ห้องสมุดคณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2533

การแสดงผลสภาวะของกระบวนการอุตสาหกรรม

(PROCESS MONITORING)

โดย

นาย สถิต วีรีรัตน์ 32.6129

อาจารย์ที่ปรึกษา

อาจารย์ ภาคร พุฒะสังกาศ

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

027924
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
18 ก.ค. 2534

ปริญญานิพนธ์ปีการศึกษา 2533

เรื่อง การแสดงผลสภาวะของกระบวนการอุตสาหกรรม
(PROCESS MONITORING)

ผู้จัดทำ

นาย สิทธิ วิริยะพันธ์ 32.6129

..... อาจารย์ที่ปรึกษา
(อาจารย์ ภาคร พุฒะสังกาศ)

..... อาจารย์ที่ปรึกษา
( )

..... อาจารย์ที่ปรึกษา
( )

..... อาจารย์ที่ปรึกษา
( )

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เลขหมู่ T 33091 ส 3
เลขทะเบียน 027924
วัน, เดือน, ปี 9 ส.ค. 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
027924

หัวข้อปริญญานิพนธ์	การแสดงผลสภาวะของกระบวนการอุตสาหกรรม
นักศึกษา	นาย สถิต วีรัชต์ 32.6129
อาจารย์ที่ปรึกษา	อ.ภาคร พุทธสังกาศ
ปีการศึกษา	2533

บทคัดย่อ

ปัจจุบันความก้าวหน้าด้านเทคโนโลยีของคอมพิวเตอร์ ได้ถูกนำมาประยุกต์ใช้ในกระบวนการทางอุตสาหกรรมเป็นอย่างมาก ไม่ว่าจะเป็นด้านการวัด หรือบันทึกตลอดจนแปรการควบคุม รวมถึงการแสดงผลทางด้านกราฟฟิค คอมพิวเตอร์สามารถนำมาใช้โดยการส่งข้อมูลสภาวะของกระบวนการ ผ่านตัวกลางชนิดต่างๆ ส่งให้คอมพิวเตอร์ ซึ่งคอมพิวเตอร์จะนำข้อมูลเหล่านั้นมาประมวลผลแล้วนำไปแสดงทางจอแสดงผล

ปริญญานิพนธ์ฉบับนี้ ขอเสนอการแสดงผลสภาวะของกระบวนการอุตสาหกรรมทางจอแสดงผล โดยใช้คอมพิวเตอร์เป็นตัวประมวลผลข้อมูลต่างๆ แล้วแสดงผลทางจอแสดงผล โดยการเปลี่ยนสี และการเปลี่ยนรูปของอุปกรณ์ในกระบวนการอุตสาหกรรม ซึ่งปริญญานิพนธ์ฉบับนี้จะมีตัวอย่างการสร้างภาพกระบวนการ และการกำหนดเงื่อนไขต่างๆ ของโปรแกรมแสดงผล โดยผู้ใช้ หรือผู้ควบคุมสามารถจะกำหนดเงื่อนไขต่างๆได้โดยง่ายด้วยตัวเอง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title PROCESS MONITORING
Name SATHIT WEEREERAT
Thesis Advisor PHAKORN HUTASANGKAS
Academic Year 1990

ABSTRACT

The fast develop of computer technology has been applying to most industrial processes nowadays, such as in measurement and control, device and system, as well as graphic display . Computer using to received variable value and parameter from process to convert it appropriate to display on monitoring.

This thesis presents the method of graphic display design in process control by using computer. Graphic display can be change the color and specific device dependent on the process value. In this thesis has a sample graphic display picture and allocation the condition of process value for display. The operator can used it easily in industrial process control.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎี และการใช้งานโปรแกรม AUTOCAD

2.1 การเตรียมจอภาพสำหรับแสดงกราฟฟิค 2
2.2 การเริ่มต้นใช้โปรแกรม AUTOCAD 5
2.3 DRAWING EDITER 9
2.4 คำสั่ง AUTOCAD และการกำหนดค่า 11

บทที่ 3 การสร้างภาพกระบวนการ และ การกำหนดอุปกรณ์ในกระบวนการ

3.1 การสร้างภาพกระบวนการ 32
3.2 การแบ่งชั้นของภาพกระบวนการ 32
3.3 การสร้างภาพลงในคอมพิวเตอร์ 36
3.4 การเก็บภาพลงแผ่นจานแม่เหล็ก 49
3.5 รูปแบบ และโครงสร้างของ DRAWING INTERCHANGE FILE 49

บทที่ 4 โปรแกรมเงื่อนไขการแสดงผล

4.1 การกำหนดช่วงเวลาในการทำงานของแต่ละงาน 69
4.2 การกำหนดสีของอุปกรณ์ในสภาวะต่างๆ 72
4.3 การกำหนดเงื่อนไขการแสดงผล 73
4.4 ตัวอย่างการเปลี่ยนภาพ AUTOCAD เป็นภาพ TURBO PASCAL 79

บทที่ 5 บทสรุป 90
กิตติกรรมประกาศ 92
เอกสารอ้างอิง 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันการติดตามการเปลี่ยนแปลงสภาวะของกระบวนการทางอุตสาหกรรมนับว่ามีความสำคัญอย่างมากสำหรับการควบคุมกระบวนการ และการควบคุมคุณภาพของผลิตภัณฑ์ ซึ่งมีอยู่หลายวิธี แต่ละวิธีก็มีข้อดีข้อเสียแตกต่างกันออกไปวิธีหนึ่งที่ได้ผลดีคือ การนำเอาคอมพิวเตอร์มาประยุกต์ใช้ในการติดตามการเปลี่ยนแปลงดังกล่าว วิธีการคือการส่งข้อมูลสภาวะกระบวนการผ่านตัวกลางชนิดต่างๆ มาให้กับคอมพิวเตอร์ ซึ่งคอมพิวเตอร์จะนำข้อมูลเหล่านั้น มาประมวลผลแล้วนำไปแสดงผลทางจอแสดงผล โดยอาจจะใช้วิธีเปลี่ยนสี เปลี่ยนข้อความ หรือเปลี่ยนรูป เป็นต้น.

ปริญญานิพนธ์ฉบับนี้ขอเสนอ การแสดงผลกราฟฟิก ของสภาวะกระบวนการทางอุตสาหกรรมโดยแสดงผลทางจอมอนิเตอร์(Monitor) ของไมโครคอมพิวเตอร์ IBM PERSONAL COMPUTERM ซึ่งจะเป็นการแสดงผลกราฟฟิก ของสภาวะของกระบวนการโดยมีการเปลี่ยนสีของอุปกรณ์ในระบบอุตสาหกรรม เช่น ปั๊ม วาล์ว และท่อ รวมถึง การเปลี่ยนระดับของเหลวในถัง(Tank) โดยในปริญญานิพนธ์ฉบับนี้จะมีเฉพาะส่วนของซอฟร์แวร์(Software)เท่านั้น ส่วนการออกแบบฮาร์ดแวร์(Hardware)จะไม่กล่าวถึงในปริญญานิพนธ์ฉบับนี้ แต่เราสามารถนำส่วนของฮาร์ดแวร์มาใช้ร่วมกันได้ แต่จะต้องเพิ่มส่วนของซอฟร์แวร์ขึ้นมาอีกบางส่วน ซึ่งสามารถทำได้โดยง่าย โดยฮาร์ดแวร์นี้จะเป็นตัวรับสภาวะจากกระบวนการอุตสาหกรรม ซึ่งในท้องตลาดก็มีจำหน่ายอยู่หลายชนิด ขึ้นอยู่กับชนิดของกระบวนการ ประเภทการใช้งาน และรูปแบบที่เชื่อมต่อกับกระบวนการ ซึ่งมีทั้งแบบดิจิตอล(Digital) และแบบอนาลอก(Analog)ตัวอย่างเช่นในแบบดิจิตอลค่าแรงดันที่รับเข้ามาจะต้องอยู่ในช่วง 3v ถึง 25v สำหรับสภาวะ"เปิด"(ON) อยู่ในช่วง 0 ถึง 3v สำหรับสภาวะ"ปิด"(OFF) และฮาร์ดแวร์ที่กล่าวมานี้ จะแยกต่างหากจากฮาร์ดแวร์ของเครื่อง IBM เมื่อไม่มีฮาร์ดแวร์ร่วมกัน ดังนั้น จึงใช้กับคอมพิวเตอร์ได้หลายชนิด.

ในบทต่อๆไปจะเป็นการกล่าวถึงการสร้างภาพกระบวนการ เพื่อใช้แสดงผล ซึ่งในการใช้งานจริงแล้ว ผู้ใช้สามารถออกแบบ และ กำหนดเงื่อนไขในการแสดงผลได้เอง โดยจะกล่าวถึงการสร้างภาพโดยละเอียด ในที่นี้ได้ใช้โปรแกรมสำเร็จรูป ซึ่งมีจำหน่ายในท้องตลาด คือ AutoCAD ในการสร้างภาพกระบวนการทางอุตสาหกรรม แล้วจะเป็นการเก็บภาพที่สร้างนั้นลงแผ่นจานแม่เหล็ก และทำการเปลี่ยนจากไฟล์ภาพที่เก็บในแผ่นจานแม่เหล็กนั้นออกมาวาดใหม่ โดยใช้โปรแกรมภาษา เทอร์โบ ปาสคาล(Turbo Pascal) อีกครั้งหนึ่ง ซึ่งในส่วนของโปรแกรมเงื่อนไขการแสดงผล ของกระบวนการทางอุตสาหกรรม ก็จะใช้โปรแกรมภาษาเทอร์โบปาสคาล เช่นเดียวกัน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎี และ การใช้โปรแกรม AUTOCAD

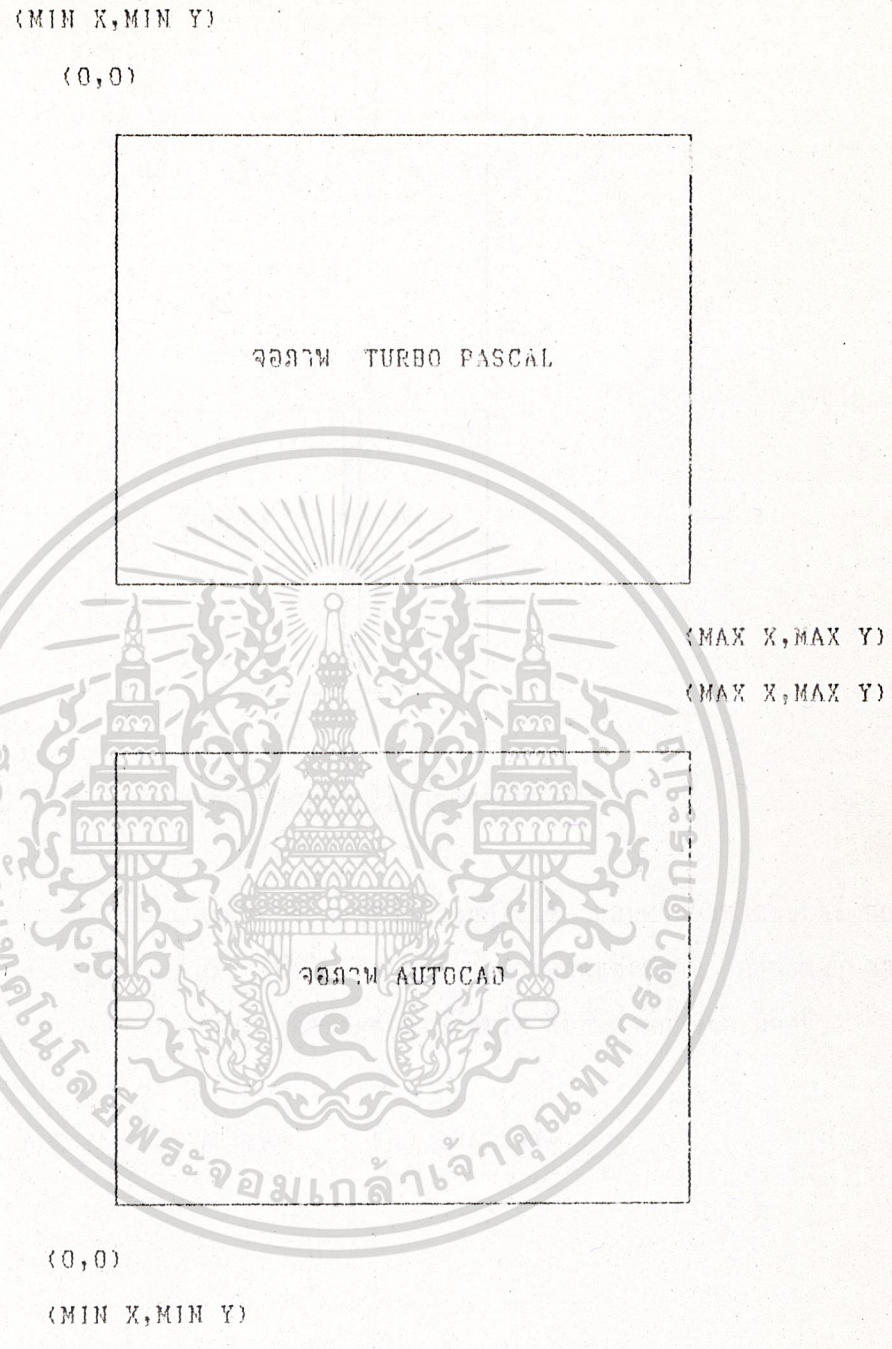
ทฤษฎี

การแสดงผลสภาวะของระบบทางจอมอนิเตอร์ เราใช้การแสดงภาพของระบบเป็นการแสดงแบบกราฟฟิก ซึ่งมีหลักการของการแสดงผลสภาวะของระบบ แบ่งออกได้เป็นสองส่วน คือ การสร้างภาพกระบวนการทั้งหมด และ ส่วนที่สองเป็นส่วนของโปรแกรมแสดงผล ซึ่งใช้โปรแกรมภาษา เทอร์โบ ปาสคาล (Turbo Pascal)

ในส่วนแรกจะเป็นการสร้างภาพของระบบที่จะใช้ในการแสดงผล โดยใช้โปรแกรม AutoCAD เป็นตัวสร้างภาพขึ้นมา ซึ่งเมื่อได้ภาพกระบวนการแล้วเราจะต้องเก็บภาพเป็นไฟล์ลงแผ่นจานแม่เหล็ก โดยใช้คำสั่งของโปรแกรม AutoCAD คือคำสั่ง DXFOUT เมื่อใช้คำสั่งนี้แล้วจะได้ไฟล์เป็นไฟล์มาตราฐานไฟล์หนึ่ง โดยไฟล์นี้จะบรรจุรายละเอียดเกี่ยวกับการสร้างภาพกระบวนการทั้งหมด ดังนั้น เราสามารถที่จะเปลี่ยนภาพที่วาดโดยโปรแกรม AutoCAD มาเป็นภาพกระบวนการที่วาดโดยเทอร์โบ ปาสคาลได้ ด้วยการอ่านรายละเอียดการวาดภาพของไฟล์ AutoCAD แล้วใช้โปรแกรมภาษา เทอร์โบ ปาสคาล สร้างภาพอีกครั้งหนึ่งโดยใช้รายละเอียดต่างๆที่ได้จากไฟล์ของ AutoCAD จึงเปรียบเสมือนกับว่า เราได้สร้างภาพกระบวนการด้วยโปรแกรมภาษา เทอร์โบ ปาสคาล เหตุผลที่ต้องใช้โปรแกรม AutoCAD วาดภาพกระบวนการในขั้นต้นก่อน คือการออกแบบโปรแกรมนี้ต้องการให้มีการยืดหยุ่นได้ผู้ใช้สามารถสร้างหรือกำหนดค่าต่างๆ ตามระบบที่ตนเองต้องการได้ ถ้าภาพกระบวนการเป็นภาพที่ซับซ้อนมากๆจะเป็นการไม่สะดวกในการสร้างภาพกระบวนการโดยใช้โปรแกรมภาษา เทอร์โบ ปาสคาล เราจึงใช้โปรแกรม AutoCAD ในการออกแบบ ซึ่งโปรแกรมนี้มีใช้โดยแพร่หลายมีจำหน่ายตามท้องตลาดและการใช้งานมีประสิทธิภาพสูงใช้งานได้ง่าย ที่สำคัญ สามารถใช้งานได้ดีในการออกแบบลักษณะงานเช่นนี้ เมื่อเราวาดภาพในกราฟฟิกโหมดของเทอร์โบปาสคาลได้แล้ว ในส่วนที่สองจะเป็นการออกแบบโปรแกรมเงื่อนไขในการแสดงผล ที่จะให้ภาพสภาวะของระบบ แสดงผลทางจอมอนิเตอร์ ที่กล่าวมาทั้งหมดนี้เป็นหลักการที่เราจะเสนอเป็นรายละเอียดในบทต่อๆไป

2.1 การเตรียมจอภาพสำหรับแสดงกราฟฟิก : ในตัวอย่างต่อไปนี้จะเป็นการแบ่งจอภาพออกเป็นสองส่วน คือ ส่วนแสดงภาพกราฟฟิก กับ ส่วนแสดงตัวอักษร ซึ่งในการเตรียมจอภาพนี้จะมีปัญหาอยู่ที่ว่า จอภาพที่วาดโดยโปรแกรม AUTOCAD กับจอภาพที่วาดโดย TURBO PASCAL จะมีจุด Coordinate ไม่ตรงกันในแนวแกนตั้ง ส่วนในแนวแกนนอนจะมีลักษณะเหมือนกัน ดังรูป.

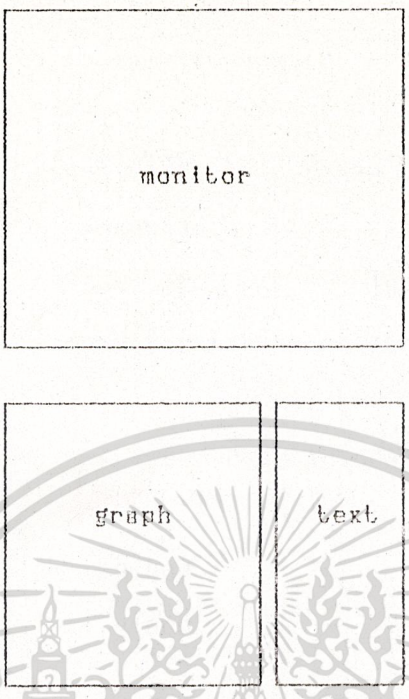
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 แสดงความแตกต่างของจอภาพ AUTOCAD กับ TURBO PASCAL

เนื่องจากสาเหตุนี้ เมื่อเราวาดภาพโดย AUTOCAD แล้ว เราจะนำภาพนั้นมาวาดในจอของ TURBO PASCAL จึงทำให้ค่า Coordinate ในแกนตั้งมีค่าไม่ตรงกัน เป็นเหตุให้ไม่ได้ภาพตามต้องการ ดังนั้น เราจึงทำการโอนย้ายจุด(Transfer Coordinate)ในแกนตั้งให้ตรงกัน โดยใช้ค่าคงที่มาใช้ในการโอนย้ายจุด ซึ่งจะมีการคำนวณค่าต่างๆขึ้นมาหลายๆค่า ซึ่งได้มาจากการเตรียมจอภาพ ดังนี้.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 แสดงการแบ่งจอภาพในการแสดงกราฟิก

เมื่อให้ได้จอภาพที่ต้องการเราจะกำหนดค่าต่างๆที่จะใช้ และค่าที่กำหนดขึ้นมานี้ จะใช้ในการแบ่งจอภาพสำหรับจอ EGA เท่านั้น เพราะว่าค่าที่กำหนดเหล่านี้ คำนวณจากจอ EGA เมื่อใช้กับจอภาพแบบอื่นๆ ค่าต่างๆ เหล่านี้จะต้องเปลี่ยนไป ในที่นี้ค่าที่กำหนด มีดังนี้.

```
Aspectratio = 1.5724138   { For EGA Only }
Maxport X   = 31000
Maxport Y   = 31000
```

เมื่อกำหนดให้ค่า Factor ที่จะใช้ในการโอนย้ายจุด เป็นดังนี้

KxFactor คือ ค่า Factor สำหรับแกน X

KyFactor คือ ค่า Factor สำหรับแกน Y

เมื่อคำนวณได้จาก

```
KxFactor = GetMax X / MaxportX / Aspectratio
KyFactor = - GetMax Y / Maxport Y
```

เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GetMax X คือ ค่าสูงสุดในแนวแกนตั้ง (X) = 550 (EGA)

GetMax Y คือ ค่าสูงสุดในแนวแกนนอน (Y) = 640 (EGA)

สำหรับค่าที่ได้นี้ เมื่อนำไปใช้ในการโอนย้ายจุด จากจอภาพ AUTOCAD มาแสดงในจอภาพ TURBO PASCAL ในแนวแกนนอน ค่า Coordinate ที่ได้จากจอภาพ AUTOCAD จะต้องคูณด้วยค่าคงที่ทางแกนนอน คือค่า KxFactor ส่วนในแนวแกนตั้ง จะต้องคูณด้วยค่าคงที่ทางแกนตั้งคือ KyFactor แล้วจะต้องนำมาบวกด้วยค่าสูงสุดทางแกนตั้ง คือค่า GetMax Y ดังในตัวอย่างการคำนวณ เช่น

เมื่อสมมติให้ Coordinate ของจอภาพ AUTOCAD มีดังนี้

ค่าทางแกนนอน (X) = 31000

ค่าทางแกนตั้ง (Y) = 31000

เมื่อทำการโอนย้ายจุด Coordinate มาแสดงบนจอภาพ TURBO PASCAL จะมีค่าดังนี้

X = 31000 * KxFactor = 31000 * 0.013129598 = 407

Y = (31000 * KyFactor) + GetMax Y

= (31000 * - 0.01129032) + 350

= 0

นั่นคือ เมื่อทำการโอนย้ายจุดบนจอภาพ AUTOCAD ที่Coordinate(31000,31000) มาเป็น Coordinate บนจอภาพ TURBO PASCAL จะได้ Coordinate เป็น(407,0) ซึ่งตรงกับจอภาพของส่วนบนสุดของจอภาพแสดงกราฟฟิก และค่าที่กล่าวมาทั้งหมดนี้ ผู้ใช้สามารถกำหนดหรือ เปลี่ยนแปลงได้ตามเหมาะสม และตามต้องการ

2.2 การเริ่มต้นใช้โปรแกรม AUTOCAD : จากการทดลองใช้โปรแกรม AUTOCAD โดยใช้ฟลอปปี้ดิสค์ จะต้องใช้แผ่นดิสค์บรรจุไฟล์ต่างๆของ AUTOCAD ดังนี้

แผ่นที่ 1 มีชื่อว่า Execution Disk

แผ่นที่ 2 มีชื่อว่า Overlay Disk

แผ่นที่ 3 มีชื่อว่า Drivers Disk

แผ่นที่ 4 มีชื่อว่า Samping Drawing Disk

Execution Disk บรรจุไฟล์ ACAD.EXE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Overlay Disk บรรจุไฟล์ .OVL

Drivers Disk บรรจุไฟล์ .DRV

Samping Drawing Disk บรรจุไฟล์ภาพที่วาดโดย AUTOCAD คือไฟล์ .DWG

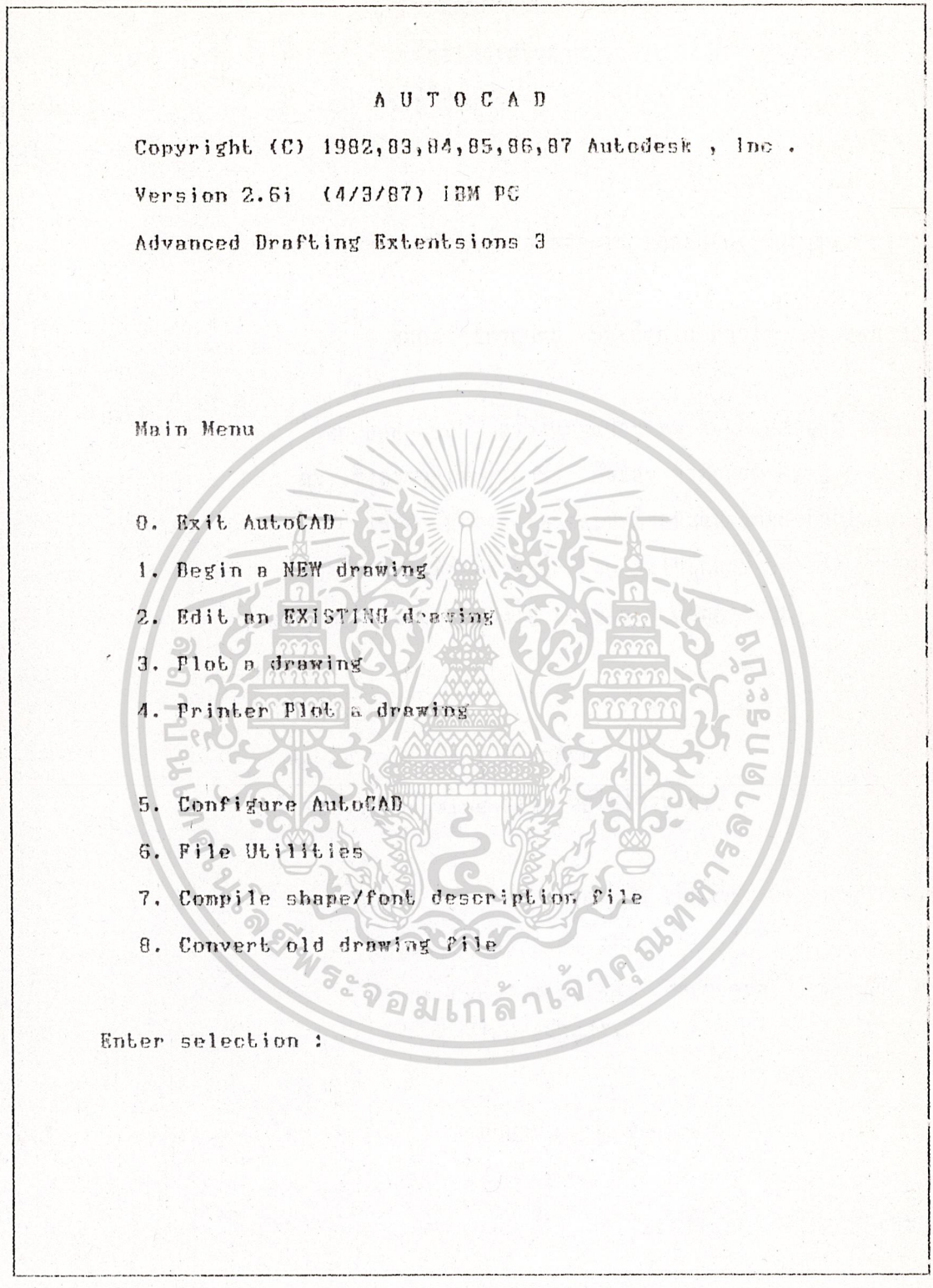
2.2.1 เริ่มต้นใช้งาน : ในการเริ่มต้นใช้งานมีลำดับขั้นตอน ดังต่อไปนี้

1. ใส่แผ่นดอส(Dos) เข้าไปในเครื่องขับ A: เปิดสวิช์ทเครื่อง ตอบวัน และ เวลา
2. เมื่อบนจอภาพปรากฏ A> ถอดแผ่นดอสออก เอาแผ่น Execution Diskใส่แทน
3. นำแผ่น Sample Drawimg Disk ใส่ในเครื่องขับ B:
4. พิมพ์ A > ACAD แล้วกด ENTER
5. เครื่องขับหมุนสักครู่ เมื่อหยุด บนจอภาพจะปรากฏดังนี้

```
Cant'n find OVERLAY file ACAD.OVL
Enter new drive letter or '.' to quit :
```

