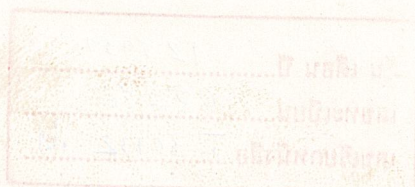


Air Traffic Approach Control Simulator



โดย

นางสาว สมจิต นาคี 33100-389

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

033256

จำลองงานควบคุมการจราจรอากาศในเขตประชิดสนามบินและเขตสนามบิน

Air Traffic Approach Control Simulator

นางสาวสมจิต นาดิ

อาจารย์ที่ปรึกษา อ.สมยศ จุณณะปิยะ

ปีการศึกษา 2536

บทคัดย่อ

โดยทั่วไปการขนส่งทางอากาศ นับว่ามีความสำคัญและมีความเจริญก้าวหน้าไปอย่างรวดเร็วเมื่อเทียบกับขนส่งอื่น ๆ องค์ประกอบสำคัญที่ช่วยให้การขนส่งเป็นไปอย่างรวดเร็ว ปลอดภัย และประหยัด ย่อมขึ้นอยู่กับปัจจัยหลายอย่าง เช่น อากาศยาน, อุปกรณ์ที่ติดตั้งบนอากาศยาน, นักบิน, เจ้าหน้าที่ควบคุมการจราจรทางอากาศ, เครื่องช่วยการเดินอากาศ, อุปกรณ์ที่ใช้ในการควบคุมการจราจรทางอากาศ เป็นต้น

เจ้าหน้าที่ควบคุมการจราจรทางอากาศเป็นส่วนหนึ่งที่สำคัญมาก เพราะเจ้าหน้าที่ควบคุมฯ มีหน้าที่รับผิดชอบต่ออากาศยานทุกเครื่องในเขตรับผิดชอบ ป้องกันการชนกันระหว่าง อากาศยานกับอากาศยาน และอากาศยานกับสิ่งกีดขวางตลอดจนควบคุมให้การจราจรทางอากาศดำเนินไปอย่างรวดเร็วและมีระเบียบ

จึงได้เสนอโครงการแอร์ทราฟิกส์แอฟพรอชคอนโทรลซิมูเลเตอร์ (Air Traffic Approach Control Simulator) ขึ้นมาช่วยให้เจ้าหน้าที่ควบคุมการจราจรทางอากาศ สำหรับฝึกหัดและเกิดทักษะในการควบคุมอากาศยานให้สามารถร่อนลงสู่สนามบินได้โดยสวัสดิภาพ

ABSTRACT

THE AIR TRANSPORTATION HAVE AN IMPORTANT AND DEVELOPED MORE THAN OTHERS TRANSPORTATION. IT HAS MANY FACTORS FOR CONVENIENCE AND SAFETY TRANSPORTATION ETC AIRCRAFT TRANSPONDER, PILOT, AIR TRAFFIC CONTROLLER, INSTRUMENT LANDING SYSTEM, THE AIR CONTROL EQUIPMENT THE AIR TRAFFIC CONTROLLER IS AN IMPORTANT PART BECAUSE THEY ARE MUST BE CONTROL THE AIRCRAFT AND AIR TRAFFIC TO PROTECT THE COLLISION BETWEEN AIRCRAFT-TO-AIRCRAFT AND AIRCRAFT-TO-THE OTHERS TARGET THROUGHOUT CONTROLLER THE AIR TRANSPORTATION IN ORDER TO THE LEAST TIME PROCESS AND CONVENIENCE PROCESS ALSO THE AIR TRAFFIC APPROACH CONTROL SIMULATOR IS OFFERED FOR PRACTISE AND SKILL TO CONTROL THE AIRCRAFT WITH THE SAFETY FOR THE AIR CONTROLLER

สารบัญ

บทที่	หน้าที่
บทที่ 1	บทนำ
บทที่ 2	งานควบคุมการจราจรทางอากาศ
บทที่ 3	วิทยุเครื่องช่วยการเดินอากาศ
บทที่ 4	การเขียนโปรแกรมกราฟิกส์
บทที่ 5	การแปลงภาพ
บทที่ 6	รูปแบบการแสดงผลหน้าจอคอมพิวเตอร์
บทที่ 7	สรุปวิจารณ์

บทที่ 1

บทนำ

ในปัจจุบันที่ สภาวะเศรษฐกิจและสังคม มีความเจริญก้าวหน้าไปตามการพัฒนาทางเทคโนโลยีต่าง ๆ นั้น การดำเนินธุรกิจต่าง ๆ มีการแข่งขันกันตลอดเวลา ความต้องการความสะดวกสบายและความรวดเร็ว ในการติดต่อสื่อสารต่าง ๆ ย่อมมีมากขึ้นและเนื่องจากการขนส่งทางอากาศและการจราจรทางอากาศเป็น ปัจจัยที่สำคัญอย่างหนึ่งของการติดต่อสื่อสารที่ความรวดเร็วและมีประสิทธิภาพสูง ประกอบกับการขนส่ง ทางอากาศเป็นการขนส่งที่มีมูลค่าขององค์ประกอบต่าง ๆ ที่ใช้ในการขนส่ง มีราคาแพงมาก ความปลอดภัย ในการขนส่งทางอากาศจึงนับว่ามีความสำคัญอย่างยิ่ง

จากเหตุผลที่ต้องการความรวดเร็ว และความปลอดภัยในการขนส่งทางอากาศทุก ๆ ประเทศทั่วโลกจะใช้ มาตรฐานและระเบียบการจราจรทางอากาศอันเดียวกันที่เป็นข้อตกลงร่วมกันระหว่างนานาชาติทั่วโลก และ จัดให้มีสถานีบริการข้อมูลข่าวสาร ที่เกี่ยวข้องกับการบินและควบคุมการจราจรทางอากาศในเขตพื้นที่รับผิดชอบ ของสถานี ควบคุมฯ และงานทางด้านควบคุมการจราจรทางอากาศนั้นมีส่วนเกี่ยวข้องโดยตรงกับเจ้าหน้าที่ควบคุมการจราจรทางอากาศ เพื่อเป็นการฝึกฝนความชำนาญในงานควบคุมฯ ให้กับเจ้าหน้าที่ควบคุม การจราจรทางอากาศ จึงนับว่าสำคัญอย่างยิ่ง

สำหรับลักษณะของโครงการแอร์ทราฟฟิกแอฟพรอชคอนโทรลซิมูเลเตอร์(Air Traffic Approach Control Simulator) นี้เป็นการจำลองสภาพการร่อนลงสู่สนามบินของเครื่องบิน ออกทงหน้าจอ คอมพิวเตอร์ จุดประสงค์เพื่อให้เจ้าหน้าที่ควบคุมการจราจรทางอากาศ ได้ใช้ฝึกหัดงานควบคุมการร่อนลงสู่ สนามบินของเครื่องบิน ให้เกิดความคุ้นเคยและความชำนาญก่อนที่จะปฏิบัติงานจริง

เนื่องจากการจำลองการทำงานของเรดาร์ที่ใช้ในการควบคุมการร่อนลงสู่สนามบินของเครื่องบิน สิ่ง ต่าง ๆ ที่เกี่ยวข้องกับรายละเอียดต่าง ๆ ของโครงการ ได้แก่ รายละเอียดของเรดาร์ที่ใช้ในงานควบคุมการร่อน ลงสู่สนามบินของเครื่องบิน,รายละเอียดเกี่ยวกับหน้าที่ความรับผิดชอบต่าง ๆ ของเจ้าหน้าที่ควบคุมการจราจร ทางอากาศ,ที่เกี่ยวข้องและเป็นแนวทางที่ใช้ในการเขียนโปรแกรมเพื่อให้ได้การทำงานของโครงการโดยมีรายละเอียดดังจะกล่าวในบทต่อไป.

บทที่ 2

เรดาร์กับงานควบคุมจราจรทางอากาศ

เรดาร์เป็นอุปกรณ์สำคัญอย่างหนึ่งที่ใช้ในการติดตามสภาพการจราจรของเครื่องบินซึ่งบินอยู่ในเขตควบคุมการบินของประเทศ (Bangkok Flight Information Region) โดยจะทำให้ผู้ควบคุมจราจรทางอากาศได้ทราบถึง ทิศทาง ความสูง และความเร็ว ตลอดจนข้อมูลที่เกี่ยวข้องและจำเป็นอื่น ๆ อีกมากทำให้การควบคุมจราจรทางอากาศเป็นไปได้ด้วยความปลอดภัยรวดเร็วและมีประสิทธิภาพสูง ได้มาตรฐานสากลตามที่ องค์การการบินพลเรือนระหว่างประเทศได้กำหนดไว้

2.1 ประวัติความเป็นมา

แนวความคิดพื้นฐานของเรดาร์ถูกนำมาใช้ครั้งแรกในปี พ.ศ. 2469 โดยในครั้งมีจุดประสงค์เพื่อวัดความสูงของชั้นบรรยากาศไอโอโนสเฟียร์ (Ionosphere) ซึ่งเป็นชั้นบรรยากาศที่สะท้อนคลื่นวิทยุบางความถี่ได้ แต่อุปกรณ์ที่เป็นเรดาร์จริง ๆ นั้น เกิดขึ้นระหว่างสงครามโลกครั้งที่ 2 ประมาณปี พ.ศ. 2473 เป็นไปเพื่อประโยชน์ทางการทหาร ในช่วงของสงคราม กองเรืออังกฤษก็มีการใช้งานเรดาร์กันแล้ว ทั้งนี้ทั้งนั้นโดยความร่วมมือกันพัฒนาระหว่างสหรัฐอเมริกากับอังกฤษ ต่อมาในปี พ.ศ. 2483 เรดาร์ได้พัฒนาประสิทธิภาพหลอดแมกนีตรอน (magnetron tube) ซึ่งนำมาใช้งานเป็นออสซิลเลเตอร์กำลังสูงในย่านไมโครเวฟ สำหรับเครื่องส่งเรดาร์แทนการใช้หลอดโคลสตรอน (klystron tube) และ ทิวป์ลิวท์ (TWT: travelling wavel tube) แบบเดิมที่มีกำลังส่งต่ำไปไม่ได้ไมไกล หลังจากสงครามโลกครั้งที่ 2 แล้วเรดาร์ก็แพร่หลายเข้ามาใช้งานในหมู่พลเรือนจนกลายเป็นอุปกรณ์ที่จำเป็นต่อมนุษย์ในปัจจุบัน

2.2 หลักการเบื้องต้นของเรดาร์

เรดาร์ (RADAR) มาจากคำว่า Radio Detection And Ranging เป็นวิธีการหาตำแหน่งและเป้าของวัตถุโดยใช้คลื่นวิทยุ เรดาร์อาศัยหลักการเบื้องต้นของการสะท้อนของคลื่นแม่เหล็กไฟฟ้า เช่นเดียวกับธรรมชาติของสัตว์บางประเภท เช่น ค้างคาว อาศัยคลื่นเสียงที่ส่งออกไปกระทบเป้าแล้วสะท้อนกลับมาเป็นตัวกำหนดตำแหน่งและทิศทางของเป้าหมาย ๆ

สำหรับการทำงานของเรดาร์นั้นได้จากการส่งคลื่นแม่เหล็กไฟฟ้าในช่วงสั้น ๆ (pulse) คลื่นแม่เหล็กไฟฟ้านั้นจะมีความเร็วของการเดินทางในอากาศเท่ากับความเร็วของแสงเมื่อเดินทางไปกระทบเป้าจะสะท้อนกลับมายังสถานีส่ง ฉะนั้นระยะทางระหว่างเป้ากับสถานีส่งจะทราบได้จากระยะเวลาที่สูญเสียไป โดยเริ่มนับจากเวลาที่ส่งคลื่นออกจากสถานีส่ง จนถึงระยะเวลาที่รับคลื่นสะท้อนกลับมาจากเป้าได้ ซึ่งสามารถแทนได้จากสูตรดังนี้

$$R = \frac{Ct}{2}$$

R = ระยะทาง (nautical miles)

c = ความเร็วแสง ($162 \cdot 10^3$ nautical miles/sec.)

t = ระยะเวลา (second)

โดยที่ 1 N.M. = 1830 เมตร

2.3 ประเภทของเรดาร์

หากจะแบ่งตามลักษณะการทำงานแล้ว เรดาร์จะถูกแบ่งออกได้เป็น

1. ซีดับเบิลเรดาร์ (CW RADAR หรือ continuous wave radar) ลักษณะคือคลื่นแม่เหล็กไฟฟ้าที่ส่งออกไปจะต่อเนื่องกันตลอดเวลา จุดประสงค์ใช้งานในการตรวจจับความเร็วโดยอาศัยพื้นฐานจากปรากฏการณ์การดอปเปลอร์ (doppler) ใช้ในการตรวจสอบความเร็วหรือความแรงของวัตถุ
2. เอฟเอ็ม เรดาร์ (FM RADAR) : ใช้วัดความสูง สำหรับการติดตั้งในอากาศยาน
3. พัลส์เรดาร์ (Pulse Radar) มีลักษณะคือ เป็นคลื่นแม่เหล็กไฟฟ้าที่ส่งออกไปจะเป็นช่วงเวลานั้นมาก ๆ เว้นระยะห่างเท่า ๆ กัน ใช้ในการตรวจจับตำแหน่ง และระยะทางของวัตถุหรือในบางรุ่นออก

แบบให้สามารถนำข้อมูลที่อ่านได้มาคำนวณความเร็วได้ด้วย ใช้ในงานควบคุมจราจรทางอากาศ ในที่นี้จะขอล่าถึงแต่ Pulse Radar ที่นำมาใช้ในงานควบคุมจราจรทางอากาศเท่านั้น ซึ่งจะแบ่งออกได้เป็น 2 ชนิดด้วยกันคือ

- 1.Primary Radar:เรดาร์ปฐมภูมิ
- 2.Secondary Surveillance Radar:เรดาร์ทุติยภูมิ

เรดาร์ปฐมภูมิ (Primary Radar)

จะใช้สำหรับงานควบคุมจราจรทางอากาศในบริเวณสนามบิน (Aerodrome control) และเขตประชิดสนามบิน (Approach control) รัศมีทำการระหว่าง 5-35 นอตีคอลไมล์(N.M.) จากหอบังคับการบินโดยระยะทางและตำแหน่งของเครื่องบินจะปรากฏให้เห็นบนจอเรดาร์

เรดาร์ปฐมภูมิอาศัยหลักการสะท้อนกลับของคลื่นแม่เหล็กไฟฟ้ากล่าวคือ เครื่องส่งวิทยุจะส่งคลื่นวิทยุออกไปในอากาศ และเมื่อกระทบวัตถุที่กีดขวางจะสะท้อนพลังงานส่วนหนึ่งกลับมายังสถานีส่ง เครื่องรับจะรับสัญญาณที่สะท้อนกลับมานี้ และจะนำมาเปลี่ยนรูปโดยแสดงให้เห็นตำแหน่งและทิศทางบนจอเรดาร์ จากการอาศัยการสะท้อนกลับของคลื่นสำหรับระบบเรดาร์แบบปฐมภูมินี้ จึงทำให้ไม่สามารถแยกแยะสัญญาณพิสูจน์ฝ่ายของเครื่องบินที่จับเป้าได้ กล่าวคือ เรดาร์จะแสดงให้เห็นแต่ตำแหน่งและทิศทางของเป้าเท่านั้น ไม่อาจบ่งบอกได้ว่าเป็นเครื่องบินของมิตรหรือศัตรู ข้อเสียอีกประการหนึ่งของเรดาร์ชนิดนี้คือ จะมีการสูญเสียของพลังงานมาก เนื่องจากเครื่องส่งจะต้องส่งออกอากาศโดยใช้กำลังสูง เพื่อจะรับสัญญาณกลับไปในขนาดพอเพียงที่จะแสดงผล ให้เห็นบนจอเรดาร์

เรดาร์ทุติยภูมิ (Secondary Surveillance Radar)

ใช้สำหรับงานควบคุมจราจรทางอากาศสำหรับเครื่องบินที่บิน ในเขตควบคุมการบินของประเทศ (Area control) โดยยกเว้นเขตประชิดท่าอากาศยาน

สำหรับการทำงาน อุปกรณ์ภาคพื้นดินจะส่งสัญญาณถาม (Interrogation) ไปยังเครื่องบินและเครื่องรับ/เครื่องส่งบนเครื่องบิน (Transponder) จะส่งสัญญาณตอบรับมายังสถานีภาคพื้นดิน สัญญาณตอบรับเมื่อนำมาแปลรหัสแล้ว จะปรากฏให้เห็นเป็นเครื่องหมายบนจอเรดาร์ซึ่งจะทำให้ทราบได้ว่าเป็นมิตรหรือศัตรู

แม้เรดาร์ระบบนี้จะมีหลักการเดียวกับเรดาร์ทั่วไป แต่ก็มีข้อแตกต่างอย่างสำคัญจากเรดาร์ปฐมภูมิ เนื่องจากการใช้ทรานสปอนเดอร์บนเครื่องบินเพื่อก่อให้เกิดสัญญาณตอบรับ ความสูญเสียเนื่องจากการสะท้อนของคลื่นวิทยุของระบบเรดาร์ปฐมภูมิ จะถูกขจัดให้หมดไป ทำให้เครื่องส่งสัญญาณที่จุดปลายทั้งสองข้าง (ถามและตอบ) สามารถทำงานได้ผลดีที่กำลังส่งที่ต่ำกว่าเรดาร์ปฐมภูมิมาก และการใช้เครื่องส่ง 2 ตัวแยกกัน ทำให้ส่งสัญญาณถามได้ทีความถี่หนึ่ง และรับสัญญาณได้ทีอีกความถี่หนึ่ง ทำให้ขจัดภาพลวงที่เกิดจากการสะท้อนคลื่นได้

ความแตกต่างที่สำคัญอีกประการหนึ่งคือ เรดาร์ทุติยภูมิไม่อาจทำงานได้ โดยปราศจากความร่วมมือจากเป้า นั่นคือ เพื่อที่จะรับสัญญาณตอบรับจากเครื่องบินได้ เครื่องบินนั้นจะต้องติดตั้งอุปกรณ์ทรานสปอนเดอร์ด้วย

เครื่องส่งเรดาร์ภาคพื้นดิน จะส่งสัญญาณถามไปยังทรานสปอนเดอร์(Transponder) ที่อยู่บนเครื่องบิน โดยส่ง Pulse เป็นคู่ที่ความถี่ 1,030 MHz ระยะห่างของพัลส์(pulse) เป็นตัวแสดงโหมด(Mode) ของการสอบถามปัจจุบันในกิจการบินพลเรือนมี 4 โหมด คือ

Mode A: กระตุ้นทรานสปอนเดอร์ให้การแสดงตน(สัญญาณเรียกขาน/หมายเลขการจดทะเบียน

Mode B: สำหรับการขยายบริการในอนาคด

Mode C: กระตุ้นทรานสปอนเดอร์ให้แสดงความสูงของเพดานบิน

Mode D: สำหรับการขยายบริการในอนาคด

คำตอบของทรานสปอนเดอร์(Transponder) ออกมาในรูปแบบของพัลส์ เช่นกัน ที่ความถี่ 1,090 เมกกะเฮิร์ตซ์ (MHz) ประกอบด้วยรหัส 4096 ชนิด ซึ่งจะตอบคำถามตามโหมด ที่ถามมาโดยอัตโนมัติ นอกจากนั้นที่ทรานสปอนเดอร์ ยังมีปุ่มบังคับที่สำคัญอีก 3 ปุ่ม ให้นักบินกดในขณะที่อยู่ที่โหมดเอ คือ

