CONTRIBUTION

ใบรับรองโครงการทางวิศวกรรม

สำนักวิชาเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีสุรนารี

เรื่อง โปรแกรมออกแบบจานรับสัญญาณดาวเทียม

โดย นายจิรพัฒน์ กิตติเมธี

นางสาวศศิวิมล วงศ์สุนทร

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม

_____คณบดี

วันที่ เดือน พ.ศ.2540

คณะกรรมการสอบ

_____ประธานกรรมการ
(ผศ. สุยุชน์ สัตยประกอบ)

_____กรรมการ
(อ. พิชโยทัย มหัทธนาภิวัฒน์)

_____กรรมการ
(อ. มานพ รุจิภากร)

_____กรรมการ
(อ. รังสรรค์ วงศ์สรรค์)

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรมออกแบบจานรับสัญญาณดาวเทียม
(DISH ANTENNA DESIGN PROGRAM)

โดย

นายจิรพัฒน์ กิตติเมธี รหัส B3600702

นางสาวศศิวิมล วงศ์สุนทร รหัส B3602539

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม สำนักวิชาเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา พ.ศ. 2539

หัวข้อ	โปรแกรมออกแบบจานรับสัญญาณดาวเทียม			
	DISH ANTENNA DESIGN PROGRAM			
โดย	นายจิรพัฒน์	กิตติเมธี	รหัส	B3600702
	นางสาวศศิวิมล	วงศ์สุนทร	รหัส	B3602539
สาขาวิชา	วิศวกรรมโทรคมนาคม			
สำนักวิชา	เทคโนโลยีอุตสาหกรรม			
ปีการศึกษา	2539			
อาจารย์ที่ปรึกษา	อ. รังสรรค์ วงศ์สรรค์			

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาโปรแกรมคอมพิวเตอร์โดยใช้ Microsoft Visual Basic4.0 เพื่อช่วยในการออกแบบจานรับสัญญาณดาวเทียม ซึ่งทั้งนี้ได้ศึกษารายละเอียดของจานรับสัญญาณดาวเทียมแบบต่างๆ ในด้านลักษณะ รูปทรง การใช้งาน คุณสมบัติต่างๆ เพื่อนำมาใช้เป็นข้อมูลในการออกแบบด้วยโปรแกรมคอมพิวเตอร์ซึ่งสามารถออกแบบจานรับสัญญาณดาวเทียมได้ 2 ชนิด คือ จานรับสัญญาณดาวเทียมแบบ Cassegrain และจานรับสัญญาณดาวเทียมแบบ Offset-fed

การออกแบบจานรับสัญญาณดาวเทียม จะต้องทราบข้อมูลที่จะนำมาใช้คำนวณเกี่ยวกับส่วนประกอบของจาน ได้แก่ ความถี่ที่ต้องการนำมาใช้งาน ขนาดของจาน อัตราส่วนของ F/D ขั้นตอนการออกแบบจะได้ระยะส่วนประกอบต่างๆของจาน เช่น ความยาวโฟกัส แต่จะต้องคำนึงขนาดของบีมวิดธ์ที่เหมาะสม เพราะบีมวิดธ์จะช่วยให้มีทิศทางการรับส่งสัญญาณที่ดี มีสัญญาณรบกวนน้อย การเปลี่ยนค่าบีมวิดธ์จะทำให้ส่วนประกอบอื่นๆ ของจานเปลี่ยนแปลงตามไปด้วย การนำโปรแกรมคอมพิวเตอร์มาช่วยออกแบบในขั้นตอนต่างๆ จะทำให้การออกแบบจานรับสัญญาณดาวเทียม และส่วนประกอบของจานเป็นไปอย่างสะดวกและรวดเร็ว

กิตติกรรมประกาศ

คณะผู้จัดทำ นายจิรพัฒน์ กิตติเมธี และนางสาวศศิวิมล วงศ์สุนทร นักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยสุรนารี ได้ทำโครงงานโทรคมนาคม โดยเริ่มปฏิบัติงานตั้งแต่วันที่ 24 กันยายน 2539 และสิ้นสุดการปฏิบัติงานในวันที่ 20 ธันวาคม 2539 ซึ่งตลอดระยะการปฏิบัติงานที่ผ่านมา คณะผู้จัดทำได้รับความช่วยเหลือทั้งด้านวิชาการ รวมถึงข้อมูลต่างๆ ที่บรรจุอยู่ในรายงานเล่มนี้

คณะผู้จัดทำขอขอบคุณ อาจารย์รังสรรค์ วงศ์สรรค์ อาจารย์ประจำสาขาวิศวกรรมโทรคมนาคม สำนักวิชาเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้คำแนะนำในการออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain และ Offset-fed รวมทั้งส่วนประกอบของฟีดฮอร์น และ ท่อนำคลื่น ซึ่งทำให้ "โปรแกรมออกแบบจานรับสัญญาณดาวเทียม" เสร็จสุล่วงไปได้ด้วยดี

คณะผู้จัดทำขอขอบคุณเพื่อนร่วมสาขาวิศวกรรมโทรคมนาคมทุกคนที่ห่วงใยและให้กำลังใจ ไต่ถาม ถึงความคืบหน้าของผลการปฏิบัติงาน

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ค
กิตติกรรมประกาศ	ง
สารบัญรูป	ช
สารบัญตาราง	ฌ
บทที่ 1 บทนำ	1
1.1 แนวคิดเรื่องจานรับสัญญาณดาวเทียม	1
1.2 วัตถุประสงค์การใช้โปรแกรมคอมพิวเตอร์ในการออกแบบจานรับสัญญาณดาวเทียม	1
1.3 ขอบเขตการทำงาน	1
1.4 ขั้นตอนการทำงาน	1
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับจานสายอากาศ	3
2.1 จานสายอากาศ	3
2.2 ประเภทของสายอากาศ	3
2.3 สายอากาศแบบ Focal Point	3
2.4 สายอากาศแบบ Cassegrain	4
บทที่ 3 Parabolic Reflector	7
3.1 รูปทรงเรขาคณิตของจานสะท้อนพาราโบลิก	7
3.1.1 การหาความยาวโฟกัส	8
3.1.2 การหาขนาดความลึก	8
3.1.3 การหาค่าทวีกำลัง	9
3.1.4 การหาขนาดของบีมวิดธ์	9
3.2 อัตราการขยายของจานรับสัญญาณ	10

บทที่ 4 จานสายอากาศแบบ Cassegrain 13
4.1 การทำงานของจานสายอากาศแบบ Cassegrain 13
บทที่ 5 จานสายอากาศแบบ Offset-fed 18
5.1 จานรับสัญญาณดาวเทียมแบบ Offset-fed 18
บทที่ 6 สายอากาศแบบ ฮอร์น (Horn) 20
6.1 ฮอร์นชนิดสี่เหลี่ยม (Rectangular Horn) 21
6.2 บีมวิดธ์ของฮอร์นชนิดสี่เหลี่ยม 23
6.3 ลักษณะฮอร์นชนิดสี่เหลี่ยม ที่เหมาะสม (Optimum Rectangular Horn) 26
บทที่ 7 สายอากาศแบบ ฮอร์น ชนิดกลม (Circular-Aperture Horn) 29
7.1 บีมวิดธ์ของ สายอากาศแบบ ฮอร์น ชนิดกลม 31
7.2 ค่าทวีกำลังของ สายอากาศแบบ ฮอร์น ชนิดกลม 33
บทที่ 8 การออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain 34
บทที่ 9 การออกแบบจานรับสัญญาณดาวเทียมแบบ Offset-fed 41
บทที่ 10 สรุปผลการดำเนินงาน 44
10.1 การออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain 44
10.2 การออกแบบจานรบสัญญาณดาวเทียมแบบ Offset-fed 54
10.3 วิธีการนำแบบไปใช้งาน 59
บรรณานุกรม 60
ภาคผนวก 61
ประวัติผู้เขียน 126

สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงลักษณะจานสายอากาศแบบพาราโบลอยด์แบบ Focal Point	3
รูปที่ 2.2 แสดงลักษณะจานสายอากาศแบบพาราโบลอยด์แบบ Cassegrain	4
รูปที่ 2.3 แสดง Beam Waveguide Feed ของสายอากาศแบบ Cassegrain	5
รูปที่ 3.1 แสดง Curve of the parabola equation for two values of focal length	7
รูปที่ 3.2 แสดง Spherical wave is converted to a plana wave by the parabolic reflector	8
รูปที่ 3.3 แสดง 2 ตัวอย่างของ radiating source using a parabolic reflector และผลของ spill over	9
รูปที่ 4.1 แสดงลักษณะจานสายอากาศแบบ Cassegrain	13
รูปที่ 4.2 แสดงลักษณะทางเรขาคณิตของจานสายอากาศแบบ Cassegrain	15
รูปที่ 5.1 แสดงลักษณะจานสายอากาศแบบ Offset-fed	19
รูปที่ 6.1 แสดงรูปแบบทางเรขาคณิตของ ฮอร์น	20
รูปที่ 6.2 แสดงรูปแบบทางเรขาคณิตRectangular-horn	21
รูปที่ 6.3 แสดง E-plane universal pattern of a rectangular horn, TE_{10} mode	25
รูปที่ 6.4 แสดง H-plane universal pattern of a rectangular horn, TE_{10} mode	26
รูปที7 .1 แสดงค่าตัวแปรที่เหมาะสมในการ Integral รูปแบบการแผ่กระจายคลื่นใน E-plane	30
รูปที่ 7.2 แสดงค่าตัวแปรที่เหมาะสมในการ Integral รูปแบบการแผ่กระจายคลื่นใน H-plane	30
รูปที8 .1 แสดงระยะต่างๆบนจานสายอากาศแบบ Cassegrain	40
รูปที่ 9.2 แสดงระยะต่างๆของฮอร์นรูปสี่เหลี่ยม	43
รูปที่10.1 แสดงการเลือกชนิดการออกแบบจานรับสัญญาณดาวเทียม	44
รูปที่10.2 แสดงการเลือกชนิดการออกแบบจานรับสัญญาณดาวเทียมแบบCassegrain	45
รูปที่10.3 แสดงการป้อนข้อมูลที่ใช้ในการออกแบบ	46
รูปที่10.4 แสดงRectangular feedhorn	47
รูปที่10.5 แสดง Rectangular waveguide	47
รูปที่10.6 แสดงระยะของ sub-reflector บนจานรับสัญญาณดาวเทียมแบบ Cassegrain	48
รูปที่10.7 แสดงภาพตัดขวางของจานรับสัญญาณดาวเทียมแบบ Cassegrain	48
รูปที่10.8 แสดงการแก้ไขขนาดของบีมวิดธ์ของจานรับสัญญาณดาวเทียมแบบ Cassegrain	50
รูปที่10.9 แสดงRectangular feedhorn	50
รูปที่10.10 แสดง Rectangular waveguide	51

หน้า

รูปที่ 10.11 แสดงระยะของ sub-reflector บนจานรับสัญญาณดาวเทียมแบบ Cassegrain 51
รูปที่ 10.12 แสดงภาพตัดขวางของจานรับสัญญาณดาวเทียมแบบ Cassegrain 53
รูปที่ 10.13 แสดงการป้อนข้อมูลในการออกแบบจานรับสัญญาณดาวเทียมแบบ Offset-fed 54
รูปที่10.14 แสดงภาพตัดของจานรับสัญญาณดาวเทียมแบบ Offset-fed 55
รูปที่10.15 แสดงการแก้ไขขนาดของบีมวิดธ์ของจานรับสัญญาณดาวเทียมแบบ Offset-fed 57
รูปที่10.16 แสดงภาพตัดของจานรับสัญญาณดาวเทียมแบบ Offset-fed 57

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงการเปรียบเทียบอัตราการขยายของจานรับสัญญาณในย่าน Ku-bandและ C-band 11

ตารางที่ 4.1 แสดงรูปแบบของจานสายอากาศแบบ Cassegrain และ Gregorian ในขนาด ต่างๆกัน โดยที่ขนาดของจานสายอากาศหลักคงที่ 16

ตารางที่ 6.1 แสดง Rectangular-Horn H-Plane Beamwidth Points, TE_{10} Mode 24

ตารางที่ 6.2 แสดง Rectangular-Horn E-plane Beamwidths: TE_{10}Mode 24

ตารางที่ 7.1 แสดง Circular-Horn Beamwidths, TE_{11} Mode 32

ตารางที่ 7.2 แสดง Phase Center Axial Location of a Circular-Waveguide Horn(TE_{11}Mode) behind the Aperture as a Ratio of the Slant Radius 33

บทที่ 1
บทนำ

1.1 แนวคิดเรื่องจานรับสัญญาณดาวเทียม

ปัจจุบันมีการใช้ดาวเทียมในการสื่อสารรูปแบบต่างๆมากมาย ทั้งทางด้านโทรทัศน์ วิทยุ โทรศัพท์ โดยดาวเทียมจะส่งสัญญาณส่งมาในย่าน C-band มีความถี่ประมาณ 4 GHz กำลังส่งประมาณ 8-16 Watt, Ku-band ส่งสัญญาณด้วยกำลัง 20-50 Watt ดังนั้นจำเป็นต้องมีสถานีภาคพื้นดินที่จะทำการรับสัญญาณและขยายขนาดของสัญญาณให้มีความแรงขึ้น สายอากาศแบบจานพาราโบลิกจึงถูกออกแบบเพื่อนำมาใช้งานดังกล่าว ทั้งนี้ขนาดของจานพาราโบลิกจะมีขนาดแตกต่างกันไปตามความแรงของสัญญาณที่ต้องการขยาย จานพาราโบลิกมีความสำคัญในการสะท้อนสัญญาณมารวมอยู่ที่จุดเดียวกัน (Focal point) ซึ่งมีผลทำให้สัญญาณแรงขึ้นได้ จานพาราโบลิกที่ดีจึงจำเป็นต้องมีส่วนโค้งที่ถูกต้อง จึงจะให้ประสิทธิภาพในการสะท้อนที่ดี เพราะส่วนโค้งของจานจะทำหน้าที่คล้ายเลนส์จะบีบสัญญาณให้แคบลงและพุ่งตรงไปข้างหน้าและตรงกับจุดศูนย์กลางของสายอากาศ

1.2 วัตถุประสงค์การใช้โปรแกรมคอมพิวเตอร์ในการออกแบบจานรับสัญญาณดาวเทียม

1.2.1 สามารถออกแบบจานรับสัญญาณได้ 2 ชนิด

1.2.2 สามารถออกแบบ ฟีดฮอร์น และ ท่อนำคลื่นได้ 2 ชนิด

1.2.2 ลดความยุ่งยากในการออกแบบด้วยมือ

1.3 ขอบเขตการทำงาน

1.3.1 ศึกษาการออกแบบจานรับสัญญาณดาวเทียมแบบต่างๆ

1.3.2 ศึกษาการออกแบบฟีดฮอร์น และ ท่อนำคลื่น

1.3.2 เปรียบเทียบลักษณะของจานรับสัญญาณแต่ละชนิด

1.3.3 เขียนโปรแกรมคอมพิวเตอร์เพื่อออกแบบจานรับสัญญาณแต่ละชนิด

1.4 ขั้นตอนการทำงาน

1.4.1 ค้นคว้าข้อมูลเกี่ยวกับจานรับ-ส่งสัญญาณดาวเทียมแบบต่างๆ

1.4.2 ศึกษาวิธีการออกแบบจานรับ-ส่งสัญญาณ

1.4.3 ศึกษาวิธีการเขียนโปรแกรมเพื่อนำมาใช้ในการออกแบบ

1.4.4 ลงมือเขียนโปรแกรม

1.4.5 ปรับปรุงแก้ไขให้เรียบร้อย

1.4.6 ทดสอบผลด้วยการพิมพ์แบบออกมา

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ได้โปรแกรมสำเร็จรูปที่สามารถออกแบบจานรับสัญญาณดาวเทียมได้

1.5.2 สามารถนำโปรแกรมมาใช้งานได้อย่างมีประสิทธิภาพ

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับจานสายอากาศ

2.1 จานสายอากาศ (Dish Antenna)

สายอากาศที่ใช้ในสถานีภาคพื้นดินเป็นอุปกรณ์ที่สำคัญอย่างหนึ่งในส่วนของอุปกรณ์ความถี่ RF เพราะว่าสายอากาศจะทำหน้าที่แพร่กระจายคลื่น RF ที่ถูกมอดูเลตแล้วไปยังดาวเทียมที่ความถี่ขาขึ้น และทำหน้าที่รับคลื่น RF จากดาวเทียมที่ความถี่ขาลง สายอากาศที่สถานีภาคพื้นดินต้องมีคุณสมบัติพื้นฐาน 3 ข้อคือ

2.1.1 ต้องมีทิศทางสูง (Highly Directive Gain) คือ มีบีมวิดธ์ (Beamwidth)ที่แคบทั้งในการรับและส่ง นอกจากนั้นแล้วต้องมี Sidelobe น้อยเพื่อลดการรบกวนกันกับสัญญาณที่ไม่ต้องการ

2.1.2 ต้องมีค่า Noise Temperature ต่ำ เพื่อสัญญาณที่รับจะได้ค่า S/N สูง

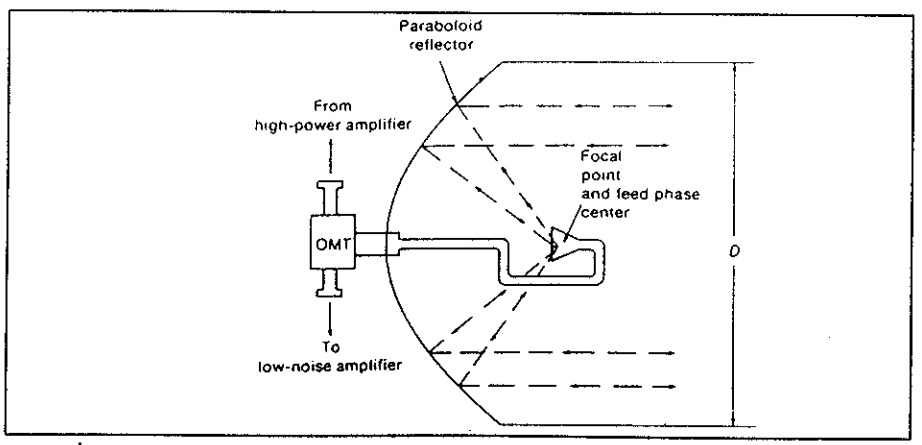
2.1.3 ต้องมีความเที่ยงตรงสูงต่อสัญญาณที่ได้รับ คือ ใช้ระบบ Tracking System ที่มี การควบคุมการเลื่อนสายอากาศไปตามตำแหน่งที่มีสัญญาณแรงที่สุด

2.2 ประเภทของสายอากาศ

สายอากาศที่เหมาะสมและเป็นที่นิยมมากที่สุดในระบบสถานีภาคพื้นดิน คือ สายอากาศแบบพาราโบลอยด์ โดยมีการแบ่งตามลักษณะการป้อนสัญญาณให้แก่แผ่นสะท้อนได้เป็นสองแบบ คือ แบบ Focal Point Feed และแบบ Cassegrain

2.3 สายอากาศแบบพาราโบลอยด์ที่มีการป้อนสัญญาณแบบ Focal Point

สายอากาศแบบนี้ ประกอบด้วยจานสะท้อนซึ่งเป็นพื้นผิวโค้งรูปพาราโบลารอบแกนทิศทางของการรับและ ฟีดเดอร์ ซึ่งมีจุดศูนย์กลางเฟสที่จุดโฟกัสของตัวสะท้อนดังในรูปที่ 2.1



รูปที่ 2.1 แสดงลักษณะจานสายอากาศแบบพาราโบลอยด์แบบ Focal Point

การที่เลือกจานสะท้อนเป็นแบบพาราโบลอยด์นั้น เนื่องจากรูปแบบพาราโบลามีข้อดีสำหรับสายอากาศระดับไมโครเวฟด้วยเหตุผล 2 ประการคือ

2.3.1 ลำคลื่นใดๆจากจุดโฟกัสจะถูกสะท้อนที่แผ่นสะท้อนพาราโบลอยด์และการสะท้อนออกมาขนานกับแกนของพาราโบลา

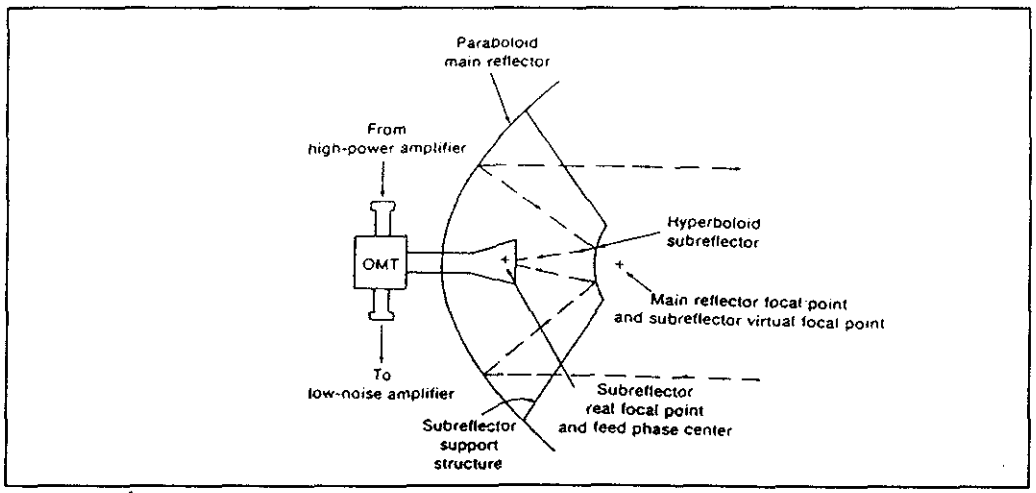
2.3.2 ระยะทางการเดินทางของคลื่นจากจุดโฟกัสไปยังแผ่นพาราโบลอยด์และจากแผ่นพาราโบลอยด์ไปยังระนาบตั้งฉากกับพาราโบลานั้นไม่ขึ้นกับเส้นทาง (คือขึ้นกับจุด 2 จุดเท่านั้น)

ดังนั้นแหล่งพลังงานแบบจุดที่อยู่ที่จุดโฟกัสจะถูกเปลี่ยนเป็นคลื่นระนาบที่มีเฟสเดียวกันในทิศทางที่ต้องการส่ง โดยใช้แผ่นสะท้อนพาราโบลอยด์ ดังรูปที่ 2.1

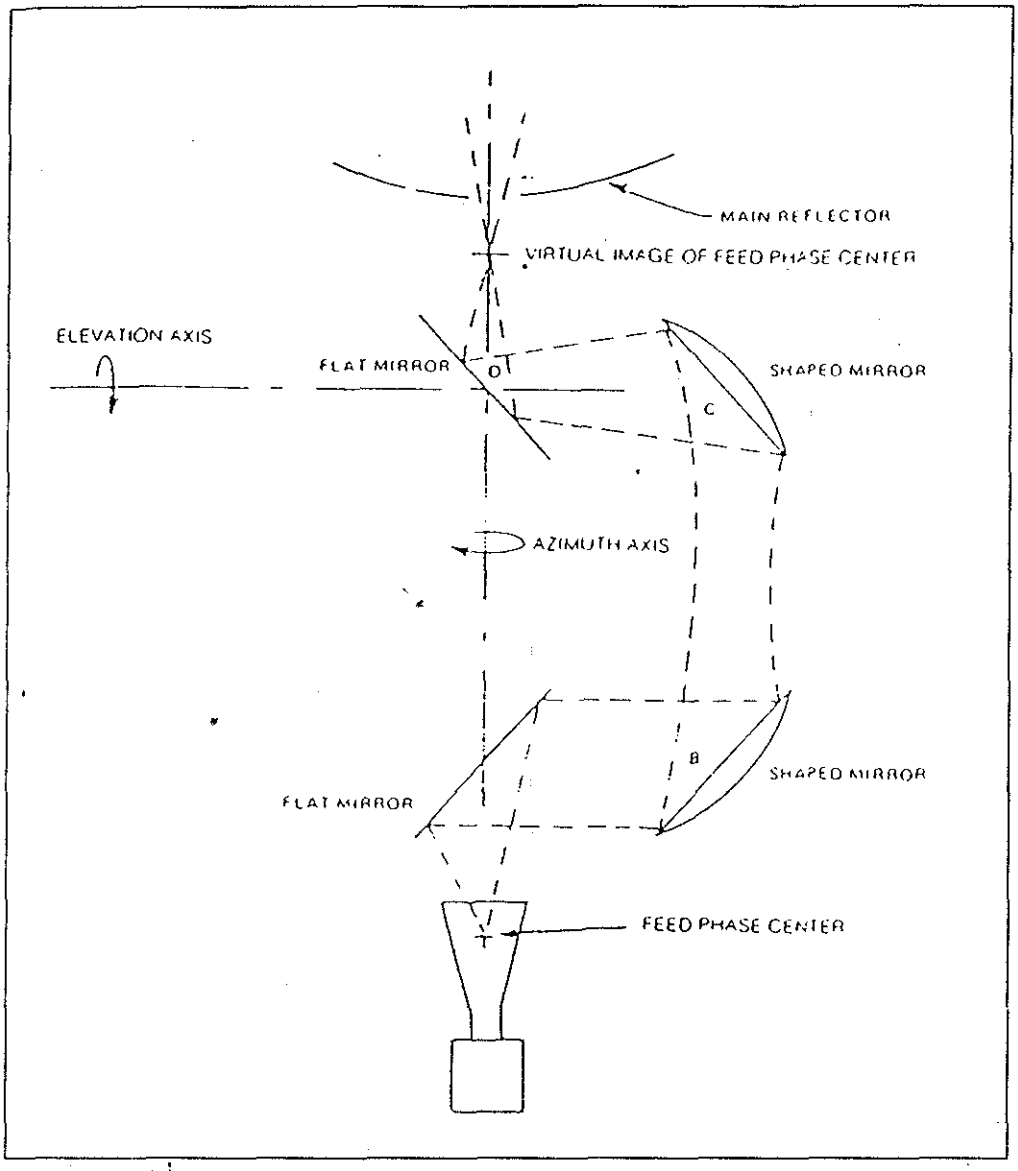
ขนาดของสายอากาศแบบนี้แสดงได้โดยเส้นผ่านศูนย์กลาง D ของจานสะท้อน Feeder เชื่อมต่อกับภาคขยาย HPA และภาคขยาย LNA ผ่านตัว OMT (Orthogonal Mode Transducer) ซึ่งเป็นอุปกรณ์ 3 พอร์ต (Port) โดยมีอัตราส่วนการแยกสัญญาณส่งและสัญญาณรับดีกว่า 40 dB ในส่วนของภาคส่งนั้น พลังงานของคลื่นพาห์ที่จะส่งจากภาคขยาย HPA จะกระจายไปเป็นเส้นแนวเดียวกับทิศทางการส่งที่ต้องการโดยแพทเทอร์น(Pattern)จะมีบีมวิดธ์(Beamwidth)ที่แคบ ในขณะที่ส่วนของภาครับนั้นจะดักจับสัญญาณในทิศทางที่ต้องการรับ ให้สัญญาณกระทบแผ่นสะท้อนแล้วสะท้อนมารวมกันที่จุดโฟกัสซึ่งจะผ่านไปยังภาคขยาย LNA ผ่าน Feeder ต่อไป รูปร่างของบีมที่กระจายออกมาถูกกำหนดได้โดยการใช้บางส่วนของรูปพาราโบลอยด์

2.4 สายอากาศแบบ Cassegrain

สายอากาศแบบนี้จะมีแผ่นสะท้อน 2 แผ่น คือ แผ่นสะท้อนหลักและแผ่นสะท้อนรอง โดยแผ่นสะท้อนหลักจะเป็นรูปพาราโบลา ส่วนแผ่นสะท้อนรองเป็นรูปไฮเปอร์โบลา จุดโฟกัสของแผ่นสะท้อนหลักจะเป็นจุดเดียวกับจุดโฟกัสเสมือนของแผ่นสะท้อนรอง และจะมี Feeder ที่มีจุดศูนย์กลางเฟสอยู่ที่จุดโฟกัสของแผ่นสะท้อนรอง ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงลักษณะจานสายอากาศแบบพาราโบลอยด์แบบ Cassegrain



รูปที่2.3 แสดง Beam Waveguide Feed ของสายอากาศแบบ Cassegrain

ในส่วนของภาคส่งนั้น สัญญาณจากภาคขยาย HPA จะแผ่กระจายมาที่จุดโฟกัสของแผ่นสะท้อนรองโดย ฟีดเดอร์ จะกระทบแผ่นสะท้อนรองที่ด้านนอกของส่วนโค้ง หลังจากนั้นจะสะท้อนไปยังจุดโฟกัสของแผ่นสะท้อนหลัก โดยเสมือนมีจุดศูนย์กลางเฟสอยู่ที่จุดโฟกัสเสมือน (หรือจุดโฟกัสจริงจากแผ่นสะท้อนหลัก) ของแผ่นสะท้อนรอง ทำให้คลื่นที่สะท้อนออกมาจากแผ่นสะท้อนหลักมีหน้าคลื่นที่มีเฟสเดียวกัน และเป็นคลื่นระนาบที่มีทิศทาง ในแง่ของส่วนรับนั้น สัญญาณจากจานดาวเทียมจะตกกระทบกับแผ่นสะท้อนหลักและจะถูกสะท้อนให้ไปรวมกันที่จุดโฟกัสของแผ่นสะท้อนหลัก ต่อมาจะถูกสะท้อนออกมาโดยแผ่นสะท้อนรอง และสัญญาณจะมารวมกันที่จุดศูนย์กลางเฟสของฟีดเดอร์ จากนั้นก็จะส่งผ่านไปยังภาคขยาย LNA ต่อไป โดยมี OMT เป็นตัวแยกสัญญาณรับและส่ง

โดยทั่วไปแล้วสายอากาศแบบ Cassegrain มีราคาแพงกว่าพาราโบลอยด์แบบธรรมดามาก เพราะว่ามีการเพิ่มส่วนของแผ่นสะท้อนรองขึ้นมา อย่างไรก็ตามสายอากาศแบบ Cassegrain นั้นมีข้อดีกว่าแบบพาราโบลอยด์ธรรมดาหลายประการ เช่นมีค่า Noise Temperature น้อยกว่า มีทิศทางที่เที่ยงตรงกว่า และมีความยืดหยุ่นในการออกแบบฟีดเดอร์ มากกว่า เนื่องจากว่าพลังงานที่หลุดออกจากฟีดเดอร์นั้น ส่วนใหญ่จะหลุดไปจากอากาศซึ่งมีค่า Noise Temperature น้อยกว่า 30K ดังนั้นค่า S/N ของสัญญาณจึงมีค่ามากกว่าแบบพาราโบลอยด์ธรรมดา เนื่องจากฟีดเดอร์อยู่ในตำแหน่งใกล้กับจุดยอดของส่วนโค้งของแผ่นสะท้อนหลัก ทำให้เสถียรภาพทางกลดีกว่าแบบพาราโบลอยด์ธรรมดาที่มีตัวป้อนลอยอยู่เหนือแผ่นสะท้อน จากการที่มีเสถียรภาพทางกลดีกว่านั้น ทำให้ความแน่นอนในทิศทางของสายอากาศชนิดนี้มีมากกว่าเพื่อที่จะลดการสูญเสียของสายนำสัญญาณที่เชื่อมระหว่างภาคขยาย HPA หรือภาคขยาย LNA กับ ฟีดเดอร์ เราจึงใช้ระบบป้อนสัญญาณท่อนำคลื่นแบบบีม (Beam Waveguide Feed System)

ท่อนำคลื่นแบบบีมนั้นประกอบด้วยประกอบด้วยกระจกสี่แผ่นยึดติดด้วยโลหะซึ่งตั้งอยู่อย่างเที่ยงตรงกับแผ่นสะท้อนรอง ฟีดเดอร์ แกนของมุม Azimuth (มุมที่ทิศทางของสายอากาศในระนาบบนพื้นดินทำมุมกับทิศเหนือ) กระจกเหล่านี้เปรียบเสมือนปล่อยพลังงานระหว่างตัวป้อนและแผ่นสะท้อนรอง เราจะต้องออกแบบให้มีการสูญเสียน้อยที่สุดเมื่อตัวป้อนถูกยึดกับฐานคอนกรีตที่พื้น ตัวโลหะที่ยึดกระจกทำหน้าที่เป็นตัวป้องกันสัญญาณรบกวนจากพื้นดิน และเพิ่มความมั่นคงให้แก่กระจกอันเนื่องมาจากลม อุณหภูมิหรือภาระโหลดอื่นๆ ส่วนล่างของโลหะนั้น ถูกยึดติดไว้โดยแท่นและสามารถหมุนรอบแกนของมุม Azimuth ได้ ส่วนบนของโลหะถูกยึด โดยแผ่นสะท้อนหลักและสามารถหมุนได้รอบแกนของมุม Elevation ท่อนำคลื่นแบบบีมนั้นจะนำหรือรับพลังงานจากหรือไปยังฟีดเดอร์ และแผ่นสะท้อนโดยใช้หลักการสะท้อนของแสง ถ้ากระจกเหล่านี้มีมีขนาดพอเหมาะและวางในตำแหน่งที่เหมาะสมแล้ว พลังงานจากตัวป้อนจะมีจุดศูนย์กลางเฟสอยู่ที่จุดโฟกัสของแผ่นสะท้อนรอง การทำงานจะเป็นดังนี้ กระจก A, B, C และ D จะเคลื่อนที่พร้อมกันเมื่อฐานของแกนของมุม Azimuth (Azimuth Platform) หมุน ส่วนกระจก D อยู่บนแกนของมุม Elevation ก็จะหมุนไปอีกแกนหนึ่งด้วยเมื่อแผ่นสะท้อนหลัก ถูกยกขึ้นขณะที่จานเงยขึ้น โดยวิธีการนี้ พลังงานที่ไปหรือจากท่อนำคลื่นจะมีทิศทางไปยังแผ่นสะท้อนหลักเสมอ

ถ้าจานสายอากาศหลักที่ติดตั้งไม่มีเสถียรภาพทางกลแล้ว ก็จะทำให้ทิศทางการรับหรือส่งคลื่นผิดไป เราสามารถควบคุมตำแหน่งของสายอากาศโดยใช้ระบบขับเคลื่อนทางเครื่องกล (Servomechanism) เพื่อควบคุมมุมของสายอากาศโดยการชดเชยผลของการเปลี่ยนแปลงตำแหน่งของสายอากาศที่คลาดเคลื่อนไป อันเนื่องมาจากสายอากาศอาจติดตั้งอยู่บนพาหนะหรือสิ่งเคลื่อนไหวใดๆก็ได้

บทที่3

Parabolic Reflectors

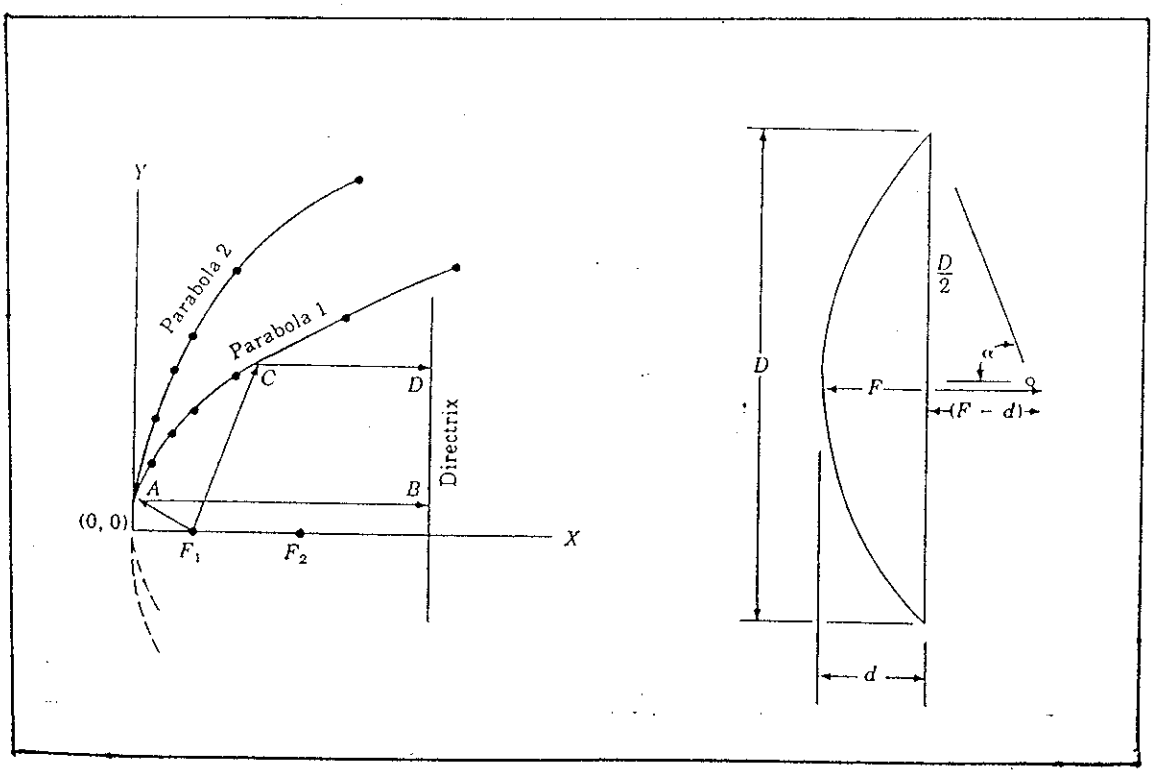
3.1 ลักษณะทั่วไปของจานสายอากาศแบบพาราโบลิก

รูปทรงเรขาคณิตของจานสะท้อนพาราโบลิก สามารถแสดงได้โดยสมการดังนี้

$$\pm Y = \sqrt{4FX} \qquad (3\text{-}1)$$

ซึ่งเป็นผลให้เกิดภาพ 3.1 เมื่อค่าของความยาวโฟกัส (F) ถูกกำหนดขึ้นตามความเหมาะสม และค่า Y หาได้จากแต่ละค่าของ X โดยแทนลงในสมการ (3-1) ค่าที่แตกต่างกันของความยาวโฟกัส (F) จะเป็นสาเหตุให้ความโค้งของจานพาราโบลิกเปิดหรือปิดได้ ดังในรูปที่ (3.1) แต่ค่าความโค้งโดยตลอดจะเป็นรูปทรงพื้นฐานที่เหมือนกันจึงใช้สูตรคำนวณความโค้งเดียวกัน จะเห็นได้ว่าถ้าความยาวโฟกัสมีค่ามากๆจะทำให้เดิร์ฟเปิด และค่า Y ก็สามารถเป็นค่าลบได้

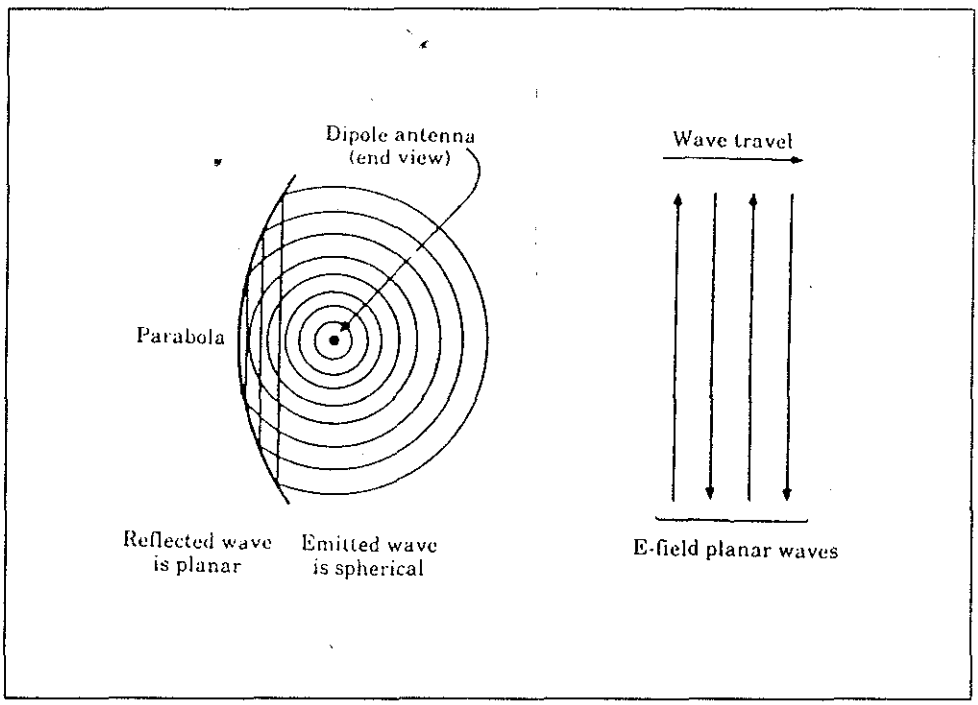
ทุกๆเส้นความยาวที่วัดจากจุดโฟกัสแล้วไปตกกระทบกับผิวด้านในของความโค้งแล้วสะท้อนออกมาขนานกับแกน X แล้วกระทบกับเส้นที่ตั้งฉากกับแกน X ที่ตำแหน่งใดๆโดยเรียกว่า Directrix ระยะทางที่คลื่นเดินทางสะท้อนมากระทบกับ directrix นั้นคลื่นแต่ละลำจะใช้ระยะทางเท่ากัน ในรูปที่ (3.1) ความยาวของเส้นทาง F_1AB จะเท่ากับเส้นทาง F_1CD



รูปที่ 3.1 Curve of the parabola equation for two values of focal length.

