



ปีการศึกษา 2537

การเขียนแบบระบบไฟฟ้าและประมาณราคาคด้วยคอมพิวเตอร์
ELECTRICAL DRAWING AND ESTIMATING BY COMPUTER



โดย
นายอนุศิลป์ สุขประทุม
นายพนมพร อาริสาพรหม
นายพรทัศน์ พงศ์ศิกรกำชัย

วัน เดือน ปี... 18 ก.ค 2537
เลขทะเบียน... 034781
เลขเรียกหนังสือ... T 37081 ค3

อาจารย์ที่ปรึกษา
รศ.สุณี บรรจงจิตร
อ.สมชาติ จิวิภากร

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเขียนแบบระบบไฟฟ้าและประมาณราคาคับด้วยคอมพิวเตอร์

ผู้จัดทำ

- 1.นายธนศิลป์ สุขประทุม
- 2.นายพนมพร อภิสรพรหม
- 3.นายพรพัฒน์ หงส์ศักดิ์กรกำจาย

.....อาจารย์ที่ปรึกษา
(รศ. ศุภี บรรจงจิตร)
.....อาจารย์ที่ปรึกษา
(อ. สมชาติ จิรวิภากร)

การเขียนแบบระบบไฟฟ้าและประมาณราคาศัลยกรรมคอมพิวเตอร์

นายอนุศิลป์	ศุภประทุม
นายพนมพร	อาทิตย์พรหม
นายพรพัฒน์	พงศ์ศักดิ์กรำจาย
ร.ศ.ศุณี	บรรจงจิตร อาจารย์ที่ปรึกษา
อ.สมชาติ	จิรวิภากร อาจารย์ที่ปรึกษา
ปีการศึกษา 2537	

บทคัดย่อ

ปัญญานิพนธ์ฉบับนี้เป็นโปรแกรมการเขียนแบบระบบไฟฟ้าและประมาณราคาศัลยกรรมคอมพิวเตอร์ เป็นโปรแกรมที่สามารถใช้ได้บนเครื่องคอมพิวเตอร์ทั่วไป โดยใช้โปรแกรมสำเร็จรูปเข้ามาช่วยคือ ออโตแคด และ ฟอกซ์โปร ในโปรแกรมจะประกอบด้วย 2 ส่วน คือ ส่วนที่ใช้ในการเขียนแบบจะใช้ ออโตแคด เป็นตัวเขียน อีกส่วนหนึ่งคือ ส่วนประเมินราคาจะใช้โปรแกรม ซึ่งเขียนจากฟอกซ์โปร และจะรวมโปรแกรมควบคุมเอาไว้ด้วยกัน และในโปรแกรมจะมีการใช้ สัญลักษณ์และ ข้อมูลเกี่ยวกับอุปกรณ์ไฟฟ้าจำนวนมาก จะเก็บไว้ในรูปแบบพื้นฐานข้อมูลในฟอกซ์โปร และสัญลักษณ์ก็จะเก็บไว้ในแฟ้มรูปภาพของออโตแคด

ELECTRICAL DRAWING AND ESTIMATING BY COMPUTER

Tanusin Sukpratoom

Panompom Apatsraphrom

Pompat Pongdasakomkanjai

Asso. Prof.suree Bunjongjit Advisor

MR.Somohart Jirivipakorn B.Eng

Advisor

1994

ABSTRACT

This Thesis is concerned about Electrical Drawing and Estimating program. This program can be used in generally PC., and Soft ware we used are AutoCad and FoxPro. In this program consisted of 2 part of program, One is the part that used in Electrical Drawing we use AutoCad and the other part is Estimating. Part we use FoxPro. In Estimating part consis of controlling program, database file of a large number of Electrical products., The symbol of Elctrical products we save in drawing file of AutoCad.

III

สารบัญภาพ

	หน้า
รูป 3.1 Screen Menu ของ Auto Cad	30
รูป 3.2 หน้าจอของ Auto Cad เมื่อเรียก Pop up Electrical	39
รูป 3.3 หน้าจอของ Auto Cad เมื่อเรียก Pop up Electrical	40
รูป 4.1 Flow chart การทำงานของโปรแกรม	54
รูป ผ ข.1 รูปหน้าจอของ Auto Cad เมื่อเรียกสัญลักษณ์	2
รูป ผ ข.2 การเพิ่มข้อมูลใหม่	9
รูป ผ ข.3 การเปลี่ยนแปลงราคาอุปกรณ์	10
รูป ผ ข.4 การลบข้อมูล	11



สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญรูป	III
บทที่ 1 บทนำ	1
บทที่ 2 คำสั่งและฟังก์ชันของ FoxPro	
2.1 คำสั่งรับข้อมูลเข้าทางคีย์บอร์ด	3
2.2 คำสั่งในการแสดงผล	4
2.3 คำสั่งกำหนดค่าให้ตัวแปร	5
2.4 คำสั่งสำหรับกระโดดข้าม	5
2.5 คำสั่งสำหรับการทำงานซ้ำ	7
2.6 ฟังก์ชันที่ใช้ในการเขียนโปรแกรม	8
บทที่ 3 การใช้งานโปรแกรม Auto CAD	
3.1 คำสั่งพื้นฐานโปรแกรม Auto CAD	10
3.2 BLOCK	16
3.3 ATTRIBUT	22
3.4 การสร้าง SCREEN MENU	30
3.5 การสร้าง POP UP MENU	37
บทที่ 4 การออกแบบและเขียนโปรแกรม	46
บทที่ 5 การทดลองและผลการทดลอง	55
บทที่ 6 สรุปผลและวิจารณ์	73
ภาคผนวก ก.	
ภาคผนวก ข.	
ภาคผนวก ค.	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

บทที่ 1

บทนำ

1.1 ความจำเป็นและความสำคัญของปัญหา

ในปัจจุบันการออกแบบและการเขียนแบบโดยใช้คอมพิวเตอร์แทนการเขียนด้วยมือมันได้ขยายขอบเขตกว้างขึ้นมากมาย ที่นิยมใช้ก็คือโปรแกรม Auto Cad ที่มีประสิทธิภาพสูงและยังใช้งานร่วมกับซอฟต์แวร์อื่นๆ เพื่อทำรายงานได้

คณะผู้จัดทำจึงเห็นว่าควรจะสร้างซอฟต์แวร์ที่ใช้ในงานเขียนแบบระบบไฟฟ้าโดยตรง โดยการประยุกต์ซอฟต์แวร์ที่มีอยู่แล้วนำมาใช้ร่วมงานกัน เป็นการประหยัดเวลาในการทำงานได้เป็นอย่างมาก

1.2 วัตถุประสงค์ของโครงการ

1.2.1 ประยุกต์การใช้งานร่วมกันระหว่างซอฟต์แวร์ Auto Cad กับซอฟต์แวร์ FoxPro

1.2.2 สามารถนำไปใช้กับเครื่อง PC ทั่วไปได้

1.2.3 ประหยัดเวลาในการทำงาน

1.2.4 เพื่อศึกษาแนวทางการเขียนโปรแกรม และการแก้ไข

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตของ Auto Cad

- เขียนแบบอาคาร โรงงาน และอื่นๆ
- จัดทำ Block อุปกรณ์ทั้งหมด
- เขียนแบบทางไฟฟ้า และลงอุปกรณ์
- สกetchข้อมูลให้กับฐานข้อมูล
- พิมพ์แบบแปลนไฟฟ้า

1.3.2 ขอบเขตของ FoxPro

- เขียนโปรแกรมเมนู และโปรแกรมการคำนวณหาจำนวนดวงโคม
- จัดเก็บข้อมูลต่างๆ ของอุปกรณ์ทั้งหมด
- เขียนโปรแกรมควบคุม
- เชื่อมโยงข้อมูลระหว่าง FoxPro กับ FoxPro และ ระหว่าง FoxPro กับ Auto Cad
- ประเมินราคาจากแบบที่ได้ลงอุปกรณ์ต่างๆ แล้ว
- พิมพ์รายงานการประเมินราคา

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษารายละเอียดการใช้งานและคำสั่งต่างๆ ของซอฟต์แวร์ Auto Cad และ FoxPro

1.4.2 ออกแบบโครงสร้างของซอฟต์แวร์

1.4.3 สร้างภาพสัญลักษณ์ของระบบไฟฟ้า และจัดเก็บไว้เป็น Block

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.4 สร้าง MENU

1.4.5 เขียนซอฟต์แวร์เพื่อการประมาณราคาจากภาพ

1.4.6 จัดทำข้อมูลทางราคาของอุปกรณ์ไฟฟ้า

1.4.7 ทดสอบการใช้งานซอฟต์แวร์

1.4.8 สรุปผลการทำโครงการงาน ข้อเสนอแนะและปัญหาในการทำโครงการงาน

1.4.9 จัดพิมพ์ปริญญานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เป็นการพัฒนาคำความรู้เกี่ยวกับการใช้ซอฟต์แวร์สำเร็จรูป ให้มีประสิทธิภาพมากขึ้น

1.5.2 เพื่อเพิ่มประสิทธิภาพในการออกแบบ และการประมาณราคาให้มากขึ้น ซึ่งจะช่วยให้สะดวกและประหยัดเวลา

1.5.3 เป็นแนวทางในการพัฒนาให้ซอฟต์แวร์มีความสามารถใช้งานอื่นๆ ได้



บทที่ 2

คำสั่งและฟังก์ชันของ FoxPro

ในบทนี้จะเป็นการกล่าวถึงคำสั่งต่างๆ และฟังก์ชันที่ใช้ในการเขียนโปรแกรม ในที่นี้จะแบ่งคำสั่งต่างๆ ที่ใช้ในการเขียนโปรแกรมออกเป็นกลุ่มๆ ดังนี้

2.1 คำสั่งรับข้อมูลเข้าทางคีย์บอร์ด

คำสั่งเหล่านี้ได้แก่

1. ACCEPT

2. INPUT

3. WAIT

4. @.....SAY.....GET

READ

คำสั่ง ACCEPT

ทำหน้าที่รับข้อมูลที่เป็นอักขระเข้าทางจอภาพ เป็นคำสั่งที่ทำการรับข้อมูลชนิดตัวอักษรทางจอภาพเข้าไปยัง <ตัวแปรในหน่วยความจำ> โดยไม่ต้องป้อนเครื่องหมายคำพูด " " ให้กับข้อมูลที่ป้อนเข้ามา หากไม่ได้ทำการทำการกำหนด <ตัวแปรหน่วยความจำ> มาก่อน ก็สามารถใช้คำสั่ง ACCEPT รับข้อมูลเข้าไปได้ ข้อมูลที่นำเข้ามาจะเริ่มทำการในบรรทัดถัดไป และเก็บในรูปแบบข้อมูลชนิดอักขระ

ตัวอย่างเช่น

ACCEPT " กรุณาป้อนชื่อ " to name

ด้วยคำสั่งนี้ข้อความคำว่า " กรุณาป้อนชื่อ " จะปรากฏบนจอภาพ และมีเคอร์เซอร์รอรับข้อความใดๆ ที่จะป้อนเข้ามา หลังจากป้อนข้อความและกด Enter แล้ว FoxPro จะนำข้อความนั้นไปเก็บในตัวแปรที่ชื่อ name ทันที

คำสั่ง INPUT

ทำหน้าที่รับข้อมูลเข้าไปไว้ในตัวแปรหน่วยความจำ โดยทำการรับข้อมูลนำเข้าจากแป้นพิมพ์ตรง <ตัวแปรหน่วยความจำ> หาก <ตัวแปรหน่วยความจำ> ไม่ได้ถูกกำหนดมาก่อน คำสั่ง INPUT ก็จะกำหนดให้ ข้อมูลที่ป้อนจะต้องอยู่ในเครื่องหมายคำพูดเดี่ยว หรือ คำพูดคู่ หรือ [] ก็ได้

ตัวอย่างเช่น

INPUT " เงินเดือนเท่าไร " to salary

INPUT ' เงินเดือนเท่าไร ' to salary

ทั้งสองคำสั่งนี้จะทำงานเหมือนกัน ด้วยคำสั่งนี้ข้อความคำว่า เงินเดือนเท่าไร จะปรากฏบนจอภาพ และมีเคอร์เซอร์รอรับข้อความคือให้พิมพ์เงินเดือนลงไป เมื่อกด Enter เงินเดือนที่พิมพ์จะเก็บไว้ในตัวแปรที่ชื่อ salary ทันที

คำสั่ง WAIT

คำสั่งนี้จะทำการหยุดการปฏิบัติการของโปรแกรมไว้ระยะหนึ่ง และจะทำงานต่อไป เมื่อคุณเป็นใครคนหนึ่ง คำสั่ง WAIT จะแสดงข้อความหรือข้อความที่กำหนด และเก็บค่าเป็นที่ยกไว้เป็นแบบตัวแปรประเภทตัวอักษร

ตัวอย่างเช่น

WAIT " คุณเป็น Y เพื่อทำงานต่อไป." to code

ด้วยคำสั่งนี้ ข้อความคำว่า คุณเป็น Y เพื่อทำงานต่อไป จะปรากฏบนจอภาพ เมื่อคุณเป็น Y โปรแกรมจะทำงานทันทีและค่าที่คีย์จะถูกเก็บไว้ในตัวแปรชื่อ code ถ้าไม่มีการกำหนดข้อความในเครื่องหมายคำพูด หรือใช้คำสั่ง WAIT อย่างเดียว จะปรากฏข้อความดังนี้แทน

Press any key to continue.....

รอให้กดคีย์ใดคีย์หนึ่งเพื่อทำงานต่อไป

คำสั่ง @....SAY....GET

เป็นคำสั่งสำหรับรับข้อมูล ณ ตำแหน่งบนจอภาพที่กำหนดขึ้น ตั้งแต่ 1 ตัวอักษรขึ้นไปที่ย้ายเข้าทางคีย์บอร์ด คำสั่ง READ จะทำให้เคอร์เซอร์ไปหยุดรอรับข้อมูลตำแหน่งที่กำหนด เมื่อกด Enter ข้อมูลจะถูกเก็บไว้ในตัวแปรที่กำหนด โดยตัวแปรจะต้องกำหนดไว้ก่อนใช้คำสั่งนี้ และชนิดของข้อมูลจะเป็นไปตามตัวแปรที่กำหนดไว้

ตัวอย่างเช่น

STORE SPACE(40) TO ADDRESS

@ 5,15 SAY " กรุณาป้อนที่อยู่ปัจจุบัน " GET ADDRESS

READ

ด้วยคำสั่งนี้ที่ตำแหน่งแถวที่ 5 และคอลัมน์ที่ 15 จะปรากฏข้อความคำว่า กรุณาป้อนที่อยู่ปัจจุบัน และมีเคอร์เซอร์รอรับข้อมูลอยู่ที่ท้ายข้อความ ข้อความที่ป้อนเข้าไปจะถูกเก็บไว้ในตัวแปรชื่อ ADDRESS

2.2 คำสั่งในการแสดงผล

การแสดงผลสามารถแสดงได้ทั้งบนจอภาพ และให้พิมพ์ออกที่เครื่องพิมพ์ คำสั่งที่ใช้มีดังนี้

1. ?

2. ??

3. @....SAY

4. SET PRINT ON/OFF

คำสั่ง ?

ในการแสดงผลลัพธ์ หรือข้อมูลของค่าตัวแปร หรือข้อความที่อยู่ในเครื่องหมายคำพูด การแสดงผลจะแสดงในบรรทัดใหม่ถัดจากบรรทัดปัจจุบัน

คำสั่ง ??

ใช้เหมือนกับคำสั่ง ? แต่การแสดงผลจะแสดงในบรรทัดเดียวกับบรรทัดปัจจุบัน

คำสั่ง @..... SAY

ใช้เหมือนกับคำสั่ง ? และคำสั่ง ?? แต่การแสดงผลเราสามารถกำหนดตำแหน่งที่ต้องการแสดงผลได้ตามต้องการ ด้วยการกำหนด @ row,col

คำสั่ง SET PRINT ON/OFF

ถ้ากำหนด SET PRINT ON จะทำให้ผลลัพธ์ที่ต้องการแสดงถูกพิมพ์ออกที่เครื่องพิมพ์ แต่ถ้ากำหนด SET PRINT OFF ผลลัพธ์จะแสดงบนจอภาพตามปกติ

2.3 คำสั่งกำหนดค่าให้ตัวแปร

ตัวแปรเป็นที่เก็บข้อมูลชั่วคราว สำหรับการทำงานตามวัตถุประสงค์อย่างหนึ่ง สามารถนำมาเก็บในแฟ้มฐานข้อมูลหรือนำมาใช้ในการคำนวณได้ คำสั่งที่ใช้มีดังนี้

1. STORE

2. =

คำสั่ง STORE

เป็นคำสั่งกำหนดค่าให้กับตัวแปร เช่น STORE " ABCD" TO X1 เป็นการกำหนดให้ตัวแปร X1 เป็นชนิดตัวอักษรและมีค่าเท่ากับ ABCD

คำสั่ง =

เป็นคำสั่งกำหนดค่าให้กับตัวแปรเช่นกับคำสั่ง STORE เช่น X1 = "ABCD" จะมีผลเช่นเดียวกับคำสั่ง STORE "ABCD" TO X1

2.4 คำสั่งสำหรับการกระโดดข้าม

ในการทำงานของโปรแกรมตามปกติจะทำงานเรียงตามลำดับ โดยเริ่มจากคำสั่งแรกไปจนถึงคำสั่งสุดท้าย ใน FoxPro มีคำสั่งที่ใช้ตัดสินใจเลือกการทำงานว่าจะให้ทำคำสั่งถัดไป หรือให้กระโดดข้ามไปทำงานยังคำสั่งอื่นได้ คำสั่งที่ใช้มี 2 คำสั่ง คือคำสั่ง IF และคำสั่ง DO CASE

คำสั่ง IF

ใช้สำหรับเลือกเงื่อนไขใดเงื่อนไขหนึ่งในสองทางเลือก มี 2 แบบ คือ

IF แบบที่ 1

IF (เงื่อนไข)

คำสั่ง.....(1)

คำสั่ง.....(2)

คำสั่ง.....(3)

.....

ENDIF

คำสั่ง.....(4)

.....

ถ้าเงื่อนไขที่อยู่หลัง IF เป็นจริง คำสั่งที่อยู่ระหว่าง IF และ ENDF จะถูกทำงาน คือ คำสั่งที่ 1 ถึง คำสั่งที่ 3 และต่อไป

ถ้าเงื่อนไขที่อยู่หลัง IF เป็นเท็จ โปรแกรมจะทำคำสั่งที่อยู่ต่อจาก ENDF ในที่นี้ คือ คำสั่งที่ 4 และคำสั่งต่อไป

IF แบบที่ 2

IF (เงื่อนไข)

คำสั่ง.....(1)

คำสั่ง.....(2)

.....

ELSE

คำสั่ง.....(3)

คำสั่ง.....(4)

.....

ENDIF

ถ้าเงื่อนไขเป็นจริง คำสั่งที่อยู่ระหว่าง IF และ ELSE จะถูกทำงาน ในที่นี้คือ คำสั่งที่ 1 และคำสั่งที่ 2 และต่อไป เมื่อทำงานเสร็จแล้วจะกระโดดไปทำงานที่คำสั่งที่อยู่ต่อจาก ENDF

ถ้าเงื่อนไขไม่เป็นจริง คำสั่งที่อยู่ระหว่าง ELSE และ ENDF จะทำงาน ในที่นี้คือ คำสั่งที่ 3 และคำสั่งที่ 4 และคำสั่งต่อไป

คำสั่ง DO CASE

เป็นคำสั่งที่ใช้เลือกสำหรับทำงานอย่างใดอย่างหนึ่ง ในกรณีที่เงื่อนไขทางเลือกมีหลายทาง โดยมีรูปแบบดังนี้ คือ

DO CASE

CASE (เงื่อนไขที่ 1)

คำสั่ง.....

คำสั่งที่ถูกทำงานเมื่อเงื่อนไขที่ 1 เป็นจริง

CASE (เงื่อนไขที่ 2)

คำสั่ง.....

คำสั่งที่ถูกทำงานเมื่อเงื่อนไขที่ 2 เป็นจริง

CASE (เงื่อนไขที่ 3)

คำสั่ง.....

คำสั่งที่ถูกต้องทำงานเมื่อเงื่อนไขที่ 3 เป็นจริง

OTHERWISE

คำสั่ง.....

คำสั่งที่ถูกต้องทำงานเมื่อเงื่อนไขทั้งหมดไม่เป็นจริง

ENDCASE

ถ้าเงื่อนไขที่ 1 เป็นจริง คำสั่งที่อยู่ระหว่าง CASE 1 และ CASE 2 จะถูกทำงาน ถ้าเงื่อนไขที่ 2 เป็นจริง คำสั่งที่อยู่ระหว่าง CASE 2 และ CASE 3 จะถูกทำงาน เป็นต้นและต่อไปเรื่อยๆ OTHERWISE จะกำหนดหรือไม่กำหนดก็ได้ ถ้ากำหนดหมายความว่า ถ้าเงื่อนไขทั้งหมดไม่เป็นจริงคำสั่งที่อยู่ระหว่าง OTHERWISE และ ENDCASE จะถูกทำงาน

2.5 คำสั่งสำหรับการทำงานซ้ำ

คำสั่งสำหรับการกระโดดข้ามเป็นการข้ามจากต้นโปรแกรมไปท้ายโปรแกรมตามไม่สามารถข้ามจากท้ายโปรแกรมไปยังต้นโปรแกรมได้ ดังนั้นการทำงานจึงทำรอบเดียวจบ ในการทำงานจริงเรามักต้องการให้กลับไปทำงานซ้ำหลายๆ รอบจนกว่าจะครบตามเงื่อนไขที่กำหนด

คำสั่งที่ใช้ในการทำงานซ้ำและคำสั่งที่เกี่ยวข้อง มีดังนี้

1. DO WHILE

ENDDO

2. LOOP

3. EXIT

4. RETURN

5. CANCEL

คำสั่ง DO WHILE ENDDO

เป็นคำสั่งให้ทำงานซ้ำครบเท่าที่เงื่อนไขยังเป็นจริง มีรูปแบบดังนี้

DO WHILE (เงื่อนไข)

คำสั่ง.....

คำสั่ง.....

คำสั่ง.....

ENDDO

คำสั่งที่อยู่ระหว่าง DO WHILE และ ENDDO จะถูกทำงานจนกว่าเงื่อนไขที่อยู่หลัง DO WHILE จะเป็นเท็จ

เงื่อนไขที่ใช้ทดสอบมีได้หลายแบบคือ

1. เงื่อนไขทดสอบค่าตรรก เช่น .T. มีผลให้การทำงานรอบไม่สิ้นสุดต้องสร้างเงื่อนไขภายในเพื่อให้ทดสอบให้ออกจากวนรอบ
2. เงื่อนไขให้นับจำนวนรอบ เช่น ให้ทำงานซ้ำ 10 รอบ แล้วสิ้นสุดการทำงาน
3. เงื่อนไขบนพื้นฐานข้อมูล เช่น EOF () เป็นการทดสอบให้ทำงานจนหมดบรรทัดสุดท้ายในแฟ้มฐานข้อมูล

คำสั่ง LOOP

เป็นคำสั่งให้วนกลับไปเริ่มต้นทำงานที่คำสั่ง DO WHILE ทันที คำสั่งนี้ต้องเขียนอยู่ภายในระหว่าง DO WHILE กับ ENDDO

คำสั่ง EXIT

เป็นคำสั่งให้ออกจากการวนรอบ เมื่อโปรแกรมทำงานถึงคำสั่งนี้จะทำให้กระโดดข้ามไปทำงานที่คำสั่งถัดจากคำสั่ง ENDDO ทันที

คำสั่ง RETURN

เป็นคำสั่งให้จบการทำงานของโปรแกรมนั้นแล้วกลับไปทำงานยังคำสั่ง ที่ตั้งให้โปรแกรมนีทำงาน

คำสั่ง CANCEL

เป็นคำสั่งให้จบการทำงานของโปรแกรมนั้น แล้วกลับสู่ Dos Prompt

2.6 ฟังก์ชันที่ใช้ในการเขียนโปรแกรม

1. ฟังก์ชัน INKEY ()
2. ฟังก์ชัน CHR ()
3. ฟังก์ชัน STR ()

1. ฟังก์ชัน INKEY ()

ฟังก์ชันนี้ จะส่งค่าตัวเลขจำนวนเต็มตามตารางรหัสเอตกีที่ป้อนเข้ามาทางแป้นพิมพ์ โดยมีค่าอยู่ระหว่าง 0 ถึง 255

หากไม่มีการกดแป้นใดๆ ก็จะได้ค่ากลับให้เป็นค่าศูนย์ และถ้ามีการกดแป้นพิมพ์มาในที่พักข้อมูลหลายตัว ค่าของแป้นพิมพ์ที่ป้อนเข้ามาในที่พักข้อมูลเป็นแรกจะเป็นค่าที่ถูกส่งไป ด้วยวิธีการเช่นนี้ จะทำให้สามารถเขียนโปรแกรมกำหนดแป้นพิมพ์ที่ต้องการให้ใช้งานได้

2. ฟังก์ชัน CHR ()

รูปแบบ CHR (< ค่าตัวเลขตามรหัสเอตกี>)

ฟังก์ชันนี้จะทำการประเมินค่าใน <ค่าตัวเลขตามรหัสเอตกี> และส่งค่ากลับเป็นอักขระตามค่าในตารางรหัสเอตกี



ตัวอย่าง เช่น

? CHR (65)

จะได้ A เพราะ A มีค่าเท่ากับ 65 ในตารางรหัสเอตกี

? CHR (97)

จะได้ a เพราะ a มีค่าเท่ากับ 97 ในตารางรหัสเอตกี

3. ฟังก์ชัน STR ()

รูปแบบ STR (<นิพจน์1> , <นิพจน์2> , <นิพจน์3>)

ฟังก์ชันนี้ จะห้แปลงนิพจน์จำนวนตัวเลข ให้เป็นนิพจน์อักขระ

<นิพจน์1> จะถูกแปลงให้เป็นอักขระ โดยความยาวของอักขระจะถูกกำหนดด้วยตัวเลขที่ระบุใน <นิพจน์2> และตัวเลขที่ระบุใน <นิพจน์3> จะเป็นจำนวนตำแหน่งทศนิยมถ้าจำนวนตัวเลขใน <นิพจน์2>มากกว่าจำนวนตัวเลขใน <นิพจน์1> ส่วนที่ขาดจะถูกแทนด้วยช่องว่างที่ด้านหน้า ตัวอย่างเช่น

? STR (123.456,10,4)

จะได้ __123.4560

เครื่องหมาย _ หน้าตัวเลข 2 ชิด แสดงว่ามีช่องว่างหน้าตัวเลข 2 ช่องว่าง

ถ้า <นิพจน์2> มีจำนวนตัวเลขน้อยกว่า <นิพจน์1> ฟังก์ชันSTR () นี้จะคิดจำนวนตัวเลขและจุดทศนิยมรวมกันโดยให้เท่ากับ <นิพจน์2> โดยใช้เลขจำนวนเต็มเป็นหลัก ส่วนที่ทศนิยมจะตัดทิ้ง

ตัวอย่างเช่น

? STR (1234567890.12345,13,4)

จะได้ 1234567890.12

บทที่ 8

การใช้งานโปรแกรม AUTOCAD

3.1 คำสั่งพื้นฐานโปรแกรม AUTOCAD

คำสั่ง QUIT

วัตถุประสงค์	ออกจาก program autocad
ทางเลือก	save change เก็บข้อมูลลงดิสก์ก่อนออกจาก program discard change ออกจาก program โดยไม่เก็บข้อมูลลงดิสก์ cancel command ยกเลิกคำสั่ง

คำสั่ง SAVE

วัตถุประสงค์	เก็บข้อมูลลงดิสก์โดยไม่ออกจาก program autocad
ทางเลือก	ระบุชื่อไฟล์ที่ต้องการเก็บข้อมูล

คำสั่ง LINE

วัตถุประสงค์	เขียนเส้นตรง
ทางเลือก	<point> - กำหนดจุดที่จะลากเส้นไปถึง U <Undo> - ยกเลิกเส้นส่วนสุดท้ายที่เขียน C <Close> - ลากเส้นจากจุดสุดท้ายไปยังจุดเริ่มต้น

คำสั่ง CIRCLE

วัตถุประสงค์	เขียนวงกลม
ทางเลือก	center point & radius หรือ diameter - กำหนดจุดศูนย์กลางและรัศมีหรือเส้นผ่านศูนย์กลาง 2p - กำหนดจุด 2 จุดบนปลายเส้นผ่านศูนย์กลาง 3p - กำหนดจุด 3 จุดบนเส้นรอบวง TTR - กำหนดวัตถุที่จะให้วงกลมไปสัมผัส 2 จุดและรัศมีของวงกลมที่จะเขียน

คำสั่ง ARC

วัตถุประสงค์	เขียนส่วนโค้งของวงกลม
ทางเลือก	ต้องให้ข้อมูล 3 ตัวในการเขียน โดยมีการเขียนหลายวิธีดังนี้ - จุดเริ่มต้น จุดกลาง และจุดปลายของส่วนโค้ง (start, second, end) - จุดเริ่มต้น จุดศูนย์กลางของวงโค้งและจุดปลาย (start, center, end) - จุดเริ่มต้น จุดศูนย์กลางของวงโค้ง และมุมภายในของส่วนโค้ง (included angle) - จุดเริ่มต้น จุดศูนย์กลางทางโค้ง และความยาว chord (length of chord) - จุดเริ่มต้น จุดปลาย และรัศมีของส่วนโค้ง (radius)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จุดเริ่มต้น จุดปลาย และมุมภายในของส่วนโค้ง
- จุดเริ่มต้น จุดปลาย และทิศทางเริ่มต้นของส่วนโค้ง (direction of arc)

คำสั่ง TRACE

- วัตถุประสงค์ เขียนเส้นตรงที่กำหนดความหนาของเส้นได้
- ทางเลือก Trace width - กำหนดความกว้างของเส้นได้
- <point> - หลังจากกำหนดความหนาแล้วจะถามถึงจุดที่จะลากเส้น เหมือนคำสั่ง LINE แต่จะเขียนเส้นในแต่ละส่วนหลังจากที่กำหนดตำแหน่งของส่วนที่ติดไปแล้ว เพราะ โปรแกรมจะรอเพื่อปากกามองเส้น (เส้นมีความหนา) เพื่อให้สัมพันธ์กับตำแหน่งถัดไป

คำสั่ง TEXT

- วัตถุประสงค์ เขียนตัวอักษรในตำแหน่งที่กำหนด
- ทางเลือก <point> - กำหนดจุดเริ่มต้นของข้อความ
 A <Align> - กำหนดขอบเขตที่จะเขียนข้อความ
 C <Center> - กำหนดจุดกึ่งกลางของข้อความ
 F <Fit> - กำหนดขอบเขตที่จะเขียนข้อความ และความสูงของอักษร
 M <Middle> - กำหนดจุดกึ่งกลางของข้อความ (กึ่งความสูงด้วย)
 R <Right> - กำหนดจุดปลายของข้อความ (right justify)
 S <Style> - เลือก style ของอักษรที่จะใช้

คำสั่ง DTEXT

- วัตถุประสงค์ เขียนอักษรในตำแหน่งที่กำหนด โดยเขียนอักษรตามการกหนดเป็นพินท์ในทันที
- ทางเลือก เหมือนคำสั่ง TEXT

คำสั่ง ERASE

- วัตถุประสงค์ ลบวัตถุที่เลือกทิ้งไป

คำสั่ง MOVE

- วัตถุประสงค์ เปลี่ยนย้ายตำแหน่งของวัตถุที่ระบุ

คำสั่ง COPY

- วัตถุประสงค์ ลอกแบบวัตถุที่ระบุไปในตำแหน่งที่ต้องการ
- ทางเลือก <point> - กำหนด base point
 M <Multiple> - กำหนดการลอกแบบชนิดซ้ำหลายครั้ง

คำสั่ง ROTATE

- วัตถุประสงค์ หมุนวัตถุที่ระบุโดยรอบจุดหมุนที่กำหนด

ทางเลือก หลังจากกำหนดจุดหมุนแล้ว

<angle> - ระบุมุมที่ต้องการให้หมุนไปจากตำแหน่งเดิม

R <Reference> - กำหนดตำแหน่งเดิมของวัตถุและมุมใหม่ที่ต้องการ

คำสั่ง ARRAY

วัตถุประสงค์ การลอกแบบซ้ำ โดยมีตำแหน่งในการลอกแบบเป็นไปอย่างมีระเบียบ

ทางเลือก R <Rectangular> - แบบเป็นแถวและคอลัมน์ (row & column) โดยกำหนดจำนวนแถว คอลัมน์ และระยะห่าง ซึ่งหากค่าระยะเป็นบวก การ copy จะ copy ไปทางขวาของ วัตถุ(สำหรับคอลัมน์)และขึ้นข้างบน (สำหรับแถว) และกระทำตรงกันข้ามหากใส่ค่าลบ

P <Polar> - แบบวงกลม

คำสั่ง CHANGE

วัตถุประสงค์ เปลี่ยนแปลงคุณสมบัติ หรือตำแหน่งบางส่วนของวัตถุ

ทางเลือก <point> - จุดที่ต้องการให้วัตถุที่เลือกเปลี่ยนตำแหน่ง (บางส่วน) ไปยังจุดนั้น

P <Properties> - การเปลี่ยนแปลงคุณสมบัติของวัตถุดังนี้

C <Color> - สีของวัตถุ

B <Elevation> - ระดับที่เขียนวัตถุนั้นๆ

LA <Layer> - layer ที่เก็บวัตถุนั้น

LT <Ltype> - ชนิดของลายเส้นที่ใช้

T <Thickness> - ความหนา หรือความสูงของวัตถุ

คำสั่ง TRIM

วัตถุประสงค์ ใช้สำหรับตัดส่วนเกินของวัตถุที่ไม่ต้องการออกโดยมีขอบเขต(edge) ของการตัดออกที่แน่นอน

ทางเลือก <Cutting Edge> - ขอบเขตที่เป็นตัวกันเขตการตัดวัตถุ

<Select Object> - วัตถุที่จะถูกตัดออกตามขอบเขตที่กำหนด โดยส่วนที่ถูกทิ้งจะเป็น ส่วนที่ถูกออก

คำสั่ง OFFSET

วัตถุประสงค์ สร้างเส้นขนานกับวัตถุที่ระบุ ด้วยระยะห่างที่กำหนด

ทางเลือก <Distance> - ระยะห่างที่ต้องการ

<Through> - ต้องการระบุระยะห่างโดยการชี้ด้วยเคอร์เซอร์

คำสั่ง EXPLODE

วัตถุประสงค์ ใช้สำหรับแยกวัตถุประเภท block หรือ polyline ให้กลายเป็นส่วนของภาพของ line หรือ arc

- การแตกตัวจะทำให้ได้ระดับเท่านั้น ยกเว้นเช่น polyline ที่ทำเป็นblock เมื่อใช้คำสั่งนี้ block จะแตกตัวออกเป็น polyline และเมื่อใช้คำสั่งนี้อีกครั้ง polyline นั้นจึงกลายเป็น line หรือ arc
- ข้อมูลของความหนาและ length ของ polyline จะถูกยกเลิก
- block ที่ insert เข้ามาโดยมีการเปลี่ยนแปลง scale จะใช้คำสั่งนี้ไม่ได้

คำสั่ง ZOOM

- วัตถุประสงค์** การขยายหรือย่อภาพด้วยที่แสดงตามที่ระบุ
- ทางเลือก** A <All> - ดูพื้นที่ที่เขียนทั้งหมด (เท่า limits หรือเท่าขนาดของชิ้นงานที่เขียนแล้วแต่อย่างไหนจะมากกว่า)
- E <Extent> - ขยาย หรือย่อชิ้นงานทั้งหมดให้เต็มจอภาพ
- W <Window> - ขยายบริเวณที่กำหนดออกมาเต็มจอภาพ
- C <Center> - กำหนดจุดกึ่งกลางของจอภาพใหม่และความสูงโดยบอกจำนวน drawing unit หรือจำนวนเท่าของความเปลี่ยนแปลง (<n>x)
- L <Left> - กำหนดจุดมุมล่างซ้ายของจอภาพใหม่และสูงโดยบอกจำนวน drawing unit หรือ จำนวนเท่าของการเปลี่ยนแปลง (<n>x)
- P <Previous> - เรียกภาพที่เคยแสดงมาก่อนย้อนหลังในทีละภาพ (เก็บไว้ได้ 5 ภาพย้อนหลัง)
- D <Dynamic> - <Version 2.5> คล้าย zoom window แต่จะแสดงภาพทั้งหมดออกมา และให้กำหนด frame ที่ต้องการให้ แสดงออกมาเต็มจอภาพ
- <n>x - จำนวนเท่าของการเปลี่ยนแปลงจากภาพที่แสดงอยู่

คำสั่ง REDRAW

- วัตถุประสงค์** สำหรับล้างเครื่องหมายกากบาท(blipmode)ที่เกิดจากการกำหนดตำแหน่ง

คำสั่ง BLOCK

- วัตถุประสงค์** เก็บ drawing ที่ระบุในลักษณะของ template เพื่อความสะดวกในการเรียกใช้ภายหลัง ไม่ว่าวัตถุที่เลือกจะมีกี่ชิ้นส่วนก็ตาม สร้างให้เป็น block แล้วจะถือว่าเป็นวัตถุชิ้นเดียวกัน และโดยปกติจะแก้ไขไม่ได้ ยกเว้นในกรณีที่ เรียก block นั้นกลับมาจะกลับมาจะใส่เครื่องหมายดอกจัน (*) นำหน้าชื่อ block ที่ระบุ หรือการใช้คำสั่ง EXPLODE
- ทางเลือก** ? - ดูรายชื่อของ block ที่มีอยู่แล้วใน drawing นั้นๆ
- <name> - ระบุชื่อ block ที่จะดึงชื่อให้วัตถุที่เลือก

คำสั่ง INSERT

วัตถุประสงค์ การเรียก block ที่เคยสร้างไว้มาใช้

ทางเลือก ? - ดูรายชื่อของ block ที่มีอยู่แล้วใน drawing นั้นๆ

<name> - รายชื่อ block ที่จะดึงชื่อให้วัตถุที่เลือก

<name> - กำหนดให้มีการเรียกข้อมูลของ block นั้นๆ เข้ามาใหม่ ในกรณีที่มีการแก้ไข block นั้นๆ block ที่มีอยู่แล้วจะยังไม่แก้ไข จนกว่าจะใช้วิธีนี้

คำสั่ง WBLOCK

วัตถุประสงค์ เหมือนคำสั่ง BLOCK แต่จะเก็บข้อมูลของ block นั้นลงดิสก์ด้วย โดยตามชื่อไฟล์ที่จะเก็บข้อมูลเหล่านี้ไว้

ทางเลือก หลังจากกำหนดชื่อไฟล์แล้ว จะถามถึงชื่อ block โดยมีทางเลือกดังนี้

<name> - นำเอาข้อมูลของ block ที่สร้างไว้ ไปเก็บไว้ในไฟล์ที่ระบุ

- นำเอาข้อมูลทาง block ที่มีชื่อเหมือนชื่อไฟล์ไปเก็บไว้

- เป็นการเก็บข้อมูลของ drawing ทั้งหมดเหมือนคำสั่ง SAVE ยกเว้นแต่ข้อมูลของ block ที่ไม่มีการใช้งาน drawing

<enter> - เพื่อเลือกวัตถุที่ต้องการเก็บเป็น block และเก็บข้อมูลเหล่านั้นลงดิสก์ในชื่อไฟล์ที่ระบุ

คำสั่ง ATTDEF (Attribute Define)

วัตถุประสงค์ สร้างรูปแบบของข้อมูลที่เป็น text เพื่อประกอบกับวัตถุที่จะสร้างเป็น block และเพื่อความสะดวกในการรวบรวม แยกแยะจัดระเบียบข้อมูลของ block ที่มีการใส่ลงไปใน drawing ได้สะดวกขึ้น

ทางเลือก ลักษณะของข้อมูลมีทางเลือกดังนี้

I<Invisible> - กำหนดให้แสดงข้อมูล หรือไม่แสดง

C<Constant> - กำหนดข้อมูลเป็นค่าคงที่ตายตัว

V<Verity> - กำหนดให้มีการถามย้ำถึงข้อมูลที่ใส่เข้าไป

คำสั่ง ATTDISP (Attribute Display)

วัตถุประสงค์ การกำหนดสถานะการแสดงผลประเภท attribute

ทางเลือก N <Normal> - ตามที่กำหนดในขณะ define (invisible or not)

ON - กำหนดให้แสดงข้อมูลออกมา

OFF - กำหนดให้ไม่แสดงข้อมูลออกมา

คำสั่ง ATTEDIT (Attribute Edit)

วัตถุประสงค์ แก้ไขข้อมูลต่างๆ ของ attribute ที่ระบุ

ทางเลือก

วิธีการแก้ไขข้อมูลทำดังนี้

- แก้ไขเฉพาะ attribute ที่เลือกทีละตัว
- แก้ไขข้อมูลของ attribute ที่ระบุ พร้อมกันทุกตัวที่มีอยู่ใน drawing

การแก้ไขข้อมูลมีทางเลือกดังนี้

- V<Value> - แก้ไขตัวข้อมูลที่บรรจุอยู่
- P<Position> - แก้ไขตำแหน่งที่แสดง attribute นั้นๆ
- H<Height> - แก้ไขความสูงของตัวอักษร
- S<Style> - แก้ไข style ของตัวอักษร
- L<Layer> - แก้ไข layer ที่บรรจุ Attribute นั้นๆ
- C<color> - แก้ไขสีของตัวอักษร
- N<Next> - เปลี่ยนไปยังข้อมูลตัวต่อไปของ attribute ที่เลือกมา แก้ไข

คำสั่ง ATTEXT (Attribute Text)

วัตถุประสงค์

สกัดเฉพาะข้อมูลของ attribute ไปเก็บไว้ในไฟล์ที่ระบุ

ทางเลือก

รูปแบบของข้อมูลที่สกัดออกไปได้ดังนี้

- C<CDF> - ข้อมูลแบบคั่นด้วยเครื่องหมาย comma (,)
- S<SDF> - ข้อมูลที่กำหนดความยาวของข้อมูลแต่ละตัวไว้อย่างแน่นอน
ส่วนมากโปรแกรมด้านฐานข้อมูล (database) จะอ่านรูปแบบนี้ได้
- D<DXF> - เป็นรูปแบบของโปรแกรมเชิงเรียกว่า drawing interchange file format
- E<Entity> - กำหนด Attribute ที่ต้องการสกัดข้อมูลไป
- Template File - เป็นไฟล์ที่เก็บรูปแบบ สำหรับการสกัดข้อมูล SDF, CDF
เพื่อให้โปรแกรมจะได้ใช้เป็นตัวอย่างในการสกัดข้อมูลออกไป ในรูปแบบที่ต้องการ (template file เป็นไฟล์ประเภท text file)
- Extract File - เป็นไฟล์ที่เก็บข้อมูลที่ถูกสกัดออกไป

คำสั่ง DIM

วัตถุประสงค์

เข้าสู่ mode ของการให้เส้นวิเคราะแบบต่างๆ

ทางเลือก

การวิเคราะห์ทางตรงมีทางเลือกนี้

- HOR<Horizontal> - ให้เส้นวิเคราะห์แนวนอน
- VER<Vertical> - ให้เส้นวิเคราะห์แนวตั้ง
- ALT<Aligned> - ให้เส้นวิเคราะห์ตามแนวของวัตถุที่วัด
- ROT<Rotated> - ให้เส้นวิเคราะห์ตามแนวที่กำหนด

- BAS<Baseline>** - การให้เส้นวัดระยะต่อเนื่องกับระยะที่ผ่านมา โดยวัดระยะจากจุดเริ่มต้นของเส้นวัดระยะที่ผ่านมา
- CON<Continue>** - การให้เส้นวัดระยะต่อเนื่องกับระยะที่ผ่านมา โดยวัดระยะต่อจากจุดปลายของเส้นวัดระยะที่ผ่านมา

การวัดระยะกับส่วนโค้งมีทางเลือกดังนี้

- ANG<Angular>** - วัดมุมระหว่างเส้น 2 เส้นที่ระบุ
- DIA<Diameter>** - วัดเส้นผ่านศูนย์กลางของวงกลมหรือส่วนโค้งที่ระบุ
- RAD<Radius>** - วัดรัศมีของวงกลมหรือส่วนโค้งที่ระบุ

คำสั่งอำนวยความสะดวกต่างๆ มีดังนี้

- CEN<Center>** - เขียนเครื่องหมายกากบาทที่จุดศูนย์กลางของวงกลมหรือส่วนโค้ง
- LEA<Leader>** - เขียนเส้นตรงเหมือนคำสั่ง LINE แต่จุดเริ่มต้นของเส้นจะถูกตรึงติดอยู่ด้วย
- EXIT** - ออกจาก Mode ของการให้ dimension
- REDRAW** - เหมือนคำสั่ง REDRAW ปกติ
- STATUS** - แสดงค่าปัจจุบันของตัวเลือกต่างๆ ที่ใช้ในการให้ dimension
- STYLE** - เปลี่ยน style ของอักษรที่นำมา
- UNDO** - ยกเลิกการให้ระยะครั้งที่ผ่านมา ย้อนหลังไปที่ละคำสั่ง

3.2 BLOCK

การใช้ BLOCK

คำสั่ง BLOCK เป็นคำสั่งหนึ่งในโปรแกรม AUTOCAD มีหน้าที่ในการรวบรวมชิ้นส่วน ที่กำหนดให้เป็นชิ้นส่วนเดียวกันและกำหนดชื่อให้กับชิ้นส่วนเหล่านั้น ในภายหลังหวะจะใช้ชื่อเหล่านั้นในการอ้างอิง เพื่อเรียกรูปแบบของชิ้นส่วนที่เคยเก็บเอาไว้ แล้วนำมาใช้ในบริเวณต่างๆ ตามความต้องการ นอกจากนี้เรายังสามารถกำหนดขนาดมาตราส่วนของชิ้นวัตถุเหล่านั้นทั้งในแกน X และแกน Y รวมทั้งมุมหมุนของชิ้นงานด้วย

ชิ้นวัตถุใดๆ ก็ตามไม่ว่าจะเป็นชิ้นเดียวหรือหลายชิ้น เมื่อถูกกำหนดให้เป็น BLOCK แล้วโปรแกรมจะมองเห็นชิ้นวัตถุเหล่านั้นเป็นเหมือนวัตถุ 1 ชิ้นเสมอ เราสามารถที่จะลบ ย้าย หรือลอกแบบได้โดยง่าย การเลือกชิ้นวัตถุเหล่านี้เพื่อการแก้ไขเปลี่ยนใดๆ จะเลือกได้โดยการเข้าไปที่ส่วนส่วนใดก็ได้ของกลุ่มวัตถุจะเท่ากับเป็นการเลือกกลุ่มวัตถุนั้นทั้งกลุ่ม

ประโยชน์ของการใช้ BLOCK

การใช้ BLOCK มีประโยชน์ 4 ประการดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การรวบรวมชิ้นงานหลายๆ ชิ้นเข้าด้วยกัน เนื่องจากการเขียนแบบด้วยคอมพิวเตอร์ จะเขียนในลักษณะของ FULL SCALE เพื่อความสะดวกในการทำงานและใช้ประโยชน์จากคำสั่งช่วยเหลือน้อยๆ ภายในโปรแกรมได้ง่าย โดยปกติแล้วแผ่นงานแต่ละแผ่นจะประกอบไปด้วยหลายๆ ชิ้นงานที่มีมาตรฐานต่างกัน ดังนั้นวิธีการทำงานจึงเป็นการเขียนชิ้นงานแต่ละชิ้นแยกจากกัน และดึงเข้ามาในแผ่นงานในรูปแบบของ BLOCK ซึ่งจะทำให้เราสามารถปรับขนาดมาตรฐานของชิ้นงานที่ต้องการได้โดยการปรับค่า SCALE FACTOR นอกจากนี้วัตถุที่เป็น BLOCK ยังสะดวกในการเคลื่อนย้ายเพราะโปรแกรมจะถือเป็นชิ้นเดียวกันในการจัดระเบียบแบบแผนหน้ากระดาษจึงทำได้ง่ายกว่า

2. ลดการทำงานลงได้มาก เราสามารถเก็บชิ้นส่วนที่เป็นรูปแบบมาตรฐานต่างๆ หรือชิ้นส่วนที่เราคาดว่า จะมีการใช้งานหลายครั้ง เอาไว้ใช้งานในภายหลังได้ทันทีที่จะต้องเขียนใหม่เสียทุกครั้ง เช่น โด๊ส เก้าอี้ โซฟา ชิ้นส่วนอุปกรณ์ รายละเอียดครอซคอต ถังดิน วิธีการแบบนี้จะทำให้เราลดงานเขียนลงไปได้มากซึ่งมีการเขียนที่เราจะได้ชิ้นส่วนมาตรฐานมากขึ้นตามลำดับ และการเขียนแบบของเราในครั้งต่อไปก็จะเร็วขึ้นด้วย

3. ความสะดวกในการแก้ไข เปลี่ยนแปลง ในการทำงานเรายังจะต้องมีการแก้ไขปรับปรุงรายละเอียดของชิ้นส่วนต่างๆ หากมีชิ้นส่วนที่เหมือนกันหลายๆ ชิ้น ในแบบเราต้องตามไปแก้ไขทุกๆ จุด ด้วย แต่ถ้าหากว่าเรากำหนดชิ้นส่วนเหล่านั้นให้เป็น BLOCK แล้ว เราสามารถที่จะแก้ไขกับต้นฉบับเพียงตัวเดียว แล้วส่วนที่เหลือโปรแกรมจะทำการปรับปรุงให้หมดไม่ว่าจะมีการใช้ BLOCK ซี่งนี้ในที่ใดก็ตาม เช่น โด๊ส ขนาด 50 X 100 เซนติเมตร ต้องแก้ไขขนาดเป็น 75 X 150 เซนติเมตร เราก็เพียงแต่แก้ไขให้เป็นขนาดของโด้สที่ต้องการและกำหนด BLOCK นั้นขึ้นมาใหม่ (REDEFINED) ในชื่อเดิมซึ่งจะมีผลทำให้ในจุดใดก็ตามที่มีการเขียนโด้สตัวนี้ไว้จะถูกแก้ไขเป็น ขนาด 75 X 150 เซนติเมตร ทั้งหมดโดยอัตโนมัติ

4. การประหยัดเนื้อที่หน่วยความจำ การใช้ BLOCK จะช่วยลดเนื้อที่หน่วยความจำลงไปได้มากโดยเฉพาะหากมีการใช้ BLOCK นั้น ๆ ซ้ำกันมากเพราะวัตถุที่เก็บเป็น BLOCK จะใช้เนื้อที่สำหรับเก็บข้อมูลชิ้นส่วนภายใน BLOCK นั้น ๆ เพียงจุดเดียว หลังจากนั้นเมื่อการเรียกใช้ Block แต่ละครั้ง จะใช้หน่วยความจำเพิ่มเติมเฉพาะสำหรับเก็บชื่อส่วนนั้นๆ และตำแหน่งที่ BLOCK นั้นๆ วางอยู่ เท่านั้น ซึ่งหากเป็นการเขียนซ้ำๆ กัน โดยการใช้คำสั่ง COPY แทนการใช้ BLOCK แล้วก็จะใช้เนื้อที่ในการเก็บข้อมูลเต็มเท่ากับจำนวนชิ้นส่วนที่เขียนลงไป ยังมีชิ้นส่วนประกอบกันมากเท่าใดการใช้ BLOCK ก็จะช่วยลดเนื้อที่ที่ใช้เก็บข้อมูลลงได้มากในทุกครั้งที่มีการเรียกใช้ BLOCK

วิธีการสร้าง BLOCK

มีคำสั่งที่ใช้ในการสร้าง BLOCK อยู่ 2 คำสั่งด้วยกัน คือ คำสั่ง BLOCK และคำสั่ง WBLOCK โดยมีข้อแตกต่างกันคือ คำสั่ง BLOCK นั้นสร้าง BLOCK และเก็บข้อมูลเหล่านั้นไว้ในคลังด้วยโดยเก็บเป็นไฟล์ที่มีนามสกุลแบบ DRAWING FILE ปกติคือ .DWG (ตัว DRAWING ปกติก็สามารถที่จะดึงเข้ามาในรูปแบบของ BLOCK ได้เช่นเดียวกัน)

สำหรับการใช้คำสั่ง BLOCK สามารถตั้งชื่อได้ยาวถึง 31 ตัวอักษร (ในคำสั่ง WBLOCK จะตั้งได้ 8 ตัวอักษรเท่ากับความยาวของชื่อไฟล์ปกติ) ในกรณีที่ตั้งชื่อ BLOCK ข้างกับของเดิมที่มีอยู่จะมีข้อความเตือนดังนี้

BLOCK XXXX ALREADY EXISTS.

REDEFINE IT?<N>

หากตอบ N จะออกจากคำสั่ง BLOCK โดยไม่มีการแก้ไขใดๆ ทั้งสิ้นและหากตอบ Y หมายความว่าเราต้องการจะสร้าง BLOCK ขึ้นมาใหม่ในชื่อเดิมเมื่อมีการแก้ไขเปลี่ยนแปลง แล้วจุดใดที่เคยใช้ BLOCK ชื่อดังกล่าวจะถูกแก้ไขใหม่ทั้งหมด และเราจะใช้ประโยชน์ในชื่อนี้ในการเปลี่ยนแปลงแก้ไขแบบ เช่น การปรับขนาด การเปลี่ยนแบบ เป็นต้น ซึ่งจะทำให้การแก้ไขแบบเป็นไปด้วยความสะดวก

คำสั่ง WBLOCK นั้นจะเป็นการเก็บข้อมูลของ BLOCK นั้นๆ ลงไปในดิสก์ด้วยเพื่อประโยชน์ในการเรียกใช้ได้จาก DRAWING FILE อื่นๆ ด้วย ซึ่งในการใช้คำสั่งนี้จะมีคำถามเพิ่มเติมจากการใช้คำสั่ง BLOCK อยู่คำถามหนึ่ง คือ ชื่อไฟล์ที่จะเก็บลงดิสก์หลังจากนั้นจึงจะถามถึงชื่อของ BLOCK ที่ต้องการเก็บ ทั้งนี้เรามีทางเลือกในการตอบเรื่องนี้ดังนี้

- <NAME> ชื่อของ BLOCK ที่จะเก็บลงไปในไฟล์ที่ระบุในคำถามแรก
- = ชื่อ BLOCK ที่เหมือนชื่อไฟล์จะถูกเก็บลงดิสก์
- * เก็บข้อมูลทั้ง DRAWING เหมือนคำสั่ง SAVE เพียงแต่ว่าข้อมูลของ BLOCK ที่ไม่อ้างถึงใน DRAWING นั้นๆ จะไม่ถูกเก็บลงไปด้วย
- <BLANK> หากกด ENTER ใดๆ จะเป็นเหมือนการสร้าง BLOCK ขึ้นมาใหม่ คือจะให้เราเลือกจำนวนวัตถุที่จะมาสร้างเป็น BLOCK ในชื่อเดียวกับชื่อ ไฟล์ที่ตั้งขึ้นมา

วิธีการเรียก BLOCK มาใช้

เมื่อสร้าง BLOCK ขึ้นมาแล้วไม่ว่าจะโดยคำสั่ง BLOCK หรือ WBLOCK ก็ตามการเรียก BLOCK เหล่านี้มาใช้งานจะได้คำสั่งเดียวกันคือคำสั่ง INSERT ในการดึงข้อมูลกลับมาใช้ เราสามารถที่จะกำหนดมาตรฐานในแบบต่างๆ ได้ โดยมีทางเลือกดังนี้

1. การกำหนดค่า X SCALE โดยการป้อนตัวเลขหลังจากนี้โปรแกรมจะถามถึงค่าในแกน Y ต่อไป
2. การกำหนดเป็นกรอบ โดยการกดตัว C การเลื่อนเตอร์เซอร์ไปในทิศทางใดจะทำให้เกิดกรอบสี่เหลี่ยมบนจอภาพ โดยจุดมุมล่างด้านซ้ายของจอภาพคือจุด INSERTION POINT จะอยู่กับที่ ส่วนจุดมุมด้านอื่นๆ จะเคลื่อนที่ไปตามการเลื่อนเตอร์เซอร์ ทั้งนี้ค่า X SCALE หรือ Y SCALE จะเกิดจากขนาดของสี่เหลี่ยมในแต่ละด้านนั่นเอง

3. การกำหนดค่า XYZ โดยการป้อนอักษร YZ ลงไป ซึ่งในทางเลือกนี้จะเป็นทางเลือกที่จะกำหนดขนาดในแกน Z สำหรับ BLOCK ได้ โดยโปรแกรมจะถามถึงระนาบจนครบ

สำหรับทั้ง 3 วิธีที่กล่าวมา เป็นการ INSERT วัตถุเข้ามาในรูปของ BLOCK ซึ่งในวิธีการแรกค่าของ SCALE FACTOR จะปรับตามค่าของ X SCALE FACTOR ดังนั้นในกรณีที่ว่าค่านับของ BLOCK นั้นจริงควบคุมขนาดในแกน Z ไว้มากกว่า 0 แล้ว จะต้องใช้วิธีการ INSERT แบบที่ 3 เท่านั้นจึงจะควบคุมขนาดในแกน Z ได้และเมื่อ INSERT BLOCK เข้ามาแล้ว โปรแกรมจะถือว่าชิ้นส่วนเหล่านั้นเป็นชิ้นเดียวและไม่สามารถแก้ไขรายละเอียดของชิ้นงานเหล่านั้น ซึ่งสำหรับใน VERSION 2.5 ขึ้นไปจะใช้คำสั่ง EXPLODE มาใช้ในการแตกตัวของชิ้นส่วนใน BLOCK ให้เป็นชิ้นส่วนอิสระจากกันจึงจะแก้ไขได้ แต่ใน VERSION ที่ต่ำลงมาหรือทราบอยู่แล้วว่าจะต้องดึง BLOCK นั้นๆ มาเพื่อแก้ไขแล้วการ INSERT โดยใช้เครื่องหมายดอกจัน (*) นำหน้าชื่อ BLOCK ที่ต้องการจะเป็นวิธีการที่สะดวกที่สุด เพราะชิ้นส่วนเหล่านั้นจะถูกดึงมาในรูปของชิ้นส่วนที่อิสระจากกันแต่ในการ INSERT แบบมีเครื่องหมายดอกจันนี้เราไม่สามารถที่จะกำหนดค่าของ SCALE FACTOR ที่แตกต่างกันในแต่ละแกนได้ (คำสั่ง EXPLODE เองนั้น ก็ไม่สามารถใช้กับชิ้นงานที่มีมาตราส่วนในแต่ละแกนต่างกันเช่นกัน)

การให้ค่าของ SCALE FACTOR

เราสามารถที่จะกำหนดค่าให้กับ SCALE FACTOR ของ BLOCK ได้ ซึ่งจะทำให้เกิดการกลับซ้ายเป็นขวา หรือกล่าวได้ว่าเกิดการพลิกวัตถุไป 180 องศา (รอบแกน X หรือ แกน Y โดยที่จุด INSERT POINT จะเป็นเสมือนระนาบ MIRROR LINE ในคำสั่ง MIRROR ขนาดของ BLOCK

ขนาดของ BLOCK จะขึ้นอยู่กับขนาดจริงของวัตถุที่จะสร้างเป็น BLOCK โดยไม่มีข้อจำกัดใดๆ แต่ในชิ้นวัตถุบางประเภทที่มีรูปร่างแบบเดียวกันแต่มีขนาดต่างกัน เช่น รูปสี่เหลี่ยม รูปหัวน็อต หรือเครื่องหมายต่างๆ เป็นต้น เราควรจะสร้างชิ้นวัตถุนั้นๆ ให้อยู่ในขอบเขตขนาด 1x1 unit เพื่อที่ว่าในขณะที่เรา INSERT ชิ้นงานเหล่านี้เข้ามา การกำหนดค่า X SCALE หรือ Y SCALE จะกลายเป็นขนาดของชิ้นส่วนนั้นๆ โดยตรง ทำให้เราไม่ต้องคำนวณหาให้เสียเวลา เช่น ต้องการเขียนหัวน็อตขนาด 145 ไมครอน ก็เพียงแต่กำหนดค่า X และ Y SCALE ให้เท่ากับ 145 เท่านั้น เป็นต้น

ประโยชน์ของ BLOCK ขนาด 1x1 unit อีกประการหนึ่งคือ ในการ INSERT ในระบบของ CONTROL นั้น ขนาดของ BLOCK ดันฉบับมีความสำคัญมาก เพราะในขณะที่เราเลื่อนเคอร์เซอร์ เพื่อกำหนดขนาดของกรอบ ซึ่งก็คือการให้ค่าของ SCALE FACTOR ในแกน X และแกน Y นั้นเอง หากว่า BLOCK ที่จะถูกสร้างขึ้นมาแล้ว แต่ถ้าหากว่า BLOCK ดันฉบับมิได้ถูกสร้างขึ้นมา ซึ่งทำให้การคาดการณ์ถึงขนาดของวัตถุที่จะถูกสร้างขึ้นมาเป็นไปด้วยความลำบาก ออกตัวอย่างเช่น เราสร้าง BLOCK รูปสี่เหลี่ยมจัตุรัสขนาด 1x1 unit เก็บเอาไว้ ใช้สำหรับในการเขียนแผนภูมิ เมื่อเราต้องการขนาดของกรอบที่ดีขึ้นมาจะเป็นขนาดของ BLOCK ที่เรารอออกมาด้วย ทำให้การใช้งานสะดวกกว่าการกำหนดขนาดดันฉบับเป็นขนาดอื่น เพราะจะต้องมีการทอนตัวดูเข้าไปอีก ทำให้ขนาดที่เรารอออกมาจริงไม่เป็นขนาดที่เห็นอยู่ในขณะนั้น

วิธีการนี้จะทำให้โปรแกรมไปค้นข้อมูลในคิสก์ทันที จะเกิดการแก้ไข BLOCK ต่างๆโดยอัตโนมัติ หลังจากนั้นเมื่อโปรแกรมมาถึง INSERTION POINT เราก็กด CTRL C เพื่อหยุดการทำงานไว้เพียงเท่านั้น เพราะ BLOCK ที่เลือกได้ถูกแก้ไขเรียบร้อยแล้ว

การเลือกจุดอ้างอิง BLOCK

การเลือกจุดอ้างอิงหรือ INSERTION POINT ของ BLOCK นั้น ไม่มีข้อจำกัดใด ๆ เพียงแต่ถ้าโดยปกติแล้วการกำหนดจุดอ้างอิงนี้มักจะใช้จุดกึ่งกลางหรือจุดมุมใดๆ ของกลุ่มวัตถุที่จะสร้างเป็น BLOCK กลับมานั้นจุดอ้างอิงนี้จะเป็จุดที่ติดอยู่กับเทอร์เซอร์ นอกจากนี้การหมุนของ BLOCK ก็จะใช้จุดนี้เป็นจุดหมุนด้วย ดังนั้นในการเลือกจุดอ้างอิงนั้นจึงขึ้นอยู่กับความสะดวกในการใช้งานเป็นหลัก ยกตัวอย่างเช่น แปลนของชิ้นส่วนประเภทที่ต้องวางติดกับผนัง (โซฟา อ่างล้างหน้า ชักโครก ฯลฯ) การเลือกจุดอาจจะใช้จุดกึ่งกลางของด้านที่จะต้องติดผนังเป็นจุดอ้างอิงก็ได้ เพราะในการดึง BLOCK ประเภทนี้เข้ามาใช้ ส่วนใหญ่มักจะวางไว้ติดผนัง (ในแปลน) เมื่อจุดอ้างอิงอยู่ด้านที่จะต้องติดผนังอยู่แล้ว การวางตำแหน่งของ BLOCK จะทำได้สะดวกกว่าการใช้จุดอื่น

BLOCK กับ LAYER และ COLOR กับ LINETYPE

BLOCK ที่สร้างขึ้นนั้นสามารถจะประกอบด้วยเส้นสายต่างๆ ที่เขียนขึ้นบนต่างๆ LAYER ต่างสี หรือชนิดของเส้นสายที่แตกต่างกันก็ได้ เช่น คุณสมบัติต่างๆ เหล่านี้จะถูกเก็บเอาไว้กับ BLOCK นั้น ๆ ด้วย เมื่อมีการ INSERT BLOCK เหล่านั้นเข้ามา ก็จะยังคงคุณสมบัติต่างๆ ขึ้นต้นเอาไว้ด้วย ไม่ว่าจะเป็น LAYER สีหรือชนิดของเส้น ในกรณีที่เรา INSERT BLOCK มาจากคิสก์ หาก BLOCK นั้นอ้างอิงถึง LAYER หรือ LINETYPE ที่ไม่เคยมีการใช้ใน DRAWING ปัจจุบันเลย โปรแกรมจะทำการสร้าง LAYER ที่ขาดไปขึ้นมาโดยอัตโนมัติ รวมทั้งการดึงข้อมูลชนิดของเส้นเข้ามาด้วย อย่างไรก็ตามหากภายหลังมีการเปลี่ยนแปลงคุณสมบัติบางประการ เช่น สี หรือชนิดของเส้น ชิ้นส่วนของ BLOCK ที่ใช้ใน LAYER นั้น ๆ อยู่ก็จะเปลี่ยนแปลงไปด้วย หรือแม้แต่การปิดหรือเปิด LAYER ก็จะมีผลทำให้ชิ้นส่วนของ BLOCK ใน LAYER ที่ถูกปิด (off หรือ freeze ก็ตาม) มองไม่เห็นไปด้วย

โดยปกติชิ้นส่วนต่างๆ ของ BLOCK จะวางอยู่บน LAYER ชื่อเดียวกับคอนสร้าง BLOCK ขึ้นมา ยกเว้น LAYER 0 เท่านั้นซึ่งข้อยกเว้นว่าวัตถุใดก็ตามใน BLOCK ที่เขียนบน LAYER 0 เมื่อ INSERT เข้ามาวัตถุชิ้น ๆ จะเปลี่ยนแปลง LAYER ที่จะบรรจุตัวมันไปตาม CURRENT ดังนั้นหากเราต้องการให้ BLOCK ของเราสามารถบรรจุลงไปใน LAYER ใดก็ตาม เราจะต้องสร้าง BLOCK นั้นบน LAYER 0 เท่านั้น และในทางกลับกัน หากเราต้องการสร้างมาตรฐานการเขียนแบบของเราเองว่าชิ้นงานประเภทใดจะต้องวางอยู่บน LAYER ใดแล้ว การสร้าง BLOCK จากชิ้นส่วนที่เขียนบน LAYER ที่ต้องการเลขจะสร้างความสะดวกและป้องกันการผิดพลาดในการวาง BLOCK ลงผิด LAYER ได้เพราะเราไม่ต้องสนใจว่า CURRENT LAYER ขณะนั้นจะเป็นอะไรก็ตาม ชิ้นส่วนของ BLOCK เหล่านั้นก็จะอยู่ตาม LAYER ที่วาง

เช่น ไม้ไผ่โดยอัตโนมัติ เช่น หาก BLOCK ที่เป็น ไม้ไผ่เก้าอี้ต่างๆ เราวางอยู่บน LAYER ชื่อ FURNITURE เสียเลย เมื่อกำหนดให้เป็น BLOCK แล้ว ทุกครั้งที่เราเรียกข้อมูลเข้ามา BLOCK เหล่านี้ก็จะถูกวางลงบน LAYER ชื่อ FURNITURE อัตโนมัติ

การดึง BLOCK เข้ามาเป็นชุด

คำสั่ง INSERT เป็นการดึง BLOCK เข้ามาทีละ 1 ชิ้น แต่เราสามารถที่จะดึงเข้ามาได้ทีละหลายๆ ชิ้นได้ โดยการใช้คำสั่ง MININSERT (MULTIPLE INSERT) คำสั่งนี้จะคล้ายกับคำสั่ง ARRAY แบบ RECTANGULAR คือให้เรากำหนดแนวและคอสมน์ รวมทั้งระยะห่างของ BLOCK แต่ละชิ้น

นอกจากคำสั่ง MININSERT แล้ว ยังมีคำสั่ง ที่สามารถดึง BLOCK เข้ามาได้เป็นชุดเช่นกันคือ คำสั่ง DIVIDE และ MEASURE โดยคำสั่งแรกเป็นการแบ่งชิ้นส่วนที่เลือกเป็นส่วน ๆ ตามจำนวนที่กำหนด ส่วน คำสั่งที่ 2 เป็นการแบ่งชิ้นส่วนออกเป็นช่วงๆ ตามขนาดระยะที่กำหนดทั้ง 2 คำสั่งนี้ มิได้เป็นการคัดลอกชิ้นส่วนที่เลือกออกจากกัน แต่เป็นเพียงการเขียนจุดลงไปใน BLOCK ที่ต้องการลงไปตามจุดที่เกิดขึ้น ซึ่งจำนวนของ BLOCK มากหรือน้อยก็ขึ้นอยู่กับเงื่อนไขการแบ่งที่กำหนด นอกจากนี้ยังสามารถกำหนดให้ BLOCK ที่ดึงเข้ามาปรับตัว (aligned) หรือหมุน (rotate) ไปตามทิศทาง (tangent) ของชิ้นส่วนที่ถูกแบ่งนั้น ซึ่งในการ INSERT ลงไปบนเส้นสายที่มีความโค้ง เช่น ARC หรือ PLINE วิธีการนี้จะสะดวกและง่ายที่สุด

3.3 ATTRIBUTE

การใช้ attribute นับเป็นความสามารถอย่างหนึ่งของการใช้คอมพิวเตอร์ในการเขียนแบบ ซึ่งเป็นลักษณะที่เราจะไม่พบในการเขียนแบบด้วยมือ คือการที่เราสามารถที่จะกรอกข้อมูลใด ๆ ในลักษณะของตัวอักษรตามที่เรต้องการลงไปประกอบกับสิ่งที่เขียนอยู่โดยในการใช้งานจะใช้ร่วมกันกับวัตถุที่เป็น block จะว่าไปแล้วก็มีลักษณะเหมือนการเขียนข้อมูลลงไปในป้าย แล้วไปติดไว้กับวัตถุที่จะสร้างเป็น block เมื่อใดก็ตามที่มีการ insert block เหล่านั้น เข้ามาโปรแกรมจะทำการถามถึงข้อมูลที่จะกรอกลงในป้ายทันทีและในภายหลังสามารถจะนำข้อมูลเหล่านั้นไปใช้ร่วมกับโปรแกรมอื่นๆ ต่อไป เช่น โปรแกรมเกี่ยวกับระบบฐานข้อมูล โปรแกรมทางด้านการพิมพ์ (word processor) ฯลฯ นั่นคือเราสามารถที่จะติดตามความเคลื่อนไหวของวัตถุที่เป็น block ในชิ้นงานของเราได้โดยง่าย ไม่ว่าจะเป็นจำนวนหรือข้อมูลต่างๆ ที่เราต้องการให้ติดอยู่กับ block แต่ละตัวก็ตาม ยกตัวอย่างเช่น เราสร้าง block รูปเก้าอี้โดยต้องการให้มีข้อมูลของรุ่นและราคาติดเข้าไปด้วย ในการนี้จะทำให้เราสามารถตรวจสอบได้ง่ายว่าในแบบทั้งชิ้นมีเก้าอี้รุ่นที่ระบุอยู่ที่ตัว ซึ่งในการเขียนแบบด้วยมือเราคงต้องมานับกันใหม่และคำนวณผลก็อาจเกิดขึ้นได้ แต่สำหรับคอมพิวเตอร์แล้วเราไม่จำเป็นต้องไปนับก็จะให้ตัวเลขกับเราได้ทันที(ส่วนใหญ่เรามักจะใช้ร่วมกับโปรแกรมอื่นเพื่อความสะดวกในการแยกแยะข้อมูลเพราะ โปรแกรม AutoCAD นั้นเก็บข้อมูลให้เราไว้อย่างดี แต่ในเรื่องของการแยกแยะข้อมูลแล้วมันก็จะสะดวกนัก)

นอกจากการติดตามความเคลื่อนไหวของ block แล้ว เราอาจจะใช้ความสามารถของ attribute สำหรับข้อมูลแบบเดียวกับการใช้คำสั่ง TEXT (เมื่อข้อมูลประเภทของ attribute ปรากฏบนจอภาพ ก็จะมีลักษณะเหมือนคำสั่ง TEXT) มีทางเดียวที่จะรู้ได้ก็คือการให้โปรแกรมแสดงรายละเอียดของข้อมูลออกมา โดยใช้คำสั่ง LIST แต่กรณีที่เหมาะสมสำหรับการใช้ attribute เป็นตัวอักษรนั้น มักจะใช้ในกรณีที่มีการใช้ข้อความภายในแบบหลายๆ แห่ง โดยแต่ละแห่งอาจมีความเปลี่ยนแปลงเล็กน้อย ในรายละเอียด เช่น การใส่ spec ของแบบขยายในแต่ละจุด ซึ่งมักจะมีข้อมูลบางส่วนของที่ต้องการแก้ไขบ้างเท่านั้น หรือการเขียนแบบรายละเอียดแสดงชิ้นส่วนอะไหล่ต่าง ๆ เรา มักจะมีการเขียนเบอร์ของอะไหล่ชิ้นนั้นติดไว้ด้วย นอกจากนี้ข้อมูลประเภท attribute ยังสามารถที่จะแก้ไขข้อมูลเหล่านั้นครั้งละมาก ๆ ได้ด้วย (global edit) ซึ่งหากมีการเปลี่ยนแปลงข้อมูลใด ๆ ของชิ้นส่วนที่อยู่ไม่ว่าชิ้นส่วนนั้น ๆ จะวางอยู่ที่ใดในแบบก็จะถูกแก้ไขข้อมูลได้โดยอัตโนมัติ

การกำหนด ATTRIBUTE กับงานขึ้นแบบร่าง

ในบางกรณีการนำเอา attribute ไปติดไว้กับวัตถุที่เป็น block ของชิ้นส่วน รูปร่างที่จะประกอบขึ้นเป็นแบบนั้น (เช่น รูปประตู่ หน้าต่าง ฯลฯ) อาจไม่ได้รับความสะดวกยกตัวอย่างเช่น รูปแปลนของประตู่ ในกรณีที่เป็นการทำแบบร่าง การใส่ประตู่ลงไปในแต่ละจุดอาจจะยังไม่ทราบถึงข้อมูลจำเพาะของประตูแต่ละบานได้ในขณะที่กำลังทำแบบร่าง (ชนิดของวัสดุที่ใช้หรือราคา เป็นต้น) ในกรณีเช่นนี้เรามักจะใช้วิธีนำเอาข้อมูลที่เป็น attribute ไปติดไว้กับ block สัญลักษณ์ที่จะประกอบกับตัวรูปแบบเช่น ประกอบไว้กับสัญลักษณ์แสดงรหัสของประตู เช่น เครื่องหมายแสดง ป1, D1, DR1 เป็นต้น ส่วนที่ตัวรูป block ประตูเป็นเพียงรูปร่างแต่ไม่มีข้อมูล ด้วยวิธีการเช่นนี้เราสามารถที่จะเขียนแบบขึ้นมาได้เรียบร้อยแล้ว และเมื่อตัดสินใจว่าจะกำหนดให้ประตูในแต่ละจุดเป็นแบบใดจึงจะ insert block สัญลักษณ์ที่ต้องการเข้ามาอีกครั้ง นี้ จึงมีข้อมูลและรายละเอียดประกอบลงไปด้วย จะว่าไปแล้ววิธีการนี้จะเหมือนกับวิธีการเขียนแบบด้วยมือ ปกติเรามักจะใส่รายละเอียดของแต่ละชิ้นส่วนประกอบเข้าไปหลังจากที่เขียนแบบเส้นสายต่างๆ เรียบร้อยแล้ว

การสร้าง ATTRIBUTE

การสร้างข้อมูลประเภท attribute จะเป็นเสมือนการเขียนป้ายไปติดไว้กับวัตถุที่เป็น block ทั้งนี้ข้อมูลของ attribute นี้จะถูกสร้างขึ้นก่อนการสร้าง block และในการสร้าง block จะต้องเลือกชิ้นวัตถุโดยรวมไปถึงข้อมูลเหล่านี้ด้วย ซึ่งเมื่อรวมเข้าไปแล้วหากมีการเรียก block ดังกล่าวขึ้นมาเมื่อใดก็ตาม จะมีการถามถึงข้อมูลตามรายละเอียดของ attribute ความที่สร้างขึ้นเสียก่อน ก่อนที่จะนำรูป block นั้น ๆ ลงมาในแบบด้วยคำสั่ง INSERT สำหรับคำสั่งที่ใช้ในการสร้าง attribute คือ ATTDEF (attribute define) หลังจากนั้นโปรแกรมจะถามถึงสถานะของตัวข้อมูลนั้น ๆ (attribute mode) ดังนี้

Attribute modes - Invisible:N Constant:N Verify:N

Enter(ICVP) to change, RETURN when done:

สถานะของ attribute มีอยู่ 3 ลักษณะด้วยกันคือ

1. **invisible** กำหนดให้มองเห็นตัวข้อมูลใน attribute หรือไม่ หากว่าตัวอักษรที่ต่อท้ายเป็น N หมายความว่า จะมองเห็นข้อมูลของ attribute หากเป็น Y จะมองไม่เห็นข้อมูล
2. **Constant** กำหนดให้ข้อมูลใน attribute เป็นค่าคงที่หรือไม่ หากเป็น N จะทำให้การตามถึงตัวข้อมูลในขณะที่ insert หากเป็น Y จะเป็นการนำค่าที่กำหนดในขณะสร้าง attribute ไปใช้ในทันที โดยไม่มีการตาม
3. **Verify** กำหนดให้มีการตามย้าถึงตัวข้อมูลที่ป้อนเข้าไปด้วยหรือไม่ หากเป็น N จะไม่มีการตาม

ในการที่จะแก้ไขสถานะต่างๆ สามารถที่จะทำได้โดยการป้อนตัวอักษรหน้าสถานะต่างๆ ที่ต้องการแก้ไขลงไป ในที่นี้คือ ICV เมื่อป้อนตัวอักษรแต่ละตัวลงไป จะทำให้มีการเปลี่ยน mode จาก Y เป็น N หรือ N เป็น Y สลับกันไป เมื่อได้สถานะต่างๆ ตามที่ต้องการแล้วให้กด Enter ใดๆ ก็จะผ่านขั้นตอนนี้ไปได้ โดยมีคำถามต่อไปนี้คือ

Attribute tag:

เป็นชื่อหัวข้อเรื่องของข้อมูล (หรือเรียกได้ว่าเป็นชื่อของป้าย) ที่จะใช้อ้างอิงในการเรียกข้อมูลมาดูภายหลัง ถ้าจะเทียบกับศัพท์ในระบบข้อมูล tag นี้ก็คือชื่อ field นั่นเอง เช่น ชื่อ อายุ รหัสสินค้า เป็นต้น คำถามต่อไปคือ

Attribute prompt:

เป็นข้อความที่จะขึ้นมาเป็นคำถามบนจอภาพ ในขณะที่มีการ Insert block เข้ามาโปรแกรมจะถามถึงข้อมูลของ tag ด้วยข้อความนี้ ซึ่งหากกด Enter ใดๆ จะเป็นการบอกให้โปรแกรมนำเอาชื่อของ attribute tag มาเป็นคำถามแทน คำถามต่อไปนี้เป็น

Default Attribute Value :

สำหรับคำถามนี้จะเป็นการกำหนดค่าที่จะให้โปรแกรมที่ใช้เป็น default value ของ tag ข้างต้น ในกรณีที่มิระบุ mode แบบ constant แล้ว ในขณะ that insert block เข้ามาจะไม่มีการตามถึงตัวข้อมูลแต่จะนำค่าที่ใหไว้ในคำถามนี้ไปใช้ทันที

หลังจากนี้คำถามที่ตามมาจะเหมือนกับคำถามในคำสั่ง TEXT ทุกประการ โดยเราจะมีหน้าที่กำหนดตำแหน่งที่จะเขียนป้ายลงไป ซึ่งอาจจะวางอยู่บนชิ้นส่วนที่สร้างเป็น block หรืออยู่ข้าง ๆ กันแล้วแต่ความเหมาะสม (เราอาจจะวางทับลงไปบนชิ้นส่วนที่จะสร้างเป็น block ก็ได้เพื่อเป็นการสะดวกในการเลือกชิ้นส่วน เพราะปกติเราไม่กำหนดให้มีการแสดงข้อมูลของ attribute บนจอภาพเพื่อมิให้ปะปนกับเส้นสายอื่นๆ ที่มีอยู่แล้ว) ในขณะที่สร้าง attribute นั้น ตัวอักษรที่ปรากฏบนจอภาพจะเป็นชื่อของ tag แต่เมื่อมีการ insert block นั้นเข้ามาภายหลังตัวอักษรที่ปรากฏจะกลายเป็นค่าที่เราป้อนเข้าไปแทนที่ อย่างไรก็ตามในขณะที่เลือกชิ้นส่วนมาสร้าง block จะต้องรวมไปถึงตัวอักษรของ attribute เหล่านี้ด้วยจึงจะสมบูรณ์

การเข้าคำสั่ง ATTDEF

การใช้คำสั่ง ATTDEF ในแต่ละครั้งจะเป็นการสร้างป้ายชื่อที่ละป้ายและกลับเข้ามาสู่ mode รับคำสั่งต่อไป (command prompt) แต่โดยมากแล้วเรามักจะมีข้อมูลที่ต้องการเก็บหลายข้อมูลด้วยกัน ในกรณีที่เราสามารถที่จะใส่ป้ายถัดไปได้สะดวกขึ้นโดยสามารถจะคงลักษณะของ text ต่าง ๆ เอาไว้เหมือนบรรทัดที่ผ่านมาได้ โดยมีการกด Enter ที่ command line หลังจากการใช้คำสั่ง ATTDEF ไปซึ่งจะเป็นการเข้าคำสั่งเดิม แต่ในกรณีนี้จะตัดคำถามในตอนท้ายเกี่ยวกับเรื่องของ text ออกทั้งหมดโดยจะเขียน tag ใหม่นี้ในบรรทัดถัดมาในรูปแบบเหมือนเดิม

การกำหนดการแสดงผลของ ATTRIBUTE

ในบางครั้งเราต้องการเก็บข้อมูลในรูปแบบของ attribute ไว้หลายค่าด้วยกันบางค่าที่ต้องการให้แสดงอยู่บนแบบแต่บางค่าก็ไม่ต้องการ ซึ่งเราสามารถควบคุมถึงเหล่านี้ จะเป็น Y หรือ N นอกจากนี้เรายังควบคุมการแสดงผลของ attribute ได้จากคำสั่ง `attdisp(attribute display)` โดยมีอยู่ 3 ลักษณะด้วยกันคือ

1. N<NORMAL> แบบปกติตามการตั้งค่าในครั้งแรกที่กำหนด ATTRIBUTE MODE ของ INVISIBLE ให้เป็น Y หรือ N ก็จะเป็นไปตามนั้น
2. ON กำหนดให้มีการแสดงค่าใน ATTRIBUTE ทั้งหมดออกมา
3. OFF กำหนดให้ไม่มีการแสดงค่าใน ATTRIBUTE ทั้งหมด

แก้ไขป้ายชื่อของ ATTRIBUTE

เราสามารถที่จะแก้ไขชื่อของ attribute (attribute tag) ได้โดยการใช้คำสั่ง `change` ซึ่งโปรแกรมจะให้ลักษณะการตอบสนองเหมือนการแก้ไข text เพียงแต่ว่าคำถามในการแก้ไขด้วยกรซึ่งปกติจะใช้คำว่า "NEW TEXT" แต่สำหรับ attribute แล้วจะใช้คำว่า "NEW TAG" แทนซึ่งก็คือการเปลี่ยนแปลงชื่อของ attribute นั้น ๆ นอกจากนี้ยังมีข้อมูลให้แก้ไขเพิ่มเติมอีกบางประการดังนี้

NEW PROMPT<XXXXXX>:

เป็นการแก้ไขคำถามที่จะปรากฏบนจอภาพ (attribute prompt) ในขณะที่โปรแกรมถามถึงค่าของ attribute ตัวนั้น ๆ และข้อมูลอีกตัวหนึ่งคือ

NEW DEFAULT VALUE <XXXX>:

เป็นการถามถึงค่าที่โปรแกรมจะตั้งไว้ให้ของ attribute ในขณะที่โปรแกรมถามหาค่าของ attribute ตัวนั้น ๆ

การแก้ไขค่าใน ATTRIBUTE

การแก้ไขค่าของ attribute ที่ใส่ลงไปแล้ว จะมีคำสั่งโดยเฉพาะที่ใช้ในการแก้ไข คือ `attdit` ทั้งนี้การแก้ไขมีอยู่ 2 วิธีด้วยกัน คือ แก้ไขทีละตัว (individual editing) และแก้ไขแบบเป็นกลุ่ม (global editing) ซึ่งการที่จะเข้าไปสู่วิธีการใดใน 2 วิธีนี้สามารถทำได้โดยการตอบคำถามแรกเมื่อเราใช้คำสั่ง `attdit` ดังนี้

EDIT ATTRIBUTES ONE AT A TIME? <Y>

หากตอบ Y จะเป็นการแก้ไขทีละตัว และหากตอบ N จะเป็นการแก้ไขแบบเป็นกลุ่ม

1. การแก้ไขทีละตัว หลังจากที่ได้เลือกการแก้ไขแบบทีละตัวแล้ว จะมีข้อความที่จะถามเจาะจงถึง attribute ที่จะเลือกให้แสดงตามลำดับดังนี้

BLOCK NAME SPECIFICATION <*>:

เป็นการถามถึง block ที่จะนำมาแก้ไข attribute ในกรณีที่ได้ให้เครื่องหมายดอกจัน (*) จะหมายถึงทุก block ที่มี attribute จะถูกนำมาแก้ไข คำถามต่อไปนี้เป็น

ATTRIBUTE TAG SPECIFICATION <*> :

หมายถึงว่า ชื่อของ attribute (attribute tag) ตัวใดบ้างที่จะนำมาแก้ไขเครื่องหมาย (*) หมายถึงทั้งหมดเช่นเดียวกัน คำถามต่อไปนี้เป็น

ATTRIBUTE VALUE SPECIFICATION <*> :

เป็นการจำกัดวงให้แสดงไปอีก โดยการถามถึงค่าของ attribute ค่าใดที่จะนำมาแก้ไข หากระบุค่าใดลงไปก็ตาม โปรแกรมจะคัดเลือกเฉพาะ attribute ที่มีค่าตามที่ระบุแก้ไขเท่านั้น หลังจาก 3 คำถามนี้แล้วจะเป็นการเลือกชิ้นส่วนที่จะแก้ไขคือ

select attributes:

เป็นการระบุ attribute ที่จะนำมาแก้ไข โดยการชี้ไปที่ attribute หรือ ศีรษะแบบ window หรือการเลือกวัตถุในแบบอื่นๆ ทั้งนี้โปรแกรมจะตรวจสอบวัตถุที่เลือกในครั้งนี่ว่าตรงกับเงื่อนไขใน 3 คำถามที่ผ่านหรือไม่ หากไม่ตรง โปรแกรมก็จะไม่สนใจ attribute ตัวนั้น ๆ ถึงแม้ว่าจะอยู่ในกลุ่มของการเลือกก็ตาม หลังจากที่ได้เลือกกลุ่ม attribute ที่จะนำมาแก้ไขแล้ว โปรแกรมจะให้คำถามต่อไปคือ

Value/Position/Height/Angle/Style/Layer/Color/Next <N> :

และในขณะที่เดียวกันจะมีกากบาทขึ้นอยู่บน attribute ตัวแรก โดยที่การแก้ไขตัวแปรต่าง ๆ ของ attribute จะกระทำกับตัวที่มีกากบาทปรากฏอยู่ โดยมีทางเลือกในการแก้ไขดังนี้

V<Value>	การแก้ไขค่าใน attribute
P<Position>	การแก้ไขตำแหน่งที่เขียน attribute
H<Height>	การแก้ไขความสูงของตัวอักษร
S<Style>	การแก้ไข Style ของตัวอักษร
L<Layer>	การแก้ไข layer ที่บรรจุ attribute
C<Color>	การแก้ไขสีที่ใช้เขียน
N<Next>	เลื่อนกากบาทไปยัง attribute ตัวถัดไป และการแก้ไขก็จะเปลี่ยน การแก้ไขค่าต่าง ๆ ของ attribute ตัวถัดไปด้วย

สำหรับการแก้ไขค่า จะมีคำถามต่อไปอีกว่า จะเป็นการแก้ไขบางส่วนของค่าเดิม หรือนำค่าใหม่ลงไปแทนที่ค่าเดิมเลย ด้วยคำถามดังนี้

Value Change or replace? <R> :

โดยที่ C<Change> : เป็นการเปลี่ยนแปลงเพียงบางส่วนของค่าเดิม โดยมีค่าเดิมดังนี้

String to change : - ค่าเดิมส่วนที่จะเปลี่ยนไป

New String : - ค่าใหม่ที่จะไปแทนที่ค่าที่เปลี่ยนให้ค่าเดิมที่ผ่านมา เช่น ค่าเดิมคือ "two side" ต้องการแก้เป็น "one side" ในกรณีนี้ String to change คือ two และ new string คือ one

2. การแก้ไขแบบเป็นกลุ่ม เราสามารถที่จะแก้ไข attribute ได้คราวละมาก ๆ แต่จะเป็นการแก้ไขเพียงค่าของ attribute อย่างเดียวเท่านั้น โดยหลังจากที่เราเลือกการแก้ไขแบบเป็นกลุ่มแล้ว (ตอบ N ในคำถามแรกของคำสั่ง edit) โปรแกรมจะแสดงข้อความดังนี้

Globe edit of attribute values.

Edit only attributes visible on screen? <Y>

โปรแกรมจะถามว่า การแก้ไขนั้นจะกระทำกับ attribute เฉพาะที่ปรากฏอยู่บนจอภาพเท่านั้นหรือไม่ หากตอบ Y จะเป็นการแก้ไขเฉพาะที่ปรากฏอยู่บนจอภาพหลังจากนั้นก็จะมีคำถามเกี่ยวกับการระบุถึง attribute 3 คำถามคือ block, tag และ value และให้เราเลือก (select object) ที่จะนำมาแก้ไขแบบทีละตัวต่างกันแต่เพียงว่าเมื่อเราเลือกเสร็จสิ้นแล้ว โปรแกรมจะให้เราป้อนค่าของ attribute (ค่าเดิม) ที่จะแก้ไขและค่าใหม่ที่จะถูกแก้ไขลงไป และทำการแก้ไข attribute ทั้งหมดที่เลือกทันที

ด้วยขั้นตอนแบบเดิมหากตอบ N แทน จะหมายความว่าจะมีการแก้ไข attribute ทุกตัวไม่ว่าจะมองเห็นหรือไม่เห็นบนจอภาพก็ตาม ซึ่งในกรณีนี้จอภาพจะเปลี่ยนไปเป็น text mode และมีข้อความว่า

Drawing must be regenerated afterwards.

ซึ่งหมายความว่า จะเกิดการสร้างภาพใหม่ หลังจากที่มีการแก้ไขค่าของ attribute เรียบร้อยแล้ว และเมื่อเราได้เลือกวัตถุที่เป็น block ต่างๆ มาแก้ไข attribute แล้ว โปรแกรมจะแสดงข้อความว่า พบ block ตามเงื่อนไขที่ระบุเป็นจำนวนเท่าใด และถามถึงค่าเดิมที่ต้องการแก้ไขใหม่ที่ต้องการด้วยข้อความดังนี้

num attributes selected

string to change:.....

New string:.....

เมื่อป้อนค่าต่าง ๆ เรียบร้อยแล้วการสร้างภาพใหม่ก็จะเกิดขึ้นและ attribute ที่ตรงตามเงื่อนไขทั้งหมดจะถูกแก้ไขใหม่ โดยการแก้ไขจะเกิดขึ้นกับ attribute ที่มีสถานะเป็น on หรือ visible เท่านั้น อย่างไรก็ตามเราสามารถที่จะควบคุมการแสดงผลของ attribute ได้ตามใจได้ โดยการใส่คำสั่ง attrdisp ซึ่งจะทำให้เราสามารถกำหนดค่าให้ attribute ที่มองไม่เห็น (invisible หรือ off) ให้มองเห็น โดยการกำหนดสถานะ เป็น on เพื่อที่จะนำ attribute ที่ซ่อนมาแก้ไขได้

การสกัดข้อมูลจาก Attribute

เมื่อเราสร้าง attribute ติดไปกับ block แล้วทุกครั้งที่เรา insert block เข้ามาโปรแกรมจะตามหาคำที่จะใส่ลงไปใน attribute ต่างๆ เสมอ หลังจากที่เราป้อนข้อมูลเหล่านี้เอาไว้ใน drawing แล้วเราสามารถที่จะสกัดข้อมูลของ attribute ไปใช้ในโปรแกรมอื่นๆ ได้ เช่น โปรแกรมพวก word processor หรือโปรแกรมพวก database เป็นต้น โดยเฉพาะในกลุ่มหลังเป็นโปรแกรมที่มีประโยชน์มากในการที่เราจะสกัด แยกแยะ และจัดระเบียบข้อมูลของ attribute ได้สะดวกที่สุดสำหรับในกรณีที่มีการ insert ไม่มากนัก วิธีการนี้ คงไม่จำเป็นเท่าใดนัก แต่ในกรณีที่ block เป็นจำนวนมากแล้ววิธีการนี้จะมีประโยชน์มากทีเดียว

การสร้าง Template File

ก่อนที่เราจะสกัดข้อมูลของ attribute ต่างๆ ออกมานั้น เราจำเป็นต้องสร้างต้นแบบหรือ template file สำหรับที่จะให้ตัวอย่างแก่โปรแกรมในการที่จะสกัดข้อมูลออกมาทั้งนี้ตัวแปรต่างๆ ที่ใช้ใน template file ก็จะเป็นชื่อของ attribute (attribute tag) ที่เราตั้งชื่อเอาไว้ตอนสร้าง attribute ขึ้นมาสำหรับ template file ก็คือ ASCII FILE ธรรมดาซึ่งสามารถสร้างขึ้นโดยใช้โปรแกรมพวก word processor (สำหรับการใช้คำสั่ง copy ใน dos นั้นเราอาจจะไม่ต้องออกจากโปรแกรม Autocad ก็ได้ แต่ใช้คำสั่ง SHELL เพื่อออกจากโปรแกรมไปที่ DOS เป็นการชั่วคราวและกลับเข้ามาที่โปรแกรม Autocad ใหม่โดยใช้คำสั่ง EXIT) สำหรับไฟล์ที่จะสร้างขึ้นประกอบไปด้วยข้อมูลต่างๆ ได้แก่ attribute tag ชนิดของข้อมูล (ตัวเลข <N>, ตัวอักษร <C>, ความยาวของข้อมูล และจำนวนหลักหลังจุดทศนิยม) สำหรับความยาวของข้อมูลและจำนวนหลักหลังจุดทศนิยมแต่ละตัวจะประกอบไปด้วยตัวเลขจำนวน 3 หลักเสมอ ตัวอย่างเช่น

- TYPE C001000 - attribute ชื่อ TYPE เป็นตัวอักษรความยาว 1 ตัวอักษร
- MATERIAL C01500 - attribute ชื่อ MATERIAL เป็นตัวอักษรความยาว 15 ตัวอักษร
- PRICE N00502 - attribute ชื่อ PRICE เป็นตัวเลขความยาวทั้งหมด 5 หลัก จำนวนเลขหลังจุดทศนิยม 2 ตำแหน่ง

นอกจากนี้เรายังสามารถสกัดข้อมูลของ block ที่มี attribute ติดอยู่ด้วยได้ (block ใดก็ตามที่ไม่มี attribute ก็จะไม่สามารถสกัดข้อมูลแบบนี้ได้ โดยในการสกัดข้อมูลเหล่านี้ค่อนข้างน้อยจะต้องมี attribute tag ประกอบไว้ใน template file ด้วย 1 ตัว) โดยหากต้องการข้อมูลตัวใด ก็สามารถทำได้โดยการเพิ่มเติมชื่อตัวแปรที่ต้องการเหล่านี้ลงไปคือ

- BL:LEVER N000000 - ระดับการซ้อนกันของ block (nested block)
- BL:NAME C000000 - ชื่อ block
- BL:X N000000 - ตำแหน่งของ block ในแกน X
- BL:Y N000000 - ตำแหน่งของ block ในแกน Y
- BL:LAYER C000000 - ชื่อ layer ที่บรรจุ block
- BL:ORIENT N000000 - มุมหมุนของ block

การสกัดข้อมูลจาก Attribute

เมื่อเราสร้าง attribute ติดไปกับ block แล้วทุกครั้งที่เรา insert block เข้ามาโปรแกรมจะถามหาค่าที่จะใส่ลงไป attribute ต่างๆ เสมอ หลังจากที่เรานำข้อมูลเหล่านี้เอาไว้ใน drawing แล้วเราสามารถที่จะสกัดข้อมูลของ attribute ไปใช้ในโปรแกรมอื่นๆ ได้ เช่น โปรแกรมพวก word processor หรือโปรแกรมพวก database เป็นต้น โดยเฉพาะในกลุ่มหลังเป็นโปรแกรมที่มีประโยชน์มากในการที่เราจะสกัด แยกแยะ และจัดระเบียบข้อมูลของ attribute ได้สะดวกที่สุดสำหรับในกรณีที่มีการ insert ไม่มากนัก วิธีการนี้ คงไม่จำเป็นเท่าใดนัก แต่ในกรณีที่ block เป็นจำนวนมากแล้ววิธีการนี้จะมีประโยชน์มากทีเดียว

การสร้าง Template File

ก่อนที่เราจะสกัดข้อมูลของ attribute ต่างๆ ออกมานั้น เราจำเป็นที่จะต้องสร้างต้นแบบหรือ template file สำหรับที่จะให้ตัวอย่างแก่โปรแกรมในการที่จะสกัดข้อมูลออกมาทั้งนี้ต้นแบบต่างๆ ที่ใช้ใน template file ก็จะเป็นชื่อของ attribute (attribute tag) ที่เราตั้งชื่อเอาไว้ตอนสร้าง attribute ขึ้นมาสำหรับ template file ก็คือ ASCII FILE ธรรมดาซึ่งสามารถสร้างขึ้นโดยใช้โปรแกรมพวก word processor (สำหรับการใช้คำสั่ง copy ใน dos นั้นเราอาจจะไม่ต้องออกจากโปรแกรม AutoCAD ก็ได้ แต่ใช้คำสั่ง SHELL เพื่อออกจากโปรแกรมไปที่ DOS เป็นการชั่วคราวและกลับเข้ามาที่โปรแกรม AutoCAD ใหม่โดยใช้คำสั่ง EXAT) สำหรับไฟล์ที่จะสร้างขึ้นประกอบไปด้วยข้อมูลต่างๆ ได้แก่ attribute tag ชนิดของข้อมูล (ตัวเลข <N>, ตัวอักษร <C>, ความยาวของข้อมูล และจำนวนหลักทศนิยม) สำหรับความยาวของข้อมูลและจำนวนหลักทศนิยมแต่ละตัวจะประกอบไปด้วยตัวเลขจำนวน 3 หลักเสมอ ตัวอย่างเช่น

TYPE C001000 - attribute ชื่อ TYPE เป็นตัวอักษรความยาว 1 ตัวอักษร

MATERIAL C01500 - attribute ชื่อ MATERIAL เป็นตัวอักษรความยาว 15 ตัวอักษร

PRICE N00502 - attribute ชื่อ PRICE เป็นตัวเลขความยาวทั้งหมด 5 หลัก จำนวนเลขทศนิยม 2 ตำแหน่ง

นอกจากนี้เรายังสามารถสกัดข้อมูลของ block ที่มี attribute ติดอยู่ด้วยได้ (block ใดก็ตามที่ไม่มี attribute ก็จะไม่สามารถสกัดข้อมูลแบบนี้ได้ โดยในการสกัดข้อมูลเหล่านี้อย่างน้อยจะต้องมี attribute tag ประกอบไว้ใน template file ด้วย 1 ตัว) โดยหากต้องการข้อมูลตัวใด ก็สามารถทำได้โดยการเพิ่มเติมชื่อต้นแบบที่ต้องการเหล่านั้นลงไปคือ

BL:LEVER N000000 - ระดับการซ้อนกันของ block (nested block)

BL:NAME C000000 - ชื่อ block

BL:X N000000 - ตำแหน่งของ block ในแกน X

BL:Y N000000 - ตำแหน่งของ block ในแกน Y

BL:LAYER C000000 - ชื่อ layer ที่บรรจุ block

BL:ORIENT N000000 - มุมหมุนของ block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BL:XSCALE NOOOOOO - ค่ามาตราส่วนในแกน X

BL:YSCALE NOOOOOO - ค่ามาตราส่วนในแกน Y

การสกัดข้อมูลออกจากรูปร่าง

หลังจากที่เราสร้าง template file ขึ้นมาแล้ว เราก็สามารถที่จะสกัดข้อมูลออกไปได้ตามรูปแบบของ template ที่สร้างขึ้น โดยการใช้คำสั่ง ATTEXT ซึ่งมีรูปแบบของข้อมูล (data format) อยู่ 3 รูปแบบด้วยกัน คือ CDF (comma delimited format) SDF (SYSTEM data format) และ DXF (data exchange format) มีทั้ง 3 รูปแบบ มีขั้นตอนการสกัดข้อมูลออกไปเหมือนกัน แต่แตกต่างกันที่รูปแบบของข้อมูลดังรายละเอียดต่อไปนี้

การใช้ CDF format CDF เป็นรูปแบบของข้อมูลที่แบ่งแยกกันด้วยเครื่องหมาย , โดยที่ attribute ในแต่ละ block จะรวมกันเป็น 1 record แต่ค่าของ attribute จะคั่นด้วยเครื่องหมาย , โดยแต่ละค่าของ attribute นั้นจะเรียกว่า field สำหรับ attribute ที่เป็นอักษรจะอยู่ในเครื่องหมาย ' (single quotes) ส่วนข้อมูลที่เป็นตัวเลขจะไม่มีส่วนมากแล้วโปรแกรมสำเร็จรูปโดยทั่วไป รวมทั้งโปรแกรมภาษา BASIC จะสามารถอ่านไฟล์ในรูปแบบนี้ได้หลังจากที่ใช้คำสั่ง ATTEXT และเลือกรูปแบบ C<CDF> แล้ว โปรแกรมจะถามว่า

Template file :

เป็นการถามถึงไฟล์ ซึ่งก็คือ template file ที่เราสร้างขึ้นนั่นเอง เมื่อป้อนชื่อของ template file แล้ว คำถามต่อไปคือ Extract file name <XXXXXX> :

เป็นการถามถึงชื่อไฟล์ที่จะให้โปรแกรมสร้างขึ้น จากการสกัดข้อมูลตามรูปแบบของ template file ออกไป (default name มักจะเป็นชื่อของ drawing ที่กำลังใช้งานอยู่) โปรแกรมจะทำงานอยู่พักหนึ่ง หลังจากนั้นจะมีข้อความว่า

nan records in extract file.

เป็นการบอกให้ผู้ใช้ทราบว่าข้อมูลถูกสกัดออกไปเป็นจำนวนเท่าใด โดยโปรแกรมได้สร้างไฟล์ขึ้นมา โดยมีนามสกุลเป็น .TXT (โปรแกรมจะตั้งนามสกุลนี้ให้โดยอัตโนมัติ) เท่านั้นที่เป็นอันเสร็จขั้นตอนการสกัดข้อมูล และเราก็สามารถที่จะนำไฟล์ดังกล่าวไปใช้ร่วมกับโปรแกรมอื่นๆ ต่อไป

สำหรับในบางโปรแกรมแล้วอาจต้องการรูปแบบของไฟล์ที่ใช้เครื่องหมายอื่น ๆ คั่นระหว่าง attribute แต่ละตัว แทนเครื่องหมาย , เราก็สามารถกำหนดได้โดยการใส่เครื่องหมายที่ต้องการเพิ่มเติมเอาไว้ที่ template file ในรูปแบบดังนี้เช่น

C:quote " - กำหนดเครื่องหมาย " แทนเครื่องหมาย ,

C:quote / - กำหนดเครื่องหมาย / แทนเครื่องหมาย ,

การใช้ SDF Format SDF เป็นรูปแบบของไฟล์แบบหนึ่ง ซึ่งสามารถอ่านได้จาก หลายๆ โปรแกรมเช่นเดียวกับ CDF เพียงแต่ว่าหากเป็นการนำไปใช้ใน word processor แล้ว format แบบ SDF นี้ อาจจะทำให้ความสะดวกกว่าเพราะแต่ละ attribute จะคั่นด้วย ช่องว่าง (space) เท่านั้น และแต่ละตัวของ

attribute ก็ไม่มีเครื่องหมายใด ๆ ปิดหัวท้าย ถึงแม้ว่าจะเป็นข้อมูลตัวอักษรก็ตาม ส่วนขั้นตอนของการสกัดข้อมูลยังคงเหมือนเดิม

การใช้ DXF Format DXF เป็นรูปแบบมาตรฐานของการแลกเปลี่ยนข้อมูลระหว่าง โปรแกรมทางด้าน CAD ด้วยกัน ทั้งนี้จะมีความแตกต่างกับการใช้คำสั่ง DXFOUT อยู่ในเรื่องของข้อมูลที่จะถูกสกัดไป โดยการเลือก DXF format จาก ATTEXT นั้นจะเป็นการสกัดข้อมูลไปเฉพาะข้อมูลของ block ที่มี attribute อยู่เท่านั้น ส่วนคำสั่ง DXFOUT นั้นจะเป็นการส่งข้อมูลไปที่ drawing แต่คนละรูปแบบของการใช้คำสั่ง END หรือ SAVE

การสร้าง Menu

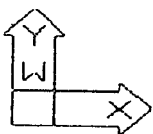
AutoCAD ให้อำนาจความสะดวกเกี่ยวกับการเรียกใช้คำสั่งให้กับผู้ใช้งานด้วยการทำการรายการเมนูคำสั่งแบบต่างๆ ได้แก่ Screen Menu (เมนูคำสั่งที่อยู่ข้างขวามือของจอภาพ) Icon Menu (เมนูรูปภาพ) และ Pop-up Menu (เมนูที่อยู่ส่วนบนของจอภาพ) เพื่อให้ผู้ใช้งานได้เลือกหยิบใช้อย่างสะดวก และนอกจากรายการเมนูมาตรฐานที่มีอยู่ เรายังสามารถสร้างเมนูต่างๆ นี้เอาไว้ใช้งานได้อีกด้วย

การสร้าง Screen Menu

AutoCAD มี Screen Menu มาตรฐานอยู่ในไฟล์ ACAD.MNU หรือ ACAD.MNX ซึ่งไฟล์ ACAD.MNX นั้นได้มาจากการแปล (compiling) ไฟล์ ACAD.MNU (ไฟล์ที่มีนามสกุลเป็น .MNU จะอยู่ในรูปของ Text file ธรรมดา) โดยปกติเมื่อเข้า Drawing Editor AutoCAD จะทำการเรียกไฟล์ Screen Menu ACAD.MNX นี้ไปใช้งานก่อน

■ Layer 0

12.4758,0.5543



Command :
Command :
Command :

AutoCAD
* * * *
ASE
BLOCKS
DISPLAY
DRAW
EDIT
INQUIRY
LAYER...
MODEL
VIEW
PLOT...
RENDER
SETTINGS
SURFACES
UCS:
UTILITY
SAVE:

รูปที่ 3.1 แสดง Screen Menu ของ Auto Cad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มสร้าง Screen menu การสร้าง Screen Menu นั้นจะต้องเก็บข้อมูล Menu File ให้อยู่ในรูปของ Text File โดยใช้ Text Editor ต่างๆ เช่น PCTOOL, NC ,EDIT

ทดลองการสร้าง Menu File ชื่อ SCREENMENU1.MNU โดยมีข้อมูลดังนี้

```
*** SCREEN
[MYMENU-]
[*****]
[REDRAW]^o^oREDRAW
[ZOOM A]^o^oZOOM A
[ZOOM B]^o^oZOOM B
[]
[ERASE L]^o^oERASE L
[ERASE W]^o^oERASE W
[]
[*CANCEL*]^o^oC
```

การเรียกใช้งาน เราสามารถเรียกเมนูมาใช้งานได้ 2 วิธี คือ

1. ใช้คำสั่ง MENU พิมพ์เข้าไปในขณะที่ command:
- 2.เลือกจากเมนูด้านข้างขวาของจอโดยเลือก [UTILITY] และเลือก [MENU] เรียก ใช้ MENU SCREEN1 ที่สร้างไว้ โดยใช้คำสั่ง MENU

Command : MENU

Menu file name or . for non <acad>: SCREEN1

Compiling menu C:\ACADR12\DWG\SCREEN1.MNU...

เมื่อ Auto CAD พบว่า Menu file ที่ต้องการเรียกนั้นยังไม่เคยมีการคอมไพล์ให้เป็น .mnx หรือ menu file .MNU ได้มีการแก้ไขใหม่ AutoCAD จะทำการคอมไพล์ให้โดยอัตโนมัติ หาก Menu file นั้นได้มีการคอมไพล์มาแล้ว ก็จะทำให้การเรียก Menu file .MNX เช่นมาแทนพร้อมกันมีข้อความบอกดังนี้

Loaded menu C:\ACADR12\DWG\SCREEN1.MNX

ข้อควรระวังหาก Menu File ไม่ได้อยู่ในใดเรกทอรี ใช้งาน จะต้องกำหนด path ให้กับ Menu File ด้วยในกรณีที่ autoCAD หา Menu file ไม่พบ อาจจะเนื่องจากการกำหนด path ผิดหรือพิมพ์ชื่อไฟล์ผิด จะมีข้อความบอกดังนี้

"SCRMENU.mnu Can't open file
in C:\ACADR12
or C:\ACADR12ADWG
Enter another menu file name (or return for none) :

โครงสร้างของ Screen Menu หลังจากที่ได้สร้าง Screen Menu ไปแล้ว หมายความว่าโครงสร้างที่จะกำหนด เป็นเมนูนั้นมีอะไรบ้าง

1. Macro Labels (ชื่อกลุ่มคำสั่ง)
2. Macro Commands (คำสั่งปฏิบัติการ)
3. Parameter (ส่วนขยายของคำสั่ง)

ตัวอย่างรายการ zoom A



Macro Labels เป็นชื่อของกลุ่มคำสั่ง (macro) เช่น zoom A, zoom B ฯลฯ ชื่อของกลุ่มคำสั่งนี้จะไปปรากฏบนจอภาพ เพื่อให้ผู้ใช้ได้เลือกใช้งานได้ข้อความที่เป็นแถบ (label) นี้จะอยู่ในเครื่องหมาย [] (square brackets) ความยาวของอักขรที่จะบรรจุลงในเครื่องหมาย [] เพื่อเป็นแถบนี้สูงสุด 8 ตัวอักษรและสามารถเว้นวรรคได้ชื่อของกลุ่มคำสั่งนี้เป็นฉลากข้อความสำหรับแต่ละคำสั่งเท่านั้น อาจจะไม่สามารถมากนัก เพียงแต่ให้สื่อความหมายได้เท่านั้น ส่วนการเลือกกลุ่มคำสั่งจะต้องใช้รูปกราฟหรือใช้คีย์ลูกศรเลื่อนไปเลือกเท่านั้น ไม่สามารถพิมพ์กลุ่มคำสั่งลงไปได้

Macro Commands เป็นคำสั่งหรือกลุ่มคำสั่งเพื่อให้ AutoCAD ไปปฏิบัติ ข้อความที่อยู่หลังเครื่องหมาย] (right square bracket) AutoCAD จะถือว่าเป็นกลุ่มคำสั่งที่จะให้ปฏิบัติจะต้องพิมพ์เครื่องหมาย] โดย ห้ามเว้นวรรค เช่น [Redraw]^C REDRAW จะมีค่าเท่ากับการกดคีย์ดังนี้

กด Ctrl C 2 ครั้ง และ REDRAW enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่เริ่มต้นกลุ่มคำสั่งด้วย ^ C ^ C เพื่อให้ AutoCAD ขกเลิกคำสั่งต่างๆ เพื่อให้แน่ใจว่าได้กลับไปอยู่ที่ Command: ก่อนปฏิบัติคำสั่งที่ได้กำหนด

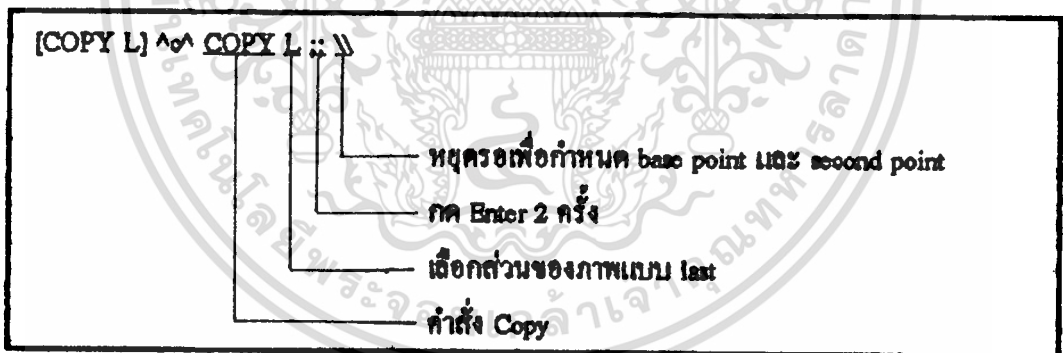
PARAMETERS หรือ พารามิเตอร์เป็นส่วนขยาย ของคำสั่งต่างๆ ถ้ามีเช่นการ ZOOM มีหลายลักษณะ ถ้าต้องการ zoom all ส่วนขยายของคำสั่ง ZOOM คือ A บางคำสั่งอาจมีพารามิเตอร์หรือส่วนขยายนี้หลายตัว บางคำสั่งอาจไม่มีเลขก็ได้ สำหรับส่วนขยายจะต้องอยู่ห่างจากคำสั่งอย่างน้อย 1 ตัวอักษร โดยเว้นช่องว่างไว้ เช่น [ZOOM A] ^ ^ ZOOM A จะมีคำสั่งนี้

กด Ctrl C 2 ครั้ง และ ZOOM enter A enter

เมื่อสิ้นสุดรายการคำสั่งแต่ละบรรทัด AutoCAD จะถือว่าเป็นการกด Enter โดยอัตโนมัติ ขกเว้นคำสั่งที่สิ้นสุดบรรทัดนั้นมีเครื่องหมาย + ปิดท้าย

การควบคุมเมนูหรือรับข้อมูล

จากที่ผ่านมามาดูเห็นว่าใน 1 กลุ่มคำสั่ง AutoCAD จะดำเนินการทำไปเรื่อย ๆ จนกว่าจะหมดคำสั่ง โดยไม่มีการหยุดรอรับข้อมูลใดๆ ทั้งสิ้น ในกรณีที่ต้องการให้มีการหยุดรับข้อมูล ช่วงใด จะต้องใช้เครื่องหมาย \ (back slashes) แทรกไว้ที่ช่วงนั้น เช่น ต้องการสร้างกลุ่มคำสั่งให้ทำการ Copy ภาพโดยเลือกส่วนของภาพล่าสุด (last) นำไปไว้ที่ตำแหน่งต่างๆ ตามต้องการ ซึ่งจากลักษณะนี้จะเห็นว่า จะต้องมีการหยุดรอ 2 ช่วง คือ ช่วงเลือก base point และ second point จะได้คำสั่งดังนี้



นอกจากเครื่องหมาย \ แล้วยังมีเครื่องหมาย ; (semi colon) ซึ่งทั้งสองเป็นอักขระพิเศษ (special character) ที่ควบคุมเมนู โดยที่เครื่องหมาย ; เพราะได้เป็นเคาะเว้นช่องว่างไว้ได้หรือไม่ คำตอบคือได้แต่ไม่เสมอไป การเว้นช่องว่างจะใช้ในการเขียนข้อความใน Text ไม่ได้ ต้องสร้างเมนูต่อไปนี้ แล้วทดลองใช้เปรียบเทียบกัน

[TEXT 1] C CTEXT \ 3 0 THIS IS TEXT

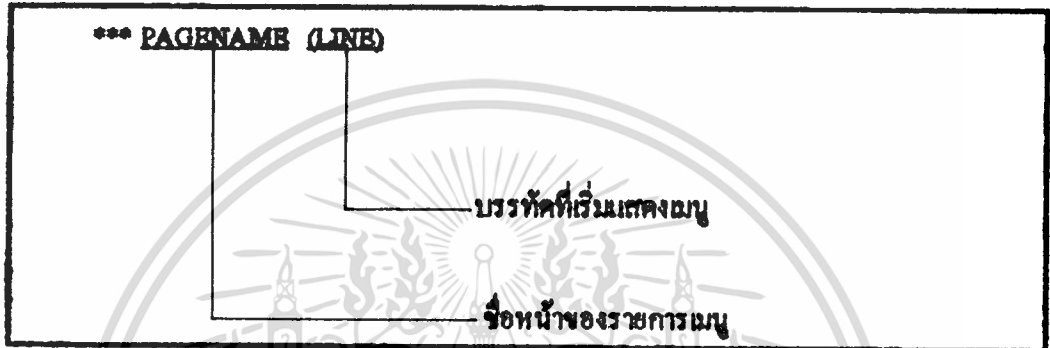
[TEXT 2] C CTEXT \ 3 0 THIS IS TEXT;

สองบรรทัดนี้ต่างกันที่เครื่องหมาย ; ซึ่งในบรรทัดแรกไม่มี เมื่อทดลองใช้ปรากฏว่าในเมนูบรรทัดแรก AutoCAD จะยังไม่เขียนข้อความให้ จะต้องกด enter อีก 1 ครั้ง แต่เมนูในบรรทัดที่ 2 AutoCAD จะเขียนข้อความให้เลยโดยไม่ต้องกด enter อีก

การสร้าง Screen Menu แบบหลายหน้า

ใน 1 หน้าหรือ 1 Screen สามารถมีเมนูคำสั่งได้ 21 รายการ สำหรับจอ EGA และ monochrome ถ้าเป็น VGA จะได้ 26 รายการ แต่ถ้าใช้จอร์นิก ADI (Autodesk Driver Interface) จะได้ถึง 34 บรรทัด หากมีรายการมากกว่านี้จะต้องแบ่งรายการเมนูออกเป็นหน้าโดยที่แต่ละหน้าจะต้องมีรายการเมนูไม่เกินความสามารถของจอ เมื่อแบ่งรายการเมนูออกเป็นหลายๆ หน้าแล้ว จะต้องมีการกำหนดชื่อหน้าแต่ละหน้าไว้ด้วยเพื่อที่จะเรียกใช้ต่อไป

รูปแบบการกำหนดชื่อหน้า มีดังนี้



การเรียกเมนูหน้าอื่นๆ มาใช้งาน

การกำหนดให้ AutoCAD นำเมนูหน้าต่างๆ ขึ้นมาปรากฏบนจอ มีรูปแบบกำหนดดังนี้

\$\$ = Pagename

โดยปกติถ้าไม่มีการกำหนด line (บรรทัดที่เริ่มแสดงเมนู) ที่ชื่อเมนูแต่ละหน้าจะเริ่มแสดงรายการตั้งแต่บรรทัดที่ 1 เสมอ แต่ถ้ามีการกำหนด line เมื่อเมนูหน้านั้นถูกเรียกใช้งาน AutoCAD จะเริ่มแสดงตั้งแต่บรรทัดที่กำหนดไว้ใน line นั้นต่อไป

เริ่มสร้างเมนูแบบหลายหน้าทดลองสร้างเมนูเขียนชื่ออุปกรณ์หรือชิ้นส่วนที่จำเป็นต้องเขียนลงใน drawing สำหรับงานไฟฟ้า ซึ่งเมนูนี้จะช่วยอำนวยความสะดวกเวลาเขียนชิ้นส่วนลงใน drawing

โปรแกรมต่อไปนี้จะแสดงรายละเอียด MENU SYMBOLS.MNU

```
***SCREEN
**symbols
[SYMBOLS]^c$=symbols
[—]
[loadMENU]^c^cmenu acad ;;

[redraw]^credraw
[EraseALL]^cErase A
```

[ATTXT]^cATTXT S D:/FOX25/CONDO2.TXT D:/WRK/CONDO.TXT

[QUICK]^cLOAD "D:/QUICK.LSP"

[HIGH V.]^c\$=x \$=HIGH

[PANEL]^c\$=x \$=PANEL

[LIGHTING]^c\$=x \$=LIGHTING

[PLUG]^c\$=x \$=PLUG

**x3

[LAST_] ^c\$= \$=

**HIGH 3

[TR30]^cINSERT D:/SYMBOLS/TR30.DWG

[TR50]^cINSERT D:/SYMBOLS/TR50.DWG

[TR100]^cINSERT D:/SYMBOLS/TR100.DWG

[TR180]^cINSERT D:/SYMBOLS/TR180.DWG

**PANEL 3

[B2500AF]^cINSERT D:/SYMBOLS/B2500AF.DWG

[B3200AF]^cINSERT D:/SYMBOLS/B3200AF.DWG

[B4000AF]^cINSERT D:/SYMBOLS/B4000AF.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเอกสารใช้งานเพื่อตรวจสอบความถูกต้องเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

****LIGHTING 3**

[L008_40]^cINSERT D:/SYMBOLS/L008_40.DWG

[L006V_40]^cINSERT D:/SYMBOLS/L006V_40.DWG

[L007_40]^cINSERT D:/SYMBOLS/L007_40.DWG

[L008_40]^cINSERT D:/SYMBOLS/L008_40.DWG

****PLUG 3**

[SRWS]^cINSERT D:/SYMBOLS/SRWS.DWG

[SRWD]^cINSERT D:/SYMBOLS/SRWD.DWG

[SRF]^cINSERT D:/SYMBOLS/SRF.DWG

Overlaying Pages Menu

หากเราตั้งเกณฑ์ให้ชื่อเมนูใน AutoCAD จะพบว่าทุกๆ ครั้งที่เลือกการหลักๆ มาใช้ จะปรากฏรายการ [LAST], [DRAW] และ [EDIT] ทุกๆ ครั้งหรือ TEXTMENU ที่เราได้สร้างขึ้นก็เช่นกัน หากมีการเลือก [HIGH V.] [PANEL] หรือ [LIGHTING] ก็จะมีรายการ [LAST] ปรากฏทุกครั้ง ลักษณะนี้เป็นการใช้เทคนิค overlaying โดยมีหลักในการพิจารณารายการ [LAST] เราต้องการให้ไปปรากฏทุกครั้งที่มีการเลือก [HIGH V.], [PANEL] และ [LIGHTING] เป็นเมนู 1 หน้ามี 2 รายการ แต่เว้นช่องว่างไว้หลายบรรทัด ตอนนี้นำชื่อหน้าเมนูเป็น X จากที่ผ่านมามีเราเลือก [HIGH V.] AutoCAD จะนำเมนู หน้า X นี้ไปแสดงก่อน จากนั้นจึงเรียกหน้า HIGH V. ไปกับอีกครั้ง

ข้อสังเกต เมื่อมีการใช้ overlaying pages แล้ว จะทำให้จำนวนรายการเมนูในแต่ละหน้าลดน้อยลงไป เนื่องจากพื้นที่บางส่วนถูกนำไปใช้ในการโอเวอร์เลย์ (overlay) ทั้งนี้ขึ้นอยู่กับความจำเป็นในการใช้พื้นที่ในการโอเวอร์เลย์มีมากน้อยแค่ไหน

การเรียกเมนูหน้าที่เพิ่มผ่านไปยังแถบการใช้งาน

AutoCAD มีความสามารถที่จะเรียกรายการเมนูหน้าที่เพิ่มผ่านไปยังแถบแสดงบนจอได้อีก โดยกำหนดรูปแบบ เป็น \$S= (ไม่ต้องกำหนด pagename) พิจารณารายการจาก TEXTMENU ที่เราได้สร้างขึ้น บางท่านอาจจะสงสัยว่า ทำไมต้องเรียก \$S= ถึง 2 ครั้งด้วยกัน คำตอบ ก่อนหน้าที่มีการเรียกหน้า X และหน้า MELIST ดังนั้นถ้าต้องการกลับไปเดิมจะต้องเรียก \$S= 2 ครั้งเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง POP UP MENU

เราสามารถสร้าง Screen Menu ซึ่งช่วยให้การใช้คำสั่งสะดวกยิ่งขึ้น นอกจากนี้ AutoCad ยังมีเมนูอีกชนิดหนึ่ง คือ POP-UP MENU หรือเรียกว่า Pull-down Menu ซึ่งเป็นเครื่องมืออย่างหนึ่ง ที่ช่วยในการปฏิบัติงาน คำสั่งในการเลือกจากหัวข้อของเมนู POP UP MENU แต่จะต้องมีอุปกรณ์ชี้เป็นตัวเลือกคำสั่งจากเมนู ไม่สามารถใช้นิ้วเป็นเครื่องมือได้ ดังนั้นผู้ที่ต้องการใช้ POP UP MENU จะต้องหาอุปกรณ์ชี้มาใช้ทำงานด้วย

POP-UP MENU ใน AutoCad จะปรากฏที่ต่อเนื่องเราเลื่อนอุปกรณ์ชี้ให้เคอร์เซอร์ขึ้นไปส่วนบนสุดของจอภาพบริเวณ status line ซึ่งเดิมแสดงสถานะ layer และ coordinate ก็จะกลายเป็นเมนูทันที หากต้องการให้ POP-UP Menu หายไปที่เลื่อนอุปกรณ์ชี้ให้เคอร์เซอร์ลงมาจาก status line เท่านั้นเอง สถานะ layer และ coordinate ก็จะปรากฏตามเดิมจะเห็นว่าต่างกับ screen Menu เนื่องจาก POP UP มีการเปิดเมื่อใช้งาน และปิดเมื่อเลิกใช้งานทำให้ประหยัดเนื้อที่บนจอภาพ ส่วน Screen Menu จะค้างอยู่ตลอดเวลา

ข้อดีของ POP UP เหนือกว่า Screen Menu คือ ข้อความแถบหรือชื่อกลุ่มคำสั่งมีความยาวมากกว่า 8 อักขรได้ ทำให้ข้อความหมายได้ชัดเจนยิ่งขึ้น ซึ่งอาจไม่จำเป็นต้องใช้แถบย่อ ดังเช่น Screen Menu และ POP UP ยังเหมาะสมกับกลุ่มคำสั่งที่ไม่ต่อเนื่องคือทำเสร็จแล้วจบแล้ว เช่นการปรับค่านปรในระบบ ฯลฯ

ใน PROGRAM AutoCad จะมี FILE ชื่อ ACAD.MNU ซึ่งอยู่ในราก SUPPORT ACAD.MNU จะมีการเขียน SCREEN MENU, POP UP MENU และการควบคุมอุปกรณ์ชี้ (Mouse) เก็บไว้ใน FILE นี้ ซึ่งโปรแกรม AutoCAD จะ LOAD ACAD.MNU เสมอ เมื่อมีการ RUN PROGRAM

เราจะเขียน POP UP MENU เพิ่มเข้าไปใน ACAD.MNU โดยเพิ่ม POP 10 เข้าไป ซึ่งเดิมมีคำสั่งอยู่ที่ POP 9 จะมีการเขียนคล้ายกับ SCREEN MENU

ตัวอย่าง POP UP ที่เขียนเพิ่มเข้าไปใน ACAD.MNU

```
***pop10
[Electrical]
[ATTEXT]^cATTEXT $ D:/FOX25/CONDO2.TXT D:/WRK/CONDO.TXT
[CHMENU]^c^cmenu D:/ACADR12/SUPPORT/SYMBOLS.MNU
[-]
[->HIGH VOLT]
[->TRANSFORMER]
[TR30]^cINSERT D:/SYMBOLS/TR30.DWG
[TR60]^cINSERT D:/SYMBOLS/TR60.DWG
[TR100]^cINSERT D:/SYMBOLS/TR100.DWG
[<-<-TR1500]^cINSERT D:/SYMBOLS/TR1500.DWG
```

{->LIGHTING}

{->INCAN}

{L033_SM}^cINSERT D:/SYMBOLS/L033_SM.DWG

{<-L038_16I}^cINSERT D:/SYMBOLS/L038_16I.DWG

{->HID}

{->MERCURY}

{L040_80}^cINSERT D:/SYMBOLS/L040_80.DWG

{<-LHDK1000}^cINSERT D:/SYMBOLS/LHDK1000.DWG

{->METAL}

{LHNF250}^cINSERT D:/SYMBOLS/LHNF250.DWG

{LHNF400}^cINSERT D:/SYMBOLS/LHNF400.DWG

{LHNF1000}^cINSERT D:/SYMBOLS/LHNF1000.DWG

{<-LHNF2000}^cINSERT D:/SYMBOLS/LHNF2000.DWG

{->SODIUM}

{LSDK70}^cINSERT D:/SYMBOLS/LSDK70.DWG

{LSDK150}^cINSERT D:/SYMBOLS/LSDK150.DWG

{LSDK250}^cINSERT D:/SYMBOLS/LSDK250.DWG

{<-LSDK400}^cINSERT D:/SYMBOLS/LSDK400.DWG

{->PL}

{LSL09}^cINSERT D:/SYMBOLS/LSL09.DWG

{<-LPLC20}^cINSERT D:/SYMBOLS/LPLC20.DWG

{->FLUORESCENT}

{->FL20}

{L006_20}^cINSERT D:/SYMBOLS/L006_20.DWG

{<-L055_42}^cINSERT D:/SYMBOLS/L055_42.DWG

{->FL32}

{L031_8I}^cINSERT D:/SYMBOLS/L031_8I.DWG

{<-L032_12I}^cINSERT D:/SYMBOLS/L032_12I.DWG

{->FL40}

{L006_40}^cINSERT D:/SYMBOLS/L006_40.DWG

{<-<-L039_CL}^cINSERT D:/SYMBOLS/L039_CL.DWG

รายละเอียด

*** POP 10 เป็นการเขียน POP UP MENU ช่องที่ 10

[-----] ข้อความที่แสดงออกหน้าจอส่วนที่ตามหลังจะเป็นคำสั่งต่างๆของ Pop up

[-->-----] เป็นเทคนิคในการเขียน POP UP หลายๆชิ้น ถ้ามีเครื่องหมายนี้ต้องมีเครื่องหมายออก

[<-----] ในตอนท้ายของชิ้นนั้น

เมื่อคลิกลูกศร mouse ไปที่บนสุดของจอ จะปรากฏคำสั่งให้มาเลือกใช้งาน ELECTRIC เป็น POP10 ที่เขียนขึ้น แสดงได้ดังรูปที่ 3.2 และ รูปที่ 3.3



รูปที่ 3.2 แสดงหน้าจอของ Auto Cad เมื่อเรียก Pop up Electrical

QUICK KEY BOARD

เป็นการเขียนคำสั่งย่อต่างๆ ที่สามารถนำไปใช้งานทางคีย์บอร์ด เพื่อเพิ่มความเร็วในการเขียนแบบ โดยจะเขียนจาก AUTO LISP ซึ่งเป็นภาษาคอมพิวเตอร์ที่ถูกสร้างขึ้นมาจากภาษา LISP (LIST PROCESSING) ที่สามารถเขียนขึ้นในลักษณะที่เป็นโครงสร้าง (Structure)

เขียนแยกส่วนหรือแบ่งกันทำงานได้ แต่ AutoLISP นี้ถือว่าเป็นต้นแบบภาษาที่จะต้องถูกเรียกใช้งานภายใต้ AutoCAD เท่านั้น จะไม่สามารถถูกเรียกใช้งานตามลำพัง โดยจะถูกคอมไพล์ในลักษณะอินเตอร์พรีเตอร์ (interpreter) คือเมื่อป้อนข้อมูลลงไปก็จะให้คำตอบออกมาทีละบรรทัดต่อบรรทัด ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AutoLISP จะใช้งานร่วมกับมาโครของ AutoCAD ในการทำงานและเราจะใช้ AutoLISP นี้เป็นเครื่องมือในการสร้าง Command ขึ้นมาใหม่ด้วยการนำคำสั่งต่างๆ ของ AutoCAD ที่มีอยู่แล้วมาประกอบกัน ใช้ฟังก์ชันพิเศษมาช่วยเสริม และใส่เงื่อนไขต่างๆ เข้าไปเพื่อให้สามารถตัดสินใจเองได้ จากนั้นก็จะนำทั้งหมดที่มีประมวณเข้าด้วยกัน สร้างเป็น Command ขึ้นมาใหม่ ที่จะสามารถเรียกใช้ได้เหมือนกับคำสั่งทั่วไปของ AutoCad



รูปที่ 9.3 แสดงหน้าจอของ Auto Cad เมื่อเรียก Pop up Electrical

โครงสร้างของ AutoLISP

AutoLISP นี้นับได้ว่าเป็นภาษาโครงสร้างภาษาหนึ่งที่มีลักษณะการเขียนโปรแกรมคล้ายกับภาษา C หรือ Pascal แต่จะกระทำได้ง่ายกว่าตรงที่ว่าไม่ต้องการการประกาศ หรือกำหนดชนิดของตัวแปรขณะขึ้นต้นโปรแกรมซึ่งจะทำให้เขียนได้ง่ายเหมือนภาษา BASIC นับได้ว่าเป็นผลคือการเขียนโปรแกรม AutoLISP

เราจะอาศัยหลักการเขียนภาษาโครงสร้างนี้ มาใช้ให้เป็นประโยชน์โดยเราจะทำการแบ่งโปรแกรมออกเป็นส่วนๆ ที่แยกกันทำงานเป็นแบบโมดูล (module) ซึ่งจะง่ายต่อการเขียนโปรแกรมและตรวจหาที่ผิด (Debug) นอกจากนี้ยังสามารถนำโมดูลนี้ไปใช้ได้กับโปรแกรมตัวอื่นๆ ได้ ซึ่งถือได้ว่าเป็นการเขียนโปรแกรมแบบมีประสิทธิภาพที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM TEST.LSP

;Quick! Command for AutoCAD R12

(DEFUN C:A () (COMMAND 'ARRAY))

(DEFUN C:AR () (COMMAND 'AREA))

(DEFUN C:B () (COMMAND 'BLOCK))

(DEFUN C:BK () (COMMAND 'BREAK))

(DEFUN C:BL () (COMMAND 'BLIPMODE' 'OFF'))

(DEFUN C:CF () (COMMAND 'CHAMFER))

(DEFUN C:CG () (COMMAND 'CHANGE'))

(DEFUN C:CR () (COMMAND 'CIRCLE' 'C'))

(DEFUN C:C () (COMMAND 'COPY'))

(DEFUN C:CC () (COMMAND 'COPY' 'C'))

(DEFUN C:CW () (COMMAND 'COPY' 'W'))

(DEFUN C:DD () (COMMAND 'DDEDIT'))

(DEFUN C:DIV () (COMMAND 'DIVIDE'))

(DEFUN C:DI () (COMMAND 'DIM'))

(DEFUN C:DI1 () (COMMAND 'DIM1'))

(DEFUN C:DH () (COMMAND 'DIM' 'HOR'))

(DEFUN C:DV () (COMMAND 'DIM' 'VER'))

(DEFUN C:D () (COMMAND 'DIST'))

(DEFUN C:DL () (COMMAND 'DIM' 'LEADER'))

(DEFUN C:DN () (COMMAND 'DONUT'))

(DEFUN C:DT () (COMMAND 'DTEXT'))

(DEFUN C:EL () (COMMAND 'ELLIPSE'))

(DEFUN C:E () (COMMAND 'ERASE'))

(DEFUN C:EC () (COMMAND 'ERASE' 'C'))

(DEFUN C:EW () (COMMAND 'ERASE' 'W'))

(DEFUN C:EX () (COMMAND 'EXPLODE'))

(DEFUN C:ET () (COMMAND 'EXTEND'))

(DEFUN C:F () (COMMAND 'FILLET'))

(DEFUN C:FS () (COMMAND 'OFFSET'))

```

(DEFUN C:F0 () (COMMAND 'FILLET 'F' '0' 'FILLET'))
(DEFUN C:G () (COMMAND 'GRID'))
(DEFUN C:H () (COMMAND 'BHATCH'))
(DEFUN C:I () (COMMAND 'INSERT'))
(DEFUN C:ISO () (COMMAND 'ISOPLANE'))
(DEFUN C:LA () (COMMAND 'LAYER'))
(DEFUN C:L= () (COMMAND 'LAYER' 'ON'))
(DEFUN C:L- () (COMMAND 'LAYER' 'OFF'))
(DEFUN C:L? () (COMMAND 'LAYER' '?'))
(DEFUN C:LF () (COMMAND 'LAYER' 'F'))
(DEFUN C:LT () (COMMAND 'LAYER' 'T'))
(DEFUN C:LS () (COMMAND 'LAYER' 'S'))
(DEFUN C:L () (COMMAND 'LINE'))
(DEFUN C:LTP () (COMMAND 'LINETYPE'))
(DEFUN C:LI () (COMMAND 'LIST'))
(DEFUN C:LM () (COMMAND 'LIMITS'))
(DEFUN C:LTS () (COMMAND 'LTSCALE'))
(DEFUN C:MS () (COMMAND 'MEASURE'))
(DEFUN C:MN () (COMMAND 'MENU'))
(DEFUN C:MR () (COMMAND 'MIRROR'))
(DEFUN C:M () (COMMAND 'MOVE'))
(DEFUN C:MW () (COMMAND 'MOVE' 'W'))
(DEFUN C:MC () (COMMAND 'MOVE' 'C'))
(DEFUN C:ML () (COMMAND 'DDLMODES'))
(DEFUN C:OFF () (COMMAND 'OFFSET'))
(DEFUN C:PO () (COMMAND 'POINT'))
(DEFUN C:P () (COMMAND 'PAN'))
(DEFUN C:PE () (COMMAND 'PEDIT'))
(DEFUN C:PL () (COMMAND 'PLINE'))
(DEFUN C:PU () (COMMAND 'PURGE' 'A'))
(DEFUN C:PT () (COMMAND 'PLOT'))

```

```

(DEFUN C:PO () (COMMAND 'POLYGON))
(DEFUN C:QY () (COMMAND 'QUIT' "Y"))
(DEFUN C:PP () (COMMAND 'PPLOT'))
(DEFUN C:RD () (COMMAND 'REDO'))
(DEFUN C:RG () (COMMAND 'REGEN'))
(DEFUN C:RW () (COMMAND 'REDRAW'))
(DEFUN C:R () (COMMAND 'ROTATE'))
(DEFUN C:S () (COMMAND 'SAVE' ""))
(DEFUN C:SN () (COMMAND 'SNAP'))
(DEFUN C:ST () (COMMAND 'STRETCH' "C"))
(DEFUN C:SC () (COMMAND 'SCALE'))
(DEFUN C:STY () (COMMAND 'STYLE'))
(DEFUN C:SCR () (COMMAND 'SCRIPT'))
(DEFUN C:TB () (COMMAND 'TABLET'))
(DEFUN C:T () (COMMAND 'TEXT'))
(DEFUN C:TR () (COMMAND 'TRIM'))
(DEFUN C:UC () (COMMAND 'UCSICON' 'off'))
(DEFUN C:V () (COMMAND 'VIEW'))
(DEFUN C:VP () (COMMAND 'VPOINT'))
(DEFUN C:VV () (COMMAND 'VIEW' 'W'))
(DEFUN C:V? () (COMMAND 'VIEW' "?"))
(DEFUN C:W () (COMMAND 'WBLOCK'))
(DEFUN C:Z () (COMMAND 'ZOOM'))
(DEFUN C:ZA () (COMMAND 'ZOOM' 'A'))
(DEFUN C:ZC () (COMMAND 'ZOOM' 'C'))
(DEFUN C:ZD () (COMMAND 'ZOOM' 'D'))
(DEFUN C:ZE () (COMMAND 'ZOOM' 'E'))
(DEFUN C:ZP () (COMMAND 'ZOOM' 'P'))
(DEFUN C:ZS () (COMMAND 'ZOOM' 'S'))
(DEFUN C:ZW () (COMMAND 'ZOOM' 'W'))
(DEFUN C:V () (COMMAND 'VSLIDE'))

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการให้บริการแก่ผู้สนใจเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(setvar "cmdecho" 0)

(prompt "Quick! command load..... loaded.")

(prin1)
```

อธิบายโปรแกรม

การเขียน AutoLISP ขึ้นมาใช้งานบน AutoCAD ถือได้ว่าเป็นการสร้างฟังก์ชันใหม่ขึ้นมาั่นเอง ซึ่งในการเขียน LISP ขึ้นมาใหม่นั้น จะต้องมีการกำหนดค่าให้เป็นฟังก์ชันเสียก่อน ด้วยการ DEFUN ซึ่งจะย่อมาจาก DEFINE FUNCTION ซึ่งมีลักษณะการกำหนดรูปแบบดังนี้

(DEFUN C:NAME (Arguments)

(Expression)

(Expression)

)

ซึ่งจะต้องมีวงเล็บเปิดและวงเล็บปิดที่สมดุลกันเสมอ ดังนั้นในการเขียนทุกครั้งควรสร้างวงเล็บขึ้นมา ก่อนแล้วจึงเขียน Expression แทรกเข้าไปซึ่งจะช่วยลดความผิดพลาดจากการเขียนโปรแกรมลงได้มาก

NAME นี้จะเป็นชื่อของฟังก์ชันที่เราจะสร้างขึ้นมา ซึ่งจะใช้ตัวอักษรตัวเล็กหรือตัวใหญ่ก็ได้ แต่ไม่ควรซ้ำกับชื่อฟังก์ชันที่มีอยู่แล้ว

C: จะมีหรือไม่ก็ได้ แต่ถ้าเราใส่หน้า NAME ก็จะหมายถึงเราจะสามารถเรียกใช้งานทาง COMMAND LINE: ได้เลย แต่ถ้าเราไม่ใส่ C: ก็จะหมายถึงเราไม่สามารถเรียกใช้งานโดยทาง Command line ได้ แต่จะต้องสร้างฟังก์ชันขึ้นมาให้เรียกใช้งานได้อีกทอดหนึ่ง

Argument จะเป็นสะพานเชื่อมไปยังฟังก์ชันตัวอื่น ซึ่งเราจะให้มีหรือไม่ก็ได้ แต่ถ้าเราใส่ Argument ลงไปก็จะมีค่าส่งต่อไปยังฟังก์ชันอื่นได้ เหมาะสำหรับการแยกเขียนทีละหลายๆ โมดูล

Expression หรือนิพจน์จะเป็นรายการคำสั่งที่เราได้เขียนขึ้นมาให้ AutoCAD ได้ทำตามตามขั้นตอนต่างๆ ตามที่เราได้วางไว้

Command (เรียกใช้คำสั่งใน AutoCAD)

รูปแบบ : (command<รายการคำสั่งและค่าใช้จำ>...)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นฟังก์ชันที่จะติดต่อกับ AutoCAD เพื่อเรียกใช้งานคำสั่งใน AutoCAD การส่งชื่อคำสั่งและคำสั่งย่อยจะต้องใช้แบบข้อความ String ที่อยู่ในเครื่องหมาย "..." แทน ส่วนค่าที่ส่งให้คำสั่งนั้น จะเป็นค่าตัวเลขหรือตัวอักษรแล้วแต่ความต้องการของคำสั่ง การส่งค่าทุกในฟังก์ชันจะต้องใช้ค่าจริงภายในฟังก์ชันแทน การส่งค่า Enter ให้คำสั่งจะใช้ " " แทน

SETVAR (ตั้งค่าให้กับตัวแปรในระบบ)

รูปแบบ : (setvar<ชื่อตัวแปรในระบบ>)

เป็นฟังก์ชันที่ใช้ในการตั้งค่าให้กับตัวแปรในระบบของ AutoCAD เช่นเดียวกับคำสั่ง Setvar ใน AutoCAD ถ้าต้องการทราบว่ามีค่าใดบ้างใน System variables ของ AutoCAD ให้เข้าไปใน drawing editor และใช้ว่า คำสั่ง Setvar? <enter> ชื่อตัวแปรที่มีคำว่า (Read only) อยู่ทางด้านขวาหมายความว่าใช้เรียกอ่านค่าได้อย่างเดียว จะตั้งค่าใหม่ไม่ได้

PROMPT (แสดงข้อความตรรกะออกไปที่จอภาพ)

รูปแบบ : (prompt<string>)

เป็นฟังก์ชันที่ใช้ในการแสดงข้อความออกไปที่จอภาพ เมื่อเสร็จแล้วให้ผลเป็น nil มีวิธีการใช้งานเช่นเดียวกับฟังก์ชัน print

PRIN1 (แสดงค่าใดๆ ไปที่จอภาพหรือไฟล์)

รูปแบบ : (print <นิพจน์>[<file-desc>])

เป็นฟังก์ชันที่ใช้ในการแสดงค่าใดๆ ก็ตามในนิพจน์ที่ไปที่จอภาพ ถ้ามีการกำหนดค่าตัวแปรที่เก็บค่า File descriptor จะทำให้แสดงผลของนิพจน์ไปที่ไฟล์นั้น ค่าที่แสดงผลจะไม่ขึ้นบรรทัดใหม่ให้เมื่อจบบรรทัด

การใช้งานบน AUTOCAD

หลังจากที่เราได้สร้างโปรแกรมเก็บไฟล์ไว้ในรูป AutoLISP ที่นามสกุล .LSP เสร็จเรียบร้อยแล้ว จะต้องทำการเรียกโปรแกรมเข้ามาในหน่วยความจำก่อนเรียกการโหลดโปรแกรม ซึ่งมีรูปแบบดังนี้

(LOAD "filename")

ซึ่งเราจะต้องใส่พาธให้ถูกต้องด้วย และ AutoCAD จะใช้เครื่องหมาย \ แทนเครื่องหมาย / จากที่เคยใช้งานปกติ เช่นต้องการโหลดโปรแกรม TEST.LSP จาก B:\DATA\TEST.LSP ก็จะใช้รูปแบบคือ

Command : (load "B:/DATA/TEST")

เมื่อลอง load เรียกมาใช้งานก็จะพบว่าเราสามารถเข้าถึงคำสั่งได้อย่างรวดเร็วมาก ยกตัวอย่างเช่น ถ้าพิมพ์ CT ลงไป ก็จะทำให้การเปลี่ยนข้อความตามที่ต้องการทันทีแทนที่จะต้องผ่านขั้นตอนถึง 8 ขั้นตอน

บทที่ 4

การออกแบบและเขียนโปรแกรม

การออกแบบและเขียนโปรแกรม มีลำดับขั้นตอนดังนี้

1. รวบรวมรายละเอียดของอุปกรณ์
2. จัดทำ Block ของอุปกรณ์ต่างๆ
3. เขียน SCREEN MENU และ POP UP MENU
4. การเขียน TEMPLATE FILE
5. การสร้างไฟล์ DATABASE ของอุปกรณ์
6. การเขียนโปรแกรม ESTIMATION
7. การเขียนแบบลงใน AUTO CAD
8. การส่งข้อมูลจากแบบ AUTO CAD ไปให้โปรแกรม ESTIMATION

1. รวบรวมรายละเอียดของอุปกรณ์

กำหนดสัญลักษณ์ (SYMBOL) ที่ใช้แทนอุปกรณ์ระบบไฟฟ้าต่างๆ อาจกำหนดจากมาตรฐาน หรือผู้จัดทำหนังสือเองหลังจากที่กำหนดสัญลักษณ์ที่ใช้ในโปรแกรมแล้วเราจะหาราคาอุปกรณ์เพื่อนำไปใช้ในการกำหนดราคาอุปกรณ์ นำรายละเอียดที่ได้ไปใส่ไว้ใน DATABASE ซึ่งจะกล่าวในหัวข้อที่ 5

2. จัดทำ Block อุปกรณ์ต่างๆ

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญขั้นตอนหนึ่ง การทำ Block เป็นการรวบรวมเอา Object ที่สร้างขึ้นเป็นรูปร่างของสัญลักษณ์ มาเป็นวัตถุชิ้นเดียวกันและเก็บไว้ใน Disk เพื่อสามารถเรียกกลับมาใช้ได้และเราจะใช้คำสั่ง "ATTDEF" เขียนข้อความกำกับไว้ในสัญลักษณ์ เพื่อให้โปรแกรม PROGRAM AUTOCAD สามารถจะนับจำนวนอุปกรณ์ส่งไปให้ POXPRO ประมาณราคาได้ ซึ่งมีขั้นตอนและวิธีการดำเนินการตามลำดับดังนี้

2.1 สร้างตามสัญลักษณ์ ในการสร้างต้องพิจารณาด้วยว่าสร้างอยู่ใน LAYER ใด หากสร้างใน LAYER 0 เมื่อทำ Block แล้วนำไปเรียกใช้จะถูกเก็บลงใน LAYER ปัจจุบันในขณะที่นั้นแต่ถ้าหากสร้าง Block ในขณะที่เลขอื่น ปัจจุบันเป็นคำอื่น ในการนำไปใช้บล็อกนั้นจะถูกเก็บลงในเลขอื่นชื่อเดียวกับที่สร้างบล็อกนั้นๆ โดยไม่สนใจว่าเลขอื่นปัจจุบันจะเป็นชื่อใดก็ตาม

2.2 การกำหนดรหัสของสัญลักษณ์

จะใช้คำสั่ง "ATTDEF" สร้างรหัสให้กับสัญลักษณ์ เมื่อเราพิมพ์คำสั่ง "ATTDEF" ของ AUTOCAD ลงไปแล้วกด ENTER ต่อจากขั้นตอนนี้ โปรแกรมจะบอกให้ผู้สร้างกำหนด สถานะของรหัส 4 สถานะดังนี้

2.2.1 Invisible เราคลิกเป็นพิมพ์ "T" เพื่อเปลี่ยนสถานะจาก "N" เป็น "Y" ถ้ากด "T" อีกครั้งก็จะเปลี่ยนจาก "Y" เป็น "N"

2.2.2 constant เราก็คเป็นพิมพ์ "C" เพื่อเปลี่ยนสถานะ

ถ้าสถานะเป็น "N" โปรแกรมจะรวมถึงข้อมูลทุกครั้งที่มีการเรียกใช้ Block สัญลักษณ์

ถ้าสถานะเป็น "Y" โปรแกรมจะไม่ตามถึงข้อมูลแต่จะนำข้อมูลที่กำหนดมาให้มาใช้ทันที

2.2.3 Verify เราก็คเป็นพิมพ์ "V" เพื่อเปลี่ยนสถานะ

ถ้าสถานะเป็น "N" โปรแกรมจะไม่มีการเข้าถึงข้อมูลที่ย้อนเข้าไป

ถ้าสถานะเป็น "Y" โปรแกรมจะมีการเข้าถึงข้อมูลที่ย้อนเข้าไปทุกครั้ง

2.2.4 Psect เราก็คเป็นพิมพ์ "P" เพื่อเปลี่ยนสถานะ

มีลักษณะคล้ายแบบ Constant คือในการ INSERT จะใส่ค่าที่ตั้งลงไปเลย แต่สามารถแก้ไขได้ในภายหลัง โดยใช้ ATTEDIT

ในที่นี้เราก็คกำหนดสถานะดังนี้

invisible มีสถานะเป็น Y

Constant มีสถานะเป็น Y

Verify มีสถานะเป็น N

Psect มีสถานะเป็น N

เราก็คกำหนดสถานะเหล่านี้ เพื่อทุกครั้งที่มีการเรียก Block สัญลักษณ์มาใช้แล้ว ไม่ต้องมีการตามถึงข้อมูลที่ย้อนเข้าไปและจะไม่มีข้อมูล ปรากฏบนจอภาพจะมีแค่ภาพสัญลักษณ์เท่านั้น

หลังจากที่กำหนดสถานะของข้อมูลเสร็จสิ้นแล้ว โปรแกรมจะตามถึงป้ายชื่อด้วยคำถาม "Attribute Tag:" เราจะต้องพิมพ์ป้ายชื่อให้ตรงกับชื่อ Field ที่อยู่ใน Template file จะกล่าวภายหลังในโปรแกรมนี้ขอให้พิมพ์คำว่า CODE จากนั้นจะพบคำถามต่อไปว่า "Attribute Value" ให้พิมพ์ชื่อรหัส Block ที่ต้องการลงไป จากนั้นก็จะให้กำหนดตำแหน่งตัวอักษรที่จะให้ปรากฏโดยใช้ Mouse เลื่อนไป ตำแหน่งที่ต้องการแล้วกดปุ่ม PICK ที่ Mouse บอกตำแหน่งจะปรากฏข้อความ "CODE" บนจอภาพ (การกำหนดตำแหน่งตัวอักษรควรกำหนดให้อยู่ใกล้กับภาพสัญลักษณ์)

ขั้นตอนต่อไปคือการทำ Block รวบรวมภาพสัญลักษณ์กับข้อมูล Attribute ที่สร้างขึ้นโดยใช้คำสั่ง "Block" เมื่อใช้คำสั่ง Block โปรแกรมจะให้ผู้สร้างตั้งชื่อ BLOCK จากนั้นโปรแกรมจะให้เลือกรุ่นของจอภาพจะให้จัดเก็บไป Block

ขั้นตอนสุดท้ายเป็นการสร้าง Block ที่ทำไว้เป็น File นามสกุล *.DWG เพื่อจัดเก็บเอาไว้ใช้งานอย่างถาวรเพราะการทำ Block เป็นการนำภาพที่แสดงไว้มาใช้ได้ชั่วคราวเท่านั้น เมื่อปิดเครื่อง computer แล้วภาพที่สร้างด้วยคำสั่ง "WBlock" จัดทำ Block ให้เป็น file พิมพ์คำสั่ง "WBlock" แล้วกด Enter โปรแกรมจะตามถึงชื่อ FILE เมื่อพิมพ์ชื่อ FILE แล้วกด ENTER ก็ให้พิมพ์ชื่อ BLOCK ที่ต้องการลงไปและกด ENTER ก็จะเสร็จวิธีการจัดทำ BLOCK สัญลักษณ์

8. การเขียน SCREEN MENU และ POPUP MENU

จากการที่เราจัดทำสัญลักษณ์เป็น Block แล้ว (เป็น File นามสกุล .DWG) ผู้สร้างสามารถเรียกออกมาใช้งานได้โดยใช้คำสั่ง "INSERT" พิมพ์ที่ Command : แต่เพื่อการสะดวกใช้งานผู้สร้างสามารถดัดแปลงโปรแกรมเพื่อให้เรียก ทาง SCREEN MENU และ POP UP MENU

อธิบายโดยยกตัวอย่างประกอบ

สัญลักษณ์อุปกรณ์ไฟฟ้าที่สร้างจึ้นมีดังนี้

1 HIGH VOLT

1.1 TR

TRANSFORMER 300KVA ชื่อ FILE TR 30.DWG

TRANSFORMER 500KVA ชื่อ FILE TR 50.DWG

1.2 HBREAKER

1.2.1 MCCB (MOLD CASE CIRCUIT BREAKER)

BRAKER 630 AF ชื่อ FILE B 30 AF.DWG

BRAKER 1000 AF ชื่อ FILE B 50 AF.DWG

1.2.2 ACB (AIR CIRCUIT BREAKER)

AIR CIRCUIT BREAKER 1000 A ชื่อ FILE BA 1000.DWG

AIR CIRCUIT BREAKER 2000 A ชื่อ FILE BA 2000.DWG

2. PANEL

3. LIGHTING

4. PLUG

5. WIRE

6. CONDUIT

POP UP จะเขียนเพิ่มลงใน FILE ACAD.MNU โดยเพิ่มเป็น POP 10 เข้าไปใน FILE ACAD.MNU SCREEN MENU SAVE เป็น FILE SYMBOLS.MNU COPY ไว้ในรอก Support ของ AUTOCAD จากตัวอย่างที่ให้มาเราสามารถเรียก Screen ออกมาใช้ โดยพิมพ์ "MENU" ที่ Command หรือเลื่อน mouse ไปที่ บนสุดของจอภาพ จะปรากฏ POP UP ขึ้นบนจอภาพ เลือก POP 10 (Electrical) จะปรากฏข้อความดังรูป เลือก CHMENU จะเปลี่ยนการใช้ MENU ซึ่งเดิมใช้ ACAD.MNU ก็จะเปลี่ยนไปใช้ SYMBOLS.MNU แทน ถ้าต้องการกลับไปใช้ ACAD.MNU ก็เลือก LOAD MENU ที่ SCREEN MENU

4. การเขียน TEMPLATE FILE

ในการเขียน Template file สำหรับใช้ในการเชื่อมโยงข้อมูลระหว่าง FoxPro กับ Auto Cad นั้นมีลำดับการเขียน ดังนี้

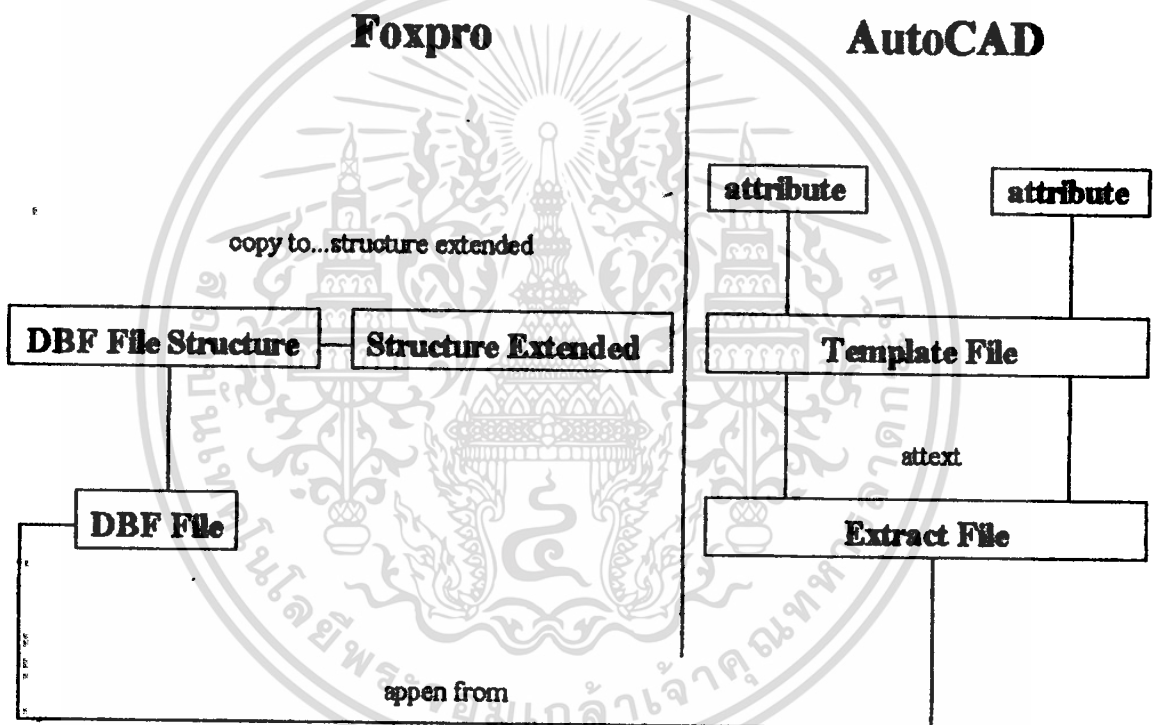
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Template File คือ textfile ที่จะบอก AutoCAD ว่าจะสร้าง ExtractFile ที่มีโครงสร้างอย่างไรเพื่อให้เข้ากับ Database File (DBF) ใน Foxpro ของเราได้ Template File จะบอกว่าจะให้เอา attribute ใดมาใช้อะไร ข้อมูลไหนมาและจะเอาข้อมูลมาในลักษณะใด Template File จะมีลักษณะคล้าย structure ของ Database File

การสร้าง Template File สามารถทำได้ 2 วิธีคือ

1. พิมพ์ลงไปใน Text Editor เช่น ใน WordStar หรือใช้วิธี Copy con ใน Dos
2. สร้างโดย Foxpro เองใช้วิธีเหมือนกับการสร้าง structure

ส่วนกรรมวิธีในการเอาข้อมูลจาก AutoCAD มายัง Foxpro สามารถเขียนเป็นแผนภูมิได้ดังนี้



5. การสร้างไฟล์ DATABASE ของอุปกรณ์

ในการสร้างไฟล์สำหรับเก็บรายละเอียดของอุปกรณ์ไฟฟ้า นั้น เราใช้ FoxPro เป็นตัวสร้างไฟล์ข้อมูล โดยให้ชื่อว่า filedata.dbf มีลักษณะโครงสร้างของไฟล์ ดังนี้

field name	type	width	dec
code	character	8	0
descrip	character	35	0
qty	numeric	4	0
unit	character	4	0

price	numeric	10	2
date	date	8	0
contact	memo	10	0

การป้อนรายละเอียดของอุปกรณ์ มีดังนี้

เขตข้อมูล code - ใช้ภาษาอังกฤษพิมพ์ใหญ่และขึ้นต้นด้วยตัวอักษร T, B, L, S, W, หรือ C เท่านั้น ใช้ความยาวไม่เกิน 8 ตัวอักษร โดยตัวอักษรแต่ละตัวที่ใช้ขึ้นต้นนั้นมีความหมาย ดังนี้

- T หมายถึง อุปกรณ์เกี่ยวกับระบบไฟฟ้าแรงสูง
- B หมายถึง อุปกรณ์เกี่ยวกับ Panel Board
- L หมายถึง อุปกรณ์จำพวกหลอดไฟ
- S หมายถึง อุปกรณ์สวิตช์และปลั๊ก
- W หมายถึง อุปกรณ์จำพวกสายไฟ
- C หมายถึง อุปกรณ์จำพวก Race Way

เขตข้อมูล descrip - เป็นชื่อของอุปกรณ์ที่ทำการป้อน จะใช้ภาษาอังกฤษหรือภาษาไทย ก็ได้ ป้อนได้ไม่เกิน 35 ตัวอักษร

เขตข้อมูล qty - เป็นจำนวนของอุปกรณ์ที่ทำการป้อนใช้ 1 เสมอ

เขตข้อมูล unit - เป็นหน่วยของอุปกรณ์ที่ทำการป้อน ที่ให้มี SET, LOT, EA, M

เขตข้อมูล price - เป็นราคาของอุปกรณ์ที่ทำการป้อน

เขตข้อมูล date - เป็นวันที่ ที่ทำการป้อนข้อมูล

เขตข้อมูล contact - ใช้สำหรับเก็บชื่อ ที่อยู่ ของร้านหรือบริษัท ที่สามารถทำการติดต่อหาซื้อได้

6. การเขียนโปรแกรม ESTIMATION

โปรแกรม ESTIMATION ที่เราเขียนขึ้นนี้ ใช้ภาษา FoxPro เป็นภาษาที่เขียนขึ้น โดยได้เขียนแยกออกเป็น 6 โปรแกรม ดังนี้

1. caloulat.prg
2. link.prg
3. screen.prg
4. printer.prg
5. editdata.prg
6. outtext.prg

ในส่วนของเมนูนั้นเราได้ใช้ภาษา C เขียนขึ้น ซึ่งตัวโปรแกรมที่เขียนขึ้นทั้งหมด (Source file) ได้แสดงไว้ในส่วนของภาคผนวก ก. แล้ว ในที่นี้เราจะอธิบายการทำงานของแต่ละโปรแกรม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรม calculat.png

เป็นโปรแกรมที่ใช้หาจำนวนดวงโคม สำหรับพื้นที่ใช้สอยต่างๆ โดยข้อมูลที่ต้องการคือ ชนิดของพื้นที่ใช้สอย, ขนาดของพื้นที่, ชนิดและขนาดของดวงโคมที่ใช้ โดยข้อมูลเหล่านี้สามารถเลือกได้จากเมนู จากนั้นก็จะได้จำนวนดวงโคม และระยะห่างระหว่างดวงโคมที่ใช้ในพื้นที่นั้น และจะมีข้อความถามว่า " ต้องการทำงานต่อหรือไม่ (Y/N) : _ " ถ้าต้องการทำงานต่อให้ตอบ Y และกด Enter แต่ถ้าไม่ต้องการทำงานต่อให้ตอบ N

2. โปรแกรม link.png

เป็นโปรแกรมที่ทำการเชื่อมโยงข้อมูลระหว่าง FoxPro กับ Auto Cad โดยการนำเอาไฟล์ condo.txt ที่ได้จากการ Attext จาก Auto Cad มาแปลงเป็นไฟล์ .dbf โดยให้ชื่อว่า con1.dbf จากนั้นก็จะนำเอาข้อมูลในไฟล์ con1.dbf มาแยกออกเป็นประเภทต่างๆ มีทั้งหมด 6 ประเภท ดังนี้คือ

ข้อมูลเกี่ยวกับ high volt	จะแยกให้อยู่ในไฟล์	hv.dbf
" panel board	"	pb.dbf
" luminair	"	lm.dbf
" switch & receptacle	"	s_r.dbf
" wire	"	wire.dbf
" race way	"	conduit.dbf

จากนั้นก็ให้นำเอาข้อมูลในแต่ละไฟล์ที่แยกประเภทแล้ว (เฉพาะไฟล์ hv.dbf, pb.dbf, lm.dbf, s_r.dbf) มานับจำนวน และเก็บไว้ในไฟล์ highvolt.dbf, panel_b.dbf, luminair.dbf, sw_recep.dbf ตามลำดับ ต่อจากนั้นก็ทำการเชื่อมโยงข้อมูลระหว่าง FoxPro กับ FoxPro โดยข้อมูลที่ได้จากการเชื่อมโยง จะถูกเก็บอยู่ในไฟล์ highvolt.dbf, panel_b.dbf, luminair.dbf, sw_recep.dbf, wire.dbf, conduit.dbf ตามลำดับ จากนั้นก็จะนำเอาข้อมูลในแต่ละไฟล์มาแสดงผลทางจอภาพต่อไป

3. โปรแกรม screen.png

เป็นการนำเอาข้อมูลในไฟล์ highvolt.dbf, panel_b.dbf, luminair.dbf, sw_recep.dbf, wire.dbf, conduit.dbf และ file.dbf มาแสดงผลออกทางจอภาพ โดยมีรูปแบบให้เหมาะสมกับหน้าจอภาพ โดยแต่ละไฟล์ได้แยกประเภทของข้อมูลไว้แล้ว เช่น ไฟล์ highvolt.dbf เป็นไฟล์ข้อมูลประเภท high volt (เช่น หม้อแปลง) และไฟล์อื่นๆ ก็มีลักษณะเดียวกัน

4. โปรแกรม printer.png

เป็นการนำเอาข้อมูลในไฟล์ file.dbf, highvolt.dbf, panel_b.dbf, luminair.dbf, sw_recep.dbf, wire.dbf, conduit.dbf มาแสดงผลทางเครื่องพิมพ์ หรือพิมพ์ออกมาเป็นรายงาน

5. โปรแกรม editdata.prg

เป็นโปรแกรมจัดการเกี่ยวกับระบบฐานข้อมูล คือ ข้อมูลเกี่ยวกับอุปกรณ์จะเก็บอยู่ในไฟล์ filedata.dbf โปรแกรมนี้จะเป็นโปรแกรมจัดการเกี่ยวกับ การเพิ่มอุปกรณ์ใหม่เข้าไป, การเปลี่ยนราคาอุปกรณ์, และการลบรายการอุปกรณ์ออก โดยทั้งหมดจะทำงานเชื่อมโยงกับไฟล์ filedata.dbf ทั้งสิ้น

6. โปรแกรม outtext.prg

เป็นโปรแกรมที่แสดงข้อความ เป็นข้อความเกี่ยวกับโครงการนี้, ส่วนประกอบของเครื่องคอมพิวเตอร์ที่ต้องการ, และการใช้โปรแกรมอย่างคร่าวๆ

ส่วนของโปรแกรมเมนูที่ใช้ภาษา C เขียนนั้น ได้แสดงไว้ในภาคผนวก ก. แล้วเช่นกันเป็นโปรแกรมเริ่มแรก สำหรับเข้าสู่การใช้งาน โปรแกรมการเขียนแบบระบบไฟฟ้าและประมาณราคาด้วยคอมพิวเตอร์ โดยเมื่อเริ่มใช้โปรแกรมนี้ครั้งแรกจะมีข้อความเกี่ยวกับผู้จัดทำโปรแกรม และอาจารย์ที่ปรึกษา จากนั้นจะรอให้กดคีย์ใดๆ เพื่อเข้าเมนูเพื่อใช้งานโปรแกรมในลำดับต่อไป

7. การเขียนแบบลงใน AUTOCAD

เริ่มเขียนแบบลงใน AUTOCAD เหมือนกับการเขียนแบบทั่วไปที่ใช้ทั่วไปแต่จะสัญลักษณ์ที่ใช้ในแบบต้องเป็นสัญลักษณ์ที่สร้างขึ้นในการทำ Block ในขั้นตอนที่ 2

ข้อเสนอแนะในการใช้ program

7.1 ในการอธิบายสัญลักษณ์ไม่ควรนำรูปสัญลักษณ์ที่สร้างขึ้นในการทำ Block ที่แสดงควรจะสร้างขึ้นใหม่ มิฉะนั้นการนำจำนวนอาจจะเกิดการนำผิดพลาดได้

7.2 สายไฟฟ้ากับอุปกรณ์ร้อยสายเราสามารถนำความยาวได้ ในโครงการนี้ ผู้สร้างควรหาวิธีในการที่จะให้คอมพิวเตอร์รู้ว่าใช้สายชนิดไหนบ้าง ใช้อุปกรณ์ร้อยสายชนิดไหนบ้าง ขอแนะนำให้ทำ Block เหมือนในขั้นตอนที่ 2 ให้นำอุปกรณ์สายไฟฟ้ากับอุปกรณ์ร้อยสายไฟฟ้า นับแต่ละชนิดเป็น 1 ชิ้น computer จะรับรู้ว่าใช้อุปกรณ์อะไรบ้าง ใน PROGRAM ESTIMATION เพื่อตรวจสอบว่าอุปกรณ์พวกนี้ จะมีของว่างในการกรอกความยาวของอุปกรณ์

8. การส่งข้อมูลจากแบบ AUTOCAD ไปให้โปรแกรม ESTIMATION มีอยู่ 3 วิธี

1. พิมพ์ "ATTEXT" ที่ COMMAND LINE

COMMAND : ATTEXT

CDF SDF OR DXF ATTRIBUTE EXTRACT (OR ENTITIES)? <>:

TEMPLATE FILE :

EXTRACT FILE NAME :

COMPUTER จะถาม attribute extract แบบไหนให้เลือก SDF โดยพิมพ์ S แล้ว ENTER ต่อไปจะถาม

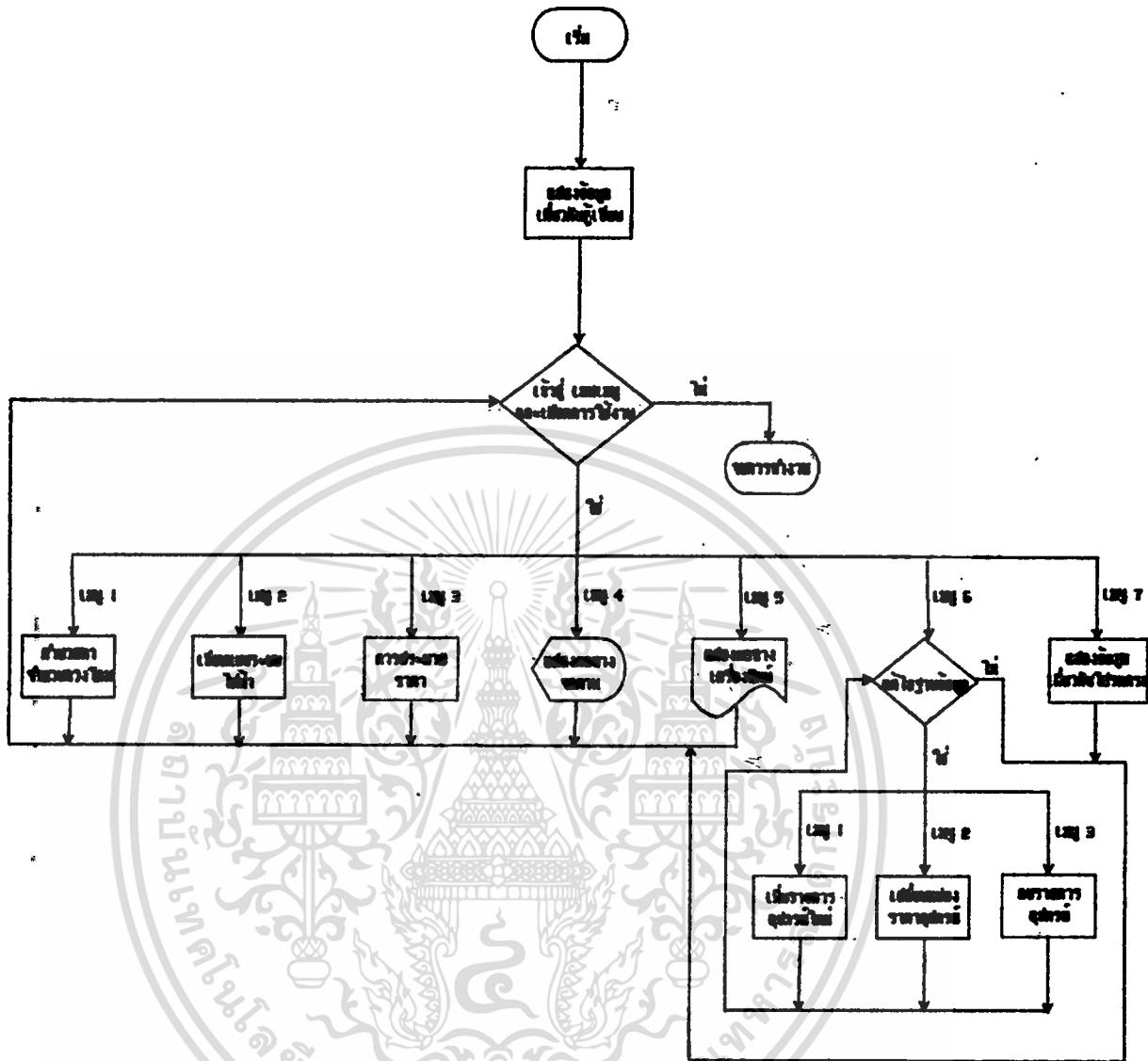
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPLATE FILE : ใส่ชื่อ File ที่จะติดต่อด้วย ผู้สร้างใช้ CONDO2 ต่อไปตาม EXTRACT FILE NAME
ใส่ชื่อ File ที่ attext ออกไปเป็น *.TXT ผู้สร้างใช้ CONDO

2. ใช้ ACAD.MNU เลือก POP UP ที่ Electrical เลื่อนแถบแสงไปที่ ATTEXT แล้วกด PICK ที่ปุ่ม
Mouse (อุปกรณ์ชี้) computer จะ Attext ให้โดยอัตโนมัติ เหมือนกรณีที่ 1

3. ใช้ SYMBOLS.MNU ไปที่ Screen Menu เลื่อนแถบแสงไปที่ ATTEXT แล้วกด Enter computer จะ
Attext ให้อัตโนมัติเหมือนกรณีที่ 1





รูปที่ 4.1 แสดง Flowchart การทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

การทดลองโปรแกรมการออกแบบระบบไฟฟ้าและประมาณราคาด้วยคอมพิวเตอร์ มีขั้นตอนดังนี้

1. ค้นหาจำนวนดวงโคม

2. เขียนแบบแปลนระบบไฟฟ้า ประกอบด้วย

- LIST OF DRAWING AND SYMBOLS
- SINGLE LINE M.D.B. AND RISER DIAGRAM
- SINGLE LC - 1
- SINGLE LC - 2
- SINGLE LC - 3
- SINGLE LCC
- LIGHTING AND RECEPTACLE FIRST FLOOR PLAN
- LIGHTING AND RECEPTACLE SECOND FLOOR PLAN
- LIGHTING AND RECEPTACLE THIRD FLOOR PLAN

3. SAVE ภาพที่เขียน

4. ทำการ PLOT ภาพที่เขียนจะได้ดังรูป

5. ทำการประมาณราคาจากแบบแปลนไฟฟ้าที่เขียนขึ้นจากหัวข้อที่ 2

- เลือกเมนูการประมาณราคา
- เติมค่าความยาวของสายไฟที่ใช้
 - สาย NYY 1 core ขนาด 70 mm² 160 M
 - สาย NYY 1 core ขนาด 16 mm² 40 M
 - สาย THW ขนาด 25 mm² 280 M
 - สาย THW ขนาด 6 mm² 70 M
 - สาย THW ขนาด 16 mm² 1800 M
 - สาย VAF ขนาด 2.5 mm² 540 M
 - สาย VAF ขนาด 4 mm² 810 M
 - สาย THW ขนาด 2.5 mm² 420 M
- เติมค่าความยาวของท่อที่ใช้
 - ท่อหนา IMC ขนาด 4" 40 M
 - ท่อบาง EMT ขนาด 2" 70 M

เอกสารนี้เป็นเอกสารทบทวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ท่อบาง EMT ขนาด 1" 880 M

- ท่อบาง EMT ขนาด 1/2" 210 M

6. ทำการพิมพ์ผลการประมาณราคา

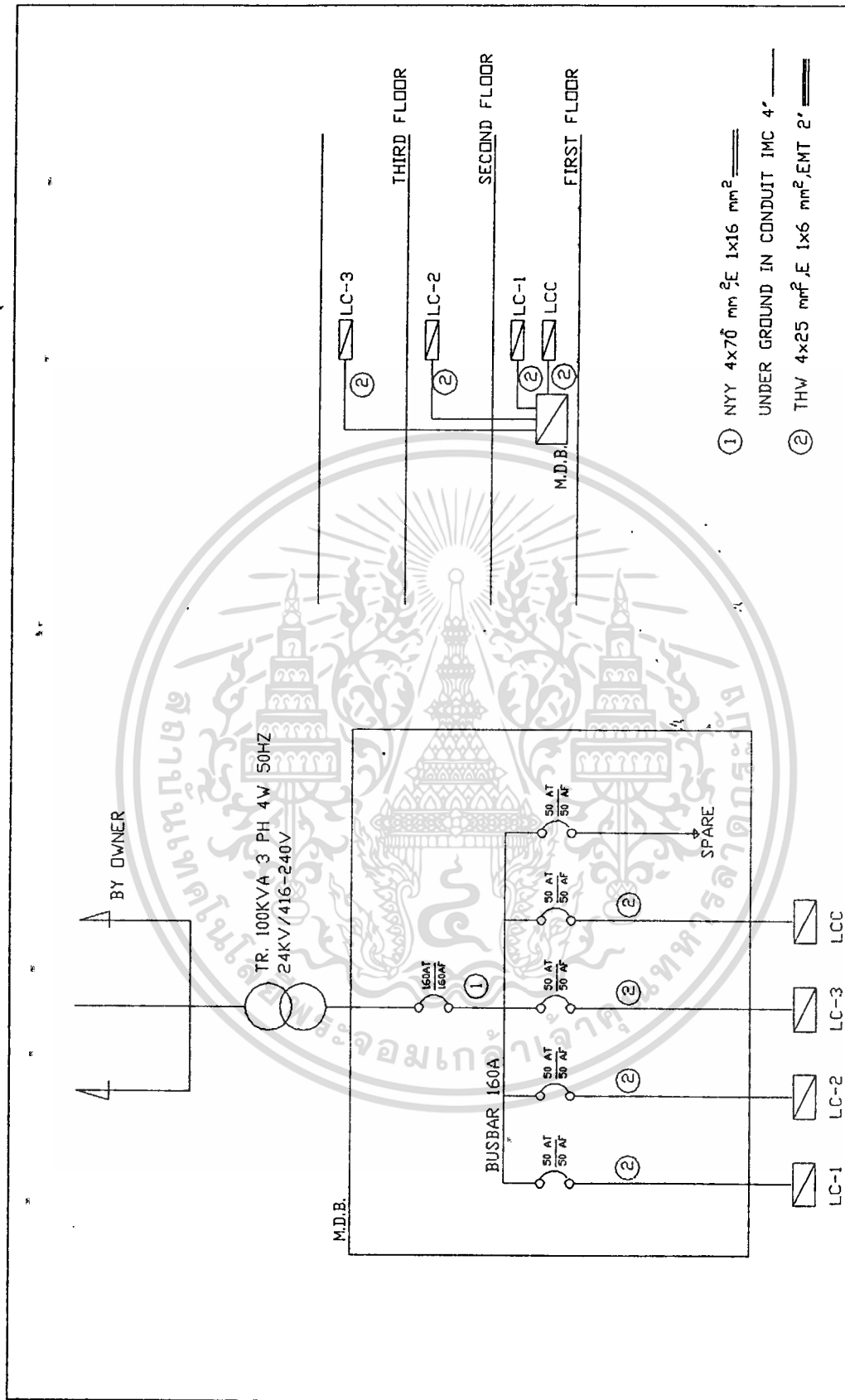


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

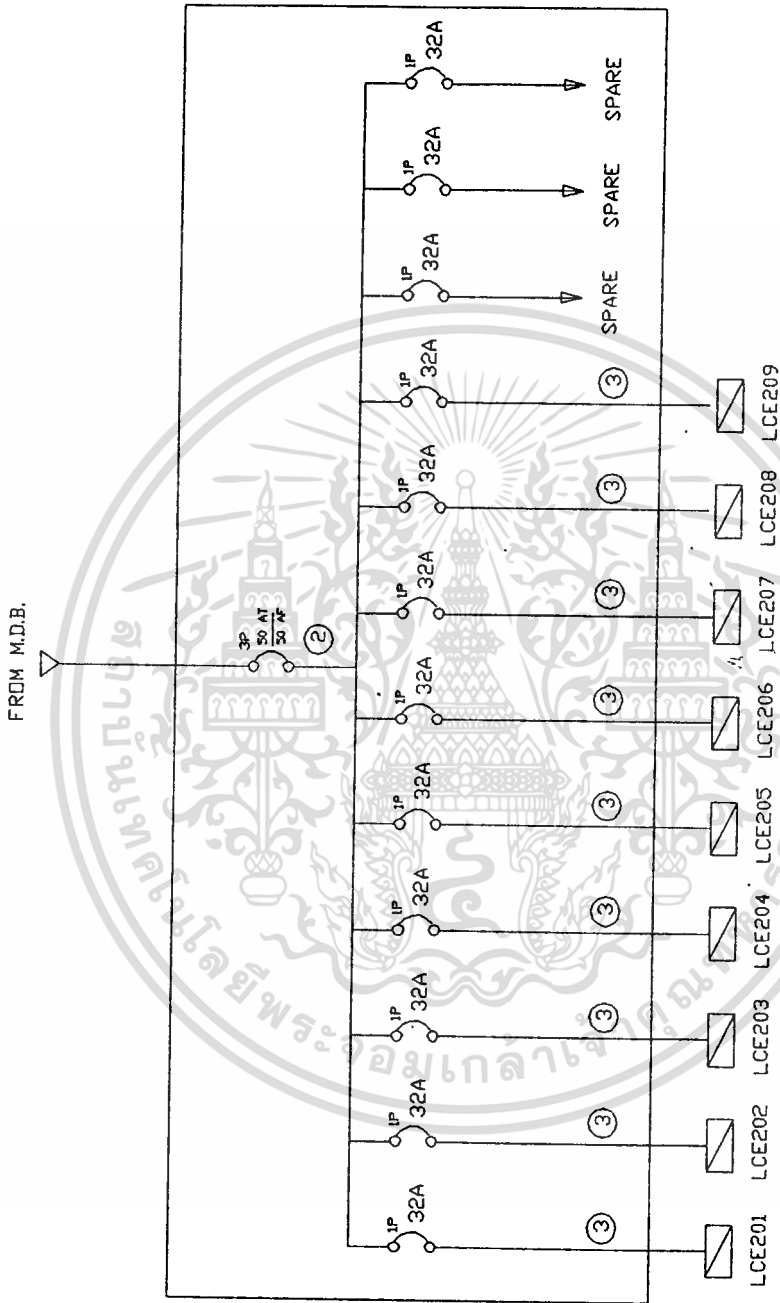
ELECTRICAL SYMBOLS			SYMBOLS
DWG NO.	LIST OF DRAWING		TRANSFORMER
EE-01	LIST OF DRAWING AND SYMBOLS		CIRCUIT BREAKER
EE-02	SINGLE LINE M.D.B. AND RISER DIAGRAM		LOAD CENTER
EE-03/1	SINGLE LINE LC-1		CONSUMER UNIT
EE-03/2	SINGLE LINE LC-2		SWITCH 1 GANG
EE-03/3	SINGLE LINE LC-3		SWITCH 2 GANG
EE-03/4	SINGLE LINE LCC		SWITCH 3 GANG
EE-04	LIGHTING AND RECEPTACLE FIRST FLOOR PLAN		TWO WAY SWITCH 1 GANG
EE-05	LIGHTING AND RECEPTACLE SECOND FLOOR PLAN		FL 40 W BARE TYPE
EE-06	LIGHTING AND RECEPTACLE THIRD FLOOR PLAN		FL 20 W BARE TYPE
			FL 20 W PASTIC COVER
			RECEPTACLE 2P WALL TYPE

PROJECT	LADKRABANG HOUSE	JOB NO	1
DWNER	LADKRABANG	SHEET NO	1
TITLE	LIST OF DRAWING AND SYMBOLS	TOTAL	9
DRAW BY	PP	DWG NO.	EE-01
DATE	6/1/95	SCALE	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

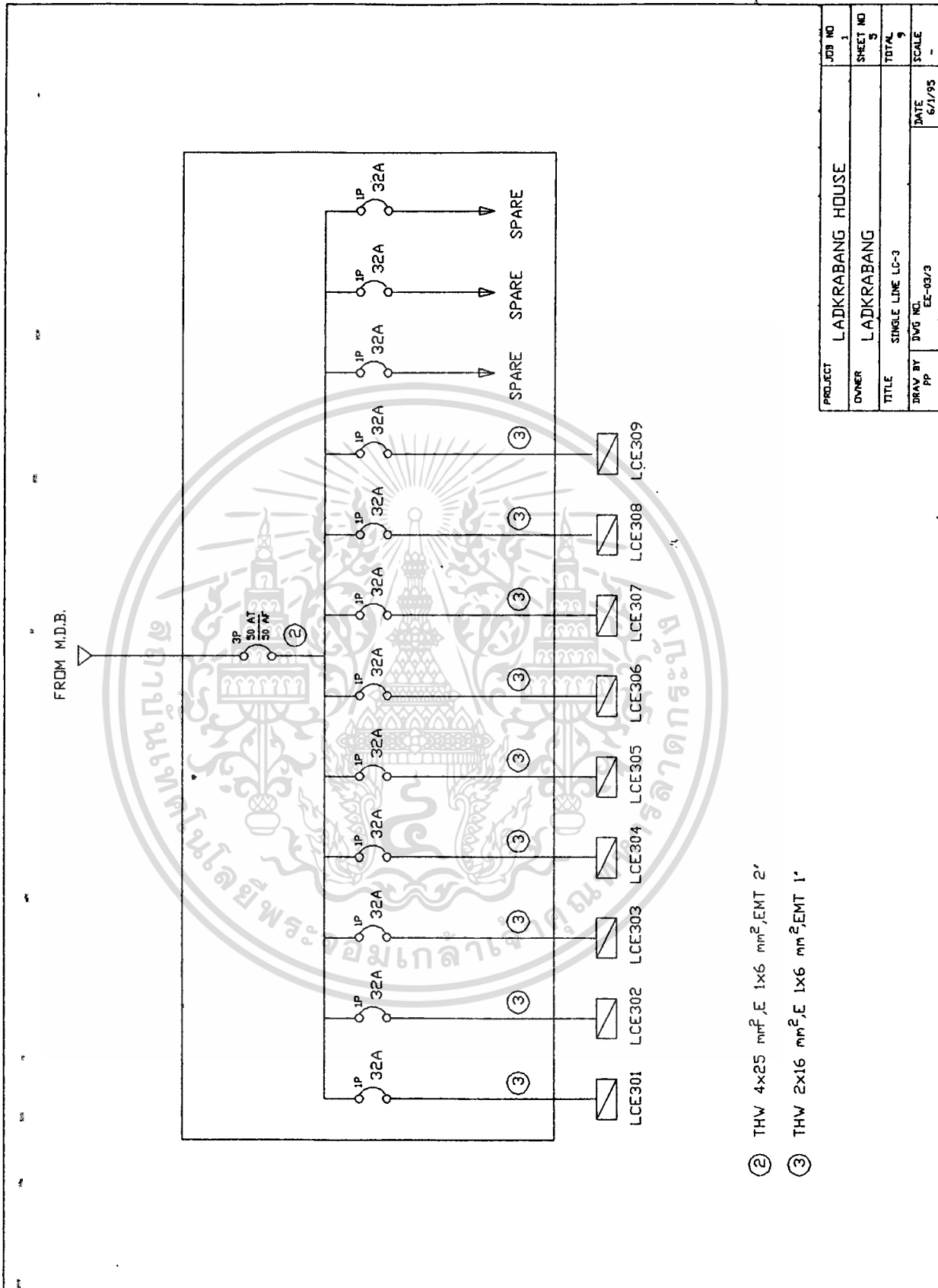


PROJECT	LADKRABANG HOUSE	JOB NO	1
OWNER	LADKRABANG	SHEET NO	2
TITLE	SINGLE LINE M.B. AND RUSER DIAGRAM	TOTAL	9
DRAW BY	PP	DATE	6/1/95
DWG NO.	EE-02	SCALE	-

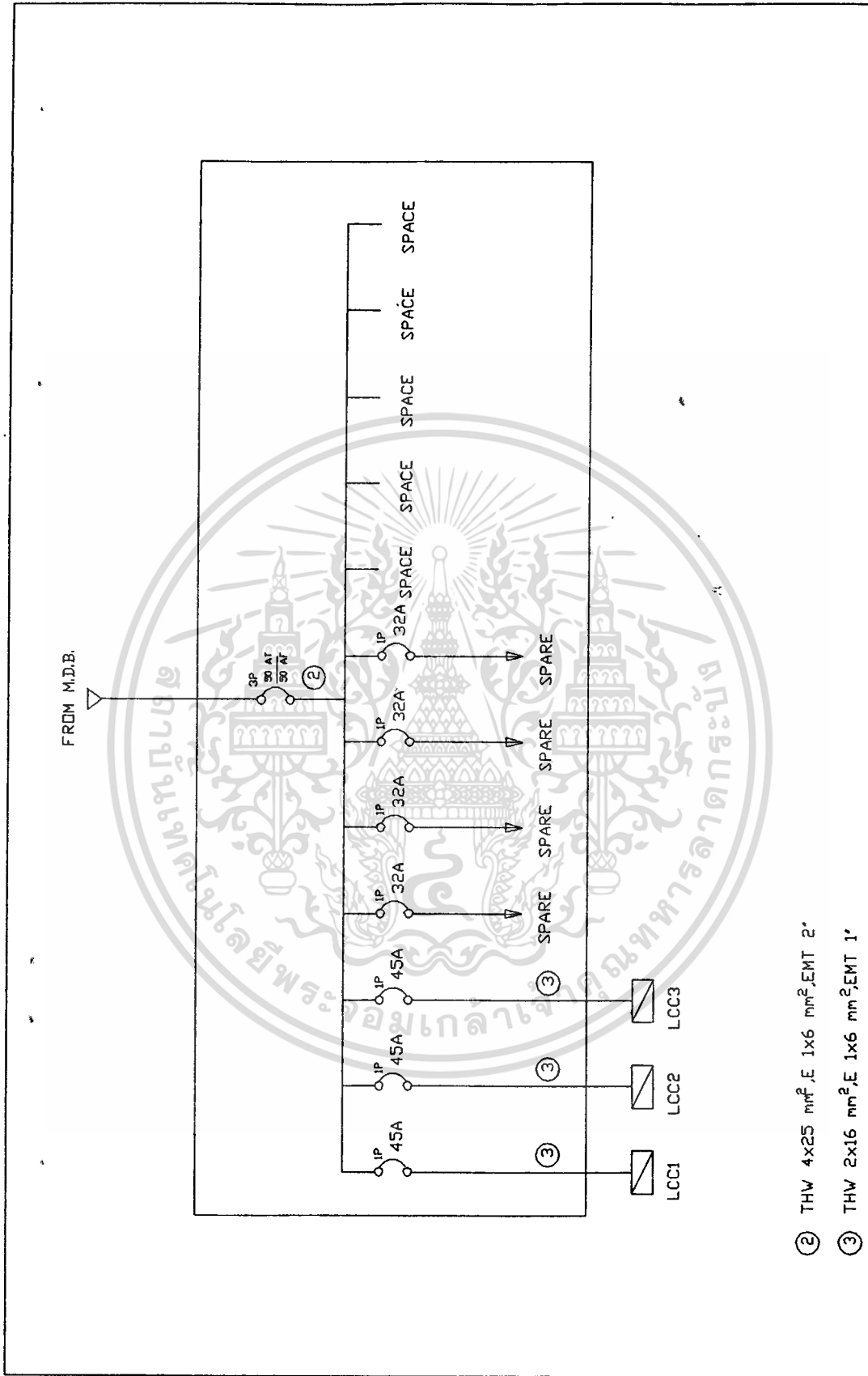


- ② THW 4x25 mm², E 1x6 mm², EMT 2'
- ③ THW 2x16 mm², E 1x6 mm², EMT 1'

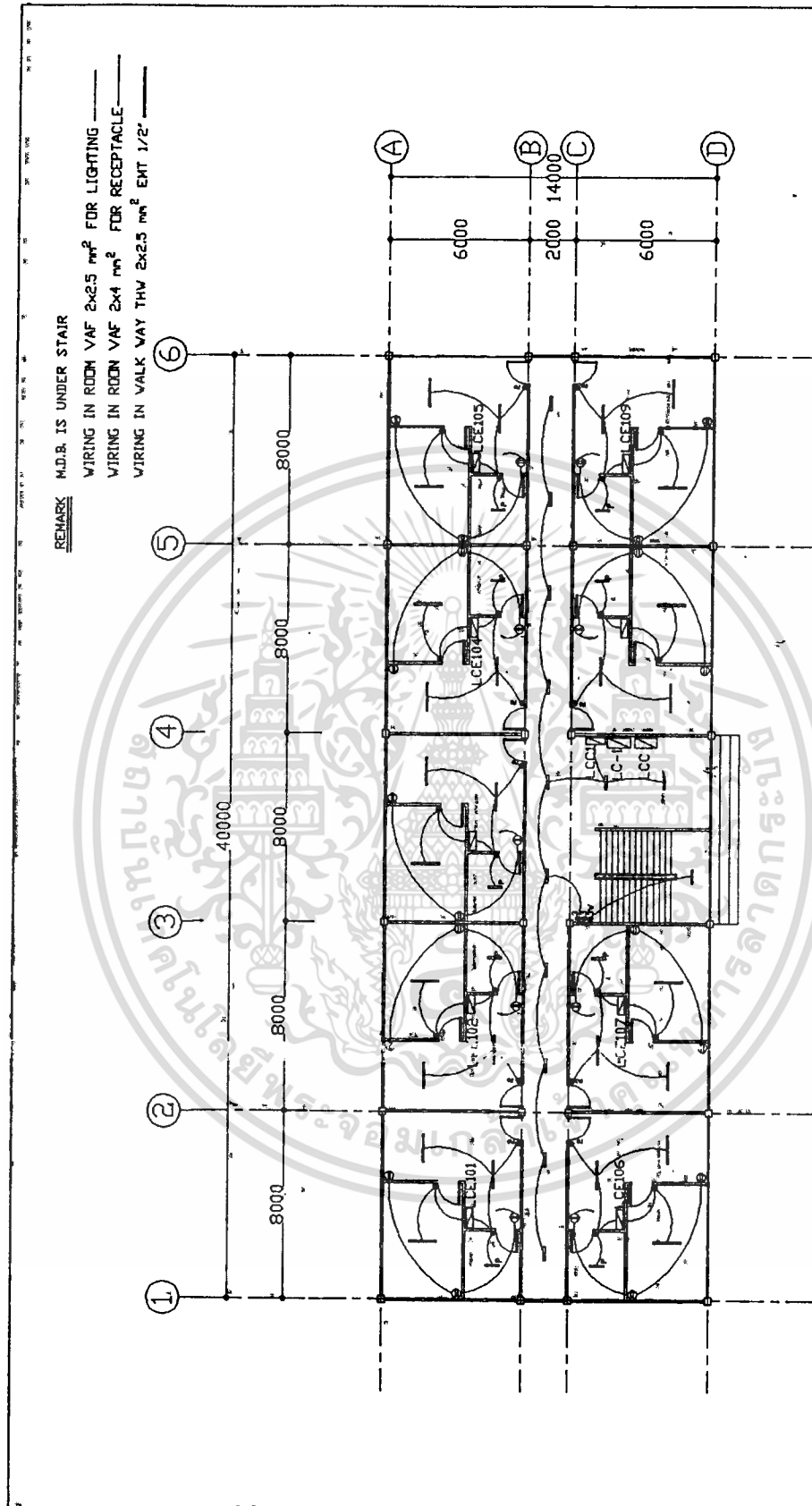
PROJECT	LADKRABANG HOUSE	JOB NO	1
DRAWER	LADKRABANG	SHEET NO	4
TITLE	SINGLE LINE LC-2	TOTAL	9
DRAW BY	DWG NO	DATE	SCALE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PROJECT	LADKRABANG HOUSE	JOB NO	1
OWNER	LADKRABANG	SHEET NO	6
TITLE	SINGLE LINE LCC	TOTAL	9
DRAWN BY	BVG NO.	DATE	11/11/2558

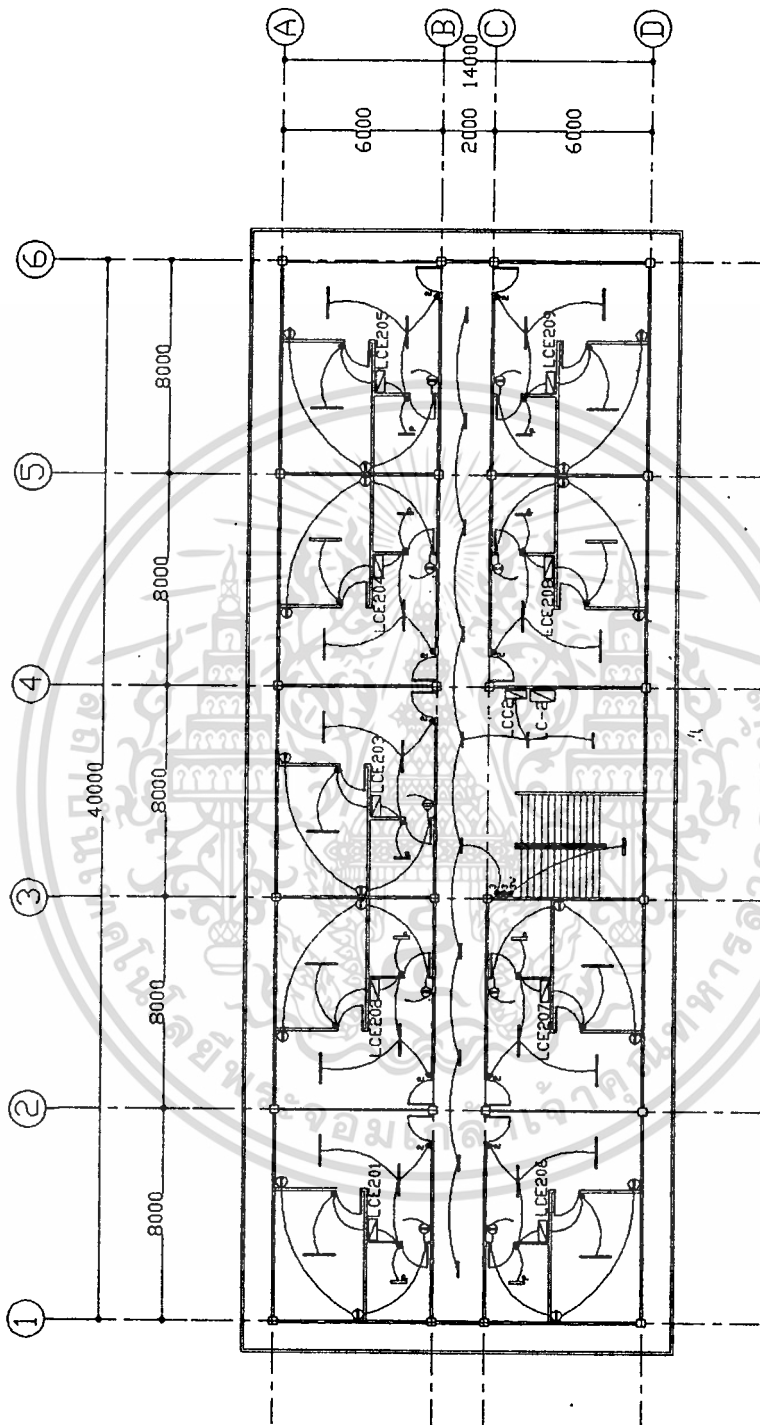


FIRST FLOOR PLAN

SCALE 1:250

PROJECT	LADKRABANG HOUSE	JOB NO	
DRAWN	LADKRABANG	SHEET NO	7
TITLE	LIGHTING AND RECEPTACLE FIRST FLOOR PLAN	TOTAL	9
DRAWN BY	THW-MT		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



SECOND FLOOR PLAN

SCALE 1:250

PROJECT	LADKRABANG HOUSE	JOB NO.	1
OWNER	LADKRABANG	SHEET NO.	8
TITLE	LIGHTING AND RECEPTACLE SECOND FLOOR PLAN	TOTAL	9
DRAW BY	DVG. NO.	DATE	6/1/95
PP	EE-05	SCALE	1:250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	HIGH VOLT	1	LOT	119600	31200	150800
2	PANEL BOARD	1	LOT	120462	20950	141412
3	LUMINAIRE	1	LOT	22425	3900	26325
4	SWITCH & RECEPTACLE	1	LOT	22563	3924	26487
5	CABLE	1	LOT	167828	43777	211605
6	RACE WAY	1	LOT	60685	15831	76516
sub-total				513563	119582	633145
overhead						18994
profit						31657
TOTAL						683797
vat						47865
GRAND TOTAL						731662

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 1 : HIGH VOLT QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	หม้อแปลง 24000-416v 100 KVA	1	SET	104000	31200	135200
2	อุปกรณ์เพิ่มเติม	1	LOT	15600.0	0	15600
sub-total				119600	31200	150800



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 2 : PANEL BOARD

QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	คอนซูมเมอร์ยูนิค 2P 4WAYS 10KA 32A	27	SET	1400.00	7560	45360
2	คอนซูมเมอร์ยูนิค 2P 4WAYS 10KA 45A	3	SET	1700.00	1020	6120
3	เมนเบรกเกอร์ 3P 50KA 160AF	1	EA	8000.00	1600	9600
4	เมนเบรกเกอร์ 3P 25KA 50AF	9	EA	2150.00	3870	23220
5	โหลดเซ็นเซอร์ 12WAYS 18KA 100A	4	SET	6400.00	5120	30720
6	เบรกเกอร์ย่อย 1P 5KA 32A	40	EA	200.00	1600	9600
7	เบรกเกอร์ย่อย 1P 5KA 45A	3	EA	300.00	180	1080
8	อุปกรณ์เพิ่มเติม	1	LOT	15712.0	0	15712
sub-total				120462	20950	141412

เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 3 : LUMINAIRE

QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	โคมครอบพลาสติกสวย1x20w	27	SET	180.00	972	5832
2	กลองเหล็กขาสปรง1x40W	81	SET	135.00	2187	13122
3	กลองเหล็กขาสปรง1x20W	39	SET	95.00	741	4446
4	อุปกรณ์เพิ่มเติม	1	LOT	2925.00	0	2925
sub-total				22425	3900	26325

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 4 : SWITCH & RECEPTICLE QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	สวทช์ ๑ ของ	54	EA	60.00	648	3888
2	สวทช์ ๒ ของ	27	EA	90.00	486	2916
3	เคำรบกคณขงแบบค	81	EA	160.00	2592	15552
4	สวทช์ ๓ ของ	6	EA	120.00	144	864
5	สวทช์ ๓ ทาง	3	EA	90.00	54	324
6	อุปกรณ์เต็มเคม	1	LOT	2943.00	0	2943
sub-total				22563	3924	26487

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 5 : CABLE

QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	สาย NYY 1 CORE 70 mm2	160	M	174.66	8383	36329
2	สาย NYY 1 CORE 16 mm2	40	M	52.80	633	2745
3	สาย THW 25 mm2	280	M	54.91	4612	19987
4	สาย THW 6 mm2	970	M	14.81	4309	18675
5	สาย THW 16 mm2	1800	M	36.62	19774	85690
6	สาย VAF 2 CORE 2.5 mm2	540	M	9.70	1571	6809
7	สาย VAF 2 CORE 4 mm2	810	M	15.40	3742	16216
8	สาย THW 2.5 mm2	420	M	5.98	753	3265
9	อุปกรณ์เพิ่มเติม	1	LOT	21890.0	0	21890
sub-total				167828	43777	211605

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT : LADKRABANG HOUSE

SUB 6 : RACE WAY

QUOTE PRICE

Item	Description	Qty	Unit	UnitPrice	Labour	Total
1	ท่อหนา IMC ขนาด 4"	40	M	363.00	4356	18876
2	ท่อบาง EMT ขนาด 2"	70	M	74.00	1554	6734
3	ท่อบาง EMT ขนาด 1"	880	M	34.00	8976	38896
4	ท่อบาง EMT ขนาด 1/2"	210	M	15.00	945	4095
5	อุปกรณ์เสริมเคม	1	LOT	7915.00	0	7915
sub-total				60685	15831	76516

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลและวิจารณ์

6.1 สรุปผลของโครงการ

จากผลการทดลอง จะเห็นได้ว่าโปรแกรมนี้สามารถประมาณราคาจากแบบแปลนไฟฟ้าได้และแสดงผลแยกออกเป็นประเภทต่างๆ ได้ ซึ่งทำให้ง่ายต่อการตรวจสอบและไม่ต้องเสียเวลาในการนับจำนวนอุปกรณ์ที่ใช้ในแบบแปลนไฟฟ้า ซึ่งก็เป็นไปตามเป้าหมายที่วางไว้

6.2 ข้อเสนอแนะในการพัฒนาโครงการ

6.2.1 โครงการนี้เป็นการประมาณราคาที่ยังไม่ครบสมบูรณ์ คือยังขาดระบบ MATV, TELEPHONE, FIRE ALARM, และ LIGHTNING & GROUNDING อยู่ ควรจะพัฒนาให้สามารถประมาณราคาได้ครบทุกระบบ

6.2.2 โครงการนี้การประมาณราคาแบบ WIRE , RACE WAY ยังต้องป้อนค่าจากคีย์บอร์ดอยู่ ดังนั้นควรจะพัฒนาให้สามารถประมาณราคาได้เลย โดยไม่ต้องป้อนความของสายไฟและท่อเดินสายไฟ

6.2.3 มาตรฐานของอุปกรณ์ไฟฟ้าที่ใช้เป็นฐานข้อมูล ไม่ได้ระบุว่าเป็นมาตรฐานใดหรือสินค้าของบริษัทใด ควรจะพัฒนาให้โปรแกรมนี้มีความสามารถใช้อุปกรณ์ได้หลายยี่ห้อคือสามารถให้ผู้ใช้งานโปรแกรมสามารถเลือกได้ว่าต้องการจะใช้สินค้ายี่ห้อใด



```

#include 'stdio.h'
#include "dos.h"
#include "conio.h"
#include "stdlib.h"
#define UP 0x4800
#define DOWN 0x5000
#define ENTER 0x1c0d
char *sp[8]={
    " ค้นหาจำนวนดวงโคบ      ",
    " เขียนแบบระบบไฟฟ้า      ",
    " ประมาณราคา      ",
    " แสดงผลทางจอภาพ      ",
    " แสดงผลทางเครื่องพิมพ์      ",
    " แก้ไขฐานข้อมูล      ",
    " เกี่ยวกับโปรแกรม      ",
    " ออกจากโปรแกรม...สู่ DOS "};
main()
{
    int i,n,x,y;
    x=26;y=9;n=0;
    clrscr();
    cursor_off();
    textbackground(0);
    table(1,1,79,24,1,7);
    textcolor(14);
    gotoxy(29,3);cprintf("<< ย้อน ดี ใหม่ รับ >>");
    textcolor(2);
    gotoxy(6,6);cprintf("โปรแกรม");
    gotoxy(19,6);cprintf("การเขียนแบบไฟฟ้าและประมาณราคาคด้วยคอมพิวเตอร์");
    gotoxy(6,9);cprintf("โดย: ");
    gotoxy(19,10);cprintf("1.นาย ชนุสิทธิ์ สุขประทุม รหัส 35.103056");
    gotoxy(19,11);cprintf("2.นาย พนมพร อภิศรพรหม รหัส 35.103066");
    gotoxy(19,12);cprintf("3.นาย พรพัฒน์ หงส์ดีสรกำชัย รหัส 35.103070");
    gotoxy(6,15);cprintf("อาจารย์ที่ปรึกษา");
    gotoxy(19,16);cprintf("รศ.สุธี บรรจงจิตร");
    gotoxy(19,17);cprintf("อ.สมชาติ จิรวิภากร");
}

```

```

gotoxy(6,21);cprintf("พัฒนาโดย... นักศึกษาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์");
gotoxy(19,22);cprintf("สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง (KMIT-L)");
getch();
do {
clrscr();
cursor_off();

textbackground(0);

clrscr();
table(1.1,79,24,1,9);

textcolor(2);

gotoxy(29,3);cprintf("<< ยี่ น คี ต์ อ น รั บ >>");
gotoxy(33,5);cprintf("โ ป ร แ ก ร ม");
gotoxy(20,6);cprintf("การเขียนแบบไฟฟ้าและประมาณราคาด้วยคอมพิวเตอร์");
gotoxy(24,20);cprintf("โปรดเลือกรายการที่ต้องการแล้วกด Enter");
gotoxy(6,22 );cprintf("พัฒนาโดย... นักศึกษาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบัง");
table(25,8,54,17,2,14);
textcolor(15);
textbackground(0);
for(i=0;i<8;i++)
{ textcolor(15);
gotoxy(x,y+i);
cprintf(sp[i]); }

n=select(x,y);
switch(n){
case 0:setdisk(3);clrscr();system("d:\_bat\cal.bat");break;
case 1:setdisk(3);clrscr();system("d:\_bat\acadrl2.bat");break;
case 2:setdisk(3);clrscr();system("d:\_bat\linker.bat");break;
case 3:setdisk(3);clrscr();system("d:\_bat\screen.bat");break;
case 4:setdisk(3);clrscr();system("d:\_bat\printer.bat");break;
case 5:setdisk(3);clrscr();system("d:\_bat\editdata bat");break;
case 6:setdisk(3);clrscr();system("d:\_bat\outtext.bat");break;
}

} while (n!=7);cursor_on();

clrscr(); exit(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select(int x,int y){
```

```
int n=0,in;
```

```
high(n,x,y,1):
```

```
for(;;){
```

```
in=bioskey(0);
```

```
switch(in){
```

```
case UP:n--;
```

```
if(n<0){
```

```
n=7;
```

```
high(n,x,y);
```

```
n=0;
```

```
low(n,x,y);
```

```
n=7;
```

```
}
```

```
else
```

```
{
```

```
high(n,x,y);
```

```
n++;
```

```
low(n,x,y);
```

```
n--;
```

```
}
```

```
break;
```

```
case DOWN: n++;
```

```
if(n>7){
```

```
n=0;
```

```
high(n,x,y);
```

```
n=7;
```

```
low(n,x,y);
```

```
n=0;
```

```
};
```

```
else
```

```
,
```

```
high(n,x,y);
```

```
n--;
```

```
low(n,x,y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
n++;
```

```
!
```

```
break;
```

```
case ENTER: return(n);
```

```
}}}
```

```
high(int n,int x,int y){
```

```
textcolor(14);
```

```
textbackground(9);
```

```
gotoxy(x,y+n);
```

```
cprintf(sp[n]);
```

```
}
```

```
low(int n,int x,int y){
```

```
textcolor(15);
```

```
textbackground(0);
```

```
gotoxy(x,y+n);
```

```
cprintf(sp[n]);
```

```
}
```

```
cursor_off(){
```

```
union REGS reg;
```

```
reg.h.ah=1;
```

```
reg.h.ch=0x20; /* 0x00 = on */
```

```
reg.h.cl=0; /* < 13 = on */
```

```
int86(0x10,&reg,&reg);}
```

```
cursor_on(){
```

```
union REGS reg;
```

```
reg.h.ah=1;
```

```
reg.h.ch=12; /* 0x00 = on */
```

```
reg.h.cl=13; /* < 13 = on */
```

```
int86(0x10,&reg,&reg);}
```

```
table(int x1, int y1, int x2, int y2, int t, int color)
```

```

int i,n0,n1,n2,n3,n4,n5,n6,n7,n8,n9;

textmode(3);

textcolor(color);

if(t==1){n0=201;n1=205;n2=187;n3=186;n4=186;n5=186;n6=200;n7=205;n8=188;n9=32;}

if(t==2){n0=213;n1=205;n2=184;n3=179;n4=179;n5=179;n6=212;n7=205;n8=190;n9=32;}

gotoxy(x1,y1);cprintf("%c",n0);

for(i=x1+1;i<=x2-1;i++){

gotoxy(i,y1);cprintf("%c",n1);}

gotoxy(x2,y1);cprintf("%c",n2);

gotoxy(x1,y1+1);cprintf("%c",n3);

gotoxy(x2,y1+1);cprintf("%c",n3);

gotoxy(x1,y1+2);cprintf("%c",n4);

gotoxy(x2,y1+2);cprintf("%c",n5);

for(i=x1+1;i<=x2-1;i++){

gotoxy(i,y1+2);cprintf("%c",n9); }

for(i=y1+3;i<=y2-1;i++){

gotoxy(x1,i);cprintf("%c",n3);}

for(i=y1+3;i<=y2-1;i++){

gotoxy(x2,i);cprintf("%c",n3);}

gotoxy(x1,y2);cprintf("%c",n6);

for(i=x1+1;i<=x2-1;i++){

gotoxy(i,y2);cprintf("%c",n7);}

gotoxy(x2,y2);cprintf("%c",n8);

textcolor(15);

textbackground(0);

textmode(-1);

}

***pop10

[Electrical]

[ATTEXT]cATTEXT S D:/FOX25/CONDO2.TXT D:/WRK/CONDO.TXT

[CHMENU]c^cmenu D:/ACADR12/SUPPORT/SYMBOLS.MNU

[--]

[.->HIGH VOLT]

[.->TRANSFORMER]

```

[TR30]^cINSERT D:/SYMBOLS/TR30.DWG
[TR50]^cINSERT D:/SYMBOLS/TR50.DWG
[TR100]^cINSERT D:/SYMBOLS/TR100.DWG
[TR160]^cINSERT D:/SYMBOLS/TR160.DWG
[TR250]^cINSERT D:/SYMBOLS/TR250.DWG
[TR315]^cINSERT D:/SYMBOLS/TR315.DWG
[TR400]^cINSERT D:/SYMBOLS/TR400.DWG
[TR500]^cINSERT D:/SYMBOLS/TR500.DWG
[TR630]^cINSERT D:/SYMBOLS/TR630.DWG
[TR800]^cINSERT D:/SYMBOLS/TR800.DWG
[TR1000]^cINSERT D:/SYMBOLS/TR1000.DWG
[TR1250]^cINSERT D:/SYMBOLS/TR1250.DWG
[TR1600]^cINSERT D:/SYMBOLS/TR1600.DWG
[TR2000]^cINSERT D:/SYMBOLS/TR2000.DWG
[TR2500]^cINSERT D:/SYMBOLS/TR2500.DWG
[TR3000]^cINSERT D:/SYMBOLS/TR3000.DWG
[TR3500]^cINSERT D:/SYMBOLS/TR3500.DWG
[TR4000]^cINSERT D:/SYMBOLS/TR4000.DWG
[TR5000]^cINSERT D:/SYMBOLS/TR5000.DWG
[<-<-TR12500]^cINSERT D:/SYMBOLS/TR12500.DWG
[->PANEL BOARD]
[->MCCB]
[B30AF]^cINSERT D:/SYMBOLS/B30AF.DWG
[B50AF]^cINSERT D:/SYMBOLS/B50AF.DWG
[B100AF]^cINSERT D:/SYMBOLS/B100AF.DWG
[B160AF]^cINSERT D:/SYMBOLS/B160AF.DWG
[B250AF]^cINSERT D:/SYMBOLS/B250AF.DWG
[B400AF]^cINSERT D:/SYMBOLS/B400AF.DWG
[B630AF]^cINSERT D:/SYMBOLS/B630AF.DWG
[B800AF]^cINSERT D:/SYMBOLS/B800AF.DWG
[B1000AF]^cINSERT D:/SYMBOLS/B1000AF.DWG
[B1250AF]^cINSERT D:/SYMBOLS/B1250AF.DWG
[B1600AF]^cINSERT D:/SYMBOLS/B1600AF.DWG
[B2000AF]^cINSERT D:/SYMBOLS/B2000AF.DWG
[B2500AF]^cINSERT D:/SYMBOLS/B2500AF.DWG

[B3200AF]^cINSERT D:/SYMBOLS/B3200AF.DWG
* [<-B4000AF]^cINSERT D:/SYMBOLS/B4000AF.DWG
[->ACB]
[BA1000]^cINSERT D:/SYMBOLS/BA1000.DWG
[BA1250]^cINSERT D:/SYMBOLS/BA1250.DWG
[BA1600]^cINSERT D:/SYMBOLS/BA1600.DWG
[BA2000]^cINSERT D:/SYMBOLS/BA2000.DWG
[BA3200]^cINSERT D:/SYMBOLS/BA3200.DWG
[BA4000]^cINSERT D:/SYMBOLS/BA4000.DWG
[<-BA5000]^cINSERT D:/SYMBOLS/BA5000.DWG
[->BBREAKER]
[BB10]^cINSERT D:/SYMBOLS/BB10.DWG
[BB16]^cINSERT D:/SYMBOLS/BB16.DWG
[BB20]^cINSERT D:/SYMBOLS/BB20.DWG
[BB32]^cINSERT D:/SYMBOLS/BB32.DWG
[BB45]^cINSERT D:/SYMBOLS/BB45.DWG
[BB63]^cINSERT D:/SYMBOLS/BB63.DWG
[BB70]^cINSERT D:/SYMBOLS/BB70.DWG
[BB80]^cINSERT D:/SYMBOLS/BB80.DWG
[<-BB100]^cINSERT D:/SYMBOLS/BB100.DWG
[->LOADCEN]
[BL12_100]^cINSERT D:/SYMBOLS/BL12_100.DWG
[BL18_100]^cINSERT D:/SYMBOLS/BL18_100.DWG
[BL24_100]^cINSERT D:/SYMBOLS/BL24_100.DWG
[BL30_100]^cINSERT D:/SYMBOLS/BL30_100.DWG
[BL12_200]^cINSERT D:/SYMBOLS/BL12_200.DWG
[BL18_200]^cINSERT D:/SYMBOLS/BL18_200.DWG
[BL24_200]^cINSERT D:/SYMBOLS/BL24_200.DWG
[BL30_200]^cINSERT D:/SYMBOLS/BL30_200.DWG
[BL36_200]^cINSERT D:/SYMBOLS/BL36_200.DWG
[<-BL42_200]^cINSERT D:/SYMBOLS/BL42_200.DWG
[->CONSUMER]
[BQ4_16]^cINSERT D:/SYMBOLS/BQ4_16.DWG
[BQ4_32]^cINSERT D:/SYMBOLS/BQ4_32.DWG
[BQ4_45]^cINSERT D:/SYMBOLS/BQ4_45.DWG

[BQ4_63]^cINSERT D:/SYMBOLS/BQ4_63.DWG
[BQ4_70]^cINSERT D:/SYMBOLS/BQ4_70.DWG
[BQ4_100]^cINSERT D:/SYMBOLS/BQ4_100.DWG
[BQ6_16]^cINSERT D:/SYMBOLS/BQ6_16.DWG
[BQ6_32]^cINSERT D:/SYMBOLS/BQ6_32.DWG
[BQ6_45]^cINSERT D:/SYMBOLS/BQ6_45.DWG
[BQ6_63]^cINSERT D:/SYMBOLS/BQ6_63.DWG
[BQ6_70]^cINSERT D:/SYMBOLS/BQ6_70.DWG
[BQ6_100]^cINSERT D:/SYMBOLS/BQ6_100.DWG
[BQ8_16]^cINSERT D:/SYMBOLS/BQ8_16.DWG
[BQ8_32]^cINSERT D:/SYMBOLS/BQ8_32.DWG
[BQ8_45]^cINSERT D:/SYMBOLS/BQ8_45.DWG
[BQ8_63]^cINSERT D:/SYMBOLS/BQ8_63.DWG
[BQ8_70]^cINSERT D:/SYMBOLS/BQ8_70.DWG
[BQ8_100]^cINSERT D:/SYMBOLS/BQ8_100.DWG
[BQ12_16]^cINSERT D:/SYMBOLS/BQ12_16.DWG
[BQ12_32]^cINSERT D:/SYMBOLS/BQ12_32.DWG
[BQ12_45]^cINSERT D:/SYMBOLS/BQ12_45.DWG
[BQ12_63]^cINSERT D:/SYMBOLS/BQ12_63.DWG
[BQ12_70]^cINSERT D:/SYMBOLS/BQ12_70.DWG
[BQ12_100]^cINSERT D:/SYMBOLS/BQ12_100.DWG
[BQ16_16]^cINSERT D:/SYMBOLS/BQ16_16.DWG
[BQ16_32]^cINSERT D:/SYMBOLS/BQ16_32.DWG
[BQ16_45]^cINSERT D:/SYMBOLS/BQ16_45.DWG
[BQ16_63]^cINSERT D:/SYMBOLS/BQ16_63.DWG
[BQ16_70]^cINSERT D:/SYMBOLS/BQ16_70.DWG
[<-<-BQ16_100]^cINSERT D:/SYMBOLS/BQ16_100.DWG
[->CONDUIT]
[->EMT]
[EMT1/2]^cINSERT D:/SYMBOLS/CEM1_2.DWG
[EMT3/4]^cINSERT D:/SYMBOLS/CEM3_4.DWG
[EMT1]^cINSERT D:/SYMBOLS/CEM1.DWG
[EMT1 1/4]^cINSERT D:/SYMBOLS/CEM1_1_4.DWG
[EMT1 1/2]^cINSERT D:/SYMBOLS/CEM1_1_2.DWG
[<-EMT2]^cINSERT D:/SYMBOLS/CEM2.DWG

[->IMC]

[IMC2]^cINSERT D:/SYMBOLS/CIM2.DWG

[IMC2 1/2]^cINSERT D:/SYMBOLS/CIM2_1_2.DWG

[IMC3]^cINSERT D:/SYMBOLS/CIM3.DWG

[IMC3 1/2]^cINSERT D:/SYMBOLS/CIM3_1_2.DWG

[<-IMC4]^cINSERT D:/SYMBOLS/CIM4.DWG

[->RSC]

[RSC2]^cINSERT D:/SYMBOLS/CRS2.DWG

[RSC2 1/2]^cINSERT D:/SYMBOLS/CRS2_1_2.DWG

[RSC3]^cINSERT D:/SYMBOLS/CRS3.DWG

[RSC3 1/2]^cINSERT D:/SYMBOLS/CRS3_1_2.DWG

[<-<-RSC4]^cINSERT D:/SYMBOLS/CRS4.DWG

[->CABLE]

[->THW]

[THW1.5]^cINSERT D:/SYMBOLS/WTHW1_5.DWG

[THW2.5]^cINSERT D:/SYMBOLS/WTHW2_5.DWG

[THW4]^cINSERT D:/SYMBOLS/WTHW4.DWG

[THW6]^cINSERT D:/SYMBOLS/WTHW6.DWG

[THW10]^cINSERT D:/SYMBOLS/WTHW10.DWG

[THW16]^cINSERT D:/SYMBOLS/WTHW16.DWG

[THW25]^cINSERT D:/SYMBOLS/WTHW25.DWG

[THW35]^cINSERT D:/SYMBOLS/WTHW35.DWG

[THW50]^cINSERT D:/SYMBOLS/WTHW50.DWG

[THW70]^cINSERT D:/SYMBOLS/WTHW70.DWG

[THW95]^cINSERT D:/SYMBOLS/WTHW95.DWG

[THW120]^cINSERT D:/SYMBOLS/WTHW120.DWG

[THW150]^cINSERT D:/SYMBOLS/WTHW150.DWG

[THW185]^cINSERT D:/SYMBOLS/WTHW185.DWG

[THW240]^cINSERT D:/SYMBOLS/WTHW240.DWG

[THW300]^cINSERT D:/SYMBOLS/WTHW300.DWG

[THW400]^cINSERT D:/SYMBOLS/WTHW400.DWG

[<-THW500]^cINSERT D:/SYMBOLS/WTHW500.DWG

[->NYY1C]

[NYY1x1.5]^cINSERT D:/SYMBOLS/W1NYY1_5.DWG

[NYY1x2.5]^cINSERT D:/SYMBOLS/W1NYY2_5.DWG

[NYY1x4]^cINSERT D:/SYMBOLS/W1NYY4.DWG
[NYY1x6]^cINSERT D:/SYMBOLS/W1NYY6.DWG
[NYY1x10]^cINSERT D:/SYMBOLS/W1NYY10.DWG
[NYY1x16]^cINSERT D:/SYMBOLS/W1NYY16.DWG
[NYY1x25]^cINSERT D:/SYMBOLS/W1NYY25.DWG
[NYY1x35]^cINSERT D:/SYMBOLS/W1NYY35.DWG
[NYY1x50]^cINSERT D:/SYMBOLS/W1NYY50.DWG
[NYY1x70]^cINSERT D:/SYMBOLS/W1NYY70.DWG
[NYY1x95]^cINSERT D:/SYMBOLS/W1NYY95.DWG
[NYY1x120]^cINSERT D:/SYMBOLS/W1NYY120.DWG
[NYY1x150]^cINSERT D:/SYMBOLS/W1NYY150.DWG
[NYY1x185]^cINSERT D:/SYMBOLS/W1NYY185.DWG
[NYY1x240]^cINSERT D:/SYMBOLS/W1NYY240.DWG
[<-NYY1x300]^cINSERT D:/SYMBOLS/W1NYY300.DWG
[->NYY3C]
[NYY3x1.5]^cINSERT D:/SYMBOLS/W3NYY1_1.5.DWG
[NYY3x2.5]^cINSERT D:/SYMBOLS/W3NYY2_1.5.DWG
[NYY3x4]^cINSERT D:/SYMBOLS/W3NYY4.DWG
[NYY3x6]^cINSERT D:/SYMBOLS/W3NYY6.DWG
[NYY3x10]^cINSERT D:/SYMBOLS/W3NYY10.DWG
[NYY3x16]^cINSERT D:/SYMBOLS/W3NYY16.DWG
[NYY3x25]^cINSERT D:/SYMBOLS/W3NYY25.DWG
[NYY3x35]^cINSERT D:/SYMBOLS/W3NYY35.DWG
[NYY3x50]^cINSERT D:/SYMBOLS/W3NYY50.DWG
[NYY3x70]^cINSERT D:/SYMBOLS/W3NYY70.DWG
[NYY3x95]^cINSERT D:/SYMBOLS/W3NYY95.DWG
[NYY3x120]^cINSERT D:/SYMBOLS/W3NYY120.DWG
[NYY3x150]^cINSERT D:/SYMBOLS/W3NYY150.DWG
[NYY3x185]^cINSERT D:/SYMBOLS/W3NYY185.DWG
[NYY3x240]^cINSERT D:/SYMBOLS/W3NYY240.DWG
[<-NYY3x300]^cINSERT D:/SYMBOLS/W3NYY300.DWG
[->NYY4C]
[NYY4x1.5]^cINSERT D:/SYMBOLS/W4NYY1_1.5.DWG
[NYY4x2.5]^cINSERT D:/SYMBOLS/W4NYY2_1.5.DWG
[NYY4x4]^cINSERT D:/SYMBOLS/W4NYY4.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[NYY4x6]^cINSERT D:/SYMBOLS/W4NYY6.DWG

[NYY4x10]^cINSERT D:/SYMBOLS/W4NYY10.DWG

[NYY4x16]^cINSERT D:/SYMBOLS/W4NYY16.DWG

[NYY4x25]^cINSERT D:/SYMBOLS/W4NYY25.DWG

[NYY4x35]^cINSERT D:/SYMBOLS/W4NYY35.DWG

[NYY4x50]^cINSERT D:/SYMBOLS/W4NYY50.DWG

[NYY4x70]^cINSERT D:/SYMBOLS/W4NYY70.DWG

[NYY4x95]^cINSERT D:/SYMBOLS/W4NYY95.DWG

[NYY4x120]^cINSERT D:/SYMBOLS/W4NYY120.DWG

[NYY4x150]^cINSERT D:/SYMBOLS/W4NYY150.DWG

[NYY4x185]^cINSERT D:/SYMBOLS/W4NYY185.DWG

[NYY4x240]^cINSERT D:/SYMBOLS/W4NYY240.DWG

[<-NYY4x300]^cINSERT D:/SYMBOLS/W4NYY300.DWG

[>-VAF]

[VAF2x1]^cINSERT D:/SYMBOLS/W2VAF1.DWG

[VAF2x1.5]^cINSERT D:/SYMBOLS/W2VAF1_5.DWG

[VAF2x2.5]^cINSERT D:/SYMBOLS/W2VAF2_5.DWG

[VAF2x4]^cINSERT D:/SYMBOLS/W2VAF4.DWG

[VAF2x6]^cINSERT D:/SYMBOLS/W2VAF6.DWG

[VAF2x10]^cINSERT D:/SYMBOLS/W2VAF10.DWG

[<-<-VAF2x16]^cINSERT D:/SYMBOLS/W2VAF16.DWG

[>-LIGHTING]

[>-INCAN]

[L033_SM]^cINSERT D:/SYMBOLS/L033_SM.DWG

[L033_MD]^cINSERT D:/SYMBOLS/L033_MD.DWG

[L033_LG]^cINSERT D:/SYMBOLS/L033_LG.DWG

[L035_6I]^cINSERT D:/SYMBOLS/L035_6I.DWG

[L035_8I]^cINSERT D:/SYMBOLS/L035_8I.DWG

[L035_10I]^cINSERT D:/SYMBOLS/L035_10I.DWG

[L035_12I]^cINSERT D:/SYMBOLS/L035_12I.DWG

[L036_SM]^cINSERT D:/SYMBOLS/L036_SM.DWG

[L037]^cINSERT D:/SYMBOLS/L037.DWG

[L038_6I]^cINSERT D:/SYMBOLS/L038_6I.DWG

[L038_8I]^cINSERT D:/SYMBOLS/L038_8I.DWG

[L038_10I]^cINSERT D:/SYMBOLS/L038_10I.DWG

[L038_12I]^cINSERT D:/SYMBOLS/L038_12I.DWG

* [L038_14I]^cINSERT D:/SYMBOLS/L038_14I.DWG

[<-L038_16I]^cINSERT D:/SYMBOLS/L038_16I.DWG

[>-HID]

* [->MERCURY]

[L040_60]^cINSERT D:/SYMBOLS/L040_60.DWG

[L040_250]^cINSERT D:/SYMBOLS/L040_250.DWG

[L041_16I]^cINSERT D:/SYMBOLS/L041_16I.DWG

[L041_20I]^cINSERT D:/SYMBOLS/L041_20I.DWG

* [L042_SL1]^cINSERT D:/SYMBOLS/L042_SL1.DWG

[L042_SL2]^cINSERT D:/SYMBOLS/L042_SL2.DWG

[L042_SL3]^cINSERT D:/SYMBOLS/L042_SL3.DWG

[LHDK80]^cINSERT D:/SYMBOLS/LHDK80.DWG

[LHDK125]^cINSERT D:/SYMBOLS/LHDK125.DWG

[LHDK250]^cINSERT D:/SYMBOLS/LHDK250.DWG

[LHDK400]^cINSERT D:/SYMBOLS/LHDK400.DWG

[<-LHDK1000]^cINSERT D:/SYMBOLS/LHDK1000.DWG

[>-METAL]

[LHNF250]^cINSERT D:/SYMBOLS/LHNF250.DWG

[LHNF400]^cINSERT D:/SYMBOLS/LHNF400.DWG

[LHNF1000]^cINSERT D:/SYMBOLS/LHNF1000.DWG

[<-LHNF2000]^cINSERT D:/SYMBOLS/LHNF2000.DWG

[>-SODIUM]

[LSDK70]^cINSERT D:/SYMBOLS/LSDK70.DWG

[LSDK150]^cINSERT D:/SYMBOLS/LSDK150.DWG

[LSDK250]^cINSERT D:/SYMBOLS/LSDK250.DWG

* [<-<-LSDK400]^cINSERT D:/SYMBOLS/LSDK400.DWG

[>-PL]

[LSL09]^cINSERT D:/SYMBOLS/LSL09.DWG

[LSL13]^cINSERT D:/SYMBOLS/LSL13.DWG

[LSL18]^cINSERT D:/SYMBOLS/LSL18.DWG

* [LSL25]^cINSERT D:/SYMBOLS/LSL25.DWG

[LPLC09]^cINSERT D:/SYMBOLS/LPLC09.DWG

[LPLC11]^cINSERT D:/SYMBOLS/LPLC11.DWG

[LPLC15]^cINSERT D:/SYMBOLS/LPLC15.DWG

[<-LPLC20]^cINSERT D:/SYMBOLS/LPLC20.DWG

[>-FLUORESCENT]

● [->FL20]

[L006_20]^cINSERT D:/SYMBOLS/L006_20.DWG

[L006V_20]^cINSERT D:/SYMBOLS/L006V_20.DWG

[L007_20]^cINSERT D:/SYMBOLS/L007_20.DWG

[L008_20]^cINSERT D:/SYMBOLS/L008_20.DWG

[L009_20]^cINSERT D:/SYMBOLS/L009_20.DWG

[L010_20]^cINSERT D:/SYMBOLS/L010_20.DWG

[L011_20]^cINSERT D:/SYMBOLS/L011_20.DWG

[L012_20]^cINSERT D:/SYMBOLS/L012_20.DWG

[L014_20]^cINSERT D:/SYMBOLS/L014_20.DWG

[L015_20]^cINSERT D:/SYMBOLS/L015_20.DWG

[L016_20]^cINSERT D:/SYMBOLS/L016_20.DWG

[L017_20]^cINSERT D:/SYMBOLS/L017_20.DWG

[L018_20]^cINSERT D:/SYMBOLS/L018_20.DWG

[L019_20]^cINSERT D:/SYMBOLS/L019_20.DWG

[L020_20]^cINSERT D:/SYMBOLS/L020_20.DWG

[L021_20]^cINSERT D:/SYMBOLS/L021_20.DWG

[L025_20]^cINSERT D:/SYMBOLS/L025_20.DWG

[L026_20]^cINSERT D:/SYMBOLS/L026_20.DWG

[L028_20A]^cINSERT D:/SYMBOLS/L028_20A.DWG

[L028_20M]^cINSERT D:/SYMBOLS/L028_20M.DWG

[L029_20]^cINSERT D:/SYMBOLS/L029_20.DWG

[L029_20F]^cINSERT D:/SYMBOLS/L029_20F.DWG

[L030_20]^cINSERT D:/SYMBOLS/L030_20.DWG

[L030_20F]^cINSERT D:/SYMBOLS/L030_20F.DWG

[L043_1x2]^cINSERT D:/SYMBOLS/L043_1x2.DWG

[L043_2x2]^cINSERT D:/SYMBOLS/L043_2x2.DWG

[L055_1x2]^cINSERT D:/SYMBOLS/L055_1x2.DWG

[L055_2x2]^cINSERT D:/SYMBOLS/L055_2x2.DWG

[L055_3x2]^cINSERT D:/SYMBOLS/L055_3x2.DWG

[<-L055_4x2]^cINSERT D:/SYMBOLS/L055_4x2.DWG

[>-FL32]

[L031_8I]^cINSERT D:/SYMBOLS/L031_8I.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[L031_10I]^cINSERT D:/SYMBOLS/L031_10I.DWG
[L031_12I]^cINSERT D:/SYMBOLS/L031_12I.DWG
[L032_6I]^cINSERT D:/SYMBOLS/L032_6I.DWG
[L032_8I]^cINSERT D:/SYMBOLS/L032_8I.DWG
[L032_10I]^cINSERT D:/SYMBOLS/L032_10I.DWG
[<-L032_12I]^cINSERT D:/SYMBOLS/L032_12I.DWG
[->FL40]
[L006_40]^cINSERT D:/SYMBOLS/L006_40.DWG
[L006V_40]^cINSERT D:/SYMBOLS/L006V_40.DWG
[L007_40]^cINSERT D:/SYMBOLS/L007_40.DWG
[L008_40]^cINSERT D:/SYMBOLS/L008_40.DWG
[L009_40]^cINSERT D:/SYMBOLS/L009_40.DWG
[L011_40]^cINSERT D:/SYMBOLS/L011_40.DWG
[L012_40]^cINSERT D:/SYMBOLS/L012_40.DWG
[L014_40]^cINSERT D:/SYMBOLS/L014_40.DWG
[L015_40]^cINSERT D:/SYMBOLS/L015_40.DWG
[L016_40]^cINSERT D:/SYMBOLS/L016_40.DWG
[L017_40]^cINSERT D:/SYMBOLS/L017_40.DWG
[L018_40]^cINSERT D:/SYMBOLS/L018_40.DWG
[L019_40]^cINSERT D:/SYMBOLS/L019_40.DWG
[L020_40]^cINSERT D:/SYMBOLS/L020_40.DWG
[L021_40]^cINSERT D:/SYMBOLS/L021_40.DWG
[L025_40]^cINSERT D:/SYMBOLS/L025_40.DWG
[L026_40]^cINSERT D:/SYMBOLS/L026_40.DWG
[L027_40A]^cINSERT D:/SYMBOLS/L027_40A.DWG
[L027_40M]^cINSERT D:/SYMBOLS/L027_40M.DWG
[L028_40A]^cINSERT D:/SYMBOLS/L028_40A.DWG
[L028_40M]^cINSERT D:/SYMBOLS/L028_40M.DWG
[L029_40]^cINSERT D:/SYMBOLS/L029_40.DWG
[L029_40F]^cINSERT D:/SYMBOLS/L029_40F.DWG
[L030_40]^cINSERT D:/SYMBOLS/L030_40.DWG
[L030_40F]^cINSERT D:/SYMBOLS/L030_40F.DWG
[L043_1x4]^cINSERT D:/SYMBOLS/L043_1x4.DWG
[L043_2x4]^cINSERT D:/SYMBOLS/L043_2x4.DWG
[L044_LG]^cINSERT D:/SYMBOLS/L044_LG.DWG

```
[L044_SM]^cINSERT D:/SYMBOLS/L044_SM.DWG
[L055_1x4]^cINSERT D:/SYMBOLS/L055_1x4.DWG
[L055_2x4]^cINSERT D:/SYMBOLS/L055_2x4.DWG
[L055_3x4]^cINSERT D:/SYMBOLS/L055_3x4.DWG
[L055_4x4]^cINSERT D:/SYMBOLS/L055_4x4.DWG
[L024]^cINSERT D:/SYMBOLS/L024.DWG
[L039_CO]^cINSERT D:/SYMBOLS/L039_CO.DWG
[<-<-<-L039_CL]^cINSERT D:/SYMBOLS/L039_CL.DWG
```

[>SWITCH]

```
[S1]^cINSERT D:/SYMBOLS/S1.DWG
[S2]^cINSERT D:/SYMBOLS/S2.DWG
[S3]^cINSERT D:/SYMBOLS/S3.DWG
[<-S3w]^cINSERT D:/SYMBOLS/S3w.DWG
```

[>PLUG]

```
[SRWS]^cINSERT D:/SYMBOLS/SRWS.DWG
[SRWD]^cINSERT D:/SYMBOLS/SRWD.DWG
[<-SRF]^cINSERT D:/SYMBOLS/SRF.DWG
```

***BUTTONS1

;

\$p0=*

^C^C

^B

^O

^G

^D

^E

^T

***BUTTONS2

\$p0=*

***AUX1

;

\$p0=*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**ร 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



[-LAST_]๙๙= ๙๙=

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

****1 HIGH 3**

[TR]¹^c\$_s=x \$_s=TR

[HBREAKER]¹^c\$_s=x \$_s=HBREAKER

****TR 3**

[TR30]¹^cINSERT D:/SYMBOLS/TR30.DWG

[TR50]¹^cINSERT D:/SYMBOLS/TR50.DWG

[TR100]¹^cINSERT D:/SYMBOLS/TR100.DWG

[TR160]¹^cINSERT D:/SYMBOLS/TR160.DWG

[TR250]¹^cINSERT D:/SYMBOLS/TR250.DWG

[TR315]¹^cINSERT D:/SYMBOLS/TR315.DWG

[TR400]¹^cINSERT D:/SYMBOLS/TR400.DWG

[TR500]¹^cINSERT D:/SYMBOLS/TR500.DWG

[TR630]¹^cINSERT D:/SYMBOLS/TR630.DWG

[TR800]¹^cINSERT D:/SYMBOLS/TR800.DWG

[TR1000]¹^cINSERT D:/SYMBOLS/TR1000.DWG

[TR1250]¹^cINSERT D:/SYMBOLS/TR1250.DWG

[TR1600]¹^cINSERT D:/SYMBOLS/TR1600.DWG

[TR2000]¹^cINSERT D:/SYMBOLS/TR2000.DWG

[TR2500]¹^cINSERT D:/SYMBOLS/TR2500.DWG

[TR3000]¹^cINSERT D:/SYMBOLS/TR3000.DWG

[TR3500]¹^cINSERT D:/SYMBOLS/TR3500.DWG

[TR4000]¹^cINSERT D:/SYMBOLS/TR4000.DWG

[TR5000]¹^cINSERT D:/SYMBOLS/TR5000.DWG

[TR12500]¹^cINSERT D:/SYMBOLS/TR12500.DWG

****HBREAKER 3**

[MCCB]¹^c\$_s=x \$_s=MCCB

[ACB]¹^c\$_s=x \$_s=ACB

****MCCB 3**

[B30AF]¹^cINSERT D:/SYMBOLS/B30AF.DWG

[B50AF]¹^cINSERT D:/SYMBOLS/B50AF.DWG

[B100AF]¹^cINSERT D:/SYMBOLS/B100AF.DWG

[B160AF]¹^cINSERT D:/SYMBOLS/B160AF.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[B250AF]^cINSERT D:/SYMBOLS/B250AF.DWG
[B400AF]^cINSERT D:/SYMBOLS/B400AF.DWG
[B630AF]^cINSERT D:/SYMBOLS/B630AF.DWG
[B800AF]^cINSERT D:/SYMBOLS/B800AF.DWG
[B1000AF]^cINSERT D:/SYMBOLS/B1000AF.DWG
[B1250AF]^cINSERT D:/SYMBOLS/B1250AF.DWG
[B1600AF]^cINSERT D:/SYMBOLS/B1600AF.DWG
[B2000AF]^cINSERT D:/SYMBOLS/B2000AF.DWG
[B2500AF]^cINSERT D:/SYMBOLS/B2500AF.DWG
[B3200AF]^cINSERT D:/SYMBOLS/B3200AF.DWG
[B4000AF]^cINSERT D:/SYMBOLS/B4000AF.DWG

****ACB 3**

[BA1000]^cINSERT D:/SYMBOLS/BA1000.DWG
[BA1250]^cINSERT D:/SYMBOLS/BA1250.DWG
[BA1600]^cINSERT D:/SYMBOLS/BA1600.DWG
[BA2000]^cINSERT D:/SYMBOLS/BA2000.DWG
[BA3200]^cINSERT D:/SYMBOLS/BA3200.DWG
[BA4000]^cINSERT D:/SYMBOLS/BA4000.DWG
[BA5000]^cINSERT D:/SYMBOLS/BA5000.DWG

****2PANEL 3**

[MCCB]^c\$S=x \$S=MCCB
[BBREAKER]^c\$S=x \$S=BBREAKER
[LOADCEN]^c\$S=x \$S=LOADCEN
[CONSUMER]^c\$S=x \$S=CONSUMER

****BBREAKER 3**

[BB10]^cINSERT D:/SYMBOLS/BB10.DWG
[BB16]^cINSERT D:/SYMBOLS/BB16.DWG
[BB20]^cINSERT D:/SYMBOLS/BB20.DWG
[BB32]^cINSERT D:/SYMBOLS/BB32.DWG
[BB45]^cINSERT D:/SYMBOLS/BB45.DWG
[BB63]^cINSERT D:/SYMBOLS/BB63.DWG
[BB70]^cINSERT D:/SYMBOLS/BB70.DWG
[BB80]^cINSERT D:/SYMBOLS/BB80.DWG
[BB100]^cINSERT D:/SYMBOLS/BB100.DWG

****LOADCEN 3**

[BL12_100]^cINSERT D:/SYMBOLS/BL12_100.DWG
[BL18_100]^cINSERT D:/SYMBOLS/BL18_100.DWG
[BL24_100]^cINSERT D:/SYMBOLS/BL24_100.DWG
[BL30_100]^cINSERT D:/SYMBOLS/BL30_100.DWG
[BL12_200]^cINSERT D:/SYMBOLS/BL12_200.DWG
[BL18_200]^cINSERT D:/SYMBOLS/BL18_200.DWG
[BL24_200]^cINSERT D:/SYMBOLS/BL24_200.DWG
[BL30_200]^cINSERT D:/SYMBOLS/BL30_200.DWG
[BL36_200]^cINSERT D:/SYMBOLS/BL36_200.DWG
[BL42_200]^cINSERT D:/SYMBOLS/BL42_200.DWG

****CONSUMER 3**

[BQ4_16]^cINSERT D:/SYMBOLS/BQ4_16.DWG
[BQ4_32]^cINSERT D:/SYMBOLS/BQ4_32.DWG
[BQ4_45]^cINSERT D:/SYMBOLS/BQ4_45.DWG
[BQ4_63]^cINSERT D:/SYMBOLS/BQ4_63.DWG
[BQ4_70]^cINSERT D:/SYMBOLS/BQ4_70.DWG
[BQ4_100]^cINSERT D:/SYMBOLS/BQ4_100.DWG
[BQ6_16]^cINSERT D:/SYMBOLS/BQ6_16.DWG
[BQ6_32]^cINSERT D:/SYMBOLS/BQ6_32.DWG
[BQ6_45]^cINSERT D:/SYMBOLS/BQ6_45.DWG
[BQ6_63]^cINSERT D:/SYMBOLS/BQ6_63.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[BQ6_70]^cINSERT D:/SYMBOLS/BQ6_70.DWG
[BQ6_100]^cINSERT D:/SYMBOLS/BQ6_100.DWG
[BQ8_16]^cINSERT D:/SYMBOLS/BQ8_16.DWG
[BQ8_32]^cINSERT D:/SYMBOLS/BQ8_32.DWG
[BQ8_45]^cINSERT D:/SYMBOLS/BQ8_45.DWG
[BQ8_63]^cINSERT D:/SYMBOLS/BQ8_63.DWG
[BQ8_70]^cINSERT D:/SYMBOLS/BQ8_70.DWG
[BQ8_100]^cINSERT D:/SYMBOLS/BQ8_100.DWG
[BQ12_16]^cINSERT D:/SYMBOLS/BQ12_16.DWG
[BQ12_32]^cINSERT D:/SYMBOLS/BQ12_32.DWG
[BQ12_45]^cINSERT D:/SYMBOLS/BQ12_45.DWG
[BQ12_63]^cINSERT D:/SYMBOLS/BQ12_63.DWG
[BQ12_70]^cINSERT D:/SYMBOLS/BQ12_70.DWG
[BQ12_100]^cINSERT D:/SYMBOLS/BQ12_100.DWG
[BQ16_16]^cINSERT D:/SYMBOLS/BQ16_16.DWG
[BQ16_32]^cINSERT D:/SYMBOLS/BQ16_32.DWG
[BQ16_45]^cINSERT D:/SYMBOLS/BQ16_45.DWG
[BQ16_63]^cINSERT D:/SYMBOLS/BQ16_63.DWG
[BQ16_70]^cINSERT D:/SYMBOLS/BQ16_70.DWG
[BQ16_100]^cINSERT D:/SYMBOLS/BQ16_100.DWG

****3LIGHTING 3**

[FL]^c\$S=x \$S=FL
[INCAN]^c\$S=x \$S=INCAN
[PL]^c\$S=x \$S=PL
[HID]^c\$S=x \$S=HID
[SW]^c\$S=x \$S=SW

****FL 3**

[FL20]^c\$S=x \$S=FL20
[FL32]^c\$S=x \$S=FL32
[FL40]^c\$S=x \$S=FL40

[MERCURY]^c\$=x \$=MERCURY

[METAL]^c\$=x \$=METAL

[SODIUM]^c\$=x \$=SODIUM

****FL20 3**

[L006_20]^cINSERT D:/SYMBOLS/L006_20.DWG

[L006V_20]^cINSERT D:/SYMBOLS/L006V_20.DWG

[L007_20]^cINSERT D:/SYMBOLS/L007_20.DWG

[L008_20]^cINSERT D:/SYMBOLS/L008_20.DWG

[L009_20]^cINSERT D:/SYMBOLS/L009_20.DWG

[L010_20]^cINSERT D:/SYMBOLS/L010_20.DWG

[L011_20]^cINSERT D:/SYMBOLS/L011_20.DWG

[L012_20]^cINSERT D:/SYMBOLS/L012_20.DWG

[L014_20]^cINSERT D:/SYMBOLS/L014_20.DWG

[L015_20]^cINSERT D:/SYMBOLS/L015_20.DWG

[L016_20]^cINSERT D:/SYMBOLS/L016_20.DWG

[L017_20]^cINSERT D:/SYMBOLS/L017_20.DWG

[L018_20]^cINSERT D:/SYMBOLS/L018_20.DWG

[L019_20]^cINSERT D:/SYMBOLS/L019_20.DWG

[L020_20]^cINSERT D:/SYMBOLS/L020_20.DWG

[L021_20]^cINSERT D:/SYMBOLS/L021_20.DWG

[L025_20]^cINSERT D:/SYMBOLS/L025_20.DWG

[L026_20]^cINSERT D:/SYMBOLS/L026_20.DWG

[L028_20A]^cINSERT D:/SYMBOLS/L028_20A.DWG

[L028_20M]^cINSERT D:/SYMBOLS/L028_20M.DWG

[L029_20]^cINSERT D:/SYMBOLS/L029_20.DWG

[L029_20F]^cINSERT D:/SYMBOLS/L029_20F.DWG

[L030_20]^cINSERT D:/SYMBOLS/L030_20.DWG

[L030_20F]^cINSERT D:/SYMBOLS/L030_20F.DWG

[L043_1x2]^cINSERT D:/SYMBOLS/L043_1x2.DWG

[L043_2x2]^cINSERT D:/SYMBOLS/L043_2x2.DWG

[L055_2x2]^cINSERT D:/SYMBOLS/L055_2x2.DWG
[L055_3x2]^cINSERT D:/SYMBOLS/L055_3x2.DWG
[L055_4x2]^cINSERT D:/SYMBOLS/L055_4x2.DWG

**FL40 3

[L006_40]^cINSERT D:/SYMBOLS/L006_40.DWG
[L006V_40]^cINSERT D:/SYMBOLS/L006V_40.DWG
[L007_40]^cINSERT D:/SYMBOLS/L007_40.DWG
[L008_40]^cINSERT D:/SYMBOLS/L008_40.DWG
[L009_40]^cINSERT D:/SYMBOLS/L009_40.DWG
[L010_40]^cINSERT D:/SYMBOLS/L010_40.DWG
[L011_40]^cINSERT D:/SYMBOLS/L011_40.DWG
[L012_40]^cINSERT D:/SYMBOLS/L012_40.DWG
[L014_40]^cINSERT D:/SYMBOLS/L014_40.DWG
[L015_40]^cINSERT D:/SYMBOLS/L015_40.DWG
[L016_40]^cINSERT D:/SYMBOLS/L016_40.DWG
[L017_40]^cINSERT D:/SYMBOLS/L017_40.DWG
[L018_40]^cINSERT D:/SYMBOLS/L018_40.DWG
[L019_40]^cINSERT D:/SYMBOLS/L019_40.DWG
[L020_40]^cINSERT D:/SYMBOLS/L020_40.DWG
[L021_40]^cINSERT D:/SYMBOLS/L021_40.DWG
[L025_40]^cINSERT D:/SYMBOLS/L025_40.DWG
[L026_40]^cINSERT D:/SYMBOLS/L026_40.DWG
[L027_40A]^cINSERT D:/SYMBOLS/L027_40A.DWG
[L027_40M]^cINSERT D:/SYMBOLS/L027_40M.DWG
[L028_40A]^cINSERT D:/SYMBOLS/L028_40A.DWG
[L028_40M]^cINSERT D:/SYMBOLS/L028_40M.DWG
[L029_40]^cINSERT D:/SYMBOLS/L029_40.DWG
[L029_40F]^cINSERT D:/SYMBOLS/L029_40F.DWG
[L030_40]^cINSERT D:/SYMBOLS/L030_40.DWG
[L030_40F]^cINSERT D:/SYMBOLS/L030_40F.DWG
[L043_1x4]^cINSERT D:/SYMBOLS/L043_1x4.DWG
[L043_2x4]^cINSERT D:/SYMBOLS/L043_2x4.DWG

[L044_LG]^cINSERT D:/SYMBOLS/L044_LG.DWG

[L044_SM]^cINSERT D:/SYMBOLS/L044_SM.DWG
[L055_1x4]^cINSERT D:/SYMBOLS/L055_1x4.DWG
[L055_2x4]^cINSERT D:/SYMBOLS/L055_2x4.DWG
[L055_3x4]^cINSERT D:/SYMBOLS/L055_3x4.DWG
[L055_4x4]^cINSERT D:/SYMBOLS/L055_4x4.DWG
[L024]^cINSERT D:/SYMBOLS/L024.DWG
[L039_CO]^cINSERT D:/SYMBOLS/L039_CO.DWG
[L039_CL]^cINSERT D:/SYMBOLS/L039_CL.DWG

****FL32 3**

[L031_8I]^cINSERT D:/SYMBOLS/L031_8I.DWG
[L031_10I]^cINSERT D:/SYMBOLS/L031_10I.DWG
[L031_12I]^cINSERT D:/SYMBOLS/L031_12I.DWG
[L032_6I]^cINSERT D:/SYMBOLS/L032_6I.DWG
[L032_8I]^cINSERT D:/SYMBOLS/L032_8I.DWG
[L032_10I]^cINSERT D:/SYMBOLS/L032_10I.DWG
[L032_12I]^cINSERT D:/SYMBOLS/L032_12I.DWG

****INCAN 3**

[L033_SM]^cINSERT D:/SYMBOLS/L033_SM.DWG
[L033_MD]^cINSERT D:/SYMBOLS/L033_MD.DWG
[L033_LG]^cINSERT D:/SYMBOLS/L033_LG.DWG
[L035_6I]^cINSERT D:/SYMBOLS/L035_6I.DWG
[L035_8I]^cINSERT D:/SYMBOLS/L035_8I.DWG
[L035_10I]^cINSERT D:/SYMBOLS/L035_10I.DWG
[L035_12I]^cINSERT D:/SYMBOLS/L035_12I.DWG
[L036_SM]^cINSERT D:/SYMBOLS/L036_SM.DWG
[L037]^cINSERT D:/SYMBOLS/L037.DWG
[L038_6I]^cINSERT D:/SYMBOLS/L038_6I.DWG
[L038_8I]^cINSERT D:/SYMBOLS/L038_8I.DWG
[L038_10I]^cINSERT D:/SYMBOLS/L038_10I.DWG
[L038_12I]^cINSERT D:/SYMBOLS/L038_12I.DWG
[L038_14I]^cINSERT D:/SYMBOLS/L038_14I.DWG
[L038_16I]^cINSERT D:/SYMBOLS/L038_16I.DWG

****PL 3**

[LSL09]^cINSERT D:/SYMBOLS/LSL09.DWG
[LSL13]^cINSERT D:/SYMBOLS/LSL13.DWG
[LSL18]^cINSERT D:/SYMBOLS/LSL18.DWG
[LSL25]^cINSERT D:/SYMBOLS/LSL25.DWG
[LPLC09]^cINSERT D:/SYMBOLS/LPLC09.DWG
[LPLC11]^cINSERT D:/SYMBOLS/LPLC11.DWG
[LPLC15]^cINSERT D:/SYMBOLS/LPLC15.DWG
[LPLC20]^cINSERT D:/SYMBOLS/LPLC20.DWG

****MERCURY 3**

[L040_60]^cINSERT D:/SYMBOLS/L040_60.DWG
[L040_250]^cINSERT D:/SYMBOLS/L040_250.DWG
[L041_16I]^cINSERT D:/SYMBOLS/L041_16I.DWG
[L041_20I]^cINSERT D:/SYMBOLS/L041_20I.DWG
[L042_SL1]^cINSERT D:/SYMBOLS/L042_SL1.DWG
[L042_SL2]^cINSERT D:/SYMBOLS/L042_SL2.DWG
[L042_SL3]^cINSERT D:/SYMBOLS/L042_SL3.DWG
[LHDK80]^cINSERT D:/SYMBOLS/LHDK80.DWG
[LHDK125]^cINSERT D:/SYMBOLS/LHDK125.DWG
[LHDK250]^cINSERT D:/SYMBOLS/LHDK250.DWG
[LHDK400]^cINSERT D:/SYMBOLS/LHDK400.DWG
[LHDK1000]^cINSERT D:/SYMBOLS/LHDK1000.DWG

****METAL 3**

[LHNF250]^cINSERT D:/SYMBOLS/LHNF250.DWG
[LHNF400]^cINSERT D:/SYMBOLS/LHNF400.DWG
[LHNF1000]^cINSERT D:/SYMBOLS/LHNF1000.DWG
[LHNF2000]^cINSERT D:/SYMBOLS/LHNF2000.DWG

****SODIUM 3**

[LSDK70]^cINSERT D:/SYMBOLS/LSDK70.DWG
[LSDK150]^cINSERT D:/SYMBOLS/LSDK150.DWG

[THW185]^cINSERT D:/SYMBOLS/WTHW185.DWG
[THW240]^cINSERT D:/SYMBOLS/WTHW240.DWG
[THW300]^cINSERT D:/SYMBOLS/WTHW300.DWG
[THW400]^cINSERT D:/SYMBOLS/WTHW400.DWG
[THW500]^cINSERT D:/SYMBOLS/WTHW500.DWG

****NYY1C 3**

[NYY1x1.5]^cINSERT D:/SYMBOLS/W1NYY1_5.DWG
[NYY1x2.5]^cINSERT D:/SYMBOLS/W1NYY2_5.DWG
[NYY1x4]^cINSERT D:/SYMBOLS/W1NYY4.DWG
[NYY1x6]^cINSERT D:/SYMBOLS/W1NYY6.DWG
[NYY1x10]^cINSERT D:/SYMBOLS/W1NYY10.DWG
[NYY1x16]^cINSERT D:/SYMBOLS/W1NYY16.DWG
[NYY1x25]^cINSERT D:/SYMBOLS/W1NYY25.DWG
[NYY1x35]^cINSERT D:/SYMBOLS/W1NYY35.DWG
[NYY1x50]^cINSERT D:/SYMBOLS/W1NYY50.DWG
[NYY1x70]^cINSERT D:/SYMBOLS/W1NYY70.DWG
[NYY1x95]^cINSERT D:/SYMBOLS/W1NYY95.DWG
[NYY1x120]^cINSERT D:/SYMBOLS/W1NYY120.DWG
[NYY1x150]^cINSERT D:/SYMBOLS/W1NYY150.DWG
[NYY1x185]^cINSERT D:/SYMBOLS/W1NYY185.DWG
[NYY1x240]^cINSERT D:/SYMBOLS/W1NYY240.DWG
[NYY1x300]^cINSERT D:/SYMBOLS/W1NYY300.DWG

****NYY3C 3**

[NYY3x1.5]^cINSERT D:/SYMBOLS/W3NYY1_5.DWG
[NYY3x2.5]^cINSERT D:/SYMBOLS/W3NYY2_5.DWG
[NYY3x4]^cINSERT D:/SYMBOLS/W3NYY4.DWG
[NYY3x6]^cINSERT D:/SYMBOLS/W3NYY6.DWG
[NYY3x10]^cINSERT D:/SYMBOLS/W3NYY10.DWG
[NYY3x16]^cINSERT D:/SYMBOLS/W3NYY16.DWG
[NYY3x25]^cINSERT D:/SYMBOLS/W3NYY25.DWG
[NYY3x35]^cINSERT D:/SYMBOLS/W3NYY35.DWG
[NYY3x50]^cINSERT D:/SYMBOLS/W3NYY50.DWG
[NYY3x70]^cINSERT D:/SYMBOLS/W3NYY70.DWG

[LSDK250]^cINSERT D:/SYMBOLS/LSDK250.DWG

[LSDK400]^cINSERT D:/SYMBOLS/LSDK400.DWG

****SW 3**

[S1]^cINSERT D:/SYMBOLS/S1.DWG

[S2]^cINSERT D:/SYMBOLS/S2.DWG

[S3]^cINSERT D:/SYMBOLS/S3.DWG

[S3w]^cINSERT D:/SYMBOLS/S3w.DWG

****4PLUG 3**

[SRWS]^cINSERT D:/SYMBOLS/SRWS.DWG

[SRWD]^cINSERT D:/SYMBOLS/SRWD.DWG

[SRF]^cINSERT D:/SYMBOLS/SRF.DWG

****5WIRE 3**

[THW]^c\$S=x \$S=THW

[NYY1C]^c\$S=x \$S=NYY1C

[NYY3C]^c\$S=x \$S=NYY3C

[NYY4C]^c\$S=x \$S=NYY4C

[VAF]^c\$S=x \$S=VAF

****THW 3**

[THW1.5]^cINSERT D:/SYMBOLS/WTHW1_5.DWG

[THW2.5]^cINSERT D:/SYMBOLS/WTHW2_5.DWG

[THW4]^cINSERT D:/SYMBOLS/WTHW4.DWG

[THW6]^cINSERT D:/SYMBOLS/WTHW6.DWG

[THW10]^cINSERT D:/SYMBOLS/WTHW10.DWG

[THW16]^cINSERT D:/SYMBOLS/WTHW16.DWG

[THW25]^cINSERT D:/SYMBOLS/WTHW25.DWG

[THW35]^cINSERT D:/SYMBOLS/WTHW35.DWG

[THW50]^cINSERT D:/SYMBOLS/WTHW50.DWG

[THW70]^cINSERT D:/SYMBOLS/WTHW70.DWG

[THW95]^cINSERT D:/SYMBOLS/WTHW95.DWG

[THW120]^cINSERT D:/SYMBOLS/WTHW120.DWG

[THW150]^cINSERT D:/SYMBOLS/WTHW150.DWG

[THW185]^cINSERT D:/SYMBOLS/WTHW185.DWG
[THW240]^cINSERT D:/SYMBOLS/WTHW240.DWG
[THW300]^cINSERT D:/SYMBOLS/WTHW300.DWG
[THW400]^cINSERT D:/SYMBOLS/WTHW400.DWG
[THW500]^cINSERT D:/SYMBOLS/WTHW500.DWG

****NYY1C 3**

[NYY1x1.5]^cINSERT D:/SYMBOLS/W1NYY1_5.DWG
[NYY1x2.5]^cINSERT D:/SYMBOLS/W1NYY2_5.DWG
[NYY1x4]^cINSERT D:/SYMBOLS/W1NYY4.DWG
[NYY1x6]^cINSERT D:/SYMBOLS/W1NYY6.DWG
[NYY1x10]^cINSERT D:/SYMBOLS/W1NYY10.DWG
[NYY1x16]^cINSERT D:/SYMBOLS/W1NYY16.DWG
[NYY1x25]^cINSERT D:/SYMBOLS/W1NYY25.DWG
[NYY1x35]^cINSERT D:/SYMBOLS/W1NYY35.DWG
[NYY1x50]^cINSERT D:/SYMBOLS/W1NYY50.DWG
[NYY1x70]^cINSERT D:/SYMBOLS/W1NYY70.DWG
[NYY1x95]^cINSERT D:/SYMBOLS/W1NYY95.DWG
[NYY1x120]^cINSERT D:/SYMBOLS/W1NYY120.DWG
[NYY1x150]^cINSERT D:/SYMBOLS/W1NYY150.DWG
[NYY1x185]^cINSERT D:/SYMBOLS/W1NYY185.DWG
[NYY1x240]^cINSERT D:/SYMBOLS/W1NYY240.DWG
[NYY1x300]^cINSERT D:/SYMBOLS/W1NYY300.DWG

****NYY3C 3**

[NYY3x1.5]^cINSERT D:/SYMBOLS/W3NYY1_5.DWG
[NYY3x2.5]^cINSERT D:/SYMBOLS/W3NYY2_5.DWG
[NYY3x4]^cINSERT D:/SYMBOLS/W3NYY4.DWG
[NYY3x6]^cINSERT D:/SYMBOLS/W3NYY6.DWG
[NYY3x10]^cINSERT D:/SYMBOLS/W3NYY10.DWG
[NYY3x16]^cINSERT D:/SYMBOLS/W3NYY16.DWG
[NYY3x25]^cINSERT D:/SYMBOLS/W3NYY25.DWG
[NYY3x35]^cINSERT D:/SYMBOLS/W3NYY35.DWG
[NYY3x50]^cINSERT D:/SYMBOLS/W3NYY50.DWG
[NYY3x70]^cINSERT D:/SYMBOLS/W3NYY70.DWG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[NYY3x95]^cINSERT D:/SYMBOLS/W3NYY95.DWG
[NYY3x120]^cINSERT D:/SYMBOLS/W3NYY120.DWG
[NYY3x150]^cINSERT D:/SYMBOLS/W3NYY150.DWG
[NYY3x185]^cINSERT D:/SYMBOLS/W3NYY185.DWG
[NYY3x240]^cINSERT D:/SYMBOLS/W3NYY240.DWG
[NYY3x300]^cINSERT D:/SYMBOLS/W3NYY300.DWG

****NYY4C 3**

[NYY4x1.5]^cINSERT D:/SYMBOLS/W4NYY1_5.DWG
[NYY4x2.5]^cINSERT D:/SYMBOLS/W4NYY2_5.DWG
[NYY4x4]^cINSERT D:/SYMBOLS/W4NYY4.DWG
[NYY4x6]^cINSERT D:/SYMBOLS/W4NYY6.DWG
[NYY4x10]^cINSERT D:/SYMBOLS/W4NYY10.DWG
[NYY4x16]^cINSERT D:/SYMBOLS/W4NYY16.DWG
[NYY4x25]^cINSERT D:/SYMBOLS/W4NYY25.DWG
[NYY4x35]^cINSERT D:/SYMBOLS/W4NYY35.DWG
[NYY4x50]^cINSERT D:/SYMBOLS/W4NYY50.DWG
[NYY4x70]^cINSERT D:/SYMBOLS/W4NYY70.DWG
[NYY4x95]^cINSERT D:/SYMBOLS/W4NYY95.DWG
[NYY4x120]^cINSERT D:/SYMBOLS/W4NYY120.DWG
[NYY4x150]^cINSERT D:/SYMBOLS/W4NYY150.DWG
[NYY4x185]^cINSERT D:/SYMBOLS/W4NYY185.DWG
[NYY4x240]^cINSERT D:/SYMBOLS/W4NYY240.DWG
[NYY4x300]^cINSERT D:/SYMBOLS/W4NYY300.DWG

****VAF 3**

[VAF2x1]^cINSERT D:/SYMBOLS/W2VAF1.DWG
[VAF2x1.5]^cINSERT D:/SYMBOLS/W2VAF1_5.DWG
[VAF2x2.5]^cINSERT D:/SYMBOLS/W2VAF2_5.DWG
[VAF2x4]^cINSERT D:/SYMBOLS/W2VAF4.DWG
[VAF2x6]^cINSERT D:/SYMBOLS/W2VAF6.DWG
[VAF2x10]^cINSERT D:/SYMBOLS/W2VAF10.DWG
[VAF2x16]^cINSERT D:/SYMBOLS/W2VAF16.DWG

****6CONDUIT 3**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[EMT]^c\$S=x \$S=EMT

[IMC]^c\$S=x \$S=IMC

[RSC]^c\$S=x \$S=RSC

****EMT 3**

[EMT1/2]^cINSERT D:/SYMBOLS/CEM1_2.DWG

[EMT3/4]^cINSERT D:/SYMBOLS/CEM3_4.DWG

[EMT1]^cINSERT D:/SYMBOLS/CEM1.DWG

[EMT1 1/4]^cINSERT D:/SYMBOLS/CEM1_1_4.DWG

[EMT1 1/2]^cINSERT D:/SYMBOLS/CEM1_1_2.DWG

[EMT2]^cINSERT D:/SYMBOLS/CEM2.DWG

****IMC 3**

[IMC2]^cINSERT D:/SYMBOLS/CIM2.DWG

[IMC2 1/2]^cINSERT D:/SYMBOLS/CIM2_1_2.DWG

[IMC3]^cINSERT D:/SYMBOLS/CIM3.DWG

[IMC3 1/2]^cINSERT D:/SYMBOLS/CIM3_1_2.DWG

[IMC4]^cINSERT D:/SYMBOLS/CIM4.DWG

****RSC 3**

[RSC2]^cINSERT D:/SYMBOLS/CRS2.DWG

[RSC2 1/2]^cINSERT D:/SYMBOLS/CRS2_1_2.DWG

[RSC3]^cINSERT D:/SYMBOLS/CRS3.DWG

[RSC3 1/2]^cINSERT D:/SYMBOLS/CRS3_1_2.DWG

[RSC4]^cINSERT D:/SYMBOLS/CRS4.DWG

*+++++

+ โปรแกรมการออกแบบระบบแสงสว่างภายใน และภายนอกอาคาร +

*+++++

set talk off

set color to w+/n,w+/n

private circle,_circle

public number,distance

store .t. to circle

do while circle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clear
_circle=space(1)
@ 0,0 fill to 0,79 color /b
@ 1,0 fill to 1,79 color /g
@ 2,0 to 23,79
@ 4,3 to 6,77
@ 7,3 to 9,77
@ 10,3 to 12,77
@ 15,3 to 17,77
@ 18,3 to 20,77
@ 2,0 fill to 24,79 color w/n
@ 0,26 say 'โปรแกรมออกแบบระบบแสงสว่าง' color gr+/b
@ 1,22 say 'ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์' color n/g
@ 3,35 say 'ข้อมูลที่ต้องการ' color g/n
@ 14,36 say 'ข้อมูลที่ได้' color gr+/n
@ 5,5 say 'ชนิดของพื้นที่ที่ต้องการออกแบบ...'color w+/n
@ 8,5 say 'ขนาดพื้นที่ [กว้าง x ยาว x สูง]'color w+/n
@ 11,5 say 'ชนิด และขนาดของหลอดไฟ ...' color w+/n
@ 16,5 say 'จำนวนดวงโคมที่ต้องใช้' color w+/n
@ 19,5 say 'ระยะห่างระหว่างดวงโคม' color w+/n
save screen to fox
do classroom
restore screen from fox
@ 5,42 say nameroom color g/n
release fox *clear ตัวแปร fox*

save screen to fox
do sizeroom
restore screen from fox
clear windows
@ 8,40 say width+' x '+ltrim(length)+' x '+ltrim(height)+' เมตร' color g/n
release fox

save screen to fox
do typelamp

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clear windows all

@ 11,40 say name:lamp+' '+size:watt color g/n

do calculate

@ 16,28 say str(number,5,1)+' ดวงโคม' color gr+/n

@ 19,28 say str(distance,3,2)+' เมตร' color gr+/n

***ตรวจสอบการทำงานต่อ***

set color to w+/b,w+/n,n

@ 22,3 fill to 22,76 color w+/b

@ 22,24 say 'ต้องการทำงานต่อหรือไม่...(Y/N) :' get _circle picture 'I';

valid _circle$'YyNy' ERROR ' ป้อนค่าผิด' color gr+/b,w+/b

read

if upper(_circle) = 'Y'

    circle = t.

else

    circle = .f.

endif

enddo

@ 23,0 say "

clear memory

release all

release windows

set color to w+/n,w+/n,n

wait "

clear

```

```

procedure calculate

```

```

*+-----+
*+ โปรแกรมคำนวณหาจำนวนดวงโคม และระยะห่างระหว่างดวงโคม +*
*+-----+

```

```

set talk off

```

```

public distance,number,lux,lumen,wid,leg

```

```

private cu.mantanane,num

```

```

cu=0.65

```

```

mantanane=0.6   && เป็นค่า LDD,LLD =>LDD=0.8 LLD=0.75

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

num=(lux*wid*len)/(cu*mantanane*lumen)
number=int(num+0.9)
distance=sqrt(wid*len/number)
set color to w+/n,w+/n,n
return

```

```

procedure classroom

```

```

*+++++*
*++ โปรแกรมย่อย เลือกชนิดพื้นที่ ++*
*+++++*

```

```

set talk off

```

```

set color to w+/b,gr+/r,rb

```

```

private mexit,choice

```

```

public nameroom,lux

```

```

store .f. to mexit

```

```

define window classroom from 7,9 to 14.40 panel shadow color w+/b,gr+/r,r+

```

```

do while .not. mexit

```

```

    activate window classroom

```

```

    @ 0,0 prompt ' 1. พื้นที่อาคารทั่วไป ...

```

```

    @ 1,0 prompt ' 2. สำนักงาน ...

```

```

    @ 2,0 prompt ' 3. สถานศึกษา ...

```

```

    @ 3,0 prompt ' 4. โรงพยาบาล ...

```

```

    @ 4,0 prompt ' 5. ที่อยู่อาศัย, โรงแรม ...

```

```

    @ 5,0 prompt ' 6. สถานกีฬา ...

```

```

    menu to choice

```

```

do case

```

```

    case choice = 1

```

```

        do circulation

```

```

    case choice = 2

```

```

        do offices

```

```

    case choice = 3

```

```

        do education

```

```

    case choice = 4

```

```

        do hospitals

```

```

    case choice = 5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

do homehotel
case choice = 6
do sports
endcase
deactivate window classroom
store .t. to mexit
enddo
clear windows all
@ 5,38 to 5,76 clear
@ 5,38 say nameroom color w+/b
@ 5,4 fill to 5,76 color w/b
set color to w+/n,w+/n,n
return

procedure circulation
*+++++*
*++ ไปแถมย่อย พื้นที่อาคารทั่วไป ++*
*+++++*
set talk off
set color to w+/b,gr+/r,rb
private mexit,choice
public nameroom,lux
store .f. to mexit
define window circulation from 7.43 to 18.69 panel color w+/b,gr+/r,r+
do while .not. mexit
activate window circulation
@ 0,0 prompt ' 1. บริเวณทั่วไป '
@ 1,0 prompt ' 2. ห้องรับรอง '
@ 2,0 prompt ' 3. ร้านอาหาร '
@ 3,0 prompt ' 4. ประชาสัมพันธ์ '
@ 4,0 prompt ' 5. ลิฟท์ '
@ 5,0 prompt ' 6. บันได, บันไดเลื่อน '
@ 6,0 prompt ' 7. ห้องเก็บของ '
@ 7,0 prompt ' 8. พื้นที่รอบๆ ทางเดิน '
@ 8,0 prompt ' 9. ทางออก '

```

```

@ 9,0 prompt '10. ห้องน้ำ
menu to choice
deactivate window circulation
do case
    case choice = 1
        lux=500
        nameroom= "บริเวณทั่วไป"
    case choice = 2
        lux=150
        nameroom= "ห้องรับรอง"
    case choice = 3
        lux=150
        nameroom= "ร้านอาหาร"
    case choice = 4
        lux=500
        nameroom= "ประชาสัมพันธ์"
    case choice = 5
        lux=150
        nameroom= "ลิฟท์"
    case choice = 6
        lux=150
        nameroom= "บันได, บันไดเลื่อน"
    case choice = 7
        lux=150
        nameroom= "ห้องเก็บของ"
    case choice = 8
        lux=100
        nameroom= "พื้นที่รอบๆ ทางเดิน"
    case choice = 9
        lux=50
        nameroom= "ทางออก"
    case choice = 10
        lux=150
        nameroom= "ห้องน้ำ"
endcase

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

store .f. to mexit
enddo
set color to w+/n,w+/n,n
return

procedure offices
*++++*
*++ โปรแกรมย่อย สำนักงาน ++*
*++++*
set talk off
set color to w+/b,gr+/r,rb
private mexit,choice
public nameroom,lux
store .f. to mexit
define window offices from 7,43 to 17,73 panel color w+/b,gr+/r,r+
do while .not. mexit
activate window offices
@ 0,0 prompt ' 1. บริเวณที่ทำงานทั่วไป
@ 1,0 prompt ' 2. ที่ทำงานส่วนตัว
@ 2,0 prompt ' 3. ห้องคอมพิวเตอร์
@ 3,0 prompt ' 4. ห้องทำงานเกี่ยวกับบัญชี
@ 4,0 prompt ' 5. ห้องเขียนแบบ, ออกแบบ
@ 5,0 prompt ' 6. ห้องประชุม
@ 6,0 prompt ' 7. ที่พักรับรอง
@ 7,0 prompt ' 8. ห้องพิมพ์เอกสาร
@ 8,0 prompt ' 9. ครัวรูมรยยนต์
menu to choice
deactivate window offices
do case
case choice = 1
lux=500
nameroom= "บริเวณที่ทำงานทั่วไป"
case choice = 2
lux=750
nameroom= "ที่ทำงานส่วนตัว"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case choice = 3
    lux=500
    nameroom= "ห้องคอมพิวเตอร์"
case choice = 4
    lux=750
    nameroom= "ห้องทำงานเกี่ยวกับบัญชี"
case choice = 5
    lux=750
    nameroom= "ห้องเขียนแบบ, ออกแบบ"
case choice = 6
    lux=750
    nameroom= "ห้องประชุม"
case choice = 7
    lux=300
    nameroom= "ที่พักรับรอง"
case choice = 8
    lux=300
    nameroom= "ห้องพิมพ์เอกสาร"
case choice = 9
    lux=500
    nameroom= "โถ้วรุมรถยนต์"
endcase
store .t. to mexit
enddo
set color to w+/n,w+/n,n
return

procedure education
*+++++*
*+- โปรแกรมย่อย สถานศึกษา +-*
*+++++*
set talk off
set color to, w+/b,gr+/r,rb
private mexit,choice
public nameroom,lux

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

stores .f. to mexit

define window education from 7,43 to 17,74 panel color w+/b.gr+/r,r+

do while .not. mexit

activate window education

@ 0,0 prompt ' 1. ห้องเรียน,ห้องบรรยาย

@ 1,0 prompt ' 2. ห้องสมุด,ห้องอ่านหนังสือ

@ 2,0 prompt ' 3. ห้องเก็บหนังสืออ้างอิง

@ 3,0 prompt ' 4. โต๊ะอ่านหนังสือ

@ 4,0 prompt ' 5. ชั้นวางหนังสือ

@ 5,0 prompt ' 6. บริเวณที่เก็บหนังสือ

@ 6,0 prompt ' 7. กระดานดำ

@ 7,0 prompt ' 8. ห้องปฏิบัติการ

@ 8,0 prompt ' 9. ห้องศิลปะ

menu to choice

deactivate window education

do case

case choice = 1

lux=300

nameroom= "ห้องเรียน, ห้องบรรยาย"

case choice = 2

lux=500

nameroom= "ห้องสมุด, ห้องอ่านหนังสือ"

case choice = 3

lux=500

nameroom= "ห้องเก็บหนังสืออ้างอิง"

case choice = 4

lux=300

nameroom= "โต๊ะอ่านหนังสือ"

case choice = 5

lux=150

nameroom= "ชั้นวางหนังสือ"

case choice = 6

lux=500

nameroom= "บริเวณที่เก็บหนังสือ"

case choice = 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lux=500
nameroom= "กระดานดำ"
case choice = 8
lux=500
nameroom="ห้องปฏิบัติการ"
case choice = 9
lux=500
nameroom="ห้องศิลปะ"
endcase
store .t. to mexit
enddo

set color to w+/n,w+/n,n
return

procedure homehotel
*+++++*
*++ โปรแกรมย่อย ที่อยู่อาศัยและโรงแรม ++*
*+++++*
set talk off
set color to w+/b,gr+/r,rb
private mexit,choice
public nameroom,lux
store .f. to mexit
define window hospital from 7.43 to 18.73 panel color w+/b,gr+/r,r+
do while .not. mexit
activate window hospital
@ 0,0 prompt ' 1. ห้องนอน : ทัวไป
@ 1,0 prompt ' 2. : ไฟหัวเตียง
@ 2,0 prompt ' 3. ห้องน้ำ : ทัวไป
@ 3,0 prompt ' 4. ห้องรับแขก : ทัวไป
@ 4,0 prompt ' 5. ครัว : ทัวไป
@ 5,0 prompt ' 6. : พื้นที่ทำงาน
@ 6,0 prompt ' 7 บันได
@ 7,0 prompt ' 8. ห้องทำงาน
@ 8,0 prompt ' 9. ห้องพักผ่อน

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

@ 9,0 prompt '10. ห้องอาหาร

menu to choice

deactivate window hospital

do case

case choice = 1

lux=50

nameroom= "ห้องนอน : ทั่วไป"

case choice = 2

lux=200

nameroom= "ห้องนอน : ไฟฟ้าเดียว"

case choice = 3

lux=100

nameroom= "ห้องน้ำ : ทั่วไป"

case choice = 4

lux=300

nameroom= "ห้องรับแขก : ทั่วไป"

case choice = 5

lux=300

nameroom= "ครัว : ทั่วไป"

case choice = 6

lux=500

nameroom= "ครัว : พื้นที่ทำงาน"

case choice = 7

lux=100

nameroom= "บันได"

case choice = 8

lux=300

nameroom= "ห้องทำงาน"

case choice = 9

lux=300

nameroom= "ห้องพักรับแขก"

case choice = 10

lux=500

nameroom= "ห้องอาหาร"

endcase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

store .f. to mexit
enddo
set color to w+/n,w+/n,n
return

```

procedure hospitals

```

*++++*
** โปรแกรมย่อย โรงพยาบาล **
*++++*

set talk off

set color to w+/b,gr+/r,rb
private mexit,choice
public nameroom,lux
store .f. to mexit
define window homeshotel from 7,43 to 19,74 panel color w+/b,gr+/r,r+
do while .not. mexit
    activate window homeshotel
    @ 0,0 prompt '1. แสงสว่าง : ทัวไป'
    @ 1,0 prompt '2. ห้องตรวจโรค : ทัวไป'
    @ 2,0 prompt '3. : เฉพาะที่'
    @ 3,0 prompt '4. ห้องผ่าตัด : ทัวไป'
    @ 4,0 prompt '5. : เฉพาะที่'
    @ 5,0 prompt '6. ห้องชันสูตรศพ'
    @ 6,0 prompt '7. ห้องปฏิบัติการ และจ่ายยา'
    @ 7,0 prompt '8. ห้องพักที่ปรึกษาแพทย์'
    @ 8,0 prompt '9. ห้องสอบถาม'
    @ 9,0 prompt '10. ห้องพักพยาบาล'
    @ 10,0 prompt '11. ที่อ่านหนังสือ'

    menu to choice

    deactivate window homeshotel

do case

    case choice = 1

        lux=100

        nameroom= "แสงสว่างทัวไป"

        case choice = 2

```

```

lux=500
nameroom= "ห้องตรวจโรค : แสงสว่างทั่วไป"
case choice = 3
lux=1000
nameroom= "ห้องตรวจโรค : เฉพาะที่"
case choice = 4
lux=750
nameroom= "ห้องผ่าตัด : แสงสว่างทั่วไป"
case choice = 5
lux=30000
nameroom= "ห้องผ่าตัด : แสงสว่างเฉพาะแห่ง"
case choice = 6
lux=750
nameroom= "ห้องชันสูตรศพ"
case choice = 7
lux=500
nameroom= "ห้องปฏิบัติการ และห้องฉายยา"
case choice = 8
lux=500
nameroom= "ห้องห้กที่ปรึกษาแพทย์"
case choice = 9
lux=300
nameroom= "ห้องสอบถาม"
case choice = 10
lux=300
nameroom= "ห้องพักพยาบาล"
case choice = 11
lux=200
nameroom= "ที่อ่านหนังสือ"
endcase
store .t. to mexit
enddo
set color to w+/n,w+/n,n
return

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure sports
*+++++*
*++ โปรแกรมย่อย สถานกีฬา ++*
*+++++*

set talk off

set color to w+/b,gr+/r,rb

private mexit,choice

public nameroom,lux

store .f. to mexit

define window sports from 7,43 to 18,71 panel color w+/b,gr+/r,+

do while .not. mexit

    activate window sports

    @ 0,0 prompt ' 1. สนามกีฬาแบดมินตัน
    @ 1,0 prompt ' 2. สถานกีฬาสนุกเกอร์
    @ 2,0 prompt ' 3. สนามกีฬาฝึกซ้อมมวย
    @ 3,0 prompt ' 4. สนามกีฬาตีกอล์ฟน้ำแข็ง
    @ 4,0 prompt ' 5. สนามกีฬาขี่ม้า
    @ 5,0 prompt ' 6. สนามกีฬาสควอช
    @ 6,0 prompt ' 7. สนามกีฬาสระว่ายน้ำ
    @ 7,0 prompt ' 8. สนามกีฬาเทเบิลเทนนิส
    @ 8,0 prompt ' 9. สนามกีฬาเทนนิส
    @ 9,0 prompt '10. แสงสว่างทั่วไป

    menu to choice

    deactivate window sports

do case

    case choice = 1
        lux=300
        nameroom= "สนามกีฬาแบดมินตัน"

    case choice = 2
        lux=100
        nameroom= "สถานกีฬาสนุกเกอร์"

    case choice = 3
        lux=500
        nameroom= "สนามกีฬาฝึกซ้อมมวย"

    case choice = 4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lux=500
nameroom= "สนามกีฬาสดน้ำแข็ง"
case choice = 5
lux=1000
nameroom= "สนามกีฬาฮังปิ่น"
case choice = 6
lux=500
nameroom= "สนามกีฬาสควอร์ท"
case choice = 7
lux=500
: nameroom= "สนามกีฬาสะพานน้ำ"
case choice = 8
lux=500
nameroom= "สนามกีฬาเทเบิลเทนนิส"
case choice = 9
lux=500
nameroom= "สนามกีฬาเทนนิส"
case choice = 10
lux=150
nameroom= "แสงสว่างทั่วไป"
endcase
store .t. to mexit
enddo
set color to w+/n,w+/n,n
return
i
procedure sizeroom
set talk off
public n,width,length,height,line,wid,leng,heig
store 5 to n
define window sizeroom from 10,15 to 20,65 panel shadow color w/b,g/r,r+
activate window sizeroom

text
พื้นที่ที่ออกแบบระบบแสงสว่าง ( เมตร )

```



- กว้าง
- ยาว
- สูง

```
endtext
@ 8,0 fill to 8,48 color /g
@ 8,10 say 'F9          F10    'color n/g
@ 8,13 say 'ป้อนค่า'color b/g
@ 8,32 say 'เลือกค่า'color b/g
```

```
key=0
do while (key <> -8 .or. -9)
  @ 2,36 say "
  store inkey(120) to key
  if key = -8 .or. key = -9
    exit
  endif
enddo
if key = -8
  do F9
endif
if key = -9
  do F10
endif
```

```
procedure F9
set talk off
public width,length,height
store 0 to wid
store 0 to leng
store 0 to heig
@ 2,34 say " get wid picture "####.##" valid wid > 0 ERROR ' ป้อนค่าผิด' color g/b,g/b
@ 3,34 say " get leng picture "####.##" valid leng > 0 ERROR ' ป้อนค่าผิด' color g/b,g/b
@ 4,34 say " get heig picture "####.##" valid heig > 0 ERROR ' ป้อนค่าผิด' color b/g,g/b
read
```

```
width=str(wid,5,1)
```

```
length=str(leng,5,1)
```

```
height=str(heig,5,1)
```

```
return
```

```
procedure F10
```

```
line=1
```

```
do inkey
```

```
wid=n
```

```
width=str(n,5,1)
```

```
line=2
```

```
do inkey
```

```
leng=n
```

```
length=str(n,5,1)
```

```
line=3
```

```
do inkey
```

```
heig=n
```

```
height=str(n,5,1)
```

```
return
```

```
deactivate window sizeroom
```

```
@ 23,5 say "
```

```
wait
```

```
return
```

```
procedure inkey
```

```
set talk off
```

```
private mkey
```

```
public n
```

```
mkey=0
```

```
do while mkey<>13
```

```
  @%1+line,35 say str(n,5,1)
```

```
  store inkey(120) to mkey
```

```
  if mkey=5
```



```

    n=n+1
  }
endif

if mkey=24
  n=n-1
  if n<1
    n=1
  endif
endif

endif

if mkey=19
  n=n-0.1
  if n<1
    n=1
  endif
endif

if mkey=4
  n=n+0.1
endif

enddo
return

procedure typelamp
*+++++*
*++ โปรแกรมเลือกชนิดและขนาดของหลอดไฟ ++*
*+++++*

set talk off
set color to w+/b,n/g,rb

public lumen, namelamp ,sizewatt

define window mainlamp from 13,13 to 19,40 title ' <<ชนิดหลอด>> ' panel shadow color w+/b,n/g,r
store 0 to lumen
store .f. to lexit
do while .not. lexit
  activate window mainlamp
  @,0,0 prompt ' หลอดอินแคนเดสเซนต์ ... '
  @ 1,0 prompt ' หลอดฟลูออเรสเซนต์ ... '
  @,2,0 prompt ' หลอดประหยัดไฟคอมแพ็ค ...'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 3,0 prompt ' หลอด HID ...
@ 4,0 prompt ' ออกจากโปรแกรม
menu to choice
do case
    case choice = 1
        do incandescent
    case choice = 2
        do fluorescent
    case choice = 3
        do compact
    case choice = 4
        do hid
    case choice = 5
        return
    store t. to lexit
loop
endcase
deactivate window mainlamp
endifdo
clear windows all
set color to w+/b
return

procedure incandescent
*+++++*
*++ โปรแกรมย่อยขนาดหลอดอินแคนเดสเซนต์ ++*
*+++++*

set talk off
set color to w+/b,n/g,r/b
private exit,choice
public lumen ,namelamp ,sizewatt
store .f. to exit

define window incan from 15,44 to 21,63 title-' <ขณใด> panel shadow color w+/b,n/g,r
do while .not. exit
    activate window incan

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 0,0 prompt '1. ขนาด 15 w '
@ 1,0 prompt '2. ขนาด 25 w '
@ 2,0 prompt '3. ขนาด 40 w '
@ 3,0 prompt '4. ขนาด 60 w '
@ 4,0 prompt '5. ขนาด 100 w '
menu to choice
deactivate window incan
do case
. case choice = 1
    lumen = 90
    sizewatt = 'ขนาด 15 w'
. case choice = 2
    lumen = 230
    sizewatt = 'ขนาด 25 w'
. case choice = 3
    lumen = 430
    sizewatt = 'ขนาด 40 w'
. case choice = 4
    lumen = 730
    sizewatt = 'ขนาด 60 w'
. case choice = 5
    lumen = 1380
    sizewatt = 'ขนาด 100 w'
endcase
nameilamp = 'หลอดอินแคนเดสเซนต์'
store .t. to exit
enddo
set color to w+/b
return

```

```

procedure fluorescent

```

```

*+++++*
*++ โปรแกรมย่อยขนาดหลอดฟลูออเรสเซนต์ ++*
*+++++*

```

```

set talk off

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set color to w+/b,n/g,r
private exit,choice
public lumen ,namelamp ,sizewatt
store, .f. to exit
define window fluor from 13,44 to 23,66 title ' <ขนาด> ' panel shadow color w+/b,n/g,r
do while .not. exit
    activate window fluor
    @ 0,0 prompt ' 1. ขนาด 1 x 20 w '
    @ 1,0 prompt ' 2. ขนาด 2 x 20 w '
    @ 2,0 prompt ' 3. ขนาด 3 x 20 w '
    @ 3,0 prompt ' 4. ขนาด 4 x 20 w '
    @ 4,0 prompt ' 5. ขนาด 1 x 40 w '
    @ 5,0 prompt ' 6. ขนาด 2 x 40 w '
    @ 6,0 prompt ' 7. ขนาด 3 x 40 w '
    @ 7,0 prompt ' 8. ขนาด 4 x 40 w '
    @ 8,0 prompt ' 9. ขนาด 32 w '
menu to choice
deactivate window fluor
do case.
    case choice.= 1
        lumen=1*1070
        sizewatt = 'ขนาด 1 x 20 w'
    case choice = 2
        lumen=2*1070
    case choice = 3
        lumen=3*1070
        sizewatt = 'ขนาด 3 x 20 w'
    case choice = 4
        lumen=4*1070
        sizewatt = 'ขนาด 4 x 20 w'
    case choice = 5
        lumen=1*2700
        sizewatt = 'ขนาด 1 x 40 w'
    case choice = 6
        lumen=2*2700

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sizewatt = 'ขนาด 2 x 40 w'
    case choice = 7
        lumen=3*2700
        sizewatt = 'ขนาด 3 x 40 w'
    case choice = 8
        lumen=4*2700
        sizewatt = 'ขนาด 4 x 40 w'
    case choice = 9
        lumen=1*1890
        sizewatt = 'ขนาด 32 w'
    endcase
    namelamp='หลอดฟลูออโรเรสเซนต์'
    store .f. to exit
enddo
set color to w+/b
return

procedure compact
*+++++*
*+  โปรแกรมย่อยหลอดประหยัดไฟคอมแพ็ค  ++*
*+++++*
set talk off
set color to w+/b,n/g,rb
private exit,choice
public lumen ,namelamp ,sizewatt
store .f. to exit

define window compact from 13,44 to 22,68 title '<ขนาด>' panel shadow color w+/b,n/g,r
do while .not. exit
    activate window compact
    @ 0,0 prompt '1. หลอด SL - 9 w '
    @ 1,0 prompt '2.      - 11 w '
    @ 2,0 prompt '3.      - 15 w '
    @ 3,0 prompt '4.      - 20 w '
    @ 4,0 prompt '5 หลอด PLC - 9 w '
    @ 5,0 prompt '6.      - 11 w

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

๒ 6,0 prompt ' 7.      - 15  w '
* @ 7,0 prompt ' 8.      - 20  w '

menu to choice

deactivate window compact

do case

    case choice = 1

        lumen=450

        sizewatt = 'SL ขนาด 9 w'

    case choice = 2

        lumen=600

        sizewatt = 'SL ขนาด 11 w'

    case choice = 3

        lumen=900

        sizewatt = 'SL ขนาด 15 w'

    case choice = 4

        lumen=1200

        sizewatt = 'SL ขนาด 20 w'

    case choice = 5

        lumen=400

        sizewatt = 'PLC ขนาด 9 w'

    case choice = 6

        lumen=600

        sizewatt = 'PLC ขนาด 11 w'

    case choice = 7

        lumen=900

        sizewatt = 'PLC ขนาด 15 w'

    case choice = 8

        lumen=1200

        sizewatt = 'PLC ขนาด 20 w'

endcase

nameLamp='หลอดประหยัดไฟคอมแพ็ค'

store .t. to exit

enddo

set color to w+/b

return

```

```

procedure hid
*+++++*
*++ โปรแกรมย่อยขนาดหลอด HID ++*
*+++++*

set,talk off
set color to w+/b,n/g,r
private exit,choice
public lumen ,namelamp

define window hid from 17,42 to 21,72 panel title '<<ชนิดหลอด>>' shadow color w+/b,n/g,r
store .f. to exit

do while .not. exit
    activate window hid
    @ 0,0 prompt '1. หลอดแสงจันทร์ ...'
    @ 1,0 prompt '2. หลอดเมทอล-ฮาไลด์ ...'
    @ 2,0 prompt '3. หลอดโซเดียมความดันสูง ...'

    menu to choice

    do case
        case choice = 1
            do mercury
        case choice = 2
            do metal_halide
        case choice = 3
            do sodium

    endcase

    store .t. to exit

deactivate window hid

enddo

set color to w+/b
return

procedure mercury
*+++++*
*++ โปรแกรมย่อยขนาดหลอดแสงจันทร์ ++*
*+++++*

```

```

set talk off

set color to w+/b,n/g,r,b

private exit,choice

public lumen ,namelamp ,sizewatt

store .f. to exit

define window mercury from 9,52 to 15,72 title ' <ขนาด> ' panel shadow color w+/b,n/g,r

do while .not. exit

    activate window mercury

    @ 0,0 prompt ' 1. ขนาด  80  w '

    @ 1,0 prompt ' 2. ขนาด  125  w '

    @ 2,0 prompt ' 3. ขนาด  250  w '

    @ 3,0 prompt ' 4. ขนาด  400  w '

    @ 4,0 prompt ' 5. ขนาด 1000  w '

    menu to choice

    deactivate window mercury

do case

    case choice = 1

        lumen = 3300

        sizewatt = 'ขนาด 80 w'

    case choice = 2

        lumen = 5900

        sizewatt = 'ขนาด 125 w'

    case choice = 3

        lumen = 12700

        sizewatt = 'ขนาด 250 w'

    case choice = 4

        lumen = 22000

        sizewatt = 'ขนาด 400 w'

    case choice = 5

        lumen = 59500

        sizewatt = 'ขนาด 1000 w'

    endcase

    namelamp='หลอดแสงจันทร์'

    store .t. to exit

enddo

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
set color to w+/b
```

```
return
```

```
procedure metal_halide
```

```
*+++++
```

```
*++ โปรแกรมข่อยขนาดหลอดเมทอล-ฮาไลด์ ++*
```

```
*+++++
```

```
set talk off
```

```
set color to w+/b,n/g,r,b
```

```
private exit,choice
```

```
public lumen ,namelamp ,sizewatt
```

```
store .f. to exit
```

```
define window metal from 10,52 to 15,72 title '<ขนาด>' panel shadow color w+/b,n/g,r
```

```
do while .not. exit
```

```
activate window metal
```

```
@ 0,0 prompt '1. ขนาด 250 w '
```

```
@ 1,0 prompt '2. ขนาด 400 w '
```

```
@ 2,0 prompt '3. ขนาด 1000 w '
```

```
@ 3,0 prompt '4. ขนาด 2000 w '
```

```
menu to choice
```

```
deactivate window metal
```

```
do case
```

```
case choice = 1
```

```
lumen = 21500
```

```
sizewatt = 'ขนาด 250 w'
```

```
* case choice = 2
```

```
lumen = 40000
```

```
* sizewatt = 'ขนาด 400 w'
```

```
case choice = 3
```

```
lumen = 83000
```

```
sizewatt = 'ขนาด 1000 w'
```

```
case choice = 4
```

```
lumen = 170000
```

```
sizewatt = 'ขนาด 2000'
```

```
endcase
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    namelamp='หลอดเมทอล-ฮาโลอิด'
    store .t. to exit
enddo

set color to w+/b

return

procedure sodium
*+++++*
*+- โปรแกรมย่อขนาดหลอดโซเดียมความดันสูง +-*
*+++++*

set talk off

set color to w+/b.n/g.r/b

private exit,choice
public lumen ,namelamp ,sizewatt

store .f. to exit

define window sodium from 10.53 to 15.72 title '<ขนาด>' panel shadow color w+/b.n/g.r

do while .not. exit

    activate window sodium

    @ 0,0 prompt '1. ขนาด 70 w '
    @ 1,0 prompt '2. ขนาด 150 w '
    @ 2,0 prompt '3. ขนาด 250 w '
    @ 3,0 prompt '4. ขนาด 400 w '

    menu to choice

    deactivate window sodium

do case

    case choice = 1

        lumen = 6000

        sizewatt = 'ขนาด 70 w'

    case choice = 2

        lumen = 14500
        sizewatt = 'ขนาด 150 w'

    case choice = 3

        lumen = 27500
        sizewatt = 'ขนาด 250 w'

    case choice = 4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lumen = 50000

sizewatt = 'ขนาด 400 w'

endcase

namelamp='หลอดโซเดียมความดันสูง'
store .t. to exit

enddo

set color to w+/b

return

*+++++*
* โปรแกรมช่วยการเชื่อมโยงข้อมูล FoxPro กับ Auto Cad *
*+++++*

set talk off
set safety off
set exact off
clear
set color to w+/n,w+/n,n
@ 10,25 say 'โปรแกรมตัวนี้ กำลังเชื่อมโยงข้อมูลอยู่' color gr+/b
close all
use d:\project\filedata\con1
zap
append from d:\WRK\CONDO sdf
close all
select 1
use d:\project\filedata\con1
select 2
use d:\project\filedata\hv
zap      && ลบข้อมูลในแฟ้มเหลือแต่โครงสร้าง
select 3
use d:\project\filedata\pb
zap
select 4
use d:\project\filedata\lm
zap
select 5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

use d:\project\filedata\s_r
zap
select 6
use d:\project\filedata\high volt
zap
select 7
use d:\project\filedata\panel_b
zap
select 8
use d:\project\filedata\luminar
zap
select 9
use d:\project\filedata\sw_recep
zap
select 10
use d:\project\filedata\wire
zap
select 11
use d:\project\filedata\conduit
zap
*****
select 1
endrec = reccount()
'dimension namecode[endrec]
public dimension codeelem[endrec],finalcode
go top
do while .not. EOF()
    namecode[recno()] = code
    skip
enddo
go top
DO DIVIDE_E
*****
select 2
numrec=reccount()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF numrec!=0
DO FC
select 6
DO REP_CODE
select 2
DO COUNTER
select 6
DO REP_QTY
ENDIF

```

```

release all
select 3
numrec=reccount()
IF numrec!=0
DO FC
select 7
DO REP_CODE
select 3
DO COUNTER
select 7
DO REP_QTY
ENDIF

```

```

release all
select 4
numrec=reccount()
IF numrec!=0
DO FC
select 8
DO REP_CODE
select 4
DO COUNTER
select 8
DO REP_QTY
ENDIF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****

release all

select 5

numrec=reccount()

IF numrec!=0

DO FC

select 9

DO REP_CODE

select 5

DO COUNTER

select 9

DO REP_QTY

ENDIF

release all

*****

CLEAR

close all

close databases

@ 10,28 say 'การเชื่อมโยงข้อมูลเสร็จสิ้นแล้ว' color gr+/n

@ 12,27 say 'โปรดคลิกที่ใดๆ เพื่อทำงานต่อไป' color g/n

set color to w+/b,w+/b

WAIT "

DO SCREEN

RELEASE ALL

CLEAR MEMORY

*** จบการทำงานของโปรแกรมแสดงผลทางจอภาพ ***

*****

```

```

PROCEDURE DIVIDE_E

```

```

set safety off

```

```

set exact off

```

```

for i=1 to endrec

```

```

testcode=substr(namecode[i],1,1)

```

```

if upper(testcode) == 'T'

```

```

select 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        append blank
        replace code with namecode[i]
    ,endif
    if upper(testcode) == 'B'
        select 3
        append blank
        replace code with namecode[i]
    ,endif
    if upper(testcode) == 'L'
        select 4
        append blank
        replace code with namecode[i]
    ,endif
    if upper(testcode) == 'S'
        select 5
        append blank
        replace code with namecode[i]
    ,endif
    if upper(testcode) == 'W'
        select 10
        append blank
        replace code with namecode[i]
    ,endif
    if upper(testcode) == 'C'
        select 11
        append blank
        replace code with namecode[i]
    ,endif
endfor
'RETURN

```

PROCEDURE COUNTER

```

set talk off
public dimension ccode[finalcode]
for i=1 to finalcode

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ccode[i]=0
endfor
go top
do while .not. EOF()
  *for i=1 to finalcode
    if code==codeelem[i]
      ccode[i]=ccode[i]+1
    endif
  endfor
  skip
enddo
RETURN

```

PROCEDURE FC

```

set talk off
endrec = recount()
dimension namecode[endrec]
public dimension codeelem[endrec]
public finalcode
go top
store 0 to row
store 1 to finalcode
store 1 to psc
do while .not. EOF()
  namecode[recno()] = code &&ชื่อเขตข้อมูลของโปรแกรมที่linkได้
  pscode = recno()
  psc = pscode - 1
  if psc !=0
    time = 0
    * for i=1 to finalcode
      if namecode[recno()] != codeelem[i]
        time = time+1
      endif
    endfor
    if time = finalcode

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        finalcode = finalcode+1
        codeelem[finalcode] = namecode[recoo()]
    endif
endif
codeelem[1] = namecode[1]
skip
enddo
RETURN

```

```

*
PROCEDURE REP_CODE
*
set talk off
for i=1 to finalcode
    append blank
    replace code with codeelem[i]
endfor
RETURN

```

```

PROCEDURE REP_QTY
public dimension ccode[finalcode]
go top
j=1
do while .not. EOFQ
    replace qty with ccode[j]
    j=j+1
    skip
enddo
RETURN

```

```

PROCEDURE SCREEN

```

```

*+++++*
*++ โปรแกรมการแสดงผลออกทางจอภาพ ++*
*+++++*

set talk off

clear

public name,_overhead,overhead

```

```

public _profit,profit,_vat,vat
store 0 to _overhead
store 0 to overhead
store 0 to _profit
store 0 to profit
store 0 to _vat
store 0 to vat
name=space(35)
@ 9,25 say 'PROJECT NAME .'
@ 10,25 say 'Overhead =  %'
@ 11,25 say 'Profit =  %'
@ 12,25 say 'Vat =  %'
@ 9,40 get name picture 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ 10,37 get _overhead picture '##':
    valid _overhead > 0 error ' ป้อนค่าผิด'
@ 11,37 get _profit picture '##' valid _profit > 0 error ' ป้อนค่าผิด'
@ 12,37 get _vat picture '##' valid _vat > 0 error ' ป้อนค่าผิด'
read
clear
DQ HIGHVOLT
DO PANELBOARD
DO LUMINAIRE
DO SWITCH_RE
DO CABLE
DO RACE_WAY
DO MAIN
RETURN

```

PROCEDURE HIGHVOLT

```

*+++++*
*++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++*
*+++++*

set talk off
set safety off

private endrec,cnt

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public name
? numlock(t.)
clear
close databases
use' filedata
set index to d:\project\filedata\codeind
*index on code to d:\project\filedata\codeind
reindex
close databases
select 1
use d:\project\filedata\highvolt
use filedata in 2 index d:\project\filedata\codeind
set relation to code into filedata

select 1
go top
do while .not. EOF()
    select 1
    *****
    select 2
    replace highvolt->descrip with descrip
    replace highvolt->unit with unit
    replace highvolt->price with price
    select 1
    skip
enddo
close databases
clear
DO PRINT1 &&แสดงผลของ HIGH VOLT
use, d:\project\filedata\file
zap
append blank
replace item with 1
replace descrip with 'HIGH VOLT'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *
        replace unit with 'LOT'
        replace pricemate with totelement
        replace price_lb with totlabour
        replace totprice with totalcost
close all
wait " && จบการแสดงผลของ HIGH VOLT
RETURN

```

PROCEDURE PANELBOARD

```

*+++++*
*++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++*
*+++++*

set talk off
? numlock(t.)
clear
close databases
use filedata
set index to d:\project\filedata\codeind
*index on code to d:\project\filedata\codeind
reindex
close databases
select 1
use d:\project\filedata\panel_b
use filedata in 2 index d:\project\filedata\codeind
set relation to code into filedata

```

```

select 1
go top
do while .not. EOF()
    select 1
    *****
    select 2
    replace panel_b->descrip with descrip
    replace panel_b->unit with unit
    replace panel_b->price with price

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:
:

select l

skip

enddo

close databases

clear

DO>PRINT2 &&แสดงผลของ PANEL BOARD

use'd\project\filedata\file

append blank

replace item with 2

replace descrip with 'PANEL BOARD'

replace qty with 1

replace unit with 'LOT'

replace pricemate with totelement

replace price_lb with totlabour

replace totprice with totalcost

set device to screen

close all

wait " && จบการแสดงผลของ PANEL BOARD

RETURN

PROCEDURE LUMINAIRE

*+++++*

*++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++*

*+++++*

set talk off

? numlock(.t.)

clear

close databases

use filedata

set index to d\project\filedata\codeind

*index on code to d\project\filedata\codeind

reindex

close databases

select l

use d\project\filedata\luminar

```

```
use filedata in 2 index d:\project\filedata\codeind
set relation to code into filedata
```

```
select 1
```

```
go top
```

```
do while .not. EOF0
```

```
select 1
```

```
*****
```

```
select 2
```

```
replace luminair->descrip with descrip
```

```
replace luminair->unit with unit
```

```
replace luminair->price with price
```

```
select 1
```

```
skip
```

```
enddo
```

```
close databases
```

```
clear
```

```
DO PRINT3 &&แสดงผลของ LUMINAIRE
```

```
use d:\project\filedata\file
```

```
append blank
```

```
replace item with 3
```

```
replace descrip with 'LUMINAIRE'
```

```
replace qty with 1
```

```
replace unit with 'LOT'
```

```
replace pricemate with totelement
```

```
replace price_lb with totlabour
```

```
replace totprice with totallcost
```

```
close all
```

```
wait " && จบการแสดงผลของ LUMINAIRE
```

```
RETURN
```

```
PROCEDURE SWITCH_RE
```

```
*+++++
```

```
*++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++*
```

```
*+++++
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set talk off
? numlock(.t.)
clear
*
close databases

use filedata

set index to d:\project\filedata\codeind
*index on code to d:\project\filedata\codeind
reindex

close databases

select 1
use d:\project\filedata\sw_recep
use filedata in 2 index d:\project\filedata\codeind
set relation to code into filedata

select 1
go top
do while .not. EOF()
    select 1
    *****
    select 2
    replace sw_recep->descrip with descrip
    replace sw_recep->unit with unit
    replace sw_recep->price with price
    select 1
    skip
enddo

close databases

clear

DO PRINT4 &&แสดงผลของ SWITCH & RECEPTACLE

use d:\project\filedata\file

    append blank

    replace item with 4

    replace descrip with 'SWITCH & RECEPTACLE'

    replace qty with 1

    replace unit with 'LOT'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

replace pricemate with totalement
replace price_lb with totlabour
replace totprice with totalcost
close all
wait " && จบการแสดงผลของ SWITCH & RECEPTACLE
RETURN

```

PROCEDURE CABLE

```

*+++++*
*++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++*
*+++++*

set talk off
set safety off
set color to w+/n,w+/n
private endrec.cnt
public name
? numlock(.t.)
clear
use d:\project\filedata\wire
wirechk=reccount()
IF WIRECHK!=0
close database
use filedata
set index to d:\project\filedata\codeind
*index on code to d:\project\filedata\codeind
reindex
close databases
select 1
use d:\project\filedata\wire
use filedata in 2 index d:\project\filedata\codeind
set relation to code into filedata

select 1
go top
do while .not. EOF()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select 1
*****

select 2
replace wire->descrip with descrip
replace wire->unit with unit
replace wire->price with price

select 1

skip

enddo

close all

clear

use d:\project\filedata\wire
endrec=reccount()
dimension _qty[endrec]

go top

row=0

cnt=1

,@ 2,11 fill to 2,70 color /b
,@ 4,12 fill to 5+endrec,69 color /b
,@ 2,20 say '### กรุณาป้อนความยาวของสายไฟขนาดต่างๆ ดังนี้ ###' color gr+/b+
,@ 3,11 to 6+endrec,70 color /g
,@ 4,15 say 'Item' color g/b
,@ 4,27 say 'Description' color g/b
,@ 4,57 say 'Qty' color g/b
,@ 4,63 say 'Unit' color g/b

do while .not. EOF()

    @ 5+row,16 say str(cnt,2,0) color w+/b
    , @ 5+row,22say descrip color w+/b
    @ 5+row,64 say unit color w+/b

    row=row+1

    cnt=cnt+1

    skip

enddo

for i=1 to endrec
store 0 to _qty[i]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    @ 4+i,56 get _qty[i] picture '####';
    valid _qty[i] > 0 error ' ป้อนค่าผิด' color /n.gr+/b
  }
  read
endfor
go top
i=1
do while .not. EOF()
  replace qty with _qty[i]
  i=i+1
  skip
enddo
*
DO PRINT5 &&แสดงผลของ CABLE
use d:\project\filedata\file
  append blank
  replace item with 5
  replace descrip with 'CABLE'
  replace qty with 1
  replace unit with 'LOT'
  * replace pricemate with totelement
  * replace price_lb with totlabour
  replace totprice with totalcost
*
close all
wait "
ELSE
use d:\project\filedata\file
* append blank
  replace item with 5
  replace descrip with 'No Wire In Plan'
close database
use d:\project\filedata\wire
  append blank
  replace descrip with 'ไม่มีสายไฟในแบบแปลนไฟฟ้า'
close all
ENDIF
RETURN
i

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCEDURE RACE_WAY

*+*****+

++ โปรแกรมการเชื่อมโยงข้อมูล และแสดงผล ++

*+*****+

set talk off

set safety off

set color to w+/n,w+/n

private endrec, cnt

public name

? numlock(.t.)

clear

use d:\project\filedata\conduit

conduitchk=reccount()

IF CONDUITCHK!=0

close database

use filedata

set index to d:\project\filedata\codeind

*index on code to d:\project\filedata\codeind

reindex

close databases

select 1

use d:\project\filedata\conduit

use filedata in 2 index d:\project\filedata\codeind

set relation to code into filedata

select 1

go top

do while .not. EOF()

select 1

select 2

replace conduit->descrip with descrip

replace conduit->unit with unit

replace conduit->price with price

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select 1
skip
enddo
close all
clear
use d:\project\filedata\conduit
endrec=reccount()
dimension _qty[endrec]
go top
row=0
cnt=1
@ 2,11 fill to 2,70 color /b
@ 4,12 fill to 5+endrec,69 color/b
@ 2,22 say '### กรุณาป้อนความยาวของท่ขนาดต่างๆ ดังนี้ ###' color gr+/b+
@ 3,11 to 6+endrec,70 color /g
@ 4,15 say 'Item' color g/b
@ 4,27 say 'Description' color g/b
@ 4,57 say 'Qty' color g/b
@ 4,63 say 'Unit' color g/b
do while not. EOFQ
    @ 5+row,16 say str(cnt,2,0) color w+/b
    @ 5+row,22 say descrip color w+/b
    @ 5+row,64 say unit color w+/b
    row=row+1
    cnt=cnt+1
skip
enddo
for i=1 to endrec
    store 0 to _qty[i]
    @ 4+i,56 get _qty[i] picture '####';
    valid _qty[i] > 0 error ' ป้อนค่าผิด' color /b.gr+/b
read
endfor
go top
i=1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do while .not. EOFQ
    replace qty with _qty[i]
    i=i+1
    skip
enddo

DO PRINT6 &&แสดงผลของ RACE WAY

use d:\project\filedata\file
    append blank
    replace item with 6
    replace descrip with 'RACE WAY'
    replace qty with 1
    replace unit with 'LOT'
    replace pricemate with totelement
    replace price_lb with totlabour
    replace totprice with totalcost
close all
wait "
ELSE
use d:\project\filedata\file
    append blank
    replace item with 6
    replace descrip with 'No Race Way In Plan'
close database
use 'd:\project\filedata\conduit'
    append blank
    replace descrip with 'ไม่มี Race Way ในแบบแปลนไฟฟ้า'
close all
ENDIF
RETURN

```

PROCEDURE MAIN

```

*+++++*
*++ โปรแกรมการแสดงผลทางจอภาพ ++*
*+++++*

set talk off

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set color to w+/n,w+/n

public name,_overhead,overhead
public _profit,profit,_vat,vat

clear

use d:\project\filedata\file

store 5 to row

store 0 to submate

store 0 to sublabour

store 0 to subtotal

store 0 to total

store 0 to grandtotal

go top

do while .not. EOF()

    submate=submate+pricemate
    sublabour=sublabour+price_lb
    subtotal=subtotal+totprice
    skip
enddo

clear

DO HEADER0

@ 2,0 say "
@ row(),0 say repl('=',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl('=',79) color g/n
overhead=_overhead/100*subtotal
profit=_profit/100*subtotal
vat=INT(_vat/100*(subtotal+overhead+profit))
total=subtotal+overhead+profit
grandtotal=subtotal+overhead+profit+vat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;

go top

replace nameproj with name
do while .not. EOF0
    @ row,1 say str(item,2,0)
    @ row,7 say descrip
    @ row,42 say str(qty,1,0)
    @ row,47 say unit
    @ row,52 say str(pricemate,8,0)
    @ row,62 say str(price_lb,7,0)
    @ row,70 say str(totprice,8,0)
    row=row+1
    skip
enddo

@ 1+row0,0 say repl('-',79) color g/n
@ 1+row0,25 say 'sub-total' color g/n
@ row0,52 say str(submate.8,0) color gr+/n
@ row0,62 say str(sublabour.7,0) color gr+/n
@ row0,69 say str(subtotal.9,0) color gr+/n
@ 1+row0,25 say 'overhead' color g/n
@ row0,68 say str(overhead,10,0) color g/n
@ 1+row0,25 say 'profit' color g/n
@ row0,68 say str(profit,10,0) color g/n
@ 1+row0,25 say 'TOTAL' color g/n
@ row0,68 say str(total,10,0) color gr+/n
@ 1+row0,25 say 'vat' color g/n
@ row0,68 say str(vat,10,0) color g/n
@ 1+row0,68 say repl('-',11)
@ 1+row0,25 say 'GRAND TOTAL' color g/n
@ row0,68 say str(grandtotal,10,0) color gr+/n
@ 1+row0,68 say repl('=',11)
use d:\project\data\file
go bottom

replace submate_ with submate,sublabour_ with sublabour
replace subtotal_ with subtotal,overhead_ with overhead
replace profit_ with profit,total_ with total

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่จำกัดสิทธิ์ในสิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

replace vat_ with vat,finaltotal with grandtotal

close all

wait "

RETURN

PROCEDURE PRINT1

*+++++

+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +

*+++++

set talk off

set century on

set mark to '.'

set date to british

set color to w+/n,w+/n

clear

*+++++

+ แสดงผลการประมาณราคานอกราชการ +

*+++++

set talk off

set color to w+/n

private cnt,value,_value

public totelement,totlabour,totalcost

use d:\project\filedata\highvolt

store 15 to access

store 30 to value

store 0 to _value

store 0 to cnt

store 0 to row

store 0 to _labour

store 0 to subtotal

store 0 to subcost

store 0 to totalcost

store 0 to totlabour

store 0 to subelement

store 0 to totelement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

store 'HIGH VOLT' to nameprj
go top
DO `HEADER1
@ 2,0 say "
@ row(),0 say repl ('=',79)    color g/n
@ row()+1,0 say 'Item'         color g/n
@ row(),10 say 'Description'    color g/n
@ row(),42 say 'Qty'           color g/n
@ row(),47 say 'Unit'          color g/n
@ row(),53 say 'UnitPrice'     color g/n
@ row(),c+4 say 'Labour'       color g/n
@ row(),73 say 'Total'         color g/n
@ row()+1,0 say repl ('=',79)  color g/n
@ 9,23 say 'ค่าติดตั้งอุปกรณ์ [%] =  %'
@ 10,23 say 'ค่าอุปกรณ์เพิ่มเติม [%] =  %'
@ 9,47 get value picture '99' valid value > 0 error ' บ่อนค่าผิด'
@ 10,49 get access picture '99' valid value > 0 error ' บ่อนค่าผิด'
read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()
    cnt = cnt + 1
    _labour=INT(_value*price*qty)
    subtotal=qty*price+_labour
    @ 5+row,0 say "
    @ row().1 say str(cnt,2,0)
    @ row(),5 say descrip
    @ row(),40 say str(qty,5,0)
    @ row(),47 say unit
    @ row(),52 say str(price,7,0)
    @ row(),61 say str(_labour,7,0)
    @ row(),69 say str(subtotal,9,0)
    subcost = subcost+subtotal
    totlabour = totlabour+_labour
    subelement= subelement+(qty*price)

```

```

        replace labour with _labour,total with subtotal
        row=row+1
        skip
    enddo
    access=INT(access/100*subelement)
    totelement=subelement+access
    totalcost=totelement+totlabour
    @ 1+row(),1 say str(cnt+1,2,0)
    @ row(),5 say 'อุปกรณ์เพิ่มเติม'
    @ row(),44 say '1'
    @ row(),47 say 'LOT'
    @ row(),52 say str(access,7,0)
    @ row(),69 say str(access,9,0)
    @ 1+row(),0 say repl('-',79) color g/n
    @ 1+row(),32 say 'sub-total' color g/n
    @ row(),51 say str(totelement,8,0)
    @ row(),60 say str(totlabour,8,0)
    @ row(),69 say str(totalcost,9,0)
    @ 1+row(),48 say repl('=',31) color g/n
    DQ ADDCOST
    close database
    RETURN

```

PROCEDURE PRINT2

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +*
*+++++*
set talk off
set century on
set mark to '.'
set date to british
set color to w+/n,w+/n
clear
_ *+++++*
_ *++ แสดงผลการประมาณราคาบนจอภาพ ++*

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

_*+*****+*
set,talk off
set color to w+/n
private cnt,value,_value
public totelement,totlabour,totalcost
use d:\project\filedata\panel_b
store 15 to access
store 20 to value
store 0 to _value
store 0 to cnt
store 0 to row
store 0 to _labour
store 0 to subtotal
store 0 to subcost
store 0 to totalcost
store 0 to totlabour
store 0 to subelement
store 0 to totelement
store 'PANEL BOARD' to nameprj
go top
DO< HEADER2
@ 2,0 say "
@ row(),0 say repl ('=',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl ('=',79) color g/n
@ 9,23 say 'ค่าติดตั้งอุปกรณ์ [%] = %'
@ 10,23 say 'ค่าอุปกรณ์เพิ่มเติม [%] = %'
@ 9,47 get value picture '99' valid value > 0 error ' ป้อนค่าผิด'
@ 10,49 get access picture '99' valid value > 0 error ' ป้อนค่าผิด'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()

    cnt = cnt + 1
    _labour=INT(_value*price*qty)
    subtotal=qty*price+_labour
    @ 5+row,0 say "
    @ 5+row,1 say str(cnt,2,0)
    @ 5+row,5 say descrip
    @ 5+row,40 say str(qty,5,0)
    @ 5+row,47 say unit
    @ 5+row,52 say str(price,7,0)
    @ 5+row,61 say str(_labour,7,0)
    @ 5+row,69 say str(subtotal,9,0)
    subcost = subcost+subtotal
    totlabour = totlabour+_labour
    subelement= subelement+(qty*price)
    replace labour with _labour,total with subtotal
    row=row+1
    skip
enddo
access=INT(access/100*subelement)
totelement=subelement+access
totalcost=totelement+totlabour
@ 1+row(),1 say str(cnt+1,2,0)
@ row(),5 say 'อุปกรณ์เพิ่มเติม'
@ row(),44 say '1'
@ row(),47 say 'LOT'
@ row(),52 say str(access,7,0)
@ row(),69 say str(access,9,0)
@ 1+row(),0 say repl('-',79) color g/n
@ 1+row(),32 say 'sub-total' color g/n
@ row(),51 say str(toelement,8,0)
@ row(),60 say str(totlabour,8,0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ row(),69 say str(totalcost,9,0)
@ 1+row(),48 say repl('=',31) color g/n
DO,ADDCOST
close database
RETURN

```

PROCEDURE PRINT3

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +*
*+++++*
set talk off
set century on
set mark to '.'
set date to british
set color to w+/n,w+/n
clear
_ *+++++*
_ *++ แสดงผลการประมาณราคาบนจอภาพ ++*
_ *+++++*
set talk off
set color to w+/n
private cnt,value,_value
public toelement,totlabour,totalcost
use d:\project\filedata\luminair
store 15 to access
store 20 to value
store 0 to _value
store 0 to cnt
store 0 to row
store 0 to _labour
store 0 to subtotal
store 0 to subcost
store 0 to totalcost
store 0 to totlabour
store 0 to subelement

```

```

store 0 to totelement
store 'LUMINAIRE' to nameprj
go top
DO HEADER3
@ 2,0 say "
@ row(),0 say repl('=',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl('=',79) color g/n
@ 9,23 say 'ค่าติดตั้งอุปกรณ์ [%] = %'
@ 10,23 say 'ค่าอุปกรณ์เพิ่มเติม [%] = %'
@ 9,47 get value picture '99' valid value > 0 error ' ป้อนค่าผิด'
@ 10,49 get access picture '99' valid value > 0 error ' ป้อนค่าผิด'
read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()
    cnt = cnt + 1
    _labour=INT(_value*price*qty)
    subtotal=qty*price+_labour
    @ 5+row(),0 say "
    @ row(),1 say str(cnt,2,0)
    @ row(),5 say descrip
    @ row(),40 say str(qty,5,0)
    @ row(),47 say unit
    @ row(),52 say str(price,7,0)
    @ row(),61 say str(_labour,7,0)
    @ row(),69 say str(subtotal,9,0)
    subcost = subcost+subtotal
    totlabour = totlabour+_labour

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

subelement= subelement+(qty*price)

replace labour with _labour,total with subtotal

row=row+1

skip

enddo

access=INT(access/100*subelement)
totelement=subelement+access

totalcost=totelement+totlabour

@ 1+row(),1 say str(cnt+1,2,0)
@ row(),5 say 'อุปกรณ์เพิ่มเติม'
@ row(),44 say 1
@ row(),47 say 'LOT'
@ row(),52 say str(access,7,0)
@ row(),69 say str(access,9,0)
@ 1+row(),0 say repl('-',79) color g/n
@ 1+row(),32 say 'sub-total' color g/n
@ row(),51 say str(totelement,3,0)
@ row(),60 say str(totlabour,8,0)
@ row(),69 say str(totalcost,9,0)
@ 1+row(),48 say repl('=',31) color g/n

DO ADDCOST

close database

RETURN

```

PROCEDURE PRINT4

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +*
*+++++*

set talk off

set century on

set mark to '-'

set date to british

set color to w+/n,w+/n

clear

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

_*++ แสดงผลการประมาณราคานจภาพ ++*
_*+++++*

set talk off

set color to w+/n

private cnt,value,_value

public totalement,totlabour,totalcost

use'd\project\filedata\sw_recep

store 15 to _access

store 20 to value

store 0 to _value

store 0 to cnt

store 0 to row

store 0 to _labour

store 0 to subtotal

store 0 to subcost

store 0 to totalcost

store 0 to totlabour

store 0 to subelement

store 0 to totalement

store 'SWITCH & RECEPTACLE' to nameprj

go top

DO HEADER4

@ 2,0 say "

@ row(),0 say repl ('=',79) color g/n

@ row()+1,0 say 'Item' color g/n

@ row(),10 say 'Description' color g/n

@ row(),42 say 'Qty' color g/n

@ row(),47 say 'Unit' color g/n

@ row(),53 say 'UnitPrice' color g/n

@ row(),64 say 'Labour' color g/n

@ row(),73 say 'Total' color g/n

@ row()+1,0 say repl ('=',79) color g/n

@ 9,23 say 'ค่าติดตั้งอุปกรณ์ [%] = %'

@ 10,23 say 'ค่าอุปกรณ์เพิ่มเติม [%] = %'

@ 9,47 get value picture '99' valid value > 0 error ' ป้อนค่าผิด'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 10,49 get access picture '99' valid value > 0 error ' ป้อนค่าผิด'
read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()
    cnt = cnt + 1
    _labour=INT(_value*price*qty)
    subtotal=qty*price+_labour
    @ 5+row,0 say "
    @ row(),1 say str(cnt,2,0)
    @ row(),5 say descrip
    @ row(),40 say str(qty,5,0)
    @ row(),47 say unit
    @ row(),52 say str(price,7,0)
    @ row(),61 say str(_labour,7,0)
    @ row(),69 say str(subtotal,9,0)
    subcost = subcost+subtotal
    totlabour = totlabour+_labour
    subelement= subelement+(qty*price)
    replace labour with _labour,total with subtotal
    row=row+1
    skip
enddo
access=INT(access/100*subelement)
totelement=subelement+access
totalcost=totelement+totlabour
@ 1+row(),1 say str(cnt+1,2,0)
@ row(),5 say 'อุปกรณ์เพิ่มเติม'
@ row(),44 say '1'
@ row(),47 say 'LOT'
@ row(),52 say str(access,7,0)
@ row(),69 say str(access,9,0)
@ 1+row(),0 say repl('-',79) color g/n
@ 1+row(),32 say 'sub-total' color g/n
@ row(),51 say str(totelement,8,0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ row(),60 say str(totlabour,8,0)
@ row(),69 say str(totalcost,9,0)
@ 1+row(),48 say repl('=','31) color g/n
DO ADDCOST
close database
RETURN

```

PROCEDURE PRINT5

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +*
*+++++*
set talk off
# define k_esc 27
set century on
set mark to '.'
set date to british
set color to w+/n,w+/n
clear
_ *+++++*
_ *++ แสดงผลการประมาณราคานอกราชภาพ ++*
_ *+++++*
set talk off
private cnt,value,_value,access
public totalement,totlabour,totalcost
use d:\project\filedata\wire
store 15 to access
store 30 to value
store 0 to _value
store 0 to cnt
store 0 to row
store 0 to _labour
store 0 to subtotal
store 0 to subcost
store 0 to totalcost
store 0 to totlabour

```

```

store 0 to subelement
store 0 to totelement
store 'CABLE' to nameprj
go top
DO HEADERS
@ 2,0 say "
@ row(),0 say repl ('=',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl ('=',79) color g/n
@ 9.23 say 'ค่าติดตั้งอุปกรณ์ [%] = %'
@ 10.23 say 'ค่าอุปกรณ์เพิ่มเติม [%] = %'
@ 9.47 get value picture '99' valid value > 0 error ' ป้อนค่าผิด'
@ 10.49 get access picture '99' valid value > 0 error ' ป้อนค่าผิด'
read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()
cnt = cnt + 1
_labour=INT(_value*price*qty)
subtotal=qty*price+_labour
@ 5+row,0 say "
@ row(),1 say str(cnt,2,0)
@ row(),5 say descrip
@ row(),40 say str(qty,5,0)
@ row(),48 say unit
@ row(),54 say str(price,7,2)
@ row(),61 say str(_labour,7,0)
@ row(),69 say str(subtotal,9,0)
subcost = subcost+subtotal

```

```

totlabour = totlabour+_labour
subelement= subelement+(qty*price)
    replace labour with _labour,total with subtotal
row=row+1
skip
    if lastkey()= k_esc
        wait 'กดแป้นใดๆเพื่อจบการทำงาน '
    ; endif
enddo
access=INT(access/100*subelement)
totelement=subelement+access
totalcost=totelement+totlabour
@ 1+row(),1 say str(cnt+1,2,0)
@ row(),5 say 'อุปกรณ์เพิ่มเติม'
@ row(),44 say '1'
@ row(),47 say 'LOT'
@ row(),51 say str(access,7,0)
*
@ row(),69 say str(access,9,0)
@ 1+row(),0 say repl('-',79) color g/n
@ 1+row(),32 say 'sub-total' color g/n
@ row(),51 say str(totelement,8,0)
@ row(),60 say str(totlabour,8,0)
@ row(),69 say str(totalcost,9,0)
@ 1+row(),48 say repl('=',31) color g/n
DO ADDCOST
close database

RETURN

```

PROCEDURE PRINT6

```

*+ โปรแกรมแสดงผลข้อมูลออกทางจอภาพ +*
*+-----+*
set talk off
*
# define k_esc 27
set century on

```

```

*
set mark to ' '
}
set date to british

set color to w+/n;w+/n

clear

*+++++*

*++ แสดงผลการประมาณราคาบนจอภาพ ++*

*+++++*

set talk off

private cnt,value,_value,access

public totelement,totlabour,totalcost

use d:\project\filedata\conduit

store 15 to access

store 30 to value

store 0 to _value

store 0 to cnt

store 0 to row

store 0 to _labour

store 0 to subtotal

store 0 to subcost

store 0 to totalcost

store 0 to totlabour

store 0 to subelement

store 0 to totelement

store 'RACE WAY' to nameprj

go top

DO HEADERS

'@ 2,0 say "

@ row(),0 say repl ('=',79)    color g/n

'@ row()+1,0 say 'Item'        color g/n

@ row(),10 say 'Description'    color g/n

@ row(),42 say 'Qty'            color g/n

@ row(),47 say 'Unit'           color g/n

@ row(),53 say 'UnitPrice'      color g/n

@ row(),64 say 'Labour'         color g/n

@ row(),73 say 'Total'          color g/n

```

```

@ row()+1,0 say repl ('=',79) color g/n
@ 9.23 say 'ค่าติดตั้งอุปกรณ์ [%] = %'
@ 10.23 say 'ค่าอุปกรณ์เพิ่มเติม [%] = %'
@ 9.47 get value picture '99' valid value > 0 error ' ป้อนค่าผิด'
@ 10.49 get access picture '99' valid value > 0 error ' ป้อนค่าผิด'
read
@ 9,0 to 10,79 clear
_value=value/100
do while .not. EOF()
    cnt = cnt + 1
    _labour=INT(_value*price*qty)
    subtotal=qty*price+_labour
    @ 5+row,0 say "
    @ row(),1 say str(cnt,2,0)
    @ row(),5 say descrip
    @ row(),40 say str(qty,5,0)
    @ row(),48 say unit
    @ row(),54 say str(price,7,2)
    @ row(),61 say str(_labour,7,0)
    @ row(),69 say str(subtotal,9,0)
    subcost = subcost+subtotal
    totlabour = totlabour+_labour
    subelement= subelement+(qty*price)
    replace labour with _labour,total with subtotal
    row=row+1
skip
    if lastkey()='k_esc
        wait 'กดแป้นใดๆ เพื่อจบการทำงาน '
    endif
enddo
access=INT(access/100*subelement)
toelement=subelement+access
totalcost=totelement+totlabour
@ 1+row(),1 say str(cnt+1,2,0)
@ row(),5 say 'อุปกรณ์เพิ่มเติม'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ row0,44 say '1'
@ row0,47 say 'LOT'
- @ row0,51 say str(access,7,0)
@ row0,69 say str(access,9,0)
@ l+row0,0 say repl('-',79) color g/n
@ l+row0,32 say 'sub-total' color g/n
@ row0,51 say str(totelement,8,0)
@ row0,60 say str(totlabour,8,0)
@ row0,69 say str(totalcost,9,0)
@ l+row0,48 say repl('= ',31) color g/n
DO ADDCOST
close database
RETURN

```

PROCEDURE HEADER0

```

set talk off
public name
@ 0,2 say 'PROJECT:'
@ 0,12 say name color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER1

```

set talk off
@ 0,2 say 'PROJECT:'
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 1'
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER2

```

set talk off
@ 0,2 say 'PROJECT:'
@ 0,12 say name color gr+/n

```

@ 1,2 say 'SUB 2 :'

@ 1,12 say nameprj color gr+/n

--@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER3

set talk off

@ 0,2 say 'PROJECT :'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 3 :'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER4

set talk off

@ 0,2 say 'PROJECT :'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 4 :'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADERS

set talk off

@ 0,2 say 'PROJECT :'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 5 :'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER6

set talk off

@ 0,2 say 'PROJECT :'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 6 :'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE ADDCOST

append blank

replace descrip with 'อุปกรณ์เพิ่มเติม'

replace qty with 1,unit with 'LOT'

replace price with access,total with access

replace telement with totelement

replace tlabour with totlabour

replace tcost with totalcost

RETURN

*+++++*****

++ โปรแกรมแสดงผลออกทางจอภาพ ++

*+++++*****

SET TALK OFF

SET COLOR TO W+/N,W+/N,N

CLEAR

PUBLIC NAME

NAME=SPACE(35)

DO PRINT0

DO PRINT1

DO PRINT2

DO PRINT3

DO PRINT4

DO PRINT5

DO PRINT6

RELEASE ALL

CLEAR MEMORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCEDURE PRINTO

+-----

++ โปรแกรมการแสดงผลทางจอภาพ ++

+-----

set talk off

set color to w+/n,w+/n

set device to screen

set console on

use d:\project\data\trfile

store 5 to row

name=space(35)

go top

name=nameproj

DO HEADERO

@ 2,0 say "

@ row(),0 say repl ('=',79) color g/n

@ row()+1,0 say 'Item' color g/n

@ row(),10 say 'Description' color g/n

@ row(),42 say 'Qty' color g/n

@ row(),47 say 'Unit' color g/n

@ row(),53 say 'UnitPrice' color g/n

@ row(),64 say 'Labour' color g/n

@ row(),73 say 'Total' color g/n

@ row()+1,0 say repl ('=',79) color g/n

go top

do while .not. EOF()

@ row,1 say str(item,2,0)

@ row,7 say descrip

@ row,43 say str(qty,1,0)

@ row,47 say unit

@ row,52 say str(pricemate,8,0)

@ row,62 say str(price_lb,7,0)

@ row,70 say str(totprice,8,0)

```

row=row+1

skip

enddo

go bottom

@ row()+1,0 say repl('-',79) color g/n
@ row()+1,25 say 'sub-total' color g/n
@ row(),52 say str(submate_,8,0) color gr+/n
@ row(),62 say str(sublabour_,7,0) color gr+/n
@ row(),69 say str(subtotal_,9,0) color gr+/n
@ row()+1,25 say 'overhead' color g/n
@ row(),68 say str(overhead_,10,0) color g/n
@ row()+1,25 say 'profit' color g/n
@ row(),68 say str(profit_,10,0) color g/n
@ row()+1,25 say 'TOTAL' color g/n
@ row(),68 say str(total_,10,0) color gr+/n
@ row()+1,25 say 'vat' color g/n
@ row(),68 say str(vat_,10,0) color g/n
@ row()+1,68 say repl('-',11)
@ row()+1,25 say 'GRAND TOTAL' color g/n
@ row(),68 say str(finalltotal,10,0) color gr+/n
@ row()+1,68 say repl('-',11)

close all

wait "

RETURN

```

PROCEDURE PRINT1

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +*
*+++++*

set talk off

set century on

set mark to '-'

set date to british

set color to w+/n,w+/n

clear

```

```

set console on
use d:\project\filedata\highvolt
store 0 to cnt
store 0 to row
DO HEADER1
@ 2,0 say "
@ row(),0 say repl('-',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl('-',79) color g/n
go top
do while .not. EOF()
, cnt = cnt + 1
@ 5+row,0 say "
@ 5+row,1 say str(cnt,2,0)
@ 5+row,5 say descrip
@ 5+row,40 say str(qty,5,0)
@ 5+row,48 say unit
@ 5+row,53 say str(price,7,0)
@ 5+row,61 say str(labour,7,0)
@ 5+row,69 say str(total,9,0)
row=row+1
skip
enddo
go bottom
@ row()+1,0 say repl('-',79) color g/n
@ row()+1,32 say 'sub-total' color g/n
@ row(),52 say str(telement,8,0)
@ row(),60 say str(tlabour,8,0)
@ row(),69 say str(tcost,9,0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ row()+1,48 say repl('=' ,31) color g/n
close database
wait "
RETURN

```

PROCEDURE PRINT2

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +*
*+++++*

set talk off
set century on
set mark to '-'
set date to british
set color to w+/n,w+/n
clear
set console on
use d:\project\filedata\panel_b
store 0 to cnt
store 0 to row
DO HEADER2
@ 2,0 say "
@ row(),0 say repl('=' ,79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl('=' ,79) color g/n
go top
do while .not. EOF()
    cnt = cnt + 1
    @ 5+row,0 say "
    @ 5+row.1 say str(cnt,2,0)

```

```
@ 5+row,5 say descrip
@ 5+row,40 say str(qty,5,0)
@ 5+row,48 say unit
@ 5+row,53 say str(price,8,2)
@ 5+row,61 say str(labour,7,0)
@ 5+row,69 say str(total,9,0)
```

```
row=row+1
```

```
skip
```

```
enddo
```

```
go bottom
```

```
@ row()+1,0 say repl('-',79) color g/n
```

```
@ row()+1,32 say 'sub-total' color g/n
```

```
@ row(),50 say str(telemt,8,0)
```

```
@ row(),60 say str(tlabour,8,0)
```

```
@ row(),69 say str(toost,9,0)
```

```
@ row()+1,48 say repl('=',31) color g/n
```

```
close database
```

```
wait "
```

```
RETURN
```

```
PROCEDURE PRINT3
```

```
*+-----*
```

```
*+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +*
```

```
*+-----*
```

```
set talk off
```

```
set century on
```

```
set mark to '.'
```

```
set date to british
```

```
set color to w+/n,w+/n
```

```
clear
```

```
set console on
```

```
use d:\project\filedata\luminaire
```

```
store 0 to cnt
```

```
store 0 to row
```

```
DO HEADER3
```

```

@ 2,0 say "
@ row(),0 say repl ('=',79)    color g/n
@ row()+1,0 say 'Item'         color g/n
@ row(),10 say 'Description'    color g/n
@ row(),42 say 'Qty'           color g/n
@ row(),47 say 'Unit'          color g/n
@ row(),53 say 'UnitPrice'     color g/n
@ row(),64 say 'Labour'        color g/n
@ row(),73 say 'Total'         color g/n
@ row()+1,0 say repl ('=',79)  color g/n
go top
do while .not. EOF()
    cnt = cnt + 1
    @ 5+row,0 say "
    @ 5+row,1 say str(cnt,2,0)
    @ 5+row,5 say descrip
    @ 5+row,40 say str(qty,5,0)
    @ 5+row,48 say unit
    @ 5+row,54 say str(price,7,2)
    @ 5+row,61 say str(labour,7,0)
    @ 5+row,69 say str(total,9,0)
    row=row+1
    skip
enddo
go bottom
@ row()+1,0 say repl('-',79) color g/n
@ row()+1,32 say 'sub-total' color g/n
@ row(),50 say str(telement,8,0)
@ row(),60 say str(tlabour,8,0)
@ row(),69 say str(tcost,9,0)
@ row()+1,48 say repl ('=',31) color g/n
close database
wait "
RETURN

```

PROCEDURE PRINT4

*+*****+

+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +

*+*****+

set talk off

set century on

set mark to '.'

set date to british

set color to w+/n,w+/n

clear

set console on

use d:\project\filedata\sw_recep

store 0 to cnt

store 0 to row

DO HEADER4

@ 2,0 say "

@ row(),0 say repl ('=',79) color g/n

@ row()+1,0 say 'Item' color g/n

@ row(),10 say 'Description' color g/n

@ row(),42 say 'Qty' color g/n

@ row(),47 say 'Unit' color g/n

@ row(),53 say 'UnitPrice' color g/n

@ row(),64 say 'Labour' color g/n

@ row(),73 say 'Total' color g/n

@ row()+1,0 say repl ('=',79) color g/n

go top

do while .not. EOF()

cnt = cnt + 1

@ 5+row,0 say "

@ 5+row,1 say str(cnt,2,0)

@ 5+row,5 say descrip

@ 5+row,40 say str(qty,5,0)

@ 5+row,48 say unit

@ 5+row,54 say str(price,7,2)

@ 5+row,61 say str(labour,7,0)

```

@ 5+row,69 say str(total,9,0)

row=row+1

skip

enddo

gđ bottom

@,row()+1,0 say repl('-',79) color g/n

@,row()+1,32 say 'sub-total' color g/n

@,row(),50 say str(telement,8,0)

@,row(),60 say str(tlabour,8,0)

@,row(),69 say str(tcost,9,0)

@,row()+1,48 say repl('=',31) color g/n

close database

wait "

RETURN

```

PROCEDURE PRINT5

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +*
*+++++*

set talk off

set century on

set mark to '.'

set date to british

set color to w+/n,w+/n

clear

set console on

use d:\project\file\data\wire

store 0 to cnt

store 0 to row

DO HEADERS

@ 2,0 say "

@,row(),0 say repl('-',79) color g/n

@,row()+1,0 say 'Item' color g/n

@,row(),10 say 'Description' color g/n

@,row(),42 say 'Qty' color g/n

```

```

@ row(),47 say 'Unit'          color g/n
@ row(),53 say 'UnitPrice'     color g/n
@ row(),64 say 'Labour'        color g/n
@ row(),73 say 'Total'         color g/n
@ row()+1,0 say repl('-',79)   color g/n

go top
do while .not. EOF()
    cnt = cnt + 1

    @ 5+row,0 say "
    @ 5+row,1 say str(cnt,2,0)
    @ 5+row,5 say descrip
    @ 5+row,40 say str(qty,5,0)
    @ 5+row,48 say unit
    @ 5+row,54 say str(price,7,2)
    @ 5+row,61 say str(labour,7,0)
    @ 5+row,69 say str(total,9,0)
    row=row+1
    skip
enddo
go bottom
@ row()+1,0 say repl('-',79) color g/n
@ row()+1,32 say 'sub-total' color g/n
@ row(),50 say str(telement,8,0)
@ row(),60 say str(tlabour,8,0)
@ row(),69 say str(tcost,9,0)
@ row()+1,48 say repl('-',31) color g/n
close database
wait "
RETURN

```

PROCEDURE PRENT6

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Screen +*
*+++++*

set talk off

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set century on
set mark to '-'
set date to british
set color to w+/n,w+/n
clear
set console on
use d:\project\filedata\conduit
store 0 to cnt
store 0 to row
DO HEADER6
@@2,0 say '
@ row(),0 say repl ('=',79) color g/n
@ ,row()+1,0 say 'Item' color g/n
@ ,row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ ,row(),47 say 'Unit' color g/n
@ ,row(),53 say 'UnitPrice' color g/n
@ ,row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl ('=',79) color g/n
go top
do*while .not. EOF()
    cnt = cnt + 1
    @ 5+row,0 say "
    @ 5+row,1 say str(cnt,2,0)
    @ 5+row,5 say descrip
    @ 5+row,40 say str(qty,5,0)
    @ 5+row,48 say unit
    @ 5+row,54 say str(price,7,2)
    @ 5+row,61 say str(labour,7,0)
    @ 5+row,69 say str(total,9,0)
    row=row+1
    skip
enddo
go bottom

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ row()+1,0 say repl('-',79) color g/n
@ row()+1,32 say 'sub-total' color g/n
@ row(),50 say str(telement,8,0)
@ row(),60 say str(labour,8,0)
@ row(),69 say str(lcost,9,0)
@ row()+1,48 say repl('=' ,31) color g/n
close database
wait "
RETURN

```

PROCEDURE HEADER0

```

set talk off
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER1

```

set talk off
store 'HIGH VOLT' to nameprj
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 1 : '
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER2

```

set talk off
store 'PANEL BOARD' to nameprj
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 2 : '
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'

```

RETURN

PROCEDURE HEADER3

set talk off

store 'LUMINAIRE' to nameprj

@ 0,2 say 'PROJECT .'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 3 .'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER4

set talk off

store 'SWITCH & RECEPTACLE' to nameprj

@ 0,2 say 'PROJECT .'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 4 .'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER5

set talk off

store 'CABLE' to nameprj

@ 0,2 say 'PROJECT .'

@ 0,12 say name color gr+/n

@ 1,2 say 'SUB 5 .'

@ 1,12 say nameprj color gr+/n

@ 1,34 say 'QUOTE PRICE'

RETURN

PROCEDURE HEADER6

set talk off

store 'RACE WAY' to nameprj

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB ๑ : '
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

```
*****
```

```
*+++++*
```

```
*++ โปรแกรมแสดงผลออกทางเครื่องพิมพ์ ++*
```

```
*+++++*
```

```
SET TALK OFF
```

```
SET COLOR TO W+N,W+N,N
```

```
CLEAR
```

```
PUBLIC NAME
```

```
NAME=SPACE(35)
```

```
DO OPRINT0
```

```
WAIT "
```

```
DO OPRINT1
```

```
WAIT "
```

```
DO OPRINT2
```

```
WAIT "
```

```
DO OPRINT3
```

```
WAIT "
```

```
DO OPRINT4
```

```
WAIT "
```

```
DO OPRINT5
```

```
WAIT "
```

```
DO OPRINT6
```

```
SET COLOR TO W+N,W+N,N
```

```
RELEASE ALL
```

```
CLEAR MEMORY
```

```
PROCEDURE OPRINT0
```

```
*+++++*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*++ โปรแกรมการแสดงผลทางเครื่องพิมพ์ ++*
*+++++*
;
set talk off
;
set color to w+/n,w+/n

'set device to printer

set console on _ ->

use d:\project\data\file

store 5 to row
',
name=space(35)

go top

name=nameproj
DO HEADER0

@ 2,0 say "

@ row0,0 say repl('=','79) color g/n
@ row0+1,0 say 'Item' color g/n
@ row0,10 say 'Description' color g/n
@ row0,42 say 'Qty' color g/n
@ row0,47 say 'Unit' color g/n
@ row0,53 say 'UnitPrice' color g/n
@ row0,64 say 'Labour' color g/n
@ row0,73 say 'Total' color g/n
@ row0+1,0 say repl('=','79) color g/n

go top

do while .not. EOF()

    @ row,1 say str(item,2,0)

    @ row,7 say descrip

    @ row,43 say str(qty,1,0)

    @ row,47 say unit

    @ row,52 say str(pricemate,8,0)

    @ row,62 say str(price_lb,7,0)

    @ row,70 say str(totprice,8,0)

    row=row+1

    skip

enddo

go bottom

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ prow()+1,0 say repl('-',79) color g/n
@ prow()+1,25 say 'sub-total' color g/n

@ prow(),52 say str(submate_,8,0) color gr+/n
@ prow(),62 say str(sublabour_,7,0) color gr+/n
@ prow(),69 say str(subtotal_,9,0) color gr+/n

@ prow()+1,25 say 'overhead' color g/n
@ prow(),68 say str(overhead_,10,0) color g/n
@ prow()+1,25 say 'profit' color g/n
@ prow(),68 say str(profit_,10,0) color g/n
@ prow()+1,25 say 'TOTAL' color g/n
@ prow(),68 say str(total_,10,0) color gr+/n
@ prow()+1,25 say 'vat' color g/n
@ prow(),68 say str(vat_,10,0) color g/n
@ prow()+1,68 say repl('-',11)
@ prow()+1,25 say 'GRAND TOTAL' color g/n
@ prow(),68 say str(finaltotal,10,0) color gr+/n
@ prow()+1,68 say repl('=',11)
close all
set device to screen

@ 8,26 say 'ไปรครอหลักๆ กำลังพิมพ์รายงานอยู่' color gr+/n
@ 9,29 say 'ESTIMATING ELECTRICAL' color gr+/n
@ 11,17 say 'เมื่อพิมพ์เสร็จ กดคีย์ใดๆเพื่อพิมพ์ SUB 1 : HIGH VOLTAGE' color w+/n
wait "
EJECT
RETURN

```

PROCEDURE OPRINT1

```

*+++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Printer +*
*+++++*

set talk off
set century on
set mark to '-'
set date to british
set color to w+/n.w+/n

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clear

set device to printer

set console on

use d:\project\filedata\highvolt

store 0 to cnt

store 0 to row

DO HEADER1

@ 2,0 say "

@ prow(),0 say repl('=','79)   color g/n

@ prow()+1,0 say 'Item'       color g/n

@@ prow(),10 say 'Description' color g/n

@ prow(),42 say 'Qty'         color g/n

@@ prow(),47 say 'Unit'       color g/n

@ prow(),53 say 'UnitPrice'   color g/n

@@ prow(),64 say 'Labour'     color g/n

@@ prow(),73 say 'Total'      color g/n

@ prow()+1,0 say repl('=','79) color g/n

go top

do, while .not. EOF()

    cnt = cnt + 1

    @ 5+row,0 say "

    @ 5+row,1 say str(cnt,2,0)

    @ 5+row,5 say descrip

    @ 5+row,40 say str(qty,5,0)

    @ 5+row,48 say unit

    @ 5+row,54 say str(price,7,2)

    @ 5+row,61 say str(labour,7,0)

    @ 5+row,69 say str(total,9,0)

    row=row+1

    skip

enddo

go bottom

@ prow()+1,0 say repl('-',79) color g/n

@ prow()+1,32 say 'sub-total' color g/n

@ prow(),51 say str(telement,8,0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@ prow(),60 say str(tlabour,8,0)
@ prow(),69 say str(tcost,9,0)
@@ prow()+1,48 say repl('=',31) color g/n
close database

set device to screen
@ 8,26 say 'โปรแกรมสต็อกวัสดุ กำลังพิมพ์รายงานอยู่' color gr+/n
@ 9,30 say 'SUB 1 : HIGH VOLTAGE' color gr+/n
@ 11,17 say 'เมื่อพิมพ์เสร็จ กดคีย์ใดๆเพื่อพิมพ์ SUB 2 . PANEL BOARD' color w+/n
wait "
EJECT
RETURN

```

PROCEDURE OPRINT2

```

*+-----*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Printer +*
*+-----*

```

```

set talk off
set century on
set mark to '.'
set date to british
set color to w+/n,w+/n
clear
set device to printer
set console on
use d:\project\filedata\panel_b

```

```
store 0 to cnt
```

```
store 0 to row
```

DO HEADER2

```

@ 2,0 say "
@ prow(),0 say repl('=',79) color g/n
@ prow()+1,0 say 'Item' color g/n
@ prow(),10 say 'Description' color g/n
@ prow(),42 say 'Qty' color g/n
@ prow(),47 say 'Unit' color g/n
@ prow(),53 say 'UnitPrice' color g/n

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

@ prow(),64 say 'Labour'      color g/n
@ prow(),73 say 'Total'      color g/n
@ prow()+1,0 say repl('=' ,79) color g/n
go top
do while .not. EOF()

    cnt = cnt + 1

    @ 5+row,0 say "
    @ 5+row,1 say str(cnt,2,0)
    @ 5+row,5 say descrip
    @ 5+row,40 say str(qty,5,0)
    @ 5+row,48 say unit
    @ 5+row,54 say str(price,7,2)
    @ 5+row,61 say str(labour,7,0)
    @ 5+row,69 say str(total,9,0)
    row=row+1
    .skip
enddo
go bottom
@ prow()+1,0 say repl('-',79) color g/n
@ prow()+1,32 say 'sub-total' color g/n
@ prow(),51 say str(telement,8,0)
@ prow(),60 say str(tlabour,8,0)
@ prow(),69 say str(tcost,9,0)
@ prow()+1,48 say repl('=' ,31) color g/n
close database

set device to screen

@ 8.26 say 'โปรดรอสักครู่ กำลังพิมพ์รายงานอยู่' color gr+/n
@ 9,30 say 'SUB 2 : PANEL BOARD' color gr+/n
@ 11,18 say 'เมื่อพิมพ์เสร็จ กดคีย์ใดๆเพื่อพิมพ์ SUB 3 : LUMINAIRE' color w+/n
wait "
EJECT
RETURN

PROCEDURE OPRINT3

*+++++*

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ โปรแกรมแสดงผลข้อมูลออกทาง Printer +

!+++++

set talk off

set century on

set mark to '-'

set date to british

set color to w+/n,w+/n

clear

set device to printer

set console on

use d:\project\filedata\Luminaire

store 0 to cnt

store 0 to row

DO HEADERS

@ 2,0 say "

@ prow(),0 say repl ('=',79) color g/n

@ prow()+1,0 say 'Item' color g/n

@ prow(),10 say 'Description' color g/n

@ prow(),42 say 'Qty' color g/n

@ prow(),47 say 'Unit' color g/n

@.prow(),53 say 'UnitPrice' color g/n

@ prow(),64 say 'Labour' color g/n

@ prow(),73 say 'Total' color g/n

@.prow()+1,0 say repl ('=',79) color g/n

sq top

do while .not. EOF()

cnt = cnt + 1

@ 5+row,0 say "

@ 5+row,1 say str(cnt,2,0)

@ 5+row,5 say descrip

@ 5+row,40 say str(qty,5,0)

@ 5+row,48 say unit

@ 5+row,54 say str(price,7,2)

@ 5+row,61 say str(labour,7,0)

@ 5+row,69 say str(total,9,0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

row=row+1

skip

enddo

go bottom

@ prw()+1,0 say repl('-',79) color g/n

@ prw()+1,32 say 'sub-total' color g/n

@ prw(),51 say str(telement,8,0)

@ prw(),60 say str(tlabour,8,0)

@ prw(),69 say str(tcost,9,0)

@ prw()+1,48 say repl('= ',31) color g/n

close database

set device to screen

@ 8,26 say 'โปรดรอสักครู์ กำลังพิมพ์รายงานอยู่' color gr+/n

@ 9,31 say 'SUB 3 : LUMINAIRE' color gr+/n

@ 11,13 say 'เมื่อพิมพ์เสร็จ กดคีย์ใดๆเพื่อพิมพ์ SUB 4 : SWITCH & RECEPTICLE' color w+/n

wait "

EJECT

RETURN

*

PROCEDURE OPRINT4

*+-----*

*+ โปรแกรมแสดงผลข้อมูลออกทาง Printer +*

*+-----*

set talk off

set century on

set mark to '-'

set date to british

set color to w+/n,w+/n

clear

set device to printer

set console on

use d:\project\filedata\sw_recep

store 0 to cnt

store 0 to row

DO HEADER4

```

```

@ 2,0 say "
@ prow(),0 say repl('=' ,79)   color g/n
@ prow()+1,0 say 'Item'         color g/n
@ prow(),10 say 'Description'   color g/n
@ prow(),42 say 'Qty'           color g/n
@ prow(),47 say 'Unit'          color g/n
@ prow(),53 say 'UnitPrice'     color g/n
@ prow(),64 say 'Labour'        color g/n
@ prow(),73 say 'Total'         color g/n
@ prow()+1,0 say repl('=' ,79)   color g/n

```

go top

do while .not. EOF()

```
cnt = cnt + 1
```

```
@ 5+row,0 say "
```

```
@ 5+row,1 say str(cnt,2,0)
```

```
@ 5+row,5 say descrip
```

```
@ 5+row,40 say str(qty,5,0)
```

```
@ 5+row,48 say unit
```

```
@ 5+row,54 say str(price,7,2)
```

```
@ 5+row,61 say str(labour,7,0)
```

```
@ 5+row,69 say str(total,9,0)
```

```
row=row+1
```

```
skip
```

enddo

go bottom

```
@ prow()+1,0 say repl('-',79) color g/n
```

```
@ prow()+1,32 say 'sub-total' color g/n
```

```
@ prow(),51 say str(telement,8,0)
```

```
@ prow(),60 say str(tlabour,8,0)
```

```
@ prow(),69 say str(tcost,9,0)
```

```
@ prow()+1,48 say repl('=' ,31) color g/n
```

close database

set device to screen

```
@ 8,26 say 'โปรแกรมสต็อกตู้ กำลังพิมพ์รายงานอยู่' color gr+/n
```

```
@ 9,26 say 'SUB 4 : SWITCH & RECEPTICLE' color gr+/n
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@ 11,20 say 'เมื่อพิมพ์เสร็จ กดที่ปุ่มใดๆเพื่อพิมพ์ SUB 5 : CABLE' color w+/n
wait "
EJECT
RETURN
```

PROCEDURE OPRINT5

```
*+ ++++++*
*+ โปรแกรมแสดงผลข้อมูลออกทาง Printer +*
*+ ++++++*

set talk off
set century on
set mark to '-'
set date to british
set color to w+/n,w+/n
clear
set device to printer
set console on
use d:\project\file\data\wire
store 0 to cnt
store 0 to row
DO HEADERS
@ 2,0 say "
@ row(),0 say repl ('=',79) color g/n
@ row()+1,0 say 'Item' color g/n
@ row(),10 say 'Description' color g/n
@ row(),42 say 'Qty' color g/n
@ row(),47 say 'Unit' color g/n
@ row(),53 say 'UnitPrice' color g/n
@ row(),64 say 'Labour' color g/n
@ row(),73 say 'Total' color g/n
@ row()+1,0 say repl ('=',79) color g/n
go top
do while .not. EOF()
    cnt = cnt + 1
    @ 5+row,0 say "
```


;

*

clear

set device to printer

set console on

use d:\project\filedata\conduit

store 0 to cnt

store 0 to row

DO HEADER6

@ 2,0 say "

@ prow(),0 say repl('=' ,79) color g/n

@ prow()+1,0 say 'Item' color g/n

@ prow(),10 say 'Description' color g/n

@ prow(),42 say 'Qty' color g/n

@ prow(),47 say 'Unit' color g/n

@ prow(),53 say 'UnitPrice' color g/n

@ prow(),64 say 'Labour' color g/n

@ prow(),73 say 'Total' color g/n

@ prow()+1,0 say repl('=' ,79) color g/n

go top

do while .not. EOFQ

cnt = cnt + 1

@ 5+row,0 say "

@ 5+row,1 say str(cnt,2,0)

@ 5+row,5 say descrip

@ 5+row,40 say str(qty,5,0)

@ 5+row,48 say unit

@ 5+row,54 say str(price,7,2)

@ 5+row,61 say str(labour,7,0)

@ 5+row,69 say str(total,9,0)

row=row+1

skip

enddo

go bottom

@ prow()+1,0 say repl('-',79) color g/n

@ prow()+1,32 say 'sub-total' color g/n

@ prow(),51 say str(telement,8,0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:
@ prow(),60 say str(tlabour,8,0)
@ prow(),69 say str(tcost,9,0)
@* prow()+1,48 say repl('=','31) color g/n
close database
set device to screen
@ 8,26 say 'ไปตรวจตักถั่ว กำลังพิมพ์รายงานอยู่' color gr+/n
@ 9,31 say 'SUB 6 : RACE WAY' color gr+/n
@* 11,30 say 'กดที่ขี้ไคๆ เมื่อพิมพ์เสร็จ' color w+/n
wait "
EJECT
RETURN
:

```

PROCEDURE HEADER0

```

set talk off
:
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER1

```

set talk off
store 'HIGH VOLT' to nameprj
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 1 : '
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

PROCEDURE HEADER2

```

set talk off
store 'PANEL BOARD' to nameprj
@ 0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 2 : '

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

PROCEDURE HEADER3

set talk off
store 'LUMINAIRE' to nameprj
@ 0,2 say 'PROJECT :'
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 3 :'
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

PROCEDURE HEADER4

set talk off
store 'SWITCH & RECEPTICLE' to nameprj
@ 0,2 say 'PROJECT :'
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 4 :'
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

PROCEDURE HEADERS

set talk off
store 'CABLE' to nameprj
@ 0,2 say 'PROJECT :'
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 5 :'
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

PROCEDURE HEADER6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set talk off
store 'RACE WAY' to nameprj
@-0,2 say 'PROJECT : '
@ 0,12 say name color gr+/n
@ 1,2 say 'SUB 6  : '
@ 1,12 say nameprj color gr+/n
@ 1,34 say 'QUOTE PRICE'
RETURN

```

```

*+++++*
*++ โปรแกรมย่อย แก้ไขไฟล์ฐานข้อมูล ++*
*+++++*

#define K_ESC 27
set talk off
set century on
set date to ITALIAN
set mark to '-'
set color to w+/b,w+/rb,g
private choice
store .t. to mchoice
clear
@ 6,14 fill to 17,65 panel color /g
@ 7,15 fill to 16,64 panel color /b
@ 11,24 to 16,53 double color w+/b
@ 8,21 say '### โปรดเลือกรายการแก้ไขไฟล์ฐานข้อมูล ###' color gr+/b
@ 9,25 say repl ('=',28) color w+/b
do while mchoice
store 0 to choice
@ 12,25 prompt ' 1. เพิ่มรายการอุปกรณ์ ' color w+/rb,g/b+,g
@ 13,25 prompt ' 2. แก้ไขราคาอุปกรณ์ ' color w+/rb,g/b+,g
@ 14,25 prompt ' 3. ลบรายการอุปกรณ์ ' color w+/rb,g/b+,g
@ 15,25 prompt ' 4. จบการทำงาน ' color w+/rb,g/b+,g
menu to choice
save screen to editdata

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do case
    case choice = 1
        DO ADDDATA
    case choice = 2
        DO CHANGE
    case choice = 3
        DO DELDATA
    case choice = 4
        mchoice= .f. &&EXIT
endcase
restore screen from editdata
enddo
set color to w+/b,w+/b,b
release all
clear memory
clear
*
PROCEDURE ADDDATA
*+++++*
*++ โปรแกรมย่อยเพิ่มข้อมูลใหม่ ++*
*+++++*
# define K_ESC 27
set talk off
set century on
set date to ITALIAN
set mark to '-'
set status off
clear
set color to w+/b,n/g,g
use d:\fox25\filedata
do while .t.
{
clear
@ 5,10 fill to 5,70 color /g
@ 19,10 fill to 19,70 color /g
@ 5,27 say 'โปรแกรม สำหรับป้อนข้อมูลเพิ่ม' color n/g

```

```

;
;
;
@ 19,30 say 'กด ESC = จบการทำงาน' color b/g
@ 6,10 to 15,70 double color r+/b

store 1 to tqty
store 0 to tprice
tcode=space(8)
tdescri=space(40)
tunit=space(8)
tdate=date()
;
@ 8,12 say 'โพธิ์ค'
;
@ 9,12 say 'รายละเอียด'
;
@ 10,12 say 'จำนวน'
;
@ 11,12 say 'หน่วย'
;
@ 12,12 say 'ราคา/หน่วย'
;
@ 13,12 say 'วันที่'
;
@ 8,25 get tcode
;
@ 9,25 get tdescri
;
@ 10,25 get tqty picture '####'
;
@ 11,25 get tunit
;
@ 12,25 get tprice picture '#####.##'
;
@ 13,25 get tdate picture '@D'
;
READ
;
if lastkey() = K_ESC
;
exit
;
endif
savedata='y'
;
@ 17,24 say 'ต้องการจัดเก็บข้อมูลหรือไม่ (Y/N) : get savedata picture '!' ;
valid savedata$'YNyn' ERROR ' ป้อนค่าผิด'
;
read
;
if upper(savedata) = 'Y'
;
append blank
;
replace code with tcode,descri with tdescri
;
replace qty with tqty,unit with tunit,price with tprice
;
replace date with tdate
;
else
;
loop
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

endif
enddo
set color to w+/b,w+/rb,g
close database
clear
RETURN

```

PROCEDURE CHANGE

```

*+++++*
*++ โปรแกรมย่อย เปลี่ยนแปลงราคาอุปกรณ์ ++*
*+++++*

# define K_ESC 27

set talk off
set century on
set date to ITALIAN
set mark to '-'
set color to w+/b,w+/rb,g
private choice
store .t. to mchoice
do while mchoice
clear
@ 7,14 fill to 17,65 panel color /g
@ 8,15 fill to 16,64 panel color /b
@ 12,25 to 16,53 double color w+/b
@ 9,20 say '### โปรดเลือกรายการเปลี่ยนแปลงราคาอุปกรณ์ ###' color gr+/b
@ 10,24 say repl ('=.32) color w+/b

store 0 to choice
@ 13,28 prompt ' 1. แก้ไขราคาทั้งหมด ' color w+/rb,b/bg,g
@ 14,28 prompt ' 2. แก้ไขราคาบางรายการ ' color w+/rb,b/bg,g
@ 15,28 prompt ' 3. จบการทำงาน ' color w+/rb,b/bg,g
menu to choice

do case

case choice = 1

DO BROW

```

```
case choice = 2
```

```
DO CHANGE_P
```

```
'case choice = 3
```

```
EXIT
```

```
endcase
```

```
enddo
```

```
clear
```

```
RETURN
```

```
PROCEDURE BROW
```

```
*+++++*****
```

```
*++ โปรแกรมย่อย แก้ไขราคาอุปกรณ์ทั้งหมด ++*
```

```
*+++++*****
```

```
set talk off
```

```
set date to ITALIAN
```

```
set mark to '.'
```

```
private ndate
```

```
ndate=date()
```

```
clear
```

```
use d:\fox25\filedata
```

```
browse fields code,descrip,price,date freeze\price nodelete
```

```
go top
```

```
do while .not. EOF()
```

```
'replace date with ndate
```

```
'
```

```
,skip
```

```
enddo
```

```
close database
```

```
clear
```

```
RETURN
```

```
PROCEDURE CHANGE_P
```

```
*+++++*****
```

```
*++ โปรแกรมย่อย แก้ไขราคาอุปกรณ์บางรายการ ++*
```

```
*+++++*****
```

```
# define K_ESC 27
```

```

set talk off
set date to ITALIAN
set mark to '.'
private ndate
use d:\fox25\filedata
set index to d:\project\filedata\codeind
reindex
clear
store 'Y' to p_ele
do while upper(p_ele) = 'Y'
store 0 to nprice
tcode = space(8)
update=date()
clear
@ 5,10 fill to 5,70 color /g
@ 18,10 fill to 18,70 color /b
@ 20,10 fill to 20,70 color /g
@ 5,29 say 'โปรแกรม แก้ไขราคาอุปกรณ์' color b/g
@ 20,30 say 'กด ESC = จบการทำงาน' color b/g
@ 6,10 to 16,70 double color w+/b
@ 7,13 say 'โค้ด :'
@ 9,13 say 'รายละเอียด :'
@ 11,13 say 'หน่วย :'
@ 13,13 say 'ราคา/หน่วย :'
@ 15,13 say 'วันที่ป้อนข้อมูล :'
@ 18,25 say 'กรุณาป้อนโค้ดที่ต้องการแก้ไขราคา : ' get tcode DEFAULT '' SIZE 1.8 color w+/b,gr+/b,b
read
if lastkey() = K_ESC
exit
endif
seek tcode
if not. FOUND()
save screen to fox1
clear
@ 6,10 to 18,70 panel color g/b

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*
@ 10.23 say '### ไม่พบเรคอร์ดที่ต้องการแก้ไขราคา ###' color gr+/b
@ 13.23 say '<<<< ----- >>>>' color g/b
@ -15.27 say 'โปรดกดคีย์ใดๆ เพื่อป้อนโค้ดใหม่' color gr+/b
wait "
restore screen from fox1
,loop
endif
@ 7.20 say code color g/b
@ 9.26 say descrip color g/b
@ 11.21 say unit color g/b
@ 13.26 say str(price,7,2) color g/b
@ 15.32 say date color g/b
@ 13.44 say 'ราคาใหม่' get nprice picture '#####.##' ;
valid nprice > 0 error ' ป้อนค่าผิด' color w+/b,gr+/b,b
@ 15.44 say 'วันที่แก้ไข' get ndate picture '@D' ;
valid ndate >= date error ' ป้อนวันที่ผิด' color w+/b,gr+/b,b
read
if nprice > 0 && เปลี่ยนราคาอุปกรณ์ &&
_change='y'
@ 22,0 to 22,79 clear
@ 22.23 say 'ต้องการเปลี่ยนราคาอุปกรณ์นี้จริง (Y/N) .' get _change picture '!';
valid _change$'YNyn' error ' ป้อนค่าผิด' color w+/b,gr+/b,b
read
if upper(_change) ='Y'
replace price with nprice,date with ndate
endif
endif
endif
*/ ตรวจสอบว่าต้องการแก้ไขราคาอุปกรณ์อื่นหรือไม่ /*
@r 22,0 to 22,79 clear
@ 22.21 say 'ต้องการแก้ไขราคาอุปกรณ์อื่นอีกหรือไม่ (Y/N) .' get p_ele picture '!';
valid p_ele$'YNyn' error ' ป้อนค่าผิด' color w+/b,gr+/b,b
read
enddo
close database
clear

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RETURN

;

PROCEDURE DELDATA

;

+++++

++ โปรแกรมย่อย ลบ/ยกเลิกการลบข้อมูล ++

+++++

define K_ESC 27

set talk off

set century on

set date to ITALIAN

set mark to .'

set color to w+/b,w+/rb,g

use d:\fox25\data

set index to d:\project\data\codeind

reindex

clear

store 'Y' to action

do while upper(action) = 'Y'

tcode = space(8)

clear

@ 5,10 fill to 5,70 color /g

@ 18,10 fill to 18,70 color /b

@ 20,10 fill to 20,70 color /g

@ 5,27 say 'โปรแกรม ลบ/ยกเลิกการลบข้อมูล' color b/g

@ 20,30 say 'กด ESC = จบการทำงาน' color b/g

@ 6,10 to 16,70 double color w+/b

@ 7,13 say 'โค้ด :'

@ 9,13 say 'รายละเอียด :'

@ 11,13 say 'หน่วย :'

@ 13,13 say 'ราคา/หน่วย :'

@ 15,13 say 'วันที่ป้อนข้อมูล :'

@ 18,28 say 'กรุณาป้อนโค้ดที่ต้องการลบ : ' get tcode DEFAULT ' ' SIZE 1,8 color w+/b,gr+/b,b

read

if lastkey() = K_ESC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
;

exit

endif

seek toode

if .not. FOUND()

    save screen to fox1

    'clear

    @ 6,10 to 18,70 panel color g/b

    @ 10,25 say '### ไม่พบเรคอร์ดที่ต้องการลบ ###' color gr+/b

    @ 13,25 say '<<<<< ----- >>>>>' color g/b

    @ 15,27 say 'โปรดกดคีย์ใดๆ เพื่อป้อนไฟล์ใหม่' color gr+/b

    wait "

    restore screen from fox1

    ;loop

endif

@ 7,20 say code color g/b

@ 9,26 say descrip color g/b

@ 11,21 say unit color g/b

@ 13,26 say str(price,7,2) color g/b

@ 15,32 say date color g/b

if delete()

    @ 7,47 say '# เรคอร์ดนี้ถูกลบแล้ว #' color gr+/b

    _recall= 'Y'

    @ 22,22 say 'ต้องการยกเลิกการลบเรคอร์ดนี้หรือไม่ (Y/N):' get _recall picture '!';

    valid _recall$'YNyn' error ' ป้อนค่าผิด' color w+/b.gr+/b,b

    read

    if _recall= 'Y'    && ยกเลิกการลบ &&

        recall

    endif

else

    _dele='n'

    @ 22,0 to 22,79 clear

    @ 22,26 say 'ต้องการลบข้อมูลนี้จริง (Y/N) !:' get _dele picture '!';

    valid _dele$'YNyn' error ' ป้อนค่าผิด' color w+/b.gr+/b,b

    read

    if upper(_dele)='Y'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    dele
endif
endif

*/- ตรวจสอบว่าต้องการลบข้อมูลหรือไม่ */
@ 22,0 to 22,79 clear
@ 22,25 say 'ต้องการลบข้อมูลหรือไม่ (Y/N) :' get action picture '!';
valid action$'YNyn' error ' ป้อนค่าผิด' color w+/b,gr+/b,b
read
enddo
close all
clear
RETURN
*****
*+++++
*++ แสดงเกี่ยวกับโปรแกรม ++*
*+++++
set talk off
clear
define window outtext from 3,6 to 22,75 shadow color w/b,n/g,gr+
activate window outtext
do previous
deactivate window outtext
clear windows
clear
release all
clear memory
return

procedure previous
set color to w/b,n/g
set talk off
clear
text

```

แสดงข้อมูลเกี่ยวกับโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้เป็นโครงการ PROJECT ของนักศึกษาชั้นปี 3 ห้อง พ
ภาควิชาวิศวกรรมไฟฟ้า สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ปีการศึกษา 2537
จัดทำโดย

นาย ธนศิลป์ สุขประทุม รหัส 35.103056

นาย พนมพร อาภัสร์พรหม รหัส 35.103066

นาย พรพัฒน์ พงศ์ศักดิ์กรำจาย รหัส 35.103070

อาจารย์ที่ปรึกษา

รศ. ศุภี บรรจงจิตร

อจ. สมชาติ จิรวินิจาร

endtext

до PCE

procedure PCE

set color to gr+/r,n/g

set escape on

private choice,key

store 0 to choice

store 0 to key

do while choice < 3

@ 17,5 prompt 'Previous'

@ 17,30 prompt ' Next '

@ 17,54 prompt ' Exit '

menu to choice

do case

case choice = 1

case choice = 2

DO continue

case choice = 3

QUIT

endcase

enddo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure continue

set color to w/b,n/g

clear

text

ส่วนประกอบของเครื่องคอมพิวเตอร์ที่ต้องการ

* CPU 486DLC-40 หรือ

486DX-40 หรือ

486DX2-66 หรือ

486DX4-100 หรือ

Pentium DX-60 ,DX-90 หรือ

อื่นๆ ที่สูงกว่า

RAM 2MB ,4MB หรือมากกว่า

SUPER VGA MONITOR

HARDDISK อย่างต่ำ 30 MB หรือ

120 MB หรือ

210 MB หรือ

420 MB หรือ มากกว่า

endtext

do PCE_1

!

procedure PCE_1

set color to gr+/r,n/g

private choice,key

store 0 to choice

store 0 to key

do while choice < 3

@ 17,5 prompt 'Previous'

@ 17,30 prompt ' Next '

@ 17,54 prompt ' Exit '

menu to choice

do case

case choice = 1

DO previous

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1
2
'case choice = 2
    DO continue_1
case choice = 3
    * clear windows
    QUIT
endcase
enddo

```

```

procedure continue_1
set color to w/b,n/g
clear
set talk off
text

```

การติดตั้งโปรแกรม

1. ติดตั้ง Software FoxPro Version 2.5 ไว้ใน Drive C
 ดังนี้ CAFox25
2. ติดตั้ง Software Auto Cad Release 12 ไว้ใน Drive C
 ดังนี้ CAAcad12

การใช้งานโปรแกรม

การใช้งานโปรแกรมนั้น ได้มีอยู่ในปริญญานิพนธ์ (Thesis) แล้ว
 แต่จะบอกการใช้ย่อย่างคร่าวๆ ดังนี้

ให้เลือกหัวข้อจากเมนูตามลำดับ ดังต่อไปนี้

- 1.คำนวณหาจำนวนดวงโคม
- 2.เขียนแบบระบบไฟฟ้า
- 3.การประมาณราคา
- 4.แสดงผลทางจอภาพ (ถ้าต้องการดูผลทางจอภาพ)
- 5.แสดงผลทางเครื่องพิมพ์ (ถ้าต้องการพิมพ์รายงาน)
- 6.ออกจากเมนู (เมื่อจบการทำงาน)

```

endtext
do PCE_2

```

```

set color to gr+/r,n/g
private choice,key
store 0 to choice
store 0 to key

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do while choice < 3
    @ 17,5 prompt 'Previous'
    @ 17,54 prompt ' Exit '

menu to choice

do case

    case choice = 1
        DO continue

    case choice = 2
        *
        clear windows
        *
        clear
        QUIT

endcase
enddo

```

```

*****
*****

```



ภาคผนวก ข.

การใช้งานโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรม

การใช้งานโปรแกรมการเขียนแบบระบบไฟฟ้าและประมาณราคาคด้วยคอมพิวเตอร์นั้น ให้พิมพ์คำว่า "DAE" และกดคีย์ Enter (DAE เป็นไฟล์ .bat) จากนั้นจะแสดงข้อความแนะนำผู้เขียนโปรแกรม จนกว่าจะกดคีย์ใดๆ เพื่อเข้าสู่เมนู เพื่อใช้งานต่อไป

รายการในเมนูมีหัวข้อให้เลือก ดังนี้

1. ค้นหาจำนวนดวงโคม
2. เขียนแบบระบบไฟฟ้า
3. ประมาณราคา
4. แสดงผลทางจอภาพ
5. แสดงผลทางเครื่องพิมพ์
6. แก้ไขฐานข้อมูล
7. เกี่ยวกับโปรแกรม
8. ออกจากโปรแกรม.....สู่ DOS

รายละเอียดการใช้งานแต่ละหัวข้อ มีดังนี้

1. ค้นหาจำนวนดวงโคม

ผู้ใช้งานสามารถเลือกหัวข้อนี้ได้โดยการเลื่อนแถบแสงมายังหัวข้อนี้แล้วกด ENTER เมนูย่อยนี้จะทำหน้าที่ค้นหาจำนวนดวงโคมที่ใช้ ในบริเวณพื้นที่ที่ต้องการ เมื่อเลือกให้เมนูนี้ข้อมูลที่ต้องการใช้คือ ชนิดของพื้นที่ใช้งาน, ขนาดของพื้นที่, ชนิดและขนาดของหลอดไฟที่ต้องการใช้ โดยสามารถเลือกได้จากเมนูย่อยที่มีให้ เมื่อเลือกตัวเลือกได้แล้วให้กดคีย์ Enter และเมื่อตอบคำถามแต่ละหัวข้อได้แล้วก็จะได้ผลลัพธ์คือจำนวนดวงโคมที่ใช้, ระยะห่างระหว่างดวงโคม โดยข้อมูลที่ได้นี้จะนำไปใช้ในการเขียนแบบไฟฟ้า เพื่อให้ได้แสงสว่างที่เหมาะสมกับชนิดของงาน และนำไปใช้เป็นข้อมูลในการเขียนแบบระบบไฟฟ้าต่อไป

2. เขียนแบบระบบไฟฟ้า

ผู้ใช้งานสามารถเลือกหัวข้อนี้ได้โดยการเลื่อนแถบแสงมายังหัวข้อนี้แล้วกด ENTER เมนูนี้ใช้ในการเขียนแบบระบบไฟฟ้า โดยมีวิธีการเขียน ดังต่อไปนี้

วิธีการเขียนแบบแปลนไฟฟ้า

คล้ายกับการเขียนแบบบน Auto Cad ธรรมดา แต่สัญลักษณ์ที่ใช้ต้องเป็นสัญลักษณ์ ที่สร้างขึ้นโดยมี code กำกับด้วย วิธีเรียก สัญลักษณ์ ออกมาใช้

1. ใช้คำสั่ง insert จาก command line
2. เรียกจาก screen menu
3. เรียกจาก pop up menu

การเรียกใช้ screen menu

1. เรียกจาก Command: โดยพิมพ์ "MENU" แล้วเลือก SYMBOLS.MNU กด ENTER
2. เรียกจาก POP UP MENU เลื่อน Mouse ไปที่ Pop Up เลือก Electrical เลื่อนแถบแสงไปที่ CHMENU แล้ว Pick ที่ mouse

เมื่อ LOAD SYMBOLS.MNU ออกมาแล้ว จะปรากฏภาพบนจอดังรูป ผ ข. 1

Layer 0

12.4758,8.1196

SYMBOLS

loadMENU

redraw

EraseALL

ATTEXT

QUICK

PANEL

LIGHTING

PLUG

WIRE

CONDUIT

Command :

Command :

Command :

รูปที่ ผ ข. 1 แสดงรูปหน้าจอของ Auto Cad เมื่อเรียกสัญลักษณ์

อธิบายรายละเอียดของ MENU

[SYMBOLS] เรียก MENU หน้าแรกกลับมา (หน้าที่แสดงดังรูป) ไม่ว่า Menu จะแสดงอยู่หน้าไหน

[LOAD MENU] เปลี่ยน MENU ไปใช้ ACAD.MNU

[REDRAW] สำหรับล้างเครื่องหมายกากบาท (Blipmode) ที่เกิดจากการกำหนดตำแหน่ง

[ERASE ALL] ลบ OBJECT ทั้งหมด

[ATTEXT] สก๊ิด FILE ให้โปรแกรม ESTIMATION

[HIGH V] เรียก MENU หน้า HIGH

[PANEL] เรียก MENU หน้า PANEL

[LIGHTING] เรียก MENU หน้า LIGHTING

[PLUG] เรียก MENU หน้า PLUG

[WIRE] เรียก MENU หน้า WIRE

[CONDUIT] เรียก MENU หน้า CONDUIT

MENU หน้า HIGH ยังมี MENU ย่อยอีก 2 MENU คือ TR กับ HBREAKER

และ HBREAKER ยังมี MENU ย่อยอีก 2 MENU

[HIGH]

[TR]

[TR30] INSERT สัญลักษณ์ TRANSFORMER 30 KVA

[TR50] INSERT สัญลักษณ์ TRANSFORMER 50 KVA

[TR100] INSERT สัญลักษณ์ TRANSFORMER 100 KVA

[HBRAKER]

[MCCB]

[B30AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 30AF

[B50AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 50AF

[B100AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 100AF

[ACB]

[BA1000] INSERT สัญลักษณ์ AIR CIRCUIT BREAKER 1000 A

[BA1250] INSERT สัญลักษณ์ AIR CIRCUIT BREAKER 1250 A

[BA1600] INSERT สัญลักษณ์ AIR CIRCUIT BREAKER 1600 A

MENU หน้า PANEL ยังมี MENU ย่อย อีก 4 MENU

[PANEL]

[MCCB]

[B30AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 30 AF

[B50AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 50 AF

[B100AF] INSERT สัญลักษณ์ MOLD CASE CIRCUIT BREAKER 100 AF

[BREAKER]

[BB10] INSERT สัญลักษณ์ BRANCH BREAKER 10 A

[BB16] INSERT สัญลักษณ์ BRANCH BREAKER 16 A

[BB20] INSERT สัญลักษณ์ BRANCH BREAKER 20 A

[LOAD CEN]

[BL 12-100] INSERT สัญลักษณ์ LOAD CENTER MAIN 100 A CIRCUIT

ย่อย 12 CIRCUIT

[BL 18-100] INSERT สัญลักษณ์ LOAD CENTER MAIN 100 A CIRCUIT

ข้อ 18 CIRCUIT

[BL 24-100] INSERT สัญลักษณ์ LOAD CENTER MAIN 100 A CIRCUIT

ข้อ 24 CIRCUIT

[CONSUMER]

[BQ4_16] INSERT สัญลักษณ์ CONSUMER UNIT MAIN 16 A
CIRCUIT ข้อ 4 CIRCUIT

[BQ4_32] INSERT สัญลักษณ์ CONSUMER UNIT MAIN 32 A
CIRCUIT ข้อ 4 CIRCUIT

[BQ4_45] INSERT สัญลักษณ์ CONSUMER UNIT MAIN 45 A
CIRCUIT ข้อ 4 CIRCUIT

MENU หน้า LIGHTING ยังมี MENU ข้อ อีก 5 MENU

[LIGHTING]

[FL]

[FL 20]

[LOO6_20] INSERT สัญลักษณ์ โคมไฟขาสปริงสำหรับใส่ไฟ 1X20 W

[FL 32]

[L031_10] INSERT สัญลักษณ์ โคมไฟ 32 W ครอบแก้วสี่เหลี่ยม 10"

[FL40W]

[L006_40] INSERT สัญลักษณ์ โคมไฟขาสปริงสำหรับใส่ไฟ 1X40 W

[INCAN]

[L033_SM] INSERT สัญลักษณ์ โคมไฟอูมเนียมฝังไว้กับไฟ INCAN (เล็ก)

[L033_MD] INSERT สัญลักษณ์ โคมไฟอูมเนียมฝังไว้กับไฟ INCAN (กลาง)

[PL]

[LSL09] INSERT สัญลักษณ์ โคมไฟพร้อมหลอดเอสแอลประหยัดไฟ 9 W

[LSL18] INSERT สัญลักษณ์ โคมไฟพร้อมหลอดเอสแอลประหยัดไฟ 18 W

[HID]

[MECURY]

[L040_60] INSERT สัญลักษณ์ โคมงานนิทานแสงจันทร์ 60 W

[L040_250] INSERT สัญลักษณ์ โคมงานนิทานแสงจันทร์ 250 W

[METAL]

[LHNF250] INSERT สัญลักษณ์ คอมฟัดไลท์พร้อมชุดหลอดฮาโลด์ 250 W

[LHNF400] INSERT สัญลักษณ์ คอมฟัดไลท์พร้อมชุดหลอดฮาโลด์ 400 W

[SODIUM]

[LSDK70] INSERT สัญลักษณ์ โคมไฮเบย์พร้อมชุดหลอดโซเดียม 70 W

[LSDK150] INSERT สัญลักษณ์ โคมไฮเบย์พร้อมชุดหลอดโซเดียม 150 W

[SW]

[S1] INSERT สัญลักษณ์ สวิตช์ 1 ช่อง

[S2] INSERT สัญลักษณ์ สวิตช์ 2 ช่อง

[S3] INSERT สัญลักษณ์ สวิตช์ 3 ช่อง

[SW] INSERT สัญลักษณ์ สวิตช์ 3 ทาง

MENU หน้า PLUG

[PLUG]

[SRWS] INSERT สัญลักษณ์ เต้ารับติดผนังแบบเดี่ยว

[SRWD] INSERT สัญลักษณ์ เต้ารับติดผนังแบบคู่

[SRF] INSERT สัญลักษณ์ เต้ารับแบบฝังพื้น

MENU หน้า WIRE

[WIRE]

[THW]

[THW2.5] INSERT สัญลักษณ์ สาย THW 2.5 MM²

[THW4] INSERT สัญลักษณ์ สาย THW 4 MM²

[NYY1C]

[NYY1X1.5] INSERT สัญลักษณ์ สาย NYY1C-1.5 MM²

[NYY1X4] INSERT สัญลักษณ์ สาย NYY1C-4 MM²

[NYY3C]

[NYY3X1.5] INSERT สัญลักษณ์ สาย NYY3C-1.5 MM²

[NYY3X4] INSERT สัญลักษณ์ สาย NYY3C-4 MM²

[VAF]

[VAF2X1] INSERT สัญลักษณ์ สาย VAF 2X1 MM²

[VAF2X1.5] INSERT สัญลักษณ์ สาย VAF 2X1.5 MM²

MENU หน้า CONDUIT

[CONDUIT]

[EMT]

[EMT1/2] INSERT สัญลักษณ์ ต่อ EMT 1/2"

[EMT3/4] INSERT สัญลักษณ์ ต่อ EMT 3/4"

[IMC]

[IMC2] INSERT สัญลักษณ์ ต่อ IMC 2"

[IMC3] INSERT สัญลักษณ์ ต่อ IMC 3"

[RSC]

[RSC2] INSERT สัญลักษณ์ ต่อ RSC 2"

[RSC3] INSERT สัญลักษณ์ ต่อ RSC 3"

3. ประมาณราคา

ผู้ใช้งานสามารถเลือกหัวข้อนี้ได้ โดยการเลื่อนแถบแสงมายังหัวข้อนี้แล้วกด ENTER เมนูนี้จะทำการเชื่อมโยงข้อมูลระหว่าง FoxPro กับ Auto Cad เพื่อหาจำนวนของอุปกรณ์ที่ใช้ในแบบแปลนไฟฟ้า

เมื่อเลือกเมนูย่อยนี้จะปรากฏข้อความ " โปรดรอสักครู่ กำลังเชื่อมโยงข้อมูลอยู่ " สักครู่ก็จะปรากฏข้อความ " การเชื่อมโยงข้อมูลเสร็จสิ้นแล้ว โปรดคลิกซ้าย เพื่อทำงานต่อไป " เมื่อผู้ใช้งานคลิกซ้ายหนึ่งก็จะปรากฏข้อความต่อไปบนหน้าจอภาพ

PROJECT NAME : _____

Overhead = ____%

Profit = ____%

Vat = ____%

รอให้ผู้ใช้งานใส่ชื่อของโครงการที่ทำการประมาณราคาในหัวข้อ PROJECT NAME จากนั้นก็จะรอให้ผู้เติมค่าต่างๆ คือ Overhead, Profit, Vat โดยแต่ละหัวข้อผู้ใช้งานสามารถเลือกได้ตั้งแต่ 1 - 99 เปอร์เซ็นต์ และในแต่ละหัวข้อเมื่อผู้ใช้งานใส่ค่าแล้วให้กดคลิก Enter และเมื่อใส่ค่าต่างๆ ครบแล้ว ก็จะปรากฏรูปแบบการแสดงผล ในลำดับแรกนี้จะเป็นการแสดงผลของระบบไฟฟ้าแรงสูง (high volt) และรอให้ผู้ใช้งานใส่ค่าคิดตั้งระบบไฟฟ้าแรงสูงว่าคิดเป็นกี่เปอร์เซ็นต์ของราคาอุปกรณ์ ผู้ใช้งานสามารถใส่ค่าคิดตั้งได้ตั้งแต่ 1 - 99 เปอร์เซ็นต์ของราคาอุปกรณ์ที่ใช้ จากนั้นก็จะให้ใส่ค่าอุปกรณ์เพิ่มเติม ผู้ใช้งานสามารถเลือกใส่ได้ตั้งแต่ 1 - 99 เปอร์เซ็นต์เช่นกัน เมื่อตอบคำถามทั้ง 2 ข้อนี้แล้วก็จะปรากฏรายการอุปกรณ์ไฟฟ้าแรงสูงที่ใช้ในแบบแปลนไฟฟ้า และราคารวมทั้งหมดเกี่ยวกับระบบไฟฟ้าแรงสูง

และเมื่อคลิกซ้าย อีกหนึ่งลงไป ก็จะเป็นการแสดงผลของระบบแผงควบคุมไฟฟ้า (Panel Board)
ตอนแรกก็จะแสดงรูปแบบของระบบแผงควบคุมไฟฟ้า จากนั้นก็จะรอให้ผู้ใส่ค่าติดตั้งและค่าอุปกรณ์
เพิ่มเติม ผู้ใช้สามารถเลือกใช้ได้ตั้งแต่ 1 - 99 เปอร์เซนต์ของราคาอุปกรณ์เช่นกัน เมื่อผู้ใส่ค่าทั้ง 2
รายการแล้ว ก็จะปรากฏรายการอุปกรณ์ของระบบแผงควบคุมทั้งหมดที่ใช้ในแบบแปลนไฟฟ้า และ
ราคารวมทั้งหมดที่เกี่ยวกับระบบแผงควบคุมไฟฟ้า

เมื่อผู้คลิกซ้าย อีกครั้งหนึ่ง ก็จะเป็นการแสดงผลของระบบแสงสว่าง (Lighting) การใช้งานก็จะ
เหมือนกับระบบไฟฟ้าแรงสูงและระบบแผงควบคุมไฟฟ้า ได้กล่าวมาแล้ว จากนั้นก็จะปรากฏรายการ
อุปกรณ์ของระบบแสงสว่างที่เกี่ยวกับหลอดไฟฟ้าต่างๆ ที่ใช้ในแบบแปลนไฟฟ้า และราคารวม

เมื่อผู้คลิกซ้าย อีกครั้ง ก็จะเป็นการแสดงผลของอุปกรณ์สวิตช์และปลั๊ก (Switch & Receptacle)
การใช้งานก็เหมือนกับที่ได้กล่าวมาแล้ว เมื่อใส่ค่าติดตั้งและค่าอุปกรณ์เพิ่มเติมแล้วก็จะแสดงรายการของ
สวิตช์และปลั๊กที่ใช้ในแบบแปลนทั้งหมดและราคารวม

เมื่อผู้คลิกซ้าย อีกครั้ง ก็จะเป็นการแสดงผลของอุปกรณ์สายไฟ (Wire) โดยในขั้นแรกจะขึ้น
รายการของสายไฟที่ใช้ในแบบแปลนไฟฟ้า รอให้ผู้ใส่ความยาวของสายไฟที่ใช้ (ความยาวของสายไฟ
ที่ใช้สามารถวัดได้จากแบบแปลนไฟฟ้าที่เขียนจากโปรแกรม Auto Cad) เมื่อผู้ใส่ค่าความยาวของสาย
ไฟแล้ว ก็จะปรากฏรูปแบบการแสดงผลของสายไฟ ให้ผู้ใส่ค่าติดตั้งและค่าอุปกรณ์ จากนั้นก็จะแสดง
รายการของสายไฟที่ใช้ทั้งหมดและราคารวม

เมื่อผู้คลิกซ้าย อีกครั้ง ก็จะเป็นการแสดงผลของอุปกรณ์ทางเดินสายไฟ (Race Way) โดยในขั้น
แรกจะขึ้นรายการของท่อที่ใช้ในแบบแปลนไฟฟ้า และรอให้ผู้ใส่ความยาวของท่อที่ใช้ (ความยาวของ
ท่อสามารถคำนวณได้จากแบบไฟฟ้า) เมื่อผู้ใส่ค่าความยาวของท่อแล้วก็จะปรากฏรูปแบบการแสดงผล
ของท่อ รอให้ผู้ใส่ค่าติดตั้งและค่าอุปกรณ์เพิ่มเติม จากนั้นก็จะแสดงรายการของท่อพร้อมทั้งราคารวมทั้ง
หมด

และเมื่อผู้คลิกซ้าย อีกครั้ง ก็จะเป็นการแสดงรายการรวมทั้งหมด โดยการนำเอาราคารวมของแต่ละ
รายการมาแสดงรวมให้อยู่ในหน้าจอเดียวกัน และเป็นการแสดงราคารวมจริงที่คิดได้จากการประมาณราคา
จากแบบแปลนไฟฟ้า

4. แสดงผลทางจอภาพ

ผู้ใช้งานสามารถเลือกรายการนี้ได้โดยการเลื่อนแถบแสงมายังหัวข้อนี้แล้วกด Enter ถ้าเลือกเมนูย่อยรายการ
รนี้จะเป็นการแสดงผลของข้อมูลที่ได้จากการเชื่อมโยงออกทางจอภาพ แสดงผลจะคล้ายกับในหัวข้อ
ประมาณราคา คือ รายการแรกเป็นการแสดง ราคารวมทั้งหมดของการประมาณราคา ต่อจากนั้นก็จะเป็น
การแสดงราคาของอุปกรณ์แต่ละรายการ โดยมีลำดับการแสดงดังนี้

1. แสดงรายการรวมทั้งหมดของการประมาณราคา
2. แสดงราคาของระบบไฟฟ้าแรงสูง
3. แสดงราคาของระบบแผงควบคุมไฟฟ้า

4. แสดงราคาของอุปกรณ์หลอดไฟฟ้า
5. แสดงราคาของอุปกรณ์สวิตช์
6. แสดงราคาของอุปกรณ์สายไฟ
7. แสดงราคาของอุปกรณ์ท่อเดินสายไฟ

ในเมนูนี้ผู้ใช้สามารถเรียกดูผลของการประมาณราคาได้ จนกว่าจะมีการเชื่อมโยงข้อมูลใหม่อีก

5. การแสดงผลทางเครื่องพิมพ์

การเลือกหัวข้อนี้ให้เลื่อนแถบแสงมายังหัวข้อนี้แล้วกด Enter ในหัวข้อนี้จะเป็นการแสดงผลออกทางเครื่องพิมพ์ แทนที่จะเป็นการแสดงผลทางจอภาพอย่างในเมนูย่อยที่ 4 คือจะเป็นการพิมพ์ผลที่ได้จากการประมาณราคาลงบนกระดาษ A4 ในการพิมพ์นั้น ผู้ใช้ควรจะตรวจสอบเครื่องพิมพ์ให้เรียบร้อยก่อนว่าต่อแล้วหรือยัง การพิมพ์จะพิมพ์ ทั้งหมด 7 รายการด้วยกัน มีรายการเหมือนกับการแสดงทางจอภาพ ในการพิมพ์แต่ละรายการก็จะมีข้อความบอกว่า

" โปรดรอสักครู่ กำลังพิมพ์อยู่ "

(ชื่อรายการที่กำลังพิมพ์อยู่)

" เมื่อพิมพ์เสร็จ โปรดกดคีย์ใดๆ เพื่อพิมพ์ (ชื่อรายการที่จะพิมพ์ต่อไป) "

6. แก้ไขฐานข้อมูล

ผู้ใช้สามารถเลือกรายการแก้ไขฐานข้อมูลได้ โดยการเลื่อนแถบแสงมายังหัวข้อนี้แล้วกด Enter ถ้าเลือกรายการนี้ จะปรากฏเมนูย่อยให้เลือกอีก 3 รายการ คือ

- 1.เพิ่มรายการอุปกรณ์ใหม่
- 2.เปลี่ยนแปลงราคาอุปกรณ์
- 3.ลบรายการอุปกรณ์ออก

รายละเอียดของแต่ละรายการ และการใช้งาน มีดังนี้

ถ้าเลือกเมนูย่อยหมายเลข 1 คือเพิ่มรายการอุปกรณ์ใหม่ จะมีข้อความให้เติมเกี่ยวกับรายละเอียดของอุปกรณ์ที่ต้องการจะเพิ่มลงไปใหม่ เช่น โค้ดของอุปกรณ์, รายละเอียด, จำนวน, หน่วย, ราคา/หน่วย, วันที่คำนวณที่ พ.ศ. 2

ข้อกำหนดของการป้อนรายละเอียดต่างๆ

โค้ด - ต้องเป็นภาษาอังกฤษเท่านั้น และต้องขึ้นต้นด้วยตัวอักษรดังต่อไปนี้ T, B, L, S, W, C มีความยาวได้ไม่เกิน 8 ตัวอักษร (น้อยกว่าได้) จะต้องเป็นตัวพิมพ์ใหญ่เท่านั้น

ตัวอักษรที่ไว้ขึ้นต้น มีความหมายดังนี้

T หมายถึง อุปกรณ์ที่เกี่ยวข้องกับระบบ high volt เช่น หม้อแปลงไฟฟ้า

B หมายถึง อุปกรณ์ที่เกี่ยวข้องกับ Panel Board เช่น Breaker, Load Center, Consumer

L หมายถึง อุปกรณ์จำพวกหลอดไฟ เช่น หลอด Incandescent, Fluorescent, หลอด HID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S หมายถึง อุปกรณ์จำพวกสวิทช์และปลั๊ก เช่น สวิทช์ 1 ทาง , สวิทช์ 3 ทาง

W หมายถึง อุปกรณ์จำพวกสายไฟ เช่น สาย NYY , VAF , THW

C หมายถึง อุปกรณ์จำพวกท่อ เช่น ท่อ IMC, EMT, RSC

รายละเอียด - เป็นรายละเอียดต่างๆ เกี่ยวกับตัวอุปกรณ์ จะป้อนเป็นภาษาไทยหรือภาษาอังกฤษก็ได้ แต่มีความยาวได้ไม่เกิน 35 ตัวอักษร (ข้อแนะนำควรป้อนให้สั้นที่สุด แต่มีใจความมากที่สุด)

จำนวน - เป็นจำนวนของอุปกรณ์ที่เพิ่มลงไปใหม่ เป็น 1 เสมอ เพราะใช้เป็นข้อมูลอ้างอิง ต้องเป็น 1 เสมอ

หน่วย - เป็นหน่วยของตัวอุปกรณ์ เช่น EA, SET, LOT, M

ราคา/หน่วย - เป็นราคาของอุปกรณ์ต่อหน่วย มีทศนิยมได้ 2 ตำแหน่ง

วันที่ - เป็นวันที่ที่ เพิ่มอุปกรณ์ เพื่อใช้อ้างอิงในการเปลี่ยนแปลงราคาในครั้งต่อไป มีรูปแบบเป็น วท - คค - คศ

โปรแกรม สำหรับป้อนข้อมูลใหม่

โค้ด	
รายละเอียด	
จำนวน	
หน่วย	
ราคา/หน่วย	
วันที่	

กท ESC = จบการป้อน

รูปที่ ๗.2 แสดงการเพิ่มข้อมูลใหม่

เมื่อป้อนข้อมูลจนครบทุกหัวข้อแล้ว ให้กด ENTER จะมีข้อความว่า " ต้องการจับเก็บข้อมูลหรือไม่ (Y/N): Y " คำที่ด้งไว้เป็น Y หรือ Yes

ถ้าต้องการจับเก็บให้ตอบ Y แต่ถ้าไม่ต้องการจับเก็บให้ตอบ N และถ้าตอบอย่างอื่นนอกเหนือจาก Y,N จะมีข้อความเตือนว่า " ป้อนค่าผิด " ให้ทำการตอบใหม่ให้ถูกต้อง และเมื่อต้องการจบการทำงาน

หรือจบการเพิ่มอุปกรณ์ใหม่ให้กลายเป็น ESC ข้อมูลที่เพิ่มเข้าไปใหม่จะถูกจัดเก็บลงในแฟ้มข้อมูลชื่อ Filedata.dbf ทันที

ถ้าเลือกเมนูย่อยหมายเลข 2 คือ เปลี่ยนแปลงราคาอุปกรณ์ จะมีเมนูย่อยให้เลือกอีก 3 รายการ คือ

1. แก้ไขราคาอุปกรณ์ทั้งหมด
2. แก้ไขราคาอุปกรณ์บางรายการ
3. จบการทำงาน

ถ้าเลือกรายการแก้ไขราคาอุปกรณ์ทั้งหมด หมายถึง ต้องการเปลี่ยนแปลงราคาอุปกรณ์ทุกรายการในแฟ้มฐานข้อมูล ถ้าเลือกรายการนี้จะมีรายละเอียดของอุปกรณ์ทุกรายการ เฉพาะเขตข้อมูลที่สำคัญแสดงให้ทราบ แต่สามารถแก้ไขได้เฉพาะราคาของอุปกรณ์เท่านั้น ในเขตข้อมูลอื่นๆ ไม่สามารถแก้ไขได้ เมื่อแก้ไขเรียบร้อยแล้วก็จะเปลี่ยนวันที่ให้เป็นวันที่ปัจจุบันโดยอัตโนมัติ เพื่อใช้เป็นข้อมูลอ้างอิงเพื่อเปลี่ยนแปลงราคาอุปกรณ์ในครั้งต่อไป เมื่อจบการแก้ไขให้กดแป้น ESC เพื่อกลับมายังเมนูเลือกการแก้ไขราคา

แต่ถ้าต้องการแก้ไขราคาอุปกรณ์บางตัว หรือบางรายการให้เลือกเมนูหมายเลข 2 คือ แก้ไขราคาบางรายการ จากนั้นก็จะมีข้อความให้ป้อนโค้ดของอุปกรณ์ที่ต้องการแก้ไขราคา ให้พิมพ์โค้ดของอุปกรณ์ที่ต้องการแก้ไขราคาลงไป และกด Enter ดังรูปที่ ผ ข. 3

โปรแกรม เปลี่ยนแปลงราคาอุปกรณ์บางรายการ

โค้ด :
รายละเอียด :
หน่วย :
ราคา/หน่วย :
วันที่ป้อนข้อมูล :

กรุณาป้อนโค้ดที่ต้องการแก้ไขราคา : _____

กด ESC = จบการทำงาน

รูปที่ ผ ข.3 แสดงการเปลี่ยนแปลงราคาอุปกรณ์

จากนั้นก็จะแสดงรายละเอียดของอุปกรณ์ตัวนั้นให้ทราบ และรอการป้อนราคาใหม่ และวันที่ทำการแก้ไข เมื่อกด Enter ก็จะมีข้อความว่า " ต้องการเปลี่ยนราคาอุปกรณ์นี้จริง (Y/N) : Y "

ค่าที่คั่งไว้คือ Y หรือ Yes (ต้องการเปลี่ยนแปลงราคา)

ถ้าตอบอย่างอื่นนอกเหนือจาก Y ,N แล้ว จะมี ข้อความเตือนว่า " ป้อนค่าผิด "

ถ้าต้องการเปลี่ยนแปลงราคาอุปกรณ์ตัวนี้ ให้ตอบ Y

แต่ถ้าไม่ต้องการเปลี่ยนให้ตอบ N

จากนั้นก็จะถามว่า " ต้องการแก้ไขราคาอุปกรณ์อื่นอีกหรือไม่ (Y/N) : Y "

ถ้าต้องการแก้ไขราคาอุปกรณ์ตัวอื่นอีกให้ตอบ Y

แต่ถ้าไม่ต้องการแก้ไขให้ตอบ N

เมื่อจบการแก้ไขให้กดเป็น ESC

เมื่อแก้ไขราคาของอุปกรณ์ต่างๆ เสร็จเรียบร้อยแล้วให้เลือกเมนูหมายเลข 3 คือ จบการทำงาน
ถ้าเลือกเมนูย่อยหมายเลข 3 คือ ลบรายการอุปกรณ์ออกจะเป็นการลบรายละเอียดของอุปกรณ์ออกจากแฟ้ม
ฐานข้อมูล ในกรณีที่อุปกรณ์ตัวนั้นเลิกทำการผลิตแล้ว หรือไม่สามารถหาซื้อมาใช้ได้อีก เพื่อลบข้อมูลที่ไม่
ได้ี้ออกจากแฟ้มฐานข้อมูล จะมีข้อความต่างๆ ปรากฏบนหน้าจอภาพ ดังรูปที่ ผ ข.4 และรอให้ป้อนโค้ด
ของอุปกรณ์ที่ต้องการลบเมื่อป้อนโค้ดแล้วให้กด Enter

โปรแกรม ลบรายการฐานข้อมูล

โค้ด :

รายละเอียด :

หน่วย :

ราคา/หน่วย :

วันที่ป้อนข้อมูล :

กรุณาป้อนโค้ดที่ต้องการลบ : _____

กด ESC = จบการทำงาน

รูปที่ ผ ข.4 แสดงการลบข้อมูล

จะมีรายละเอียดต่างๆ ของอุปกรณ์ที่จะทำการลบแสดงให้ทราบ และจะมีข้อความว่า

" ต้องการลบอุปกรณ์นี้จริง (Y/N) : N "

ค่าที่คั่งไว้คือ N (ไม่ต้องการลบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่อนุญาตให้เผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต และจะถือว่าผิดกฎหมายหากฝ่าฝืน

ถ้าต้องการลบรายละเอียดของอุปกรณ์จริงให้ตอบ Y

แต่ถ้าไม่ต้องการลบให้ตอบ N

และถ้าตอบอย่างอื่นนอกเหนือจาก Y,N จะมีข้อความเตือนว่า " ป้อนค่าผิด "

ในกรณีที่ตอบ Y หรือ N แล้วก็จะมีข้อความถามว่า

" ต้องการลบข้อมูลต่อไปหรือไม่ (Y/N) : Y "

ค่าที่คั่นไว้คือ Y (ต้องการลบข้อมูลต่อ) ถ้าต้องการลบข้อมูลอีกให้ตอบ Y

แต่ถ้าไม่ต้องการลบข้อมูลต่อให้ตอบ N

ในกรณีที่ป้อนโค้ดของอุปกรณ์ที่ไม่มีในพื้นฐานข้อมูล หรือป้อนชื่อผิด ก็จะมีข้อความบอกให้ทราบว่า " ไม่พบเรคคอร์ดที่ต้องการลบ " ให้กดคีย์ใดๆ เพื่อที่จะทำงานต่อไป

เมื่อจบการทำงานให้กดแป้น ESC

7. เกี่ยวกับโปรแกรม

รายการนี้เป็นรายการที่นอกเกี่ยวกับ ผู้เขียนโปรแกรม, เครื่องคอมพิวเตอร์และส่วนประกอบที่จำเป็น
ต้องมีการใช้โปรแกรมนี้ และนอกเกี่ยวกับการใช้โปรแกรมอย่างคร่าวๆ

8. ออกจากโปรแกรม.....สู่ DOS

เป็นหัวข้อที่เลือกสำหรับจบการทำงานในการประมาณราคาเพื่อออกสู่ DOS

ภาคผนวก ก.

การติดตั้งโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม

โปรแกรมการเขียนแบบระบบไฟฟ้าและประมาณราคาคำนวณด้วยคอมพิวเตอร์นั้น บรรจุอยู่ในแผ่น High Density ขนาด 3.5" จำนวน 14 แผ่น ประกอบด้วย Software Auto CAD Release 12 จำนวน 7 แผ่น และ Software FoxPro Version 2.5 จำนวน 6 แผ่น และเป็นฐานข้อมูลกับโปรแกรมที่เขียนขึ้นอีก 1 แผ่น (โปรแกรมนี้ใช้เนื้อที่ HardDisk ประมาณ 44 Mb เพราะฉะนั้นควรมีเนื้อที่ HardDisk วางประมาณ 50 Mb)

วิธีการติดตั้ง

1. ติดตั้ง SOFTWARE AUTO CAD

ใส่แผ่นที่ 1 ในเครื่องขับ (drive) พิมพ์คำว่า " install1 " แล้วกด Enter จากนั้นจะมีข้อความบอกว่าให้ใส่แผ่นสุดท้ายเข้าไป (แผ่นที่ 7) แล้วกด Enter จากนั้นจะมีข้อความบอกว่าให้ใส่แผ่นที่ 1 เข้าไปใหม่เมื่อใส่แผ่นที่ 1 แล้วให้กด Enter โปรแกรมจะทำการสร้าง Sub - directory และทำการขยายไฟล์ในแผ่นลงสู่ Hard Disk Drive C และเมื่อขยายไฟล์ในแผ่นที่ 1 หหมดแล้วจะถามหาแผ่นที่ 2 ให้ใส่แผ่นที่ 2 แทนแผ่นที่ 1 แล้วกด Enter และเมื่อโปรแกรมขยายไฟล์ในแผ่นที่ 2 หหมดแล้วก็จะถามหาแผ่นที่ 3 ให้ทำตามขั้นตอนเหมือนเดิมจนครบทั้งหมด 7 แผ่น เมื่อครบ 7 แผ่นแล้วโปรแกรมก็จะถามหาแผ่นที่ 1 ให้ใส่แผ่นที่ 1 อีกครั้ง แล้วกด Enter ก็เป็นอันว่า Install Software Auto CAD เสร็จเรียบร้อยแล้ว

2. ติดตั้ง SOFTWARE FOXPRO

การติดตั้งก็เหมือนกับการติดตั้ง Auto CAD คือให้ใส่แผ่นที่ 1 ในเครื่องขับ พิมพ์คำว่า " Install2 " แล้วกด Enter จากนั้นให้ทำเหมือนการติดตั้ง Software Auto CAD (Software FoxPro มีทั้งหมด 6 แผ่น)

3. การติดตั้งฐานข้อมูลและโปรแกรมที่เขียนขึ้น

ให้ใส่แผ่นที่ชื่อว่า "Database and Program" ในเครื่องขับ จากนั้นให้พิมพ์คำว่า " Install3 " แล้วกด Enter โปรแกรมจะสร้าง Sub - directory และทำการขยายไฟล์จากแผ่นลงสู่ Hard Disk Drive C และเมื่อทำการขยายไฟล์เสร็จแล้ว ก็เสร็จสิ้นการติดตั้งโปรแกรมการเขียนแบบระบบไฟฟ้าและประมาณราคาคำนวณด้วยคอมพิวเตอร์

หมายเหตุ ให้เพิ่มคำสั่งต่อไปนี้ในไฟล์ Autoexec.bat

```
SET ACAD = C:\ACAD\12\SUPPORT; C:\ACAD\12\FONTS; C:\ACAD\12\ADS; C:\ACAD\12\MNU;  
SET ACADCFG = C:\ACAD\12  
SET ACADDRV = C:\ACAD\12\DRV
```

กิจกรรมประกาศ

ผู้จัดทำโครงการขอขอบคุณ อาจารย์รศ. ศุภี บรรจงจิตร และอาจารย์ สมชาติ จิรวិภากร ที่เป็นอาจารย์ที่ปรึกษาโครงการนี้ ซึ่งคอยให้คำแนะนำชี้แนะต่างๆ ซึ่งเป็นประโยชน์ในการทำโครงการนี้มาก, ขอขอบคุณอาจารย์ประเสริฐ ภูมิศิริโช อาจารย์สอนคอมพิวเตอร์ สำนักงานวิจัยและพัฒนาคอมพิวเตอร์ ที่ให้คำแนะนำเกี่ยวกับการแสดงผลทางเครื่องพิมพ์และขอขอบคุณ บริษัท TOSHIBA ประเทศไทย จำกัด ที่กรุณาให้ข้อมูลเกี่ยวกับหลอดไฟต่างๆ

ผู้ร่วมทำโครงการหวังเป็นอย่างยิ่งว่าโครงการนี้มีประโยชน์ต่อท่านผู้ศึกษาทางด้านนี้ ความดีที่ได้จากโครงการนี้ขอมอบให้ อาจารย์รศ.ศุภี บรรจงจิตร และอาจารย์ สมชาติ จิรวิภากร ได้ให้คำแนะและคำปรึกษา ตลอดมา



บรรณานุกรม

- พิบูลย์ คิมธุคม, " การออกแบบระบบแสงสว่าง, " บริษัท ชีเอคยูเคชั่น จำกัด, 2535
- ควง บงกชเกตุตถกุล, " ระบบพัฒนาฐานข้อมูลFoxPro, " บริษัท ชีเอคยูเคชั่น จำกัด, 2535
- ควง บงกชเกตุตถกุล, " คู่มือการใช้คำสั่งและฟังก์FoxPro, " บริษัท ชีเอคยูเคชั่น จำกัด, 2537
- ศิริชัย ขนบศีลธิมรุ่ง, " เทคนิคการใช้ FoxPro 2.5, " บริษัท สกายบุ๊กส์ จำกัด , 2536
- วรรณ พันธ์พิริยะ, " คู่มือการใช้โปรแกรมจัดการฐานข้อมูล FoxPro 2, " บริษัท อินฟอร์มติก บิซิเนสส์ จำกัด, " 2536
- นิพนธ์ กิตติปภัตสร, " โปรแกรมคอมพิวเตอร์ Clopper 5, " บริษัท ชีเอคยูเคชั่น จำกัด, 2537
- ชัยวุฒิ จันมา, " แอควานซ์ ดีเบต, FoxBase, Clipper, " สยามคอมพิวเตอร์, 2532
- อิน ภู่วรรณ, " แอควานซ์ดีเบต รวบรวมเทคนิคการประยุกต์ใช้โปรแกรมดีเบตวิฟัส, " บริษัท ชีเอคยูเคชั่น จำกัด, 2532
- พ.ศ.พัฒน์ อุกโสภา, " คู่มือการใช้โปรแกรม AutoCAD, " บริษัท ชีเอคยูเคชั่น จำกัด, 2535
- พันธุ์ธ หัตถโกศล, " คู่มือการใช้งานโปรแกรม AutoCAD Release 12, " โอ. เอส. พริ้นติ้ง เฮ้าส์, กรุงเทพฯ 2536
- รองศาสตราจารย์มณฑา ปราการสมุทร, " การเขียนชุดคำสั่งภาษา C, " ไชยทศพรินติ้ง จำกัด , 2535