- ปุ่มอีเมอร์เจนซี(Emergency) เมื่อกดปุ่มนี้ ทรานสปอนเดอร์ จะส่งรหัส 7700 ไปยังภาคพื้นดินแจ้งว่ามีเหตุฉุกเฉินเกิดขึ้นบนเครื่องบิน

- ปุ่มแจ้งระบบสื่อสารขัดข้องจะเป็นรหัส 7600

- ปุ่มแจ้งว่าเครื่องบินถูกไฮแจค(Hijacked) จะเป็นรหัส 7500

อนาคตของเรดาร์ทุติยภูมิ

เทคโนโลยีของเรดาร์ทุติยภูมิได้รับการพัฒนาขึ้นมาเรื่อยๆ อย่างไม่หยุดยั้ง ปัจจุบันมีโมโนพัลส์เทคนิค (Monopulse Technique) เพื่อปรับปรุงความแม่นยำในการจัดทิศทางของเป้า (Azimuth accuracy) และ สายอากาศอาร์กเวอร์ติคัลอะเพอร์เจอร์ (Large Vertical Aperture Antenna) เพื่อปรับปรุงรูบกรวยลำคลื่น เพื่อลดสัญญาณสะท้อนภาคพื้นดิน นอกจากนั้นในอนาคตจะมี Mode S เพิ่มขึ้นอีก

โหมดเอส Mode S

ปัญหาอันเกิดจากการติดตั้งสถานีเรดาร์ทุติยภูมิภาคพื้นดินกันอย่างมากมาย โดยปราศจากการควบคุมจะทำให้เกิดผลเสียดังนี้

1. สถานีภาคพื้นดินส่งคำถามไปยังเครื่องบินพร้อม ๆ กันมากเกินไป
2. เกิดสภาพกดข้าม (Suppression) แก่ Transponder มากเกินไป
3. เกิดภาพรบกวนชนิดอะซิงโครนัส(เนื่องจาก "Fruit")
4. เกิดภาพรบกวนชนิดอะซิงโครนัส(เนื่องจากภาพทับซ้อนและการตอบสนองแทรก)
5. บริเวณครอบคลุม (Coverage) ด้านตั้งไม่สมบูรณ์
6. มีข้อมูลจากเป้าปลอมที่เกิดจากการสะท้อนของคำถามและ/หรือคำตอบ
7. ความแม่นยำในการบอกทิศทางของเป้าลดลง และ
8. ไม่สามารถแยกแยะให้เห็นความแตกต่างระหว่างเครื่องบิน 2 ลำที่อยู่ใกล้กันได้ จึงมีการคิดค้นหาทางแก้ไข จึงเกิดโหมดเอสขึ้นซึ่งมีการปรับปรุงดังนี้

8.1 แทนที่สถานีภาคพื้นดินจะส่งคำถามกระจายทั่วไปในอากาศ ก็ส่งคำถามจำเพาะเจาะจงไปยังเครื่องบินแต่ละลำ (ในกรณีนี้เครื่องบินแต่ละลำต้องมีรหัสหรือชื่อเรียกประจำตัว)

8.2 เพิ่มจำนวนข้อมูลที่สามารถบรรจุลงในดาต้าลิงค์ (Data Link) ที่ใช้ได้ต่อระหว่างเครื่องบินกับสถานีภาคพื้นดิน และระหว่างเครื่องบินด้วยกันเองให้มากขึ้น ซึ่งวิธีนี้ก่อให้เกิดประโยชน์ 2 ประการคือ

- ใช้ในระบบป้องกันการชนกันในอากาศ (Air to Air Collision Avoidance System)
- ใช้เป็นระบบสื่อสารสำรอง ระหว่างนักบินกับเจ้าหน้าที่ควบคุมจราจรทางอากาศภาคพื้นดิน ซึ่งปัจจุบันใช้ระบบวิทยุ วีเอชเอฟ(VHF) เป็นระบบสื่อสารหลัก

2.4 การจราจรทางอากาศ

การขนส่งทางอากาศมีบทบาทสำคัญยิ่ง ในการพัฒนาทางเศรษฐกิจและสังคม เป็นช่วยการคมนาคมที่เชื่อมโยง และครอบคลุมทุกประเทศไว้ด้วยกัน สามารถที่จะทำการขนส่งคนโดยสาร สินค้าและพัสดุภัณฑ์ได้อย่างรวดเร็ว กิจการขนส่งทางอากาศได้เจริญรุดหน้าไปอย่างสม่ำเสมอตลอดมา

ปัจจัยที่สำคัญประการหนึ่ง ที่สามารถทำให้การขนส่งทางอากาศระหว่างประเทศ เป็นไปได้ด้วยความปลอดภัย มีประสิทธิภาพและเป็นไปได้ด้วยความเรียบร้อยคือ "บริการสื่อสารการบินหรือบริการวิทยุการบิน"

ในปัจจุบันนี้เครื่องบินที่ใช้ในการขนส่งทางอากาศ ส่วนใหญ่จะเป็นเครื่องบินไอพ่นที่มีความเร็วสูงมีขนาดใหญ่ สามารถจะบรรจุคนโดยสารได้คราวละมากๆ มีรัศมีทำการบินได้ไกล และใช้เพดานบินสูง

บริการสื่อสารจึงเป็นความจำเป็นที่จะขาดเสียไม่ได้ เนื่องจากการปฏิบัติงานการเดินอากาศที่ไม่เสี่ยงต่ออันตราย โดยมีเจ้าหน้าที่ควบคุมจราจรทางอากาศยาน ฝ้าติดตาม มองดูในจอเรดาร์อย่างใกล้ชิด หากมีสิ่งผิดปกติเกิดขึ้น เจ้าหน้าที่ควบคุมจะแจ้งให้นักบินทราบได้ สำหรับการให้คำแนะนำต่ออากาศยานที่กำลังบิน ออกจากสนามบินก็เช่นกัน เจ้าหน้าที่ควบคุมฯ จะให้คำแนะนำต่ออากาศยานด้วยการกำหนดทิศทางระยะสูง เพื่อที่จะให้อากาศยานนั้น ๆ สามารถบินเข้าสู่ทางบินตามความประสงค์ได้โดยรวดเร็ว โดยอาศัยเรดาร์เป็นอุปกรณ์ในการนำทางโดยตลอด เมื่ออากาศยานนั้นบินเข้าสู่เส้นทางบินเรียบร้อยแล้ว นักบินก็จะอาศัยเครื่องช่วยในการเดินอากาศที่มีอยู่ทำการบินออกไปได้โดยถูกต้องและปลอดภัย ทั้งนี้จะมีเจ้าหน้าที่ควบคุมจราจรทางอากาศเป็นผู้ให้คำแนะนำตลอดเส้นทางบินด้วยความปลอดภัย รวดเร็ว และมีระเบียบตรงกับจุดมุ่งหมายที่วางไว้ นอกจากนี้ยังมีการนำเอาวิทยุเครื่องช่วยการเดินอากาศเข้ามาช่วยในการทำงานร่วมกับระบบเรดาร์ เพื่อเพิ่มประสิทธิภาพงานควบคุม ดังจะกล่าวรายละเอียดในบทต่อไป

2.5 เจ้าหน้าที่ควบคุมจราจรทางอากาศ

ทั่วโลกยอมรับว่าอาชีพของเจ้าหน้าที่ควบคุมจราจรทางอากาศ เป็นอาชีพที่ต้องอาศัยความรู้ความชำนาญมากเป็นพิเศษ มีประสิทธิภาพการทำงานที่พอเพียง จึงอาจกล่าวได้ว่ามีความรับผิดชอบมากกว่านักบิน เนื่องจากนักบินรับผิดชอบเฉพาะอากาศยานที่ตนบินอยู่เท่านั้นแต่เจ้าหน้าที่ควบคุมจราจรทางอากาศต้องรับผิดชอบต่ออากาศยานทุกเครื่องที่กำลังบินอยู่ในเขตพื้นที่ที่รับผิดชอบ ความกลัวที่ว่าอากาศยานจะชนกันกลางอากาศ จะทำให้เกิดความเครียดกับเจ้าหน้าที่ควบคุมจราจรทางอากาศว่านักบิน จะเห็นได้ว่าทำอากาศยานบางแห่ง เจ้าหน้าที่ควบคุมฯ จะถูกกำหนดให้ทำงานในช่วงระยะเวลาสั้น ๆ เท่านั้น ผู้เชี่ยวชาญทางด้าน การควบคุมจราจรทางอากาศมักจะเตือนเจ้าหน้าที่ควบคุมฯ ว่าอย่าจ้องดูที่อากาศยาน และพะวงว่ามันบรรทุกชีวิตมาตั้ง 300 คน เพราะจะทำให้เครียดทันที แต่ควรมองดูว่ามันเป็นเพียงเศษตบุงชิ้นหนึ่งที่เคลื่อนที่ได้เท่านั้น

การปฏิบัติหน้าที่ของเจ้าหน้าที่ควบคุมฯ ด้าน แอพรอช คอนโทรล (Approach Control) นี้มีวัตถุประสงค์ เช่นเดียวกับเจ้าหน้าที่ควบคุมจราจรทางอากาศในด้านอื่น ๆ กล่าวคือ

1. ป้องกันการชนกันระหว่างอากาศยานกับอากาศยาน และ ระหว่าง คอนโทรลเซอร์วิส (Control Service) จะให้คำแนะนำ และควบคุมอากาศยานทุกเครื่องที่กำลังบินเข้าสู่สนามบินตามเส้นทางบินต่าง ๆ ด้วยการใช้เรดาร์นำทาง แนะนำอากาศยานให้บินเข้าหาเครื่องช่วยในการเดินอากาศ หลังจากนั้นจะแนะนำให้อากาศยาน ร่อนลงสู่สนามบิน ตามวิธีการร่อนลงสู่สนามบินที่ได้กำหนด และประกาศให้นักบินทราบทั่วกันแล้ว ซึ่งเครื่องช่วยในการเดินอากาศเหล่านี้สามารถที่จะให้ข้อมูลต่าง ๆ แก่นักบินได้ เช่น ให้ทิศทางบินเข้าสู่สนามบินตามแนวเส้นกึ่งกลางของทางวิ่ง ให้มุมร่อน (Clide Path) ที่ถูกต้องแน่นอนขณะที่กำลังร่อนลงสู่สนามบินด้วยความปลอดภัย

2. เพื่อให้การจราจรทางอากาศดำเนินไปอย่างรวดเร็วและมีระเบียบ

การที่จะให้การเดินอากาศของอากาศยาน เป็นไปได้ด้วยความปลอดภัย มีความรวดเร็วและเป็นระเบียบเจ้าหน้าที่ควบคุมจราจรทางอากาศต้องมีความรู้ความสามารถ และมีความเข้าใจในกฎเกณฑ์มาตรฐานขององค์การการบินพลเรือนโดยเฉพาะการใช้ภาษาไทยใช้ภาษาอังกฤษในการติดต่อระหว่างเจ้าหน้าที่ควบคุมฯ กับนักบิน วิธีการในการติดต่อจะทำตามข้อกำหนดที่มีในองค์การการบินพลเรือนระหว่างประเทศ (ICAO) อีกทั้งมีสิ่งอำนวยความสะดวกในการเดินอากาศเพื่อเพิ่มความปลอดภัยยิ่งขึ้น

สำหรับที่ทำอากาศยานฯ อุปกรณ์ที่ให้ประโยชน์กับนักบินและเจ้าหน้าที่ควบคุมฯ มาก คือเรดาร์ (RADAR) ที่สามารถให้ข้อมูลต่าง ๆ แก่เจ้าหน้าที่ควบคุมฯ เช่น

- เป้าหมายและชื่อเรียกขานของเครื่องบิน
- ตำบลและทิศทางที่อากาศยานเคลื่อนที่ไป
- ระยะสูงของอากาศยานที่กำลังบินอยู่

จากข้อมูลต่าง ๆ ในจอเรดาร์เจ้าหน้าที่ควบคุมฯ จะใช้ประกอบการควบคุม เช่นการจัดระยะระหว่างอากาศยาน การแนะนำ สิ่งอื่น ๆ ที่เกี่ยวข้องให้อากาศยานในเขตพื้นที่ควบคุมฯ ได้ทราบ เจ้าหน้าที่ควบคุมฯ ใน

ในด้านแอพพรอช คอนโทรล เซอร์วิส (Approach Control Service) จะให้คำแนะนำ และควบคุมอากาศยานทุกเครื่องที่กำลังบินเข้าสู่สนามบิน ตามเส้นทางบินต่าง ๆ โดยการให้เรดาร์นำทาง แนะนำอากาศยานให้บินเข้าหาเครื่องช่วยการเดินอากาศ หลังจากนั้นแนะนำให้ร่อนลงสู่สนามบินตามวิธีการที่ได้กำหนด และประกาศให้นักบินได้ทราบทั่วกัน เครื่องช่วยการเดินอากาศเหล่านี้สามารถให้ข้อมูลต่าง ๆ แก่นักบินได้ เช่นทิศทางบินเข้าสู่สนามบินตามแนวเส้นทางกึ่งกลางทางวิ่ง ให้มุมร่อนที่ถูกต้อง(Clide Path) ขณะที่กำลังร่อนลงสู่สนามบิน โดยมีเจ้าหน้าที่ควบคุมฯเฝ้าติดตามมองดูในจอเรดาร์อย่างใกล้ชิด และให้คำแนะนำตามความเหมาะสม

2.6 การควบคุมการจราจรทางอากาศที่ท่าอากาศยาน

(Air Traffic Approach Control)

ในบรรดาการขนส่งทั้งหมด ที่ผู้ใช้ได้รับประโยชน์ทั้งในด้านความเร็ว ความปลอดภัย และประหยัดทางเศรษฐกิจในปัจจุบันนี้ การขนส่งทางอากาศนับว่ามีความสำคัญเป็นที่หนึ่ง และมีความเจริญก้าวหน้าไปอย่างรวดเร็วมากเมื่อเทียบกับการขนส่งอื่น ๆ

องค์ประกอบสำคัญ ที่ช่วยให้การขนส่งทางอากาศเป็นไปด้วยความเร็ว ปลอดภัยและประหยัดนั้นย่อมประกอบด้วยปัจจัยหลายอย่าง เช่น อากาศยาน และอุปกรณ์ที่ติดตั้งบนอากาศยาน นักบิน เครื่องช่วยในการเดินอากาศ อุปกรณ์ที่ใช้ในการควบคุมจราจรทางอากาศสนามบิน และสิ่งอำนวยความสะดวกประจำสนามบิน รวมทั้งเจ้าหน้าที่ควบคุมจราจรทางอากาศเป็นต้น สำหรับประเทศไทยนั้น สนามบินดอนเมืองเป็นท่าอากาศยานสากล (International Airport) ซึ่งถือว่าเป็นศูนย์กลางการขนส่งทางอากาศของภาคเอเชียตะวันออกเฉียงใต้ เป็นจุดรวมของสายการบินพาณิชย์ต่างประเทศมากกว่า 35 บริษัทที่มาให้บริการของสนามบินแห่งนี้ตั้งจะเห็นได้จากแผนที่การเดินทางอากาศว่า สนามบินดอนเมืองมีเส้นทางของการบินต่าง ๆ มากมายต่อเชื่อมโยงไปยังสนามบินของประเทศเพื่อนบ้านใกล้เคียงและเชื่อมโยงไปทั่วโลก เส้นทางบิน (Air Way) นั้นนับว่ามีความสำคัญต่อการขนส่งทางอากาศมาก ทั้งนี้เนื่องจากตามเส้นทางบินที่ถูกกำหนดขึ้นนั้น มีเครื่องช่วยในการเดินอากาศชนิดต่าง ๆ ติดตั้งไว้ที่พื้นดิน ทำให้นักบินที่บินผ่านเส้นทางบินแต่ละเส้นทาง สามารถทราบตำบลที่อยู่ของตนเองได้ถูกต้องตลอดเวลา โดยอาศัยอุปกรณ์ที่ติดตั้งบนอากาศยานประกอบ

การสร้างเส้นทางบินขึ้นนี้ก็เช่นเดียวกับการสร้างถนนให้รถยนต์วิ่งสัญจรไปมา การจราจรบนท้องถนนนั้น หากไม่กำหนดระเบียบในการจราจรแล้ว อุบัติเหตุก็มักจะเกิดขึ้นได้เสมอ ดังจะเห็นได้ว่าในบริเวณที่มีทางแยก หรือตามวงเวียนต่าง ๆ มักจะมีอุบัติเหตุเป็นประจำ ทั้ง ๆ ที่มีตำรวจจราจรคอยให้สัญญาณกับผู้ใช้รถยนต์

ในทำนองเดียวกับท่าอากาศยาน ถ้าปล่อยให้บินไปอย่างไม่มีระเบียบ ก็ย่อมจะต้องมีอุบัติเหตุชุกกันเกิดขึ้นได้ ดังนั้น การกำหนดให้อากาศยาน บินอยู่ในเส้นทางบินตามทิศทางและระยะความสูงที่ต้องการ ก็จำเป็นต้องมีเจ้าหน้าที่จราจรทางอากาศคอยให้คำแนะนำอยู่ตลอดเวลาทั้งนี้โดยอาศัยกฎเกณฑ์ในการจราจรทางอากาศที่เป็นมาตรฐานเดียวกัน อุปกรณ์และเครื่องมืออำนวยความสะดวกในการบินขึ้น-ลง สิ่งต่าง ๆ เหล่านี้แตกต่างไปจากการจราจรทางบกมากมายที่เป็นเช่นนี้เพราะอากาศยานมีความเร็วสูงมากหากนักบิน ไม่ปฏิบัติตามคำแนะนำของเจ้าหน้าที่จราจรทางอากาศ หรือเจ้าหน้าที่จราจรทางอากาศมีขีดความสามารถในการให้คำแนะนำ อุบัติเหตุจะเกิดขึ้นได้ในชั่วพริบตาเดียว และผลที่ตามมาก็คือชีวิตมนุษย์และทรัพย์สินจำนวนมากจะสูญสิ้นไปด้วยทันที ดังที่เคยได้ทราบอุบัติเหตุการชนกันของอากาศยานรายใหญ่ ๆ อยู่เสมอ และจากผลของการสอบสวน มักจะปรากฏว่าความผิดพลาดที่เกิดขึ้นนั้นมักจะมีมาจากนักบินหรือเจ้าหน้าที่จราจรทางอากาศบกพร่อง

ประเทศไทยในฐานะที่เป็นสมาชิกขององค์การการบินพลเรือนระหว่างประเทศ จึงจำเป็นต้องถือปฏิบัติตามกฎเกณฑ์และมาตรฐานสากล รวมทั้งข้อเสนอแนะขององค์การฯที่กำหนดไว้ เพื่อให้เป็นมาตรฐานเดียวกัน ส่วนข้อแตกต่างในรายละเอียดปลีกย่อยอื่น ๆ นั้นหากมีกำหนดไว้จะต้องแจ้งให้ประเทศสมาชิกขององค์การฯได้ทราบโดยทั่วกัน สำหรับการบริการการควบคุมจราจรทางอากาศของประเทศไทยนั้น ได้แบ่งมอบ

ความรับผิดชอบให้กับหน่วยงานต่าง ๆ ดำเนินการต่าง ๆ ดังนี้

1. การบริการการควบคุมจราจรทางอากาศด้านแอเรียคอนโทรลเซอร์วิส (Area Control Service) บริษัทวิทยุการบินแห่งประเทศไทย จำกัด เป็นหน่วยงานในการบริการการควบคุมจราจรทางอากาศต่ออากาศยานทุกเครื่อง ที่ทำการบินด้วยเครื่องประกอบการบิน (Instrument Flight Rules) ภายในคอนโทรลแอเรีย (Control Area) ที่กำหนด รวมทั้งตามเส้นทางบินต่าง ๆ ภายในประเทศจนถึงเส้นเขตแดนของประเทศไทย และมีชื่อเรียกหน่วยงานนี้ว่า ศูนย์ควบคุมจราจรทางอากาศกรุงเทพฯ (Bangkok Area Control Centre)