จานสะท้อนพาราโบลิกจะถูกสร้างขึ้นโดยการหมุนรูป 3.1 รอบแกน X เพื่อฟอร์มให้เป็นรูปจานพื้นผิวสามมิติ เรียกว่าพาราโบลอยด์ (Paraboloid) ตัวจานสะท้อนนี้จะมีขนาดเท่าไรนั้น ขึ้นอยู่กับขนาดของเส้นผ่าศูนย์กลางของจาน (minimum D = 2λ) และอัตราส่วนของ F/D จะเป็น 0.15-2.5 แต่ที่นิยมมักใช้ F/D=0.5

หน้าคลื่นวงกลมที่ฉายจากแหล่งกำเนิดที่จุดโฟกัส (F) จะกระทบกับผิวด้านในของจานพาราโบลิก และความสัมพันธ์ของลำดับเวลาจะเป็นสาเหตุให้คลื่นที่สะท้อนออกจากจานพาราโบลาเป็นเส้นตรง เช่นเดียวกับหน้าคลื่นในรูปที่ 3.2 เป็น directrix ซึ่งจะเคลื่อนออกจากแผ่นจานสะท้อนไปทางขวามือของรูป



รูปที่ 3.2 Spherical wave is converted to a plana wave by the parabolic reflector.

การแทนค่าเส้นผ่าศูนย์กลางให้เป็นครึ่งหนึ่งคือ D/2 เพื่อหาค่า Y ในสมการ (3-1) และความลึกของจานพาราโบลิก (d) เพื่อหาค่า X นั้นจะสามารถแก้สมการจานพาราโบลิกได้ทุกขนาด

$$\text{ความยาวโฟกัส (Focal length : F)} = \frac{D^2}{16d}$$

$$\text{ความลึก (Depth : d)} = \frac{D^2}{16F}$$

$$\text{เส้นผ่านศูนย์กลาง (Diameter : D)} = 2\sqrt{4dF} = 4\sqrt{dF} \qquad (3\text{-}2)$$

สำหรับค่าของเส้นผ่าศูนย์กลางใดๆ ความลึกของจานและจุดโฟกัสจะแปรตามไปด้วย จานที่ตื้นจะทำให้ได้ความยาวโฟกัสมาก ได้ 3-dB Beamwidth กว้าง และไดเรคติวิตี้เกนต่ำ ส่วนจานที่ลึกนั้นจะทำให้ความยาวโฟกัสสั้น Beamwidth แคบและไดเรคติวิตี้เกนสูง

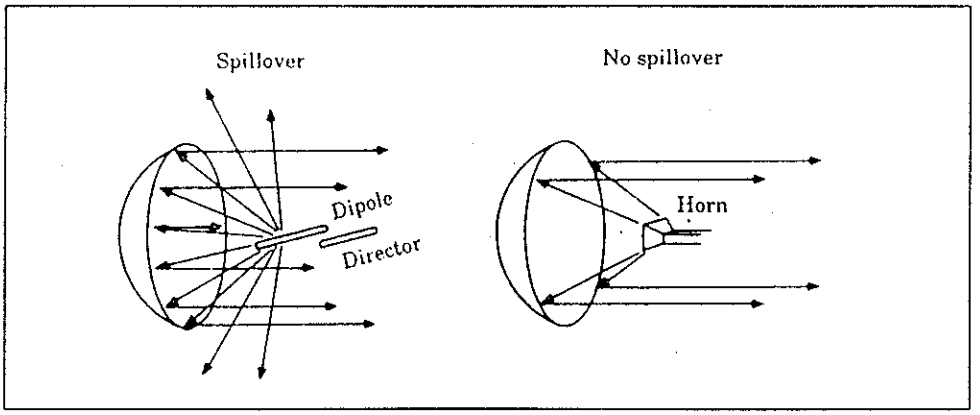
เนื่องจากแหล่งกำเนิดคลื่นถูกวางที่ตำแหน่งจุดโฟกัส เราสามารถคำนวณหา Directivity Gain ในหน่วยเดซิเบลได้ ดังนี้

$$\text{Gain in decibels} = 10 \log\left(2 + \left[\frac{d}{D}\left(\frac{\pi dD}{3\lambda}\right)^2\right]\right) \qquad (3\text{-}3)$$

-3-dB Beamwidth มีความสัมพันธ์ ดังนี้

$$\theta_{-3dB} = 2 \arctan \frac{\lambda F}{d^2 D} \qquad (3\text{-}4)$$

เมื่อเราใช้สายอากาศไดโพลสั้นๆ ซึ่งมีคลื่นแผ่กระจายตั้งฉากรอบๆไดโพล วางไว้ที่จุดโฟกัส คลื่นที่แผ่ออกไปกระทบกับผิวด้านในของจานสะท้อนและสะท้อนกลับออกมาเป็นลำคลื่นขนาน จะมีส่วนที่เราต้องพิจารณาคือ Spillover ดังแสดงในรูป 3.3



รูปที่ 3.3 Two examples of a radiating source using a parabolic reflector and the resulting spillover.

แต่เราสามารถลดเหตุการณ์ Spillover ได้ด้วยการเพิ่ม parasitic director ความยาวคลื่น 0.1 ห่างจากตัวขับสัญญาณ (driven element) โดยวางไว้หน้าจุดโฟกัส เนื่องจากสายอากาศไดโพลเป็นตัวขับสัญญาณ ดังนั้นแท่ง parasitic จะถูกใช้เป็นตัวชี้ทิศทาง (director) 1 และจานสะท้อนพาราโบลิกจะกลายเป็นตัวสะท้อน parasitic1 ในที่นี้จานพาราโบลิกไม่ได้เป็นสายอากาศ

แต่เป็นตัวสะท้อนจากจุดโฟกัสหรือเรียกได้ว่าเป็นเลนส์ เมื่อเรานำสายอากาศฮอร์นไปวางไว้ที่จุดโฟกัสโดยให้เป็นเหมือนแหล่งกำเนิดคลื่น คลื่นทุกลำจะพุ่งไปกระทบกับจานพาราโบลิก แต่จะมีคลื่นเพียงเล็กน้อยที่หลุดจากจาน หรือ spillover การนำเอาสายอากาศฮอร์นมาใช้จึงลดปัญหา spillover และช่วยเพิ่มประสิทธิภาพให้ดีขึ้น

ในส่วนของตัวรับสัญญาณก็เช่นเดียวกันเมื่อลำคลื่นขนานพุ่งไปกระทบจานพาราโบลิกซึ่งทำหน้าที่เป็นตัวรับ ก็จะสะท้อนกลับไปยังจุดโฟกัส ซึ่งมีสายอากาศฮอร์นวางอยู่ทำให้สามารถรับสัญญาณได้

3.2 อัตราการขยายของจานรับสัญญาณ

สายอากาศที่มีรูปร่างคล้ายจานจะมีค่าแสดงอัตราต่างๆเหมือนกับสายอากาศแบบอื่นเช่นกัน สิ่งหนึ่งที่จะกล่าวก็คือ ค่าอัตราการขยาย หรือที่มักจะเรียกทับศัพท์ภาษาอังกฤษว่า เกน (Gain) นั่นเอง

อัตราการขยายของสายอากาศก็คือ การวัดความสามารถของสายอากาศที่รับหรือส่งพลังงานออกไปในทิศทางใดทิศทางหนึ่ง สิ่งที่จะกล่าวต่อไปนี้มีผลกระทบต่ออัตราการขยายของจานรับสัญญาณ ซึ่งถือว่าเป็นสายอากาศทั้งสิ้น ได้แก่

1. พื้นที่หน้าตัดของจานรับสัญญาณ(ไม่ใช่พื้นที่ผิวของจานบางตำราใช้คำว่าพื้นที่ช่องเปิด)
2. ช่องเปิดของ LNB หรือขนาดของรีเฟล็กเตอร์ย่อย (Sub-reflector)
3. ผิวของจานรับสัญญาณ
4. รีเฟล็กเตอร์ย่อยไม่อยู่ในตำแหน่งโฟกัสที่ดีที่สุด
5. ความโค้งของผิวจานที่ไม่เป็นไปตามลักษณะพาราโบลิก

หากเกิดสิ่งผิดปกติหรือข้อผิดพลาดจากปัจจัยทั้งห้านี้ จะเป็นสาเหตุให้อัตราการขยายลดลง โดยจานรับสัญญาณแบบพาราโบลิกจะมีค่าสัมประสิทธิ์ของประสิทธิภาพ (η) กำหนดเอาไว้เช่นเดียวกับสายอากาศแบบอื่นเช่นเดียวกันซึ่งค่า η ของสายอากาศแบบพาราโบลิกจะมีประมาณ 60-75% ในการออกแบบจานรับสัญญาณแบบพาราโบลิกนั้น ผู้ออกแบบสามารถคำนวณอัตราการขยายกำลังของจานรับสัญญาณแบบนี้ได้โดยใช้สูตร

$$G = 10 \log [\pi^2 \eta (D/\lambda)] \text{ dB} \qquad (3\text{-}5)$$

ขณะที่ D = เส้นผ่าศูนย์กลางของจานรับสัญญาณ หน่วยเป็นเมตร

λ = ความยาวคลื่น

η = ค่าสัมประสิทธิ์ของประสิทธิภาพ

ตารางที่ 3.1 แสดงการเปรียบเทียบอัตราการขยายของจานรับสัญญาณในย่าน Ku-band และ C-band

Size in ft.	Size in m.	C-Band(3.7~4.2 GHz) Efficiency (%)		Ku-Band (11.7~12.2 GHz) Efficiency (%)	
		55%	70%	55%	70%
1	0.30	19.4 †	20.5	29.0	30.1
2	0.61	25.4	26.5	35.1	36.1
3	0.91	29.0	30.0	38.6	39.6
4	1.22	31.5	32.5	40.7	42.1
5	1.52	33.4	34.4	43.0	44.1
7	2.13	35.0	36.0	44.6	45.6
8	2.43	37.5	38.5	47.1	48.1
9	2.74	38.5	39.5	48.1	49.2
10	3.05	39.4	40.5	49.0	50.1
11	3.35	40.2	41.3	49.9	50.9
12	3.65	41.0	42.0	50.6	51.7
13	3.96	41.7	42.7	51.3	52.4
16	4.87	43.5	44.5	53.1	54.1
20	6.09	45.4	46.5	55.1	56.1
25	7.62	47.4	48.4	57.0	58.0
33	10.05	49.8	50.8	59.4	60.4

† in dB - gain in decibels referenced to an isotropic antenna

ตารางที่ 3.1 เป็นตารางเปรียบเทียบอัตราการขยายของจานรับสัญญาณระหว่างจานที่ใช้รับสัญญาณในย่าน Ku-band และ C-band ซึ่งมีขนาดของเส้นผ่านศูนย์กลางเท่ากัน และมีประสิทธิภาพของจาน 65% เท่ากัน ส่วนโค้งของจานจะทำหน้าที่คล้ายกับเลนส์ คือทำการบีบสัญญาณให้แคบลงแล้วพุ่งตรงไปยังด้านหน้า และตรงกับจุดศูนย์กลางของสายอากาศ (ลักษณะนี้เรียกว่า Boresight) จะทำให้สัญญาณที่รับได้มีความแรงมากที่สุด ขณะเดียวกันก็จะมีการกำจัดสัญญาณอื่นๆ (อาจจะเป็น Noise ก็ได้)ที่มาจากทิศทางอื่นออกไปด้วย ปริมาณของสัญญาณที่ถูกส่งจากจานออกไปนั้นจะเป็นจุดที่ LNB รับได้ดีที่สุด

3dB	=	X 2	เท่าของเพาเวอร์ที่รับได้
10dB	=	X 10	เท่าของเพาเวอร์ที่รับได้
20dB	=	X 100	เท่าของเพาเวอร์ที่รับได้
30dB	=	X 1000	เท่าของเพาเวอร์ที่รับได้
40dB	=	X 10000	เท่าของเพาเวอร์ที่รับได้
50dB	=	X 100000	เท่าของเพาเวอร์ที่รับได้

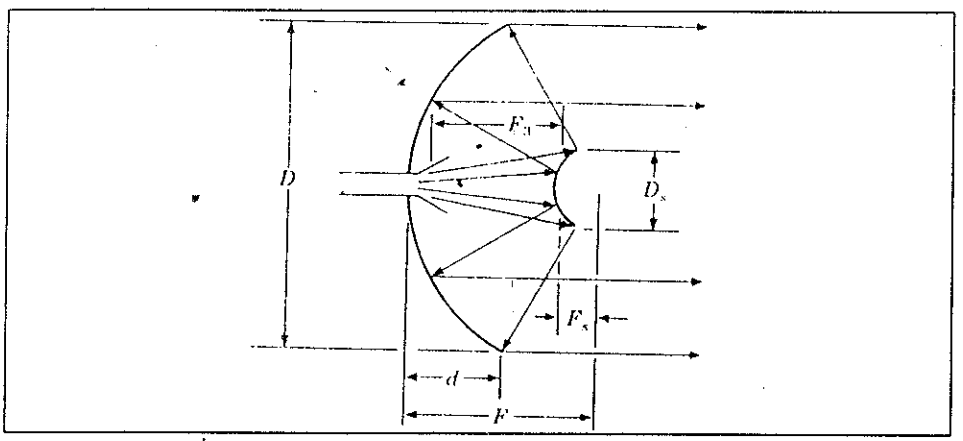
จากตารางที่3.1 จะแสดงถึงอัตราขยายของจานรับสัญญาณที่มีขนาดต่างๆกัน ซึ่งโรงงานผู้ผลิตบางแห่งอาจจะให้อัตราขยายสูงกว่าค่านี้เล็กน้อย โดยการปรับปรุงประสิทธิภาพ (Efficiency) ของจานให้ดีขึ้น และมีการตรวจสอบส่วนโค้งของจานให้มีความแน่นอนยิ่งขึ้น หรือปรับปรุงในเรื่องของเทคนิคการฟีดสัญญาณ

บทที่ 4

ชนิดของจานสายอากาศแบบ Cassegrain

4.1 จานสายอากาศแบบ Cassegrain

การทำงานของจานสายอากาศแบบ Cassegrain สามารถอธิบายได้โดยอ้างถึงรูปที่ 4.1 และสมมติฐานของระบบนี้คือการทำงานใน receiving mode หรือ transmitting mode เพื่อแสดงให้เห็นถึงหลักการ ดังนั้นตัว receiving mode จึงถูกนำมาใช้



รูปที่ 4.1 แสดงลักษณะของจานสายอากาศแบบ Cassegrain

สมมติว่าเรามีพลังงานในรูปของลำคลื่นขนานตกกระทบบนระบบจานสะท้อน พลังงานที่ถูกขวางโดยจานสะท้อนหลักซึ่งมีผิวเว้าเข้าข้างในขนาดใหญ่ จะถูกสะท้อนไปยัง Subreflector พลังงานที่ถูกสะสมโดยผิวโค้งเว้าออกข้างนอกของตัว subreflector นี้จะถูกสะท้อนกลับอีกครั้งโดยมีทิศทางไปยังจุดยอด (Vertex) ของจานสะท้อนหลัก ถ้ามีลำคลื่นขนานตกกระทบบนจานสะท้อนหลักพาราโบลอยด์ และ subreflector แบบไฮเพอร์โบลอยด์แล้ว ลำคลื่นทั้งหมดจะรวมเป็นจุดเดียว ซึ่งเราจะนำตัวรับสัญญาณไปวางไว้ที่ตำแหน่งนี้

ด้วยหลักการเดียวกันนี้เอง เราสามารถนำไปใช้ได้กับ transmitting mode โดยนำฟีดไปวางไว้ที่ตำแหน่งโฟกัสและโดยปกติขนาดของฟีดจะใช้เพียงเล็กๆ ดังนั้นตำแหน่งของ subreflector จะอยู่ในสนามระยะไกล (Fax - field region) ในกรณีนี้ขนาดของ subreflector ต้องใหญ่พอที่จะรับคลื่นที่แพร่ออกจากฟีดได้ทั้งหมด การกำหนดให้จานสายอากาศมีเรขาคณิตแบบพาราโบลอยด์และไฮเพอร์โบลอยด์นั้น คลื่นที่สะท้อนโดยจานสะท้อนหลักจะมีลำคลื่นขนานแอมปลิจูดของคลื่นที่ออกมากจะหาได้จากฟีดแพทเทิร์น (feed pattern) และ tapering effect ของรูปทรงเรขาคณิต

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

เรขาคณิตของระบบ Classical Cassegrain นั้น ใช้ความเว้าของพาราโบลอยด์เป็นจานสะท้อนหลัก และความนูนของไฮเพอร์โบลอยด์เป็น subreflector นี้เป็นรูปแบบง่ายๆ ซึ่งอธิบายได้อย่างสมบูรณ์แบบโดยใช้เพียงสี่ตัวแปร (อย่างละสองสำหรับแต่ละจานสะท้อน)

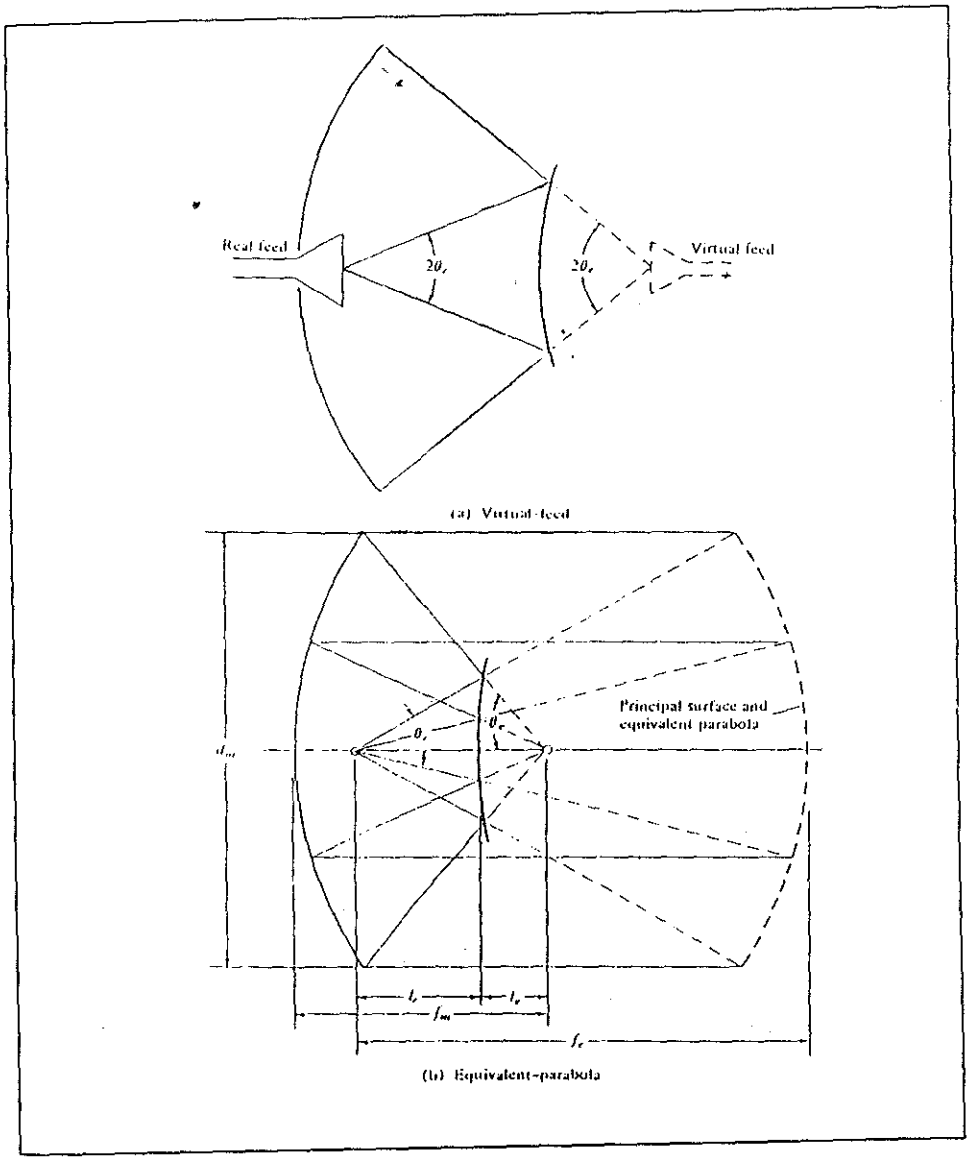
เพื่อให้เกิดความเข้าในได้ดียิ่งขึ้น จะใช้ภาพที่4.2 ในการอธิบายหลักการนี้ ฟีดจริง และ subreflector จะถูกแทนด้วยระบบสมมูลย์ ซึ่งประกอบด้วยฟีดเสมือน (Virtual - feed) นำไปวางไว้ที่จุดโฟกัสของจานสะท้อนหลัก ในภาพแสดงด้วยเส้นประ รูปร่างของ virtual feed สามารถหาได้จากภาพฉายของฟีดจริง (real feed) เทคนิคนี้จะใช้ได้อย่างถูกต้องเมื่อตรวจสอบเอฟเฟคทีฟอะเพอร์เจอร์ (effective aperture) ของฟีด และเมื่อขนาดของฟีดจริงและ virtual feed ใหญ่กว่าความยาวคลื่น ในความเป็นจริงแล้วสำหรับลักษณะ classical cassegrain ของภาพ 4.2(a) นั้น virtual feed มีขนาดของ effective aperture และ ความกว้างของบีมวิดธ์เล็กกว่า real feed การเพิ่มในบีมวิดธ์เป็นผลจากความโค้งเว้าของตัว subreflector และมันสามารถหาได้โดยสมการอัตราส่วนบีมวิดธ์ของ virtual ต่อ real - feed ต่ออัตราส่วนของมุม θ_v / θ_r

ความสามารถในการหา effective aperture ในแบบต่างๆ สำหรับ virtual feed เทียบกับของ real feed จะถูกนำมาใช้ประโยชน์หลายๆ ด้าน เช่น ใน monopulse antenna เพื่อสนับสนุนประสิทธิภาพ และให้ได้ช่องสัญญาณที่กว้างในการทำงานเพื่อนำไปใช้ประโยชน์ของอะเพอร์เจอร์หลัก (main aperture) ระบบนี้จึงต้องการฟีดอะเพอร์เจอร์ขนาดใหญ่ มีความยาวโฟกัสยาวและมีโครงสร้างจานสายอากาศขนาดใหญ่ ขนาดของจานสายอากาศต้องสัมพันธ์กับรูปแบบของระบบ Cassegrain ซึ่งมีฟีดขนาดใหญ่และความยาวโฟกัสของจานสะท้อนหลักสั้น

แม้ว่าแนวความคิดของ virtual feed จะถูกนำมาใช้ประโยชน์ในระบบ cassegrain แต่ก็ยังไม่เหมาะสมนัก ต่อมาจึงได้มีการประยุกต์วิธีการจาก virtual feed ให้กลายมาเป็นแนวความคิดของพาราโบลา สมมูลย์ (Equivalent) การใช้เทคนิค equivalent parabola นี้ตัวจานสะท้อนหลักและ subreflector จะถูกวางตำแหน่งโดยใช้จุดโฟกัสของพื้นผิวสมมูลย์ที่ระยะเหมาะสมจากจุดโฟกัสจริง พื้นผิวสมมูลย์นี้แสดงเป็นเส้นประในรูปที่ 4.2 พื้นฐานทางเรขาคณิตง่ายๆของเส้นคลื่นที่ปรากฏบนพื้นผิวสมมูลย์นั้นสำหรับจาน Cassegrain จะมีรูปทรงเป็นพาราโบลอยด์ ซึ่งมีความยาวโฟกัสเท่ากับระยะทางจากจุดยอด(vertex)ไปยังจุดโฟกัสจริง ระบบสมมูลย์นี้จะลดทอนเป็นจานสะท้อนเดี่ยว (single-reflector) ซึ่งมีฟีดอันเดียวกัน แต่แตกต่างกันที่จานสะท้อนหลัก

จากที่กล่าวมาแล้วใน classical cassegrain นั้นยังมีรูปแบบจานสะท้อนหลักและ subreflector ที่แตกต่างกันออกไปอีก รวมไปถึงความโค้งเว้าและความแบนอีกด้วย ในบางแบบจานสะท้อนหลักจะมีขนาดตายตัวในขณะที่บีมวิดธ์ของมันเพิ่มขึ้นเรื่อยๆ และขนาดมุมของจานสายอากาศลดลง บางแบบบีมวิดธ์จะถูกจัดไว้ตายตัวในขณะที่จานสะท้อนหลักกลายเป็นตัวเปลี่ยนแปลงให้แบนขึ้นและขนาดของมุมกลับลดลง

ในตารางที่ 4.1 แสดงรูปแบบของจานสายอากาศแบบ Cassegrain และ Gregorian ในขนาดต่างๆกันโดยที่ขนาดของจานสายอากาศหลักคงที่ ในห้าภาพแรกจะเป็นรูปแบบของจานสายอากาศแบบ Cassegrain ส่วนสองภาพหลังเป็นแบบ Gregorian จำนวนช่วงของพารามิเตอร์ตามลักษณะที่สังเกตได้จะถูกแสดงไว้ในแต่ละภาพแล้ว จานสะท้อนหลักในภาพที่สี่ มีลักษณะแบนราบส่วน subreflector เป็นพาราโบลิกและสามารถถูกแบ่งโดยระยะทาง ซึ่งใช้การประมาณเส้นคลื่น สำหรับภาพที่ห้านั้น subreflector มีลักษณะเป็น elliptical และจานสะท้อนหลักเป็นรูปทรงพาราโบลิก ด้วยรูปแบบนี้เองที่ทำให้ได้ขนาดใหญ่กว่าที่ผ่านมา

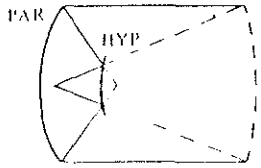
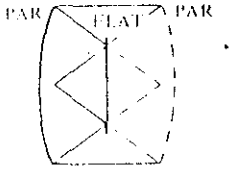
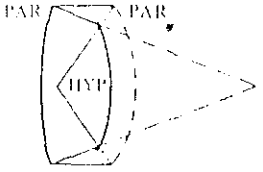
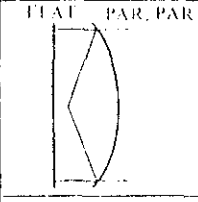
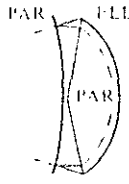
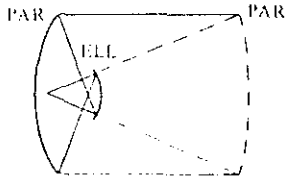
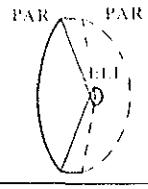


รูปที่ 4.2 แสดงลักษณะทางเรขาคณิตของจานสายอากาศแบบ Cassegrain

(a) Virtual Feed

(b) Equivalent-parabola

ตารางที่ 4.1 แสดงรูปแบบของจานสายอากาศแบบ Cassegrain และ Gregorian ในขนาด ต่างๆกัน โดยที่ขนาดของจานสายอากาศหลักคงที่

	Illustration	θ_e/θ_r and f_e/f_m	f_m and f_e	e
Cassegrain reflector forms		>1	>0	>1
	Convex subdish (classical form)			
		1	>0	∞
	Flat subdish			
		<1 >0	>0	<−1
	Concave subdish			
		0	∞	1
	Flat main dish			
		<0 >−1	<0	<0 >−1
	Convex main dish			
Gregorian reflector forms		>1	>0	>0 <1
	Feed in rear (classical form)			
		<1	<0	>0 <1
	Feed in front of main dish focus			

สำหรับสองภาพสุดท้ายที่เป็น Gregorian จะเห็นว่าจุดโฟกัสของจานสะท้อนหลักจะย้ายไปยังตำแหน่งระหว่างจานสะท้อนทั้งสอง และ subreflector จะเป็นรูปทรง elliptical เมื่อทุกๆขนาดและฟีดบีมวิดธ์ของ classical Gregorian เป็นเช่นเดียวกับของ Classical Cassegrain รูปแบบ Gregorian จะต้องการความยาวโฟกัสที่สั้นกว่า สำหรับจานสะท้อนหลักในภาพที่สองของ Gregorian นั้นตัวฟีดถูกย้ายตำแหน่งไปอยู่ระหว่างจุดโฟกัสของจานหลักกับ subreflector ขณะที่จานสะท้อนหลักมีขนาดเท่ากับแบบแรก

จากข้อมูลในตารางที่ 4.1 ความสัมพันธ์ของขนาดของ effective aperture ของ virtual และ real feed จะเหมือนกัน Virtual feed ของ cassegrain ซึ่งมี subreflector เว้า บีมวิดธ์จะถูกควบคุมให้เล็กกว่า และ effective aperture ใหญ่กว่า real feed อย่างไรก็ดี virtual feed ของ classical gregorian ที่มี subreflector เว้า ตัวeffective aperture จะถูกควบคุมให้ขนาดเล็กกว่า real feed

แนวความคิดพาราโบลาสมมูลย์สามารถประยุกต์ได้กับจาน Cassegrain และ Gregorian ทุกแบบ ซึ่งแสดงเป็นเส้นประในตารางที่ 4.1 รูปทรงของจานแบบ Cassegrain และ Gregorian มีความยาวโฟกัสสมมูลย์ซึ่งใหญ่กว่าความยาวโฟกัสของจานสะท้อนหลัก สำหรับ Cassegrain ที่มี subreflector แบนนั้นจะมีความยาวโฟกัสสมมูลย์เท่ากับความยาวโฟกัสของจานสะท้อนหลัก โดยทั่วไปความยาวโฟกัสสมมูลย์ของ Cassegrain ที่มี subreflector เว้านั้นจะสั้นกว่าความยาวโฟกัสของจานสะท้อนหลัก สำหรับจานสะท้อนหลักที่มีลักษณะแบนราบ พาราโบลาสมมูลย์จะเท่ากับ subreflector

บทที่ 5

จานรับสัญญาณดาวเทียมแบบ Offset-fed

5.1 จานรับสัญญาณดาวเทียมแบบ Offset-fed

จานรับสัญญาณแบบ Offset-fed ถูกออกแบบเพื่อหลีกเลี่ยงและลดปัญหาที่เกิดขึ้นกับจานดาวเทียมชนิด Center Focus เช่น การสูญเสียที่เกิดจาก Sidelobes โดยจานรับสัญญาณแบบ Offset-fed จะมีขนาดประมาณ 75 cm

จานรับสัญญาณแบบ Offset-fed ได้จากการนำเอาส่วนหนึ่งของจานรับสัญญาณแบบ Center Focus ขนาดใหญ่ ซึ่งมีพื้นที่ผิวเป็นแบบพาราโบลิกมาใช้งาน อุปกรณ์ที่ใช้รับสัญญาณ (Feedhorn) ยังคงติดตั้งอยู่ที่ตำแหน่งเดิม คือที่จุด Focal ของจานรับสัญญาณแบบ Center Focus ขนาดใหญ่ดั้นแบบเหมือนเดิม ทำให้มองดูแล้วเหมือนกับว่า Feedhorn ยื่นออกมาจากส่วนล่างของจานรับสัญญาณ หากจานรับสัญญาณแบบนี้มีขนาดใหญ่กว่า 1.5 เมตร จะทำให้มีระยะของจุด Focal ยาวมากขึ้น นั่นคือแขนที่จะยึด Feedhorn ก็จะยาวมากขึ้น ทำให้ไม่ได้สัดส่วนและติดตั้งยาก

Feedhorn ที่ใช้กับจานรับสัญญาณแบบ Offset-fed จะมีปากหรือช่องของ Feedhorn ที่แตกต่างจาก Feedhorn ที่ใช้กับจานรับสัญญาณแบบ Center Focus โดยช่องของ Feedhorn ค่อนข้างจะบานออก เพื่อลด Beamwidth ของคลื่นให้เล็กลง

การหามุมของ Offset-fed โดยเป็นอัตราส่วนของ F/D หาได้จากสมการ

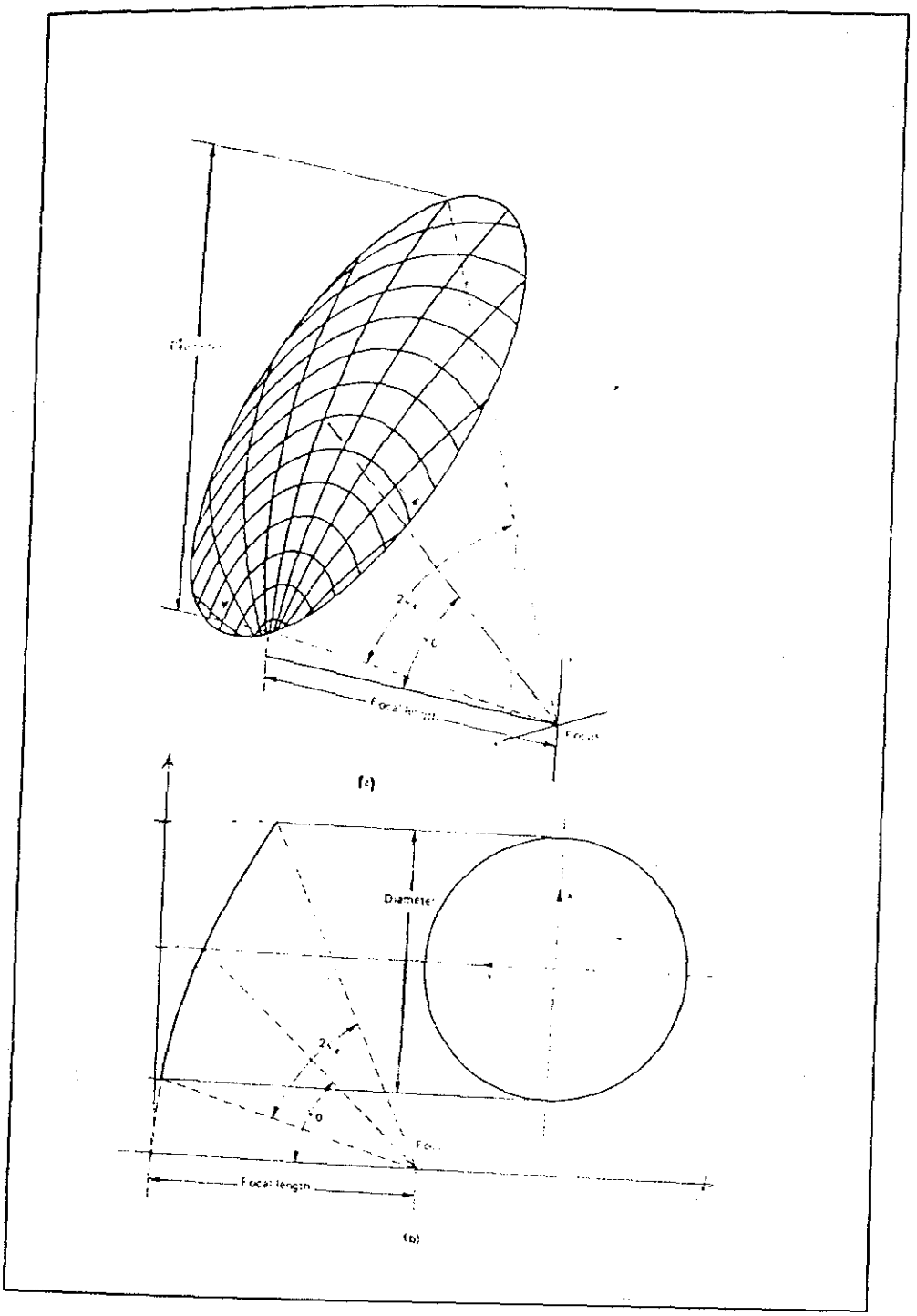
$$\frac{F}{D} = \frac{\cos\psi_c + \cos\psi_o}{4\sin\psi_c} \tag{5.1}$$

เมื่อ ψ_c คือมุมจากแกนของพาราโบลา ถึงจุดศูนย์กลางของจานรับสัญญาณดาวเทียม และจานมีขนาดเป็น $2\psi_c$

เราสามารถวิเคราะห์จานแบบ Offset โดยวิธีการเดียวกับที่ใช้กับจานแบบแกนสมมาตร ความสมมาตรนี้จะช่วยป้องกัน Cross-Polarization ในระนาบของแกน X ดังในรูปที่ 5.1 แต่ Cross-Polarization ระนาบแกนY จะเพิ่มขึ้น โดยอัตราส่วน F/D ลดลง แต่ Spherical wave จะเดินทางไปยังขอบนอกไกลกว่าขอบล่างของจานสะท้อน ดังนั้นจึงทำให้เกิดมุมเอียงของ Circular Polarize โดยจะไม่มี Cross-Polarization

สมการมุมเอียงหาได้จาก

$$\psi_s = \sin^{-1}\frac{\lambda\sin\psi_o}{4\pi F} \tag{5.2}$$



รูปที่ 5.1 แสดงลักษณะจานสายอากาศแบบ Offset-fed

(a) perspective

(b) orthographic

บทที่ 6

Horn Antenna

จากรูปที่ 6.1 แสดงภาพเรขาคณิตเบื้องต้นของฮอร์น ซึ่งท่อนำคลื่นนี้สามารถเป็นได้ทั้งแบบสี่เหลี่ยมผืนผ้า (Rectangular) หรือแบบวงกลม (Circle) หรือวงรี (Elliptical) จากภาพ W คือความกว้างของ rectangular aperture และ a คือ รัศมีของ circular aperture ระยะทางจากจุดต่อของเส้นขอบที่ลากต่อไปยัง aperture คือรัศมีเอียง (R) ระยะทางตลอดเส้นผ่ากึ่งกลางจาก aperture ไปยังท่อนำคลื่นคือ axial length

เราพบ aperture field amplitude จาก input waveguide mode. การกระจายเฟสคลื่นเป็นการประมาณกำลังสองผ่าน aperture ในทรงกลมแพร่กระจายจากจุดที่ถูก project ซึ่งเป็นจุดต่อของขอบ aperture ระยะทางนอกขอบเขตตลอดด้านข้างถูกเปรียบเทียบกับระยะทางที่ไปยังจุดศูนย์กลางของ aperture ดังนี้

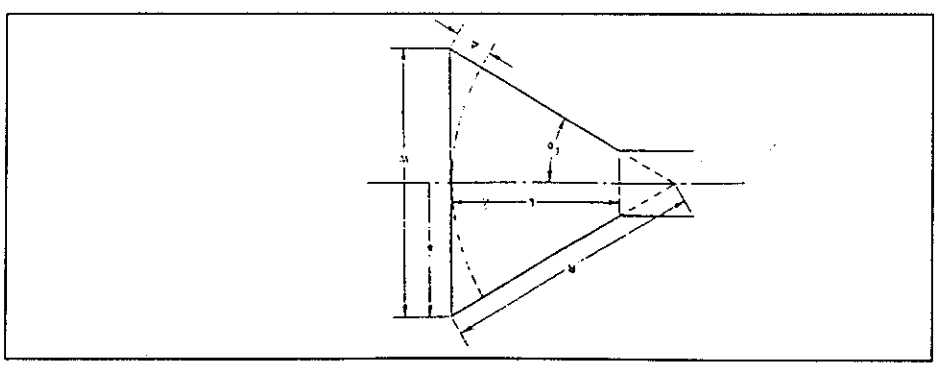
$$\Delta = R - \sqrt{R^2 - a^2}$$

$$= R\left(1 - \sqrt{1 - \left(a^2 / R^2\right)}\right)$$

$$\approx R\left[1 - \left(1 - \frac{a^2}{2R^2}\right)\right] = \frac{a^2}{2R} = \frac{W^2}{8R} \qquad (6\text{-}1)$$

บนการแบ่งโดยใช้ความยาวคลื่น เราสามารถหาขนาดคงที่ S ของสมการกำลังสอง Phase distribution

$$S = \frac{\Delta}{\lambda} = \frac{W^2}{8\lambda R} = \frac{a^2}{2\lambda R}$$



<u>รูปที่ 6.1</u> รูปแบบทางเรขาคณิตของ Horn

เมื่อมุมครึ่งที่หนึ่งบานออก (θ_0) ของฮอร์นมีขนาดเล็กเราจะใช้สมการกำลังสอง Phase distribution ในการประมาณค่า

6.1 Rectangular Horn (Pyramid)

ปากของฮอร์นที่มีลักษณะสี่เหลี่ยมผืนผ้าที่ออกจากท่อนำคลื่นซึ่งทำด้วยโลหะ รัศมีของแนวเอียงด้านข้างของฮอร์นนั้นจะไม่เท่ากันในกรณีปกติ Input waveguide (ท่อนำคลื่น) จะมี a เป็นความกว้าง b เป็นความสูง ส่วน apertuer มีความกว้างเป็น Wในระนาบ H และความสูงเป็น H ในระนาบ E แต่ละตำแหน่ง aperture มีสมการกำลังสอง Phase distribution เป็นค่าคงที่เฉพาะตัวคือ

$$S_e = \frac{H^2}{8\lambda R_e} \qquad S_h = \frac{W^2}{8\lambda R_h} \tag{6-2}$$

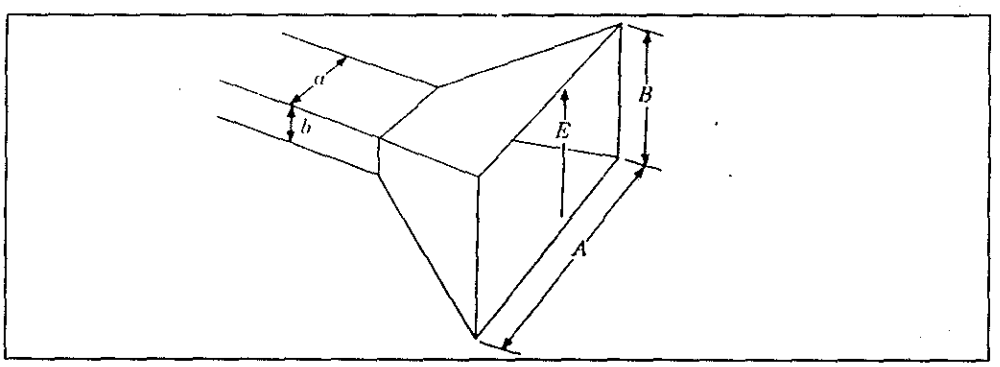
ใน TE_{10} mode ของ lower-order waveguide mode

$$E_y = E_0 \cos (\pi X/a)$$

ใน aperture สนามไฟฟ้าจะถูกประมาณโดย

$$E_y = E_0 \cos (\pi X/W) \exp \{-j2\pi[Se(2y/H)^2+S_h(2X/W)^2]\} \tag{6-3}$$

อัตราส่วนของสนามไฟฟ้ากับสนามแม่เหล็กจะเข้าใกล้อิมพีแดนซ์สำหรับ aperture ขนาดใหญ่ ในกรณีนี้เราใช้การประมาณของ Huygens และต้องการเพียงสนามไฟฟ้าเพื่อหาแพทเทิร์น ส่วนใน small aperture ฮอร์นใช้สมการ (6 - 7) ซึ่งไม่แปรตามอัตราส่วนของสนามแม่เหล็กและสนามไฟฟ้า



รูปที่ 6-2 รูปแบบทางเรขาคณิตของ Rectangular-horn

แพทเทิร์นระนาบ E มาจาก Uniform aperture distribution และระนาบ H มี โคไซน์ Distribution ทั้งสมการกำลังสอง Phase error ในภาพ 6-3 และ 6-4 เป็นแพทเทิร์นระนาบ E ใน U space ของ Taylor ซึ่ง S เป็น พารามิเตอร์ที่ถูกพล็อต เราสามารถใช้มันหาแพทเทิร์นของ Rectangular horn ได้

ตัวอย่าง หาระดับแพทเทิร์นที่ $\theta = 15^\circ$ ในระนาบ E และ H ของ Horn อันหนึ่งซึ่งมีขนาดดังนี้

Aperture : W(H plane) = 28.9 cm. H (E plane) = 21.3 cm.