6. ถอดแผ่น Execution Disk ในเครื่องขับ A: ออก เอาแผ่นชื่อ Overlay Disk ใส่แทน พิมพ์ A: แล้วกด ENTER บนจอภาพจะปรากฏเมนู ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.3 แสดงเมนูของ AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้เราจะกล่าวถึงรายละเอียดของเมนูเฉพาะเมนูหมายเลข 0,1 และ 2 เท่านั้นเพื่อความรวดเร็วจะไม่บอกรายละเอียดของเมนูหมายเลขอื่นๆ ซึ่งผู้ใช้สามารถค้นหารายละเอียดได้จากหนังสือ " คู่มือการใช้โปรแกรม AUTOCAD " ชื่อผู้แต่ง " พ.ต.ประพัฒน์ อุทโยกาศ " ของบริษัท ซีเอ็ดยูเคชั่น จำกัด

2.2.2 เมนู : เรามาดูว่าเมนูแต่ละอันคืออะไร เมื่อเลือกเมนู 0,1 และ2 จะเกิดอะไรขึ้น

0. Exit AutoCAD - ออกจากโปรแกรม เมื่อเลือก 0 การใช้ AUTOCAD สิ้นสุดลง และกลับมาที่ Operating system

1. Begin a NEW Drawing - สร้างแบบแปลนใหม่ Autocad จะถามชื่อแบบแปลนจะต้องยาวไม่เกิน 8 ตัว ซึ่งอาจจะเป็นตัวอักษร ตัวเลข ตัวอักษรพิเศษคือ $ และ _ หลังจากพิมพ์ชื่อของแบบแปลนแล้ว ให้กด ENTER ชื่อนี้จะเป็นชื่อแบบแปลนที่จะบันทึกไว้ในดิสค์ ชื่อเหล่านี้จะเป็นไฟล์แบบ ".DWG"โดยอัตโนมัติเอย่าพิมพ์ .DWG ลงไปว่าและชื่อจะบันทึกอยู่ในดิสค์แผ่นที่อยู่ในไดร์ที่กำลังใช้งานอยู่ นอกจากเราระบุลงไป (เช่น B:) ตัวอย่างเช่น

```
Enter selection : 1
Enter name of Drawing : B:HOUSE
(กรณีที่แผ่นที่จะบรรจุ Drawing File อยู่ในไดร์ฟ B:)
```

AutoCAD จะโหลดโปรแกรม Drawing Editer เพื่อให้เราสร้างภาพหากเราต้องการไฟล์ของเราอยู่ในไดร์ฟ B: ซึ่งตั้งใจให้เป็น Data File

ถ้าเรามีแบบแปลนต้นแบบ(Prototype Drawing) สมมติว่าชื่อ STANDARD อยู่ เราต้องการให้เอาแบบแปลนต้นแบบมาใช้กับไฟล์ใหม่เราพิมพ์ดังนี้

```
Enter NAME of Drawing : B:HOUSE = B:STANDARD
```

HOUSE คือไฟล์ใหม่ ส่วน STANDARD คือภาพต้นแบบที่มีชื่อว่า STANDARD และอยู่ในไดร์ฟ B:

ถ้าเรามีไฟล์เดิมชื่อ HOUSE อยู่ เครื่องจะถามดังนี้

```
Warning! a drawing with this name already exists.
Do you want to replace it with the new drawing? (N)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากด ENTER คือตอบ NO <N> เครื่องจะกลับไปที่ Main Menu และถาม File Name ใหม่ แต่ถ้าตอบ YES <Y> เครื่องจะทำการสร้างภาพ เมื่อจบจะบันทึกทับไฟล์เดิม

2. Edit an EXISTING drawing - ตรวจแก้แบบแปลนที่มีอยู่ เมื่อต้องการเปลี่ยนหรือเพิ่มเติมแบบแปลนที่มีอยู่ หรือเพียงแต่เรียกแบบแปลนที่มีอยู่มาดูบนจอภาพ เลือกเมนูหมายเลข 2 เครื่องจะถามดังนี้

```
Enter selection : 2
Enter NAME of drawing : B:HOUSE
```

ถ้าหากเราได้ให้ชื่อของภาพนี้มาก่อนแล้ว AUTOCAD จะถือว่าภาพนี้เป็น Default คือ"ที่ได้กำหนดไว้แล้ว" กรณีนี้เราเพียงกด ENTER เช่น

```
Enter NAME of drawing (default B:HOUSE) :
```

เมื่อกำหนดเช่นนี้แล้ว AUTOCAD จะโหลดไฟล์ภาพเก่าที่ชื่อ HOUSE.DWG ขึ้นมาแสดงบนจอภาพเพื่อใช้แก้ไขต่อไป

2.3 Drawing Editer : มาถึงตอนนี้เรามาเข้าไปสู่ Drawing Editer ซึ่งจะเข้าได้สองทางดังกล่าวมา คือ เลือกเมนูหมายเลข 1 เพื่อเขียนแบบแปลนใหม่ หรือเรียกเมนูหมายเลข 2 เพื่อเอาแบบแปลนที่เขียนไว้แล้วมาดู หรือเอามาตรวจแก้ เราเลือก 1 เมื่อกด ENTER จะถามชื่อไฟล์ ดังรูป 2.4

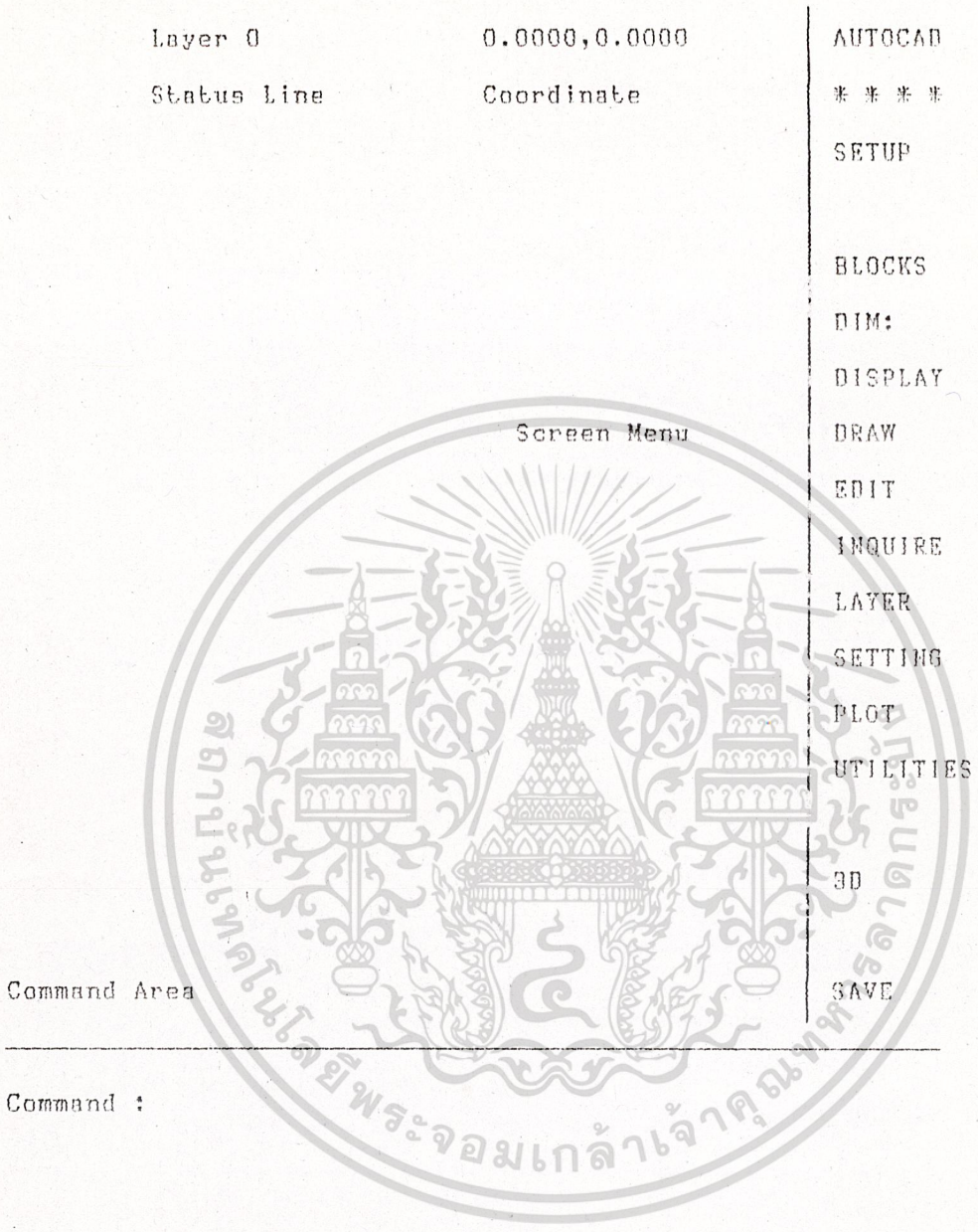
ให้เลือกเมนู 1 โดยตอบดังนี้

```
enter selection : 1
Enter NAME of Drawing : B:ABC     ( ให้ชื่อแบบแปลนเป็น ABC)
```

เมื่อใส่ชื่อแบบแปลนเสร็จกด ENTER เมื่อทุกอย่างเรียบร้อยจะปรากฏภาพออกทางจอภาพดังนี้

127924

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.4 แสดง Drawing Editer ของ AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.4 ผมจะอธิบายชื่อต่างๆดังนี้ บรรทัดบนสุดเราเรียกว่า Status line ซึ่งแบ่งเป็น 2 ส่วน ส่วนหน้าจะบอกว่าขณะนี้เรากำลังทำงานอยู่ในชั้น(Layer)ใด Layer เปรียบเสมือนกระดาษแก้วใสวางทับกันเป็นชั้นๆแต่ละชั้นสามารถกำหนดชื่อไว้ได้ เราสามารถเขียนรูปลงในชั้นไหนก็ได้ และสั่งให้แสดงรูปในชั้นใด หรือ ไม่แสดงรูปชั้นใดก็ได้ ในรูป 2.5 แสดงว่าขณะนี้เรากำลังเขียนรูปอยู่ในชั้นชื่อ 0 อยู่

ส่วนหลังของ Status line จะเป็นการแสดงจุด Coordinate ของเส้นสายใยซึ่งจะสั่งให้เป็นทศนิยมก็ตำแหน่งก็ได้ ทางด้านขวาของจอภาพเรียกว่า Screen Menu ซึ่งเป็นกลุ่มของคำสั่งซึ่งเราสามารถเรียกใช้คำสั่งใดก็ได้ ถ้าไม่มี Mouse ก็สามารถพิมพ์คำสั่งด้วยแป้นพิมพ์ทีละตัวอักษรถ้าใช้ Mouse เราสามารถเลื่อนแถบสีเคอร์เซอร์เลือกคำสั่งได้ บรรทัดล่างสุดเป็น Command Area คือที่ที่เราจะให้คำสั่งด้วยการใช้แป้นพิมพ์ พิมพ์คำสั่งลงไปตรงๆ

2.4 คำสั่ง AutoCAD และการกำหนดค่า : ในโปรแกรม AutoCAD คำสั่งที่ใช้มีมากมายซึ่งล้วนแต่เป็นคำสั่งที่อำนวยความสะดวกในการวาดภาพทั้งสิ้น ในส่วนนี้เราจะกล่าวถึงเฉพาะคำสั่งที่ได้ใช้ในการสร้างภาพกระบวนการในปริญญานิพนธ์ฉบับนี้เท่านั้น ซึ่งผมจะแบ่งเป็นส่วนๆดังต่อไปนี้

2.4.1 คำสั่งที่ใช้ในการเขียนส่วนของภาพ : ก่อนอื่นเรามารู้จักคำสั่งที่ใช้ในการวาดภาพกระบวนการก่อน ผมทราบแล้วเราจะสามารถวาดภาพได้สะดวกและรวดเร็ว คำสั่งในการเขียนส่วนของภาพ จะประกอบด้วยคำสั่งและรายละเอียดต่อไปนี้

คำสั่ง LINE : เป็นคำสั่งพื้นฐานในการลากเส้นในกรณีนี้เราจะต้องบอกจุดปลายทั้งสองของเส้น มีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : LINE
From point : 1,1
To point   : 5,2
To point   : (กดแป้นเว้นวรรค หรือ ENTER)
```

เมื่อใช้คำสั่ง LINE เครื่องจะถามบรรทัดต่อมา คือ

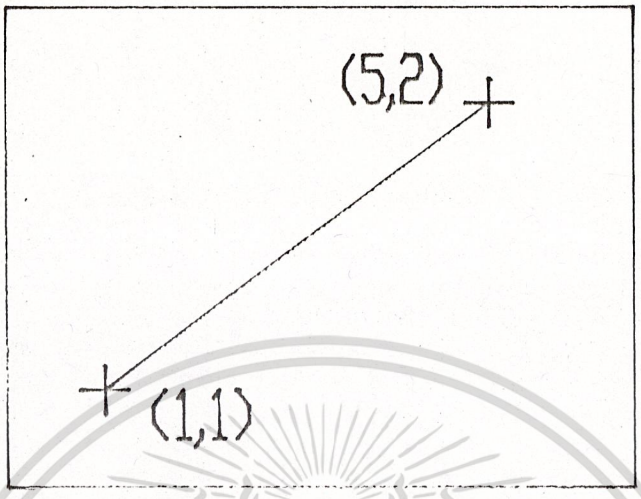
```
From point : 1,1          (จุดเริ่มต้น Coordinate 1,1)
```

เมื่อใส่จุดเริ่มต้น (1,1) เครื่องจะถามบรรทัดต่อมา คือ

```
To point : 5,2            (จุดปลาย Coordinate 5,2)
```

นั่นคือการใช้คำสั่ง LINE ให้ลากเส้นจากจุดที่มี Coordinate (1,1) ถึงจุดที่มี Coordinate (5,2)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 แสดงภาพที่วาดโดยคำสั่ง LINE

คำสั่ง CIRCLE : คือคำสั่งในการเขียนวงกลม จะมีอยู่ 4 แบบ คือ

1 กำหนดจุดศูนย์กลาง และรัศมีของวงกลม

2 กำหนดจุดศูนย์กลาง และเส้นผ่าศูนย์กลางของวงกลม

3 กำหนดจุด 3 จุดบนเส้นรอบวงของวงกลม

4 กำหนดจุด 2 จุดบนเส้นรอบวงของวงกลม

ในที่นี้จะกล่าวถึงการใช้แบบที่ 1 และกล่าวถึงรายละเอียดเฉพาะแบบที่ 1 เท่านั้น คือการกำหนดจุดศูนย์กลาง และรัศมีของวงกลม ซึ่งมีรูปแบบคำสั่งดังต่อไปนี้

```
Command : CIRCLE
3P/2P/<Center point> :
```

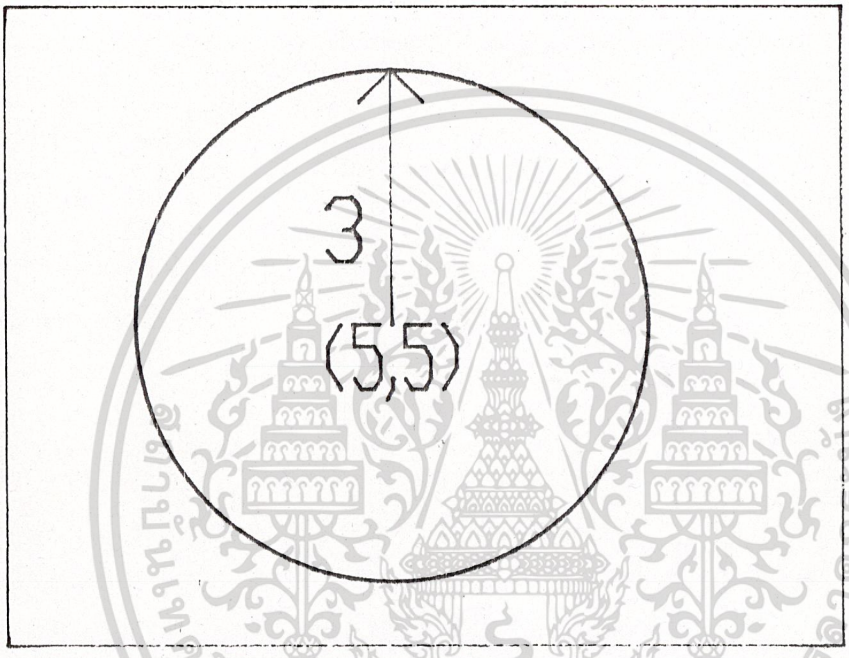
ถ้าเราให้ค่าลงไปด้วยการพิมพ์ หรือด้วยตัวชี้ก็ตาม ค่านั้นจะเป็นค่า Coordinate ของจุดศูนย์กลาง เช่น

```
Command : CIRCLE
3P/2P/<Center Point> : 5,5
Dimmeter/<Radius> :
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจะตั้งไว้ว่าถ้ากำหนดตัวเลขนั้นคือ รัศมี หรือถ้าเราเลื่อนตัวชี้ออกจากจุดศูนย์กลางตำแหน่งใหม่ของของตัวชี้จะอยู่บนเส้นรอบวงของวงกลมนั้น เมื่อใส่ค่ารัศมี จะได้

```
Diameter/<Radius> : 3
```



รูป 2.6 แสดงภาพวงกลมที่วาดโดยคำสั่ง CIRCLE

จากรูป 2.6 เป็นการเขียนวงกลมโดยมีจุด(5,5)เป็นจุดศูนย์กลางและเส้นรอบวงจะอยู่ห่างจากจุดศูนย์กลางไป 3 หน่วย ซึ่งเป็นค่ารัศมี นั่นเอง

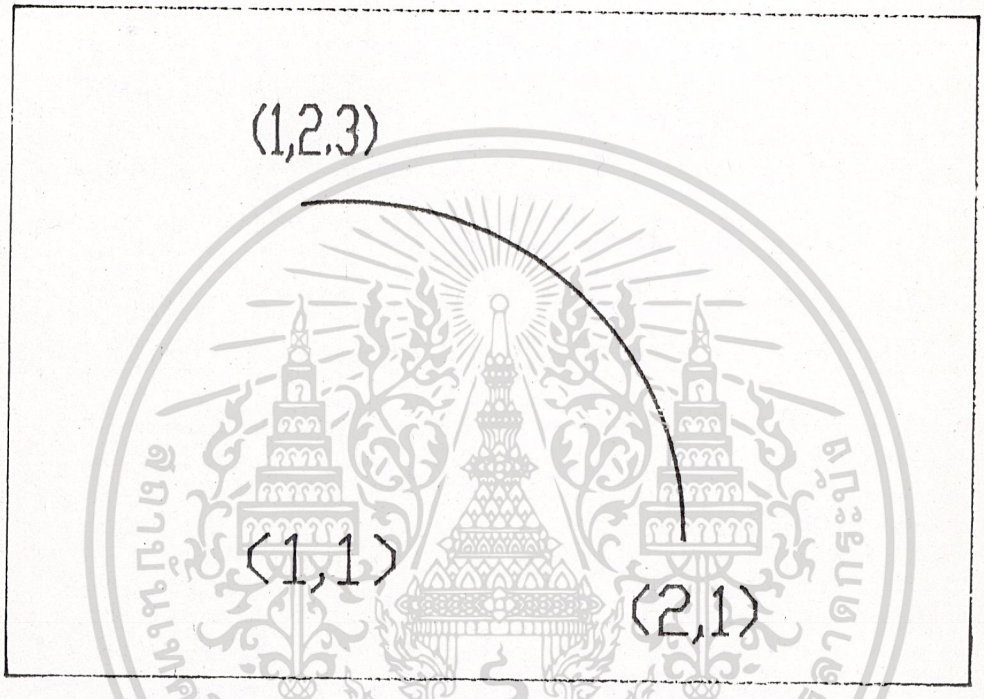
คำสั่ง ARC : คือคำสั่งเขียนส่วนโค้งของวงกลมซึ่งมีอยู่หลายแบบในที่นี้จะกล่าวถึงแบบกำหนด จุดตั้งต้น จุดศูนย์กลาง และจุดสิ้นสุดของส่วนโค้ง ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : ARC
Center/<Start Point> : 2,1                    (จุดเริ่มต้น)
Center/End/<Second Point> : C
Center : 1,1                                  (จุดศูนย์กลาง)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Angle/Lenght of Chord/<End Point> : 1,2.3 (จุดสิ้นสุด)

การเขียนจะเขียนจากจุดตั้งต้นทวนเข็มนาฬิกาไปยังจุดสุดท้าย เนื่องด้วยมีข้อจำกัดที่ว่ารัศมีของส่วนโค้งจะถือตามระยะจากจุดศูนย์กลางไปยังจุดเริ่มต้น ดังนั้นจุดสุดท้ายอาจจะไม่อยู่บนส่วนโค้งก็ได้ แต่ปลายจุดสุดท้ายของส่วนโค้งจะอยู่ในแนวเดียวกับจุดศูนย์กลาง และจุดสุดท้ายที่เรากำหนด



รูป 2.7 แสดงรูปที่วาดโดยคำสั่ง ARC

คำสั่ง TEXT : เราสามารถใส่กลุ่มของตัวอักษรเข้าไปในภาพด้วยคำสั่ง TEXT ซึ่งสามารถเขียนส่วนของตัวอักษรในลักษณะต่างๆ และ ยืด หด เอียง กลับข้าง วางตัวอักษรตรงไหนก็ได้ ขนาดเท่าใดก็ได้ ในที่นี้ จะเลือกใช้แบบเดียว คือ การเขียนตัวอักษรแบบปกติ

คำสั่ง STYLE : เป็นการกำหนดสไตล์ของตัวอักษรในแบบใดแบบหนึ่ง ซึ่งในที่นี้จะใช้แบบ Standard มีรูปแบบดังนี้

```
Command : STYLE
Text Style name (or?) : STANDARD
Font File <STANDARD> : ENTER
Height <0.0000> : 500
Width <1.00> : 150
Obliquing angle <0> : ENTER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Backwards? <N> ENTER
Upside - down? <N> ENTER
```

จากคำสั่งข้างบนเป็นการกำหนดสไตล์ของตัวอักษรซึ่ง ข้อกำหนดมีดังนี้

- กำหนดสไตล์เป็นแบบ Standard
- กำหนดความสูงของตัวอักษร 500 หน่วย
- กำหนดความกว้างของตัวอักษร 150 หน่วย
- มุมเอนของตัวอักษรเป็น 0
- ไม่เขียนกลับหลัง
- ไม่เขียนกลับหัว

เมื่อกำหนดสไตล์แล้วเราสามารถเรียกมาใช้ในคำสั่ง TEXT ได้ มีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : TEXT
Startpoint or Align/Center/Fit/Middle/Right/Style : Fit (F)
First text line point : 4,6
Second text line point : 7,6
TEXT : KITCHEN
```

จากข้างบนการแสดงตัวอักษรแบบ Fit คือ การให้ตัวอักษรแสดงภายในจุดสองจุดที่กำหนด ในที่นี้เราให้แสดงตัวอักษร KITCHEN อยู่ระหว่างจุด Coordinate (4,6) กับ (7,6) ซึ่งตัวอักษรจะถูกจัดให้อยู่ระหว่างสองจุดนี้อย่างเหมาะสม ดังรูป

รูป 2.8 แสดงการใช้คำสั่ง TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 คำสั่งในการเลือกส่วนของภาพ : คำสั่งบางคำสั่งเมื่อเราให้ไปแล้วจะย้อนถามกลับมาว่าจะให้เลือกเอาส่วนใดของภาพ ตัวอย่างเช่น

คำสั่ง ERASE : เป็นคำสั่งที่ใช้ในการลบภาพ หรือส่วนของภาพบางส่วน มีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : ERASE
Select Object or Window or Last :
```

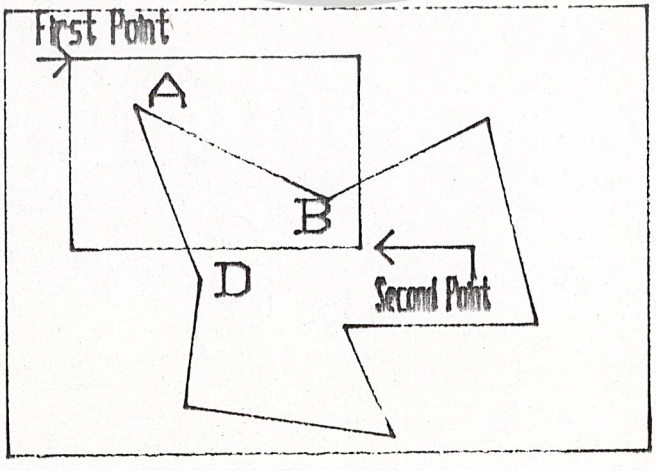
หมายถึงว่าวัตถุที่จะลบนั้นคือส่วนไหน ให้เราชี้ หรือตีกรอบล้อมรอบ หรือว่าให้ลบส่วนของภาพที่เขียนขึ้นมาเป็นส่วนสุดท้ายที่สุด การระบุส่วนของภาพทำได้โดยนิมน์ตัวอักษรลงไป มีดังนี้

M (Multiple) - เมื่อนิมน์ M แล้วกด ENTER จากนั้นใช้ Crosshair ชี้ไปที่วัตถุที่ละส่วนแล้วกด ENTER เครื่องจะทำเครื่องหมายไว้ก่อน เมื่อกด ENTER อีกทีภาพที่ถูกเลือกจะถูกลบออกไป

W (Window) - ด้วยการนิมน์ W แล้วกด ENTER จะกำหนดวัตถุที่อยู่ในสี่เหลี่ยมหรือหน้าต่างที่สามารถกำหนดขนาดได้โปรแกรมจะถามจุดสองจุดนี้

```
First Point :
Second Point :
```

ด้วยการใช้แป้นบังคับทิศทางบอกจุดมุมทั้งสอง จะได้รูปสี่เหลี่ยมบนจอภาพเมื่อขยายออก หรือหดเข้าจนคลอบคลุมพื้นที่ที่ต้องการแล้วให้กด ENTER วัตถุที่เราเลือกจะต้องอยู่ในหน้าต่างทั้งหมดจะไม่มีผลเพียงส่วนใดส่วนหนึ่ง



รูป 2.9 แสดงการลบภาพด้วยคำสั่ง ERASE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 คำสั่งที่อำนวยความสะดวกในการเขียนภาพ : มีคำสั่งต่างๆ ดังนี้

คำสั่ง SNAP : เป็นคำสั่งในการที่จะกำหนดให้เส้นสายใยเคลื่อนที่ไปทีละน้อย หรือกระโดดไปด้วยระยะห่างเท่าๆกัน กำหนดได้โดยใช้คำสั่ง SNAP เป็นการกำหนดระยะห่างการเคลื่อนตัวนี้ มีรูปแบบของคำสั่งดังนี้

```
Command : SNAP
On/Off/Value/Aspect/Rotate/Style :
```

เมื่อตอบ On เครื่องจะถามว่า

```
Value/Aspect<0.0000> : 0.5                    <ตั้งค่าใหม่>
```

ค่าความละเอียดของ SNAP จะเปลี่ยนเป็น 0.5
เมื่อตอบ Off เป็นการเลิกใช้ SNAP แต่ยังคงจำค่าเดิมของ SNAP ไว้ได้

คำสั่ง GRID : คำสั่งจะให้จุดวางเป็นแนวเส้นตรงตามระยะห่างที่กำหนด เราเปิดให้มี หรือ ปิดจุดที่เป็นกริดนี้ได้ สามารถเปลี่ยนระยะห่างได้ จุดกริดที่ปรากฏบนจอภาพไม่ได้เป็นส่วนของภาพ และไม่สามารถพลอ๊ตลงได้ กริดมีไว้เพื่อเราจะกะขนาดได้ถูกต้องเท่านั้น ถ้าเราจัดระยะห่างของกริดให้สัมพันธ์กับค่า SNAP แล้ว จะทำให้สามารถเขียนภาพได้สะดวก มีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : GRID
On/Off/Value(X)/Aspect :
```

On เป็นการแสดงเส้นกริดบนจอภาพ
Off เป็นการไม่ให้แสดงเส้นกริดบนจอภาพ
Value(X) ให้ค่าตัวเลขเป็นหน่วย ซึ่งจะเป็นระยะห่างระหว่างจุดที่เป็นเส้นกริด

คำสั่ง TRACE : เส้นตรงที่มีความกว้างเราเรียกว่า TRACE การใช้คำสั่ง TRACE ก็เหมือนคำสั่ง LINE แต่ต้องมีการกำหนดความกว้างของเส้นก่อน รูปแบบของคำสั่งมีดังนี้

```
Command : TRACE
Width : 0.25
From Point : 6,5
To Point   : 4,5
```

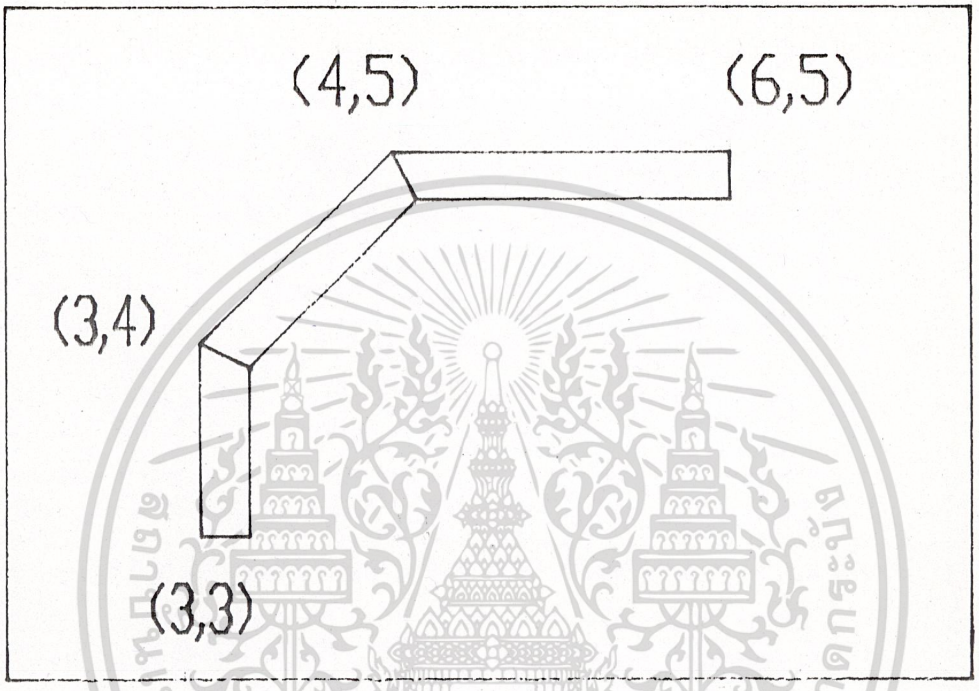
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

To Point : 3,4

To Point : 3,3

To Point : ENTER

จากการใช้คำสั่งข้างบนจะได้ภาพ ดังนี้



รูป 2.10 แสดงภาพที่เขียนโดยคำสั่ง TRACE

2.4.4 คำสั่งที่ช่วยในการทำงาน : เราได้ทราบคำสั่งในการเขียนส่วนของภาพมาข้างแล้ว ต่อไปจะเป็นคำสั่งช่วยในการทำงาน เช่น การออกจากโปรแกรม Drawing Editer เมื่อเราเขียนแบบแปลนเสร็จ เพื่อเข้า Main Menu ซึ่งมีการใช้คำสั่งดังนี้

คำสั่ง END : เมื่อใช้คำสั่งนี้แบบแปลนที่เขียนใน Drawing Editer ครั้งหลังสุดจะถูกบันทึกเข้าไว้ในไฟล์ที่เราตั้งชื่อไว้เมื่อตอนเข้าสู่ Drawing Editer ถ้าให้ชื่อ ABC ไฟล์นี้จะมีชื่อเป็น ABC.DWG ถ้าเรามีไฟล์ชื่อนี้อยู่ก่อน ไฟล์เดิมจะถูกเปลี่ยนชื่อเป็น ABC.BAK ซึ่งเป็นการ Update File

คำสั่ง QUIT : ด้วยคำสั่งนี้ภาพที่เขียนไว้ครั้งหลังสุดจะถูกยกเลิกไม่มีการบันทึกไฟล์เดิมที่มีอยู่จะคงอยู่อย่างเดิมไม่ถูกแก้ไข ก่อนที่จะออกจาก Drawing Editer เครื่องจะถามให้แน่ใจว่า

Really want to discard all changes to drawing?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตอบ N หรือกด ENTER เครื่องจะยังไม่ออกจาก Drawing Editer จะตอบ Y เมื่อกด ENTER เครื่องถึงจะออกจาก Drawing Editer และเข้าสู่ Main Menu ต่อไป

คำสั่ง SAVE : คำสั่งนี้จะบันทึกภาพที่แก้ไขครั้งสุดท้ายไว้แทนไฟล์เก่า โดยไม่ต้องออกจาก Drawing Editer ซึ่งจะมีประโยชน์ในการป้องกันไฟล์หายไปเมื่อเกิดไฟดับระหว่างการทำงาน

คำสั่ง LIMITS : เราสามารถกำหนดขอบเขตของภาพที่เราเขียน และมีการตรวจสอบอยู่ตลอดเวลาด้วยคำสั่ง LIMITS ซึ่งมีหน้าที่อยู่สามประการ คือ

1. ถ้าเรากำหนดให้มีการตรวจสอบขอบเขต เราสามารถจะกำหนด Coordinate ได้ โดยไม่มีการผิดพลาดว่าเขียนออกไปนอกขอบเขต
2. เส้นกริด(GRID)ที่มองเห็นจะแสดงขอบเขต
3. ขอบเขตที่กำหนดจะเป็นแฟคเตอร์ที่บอกว่า เมื่อภาพที่ถูกขยาย หรือหดด้วยคำสั่ง ZOOM ALL เป็นเท่าไร รูปแบบของคำสั่งมีดังนี้

```
Command : LIMITS
On/Off/Lower left corner<ค่าที่ใช้อยู่> :
```

On จะกำหนดให้มีการตรวจสอบขอบเขต

Off จะเป็นการไม่ให้ตรวจสอบขอบเขต แต่จะยังเก็บค่าขอบเขตเดิมไว้ในกรณีที่มีการ On อีก

การกำหนดจุด สามารถกำหนดได้ด้วยการใช้แป้นพิมพ์ หรือเครื่องชี้จะเป็นการกำหนดขอบเขตเป็นค่าของ Coordinate ของมุมล่างซ้ายระหว่างที่เราทำงานเราสามารถเปลี่ยนขอบเขตได้ตลอดเวลาโดยไม่คำนึงว่าของเดิมเป็นอะไรอยู่ สมมติให้ของเดิมที่มุมล่างซ้ายเป็น (0,0) บนขวาเป็น (10,10) เราจะเปลี่ยนค่าล่างซ้ายเป็น (4,0) บนขวาเป็น (7,14) สามารถทำได้ดังนี้

```
Command : Limits
On/Off/Lower left corner<0.0000,0.0000> : 4,0
Upper right corner<10,10> : 7,14
```

2.4.5 คำสั่งควบคุมการแสดงภาพ : การควบคุมตำแหน่งและการขยายภาพภายในกรอบ เราสามารถกำหนดได้โดยการใช้ชุดคำสั่งดังนี้

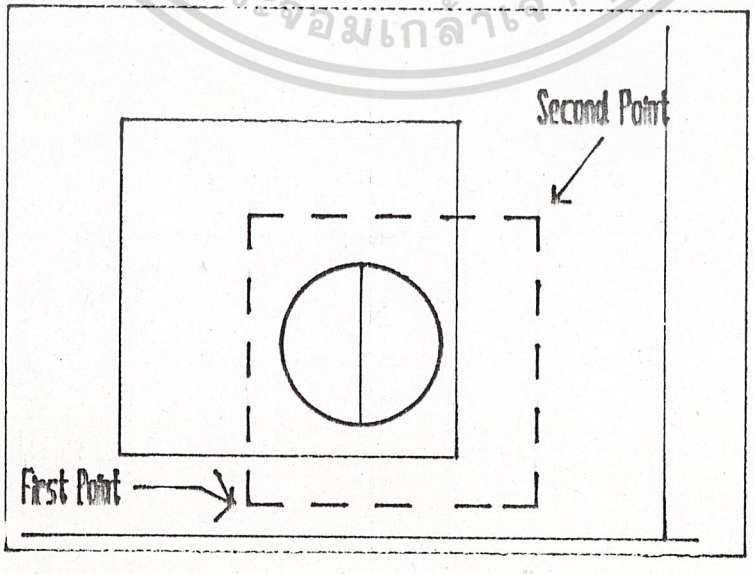
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง ZOOM : เหมือนกับเลนซ์ซูมของกล้องถ่ายรูป ที่สามารถเพิ่มขนาด หรือลดขนาดของสิ่งของที่เราดูอยู่บนจอภาพได้ การ ZOOM มีรายละเอียดอยู่หลายอย่าง ในที่นี้จะเลือกกล่าวถึงอยู่สองแบบ คือ การ ZOOM WINDOW กับการ ZOOM DYNAMIC ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้

```
Command : ZOOM
All/Center/Dynamic/Extents/Left/Previous/
Window/Scale(X) :
```

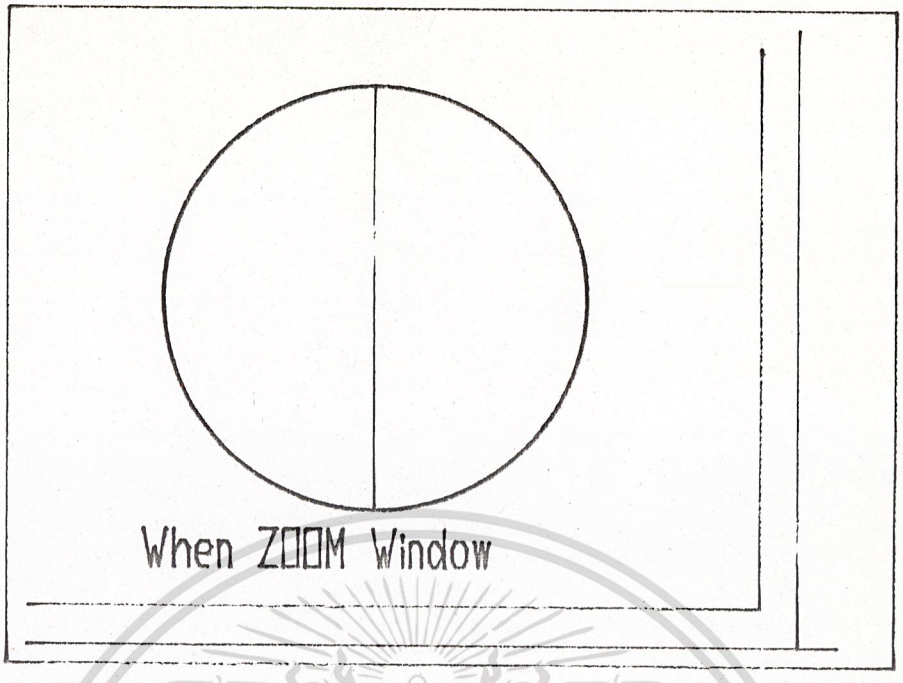
เมื่อเลือกการ ZOOM WINDOW เครื่องจะให้เรากำหนดพื้นที่ที่เราประสงค์ที่จะดูด้วยการให้ขอบมุมของจุดสองจุดที่อยู่ตรงข้ามกัน ซึ่งจะสร้างเป็นกรอบ ภาพที่อยู่ในกรอบจะขยายออกเต็มจอภาพ และศูนย์กลางของกรอบ จะอยู่ตรงจุดศูนย์กลางของจอภาพ รูปแบบของคำสั่งแสดงดังตัวอย่างต่อไปนี้

```
Command : ZOOM
All/Center/Dynamic/Extents/Left/Previous/
Window/Scale(X) : Window          ( หรือพิมพ์ W)
First Point :  ชี้ตรงจุดแรกของกรอบ
Second Point : ชี้ตรงจุดตรงกันข้ามของกรอบ
```



รูป 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



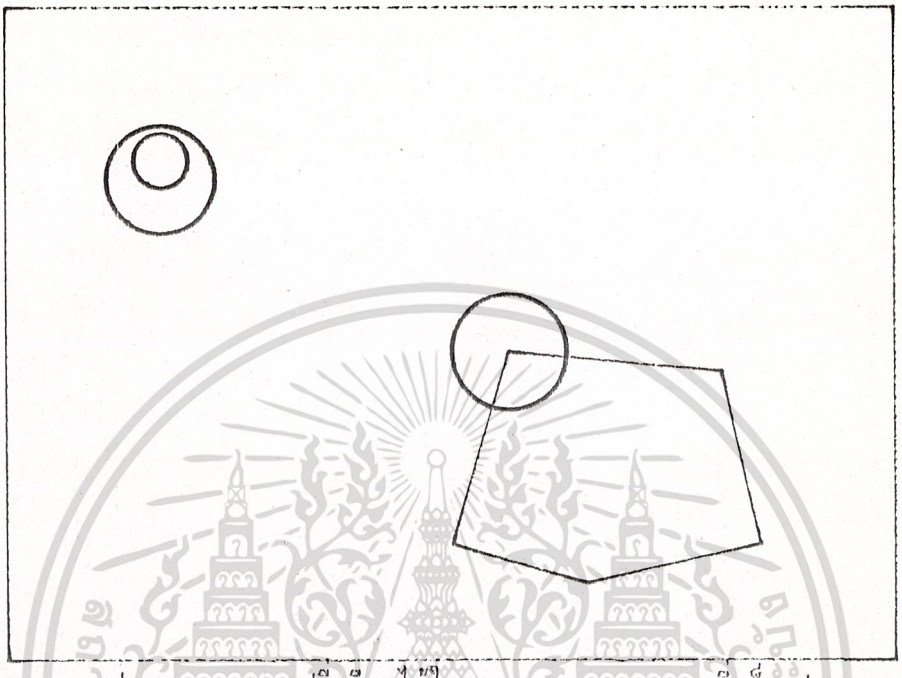
รูป 2.12 แสดงการใช้คำสั่ง ZOOM Window

เมื่อเลือกการ ZOOM Dynamic จะมีกรอบเกิดขึ้นเรียกว่า "กรอบมอง" กรอบนี้สามารถจะหด หรือ ขยาย และเลื่อนไปมาในจอภาพได้ เมื่อชอบใจก็ให้กรอบนี้ขยายออกมาเต็มจอภาพ รูปแบบของคำสั่งมีดังนี้

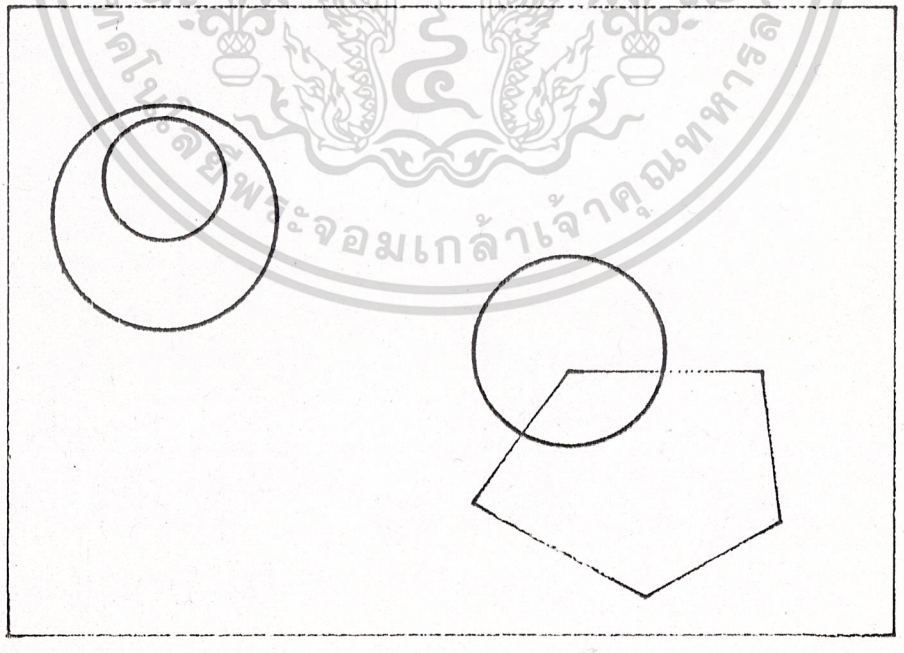
```
Command : ZOOM
All/Center/Dynamic/Extents/Left/Previous/
Window/Scale(X) : Dynamic      (หรือพิมพ์ D)
```

การ ZOOM Dynamic ค่อนข้างจะซับซ้อน แต่จะเห็นรูปได้ดีกว่าวิธีอื่น รูปภาพต่อไปจะแสดงขั้นตอนการ ZOOM Dynamic การ ZOOM แบบนี้ ถ้าใช้ Mouse จะทำให้สะดวกกว่าการใช้แป้นพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



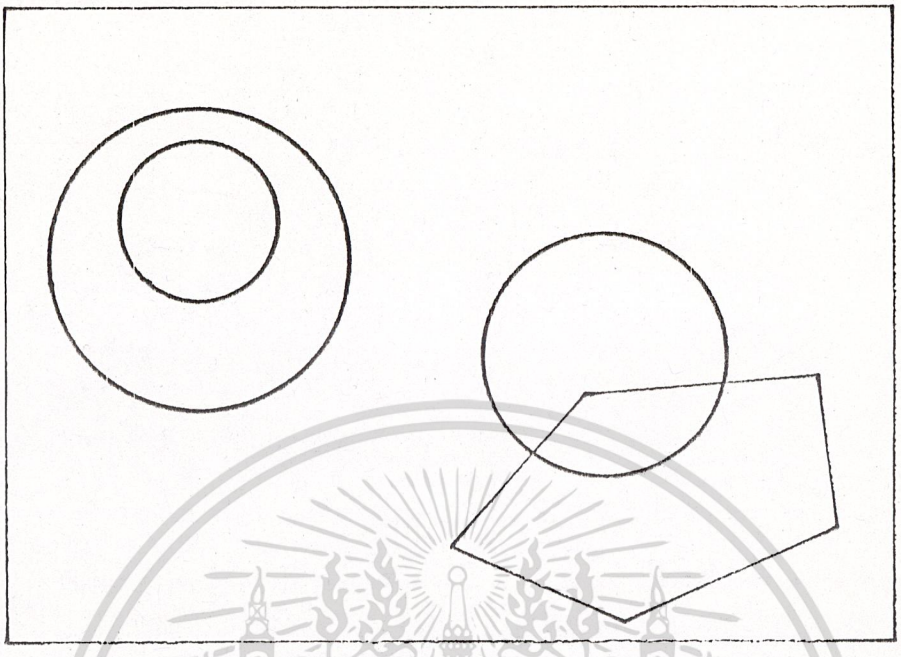
รูป 2.13 แสดงรูปที่เขียนไว้ใน Drawing Editor ที่เต็มรูป



รูป 2.14

ใช้คำสั่ง zoom window เพื่อขยายรูปทั้งสองให้เต็มที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.15

ในรูป 2.16 เมื่อใช้คำสั่ง ZOOM Dynamic จะได้รูปออกมาดังแสดงในรูป 2.17 ออกมา

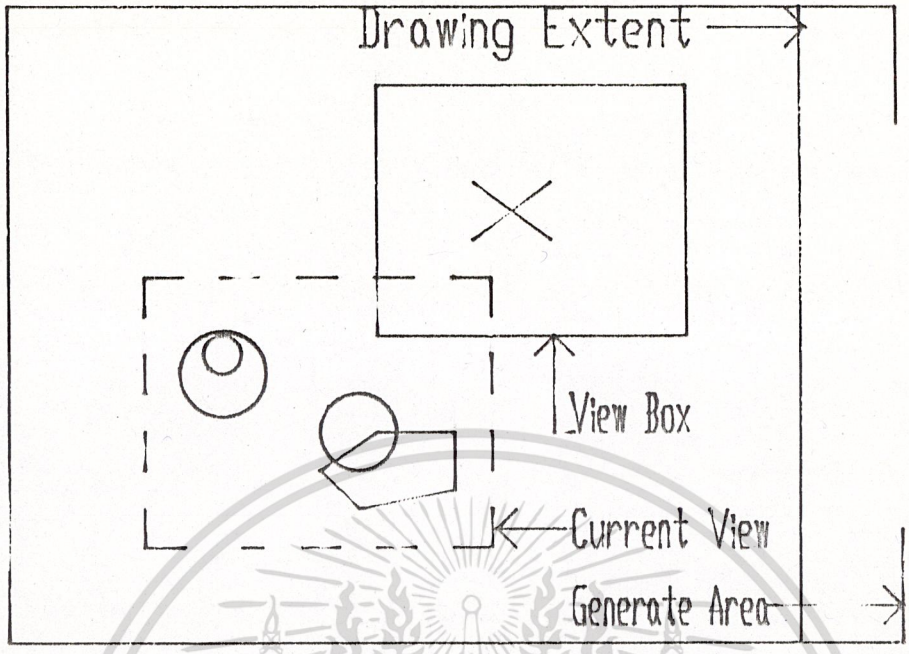
Drawing Extent เป็นรูปสี่เหลี่ยมแสดงขอบเขตของรูปคล้ายกับกรอบที่แสดงในรูปที่ 2.15 แต่มีขนาดเล็กลงมา

Generate Area พื้นที่สร้างภาพใหม่ ความจริงพื้นที่นี้ประกอบไปด้วยกรอบมุมดังนี้ [ ] แต่ถูก Drawing Extent หับเสียสามเส้น

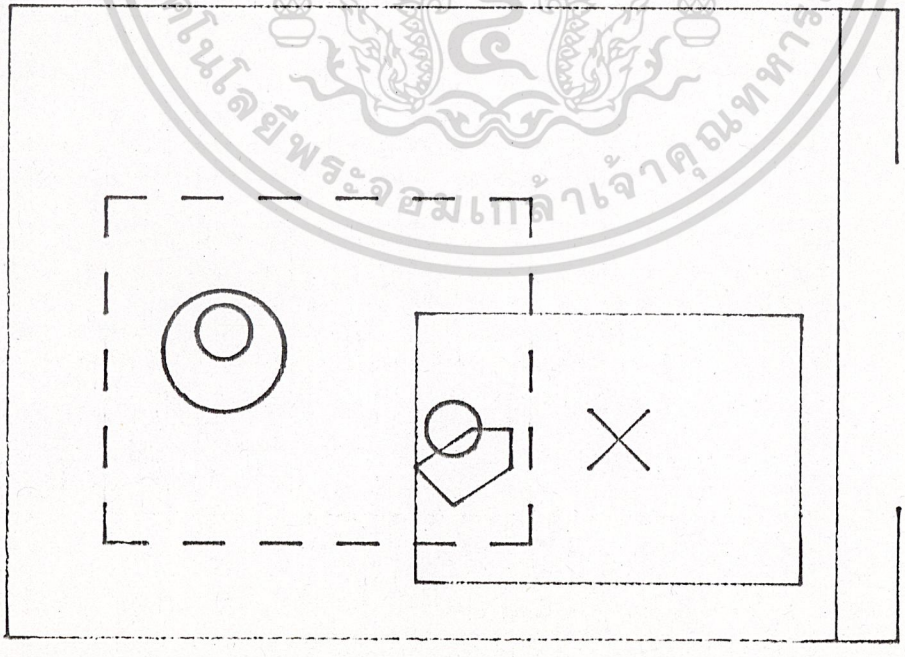
Current View คือแสดงรูปที่เห็นที่เห็นแต่บนจอในรูปที่ 2.16เพียงแต่ย่อลงมา

View Box เป็นกรอบสี่เหลี่ยมที่มีขนาดเท่ากับ Current view แต่เลื่อนไปมาได้ด้วยอุปกรณ์การชี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.16

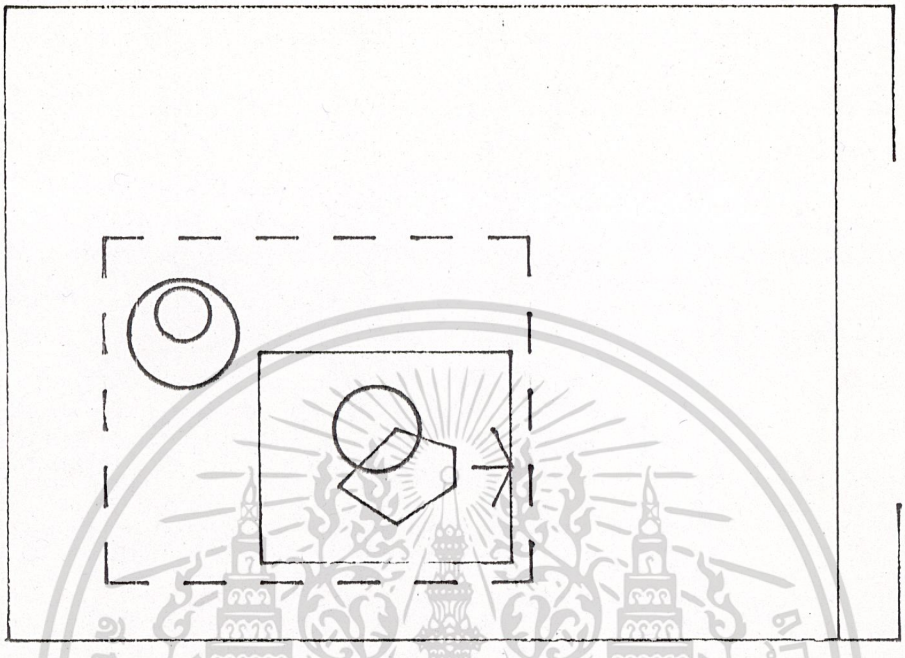


รูป 2.17

รูป 2.17 แสดงการเลื่อน View Box ให้ทางด้านซ้ายอยู่ใกล้ๆกับรูปทางขวาที่เรา จะ zoom ออกมา ถ้าเราใช้แป้นพิมพ์เมื่อได้ตำแหน่งที่ต้องการให้กดปุ่ม Pick ของ Mouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

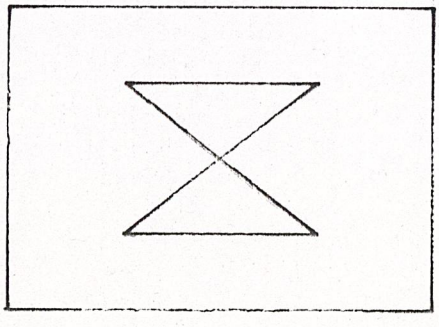
ทางซ้ายของ View Box จะตรึงอยู่กับที่



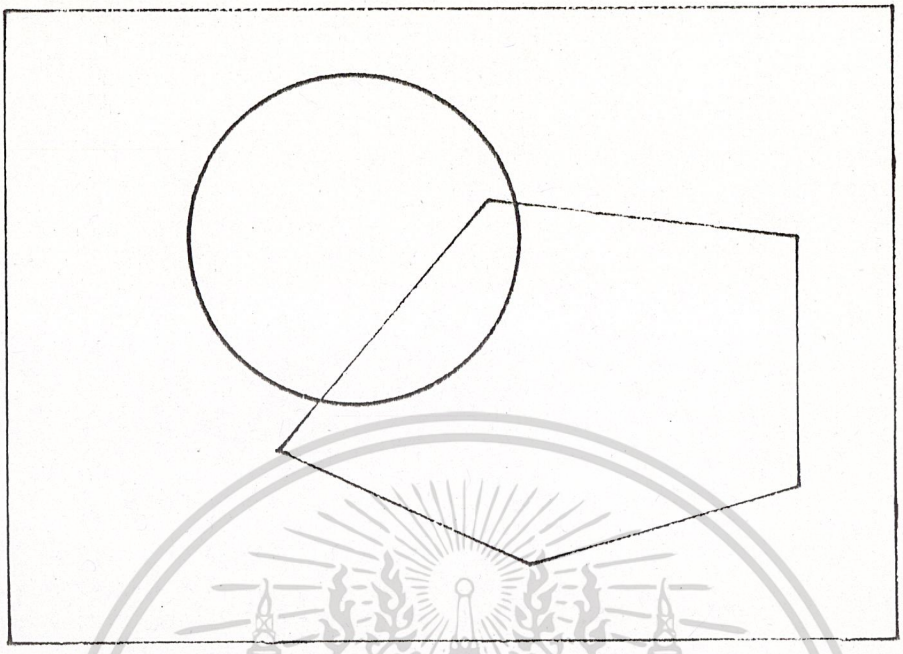
รูป 2.18

จากนี้เลื่อน Mouse เพื่อให้กรอบด้านซ้ายของ View Box ได้เหมือนแสดงในรูป 2.18 โปรดสังเกตว่าด้านซ้ายของ View Box ทางซ้ายมือจะเลื่อนไปซ้ายมือ หรือขวาไม่ได้ แต่ขึ้นลงได้ เมื่อได้พื้นที่คลอบคลุมรูปทางขวามือหมดแล้วกด ENTER จะได้รูปดังรูปที่ 2.19

การ ZOOM นั้นถ้า View Box อยู่ใน Generate Area แล้ว การ ZOOM จะเป็นไปด้วยความรวดเร็วแต่ถ้ารูปนี้อยู่นอก Area ก็ช้าหน่อยและมีเครื่องหมายดังนี้แสดงอยู่มุมล่างซ้าย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.19

คำสั่ง REDRAW : เป็นคำสั่งที่ให้เขียนภาพบนจอใหม่ และลบเครื่องหมายบาก(Blip)ที่ทำเป็นเครื่องหมายไว้ขณะเขียนภาพออกให้หมด รูปแบบของคำสั่งมีดังนี้

```
Command : REDRAW
```

2.4.5 คำสั่งเกี่ยวกับชั้นของภาพ : ชั้น(Layer)เปรียบเสมือนกระดาษแก้วใสวางทับกันบนแบบแปลนเป็นชั้นๆ เราสามารถจะกำหนดให้เขียนส่วนของภาพบนชั้นใดก็ได้ และชั้นของภาพจะมีกี่ชั้นก็ได้ ในแต่ละชั้นในภาพเดียวกันจะมี Limits ระบบ Coordinate และอัตราการ ZOOM เหมือนกัน การกำหนดจุดแต่ละจุดในแต่ละชั้นจะตรงกันหมด ในหัวข้อนี้ รายละเอียดจะมีมากขอกล่าวถึงเฉพาะที่ใช้ในการสร้างภาพในที่นี้เท่านั้น

การสร้างชั้นใหม่ด้วยคำสั่ง LAYER MAKE : เป็นคำสั่งในการสร้างชั้นใหม่ และกำหนดให้เป็นชั้นที่ใช้งานได้เลย ซึ่งสามารถกำหนดชั้นได้ไม่ว่าในขณะนั้น Drawing Editer จะแสดง Layer ใดอยู่ก็ตาม รูปแบบของคำสั่ง Layer Make มีดังนี้

```
Command : LAYER

?/Make/Set/New/On/Off/Color/Ltype/Freeze/Thaw : MAKE
New current layer <ของเดิมที่ใช้อยู่> : TANK (ชื่อชั้นใหม่)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใช้คำสั่งดังนี้แล้ว ชั้น Tank จะเกิดขึ้นใหม่และจะถูกกำหนดให้เป็นชั้นที่สามารถใช้งานได้เลย คือ สามารถวาดภาพลงไปได้เลยชั้น Tank จะเก็บรายละเอียดของภาพนั้น เช่น สีที่ใช้ในชั้นนั้น แบบของเส้นที่ใช้ ไว้เมื่อการวาดภาพสิ้นสุดลง

คำสั่ง COLOR : ปกติเราจะกำหนดสีของส่วนของภาพที่เราเขียนโดยใช้คำสั่ง Layer Color คือชั้นนี้รูปที่เขียนจะต้องเป็นสีนี้แต่ด้วยคำสั่ง COLOR นี้เราจะกำหนดส่วนของภาพที่กำลังจะเขียนนี้ให้เป็นสีอะไรก็ได้ โดยไม่คำนึงถึงว่าจะอยู่ในชั้นไหน รูปแบบของคำสั่งมีดังนี้

```
Command : COLOR
New entity color <BYLAYER> :
```

จากรูปแบบของคำสั่ง เราสามารถใส่ค่าสีได้ตั้งแต่ 1 ถึง 255 หรือใช้ชื่อสี เช่น RED ลงไปตรงๆได้

คำสั่ง WBLOCK : โดยทั่วไป BLOCK คือส่วนของภาพชุดหนึ่งซึ่งรวมกันเป็นรูปที่ซับซ้อนและมีชื่อเรียก เราสามารถเรียกชื่อนี้นำมาสอดแทรกในภาพที่ใดก็ได้ตามต้องการ ในการสอดแทรกนี้ เราสามารถเปลี่ยนมาตราส่วน สามารถหมุน กลับข้างกลับหลังได้

คำสั่ง BLOCK จะเป็นการเก็บภาพที่กำหนดเป็น BLOCK ที่เราสามารถกำหนดชื่อได้ แต่คำสั่ง BLOCK จะใช้ได้เฉพาะแบบแปลนที่กำลังใช้งานอยู่เท่านั้น นำไปใช้กับแบบแปลนอื่นไม่ได้ แต่ถ้าเราใช้คำสั่งสร้าง BLOCK คือคำสั่ง WBLOCK โปรแกรมจะเขียน BLOCK เป็น Drawing File (ก่อนอื่นเราต้องมี BLOCK เสียก่อน จึงจะเขียนลงไปได้) รูปแบบของคำสั่งมีดังต่อไปนี้

```
Command : WBLOCK
File name : (ให้ชื่อของไฟล์)
Block name : (ให้ชื่อของ Block)
```

การให้ชื่อของ Block มีอยู่ 4 แบบ แต่ละแบบมีผลในการสร้าง Block ดังนี้

1. ให้ชื่อของ Block โดยตรง - จะบันทึก Block นั้นไว้ในดิสค์
2. พิมพ์ = แทนชื่อ - แสดงว่าได้ให้ชื่อของไฟล์เหมือนกับชื่อของ Block
3. พิมพ์ * แทนชื่อ - แบบแปลนทั้งแผ่นจะถูกบันทึกลงในดิสค์ คล้ายกับคำสั่ง Save เว้นแต่ข้อมูลใดที่ไม่เกี่ยวกับ Block จะไม่ถูกบันทึก ทั้งนี้เมื่อ Block ถูกลบข้อมูลที่เกี่ยวข้องจะถูกลบไปหมด
4. กด ENTER - เป็นการสร้าง Block ใหม่เหมือนคำสั่ง Block แต่บันทึกโดยตรงเข้าดิสค์เลยโดยเครื่องจะถาม INSERT POINT และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ระบุว่าจะเอารูปไหน ชื่อของ Block จะเหมือนกับชื่อของไฟล์

คำสั่ง INSERT : คือคำสั่งที่ใช้สำหรับแทรกภาพ Block ลงในภาพตามตำแหน่งที่ต้องการ รูปแบบของคำสั่งมีดังนี้

Command : INSERT

Block name (or?) : (ให้ชื่อของ Block)

Insert Point : (ระบุว่าจะให้วางภาพที่จุดไหน)

X Scale factor <1>/corner/XYZ : (ให้ค่า)

Y Scale factor <defdault = x> : (ให้ค่า)

Rotate Angle <0> : (ให้ค่าหรือชี้จุด)

ในที่นี้เราให้ตัวคูณของมาตราส่วน X Scale = 1 และ Y Scale = 1 มุม Rotate = 0 และ 270 ส่วน Insert Point นั้นขึ้นอยู่กับผู้ใช้ว่าต้องการแทรกภาพเข้าที่จุดใดสามารถทำได้โดยใช้ตัวชี้แล้วกด ENTER หรือให้ค่า Coordinate

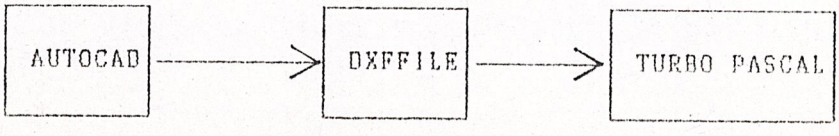
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างภาพกระบวนการ และ การกำหนดอุปกรณ์ในกระบวนการ

ในบทที่ 2 เป็นการแนะนำทฤษฎีและการใช้คำสั่งต่างๆของ AUTOCAD ที่ใช้ในการสร้างภาพกระบวนการ ในบทนี้จะเป็นการสร้างภาพลงในคอมพิวเตอร์โดยใช้คำสั่งที่กล่าวมาแล้ว การสร้างภาพหรือผังภาพของกระบวนการเพื่อนำมาแสดงผลด้วยระบบนี้นั้น ในการออกแบบได้กำหนดให้ผู้ใช้สร้างขึ้นเองโดยใช้โปรแกรมสำเร็จรูปต่างๆ ที่มีอยู่ในท้องตลาด เช่น AutoCAD,Prodesign II,Autosketh ในที่นี้จะใช้โปรแกรมในการวาดภาพของกระบวนการ คือ AutoCAD วัตถุประสงค์ที่ใช้โปรแกรมสำเร็จรูปนี้เนื่องมาจากว่า ในปัจจุบันเทคโนโลยีของคอมพิวเตอร์ช่วยในการออกแบบและวาด(COMPUTER - AIDED DESIGN AND DRAFTING) ได้พัฒนาไปมาก ประกอบกับโปรแกรมสำเร็จรูปสำหรับงานดังกล่าวก็มีแพร่หลาย และโปรแกรมต่างๆเหล่านี้ใช้งานได้ง่ายและมีประสิทธิภาพสูง ดังนั้นในการออกแบบจึงกำหนดใช้โปรแกรมสำเร็จรูปนี้ในการสร้างภาพ หรือผังภาพของกระบวนการ แต่การนำโปรแกรมสำเร็จรูปเหล่านี้มาใช้ก็มีข้อกำหนดอยู่บ้าง คือ

- ต้องสามารถกำหนดชั้น(LAYER)ในการวาดได้
- ต้องมีเอาท์พุทเป็นภาษา HPGL (Hewlet Packard Graphics Languge) และต้องสามารถเก็บไฟล์ HPGL ดังกล่าวลงแผ่นจานแม่เหล็กได้

การมีข้อกำหนดเหล่านี้เนื่องมาจาก การออกแบบโปรแกรมควบคุมการแสดงผล ได้กำหนดส่วนการวาดภาพแสดงผลเป็นกราฟฟิก ซึ่งต้องสามารถแสดงผลได้อย่างรวดเร็ว การที่จะแสดงได้อย่างรวดเร็วการวาดต้องใช้คำสั่งในการวาดเป็นเส้นตรงอย่างเดียว เนื่องจากการวาดเส้นตรงเป็นคำสั่งที่ทำงานได้รวดเร็วที่สุด เส้นโค้งและวงกลมต่างๆก็ใช้เส้นตรงเล็กๆประกอบกัน ซึ่งโปรแกรมสำเร็จรูปที่สามารถกำหนดการวาดในลักษณะนี้ได้คือโปรแกรม AutoCAD เมื่อใช้ AutoCAD วาดภาพกระบวนการเรียบร้อยแล้วเราจะทำการเปลี่ยนภาพที่วาดโดยใช้ชุดคำสั่งของ AutoCAD มาเป็นภาพที่วาดโดยชุดคำสั่งของเทอร์โบ ปาสคาล เนื่องจากว่าในส่วนของโปรแกรมแสดงผลเราใช้โปรแกรมภาษาเทอร์โบปาสคาล

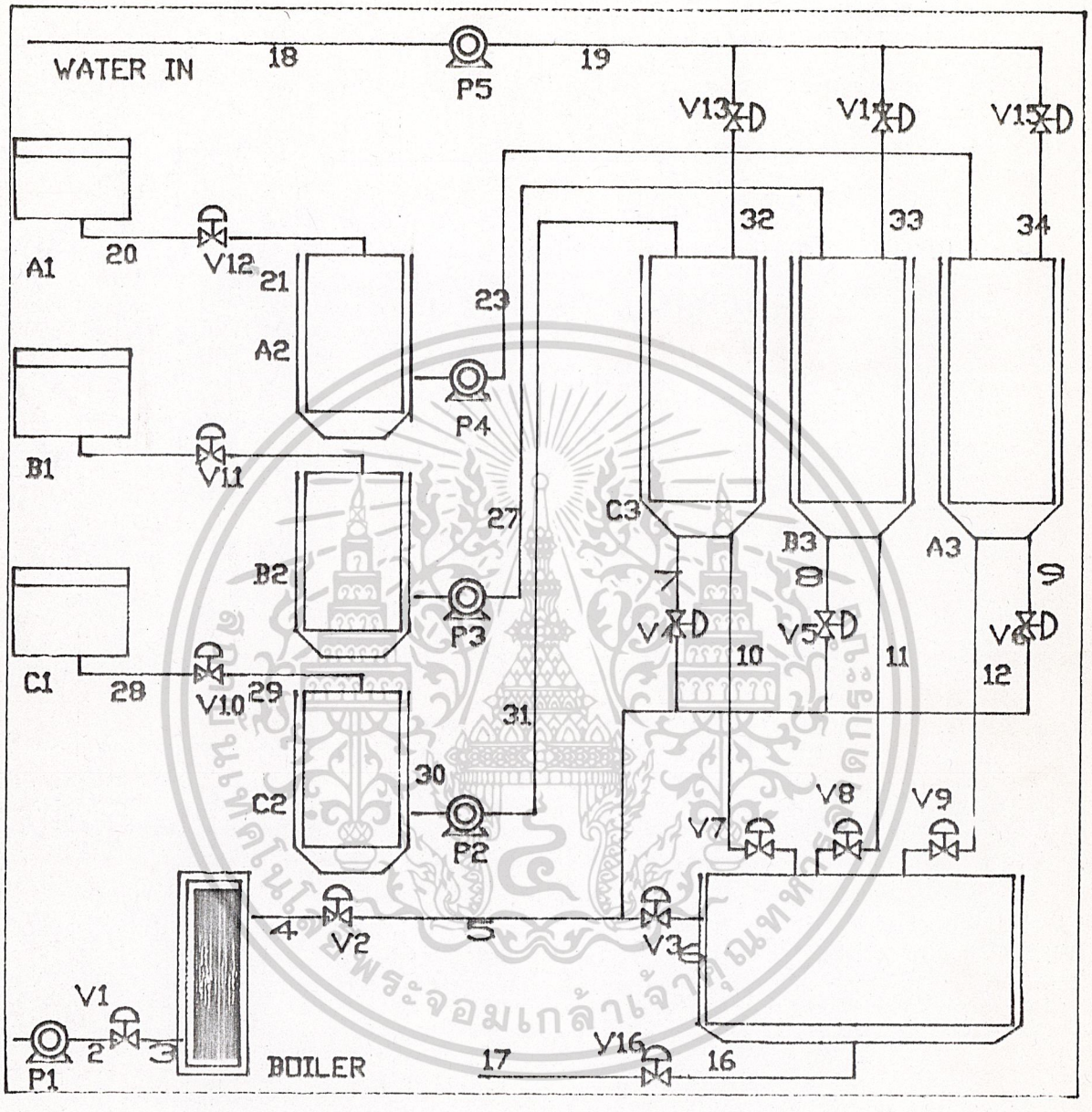


รูป 3.1 แสดงการเชื่อมต่อไฟล์ AutoCAD กับ Turbo Pascal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนและข้อกำหนดในการสร้างภาพของกระบวนการนี้มีอยู่หลายข้อแต่ในการใช้งานจริงแล้วผู้ใช้อาจจะเปลี่ยนแปลงข้อกำหนดเหล่านี้ได้ เนื่องจากระบบมีความยืดหยุ่นในลักษณะนี้สูง แต่ในขั้นตอนหลักจะมีอยู่ 5 ขั้นตอน ส่วนข้อกำหนดต่างๆ จะได้รวมไว้ในหัวข้อที่เกี่ยวข้องด้วย โดยสมมติให้กระบวนการ มีลักษณะดังรูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.2 แสดงภาพตัวอย่างของกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

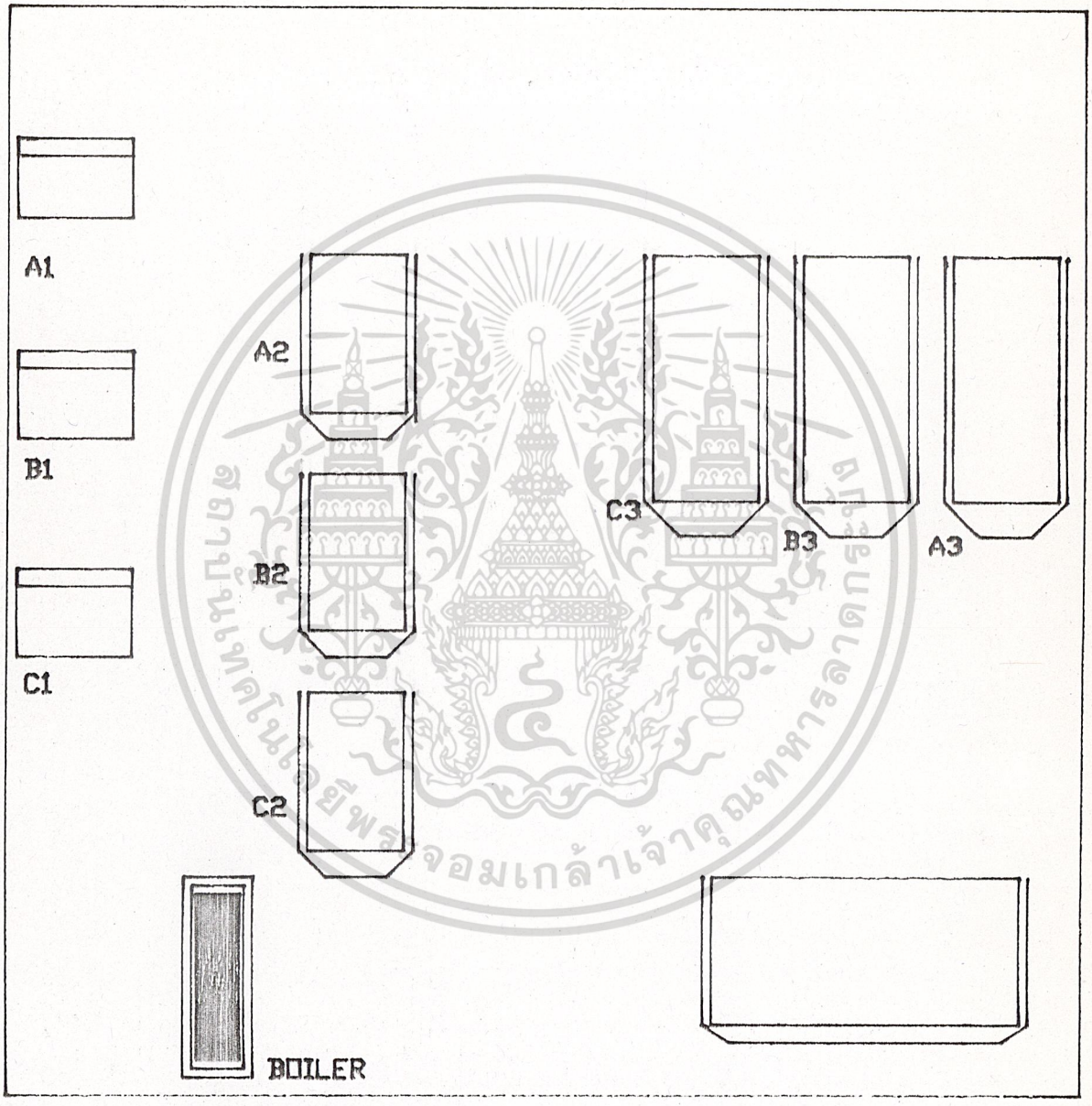
สมมติให้ กระบวนการตัวอย่าง เป็นการผสมสารเคมีซึ่งเป็นของเหลว 3 ชนิด (สาร A,B,C) เข้าด้วยกัน โดยวัตถุดิบของสารเคมี A จะอยู่ในถัง A1 สารเคมี B จะเก็บในถัง B1 สารเคมี C จะเก็บในถัง C1แล้วนำมาเพิ่มอุณหภูมิที่ถัง A2,B2 และC2 โดยมีวาล์ว 10,11 และ12 เป็นตัวควบคุมจากนั้นต้องนำไปผสมกับน้ำจาก Boiler เพื่อให้อุณหภูมิสูงขึ้นอีก หลังจากนั้นจะถูกส่งไปยังถังผสม(MIXTANK)ซึ่งอุณหภูมิจะถูกควบคุมให้คงที่ จากนั้นจะถูกปล่อยออจากถังผสม ด้วยการเปิดวาล์ว 16 ดังนั้นการสร้างภาพก็จะต้องกำหนดว่าอุปกรณ์ใดจะสร้างอย่างไรตามขั้นตอนต่อไปนี้

3.1 การสร้างภาพกระบวนการ : ในขั้นตอนนี้ผู้ใช้ยังอาจจะยังไม่ต้องสร้างภาพลงในคอมพิวเตอร์ก็ได้ อาจจะใช้วิธีวาดลงกระดาษเสียก่อน เนื่องจากขั้นตอนนี้เป็นการสร้างภาพกระบวนการทั้งหมดขึ้นมาเพื่อใช้อ้างอิงในการวาดภายหลังเท่านั้น และนอกเหนือจากภาพกระบวนการที่ปรากฏบนจอแสดงผลแล้ว ก็สามารถกำหนด หรือสร้างภาพที่ต้องการแสดงผลลงไปด้วยก็ได้ ในที่นี้จะใช้ภาพดังรูป 3.2 เท่านั้น

3.2 การแบ่งชั้นของภาพกระบวนการ : การแบ่งชั้นของภาพออกเป็นชั้น จะทำให้สามารถตรวจสอบ หรือแก้ไขภาพได้โดยสะดวก โดยเฉพาะในกระบวนการที่มีความซับซ้อนสูงๆซึ่งหลักการเราจะแบ่งตามลักษณะการแสดงผลของอุปกรณ์เป็นหลักเช่นจากตัวอย่างรูป 3.2 นั้นจะแบ่งอุปกรณ์ออกเป็นกลุ่มดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

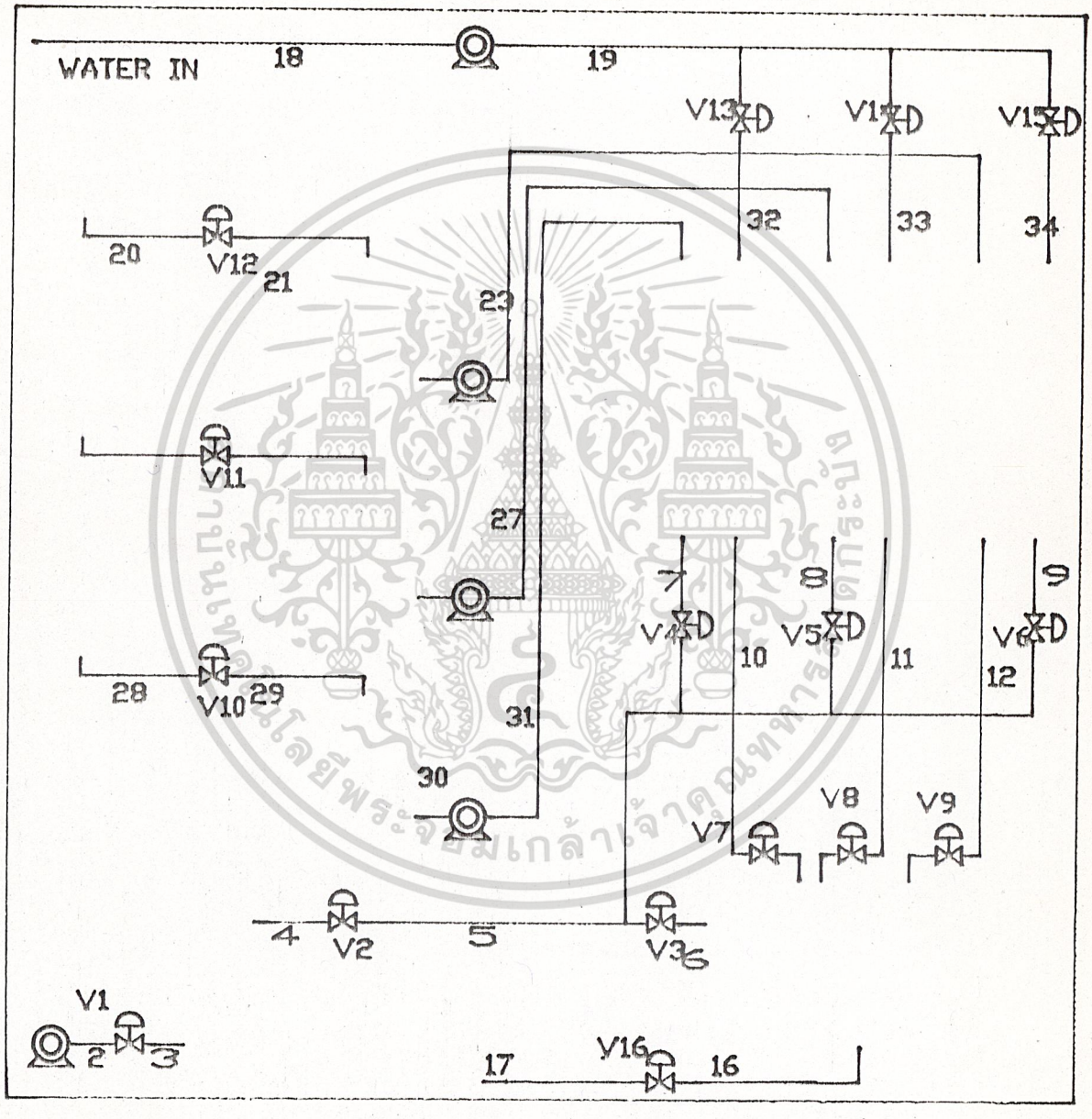
กลุ่มที่ 1 อุปกรณ์ที่ไม่มีการเปลี่ยนแปลงสภาวะ : อุปกรณ์ในกลุ่มนี้ได้แก่ ตัวอักษรที่ใช้เป็นชื่อของอุปกรณ์ต่างๆ และถัง(TANK)ซึ่งการเปลี่ยนแปลงจะเป็นระดับของเหลวในถังเท่านั้น แสดงดังรูป 3.3



รูป 3.3 แสดงอุปกรณ์ที่ไม่มีการเปลี่ยนแปลงสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

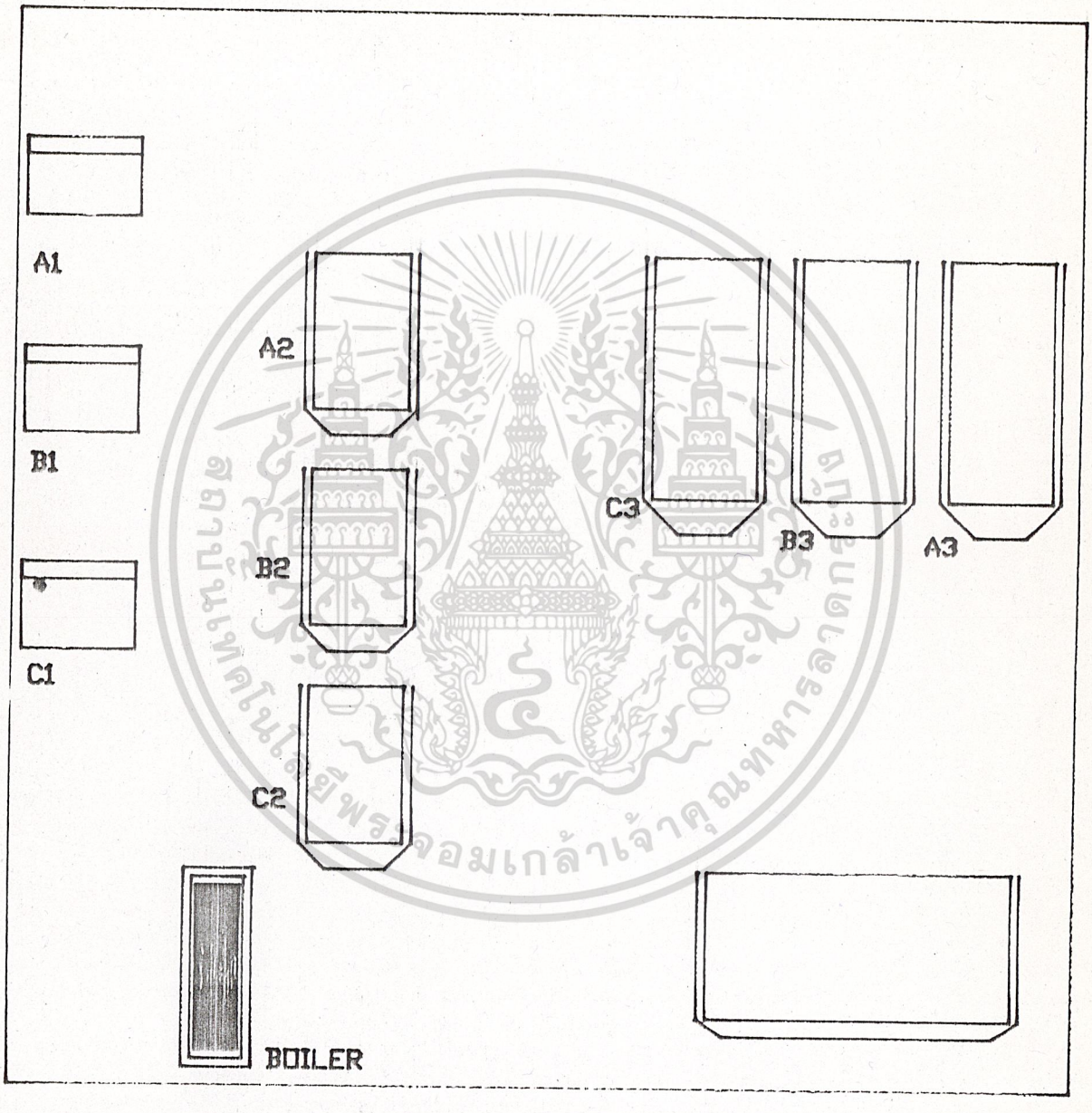
กลุ่มที่ 2 อุปกรณ์ที่แสดงการเปลี่ยนแปลงสภาวะโดยการเปลี่ยนสี : อุปกรณ์กลุ่มนี้ได้แก่ ปั๊ม(PUMP) วาล์ว(VALVE) และท่อ(PIPE) ซึ่งอุปกรณ์กลุ่มนี้จะมีสภาวะเป็นแบบดิจิตอล คือ มีสภาวะ "ปิด" กับ "เปิด" ดังนั้น เราสามารถกำหนดได้ว่าในสภาวะเปิดจะให้อุปกรณ์แสดงเป็นสีใด และ สภาวะปิดแสดงสีใดซึ่งอุปกรณ์แต่ละตัวสามารถกำหนดสีได้เฉพาะตัว เช่น วาล์วตัวที่ 11 ในที่นี้สภาวะปิดมีสีขาว สภาวะเปิดใช้สี แดง เป็นต้น ดังแสดงในรูป 3.4



รูป 3.4 แสดงอุปกรณ์ที่แสดงการเปลี่ยนแปลงสภาวะโดยการเปลี่ยนสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่ 3 อุปกรณ์ที่แสดงการเปลี่ยนแปลงสภาวะโดยการเปลี่ยนระดับ : อุปกรณ์ในกลุ่มนี้ได้แก่ ระดับของเหลวในถัง ในที่นี้ใช้การเลื่อนแถบสีขึ้นลงอุปกรณ์กลุ่มนี้ส่วนมากจะรับสภาวะแบบอนาลอกมาจากกระบวนการซึ่งระดับในถังจะเป็นค่าร้อยละ (PERCENTAGE) ของระดับสูงสุดของถัง อุปกรณ์ในกลุ่มนี้แสดงได้ดังรูป



รูป 3.5 แสดงอุปกรณ์ที่แสดงการเปลี่ยนแปลงสภาวะโดยการเปลี่ยนระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสร้างภาพลงในคอมพิวเตอร์ : ขั้นตอนนี้จะเป็นการนำแบบที่ได้ร่างไว้ทั้งหมดนำมาสร้างลงในคอมพิวเตอร์โดยแบ่งเป็นชั้นตามกำหนดไว้ส่วนวิธีการสร้างนั้นจะขึ้นอยู่กับโปรแกรมที่นำมาใช้ซึ่งแต่ละโปรแกรม จะมีคำสั่งและการใช้งานแตกต่างกันออกไป แต่ในการสร้างจำเป็นจะต้องสร้างตามเงื่อนไขที่กำหนด คือแบ่งแยกอุปกรณ์แต่ละตัวออกจากกัน ดังตัวอย่างการสร้างอุปกรณ์แต่ละกลุ่มดังนี้

ตัวอย่างการสร้างภาพกระบวนการและการกำหนดอุปกรณ์ในกระบวนการ : ในตัวอย่างนี้จะใช้โปรแกรม AutoCAD ในการสร้างภาพ ส่วนการใช้โปรแกรมอื่นๆก็จะมีลักษณะเดียวกันแตกต่างที่รายละเอียดเท่านั้น

- ร่างแบบของกระบวนการ สมมติให้ใช้ดังรูป 3.2
- การแบ่งชั้นของภาพกระบวนการ จะแบ่งตามกลุ่มอุปกรณ์ ซึ่งแต่ละกลุ่มมีดังนี้

ชั้นของท่อ - ท่อมี 34 ท่อ แบ่งเป็นชั้นละท่อ 34 ชั้น
ชั้นของวาล์ว - วาล์ว 16 วาล์ว แบ่งเป็นชั้นละตัว 16 ชั้น
ชั้นของปั๊ม - ปั๊มมี 5 ตัว แบ่งเป็นชั้นละตัว 5 ชั้น
ชั้นของถัง - อยู่ในชั้นของถัง

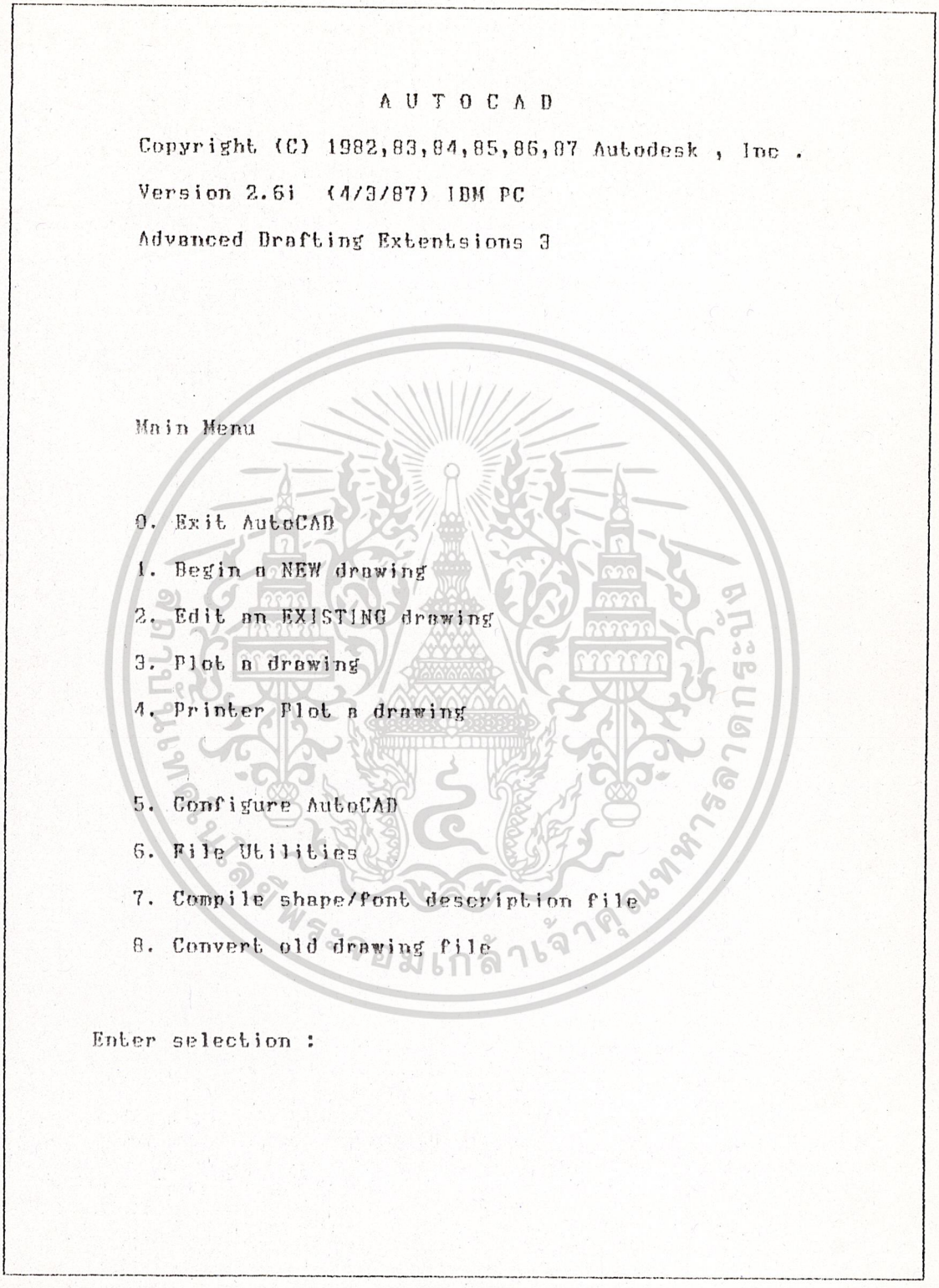
การสร้างภาพด้วยโปรแกรม AutoCAD มีขั้นตอนดังนี้

- เปิดเครื่องคอมพิวเตอร์ เมื่อเข้าสู่ Dos แล้ว ให้ใส่แผ่น AutoCAD ที่มีชื่อไฟล์ ACAD.EXE แล้วกด ENTER

A>ACAD ENTER_

รอสักครู่บนจอภาพจะปรากฏเมนูดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.6 แสดงเมนของ AutoCAD

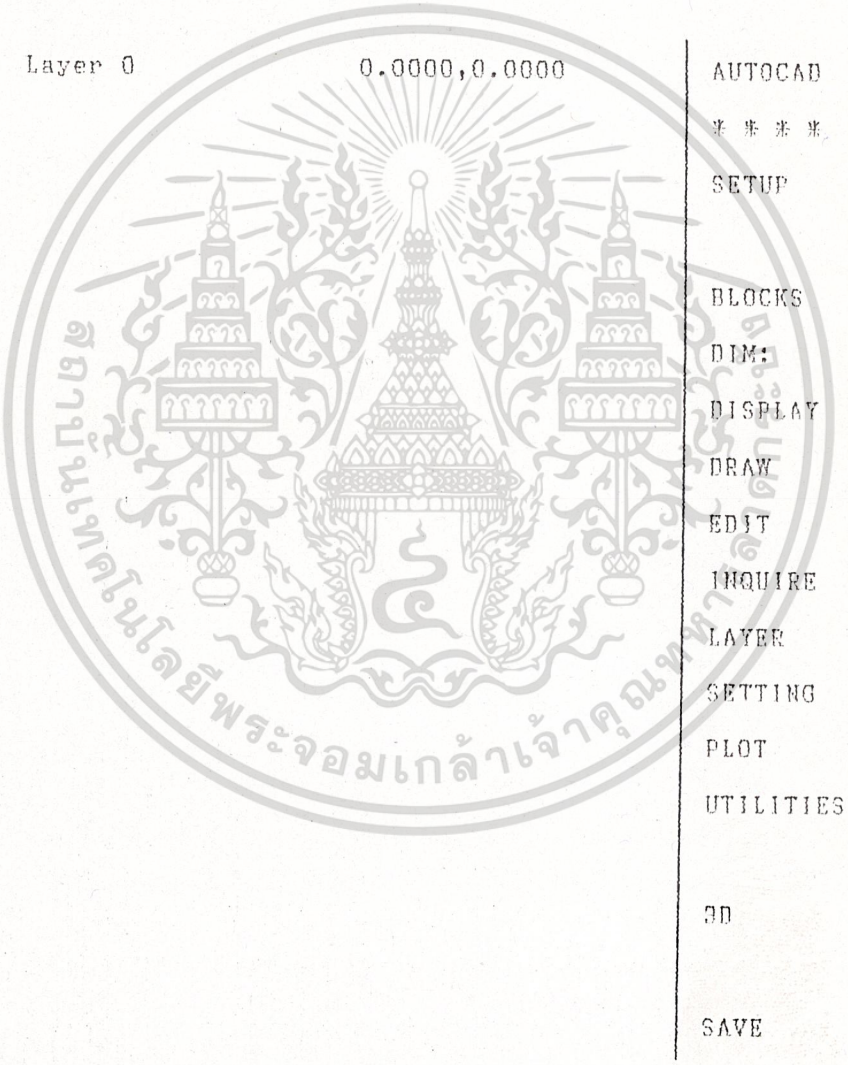
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเมนูให้เลือกเมนูหมายเลข 1 โดยตอบดังนี้

selection : 1 [ENTER_

Enter NAME of drawing : B:ABC

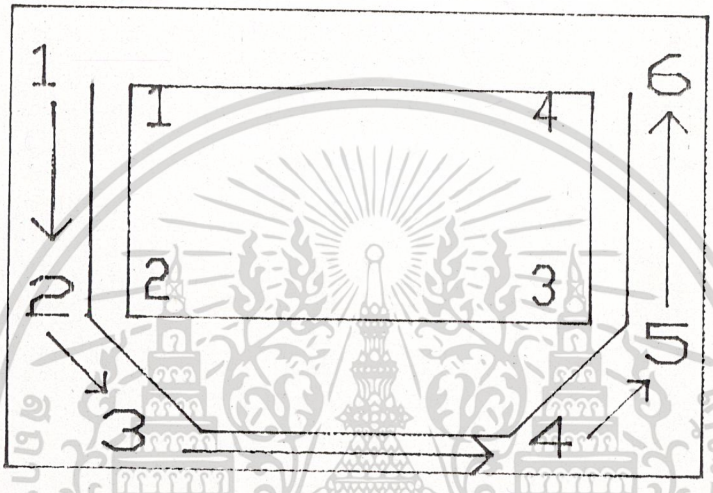
เป็นการกำหนดชื่อแบบแปลนชื่อ ABC ซึ่งจะเป็นชื่อที่จะถูกโหลดลงบนแผ่นจานแม่เหล็กในไดร์ฟที่ใช้งานอยู่เมื่อวาดแบบแปลนเสร็จ เมื่อเลือกเมนูหมายเลข 1 จะได้ภาพปรากฏบนจอภาพดังนี้



รูป 3.7 แสดงจอภาพ AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การวาดอุปกรณ์ที่ไม่มีการเปลี่ยนแปลงสภาวะ : คือการวาด TANK ซึ่งเราจะกำหนดให้ TANK อยู่ในชั้นที่ชื่อว่า TANK การวาดโดยใช้โปรแกรม AutoCAD มีการใช้ชุดคำสั่งที่มีรูปแบบดังต่อไปนี้



รูป 3.8 แสดงการวาดถัง(TANK)

รูปแบบคำสั่งมีดังนี้

```
Command : LINE
From Point : ( start coordinate 1)
To Point   : (coordinate 2)
To Point   : (coordinate 3)
To Point   : (coordinate 4)
To Point   : (coordinate 5)
To Point   : (coordinate 6)
To Point   : ENTER
```

รูปสี่เหลี่ยมด้านในก็ใช้คำสั่ง line เช่นเดียวกัน

```
Command : LINE
```

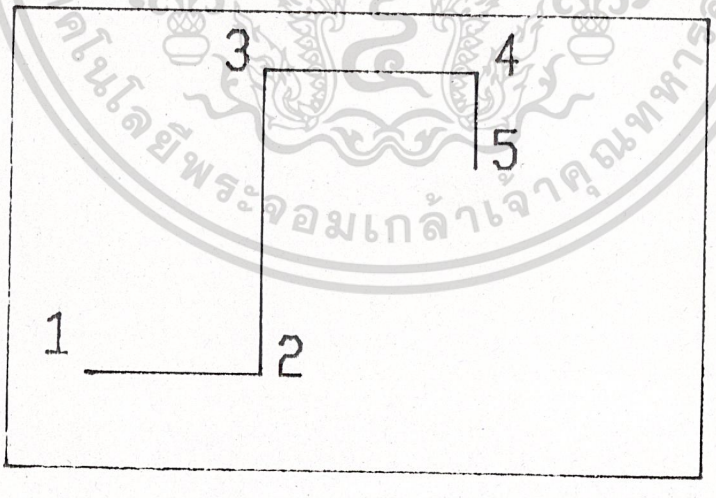
เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
From Point : ( start coordinate 1)
To Point   : (coordinate 2)
To Point   : (coordinate 3)
To Point   : (coordinate 4)
To Point   : (back to coordinate 1)
To Point   : ENTER
```

เมื่อใช้ชุดคำสั่งดังข้างบน เราจะได้รูปภาพดังรูป 3.8

3.3.2 การวาดภาพอุปกรณ์ที่แสดงการเปลี่ยนแปลงสภาวะโดยการเปลี่ยนสี : ได้แก่ รูปภาพท่อ วาล์ว ปั๊ม มีการใช้คำสั่งดังต่อไปนี้

การวาด ท่อ(PIPE) : ท่อในภาพกระบวนการตัวอย่างมีทั้งหมด 34 ท่อ โดยแยกกันอยู่ชั้นละท่อ ซึ่งมีชื่อของชั้น(layer)เป็นตัวบอกว่าท่อนั้นเป็นท่อหมายเลขอะไร เช่น ท่อหมายเลข 10 จะอยู่ในชั้นที่ชื่อว่า PIPE10 ท่อหมายเลข34 จะอยู่ในชั้นที่ชื่อว่า PIPE34 ในส่วนของปั๊ม และวาล์วก็เช่นเดียวกัน การวาดท่อใช้คำสั่งในการลากเส้นคือคำสั่ง line เช่นเดียวกันดังตัวอย่าง



รูป 3.9 แสดงรูปท่อโดยใช้คำสั่ง line

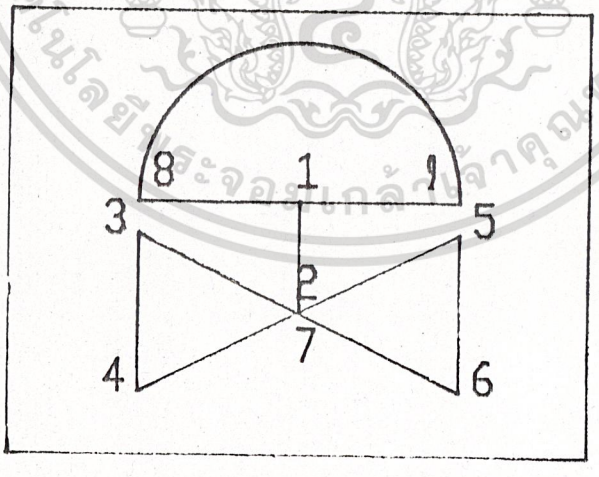
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการใช้คำสั่งดังนี้

```
Command : LINE
From point : (coordinate 1)
To point   : (coordinate 2)
To point   : (coordinate 3)
To point   : (coordinate 4)
To point   : (coordinate 5)
To point   : ENTER
```

เมื่อใช้ชุดคำสั่งข้างบนจะได้รูปท่อดังรูป 3.9

การวาด และการสร้าง BLOCK วาล์ว : ในการวาดวาล์วเราจะวาดวาล์วขึ้นมาต้วเดียวแล้วทำการเก็บภาพวาล์วนั้นเป็น Block โดยใช้คำสั่ง WBlock ในขั้นแรกเราจะต้องโหลด Drawing Editer ของ AutoCAD ขึ้นมาก่อนโดยตั้งชื่อแบบแปลนชื่อว่า VALVE เมื่อกำหนดเรียบร้อยทำการวาดดังรูป



รูป 3.10 แสดงรูปวาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ในการวาดภาพวาล์ว มีรูปแบบดังนี้

```
Command : LINE
From Point : ( start coordinate 1)
To Point   : (coordinate 2)
To Point   : (coordinate 3)
To Point   : (coordinate 4)
To Point   : (coordinate 5)
To Point   : (coordinate 6)
To Point   : (coordinate 7)
To Point   :  ENTER

Command : LINE
From Point : (coordinate 8)
To Point   : (coordinate 9)
To Point   :  ENTER

Command : ARC
Center/<Start Point> : (coordinate 9) (จุดเริ่มต้นของส่วนโค้ง)
Center/End/<Second Point> :(coordinate1) (จุดบังกลางของส่วนโค้ง)
Angle /Lenght of chord/<End Point> : (coordinate 8) (จุดสิ้นสุด)
```

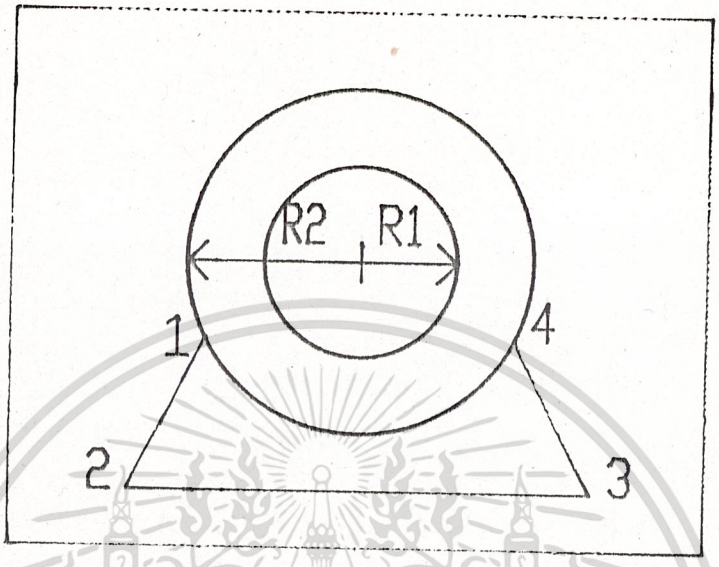
เมื่อเสร็จสิ้นคำสั่งชุดนี้แล้วเราจะได้ภาพวาล์วดังรูป 3.10 ต่อไปเป็นการเก็บภาพเป็น Block โดยใช้คำสั่ง WBlock ซึ่งเมื่อสร้างเป็น Block แล้วเราสามารถนำภาพวาล์วนี้ไปแทรกที่จุดใดของภาพก็ได้ตามต้องการ โดยไม่ต้องวาดภาพวาล์วอีก เพียงแต่ใช้คำสั่ง INSERT รูปแบบคำสั่งในการสร้าง Block มีดังนี้

```
Command : WBLOCK
File name : VALVE (ให้ชื่อไฟล์)
Block name : VALVE (ให้ชื่อบล็อก)
```

การวาด และการสร้าง BLOCK ปั๊ม : ในการวาดภาพปั๊มเราจะวาดภาพปั๊มขึ้นมาตัวเดียวแล้วทำการเก็บภาพปั๊มนั้นเป็น Block โดยใช้คำสั่ง WBlock ในขั้นแรกต้องโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

Drawing Editer ของ AutoCAD ขึ้นมาก่อนโดยตั้งชื่อแบบแปลนชื่อว่า PUMP เมื่อกำหนดเรียบร้อยทำการวาดดังรูป



รูป 3.11 แสดงรูปปั๊ม

มีการใช้ชุดคำสั่งดังนี้

Command : CIRCLE

3P/2P/<Center Point> : (ให้ค่าจุดศูนย์กลาง)

Diameter/<Radius> : (ให้ค่ารัศมีวงกลมวงใน) ENTER

Command : CIRCLE

3P/2P/<Center Point> : (ให้ค่าจุดศูนย์กลาง)

Diameter/<Radius> : (ให้ค่ารัศมีวงกลมวงนอก) ENTER

ใช้คำสั่ง line ในการวาดฐานปั๊ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

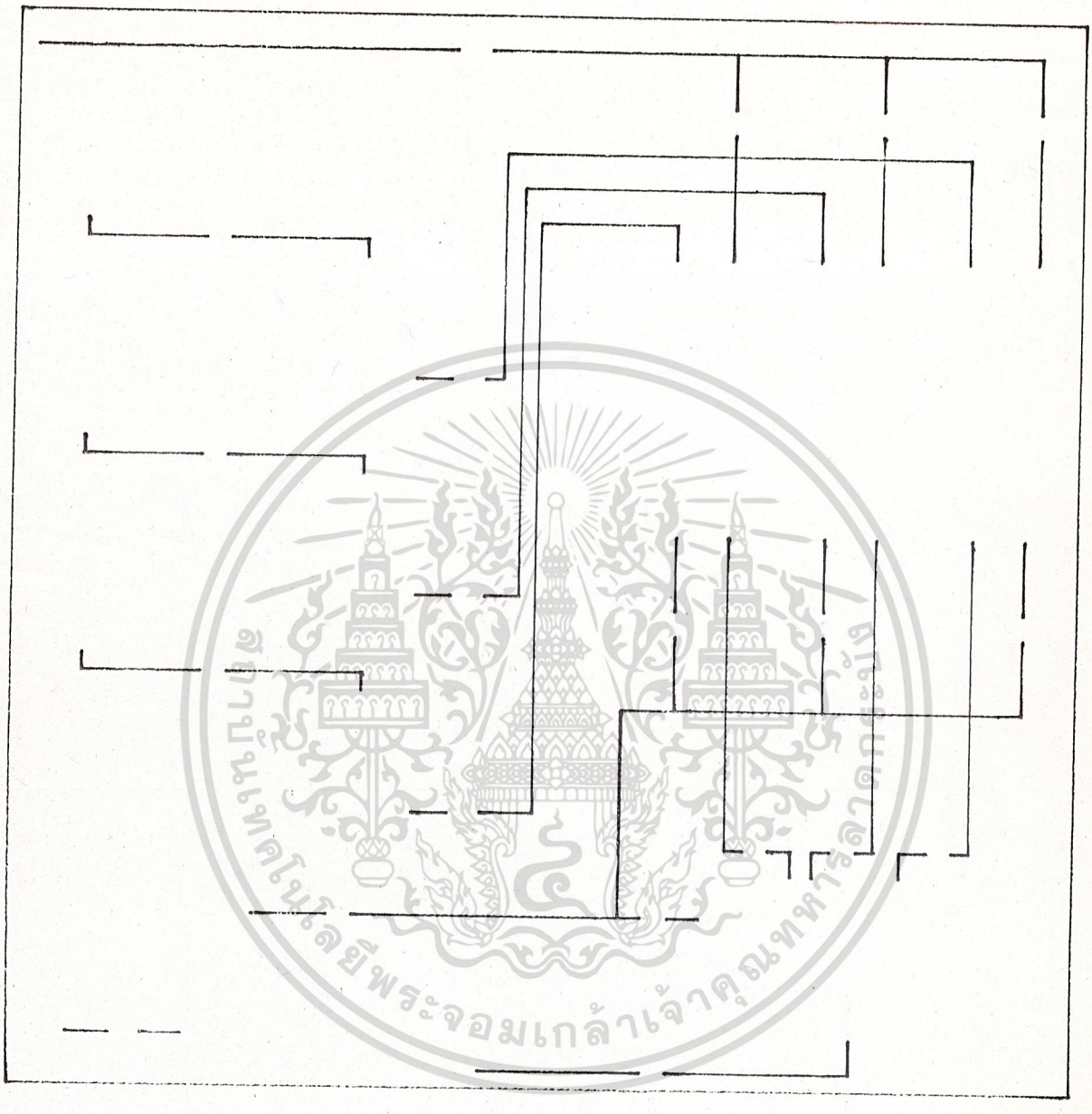
```
Command : LINE
From Point : (start coordinate 1)
To Point : (coordinate 2)
To Point : (coordinate 3)
To Point : (coordinate 4)
To Point : ENTER
```

เมื่อสิ้นสุดชุดคำสั่งดังข้างบน เราจะได้รูปปั๊มดังรูป 3.11 ต่อไปจะเป็นการเก็บภาพปั๊มเป็น Block โดยใช้คำสั่ง WBlock ซึ่งเมื่อเราเก็บภาพปั๊มเป็น Block แล้วเราสามารถที่จะนำภาพปั๊มนี้ไปแทรกที่จุดใดของภาพก็ได้ตามต้องการ โดยไม่ต้องทำการวาดปั๊มอีก เพียงแต่ใช้คำสั่ง INSERT รูปแบบในการสร้าง Block มีดังนี้

```
Command : WBLOCK
File name : PUMP (ให้ชื่อไฟล์)
Block name : PUMP (ให้ชื่อบล๊อก)
```

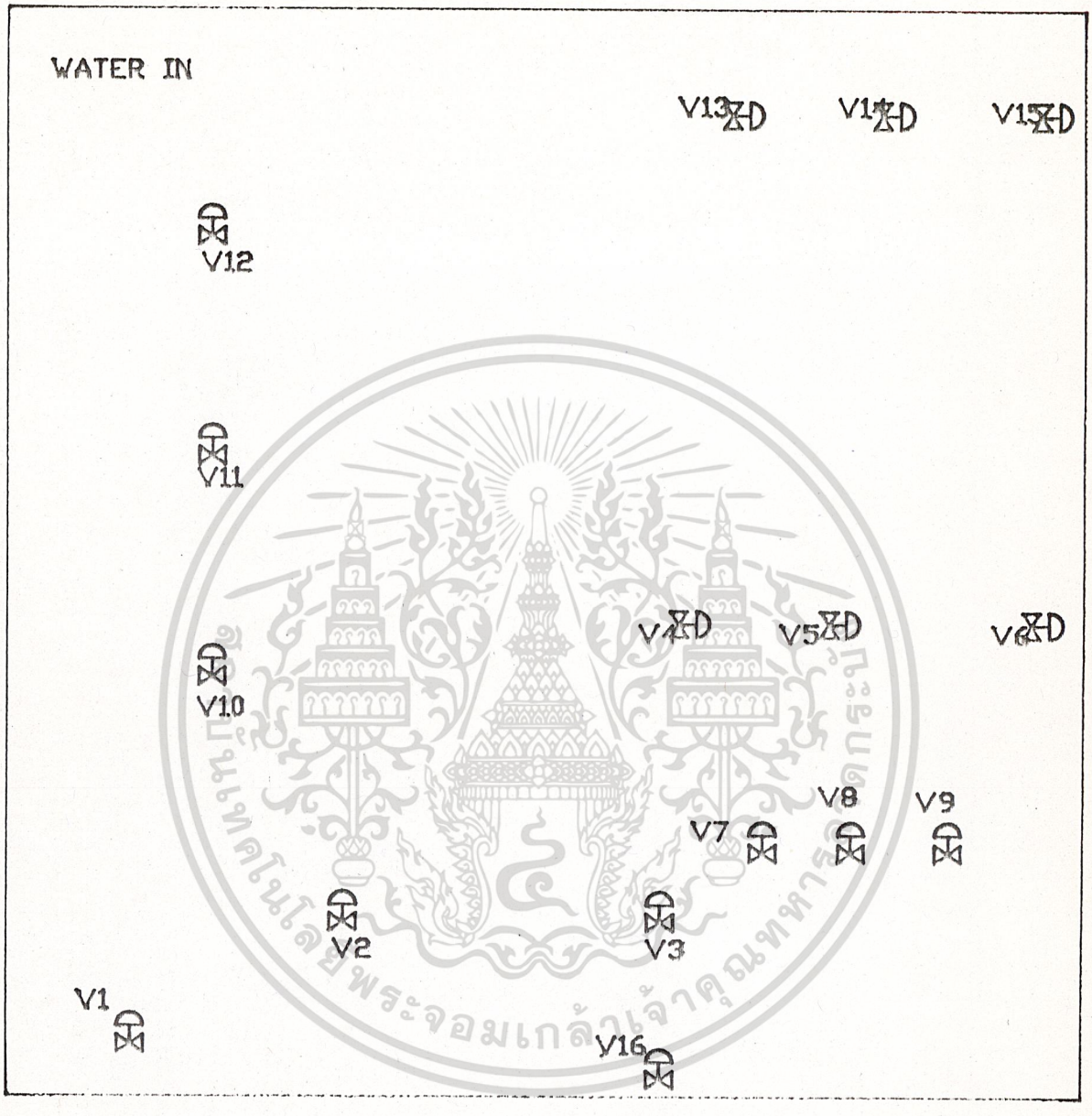
เมื่อสิ้นสุดคำสั่งนี้แล้วแผ่นดิสค์ที่อยู่ในไดร์ฟที่ใช้งานอยู่จะบันทึกไฟล์ PUMP.DWG ลงในดิสค์ นั่นคือ บล๊อกชื่อ PUMP ได้ถูกสร้างขึ้นใช้งานโดยเสร็จสมบูรณ์แล้ว เราสามารถเรียกใช้งานได้ เมื่อต้องการวาดปั๊มดังรูป 3.11 เราเพียงแต่ใช้คำสั่ง INSERT เรียกภาพปั๊มออกมาแล้วนำไปแทรกในแบบแปลนยังจุดที่ต้องการ ซึ่งสามารถจะหมุนบล๊อกให้อยู่ในลักษณะใดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



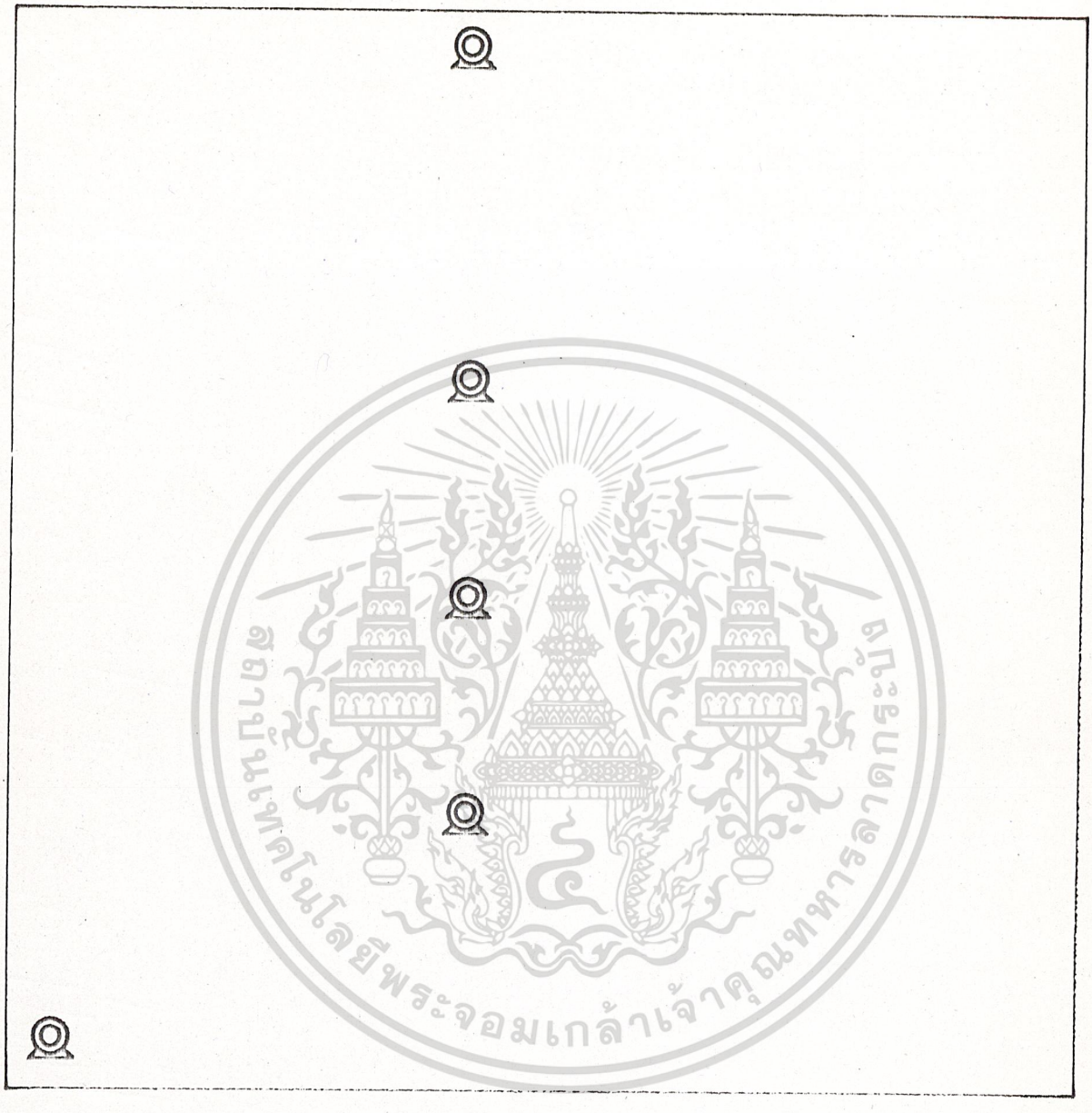
รูป 3.12 แสดงภาพท่อ(PIPE)ทั้งหมด 34 ท่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



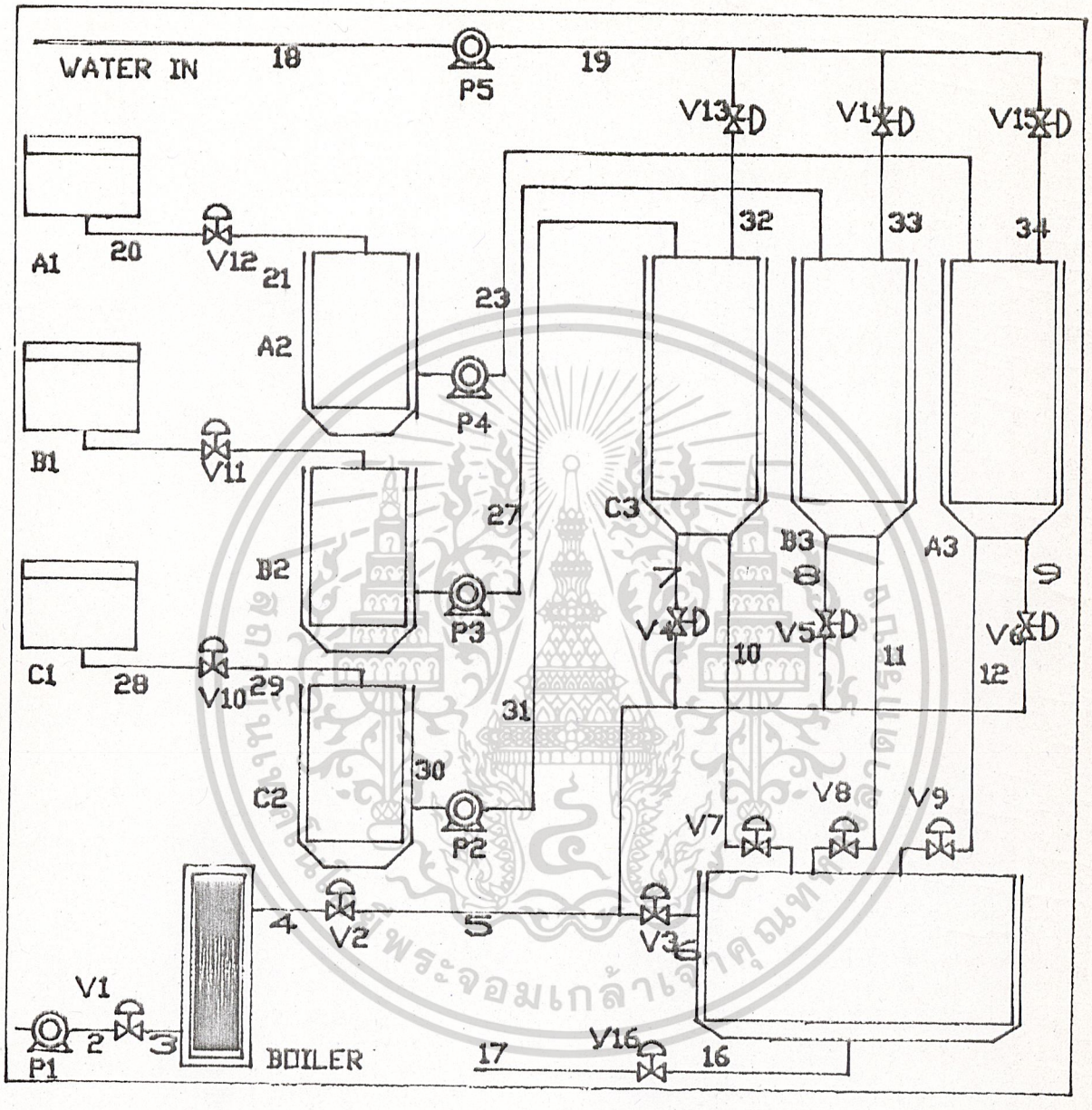
รูป 3.13 แสดงภาพวาล์วทั้งหมด 16 วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.14 แสดงภาพปั๊ม(PUMP)ทั้งหมด

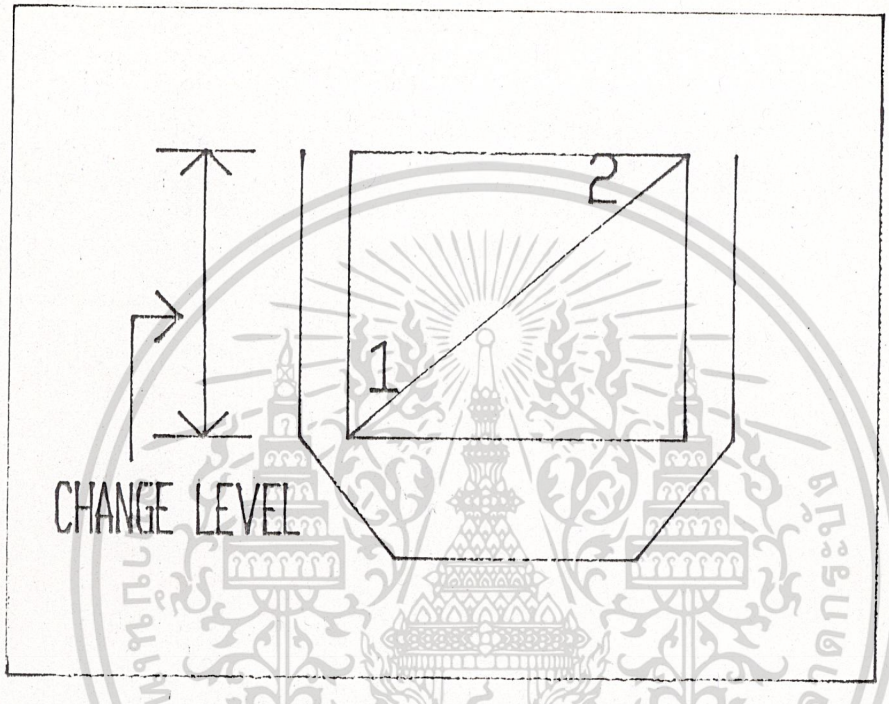
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.15 แสดงภาพถัง(TANK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การกำหนดขอบเขตการเปลี่ยนแปลงระดับของเหลว : อุปกรณ์ที่เปลี่ยนแปลงโดยการเปลี่ยนระดับได้แก่ ถัง เราจะกำหนดให้การเปลี่ยนแปลงระดับอยู่ในชั้นเดียวกับถัง คือชั้น TANK การกำหนดขอบเขตของระดับทำได้โดยลากเส้นจากก้นถัง ให้ถึงปากขอบถัง ดังรูป



รูป 3.16 แสดงการกำหนดขอบเขตการเปลี่ยนระดับ

3.4 การเก็บภาพลงแผ่นจานแม่เหล็ก : ขั้นตอนสุดท้ายของการสร้างภาพคือการเก็บภาพลงแผ่นจานแม่เหล็ก ให้เก็บภาพทั้งหมดทุกชั้นลงในไฟล์เดียวกันหมด โดยเมื่อจะออกจาก Drawing Editer ให้ใช้คำสั่ง END ภาพจะถูกบันทึกลงเป็นไฟล์ .DWG

3.5 รูปแบบ และโครงสร้างของ DRAWING INTERCHANGE FILES : ใน DRAWING EDITOR ของ Autocad สามารถสร้างภาพให้เสร็จสมบูรณ์ แล้วเปลี่ยนไฟล์ภาพที่เราวาดนั้น ให้อยู่ในรูปไฟล์มาตราฐาน(DXF FILE) ซึ่งไฟล์นี้จะเก็บรายละเอียดทุกอย่างเกี่ยวกับภาพที่เราวาดนั้น ซึ่งประกอบไปด้วยค่าพารามิเตอร์(Parameter) และโค๊ต(Code)มากมายรวมกันเป็นไฟล์ ดังนั้น การที่จะเปลี่ยนภาพจากภาพของ AutoCAD มาเป็นภาพที่วาดโดยเทอร์โบ ปาสคาลจึงต้องรู้จักโครงสร้างและรายละเอียดของ Drawing Interchange File ซึ่งส่วนประกอบต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 โครงสร้างโดยทั่วไป : Drawing Interchange File คือ ไฟล์ที่เป็น ASCII text ที่มีชนิดของไฟล์เป็น ".DXF" ซึ่งไฟล์นี้จะรวบรวมรายละเอียดในการสร้างภาพโดยโปรแกรม AutoCAD ประกอบขึ้นเป็นไฟล์นี้ โครงสร้างของไฟล์โดยทั่วไปประกอบด้วยส่วนต่างๆ ดังนี้

1. HEADER section -- ส่วนนี้จะบรรจุรายละเอียดทั่วไป ซึ่ง Parameter จะประกอบด้วย ชื่อของตัวแปร และรวมถึงค่าของตัวแปรนั้น เช่น รายละเอียดในการกำหนดขอบเขตของการแสดงภาพ(LIMITS) เป็นต้น

2. TABLES section -- ส่วนนี้จะบรรจุ Definition ของชื่อรายการต่างๆที่ใช้ในการกำหนดแบบของการวาด เช่น

- Linetype (LTYPE) table
- Layer table
- Style table
- view table

3. BLOCKS section -- ส่วนนี้จะเป็นส่วนที่บรรจุรายละเอียดในการสร้าง Block ว่ามีจุด Base Point อยู่ที่ coordinate เท่าใด Block นั้นประกอบด้วยคำสั่งในการสร้างมีการใช้คำสั่งใดบ้าง เช่น Block Valve จะประกอบด้วยคำสั่ง LINE กับ ARC เป็นต้น

4. ENTITIES section -- ส่วนนี้จะบรรจุรายละเอียดในการสร้างภาพที่มีอยู่ทั้งหมด เช่น คำสั่งที่วาดภาพทั้งหมดว่ามีคำสั่งใดบ้าง รวมถึง มีการเรียกใช้ Block ใด แต่กรณีการเรียกใช้ Block ในส่วนนี้จะบอกเฉพาะจุดที่นำ Block ไปแทรกสอด(Insert Point)ในแบบแปลนเท่านั้น

5. END OF FILE -- บรรทัดสุดท้ายของไฟล์.DXF จะบอกว่าไฟล์นี้สิ้นสุดแล้ว

DXF FILE เป็นไฟล์ที่ประกอบ Groups มากมาย แต่ละ Groups จะเกิดจาก 2 Line , Line ที่ 1 จะเป็นกลุ่มของรหัส(Group Code) ซึ่งเป็นตัวเลขแบบจำนวนเต็มบวกอยู่ในรูปของ FORTRAN "I3" Line ที่ 2 จะเป็นกลุ่มของค่า(Group Value) ในรูปแบบซึ่งขึ้นอยู่กับชนิดของGroups ที่ถูกกำหนดโดยกลุ่มของรหัส

ในการกำหนดกลุ่มของรหัสจะขึ้นอยู่กับ item ภายในไฟล์นั้น อย่างไรก็ตามชนิดของค่าที่อยู่ในGroups จะได้มาจากกลุ่มของรหัสตามรูปแบบที่กำหนดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Group Code Range	Following Value
0 - 9	String
10 - 59	Floating Point
60 - 79	Integer

ดังนั้นโปรแกรมของเราจะสามารถอ่านค่าตาม Group Code ได้ง่ายโดยไม่ต้องรู้ถึงการใช้งานโดยเฉพาะของ Group ใน item ที่บรรจุอยู่ในไฟล์ ค่าที่ปรากฏขึ้นใน DXF FILE จะไม่กระทบกระเทือนถึงการเซตค่าต่างๆที่เซตโดยคำสั่ง UNITS : ค่า Coordinate จะแสดงในรูปแบบของเลขฐานสิบ(Decimal Number) เสมอ ส่วนค่าของมุมแสดงในรูปแบบเลขฐานสิบเป็นองศา โดยที่ 0 องศาจะอยู่ทางทิศตะวันออกของจุดเริ่มต้น(Origin)

3.5.2 GROUP CODE : กลุ่มของรหัสนี้จะเป็นตัวชี้บอกถึงชนิดของค่าของ Group นั้นๆ ที่จะแสดงต่อไปนี้จะเป็นกลุ่มของรหัสที่จะชี้บอกค่าต่างๆที่ถูกกำหนดขึ้นที่จะต้องมีรูปแบบดังต่อไปนี้

0	Identitifies the start of an entity ,table entry,or file seperator.The text value thatfollows indicates which
1	The primary text value for entity
2	A name; attribute tag, Block name, etc.
3 - 5	Other textual or name value.
6	Line type name (fixed).
7	Text style name (fixed).
8	Layer name (fixed).
9	Variable name identifier (used only in HEADER section of DXF File).
10	Primary X coordinate (start point of a line or Text entity,center of a circle,etc.).
11 - 18	Other X coordinate.
20	Primary Y coordinate.
21 - 28	Other Y coordinate.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

30 - 38 Reserved for Z coordinate.

40 - 48 Floating point value.

49 Repeated value -- multiple 49 group may appear in one entity for variable length tables. (such as the dash length in the LTYPE table).A 7x group always appear befor the first 49 group to specify the table length.

50 - 58 Angles.

62 Color number (fixed).

66 "Entities follow" flag (fixed).

70 - 78 Integer value,such as repeat counts, flag bits, or mode .

DXF FILE จะแบ่งออกเป็น 4 ส่วน โดยแต่ละส่วนแบ่งโดยมีคำว่า section เป็นตัวบอกว่าขอบเขตเริ่มต้น และขอบเขตสิ้นสุด ซึ่งมีรูปแบบมาตราฐานเป็นรูปแบบดังนี้

```
  0                          (Begin HEADER section)
SECTION
  2
HEADER
        <<Header variable items go here>>
  0
ENDSEC                       (End Header section)
  0                          (Begin TABLES section)
SECTION
  2
TABLES
  0
TABLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  2
LTYPE
  70
  (Line type table maximum item count)
      <<Line type table items go here>>
  0
ENDTAB
  0
TABLE
  2
LAYER
  70
  (Layer table maximun items count)
      <<Layer table items go here>>
  0
ENDTAB
  0
TABLE
  2
STYLE
  70
  (Text style table maximum items count)
      <<Text style table items go here>>
  0
ENDTAB
  0
TABLE
  2
VIEW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  70
  (View table maximum items count)
      <<View table items go here>>
  0
ENDTAB
  0
ENDSEC                          (End TABLES section)
  0                             (Begin BLOCKS section)
SECTION
  2
BLOCKS
      <<Block definition entities go here>>
  0
ENDSEC                          (End BLOCKS section)
  0                             (Begin ENTITIES section)
SECTION
  2
ENTITIES
      <<Drawing entities go here>>
  0
ENDSEC                          (End ENTITIES section)
  0
END OF FILE                     (END OF FILE)
```

เราได้เห็นรูปแบบของ Drawing Interchange File แล้วด้งแสดงข้างบน ซึ่งได้มาจากการสร้างภาพของ AUTOCAD แล้วใช้คำสั่งสร้าง Drawing Interchange File คือ คำสั่ง DXFOUT: ต่อไปจะแสดงถึง Drawing Interchange File ซึ่งเราได้สร้างขึ้นในปริญญานิพนธ์ฉบับนี้ จะแสดงโดยย่อเพราะเป็นไฟล์ที่ยาวมาก แต่จะแสดงให้เห็นในแต่ละ Section ครบทุก Section ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  0                                    (เริ่ม HEADER SECTION)
SECTION
  2
HEADER
  9
$ACADVER
  1
AC1003
  9
$INSBASE
 10
0
 20
0
  9
$EXTMIN
 10
-40
 20
-40
  9
$EXTMAX
 10
  .
  .
  .
$USERR4
 40
0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  9
 $USERR5
  40
 0
  0
 ENDSEC                    (สิ้นสุด HEADER SECTION)
  0
 SECTION                   (เริ่ม TABLES SECTION)
  2
 TABLES
  0
 TABLE
  2
 LTYPE
 70
     1
  0
 LTYPE
  2
 CONTINUOUS
 70
    64
  3
Solid line
 72
    65
 73
     0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STANDARD

70

0

40

500

41

150

50

0

71

0

42

500

3

tkL

4

0

ENDTAB

0

TABLE

2

VIEW

70

0

0

ENDTAB

0

ENDSEC

(สิ้นสุด TABLES SECTION)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  0                                    (เริ่ม BLOCKS SECTION)
SECTION
  2
BLOCKS
  0
BLOCK
  8
F
 62
     5
  2
VALVE
 70
    64
 10
7250
 20
22000

  0
LINE
  8
V
 62
    7
 10
6075
 20
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  0
18625
  0
LINE
  8
P


 62
    7
 10
14125
 20
18875
 11
14375
 21
18625
  0

ENDBLK
  8
PUMP5
 62
    5
  0
ENDSEC                    (สิ้นสุด BLOCKS SECTION)
  0                       (เริ่ม ENTITIES SECTION)
SECTION
  2
ENTITIES
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0

LINE

8

T

10

250

20

27250

11

3500

21

27250

0

LINE

9

T

10

27250

11

3500

21

25000

0

LINE

9

T

10

LINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8

T

10

250

20

25000

11

250

21

27250

0

LINE

.

.

.

.

5

11

20750

21

750

0

TEXT

8

PIPE17

62

7

10

19625

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  20
 750
  40
 500
   1
 17
  72
     5
  11
14250
  21
750
  0
ENDSEC                    (สิ้นสุด ENTITIES SECTION)
  0
EOF                       (สิ้นสุดไฟล์)
```

3.5.3 HEADER SECTION : ส่วนของ HEADER ของไฟล์ DXF จะบรรจุการเซตค่าของตัวแปรที่รวมอยู่ในการวาดภาพ ตัวแปรเหล่านี้จะเซตค่าตามคำสั่งในการแสดงภาพโดย STATUS Command : แต่ละตัวแปรที่กำหนดใน HEADER Section นี้ มี 9 Group ตามชื่อ และตาม Group นั้นๆก็จะมีค่าอยู่ ซึ่งตัวแปรใน HEADER จะมี Group และมีความหมายตาม table ดังต่อไปนี้

$ACADVER	1 (the AutoCAD version number)
$ATTMODE	70 (Attribute visibility: 0=none, 1=normal, 2=all).
$AUNITS	70 (Units format for angle)
$AUPREC	70 (Units precision for angle)
$AXISMODE	70 (Axis on if nonzero)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$AXISUNIT	10 and 20 (axis X and Y tick spacing)
$CLAYER	8 (current layer name)
$DIMASZ	40 (dimensioning arrow size)
$DIMCEN	40 (size of center mark/line)
$DIMDLI	40 (dimension line increment)
$DIMEXE	40 (extension line extension)
$DIMEXO	40 (extension line offset)
$DIMLIM	70 (dimension limits generate if nonzero)
$DIMSCALE	40 (overall dimensioning scale factor)
$DIMSE1	70 (first extension line suppressed if nonzero)
$DIMSE2	70 (second extension line suppressed if nonzero)
$DIMTAD	70 (text above dimension line if nonzero)
$DIMTIH	70 (text inside horizontal if nonzero)
$DIMTM	40 (minus tolerance)
$DIMTOH	70 (text outside horizontal if nonzero)
$DIMTOL	70 (dimension tolerances generated if nonzero)
$DIMTP	40 (plus tolerance)
$DIMTSZ	40 (dimensioning tick size: 0=no ticks)
$DIMTXT	40 (dimensioning text height)
$DRAGMODE	70 (DRAGMODE on if nonzero)
$EXTMAX	10 and 20 (drawing extents upperright corner)
$EXTMIN	10 and 20 (drawing extents lower left corner)
$FILLETRAD	40 (fillet radius)
$FILLMODE	70 (FILL mode on if nonzero)
$GRIDMODE	70 (GRID mode on if nonzero)
$GRIDUNIT	10 and 20 (grid X and Y spacing)
$INSBASE	10 and 20 (insertion base setby BASE Command)
$LIMMAX	10 and 20 (drawing limits upper right corner)
$LIMMIN	10 and 20 (drawing limits lower left corner)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$LTSCALE	40 (global linetype scale)
$LUNITS	70 (UNITS format for coordinate and distance)
$LUPREC	70 (UNITS precision for coordinate and distance)
$OTHERMODE	70 (OTHER mode on if nonzero)
$OSMODE	70 (running object snap mode)
$QTEXTMODE	70 (quick text mode on if nonzero)
$REGENMODE	70 (REGENAUTO mode on if nonzero)
$SKETCHINC	40 (sketch record increment)
$SNAPANG	50 (snap grid rotation angle)
$SNAPBASE	10 and 20 (snap grid base point)
$SNAPISOPAIR	70 (isometric plane: 0=left, 1=top, 2=right)
$SNAPMODE	70 (snap mode on if nonzero)
$SNAPSTYLE	70 (snap style: 0=standard, 1=isometric)
$SNAPUNIT	10 and 20 (snap grid X and Y spacing)
$TEXTSIZE	40 (default text hight)
$TEXTSTYLE	7 (current text style name)
$TRACEWID	40 (default trace width)
$VIEWCET	10 and 20 (center of current view on screen)
$VIEWSIZE	40 (hight of current view on screen)

3.5.4 TABLES SECTION : TABLES SECTION ของ DXF FILE จะประกอบด้วย 4 TABLE ย่อย TABLES SECTION นี้จะเริ่มต้นด้วย 0 group แล้วบรรทัดต่อมาจะตามด้วยลาเบล(label) "TABLE" แต่ละTABLE ย่อยจะตามด้วย 2 group บรรทัดต่อมาจะเป็นชื่อ TABLE คือ "LTYPE","LAYER","STYLE" หรือ "VIEW" และ 70 group จะกำหนดจำนวนสูงสุดของ table entries ที่จะบรรจุลงไปในแต่ละ table ย่อย ใน table ที่แสดงนี้จะมีรายละเอียดในการลบภาพขณะที่เราเขียนภาพ หรือ มีการลบ items ต่างๆ แต่จะไม่ถูกนำมาเขียนลงใน DXF FILE เมื่อเป็นเช่นนี้ จำนวน items ที่นำมาแสดงใน table header จะมีจำนวนน้อยกว่าจำนวนที่แสดงไว้โดย 70 group ดังนั้นเราจะไม่ใช้จำนวนตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

70 group เป็นตัวชี้(index) ในการอ่าน table. การเตรียมโปรแกรมสำหรับอ่านส่วนนี้สามารถใช้ Array ในการอ้างถึงตาม table items นั้น

สำหรับส่วนหัวของแต่ละ table จะเริ่มต้นด้วย 0 group เป็นตัวบอกชนิดของ table (e.g.,"LTYPE" หรือ "LAYER") 2 group จะเป็นตัวบอกชื่อตลอด table และ 70 group จะกำหนด flags relevant ตัวที่ชี้บอกจบ table จะบอกโดย 0 group แล้วตามด้วยลาเบล "ENDTAB".

Group ที่ใช้สำหรับแต่ละชนิดของ table items ทุก Group จะแสดงแต่ละ table itms ดังนี้

LTYPE — 3 (prose description of linetype), 72 (alignment code), 73 (number of dash length items), 40 (total pattern length), 49 (dash length 1), 49 (dash leng 2),...

LAYER — 62 (color number, 6 (linetype name).

STYLE — 40 (fixed text height; 0 if not fixed),50 (obliguing angle), 71 (text generation flags), 42 (last height used),3 (text font of shape file name).

VIEW — 40 (view vertical height), 10 and 20 (view center point).

3.5.5 BLOCKS SECTION : ส่วนนี้จะบรรจุรายละเอียดการสร้าง Block ตลอด Section รูปแบบของ Section Block จะใช้อ้างถึงในส่วนต่อไปคือ ส่วนของ Entities Section ค่าต่างๆใน BLOCKS SECTION นี้จะปรากฏอยู่ระหว่างลาเบล "BLOCK" กับ "ENDBLK".

3.5.6 ENTITIES SECTION : ค่าต่างๆ ที่ปรากฏอยู่ในส่วนของ BLOCKS กับ ENTITIES ของไฟล์ DXF คือค่าที่บอกถึงรูปร่างลักษณะของภาพที่วาดทั้งหมด ไม่ว่าจะเป็น

เอกสารนี้สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ทุกคำสั่ง จุด Insert point เมื่อเรียกใช้ Block ซึ่งแต่ละตัวแปร หรือแต่ละคำสั่งจะเริ่มต้นด้วย 0 group และทุกๆ entities ที่บรรจุอยู่ในส่วนนี้จะมี 8 group เป็นตัวชี้บอก Layer ที่จะแสดงต่อไปนี้เป็นคำสั่งต่างๆที่ใช้ขณะวาดภาพ เมื่อสร้าง Drawing Interchange File เราจะได้รายละเอียด และ group ต่างๆ ดังที่จะกำหนดในรูปแบบดังต่อไปนี้

LINE	10 and 20 (start point),11 and 21 (end point)
POINT	10 and 20
CIRCLE	10 and 20 (center), 40 (radius)
ARC	10 and 20 (center), 40 (radius), 50 (start angle), 51 (end angle)
TRACE	Four point defining the corner of the trace: 10 and 20 ,11 and 21, 12 and 22, 13 and 23.
SOLID	Four point defining the corner of the solid: 10 and 20 ,11 and 21, 12 and 22, 13 and 23. if solid only has 3 sides, the coordinate specified by the 12 and 22,and 13 and 23.
REPEAT	no group
ENDREP	70 (number of columns), 71 (number of rows), 40 (columns spacing), 41 row spacing.
TEXT	10 and 20 (insert point), 40 (height), 1 (text value),50 (rotate angle-optional 0),

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

41 (relative X scale factor-option 1), 51(obliguing angle-option 0), 7 (text style name-option "standard"),71 (generation flags -option 0), 72 (justification type-option 0), 11 and 21 (alignment point option,appear only if 72 group is present and nonzero).

SHAPE — 10 and 20 (insertion point), 40 (size), 2 (shape name),50 (rotation angle-option 0), 41 (relative X scale factor-option 1), 51 (obliguing angle-option 0).

BLOCK — 2 (block name), 70 (block type flags), 10 and 20 (block base point), appear only in BLOKCS SECTION.

ENDBLK — no group. appear only in BLOKCS SECTION.

INSERT — 66 ("attributed follow" flags-option 0), 2 (block name), 10 and 20 (insertion point), 41 (X scale-option 1), 50 (rotation angle-option 0).

ATTDEF — 10 and 20 (text start), 40 (text height), 1 (default value), 3 (prompt string), 2 (tag string), 70 (attributed flags), 73(field lenght-option 0), 50 (text rotation-option 0), 41 (relative X scale factor-option 1), 51 (obliguing angle-option 0), 7 (text style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

name-option "standard"),71(text generation flags-optional 0), 72 (text justification type-optional 0), 11 and 21 (alignment point optional, appear only if 72 group is present and nonzero)

ATTRIB — 10 and 20 (text start), 40 (text height), 1 (value), 2 (attributed tag), 70 (attributed flags), 73 (field lenght-option 0), 50 (text rotation-option 0), 41 (relative X scale factor-option 1), 51 (obliguing angle -option 0), 7 (text style name-option "standard"), 71 (text generation flags-optional 0), 72 (text justification type -optional 0), 11 and 21 (alignment point optional, appear only if 72 group is present and nonzero)

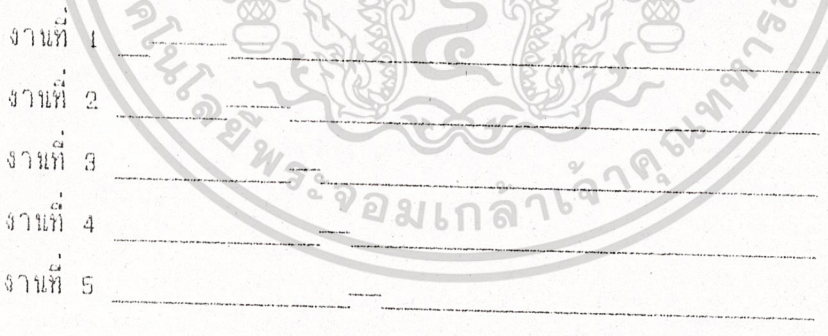
SEQUENCE — No field.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โปรแกรมเงื่อนไขการแสดงผล

โปรแกรมเงื่อนไขการแสดงผลนั้นในการใช้งานจริงแล้วผู้ใช้จะเป็นผู้พัฒนาขึ้นเอง ซึ่งวัตถุประสงค์คือ ต้องการทำให้ระบบมีความยืดหยุ่นสูง ถึงแม้ว่าในขั้นแรกจะต้องใช้เวลามาก ในการศึกษาและพัฒนาโปรแกรม แต่ในระยะยาวแล้วผู้ใช้สามารถพัฒนาโปรแกรมให้ใช้งานได้อย่างมีประสิทธิภาพ โปรแกรมเงื่อนไขการแสดงผลสามารถแบ่งออกเป็นส่วนใหญ่ๆ ได้ 3 ส่วน คือ

4.1 การกำหนดช่วงเวลาในการทำงานของแต่ละงาน : เนื่องจากเวลาส่วนใหญ่จะใช้ในการแสดงผล และการกระทำทางตรรก(LOGIC) ในที่นี้เรามีอุปกรณ์ที่ต้องแสดงผลคือ ปั๊ม,วาล์ว,ท่อ และการเปลี่ยนระดับ ซึ่งในการแสดงผลเราได้ใช้โปรแกรมย่อยเป็นตัวกำหนดสภาวะ คือไม่ต้องรับสภาวะจริง แต่ใช้โปรแกรมกำหนด ซึ่งในการกำหนดงานจะกำหนดให้ทำตามลำดับกันไป โดยเราใช้วิธีการทางซอฟท์แวร์ทั้งหมดคือ การมัลติเพลกซ์ (MULTIPLEXING) จะเป็นการแบ่งช่วงเวลาของโปรแกรมออกเป็นส่วนๆ แล้วกำหนดว่า จะให้งานใดทำในช่วงเวลาใด ดังรูป 4.1



รูป 4.1 แสดงการมัลติเพลกซ์งานหลายงาน

จากรูป จะเห็นว่าการทำมัลติเพลกซ์ จะสามารถกำหนดความถี่ และช่วงเวลาของการทำงานแต่ละงานได้ ในที่นี้ให้มีงานทั้งหมด 5 งาน

งานที่ 1 : เก็บค่าสภาวะเก่าของกระบวนการ

งานที่ 2 : อ่านค่าสภาวะของกระบวนการ

งานที่ 3 : แสดงสภาวะอุปกรณ์ประเภทที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 4 : แสดงสภาวะอุปกรณ์ประเภทที่ 2

งานที่ 5 : แสดงสภาวะอุปกรณ์ประเภทที่ 3

งานทั้งหมดมีความสำคัญตามลำดับ วิธีการในขั้นแรก จัดการทำงานของแต่ละงานเป็นลำดับไป เช่น งานที่ 1,2,3,4 และ 5 จะต้องทำงานเป็นลำดับกันไป โดยงานที่ 3,4,5 สามารถที่จะทำสลับกันได้ แต่จะต้องมีการทำงานที่ 1,2 ก่อนเสมอ เนื่องจากงานที่ 1 และ 2 เป็นการเก็บค่าสภาวะเก่า และการรับค่าสภาวะของกระบวนการ เพราะฉะนั้นข้อมูลจะต้องถูกปรับปรุงใหม่(UPDATE) ก่อนแล้วจึงจะนำไปแสดงผล มิฉะนั้นข้อมูลที่นำไปแสดงผลจะเป็นข้อมูลเก่า ซึ่งช้ากว่าความเป็นจริงอยู่ 1 รอบการทำงาน ดังตัวอย่างเป็นการสมมติให้งานทุกงานถูกจัดลำดับมาถูกต้องแล้ว

การกำหนดความถี่ในการทำงานของแต่ละงาน หลักการกำหนดความถี่ส่วนใหญ่แล้วงานจะถูกทำเป็นลำดับ เช่นงานที่ 1,2,3,4 และ 5 จะมีความถี่ในการทำงานเท่ากัน การกำหนดให้งานใดงานหนึ่งทำงานมากกว่างานอื่น ก็ไม่มีผลต่อการเปลี่ยนแปลง ทำให้เสียเวลาไปโดยเปล่าประโยชน์ ถ้ากำหนดให้งาน 3,4 และ 5 ทำงานมากกว่างานอื่น ก็จะพบว่า ไม่มีการเปลี่ยนแปลงใดๆเลย เนื่องจากข้อมูลไม่มีการปรับปรุงใหม่นั่นเอง ส่วนการที่จะให้งาน 1 และงาน 2 คือการเก็บค่าสภาวะเก่า และรับค่าสภาวะของกระบวนการทำงานมากกว่างานอื่นก็จะเป็นการปรับปรุงข้อมูลซ้ำซ้อนกัน ซึ่งข้อมูลที่รับเข้ามาครั้งแรก ยังไม่ถูกนำไปแสดงผลด้วยงานที่ 3,4 และ 5 ดังนั้น จะเห็นได้ว่า การกำหนดความถี่ของการทำงาน จะสัมพันธ์กับการกำหนดช่วงเวลาด้วย

ตัวอย่าง การกำหนดช่วงเวลา ความถี่ และการทำงานของการมัลติเพลกซ์ก็จะแบ่งได้ดังนี้

- **การกำหนดช่วงความถี่ และ ช่วงเวลา** : จะกำหนดด้วยเลขฐานสอง (BINARY NUMBER) ตามช่วงเวลา เช่น ถ้ากำหนดช่วงเวลาต่อ 1 รอบเป็น 8 ช่วงเวลา ก็จะใช้เพียง 1 ไบท์ แต่ถ้ากำหนดให้เป็น 16 ช่วงเวลา ก็จะใช้ 2 ไบท์ เป็นต้น โดยตั้งค่าคงที่(CONSTANT) ตามขนาดดังกล่าว ดังตัวอย่าง

```
const                {...:...:...:...:}
  Mask1  = $0101; {_______ _______}
  Mask2  = $FFFF; {________________}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Mask3  = $0404; {_____-_____-__}
Mask4  = $0808; {____-_____-___}
Mask5  = $1010; {___-_____-____}
Mask6  = $2020; {__-_____-_____}
Mask7  = $4040; {_-_____-_____}
Mask8  = $8080; {-_____-_____}
Mask9  = $0101; {_____-_____-}
Mask10 = $FFFF; {----------------}
Mask11 = $0404; {_-_____-_____}
Mask12 = $0808; {__-_____-_____}
Mask13 = $1010; {___-_____-____}
Mask14 = $2020; {____-_____-___}
Mask15 = $4040; {_____-_____-__}
Mask16 = $8080; {______-_____-_}
```

จะเห็นได้ว่า การกำหนดจะใช้เลขฐาน 16 (HEXADECIMAL NUMBER) แต่การออกแบบจะถูกกำหนดมาจากเลขฐาน 2 เสียก่อน เช่นงานที่ 1 ใช้ชื่อว่า Mask1 จะมีช่วงเวลาเป็น 1000 0000 1000 0000 เป็นต้น

- การเลื่อนลำดับของช่วงเวลา : จะตั้งตัวแปร (VARIABLE) ที่มีขนาดเท่ากับค่าคงที่ดังกล่าวขึ้นมา ในที่นี้ใช้ตัวแปรชื่อ Sequence ซึ่งค่าใน Sequence จะเป็นตัวบอกช่วงเวลา การเลื่อนลำดับของช่วงเวลากระทำได้โดยเลื่อนค่าใน Sequence ไป 1 ตำแหน่ง ดังตัวอย่าง

```
procedure RotateSequence;
begin {RotateSequence}
  Sequence := Sequence shl 1;
  If Sequence = $0000 then Sequence := $0001;
end {RotateSequence};
```

เนื่องจากในภาษาปาสคาลไม่มีคำสั่งหมุนข้อมูลจึงใช้วิธีเลื่อนข้อมูล แล้วตั้งค่าใหม่เมื่อเลื่อนจนสุดความยาวแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตรวจสอบลำดับของงาน : เป็นการตรวจสอบลำดับของช่วงเวลาเปรียบเทียบกับลำดับของงานที่กำหนด ดังตัวอย่าง

```
begin
   if (Sequence and Mask1 = Sequence) then Task1;
   if (Sequence and Mask2 = Sequence) then Task2;
   if (Sequence and Mask3 = Sequence) then Task3;
   if (Sequence and Mask4 = Sequence) then Task4;
   if (Sequence and Mask5 = Sequence) then Task5;
   if (Sequence and Mask6 = Sequence) then Task6;
   if (Sequence and Mask7 = Sequence) then Task7;
   if (Sequence and Mask8 = Sequence) then Task8;
   if (Sequence and Mask9 = Sequence) then Task9;
   if (Sequence and Mask10 = Sequence) then Task10;
   if (Sequence and Mask11 = Sequence) then Task11;
   if (Sequence and Mask12 = Sequence) then Task12;
   if (Sequence and Mask13 = Sequence) then Task13;
   if (Sequence and Mask14 = Sequence) then Task14;
   if (Sequence and Mask15 = Sequence) then Task15;
   if (Sequence and Mask16 = Sequence) then Task16;
end;
```

4.2 การกำหนดสีของอุปกรณ์ในสภาวะต่างๆ : การกำหนดสีของอุปกรณ์ต่างๆซึ่งโดยปกติแล้วก็จะให้อุปกรณ์แต่ละชนิดมีสีที่แตกต่างกัน เพื่อให้สังเกตได้ง่าย ขอสำคัญอีกประการหนึ่งก็คือ การกำหนดสีในสภาวะที่แตกต่างกัน เช่นอุปกรณ์ที่มีสภาวะเพียง 2 สภาวะคือ เปิดและปิด นั้นก็จะกำหนดให้สีของอุปกรณ์ขณะที่อยู่ในสภาวะทั้งสองให้แตกต่างกัน ส่วนอุปกรณ์ที่มีการรับค่าเป็นอนาลอกนั้น จะกำหนดให้สีของอุปกรณ์จะใช้เพียงสีเดียว เนื่องมาจาก การเปลี่ยนแปลงสามารถสังเกตได้จากระดับ ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HideColor = Black
NormalColor = White
WaterPipeColor = LightBlue
BoilerPipeColor = LightBlue
MixTankPipeColor = LightRed
TankAPipeColor = Green
TankBPipeColor = Red
TankCPipeColor = Yellow
```

จากตัวอย่าง ได้กำหนดให้สีของท่อต่างๆนั้นแตกต่างกันตามชนิดของการใช้งาน โดยที่สีขณะอยู่ในสภาวะปิดจะเป็นสีเดียวกันหมดคือ สีขาว ส่วนระดับของเหลวในถังก็กำหนดให้แตกต่างกัน

4.3 การกำหนดเงื่อนไขการแสดงผล : การกำหนดเงื่อนไขการแสดงผลจะเป็นส่วนสำคัญที่สุดในส่วนนี้ เป็นการกำหนดเงื่อนไขการแสดงผลให้สอดคล้องกับการทำงานของกระบวนการ และเนื่องจากอุปกรณ์บางชนิดไม่มีการอ่าน หรือรับค่าสภาวะจากกระบวนการ เช่นท่อเป็นต้น ดังนั้น การที่จะทำให้ท่อดังกล่าวสามารถแสดงสภาวะได้ว่ามีการไหลในท่อหรือไม่ ก็จะนำสภาวะของวาล์วมาพิจารณาในกรณีที่วาล์วต่ออยู่กับท่อนั้นมีสภาวะเปิดก็อาจสันนิษฐานได้ว่ามีการไหลในท่อนั้น หรืออาจจะมีเงื่อนไขอื่นมาเกี่ยวข้องอีกก็ได้ การสร้างเงื่อนไขต่างๆนั้น เราจะต้องมีการศึกษาการทำงานของกระบวนการ และรูปแบบการแสดงผลเป็นสำคัญ ดังจะขอยกตัวอย่างเพียงบางส่วน ดังนี้

```
procedure DisplayDeviceNumber (Device : byte);
begin {DisplayDeviceNumber}
  case Device of
    1 : begin
          if Valve1Status xor DeviceStatus[1] then
          begin
            if Valve1Status then
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
          Valve1(WaterPipeColor);
          Pipe3(WaterPipeColor);
        end
        else
        begin
          Valve1(NormalColor);
          Pipe3(NormalColor);
        end;
      end;
    end;
2 : begin
      if Valve2Status xor DeviceStatus[2] then
      begin
        if Valve2Status then
        begin
          Valve2(BoilerPipeColor);
          Pipe5(BoilerPipeColor);
        end
        else
        begin
          Valve2(NormalColor);
                    .
                    .
                    .
                    .
                    .
        else
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Pump3(NormalColor);
            Pipe27(NormalColor);
          end;
        end;
      end;
20 : begin
      if Pump4Status xor DeviceStatus[20] then
      begin
        if Pump4Status then
        begin
          Pump4(TankAPipeColor);
          Pipe23(TankAPipeColor);
        end
        else
        begin
          Pump4(NormalColor);
          Pipe23(NormalColor);
        end;
      end;
    end;
21 : begin
      if Pump5Status xor DeviceStatus[21] then
      begin
        if Pump5Status then
        begin
          Pump5(WaterPipeColor);
          Pipe19(WaterPipeColor);
        end
        else
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Pump5(NormalColor);
                Pipe19(NormalColor);
              end;
            end;
          end;
    end;
end {DisplayDeviceNumber};
```

จากตัวอย่างที่ได้ยกมานี้เป็นการทำเงื่อนไขในการแสดงผลของวาล์วและท่อ จะเห็นได้ว่าสภาวะของท่อนั้นจะขึ้นอยู่กับสภาวะของวาล์วโดยตรง ซึ่งในความเป็นจริงแล้วอาจจะมีเงื่อนไขอื่นมาเกี่ยวข้องอีกก็ได้ ส่วนตัวแปรที่นำมาตรวจสอบเงื่อนไขคือ สภาวะของวาล์วนั้นๆ เช่น ValveStatus[1] ก็คือสภาวะของวาล์วตัวที่ 1 ส่วนท่อหมายเลข 3 นั้นเป็นท่อที่ต่ออยู่กับวาล์วตัวที่ 1 ส่วนตัวแปร DeviceStatus นั้น เป็นตัวแปรแบบอาเรย์ (ARRAY) ใช้ในการเก็บสภาวะเก่าของกระบวนการในช่วงเวลาที่ผ่านมา เพื่อไม่ให้ต้องมีการวาดรูปอุปกรณ์ทุกๆ รอบการทำงาน จะวาดก็ต่อเมื่อมีการเปลี่ยนแปลงสภาวะเท่านั้น คือ จะมีการวาดก็ต่อเมื่อมีการเปลี่ยนแปลงสภาวะจากปิด เป็น เปิด หรือจาก เปิด เป็น ปิด เท่านั้น วิธีนี้จะช่วยให้ระบบทำงานได้เร็วขึ้น เนื่องจากไม่ต้องวาดใหม่ทุกครั้ง

เมื่อตรวจสอบพบว่าเกิดการเปลี่ยนแปลงขึ้นและเปลี่ยนไปอย่างไรแล้ว ก็จะเป็นคำสั่งในการวาด จะเห็นว่าวาล์วตัวใดมีสภาวะเปิดก็จะทำการวาดวาล์วตัวนั้นและท่อที่เกี่ยวข้องด้วยสีที่กำหนด ส่วนที่วาล์วมีสภาวะปิด ก็จะใช้สีอีกสีหนึ่งในที่นี้ใช้สีขาว ดังนั้นจึงสามารถสังเกตเห็นการเปลี่ยนแปลงสภาวะของกระบวนการได้อย่างชัดเจน

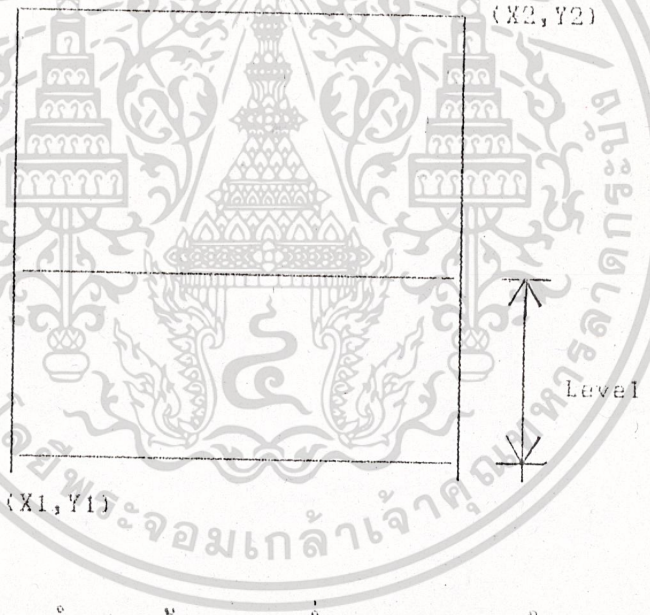
การเปลี่ยนสภาวะโดยการเปลี่ยนระดับนั้น เป็นการเรียกใช้ชื่อของอุปกรณ์ที่เราตั้งชื่อไว้ หรือที่เรากำหนดไว้เมื่อตอนสร้างภาพ ตามด้วยระดับของเหลวในถัง เช่น TankA2(Level); เป็นต้น ดังตัวอย่าง

```
procedure TankA2(var percent : integer);
begin
  setcolor(TankA2Color);
  CurrentLevel:= TankA2Level;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DrawLevel(114,118,145,68,Percent);
    TankA2Level:= CurrentLevel;
end;
```

คำสั่ง Setcolor เป็นการเปลี่ยนสีในการวาดซึ่งจะทำหน้าที่เปลี่ยนสีตามที่ได้กำหนดไว้ ส่วนตัวแปร CurrentLevel และ TankA2Level นั้นมีหน้าที่ตรวจสอบการเปลี่ยนแปลงของระดับ โดยจะทำการวาด เมื่อมีการเปลี่ยนระดับไปจากเดิมเท่านั้น ส่วนคำสั่ง DrawLevel เป็นคำสั่ง (PROCEDURE) ที่สร้างขึ้นสำหรับการวาดระดับโดยมีค่าพารามิเตอร์ที่ต้องการ 5 ตัว คือ X1,Y1,X2,Y2 และระดับที่ต้องการวาด ดังรูป



รูป 4.2 การกำหนดระดับสูงสุดและต่ำสุดของของเหลวในถัง

ค่าพารามิเตอร์ X1,Y1,X2,Y2 จะต้องเป็นค่าที่เปลี่ยนจาก coordinate ของจอภาพ AUTOCAD มาเป็นค่า coordinate ของจอภาพ TURBO PASCAL แล้วเท่านั้น

ตัวอย่างการสร้างโปรแกรมเงื่อนไขการแสดงผล : โปรแกรมเงื่อนไขการแสดงผลจะแตกต่างกันตามกระบวนการ ในที่นี้ใช้กระบวนการที่กล่าวถึงในบทที่ 3

ก่อนอื่นเราต้องสร้างภาพกระบวนการดังแสดงในบทที่ 3 ก่อน แล้วทำการเปลี่ยนภาพกระบวนการนั้นเป็นไฟล์ DXF เมื่อได้ไฟล์ DXF แล้ว เราจึงทำการเปลี่ยนไฟล์ DXF นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเป็นภาพที่วาดโดย TURBO PASCAL การเขียนโปรแกรมเข้าไปอ่านไฟล์ DXF นี้ ขึ้นอยู่กับผู้ใช้ที่จะเลือกออกแบบโปรแกรมของผู้ใช้เอง เพราะภาพกระบวนการที่เสนอไปในบทที่ 3 นั้น เป็นภาพตัวอย่าง ที่ใช้ในการทดลองเท่านั้น เมื่อเราได้ภาพกระบวนการที่วาดโดยโปรแกรมภาษา TURBO APSCAL แล้ว เราจึงออกแบบโปรแกรมเงื่อนไขแสดงผลเป็นภาษา TURBO PASCAL ในขั้นตอนต่อมา

การพัฒนาโปรแกรมเงื่อนไขการแสดงผลนั้นจำเป็นต้องใช้ภาษาปาสคาล และใช้ตัวแปรภาษาเทอร์โบ ปาสคาล เวอร์ชั่น 5.5 (TURBO PASCAL V 5.5) เนื่องจากโปรแกรมส่วนอื่นๆ เป็นภาษาปาสคาล ทั้งหมด

โปรแกรมเงื่อนไขการแสดงผลจะเป็นส่วนหนึ่งของโปรแกรมหลัก เรียกว่ายูนิท (UNIT) การตั้งชื่อยูนิทจะต้องใช้ชื่อว่า uses ถ้ากำหนดเป็นชื่ออื่นจะต้องแก้ไขในโปรแกรมหลักด้วย ส่วนการ uses ยูนิทใดนั้นขึ้นอยู่กับโปรแกรมว่าต้องเรียกใช้ยูนิทใดบ้าง ในตัวอย่างนี้จะต้องเรียกใช้ uses Dos,Crt,Graph,AcadTp,Variable,DataPV,DataPIPE, DrawPVP; ดังนั้น ในตอนต้นของยูนิทนี้จะมีลักษณะดังนี้

```
unit process;

interface

uses Dos,Crt,Graph,AcadTp,Variable,DataPV,DataPIPE,DrawPVP;
```

จากนั้นจะเป็นการกำหนดความถี่และช่วงเวลาในการทำงานของงานต่างๆ แต่ในครั้งแรกยังอาจจะไม่มีข้อมูลเพียงพอที่จะกำหนด เนื่องจากการกำหนดช่วงความถี่และช่วงเวลาเท่ากันหมดก่อน เมื่อเขียนโปรแกรมไปช่วงหนึ่งแล้วสามารถทดสอบโปรแกรมได้ก็จะนิจารณาจากช่วงเวลาในการทำงานและการตอบสนองของระบบ แล้วจึงนำข้อมูลเหล่านี้มานิจารณาในการกำหนดความถี่และช่วงเวลาดังกล่าว ต่อจากนั้นก็จะเป็นส่วนของเงื่อนไขในการแสดงผลของกระบวนการซึ่งส่วนนี้จะไม่อธิบาย เนื่องจากจะแตกต่างกันตามกระบวนการและความต้องการของผู้ใช้.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเปลี่ยนไฟล์ภาพ AUTOCAD เป็นภาพของ TURBO PASCAL : จากที่กล่าวไว้ตอนต้นแล้วว่า ในการสร้างภาพกระบวนการ จะทำการสร้างภาพด้วยโปรแกรม AUTOCAD ก่อน เมื่อได้ภาพกระบวนการที่สมบูรณ์แล้ว จึงทำการเก็บภาพกระบวนการนั้น ลงในแผ่นจานแม่เหล็ก ซึ่งจะเก็บภาพเป็นไฟล์ โดยสร้างเป็น DRAWING INTERCHANGE FILE ดังที่กล่าวถึงรูปแบบของDARWING INTERCHANGEFILE ในบทที่ 3 เมื่อได้เก็บภาพเป็น Drawing Interchange File แล้ว เราจึงจะทำการอ่านรายละเอียดของการสร้างภาพจากไฟล์นี้ ออกมาโดยใช้เทอร์โบ ปาสคาล เมื่อได้ค่าต่างๆออกมา จึงทำการสร้างภาพกระบวนการอีกครั้ง โดยใช้รายละเอียด หรือค่าต่างๆที่อ่านได้จาก Drawing Interchange File แต่คำสั่งในการวาดครั้งใหม่นี้จะเป็นชุดคำสั่งของเทอร์โบ ปาสคาล

ดังนั้นเมื่อเราอ่าน Drawing Interchange File แล้วนำค่าที่ได้มาสร้างภาพอีกครั้ง ด้วยชุดคำสั่งของเทอร์โบ ปาสคาล จึงเสมือนกับว่า เราได้สร้างภาพกระบวนการด้วยเทอร์โบ ปาสคาล ซึ่งเราใช้การพิจารณาจากข้อมูลภายใน Drawing Interchange File มีขั้นตอนการพิจารณา ดังนี้

1 เปิดไฟล์ Drawing Interchang File

2 ทำการอ่านข้อมูลจนจบไฟล์ โดยพิจารณา

- **เมื่ออ่านไฟล์พบกับ String"LINE":** เมื่อพบ String "LINE" หมายความว่า ภาพที่สร้างโดยโปรแกรม AutoCAD ได้มีการใช้คำสั่ง LINE ให้อ่านค่า coordinate ซึ่งจะอยู่ในบรรทัดถัดมาจาก string LINE โดยพิจารณา code number ต่อไปนี้

10 - Integer ตัวถัดมาจะเป็น coordinate X1

11 - Integer ตัวถัดมาจะเป็น coordinate Y1

20 - Integer ตัวถัดมาจะเป็น coordinate X2

21 - Integer ตัวถัดมาจะเป็น coordinate Y2

เมื่อพิจารณาดังนี้แล้ว เราจะทำการตั้งตัวแปรเพื่อใช้ในการเก็บค่า coordinatนี้ไว้เมื่อได้ครบ ก็จะทำการสร้างภาพโดยใช้คำสั่งของเทอร์โบปาสคาล คือคำสั่ง"LINE (X1, Y1,X2,Y2)" ก็จะได้ภาพการเขียนเส้นตรงที่มี coordinate อยู่ที่จุด (X1,Y2)และ(X2,Y2) เป็นต้น

-เมื่ออ่านไฟล์พบกับ String"CIRCLE" : เมื่อพบกับ String "CIRCLE" หมายความว่าภาพที่สร้างโดยโปรแกรม AUTOCAD ได้ใช้คำสั่งในการสร้างภาพวงกลม ด้วยคำสั่ง"CIRCLE" ให้ทำการอ่านค่า coordinate คือจุดศูนย์กลาง และรัศมีของวงกลมนั้น โดยพิจารณาจาก Code Number ต่อไปนี้

10 - Integer ตัวถัดมาจะเป็น coordinate X

20 - Integer ตัวถัดมาจะเป็น coordinate Y

40 - Integer ตัวถัดมาจะเป็น รัศมี(Radius)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาดังนี้แล้ว เราจะทำการตั้งตัวแปรเพื่อใช้ในการเก็บค่า coordinateนี้ไว้เมื่ออ่านจนครบ ก็จะทำการสร้างภาพโดยใช้คำสั่งของเทอร์โบปาสคาล คือคำสั่ง"CIRCLE (X,Y,R))" ก็จะได้ภาพการเขียนวงกลมที่มีจุดศูนย์กลาง coordinate อยู่ที่จุด (X,Y)และรัศมีเท่ากับ R

-เมื่ออ่านไฟล์พบกับ String"TRACE" :เมื่อพบกับstring "trace"หมายความว่าภาพที่สร้างโดยโปรแกรม AUTOCAD ได้ใช้คำสั่งในการสร้างภาพเป็นการระบายทึบโดยใช้คำสั่ง"TRACE" ให้ทำการอ่านค่า coordinate โดยพิจารณาจาก Code Number ต่อไปนี้

10-Integer ตัวถัดไปจะเป็น coordinate X1

11-Integer ตัวถัดไปจะเป็น coordinate Y1

20-Integer ตัวถัดไปจะเป็น coordinate X2

21-Integer ตัวถัดไปจะเป็น coordinate Y2

เมื่อพิจารณาดังนี้แล้วเราจะทำการตั้งตัวแปรเพื่อเก็บค่า coordinate นี้ไว้เมื่ออ่านจนครบ ก็จะทำการสร้างภาพ โดยใช้คำสั่งของเทอร์โบปาสคาลด้วยคำสั่ง"BAR(X1,Y2, X2,Y2)" ก็จะเป็นการระบายทึบที่มีขนาดการระบายอยู่ที่มุมบล้างซ้ายคือ coordinate (X1, Y1)และมุมบนขวา คือ coordinate (X2,Y2)

- เมื่ออ่านไฟล์พบกับ String"ARC" : เมื่อพบกับString"ARC"หมายความว่าภาพที่สร้างโดยโปรแกรม Autocadได้ใช้คำสั่งในการสร้างภาพส่วนโค้งวงกลม ให้ทำการอ่านค่า coordinate เหมือนกับคำสั่ง "circle"แต่จะเป็นการอ่านค่า coordinate จุดศูนย์กลาง มุมเริ่มต้น มุมสิ้นสุด และรัศมี เมื่อได้ค่าครบเราก็ใช้คำสั่ง "ARC"ของเทอร์โบ ปาสคาลทำการสร้างภาพส่วนโค้งวงกลมอีกครั้ง โดยมีรูปแบบของคำสั่งคือ

ARC(X,Y,SA,ER,R); เมื่อ

coordinate(X,Y) เป็นจุดศูนย์กลาง

SA มุมเริ่มต้นของส่วนโค้ง

EA มุมสิ้นสุดของส่วนโค้ง

R เป็นรัศมีของส่วนโค้ง

- เมื่ออ่านไฟล์พบกับ String"TEXT" : เมื่อพบกับ String "TEXT" หมายความว่าได้มีการเขียนตัวอักษรลงในแบบแปลนหรือภาพที่สร้างโดย Autocad ด้วยคำสั่ง "TEXT"เมื่อเราทำการเก็บค่าต่างๆ จนครบแล้วเราจะใช้เทอร์โบปาสคาลเขียนตัวอักษรนั้น ด้วยคำสั่ง

outtextXY (x,y'TEXT'); เมื่อ

coordinate(x,y)เป็นตำแหน่งเริ่มต้นแสดงอักษรตัวแรก

'TEXT' คือ อักษรที่ต้องการทำออกแสดง

- เมื่ออ่านไฟล์พบกับ String"BLOCK" : เมื่อพบกับ String "Block" หมายความว่าได้มีการสร้างภาพโดยโปรแกรม Autocad แล้วทำการเก็บภาพนั้นไว้เป็น "Block"ซึ่ง Blockนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีชื่อเรียกในที่นี้เราได้สร้าง Block ไว้ 2 Block คือ Block ชื่อว่า "VALVE" และ Block ที่ชื่อว่า "PUMP" ดังนั้นเมื่อพบกับ String "Block" ถ้าเราอ่านไฟล์ต่อไปจะพบกับ

string "VALVE" : เมื่อพบ string "VALVE" ให้ทำการอ่านค่าที่อยู่ใน Blockvalve โดยเมื่ออ่านจนจบ Block จะพบกับ string"ENDBLK" นั่นหมายความว่าเราได้เก็บค่าต่างๆ ของ Blockvalve จนหมดแล้ว

string "PUMP" : เมื่อพบ string "PUMP" ให้ทำการอ่านค่าที่อยู่ใน blockpump โดยเมื่ออ่านค่าจนจบ block จะพบกับ string "ENDBLK" นั่นหมายความว่าเราได้เก็บค่าต่าง ๆ ของ blockpump จนหมดแล้ว

จากข้างบนเมื่อเราอ่านค่า block pump และ block valve เราจะต้องทำการเก็บค่านี้ ไว้ในหน่วยความจำโดยตั้งตัวแปรขึ้นเพื่อเก็บค่านี้ไว้ เมื่อมีการอ้างอิงถึงการสร้างภาพ pump หรือ valve เราจะใช้ค่าที่เก็บไว้นี้ออกมาทำการสร้างภาพ pump หรือ valve โดยใช้ชุดคำสั่ง turbo pascal

-เมื่ออ่านไฟล์พบกับ String"INSERT" : เมื่ออ่านไฟล์ พบกับ string "INSERT" นั่นคือได้มีการเรียกใช้ blockpump หรือ blockvalve ไปใช้ในการสร้างภาพ เช่น ในแบบแปลนของเราต้องการสร้างภาพ valve ขึ้นมา ก็ใช้คำสั่ง "INSERT" แล้วเรียกชื่อ blockvalve นำภาพ valve นั้นไปแทรกจุดที่ต้องการได้เลย เมื่อเราสร้าง drawing interchange file ภายในไฟล์จะมี string "INSERT" บรรจุอยู่ แล้วบรรทัดถัดมาจะบอกว่าได้เรียกใช้ block valve หรือ pump เมื่ออ่านพบกับ string "INSERT" ให้อ่านค่าของ "INSERT POINT" เมื่อได้ค่าจุด insert point เราก็จะนำค่าที่เก็บไว้จากคำสั่ง block มาสร้างภาพ valve หรือ pump ตามจุด insert point อีกครั้งจะได้ภาพ valve pump ตามจุดต่างๆ ในแบบแปลน

จากที่กล่าวมายังอาจจะสับสนอยู่ เพราะเป็นการอธิบายค่อนข้างจะสั้น และรวบรัด ดังนั้น จะขอแสดงตัวอย่างชุดคำสั่ง ที่ได้สร้างขึ้น เพื่อใช้ในการเปลี่ยนภาพกระบวนการที่สร้างโดย AutoCad เป็นภาพกระบวนการที่สร้างด้วยชุดคำสั่งของเทอร์โบ ปาสคาล ดังมี อัลกอริธึม ดังต่อไปนี้

```
procedure DrawDXFFile (DXFFileName : string; Scale,OfsX,OfsY : real);
var
  i,j,k,X,Y,R,X1,Y1,X2,Y2,M : LongInt;
  EAngle,SAngle : byte;
  F : real;
  CurColor : byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  CurColor := GrphColor;
  GrPointX := GrPointX + OfsX;
  GrPointY := GrPointY + OfsY;
  Assign(DXFFile,DXFFileName);      Assign(DXFFile,DXFFileName);
  {$I-} Reset(DXFFile); {$I+}
  if IOResult = 0 then
  begin
    while not EOF(DXFFile) do
    begin
      Readln(DXFFile,Code);
      if Code = 'CIRCLE' then
      begin { Circle }
        DXFCircle.Color := DefGrphColor;
        repeat
          Readln(DXFFile,Code);
          Val(Code,Data,Result);
          if Data <> 0 then
            CircleCommand;
        until Data = 0;
        X := Round((DXFCircle.X + GrPointX) * Scale);
        Y := Round((DXFCircle.Y + GrPointY) * Scale);
        R := Round(DXFCircle.R * Scale);
        GrSetColor(DXFCircle.Color);
        GrCircle(X,Y,R);
      end { Circle }
      else
      begin
        if Code = 'LINE' then
        begin { Line }
          DXFLine.Color := DefGrphColor;
          repeat
            Readln(DXFFile,Code);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Val(Code,Data,Result);
    if Data <> 0 then
      LineCommand;
  until Data = 0;
  X1 := Round((DXFLine.X1 + GrPointX) * Scale);
  Y1 := Round((DXFLine.Y1 + GrPointY) * Scale);
  X2 := Round((DXFLine.X2 + GrPointX) * Scale);
  Y2 := Round((DXFLine.Y2 + GrPointY) * Scale);
  GrSetColor(DXFLine.Color);
  GrMoveTo(X1,Y1);
  GrLineTo(X2,Y2);
end { Line }
else
begin
  if Code = 'TRACE' then
  begin { Trace }
    DXFTrace.Color := DefGrphColor;
    repeat
      Readln(DXFFile,Code);
      Val(Code,Data,Result);
      if Data <> 0 then
        TraceCommand;
    until Data = 0;
    X1 := Round((DXFTrace.X1 + GrPointX) * Scale);
    Y1 := Round((DXFTrace.Y1 + GrPointY) * Scale);
    X2 := Round((DXFTrace.X2 + GrPointX) * Scale);
    Y2 := Round((DXFTrace.Y2 + GrPointY) * Scale);
    GrSetColor(DXFTrace.Color);
    GrBar(X1,Y1,X2,Y2);
  end { Trace }
  else
  begin
    if Code = 'TEXT' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin { Text }
  DXFText.Color := DefTxtColor;
  repeat
    Readln(DXFFile,Code);
    Val(Code,Data,Result);
    if Data <> 0 then
      TextCommand;
  until Data = 0;
  X := Round((DXFText.X + GrPointX) * Scale);
  Y := Round((DXFText.Y + DXFText.Size + GrPointY) * Scale);
  M := Round(DXFText.Size * KxFactor[CurView] * Scale / 20 * 100)
  SetUserCharSize(M,100,M,100);
  SetTextStyle(TriplexFont,HorizDir,UserCharSize);
  GrSetColor(DXFText.Color);
  GrMoveTo(X,Y);
  OutText(DXFText.Message);
end { Text }
else
begin
if Code = 'ARC' then
begin
  DXFArc.Color:= DefGrphColor;
  repeat
    Readln(DXFFile,Code);
    Val(Code,Data,Result);
    if Data <> 0 then
      ARCCommand;
  until Data = 0 ;
  X := Round((DXFArc.X + GrPointX) * Scale);
  Y := Round((DXFArc.Y + GrPointY) * Scale);
  R := Round(DXFArc.R * Scale);
  SAngle := Round(DXFArc.SA * Scale);
  EAngle := Round(DXFArc.EA * Scale);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            GrSetColor(DXFArc.Color);
            GrARC(X,Y,SAngle,EAngle,R);
          end { ARC }
          else
          begin
          if Code = 'BLOCK' then
          begin
            ReadBlock;
          end { BLOCK }
          else
          begin
          if Code = 'INSERT' then
          begin
            InsertBlock;
          end { INSERT }
          else
          begin { anything else }
          end; { anything else }
          end;
        end;
      end;
    end;
  end;
  Close(DXFFile);
  end;
  SetTextStyle(DefaultFont,HorizDir,1);
  GrphColor := CurColor;
  GrSetColor(GrphColor);
end { DrawDXFFile };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure InsertBlock;
begin
 Repeat
  ReadIn(DXFFile,Code);
  Val(Code,Data,Result);
  if Data <> 2 then
     LayerColorBlock;
 Until Data = 2;
 ReadIn(DXFFile,Code);
 if Code = 'PUMP'then
 begin
  BLKNameInsert.Name:='PUMP';
  Repeat
   ReadIn(DXFFile,Code);
   Val(Code,Data,Result);
   if Data <> 0 then
      ReadPumpInsertPoint;
   Until Data = 0;
   DrawInsertPump;
 end {Pump}
else
 begin
  if Code = 'VALVE' then
  begin
   BLKNameInsert.Name:='VALVE';
   Repeat
    ReadIn(DXFFile,Code);
    Val(Code,Data,Result);
    if Data <> 0 then
       ReadValveInsertPoint;
   Until Data = 0;
   if CheckRotate = 1 then
   DrawInsertValveRotate
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      else DrawInsertValve;
    end {Valve};
  end {else};
end {InsertBlock};

procedure ReadBlock;
begin
i := 0;
j := 0;
k := 0;
Repeat
ReadIn(DXFFile,Code);
Val(Code,Data,Result);
if Data <> 2 then
   LayerColorBlock;
Until Data = 2;
ReadIn(DXFFile,Code);
if Code = 'VALVE' then
begin
Repeat
 ReadIn(DXFFile,Code);
 Val(Code,Data,Result);
 if Data <> 0 then
   ReadValveBasePoint;
Until Data = 0;
ReadIn(DXFFile,Code);
While Code <> 'ENDBLK' do
begin
 if Code = 'LINE' then
 begin
  i := i+1;
  BlockW.Color:=DefGraphColor;
  Repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ReadLn(DXFFile,Code);
    Val(Code,Data,Result);
    if Data <> 0 then
      ReadValveLINE;
  until Data = 0;
  end {Line}
  else
  begin
   if Code = 'ARC' then
   begin
    BlockV.Color:=DefGrphColor;
    Repeat
     ReadLn(DXFFile,Code);
     Val(Code,Data,Result);
     if Data <> 0 then
       ReadValveARC;
    Until Data = 0;
   end {ARC};
  end {else};
  ReadLn(DXFFile,Code);
 end {ENDBLK};
end {Valve}
else
begin
 if Code = 'PUMP' then
 begin
  Repeat
   ReadLn(DXFFile,Code);
   Val(Code,Data,Result);
   if Data <> 0 then
     ReadPumpBasePoint;
  Until Data = 0;
  ReadLn(DXFFile,Code);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
While Code <> 'ENDBLK' do
begin
 if Code = 'CIRCLE' then
 begin
  k := k+1;
  BlockP.Color:=DefGrphColor;
  Repeat
   ReadIn(DXFFile,Code);
   Val(Code,Data,Result);
   if Data <> 0 then
     ReadPumpCIRCLE;
  Until Data = 0;
 end {Circle}
 else
  begin
  if Code = 'LINE' then
  begin
   j := j+1;
   BlockP.Color:=DefGrphColor;
   Repeat
    ReadIn(DXFFile,Code);
    Val(Code,Data,Result);
    if Data <> 0 then
      ReadPumpLINE;
   Until Data = 0;
  end {Line};
 end {else};
 ReadIn(DXFFile,Code);
end {ENDBLK};
end {Pump};
end {else};
end {ReadBlock};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

การแสดงผลสภาวะของกระบวนการทางอุตสาหกรรมนี้ ได้ออกแบบมาเพื่อให้สามารถนำไปประยุกต์ใช้ในงานอุตสาหกรรม โดยการรับค่าสภาวะต่างๆทั้งชนิดอนาลอก และดิจิตอล ซึ่งในงานอุตสาหกรรมส่วนใหญ่จะมีรูปแบบของสัญญาณหรือข้อมูลในการวัดและควบคุมกระบวนการในลักษณะนี้ทั้งสิ้น ทางด้านการใช้งานอาจจะยากในส่วนของการสร้างภาพ ซึ่งโปรแกรมลักษณะนี้ในท้องตลาดมีอยู่ด้วยกันหลายโปรแกรม ซึ่งการใช้งานสามารถใช้งานได้ง่าย แต่มีความยืดหยุ่นของระบบต่ำ เช่นภาพที่แสดงบนจอแสดงผลเป็นภาพกราฟฟิกความละเอียดต่ำ ภาพที่แสดงบนจอแสดงผลจะเป็นแบบหยาบๆ เนื่องจากความละเอียดของจอแสดงผลและส่วนควบคุมต่ำ

ปัญหาที่พบในการทำปริญญานิพนธ์ครั้งนี้ เมื่อเริ่มต้นจะเป็นปัญหาในการสร้างภาพกระบวนการ เนื่องจากต้องสร้างภาพกระบวนการในส่วนแรกโดยโปรแกรม Autocad และเปลี่ยนกลับมาเป็นภาพกระบวนการที่สร้างโดยโปรแกรมภาษาเทอร์โบ ซ่าสคาล ซึ่งในการเปลี่ยนภาพนี้ จอภาพที่ใช้สร้างภาพกระบวนการจะมี coordinate ไม่สัมพันธ์กัน จึงทำให้ส่วนของการสร้างภาพเป็นลักษณะการพัฒนาโปรแกรมที่ยุ่งยาก และเสียเวลานาน เป็นต้น.

หมายเหตุ

จากคำสั่ง(Procedure) DrawLevel เราใช้เป็นคำสั่งในการวาดการเปลี่ยนแปลงระดับในถังทุกถัง ซึ่งจะต้องส่งค่าความสูงของถัง คือค่า coordinate ของมุมล่างซ้าย และมุมบนขวาของถังให้กับคำสั่งนี้ โดยจะต้องกำหนดในภาพกระบวนการที่จะสร้างโดยโปรแกรม AutoCAD เมื่อทำเป็นไฟล์ DXF แล้วภายในไฟล์จะมี coordinate ดังกล่าวนี้บรรจุอยู่ เมื่อทำการอ่านไฟล์ DXF ออกมาจะได้ coordinate จุดล่างซ้าย และจุดขวาบนของถังซึ่งการเปลี่ยนแปลงระดับจะอยู่ในช่วงความสูงของถังที่กำหนด

เนื่องจากในส่วนของการกำหนดระดับสูงสุด และต่ำสุดของการเปลี่ยนแปลงระดับในแต่ละถัง ในที่นี้ ได้กำหนดจากแบบแปลนของภาพกระบวนการ คือ กำหนดจุดล่างซ้าย และจุดบนขวาของถังจากภาพกระบวนการ ไม่ได้กำหนดโดยการลากเส้นตรงจากมุมล่างซ้าย ถึงมุมบนขวาของถัง เหตุผล เนื่องจากการที่จะกำหนดเส้นตรงจากจุดล่างซ้าย ถึงมุมบนขวาของถังในแต่ละถังจะต้องแยกชั้นในการวาดภาพอีก ตามจำนวนถังในภาพกระบวนการ ซึ่งจะทำให้จำนวนชั้นของภาพกระบวนการเพิ่มขึ้นอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ในปริญญานิพนธ์ฉบับนี้จึงไม่ได้มีการกำหนดการเปลี่ยนแปลงระดับของถังโดยการสร้างเส้นตรงลงไปในภาพกระบวนการ แต่การที่จะกำหนดโดยการลากเส้นตรงจากมุมล่างซ้าย ถึงมุมขวาบนของถังลงไปในภาพกระบวนการก็สามารถทำได้ โดยการแยกชั้นออกไปอีก เช่น ระดับการเปลี่ยนแปลงของถัง A2 เราอาจจะตั้งชื่อชั้นว่า "Level A2" แล้วทำการลากเส้นตรงจากก้นถัง ถึงปากถัง A2 เมื่อสร้าง Drawing Interchange File (DXF) จุด coordinate ล่างซ้าย และขนขวาของถังA2 จะถูกบรรจุอยู่ในชั้น"Level A2"เมื่อทราบดังนี้ จึงง่ายในการอ่านค่า coordinate ของจุด 2 จุดนี้ เมื่ออ่านออกมาได้ก็ส่งค่า 2 ค่านี้เป็นการกำหนดระดับต่ำสุด และสูงสุดของถัง A2 ซึ่งถ้ามีการเปลี่ยนภาพกระบวนการ เช่น ย่อ หรือ ขยายขนาดของถังก็ไม่ต้องกำหนดจุดเหล่านี้ใหม่ แต่ในการกำหนดระดับโดยวิธีนี้ จะต้องสร้างชั้นของภาพเพิ่มขึ้นอีก ดังกล่าว.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

(ACKNOWLEDGMENT)

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ก็ด้วยความร่วมมือ และความช่วยเหลือจากหลายฝ่ายด้วยกัน โดยเฉพาะอย่างยิ่งคุณพ่อ คุณแม่ รวมถึง อาจารย์ ภากร หุตะสังกาศ ผู้ซึ่งเป็นอาจารย์ที่ปรึกษา และ ผู้คอยช่วยเหลือ ให้คำแนะนำ

คุณ ชัยรัตน์ ขาวเอี่ยม ลงมือช่วยบางครั้งคราว
คุณ นิพนธ์ ศิริรัตน์ ให้คำแนะนำ
คุณ เรืองไกร รังสิพล ให้คำแนะนำ
คุณ สุรพงษ์ ชุมหะโสภาค เอื้อเฟื้อที่พัก เสื้อผ้า
คุณ อวัชชัย สุมล,ตุ่ ส่วนหน้าของปริญญานิพนธ์
คุณ ชินกร ชลธีชาชลาลักษ์,เก๋ ส่วนหลังปริญญานิพนธ์
คุณ นิวัต เจริญชล,เก๊ก ตำราประกอบ
คุณ สาท ค้ามูล บริการห้องคอมพิวเตอร์

จึงขอขอบพระคุณไว้ ณ ที่นี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

(REFERENCES)

1. ภากร หุตะสังกาศ; ระบบแสดงผลความละเอียดสูงสำหรับกระบวนการทางอุตสาหกรรม
2. ชัชวาล ยนต์หงส์; แนะนำภาษาปาสคาล โดย เทอร์โบปาสคาล, 2532
3. บุญเลิศ เอื้อมทัศนา; คู่มือ เทอร์โบปาสคาล รุ่น 4.0 - 5.0, 2532
4. ประพัฒน์ อุทโยภาศ; คู่มือ การใช้โปรแกรม AUTOCAD, 2532
5. Tom Swan ; MASTERING TURBO PASCAL 4.0,
   Howard W. Sams & Co., Inc., 1983

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้