2. การบริการการควบคุมจราจรทางอากาศด้านแอฟพรอชคอนโทรล (Approach Control Service) ณ ท่าอากาศยานกรุงเทพฯ การทำอากาศยานแห่งประเทศไทย เป็นหน่วยงานรับผิดชอบในงานบริการควบคุมจราจรทางอากาศต่ออากาศยานทุกเครื่อง ที่ทำการบินด้วยเครื่องวัดประกอบการบินภายในเขตที่ต่อจากคอนโทรล แอเรีย (Control Area) จนถึงสนามบิน โดยทั่วไปแล้วจะกำหนดเป็นลักษณะของคอนโทรลโซน (Control Zone) และมีชื่อเรียกหน่วยควบคุมนี้ว่า เบงคอกแอฟพรอชคอนโทรล (Bangkok Approach Control) สำหรับการบริการทางเดินนี้ ปัจจุบันใช้เรดาร์เป็นเครื่องมือประกอบในการควบคุมจราจรทางอากาศ ซึ่งเรียกการบริการนี้ว่า เรดาร์แอฟพรอชคอนโทรล (Radar Approach Control: RAPCON) หากเรดาร์ขัดข้องการบริการทางเดินนี้ก็ใช้เครื่องช่วยในการเดินทางอากาศ ที่ติดตั้งไว้บริเวณสนามบิน และการรายงานของนักบิน เป็นเครื่องช่วยประกอบในการควบคุม โดยเรียกบริการนี้ว่า คอนเวนชันนอลแอฟพรอชคอนโทรล (Conventional Approach Control) สำหรับสนามบินพาณิชย์แห่งอื่นนั้น กรมการบินพาณิชย์ กระทรวงคมนาคม เป็นหน่วยงานรับผิดชอบในการบริการการควบคุม ซึ่งการบริการก็ถือหลักเช่นเดียวกับสนามบินดอนเมือง แต่ส่วนมากจะใช้ระบบการควบคุมแบบคอนเวนชันนอลแอฟพรอชคอนโทรล

3. การบริการการควบคุมทางอากาศด้านแอโรโดรมคอนโทรลเซอร์วิส (Aerodrome Control Service) ณ ท่าอากาศยานกรุงเทพฯ การทำอากาศยานแห่งประเทศไทย เป็นหน่วยงานรับผิดชอบในการควบคุมเช่นเดียวกับ การบริการการควบคุมด้านแอฟพรอชคอนโทรลเซอร์วิส โดยให้บริการต่ออากาศยานทุกเครื่องที่ทำการบินด้วยทัศนวิสัย (Visual Flight Rules) ภายในคอนโทรลโซน (Control Zone) ตลอดจนอากาศยานทุกเครื่องที่ขับเคลื่อน อยู่บนพื้นที่ขับเคลื่อนของสนามบิน

ส่วนสนามบินพาณิชย์แห่งอื่นๆ กรมการบินพาณิชย์ เป็นหน่วยรับผิดชอบโดยตลอดเช่นเดียวกัน หน่วยที่บริการนี้มีชื่อว่า หอบังคับการบิน ตามด้วยชื่อของสนามบิน เช่น หอบังคับการบินดอนเมือง เป็นต้น

บทที่ 3

วิทยุเครื่องช่วยการเดินอากาศ

3.1 ระบบนำร่องเครื่องบิน

มนุษย์เราต่างก็มีระบบนำร่องด้วยกันทุกคน และใช้ประโยชน์จากสิ่งเหล่านี้อยู่ตลอดเวลาไม่ว่าจะเป็นการเดินทางไปยังที่ใด ๆ จะต้องเริ่มต้นจากการกำหนดเส้นทางจากจุดเริ่มต้นไปยังจุดหมายปลายทางเสมอ และคำนึงถึงสิ่งต่าง ๆ ซึ่งอาจส่งผลกระทบต่อการเดินทาง สิ่งเหล่านี้เป็นระบบนำร่องอัตโนมัติที่มีอยู่ประจำตัวของแต่ละคน สำหรับระบบนำร่องของเครื่องบินก็เช่นเดียวกันจำเป็นต้องมีการกำหนดจุดเริ่มต้นและจุดปลายทางเสมอและยังต้องศึกษาสภาพอากาศซึ่งเป็นปัจจัยสำคัญต่อการเดินทาง ในอดีตการนำร่องใช้การคำนวณทิศทางจากดวงดาวเป็นหลักหากเกิดมีทัศนวิสัยที่ไม่สะดวกต่อการกำหนดตำแหน่งดวงดาวเช่นมีเมฆมากหรือมีลมพายุพัดจัดนักบินก็ไม่สามารถนำเครื่องบินขึ้นได้ แต่จากความก้าวหน้าทางเทคโนโลยีทำให้มีระบบนำร่องขึ้นมาเพื่อช่วยให้นักบินสามารถเดินทางถึงจุดหมายปลายทางได้อย่างสะดวกและปลอดภัย แม้ทัศนวิสัยจะไม่เอื้ออำนวยก็ตามที และจนถึงปัจจุบันพัฒนาจนมีระบบแสดงตำแหน่งเครื่องบินอัตโนมัติ เพียงแต่ต้องทราบความถี่ ของสถานีที่ต้องการเพื่อปรับคลื่นให้ถูกต้อง แล้วอุปกรณ์จะทำหน้าที่บอกตำแหน่งโดยอัตโนมัติ โดยตำแหน่งนี้จะปรากฏบนแผงควบคุมและแผงบังคับการบินของนักบิน เรียกว่า ระบบค้นหาตำแหน่งอัตโนมัติ (Automatic Direction Finder : ADF) หรือระบบเข็มทิศการบินนั่นเอง ด้วยเหตุที่ระบบนี้สามารถใช้ร่วมกับการส่งคลื่นวิทยุมันจึงเพิ่มความคล่องตัวให้แก่เครื่องบิน ยิ่งถ้าหากสถานีภาคพื้นดินได้ทราบถึง จุดลงจอดหรือจุดปลายทางของเครื่องหรือเส้นทางการบิน ซึ่งอาจจะเรียกได้ว่า มันใช้ระบบชี้บอกตำแหน่งของเครื่องบินด้วยคลื่นวิทยุความถี่สูง หรือบีคอน(Beacon) อันเป็นการกระจายคลื่นวิทยุแบบไม่เจาะจงทิศทาง หรือระบบ เอ็นดีบี(NDB:Non Directional Beacons) จุดสำคัญของระบบนี้ก็คือการแปลงคลื่นวิทยุรหัสมอร์ส เพื่อให้สามารถกำหนดรหัสประจำตัวของเครื่องบินนั้นๆได้ง่ายยิ่งขึ้นความถี่ที่ใช้อยู่ มีค่าประมาณ 250-400 กิโลเฮิร์ตซ์ ยกเว้นการทำงานบางอย่าง ซึ่งอาจใช้ความถี่นอกเหนือจากที่กล่าว

สถานที่และภูมิประเทศนับเป็นปัจจัยสำคัญอีกประการต่อระบบ บางพื้นที่มีการรบกวนสูงต้องใช้การกระจายคลื่นที่มีความยาวมากต้องใช้สายอากาศขนาดใหญ่และสิ้นเปลืองพลังงานการส่งมากอีกด้วย และอาจต้องมี 2 เครื่องโดยอีกเครื่องเป็นเครื่องสำรอง เพื่อทำงานแทนในกรณีที่เครื่องหลักเกิดการขัดข้องและไม่สามารถทำงานต่อได้

สำหรับการทำงานของระบบนำร่องมีข้อเสียอยู่ที่ ในขณะที่มีลมแรง นักบินจะไม่สามารถบินในทิศทางที่กำหนด ไว้อย่างถูกต้องแม่นยำได้เลย ดังนั้นจึงอาจเกิดการบินไปยังเส้นทางที่ผิดพลาด และอาจเกิดอุบัติเหตุได้

โดยปกตินักบินจะต้องรับทราบ รายงานสภาพอากาศก่อนที่เครื่องจะขึ้นบิน เพื่อคาดคะเนล่วงหน้าถึงผลกระทบของลมทั้งจากทิศทางและความเร็ว เพื่อปรับเครื่องมือให้เหมาะสมกับสภาพอากาศต่างๆ ความยุ่งยากเหล่านี้ทำให้ต้องมีการปรับปรุงและพัฒนาจนกลายเป็นเครื่องชี้ตำแหน่งรอบทิศทาง โดยใช้ความถี่ย่านวีเอชเอฟ หรือเรียกกันทั่วไปว่าระบบ วีโออาร์ (VHF Omni Range Beacons:VOR)

วิทยุเครื่องช่วยการเดินอากาศเป็นการสื่อสารทางเดียว เป็นสถานีวิทยุที่ตั้งอยู่บนพื้นดิน กระจายข่าวออกอากาศเช่นเดียวกับวิทยุกระจายเสียง แต่คลื่นวิทยุที่ส่งออกไปนั้น ไม่ได้ประสงค์จะให้ผู้รับฟังเสียง แต่ด้วยเทคนิคการจัดระบบการกระจายคลื่น ทำให้ผู้ต้องการทราบข่าวหรือผู้ที่ต้องการใช้บริการ ได้แก่ นักบิน สามารถรับคลื่นวิทยุที่ส่งออกไปในอากาศด้วยเครื่องรับเฉพาะอย่าง แสดงทิศทางของผู้รับจากตำแหน่งของสถานีเทียบกับทิศเหนือได้มุมเป็นองศาที่ถูกต้อง ซึ่งสถานีนี้เรียกว่า สถานีวีโออาร์ (Very High Frequency Omni-Range)



ระบบวีโออาร์ (VOR)

ระบบนำร่องที่ใช้สายอากาศแบบหมุน เป็นที่รู้จักกันดี และได้รับการพัฒนามาใช้ตั้งแต่ปี 1907 โดยบริษัทของเยอรมันนี ชื่อเทลเฟุงเกิน (Telefunken) เปิดตัวระบบนำร่องแบบเข็มทิศ บรรดาวิศวกรทั้งหลายได้ค้นคว้าและพบว่า ถ้าหากสามารถสร้างสายอากาศที่มีส่วนวางในแนวระนาบยาวกว่าส่วนที่เป็นแนวตั้ง จะให้คุณสมบัติที่ดีกว่า

ในปี 1906 มาร์โคนีได้จดลิขสิทธิ์การออกแบบสายอากาศคร่าว ๆ ของเขา โดยมีลักษณะการจัดวางสายเป็นวงกลม สัญญาณที่ได้แรงที่สุดจะเป็นตัวกำหนดตำแหน่งของเครื่องบิน ปีต่อมาบริษัทเทลเฟุงเกินได้นำเสนอความคิดคล้าย ๆ กัน ออกมาตั้งรูปที่ ประกอบด้วยสายอากาศ 32 ชุด สัญญาณจะเริ่มต้นแพร่กระจายจากจุดศูนย์กลางของสายอากาศแบบรอบทิศทาง ไปรอบ ๆ ใช้เวลา 1 วินาทีต่อสายอากาศแต่ละชุดนักบินสามารถจะเปรียบเทียบการรับสัญญาณได้ดีที่สุดกับเข็มทิศเครื่องบินเพื่อกำหนดตำแหน่งได้

มาร์โคนี ได้พัฒนาเครื่องชี้บอกตำแหน่ง โดยใช้คลื่นวิทยุย่านวีเอชเอฟในการแพร่กระจาย และเริ่มนำระบบนี้มาใช้ในปี 1922 ระบบนี้ประกอบด้วยสายอากาศทรงพาราโบลา 2 ตัวที่ติดหันหลังให้กัน เคลื่อนที่ครบรอบทุก ๆ 2 นาที ความกว้างหน้าอันยิ่งใหญ่อันหนึ่งอยู่ที่อุปกรณ์ที่ใช้เป็นกุญแจสำคัญในการส่งสัญญาณผ่านจุดต่อของวงแหวนที่ติดตั้งในฐานของสายอากาศ ข้อความจะถูกส่งกระจายคลื่น ออกในรูปของรหัสมอร์ส และเข้าสู่สายอากาศของเครื่องบิน นักบินเพียงแต่สังเกตข้อมูลที่ได้รับเข้ามา ในทิศทางที่ตั้งชัดเจนที่สุดเพื่อให้ทราบตำแหน่งของตนจากเครื่องชี้บอกตำแหน่ง

ในช่วงปี 1920 ถึงต้นปี 1930 ประเทศสหรัฐฯ ได้พยายามพัฒนาระบบการกระจายคลื่น ให้สามารถสื่อสารข้ามประเทศได้ แต่ไม่ได้รับความนิยมมากนัก เนื่องจากเครื่องชี้บอกตำแหน่งนี้สามารถระบุเส้นทางได้เพียง 4 สาย ต่อ 1 เครื่อง และจะทำงานได้ต่อเมื่อเที่ยวบินนั้น ๆ ระบุเส้นทางบินที่แน่นอนไว้ล่วงหน้า

ต่อมาปี 1937 จึงเริ่มมีการตกลงในการพัฒนาเครื่องชี้บอกตำแหน่งให้ทันสมัยขึ้นแต่ก่อนที่จะเปิดให้บริการได้เกิดสงครามโลกขึ้นเสียก่อนจึงเลื่อนการใช้บริการมาเป็นในปี 1950 ในหลักการควบคุมจราจรทางอากาศ การทราบทิศทางของเครื่องบินมีความสำคัญมาก เพื่อที่นักบินจะได้บินไปในทิศทางที่ถูกต้อง และหากจะให้เกิดความแน่นอนจะต้องทราบด้วยว่า เครื่องบินอยู่ที่ระยะห่างจากจุดสถานีเท่าใดด้วย ซึ่งเป็นการกำหนดจุดของเครื่องบินแต่ละขณะได้ จึงได้มีวิทยุเครื่องช่วยการเดินอากาศอีกประเภทหนึ่งที่บอกระยะทางเรียกว่า ดีเอ็มอี (DME: Distance Measuring Equipment)

เมื่อนักบินมีแผนที่อยู่ในมือ และรู้ความสูงของตนเองโดยอัลติมิเตอร์ (Altimeter) อีกทั้งรู้ทิศทางและระยะห่างจากจุดที่ตั้งสถานี ซึ่งกำหนดไว้ในแผนที่ นักบินก็จะสามารถกำหนดจุดของตนเองในอากาศได้แน่นอนตลอดเวลา เมื่อติดต่อทางวิทยุกับเจ้าหน้าที่จราจรทางอากาศภาคพื้นดิน แล้วรายงานตำแหน่งของตนให้พนักงานควบคุมจราจรทางอากาศทราบ พร้อมกันนั้นหากใช้เรดาร์ ในการควบคุมจราจรทางอากาศด้วยแล้ว ก็จะเป็นการชี้ให้เห็นแน่นอนว่า นักบินอยู่ในตำแหน่งที่ถูกต้องตามที่รายงานนั้น ซึ่งก็จะเห็นได้แล้วว่าเรดาร์มีความสำคัญในจราจรทางอากาศมาก เป็นการเพิ่มความปลอดภัยในการบินตามเส้นทางบินได้เป็นอย่างมาก

ในการร่อนลงเมื่อเครื่องบิน บินมาใกล้สนามบินแล้ว ก็ยังต้องการวิทยุเครื่องช่วยการเดินอากาศที่แน่นอนยิ่งขึ้นไปอีก ทั้งนี้เพราะหากจะเปรียบเทียบความกว้างยาวทางวิ่งบนสนามบินกับระยะที่บินไปกว่าจะถึงสนามบินแต่ละครั้งนั้นก็จะเป็นอัตราส่วน 1:1000 ดังนั้นจึงไม่ใช่ของง่ายที่นักบินจะหาสนามบิน ที่ต้องการได้พบที่ระยะความสูง 8,000-40,000 ฟุต และบินลงในทิศทางที่ถูกต้อง หรือจะทำมร่อนที่พอดี และโดยเฉพาะอย่างยิ่งในเวลากลางคืนหรือในเวลาที่สภาพอากาศไม่ดี อีกทั้งเครื่องบินในปัจจุบันมีความเร็วสูง หากจะคิดว่าเป็นการเสี่ยง ก็มีอัตราเสี่ยงอยู่สูงมาก หากจะเปรียบก็ใกล้เคียงกับคำว่า 'งมเข็มในมหาสมุทร' หรืออาจจะยิ่งกว่า เพราะการบินภายใต้สภาวะการณ์ที่นักบินหาสนามบินไม่พบนั้น หมายถึงการสูญเสียอันยิ่งใหญ่

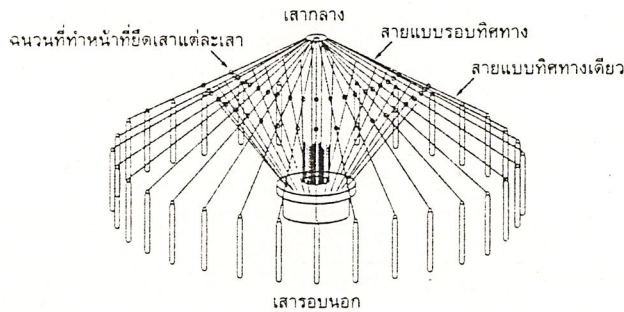
ในการควบคุมจราจรทางอากาศในบริเวณใกล้สนามบินนั้น จึงต้องมีอุปกรณ์วิทยุเครื่องช่วยการเดินอากาศอีกชนิดหนึ่งเรียกว่า ไอแอลเอส (ILS: Instrument Landing System) ช่วยบอกมร่อนที่ถูกต้อง

แนวทึ่งกลางทางวิ่งที่แน่นอน เพื่อให้ให้นักบินร่อนลงสู่สนามบินได้อย่างปลอดภัย อย่างไรก็ตามเครื่องมือนี้เป็นเพียงเครื่องช่วยในการร่อนลงเท่านั้น เมื่อใกล้จะถึงพื้นทางวิ่งจริงๆแล้ว นักบินก็สามารถมองเห็นทางวิ่งด้วยตาและใช้ความสามารถในการบังคับเครื่องบินให้แตะพื้นด้วยความรู้สึกของตนเอง อุปกรณ์วิทยุที่จะช่วยให้นักบินลงโดยอัตโนมัติจนถึงพื้นทางวิ่งนั้น มีผู้คิดประดิษฐ์อยู่ แต่ยังไม่เป็นที่ยอมรับให้นำมาใช้

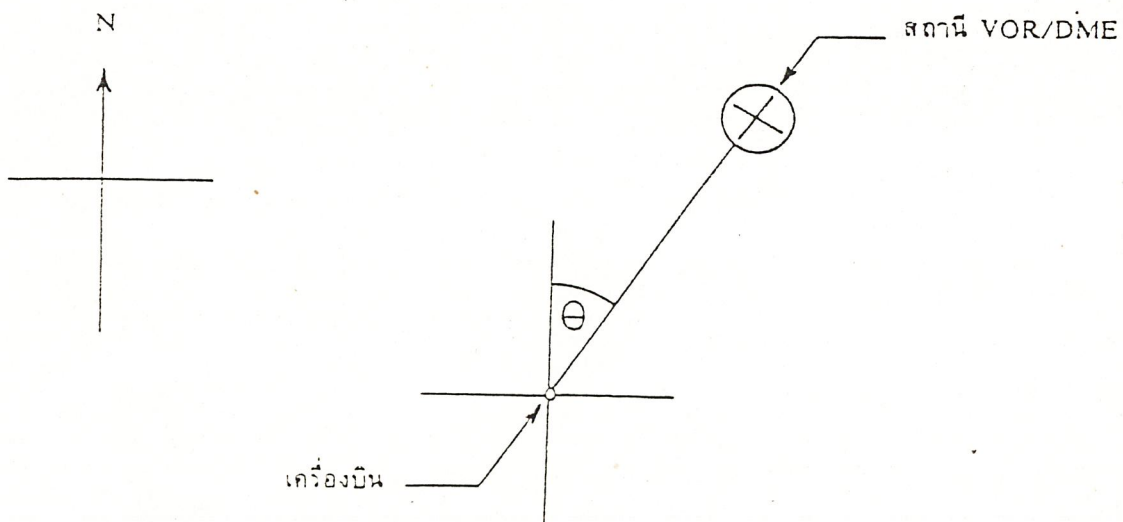
ที่กล่าวมานี้เป็นการสื่อสารทางวิทยุอีกประเภทหนึ่งที่ ส่งและรับสัญญาณทางเดียวไม่มีการตอบ และเป็นการส่งแบบนำสัญญาณไปผสมผสานกัน ให้เกิดเป็นประโยชน์ในการบอกทิศทาง มิได้ใช้เสียงพูดเป็นสื่อความหมาย ซึ่งนักบินทุกชาติทุกภาษา สามารถรู้ได้โดยง่ายจากเข็มบนหน้าปัทม์เท่านั้น

หลักการระบบของระบบวีไออาร์อยู่ที่การมอดูเลตความถี่ 30 เฮิร์ตซ์(Hz) ออกมา 2 ชุดเรียกว่า ความถี่อ้างอิงและความถี่แปรเปลี่ยน โดยความแตกต่างระหว่างความถี่ทั้งสองจะเป็นตัวแสดงตำแหน่งของสถานีรับสัญญาณ โดยอุปกรณ์เหล่านี้ทำงานอยู่ที่ย่านความถี่ 112.0-117.9 เมกะเฮิร์ตซ์ และใช้กำลังส่ง 220 วัตต์ ความถี่แปรเปลี่ยน 30 เฮิร์ตซ์ ความถี่อ้างอิง 30 เฮิร์ตซ์ ซึ่งจะมอดูเลตกับคลื่นพาห์ 9,960 เฮิร์ตซ์ เป็นหลักการแพร่สัญญาณของระบบนำร่องนั่นเอง

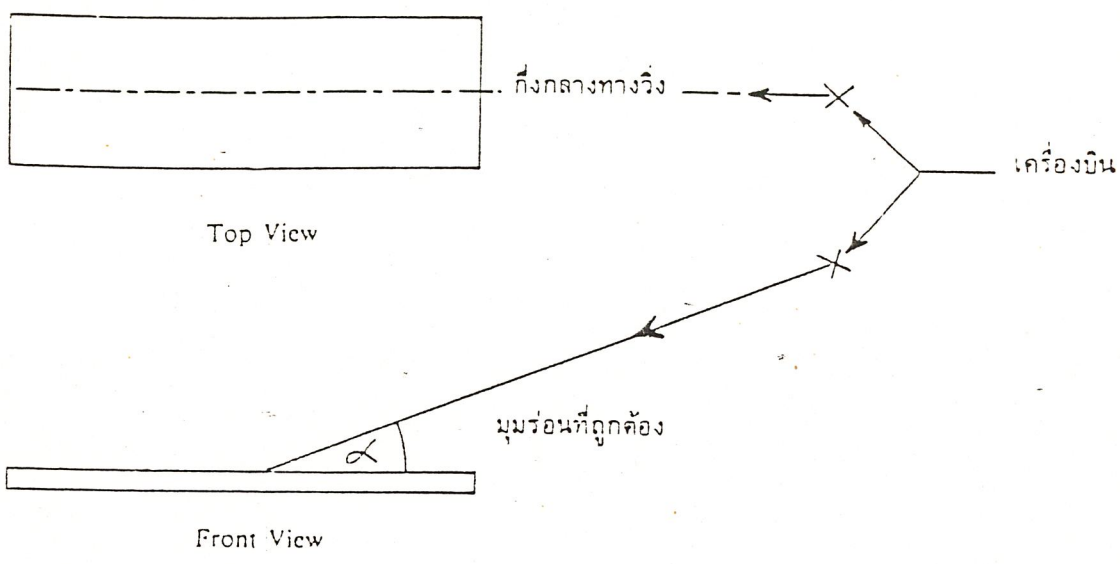
รูปที่ 1 แสดงสายอากาศแบบรอบทิศทางซึ่งผลิตในปี 1907



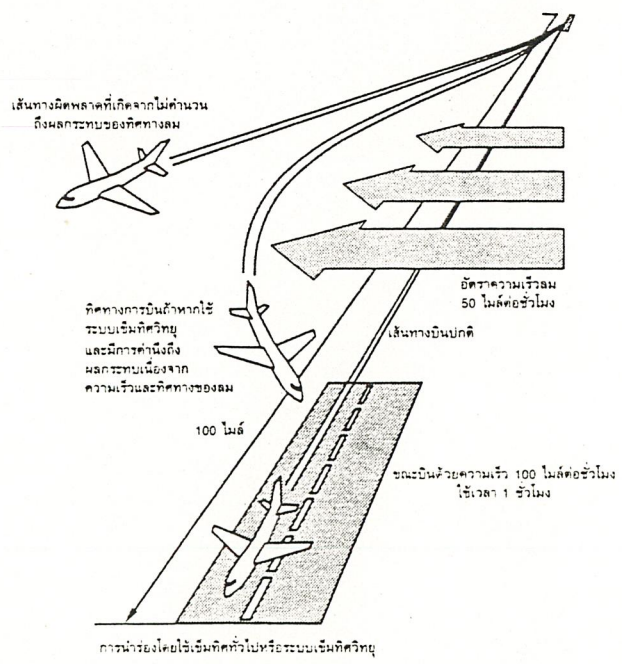
รูปที่ 2 แสดงทิศทางของผู้รับจากตำแหน่งของสถานีเทียบกับทิศเหนือได้มุมที่ถูกต้อง



รูปที่ 3 แสดงมุมร่อนและแนวทิ่งกลางทางวิ่งที่ถูกตัด



รูปที่ 4 แสดงการเปรียบเทียบเส้นทางการบินโดยแสดงผลกระทบจากสภาพอากาศ



3.2 อุปกรณ์ภายในเครื่องบิน

อุปกรณ์ VOR ในเครื่องบิน ใช้หลักการพื้นฐานของเครื่องรับชนิดซูเปอร์เฮเทอโรไดน์ ทำงานที่ความถี่ระหว่าง 112.0-117.9 เมกะเฮิร์ตซ์ ซึ่งบ่อยครั้งที่นำอุปกรณ์ สำหรับระบบลงจอดของเครื่องบินจะใช้ความถี่เท่ากับ 108 เมกะเฮิร์ตซ์

เครื่องรับจะทำการตีความเลตสัญญาณที่ระบบเข้ามาแล้วจ่ายให้กับ วงจรกรองความถี่ 2 ชุด ชุดแรกกรองความถี่ 30 เฮิร์ตซ์ และชุดที่สองเท่ากับ 9,960 เฮิร์ตซ์ โดยเอาต์พุตจากวงจรกรองความถี่ 9,960 เฮิร์ตซ์ จะถูกขยาย และส่งต่อไปยังวงจรแยกความถี่ซึ่งจะแยกเฉพาะสัญญาณอ้างอิง 30 เฮิร์ตซ์อีกส่วน โดยผลที่ได้ จะนำไปแสดงบนจอภาพภายในห้องบังคับการบิน

อุปกรณ์แสดงผลบนเครื่องบิน ประกอบด้วยโครงสร้าง 2 ส่วน คือเครื่องมือแสดงทิศทางที่เครื่องบินมุ่งหน้าไป และส่วนแสดงทิศทางการเคลื่อนที่ซ้ายขวาของเครื่องบินสามารถควบคุมทิศทางการบินได้ดีขึ้น

การใช้ระบบ VOR เพื่อช่วยการบิน เป็นการกำหนดตำแหน่งจากเครื่องชี้บอกตำแหน่งหลาย ๆ เครื่องเปรียบเทียบกันแต่จะง่ายกว่า ถ้าหากสามารถกำหนดตำแหน่งของเครื่องได้จากจุดใดจุดหนึ่งเท่านั้น และการจะทำเช่นนั้นได้ ก็ได้จากอุปกรณ์ ดีเอ็มอี(DME:Distance Measuring Equipment) ร่วมกับเครื่องชี้บอกตำแหน่งแบบ วิโออาร์

อุปกรณ์ดีเอ็มอี เป็นเสมือนเรดาร์ส่วนที่สองของอากาศยานทั้งหลาย ซึ่งแสดงระยะระหว่างเครื่องกับสถานีภาคพื้นดิน จากการคำนวณเวลาในการเดินทางไป-กลับของสัญญาณ แล้วนำมาแสดงผลบนจอภาพในห้องบังคับการบินของนักบิน

ระบบดีเอ็มอีระยะแรก ๆ ถูกพัฒนาขึ้นมาเพื่อใช้กับงานทางทหารในช่วงสงครามโดยทำงานที่ความถี่ 200 เมกะเฮิร์ตซ์ ออสเตรเลียได้นำระบบนี้มาปรับปรุงต่อและยังใช้อยู่ภายในประเทศมาจนถึงทุกวันนี้ แต่ไม่เป็นที่นิยมสำหรับประเทศอื่น ๆ ในปี 1959 ได้มีการกำหนดมาตรฐานสำหรับระบบดีเอ็มอีให้ทำงานที่ความถี่ 1000 เมกะเฮิร์ตซ์ โดยองค์การการบินพลเรือนระหว่างประเทศ (International Civil Aviation Organization) และในปัจจุบันก็ยังคงใช้ระบบนี้อยู่

การส่งสัญญาณสอบถามจากเครื่องบินไปยังสถานีภาคพื้นดิน จะเริ่มต้นขึ้นเมื่อมีการส่งสัญญาณคู่พัลส์ขนาด 3.5 ไมโครวินาทีต่อพัลส์ ระยะห่างระหว่างพัลส์ชุดหนึ่ง ๆ เท่ากับ 12 หรือ 36 ไมโครวินาที โดยมีอัตราการส่งประมาณ 150 คู่พัลส์ต่อวินาที เมื่อได้รับสัญญาณ สถานีภาคพื้นดินจะทำการหน่วงเวลาประมาณ 50 ไมโครวินาที แล้วจึงส่งสัญญาณโต้ตอบออกไป อุปกรณ์บนเครื่องบินจะรับสัญญาณ แล้วทำการคำนวณหาระยะห่างระหว่างอากาศยานและเครื่องชี้บอกตำแหน่งภาคพื้นดิน

สิ่งสำคัญประการหนึ่งสำหรับเครื่องส่ง DME ก็คือจะต้องทำให้เครื่องสามารถรับได้แต่เฉพาะสัญญาณโต้ตอบกลับของตนเอง โดยไม่ไปสับสนกับสัญญาณของเครื่องอื่น ซึ่งอาจเกิดผิดพลาดในการคำนวณได้ ดังนั้นจึงต้องออกแบบให้สัญญาณสอบถามมีอัตราการส่งแตกต่างกันเฉพาะตัว และเครื่องรับจะต้องสามารถรับได้เฉพาะสัญญาณโต้ตอบของตนเองเท่านั้น

3.3 ระบบดีเอ็มอีของสถานีภาคพื้นดิน

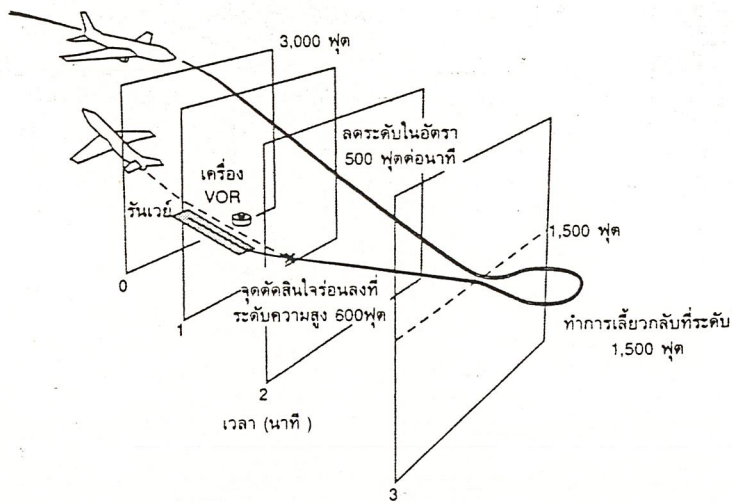
หลักการพื้นฐานของระบบดีเอ็มอีในภาคพื้นดินประกอบด้วย ภาครับสัญญาณ, วงจรตรวจสอบความถูกต้องของสัญญาณสอบถาม, วงจรหน่วงเวลา และภาคส่งสัญญาณโต้ตอบ เพื่อลดความรับผิดชอบจากสัญญาณแทรกแซงจะต้องจำกัดระดับการส่งสัญญาณโต้ตอบให้พอเหมาะ ในที่นี้ประมาณ 3000 คู่พัลส์ต่อวินาที และ เครื่องรับจะถูกออกแบบมาเพื่อป้องกันการโอเวอร์โหลดของสัญญาณสอบถามเครื่องจะไม่ตอบสนองต่อสัญญาณ ถ้าหากสัญญาณป้อนเข้ามาแรงเกินไปด้วยเหตุนี้เครื่องส่งจึงมีอัตราการตอบสนองคงที่ และทำให้เครื่องชี้บอกตำแหน่งทำงานได้เร็วขึ้น

ในซีกโลกตะวันตกที่เต็มไปด้วยอากาศยานจำนวนมาก การนำระบบวิโออาร์เข้ามารวมกับระบบดีเอ็มอีจึงเป็นการอำนวยความสะดวก และเพิ่มประสิทธิภาพให้แก่การจัดการจราจรเครื่องบินต่าง ๆ

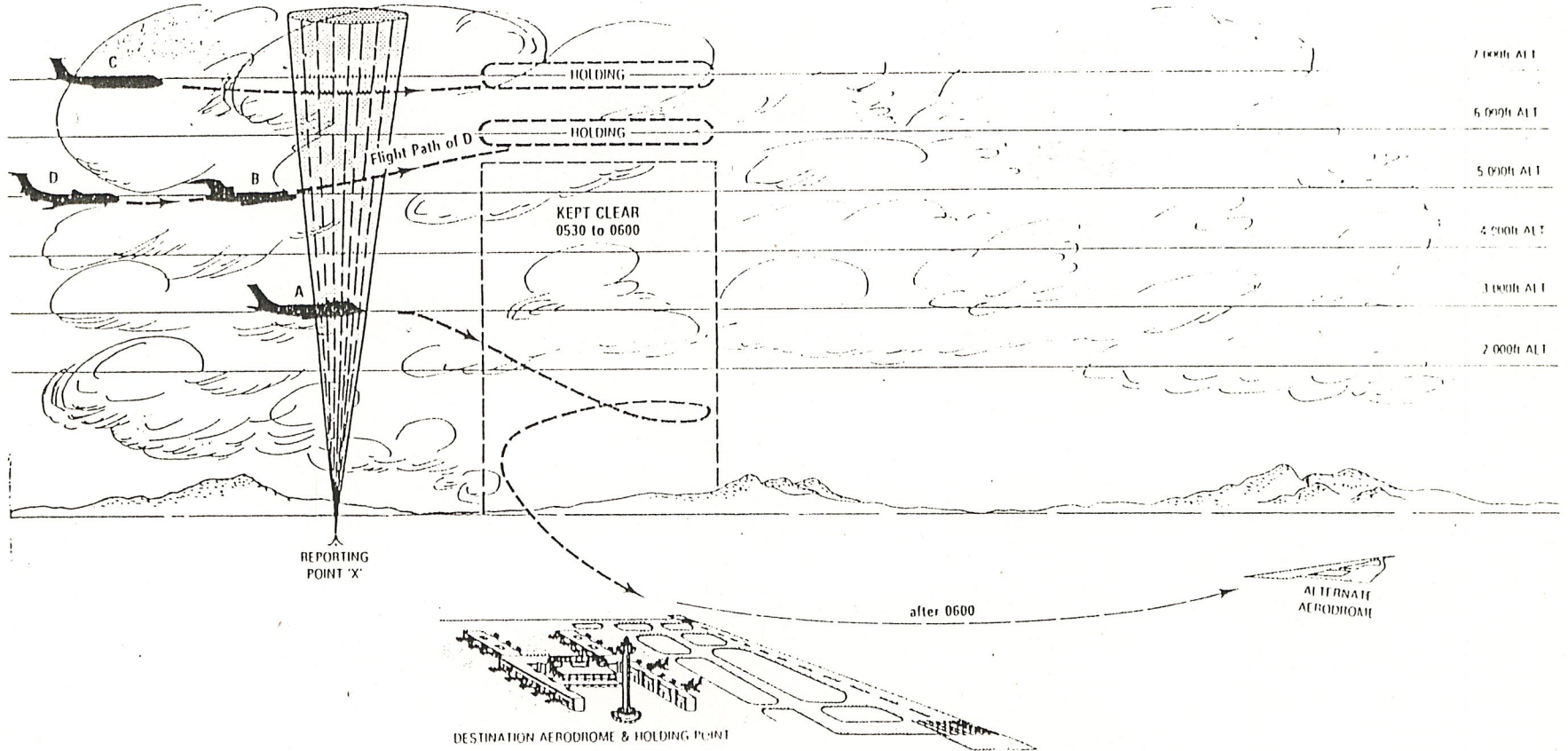
สำหรับในบริเวณที่มีการจราจรทางอากาศด่ำระบบเอ็นดีบี(NDB) และระบบเอดีเอส(ADS) ก็สามารถรองรับหน้าที่เหล่านี้ได้อย่างเพียงพอ และไม่สิ้นเปลืองงบประมาณมากนัก หากแต่ความก้าวหน้าทางด้าน

งด้านเทคโนโลยีในปัจจุบันกำลังเพิ่มขึ้นอย่างรวดเร็ว และ ระบบที่ถูกคิดขึ้นมาใหม่ๆ ที่ให้ประสิทธิภาพสูงๆ เช่น ระบบค้นหาตำแหน่งโดยใช้ ดาวเทียม หรือระบบจีพีเอส(GPS:Global Position System)จะช่วยเพิ่มขอบเขตการส่งสัญญาณได้มากยิ่งขึ้น สามารถรองรับการขยายตัวของ ปริมาณการจราจร ทางอากาศ ได้เป็นอย่างดี ซึ่งกำลังเข้ามาแทนที่ระบบนำร่องใน ปัจจุบันที่ละน้อย และก่อนที่จะก้าวสู่ศตวรรษที่ 21 เราคงจะได้เห็นความก้าวหน้าของเทคโนโลยีที่จะหันเหระบบนำร่องเครื่องบินไปสู่โฉมหน้าใหม่ ๆ ก็เป็นไปได้...

รูปที่ 5 แสดงการใช้ไออาร์ในรันเวย์ที่เครื่องบินลงจอด



รูปที่ 6 แสดงถึงวิธีการควบคุมการลงจอดเครื่องบินในสนามบิน



บทที่ 4

การเขียนโปรแกรมกราฟิกส์

สำหรับในการศึกษาระบบกราฟิกส์นั้น มีสิ่งที่ควรรู้ก่อนการเขียนโปรแกรมกราฟิกส์ดังต่อไปนี้

1 การทำงานของจอภาพ

สำหรับเครื่องพีซีไมโครคอมพิวเตอร์ จอภาพสามารถทำงานได้ 2 โหมด คือโหมดตัวอักษรและโหมดกราฟิกส์ ในลักษณะการทำงานปกติ จอภาพทำงานในโหมดตัว

อักษร สามารถแสดงข้อความได้ 25 บรรทัด บรรทัดละ 80 ตัวอักษร

ส่วนโหมดกราฟิกส์ นั้นแตกต่างจากโหมดตัวอักษรคือเป็นการแสดงผลโดยใช้จุดเล็กๆ หรือ Pixel ประกอบกันเป็นภาพบนจอภาพ ทำให้เราสามารถสร้างภาพต่างๆได้ซึ่งต่างกับโหมดตัวอักษร ที่แสดงผลได้เฉพาะตัวอักษรเท่านั้นการทำงานในโหมด

กราฟิกส์นี้ ยังแบ่งออกไป เป็นโหมดต่างๆได้อีก แต่ละโหมดจะ มีความละเอียด (Resolution) ในการแสดงผลและ จำนวนสีที่แสดงผลได้แตกต่างกันขึ้นกับชนิดของจอภาพ เช่นจอที่ใช้ในโครงการเป็นจอ VGAMed ความละเอียด 640x350 มี 16 สี และจำนวนเพจ 2 เพจ

การที่จะเปลี่ยนการทำงานของจอภาพจาก โหมดตัวอักษรไปเป็นโหมดกราฟิกส์ต้องมีการเขียนโปรแกรมควบคุมจอภาพจอภาพเราสามารถใส่โปรแกรมเปลี่ยนการทำงานจาก โหมดตัวอักษร ไปเป็นโหมดกราฟิกส์หรือจากโหมดกราฟิกส์เป็นโหมดตัวอักษรก็ได้และยังสามารถเลือกโหมดทางกราฟิกส์ที่ต้องการได้ด้วยหลังจากที่จอภาพทำงานในโหมดกราฟิกส์แล้ว เราจึงจะสามารถใช้คำสั่งต่างๆทางด้านกราฟิกส์ของเทอร์โบปาสคาลได้

2 เพจ

สำหรับจอภาพบางชนิดมีเพจ(page) ที่ใช้แสดงภาพได้หลายเพจ เพจหนึ่งๆ หมายถึงจอภาพ 1 จอ ถ้ามีหลายเพจก็หมายความว่า ในการทำงานจากโหมดกราฟิกส์ของจอภาพประเภทนั้น มีจอภาพประเภทนั้น มีจอภาพหลายจอให้เราเลือกดู เช่นจอโมโนโครมมี 2 เพจ ถ้าเราเลือกที่จะดูภาพในเพจแรก การวาดรูปต่าง ๆ ลงไปในเพจแรกก็จะปรากฏขึ้นในจอภาพ แต่ถ้าเราวาดรูปลงบนเพจที่ 2 รูปนั้นจะไม่ถูกแสดงออกทางจอภาพแต่จะถูกเก็บไว้ก่อน เมื่อใดที่เราเลือกดูภาพในเพจที่ 2 ภาพในเพจที่ 2 ถึงปรากฏออกทางจอภาพ จำนวนเพจของจอภาพขึ้นอยู่กับฮาร์ดแวร์ของจอภาพ และจำนวนหน่วยความจำที่ต้องการใช้ ยิ่งมีจำนวนเพจมากยิ่งต้องการหน่วยความจำมาก สำหรับเทอร์โบปาสคาลมีคำสั่ง SetVisual Page และ SetActivePage สำหรับเลือกเพจที่จะดูและเพจที่ใช้วาดภาพตามลำดับ

3 โคออร์ดิเนต

โคออร์ดิเนตของจอภาพในกราฟิกส์โหมด จะมีลักษณะต่างจากระบบโคออร์ดิเนตปกติ กล่าวคือ มุมบนซ้ายของจอภาพคือจุดกำเนิด(0,0) ค่าของโคออร์ดิเนตทางแกน x เพิ่มขึ้นจากซ้ายไปขวา ค่าโคออร์ดิเนตทางแกน y เพิ่มขึ้นจากบนลงล่าง เช่น จอภาพ VGAMed มีความละเอียด 640x350 โคออร์ดิเนตของจอภาพจะมีลักษณะตามที่แสดงไว้ในรูป

4 อัตราส่วนทางความกว้างต่อความสูงของจอภาพ (Aspect Ratio)

จอภาพทั่ว ๆ ไปจะมีความกว้างยาวของหน้าจอไม่เท่ากัน และนอกจากนี้ยังมีจำนวนพิกเซลในแนวนอนและแนวตั้งไม่เท่ากันด้วย อัตราส่วนความยาวในแนวนอนต่อความยาวในแนวตั้ง ของจอภาพนี้เรียกว่า Aspect Ratio เช่นจอโทรทัศน์มีค่า Aspect ratio คือ 4:3 หมายความว่า อัตราส่วนทางความกว้างต่อความสูงเป็น 4:3 ถ้าค่า Aspect Ratio ของจอภาพไม่เท่ากับ 1:1 เปรียบเหมือนกับว่าพิกเซลมีรูปร่างเป็นสี่เหลี่ยมผืนผ้า ทำให้การวาดรูปต่าง ๆ อาจบิดเบือนไปในสายตาของผู้มอง เช่นถ้าลากเส้นตรงที่มีความยาว 10 จุดตามแนวตั้งและแนวนอน เราจะเห็นเส้นตรง 2 เส้นนี้มีความยาวไม่เท่ากันบนจอภาพ ถ้าวาดรูปสี่เหลี่ยมจัตุรัส สี่เหลี่ยมนั้นก็จะมีมองเห็นเป็นสี่เหลี่ยมผืนผ้า แต่ถ้าค่า Aspect Ratio เป็น 1:1 เส้นตรงสองเส้นที่กล่าวมาก็มีความยาวเท่ากัน รูปสี่เหลี่ยมจัตุรัสก็จะดูแล้วมีความยาวเท่ากันทั้ง 4 ด้าน ดังนั้นถ้ารู้ค่า Aspect Ratio ของจอภาพ เราก็สามารถนำค่านี้มาปรับแต่งการวาดรูป เพื่อให้ภาพที่เกิดขึ้นบนจอภาพดูแล้วไม่บิดเบือนไป จากที่กล่าวข้างต้นจะเห็นได้ว่า

Aspect Ratio (ของจอที่ใช้ในโครงการ) มีค่าดังนี้

$$\begin{aligned} \text{Aspect Ratio} &= \text{จำนวนพิกเซลในแนวนอน} / \text{จำนวนพิกเซลในแนวตั้ง} \\ &= 640 / 350 \end{aligned}$$

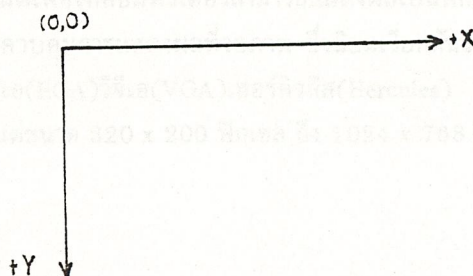
5 Current Pointer : CP

เทอร์โบปาสคาลได้กำหนด CP ไว้ให้จอภาพในโหมดกราฟิกส์ให้ด้วย ซึ่ง CP นี้จะถูกใช้สำหรับการทำงานของคำสั่งบางคำสั่งเช่น LineTo LineRel และ OutText เป็นต้น นอกจากนี้ยังมีคำสั่งสำหรับเปลี่ยนตำแหน่งของ CP อีกด้วยคือ MoveTo และ MoveRel

6 มุม

คำสั่งบางคำสั่งใน Graph Unit เช่น Arc Ellipse Sector เป็นต้น มีพารามิเตอร์สำหรับกำหนดค่าของมุม ซึ่งใช้สำหรับการทำงานของคำสั่งนั้น ค่าของมุมที่ส่งให้กับพารามิเตอร์นี้ จะถือว่ามีหน่วยเป็นองศา และลักษณะการกำหนดขนาดของมุมที่ใช้มีรูปแบบเหมือนกันหมดดังนี้ 3 นาฬิกาคือมุม 0 องศา 12 นาฬิกาคือมุม 90 องศา 9 นาฬิกาคือมุม 180 องศา 6 นาฬิกาคือมุม 270 องศา มุมในทิศทวนเข็มนาฬิกาเป็นบวก และทิศตามเข็มนาฬิกาเป็นลบ

รูปที่ 7 แสดงระบบโคออร์ดิเนตของจอภาพ VGAMed



ภาพแบบบิตแมพและภาพแบบเวกเตอร์

ภาพที่สร้างด้วยคอมพิวเตอร์ มีวิธีการสร้างสองแบบคือ แบบบิตแมพ(bit mapped)และแบบเวกเตอร์(vector)หรือ สโตรก(stroked) แต่ละแบบมีวิธีการสร้างภาพดังต่อไปนี้

1. ภาพแบบบิตแมพ สร้างขึ้นโดยการจัดให้มีพิกเซลอยู่ในตำแหน่งต่าง ๆ จนสามารถมองเห็นเป็นภาพที่ต้องการ จะเห็นได้ว่าภาพแบบบิตแมพ มีวิธีการสร้างคล้ายกับการเขียนจุดลงบนกระดาษกราฟ ซึ่งถ้าเขียนจุดที่ตำแหน่งต่าง ๆ ได้เหมาะสมเราจะมองเห็นเป็นภาพ

2. ภาพแบบเวกเตอร์ สร้างขึ้นโดยการใช้คำสั่งลากเส้นตรงเชื่อมต่อจุดต่าง ๆ ในทิศทางและตำแหน่งที่กำหนดเอาไว้ เช่น การสร้างรูปสามเหลี่ยมจะมีคำสั่งดังนี้

เลื่อนไปที่จุด (0,0)

ลากเส้นไปที่จุด (300,0)

ลากเส้นไปที่จุด (150,200)

ลากเส้นไปที่จุด (0,0)

เปรียบเทียบคุณสมบัติของภาพบิตแมพและภาพเวกเตอร์ในด้านความเร็วของการแสดงภาพที่จอภาพและความสามารถในการเปลี่ยนขนาดภาพจะได้ผลดังนี้

1. ภาพแบบบิตแมพ สามารถแสดงให้เห็นที่จอภาพได้เร็วกว่าภาพแบบเวกเตอร์ เช่น การแสดงภาพแบบบิตแมพขนาด 1000ไบต์จะทำให้การใช้คำสั่งย้ายข้อมูลขนาด 1000ไบต์จากหน่วยความจำที่เก็บภาพไปยังหน่วยความจำของจอภาพ (video display buffer) ภาพนั้นก็จะเป็นปรากฏบนจอภาพทันที การแสดงภาพแบบเวกเตอร์คอมพิวเตอร์จะใช้เวลามากกว่า เนื่องจากคอมพิวเตอร์ต้องทำตามคำสั่งที่มีจำนวนมากกว่า

2. การเปลี่ยนแปลงขนาดภาพให้โตขึ้นหรือเล็กลงกว่าภาพเดิม กรณีภาพแบบบิตแมพจะทำได้ไม่มาก นอกจากนั้นยังอาจจะทำให้ลักษณะของภาพผิดเพี้ยนไปจากเดิมด้วย เนื่องจากการเปลี่ยนแปลงขนาดภาพทำโดยวิธีการเพิ่มหรือลดพิกเซลจากที่มีอยู่เดิม ภาพที่ขยายโตขึ้นจะมองเห็นเป็นตารางสี่เหลี่ยมเรียงต่อกัน ทำให้ขาดความสวยงาม แต่ภาพแบบเวกเตอร์จะสามารถย่อและขยายขนาดได้มากกว่า โดยสัดส่วนและลักษณะของภาพยังคงคล้ายเดิม ยิ่งกว่านั้นเรายังสามารถขยายเฉพาะความกว้างหรือความสูง เพื่อมองให้เห็นเป็นภาพพอมหรืออ้วนกว่าภาพเดิมได้ด้วย

ระบบกราฟิกส์

คอมพิวเตอร์ในกลุ่มไอบีเอ็มพีซีเอกซ์ทีเอที หรือเครื่องที่ทำงานเหมือนกันจะแบ่งการแสดงผลที่จอภาพเป็นสองแบบหรือสองโหมด (mode) คือเท็กซ์โหมด(text mode)และกราฟิกส์โหมด(graphics mode) แต่ละโหมดมีรายละเอียดดังนี้

1. เท็กซ์โหมด คอมพิวเตอร์ทุกเครื่องจะแสดงผลในโหมดนี้ได้ โดยการนำตัวอักษร ตัวเลข และเครื่องหมายต่าง ๆ ที่มีอยู่ในหน่วยความจำของคอมพิวเตอร์มาแสดงที่จอภาพตามคำสั่ง แต่เนื่องจากตัวอักษร ตัวเลขและเครื่องหมายที่มีอยู่ ถูกกำหนดรูปร่างไว้แน่นอนแล้ว และมีจำนวนจำกัด จึงไม่สามารถนำมาประกอบกันให้เกิดเป็นภาพต่าง ๆ ที่ถูกต้องสวยงามได้เท่าที่ควร

2. กราฟิกส์โหมดเพื่อให้คอมพิวเตอร์สามารถแสดงผลเป็นพิกเซลได้จำนวนมากจึงได้มีการสร้างวงจรอิเล็กทรอนิกส์ เพื่อใช้สำหรับควบคุมการแสดงผลที่จอภาพ ซึ่งนิยมเรียกกันว่า ระบบกราฟิกส์ ระบบกราฟิกส์มีหลายชนิด เช่น ซีจีเอ(CGA) อีจีเอ(EGA)วีจีเอ(VGA)เฮอรัลคิวลิส(Hercules) ซึ่งแต่ละชนิดจะมีคุณสมบัติในการแสดงพิกเซลได้แตกต่างกันคือตั้งแต่ขนาด 320 x 200 พิกเซล ถึง 1024 x 768 พิกเซล

บทที่ 5

การแปลง

สิ่งที่ทำให้คอมพิวเตอร์กราฟิกส์ มีประโยชน์อย่างมากใน การสร้างภาพกราฟิกส์ก็คือ ความง่ายและความสะดวกในการเปลี่ยนแปลงภาพที่วาดขึ้นมาแล้วเช่นการเปลี่ยน เสกล เปลี่ยนมุมมอง หรือการเปลี่ยนขนาด สิ่งเหล่านี้สามารถทำได้ง่ายและรวดเร็ว

โดยใช้คอมพิวเตอร์กราฟิกส์ เพราะข้อมูลของรูปภาพต่าง ๆ หรือ ข้อมูลทางกราฟิกส์ ได้ถูกป้อนเก็บไว้ในคอมพิวเตอร์ เพียงแค่เปลี่ยนแปลงวิธีที่จะแปลงข้อมูลเหล่านี้ออกมาเป็นภาพเท่านั้นเราก็จะได้ภาพในลักษณะที่ต้องการ การเปลี่ยนแปลงนี้อาศัยการคำนวณทางคณิตศาสตร์ โดยการเปลี่ยนแปลงพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการสร้างภาพเท่านั้น เราเรียกวิธีการนี้ว่าการแปลง (Transformation)

1. เมตริกซ์ (Matrix)

โดยพื้นฐานแล้ว ภาพของระบบคอมพิวเตอร์กราฟิกส์ ถูกสร้างขึ้นด้วยส่วนของเส้นตรง หรือเวกเตอร์มาประกอบกัน ซึ่งเราสามารถกำหนดเส้นตรงหรือเวกเตอร์เหล่านี้ได้ด้วยจุดปลายทั้งสองของมัน การที่จะเปลี่ยนแปลงลักษณะภาพที่วาดจำเป็นจะต้องใช้การคำนวณทางคณิตศาสตร์กระทำกับจุดต่าง ๆ เหล่านี้ วิธีหนึ่งที่จะช่วยให้เข้าใจได้ง่ายก็คือการใช้เมตริกซ์เข้าช่วย หรืออาจจะจำสูตรไปใช้งานได้เช่นกัน

เราใช้เมตริกซ์ 2 มิติเพื่อช่วยทำการแปลงโดยแทนจุดต่าง ๆ ด้วยเมตริกซ์ขนาด 1×2 เช่นจุด $(5,6)$ แทนด้วยเมตริกซ์ $[5,6]$ เป็นต้นและเมื่อต้องการแปลงใด ๆ ก็เพียงแต่หาเมตริกซ์ที่เหมาะสมมาคูณกับเมตริกซ์ของจุด ผลลัพธ์ที่ได้คือเมตริกซ์ใหม่ ซึ่งก็คือ จุดหรือตำแหน่งใหม่ของจุดปลายของเส้นตรงต่าง ๆ ทำให้เส้น หรือเวกเตอร์เหล่านี้มีลักษณะที่เปลี่ยนไป เมตริกซ์ที่นำมาคูณเพื่อเปลี่ยนตำแหน่ง นี้ เราเรียกว่า เมตริกซ์การแปลง (Transformation matrix) ในที่นี้จะขอกล่าวถึงเฉพาะวิธีที่ใช้ในโครงงานเท่านั้น คือ การแปลงแบบหมุน

2 การแปลงแบบหมุน (Rotation Transformation)

เป็นวิธีการเปลี่ยนภาพโดยการหมุนจุดหรือภาพ ในทิศทางทวนเข็มหรือตามเข็มนาฬิกา โดยมีจุดศูนย์กลางการหมุนอยู่ที่จุดกำเนิด สิ่งที่เราต้องการทราบก็คือ เมตริกซ์ที่จะนำมาคูณจุดที่เราต้องการหมุน เพื่อไปอยู่ที่ตำแหน่งใหม่

สมมติว่าเราทำการหมุนจุด $(1,0)$ ไปในทิศทางทวน (ทวนเข็มนาฬิกา) เป็นมุม a องศา ตำแหน่งใหม่จะเป็นจุด $(\cos(a), \sin(a))$ แสดงดังรูป ถ้าเมตริกซ์การแปลงแบบหมุนคือ

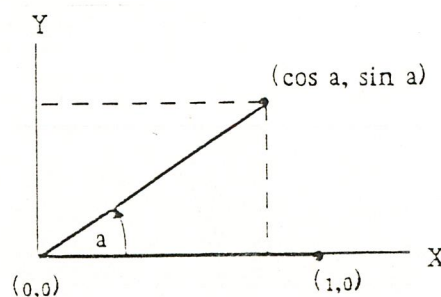
$$S = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

ดังนั้นจะได้ว่า

$$[\cos(a), \sin(a)] = [0 \ 1] \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \dots(1)$$

$$= [a \ b]$$

รูปที่ 8 แสดงการหมุนทวนเข็ม ตำแหน่ง $(1,0)$



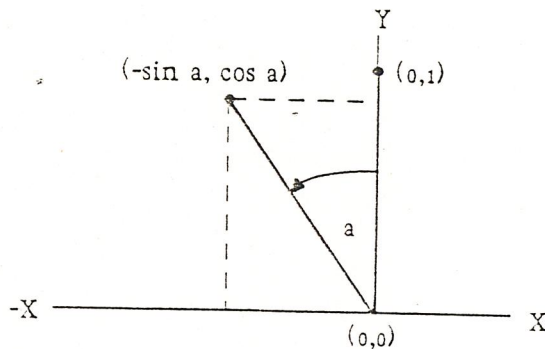
ถ้าทำการหมุนจุด $(0,1)$ ไปในทิศทวนเข็มนาฬิกา (ทวนเข็มนาฬิกา) เป็นมุม a องศา ตำแหน่งใหม่จะเป็นจุด $(-\sin(a), \cos(a))$ แสดงดังรูป ถ้าเมตริกซ์การแปลงแบบหมุนคือ

ดังนั้นจะได้ว่า

$$[-\sin(a), \cos(a)] = [0 \ 1] \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \dots(2)$$

$$= [c \ d]$$

รูปที่ 9 แสดงการหมุนทวนเข็มนาฬิกา ตำแหน่ง $(0,1)$



จากสมการ (1) และ (2) จะได้ว่า

$$A = \cos(a)$$

$$B = \sin(a)$$

$$D = \cos(a)$$

$$C = -\sin(a)$$

ดังนั้นเมตริกซ์การแปลงแบบหมุนทวนเข็มนาฬิกาคือ

$$R = \begin{bmatrix} \cos(a) & \sin(a) \\ \sin(a) & \cos(a) \end{bmatrix}$$

ดังนั้นเมตริกซ์การแปลงแบบหมุนตามเข็มนาฬิกาคือ

$$R = \begin{bmatrix} \cos(a) & \sin(a) \\ \sin(a) & \cos(a) \end{bmatrix}$$

เช่น ต้องการหมุนจุด $P1(3,2)$ ในทิศทวนเข็มนาฬิกาเป็นมุม 30 องศาเมตริกซ์การแปลงจะเป็น

$$P2 = [3 \ 2] \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

$$= [1.598 \ 3.232]$$

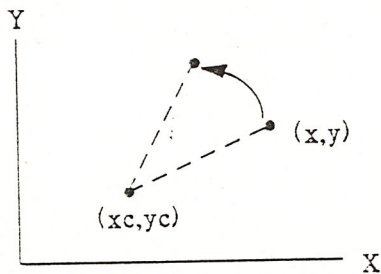
การหมุนรอบจุดใด ๆ

การแปลงแบบหมุนที่กล่าวมา เป็นการหมุนรอบจุดกำเนิดเท่านั้น แต่ต่อไปจะเป็นการหมุนรอบจุดใด ๆ (X_c, Y_c)

วิธีที่จะทำการหมุนแบบนี้มีขั้นตอนอยู่ 3 ขั้นตอนตามลำดับ คือ

- 1) ทำการย้ายภาพเพื่อให้จุดศูนย์กลางของการหมุน (X_c, Y_c) ไปอยู่ที่จุดกำเนิด
- 2) ทำการหมุนรอบจุดกำเนิด
- 3) ย้ายภาพเพื่อให้จุดศูนย์กลางของการหมุนกลับไปอยู่ที่ตำแหน่งเดิม

รูปที่ 10 แสดงการหมุนรอบจุดใด ๆ



เพื่อความสะดวกเราอาจเขียนเป็นสูตรสำหรับการหมุนจุด $P(x, y)$ รอบจุด (X_c, Y_c) เป็นมุม a องศาได้เป็นจุดใหม่ $P'(x', y')$ ดังนี้

$$x' = (x - x_c) \cos(a) - (y - y_c) \sin(a) + x_c$$

$$y' = (x - x_c) \sin(a) + (y - y_c) \cos(a) + y_c$$

บทที่ 6

รูปแบบการแสดงผลทางหน้าจอคอมพิวเตอร์

การแสดงผลของโครงการงาน Air Traffic Approach Control Simulator ทางหน้าจอคอมพิวเตอร์จะใช้รูปแบบการแสดงผล ให้เหมือนการแสดงผลทางจอเรดาร์ที่ใช้ควบคุมการร่อนลงสู่สนามบินของเครื่องบินใช้งานร่วมกับระบบกราฟิกส์ของเทอร์โบปาสคาลเวอร์ชัน 7.0 ทำการซิมูเลตข้อมูลต่างๆที่เกี่ยวข้องในงานควบคุมการจราจรทางอากาศ เช่น จอเรดาร์และรายละเอียดที่เจ้าหน้าที่จราจรอากาศต้องการ, เครื่องบิน, และสามารถใช้คำสั่งที่สร้างขึ้นในการควบคุมเครื่องบินที่เป็นเป้าในจอให้สามารถจอดในสนามบิน (Landing) และบินขึ้นได้ เพื่อประโยชน์ในการฝึกความชำนาญในงานควบคุมดังกล่าวให้กับเจ้าหน้าที่ควบคุมการจราจรทางอากาศ ส่งผลในการเพิ่มประสิทธิภาพ และความปลอดภัยในงานควบคุมฯ ให้มีมากยิ่งขึ้น ก่อนที่จะกล่าวถึงรูปแบบการแสดงผลของโครงการงานทางหน้าจอคอมพิวเตอร์นั้น จะขอกกล่าวถึงรูปแบบการแสดงผลของหน้าจอเรดาร์ก่อนดังนี้

โดยทั่วไปแล้ว รายละเอียดที่จำเป็นต้องแสดงในหน้าจอเรดาร์ เพื่อเป็นข้อมูลที่ใช้สำหรับ งานควบคุมการจราจรทางอากาศให้กับเจ้าหน้าที่ควบคุมการจราจรทางอากาศมีดังต่อไปนี้คือ

1. นามเรียกขานของเครื่องบิน
2. ความสูงของเครื่องบิน
3. ทิศทางของเครื่องบิน
4. ระยะทางที่ห่างจากสนามบิน

จากรายละเอียดที่ต้องใช้ในการแสดงผลทางหน้าจอของเรดาร์ ที่กล่าวมาข้างต้นนั้นเราสามารถกำหนดรูปแบบการแสดงผลของโครงการงานได้ดังต่อไปนี้คือ

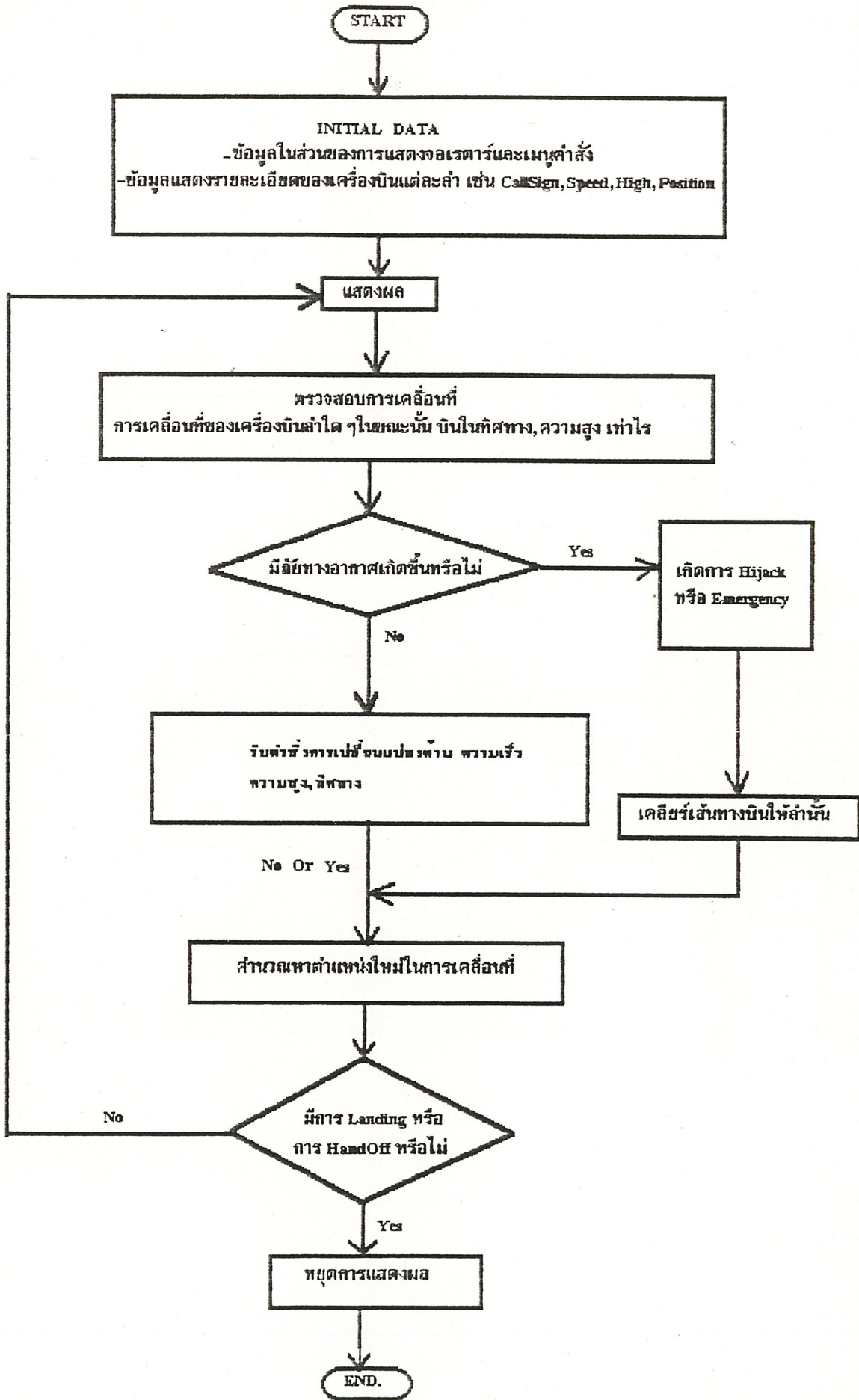
แบ่งหน้าจอคอมพิวเตอร์ออกเป็น 2 ส่วนคือ

1. ส่วนที่แสดงจอเรดาร์ซึ่งจะมีลักษณะเป็นวงกลมที่มีสเกลสำหรับการบอกรายละเอียดในเรื่องของมุมรอบวงกลมเพื่อใช้ในการกำหนดตำแหน่งของเครื่องบินพื้นที่ในระยะรัศมีของวงกลมดังกล่าวจะหมายถึงพื้นที่ ที่สถานีควบคุมการจราจรทางอากาศนั้นรับผิดชอบอยู่ และโดยปกติจุดศูนย์กลางของวงกลม หมายถึงที่ตั้งของสถานีควบคุมการจราจรทางอากาศและในส่วนของจอเรดาร์นี้ จะต้องแสดงการเคลื่อนที่ของเครื่องบินที่อยู่ในความรับผิดชอบของสถานีควบคุมฯ ด้วย และเพื่อความชัดเจนในการมองในโครงการงานจะใช้สีในการแสดงผลแต่ละส่วนให้แตกต่างกัน

2. ส่วนที่แสดงรายละเอียดของเครื่องบินแต่ละลำ โดยจะจัดให้อยู่ในด้านขวามือของจอคอมพิวเตอร์แต่เป็นพื้นที่เล็ก ๆ (ส่วนของวงกลมจอเรดาร์จะเป็นพื้นที่ส่วนมาก และจัดให้อยู่ในด้านซ้ายมือของจอคอมพิวเตอร์) ที่พอจะแสดงรายละเอียดของเครื่องบินแต่ละลำเป็นภาษาของข้อความได้อย่างครบถ้วน เช่น นามเรียกขานของเครื่องบิน, ทิศทาง, ความสูง และระยะทางจากสนามบิน เป็นต้น

3. มีระบบป้องกันภัยทางอากาศ เมื่ออาจจะเกิดการชนกันของเครื่องบิน(เป้า) สีของเครื่องบินจะเปลี่ยนไป ทำให้สังเกตเห็นได้โดยง่าย

และในหน้าถัดไปเป็นไฟล์ชาร์ตของการเขียนโปรแกรมของโครงการงาน



บทที่ 7

สรุปวิจารณ์

จากการศึกษาการเขียนโปรแกรมในการทำโครงการ Air Traffic Approach Control Simulation ในการเขียนโปรแกรมจะใช้ Graphic Turbo Pascal V.7 เพื่อทำการ Simulated Data ในการแสดงผล ทั้งในส่วนของการแสดงผลจอเรดาร์ และ ข้อมูลในส่วนของเครื่องบิน ตลอดจนวิธีการทำให้ภาพ(เครื่องบิน)เคลื่อนที่นั้น มีหลายสิ่งที่มีผู้จัดทำคิดว่ายังเป็นข้อเสียที่ควรจะต้องแก้ไข เช่น เนื่องจากหัวใจสำคัญของการสร้างภาพเคลื่อนที่คือ การที่เราสามารถหาวิธีการเปลี่ยนแปลงข้อมูลที่มีค่าแน่นอนค่าหนึ่ง ให้สามารถมีการแสดงผลในจอคอมพิวเตอร์ได้ในหลายๆลักษณะ และเหล่านั้นที่กล่าวมาก็แล้วแต่ผู้ออกแบบโปรแกรมคนใดจะสามารถสร้างสรรค์วิธีการขึ้นมาได้ อีกทั้งตำราเกี่ยวกับระบบกราฟิกส์ และความสนใจในเรื่องดังกล่าวของบ้านเรายังมีน้อย จึงค้นคว้าได้ค่อนข้างลำบาก สำหรับปัญหาต่างๆที่พบในการทำโครงการ คือ ปัญหาเรื่องของระบบโคออร์ดิเนตของจอภาพจะตรงข้ามกับระบบโคออร์ดิเนตที่เราใช้กัน ดังอธิบายในเนื้อหาที่ผ่านมา เป็นปัญหาในการคำนวณหาตำแหน่งของจุดต่างๆที่ต้องการทราบตำแหน่งที่แน่นอน และปัญหาเรื่องความละเอียดของจอภาพ ในโครงการใช้จอภาพ VGAMed มีความละเอียดไม่มากนัก (แต่มีเพจ 2 เพจ ผู้จัดทำใช้การสลับเพจในการสร้างภาพเคลื่อนที่) เมื่อมีการสร้างภาพเส้นตรงจึง มีการเกิด Aliasing คือลักษณะเส้นที่ปรากฏบนจอไม่ราบเรียบ อันเนื่องมาจากการนำจุดมาประกอบเป็นภาพ ถ้าขนาดของจุดยังมีขนาดใหญ่มากขึ้น ความขรุขระของเส้นก็จะยังมีมากยิ่งขึ้น วิธีการแก้ปัญหานั้นที่พื้นฐานที่สุดก็คือ การทำให้พิกเซลมีขนาดเล็กลง เคยมีการศึกษาพบว่าถ้าจุดในการสร้างภาพถ้าจุดในการสร้างภาพมีขนาดเล็กกว่า $1/300$ นิ้ว (300 จุดต่อนิ้ว)ตาของคนเราจะไม่สามารถสังเกตเห็นจุดเหล่านั้นได้ นั่นคือไม่สามารถสังเกตเห็นการเกิดอิลีแอสซิ่ง

```

PROGRAM CONTROL;
USES Crt, Graph;
Type St8 = String[8];
      P_Rec = RECORD
          CenX, CenX1 : Integer;
          CenY, CenY1 : Integer;
          PosX       : Integer;
          PosY       : Integer;
          Angle_P    : Integer;
          Angle_New_P : Integer;
          Length_P   : Integer;
          LengthDir  : Integer;
          CallSign   : St8;
          Speed      : Integer;
          High       : Integer;
          Chk_P      : Boolean;
          View_P     : Boolean;
          Dan        : Boolean;
          Dir        : Boolean;
          Landing    : Boolean;
          ChkDir     : Integer;
          LedX       : Integer;
          LedY       : Integer;
          Pic        : Pointer;
          History_TrialX : ARRAY [1..3] OF Byte;
          History_TrialY : ARRAY [1..3] OF Byte;
          StX, StY    : Integer;
      END;

```

```

Var P : Array[1..10] OF P_Rec;
      GraphDriver, GraphMode, Length : integer;
      PicChr : Pointer;
      Ch      : Char;
      ChkP    : Boolean;
      N       : Byte;
      CenterX, CenterY : integer;
      Dx      : integer;
      C, D, E, chkr    : Boolean;
      St : String; Pt : integer;
      NP : Byte;
      Warn : Byte;
Procedure CallGraph;
begin
    GraphDriver :=VGA;
    GraphMode   :=VGAMed;
    InitGraph(GraphDriver, GraphMode, 'C:\TP\BGI');
end;

```

```

Procedure S_1(X1, Y1, X2, Y2 : integer);
Var A, B, E, F : integer;
begin
    A :=((x2+x1) DIV 2);
    B :=((y2+y1) DIV 2);
    E :=A;
    F :=B;
    Repeat
        If A>=x1 Then A :=A-1; If B>=y1 Then B :=B-1;
        If E<=x2 Then E :=E+1; If F<=y2 Then F :=F+1;
        Rectangle(A, B, E, F);
    Until
        (A<=x1) and (B<=y1) and (E>=x2) and (F>=y2);
end;

```

```

Procedure BackGrnd;
begin {BACKGROUND}
    SetFillStyle(SolidFill,Blue);
    Bar(0,0,465,387);
    SetLineStyle(0,0,1);
    SetColor(LightGray);
    Circle(229,175,212);
    SetFillStyle(SolidFill,Black);
    FloodFill(229,175,LightGray);
    SetColor(0);
    Circle(229,175,212);
    SetLineStyle(0,0,3);
    SetColor(LightGreen);
    Rectangle(0,0,465,349);
    SetColor(LightMagenta);
    Rectangle(5,4,461,347);
    SetLineStyle(0,0,1);
    SetColor(Yellow);
    Circle(229,175,210);

{ANGLE}
    SetLineStyle(0,0,1);
    SetColor(Yellow);
    Circle(229,175,210);

{Draw Thin Line}
    SetLineStyle(0,0,1);

{10}    MoveTo (437,200); LineTo (428,198);
        MoveTo(440,198); OutText('10');
{11}    MoveTo (427,227); LineTo (419,224);
        MoveTo(432,227); OutText('11');

{13}    MoveTo (395,272); LineTo (387,267);
        MoveTo(400,274); OutText('13');
{14}    MoveTo (373,293); LineTo (365,289);
        MoveTo(378,298); OutText('14');

{16}    MoveTo (313,322); LineTo (309,317);
        MoveTo(313,328); OutText('16');
{17}    MoveTo (280,332); LineTo (277,325);
        MoveTo(278,339); OutText('17');

{19}    MoveTo (200,333); LineTo (202,328);
        MoveTo(187,340); OutText('19');
{20}    MoveTo (160,327); LineTo (162,322);
        MoveTo(150,333); OutText('20');

{22}    MoveTo (100,300); LineTo (106,295);
        MoveTo(82,305); OutText('22');
{23}    MoveTo (76,282); LineTo (83,276);
        MoveTo(55,287); OutText('23');

{25}    MoveTo (35,230); LineTo (43,226);
        MoveTo(12,228); OutText('25');
{26}    MoveTo (25,203); LineTo (35,201);
        MoveTo(4,200); OutText('26');

{28}    MoveTo (23,145); LineTo (33,148);
        MoveTo(4,138); OutText('28');
{29}    MoveTo (36,110); LineTo (45,113);
        MoveTo(18,106); OutText('29');

{31}    MoveTo (75,63); LineTo (83,68);

```

```

    MoveTo(60,50);    OutText('31');
{32}  MoveTo (102,45); LineTo (107,50);
    MoveTo(90,35);    OutText('32');

{34}  MoveTo (168,20); LineTo (172,28);
    MoveTo(162,10);   OutText('34');
{35}  MoveTo (203,15); LineTo (204,22);
    MoveTo(195,5);    OutText('35');

{01}  MoveTo(283,19); LineTo(280,26);
    MoveTo(275,6);    OutText('01');
{02}  MoveTo(323,30); LineTo(320,36);
    MoveTo(322,21);   OutText('02');

{04}  MoveTo(381,65); LineTo(374,69);
    MoveTo(385,57);   OutText('04');
{05}  MoveTo(401,83); LineTo(393,88);
    MoveTo(410,78);   OutText('05');

{07}  MoveTo(429,128); LineTo(422,130);
    MoveTo(435,125);  OutText('07');
{08}  MoveTo(435,150); LineTo(428,152);
    MoveTo(442,148);  OutText('08');

```

```
{ Draw Thick Line }
```

```

    SetColor(LightRed);
    SetLineStyle(0,0,1);

{09}  MoveTo (439,175); LineTo (429,175);
{12}  MoveTo (412,250); LineTo (403,245);
{15}  MoveTo (349,309); LineTo (342,302);
{18}  MoveTo (240,335); LineTo (240,325);
{21}  MoveTo (130,317); LineTo (137,310);
{24}  MoveTo (50,255);   LineTo (60,248);
{27}  MoveTo (19,175);  LineTo (30,175);
{30}  MoveTo(55,83);    LineTo(62,88);
{33}  MoveTo (135,30);  LineTo (140,40);
{36}  MoveTo (240,13);  LineTo (240,23);
{03}  MoveTo (359,49);  LineTo (349,55);
{06}  MoveTo (418,103); LineTo (408,110);

```

```
{Number Of Angle}
```

```

    MoveTo (445,173); OutText('09');
    MoveTo (418,252); OutText('12');
    MoveTo (352,315); OutText('15');
    MoveTo (233,342); OutText('18');
    MoveTo (118,322); OutText('21');
    MoveTo (30,261);  OutText('24');
    MoveTo (0,171);   OutText('27');
    MoveTo (38,75);   OutText('30');
    MoveTo (125,18);  OutText('33');

```

```
MoveTo (232,2); OutText('36');
MoveTo (236,25); OutText('N');
MoveTo (360,40); OutText('03');
MoveTo (422,97); OutText('06');
SetLineStyle(0,$FF,1);
```

```
end;
```

```
Procedure Rangel(Co : Byte);
```

```
Var E1 : Integer;
```

```
begin
```

```
Dx:= -91;
```

```
IF Co = 1 Then
```

```
begin
```

```
SetColor(LightCyan)
```

```
end
```

```
Else
```

```
begin
```

```
SetColor(0);
```

```
end;
```

```
{RUNPLANE}
```

```
SetLineStyle(0,$FF,1);
```

```
MoveTo(GetMaxX Div 2+Dx-10,GetMaxY Div 2+2);
```

```
LineTo(GetMaxX Div 2+Dx+10,GetMaxY Div 2+2);
```

```
MoveTo(GetMaxX Div 2+Dx-10,GetMaxY Div 2-2);
```

```
LineTo(GetMaxX Div 2+Dx+10,GetMaxY Div 2-2);
```

```
{VOR}
```

```
SetLineStyle(2,$AA,1);
```

```
Line(GetMaxX Div 2+Dx-1,GetMaxY Div 2+1,  
GetMaxX Div 2+Dx+60,GetMaxY Div 2-60);
```

```
Line(GetMaxX Div 2+Dx+1,GetMaxY Div 2+1,  
GetMaxX Div 2+Dx+62,GetMaxY Div 2-58);
```

```
Line(GetMaxX Div 2+Dx,GetMaxY Div 2,  
GetMaxX Div 2+Dx+25*3,GetMaxY Div 2-52*3);
```

```
MoveTo(GetMaxX Div 2+Dx,GetMaxY Div 2);
```

```
Line(GetMaxX Div 2+Dx,GetMaxY Div 2,  
GetMaxX Div 2+Dx+46*4,GetMaxY Div 2-20*4);
```

```
{RANGE1}
```

```
MoveTo(125,75);
```

```
SetTextStyle(2,0,0);
```

```
OutText('RANGE=5 N.M.');
```

```
For E1 := 1 To 6 Do
```

```
Circle(GetMaxX Div 2+Dx,GetMaxY Div 2,30*E1);
```

```
SetLineStyle(0,$FF,1);
```

```
end;
```

```
Procedure Range2( Co : Byte);
```

```
Var E2 : Integer;
```

```
begin
```

```
Dx:= -91;
```

```
IF Co = 1 Then
```

```
begin
```

```
SetColor(LightCyan)
```

```
end
```

```
Else
```

```
begin
```

```
SetColor(0);
```

```
end;
```

```
{RUNPLANE}
```

```
SetLineStyle(0,$FF,1);  
MoveTo(GetMaxX Div 2+Dx-10,GetMaxY Div 2+2);  
LineTo(GetMaxX Div 2+Dx+10,GetMaxY Div 2+2);  
MoveTo(GetMaxX Div 2+Dx-10,GetMaxY Div 2-2);  
LineTo(GetMaxX Div 2+Dx+10,GetMaxY Div 2-2);
```

```
{VOR}
```

```
SetLineStyle(2,$AA,1);  
Line(GetMaxX Div 2+Dx-1,GetMaxY Div 2+1,  
GetMaxX Div 2+Dx+60,GetMaxY Div 2-60);  
Line(GetMaxX Div 2+Dx+1,GetMaxY Div 2+1,  
GetMaxX Div 2+Dx+62,GetMaxY Div 2-58);  
Line(GetMaxX Div 2+Dx,GetMaxY Div 2,  
GetMaxX Div 2+Dx+17*3,GetMaxY Div 2-52*3);  
MoveTo(GetMaxX Div 2+Dx,GetMaxY Div 2);  
Line(GetMaxX Div 2+Dx,GetMaxY Div 2,  
GetMaxX Div 2+Dx+61*3,GetMaxY Div 2-25*3);
```

```
{RANGE2}
```

```
MoveTo(90,60);  
SetTextStyle(2,0,0);  
OutText('RANGE=10 N.M.');
```

```
For E2 := 1 To 4 Do
```

```
Circle(GetMaxX Div 2+Dx,GetMaxY Div 2,50*E2);  
SetLineStyle(0,$FF,1);
```

```
end;
```

```
Procedure AirRoute( Co :Boolean);
```

```
Var CenterX,CenterY :integer;  
Dx :integer;  
Xasp,Yasp :word;
```

```
begin
```

```
Dx := -50;  
If Co Then  
begin  
SetColor(White)  
end  
Else  
begin  
SetColor(0)  
end;
```

```
SetLineStyle(1,0,1);  
CenterX :=GetMaxX Div 2+DX;  
CenterY :=GetMaxY Div 2;
```

```
{10} MoveTo(CenterX+DX+99,CenterY+18);  
LineTo(CenterX+DX+208,CenterY+60);  
  
{35} MoveTo(CenterX+DX+82,CenterY+58);  
LineTo(CenterX+DX+164,CenterY+115);  
  
{60} MoveTo(CenterX+DX+50,CenterY+87);  
LineTo(CenterX+DX+100,CenterY+143);  
  
{83} MoveTo(CenterX+DX+13,CenterY+100);  
LineTo(CenterX+DX+22,CenterY+160);  
  
{95} MoveTo(CenterX+DX-9,CenterY+100);  
LineTo(CenterX+DX-17,CenterY+160);
```

```

{120} MoveTo (CenterX+DX-50, CenterY+87);
      LineTo (CenterX+DX-100, CenterY+145);

{140} MoveTo (CenterX+DX-77, CenterY+65);
      LineTo (CenterX+DX-153, CenterY+110);

{160} MoveTo (CenterX+DX-94, CenterY+35);
      LineTo (CenterX+DX-187, CenterY+50);

{180} MoveTo (CenterX+DX-100, CenterY+0);
      LineTo (CenterX+DX-200, CenterY+0);

{205} MoveTo (CenterX+DX-91, CenterY-43);
      LineTo (CenterX+DX-170, CenterY-80);

{230} MoveTo (CenterX+DX-65, CenterY-77);
      LineTo (CenterX+DX-128, CenterY-120);

{250} MoveTo (CenterX+DX-34, CenterY-94);
      LineTo (CenterX+DX-68, CenterY-155);

{280} MoveTo (CenterX+DX+18, centerY-99);
      LineTo (CenterX+DX+34, CenterY-163);

{310} MoveTo (CenterX+DX+65, CenterY-77);
      LineTo (CenterX+DX+120, CenterY-145);

{330} MoveTo (CenterX+DX+87, CenterY-50);
      LineTo (CenterX+DX+160, CenterY-118);

{345} MoveTo (CenterX+DX+98, CenterY-26);
      LineTo (CenterX+DX+190, CenterY-80);

{350} MoveTo (CenterX+DX+98, CenterY-18);
      LineTo (CenterX+DX+213, CenterY-30);
          SetLineStyle (0, $FF, 1);
end;

```

```

Procedure MAP1 (Co : Boolean);

```

```

  Var CenterX, CenterY      : Integer;
      x, y, Dx              : integer;

```

```

begin

```

```

  IF Co Then

```

```

    SetColor (LightMagenta)

```

```

  Else

```

```

    SetColor (0);

```

```

    SetLineStyle (0, $FF, 1);

```

```

{AERODROME AREA}

```

```

  MoveTo (180, 100);

```

```

  LineTo (255, 68);

```

```

  LineTo (330, 80);

```

```

  LineTo (375, 135);

```

```

  LineTo (400, 200);

```

```

  LineTo (320, 270);

```

```

  LineTo (200, 290);

```

```

  LineTo (130, 255);

```

```

  LineTo (150, 215);

```

```

  LineTo (105, 195);

```

```

  LineTo (95, 125);

```

```

  LineTo (180, 125);

```

```

  LineTo (180, 100);

```

{ENVIRONMENT AREA}

```
    SetLineStyle(1,1,1);
    Moveto(180,100);
{1}   Lineto(120,50);
      Moveto(255,68);
{2}   Lineto(250,20);
      Moveto(330,80);
{3}   Lineto(360,55);
      Moveto(375,135);
{4}   Lineto(425,110);
      Moveto(400,200);
{5}   Lineto(430,200);
      Moveto(320,270);
{6}   Lineto(340,290);
      Moveto(200,290);
{7}   Lineto(200,320);
      Moveto(150,215);
{8}   Lineto(50,230);
      Moveto(105,195);
{9}   Lineto(40,200);
      Moveto(95,125);
{10}  Lineto(60,115);
```

{RESTRICT AREA}

```
    SetTextStyle(0,0,0);
    MoveTo(40,175); OutText('R1');
    MoveTo(110,70); OutText('R2');
    MoveTo(90,280); OutText('MAP1');
    SetLineStyle(0,$FF,1);
```

End;

Procedure MAP1x(Co : Boolean);

Var CenterX,CenterY :integer;
 x,y,Dx :integer;

begin

```
    IF Co Then
    SetColor(LightMagenta)
    Else
    SetColor(0);
```

{AERODROME AREA}

```
    SetLineStyle(0,$FF,1);
    MoveTo(200,110);
    LineTo(260,85);
    LineTo(320,100);
    LineTo(345,140);
    LineTo(365,200);
    LineTo(310,250);
    LineTo(200,265);
    LineTo(160,250);
    LineTo(175,210);
    LineTo(130,180);
    LineTo(120,140);
    LineTo(200,135);
    LineTo(200,110);
```

{ENVIRONMENT EREA}

```
    SetLineStyle(1,1,1);
    Moveto(200,110);
{1}   Lineto(120,50);
      Moveto(260,85);
{2}   Lineto(250,20);
      Moveto(320,100);
{3}   Lineto(360,55);
```

```

Moveto(345,140);
{4} Lineto(425,110);
Moveto(365,200);
{5} Lineto(430,200);
Moveto(310,250);
{6} Lineto(340,290);
Moveto(200,265);
{7} Lineto(200,320);
Moveto(175,210);
{8} Lineto(50,230);
Moveto(130,180);
{9} Lineto(40,200);
Moveto(120,140);
{10} Lineto(60,115);

```

```
{RESTRICT EREA}
```

```

SetTextStyle(0,0,0);
MoveTo(95,170); OutText('R1');
MoveTo(155,100); OutText('R2');
MoveTo(120,250); OutText('MAP1');
SetLineStyle(0,$FF,1);

```

```
end;
```

```
Procedure Getpic(x1,y1,x2,y2:integer;Var max:pointer);
```

```
Var Size : Word;
```

```
Begin
```

```
Size:=imagesize(x1,y1,x2,y2);
```

```
Getmem(max,size);
```

```
Getimage(x1,y1,x2,y2,max^);
```

```
End;
```

```
Procedure Retpic(x1,y1,x2,y2:integer;Var max:pointer);
```

```
Var Size : Word;
```

```
begin
```

```
Putimage(x1,y1,max^,normalput);
```

```
Size:=Imagesize(x1,y1,x2,y2);
```

```
Freemem(max,size)
```

```
End;
```

```
Function Direction(Angle:integer) : integer;
```

```
Begin
```

```
Direction:= 180-Angle;
```

```
End;
```

```
Function Radian(Angle:integer) : Real ;
```

```
begin
```

```
Radian := (Direction(Angle)) * Pi/180;
```

```
end;
```

```
Procedure ViewPort(No : Byte);
```

```
Var X0,Y0 : Integer;
```

```
Q,X,Y : LongInt;
```

```
Begin
```

```
X := P[No].PosX-239;
```

```
Y := 175-P[No].PosY;
```

```
Q := Round(Sqrt(Abs(Sqr(X)+Sqr(Y))));
```

```
{ Str(Y,St1);
```

```
St :=St1;
```

```
Str(Q,St1);
```

```
St := St+' '+st1;
```

```
GetPic(400,40,500,60,PicChr);
```

```
OutTextXY(400,40,st);}
```

```

IF Q < 180 Then P[No].View_P := True Else P[No].View_P := False;

End;

Procedure DangerousRange(NoP1, NoP2 : Byte);
Var P1X, P1Y, P2X, P2Y : Integer;

    Q, X, Y : LongInt;
Begin
    P1X := P[NoP1].PosX;
    P1Y := P[NoP1].PosY;
    P2X := P[NoP2].PosX;
    P2Y := P[NoP2].PosY;

    X := P2X - P1X;
    Y := P2Y - P1Y;
    Q := Round(Sqrt(Abs(Sqr(X) + Sqr(Y))));

    IF Q <= 50 Then
        begin P[NoP1].Dan := True;
              P[NoP2].Dan := True;
        end
    Else
        begin P[NoP1].Dan := False;
              P[NoP2].Dan := False;
        end;
End;

Procedure New_Pos(No : Byte);
Var ChkQ : Byte;
Begin
    IF ((P[No].PosX > P[No].CenX) And (P[No].PosY < P[No].CenY)) Then ChkQ := 1;
    IF ((P[No].PosX > P[No].CenX) And (P[No].PosY > P[No].CenY)) Then ChkQ := 2;
    IF ((P[No].PosX < P[No].CenX) And (P[No].PosY > P[No].CenY)) Then ChkQ := 3;
    IF ((P[No].PosX < P[No].CenX) And (P[No].PosY < P[No].CenY)) Then ChkQ := 4;
    Case ChkQ OF
        1 : Begin
            P[No].CenX := P[No].PosX + Round(P[No].Length_P * Sin(Radian(P[No].Angle_P)));
            P[No].CenY := P[No].PosY + Round(P[No].Length_P * Cos(Radian(P[No].Angle_P)));
            End;
        2 : Begin
            P[No].CenX := P[No].PosX + Round(P[No].Length_P * Sin(Radian(P[No].Angle_P)));
            P[No].CenY := P[No].PosY + Round(P[No].Length_P * Cos(Radian(P[No].Angle_P)));
            End;
        3 : Begin
            P[No].CenX := P[No].PosX + Round(P[No].Length_P * Sin(Radian(P[No].Angle_P)));
            P[No].CenY := P[No].PosY + Round(P[No].Length_P * Cos(Radian(P[No].Angle_P)));
            End;
        4 : Begin
            P[No].CenX := P[No].PosX + Round(P[No].Length_P * Sin(Radian(P[No].Angle_P)));
            P[No].CenY := P[No].PosY + Round(P[No].Length_P * Cos(Radian(P[No].Angle_P)));
            End;
    End;
    IF P[No].Chk_P Then P[No].Angle_P := P[No].Angle_P + 180
    Else P[No].Angle_P := P[No].Angle_P - 180;
End;

Procedure Rotate(No : Byte);
Var St, St1 : String;
    Xt, Yt : Integer;
Begin
    {
        GetPic(400, 40, 600, 60, PicChr);
        Str(Angle_New, St1);
    }

```

```

    St := St1;
    Str(P[No].CenX,St1);
    St := St+'X = '+St1;
    Str(P[No].CenY,St1);
    St := St+' Y = '+St1;
    OutTextXY(400,40,St);}
P[No].PosX :=P[No].CenX+Round(P[No].Length_P*Sin(Radian(P[No].Angle_P)));
P[No].PosY :=P[No].CenY+Round(P[No].Length_P*Cos(Radian(P[No].Angle_P)));
{
    RetPic(400,40,600,60,PicChr);}
IF P[No].chk_P then
Begin
    IF P[No].Angle_New_P = 0 Then P[No].Angle_New_P := 360
    Else
        Begin Dec(P[No].Angle_New_P);
            IF P[No].Angle_P = 0 Then P[No].Angle_P:= 360
            Else Dec(P[No].Angle_P);
        End;
End
else
Begin
IF P[No].Angle_New_P = 360 Then P[No].Angle_New_P := 0
Else
    Begin Inc(P[No].Angle_New_P);
        IF P[No].Angle_P = 360 Then P[No].Angle_P:= 0
        Else Inc(P[No].Angle_P);
    End;
End;
End;

```

End;

```

Procedure Run_Dir(No : Byte);
Begin
P[No].PosX :=P[No].CenX1+Round(P[No].LengthDir*Sin(Radian(P[No].Angle_New_P)));
P[No].PosY :=P[No].CenY1+Round(P[No].LengthDir*Cos(Radian(P[No].Angle_New_P)));
Inc(P[No].LengthDir);
End;

```

```

Procedure InitialData1;
begin
    P[1].CenX :=400;
    P[1].CenY :=80;
    P[1].Length_P :=20;
    P[1].LengthDir :=1;
    P[1].Angle_P :=0;
    P[1].Angle_new_P := 90;
    P[1].chk_P := False;
    P[1].CallSign := 'THA600';
    P[1].Speed := 250;
    P[1].High := 100;
end;

```

```

Procedure InitialData2;
begin
    P[2].CenX :=100;
    P[2].CenY :=200;
    P[2].Length_P :=40;
    P[2].LengthDir :=1;
    P[2].Angle_P :=180;
    P[2].Angle_new_P := 90;
    P[2].chk_P := True;
    P[2].CallSign := 'AMC800';
    P[2].Speed := 150;
    P[2].High := 150;

```

end;

{Procedure InitialData3;

begin

P3.CenX :=200; P3.CenY :=300;
Length :=30;
P3.Angle_P :=10;
P3.Angle_new_P := 90;
P3.chk_P := False;

end;

Procedure InitialData4;

begin

P2.CenX :=300; P4.CenY :=400;
Length :=50;
P4.Angle_P :=20;
P4.Angle_new_P := 90;
P4.chk_P := True;

end;

Procedure InitialData5;

begin

P5.CenX :=400; P5.CenY :=500;
Length :=20;
P5.Angle_P :=30;
P5.Angle_new_P := 90;
P5.chk_P := False;

end;

Procedure InitialData6;

begin

P6.CenX :=100; P6.CenY :=600;
Length :=40;
P6.Angle_P :=40;
P6.Angle_new_P := 90;
P6.chk_P := True;

end;

Procedure InitialData7;

begin

P7.CenX :=230; P7.CenY :=450;
Length :=20;
P7.Angle_P :=50;
P7.Angle_new_P := 90;
P7.chk_P := False;

end;

Procedure InitialData8;

begin

P8.CenX :=450; P8.CenY :=260;
Length :=40;
P8.Angle_P :=60;
P8.Angle_new_P := 90;
P8.chk_P := True;

end;

Procedure InitialData9;

begin

P9.CenX :=450; P9.CenY :=115;
Length :=60;
P9.Angle_P :=70;
P9.Angle_new_P := 90;
P9.chk_P := False;

end;

```

Procedure InitialData10;
begin
    P10.CenX :=290; P10.CenY :=200;
    Length :=40;
    P10.Angle_P :=80;
    P10.Angle_new_P := 90;
    P10.chk_P := True;
end;}

```

```

Procedure InitialData;
begin
    InitialData1;
    InitialData2;
    { InitialData3;
    InitialData4;
    InitialData5;
    InitialData6;
    InitialData7;
    InitialData8;
    InitialData9;
    InitialData10;}
end;

```

```

Procedure Leader(No : Byte);
Var XL3,YL3 : Integer;
    Lt : St8;
Begin
    XL3 := (P[No].PosX+15);
    YL3 := (P[No].PosY+15);
    SetColor(15);
    SetLineStyle(0,$FF,1);
    Line(P[No].PosX,P[No].PosY,XL3,YL3);
    MoveTo(XL3+5,YL3);
    SetTextStyle(2,0,0);
    OutText(P[No].CallSign);
    MoveTo(XL3+5,YL3+10);
    Str(P[No].PosX,Lt);
    OutText(Lt);
    MoveTo(XL3+5,YL3+20);
    Str(P[No].PosY,Lt);
    OutText(Lt);
End;

```

```

Procedure Call_Rangel;
Begin
    SetActivePage(1);
    Map1x(False);
    Range2(0);
    Rangel(1);
    Map1(C);
    SetActivePage(0);
    Map1x(False);
    Range2(0);
    Rangel(1);
    Map1(C);
    Chkr := True;
End;

```

```

Procedure Call_Rage2;
Begin
    SetActivePage(1);
    Map1(False);
    Rangel(0);

```

```

Range2(1);
Maplx(C);
  SetActivePage(0);
  Map1(False);
  Range1(0);
  Range2(1);
  Maplx(C);
  Chkr := False;
End;

Procedure Call_Or_StopMap;
Begin
  C:= Not C;
  IF Chkr Then
    Begin
      SetActivePage(1);
      MAP1(C);
      SetActivePage(0);
      MAP1(C);
      End
    Else
      Begin
        SetActivePage(1);
        MAP1x(C);
        SetActivePage(0);
        MAP1x(C);
        End;
      End;
End;

Procedure Call_Or_StopAirRoute;
Begin
  E := Not E;
  SetActivePage(1);
  AirRoute(E);
  SetActivePage(0);
  AirRoute(E);
End;

Procedure For_New_Position;
Begin
  SetActivePage(1);
  P[1].chk_P := not P[1].chk_P;
  New_Pos(1);
  SetActivePage(0);
  P[1].chk_P := not P[1].chk_P;
  New_Pos(1);
End;

Procedure Swap_Direction;
Begin
  IF P[2].Dir Then
    Begin
      P[2].CenX := (P[2].PosX-P[2].CenX1)+P[2].CenX;
      P[2].CenY := (P[2].PosY-P[2].CenY1)+P[2].CenY;
      P[2].Dir := Not P[2].Dir;
      End;
    P[2].chk_P := not P[2].chk_P;
    New_Pos(2);
  End;

Procedure P1_Direct;
Begin
  IF P[1].Dir Then
    Begin

```

```

        P[1].CenX := (P[1].PosX-P[1].CenX1)+P[1].CenX;
        P[1].CenY := (P[1].PosY-P[1].CenY1)+P[1].CenY;
        End
    Else
        Begin
            P[1].CenX1 :=P[1].PosX;
            P[1].CenY1 :=P[1].PosY;
            P[1].LengthDir := 1;
            End;
        P[1].Dir := Not P[1].Dir;
    End;

Procedure P2_Direct;
Begin
    IF P[2].Dir Then
        Begin
            P[2].CenX := (P[2].PosX-P[2].CenX1)+P[2].CenX;
            P[2].CenY := (P[2].PosY-P[2].CenY1)+P[2].CenY;
            End
        Else
            Begin
                P[2].CenX1 :=P[2].PosX;
                P[2].CenY1 :=P[2].PosY;
                P[2].LengthDir := 1;
                End;
    End;

Procedure To_Landing(No : byte);
Begin
    IF (((P[NP].PosX>219) And (P[NP].PosX<265)) And
        ((P[NP].PosY>150) And (P[NP].PosY<185))) And
        ((P[NP].Angle_New_P<180) And (P[NP].Angle_New_P>90))
    Then
        Begin P[NP].Landing :=True;
            P[NP].PosX :=239;P[NP].PosY :=175;
            P[NP].CenX :=P[NP].CenX- 239;P[NP].CenY :=P[NP].CenY-175;
        End
    Else P[NP].Landing :=False;
End;

{Procedure RetString(X,Y : Integer;Var St:String);
Var Ch:Char; B:String;
  L : Integer;
Begin
    B :=St;
    St :='';
    Repeat
        IF KeyPressed Then
            Begin
                Ch :=ReadKey;
                {IF Ch = #13 Then Begin IF St = '' Then St := B;Exit;End;
                IF Ch= #8 Then}
                begin
                    SetColor(0);
                    OutTextXY(X,Y,St);
                    L :=Length(St);
                    Delete(St,L,1);
                end Else St :=St+Ch;
                SetColor(15);
                OutTextXY(X,Y,St);
                End;
            Case Ch OF
                #13 : Begin IF St = '' Then St := B;Exit;End;
                #8  : Begin

```

```

        SetColor(0);
        OutTextXY(X,Y,St);
        L :=Length(St);
        Delete(St,L,1);
        end
        Else St :=St+Ch;
        SetColor(15);
        OutTextXY(X,Y,St);
    End;
    'A','a' : Call_Rangel;
    'B','b' : Call_Range2;
    'P','p' : Call_Or_StopMap;
    'Q','q' : Call_Or_StopAirRoute;
    'R','r' : For_New_Position;
    'S','s' : Swap_Direction;
    'D','d' : P1_Direct;
    'E','e' : P2_Direct;
    End;
Until Ch = #13;
    SetColor(15);
    OutTextXY(450,30,'Ok!');
end;

Procedure RetInt(X,Y :integer;Var Pt:Integer);
Var Ch:Char; B:Integer;St:String;
    L : Integer;
    Code :Integer;
Begin
    B :=Pt;
    St:='';
    Repeat
        IF KeyPressed Then
            Begin
                Ch :=ReadKey;
                Case Ch Of
                    '0'..'9':Begin
                        St :=St+Ch;
                        End;
                    #8 : Begin
                        SetColor(0);
                        OutTextXY(X,Y,St);
                        L :=Length(St);
                        Delete(St,L,1);
                        End;
                    #13 : Begin
                        IF St = '' Then Pt := B
                        Else Begin Val(St,Pt,Code);Exit;End;
                        End;
                End;
            End;
        End;
        SetColor(15);

```

```

        OutTextXY(X,Y,St);
Until Ch = #13;
        SetColor(15);
        OutTextXY(450,30,'Ok!');

End;}

Procedure MENU;
begin
    SetTextStyle(2,0,0);
    SetColor(LightCyan);
    OutTextXY(490,3,'..Main Menu.. ');
    SetColor(LightGreen);
    Rectangle(469,2,633,15);
    Rectangle(469,17,633,30);
    Rectangle(469,32,633,45);
    Rectangle(469,47,633,60);
    SetColor(Yellow);
    Rectangle(469,62,633,190);
{   Rectangle(469,122,633,140);}
    Rectangle(469,192,633,349);
    SetColor(LightMagenta);
    OutTextXY(472,17,'R:Radar Data Selection');
    OutTextXY(472,32,'F:Fligh Data Entry');
    OutTextXY(472,47,'C:Correct Flight Plan');
    SetColor(LightCyan);
    OutTextXY(472,60,'Pending Area:');
    OutTextXY(472,193,'Control Area:');
    SetColor(White);
    OutTextXY(472,339,'Supervisor:');

{
    SetColor(Green);
    S_1(239,10,466,349);}

    SetColor(LightGreen);
    S_1(10,175,455,340);
    SetColor(LightMagenta);
    Rectangle(11,176,454,339);
    Rectangle(13,178,452,337);

SetTextStyle(0,0,0);
SetColor(Blue);
OutTextXY(180,180,'--RADAR DATA SELECT--');
Line(180,188,350,188);
SetColor(Magenta);
OutTextXY(20,197,'1 :RANGE 1 = 10 N.M. ');
OutTextXY(20,207,'2 :RANGE 1 = 5 N.M. ');
OutTextXY(20,217,'3 :MAP 1 --> ON ');
OutTextXY(20,227,'4 :MAP 1 --> OFF ');
OutTextXY(20,237,'5 :AIRROUTE --> ON ');
OutTextXY(20,247,'6 :AIRROUTE --> OFF ');
OutTextXY(20,257,'7 :SWEEP --> ON ');
OutTextXY(20,267,'8 :SWEEP --> OFF ');
SetColor(Blue);
OutTextXY(20,277,'Pressed KEY To Select : ');
OutTextXY(20,287,'Do You Want To Select Other? Y/N : ');
OutTextXY(20,297,'Are You Ready? Y/N : ');

{
    SetTextStyle(0,0,0);
    SetColor(Yellow);
    OutTextXY(180,180,'--FLIGHT DATA ENTRY--');
    Line(180,190,350,190);
    SetColor(LightMagenta);

```

```

OutTextXY(20,192,'Do You Want To Entry Data? Y/N :');
{IF KEY='Y' Then}
{
    OutTextXY(20,202,'Plane_CallSign =');
    OutTextXY(20,212,'Plane_Speed   =');
    OutTextXY(20,222,'Plane_Height   =');
    OutTextXY(20,232,'Plane_AirCRAFT Type =');
    OutTextXY(20,242,'Plane_Time In =');
    SetColor(Yellow);
    OutTextXY(20,262,'YOUR DATA READY ? Y/N :');
    {IF Key='y' Then Delete Line up}
{
    SetColor(LightMagenta);
    OutTextXY(20,272,'Do You Want To Entry Other Data Continue ?Y/N :');
    {IF Key='y' Then Delete Line up and delete data up And Repeat up}

{
    SetTextStyle(0,0,0);
    SetColor(Yellow);
    OutTextXY(180,180,'--CORRECT FLIGHT PLAN--');
    Line(180,190,360,190);
    SetColor(LightMagenta);
    OutTextXY(20,192,'Change Data Plane_CallSign : ');
    OutTextXY(20,202,'New_CallSign =');
    OutTextXY(20,212,'New_Speed   =');
    OutTextXY(20,222,'New_Height   =');
    OutTextXY(20,232,'New_AirCRAFT Type =');
    OutTextXY(20,242,'New_Time In =');
    SetColor(Yellow);
    OutTextXY(20,262,'YOUR DATA READY ? Y/N :');
    Delay(80);
    {IF Key='y' Then Delete Line up}
{
    SetColor(LightMagenta);
    OutTextXY(20,272,'Do You Want To Change Other Data Continue ?Y/N :');
    {IF Key='y' Then Delete Line up and delete data up And Repeat up}

```

End;

```

BEGIN {ACT1}
    CallGraph;
    BackGrnd;
    MENU;

{
    Bar(465,1,636,36); }
{
    SetColor(LightGreen);
    Rectangle(467,1,635,35);
    Rectangle(468,0,634,34);
    SetColor(LightMagenta);
    OutTextXY(488,15,'ESC To Main Menu');
    SetColor(0);
{
    Bar(465,280,636,350);}

```

```

SetTextStyle(0,0,0);
{Process}
    C := False;
    D := False;
    E := False;
    SetVisualPage(0);
    SetActivePage(1);
    BackGrnd;

```

```

    SetVisualPage(1);
    SetActivePage(0);
    BackGrnd;

    SetVisualPage(0);
    ChkP := True;
    InitialData;
Repeat
    SetColor(15);

For NP := 1 To 2 Do
    begin
    IF P[NP].Dir Then Run_Dir(NP) Else Rotate(NP);
For Warn := NP+1 To 2 Do
    DangerousRange(NP,Warn);
    IF P[NP].Dan Then SetColor(Red) Else SetColor(LightGreen);
To_Landing(NP);
ViewPort(NP);
    IF P[NP].Landing =False Then
        IF P[NP].View_P = True Then
            Begin
                SetActivePage(1);
                GetPic(P[NP].PosX,P[NP].PosY,P[NP].PosX+70,P[NP].PosY+70,P[NP].Pic);
                SetActivePage(0);
            End;
        IF P[NP].Landing =False Then
            IF P[NP].View_P = True Then
                Begin
                    Rectangle(P[NP].PosX,P[NP].PosY,P[NP].PosX+5,P[NP].PosY+5);
                    Leader(NP);
                    Dec(P[2].High);
                End;
            Delay(1);
        End;
For NP := 1 To 2 Do
    IF P[NP].Landing =False Then
        IF P[NP].View_P = True Then
            RetPic(P[NP].PosX,P[NP].PosY,P[NP].PosX+70,P[NP].PosY+70,P[NP].Pic);

{
    RetPic(400,40,600,60,PicChr);}

{
    IF KeyPressed Then
    Begin
    Ch := ReadKey;
    {ReadString(StX,StY,St);
    ReadInt(StX,StY,Pt);}

{
    Case Ch Of
        'A','a' :begin
            SetActivePage(1);
            Maplx(False);
            Range2(0);
            Rangel(1);
            Mapl(C);
            SetActivePage(0);
            Maplx(False);
            Range2(0);
            Rangel(1);
            Mapl(C);
            Chkr := True;
        end;

```

```

        'B','b' :begin
            SetActivePage(1);
            Map1(False);
            Rangel(0);
            Range2(1);
            Map1x(C);
            SetActivePage(0);
            Map1(False);
            Rangel(0);
            Range2(1);
            Map1x(C);
            Chkr := False;
        end;

        'P','p' :begin
            C:= Not C;
            IF Chkr Then
            Begin
                SetActivePage(1);
                MAP1(C);
                SetActivePage(0);
                MAP1(C);
            End
            Else
            Begin
                SetActivePage(1);
                MAP1x(C);
                SetActivePage(0);
                MAP1x(C);
            End;
            End;

        'Q','q' :begin
            E := Not E;
            SetActivePage(1);
            AirRoute(E);
            SetActivePage(0);
            AirRoute(E);
        end;

'R','r' : Begin
    SetActivePage(1);
    P[1].chk_P := not P[1].chk_P;
    New_Pos(1);
    SetActivePage(0);
    P[1].chk_P := not P[1].chk_P;
    New_Pos(1);
End;

'T','t' : Begin
    SetActivePage(1);
    P[2].chk_P := not P[2].chk_P;
    New_Pos(2);
    SetActivePage(0);
    P[2].chk_P := not P[2].chk_P;
    New_Pos(2);
End;

'S','s' : Begin
    IF P[2].Dir Then
    Begin
        P[2].CenX := (P[2].PosX-P[2].CenX1)+P[2].CenX;
        P[2].CenY := (P[2].PosY-P[2].CenY1)+P[2].CenY;
        P[2].Dir := Not P[2].Dir;
    End;
    P[2].chk_P := not P[2].chk_P;
    New_Pos(2);

```

```

End;
'U','u' : Begin
  IF P[1].Dir Then
  Begin
    P[1].CenX := (P[1].PosX-P[1].CenX1)+P[1].CenX;
    P[1].CenY := (P[1].PosY-P[1].CenY1)+P[1].CenY;
    P[1].Dir := Not P[1].Dir;
  End;
  P[1].chk_P := not P[1].chk_P;
  New_Pos(1);
End;
'D','d' : Begin
  IF P[1].Dir Then
  Begin
    P[1].CenX := (P[1].PosX-P[1].CenX1)+P[1].CenX;
    P[1].CenY := (P[1].PosY-P[1].CenY1)+P[1].CenY;
  End Else
  Begin
    P[1].CenX1 :=P[1].PosX;
    P[1].CenY1 :=P[1].PosY;
    P[1].LengthDir := 1;
  End;
  P[1].Dir := Not P[1].Dir;
End;
'E','e' : Begin
  IF P[2].Dir Then
  Begin
    P[2].CenX := (P[2].PosX-P[2].CenX1)+P[2].CenX;
    P[2].CenY := (P[2].PosY-P[2].CenY1)+P[2].CenY;
  End Else
  Begin
    P[2].CenX1 :=P[2].PosX;
    P[2].CenY1 :=P[2].PosY;
    P[2].LengthDir := 1;
  End;
  P[2].Dir := Not P[2].Dir;
End;
End;
} End;}
Until Ch =#13;
ReadLn;
CloseGraph

```

END.

เอกสารอ้างอิง

- (1) มงคล อัสวโกวิทกรณ์, "การเขียนโปรแกรมกราฟิกส์", บริษัทดวงสมัยจำกัด, พิมพ์ครั้งที่ 2, 2535
- (2) นุกูล กระจาย, "การเขียนโปรแกรมและประมวลผลข้อมูลด้วยเทอร์โบปาสคาล", บริษัทซีเอ็ดยูเคชั่น จำกัด, 2521
- (3) นุกูล กระจาย, "กราฟิกส์และเกมส์คอมพิวเตอร์ด้วยเทอร์โบปาสคาล", บริษัทซีเอ็ดยูเคชั่น จำกัด, 2521
- (4) บริษัทวิทยุการบิน, "วิทยุเครื่องช่วยการเดินอากาศ", หนังสือวันสื่อสารแห่งชาติ 2527
- (5) ชาญณรงค์ เชื้อเจริญ, "เรดาร์กับการควบคุมจราจรทางอากาศ", หนังสือวันสื่อสารแห่งชาติ 2531
- (6) นาวาอากาศโทธีระ จันทรสฤษ การทำอากาศยานแห่งประเทศไทย, "การควบคุมจราจรทางอากาศที่ทำอากาศยาน", หนังสือวันสื่อสารแห่งชาติ 2528,

กิติกรรมประกาศ

การทำปริญญานิพนธ์ เรื่อง Air Traffic Approach Control Simulator สามารถสำเร็จลุล่วงได้ โดยได้รับความร่วมมือ และความอนุเคราะห์จากหลายฝ่ายด้วยกัน ดังต่อไปนี้

- 1.อ.สมยศ จุณณะปิยะ ภาควิชาโทรคมนาคมที่ให้คำแนะนำที่ดีมาโดยตลอด
- 2.คุณชัชวาล ถาวร โรงเรียนช่างฝีมือทหาร กรมยุทธศึกษาทหาร กองบัญชาการทหารสูงสุด
- 3.คุณเมธี เสรีรุโณ และ พี่ๆ ที่กองเรดาร์ ท่าอากาศยานดอนเมือง และ ห้องควบคุม Radar Approach Control บริษัท วิทยุการบินแห่งประเทศไทย จำกัด ที่ต้อนรับด้วยความอบอุ่น และอนุเคราะห์ข้อมูลในการทำโครงการ
4. สถาบันการบินพลเรือน
- 5.คณะอาจารย์ ภาควิชาโทรคมนาคม สจล. ที่ช่วยชี้แนะแนวทาง, พี่สมเกียรติ ห้องคอม, กำลังใจอันยิ่งใหญ่จากพ่อกับแม่, เพื่อนๆ และ น้องๆ ที่คอยช่วยเหลือเสมอมา
จึงขอขอบคุณมา ณ ที่นี้

ผู้จัดทำ