Input waveguid : Width a = 3.50 cm. height = 1.75 cm. ระยะเอียงจาก aperture ถึงท่อนำคลื่นตลอดศูนย์กลางของแต่ละ plate ของปากวัดได้

$D_h = 44.8$ cm. $D_e = 44.1$ cm.

เราจะหารัศมีเอียงจากสามเหลี่ยมคล้าย

$$\frac{R_h}{D_h} = \frac{W}{W-a} = \frac{R_e}{D_e} = \frac{H}{H-b} \qquad (6.4)$$

ความถี่คือ 8 GHz ($\lambda = 3.75$ cm.) จากสมการ (6-2) เราหา Sh = 0.55 และ Se = 0.31 เราใช้ (6-3) และ (6-4) เพื่อหา Unversal Pattern Field Intensity (Voltage)

$$\frac{W}{\lambda}\sin\theta = 2.0 \qquad \frac{H}{\lambda}\sin\theta = 1.47$$

สนามจากภาพเป็น 0.27 (ระนาบ H) และ 0.36 (ระนาบ E) เราจะรวมแฟคเตอร์ในแนวเอียงของ Hygen source element pattern : $(1+\cos\theta)/2$ เพื่อให้ได้ระดับแพทเทิร์นที่ต้องการ ที่ $\theta = 15^\circ$ แฟคเตอร์ความเอียงคือ 0.983 เราหาระดับแพทเทิร์นในหน่วย dB จาก 20 เท่าของ logarithm ของผลที่ได้จากความเข้มของสนามไฟฟ้าจากรูปและแฟคเตอร์ความเอียง

H plane : -11.5 dB E plane : -9 dB

เราสามารถหาเกน (gain) ของฮอร์นตัวนี้ได้โดยการใช้ aperture efficiencies

ATL H plane (cosine) : 0.91 dB

E plane (uniform) : 0 dB

ค่าเหล่านี้ใช้ได้กับทุกๆ rectangular horn ที่เป็น TE_{10} mode สมการกำลังสอง phase distributions ให้ค่า Phase error loss แก่เรา

$S_h = 0.55$ cosine distribution $PEL_h = 2.09$ dB
$S_e = 0.31$ uniform distribution $PEL_e = 1.5$ dB

ค่าไดเรคติวิตีหาได้โดย

$$\text{Directivity} = 10 \log (4\pi WH / \lambda^2) - ATL_h - ATL_e - PEL_h - PEL_e$$
$$= 22.9 \text{ dB} \qquad (6\text{-}5)$$

Aperture efficiency คือ 35.5% เมื่อ $ATL_h - PEL_h + PEL_e = 4.5$ dB

โดยปกติเราใช้สมการเกนเพื่อหาไดเรคติวิตีเมื่อผนังท่อมี loss เล็กน้อยสำหรับฮอร์นที่มี wave เป็น มิลลิเมตร เราจะต้องรวม loss ที่ผนังด้วย จาก Schelkunoff & Friis ให้ สมการที่เป็นรูปแบบบิดสำหรับ Directivity ดังนี้

$$\text{Directivity} = 8R_hR_e/WH \{[C(u) - C(v)]^2 + [S(u) - S(v)]^2\}[C^2(z)+S^2(z)] \qquad (6\text{-}6)$$

เมื่อ $u = 1/\sqrt{2}\left(\sqrt{\frac{\lambda R_h}{W}} + \frac{W}{\sqrt{\lambda R_h}} \right)$, $v = 1/\sqrt{2}\left(\sqrt{\frac{\lambda R_h}{W}} - \frac{W}{\sqrt{\lambda R_h}} \right)$

$z = H / \left(\sqrt{2\lambda R_e} \right)$

และ $C(x) = \int \cos \left(\pi t^2/2 \right) dt$ $S(x) = \int \sin \left(\pi t^2/2 \right) dt$

ซึ่งเป็นการใช้ Fresnel integrals

6.2 บีมวิดธ์

เราใช้รูป 6.3 และ 6.4 เพื่อหา บีมวิดธ์ เราแปลง 3-dB ให้เป็น 0.707 และ 10-dB ให้เป็น 0.316 บนกราฟ ตาราง 6.1 บอกจุด 3dB และ 10dB (เป็นค่าของU) สำหรับสมการกำลังสองที่แตกต่างกันของตัวคงที่ S ในระนาบH เช่นเดียวกันกับตาราง 6.2 ซึ่งแสดงตำแหน่งในระนาบ E การใช้ตารางจะสะดวกกว่าการใช้กราฟ เพราะเราย้ายแฟคเตอร์ความเอียงเพื่อให้ได้รับ Universal pattern เราต้องปรับปรุง บีมวิดธ์ ที่ได้โดยใช้ตาราง เราต้องเพิ่มแฟคเตอร์ความเอียงเพื่อหาระดับบีมวิดธ์ บีมวิดธ์ที่ได้นี้สำหรับแพทเทิร์นระดับต่ำกว่าที่ระบุ

ตารางที่ 6.1 แสดง Rectangular-Horn H-Plane Beamwidth Points TE_{10} Mode

S	Beamwidth level		S	Beamwidth level	
	3dB	10dB		3dB	10dB
	Location	$(W/\lambda)\sin\theta$		Location	$(W/\lambda)\sin\theta$
0	0.5945	1.0194	0.52	0.8070	1.8062
0.04	0.5952	1.0220	0.56	0.8656	1.8947
0.08	0.5976	1.0301	0.60	0.9401	1.9861
0.12	0.6010	1.0442	0.64	1.0317	2.0872
0.16	0.6073	1.0652	0.68	1.1365	2.2047
0.20	0.6150	1.0949	0.72	1.2445	2.3418
0.24	0.6248	1.1358	0.76	1.3473	2.4876
0.28	0.6372	1.1921	0.80	1.4425	2.6246
0.32	0.6526	1.2700	0.84	1.5320	2.7476
0.36	0.6716	1.3742	0.88	1.6191	2.8618
0.40	0.6951	1.4959	0.92	1.7071	2.9744
0.44	0.7243	1.6123	0.96	1.7991	3.0924
0.48	0.7609	1.7143	1.00	1.8970	3.2208

ตารางที่ 6.2 แสดง Rectangular-Horn E-plane Beamwidths: TE_{10}Mode

S	Beamwidth level		S	Beamwidth level	
	3dB	10dB		3dB	10dB
	Location	$(H/\lambda)\sin\theta$		Location	$(H/\lambda)\sin\theta$
0	0.4430	0.7380	0.24	0.4676	1.4592
0.04	0.4435	0.7405	0.28	0.4793	1.5416
0.08	0.4452	0.7405	0.32	0.4956	1.6034
0.12	0.4482	0.7631	0.36	0.5193	1.6605
0.16	0.4527	0.7879	0.40	0.5565	1.7214
0.20	0.4590	08326	0.44	0.6281	1.8004

$$\frac{BW1}{BW2} = \sqrt{\frac{level_1(dB)}{level_2(dB)}} \qquad (6\text{-}7)$$

<u>ตัวอย่าง</u> หา 3-dB และ10-dB beamwidths สำหรับฮอร์นในตัวอย่างที่ผ่านมา เรามี $S_h = 0.55$ และ $S_e = 0.31$ จากตาราง 6.1 และ 6.2

$(W/\lambda)\sin\theta_{h3} = 0.851$ $(H/\lambda)\sin\theta_{e3} = 0.4915$

$(W/\lambda)\sin\theta_{h10} = 1.8726$ $(H/\lambda)\sin\theta_{e10} = 1.588$

$(W/\lambda) = 28.9/3.75 = 7.707$ $(H/\lambda) = 31.3/3.75 = 5.68$

$\theta_{h3} = 6.34$ $\theta_{h10} = 14.06$ $\theta_{e3} = 4.96$ $\theta_{e10} = 16.24$

เราพิจารณา obliquity factor $(1+\cos\theta)/2$ ที่มีมุมเหล่านี้ และประยุกต์สมการ 6-7 เพื่อหาบีมวิดธ์

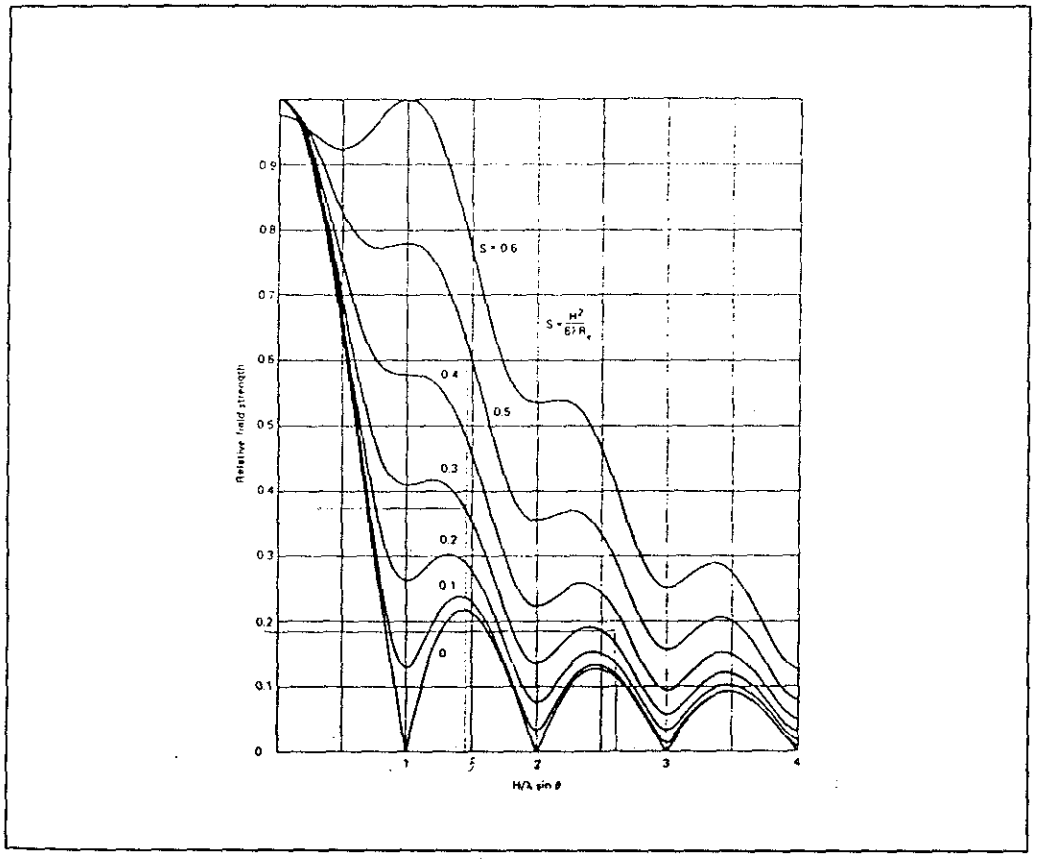
$BW_{h3} = 12.68^\circ$ at 3.03 dB $BW_{h3} = 12.62^\circ$ at 3.01 dB

$BW_{e3} = 9.92^\circ$ at 3.02 dB $BW_{e3} = 9.89^\circ$ at 3.01 dB

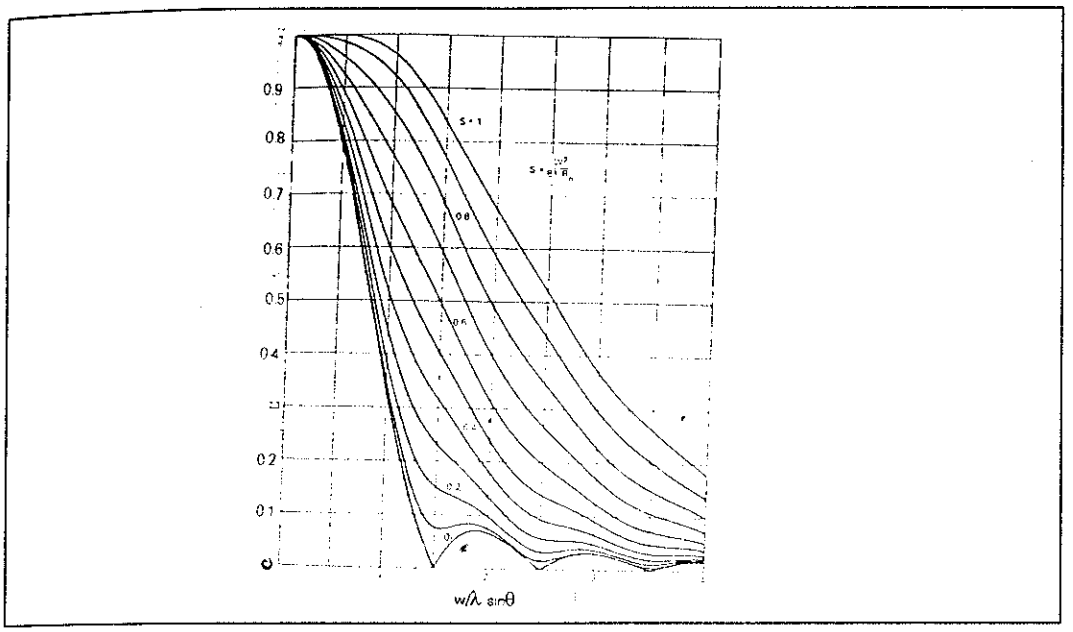
$BW_{h10} = 28.12^\circ$ at 10.13 dB $BW_{h10} = 27.94^\circ$ at 10 dB

$BW_{e10} = 32.48^\circ$ at 10.18 dB $BW_{e10} = 32.2^\circ$ at 10 dB

การรวมเอา obliqity factor มีผลต่อคำตอบน้อยมาก แต่จะมีผลเพิ่มขึ้นถ้า บีมวิดธ์มีขนาดใหญ่ (small aperture)



<u>รูปที่ 6.3</u> แสดง E-plane universal pattern of a rectangular horn TE_{10} mode



รูปที่ 6.4 แสดง H-plane universal pattern of a rectangular horn TE_{10} mode

6.3 Optimum Rectangular Horn

Rectangular horn แต่ละอันจะมีพารามิเตอร์พิเศษ เราสามารถใช้มันออกแบบฮอร์นเพื่อให้ได้ผลดีที่สุด เพื่อให้ได้เกนที่ต้องการเราสามารถออกแบบฮอร์นได้หลายขนาดโดยที่ยังมีเกนเท่าเดิม การออกแบบขึ้นอยู่กับสิ่งที่เราต้องการ เมื่อปราศจากรายละเอียดของสิ่งที่เราต้องการแล้วเราจะนำสายอากาศที่มีระนาบ EและH เท่ากับ3-dB บีมวิดธ์ มาใช้ ถ้าเรานำค่ารัศมีความเอียงที่คงที่และแปรตามความกว้างของ aperture แล้วเกนก็จะเพิ่มขึ้นด้วย การเพิ่มความกว้าง aperture แล้วสมการ phase error loss เพิ่มขึ้นเร็วกว่าและผลที่ได้จะเป็นค่าสูงสุด ค่าสูงสุดนี้จะเกิดขึ้นในทั้ง 2ระนาบที่ประมาณค่าคงที่เฟสด้วยรัศมีความเอียง

$$S_h = 0.40 \qquad S_e = 0.26 \tag{6-8}$$

ที่จุดเหล่านี้เราพบจุด 3-dB จากตาราง 6-1 และ 6-2

$$(W/\lambda)\sin\theta = 0.6951 \qquad (H/\lambda)\sin\theta = 0.4735$$

ในการแบ่งสมการเพื่อหาค่าคงที่ $\sin\theta$ ในสองระนาบ เราพบว่าอัตราส่วนของความกว้างต่อความสูงนั้นจะให้ค่า 3-dB ในทั้งสองระนาบสำหรับจุดที่ดีที่สุด

$$H/W = 0.68 \qquad 3dB \tag{6-9}$$

อัตราส่วนแปรตามระดับบีมวิดธ์ สำหรับ 10 dB beamwidth

$$H/W = 1.00 \qquad 10dB \tag{6-10}$$

ค่าของ S กำหนดประสิทธิภาพของฮอร์นได้ดีที่สุด เราพบว่า PEL ของสมการกำลังสองของ Phase distribution ถูกนำมาใช้สำหรับ H-plane และ Uniform distribution สำหรับ E-plane โดย H-plane distribution มี ATLของ0.91 dB

$PEL_h = 1.14$ dB $\qquad PEL_e = 1.05$ dB $\qquad ATL_h = 0.91$ dB

หรือมีประสิทธิภาพ 49%

$$\text{Gain} = \frac{4\lambda HW}{\lambda^2}(0.49)$$

เราสามารถแก้สมการ H และ W สำหรับหาค่า Gain เมื่อเรารู้อัตราส่วนระหว่างมันจากสมการ (6-9)

$$\frac{W}{\lambda} = \sqrt{\frac{\text{gain}}{4\pi(0.68)(0.49)}} \qquad \frac{H}{\lambda} = \sqrt{\frac{\text{gain}(0.68)}{4\pi(0.49)}}$$

$$\frac{W}{\lambda} = 0.489\sqrt{\text{gain}} \qquad \frac{H}{\lambda} = 0.332\sqrt{\text{gain}} \tag{6-11}$$

เรารวมสมการ 6-8 และ 6-11 เพื่อหารัศมีเอียง

$$\frac{R_h}{\lambda} = 0.0746\ \text{gain} \qquad \frac{R_e}{\lambda} = 0.0531\ \text{gain} \tag{6-12}$$

ถ้าหาเกนได้แล้วเราใช้สมการ 6-11 และ 6-12 เพื่อออกแบบฮอร์นขนาดไม่ขึ้นกับท่อนำคลื่น ความยาวของแกนจากท่อนำคลื่นไปยัง aperture จะต้องเท่ากันในระนาบ E และ H ดังนั้นฮอร์นจะพบท่อนำคลื่นในระนาบเดียว

ให้ขนาดของท่อนำคลื่นเป็น a และ b จะได้แกนความยาวเป็น

$$L_h = \frac{W-a}{W}\sqrt{R_h^2 - \left(R_h^2/4\right)} \qquad L_e = \frac{H-b}{H}\sqrt{R_e^2 - \left(R_e^2/4\right)} \tag{6-13}$$

เรามีตัวเลือกระหว่างระนาบ E หรือ H รัศมีความเอียงหาได้จากสมการ 6-12 และทำให้รัศมีอื่นๆได้จากแกนความยาวเดียวกัน แฟคเตอร์เบื้องต้นคือขนาดของ aperture ซึ่งเราได้จากสมการ 6-11 ส่วนรัศมีความเอียงในอันดับต่อมาเราได้จากระนาบ H จากสมการ 6-12 และพัฒนาระนาบ E การพัฒนาระนาบ H จะได้

$$R_e = \frac{H}{H - b}\sqrt{L^2 + (H-b)^2 h/4} \qquad (6\text{-}14)$$

เพื่อให้ได้เกนที่ต้องการ เราต้องทำซ้ำ เมื่อเราไม่สามารถใช้สมการที่ 6-12 การออกแบบฮอร์นโดยใช้สมการ 6-11, 6-12a, 6-13a และ 6-14 การคำนวณเกนจากขนาดจะได้การออกแบบเกนใหม่จาก

$$G_{d,new} = \frac{G_{reg}\, G_{d,old}}{G_{actual}} \qquad (6\text{-}15)$$

โดยที่ G_{reg} คือเกนที่ต้องการ G_{actual} คือ actual gain และG_d คือเกนเดิมที่ได้จากการออกแบบ

บทที่ 7

Circular- Aperture Horn

เมื่อเราใช้ Circular-Aperture Horn ทำให้เราไม่จำเป็นต้องควบคุม beamwidths ในระนาบ E และ H เพราะ Circular Waveguide สามารถเข้าได้กับทุกทิศทางของสนามไฟฟ้า ดังนั้นจึงสามารถเข้าได้ทุก Polarization ของ Horn Aperture phase สามารถประมาณเป็นสมการกำลังสอง

สนามของ ท่อนำคลื่น สามารถหาได้จากสมการ (7-1)

$$E_\rho = \frac{E_0}{\rho} J_1\left(\frac{x'_{11}\rho}{a}\right) \cos \phi_c$$

$$E_{\phi_c} = -\frac{E_0 x'_{11}}{a} J'_1\left(\frac{x'_{11}\rho}{a}\right) \sin \phi_c \qquad (7\text{-}1)$$

เมื่อ J_1 คือ Bessel function ρ คือรัศมีส่วนประกอบของ ท่อนำคลื่น, a คือรัศมี และ ϕ_c คือ cylindrical coordinate, $x'_{11}(1.841)$ คือ first zero ของ $J'_1(x)$ สมการ (7-1) คือสนามไฟฟ้าสูงสุดที่มีทิศทางตามระนาบ $\phi_c = 0$

เมื่อแทนค่า สมการกำลังสองของ phase factor ลงในสมการ (7-1) และคำนวณ Fourier Transform บน Circular Aperture เพื่อที่จะหา far field ทิศทางของสนามไฟฟ้าจะเปลี่ยนจุดต่อจุดใน Aperture สำหรับทิศทาง a_θ และ a_ϕ ก่อน integrating บน Aperture

$$E_\theta = E_0 \int_0^{2\pi} \int_0^a \left[\frac{J_1(x'_{11}\rho/a)}{\rho} \cos \phi_c \frac{a_\rho \cdot a_{\rho_c}}{\cos \theta} - \frac{x'_{11}}{a} J'_1\left(\frac{x'_{11}\rho}{a}\right) \sin \phi_c \frac{a_\theta \cdot a_{\phi_c}}{\cos \theta} \right]$$
$$\times \rho \exp\left\{ j\left[k\rho \sin \theta \cos(\phi - \phi_c) - 2\pi S\left(\frac{\rho}{a}\right)^2 \right] \right\} d\rho\, d\phi_c$$

$$E_\phi = E_0 \int_0^{2\pi} \int_0^a \left[\frac{J_1(x'_{11}\rho/a)}{\rho} \cos \phi_c a_\rho \cdot a_{\rho_c} - \frac{x'_{11}}{a} J'_1\left(\frac{x'_{11}\rho}{a}\right) \sin \phi_c a_\rho \cdot a_{\phi_c} \right]$$
$$\times \rho \exp\left\{ j\left[k\rho \sin \theta \cos(\phi - \phi_c) - 2\pi S\left(\frac{\rho}{a}\right)^2 \right] \right\} d\rho\, d\phi_c$$

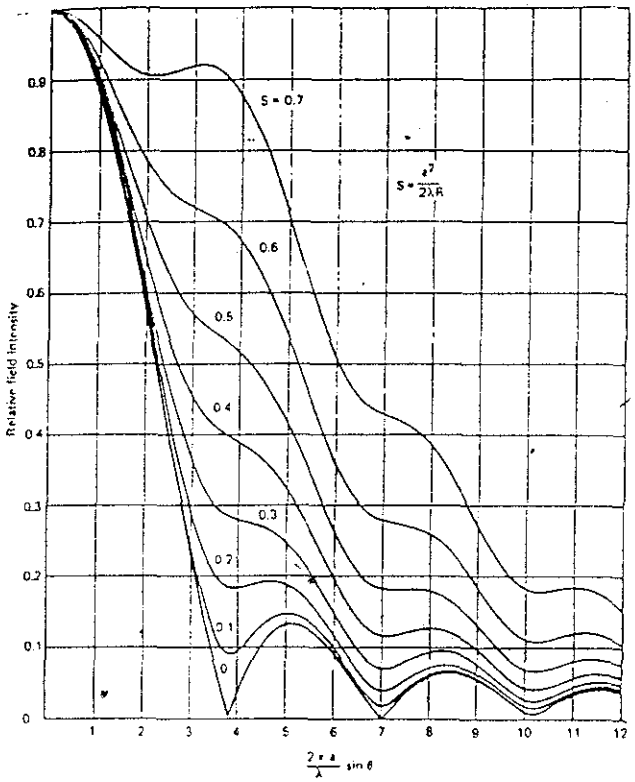
$$a_\theta \cdot a_\rho = \cos \theta(\cos \phi \cos \phi_c + \sin \phi \sin \phi_c)$$

$$a_\theta \cdot a_{\phi_c} = \cos \theta(\sin \phi \cos \phi_c - \cos \phi \sin \phi_c)$$

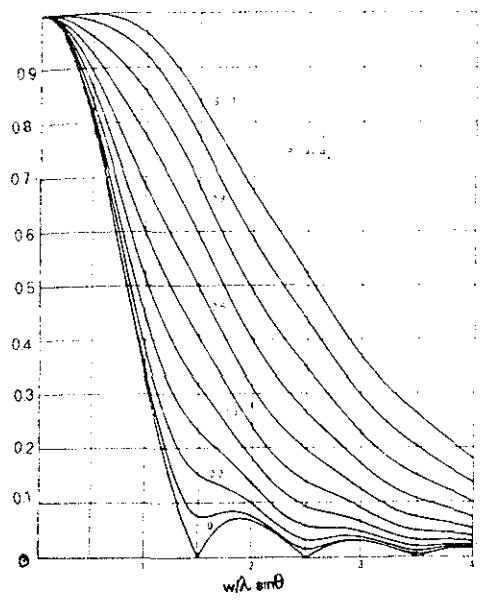
$$a_\phi \cdot a_\rho = \cos \phi \sin \phi_c - \sin \phi \cos \phi_c$$

$$a_\phi \cdot a_{\phi_c} = \cos \phi \cos \phi_c + \sin \phi \sin \phi_c$$

การเปลี่ยนค่าตัวแปรที่เหมาะสมในการ integral รูปแบบการแผ่กระจายคลื่น สำหรับ E และ H plane (รูป 7-1 และ 7-2)



รูปที่ 7.1 แสดง E-plane universal pattern of cicular Horn TE_{11} mode. (source: T. Milligan, "Universal Patterns Ease Circular Horn Design," Microwave, vol.20 no. 3, March 1981, p. 84.)



รูปที่ 7.2 แสดง H-plane universal pattern of cicular Horn TE_{11} mode. (source: T. Milligan,"Universal Patterns Ease Circular Horn Design," Microwave, vol.20 no. 3, March1981, p. 84.)

ตัวอย่างที่1

Horn มี Aperture radius 12 cm และ รัศมีเอียง (slant radius) 50 cm จงหาระดับ Pattern เมื่อ $\theta = 20^\circ$ ความถี่ 5 GHz

$$S = \frac{a^2}{2\lambda R} = \frac{12^2}{2(6)(50)} = 0.24$$

$$\frac{2\pi a}{\lambda} \sin\theta = 4.3$$

จากรูป 7-1 และ 7-2 สามารถหารูปแบบของระดับแรงดัน:

H-plane level = 0.18 (-14.7 dB) E-plane level = 0.22 (-13.1 dB)

The obliquity factor is 20 log [(1+cos20°)/2] = -0.3 dB

ระนาบของระดับ 20° กลายเป็น

H-plane level = -15 dB E-plane level = -13.4 dB

7.1 Beamwidth

ตาราง 7-1 แสดง จุดระหว่าง 3dB และ 10 dB จากรูปที่ 7-1 และ 7-2 ซึ่งตารางนี้จะช่วยให้หาขนาดของ Beamwidth ได้

ตัวอย่างที่ 2

จงหา 10-dB beamwidth ของ horn ในตัวอย่างที่ 1

S = 0.24 a = 12 cm λ =6 cm

จากตาราง 7-1 สามารถหา k-space ที่ 10 dB

$$\text{H-plane} \quad \text{k-space} = \frac{2\pi a}{\lambda} \sin\theta_h = 3.6155$$

$$\text{E-plane} \quad \text{h-space} = \frac{2\pi a}{\lambda} \sin\theta_c = 3.0024$$

$$BW = 2\sin^{-1}\frac{3.6155}{4\pi} \qquad BW = 2\sin^{-1}\frac{3.0024}{4\pi}$$

$$= 33.40^\circ \qquad = 27.65^\circ$$

ตาราง 7.1 แสดง Circular-Horn Beamwidths TE_{11} Mode

S	Beamwidth level				ATL+PEL (dB)
	3dB		10dB		
	E-Plane	H-Plane	E-Plane	H-Plane	
	Location	$(2\pi a/\lambda)\sin\theta$	Location	$(2\pi a/\lambda)\sin\theta$	
0	1.6163	2.0376	2.7314	3.5189	0.77
0.04	1.6175	2.0380	2.7368	3.5211	0.80
0.08	1.6212	2.0391	2.7536	3.5278	0.86
0.12	1.6273	2.0410	2.7835	3.5393	0.96
0.16	1.6364	2.0438	2.8296	3.5563	1.11
0.20	1.6486	2.0477	2.8982	3.5799	1.30
0.24	1.6647	2.0527	3.0024	3.6115	1.54
0.28	1.6855	2.0592	3.1757	3.6536	1.82
0.32	1.7123	2.0676	3.5720	3.7099	2.15
0.36	1.7471	2.0783	4.6423	3.7863	2.53
0.40	1.7930	2.0920	5.0492	3.8933	2.96
0.44	1.8552	2.1100	5.3139	4.0504	3.45
0.48	1.9441	2.1335	5.5375	4.2967	3.99
0.52	2.0823	2.1652	5.7558	4.6962	4.59
0.56	2.3435	2.2089	6.0012	5.2173	5.28
0.60	3.4329	2.2712	6.3500	5.6872	5.98
0.64	4.3656	2.3652	7.6968	6.0863	6.79
0.68	4.8119	2.5195	8.4389	6.4622	7.66
0.72	5.1826	2.8181	8.8519	6.8672	8.62

เราสามารถใช้ตารางที่ 7-1 สำหรับในการออกแบบ Horn ตามขนาด บีมวิดธ์ ที่ต้องการ แต่สามารถออกแบบได้เพียงระนาบเดียวเท่านั้น จำนวนของHorn สามารถถูกออกแบบได้ตาม บีมวิดธ์ แต่ S ไม่ขึ้นกับพารามิเตอร์

7.2 ค่าทวีกำลัง

ตารางที่ 7-1 แสดงทั้ง amplitude taper loss และ phase error loss ซึ่งเป็นฟังก์ชั่นของ S ใน Circular horn ดังนั้นจากตารางนี้จึงเป็นการง่ายที่จะหา gain จาก horn ที่กำหนดให้หรือออกแบบ horn จากเกน ที่ให้มา

ตัวอย่างที่ 3

จงหา เกนของฮอร์น ที่มีรัศมี aperture 12 cm และ slant radius 50 cm ที่ความถี่ 5 GHz จากตัวอย่างที่ผ่านมา S = 0.24 และ λ =6 cm

$$\text{Gain} = 20 \log\left(\frac{\pi D}{\lambda}\right) - GF \qquad \text{where } GF = ATL + PEL \text{ (dB)}$$

$$\text{Gain} = 20 \log\left(\frac{\pi 24}{6}\right) - 1.54 = 20.4 \text{ dB}$$

7.3 Phase Center

ตำแหน่งของ phase center จะอยู่ที่ด้านหลังของระนาบ Aperture ซึ่งเป็นฟังก์ชันของ S ตารางที่ 7-2 จะแสดง phase center ที่ตำแหน่งตามอัตราส่วน slant radius เมื่อ S เพิ่มขึ้นตำแหน่งของ phase center จะเข้าใกล้ Horn และความแตกต่างระหว่าง phase center ใน E และ H จะเพิ่มขึ้น

ตารางที่ 7.2 แสดง Phase Center Axial Location of a Circular-Waveguide Horn(TE_{11}Mode) behind the Aperture as a Ratio of the Slant Radius

	H-plane	E-Plane		H-plane	E-Plane
S	L_{ph}/R	L_{pe}/R	S	L_{ph}/R	L_{pe}/R
0	0	0	0.28	0.235	0.603
0.04	0.0046	0.012	0.32	0.310	0.782
0.08	0.018	0.048	0.36	0.397	0.801
0.16	0.075	0.194	0.44	0.604	0.836
0.20	0.117	0.305	0.48	0.715	0.872
0.24	0.171	0.441			

บทที่ 8

การออกแบบจานรับ-ส่งสัญญาณดาวเทียมแบบ Cassegrain

การ Design Dish Antenna และ Circular feed horn

ต้องการ: Dish antenna แบบ Cassegrain และ Feed horn แบบ Circular ที่มี Gainรวม เท่ากับ 35 dB ขึ้นไป

กำหนดให้: ความถี่ (f) = 28 GHz F/D = 0.6 เส้นผ่าศูนย์กลางจานขนาด = 35 cm.

สิ่งที่ได้ : λ = 1.07 cm

D = 35 cm D: Diameterของจานสายอากาศ

และจากสมการ

$$\text{Focal length(F)} = \frac{D^2}{16d}$$

จะทำให้ทราบค่า ดังต่อไปนี้

d = 3.65 cm d: ความลึกของจานสายอากาศ

F = 21 cm F: ความยาวโฟกัสของจานสายอากาศ

การคำนวณGain ของจานสายอากาศแบบCassegrain

$$\text{Gain(dB)} = 10\log\left(2+\left[\frac{d}{D}\left(\frac{\pi dD}{3\lambda}\right)^2\right]\right)$$

$$\text{แทนค่า Gain(dB)} = 10\log\left(2+\left[\frac{3.65}{35}\left(\frac{\pi\times 3.65\times 35}{3\times 1.07}\right)^2\right]\right)$$

$$= 32.13 \text{ dB}$$

คำนวณ บีมวิดธ์ ของจานสายอากาศ

$$\theta_{-3dB} = 2\tan^{-1}\left(\frac{\lambda\times F}{d^2 D}\right)$$

แทนค่า $\theta_{-3dB} = 2\tan^{-1}(1.07x21/3.65^2x35)$

$= 5.52^\circ$

แต่เนื่องจาก บีมวิดธ์ สูงเกินไป เราต้องการให้ได้ประมาณ 2° เท่านั้น จึงทำการเปลี่ยนแปลงค่า ความลึก (d) Focal length ใหม่ เพื่อให้ได้ บีมวิดธ์ ที่ต้องการ ดังนี้

$$2^\circ = 2\tan^{-1}\left(\frac{1.07xF}{d^2x35}\right)$$

และจาก Focal length(F) $= \frac{D^2}{16d}$

จะได้ $2^\circ = 2\tan^{-1}\left(\frac{1.07x35^2}{d^2x35x16d}\right)$

$\therefore d = 5.11$ cm

แทนค่าหา Focal length

$F = 35^2/(5.11x16)$

$F = 14.95$ cm

เมื่อค่าของ Focal length เปลี่ยนไปจึงมีผลทำให้ F/D ratio เปลี่ยนแปลงด้วย

$F/D = 14.95 / 35$

$= 0.43$

เมื่อเปลี่ยนค่า d ความลึกของจานแล้วจะมีผลต่อ gain ของจานสายอากาศด้วย จึงต้องกลับไปคิดใหม่ โดยเปลี่ยนค่า d = 5.11 cm จะได้ค่าเกนใหม่ดังนี้

$$\text{Gain(dB)} = 10\log\left(2+\left[\frac{5.11}{35}\left(\frac{\pi x5.1135}{3x1.07}\right)^2\right]\right)$$

$= 36.5$ dB

ซึ่งเป็นเกนที่สูงขึ้นจากเดิม แต่ได้ บีมวิดธ์ ตามที่ต้องการ

เราได้gain ของจานสายอากาศเท่ากับ 36.5 dB ในที่นี้ยังไม่รวม gain ของ Feed horn

การคำนวณค่าทวีกำลังของCircular feed horn

ที่ Optimum Circular feed horn

S = 0.39 GF = 2.85 dB (ATL+PEL)

S: Dimensionless constant of the quadratic phase distribution ($S=a^2/2\lambda R$)

GF : ATL+PEL in dB

ATL : Amplitude taper loss

PEL : Phase error efficiency

คำนวณขนาดเส้นผ่านศูนย์กลางของ feed horn

$$D = \left(\frac{\lambda}{\pi}\right)10\log^{(\text{gain+GF})/20}$$

แทนค่า

$$D = (1.07/\pi)\ 10^{(12 + 2.85)/20}$$

$$D = 1.88 \text{ cm.}$$

หา Slant Radius (R)

$$R = D^2/8\pi S$$

แทนค่า

$$R = (1.88)^2 / 8(1.07)(0.39)$$

$$= 1.06 \text{ cm.}$$

หมายเหตุ : เราเลือกใช้ Gain ของ Feed horn = 12 dB เพราะ ถ้า Gain ต่ำกว่านี้ จะไม่สามารถสร้าง Feed horn ได้ ดังจะแสดงให้เห็นดังตารางต่อไปนี้

Gain	D (cm.)	D/2 (cm.)	R (cm.)
3 dB	0.667	0.334	0.134
5 dB	0.841	0.42	0.212
7 dB	1.058	0.53	0.335
9 dB	1.332	0.67	0.532
11 dB	1.678	0.84	0.843
12 dB	1.88	0.94	1.06

จะเห็นว่า Gainตั้งแต่ 3dB - 11dB ไม่สามารถสร้าง Circular feed horn ได้เพราะ R (Slant Radius) มีค่าน้อยกว่า D/2

คำนวณขนาดของ Wave Guide.

$$\lambda = 2\pi r / \mu_{1,1}$$

ที่ $TE_{1,1}$ mode มี $\mu_{1,1} = 1.84$ (หา Bessel Root)

radius of waveguide : $r = (\lambda \times \mu_{1,1}) / 2\pi$

$$= (1.07\text{cm.} \times 1.84) / 2\pi$$

$$r = 0.313 \text{ cm. หรือ } 3.13 \text{ mm.}$$

diameter of waveguide $d = 0.313 \times 2 = 0.627$ cm. หรือ 6.27 mm.

การคำนวณ บีมวิดธ์ ของ circular horn

จากตารางที่ 7.1

S	Beamwidth level				ATL+PEL (dB)
	3dB		10dB		
	E-Plane	H-Plane	E-Plane	H-Plane	
	Location	$(2\pi a/\lambda)\sin\theta$	Location	$(2\pi a/\lambda)\sin\theta$	
0	1.6163	2.0376	2.7314	3.5189	0.77
0.04	1.6175	2.0380	2.7368	3.5211	0.80
0.08	1.6212	2.0391	2.7536	3.5278	0.86
0.12	1.6273	2.0410	2.7835	3.5393	0.96
0.16	1.6364	2.0438	2.8296	3.5563	1.11
0.20	1.6486	2.0477	2.8982	3.5799	1.30
0.24	1.6647	2.0527	3.0024	3.6115	1.54
0.28	1.6855	2.0592	3.1757	3.6536	1.82
0.32	1.7123	2.0676	3.5720	3.7099	2.15
0.36	1.7471	2.0783	4.6423	3.7863	2.53
0.40	1.7930	2.0920	5.0492	3.8933	2.96
0.44	1.8552	2.1100	5.3139	4.0504	3.45
0.48	1.9441	2.1335	5.5375	4.2967	3.99
0.52	2.0823	2.1652	5.7558	4.6962	4.59
0.56	2.3435	2.2089	6.0012	5.2173	5.28
0.60	3.4329	2.2712	6.3500	5.6872	5.98
0.64	4.3656	2.3652	7.6968	6.0863	6.79
0.68	4.8119	2.5195	8.4389	6.4622	7.66
0.72	5.1826	2.8181	8.8519	6.8672	8.62

ที่ S = 0.39 จะได้

H-plan k space = $(2\pi a/\lambda)\sin\theta_h = 2.008$

E-plane k space = $(2\pi a/\lambda)\sin\theta_e = 1.728$

นำมาหาค่า บีมวิดธ์.

$BW_h = 2\sin^{-1}[2.088/(2\pi a/\lambda)]$ $BW_e = 2\sin^{-1}[1.782/(2\pi a/\lambda)]$

$$= 2\sin^{-1}\left(\frac{2.088}{2\pi \times 0.94 / 1.07}\right) \qquad = 2\sin^{-1}\left(\frac{1.782}{2\pi \times 0.94 / 1.07}\right)$$

$$= 44.45^{\circ} \qquad = 37.67^{\circ}$$

คำนวณ Phase Center location in E and H plane for circular horns.

จากตารางที่ 6-2 ที่ s= 0.39 with R=1.061cm.

H-plane phase center = 0.471(1.061cm.) = 0.5 cm.

E-plane phase center = 0.807(1.061cm.) = 0.856 cm.

