j + 1
Loop
j = 1
i = i + 1
Loop
Close #2
' Display name of selected file
Exit Sub
End If

ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.2.5 FileSaveAs_Click โปรแกรมย่อยบันทึกแฟ้มของข้อมูลเรดาร์ และเส้นทางการโดยสามารถกำหนดชื่อได้ตามที่ต้องการ

```
Private Sub FileSaveAs_Click()
Dim i, j
If (RadarCount < 2 Or TrackDataCount < 2) Then
MsgBox ("ต้องสร้างเส้นทางการบินและเรดาร์ก่อนที่จะบันทึก")
Else
i = 1
' Set working path
CommonDialog1.InitDir = App.Path & "\Radar"
' Set CancelError is True
CommonDialog1.CancelError = True
On Error GoTo ErrHandler
' Set flags
CommonDialog1.Flags = cdlOFNHideReadOnly
' Set filters
CommonDialog1.Filter = "Radar Files (*.Tck)|*.Tck"
' Specify default filter
CommonDialog1.FilterIndex = 2
' Display the Save dialog box
CommonDialog1.DialogTitle = "Save As"
CommonDialog1.ShowSave
Open CommonDialog1.FileName For Output As #2
Write #2, RadarCount
Do While (i < RadarCount)
Write #2, RadarPosition(i, 1)
Write #2, RadarPosition(i, 2)
i = i + 1
Loop
i = 1
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
j = 1
Do While TrackInfo(i, j).ACType <> "X"
Do While TrackInfo(i, j).ACType <> "X"
Write #2, i
Write #2, j
Write #2, TrackInfo(i, j).ACType
Write #2, TrackInfo(i, j).ACSpeed
Write #2, TrackInfo(i, j).Lon
Write #2, TrackInfo(i, j).Lat
Write #2, TrackInfo(i, j).Class
Write #2, TrackInfo(i, j).Sign
j = j + 1
Loop
j = 1
i = i + 1
Loop
Close #2
' Display name of selected file
Exit Sub
End If
ErrHandler:
'User pressed the Cancel button
Exit Sub
End Sub
```

2.2.6 TrackCreate_Click โปรแกรมย่อยสร้างเส้นทางการบินโดยการ

```
Private Sub TrackCreate_Click()
'Me.Hide
File.Enabled = False
Track.Enabled = False
TrackData.Enabled = False
View.Enabled = False
AirCraftDetail.Show
End Sub
```

2.2.7 TrackMove_Click โปรแกรมย่อยย้ายเส้นทางการบินโดยการ

```
Private Sub TrackMove_Click()
File.Enabled = False
Track.Enabled = False
TrackData.Enabled = False
View.Enabled = False
MouseIcon = LoadPicture("C:\thesis\icon\move.ico")
MousePointer = 99
MyMode = "MOVE"
End Sub
```

2.2.8 TrackDelete_Click โปรแกรมย่อยลบเส้นทางการบินโดยการ

```
Private Sub TrackDelete_Click()
MyMode = "DELETE"
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2.9 TrackDataCreate_Click โปรแกรมย่อยสร้างข้อมูลของเรดาร์จากข้อมูลเส้นทางการบินแลนและสถานที่ตั้งของเรดาร์

```
Private Sub TrackDataCreate_Click()
ProgressStatus.Show
GenerrateFlightData
LoadFlightData2DB
ProgressStatus.Hide
ProgressStatus.Caption = "ระบบทำการจำลำดับข้อมูล"
ProgressStatus.Show
GenerateProtocolData
ProgressStatus.Caption = "ระบบทำการสร้างข้อมูลเพื่อสร้างสถิติ"
ProgressStatus.Show
TotalAircraft
ProgressStatus.Hide
End Sub
```

2.2.10 CreateTimeSlice_Click โปรแกรมย่อยรับค่าส่วนของวินาทีเพื่อใช้ในการสร้างข้อมูลของเรดาร์

```
Private Sub CreateTimeSlice_Click()
TimeSlicefrm.Show
End Sub
```

2.2.11 ViewModify1_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด100%

```
Private Sub ViewModify1_Click()
MyScale = 1
FindBL_URGEO ("0950000E0050000N1110000E0210000N")
PlotMap
ClearModify
ViewModify1.Checked = True
ReDrawRadar
ReDrawFlight
End Sub
```

2.2.12 ViewModify2_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด200%

```
Private Sub ViewModify2_Click()
MyScale = 2
MyMode = "MODIFY"
ModifyClick = True
MouseIcon = LoadPicture("C:\thesis\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Track.Enabled = False
ClearModify
ViewModify2.Checked = True
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2.13 ViewModify4_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด400%

```
Private Sub ViewModify4_Click()
MyScale = 4
MyMode = "MODIFY"
ModifyClick = True
MouseIcon = LoadPicture("C:\thesis\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Track.Enabled = False
ClearModify
ViewModify4.Checked = True
End Sub
```

2.2.14 ViewModify8_Click โปรแกรมย่อยปรับขนาดของเรดาร์ เส้นทางการบินและแผนที่ขนาด800%

```
Private Sub ViewModify8_Click()
MyScale = 8
ModifyClick = True
MyMode = "MODIFY"
MouseIcon = LoadPicture("C:\thesis\icon\modify.ico")
MousePointer = 99
File.Enabled = False
Track.Enabled = False
ClearModify
ViewModify8.Checked = True
End Sub
```

2.3 AirCraftDetail.frm แสดงและเก็บรายละเอียดของอากาศยาน เครื่องหมายของอากาศยาน

2.3.1 Combo1_Click โปรแกรมย่อยแสดงภาพและความเร็วสูงสุดของอากาศยาน

```
Private Sub Command1_Click()
AirCraft.MyMode = "CREATETRACK"
AirCraft.MouseIcon = LoadPicture("C:\thesis\icon\airplain.ico")
AirCraft.MousePointer = 99
Select Case MySign
Case "F", "I", "K", "P", "E", "Z"
AirCraft.MyClass = "F"
Case Else
AirCraft.MyClass = "H"
End Select
Me.Visible = False
End Sub
```

2.3.2 Command2_Click โปรแกรมย่อยเก็บค่าความเร็วและฝ่ายของอากาศ

```
Private Sub Command2_Click()
AirCraft.MyMode = "NON"
Unload Me
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
AirCraft.File.Enabled = True
AirCraft.Track.Enabled = True
AirCraft.TrackData.Enabled = True
AirCraft.View.Enabled = True
End Sub
```

2.3.3 Command3_Click โปรแกรมย่อยยกเลิกการเลือกอากาศยาน

```
Private Sub Command3_Click()
Me.Hide
frmACDetail.Show
End Sub
```

2.4 Pointdetail.frm แสดงและเก็บตำแหน่งของเส้นทางการบินที่ได้กำหนด

2.4.1 Command1_Click โปรแกรมย่อยรับค่าตำแหน่งของเส้นทางการบิน

```
Private Sub Command1_Click()
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).ACType = Mystr
'Label2.Caption
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).ACSpeed = Val(Text7.Text)
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Lon = Second(Val(Text1.Text), Val(Text2.Text), Val(Text3.Text))
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Lat = Second(Val(Text4.Text), Val(Text5.Text), Val(Text6.Text))
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Class = AirCraft.MyClass
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Sign = AirCraftDetail.MySign
Me.Visible = False
AirCraft.DrawFlight
End Sub
```

2.4.2 Command2_Click โปรแกรมย่อยยอเลิกตำแหน่งของเส้นทางการบิน

```
Private Sub Command2_Click()
AirCraft.File.Enabled = True
AirCraft.Track.Enabled = True
AirCraft.TrackData.Enabled = True
AirCraft.View.Enabled = True
Me.Visible = False
End Sub
```

2.4.3 Command3_Click โปรแกรมย่อยจบการกำหนดเส้นทางการบิน

```
Private Sub Command3_Click()
AirCraft.MousePointer = 0
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).ACType = Mystr
'Label2.Caption
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).ACSpeed = Val(Text7.Text)
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Lon = Second(Val(Text1.Text), Val(Text2.Text), Val(Text3.Text))
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Lat = Second(Val(Text4.Text), Val(Text5.Text), Val(Text6.Text))
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Class = Trim(AirCraft.MyClass)
TrackInfo(AirCraft.TrackDataCount, AirCraft.TrackNo).Sign = Trim(AirCraftDetail.MySign)
AirCraft.CurrTrack = False
AirCraft.DrawFlight
Me.Visible = False
AirCraft.MyMode = "NON"
AirCraft.File.Enabled = True
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
AirCraft.Track.Enabled = True
AirCraft.TrackData.Enabled = True
AirCraft.View.Enabled = True
End Sub
```

2.5 TimeSlicefrm.frm รับข้อมูลช่วงของเวลาในส่วนของวินาที

```
Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Command2_Click()
AirCraft.TimeSlice = Val(Text1)
Unload Me
End Sub

Private Sub Form_Load()
'set form
Left = (Screen.Width - Me.Width) / 2     ' Center form horizontally.
Top = (Screen.Height - Me.Height - 2000) / 2 ' Center form vertically.
Text1.Text = AirCraft.TimeSlice
End Sub
```

3.1 oldradar.java สร้างข้อมูลเส้นทางการบินและเรดาร์ในระบบเก่า

```
//Title:       Your Product Name
//Version:
//Copyright:    Copyright (c) 1998
//Author:      Nukool  Taveesak
//Company:     Chulalongkorn University
//Description:  Program to create aircraft data and radar syncronization


package radar;
import java.io.*;

public class oldradar  extends Thread{
private RandomAccessFile input;
public static long sleeptime = 4000;
public int nradar;
public String lon;
public String lat;
public String strnradar;
public int radarnum;
//Construct the application

public oldradar() {
}
```

3.1.1 readpositionradar โปรแกรมย่อยอ่านตำแหน่งของเรดาร์จากแฟ้มข้อมูล

```
public String readpositionradar()
{
String data="";
// input the values from the file
```

```
try {
data = input.readLine();
System.out.println("Position of Radar = " + data);
return(data.trim());
}
catch ( EOFException eof ) {
System.err.println( "****Error read EOF from file\n");
return(data);
}

catch ( IOException e ) {
System.err.println( "Error during read from file\n" +
e.toString() );
return(data);
}
}
```

3.1.2 readnumberradar โปรแกรมย่อยอ่านจำนวนของเรดาร์จากแฟ้มข้อมูล

```
public int readnumberradar()
{
String data;
int myradar=0;
// input the values from the file
try {
data = new String(input.readLine());
data = data.trim();
Integer test = new Integer(data);
myradar = test.intValue();
System.out.println("Nuber of Radar = " + myradar );
return(myradar);
}

catch ( IOException e ) {
System.err.println( "Error during read from file\n" +
e.toString() );
System.exit( 1 );
}
return(myradar);
}
```

3.1.3 openfile โปรแกรมย่อยเปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

```
private int openfile() {
try {
input = new RandomAccessFile( "c:/Thesis/Radar/Flight/RadarData.RAD", "r");
return (1);
}
catch ( IOException e ) {
System.err.println( "File not opened properly\n" +
e.toString() );
return (0);
}
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.1.4 closeFile โปรแกรมย่อยปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

```
private void closeFile()
{
try {
input.close();
//System.exit( 0 );
}
catch ( IOException e ) {
System.err.println( "Error closing file\n + e.toString()");
//System.exit( 1 );
}
}
```

3.1.5 oldradar ส่งข้อมูลที่ตั้งของเรดาร์

```
public oldradar() {
int i;
if (openfile() == 1) {
System.out.println("open file OK!");
nradar =  readnumberradar();
strnradar = strnradar.valueOf(nradar);

for ( radarnum=1; radarnum<nradar; radarnum++ ){
lon = readpositionradar();
lat = readpositionradar();

Long pox = new Long(lon);
long llon = pox.longValue();
llon = (long)(llon/7.5);
pox = new Long(llon);
Long poy = new Long(lat);
long llat = poy.longValue();
llat = (long)(llat/7.5);
poy = new Long(llat);

lon = pox.toString();
lat = poy.toString();

senddata t1 = new senddata(radarnum,lon,lat,strnradar);
t1.start();
}
}
else {
System.out.println("open file Error!");
System.exit( 1 );
}
}
```

3.1.6 main โปรแกรมหลักเรียก class oldradar

```
public static void main(String[] args) {
sleeptime = 4000;
try  {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
}
catch (Exception e) {
}
new oldradar();
}
```

3.2 sendolddata.java ส่งข้อมูลไปยังจอแสดงผลระบบเก่าโดยผ่านทางช่องส่งสัญญาณ

```
public class sendolddata extends Thread{
private int myradar;
private String mylon;
private String mylat;
private String myradarnumber;
private String dataout;
private String protocoldata;

private DatagramPacket sendpacket;
private DatagramSocket socket;
private Socket Simsocket;
private RandomAccessFile input;
private long radartime;

//private DataOutputStream output;
private int socketadd;
```

3.2.1 senddata ส่งข้อมูลผ่านช่องส่งสัญญาณ

```
senddata(int radarnumber, String lon, String lat, String numradar) {
myradar = radarnumber;
mylon = lon;
mylat = lat;
myradarnumber = numradar;
}
```

3.2.2 clientSendData โปรแกรมย่อยเตรียมข้อมูลเป็นกลุ่มข้อมูล(package)เพื่อทำการส่งข้อมูล

```
public void ClientSendData(String mydata){
try {
byte tdata[] =  mydata.getBytes();
sendpacket = new DatagramPacket(tdata, tdata.length, InetAddress.getLocalHost(), socketadd);
socket.send(sendpacket);
System.out.println("Send data address = " + socketadd + "  data =" + mydata);
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}
```

3.2.3 openfile โปรแกรมย่อยเปิดแฟ้มข้อมูลของเรดาร์

```
private void openfile() {
```

```
String filename;
filename = "c:/Thesis/Radar/Flight/ProtocolRadar" + myradar + ".PTC";
try {
input = new RandomAccessFile(filename, "r" );
}
catch ( IOException e ) {
System.err.println( "File not opened properly\n" + e.toString() );
this.stop();
}

}
```

3.2.4 closeFile โปรแกรมย่อยปิดแฟ้มข้อมูลของเรดาร์

```
private void closeFile()
{
try {
input.close();
}
catch ( IOException e ) {
System.err.println( "Error closing file\n ");
}
this.stop();
}
```

3.2.5 openport โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล

```
public void openport(int myadd) {
try {
socketadd = myadd;
socket = new DatagramSocket();
//Simsocket = new Socket(InetAddress.getLocalHost(), myadd);
System.out.println("Client socket be created address = " + socketadd);
}
catch ( IOException se) {
se.printStackTrace();
this.stop();
}
}
```

3.2.6 readprotocol โปรแกรมย่อยอ่านข้อมูลของเรดาร์จากแฟ้มข้อมูลของเรดาร์

```
public String readprotocol()
{
String data="";
// input the values from the file
try {
data = input.readLine();
if (data != null) {
return(data.trim());
}
closeFile();
}
catch ( EOFException eof ) {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
System.err.println( "****Error read EOF from file\n");
closeFile();
return(null);
}
catch ( IOException e ) {
System.err.println( "****Error during read from file\n");
closeFile();
return(null);
}
return(null);
}
```

3.2.7 readprotocollong โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นตัวเลข

```
public long readprotocollong()
{
String data="";
long mylong;

//int test;
// input the values from the file
try {
data = input.readLine();
if (data != null) {
data = data.trim();
Long temp = new Long(data);
mylong =  temp.longValue();
return(mylong);
}
}
catch ( EOFException eof ) {
System.err.println( "****Error read EOF from file\n");
closeFile();
return(0);
}
catch ( IOException e ) {
System.err.println( "****Error during read from file\n");
closeFile();
return(0);
}
return(0);
}
```

3.2.8 readprotocolstring โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นอักขระ

```
public String readprotocolstring()
{
String data="";

// input the values from the file
try {
data = input.readLine();
return(data.substring(1,2));
}
catch ( EOFException eof ) {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
System.err.println( "****Error read EOF from file\n");
closeFile();
return("");
}
catch ( IOException e ) {
System.err.println( "****Error during read from file\n");
closeFile();
return("");
}
}
```

3.2.9 run โปรแกรมวงวนไม่จุดจบ(infinite loop)เพื่อส่งข้อมูล

```
public void run() {
long timecount;
long i;
int  account;
long starttime = 0;
long sleeptime;
long endtime;
long timetosleep;
String protocolout;
String noprotocolout = new String("*#");
noprotocolout += myradar + "#";

openport(1000 + myradar);
dataout = myradarnumber + "#" + myradar + "#" + mylon + "#" + mylat;
// send radar position
ClientSendData(dataout);

openfile();
System.out.println("open file protocol # " + myradar + " OK!");
timecount = 0;
while(true) {

protocolout = "";
//System.out.println("Time count  # " + myradar + " = " + timecount);
radartime = readprotocollong();
timecount += 1;

// end of data

if  (radartime == 0) {
while(true) {
ClientSendData(noprotocolout);
System.out.println("Radar  # " + myradar + " *** Sleep at Time count # " + timecount + " No Data to send  ***");
try{
this.sleep(oldradar.sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "****Sleep error!\n");
this.stop();
}
System.out.println("Radar  # " + myradar + " *** Weak up at Time count # " + timecount + " No Data to send  ***");
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
} // end of while

}

//System.out.println("Time count  # " + myradar + " = " + timecount);
//timetosleep =  (radartime-timecount) * 4000 ;
//System.out.println("Time count  # " + myradar + " = " + timecount + " time to  sleep = " + timetosleep);

if (timecount == radartime) {  // save satrtime
Date startdate = new Date();
starttime = startdate.getTime();
}  // end start time

protocolout  =  readprotocol();
Integer tempaccount =  new Integer(protocolout);
account = tempaccount.intValue();
protocolout += "#";
for (i=1; i <= account; i++){
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
String type =  readprotocolstring();
protocolout += type + "#";
protocolout += readprotocol() + "#";
}
int strlen = protocolout.length();
protocolout = "*" + String.valueOf(strlen) + "#" + protocolout;
if (timecount < radartime) {
while (timecount < radartime) {  // Begin while
ClientSendData(noprotocolout);
System.out.println("Radar  # " + myradar + " *** Sleep at Time count #" + timecount);
try{
this.sleep(oldradar.sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "*****Sleep error!\n");
this.stop();
}
System.out.println("Radar  # " + myradar + " *** Wake up Time count #" + timecount);
timecount  += 1;
}  // end while
Date startdate = new Date();
starttime = startdate.getTime();
}  // end if

ClientSendData(protocolout);
System.out.println("Protocol Out " + protocolout);
Date enddate = new Date();
endtime = enddate.getTime();
sleeptime = oldradar.sleeptime + starttime - endtime;
//System.out.println("Starttime =  " + starttime + " EndTime = " + endtime);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
try{
this.sleep(sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "*****Sleep error!\n");
this.stop();
}

} // end while
}  // end function
```

4.1 newradar.java อ่านข้อมูลเรดาร์ในระบบใหม่ เริ่มเปิดข้อมูลจากแฟ้มของเรดาร์เพื่ออ่านหมายเลขและตำแหน่งของเรดาร์

4.1.1 readpositionradar โปรแกรมย่อยอ่านตำแหน่งของเรดาร์จากแฟ้มข้อมูล

```
public String readpositionradar()
{
String data="";
// input the values from the file
try {
data = input.readLine();
System.out.println("Position of Radar = " + data);
return(data.trim());
}
catch ( EOFException eof ) {
System.err.println( "*****Error read EOF from file\n");
return(data);
}

catch ( IOException e ) {
System.err.println( "Error during read from file\n" +
e.toString() );
return(data);
}
}
```

4.1.2 readnumberradar โปรแกรมย่อยอ่านจำนวนของเรดาร์จากแฟ้มข้อมูล

```
public int readnumberradar()
{
String data;
int myradar=0;
// input the values from the file
try {
data = new String(input.readLine());
data = data.trim();
Integer test = new Integer(data);
myradar = test.intValue();
System.out.println("Nuber of Radar = " + myradar );
return(myradar);
}

catch ( IOException e ) {
```

```
System.err.println( "Error during read from file\n" +
e.toString() );
System.exit( 1 );
}
return(myradar);
}
```

4.1.3 openfile โปรแกรมย่อยเปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

```
private int openfile() {
try {
input = new RandomAccessFile( "c:/Thesis/Radar/Flight/RadarData.RAD", "r");
return (1);
}
catch ( IOException e ) {
System.err.println( "File not opened properly\n" +
e.toString() );
return (0);

}
}
```

4.1.4 closeFile โปรแกรมย่อยปิดแฟ้มข้อมูลที่ตั้งของเรดาร์

```
private void closeFile()
{
try {
input.close();
//System.exit( 0 );
}
catch ( IOException e ) {
System.err.println( "Error closing file\n + e.toString()");
//System.exit( 1 );
}
}
```

4.1.5 newradar ส่งข้อมูลที่ตั้งของเรดาร์

```
public newradar() {
int i;
for (int k=0; k<400; k++) { // Begin for
ptcaircraftno[k] = 0;
ptcaircrafttime[k] = 0;
ptcClass[k] = "";
ptclon[k] = 0;
ptclat[k] = 0;
ptcradarnumber[k] = 0;
ptcradartheta[k] = 0;
ptcspeedlon[k] =0;
ptcspeedlat[k]= 0;
ptcactive[k] = 1;
//ptctracknum[k] =0;

} //end for
if (openfile() == 1) {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
System.out.println("open file OK!");
nradar =  readnumberradar();
strnradar = strnradar.valueOf(nradar);

for ( radarnum=1; radarnum<nradar; radarnum++ ){
lon = readpositionradar();
lat = readpositionradar();
receivedata rec = new receivedata(radarnum);
rec.start();

Long pox = new Long(lon);
long llon = pox.longValue();
llon = (long)(llon/7.5);
pox = new Long(llon);
Long poy = new Long(lat);
long llat = poy.longValue();
llat = (long)(llat/7.5);
poy = new Long(llat);

lon = pox.toString();
lat = poy.toString();

senddata send = new senddata(radarnum,lon,lat,strnradar);
send.start();
}
}
else {
System.out.println("open file Error!");
System.exit( 1 );
}
}
```

4.1.6 Main โปรแกรมหลักเรียก class newradar

```
public static void main(String[] args) {

//radardata[] ptcdata = new radardata[200];

radarstr = "";
datastr = "";
consume = 0;
produce =0;
syncronize syn = new syncronize();
syn.start();
syncronizedata syndata = new syncronizedata();
syndata.start();
sleeptime = 4000;
new newradar();
}
}
```

4.2 realtable.java เก็บข้อมูลของอากาศยานข้อมูลเพื่อการนำไปใช้ในการขจัดความซ้ำซ้อน

```
//Title:      Your Product Name
```

```
//Version:
//Copyright:    Copyright (c) 1998
//Author:       Nukool  Taveesak
//Company:      Chulalongkorn University
//Description:  Your description

package newradar;

class realtable {
public int radarnum[] = new int[400];
public int aircraftno[] = new int[400];
public int state[] = new int[400];
public long lon[] = new long[400];
public long lat[] = new long[400];
public long slon[] = new long[400];
public long slat[] = new long[400];
public int acindex[] = new int[400];
public int active[] = new int[400];
public String Class[] = new String[400];
public int hindex;
public int id;

public realtable() {
for (int i=0; i<400; i++){
radarnum[i] = 0;
aircraftno[i] = 0;
lon[i] = 0;
lat[i] = 0;
slon[i] = 0;
slat[i] = 0;
state[i]  = 0;
acindex[i] = 0;
active[i] = 0;
Class[i] = "";
}
hindex = 0;
id = 0;
}

public int ifexist(int radar, int track)  {
int found =0;
for (int k=0; k<hindex; k++) {
if ((newradar.ptcradarnumber[getkey(radar, track)] == radarnum[k] ) &&
(newradar.ptcaircraftno[getkey(radar, track)] == aircraftno[k])) {
found = 1;
k = 400;
}
else
found = 0;
}
return (found);
}

public void adddata(int radar, int track)  {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
//System.out.print(" Start " );
int index;
index = getkey(radar, track);
if (ifexist(radar, track) == 0) {
System.out.println(" hindex  "  + hindex);

radarnum[hindex] = newradar.ptcradarnumber[index];
aircraftno[hindex] = newradar.ptcaircraftno[index];
lon[hindex] = newradar.ptclon[index];
lat[hindex] = newradar.ptclat[index];
slon[hindex] = newradar.ptcspeedlon[index];
slat[hindex] = newradar.ptcspeedlat[index];
active[hindex] = newradar.ptcactive[index];
Class[hindex] = newradar.ptcClass[index];
state[hindex] = 1;
acindex[hindex] = 0;
//System.out.println(" radar " + radarnum[hindex] + " = " +newradar.ptcradarnumber[getkey(radarnum[hindex], track)]+  " acno " + aircraftno[hindex]);
hindex = hindex + 1;
}
else
{
for (int i=0; i < hindex; i++) {
if ((radarnum[i] == newradar.ptcradarnumber[index])
&& (aircraftno[i] == newradar.ptcaircraftno[index])) {
radarnum[i] = newradar.ptcradarnumber[index];
aircraftno[i] = newradar.ptcaircraftno[index];
lon[i] = newradar.ptclon[index];
lat[i] = newradar.ptclat[index];
slon[i] = newradar.ptcspeedlon[index];
slat[i] = newradar.ptcspeedlat[index];
active[i] = newradar.ptcactive[index];
Class[i] = newradar.ptcClass[index];
state[i] = 3;
//System.out.println("Radar # " + newradar.ptcradarnumber[getkey(radar, track)]+ " track # " + newradar.ptcaircraftno[getkey(radar, track)] + " Index = " +
getindex(radarnum[i], aircraftno[i]) + " = " + i);
}  // end if
}  // end for
}
//System.out.println("Radar # " + radar + " track # " + track + " Index = " + hindex);
}

public void removedata(int index)  {
for (int k=index; k<hindex; k++) {
radarnum[k] = radarnum[k+1];
aircraftno[k] = aircraftno[k+1];
lon[k] = lon[k+1];
lat[k] = lat[k+1];
slon[k] = slon[k+1];
slat[k] = slat[k+1];
state[k] = state[k+1];
acindex[k] = acindex[k+1];
active[k] = active[k+1];
Class[k] = Class[k+1];
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
radarnum[hindex] = 0;
aircraftno[hindex] = 0;
lon[hindex] = 0;
lat[hindex] = 0;
slon[hindex] = 0;
slat[hindex] = 0;
state[hindex] = 0;
acindex[hindex] = 0;
active[hindex] = 0;
Class[hindex] = "";
if (hindex > 0)
hindex -= 1;
}

private int  getkey(int radar, int track) {
return (radar*50+track);
}

public void addnewid(int index)  {
acindex[index] = getid();
}

public  int getid() {
if (id == 999)
id = 0;
else
id = id + 1;
return (id);
}

public int getmaxindex()  {
return (hindex);
}

public int getacid(int radar, int track)  {
return (acindex[getkey(radar, track)]);
}

public int getradarindex(int radar, int track) {
int index=0;
for (int i=0; i<400; i++)
if ((newradar.ptcradarnumber[i] == radar) && (newradar.ptcaircraftno[i] == track))
index = i;
return index;
}

public int getindex(int radar, int track) {
int index=0;
for (int i=0; i<hindex; i++)
if ((radarnum[i] == radar) && (aircraftno[i] == track))
index = i;
return index;
}
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.3 receivedata.java รับข้อมูลจากช่องรับสัญญาณ

4.3.1 receivedata รับข้อมูลจากช่องรับสัญญาณ

```
public class receivedata extends Thread{
public String datain;
private DatagramPacket receivepacket;
private DatagramPacket sendpacket;
private DatagramSocket revsocket, sendsocket;
int portadd;

// Open port for receive data

public receivedata(int radarnum) {
portadd = 3000 +  radarnum;
try {
revsocket = new DatagramSocket(portadd);
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}
```

4.3.2 ClientSendData โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ

```
public void ClientSendData(String mydata){
openport();
try {
byte senddata[] =  mydata.getBytes();
sendpacket = new DatagramPacket(senddata, senddata.length, InetAddress.getLocalHost(), 2000);
sendsocket.send(sendpacket);
//System.out.println("Send data  = " + mydata);
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}
```

4.3.3 openport โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล

```
public void openport() {
try {
sendsocket = new DatagramSocket();
}
catch ( IOException se) {
se.printStackTrace();
this.stop();
}
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.3.4 waitforpacket โปรแกรมย่อยรับข้อมูลมาเก็บไว้เพื่อทำการส่งต่อ

```
private void waitforpacket() {
try {
byte revdata[] =  new byte[8600];
receivepacket = new DatagramPacket(revdata, revdata.length, InetAddress.getLocalHost(), portadd);
revsocket.receive(receivepacket);
String str =  new String(receivepacket.getData());
str = str.trim();
//System.out.println("Received data at port # " + portadd + " = " + str);
ClientSendData(str);


//System.out.println("Reveice Data" + str.charAt(0));
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}
```

4.3.5 run โปรแกรมวงวนไม่จุดจบเพื่อรับข้อมูล

```
public void run() {
while (true) {
waitforpacket();
}
}
```

4.4 repeatable.java เก็บข้อมูลอากาศยานที่ซ้ำซ้อน

4.4.1 repeatedtable เพื่อเก็บข้อมูล

```
public class repeatedtable{
public int radarnum1[] = new int[400];
public int aircraftno1[] = new int[400];
public int active1[] = new int[400];
public int radarnum2[] = new int[400];
public int aircraftno2[] = new int[400];
public int active2[] = new int[400];

public int rindex;

public repeatedtable() {
for (int i=0; i<400; i++){
radarnum1[i] = 0;
aircraftno1[i] = 0;
active1[i] = 0;
radarnum2[i] = 0;
aircraftno2[i] = 0;
active2[i] = 0;
}
rindex = 0;
}
```

4.4.2 addrepeat โปรแกรมย่อยเพิ่มข้อมูลในตาราง

```
public void addrepeat(int i, int j, int l, int k) {
int index1,index2;
index1 = i*50+j;
index2 = l*50+k;
radarnum1[rindex] = newradar.ptcradarnumber[index1];
aircraftno1[rindex] = newradar.ptcaircraftno[index1];
active1[rindex] =   newradar.ptcactive[index1];
radarnum2[rindex] = newradar.ptcradarnumber[index2];
aircraftno2[rindex] = newradar.ptcaircraftno[index2];
active2[rindex] =   newradar.ptcactive[index2];
rindex += 1;
}
```

4.4.3 clear โปรแกรมย่อยลบข้อมูลจากตาราง

```
public void clear() {
for (int i=0; i<=rindex; i++){
radarnum1[i] = 0;
aircraftno1[i] = 0;
active1[i] = 0;
radarnum2[i] = 0;
aircraftno2[i] = 0;
active2[i] = 0;
}
rindex =0;
}
```

4.5 senddata.java ส่งข้อมูลที่ผ่านการขจัดความซ้ำซ้อนและกรองข้อมูลไปยังจอแสดงผลระบบใหม่

4.5.1 senddata ส่งข้อมูลผ่านช่องส่งสัญญาณ

```
public class senddata extends Thread{
private int myradar;
private String mylon;
private String mylat;
private String myradarnumber;
private String dataout;
private String protocoldata;

private DatagramPacket sendpacket;
private DatagramSocket socket;
//private Socket Simsocket;
private RandomAccessFile input;
private long radartime;

//private DataOutputStream output;
private int socketadd;

senddata(int radarnumber, String lon, String lat, String numradar) {
myradar = radarnumber;
mylon = lon;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
mylat = lat;
myradarnumber = numradar;
}
```

4.5.2 ClientSendData โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ

```
public void ClientSendData(String mydata){

try {
byte tdata[] =  mydata.getBytes();
sendpacket = new DatagramPacket(tdata, tdata.length, InetAddress.getLocalHost(), socketadd);
socket.send(sendpacket);
//System.out.println("Send data address = " + socketadd + "  data =" + mydata);
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}
```

4.5.3 openfile โปรแกรมย่อยเปิดแฟ้มข้อมูลของเรดาร์

```
private void openfile() {
String filename;
filename = "c:/Thesis/Radar/Flight/ProtocolRadar" + myradar + ".PTC";
try {
input = new RandomAccessFile(filename, "r" );
}
catch ( IOException e ) {
System.err.println( "File not opened properly\n" + e.toString() );
this.stop();
}
}
```

4.5.4 closeFile โปรแกรมย่อยปิดแฟ้มข้อมูลของเรดาร์

```
private void closeFile()
{
try {
input.close();
}
catch ( IOException e ) {
System.err.println( "Error closing file\n ");
}
this.stop();
}
```

4.5.5 openport โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล

```
public void openport(int myadd) {
try {
socketadd = myadd;
socket = new DatagramSocket();
//Simsocket = new Socket(InetAddress.getLocalHost(), myadd);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
//System.out.println("Client socket be created address = " + socketadd);
}
catch ( IOException se) {
se.printStackTrace();
this.stop();
}
}
```

4.5.6 readprotocol โปรแกรมย่อยอ่านข้อมูลจากแฟ้มข้อมูล

```
public String readprotocol()
{
String data="";
// input the values from the file
try {
data = input.readLine();
if (data != null) {
return(data.trim());
}
closeFile();
}
catch ( EOFException eof ) {
System.err.println( "****Error read EOF from file\n");
closeFile();
return(null);
}
catch ( IOException e ) {
System.err.println( "****Error during read from file\n");
closeFile();
return(null);
}
return(null);
}
```

4.5.7 readprotocollong โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นตัวเลข

```
public long readprotocollong()
{
String data="";
long mylong;

//int test;
// input the values from the file
try {
data = input.readLine();
if (data != null) {
data = data.trim();
Long temp = new Long(data);
mylong =  temp.longValue();
return(mylong);
}
}
catch ( EOFException eof ) {
System.err.println( "****Error read EOF from file\n");
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
closeFile();
return(0);
}
catch ( IOException e ) {
System.err.println( "*****Error during read from file\n");
closeFile();
return(0);
}
return(0);
}
```

4.5.8 readprotocolstring โปรแกรมย่อยแปลงข้อมูลเรดาร์ให้เป็นอักขระ

```
public String readprotocolstring()
{
String data="";

// input the values from the file
try {
data = input.readLine();
return(data.substring(1,2));
}
catch ( EOFException eof ) {
System.err.println( "*****Error read EOF from file\n");
closeFile();
return("");
}
catch ( IOException e ) {
System.err.println( "*****Error during read from file\n");
closeFile();
return("");
}
}
```

4.5.9 run โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

```
public void run() {
long timecount;
long i;
int  account;
long starttime =0;
long sleeptime;
long endtime;
long timetosleep;

String protocolout;
String noprotocolout = new String("*#");
noprotocolout += myradar + "#";

openport(3000 + myradar);

dataout = myradarnumber + "#" + myradar + "#" + mylon + "#" + mylat;
// send radar position
ClientSendData(dataout);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
openfile();
System.out.println("open file protocol # " + myradar + " OK!");
timecount = 0;
while(true) {

protocolout = "";
//System.out.println("Time count  # " + myradar + " = " + timecount);
radartime = readprotocollong();
timecount += 1;

// end of data

if  (radartime == 0) {
while(true) {
ClientSendData(noprotocolout);
System.out.println("Radar  # " + myradar + " *** Sleep at Time count # " + timecount + " No Data to send  ***");
try{
this.sleep(newradar.sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "*****Sleep error!\n");
this.stop();
}
System.out.println("Radar  # " + myradar + " *** Weak up at Time count # " + timecount + " No Data to send  ***");

}  // end of while

}

//System.out.println("Time count  # " + myradar + " = " + timecount);
//timetosleep =  (radartime-timecount) * 4000 ;
//System.out.println("Time count  # " + myradar + " = " + timecount + " time to  sleep = " + timetosleep);

if (timecount == radartime) {  // save satrtime
Date startdate = new Date();
starttime = startdate.getTime();
}  // end start time

protocolout  =  readprotocol();
Integer tempaccount =  new Integer(protocolout);
account = tempaccount.intValue();
protocolout += "#";
for (i=1; i <= account; i++){
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
protocolout += readprotocol() + "#";
String type =  readprotocolstring();
protocolout += type + "#";
protocolout += readprotocol() + "#";
}
```

```
int strlen = protocolout.length();
protocolout = "*" + String.valueOf(strlen) + "#" + protocolout;
if (timecount < radartime) {
while (timecount < radartime) {  // Begin while
ClientSendData(noprotocolout);
System.out.println("Radar  # " + myradar + " *** Sleep at Time count #" + timecount);
try{
this.sleep(newradar.sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "*****Sleep error!\n");
this.stop();
}
System.out.println("Radar  # " + myradar + " *** Wake up Time count #" + timecount);
timecount  += 1;
}   // end while
Date startdate = new Date();
starttime = startdate.getTime();
}   // end if


ClientSendData(protocolout);
//System.out.println("Protocol Out " + protocolout);
Date enddate = new Date();
endtime = enddate.getTime();
sleeptime = newradar.sleeptime + starttime - endtime;
//System.out.println("Starttime =  " + starttime + " EndTime = " + endtime);
try{
this.sleep(sleeptime);
}
catch ( InterruptedException e) {
System.err.println( "*****Sleep error!\n");
this.stop();
}

} // end while
}  // end function
```

4.6 syncronize.java เตรียมข้อมูลเพื่อการขจัดความซ้ำซ้อน

4.6.1 syncronize เตรียมข้อมูลเพื่อการขจัดความซ้ำซ้อน

```
public class syncronize extends Thread{
private DatagramPacket receivepacket;
private DatagramSocket socket;
private DatagramSocket revsocket, sendsocket;
private int portadd;
private String str;

public syncronize() {
}
```

4.6.2 openport โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล

```
private void openport() { // Begin received data
portadd = 2000;
try {
revsocket = new DatagramSocket(portadd);
}
catch (IOException e) {
System.out.println(e.toString());
e.printStackTrace();
}
}  // End open port
```

4.6.3 storedata โปรแกรมย่อยเก็บข้อมูลของเรดาร์

```
// store radar location to array


public void storeradar(String strptc){  // begin store radar
int i = 0;
int s = 0;
int e = 0;
int radarnum;
int strlen = strptc.length();
long lon, lat;


// Find radar number
while (strptc.charAt(i) != '#') {  // while not end of string to find #
i++;
} // end while
i++;
s = i;
while (strptc.charAt(i) != '#') {  // while not end of string to find #
i++;
} // end while
e = i;
i++;
Integer tempradarnum =  new Integer(strptc.substring(s,e));
radarnum = tempradarnum.intValue();
//System.out.println("Radar num = " + radarnum);


// Find lontitude
s = i;
while (strptc.charAt(i) != '#') {  // while not end of string to find #
i++;
} // end while
e = i;
i++;
Long templon =  new Long(strptc.substring(s,e));
lon = templon.longValue();
//System.out.println("Radar lon = " + lon);
s = i;
Long templat =  new Long(strptc.substring(s,strlen));
lat = templat.longValue();
newradar.radarlon[radarnum] = lon;
newradar.radarlat[radarnum] = lat;
//System.out.println("Store radar # " + radarnum +"  lon = " + newradar.radarlon[radarnum] + " lat = " + newradar.radarlat[radarnum]);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
} //end stroe radar data
```

4.6.4 storeempty โปรแกรมย่อยลบข้อมูลทั้งหมด

```
private void storeempty(String strptc) {  // begin story for data for no data
int e ,s;
int radarnum;
e = strptc.length() -1 ;
Integer myradarnum =  new Integer(strptc.substring(e-1,e));
radarnum = myradarnum.intValue();

// clear data for new data
int m = 50 * (radarnum-1);
// clear data for new data
for (int k=0; k<50; k++) { // Begin for
newradar.ptcaircraftno[m+k] = 0;
newradar.ptcaircrafttime[m+k] = 0;
newradar.ptcClass[m+k] = "";
newradar.ptclon[m+k] = 0;
newradar.ptclat[m+k] = 0;
newradar.ptcradarnumber[m+k] = 0;
newradar.ptcradartheta[m+k] = 0;
newradar.ptcspeedlon[m+k] =0;
newradar.ptcspeedlat[m+k]= 0;
} //end for
}
```

4.6.5 waitforpacket โปรแกรมย่อยรับข้อมูลมาเก็บไว้เพื่อทำการส่งต่อ

```
 private void waitforpacket() {
   try {
     byte revdata[] =  new byte[8600];
     receivepacket = new DatagramPacket(revdata, revdata.length, InetAddress.getLocalHost(), portadd);
     revsocket.receive(receivepacket);
     str =  new String(receivepacket.getData());
     str = str.trim();
     //System.out.println("Syn data = " + str);
   }
   catch (IOException e) {
     System.out.println(e.toString());
     e.printStackTrace();
   }
 }
```

4.6.6 run โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

```
 // Run thread

 public void run(){ // Begin run tread
   openport();

   // wait for data to syncronize

   while (true) {  // Begin endless loop
     waitforpacket();
```

```
    //  store data to array

    while (newradar.consume == 1) {
     try   {
       //System.out.println("Wait to produce data");
       this.sleep(10);
     }
     catch ( InterruptedException e) {
       System.err.println( "*****Sleep error!\n");
       this.stop();
     }
    }   // end while wait for consum data
    newradar.produce = 1;
    //System.out.println("Start to Produce data");
    if ((str.charAt(0) == '*') && (str.charAt(1) == '#'))
      storeempty(str);
    else if (str.charAt(0) != '*')
      storeradar(str);
    else
      storedata(str);

    //System.out.println("End Produce data");
    newradar.produce = 0;
  }  // End endless loop
  //syncronizedata syndata = new syncronizedata();
  //syndata.start();
}   // End run tread
```

4.7 syncronizedata.java ขจัดความซ้ำซ้อนของข้อมูลเรดาร์

4.7.1 syncronizedata ขจัดความซ้ำซ้อนของข้อมูลเรดาร์

```
public class syncronizedata extends Thread{

 private int adjspeech = 0;
 private int adjposition = 20;
 private int acid = 0;
 private int myfirst = 1;
 private DatagramPacket sendpacket;
 private DatagramSocket socket;
 private int socketadd;
 private String PTCData;
 realtable activeac  = new realtable();
 repeatedtable repeattable = new  repeatedtable();
```

4.7.2 adddata โปรแกรมย่อยเก็บข้อมูลของเรดาร์

```
 public void assignid() {
   int found = 0;
   int id =0;
   int k,l;
   for (int j=0; j<repeattable.rindex; j++)   {
     k = activeac.getindex(repeattable.radarnum1[j],repeattable.aircraftno1[j]);
     l = activeac.getindex(repeattable.radarnum2[j],repeattable.aircraftno2[j]);
     if  ((activeac.acindex[k] == 0) && (activeac.acindex[l] == 0)) {
```

```
    id = activeac.getid();
    System.out.println("id = " + id);
    activeac.acindex[k] = id;
    activeac.acindex[l] = id;
    activeac.active[k] =  repeattable.active1[j];
    activeac.active[l] =  repeattable.active2[j];
    System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
   }  // end if

   else if  ((activeac.acindex[k] != 0) && (activeac.acindex[l] == 0)) {
    id = activeac.acindex[k];
    activeac.acindex[l] = id;
    activeac.active[k] =  repeattable.active1[j];
    activeac.active[l] =  repeattable.active2[j];
    System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
   }  // end if

   else if  ((activeac.acindex[k] == 0) && (activeac.acindex[l] != 0)) {
    id = activeac.acindex[l];
    activeac.acindex[k] = id;
    activeac.active[k] =  repeattable.active1[j];
    activeac.active[l] =  repeattable.active2[j];
   }  // end if
   //System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
  }  // end for i


  for (k=0; k<activeac.getmaxindex(); k++)  {
   if (activeac.acindex[k] == 0)
     activeac.addnewid(k);
     //activeac.active[k] = 1;
  } // end for
 }  // end function
```

4.7.3 assignid โปรแกรมย่อยกำหนดหมายเลขของอากาศยาน

```
 public void assignid() {
  int found = 0;
  int id =0;
  int k,l;
  for (int j=0; j<repeattable.rindex; j++)   {
   k = activeac.getindex(repeattable.radarnum1[j],repeattable.aircraftno1[j]);
   l = activeac.getindex(repeattable.radarnum2[j],repeattable.aircraftno2[j]);
   if  ((activeac.acindex[k] == 0) && (activeac.acindex[l] == 0)) {
    id = activeac.getid();
    System.out.println("id = " + id);
    activeac.acindex[k] = id;
    activeac.acindex[l] = id;
    activeac.active[k] =  repeattable.active1[j];
    activeac.active[l] =  repeattable.active2[j];
    System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
   }  // end if

   else if  ((activeac.acindex[k] != 0) && (activeac.acindex[l] == 0)) {
```

```
      id = activeac.acindex[k];
      activeac.acindex[l] = id;
      activeac.active[k] =  repeattable.active1[j];
      activeac.active[l] =  repeattable.active2[j];
      System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
     }  // end if

     else if  ((activeac.acindex[k] == 0) && (activeac.acindex[l] != 0)) {
      id = activeac.acindex[l];
      activeac.acindex[k] = id;
      activeac.active[k] =  repeattable.active1[j];
      activeac.active[l] =  repeattable.active2[j];
     }  // end if
     //System.out.println("Active1 = " + repeattable.active1[j] + " Active2 = " + repeattable.active2[j]);
   }  // end for i


   for (k=0; k<activeac.getmaxindex(); k++)  {
     if (activeac.acindex[k] == 0)
       activeac.addnewid(k);
       //activeac.active[k] = 1;
   } // end for
 }  // end function
```

4.7.4 ClientSendData โปรแกรมย่อยส่งข้อมูลผ่านช่องส่งสัญญาณ

```
public void ClientSendData(String mydata){

try {
    byte tdata[] =  mydata.getBytes();
    // connect by name change here
    //InetAddress address = InetAddress.getByName("NUKOOL1") ;
    //sendpacket = new DatagramPacket(tdata, tdata.length, address, socketadd);
    sendpacket = new DatagramPacket(tdata, tdata.length, InetAddress.getLocalHost(), socketadd);
    socket.send(sendpacket);
    //System.out.println("Send data address = " + socketadd + "  data =" + mydata);
  }
  catch (IOException e) {
    System.out.println(e.toString());
    e.printStackTrace();
  }
}
```

4.7.5 findclass โปรแกรมย่อยเปรียบเทียบชนิดของอากาศยาน

```
private int findclass(String firstclass, String secondclass) {  // start function
  if (firstclass.equals(secondclass))
    return 10;
  else
    return 0;
} // end function
```

4.7.6 finddirection โปรแกรมย่อยเปรียบเทียบทิศทางของอากาศยาน

```
private int finddirection(long fdirecx, long fdirecy, long sdirecx, long sdirecy) {  // start function
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
  int count = 0;
  if ((fdirecx >= 0) && (sdirecx >= 0))
    count += 10;
  if ((fdirecx <= 0) && (sdirecx <= 0))
    count += 10;
  if ((fdirecy >= 0) && (sdirecy >= 0))
    count += 10;
  if ((fdirecy <= 0) && (sdirecy <= 0))
    count += 10;

  if (count ==  20)
    return 10;
  else
    return 0;
 } // end function
```

4.7.7 findposition โปรแกรมย่อยเปรียบเทียบตำแหน่งของอากาศยาน

```
 private long findposition(int i, int j, int r1, int r2) {// start function
  long pox1, poy1;
  double acradius, timediff;

  timediff =   newradar.ptcaircrafttime[i] - newradar.ptcaircrafttime[j];
  pox1 = (long)(newradar.ptclon[i] -  (newradar.ptcspeedlon[i] * timediff )/3600);
  poy1 = (long)(newradar.ptclat[i] -  (newradar.ptcspeedlat[i] * timediff )/3600);
  //System.out.println("Time " +  newradar.ptcaircrafttime[i] + " = " + newradar.ptcaircrafttime[j] + " Diff = " + timediff  );
  //System.out.println(newradar.ptclon[i]+ " = " + pox1 + "  = " +  newradar.ptclon[j] + " | " + newradar.ptclat[i]+ " = " + poy1 + "  = " +  newradar.ptclat[j] + "
lat = " + (newradar.ptcspeedlat[i] * timediff /3600));
  if (((pox1 + adjposition >= newradar.ptclon[j]) && (pox1-adjposition <= newradar.ptclon[j])) && ((poy1 +adjposition >= newradar.ptclat[j]) && (poy1 -
adjposition <= newradar.ptclat[j]))) {

    //System.out.println(newradar.ptclon[i]+ " = " + pox1 + "  = " +  newradar.ptclon[j] + " | " + newradar.ptclat[i]+ " = " + poy1 + "  = " +  newradar.ptclat[j] + "
lat = " + (newradar.ptcspeedlat[i] * timediff /3600));
    return 10;

  }
  else
    return 0;
 } // end function
```

4.7.8 findradius โปรแกรมย่อยคำนวณตำแหน่งของอากาศยาน

```
 private double findradius(long rx, long ry, long fx, long fy) {
  double delx, dely, theta, actan, radius;

  delx = fx- rx;
  dely = fy - ry;
  // protect divide by zero
  if (dely == 0)
    theta = 360;
  else
    theta = Math.abs(Math.atan(delx/dely));
  radius = theta / 3.141592654 * 180;
  // Adjust radius value
```

```
  if ((delx < 0 ) && (dely > 0))
    radius = 360 - radius;
  else if ((delx < 0 ) && (dely < 0))
    radius = 180 + radius;
  else if ((delx > 0 ) && (dely < 0))
    radius = 180 - radius;
  return (radius);
 }
```

4.7.9 findrepeat โปรแกรมย่อย

```
 private void findrepeat() {
  int r,i,j,k,m,n;
  int totalscore = 0;
  for (r = 1; r < newradar.nradar-1; r++) {
    n = 50 * (r-1);
    //System.out.println("@@@@@@@@@@@@@@@@@@   Radar Number = " + r);
    for (i = 0; i < 50; i++) {
      //n = 50 * i;
      //System.out.println(" Class = " + newradar.ptcClass[n+i]);
      if (newradar.ptcClass[n+i] != "") {
        //System.out.println(" Class = " + newradar.ptcClass[n+i]);
      //m = 50 * r;
      for (j = r; j < newradar.nradar; j++) {
        m = 50 * j;
        for (k = 0; k < 50; k++) {

          if (newradar.ptcClass[m+k] != "") {
            //System.out.println("index = " + (m+k) + "  i = " + i);
            totalscore = 0;
            totalscore += findclass(newradar.ptcClass[n+i], newradar.ptcClass[m+k]);
            //System.out.println(" Score = " + totalscore + " radar " + r + " Check data = " + (n+i) + " with data " + (m+k));

            totalscore += finddirection(newradar.ptcspeedlon[n+i], newradar.ptcspeedlat[n+i], newradar.ptcspeedlon[m+k], newradar.ptcspeedlat[m+k]);
            //System.out.println(" Score = " + totalscore + " radar " + r + " Check data = " + (n+i) + " with data " + (m+k));

            totalscore += findspeed(newradar.ptcspeedlon[n+i], newradar.ptcspeedlat[n+i], newradar.ptcspeedlon[m+k], newradar.ptcspeedlat[m+k]);
            //System.out.println("radar " + r + " Score = " + totalscore + " 1lon " + newradar.ptcspeedlon[n+i] + " 2lon " + newradar.ptcspeedlon[m+k] + " | 1lat " +
newradar.ptcspeedlat[n+i] + " 2lat " + newradar.ptcspeedlat[m+k]);

            totalscore += findposition(n+i, m+k, r, j);

            //totalscore += repeattable.checkrepeat(activeac.getindex((r-1),i));

            if (totalscore >= 40) {  // Begin if
              //System.out.println("###  Add repeated data for radar # " + r);

              if (newradar.ptcaircrafttime[n + i] >= newradar.ptcaircrafttime[m+k])
              {
                newradar.ptcactive[m+k] = 0;
                newradar.ptcactive[n+i] = 1;
              }
              else
              {
```

```
            newradar.ptcactive[m+k] = 1;
            newradar.ptcactive[n+i] = 0;
           }
           repeattable.addrepeat((r-1),i,j,k);

           System.out.println(" R1: " + (r-1) + " T1: " + i + " R2: " + j + " T2: "+k);
          } // end if update active track
          //System.out.println(" Score = " + totalscore + " radar " + r + " Check data = " + (n+i) + " with data " + (m+k));
         } // end if
        //System.out.println("Time detect Radar # " + i + " ac No " + newradar.ptcaircraftno[m+j] + " Time " + findtime(acradius) );
       }  // end k loop
      }  // end j loop
     } // end of if
    }  // end n loop
   }  // end i loop
  }  //  end function
```

4.7.10 findspeed โปรแกรมย่อยคำนวณความเร็วของอากาศยาน

```
 private int findspeed(long fspeedlon, long fspeedlat, long sspeedlon, long sspeedlat) {  // start function
  //System.out.println(fspeedlon + " = "  + sspeedlon  + " | " + fspeedlat + " = "  + sspeedlat);
  if (((fspeedlon + adjspeech >= sspeedlon) && (fspeedlon - adjspeech <= sspeedlon)) && ((fspeedlat + adjspeech >= sspeedlat)) && (fspeedlat -
adjspeech <= sspeedlat))
    return 10;
  else
    return 0;
 } // end function
```

4.7.11 findtime โปรแกรมย่อยคำนวณความเวลาของอากาศยาน

```
 private void findtime() {
 double acradius;
 int i,j,m;

  for (i = 1; i < newradar.nradar; i++) {
    //System.out.println("Retrive radar # " + i + "  lon = " + newradar.radarlon[i] + " lat = " + newradar.radarlat[i]);
    m = 50 * (i-1);
    for (j = 0; j < 50; j++) {
     if    (newradar.ptcClass[m+j] != "") {
       acradius = findradius(newradar.radarlon[i], newradar.radarlat[i], newradar.ptclon[m+j], newradar.ptclat[m+j]);
       newradar.ptcaircrafttime[m+j] = (acradius /360) * 4;
       //System.out.println("Time detect Radar # " + i + " ac No " + newradar.ptcaircraftno[m+j] + " Time " + newradar.ptcaircrafttime[m+j] );
     } // end if
    }  // end for

 }  //end for
 } // end function
```

4.7.12 inradius โปรแกรมย่อยคำนวณตำแหน่งของอากาศยานว่าอยู่ในรัศมีของเรดาร์หรือไม่

```
 private int inradius(double rx, double ry, double fx, double fy) {
   double delx, dely, theta, actan, radius;
   long ldelx, ldely;

   int result = 0;
```

```
  delx = fx- rx;
  dely = fy - ry;
  ldelx = (long)(delx);
  ldely = (long)(dely);
  radius = Math.sqrt((ldelx)^2 + (ldely)^2 );
  if (radius < 600)
    result = 1;
  else
    result = 0;
  return (result);
 }
```

4.7.13 makeprotocoldata โปรแกรมย่อยสร้างโปรโตคอลของเรดาร์

```
 private void makeprotocoldata()  {
  int totalac, acnum;
  long systemtime;
  String tempstr;
  PTCData = "";
  acnum = 0;
  Date sysdate = new Date();
  systemtime = sysdate.getTime();

  //System.out.println("@@@@@@@@@@@@@"+activeac.getmaxindex());
  for (int i=0; i<=activeac.getmaxindex(); i++) {
   //if ((activeac.getactive(i) == 1) && (activeac.state[i] != 1) ) {
   if (activeac.active[i] == 1)  {
     acnum = acnum + 1;
     tempstr = "";
     if (activeac.acindex[i] < 10)
       tempstr =  "00" + activeac.acindex[i];
     else if (activeac.acindex[i] < 100)
       tempstr =  "0" + activeac.acindex[i];
     PTCData += "#" + acnum + "#" +  activeac.lon[i] + "#" +  activeac.lat[i];
     PTCData += "#" +  activeac.slon[i] + "#" +  activeac.slat[i];
     PTCData += "#" + tempstr + "#" + activeac.Class[i];
   }  // end for i
   activeac.state[i] = 2;
  }  // end if
  if (acnum == 0)
    PTCData = "*#"  + systemtime + "#";
  else
    PTCData = "*" + acnum + PTCData + "#" + systemtime + "#";
  System.out.println(PTCData);
 } // end function
```

4.7.14 missingdata โปรแกรมย่อยปรับค่าการซ้ำซ้อนของอากาศยาน

```
 public int missingdata(int k){
  int inradar=0;
  int result=0;
  int radius359=0;
  int samepoint=0;
  int found =0;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
  double temp =0, t=0;
    for (int f=0; f<newradar.nradar; f++)   {
   if ((result = inradius(newradar.radarlon[f], newradar.radarlat[f], activeac.lon[k], activeac.lat[k])) == 1){
    inradar =1;
    //System.out.println("  @@@@@@@@@@@ found @@@@@@@@@@");
   }
   if ((temp = findradius(newradar.radarlon[f], newradar.radarlat[f], activeac.lon[k], activeac.lat[k])) > 359){
    radius359=1;
    t = temp;
    //System.out.println("  @@@@@@@@@@@ found   radius  = " +temp + " State = " + activeac.state[k]);
   }
   else
   {
    //System.out.println("  @@@@@@@@@@@ radius = " + temp);
   }
  }

  if ((inradar == 1) && (radius359 == 1) && (activeac.state[k] == 2) ) {
   activeac.state[k] = 3;
   found = 1;
   activeac.lon[k] =  activeac.lon[k] +  activeac.slon[k]/2700;
   activeac.lat[k] =  activeac.lat[k] +  activeac.slat[k]/2700;
   System.out.println("  @@@@@@@@@@@ found   radius  = " + t);
  }  // end if
  return (found);
}
```

4.7.15 openport โปรแกรมย่อยเปิดช่องสัญญาณเพื่อส่งข้อมูล

```
public void openport(int myadd) {
  try {
    socketadd = myadd;
    socket = new DatagramSocket();
    //Simsocket = new Socket(InetAddress.getLocalHost(), myadd);
    //System.out.println("Client socket be created address = " + socketadd);
  }
  catch ( IOException se) {
    se.printStackTrace();
    this.stop();
  }
}
```

4.7.16 sendradarposition โปรแกรมย่อยส่งตำแหน่งของเรดาร์

```
private void sendradarposition() {  // begin Send radar position
  PTCData = String.valueOf(newradar.nradar);
  for (int i = 1; i < newradar.nradar; i++) {   // begin if
    PTCData +=  "#" +  i + "#" + newradar.radarlon[i] + "#" + newradar.radarlat[i];
  }  // end if
}   // end Send radar position
```

4.7.17 run โปรแกรมวงวนไม่จุดจบเพื่อส่งข้อมูล

```
public void run(){ // Begin run tread
  long starttime, endtime, sleeptime;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
openport(4000);
try   {
        this.sleep(newradar.sleeptime);
        sendradarposition();
        ClientSendData(PTCData);
  }
catch ( InterruptedException e) {
        System.err.println( "*****Sleep error!\n");
        this.stop();
  }

/*
try   {
        this.sleep(newradar.sleeptime);
  }
catch ( InterruptedException e) {
        System.err.println( "*****Sleep error!\n");
        this.stop();
  }
*/

while (true) {  // Begin endless loop
  try{
        //findrepeat();
        //System.out.println("*********test");


        while (newradar.produce == 1) {
          try   {
            //System.out.println(" Wait to consume data");
            this.sleep(10);
          }
          catch ( InterruptedException e) {
            System.err.println( "*****Sleep error!\n");
            this.stop();
          }
        }    // end while wait for consum data

        //System.out.println("**** End to Produce data");
        Date startdate = new Date();
        starttime = startdate.getTime();
        newradar.consume = 1;
        //adddata();
        findtime();
        findrepeat();
        adddata();
        assignid();

        makeprotocoldata();
        ClientSendData(PTCData);
        repeattable.clear();
        newradar.consume = 0;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        Date wakeupdate = new Date();
        endtime = wakeupdate.getTime();
        sleeptime = newradar.sleeptime + starttime - endtime;
        //System.out.println("@@@@@@@@@@Sleep time = " + sleeptime);
        this.sleep(sleeptime);
      }
      catch ( InterruptedException e) {
        System.err.println( "*****Sleep error!\n");
        this.stop();
      }
    }  // End endless loop
  }   // End run tread
```

5.1 OldMonitor.frm สร้างแผนที่ในระบบแสดงผลแบบเก่า

5.1.1 Form_Load โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์

```
Private Sub Form_Load()
    Start = False
    Dim k
    For k = 1 To 400 Step 1
        Load Label1(k)
        Load Label2(k)
        Load Shape1(k)
        Load Shape2(k)
    Next k
    Close #3
    Open "C:\Thesis\Radar\Flight\totalaircraft.tot" For Input As #3

    TotalOldDisplay = 0
    TotalOldReal = 0

    TotalBox1 = 0
    TotalBox2 = 0
    TotalLabel1 = 0
    TotalLabel2 = 0

    With tcpServer1
        ' IMPORTANT: be sure to change the RemoteHost
        ' value to the name of your computer.
        .RemoteHost = tcpServer1.LocalHostName
        .RemotePort = 2024   ' Port to connect to.
        .Bind 1001           ' Bind to the local port.
    End With
    With tcpServer2
        ' IMPORTANT: be sure to change the RemoteHost
        ' value to the name of your computer.
        .RemoteHost = tcpServer2.LocalHostName
        .RemotePort = 2024   ' Port to connect to.
        .Bind 1002           ' Bind to the local port.
    End With
    With tcpServer3
        ' IMPORTANT: be sure to change the RemoteHost
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    ' value to the name of your computer.
    .RemoteHost = tcpServer3.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1003              ' Bind to the local port.
End With
With tcpServer4
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer4.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1004          ' Bind to the local port.
End With

With tcpServer5
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer5.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1005          ' Bind to the local port.
End With

With tcpServer6
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer6.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1006          ' Bind to the local port.
End With

With tcpServer7
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer7.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1007          ' Bind to the local port.
End With

With tcpServer8
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer8.LocalHostName
    .RemotePort = 2024   ' Port to connect to.
    .Bind 1008           ' Bind to the local port.
End With

Move 0, 0
Dim i, j
MyFirst1 = True
MyFirst2 = True
MyFirst3 = True
MyFirst4 = True
RadarName = ""
i = 1
Do While (i <= 200)
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        j = 1
        Do While (j <= 32)
            TrackInfo(i, j).ACType = "X"
            j = j + 1
        Loop
        i = i + 1
    Loop
    FindBL_URGEO ("0950000E0050000N1110000E0210000N")
    MyScale = 1
    'RadarCount = 5
    TrackDataCount = 1
    CurrTrack = True
    TrackNo = 1
    MyViewScale = True
    MyMode = "NON"
    'ViewScale.Checked = True
    'ViewNoScale.Checked = True
    ViewModify1.Checked = True
    ModifyClick = False
    PlotMap
End Sub
```

5.1.2 Timer1_Timer

โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน

```
Private Sub Timer1_Timer()
    Dim MySignD, MySignR, MySign1, MySign2, MySign3, MySign4
    Dim AllDisplay
    If (Start = True) Then
        If (Not EOF(3)) Then
            Line Input #3, RadarText
            OldStat.Text7 = Val(Trim(RadarText))
            AllDisplay = TrackDraw(1, 1).MaxTrack + TrackDraw(2, 1).MaxTrack + TrackDraw(3, 1).MaxTrack + TrackDraw(4, 1).MaxTrack
            Line Input #3, RadarText

            If (OldStat.Text2 < TrackDraw(1, 1).MaxTrack) Then
                MySign1 = "+"
            ElseIf (OldStat.Text2 = TrackDraw(1, 1).MaxTrack) Then
                MySign1 = " "
            ElseIf (OldStat.Text2 > TrackDraw(1, 1).MaxTrack) Then
                MySign1 = "-"
            End If

            If (OldStat.Text3 < TrackDraw(2, 1).MaxTrack) Then
                MySign2 = "+"
            ElseIf (OldStat.Text3 = TrackDraw(2, 1).MaxTrack) Then
                MySign2 = " "
            ElseIf (OldStat.Text3 > TrackDraw(2, 1).MaxTrack) Then
                MySign2 = "-"
            End If

            If (OldStat.Text4 < TrackDraw(3, 1).MaxTrack) Then
                MySign3 = "+"
```

```
        ElseIf (OldStat.Text4 = TrackDraw(3, 1).MaxTrack) Then
            MySign3 = " "
        ElseIf (OldStat.Text4 > TrackDraw(3, 1).MaxTrack) Then
            MySign3 = "-"
        End If

        If (OldStat.Text5 < TrackDraw(4, 1).MaxTrack) Then
            MySign4 = "+"
        ElseIf (OldStat.Text5 = TrackDraw(4, 1).MaxTrack) Then
            MySign4 = " "
        ElseIf (OldStat.Text5 > TrackDraw(4, 1).MaxTrack) Then
            MySign4 = "-"
        End If

        If (Val(Trim(OldStat.Text1)) < AllDisplay) Then
            MySignD = "+"
        ElseIf (Val(Trim(OldStat.Text1)) = AllDisplay) Then
            MySignD = " "
        ElseIf (Val(Trim(OldStat.Text1)) > AllDisplay) Then
            MySignD = "-"
        End If

        If (OldStat.Text6 < Val(Trim(RadarText))) Then
            MySignR = "+"
        ElseIf (OldStat.Text6 = Val(Trim(RadarText))) Then
            MySignR = " "
        ElseIf (OldStat.Text6 > Val(Trim(RadarText))) Then
            MySignR = "-"
        End If

        OldStat.Text6 = Val(Trim(RadarText))
        OldStat.Text1 = AllDisplay
        OldStat.Text2 = TrackDraw(1, 1).MaxTrack
        OldStat.Text3 = TrackDraw(2, 1).MaxTrack
        OldStat.Text4 = TrackDraw(3, 1).MaxTrack
        OldStat.Text5 = TrackDraw(4, 1).MaxTrack
        Write #2, MySign1 & TrackDraw(1, 1).MaxTrack, MySign2 & TrackDraw(2, 1).MaxTrack, MySign3 & TrackDraw(3, 1).MaxTrack, MySign4 & TrackDraw
(4, 1).MaxTrack, MySignD & Val(Trim(OldStat.Text1)), MySignR & Val(OldStat.Text6)
      End If
   End If
   RedrawF
End Sub
```

5.1.3 PlotMap โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
Private Sub PlotMap()
   Dim i
   Cls
   PoltThaiBorder
   PoltNeighborBorder
End Sub
```

5.1.4 DrawFlight โปรแกรมย่อยแสดงอากาศยานเริ่มต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Public Sub DrawFlight()
    Dim x, Y, X2, Y2
    Dim i
    i = 1
    If MyClass = "F" Then
        ForeColor = &HFF00&
    Else
        ForeColor = &HFF&
    End If
    'X = (Second(Val(PointDetail.Text1.Text), Val(PointDetail.Text2.Text), Val(PointDetail.Text3.Text)) _
    '- Second(BLLonDeg, BLLonMin, BLLonSec))
    'Y = Second(URLatDeg, URLatMin, URLatSec) - Second(Val(PointDetail.Text4.Text), Val(PointDetail.Text5.Text), Val(PointDetail.Text6.Text))
    'Circle (X * SCX, Y * SCY), 30

    Circle ((TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY), 30

    'If TrackDataCount = CurrTrack Then
        If (TrackNo > 1) Then
            x = (TrackInfo(TrackDataCount, TrackNo - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo - 1).Lat) * SCY
            X2 = (TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY
            Line (x, Y)-(X2, Y2)
        End If
    'End If
    If Not CurrTrack Then
        TrackNo = 1
        TrackDataCount = TrackDataCount + 1
        CurrTrack = True
    Else
        TrackNo = TrackNo + 1
    End If
End Sub
```

5.1.5 ReDrawFlight โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง

```
Private Sub ReDrawFlight()
    Dim x, Y, X2, Y2
    Dim i, j
    i = 1
    j = 1
    Do While (TrackInfo(i, j).ACType <> "X")

        If TrackInfo(i, j).Class = "F" Then
            ForeColor = &HFF00&
        Else
            ForeColor = &HFF&
        End If
        Do While (TrackInfo(i, j).ACType <> "X")
            Circle ((TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
            (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY), 30

            If (j > 1) Then
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            x = (TrackInfo(i, j - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j - 1).Lat) * SCY
            X2 = (TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY
            Line (x, Y)-(X2, Y2)
          End If

          j = j + 1
        Loop
        i = i + 1
        j = 1
    Loop
End Sub
```

5.1.6 DrawRadarNuber โปรแกรมย่อยแสดงรัศมีของเรดาร์

```
Public Sub DrawRadarNuber(RadarNum As Integer)
    ForeColor = &HFFC0C0
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
       (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 4500 * SCX
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
       (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 30

End Sub
```

5.1.7 ReDrawRadar โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง

```
Public Sub ReDrawRadar()
    Dim i
    i = 1
    ForeColor = &HFFC0C0
    Do While (i < RadarCount)
       Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
       (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 4500 * SCX
       Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
       (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 30
       i = i + 1
    Loop
End Sub
```

5.1.8 GetRadarACPosition โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน

```
Private Sub GetRadarACPosition(strData As String)
    Dim MyPos, OldPos
    Dim RadarNume As Integer
    Dim FirstStr
    Dim TotalTrack
    Dim Lat, Lon, SpeedLon, SpeedLat
    Dim AcNo, RadarNum
    Dim TrackSign As String
    Dim i, x1, y1
    Dim x As Long, Y As Long
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
FirstStr = Mid(strData, 1, 1)
If (FirstStr <> "*") Then
    MyPos = 1
    MyPos = InStr(MyPos, strData, "#", 1)
    OldPos = MyPos
    RadarCount = Val(Mid(strData, 1, MyPos - 1))
    MyPos = InStr(MyPos + 1, strData, "#", 1)
    RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
    OldPos = MyPos
    MyPos = InStr(MyPos + 1, strData, "#", 1)
    RadarPosition(RadarNum, 1) = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 7.5
    OldPos = MyPos
    MyPos = InStr(MyPos + 1, strData, "#", 1)
    RadarPosition(RadarNum, 2) = Val(Mid(strData, OldPos + 1, Len(strData))) * 7.5
    DrawRadarNuber (RadarNum)
    'open statistic file

ElseIf (Mid(strData, 1, 2) = "*#") Then
    RadarNum = Val(Mid(strData, 3, Len(strData)))
    TrackDraw(RadarNum, 1).MaxTrack = 0
    Start = True

Else
    'MsgBox ("Data = " & strData)
    MyPos = 2
    MyPos = InStr(MyPos, strData, "#", 1)
    OldPos = MyPos
    MyPos = InStr(MyPos + 1, strData, "#", 1)
    TotalTrack = Val(Mid(strData, OldPos + 1, MyPos - 1))
    'MsgBox ("Track Count = " & TotalTrack)
    For i = 1 To TotalTrack Step 1
        'MsgBox ("i = " & i)
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        Lon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        Lat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SpeedLon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SpeedLat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        AcNo = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TrackSign = Trim(Mid(strData, OldPos + 1, 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        x = (Lon * 7.5 - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX \ 1
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        Y = (Second(URLatDeg, URLatMin, URLatSec) - Lat * 7.5) * SCY \ 1
        TrackDraw(RadarNum, i).MaxTrack = TotalTrack
        TrackDraw(RadarNum, i).TrackNo = AcNo
        TrackDraw(RadarNum, i).Lon = x
        TrackDraw(RadarNum, i).Lat = Y
        TrackDraw(RadarNum, i).Sign = TrackSign
      Next i
      Start = True
    End If
End Sub
```

5.1.9 RedrawF โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

```
Private Sub RedrawF()
 Dim i, j, k
   For k = 1 To 400 Step 1
      Label1(k).Visible = False
      Label2(k).Visible = False
      Shape1(k).Visible = False
      Shape2(k).Visible = False
   Next k
   'PlotMap
   'ReDrawRadar
   i = 1
   Do While (i < RadarCount)
      For j = 1 To TrackDraw(i, 1).MaxTrack Step 1
      If TrackDraw(i, j).Sign = "F" Then
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "F" & Format(TrackDraw(i, j).TrackNo, "000")
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
         Else
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "H" & Format(TrackDraw(i, j).TrackNo, "000")
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
         End If
      Next j
      i = i + 1
   Loop
   i = 1
   Do While (i < RadarCount)
      TrackDraw(i, 1).MaxTrack = 0
      i = i + 1
   Loop
End Sub
```

5.2 OldMonitor2.frm สร้างแผนที่ในระบบแสดงผลแบบเก่าที่ทำงานร่วมกับระบบใหม่

5.2.1 Form_Load โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์

```
Private Sub Form_Load()
   Start = False
   Dim k
   For k = 1 To 400 Step 1
      Load Label1(k)
      Load Label2(k)
      Load Shape1(k)
      Load Shape2(k)
   Next k
   'Close #3
   'Open "C:\Thesis\Radar\Flight\totalaircraft.tot" For Input As #3

   TotalOldDisplay = 0
   TotalOldReal = 0

   TotalBox1 = 0
   TotalBox2 = 0
   TotalLabel1 = 0
   TotalLabel2 = 0

   With tcpServer1
      ' IMPORTANT: be sure to change the RemoteHost
      ' value to the name of your computer.
      .RemoteHost = tcpServer1.LocalHostName
      .RemotePort = 2024   ' Port to connect to.
      .Bind 1001           ' Bind to the local port.
   End With
   With tcpServer2
      ' IMPORTANT: be sure to change the RemoteHost
      ' value to the name of your computer.
      .RemoteHost = tcpServer2.LocalHostName
      .RemotePort = 2024   ' Port to connect to.
      .Bind 1002           ' Bind to the local port.
   End With
   With tcpServer3
      ' IMPORTANT: be sure to change the RemoteHost
      ' value to the name of your computer.
      .RemoteHost = tcpServer3.LocalHostName
      .RemotePort = 2024  ' Port to connect to.
      .Bind 1003           ' Bind to the local port.
   End With
   With tcpServer4
      ' IMPORTANT: be sure to change the RemoteHost
      ' value to the name of your computer.
      .RemoteHost = tcpServer4.LocalHostName
      .RemotePort = 2024  ' Port to connect to.
      .Bind 1004           ' Bind to the local port.
   End With

   With tcpServer5
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer5.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1005          ' Bind to the local port.
End With

With tcpServer6
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer6.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1006          ' Bind to the local port.
End With

With tcpServer7
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer7.LocalHostName
    .RemotePort = 2024  ' Port to connect to.
    .Bind 1007          ' Bind to the local port.
End With

With tcpServer8
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer8.LocalHostName
    .RemotePort = 2024   ' Port to connect to.
    .Bind 1008          ' Bind to the local port.
End With

Move 0, 0
Dim i, j
MyFirst1 = True
MyFirst2 = True
MyFirst3 = True
MyFirst4 = True
RadarName = ""
i = 1
Do While (i <= 200)
    j = 1
    Do While (j <= 32)
        TrackInfo(i, j).ACType = "X"
        j = j + 1
    Loop
    i = i + 1
Loop
FindBL_URGEO ("0950000E0050000N1110000E0210000N")
MyScale = 1
'RadarCount = 5
TrackDataCount = 1
CurrTrack = True
TrackNo = 1
MyViewScale = True
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    MyMode = "NON"
    'ViewScale.Checked = True
    'ViewNoScale.Checked = True
    ViewModify1.Checked = True
    ModifyClick = False
    PlotMap

End Sub
```

5.2.2 Timer1_Timer โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน

```
Private Sub Timer1_Timer()
    Dim AllAc
    If (Start = True) Then
        If (Not EOF(3)) Then
            'Line Input #3, RadarText
            'Line Input #3, RadarText
            'AllStat.Text7 = Val(Trim(RadarText))

            AllAc = TrackDraw(1, 1).MaxTrack + TrackDraw(2, 1).MaxTrack + TrackDraw(3, 1).MaxTrack + TrackDraw(4, 1).MaxTrack

            If (AllStat.Text2 < TrackDraw(1, 1).MaxTrack) Then
                MySign1 = "+"
            ElseIf (AllStat.Text2 = TrackDraw(1, 1).MaxTrack) Then
                MySign1 = ""
            ElseIf (AllStat.Text2 > TrackDraw(1, 1).MaxTrack) Then
                MySign1 = "-"
            End If

            If (AllStat.Text3 < TrackDraw(2, 1).MaxTrack) Then
                MySign2 = "+"
            ElseIf (AllStat.Text3 = TrackDraw(2, 1).MaxTrack) Then
                MySign2 = ""
            ElseIf (AllStat.Text3 > TrackDraw(2, 1).MaxTrack) Then
                MySign2 = "-"
            End If

            If (AllStat.Text4 < TrackDraw(3, 1).MaxTrack) Then
                MySign3 = "+"
            ElseIf (AllStat.Text4 = TrackDraw(3, 1).MaxTrack) Then
                MySign3 = ""
            ElseIf (AllStat.Text4 > TrackDraw(3, 1).MaxTrack) Then
                MySign3 = "-"
            End If

            If (AllStat.Text5 < TrackDraw(4, 1).MaxTrack) Then
                MySign4 = "+"
            ElseIf (AllStat.Text5 = TrackDraw(4, 1).MaxTrack) Then
                MySign4 = ""
            ElseIf (AllStat.Text5 > TrackDraw(4, 1).MaxTrack) Then
                MySign4 = "-"
            End If
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        If (AllStat.Text7 < AllAc) Then
          MysignO = "+"
        ElseIf (AllStat.Text7 = AllAc) Then
          MysignO = ""
        ElseIf (AllStat.Text7 > AllAc) Then
          MysignO = "-"
        End If

        AllStat.Text7 = AllAc
        AllStat.Text2 = TrackDraw(1, 1).MaxTrack
        AllStat.Text3 = TrackDraw(2, 1).MaxTrack
        AllStat.Text4 = TrackDraw(3, 1).MaxTrack
        AllStat.Text5 = TrackDraw(4, 1).MaxTrack
        RadarStataStr = MysignO & AllAc & "  " & MySign1 & AllStat.Text2 & "  " & MySign2 & AllStat.Text3 & "  " & MySign3 & AllStat.Text4 & "  " & MySign4 &
AllStat.Text5 & "  "
      End If
    End If
    RedrawF
End Sub
```

5.2.3 PlotMap โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
Private Sub PlotMap()
    Dim i
    Cls
    PoltThaiBorder
    PoltNeighborBorder
End Sub
```

5.2.4 DrawFlight โปรแกรมย่อยแสดงอากาศยานเริ่มต้น

```
Private Sub ReDrawFlight()
    Dim x, Y, X2, Y2
    Dim i, j
    i = 1
    j = 1
    Do While (TrackInfo(i, j).ACType <> "X")

      If TrackInfo(i, j).Class = "F" Then
        ForeColor = &HFF00&
      Else
        ForeColor = &HFF&
      End If
      Do While (TrackInfo(i, j).ACType <> "X")
        Circle ((TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY), 30

        If (j > 1) Then
          x = (TrackInfo(i, j - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j - 1).Lat) * SCY
          X2 = (TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY
          Line (x, Y)-(X2, Y2)
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        End If

        j = j + 1
      Loop
      i = i + 1
      j = 1
    Loop
End Sub
```

5.2.5 ReDrawFlight โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง

```
Private Sub ReDrawFlight()
    Dim x, Y, X2, Y2
    Dim i, j
    i = 1
    j = 1
    Do While (TrackInfo(i, j).ACType <> "X")

      If TrackInfo(i, j).Class = "F" Then
        ForeColor = &HFF00&
      Else
        ForeColor = &HFF&
      End If
      Do While (TrackInfo(i, j).ACType <> "X")
        Circle ((TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY), 30

        If (j > 1) Then
          x = (TrackInfo(i, j - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j - 1).Lat) * SCY
          X2 = (TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY
          Line (x, Y)-(X2, Y2)
        End If

        j = j + 1
      Loop
      i = i + 1
      j = 1
    Loop
End Sub
```

5.2.6 DrawRadarNuber โปรแกรมย่อยแสดงรัศมีของเรดาร์

```
Public Sub DrawRadarNuber(RadarNum As Integer)
    ForeColor = &HFFC0C0
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 4500 * SCX
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 30

End Sub
```

5.2.7 ReDrawRadar โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Public Sub ReDrawRadar()
    Dim i
    i = 1
    ForeColor = &HFFC0C0
    Do While (i < RadarCount)
        Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 4500 * SCX
        Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 30
        i = i + 1
    Loop
End Sub
```

5.2.8 GetRadarACPosition

โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน

```
Private Sub GetRadarACPosition(strData As String)
    Dim MyPos, OldPos
    Dim RadarNume As Integer
    Dim FirstStr
    Dim TotalTrack
    Dim Lat, Lon, SpeedLon, SpeedLat
    Dim AcNo, RadarNum
    Dim TrackSign As String
    Dim i, x1, y1
    Dim x As Long, Y As Long

    FirstStr = Mid(strData, 1, 1)
    If (FirstStr <> "*") Then
        MyPos = 1
        MyPos = InStr(MyPos, strData, "#", 1)
        OldPos = MyPos
        RadarCount = Val(Mid(strData, 1, MyPos - 1))
        'MsgBox ("Message = " & strData & "Radar count = " & RadarCount & " mypos = " & MyPos & " olpos = " & OldPos)
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        RadarPosition(RadarNum, 1) = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 7.5
        OldPos = MyPos
        'MsgBox ("Message = " & strData & "lon = " & RadarPosition(1, 1))
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        RadarPosition(RadarNum, 2) = Val(Mid(strData, OldPos + 1, Len(strData))) * 7.5
        'MsgBox ("Message = " & strData & "lat = " & RadarPosition(1, 2))
        DrawRadarNuber (RadarNum)
        'open statistic file

    ElseIf (Mid(strData, 1, 2) = "*#") Then
        RadarNum = Val(Mid(strData, 3, Len(strData)))
        TrackDraw(RadarNum, 1).MaxTrack = 0
        Start = True

    Else
```

```
        'MsgBox ("Data = " & strData)
        MyPos = 2
        MyPos = InStr(MyPos, strData, "#", 1)
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TotalTrack = Val(Mid(strData, OldPos + 1, MyPos - 1))
        'MsgBox ("Track Count = " & TotalTrack)
        For i = 1 To TotalTrack Step 1
          'MsgBox ("i = " & i)
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          Lon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          Lat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          SpeedLon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          SpeedLat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          AcNo = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          TrackSign = Trim(Mid(strData, OldPos + 1, 1))
          OldPos = MyPos
          MyPos = InStr(MyPos + 1, strData, "#", 1)
          RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
          x = (Lon * 7.5 - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX \ 1
          Y = (Second(URLatDeg, URLatMin, URLatSec) - Lat * 7.5) * SCY \ 1
          'Line (x, Y)-(x1, y1), B
          TrackDraw(RadarNum, i).MaxTrack = TotalTrack
          TrackDraw(RadarNum, i).TrackNo = AcNo
          TrackDraw(RadarNum, i).Lon = x
          TrackDraw(RadarNum, i).Lat = Y
          TrackDraw(RadarNum, i).Sign = TrackSign
        Next i
        Start = True
      End If
End Sub
```

5.2.9 RedrawF โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

```
Private Sub RedrawF()
 Dim i, j, k
   For k = 1 To 200 Step 1
      Label1(k).Visible = False
      Label2(k).Visible = False
      Shape1(k).Visible = False
      Shape2(k).Visible = False
   Next k
   'PlotMap
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    'ReDrawRadar
    i = 1
    Do While (i < RadarCount)
       For j = 1 To TrackDraw(i, 1).MaxTrack Step 1
       If TrackDraw(i, j).Sign = "F" Then
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "F" & Format(TrackDraw(i, j).TrackNo, "000")
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
         Else
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "H" & Format(TrackDraw(i, j).TrackNo, "000")
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
         End If
       Next j
       i = i + 1
    Loop
    i = 1
    Do While (i < RadarCount)
       TrackDraw(i, 1).MaxTrack = 0
       i = i + 1
    Loop
End Sub
```

5.3 OldStat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงในระบบเก่า

```
Private Sub Form_Load()
    Dim FileNameMax As String
    Dim FileNameStat As String
    Dim RadarText As String
    Dim Max As Integer

    Left = 8000
    Top = 300
    Text1 = 0
    Text2 = 0
    Text3 = 0
    Text4 = 0
    Text5 = 0
    Text6 = 0
    Text7 = 0

    ' Read max index
    FileNameMax = App.Path & "/Radar/statistic/old.txt"
    Open FileNameMax For Input As #1
    Line Input #1, RadarText
    Max = Val(Trim(RadarText))
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    Close #1

    ' Write max index
    FileNameMax = App.Path & "/Radar/statistic/old.txt"
    Open FileNameMax For Output As #1
    Write #1, Max + 1
    Close #1

    ' Write out put
    FileNameStat = App.Path & "/Radar/statistic/old" & Max & ".txt"
    Open FileNameStat For Output As #2

End Sub

Public Sub UnloadMe()
    Close #2
    Close #3
    Unload Me
End Sub

Private Sub Form_Terminate()
    Close #2
    Close #3
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Close #2
    Close #3
End Sub
```

6.1 NewMonitor.frm สร้างแผนที่ในระบบแสดงผลเมื่อมีการขจัดความซ้ำซ้อน

6.1.1 Form_Load โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์

```
Private Sub Form_Load()
    Dim k
    Start = False
    For k = 1 To 400 Step 1
        Load Label1(k)
        Load Label2(k)
        Load Shape1(k)
        Load Shape2(k)
    Next k

    Close #3
    Open "C:\Thesis\Radar\Flight\totalaircraft.tot" For Input As #3

    TotalNewDisplay = 0
    TotalOldReal = 0
    TotalMiss = 0
    TotalReal = 0
    Counter = 0

    TotalTrack = 0
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    TotalBox1 = 0
    TotalBox2 = 0
    TotalLabel1 = 0
    TotalLabel2 = 0

    With tcpServer1
       ' IMPORTANT: be sure to change the RemoteHost
       ' value to the name of your computer.
       .RemoteHost = tcpServer1.LocalHostName
       .RemotePort = 4048   ' Port to connect to.
       .Bind 4000           ' Bind to the local port.
    End With

    Move 0, 0
    Dim i, j

    RadarName = ""
    i = 1
    Do While (i <= 200)
       j = 1
       Do While (j <= 32)
          TrackInfo(i, j).ACType = "X"
          j = j + 1
       Loop
       i = i + 1
    Loop
    FindBL_URGEO ("0950000E0050000N1110000E0210000N")
    MyScale = 1
    'RadarCount = 5
    TrackDataCount = 1
    CurrTrack = True
    TrackNo = 1
    MyViewScale = True
    MyMode = "NON"
    'ViewScale.Checked = True
    'ViewNoScale.Checked = True
    ViewModify1.Checked = True
    ModifyClick = False
    PlotMap

End Sub
```

6.1.2 Timer1_Timer โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน

```
Private Sub Timer1_Timer()
    Dim Miss As Integer
    Dim MySign
    If (Start = True) Then
       Counter = Counter + 1

       If (Not EOF(3)) Then
          Line Input #3, RadarText
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        NewStat.Text1 = Val(Trim(RadarText)) - 1
        Line Input #3, RadarText
        If (NewStat.Text2 < TrackDraw(1, 1).MaxTrack) Then
          MySign = "+"
        ElseIf (NewStat.Text2 = TrackDraw(1, 1).MaxTrack) Then
          MySign = ""
        ElseIf (NewStat.Text2 > TrackDraw(1, 1).MaxTrack) Then
          MySign = "-"
        End If
        NewStat.Text2 = TrackDraw(1, 1).MaxTrack
        NewStat.Text3 = Val(Trim(RadarText))
        Miss = Abs(TrackDraw(1, 1).MaxTrack - Val(Trim(RadarText)))
        TotalMiss = TotalMiss + Miss
        TotalReal = TotalReal + Val(Trim(RadarText))
        If (TotalReal <> 0) Then
          NewStat.Text4 = Format(((TotalReal - TotalMiss) / TotalReal), "00.00%")
        Else
          NewStat.Text4 = Format(0, "00.00%")
        End If
        Write #2, Val(Trim(NewStat.Text1)), Val(Trim(NewStat.Text3)), MySign & TrackDraw(1, 1).MaxTrack, TotalMiss, TotalReal, ((TotalReal - TotalMiss) /
TotalReal) * 100
      Else
        NewStat.Text3 = 0
        NewStat.Text2 = 0
        TotalReal = TotalReal
      End If
    End If
    RedrawF
End Sub
```

6.1.3 PlotMap โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
Private Sub PlotMap()
    Dim i
    Cls
    PoltThaiBorder
    PoltNeighborBorder
End Sub
```

6.1.4 DrawFlight โปรแกรมย่อยแสดงอากาศยานเริ่มต้น

```
Public Sub DrawFlight()
    Dim x, Y, X2, Y2
    Dim i
    i = 1
    If MyClass = "F" Then
      ForeColor = &HFF00&
    Else
      ForeColor = &HFF&
    End If
    'X = (Second(Val(PointDetail.Text1.Text), Val(PointDetail.Text2.Text), Val(PointDetail.Text3.Text)) _
    '- Second(BLLonDeg, BLLonMin, BLLonSec))
    'Y = Second(URLatDeg, URLatMin, URLatSec) - Second(Val(PointDetail.Text4.Text), Val(PointDetail.Text5.Text), Val(PointDetail.Text6.Text))
    'Circle (X * SCX, Y * SCY), 30
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    Circle ((TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY), 30

    'If TrackDataCount = CurrTrack Then
      If (TrackNo > 1) Then
        x = (TrackInfo(TrackDataCount, TrackNo - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
        Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo - 1).Lat) * SCY
        X2 = (TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
        Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY
        Line (x, Y)-(X2, Y2)
      End If
    'End If
    If Not CurrTrack Then
      TrackNo = 1
      TrackDataCount = TrackDataCount + 1
      CurrTrack = True
    Else
      TrackNo = TrackNo + 1
    End If
End Sub
```

6.1.5 ReDrawFlight โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง

```
Private Sub ReDrawFlight()
    Dim x, Y, X2, Y2
    Dim i, j
    i = 1
    j = 1
    Do While (TrackInfo(i, j).ACType <> "X")

      If TrackInfo(i, j).Class = "F" Then
        ForeColor = &HFF00&
      Else
        ForeColor = &HFF&
      End If
      Do While (TrackInfo(i, j).ACType <> "X")
        Circle ((TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY), 30

        If (j > 1) Then
          x = (TrackInfo(i, j - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j - 1).Lat) * SCY
          X2 = (TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
          Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY
          Line (x, Y)-(X2, Y2)
        End If

        j = j + 1
      Loop
      i = i + 1
      j = 1
    Loop
End Sub
```

6.1.6 DrawRadarNuber โปรแกรมย่อยแสดงรัศมีของเรดาร์

```
Public Sub DrawRadarNuber(RadarNum As Integer)
   ForeColor = &HFFC0C0
   Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 4500 * SCX
   Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 30

End Sub
```

6.1.7 ReDrawRadar โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง

```
Public Sub ReDrawRadar()
   Dim i
   i = 1
   ForeColor = &HFFC0C0
   Do While (i < RadarCount)
      Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 4500 * SCX
      Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
      (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 30
      i = i + 1
   Loop
End Sub
```

6.1.8 GetRadarACPosition โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน

```
Private Sub GetRadarACPosition(strData As String)
   Dim MyPos, OldPos
   Dim RadarNume As Integer
   Dim FirstStr
   Dim Lat, Lon, SpeedLon, SpeedLat, TrackNum, Order, SystemTime
   Dim AcNo, RadarNum
   Dim TrackSign As String
   Dim i, x1, y1, j
   Dim x As Long, Y As Long

   FirstStr = Mid(strData, 1, 1)
   If (FirstStr <> "*") Then
      MyPos = 1
      MyPos = InStr(MyPos, strData, "#", 1)
      OldPos = MyPos
      RadarCount = Val(Mid(strData, 1, MyPos - 1))
      'MsgBox ("Message = " & strData & "Radar count = " & RadarCount & " mypos = " & MyPos & " olpos = " & OldPos)
      For j = 1 To RadarCount - 1 Step 1
         MyPos = InStr(MyPos + 1, strData, "#", 1)
         RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
         OldPos = MyPos
         MyPos = InStr(MyPos + 1, strData, "#", 1)
         RadarPosition(RadarNum, 1) = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 7.5
         OldPos = MyPos
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        'MsgBox ("Message = " & strData & "lon = " & RadarPosition(1, 1))
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        RadarPosition(RadarNum, 2) = Val(Mid(strData, OldPos + 1, Len(strData))) * 7.5
        'MsgBox ("Message = " & strData & "lat = " & RadarPosition(1, 2))
        DrawRadarNuber (RadarNum)
        OldPos = MyPos
    Next j
    If (Not EOF(3)) Then
        Line Input #3, RadarText
        Line Input #3, RadarText
    End If

ElseIf (Mid(strData, 1, 2) = "*#") Then
    RadarNum = Val(Mid(strData, 3, Len(strData)))
    TrackDraw(1, 1).MaxTrack = 0
    Start = True
Else
    'MsgBox ("Data = " & strData)
    MyPos = 2
    MyPos = InStr(MyPos, strData, "#", 1)
    TotalTrack = Val(Mid(strData, 2, MyPos - 1))
    'MsgBox ("Track Count = " & TotalTrack)
    For i = 1 To TotalTrack Step 1
        'MsgBox ("i = " & i)
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        Order = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        'MsgBox ("order = " & Order)
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        Lon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        'MsgBox ("LON = " & Lon)
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        Lat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SpeedLon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SpeedLat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TrackNo = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TrackSign = Trim(Mid(strData, OldPos + 1, 1))

        x = (Lon * 7.5 - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX \ 1
        Y = (Second(URLatDeg, URLatMin, URLatSec) - Lat * 7.5) * SCY \ 1

        TrackDraw(1, Order).MaxTrack = TotalTrack
        TrackDraw(1, Order).TrackNo = TrackNo
        TrackDraw(1, Order).Lon = x
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            TrackDraw(1, Order).Lat = Y
            TrackDraw(1, Order).Sign = TrackSign
        Next i
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SystemTime = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        Start = True
    End If
End Sub
```

6.1.9 RedrawF โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

```
Private Sub RedrawF()
 Dim i, j, k
    For k = 0 To 400 Step 1
        Label1(k).Visible = False
        Label2(k).Visible = False
        Shape1(k).Visible = False
        Shape2(k).Visible = False
    Next k
    'PlotMap
    'ReDrawRadar
    i = 1

    For j = 1 To TrackDraw(i, 1).MaxTrack Step 1
        If TrackDraw(i, j).Sign = "F" Then
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "F" & Format(TrackDraw(i, j).TrackNo, "000")
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label1(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
        Else
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Shape2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Caption = "H" & Format(TrackDraw(i, j).TrackNo, "000")
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Top = TrackDraw(i, j).Lat + 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Left = TrackDraw(i, j).Lon - 50
            Label2(TrackDraw(i, j).TrackNo + ((i * 50) - 50)).Visible = True
        End If
    Next j
    TrackDraw(1, 1).MaxTrack = 0
End Sub
```

6.2 NewMonitor2.frm สร้างแผนที่ในระบบแสดงผลที่ทำงานร่วมกับระบบเก่า

6.2.1 Form_Load โปรแกรมย่อยเปิดช่องรับสัญญาณเพื่อรับข้อมูลเรดาร์

```
Private Sub Form_Load()
    Dim k
    Start = False
    For k = 1 To 400 Step 1
        Load Label1(k)
```

```
    Load Label2(k)
    Load Shape1(k)
    Load Shape2(k)
Next k

Close #3
Open "C:\Thesis\Radar\Flight\totalaircraft.tot" For Input As #3

TotalNewDisplay = 0
TotalOldReal = 0
TotalMiss = 0
TotalReal = 0
Counter = 0

TotalTrack = 0
TotalBox1 = 0
TotalBox2 = 0
TotalLabel1 = 0
TotalLabel2 = 0

With tcpServer1
    ' IMPORTANT: be sure to change the RemoteHost
    ' value to the name of your computer.
    .RemoteHost = tcpServer1.LocalHostName
    .RemotePort = 4048    ' Port to connect to.
    .Bind 4000            ' Bind to the local port.
End With

Move 0, 0
Dim i, j

RadarName = ""
i = 1
Do While (i <= 200)
    j = 1
    Do While (j <= 32)
        TrackInfo(i, j).ACType = "X"
        j = j + 1
    Loop
    i = i + 1
Loop
FindBL_URGEO ("0950000E0050000N1110000E0210000N")
MyScale = 1
'RadarCount = 5
TrackDataCount = 1
CurrTrack = True
TrackNo = 1
MyViewScale = True
MyMode = "NON"
'ViewScale.Checked = True
'ViewNoScale.Checked = True
ViewModify1.Checked = True
ModifyClick = False
PlotMap
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
End Sub
```

6.2.2 Timer1_Timer โปรแกรมย่อยควบคุมการบันทึกข้อมูลความถูกต้องของระบบและการแสดงภาพเส้นทางการบิน

```
Private Sub Timer1_Timer()
   Dim Miss As Integer
   Dim MySignN, MyPercent
   If (Start = True) Then
      Counter = Counter + 1

      If (Not EOF(3)) Then
         Line Input #3, RadarText
         AllStat.Text11 = Val(Trim(RadarText)) - 1
         Line Input #3, RadarText

         If (AllStat.Text1 < TrackDrawNew(1, 1).MaxTrack) Then
            MySignN = "+"
         ElseIf (AllStat.Text1 = TrackDrawNew(1, 1).MaxTrack) Then
            MySignN = " "
         ElseIf (AllStat.Text1 > TrackDrawNew(1, 1).MaxTrack) Then
            MySignN = "-"
         End If

         AllStat.Text1 = TrackDrawNew(1, 1).MaxTrack
         AllStat.Text6 = Val(Trim(RadarText))
         Miss = Abs(TrackDrawNew(1, 1).MaxTrack - Val(Trim(RadarText)))
         TotalMiss = TotalMiss + Miss
         TotalReal = TotalReal + Val(Trim(RadarText))
         If (TotalReal <> 0) Then
            AllStat.Text12 = Format(((TotalReal - TotalMiss) / TotalReal), "00.00%")
         Else
            AllStat.Text12 = Format(0, "00.00%")
         End If
         MyPercent = ((TotalReal - TotalMiss) / TotalReal) * 100
         'Print #2, Val(Trim(AllStat.Text11)), OldMonitor2.MysignO & Val(Trim(AllStat.Text7)), MySignN & TrackDrawNew(1, 1).MaxTrack, Val(Trim(AllStat.Text6)), TotalMiss, TotalReal, MyPercent
         Print #2, Val(Trim(AllStat.Text11)), OldMonitor2.RadarStataStr, MySignN & TrackDrawNew(1, 1).MaxTrack, Val(Trim(AllStat.Text6)), TotalMiss, TotalReal, MyPercent
      Else
         AllStat.Text6 = 0
         AllStat.Text1 = 0
         TotalReal = TotalReal
      End If
   End If
   RedrawF
End Sub
```

6.2.3 PlotMap โปรแกรมย่อยแสดงแผนที่ประเทศไทยและประเทศใกล้เคียง

```
Private Sub PlotMap()
   Dim i
```

```
    Cls
    PoltThaiBorder
    PoltNeighborBorder
End Sub
```

6.2.4 DrawFlight โปรแกรมย่อยแสดงอากาศยานเริ่มต้น

```
Public Sub DrawFlight()
    Dim x, Y, X2, Y2
    Dim i
    i = 1
    If MyClass = "F" Then
        ForeColor = &HFF00&
    Else
        ForeColor = &HFF&
    End If
    'X = (Second(Val(PointDetail.Text1.Text), Val(PointDetail.Text2.Text), Val(PointDetail.Text3.Text)) _
    '- Second(BLLonDeg, BLLonMin, BLLonSec))
    'Y = Second(URLatDeg, URLatMin, URLatSec) - Second(Val(PointDetail.Text4.Text), Val(PointDetail.Text5.Text), Val(PointDetail.Text6.Text))
    'Circle (X * SCX, Y * SCY), 30

    Circle ((TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY), 30

    'If TrackDataCount = CurrTrack Then
        If (TrackNo > 1) Then
            x = (TrackInfo(TrackDataCount, TrackNo - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo - 1).Lat) * SCY
            X2 = (TrackInfo(TrackDataCount, TrackNo).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
            Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(TrackDataCount, TrackNo).Lat) * SCY
            Line (x, Y)-(X2, Y2)
        End If
    'End If
    If Not CurrTrack Then
        TrackNo = 1
        TrackDataCount = TrackDataCount + 1
        CurrTrack = True
    Else
        TrackNo = TrackNo + 1
    End If
End Sub
```

6.2.5 ReDrawFlight โปรแกรมย่อยแสดงอากาศยานที่ได้รับการปรับปรุง

```
Private Sub ReDrawFlight()
    Dim x, Y, X2, Y2
    Dim i, j
    i = 1
    j = 1
    Do While (TrackInfo(i, j).ACType <> "X")

        If TrackInfo(i, j).Class = "F" Then
            ForeColor = &HFF00&
        Else
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            ForeColor = &HFF&
        End If
        Do While (TrackInfo(i, j).ACType <> "X")
            Circle ((TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
            (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY), 30

            If (j > 1) Then
                x = (TrackInfo(i, j - 1).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
                Y = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j - 1).Lat) * SCY
                X2 = (TrackInfo(i, j).Lon - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX
                Y2 = (Second(URLatDeg, URLatMin, URLatSec) - TrackInfo(i, j).Lat) * SCY
                Line (x, Y)-(X2, Y2)
            End If
            j = j + 1
        Loop
        i = i + 1
        j = 1
    Loop
End Sub
```

6.2.6 DrawRadarNuber โปรแกรมย่อยแสดงรัศมีของเรดาร์

```
Public Sub DrawRadarNuber(RadarNum As Integer)
    ForeColor = &HFFC0C0
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 4500 * SCX
    Circle ((RadarPosition(RadarNum, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(RadarNum, 2)) * SCY), 30

End Sub
```

6.2.7 ReDrawRadar โปรแกรมย่อยแสดงรัศมีของเรดาร์ที่ได้รับการปรับปรุง

```
Public Sub ReDrawRadar()
    Dim i
    i = 1
    ForeColor = &HFFC0C0
    Do While (i < RadarCount)
        Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 4500 * SCX
        Circle ((RadarPosition(i, 1) - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX, _
        (Second(URLatDeg, URLatMin, URLatSec) - RadarPosition(i, 2)) * SCY), 30
        i = i + 1
    Loop
End Sub
```

6.2.8 GetRadarACPosition โปรแกรมย่อยรับข้อมูลจากช่องรับสัญญาณเพื่อการคำนวณตำแหน่งของเรดาร์และตำแหน่งของอากาศยาน

```
Private Sub GetRadarACPosition(strData As String)
    Dim MyPos, OldPos
    Dim RadarNume As Integer
    Dim FirstStr
    Dim Lat, Lon, SpeedLon, SpeedLat, TrackNum, Order, SystemTime
```

```
Dim AcNo, RadarNum
Dim TrackSign As String
Dim i, x1, y1, j
Dim x As Long, Y As Long

FirstStr = Mid(strData, 1, 1)
If (FirstStr <> "*") Then
   MyPos = 1
   MyPos = InStr(MyPos, strData, "#", 1)
   OldPos = MyPos
   RadarCount = Val(Mid(strData, 1, MyPos - 1))
   'MsgBox ("Message = " & strData & "Radar count = " & RadarCount & " mypos = " & MyPos & " olpos = " & OldPos)
   For j = 1 To RadarCount - 1 Step 1
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      RadarNum = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      RadarPosition(RadarNum, 1) = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 7.5
      OldPos = MyPos
      'MsgBox ("Message = " & strData & "lon = " & RadarPosition(1, 1))
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      RadarPosition(RadarNum, 2) = Val(Mid(strData, OldPos + 1, Len(strData))) * 7.5
      'MsgBox ("Message = " & strData & "lat = " & RadarPosition(1, 2))
      DrawRadarNuber (RadarNum)
      OldPos = MyPos
   Next j

ElseIf (Mid(strData, 1, 2) = "*#") Then
   RadarNum = Val(Mid(strData, 3, Len(strData)))
   TrackDrawNew(1, 1).MaxTrack = 0
   Start = True
Else
   'MsgBox ("Data = " & strData)
   MyPos = 2
   MyPos = InStr(MyPos, strData, "#", 1)
   TotalTrack = Val(Mid(strData, 2, MyPos - 1))
   'MsgBox ("Track Count = " & TotalTrack)
   For i = 1 To TotalTrack Step 1
      'MsgBox ("i = " & i)
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      Order = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
      'MsgBox ("order = " & Order)
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      Lon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
      'MsgBox ("LON = " & Lon)
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      Lat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      SpeedLon = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
      OldPos = MyPos
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        SpeedLat = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1)) * 8
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TrackNo = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
        OldPos = MyPos
        MyPos = InStr(MyPos + 1, strData, "#", 1)
        TrackSign = Trim(Mid(strData, OldPos + 1, 1))
        x = (Lon * 7.5 - Second(BLLonDeg, BLLonMin, BLLonSec)) * SCX \ 1
        Y = (Second(URLatDeg, URLatMin, URLatSec) - Lat * 7.5) * SCY \ 1
        TrackDrawNew(1, Order).MaxTrack = TotalTrack
        TrackDrawNew(1, Order).TrackNo = TrackNo
        TrackDrawNew(1, Order).Lon = x
        TrackDrawNew(1, Order).Lat = Y
        TrackDrawNew(1, Order).Sign = TrackSign
      Next i
      OldPos = MyPos
      MyPos = InStr(MyPos + 1, strData, "#", 1)
      SystemTime = Val(Mid(strData, OldPos + 1, MyPos - OldPos - 1))
      Start = True
    End If
End Sub
```

6.2.9 RedrawF โปรแกรมย่อยปรับปรุงตำแหน่งของอากาศยาน

```
Private Sub RedrawF()
 Dim i, j, k
   For k = 0 To 400 Step 1
      Label1(k).Visible = False
      Label2(k).Visible = False
      Shape1(k).Visible = False
      Shape2(k).Visible = False
   Next k
   'PlotMap
   'ReDrawRadar
   i = 1

   For j = 1 To TrackDrawNew(i, 1).MaxTrack Step 1
      If TrackDrawNew(i, j).Sign = "F" Then
         Shape1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Top = TrackDrawNew(i, j).Lat - 50
         Shape1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Left = TrackDrawNew(i, j).Lon - 50
         Shape1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Visible = True
         Label1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Caption = "F" & Format(TrackDrawNew(i, j).TrackNo, "000")
         Label1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Top = TrackDrawNew(i, j).Lat + 50
         Label1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Left = TrackDrawNew(i, j).Lon - 50
         Label1(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Visible = True
      Else
         Shape2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Top = TrackDrawNew(i, j).Lat - 50
         Shape2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Left = TrackDrawNew(i, j).Lon - 50
         Shape2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Visible = True
         Label2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Caption = "H" & Format(TrackDrawNew(i, j).TrackNo, "000")
         Label2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Top = TrackDrawNew(i, j).Lat + 50
         Label2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Left = TrackDrawNew(i, j).Lon - 50
         Label2(TrackDrawNew(i, j).TrackNo + ((i * 200) - 200)).Visible = True
```

```
            End If
        Next j
    TrackDrawNew(1, 1).MaxTrack = 0
End Sub
```

6.3 NewStat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงในระบบใหม่

```
Private Sub Form_Load()
    Dim FileNameMax As String
    Dim FileNameStat As String
    Dim RadarText As String
    Dim Max As Integer

   Left = 8000
   Top = 1200
   Text1 = 0
   Text2 = 0
   Text3 = 0
   Text4 = 0
    ' Read max index
    FileNameMax = App.Path & "/Radar/statistic/new.txt"
    Open FileNameMax For Input As #1
    Line Input #1, RadarText
    Max = Val(Trim(RadarText))
    Close #1

    ' Write max index
    Close #1
    FileNameMax = App.Path & "/Radar/statistic/new.txt"
    Open FileNameMax For Output As #1
    Write #1, Max + 1
    Close #1

    ' Write out put
    Close #2
    FileNameStat = App.Path & "/Radar/statistic/new" & Max & ".txt"
    Open FileNameStat For Output As #2

End Sub

Public Sub UnloadMe()
   Unload Me
   Close #2
End Sub

Private Sub Form_Terminate()
    Close #3
    Close #2
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Close #3
    Close #2
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
End Sub
```

6.4 Allstat.frm แสดงจำนวนของอากาศยานที่เรดาร์แสดงและจำนวนจริงทั้งสองระบบ

```
Private Sub Command1_Click()
   OldMonitor2.SetFocus
End Sub

Private Sub Command2_Click()
   NewMonitor2.SetFocus
End Sub

Private Sub Form_Load()
   Dim FileNameMax As String
   Dim FileNameStat As String
   Dim RadarText As String
   Dim Max As Integer

  Left = 8000
  Top = 250
  Text1 = 0
  Text2 = 0
  Text3 = 0
  Text4 = 0
  Text5 = 0
  Text6 = 0
  Text7 = 0
  Text8 = 0
  Text9 = 0
  Text10 = 0
  Text11 = 0
  Text12 = 0

 ' Read max index
  FileNameMax = App.Path & "/Radar/statistic/both.txt"
  Open FileNameMax For Input As #1
  Line Input #1, RadarText
  Max = Val(Trim(RadarText))
  Close #1

  ' Write max index
  Close #1
  FileNameMax = App.Path & "/Radar/statistic/both.txt"
  Open FileNameMax For Output As #1
  Write #1, Max + 1
  Close #1

  ' Write out put
  Close #2
  FileNameStat = App.Path & "/Radar/statistic/both" & Max & ".txt"
  Open FileNameStat For Output As #2

End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
Public Sub UnloadMe()
  Unload Me
  Close #2
  Close #3
End Sub

Private Sub Form_Terminate()
   Close #2
   Close #3
End Sub

Private Sub Form_Unload(Cancel As Integer)
   Close #2
   Close #3
End Sub
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สถิติของทดสอบโปรแกรมต้นแบบกรองข้อมูลและการขจัดความซ้ำซ้อนของอากาศยานในพื้นที่ของระบบเรดาร์หลายเครื่อง

ตารางที่ ง-1 การทดสอบครั้งที่ 2

ตารางที่ 1 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 2											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
0	3	1	+5	0	+9	+8	9	1	1	9	88.89
1	3	1	5	0	9	+9	9	0	1	18	94.44
2	3	1	5	0	9	9	9	0	1	27	96.3
3	3	1	5	0	9	9	9	0	1	36	97.22
4	3	1	5	0	9	9	9	0	1	45	97.78
5	3	1	5	0	9	9	9	0	1	54	98.15
6	3	1	5	0	9	9	9	0	1	63	98.41
7	3	1	5	0	9	9	9	0	1	72	98.61
8	3	1	5	0	9	9	9	0	1	81	98.77
9	3	1	5	0	9	9	9	0	1	90	98.89
10	3	1	5	0	9	9	9	0	1	99	98.99
11	3	1	5	0	9	9	9	0	1	108	99.07
12	3	1	5	0	9	9	9	0	1	117	99.15
13	3	1	5	0	9	9	9	0	1	126	99.21
14	3	1	5	0	9	9	9	0	1	135	99.26
15	3	1	5	0	9	9	9	0	1	144	99.31
16	3	1	5	0	9	9	9	0	1	153	99.35
17	3	1	5	0	9	9	9	0	1	162	99.38
18	3	1	5	0	9	9	9	0	1	171	99.42
19	3	1	5	0	9	9	9	0	1	180	99.44
20	3	1	5	0	9	9	9	0	1	189	99.47
21	3	+2	5	0	+10	9	9	0	1	198	99.49
22	3	2	5	0	10	9	9	0	1	207	99.52
23	3	2	5	0	10	9	9	0	1	216	99.54
24	3	2	5	0	10	9	9	0	1	225	99.56
25	3	2	5	0	10	9	9	0	1	234	99.57
26	3	2	5	0	10	9	9	0	1	243	99.59
27	3	2	-4	0	-9	9	9	0	1	252	99.6
28	+4	2	+5	0	+11	9	9	0	1	261	99.62
29	4	2	5	0	11	9	9	0	1	270	99.63
30	4	2	5	0	11	9	9	0	1	279	99.64
31	4	2	5	0	11	9	9	0	1	288	99.65
32	4	2	5	0	11	9	9	0	1	297	99.66
33	4	2	5	0	11	9	9	0	1	306	99.67
34	4	2	5	0	11	9	9	0	1	315	99.68
35	4	+3	5	0	+12	9	9	0	1	324	99.69
36	+5	3	5	0	+13	9	9	0	1	333	99.7
37	5	3	5	0	13	9	9	0	1	342	99.71
38	5	3	5	0	13	9	9	0	1	351	99.72

ตารางที่ 1 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 2											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
39	5	3	5	0	13	9	9	0	1	360	99.72
40	5	+4	5	0	+14	9	9	0	1	369	99.73
41	5	4	5	0	14	9	9	0	1	378	99.74
42	5	4	5	0	14	9	9	0	1	387	99.74
43	5	4	5	0	14	9	9	0	1	396	99.75
44	5	4	5	0	14	9	9	0	1	405	99.75
45	5	4	5	0	14	9	9	0	1	414	99.76
46	5	4	5	0	14	9	9	0	1	423	99.76
47	5	4	5	0	14	9	9	0	1	432	99.77
48	5	4	5	0	14	9	9	0	1	441	99.77
49	5	4	5	0	14	9	9	0	1	450	99.78
50	5	4	5	+1	+15	9	9	0	1	459	99.78
51	5	4	5	1	15	9	9	0	1	468	99.79
52	5	4	5	1	15	9	9	0	1	477	99.79
53	5	4	5	1	15	9	9	0	1	486	99.79
54	5	4	5	1	15	9	9	0	1	495	99.8
55	5	4	5	1	15	9	9	0	1	504	99.8
56	5	4	5	1	15	9	9	0	1	513	99.81
57	5	4	5	+3	+17	9	9	0	1	522	99.81
58	5	4	5	3	17	9	9	0	1	531	99.81
59	5	4	5	3	17	9	9	0	1	540	99.81
60	5	4	5	+4	+18	9	9	0	1	549	99.82
61	5	4	5	4	18	9	9	0	1	558	99.82
62	5	4	5	4	18	9	9	0	1	567	99.82
63	5	4	5	4	18	9	9	0	1	576	99.83
64	5	4	5	4	18	9	9	0	1	585	99.83
65	5	4	5	4	18	9	9	0	1	594	99.83
66	5	4	5	4	18	9	9	0	1	603	99.83
67	5	4	5	4	18	9	9	0	1	612	99.84
68	5	4	5	4	18	9	9	0	1	621	99.84
69	5	4	5	4	18	9	9	0	1	630	99.84
70	5	4	5	4	18	9	9	0	1	639	99.84
71	5	4	5	4	18	9	9	0	1	648	99.85
72	5	4	5	4	18	9	9	0	1	657	99.85
73	5	4	5	4	18	9	9	0	1	666	99.85
74	5	4	5	4	18	9	9	0	1	675	99.85
75	5	4	5	4	18	9	9	0	1	684	99.85
76	5	4	5	4	18	9	9	0	1	693	99.86
77	5	4	5	4	18	9	9	0	1	702	99.86
78	5	4	5	4	18	9	9	0	1	711	99.86
79	5	4	5	4	18	9	9	0	1	720	99.86
80	5	4	5	4	18	9	9	0	1	729	99.86
81	5	4	5	4	18	9	9	0	1	738	99.86
82	5	4	5	4	18	9	9	0	1	747	99.87
83	5	4	5	4	18	9	9	0	1	756	99.87
84	5	4	5	4	18	9	9	0	1	765	99.87
85	5	4	5	4	18	9	9	0	1	774	99.87
86	5	4	5	4	18	9	9	0	1	783	99.87
87	5	4	5	4	18	9	9	0	1	792	99.87

ตารางที่ 1 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 2											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
88	5	4	5	4	18	9	9	0	1	801	99.88
89	5	4	5	4	18	9	9	0	1	810	99.88
90	5	4	5	4	18	9	9	0	1	819	99.88
91	5	4	5	4	18	9	9	0	1	828	99.88
92	5	4	5	4	18	9	9	0	1	837	99.88
93	5	4	5	4	18	9	9	0	1	846	99.88
94	5	4	5	4	18	9	9	0	1	855	99.88
95	5	4	5	4	18	9	9	0	1	864	99.88
96	5	4	5	4	18	9	9	0	1	873	99.89
97	5	4	5	4	18	9	9	0	1	882	99.89
98	5	4	5	-3	-17	9	9	0	1	891	99.89
99	5	4	5	+4	+18	9	9	0	1	900	99.89
100	5	4	5	4	18	9	9	0	1	909	99.89
101	5	-3	5	4	-17	9	9	0	1	918	99.89
102	5	3	5	4	17	9	9	0	1	927	99.89
103	5	3	5	4	17	9	9	0	1	936	99.89
104	5	3	5	4	17	9	9	0	1	945	99.89
105	5	3	-4	4	-16	9	9	0	1	954	99.9
106	-4	3	4	4	-15	9	9	0	1	963	99.9
107	4	3	4	4	15	9	9	0	1	972	99.9
108	4	3	4	4	15	9	9	0	1	981	99.9
109	4	3	4	4	15	9	9	0	1	990	99.9
110	4	3	4	4	15	9	9	0	1	999	99.9
111	4	3	4	4	15	9	9	0	1	1,008	99.9
112	4	3	4	4	15	9	9	0	1	1,017	99.9
113	4	3	4	4	15	9	9	0	1	1,026	99.9
114	4	3	4	4	15	9	9	0	1	1,035	99.9
115	4	3	4	4	15	9	9	0	1	1,044	99.9
116	4	3	4	4	15	9	9	0	1	1,053	99.91
117	4	3	4	4	15	9	9	0	1	1,062	99.91
118	4	3	4	4	15	9	9	0	1	1,071	99.91
119	4	3	4	4	15	9	9	0	1	1,080	99.91
120	4	3	4	4	15	9	9	0	1	1,089	99.91
121	4	3	4	4	15	9	9	0	1	1,098	99.91
122	4	3	4	4	15	9	9	0	1	1,107	99.91
123	4	3	4	4	15	9	9	0	1	1,116	99.91
124	4	3	4	4	15	9	9	0	1	1,125	99.91
125	4	3	4	4	15	9	9	0	1	1,134	99.91
126	4	3	4	4	15	9	9	0	1	1,143	99.91
127	4	3	4	4	15	9	9	0	1	1,152	99.91
128	4	3	4	4	15	9	9	0	1	1,161	99.91
129	4	3	-3	4	-14	9	9	0	1	1,170	99.91
130	4	3	3	4	14	9	9	0	1	1,179	99.92
131	4	3	3	4	14	9	9	0	1	1,188	99.92
132	4	3	3	4	14	9	9	0	1	1,197	99.92
133	4	3	-2	4	-13	9	9	0	1	1,206	99.92
134	4	3	2	4	13	9	9	0	1	1,215	99.92
135	4	3	2	4	13	9	9	0	1	1,224	99.92
136	4	3	2	4	13	9	9	0	1	1,233	99.92

ตารางที่ 1 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 2											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
137	4	3	2	4	13	9	9	0	1	1,242	99.92
138	4	3	2	4	13	9	9	0	1	1,251	99.92
139	4	3	2	4	13	9	9	0	1	1,260	99.92
140	4	3	2	4	13	9	9	0	1	1,269	99.92
141	4	3	2	4	13	9	9	0	1	1,278	99.92
142	4	3	2	4	13	9	9	0	1	1,287	99.92
143	4	3	2	4	13	9	9	0	1	1,296	99.92
144	-3	3	2	4	-12	9	9	0	1	1,305	99.92
145	3	3	2	4	12	9	9	0	1	1,314	99.92
146	3	3	2	4	12	9	9	0	1	1,323	99.92
147	3	3	2	4	12	9	9	0	1	1,332	99.92
148	3	3	2	4	12	9	9	0	1	1,341	99.93
149	3	3	2	4	12	9	9	0	1	1,350	99.93
150	3	3	2	4	12	9	9	0	1	1,359	99.93
151	3	3	2	4	12	9	9	0	1	1,368	99.93
152	3	-2	2	4	-11	9	9	0	1	1,377	99.93
153	3	2	2	4	11	-8	9	1	2	1,386	99.86
154	3	2	2	4	11	8	8	0	2	1,394	99.86
155	3	2	2	4	11	8	8	0	2	1,402	99.86
156	3	2	2	4	11	8	8	0	2	1,410	99.86
157	3	2	2	4	11	8	8	0	2	1,418	99.86
158	3	2	-1	4	-10	8	8	0	2	1,426	99.86
159	3	2	1	4	10	8	8	0	2	1,434	99.86
160	3	+3	1	4	+11	8	8	0	2	1,442	99.86
161	3	3	1	4	11	8	8	0	2	1,450	99.86
162	3	3	1	4	11	8	8	0	2	1,458	99.86
163	3	3	1	-3	-10	8	8	0	2	1,466	99.86
164	3	3	1	3	10	-7	8	1	3	1,474	99.8
165	3	3	1	3	10	7	7	0	3	1,481	99.8
166	3	3	1	3	10	7	7	0	3	1,488	99.8
167	3	3	-0	3	+9	7	7	0	3	1,495	99.8
168	3	3	0	3	9	7	7	0	3	1,502	99.8
169	3	3	0	3	9	7	7	0	3	1,509	99.8
170	3	3	0	3	9	7	7	0	3	1,516	99.8
171	3	3	0	3	9	7	7	0	3	1,523	99.8
172	3	3	0	3	9	7	7	0	3	1,530	99.8
173	3	3	0	3	9	7	7	0	3	1,537	99.8
174	3	3	0	3	9	7	7	0	3	1,544	99.81
175	3	3	0	3	9	7	7	0	3	1,551	99.81
176	3	3	0	3	9	7	7	0	3	1,558	99.81
177	3	-2	0	3	-8	7	7	0	3	1,565	99.81
178	3	2	0	3	8	-6	7	1	4	1,572	99.75
179	3	2	0	3	8	6	7	1	5	1,579	99.68
180	3	2	0	-1	-6	6	6	0	5	1,585	99.68
181	-2	2	0	1	-5	-5	5	0	5	1,590	99.69
182	2	2	0	1	5	-4	5	1	6	1,595	99.62
183	2	2	0	1	5	4	4	0	6	1,599	99.62
184	2	2	0	1	5	4	4	0	6	1,603	99.63
185	2	2	0	1	5	4	4	0	6	1,607	99.63

ตารางที่ 1 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 2											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
186	2	2	0	1	5	4	4	0	6	1,611	99.63
187	2	2	0	1	5	4	4	0	6	1,615	99.63
188	2	2	0	1	5	4	4	0	6	1,619	99.63
189	2	2	0	1	5	4	4	0	6	1,623	99.63
190	2	2	0	1	5	4	4	0	6	1,627	99.63
191	2	2	0	1	5	4	4	0	6	1,631	99.63
192	-1	2	0	1	-4	-3	4	1	7	1,635	99.57
193	1	2	0	1	4	3	3	0	7	1,638	99.57
194	1	2	0	1	4	3	3	0	7	1,641	99.57
195	1	2	0	1	4	3	3	0	7	1,644	99.57
196	1	2	0	1	4	3	3	0	7	1,647	99.57
197	1	2	0	1	4	3	3	0	7	1,650	99.58
198	1	2	0	1	4	3	3	0	7	1,653	99.58
199	1	2	0	1	4	3	3	0	7	1,656	99.58
200	1	2	0	1	4	3	3	0	7	1,659	99.58
201	1	2	0	1	4	3	3	0	7	1,662	99.58
202	1	2	0	1	4	3	3	0	7	1,665	99.58
203	1	2	0	1	4	3	3	0	7	1,668	99.58
204	-0	2	0	1	-3	3	3	0	7	1,671	99.58
205	0	2	0	1	3	-2	3	1	8	1,674	99.52
206	0	2	0	1	3	2	3	1	9	1,677	99.46
207	0	2	0	1	3	2	2	0	9	1,679	99.46
208	0	2	0	1	3	2	2	0	9	1,681	99.46
209	0	2	0	1	3	2	2	0	9	1,683	99.47
210	0	2	0	1	3	2	2	0	9	1,685	99.47
211	0	2	0	1	3	2	2	0	9	1,687	99.47
212	0	2	0	1	3	2	2	0	9	1,689	99.47
213	0	-1	0	1	-2	2	2	0	9	1,691	99.47
214	0	1	0	1	2	-1	2	1	10	1,693	99.41
215	0	1	0	1	2	1	1	0	10	1,694	99.41
216	0	1	0	1	2	1	1	0	10	1,695	99.41
217	0	1	0	1	2	1	1	0	10	1,696	99.41
218	0	1	0	1	2	1	1	0	10	1,697	99.41
219	0	1	0	1	2	1	1	0	10	1,698	99.41
220	0	1	0	1	2	1	1	0	10	1,699	99.41
221	0	1	0	1	2	1	1	0	10	1,700	99.41

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ง-2 การทดสอบครั้งที่ 3

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
0	6	1	14	6	27	26	27	1	1	27	96.3
1	6	1	15	6	28	27	27	0	1	54	98.15
2	6	1	15	6	28	27	27	0	1	81	98.77
3	6	1	15	6	28	27	27	0	1	108	99.07
4	6	1	15	6	28	27	27	0	1	135	99.26
5	6	1	15	6	28	27	27	0	1	162	99.38
6	6	1	15	6	28	27	27	0	1	189	99.47
7	6	1	15	6	28	27	27	0	1	216	99.54
8	6	1	15	6	28	27	27	0	1	243	99.59
9	6	1	15	6	28	27	27	0	1	270	99.63
10	6	1	15	6	28	27	27	0	1	297	99.66
11	6	1	15	6	28	27	27	0	1	324	99.69
12	6	2	15	6	29	27	27	0	1	351	99.72
13	6	2	15	6	29	27	27	0	1	378	99.74
14	6	2	15	6	29	27	27	0	1	405	99.75
15	6	2	16	6	30	27	27	0	1	432	99.77
16	6	2	16	6	30	27	27	0	1	459	99.78
17	6	2	16	6	30	27	27	0	1	486	99.79
18	6	2	16	6	30	27	27	0	1	513	99.81
19	6	2	16	6	30	27	27	0	1	540	99.81
20	6	2	16	6	30	27	27	0	1	567	99.82
21	6	2	17	6	31	27	27	0	1	594	99.83
22	6	3	17	6	32	27	27	0	1	621	99.84
23	6	3	17	6	32	27	27	0	1	648	99.85
24	6	3	17	6	32	-26	27	1	2	675	99.7
25	6	4	17	6	33	27	27	0	2	702	99.72
26	6	4	17	6	33	27	27	0	2	729	99.73
27	6	4	18	6	34	27	27	0	2	756	99.74
28	6	4	18	7	35	-26	27	1	3	783	99.62
29	7	4	18	7	36	27	27	0	3	810	99.63
30	7	4	18	7	36	27	27	0	3	837	99.64
31	7	4	-17	7	-35	27	27	0	3	864	99.65
32	7	4	19	7	37	-26	27	1	4	891	99.55
33	7	4	-18	7	-36	28	28	0	4	919	99.56
34	7	4	19	7	37	-27	28	1	5	947	99.47
35	8	4	19	7	38	28	28	0	5	975	99.49
36	9	4	-18	-6	-37	28	28	0	5	1,003	99.5
37	9	4	19	6	38	-27	28	1	6	1,031	99.42
38	11	4	19	6	40	28	28	0	6	1,059	99.43
39	11	4	19	6	40	28	28	0	6	1,087	99.45
40	11	5	19	6	41	28	28	0	6	1,115	99.46

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
41	11	5	19	7	42	29	28	1	7	1,143	99.39
42	11	5	19	7	42	-28	28	0	7	1,171	99.4
43	12	5	20	7	44	30	28	2	9	1,199	99.25
44	12	5	20	7	44	30	30	0	9	1,229	99.27
45	12	5	20	7	44	30	30	0	9	1,259	99.29
46	12	5	20	7	44	30	30	0	9	1,289	99.3
47	12	6	20	7	45	30	30	0	9	1,319	99.32
48	12	6	20	7	45	30	30	0	9	1,349	99.33
49	12	6	20	7	45	30	30	0	9	1,379	99.35
50	12	6	20	8	46	30	30	0	9	1,409	99.36
51	12	6	20	8	46	30	30	0	9	1,439	99.37
52	12	6	20	8	46	30	30	0	9	1,469	99.39
53	12	6	20	8	46	30	30	0	9	1,499	99.4
54	12	6	21	8	47	31	30	1	10	1,529	99.35
55	13	6	21	8	48	31	31	0	10	1,560	99.36
56	13	6	21	8	48	31	31	0	10	1,591	99.37
57	13	6	21	10	50	31	31	0	10	1,622	99.38
58	13	6	21	-9	-49	31	31	0	10	1,653	99.4
59	13	6	21	9	49	31	31	0	10	1,684	99.41
60	13	6	21	9	49	31	31	0	10	1,715	99.42
61	13	6	21	9	49	31	31	0	10	1,746	99.43
62	14	6	21	9	50	31	31	0	10	1,777	99.44
63	14	6	22	9	51	31	31	0	10	1,808	99.45
64	14	6	22	9	51	31	31	0	10	1,839	99.46
65	14	6	22	9	51	31	31	0	10	1,870	99.47
66	14	6	22	9	51	31	31	0	10	1,901	99.47
67	14	6	22	9	51	31	31	0	10	1,932	99.48
68	14	6	22	9	51	31	31	0	10	1,963	99.49
69	14	6	22	9	51	31	31	0	10	1,994	99.5
70	14	6	22	9	51	31	31	0	10	2,025	99.51
71	14	6	22	-8	-50	31	31	0	10	2,056	99.51
72	14	6	22	8	50	31	31	0	10	2,087	99.52
73	14	6	22	8	50	31	31	0	10	2,118	99.53
74	14	6	22	8	50	31	31	0	10	2,149	99.53
75	14	6	22	8	50	31	31	0	10	2,180	99.54
76	14	6	22	8	50	31	31	0	10	2,211	99.55
77	14	6	22	8	50	31	31	0	10	2,242	99.55
78	14	6	-21	8	-49	31	31	0	10	2,273	99.56
79	14	6	-20	8	-48	31	31	0	10	2,304	99.57
80	14	6	20	8	48	31	31	0	10	2,335	99.57
81	14	6	20	8	48	31	31	0	10	2,366	99.58
82	14	6	20	8	48	31	31	0	10	2,397	99.58
83	14	6	20	8	48	31	31	0	10	2,428	99.59
84	14	6	20	8	48	31	31	0	10	2,459	99.59
85	14	6	20	8	48	31	31	0	10	2,490	99.6
86	14	6	20	8	48	31	31	0	10	2,521	99.6
87	14	6	20	8	48	31	31	0	10	2,552	99.61
88	-13	6	20	8	-47	31	31	0	10	2,583	99.61
89	13	6	20	8	47	31	31	0	10	2,614	99.62

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
90	13	6	20	8	47	31	31	0	10	2,645	99.62
91	13	-5	20	8	-46	31	31	0	10	2,676	99.63
92	13	5	20	-7	-45	-30	31	1	11	2,707	99.59
93	13	5	20	8	46	30	30	0	11	2,737	99.6
94	13	5	-19	8	-45	30	30	0	11	2,767	99.6
95	13	5	19	8	45	-29	30	1	12	2,797	99.57
96	13	5	19	8	45	29	29	0	12	2,826	99.58
97	13	5	-18	8	-44	29	29	0	12	2,855	99.58
98	13	5	18	8	44	-28	29	1	13	2,884	99.55
99	13	5	18	8	44	28	28	0	13	2,912	99.55
100	13	-4	18	8	-43	28	28	0	13	2,940	99.56
101	13	4	18	8	43	28	28	0	13	2,968	99.56
102	13	4	-16	8	-41	-27	28	1	14	2,996	99.53
103	13	4	16	8	41	27	28	1	15	3,024	99.5
104	13	4	16	8	41	27	27	0	15	3,051	99.51
105	-12	4	16	8	-40	27	27	0	15	3,078	99.51
106	12	4	-15	-7	-38	-26	27	1	16	3,105	99.48
107	12	4	15	7	38	26	27	1	17	3,132	99.46
108	12	4	15	7	38	26	26	0	17	3,158	99.46
109	12	4	15	7	38	26	26	0	17	3,184	99.47
110	12	4	-14	7	-37	26	26	0	17	3,210	99.47
111	12	4	14	7	37	26	26	0	17	3,236	99.47
112	12	4	14	7	37	26	26	0	17	3,262	99.48
113	12	4	14	7	37	26	26	0	17	3,288	99.48
114	12	4	14	7	37	26	26	0	17	3,314	99.49
115	12	4	15	7	38	26	26	0	17	3,340	99.49
116	12	4	15	7	38	26	26	0	17	3,366	99.49
117	-11	4	15	7	-37	26	26	0	17	3,392	99.5
118	11	4	15	7	37	26	26	0	17	3,418	99.5
119	11	4	15	7	37	26	26	0	17	3,444	99.51
120	11	4	-14	7	-36	-25	26	1	18	3,470	99.48
121	11	4	-13	7	-35	-24	26	2	20	3,496	99.43
122	11	-3	13	7	-34	24	25	1	21	3,521	99.4
123	11	3	13	7	34	24	25	1	22	3,546	99.38
124	11	3	13	-6	-33	-23	24	1	23	3,570	99.36
125	11	3	13	6	33	23	24	1	24	3,594	99.33
126	11	3	-12	6	-32	-22	23	1	25	3,617	99.31
127	11	3	12	6	32	22	23	1	26	3,640	99.29
128	11	3	-11	6	-31	22	22	0	26	3,662	99.29
129	11	3	11	6	31	22	22	0	26	3,684	99.29
130	-10	3	11	6	-30	22	22	0	26	3,706	99.3
131	10	3	11	6	30	22	22	0	26	3,728	99.3
132	10	3	-10	6	-29	22	22	0	26	3,750	99.31
133	10	3	10	6	29	22	22	0	26	3,772	99.31
134	10	3	10	6	29	22	22	0	26	3,794	99.31
135	10	3	10	6	29	22	22	0	26	3,816	99.32
136	10	3	10	6	29	22	22	0	26	3,838	99.32
137	10	3	10	6	29	22	22	0	26	3,860	99.33
138	10	3	10	6	29	22	22	0	26	3,882	99.33

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
139	10	4	10	6	30	22	22	0	26	3,904	99.33
140	10	4	10	-5	-29	-21	22	1	27	3,926	99.31
141	-9	4	10	5	-28	-20	22	2	29	3,948	99.27
142	-8	4	10	5	-27	-19	22	3	32	3,970	99.19
143	-7	4	10	5	-26	19	19	0	32	3,989	99.2
144	7	4	10	5	26	19	19	0	32	4,008	99.2
145	-6	4	10	5	-25	-18	19	1	33	4,027	99.18
146	6	4	10	5	25	18	19	1	34	4,046	99.16
147	6	4	10	5	25	18	19	1	35	4,065	99.14
148	6	4	-9	5	-24	18	18	0	35	4,083	99.14
149	6	4	9	5	24	18	18	0	35	4,101	99.15
150	6	4	9	5	24	18	18	0	35	4,119	99.15
151	6	4	-8	-4	-22	18	18	0	35	4,137	99.15
152	6	-3	8	4	-21	-17	18	1	36	4,155	99.13
153	6	3	8	4	21	17	18	1	37	4,173	99.11
154	6	3	8	4	21	17	17	0	37	4,190	99.12
155	6	3	8	4	21	17	17	0	37	4,207	99.12
156	6	3	8	4	21	17	17	0	37	4,224	99.12
157	6	3	7	4	20	17	17	0	37	4,241	99.13
158	6	3	7	4	20	17	17	0	37	4,258	99.13
159	6	3	7	4	20	17	17	0	37	4,275	99.13
160	6	3	7	4	20	17	17	0	37	4,292	99.14
161	6	3	7	4	20	17	17	0	37	4,309	99.14
162	6	3	7	-3	-19	-16	17	1	38	4,326	99.12
163	6	3	-6	3	-18	-15	17	2	40	4,343	99.08
164	6	3	6	3	18	15	17	2	42	4,360	99.04
165	6	3	6	3	18	15	16	1	43	4,376	99.02
166	6	3	6	3	18	15	15	0	43	4,391	99.02
167	6	3	-4	-2	-15	-14	15	1	44	4,406	99
168	6	3	4	2	15	14	15	1	45	4,421	98.98
169	6	3	4	2	15	14	15	1	46	4,436	98.96
170	-5	3	4	2	-14	-13	14	1	47	4,450	98.94
171	5	3	4	2	14	13	14	1	48	4,464	98.92
172	5	3	4	2	14	13	14	1	49	4,478	98.91
173	5	3	4	2	14	13	13	0	49	4,491	98.91
174	5	3	-3	2	-13	-12	13	1	50	4,504	98.89
175	5	3	3	2	13	-11	13	2	52	4,517	98.85
176	-4	-2	3	2	-11	11	13	2	54	4,530	98.81
177	4	2	3	2	11	11	13	2	56	4,543	98.77
178	4	2	3	0	9	-10	12	2	58	4,555	98.73
179	4	2	3	0	9	-9	12	3	61	4,567	98.66
180	4	2	3	0	9	9	11	2	63	4,578	98.62
181	4	2	3	0	9	9	10	1	64	4,588	98.61
182	4	2	-2	0	-8	-8	10	2	66	4,598	98.56
183	4	2	2	0	8	8	9	1	67	4,607	98.55
184	4	2	2	0	8	8	9	1	68	4,616	98.53
185	4	2	2	0	8	8	8	0	68	4,624	98.53
186	4	3	2	0	9	8	8	0	68	4,632	98.53
187	4	3	2	0	9	8	8	0	68	4,640	98.53

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
188	4	3	2	0	9	8	8	0	68	4,648	98.54
189	4	3	2	0	9	-7	8	1	69	4,656	98.52
190	-3	-2	2	0	-7	-6	8	2	71	4,664	98.48
191	-2	2	-1	0	-5	-4	8	4	75	4,672	98.39
192	2	2	1	0	5	4	8	4	79	4,680	98.31
193	2	2	1	0	5	4	7	3	82	4,687	98.25
194	2	2	1	0	5	4	5	1	83	4,692	98.23
195	2	2	1	0	5	4	4	0	83	4,696	98.23
196	2	2	1	0	5	4	4	0	83	4,700	98.23
197	2	2	1	0	5	4	4	0	83	4,704	98.24
198	2	2	1	0	5	4	4	0	83	4,708	98.24
199	2	2	1	0	5	4	4	0	83	4,712	98.24
200	2	2	0	0	-4	4	4	0	83	4,716	98.24
201	2	2	0	0	4	-3	4	1	84	4,720	98.22
202	2	2	0	0	4	3	4	1	85	4,724	98.2
203	2	2	0	0	4	3	4	1	86	4,728	98.18
204	-1	2	0	0	-3	-2	4	2	88	4,732	98.14
205	1	2	0	0	3	2	3	1	89	4,735	98.12
206	1	2	0	0	3	2	3	1	90	4,738	98.1
207	1	2	0	0	3	2	2	0	90	4,740	98.1
208	1	2	0	0	3	2	2	0	90	4,742	98.1
209	1	2	0	0	3	2	2	0	90	4,744	98.1
210	1	2	0	0	3	2	2	0	90	4,746	98.1
211	1	2	0	0	3	2	2	0	90	4,748	98.1
212	1	2	0	0	3	2	2	0	90	4,750	98.11
213	1	2	0	0	3	2	2	0	90	4,752	98.11
214	1	2	0	0	3	2	2	0	90	4,754	98.11
215	1	2	0	0	3	2	2	0	90	4,756	98.11
216	1	2	0	0	3	2	2	0	90	4,758	98.11
217	1	2	0	0	3	2	2	0	90	4,760	98.11
218	1	-1	0	0	-2	-1	2	1	91	4,762	98.09
219	1	1	0	0	2	1	2	1	92	4,764	98.07
220	1	1	0	0	2	1	2	1	93	4,766	98.05
221	1	1	0	0	2	1	2	1	94	4,768	98.03
222	1	1	0	0	2	1	1	0	94	4,769	98.03
223	1	1	0	0	2	1	1	0	94	4,770	98.03
224	1	1	0	0	2	1	1	0	94	4,771	98.03
225	1	1	0	0	2	1	1	0	94	4,772	98.03
226	1	1	0	0	2	1	1	0	94	4,773	98.03
227	1	1	0	0	2	1	1	0	94	4,774	98.03
228	1	1	0	0	2	1	1	0	94	4,775	98.03
229	1	1	0	0	2	1	1	0	94	4,776	98.03
230	1	1	0	0	2	1	1	0	94	4,777	98.03
231	1	1	0	0	2	1	1	0	94	4,778	98.03
232	1	1	0	0	2	1	1	0	94	4,779	98.03
233	1	1	0	0	2	1	1	0	94	4,780	98.03
234	1	1	0	0	2	1	1	0	94	4,781	98.03
235	1	1	0	0	2	1	1	0	94	4,782	98.03
236	1	1	0	0	2	1	1	0	94	4,783	98.03

ตารางที่ 2 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 3											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
237	0	0	0	0	0	1	1	0	94	4,784	98.04
238	0	0	0	0	0	0	1	1	95	4,785	98.01

ตารางที่ ง-3 การทดสอบครั้งที่ 4

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
0	6	1	14	6	27	26	27	1	1	27	96.30
1	6	1	15	6	28	27	27	0	1	54	98.15
2	6	1	15	6	28	27	27	0	1	81	98.77
3	6	1	15	6	28	27	27	0	1	108	99.07
4	6	1	15	6	28	27	27	0	1	135	99.26
5	6	1	15	6	28	27	27	0	1	162	99.38
6	6	1	15	6	28	27	27	0	1	189	99.47
7	6	1	15	6	28	27	27	0	1	216	99.54
8	6	1	15	6	28	27	27	0	1	243	99.59
9	6	1	15	6	28	27	27	0	1	270	99.63
10	6	1	15	6	28	27	27	0	1	297	99.66
11	6	1	15	6	28	27	27	0	1	324	99.69
12	6	2	15	6	29	27	27	0	1	351	99.72
13	6	2	15	6	29	27	27	0	1	378	99.74
14	6	2	15	6	29	27	27	0	1	405	99.75
15	6	2	16	6	30	27	27	0	1	432	99.77
16	6	2	16	6	30	27	27	0	1	459	99.78
17	6	2	16	6	30	27	27	0	1	486	99.79
18	6	2	16	6	30	27	27	0	1	513	99.81
19	6	2	16	6	30	27	27	0	1	540	99.81
20	6	2	16	6	30	27	27	0	1	567	99.82
21	6	2	17	6	31	27	27	0	1	594	99.83
22	6	3	17	6	32	27	27	0	1	621	99.84
23	6	3	17	6	32	27	27	0	1	648	99.85
24	6	3	17	6	32	-26	27	1	2	675	99.7
25	6	4	17	6	33	27	27	0	2	702	99.72
26	6	4	17	6	33	27	27	0	2	729	99.73
27	6	4	18	6	34	27	27	0	2	756	99.74
28	6	4	18	7	35	-26	27	1	3	783	99.62
29	7	4	18	7	36	27	27	0	3	810	99.63
30	7	4	18	7	36	27	27	0	3	837	99.64
31	7	4	-17	7	-35	27	27	0	3	864	99.65
32	7	4	19	7	37	-26	27	1	4	891	99.55

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
33	7	4	-18	7	-36	28	28	0	4	919	99.56
34	7	4	19	7	37	-27	28	1	5	947	99.47
35	8	4	19	7	38	28	28	0	5	975	99.49
36	9	4	-18	-6	-37	28	28	0	5	1,003	99.5
37	9	4	19	6	38	-27	28	1	6	1,031	99.42
38	11	4	19	6	40	28	28	0	6	1,059	99.43
39	11	4	19	6	40	28	28	0	6	1,087	99.45
40	11	5	19	6	41	28	28	0	6	1,115	99.46
41	11	5	19	7	42	29	28	1	7	1,143	99.39
42	11	5	19	7	42	-28	28	0	7	1,171	99.4
43	12	5	20	7	44	30	28	2	9	1,199	99.25
44	12	5	20	7	44	30	30	0	9	1,229	99.27
45	12	5	20	7	44	30	30	0	9	1,259	99.29
46	12	5	20	7	44	30	30	0	9	1,289	99.3
47	12	6	20	7	45	30	30	0	9	1,319	99.32
48	12	6	20	7	45	30	30	0	9	1,349	99.33
49	12	6	20	7	45	30	30	0	9	1,379	99.35
50	12	6	20	8	46	30	30	0	9	1,409	99.36
51	12	6	20	8	46	30	30	0	9	1,439	99.37
52	12	6	20	8	46	30	30	0	9	1,469	99.39
53	12	6	20	8	46	30	30	0	9	1,499	99.4
54	12	6	21	8	47	31	30	1	10	1,529	99.35
55	13	6	21	8	48	31	31	0	10	1,560	99.36
56	13	6	21	8	48	31	31	0	10	1,591	99.37
57	13	6	21	10	50	31	31	0	10	1,622	99.38
58	13	6	21	-9	-49	31	31	0	10	1,653	99.4
59	13	6	21	9	49	31	31	0	10	1,684	99.41
60	13	6	21	9	49	31	31	0	10	1,715	99.42
61	13	6	21	9	49	31	31	0	10	1,746	99.43
62	14	6	21	9	50	31	31	0	10	1,777	99.44
63	14	6	22	9	51	31	31	0	10	1,808	99.45
64	14	6	22	9	51	31	31	0	10	1,839	99.46
65	14	6	22	9	51	31	31	0	10	1,870	99.47
66	14	6	22	9	51	31	31	0	10	1,901	99.47
67	14	6	22	9	51	31	31	0	10	1,932	99.48
68	14	6	22	9	51	31	31	0	10	1,963	99.49
69	14	6	22	9	51	31	31	0	10	1,994	99.5
70	14	6	22	9	51	31	31	0	10	2,025	99.51
71	14	6	22	-8	-50	31	31	0	10	2,056	99.51
72	14	6	22	8	50	31	31	0	10	2,087	99.52
73	14	6	22	8	50	31	31	0	10	2,118	99.53
74	14	6	22	8	50	31	31	0	10	2,149	99.53
75	14	6	22	8	50	31	31	0	10	2,180	99.54
76	14	6	22	8	50	31	31	0	10	2,211	99.55
77	14	6	22	8	50	31	31	0	10	2,242	99.55
78	14	6	-21	8	-49	31	31	0	10	2,273	99.56
79	14	6	-20	8	-48	31	31	0	10	2,304	99.57
80	14	6	20	8	48	31	31	0	10	2,335	99.57
81	14	6	20	8	48	31	31	0	10	2,366	99.58

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
82	14	6	20	8	48	31	31	0	10	2,397	99.58
83	14	6	20	8	48	31	31	0	10	2,428	99.59
84	14	6	20	8	48	31	31	0	10	2,459	99.59
85	14	6	20	8	48	31	31	0	10	2,490	99.6
86	14	6	20	8	48	31	31	0	10	2,521	99.6
87	14	6	20	8	48	31	31	0	10	2,552	99.61
88	-13	6	20	8	-47	31	31	0	10	2,583	99.61
89	13	6	20	8	47	31	31	0	10	2,614	99.62
90	13	6	20	8	47	31	31	0	10	2,645	99.62
91	13	-5	20	8	-46	31	31	0	10	2,676	99.63
92	13	5	20	-7	-45	-30	31	1	11	2,707	99.59
93	13	5	20	8	46	30	30	0	11	2,737	99.6
94	13	5	-19	8	-45	30	30	0	11	2,767	99.6
95	13	5	19	8	45	-29	30	1	12	2,797	99.57
96	13	5	19	8	45	29	29	0	12	2,826	99.58
97	13	5	-18	8	-44	29	29	0	12	2,855	99.58
98	13	5	18	8	44	-28	29	1	13	2,884	99.55
99	13	5	18	8	44	28	28	0	13	2,912	99.55
100	13	-4	18	8	-43	28	28	0	13	2,940	99.56
101	13	4	18	8	43	28	28	0	13	2,968	99.56
102	13	4	-16	8	-41	-27	28	1	14	2,996	99.53
103	13	4	16	8	41	27	28	1	15	3,024	99.5
104	13	4	16	8	41	27	27	0	15	3,051	99.51
105	-12	4	16	8	-40	27	27	0	15	3,078	99.51
106	12	4	-15	-7	-38	-26	27	1	16	3,105	99.48
107	12	4	15	7	38	26	27	1	17	3,132	99.46
108	12	4	15	7	38	26	26	0	17	3,158	99.46
109	12	4	15	7	38	26	26	0	17	3,184	99.47
110	12	4	-14	7	-37	26	26	0	17	3,210	99.47
111	12	4	14	7	37	26	26	0	17	3,236	99.47
112	12	4	14	7	37	26	26	0	17	3,262	99.48
113	12	4	14	7	37	26	26	0	17	3,288	99.48
114	12	4	14	7	37	26	26	0	17	3,314	99.49
115	12	4	15	7	38	26	26	0	17	3,340	99.49
116	12	4	15	7	38	26	26	0	17	3,366	99.49
117	-11	4	15	7	-37	26	26	0	17	3,392	99.5
118	11	4	15	7	37	26	26	0	17	3,418	99.5
119	11	4	15	7	37	26	26	0	17	3,444	99.51
120	11	4	-14	7	-36	-25	26	1	18	3,470	99.48
121	11	4	-13	7	-35	-24	26	2	20	3,496	99.43
122	11	-3	13	7	-34	24	25	1	21	3,521	99.4
123	11	3	13	7	34	24	25	1	22	3,546	99.38
124	11	3	13	-6	-33	-23	24	1	23	3,570	99.36
125	11	3	13	6	33	23	24	1	24	3,594	99.33
126	11	3	-12	6	-32	-22	23	1	25	3,617	99.31
127	11	3	12	6	32	22	23	1	26	3,640	99.29
128	11	3	-11	6	-31	22	22	0	26	3,662	99.29
129	11	3	11	6	31	22	22	0	26	3,684	99.29
130	-10	3	11	6	-30	22	22	0	26	3,706	99.3

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
131	10	3	11	6	30	22	22	0	26	3,728	99.3
132	10	3	-10	6	-29	22	22	0	26	3,750	99.31
133	10	3	10	6	29	22	22	0	26	3,772	99.31
134	10	3	10	6	29	22	22	0	26	3,794	99.31
135	10	3	10	6	29	22	22	0	26	3,816	99.32
136	10	3	10	6	29	22	22	0	26	3,838	99.32
137	10	3	10	6	29	22	22	0	26	3,860	99.33
138	10	3	10	6	29	22	22	0	26	3,882	99.33
139	10	4	10	6	30	22	22	0	26	3,904	99.33
140	10	4	10	-5	-29	-21	22	1	27	3,926	99.31
141	-9	4	10	5	-28	-20	22	2	29	3,948	99.27
142	-8	4	10	5	-27	-19	22	3	32	3,970	99.19
143	-7	4	10	5	-26	19	19	0	32	3,989	99.2
144	7	4	10	5	26	19	19	0	32	4,008	99.2
145	-6	4	10	5	-25	-18	19	1	33	4,027	99.18
146	6	4	10	5	25	18	19	1	34	4,046	99.16
147	6	4	10	5	25	18	19	1	35	4,065	99.14
148	6	4	-9	5	-24	18	18	0	35	4,083	99.14
149	6	4	9	5	24	18	18	0	35	4,101	99.15
150	6	4	9	5	24	18	18	0	35	4,119	99.15
151	6	4	-8	-4	-22	18	18	0	35	4,137	99.15
152	6	-3	8	4	-21	-17	18	1	36	4,155	99.13
153	6	3	8	4	21	17	18	1	37	4,173	99.11
154	6	3	8	4	21	17	17	0	37	4,190	99.12
155	6	3	8	4	21	17	17	0	37	4,207	99.12
156	6	3	8	4	21	17	17	0	37	4,224	99.12
157	6	3	7	4	20	17	17	0	37	4,241	99.13
158	6	3	7	4	20	17	17	0	37	4,258	99.13
159	6	3	7	4	20	17	17	0	37	4,275	99.13
160	6	3	7	4	20	17	17	0	37	4,292	99.14
161	6	3	7	4	20	17	17	0	37	4,309	99.14
162	6	3	7	-3	-19	-16	17	1	38	4,326	99.12
163	6	3	-6	3	-18	-15	17	2	40	4,343	99.08
164	6	3	6	3	18	15	17	2	42	4,360	99.04
165	6	3	6	3	18	15	16	1	43	4,376	99.02
166	6	3	6	3	18	15	15	0	43	4,391	99.02
167	6	3	-4	-2	-15	-14	15	1	44	4,406	99
168	6	3	4	2	15	14	15	1	45	4,421	98.98
169	6	3	4	2	15	14	15	1	46	4,436	98.96
170	-5	3	4	2	-14	-13	14	1	47	4,450	98.94
171	5	3	4	2	14	13	14	1	48	4,464	98.92
172	5	3	4	2	14	13	14	1	49	4,478	98.91
173	5	3	4	2	14	13	13	0	49	4,491	98.91
174	5	3	-3	2	-13	-12	13	1	50	4,504	98.89
175	5	3	3	2	13	-11	13	2	52	4,517	98.85
176	-4	-2	3	2	-11	11	13	2	54	4,530	98.81
177	4	2	3	2	11	11	13	2	56	4,543	98.77
178	4	2	3	0	9	-10	12	2	58	4,555	98.73
179	4	2	3	0	9	-9	12	3	61	4,567	98.66

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
180	4	2	3	0	9	9	11	2	63	4,578	98.62
181	4	2	3	0	9	9	10	1	64	4,588	98.61
182	4	2	-2	0	-8	-8	10	2	66	4,598	98.56
183	4	2	2	0	8	8	9	1	67	4,607	98.55
184	4	2	2	0	8	8	9	1	68	4,616	98.53
185	4	2	2	0	8	8	8	0	68	4,624	98.53
186	4	3	2	0	9	8	8	0	68	4,632	98.53
187	4	3	2	0	9	8	8	0	68	4,640	98.53
188	4	3	2	0	9	8	8	0	68	4,648	98.54
189	4	3	2	0	9	-7	8	1	69	4,656	98.52
190	-3	-2	2	0	-7	-6	8	2	71	4,664	98.48
191	-2	2	-1	0	-5	-4	8	4	75	4,672	98.39
192	2	2	1	0	5	4	8	4	79	4,680	98.31
193	2	2	1	0	5	4	7	3	82	4,687	98.25
194	2	2	1	0	5	4	5	1	83	4,692	98.23
195	2	2	1	0	5	4	4	0	83	4,696	98.23
196	2	2	1	0	5	4	4	0	83	4,700	98.23
197	2	2	1	0	5	4	4	0	83	4,704	98.24
198	2	2	1	0	5	4	4	0	83	4,708	98.24
199	2	2	1	0	5	4	4	0	83	4,712	98.24
200	2	2	0	0	-4	4	4	0	83	4,716	98.24
201	2	2	0	0	4	-3	4	1	84	4,720	98.22
202	2	2	0	0	4	3	4	1	85	4,724	98.2
203	2	2	0	0	4	3	4	1	86	4,728	98.18
204	-1	2	0	0	-3	-2	4	2	88	4,732	98.14
205	1	2	0	0	3	2	3	1	89	4,735	98.12
206	1	2	0	0	3	2	3	1	90	4,738	98.1
207	1	2	0	0	3	2	2	0	90	4,740	98.1
208	1	2	0	0	3	2	2	0	90	4,742	98.1
209	1	2	0	0	3	2	2	0	90	4,744	98.1
210	1	2	0	0	3	2	2	0	90	4,746	98.1
211	1	2	0	0	3	2	2	0	90	4,748	98.1
212	1	2	0	0	3	2	2	0	90	4,750	98.11
213	1	2	0	0	3	2	2	0	90	4,752	98.11
214	1	2	0	0	3	2	2	0	90	4,754	98.11
215	1	2	0	0	3	2	2	0	90	4,756	98.11
216	1	2	0	0	3	2	2	0	90	4,758	98.11
217	1	2	0	0	3	2	2	0	90	4,760	98.11
218	1	-1	0	0	-2	-1	2	1	91	4,762	98.09
219	1	1	0	0	2	1	2	1	92	4,764	98.07
220	1	1	0	0	2	1	2	1	93	4,766	98.05
221	1	1	0	0	2	1	2	1	94	4,768	98.03
222	1	1	0	0	2	1	1	0	94	4,769	98.03
223	1	1	0	0	2	1	1	0	94	4,770	98.03
224	1	1	0	0	2	1	1	0	94	4,771	98.03
225	1	1	0	0	2	1	1	0	94	4,772	98.03
226	1	1	0	0	2	1	1	0	94	4,773	98.03
227	1	1	0	0	2	1	1	0	94	4,774	98.03
228	1	1	0	0	2	1	1	0	94	4,775	98.03

ตารางที่ 3 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 4											
เรดาร์ รอบที่	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศยานสะสม	ความถูกต้อง (%)
	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม		
229	1	1	0	0	2	1	1	0	94	4,776	98.03
230	1	1	0	0	2	1	1	0	94	4,777	98.03
231	1	1	0	0	2	1	1	0	94	4,778	98.03
232	1	1	0	0	2	1	1	0	94	4,779	98.03
233	1	1	0	0	2	1	1	0	94	4,780	98.03
234	1	1	0	0	2	1	1	0	94	4,781	98.03
235	1	1	0	0	2	1	1	0	94	4,782	98.03
236	1	1	0	0	2	1	1	0	94	4,783	98.03
237	0	0	0	0	0	1	1	0	94	4,784	98.04
238	0	0	0	0	0	0	1	1	95	4,785	98.01

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ง-4 การทดสอบครั้งที่ 5

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
0	11	7	+25	10	+53	+51	52	1	1	52	98.08
1	11	7	25	10	53	+52	52	0	1	104	99.04
2	11	7	-24	10	-52	52	52	0	1	156	99.36
3	11	7	+25	10	+53	52	52	0	1	208	99.52
4	11	7	25	+11	+54	52	52	0	1	260	99.62
5	11	7	25	11	54	52	52	0	1	312	99.68
6	11	7	25	11	54	52	52	0	1	364	99.73
7	11	7	25	11	54	52	52	0	1	416	99.76
8	+12	7	25	11	+55	52	52	0	1	468	99.79
9	12	7	25	11	55	52	52	0	1	520	99.81
10	12	7	25	11	55	52	52	0	1	572	99.83
11	+13	+8	25	11	+57	52	52	0	1	624	99.84
12	+15	8	25	11	+59	52	52	0	1	676	99.85
13	15	8	25	11	59	52	52	0	1	728	99.86
14	15	8	+26	11	+60	52	52	0	1	780	99.87
15	15	8	26	11	60	52	52	0	1	832	99.88
16	15	8	26	11	60	52	52	0	1	884	99.89
17	+16	8	26	+12	+62	52	52	0	1	936	99.89
18	+18	8	26	12	+64	52	52	0	1	988	99.9
19	18	8	+28	12	+66	52	52	0	1	1,040	99.9
20	18	8	28	12	66	52	52	0	1	1,092	99.91
21	18	+9	28	12	+67	52	52	0	1	1,144	99.91
22	+19	9	28	12	+68	52	52	0	1	1,196	99.92
23	19	9	28	12	68	-51	52	1	2	1,248	99.84
24	19	+10	28	+13	+70	+52	52	0	2	1,300	99.85
25	+20	10	+29	-12	+71	+53	52	1	3	1,352	99.78
26	20	-9	-28	12	-69	53	53	0	3	1,405	99.79
27	20	9	+29	+14	+72	53	53	0	3	1,458	99.79
28	+21	9	29	14	+73	53	53	0	3	1,511	99.8
29	21	9	29	14	73	53	53	0	3	1,564	99.81
30	21	9	29	14	73	53	53	0	3	1,617	99.81
31	21	9	+31	14	+75	+54	53	1	4	1,670	99.76
32	21	9	-30	14	-74	54	53	1	5	1,723	99.71
33	21	9	+31	14	+75	54	54	0	5	1,777	99.72
34	+22	9	31	+15	+77	54	54	0	5	1,831	99.73
35	+24	9	31	-14	+78	54	54	0	5	1,885	99.73
36	24	9	+32	14	+79	-53	54	1	6	1,939	99.69
37	+26	9	32	14	+81	+54	54	0	6	1,993	99.7
38	26	9	32	14	81	54	54	0	6	2,047	99.71
39	26	+10	+33	14	+83	54	54	0	6	2,101	99.71

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
40	26	10	33	+15	+84	54	54	0	6	2,155	99.72
41	26	10	33	15	84	+55	54	1	7	2,209	99.68
42	+27	10	-32	15	84	-54	54	0	7	2,263	99.69
43	27	10	32	15	84	+56	54	2	9	2,317	99.61
44	27	10	32	15	84	56	56	0	9	2,373	99.62
45	27	10	32	+16	+85	56	56	0	9	2,429	99.63
46	27	+11	32	16	+86	56	56	0	9	2,485	99.64
47	27	+12	-31	16	86	56	56	0	9	2,541	99.65
48	27	12	31	16	86	56	56	0	9	2,597	99.65
49	27	12	31	+17	+87	+57	56	1	10	2,653	99.62
50	27	12	31	17	87	-56	56	0	10	2,709	99.63
51	27	12	31	17	87	56	56	0	10	2,765	99.64
52	27	12	31	17	87	56	56	0	10	2,821	99.65
53	27	12	+32	17	+88	56	56	0	10	2,877	99.65
54	+28	12	32	17	+89	+57	56	1	11	2,933	99.62
55	28	12	32	17	89	57	57	0	11	2,990	99.63
56	28	12	32	+19	+91	57	57	0	11	3,047	99.64
57	28	12	32	-18	-90	57	57	0	11	3,104	99.65
58	28	12	32	18	90	57	57	0	11	3,161	99.65
59	28	12	32	18	90	57	57	0	11	3,218	99.66
60	28	12	32	18	90	57	57	0	11	3,275	99.66
61	+29	-11	32	+19	+91	57	57	0	11	3,332	99.67
62	29	11	+33	19	+92	57	57	0	11	3,389	99.68
63	-28	11	-32	19	-90	57	57	0	11	3,446	99.68
64	28	-10	32	+20	90	-56	57	1	12	3,503	99.66
65	28	10	32	20	90	56	56	0	12	3,559	99.66
66	28	10	32	20	90	56	56	0	12	3,615	99.67
67	28	10	32	20	90	56	56	0	12	3,671	99.67
68	28	10	32	20	90	56	56	0	12	3,727	99.68
69	28	10	32	20	90	56	56	0	12	3,783	99.68
70	28	10	32	-19	-89	56	56	0	12	3,839	99.69
71	28	10	32	-18	-88	-55	56	1	13	3,895	99.67
72	-27	10	32	18	-87	-54	56	2	15	3,951	99.62
73	+28	10	32	18	+88	+55	55	0	15	4,006	99.63
74	-27	-9	+33	18	-87	55	55	0	15	4,061	99.63
75	27	9	33	18	87	55	55	0	15	4,116	99.64
76	27	-8	33	18	-86	55	54	1	16	4,170	99.62
77	27	8	-32	18	-85	55	54	1	17	4,224	99.6
78	27	8	-30	18	-83	55	54	1	18	4,278	99.58
79	27	8	+31	18	+84	55	54	1	19	4,332	99.56
80	27	8	31	+19	+85	55	54	1	20	4,386	99.54
81	27	8	-30	19	-84	55	54	1	21	4,440	99.53
82	27	8	30	19	84	55	54	1	22	4,494	99.51
83	27	8	30	19	84	55	54	1	23	4,548	99.49
84	27	8	+31	19	+85	55	54	1	24	4,602	99.48
85	27	8	31	-18	-84	-54	54	0	24	4,656	99.48
86	27	8	31	18	84	54	54	0	24	4,710	99.49
87	-26	8	31	18	-83	54	54	0	24	4,764	99.5
88	26	8	31	18	83	54	54	0	24	4,818	99.5

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
89	26	8	31	18	83	54	53	1	25	4,871	99.49
90	26	-7	31	18	-82	54	53	1	26	4,924	99.47
91	26	7	31	-17	-81	-53	53	0	26	4,977	99.48
92	-25	7	31	+18	81	53	53	0	26	5,030	99.48
93	25	7	-30	-17	-79	53	52	1	27	5,082	99.47
94	25	7	30	17	79	-51	52	1	28	5,134	99.45
95	25	7	30	17	79	51	52	1	29	5,186	99.44
96	25	7	-29	17	-78	51	50	1	30	5,236	99.43
97	25	7	29	17	78	-50	50	0	30	5,286	99.43
98	25	7	29	17	78	50	50	0	30	5,336	99.44
99	25	-6	29	17	-77	50	49	1	31	5,385	99.42
100	25	6	29	17	77	50	49	1	32	5,434	99.41
101	25	6	-28	17	-76	50	49	1	33	5,483	99.4
102	25	6	28	17	76	50	49	1	34	5,532	99.39
103	-24	6	28	17	-75	50	49	1	35	5,581	99.37
104	-23	6	-27	17	-73	-49	49	0	35	5,630	99.38
105	23	6	-26	-16	-71	-48	48	0	35	5,678	99.38
106	23	-5	26	16	-70	-47	48	1	36	5,726	99.37
107	23	5	26	-15	-69	47	48	1	37	5,774	99.36
108	23	5	26	-14	-68	-46	46	0	37	5,820	99.36
109	23	5	-25	14	-67	46	46	0	37	5,866	99.37
110	23	5	25	14	67	46	45	1	38	5,911	99.36
111	23	5	+26	14	+68	46	45	1	39	5,956	99.35
112	23	5	-25	14	-67	46	45	1	40	6,001	99.33
113	23	5	25	14	67	46	45	1	41	6,046	99.32
114	23	5	+26	14	+68	46	45	1	42	6,091	99.31
115	23	+6	26	14	+69	46	45	1	43	6,136	99.3
116	-22	6	26	14	-68	46	45	1	44	6,181	99.29
117	-21	6	26	14	-67	46	45	1	45	6,226	99.28
118	21	6	26	14	67	-45	45	0	45	6,271	99.28
119	21	6	-24	14	-65	45	45	0	45	6,316	99.29
120	21	6	-23	14	-64	-44	44	0	45	6,360	99.29
121	21	-5	23	14	-63	-43	44	1	46	6,404	99.28
122	21	5	23	14	63	43	43	0	46	6,447	99.29
123	21	5	23	-13	-62	43	43	0	46	6,490	99.29
124	21	+6	23	13	+63	-42	42	0	46	6,532	99.3
125	21	6	-22	13	-62	-41	42	1	47	6,574	99.29
126	21	6	-21	13	-61	41	41	0	47	6,615	99.29
127	-20	6	-20	13	-59	41	41	0	47	6,656	99.29
128	20	6	20	13	59	41	40	1	48	6,696	99.28
129	-19	6	20	13	-58	41	40	1	49	6,736	99.27
130	19	-5	20	13	-57	41	40	1	50	6,776	99.26
131	19	5	-19	13	-56	41	40	1	51	6,816	99.25
132	19	5	19	13	56	41	40	1	52	6,856	99.24
133	19	5	19	13	56	41	40	1	53	6,896	99.23
134	19	5	19	13	56	41	40	1	54	6,936	99.22
135	19	5	19	13	56	41	40	1	55	6,976	99.21
136	-18	5	19	13	-55	-40	40	0	55	7,016	99.22
137	18	5	19	13	55	40	40	0	55	7,056	99.22

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
138	18	+6	19	-12	55	40	40	0	55	7,096	99.22
139	18	6	19	12	55	40	39	1	56	7,135	99.22
140	-16	6	19	12	-53	-38	39	1	57	7,174	99.21
141	-14	6	-18	12	-50	-36	39	3	60	7,213	99.17
142	14	6	18	-11	-49	36	39	3	63	7,252	99.13
143	-13	6	-17	11	-47	36	35	1	64	7,287	99.12
144	13	6	17	11	47	-35	35	0	64	7,322	99.13
145	13	6	17	-10	-46	35	35	0	64	7,357	99.13
146	13	6	-16	10	-45	-34	35	1	65	7,392	99.12
147	13	6	16	10	45	34	35	1	66	7,427	99.11
148	13	6	16	10	45	34	34	0	66	7,461	99.12
149	13	6	-15	-9	-43	34	33	1	67	7,494	99.11
150	13	-5	15	9	-42	34	33	1	68	7,527	99.1
151	13	5	15	9	42	-33	33	0	68	7,560	99.1
152	13	5	15	9	42	33	33	0	68	7,593	99.1
153	13	5	15	-8	-41	33	33	0	68	7,626	99.11
154	13	5	-14	+9	41	+34	32	2	70	7,658	99.09
155	13	5	-12	9	-39	-32	32	0	70	7,690	99.09
156	-12	5	12	9	-38	32	32	0	70	7,722	99.09
157	-11	5	12	9	-37	-31	31	0	70	7,753	99.1
158	-10	5	12	9	-36	31	31	0	70	7,784	99.1
159	10	5	12	9	36	-30	31	1	71	7,815	99.09
160	10	5	12	-8	-35	30	30	0	71	7,845	99.09
161	10	5	12	-7	-34	30	30	0	71	7,875	99.1
162	10	5	-11	-6	-32	-29	29	0	71	7,904	99.1
163	10	5	11	+7	+33	-27	29	2	73	7,933	99.08
164	10	-4	11	7	-32	+28	29	1	74	7,962	99.07
165	10	4	11	7	32	-27	28	1	75	7,990	99.06
166	-9	4	-9	-6	-28	27	27	0	75	8,017	99.06
167	9	4	9	6	28	-25	27	2	77	8,044	99.04
168	9	4	9	6	28	25	27	2	79	8,071	99.02
169	-8	4	9	6	-27	25	26	1	80	8,097	99.01
170	8	4	9	6	27	-24	24	0	80	8,121	99.01
171	8	4	9	6	27	24	24	0	80	8,145	99.02
172	8	4	9	6	27	24	24	0	80	8,169	99.02
173	8	4	-8	-5	-25	24	23	1	81	8,192	99.01
174	8	4	8	5	25	-22	23	1	82	8,215	99
175	8	4	8	5	25	22	23	1	83	8,238	98.99
176	-7	-2	8	5	-22	-20	23	3	86	8,261	98.96
177	7	2	8	5	22	20	22	2	88	8,283	98.94
178	7	2	8	5	22	20	21	1	89	8,304	98.93
179	7	2	8	-3	-20	-19	20	1	90	8,324	98.92
180	7	2	8	2	19	-17	19	2	92	8,343	98.9
181	7	2	-7	2	-18	17	18	1	93	8,361	98.89
182	7	2	7	2	18	-16	18	2	95	8,379	98.87
183	7	2	7	2	18	16	16	0	95	8,395	98.87
184	7	2	7	2	18	16	16	0	95	8,411	98.87
185	-6	2	7	2	-17	16	15	1	96	8,426	98.86
186	6	+3	7	2	+18	16	15	1	97	8,441	98.85

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
187	6	3	7	2	18	16	15	1	98	8,456	98.84
188	6	3	7	2	18	16	15	1	99	8,471	98.83
189	6	3	7	2	18	16	15	1	100	8,486	98.82
190	-5	-2	7	2	-16	-14	15	1	101	8,501	98.81
191	-4	2	-6	2	-14	-12	15	3	104	8,516	98.78
192	4	2	6	2	14	12	15	3	107	8,531	98.75
193	4	2	6	2	14	12	14	2	109	8,545	98.72
194	4	2	6	2	14	12	12	0	109	8,557	98.73
195	4	2	6	2	14	12	11	1	110	8,568	98.72
196	4	2	6	2	14	12	11	1	111	8,579	98.71
197	4	2	6	2	14	12	11	1	112	8,590	98.7
198	4	2	6	-1	-13	-11	11	0	112	8,601	98.7
199	4	2	-5	1	-12	-10	11	1	113	8,612	98.69
200	4	2	5	1	12	10	11	1	114	8,623	98.68
201	4	2	5	1	12	10	10	0	114	8,633	98.68
202	4	2	-4	1	-11	-9	9	0	114	8,642	98.68
203	4	2	4	-0	-10	9	9	0	114	8,651	98.68
204	-3	2	4	+1	10	-8	9	1	115	8,660	98.67
205	3	2	4	1	10	8	8	0	115	8,668	98.67
206	3	2	4	1	10	8	8	0	115	8,676	98.67
207	3	2	4	1	10	8	7	1	116	8,683	98.66
208	3	2	4	1	10	8	7	1	117	8,690	98.65
209	3	2	4	1	10	8	7	1	118	8,697	98.64
210	3	2	4	1	10	8	7	1	119	8,704	98.63
211	3	2	4	1	10	8	7	1	120	8,711	98.62
212	3	2	4	1	10	8	7	1	121	8,718	98.61
213	3	2	4	1	10	8	7	1	122	8,725	98.6
214	3	2	4	1	10	8	7	1	123	8,732	98.59
215	3	2	4	1	10	8	7	1	124	8,739	98.58
216	3	2	4	1	10	8	7	1	125	8,746	98.57
217	3	2	4	1	10	8	7	1	126	8,753	98.56
218	3	2	4	1	10	8	7	1	127	8,760	98.55
219	-2	-1	4	1	+8	-7	7	0	127	8,767	98.55
220	2	1	4	1	8	7	7	0	127	8,774	98.55
221	2	1	4	1	8	7	7	0	127	8,781	98.55
222	2	1	4	1	8	7	6	1	128	8,787	98.54
223	2	1	4	1	8	7	6	1	129	8,793	98.53
224	2	1	4	1	8	7	6	1	130	8,799	98.52
225	2	1	4	1	8	7	6	1	131	8,805	98.51
226	2	1	4	1	8	7	6	1	132	8,811	98.5
227	2	1	4	1	8	7	6	1	133	8,817	98.49
228	2	1	4	1	8	7	6	1	134	8,823	98.48
229	2	1	4	1	8	7	6	1	135	8,829	98.47
230	2	1	4	1	8	7	6	1	136	8,835	98.46
231	2	1	4	1	8	7	6	1	137	8,841	98.45
232	2	1	4	1	8	7	6	1	138	8,847	98.44
233	2	1	4	1	8	7	6	1	139	8,853	98.43
234	2	1	-3	1	-7	7	6	1	140	8,859	98.42
235	2	1	+4	1	+8	7	6	1	141	8,865	98.41

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5

เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
236	2	1	4	1	8	7	6	1	142	8,871	98.4
237	2	1	4	1	8	-6	6	0	142	8,877	98.4
238	2	-0	4	-0	-6	6	6	0	142	8,883	98.4
239	-1	0	4	0	-5	-5	6	1	143	8,889	98.39
240	1	0	4	+1	+6	5	5	0	143	8,894	98.39
241	1	0	4	1	6	5	5	0	143	8,899	98.39
242	1	0	4	1	6	5	4	1	144	8,903	98.38
243	1	0	-3	1	-5	-4	4	0	144	8,907	98.38
244	1	0	3	1	5	4	4	0	144	8,911	98.38
245	1	0	3	1	5	4	4	0	144	8,915	98.38
246	1	0	3	1	5	4	4	0	144	8,919	98.39
247	1	0	3	1	5	4	3	1	145	8,922	98.37
248	1	0	-2	1	-4	4	3	1	146	8,925	98.36
249	1	0	2	1	4	-3	3	0	146	8,928	98.36
250	1	0	2	1	4	3	3	0	146	8,931	98.37
251	1	0	2	1	4	3	2	1	147	8,933	98.35
252	1	0	2	1	4	3	2	1	148	8,935	98.34
253	1	0	2	-0	-3	3	2	1	149	8,937	98.33
254	-0	0	-1	0	-1	-2	2	0	149	8,939	98.33
255	0	0	1	0	1	2	2	0	149	8,941	98.33
256	0	0	1	0	1	2	2	0	149	8,943	98.33
257	0	0	1	0	1	2	2	0	149	8,945	98.33
258	0	0	1	0	1	2	1	1	150	8,946	98.32
259	0	0	1	0	1	2	1	1	151	8,947	98.31
260	0	0	1	0	1	2	1	1	152	8,948	98.3
261	0	0	1	0	1	2	1	1	153	8,949	98.29
262	0	0	1	0	1	2	1	1	154	8,950	98.28
263	0	0	1	0	1	2	1	1	155	8,951	98.27
264	0	0	1	0	1	2	1	1	156	8,952	98.26
265	0	0	1	0	1	2	1	1	157	8,953	98.25
266	0	0	1	0	1	2	1	1	158	8,954	98.24
267	0	0	1	0	1	2	1	1	159	8,955	98.22
268	0	0	1	0	1	2	1	1	160	8,956	98.21
269	0	0	1	0	1	2	1	1	161	8,957	98.2
270	0	0	1	0	1	2	1	1	162	8,958	98.19
271	0	0	1	0	1	2	1	1	163	8,959	98.18
272	0	0	1	0	1	2	1	1	164	8,960	98.17
273	0	0	1	0	1	2	1	1	165	8,961	98.16
274	0	0	1	0	1	2	1	1	166	8,962	98.15
275	0	0	1	0	1	2	1	1	167	8,963	98.14
276	0	0	1	0	1	2	1	1	168	8,964	98.13
277	0	0	1	0	1	2	1	1	169	8,965	98.11
278	0	0	1	0	1	2	1	1	170	8,966	98.1
279	0	0	1	0	1	2	1	1	171	8,967	98.09
280	0	0	1	0	1	2	1	1	172	8,968	98.08
281	0	0	1	0	1	2	1	1	173	8,969	98.07
282	0	0	1	0	1	2	1	1	174	8,970	98.06
283	0	0	1	0	1	2	1	1	175	8,971	98.05
284	0	0	1	0	1	2	1	1	176	8,972	98.04

ตารางที่ 4 แสดงจำนวนอากาศยานและความถูกต้องในแต่ละรอบของการทดสอบครั้งที่ 5											
เรดาร์	จำนวนอากาศยานของเรดาร์หมายเลข				จำนวนอากาศยาน			ความผิดพลาด		ยอดอากาศ	ความถูกต้อง
รอบที่	# 1	# 2	# 3	# 4	ระบบเดิม	ระบบใหม่	จริง	ต่อรอบ	สะสม	ยานสะสม	(%)
285	0	0	1	0	1	2	1	1	177	8,973	98.03
286	0	0	1	0	1	2	1	1	178	8,974	98.02
287	0	0	-0	0	-0	-1	1	0	178	8,975	98.02

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายนุกูล ทวีศักดิ์เกิดที่อำเภอถลาง จังหวัดภูเก็ต สำเร็จการศึกษาปริญญาตรี สาขาวิทยาศาสตร์คอมพิวเตอร์ Concordia University (Montreal, Canada) เข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต (นอกเวลาราชการ) สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยเมื่อ พ.ศ. 2541

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย