



การทดสอบสายอากาศแบบสนามระยะใกล้
NEAR-FIELD ANTENNA MEASUREMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ระบบที่วารสารของคุณเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปีการศึกษา 2537

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง 034706

ปริญญาบัตรปีการศึกษา 2537

ภาควิชา วิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การทดสอบสายอากาศแบบสนามระยะใกล้

ผู้จัดทำ

1. นาย ศรพล ภัทรภากรกุล

2. นาย สุภชัย อุดลประเสริฐสุข

2. นาย สมिति โกสิยเจริญ



..... อาจารย์ที่ปรึกษา

(รศ.ดร. โมไนย ไกรฤกษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบสายอากาศแบบสนามระยะใกล้
NEAR-FIELD ANTENNA MEASUREMENT

โดย นาย ศรพล ภัทรภากรกุล
นาย ศุภชัย อุดลประเสริฐสุข
นาย สมिति โกสัยเจริญ

อาจารย์ที่ปรึกษา รศ.ดร.โมไนย ไกรฤกษ์

บทคัดย่อ

ในรายงานฉบับนี้เป็นการศึกษาและทำการทดลองวัดสายอากาศแบบสนามระยะใกล้ โดยใช้เครื่องมือวิเคราะห์โครงข่าย คอมพิวเตอร์ และ สแกนเนอร์ ทำการหารูปแบบการแพร่กระจายของสายอากาศ และสรุปผลการเปรียบเทียบกับทางทฤษฎี โดยใช้หลักการของสมการคลื่นมาทำการวิเคราะห์ ด้วยฟูเรียร์ทรานฟอร์ม และการเขียนโปรแกรมควบคุมสแกนเนอร์ให้ทำการเคลื่อนโพรับไปตามจุดต่างๆ และควบคุมการรับส่งข้อมูลระหว่างสแกนเนอร์ คอมพิวเตอร์ และเครื่องมือวิเคราะห์โครงข่าย ซึ่งสามารถควบคุมการทำงานทั้งหมดได้ที่คอมพิวเตอร์

ABSTRACT

THIS REPORT IS CASE OF STUDYING AND TESTING OF NEAR-FIELD ANTENNA MEASUREMENT. BY THE NETWORK ANALYZER, COMPUTER AND SCANNER, WE CAN FIND PATTERN OF ANTENNA TO COMPARE WITH THE RESULT OF THEORY THAT USE THE PRINCIPLE OF WAVE EQUATION. COMPUTER, NETWORK ANALYZER AND SCANNER ARE CONTROLLED BY SOFTWARE. IT CONTROL THE ACTIVITY OF EACH INSTRUMENTS IT CAN MANIPULATE AND PLOT PATTERN OF ANTENNA UNDER TEST.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 ทฤษฎีการทดสอบสายอากาศแบบสนามระยะใกล้ (Near-field Antenna Measurement)	1
บทที่ 2 โปรแกรมที่ใช้ในการวัดแพทเทิร์นของสายอากาศ	17
บทที่ 3 ผลการทดลอง	85
บทที่ 4 สรุปและวิจารณ์ผลการทดลอง	96
บทแทรกที่ 1 มาตรฐานการเชื่อมต่อแบบ GPIB	97
บทแทรกที่ 2 การใช้เครื่องวิเคราะห์โครงข่ายสำหรับการวัดแบบส่งผ่าน	114
บทแทรกที่ 3 สแกนเนอร์	121
เอกสารอ้างอิง	129
กิตติกรรมประกาศ	130

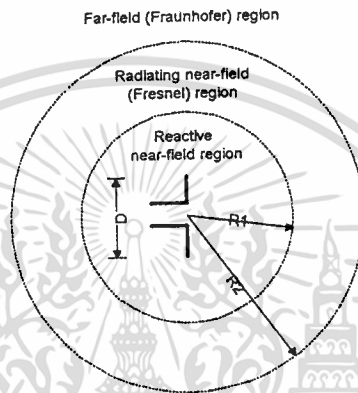


บทที่ 1

ทฤษฎีการทดสอบสายอากาศแบบสนามระยะใกล้

บริเวณสนาม

พื้นที่รอบๆ สายอากาศ สามารถแบ่งออกได้เป็น 3 บริเวณ ได้แก่ รีแอกทีฟ เนียร์ฟิลด์ (reactive near-field), เรดิเอติง เนียร์ฟิลด์ (radiating near-field ; Fresnel) และ ฟาร์ฟิลด์ (far-field ; Fraunhofer) แสดงได้ดังรูปที่ 1



$$R1 = 0.62\sqrt{D^3/\lambda}$$

$$R2 = 2D^2/\lambda$$

รูปที่ 1 บริเวณสนามของสายอากาศ

บริเวณรีแอกทีฟ เนียร์ฟิลด์ (reactive near-field region)

เป็นบริเวณสนามที่อยู่รอบๆ ใกล้ตัวสายอากาศมากที่สุด ซึ่งบริเวณนี้พลังงานสนามแม่เหล็กไฟฟ้าจะลดลงอย่างรวดเร็วเมื่อระยะห่างจากสายอากาศเพิ่มขึ้น จึงแทบจะไม่มีกระแสกระจายคลื่นเกิดขึ้นในบริเวณนี้ สำหรับสายอากาศโดยทั่วไป ขอบเขตของบริเวณนี้จะกำหนดไว้ที่ระยะ $R < 0.62\sqrt{D^3/\lambda}$ จากพื้นผิวของสายอากาศ เมื่อ λ แทนความยาวคลื่น และ D แทนขนาดที่ใหญ่ที่สุดของสายอากาศ

บริเวณเรดิเอติงเนียร์ฟิลด์ (radiating near-field region)

เป็นบริเวณสนามที่มีการแพร่กระจายคลื่นสัมพันธ์กับระยะทางจากสายอากาศโดยเราจะทำการวัดสนามระยะใกล้ของสายอากาศในบริเวณนี้และแปลงข้อมูลที่ได้จากการวัดโดยใช้การแปลงทางคณิตศาสตร์ ไปเป็นข้อมูลของสนามระยะไกล สำหรับขอบเขตบริเวณนี้อยู่ระหว่างระยะ $R \geq 0.62\sqrt{D^3/\lambda}$ กับระยะ $R < 2D^2/\lambda$

บริเวณฟาร์ฟิลด์ (far-field region)

เป็นบริเวณสนามที่มีการแจกแจงเชิงมุมไม่ขึ้นกับระยะทาง จากสายอากาศ ดังนั้นจึงใช้บริเวณนี้สำหรับทดสอบการวัดโดยตรงจากสนามระยะไกล ซึ่งจะวัดที่ระยะห่าง $R > 2D^2/\lambda$ จากสายอากาศ สำหรับขอบเขตบริเวณนี้จะเริ่มตั้งแต่ $R = 2D^2/\lambda$ ไปจนถึงจุดอินฟินิตี้ (infinity)

การแบ่งขอบเขตสนาม

ก่อนที่จะพยายามหาสนามการแพร่กระจายของไดโพลที่มีความยาวจำกัดในแต่ละความยาวนั้นจำเป็นต้องรู้ลักษณะของสนามทั้ง 3 ก่อนดังที่ได้กล่าวไว้แล้ว จาก สมการ

$$A(x,y,z) = (\mu/4) \int_c I_e(x',y',z') (e^{-jkR}/R) dl'$$

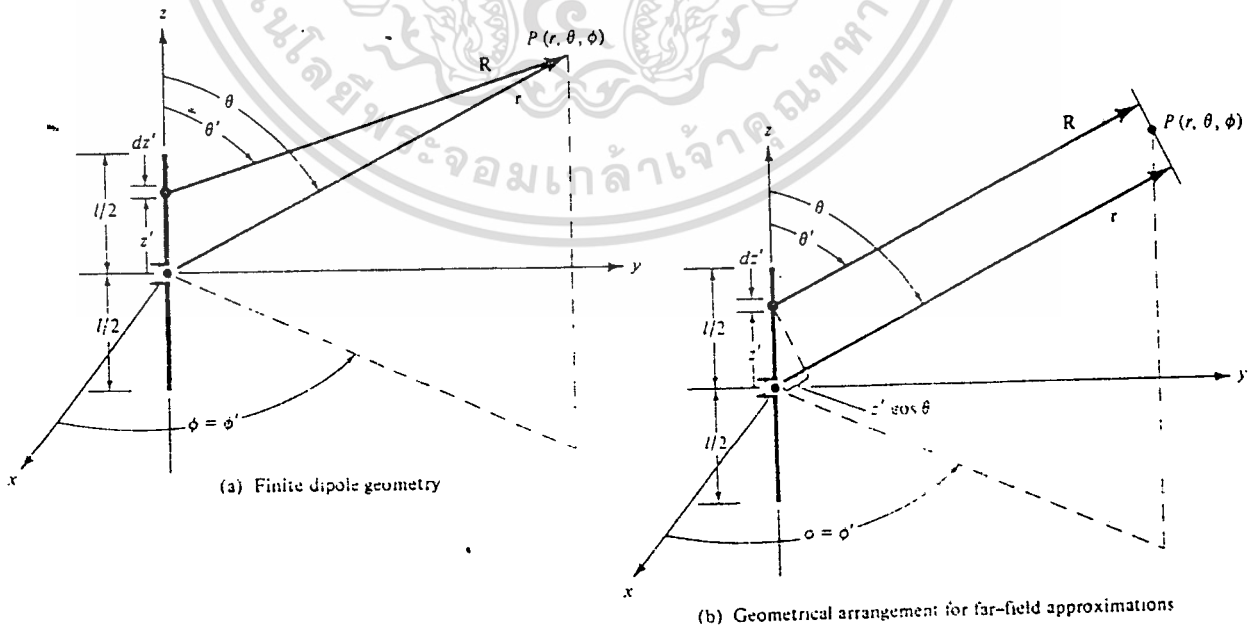
ซึ่ง

$$R = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}$$

ในเสาอากาศแบบ ไดโพลจิว ($l \ll \lambda$) เราจะได้ $R = r$ และในไดโพลเล็ก

($l \leq \lambda/10$)

เราจะ ได้ $R \approx r$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อรูปที่ 2a, 2b นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะพิจารณาจากไดโพลสมมาตรตัวหนึ่งที่มีความยาวเป็น l วางที่ตำแหน่งจุดกำเนิดโดยวางตามแนวแกน z ดังรูป จากที่เป็นไดโพลสมมาตรๆดังนั้นเราจึงสมมติว่า

$$x' = y' = 0 \text{ ซึ่งเราจะได้ว่า}$$

$$R = \sqrt{x'^2 + y'^2 + (z - z')^2}$$

ทำการกระจายสมการจะได้

$$\begin{aligned} R &= \sqrt{(x'^2 + y'^2 + z^2) + (-2zz' + z'^2)} \\ &= \sqrt{(2rz' \cos \theta + z'^2)} \end{aligned}$$

$$\text{ซึ่ง } r^2 = x'^2 + y'^2 + z^2$$

$$z = r \cos \theta$$

จากนั้นจะกระจายสมการโดยใช้ การกระจายแบบไบนอมิเยล (binomial expansion) จะได้ว่า

$$R = r - z' \cos \theta + (1/r) (z'^2/2) (\sin^2 \theta) + (1/r^2) (z'^3/2) (\cos \theta) (\sin^2 \theta) + \dots$$

โดยเทอมที่สูงขึ้นไปนั้นจะมีค่าน้อยลงเรื่อยๆเนื่องจาก $r \gg z'$

การพิจารณาขอบเขตของ Far-Field

ในสนามระยะไกลนั้นการประมาณค่า R เราจะได้ว่า

$$R \approx r - z' \cos \theta$$

ค่า R จะประมาณเพียง 2 เทอมเท่านั้น เนื่องจาก $r \gg z'$ โดยเทอมที่ตัดออกและสำคัญที่สุด จะเป็นเทอมที่ 3 โดยมีค่าสูงสุดเป็น

$$\left(\frac{1}{r} \right) \left(\frac{z'^2}{2} \right) (\sin^2 \theta)_{\max} = \frac{z'^2}{2r} \text{ เมื่อ } \theta = \pi/2$$

ดังนั้นเทอมที่ 4 และเทอมอื่นๆที่สูงกว่าจะหายไปด้วยเมื่อ $\theta = \pi/2$ นั่นคือที่ $\theta = \pi/2$ จะเกิดค่าผิดพลาดสูงสุด (maximum error)

ในส่วนของค่าเฟสผิดพลาดสูงสุด (maximum total phase error) นั้นมาตรฐานกำหนดให้ว่าจะต้องเท่ากับหรือน้อยกว่า $\pi/8$ (22.5) ดังนั้น

$$k(z')^2/2r \leq \pi/8$$

$$\text{โดยที่ } -l/2 \leq z' \leq l/2$$

$$\text{จะได้ว่า } r > 2l^2/\lambda$$

การประมาณค่าใน Far-Field

$R \approx r - z' \cos \theta$ สำหรับเฟส

แล้ว $R \approx r$ สำหรับขนาด

เมื่อ θ เป็นมุมที่ทำกับแกน z ในรูปที่ เวกเตอร์ R และ r จะขนานกันไปและถ้า

D เป็นขนาดที่ใหญ่ที่สุดของเสาอากาศก็จะได้จุดสังเกตในระยะ Far-Field

เป็น $r \geq 2D^2/\lambda$

สนามระยะใกล้ (Near-Field Region)

ถ้าเราใช้จุดสังเกตที่มีระยะห่างจากจุดกำเนิด(Source)น้อยๆ โดยให้ความผิดพลาดทางเฟส (phase error) เท่ากับ $\pi/8$ เรเดียน หรือ 22.5 องศา เราสามารถเขียนสมการได้ว่า

$$R \approx r - z' \cos \theta + (1/r)(z'^2/2)(\sin^2 \theta)$$

เพื่อที่จะหาความผิดพลาดทางเฟสโดย ดิฟเฟอเรนเชียลค่า θ ของเทอมที่ 4 ซึ่งจะต้องได้ผลลัพธ์เท่ากับศูนย์ ดังนั้น

$$\partial \left((1/r)(z'^2/2)(\cos \theta \sin^2 \theta) \right) / \partial \theta = 0$$

สำหรับค่า $\theta = 0$ จะให้ค่าความผิดพลาดน้อยที่สุด ความผิดพลาดมากที่สุดจะเท่ากับ

$$(-\sin^2 \theta + 2\cos^2 \theta)_{\theta=0} = 0$$

หรือ $\theta = \tan^{-1}(\pm\sqrt{2})$

ถ้าค่าความผิดพลาดทางเฟสสูงสุดเท่ากับหรือน้อยกว่า $\pi/8$ ระยะ หาได้จาก

$$\pi/(12\sqrt{3})(d^3/\lambda r^2) \leq \pi/8$$

สามารถลดรูปได้ เป็น

$$r^2 > (2/3\sqrt{3})(d^3/\lambda)$$

หรือ $r > 0.62 \sqrt{d^3/\lambda}$

ดังนั้นค่า r ที่มีค่าผิดพลาดทางเฟสสูงสุดไม่เกิน $\pi/8$ จะมีค่าเท่ากับ

$$2d^2/\lambda > r > 0.62\sqrt{d^3/\lambda}$$

โดย 1 เป็นความยาวของเสาอากาศ เรียกระยะ ช่วงนี้ว่า การกระจายสนามระยะใกล้ (radiating near-field)

สำหรับค่าที่อยู่ในช่วง $0.62\sqrt{d^3/\lambda} > r > 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่าที่อยู่ในช่วง $0.62\sqrt{D^3/\lambda} > r > 0$

เราเรียกว่า ช่วงปฏิกิริยาของสนามระยะใกล้ (reactive near-field) โดยเราสามารถสรุประยะทั้ง 3 ได้ดังนี้

reactive near field $0.62\sqrt{D^3/\lambda} > r > 0$

radiating near field $2D^2/\lambda > r > 0.62\sqrt{D^3/\lambda}$

far field $\infty > r > 2D^2/\lambda$

โดย D เป็นค่าที่ความยาวที่มากที่สุดของสายอากาศ

ทฤษฎีการวัดสนามระยะใกล้แบบระนาบ

ในการหารูปแบบการแพร่กระจายคลื่นของสายอากาศ โดยใช้เทคนิคการวัดสนามระยะใกล้แบบระนาบนี้ จำเป็นต้องทราบพารามิเตอร์ต่างๆ ที่สำคัญในการวัดดังต่อไปนี้

- รูปร่างของอะเพอร์เจอร์ที่ใช้ในการสแกน
- ขนาดของการสแกน
- คุณสมบัติของโพรับที่ใช้วัด
- รูปแบบการสแกน

- ระยะห่างที่เหมาะสมกับสายอากาศทดสอบ
- ระยะห่างในการสแกนบนระนาบที่เหมาะสม

นอกจากพารามิเตอร์เหล่านี้แล้ว เราจำเป็นที่จะต้องกำหนดเงื่อนไขต่างๆ ของการทดสอบให้เหมาะสมกับอะเพอร์เจอร์ที่จะทำการวัดด้วย ดังต่อไปนี้

- ขนาดของอะเพอร์เจอร์แบบระนาบที่จะทำการวัด จะต้องมีความใหญ่เพียงพอที่จะสามารถรองรับพลังงานทั้งหมดที่แพร่ออกมาจากสายอากาศทดสอบได้

- โพรับที่ใช้วัดจะต้องมีความเหมาะสมโดยจะทำการหา โพรับ แพทเทิร์น (probe pattern) เพื่อนำมาทำการชดเชยคุณสมบัติของโพรับกับรูปแบบการแพร่กระจายคลื่น ที่ได้จากการคำนวณ ซึ่งเรียกวิธีการนี้ว่า โพรับ คอลเลคชัน (probe correction)

สำหรับรายละเอียดของพารามิเตอร์ต่างๆ ที่ได้กล่าวไว้ในตอนต้นนั้น สามารถแบ่งออกเป็นหัวข้อต่างๆ ได้ดังนี้

รูปร่างของอะเพอร์เจอร์ที่ใช้ในการสแกน

ในบทความฉบับนี้จะใช้อะเพอร์เจอร์แบบระนาบ ในการทดสอบการวัดสนาม ระยะใกล้โดยใช้โพรับเคลื่อนที่ติดไปกับเครื่องมือที่เรียกว่า “สแกนเนอร์” (scanner) ซึ่ง จะทำการสแกนไปตามแนวเส้นทางที่ได้กำหนดขึ้นตลอด อะเพอร์เจอร์บนสายอากาศ ทดสอบที่จะทำการวัดในขณะเดียวกันก็จะทำการวัดข้อมูล คือ ความแรงของสัญญาณ (แอมพลิจูด) และตำแหน่งของเฟสไปพร้อมกับการสแกนในแต่ละตำแหน่งที่เคลื่อนที่ไป บนอะเพอร์เจอร์ที่ทำการทดสอบ

โดยปกติทั่วไปนั้น พื้นผิวของการสแกนแบบระนาบ จะนิยมใช้ทดสอบกับสาย อากาศที่มีอัตราขยายสูงเนื่องจากพลังงานของคลื่นระนาบทั้งหมดจะถูกส่งออกมาด้วย มุมเล็กๆ ซึ่งเป็นการง่ายในการปรับแต่งให้ถูกต้องในการทดสอบแต่ละครั้ง นอกจากนี้ แล้วการใช้การสแกนแบบระนาบ ยังมีข้อดีในด้านของการชดเชยโพรับได้ง่ายจากเหตุผล ดังกล่าวจึงเป็นที่นิยมใช้วิธีนี้วัดกันมาก ซึ่งเป็นเหตุผลที่ได้เลือกใช้วิธีนี้มาทำการทดสอบ ในงานวิจัยฉบับนี้

ขนาดของการสแกน

ในการสแกนโพรับไปบนอะเพอร์เจอร์แบบระนาบนั้น พื้นที่ของการสแกนจะ ต้องมีขนาดใหญ่เพียงพอที่จะรองรับพลังงานทั้งหมดของสนามระยะใกล้ที่แพร่ออกมาจาก สายอากาศทดสอบ ขนาดของการสแกนที่ถูกต้อง และเหมาะสมสำหรับการวัดสนาม ระยะใกล้แบบระนาบ สามารถหาได้จากกฎเกณฑ์ 3 ข้อ ดังนี้

1.) สำหรับบริเวณสนามระยะใกล้แบบระนาบ ขนาดของการสแกนจะขึ้นอยู่กับ ขนาดรูปร่างทางกายภาพของสายอากาศทดสอบเป็นหลัก โดยสามารถหาได้จากสูตรซึ่ง ไม่คิดค่าขนาดของโพรับที่นำมาใช้ดังนี้

$$\text{ขนาดการสแกนบนอะเพอร์เจอร์แบบระนาบ} = D + 2 Z \tan \theta$$

เมื่อ D = ขนาดใหญ่ที่สุดของสายอากาศทดสอบ

Z = ระยะห่างระหว่างสายอากาศทดสอบกับโพรับ

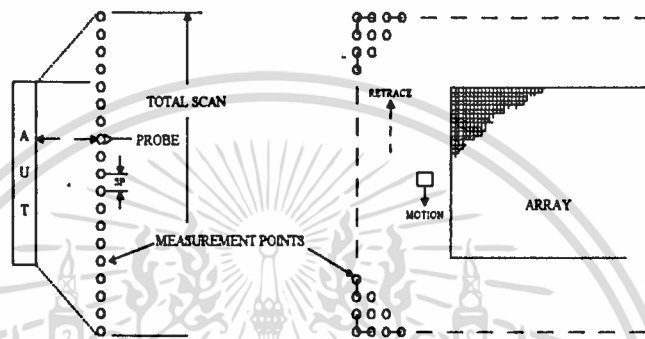
θ = มุมของการขยายขนาดอะเพอร์เจอร์ที่ใช้วัด

โดยค่ามุม θ ที่ใช้จะมีการใช้อยู่ 2 ค่า คือแบบธรรมดา 60 องศาและประสิทธิภาพ

สูง 75 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.) ไซค์โลปของโพรบไม่ควรที่จะรับสัญญาณของสายอากาศ
 - 3.) ขนาดของการสแกนบนอะเพอร์เจอร์จะต้องไม่ใหญ่เกินกว่า ที่ได้กำหนดไว้ใน กฎข้อ 1 เนื่องจากจะทำให้การคำนวณข้อมูลใช้เวลานาน และข้อมูลที่ได้อาจจะมีคุณภาพที่ต่ำลง อันเนื่องมาจากเกิดสัญญาณรบกวนกับข้อมูลเหล่านั้น
- จากกฎเกณฑ์ทั้ง 3 ข้อ ที่กล่าวมาแล้วนั้น สามารถแสดงให้เห็นได้ชัดเจนยิ่งขึ้น ดังรูปที่ 3



รูปที่ 3 บริเวณทดสอบสนามระยะใกล้แบบระนาบ

คุณสมบัติของโพรบ

ในการเลือกโพรบเพื่อนำมาใช้วัดสนามระยะใกล้จะต้องคำนึงถึงคุณสมบัติของโพรบให้เหมาะสมกับระบบที่ทำการทดสอบด้วยโดยพิจารณาจากคุณสมบัติ ดังต่อไปนี้

- มีขนาดเล็กเพื่อลดสัญญาณรบกวนต่างๆ
- ไม่มีจุดบอด (null) ในทุกทิศทางของบริเวณที่ทำการวัด
- สามารถเก็บข้อมูลของ การแพร่กระจายคลื่นของสายอากาศ ได้ตลอดอะเพอร์เจอร์ที่ทำการวัด

โพรบที่ทำการวัด

- มีการแพร่กระจายคลื่นของตัวโพรบไปทางด้านหลังน้อย
- มีลักษณะของการโพลาไรเซชันที่เหมาะสม
- สามารถใช้งานในย่านความถี่ที่กว้างได้

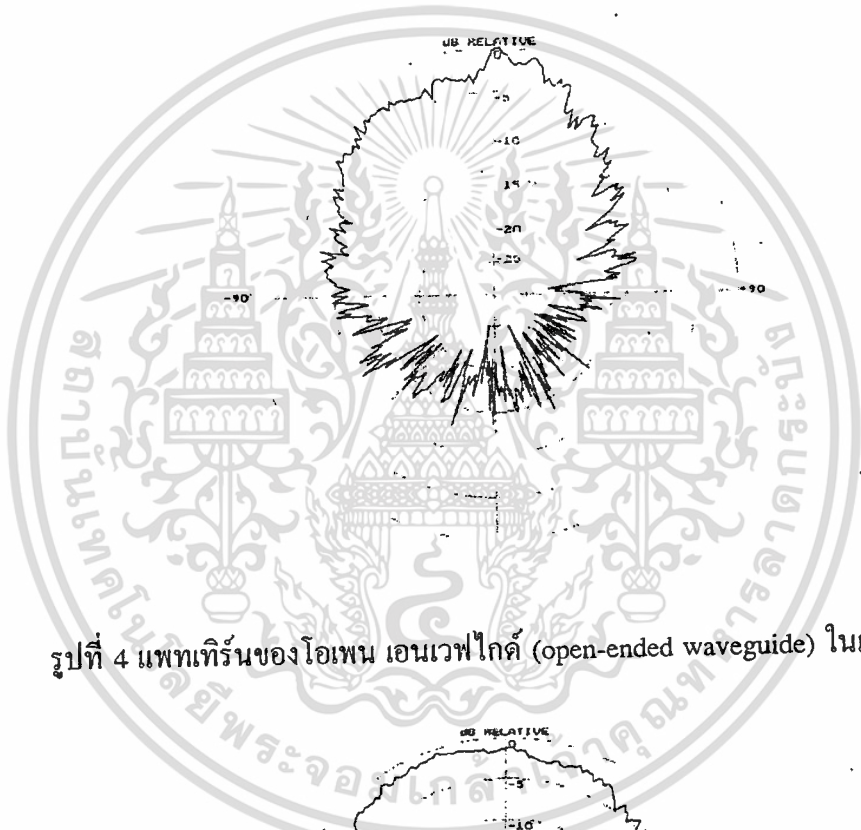
จากคุณสมบัติดังกล่าวนี้จึงได้เลือกเอา โอเพน เอนเวฟไกด์ (open-ended waveguide) มาใช้เป็นโพรบ เนื่องจากมีคุณสมบัติใกล้เคียงกับข้อกำหนดข้างต้น ซึ่งโอเพนเอน เวฟไกด์ นี้เหมาะสำหรับใช้กับการวัดแบบทุกทิศทาง (omnidirectional measurement) ของสนามไฟฟ้า E ซึ่งจะมีสัญญาณมารบกวนสนามแม่เหล็กไฟฟ้า น้อยมาก จากคุณสมบัติดังกล่าวจึงเป็นที่นิยมใช้กันมาก นอกจากจะพิจารณาคุณสมบัติของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

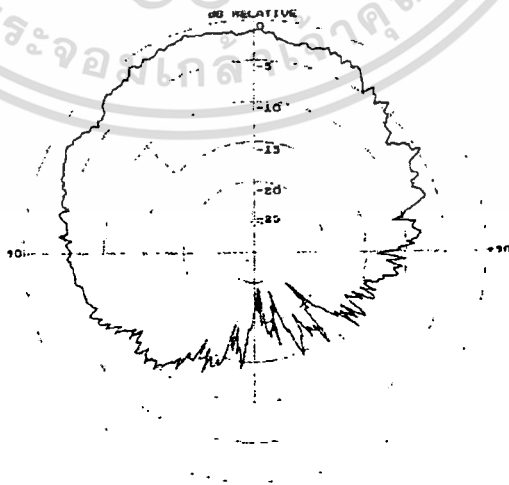
โพรับแล้ว ยังต้องทำการชดเชยโพรับหรือวิธีการที่เรียกว่า โพรับ คอลเลคชั่น (probe correction) อีกด้วย

การชดเชยโพรับ (probe correction)

เนื่องจากโพรับในอุดมคติที่ใช้ในการวัดค่าสนามไฟฟ้าและสนามแม่เหล็กไม่มีจริง ในทางปฏิบัติและโพรับในทางปฏิบัติที่เราใช้มีแพทเทิร์นซึ่งเป็นข้อจำกัดในการวัดค่าสนามไฟฟ้าและสนามแม่เหล็กจากสายอากาศที่เราทำการทดสอบยกตัวอย่างเช่น โอเพน เอนเวฟไกด์ (open-ended waveguide) ซึ่งมีแพทเทิร์นทางด้านแนวตั้งและแนวนอน ดังรูปที่ 4 และรูปที่ 5



รูปที่ 4 แพทเทิร์นของโอเพน เอนเวฟไกด์ (open-ended waveguide) ในแนวตั้ง



รูปที่ 5 แพทเทิร์นของโอเพน เอนเวฟไกด์ (open-ended waveguide) ในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สำหรับการสแกนแบบระนาบ (planar scanning) ในกรณีที่จะทำให้ต้องใช้การชดเชยโพรับได้แก่ การทดสอบสายอากาศที่เลียบมีหลักของโพรับออกไปและการสแกนแบบระนาบนี้ทิศทางของโพรับที่ชี้ออกจะเป็นทิศทางเดียวกันตลอดและอาจเกิดการวัดค่าจากไซด์โลบ (sidelobe) ของโพรับ

ดังนั้นใน การสแกน แบบระนาบ (planar scanning) นี้ เราจำเป็นต้องใช้โพรับคอลเลกชัน (probe correction) เพื่อความแม่นยำในการทดสอบสายอากาศแบบสนามระยะใกล้

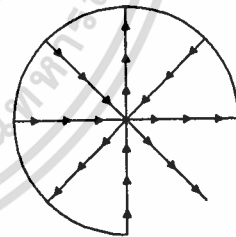
รูปแบบการสแกน

ในที่นี้หมายถึงทิศทางหรือแนวทางการเคลื่อนที่ไปบนอะเพอร์เจอร์แบบระนาบ รูปแบบการสแกนที่นิยมใช้กันมากคือ แบบราสเตอร์ (raster) เป็นการสแกนในระนาบ 2 มิติ แกน X แกน Y สามารถแสดงได้ดังรูปที่ 6(a) นอกจากนี้แบบราสเตอร์แล้ว ยังมีวิธีการสแกนอีกแบบหนึ่ง คือ แบบเพลน โพลาร์ (plane polar) หรือเรียกอีกอย่างหนึ่งว่า สตาร์เบิร์สท (starburst) มีลักษณะการสแกนคล้ายกับดาวกระจายแตกออกเป็นวงกลม แสดงได้ดังรูปที่ 6(b)

สำหรับงานวิจัยในบทความฉบับนี้ จะใช้การทดสอบแบบราสเตอร์ เนื่องจากเป็นการง่ายในการออกแบบสร้างอุปกรณ์ ที่ใช้ในการขับเคลื่อนโพรับ ให้สแกนเป็นแบบราสเตอร์ โดยจะเรียกเครื่องมือชนิดนี้ว่า สแกนเนอร์ (scanner) ซึ่งจะได้กล่าวในรายละเอียดอีกครั้งในหัวข้อของสแกนเนอร์ต่อไป



RASTER
(a)



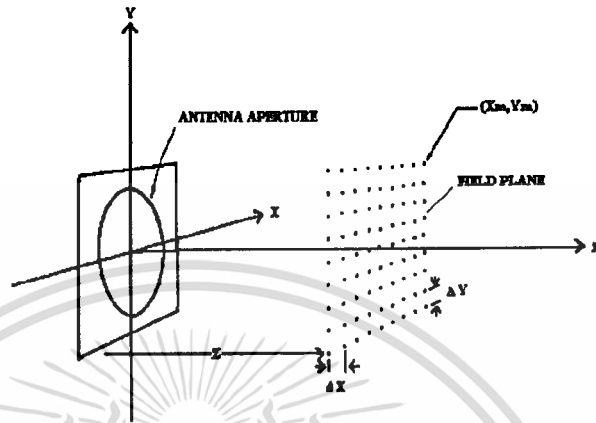
STARBURST
(b)

รูปที่ 6 รูปแบบการสแกนโพรับทั้งแบบราสเตอร์ และแบบสตาร์เบิร์สท

ระยะห่างของระนาบที่เหมาะสมกับสายอากาศทดสอบ

ในการวัดสนามระยะใกล้แบบระนาบ เราจำเป็นต้องกำหนดระยะนาบที่จะทำการสแกนและระยะห่างในการสแกนให้เหมาะสม เพื่อผลลัพธ์ที่ออกมาจะได้ค่าใกล้เคียงกับความจริงมากที่สุด สามารถหาได้โดยอาศัยหลักการ ดังนี้

กำหนดให้สายอากาศวางอยู่บนระนาบ $Z = 0$ โดยจะทำการสแกนหาสนาม ระยะใกล้บนระนาบ $Z = Z_1$ และระยะห่างระหว่างตำแหน่งของการแซมปลิง ในแนวแกน X และแนวแกน Y เป็น ΔX และ ΔY ตามลำดับ แสดงดังรูปที่ 7



รูปที่ 7 ระนาบของการวัดสนามระยะใกล้

จากการศึกษาเพื่อหาระยะ Z_1 ที่เหมาะสมได้มีการค้นคว้าและทดลองจนสรุปได้ว่าระยะ Z_1 ที่เหมาะสมควรมีค่ามากกว่า 1λ เนื่องจากในช่วงที่ $Z_1 < 1\lambda$ นั้น คลื่นที่อยู่ในย่านบริเวณสนามรีแอดทิฟจะมีผลต่อการวัด โดยสามารถวิเคราะห์ได้จากสมการ $|k| = k_0$ เมื่อ $k_x^2 + k_y^2 > k_0^2$ ซึ่งค่าคงที่ในการแพร่กระจายคลื่น k จะเป็นส่วนจินตภาพ เนื่องจาก $k_0^2 = k_x^2 + k_y^2 + k_z^2$ ดังนั้นคลื่นที่เกิดจากส่วนของ k_z จะลดลงแบบเอ็กซ์โพเนนเชียล (exponential) หรืออาจหายไปทิศทาง $Z_1 > 1\lambda$ หรือกล่าวอีกนัยหนึ่งได้ว่าคลื่นที่เกิดจากส่วนของ k_z จะมีผลในระยะ $Z_1 < 1\lambda$ เท่านั้น ซึ่งที่ระยะ $Z_1 > 1\lambda$ คลื่นส่วนนี้จะถูกกลทอนไปหมด ดังนั้นในการกำหนดระนาบที่จะทำการวัด ควรจะให้ระยะห่าง จากสายอากาศทดสอบเป็น $Z_1 > 1\lambda$ เสมอแต่จะต้องไม่เกินกว่า $2D^2/\lambda$ นอกจากนี้แล้วในการกำหนดระยะห่าง Z_1 จะต้องทำให้เกิดการคับปลิง (coupling) ระหว่างโพรับกับสายอากาศทดสอบให้น้อยที่สุดด้วย

ระยะห่างในการสแกนบนระนาบที่เหมาะสม

ในการกำหนดระยะห่างในการแซมปลิงข้อมูล ΔX และ ΔY สามารถหาได้จากความสัมพันธ์ของ เพลนเวฟ สเปกตรัม (plane-wave spectrum) ใน k สเปซ โดยที่ $k_x \in [-k_{xm}, k_{xm}]$ และ $k_y \in [-k_{ym}, k_{ym}]$ ดังนี้

$$\Delta X = 2\pi/T = 2\pi/2k_{xm} = \pi/k_{xm}$$

$$\Delta Y = 2\pi/T = 2\pi/2k_{ym} = \pi/k_{ym}$$

ในการกำหนด ระยะห่างของสเปกตรัม (spectral spacing) k_x และ k_y ในแกนโคออร์ดิเนต X, Y จะสัมพันธ์กับ อินเวอร์ส ดิสครีต ฟูเรียร์ ทรานฟอร์ม (inverse discrete fourier transform) ใน X, Y สเปซโดยที่ $X \in [-X_m, X_m]$ และ $Y \in [-Y_m, Y_m]$ ดังนี้

$$\Delta k_x = 2\pi/T = 2\pi/2X_m = \pi/X_m$$

$$\Delta k_y = 2\pi/T = 2\pi/2Y_m = \pi/Y_m$$

และเนื่องจาก $k_x = k_0 \sin \theta \cos \phi$ ดังนั้น $k_{xm} = k_0 = 2\pi/\lambda$

$k_y = k_0 \sin \theta \sin \phi$ ดังนั้น $k_{ym} = k_0 = 2\pi/\lambda$

เพราะฉะนั้น ระยะห่างในการแซมปลิงข้อมูลของสนามระยะใกล้แบบระนาบคือ

$$\Delta X = \Delta Y = \lambda/2$$

ทฤษฎีการคำนวณหาสนามระยะใกล้แบบระนาบและโพรับคอลเลกชัน

วิธีการที่จะใช้ในการหาสนามที่แพร่กระจายจากอะพอร์เจอร์ระนาบ คือจะใช้ฟูเรียร์ทรานซฟอร์ม (Fourier Transform) สิ่งสำคัญของวิธีนี้อยู่ที่การแสดงให้เห็นว่า แพทเทิร์นของสนามที่แพร่กระจายเป็นฟูเรียร์ทรานฟอร์มของสนามที่อยู่บนอะพอร์เจอร์ทำให้เราสามารถใช้คุณสมบัติต่างๆ ของคู่ฟูเรียร์ทรานฟอร์มในการทำนายคุณสมบัติของสายอากาศได้ ซึ่งในบริเวณฟรีสเปซ (free space) เราจะทำการวัดสนามระยะใกล้แบบระนาบ โดยนำข้อมูลที่ได้จากการวัดไปคำนวณหาสนามระยะใกล้ต่อไป ดังนั้นการหาสนามที่แพร่กระจายในบริเวณ $z > 0$ จะสามารถคิดได้ว่าสนามที่อะพอร์เจอร์เกิดจากต้นกำเนิดตัวหนึ่งในบริเวณ $z < 0$ โดยที่ไม่จำเป็นต้องรู้รายละเอียดของต้นกำเนิดนี้ เพราะสนาม E บนอะพอร์เจอร์จะเป็นตัวกำหนดสนามในครึ่งสเปซ $z > 0$ เท่านั้น

จากสมการของแมกซ์เวล (Maxwell's equation) ในกรณีที่สนามมีการเปลี่ยนแปลงสัมพันธ์กับเวลา เมื่อ $z > 0$ ทั้ง J และ ρ จะมีค่าเท่ากับศูนย์ทั้งคู่ ดังนั้นจะได้สมการคลื่นเวกเตอร์ดังนี้

$$\nabla^2 \vec{E} + k^2 \vec{E} = 0 \quad (1a)$$

$$\nabla \cdot \vec{E} = 0 \quad (1b)$$

จากกฎเกณฑ์พื้นฐานของฟูเรียร์ทรานฟอร์ม เมื่อมีฟังก์ชันของ $f(t)$ ใดๆ สามารถจัดรูปฟูเรียร์ทรานฟอร์มให้เป็นไปตามกฎของการเท่ากัน (symmetry) ทั้งรูปสมการฟูเรียร์ ทรานฟอร์ม และอินเวอร์สฟูเรียร์ ทรานฟอร์ม ได้ดังสมการ

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (2a)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อทำการอินเวอร์ส ฟูรีเยร์ ทรานฟอร์ม จะได้

$$F(\omega) = (1/2\pi) \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{2b}$$

สำหรับในบทความฉบับนี้ จะกำหนดตัวแปรใหม่ เพื่อให้เหมาะสมกับระบบสมการคลื่นแม่เหล็กไฟฟ้าโดยกำหนดให้ k_x แทน ω และ x แทน t ในทำนองเดียวกันถ้าเรามีฟังก์ชันของทั้ง x และ y ดังนั้นในการใช้ความสัมพันธ์ของฟูรีเยร์ ทรานฟอร์ม ผลของสนามไฟฟ้าที่ได้จะเป็นดังนี้

$$U(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x,y) e^{+jk_x x + jk_y y} dx dy \tag{3a}$$

ซึ่งค่าอินเวอร์สฟูรีเยร์ทรานซฟอร์มจะมีค่าเป็นดังนี้

$$u(x,y) = (1/4\pi^2) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(k_x, k_y) e^{-jk_x x - jk_y y} dk_x dk_y \dots(3b)$$

จากคุณสมบัติของฟูรีเยร์ ทรานฟอร์ม

$$\mathcal{F}_x \partial u(x,y) / \partial x = -jk_x \mathcal{F}_x u(x,y) \tag{4a}$$

$$\mathcal{F}_x \partial^2 u(x,y) / \partial x^2 = (-jk_x)^2 \mathcal{F}_x u(x,y) \tag{4b}$$

$$\mathcal{F}_{y,x} \partial^2 u(x,y) / \partial x^2 = -k_x^2 \mathcal{F}_{y,x} u(x,y) \tag{4c}$$

จากสมการ (1) สามารถจัดรูปได้ใหม่ ดังนี้

$$\left| \partial^2 / \partial x^2 + \partial^2 / \partial y^2 + \partial^2 / \partial z^2 + k^2 \right| \bar{E}(x,y,z) = 0 \tag{5a}$$

$$\partial \bar{E}_x(x,y,z) / \partial x + \partial \bar{E}_y(x,y,z) / \partial y + \partial \bar{E}_z(x,y,z) / \partial z = 0 \tag{5b}$$

ถ้าทำการฟูรีเยร์ ทรานฟอร์ม ในสมการ (5a) และ (5b) จากคุณสมบัติฟูรีเยร์ ทรานฟอร์ม ข้างต้น จะได้

$$\left| \partial^2 / \partial z^2 + (k^2 - k_x^2 - k_y^2) \right| \bar{E}(k_x, k_y, z) = 0 \tag{6a}$$

$$k_x \bar{E}_x(k_x, k_y, z) + k_y \bar{E}_y(k_x, k_y, z) + j \partial \bar{E}_z / \partial z (k_x, k_y, z) = 0 \tag{6b}$$

เมื่อ $\bar{E}(k_x, k_y, z)$ เป็นฟูรีเยร์ทรานฟอร์มของสนามไฟฟ้าที่เทียบกับ x และ y

ถ้ากำหนดให้

$$k_z^2 = k^2 - k_x^2 - k_y^2 \tag{7}$$

ดังนั้นสมการ (6a) จะเป็น

$$\partial^2 \bar{E}(k_x, k_y, z) / \partial z^2 + k_z^2 \bar{E}(k_x, k_y, z) = 0 \tag{8}$$

ซึ่งผลลัพธ์ของสมการจะอยู่ในรูปของ $e^{+jk_z z}$, $e^{-jk_z z}$ เนื่องจากสนามประกอบไปด้วยคลื่นที่แพร่กระจายออกจากอะพอร์เจอร์(aperture) ไปในทิศทาง Z เพราะฉะนั้นจึงมีเฉพาะฟังก์ชัน $e^{-jk_z z}$ เท่านั้นที่เป็นไปได้ ดังนั้นผลลัพธ์ของสมการ (8) จะเป็นดังนี้

$$\bar{E}(k_x, k_y, z) = \bar{A}(k_x, k_y) e^{-jk_z z} \tag{9}$$

เมื่อแทนสมการ (9) ลงในสมการ (6b) จะได้

$$k_x \bar{A}_x + k_y \bar{A}_y + k_z \bar{A}_z = 0 \tag{10a}$$

$$\text{หรือ } \vec{k} \cdot \vec{A} = 0 \quad (10b)$$

เมื่อ k เป็นเวกเตอร์ที่มีส่วนประกอบเป็น k_x, k_y, k_z สมการที่ (10a) และ (10b) บอกให้เราทราบว่าเฉพาะสองส่วนประกอบของเวกเตอร์ A เท่านั้นที่เป็นอิสระต่อกัน ซึ่งผลอันนี้จะส่งผลบังคับให้สนามไฟฟ้าถูกกำหนดโดยการที่ไดเวอร์เจนซ์ที่เป็นศูนย์ควแสดงในสมการที่ (6b) และ สมการที่ (1b)

เมื่อทำการอินทิเกรตฟูเรียร์ทรานฟอร์ม จะเขียนสนามไฟฟ้าได้ดังนี้

$$\vec{E}(x, y, z) = (1/4\pi^2) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \vec{A}(k_x, k_y) e^{-j\vec{k} \cdot \vec{r}} dk_x dk_y \quad \dots(11)$$

$$\text{โดยที่ } \vec{k} \cdot \vec{r} = k_x x + k_y y + k_z z$$

$$k_x = k \sin \theta \cos \phi$$

$$k_y = k \sin \theta \sin \phi$$

$$k_z = k \cos \theta$$

$$\text{ซึ่ง } k = 2\pi/\lambda$$

เมื่อ $z = 0$ ค่าส่วนประกอบ x และ y ของสนามไฟฟ้าต้องมีค่าเท่ากับสนามไฟฟ้าที่สัมผัสกับอะเพอร์เจอร์ซึ่งเรารู้ค่า ดังนั้นถ้าให้ A_t แทนส่วนประกอบ x และ y ของ A

$$E_{tan}(x, y, 0) = (1/4\pi^2) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A_t(k_x, k_y) e^{-j(k_x x + k_y y)} dk_x dk_y \quad (12)$$

จากสมการ (12) ทำฟูเรียร์ ทรานฟอร์ม เพื่อหา $A_t(k_x, k_y)$ ได้ดังสมการ

$$A_t(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E_{tan}(x, y, 0) e^{j(k_x x + k_y y)} dx dy \quad (13)$$

ถึงตอนนี้เราได้คำตอบของสนามไฟฟ้าที่ทุกๆ แห่ง ในบริเวณ $z > 0$ แล้วถ้าเราสามารถอินทิเกรตในสมการที่ 11 ได้โดยทั่วไปการคำนวณทำได้ยาก นอกจากจะคำนวณในบริเวณที่ r มีค่ามากเมื่อเปรียบเทียบกับ λ นั่นคือ kr มีค่ามาก เนื่องจากเราสนใจสนามที่แพร่กระจายออกไป ดังนั้นสิ่งที่เราต้องการทราบคือค่าประมาณของสมการที่ 11 เมื่อ r มีค่าเข้าสู่อินฟินิตี้ ซึ่งประมาณได้ว่ามีค่า

$$\vec{E}(r) = j(k \cos \theta / 2\pi r) e^{jkr} \vec{A}(k_x, k_y) e^{-j\vec{k} \cdot \vec{r}} \quad (14)$$

ผลจากการคำนวณสนามระยะไกลจากเพลนเวฟ สเปกตรัม $A(k_x, k_y)$ ที่ได้นั้นยังไม่ใช่ว่าผลที่ถูกต้องที่สุด เนื่องจากเรายังไม่ได้มีการนำเอาคุณสมบัติของโพรับ มาชดเชยด้วยเลข เพราะโพรับที่ใช้วัดก็ถือว่าเป็นสายอากาศอีกตัวหนึ่งเช่นกัน ดังนั้นโพรับแพทเทิร์นของตัวเองจึงมีผลต่อการวัดสิ่งที่กล่าวไว้ในการชดเชยโพรับ จากหัวข้อขั้นตอนการออกแบบสนามระยะไกลแบบระนาบ จะเห็นได้ว่ามีขั้นตอนของการชดเชยโพรับร่วมด้วย นั่นคือได้ทำการหาแพทเทิร์นของโพรับก่อน แล้วจึงนำมาทำการชดเชยโพรับ (หรือที่เรียกว่าโพรับคอลเลกชัน) กับเพลนเวฟ สเปกตรัมที่ได้จากการแปลงข้อมูลที่วัดได้

สมการการแก้ไขปรับค่าสนามจากโพรับ

ในการปรับค่าสนามที่คำนวณได้ จากแพทเทิร์นของโพรับนี้จะพิจารณาจากสนามที่วัดได้จากระบบการวัด ซึ่งจะมีทั้งสนามจากสายอากาศและจากโพรับซึ่งมีผลต่อสัญญาณที่วัดได้

ปริมาตร V เป็นปริมาตรที่ปราศจากแหล่งกำเนิดสนามใดๆ ซึ่งถูกล้อมรอบด้วยพื้นผิวปิด Σ ประกอบด้วย ระบาย S_0 ที่ $x=a$, พื้นผิวที่ระยะอนันต์ S_∞ และพื้นผิวปิด Σ_∞ ซึ่งอยู่รอบๆสายอากาศโพรับ B ถ้ากำหนดให้ E_a, H_a เป็นสนามของสายอากาศที่ทดสอบ A E_b, H_b เป็นสนามของโพรับสายอากาศ B E_{as}, H_{as} เป็นสนามที่แตกกระจายโดยสายอากาศ A E_{bs}, H_{bs} เป็นสนามที่แตกกระจายโดยสายอากาศโพรับเมื่อ A เป็นตัวส่ง และให้ความหนาแน่นกระแส J_a, J_b, J_{as}, J_{bs}

จากทฤษฎี ความเหมือนกันของลอเรนซ์ (Lorenz Reciprocity)

$$\oint_{\Sigma} [(E_a + E_{bs}) \times (H_b + H_{as}) - (E_b + E_{as}) \times (H_a + H_{bs})] \cdot \hat{n} da = 0$$

Σ เป็นพื้นผิวปิด

จะได้เป็น

$$\int_{V_B} [(E_a + E_{bs}) \times (H_b + H_{as}) - (E_b + E_{as}) \times (H_a + H_{bs})] \cdot (-\hat{n}) dv = - \int_{V_B} [(E_b + E_{as}) \cdot (J_a + J_b) - (E_a + E_{bs}) \cdot (J_b + J_{as})] dv$$

ซึ่ง $E_{bs} \cdot J_a$ และ $J_{as} \cdot E_b$ เป็นอัตราผลิตพลังงานในปริมาตร V_B ในการกักจะไม่เกิดค่า E_{as} เมื่อเทียบกับค่า E_b หมายถึงว่าสนามที่แพร่กระจายโดยโพรับไม่ถือว่าเป็นการแตกกระจาย (scatter) ดังนั้นฝั่งขวาของสมการแสดงได้เป็น

$$\int_{V_B} [E_b \cdot J_{bs} - E_a \cdot J_b] dv \equiv P_B(r_0)$$

ซึ่ง $P_B(r_0)$ เป็นสัดส่วนกับโวลเตจที่รับได้ที่โพรับ จากทฤษฎีเนทเวิร์ค โวลเตจที่โหลดจะเป็นสัดส่วนกับโวลเตจที่เปิดวงจรที่ความถี่คงที่ ดังนั้น $P_B(r_0)$ เป็นค่าสนามไฟฟ้าที่วัดได้ในสัดส่วนที่คงที่

จะเขียนได้เป็น

$$\int_{S_0} \{ (E_a + E_{bs}) \times (H_b + H_{as}) - (E_b + E_{as}) \times (H_a + H_{bs}) \} \cdot \hat{n} da = P_B(r_0)$$

ในเทอมโปรดัคส์ของสนามจากสายอากาศและสนามที่แตกกระจายจะหาค่าได้เป็น

$$\begin{aligned} & \int_{S_0} (E_a \times H_{as} - E_{as} \times H_a) \cdot \hat{n} da \\ &= \oint_{S_0 + S_\infty} (E_a \times H_{as} - E_{as} \times H_a) \cdot \hat{n} da \\ &= \int_{V + V_B} (E_{as} \cdot J_a - E_a \cdot J_{as}) dv \\ &\equiv 0 \end{aligned}$$

เทอมของสนามที่แตกกระจายมีค่าน้อยมาก เมื่อเปรียบเทียบกับสนามของสายอากาศ จึงตัดทิ้งได้

$$\int_{S_0} (E_a \times H_b - E_b \times H_a) \cdot \hat{n} da = P_B(r_0)$$

เขียนสนามต่างๆในรูปของ wavenumber สเปกตรัม บนระนาบ S_0

$$E_a(x, y, a) = \iint_{-\infty-\infty} F(k_x, k_y) \exp(-jk \cdot r) dk_x dk_y$$

$$H_a(x, y, a) = \frac{1}{\omega\mu} \iint_{-\infty-\infty} k \times F(k_x, k_y) \exp(-jk \cdot r) dk_x dk_y$$

$$E_b(x', y', b) = \iint_{-\infty-\infty} G(k'_x, k'_y) \exp(-jk' \cdot r') dk'_x dk'_y$$

$$H_b(x', y', b) = \frac{1}{\omega\mu} \iint_{-\infty-\infty} k' \times G(k'_x, k'_y) \exp(-jk' \cdot r') dk'_x dk'_y$$

เวกเตอร์ k' และ k มีทิศตรงข้ามกัน

$$\begin{aligned} k' &= k'_x \hat{X}' + k'_y \hat{Y}' + k'_z \hat{Z}' \\ &= k_x (-\hat{X}) + (-k_y) \hat{Y} + k_z (-\hat{Z}) \\ &= -k \end{aligned}$$

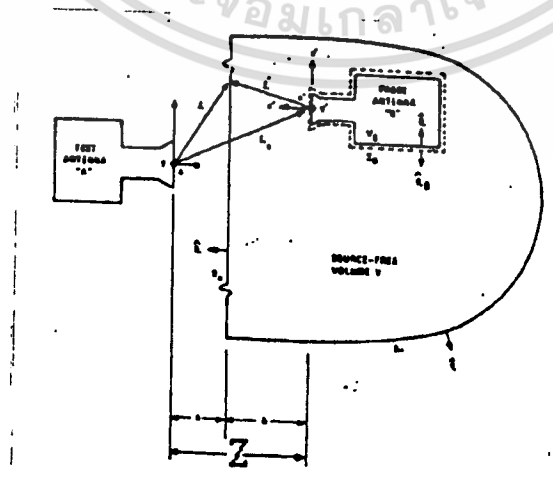
เมื่อนำสมการสนามระยะไกลมาเขียนรวมกัน และแทนพิกัด k' ด้วย k จะได้เป็น

$$\frac{4\pi^2}{\omega\mu} \iint_{-\infty-\infty} [F(k_x, k_y) \times \{k \times G(k_x, -k_y)\} + G(k_x, -k_y) \times \{k \times F(k_x, k_y)\}] \cdot \hat{X} \exp(-jk \cdot r_0) dk_x dk_y = P_B(r_0)$$

ดังนั้นผลลัพธ์ที่ได้จะเป็น

$$\frac{8\pi^2}{\omega\mu} \iint_{-\infty-\infty} k_z F(k_x, k_y) \cdot G(k_x, -k_y) \exp(-jk \cdot r_0) dk_x dk_y = P_B(r_0)$$

เมื่อทำการอินทิเกรตฟูรีเยร์ จะได้สนามระยะไกลของสนามในระนาบที่วัดได้ $P_B(r_0)$ ดังนั้น เมื่อย้ายข้างสมการและนำค่าสนามระยะไกลของโพรับไปหาร ก็จะได้สนามระยะไกลของสายอากาศ ที่ได้ทำการชดเชยค่าโพรับแล้ว



ขั้นตอนการออกแบบสนามระยะใกล้แบบระนาบ

การวัดสนามระยะใกล้แบบระนาบสำหรับงานวิจัยในบทความฉบับนี้มีขั้นตอนต่างๆขั้นตอนพื้นฐานได้แก่ ในระบบแกนโคออร์ดิเนต X-Y ในระนาบของสายอากาศ และแนวแกน Z จะตั้งฉากกับระนาบ X-Y สมมติให้ สายอากาศทดสอบ (antenna under test ; AUT) ทำการส่งสัญญาณ $E(x, y)$ ซึ่งถูกเข้มปั้งด้วยโพรับ สำหรับสนามที่วัดได้เหล่านี้จะนำไปคำนวณเพื่อแปลงหา เฟลนเวฟ สเปกตรัม $A(k_x, k_y)$ ซึ่งแพทเทิร์นเหล่านี้ จะถูกรบกวนด้วย คุณสมบัติของโพรับที่ใช้วัดจึงต้องมีการชดเชยคุณสมบัติของโพรับก่อน ดังนั้นสเปกตรัมที่เหมาะสมที่ได้จากการแปลงไปสู่แพทเทิร์นสนามระยะไกลจะเป็น $A'(k_x, k_y)$ หลังจากนั้นจึงนำ เฟลนเวฟ สเปกตรัม $A'(k_x, k_y)$ ที่ได้ใหม่นี้ไปคำนวณหาสนามระยะไกลต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

โปรแกรมที่ใช้ในการวัดแพทเทิร์นของสายอากาศแบบสนามระยะใกล้

โปรแกรมที่ใช้ในการวัดแพทเทิร์นของสายอากาศแบบสนามระยะใกล้จะประกอบด้วย

2 ส่วน ได้แก่

1. โปรแกรมควบคุมเครื่องวิเคราะห์โครงข่าย (network analyzer) และเก็บข้อมูล
2. โปรแกรมวิเคราะห์ทางคณิตศาสตร์และนำข้อมูลที่ได้ออกไปเขียนแพทเทิร์นของสายอากาศ

โปรแกรมควบคุมเครื่องวิเคราะห์โครงข่ายและเก็บข้อมูล

โปรแกรมในส่วนนี้จะทำการควบคุมเครื่องวิเคราะห์โครงข่ายให้สามารถทำการวัดค่าสำหรับการวัดแพทเทิร์นแบบสนามระยะใกล้ ทำการเก็บข้อมูลที่วัดได้ลงไฟล์ และจะทำการ นอร์มอลไรส์ค่าเฟสกับค่าอ้างอิง สุดท้ายจะนำค่าที่ได้มาทำการเก็บลงไฟล์อีกครั้งหนึ่งเพื่อรอการนำไปวิเคราะห์ทางคณิตศาสตร์ในส่วนต่อไป

การทำงานของโปรแกรมในส่วนนี้จะมีลักษณะการทำงานเป็นลำดับขั้นตอนดังรูปโฟลชาร์ตที่ 1 ซึ่งสามารถแบ่งเป็นข้อๆ ได้ดังนี้

1. การเลือกรูปแบบของการวัดในแต่ละช่องสัญญาณ (meas_setup) โปรแกรมจะถามหารูปแบบของแต่ละช่องสัญญาณว่าจะใช้แบบส่งผ่าน (transmission) หรือจะใช้แบบสะท้อนกลับ (reflection) หลังจากนั้นโปรแกรมจะถามถึงรูปแบบของการวัดของแต่ละช่องสัญญาณว่าจะใช้แบบใด ซึ่งรูปแบบของการวัดนี้เครื่องวิเคราะห์โครงข่ายสามารถใช้ได้หลายแบบ ได้แก่ แบบขนาดในหน่วยลอก (dBm) ,แบบเฟส ,แบบดีเลย์ ,แบบสมิตชาร์ต ,แบบโพลาร์ ,แบบขนาดเชิงเส้น ,แบบค่าจำนวนจริง และแบบค่า swr

สำหรับการวัดแพทเทิร์นของสายอากาศแบบสนามระยะใกล้นี้เราจะใช้การวัดแบบส่งผ่านจากพอร์ทที่ 1 ไปยังพอร์ทที่ 2 (s21) และใช้รูปแบบการวัดแบบขนาดเชิงเส้นสำหรับช่องสัญญาณหนึ่งและใช้รูปแบบการวัดแบบเฟสสำหรับช่องสัญญาณที่เหลือ

การทำงานของโปรแกรมในส่วนนี้จะเป็นดังรูปโฟลชาร์ตที่ 2

2. การเลือกช่วงของเพาเวอร์ในการส่ง (select_power) โดยโปรแกรมจะให้มีการเลือกช่วงของเพาเวอร์ที่จะใช้ ซึ่งเครื่องวิเคราะห์โครงข่ายนี้จะมีช่วงของเพาเวอร์ให้เลือก 12 ช่วงด้วยกัน ดังนี้ -10 ถึง +10 dBm , -15 ถึง +5 dBm , -20 ถึง 0 dBm , -25 ถึง -5 dBm , -30 ถึง -10 dBm , -35 ถึง -15 dBm , -40 ถึง -20 dBm , -45 ถึง -25 dBm , -50 ถึง -30 dBm , -55 ถึง -35 dBm , -60 ถึง -40 dBm และ -65 ถึง -45 dBm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการวัดแพทเทิร์นของสายอากาศแบบสนามระยะใกล้นี้เราจะใช้ช่วงเฟาเวอร์ที่ -10 ถึง +10 dBm

การทำงานของโปรแกรมในส่วนนี้จะเป็นดังรูปโฟลชาร์ตที่ 3

3. การเลือกช่วงของความถี่ที่ใช้สำหรับการวัด (select_freq) โปรแกรมจะให้เลือกรูทที่ค่าความถี่ที่ใช้งานได้ 2 แบบ ได้แก่

- การเลือกใส่ค่าความถี่เริ่มต้น และความถี่สิ้นสุด

- การเลือกใส่ค่าความถี่กลาง และสเปนของความถี่

การเลือกช่วงของความถี่ที่ใช้สำหรับการวัด จะขึ้นอยู่กับว่า เราจะใช้ความถี่ใดในการทดสอบสายอากาศ

การทำงานของโปรแกรมในส่วนนี้จะเป็นดังรูปโฟลชาร์ตที่ 4

4. การคาร์ริเบรต (calibration) จากในรูปโฟลชาร์ตที่ 1 โปรแกรมจะให้เลือกว่าจะทำการคาร์ริเบรตหรือไม่ เนื่องจากในกรณีที่เราได้ทำการคาร์ริเบรตไว้แล้ว เราก็คงไม่มีความจำเป็นที่จะต้องทำการคาร์ริเบรตซ้ำอีก การคาร์ริเบรตนี้เราจะใช้การคาร์ริเบรตแบบ 2 พอร์ต (full-2-port)

5. การใส่ค่าความถี่สำหรับมาร์คเกอร์ (marker_freq) การสังเกตค่าที่วัดได้จากความถี่ใด ความถี่หนึ่งเราสามารถทำได้โดย กำหนดค่าความถี่ของมาร์คเกอร์นั้น ๆ โปรแกรมจะให้มีการ์คเกอร์ได้สูงสุด 5 ตัวเท่ากับจำนวนมาร์คเกอร์ที่มีได้สูงสุดของเครื่องวิเคราะห์โครงข่าย ทำให้เราสามารถอ่านค่าได้ 5 ค่าในแต่ละความถี่ 5 ความถี่ต่อ 1 ช่องสัญญาณ

การทำงานของโปรแกรมในส่วนนี้จะเป็นดังรูปโฟลชาร์ตที่ 5

6. การเลื่อนโพรบไปยังจุดที่ต้องการและทำการอ่านค่าออกมาทางหน้าจอ (scanner,read) โปรแกรมจะทำการเลื่อนโพรบไปยังจุดแรก แล้วทำการอ่านค่าออกมา หลังจากนั้นโปรแกรมจะเลื่อนโพรบไปทางแกน x 1 ครั้ง (1 step) แล้วทำการอ่านค่าออกมาอีก จนกระทั่งครบทุกจุดในแกน x โปรแกรมจึงเลื่อนโพรบลงไปทางแกน y 1 ครั้ง (1 step) อ่านค่าที่วัดได้ออกมาแล้วโปรแกรมจึงเลื่อนไปทางแกน x อีก และในการสแกนแบบ ราสเตอร์ (raster) เมื่อสแกนจบ 1 แถวแล้วจะลงมาสแกนในแถวต่อไปโดยจะสแกนในทิศทางกลับกัน ทำซ้ำไปเรื่อย ๆ จนกว่าจะครบทุก ๆ จุดที่ต้องการวัดค่าออกมา จึงจะสิ้นสุด

การทำงานของโปรแกรมเลื่อนโพรบนี้จะเป็นดังรูปโฟลชาร์ตที่ 6 ส่วนการทำงานของโปรแกรมการอ่านค่าจะเป็นดังรูปโฟลชาร์ตที่ 7

7. การเก็บค่าที่อ่านออกมาได้ (save) ก่อนที่จะทำการนอร์มอลไรส์กับค่าอ้างอิง จากรูปโฟลชาร์ตที่ 1 เราจะเห็นได้ว่าในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟส โปรแกรมจะให้เราทำการเก็บค่าที่วัดออกมาได้แล้วก่อน โดยโปรแกรมจะให้เลือกว่าจะทำการเก็บทั้ง 2 ช่องสัญญาณในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟส หรือ จะเก็บเพียงช่องสัญญาณเดียวที่ใช้การวัดแบบเฟส ซึ่งจะนำไปทำการนอร์มอลไรส์กับค่าอ้างอิง

เพื่อเก็บไว้พิจารณา การเก็บค่าที่อ่านออกมาได้นี้เราสามารถทำการเก็บค่าได้ 5 ค่าในแต่ละความถี่ 5 ความถี่ต่อ 1 ช่องสัญญาณ

การทำงานของโปรแกรมการเก็บข้อมูลก่อนที่จะทำการนอร์มอลไรส์กับค่าอ้างอิงโดยเลือกที่จะทำการเก็บข้อมูลทั้ง 2 ช่องสัญญาณในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟสนี้จะเป็นดังรูปโพลชาร์ตที่ 8 ส่วนการเก็บเพียงช่องสัญญาณเดียวที่ใช้การวัดแบบเฟสจะเป็นดังรูปโพลชาร์ตที่ 9

8. การการนอร์มอลไรส์กับจุดอ้างอิงตรงกึ่งกลาง (abb) ซึ่งจะกล่าวโดยละเอียดในข้อต่อไป
การทำงานของโปรแกรมนี้จะเป็นดังรูปโพลชาร์ตที่ 10

9. การหาค่าจากจุดที่จะนำมาเป็นจุดอ้างอิง (medium) ในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟส โปรแกรมจะทำการหาจุดที่จะนำมาเป็นจุดอ้างอิงเพื่อการนอร์มอลไรส์กับค่าอ้างอิง โปรแกรมจะให้เลือกการหาค่าจากจุดที่จะนำมาเป็นจุดอ้างอิงที่จุดกึ่งกลางซึ่งในการใช้จุดกึ่งกลาง

ในการหาค่าจากจุดกึ่งกลาง เราจะใช้การทำงานของโปรแกรมหาดังรูปโพลชาร์ตที่ 11

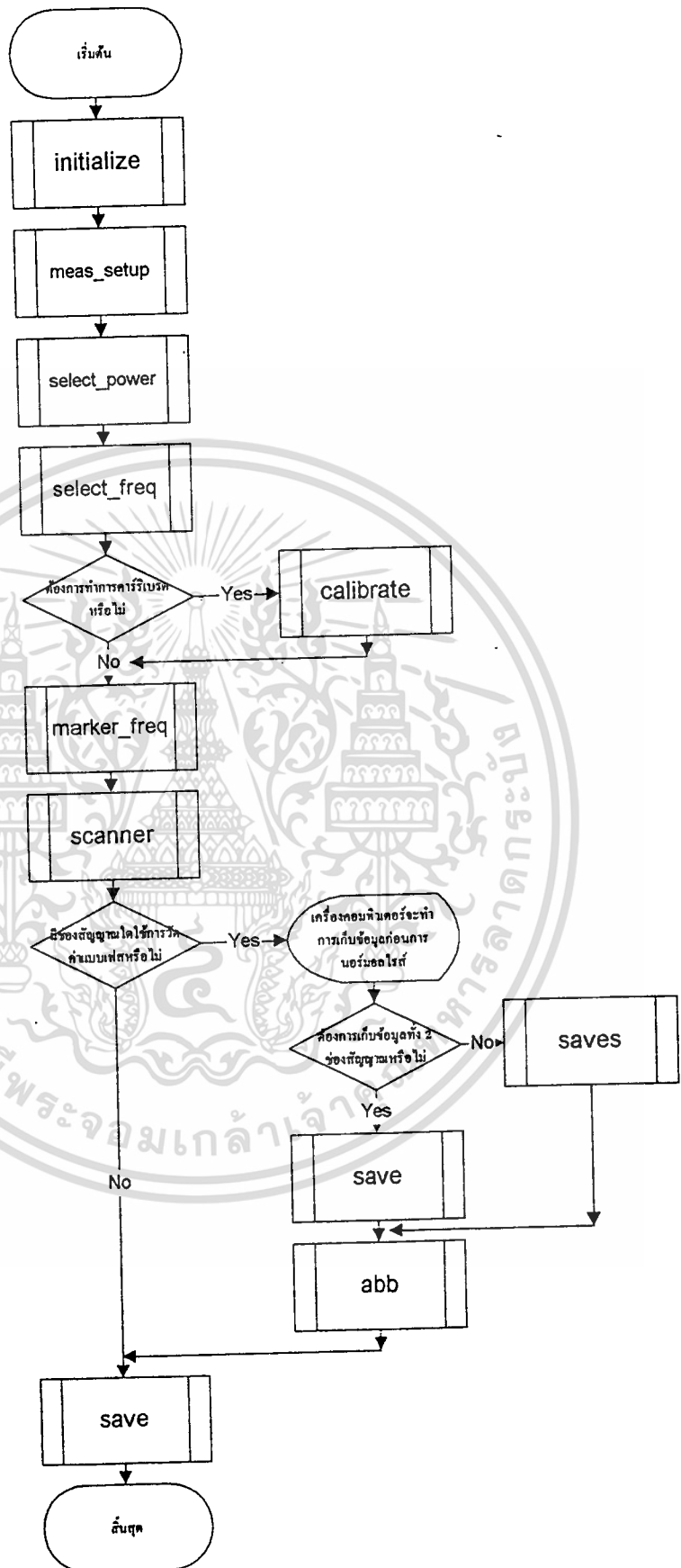
10. การนอร์มอลไรส์กับค่าอ้างอิง (normal_phas) ในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟส โปรแกรมจะทำการนอร์มอลไรส์กับค่าอ้างอิงที่หามาได้จากข้อที่แล้ว เพื่อนำไปใช้ในการวิเคราะห์ทางคณิตศาสตร์ต่อไป

การทำงานของโปรแกรมการนอร์มอลไรส์กับค่าอ้างอิงในกรณีที่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟส จะเป็นดังรูปโพลชาร์ตที่ 12

11. การเก็บค่าที่อ่านออกมาได้ (save) หลังการนอร์มอลไรส์กับค่าอ้างอิง หรือในกรณีที่ไม่มีการใช้ช่องสัญญาณใดช่องสัญญาณหนึ่งในการวัดค่าในรูปแบบของเฟสหรือค่าในรูปแบบขนาดเชิงเส้นดังรูปโพลชาร์ตที่ 1 เพื่อนำไปใช้ในการวิเคราะห์ทางคณิตศาสตร์ต่อไป

การทำงานของโปรแกรม การเก็บค่าที่อ่านออกมาได้ หลังการนอร์มอลไรส์กับค่าอ้างอิงแล้วหาร 2 และทำการถอดรากที่ 2 จะเป็นดังรูปโพลชาร์ตที่ 8

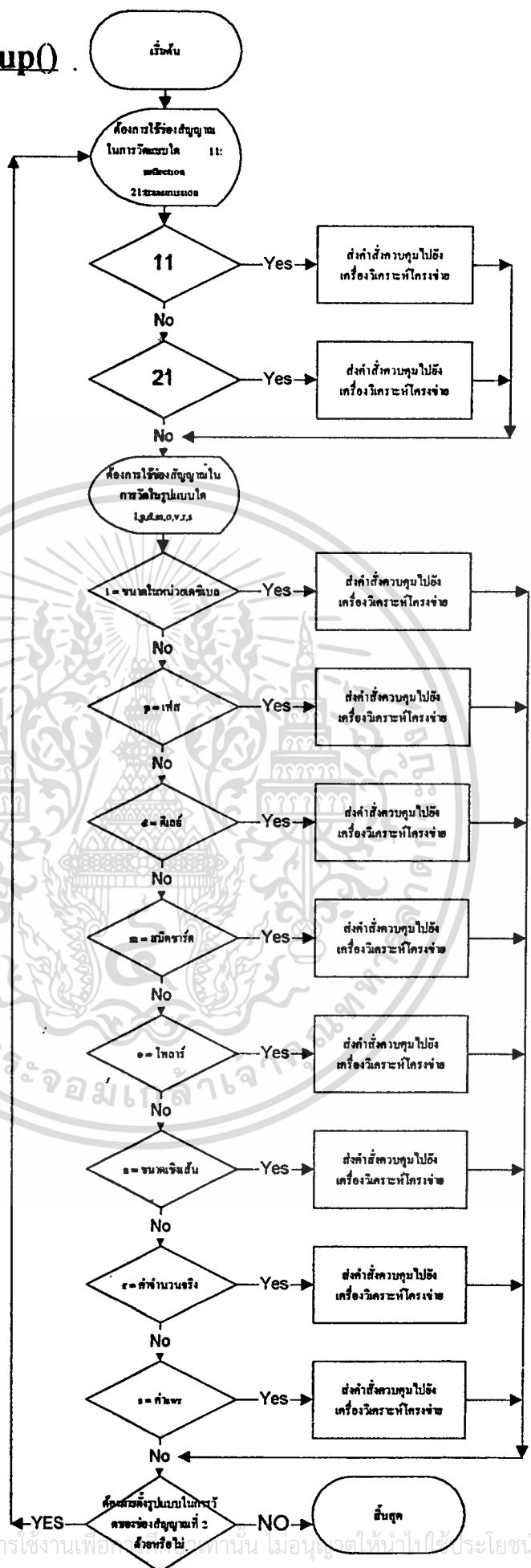
main()



รูปที่ 1

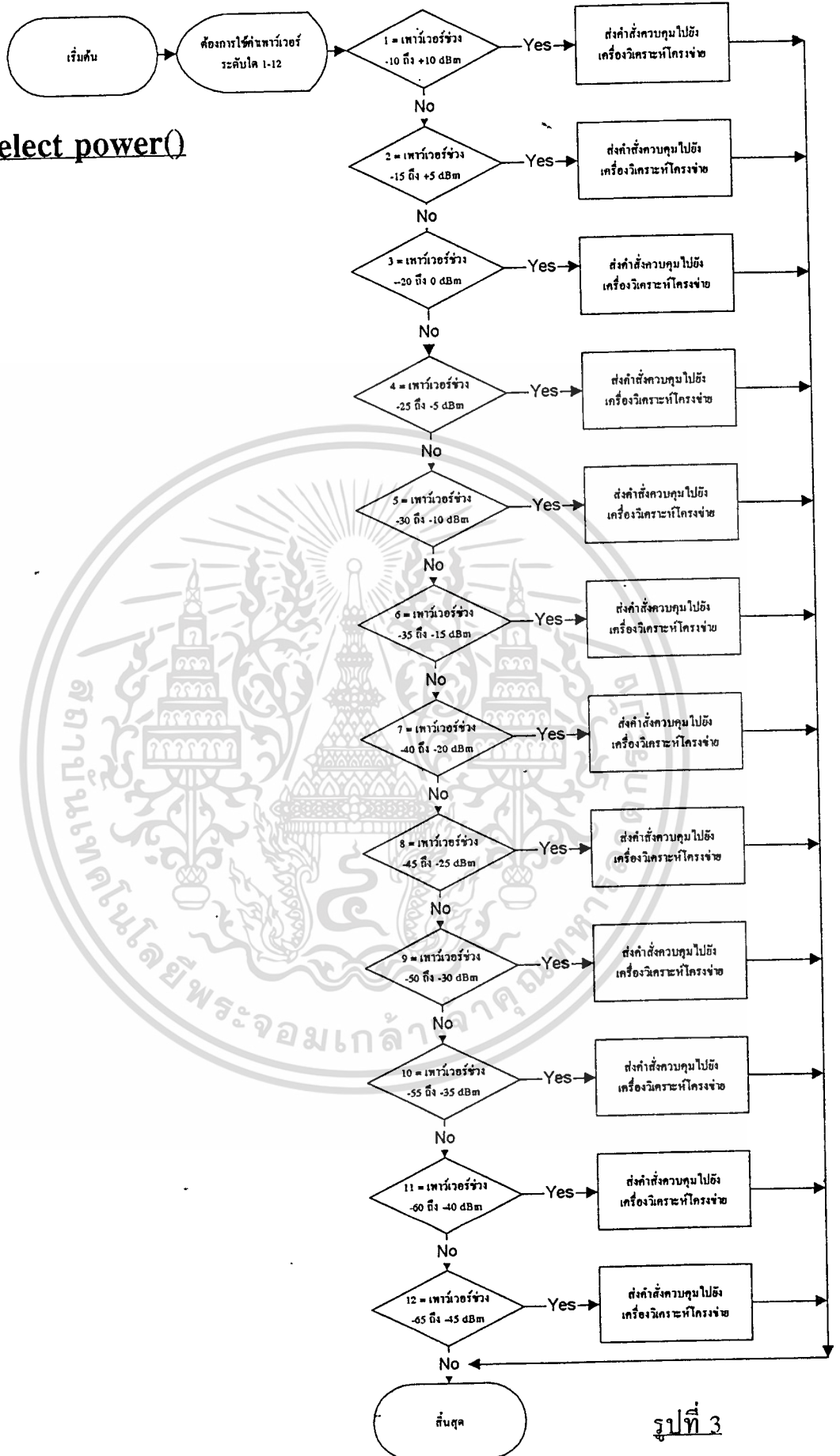
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

meas_setup()



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ... ไม่ควรแก้ไข...
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา...
 รูปที่ 2

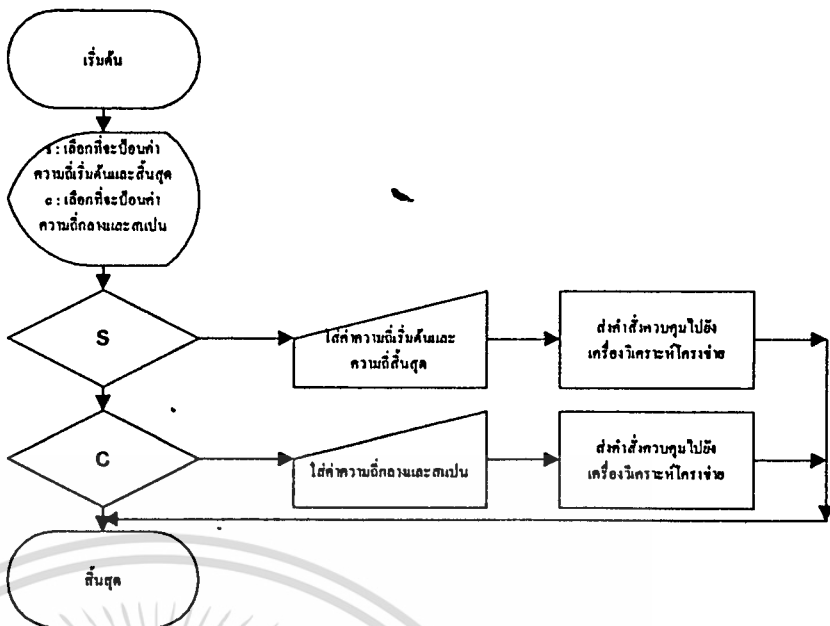
select power()



รูปที่ 3

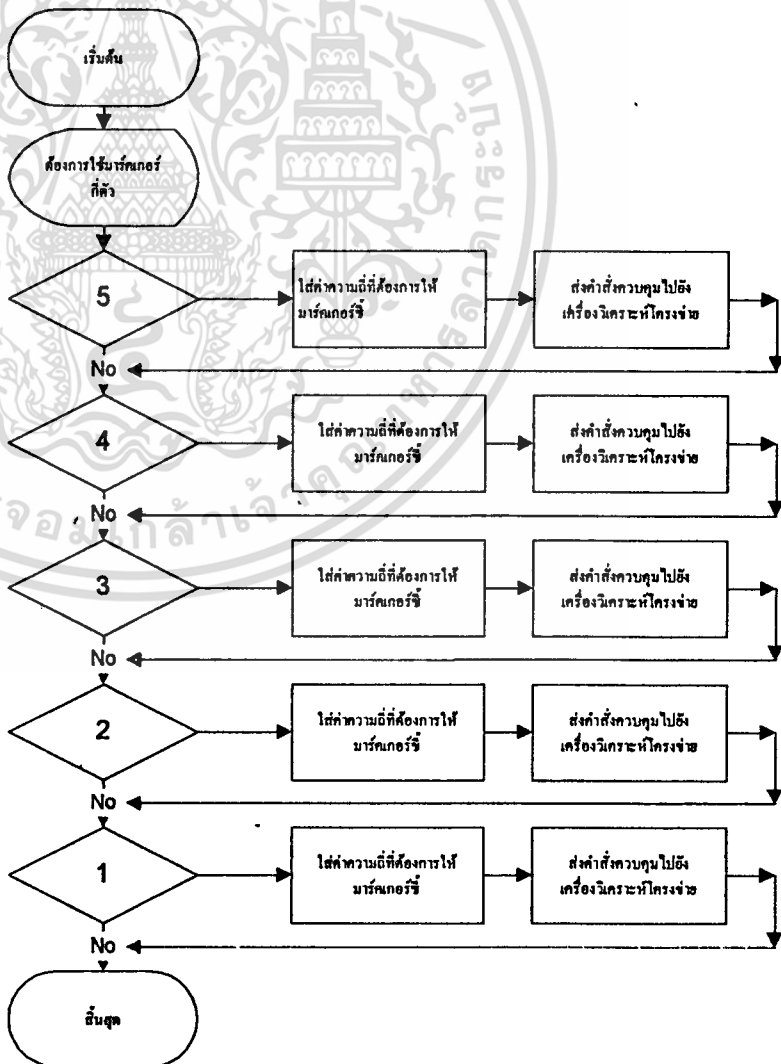
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

select_freq()



รูปที่ 4

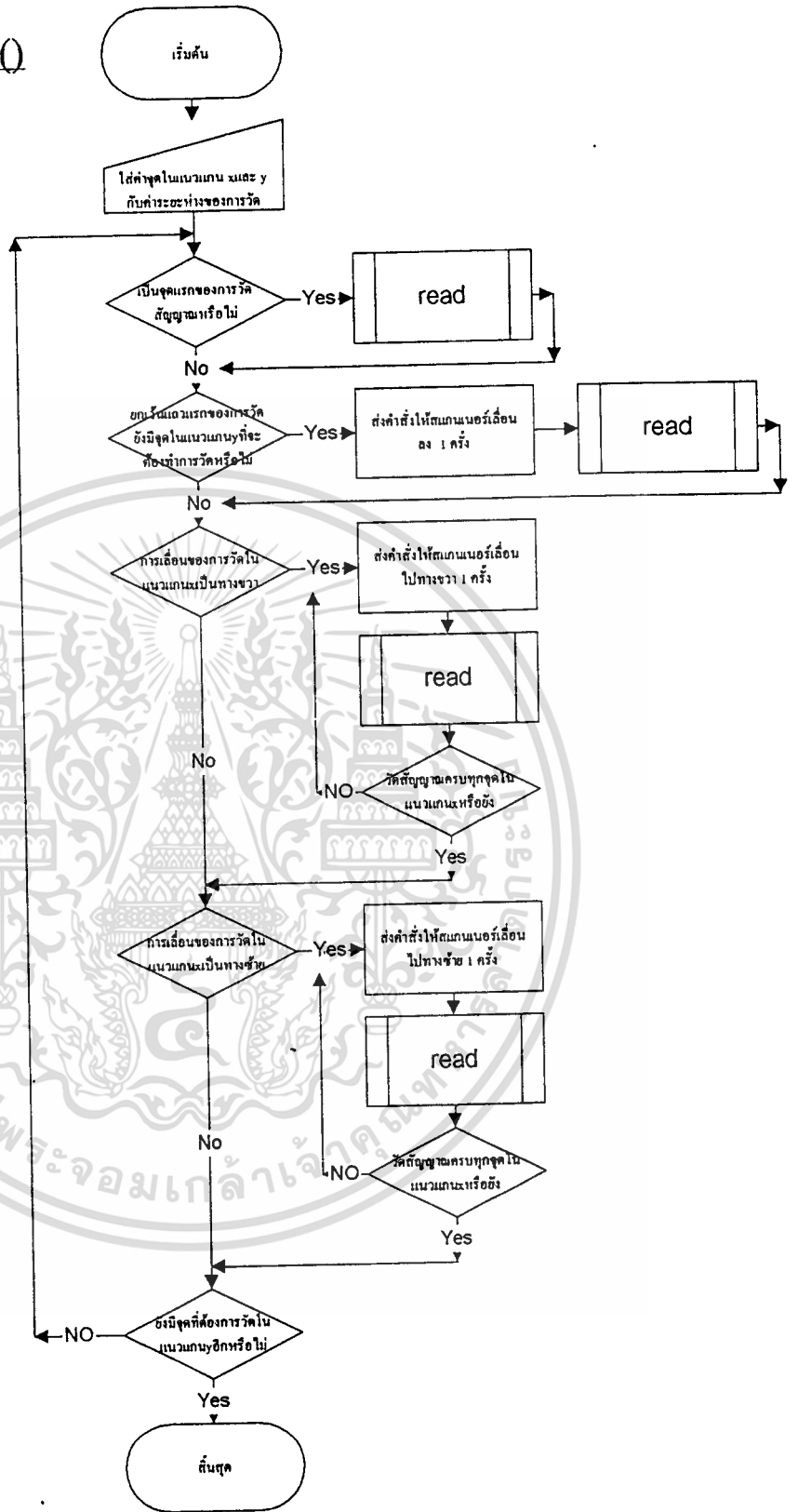
marker_freq()



รูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

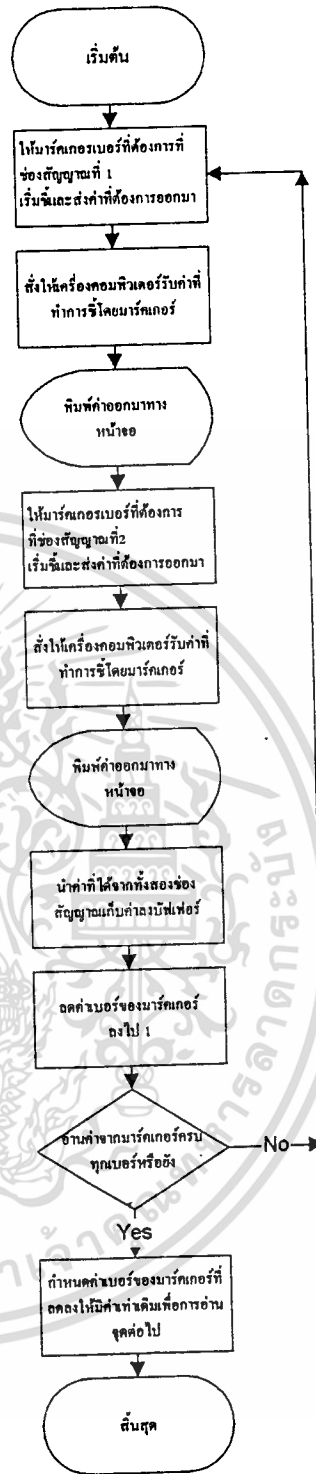
scanner()



รูปที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

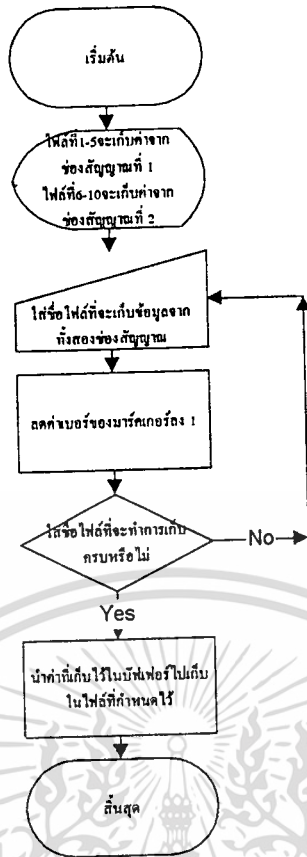
read()



รูปที่ 7

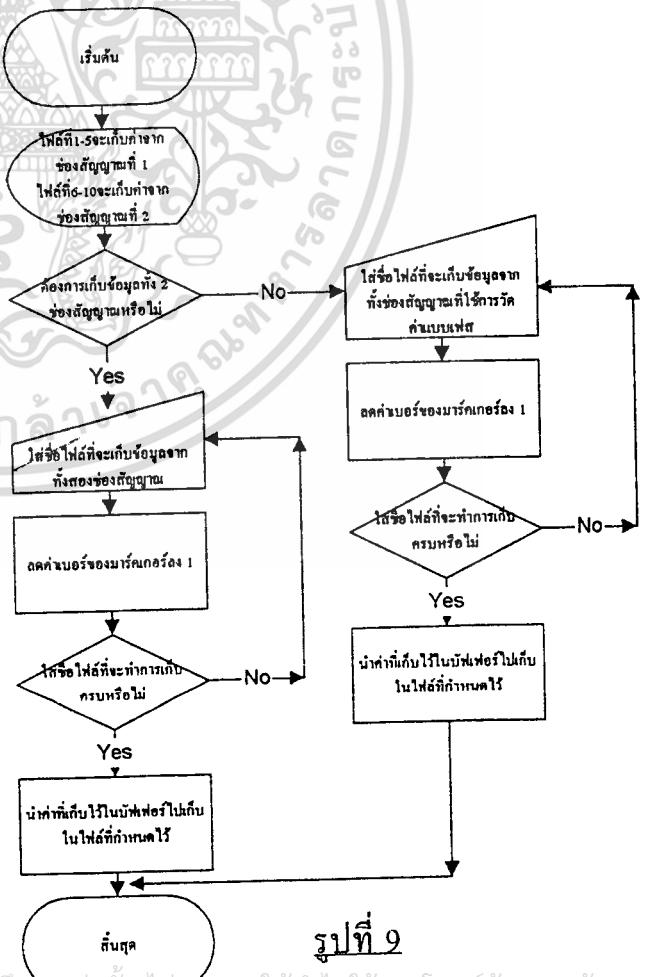
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

save()



รูปที่ 8

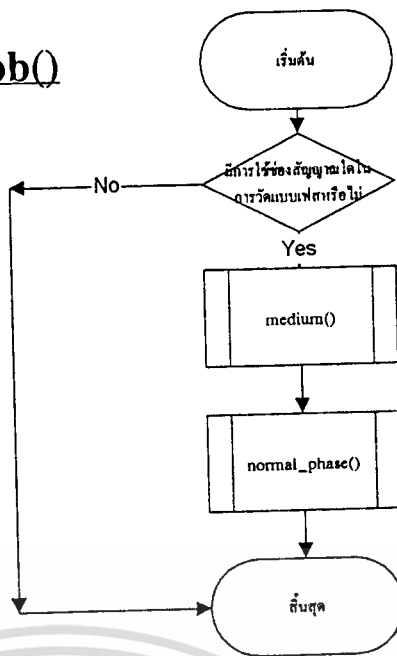
saves()



รูปที่ 9

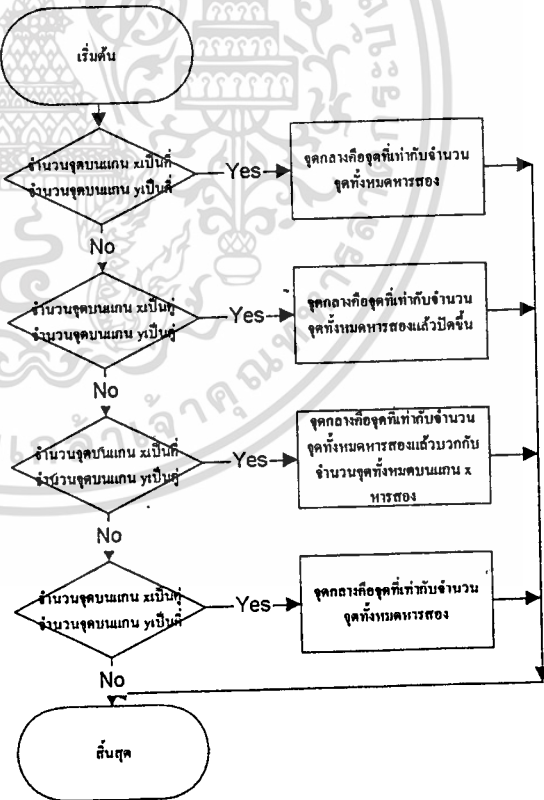
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

abb()



รูปที่ 10

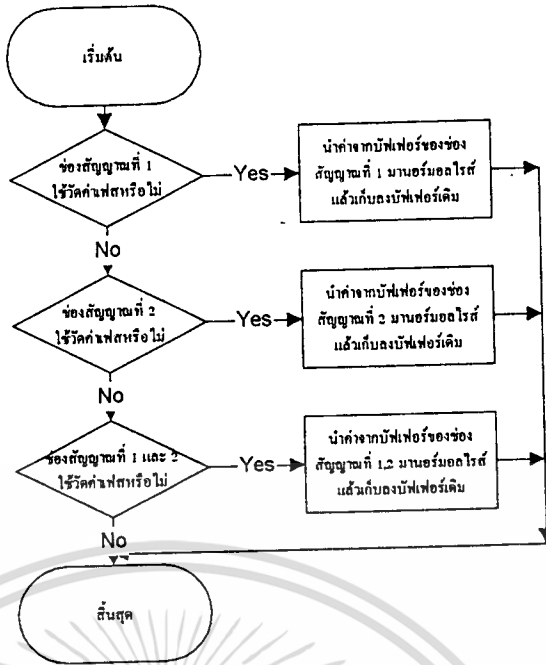
medium()



รูปที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

normal phase



รูปที่ 12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HPIB COMMAND

ในส่วนนี้จะอธิบายรายละเอียดของคำสั่งที่ใช้ในการควบคุมเครื่องวิเคราะห์โครงข่าย (network analyzer) ของ การ์ดอินเทอร์เฟซ HPIB

อุปกรณ์ที่ต้องการทำการติดต่อกันจะต้องมี การ์ดอินเทอร์เฟซ HPIB เป็นตั้งอินเทอร์เฟซระหว่าง เครื่องควบคุม (controller) กับอุปกรณ์ภายนอก เช่น เครื่องวิเคราะห์โครงข่าย (network analyzer) , เครื่องวิเคราะห์สเปกตรัมความถี่ (spectrum analyzer) , เครื่องออสซิลโลสโคป (ossilloscope) เป็นต้น

คำสั่งของ HPIB จะช่วยให้โปรแกรมมีความยืดหยุ่นมากขึ้นเช่น คนใช้สามารถส่งหรือรับ ข้อมูลเป็นตัวอักษรได้โดยตรง นอกจากนี้คำสั่งของ HPIB เกือบ 80 % จะใช้ติดต่อกับอินพุทและ เอาท์พุท ของคอมพิวเตอร์ ดังนั้นโปรแกรมที่ใช้จึงไม่จำเป็นต้องมีส่วนในการแปลง ข้อมูลแบบตัวอักษร ไปสู่ข้อมูลแบบตัวเลข จึงทำให้สามารถลดความยาวของโปรแกรมลงได้มาก

คำสั่งต่างๆ ของ HPIB ที่สำคัญมีดังนี้

1. IOABORT คำสั่งนี้ใช้สำหรับยกเลิกการติดต่อกับอุปกรณ์ภายนอกอื่นๆ ซึ่งคำสั่งนี้จะถูกควบคุมจากตัวควบคุม (controller) ปกติมักจะเป็นเครื่องคอมพิวเตอร์ (computer)

รูปแบบ IOABORT (select_code)

เมื่อ select code : เป็นหมายเลขของตัวอินเทอร์เฟซ ปกติจะเท่ากับ 7

2. IOCLEAR คำสั่งนี้ใช้เมื่อต้องการให้อุปกรณ์ภายนอก เช่น เครื่องวิเคราะห์โครงข่าย (network analyzer) อยู่ในสภาวะเริ่มต้น

รูปแบบ IOCLEAR (device_address)

IOCLEAR (select_code)

เมื่อ device address : เป็น address ของอุปกรณ์ที่ต้องการเคลียร์

select code : เป็นหมายเลขของตัวอินเทอร์เฟซที่อยู่บนอุปกรณ์ที่ต้องการเคลียร์

3. IOENTER เป็นคำสั่งที่ใช้สำหรับรับค่าข้อมูลแบบจำนวนจริง (read number) ซึ่งจะรับข้อมูลเข้ามาจนกว่าจะเกิดเงื่อนไขต่อไปนี้

-ได้รับจำนวนไบต์ (byte) ทั้งหมดแล้ว

-EOI (end or identify) มีค่าเป็น "1"

รูปแบบ IOENTER (device_address,data_REF)

IOENTER (select_code,data_REF)

เมื่อ device address : เป็น address อุปกรณ์

select code : เป็นหมายเลขของตัวอินเทอร์เฟซ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

data_REF : เป็นตัวแปรที่ใช้เก็บข้อมูล

4. IOMATCH เป็นคำสั่งที่ใช้กำหนดตัวอักษรที่คำสั่ง IOENTER จะนำไปใช้เป็นตัวปิดท้ายการส่งข้อมูลซึ่งตัวอักษรนี้มักจะเป็น line feed

รูปแบบ IOMATCH (select_code,character,flag)

เมื่อ select code : เป็นหมายเลขของตัวอินเตอร์เฟส

character : เป็นตัวอักษรหนึ่งตัวที่คำสั่ง IOENTER จะนำไปใช้เป็นตัวปิดท้าย

การส่งข้อมูล

flag : เป็นตัวแสดงสภาพ matching

5. IOOUTPUT คำสั่งนี้ใช้สำหรับส่งตัวอักษรไปยังอุปกรณ์ และหลังจากที่ส่งตัวอักษรออกไปแล้วจะส่ง EOL (end of line) หลังจากนั้น EOI (end of identify) จะมีค่าเป็น 1

รูปแบบ IOOUTPUT (device_address,data_REF,length)

IOOUTPUT (select_code,data_REF,length)

เมื่อ device address : เป็น address อุปกรณ์

select code : เป็นหมายเลขของตัวอินเตอร์เฟส

data REF : เป็น array ที่ใช้เก็บตัวอักษรที่จะส่งออกไป

6. IOTIMEOUT คำสั่งนี้ใช้สำหรับตั้งค่าเวลาที่เพื่อไว้ของตัวอินเตอร์เฟสสำหรับการกระทำด้านอินพุต และเอาท์พุท ที่ไม่สมบูรณ์

รูปแบบ IOTIMEOUT (select_code,timeout)

เมื่อ select code : เป็นหมายเลขของตัวอินเตอร์เฟส

time out : เป็นตัวกำหนดคาบเวลา ถ้ามีค่าเป็น 0 แสดงว่าไม่มีการตั้งค่าเวลาที่

เพื่อไว้ของตัวอินเตอร์เฟสสำหรับการกระทำด้านอินพุต และเอาท์พุท ที่ไม่สมบูรณ์ แต่ถ้าเป็นลบจะทำให้เกิดข้อผิดพลาดได้

อนึ่ง ตัวกำหนดคาบเวลาจะต้องตั้งไว้ที่ส่วนต้นของโปรแกรมเพื่อคชเพื่อเวลาสำหรับการทำงานด้านอินพุตและเอาท์พุทที่ไม่สมบูรณ์ซึ่งค่าตัวกำหนดคาบเวลานี้จะต้องเป็นค่าจำนวนจริงและมีหน่วยเป็นวินาที ค่าตัวกำหนดคาบเวลาจะมีค่าอยู่ระหว่าง 0-4096 วินาที

รูปแบบการเคลื่อนย้ายข้อมูล

ในหัวข้อนี้จะกล่าวถึงรายละเอียดในการส่งผ่านข้อมูลระหว่างเครื่องคอมพิวเตอร์ กับเครื่องวิเคราะห์โครงข่าย (network analyzer) ซึ่งในการส่งข้อมูลนี้จำเป็นที่จะต้องคำนึงถึงปัจจัยสำคัญ

3 ประการ ดังต่อไปนี้

1. รูปแบบของข้อมูลที่จะเคลื่อนย้าย

2. ประเภทของตัวแปรในโปรแกรมที่จะใช้รับส่งข้อมูล

3. ความเร็วที่ใช้ในการเคลื่อนย้ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การดูแลของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของความเร็วในการเคลื่อนย้ายข้อมูล ผู้ใช้สามารถเพิ่มความเร็วในการเคลื่อนย้ายข้อมูลได้หลายวิธีเช่น

-เคลื่อนย้ายข้อมูลเป็นกลุ่มๆ จะทำให้สามารถเพิ่มความเร็วในการเคลื่อนย้ายข้อมูลได้ประมาณ 10 เท่าเมื่อทำการเปรียบเทียบกับวิธีการเคลื่อนย้ายข้อมูลแบบเดิม

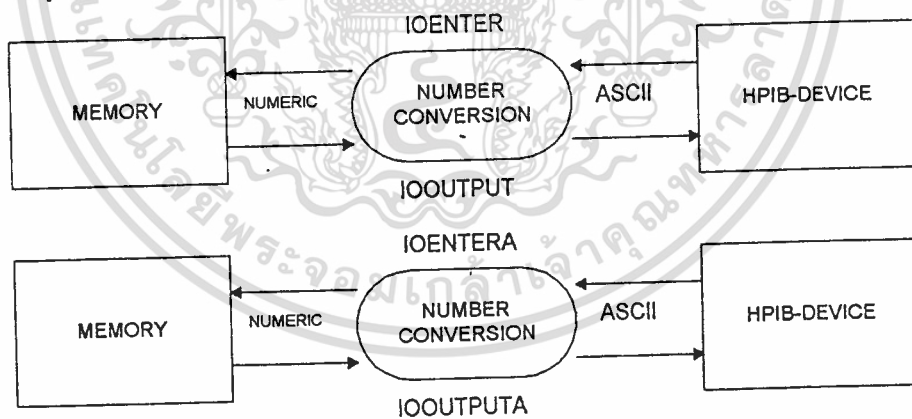
-ใช้สายส่งที่มีความเร็วสูง (high speed bus) จะทำให้มีความเร็วเพิ่มขึ้นประมาณ 30%

เมื่อเราพิจารณารูปแบบการเคลื่อนย้ายข้อมูลระหว่างเครื่องคอมพิวเตอร์และเครื่องวิเคราะห์โครงข่าย (network analyzer) จะมีอยู่หลายประเภทซึ่งผู้ใช้จะต้องเลือกให้เหมาะสมกับงานที่ทำอยู่

รูปแบบการเคลื่อนย้ายข้อมูลสามารถแบ่งออกได้ 2 แบบ ดังต่อไปนี้

1.การเคลื่อนย้ายรูปแบบข้อมูล (Formatted data transfer) อุปกรณ์ HPIB ส่วนใหญ่จะส่งและรับข้อมูลตัวเลข (numeric data) ด้วยตัวแอสกี (ASCII) ถึงแม้ว่าตัวแอสกี (ASCII) นี้ไม่สามารถที่จะนำไปทำการคำนวณได้ แต่ก็สามารถทำการเปลี่ยนกลับไปเป็นตัวเลข (numeric data) แล้วนำไปทำการคำนวณได้

การทำงาน-การเคลื่อนย้ายรูปแบบข้อมูล (Format data) นี้จำเป็นที่จะต้องทำการแปลงตัวเลขไปเป็นตัวอักษรและทำการเปลี่ยนจากตัวอักษรไปเป็นตัวเลข เช่น ถ้าอุปกรณ์ HPIB ส่งและรับข้อมูลแบบตัวแอสกี (ASCII) แล้ว จะต้องใช้คำสั่งใน HPIB library เพื่อเปลี่ยนข้อมูลนั้นกลับมาเป็นตัวเลขดังรูปด้านล่าง



รูปแสดงไดอะแกรมการเคลื่อนย้ายข้อมูลแบบการเคลื่อนย้ายรูปแบบข้อมูล (Formatted data transfer)

จากรูปจะเห็นได้ว่าการเคลื่อนย้ายข้อมูลแบบการเคลื่อนย้ายรูปแบบข้อมูล (Formatted data transfer) จำเป็นที่จะต้องใช้ขบวนการทางคอมพิวเตอร์ เพื่อการแปลงข้อมูล ดังนั้นการเคลื่อนย้ายแบบนี้ จะไม่สามารถทำแบบ DMA (direct memory access) ได้ นอกจากนี้ในขบวนการแปลงข้อมูลจะใช้เวลา น้อยมาก แต่โดยรวมแล้วการเคลื่อนย้ายข้อมูลแบบนี้จะช้ากว่าการเคลื่อนย้ายข้อมูลแบบกลุ่มข้อมูล (block data) ประมาณ 10 เท่า แต่อย่างไรก็ตามการเคลื่อนย้ายข้อมูลแบบนี้ก็มีประสิทธิภาพที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยหน่วยงานที่เกี่ยวข้องกับการศึกษาวิจัยและการค้า
องค์ประกอบของการเคลื่อนย้ายรูปแบบข้อมูล (Formatted data transfer) โดยขั้นตอนการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การป้องกัน EOI เป็น 1 สำหรับอินพุทและเอาต์พุท โดยปกติแล้ว EOI จะมีค่าเป็น 1 เมื่อข้อมูลไบต์สุดท้ายถูกเคลื่อนย้ายเข้ามา แต่ผู้ใช้สามารถทำให้ EOI เป็น 0 ได้โดยใช้คำสั่ง IOEOI

การเปลี่ยนตัวอักษร EOL-(end of line) สำหรับเอาต์พุทโดยปกติข้อมูลจะถูกปิดท้ายด้วยตัวอักษร carriage-return และ line feed แต่ผู้ใช้สามารถเปลี่ยนตัวอักษรนี้ได้โดยใช้คำสั่ง IOEOL

การตั้งเวลาสำหรับการเคลื่อนย้ายข้อมูล โดยปกติแล้วการเคลื่อนย้ายข้อมูลจะใช้เวลาเป็นมาตรฐานคงที่เสมอ แต่ผู้ใช้สามารถกำหนดเวลาในการเคลื่อนย้ายข้อมูลให้เร็วขึ้นได้ โดยใช้คำสั่ง IOFASTOUT

2.การเคลื่อนย้ายข้อมูลแบบกลุ่ม (block data transfer) อุปกรณ์ HPIB บางตัวสามารถส่งและรับข้อมูลแบบกลุ่มได้ ซึ่งแต่ละกลุ่มจะประกอบด้วยข้อมูลที่บรรจุอยู่ภายในเพื่อที่จะนำไปใช้คำนวณและแปลความหมายต่อไป

การเคลื่อนย้ายข้อมูลแบบกลุ่มนี้มีประสิทธิภาพในการเคลื่อนย้ายข้อมูลแบบ ไบต์ระหว่างอุปกรณ์ภายนอก กับหน่วยความจำของคอมพิวเตอร์ อย่างไรก็ตามข้อมูลยังไม่ถูกแปลความหมายในขณะที่ทำการเคลื่อนย้ายข้อมูล

โปรแกรมควบคุมเครื่องวิเคราะห์โครงข่าย

/* HP 8719C,8720C,8722A QuickC IPG */

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <cfunc.h>

#include <chpib.h>

#include <stdlib.h>

#include <dos.h>

#include <math.h>

#include <print.h>

#define isc 7L /*type of interface card(hpib)(long7)*/

#define instr 716L /*address of network analyzer(long16)*/

void error_handle (int, char *);

void output (char *);

void select_freq(void);

void select_power(void);

void initialize (void);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void opc(void);
void disp_promt(char *);
void meas_setup(void);
void marker_freq(void);
void calibrate(void);
void read(void);
void save(void);
void saves(void);
void normal_phas(void);
void medium(void);
void scanner(void);
void abb(void);

int mkr; /*use in marker_freq() and read()*****/
int mk; /*use in read() and save()*****/
int i; /*use in read() and scanner()*****/
int point; /*use in save() and scanner()***** */
int phas; /*use in meas_setup() ,scanner() and normal_phas()***/
int CntX,orgX,CntY,orgY; /*use in scanner() and medium*****/
float buf[11][1000]; /*use in read() and save()***** */
float medy; /*use in medium() and normal_phase()*****/

void error_handle (int error,char *routine)
{
    if (error != NOERR)
    {
        printf("HPIB error in call to %s: %d, %s\n",routine,error,errstr(error));
        exit(1);
    }
}

```

เอกสารนี้เป็น **return:** ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void output (char *sendstr)
{
error_handle (IOOUTPUTS (instr,sendstr,strlen(sendstr)),"IOOUTPUTS");
}

```

```

void opc()
{
int one =1;
char reply;
error_handle(IOENTERS (instr,&reply,&one),"IOENTERS");
}

```

```

void initialize ()
{
error_handle (IOTIMEOUT (isc,45.0),"IOTIMEOUT");
error_handle (IOABORT (isc),"IOABORT");
error_handle (IOCLEAR (isc),"IOCLEAR");
output ("PRES;CLES;");
}

```

```

void disp_promt( char *promt)
{
printf("%s",promt);
printf(", then press [RETURN] \n");
getche();
printf("\n");
}

```

```

void select_freq()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 {
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char ansf,cmd[80];
float f_start,f_stop;
printf("CHOOSE THE FORMAT OF FREQUENCY\n");
printf(" s : start and stop frequency\n");
printf(" c : center and span frequency\n");
printf("Enter character : ");
ansf = getche();
printf("\n");
switch(ansf)
{
case 's' : printf("Enter start frequency (MHz);");
scanf("%f",&f_start);
printf("Enter stop frequency (MHz);");
scanf("%f",&f_stop);
sprintf(cmd,"STAR %fMHZ;STOP %fMHZ;",f_start,f_stop);
output(cmd);
break;
case 'c' : printf("Enter center frequency (MHz);");
scanf("%f",&f_start);
printf("Enter span frequency (MHz);");
scanf("%f",&f_stop);
sprintf(cmd,"CENT %fMHZ;SPAN %fMHZ;",f_start,f_stop);
output(cmd);
break;
default : printf("out of menu,do again..\n");
select_freq();
}
}
void marker_freq()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float mkf;

char cmd[80];

printf("HOW MANY MARKER (1-5)?");

scanf("%d",&mkr);

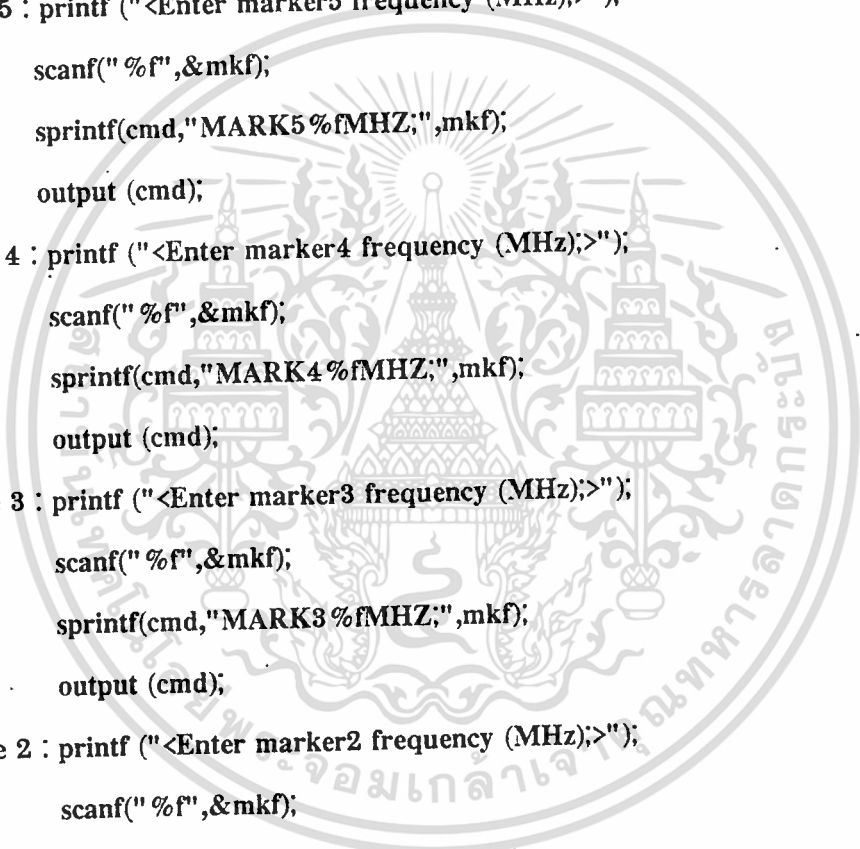
printf("\n");

switch(mkr)
{
    case 5 : printf("<Enter marker5 frequency (MHz);>");
              scanf("%f",&mkf);
              sprintf(cmd,"MARK5%fMHZ;",mkf);
              output (cmd);
    case 4 : printf("<Enter marker4 frequency (MHz);>");
              scanf("%f",&mkf);
              sprintf(cmd,"MARK4%fMHZ;",mkf);
              output (cmd);
    case 3 : printf("<Enter marker3 frequency (MHz);>");
              scanf("%f",&mkf);
              sprintf(cmd,"MARK3%fMHZ;",mkf);
              output (cmd);
    case 2 : printf("<Enter marker2 frequency (MHz);>");
              scanf("%f",&mkf);
              sprintf(cmd,"MARK2%fMHZ;",mkf);
              output (cmd);
    case 1 : printf("<Enter marker1 frequency (MHz);>");
              scanf("%f",&mkf);
              sprintf(cmd,"MARK1%fMHZ;",mkf);
              output (cmd);

              break;

    default : printf("<out of menu ,choose again!!>\n");
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 marker_freq();
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
void read(void)
{
    int length;
    char cmd[80];
    float val1[3],val2[3];
    length = 3;
    mkr = mkr;
    do
    {
        output("HOLD;");
        sprintf(cmd,"MARK%d",mkr);
        output ("CHAN1;MARKDISC;FORM5;OUTPMARK;");
        output(cmd);
        error_handle (IOENTERA(instr,val1,&length),"IOENTERA");
        printf("[point_ %3d] <..CHAN1 marker%d: %5.2f.mV.>_",i,mkr,val1[0]*1000.0);
        output ("CHAN2;MARKDISC;FORM5;OUTPMARK;");
        output(cmd);
        error_handle (IOENTERA(instr,val2,&length),"IOENTERA");
        printf("___ <..CHAN2 marker%d: %5.2f.degree.>\n",mkr,val2[0]);
        /* buf[5][i](channel1) and buf[10][i](channel2) are buffer of fifth marker */
        /* buf[4][i](channel1) and buf [9][i](channel2) are buffer of fourth marker */
        /* buf[3][i](channel1) and buf [8][i](channel2) are buffer of third marker */
        /* buf[2][i](channel1) and buf [7][i](channel2) are buffer of second marker */
        /* buf[1][i](channel1) and buf [6][i](channel2) are buffer of first marker */
        buf[mkr][0] =0;
        buf[mkr][i] =val1[0];
        buf[mkr+5][i]=val2[0];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
mkr=mkr-1;
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output("CONT;");
/** wait until get stable value */
delay(500);
}
while ( mkr > 0);
if (mkr==0) mkr=mk;
}

void medium()
{
float p1,p2,evod,odev;
int pp1,pp2;
pp1 = (CntX-orgX+1);
pp2 = (CntY-orgY+1);
p1=fmod(pp1,2);
p2=fmod(pp2,2);
/***** find medium (x = no.of column,y= no.of row) *****/
/** x = odd and y = odd */
if( (p1==1) && (p2==1) )
{
medy = ceil(i/2.0);
}

/*****/
/** x = even and y = even */
else if( (p1==0) && (p2==0) )
{
medy = (i/2.0)+(pp1)/2.0;
}

/*****/

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if( (p1==1) && (p2==0) )
{
evod = ceil((pp1)/2.0);
medy = (i/2.0)+evod;
    /* in case of x=1*/
    if ((pp1)==1)
        medy = i/2.0;
    /*******/
}
/*******/
/** x = even and y = odd ***/
else if( (p1==0) && (p2==1) )
{
medy = i/2;
}
/*******/
/**** in case of error programming ***/
else {
    printf("error programming(medium)\n");
    exit(0);
}
printf("\007");
}

```

```

void normal_phas()
{
    int yy,xx,yy1,xx1;
    float norm;
    if (phas==1)

```

```

/* (ch.1) normalize loop */
for(xx=mk;xx>0;xx--)
{
    for(yy=i;yy>0;yy--)
    {
        norm = buf[xx][medy];
        buf[xx][yy]= (buf[xx][yy]-buf[xx][medy]);
        if (buf[xx][yy]==0) buf[xx][medy]=norm;
    }
    buf[xx][medy] = 0;
}
/*****/
}
else if (phas==2)
{
    printf("/*(ch.2) normalize loop *\n");
    /* (ch.2) normalize loop */
    for(xx1=mk+5;xx1>5;xx1--)
    {
        for(yy1=i;yy1>0;yy1--)
        {
            norm = buf[xx1][medy];
            buf[xx1][yy1]= (buf[xx1][yy1]-buf[xx1][medy]);
            if (buf[xx1][yy1]==0) buf[xx1][medy]=norm;
        }
        buf[xx1][medy] = 0;
    }
}
/*****/
}

```

เอกสารนี้ **else if (phas==3)** ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
printf("/*(ch.1 and ch.2) normalize loop */\n");
/* (ch.1 and ch.2) normalize loop */
for(xx=mk;xx>0;xx--)
{
for(yy=i;yy>0;yy--)
{
norm = buf[xx][medy];
buf[xx][yy]= (buf[xx][yy]-buf[xx][medy]);
if (buf[xx][yy]==0) buf[xx][medy]=norm;
}
buf[xx][medy] = 0;
}
for(xx1=mk+5;xx1>5;xx1--)
{
for(yy1=i;yy1>0;yy1--)
{
norm = buf[xx1][medy];
buf[xx1][yy1]= (buf[xx1][yy1]-buf[xx1][medy]);
if (buf[xx1][yy1]==0) buf[xx1][medy]=norm;
}
buf[xx1][medy] = 0;
}
/*****
}
/*****in case of error programming *****/
else
{ printf("error in programming(normal_phase)\n");
exit(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void save()
{ FILE *data;
  FILE *data1;
  int k,h;
  char finame[10][30];
  printf("****packing data****\n\n");
  /*file1 for ch1 marker1 , file6 for ch2 marker1
   file2 for ch1 marker2 , file7 for ch2 marker2
   file3 for ch1 marker3 , file8 for ch2 marker3
   file4 for ch1 marker4 , file9 for ch2 marker4
   file5 for ch1 marker5 , file10 for ch2 marker5*/
  printf("file1 for ch1 marker1 , file6 for ch2 marker1\n");
  printf("file2 for ch1 marker2 , file7 for ch2 marker2\n");
  printf("file3 for ch1 marker3 , file8 for ch2 marker3\n");
  printf("file4 for ch1 marker4 , file9 for ch2 marker4\n");
  printf("file5 for ch1 marker5 , file10 for ch2 marker5\n");
  for(h=0;h<mk,h++)
  {
  printf("Enter saved file%d name:=>",h+1);
  scanf(" %s",&finame[h]);
  printf("Enter saved file%d name:=>",h+6);
  scanf(" %s",&finame[h+5]);
  }
  k=0;
  while(k<mk)
  {
  /******channel 1******/
  printf("start saving file of marker%d ,channel 1.....)\n",mk);
  if ((data = fopen(finame[k],"wb")) == NULL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    exit(0);
}
fwrite(&buf[k+1][1],sizeof(float),point,data);
fclose(data);
/*****channel 2*****/
printf("start saving file of marker%d ,channel 2.....:\n",mk);
if ((data1 = fopen(fname[k+5],"wb")) == NULL)
{ printf("Error in open file\n");
  exit(0);
}
fwrite(&buf[k+6][1],sizeof(float),point,data1);
fclose(data1);
k++;
}
}
void saves()
{ FILE *data;
  FILE *data1;
  int k,h;
  char fname[10][80];
  printf("/****packing data****/\n\n");
  /*file1 for ch1 marker1 , file6 for ch2 marker1
   file2 for ch1 marker2 , file7 for ch2 marker2
   file3 for ch1 marker3 , file8 for ch2 marker3
   file4 for ch1 marker4 , file9 for ch2 marker4
   file5 for ch1 marker5 , file10 for ch2 marker5*/
  printf("file1 for ch1 marker1 , file6 for ch2 marker1\n");
  printf("file2 for ch1 marker2 , file7 for ch2 marker2\n");
  printf("file3 for ch1 marker3 , file8 for ch2 marker3\n");
  printf("file4 for ch1 marker4 , file9 for ch2 marker4\n");

```

```

printf("file5 for ch1 marker5 , file10 for ch2 marker5\n");

for(h=0;h<mk;h++)
{
if (phas==1)
{
printf("Enter saved file%d name:=>",h+1);
scanf(" %s",&fname[h]);
}
else if (phas==2)
{
printf("Enter saved file%d name:=>",h+6);
scanf(" %s",&fname[h+5]);
}
else if (phas==3)
{
printf("Enter saved file%d name:=>",h+1);
scanf(" %s",&fname[h]);
printf("Enter saved file%d name:=>",h+6);
scanf(" %s",&fname[h+5]);
}
else
{
printf("error in saves\n");
printf("\007");
exit(0);
}
}
k=0;
while(k<mk)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (phas==1)
{
/*****channel 1*****/
printf("start saving file of marker%d ,channel 1.....)\n",mk);
if ((data = fopen(finame[k],"wb")) == NULL)
{
printf("Error in open file\n");
exit(0);
}
fwrite(&buf[k+1][1],sizeof(float),point,data);
fclose(data);
k++;
}
else if (phas==2)
{
/*****channel 2*****/
printf("start saving file of marker%d ,channel 2.....)\n",mk);
if ((data1 = fopen(finame[k+5],"wb")) == NULL)
{ printf("Error in open file\n");
exit(0);
}
fwrite(&buf[k+6][1],sizeof(float),point,data1);
fclose(data1);
k++;
}
else if (phas==3)
{
/*****channel 1*****/
printf("start saving file of marker%d ,channel 1.....)\n",mk);

```

```

    { printf("Error in open file\n");
      exit(0);
    }

fwrite(&buf[k+1][1],sizeof(float),point,data);
fclose(data);
/*****channel 2*****/
printf("start saving file of marker%d ,channel 2.....)\n",mk);
if ((data1 = fopen(fname[k+5],"wb")) == NULL)
    { printf("Error in open file\n");
      exit(0);
    }
fwrite(&buf[k+6][1],sizeof(float),point,data1);
fclose(data1);
k++;
}
else
{
printf("error in saves\n");
printf("\007");
exit(0);
}

}

}

void meas_setup()
{
    int ans,ans3;
    char ans1,ans2,form1[30],form2[30],head1[70],head2[70],cmd[80],cmd1[80];

```

เอกสารนี้เป็น **phas = 0**; ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("MEASUREMENT SETUP");
getch();
clrscr(); gotoxy(30,8);
printf("  SETUP CHANNEL 1 (11 or 21) \n");
printf("  11: FOR REFLECTION \n");
printf("  21: FOR TRANSMISSION \n");
scanf("%d",&ans);
clrscr();
if(ans==11)
{
gotoxy(30,8);
printf("  SETUP CHANNEL 1 FOR REFLECTION \n");
strcpy(head1,"SETUP CHANNEL 1 FOR REFLECTION");
}
else if(ans==21)
{
gotoxy(30,8);
printf("  SETUP CHANNEL 1 FOR TRANSMISSION \n");
strcpy(head1,"SETUP CHANNEL 1 FOR TRANSMISSION");
}
else
{
printf("out of menu,do again\n");
meas_setup();
}
clrscr();
printf("  SELECT FORMAT \n");
printf("  l : LOG MAGNITUDE\n");
printf("  p : PHASE\n");
printf("if you choose PHASE ,we will normalize phase automatically \n");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf(" d : DELAY\n");
printf(" m : SMITH CHART\n");
printf(" o : POLAR\n");
printf(" n : LIN`MAGNITUDE\n");
printf(" r : REAL\n");
printf(" s : SWR\n");
ans1=getch();
switch(ans1)
{
case 'l' : sprintf(cmd,"CHAN1;S%d;LOGM;",ans);
            output (cmd);strcpy(form1,"LOG MAGNITUDE");
            break;
case 'p' : sprintf(cmd,"CHAN1;S%d;PHAS;",ans);
            output (cmd);strcpy(form1,"PHASE");
/* normalize phase */
            phas=phas+1;
            break;
case 'd' : sprintf(cmd,"CHAN1;S%d;DELY;",ans);
            output (cmd);strcpy(form1,"DELAY");
            break;
case 'm' : sprintf(cmd,"CHAN1;S%d;SMIC;",ans);
            output (cmd);strcpy(form1,"SMITH CHART");
            break;
case 'o' : sprintf(cmd,"CHAN1;S%d;POLA;",ans);
            output (cmd);strcpy(form1,"POLAR");
            break;
case 'n' : sprintf(cmd,"CHAN1;S%d;LINM;",ans);
            output (cmd);strcpy(form1,"LIN MAGNITUDE");
            break;
case 'r' : sprintf(cmd,"CHAN1;S%d;REAL;",ans);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        output (cmd);strcpy(form1,"REAL");
        break;
    case 's' : sprintf(cmd,"CHAN1;S%d;SWR;",ans);
        output (cmd);strcpy(form1,"SWR");
        break;
}

clrscr();
gotoxy(30,8);
printf(head1);printf("\n"); gotoxy(35,9);
printf(form1);printf("\n");
printf(" PRESS ANY KEY TO CONTINUE\n");
getch();
clrscr();
printf("SETUP CHANNEL 2 ? (Y or N)\n");
ans2=getch();
clrscr();
if (ans2=='y')
{
    clrscr(); gotoxy(30,8);
    printf("  SETUP CHANNEL 2 (11 or 21) \n");
    printf("  11: FOR REFLECTION \n");
    printf("  21: FOR TRANSMISSION \n");
    scanf("%d",&ans3);
    clrscr();
    if(ans==11)
    {
        gotoxy(30,8);
        printf("  SETUP CHANNEL 2 FOR REFLECTION  \n");
        strcpy(head2,"SETUP CHANNEL 2 FOR REFLECTION");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (ans==21)
{
gotoxy(30,8);
printf("  SETUP CHANNEL 2 FOR TRANSMISSION \n");
strcpy(head2,"SETUP CHANNEL 2 FOR TRANSMISSION");
}
else
{
printf("out of menu,do again.");
meas_setup();
}
clrscr();
printf(" SELECT FORMAT \n");
printf(" l : LOG MAGNITUDE\n");
printf(" p : PHASE\n");
printf("if you choose PHASE,we will nomalize phase automatically \n");
printf(" d : DELAY\n");
printf(" m : SMITH CHART\n");
printf(" o : POLAR\n");
printf(" n : LIN MAGNITUDE\n");
printf(" r : REAL\n");
printf(" s : SWR\n");
ans1=getch();
switch(ans1)
{
case 'l' :sprintf(cmd,"CHAN2;S%d;LOGM;",ans);
output (cmd);strcpy(form2,"LOG MAGNITUDE");
break;
case 'p' :sprintf(cmd,"CHAN2;S%d;PHAS;",ans);
output (cmd);strcpy(form2,"PHASE");

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* normalize phase */
phas = phas+2;
break;
case 'd' :sprintf(cmd,"CHAN2;S%d;DELY;",ans);
        output (cmd);strcpy(form2,"DELAY");
        break;
case 'm' :sprintf(cmd,"CHAN2;S%d;SMIC;",ans);
        output (cmd);strcpy(form2,"SMITH CHART");
        break;
case 'o' :sprintf(cmd,"CHAN2;S%d;POLA;",ans);
        output (cmd);strcpy(form2,"POLAR");
        break;
case 'n' :sprintf(cmd,"CHAN2;S%d;LINM;",ans);
        output (cmd);strcpy(form2,"LIN MAGNITUDE");
        break;
case 'r' :sprintf(cmd,"CHAN2;S%d;REAL;",ans);
        output (cmd);strcpy(form2,"REAL");
        break;
case 's' :sprintf(cmd,"CHAN2;S%d;SWR;",ans);
        output (cmd);strcpy(form2,"SWR");
        break;
}
clrscr();
gotoxy(30,8);
printf(head2);printf("\n"); gotoxy(35,9);
printf(form2);printf("\n");
printf(" PRESS ANY KEY TO CONTINUE\n");
}

```

```

gotoxy(30,6);
printf(head1);gotoxy(35,7);
printf(form1);gotoxy(30,8);
if(ans2=='y')
{
printf(head2);gotoxy(35,9);
printf(form2);gotoxy(30,20);
}
printf("PRESS ANY KEY TO CONTINUE\n");
getch();
output("DUACON;");
}

```

```

void select_power()
{
int ansp;
clrscr();
printf("SELECT POWER LEVEL \n");
printf(" 1 : range (-10 to +10 dBm)\n");
printf(" 2 : range (-15 to +5 dBm)\n");
printf(" 3 : range (-20 to 0 dBm)\n");
printf(" 4 : range (-25 to -5 dBm)\n");
printf(" 5 : range (-30 to -10 dBm)\n");
printf(" 6 : range (-35 to -15 dBm)\n");
printf(" 7 : range (-40 to -20 dBm)\n");
printf(" 8 : range (-45 to -25 dBm)\n");
printf(" 9 : range (-50 to -30 dBm)\n");
printf(" 10 : range (-55 to -35 dBm)\n");
printf(" 11 : range (-60 to -40 dBm)\n");
printf(" 12 : range (-65 to -45 dBm)\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("WHICH RANGE ??: ");
scanf("%d",&ansp);
switch(ansp)
{
case '1' : output ("POWE10;"); break;
case '2' : output ("POWE5;"); break;
case '3' : output ("POWE0;"); break;
case '4' : output ("POWE-5;"); break;
case '5' : output ("POWE-10;"); break;
case '6' : output ("POWE-15;"); break;
case '7' : output ("POWE-20;"); break;
case '8' : output ("POWE-25;"); break;
case '9' : output ("POWE-30;"); break;
case '10' : output ("POWE-35;"); break;
case '11' : output ("POWE-40;"); break;
case '12' : output ("POWE-45;"); break;
default : printf("out of menu,choose again..\n");
        select_power();
}
}

```

```

void calibrate()
{
output ("CALK35MM;CLES;CALIFUL2;REFL;");
disp_promt("Connect OPEN at port 1 ");
output ("OPC?;CLASS11A;");
opc();
disp_promt("Connect SHORT at port 1 ");
output ("OPC?;CLASS11B;");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

disp_promt("Connect BROADBAND LOAD at port 1 ");
output ("CLASS11C;OPC?;STANA;");
opc();
disp_promt("Connect OPEN at port 2");
output ("OPC?;CLASS22A;");
opc();
disp_promt ("Connect SHORT at port 2 ");
output ("OPC?;CLASS22B;");
opc();
disp_promt("Connect BROADBAND LOAD at port 2 ");
output ("CLASS22C;OPC?;STANA;");
opc();
output ("REFD;");
printf ("Computing reflection calibration coefficients :(\n");
output ("TRAN;");
disp_promt("Connect THRU [port1 to port 2]");
printf ("Measuring forward transmission\n");
output ("OPC?;FWDT;");
opc();
output ("OPC?;FWDM;");
opc ();
printf ("Measuring reverse transmission :(\n");
output ("OPC?;REVT;");
opc();
output ("OPC?;REVM;");
opc();
output ("TRAD;");
disp_promt("Isolate test ports");
output ("AVERFACT16;AVERON;ISOL;");
printf ("Measuring reverse isolation\n");

```

```

output ("OPC?;REVI;");
opc();
printf ("Measuring forward isolation\n");
output ("OPC?;FWDI;");
opc();
output ("ISOD;AVEROFF;");
printf ("computing calibration coefficients :(\n");
output ("OPC?;SAV2;");
opc();
printf("DONE :)\n");
}
void scanner()
{
int endp;
int x,y;
int argv[4];
char com[10];
clrscr();
endp = 0;
printf("CntX=");
scanf(" %d",&argv[1]);
printf("CntY=");
scanf(" %d",&argv[2]);
printf("step=");
scanf(" %d",&argv[3]);
CntX = argv[1];
CntY = argv[2];
point = (CntX+1)*(CntY+1);
sprintf(com,"PA0,0;");

```



เอกสารนี้ **fprintf(stdprn,"%s",com);** การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for ( y = 0; y <= CntY; y++)
{
/*first point*/
if ( y < 1)
{ delay(200);
  i=1;
  read();/*access*/
  printf("\007");
}
/*****/
/*if y is not equal to last point of scanning and not in the first row*/
if ( y != 0 )
if (( CntY != 0 )||( y != CntY ))
{
  sprintf(com,"PR0,%d;",argv[8]);
  fprintf(stdprn," %s",com);
}
/*****/
/*if y is in other row*/
if ( y >= 1 )
{
  delay(200);
  i++;
  read();/*access*/
  printf("\007");
}
/*****/
/*scanning in x axis */
if (endp)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for ( x = CntX-1; x >= 0; x--)
{
sprintf(com,"PR-%d,0;",argv[3]);
fprintf(stdprn,"%s",com);
delay(200);
i++;
read();/*access*/
printf("\007");
}
}

endp = 0;
}
/*****/
/*scanning in x axis*/
else
{
for ( x = 0; x <= CntX-1; x++)
{
sprintf(com,"PR %d,0;",argv[3]);
fprintf(stdprn,"%s",com);
{ delay(200);
i++;
read();/*access*/
printf("\007");
}
}

endp = 1;
}
/*****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void abb()
{
/******normalize phase******/
  if (phas != 0)
  {
    medium();
    normal_phas();
  }
}

void main()
{
  char cali,both;
  initialize();
  meas_setup();
  /*select_power();*/ /* (cannot use)*/
  select_freq();
  /** select to calibrate or not ***/
  printf("do you want to calibrate??(y=yes!/any key=no!)\n");
  cali = getch();
  if (cali=='y')
  { calibrate();
  }

/*******/

  marker_freq();
  scanner();
  printf("\n\n\n");

/** if you choose phase you must ******/

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( phas!=0 )
{ printf("[ save before normalize ]\n");
  printf("\007");
  printf("press [ENTER] to cont..\n");
  getch();
  printf("Do you want to save both of channel?..(y/n).. \n");
  printf("y= save ch1 and ch2\n");
  printf("n= save only ch. that use phase measurement\n");
  both = getch();
  if (both=='y') save();
  else saves();
  abb();
}
/** save after normalize and root2 ( if you choose phase) ***/
/** or save if you're not choose of phase) *****/
printf("\n\n");
printf("!! save data++ !!\n");
printf("\007");
printf("press [ENTER] to cont..\n");
getch();
save();
/*****/
printf("end of program :) ");
sound(100);
delay(300);
nosound();
getch();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการแปลงสนามระยะใกล้เป็นสนามระยะไกล และโปรแกรมพล็อตกราฟ

ในโปรแกรมนี ในเมนูหลักจะแบ่งเป็น 4 รายการ คือ

1. การแปลงสนามระยะใกล้เป็นสนามระยะไกล
2. การพล็อตกราฟสี่เหลี่ยม(RECTANGULAR GRAPH)
3. การพล็อตกราฟวงกลม(POLAR GRAPH)
4. ออกจากโปรแกรม

ซึ่งเราสามารถอธิบายการทำงานของแต่ละส่วนได้ดังนี้

1. การแปลงสนามระยะใกล้เป็นสนามระยะไกล สำหรับส่วนที่หนึ่งต้องกำหนดค่าเริ่มต้นในการคำนวณคือ ความถี่ จำนวนจุด และระยะห่างในการสแกนทั้งแกน x และแกน y กำหนดชื่อไฟล์ที่เก็บค่าขนาดและเฟสจากการสแกนเพื่อเป็นข้อมูลในการคำนวณ และชื่อไฟล์เพทเทิร์นของโพรับ ทั้งแนวตั้งและแนวนอนสำหรับปรับแต่งค่าสนามที่คำนวณได้ จากนั้นจึงสลับตำแหน่งข้อมูลในแถวที่เป็นเลขคี่ เพราะในการสแกนแบบ ราสเตอร์ (raster) เมื่อสแกนจบ 1 แถวแล้วจะลงมา สแกนในแถวต่อไปโดยจะสแกนในทิศทางกลับกันทำให้ลำดับของข้อมูลที่เก็บกับลำดับของตำแหน่งจุดที่สแกนสลับกัน เช่น เมื่อสแกนจาก ตำแหน่งที่ (0,0) มาถึงตำแหน่งที่ (5,0) แล้วลงมา สแกนในแถวที่ 2 ซึ่งเป็นตำแหน่งที่ (5,1) ในส่วนของการเก็บข้อมูลเราจะเก็บข้อมูลเป็นลำดับที่ 7 แต่ในการคำนวณเราจะต้องใช้ข้อมูลที่ตำแหน่ง (0,1) มาเป็นข้อมูลลำดับที่ 7 แทนดังนั้นเราจะต้องทำการสลับตำแหน่งข้อมูลแล้วค่อยมาคำนวณตามสมการดังต่อไปนี้

$$dx dy \sum_{m=0}^{px-1} e^{jkyh \cdot (pty/2-m) dy} \sum_{n=0}^{px-1} \text{amp}[nn] e^{jphi[nn]} e^{jkxh \cdot (ptx/2-n) dx}$$

สำหรับเพทเทิร์นของสนามระยะไกลในแนวนอน(horizontal) และ

$$dx dy \sum_{m=0}^{px-1} e^{jkyv \cdot (pty/2-m) dy} \sum_{n=0}^{px-1} \text{amp}[nn] e^{jphi[nn]} e^{jkcv \cdot (ptx/2-n) dx}$$

สำหรับเพทเทิร์นของสนามระยะไกลในแนวตั้ง(vertical)

โดยที่ k_xh, k_{xv} คือ k_x ที่อยู่ในแนวแกนแนวนอน และแกนตั้งตามลำดับ ซึ่ง

$$k_x = k_0 \sin(z) \cos(\phi)$$

ส่วน k_yh, k_{yv} คือ k_y ที่อยู่ในแนวแกนแนวนอน และแกนตั้งตามลำดับ

$$k_y = k_0 \sin(z) \sin(\phi)$$

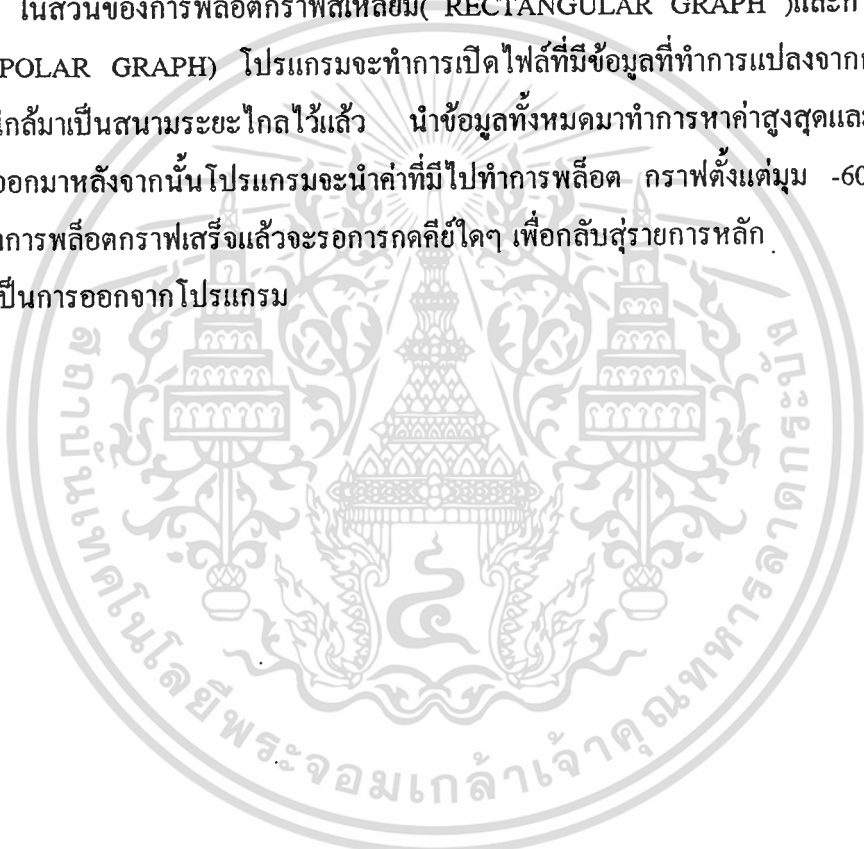
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

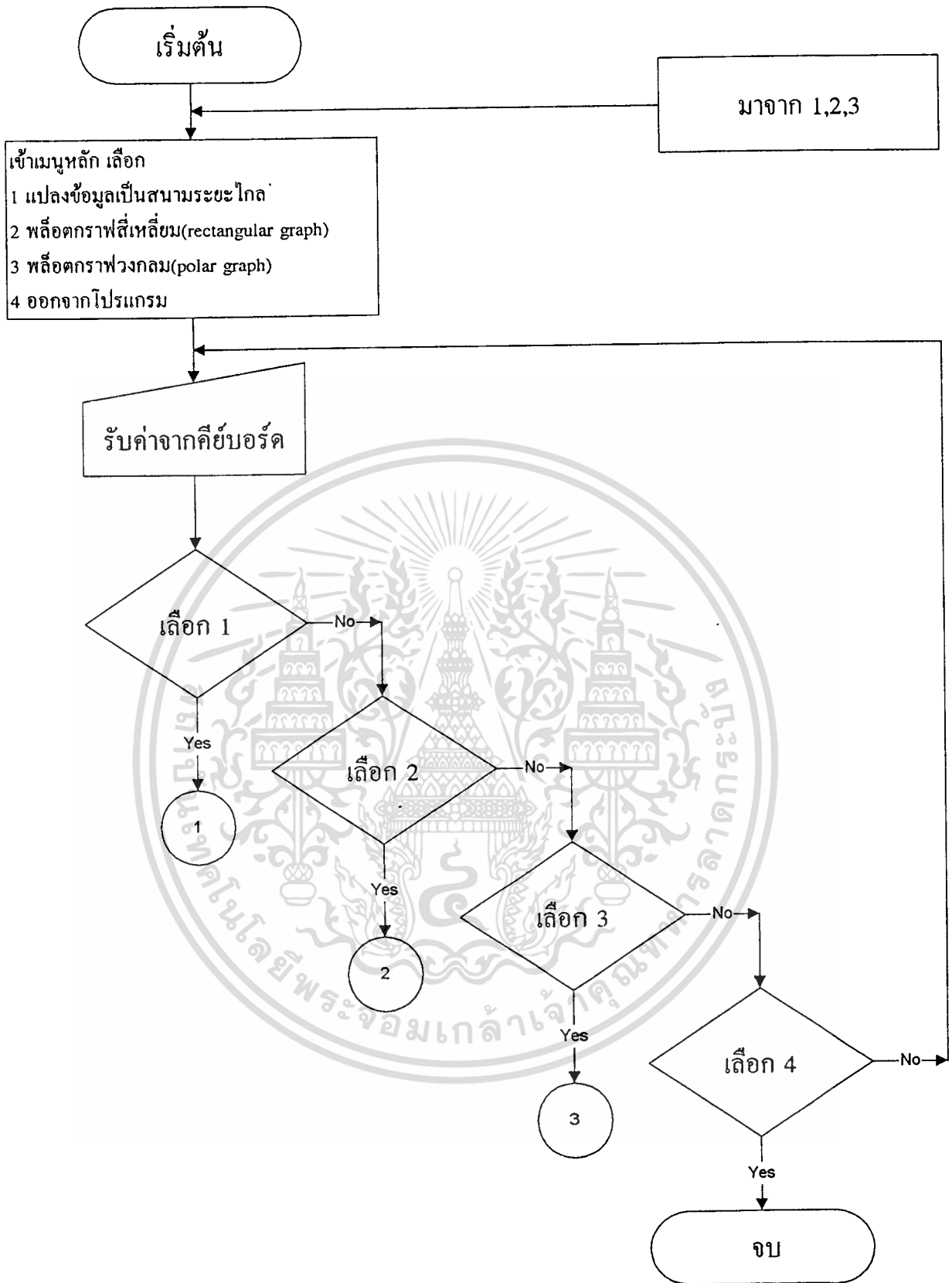
z คือค่า θ ที่มีค่าตั้งแต่ -60 ถึง $+60$ องศา และ $k_0 = 2\pi f/c$ เป็นค่าเวฟนัมเบอร์

หลังจากนั้นจึงต้องทำการปรับค่าด้วยสนามของโพรับ จากทฤษฎีจะได้สนามที่ปรับค่าด้วยโพรับสำหรับสนามในแนวแกนนอนเป็น $k_{0\text{outh}}[z]$ $\text{probeh}[z]$ และสำหรับสนามในแนวแกนตั้งเป็น $k_{0\text{outv}}[z]$ $\text{probev}[180-z]$ outh และ outv เป็นสนามระยะไกลของสายอากาศที่ได้จากการแปลง ส่วน probeh และ probev เป็นสนามระยะไกลของโพรับที่ได้จากการวัดระยะไกลเอทพุทที่ได้จะอยู่ในรูปแนวแกนตั้งและแกนนอน (vertical และ horizontal) โปรแกรมจะนำค่าที่คำนวณออกมาทำการเก็บลงไฟล์แล้วกลับมาที่รายการหลัก

2,3. ในส่วนของกราฟสี่เหลี่ยม (RECTANGULAR GRAPH) และการพล็อตกราฟวงกลม (POLAR GRAPH) โปรแกรมจะทำการเปิดไฟล์ที่มีข้อมูลทำการแปลงจากการวัดแบบสนามระยะไกลมาเป็นสนามระยะไกลไว้แล้ว นำข้อมูลทั้งหมดมาทำการหาค่าสูงสุดและต่ำสุดและพิมพ์ตารางออกมาหลังจากนั้นโปรแกรมจะนำค่าที่มีไปทำการพล็อตกราฟตั้งแต่มุม -60 ถึง $+60$ องศา เมื่อทำการพล็อตกราฟเสร็จแล้วจะรอการกดคีย์ใดๆ เพื่อกลับสู่รายการหลัก

4. เป็นการออกจากโปรแกรม



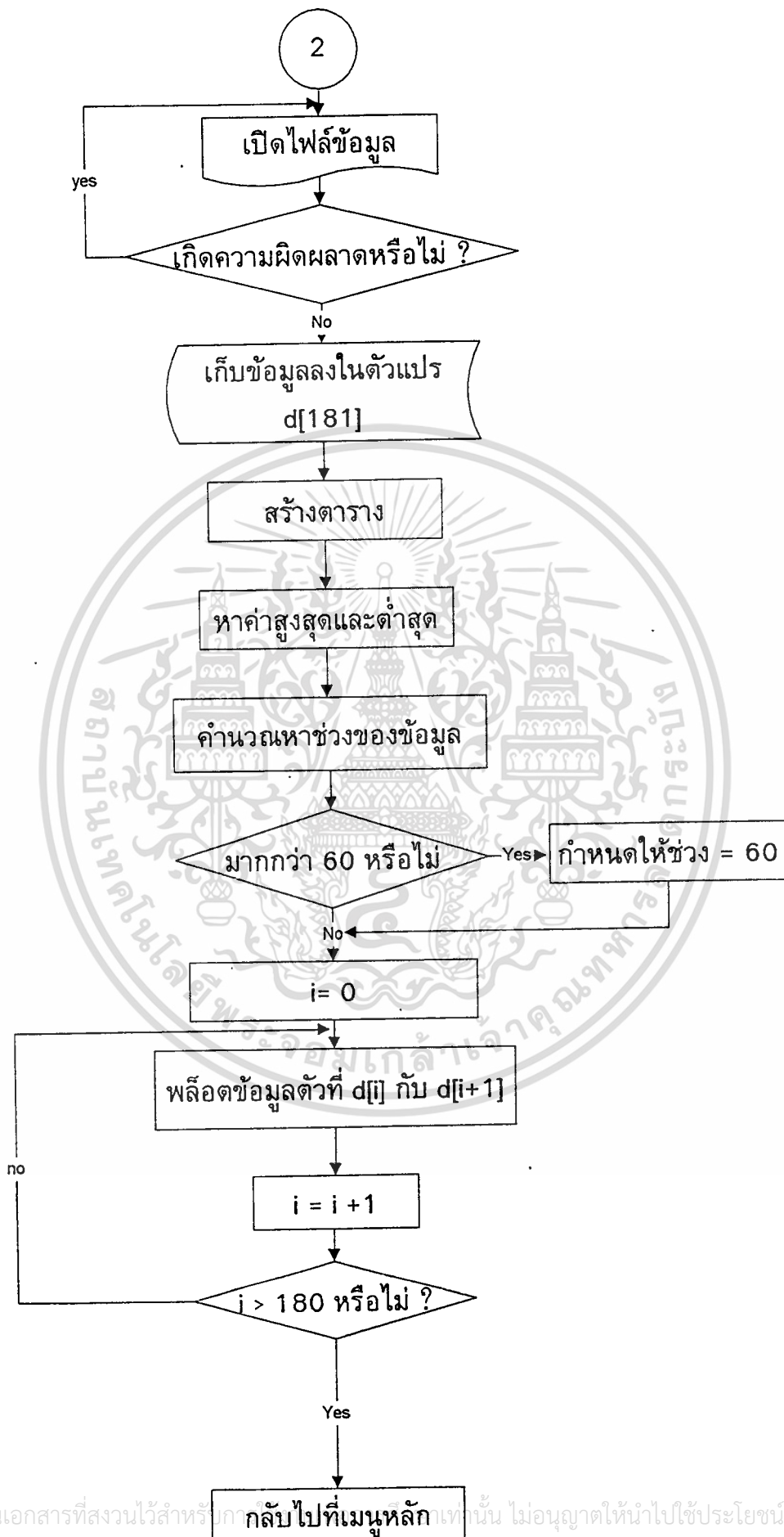


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

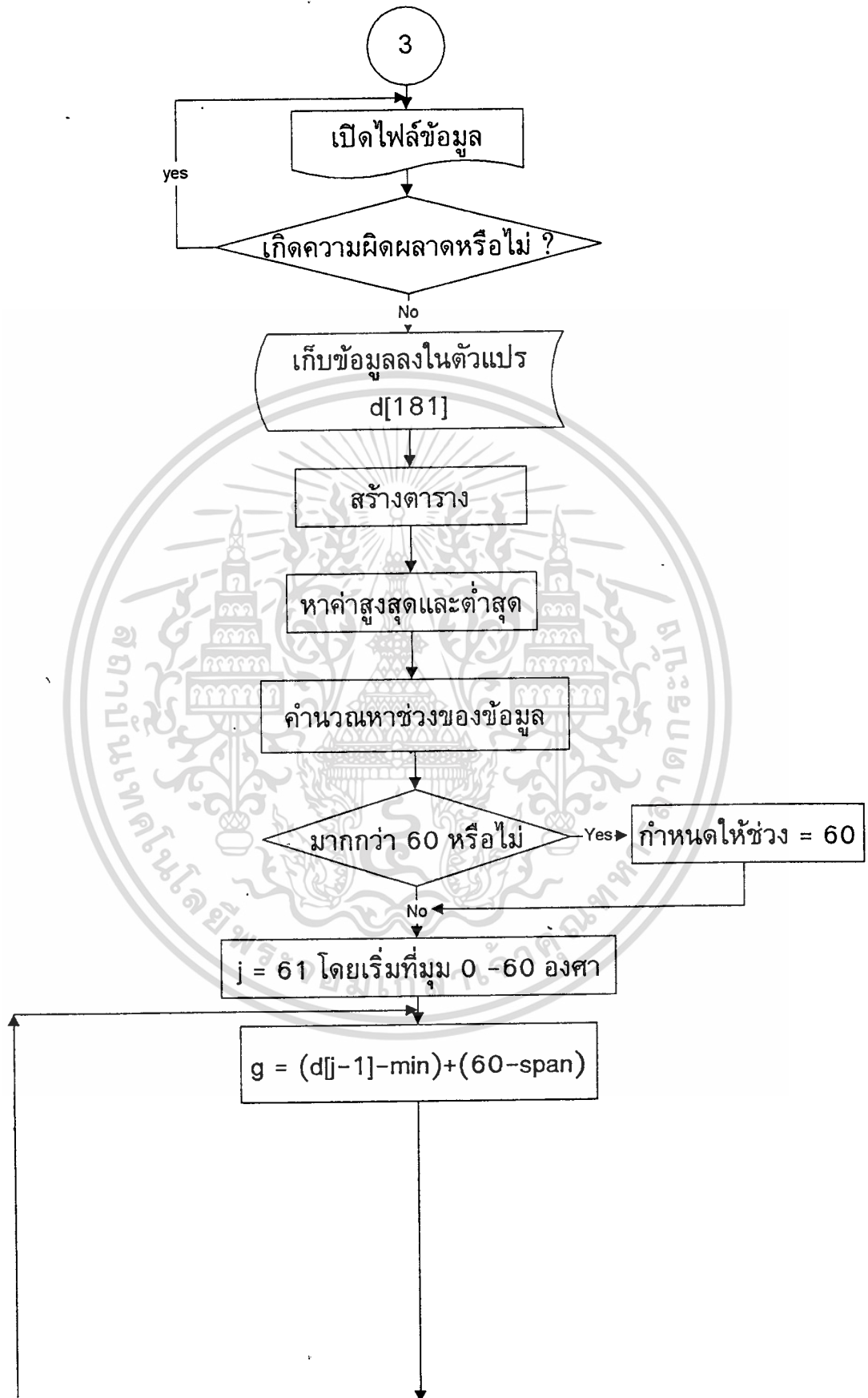


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงหรือเผยแพร่ข้อมูลข้างต้นไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

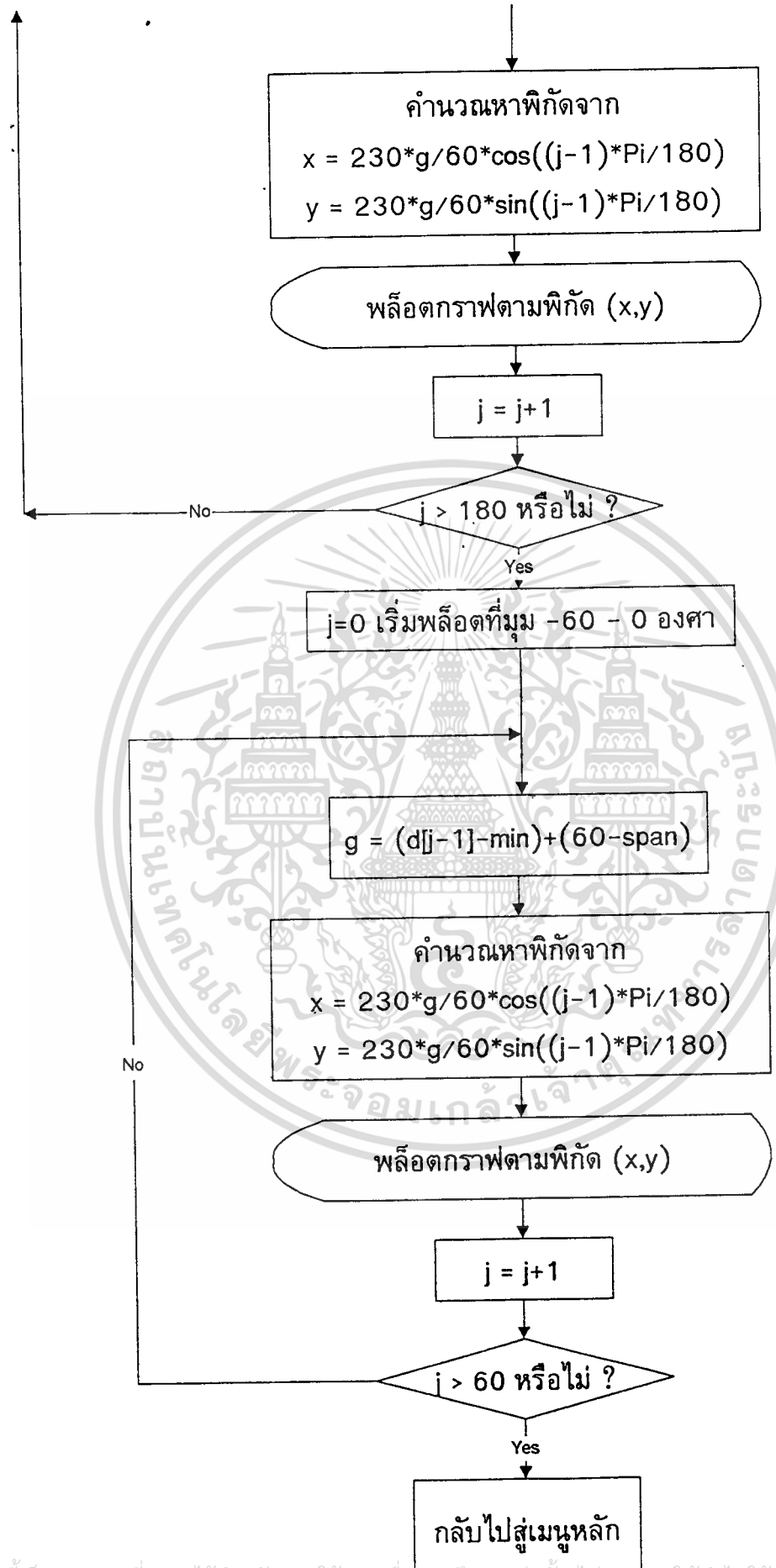
กลับไปเมนูหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมคำนวณและพล็อตกราฟ

```

#include <stdio.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <string.h>
#define dat1 2052
#define PI 3.141592654
#define C 299792458
typedef struct cmplx{
    float r;
    float i;
}complex;
float amp[dat1],phi[dat1],outv[181],outh[181],probeh[181],probev[181];
float dx,dy,freq;
int dat,ptx,pty;

void getdata();
void swap();
void outdata();
void initial();
void dft();
void rectan_plot();
void polar_plot();
void color_line();
void color_line1();

main() /* MAIN MENU */
{ char choice;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i;
```

```
clrscr();
```

```
gotoxy(22,12);
```

```
printf("NEAR FIELD TO FAR FIELD TRANSFORMATION\n");
```

```
getch();
```

```
choose: clrscr();
```

```
gotoxy(24,8);
```

```
printf("1. TRANSFORM NEAR FIELD TO FAR FIELD");
```

```
gotoxy(24,10);
```

```
printf("2. PLOT RECTANGULAR GRAPH");
```

```
gotoxy(24,12);
```

```
printf("3. PLOT POLAR GRAPH");
```

```
gotoxy(24,14);
```

```
printf("4. EXIT PROGRAM");
```

```
gotoxy(24,16);
```

```
printf("CHOOSE 1 2 3 OR 4 ");
```

```
choice=getche();
```

```
if ((choice!='1')&&(choice!='2')&&(choice!='3')&&(choice!='4'))
```

```
goto choose;
```

```
switch(choice){
```

```
case '1':
```

```
initial();
```

```
getdata();
```

```
swap();
```

```
dft();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;

    case '2':
        rectan_plot();
        break;

    case '3':
        polar_plot();
        break;

    case '4':
        clrscr();
        exit(1);
    }
    goto choose;
}

void getdata() /* OPEN DATA FILES */
{ FILE *fil;
  char   fname[50];
  int    j;
  float  maxv,max lv,maxh,max lh;
  float  tr,trf[360];

geta: printf("ENTER AMPLITUDE FILE NAME      : ");
      scanf("%s",fname);
      fil=fopen(fname,"rb");
      if (fil == NULL)
      { printf("ERROR IN OPEN FILE\n");
        goto geta;
      }

```

```

getp: printf("ENTER PHASE FILE NAME      : ");

scanf("%s",filename);

fil=fopen(filename,"rb");

if (fil == NULL)

{ printf("ERROR IN OPEN FILE\n");

goto getp;

}

fread(phi,sizeof(float),dat,fil);

getprv: printf("ENTER PROBE VERTICAL FILE NAME      : ");

scanf("%s",filename);

fil=fopen(filename,"rb");

if (fil == NULL)

{ printf("ERROR IN OPEN FILE\n");

goto getprv;

}

fread(trf,sizeof(float),360,fil);

maxv=trf[0];

maxlv=trf[0];

for (j=1;j<360;j++) /*Find max*/

if(trf[j]>maxv)

maxv=trf[j];

for(j=0;j<360;j++) /* ANTILOG FROM dBm TO mV AND NORMALIZE*/

{ trf[j]=trf[j]-maxv;

trf[j]=pow(10,(trf[j]/20));

}

for(j=0;j<91;j++)

probev[j]=trf[90-j];

for(j=0;j<90;j++)

```

```

        probev[j+91]=trf[359-j];

getprh: printf("ENTER PROBE HORIZONTAL FILE NAME : ");

        scanf("%s",filename);

        fil=fopen(filename,"rb");

        if (fil == NULL)

        { printf("ERROR IN OPEN FILE\n");

          goto getprh;

        }

        fread(trf,sizeof(float),360,fil);

        maxh=trf[0];

        maxlh=trf[0];

        for (j=1;j<360;j++)          /*Find max*/

            if(trf[j]>maxh)

                maxh=trf[j];

        for(j=0;j<360;j++)

        {   trf[j]=trf[j]-maxh;

            trf[j]=pow(10,(trf[j]/20));

        }

        for(j=0;j<91;j++)

            probeh[j]=trf[90-j];

        for(j=0;j<90;j++)

            probeh[j+91]=trf[359-j];

    }

void swap()    /* SWAP DATA IN ODD ROW TO BE IN CORRECT POSITION*/

{   float as[54],ps[54];

    int xx,x,y;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (y=1;y<=pty;y=y+2)
{ for (x=0;x<ptx;x++)
  { as[x]=amp[(y*ptx)+ptx-1-x];
    ps[x]=phi[(y*ptx)+ptx-1-x];
  }
for (x=0;x<ptx;x++)
{ xx=y*ptx+x;
  amp[xx]=as[x];
  phi[xx]=ps[x];
}
}

void outdata() /*SAVE TRANSFORM DATA TO FILE*/
{ FILE *fil;
  char fname[50];
  int i;
  float t[181];

  clrscr();
  outh: printf("ENTER HORIZONTAL FILE NAME      : ");
  scanf("%s",fname);
  fil=fopen(fname,"wb");
  if (fil == NULL)
  { printf("ERROR IN OPEN FILE\n");
    goto outh;
  }

  fwrite(outh,sizeof(float),181,fil);

  outh: printf("ENTER VERTICAL FILE NAME      : ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scanf("%s",filename);
fil=fopen(filename,"wb");
if (fil == NULL)
{ printf("ERROR IN OPEN FILE\n");
  goto outv;
}

fwrite(outv,sizeof(float),181,fil);
}

}

void initial() /*INITIAL DATA FOR CALCULATION*/
{ clrscr();
  inital: printf("ENTER FREQUENCY (MHz) : ");
  scanf("%f",&freq);
  printf("NUMBER OF SCAN POINTS IN X AXIS : ");
  scanf("%d",&ptx);
  printf("NUMBER OF SCAN POINTS IN Y AXIS : ");
  scanf("%d",&pty);
  printf("SPACE OF SCANNING IN X AXIS (cm) : ");
  scanf("%f",&dx);
  printf("SPACE OF SCANNING IN Y AXIS (cm) : ");
  scanf("%f",&dy);

  dat=ptx*pty; /*CHECK DIMENSION OF SCANNING AREA*/
  if ((dat > dat1)|| (ptx > 54)|| (pty > 38))
  { printf("\n\n TOO MANY NUMBER OF SCAN POINTS .\n");
    goto inital;
  }

  printf("\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void dft()      /*TRANSFORM TO FAR FIELD PATTERN*/
{
  complex xh,xlh,xv,xlv;
  int i,m,n,z,nn,m1,n1;
  float kz,k0,kyh,kxh,kyv,kxv,kkh,kkv;

  k0=2*PI*freq*1000000/C;    /*WAVENUMBER*/
  printf("K0 = %f\n",k0);
  printf("PRESS ANY KEY");
  getch();
  dx=dx/100;
  dy=dy/100;
  clrscr();
  printf("    * HORIZONTAL(=0°)          VERTICAL(=90°) \n");
  window(1,3,80,25);
  for (i=0;i<181;i++)
  { outv[i]=-900;
    outh[i]=-900;
  }
  for (nn=0;nn<181;nn++)
  { amp[nn]=pow(amp[nn],2);
    phi[nn]=2*phi[nn];
  }
  for (z=-60;z<=60;z++)    /*CALCULATE FROM -60 DEGREE TO 60 DEGREE*/
  { kz=k0*cos(z*PI/180);
    kyh=0;                  /* kx,ky in HORIZONTAL*/
    kxh=k0*dx*sin(z*PI/180);
    kyv=k0*dy*sin(z*PI/180); /* kx,ky in VERTICAL*/
    kxv=k0*dx*cos(z*PI/180);
    kkh=k0*cos(z*PI/180);
    kkv=k0*sin(z*PI/180);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้: **kyv=k0*dy*sin(z*PI/180);** /* kx,ky in VERTICAL*/
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

kxv=0;
x1h.r=0;
x1h.i=0;
x1v.r=0;
x1v.i=0;
for (m=0;m<pty;m++) /* INTEGRATE IN Y AXIS*/
{
  xh.r=0;
  xh.i=0;
  xv.r=0;
  xv.i=0;
  m1=(pty/2)-m;
  for (n=0;n<ptx;n++) /* INTEGRATE IN X AXIS*/
  {
    nn=ptx*m+n;
    n1=(ptx/2)-n;
    kkh=n1*kxh+phi[nn]*PI/180;
    kkv=phi[nn]*PI/180;
    xh.r+=(amp[nn]*cos(kkh));
    xh.i+=(amp[nn]*sin(kkh));
    xv.r+=(amp[nn]*cos(kkv));
    xv.i+=(amp[nn]*sin(kkv));
  }
  x1h.r+=(xh.r*dx*dy);
  x1h.i+=(xh.i*dx*dy);
  x1v.r+=((xv.r*cos(m1*kyv)-xv.i*sin(m1*kyv))*dx*dy);
  x1v.i+=((xv.r*sin(m1*kyv)+xv.i*cos(m1*kyv))*dx*dy);
}
outh[90+z]=sqrt(pow(x1h.r,2)+pow(x1h.i,2)); /*OUTPUT IN HORIZONTAL*/
outv[90+z]=sqrt(pow(x1v.r,2)+pow(x1v.i,2)); /*OUTPUT IN VERTICAL */

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outh[90+z]=outh[90+z]/(kz*probeh[90-z]); /*PROBE CORRECTION*/
outv[90+z]=outv[90+z]/(kz*probev[90+z]);
outh[90+z]=20*log10(outh[90+z]); /*Power Pattern*/
outv[90+z]=20*log10(outv[90+z]); /*Power Pattern*/
  cprintf(" %3d",z);
  cprintf("          %11f",outh[90+z]);
  cprintf("          %11f\n\r",outv[90+z]);
}

```

```

window(1,1,80,25);
clear();
printf("READY!");
printf("\007");
getch();
}

```

```

void color_circle(int x,int y,int rad,int linecolor)
{
    /*DRAW CIRCLE WITH COLOR */
    setcolor(linecolor);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    circle(x,y,rad);
}

void color_line(int x1,int y1,int x2,int y2,int linecolor)
{
    /* DRAW LINE WITH COLOR */
    setcolor(linecolor);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void color_line1(int x1,int y1,int x2,int y2,int linecolor)
{
    /* DRAW THICK LINE WITH COLOR*/
    setcolor(linecolor);
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    line(x1,y1,x2,y2);
}

void rectan_plot() /* PLOT RECTANGULAR GRAPH */
{
    int i,j,k,c;
    float max,min,max1,min1,dmx,dmn,span;
    FILE *fil;
    char str[10],filename[50];
    float d[181];
    int polyarray[]={0,0,640,0,640,480,0,480};
    int gdriver = DETECT, gmode, errorcode;
    c=3;
    clrscr();
    plotr: printf("ENTER FILE NAME TO PLOT RECTANGULAR GRAPH : ");
    scanf("%s",filename);
    fil=fopen(filename,"rb");
    if (fil == NULL)
        { printf("ERROR IN OPEN FILE\n");
          goto plotr;
        }
    fread(d,sizeof(float),181,fil);
    initgraph(&gdriver, &gmode, "c:\\bc\\bgi\\");
    fillpoly(4,polyarray);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

settextjustify(1,2);
color_line(80,7,80,445,c);
color_line(80,445,620,445,c);
j=-90; k=0;
for (i=80;i<=620;i=i+15)
{
    color_line(i,440,i,445,c);
    if (k%3 == 0)
    {
        sprintf(str,"%d",j);
        color_line(i,7,i,450,c);
        outtextxy(i,455,str);
        j=j+15;
    }
    k++;
}
j=0;
for (i=7;i<=445;i=i+73)
{
    color_line(75,i,620,i,c);
    sprintf(str,"%d",j*-10);
    outtextxy(52,i-2,str);
    j++;
}
setcolor(2);
settextstyle(2,HORIZ_DIR,3);
outtextxy(25,0,"dB");
outtextxy(600,470,"Degree");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

max=d[0];
max l=d[0];
min=d[0];
min l=d[0];

dmx=0;
dmn=0;

for (j=0;j<181;j++) /*Find max*/
{
    if(d[j]>max)
    {
        max=d[j];
        dmx=j;
    }
    if(d[j]<min) /*Find min*/
    {
        min=d[j];
        dmn=j;
    }
}

span=60.0;
settextstyle(2,HORIZ_DIR,0); /* SET SPAN TO PLOT */
for (i=0;i<181;i++)
{
    d[i]=d[i]-max;
    if (d[i]<(0-span))
        d[i]=(0-span);
    d[i]=(d[i]+span)*438/span;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    color_line(170,445,170,(int)(445-d[31]),c+1); /* PLOT GRAPH*/
    for (i=30;i<150;i++)
        color_line(3*i+80,(int)(445-d[i]),
            3*(i+1)+80,(int)(445-d[(i+1)]),c+1);

    color_line(530,(int)(445-d[150]),530,445,c+1);

    getch();
    closegraph();
}

void polar_plot() /* PLOT POLAR GRAPH */
{
    float pi,x,y,d[181];
    int i,j,k,l,h,m,n,ans,ans1,e[181],f[181];
    float max,max1,span,min,min1,g;
    FILE *fil;
    char filename[30],msg[80];
    int gdriver = DETECT, gmode, errorcode;
    float o;
    int polyarray[]={0,0,640,0,640,480,0,480};

    clrscr();

    plotp: printf("ENTER FILE NAME TO PLOT POLAR GRAPH : ");
    scanf("%s",filename);
    fil=fopen(filename,"rb");
    if (fil == NULL)
        printf("ERROR IN OPEN FILE\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        goto plotp;
    }

    clrscr();
    initgraph(&gdriver, &gmode, "c:\\bc\\bgi\\");
    fillpoly(4,polyarray);
    color_circle(320,240,230,BLUE);
    color_circle(320,240,190,BLUE);
    color_circle(320,240,150,BLUE);
    color_circle(320,240,110,BLUE);
    color_circle(320,240,70,BLUE);
    color_circle(320,240,30,BLUE);
    color_line(320,10,320,470,GREEN);
    color_line(90,240,550,240,GREEN);
    color_line(483,403,157,77,GREEN);
    color_line(157,403,483,77,GREEN);
    pi=3.141592654;

    fread(d,sizeof(float),181,fil);
    fclose(fil);
    max=d[0];
    maxl=d[0];
    k=0;n=0;
    min=d[0];
    minl=d[0];
    for (j=0;j<181;j++)        /*Find max*/
    {
        if(d[j]>max)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับราชการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

max=d[j];
k=j;
}

if(d[j]<min)
{
min=d[j];
n=j;
}
}

minl=min;
span=max-min;

if(span>60.0) /* SET SPAN RANGE */
{
span=60.0 ;
min=max-60.0;
}

for (j=0;j<181;j++)
{
if((max-d[j])>60)
d[j]=max-60;
}

for (j=92;j<182;j++) /*PLOT GRAPH*/
{
g=(d[(j-1)]-min)+(60.0-span);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y=230*g/60.0*sin((j-1)*pi/180);
```

```
x=230*g/60.0*cos((j-1)*pi/180);
```

```
e[j]=320.0-(int)x;
```

```
f[j]=240.0-(int)y;
```

```
if(j==92)
```

```
continue;
```

```
else
```

```
color_line(e[j-1],f[j-1],e[j],f[j],LIGHTRED);
```

```
}
```

```
color_line(e[181],f[181],320,240,LIGHTRED);
```

```
for (j=1;j<93;j++)
```

```
{
```

```
g=(d[(j-1)]-min)+(60.0-span);
```

```
y=230*g/60.0*sin((j-1)*pi/180);
```

```
x=230*g/60.0*cos((j-1)*pi/180);
```

```
e[j]=320-(int)x;
```

```
f[j]=240-(int)y;
```

```
if(j==1)
```

```
continue;
```

```
else
```

```
color_line(e[j-1],f[j-1],e[j],f[j],LIGHTRED);
```

```
}
```

```
color_line(e[1],f[1],320,240,LIGHTRED);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อผู้จัดทำให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(5);
settextstyle(0,HORIZ_DIR,0);
outtextxy(275,1,"dB RELATIVE");
outtextxy(323,13,"0");
outtextxy(323,53,"-10");
outtextxy(323,94,"-20");
outtextxy(323,135,"-30");
outtextxy(323,176,"-40");
outtextxy(323,204,"-50");
outtextxy(65,236,"-90");
outtextxy(552,236,"+90");
getch();
closegraph();
}

```



บทที่ 3

การทดลอง

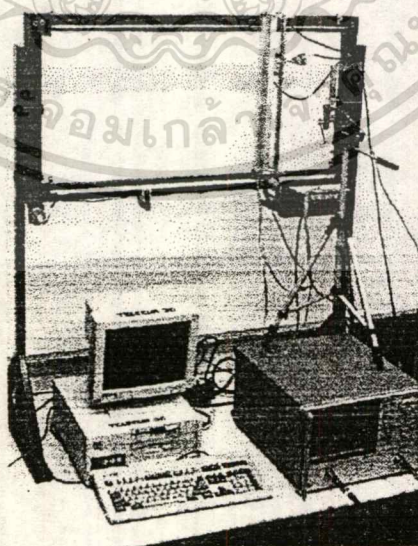
ในการทดสอบ การวัดสนามระยะใกล้แบบระนาบนี้มีจุดมุ่งหมายเพื่อที่จะหารูปแบบการแพร่กระจายคลื่นของสายอากาศแทนการวัดจากสนามระยะไกลโดยตรง ดังนั้นในการทดลองจึงได้ทำการวัดทั้งสองแบบทั้งในสนามระยะใกล้ และสนามระยะไกลเพื่อนำผลที่ได้มาทำการเปรียบเทียบกัน

อุปกรณ์ที่ใช้ในการทดลอง มีดังนี้

- สายอากาศทดสอบ มี 2 แบบ คือ สายอากาศแบบฮอร์น (horn antenna) กับสายอากาศแบบสลิต อาร์เรย์ (slot array)
- โพรบที่ใช้วัด เป็นแบบ โอเพนเอน เวฟไกด์
- เครื่องสแกนเนอร์
- เน็ทเวอร์ค อนุบาลิเซอร์

ขั้นตอนการทดลอง

1.) ทำการคำนวณหาขนาดของอะพอร์เจอร์ของสายอากาศที่จะทำการทดลองทั้งสองแบบและทำการต่ออุปกรณ์ดังรูปที่ 1 โดยใช้สายอากาศทดสอบเป็นตัวส่ง และโพรบเป็นตัวรับสัญญาณ ซึ่งทั้งสองอุปกรณ์นี้จะต่ออยู่กับพอร์ทของเน็ทเวอร์ค อนุบาลิเซอร์ทั้งคู่ โดยสายอากาศทดสอบที่เป็นตัวส่งจะต่ออยู่กับพอร์ทที่ 1 และ โพรบจะต่ออยู่กับพอร์ทที่ 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับที่ 1 การทดลองวัดสนามระยะใกล้แบบระนาบ ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) นำข้อมูลที่วัดได้ไปเข้าโปรแกรมแปลงฟูเรียร์ ทรานฟอร์ม เพื่อหาข้อมูลสนามระยะไกล

5.) นำข้อมูลสนามระยะไกลที่ได้จากการคำนวณมาพล็อตกราฟ

6.) กราฟที่ได้จะเป็นรูปแบบการแพร่กระจายคลื่นของสายอากาศทดสอบชนิดนั้น

ค่ามุม θ ที่ใช้ในการทดสอบสายอากาศแบบสนามระยะไกล ที่ใช้กันทั่วไปมีอยู่สองค่าคือ 60 องศา และ 75 องศา [3]

แต่ค่าของมุม θ ที่ใช้ในการทดลองนี้มีค่าเท่ากับ 60 องศาเนื่องจากขนาดในการสแกนที่มีจำกัดของเครื่องสแกนเนอร์

ค่าระยะความห่างที่ใช้ในการทดสอบสายอากาศแบบสนามระยะไกล ที่ใช้กันจะมีค่าเท่ากับ 3λ (7.5 เซนติเมตร.) โดยจะอยู่ในบริเวณสนามระยะใกล้ (near field) ถ้าเรากำหนดให้มีค่าระยะห่างที่เข้าไปอยู่ในบริเวณสนามรีแอกทีฟหรือบริเวณสนามระยะไกล จะมีผลทำให้การทดสอบสายอากาศมีความผิดพลาดเกิดขึ้น ซึ่งระยะห่างของสนามชนิดต่างๆ ได้อธิบายไว้ในบทที่ 2

(ในการทดสอบสายอากาศแบบ ฮอร์น (horn) ที่ความถี่ 12 GHz จะมีบริเวณสนามรีแอกทีฟที่ระยะต่ำกว่า 5.76 เซนติเมตร ถัดมาจะเป็นบริเวณสนามระยะไกลจนถึงระยะ 28.8 เซนติเมตร และถัดไปจะเป็นบริเวณสนามระยะไกล, ส่วนในการทดสอบสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ที่ความถี่ 12 GHz จะมีบริเวณสนามรีแอกทีฟที่ระยะต่ำกว่า 6.5 เซนติเมตร ถัดมาจะเป็นบริเวณสนามระยะไกลจนถึงระยะ 33.8 เซนติเมตร และถัดไปจะเป็นบริเวณสนามระยะไกล)

สำหรับการทดสอบสายอากาศแบบสนามระยะไกลเพื่อนำมาเปรียบเทียบกับผลของการทดสอบสายอากาศแบบสนามระยะไกล จะขอกกล่าวโดยสรุปได้ดังนี้

- โดยทำการวัดในระยะฟาร์ฟิลด์ที่ระยะมากกว่า $2D^2/\lambda$ ในที่โล่ง(แดดฟ้าของอาคารเรียน) และตั้งวางสายอากาศทดสอบไว้บนแท่นหมุน (turn table) เพื่อสามารถหมุนสายอากาศทดสอบให้ได้ครบ 360 องศา ทั้งในแนวตั้งและแนวนอน และใช้สายอากาศตัวส่งให้ส่งสัญญาณความถี่ 12 GHz ไปยังสายอากาศทดสอบ เพื่อทำการวัดความแรงของสัญญาณในแต่ละองศาที่แท่นหมุน หมุนไป หลังจากนั้นจึงนำข้อมูลที่ได้มาทำการพล็อตกราฟ เพื่อหารูปแบบการแพร่กระจายคลื่นของสายอากาศทดสอบในสนามระยะไกลต่อไป

ผลการทดลอง

1.สายอากาศแบบ สล็อต อาร์เรย์ (slot array)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เนื่องจากสายอากาศแบบ สล็อต อาร์เรย์ (slot array) มีขนาดความกว้าง 1.8 เซนติเมตร และมีขนาดความยาว 6.5 เซนติเมตร ดังนั้นขนาดของการสแกนในแนวแกน x และขนาดของการสแกนในแนวแกน y จะมีค่าดังนี้

$$\begin{aligned} \text{ขนาดของการสแกนในแนวแกน } x &= D + 2Z \tan \theta \\ &= 1.8 + 2(3 * 2.5) \tan 60 \\ &= 27.78 \text{ เซนติเมตร} \end{aligned}$$

$$\begin{aligned} \text{ขนาดของการสแกนในแนวแกน } y &= D + 2Z \tan \theta \\ &= 6.5 + 2(3 + 2.5) \tan 60 \\ &= 32.48 \text{ เซนติเมตร} \end{aligned}$$

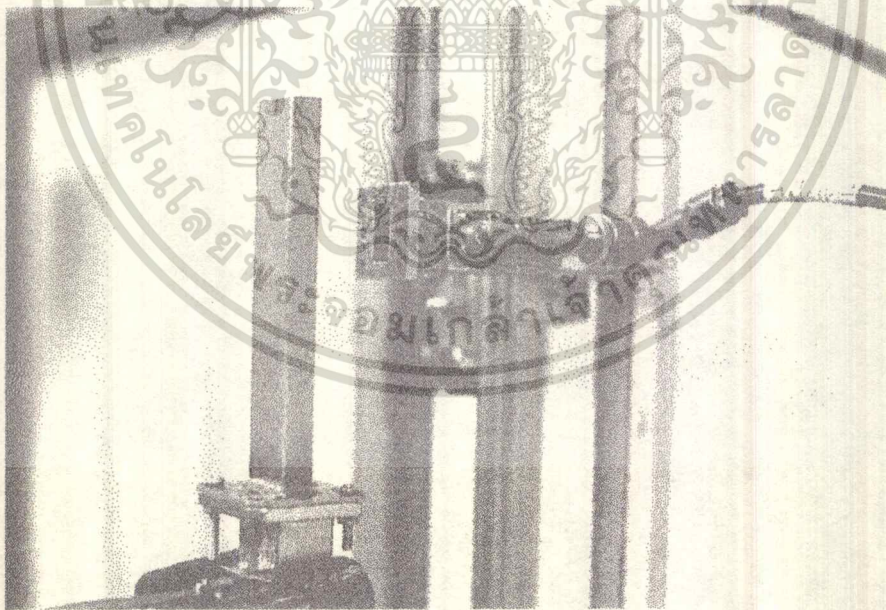
ดังนั้น จะได้จำนวนจุดของการสแกนในแนวแกน x และแกน y โดยนำค่าระยะห่างในการสแกน ($\lambda/2 = 1.25$ เซนติเมตร) ไปหารดังนี้

$$\text{จำนวนจุดของการสแกนในแนวแกน } x = 24 \text{ จุด}$$

$$\text{จำนวนจุดของการสแกนในแนวแกน } y = 27 \text{ จุด}$$

$$\text{ดังนั้นจำนวนจุดทั้งหมดของการสแกน} = 648 \text{ จุด}$$

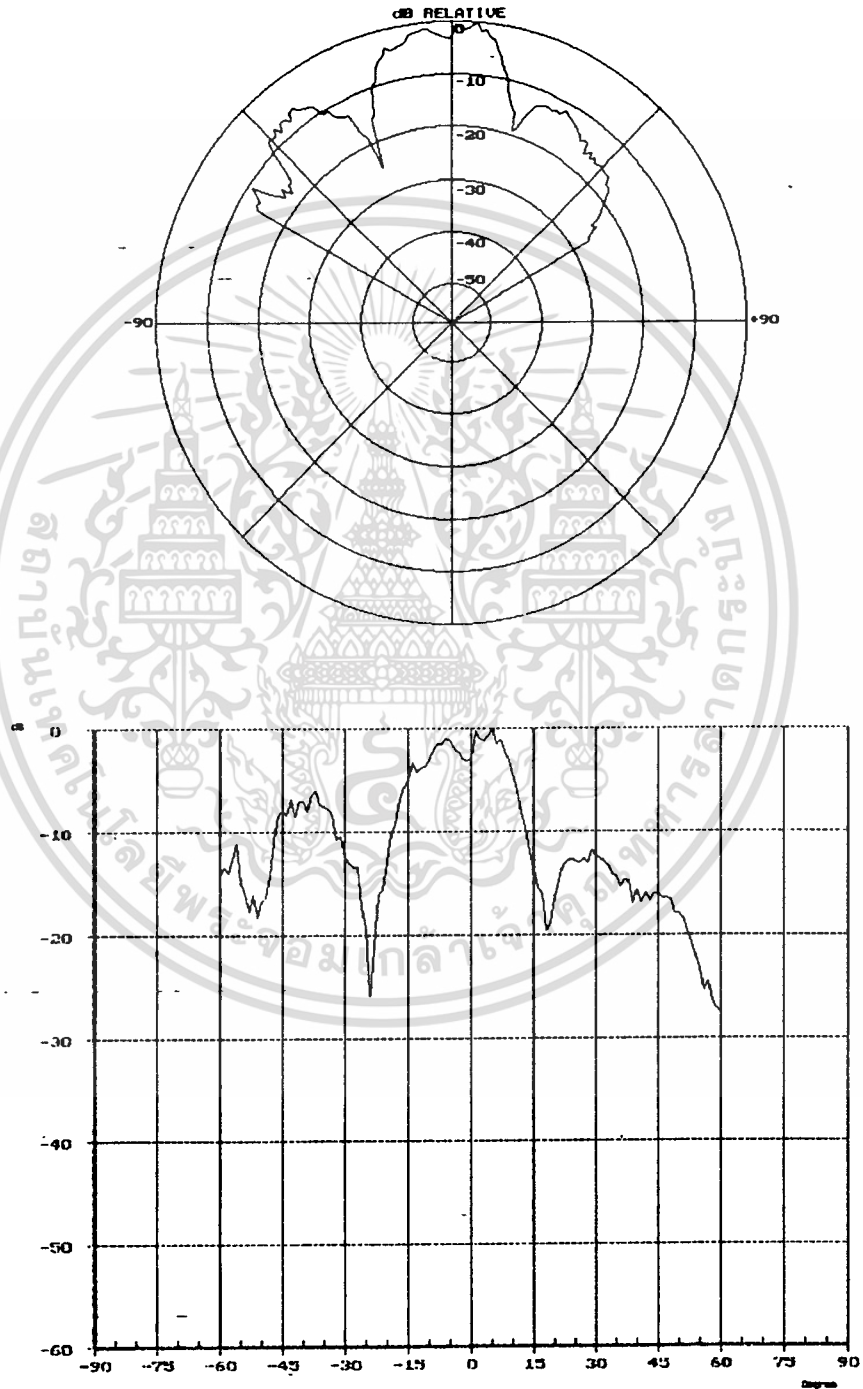
เมื่อทำการติดตั้งตามการคำนวณจะได้รูปดังรูปที่ 4



รูปที่ 4 แสดงการติดตั้งสายอากาศแบบสล็อต อาร์เรย์

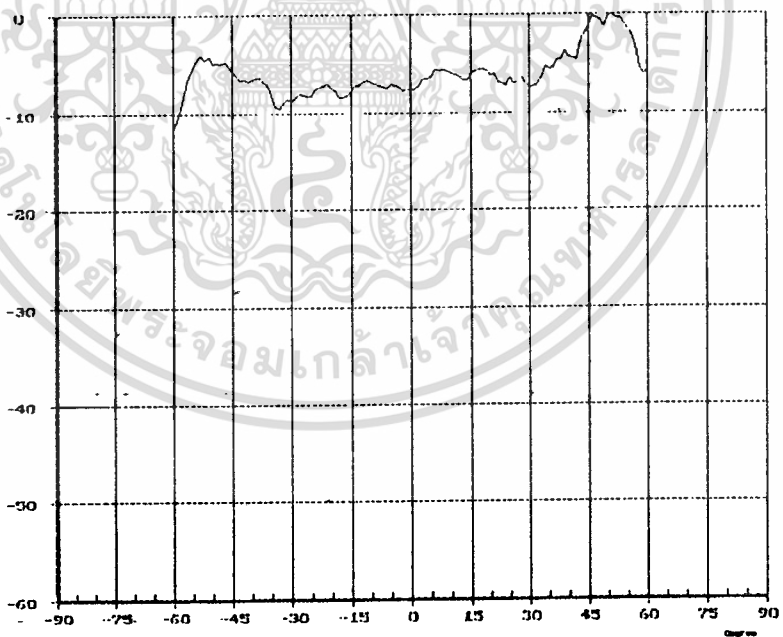
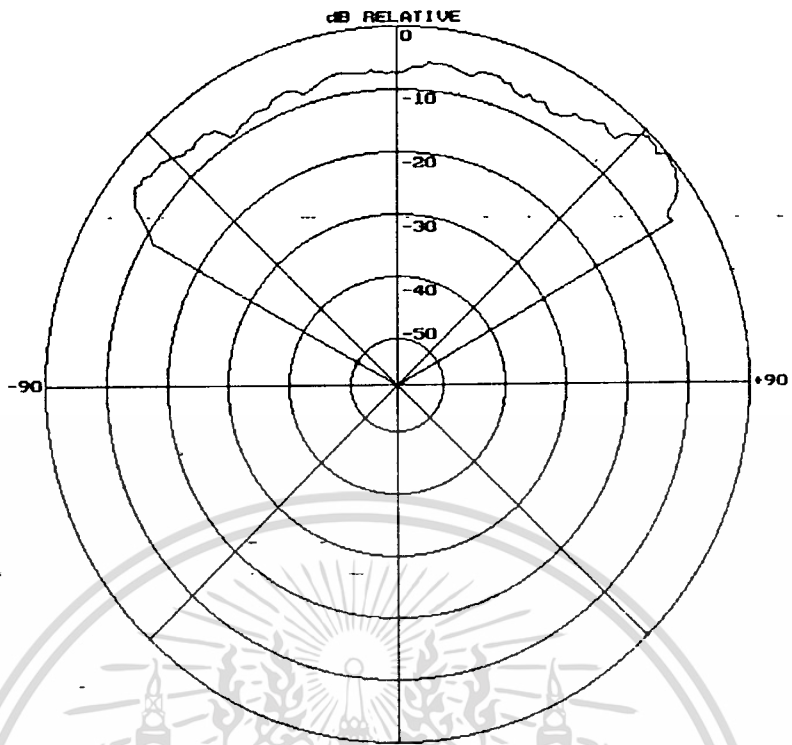
เมื่อทำการทดลองเสร็จแล้ว แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนวแกนตั้งและแนวนอน โดยใช้วิธีการวัดแบบสนามระยะใกล้จะเป็นดังรูปที่ 5 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และรูปที่ 6 ตามลำดับโดยเราจะนำมาเปรียบเทียบกับ แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนวแกนตั้งและแนวนอน โดยใช้วิธีการวัดแบบสนามระยะไกลจะเป็นดังรูปที่ 7 และ 8 ตามลำดับ เมื่อเรานำมาเปรียบเทียบกันแล้วจะเห็นได้ว่า คล้ายคลึงกัน



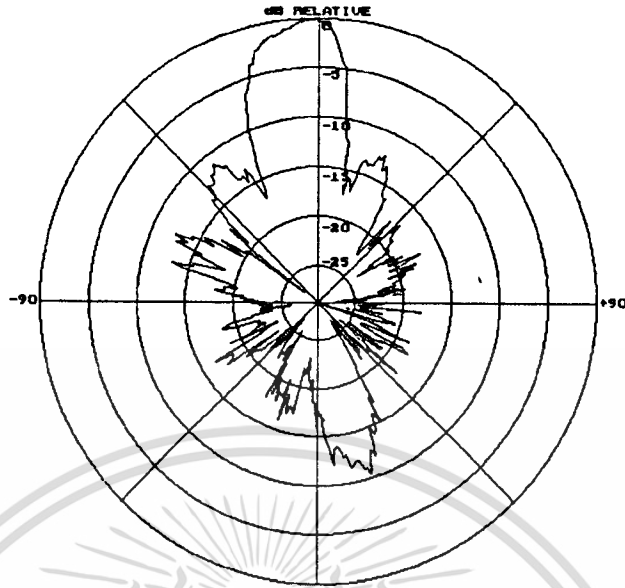
รูปที่ 5 แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนว

เอกสารนี้เป็นเอกสารที่อนุญาตให้ส่วนหนึ่งหรือทั้งหมดสามารถใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายและไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 แกนตั้งโดยใช้วิธีการวัดแบบสนามระยะไกล
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

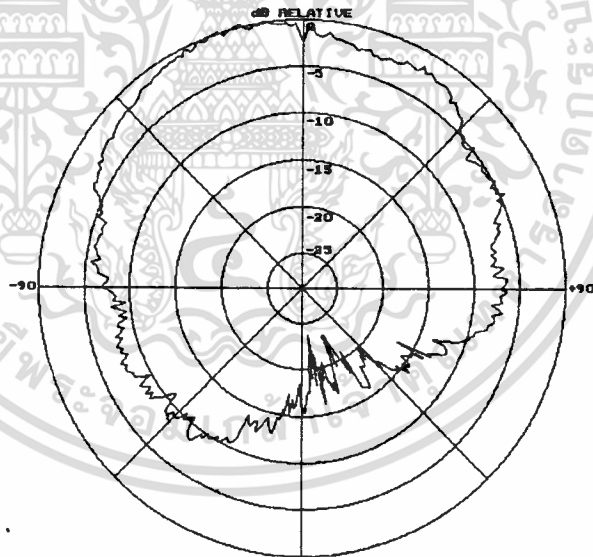


รูปที่ 6 แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนวแกนนอนโดยใช้วิธีการวัดแบบสนามระยะใกล้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนวแกนตั้งโดยใช้วิธีการวัดแบบสนามระยะไกล



รูปที่ 8 แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์ (slot array) ในแนวแกนนอนโดยใช้วิธีการวัดแบบสนามระยะไกล

2. สายอากาศแบบ ฮอร์น (horn)

-เนื่องจากสายอากาศแบบ ฮอร์น (horn) มีขนาดความกว้าง 6 เซนติเมตร และมีขนาดความยาว 6 เซนติเมตร ดังนั้นขนาดของการสแกนในแนวแกน x และขนาดของการสแกนในแนวแกน y จะมีค่าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเอาไว้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ขนาดของการสแกนในแนวแกน $x = D + 2Z \tan \theta$
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= 6+2(3*2.5)\tan 60$$

$$= 31.98 \text{ เซนติเมตร}$$

ขนาดของการสแกนในแนวแกน $y = D + 2Z\tan\theta$

$$= 6+2(3+2.5)\tan 60$$

$$= 31.98 \text{ เซนติเมตร}$$

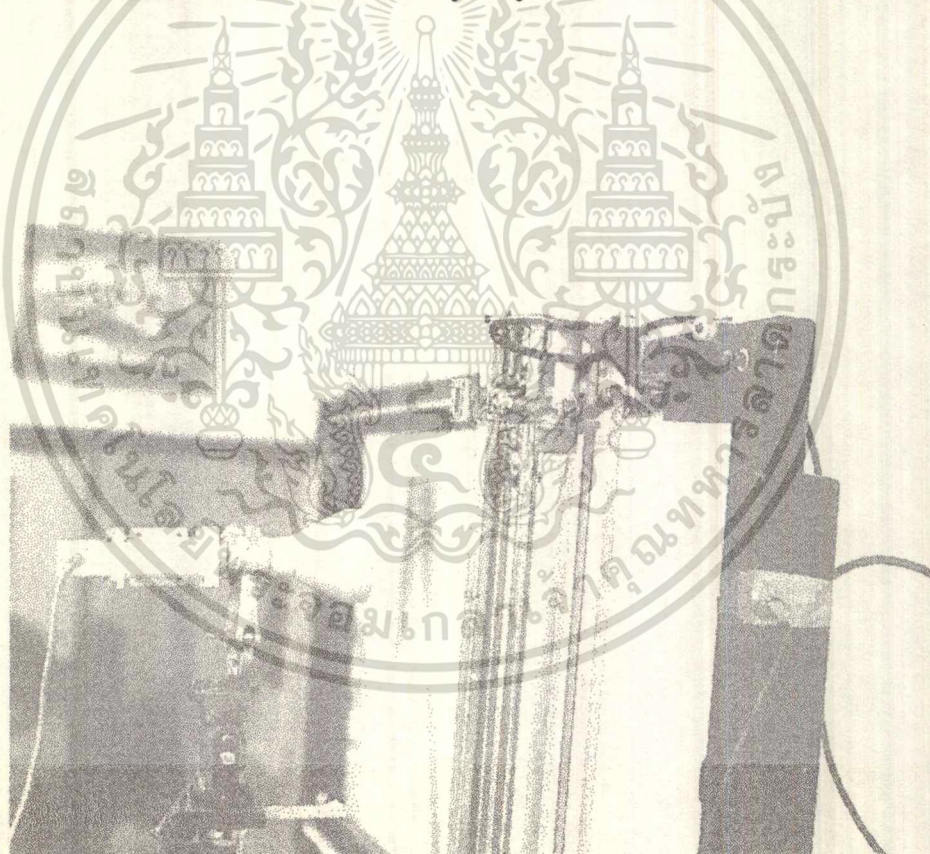
ดังนั้น จะได้จำนวนจุดของการสแกนในแนวแกน x และแกน y โดยนำค่าระยะห่างในการสแกน ($\lambda/2 = 1.25$ เซนติเมตร) ไปหารดังนี้

จำนวนจุดของการสแกนในแนวแกน $x = 27$ จุด

จำนวนจุดของการสแกนในแนวแกน $y = 27$ จุด

ดังนั้นจำนวนจุดทั้งหมดของการสแกน = 729 จุด

เมื่อทำการติดตั้งตามการคำนวณจะได้รูปดังรูปที่ 9

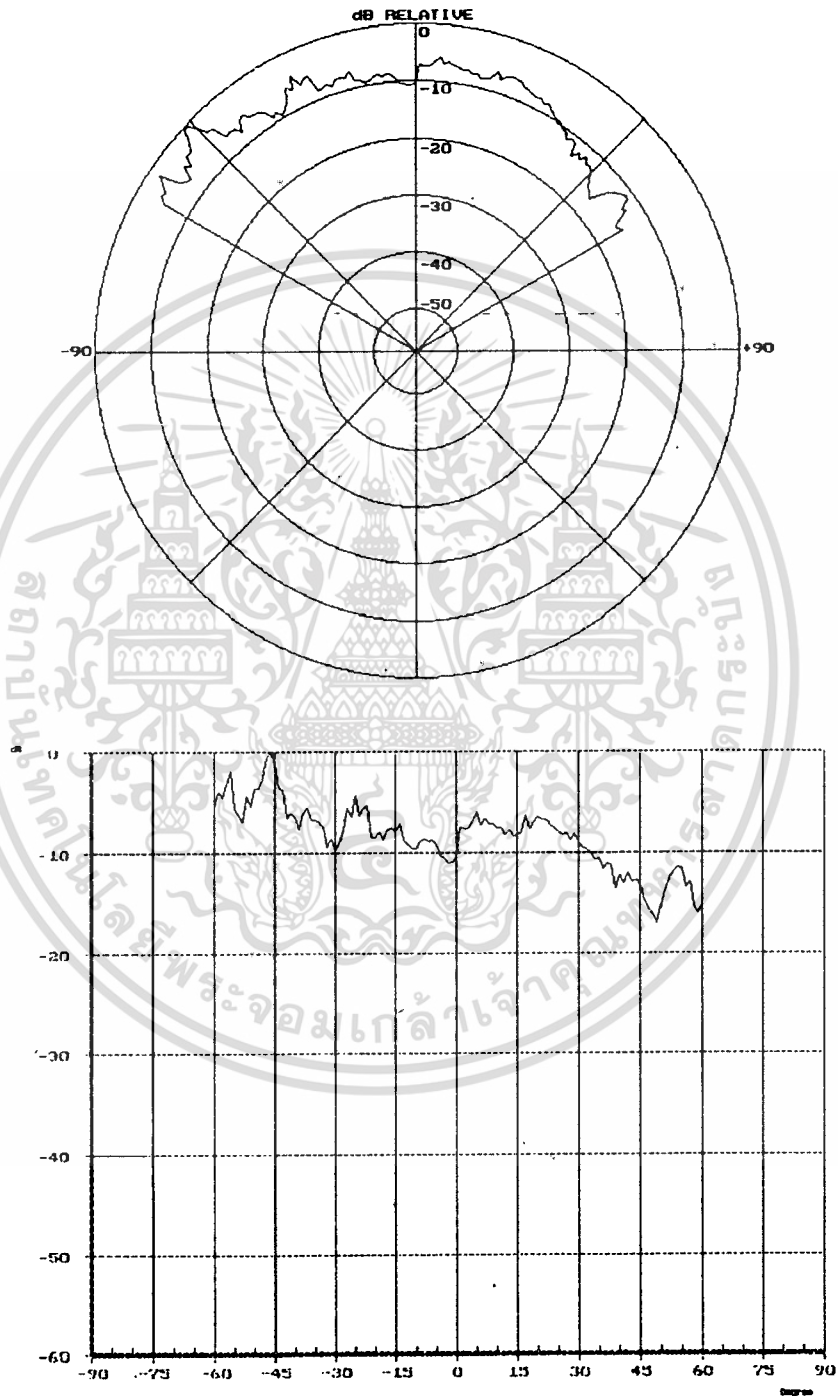


รูปที่ 9 แสดงการติดตั้งสายอากาศแบบ ฮอ์น (horn)

เมื่อทำการทดลองเสร็จแล้ว แพทเทิร์นของสายอากาศแบบ ฮอ์น (horn) ในแนวแกนตั้งและแนวนอนโดยใช้วิธีการวัดแบบสนามระยะใกล้จะเป็นดังรูปที่ 10 และรูปที่ 11 ตามลำดับ โดยเราจะนำมาเปรียบเทียบกับ แพทเทิร์นของสายอากาศแบบ สล็อต อาร์เรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(slot array) ในแนวแกนตั้งและแนวนอน โดยใช้วิธีการวัดแบบสนามระยะไกลจะเป็นดังรูปที่ 12 และ 13 ตามลำดับ เมื่อนำมาเปรียบเทียบกันแล้วจะเห็นได้ว่าคล้ายคลึงกัน

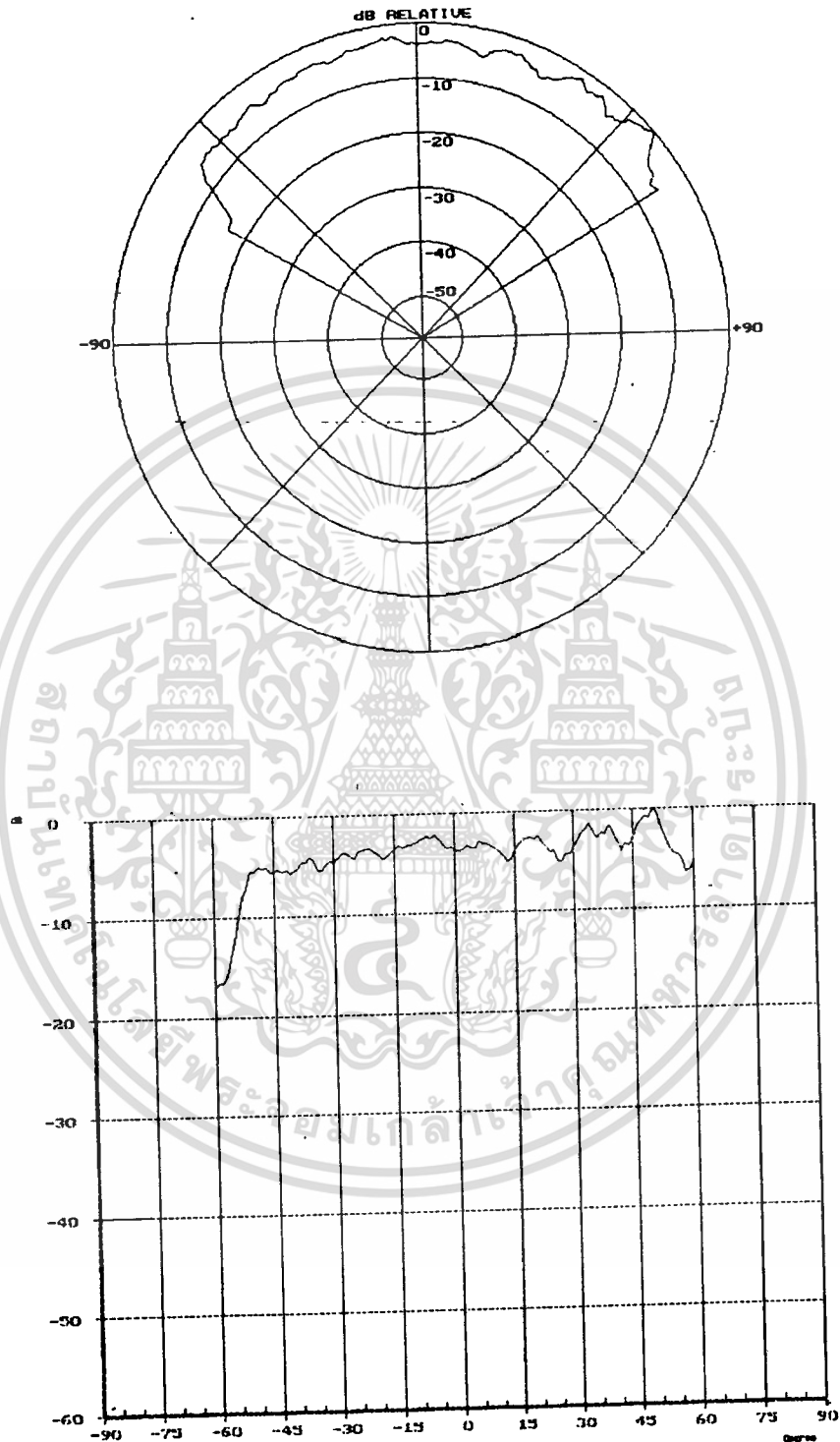


รูปที่ 10 แพทเทิร์นของสายอากาศอากาศแบบ ฮอร์น (horn) ในแนวแกนตั้งโดย

ใช้วิธีการวัดแบบสนามระยะใกล้

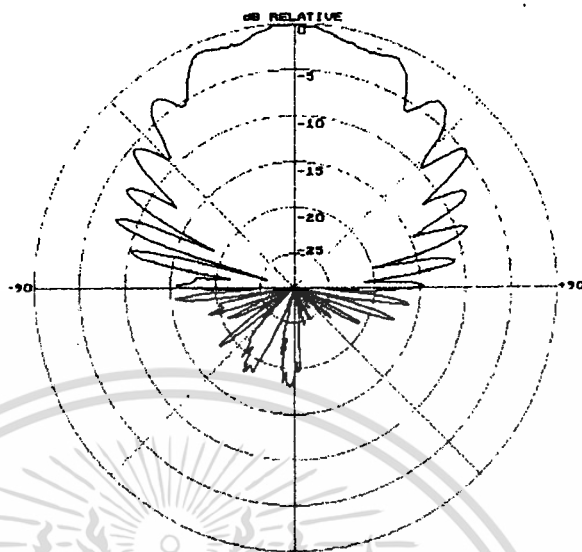
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

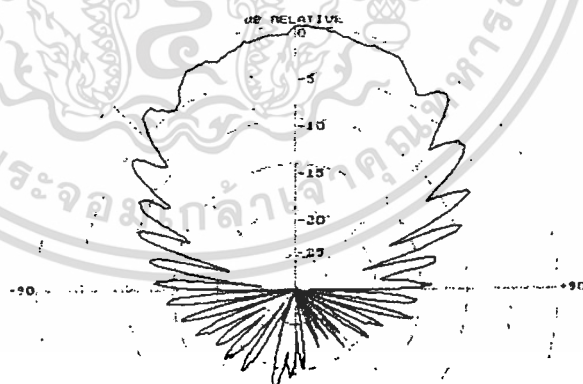


รูปที่ 11 แพทเทิร์นของสายอากาศอากาศแบบ ฮอร์น (horn) ในแนวแกนอน
โดยใช้วิธีการวัดแบบสนามระยะใกล้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 แพทเทิร์นของสายอากาศอากาศแบบ ฮอร์น (horn) ในแนวแกนตั้งโดย
ใช้วิธีการวัดแบบสนามระยะไกล



รูปที่ 13 แพทเทิร์นของสายอากาศอากาศแบบ ฮอร์น (horn) ในแนวแกนนอน
โดยใช้วิธีการวัดแบบสนามระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สรุปและวิจารณ์ผลการทดลอง

จากการเปรียบเทียบเพทเทิร์นของสายอากาศโดยใช้การทดสอบสายอากาศแบบสนามระยะใกล้กับการทดสอบสายอากาศแบบสนามระยะไกลจะเห็นได้ว่ารูปเพทเทิร์นที่ได้แตกต่างกัน เนื่องจากการทดสอบสายอากาศแบบสนามระยะใกล้มีข้อจำกัดในการกวาดของมุม θ ปกติจะใช้มุม θ ในการทดสอบสายอากาศเท่ากับ 60 องศาหรือ 75 องศาซึ่งในการทดสอบนี้ใช้มุม θ เท่ากับ 60 องศาเพราะเครื่องสแกนเนอร์มีขนาดของการสแกนในแนวแกน X และแกน Y ที่จำกัด จึงไม่สามารถใช้ค่ามุม เท่ากับ 75 องศา

ในทางทฤษฎีถ้าจะให้การทดสอบสายอากาศแบบสนามระยะใกล้เกิดความผิดพลาดน้อยที่สุดจะต้องใช้มุม θ ในการกวาด 90 องศา (จำนวนจุดในแนวแกน X และ แกน Y มีจำนวนเป็นอนันต์) ซึ่งเป็นไปไม่ได้ในทางปฏิบัติดังนั้นจึงมีความแตกต่างของเพทเทิร์นที่ได้จากการทดสอบสายอากาศทั้งสองแบบ

ข้อผิดพลาดในการทดสอบสายอากาศและแนวทางการพัฒนา

- ความไม่เที่ยงตรงในการเคลื่อนที่ของสแกนเนอร์ ในการเคลื่อนที่แต่ละสเต็ปโดยอาจปรับปรุงกลไกในการขับเคลื่อนให้เที่ยงตรงมากขึ้น
- ขนาดของการสแกนสูงสุด ทั้งทางด้านแกน X และแกน Y มีค่าน้อยเกินไปซึ่งมีผลทำให้ไม่สามารถใช้มุม เท่ากับ 75 องศาได้ซึ่งอาจปรับปรุงโดยเพิ่มขนาดของสแกนเนอร์ให้ใหญ่ขึ้น
- การสะท้อนของคลื่นภายในห้องทดลอง ทำให้รบกวนการทดสอบ ซึ่งสามารถลดการรบกวนลงได้โดยการติดตั้งอุปกรณ์ดูดซับคลื่น
- ซอฟต์แวร์ที่ใช้มีความยุ่งยากพอสมควร ซึ่งอาจปรับปรุงโดยการจัดหน้าจอให้ใช้งานได้ง่ายขึ้น

บทแทรกที่ 1

มาตรฐานการเชื่อมต่อ IEEE-488 (GPIB)

โครงสร้างของ IEEE-488

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์ คือ ผู้ส่ง (Talker), ผู้รับ (Listener) และผู้ควบคุม (Controller)

-Talker ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัว แต่จะมีเพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่

-Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัว เช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลายๆตัวนั่นคือ ในระบบที่กำลังทำงาน จะมี Talker 1 ตัว แต่จะมี Listener หลายตัวได้

-Controller ทำหน้าที่ควบคุมอุปกรณ์ต่างๆในระบบ โดยกำหนดให้ Talker ทำการส่งข้อมูล หรือกำหนดให้ Listener ทำการรับข้อมูล

อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งตามหน้าที่ได้ดังนี้

1. ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น
2. ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์ (Printer) เป็นต้น
3. ทำหน้าที่เป็นทั้ง Talker และ Listener เช่น คอมพิวเตอร์ , เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น
4. ทำหน้าที่เป็น Talker , Listener และ Controller ในตัวเดียวกัน เช่น คอมพิวเตอร์ ที่ ทำหน้าที่ควบคุมระบบ

ขีดจำกัดของ IEEE-488

1. จำนวนอุปกรณ์ในระบบ (Talker,Listener,Controller) ที่ต่อกับสายสัญญาณ 1 เส้น จะต้องมีไม่เกิน 15 เครื่อง
2. สายเคเบิลที่ต่อระหว่างอุปกรณ์ จะต้องยาวไม่เกิน 4 เมตร และความยาวรวมของสายเคเบิลในระบบจะต้องไม่เกิน 20 เมตร
3. ความเร็วในการส่งข้อมูลจะต้องไม่เกิน 1Mb / Sec (1 ล้านบิตต่อวินาที)
4. ต้องมีการจ่ายไฟให้กับอุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับ IEEE-488

ลักษณะทางกายภาพของ IEEE-488 นั้นคือ เป็นสายสัญญาณแบบ 24 เส้นขนานกัน และมีหัวต่ออยู่ปลายทางทั้งสองสาย เพื่อต่อกับอุปกรณ์ หรือต่อกัน เพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น ในจำนวนสายสัญญาณ 24 เส้นนี้มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนที่เหลืออีก 8 เส้นทำหน้าที่ กราวนด์ (ground) และ ชีลด์ (shield) โดยจำนวนสายที่ใช้นำสัญญาณ 16 เส้น นั้นยังแบ่ง ได้เป็น 3 ประเภท ตามรูปที่ 1 คือ

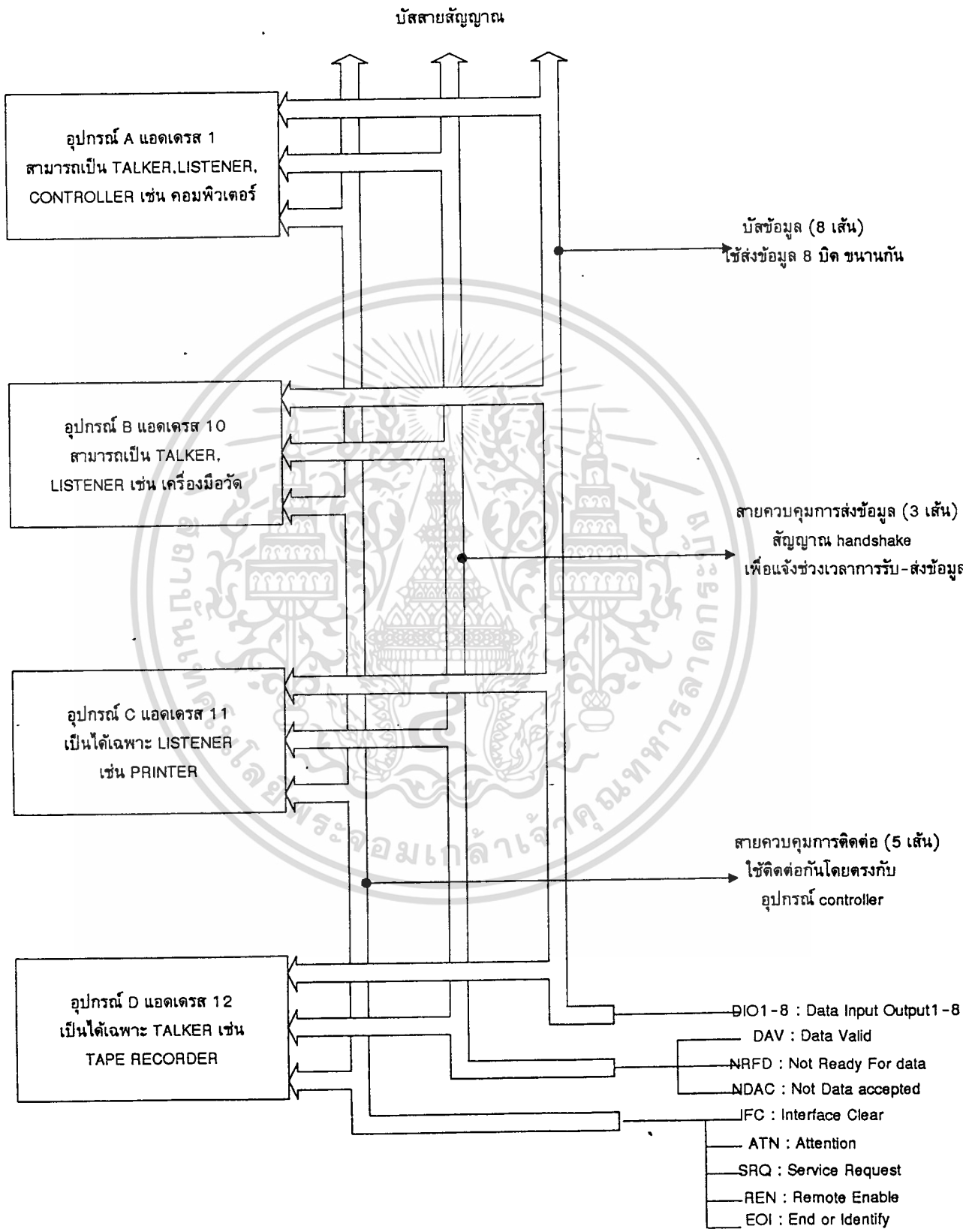
1. บัสข้อมูล (Data Bus) จำนวน 8 สาย คือ
DIO1-DIO8
2. สายสัญญาณควบคุม (Control Line) จำนวน 5 สาย คือ
IFC (Interface Clear)
ANT (Attention)
SRQ (Service request)
REN (Remote Enable)
EOI (End or Identify)
3. สายแฮนด์เชก (Hand Shake)
DAV (Daata Valid)
NRFD (Not Ready For Data)
NDAC (Not Data Accepted)

ความหมายของสัญญาณต่างๆภายใน IEEE-488

ดังที่ได้กล่าวมาแล้วว่าสายสัญญาณต่างๆใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้จะอธิบายความหมายของสัญญาณต่างๆดังนี้

กลุ่มสัญญาณข้อมูล

1. DIO1-DIO8 (Data Input/Output) สายสัญญาณทั้ง 8 เส้นนี้ทำ หน้าที่เป็นทางผ่านของข้อมูลในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

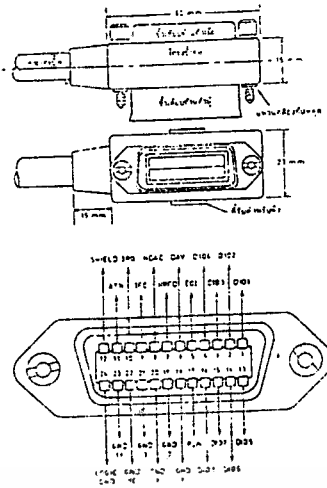
กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interface)

1. IFC (Interface Clear) เป็นสัญญาณรีเซ็ต หรือเคลียร์ระบบ กำเนิดได้โดยตัวควบคุม (Controller) เท่านั้นเมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้ จะกลับคืนสู่สถานะเริ่มต้นใหม่ ซึ่งเป็นสถานะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนแรกเปิดสวิตช์
2. ATN (Attention) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ที่เป็นตัวควบคุม เช่นเดียวกัน ใช้ในการสั่งให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป
3. SQR (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่างๆ เพื่อเป็นการ บอกแก่ระบบว่าขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อกับตัวควบคุม
4. REN (Remote Enable) เป็นสัญญาณที่ถูกส่งมาจากตัวควบคุมเพียง ตัวเดียวเท่านั้น เพื่อใช้สั่งให้อุปกรณ์ต่างๆเปลี่ยนจากโหมดที่ใช้งานปกติมาเป็นการควบคุมโดยตัวควบคุมแทน
5. EOI (End or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม (Controller) หรืออุปกรณ์ที่เป็นตัวส่ง (Talker) ก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้น ได้เสร็จสิ้นลงแล้ว

กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูล

1. DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก “Low” โดย อุปกรณ์ ที่เป็นตัวส่ง(Talker) เป็นการแจ้งแก่ระบบบัสว่าขณะนี้ตัวส่งได้ทำการ ส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว
 2. NRFD (Not Ready For Data) เมื่อสัญญาณนี้มีลอจิกเป็น “Low” จะ เป็นการแสดงว่าในขณะนี้ระบบบัสยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบ ยังพร้อมไม่หมดทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่เป็น “Hi” จนกว่าอุปกรณ์ทุกตัว ให้ลอจิกที่เป็น “Hi” ครบถ้วนแล้ว สัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ในระบบ มีความเร็วแตกต่างกัน
 3. NDAC (Not Data Accepted) สัญญาณเส้นนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ (Listener) โดยสัญญาณนี้จะมีลอจิกเป็น “Low” ใน ขณะ ที่อุปกรณ์ที่เป็นตัวรับกำลังเก็บข้อมูลจากสายข้อมูล (Data Bus) และจะเป็น “Hi” เมื่ออุปกรณ์นั้น ได้ทำการอ่านข้อมูลเรียบร้อยแล้ว
- โดยสัญญาณลอจิกที่ใช้ใน DATA BUS(D1-D8) ของ IEEE-488 BUS นี้มีลักษณะเป็นคอมพลิเมนต์ทั้งหมด คือ “1” เท่ากับ “Low” และ “0” เท่ากับ “Hi” ซึ่ง ตรงข้ามกับในวงจรลอจิกที่เราคุ้นเคย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

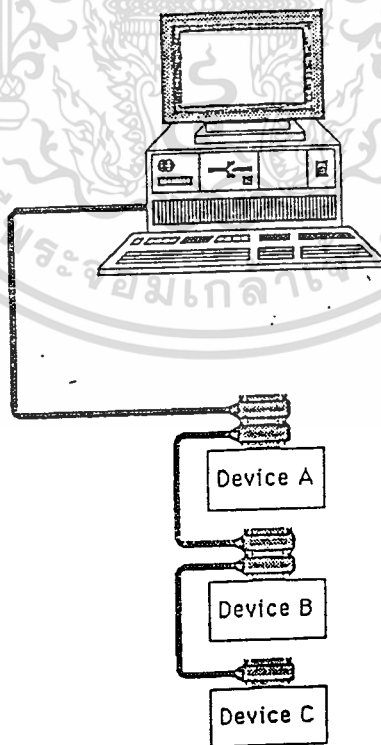


รูปที่ 2 ขั้วต่อของ GPIB และการจัดขาของสัญญาณ

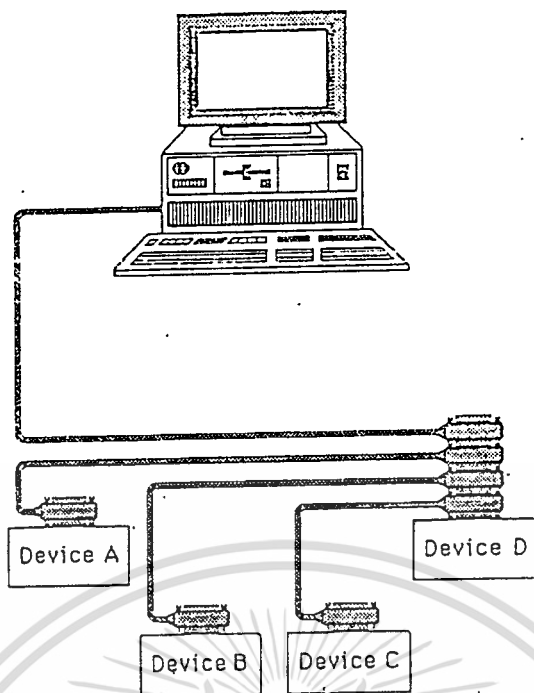
การเชื่อมต่ออุปกรณ์ต่างๆในระบบ IEEE-488 BUS

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่างๆในระบบ IEEE-488 Bus นั้น มีอยู่ 2 วิธีคือ

1. การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)
2. การเชื่อมต่อแบบกระจาย (Star Configuration)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3 การเชื่อมต่อแบบเรียงต่อเนื่องกัน
 ใ้ใช้ประโยชน์ในเชิงพาณิชย์โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 การเชื่อมต่อแบบกระจาย

ขบวนการแฮนด์เชก (Handshake Procedure)

เมื่อมีการเชื่อมต่อระหว่างอุปกรณ์ต่างๆในระบบ ค้างนั้นเมื่อระบบจะทำงาน จึงต้องมีการตรวจสอบเสียก่อน ซึ่งการตรวจสอบนี้เราเรียกว่าขบวนการแฮนด์เชก (Handshake Procedure)



รูปที่ 5

พิจารณาระบบที่ไม่จับซ้อนกัน โดยกำหนดให้มีตัวส่ง (Talker) และตัวรับ (Listener) อย่างละ 1 ตัว

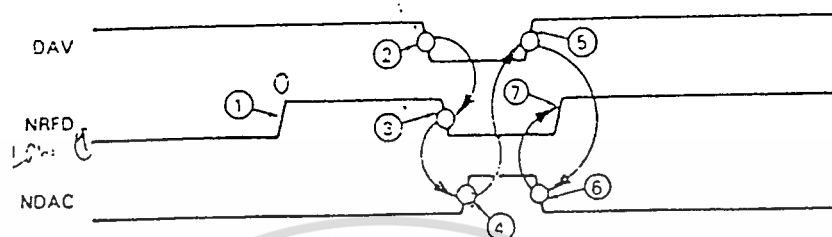
การที่จะมีการสื่อสารระหว่างตัวส่งและตัวรับจะกระทำได้โดยการที่ตัวส่งจะ ทำหน้าที่ส่งสัญญาณออกมาเพื่อให้ตัวรับทราบว่า มีข้อมูลในสายสัญญาณ และตัวรับ จะทำการเก็บข้อมูลจากสายสัญญาณ เมื่อเก็บข้อมูลเสร็จเรียบร้อยแล้ว ตัวรับก็ จะส่งสัญญาณเพื่อให้ตัวส่งทราบว่า ได้ทำการรับข้อมูลเป็นที่เรียบร้อยแล้ว

การส่งสัญญาณในการสื่อสารระหว่างตัวส่งและตัวรับนั้น จะส่งมาทางสาย สัญญาณ 3 สาย ซึ่งสายสัญญาณดังกล่าวนี้เป็นสายสัญญาณในกลุ่มควบคุมการ รับ-ส่งข้อมูล คือ NRFD

(Not Ready For Data), DAV (Data Valid) และ NDAC (Not Data Accepted)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DAV เป็นสัญญาณที่ถูกส่งโดยตัวส่ง NRFD และ NDAC เป็นสัญญาณที่ถูกส่งโดยตัวรับ โดยตัวรับจะใช้สัญญาณ NDAC เพื่อชี้ให้เห็นว่า ในขณะที่นั้นพร้อม หรือไม่พร้อมในการเก็บข้อมูล และ จะใช้สัญญาณ NRFD เพื่อชี้ให้เห็นว่าพร้อม หรือไม่พร้อมที่จะรับข้อมูล

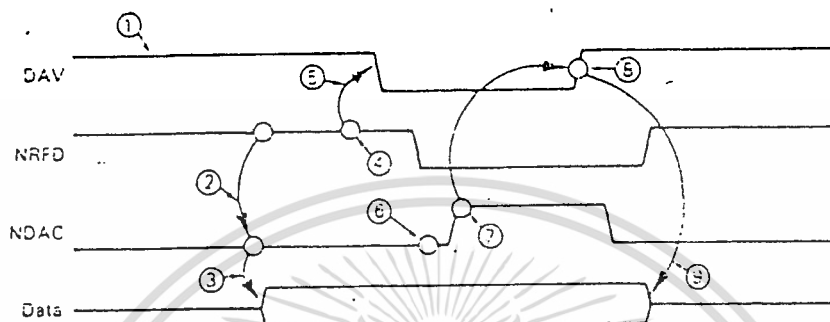


NOTE. The timing shown is relative

รูปที่ 6

ขบวนการแฮนด์เชกเริ่มขึ้นเมื่อ ตัวส่งมีข้อมูลที่จะถ่ายทอดลงสายสัญญาณ ไปยังตัวรับ เริ่มด้วยการที่ตัวรับจะทำให้สัญญาณ NRFD เป็น “Hi” (มีค่าเป็น 0) นั่นคือการบอกให้ทราบว่าตัวรับพร้อมที่จะให้ตัวส่งทำการถ่ายข้อมูลลงในสาย สัญญาณแล้ว (ในจุดที่ 1) ตัวส่งจึงทำการถ่ายข้อมูลลงในสายสัญญาณ DIO1-DIO8 หลังจากทำการรอเวลาเพื่อให้ตัวส่งถ่ายข้อมูลแล้ว ตัวส่งจะทำให้สัญญาณ DAV มีลอจิกเป็น “Low” (มีค่าเป็น 1) เพื่อที่จะบอกให้ทราบว่าขณะนี้ได้มีข้อมูล ในสายสัญญาณทั้ง 8 เส้นแล้ว (ในจุดที่ 2) ต่อมาเมื่อผู้รับทราบว่า มีข้อมูลอยู่ในบัล ก็จะทำให้สัญญาณ NRFD ให้มีค่าเป็น “Low” เพื่อบอกให้ตัวส่งทราบว่ายังไม่พร้อม ที่จะให้ตัวส่งทำการส่งข้อมูลชุดต่อไป และตัวรับพร้อมที่จะเก็บข้อมูลจาก สายสัญญาณ (ในจุดที่ 3) หลังจากตัวรับทำการเก็บข้อมูลจากสายสัญญาณเสร็จ แล้วตัวรับจะทำให้สัญญาณ NDAC ให้มีสถานะเป็น “Hi” ก็จะทำให้สัญญาณ DAV ให้มี ลอจิกเป็น “Hi” ด้วยเพื่อไม่ให้ตัวรับทำการรับข้อมูลในบัลอีก (ในจุดที่ 5) เมื่อตัวรับทราบว่าสัญญาณ DAV มีลอจิกเป็น “Hi” ก็จะทำให้ สัญญาณ NDAC ให้เป็น “Low” ทำให้ข้อมูลในบัลถูกกำจัดออกไป หลังจากนั้น ก็จะทำให้สัญญาณ NRFD ให้ เป็น “Hi” เพื่อบอกให้ทราบว่าพร้อมที่จะรับข้อมูล ชุดต่อไปแล้ว ขบวนการแฮนด์เชกจึงเสร็จสิ้นลงและจะเริ่มขึ้นใหม่เมื่อมีสัญญาณชุดใหม่เข้ามา

ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวส่ง

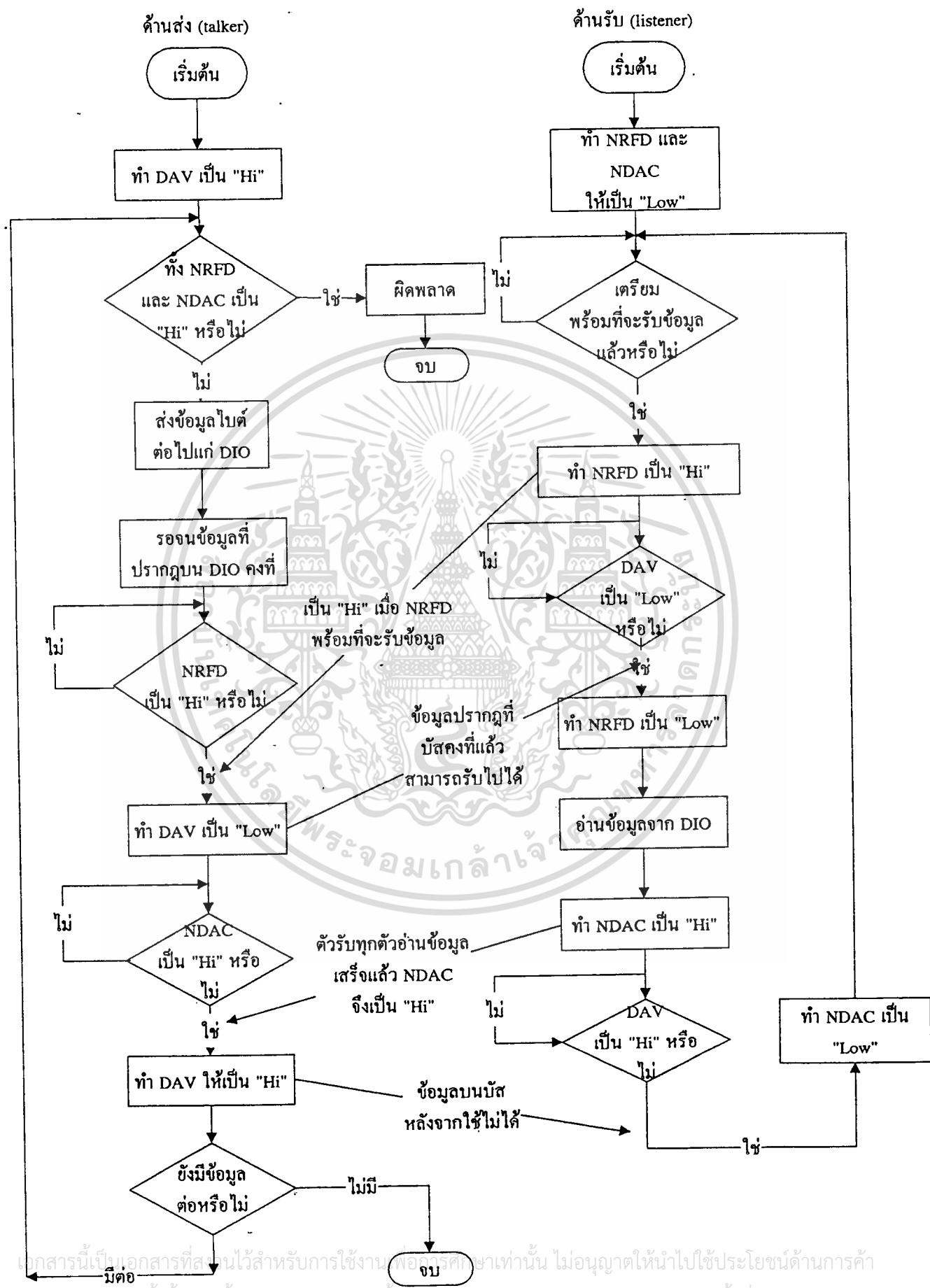


รูปที่ 7

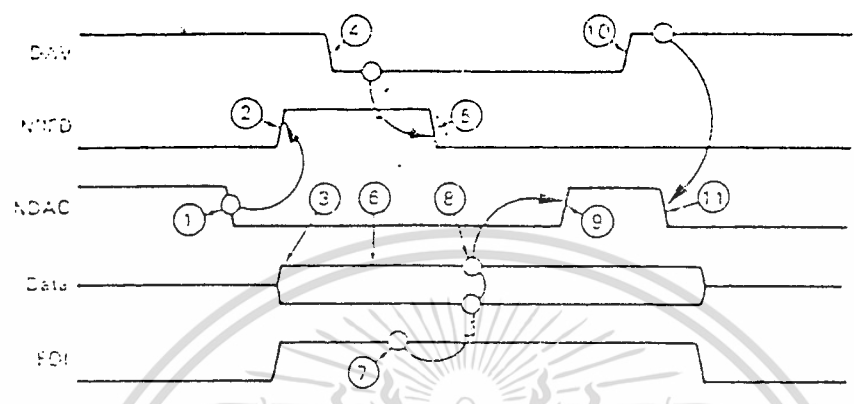
ขบวนการแฮนด์เชค เริ่มขึ้นโดยการที่ตัวควบคุมทำการเซตค่าให้สัญญาณ DAV มีค่าเป็น “Hi” (ในจุดที่ 1) ต่อมาเป็นการตรวจสอบว่าสัญญาณ NDAC และ NRFD มีลอจิกเป็น “Hi” ทั้งคู่หรือไม่ (ในจุดที่ 2) ถ้าทั้งคู่เป็น “Hi” นั้น คือเกิดการผิดปกติคืออุปกรณ์นั้นไม่พร้อมที่จะทำงานจึงทำการออกจากขบวนการ แฮนด์เชค

กลับมาดูในจุดที่ 2 หากสัญญาณใดสัญญาณหนึ่งมีลอจิกเป็น “Low” ตัว ควบคุมจะทำการส่งข้อมูลลงใน GPIB BUS (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะ ทำการตรวจสอบว่าสัญญาณ NRFD มีลอจิกเป็น “Hi” ใช่หรือไม่ หากไม่ใช่ก็จะทำ การตรวจสอบซ้ำอีกครั้งจนกระทั่งสัญญาณ NRFD มีลอจิกเป็น “Hi” (ในจุดที่ 4) หลังจากนั้นตัวควบคุมจะทำให้สัญญาณ DAV ให้มีลอจิกเป็น “Low” เพื่อให้ตัวรับ ทราบว่ามีข้อมูลอยู่ในบัสข้อมูล (ในจุดที่ 5) ตัวควบคุม จะรอเวลาเพื่อให้ตัวรับทำ การเก็บข้อมูลในบัสข้อมูล เมื่อครบกำหนดเวลาดตัวควบคุมจะทำการเชค ว่าตัวรับทำ งานเกินเวลาที่กำหนดหรือไม่หากเกินเวลาที่จะออกจากขบวนการแฮนด์เชค หากไม่เกิน เวลาที่จะทำการตรวจสอบต่อไปว่าสัญญาณ NDAC มีลอจิกเป็น “Hi” ใช่หรือไม่ (ในจุดที่ 6) หากไม่ก็จะทำการตรวจสอบอีกครั้งว่าเกินเวลาหรือไม่ กลับไปดูที่ จุดที่ 6 หากสัญญาณ NDAC เป็น “Hi” (ในจุดที่ 7) ตัวควบคุมจะทำการตอบ สอนองโดยการทำให้สัญญาณ DAV ให้มีลอจิก เป็น “Hi” เพื่อบอกให้ตัวรับทราบว่าให้ หยุดการเก็บข้อมูลจากบัส (ในจุดที่ 8) หลังจากนั้น ข้อมูลในบัสข้อมูลจะถูกนำ ออกไป (จุดที่ 9) จึงเป็นการเสร็จสิ้นขบวนการแฮนด์เชค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวรับ



รูปที่ 8

ขบวนการแฮนด์เชคเริ่มขึ้น หลังจากการที่ตัวควบคุมรับรู้ว่าตัวส่งจะทำการ ส่งข้อมูล เมื่อตัวควบคุมทราบว่าตัวส่งจะทำการส่งข้อมูล ตัวควบคุมจะทำการให้ สัญญาณ NDAC มีลอจิก เป็น “Low” เพื่อบอกให้ทราบว่าตัวควบคุมยังไม่ได้ทำ การ เก็บข้อมูล (ข้อมูลในที่นี้คือ แอดเดรส หรือ ข้อมูลทั่วไปซึ่งตัวส่งทำการส่งมาในบัสข้อมูล) นั่นคือในจุดที่ 1

ต่อมาตัวควบคุมจะทำการเซตค่าสัญญาณ NRFD ให้เป็น “Hi” เพื่อที่จะให้ตัวส่งทราบว่า ขณะนี้ตัวควบคุมพร้อมที่จะรับข้อมูลแล้ว (ในจุดที่ 2) เมื่อสัญญาณ NRFD มีค่าเป็น “Hi” และ สัญญาณ NDAC มีค่าเป็น “Low” แล้วตัวส่งจะทราบทันทีว่าทำการส่งข้อมูลลงมาในบัสข้อมูล ได้แล้ว (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะทำการรอเวลาให้ตัวส่งทำการส่งข้อมูลลงในบัสข้อมูล ให้เสร็จ โดยจะทำการตรวจสอบ ไปด้วยว่าเกินเวลาที่กำหนดหรือยังหากเกินเวลาที่กำหนดก็จะออก จากขบวนการแฮนด์เชค หากไม่เกินกำหนดเวลาจะทำการตรวจสอบต่อไปว่าตัวส่งทำการเซต ค่าให้ สัญญาณ DAV เป็น “Low” หรือไม่หากไม่ก็จะตรวจสอบเวลาว่าเกินเวลาที่ต้องรอแล้วหรือยัง หากยังไม่เกินเวลาที่จะทำการเซตต่อไปจนกระทั่งหมดเวลาหรือ จนกว่าสัญญาณ DAV จะมีค่าเป็น “Low” เมื่อสัญญาณ DAV เป็น “Low” แล้ว ตัวควบคุมจะตอบรับโดยการทำให้ สัญญาณ NRFD มีค่าเป็น “Low” (ในจุดที่ 5) นั่นเป็นการบอกว่าตัวควบคุม พร้อมที่จะเริ่มเก็บข้อมูล (ในจุดที่ 6) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ EOI เพื่อตรวจสอบว่าข้อมูลที่ตัวส่งทำการส่ง นั้นหมดแล้วหรือยัง (ถ้าตัวส่งทำ การส่งข้อมูลลงในบัสหมดแล้วจะทำการตั้งค่าของสัญญาณ EOI ให้เป็น “Low”) (ในจุดที่ 7) หากสัญญาณ EOI เป็น “Low” ตัวควบคุมจะเซต สถานะว่า

ขณะนี้ตัวส่ง ได้ทำการส่งข้อมูลลงมาในบัสหมดแล้ว หากสัญญาณ EOI เป็น “Hi” ก็ไม่ทำการ ... การค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขตสถานะ หลังจากนั้นตัวควบคุมจะทำการเก็บข้อมูลในบัส (ในจุดที่ 8) แล้วทำให้ สัญญาณ NDAC มีค่าเป็น “Hi” เพื่อบอกให้ทราบว่าตัวควบคุมได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว (ในจุดที่ 9) เมื่อตัวควบคุมทำให้สัญญาณ NDAC เป็น “Hi” แล้วตัวส่งจะลบข้อมูลในบัสข้อมูล โดยการทำให้ สัญญาณ DAV มีค่าเป็น “Hi” (ในจุดที่ 10) ซึ่งก่อนหน้านี้นี้ตัวควบคุมจะทำการตรวจสอบอยู่แล้วว่าสัญญาณ DAV เป็น “Hi” หรือยัง

เมื่อสัญญาณ DAV เป็น “Hi” แล้ว ตัวควบคุมจะทำให้สัญญาณ NDAC มีค่าเป็น “Low” (ในจุดที่ 11) แล้วจึงออกจากขบวนการแฮนด์เชค

คำสั่งใช้งานของ GPIB

การสั่งการต่างๆเพื่อกำหนดหน้าที่การทำงาน และกำหนดฟังก์ชัน เช่น กำหนดช่วงการวัด โหมดการวัดหรืออื่นๆแก่เครื่องวัดที่ต่ออยู่ เหล่านี้มันตัวควบคุม จะเป็นตัวกำหนดโดยการส่งรหัสคำสั่งไปที่อุปกรณ์ โดยผ่าน DI₁-DI₈ รหัสคำสั่งนี้ จะถูกส่งไปในช่วงที่สายสัญญาณ ATN เป็น LOW

คำสั่งสำหรับกำหนดหน้าที่การทำงานต่างๆตามมาตรฐานของ GPIB มี อยู่ด้วยกัน 128 คำสั่งดังแสดงในตารางที่ 1 ต่อไปนี้ โดยแบ่งเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIB บัสนั้นใช้ร่วมกันทั้งรหัสข้อมูล และรหัสคำสั่ง นั่นคือรหัสเดียวกัน มีความหมายได้ 2 อย่าง คือเมื่อ ATN เป็น LOW จะหมายถึง รหัสคำสั่ง แต่ถ้า ATN เป็น HIGH รหัสนี้จะแทนข้อมูลที่เป็น ASCII แทน ซึ่งในตารางก็ได้แบ่งความหมายออกเป็น 2 คอลัมน์

ASCII — IEEE 488 BUS MESSAGES (COMMANDS AND ADDRESSES) HEX CODES

MSD \ LSD	0		1		2		3		4		5		6		7	
	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16	.		o	
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	O	17	a		c	
2	STX		DC2		"	02	2	18	B	02	R	18	b		r	
3	ETX		DC3		#	03	3	19	C	03	S	19	c		s	
4	EOT	SDC	DC4	DCL	\$	04	4	20	D	04	T	20	d		t	
5	END	PPC	NAK	PPU	%	05	5	21	E	05	U	21	e		u	
6	ACK		SYN		&	06	6	22	F	06	V	22	f		v	
7	BEL		ETB		'	07	7	23	G	07	W	23	g		w	
8	BS	GET	CAN	SPE	(08	8	24	H	08	X	24	h		x	
9	HT	TCT	EM	SPD)	09	9	25	I	09	Y	25	i		y	
A	LF		SUB		*	10	:	26	J	10	Z	26	j		z	
B	VT		ESC		+	11	:	27	K	11	[27	k		[
C	FF		FS		,	12	<	28	L	12	\	28	l		\	
D	CR		GS		-	13	=	29	M	13]	29	m]	
E	SD		RS		.	14	>	30	N	14	_	30	n		_	
F	SI		US		/	15	?	31	U	15	~	31	u		~	

ADDRESSED
COMMAND
GROUP

UNIVERSAL
COMMAND
GROUP

LISTEN
ADDRESS
GROUP

TALK
ADDRESS
GROUP

SECONDARY
COMMAND
GROUP

PRIMARY COMMAND GROUP (PCG)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ตารางที่ 1 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กลุ่มคำสั่งเจาะจงจุดหมาย (Addressed command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ที่เป็นตัวส่งหรือตัวรับที่กำหนดไว้ล่วงหน้าแล้วคำสั่งนี้ประกอบด้วย

- GTL (go to local) สั่งให้อุปกรณ์กลับสู่สภาพการควบคุมปกติด้วยมือ
- SDC (selected device clear) สั่งให้อุปกรณ์เคลียร์ตัวเองสู่สภาพเริ่มต้นใหม่
- PPC (parallel poll configure) เป็นคำสั่งสำหรับการจัดสายสัญญาณของการทำการจัดสรรสายสัญญาณของการทำกระบวนการตรวจสอบสภาพ อุปกรณ์โดยวิธีขนาน โดยใช้กับกลุ่มคำสั่งรอง

- GET (group execute trigger) ใช้สั่งเริ่มต้นการทำงานของอุปกรณ์ที่หลายตัว
- TCT (take control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

2. กลุ่มคำสั่งครอบคลุม (universal command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่ต่ออยู่ในบัส ประกอบด้วย

- LLO (local lockout) เป็นการสั่งให้อุปกรณ์ล๊อคอยู่ที่สภาวะควบคุม โดยป้อนปรับที่หน้าปัดตามปกติ

- DCL (device clear) สั่งให้กับอุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น

- PPU (parallel poll unconfigure) ใช้ยกเลิกกระบวนการตรวจสอบสภาพแบบขนานทั้งหมด

- SPE (serial poll enable) เปลี่ยนโหมดของการตรวจสอบสภาพเป็นแบบอนุกรมในโหมดนี้ จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล

- SPD (serial poll disable) ยกเลิกการตรวจสอบแบบอนุกรม

3. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ (listener address group) เป็นคำสั่ง สำหรับกำหนดให้อุปกรณ์เป็นตัวรับ ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNL (unlisten) สำหรับยกเลิก

4. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวส่ง (talker address group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่งตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNT (untalker) สำหรับยกเลิกเช่นกัน

คำสั่งในกลุ่มที่ 1 ถึง 4 นั้น จัดเป็นกลุ่มคำสั่งหลัก ที่มีความหมายตายตัว ยังมีคำสั่งอีกกลุ่มที่ขึ้นอยู่กับการกำหนดภายหลังนั่นคือกลุ่มคำสั่งรอง

5. กลุ่มคำสั่งรอง (secondary command group) เป็นคำสั่งที่กำหนด รายละเอียดย่อยของอุปกรณ์แต่ละตัวที่ต่ออยู่ในระบบให้มีการทำงานอย่างไรตาม จุดประสงค์ใช้งานของเครื่องมีอนั้น เช่นเดียวกับการปรับป้อนต่างๆด้วยมีอนั้นเอง คำสั่งรองนี้จะตามหลังคำสั่งหลัก ก็จะใช้หลังจากอุปกรณ์ต่างๆถูกกำหนดวางตัว ในระบบเรียบร้อยแล้ว

คำสั่งต่างๆที่กล่าวไป ซึ่งใช้ในการกำหนดสภาวะการทำงานของอุปกรณ์ แต่ละสภาวะที่กำหนดไปนั้นเป็นอย่างไร และมีจุดประสงค์เพื่ออะไร ตัวต่อไปนี้

Device Clear/Interface Clear

Device Clear ใช้ในการทำให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปสู่สถานะเริ่ม ต้น ยังไม่มีการกำหนดฟังก์ชันใดๆ สถานะเริ่มต้นนี้จะแตกต่างกันไปแล้วแต่ว่า อุปกรณ์ นั้นออกแบบไว้อย่างไร Device Clear มีอยู่ 2 ลักษณะคือเคลียร์หมดทุก ตัวที่ต่ออยู่ (DCL) กับเคลียร์เฉพาะจงอุปกรณ์ตัวใดตัวหนึ่ง (SDC)

แต่ว่าในการเคลียร์อุปกรณ์ให้อยู่ในสถานะเริ่มต้นนั้น ไม่ได้หมายความว่า interface function ของ GPIB จะถูกเคลียร์ให้ไปอยู่ในสถานะเริ่มต้นด้วยแต่ อย่างไรก็ดี interface function คือสภาพการ interface ที่ได้กำหนดไว้ในระบบ ประกอบด้วยฟังก์ชันต่างๆ ดังแสดงในตารางที่ 2

ฟังก์ชัน	สัญลักษณ์	การกลับสู่สถานะเริ่มต้นโดย IFC
source handshake	SH	ได้
acceptor handshake	AH	ได้
talker or enlarge talker	T or TE	ได้
listener or enlarge listener	L or LT	ได้
service request	SR	ไม่ได้
remote/local	RL	ไม่ได้
parallell poll	PP	ไม่ได้
device clear	DC	ได้
device trigger	DT	ได้
controller	C	ได้

ตารางที่ 2

Remote/Local

remote เป็นการกำหนดให้อุปกรณ์ที่อยู่ในระบบเช่น เครื่องมือวัด ให้อยู่ ในการควบคุมของอุปกรณ์ของอุปกรณ์ตัวอื่นแทน ซึ่งปุ่มปรับต่างๆบนหน้าปัด เครื่องจะไม่มีผลต่อการทำงาน ส่วน local เป็นการควบคุมการทำงานของเครื่อง มือวัดด้วยปุ่มปรับบนหน้าปัดตามปกติ

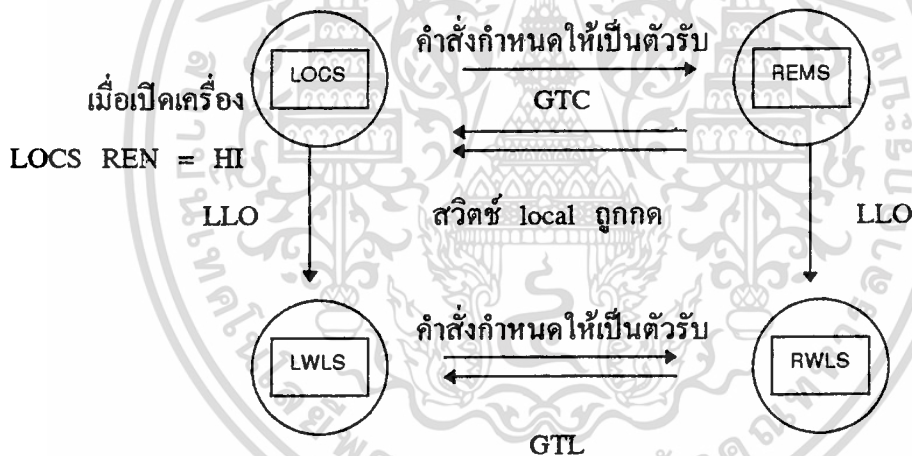
การใช้ remote มีประโยชน์ในแง่ที่ขณะที่ตัวการควบคุมเช่น คอมพิวเตอร์ กำลังติดต่อ อุปกรณ์ตัวนั้นอยู่ หากไม่มีการตัดการควบคุมโดยปุ่มปรับบนหน้าปัด ออก ถ้ามีใครมาปรับแต่ง

เอกสารนี้เป็นทรัพย์สินของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ ถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะทำให้การทำงานผิดพลาดไปได้ การทำงานของ GPIB ใน remote and local มี 4 ลักษณะดังนี้

1. LOCS ก็คือ local นั่นเอง เป็นสภาพการควบคุมที่ปุ่มปรับตามปกติ จะอยู่ในสภานี้ ตอนเปิดเครื่อง หรือ REN เป็น HIGH หรือเมื่อได้รับคำสั่ง GTL
2. REMS คือ remote หมายถึงการตัดการควบคุมโดยปุ่มหน้าปิดออก จะเกิดขึ้นเมื่อ REN เป็น LOW และจะถูกล็อกไว้ เว้นแต่ว่าสวิทช์ local ที่ตัว อุปกรณ์จะถูกเปลี่ยนไปตำแหน่ง local
3. RWLS เป็นสภาพ remote ที่ถูกล็อกเอาไว้เช่นกัน แต่จะตัดการควบคุมตรงสวิทช์ local ที่อุปกรณ์ออกไป สภาพ remote โดย RWLS จึงมีความสำคัญสูงกว่า REMS อย่างไรก็ตาม ยังถูกยกเลิกได้ด้วยคำสั่ง LLO
4. LWLS มีสภาพเช่นเดียวกับ local แต่จะแตกต่างกันตรงที่สภาพ local โดย LWLS นี้ เมื่อได้รับคำสั่งกำหนดอุปกรณ์ตัวรับจะเปลี่ยน ไปอยู่ในสภาพแบบล็อกหรือ RWLS ทั้งนี้ในการที่จะมาที่สภาพ LWLS นี้ได้ก็มี 2 กรณีคือ เมื่ออยู่ในสภาพ local ธรรมดา (LOCS) แล้วได้รับคำสั่ง LLO หรืออยู่ใน RWLS แล้ว ได้รับคำสั่ง GTL



รูปที่ 9

การขอบริการและการตรวจสอบ (Service Request and Polling)

เมื่อตัวควบคุมได้รับ SRQ เป็น LOW จะให้อุปกรณ์ส่งข้อมูลแสดงสถานะ การทำงาน ซึ่งมีอยู่ 2 วิธีคือ

1. การตรวจสอบแบบอนุกรม ซึ่งมีขั้นตอนดังนี้
 - 1.1 ANT ถูกดึงเป็น LOW หลังจากได้รับ LOW จากสายสัญญาณ SRQ
 - 1.2 คำสั่ง UNL ถูกส่งไปยังอุปกรณ์
 - 1.3 ตัวควบคุมจะแจ้งรหัสตัวรับของตน และกำหนดรหัสตัวส่งของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะตรวจสอบไปที่บัส

1.4 ตามด้วยคำสั่ง SPE และสาย ATN กลายเป็น HI ซึ่งอุปกรณ์ที่ถูกเรียกจะส่งข้อมูลแสดงสถานะออกมา 1 ไบต์โดยบิตที่ 7 จะเป็นตัวชี้ว่าอุปกรณ์นี้เป็นตัวขอบริการ ถ้าใช่จะเป็น LOW ส่วนบิตอื่นๆก็ใช้บอกข้อมูลอื่นๆซึ่งมิได้กำหนดเฉพาะ

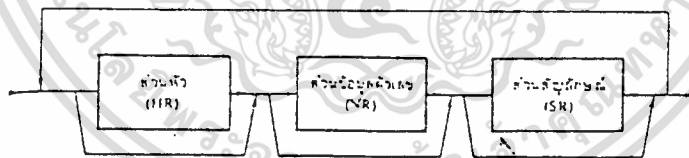
1.5 สาย ATN ถูกดึงเป็น LOW อีกทีเพื่อส่งคำสั่งยกเลิกการตรวจสอบคือ SPD

1.6 จากนั้นคำสั่ง UNT ก็ถูกส่งไปยังอุปกรณ์เพื่อยกเลิกการเป็นตัวส่งซึ่งถ้าหาก SQR ยังคงเป็น LOW อยู่ ก็จะมีการตรวจสอบไปยังอุปกรณ์ตัวอื่นๆต่อไปตามขั้นตอนเดิม

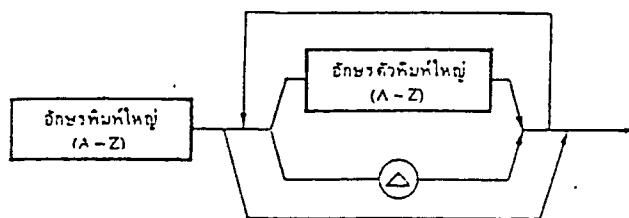
2. การตรวจสอบแบบขนาน สามารถทำได้เร็วกว่าแบบอนุกรม ทั้งนี้เพราะ สามารถอ่านข้อมูลเพียงไบต์เดียว ก็สามารถรู้ได้ทันทีว่าอุปกรณ์ตัวใดเป็นผู้ขอบริการ

รูปแบบของข้อมูล

โดยทั่วไปข้อมูลจากอุปกรณ์ (device message) แบ่งได้ออกเป็น 3 ส่วน ตัวแสดงในรูปที่ 10 อันได้แก่ส่วนหัว(HR) ซึ่งจะอยู่ส่วนหน้าสุด เป็นตัวบอก ชนิดข้อมูล ส่วนประกอบ HR แสดงในรูปที่ 11 จะเห็นว่าประกอบด้วยตัวอักษร ภาษาอังกฤษตัวพิมพ์ใหญ่กับช่องว่างที่เป็นเว้นวรรค (Δ) ปกติจะมีอักษรประมาณ 1-3 ตัว

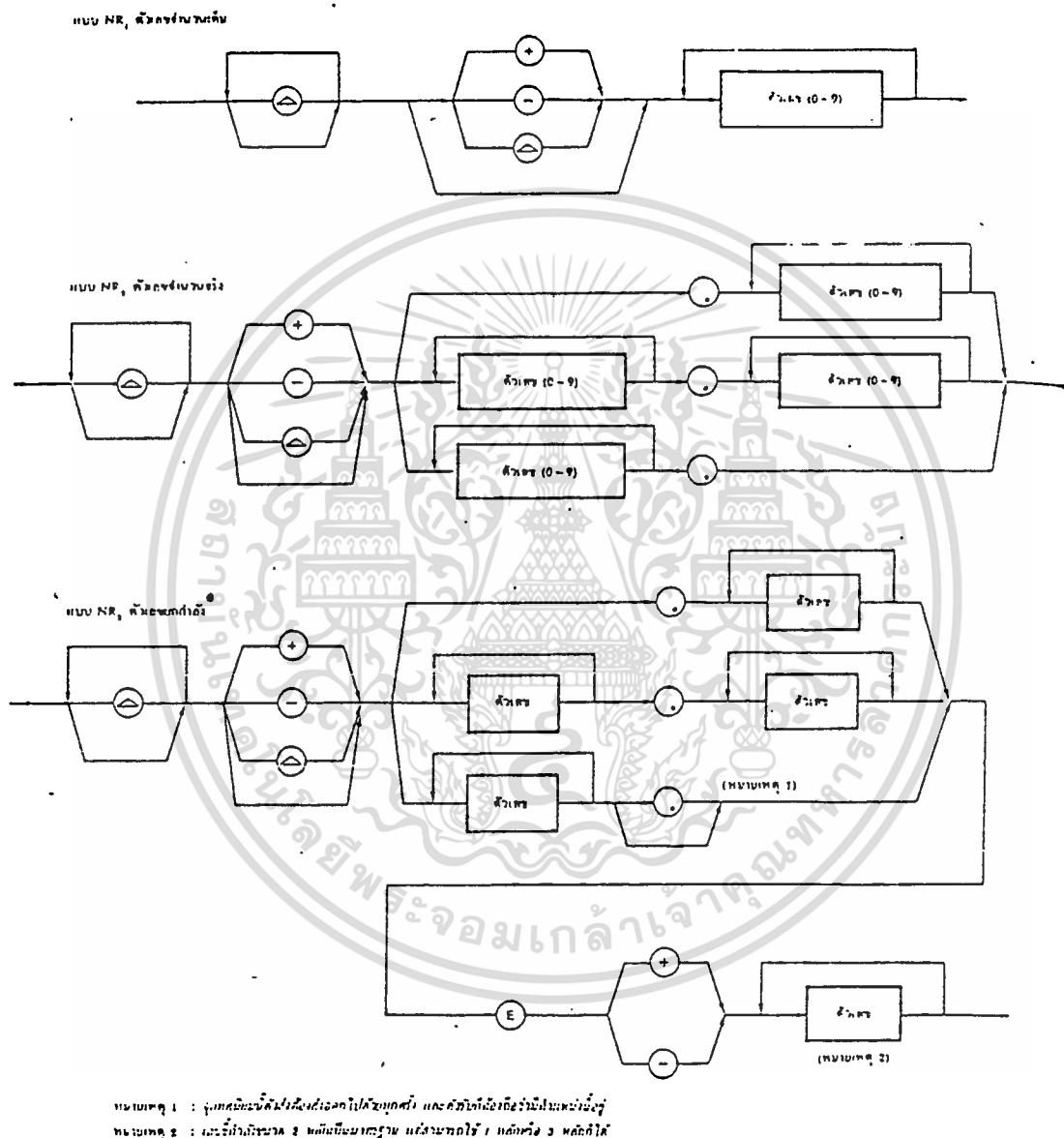


รูปที่ 10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 11
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

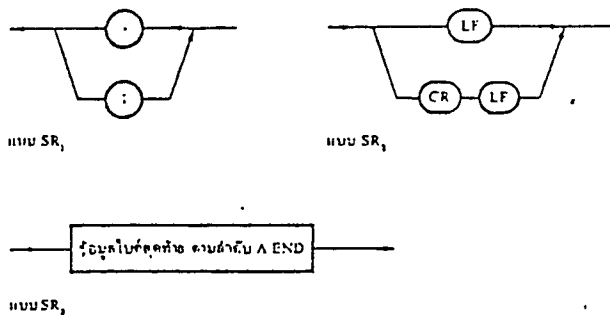
ส่วนที่สอง คือ เนื้อหาของข้อมูล (NR) ซึ่งใช้แสดงค่าตัวเลข มีอยู่ 3 แบบ คือ NR_1 , NR_2 และ NR_3 ดังแสดงในรูปที่ 12 ส่วนท้ายของ NR ยังอาจมีตัวอักษรแสดงหน่วยตามมา



รูปที่ 12

ส่วนที่ 3 คือสัญญาณแบ่งข้อมูลแต่ละชุด (SR) ดังแสดงในรูปที่ 13 โดย SR_1 ใช้แสดงการต่อเนื่องของข้อมูล (ข้อมูลยังมีต่อ) SR_2 และ SR_3 แสดงการสิ้นสุดของข้อมูลแต่ SR_3 เป็นการบอกการเสร็จสิ้นข้อมูลทั้งหมดจากการวัด

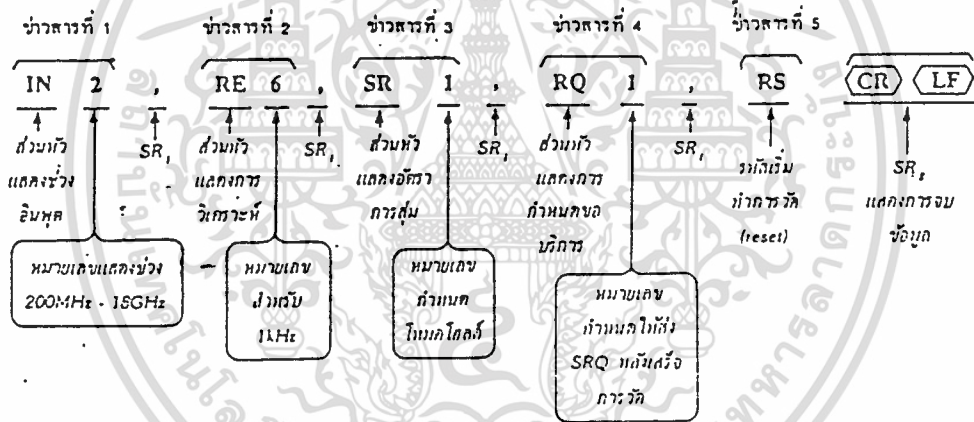
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



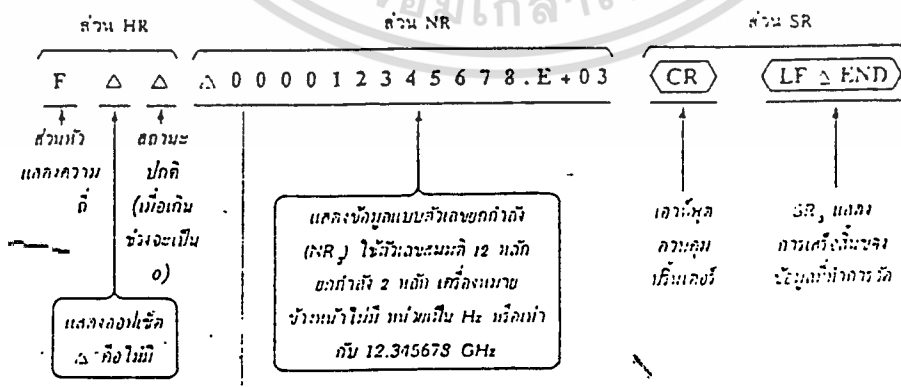
รูปที่ 13

รูปที่ 14 เป็นตัวอย่างข้อมูลสำหรับการกำหนดฟังก์ชันให้เครื่องวัดความถี่ และข้อมูลความถี่ที่วัดได้

ตัวอย่างข้อมูลสำหรับการกำหนดฟังก์ชัน



ตัวอย่างข้อมูลรหัสทุกที่ได้จากเครื่องวัด



รูปที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทแทรกที่ 2

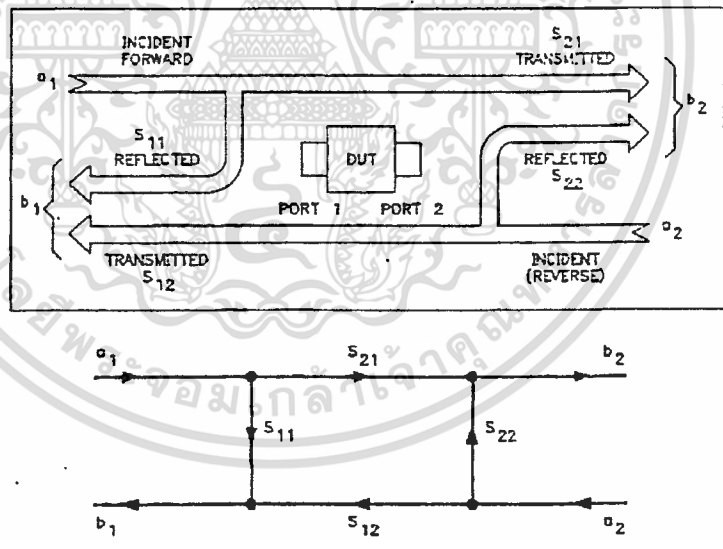
การใช้เครื่องวิเคราะห์โครงข่ายสำหรับการวัดแบบส่งผ่าน

S-พารามิเตอร์

S-พารามิเตอร์ หรือ พารามิเตอร์ของการแตกกระจาย (Scattering Parameters) เป็นข้อตกลง ที่ถูกใช้เป็นลักษณะเฉพาะตัวของอุปกรณ์ที่ใช้ปรับปรุงการไหลของสัญญาณ คำอธิบายคร่าว ๆ ของ S-พารามิเตอร์ คือถูกใช้ในอุปกรณ์ 2 ช่องทาง (two-port device) สำหรับรายละเอียดจะมีอยู่ในคู่มือ AN 95-1 และ AN 154 ซึ่งเป็นเอกสารการประยุกต์ใช้ของเฮวลิต-แพคการ์ด (Hewlett-Packard)

S-พารามิเตอร์ เป็นอัตราส่วนของปริมาณเชิงซ้อน 2 ตัว คือ แมกนิจูด (magnitude) และเฟส (phase) S-พารามิเตอร์ สามารถแสดงโดยใช้ตัวเลขประกอบเช่น S out in

ตัวเลขตัวแรก (out) หมายถึง ช่องทางที่สัญญาณออกมาจากอุปกรณ์ภายใต้การทดสอบ (DUT) และตัวเลขตัวที่สอง (in) คือ ช่องทางที่สัญญาณเกี่ยวเนื่อง ตัวอย่างเช่น S-พารามิเตอร์เป็น S₂₁ หมายถึง การวัดอัตราส่วนเชิงซ้อนของสัญญาณที่ออกมาจาก ช่องทางหมายเลข 2 ต่อสัญญาณทางช่องทาง หมายเลข 1



รูปที่ 1

รูปที่ 1 แสดง S-พารามิเตอร์ของอุปกรณ์ 2 ช่องทาง และแสดงภาพการไหลโดยเสมือนของสัญญาณ โดย "a" หมายถึง สัญญาณทางด้านเข้า และ "b" หมายถึงสัญญาณทางด้านออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการวัดทางฟอร์เวิร์ด (FWD) สัญญาณต้นกำเนิดออกมาทางช่องทางหมายเลข 1 และ สำหรับทางรีเวิร์ด (REV) สัญญาณต้นกำเนิดจะออกมาทางช่องทางหมายเลข 2 ตัวอย่างเช่น

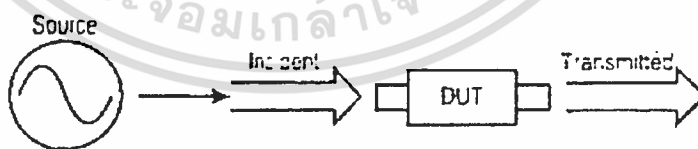
S_{21} จะมีค่าเท่ากับ $|S_{21}| \angle \theta_{21}$ โดย S_{21} จะมีค่าเท่ากับ $E_2 \angle \theta_2$ หารด้วย $E_1 \angle \theta_1$ โดยค่า S-พารามิเตอร์ต่าง ๆ จะแสดงในตารางที่ 1

Measurement	S-Parameter	Test Direction	Signal Path
Input reflection (port 1)	S11	FWD	A/R
Forward gain/loss (transmission)	S21	FWD	B/R
Reverse gain/loss (transmission)	S12	REV	A/R
Output reflection (port 2)	S22	REV	B/R

ตารางที่ 1 แสดง S-พารามิเตอร์ชนิดต่าง ๆ

การวัดการส่งผ่านสัญญาณ (Transmission Measurements)

Insertion Loss and Gain



รูปที่ 2

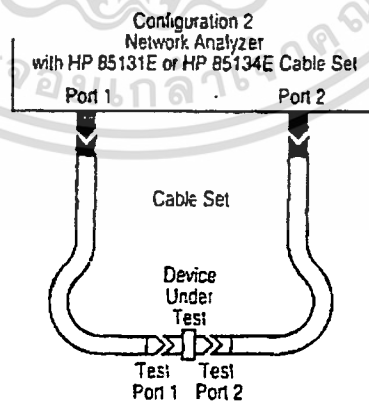
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราสูญเสีย (loss) และอัตราขยาย (gain) เป็นอัตราส่วนลอการิทึมของแรงดันที่ถูกส่งผ่านออกมากับแรงดันที่เข้าไป โดยผ่านอุปกรณ์ที่มี 2 ช่องทาง การวัดทั้งสองอย่าง จะทำการจัดตั้ง (setup) และปรับให้อยู่ตำแหน่งมาตรฐาน (calibration) เหมือน ๆ กันแต่ยกเว้นการวัดอัตราขยาย (gain) ที่ต้องการระดับกำลังสัญญาณทางเข้าต่ำกว่า โดยเลือกปุ่มกดดังนี้

Preset (factory)	[USER PRESET]
Setup	
Measurement :	[MEAS][Trans:FWD S21]
Format :	[FORMAT] [REAL]
Stimulus :	[START] 11500 [M/μ]
	[STOP] 12500 [M/μ]

การเชื่อมต่อและการวัด

เพื่อทำการปรับให้อยู่ในตำแหน่งมาตรฐาน โดยเชื่อมต่ออุปกรณ์ที่จะทดสอบ และกด [MEAS] หรือ [MEASURE RESTART] เพื่อปรับปรุงข้อมูลของ S-พารามิเตอร์ ทั้งทางด้านไปข้างหน้า (forward) และย้อนกลับ (reverse) และพารามิเตอร์ทางด้านไปข้างหน้า (forward) เท่านั้นที่จะถูกวัดจากการกวาด (sweep to sweep) เมื่อจะทำการวัดอุปกรณ์ ตัวใหม่เพื่อทำการวัด ซึ่งจะถูกแนะนำให้วัด S-พารามิเตอร์ทั้ง 4 ใหม่ โดยการกดปุ่ม [MEAS]



รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการปรับสเกลของจอแสดงผลของการวัดเรียบร้อยแล้ว จึงเลือก marker ที่ตำแหน่งตรงกลางของจอแสดงผล เพื่อวัดอัตราการสูญเสีย (loss) ค่าของ marker จะแสดงที่มุมขวาบนของจอภาพ ซึ่งเราสามารถเลือก marker ไปยังตำแหน่งที่เราต้องการได้

การปรับปรุงข้อมูลของ S-พารามิเตอร์

โดยกดปุ่มดังนี้

[MEAS]

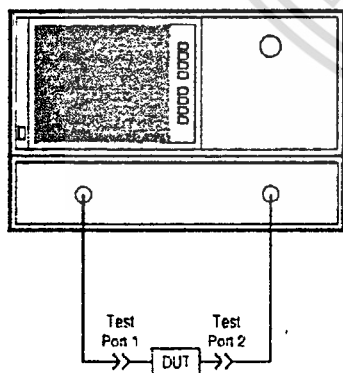
[SCALE][AUTOSCALE]

[MKR] [MARKER1] 12000 [G/N]

อ่านค่า loss จากค่าที่ marker

การเก็บและแสดงค่าผลลัพธ์ : การพล็อต

ก่อนทำการพล็อตค่าเราต้องแน่ใจในอุปกรณ์พล็อต (plotter) ว่าถูกจัดตั้งเรียบร้อยแล้ว จากนั้นกดปุ่ม [COPY]ตามด้วย [PLOT] ซึ่งจะพล็อตค่าปัจจุบันทุกอย่างที่ถูกแสดงผลยกเว้น softkey menu ส่วนปุ่ม [DEFINE PLOT] softkey จะเอาไว้เลือกการพล็อตแบบ พิเศษ ๆ เช่น [PLOT DATA] การวัดการตอบสนองของเฟส



รูปที่ 4

การวัดการตอบสนองของเฟสก่อนอื่นต้องทำการปรับตำแหน่งมาตรฐานอีกครั้งหนึ่งซึ่งแสดงวิธีการดังนี้

Recall Cal [RECALL] [RECALL REG1]

Adjust Setup

Measurement [CH]

[MAS] [Trans:FWD S21]

Format

[FORMAT] [PHASE]

Stimulus

[CAL] [FRQ SUBSET ON off]

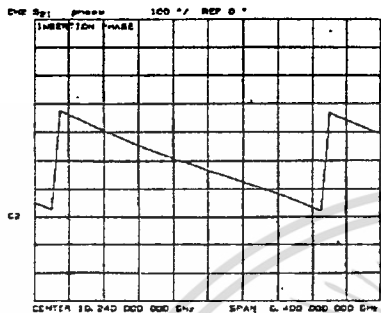
[START] 11500 [M/μ]

[STOP] 12500 [M/μ]

Save

[SAVE] [SAVE REG2]

การปรับปรุงข้อมูล S-พารามิเตอร์



การปรับปรุงให้กดปุ่มดังนี้

[MEAS]

[SCALE REF] [AUTO

SCALE]

รูปที่ 5

การเก็บค่าและการแสดงผลลัพธ์ : การพิมพ์

ทำการติดตั้งเครื่องพิมพ์ให้พร้อมใช้งาน เลือกปุ่ม [COPY] ตามด้วย [PRINT] เครื่องพิมพ์จะพิมพ์ข้อมูลทุกชนิดที่แสดงผลอยู่บนจอ ยกเว้น softkey menu

การแสดงผลแบบ 2 ช่อง

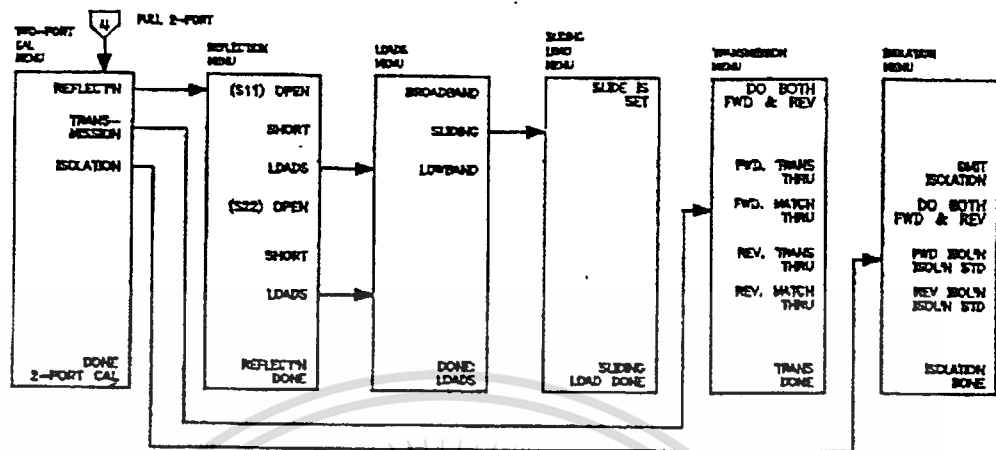
การแสดงผลแบบ 2 ช่องบนจอในเวลาเดียวกันกด [DISPLAY] แล้วเลือก [DUAL CHAN ON off] ก็จะได้ภาพแบบ 2 ช่องสัญญาณ

การปรับตำแหน่งมาตรฐานสำหรับการสะท้อนกลับและส่งผ่านทั้ง 2 ช่องทาง

การปรับตำแหน่งมาตรฐาน (Calibration) ที่แสดงในรูปที่ 6 มีความสมบูรณ์และถูกต้องสำหรับทั้ง 4 ตัวของ S-พารามิเตอร์ เป็นวิธีการปรับตำแหน่งมาตรฐานที่แม่นยำที่สุดของการวัดแบบ 2 ช่องทาง ขั้นตอนปฏิบัติดังนี้

- ใช้ MEASURE RESTART ใน S-พารามิเตอร์ เมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6

- เลือกชนิดของการปรับตำแหน่งมาตรฐานกลุ่ม CALIBRATE MENU ตามด้วย FULL2-PORT ตามด้วย REFLECT'N จากนั้นต่อ open circuit เข้ากับช่องทาง 1 กด (S11) OPEN ซึ่งหมายถึงข้อมูลของการเปิดวงจรจะถูกวัด
- ถอด open circuit ออกแล้วต่อ short circuit เข้าแทนที่ แล้วทำการกด (S11) SHORT ข้อมูลของ SHORT CIRCUIT จะถูกวัด
- ถอด SHORT CIRCUIT ออก แทนที่ด้วย impedance-match load แล้วทำการกด (S11) LOADS ทำการเลือก SLIDE IS SET
- กดปุ่ม SLIDING LOAD DONE แล้วกดปุ่ม DONE LOADS หลังจากนั้น ทำการวัดแบบ open-short-load ตามที่กล่าวมาข้างต้น แต่ให้เปลี่ยนอุปกรณ์ไปเชื่อมต่อกับช่องทาง 2 และใช้ปุ่ม (S22) soft key เมื่อทำเสร็จแล้วให้กด REFLECT'N DONE (ถ้ากด DONE โดยไม่ได้ทำการวัดตามมาตรฐานให้ครบ จอภาพจะแสดงข้อความว่า "CAUTION : ADDITIONAL STANDARDS NEED")
- กด TRANSMISSION แล้วทำการเชื่อมต่อ ช่องทาง 1 กับ ช่องทาง 2 โดยใช้อุปกรณ์ที่จะทดสอบอยู่ระหว่างช่องทาง 1 และช่องทาง 2 กดปุ่ม DO BOTH FWD & REV ตามด้วย TRANS : DONE ต่อจากนั้นถอดการเชื่อมต่อออก เมื่อต้องการความถูกต้องของการแยกจากกัน (isolation) กด AVG ตามด้วย AVERAGING ON แล้วจึงกด CAL ตามด้วย RESUME , CAL SEQUENCE , ISOLATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการต่อ impedance-math loadsเข้ากับช่องทาง 1 และ 2 แล้วกด DO BOTH FWD & REV เพื่อทำการวัดโดยแยกกัน ทั้งทาง Forward และทาง Reverse จากนั้นกดปุ่ม ISOLATION DONE กดปุ่ม AVG ตามด้วย AVERAGING OFF

- กดปุ่ม CAL ตามด้วย RESUME , CAL SEQUENCE กดปุ่ม DONE 2-PORT CAL (ถ้ากด DONE โดยไม่มีการวัดตามมาตรฐานที่ครบด้านตามต้องการ จอภาพจะแสดงข้อความ "CAUTION : ADDITIONAL STANDARDS NEEDED") ค่าสัมประสิทธิ์การปรับตำแหน่งมาตรฐาน จะถูกคำนวณและเก็บเอาไว้ และจอภาพจะแสดง "C2" ที่มุมซ้ายของจอแสดงว่า error correction ของ two-port แสดงค่า on

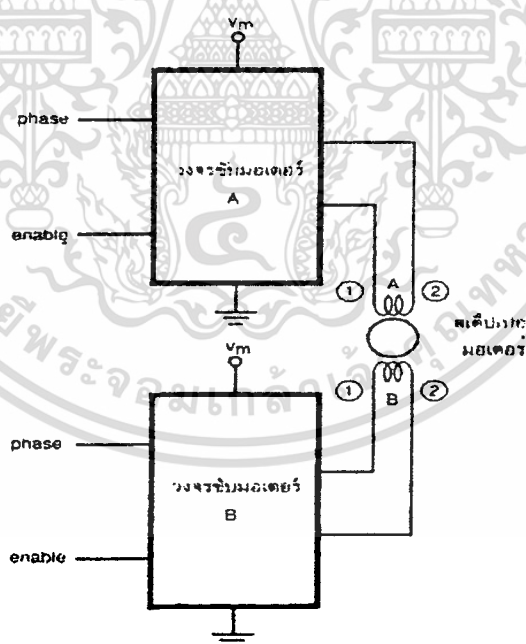


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทแทรกที่ 8 เครื่องสแกนเนอร์

การเคลื่อนที่ของพล็อตเตอร์

เริ่มแรกต้องกำหนดพื้นที่เป็นจุด ๆ โดยกำหนดเป็นแกน X และ แกน Y เริ่มจากจุด X_0, Y_0 เลื่อนไปทางขวาเป็นแกน X เลื่อนขึ้นบนเป็นแกน Y เส้นแบ่งในแกน X และ Y หมายถึง สเต็ปที่ละเอียดที่สุดของเครื่อง พล็อตเตอร์มาตรฐานทั่วไปความละเอียดของสเต็ปจะเท่ากับ 25 ไมครอน (ไมโครเมตร) หรือ 1/1016 นิ้ว พล็อตเตอร์เครื่องนี้ใช้สเต็ปมอเตอร์แบบเคลื่อนที่ 1.8 องศาต่อสเต็ป ในหนึ่งรอบของมอเตอร์มีการเคลื่อนที่ 200 สเต็ป (360 องศา หารด้วย 1.8) และถ้าเคลื่อนที่แบบครึ่ง สเต็ปจะเคลื่อนที่ได้ 400 สเต็ปต่อรอบ ฉะนั้น ในระยะทาง 1 นิ้ว มอเตอร์จะต้องหมุนถึงสองรอบครึ่ง จึงจะได้จำนวนสเต็ปเท่ามาตรฐาน ซึ่งในทางปฏิบัติภาคกลไกทำได้ยาก ถ้าใช้มอเตอร์ที่มีความละเอียดสูงกว่านี้ ก็จะหามอเตอร์ได้ยาก ราคาแพง แต่สามารถที่จะทำให้มอเตอร์นั้นมีการเคลื่อนที่ที่ละเอียดมากขึ้น โดยการใช่วงจรขับมอเตอร์แบบแบ่งสเต็ปละเอียด (micro step driver)



รูปที่ 1 วงจรขับสเต็ปมอเตอร์เบื้องต้น

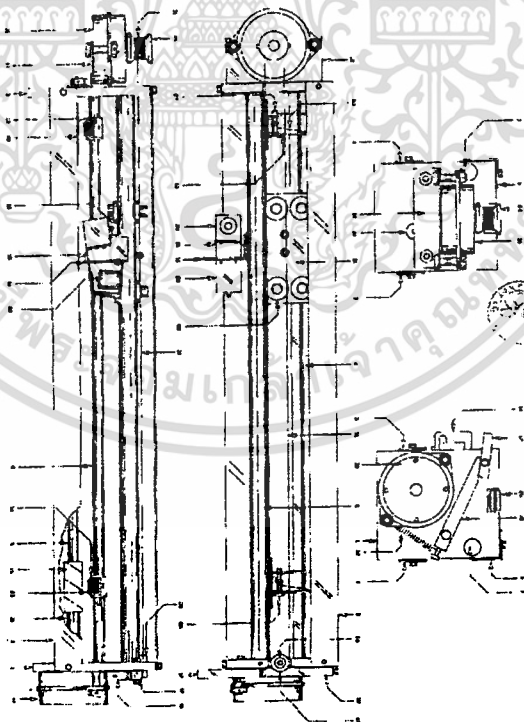
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขับมอเตอร์

การขับสเต็ปมอเตอร์ในพล็อตเตอร์นี้ ใช้วงจรขับมอเตอร์แบบแบ่งละเอียดจากรูปที่ 1 ใช้วงจรขับมอเตอร์ 2 วงจรที่มีคุณสมบัติเหมือนกันทุกประการ เพื่อขับมอเตอร์หนึ่งตัว ขา PHASE เป็นตัวกำหนดทิศทางการไหลของกระแสในขดลวด ขา ENABLE ใช้ควบคุมให้เกิดกระแสไหลในขดลวดในการขับมอเตอร์แบบเต็มสเต็ป (full step driver) นั้น มีหลักการง่าย ๆ คือ ให้มีกระแสไหลในขดลวด 2 ขดสลับกัน ซึ่งสามารถสลับได้ 4 แบบ

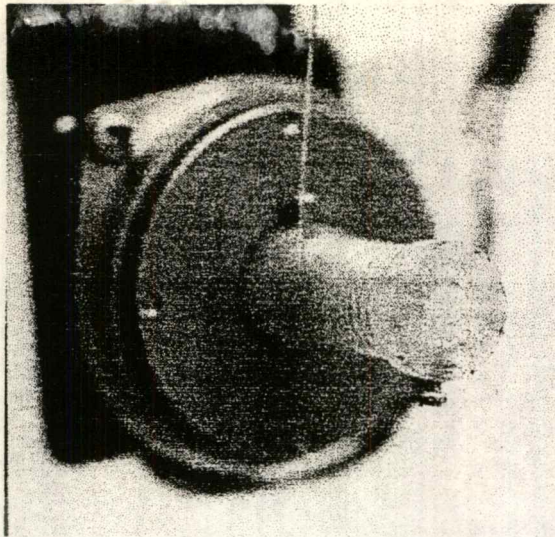
ถ้าต้องการขับให้มอเตอร์หมุนแบบครึ่งสเต็ป (half step driver) ก็กำหนดให้มีการหยุดไหลของกระแสในขดลวด A และ B คั่นสลับอยู่ในลักษณะการหมุนแบบเต็มสเต็ป ก็จะได้การหมุนของมอเตอร์แบบครึ่งสเต็ป จะเห็นว่าสามารถทำให้มอเตอร์เคลื่อนที่ได้ละเอียดขึ้นอีกเท่าตัว เมื่อควบคุมกระแสในขดลวดได้ จากแนวความคิดอันนี้จึงนำมาพัฒนาเป็นวงจรขับมอเตอร์แบบแบ่งละเอียด ซึ่งสามารถทำให้ สเต็ปมอเตอร์ธรรมดากลายเป็นสเต็ปเปอร์มอเตอร์ที่มีสเต็ปการหมุนที่มีความละเอียดสูง โดยสามารถทำให้ละเอียดเท่าไรก็ได้ตามต้องการ ในโครงงานนี้จะกำหนดไว้ 1600 สเต็ปต่อรอบ ส่วนกลไก

กลไกของพล็อตเตอร์ แสดงแบบและขนาดไว้แล้วในรูปที่ 2



รูปที่ 2 กลไกของพล็อตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



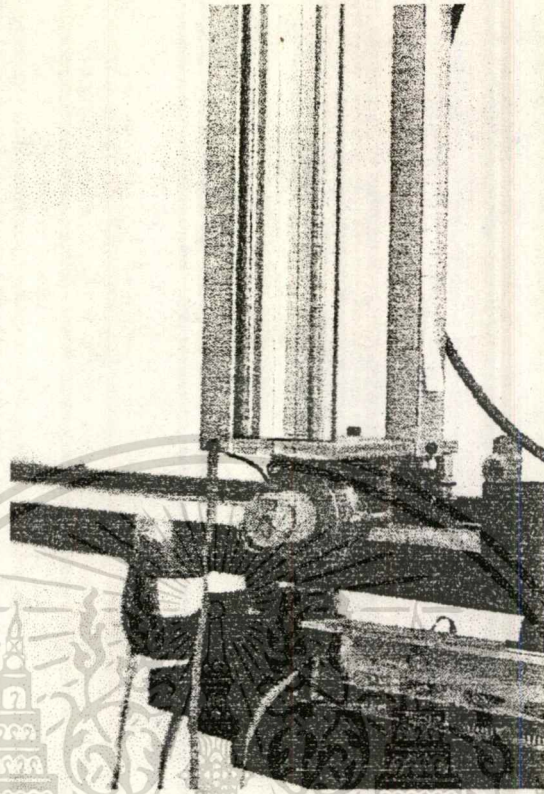
รูปที่ 3 มอเตอร์ขับเคลื่อนวาระนาบ

การใช้งาน

เนื่องจากวงจรควบคุมเป็นแบบอิเล็กทรอนิกส์ จึงไม่มีหน่วยความจำบัฟเฟอร์ การส่งคำสั่งจากคอมพิวเตอร์ จึงใช้พอร์ตขนานเพียงอย่างเดียว เพราะสามารถตรวจสอบแฮนด์เชกกิ้งได้ดีกว่าพอร์ตอนุกรม RS-232 และมีความเร็วในการส่งสูงกว่ามาก เสมือนว่าคอมพิวเตอร์เป็นหน่วยความจำบัฟเฟอร์ โปรแกรม CAD ทั่วไปสามารถส่งคำสั่งเข้ามาทางพอร์ตขนานได้อยู่แล้ว ขนาดของการ SCAN ใช้ได้ 2 ขนาด คือ A2 หรือ A3 กำหนดด้วยการเลื่อนสวิตช์ S1

การประยุกต์พล็อตเตอร์เป็นสแกนเนอร์

สำหรับการประยุกต์พล็อตเตอร์เป็นสแกนเนอร์ ได้นำเอามอเตอร์ส่วนที่ขับเคลื่อนกระดาษของพล็อตเตอร์ นำมาเป็นมอเตอร์ที่ทำหน้าที่เลื่อนชุดพล็อตเตอร์ทั้งชุดให้วิ่งแนวระนาบ ดังแสดงในรูปที่ โดยต้องออกแบบระบบการขับเคลื่อนแนวระนาบเพิ่มเติม ซึ่งต้องเพิ่มรอกพลาสติกที่มุมของโครงสแกนเนอร์ แต่ทางด้านซอฟต์แวร์ก็ยังคงเหมือนเดิมไม่มีการเปลี่ยนแปลง

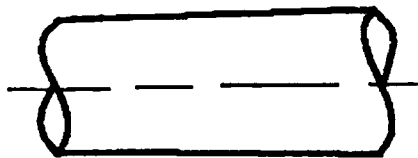


รูปที่ 4 มอเตอร์ขับเคลื่อน

ข้อผิดพลาดของโลก

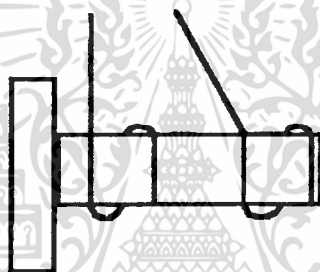
ความแม่นยำในการเคลื่อนที่แต่ละสแต็ปของสแกนเนอร์มีความแตกต่างกันเล็กน้อยเนื่องจาก
 แนวนวนาบบมีการเคลื่อนที่ระยะทางไกล ลวดสลิงที่ใช้ก็ต้องยาวมากขึ้น ซึ่งภายหลังได้ทำการเปลี่ยน
 ลวดสลิงมาเป็นสายเอ็น สายเอ็นนั้นแม้จะมีความยืดหยุ่นมากกว่า แต่ก็ให้ความถูกต้องมากกว่าลวด
 สลึง ทำให้มีความแม่นยำมากขึ้น ส่วนการขับเคลื่อนของมอเตอร์ในแนวดิ่ง ก็ยังคงใช้ลวดสลิง
 เหมือนเดิม สำหรับการผิดพลาดของสแต็ปการเคลื่อนที่แนวนอนอาจวิเคราะห์ ได้ดังนี้

1. เส้นผ่านศูนย์กลางของมอเตอร์ขับเคลื่อนแนวนอนไม่สม่ำเสมอ เช่น กำหนดให้สแกนเนอร์หมุน
 2.5 รอบ จะได้ระยะทาง 1 นิ้ว ดังนั้นเส้นผ่านศูนย์กลางของแกนมอเตอร์จะต้องเท่ากับ 3.21
 มิลลิเมตร เท่ากันตลอดแนวแกน ซึ่งในทางปฏิบัติทำได้ยาก ดังแสดงในรูปที่ 5



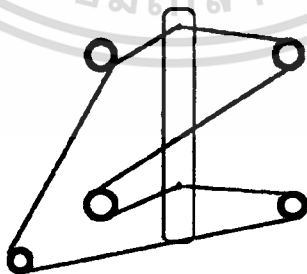
รูปที่ 5 แกนขั้วของมอเตอร์

2. ระยะทางของสายเอ็นไม่คงที่ เนื่องจากในการพันสายเอ็นรอบแกนขั้วในแนวระนาบจะมีจำนวนรอบมากกว่ามอเตอร์ขั้วแกนตั้ง เพราะในระนาบแนวนอนมีระยะเคลื่อนที่มากกว่าระนาบแนวตั้ง ดังนั้นจึงเกิดความเบี่ยงเบนของแนวสายเอ็น ทำให้ระยะทางเคลื่อนที่ผิดพลาดได้ ดังแสดงในรูปที่ 6



รูปที่ 6 แนวของสายเอ็น

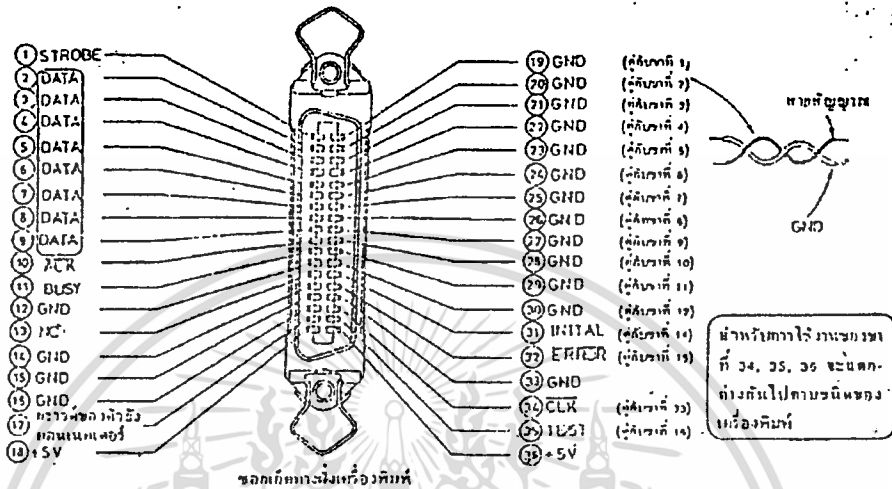
3. การไม่ได้แนวระดับอย่างแท้จริงของจุดยึดสายเอ็นกับชุดแกนเนอร์ ทำให้เกิดความผิดเมื่อชุดแกนเนอร์อยู่บริเวณขอบของโครงเหล็ก ดังแสดงในรูปที่ 7



รูปที่ 7 แนวยึดสายเอ็น

มาตรฐานการเชื่อมต่อแบบขนาน

สำหรับการติดต่อระหว่างคอมพิวเตอร์กับพอร์พรีนเตอร์นั้น จะใช้การติดต่อระบบเซ็นทรอนิกส์ (CENTRONICS PARALLEL PORT) ซึ่งรายละเอียดของขาสัญญาณต่าง ๆ แสดงดังรูปที่ 8 โดยขาสัญญาณต่าง ๆ ที่มาตรฐานมีดังนี้



รูปที่ 8 แสดงขาของเซ็นทรอนิกส์

เซ็นทรอนิก เป็นพอร์ตอินเตอร์เฟสประกอบด้วย 36 ขา ดังนี้

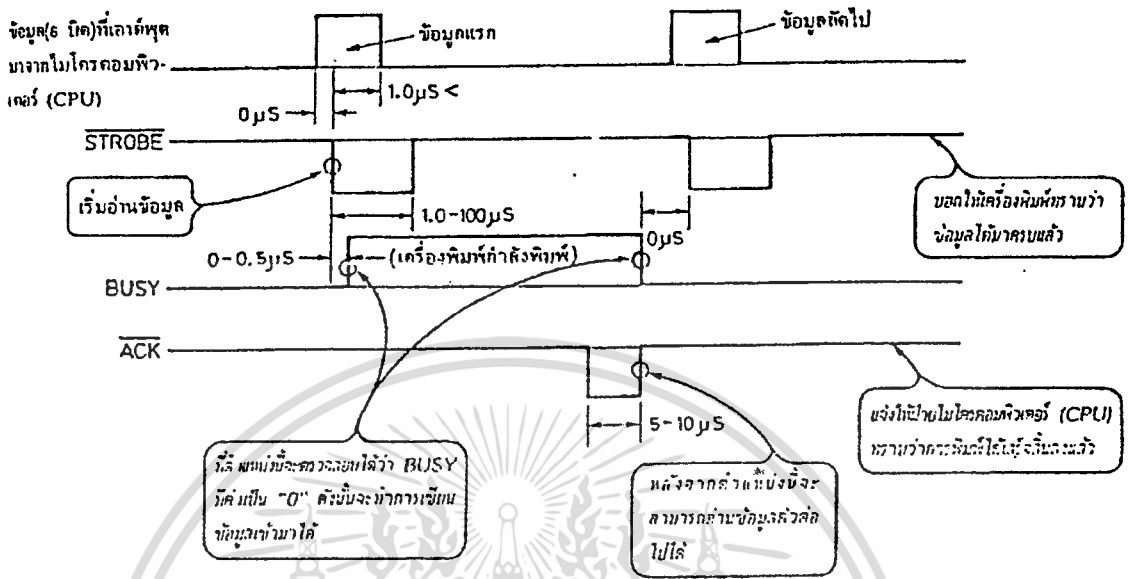
PIN NO. SIGNAL	RETRUN PIN NO.	SIGNAL	C-P	DESCRIPTION
1	19	STROBE	C->P	สัญญาณ STROBE จะถูกส่งออกไปในขณะที่มีการส่งข้อความออกไป ระดับของสัญญาณมีค่า "0"
2	20	DATA 1	C->P	สัญญาณนี้ขึ้นอยู่กับข้อมูลที่ส่งมาตั้งแต่บิตที่ 1 ถึงบิต ที่ 8 ของข้อมูล BYTE นั้น
3	21	DATA 2	C->P	
4	22	DATA 3	C->P	
5	23	DATA 4	C->P	
6	24	DATA 5	C->P	
7	25	DATA 6	C->P	
8	26	DATA 7	C->P	
9	27	DATA 8	C->P	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10	28	ACKNLG	C->P	เป็นสัญญาณในการตอบรับเมื่อได้รับข้อมูลแล้ว โดยสัญญาณมีค่าเป็น "0"
11	29	BUSY	C<-P	สัญญาณมีค่าเป็น "1" แสดงให้รู้ว่าทางด้านรับไม่พร้อมที่จะรับข้อมูล อาจเกิดได้จาก 1. หลังจากรับข้อมูลเข้ามาแล้ว 2. ขณะที่ PRINTER กำลังทำงาน 3. เกิดการ OFF LINE 4. เกิดความผิดพลาดของสถานะขึ้น
12	30	PE	C<-P	สัญญาณ "1" แสดงให้ทราบว่ากระดาษพิมพ์หมด
13	-	SLCT	C<-P	สัญญาณ "1" เป็นการตอบรับการถูกเลือกจากคอมพิวเตอร์
14	-	AUTO FEED XT	-	เมื่อ PRINTER ได้รับสัญญาณนี้มีค่า "0" ทุก ๆ บรรทัดที่พิมพ์เสร็จจะเว้นเพิ่มอีก 1 บรรทัด
15	-	NC	-	สำรองไว้
16	-	OV.	-	มีค่าเป็นกราวด์
17	-	CHASSIS	-	เป็นกราวด์แทน
18	-	NC	-	สำรองไว้
19-30	-	GND	-	TWISTED-PAIR
31	-	INIT	C->P	สัญญาณมีค่าเป็น "0" ทำให้เป็นการเริ่มต้น PRINTER BUFFER และสถานะต่าง ๆ
32	-	ERROR	C<-P	สัญญาณนี้จะมามีค่าเป็น "0" เมื่อ 1. กระดาษหมด 2. OFF LINE 3. เกิด STATUS ERROR ขึ้น
33	-	GND	-	เหมือน 19-30
34	-	NC	-	สำรองไว้
35	-	PWR	C<-P	บอกให้รู้ว่า POWER DOWN ใน PRINTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

36	-	SLCT IN	C->P	สัญญาณมีค่า "0" แสดงว่า PRINTER ถูกเลือก
----	---	---------	------	--



รูปที่ 9 แสดงสัญญาณรับ-ส่งข้อมูล

จาก TIMING DIAGRAM การส่งข้อมูลจาก COMPUTER ไปยัง PRINTER 1 byte ในตอนแรก COMPUTER จะตรวจสอบค่า BUSY จาก PRINTER นั้นเท่ากับ LOGIC "0" ถ้าเป็น LOGIC "0" แสดงว่า PRINTER พร้อมที่จะรับข้อมูลจากนั้น COMPUTER จะส่งข้อมูลมาพร้อมในเวลาที่ไม่น้อยกว่า 0.5 วินาที จะส่งสัญญาณ STROBE เป็น PULSE "0" ด้วยคาบเวลาไม่น้อยกว่า 0.5 วินาที ในช่วงเวลาไม่น้อยกว่า 0.5 วินาที ในช่วงเวลาขอขาสัญญาณ STROBE PRINTER จะส่งสัญญาณ BUSY ให้มี LOGIC เป็น "1" เพื่อแสดงว่า PRINTER ไม่พร้อมที่จะรับข้อมูลต่อในช่วงเวลานี้ PRINTER จะรับเอาข้อมูลจาก COMPUTER ที่ส่งมาในช่วงของขาของสัญญาณ STROBE เก็บส่ง MEMORY BUFFER ภายใน PRINTER เองหรือใช้เวลาในพิมพ์ ข้อมูลนี้สัญญาณ STROBE กลับเป็น Logic "1" อีกครั้งหนึ่งนี้จะต้องรอเวลาไม่น้อยกว่า 0.5 วินาที จึงจะมีการเปลี่ยนแปลงข้อมูลที่ส่งมายัง PRINTER เพื่อให้ PRINTER รับข้อมูลที่ส่งมาอย่างถูกต้องและเมื่อ PRINTER รับข้อมูลเรียบร้อยแล้วและมี MEMORY BUFFER ระหว่างก็จะส่งสัญญาณ BUSY เป็น "0" และส่งสัญญาณการตอบรับข้อมูลที่ส่งมีที่ขา (ACKNOWLEDGE) มี PULSE "0" ด้วยคาบเวลาประมาณ 3-12 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Constanine A. Balanis, "Antenna Theory Analysis and Design," Arizona State University, pp.22-23.
- [2] Demetrius T.Paris,W.marshall Leach,Jr.,and Edward B.Joy "Basic Theory of Probe Compensated Near Field Measurements," IEEE Transactions of Antenna and Propagation, vol.AP-26, No.3, May 1978, pp. 375-377.
- [3] Gary E. Evans, "Antenna Measurement Techniques," Artech House, Boston, London, pp.137-168.
- [4] Yoshiyuki Fujino and Cheuk-yu Edward Tong, "Analysis of an Open - Ended Waveguide as a Probe for Near Field Antenna Measurements by Using TLM Method," IEICE Trans. Commun., vol.E77-B, no.8, August 1994, pp.1048-1055.
- [5] วัฒนา พลพละ, "กราฟิกพล็อตเตอร์," เขมิกอนคักเตอร์ อิเล็กทรอนิกส์,ฉบับที่ 127,พฤษภาคม 2536, หน้า 25-36.
- [6] ไม่นาย ไกรฤกษ์ ,วิวัฒน์ กิรานนท์ , "ทฤษฎีสายอากาศ"
- [7] "Using The HP-IB Interface and Library with Dos," Hewlett Packard.
- [8] "Quick C programming," Hewlett Packard

กิติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ในด้านต่างๆ ขอขอบพระคุณ อาจารย์ โมโนย ไกรฤกษ์ ที่ให้คำแนะนำทั้งทางด้านวิชาการ และทางด้านเทคนิค พี่เอก ผู้ออกแบบเครื่องสแกนเนอร์ พี่ตั๋ย และ พี่คัมพ์ ให้ความสะดวกในการใช้เครื่องมือต่าง ๆ และให้คำแนะนำที่เป็นประโยชน์ คุณนฤมล ปัญญาสกุลวงศ์ ที่มาช่วยพิมพ์งาน รวมไปถึงน้อง ๆ และเพื่อน ๆ ที่ให้ความสนใจในโครงการนี้และคอยถามไถ่ความคืบหน้าอยู่เสมอ หากรายงานฉบับนี้มีข้อผิดพลาดประการใด ทางคณะผู้จัดทำก็ขออภัยมา ณ ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้