การคำนวณหา Cassegrain Dish

กำหนด D = 35 cm

F/D = 0.43

F = 14.95 cm

d = 5.11 cm

หา Ds : Hyperbola Diameter

$$D_s = \sqrt{2\lambda F}$$

แทนค่า $D_S = 2 \times 1.07 \times 14.95$

$= 5.65$ cm

หา Fs : Focal length of hyperbola

$$F_s = \frac{D_s F}{D}$$

แทนค่า $$F_s = \frac{5.65 \times 14.95}{35}$$

$= 2.416$ cm

หา F_3 : Focal length of the horn to hyperbola

$$F_3 = \frac{5.56}{2\tan\frac{44.45}{2}}$$

แทนค่า $= 6.91$ cm

การ plot curve ของ main reflector (parabola)

$$Y = \sqrt{4FX}$$

โดยที่ Y มีค่า ตั้งแต่ 0-D/2 หรือ 0-17.5

Y	X	Y	X
0	0	9	0.964
0.5	2.97×10^{-3}	9.5	1.074
1	1.19×10^{-2}	10	1.19
1.5	2.67×10^{-2}	10.5	1.312
2	4.76×10^{-2}	11	1.44
2.5	4.77×10^{-2}	11.5	1.57
3	0.107	12	1.741
3.5	0.146	12.5	1.86
4	0.190	13	2.012
4.5	0.241	13.5	2.169
5	0.297	14	2.33
5.5	0.360	14.5	2.5
6	0.428	15	2.678
6.5	0.503	15.5	2.86
7	0.583	16	3.047
7.5	0.669	16.5	3.241
8	0.762	17	3.44
8.5	0.260	17.5	3.645

การ plot curve ของ subreflector (hyperbola)

สมการ Hyperbola คือ

$$\frac{X^2}{a^2} - \frac{Y^2}{f^2 - a^2} = 1$$

โดยที่ a=OA=OB , f = OF'=OF

จากที่ผ่านมา เราหาตำแหน่งที่จะวาง Subreflector ซึ่งจะได้ค่า

$F_3 = 6.91$ cm

$F_s = 2.416$ cm

เมื่อเปรียบเทียบรูปทั้งสองแล้วจะได้

$$OF'=OF=\frac{F_3 + F_S}{2}$$

แทนค่า OF = (6.91+2.42)/2

= 4.67 cm

OA=OB= OF-F_S

แทนค่า OB = 4.67-2.42

= 2.25 cm

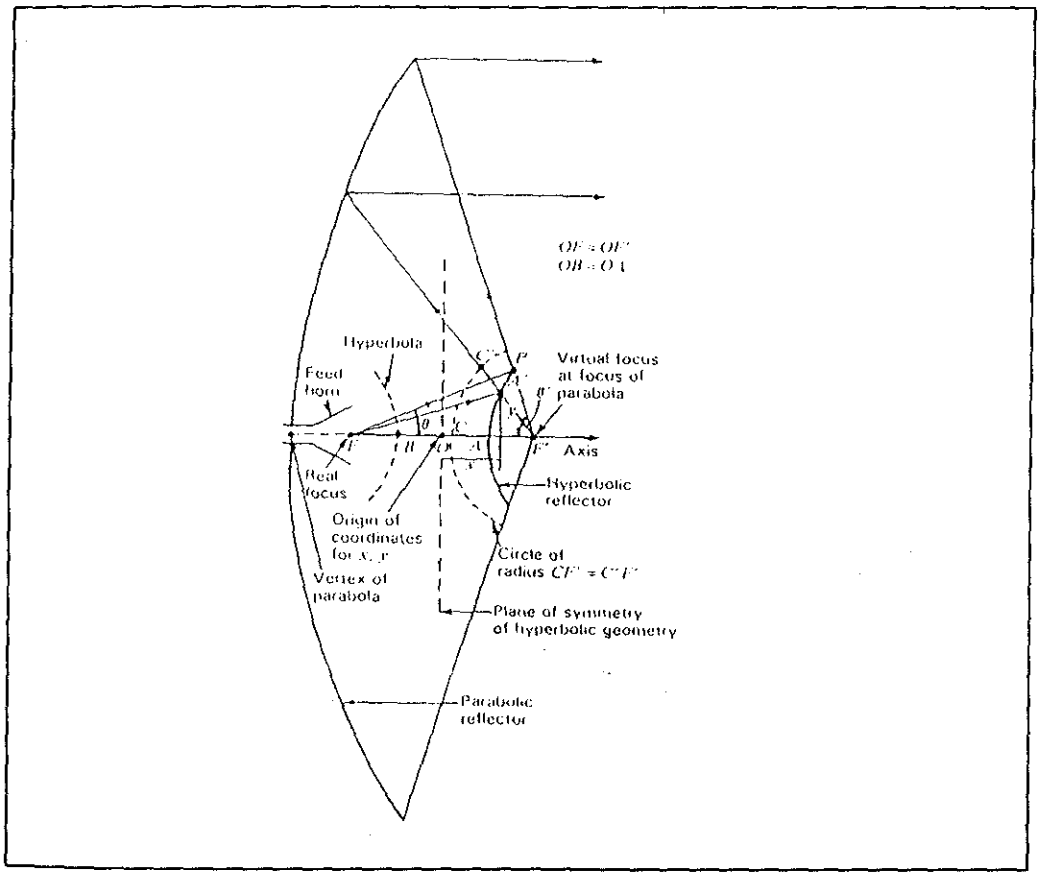
แทนค่าลงในสมการ Hyperbola

$$\frac{X^2}{OB^2} - \frac{Y^2}{OF^2 - OB^2} = 1$$

$$\frac{X^2}{2.25^2} - \frac{Y^2}{4.67^2 - 2.25^2} = 1$$

$$Y^2 = (X^2/2.25^2 - 1)(15.55)$$

โดยที่ Y มีค่าตั้งแต่ 0 ถึง $D_S/2$ หรือ 0 ถึง 2.81



รูปที่ 8.1 แสดงระยะต่างๆบนจานสายอากาศแบบ Cassegrain

บทที่ 9

การออกแบบจานรับ-ส่งสัญญาณดาวเทียมแบบ Offset-fed

การ Design Dish Antenna และ Rectangular feed horn กรณีที่ทราบค่ามุม $\psi_o = 40^\circ$

ต้องการ: Dish antenna แบบ Offset-fed ที่มีGain รวมเท่ากับ 40 dB ขึ้นไป

กำหนดให้: ความถี่ (f) = 14 GHz , F/D = 0.5 , เส้นผ่าศูนย์กลางจานขนาด = 35 cm.

สิ่งที่ได้ : λ = 2.14 cm

D = 35 cm D: Diameterของจานสายอากาศ

และจากสมการ

$$\text{Focal length(F)} = \frac{D^2}{16d}$$

จะทำให้ทราบค่าดังต่อไปนี้

d = 4.38 cm d: ความลึกของจานสายอากาศ

F = 17.5 cm F: ความยาวโฟกัสของจานสายอากาศ

การคำนวณGain ของจานสายอากาศแบบCassegrain

$$\text{Gain(dB)} = 10\log\left(2+\left[\frac{d}{D}\left(\frac{\pi dD}{3\lambda}\right)^2\right]\right)$$

$$\text{แทนค่า Gain(dB)} = 10\log\left(2+\left[\frac{4.38}{35}\left(\frac{\pi\times 4.38\times 35}{3\times 2.14}\right)^2\right]\right)$$

$$= 28.48 \text{ dB}$$

คำนวณ บีมวิดธ์ ของจานสายอากาศ

$$\theta_{-3dB} = 2\tan^{-1}\left(\frac{\lambda\times F}{d^2 D}\right)$$

แทนค่า $$\theta_{-3dB} = 2\tan^{-1}\left(\frac{2.14\times 17.5}{4.38^2\times 35}\right)$$

$$= 6.39^\circ$$

แต่เนื่องจาก บีมวิดธ์ สูงเกินไป เราต้องการให้ได้ประมาณ 2° เท่านั้น จึงทำการเปลี่ยนแปลงค่า ความลึก (d), Focal length ใหม่ เพื่อให้ได้ บีมวิดธ์ ที่ต้องการ ดังนี้

$$2^\circ = 2\tan^{-1}\left(\frac{2.14xF}{d^2x35}\right)$$

และจาก $$\text{Focal length(F)} = \frac{D^2}{16d}$$

จะได้ $2^\circ = 2\tan^{-1}\left(\frac{2.14x35^2}{d^2x35x16d}\right)$

$\therefore d = 6.45$ cm

แทนค่าหา Focal length

$F = 35^2/(6.45x16)$

$F = 11.87$ cm

เมื่อค่าของ Focal length เปลี่ยนไปจึงมีผลทำให้ F/D ratio เปลี่ยนแปลงด้วย

$F/D = 11.87 / 35$

$= 0.34$

เมื่อเปลี่ยนค่า d ความลึกของจานแล้วจะมีผลต่อ gain ของจานสายอากาศด้วย จึงต้องกลับไปคิดใหม่ โดยเปลี่ยนค่า d = 6.45 cm จะได้ค่าเกนใหม่ดังนี้

$$\text{Gain(dB)} = 10\log\left(2+\left[\frac{6.45}{35}\left(\frac{\pi\times 6.45\times 35}{3\times 2.14}\right)^2\right]\right)$$

$= 33.52$ dB

ซึ่งเป็นเกนที่สูงขึ้นจากเดิม แต่ได้ บีมวิดธ์ ตามที่ต้องการ

การคำนวณค่าทวีกำลังของ Rectangular feed horn

พบว่าต้องการ Gain ที่ 12 dB และใช้งานที่ความถี่ 14 GHz (λ =2.14)

$G = 10^{(12/10)} = 15.85$ เท่า

$\frac{W}{\lambda} = 0.489\sqrt{\text{gain}}$ $\qquad$ $\frac{H}{\lambda} = 0.332\sqrt{\text{gain}}$

$\frac{W}{2.14} = 0.489\sqrt{15.85}$ $\qquad$ $\frac{H}{2.14} = 0.332\sqrt{15.85}$

$W = 4.16$ cm $\qquad$ $H = 2.82$ cm

$\frac{R_h}{\lambda} = 0.0746$ gain

$=0.0746\,(15.85)(2.14) = 2.53$

$$L_h = \frac{W-a}{W}\sqrt{R_h^2-\left(W^2/4\right)}$$

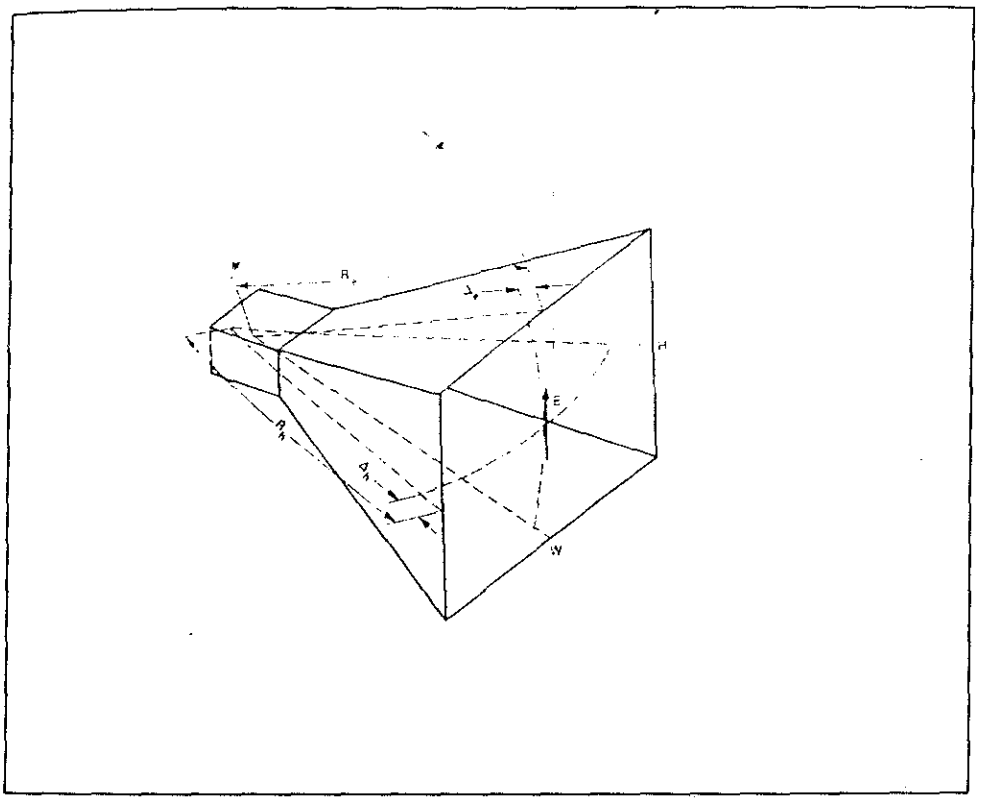
$$L_h = \frac{4.16-0.54}{4.16}\sqrt{(2.53)^2-\left(4.16^2/4\right)}$$

$$= 0.87\sqrt{6.4 - 4.32} = 1.25 \text{ cm}$$

$$R_e = \frac{H}{H - b}\sqrt{L^2 + (H - b)^2 / 4}$$

$$R_e = \frac{2.82}{2.82 - 0.27}\sqrt{1.25^2 + (2.82 - 0.27)^2 / 4}$$

$$Re = 1.11\sqrt{1.56 + 1.63} = 1.98$$



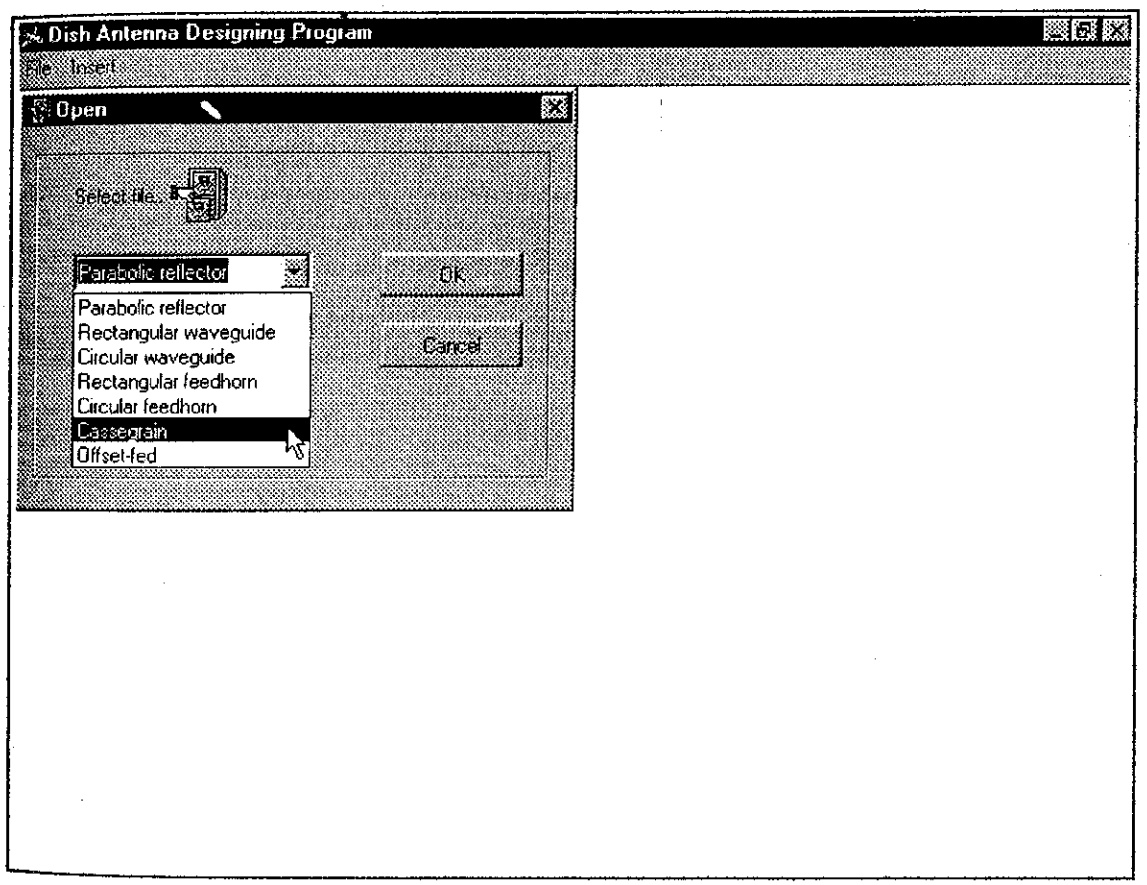
รูปที่ 9.1 แสดงระยะต่างๆ ของ Rectangular Feedhorn

บทที่ 10
สรุปผลการดำเนินงาน

จากการดำเนินงานได้โปรแกรมออกแบบจานรับสัญญาณดาวเทียมที่สามารถออกแบบจานรับสัญญาณดาวเทียมได้ทั้งแบบ Cassegrain และ Offset-fed รวมทั้งฟีดฮอร์นและท่อนำคลื่น โดยมีรายละเอียดของโปรแกรมดังต่อไปนี้

10.1 การออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain

10.1.1 เลือก File open และ เลือกการออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain ดังรูป



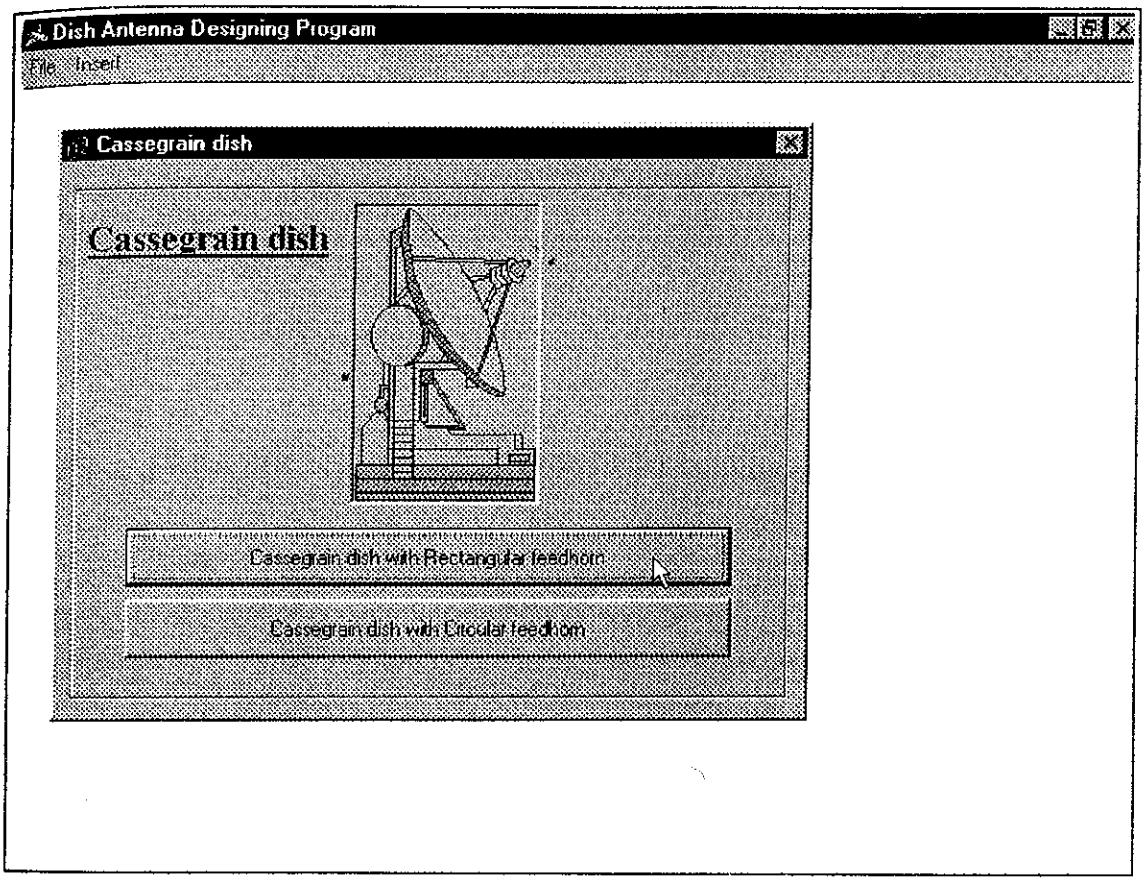
รูปที่10.1 แสดงการเลือกชนิดการออกแบบจานรับสัญญาณดาวเทียม

10.1.2 การออกแบบจานรับสัญญาณดาวเทียมแบบ Cassegrain มีให้เลือก 2 แบบ

10.1.2.1 Cassegrain dish with Rectangular Feedhorn

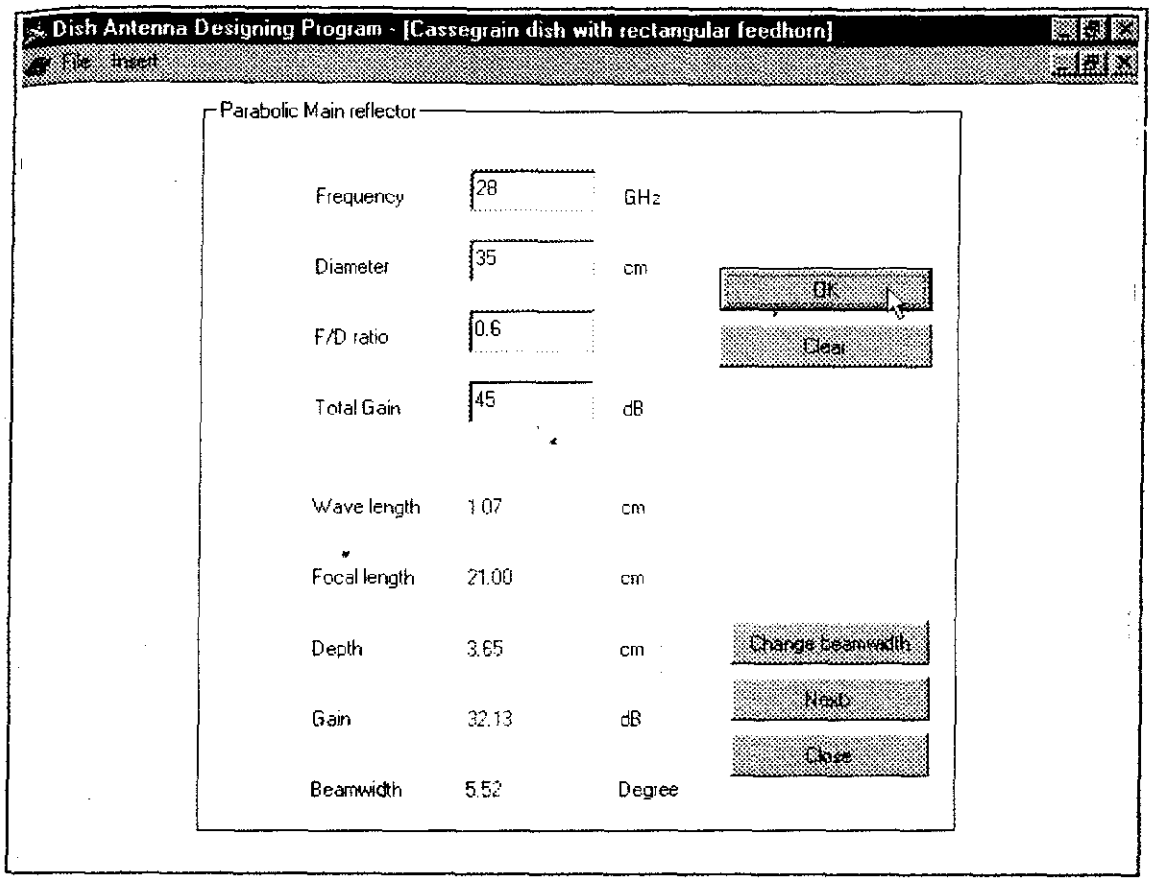
10.1.2.2 Cassegrain dish with Circular Feedhorn

เช่น เลือกการออกแบบจาน Cassegrain dish with Rectangular Feedhorn



รูปที่10.2 แสดงการเลือกชนิดการออกแบบจานรับสัญญาณดาวเทียมแบบCassegrain

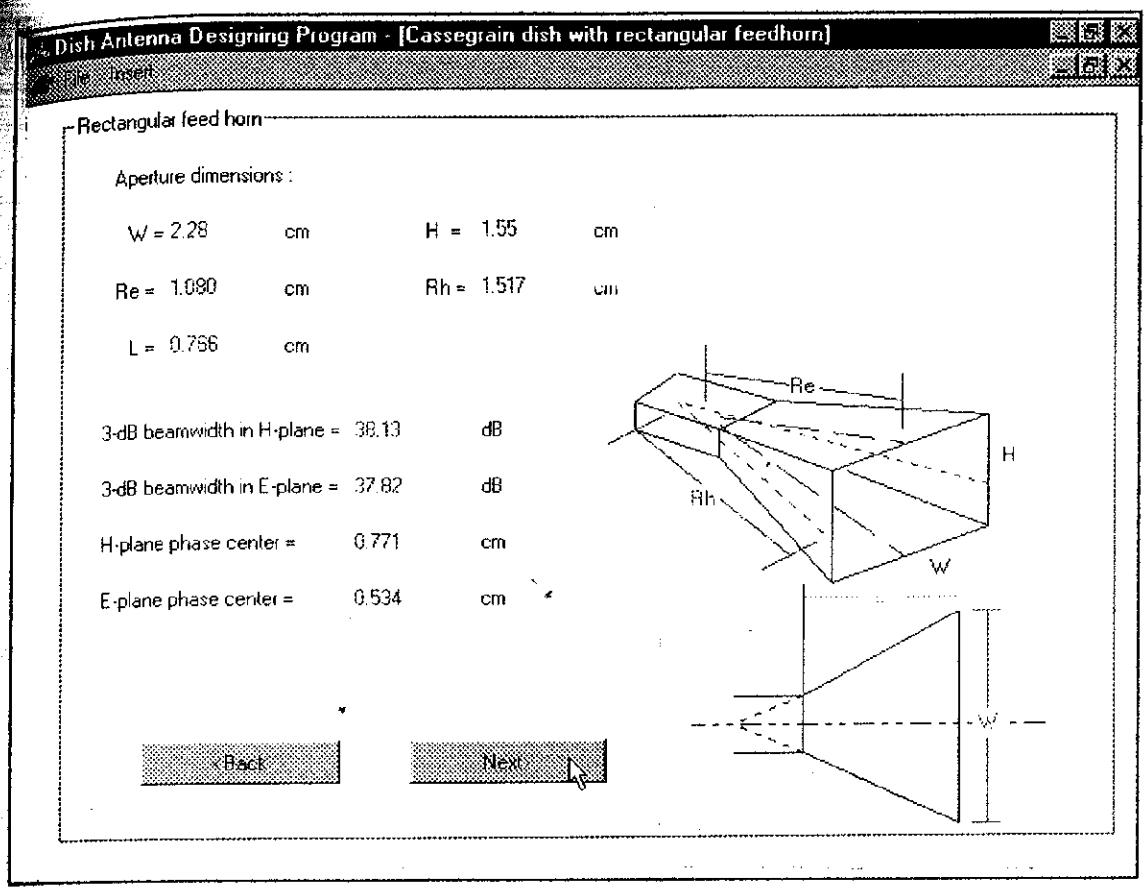
10.1.3 ป้อนข้อมูลที่ใช้ในการออกแบบ



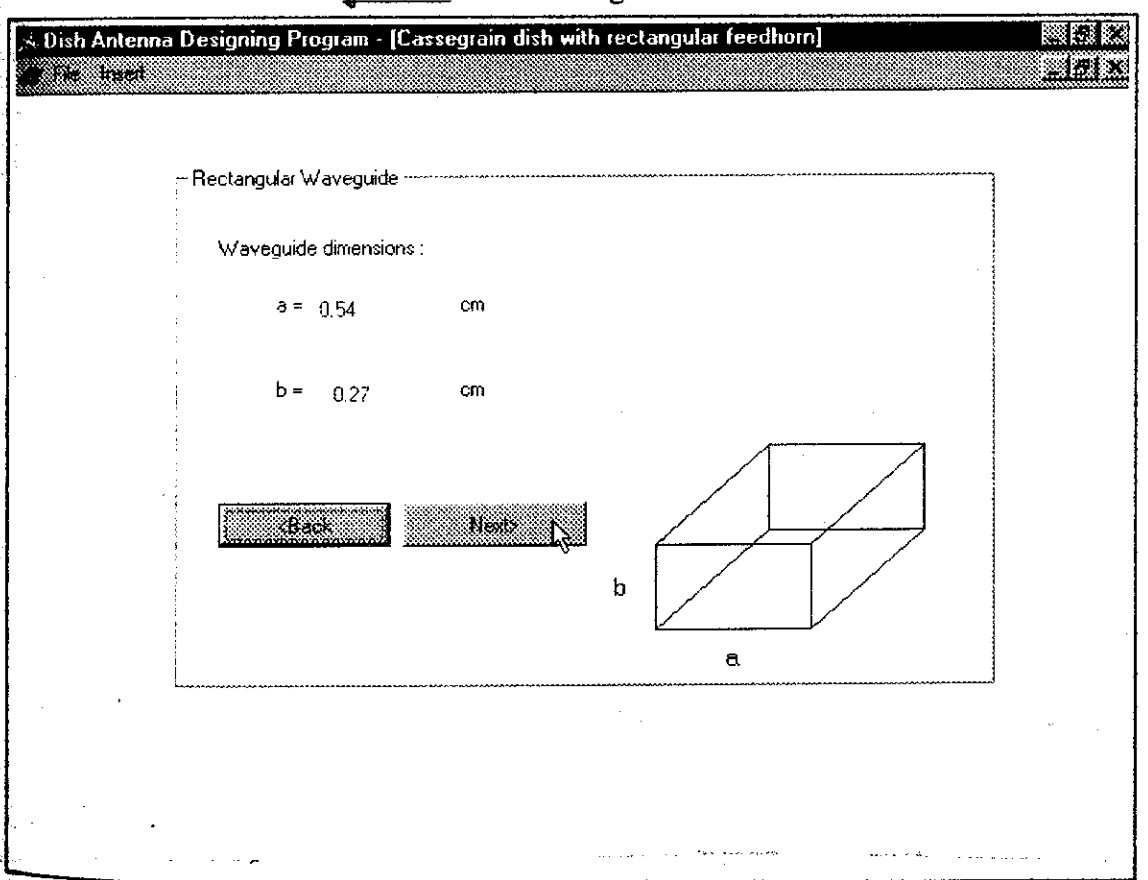
รูปที่10.3 แสดงการป้อนข้อมูลที่ใช้ในการออกแบบ

10.1.4 สามารถเลือกให้โปรแกรมคำนวณค่าฟีดฮอร์นและท่อนำคลื่นในขั้นตอนต่อไป หรือเลือกให้ทำการแก้ไขบีมวิดธ์

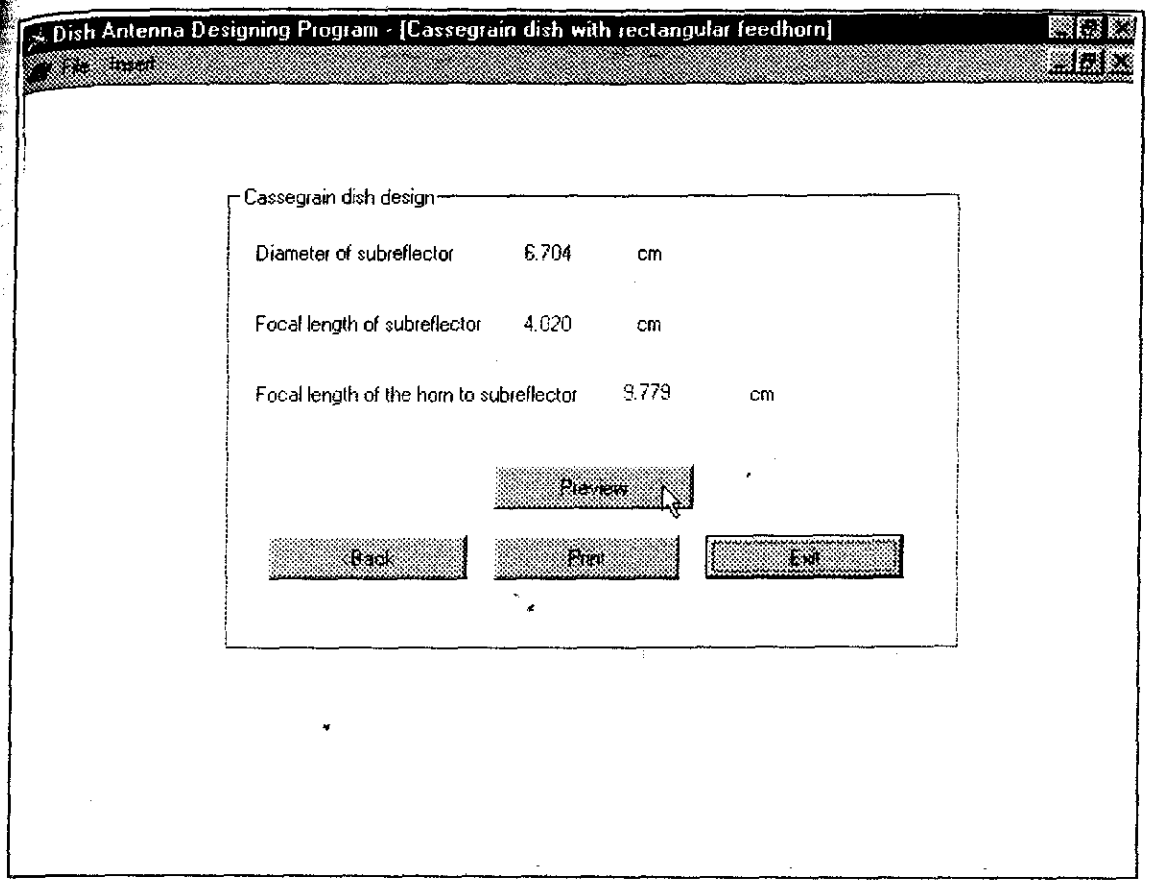
กรณีให้ทำต่อไป



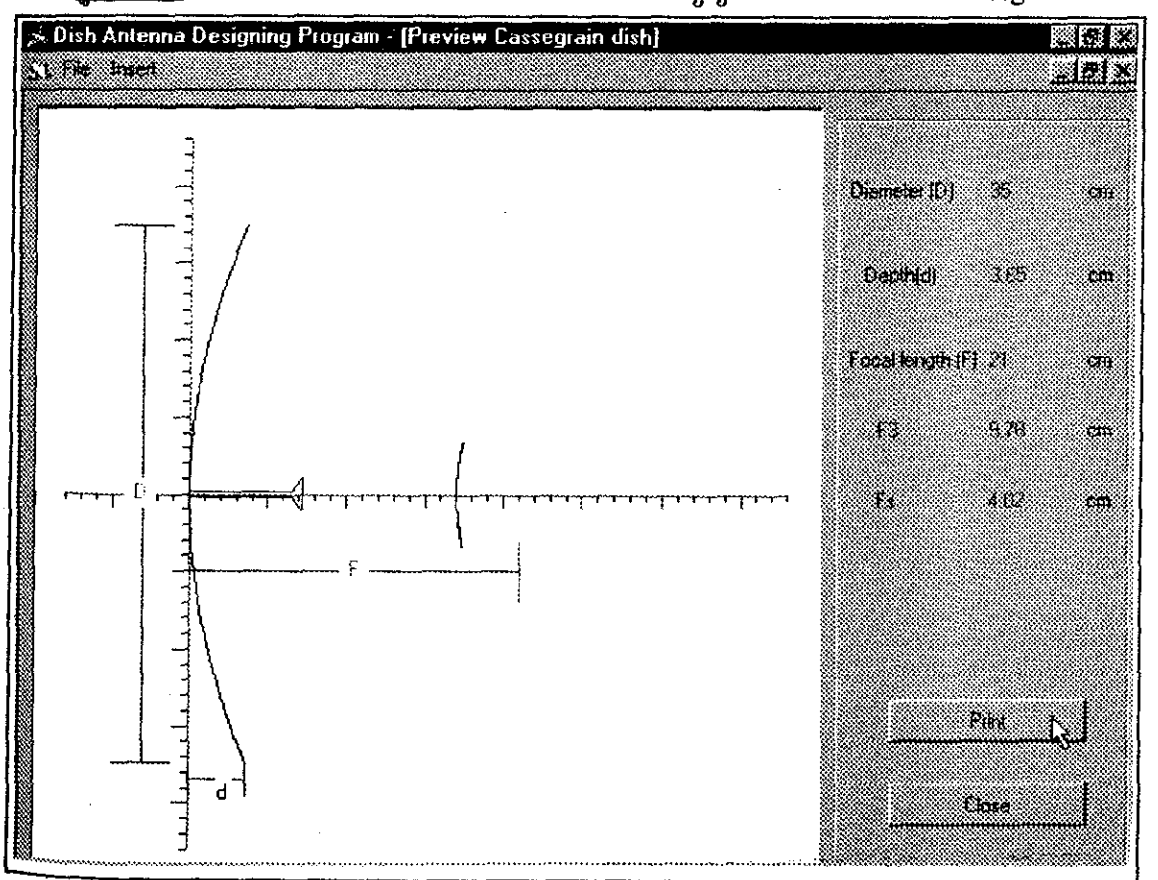
รูปที่10.4 แสดงRectangular feedhorn



รูปที่10.5 แสดง Rectangular waveguide



รูปที่10.6 แสดงระยะของ subreflector บนจานรับสัญญาณดาวเทียมแบบ Cassegrain

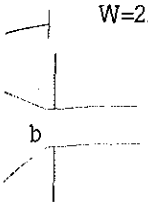


รูปที่10.7 แสดงภาพตัดขวางของจานรับสัญญาณดาวเทียมแบบ Cassegrain

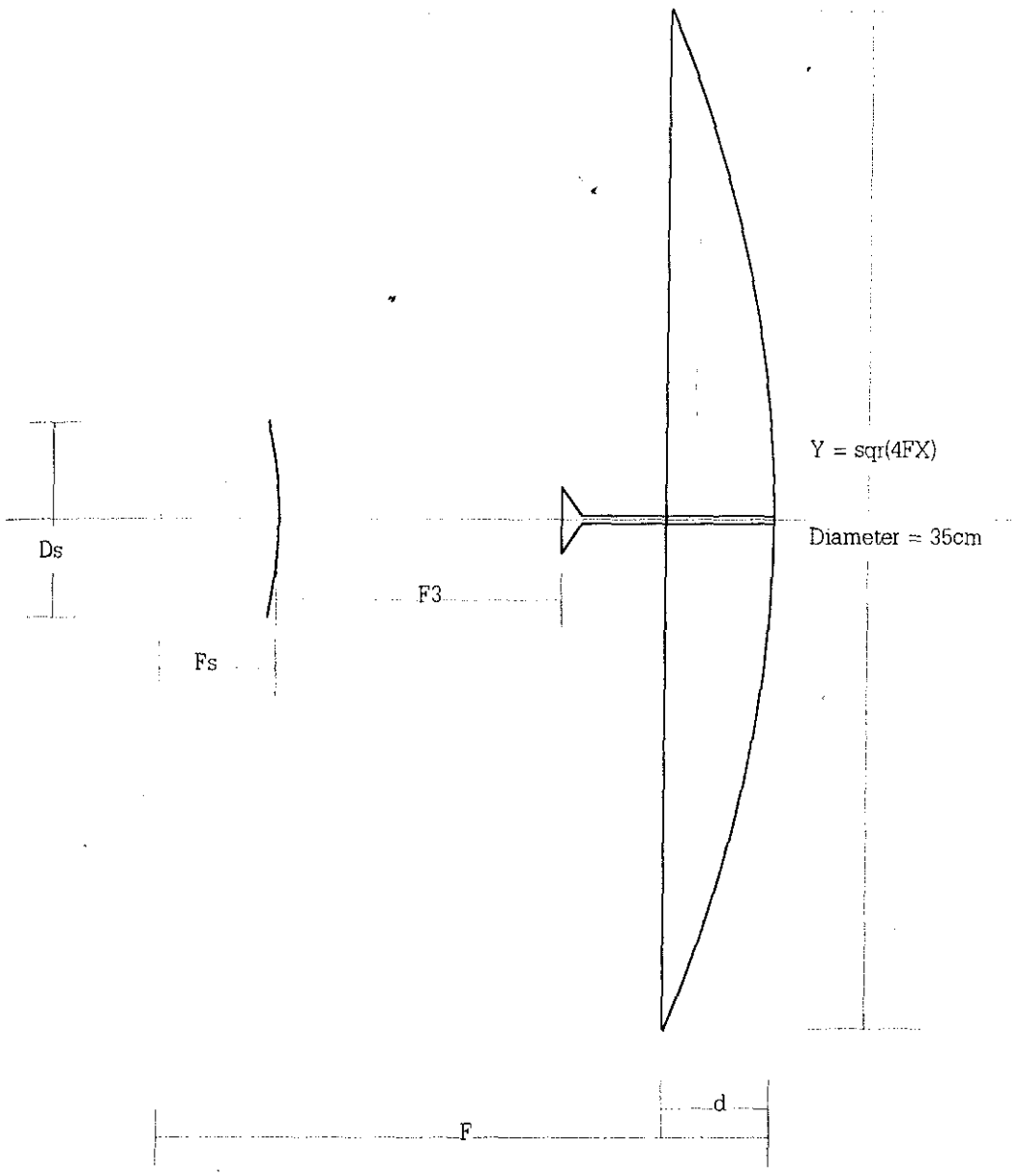
Cassegrain dish with rectangular feedhorn

)n of rectangular feedhorn

W=2.28, L=0.766, b= 0.268 cm



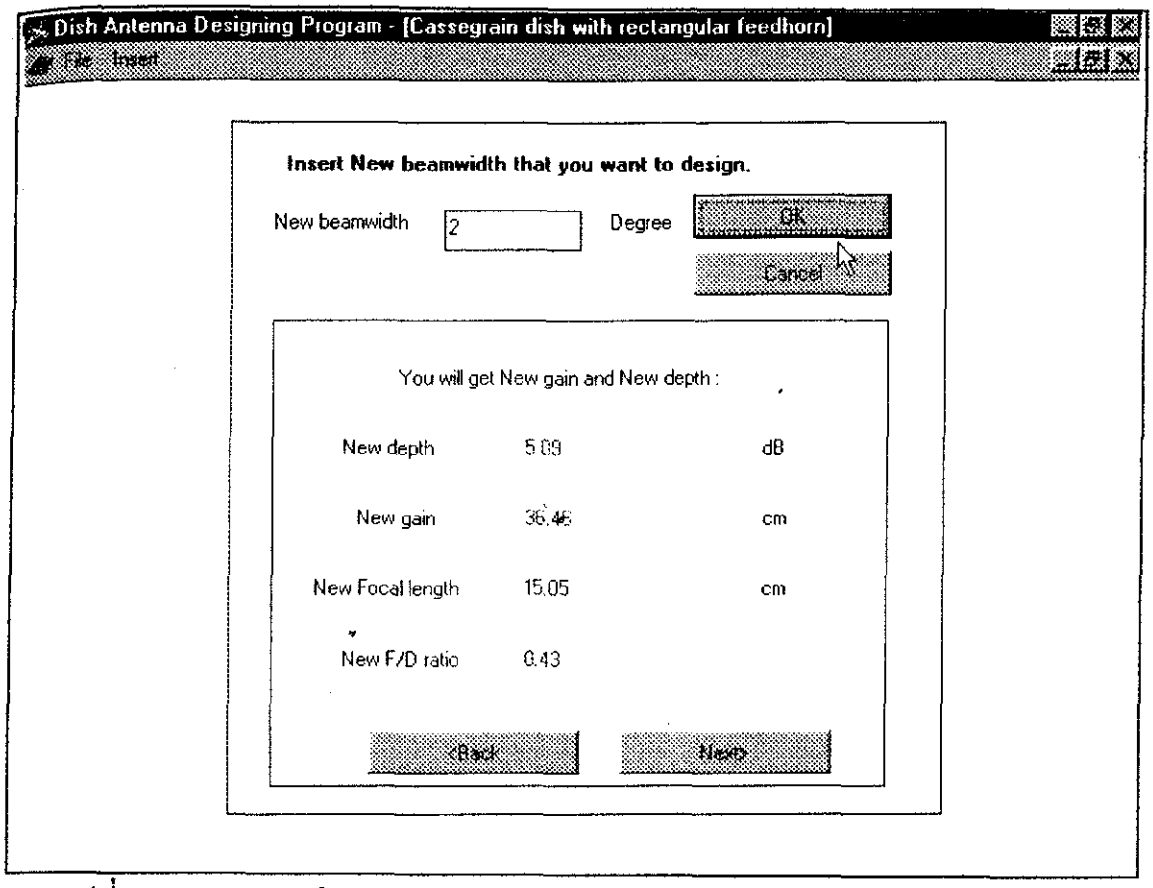
Diameter	35	cm
Depth(d)	3.65	cm
F/D ratio	0.6	
Focal length(F)	21	cm
Ds	6.70	cm
F3	9.78	cm.
Fs	4.02	cm



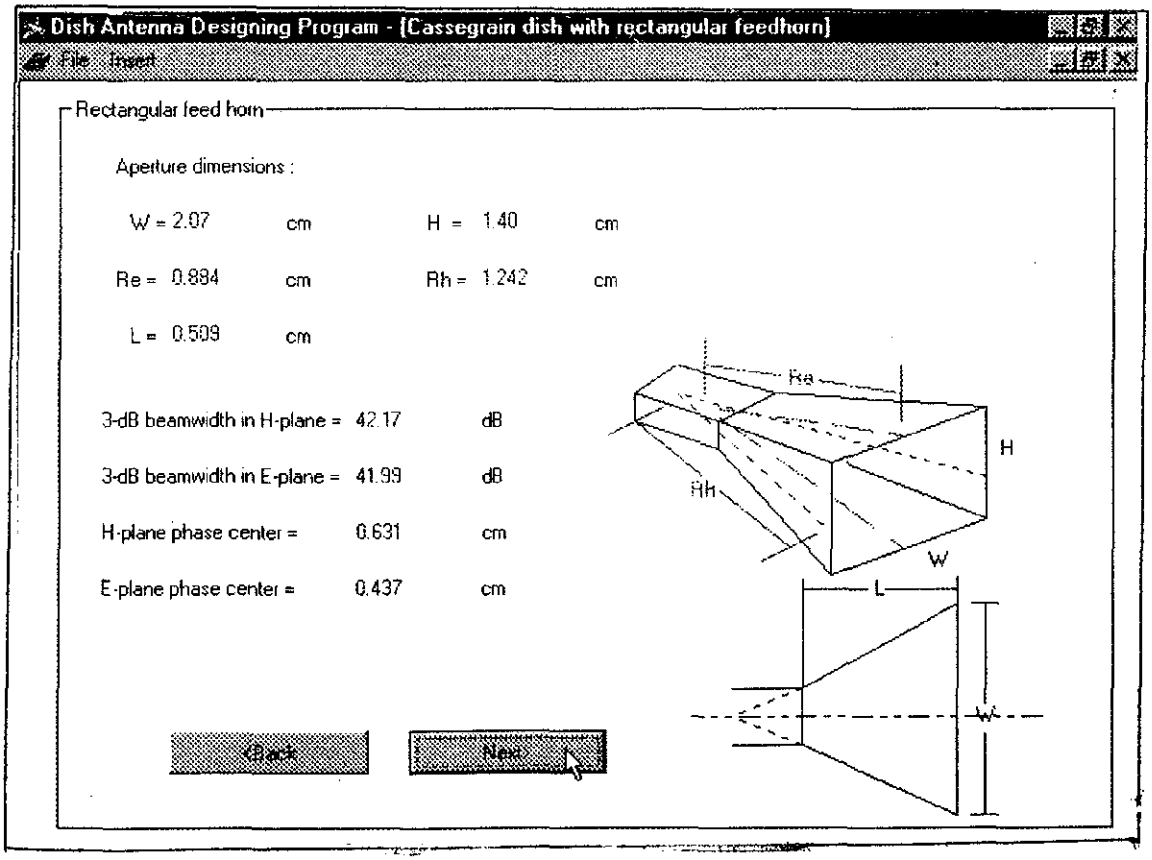
Suranaree University of Technology school of Telecommunication Engineering

y:	Title: Cassegrain dish	Date: 22/04/1997	Scale: 1:2.5

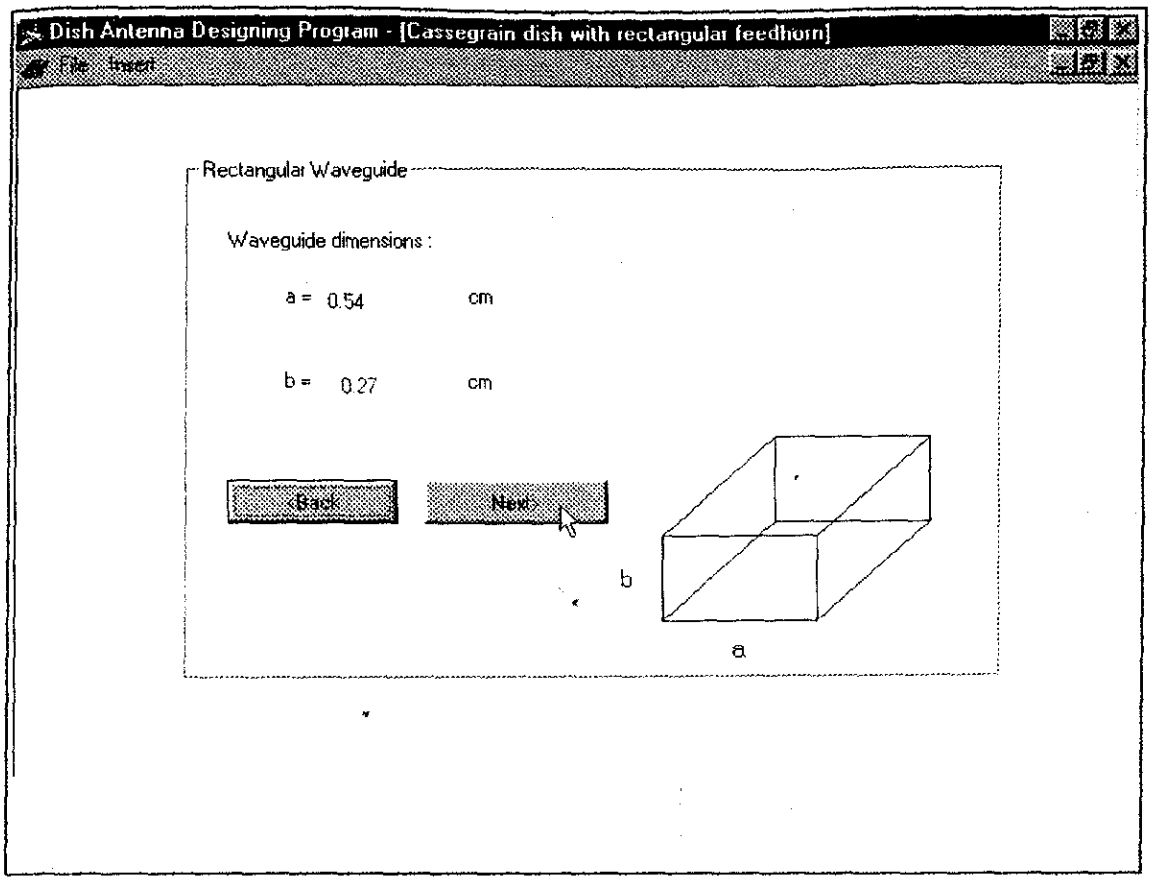
กรณีแก้ไขขนาดของบีมวิดธ์



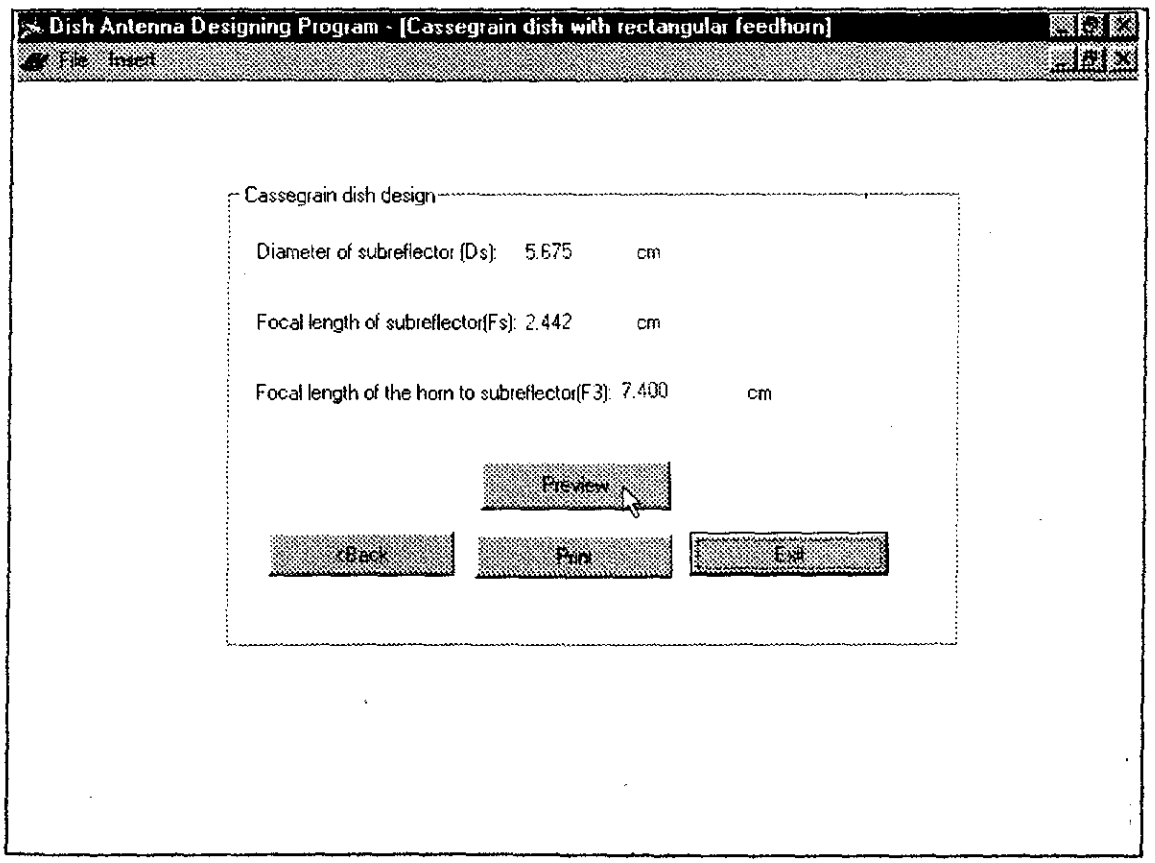
รูปที่10.8 แสดงการแก้ไขขนาดของบีมวิดธ์ของจานรับสัญญาณดาวเทียมแบบ Cassegrain



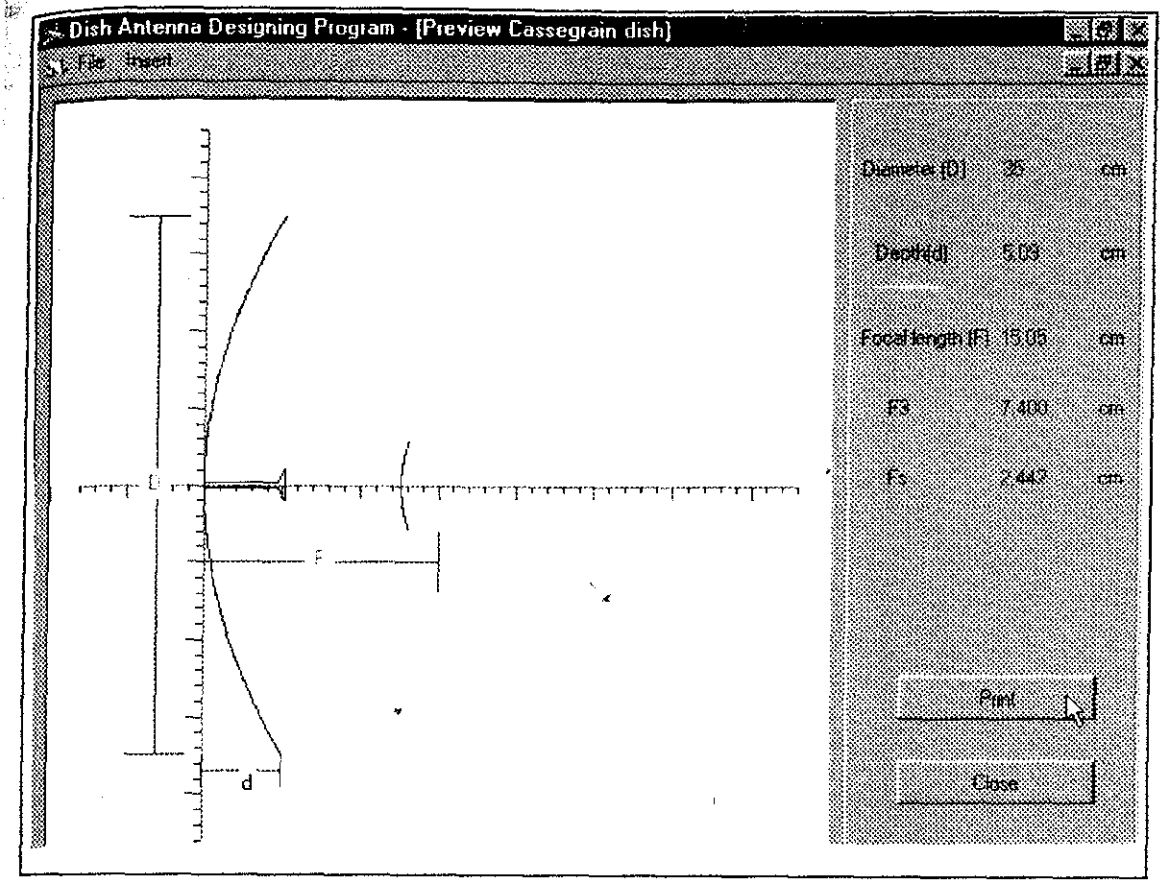
รูปที่10.9 แสดงRectangular feedhorn



รูปที่10.10 แสดง Rectangular Waveguide



รูปที่10.11 แสดงระยะของ sub-reflector บนจานรับสัญญาณดาวเทียมแบบ Cassegrain



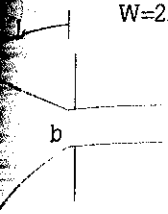
รูปที่10.12 แสดงภาพตัดขวางของจานรับสัญญาณดาวเทียมแบบ Cassegrain

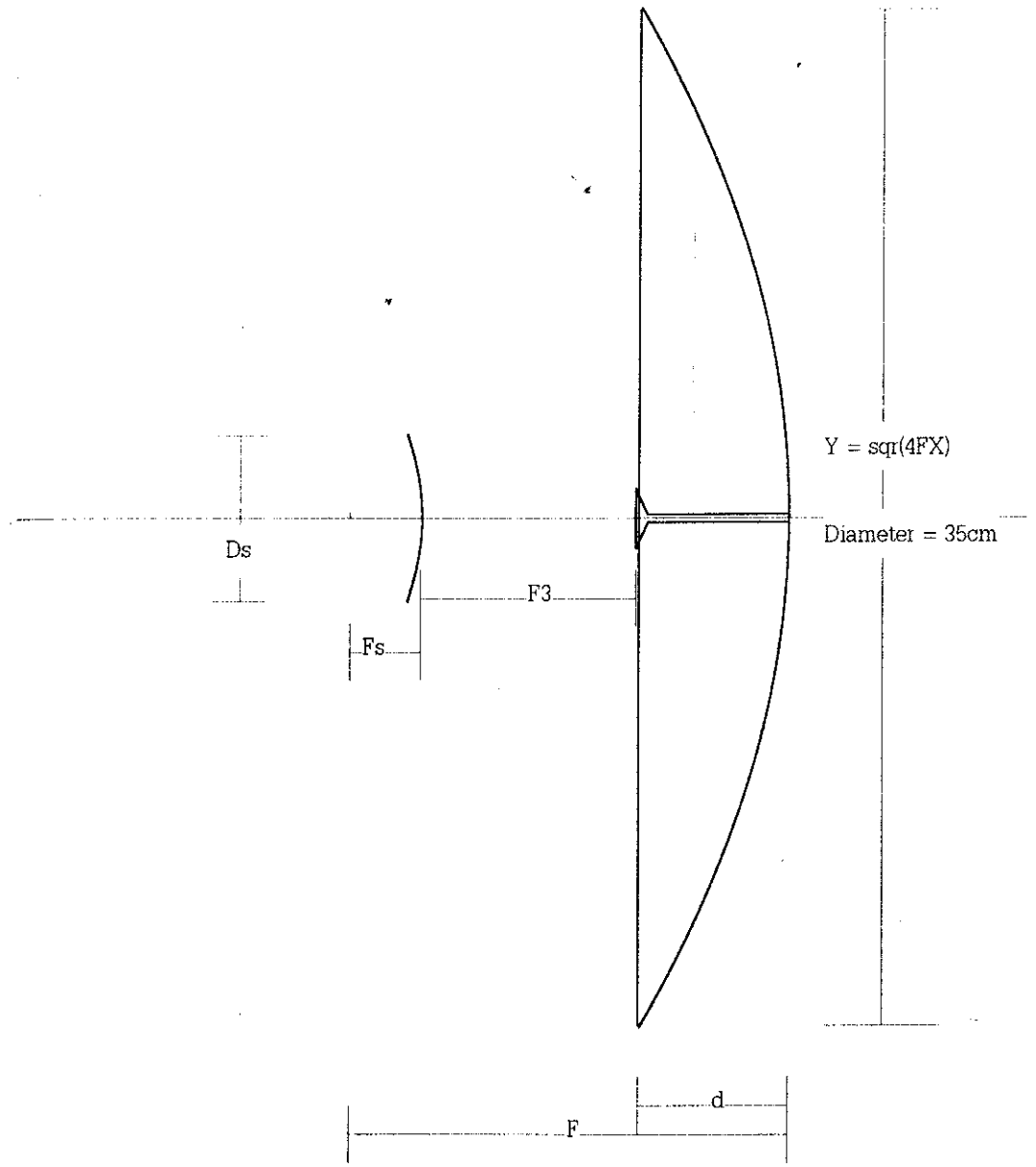
Cassegrain dish with rectangular feedhorn

Diameter	35	cm
Depth(d)	5.09	cm
F/D ratio	0.43	
Focal length(F)	15.05	cm
Ds	5.68	cm
F3	7.400	cm, Fs 2.442 cm

...on of rectangular feedhorn

W=2.07, L=0.509, b= 0.268 cm



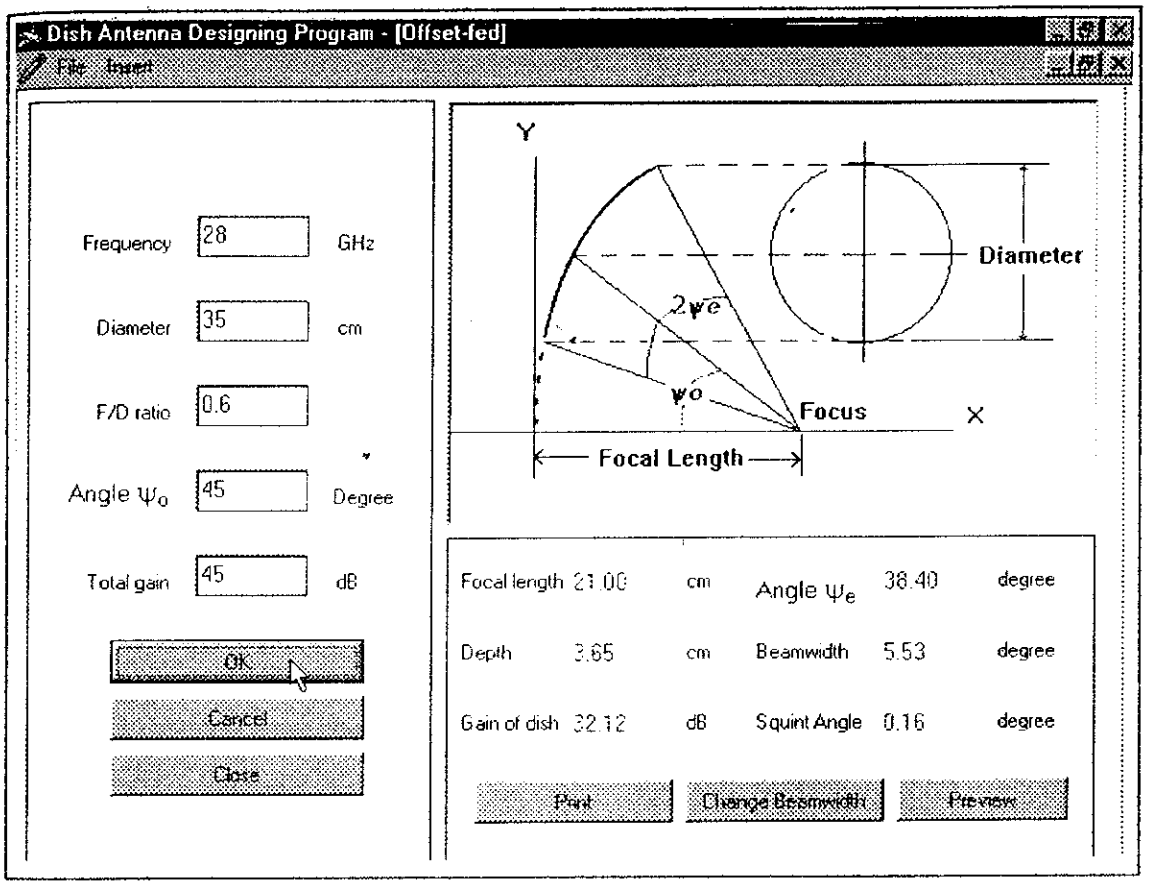


Suranaree University of Technology school of Telecommunication Engineering

...:	Title: Cassegrain dish	Date: 22/04/1997	Scale: 1:2.5

10.2 การออกแบบจานรับสัญญาณดาวเทียมแบบ Offset-fed

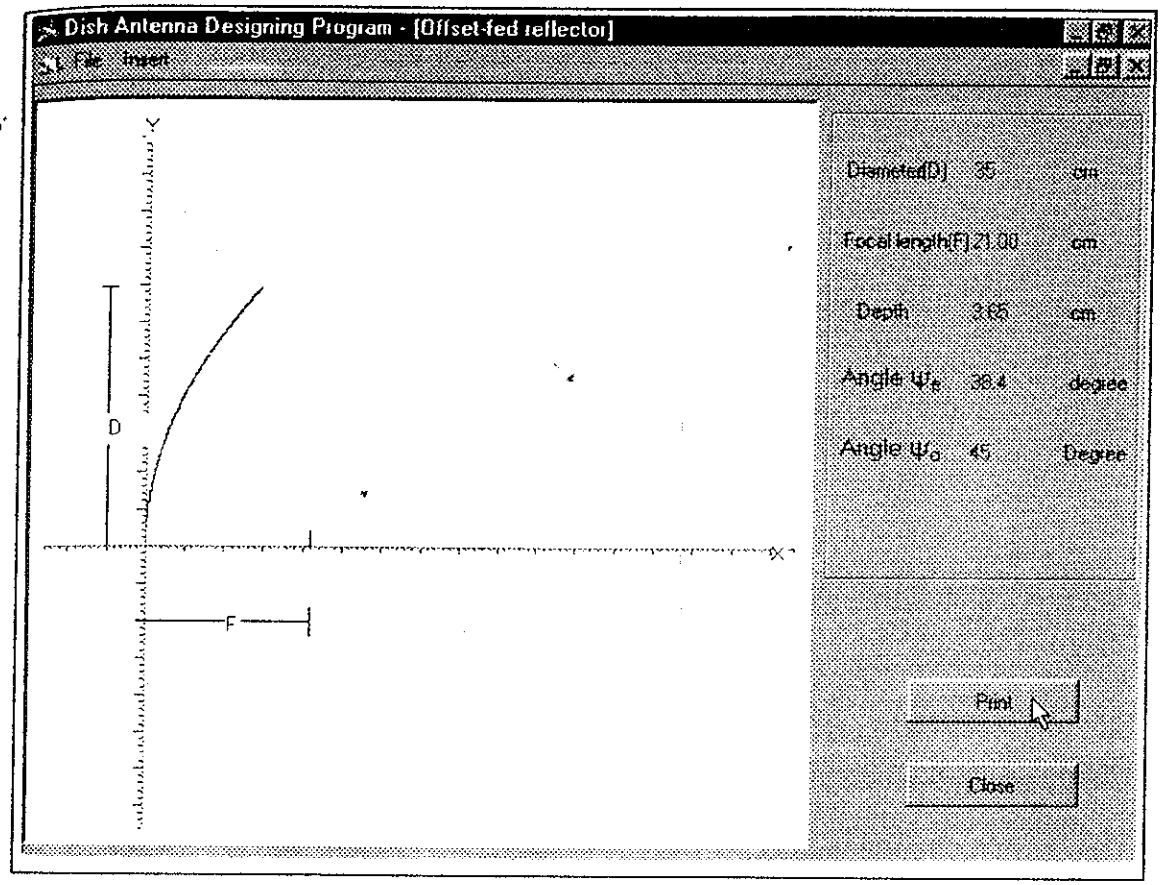
10.2.1 เลือก File Open และเลือกการออกแบบจานรับสัญญาณดาวเทียมแบบ Offset-fed และป้อนข้อมูลที่ใช้ในการออกแบบ



รูปที่ 10.13 แสดงการป้อนข้อมูลในการออกแบบจานรับสัญญาณดาวเทียมแบบ Offset-fed

10.2.2 สามารถเลือกเปลี่ยนขนาดของบีมวิดธ์ตามที่ต้องการได้หรือให้แสดงภาพตัดของจานรับสัญญาณดาวเทียม

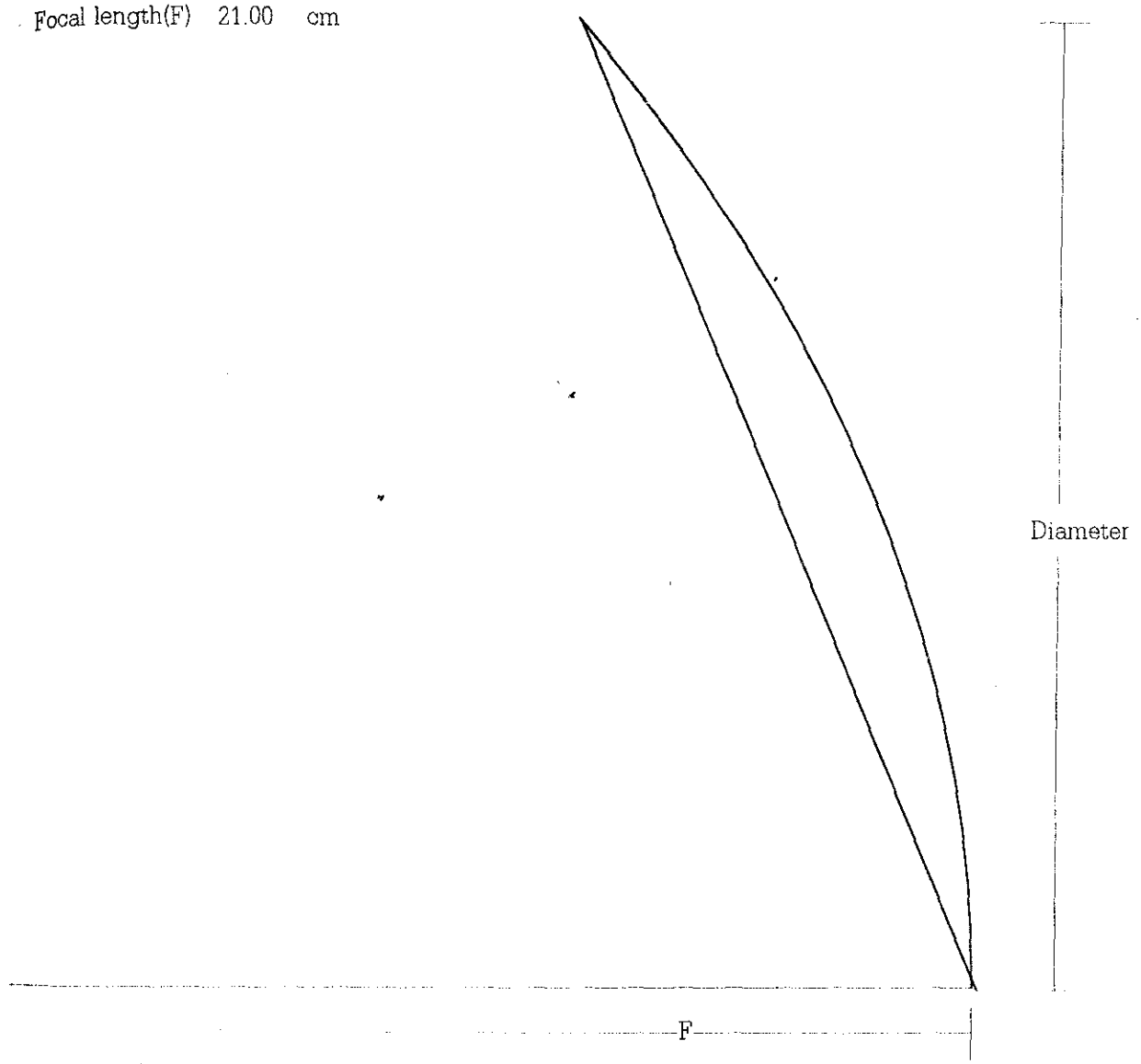
กรณีการเลือกให้แสดงภาพตัดของจานรับสัญญาณดาวเทียม



<u>รูปที่10.14</u> แสดงภาพตัดของจานรับสัญญาณดาวเทียมแบบ Offset-fed

Offset-fed reflector

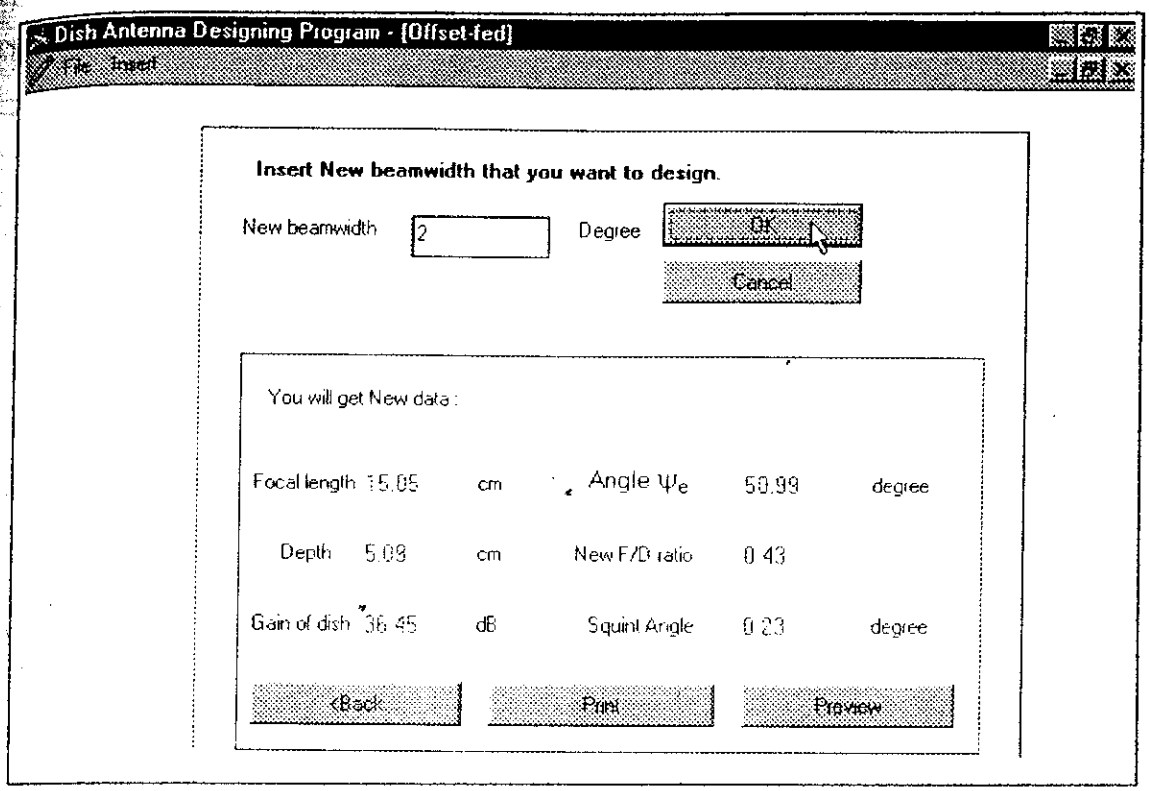
Diameter	35	cm
Depth(d)	3.65	cm
F/D ratio	0.6	
Focal length(F)	21.00	cm



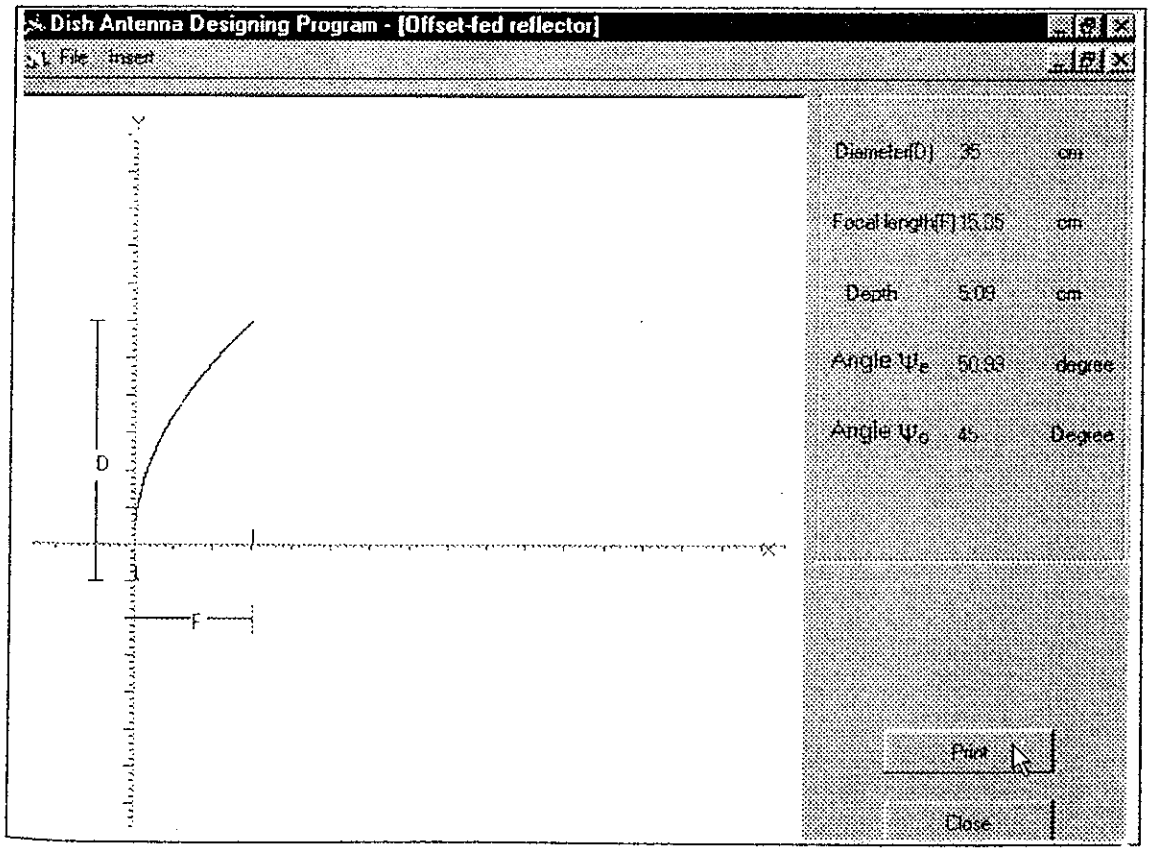
Suranaree University of Technology school of Telecommunication Engineering

By:	Title: Offset-fed reflector	Date: 22/04/1997	Scale: 1:2.5

กรณีการเลือกการแก้ไขบีมวิดธ์



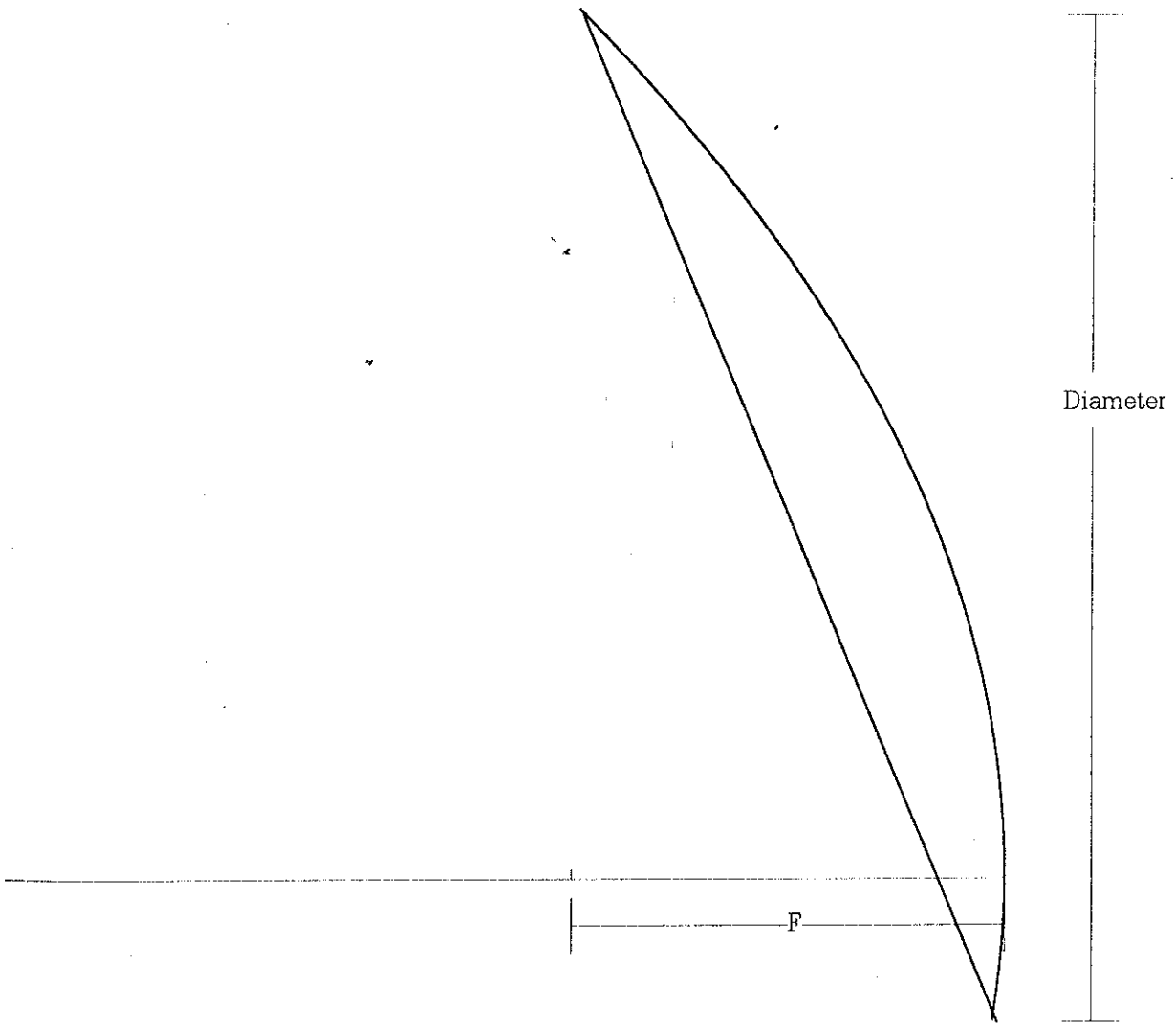
รูปที่10.15 แสดงการแก้ไขขนาดของบีมวิดธ์ของจานรับสัญญาณดาวเทียมแบบ Offset-fed



รูปที่10.16 แสดงภาพตัดของจานรับสัญญาณดาวเทียมแบบ Offset-fedที่มีการแก้ไขบีมวิดธ์

Offset-fed reflector

Diameter	35	cm
Depth(d)	5.09	cm
F/D ratio	0.43	
Focal length(F)	15.05	cm



Suranaree University of Technology school of Telecommunication Engineering

by:	Title: Offset-fed reflector	Date: 22/04/1997	Scale: 1:2.5

จากผลการดำเนินงานจะได้โปรแกรมออกแบบจานรับสัญญาณดาวเทียมสำเร็จรูปที่สามารถติดตั้งได้ทันที แม้ว่าเครื่องคอมพิวเตอร์นั้นจะไม่ได้ติดตั้งโปรแกรม Microsoft Visual Basic 4.0 อยู่ก่อนก็ตาม โปรแกรมสามารถออกแบบจานได้หลากหลายขนาด ตามที่มีใช้อยู่จริงในปัจจุบัน โดยสามารถปรับเปลี่ยนขนาดของบีมวิดธ์ได้ตามขนาดที่ต้องการ และยังช่วยในการออกแบบอุปกรณ์ที่ใช้ร่วมกับจานรับสัญญาณดาวเทียมได้แก่ ฟีดฮอร์น และ ท่อนำคลื่น ทั้งนี้ส่วนประกอบต่างๆสามารถที่จะแสดงเป็นแบบออกทางเครื่องพิมพ์ได้

10.3 วิธีการนำแบบไปใช้งาน

10.3.1 โปรแกรมออกแบบจานรับสัญญาณดาวเทียมจะช่วยให้การนำแบบที่ได้ไปใช้งานจริงง่ายยิ่งขึ้นเพราะได้แสดงอัตราส่วนของแบบกับขนาดจริงที่จะนำไปสร้างใช้งาน เช่น อัตราส่วน 1 : 2.5 หรือ 1: 10 ขึ้นกับขนาดจานว่ามีขนาดเล็กหรือใหญ่ โดยแบบที่ได้จะมีขนาดที่เหมาะสมอยู่ในกระดาษ A4

10.3.2 แบบที่ได้จะแสดงกริดซึ่งสามารถนำไปขยายเพื่อทำแบบเท่าจริงได้ จะช่วยให้การเห็นระยะของอัตราส่วนที่ชัดเจนยิ่งขึ้น

10.3.3 กริดในแต่ละช่องจะมีขนาด 1 × 1 ตารางเซ็นติเมตร เมื่อนำกริดในแต่ละช่องมาขยายเท่าขนาดจริงจะได้ชิ้นส่วนของจานพาราโบลิกด้วย ถ้านำกริดในทุกๆช่องมาขยายและนำมาต่อเป็นรูปจานรับสัญญาณดาวเทียม ก็จะได้แบบเท่าจริงของจานรับสัญญาณดาวเทียม

10.3.4 โปรแกรมออกแบบจานรับสัญญาณดาวเทียมจะแสดงระยะของส่วนประกอบต่างๆ เช่น ระยะของ sub reflecter เมื่อมีอุปกรณ์ครบถ้วน สามารถที่จะประกอบเพื่อนำไปใช้งานได้ทันที

บรรณานุกรม

รังสรรค์ วงศ์สรรค์, 2536, "จานรับสัญญาณดาวเทียม",โลกของการรับสัญญาณโทรทัศน์ผ่านดาวเทียม, สำนักพิมพ์ นิตยสารซีคิว, กรุงเทพฯ.

ประสิทธิ์ ทีฆพุฒิ, 2536, "จานสายอากาศ", การสื่อสารดาวเทียม, วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, กรุงเทพฯ.

John D. Kraus,1988, "Reflector Antennas and Their feed systems", Antennas Second Edition, McGraw-Hill Book Company, NewYork U.S.A.

Schoenbeck. J., Robert ., Electronic Communication Modulation and Transmission Seccond Edition, Macmillan Publishing Company (USA.) 1992,

Constantine A. Balanis,1982, "Reflector and Lens Antennas", Antenna Theory Analysis and Design, John Wiley & Son, Inc.

Thomas A. Milligan,1985, "Radiation from Apertures", Modern Antenna Design, McGraw-Hill Book Company, NewYork U.S.A.

Thomas A. Milligan,1985, "Horn Antennas", Modern Antenna Design, McGraw-Hill Book Company, NewYork U.S.A.

Thomas A. Milligan,1985, "Reflector Antennas", Modern Antenna Design, McGraw-Hill Book Company, NewYork U.S.A.

ภาคผนวก

```
Private Sub Command1_Click()
End Sub

Private Sub cmdCas_C_Click()
frmCas_C.Show
frmChoice.Visible = False
End Sub

Private Sub cmdCasR_Click()
frmCasR.Show
frmChoice.Visible = False
End Sub

Private Sub Form_Load()
Frame1.BackColor = vbActiveBorder
frmChoice.BackColor = vbActiveBorder
Label1.BackColor = vbActiveBorder
End Sub

Private Sub Form_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Form_Load()
frmAbout.BackColor = vbActiveBorder
Frame1.BackColor = vbActiveBorder
Label1.BackColor = vbActiveBorder
Label2.BackColor = vbActiveBorder
Label5.BackColor = vbActiveBorder
Label6.BackColor = vbActiveBorder
Label3.BackColor = vbActiveBorder
End Sub

Private Sub Frame1_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label1_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label2_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label3_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label4_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label5_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Label6_Click()
frmAbout.Visible = False
MDIDesign.Visible = True
End Sub

Private Sub Picture1_Click()
End Sub

Private Sub Timer2_Timer()
Image2.Visible = False
Image1.Visible = True
Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
Image1.Visible = False
```

```
Image2.Visible = True
Timer2.Enabled = True
End Sub

Private Sub cmdBack_Click()
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdBack2_2_Click()
fraCF2.Visible = False
fraNewbeam.Visible = True
End Sub

Private Sub cmdBack2_Click()
fraCF.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdBack3_2_Click()
fraCW2.Visible = False
fraCF2.Visible = True
End Sub

Private Sub cmdBack3_Click()
fraCW.Visible = False
fraCF.Visible = True
End Sub

Private Sub cmdBack4_2_Click()
fraCasse2.Visible = False
fraCW2.Visible = True
End Sub

Private Sub cmdBack4_Click()
fraCasse.Visible = False
fraCW.Visible = True
End Sub

Private Sub cmdBack1_Click()
End Sub

Private Sub cmdCancel_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
End Sub

Private Sub cmdClear_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
txtTTG.Locked = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNBW.Locked = False
LabelD1 = ""
End Sub

Private Sub cmdExit_Click()
frmChoise.Show
frmCas_C.Visible = False
End Sub

Private Sub cmdExit2_2_Click()
Unload frmCas_C
End Sub
```

```
Private Sub cmdExit2_Click()
Unload frmCas_C
End Sub

Private Sub cmdClose1_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
Unload frmCas_C
End Sub

Private Sub cmdNext_Click()
fraPara.Visible = False
fraNewbeam.Visible = True
End Sub

Private Sub cmdNext1_Click()
fraPara.Visible = False
fraCF.Visible = True
Dim TTG, F, WL, Pi, G As Double
Pi = 4 * Atn(1)
TTG = txtTTG.Text
F = txtFreq.Text
WL = txtWL.Text
G = txtG.Text
  If (TTG - G) <= 12 Then
  GF = 12
  Else
  GF = TTG - G
  End If
  txtDF.Text = (WL / Pi) * 10 ^ ((GF + 2.85) / 20)
  txtDF.Text = Format(txtDF.Text, "##0.00")
DF = txtDF.Text
  txtSR.Text = DF ^ 2 / (8 * WL * 0.39)
  txtSR.Text = Format(txtSR.Text, "##0.00")

Dim X, Y As Single
X = (2.088 * WL / (Pi * DF))
txtdBH.Text = 2 * (Atn(X / Sqr(-X * X + 1))) * 180 / Pi
Y = (1.782 * WL / (Pi * DF))
txtdBE.Text = 2 * (Atn(Y / Sqr(-Y * Y + 1))) * 180 / Pi
txtdBH.Text = Format$(txtdBH.Text, "##0.00")
txtdBE.Text = Format$(txtdBE.Text, "##0.00")

SR = txtSR.Text
txtHPC.Text = 0.471 * SR
txtEPC.Text = 0.807 * SR
txtHPC.Text = Format$(txtHPC.Text, "##0.00")
txtEPC.Text = Format$(txtEPC.Text, "##0.00")
End Sub

Private Sub cmdNext2_Click()
Dim TTG, F, WL, Pi, NG As Double
Pi = 4 * Atn(1)
TTG = txtTTG.Text
F = txtFreq.Text
WL = txtWL.Text
NG = txtNG.Text
  If (TTG - NG) <= 12 Then
  GF = 12
  Else
  GF = TTG - NG
  End If
  txtDF2.Text = (WL / Pi) * 10 ^ ((GF + 2.85) / 20)
  txtDF2.Text = Format(txtDF2.Text, "##0.00")
DF2 = txtDF2.Text
```

```
DF2 = Format(DF2, "##0.00")
    txtSR2.Text = DF2 ^ 2 / (8 * WL * 0.39)
    txtSR2.Text = Format(txtSR2.Text, "##0.00")
Dim X, Y As Single
X = (2.088 * WL / (Pi * DF2))
txtdBH2.Text = 2 * (Atn(X / Sqr(-X * X + 1))) * 180 / Pi
Y = (1.782 * WL / (Pi * DF2))
txtdBE2.Text = 2 * (Atn(Y / Sqr(-Y * Y + 1))) * 180 / Pi
txtdBH2.Text = Format(txtdBH2.Text, "##0.00")
txtdBE2.Text = Format(txtdBE2.Text, "##0.00")
SR2 = txtSR2.Text
SR2 = Format(SR2, "##0.00")
txtHPC2.Text = 0.471 * SR2
txtEPC2.Text = 0.807 * SR2
txtHPC2.Text = Format$(txtHPC2.Text, "##0.00")
txtEPC2.Text = Format$(txtEPC2.Text, "##0.00")
fraNewbeam.Visible = False
fraCF2.Visible = True
End Sub

Private Sub cmdNext3_2_Click()
fraCW2.Visible = True
fraCF2.Visible = False
Dim Pi, WL As Double
Pi = 4 * Atn(1)
WL = txtWL.Text
    txtR2.Text = (WL * 1.84) / (2 * Pi)
    txtR2.Text = Format(txtR2.Text, "##0.000")
    txtDw2.Text = (WL * 1.84) / Pi
    txtDw2.Text = Format(txtDw2.Text, "##0.000")
Dw2 = txtDw2.Text
Dw2 = Format(Dw2, "##0.000")
R2 = txtR2.Text
R2 = Format(R2, "##0.00")
End Sub

Private Sub cmdNext3_Click()
fraCW.Visible = True
fraCF.Visible = False
Dim Pi, WL As Double
Pi = 4 * Atn(1)
WL = txtWL.Text
    txtR.Text = (WL * 1.84) / (2 * Pi)
    txtR.Text = Format(txtR.Text, "##0.000")
    txtDw.Text = (WL * 1.84) / Pi
    txtDw.Text = Format(txtDw.Text, "##0.000")
Dw = txtDw.Text
Dw = Format(Dw, "##0.000")
R = txtR.Text
R = Format(R, "##0.00")
    End Sub

Private Sub cmdNext4_2_Click()
fraCasse2.Visible = True
fraCW2.Visible = False
Dim dia, NFD, NFC, ND, WL As Double
dia = txtDia.Text
NFD = txtNFD.Text
NFC = txtNFC.Text
ND = txtND.Text
WL = txtWL.Text

    txtDs2.Text = Sqr(2 * WL * NFC)
    txtDs2.Text = Format(txtDs2.Text, "##0.000")
Ds2 = txtDs2.Text
Ds2 = Format(Ds2, "##0.00")
    txtFs2.Text = Ds2 * NFC / dia
    txtFs2.Text = Format(txtFs2.Text, "##0.000")
Fs2 = Format(txtFs2.Text, "##0.00")
Dim dBH2, Pi As Double
dBH2 = txtdBH2.Text
Pi = 4 * Atn(1)
    txtF32.Text = Ds2 / (2 * Tan(dBH2 * Pi / 360))
    txtF32.Text = Format(txtF32.Text, "##0.000")
F32 = Format(txtF32.Text, "##0.00")
LL = Sqr((SR2 ^ 2) - (DF2 / 2 ^ 2))
LL = Format(LL, "##0.00")
Ls = (Dw2 / DF2) * LL
End Sub

Private Sub cmdNext4_Click()
fraCasse.Visible = True
```

```
fraCW.Visible = False
dia = txtDia.Text
FD = txtFD.Text
FC = txtFC.Text
d = txtD.Text
WL = txtWL.Text

   txtDs.Text = Sqr(2 * WL * FC)
   txtDs.Text = Format$(txtDs.Text, "##0.000")
Ds = Format(txtDs.Text, "##0.00")
   txtFs.Text = Ds * FC / dia
   txtFs.Text = Format$(txtFs.Text, "##0.000")
Fs = Format(txtFs, "##0.00")
Dim dBH, Pi As Double
dBH = txtdBH.Text
Pi = 4 * Atn(1)
   txtF3.Text = Ds / (2 * Tan(dBH * Pi / 360))
   txtF3.Text = Format$(txtF3.Text, "##0.000")
F3 = Format(txtF3.Text, "##0.00")
LL = Sqr((SR ^ 2) - (DF / 2 ^ 2))
LL = Format(LL, "##0.00")
Ls = (Dw / DF) * LL
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4 GHz"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diameter data."), (0), ("Invalid
data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diameter."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0), ("Invalid
data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFD.Text = ""
ElseIf txtTTG.Text = "" Then
MsgBox ("If you don't insert Total Gain, you will get
minimum gain of feedhorn."), (0), ("Invalid data")
txtTTG.Text = "0"
Again:
Msg = "The data error.Try again!"   ' Define message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
cmdClear.TabStop = True
cmdClear.TabIndex = 1
Else
txtFreq.Text = Format(txtFreq.Text, "##0.00")
txtDia.Text = Format(txtDia.Text, "##0.00")
txtFD.Text = Format(txtFD.Text, "##0.00")
Dim F, FD, TTG As Double
F = txtFreq.Text
dia = txtDia.Text
FD = txtFD.Text
TTG = txtTTG.Text
   txtWL.Text = 30 / F
   txtWL.Text = Format(txtWL.Text, "##0.00")
   txtFC.Text = FD * dia
   txtFC.Text = Format(txtFC.Text, "##0.00")
Dim WL As Double
FC = txtFC.Text
WL = txtWL.Text
   txtD.Text = dia ^ 2 / (16 * FC)
```

```
    txtD.Text = Format(txtD.Text, "##0.00")
Dim d, Pi As Double
d = txtD.Text
Pi = 4 * Atn(1)
    txtG.Text = 10 * Log(2 + (d * Pi ^ 2 * dia ^ 2 * d ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
    txtG.Text = Format$(txtG.Text, "##0.00")
    txtBW.Text = 2 * Atn(WL * FC / (d ^ 2 * dia)) * (180 /
Pi)
    txtBW.Text = Format$(txtBW.Text, "##0.00")
txtFreq.Locked = True
txtDia.Locked = True
txtFD.Locked = True
txtTTG.Locked = True
LabelD1 = dia

End If
End Sub

Private Sub cmdAccept_Click()
If txtFreq.Text = "" Or txtDia.Text = "" Or txtFD.Text = ""
Or txtTTG.Text = "" Then
MsgBox ("Please insert all value."), (0), ("Error value")
ElseIf txtFreq.Text = "0" Or txtDia.Text = "0" Or
txtFD.Text = "0" Or txtTTG.Text = "0" Then
MsgBox ("Value can't be zero"), (0), ("Error value")
Else
cmdAccept.Enabled = False
cmdOK.Enabled = True
End If
End Sub


Private Sub cmdOK0_Click()
If txtFreq.Text = "" Then
MsgBox ("You must insert frequency data."), (0), ("Invalid
data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert diameter data."), (0), ("Invalid
data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0), ("Invalid
data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtTTG.Text = "" Then
MsgBox ("If you don't insert total gain,this program will
use minimum total gain."), (0), ("Total gain")
txtTTG.Text = 0
Else
Dim F, dia, FD As Double
F = txtFreq.Text
dia = txtDia.Text
FD = txtFD.Text
    txtWL.Text = 30 / F
    txtWL.Text = Format$(txtWL.Text, "##0.00")
    txtFC.Text = FD * dia
    txtFC.Text = Format$(txtFC.Text, "##0.00")
Dim FC, WL As Double
FC = txtFC.Text
WL = txtWL.Text
    txtD.Text = dia ^ 2 / (16 * FC)
    txtD.Text = Format$(txtD.Text, "##0.00")
Dim d, Pi As Double
d = txtD.Text
Pi = 4 * Atn(1)
 txtG.Text = 10 * Log(2 + (d * Pi ^ 2 * dia ^ 2 * d ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
    txtG.Text = Format$(txtG.Text, "##0.00")
    txtBW.Text = 2 * Atn(WL * FC / (d ^ 2 * dia)) * (180 /
Pi)
    txtBW.Text = Format$(txtBW.Text, "##0.00")
cmdNext1.Enabled = True
txtFreq.Locked = True
txtDia.Locked = True
```

```
txtTTG.Locked = True
txtFD.Locked = True
End If
End Sub

Private Sub cmdOK2_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth data."), (0),
("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0), ("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"  ' Define message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Dim NBW, F, dia, WL, Pi As Double
NBW = txtNBW.Text
NBW = Format(NBW, "##0.00")
F = txtFreq.Text
dia = txtDia.Text
WL = txtWL.Text
Pi = 4 * Atn(1)
   NFD = ((Tan(NBW * Pi / 360)) * dia ^ 2 / (WL * 16 ^
2)) ^ (1 / 3)
NFD = Format(NFD, "##0.00")
   txtNFD.Text = NFD
   NFC = dia * NFD
NFC = Format(NFC, "##0.00")
   txtNFC.Text = NFC
   ND = dia ^ 2 / (16 * NFC)
   txtND.Text = ND
ND = Format(ND, "##0.00")
   NG = 10 * Log(2 + (ND * Pi ^ 2 * dia ^ 2 * ND ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
   txtNG.Text = NG
txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")

txtNBW.Locked = True
LabelD1 = dia
End If
End Sub

Private Sub cmdPrev1_Click()
If dia > 90 Then
frmPrevCC3.Show
Else
frmPrevCC1.Show
End If
End Sub

Private Sub cmdPrev2_Click()
If dia > 90 Then
frmPrevCC4.Show
Else
frmPrevCC2.Show
End If
End Sub

Private Sub Command2_Click()

End Sub

Private Sub cmdPrint1_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.Print , ""
Printer.Print , ""
Printer.Print , "Cassegrain dish with circular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
```

```
Printer.Print , Spc(5); "Give data"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , Spc(5); "Get data"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "focal length"; Spc(7); txtFC.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Depth"; Spc(14); txtD.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Gain"; Spc(15); txtG.Text; Spc(3); "dB"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Beamwidth"; Spc(10); txtBW.Text; Spc(3);
"Degrees"
'---------------
Printer.NewPage
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular feedhorn"   'Topic'
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter of feedhorn(d)"; Spc(5);
txtDF.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Slant radius(R)"; Spc(13); txtSR.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtdBH.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtdBE.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC.Text; Spc(3); "cm"
Printer.Print ""
Printer.Print , ""
'---------------
Printer.NewPage
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular wavguide"   'Topic'
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Radius of wavguide"; Spc(10); txtR.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter of wavguide"; Spc(8);
txtDw.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
```

```
Printer.Print , ""    'enter line'
Printer.Print , "Cassegrain dish"   'Topic'
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(5);
txtDs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(5); txtFs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(5); txtF3.Text; Spc(3); "cm"
Printer.EndDoc


End Sub

Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish with circular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New beamwidth"; Spc(6); txtNBW.Text;
Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New depth"; Spc(10); txtND.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New gain"; Spc(11); txtNG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New focal length"; Spc(3); txtNFC.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New F/D ratio"; Spc(6); txtNFD.Text


'---------------
Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , ""    'enter line'
```

```
Printer.Print , ""    'enter line'
Printer.Print , "Circular feedhorn"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print ,
Printer.Print , "Diameter of feedhorn(d)"; Spc(5);
txtDF2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Slant radius(R)"; Spc(13); txtSR2.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtdBH2.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtdBE2.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC2.Text; Spc(3); "cm"
Printer.Print , ""
Printer.Print , ""

'---------------
Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.Print , "Circular wavguide"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print ,
Printer.Print , "Radius of wavguide"; Spc(10); txtR2.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter of wavguide"; Spc(8);
txtDw2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(8);
txtDs2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(8); txtFs2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(8); txtF32.Text; Spc(3); "cm"
Printer.EndDoc
End Sub

Private Sub Form_Load()
txtDs.BackColor = vbActiveBorder
txtDs2.BackColor = vbActiveBorder
txtFs.BackColor = vbActiveBorder
txtFs2.BackColor = vbActiveBorder
txtF32.BackColor = vbActiveBorder
txtF3.BackColor = vbActiveBorder
txtR2.BackColor = vbActiveBorder
txtR.BackColor = vbActiveBorder
txtDw2.BackColor = vbActiveBorder
txtDw.BackColor = vbActiveBorder
txtDF2.BackColor = vbActiveBorder
txtDF.BackColor = vbActiveBorder
txtSR2.BackColor = vbActiveBorder
txtSR.BackColor = vbActiveBorder
```

```
txtdBH2.BackColor = vbActiveBorder
txtdBH.BackColor = vbActiveBorder
txtdBE2.BackColor = vbActiveBorder
txtdBE.BackColor = vbActiveBorder
txtHPC2.BackColor = vbActiveBorder
txtHPC.BackColor = vbActiveBorder
txtEPC2.BackColor = vbActiveBorder
txtEPC.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtWL.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder
txtG.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
fraPara.BackColor = vbActiveBorder
fraNewbeam.BackColor = vbActiveBorder
Frame1.BackColor = vbActiveBorder
Frame2.BackColor = vbActiveBorder

fraNewbeam.Visible = False
fraPara.Visible = True
End Sub



Private Sub txtBW_Change()
cmdNext.Enabled = (Len(txtBW.Text) > 0)
cmdNext1.Enabled = (Len(txtBW.Text) > 0)
End Sub

Private Sub txtDia_Change()
dia = Val(txtDia.Text)
If dia > 500 Then
Beep
MsgBox ("Diameter must between 20-500 cm"), (0),
("Invalid data")
txtDia.Text = ""
End If
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub


Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid data")
txtFreq.Text = ""
End If
End Sub


Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0.1 Or NBW > 20 Then
Beep
txtNBW.Text = ""
End If
End Sub

Private Sub txtNFC_Change()
cmdNext2.Enabled = (Len(txtNFC.Text) > 0)
End Sub

Private Sub txtTTG_Change()
TTG = Val(txtTTG.Text)
If TTG < 0 Or TTG > 100 Then
Beep
txtTTG.Text = ""
End If
```

```
End Sub

Private Sub cmdBack_Click()
fraNewbeam.Visible = False
fraPara.Visible = True

End Sub

Private Sub cmdBack2_2_Click()
fraCF2.Visible = False
fraNewbeam.Visible = True
End Sub

Private Sub cmdBack2_Click()
fraCF.Visible = False
fraPara.Visible = True
End Sub


Private Sub cmdBack3_2_Click()
fraCW2.Visible = False
fraCF2.Visible = True
End Sub

Private Sub cmdBack3_Click()
fraCW.Visible = False
fraCF.Visible = True
End Sub

Private Sub cmdBack4_2_Click()
fraCasse2.Visible = False
fraCW2.Visible = True
End Sub

Private Sub cmdBack4_Click()
fraCasse.Visible = False
fraCW.Visible = True
End Sub

Private Sub cmdBack1_Click()
End Sub

Private Sub cmdCancel_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
End Sub


Private Sub cmdClear_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
txtTTG.Locked = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNBW.Locked = False
Label1 = ""
End Sub

Private Sub cmdExit_Click()
frmChoise.Show
frmCas_C.Visible = False

End Sub
```

```
Private Sub cmdExit2_2_Click()
Unload frmCas_C
End Sub

Private Sub cmdExit2_Click()
Unload frmCas_C
End Sub

Private Sub cmdClose1_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
Unload frmCas_C
End Sub

Private Sub cmdNext_Click()
fraPara.Visible = False
fraNewbeam.Visible = True
End Sub

Private Sub cmdNext1_Click()
fraPara.Visible = False
fraCF.Visible = True
Dim TTG, F, WL, Pi, G As Double
Pi = 4 * Atn(1)
TTG = txtTTG.Text
F = txtFreq.Text
WL = txtWL.Text
G = txtG.Text
  If (TTG - G) <= 12 Then
  GF = 12
  Else
  GF = TTG - G
  End If
  txtDF.Text = (WL / Pi) * 10 ^ ((GF + 2.85) / 20)
  txtDF.Text = Format(txtDF.Text, "##0.00")
DF = txtDF.Text
  txtSR.Text = DF ^ 2 / (8 * WL * 0.39)
  txtSR.Text = Format(txtSR.Text, "##0.00")

Dim X, Y As Single
X = (2.088 * WL / (Pi * DF))
txtdBH.Text = 2 * (Atn(X / Sqr(-X * X + 1))) * 180 / Pi
Y = (1.782 * WL / (Pi * DF))
txtdBE.Text = 2 * (Atn(Y / Sqr(-Y * Y + 1))) * 180 / Pi
txtdBH.Text = Format$(txtdBH.Text, "##0.00")
txtdBE.Text = Format$(txtdBE.Text, "##0.00")

SR = txtSR.Text
txtHPC.Text = 0.471 * SR
txtEPC.Text = 0.807 * SR
txtHPC.Text = Format$(txtHPC.Text, "##0.00")
txtEPC.Text = Format$(txtEPC.Text, "##0.00")
End Sub

Private Sub cmdNext2_Click()
Dim TTG, F, WL, Pi, NG As Double
Pi = 4 * Atn(1)
TTG = txtTTG.Text
F = txtFreq.Text
WL = txtWL.Text
NG = txtNG.Text
  If (TTG - NG) <= 12 Then
  GF = 12
  Else
  GF = TTG - NG
```

```
    End If
    txtDF2.Text = (WL / Pi) * 10 ^ ((GF + 2.85) / 20)
    txtDF2.Text = Format(txtDF2.Text, "##0.00")
DF2 = txtDF2.Text
DF2 = Format(DF2, "##0.00")
    txtSR2.Text = DF2 ^ 2 / (8 * WL * 0.39)
    txtSR2.Text = Format(txtSR2.Text, "##0.00")
Dim X, Y As Single
X = (2.088 * WL / (Pi * DF2))
txtdBH2.Text = 2 * (Atn(X / Sqr(-X * X + 1))) * 180 / Pi
Y = (1.782 * WL / (Pi * DF2))
txtdBE2.Text = 2 * (Atn(Y / Sqr(-Y * Y + 1))) * 180 / Pi
txtdBH2.Text = Format(txtdBH2.Text, "##0.00")
txtdBE2.Text = Format(txtdBE2.Text, "##0.00")
SR2 = txtSR2.Text
SR2 = Format(SR2, "##0.00")
txtHPC2.Text = 0.471 * SR2
txtEPC2.Text = 0.807 * SR2
txtHPC2.Text = Format$(txtHPC2.Text, "##0.00")
txtEPC2.Text = Format$(txtEPC2.Text, "##0.00")
fraNewbeam.Visible = False
fraCF2.Visible = True
End Sub

Private Sub cmdNext3_2_Click()
fraCW2.Visible = True
fraCF2.Visible = False
Dim Pi, WL As Double
Pi = 4 * Atn(1)
WL = txtWL.Text
    txtR2.Text = (WL * 1.84) / (2 * Pi)
    txtR2.Text = Format(txtR2.Text, "##0.000")
    txtDw2.Text = (WL * 1.84) / Pi
    txtDw2.Text = Format(txtDw2.Text, "##0.000")
Dw2 = txtDw2.Text
Dw2 = Format(Dw2, "##0.000")
R2 = txtR2.Text
R2 = Format(R2, "##0.00")
End Sub

Private Sub cmdNext3_Click()
fraCW.Visible = True
fraCF.Visible = False
Dim Pi, WL As Double
Pi = 4 * Atn(1)
WL = txtWL.Text
    txtR.Text = (WL * 1.84) / (2 * Pi)
    txtR.Text = Format(txtR.Text, "##0.000")
    txtDw.Text = (WL * 1.84) / Pi
    txtDw.Text = Format(txtDw.Text, "##0.000")
Dw = txtDw.Text
Dw = Format(Dw, "##0.000")
R = txtR.Text
R = Format(R, "##0.00")
    End Sub

Private Sub cmdNext4_2_Click()
fraCasse2.Visible = True
fraCW2.Visible = False
Dim dia, NFD, NFC, ND, WL As Double
dia = txtDia.Text
NFD = txtNFD.Text
NFC = txtNFC.Text
ND = txtND.Text
WL = txtWL.Text

    txtDs2.Text = Sqr(2 * WL * NFC)
    txtDs2.Text = Format(txtDs2.Text, "##0.000")
Ds2 = txtDs2.Text
Ds2 = Format(Ds2, "##0.00")
    txtFs2.Text = Ds2 * NFC / dia
    txtFs2.Text = Format(txtFs2.Text, "##0.000")
Fs2 = Format(txtFs2.Text, "##0.00")
Dim dBH2, Pi As Double
dBH2 = txtdBH2.Text
Pi = 4 * Atn(1)
    txtF32.Text = Ds2 / (2 * Tan(dBH2 * Pi / 360))
    txtF32.Text = Format(txtF32.Text, "##0.000")
F32 = Format(txtF32.Text, "##0.00")
LL = Sqr((SR2 ^ 2) - (DF2 / 2 ^ 2))
LL = Format(LL, "##0.00")
Ls = (Dw2 / DF2) * LL
```

```
End Sub

Private Sub cmdNext4_Click()
fraCasse.Visible = True
fraCW.Visible = False
dia = txtDia.Text
FD = txtFD.Text
FC = txtFC.Text
d = txtD.Text
WL = txtWL.Text

   txtDs.Text = Sqr(2 * WL * FC)
   txtDs.Text = Format$(txtDs.Text, "##0.000")
Ds = Format(txtDs.Text, "##0.00")
   txtFs.Text = Ds * FC / dia
   txtFs.Text = Format$(txtFs.Text, "##0.000")
Fs = Format(txtFs, "##0.00")
Dim dBH, Pi As Double
dBH = txtdBH.Text
Pi = 4 * Atn(1)
   txtF3.Text = Ds / (2 * Tan(dBH * Pi / 360))
   txtF3.Text = Format$(txtF3.Text, "##0.000")
F3 = Format(txtF3.Text, "##0.00")
LL = Sqr((SR ^ 2) - (DF / 2 ^ 2))
LL = Format(LL, "##0.00")
Ls = (Dw / DF) * LL
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4 GHz"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diameter data."), (0), ("Invalid
data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diameter."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0), ("Invalid
data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFD.Text = ""
ElseIf txtTTG.Text = "" Then
MsgBox ("If you don't insert Total Gain, you will get
minimum gain of feedhorn."), (0), ("Invalid data")
txtTTG.Text = "0"
Again:
Msg = "The data error.Try again!"  ' Define message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
cmdClear.TabStop = True
cmdClear.TabIndex = 1
Else
txtFreq.Text = Format(txtFreq.Text, "##0.00")
txtDia.Text = Format(txtDia.Text, "##0.00")
txtFD.Text = Format(txtFD.Text, "##0.00")
Dim F, FD, TTG As Double
F = txtFreq.Text
dia = txtDia.Text
FD = txtFD.Text
TTG = txtTTG.Text
   txtWL.Text = 30 / F
   txtWL.Text = Format(txtWL.Text, "##0.00")
   txtFC.Text = FD * dia
   txtFC.Text = Format(txtFC.Text, "##0.00")
```

```
Dim WL As Double
FC = txtFC.Text
WL = txtWL.Text
   txtD.Text = dia ^ 2 / (16 * FC)
   txtD.Text = Format(txtD.Text, "##0.00")
Dim d, Pi As Double
d = txtD.Text
Pi = 4 * Atn(1)
   txtG.Text = 10 * Log(2 + (d * Pi ^ 2 * dia ^ 2 * d ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
   txtG.Text = Format$(txtG.Text, "##0.00")
   txtBW.Text = 2 * Atn(WL * FC / (d ^ 2 * dia)) * (180 /
Pi)
   txtBW.Text = Format$(txtBW.Text, "##0.00")
txtFreq.Locked = True
txtDia.Locked = True
txtFD.Locked = True
txtTTG.Locked = True
LabelD1 = dia

End If
End Sub

Private Sub cmdAccept_Click()
If txtFreq.Text = "" Or txtDia.Text = "" Or txtFD.Text = ""
Or txtTTG.Text = "" Then
MsgBox ("Please insert all value."), (0), ("Error value")
ElseIf txtFreq.Text = "0" Or txtDia.Text = "0" Or
txtFD.Text = "0" Or txtTTG.Text = "0" Then
MsgBox ("Value can't be zero"), (0), ("Error value")
Else
cmdAccept.Enabled = False
cmdOK.Enabled = True
End If
End Sub


Private Sub cmdOK0_Click()
If txtFreq.Text = "" Then
MsgBox ("You must insert frequency data."), (0), ("Invalid
data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert diameter data."), (0), ("Invalid
data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0), ("Invalid
data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
ElseIf txtTTG.Text = "" Then
MsgBox ("If you don't insert total gain,this program will
use minimum total gain."), (0), ("Total gain")
txtTTG.Text = 0
Else
Dim F, dia, FD As Double
F = txtFreq.Text
dia = txtDia.Text
FD = txtFD.Text
   txtWL.Text = 30 / F
   txtWL.Text = Format$(txtWL.Text, "##0.00")
   txtFC.Text = FD * dia
   txtFC.Text = Format$(txtFC.Text, "##0.00")
Dim FC, WL As Double
FC = txtFC.Text
WL = txtWL.Text
   txtD.Text = dia ^ 2 / (16 * FC)
   txtD.Text = Format$(txtD.Text, "##0.00")
Dim d, Pi As Double
d = txtD.Text
Pi = 4 * Atn(1)
   txtG.Text = 10 * Log(2 + (d * Pi ^ 2 * dia ^ 2 * d ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
   txtG.Text = Format$(txtG.Text, "##0.00")
   txtBW.Text = 2 * Atn(WL * FC / (d ^ 2 * dia)) * (180 /
Pi)
```

```
    txtBW.Text = Format$(txtBW.Text, "##0.00")
cmdNext1.Enabled = True
txtFreq.Locked = True
txtDia.Locked = True
txtTTG.Locked = True
txtFD.Locked = True
End If
End Sub



Private Sub cmdOK2_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth data."), (0),
("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0), ("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define message.
Style = vbOKOnly + vbCritical ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style. Title)

Else
Dim NBW, F, dia, WL, Pi As Double
NBW = txtNBW.Text
NBW = Format(NBW, "##0.00")
F = txtFreq.Text
dia = txtDia.Text
WL = txtWL.Text
Pi = 4 * Atn(1)
   NFD = ((Tan(NBW * Pi / 360)) * dia ^ 2 / (WL * 16 ^
2)) ^ (1 / 3)
NFD = Format(NFD, "##0.00")
   txtNFD.Text = NFD
   NFC = dia * NFD
NFC = Format(NFC, "##0.00")
   txtNFC.Text = NFC
   ND = dia ^ 2 / (16 * NFC)
   txtND.Text = ND
ND = Format(ND, "##0.00")
   NG = 10 * Log(2 + (ND * Pi ^ 2 * dia ^ 2 * ND ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
   txtNG.Text = NG

txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")

txtNBW.Locked = True
LabelD1 = dia

End If

End Sub

Private Sub cmdPrev1_Click()
If dia > 90 Then
frmPrevCC3.Show
Else
frmPrevCC1.Show
End If
End Sub

Private Sub cmdPrev2_Click()
If dia > 90 Then
frmPrevCC4.Show
Else
frmPrevCC2.Show
End If
End Sub

Private Sub Command2_Click()

End Sub

Private Sub cmdPrint1_Click()
```

```
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.Print , ""
Printer.Print , ""
Printer.Print , "Cassegrain dish with circular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.Print , Spc(5); "Give data"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , Spc(5); "Get data"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "focal length"; Spc(7); txtFC.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Depth"; Spc(14); txtD.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Gain"; Spc(15); txtG.Text; Spc(3); "dB"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Beamwidth"; Spc(10); txtBW.Text; Spc(3);
"Degrees"
'-----------------
Printer.NewPage
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular feedhorn"   'Topic'
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter of feedhorn(d)"; Spc(5);
txtDF.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Slant radius(R)"; Spc(13); txtSR.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtdBH.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtdBE.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC.Text; Spc(3); "cm"
Printer.Print ""
Printer.Print , ""
'---------------
Printer.NewPage
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular wavguide"   'Topic'
```

```
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Radius of wavguide"; Spc(10); txtR.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter of wavguide"; Spc(8);
txtDw.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.Print , "Cassegrain dish"   'Topic'
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(5);
txtDs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(5); txtFs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(5); txtF3.Text; Spc(3); "cm"
Printer.EndDoc



End Sub

Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish with circular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New beamwidth"; Spc(6); txtNBW.Text;
Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New depth"; Spc(10); txtND.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New gain"; Spc(11); txtNG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New focal length"; Spc(3); txtNFC.Text;
Spc(3); "cm"
```

```
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New F/D ratio"; Spc(6); txtNFD.Text

'.....

Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular feedhorn"   'Topic'
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print ,
Printer.Print , "Diameter of feedhorn(d)"; Spc(5);
txtDF2.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Slant radius(R)"; Spc(13); txtSR2.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtdBH2.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtdBE2.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC2.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC2.Text; Spc(3); "cm"
Printer.Print , ""
Printer.Print , ""

'.....

Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.Print , "Circular wavguide"   'Topic'
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print ,
Printer.Print , "Radius of wavguide"; Spc(10); txtR2.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter of wavguide"; Spc(8);
txtDw2.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish"   'Topic'
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(8);
txtDs2.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(8); txtFs2.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(8); txtF32.Text; Spc(3); "cm"
Printer.EndDoc
End Sub

Private Sub Form_Load()
txtDs.BackColor = vbActiveBorder
txtDs2.BackColor = vbActiveBorder
txtFs.BackColor = vbActiveBorder
txtFs2.BackColor = vbActiveBorder
txtF32.BackColor = vbActiveBorder
```

```
txtF3.BackColor = vbActiveBorder
txtR2.BackColor = vbActiveBorder
txtR.BackColor = vbActiveBorder
txtDw2.BackColor = vbActiveBorder
txtDw.BackColor = vbActiveBorder
txtDF2.BackColor = vbActiveBorder
txtDF.BackColor = vbActiveBorder
txtSR2.BackColor = vbActiveBorder
txtSR.BackColor = vbActiveBorder
txtdBH2.BackColor = vbActiveBorder
txtdBH.BackColor = vbActiveBorder
txtdBE2.BackColor = vbActiveBorder
txtdBE.BackColor = vbActiveBorder
txtHPC2.BackColor = vbActiveBorder
txtHPC.BackColor = vbActiveBorder
txtEPC2.BackColor = vbActiveBorder
txtEPC.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtWL.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder
txtG.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
fraPara.BackColor = vbActiveBorder
fraNewbeam.BackColor = vbActiveBorder
Frame1.BackColor = vbActiveBorder
Frame2.BackColor = vbActiveBorder

fraNewbeam.Visible = False
fraPara.Visible = True
End Sub


Private Sub txtBW_Change()
cmdNext.Enabled = (Len(txtBW.Text) > 0)
cmdNext1.Enabled = (Len(txtBW.Text) > 0)
End Sub

Private Sub txtDia_Change()
dia = Val(txtDia.Text)
If dia > 500 Then
Beep
MsgBox ("Diameter must between 20-500 cm"), (0),
("Invalid data")
txtDia.Text = ""
End If
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub


Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid data")
txtFreq.Text = ""
End If
End Sub

Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0.1 Or NBW > 20 Then
Beep
txtNBW.Text = ""
End If
End Sub

Private Sub txtNFC_Change()
cmdNext2.Enabled = (Len(txtNFC.Text) > 0)
```

```
End Sub


Private Sub txtTTG_Change()
TTG = Val(txtTTG.Text)
If TTG < 0 Or TTG > 100 Then
Beep
txtTTG.Text = ""
End If
End Sub

Private Sub cmdBack_1_Click()
fraRecfh2.Visible = False
fraNewbeam.Visible = True
End Sub

Private Sub cmdBack_Click()
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdBack1_Click()
fraRecfh.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdBack3_2_Click()
fraRecfh2.Visible = True
fraRecWG2.Visible = False
End Sub

Private Sub cmdBack3_Click()
fraRecfh.Visible = True
fraRecWG.Visible = False
End Sub

Private Sub cmdBack4_2_Click()
fraDesg2.Visible = False
fraRecWG2.Visible = True
End Sub
```

```
Private Sub cmdBack4_Click()
fraDesg_R.Visible = False
fraRecWG.Visible = True
End Sub

Private Sub cmdBye_Click()
Unload frmCasR
End Sub

Private Sub cmdCancel_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
End Sub


Private Sub cmdClear_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
txtTTG.Locked = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNBW.Locked = False
Label1 = ""
End Sub
```

```
Private Sub cmdClose1_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtTTG.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
Unload frmCasR
End Sub


Private Sub cmdExit_Click()
Unload frmCasR
End Sub


Private Sub cmdNext_Click()
fraPara.Visible = False
fraNewbeam.Visible = True
End Sub


Private Sub cmdNext1_Click()
Dim F, G, Pi, TTG As Single
F = txtFreq.Text
G = txtG.Text
TTG = txtTTG.Text
Pi = 4 * Atn(1)
  L = 30 / F        'L=wavelength'
  a = L / 2         'Dimension of waveguide'
  b = a / 2         '        "--"      '
b = Format(b, "##0.000")
If (TTG - G) < 12 Then
Grf = 12
Else: Grf = TTG - G
End If

  Gd = 10 ^ (Grf / 10)  'Gd=gain require,not dB'
  W = 0.489 * L * Sqr(Gd)
  h = 0.332 * L * Sqr(Gd)
  Rh = 0.0746 * L * Gd
  RE = 0.0531 * L * Gd
  Lh = (W - a) / W * Sqr(Rh ^ 2 - (W ^ 2 / 4))
  Le = Lh
  RE = h / (h - b) * Sqr(Lh ^ 2 + ((h - b) ^ 2 / 4))
  Gactual = 10 * Log(4 * Pi * h * W / L ^ 2) / Log(10) -
0.91 - 1.14 - 0.96
  Gdnew = Grf * Grf / (Gactual)   'dB'
  Gdn = 10 ^ (Gdnew / 10)    'not dB'
  txtW.Text = (0.489 * Sqr(Gdn) * L)
  txtH.Text = (0.332 * Sqr(Gdn) * L)
  txtRh.Text = (0.0746 * Gdn * L)
  txtRe.Text = (0.0531 * Gdn * L)
txtW.Text = Format$(txtW.Text, "##0.00")
txtH.Text = Format$(txtH.Text, "##0.00")
txtRh.Text = Format$(txtRh.Text, "##0.000")
txtRe.Text = Format$(txtRe.Text, "##0.000")
WW = txtW.Text
RRh = txtRh.Text
  txtL.Text = Sqr(RRh ^ 2 - (WW / 2) ^ 2) * (WW - a) /
WW
txtL.Text = Format$(txtL.Text, "##0.000")
LL = txtL.Text
Dim X, C As Single
  HH = txtH.Text
  X = 0.6951 * (L / WW)
  C = Atn(X / Sqr(-X * X + 1))
  txtHdB.Text = 2 * C * 180 / Pi
```

```
    txtHdB.Text = Format$(txtHdB.Text, "##0.00")
    Y = 0.4735 * L / h
    CY = Atn(Y / Sqr(-Y * Y + 1))
    txtEdB.Text = 2 * CY * 180 / Pi
    txtEdB.Text = Format$(txtEdB.Text, "##0.00")
  Dim RRe As Single
  RRe = txtRe.Text
    txtHPC.Text = 0.508 * RRh
    txtEPC.Text = 0.494 * RRe
    txtHPC.Text = Format$(txtHPC.Text, "##0.000")
    txtEPC.Text = Format$(txtEPC.Text, "##0.000")
  fraRecfh.Visible = True
  G3 = txtG.Text
  fraRecWG.Visible = False
  End Sub

  Private Sub cmdNext2_Click()
  Dim F, NG, Pi, TTG As Single
  F = txtFreq.Text
  NG = txtNG.Text
  TTG = txtTTG.Text
  Pi = 4 * Atn(1)
    L = 30 / F        'L=wavelength'
    a = L / 2         'Dimension of waveguide'
    b = a / 2 '       "--"
  b = Format(b, "##0.000")
  If (TTG - NG) < 12 Then
  Grf = 12
  Else: Grf = TTG - NG
  End If

    Gd = 10 ^ (Grf / 10)  'Gd=gain require,not dB'
    W = 0.489 * L * Sqr(Gd)
    h = 0.332 * L * Sqr(Gd)
    Rh = 0.0746 * L * Gd
    RE = 0.0531 * L * Gd
    Lh = (W - a) / W * Sqr(Rh ^ 2 - (W ^ 2 / 4))
    Le = Lh
    RE = h / (h - b) * Sqr(Lh ^ 2 + ((h - b) ^ 2 / 4))
    Gactual = 10 * Log(4 * Pi * h * W / L ^ 2) / Log(10) -
0.91 - 1.14 - 0.96
    Gdnew = Grf * Grf / (Gactual)   'dB'
    Gdn = 10 ^ (Gdnew / 10)    'not dB'
    Gdn = Format$(Gdn, "##0.00")
    txtW2.Text = (0.489 * Sqr(Gdn) * L)
    txtH2.Text = (0.332 * Sqr(Gdn) * L)
    txtRh2.Text = (0.0746 * Gdn * L)
    txtRe2.Text = (0.0531 * Gdn * L)
  txtW2.Text = Format$(txtW2.Text, "##0.00")
  txtH2.Text = Format$(txtH2.Text, "##0.00")
  txtRh2.Text = Format$(txtRh2.Text, "##0.000")
  txtRe2.Text = Format$(txtRe2.Text, "##0.000")
  WW = txtW2.Text
  RRh = txtRh2.Text
    txtL2.Text = Sqr(RRh ^ 2 - (WW / 2) ^ 2) * (WW - a) /
  WW
  txtL2.Text = Format$(txtL2.Text, "##0.000")
  LL = txtL2.Text
  Dim X, C As Single
    HH = txtH2.Text
    X = 0.6951 * (L / WW)
    C = Atn(X / Sqr(-X * X + 1))
    txtHdB2.Text = 2 * C * 180 / Pi
    txtHdB2.Text = Format$(txtHdB2.Text, "##0.00")
    Y = 0.4735 * L / h
    CY = Atn(Y / Sqr(-Y * Y + 1))
    txtEdB2.Text = 2 * CY * 180 / Pi
    txtEdB2.Text = Format$(txtEdB2.Text, "##0.00")
  Dim RRe As Single
  RRe = txtRe2.Text
    txtHPC2.Text = 0.508 * RRh
    txtEPC2.Text = 0.494 * RRe
    txtHPC2.Text = Format$(txtHPC2.Text, "##0.000")
    txtEPC2.Text = Format$(txtEPC2.Text, "##0.000")
  fraNewbeam.Visible = False
  fraRecfh2.Visible = True
  G4 = txtNG.Text
  End Sub

  Private Sub cmdNext3_2_Click()
  Dim F As Single
  F = txtFreq.Text
```

```
L = 30 / F
txta2.Text = L / 2
txtb2.Text = L / 4
txta2.Text = Format$(txta2.Text, "##0.00")
txtb2.Text = Format$(txtb2.Text, "##0.00")
fraRecfh2.Visible = False
fraRecWG2.Visible = True
End Sub

Private Sub cmdNext3_Click()
Dim F As Single
F = txtFreq.Text
L = 30 / F
txta.Text = L / 2
txtb.Text = L / 4
txta.Text = Format$(txta.Text, "##0.00")
txtb.Text = Format$(txtb.Text, "##0.00")
fraRecWG.Visible = True
fraRecfh.Visible = False
fraPara.Visible = False
End Sub

Private Sub cmdNext4_2_Click()
Dim F, WL, NFC, dia As Single
F = txtFreq.Text
WL = txtWL.Text
NFC = txtNFC.Text
dia = txtDia.Text
txtDs2.Text = Sqr(2 * WL * NFC)
Ls2 = (b * LL) / WW
  Dim EdB2, Pi As Single
   Ds2 = txtDs2.Text
   Ds2 = Format(Ds2, "##0.00")
   EdB2 = txtEdB2.Text
   Pi = 4 * Atn(1)
   txtFs2.Text = Ds2 * NFC / dia
   txtF32.Text = Ds2 / (2 * Tan(EdB2 * Pi / 360))
txtFs2.Text = Format$(txtFs2.Text, "##0.000")
Fs2 = txtFs2.Text
txtF32.Text = Format$(txtF32.Text, "##0.000")
F32 = txtF32.Text
txtDs2.Text = Format$(txtDs2.Text, "##0.000")
fraDesg2.Visible = True
fraRecWG2.Visible = False
End Sub

Private Sub cmdNext4_Click()
Dim F, WL, FC, dia As Single
F = txtFreq.Text
WL = txtWL.Text
FC = txtFC.Text
dia = txtDia.Text
txtDs.Text = Sqr(2 * WL * FC)
  Dim EdB, Pi As Single
   Ds = txtDs.Text
   Ds = Format(Ds, "##0.00")
   EdB = txtEdB.Text
   Pi = 4 * Atn(1)
   txtFs.Text = Ds * FC / dia
   txtF3.Text = Ds / (2 * Tan(EdB * Pi / 360))
txtFs.Text = Format$(txtFs.Text, "##0.000")
Fs = txtFs.Text
Fs = Format(Fs, "##0.00")
txtF3.Text = Format$(txtF3.Text, "##0.000")
F3 = txtF3.Text
F3 = Format(F3, "##0.00")
Ls = (b * LL) / WW
txtDs.Text = Format$(txtDs.Text, "##0.000")
fraDesg_R.Visible = True
fraRecWG.Visible = False
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 1 Then
```

```
MsgBox ("Frequency must not less than 1 GHz"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diameter data."), (0), ("Invalid
data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diameter."), (0), ("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0), ("Invalid
data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0), ("Invalid data")
txtFD.Text = ""
ElseIf txtTTG.Text = "" Then
MsgBox ("If you don't insert Total Gain, you will get
minimum gain of feedhorn."), (0), ("Invalid data")
txtTTG.Text = "0"
Again:
Msg = "The data error.Try again!"  ' Define message.
Style = vbOKOnly + vbCritical ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
cmdClear.TabStop = True
cmdClear.TabIndex = 1
Else
F = txtFreq.Text
dia = txtDia.Text
FD = txtFD.Text
TTG = txtTTG.Text
   txtWL.Text = 30 / F
   txtWL.Text = Format$(txtWL.Text, "##0.00")
   txtFC.Text = FD * dia
   txtFC.Text = Format$(txtFC.Text, "##0.00")
Dim WL As Double
FC = txtFC.Text
WL = txtWL.Text
   txtD.Text = dia ^ 2 / (16 * FC)
   txtD.Text = Format$(txtD.Text, "##0.00")
Dim Pi As Double
d = txtD.Text
Pi = 4 * Atn(1)
   txtG.Text = 10 * Log(2 + (d * Pi ^ 2 * dia ^ 2 * d ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
   txtG.Text = Format$(txtG.Text, "##0.00")
   txtBW.Text = 2 * Atn(WL * FC / (d ^ 2 * dia)) * (180 /
Pi)
   txtBW.Text = Format$(txtBW.Text, "##0.00")
txtFreq.Locked = True
txtDia.Locked = True
txtFD.Locked = True
txtTTG.Locked = True
LabelD1 = dia

End If
End Sub


Private Sub cmdOK2_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth data."), (0),
("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0), ("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"  ' Define message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
Else
Dim NBW, F, dia, WL, Pi As Double
NBW = txtNBW.Text
F = txtFreq.Text
```

```
dia = txtDia.Text
WL = txtWL.Text
Pi = 4 * Atn(1)
  NFD = ((Tan(NBW * Pi / 360)) * dia ^ 2 / (WL * 16 ^
2)) ^ (1 / 3)
  txtNFD.Text = NFD
  NFD = Format(NFD, "##0.00")
  NFC = dia * NFD
  NFC = Format(NFC, "##0.00")
  txtNFC.Text = NFC
  ND = dia ^ 2 / (16 * NFC)
  ND = Format(ND, "##0.00")
  txtND.Text = ND
  NG = 10 * Log(2 * (ND * Pi ^ 2 * dia ^ 2 * ND ^ 2) /
(dia * 9 * WL ^ 2)) / Log(10)
  txtNG.Text = NG
txtNFC.Text = Format(txtNFC.Text, "##0.00")
txtNFD.Text = Format(txtNFD.Text, "##0.00")
txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")

txtNBW.Locked = True
LabelD1 = dia

End If
End Sub



Private Sub Command2_Click()

End Sub

Private Sub Frame2_DragDrop(Source As Control, X As
Single, Y As Single)

End Sub

Private Sub Text9_Change()

End Sub
```

```
Private Sub Text7_Change()

End Sub

Private Sub Text4_Change()

End Sub

Private Sub cmdPreview_Click()
fraPrev1.Visible = True
FH = FC - (F3 + Fs)
lblD.Left = 11
lblD.Top = 50
Picture1.Line (15, 98)-(15, 2), QBColor(8)
Picture1.Line (2, 50)-(98, 50), QBColor(8)
Dim Y, P, M As Single
For Y = (-dia / 2) To (dia / 2) Step 0.1
X = Y ^ 2 / (4 * FC)
 Picture1.PSet ((X + 15), (Y + 50))
Next Y
'.....
M = (dia / 2) ^ 2 / (4 * FC)
X1 = M + 8 + 15
Y1 = (dia / 2) + 50
Y2 = (-dia / 2) + 50
'dimension of diameter
Picture1.Line (12, Y1)-(12, Y2), QBColor(4)
Picture1.Line (10, Y1)-(14, Y1), QBColor(4)
Picture1.Line (10, Y2)-(14, Y2), QBColor(4)
'dimension of focal length
lblFC.Left = (FC / 2) + 15
lblFC.Top = 45
Picture1.Line (15, 45)-((FC + 15), 45), QBColor(4)
Picture1.Line ((FC + 15), 47)-((FC + 15), 43), QBColor(4)
'scale
VX1 = 14.5
VX2 = 15
Dim VY1 As Integer
For VY1 = 2 To 98 Step 1
VY2 = VY1
```

```
Picture1.Line (VX1, VY1)-(VX2, VY2)
Next VY1
For VYY = 0 To 98 Step 5
Picture1.Line (14, VYY)-(VX2, VYY), QBColor(9)
Next VYY
Dim HX1 As Integer
   For HX1 = 3 To 98 Step 1
   HX2 = HX1
   HY1 = 50
   HY2 = 49.5
   Picture1.Line (HX1, HY1)-(HX2, HY2)
Next HX1

For HXX = 0 To 98 Step 5
Picture1.Line (HXX, HY1)-(HXX, 49), QBColor(9)
Next HXX
'"rectangular feedhorn plot
Picture1.Line (15, (50 + b / 2))-((15 + FH - LL + Ls), (50 +
b / 2)), QBColor(12)
Picture1.Line (15, (50 - b / 2))-((15 + FH - LL + Ls), (50 -
b / 2)), QBColor(12)
Picture1.Line ((15 + FH - LL + Ls), (50 + b / 2))-((15 +
FH), (50 + WW / 2)), QBColor(12)
Picture1.Line ((15 + FH - LL + Ls), (50 - b / 2))-((15 +
FH), (50 - WW / 2)), QBColor(12)
Picture1.Line ((15 + FH), (50 + WW / 2))-((15 + FH), (50
- WW / 2)), QBColor(12)
'Plot hyperbola
FF = (F3 + Fs) / 2
AA = FF - Fs
For Ys = -Ds / 2 To Ds / 2 Step 0.001
Xs = AA * Sqr((Ys ^ 2 / (FF ^ 2 - AA ^ 2)) + 1)
Picture1.PSet ((Xs + 15 + Fs), (Ys + 50))
Next Ys
Text1.Text = Fs

End Sub

Private Sub cmdPrev2_Click()
Printer.NewPage
Printer.FontSize = 16
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Rectangular feedhorn"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Aperture dimension"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "W"; Spc(10); txtW.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Re"; Spc(9); txtRe.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "L"; Spc(10); txtL.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H"; Spc(10); txtH.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Rh"; Spc(9); txtRh.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtHdB.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtEdB.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC.Text; Spc(3); "cm"
Printer.Print ""
```

```
Printer.Print , ""
'----------
Printer.NewPage
Printer.FontSize = 16
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Rectangular wavguide"   'Topic'
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Waveguide dimension"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "a"; Spc(10); txta.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "b"; Spc(10); txtb.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish"   'Topic'
Printer.FontBold = flase
Printer.FontUnderline = False
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(5);
txtDs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(5); txtFs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(5); txtF3.Text; Spc(3); "cm"
Printer.EndDoc


End Sub

Private Sub cmdPrint1_Click()
Printer.Orientation = 1
Printer.FontSize = 16
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish with rectangular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
```

```
Printer.Print , "focal length"; Spc(7); txtFC.Text; Spc(3);
"cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Depth"; Spc(14); txtD.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Gain"; Spc(15); txtG.Text; Spc(3); "dB"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Beamwidth"; Spc(10); txtBW.Text; Spc(3);
"Degrees"
'---------------
Printer.NewPage
Printer.FontSize = 16
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Rectangular feedhorn"   'Topic'
Printer.Print , ""   'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Aperture dimension"  'Topic'
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "W"; Spc(10); txtW.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Re"; Spc(9); txtRe.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "L"; Spc(10); txtL.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "H"; Spc(10); txtH.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Rh"; Spc(9); txtRh.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtHdB.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtEdB.Text; Spc(3); "degrees"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC.Text; Spc(3); "cm"
Printer.Print ""
Printer.Print , ""
'---------------
Printer.NewPage
Printer.FontSize = 16
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Rectangular wavguide"   'Topic'
Printer.Print , ""   'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Waveguide dimension"
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "a"; Spc(10); txta.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "b"; Spc(10); txtb.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish"   'Topic'
Printer.FontBold = flase
Printer.FontUnderline = False
```

```
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(5);
txtDs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(5); txtFs.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(5); txtF3.Text; Spc(3); "cm"
Printer.EndDoc

End Sub

Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish with rectangular feedhorn"
Printer.Print , ""
Printer.Print , "Parabolic main reflector"
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10); txtFreq.Text; Spc(3);
"GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(11); txtDia.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Total gain"; Spc(9); txtTTG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New beamwidth"; Spc(6); txtNBW.Text;
Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Wave length"; Spc(8); txtWL.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New depth"; Spc(10); txtND.Text; Spc(3);
"cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New gain"; Spc(11); txtNG.Text; Spc(3);
"dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New focal length"; Spc(3); txtNFC.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New F/D ratio"; Spc(6); txtNFD.Text
'-------------
Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontSize = 16
Printer.Print , "Rectangular feedhorn"    'Topic'
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
```

```
Printer.Print , Spc(5); "Aperture dimension"  'Topic'
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "W"; Spc(10); txtW2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Re"; Spc(9); txtRe2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "L"; Spc(10); txtL2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H"; Spc(10); txtH2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Rh"; Spc(9); txtRh2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane"; Spc(3);
txtHdB2.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane"; Spc(3);
txtEdB2.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC2.Text; Spc(3); "cm"
Printer.Print ""
Printer.Print , ""
'--------------
Printer.NewPage
Printer.FontBold = True
Printer.FontUnderline = True
Printer.FontSize = 16
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontSize = 16
Printer.Print , "Rectangular wavguide"    'Topic'
Printer.Print , ""    'enter line'
Printer.FontUnderline = False
Printer.Print , Spc(5); "Waveguide dimension"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "a"; Spc(10); txta2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "b"; Spc(10); txtb2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Cassegrain dish"    'Topic'
Printer.FontBold = False
Printer.FontUnderline = False
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Diameter of subreflector"; Spc(5);
txtDs2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length of subreflector";
Spc(5); txtFs2.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print , Spc(3); "Focal length from feedhorn to
subreflector"; Spc(5); txtF32.Text; Spc(3); "cm"
Printer.EndDoc


End Sub


Private Sub Next2_Click()

End Sub

Private Sub Command1_Click()
fraPrev1.Visible = False
```

```
End Sub

Private Sub Form_Load()
txtDs.BackColor = vbActiveBorder
txtDs2.BackColor = vbActiveBorder
txtFs.BackColor = vbActiveBorder
txtFs2.BackColor = vbActiveBorder
txtF32.BackColor = vbActiveBorder
txtF3.BackColor = vbActiveBorder
txta.BackColor = vbActiveBorder
txta2.BackColor = vbActiveBorder
txtb.BackColor = vbActiveBorder
txtb2.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtWL.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder
txtG.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
fraPara.BackColor = vbActiveBorder
fraNewbeam.BackColor = vbActiveBorder
fraRecfh2.BackColor = vbActiveBorder
fraRecfh.BackColor = vbActiveBorder
txtW2.BackColor = vbActiveBorder
txtW.BackColor = vbActiveBorder
txtRe.BackColor = vbActiveBorder
txtRe2.BackColor = vbActiveBorder
txtL2.BackColor = vbActiveBorder
txtL.BackColor = vbActiveBorder
txtH.BackColor = vbActiveBorder
txtH2.BackColor = vbActiveBorder
txtRh.BackColor = vbActiveBorder
txtRh2.BackColor = vbActiveBorder
txtEdB2.BackColor = vbActiveBorder
txtEdB.BackColor = vbActiveBorder
txtHdB.BackColor = vbActiveBorder
txtHdB2.BackColor = vbActiveBorder
txtHPC.BackColor = vbActiveBorder
txtHPC2.BackColor = vbActiveBorder
txtEPC2.BackColor = vbActiveBorder
txtEPC.BackColor = vbActiveBorder
End Sub

Private Sub Print_Click()
If dia > 100 And dia <= 400 Then
frmPrec1.Show
ElseIf dia > 400 And dia <= 750 Then
frmPrec2.Show
ElseIf dia > 750 And diap <= 1600 Then
frmPrec3.Show
Else
frmPrevC1.Show
End If
End Sub

Private Sub Print2_Click()
If dia > 100 And dia <= 400 Then
frmPrecn1.Show
ElseIf dia > 400 And dia <= 750 Then
frmPrecn2.Show
ElseIf dia > 750 And dia <= 1600 Then
frmPrecn3.Show
Else
frmPrevC2.Show
End If
End Sub

Private Sub Text1_Change()
End Sub

Private Sub txtBW_Change()
cmdNext.Enabled = (Len(txtBW.Text) > 0)
cmdNext1.Enabled = (Len(txtBW.Text) > 0)

End Sub


Private Sub txtDia_Change()
dia = Val(txtDia.Text)
```

```
If dia > 1600 Then
Beep
MsgBox ("Diameter must between 20-1600 cm"), (0),
("Invalid data")
txtDia.Text = ""
End If
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid data")
txtFreq.Text = ""
End If

End Sub


Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0.1 Or NBW > 20 Then
Beep
txtNBW.Text = ""
End If
End Sub

Private Sub txtNFC_Change()
cmdNext2.Enabled = (Len(txtNFC.Text) > 0)
End Sub


Private Sub txtTTG_Change()
TTG = Val(txtTTG.Text)
If TTG < 0 Or TTG > 100 Then
Beep
txtTTG.Text = ""
End If
End Sub
```

```
rivate Sub cmdExit_Click()
Unload frmCF
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtGF.Text = "" Then
MsgBox ("You must insert Gain of feed horn
data."), (0), ("Invalid data")
ElseIf txtGF.Text = "0" Then
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtGF.Text = ""
ElseIf txtGF.Text < 12 Then
MsgBox ("The gain of feed horn must not
lessthan 12 dB."), (0), ("Invalid data")
txtGF.Text = ""
txtD.Text = ""
txtSR.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
Else
Dim GF A Single
GF = txtGF.Text
Dim L, F As Single
Pi = (4 * Atn(1))
F = txtFreq.Text
L = 30 / F        'L=wavelength'
txtD.Text = (L / Pi) * 10 ^ ((GF + 2.85) / 20)
txtD.Text = Format(txtD.Text, "##0.00")
dc = txtD.Text
dc = Format(txtD.Text, "##0.00")
txtSR.Text = dc ^ 2 / (8 * L * 0.39)
txtSR.Text = Format(txtSR.Text, "##0.00")
SR = txtSR.Text
SR = Format(SR, "##0.00")
R = (L * 1.84) / (2 * Pi) 'Radius of cirular
waveguide
R = Format(R, "##0.00")
Dw = 2 * R
Dw = Format(Dw, "##0.00")
SRW = SR * R / dc
LLc = Sqr((SR ^ 2) - (dc / 2 ^ 2))
LLc = Format(LLc, "##0.00")
txtHPC.Text = 0.471 * SR
txtEPC.Text = 0.807 * SR
txtHPC.Text = Format(txtHPC.Text, "##0.00")
txtEPC.Text = Format(txtEPC.Text, "##0.00")
Dim X, Y As Single
X = (2.088 * L / (Pi * dc))
txtdBH.Text = 2 * (Atn(X / Sqr(-X * X + 1))) *
180 / Pi
Y = (1.782 * L / (Pi * dc))
txtdBE.Text = 2 * (Atn(Y / Sqr(-Y * Y + 1))) *
180 / Pi
txtdBH.Text = Format(txtdBH.Text, "##0.00")
txtdBE.Text = Format(txtdBE.Text, "##0.00")
End If

End Sub

Private Sub cmdPrev_Click()
If (dc + LLc + 3) > 10 Then
frmprevCF2.Show
Else
frmPrevCF1.Show
End If
End Sub

Private Sub cmdPrint_Click()
Printer.CurrentX = 0
```

```
Printer.CurrentY = 0
Printer.Orientation = 1

Printer.Print , ""
Printer.Print , ""
Printer.FontSize = 16
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Circular feedhorn"
Printer.Print , ""
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.FontBold = False
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Gain of feedhorn"; Spc(4);
txtGF.Text; Spc(3); "dB"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.FontBold = False
Printer.Print , "Diameter of feedhorn(d)";
Spc(5); txtD.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Slant radius(R)"; Spc(13);
txtSR.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in H-plane";
Spc(3); txtdBH.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "3-dB beamwidth in E-plane";
Spc(3); txtdBE.Text; Spc(3); "degrees"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "H-plane phase center"; Spc(8);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "E-plane phase center"; Spc(8);
txtEPC.Text; Spc(3); "cm"
Printer.EndDoc
'.....
End Sub

Private Sub Command1_Click()

txtGF.Text = ""
txtFreq.Text = ""
txtD.Text = ""
txtSR.Text = ""
txtdBH.Text = ""
txtdBE.Text = ""
txtHPC.Text = ""
txtEPC.Text = ""
End Sub

Private Sub mnuExit_Click()
frmChoise.Show
frmCF.Visible = False

End Sub

Private Sub mnuMain_Click()
frmChoise.Show
frmCF.Visible = False
End Sub

Private Sub Form_Load()
txtD.BackColor = vbActiveBorder
txtSR.BackColor = vbActiveBorder
txtdBH.BackColor = vbActiveBorder
txtdBE.BackColor = vbActiveBorder
txtHPC.BackColor = vbActiveBorder
txtEPC.BackColor = vbActiveBorder
```

```
End Sub

Private Sub txtD_Change()
cmdPrint.Enabled = Len(txtD.Text) > 0
cmdPrev.Enabled = Len(txtD.Text) > 0
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency "), (0), ("Invalid
data")
txtFreq.Text = ""
End If
End Sub

Private Sub txtGF_Change()
GF = Val(txtGF.Text)
If GF > 30 Then
Beep
txtGF.Text = ""
End If
End Sub
Private Sub cmdCancel_Click()
txtFreq.Text = ""
txtR.Text = ""
txtDw.Text = ""
End Sub

Private Sub cmdExit_Click()
Unload frmCirF
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
Else
Dim F, L, Pi As Single
Pi = 4 * Atn(1)
F = txtFreq.Text
L = 30 / F
txtR.Text = (L * 1.84) / (2 * Pi)
R = txtR.Text
txtR.Text = Format(txtR.Text, "##0.000")
txtDw.Text = 2 * R
txtDw.Text = Format$(txtDw.Text, "##0.000")
End If
End Sub

Private Sub cmdPrint_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1

Printer.FontSize = 16
Printer.Print ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Circular waveguide"
Printer.Print ""
Printer.FontUnderline = False
Printer.Print , "Give data"
Printer.Print ""
```

```
Printer.FontBold = False
Printer.Print , , "Frequency of waveguide";
Spc(3); txtFreq.Text; Spc(3); "GHz"
Printer.FontBold = True
Printer.Print ""
Printer.Print ""
Printer.Print , "Get data"
Printer.Print ""
Printer.FontBold = False
Printer.Print , , "Radius of waveguide"; Spc(6);
txtR.Text; Spc(3); "cm"
Printer.Print ""
Printer.Print , , "Diameter of waveguide";
Spc(4); txtDw.Text; Spc(3); "cm"
Printer.Print EndDoc
End Sub

Private Sub Date_Change()

End Sub

Private Sub Data_Change()
Data = Format(Data.Text, "Long Date")
End Sub

Private Sub Form_Load()
txtR.BackColor = vbActiveBorder
txtDw.BackColor = vbActiveBorder

End Sub

Private Sub fraCW_DragDrop(Source As
Control, X As Single, Y As Single)


' Returns current system time in the system-
defined long time format.


End Sub

Private Sub Time_Change()
Time = Format(Time.Text, "Long Time")
' Returns current system date in the system-
defined long date format.


End Sub


Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid
data")
txtFreq.Text = ""
End If
End Sub


Private Sub txtR_Change()
cmdPrint.Enabled = Len(txtR) > 0
End Sub

Private Sub cmdPrint1_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Offset-fed reflector"
Printer.Print ""
Printer.FontUnderline = False
Printer.Print , "Give data"
Printer.Print ""
Printer.FontBold = False
Printer.Print , , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print , , "Diameter"; Spc(11);
txtDia.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
```

```
Printer.Print , , "F/D ratio"; Spc(10); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print , , "Angle Wo"; Spc(11);
txtWo.Text; Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , "Total gain"; Spc(9); txtTTG.Text;
Spc(3); "dB"
Printer.Print ""
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , "Get data"
Printer.Print , ""    'enter line'
Printer.FontBold = False


End Sub

Private Sub cmdClose_Click()
Unload frmOff1
End Sub



Private Sub cmdNext_Click()
Pi = 4 * Atn(1)
ddp = (4 * Sin(Woo)) * FCo / (Cos(Woo) + 1)
Dt = ddp / 2 + diao / 2
depth = (Dt) ^ 2 / (4 * FCo) 'Depth
Dtl = ddp / 2 - diao / 2
depthl = (Dtl) ^ 2 / (4 * FCo) 'depth low
If (diao) > 90 And (diao) <= 400 Then
frmPreo1.Show
ElseIf diao > 400 And diao <= 600 Then
frmPreo2.Show
Else
frmPrevO3.Show
End If
End Sub



Private Sub cmdPrev2_Click()
Pi = 4 * Atn(1)
ddp2 = (4 * Sin(Woo)) * NFCo / (Cos(Woo) +
1)
Dt2 = ddp2 / 2 + diao / 2
depth2 = (Dt2) ^ 2 / (4 * NFCo) 'Depth
Dtl2 = ddp2 / 2 - diao / 2
depthl2 = (Dtl2) ^ 2 / (4 * NFCo) 'depth low
frmPrevO2.Show
End Sub


Private Sub cmdPrint_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Offset-fed reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print , , "Angle Wo"; Spc(11);
txtWo.Text; Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , , "Total gain"; Spc(11);
txtTTG.Text; Spc(3); "dB"
Printer.Print , ""
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.FontBold = False
Printer.Print , ""
Printer.Print , , "Focal length"; Spc(8);
txtFC.Text; Spc(3); "cm"
```

```
Printer.Print , ""
Printer.Print , , "Depth"; Spc(15); txtDepth.Text;
Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "Gain of dish"; Spc(8);
txtGain.Text; Spc(3); "dB"
Printer.Print , ""
Printer.Print , , "Angle We"; Spc(11);
txtWe.Text; Spc(3); "Degrees"
Printer.Print , ""
Printer.Print , , "Beamwidth"; Spc(9);
txtBW.Text; Spc(3); "Degrees"
Printer.Print , ""
Printer.Print , , "Squint angle"; Spc(8);
txtWs.Text; Spc(3); "Degrees"
Printer.EndDoc
End Sub


Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Offset fed reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print , , "Angle Wo"; Spc(11);
txtWo.Text; Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , , "Total gain"; Spc(11);
txtTTG.Text; Spc(3); "dB"
Printer.Print , ""
Printer.Print , , "New beamwidth"; Spc(6);
txtNBW.Text; Spc(3); "dB"
Printer.Print , ""
Printer.FontBold = True
Printer.Print , Spc(5); "Get new data"
Printer.FontBold = False
Printer.Print , ""
Printer.Print , , "New Focal length"; Spc(7);
txtNFC.Text; Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "New Depth"; Spc(14);
txtND.Text; Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "New Gain of dish"; Spc(7);
txtNG.Text; Spc(3); "dB"
Printer.Print , ""
Printer.Print , , "New Angle We"; Spc(10);
txtNWe.Text; Spc(3); "Degrees"
Printer.Print , ""
Printer.Print , , "New F/D ratio"; Spc(11);
txtNFD.Text
Printer.Print , ""
Printer.Print , , "New Squint angle"; Spc(7);
txtNWs.Text; Spc(3); "Degrees"
Printer.EndDoc
End Sub

Private Sub Command1_Click()

Unload frmOff1
End Sub


Private Sub cmdBack_Click()
fraNewbeam.Visible = False
```

```
fraG1.Visible = True
End Sub


Private Sub cmdCancel_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtWo.Text = ""
txtFD.Text = ""
txtFC.Text = ""
txtDepth.Text = ""
txtWo.Text = ""
txtGain.Text = ""
txtWs.Text = ""
txtBW.Text = ""
cmdNext.Enabled = False
txtNBW.Text = ""
txtND.Text = ""
txtNWs.Text = ""
txtNWe.Text = ""
txtNFD.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
End Sub


Private Sub cmdCancel1_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNWs.Text = ""
txtNWe.Text = ""
txtNBW.Locked = False
End Sub

Private Sub cmdChange_Click()
fraNewbeam.Visible = True
fraG1.Visible = False
End Sub

Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diameter data."), (0),
("Invalid data")
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zer0."), (0),
("Invalid data")
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diameter."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0),
("Invalid data")
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
ElseIf txtWo.Text = "" Then
MsgBox ("You must insert Wo data."), (0),
("Invalid data")
ElseIf txtWo.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
Again:
Msg = "The data error.Try again!"  ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
cmdNext.Enabled = True
Fo = txtFreq.Text
```

```
diao = txtDia.Text
FDo = txtFD.Text
Pi = 4 * Atn(1)
L = 30 / Fo
WWoo = txtWo.Text
Woo = Pi * WWoo / 180
'--------------------
txtFC.Text = diao * FDo
txtFC.Text = Format(txtFC.Text, "##0.00")
FCo = txtFC.Text
txtDepth = diao ^ 2 / (16 * diao * FDo)
txtDepth.Text = Format(txtDepth.Text,
"##0.00")
doo = txtDepth.Text
txtGain.Text = 10 * Log(2 + ((doo / diao) * (Pi *
doo * diao) ^ 2 / (3 * L) ^ 2)) / Log(10)
txtGain.Text = Format(txtGain.Text, "##0.00")
P1 = -Cos(Woo) / Sqr((FDo * 4) ^ 2 + 1)
P2 = (Atn(-P1 / Sqr(-P1 * P1 + 1)) + 2 * Atn(1))
* 180 / Pi
txtWe.Text = P2 + (Atn(-4 * FDo) * 180 / Pi)
txtWe.Text = Format(txtWe.Text, "##0.00")
WWeo = txtWe.Text
Weo = WWeo * Pi / 180
Weo = Format(Weo, "##0.00")
txtBW.Text = (2 * Atn(L * FCo / (doo ^ 2 *
diao))) * 180 / Pi
txtBW.Text = Format(txtBW.Text, "##0.00")
Dim K As Double
K = L * Sin(Woo) / (4 * Pi * FCo)
Arcsin = Atn(K / Sqr(-K * K + 1))
txtWs.Text = Arcsin * 180 / Pi
txtWs.Text = Format(txtWs.Text, "##0.00")
End If
End Sub


Private Sub cmdOK1_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth
data."), (0), ("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0),
("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Dim NBWo, Fo, diao, WLo, Pi As Double
NBWo = txtNBW.Text
Fo = txtFreq.Text
diao = txtDia.Text
WLo = L
Pi = 4 * Atn(1)
  NFDo = ((Tan(NBWo * Pi / 360)) * diao ^ 2 /
(WLo * 16 ^ 2)) ^ (1 / 3)
  txtNFD.Text = NFDo
  NFDo = Format(NFDo, "##0.00")
  NFCo = diao * NFDo
  txtNFC.Text = NFCo
  NFCo = Format(NFCo, "##0.00")
  NDo = diao ^ 2 / (16 * NFCo)
  txtND.Text = NDo
  NDo = Format(NDo, "##0.00")
  NGo = 10 * Log(2 + (NDo * Pi ^ 2 * diao ^ 2
* NDo ^ 2) / (diao * 9 * WLo ^ 2)) / Log(10)
  txtNG.Text = NGo
txtNFC.Text = Format(txtNFC.Text, "##0.00")
txtNFD.Text = Format(txtNFD.Text, "##0.00")
txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")
NP1 = -Cos(Woo) / Sqr((NFDo * 4) ^ 2 + 1)
NP2 = (Atn(-NP1 / Sqr(-NP1 * NP1 + 1)) + 2 *
Atn(1)) * 180 / Pi
txtNWe.Text = NP2 + (Atn(-4 * NFDo) * 180 /
Pi)
```

```
txtNWe.Text = Format(txtNWe.Text, "##0.00")
NWe = txtNWe.Text
NWe = Format(NWe, "##0.00")
Dim NK As Double
NK = L * Sin(Woo) / (4 * Pi * NFCo)
NArcsin = Atn(NK / Sqr(-NK * NK + 1))
txtNWs.Text = NArcsin * 180 / Pi
txtNWs.Text = Format(txtNWs.Text, "##0.00")
txtNBW.Locked = True
End If
End Sub



Private Sub Text5_Change()


End Sub



Private Sub Form_Load()
Frame1.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNWs.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
OLE2.BackColor = vbActiveBorder
txtNWe.BackColor = vbActiveBorder
txtWe.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtGain.BackColor = vbActiveBorder
txtDepth.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
txtWs.BackColor = vbActiveBorder
frmOff1.BackColor = vbActiveBorder
OLE1.BackColor = vbActiveBorder
OLE3.BackColor = vbActiveBorder
OLE4.BackColor = vbActiveBorder
Label1.BackColor = vbActiveBorder
Label2.BackColor = vbActiveBorder
Label3.BackColor = vbActiveBorder
Label4.BackColor = vbActiveBorder
Label5.BackColor = vbActiveBorder
Label6.BackColor = vbActiveBorder
Label7.BackColor = vbActiveBorder

End Sub

Private Sub txtDia_Change()
dia = Val(txtDia.Text)
If dia > 600 Then
Beep
MsgBox ("Diameter must between 20-600 cm"),
(0), ("Invalid data")
txtDia.Text = ""
End If
End Sub

Private Sub txtFC_Change()
cmdPrint.Enabled = Len(txtFC.Text) > 0
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub


Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq < 1 Or freq > 300 Then
Beep
txtFreq.Text = ""
End If
End Sub

Private Sub txtGain_Change()
cmdChange.Enabled = Len(txtGain) > 0
cmdPrint.Enabled = Len(txtGain) > 0
End Sub

Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0 Or NBW > 30 Then
```

```
Beep
txtNBW.Text = ""
End If
End Sub



Private Sub txtNFC_Change()
cmdPrint2.Enabled = Len(txtNFC.Text) > 0
End Sub

Private Sub txtTTG_Change()
TTG = Val(txtTTG.Text)
If TTG < 0 Or TTG > 500 Then
Beep
txtTTG.Text = ""
End If
End Sub

Private Sub txtWo_Change()
Lo = Val(txtWo.Text)
If Lo > 90 Then
Beep
MsgBox ("The angle must less than 90
degree."), (0), ("invalid data")
txtWo.Text = ""
End If
End Sub

Private Sub List1_Click()

End Sub



Private Sub cmdCancel_Click()
frmOpen.Visible = False
End Sub



Private Sub cmdOK_Click()
If Combo1.Text = "Parabolic reflector" Then
frmPara.Show
frmOpen.Visible = False
ElseIf Combo1.Text = "Offset-fed" Then
frmOff1.Show
frmOpen.Visible = False
ElseIf Combo1.Text = "Rectangular waveguide"
Then
frmRecwg.Show
frmOpen.Visible = False
ElseIf Combo1.Text = "Circular waveguide"
Then
frmCirF.Show
frmOpen.Visible = False
ElseIf Combo1.Text = "Circular feedhorn" Then
frmCF.Show
frmOpen.Visible = False
ElseIf Combo1.Text = "Rectangular feedhorn"
Then
frmRecfh.Show
frmOpen.Visible = False
'-----
ElseIf Combo1.Text = "Cassegrain" Then
frmChoice.Show
frmOpen.Visible = False
End If
End Sub

Private Sub cmdClose1_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
```

```
End Sub


Private Sub cmdPrev1_Click()
If diap > 100 And diap <= 500 Then
frmPre1.Show
ElseIf diap > 500 And diap <= 1000 Then
frmPre2.Show
ElseIf diap > 1000 And diap <= 1600 Then
frmPre3.Show
Else
frmPrev1.Show
End If
End Sub


Private Sub cmdBack_Click()
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdCancel_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
End Sub

Private Sub cmdClear_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNBW.Locked = False
LabelD1 = ""
End Sub


Private Sub cmdExit_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNBW.Locked = False
End Sub


Private Sub cmdNext_Click()
fraPara.Visible = False
fraNewbeam.Visible = True
End Sub


Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
```

```
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diarneter data."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diarneter."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0),
("Invalid data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
txtFD.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Fp = txtFreq.Text
Fp = Format(Fp, "##0.00")
diap = txtDia.Text
diap = Format(diap, "##0.00")
FDp = txtFD.Text
FDp = Format(FDp, "##0.00")
   txtWL.Text = 30 / Fp
   txtWL.Text = Format$(txtWL.Text,
"##0.00")
   txtFC.Text = FDp * diap
   txtFC.Text = Format$(txtFC.Text, "##0.00")
Dim WLp As Double
FCp = txtFC.Text
FCp = Format(FCp, "##0.00")
WLp = txtWL.Text
   txtD.Text = diap ^ 2 / (16 * FCp)
   txtD.Text = Format$(txtD.Text, "##0.00")
Dim Pi As Double
dp = txtD.Text
Pi = 4 * Atn(1)
   txtG.Text = 10 * Log(2 + (dp * Pi ^ 2 * diap ^
2 * dp ^ 2) / (diap * 9 * WLp ^ 2)) / Log(10)
   txtG.Text = Format$(txtG.Text, "##0.00")
   txtBW.Text = 2 * Atn(WLp * FCp / (dp ^ 2 *
diap)) * (180 / Pi)
   txtBW.Text = Format$(txtBW.Text,
"##0.00")
txtFreq.Locked = True
txtDia.Locked = True
txtFD.Locked = True
LabelD1 = dia

End If
End Sub



Private Sub cmdOK2_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth
data."), (0), ("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0),
("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
```

```
Else
Dim NBWp, Fp, diap, WLp, Pi As Double
NBWp = txtNBW.Text
Fp = txtFreq.Text
diap = txtDia.Text
WLp = txtWL.Text
Pi = 4 * Atn(1)
   NFDp = ((Tan(NBWp * Pi / 360)) * diap ^ 2 /
(WLp * 16 ^ 2)) ^ (1 / 3)
   txtNFD.Text = NFDp
   NFCp = diap * NFDp
   txtNFC.Text = NFCp
   NDp = diap ^ 2 / (16 * NFCp)
   txtND.Text = NDp
   NGp = 10 * Log(2 + (NDp * Pi ^ 2 * diap ^ 2
* NDp ^ 2) / (diap *^9 * WLp ^ 2)) / Log(10)
   txtNG.Text = NGp
txtNFC.Text = Format(txtNFC.Text, "##0.00")
txtNFD.Text = Format(txtNFD.Text, "##0.00")
txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")
NFDp = txtNFD.Text
NFDp = Format(NFDp, "##0.00")
NDp = txtND.Text
NDp = Format(NDp, "##0.00")
NFCp = txtNFC.Text
NFCp = Format(NFCp, "##0.00")
txtNBW.Locked = True

End If
End Sub

Private Sub mnuExit_Click()
frmPara.Visible = False
End Sub

Private Sub mnuMain_Click()

End Sub

Private Sub cmdPrev2_Click()
If diap > 90 Then
frmPrev2a.Show
Else
frmPrev2.Show
End If
End Sub

Private Sub cmdPrint_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.FontBold = False
Printer.Print , "Wave length"; Spc(9);
txtWL.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "focal length"; Spc(10);
txtFC.Text; Spc(3); "cm"
```

```
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Depth"; Spc(16); txtD.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Gain"; Spc(17); txtG.Text;
Spc(3); "dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Beamwidth"; Spc(11);
txtBW.Text; Spc(3); "Degrees"
Printer.EndDoc
End Sub

Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New beamwidth"; Spc(5);
txtNBW.Text; Spc(3); "Degrees"
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Wave length"; Spc(8);
txtWL.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New depth"; Spc(10);
txtND.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New gain"; Spc(11); txtNG.Text;
Spc(3); "dB"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New focal length"; Spc(4);
txtNFC.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "New F/D ratio"; Spc(7);
txtNFD.Text
Printer.EndDoc
End Sub

Private Sub Form_Load()
frmPara.BackColor = vbActiveBorder
fraPara.BackColor = vbActiveBorder
fraNewbeam.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder
txtG.BackColor = vbActiveBorder
txtWL.BackColor = vbActiveBorder
```

```
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub txtBW_Change()
cmdNext.Enabled = (Len(txtBW.Text) > 0)
cmdPrev1.Enabled = (Len(txtBW.Text) > 0)
cmdPrint.Enabled = Len(txtBW.Text) > 0
End Sub

Private Sub txtDia_Change()
dia = Val(txtDia.Text)
If dia >= 1600 Then
Beep
MsgBox ("Diameter must between 20-500 cm"),
(0), ("Invalid data")
txtDia.Text = ""
End If
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid
data")
txtFreq.Text = ""
End If
End Sub

Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0 Or NBW > 30 Then
Beep
txtNBW.Text = ""
End If
End Sub

Private Sub txtNFC_Change()
cmdPrev2.Enabled = (Len(txtNFC.Text) > 0)

End Sub

Private Sub txtNFD_Change()
cmdPrin(2.Enabled = Len(txtNFD.Text) > 0
End Sub

Private Sub cmdClose1_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False

End Sub

Private Sub cmdPrev1_Click()
If diap > 100 And diap <= 500 Then
frmPre1.Show
ElseIf diap > 500 And diap <= 1000 Then
frmPre2.Show
ElseIf diap > 1000 And diap <= 1600 Then
frmPre3.Show
```

```
Else
frmPrev1.Show
End If
End Sub


Private Sub cmdBack_Click()
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub cmdCancel_Click()
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNFD.Text = ""
txtNBW.Locked = False
End Sub

Private Sub cmdClear_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNFC.Text = ""
txtNBW.Locked = False
LabelD1 = ""
End Sub

Private Sub cmdExit_Click()
txtFreq.Text = ""
txtDia.Text = ""
txtFD.Text = ""
txtBW.Text = ""
txtG.Text = ""
txtWL.Text = ""
txtFC.Text = ""
txtD.Text = ""
txtFreq.Locked = False
txtDia.Locked = False
txtFD.Locked = False
frmPara.Visible = False
txtNBW.Text = ""
txtND.Text = ""
txtNG.Text = ""
txtNBW.Locked = False
End Sub


Private Sub cmdNext_Click()
fraPara.Visible = False
fraNewbeam.Visible = True
End Sub


Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtDia.Text = "" Then
MsgBox ("You must insert Diameter data."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text = "0" Then
```

```
MsgBox ("The data can't be zero."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtDia.Text < 20 Then
MsgBox ("It's too small diameter."), (0),
("Invalid data")
txtDia.Text = ""
ElseIf txtFD.Text = "" Then
MsgBox ("You must insert F/D ratio data."), (0),
("Invalid data")
txtFD.Text = ""
ElseIf txtFD.Text = "0" Then
MsgBox ("The data can't be zero."), (0),
("Invalid data")
txtFD.Text = ""
Again:
Msg = "The data error.Try again!"  ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Fp = txtFreq.Text
Fp = Format(Fp, "##0.00")
diap = txtDia.Text
diap = Format(diap, "##0.00")
FDp = txtFD.Text
FDp = Format(FDp, "##0.00")
  txtWL.Text = 30 / Fp
  txtWL.Text = Format$(txtWL.Text,
"##0.00")
  txtFC.Text = FDp * diap
  txtFC.Text = Format$(txtFC.Text, "##0.00")
Dim WLp As Double
FCp = txtFC.Text
FCp = Format(FCp, "##0.00")
WLp = txtWL.Text
  txtD.Text = diap ^ 2 / (16 * FCp)
  txtD.Text = Format$(txtD.Text, "##0.00")
Dim Pi As Double
dp = txtD.Text
Pi = 4 * Atn(1)
  txtG.Text = 10 * Log(2 + (dp * Pi ^ 2 * diap ^
2 * dp ^ 2) / (diap * 9 * WLp ^ 2)) / Log(10)
  txtG.Text = Format$(txtG.Text, "##0.00")
  txtBW.Text = 2 * Atn(WLp * FCp / (dp ^ 2 *
diap)) * (180 / Pi)
  txtBW.Text = Format$(txtBW.Text,
"##0.00")
txtFreq.Locked = True
txtDia.Locked = True
txtFD.Locked = True
LabelD1 = dia

End If
End Sub



Private Sub cmdOK2_Click()
On Error GoTo Again
If txtNBW.Text = "" Then
MsgBox ("You must insert New beamwidth
data."), (0), ("Invalid data")
txtNBW.Text = ""
ElseIf txtNBW.Text = "0" Then
MsgBox ("The value can't be zero"), (0),
("Invalid data")
txtNBW.Text = ""
Again:
Msg = "The data error.Try again!"  ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)
Else
Dim NBWp, Fp, diap, WLp, Pi As Double
NBWp = txtNBW.Text
Fp = txtFreq.Text
diap = txtDia.Text
WLp = txtWL.Text
Pi = 4 * Atn(1)
  NFDp = ((Tan(NBWp * Pi / 360)) * diap ^ 2 /
(WLp * 16 ^ 2)) ^ (1 / 3)
  txtNFD.Text = NFDp
```

```
    NFCp = diap * NFDp
    txtNFC.Text = NFCp
    NDp = diap ^ 2 / (16 * NFCp)
    txtND.Text = NDp
    NGp = 10 * Log(2 + (NDp * Pi ^ 2 * diap ^ 2
* NDp ^ 2) / (diap * 9 * WLp ^ 2)) / Log(10)
    txtNG.Text = NGp
txtNFC.Text = Format(txtNFC.Text, "##0.00")
txtNFD.Text = Format(txtNFD.Text, "##0.00")
txtND.Text = Format(txtND.Text, "##0.00")
txtNG.Text = Format(txtNG.Text, "##0.00")
NFDp = txtNFD.Text
NFDp = Format(NFDp, "##0.00")
NDp = txtND.Text
NDp = Format(NDp, "##0.00")
NFCp = txtNFC.Text
NFCp = Format(NFCp, "##0.00")
txtNBW.Locked = True

End If
End Sub

Private Sub mnuExit_Click()
frmPara.Visible = False
End Sub

Private Sub mnuMain_Click()

End Sub

Private Sub cmdPrev2_Click()
If diap > 90 Then
frmPrev2a.Show
Else
frmPrev2.Show
End If
End Sub

Private Sub cmdPrint_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""    'enter line'
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.FontBold = False
Printer.Print , "Wave length"; Spc(9);
txtWL.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "focal length"; Spc(10);
txtFC.Text; Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Depth"; Spc(16); txtD.Text;
Spc(3); "cm"
Printer.Print , ""    'enter line'
Printer.Print ,
Printer.Print , "Gain"; Spc(17); txtG.Text;
Spc(3); "dB"
Printer.Print , ""    'enter line'
Printer.Print ,
```

```
Printer.Print , "Beamwidth"; Spc(11);
txtBW.Text; Spc(3); "Degrees"
Printer.EndDoc
End Sub

Private Sub cmdPrint2_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Parabolic main reflector"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.FontBold = False
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "Diameter"; Spc(12); txtDia.Text;
Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "F/D ratio"; Spc(12); txtFD.Text
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New beamwidth"; Spc(5);
txtNBW.Text; Spc(3); "Degrees"
Printer.Print , ""   'enter line'
Printer.Print , ""   'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , ""   'enter line'
Printer.FontBold = False
Printer.Print ,
Printer.Print , "Wave length"; Spc(8);
txtWL.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New depth"; Spc(10);
txtND.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New gain"; Spc(11); txtNG.Text;
Spc(3); "dB"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New focal length"; Spc(4);
txtNFC.Text; Spc(3); "cm"
Printer.Print , ""   'enter line'
Printer.Print ,
Printer.Print , "New F/D ratio"; Spc(7);
txtNFD.Text
Printer.EndDoc
End Sub

Private Sub Form_Load()
frmPara.BackColor = vbActiveBorder
fraPara.BackColor = vbActiveBorder
fraNewbeam.BackColor = vbActiveBorder
txtBW.BackColor = vbActiveBorder
txtND.BackColor = vbActiveBorder
txtNG.BackColor = vbActiveBorder
txtNFC.BackColor = vbActiveBorder
txtNFD.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder
txtG.BackColor = vbActiveBorder
txtWL.BackColor = vbActiveBorder
fraNewbeam.Visible = False
fraPara.Visible = True
End Sub

Private Sub txtBW_Change()
cmdNext.Enabled = (Len(txtBW.Text) > 0)
cmdPrev1.Enabled = (Len(txtBW.Text) > 0)
cmdPrint.Enabled = Len(txtBW.Text) > 0
End Sub

Private Sub txtDia_Change()
```

```
dia = Val(txtDia.Text)
If dia > 1600 Then
Beep
MsgBox ("Diameter must between 20-500 cm"),
(0), ("Invalid data")
txtDia.Text  ""
End If
End Sub

Private Sub txtFD_Change()
FD = Val(txtFD.Text)
If FD < 0 Or FD > 1 Then
Beep
txtFD.Text = ""
End If
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid
data")
txtFreq.Text = ""
End If
End Sub

Private Sub txtNBW_Change()
NBW = Val(txtNBW.Text)
If NBW < 0 Or NBW > 30 Then
Beep
txtNBW.Text = ""
End If
End Sub

Private Sub txtNFC_Change()
cmdPrev2.Enabled = (Len(txtNFC.Text) > 0)

End Sub

Private Sub txtNFD_Change()
cmdPrint2.Enabled = Len(txtNFD.Text) > 0
End Sub

Private Sub Command1_Click()

End Sub

Private Sub cmdClose_Click()
Unload frmPrevO1
End Sub

Private Sub cmdPrint_Click()
On Error GoTo shutdown
Printer.Orientation = 1
Printer.PaperSize = vbPRPSA4
Printer.FontSize = 16
Printer.Print , ""
Printer.FontBold = True
Printer.Print , "Offset-fed reflector"
Printer.FontBold = False
Printer.Print , "Diameter"; Spc(7); diao; Spc(3);
"cm"
Printer.Print , "Depth(d)"; Spc(7); doo; Spc(3);
"cm"
Printer.Print , "F/D ratio"; Spc(7); FDo
Printer.Print , "Focal length(F)"; Spc(2); FCo;
Spc(3); "cm"
Printer.ScaleTop = 71.9
Printer.ScaleWidth = 50
Printer.ScaleHeight = -71.9
Printer.ScaleLeft = 0
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.DrawWidth = 3
For Y = ((-ddp / 2) - (diao / 2)) To ((-ddp / 2) +
(diao / 2)) Step 0.01
X = Y ^ 2 / (FCo * 4)
Printer.PSet ((Y + 45), (X + 31.75))
Next Y
Printer.Line ((40.2 - depth), (diao / 2 + ddp / 2 +
26.625))-((40.2 - depthl), (Dtl + 26.625))
'dimension
Printer.DrawWidth = 1
```

```
Printer.Line (5, 26.625)-(40, 26.625)
Printer.Line ((40 - FCo), 26.625)-((40 - FCo),
27)
Printer.Line (43, (26.625 + diao / 2 + ddp / 2))-
(43, (26.625 - diao / 2 + ddp / 2))
Printer.Line (42, (26.625 + diao / 2 + ddp / 2))-
(44, (26.625 + diao / 2 + ddp / 2))
Printer.Line (42, (26.625 - diao / 2 + ddp / 2))-
(44, (26.625 - diao / 2 + ddp / 2))
Printer.CurrentX = 42
Printer.CurrentY = 26.625 + diao / 2
Printer.Print "Diameter"
Printer.Line (40, 25)-((40 - FCo), 25)
Printer.Line (40, 24)-(40, 26)
Printer.Line ((40 - FCo), 24)-((40 - FCo), 26)
Printer.CurrentX = 40 * (FCo / 2)
printercurrenty = 26
Printer.Print "F"
Printer.DrawWidth = 3

'border
Printer.Line (1, 1)-(49, 1)
Printer.Line (1, 1)-(1, 70.9)
Printer.Line (1, 70.9)-(49, 70.9)
Printer.Line (49, 70.9)-(49, 1)
Printer.Line (1, 6)-(49, 6)
Printer.Line (1, 3.5)-(49, 3.5)
Printer.Line (1, 3.5)-(18, 3.5)
Printer.Line (18, 3.5)-(18, 1)
Printer.Line (18, 3.5)-(49, 3.5)
Printer.Line (35, 3.5)-(35, 1)
Printer.Line (42, 3.5)-(42, 1)
Printer.FontSize = 14
Printer.CurrentX = 0
Printer.CurrentY = 3
Printer.Print Spc(4); "Design by:"; Spc(40);
"Title: Offset-fed reflector"; Spc(24); "Date: ";
Date; Spc(4); "Scale: 1:2.5"
Printer.CurrentX = 12.5
Printer.CurrentY = 5.5
Printer.Print "Suranaree University of
Technology school of Telecommunication
Engineering"
Printer.EndDoc
shutdown:
Printer.EndDoc
End Sub


Private Sub Form_Load()
lblFocal.Left = (FCo / 2) + 15
txtDia.Text = diao
txtFC.Text = FCo
txtDepth.Text = doo
txtWe.Text = WWeo
txtWo.Text = WWoo
lblD.Top = 40 + (ddp / 2)
lblD.Left = 10
'background
Frame1.BackColor = vbActiveBorder
frmPrevO1.BackColor = vbActiveBorder
txtDia.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtDepth.BackColor = vbActiveBorder
txtWe.BackColor = vbActiveBorder
txtWo.BackColor = vbActiveBorder
OLE3.BackColor = vbActiveBorder
OLE4.BackColor = vbActiveBorder
Label2.BackColor = vbActiveBorder
Label6.BackColor = vbActiveBorder
Label3.BackColor = vbActiveBorder

End Sub

Private Sub Form_Paint()
Picture1.Line (15, 98)-(15, 2), QBColor(8)
For Y = (ddp / 2 - diao / 2) To (ddp / 2 + diao /
2) Step 0.01
X = Y ^ 2 / (4 * FCo)
Picture1.DrawWidth = 1
Picture1.PSet ((X + 15), (Y + 40)), QBColor(1)
Next Y
'----------------
Picture1.Line ((FCo + 15), 42)-((FCo + 15), 40),
QBColor(0)
```

```
Picture1.Line (15, 30)-((FCo + 15), 30),
QBColor(9)
Picture1.Line ((15 + FCo), 28)-((15 + FCo), 32),
QBColor(9)
'dimension of dia
Picture1.Line (10, (40 + diao / 2 + ddp / 2))-(10,
(40 - diao / 2 + ddp / 2))
Picture1.Line (11, (40 + diao / 2 + ddp / 2))-(9,
(40 + diao / 2 + ddp / 2))
Picture1.Line (11, (40 - diao / 2 + ddp / 2))-(9,
(40 - diao / 2 + ddp / 2))
'scale
VX1 = 14.5
VX2 = 15
Dim VY1 As Integer
For VY1 = 2 To 98 Step 1
VY2 = VY1
Picture1.Line (VX1, VY1)-(VX2, VY2)
Next VY1
For VYY = 0 To 98 Step 5
Picture1.Line (14, VYY)-(VX2, VYY),
QBColor(9)
Next VYY
Dim HX1 As Integer
  For HX1 = 2 To 98 Step 1
  HX2 = HX1
  HY1 = 40
  HY2 = 39.5
  Picture1.Line (HX1, HY1)-(HX2, HY2)
Next HX1

For HXX = 0 To 98 Step 5
Picture1.Line (HXX, HY1)-(HXX, 39),
QBColor(9)
Next HXX
End Sub



Private Sub Label7_Click()


End Sub


Private Sub cmdBack_Click()
Unload frmPre1
End Sub


Private Sub cmdPrint_Click()
Printer.Orientation = 1
Printer.PaperSize = vbPRPSA4
Printer.Print , ""
Printer.FontSize = 14
Printer.Print , ""
Printer.FontBold = True
Printer.Print , , "Parabolic main reflector"
Printer.FontBold = False
Printer.Print , , "Frequency"; Spc(5); Fp; Spc(3);
"GHz"
Printer.Print , , "Diameter"; Spc(7); diap; Spc(3);
"cm"
Printer.Print , , "Depth"; Spc(9); dp; Spc(3);
"cm"
Printer.Print , , "F/D ratio"; Spc(7); FDp
Printer.Print , , "Focal length"; Spc(2); FCp;
Spc(3); "cm"
Printer.ScaleWidth = 50
Printer.ScaleHeight = -71.9
Printer.ScaleLeft = 0
Printer.ScaleTop = 71.9
'Grid
Printer.DrawWidth = 1
For G = 0 To (dp / 10) + 5 Step 2.5
Printer.Line (31.77 - G, 36.75 + diap / 20)-(31.77
- G, 36.75 - diap / 20)
Next G
For Gx = 36.75 - diap / 20 To 36.75 + diap / 20
Step 2.5
Printer.Line (34, Gx)-(30 - dp / 10, Gx)
Next Gx
Printer.DrawWidth = 3
Printer.CurrentX = 0
Printer.CurrentY = 0
Dim Y, X As Single
For Y = (-diap / 20) To (diap / 20) Step 0.01
X = Y ^ 2 / (4 * FCp / 10)
Printer.PSet ((Y + 35), (X + 40))
Next Y
```

```
'dimension
Printer.Line (35, (36.75 + diap / 20))-(35, (36.75
- diap / 20))
Printer.Line (33, (36.75 + diap / 20))-(37, (36.75
+ diap / 20))
Printer.Line (33, (36.75 - diap / 20))-(37, (36.75
- diap / 20))
Printer.Line ((31.77 - dp / 10), (36.75 + diap /
20))-((31.77 - dp / 10), (36.75 - diap / 20))
Printer.Line ((31.77 - FCp / 10), 36.75)-((31.77 -
FCp / 10), 37.75)
Printer.DrawWidth = 1
Printer.Line (5, 36.75)-(40, 36.75)
Printer.DrawWidth = 3
'-----------------
Printer.CurrentX = 33
Printer.CurrentY = 40
Printer.FontSize = 16
Printer.Print "Y = sqr(4FX)"
Printer.CurrentX = 33
Printer.CurrentY = 37
Printer.Print "Diameter = "; diap; "cm"
'Border
Printer.ScaleTop = 71.9
Printer.ScaleWidth = 50
Printer.ScaleHeight = -71.9
Printer.ScaleLeft = 0
Printer.Line (1, 1)-(49, 1)
Printer.Line (1, 1)-(1, 70.9)
Printer.Line (1, 70.9)-(49, 70.9)
Printer.Line (49, 70.9)-(49, 1)
Printer.Line (1, 6)-(49, 6)
Printer.Line (1, 3.5)-(49, 3.5)
Printer.Line (1, 3.5)-(18, 3.5)
Printer.Line (18, 3.5)-(18, 1)
Printer.Line (18, 3.5)-(49, 3.5)
Printer.Line (35, 3.5)-(35, 1)
Printer.Line (42, 3.5)-(42, 1)
Printer.FontSize = 14
Printer.CurrentX = 0
Printer.CurrentY = 3
Printer.Print Spc(4); "Design by:"; Spc(40);
"Title: Parabolic main reflector"; Spc(20); "Date:
"; Date; Spc(4); "Scale: 1:25"
Printer.CurrentX = 12.5
Printer.CurrentY = 5.5
Printer.Print "Suranaree University of
Technology school of Telecommunication
Engineering"

Printer.EndDoc
End Sub

Private Sub Form_Load()
Dim M As Single
M = (diap / 20) ^ 2 / (4 * FCp / 10) 'Depth
X1 = M + 8
Label1.Left = 11
Label1.Top = 51
txtDia.Text = diap
txtD.Text = dp
txtFC.Text = FCp
lblFC.Left = (FCp / 20) + 14
lblFC.Top = 47

'background
txtDia.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder


End Sub

Private Sub Form_Paint()
Picture1.Line (15, 98)-(15, 2), QBColor(8)
Picture1.Line (2, 50)-(98, 50), QBColor(8)
Dim Y, P, M As Single
For Y = (-diap / 20) To (diap / 20) Step 0.1
X = Y ^ 2 / (4 * FCp / 10)
Picture1.PSet ((X + 15), (Y + 50))
Next Y
'-----------------
M = (diap / 20) ^ 2 / (4 * FCp / 10)
X1 = M + 8 + 15
```

```
Y1 = (diap / 20) + 50
Y2 = (-diap / 20) + 50
'dimension of diameter
Picture1.Line (12, Y1)-(12, Y2), QBColor(4)
Picture1.Line (10, Y1)-(14, Y1), QBColor(4)
Picture1.Line (10, Y2)-(14, Y2), QBColor(4)
'dimension of focal length
Picture1.Line (15, 45)-((FCp / 10 + 15), 45),
QBColor(4)
Picture1.Line ((FCp / 10 + 15), 47)-((FCp / 10 +
15), 43), QBColor(4)
'scale
VX1 = 14.5
VX2 = 15
Dim VY1 As Integer
For VY1 = 2 To 98 Step 1
VY2 = VY1
Picture1.Line (VX1, VY1)-(VX2, VY2)
Next VY1
For VYY = 0 To 98 Step 5
Picture1.Line (14, VYY)-(VX2, VYY),
QBColor(9)
Next VYY
Dim HX1 As Integer
  For HX1 = 3 To 98 Step 1
  HX2 = HX1
  HY1 = 50
  HY2 = 49.5
  Picture1.Line (HX1, HY1)-(HX2, HY2)
Next HX1

For HXX = 0 To 98 Step 5
Picture1.Line (HXX, HY1)-(HXX, 49),
QBColor(9)
Next HXX
'Grid
Dim GY1 As Integer
For GY1 = 36.75 + diap / 20 To 36.75 - diap / 20
Step 1
Picture1.Line (31.77, GY1)-(31.77 - dp / 10,
GY1)
Next GY1
End Sub

Private Sub cmdBack_Click()
Unload frmPre2
Unload frmPre1
Unload frmPrev1
End Sub

Private Sub cmdPrint_Click()
Printer.Orientation = 1
Printer.PaperSize = vbPRPSA4
Printer.Print , ""
Printer.FontSize = 14
Printer.Print , ""
Printer.FontBold = True
Printer.Print , , "Parabolic main reflector"
Printer.FontBold = False
Printer.Print , , "Frequency"; Spc(5); Fp; Spc(3);
"GHz"
Printer.Print , , "Diameter"; Spc(7); diap; Spc(3);
"cm"
Printer.Print , , "Depth"; Spc(9); dp; Spc(3);
"cm"
Printer.Print , , "F/D ratio"; Spc(7); FDp
Printer.Print , , "Focal length"; Spc(2); FCp;
Spc(3); "cm"
Printer.ScaleWidth = 50
Printer.ScaleHeight = -71.9
Printer.ScaleLeft = 0
Printer.ScaleTop = 71.9
'Grid
Printer.DrawWidth = 1
For G = 0 To (dp / 20) + 5 Step 2.5
Printer.Line (31.77 - G, 36.75 + diap / 40)-(31.77
- G, 36.75 - diap / 40)
Next G
For Gx = 36.75 - diap / 40 To 36.75 + diap / 40
Step 2.5
Printer.Line (34, Gx)-(30 - dp / 20, Gx)
Next Gx
Printer.DrawWidth = 3
Printer.CurrentX = 0
Printer.CurrentY = 0
Dim Y, X As Single
For Y = (-diap / 40) To (diap / 40) Step 0.01
```

```
X = Y ^ 2 / (4 * FCp / 20)
 Printer.PSet ((Y + 35), (X + 40))
Next Y
'dimension
Printer.Line (35, (36.75 + diap / 40))-(35, (36.75
- diap / 40))
Printer.Line (33, (36.75 + diap / 40))-(37, (36.75
+ diap / 40))
Printer.Line (33, (36.75 - diap / 40))-(37, (36.75
- diap / 40))
Printer.Line ((31.77 - dp / 20), (36.75 + diap /
40))-((31.77 - dp / 20), (36.75 - diap / 40))
Printer.Line ((31.77 - FCp / 20), 36.75)-((31.77 -
FCp / 20), 37.75)
Printer.DrawWidth = 1
Printer.DrawStyle = 1
Printer.Line (2, 36.75)-(40, 36.75)
Printer.DrawStyle = 0
Printer.DrawWidth = 3
'-----------------
Printer.CurrentX = 33
Printer.CurrentY = 40
Printer.FontSize = 16
Printer.Print "Y = sqr(4FX)"
Printer.CurrentX = 33
Printer.CurrentY = 37
Printer.Print "Diameter = "; diap; "cm"
'Border
Printer.ScaleTop = 71.9
Printer.ScaleWidth = 50
Printer.ScaleHeight = -71.9
Printer.ScaleLeft = 0
Printer.Line (1, 1)-(49, 1)
Printer.Line (1, 1)-(1, 70.9)
Printer.Line (1, 70.9)-(49, 70.9)
Printer.Line (49, 70.9)-(49, 1)
Printer.Line (1, 6)-(49, 6)
Printer.Line (1, 3.5)-(49, 3.5)
Printer.Line (1, 3.5)-(18, 3.5)
Printer.Line (18, 3.5)-(18, 1)
Printer.Line (18, 3.5)-(49, 3.5)
Printer.Line (35, 3.5)-(35, 1)
Printer.Line (42, 3.5)-(42, 1)
Printer.FontSize = 14
Printer.CurrentX = 0
Printer.CurrentY = 3
Printer.Print Spc(4); "Design by:"; Spc(40);
"Title: Parabolic main reflector"; Spc(20); "Date:
"; Date; Spc(4); "Scale: 1:50"
Printer.CurrentX = 12.5
Printer.CurrentY = 5.5
Printer.Print "Suranaree University of
Technology school of Telecommunication
Engineering"
Printer.EndDoc
End Sub

Private Sub Form_Load()
Dim M As Single
M = (diap / 2) ^ 2 / (4 * FCp / 20) 'Depth
X1 = M + 8
Label1.Left = 11
Label1.Top = 51
txtDia.Text = diap
txtD.Text = dp
txtFC.Text = FCp
lblFC.Left = (FCp / 40) + 14
lblFC.Top = 47
'background
txtDia.BackColor = vbActiveBorder
txtFC.BackColor = vbActiveBorder
txtD.BackColor = vbActiveBorder


End Sub


Private Sub Form_Paint()
Picture1.Line (15, 98)-(15, 2), QBColor(8)
Picture1.Line (2, 50)-(98, 50), QBColor(8)
Dim Y, P, M As Single
For Y = (-diap / 40) To (diap / 40) Step 0.1
X = Y ^ 2 / (4 * FCp / 20)
 Picture1.PSet ((X + 15), (Y + 50))
Next Y
'----------------
```

```
M = (diap / 40) ^ 2 / (4 * FCp / 20)
X1 = M + 8 + 15
Y1 = (diap / 40) + 50
Y2 = (-diap / 40) + 50
'dimension of diameter
Picture1.Line (12, Y1)-(12, Y2), QBColor(4)
Picture1.Line (10, Y1)-(14, Y1), QBColor(4)
Picture1.Line (10, Y2)-(14, Y2), QBColor(4)
'dimension of focal length
Picture1.Line (15, 45)-((FCp / 20 + 15), 45),
QBColor(4)
Picture1.Line ((FCp / 20 + 15), 47)-((FCp / 20 +
15), 43), QBColor(4)
'scale
VX1 = 14.5
VX2 = 15
Dim VY1 As Integer
For VY1 = 2 To 98 Step 1
VY2 = VY1
Picture1.Line (VX1, VY1)-(VX2, VY2)
Next VY1
For VYY = 0 To 98 Step 5
Picture1.Line (14, VYY)-(VX2, VYY),
QBColor(9)
Next VYY
Dim HX1 As Integer
  For HX1 = 3 To 98 Step 1
  HX2 = HX1
  HY1 = 50
  HY2 = 49.5
  Picture1.Line (HX1, HY1)-(HX2, HY2)
Next HX1

For HXX = 0 To 98 Step 5
Picture1.Line (HXX, HY1)-(HXX, 49),
QBColor(9)
Next HXX
End Sub

Private Sub cmdCancel_Click()
txtFreq.Text = ""
txtG.Text = ""
txtH.Text = ""
txtW.Text = ""
txtRe.Text = ""
txtRh.Text = ""
txtL.Text = ""
txtEPC.Text = ""
txtHPC.Text = ""
txtEdB.Text = ""
txtHdB.Text = ""
End Sub

Private Sub cmdClose_Click()
txtFreq.Text = ""
txtG.Text = ""
txtH.Text = ""
txtW.Text = ""
txtRe.Text = ""
txtRh.Text = ""
txtL.Text = ""
txtEPC.Text = ""
txtHPC.Text = ""
txtEdB.Text = ""
txtHdB.Text = ""
frmRecfh.Visible = False
End Sub


Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtFreq.Text = ""
ElseIf txtFreq.Text < 4 Then
MsgBox ("Frequency must not less than 4
GHz"), (0), ("Invalid data")
txtFreq.Text = ""
ElseIf txtG.Text = "" Then
MsgBox ("You must insert Gain of feed horn
data."), (0), ("Invalid data")
ElseIf txtG.Text = "0" Then
```

```
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtG.Text = ""
ElseIf txtG.Text < 12 Then
MsgBox ("The gain of feed horn must not
lessthan 12 dB."), (0), ("Invalid data")
txtG.Text = ""
Again:
Msg = "The data error.Try again!"   ' Define
message.
Style = vbOKOnly + vbCritical  ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Dim F, G, Pi As Single
Fr = txtFreq.Text
G = txtG.Text
Pi = 4 * Atn(1)
   L = 30 / Fr        'L=wavelength'
   a = L / 2          'Dimension of waveguide'
   b = a / 2          '        "--"      '
   a = Format(a, "##0.00")
   b = Format(b, "##0.00")
   Gd = 10 ^ (G / 10)  'Gd=gain require,not dB'
   W = 0.489 * L * Sqr(Gd)
   h = 0.332 * L * Sqr(Gd)
   Rh = 0.0746 * L * Gd
   RE = 0.0531 * L * Gd
   Lh = (W - a) / W * Sqr(Rh ^ 2 - (W ^ 2 / 4))
   Le = Lh
   RE = h / (h - b) * Sqr(Lh ^ 2 + ((h - b) ^ 2 /
4))
   Gactual = 10 * Log(4 * Pi * h * W / L ^ 2) /
Log(10) - 0.91 - 1.14 - 0.96
   Gdnew = G * G / (Gactual)   'dB'
   Gdn = 10 ^ (Gdnew / 10)    'not dB'
   Gdn = Format$(Gdn, "##0.00")
   txtW.Text = (0.489 * Sqr(Gdn) * L)
   txtH.Text = (0.332 * Sqr(Gdn) * L)
   txtRh.Text = (0.0746 * Gdn * L)
   txtRe.Text = (0.0531 * Gdn * L)
txtW.Text = Format$(txtW.Text, "##0.00")
txtH.Text = Format$(txtH.Text, "##0.00")
txtRh.Text = Format$(txtRh.Text, "##0.000")
txtRe.Text = Format$(txtRe.Text, "##0.000")

WWr = txtW.Text 'This is W
RRhr = txtRh.Text 'This is Rh
   txtL.Text = Sqr(RRhr ^ 2 - (WWr / 2) ^ 2) *
(WWr - a) / WWr
txtL.Text = Format$(txtL.Text, "##0.000")
LLr = txtL.Text 'This is L
Dim X, C As Single
   HHr = txtH.Text 'This is H
   X = 0.6951 * (L / WWr)
   C = Atn(X / Sqr(-X * X + 1))
   txtHdB.Text = 2 * C * 180 / Pi
   txtHdB.Text = Format$(txtHdB.Text,
"##0.00")
   Y = 0.4735 * L / h
   d = Atn(Y / Sqr(-Y * Y + 1))
   txtEdB.Text = 2 * d * 180 / Pi
   txtEdB.Text = Format$(txtEdB.Text,
"##0.00")
RRer = txtRe.Text 'This is Re
   txtHPC.Text = 0.508 * RRhr
   txtEPC.Text = 0.494 * RRer
   txtHPC.Text = Format$(txtHPC.Text,
"##0.000")
   txtEPC.Text = Format$(txtEPC.Text,
"##0.000")
End If
End Sub

Private Sub mnuMain_Click()
frmChoise.Show
frmRecfh.Visible = False
End Sub

Private Sub cmdPrev_Click()
If (LLr + WWr + 3) >= 10 And (LLr + WWr +
3) < 40 Then
frmPrevRF1.Show
ElseIf (LLr + WWr + 3) >= 40 Then
frmPrevRF2.Show
```

```
Else
frmPrevRFH.Show
End If
End Sub



Private Sub cmdPrint_Click()
Printer.CurrentX = 0
Printer.CurrentY = 0
Printer.Orientation = 1
Printer.FontSize = 16
Printer.Print , ""
Printer.Print , ""
Printer.FontBold = True
Printer.FontUnderline = True
Printer.Print , "Rectangular feed horn"
Printer.Print , ""
Printer.FontUnderline = False
Printer.Print , Spc(5); "Give data"
Printer.Print , ""    'enter line'
Printer.FontBold = False
Printer.Print , , "Frequency"; Spc(10);
txtFreq.Text; Spc(3); "GHz"
Printer.Print , ""    'enter line'
Printer.Print , , "Gain of feedhorn"; Spc(4);
txtG.Text; Spc(3); "dB"
Printer.Print , ""    'enter line'
Printer.FontBold = True
Printer.Print , Spc(5); "Get data"
Printer.Print , Spc(5); "Aperture dimension"
Printer.Print , ""
Printer.FontBold = False
Printer.Print , , "W"; Spc(18); txtW.Text;
Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "Re"; Spc(17); txtRe.Text;
Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "L"; Spc(18); txtL.Text; Spc(3);
"cm"
Printer.Print , ""
Printer.Print , , "H"; Spc(18); txtH.Text; Spc(3);
"cm"
Printer.Print , ""
Printer.Print , , "Rh"; Spc(17); txtRh.Text;
Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "3-dB beamwidth in H-plane";
Spc(4); txtHdB.Text; Spc(3); "dB"
Printer.Print , ""
Printer.Print , , "3-dB beamwidth in E-plane";
Spc(4); txtEdB.Text; Spc(3); "dB"
Printer.Print , ""
Printer.Print , , "H-plane phase center"; Spc(12);
txtHPC.Text; Spc(3); "cm"
Printer.Print , ""
Printer.Print , , "E-plane phase center"; Spc(12);
txtEPC.Text; Spc(3); "cm"
Printer.EndDoc
End Sub

Private Sub Form_Load()
fraRecfh.BackColor = vbActiveBorder
txtW.BackColor = vbActiveBorder
txtRe.BackColor = vbActiveBorder
txtL.BackColor = vbActiveBorder
txtRh.BackColor = vbActiveBorder
txtH.BackColor = vbActiveBorder
txtEdB.BackColor = vbActiveBorder
txtHdB.BackColor = vbActiveBorder
txtEPC.BackColor = vbActiveBorder
txtHPC.BackColor = vbActiveBorder
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq > 50 Then
Beep
MsgBox ("Too much Frequency"), (0), ("Invalid
data")
txtFreq.Text = ""
End If
End Sub

Private Sub txtG_Change()
G = Val(txtG.Text)
```

```
If G > 30 Then
Beep
MsgBox ("Too much Gain"), (0), ("Invalid
data")
txtG.Text = ""
End If
End Sub


Private Sub txtHdB_Change()
cmdPrev.Enabled = Len(txtHdB.Text) > 0

End Sub



Private Sub txtHPC_Change()
cmdPrev.Enabled = Len(txtHPC.Text) > 0
End Sub



Private Sub txtW_Change()
cmdPrint.Enabled = Len(txtW.Text) > 0
End Sub

Private Sub cmdCancel_Click()
txtFreq.Text = ""
txta.Text = ""
txtb.Text = ""

End Sub

Private Sub cmdClose_Click()
txtFreq.Text = ""
txta.Text = ""
txtb.Text = ""
frmRecwg.Visible = False

End Sub


Private Sub cmdOK_Click()
On Error GoTo Again
If txtFreq.Text = "" Then
MsgBox ("You must insert Frequency data."),
(0), ("Invalid data")
ElseIf txtFreq.Text = "0" Then
MsgBox ("The data can't be zero"), (0),
("Invalid data")
txtFreq.Text = ""
Again:
Msg = "The data error.Try again!"  ' Define
message.
Style = vbOKOnly + vbCritical ' Define buttons.
Title = "Invalid data"  ' Define title.
Response = MsgBox(Msg, Style, Title)

Else
Dim F As Double
F = txtFreq.Text
  L = 30 / F
  txta.Text = L / 2
  txta.Text = Format$(txta.Text, "##0.00")
Dim a As Double
a = txta.Text
  txtb.Text = a / 2
  txtb.Text = Format$(txtb.Text, "##0.00")
End If
  End Sub

Private Sub OLE1_Updated(Code As Integer)

End Sub


Private Sub mnuMain_Click()
frmChoise.Show
frmRecwg.Visible = False
End Sub

Private Sub mnuClock_Click()
frmClock.WindowState = 1
frmClock.Show
End Sub

Private Sub mnuClose_Click()
txtFreq.Text = ""
```

```
txta.Text = ""
txtb.Text = ""
frmRecwg.Visible = False
End Sub


Private Sub Form_Load()
txta.BackColor = vbActiveBorder
txtb.BackColor = vbActiveBorder
End Sub

Private Sub txtFreq_Change()
freq = Val(txtFreq.Text)
If freq < 1 Or freq > 300 Then
Beep
txtFreq.Text = ""
End If
End Sub
```

ประวัติผู้เขียน

ชื่อ	นายจิรพัฒน์ กิตติเมธี
	นางสาวศศิวิมล วงศ์สุนทร
โครงงาน	โปรแกรมออกแบบจานรับสัญญาณดาวเทียม
สาขาวิชา	วิศวกรรมโทรคมนาคม

ประวัติ

นายจิรพัฒน์ กิตติเมธี เกิดเมื่อวันที่ 19 พฤษภาคม พ.ศ. 2517 สถานที่เกิด อ. ศรีสำโรง จ. สุโขทัย เข้าศึกษาระดับประถมศึกษาที่ โรงเรียนบ้านคลองตาลกระจ่างจินดา ระดับมัธยมศึกษาที่โรงเรียนศรีสำโรงชนูปถัมภ์ เข้าศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2536

นางสาวศศิวิมล วงศ์สุนทร เกิดเมื่อวันที่ 24 ธันวาคม พ.ศ. 2518 ภูมิลำเนา อ. เมือง จ. นครราชสีมา เข้าศึกษาระดับประถมศึกษาที่ โรงเรียนวัดสระแก้ว ระดับมัธยมศึกษาที่โรงเรียนสุรนารีวิทยา เข้าศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2536

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี