สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

DATA ACQUISITION AND ANALYSIS SYSTEM FOR HPLC

หนังสืออ้างอิง
ห้ามนำออกนอกห้องสมุด

นายจิรวัฒน์ ตันตราจิณ
MR. JIRAWAT TANTRAJIN

T021395

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาฟิสิกส์ประยุกต์
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2536

ISBN 974-621-130-7

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 21395
วัน, เดือน, ปี 19 ก.ย. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA ACQUISITION AND ANALYSIS SYSTEM FOR HPLC

MR. JIRAWAT TANTRAJIN

A THESIS SUBMITTED IN PATIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE IN APPLIED PHYSICS

GRADUATE SCHOOL

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADGRABANG

1993

ISBN 974-621-130-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC
นักศึกษา	นายจิรวัฒน์ ตันตราจิณ
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.อารีย์ วิเชียรฉาย
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์
ภาควิชา	ฟิสิกส์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2536

บทคัดย่อ

ชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับเครื่อง High Performance Liquid Chromatography(HPLC) ได้พัฒนาขึ้นมาแทนเครื่อง Computing Integrator ซึ่งเป็นส่วนประกอบหลักอันหนึ่งของเครื่อง HPLC ที่มีราคา 1/3 ของราคาเครื่องวิเคราะห์ HPLC ชุดเก็บข้อมูลที่พัฒนาขึ้นดังกล่าวมีความสามารถทัดเทียมกับเครื่อง Computing Integrator ที่ผลิตขึ้นจากต่างประเทศ แต่ยังมีส่วนที่มีคุณสมบัติเหนือไปจากเครื่อง Computing Integrator ที่ใช้กับเครื่อง HPLC คือ ชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับเครื่อง HPLC นี้ สามารถบันทึกข้อมูลลงบนแผ่นจานแม่เหล็กได้ สำหรับซอฟต์แวร์วิเคราะห์ผลที่ได้พัฒนาขึ้นมาโดยใช้ภาษาปาสคาล สามารถแสดงรูปสัญญาณและวิเคราะห์ผลในขณะที่เก็บข้อมูล และได้ปรับเทียบกับเครื่อง Computing Integrator ของ Spectra Physics รุ่น SP4200 พบว่ามีความแตกต่างกันน้อยกว่า 2 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	DATA ACQUISITION AND ANALYSIS SYSTEM FOR HPLC
Student	Mr.Jirawat Tantrajin
Thesis Advisor	Asst.Prof.Dr.Aree Wichainchai
Level of Study	Master of Science in Applied Physics
Department	Applied Physics,King Mongkut's Institute of Technology Ladgrabang
Year	1993

ABSTRACT

Data Acquisition and Analysis System(DAS) for High Performance Liquid Chromatography(HPLC) is developed to be replaced of Computing Integrator which is about 1/3 higher price than DAS. This system has equal ability to the import one,in advantage,the DAS has extra ability that it can records data into the disks.

The analysis software is special developed by using PASCAL, the system can display the signal and analyse data in the same time of collecting data. The reliability of DAS is high in comparison with the computing integrator model SP4200 (from SPECTRA PHYSICS) because the difference was less than 2 percents.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับความช่วยเหลือจากท่านผู้มีพระคุณหลายท่าน โดยเฉพาะอย่างยิ่ง คุณพ่อและคุณแม่ของข้าพเจ้าที่ให้การสนับสนุนและช่วยเหลือโดยซื้อเครื่องคอมพิวเตอร์สำหรับใช้ทำวิทยานิพนธ์โดยเฉพาะ อีกทั้งยังให้กำลังใจเป็นอย่างดีตลอดมา

ขอขอบพระคุณ ผศ.ดร.อารีย์ วิเชียรฉาย เป็นอย่างสูง ที่ได้ให้ความกรุณาแนะนำและให้ความช่วยเหลือในการทำวิทยานิพนธ์นี้

ขอขอบพระคุณ ผศ.ดร.บุญส่ง ศิวโมกษธรรม ผศ.สุวรรณ คูสำราญ ผศ.ศิริศักดิ์ เตชะทวีกุล และคณาจารย์ภาควิชาฟิสิกส์ประยุกต์ทุกท่าน ที่ได้กรุณาให้คำแนะนำอันเป็นประโยชน์ในการทำวิทยานิพนธ์แก่ข้าพเจ้า

ขอขอบพระคุณ ดร.วิโรจน์ ตันตราภรณ์ เป็นอย่างสูง ที่กรุณาให้ความสนับสนุนในการทำวิทยานิพนธ์แก่ข้าพเจ้า

ขอขอบพระคุณ คุณมานพ แม้นโกศล เจ้าหน้าที่กองเกษตรเคมี มหาวิทยาลัยเกษตรศาสตร์ ที่ได้ให้ความอนุเคราะห์ และให้ความร่วมมือในการทดสอบชุดเก็บข้อมูลและวิเคราะห์ผลกับเครื่อง HPLC อีกทั้งได้ช่วยให้ข้อมูลประกอบการวิจัยแก่ข้าพเจ้าเป็นอย่างดี

ขอขอบคุณ พี่วิชิต พี่วิชาญ และพี่จิ๊บ ที่ได้ให้ความช่วยเหลือและเป็นกำลังใจแก่ข้าพเจ้า

ขอขอบคุณ เจ้าหน้าที่ธุรการภาควิชาฟิสิกส์ประยุกต์ เจ้าหน้าที่ธุรการคณะวิทยาศาสตร์ และเจ้าหน้าที่บัณฑิตวิทยาลัย สำหรับความช่วยเหลือ ให้ความสะดวกในการติดต่องานอย่างดียิ่ง

สุดท้ายนี้ ข้าพเจ้าขอขอบคุณ พี่นา หน่อง เอ๋ เติบ ไก่ และน้องๆ ภาควิชาฟิสิกส์ประยุกต์ทุกคนที่เป็นกำลังใจตลอดมา

นายจิรวัฒน์ ตันตราจิณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VI
สารบัญภาพ.....VII
บทที่
1.บทนำ.....1
1.1 ความเป็นมาและความสำคัญของปัญหา.....1
1.2 วัตถุประสงค์ของการทำวิทยานิพนธ์.....2
1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการทำวิทยานิพนธ์.....2
1.4 ขอบเขตของการทำวิทยานิพนธ์.....3
1.5 วิธีที่ใช้ในการดำเนินการทำวิทยานิพนธ์.....3
2.หลักการพื้นฐานของ HPLC.....5
2.1 ประวัติความเป็นมาของ HPLC.....5
2.2 ระบบการทำงานของ HPLC.....6
2.3 การวิเคราะห์ผลจากสัญญาณของ HPLC.....8
3.หลักการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....10
4.การออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....20
4.1 หลักการออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....20
4.2 การออกแบบชุดเก็บและรับส่งข้อมูล.....21
4.3 การออกแบบซอฟต์แวร์วิเคราะห์ผลข้อมูล.....30
4.4 การเชื่อมต่อระหว่างชุดเก็บและรับส่งข้อมูลกับซอฟต์แวร์วิเคราะห์ผลข้อมูล.....43
5.การใช้งานชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....45
5.1 การติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....45
5.2 การใช้โปรแกรมในการวิเคราะห์ผลข้อมูลของ HPLC.....47
6.การทดสอบการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....61
6.1 หาค่ารีเทนชันไทม์ของคาเฟอีนจากสารคาเฟอีนมาตรฐาน.....61
6.2 หาค่าเรสปอนด์แฟกเตอร์เฉลี่ยของคาเฟอีน.....63
6.3 หาค่าความเข้มข้นของคาเฟอีนในเครื่องดื่มโคลา 2 ชนิด.....67
6.4 สรุปผลการทดสอบ.....68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

7.สรุปผลการวิจัยและข้อเสนอแนะ.....69
บรรณานุกรม.....70

ภาคผนวก ก.
เครื่องเก็บและวิเคราะห์ผลข้อมูลที่สร้างขึ้นและเครื่องวิเคราะห์ผลข้อมูลจากต่างประเทศ..71

ภาคผนวก ข.
ซอฟต์แวร์วิเคราะห์ข้อมูลสำหรับเครื่อง HPLC (ใช้กับเครื่องไมโครคอมพิวเตอร์).....74

ภาคผนวก ค.
ซอฟต์แวร์ของเครื่องเก็บและรับส่งข้อมูล (ใช้กับไมโครคอนโทรลเลอร์ 8751).....164

ภาคผนวก ง.
วงจรเครื่องเก็บและรับส่งข้อมูล.....184

ภาคผนวก จ.
ตัวอย่างการรายงานผลการวิเคราะห์จากซอฟต์แวร์ที่พัฒนาขึ้น
และจากเครื่องวิเคราะห์ผลข้อมูลของต่างประเทศ.....192

ประวัติผู้เขียน.....195

สารบัญตาราง

ตารางที่ หน้า

6.1 แสดงค่ารีเทนชันไทม์ของคาเฟอีนที่ได้จาก DAS001 และ SP4200.....62

6.2 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 100 PPM ซึ่งได้จาก DAS001 และ SP4200.....63

6.3 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 150 PPM ซึ่งได้จาก DAS001 และ SP4200.....64

6.4 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 200 PPM ซึ่งได้จาก DAS001 และ SP4200.....65

6.5 แสดงค่าความเข้มข้นของคาเฟอีนในเครื่องดื่มโคลา 2 ชนิด โดยใช้ DAS001 และ SP4200.....67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2.1 แสดงระบบการทำงานของ HPLC.....7

2.2 แสดงลักษณะของพีคสัญญาณของสารตัวอย่างซึ่งประกอบไปด้วยสาร 4 ชนิด คือ A B C และ D.....8

3.1 แสดงการประมวลผลสัญญาณของ HPLC.....10

3.2 แสดงดาตาบันช์ที่เกิดจากดาตาแซมเปิล 5 ชุดมารวมกัน.....11

3.3 แสดงการคำนวณหาความชันของข้อมูล.....12

3.4 แสดงการตรวจจุดเริ่มต้นและจุดปลายของพีคจากความชันของสัญญาณ.....12

3.5 แสดงการหาจุดยอดของพีคจากความชัน.....13

3.6 แสดงการหา Rt_{s1} ของคาเฟอีนมาตรฐาน.....15

3.7 แสดงการตรวจหาพีคของคาเฟอีนในสารตัวอย่าง.....15

3.8 แสดงการหาค่า Rf_{s1} ของคาเฟอีนมาตรฐาน 200 PPM.....16

3.9 แสดงการหาค่าความเข้มข้นของคาเฟอีนในสารตัวอย่าง.....16

3.10 แสดงลักษณะการต่อเครื่องวิเคราะห์ใน Labnet Mode.....17

3.11 แสดงการประมวลผลข้อมูลใน Labnet Mode.....18

3.12 แสดงการประมวลผลข้อมูลใน File Mode.....19

4.1 การทำงานของชุดเก็บและรับส่งข้อมูล.....23

4.2 แสดงวงจรแปลงสัญญาณอนาลอกเป็นความถี่.....24

4.3 แสดงวงจรนับและคงสถานะข้อมูล.....25

4.4 แสดงภาคประมวลผลและรับส่งข้อมูล.....26

4.5 แสดงวงจรของภาคจ่ายไฟเลี้ยง.....27

4.6 แสดงวงจร DUAL TRACKING REGULATOR.....28

4.7 แสดงภาคสร้างสัญญาณนาฬิกาขนาด 2 MHz และ 50 Hz.....29

4.8 แสดงการแปลงข้อมูลจากรหัสแอสกีเป็นจำนวนเต็มชนิด Longint.....32

4.9 แสดงการคำนวณโดยใช้ Moving average แบบเฉลี่ย 5 จุด.....33

4.10 แสดงขั้นตอนการคำนวณโดยวิธี Moving average แบบเฉลี่ย 5 จุด.....34

4.11 แสดงหลักการตรวจหาพีคสัญญาณและวิเคราะห์ข้อมูล.....35

4.12 แสดงหลักการตรวจหาจุดเริ่มต้นพีค.....36

4.13 แสดงหลักการตรวจหาจุดยอดพีค(Rt).....37

4.14 แสดงหลักการตรวจหาจุดปลายพีค.....38

4.15 หลักการแสดงกราฟสัญญาณบนจอภาพ.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

4.16 แสดงลักษณะของกราฟสัญญาณที่พิมพ์ออกมาทางเครื่องพิมพ์.....42
5.1 แสดงการติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC.....46
5.2 แสดงลักษณะของโปรแกรมวิเคราะห์ข้อมูลสำหรับ HPLC.....51
5.3 แสดงการกำหนด Mode การทำงานของโปรแกรม.....52
5.4 แสดงการกำหนดพารามิเตอร์ในรายการ Option.....53
5.5 แสดงการเลือกวิธีวิเคราะห์จากรายการ Method.....54
5.6 แสดงการใส่ข้อมูลของสารมาตรฐานใน External Std. Method.....55
5.7 แสดงการกำหนดชื่อไฟล์ข้อมูลและไดเรกทอรีที่ต้องการใช้งาน.....56
5.8 แสดงกราฟสัญญาณและการรายงานผลทางจอภาพ(Normal Method).....57
5.9 แสดงกราฟสัญญาณและการรายงานผลทางจอภาพ(External Std. Method).....58
5.10 แสดงตัวอย่างการรายงานผลการวิเคราะห์ เมื่อพิมพ์ออกมาแล้ว.....59
5.11 แสดงการยกเลิกการทำงานของโปรแกรม.....60
6.1 แสดงกราฟความสัมพันธ์ระหว่างพื้นที่พีค กับความเข้มข้นของคาเฟอีน
ซึ่งได้จาก DAS001 และ SP4200.....66

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การวิเคราะห์สารของ HPLC จะอาศัยสัญญาณแรงดันไฟฟ้าที่ได้จากตัวตรวจวัด(Detector) มาใช้เป็นข้อมูล เพื่อตรวจสอบชนิดของสารที่สนใจว่ามีเจือปนอยู่บ้างหรือไม่และมีปริมาณเป็นเท่าใด โดยแต่เดิมนั้นจะใช้เครื่องบันทึกสัญญาณ(Y-t Recoder) เป็นตัวบันทึกลักษณะของสัญญาณ จากนั้นจึงนำกราฟสัญญาณที่ได้ไปวิเคราะห์หาชนิดหรือความเข้มข้นของสารที่เจือปนอยู่แต่ก็ยังมีข้อเสียในแง่ของความผิดพลาดในการวัด และการใช้คนเป็นผู้วัดวิเคราะห์โดยตรงจากกราฟทำให้ต้องเสียเวลาในการตรวจสอบและคำนวณ ไม่เหมาะกับงานที่ต้องการความรวดเร็วหรือมีการวิเคราะห์มาก ๆ

ปัจจุบันได้มีการพัฒนาเครื่องวิเคราะห์ผล(Computing Integrator) เพื่อนำมาใช้กับการวิเคราะห์ข้อมูลที่ได้จากสัญญาณของเครื่อง HPLC โดยตรง ทำให้สามารถวิเคราะห์ ชนิดและความเข้มข้นของสารที่พบได้ โดยมีความละเอียดและมีความถูกต้องสูง สามารถวิเคราะห์ได้รวดเร็ว อีกทั้งยังสามารถกำหนดรูปแบบของการวิเคราะห์ได้หลายวิธี แต่มีราคาค่อนข้างสูง

การสร้างชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC นี้ เป็นการนำเอาไมโครคอมพิวเตอร์มาใช้ประโยชน์ในการวิเคราะห์สัญญาณของ HPLC ซึ่งสามารถวิเคราะห์หาชนิดและความเข้มข้นของสารที่พบได้อย่างรวดเร็ว โดยมีความละเอียด และมีความถูกต้องสูงใกล้เคียงกับเครื่องวิเคราะห์ผลจากต่างประเทศ และยังสามารถกำหนดรูปแบบของการวิเคราะห์ได้ด้วย โดยเสียค่าใช้จ่ายน้อยกว่าเครื่องประมวลผลจากต่างประเทศทั้งในด้านราคา และวัสดุสิ้นเปลืองที่ต้องใช้ในการรายงานผลการวิเคราะห์ เช่น กระดาษต่อเนื่องซึ่งมักเป็นกระดาษที่ใช้งานโดยเฉพาะและมีราคาแพง ในขณะที่ซอฟต์แวร์ที่พัฒนาขึ้นมานี้สามารถแสดงผลการวิเคราะห์บนจอภาพ และสามารถเก็บข้อมูลลงบนแผ่นจานแม่เหล็ก(Diskette)ได้ด้วย นอกจากนี้ยังมีข้อดีในแง่ของการนำข้อมูลที่เก็บไว้มาวิเคราะห์ภายหลังได้อีก ทำให้สามารถวิเคราะห์ข้อมูลในรูปแบบที่แตกต่างกันได้ ตามพารามิเตอร์หรือวิธีการที่กำหนดขึ้นมาใช้ในการวิเคราะห์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของการทำวิทยานิพนธ์

การสร้างชุดเก็บและวิเคราะห์ผลสำหรับ HPLC นี้ มีวัตถุประสงค์อยู่ 4 ประการ คือ

1.2.1 เพื่อพัฒนาวิธีการเก็บและวิเคราะห์ข้อมูลสำหรับ HPLC บนไมโครคอมพิวเตอร์

1.2.2 เพื่อสร้างเครื่องมือที่ช่วยให้สามารถวิเคราะห์ผลข้อมูลจากสัญญาณของ HPLC ได้อย่างถูกต้องและรวดเร็วขึ้น

1.2.3 เพื่อให้ความรู้ด้านเทคนิคการเก็บและวิเคราะห์ข้อมูล โดยใช้ไมโครคอมพิวเตอร์เพื่อการนำไปประยุกต์ใช้กับเครื่องมือวิเคราะห์ชนิดอื่นๆ

1.2.4 เพื่อกระตุ้นและส่งเสริมให้เกิดความคิดริเริ่มที่จะสร้างและพัฒนาเทคโนโลยีขึ้นมาใช้เองภายในประเทศ

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการทำวิทยานิพนธ์

ในกระบวนการวิเคราะห์ เพื่อตรวจสอบชนิด และความเข้มข้นของสารต่าง ๆ ที่เจือปนอยู่ในสารละลายโดยวิธี HPLC นี้ จะใช้ลักษณะของสัญญาณที่ได้จากตัวตรวจวัดของ HPLC โดยอาศัยหลักการที่ว่า สารชนิดเดียวกันจะใช้เวลาในการเดินทางจากจุดเริ่มต้น ผ่านคอลัมน์จนถึงตัวตรวจวัดของเครื่อง HPLC เท่ากันเสมอ(ภายใต้สภาวะเดียวกัน) และเรียกช่วงเวลานี้ว่า รีเทนชันไทม์ (Retention time) ดังนั้นสารชนิดเดียวกันจะปรากฏเป็นพีคสัญญาณในตำแหน่งเวลาเดียวกันเสมอ จึงสามารถใช้วิธีการตรวจสอบชนิดจากการเปรียบเทียบกับสารมาตรฐานที่ทราบชนิดได้ และสามารถคำนวณหาค่าความเข้มข้นของสารที่ตรวจพบ จากการเปรียบเทียบขนาดพื้นที่ของพีคสัญญาณของสารตัวอย่างกับพื้นที่พีคสัญญาณสารมาตรฐานที่เป็นชนิดเดียวกันได้

การวิเคราะห์จากกราฟสัญญาณที่ได้จากเครื่อง Y-t Recoder นั้น สามารถที่จะหาได้ทั้งชนิดและความเข้มข้นของสาร แต่ก็ยังมีความผิดพลาดในการคำนวณหาความเข้มข้นของสารได้มาก ในปัจจุบันสามารถนำเอาสัญญาณไฟฟ้าที่ได้จากตัวตรวจวัดมาแปลงเป็นข้อมูลดิจิตอล และนำไปวิเคราะห์ข้อมูลด้วยคอมพิวเตอร์ ซึ่งช่วยให้การวิเคราะห์ข้อมูลมีความถูกต้องสูง รวดเร็ว และสามารถกำหนดรูปแบบการวิเคราะห์ได้หลายวิธี ซึ่งในการวิจัยนี้เป็นการนำเอาหลักการวิเคราะห์ของเครื่องวิเคราะห์ผลมาใช้ เพื่อพัฒนาชุดเก็บและวิเคราะห์ผลสำหรับ HPLC ขึ้นมาในราคาที่ต่ำกว่าเครื่องวิเคราะห์ผลจากต่างประเทศ อีกทั้งยังสามารถนำไปพัฒนาต่อเพื่อเพิ่มขีดความสามารถในการวิเคราะห์สารได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของการทำวิทยานิพนธ์

เนื่องจากการวิจัยนี้ ยังเป็นการวิจัยในระดับเบื้องต้น โดยการนำเอาหลักการพื้นฐานของการวิเคราะห์ข้อมูลมาใช้ในการพัฒนาซอฟต์แวร์ต้นแบบขึ้นมา ซึ่งมีวิธีการวิเคราะห์ข้อมูลจากสัญญาณของเครื่อง HPLC อย่างง่าย ๆ คือ การหาค่ารีเทนชันไทม์ของพีคสัญญาณต่างๆ ที่ตรวจพบ และการหาขนาดของพื้นที่ใต้พีคสัญญาณที่ตรวจพบ เพื่อใช้ตรวจสอบชนิด และความเข้มข้นของสารที่ตรวจพบโดยวิธีการคำนวณเปรียบเทียบกับสารมาตรฐาน สำหรับวิธีการวิเคราะห์ที่ซับซ้อนกว่านี้จำเป็นที่จะต้องนำไปพัฒนาเพิ่มเติมต่อไป

1.5 วิธีที่ใช้ในการดำเนินการทำวิทยานิพนธ์

แผนการดำเนินงาน สามารถแบ่งเป็นขั้นตอนต่างๆ ได้ดังต่อไปนี้

ขั้นที่ 1 การศึกษาข้อมูลซึ่งจะศึกษาข้อมูลต่างๆ เช่นการวิเคราะห์ข้อมูลของเครื่องประมวลผล ระบบการทำงานของเครื่อง HPLC ศึกษาข้อมูลทางเทคนิคของสัญญาณเอาท์พุทจากตัวตรวจวัด รูปแบบการรายงานผลการวิเคราะห์

ขั้นที่ 2 ออกแบบระบบประมวลผล โดยจะเริ่มพิจารณาตั้งแต่การนำเอาสัญญาณที่ได้จากตัวดีเทคเตอร์ไปทำการประมวลผลจนถึงการรายงานผลขั้นสุดท้าย โดยสามารถแบ่งเป็น 2 ส่วนใหญ่ๆ ด้วยกัน คือ ฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งในส่วนของฮาร์ดแวร์นั้นจะเป็นส่วนที่นำเอาสัญญาณมาเปลี่ยนเป็นข้อมูลดิจิตอลแล้วส่งข้อมูลเป็นรหัสแอสกีผ่านให้กับไมโครคอมพิวเตอร์ ส่วนของซอฟต์แวร์จะเป็นการนำเอาข้อมูลที่ได้รับไปวิเคราะห์และรายงานผล

ขั้นที่ 3 สร้างเครื่องรับส่งข้อมูลระหว่างตัวตรวจวัดกับคอมพิวเตอร์ สำหรับในขั้นนี้จะเป็นการสร้างเครื่องมือที่จะรับสัญญาณอนาลอกจากตัวตรวจวัดมาเปลี่ยนเป็นข้อมูลทางดิจิตอล โดยอาศัยวงจรเปลี่ยนแรงดันเป็นความถี่ ซึ่งข้อมูลดิจิตอลที่ได้นี้จะถูกเก็บและส่งผ่านให้กับไมโครคอมพิวเตอร์ต่อไป

ขั้นที่ 4 เขียนซอฟต์แวร์เพื่อใช้ในการวิเคราะห์ข้อมูล โดยเมื่อเลือกภาษาคอมพิวเตอร์ที่จะใช้ในการเขียนโปรแกรมแล้ว จะต้องเตรียมข้อมูลที่ต้องใช้ รูปแบบและวิธีการวิเคราะห์ และลักษณะการรายงานผลการวิเคราะห์ เพื่อที่จะใช้ในการกำหนดโครงสร้างและออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5 ติดตั้งระบบและทดสอบการใช้งาน เมื่อสร้างเครื่องรับส่งข้อมูลและมีซอฟต์แวร์เพื่อใช้ในการวิเคราะห์ข้อมูลเรียบร้อยแล้ว ในขั้นนี้จะเป็นการติดตั้งระบบเพื่อทดสอบการใช้งานว่ามีข้อบกพร่องที่จะต้องปรับปรุงแก้ไขหรือไม่ อย่างไร หรือควรปรับปรุงเพิ่มเติมในส่วนใดบ้าง

ขั้นที่ 6 ปรับปรุงและแก้ไขข้อบกพร่องต่างๆ ที่พบ

ขั้นที่ 7 สรุปผลการทำวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการพื้นฐานของ HPLC

HPLC(High-Performance Liquid Chromatography) เป็นเทคนิคที่ใช้ในการแยกแยะและวิเคราะห์เพื่อหาชนิดกับปริมาณของสารต่างๆ ที่ผสมอยู่ในสารสารละลายตัวอย่าง ซึ่งเป็นเทคนิคที่ได้รับการพัฒนามาจากคอลัมน์โครมาโตกราฟีที่ใช้ของเหลวเป็นตัวทำละลาย

โครมาโตกราฟี(Chromatography) คือวิธีการแยกและวิเคราะห์ของผสมของสารประกอบทางเคมีโดยใช้การดูดซับด้วยสารดูดซับ(Adsorbing Material) โดยกระบวนการนี้สารที่ถูกดูดซับได้น้อยจะออกมาได้เร็วกว่าสารที่ถูกดูดซับได้ดี ซึ่งในกรณีของ HPLC จะใช้ตัวทำละลายที่เป็นของเหลวเรียกว่าโมบายเฟส(Mobile Phase) สำหรับพาสารตัวอย่างที่ต้องการทดสอบเข้าไปในคอลัมน์ซึ่งบรรจุสารดูดซับเรียกว่าสเตชันนารีเฟส(Stationary Phase) สารที่มีคุณสมบัติการถูกดูดซับแตกต่างกันจะถูกแยกออกมาที่เวลาที่แตกต่างกันด้วย และเมื่อนำไปผ่านตัวตรวจวัดซึ่งทำหน้าที่ตรวจสอบคุณสมบัติของสารที่ไหลผ่านมาโดยให้ผลออกมาเป็นสัญญาณไฟฟ้า จึงสามารถนำเอาสัญญาณไฟฟ้าที่ได้ไปวิเคราะห์หาชนิดและปริมาณสารต่างๆ ต่อไป

2.1 ประวัติความเป็นมาของ HPLC

จากการค้นพบคอลัมน์โครมาโตกราฟี ประมาณในช่วงต้นๆ ของคริสต์ศตวรรษที่ 19 ซึ่งมีนักพฤกษศาสตร์ชาวรัสเซีย ชื่อ มิคาอิล สเว็ตต์ (Mikhail Tswett) ได้ประดิษฐ์เครื่องมือชนิดหนึ่งซึ่งใช้ในการแยกรงควัตถุของพืช เช่นคลอโรฟิลล์(Chlorophylls) แซนโทฟิลล์(Xanthophylls) โดยการนำสารละลายของรงควัตถุที่ผสมกันอยู่ผ่านลงไปในคอลัมน์ ซึ่งเป็นหลอดแก้วบรรจุผงหินปูนละเอียดเอาไว้ ในการแยกรงควัตถุออกจากกันนั้น จะปรากฏเป็นแถบสีแยกออกมาบนคอลัมน์ ด้วยเหตุนี้จึงเกิดเป็นชื่อเทคนิคการแยกสารที่เรียกว่าโครมาโตกราฟี อันมาจากรากศัพท์ภาษากรีก 2 คำ คือ โครมา(Chroma) ซึ่งมีความหมายว่า สี และกราฟีอิน(Graphein) ซึ่งมีความหมายว่า เขียน

คอลัมน์โครมาโตกราฟีในยุคแรกนั้นยังอาศัยความโน้มถ่วงของโลกเป็นตัวช่วยให้ของเหลวไหลผ่านคอลัมน์ ซึ่งคอลัมน์ที่ใช้กันก็ยังมีขนาดใหญ่ โดยมีขนาดเส้นผ่าศูนย์กลางประมาณ 1-5 เซนติเมตร และมีความยาวตั้งแต่ 50-500 เซนติเมตร และขนาดของอนุภาคที่บรรจุในหลอดแก้วโดยทั่วไปจะมีขนาดอยู่ในช่วง 150-200 ไมโครเมตร ในการแยกสารแต่ละครั้งอาจใช้เวลานานหลายชั่วโมง และยังมีประสิทธิภาพในการแยกไม่ดีนัก

ต่อมาได้มีการปรับปรุงพัฒนาเทคนิคให้มีประสิทธิภาพสูงขึ้น และใช้เวลาในการแยกสารน้อยลง โดยการลดขนาดของอนุภาคที่ใช้ในคอลัมน์ ซึ่งมีเส้นผ่าศูนย์กลางของอนุภาคอยู่ในช่วงประมาณ 3-20 ไมโครเมตรแล้วเพิ่มแรงดันให้กับของเหลวมากขึ้นประมาณ 300-400 บรรยากาศ เราเรียกเทคนิคการแยกสารที่พัฒนาขึ้นมาใหม่นี้ว่า High Performance Liquid Chromatography หรือ HPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ระบบการทำงานของ HPLC

ในระบบการทำงานของ HPLC จะประกอบไปด้วยส่วนสำคัญต่างๆดังนี้

2.2.1 ปั๊มท์(Pump) เป็นอุปกรณ์สำคัญตัวหนึ่งในระบบ HPLC โดยจะทำหน้าที่ขับเคลื่อนตัวทำละลาย(Solvent) ให้ไหลผ่านคอลัมน์ไปยังตัวตรวจวัด(Detector)

2.2.2 วาล์วฉีดสาร(Injection valve) เป็นอุปกรณ์ที่ใช้เป็นประตูเปิดปิดในการฉีดสารละลายที่ต้องการวิเคราะห์ โดยสามารถกำหนดปริมาณของสารละลายที่จะฉีดได้ตั้งแต่ 5-500 ไมโครลิตร

2.2.3 คอลัมน์(Column) เป็นอุปกรณ์ที่ใช้ในการแยกสารต่าง ๆ ออกจากกัน โดยทั่วไปตัวคอลัมน์จะเป็นท่อสแตนเลสขนาดเส้นผ่าศูนย์กลางประมาณ 4-10 มิลลิเมตรและยาวประมาณ 10-30 เซนติเมตร ภายในบรรจุสารดูดซับซึ่งเป็นอนุภาคเล็กๆ ขนาดเส้นผ่าศูนย์กลางประมาณ 5-10 ไมครอน สามารถแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ Pellicular particle และ Porous particle ซึ่ง Pellicular particle นี้ จะประกอบไปด้วยอนุภาคกลมๆ เล็กๆ ของวัสดุประเภทแก้ว หรือโพลิเมอร์ที่ไม่มีรูพรุน และมี ซิลิกา อลูมินา หรือ เรซินเคลือบเป็นชั้นบางๆ บนผิวของอนุภาครองรับดังกล่าว ส่วน Porous particle จะเป็นอนุภาคที่มีรูพรุน ซึ่งประกอบไปด้วย ซิลิกา อลูมินา หรือเรซินกับซิลิกา โดยทั่วไปแล้วมักจะใช้ Porous particle เป็นสารดูดซับ นอกจากนี้ อนุภาคที่ใช้บรรจุอาจมีการเคลือบด้วยสารเคมีอื่นๆ อีก ซึ่งจะใช้งานในกรณีที่แตกต่างกันออกไป

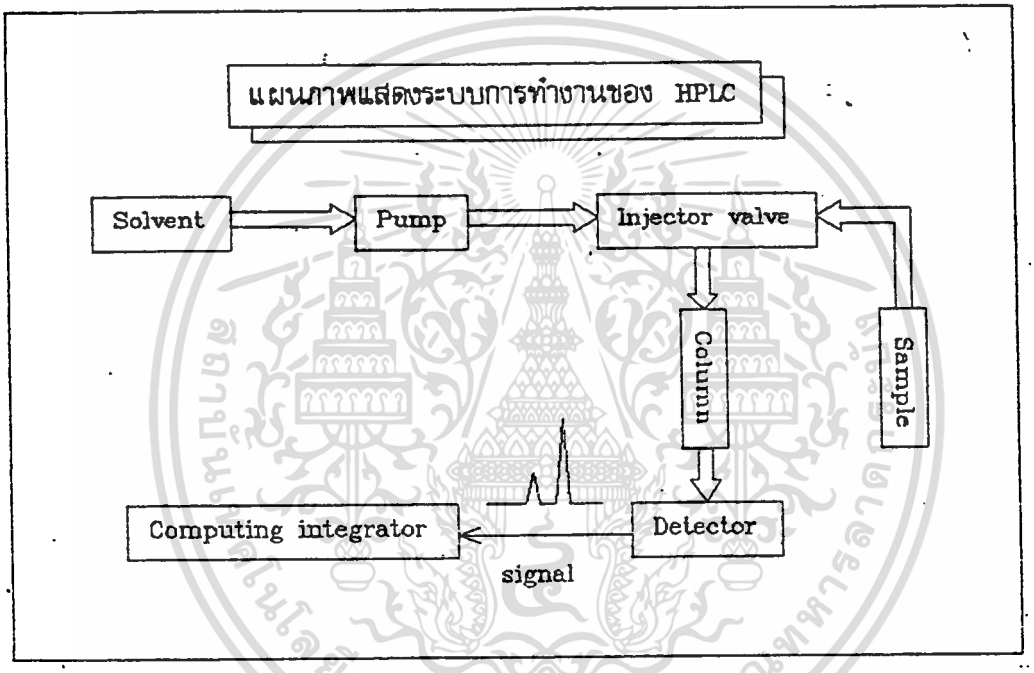
2.2.4 ตัวตรวจวัด(Detector) เป็นอุปกรณ์ที่ใช้สำหรับตรวจวัดสารต่าง ๆ ที่ไหลผ่านโดยอาศัยคุณสมบัติที่แตกต่างกันของสารแต่ละชนิด ตัวตรวจวัดที่ใช้กันมีอยู่หลายชนิด เช่น Fluorometry UV absorption IR absorption Refraction index เป็นต้น ในการเลือกตัวตรวจวัดเพื่อใช้งานจะพิจารณาจากชนิดของสารที่ต้องการตรวจสอบ การเลือกตัวตรวจวัดที่เหมาะสมกับสารที่ตรวจสอบจะสามารถตรวจสอบชนิดและปริมาณของสารได้ดีและถูกต้องมากขึ้น

2.2.5 โมบายเฟส(Mobile phase) คือตัวทำละลายที่ใช้พาสารตัวอย่างให้ไหลผ่านเข้าไปในคอลัมน์เพื่อแยกสารต่าง ๆ ในสารละลายออกจากกัน ก่อนที่จะผ่านเข้าไปยังตัวตรวจวัดต่อไป นอกจากนี้ ตัวทำละลายยังมีผลในการแยกสารออกจากกันภายในคอลัมน์ด้วย ดังนั้นในการเลือกตัวทำละลายที่เหมาะสมจะช่วยให้สามารถวิเคราะห์สารได้ง่ายและถูกต้องยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 เครื่องบันทึกสัญญาณไฟฟ้า(Y-t Recoder) หรือเครื่องวิเคราะห์ผล (Computing Integrator) เป็นเครื่องมือที่ใช้ในการวิเคราะห์สัญญาณไฟฟ้าที่ได้จากตัวตรวจวัด เครื่องบันทึกสัญญาณไฟฟ้าเป็นเครื่องมืออย่างง่ายที่สุดที่ใช้ในการวิเคราะห์ โดยจะแสดงผลเป็นกราฟสัญญาณไฟฟ้าที่วัดได้เทียบกับเวลา ซึ่งในการวิเคราะห์จะต้องใช้การวัดเวลาที่เกิดพีค การวัดความสูงหรือหาพื้นที่ใต้พีค และนำไปเปรียบเทียบกับสารมาตรฐานเพื่อวิเคราะห์ชนิดและความเข้มข้นของสารต่อไป

สำหรับเครื่องวิเคราะห์ผลเป็นเครื่องมือที่พัฒนาขึ้นมาใช้กับงานวิเคราะห์สัญญาณไฟฟ้าที่ได้จากตัวตรวจวัดโดยตรง ทำให้สามารถวิเคราะห์สารต่างๆ ได้อย่างสะดวกและรวดเร็วขึ้นมาก มีความถูกต้องแม่นยำสูงและยังมีชีดความสามารถในการวิเคราะห์ได้หลายรูปแบบอีกด้วย



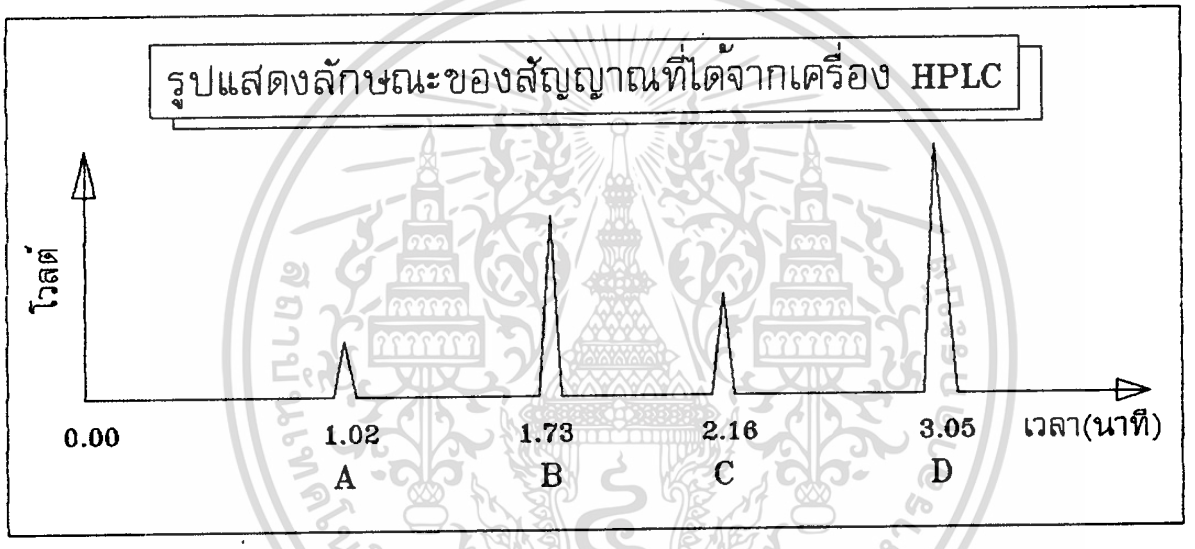
รูปที่ 2.1 แสดงระบบการทำงานของ HPLC

จากรูปแสดงถึงระบบการทำงานของ HPLC ซึ่งมีขั้นตอนการทำงานโดยเริ่มจาก ตัวปั๊มท์ซึ่งทำหน้าที่สูบตัวทำละลายผ่านวาล์วฉีดสารไปยังคอลัมน์และไหลผ่านไปที่รองรับเก็บสารละลาย เป็นการเตรียมพร้อมที่จะทำการวิเคราะห์ และเมื่อต้องการวิเคราะห์สารละลายตัวอย่าง จะฉีดสารละลายตัวอย่างนั้นเข้าไปที่วาล์วฉีดสาร ตัวทำละลายจะพาสารละลายตัวอย่างผ่านเข้าสู่คอลัมน์ ซึ่งทำหน้าที่เป็นตัวดูดซับ ทำให้สารชนิดต่างๆ ที่ผสมกันอยู่ถูกแยกออกจากกันด้วยกระบวนการของโครมาโตกราฟี สารใดที่ถูกดูดซับน้อยจะผ่านคอลัมน์ออกมาก่อน ส่วนสารใดที่ถูกดูดซับมากก็จะออกมาทีหลัง เมื่อสารแต่ละชนิดไหลออกมาจากคอลัมน์แล้ว ก็จะผ่านเข้าไปที่ตัวตรวจวัด และจะได้สัญญาณเอาท์พุทออกมา จากสัญญาณที่ได้นี้จะส่งต่อไปที่เครื่องบันทึกสัญญาณ หรือเครื่องวิเคราะห์ผล เพื่อวิเคราะห์หาชนิดและปริมาณของสารต่างๆ ที่ผสมอยู่ในสารละลายตัวอย่างดังกล่าว โดยการเปรียบเทียบกับข้อมูลของสารมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การวิเคราะห์ผลจากสัญญาณของ HPLC

ในการวิเคราะห์สารละลายตัวอย่างด้วยวิธี HPLC ก็เพื่อหาว่าสารตัวอย่างประกอบด้วยสารชนิดใดบ้างและมีปริมาณเท่าใด ซึ่งเทคนิค HPLC นี้จะแยกสารแต่ละชนิด ออกมาจากสารละลายตัวอย่างที่เวลาต่างๆ กัน และสามารถตรวจวัดได้โดยผ่านเข้าไปในตัวตรวจวัดที่เหมาะสม เมื่อสารแต่ละตัวไหลผ่านตัวตรวจวัดก็จะปรากฏเป็นพีคสัญญาณขึ้นมา ขนาดของพื้นที่ใต้พีคจะมากน้อยแค่ไหนก็ขึ้นอยู่กับปริมาณของสารที่ไหลผ่านและคุณสมบัติที่ตอบสนองกับชนิดของตัวตรวจวัดที่ใช้ด้วย ดังนั้นในการวิเคราะห์สารตัวอย่างใดๆ ถ้ามีการเลือกตัวตรวจวัดที่เหมาะสมก็จะช่วยให้การวิเคราะห์มีความแม่นยำสูง



รูปที่ 2.2 แสดงลักษณะของพีคสัญญาณของสารตัวอย่าง ซึ่งประกอบด้วยสาร 4 ชนิด คือ A B C และ D

จากรูปจะเป็นการแสดงลักษณะพีคสัญญาณที่ได้จากเครื่อง HPLC ซึ่งจะเห็นได้ว่าในสารละลายตัวอย่างประกอบไปด้วยสารต่าง ๆ ผสมกันอยู่ 4 ชนิด คือ A B C และ D ในการตรวจสอบชนิดของสารนั้นจะอาศัยหลักการที่ว่า สารชนิดเดียวกันจะมีค่ารีเทนชันไทม์เท่ากันเสมอ ภายใต้การควบคุมตัวแปรอื่นๆ ที่เหมือนกันทุกอย่าง เช่นชนิดของตัวทำละลายที่ใช้ ชนิดของคอลัมน์ อุณหภูมิ ฯลฯ และใช้ค่ารีเทนชันไทม์ในการบ่งบอกชนิดของสารที่เจือปนอยู่ โดยการเปรียบเทียบกับสารมาตรฐานที่ทราบชนิด และค่ารีเทนชันไทม์ที่หาได้ภายใต้การการควบคุมตัวแปรต่างๆ ชุดเดียวกันกับที่ใช้ในสารตัวอย่าง ดังนั้นในการตรวจสอบเพื่อหาสารชนิดใดๆ ที่ผสมอยู่ จำเป็นต้องมีข้อมูลของสารมาตรฐานมาใช้ในการเปรียบเทียบเสมอ สมมติว่า มีสารมาตรฐานคาเฟอีนซึ่งมีค่ารีเทนชันไทม์ 1.73 นาที ถ้าดูจากกราฟสัญญาณในแผนภาพ จะบอกได้ว่าสาร B คือ สารคาเฟอีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม ในทางปฏิบัติ ค่ารีเทนชันไทม์ที่อ่านได้จากสารเดียวกันอาจมีค่าที่เบี่ยงเบนไป จึงใช้วิธีตรวจสอบแบบเปิดช่วงเวลาเอาไว้ โดยคิดช่วงของการเบี่ยงเบนเป็นเปอร์เซนต์ของค่ารีเทนชันไทม์ เช่น คิดการเบี่ยงเบน 10% สารมาตรฐานที่มีค่ารีเทนชันไทม์ 1.00 นาที จะเปิดช่วงเวลาเป็น (1.00-0.10) = 0.90 นาที จนถึง (1.00+0.10) = 1.10 นาที จะเห็นได้ว่าถ้าค่ารีเทนชันไทม์ยิ่งมาก ช่วงเวลาที่ใช้ก็จะกว้างมากขึ้นไปด้วย เพื่อให้สอดคล้องกับการเลื่อนไปของพีคจริงๆ ในทางปฏิบัติ สำหรับปัญหาที่มักเกิดขึ้นได้บ่อยๆ ในกรณีของการตรวจสอบแบบนี้ก็คือ มีพีคหลายพีคในช่วงดังกล่าว ซึ่งอาจใช้วิธีเลือกพีคที่ใกล้เคียงค่ารีเทนชันไทม์มาตรฐานมากที่สุด หรือ เปลี่ยนตัวทำละลายใหม่เพื่อแยกพีคที่ไม่ต้องการออก แต่ก็ต้องหาค่ารีเทนไทม์ของสารมาตรฐานที่จะใช้ตรวจสอบใหม่ด้วยเช่นกัน นอกจากนี้ก็ยังมีวิธีตรวจสอบเทียบกับสารอ้างอิง ซึ่งจะช่วยให้มีความแม่นยำในการตรวจสอบชนิดของสารสูงขึ้น ซึ่งจะสามารถช่วยแก้ปัญหาในกรณีที่มีการเลื่อนค่ารีเทนชันไทม์ออกไปทั้งชุด

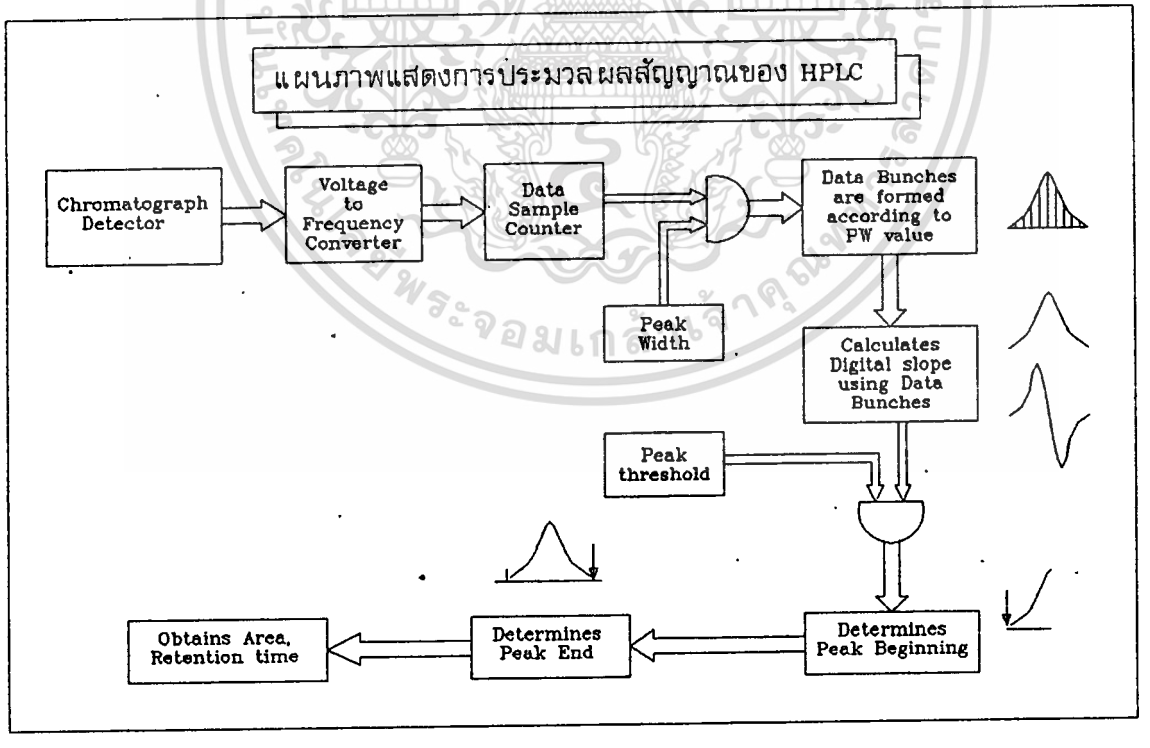
สำหรับในกรณีของการตรวจหาปริมาณหรือความเข้มข้นที่เจือปนอยู่ในสารตัวอย่างนั้น มีวิธีคำนวณหาได้ 2 วิธี วิธีแรกคือการคำนวณหาจากความสูงพีคสัญญาณของสารตัวอย่างเทียบกับความสูงของพีคสัญญาณสารมาตรฐานที่ทราบค่าความเข้มข้นแล้ว ในกรณีนี้เป็นเทคนิคการคิดอย่างง่ายๆ ค่าความเข้มข้นที่หาได้จะเป็นค่าประมาณอย่างหยาบๆ มีความผิดพลาดได้มาก ส่วนอีกวิธีหนึ่งก็คือการคำนวณหาจาก พื้นที่ใต้พีคสัญญาณของสารตัวอย่าง เทียบกับพื้นที่ใต้พีคสัญญาณของสารมาตรฐานที่ทราบความเข้มข้นแล้ว ในทางปฏิบัติ อาจใช้กราฟสัญญาณจาก Y-t Recoder มาวัดความสูงของพีคสัญญาณแล้วนำมาคำนวณ หรือ ในกรณีของการหาพื้นที่ใต้พีคนั้น โดยอาจเลี่ยงมาใช้การเปรียบเทียบน้ำหนักของกระดาษที่ตัดเป็นรูปพีคสัญญาณระหว่าง พีคสารตัวอย่างกับสารมาตรฐานแทน วิธีนี้จะเสียเวลามาก ตามความเป็นจริงควรจะมีความผิดพลาดน้อยกว่าวิธีวัดความสูง แต่ในทางปฏิบัตินั้น ความละเอียดหรือความถูกต้องยังขึ้นอยู่กับการตัดกระดาษ ชนิดของกระดาษ และการชั่งน้ำหนักด้วย

ในปัจจุบัน คอมพิวเตอร์สามารถช่วยคำนวณหาพื้นที่ของกราฟสัญญาณและรายงานการวิเคราะห์ให้เสร็จสมบูรณ์ได้ โดยใช้เวลาในการประมวลผลรวดเร็วมาก อีกทั้งยัง มีความถูกต้องแม่นยำสูง และเรียกเครื่องที่ออกแบบมาสำหรับใช้งานวิเคราะห์ของ HPLC นี้ว่าเครื่องประมวลผลซึ่งสามารถใช้เครื่องนี้แทนเครื่องบันทึกสัญญาณได้เลย สำหรับเทคนิคในการคำนวณหาความเข้มข้นนั้นยังมีรายละเอียดอีกมาก เช่น ต้องมีการปรับเทียบ (Calibration) ข้อมูลของความเข้มข้นของสารมาตรฐานหลายๆ ค่า เพราะความสัมพันธ์ระหว่างพื้นที่ใต้พีคกับความเข้มข้นอาจมีความสัมพันธ์ที่ไม่เป็นเชิงเส้นก็ได้ ดังนั้นจึงต้องมีการคำนวณเพื่อปรับแก้ค่าต่างๆ ให้ถูกต้องตามความเป็นจริงมากที่สุด นอกจากนี้ ปัญหาต่างๆ เช่นการหาพื้นที่ของพีคที่มีการซ้อนกันหรือพีคที่มีไหล่ ปัญหาของการเลื่อนไปของระดับฐานพีค ก็จะมีเทคนิคในการแก้ไขแตกต่างกันไป ในเครื่องประมวลผลบางรุ่นสามารถกำหนดเทคนิคการแก้ไขได้หลายแบบในบางครั้งเราอาจแก้ไขปัญหาเหล่านี้ได้โดยวิธีอื่น ซึ่งอาจให้ผลการวิเคราะห์ที่ถูกต้องมากขึ้น เช่น ในกรณีที่มีพีคซ้อนกัน อาจเปลี่ยนตัวโมบายเฟสหรือชนิดของคอลัมน์ที่ใช้ให้เหมาะสมก็สามารถที่จะแยกพีคสัญญาณที่ซ้อนกันอยู่ออกจากกันได้

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

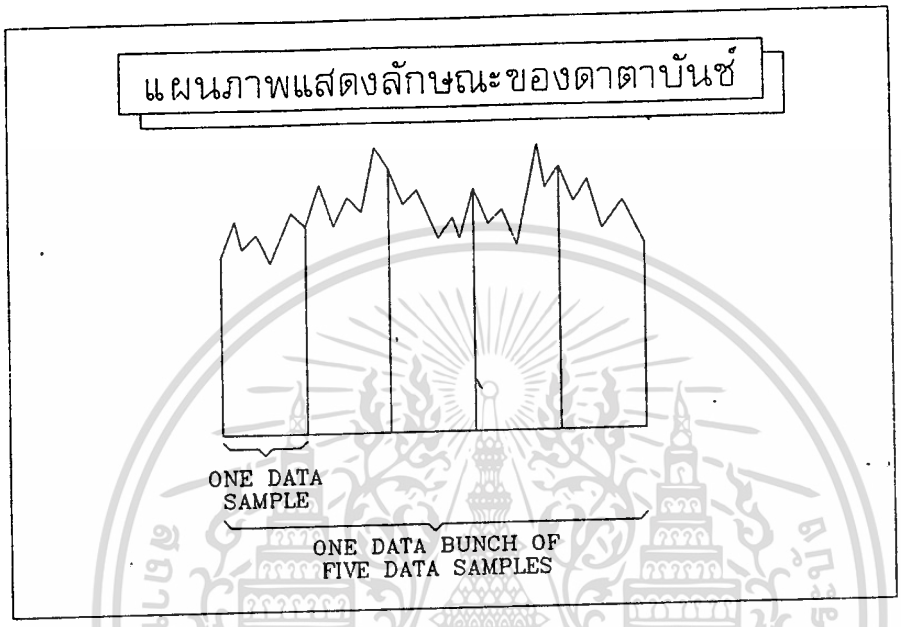
บทที่ 3
หลักการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ในการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC ดังรูปที่ 3.1 จะเริ่มจากการนำเอาสัญญาณไฟฟ้าจากตัวตรวจวัดซึ่งเป็นสัญญาณแบบอนาลอกมาเปลี่ยนเป็นข้อมูลดิจิตอลโดยใช้วงจรเปลี่ยนแรงดันไฟฟ้าเป็นความถี่ และนับจำนวนพัลส์(Pulse)ทั้งหมดในช่วงเวลาทุก ๆ 1/50 วินาที เก็บเป็นข้อมูลย่อยเรียกว่า ดาตาแซมเปิล(Data Sample) และนำดาตาแซมเปิลหลายตัวมารวมกันเป็นข้อมูลที่มีขนาดใหญ่ขึ้นเรียกว่า ดาตาบันช์(Data Bunches) ซึ่งจะมีพีควิดท์(Peak Width)เป็นตัวกำหนดจำนวนข้อมูลย่อยในหนึ่งดาตาบันช์ และนำค่าดาตาบันช์มาคำนวณเพื่อหาความชันของสัญญาณเพื่อกำหนดจุดเริ่มต้นของพีคสัญญาณ โดยมีค่าระดับการตรวจจับ(Peak Threshold) ค่าหนึ่ง ซึ่งกำหนดไว้เพื่อเปรียบเทียบหาจุดเริ่มต้นของสัญญาณ การกำหนดค่าระดับการตรวจจับที่เหมาะสมจะช่วยลดการตรวจจับพีคที่เกิดจากสัญญาณรบกวน จากนั้นจะหาค่ารีเทนชันไทม์ของพีคสัญญาณจากเวลาที่เริ่มฉีดสารจนถึงจุดยอดของพีคที่พบและหาพื้นที่ใต้พีคจากผลบวกของค่าสัญญาณทั้งหมดที่อยู่เหนือเส้นระดับฐานพีค(Baseline) โดยเริ่มจากจุดเริ่มต้นของพีคไปจนถึงจุดปลายพีค จากค่ารีเทนชันไทม์และพื้นที่ใต้พีคที่ได้จะนำไปตรวจสอบชนิดและปริมาณความเข้มข้นของสารได้ โดยการเปรียบเทียบกับสารมาตรฐาน



รูปที่ 3.1 แสดงการประมวลผลสัญญาณของ HPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

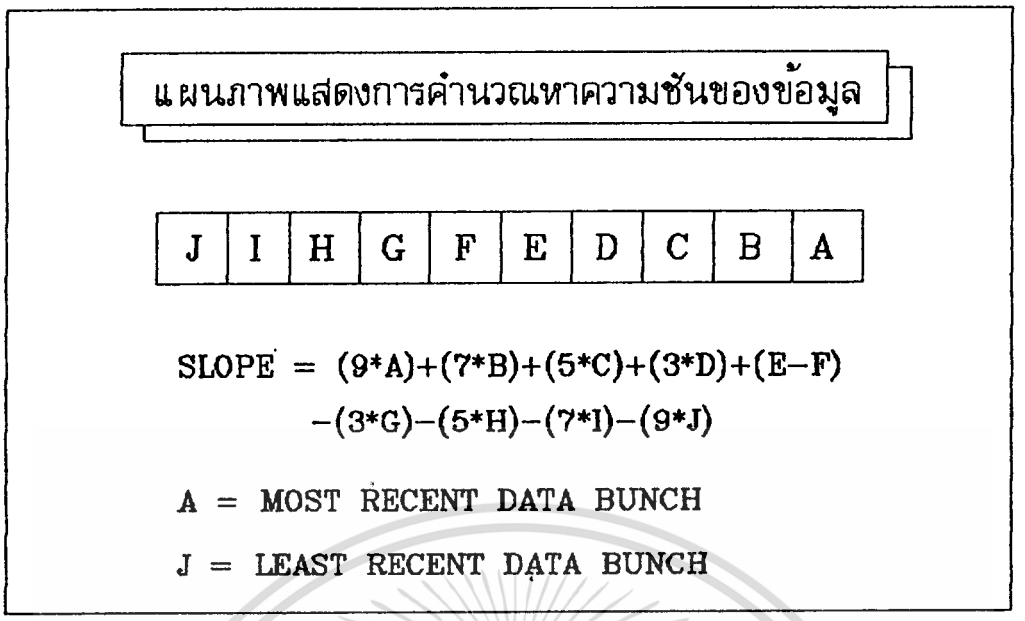


รูปที่ 3.2 แสดงดาตาบันช์ที่เกิดจากดาตาแซมเปิล 5 ชุด มารวมกัน

ดาตาบันช์เป็นข้อมูลสำคัญที่จะนำไปใช้ในการวิเคราะห์ผลในขั้นถัดไป ซึ่งในชุดของฮาร์ดแวร์ที่สร้างขึ้นนี้จะมีไมโครคอนโทรลเลอร์ 8751 เป็นตัวจัดการข้อมูลในเบื้องต้น เมื่ออ้างถึงแผนภาพการประมวลผลสัญญาณของ HPLC (รูปที่ 3.1) ชุดเก็บและรับส่งข้อมูล(Data Acquisition System) จะทำงานตั้งแต่การแปลงแรงดันไฟฟ้าเป็นความถี่ จนกระทั่งได้ข้อมูลซึ่งเป็นดาตาบันช์ออกมา ข้อมูลที่ได้สามารถเก็บและส่งต่อให้กับเครื่องไมโครคอมพิวเตอร์เพื่อนำไปวิเคราะห์ผลต่ออีกทีหนึ่ง

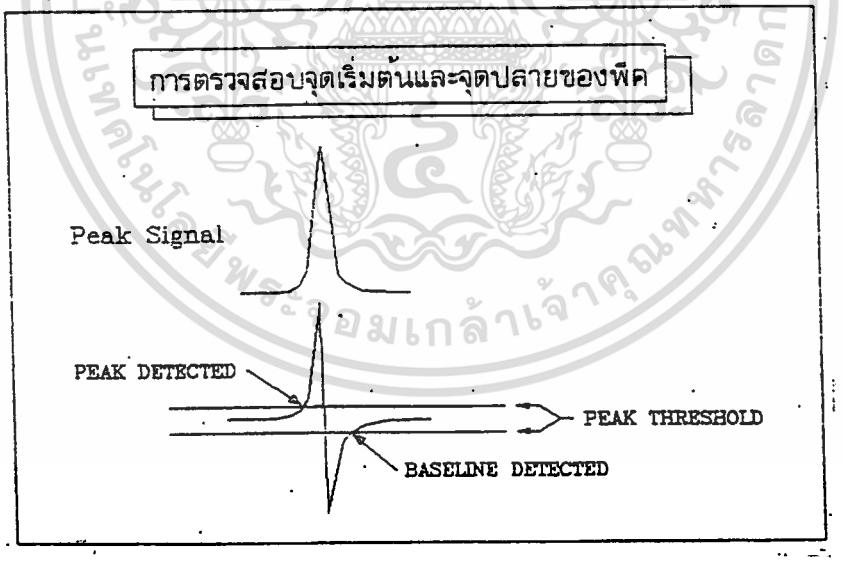
เมื่อไมโครคอมพิวเตอร์ได้รับข้อมูลเข้ามา ก็จะนำข้อมูลที่ได้ไปคำนวณเพื่อวิเคราะห์หาพีคโดยมีหลักการดังต่อไปนี้ คือ ในกรณีที่ข้อมูลมีสัญญาณรบกวนปะปนอยู่มาก สามารถกำหนดให้มีการกรองสัญญาณรบกวนออกด้วยเทคนิคทางซอฟต์แวร์ เรียกว่า Moving Average จากนั้นจะนำข้อมูลที่ได้ไปคำนวณหาความชันเพื่อใช้ในการหาจุดเริ่มต้น จุดยอด และจุดปลายของพีค ในการคำนวณหาความชันจะมีวิธีการคำนวณแบบเฉลี่ยน้ำหนักหลายจุด ซึ่งจะช่วยลดปัญหาการตรวจจับพีคขนาดเล็กๆ จากสัญญาณรบกวน วิธีการคำนวณค่าความชันของข้อมูลได้แสดงไว้ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.3 แสดงการคำนวณหาความชันของข้อมูล

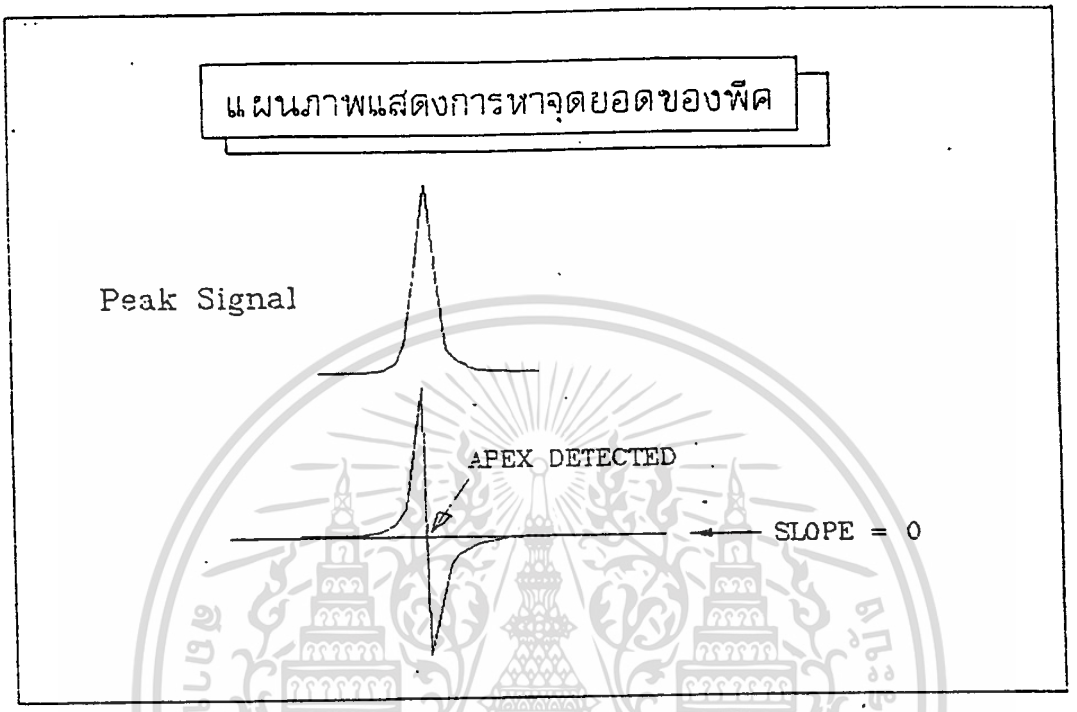
จากค่าความชันที่คำนวณได้ จะนำไปหาจุดเริ่มต้น และจุดปลายพีค โดยใช้การเปรียบเทียบกับระดับตรวจจับที่กำหนดไว้ ดังรูปที่ 3.4



รูปที่ 3.4 แสดงการตรวจหาจุดเริ่มต้นและจุดปลายของพีคจากความชันของสัญญาณ

เมื่อพิจารณาจากความชัน จุดเริ่มต้นของพีค จะเป็นจุดแรกของความชันที่ได่ขึ้นไปเหนือระดับตรวจจับด้านบวกและจุดปลายของพีคก็คือจุดแรกที่ความชันได่ขึ้นไปเหนือระดับตรวจจับด้านลบ สำหรับกรณีของจุดยอดนั้นจะพิจารณาจากจุดของความชันที่มีค่าเป็นศูนย์ ซึ่งจุดยอดนี้จะอยู่ระหว่างจุดเริ่มต้นกับจุดปลายของพีค

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงการหาจุดยอดของพีคจากความชัน

ในเบื้องต้นนี้ จะใช้จุดยอดของพีคเป็นตัวบ่งบอกตำแหน่งพีคของสารต่างๆ ที่ตรวจสอบ และเรียกเวลาที่เกิดพีคว่ารีเทนชันไทม์ จากค่ารีเทนชันไทม์สามารถใช้บอกชนิดของสารที่พบได้โดยการเปรียบเทียบกับ ค่ารีเทนชันไทม์ของสารมาตรฐานที่ทราบชนิด ภายใต้สภาวะการทดสอบเดียวกัน ซึ่งจะหาได้จากอนุพันธ์ของข้อมูลเทียบกับเวลา ดังรูปที่ 3.5

สำหรับในกรณีของการหาพื้นที่พีค จะใช้การคำนวณหาพื้นที่ทั้งหมดของพีคแต่ละพีค ในช่วงเวลาตั้งแต่จุดเริ่มต้นจนถึงจุดปลายของพีค แล้วหักออกด้วยค่าพื้นที่ใต้ฐานพีคนั้นๆ ในช่วงเวลาเดียวกัน ค่าพื้นที่พีคที่ได้นี้ จะนำไปคำนวณหาความเข้มข้นของสารได้โดยการเปรียบเทียบกับสารมาตรฐานชนิดเดียวกัน ที่ทราบความสัมพันธ์ระหว่างความเข้มข้นกับพื้นที่พีคแล้ว

ชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC จะแบ่งลักษณะการวิเคราะห์เป็น 3 วิธี คือ

1.Normal Method เป็นการวิเคราะห์หาค่ารีเทนชันไทม์ และค่าพื้นที่ของพีคแต่ละพีคที่ตรวจพบ

2.External Standard Method เป็นการตรวจสอบ เพื่อหาชนิด และค่าความเข้มข้นของสารที่กำหนดของพีคที่ตรวจพบ โดยใช้การอ้างอิงเปรียบเทียบจากข้อมูลที่เป็นสารมาตรฐาน

3.Baseline Calibration Method เป็นการปรับหาค่า Baseline แบบอัตโนมัติ จากการ Run ตัวทำละลายเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์เพื่อหาชนิดและปริมาณของสาร สามารถกล่าวเป็นหลักการได้ดังนี้

การวิเคราะห์ชนิดของสาร ใช้หลักการเปรียบเทียบค่ารีเทนชันไทม์กับสารมาตรฐาน สามารถแบ่งได้ 2 ขั้นตอน คือ

ขั้นแรก หาค่ารีเทนชันไทม์ (Rt_{s1}) ของสารมาตรฐานที่เหมือนกับสารตรวจสอบ

ขั้นที่สอง ทดสอบสารตัวอย่างเพื่อหารีเทนชันไทม์ (Rt_1) ของพีคต่างๆที่พบ ภายใต้สภาวะเดียวกันกับการหาค่ารีเทนชันไทม์ของสารมาตรฐาน จากนั้นจึงเปรียบเทียบค่า Rt_1 แต่ละตัวที่พบว่าอยู่ในช่วง $Rt_{s1}-0.1Rt_{s1}$ ถึง $Rt_{s1}+0.1Rt_{s1}$ หรือไม่ (กรณีเปิดช่วงหน้าต่าง 10%) ซึ่งถ้าอยู่ในช่วงดังกล่าว จึงสรุปว่า สารนั้นคือสารเดียวกันกับสารมาตรฐานที่เป็นตัวกำหนดช่วงดังกล่าว และในกรณีที่มีสารหลายตัวอยู่ในช่วงเดียวกันให้ถือเอาตัวที่ใกล้ Rt_{s1} มากที่สุดเป็นหลัก

การวิเคราะห์ความเข้มข้นของสาร ใช้หลักการเปรียบเทียบพื้นที่พีคกับสารมาตรฐาน สามารถแบ่งเป็น 2 ขั้นตอน คือ

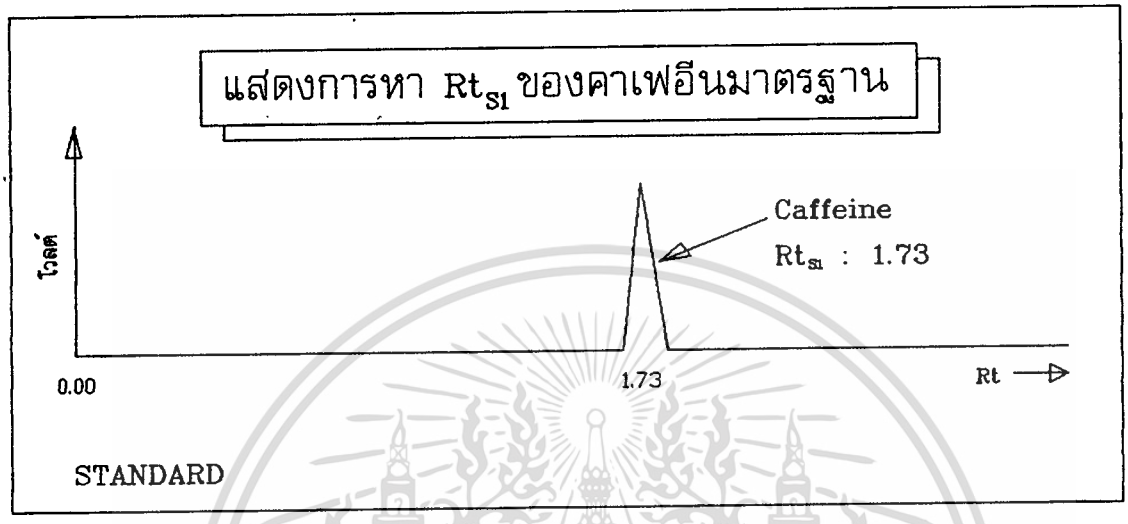
ขั้นแรก จะต้องนำสารมาตรฐานที่ทราบค่าความเข้มข้นแน่นอน มาหาพื้นที่พีค (Peak area) เพื่อเก็บเป็นข้อมูลในการคำนวณค่า เรสปอนด์แฟคเตอร์(Response factor) หรือเขียนย่อว่า Rf เมื่อ $Rf_{s1} = Area_{s1}/Conc._{s1}$ ค่าเรสปอนด์แฟกเตอร์นี้จะใช้ในการคำนวณเพื่อคำนวณหาความเข้มข้นของสารในขั้นตอนต่อไป

ขั้นที่สอง นำสารตัวอย่างมาทดสอบหาพื้นที่พีคเช่นเดียวกันกับสารมาตรฐาน ทั้งนี้ต้องทดสอบภายใต้สภาวะเดียวกัน การหาความเข้มข้นของสารสามารถคำนวณจาก $Conc._1 = Area_1/Rf_{s1}$ เมื่อ Rf_{s1} คือค่าเรสปอนด์แฟคเตอร์ที่คำนวณได้จากสารมาตรฐาน และค่า Rf_{s1} จะเป็นค่าเฉพาะของสารแต่ละชนิด ดังนั้น ถ้าต้องการหาปริมาณของสารหลายชนิดก็จะต้องหาค่า Rf_{s1} หลายตัวจากสารมาตรฐานแต่ละชนิดด้วยเช่นกัน

วิธีที่กล่าวถึงนี้ใช้ได้เฉพาะกรณีที่ กราฟความสัมพันธ์ระหว่างความเข้มข้นกับพื้นที่พีคของสารเป็นแบบเส้นตรงในช่วงความเข้มข้นที่ใช้วิเคราะห์เท่านั้น

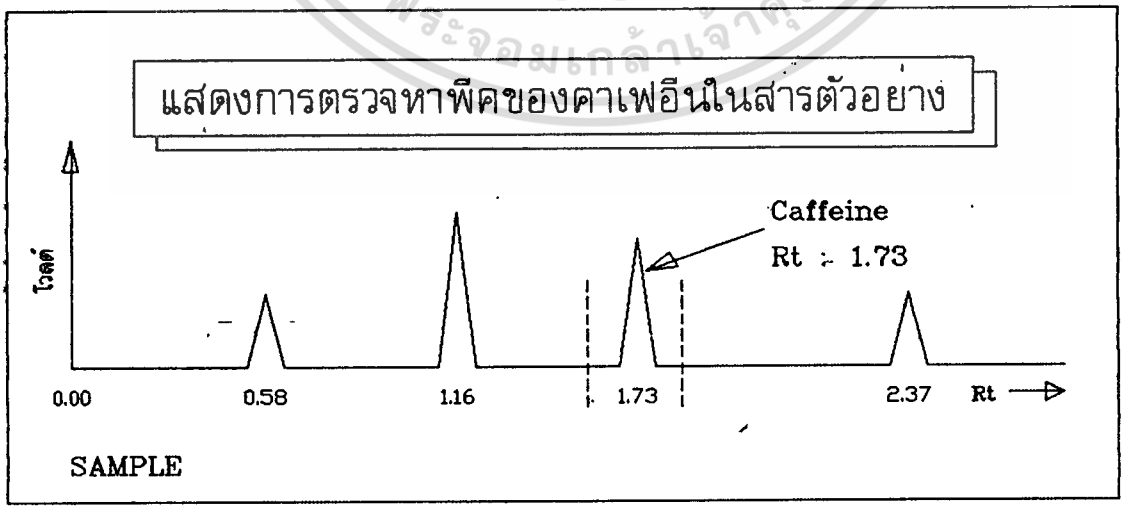
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างแสดงการตรวจสอบคาเฟอีน ในสารตัวอย่าง โดยเปรียบเทียบกับสารคาเฟอีนมาตรฐาน ดังรูปที่ 3.6



รูปที่ 3.6 แสดงการหา Rt_{S1} ของคาเฟอีนมาตรฐาน

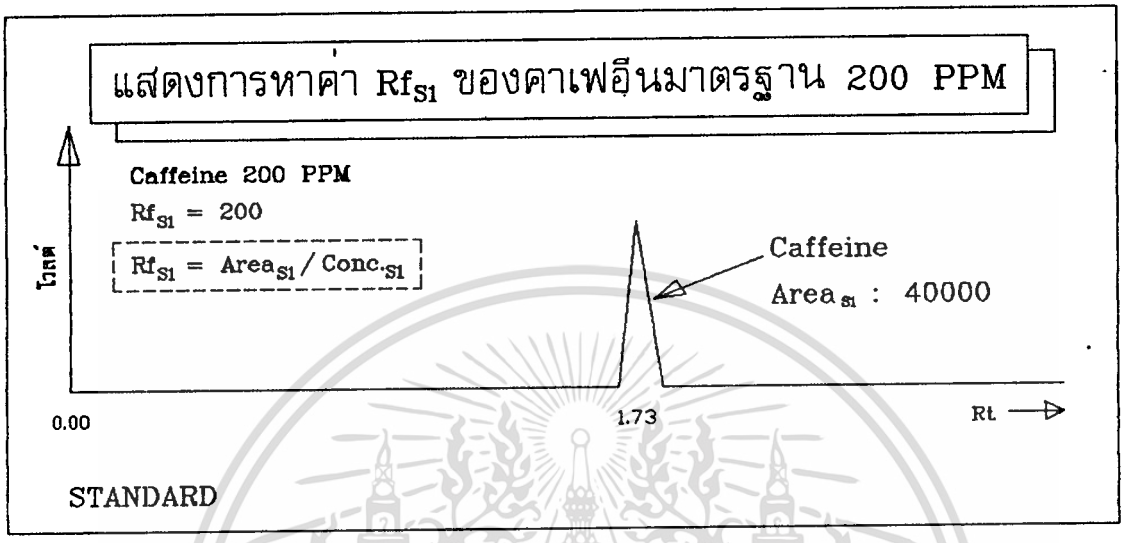
จากการตรวจวัดด้วยวิธี HPLC จะได้ลักษณะของพีคสัญญาณของสารตัวอย่างจากตัวตรวจวัด ดังรูปที่ 3.7 ซึ่งจะเห็นว่ามีพีคที่อยู่ในช่วงหน้าต่างของคาเฟอีน จึงสรุปว่าเป็นพีคของคาเฟอีน



รูปที่ 3.7 แสดงการตรวจหาพีคของคาเฟอีนในสารตัวอย่าง

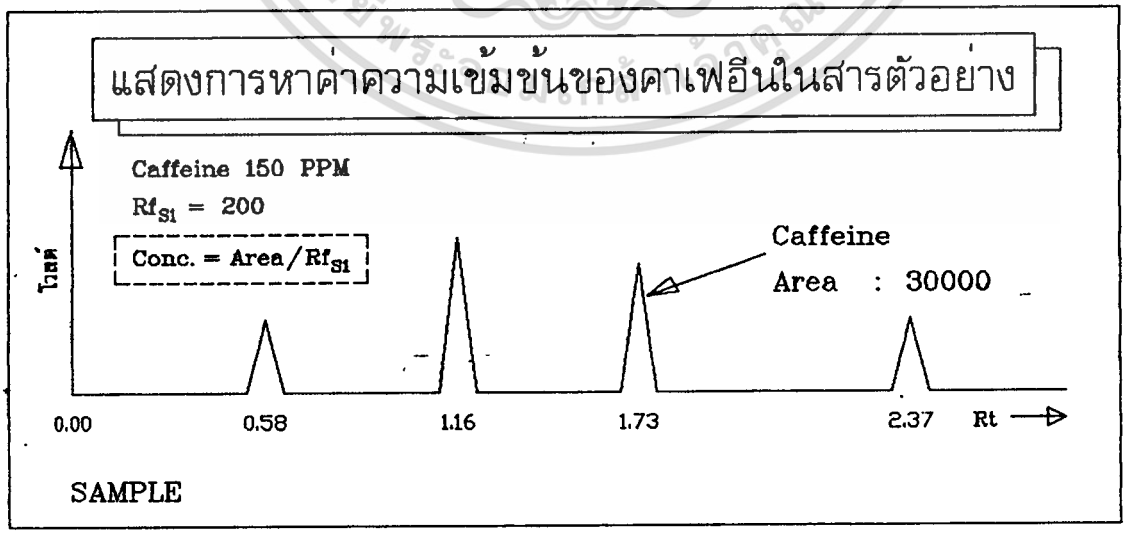
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างแสดงการตรวจหาความเข้มข้นของคาเฟอีน ในสารตัวอย่าง โดยเปรียบเทียบกับสารคาเฟอีนมาตรฐาน ดังรูปที่ 3.8



รูปที่ 3.8 แสดงการหาค่า Rf_{s1} ของคาเฟอีนมาตรฐาน 200 PPM

รูปที่ 3.9 แสดงการหาค่าความเข้มข้นของคาเฟอีนจากสารตัวอย่าง โดยคำนวณจาก ค่าพื้นที่ที่ได้หารด้วยค่าเรสปอนด์แฟกเตอร์ Rf_{s1}



รูปที่ 3.9 แสดงการหาค่าความเข้มข้นของคาเฟอีนในสารตัวอย่าง

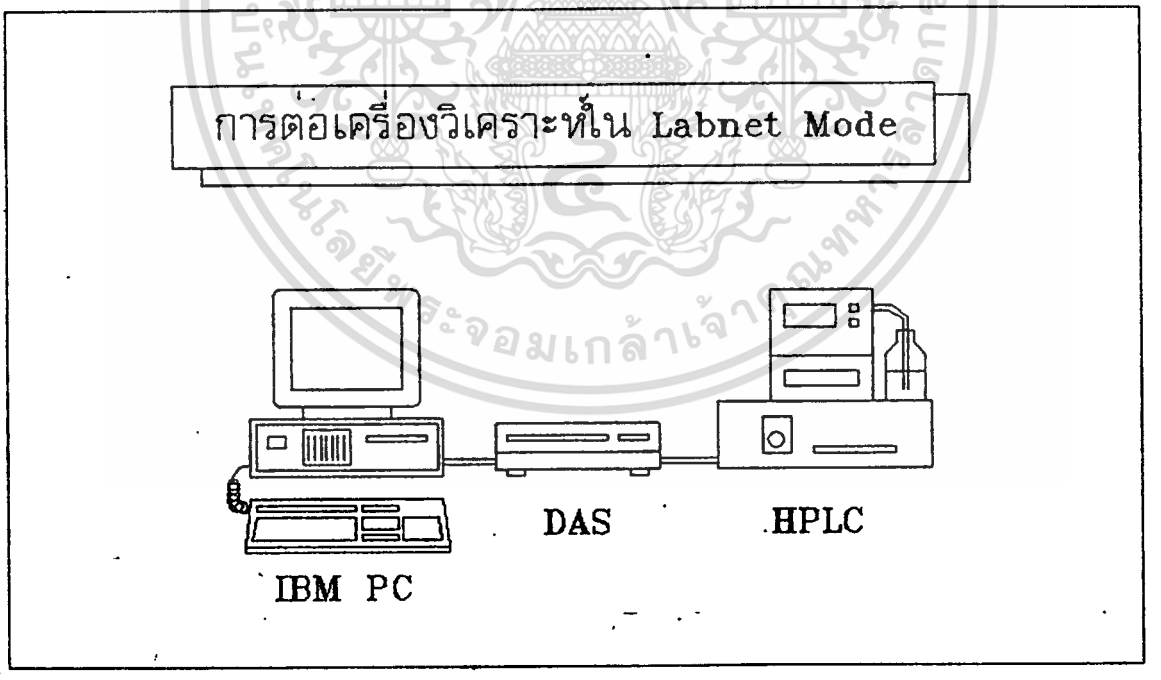
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การทำงานของชุดวิเคราะห์ผลที่พัฒนาขึ้น สามารถแบ่งตามลักษณะการทำงานได้เป็น 2 แบบ คือ การทำงานใน Labnet Mode ซึ่งเป็นการทำงานในลักษณะที่ต้องการเก็บและวิเคราะห์ผลข้อมูลในขณะที่เดินเครื่อง HPLC อยู่ และ การทำงานใน File Mode จะเป็นการทำงานซึ่งนำเอาข้อมูลที่ได้เก็บเป็นไฟล์ไว้แล้ว มาวิเคราะห์ในภายหลัง ดังมีรายละเอียดต่อไปนี้

การทำงานใน Labnet Mode

การทำงานใน Labnet Mode นี้ จะต้องมีชุดเก็บและรับส่งข้อมูล(Data Acquisition System) หรือ DAS ต่วงต่ออยู่กับเครื่อง HPLC และเครื่องไมโครคอมพิวเตอร์ เพื่อนำสัญญาณจากตัวตรวจวัดมาแปลงเป็นข้อมูลทางดิจิตอล แล้วส่งต่อให้กับไมโครคอมพิวเตอร์ดังรูปที่ 3.10 เมื่อต่อเครื่องทุกอย่างถูกต้องแล้วจึงเรียกโปรแกรมวิเคราะห์ชื่อ HPLC.EXE ขึ้นมาทำงาน โดยกำหนดการทำงานเป็น LabNet Mode และสามารถเก็บข้อมูลเป็นไฟล์ลงบนแผ่นจานแม่เหล็กซึ่งสามารถนำกลับมาวิเคราะห์ผลใหม่ได้โดยใช้การทำงานใน File Mode

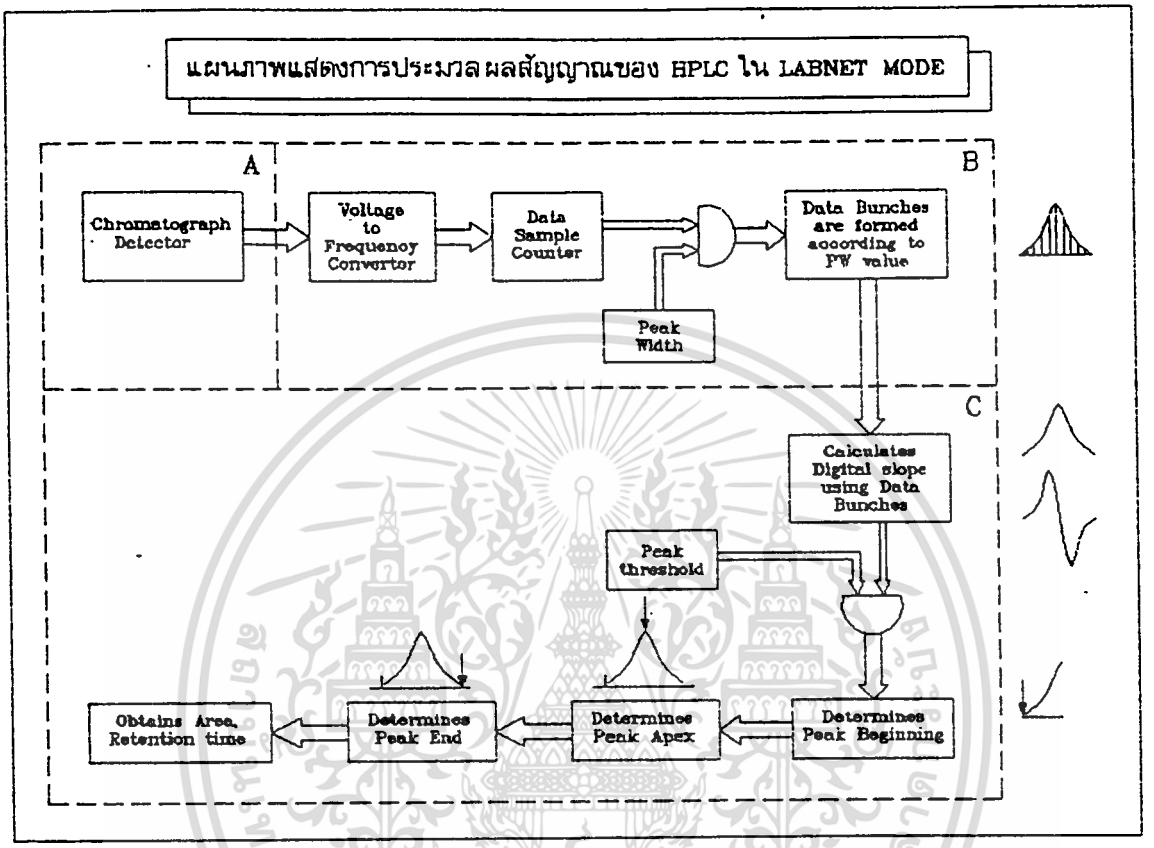


รูปที่ 3.10 แสดงลักษณะการต่อเครื่องวิเคราะห์ใน Labnet Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21395

ในกระบวนการวิเคราะห์ข้อมูลใน Labnet Mode สามารถเขียนเป็นแผนภาพแสดงการประมวลผลสัญญาณของ HPLC ได้ ดังรูปที่ 3.11



รูปที่ 3.11 แสดงการประมวลผลข้อมูลใน Labnet Mode

จากแผนภาพการแสดงการประมวลผลข้อมูลใน Labnet Mode สามารถแบ่งได้เป็น 3 ส่วน ดังนี้

A: ส่วนของตัวตรวจวัดที่ใช้ในเครื่อง HPLC ซึ่งจะเป็นตัววัดและส่งสัญญาณที่ได้ให้กับเครื่องเก็บและรับส่งข้อมูล(B)

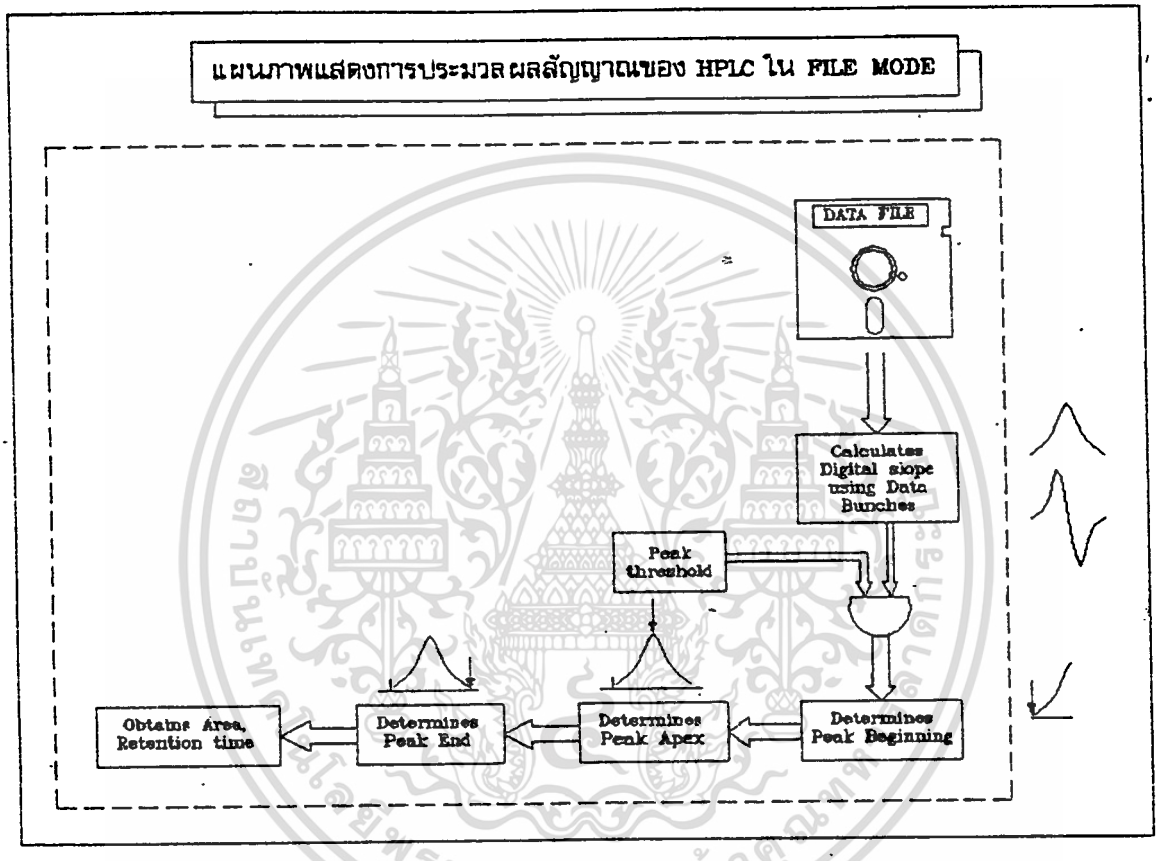
B: ส่วนของเครื่องเก็บและรับส่งข้อมูล(DAS) ซึ่งจะเป็นตัวแปลงสัญญาณอนาลอกให้เป็นข้อมูลดิจิตอล และส่งต่อให้กับเครื่องไมโครคอมพิวเตอร์โดยผ่านสาย RS232C

C: ส่วนของซอฟต์แวร์ที่ใช้วิเคราะห์ผลข้อมูลของ HPLC บนเครื่องไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานใน File Mode

การทำงานใน File Mode นี้ เป็นการนำข้อมูลที่เก็บไว้ในแผ่นจานแม่เหล็กมาวิเคราะห์ ซึ่งการวิเคราะห์ในลักษณะนี้ สามารถที่จะเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ที่ใช้ในการวิเคราะห์ได้หลายรูปแบบตามต้องการ โดยที่มีลักษณะการวิเคราะห์ข้อมูลดังรูปที่ 3.12



รูปที่ 3.12 แสดงการประมวลผลข้อมูลใน File Mode

ใน File Mode นี้ จะมีกระบวนการวิเคราะห์ข้อมูลบนไมโครคอมพิวเตอร์เหมือนกับใน LabNet Mode ทุกประการ โดยข้อมูลที่จะนำมาวิเคราะห์ในรูปแบบของ File Mode นั้น จะเป็นไฟล์ข้อมูลที่บันทึกจาก LabNet Mode ซึ่งข้อมูลที่เก็บบันทึกไว้นี้เป็นข้อมูลของดาตาบันช์ที่รับมาจากเครื่องเก็บและรับส่งข้อมูล ดังนั้นจึงสามารถนำเอาข้อมูลที่เก็บได้กลับมาวิเคราะห์ใหม่ โดยเลือกเปลี่ยนค่าพารามิเตอร์ที่ใช้ในการวิเคราะห์ใหม่ได้ ซึ่งมีประโยชน์ในการกำหนดการวิเคราะห์ให้เหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ในหัวข้อนี้จะกล่าวถึงหลักการที่ใช้ในการออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC ซึ่งสามารถสรุปได้เป็น 4 หัวข้อ เริ่มจากการออกแบบโดยรวม การออกแบบในส่วนของฮาร์ดแวร์ ซอฟต์แวร์ และการเชื่อมต่อระหว่างชุดฮาร์ดแวร์กับซอฟต์แวร์ โดยกล่าวเป็นหลักการโดยสรุปได้ดังนี้

4.1 หลักการออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ในการออกแบบชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC นั้น จะต้องศึกษาข้อมูลต่างๆ ที่เกี่ยวข้อง ตั้งแต่ ระบบการทำงานของ HPLC คุณสมบัติของตัวตรวจวัดที่ใช้ วิธีการวิเคราะห์สัญญาณ หลักการทำงานของเครื่องวิเคราะห์ผล(Computing Integrator) ฯลฯ จากนั้นจึงนำข้อมูลที่ได้มาใช้ในการกำหนดรูปแบบและการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลต่อไป

ขั้นแรก จะพิจารณาการออกแบบเป็น 4 ประการ คือ การนำข้อมูลเข้ามาวิเคราะห์ การประมวลผลและวิเคราะห์ข้อมูล การแสดงผลการวิเคราะห์ และการเก็บบันทึกข้อมูล

4.1.1 การนำข้อมูลเข้ามาวิเคราะห์

เนื่องจากการวิเคราะห์จะต้องใช้สัญญาณไฟฟ้าที่ได้จากตัวตรวจวัดของ HPLC ดังนั้นจึงต้องเปลี่ยนสัญญาณไฟฟ้าที่ได้ให้เป็นข้อมูลที่คอมพิวเตอร์สามารถใช้งานได้ โดยใช้วงจรอิเล็กทรอนิกส์มาแปลงข้อมูลจากสัญญาณไฟฟ้าซึ่งเป็นสัญญาณอนาลอกให้เป็นข้อมูลชนิดดิจิตอล ซึ่งชุดวงจรอิเล็กทรอนิกส์นี้ เรียกว่าชุดเก็บและรับส่งข้อมูล จากข้อมูลที่แปลงได้จะส่งต่อให้กับเครื่องไมโครคอมพิวเตอร์โดยผ่านสาย RS232C ซึ่งจะรับส่งข้อมูลรหัสแอสกีแบบอนุกรม กล่าวคือเป็นการรับส่งแบบเรียงลำดับทีละตัว ต่อจากนั้นเครื่องไมโครคอมพิวเตอร์ก็จะรับข้อมูลเพื่อนำไปวิเคราะห์และรายงานผลต่อไป

4.1.2 การประมวลผลและวิเคราะห์ข้อมูล

จากการศึกษากระบวนการวิเคราะห์ข้อมูลของ HPLC และการทำงานของเครื่องวิเคราะห์ผล จะนำมาออกแบบการวิเคราะห์ข้อมูล โดยใช้โปรแกรมภาษาปาสคาลในการพัฒนาซอฟต์แวร์วิเคราะห์พีคจากข้อมูลที่ได้รับ ซึ่งออกแบบให้มีความสามารถในการประมวลผล และวิเคราะห์ข้อมูลในขณะที่กำลังรับข้อมูลจากเครื่อง HPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การแสดงผลการวิเคราะห์

เมื่อพิจารณาการวิเคราะห์ข้อมูลในรูปแบบต่างๆ แล้ว จะนำข้อมูลมาใช้ในการออกแบบการแสดงผลการวิเคราะห์ว่าควรแสดงผลการวิเคราะห์อย่างไร ซึ่งในที่นี้ได้ออกแบบให้สามารถรายงานผลการวิเคราะห์ได้ทั้งทางจอภาพและทางเครื่องพิมพ์(Printer) โดยเพิ่มขีดความสามารถในการแสดงผลการวิเคราะห์ทางจอภาพได้ทันที(Realtime Analysis & Realtime Display)

4.1.4 การเก็บบันทึกข้อมูล

เนื่องจากต้องการให้สามารถเก็บบันทึกข้อมูลที่ได้ลงบนแผ่นจานแม่เหล็ก เพื่อที่จะนำมาใช้วิเคราะห์ข้อมูลซ้ำในภายหลังได้อีก จึงออกแบบให้สามารถบันทึกและอ่านข้อมูลจากแผ่นจานแม่เหล็ก โดยจะเก็บเป็นไฟล์ข้อมูลชนิดเลขจำนวนเต็ม เพื่อให้สามารถนำข้อมูลไปประมวลผลได้ทันที ซึ่งไม่จำเป็นต้องแปลงข้อมูลอีก

4.2 การออกแบบชุดเก็บและรับส่งข้อมูล

การทำงานของชุดเก็บและรับส่งข้อมูล คือการเปลี่ยนสัญญาณไฟฟ้าที่ตรวจวัดซึ่งเป็นสัญญาณแบบอนาลอกให้เป็นข้อมูลทางดิจิตอล แล้วส่งให้กับเครื่องไมโครคอมพิวเตอร์ โดยจะส่งข้อมูลขนาดของสัญญาณในรูปของรหัสแอสกี ผ่านทางสาย RS232C ให้กับไมโครคอมพิวเตอร์

การทำงานสามารถแบ่งได้เป็น 8 ส่วนใหญ่ๆ คือ

4.2.1 ภาคกรองสัญญาณ

4.2.2 ภาคขยายสัญญาณ

4.2.3 ภาคเปลี่ยนแรงดันเป็นความถี่

4.2.4 ภาควงจรนับและคงสถานะข้อมูล

4.2.5 ภาคประมวลผลข้อมูล

4.2.6 ภาครับส่งข้อมูล

4.2.7 ภาคจ่ายไฟเลี้ยง

4.2.8 ภาคสร้างสัญญาณนาฬิกา

สัญญาณที่ต้องการวัด(อยู่ในช่วง -10mV - 1V) จะนำมาผ่านภาคกรองสัญญาณซึ่งเป็นตัวกรองสัญญาณความถี่ต่ำอันดับที่สอง เพื่อตัดสัญญาณรบกวนความถี่สูงออกไป จากนั้นจะนำสัญญาณที่กรองแล้วผ่านเข้าไปยังภาคขยายสัญญาณซึ่งมีอัตราการขยาย 10 เท่า นำสัญญาณที่ได้ไปยังส่วนเปลี่ยนแรงดันเป็นความถี่แบบเชิงเส้นซึ่งจะรับสัญญาณตั้งแต่ 0-10.0 V และให้ความถี่ออก 0-1.0 MHz ความถี่ที่ได้จะส่งไปนับด้วยวงจรนับและคงสถานะข้อมูลขนาด 16 บิท(เป็นวงจรขนาด 8 บิทสองชุด) โดยจะมีสัญญาณนาฬิกาขนาดคาบเวลา 1/50 วินาที เป็นตัวกำหนดการนับและคงสถานะข้อมูล และการอ่านข้อมูลที่นับได้มาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

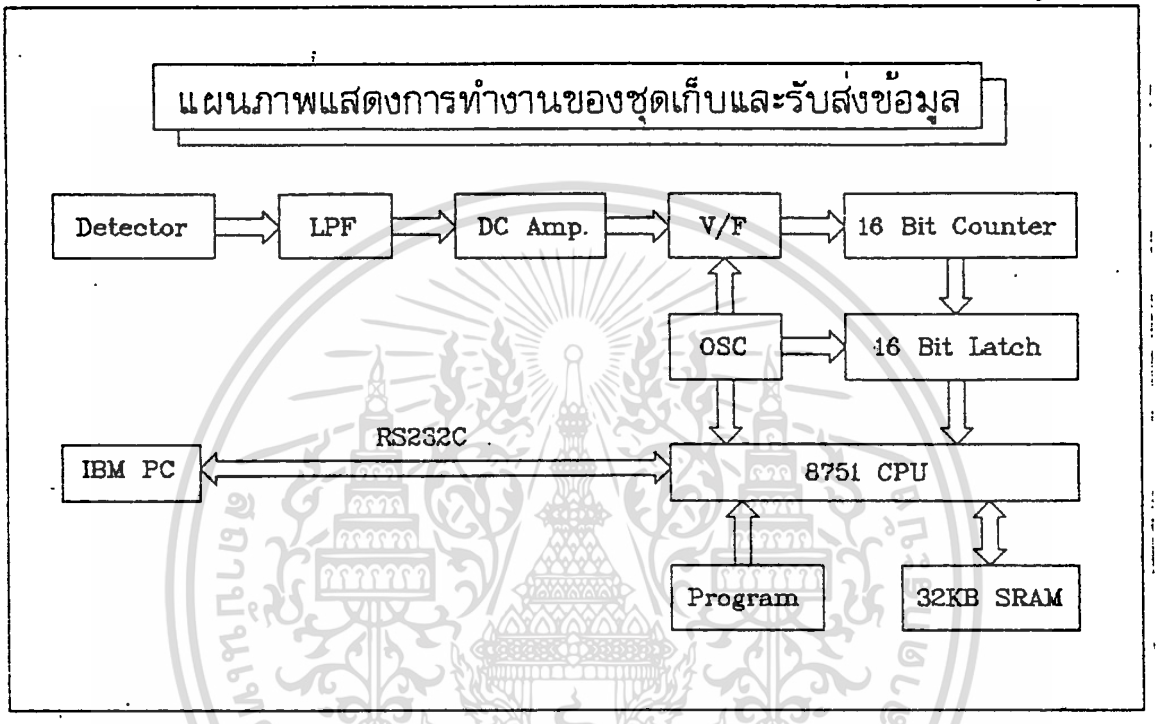
ภาคประมวลผลและภาครับส่งข้อมูลได้แสดงไว้ดังรูปที่ 4.1 โดยภาคประมวลผล จะมีไมโครคอนโทรลเลอร์ควบคุมการอ่าน การประมวลผลข้อมูล การเก็บบันทึกข้อมูล และการรับส่งข้อมูลให้กับไมโครคอมพิวเตอร์ การประมวลผลภายในไมโครคอนโทรลเลอร์ จะเริ่มจากสัญญาณอินเตอรัปต์ (Interrupt Signal) ซึ่งจะส่งมาทุกๆ 1/50 วินาที กำหนดให้มีการอ่านจากข้อมูลจากค่าที่นับได้จากวงจรนับและคงสถานะข้อมูลขนาด 16 บิท โดยจะผ่านเข้ามาทีละ 8 บิท จากนั้นจึงจะนำค่าที่นับได้ มาคำนวนหาขนาดของสัญญาณในช่วง 1/50 วินาทีเรียกดาตาแซมเปิล และนำดาตาแซมเปิลมารวมกันเป็นดาตาบันช์ และเก็บค่าที่ได้ในหน่วยความจำชนิดแรมขนาด 32 กิโลไบท์ เมื่อมีสัญญาณจากเครื่องไมโครคอมพิวเตอร์ส่งมาอ่านข้อมูล โปรแกรมจะเปลี่ยนค่าดาตาบันช์ซึ่งเป็นข้อมูลแบบไบนารีให้เป็นรหัสแอสกี และส่งให้กับเครื่องไมโครคอมพิวเตอร์ทางสาย RS232C แต่ถ้ายังไม่มีการอ่านข้อมูล โปรแกรมก็จะเก็บบันทึกข้อมูลต่อไปเรื่อยๆ

ในส่วนของภาครับส่งข้อมูลนั้น จะเป็นการสื่อสารข้อมูลแบบอนุกรม โดยใช้สาย RS232C เป็นสายรับส่งข้อมูล และในกรณีที่ต้องการป้องกันสัญญาณรบกวนระหว่างเครื่องไมโครคอมพิวเตอร์ กับชุดเก็บและรับส่งข้อมูลก็อาจเพิ่มส่วน Opto-isolated RS232C Interface ซึ่งเป็นวงจรส่งผ่านข้อมูลด้วยแสงเข้ากับสาย RS232C เพื่อป้องกันปัญหาไฟฟ้ารั่วหรือสัญญาณรบกวนจากเครื่องไมโครคอมพิวเตอร์มาทำความเสียหายให้กับเครื่องเก็บและรับส่งข้อมูล

แหล่งจ่ายไฟที่ใช้กับวงจรจะใช้แรงดันไฟฟ้า DC $\pm$15V จาก Dual Tracking Regulator สำหรับจ่ายให้กับภาคขยายสัญญาณ ภาคเปลี่ยนแรงดันไฟฟ้าเป็นความถี่ และแรงดันไฟฟ้าขนาด 5 โวลต์ ในการจ่ายให้กับภาควงจรนับและคงสถานะข้อมูลภาคประมวลผลข้อมูล และภาครับส่งข้อมูล

สัญญาณนาฬิกาจะมีอยู่ 3 ชุด คือ สัญญาณนาฬิกาความถี่ 2 เมกะเฮิร์ทช์ใช้กับวงจรเปลี่ยนแรงดันไฟฟ้าจาก 0-10 โวลต์ ไปเป็นเป็นความถี่ขนาด 0-1 เมกะเฮิร์ทช์ สัญญาณนาฬิกาความถี่ 50 เฮิร์ทช์ ใช้เป็นสัญญาณอินเตอรัปต์และคงสถานะข้อมูล และสัญญาณนาฬิกาความถี่ 11.0592 เมกะเฮิร์ทช์ สำหรับเป็นฐานเวลาของไมโครคอนโทรลเลอร์

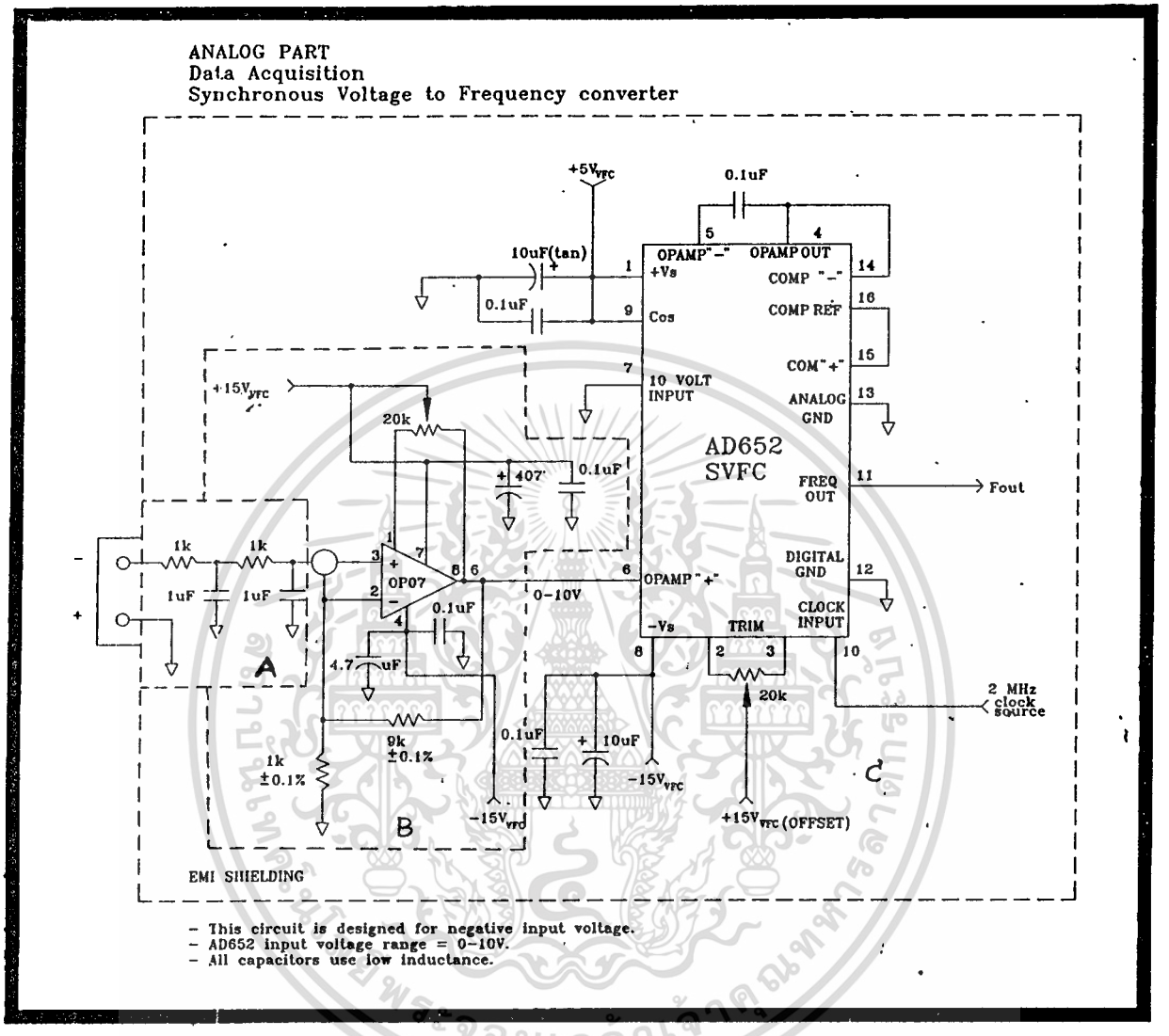
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การทำงานของชุดเก็บและรับส่งข้อมูล

จากรูปที่ 4.1 การทำงานของชุดเก็บและรับส่งข้อมูลจะเริ่มจาก สัญญาณที่ได้จากตัวตรวจวัดของเครื่อง HPLC จะนำมาผ่านภาคกรองสัญญาณ ซึ่งจะทำหน้าที่ตัดสัญญาณรบกวนความถี่สูงออกไป แล้วจึงนำไปขยายสัญญาณไฟตรงขึ้นไปอีก 10 เท่า จากนั้นจะส่งไปยังภาคเปลี่ยนแรงดันเป็นความถี่ และจะมีวงจรนับพัลส์ที่ได้เป็นข้อมูลขนาด 16 บิท โดยจะมีวงจรคงค่าข้อมูลทุก ๆ 1/50 วินาที ไมโครคอนโทรลเลอร์จะอ่านค่าข้อมูลทุก ๆ 1/50 วินาที ตามจังหวะที่คงค่าข้อมูล และต้องนำข้อมูลที่อ่านได้ไปคำนวณหาค่าขนาดสัญญาณที่แท้จริงออกมา เนื่องจากค่าข้อมูลที่อ่านได้จะมีค่ามากขึ้นเรื่อยๆ จากการนับพัลส์ของสัญญาณและจะมีค่ากลับไปเป็นค่าศูนย์ใหม่ได้เมื่อข้อมูลมีการนับเกินขนาดของข้อมูล 16 บิท ไมโครคอนโทรลเลอร์จึงมีโปรแกรมคำนวณเพื่อหาค่าขนาดของสัญญาณที่แท้จริงในแต่ละช่วงเวลา 1/50 วินาที ของทั้งสองกรณี และเก็บข้อมูลเป็นค่าไบนารีไว้ในหน่วยความจำชนิดแรม และเมื่อต้องการส่งต่อให้กับเครื่องไมโครคอมพิวเตอร์ โปรแกรมจะแปลงค่าข้อมูลที่เป็นแบบไบนารีให้เป็นข้อมูลในรูปของรหัสแอสกีก่อน แล้วจึงส่งให้กับไมโครคอมพิวเตอร์โดยผ่านทางสาย RS232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

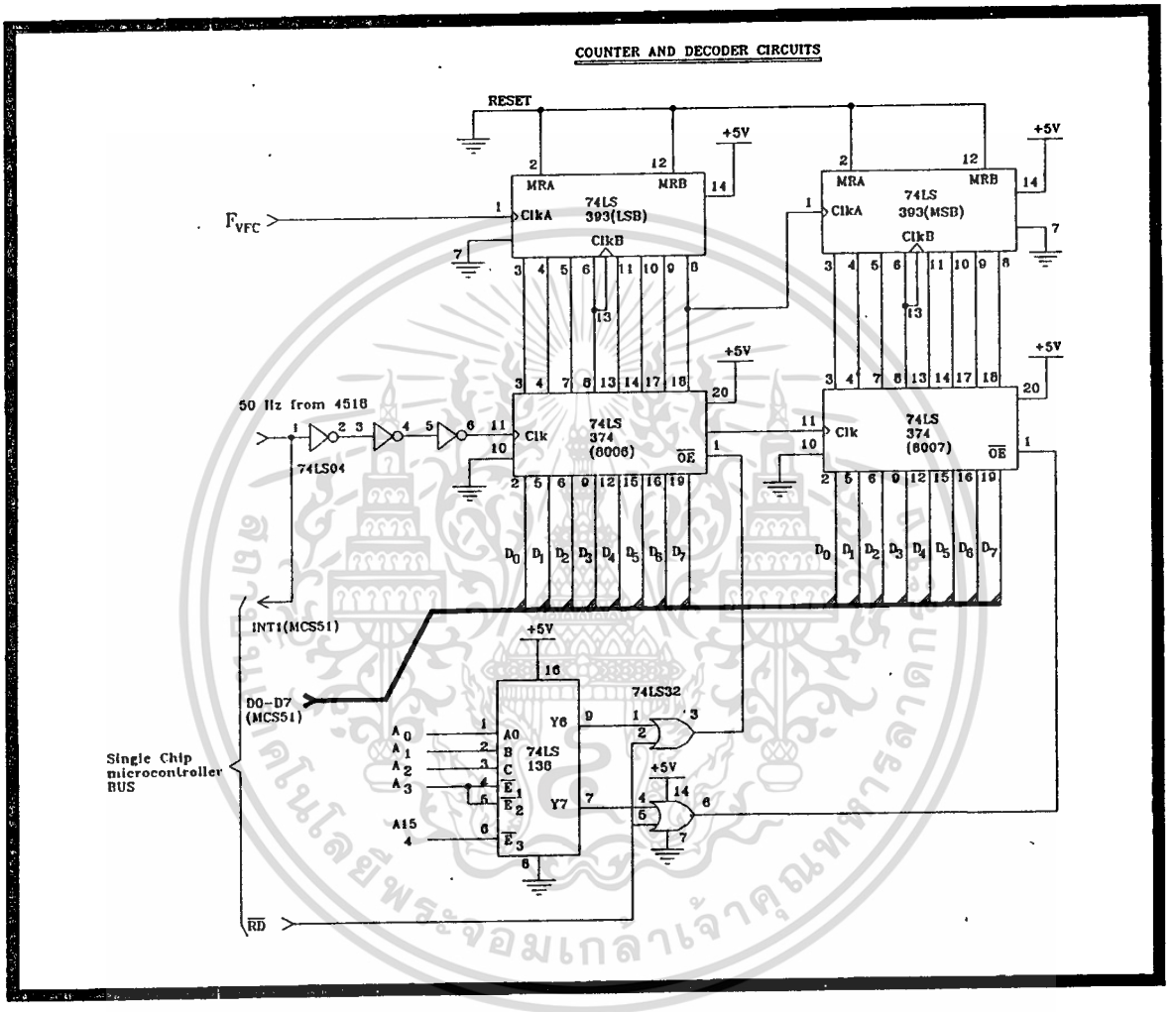


A : 2nd Lowpass filter, B : DC Amplifier, C : V-F Converter

รูปที่ 4.2 แสดงวงจรแปลงสัญญาณอนาลอกเป็นความถี่

จากรูปที่ 4.2 จะเป็นส่วนของวงจรอนาลอก ซึ่งจะทำหน้าที่แปลงสัญญาณที่ได้รับให้เป็นความถี่ จากวงจรจะเริ่มจากภาคกรองสัญญาณซึ่งเป็นวงจรกรองสัญญาณความถี่ต่ำอันดับที่ 2 ที่ประกอบด้วย RC สองชุด ใช้สำหรับตัดสัญญาณรบกวนความถี่สูงออกไป แล้วจึงนำไปขยายสัญญาณให้มีขนาดใหญ่ขึ้น 10 เท่า ซึ่งภาคขยายสัญญาณนี้ใช้ออปแอมป์เบอร์ OP07 ต่อเป็นวงจรขยายสัญญาณแบบ Noninverting Amplifier และส่งไปยังภาคเปลี่ยนแรงดันไฟฟ้าเป็นความถี่โดยมีไอซี AD652 เป็นตัวแปลงสัญญาณอนาลอกขนาด 0-10 โวลต์เป็นความถี่ 0-1 เมกะเฮิร์ทซ์ โดยใช้ความถี่อ้างอิง 2 เมกะเฮิร์ทซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

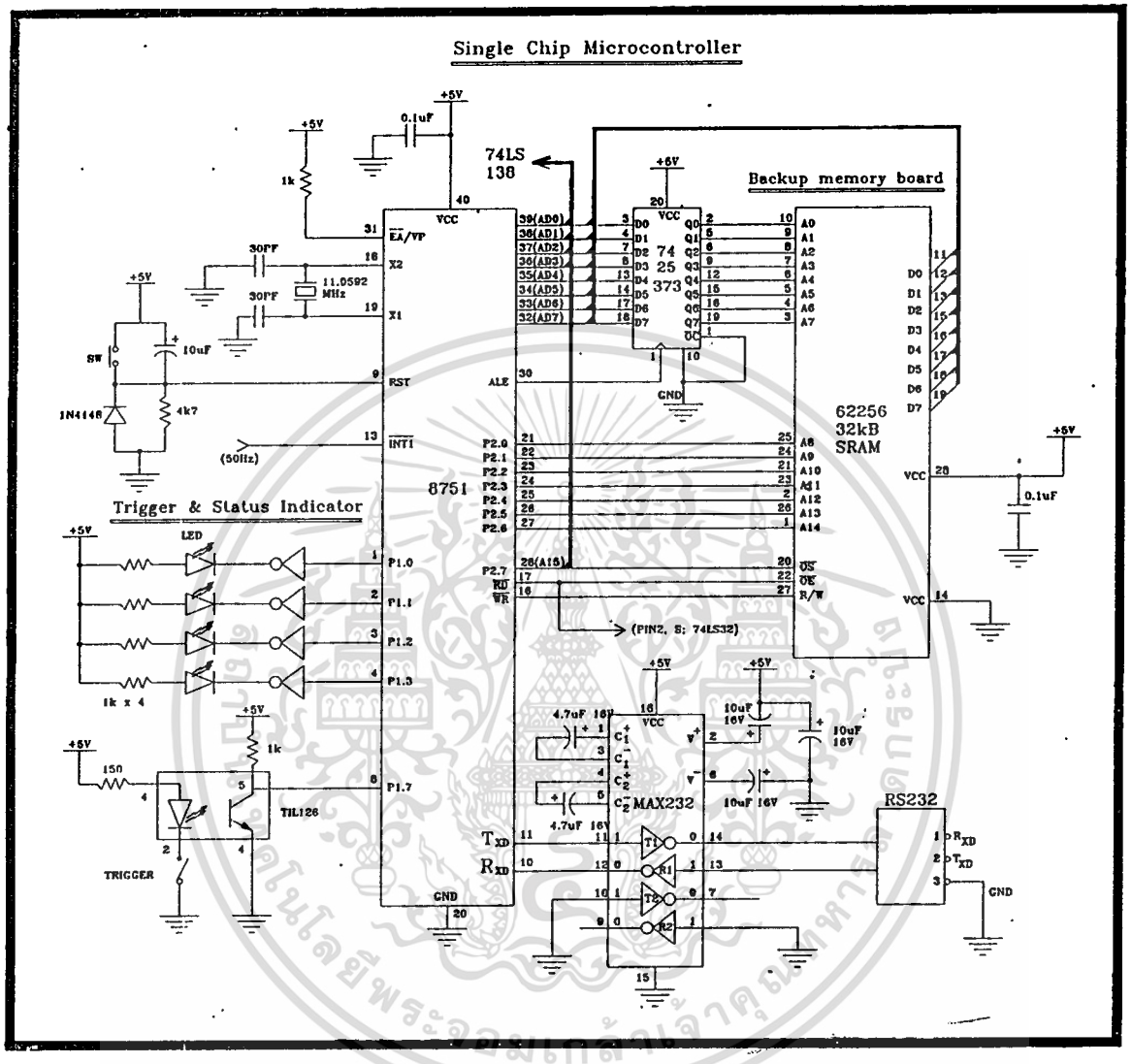


รูปที่ 4.3 แสดงวงจรนับและคงสถานะข้อมูล

ความถี่ที่ได้จากภาคเปลี่ยนแรงดันเป็นความถี่ จะส่งเข้ามาที่ภาควงจรนับและคงสถานะข้อมูล ซึ่งจะส่งผ่านสัญญาณโดยใช้แสง เพื่อแยกส่วนระหว่างวงจรอนาลอกและวงจรดิจิตอลออกจากกัน

ภาควงจรนับและคงสถานะข้อมูลประกอบไปด้วยวงจรนับขนาด 8 บิท 2 ชุด รวมเป็น 16 บิท และมีวงจรคงสถานะข้อมูล(latch) ขนาด 8 บิท 2 ชุด เพื่อใช้คงค่าข้อมูลที่นับได้ขนาด 16 บิท (รูปที่ 4.3) มีสัญญาณนาฬิกาความถี่ 50 เฮิร์ทซ์ เป็นสัญญาณคงค่าข้อมูลและเป็นสัญญาณอินเตอรัปต์ที่จะให้ไมโครคอนโทรลเลอร์มาอ่านข้อมูลที่คงค่าไว้นี้ ไปประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

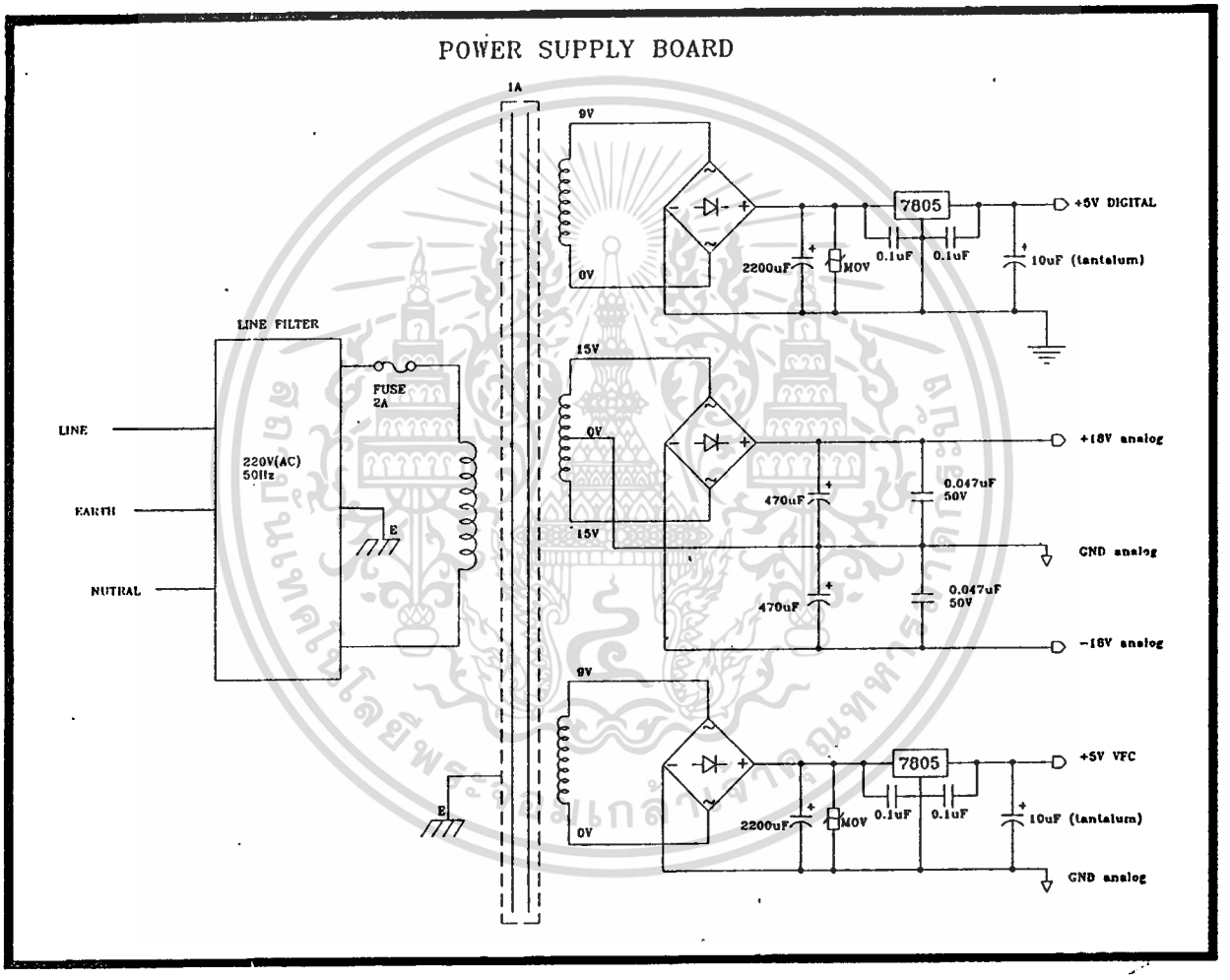


รูปที่ 4.4 แสดงภาคประมวลผลและรับส่งข้อมูล

ภาคประมวลผลข้อมูล (รูปที่ 4.4) จะประกอบไปด้วยไมโครคอนโทรลเลอร์เบอร์ 8751 และหน่วยความจำขนาด 32 กิโลไบท์ ทำหน้าที่อ่านค่าขนาดของสัญญาณที่นับได้จาก ภาควงจรนับและคงสถานะข้อมูลมาประมวลผลและเก็บไว้ในหน่วยความจำ และจะแปลงค่าที่ได้นี้เป็นรหัสแอสกี ซึ่งจะมีภาครับส่งข้อมูลเป็นตัวส่งข้อมูลให้กับไมโครคอมพิวเตอร์ต่อไป

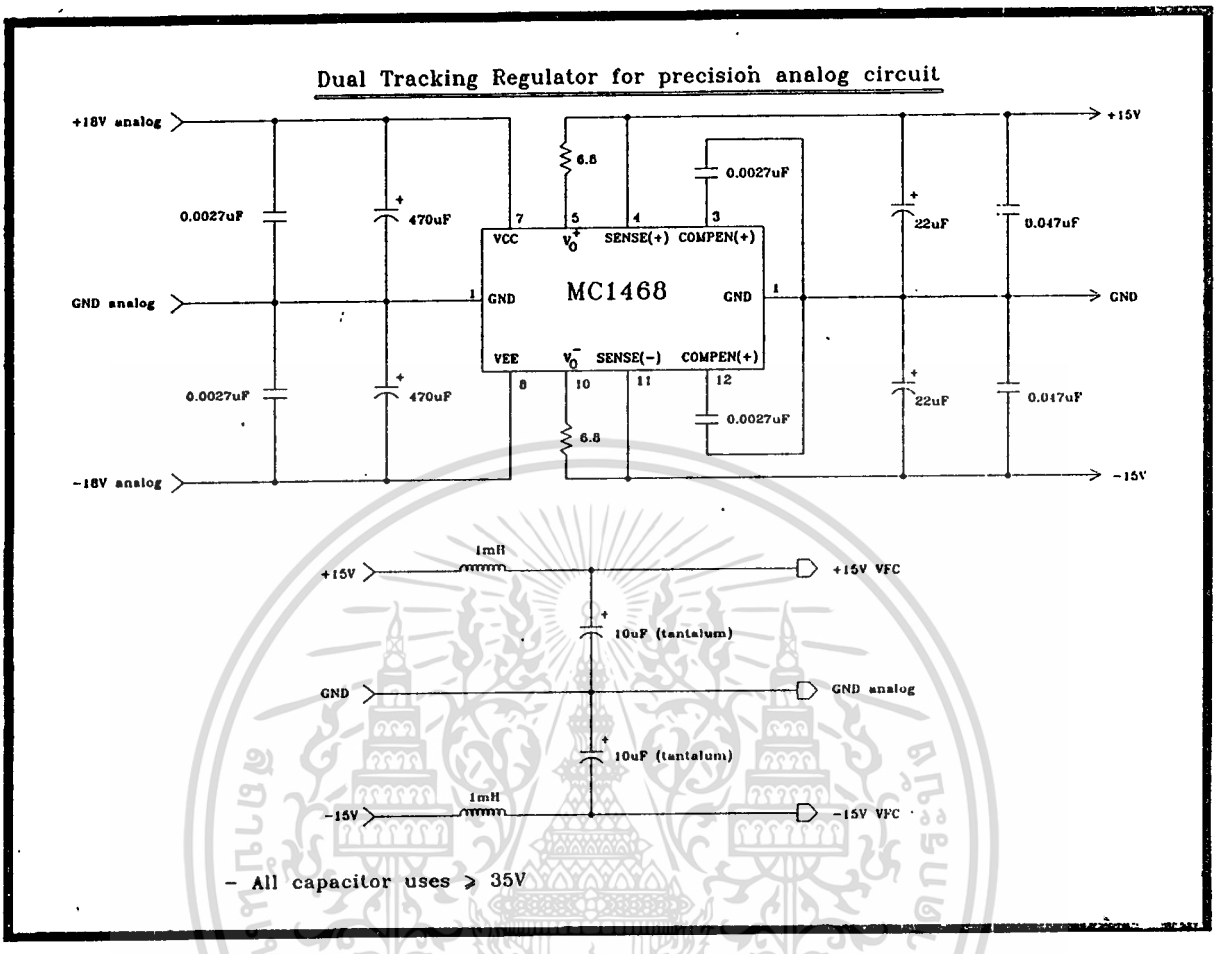
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับภาคจ่ายไฟเลี้ยงดังรูปที่ 4.5 จะแบ่งเป็น 3 ส่วน คือไฟเลี้ยงขนาด +5 Vdc สำหรับวงจรดิจิตอล ไฟเลี้ยงขนาด +5 Vdc และขนาด $\pm$ 18 Vdc สำหรับวงจรอนาลอกซึ่งไฟเลี้ยงขนาด $\pm$ 18 Vdc นี้ จะถูกแปลงเป็น $\pm$ 15 Vdc เพื่อเลี้ยงวงจรอนาลอก ซึ่งต้องการความเที่ยงตรงสูง โดยใช้วงจร Dual Tracking Regulator ดังรูปที่ 4.6 เพื่อให้การทำงานของภาคแปลงสัญญาณมีความเที่ยงตรงและถูกต้องที่สุด



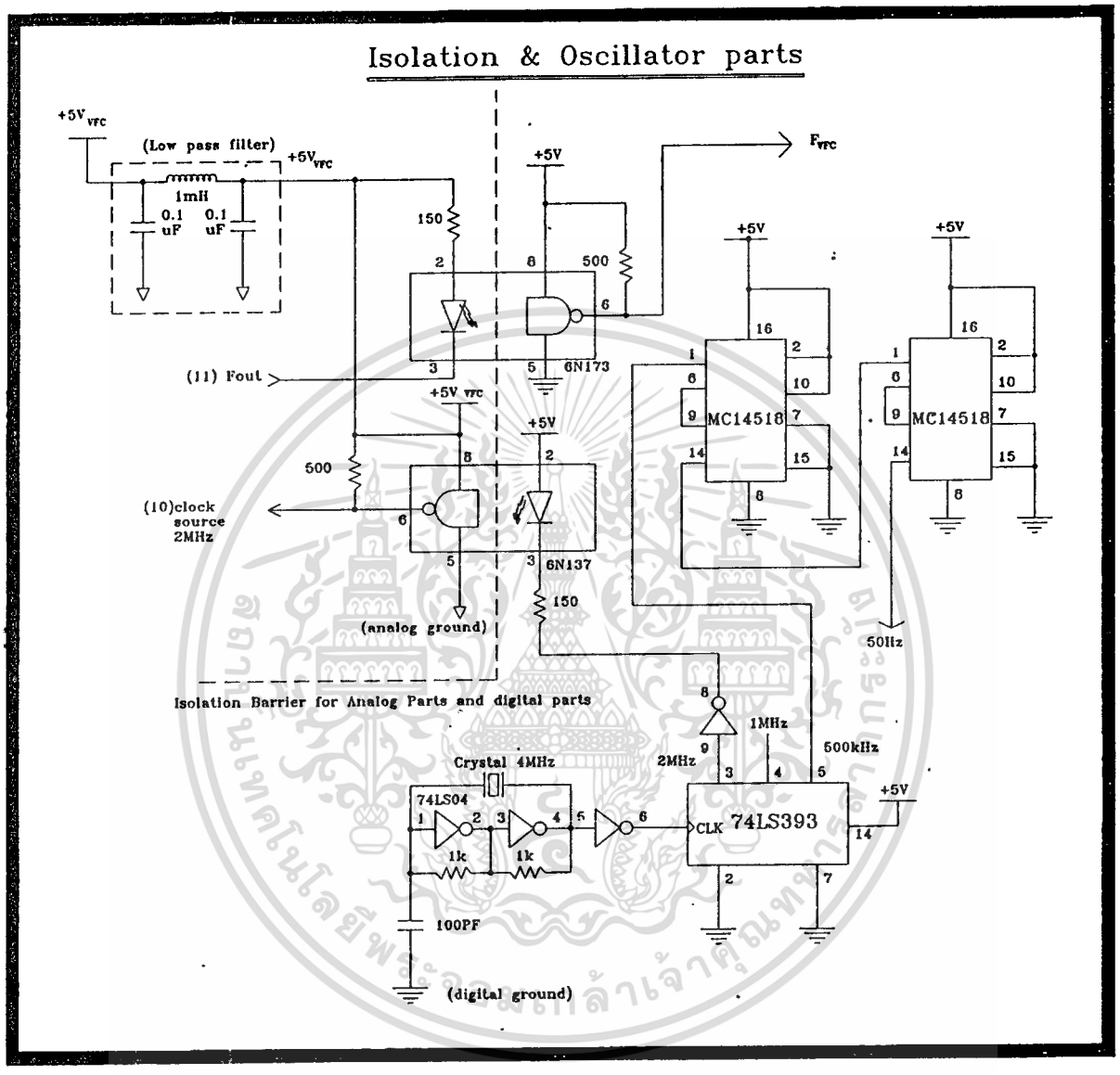
รูปที่ 4.5 แสดงวงจรของภาคจ่ายไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงวงจร DUAL TRACKING REGULATOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงภาคสร้างสัญญาณนาฬิกาขนาด 2 MHz และ 50 Hz

ภาคสร้างสัญญาณนาฬิกาในรูปที่ 4.7 จะสร้างสัญญาณนาฬิกาขนาด 2 เมกะเฮิร์ทซ์ และ 50 เฮิร์ทซ์ จาก X-TAL 4 เมกะเฮิร์ทซ์ โดยที่สัญญาณนาฬิกาความถี่ 2 เมกะเฮิร์ทซ์ จะใช้กับวงจรแปลงแรงดันเป็นความถี่ ส่วนสัญญาณนาฬิกาความถี่ 50 เฮิร์ทซ์ จะใช้กับการคงค่าข้อมูลของค่าขนาดสัญญาณที่นับได้ และใช้เป็นสัญญาณอินเตอรัปต์ของไมโครคอนโทรลเลอร์ เพื่อใช้ในการอ่านค่าขนาดของสัญญาณที่นับไว้ นอกจากนี้จะมีสัญญาณนาฬิกาความถี่ 11.0592 เมกะเฮิร์ทซ์ ซึ่งจะอยู่ในส่วนของไมโครคอนโทรลเลอร์ เพื่อใช้เป็นฐานเวลาสำหรับการทำงานของวงจรในภาคประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบซอฟต์แวร์วิเคราะห์ผลข้อมูล

ซอฟต์แวร์วิเคราะห์ผลข้อมูลสำหรับ HPLC เป็นซอฟต์แวร์ที่พัฒนาขึ้นมาโดยใช้ ภาษาปาสคาล ซึ่งใช้เทอร์โบปาสคาล เวอร์ชัน 5.0 เป็นตัวแปลภาษา(Compiler) เนื่องจากภาษาปาสคาลเป็นภาษาระดับสูงภาษาหนึ่ง ซึ่งมีลักษณะการเขียนโปรแกรมเป็นโครงสร้าง สามารถเขียนอ่านได้ง่าย และใกล้เคียงภาษามนุษย์ จึงเหมาะที่จะนำไปใช้ในการพัฒนาโปรแกรมต้นแบบ ซึ่งจะช่วยให้การนำซอฟต์แวร์ต้นแบบนี้ไปแก้ไขหรือพัฒนาต่อไปได้ง่ายและรวดเร็วขึ้น

การทำงานของซอฟต์แวร์ตัวนี้ จะเป็นการนำเอาข้อมูลที่ได้จากเครื่อง HPLC มาประมวลผลและวิเคราะห์เพื่อหาพีคสัญญาณ พื้นที่พีคสัญญาณ ชนิดของสารที่พบ และความเข้มข้นของสารที่พบ โดยสามารถแสดงผลของกราฟสัญญาณแบบทันที รวมทั้งการรายงานผลแบบทันทีด้วยเช่นกัน สามารถที่จะกำหนดรูปแบบ เงื่อนไขและพารามิเตอร์ต่างๆ ที่ใช้ในการวิเคราะห์ได้ นอกจากนี้ยังสามารถเก็บข้อมูลที่ได้ลงบนแผ่นจานแม่เหล็ก เพื่อนำกลับไปวิเคราะห์ใหม่ได้

หลักการทำงานของซอฟต์แวร์ประมวลผลและวิเคราะห์ข้อมูลสามารถแบ่งออกเป็น 7 ขั้นตอน ดังนี้

4.3.1 การรับข้อมูลเข้ามาวิเคราะห์

4.3.2 การแปลงข้อมูลที่รับเข้ามาให้เป็นตัวเลขจำนวนเต็ม

4.3.3 การกรองสัญญาณด้วยวิธี Moving Average

4.3.4 การตรวจหาพีคสัญญาณและวิเคราะห์ข้อมูล

4.3.5 การแสดงผลสัญญาณทางจอภาพ

4.3.6 การรายงานผลการวิเคราะห์

4.3.7 การอ่านและบันทึกข้อมูลลงบนแผ่นจานแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 การรับข้อมูลเข้ามาวิเคราะห์

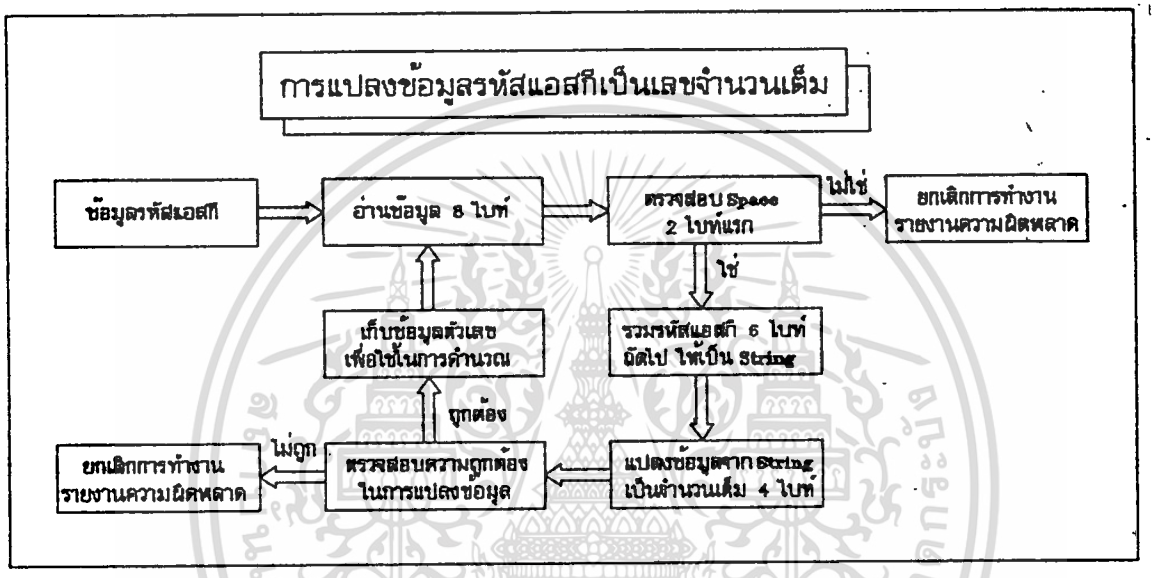
สามารถแบ่งรูปแบบการรับข้อมูลออกได้เป็น 2 รูปแบบ คือ การรับข้อมูลจากเครื่องเก็บและรับส่งข้อมูล และจากไฟล์ข้อมูลที่เก็บไว้ในแผ่นจานแม่เหล็ก ในกรณีของการรับข้อมูลจากเครื่องเก็บและรับส่งข้อมูลนั้น โปรแกรมจะรับข้อมูลซึ่งเป็นรหัสแอสกีผ่านทางพอร์ทแบบอนุกรม ข้อมูลรหัสแอสกีแต่ละชุดซึ่งมีขนาด 8 ไบท์ จะนำไปแปลงเป็นเลขแบบไบนารี 4 ไบท์ โดยในเทอร์โบปาสคาลจะเรียกข้อมูลไบนารีขนาด 4 ไบท์เป็นข้อมูลชนิด LongInt ซึ่งสามารถแทนค่าเป็นเลขจำนวนเต็มได้ตั้งแต่ -2147483648..2147483647 จึงสามารถเก็บค่าข้อมูลตัวเลขขนาด 6 หลักซึ่งเป็นค่าข้อมูลที่สูงสุดของค่าสัญญาณที่ใช้งาน เมื่อโปรแกรมแปลงข้อมูลจากรหัสแอสกีเป็นจำนวนเต็มชนิด LongInt แล้ว จึงจะสามารถนำข้อมูลไปประมวลผลในขั้นตอนต่อไปได้ ส่วนกรณีของการอ่านข้อมูลจากไฟล์ข้อมูลนั้นจะอ่านเป็นข้อมูลเลขจำนวนเต็มชนิด LongInt ซึ่งสามารถนำไปประมวลผลได้ทันทีโดยไม่ต้องผ่านขั้นตอนการแปลงข้อมูลอีก

การรับข้อมูลของซอฟต์แวร์จะแบ่งออกเป็น 2 ลักษณะ คือการรับข้อมูลใน Labnet Mode ซึ่งจะเป็นการรับข้อมูลมาจากเครื่องเก็บและรับส่งข้อมูลโดยผ่านทางสาย RS232C ในรูปของรหัสแอสกี และการรับข้อมูลใน File Mode จะเป็นการอ่านไฟล์ข้อมูลที่เก็บไว้จากแผ่นจานแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การแปลงข้อมูลจากรหัสแอสกีเป็นตัวเลขจำนวนเต็ม

ในกรณีที่โปรแกรมรับข้อมูลตัวเลขเข้ามาในรูปของรหัสแอสกี จะไม่สามารถนำไปใช้ในการคำนวณได้ทันที จึงต้องมีการแปลงข้อมูลจากรหัสแอสกีไปเป็นเลขจำนวนเต็มที่โปรแกรมจะสามารถนำไปใช้ในการประมวลผลได้ ซึ่งในส่วนนี้จะเป็นการแปลงข้อมูลจากรหัสแอสกีขนาด 8 ไบท์เป็นตัวเลขจำนวนเต็มชนิด LongInt ขนาด 4 ไบท์เพื่อนำไปใช้ในการประมวลผลในขั้นต่อไป



รูปที่ 4.8 แสดงการแปลงข้อมูลจากรหัสแอสกีเป็นตัวเลขจำนวนเต็มชนิด LongInt

จากรูปที่ 4.8 โปรแกรมจะอ่านข้อมูลรหัสแอสกีเข้ามาทีละ 8 ไบท์ ซึ่งจะประกอบไปด้วยตัวอักษรที่เป็นอักษรว่าง(Space) 2 ตัว และตัวอักษรที่เป็นค่าตัวเลข 0 - 9 ดังตัวอย่าง

__023100 หรือ #20#20#30#32#33#31#30#30 (_ ใช้แทน Space)

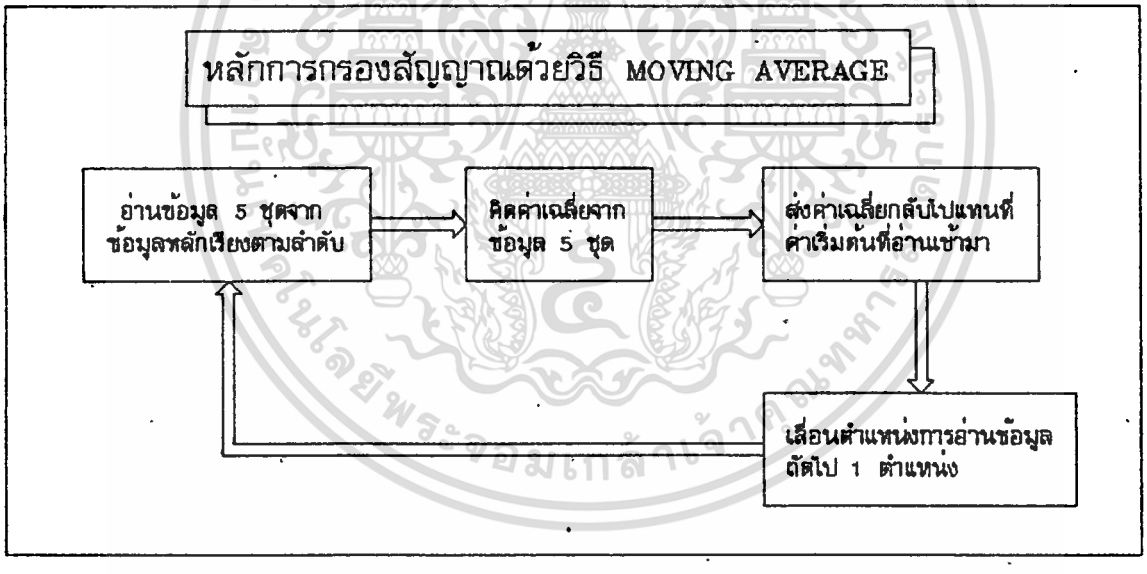
การตรวจสอบรหัสแอสกีจะเริ่มจากการตรวจรหัสแอสกี(HEX CODE)ของอักษรว่าง(#20)สองไบท์แรกก่อน เพื่อทราบว่าข้อมูลที่ได้รับเป็นข้อมูลที่ถูกต้อง จากนั้นจึงนำเอารหัสแอสกีที่เหลือ 6 ตัวถัดไปมารวมกันเป็นสตริง(String) แล้วใช้คำสั่งแปลงข้อมูลสตริงเป็นเลขชนิด LongInt ถ้าเกิดความผิดพลาดในการแปลงข้อมูล เช่น มีอักษรตัวอื่นที่ไม่ใช่ตัวเลขปะปนเข้ามา โปรแกรมจะยกเลิกการทำงานและรายงานความผิดพลาดให้ทราบ

เลขจำนวนเต็มที่แปลงได้จะมีการเก็บข้อมูลเป็นแบบอนุกรมชนิด Link List ทางเดียว ซึ่งเป็นการเก็บข้อมูลโดยใช้ตัวชี้(Pointer)แบบหนึ่งของเทอร์โบปาสคาล เพื่อนำไปใช้ในการคำนวณในขั้นถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

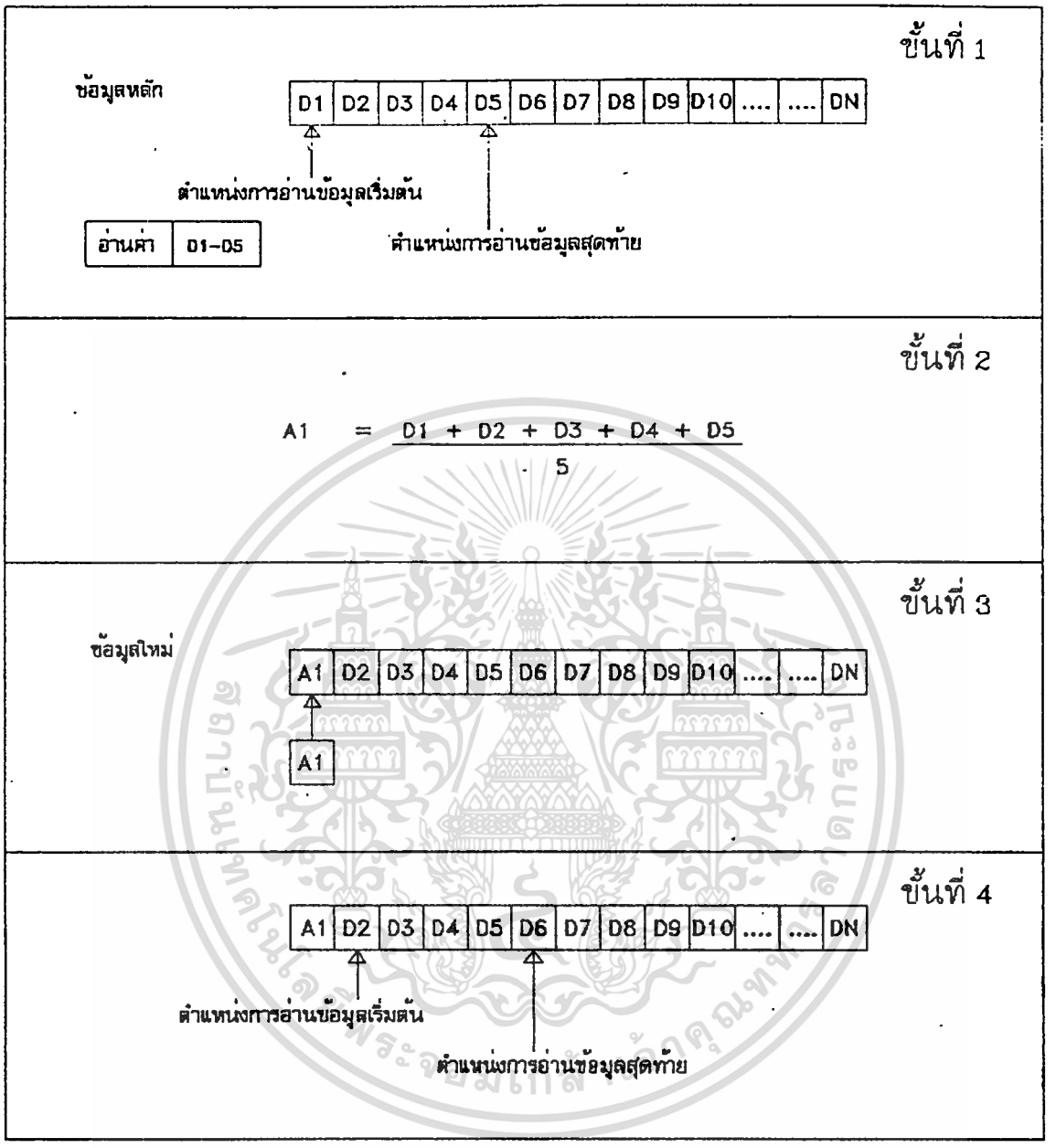
4.3.3 การกรองสัญญาณด้วยวิธี Moving average

โดยปกติสัญญาณที่ได้รับจากเครื่อง HPLC อาจจะมีสัญญาณรบกวนขนาดเล็กๆ ปะปนอยู่ ซึ่งในบางครั้งอาจจะส่งผลให้มีการตรวจพบเป็นพีคสัญญาณที่ไม่ถูกต้องขึ้นมาได้ กรณีที่มีสัญญาณรบกวนในลักษณะเช่นนี้ จึงควรกรองสัญญาณรบกวนที่ปะปนเข้ามาออกโดยอาศัยวิธี Moving average ซึ่งวิธีนี้จะช่วยให้สัญญาณที่ได้มีความเรียบมากขึ้น ทำให้ลดผลกระทบจากสัญญาณรบกวนขนาดเล็กๆได้มาก การตรวจเช็คพีคสัญญาณที่ได้ก็จะมีความถูกต้องมากขึ้น ซึ่งในซอฟต์แวร์ที่พัฒนาขึ้นมานี้สามารถกำหนดให้มีการกรองสัญญาณหรือไม่กรองสัญญาณก็ได้ โดยการกรองสัญญาณที่ใช้จะเป็น Moving average แบบเฉลี่ย 5 จุด



รูปที่ 4.9 แสดงการคำนวณโดยใช้ Moving average แบบเฉลี่ย 5 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



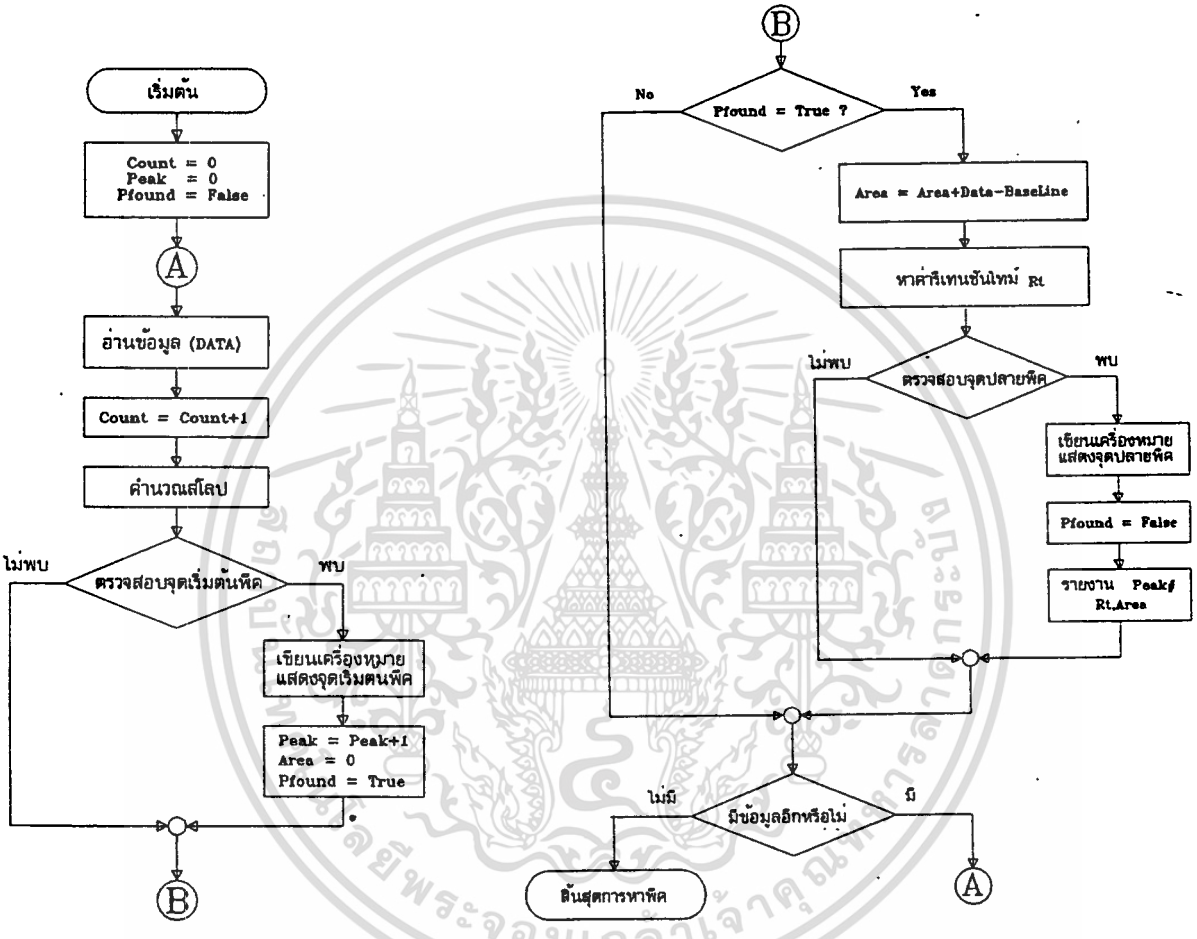
รูปที่ 4.10 แสดงขั้นตอนการคำนวณโดยวิธี Moving average แบบเฉลี่ย 5 จุด

เมื่อดำเนินการไปจนหมดข้อมูล D_n ค่า จะได้ข้อมูลเฉลี่ยทั้งสิ้น D_{n-4} ค่า ซึ่งในกรณีเช่นนี้จะทำให้ค่าเฉลี่ยที่ได้เลื่อนไปทางซ้ายเล็กน้อย ดังนั้นค่ารีเทนชันไทม์ของพีคที่พบหลังจากใช้วิธีเฉลี่ยแบบ Moving average นี้ จะมีค่าน้อยกว่าค่าจริงเล็กน้อย เช่น ในกรณีของการคำนวณจากข้อมูลซึ่งเป็นข้อมูลที่ได้ในแต่ละช่วงเวลา 0.1 วินาที หรือ ได้ข้อมูล 10 ค่าใน 1 วินาที จะมีความคลาดเคลื่อนประมาณ 0.25 วินาที(คิดเวลาจากครึ่งหนึ่งของจำนวนข้อมูลที่ใช้เฉลี่ย) ซึ่งสามารถชดเชยได้ในภายหลัง แต่ในทางปฏิบัติแล้วถือว่าเป็นค่าความคลาดเคลื่อนที่น้อยมากจนไม่มีผลต่อการตรวจสอบพีคแต่อย่างใด เนื่องจากใช้การตรวจสอบชนิดของพีคแบบเปิดช่วงเวลา

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การตรวจหาพีคสัญญาณและวิเคราะห์ข้อมูล

ในการตรวจหาพีคสัญญาณและวิเคราะห์ข้อมูล นับได้ว่าเป็นหัวใจสำคัญของโปรแกรมที่จะตรวจพีคสัญญาณต่างๆ ที่พบ และรายงานผลการวิเคราะห์ที่ได้ ซึ่งสามารถเขียนเป็นแผนภาพแสดงการทำงานได้ดังนี้



รูปที่ 4.11 แสดงหลักการตรวจหาพีคสัญญาณและวิเคราะห์ข้อมูล

สำหรับแผนภาพที่นำมาแสดงนี้จะเป็นหลักการทำงานของโปรแกรมตามปกติ ซึ่งมีค่าพีควิดท์ (Peak Width) เป็น 1 วินาที ในช่วงแรกของโปรแกรมจะเป็นการกำหนดค่าเริ่มต้นต่างๆ เช่น Count = 0, Peak = 0 และ Pfound = False โดยที่ Count คือ จำนวนข้อมูลที่ส่งเข้ามา Peak คือ จำนวนพีคที่พบ และ Pfound คือ สถานะของการตรวจพบพีค ซึ่งจะใช้ในการคำนวณหาพื้นที่พีค จุดยอดพีค และจุดปลายพีค การคำนวณค่าความชันของสัญญาณจะใช้วิธีการคำนวณแบบเฉลี่ยน้ำหนักหลายจุดซึ่งได้กล่าวไว้เป็นสมการดังรูปที่ 3.3 ในบทที่ 3 สำหรับการตรวจหาจุดเริ่มต้น การหาจุดยอดพีค และจุดปลายพีคมีหลักการตรวจหาดังจะกล่าวต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

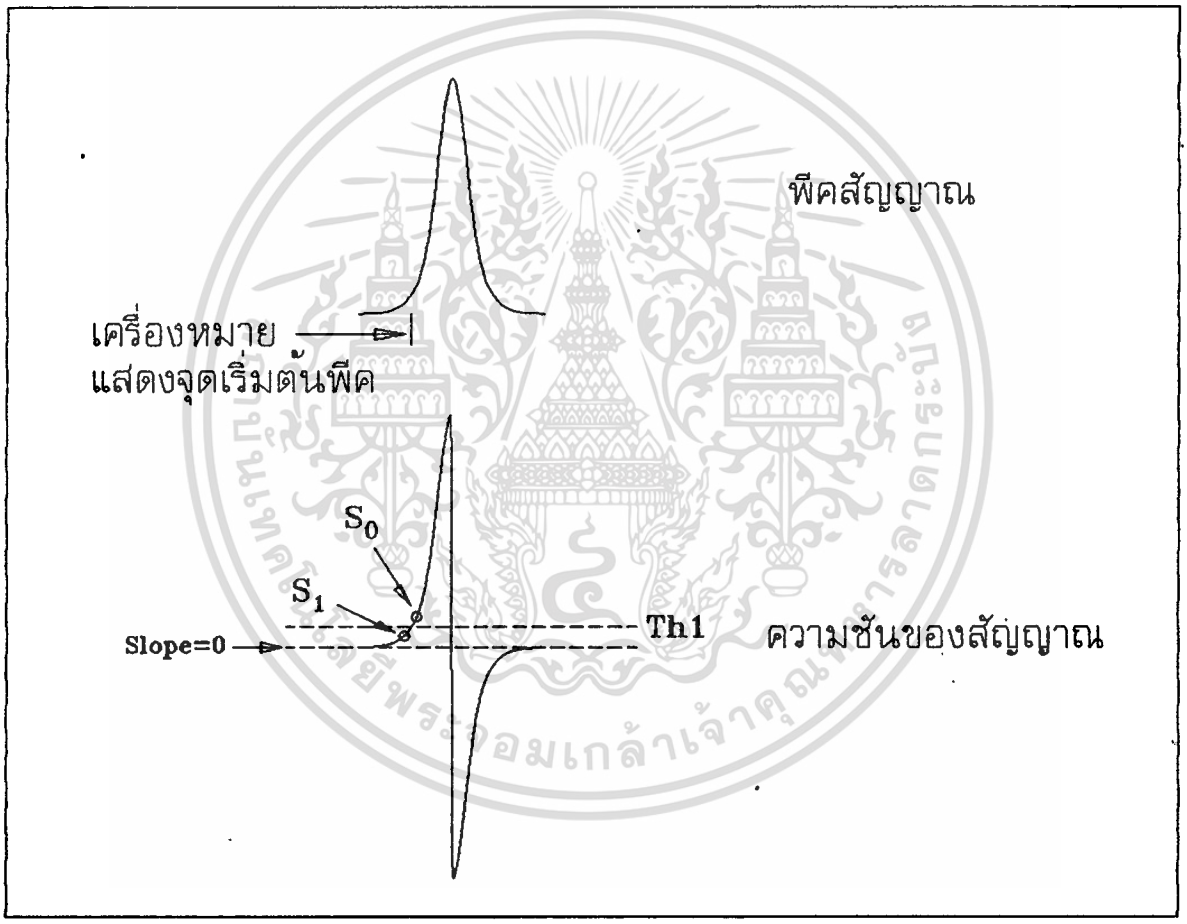
การหาจุดเริ่มต้นพีค

จะใช้การอ่านค่าความชันที่คำนวณได้ มาเปรียบเทียบกับค่าเทรสโฮลที่กำหนด โดยมีเงื่อนไขการตรวจพบพีคดังนี้

S_0 >= Th1 และ S_1 < Th1 เมื่อ S_0 คือ ความชันค่าแรก ที่มากกว่าหรือเท่ากับระดับ Th1

S_1 คือ ความชันค่าถัดมา ซึ่งมีค่าน้อยกว่าระดับ Th1

และจุด S_0 คือจุดเริ่มต้นของพีค



รูปที่ 4.12 แสดงหลักการตรวจหาจุดเริ่มต้นพีค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

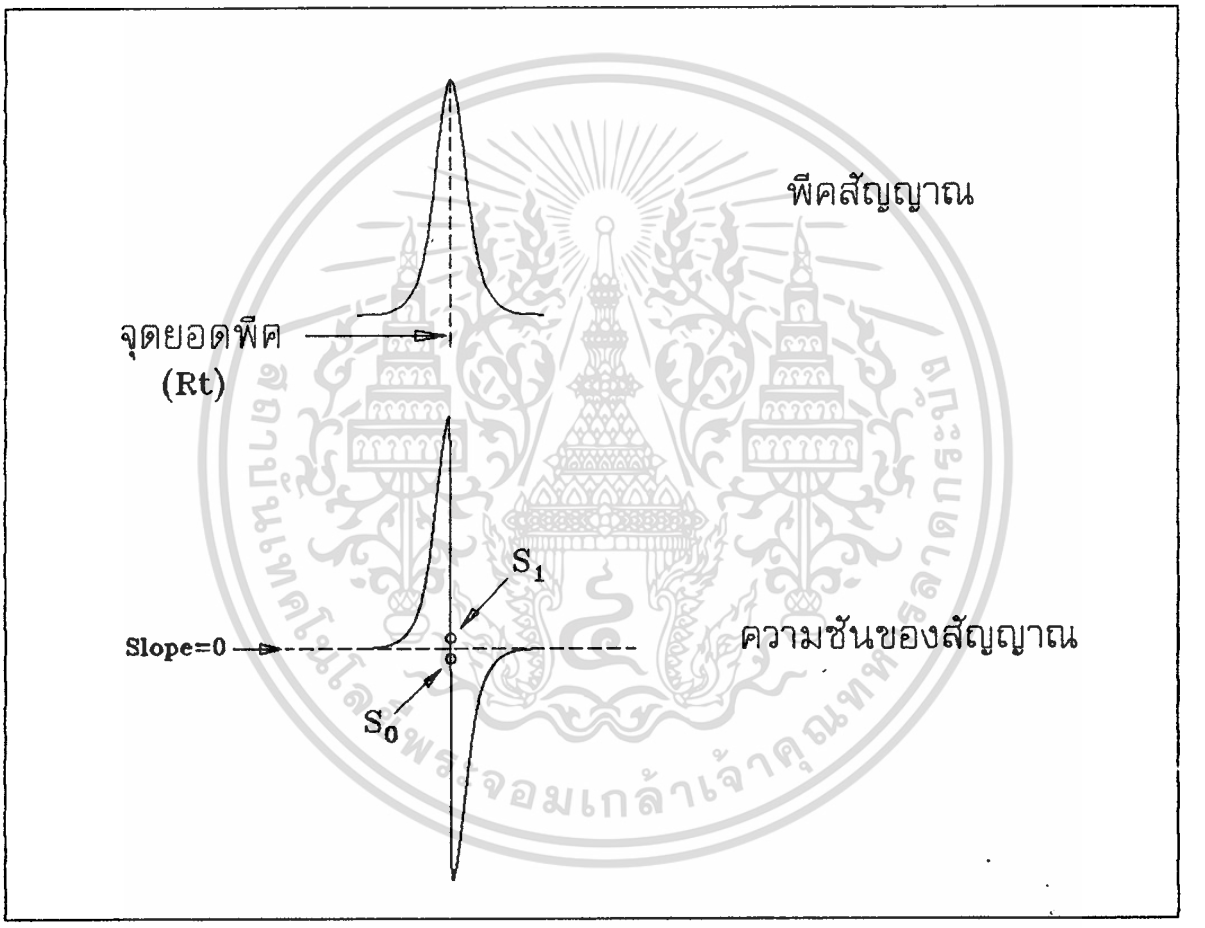
การหาจุดยอดพีค(Rt)

มีเงื่อนไขในการตรวจหาดังนี้

$S_0 <= 0$ และ $S_1 > 0$ เมื่อ S_0 คือ ความชันค่าแรก ที่น้อยกว่าหรือเท่ากับศูนย์

S_1 คือ ความชันค่าถัดมา ซึ่งมีค่ามากกว่าศูนย์

และจุด S_0 คือจุดยอดของพีค



รูปที่ 4.13 แสดงหลักการตรวจหาจุดยอดพีค(Rt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

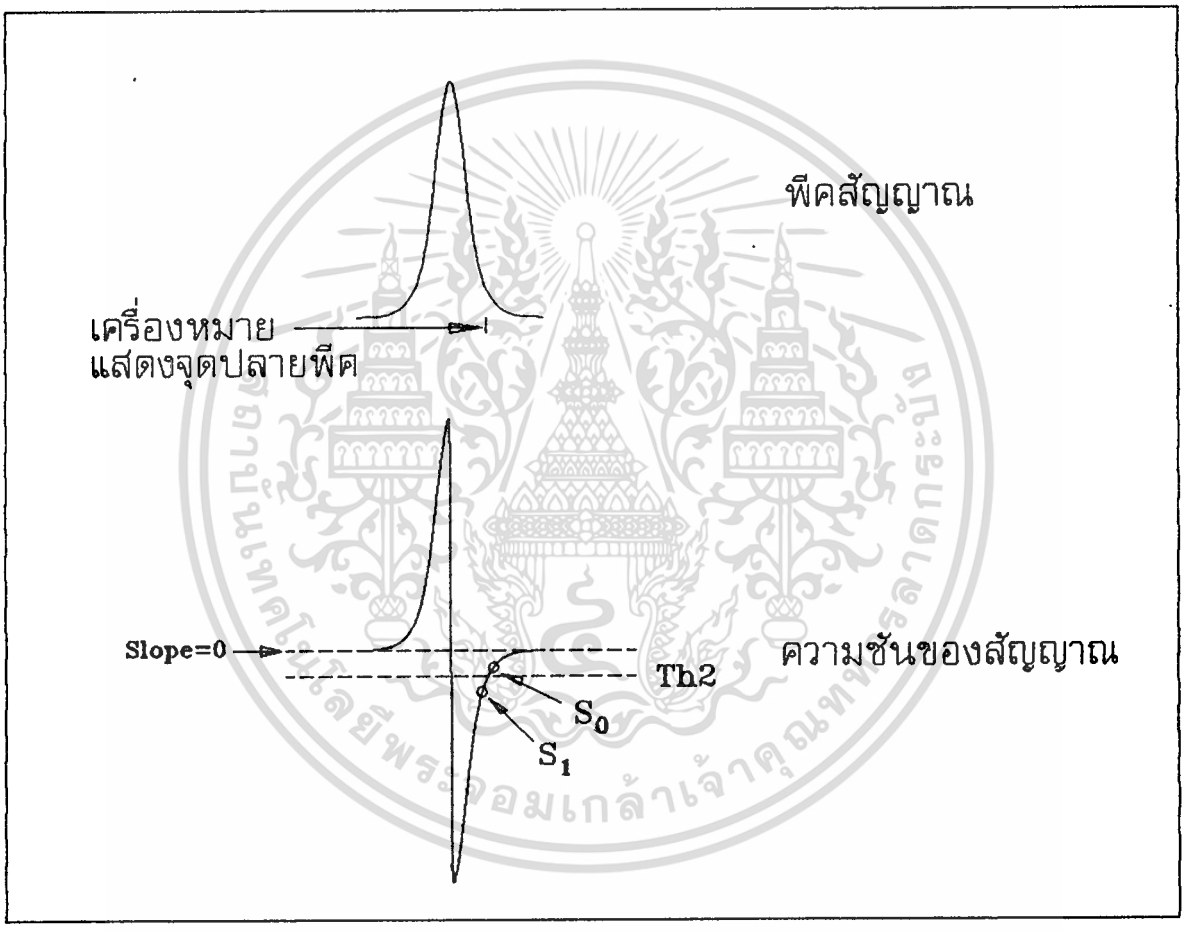
การหาจุดปลายพีค

มีเงื่อนไขในการตรวจหาดังนี้

S_0 >= Th2 และ S_1 < Th2 เมื่อ S_0 คือ ความชันค่าแรก ที่มากกว่าหรือเท่ากับระดับ Th2

S_1 คือ ความชันค่าถัดมา ซึ่งมีค่าน้อยกว่าระดับ Th2

และจุด S_0 คือจุดปลายของพีค



รูปที่ 4.14 แสดงหลักการตรวจหาจุดปลายพีค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณความชันนั้นสามารถใช้สมการในรูปที่ 3.3 ในบทที่ 3 สำหรับการคำนวนพื้นที่จะใช้วิธีคำนวณเป็นผลบวกโดยรวมของข้อมูลที่อยู่เหนือระดับฐานพีค(Baseline) ตั้งแต่จุดเริ่มต้นพีคจนถึงจุดปลายพีค ซึ่งสามารถคำนวณได้ดังนี้

$$Area = \sum(Data - Baseline)$$

ในการคำนวนที่มีค่า Peak Width มากกว่าหนึ่ง ทำได้โดยการบวกข้อมูลย่อยเข้าด้วยกัน เช่นกรณี Peak Width = 2 จะเกิดจากการนำข้อมูลสองค่ามารวมกันเป็นข้อมูลใหม่ค่าเดียว ซึ่งนำมาใช้ในการคำนวนรูปแบบเดียวกันกับกรณี Peak Width = 1 เพียงแต่ ต้องกำหนดรูปแบบการคำนวณให้สอดคล้องกับการเปลี่ยนแปลงขนาดของ Peak Width ด้วย เช่นในการหาค่าพื้นที่จะได้ว่า

$$Area = \sum(NData - Baseline * (Peak\ Width))$$

เมื่อ NData คือ ข้อมูลที่เกิดจากการรวมข้อมูลย่อย(Data) เข้าด้วยกันตามค่าของพีควิดท์

ค่ารีเทนชันไทม์(Rt) = NCount * (Peak Width) / 600 มีหน่วยเป็นนาที

เมื่อค่า NCount คือจำนวนข้อมูลของ NData

ค่าระดับตรวจจับ(Threshold) จะเพิ่มขึ้นด้วยเพื่อให้สอดคล้องกับการรวมข้อมูลที่เพิ่มขึ้นโดยจะเป็นสัดส่วนกับค่าพีควิดท์คือ ค่าระดับตรวจจับใหม่ จะเท่ากับ ค่าระดับตรวจจับเดิมคูณกับค่าพีควิดท์

ข้อดีสำหรับการใช้พีควิดท์ค่ามากๆ คือ ทำให้สามารถตรวจหาค่ามีความชันน้อยได้ดีขึ้น แต่จะมีข้อเสียด้านความถูกต้องในการคำนวณค่ารีเทนชันไทม์ซึ่งจะผิดพลาดได้มากขึ้นเป็นสัดส่วนกับค่าพีควิดท์

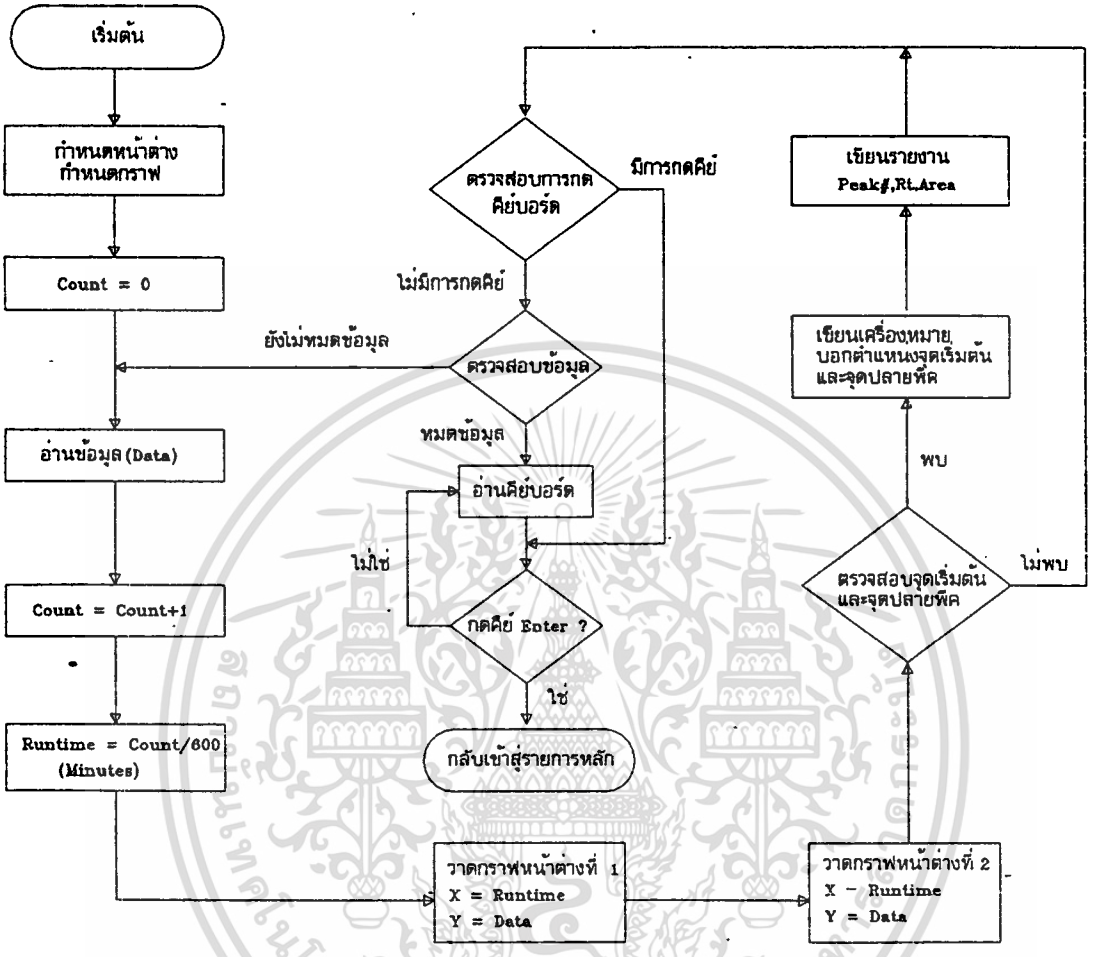
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 การแสดงผลสัญญาณทางจอภาพ

ในการแสดงผลสัญญาณทางจอภาพจะแบ่งการแสดงกราฟสัญญาณเป็น 2 หน้าต่าง โดยหน้าต่างแรกจะแสดงกราฟสัญญาณทั้งหมด และอีกหน้าต่างจะแสดงกราฟสัญญาณบางส่วนตามช่วงเวลาที่กำหนด การทำงานจะเริ่มจากการกำหนดค่าต่างๆ ให้กับหน้าต่างแสดงกราฟสัญญาณ พร้อมทั้งแสดงหน้าต่างบนจอภาพ และกำหนดหน่วยนับ(Count) ให้เป็นศูนย์ ซึ่งหน่วยนับนี้ จะใช้ในการกำหนดเวลาของข้อมูลแต่ละค่า โดยช่วงเวลาสำหรับแต่ละข้อมูลจะห่างกัน 0.1 วินาที และจะเริ่มอ่านข้อมูลเข้ามาทีละค่า เพิ่มค่าให้กับ Count ทีละหนึ่ง และคำนวณค่าเวลาทำงาน(Runtime) ในขณะนั้น เมื่อค่าเวลาการทำงาน คำนวณได้จาก Runtime = Count/600 มีหน่วยเป็นนาที จากนั้นจะวาดกราฟสัญญาณบนหน้าต่างทั้งสองหน้าต่างตามลำดับ โดยใช้แกน Y เป็นขนาดของสัญญาณ และแกน X เป็นเวลา(นาที) จากนั้นจะตรวจสอบพีคของสัญญาณที่ได้รับ ถ้าพบพีคก็จะเขียนเส้นแสดงตำแหน่ง(Peak Marker) ของจุดเริ่มต้นและจุดปลายของพีค รวมทั้งแสดงถึงจำนวนพีคที่พบ ค่ารีเทนชันไทม์ของพีคและพื้นที่พีค แต่ถ้าไม่พบพีคก็จะผ่านไป โปรแกรมจะตรวจสอบข้อมูลว่าหมดหรือยังจากค่า RunTime ที่กำหนดไว้ หรือจากจำนวนข้อมูลสูงสุดที่มี และตรวจสอบการกดคีย์ Enter ในกรณีที่ยังไม่หมดข้อมูลและยังไม่มีการกดคีย์ Enter โปรแกรมจะอ่านข้อมูลและแสดงผลต่อไปเรื่อยๆ จนกระทั่งหมดข้อมูล เมื่อมีการกดคีย์ Enter โปรแกรมจะออกไปสู่รายการหลัก

สำหรับในการวาดกราฟสัญญาณในหน้าต่างที่ 2 จะมีการเปลี่ยนช่วงเวลาการแสดงกราฟใหม่เมื่อกราฟสัญญาณเกินค่าเวลาที่กำหนดไว้ ซึ่งช่วงเวลานี้สามารถกำหนดได้จากค่า Page Range ในรายการ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



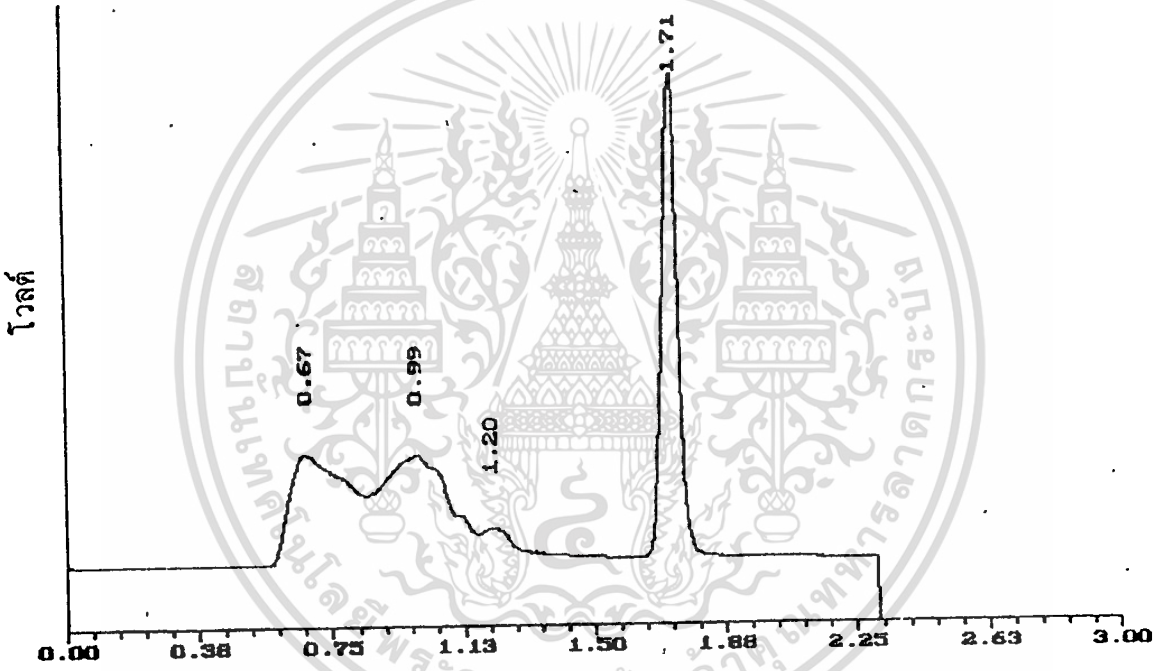
รูปที่ 4.15 หลักการแสดงกราฟสัญญาณบนจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 การรายงานผลการวิเคราะห์

เราสามารถแสดงผลการวิเคราะห์ทางเครื่องพิมพ์ได้ โดยแสดงผลการวิเคราะห์ที่ประกอบไปด้วยกราฟของสัญญาณและข้อมูลของพีค ซึ่งจะมีหมายเลขพีค ค่ารีเทนชันไทม์ และพื้นที่พีค ในกรณีที่ใช้ External Standard Method จะรายงานเป็น ชื่อพีค ค่ารีเทนชันไทม์ และค่าความเข้มข้น

สำหรับการพิมพ์กราฟสัญญาณนั้น จะใช้วิธีพิมพ์จากจอภาพ โดยจะเขียนกราฟสัญญาณและแสดงค่ารีเทนชันไทม์ของพีคบนจอภาพก่อน แล้วจึงสั่งพิมพ์รายงานทั้งหมดออกทางเครื่องพิมพ์ ซึ่งสามารถใช้ได้กับเครื่องพิมพ์ดอตเมตริกซ์ชนิด 8 เข็ม และ 24 เข็ม



รูปที่ 4.16 แสดงลักษณะของกราฟสัญญาณที่พิมพ์ออกมาทางเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 การอ่านและบันทึกข้อมูลลงบนแผ่นจานแม่เหล็ก

การทำงานของโปรแกรม จะสามารถบันทึกไฟล์ข้อมูลลงบนแผ่นจานแม่เหล็กได้เมื่อมีข้อมูลและอยู่ใน Labnet Mode เท่านั้น ในขณะที่ในการอ่านไฟล์ข้อมูลเพื่อวิเคราะห์ข้อมูลใหม่จะต้องทำงานอยู่ใน File Mode การอ่านไฟล์ข้อมูลจากแผ่นจานแม่เหล็กจะเป็นการอ่านข้อมูลที่เป็นตัวเลขจำนวนเต็มชนิด LongInt ซึ่งมีขนาด 4 ไบท์ การเก็บข้อมูลแบบนี้จะมีข้อดีกว่าการเก็บข้อมูลเป็นรหัสแอสกี 2 ประการ คือ จะมีขนาดข้อมูลเล็กกว่าการเก็บแบบรหัสแอสกี 2 เท่า และไม่ต้องแปลงข้อมูลจากรหัสแอสกีเป็นเลขจำนวนเต็มทำให้ใช้เวลาในการทำงานน้อยกว่า

การบันทึกข้อมูลจะบันทึกข้อมูลเป็นนาที(600 ข้อมูล/นาที) กรณีที่ข้อมูลไม่ครบนาทีเช่น 8.5 นาทีโปรแกรมจะเติมข้อมูลค่า 0 ให้จนครบ 9.0 นาที แล้วจึงบันทึกข้อมูลขนาด 9.0 นาทีลงบนแผ่นจานแม่เหล็ก สำหรับการอ่านข้อมูลจะอ่านข้อมูลเลขจำนวนเต็มจากไฟล์ข้อมูล โปรแกรมจะกำหนดเวลาสูงสุดในการแสดงผลจากข้อมูลที่อ่านเข้ามาโดยอัตโนมัติ

4.4 การเชื่อมต่อระหว่างชุดเก็บและรับส่งข้อมูลกับซอฟต์แวร์วิเคราะห์ผลข้อมูล

ในการสื่อสารข้อมูลระหว่างชุดเก็บและรับส่งข้อมูลกับไมโครคอมพิวเตอร์ จะใช้การสื่อสารข้อมูลแบบอนุกรม โดยใช้สาย RS232C โดยใช้ข้อมูลเป็นรหัสแอสกี

คำสั่งที่ใช้ในการติดต่อกับชุดเก็บและรับส่งข้อมูล(DAS) จะมีอยู่ 5 คำสั่งด้วยกัน คือ

4.4.1 Link Active เป็นการส่งรหัสแอสกี #20 ให้ DAS อยู่ในสภาวะเตรียมพร้อมที่จะรับการติดต่อ

4.4.2 Trigger Active เป็นการส่งรหัสแอสกี #53 ให้ DAS รอสัญญาณทริกเกอร์ซึ่งจะเป็นสัญญาณจากการกดสวิทช์จากภายนอก เมื่อ DAS ได้รับสัญญาณทริกเกอร์ ก็จะเริ่มเก็บข้อมูลที่อ่านได้จากค่าสัญญาณในขณะนั้น แต่ถ้ารอนานเกินช่วงเวลาที่กำหนดโปรแกรมจะรายงานว่า " Time Out Error ! "

4.4.3 XON เป็นการส่งรหัสแอสกี #11 ไปให้ DAS เพื่อให้ส่งข้อมูลมาให้ไมโครคอมพิวเตอร์

4.4.4 XOFF เป็นการส่งรหัสแอสกี #13 ไปให้ DAS เพื่อให้หยุดการส่งข้อมูล

4.4.5 BREAK เป็นการส่งรหัสแอสกี #00 ไปให้ DAS เพื่อยกเลิกการรับส่งข้อมูลและกำหนดให้เครื่อง DAS กลับไปอยู่ในสถานะเริ่มต้นอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดต่อสื่อสารข้อมูลระหว่าง DAS กับไมโครคอมพิวเตอร์สามารถสรุปได้เป็น 7 ขั้นตอน คือ

ขั้นที่ 1 กำหนดอัตราเร็วในการส่งข้อมูล และรูปแบบของข้อมูลที่ใช้สื่อสารระหว่าง DAS กับไมโครคอมพิวเตอร์ ดังนี้

Baud Rate	9600 Bit/Sec
Data	8 Bit
Stop Bit	1 Bit
Parity	None

ขั้นที่ 2 ไมโครคอมพิวเตอร์ส่งคำสั่ง Link Active ให้กับ DAS เพื่อให้ DAS เตรียมพร้อมที่จะทำงาน

ขั้นที่ 3 DAS ตอบรับคำสั่ง Link Active และส่งข้อมูลตอบรับการติดต่อให้กับไมโครคอมพิวเตอร์

ขั้นที่ 4 ไมโครคอมพิวเตอร์จะแจ้งให้ทราบว่าการติดต่อเรียบร้อย โดยแสดงข้อความบนจอภาพว่า "Link Active" แต่ถ้าไม่ได้รับการติดต่อจะแสดงข้อความ "Time Out Error !"

ขั้นที่ 5 ไมโครคอมพิวเตอร์จะส่งคำสั่ง Trigger Active ไปให้ DAS ในช่วงนี้ DAS จะรอสัญญาณทริกเกอร์เพื่อเริ่มการทำงาน ไมโครคอมพิวเตอร์จะพิมพ์ "Wait for Trigger" อยู่ช่วงระยะเวลาหนึ่ง ถ้าไม่มีสัญญาณทริกเกอร์มาเลย โปรแกรมจะรายงาน "Time Out Error !" และยกเลิกการติดต่อ

ขั้นที่ 6 เมื่อ DAS รับสัญญาณทริกเกอร์ ก็จะเริ่มเก็บข้อมูลและคอยส่งข้อมูลให้กับไมโครคอมพิวเตอร์ โดยที่ไมโครคอมพิวเตอร์จะส่งคำสั่ง XON มาก่อนเพื่อให้ DAS ส่งข้อมูลมาให้ 8 ไบท์ จากนั้นจะส่งคำสั่ง XOFF ไปให้ DAS เพื่อให้หยุดการส่งข้อมูลชั่วคราว เพื่อโปรแกรมจะนำข้อมูลไปประมวลผลต่อ ทั้งนี้โปรแกรมจะอ่านข้อมูลไปเรื่อยๆ จนกว่าจะครบเวลาที่กำหนดไว้ หรือเมื่อผู้ใช้ต้องการหยุดรับข้อมูล

ขั้นที่ 7 เมื่อไมโครคอมพิวเตอร์ไม่ต้องการรับข้อมูลอีกต่อไป จะส่งคำสั่ง Break ไปให้ DAS เพื่อยกเลิกการทำงานและกลับไปสู่สภาวะเริ่มต้นอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การใช้งานชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ในการใช้งานชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC จะต้องติดตั้งอุปกรณ์และซอฟต์แวร์ก่อนที่จะนำไปวิเคราะห์สัญญาณจากเครื่อง HPLC ในหัวข้อนี้จะแบ่งออกเป็น การติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC และการใช้โปรแกรมในการวิเคราะห์ผลข้อมูลของ HPLC โดยมีรายละเอียดดังต่อไปนี้

5.1 การติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ในการติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC จะประกอบไปด้วยอุปกรณ์ต่างๆ คือ

5.1.1 ชุดเก็บและรับส่งข้อมูล (DAS)

5.1.2 สาย RS232C สำหรับใช้ติดต่อสื่อสารข้อมูลระหว่าง DAS กับไมโครคอมพิวเตอร์

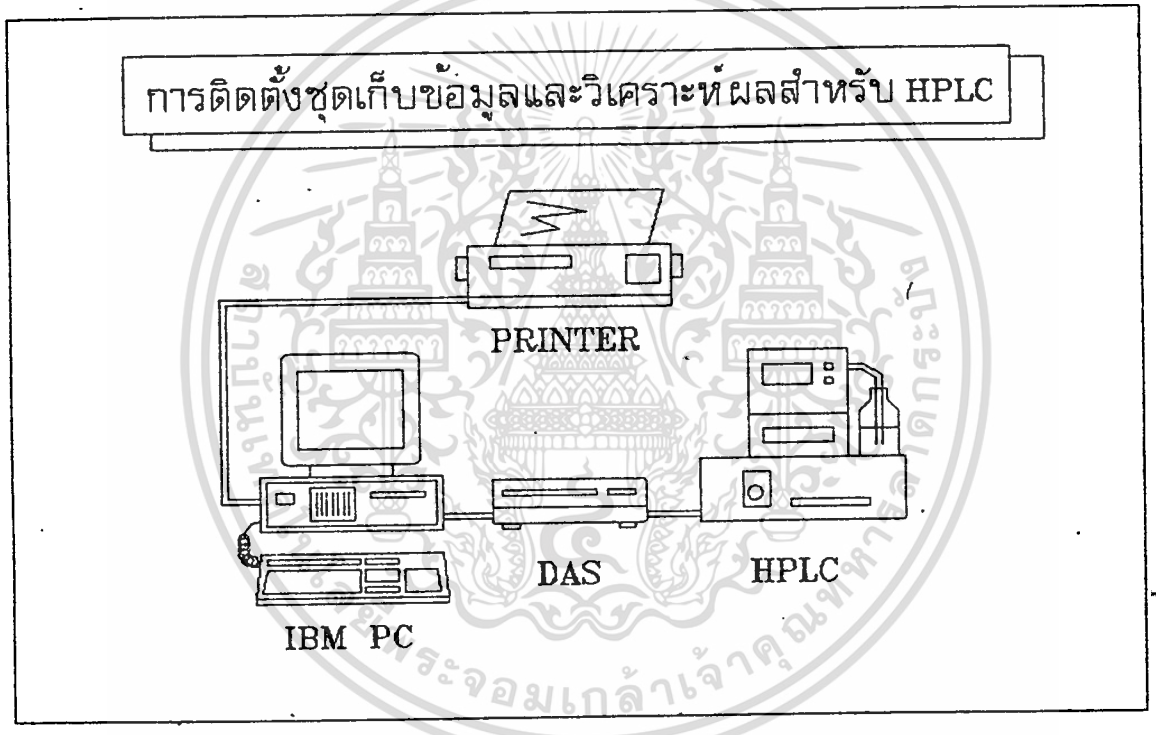
5.1.3 เครื่องไมโครคอมพิวเตอร์(IBM PC XT,AT,386,486 or Compatible PC)
- Memory 640 KB
- Floppy Disk Drive 1 ตัว
- Monitor EGA/VGA Color หรือ VGA Monochrome
- Dos 3.31 หรือใหม่กว่า
- Serial Port 1 Port (for DAS)
- Parallel Port 1 Port (for Printer)

5.1.4 ซอฟต์แวร์ HPLC-Soft version 1.2 (for EGA/VGA Monitor)
และแผ่นจานแม่เหล็กสำหรับบันทึกข้อมูล

5.1.5 เครื่องพิมพ์แบบ Dot Matrix 8 เข็ม หรือ 24 เข็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดตั้งจะเริ่มจากการต่ออุปกรณ์ต่างๆ เข้าด้วยกัน ดังรูปที่ 5.1 จากนั้นจึงเปิดเครื่องคอมพิวเตอร์โดยอาจจะ Boot เครื่องจาก Hard Disk หรือ แผ่นจานแม่เหล็กที่มี DOS อยู่ก็ได้ ใส่แผ่นโปรแกรม HPLC เข้าไปใน Floppy Disk Drive เปลี่ยนช่อง Drive และ Directory ไปที่ที่มีโปรแกรม HPLC พิมพ์คำว่า HPLC ลงไป แล้วกด Enter สักครู่ โปรแกรมจะแสดงรายการการทำงานให้ผู้ใช้เลือกและกำหนดวิธีการวิเคราะห์ให้กับโปรแกรม ในกรณีที่ใช้งานกับ DAS ควรตรวจสอบการต่อสาย RS232C ว่าต่อถูกต้องหรือไม่ เปิดเครื่อง DAS แล้วหรือยัง กำหนด Port การสื่อสารถูกต้องหรือไม่ มิฉะนั้นเครื่องคอมพิวเตอร์จะไม่สามารถติดต่อรับข้อมูลจาก DAS ได้



รูปที่ 5.1 แสดงการติดตั้งชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การใช้โปรแกรมในการวิเคราะห์ผลข้อมูลของ HPLC

โปรแกรมวิเคราะห์ผลข้อมูลของ HPLC จะมีรายการต่างๆ ให้เลือก 7 รายการด้วยกัน โดยรายการต่าง ๆ จะเรียงลำดับตามการใช้งานเพื่อให้สามารถใช้งานได้ง่ายขึ้น ในการเลือกรายการใช้งานให้ใช้ปุ่มลูกศรขึ้น-ลง รายการที่ถูกเลือกจะมีแถบสว่างสีเหลืองปรากฏอยู่ เมื่อต้องการเข้าไปในเมนูเพื่อต้องการใช้งาน ให้กดปุ่ม ENTER รายการทั้ง 7 รายการมีดังต่อไปนี้

5.2.1) Mode

5.2.2) Option

5.2.3) Method

5.2.4) File

5.2.5) Run

5.2.6) Report

5.2.7) Quit

5.2.1 Mode ในการใช้งาน เริ่มแรกจะต้องเลือก Mode การใช้งานให้เหมาะสม ซึ่งในที่นี้จะมีอยู่ 2 Mode คือ Labnet Mode และ File Mode ใน Labnet Mode จะเป็นการกำหนดให้ซอฟต์แวร์รับข้อมูลจากเครื่อง DAS โดยผ่านทางสาย RS232C และสามารถเลือกพอร์ทที่ติดต่อกับไมโครคอมพิวเตอร์ได้ทั้ง Com1 และ Com2 ซึ่งพอร์ทนี้เป็นพอร์ทรับส่งข้อมูลแบบอนุกรม เรียกว่าพอร์ทอนุกรม(Serial Port) โดยปกติแล้วไมโครคอมพิวเตอร์จะมีพอร์ทอนุกรมไว้ให้ใช้ 2 พอร์ท ซึ่งพอร์ทอนุกรมนี้จะมีคอนเนคเตอร์(Connector) 2 แบบ คือชนิด DB9 และชนิด DB25 โดยเป็นคอนเนคเตอร์แบบตัวผู้มี 9 ขา (DB9) และ 25 ขา (DB25) ในการใช้งานต้องต่อสาย RS232C เข้ากับพอร์ทอนุกรมนี้ พอร์ทใดพอร์ทหนึ่ง แล้วไปกำหนดพอร์ทในรายการหลัก Option ที่รายการ Serial Port ให้ตรงกับการใช้งาน ในภาวะการใช้งานจะต้องต่อสาย RS232C เข้ากับเครื่อง DAS ให้เรียบร้อย และเปิดเครื่อง DAS ไว้รอรับข้อมูล

สำหรับ File Mode จะเป็นการอ่านข้อมูลจากไฟล์ที่บันทึกไว้แล้วมาวิเคราะห์ ซึ่งสามารถกำหนดค่าพารามิเตอร์ต่าง ๆ ที่ใช้ในการวิเคราะห์ได้จากรายการ Option และกำหนดวิธีการวิเคราะห์ได้จากรายการ Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 Option เป็นรายการที่ใช้กำหนดการทำงาน การแสดงผลบนจอภาพและพารามิเตอร์ที่ใช้ในการคำนวณต่าง ๆ ซึ่งมีทั้งหมด 10 รายการด้วยกัน ในการเลือกรายการจะใช้ลูกศรขึ้น-ลง และใช้ SPACE BAR ในการแก้ไขเปลี่ยนแปลงข้อมูล และเมื่อเสร็จแล้วให้กดปุ่ม ESC เพื่อกลับไปที่รายการหลัก รายการต่าง ๆ 10 รายการมีดังต่อไปนี้

5.2.2.1 Runtime เป็นการตั้งค่าเวลาของการเก็บข้อมูล (ใช้กับ Labnet Mode) ซึ่งช่วงเวลาการเก็บข้อมูลสามารถเลือกได้ตั้งแต่ 10 Min., 15 Min., 20 Min., 30 Min., 45 Min., 60 Min., 75 Min., 5 Min. : ค่าปกติ <10 Min.> ; Min. = นาที

5.2.2.2 Atten เป็นการกำหนดขนาดของสัญญาณที่จะแสดงบนจอภาพ กรณีที่สัญญาณมีขนาดเล็กควรเลือกค่า Atten ค่าน้อย ๆ แต่ถ้าสัญญาณมีขนาดใหญ่ควรเลือกค่า Atten ให้มากขึ้นตามความเหมาะสม การเลือกค่า Atten ที่ไม่เหมาะสมจะทำให้การแสดงรูปสัญญาณบนจอภาพมีขนาดเล็กเกินไปแต่ไม่มีผลกระทบต่อการคำนวณหาค่ารีเทนชันไทม์ และพื้นที่ของพีคแต่อย่างใด

5.2.2.3 Page Range การแสดงรูปสัญญาณบนจอภาพจะมีอยู่ 2 หน้าต่างคือหน้าต่างด้านบนซึ่งจะแสดงสัญญาณที่ปรากฏขึ้นในช่วงเวลาที่กำหนดไว้สูงสุด(กรณีของ Labnet Mode) หรือช่วงเวลาที่ใช้ในการแสดงรูปสัญญาณทั้งหมด(กรณีของ File Mode) และอีกหน้าต่างหนึ่งซึ่งอยู่ทางด้านล่างขวาจะแสดงรูปสัญญาณในช่วงเวลาย่อยๆ ซึ่งกำหนดได้จาก PageRange โดยสามารถเลือกกำหนดค่า Page Range ได้ตั้งแต่ 4 Min., 6 Min., 8 Min., 10 Min., 15 Min., 20 Min., 30 Min., 1 Min., 2 Min. : ค่าปกติ <4 Min.> ; Min. = นาที

5.2.2.4 Peak Width เป็นค่าพารามิเตอร์ ที่จะต้องนำไปใช้ในการคำนวณ เพื่อหาจุดเริ่มต้นและจุดปลายของพีค เช่นเดียวกับค่าระดับตรวจจับพีค ค่า Peak Width ที่มีค่ามากจะช่วยให้สามารถตรวจหาจุดเริ่มต้นและจุดปลายของพีคที่มีความลาดชันน้อยได้ อย่างไรก็ตามการตั้งค่า Peak Width อาจทำให้ค่ารีเทนชันไทม์เปลี่ยนแปลงไปบ้างเล็กน้อย และอาจทำให้ไม่สามารถตรวจสอบพีคที่อยู่ติดกันมากๆ ได้ โดยปกติแล้วควรใช้ค่า 1 Sec. เว้นแต่ในกรณีที่ต้องการหาพื้นที่ของพีคที่มีความลาดชั้นน้อยๆเท่านั้น และสามารถเลือกค่า Peak Width ได้ตั้งแต่ 1 Sec., 2 Sec., 4 Sec., 6 Sec., 8 Sec., 10 Sec., 12 Sec. : ค่าปกติ <1 Sec.> ; Sec. = วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.5 Threshold หรือค่าระดับตรวจจับพีคเป็นค่าพารามิเตอร์ที่ใช้ในการคำนวณหาจุดเริ่มต้นและจุดปลายของพีค ถ้ากำหนดค่า Threshold น้อยๆ ก็สามารถตรวจหาพีคขนาดเล็กๆ ได้ดี แต่ถ้าไม่สนใจพีคขนาดเล็กๆ หรือเพื่อหลีกเลี่ยงพีคที่เกิดจากสัญญาณรบกวนขนาดเล็กๆ สามารถกำหนดค่า Threshold ให้สูงขึ้นได้ตามความเหมาะสม : ค่าปกติ <400>

5.2.2.6 Serial Port เป็นการกำหนดพอร์ทของเครื่องไมโครคอมพิวเตอร์ที่ใช้ติดต่อกับเครื่องDAS มีให้เลือกได้ 2 พอร์ท คือ Com1 และ Com2

5.2.2.7 Peak Window เป็นการกำหนดช่วงเวลาการยอมรับได้ของการตรวจสอบชนิดของสารจากการเปรียบกับสารมาตรฐานที่กำหนดไว้ ซึ่งใช้ในวิธี External Std. Method โดยช่วงเวลาที่ยอมรับได้นี้ จะคิดเป็นช่วง $\pm d\%$ ของค่ารีเทนชันไทม์ของพีคมาตรฐาน และกรณีที่พบพีคหลายพีคในช่วงเวลาดังกล่าว จะถือเอาพีคที่อยู่ใกล้ค่ารีเทนชันไทม์ของพีคมาตรฐานมากที่สุดเป็นหลัก ค่า Peak Window นี้จะมีช่วงตั้งแต่ 10%, 12%, 15%, 20%, 4%, 6%, 8% : ค่าปกติ <10%>

5.2.2.8 Peak Marker เป็นการกำหนดสัญญลักษณ์แสดงจุดเริ่มต้น และจุดปลายพีค โดยที่จุดเริ่มต้นพีคจะแสดงเป็นขีดยาวสีเขียว ที่บริเวณด้านล่างติดกับสเกลแสดงเวลาของหน้าต่างแสดงสัญญาณทางด้านล่างขวา ส่วนจุดปลายพีคจะแสดงเป็นขีดสั้นสีแดง ซึ่งสามารถกำหนดให้แสดงหรือไม่แสดงก็ได้โดยกำหนดเป็น ON, OFF : ค่าปกติ <ON>

5.2.2.9 Filter เป็น Option ที่ใช้สำหรับกำหนดให้มีการกรองสัญญาณให้เรียบขึ้น โดยใช้วิธี Moving Average แบบ 5 จุด สามารถใช้ในการกำจัดสัญญาณรบกวนขนาดเล็กๆ ได้ดี สามารถกำหนดสถานะการกรองสัญญาณได้ 2 แบบ โดยกำหนดเป็น ON, OFF : ค่าปกติ <OFF>

5.2.2.10 Baseline เป็นการกำหนดระดับ Baseline ที่ใช้ในการคำนวณหาพื้นที่พีคซึ่งจะมีค่าคงที่ค่าหนึ่ง ดังนั้นจึงยังมีข้อจำกัดในการใช้งานกับการวัดที่มีค่า Baseline คงที่เท่านั้น การตั้งค่า Baseline อาจทำได้ 2 วิธี คือ การกำหนดค่า Baseline โดยตรง และการใช้วิธี Baseline Cal. ในรายการ Method ซึ่งจะช่วยหา Baseline อัตโนมัติ : ค่าปกติ <1000>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 Method เราสามารถกำหนดวิธีการวิเคราะห์ได้จาก Method ซึ่งมีอยู่ 3 วิธีด้วยกัน คือ

5.2.3.1 Normal Method เป็นวิธีการวิเคราะห์แบบปกติ กล่าวคือ จะวิเคราะห์หาค่ารีเทนชันไทม์ และค่าพื้นที่พีคของสัญญาณที่ตรวจสอบ โดยใช้พารามิเตอร์ในการคำนวณจากรายการ Option

5.2.3.2 External Std. Method เป็นวิธีการวิเคราะห์เพื่อหาความเข้มข้นของสารที่ต้องการตรวจสอบโดยอาศัยข้อมูลจากสารมาตรฐาน ซึ่งต้องทราบชนิดของสาร ค่ารีเทนชันไทม์ และค่าเรสปอนด์แฟกเตอร์(RF) ซึ่งหาได้จาก พื้นที่พีค/ความเข้มข้นของสารมาตรฐาน

5.2.3.3 Baseline Cal. Method เป็นวิธีการหาค่าระดับ Baseline อัตโนมัติ โดยการ Run ตัวทำละลายเปล่า และคำนวณระดับ Baseline โดยการเฉลี่ยระดับของสัญญาณในช่วงเวลาที่ Run ทั้งหมด

5.3.4 File ในการทำงานทั้งใน Labnet Mode และ File Mode เราสามารถกำหนดชื่อไฟล์(File name) และไดเรกทอรี(Directory) ที่ต้องการใช้งานได้ โดยกำหนดในรายการ File ซึ่งมีหัวข้อรายการให้เลือก 2 อย่าง คือ Data File ใช้สำหรับกำหนดชื่อไฟล์ที่ต้องการใช้ โดยในกรณีของ Labnet Mode จะเป็นชื่อไฟล์ที่ใช้ในการบันทึกข้อมูล และในกรณีของ File Mode จะเป็นการกำหนดชื่อไฟล์ที่ต้องการอ่านข้อมูลเข้ามาวิเคราะห์ อีกรายการหนึ่ง คือ Data Dir เป็นการกำหนดชื่อไดเรกทอรีที่ใช้บันทึกหรืออ่านไฟล์ข้อมูล อย่างไรก็ตาม ในการใช้งานจริงจะมีข้อความบอกให้กำหนดหรือเปลี่ยนแปลงชื่อไฟล์และไดเรกทอรีได้ ก่อนที่จะอ่านหรือบันทึกไฟล์ข้อมูลนั้น

5.3.5 Run เป็นการสั่งให้นำข้อมูลเข้ามาวิเคราะห์ตาม Mode และวิธีการที่กำหนด โดยจะแสดงลักษณะของสัญญาณที่รับเข้ามา และแสดงผลการวิเคราะห์พีคสัญญาณดังกล่าว

5.3.6 Report จะเป็นการรายงานผลออกทางเครื่องพิมพ์ ซึ่งจะประกอบไปด้วยรูปสัญญาณและรายงานผลของพีคที่ตรวจพบ ดังนี้ หมายเลขพีคที่พบ ค่ารีเทนชันไทม์ของพีค ค่าพื้นที่พีค และในกรณีที่เป็นExternal Std. Method จะมีการรายงานผลของพีคที่ต้องการตรวจสอบหาว่า มีพีคของสารที่ต้องการตรวจอยู่หรือไม่ และมีความเข้มข้นเท่าใด

5.3.7 Quit เป็นการยกเลิกการทำงานของโปรแกรมและกลับเข้าสู่ DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม HPLC-SOFT ที่พัฒนาขึ้นมีขีดความสามารถ ดังนี้

ใช้ในการวิเคราะห์ข้อมูลของพีคสัญญาณต่างๆ โดยสามารถทำงานได้ 2 แบบ คือ

- Labnet Mode เป็นการวิเคราะห์ข้อมูลในขณะเก็บข้อมูลจากเครื่อง HPLC
- File Mode เป็นการวิเคราะห์ข้อมูลจากไฟล์ที่บันทึกไว้

สามารถกำหนดเวลาและพารามิเตอร์ต่างๆ ในการวิเคราะห์ผล

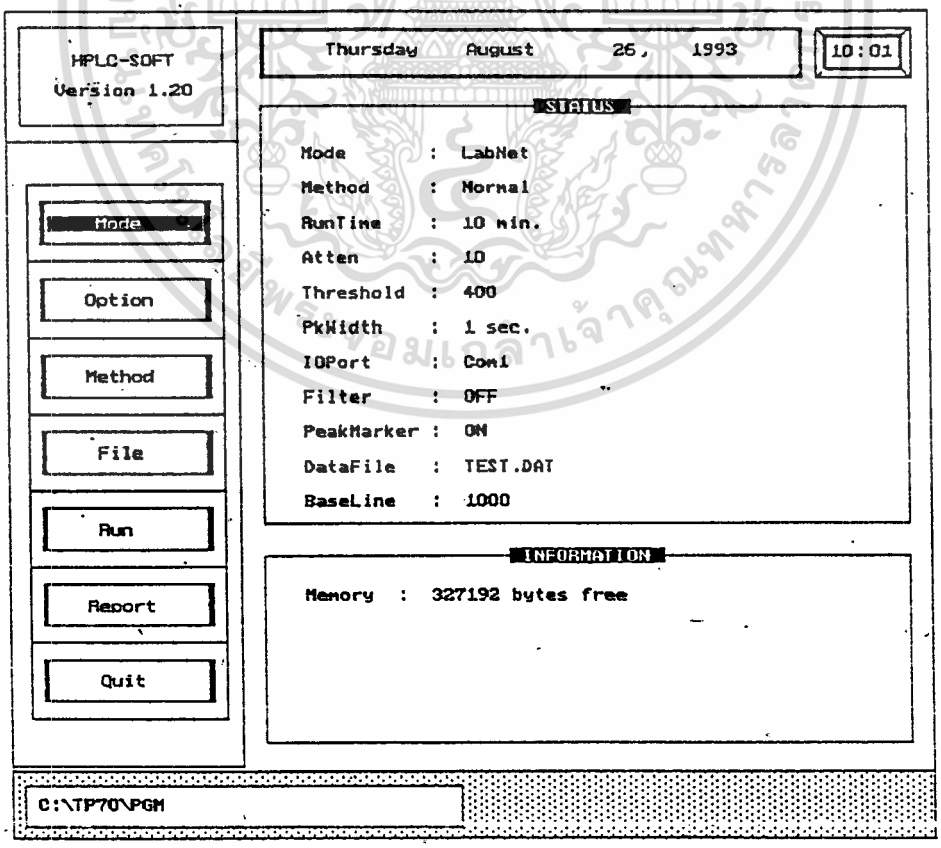
มีวิธีการวิเคราะห์ 3 วิธี คือ

- Normal เป็นวิธีการวิเคราะห์เบื้องต้น เพื่อหาค่ารีเทนชันไทม์และพื้นที่ของพีค
- External Standard เป็นวิธีการวิเคราะห์เพื่อหาชนิดและความเข้มข้นของสาร โดยใช้สารมาตรฐานเป็นตัวเปรียบเทียบ
- Baseline Calibration เป็นวิธีการวิเคราะห์หาค่าระดับฐานพีคจากการเดินตัวทำละลายเพียงอย่างเดียว

สามารถกรองสัญญาณรบกวนด้วยวิธี Moving Average แบบ 5 จุด

สามารถบันทึกข้อมูลลงบนแผ่นจานแม่เหล็ก

การรายงานผลสามารถแสดงทางจอภาพ(สี) และทางเครื่องพิมพ์

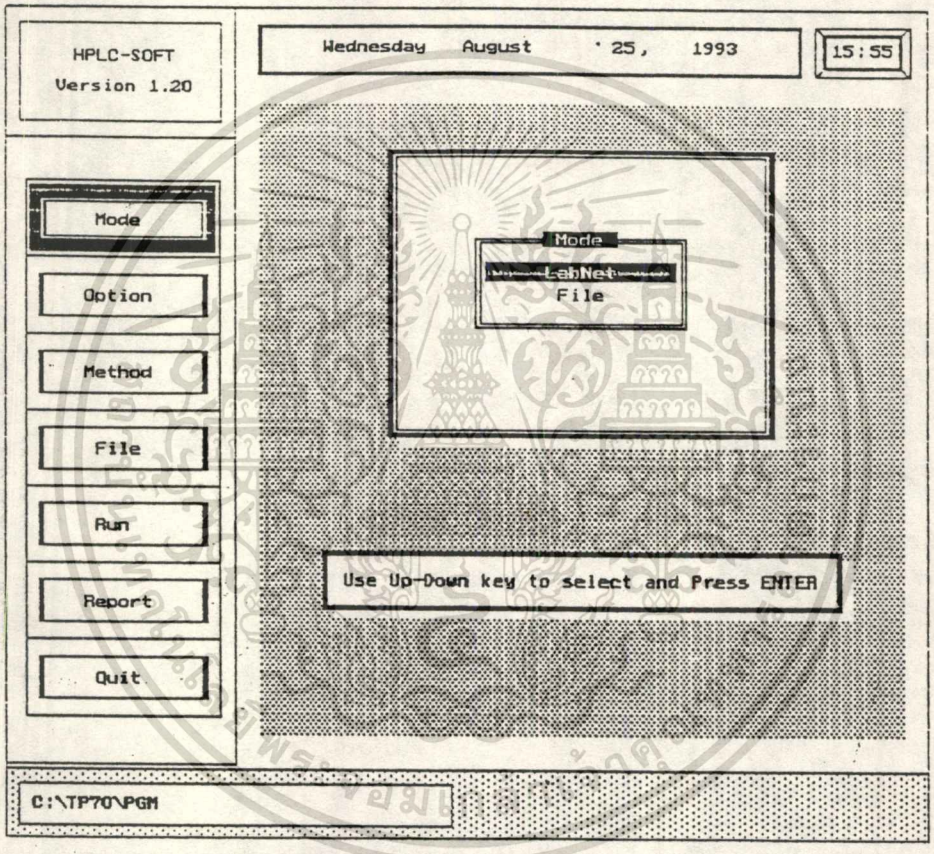


รูปที่ 5.2 แสดงลักษณะของโปรแกรมวิเคราะห์ข้อมูลสำหรับ HPLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานสามารถสรุปเป็นขั้นตอนได้ 7 ขั้นตอนดังนี้

ขั้นที่ 1 จากแหล่งข้อมูลที่ใช้ในการวิเคราะห์ จะนำมากำหนด Mode ของการทำงาน คือในกรณีที่ใช้ DAS เก็บข้อมูล ก็จะใช้ Labnet Mode และในกรณีที่ใช้ข้อมูลจาก File ที่บันทึกเก็บไว้ก็จะใช้ File Mode โดยการเลือกรายการ Mode แล้วกดคีย์ Enter และใช้คีย์ลูกศรในการเลือก Mode ที่ต้องการ แล้วกดคีย์ Enter อีกครั้ง



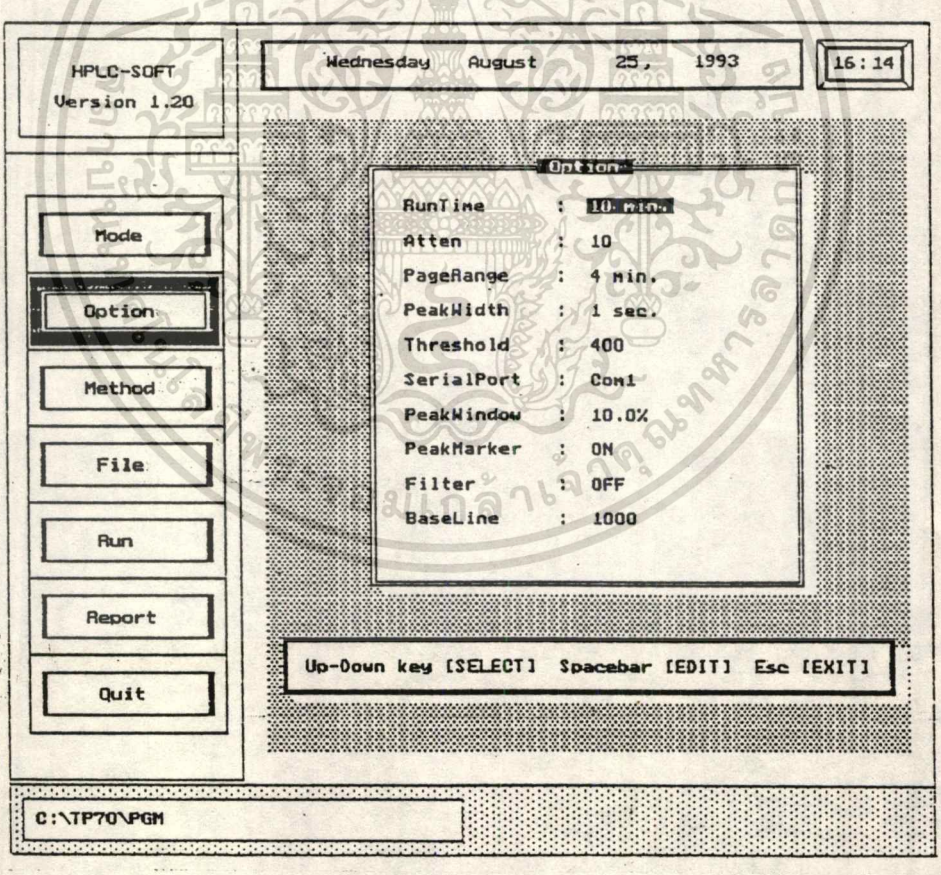
รูปที่ 5.3 แสดงการกำหนด Mode การทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 เลือกหัวข้อ Option แล้วกำหนดค่า Option ต่างๆ ที่ใช้ เช่น Runtime (กรณีของ Labnet Mode), Atten, Page Range, Peak Width, Threshold, Serial Port (กรณีของ Labnet Mode), Peak Window (กรณีของ External Std. Method), Peak Marker, Filter และ Baseline

ในกรณีที่ไม่ทราบค่าว่าควรกำหนดเท่าใด ก็ให้ใช้ตามที่กำหนดไว้ก่อน และในการใช้งานจะต้องกำหนดค่า Baseline ก่อน โดยการใช้ Baseline Cal. Method ซึ่งจะต้องใช้ข้อมูลจากการ Run โมบายเฟสเปล่าที่มีระดับ Baseline คงที่ แล้วโปรแกรมจะคำนวณหาค่า Baseline เฉลี่ยให้โดยอัตโนมัติ

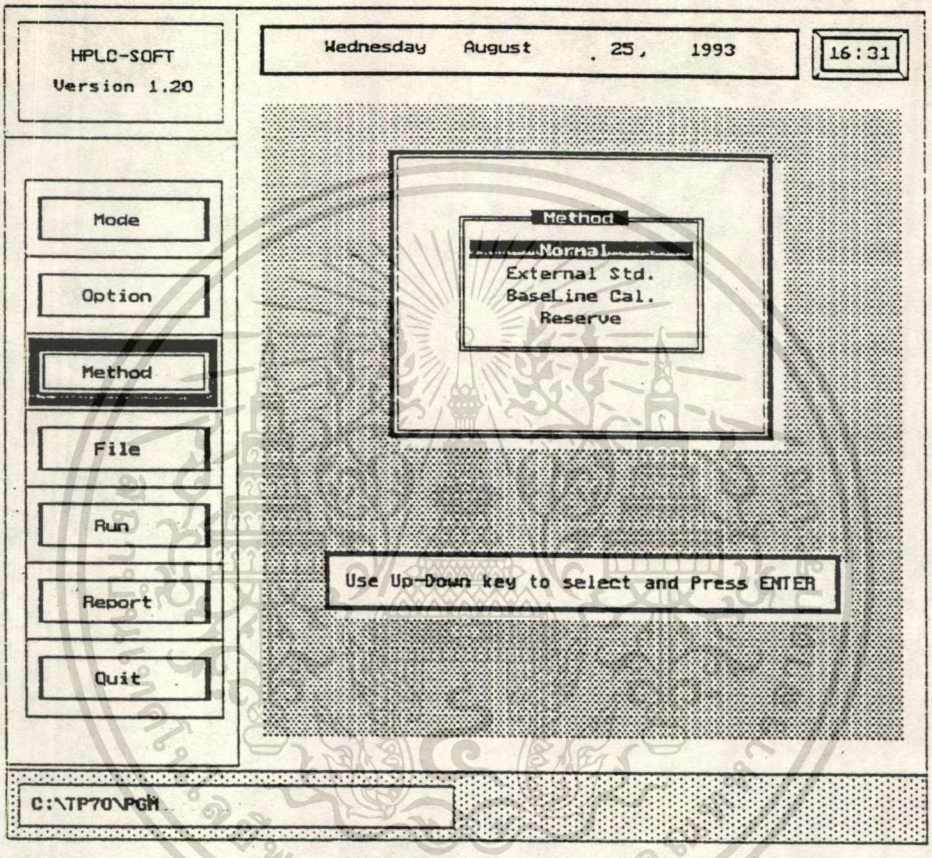
ในการเปลี่ยนค่าใน Option ให้ใช้คีย์ Spacebar สำหรับเปลี่ยนค่า โดยจะมีลักษณะการเปลี่ยนค่าอยู่ 2 แบบ คือ การเปลี่ยนค่าตามที่กำหนดเอาไว้ในโปรแกรมอยู่แล้ว เช่น Runtime, Atten, Peak Width เป็นต้น และการเปลี่ยนค่าโดยการแก้ไขค่าที่กำหนดไว้เดิมโดยพิมพ์แก้ไขทางคีย์บอร์ด เช่น Threshold และ Baseline



รูปที่ 5.4 แสดงการกำหนดพารามิเตอร์ในรายการ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 เลือกวิธีการวิเคราะห์จากหัวข้อ Method ในกรณีที่ยังไม่ได้หาค่า Baseline ควรใช้ Baseline Cal. Method ก่อน โดยการ Run โมบายเฟสแล้วหาค่า Baseline เฉลี่ย ซึ่งโปรแกรมจะคำนวณหาค่า Baseline เฉลี่ย แล้วจะกำหนดค่า Baseline ให้โดยอัตโนมัติ แต่ถ้าเป็นกรณีของ File ให้ Run File ข้อมูลของโมบายเฟสเปล่าๆ ที่เหมาะสมแทน โดยการใช้ Baseline Cal. Method เช่นเดียวกัน

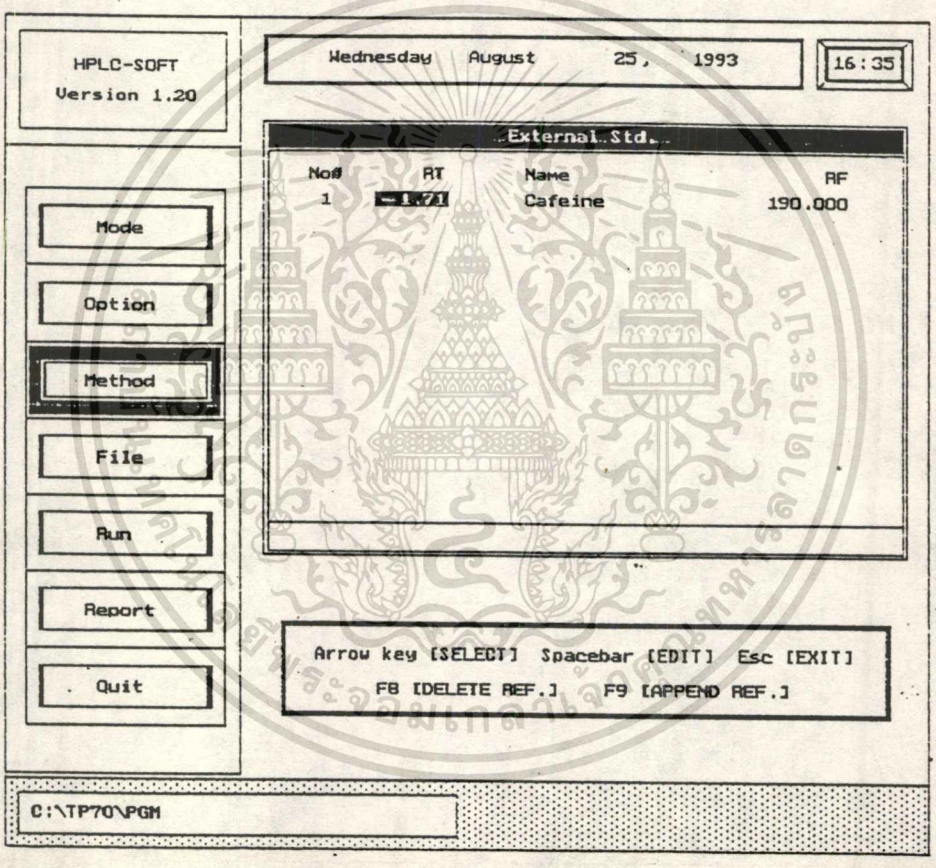


รูปที่ 5.5 แสดงการเลือกวิธีวิเคราะห์จากรายการ Method

เมื่อต้องการวิเคราะห์ใน Normal Method ก็สามารถทำได้เลย แต่ในกรณีที่เป็น External Std. Method จะต้องมีสารมาตรฐานมาเปรียบเทียบ โดยการกำหนดชื่อสาร ค่ารีเทนชันไทม์และค่าเรสปอนด์แฟกเตอร์ เก็บไว้ในตาราง ซึ่งเก็บได้สูงสุด 12 รายการด้วยกัน ดังนั้นในกรณีของการใช้งานใน External Std. Method จะต้อง Run สารมาตรฐานต่างๆ ซึ่งทราบชื่อและความเข้มข้นแล้วใน Normal Method เมื่อหาค่ารีเทนชันไทม์และค่าพื้นที่พีค ซึ่งค่าเรสปอนด์แฟกเตอร์นี้ สามารถคำนวณได้จากการหาค่าอัตราส่วนระหว่างพื้นที่พีค/ความเข้มข้นของสาร ควรทำหลายๆ ครั้ง ที่ความเข้มข้นต่างๆ กัน เพื่อหาค่าเรสปอนด์แฟกเตอร์ที่มีความถูกต้องมากขึ้น และจึงนำข้อมูลที่ได้ไปใส่ไว้ในตาราง เพื่อใช้ในการเปรียบเทียบหาชนิดและความเข้มข้นของสารตัวอย่างต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

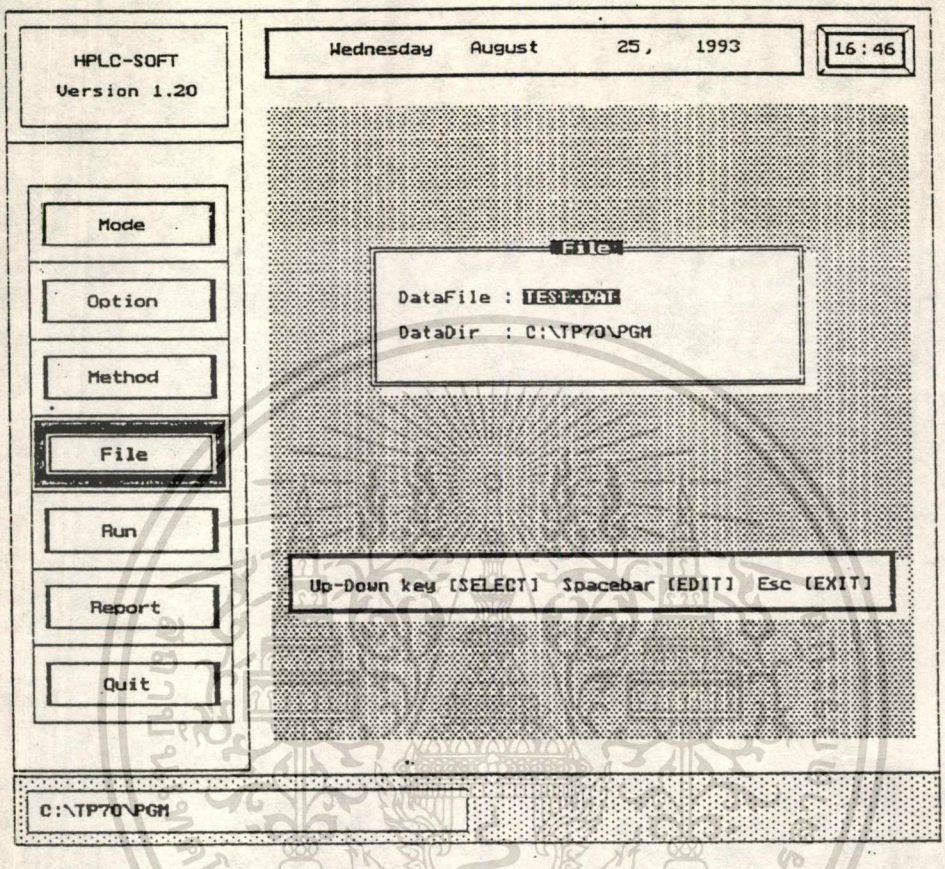
สำหรับการใส่ข้อมูลเพื่อใช้ในการวิเคราะห์ ในกรณีของ External Std. เมื่อใส่ข้อมูลครั้งแรก โปรแกรมจะรออ่านข้อมูลทีละค่า คือ ค่ารีเทนชันไทม์(RT) ของสารมาตรฐาน ชื่อของสารมาตรฐาน(Name) และค่าเรสปอนด์แฟกเตอร์(RF) ถ้าต้องการใส่ข้อมูลเพิ่มให้กดคีย์ ENTER โปรแกรมจะรออ่านข้อมูลชุดถัดไป และเมื่อใส่ครบแล้วให้กดคีย์ ESC โปรแกรมจะขึ้นแถบสว่างสำหรับให้เลือกข้อมูลที่จะแก้ไข โดยใช้คีย์ลูกศรเป็นตัวเลื่อนแถบสว่าง และใช้คีย์ Spacebar ในการแก้ไขข้อมูล และยังมีคีย์ F8 สำหรับลบรายการที่ไม่ใช้ออก และคีย์ F9 สำหรับเพิ่มรายการ เมื่อแก้ไขแล้วให้กดคีย์ ESC เพื่อกลับไปยังรายการหลัก



รูปที่ 5.6 แสดงการใส่ข้อมูลของสารมาตรฐานใน External Std. Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 4 กำหนดชื่อไฟล์และไดเรกทอรีที่ต้องการใช้งาน เช่น การ Save ข้อมูล (กรณีของ Labnet Mode) และการ Load ข้อมูล (กรณีของ File Mode)



รูปที่ 5.7 แสดงการกำหนดชื่อไฟล์ข้อมูลและไดเรกทอรีที่ต้องการใช้งาน

การแก้ไขชื่อไฟล์สามารถทำได้โดยการใช้ลูกศรเลือกรายการและกดคีย์ Spacebar เพื่อแก้ไขชื่อไฟล์หรือไดเรกทอรีที่ต้องการ ในกรณีการป้อนชื่อไฟล์และไดเรกทอรีใหม่โดยการใส่ข้อมูลเพียงครั้งเดียวก็สามารถทำได้จากการใส่ชื่อไฟล์พร้อมกับชื่อ Drive:[Path] เช่น B:\Data\Caffeine.dat ลงในรายการ Data File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

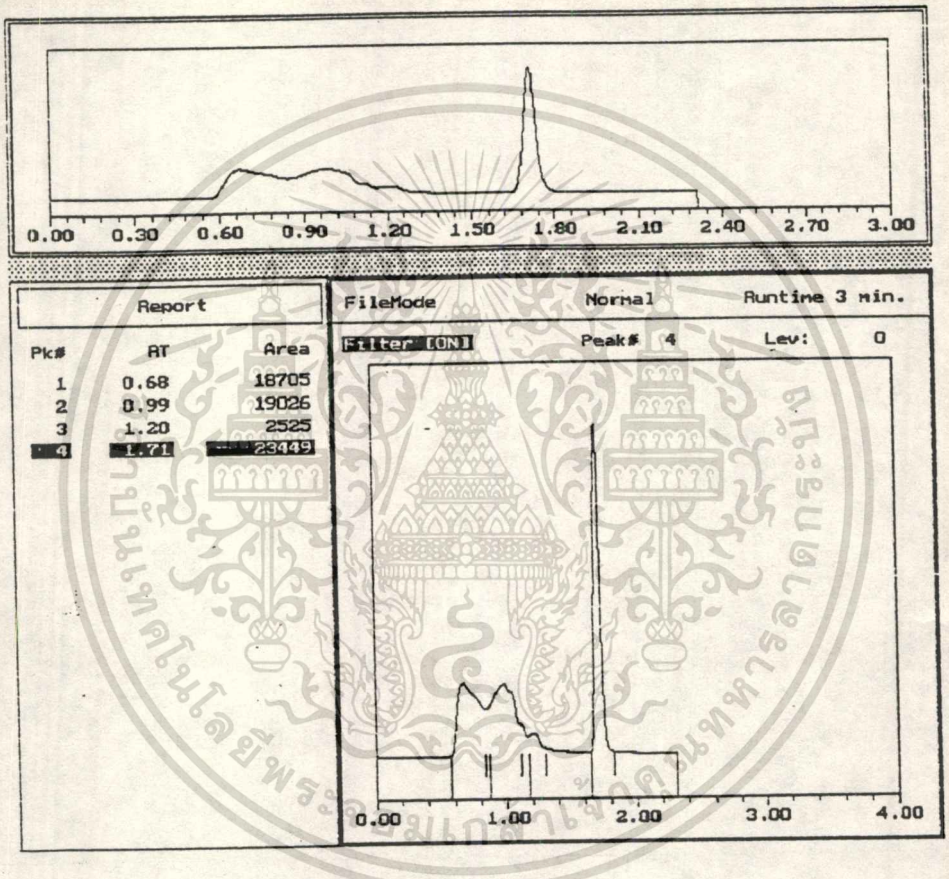
ขั้นที่ 5 เป็นขั้นตอนของการแสดงผลและวิเคราะห์ข้อมูล

ในขั้นนี้จะมีการแสดงผลและวิเคราะห์ข้อมูลสัญญาณ ดังรูปที่ ประกอบไปด้วย 3 ส่วน หลักๆ คือ

A. หน้าต่างแสดงผลของสัญญาณตลอดช่วงเวลาทั้งหมด

B. หน้าต่างแสดงผลบางช่วงของสัญญาณ

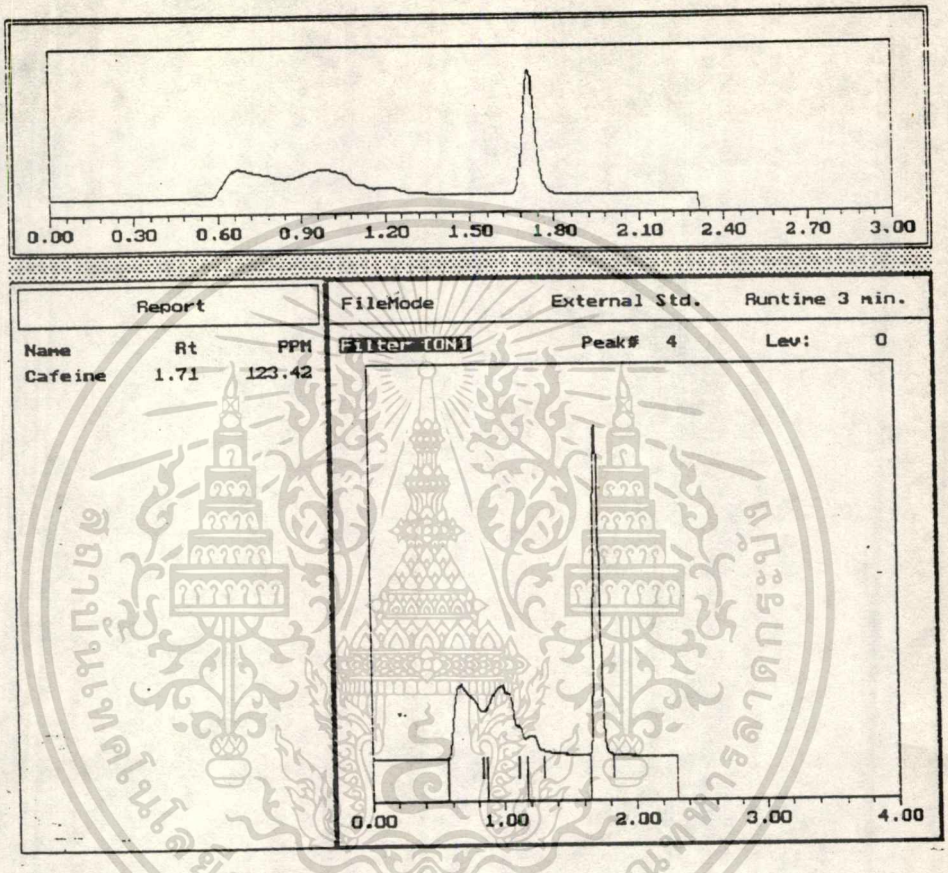
C. หน้าต่างแสดงผลการวิเคราะห์พีคสัญญาณ



รูปที่ 5.8 แสดงกราฟสัญญาณและการรายงานผลทางจอภาพ(Normal Method)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

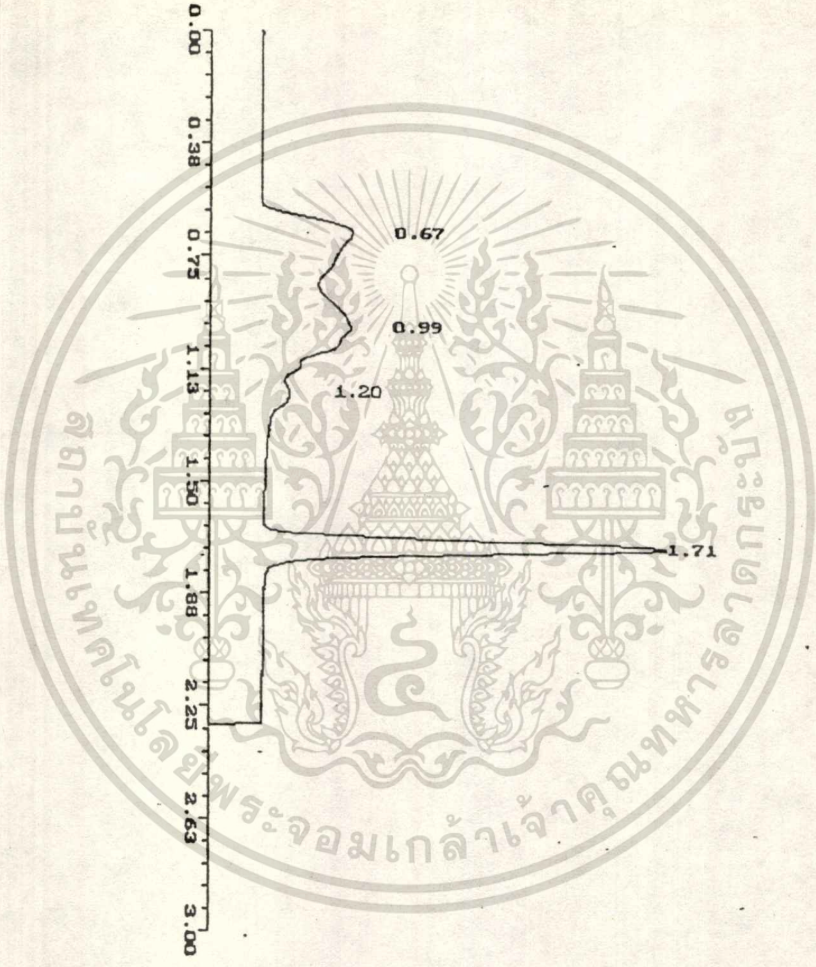
ในกรณีที่เป็น Labnet Mode เมื่อโปรแกรมทำงานจนครบเวลาที่กำหนดแล้ว จะหยุดรับสัญญาณ และถามว่าต้องการบันทึกข้อมูลเก็บไว้ในแผ่นจานแม่เหล็กหรือไม่ ให้เลือกว่าจะบันทึก หรือยกเลิกการบันทึกข้อมูลจากนั้นจึงออกจากการแสดงผลนี้โดยการกด Enter เพื่อกลับไปรายการหลัก



รูปที่ 5.9 แสดงกราฟสัญญาณและการรายงานผลทางจอภาพ(External Std. Method)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 6 การแสดงผลทางเครื่องพิมพ์ สามารถทำได้ โดยการเลือกรายการ Report โปรแกรมจะวาดรูปสัญญาณ และแสดงค่ารีเทนชันไทม์ที่พบบนพีคสัญญาณ และถามว่าต้องการพิมพ์หรือไม่ ถ้ายังยืนยันการพิมพ์อยู่ก็ให้เลือก OK แล้วกดคีย์ Enter โปรแกรมจะพิมพ์รูปสัญญาณออกทางเครื่องพิมพ์ และพิมพ์รายงานผลการวิเคราะห์ทั้งหมดไว้ กระดาษที่ใช้พิมพ์ควรเป็นกระดาษต่อเนื่องจะดีที่สุด



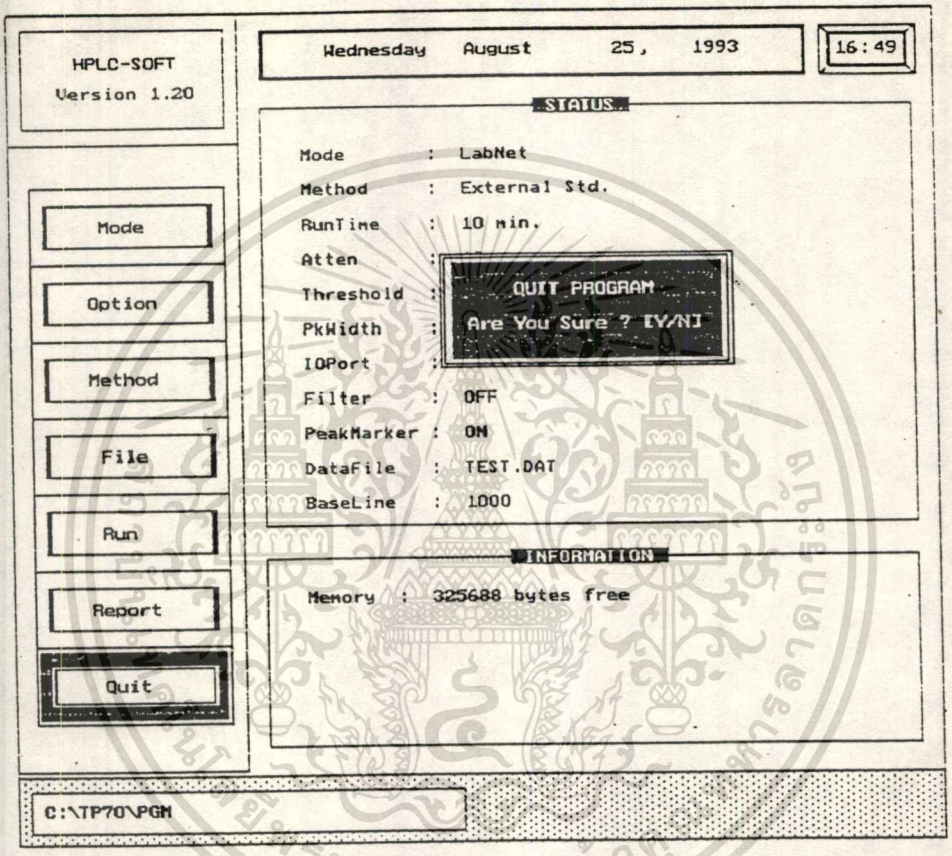
Peak	RT	AREA
1	0.67	18705
2	0.99	19026
3	1.20	2525
4	1.71	23449

Name	RT	PPM
Caffeine	1.71	123.42

รูปที่ 5.10 แสดงตัวอย่างการรายงานผลการวิเคราะห์ เมื่อพิมพ์ออกมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 7 การยกเลิกการทำงาน โดยการเลือกรายการ QUIT แล้วกดคีย์ ENTER โปรแกรมจะถามย้ำการยกเลิกการทำงาน ถ้าต้องการยกเลิกการทำงานให้กดคีย์ Y และคีย์ Enter นอกจากนี้ อาจใช้คีย์ ESC ในขณะที่อยู่ในรายการหลักก็ได้ โปรแกรมจะแสดงข้อความถามย้ำการยกเลิกการทำงาน เช่นเดียวกับกรณีแรก



รูปที่ 5.11 แสดงการยกเลิกการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดสอบการทำงานของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC

ได้ทำการทดลองเพื่อทดสอบการทำงาน และความสามารถของชุดวิเคราะห์ข้อมูลของ HPLC โดยการเปรียบเทียบกับสารมาตรฐานที่ทราบชนิดและความเข้มข้น และเปรียบเทียบกับการวิเคราะห์ข้อมูลที่ได้จากเครื่องประมวลผลของ Spectra Physics รุ่น SP4200 ซึ่งทดลองหาความเข้มข้นของสารคาเฟอีนในเครื่องดื่มโคลา 2 ชนิด โดยสามารถสรุปการทดลองเป็นขั้นตอนได้ดังนี้

6.1 หาค่ารีเทนชันไทม์ของคาเฟอีนจากสารคาเฟอีนมาตรฐาน

6.2 หาค่าเรสปอนด์แฟกเตอร์เฉลี่ยของคาเฟอีน

6.3 หาค่าความเข้มข้นของคาเฟอีน ในเครื่องดื่มโคลา 2 ชนิด

6.4 สรุปผลการทดสอบ

6.1 หาค่ารีเทนชันไทม์ของคาเฟอีนจากสารคาเฟอีนมาตรฐาน

จากการทดลองหาค่า รีเทนชันไทม์ ของคาเฟอีนมาตรฐานจำนวน 10 ครั้ง ระหว่างชุดเก็บข้อมูลและวิเคราะห์ที่พัฒนาขึ้น(DAS001) กับเครื่องประมวลผลข้อมูลของ Spectra Physics รุ่น SP4200 ซึ่งการทดลองในแต่ละครั้งจะใช้สารคาเฟอีนมาตราฐานฉีดเข้าไป โดยเริ่มเก็บข้อมูลในเวลาเดียวกัน เครื่อง SP4200 จะเก็บข้อมูลโดยอัตโนมัติทันทีที่เริ่มฉีดสารทุกครั้งเพราะมีสวิทช์ต่ออยู่กับวาล์วฉีดสาร ในขณะที่ DAS001 ใช้การกดคีย์บอร์ดของเครื่องคอมพิวเตอร์เป็นการเริ่มต้นการเก็บข้อมูล ซึ่งได้ผลการทดลองดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่ารีเทนชันไทม์ของคาเฟอีน(นาที)		
ครั้งที่	DAS001	SP4200
1	1.71	1.73
2	1.71	1.73
3	1.70	1.73
4	1.71	1.73
5	1.71	1.73
6	1.72	1.73
7	1.71	1.73
8	1.71	1.73
9	1.71	1.73
10	1.71	1.73

ตารางที่ 6.1 แสดงค่ารีเทนชันไทม์ของคาเฟอีนที่ได้จาก DAS001 และ SP4200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 หาค่าเรสปอนด์แฟกเตอร์เฉลี่ยของคาเฟอีน

จากการทดลองหาค่าเรสปอนด์แฟกเตอร์ ของคาเฟอีนมาตรฐานที่ความเข้มข้นต่าง ๆ คือ 100PPM 150PPM และ 200PPM ความเข้มข้นละ 10 ครั้ง ระหว่าง DAS001 กับ SP4200 ซึ่งได้ผลการทดลองดังตารางต่อไปนี้

ครั้งที่	DAS001		SP4200	
	Area	RF	Area	RF
1	21859	218.59	41250	412.50
2	21783	217.83	41185	411.85
3	21883	218.83	41218	412.18
4	21800	218.00	41078	410.78
5	21817	218.17	41030	410.30
6	21797	217.97	41063	410.63
7	21828	218.28	41224	412.24
8	21777	217.77	41312	413.12
9	21808	218.08	41438	414.38
10	21840	218.40	41502	415.02
ค่าเฉลี่ย	21819	218.19	41230	412.30

ตารางที่ 6.2 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 100 PPM ซึ่งได้จาก DAS001 และ SP4200

จากการใช้ค่าเฉลี่ยของพื้นที่พีคเป็นค่าอ้างอิงในการหาค่าความเบี่ยงเบนของพื้นที่พีคสูงสุดที่ความเข้มข้นของคาเฟอีน 100 PPM พบว่า

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ DAS001 มีค่า $\pm$0.3%

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ SP4200 มีค่า $\pm$0.7%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่	DAS001		SP4200	
	Area	RF	Area	RF
1	36134	240.89	67877	452.51
2	36283	241.89	68418	456.12
3	36480	243.20	68921	459.47
4	36407	242.71	68791	458.61
5	36454	243.03	68619	457.46
6	36194	241.29	68509	456.73
7	36364	242.43	68822	458.81
8	36425	242.83	69220	461.47
9	36299	241.99	68413	456.09
10	36330	242.20	68604	457.36
ค่าเฉลี่ย	36337	242.25	68619	457.46

ตารางที่ 6.3 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 150 PPM ซึ่งได้จาก DAS001 และ SP4200

จากการใช้ค่าเฉลี่ยของพื้นที่พีคเป็นค่าอ้างอิงในการหาค่าความเบี่ยงเบนของพื้นที่พีคสูงสุดที่ความเข้มข้นของคาเฟอีน 150 PPM พบว่า

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ DAS001 มีค่า $\pm$0.6%

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ SP4200 มีค่า $\pm$1.1%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่	DAS001		SP4200	
	Area	RF	Area	RF
1	48535	242.68	90926	454.63
2	48452	242.26	90806	454.03
3	48437	242.19	90619	453.10
4	48511	242.56	90755	453.78
5	48575	242.88	90872	454.36
6	48499	242.50	90795	453.98
7	48348	241.74	90608	453.04
8	48481	242.41	90842	454.21
9	48376	241.88	90538	452.69
10	48524	242.62	91061	455.31
ค่าเฉลี่ย	48476	242.37	90782	453.91

ตารางที่ 6.4 แสดงค่า Area และ RF ของคาเฟอีนที่ความเข้มข้น 200 PPM ซึ่งได้จาก DAS001 และ SP4200

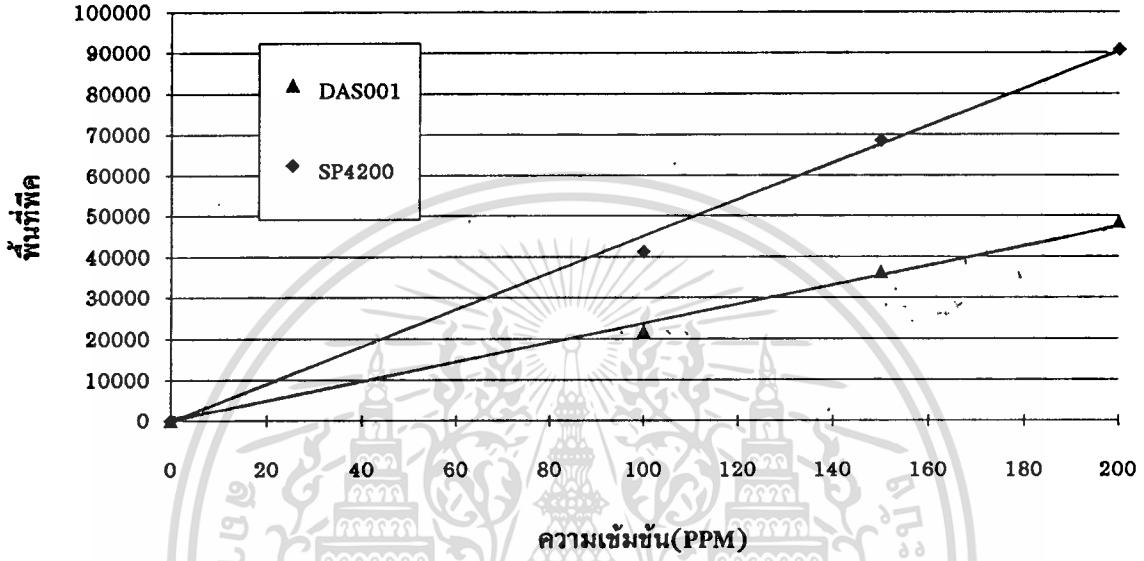
จากการใช้ค่าเฉลี่ยของพื้นที่พีคเป็นค่าอ้างอิงในการหาค่าความเบี่ยงเบนของพื้นที่พีคสูงสุดที่ความเข้มข้นของคาเฟอีน 200 PPM พบว่า

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ DAS001 มีค่า $\pm$0.3%

ค่าความเบี่ยงเบนสูงสุดของพื้นที่พีคของ SP4200 มีค่า $\pm$0.3%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อมูลที่ได้ สามารถเขียนกราฟแสดงความสัมพันธ์ระหว่างพื้นที่พีค(Area) กับความเข้มข้น (Conc.) เป็น PPM ซึ่งจะได้กราฟเส้นตรงโดยค่าความชันเป็นค่า RF ดังรูป



รูปที่ 6.1 แสดงกราฟความสัมพันธ์ระหว่างพื้นที่พีค กับความเข้มข้นของคาเฟอีน ซึ่งได้จาก DAS001 และ SP4200

จากกราฟจะได้ค่า RF เฉลี่ยของ DAS001 เป็น 242.81 และ SP4200 เป็น 457.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 หาค่าความเข้มข้นของคาเฟอีน ในเครื่องดื่มโคลา 2 ชนิด

ได้ทดลองให้โปรแกรม หาค่าความเข้มข้นของคาเฟอีน ในเครื่องดื่มโคลา 2 ชนิด โดยใช้วิธี External Std. เปรียบเทียบกับการใช้เครื่องประมวลผล SP4200 ซึ่งมีผลการทดลองดังนี้

	DAS001	SP4200
Cola A		
Rt(Minutes)	1.71	1.73
Area	16482	31638
RF	242.81	457.20
Conc.(PPM)	68	69
Cola B		
Rt(Minutes)	1.71	1.73
Area	22891	42411
RF	242.81	457.20
Conc.(PPM)	92	93

ตารางที่ 6.5 แสดงค่าความเข้มข้นของคาเฟอีนในเครื่องดื่มโคลา 2 ชนิด โดยใช้ DAS001 และ SP4200

จากการหาค่าความเข้มข้นของคาเฟอีนจากเครื่องดื่มโคลา 2 ชนิดพบว่า ค่าความเข้มข้นของเครื่องดื่มโคลา A ที่ได้จาก DAS001 แตกต่างจาก SP4200 1.5% และ ค่าความเข้มข้นของเครื่องดื่มโคลา B ที่ได้จาก DAS001 แตกต่างจาก SP4200 1.1%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 สรุปผลการทดสอบ

การทดสอบการวิเคราะห์สารคาเฟอีน ด้วยชุดเก็บและวิเคราะห์ผลสำหรับ HPLC (DAS001) และเครื่องประมวลผล SP4200 จำนวน 10 ครั้ง สามารถสรุปได้ดังนี้

6.4.1 ในการหาค่ารีเทนชันไทม์ของคาเฟอีนด้วย DAS001 กับ SP4200 ผลที่ได้พบว่ามีค่าแตกต่างกันประมาณ 1.2 วินาที โดย DAS001 จะมีค่ารีเทนชันไทม์ประมาณ 1.71 นาที และ SP4200 จะมีค่ารีเทนชันไทม์ประมาณ 1.73 นาที ซึ่งสาเหตุที่ DAS001 มีค่ารีเทนชันไทม์น้อยกว่าเพราะโปรแกรมคอมพิวเตอร์จะใช้เวลาเตรียมเก็บข้อมูลและแสดงผลประมาณ 1-2 วินาที ก่อนการเก็บข้อมูลจริงและค่ารีเทนชันไทม์ที่ได้อาจมีค่าเปลี่ยนแปลงไปบ้างเล็กน้อยอันเนื่องมาจากจังหวะการกดคีย์บอร์ดเริ่มทำงานที่ช้าหรือเร็วไป ซึ่งสามารถแก้ไขได้โดยติดสวิทช์ทริกเกอร์เข้ากับวาล์วฉีดสารเพื่อกระตุ้นให้เครื่อง DAS001 เก็บข้อมูลทันทีเมื่อมีการฉีดสารตัวอย่างเข้าไปในคอลัมน์ สำหรับปัญหาเรื่องโปรแกรมคอมพิวเตอร์ใช้เวลาเตรียมเก็บข้อมูล 1-2 วินาทีนั้น สามารถแก้ไขได้เมื่อติดตั้งสวิทช์ทริกเกอร์แล้ว โดยสั่งให้โปรแกรมเริ่มทำงานก่อนที่จะฉีดสารตัวอย่าง โปรแกรมจะคอยสัญญาณจากสวิทช์ทริกเกอร์ โดยจะเก็บข้อมูลทันทีที่ได้รับสัญญาณจากสวิทช์ทริกเกอร์ทำให้สามารถทำงานได้อย่างถูกต้องเช่นเดียวกับเครื่อง SP4200

6.4.2 การหาค่าเรสปอนด์แฟกเตอร์เฉลี่ยของคาเฟอีนด้วย DAS001 และ SP4200 พบว่ามีค่าแตกต่างกันประมาณ 1.88 เท่า อันเนื่องมาจากค่าพื้นที่พีคซึ่งมีความแตกต่างกันประมาณ 1.88 เท่า ทั้งนี้อาจเป็นเพราะเครื่อง SP4200 มีการขยายสัญญาณไฟฟ้าเพิ่มขึ้นก่อนที่จะผ่านเข้าสู่วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลแตกต่างไปจาก DAS001 อย่างไรก็ตามในการคำนวณหาค่าความเข้มข้นจากเครื่องทั้งสองก็ยังมีค่าที่ใกล้เคียงกันมาก โดยแตกต่างกันน้อยกว่า 2%

6.4.3 ในทางปฏิบัติ ค่าเรสปอนด์แฟกเตอร์ของสารเดียวกันอาจมีค่าแตกต่างกันได้ตามความเข้มข้นของสาร กล่าวคือค่าความสัมพันธ์ระหว่างพื้นที่พีคกับความเข้มข้นของสารอาจไม่เป็นเชิงเส้นตลอดทุกค่าความเข้มข้น ดังนั้นจึงควรหาลักษณะของกราฟความสัมพันธ์ระหว่างพื้นที่พีคกับความเข้มข้นในช่วงที่ต้องการตรวจวัด เพื่อใช้ในการเปรียบเทียบและคำนวณหาค่าความเข้มข้นที่ถูกต้องใกล้เคียงความเป็นจริงมากที่สุด

6.4.4 ค่าความเบี่ยงเบนของพื้นที่พีคจากการวัดโดยใช้ DAS001 จะน้อยกว่าหรือเท่ากับค่าความเบี่ยงเบนของพื้นที่พีคโดยใช้ SP4200 แสดงว่าชุดเก็บและวิเคราะห์ผลข้อมูลที่พัฒนาขึ้นมานี้ มีความสามารถในการวิเคราะห์ผลข้อมูลได้เที่ยงตรงและแม่นยำดีกว่าหรือใกล้เคียงกับ SP4200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการวิจัยและเสนอแนะ

เทคโนโลยีด้านคอมพิวเตอร์ซอฟต์แวร์และฮาร์ดแวร์ในปัจจุบัน มีส่วนช่วยเพิ่มประสิทธิภาพของเครื่องมือต่างๆ เป็นอย่างมาก การสร้างชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC ก็เป็นตัวอย่างหนึ่งซึ่งใช้เทคโนโลยีทางด้านคอมพิวเตอร์เข้ามาช่วยในการวิเคราะห์ข้อมูลจากสัญญาณไฟฟ้า ของตัวตรวจวัดของเครื่อง HPLC ซึ่งแต่เดิมจะใช้ตัว Y-t Recorder เป็นตัวช่วยในการแสดงผลสัญญาณของ HPLC แล้วจึงนำกราฟที่ได้ไปวิเคราะห์โดยการหาพื้นที่พีคจากกราฟ ทำให้ต้องเสียเวลาในการวิเคราะห์มาก หรืออาจใช้การวัดความสูงของพีคซึ่งเร็วกว่า แต่ก็จะมีความผิดพลาดมากกว่าการคำนวณจากพื้นที่พีค การนำเอาคอมพิวเตอร์เข้ามาช่วยในการวิเคราะห์ข้อมูลทำให้สามารถวิเคราะห์ข้อมูลได้อย่างรวดเร็วและถูกต้องมากกว่าเดิม และยังสามารถพัฒนาเพื่อเพิ่มขีดความสามารถในการวิเคราะห์ข้อมูลได้ง่ายอีกด้วย

ในกรณีของชุดเก็บข้อมูลและวิเคราะห์ผลสำหรับ HPLC นี้ เป็นเครื่องมือที่มีการพัฒนาจนกลายเป็นเครื่องต้นแบบที่ใช้งานได้เท่านั้น แต่ถ้าต้องนำไปใช้งานในเชิงพาณิชย์ จำเป็นต้องมีการพัฒนาขีดความสามารถของฮาร์ดแวร์และซอฟต์แวร์เพิ่มเติมอีก สำหรับข้อเสนอแนะในการพัฒนาเพิ่มเติมสามารถแบ่งได้เป็น 2 ส่วน คือด้านฮาร์ดแวร์ และด้านซอฟต์แวร์ ทางด้านฮาร์ดแวร์นั้นควรมีการปรับปรุงด้านความเร็วของการรับส่งข้อมูล โดยให้สามารถปรับอัตราการรับส่งข้อมูลระหว่าง DAS001 กับเครื่องไมโครคอมพิวเตอร์ เช่น การตั้งค่า Baud Rate เป็น 2400 4800 9600 บิทต่อวินาที เพื่อป้องกันปัญหาเรื่องความผิดพลาดในการรับส่งข้อมูลของเครื่องไมโครคอมพิวเตอร์กับ DAS001 เพราะมีเครื่องไมโครคอมพิวเตอร์บางเครื่องที่ยังมีอัตราการรับส่งข้อมูลทางพอร์ทอนุกรมไม่สูงพอ แต่สามารถแก้ไขได้โดยการลดความเร็วในการรับส่งข้อมูลให้เหมาะสม นอกจากนี้ควรมีวงจรป้องกันภาครับส่งข้อมูลทางสาย RS232C โดยใช้การเชื่อมโยงสัญญาณทางแสงเพื่อแยกระบบไฟฟ้าระหว่าง DAS001 กับเครื่องไมโครคอมพิวเตอร์ออกจากกัน ทั้งนี้เพื่อป้องกันไม่ให้มีไฟฟ้ารั่วจากเครื่องไมโครคอมพิวเตอร์มาทำให้วงจรของ DAS001 เกิดความเสียหายได้

ส่วนทางด้านซอฟต์แวร์ควรมีการปรับปรุงเพื่อเพิ่มขีดความสามารถในการวิเคราะห์พีคที่มีความซับซ้อนมากขึ้น เช่นใช้วิเคราะห์พีคที่ซ้อนกัน ใช้กับกรณีที่เส้นระดับฐานมีการเลื่อนระดับไป ใช้ในการหาความเข้มข้นสารที่มีความสัมพันธ์ระหว่างพื้นที่พีคกับความเข้มข้นของสารที่ไม่เป็นเชิงเส้น หรือการเพิ่มวิธีการวิเคราะห์แบบต่าง ๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Douglas A.Skoog,Principles of Instrumental Analysis,Saunders College Publishing,Orando,Florida,(1985)

[2] Peter A. Sewell & Brain Charke,Chromatographic Separations,John Wiley & Son,Singapore,New York,(1991)

[3] Sandy Lindsay,High Performance Liquid Chromatography,John Wiley & Son Singapore,New York,(1991)

[4] Skoog,D.A.,and West,D.M.,Fundamental Analytical Chemistry,Holt,Renehert and Winston,New York,1969

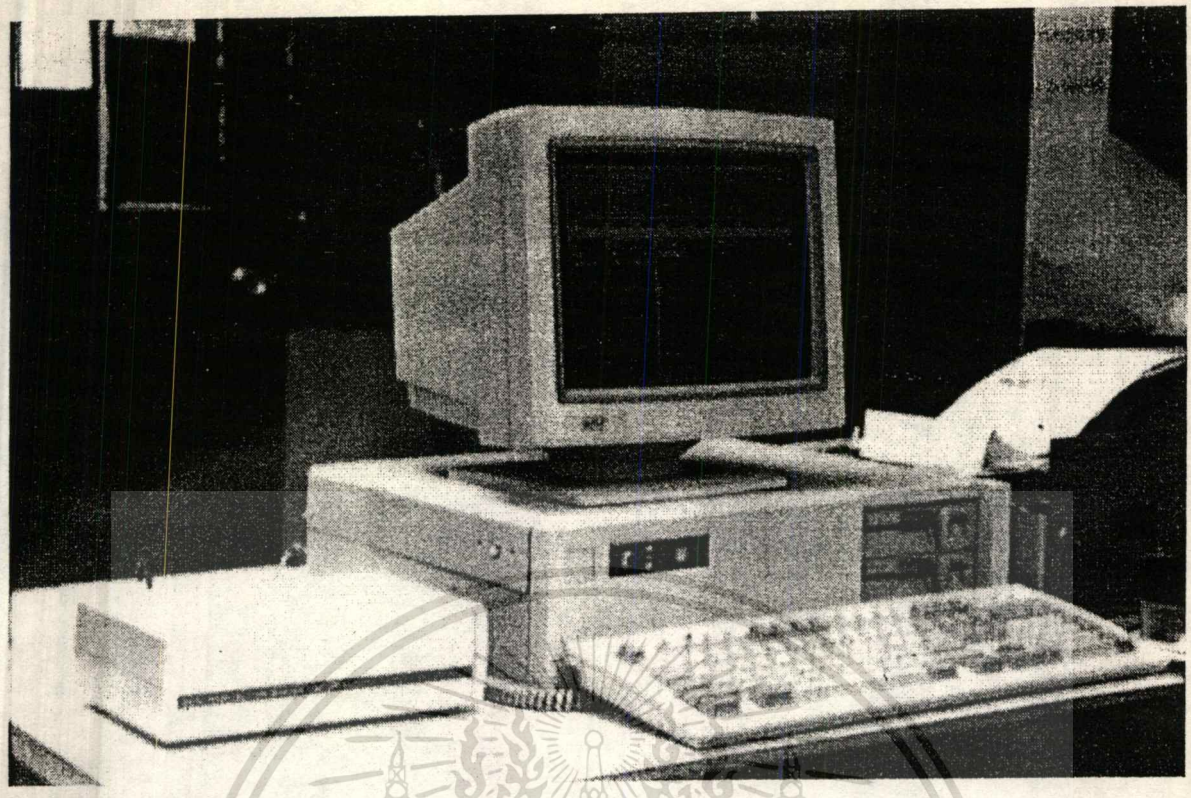
[5] David E.Goldorg and Clyde R. Dilland,College Chemistry, Macmillan Publishing Co.,Inc,New York,1974

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

เครื่องเก็บและวิเคราะห์ผลข้อมูลที่สร้างขึ้นและเครื่องวิเคราะห์ผลข้อมูลจากต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ชุดเก็บและวิเคราะห์ผลข้อมูลสำหรับ HPLC ที่ได้พัฒนาขึ้น

เครื่องวิเคราะห์ผลข้อมูลของต่างประเทศ(SP4200)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งและทดสอบการใช้งานระหว่างชุดเก็บและวิเคราะห์ข้อมูลที่พัฒนาขึ้น
กับเครื่องวิเคราะห์ผลข้อมูล SP4200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ซอฟต์แวร์วิเคราะห์ข้อมูลสำหรับเครื่อง HPLC (ใช้กับเครื่องไมโครคอมพิวเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM HPLC_PAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1:PROGRAM MainMenu;
 2:Uses Crt,Dos,Graph,Utils,Part1,GraphPrn,Printer;
 3:
 4:Var  ActiveMode   : byte;
 5:
 6:Const { ActiveMode }
 7:  LabNetMode = 1;
 8:  FileMode   = 2;
 9:
10:Var Filter : byte;
11:
12:Const { FilterMode }
13:  FilterOff  = 0;
14:  FilterOn   = 1;
15:
16:Var  ActiveMethod : byte;
17:
18:Const { ActiveMethod }
19:  Method_A   = 1;     {    Normal     }
20:  Method_B   = 2;     { External Std. }
21:  Method_C   = 3;     { BaseLine Cal. }
22:  Method_D   = 4;     {    Reserve    }
23:
24:Var  SerialPort : byte;
25:
26:Const { SerialPort }
27:  Com1  = 0;
28:  Com2  = 1;
29:
30:var PeakMarker : byte;
31:
32:Const
33:  MarkerOff  = 0;
34:  MarkerOn   = 1;
35:
36:Var WorkTime     : word;      { Minutes }
37:    AttenLevel   : word;
38:    PeakWidth    : byte;      { DataSample/Bunch }
39:    Threshold    : LongInt;
40:    PageRange    : byte;      { Minutes }
41:    BaseLine     : LongInt;
42:    BottomLimit  : real;
43:
44:var i            : integer;
45:    ChangeFname  : boolean;
46:
47:Procedure DisplayMenu;
48:begin
49:  DrawMainWindow;
50:  DrawSubWindow;
51:  DrawLOGO;
52:  DrawCommandWindow;
53:  For i := ModeMenu_ to QuitMenu_ do
54:    DrawCommandMenu(i,ShowOn);
55:  DrawCommandMenu(ModeMenu_,SelectOn);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  NewActiveMenu := 1;
  OldActiveMenu := 1;
  DrawCommandMenu(NewActiveMenu,SelectOn);
  WriteCurrentDate;
  DrawTimeWindow;
  WriteCurrentTime;
  DrawStatusWindow;
  DrawInfoWindow;
  WriteCurrentDirectory;
end;

Procedure InputFileName(x,y : word; var FileName : string);
var  SavWin   : ScreenWindow;
     St       : String;
begin
  SetScreenWindow(x,y-4,x+380,y+45,SavWin);
  PushScreenWindow(SavWin);
  DrawWin(x,y,x+380,y+45,ThickType,Yellow,Red,ShadowOff,
       Black,'',ReverseType);
  SetBkTextColor(Red);
  St := 'Enter File Name ['+FileName+']';
  PutTextXY(x+10,y+15,St,NormalType,White);
  MoveTo(x+10,y+28);
  St := '';
  ReadGraphString(St);
  if St <> '' then
  begin
    FileName := St;
    ChangeFname := True;
  end;
  SetBkTextColor(Black);
  PopScreenWindow(SavWin);
end;

Const  { Process Mode }
  PcShow   = 0;
  PcActive = 1;
  PcEdit   = 2;

Const  { Choice Type }
  NumberType   = 0;
  StringType   = 1;
  OptionTypeA  = 2;
  OptionTypeB  = 3;

Type
  String100  = String[100];
  String50   = String[50];

Type
  SubMenu    = ^MenuChoice;
  MenuChoice = record
                 x,y    : word;
               Name     : String30;
               Lib      : String100;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
111:                Cidx    : byte;
112:                Ctype   : byte;
113:                VaLib   : String50;
114:                SvIdx   : byte;
115:                Fwd     : SubMenu;
116:                Bwd     : SubMenu;
117:                end;
118:
119:Var
120:  TempChoice : SubMenu;
121:
122: { ---------------- ModeMenu ---------------- }
123:  ModeChoice : SubMenu;
124:
125: { ---------------- OptionMenu ---------------- }
126:  RunTimeChoice       : SubMenu;
127:  AttenChoice         : SubMenu;
128:  PageRangeChoice     : SubMenu;
129:  PeakWidthChoice     : SubMenu;
130:  ThresholdChoice     : SubMenu;
131:  SerialPortChoice    : SubMenu;
132:  PeakWindowChoice    : SubMenu;
133:  PeakStatusChoice    : SubMenu;
134:  PeakMarkerChoice    : SubMenu;
135:  FilterChoice        : SubMenu;
136:  BaseLineChoice      : SubMenu;
137:
138: { ---------------- MethodMenu ---------------- }
139:  MethodChoice        : SubMenu;
140:
141: { ---------------- EnvMenu ---------------- }
142:  DataFileChoice      : SubMenu;
143:  DataDirChoice       : SubMenu;
144:
145:var
146:  Ddir : string;
147:
148:Procedure SetDataDirectory;
149:begin
150:  GetDir(0,Ddir);
151:end;
152:
153:Function GetMaxChoice(SbMenu:SubMenu) : byte;
154:var i    : integer;
155:    Cmax : integer;
156:begin
157:  Cmax  := 1;
158:  for i := 1 to Length(SbMenu^.Lib) do
159:    if SbMenu^.Lib[i] = ',' then inc(Cmax);
160:  GetMaxChoice := Cmax;
161:end;
162:
163:Function GetChoice(SbMenu:SubMenu) : string;
164:var STem  : string;
165:    Epos  : byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
166:    Idx   : byte;
167:begin
168:  with SbMenu^ do
169:  Case CType Of
170:    NumberType : GetChoice := Lib;
171:    StringType : GetChoice := Lib;
172:    OptionTypeA,OptionTypeB
173:           : begin
174:                STem := Lib;
175:                Idx  := 1;
176:                While Idx <> Cidx do
177:                begin
178:                  Epos := Pos(',',STem);
179:                  Delete(STem,1,Epos);
180:                  Inc(Idx);
181:                end;
182:                  Epos := Pos(',',STem);
183:                  if Epos = 0 then GetChoice := STem
184:                              else GetChoice := Copy(STem,1,Epos-1);
185:             end;
186:  end; { Case }
187:end;
188:
189:Function Choice(Wx,Wy:word; SbMenu:SubMenu) : byte;
190:Const Yspace  = 5;
191:var Ex,Ey,Tx,Ty : word;
192:    Count,i    : byte;
193:    Smax,Snum  : byte;
194:    Xmsg       : string;
195:    SavLib     : string;
196:    SavWin     : ScreenWindow;
197:begin
198:  With SbMenu^ do
199:  begin
200:    SavLib := Lib;
201:    Count := 1;
202:    Smax := 0;
203:    Snum  := 0;
204:
205:  for i := 1 to Length(Lib) do
206:    if SavLib[i] = ',' then
207:    begin
208:      inc(Count);
209:      Snum := 0;
210:    end else
211:    begin
212:      inc(Snum);
213:      if Snum > Smax then Smax := Snum;
214:    end;
215:
216:  if Smax >= 12 then Ex := Wx + TextWidth('X')*(Smax+4)
217:                else Ex := Wx + TextWidth('X')*16;
218:  Ey := Wy + (TextHeight('X')+Yspace)*(Count+1) + 12;
219:
220:  SetScreenWindow(Wx,Wy-4,Ex,Ey,SavWin);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
221:  PushScreenWindow(SavWin);
222:
223:  DrawWin(Wx,Wy,Ex,Ey,DoubleType,LightRed,Black,
224:          ShadowOff,Black,Name,ReverseType);
225:  SetColor(Yellow);
226:
227:  Tx := Wx + TextWidth('X')*2 + 1;
228:  Ty := Wy + 6;
229:
230:  for i := 1 to Count do
231:  begin
232:    if i = Count then Xmsg := Copy(SavLib,1,Smax)
233:                 else begin
234:                     Xmsg := Copy(SavLib,1,Pos(',',SavLib)-1);
235:                     Delete(SavLib,1,Pos(',',SavLib));
236:                      end;
237:    Ty := Ty + TextHeight('X') + Yspace;
238:    PutTextXY(Tx,Ty,Xmsg,NormalType,GetColor);
239:  end;
240:  Ty := Wy+6+Cidx*(TextHeight('X')+Yspace);
241:  Xmsg := GetChoice(SbMenu);
242:  SetFillStyle(SolidFill,GetColor);
243:  Bar(Wx+7,Ty-5,Ex-7,Ty+5);
244:  PutTextXY(Tx,Ty,Xmsg,ReverseType,GetColor);
245:  Repeat
246:  Repeat
247:    WriteCurrentTime;
248:  Until KeyPressed;
249:  Key := ReadKey;
250:  Case Key of
251:    Up_Key : begin
252:                Xmsg := GetChoice(SbMenu);
253:             SetFillStyle(SolidFill,GetBkColor);
254:             Bar(Wx+7,Ty-5,Ex-7,Ty+5);
255:             PutTextXY(Tx,Ty,Xmsg,NormalType,GetColor);
256:             Dec(Cidx);
257:             if Cidx < 1 then Cidx := Count;
258:             Ty := Wy+6+Cidx*(TextHeight('X')+Yspace);
259:             Xmsg := GetChoice(SbMenu);
260:             SetFillStyle(SolidFill,GetColor);
261:             Bar(Wx+7,Ty-5,Ex-7,Ty+5);
262:             PutTextXY(Tx,Ty,Xmsg,ReverseType,GetColor);
263:              end;
264:    Dn_Key : begin
265:                Xmsg := GetChoice(SbMenu);
266:                SetFillStyle(SolidFill,GetBkColor);
267:             Bar(Wx+7,Ty-5,Ex-7,Ty+5);
268:             PutTextXY(Tx,Ty,Xmsg,NormalType,GetColor);
269:             Inc(Cidx);
270:             if Cidx > Count then Cidx := 1;
271:             Ty := Wy+6+Cidx*(TextHeight('X')+Yspace);
272:             Xmsg := GetChoice(SbMenu);
273:                SetFillStyle(SolidFill,GetColor);
274:             Bar(Wx+7,Ty-5,Ex-7,Ty+5);
275:             PutTextXY(Tx,Ty,Xmsg,ReverseType,GetColor);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
276:              end;
277:  end;
278:
279:  Until (Key = Return_Key) or (Key = Esc_Key);
280:  Choice := Cidx;
281:  PopScreenWindow(SavWin);
282:  end; { with }
283:end;
284:
285:Function GetStringValue(St : string; var Code : Integer) : LongInt;
286:var SavVal : LongInt;
287:begin
288:  Val(St,SavVal,Code);
289:  GetStringValue := SavVal;
290:end;
291:
292:Procedure EditChoice(var SbMenu:SubMenu);
293:label Again;
294:var STem : String;
295:    Epos : byte;
296:    Idx  : byte;
297:    Code : integer;
298:    Num  : LongInt;
299:begin
300:  with SbMenu^ do
301:  begin
302:    Case CType Of
303:      NumberType : begin
304:                     STem := Lib;
305:                     Again:MoveTo(x+TextWidth('X')*Length(Name),y-3);
306:                     EditGraphString(STem);
307:                     Val(Stem,Num,Code);
308:                       if (Code = 0) and (Num >= 0) then Lib := STem else
309:                     begin
310:                       Sound(3200);Delay(80);NoSound;
311:                       ShowMsg(290,220,'Positive Integer only !,'+
312:                                       '     Try again ...','Error');
313:                       Goto Again;
314:                     end;
315:                   end;
316:     StringType : begin
317:                       STem := Lib;
318:                     MoveTo(x+TextWidth('X')*Length(Name),y-3);
319:                     EditGraphString(STem);
320:                     Lib := STem;
321:                   end;
322:    OptionTypeA : begin
323:                       STem := Lib;
324:                     Idx  := 1;
325:                     While Idx <> Cidx do
326:                     begin
327:                       Epos := Pos(',',STem);
328:                       Delete(STem,1,Epos);
329:                       Inc(Idx);
330:                     end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
331:                    Epos := Pos(',',STem);
332:                    if Epos = 0 then
333:                    begin
334:                      STem := Copy(Lib,1,Pos(',',Lib)-1);
335:                      Cidx := 1;
336:                    end else
337:                    begin
338:                      Inc(Cidx);
339:                      STem := GetChoice(SbMenu);
340:                    end;
341:                    end;
342:
343:     OptionTypeB : begin
344:                    Cidx := Choice(400,50,SbMenu);
345:                  end;
346:
347:    end; { Case }
348:  end;
349:end;
350:
351:Procedure InitMenuChoice(LocX,LocY:word; CName:String30; CLib:string;
352:                        ChoiceType :byte; var SbMenu : SubMenu);
353:begin
354:  New(SbMenu);
355:  With SbMenu^ do
356:  begin
357:    x      := LocX;
358:    y      := LocY;
359:    Name   := CName;
360:    Lib    := Clib;
361:    Cidx   := 1;
362:    SvIdx  := 1;
363:    Ctype  := ChoiceType;
364:    Fwd    := nil;
365:    Bwd    := nil;
366:  end;
367:end;
368:
369:Procedure InitValueChoice(ValLib : String;var SbMenu : SubMenu);
370:begin
371:  With SbMenu^ do
372:  begin
373:    VaLib := ValLib;
374:  end;
375:end;
376:
377:Function GetValueChoice(SbMenu : SubMenu) : LongInt;
378:var SavVal : LongInt;
379:    Code   : integer;
380:    STem   : string;
381:    Epos   : byte;
382:    Idx    : byte;
383:begin
384:  With SbMenu^ do
385:  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
386:   Case Ctype Of
387:     NumberType : Val(Lib,SavVal,Code);
388:     StringType : SavVal := 0;
389:     OptionTypeA,OptionTypeB
390:            : begin
391:                 STem := VaLib;
392:                 Idx  := 1;
393:                 While Idx <> Cidx do
394:                 begin
395:                   Epos := Pos(',',STem);
396:                   Delete(STem,1,Epos);
397:                   Inc(Idx);
398:                 end;
399:                 Epos := Pos(',',STem);
400:                 if Epos = 0 then Val(STem,SavVal,Code)
401:                                  else begin
402:                                         STem := Copy(STem,1,Epos-1);
403:                                         Val(STem,SavVal,Code);
404:                                       end;
405:              end;
406:    end; { Case }
407:    GetValueChoice := SavVal;
408:  end; { With }
409:end;
410:
411:Procedure LinkSubMenu(var SubA,SubB : SubMenu);
412:begin
413:  SubA^.Fwd  := SubB;
414:  SubB^.Bwd  := SubA;
415:end;
416:
417:Procedure ProcessMenuChoice(Pmode : byte; var SbMenu : SubMenu);
418:begin
419:  SetBkTextColor(Blue);
420:  With SbMenu^ do
421:  Case Pmode of
422:  PcShow :
423:     begin
424:       PutTextXY(x,y,Name,NormalType,GetColor);
425:       PutTextXY(x+TextWidth('X')*Length(Name),y,GetChoice(SbMenu),
426:       NormalType,GetColor);
427:     end;
428:
429:  PcActive :
430:     begin
431:          PutTextXY(x,y,Name,NormalType,GetColor);
432:       PutTextXY(x+TextWidth('X')*Length(Name),y,GetChoice(SbMenu),
433:       ReverseType,GetColor);
434:     end;
435:
436:  PcEdit :
437:     begin
438:          SetBkTextColor(Blue);
439:          PutTextXY(x+TextWidth('X')*Length(Name),y,
440:       GetChoice(SbMenu),ReverseType,BkTextColor);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
441:        SetColor(Yellow);
442:        EditChoice(SbMenu);
443:           PutTextXY(x+TextWidth('X')*Length(Name),y,
444:        GetChoice(SbMenu),ReverseType,GetColor);
445:        SetBkTextColor(Black);
446:      end;
447:
448:  end; { Case }
449:  SetBkTextColor(Black);
450:end;
451:
452:Procedure SettingMenuChoice;
453:begin
454:  New(TempChoice);
455:
456:  InitMenuChoice(310,150,' Mode ',
457:                 '    LabNet     ,     File     ',
458:                OptionTypeB,ModeChoice);
459:
460:  InitMenuChoice(275,105, 'RunTime      : ',
461:                 '10 min.,15 min.,20 min.,30 min.,45 min.,'+
462:                 '60 min.,90 min.,5 min.',OptionTypeA,RuntimeChoice);
463:
464:  InitMenuChoice(275,125, 'Atten        : ',
465:                 '10,20,50,100,200,500,'+
466:                 '1000,1,2,5',OptionTypeA,AttenChoice);
467:
468:  InitMenuChoice(275,145,'PageRange     : ',
469:                 '4 min.,6 min.,8 min.,10 min.,15 min.,20 min.,'+
470:                 '30 min.,1 min.,2 min.',OptionTypeA,PageRangeChoice);
471:
472:  InitMenuChoice(275,165,'PeakWidth     : ',
473:                 '1 sec.,2 sec.,4 sec.,6 sec.,8 sec.,'+
474:                 '10 sec.,12 sec.',OptionTypeA,PeakWidthChoice);
475:
476:  InitMenuChoice(275,185,'Threshold     : ',
477:                 '400',NumberType,ThresholdChoice);
478:
479:  InitMenuChoice(275,205,'SerialPort    : ',
480:                 'Com1,Com2',OptionTypeA,SerialPortChoice);
481:
482:  InitMenuChoice(275,225,'PeakWindow    : ',
483:                 '10.0%,12.0%,15.0%,20.0%,4.0%,6.0%,8.0%',
484:                  OptionTypeA,PeakWindowChoice);
485:
486:  InitMenuChoice(275,245,'PeakMarker    : ',
487:                 'ON,OFF',OptionTypeA,PeakMarkerChoice);
488:
489:  InitMenuChoice(275,265,'Filter        : ',
490:                 'OFF,ON',OptionTypeA,FilterChoice);
491:
492:  InitMenuChoice(275,285,'BaseLine      : ',
493:                 '1000',NumberType,BaseLineChoice);
494:
495:  InitMenuChoice(275,150,' Method ',
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
496:                '     Normal      , External Std.  ,'+
497:                '  BaseLine Cal.  ,     Reserve     ',
498:               OptionTypeB,MethodChoice);
499:
500:  InitMenuChoice(260,165,' DataFile : ',
501:               'TEST.DAT',StringType,DataFileChoice);
502:
503:  InitMenuChoice(260,185,' DataDir  : ',
504:               Ddir,StringType,DataDirChoice);
505:
506:end;
507:
508:Procedure SettingValueChoice;
509:begin
510:  InitValueChoice('1,2',ModeChoice);
511:  InitValueChoice('10,15,20,30,45,60,90,5',RunTimeChoice);
512:  InitValueChoice('10,20,50,100,200,500,1000,1,2,5',AttenChoice);
513:  InitValueChoice('4,5,8,10,15,20,30,1,2',PageRangeChoice);
514:  InitValueChoice('1,2,4,6,8,10,12',PeakWidthChoice);
515:  InitValueChoice('0,1',SerialPortChoice);
516:  InitValueChoice('10,12,15,20,4,6,8',PeakWindowChoice);
517:  InitValueChoice('1,0',PeakMarkerChoice);
518:  InitValueChoice('0,1',FilterChoice);
519:end;
520:
521:Procedure ProcessLinkSubMenu;
522:begin
523:  { ***************** Option Menu ****************** }
524:
525:  LinkSubMenu(RuntimeChoice,AttenChoice);
526:  LinkSubMenu(AttenChoice,PageRangeChoice);
527:  LinkSubMenu(PageRangeChoice,PeakWidthChoice);
528:  LinkSubMenu(PeakWidthChoice,ThresholdChoice);
529:  LinkSubMenu(ThresholdChoice,SerialPortChoice);
530:  LinkSubMenu(SerialPortChoice,PeakWindowChoice);
531:  LinkSubMenu(PeakWindowChoice,PeakMarkerChoice);
532:  LinkSubMenu(PeakMarkerChoice,FilterChoice);
533:  LinkSubMenu(FilterChoice,BaseLineChoice);
534:  LinkSubMenu(BaseLineChoice,RunTimeChoice);
535:
536:  { ******************* File Menu ****************** }
537:
538:  LinkSubMenu(DataFileChoice,DataDirChoice);
539:  LinkSubMenu(DataDirChoice,DataFileChoice);
540:end;
541:
542:Procedure SetMainGraph(var Gdef : GraphDefine);
543:var MaxLevel  : real;
544:     Prange    : real;
545:begin
546:  MaxLevel  := (AttenLevel*1000.0)+BottomLimit;
547:  Prange    := PageRange*60.0;
548:  TimeFinish := TimeStart+Prange;
549:  SetGraphLimit(TimeStart,BottomLimit,TimeFinish,MaxLevel,Gdef);
550:  SetGraphGrid(Prange/20.0,MaxLevel/20.0,Gdef);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
551:end;
552:
553:Procedure SetMapGraph(var Gdef : GraphDefine);
554:var MaxLevel    : real;
555:    RunTimeMax : real;    { Seconds }
556:begin
557:  MaxLevel    := (AttenLevel*1000.0)+BottomLimit;
558:  RunTimeMax := WorkTime*60.0;
559:  SetGraphLimit(0.0,BottomLimit,RunTimeMax,MaxLevel,Gdef);
560:end;
561:
562:Procedure SetTimeGraph(var Gdef : GraphDefine);
563:var RunTimeMax : real;    { Seconds }
564:begin
565:  RunTimeMax := WorkTime*60.0;
566:  SetGraphLimit(0.0,0.0,RunTimeMax,10.0,Gdef);
567:  SetGraphGrid(RuntimeMax/50.0,5.0,Gdef);
568:end;
569:
570:Procedure SetPrintGraph(var Gdef : GraphDefine);
571:var MaxLevel    : real;
572:    RunTimeMax : real;    { Seconds }
573:begin
574:  MaxLevel    := (AttenLevel*1000.0)+BottomLimit;
575:  RunTimeMax := WorkTime*60.0;
576:  SetGraphLimit(0.0,BottomLimit,RunTimeMax,MaxLevel,Gdef);
577:  SetGraphGrid(RuntimeMax/40.0,5.0,Gdef);
578:  SetGraphPlotColor(Yellow,Gdef);
579:end;
580:
581:Procedure WriteStatus(x,y:integer; Name:string30; SbMenu:SubMenu);
582:var st  : string;
583:begin
584:  st  := GetChoice(SbMenu);
585:  while st[1] = ' ' do Delete(st,1,1);
586:  st := Concat(Name,st);
587:  PutTextXY(x,y,st,NormalType,White);
588:end;
589:
590:
591:Procedure ShowMainStatus;
592:begin
593:  SetBktextColor(Blue);
594:  WriteStatus(204,82 ,'Mode       :  ',ModeChoice);
595:  WriteStatus(204,102,'Method     :  ',MethodChoice);
596:  WriteStatus(204,122,'RunTime    :  ',RuntimeChoice);
597:  WriteStatus(204,142,'Atten      :  ',AttenChoice);
598:  WriteStatus(204,162,'Threshold  :  ',ThresholdChoice);
599:  WriteStatus(204,182,'PkWidth    :  ',PeakWidthChoice);
600:  WriteStatus(204,202,'IOPort     :  ',SerialPortChoice);
601:  WriteStatus(204,222,'Filter     :  ',FilterChoice);
602:  WriteStatus(204,242,'PeakMarker :  ',PeakMarkerChoice);
603:  WriteStatus(204,262,'DataFile   :  ',DataFileChoice);
604:  WriteStatus(204,282,'BaseLine   :  ',BaseLineChoice);
605:  SetBkTextColor(Black);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
606:end;
607:
608: Type  Config = record
609:                 CRunTime      : byte;
610:                 CAtten        : byte;
611:                 CPageRange    : byte;
612:                 CPeakWidth    : byte;
613:                   CThreshold    : String50;
614:                 CSerialPort   : byte;
615:                 CPeakWindow   : byte;
616:                 CPeakMarker   : byte;
617:                 CFilter       : byte;
618:                 CBaseLine     : String50;
619:               end;
620:
621:Var   CfgFile : File of Config;
622:     Cfg     : Config;
623:
624:Procedure SaveConfig;
625:const CFname = 'HPLC.CFG';
626:begin
627:  DrawInfoWindow;
628:  SetBkTextColor(Blue);
629:  PutTextXY(204,336,'--- Save Config ---',ReverseType,LightGreen);
630:  PutTextXY(204,351,'     HPLC.CFG      ',NormalType,Yellow);
631:  Assign(CfgFile,CFname);
632:  {$I-} ReWrite(CfgFile) {$I+} ;
633:  if IOresult <> 0 then
634:  begin
635:    Sound(3000);Delay(60);NoSound;
636:    PutTextXY(204,366,'ERROR : Disk Write Error ! ',Normaltype,LightRed);
637:    PutTextXY(204,381,'Press ENTER to Continue',Normaltype,White);
638:    Readln;
639:  end else
640:  begin
641:    With Cfg do
642:    begin
643:      CRunTime      := RunTimeChoice^.Cidx;
644:      CAtten        := AttenChoice^.Cidx;
645:      CPageRange    := PageRangeChoice^.Cidx;
646:      CPeakWidth    := PeakWidthChoice^.Cidx;
647:      CThreshold    := ThresholdChoice^.Lib;
648:      CSerialPort   := SerialPortChoice^.Cidx;
649:      CPeakWindow   := PeakWindowChoice^.Cidx;
650:      CPeakMarker   := PeakMarkerChoice^.Cidx;
651:      CFilter       := FilterChoice^.Cidx;
652:      CBaseLine     := BaseLineChoice^.Lib;
653:    end;
654:    Write(CfgFile,Cfg);
655:    Close(CfgFile);
656:  end;
657:  Delay(500);
658:  SetBkTextColor(Black);
659:  DrawInfoWindow;
660:end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
661:
662:Procedure LoadConfig;
663:Const CFname = 'HPLC.CFG';
664:begin
665:  DrawInfoWindow;
666:  SetBkTextColor(Blue);
667:  PutTextXY(204,336,'--- Load Config ---',ReverseType,LightCyan);
668:  PutTextXY(204,351,'     HPLC.CFG      ',NormalType,Yellow);
669:  Assign(CfgFile,CFname);
670:  {$I-} Reset(CfgFile) {$I+} ;
671:  if IOresult <> 0 then
672:  begin
673:    Sound(3000);Delay(60);NoSound;
674:    PutTextXY(204,366,'ERROR : File not found ! ',NormalType,LightRed);
675:    PutTextXY(204,381,'Press ENTER to Continue',Normaltype,White);
676:    Readln;
677:  end else
678:  begin
679:    Read(CfgFile,Cfg);
680:    With Cfg do
681:    begin
682:      RunTimeChoice^.Cidx      := CRunTime;
683:      AttenChoice^.Cidx        := CAtten ;
684:      PageRangeChoice^.Cidx    := CPageRange;
685:      PeakWidthChoice^.Cidx    := CPeakWidth;
686:      ThresholdChoice^.Lib     := CThreshold;
687:      SerialPortChoice^.Cidx   := CSerialPort;
688:      PeakWindowChoice^.Cidx   := CPeakWindow;
689:      PeakMarkerChoice^.Cidx   := CPeakMarker;
690:      FilterChoice^.Cidx       := CFilter;
691:      BaseLineChoice^.Lib      := CBaseLine;
692:    end;
693:    Close(CfgFile);
694:  end;
695:  Delay(500);
696:  SetBkTextColor(Black);
697:  DrawInfoWindow;
698:end;
699:
700:Procedure GetDiskFreeSpace;
701:var DriveName,st : string;
702:begin
703:  DrawInfoWindow;
704:  GetDir(0,DriveName);
705:  Str(DiskFree(0),st);
706:  Delete(DriveName,4,Length(DriveName)-2);
707:  st := Concat(DriveName,'  ',st,' bytes free');
708:  SetBkTextColor(Blue);
709:  PutTextXY(204,336,st,NormalType,LightMagenta);
710:  SetBkTextColor(Black);
711:end;
712:
713:Procedure GetFreeMemory;
714:type  String30 = string[30];
715:var  st : String30;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
716:begin
717:  DrawInfoWindow;
718:  Str(MemAvail,st);
719:  st := Concat('Memory  :  ',st,' bytes free');
720:  SetBkTextColor(Blue);
721:  PutTextXY(204,336,st,NormalType,LightMagenta);
722:  SetBkTextColor(Black);
723:end;
724:
725:Procedure SetDataFileName(Dname:string);
726:var  Cn   : byte;
727:begin
728:  if Pos(':',Dname) <> 0 then
729:  begin
730:    if Pos('\',Dname) = 0 then
731:    begin
732:      Ddir := Copy(Dname,1,2)+'\';
733:      Delete(Dname,1,2);
734:    end else
735:    begin
736:      Cn   := 0;
737:      Ddir := '';
738:      Repeat
739:        Cn   := Cn+1;
740:        Ddir := Ddir+Copy(Dname,1,Pos('\',Dname));
741:        Delete(Dname,1,Pos('\',Dname));
742:      Until Pos('\',Dname) = 0;
743:      if Cn > 1 then Delete(Ddir,Length(Ddir),1);
744:    end;
745:  end;
746:  DataFileChoice^.Lib := Dname;
747:  DataDirChoice^.Lib  := Ddir;
748:end;
749:
750:Function GetDatafileName : String;
751:var  Dname : string;
752:begin
753:  Dname := GetChoice(DataFileChoice);
754:  if Ddir[Length(Ddir)] = '\' then GetDataFileName := Ddir+Dname
755:
756:end;
757:
758:Function GetMenu : byte;
759:begin
760:  Repeat
761:  Repeat
762:      WriteCurrentTime;
763:  Until KeyPressed;
764:  Key := ReadKey;
765:  Case Key of
766:
767:      Up_Key  : begin
768:                  OldActiveMenu := NewActiveMenu;
769:                  Dec(NewActiveMenu);
770:                  if NewActiveMenu < 1 then NewActiveMenu := 7;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
771:                    Sound(3200);delay(15);NoSound;
772:                    DrawCommandMenu(OldActiveMenu,SelectOff);
773:                    DrawCommandMenu(NewActiveMenu,SelectOn);
774:                  end;
775:
776:      Dn_Key   : begin
777:                    OldActiveMenu := NewActiveMenu;
778:                    Inc(NewActiveMenu);
779:                    if NewActiveMenu > 7 then NewActiveMenu := 1;
780:                    Sound(3200);delay(15);NoSound;
781:                    DrawCommandMenu(OldActiveMenu,SelectOff);
782:                    DrawCommandMenu(NewActiveMenu,SelectOn);
783:                 end;
784:
785:      F2_Key   : SaveConfig;
786:
787:      F3_Key   : begin
788:                    LoadConfig;
789:                    DrawStatusWindow;
790:                    ShowMainStatus;
791:                 end;
792:
793:   end; { Case }
794:
795:   Until (Key = Return_Key) or (Key = ESC_Key);
796:
797:   If Key = ESC_Key Then
798:   begin
799:     OldActiveMenu := NewActiveMenu;
800:     NewActiveMenu := QuitMenu_;
801:     Sound(3500);delay(50);NoSound;
802:     DrawCommandMenu(OldActiveMenu,SelectOff);
803:     DrawCommandMenu(NewActiveMenu,SelectOn);
804:   end;
805:
806:   GetMenu := NewActiveMenu;
807:
808:end;
809:
810:Type
811:   Rdat      = ^RawData;
812:   RawData   = record
813:                Data : LongInt;
814:                Next : Rdat;
815:              end;
816:
817:   Cdat      = ^CalData;
818:   CalData   = record
819:                  Data : LongInt;
820:                Diff : LongInt;
821:                Idex : word;
822:                Prev : Cdat;
823:                Next : Cdat;
824:                 end;
825:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
826:  Parray   = array[1..10] of Cdat;
827:
828:  Peaks    = ^PeakInfo;
829:  PeakInfo = record
830:                Pnum : integer;
831:              Name : string[12];
832:              Id   : integer;
833:              Rt   : real;
834:              Ht   : LongInt;
835:              Area : LongInt;
836:              Rf   : real;
837:              Conc : real;
838:              Next : Peaks;
839:              Prev : Peaks;
840:               end;
841:
842:  PeakRef  = Array[0..12] of PeakInfo;
843:
844:Var
845:  HeapTop   : ^integer;
846:  DataExist : boolean;
847:  Par1,Par2 : Parray;                { Use in FindPeak routine }
848:  Rat,FirstRat,LastRat : Rdat;
849:  FilRat    : Rdat;                 { Use in Filter routine }
850:  Ffil,Cfil,Lfil : Rdat;            { Use in Filter routine }
851:  Count     : LongInt;
852:  Cat,FirstCat,LastCat : Cdat;      { Use in InitCatCycle routine }
853:  CatA,CatB : Cdat;                 { Use in FindPeak routine }
854:  CatX,CatY : Cdat;                 { Use in FindPeak routine }
855:  CatI,CatJ : Cdat;                 { Use in FindPeak routine }
856:  Pek,FirstPek,LastPek : Peaks;     { Use in FindPeak routine }
857:  Rpek      : Peaks;
858:  Pref,Pidf : PeakRef;
859:  Psav      : PeakInfo;
860:  PrefMax   : byte;
861:
862:Procedure ClearDataBuffer;
863:begin
864:  Rat := FirstRat;
865:  FirstRat^.Next := nil;
866:  Pek := FirstPek;
867:  FirstPek^.Next := nil;
868:  Release(HeapTop);
869:end;
870:
871:Procedure InitLabNet(var ErrCode : byte;Com : Word);
872:var Attrib : byte;
873:    Ascii  : byte;
874:    Ch     : Char;
875:    i,j    : integer;
876:begin
877:  InitRs232(Com);
878:  LinkActiveOn(Com);
879:  for i := 1 to 61 do
880:  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
881:    RsRecive(Ascii,ErrCode,Com);
882:    if ErrCode = 128 then i := 61;
883:  end;
884:  if ErrCode <> 128 then
885:  begin
886:    SetBkTextColor(Blue);
887:    PutCenterText(112,200,' Link Active ',NormalType,LightRed);
888:    TriggerActiveOn(Com);
889:    PutCenterText(112,220,' Wait for Trigger ',ReverseType,Yellow);
890:    SetBkTextColor(Black);
891:    Attrib := ReverseType;
892:    j := 0;
893:    for i := 1 to 41 do
894:    begin
895:      RsRecive(Ascii,ErrCode,Com);
896:      if ErrCode = 128 then
897:      begin
898:        if Attrib = ReverseType then Attrib := NormalType
899:                                else Attrib := ReverseType;
900:        PutCenterText(112,220,' Wait for Trigger ',Attrib,White);
901:      end;
902:      inc(j);
903:      if j = 40 then i := 41 else Dec(i);  { Check Loop }
904:    end;
905:    SendXOff(Com);
906:    Sound(3500);Delay(60);Nosound;
907:    Mark(HeapTop);
908:    New(Rat);
909:    FirstRat := Rat;
910:  end else
911:  begin
912:    Sound(3500);Delay(120);Nosound;
913:    ShowMsg(23,200,' Time Out Error ! ','Error');
914:    Ch := ReadKey;
915:  end;
916:  SetFillStyle(SolidFill,Blue);
917:  Bar(1,179,217,300);
918:end;
919:
920:Procedure TakeRawData(var ErrCode : byte; Com : word);
921:begin
922:  Rat^.Data := GetRsData(ErrCode,Com);
923:  New(LastRat);
924:  Rat^.Next := LastRat;
925:end;
926:
927:Procedure InitLinkDataFilter;
928:var i : integer;
929:begin
930:  New(Ffil);
931:  Cfil := Ffil;
932:  for i := 1 to 4 do
933:  begin
934:    New(Lfil);
935:    Cfil^.Next := Lfil;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
936:    Cfil := Lfil;
937:  end;
938:  Cfil^.Next := Ffil;
939:  Cfil := Ffil;
940:end;
941:
942:Procedure DataFilter;
943:var idx : integer;
944:    sum : LongInt;
945:begin
946:  if Count < 5 then              { Start.. Reading RawData for Filter }
947:  begin
948:    Cfil^.Data := FilRat^.Data;
949:    FilRat := FilRat^.Next;
950:    Cfil   := Cfil^.Next;
951:  end else
952:  begin                          { 5 Points Average -> RawNo 5 }
953:    Cfil^.Data := FilRat^.Data;
954:    Lfil := Ffil;
955:    sum := 0;
956:    for idx := 1 to 5 do
957:    begin
958:      sum := sum + Lfil^.Data;
959:      Lfil := Lfil^.Next;
960:    end;
961:    FilRat^.Data := sum div 5;
962:    FilRat := FilRat^.Next;
963:    Ffil := Ffil^.Next;
964:    Cfil := Cfil^.Next;
965:  end;
966:end;
967:
968:Procedure InitLinkCat;
969:var i : integer;
970:begin
971:  New(FirstCat);
972:  Cat := FirstCat;
973:  for i := 1 to 20 do
974:  begin
975:    New(LastCat);
976:    Cat^.Next := LastCat;
977:    LastCat^.Prev := Cat;
978:    Cat := LastCat;
979:  end;
980:  Cat^.Next := FirstCat;
981:  FirstCat^.Prev := Cat;
982:  Cat := FirstCat;
983:end;
984:
985:Procedure InitPeak;
986:begin
987:  New(FirstPek);
988:  FirstPek^.Prev := nil;
989:  FirstPek^.Next := nil;
990:  Pek := FirstPek;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 991:end;
 992:
 993:Procedure GetPData(Pnew : Cdat; var Pn : Parray);
 994:var i : integer;
 995:begin
 996:  for i := 1 to 10 do
 997:  begin
 998:    Pn[i] := Pnew;
 999:    Pnew := Pnew^.Prev;
1000:  end;
1001:end;
1002:
1003:Procedure FirstDiff(var Pdif : Cdat; Pn : Parray);
1004:var i    : integer;
1005:    Fdif : LongInt;
1006:begin
1007:  Fdif := 0;
1008:  for i := 1 to 10 do
1009:    Fdif := Fdif+Pn[i]^.Data*(11-2*i);
1010:  Pdif^.Diff := Fdif;
1011:end;
1012:
1013:Const
1014:  PeakNumber = 1;
1015:  PeakName   = 2;
1016:  PeakRT     = 3;
1017:  PeakArea   = 4;
1018:  PeakRf     = 5;
1019:  PeakConc   = 6;
1020:
1021:Procedure WriteReport(x,y:word; Rtype:byte; Attrib:byte; Pinf:Peaks);
1022:var Rst : string;
1023:begin
1024:  FillChar(Rst,SizeOf(Rst),#0);
1025:  SetBkTextColor(Blue);
1026:  Case Rtype of
1027:    PeakNumber :begin
1028:                Str(Pinf^.Pnum:3,Rst);
1029:                PutTextXY(x,y,'   ',NormalType,White);
1030:                end;
1031:    PeakName   :begin
1032:                Rst := Pinf^.Name;
1033:              end;
1034:    PeakRT     :begin
1035:                Str((Pinf^.Rt/60):5:2,Rst);
1036:                PutTextXY(x,y,'     ',NormalType,Yellow);
1037:               end;
1038:    PeakArea   :begin
1039:                Str(Pinf^.Area:9,Rst);
1040:                PutTextXY(x,y,'         ',NormalType,Yellow);
1041:              end;
1042:    PeakRf     :begin
1043:                Str(Pinf^.Rf,Rst);
1044:                PutTextXY(x,y,'     ',NormalType,Yellow);
1045:                end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1046:    PeakConc   :begin
1047:                Str(Pinf^.Conc,Rst);
1048:                PutTextXY(x,y,'      ',NormalType,Yellow);
1049:                end;
1050:  end;
1051:
1052:  PutTextXY(x,y,Rst,Attrib,Yellow);
1053:  SetBkTextColor(Black);
1054:end;
1055:
1056:Procedure PeakReport(Pnum : integer; Attrib : byte; Rpk:Peaks);
1057:var my : integer;
1058:    dy : integer;
1059:begin
1060:    my := (Pnum-1) mod 20;
1061:    dy := my*13;
1062:    SetBkTextColor(Blue);
1063:    PutTextXY(14,192,'Pk#       RT        Area',NormalType,Yellow);
1064:    WriteReport(14,210+dy,PeakNumber,Attrib,Rpk);
1065:    WriteReport(68,210+dy,PeakRT,Attrib,Rpk);
1066:    WriteReport(134,210+dy,PeakArea,Attrib,Rpk);
1067:    SetBkTextColor(Black);
1068:end;
1069:
1070:Procedure PeakIdentify;
1071:var IdCheck  : integer;
1072:    Pn       : integer;
1073:    LimLeft  : real;
1074:    LimRight : real;
1075:    Rmin     : real;
1076:    i        : integer;
1077:begin
1078:  Pn := 0;
1079:  for i := 1 to PrefMax do
1080:  begin
1081:    Pek := FirstPek;
1082:    IdCheck := 0;
1083:    Pidf[i].Id := 0;
1084:    Pidf[i].Rf := 1;
1085:    Rmin := 100000.0;
1086:    LimLeft  := (1.00-GetValueChoice(PeakWindowChoice)/100.0)*Pref[i].Rt;
1087:    LimRight := (1.00+GetValueChoice(PeakWindowChoice)/100.0)*Pref[i].Rt;
1088:
1089:    while Pek^.Next <> nil do
1090:    begin
1091:      if Pek^.Id = 0 then
1092:     if (Pek^.Rt < LimRight) and
1093:        (Pek^.Rt > LimLeft ) then
1094:     begin
1095:       IdCheck := IdCheck+1;
1096:       if IdCheck = 1 then
1097:       begin
1098:         Pidf[i].Id := 1;
1099:         Rmin := Abs(Pref[i].Rt-Pek^.Rt);
1100:         Pidf[i].Name := Pref[i].Name;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1101:         Pidf[i].Rt   := Pekˆ.Rt;
1102:         Pidf[i].Area := Pekˆ.Area;
1103:         Pidf[i].Rf   := Pref[i].Rf;
1104:       end else
1105:       if (IdCheck > 1) and (Rmin > Abs(Pref[i].Rt-Pekˆ.Rt)) then
1106:       begin
1107:            Rmin := Abs(Pref[i].Rt-Pekˆ.Rt);
1108:         Pidf[i].Rt   := Pekˆ.Rt;
1109:         Pidf[i].Area := Pekˆ.Area;
1110:       end;
1111:     end;
1112:     Pek := Pekˆ.Next;
1113:    end;
1114:
1115:    if IdCheck > 0 then
1116:    begin
1117:      Inc(Pn);
1118:      Pidf[i].Pnum := Pn;
1119:      Pidf[i].id := 1;
1120:      Pek := FirstPek;
1121:      while Pekˆ.Next <> nil do
1122:      begin
1123:     if Pidf[i].Rt = Pekˆ.Rt then Pekˆ.Id := 1;
1124:     Pek := Pekˆ.Next;
1125:      end;
1126:    end;
1127:
1128:  end;
1129:end;
1130:
1131:Procedure PeakReportMtd_B;
1132:const ColName = 10;
1133:       ColRt   = 90;
1134:       ColConc = 138;
1135:       RefRow  = 190;
1136:var    j : integer;
1137:      St : string;
1138:begin
1139:  SetFillStyle(SolidFill,Blue);
1140:  Bar(1,180,217,478);
1141:  PeakIdentify;
1142:  SetBkTextColor(Blue);
1143:  PutTextXY(10,190,'Name          Rt      PPM',NormalType,LightGreen);
1144:  for j := 1 to PrefMax do
1145:  begin
1146:    if Pidf[j].Id = 1 then
1147:    begin
1148:      SetBkTextColor(Blue);
1149:      Pidf[j].Conc := Pidf[j].Area/Pidf[j].Rf;
1150:      PutTextXY(ColName,RefRow+Pidf[j].Pnum*15,Pidf[j].Name,NormalType,White)
1151:      Str(Pidf[j].Rt/60:5:2,St);
1152:      PutTextXY(ColRt,RefRow+Pidf[j].Pnum*15,St,NormalType,White);
1153:      Str(Pidf[j].Conc:9:2,St);
1154:      PutTextXY(ColConc,RefRow+Pidf[j].Pnum*15,St,NormalType,White);
1155:      SetBkTextColor(Black);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1156:    end;
1157:  end;
1158:end;
1159:
1160:Procedure WriteSignalLevel(sl:LongInt);
1161:var Slevel : string;
1162:begin
1163:  Str(sl:6,Slevel);
1164:  PutTextXY(560,190,Slevel,NormalType,LightRed);
1165:end;
1166:
1167:var Psum    : LongInt;
1168:    Parea   : LongInt;
1169:    Th1,Th2 : LongInt;
1170:    PmarkA  : real;
1171:    PmarkB  : real;
1172:    MarkA   : boolean;
1173:    MarkB   : boolean;
1174:    Pfound  : boolean;
1175:    Pcount  : integer;
1176:
1177:Procedure FindPeak;
1178:var i    : integer;
1179:    Cdx  : Longint;
1180:    OK   : boolean;
1181:begin
1182:  if Count = 1 then
1183:  begin
1184:    CatA := FirstCat;
1185:    CatB := FirstCat^.Prev;
1186:
1187:    CatX := FirstCat;
1188:    for i := 1 to 4 do
1189:      CatX := CatX^.Prev;
1190:    CatY := CatX^.Prev;
1191:
1192:    CatI := FirstCat^.Next;
1193:
1194:    Th1 :=      Threshold*PeakWidth;
1195:    Th2 := (-1)*Threshold*PeakWidth div 4;
1196:    Pfound := False;
1197:    Psum   := 0;
1198:    Parea  := 0;
1199:    Pek := FirstPek;
1200:    Pek^.Id := 0;
1201:    Pcount := 0;
1202:  end;
1203:  Cdx := (Count div PeakWidth)+1;
1204:  if (Count mod PeakWidth) = 0 then OK := True else OK := False;
1205:
1206:  if Cdx <= 20 then
1207:  begin
1208:    if OK then
1209:    begin
1210:      Psum := Psum + Rat^.Data;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1211:       CatI^.Data := Psum;
1212:       CatI^.Idex := Cdx;
1213:       CatI := CatI^.Next;
1214:       Psum := 0;
1215:     end else
1216:     begin
1217:       Psum := Psum + Rat^.Data;
1218:     end;
1219:   end else
1220:   begin
1221:     if Ok then
1222:     begin
1223:       GetPData(CatA,Par1);
1224:       FirstDiff(CatX,Par1);
1225:       GetPData(CatB,Par2);
1226:       FirstDiff(CatY,Par2);
1227:
1228:       if (Pfound = False) and (Count > 100) then
1229:       begin
1230:      if (CatX^.Diff >= Th1) and (CatY^.Diff < Th1) then
1231:      begin
1232:        Sound(3000);delay(50);NoSound;
1233:        PmarkA  := (CatY^.Idex*PeakWidth-(PeakWidth div 2))*0.1;
1234:        MarkA   := True;
1235:        Parea  := 0;
1236:        Pfound := True;
1237:        Inc(Pcount);
1238:        Pek^.Pnum := Pcount;
1239:        SetBkTextColor(Blue);
1240:        PutTextXY(396,190,'Peak# ',NormalType,Yellow);
1241:        WriteReport(438,190,PeakNumber,NormalType,Pek);
1242:        SetBkTextColor(Black);
1243:      end;
1244:       end;
1245:
1246:       if Pfound = True then
1247:       begin
1248:      Parea := Parea + CatX^.Data - (BaseLine*PeakWidth);
1249:
1250:      if (CatX^.Diff <= 0) and (CatY^.Diff > 0) then
1251:      begin
1252:        Pek^.Rt := ((CatX^.Idex * PeakWidth) - (PeakWidth div 2)) * 0.1;
1253:        Pek^.Ht := Rat^.Data;
1254:      end;
1255:
1256:      if (CatX^.Diff >= Th2) and (CatY^.Diff < Th2) then
1257:      begin
1258:        Sound(4000);delay(50);Nosound;
1259:        Pek^.Area := Parea;
1260:        PeakReport(Pcount,ReverseType,Pek);
1261:        if Pcount > 1 then
1262:          PeakReport(Pcount-1,NormalType,Rpek);
1263:        Rpek      := Pek;
1264:        if Pek^.Next = nil then
1265:        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1266:         New(LastPek);
1267:         LastPek^.Prev := Pek;
1268:         LastPek^.Next := nil;
1269:         Pek^.Next := LastPek;
1270:         Pek := LastPek;
1271:       end else Pek := Pek^.Next;
1272:       Pek^.Id := 0;
1273:       PmarkB := (CatY^.Idex*PeakWidth-(PeakWidth div 2))*0.1;
1274:       MarkB  := True;
1275:       Pfound := False;
1276:
1277:          if (CatX^.Diff >= Th1) and (CatY^.Diff < Th1) then
1278:       begin
1279:         Sound(3000);delay(50);NoSound;
1280:         PmarkA  := (CatX^.Idex*PeakWidth-(PeakWidth div 2))*0.1;
1281:         MarkA   := True;
1282:         Parea  := 0;
1283:         Pfound := True;
1284:            Inc(Pcount);
1285:         Pek^.Pnum := Pcount;
1286:         SetBkTextColor(Blue);
1287:         PutTextXY(396,190,'Peak# ',NormalType,Yellow);
1288:         WriteReport(438,190,PeakNumber,NormalType,Pek);
1289:         SetBkTextColor(Black);
1290:       end;
1291:     end;
1292:       end;
1293:
1294:       CatA := CatA^.Next;
1295:       CatB := CatB^.Next;
1296:       CatX := CatX^.Next;
1297:       CatY := CatY^.Next;
1298:
1299:       Psum := Psum + Rat^.Data;
1300:       CatA^.Data := Psum;
1301:       CatA^.Idex := Cdx;
1302:       Psum := 0;
1303:     end else
1304:     begin
1305:       Psum := Psum + Rat^.Data;
1306:     end;
1307:
1308:   end;
1309:end;
1310:
1311:(* ---------------------------- Mode Menu Section ---------------------------- *)
1312:
1313:Procedure ShowModeMenu;
1314:begin
1315:   SetFillStyle(InterleaveFill,Green);
1316:   Bar(178,55,622,418);
1317:   SetBkTextColor(Blue);
1318:   DrawWin(267,82,533,247,MixedType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1319:   DrawWin(221,313,579,346,ThickType,Yellow,Blue,ShadowOn,Black,'',NormalType)
1320:   PutCenterText(400,329,' Use Up-Down key to select and Press ENTER ',
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1321:                 NormalType,White);
1322:  SetBkTextColor(Black);
1323:end;
1324:
1325:Procedure RunModeMenu;
1326:var ExitOK : boolean;
1327:begin
1328:  DrawCommandMenu(ModeMenu_,RunOn);
1329:  ClearMainScreen;
1330:  ShowModeMenu;
1331:  ModeChoice^.Cidx := Choice(328,133,ModeChoice);
1332:  ActiveMode := ModeChoice^.Cidx;
1333:  DrawCommandMenu(ModeMenu_,ShowOn);
1334:  DrawCommandMenu(ModeMenu_,SelectOn);
1335:end;
1336:
1337:(* ------------------------ Option Menu Section ------------------------ *)
1338:
1339:Procedure ShowOptionMenu;
1340:begin
1341:  SetFillStyle(InterLeaveFill,Green);
1342:  Bar(178,55,622,418);
1343:  DrawWin(250,82,550,324,DoubleType,Yellow,Blue,ShadowOn,Black,' Option ',
1344:      ReverseType);
1345:  DrawWin(188,354,612,386,ThickType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1346:  SetBkTextColor(Blue);
1347:  PutCenterText(400,370,' Up-Down key [SELECT]  Spacebar [EDIT]  Esc [EXIT] ',
1348:              NormalType,Yellow);
1349:  ProcessMenuChoice(PcShow,RunTimeChoice);
1350:  ProcessMenuChoice(PcShow,AttenChoice);
1351:  ProcessMenuChoice(PcShow,PageRangeChoice);
1352:  ProcessMenuChoice(PcShow,PeakWidthChoice);
1353:  ProcessMenuChoice(PcShow,ThresholdChoice);
1354:  ProcessMenuChoice(PcShow,SerialPortChoice);
1355:  ProcessMenuChoice(PcShow,PeakWindowChoice);
1356:  ProcessMenuChoice(PcShow,PeakMarkerChoice);
1357:  ProcessMenuChoice(PcShow,FilterChoice);
1358:  ProcessMenuChoice(PcShow,BaseLineChoice);
1359:  SetBkTextColor(Black);
1360:end;
1361:
1362:Procedure SetValueChoice;
1363:begin
1364:  WorkTime   := GetValueChoice(RunTimeChoice);
1365:  AttenLevel := GetValueChoice(AttenChoice);
1366:  PageRange  := GetValueChoice(PageRangeChoice);
1367:  PeakWidth  := GetValueChoice(PeakWidthChoice);
1368:  Threshold  := GetValueChoice(ThresholdChoice);
1369:  SerialPort := GetValueChoice(SerialPortChoice);
1370:  PeakMarker := GetValueChoice(PeakMarkerChoice);
1371:  Filter     := GetValueChoice(FilterChoice);
1372:  BaseLine   := GetValueChoice(BaseLineChoice);
1373:  Ddir       := GetChoice(DataDirChoice);
1374:end;
1375:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1376:Procedure RunOptionMenu;
1377:var i  : byte;
1378:    Ch : char;
1379:begin
1380:  DrawCommandMenu(OptionMenu_,RunOn);
1381:  ClearMainScreen;
1382:  ShowOptionMenu;
1383:  TempChoice := RunTimeChoice;
1384:  ProcessMenuChoice(PcActive,RunTimeChoice);
1385:  Repeat
1386:    Repeat
1387:      WriteCurrentTime;
1388:      Until KeyPressed;
1389:      Ch := ReadKey;
1390:      Case Ch of
1391:     Up_Key :begin
1392:                 ProcessMenuChoice(PcShow,TempChoice);
1393:                 TempChoice := TempChoice^.Bwd;
1394:                 ProcessMenuChoice(PcActive,TempChoice);
1395:               end;
1396:     Dn_Key :begin
1397:                   ProcessMenuChoice(PcShow,TempChoice);
1398:                 TempChoice := TempChoice^.Fwd;
1399:                 ProcessMenuChoice(PcActive,TempChoice);
1400:               end;
1401:     ' '    :begin
1402:                 ProcessMenuChoice(PcEdit,TempChoice);
1403:               end;
1404:       end;
1405:  Until Ch = Esc_Key;
1406:  SetValueChoice;
1407:  BottomLimit := BaseLine-(AttenLevel*100);
1408:  SetMainGraph(MainGraph);
1409:  SetMapGraph(MapGraph);
1410:  SetTimeGraph(TimeGraph);
1411:  DrawCommandMenu(OptionMenu_,ShowOn);
1412:  DrawCommandMenu(OptionMenu_,SelectOn);
1413:end;
1414:
1415:(* ------------------------ Method Menu Section ---------------------- *)
1416:Procedure InitPeakRef;
1417:var Pn : integer;
1418:begin
1419:  for Pn := 0 to 12 do
1420:  begin
1421:    FillChar(Pref[Pn].Name,Length(Pref[Pn].Name),#0);
1422:    Pref[Pn].Name := 'None';
1423:    Pref[Pn].Rt   := 0.0;
1424:    Pref[Pn].Area := 0;
1425:    Pref[Pn].Rf   := 1.0;
1426:    Pref[Pn].Conc := 0.0;
1427:  end;
1428:  PrefMax := 0;
1429:end;
1430:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1431:Procedure WritePrefName(x,y:word; Attrib:byte; Pnum:byte);
1432:begin
1433:  PutTextXY(x,y,'          ',Normaltype,White);
1434:  PutTextXY(x,y,Pref[Pnum].Name,Attrib,White);
1435:end;
1436:
1437:Procedure WritePrefRt(x,y:word; Attrib:byte; Pnum:byte);
1438:var St : string;
1439:begin
1440:  Str(Pref[Pnum].Rt/60:6:2,St);
1441:  PutTextXY(x,y,St,Attrib,White);
1442:end;
1443:
1444:Procedure WritePrefArea(x,y:word; Attrib:byte; Pnum:byte);
1445:var St : string;
1446:begin
1447:  Str(Pref[Pnum].Area:9,St);
1448:  PutTextXY(x,y,St,Attrib,White);
1449:end;
1450:
1451:Procedure WritePrefRf(x,y:word; Attrib:byte; Pnum:byte);
1452:var St : string;
1453:begin
1454:  Str(Pref[Pnum].Rf:8:3,St);
1455:  PutTextXY(x,y,St,Attrib,White);
1456:end;
1457:
1458:Procedure WritePrefConc(x,y:word; Attrib:byte; Pnum:byte);
1459:var St : string;
1460:begin
1461:  Str(Pref[Pnum].Rt:7:2,St);
1462:  PutTextXY(x,y,St,Attrib,White);
1463:end;
1464:
1465:Procedure EditPrefName(x,y:word; Pnum:byte);
1466:var St : string;
1467:begin
1468:  MoveTo(x,y-3);
1469:  FillChar(St,Length(St),#0);
1470:  PutTextXY(x,y,'          ',Normaltype,White);
1471:  EditGraphString(St);
1472:  if St = '' then Pref[Pnum].Name := 'None'
1473:         else Pref[Pnum].Name := St;
1474:  PutTextXY(x,y,'          ',Normaltype,White);
1475:  WritePrefName(x,y,NormalType,Pnum);
1476:end;
1477:
1478:Procedure EditPrefRt(x,y:word; Pnum:byte);
1479:var St   : string;
1480:    Code : integer;
1481:begin
1482:  MoveTo(x,y-3);
1483:  FillChar(St,Length(St),#0);
1484:  PutTextXY(x,y,'      ',Normaltype,White);
1485:  EditGraphString(St);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1486:  Val(St,Pref[Pnum].Rt,Code);
1487:  Pref[Pnum].Rt := Pref[Pnum].Rt*60;
1488:  PutTextXY(x,y,'      ',Normaltype,White);
1489:  WritePrefRt(x,y,NormalType,Pnum);
1490:end;
1491:
1492:Procedure EditPrefArea(x,y:word; Pnum:byte);
1493:var St   : string;
1494:    Code : integer;
1495:begin
1496:  MoveTo(x,y-3);
1497:  FillChar(St,Length(St),#0);
1498:  PutTextXY(x,y,'         ',Normaltype,White);
1499:  EditGraphString(St);
1500:  Val(St,Pref[Pnum].Area,Code);
1501:  PutTextXY(x,y,'         ',Normaltype,White);
1502:  WritePrefArea(x,y,NormalType,Pnum);
1503:end;
1504:
1505:Procedure EditPrefRf(x,y:word; Pnum:byte);
1506:var St   : string;
1507:    Code : integer;
1508:begin
1509:  MoveTo(x,y-3);
1510:  FillChar(St,Length(St),#0);
1511:  PutTextXY(x,y,'         ',Normaltype,White);
1512:  EditGraphString(St);
1513:  Val(St,Pref[Pnum].Rf,Code);
1514:  if Pref[Pnum].Rf = 0.0 then Pref[Pnum].Rf := 1.0;
1515:  PutTextXY(x,y,'         ',Normaltype,White);
1516:  WritePrefRf(x,y,NormalType,Pnum);
1517:end;
1518:
1519:Procedure EditPrefConc(x,y:word; Pnum:byte);
1520:var St   : string;
1521:    Code : integer;
1522:begin
1523:  MoveTo(x,y-3);
1524:  FillChar(St,Length(St),#0);
1525:  PutTextXY(x,y,'       ',Normaltype,White);
1526:  EditGraphString(St);
1527:  Val(St,Pref[Pnum].Conc,Code);
1528:  PutTextXY(x,y,'       ',Normaltype,White);
1529:  WritePrefConc(x,y,NormalType,Pnum);
1530:end;
1531:
1532:Procedure EditPeakRefMtd_B; { For Method B only }
1533:const PkRefMax = 12;
1534:      Col1     = 256;
1535:      Col2     = 360;
1536:      Col3     = 520;
1537:      RefCol   = 90;
1538:var  Erow,Ecol : integer;
1539:     Orow,Ocol : integer;
1540:     Pn,i,j    : integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1541:     Ch         : char;
1542:     St         : string;
1543:begin
1544:  ShowMenu(176,60,52,16,' External Std. ');
1545:  SetBkTextColor(Blue);
1546:  PutTextXY(208,RefCol,'No#       RT       Name                          RF',
1547:        NormalType,Yellow);
1548:  DrawWin(188,348,612,402,ThickType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1549:  SetBkTextColor(Blue);
1550:  PutCenterText(400,366,' Arrow key [SELECT]  Spacebar [EDIT]  Esc [EXIT] ',
1551:             NormalType,Yellow);
1552:  PutCenterText(400,386,'      F8 [DELETE REF.]    F9 [APPEND REF.]      ',
1553:             NormalType,LightRed);
1554:  SetBkTextColor(Black);
1555:  Pn := 1;
1556:  if PrefMax = 0 then
1557:  begin       { Create New PeakRef }
1558:    Repeat
1559:      WriteCurrentTime;
1560:      Str(Pn:2,St);
1561:      PutTextXY(208,RefCol+Pn*15,St,NormalType,LightMagenta);
1562:      EditPrefRt(Col1,RefCol+Pn*15,Pn);
1563:      WriteCurrentTime;
1564:      EditPrefName(Col2,RefCol+Pn*15,Pn);
1565:      WriteCurrentTime;
1566:      EditPrefRf(Col3,RefCol+Pn*15,Pn);
1567:      Inc(Pn); Inc(PrefMax);
1568:      Repeat
1569:     WriteCurrentTime;
1570:      Until KeyPressed;
1571:     Ch := ReadKey;
1572:    Until (Ch = Esc_Key) or (Pn > PkRefMax);
1573:  end else
1574:  begin        { List Current PeakRef }
1575:    for Pn := 1 to PrefMax do
1576:    begin
1577:      Str(Pn:2,St);
1578:      PutTextXY(208,RefCol+Pn*15,St,NormalType,LightMagenta);
1579:      WritePrefRt(Col1,RefCol+Pn*15,NormalType,Pn);
1580:      WritePrefName(Col2,RefCol+Pn*15,NormalType,Pn);
1581:      WritePrefRf(Col3,RefCol+Pn*15,NormalType,Pn);
1582:    end;
1583:  end;
1584:  Erow := 1; Orow := 1;
1585:  Ecol := 1; Ocol := 1;
1586:  WritePrefRt(Col1,RefCol+Erow*15,ReverseType,Erow);
1587:
1588:  Repeat           { Edit Current PeakRef }
1589:    Repeat
1590:      WriteCurrentTime;
1591:    Until KeyPressed;
1592:  Ch := ReadKey;
1593:  Case Ch of
1594:    Up_Key : begin
1595:          Dec(Erow);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1596:            if Erow < 1 then Erow := PrefMax;
1597:          end;
1598:     Dn_Key : begin
1599:            Inc(Erow);
1600:            if Erow > PrefMax then Erow := 1;
1601:          end;
1602:     Lt_Key : begin
1603:            Dec(Ecol);
1604:            if Ecol < 1 then Ecol := 3;
1605:          end;
1606:     Rt_Key : begin
1607:            Inc(Ecol);
1608:               If Ecol > 3 then Ecol := 1;
1609:             end;
1610:     ' '    : begin
1611:            Case Ecol of
1612:               1 : EditPrefRt(Col1,RefCol+Erow*15,Erow);
1613:               2 : EditPrefName(Col2,RefCol+Erow*15,Erow);
1614:               3 : EditPrefRf(Col3,RefCol+Erow*15,Erow);
1615:            end;
1616:             end;
1617:
1618:     F8_Key : begin
1619:               ShowMenu(176,60,52,16,' External Std. ');
1620:            SetBkTextColor(Blue);
1621:            PutTextXY(208,RefCol,'No#       RT          '+
1622:                 'Name                    RF',NormalType,Yellow);
1623:            SetBkTextColor(Black);
1624:            for i := Erow to (PrefMax-1) do
1625:             Pref[i] := Pref[i+1];
1626:            PrefMax := PrefMax-1;
1627:            if PrefMax < 1 then
1628:            begin
1629:               PrefMax := 1;
1630:               Pref[1] := Pref[0];
1631:            end;
1632:               for Pn := 1 to PrefMax do
1633:            begin
1634:               Str(Pn:2,St);
1635:               PutTextXY(208,RefCol+Pn*15,St,NormalType,LightMagenta);
1636:               WritePrefRt(Col1,RefCol+Pn*15,NormalType,Pn);
1637:               WritePrefName(Col2,RefCol+Pn*15,NormalType,Pn);
1638:               WritePrefRf(Col3,RefCol+Pn*15,NormalType,Pn);
1639:            end;
1640:                 Erow := 1; Orow := 1;
1641:               Ecol := 1; Ocol := 1;
1642:               WritePrefRt(Col1,RefCol+Erow*15,ReverseType,Erow);
1643:          end;
1644:     F9_Key : begin
1645:               ShowMenu(176,60,52,16,' External Std. ');
1646:            SetBkTextColor(Blue);
1647:            PutTextXY(208,RefCol,'No#       RT          '+
1648:                        'Name                       RF',NormalType,Yellow);
1649:            SetBkTextColor(Black);
1650:               for Pn := 1 to PrefMax do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1651:            begin
1652:               Str(Pn:2,St);
1653:               PutTextXY(208,RefCol+Pn*15,St,NormalType,LightMagenta);
1654:               WritePrefRt(Col1,RefCol+Pn*15,NormalType,Pn);
1655:               WritePrefName(Col2,RefCol+Pn*15,NormalType,Pn);
1656:               WritePrefRf(Col3,RefCol+Pn*15,NormalType,Pn);
1657:            end;
1658:            PrefMax := PrefMax+1;
1659:            if PrefMax > 12 then PrefMax := 12;
1660:            Pn := PrefMax;
1661:            Pref[Pn] := Pref[0];
1662:               Str(Pn:2,St);
1663:            PutTextXY(208,RefCol+Pn*15,St,NormalType,LightMagenta);
1664:               WritePrefRt(Col1,RefCol+Pn*15,NormalType,Pn);
1665:            WritePrefName(Col2,RefCol+Pn*15,NormalType,Pn);
1666:            WritePrefRf(Col3,RefCol+Pn*15,NormalType,Pn);
1667:            Erow := Pn; Orow := Pn;
1668:            Ecol := 1; Ocol := 1;
1669:            WritePrefRt(Col1,RefCol+Erow*15,ReverseType,Erow);
1670:             end;
1671:  end;
1672:
1673:  Case Ocol of
1674:    1 : WritePrefRt(Col1,RefCol+Orow*15,NormalType,Orow);
1675:    2 : WritePrefName(Col2,RefCol+Orow*15,NormalType,Orow);
1676:    3 : WritePrefRf(Col3,RefCol+Orow*15,NormalType,Orow);
1677:  end;
1678:
1679:  Case Ecol of
1680:    1 : begin
1681:       WritePrefRt(Col1,RefCol+Erow*15,ReverseType,Erow);
1682:       Orow := Erow; Ocol := Ecol;
1683:     end;
1684:    2 : begin
1685:       WritePrefName(Col2,RefCol+Erow*15,ReverseType,Erow);
1686:       Orow := Erow; Ocol := Ecol;
1687:     end;
1688:    3 : begin
1689:       WritePrefRf(Col3,RefCol+Erow*15,ReverseType,Erow);
1690:       Orow := Erow; Ocol := Ecol;
1691:     end;
1692:  end;
1693:
1694:  Until Ch = Esc_Key;
1695:
1696:  (*   PeakRef Sorting   *)
1697:
1698:  for i := 2 to PrefMax do
1699:  begin
1700:    Psav := Pref[i];
1701:    j := i-1;
1702:    While (Psav.Rt < Pref[j].Rt) and (j > 0) Do
1703:    begin
1704:      Pref[j+1] := Pref[j];
1705:      j := j-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1706:    end;
1707:    Pref[j+1] := Psav;
1708:  end;
1709:end;
1710:
1711:Procedure SetMethod_A;
1712:begin
1713:  ClearMainScreen;
1714:end;
1715:
1716:Procedure SetMethod_B;
1717:begin
1718:  ClearMainScreen;
1719:  EditPeakRefMtd_B;
1720:end;
1721:
1722:Procedure SetMethod_C;
1723:begin
1724:  ClearMainScreen;
1725:end;
1726:
1727:Procedure SetMethod_D;
1728:begin
1729:  ClearMainScreen;
1730:end;
1731:
1732:Procedure RunMethodMenu;
1733:begin
1734:  DrawCommandMenu(MethodMenu_,RunOn);
1735:  ClearMainScreen;
1736:  SetFillStyle(InterleaveFill,Green);
1737:  Bar(178,55,622,418);
1738:  SetBkTextColor(Blue);
1739:  DrawWin(267,82,533,247,MixedType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1740:  DrawWin(221,313,579,346,ThickType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1741:  PutCenterText(400,329,' Use Up-Down key to select and Press ENTER ',
1742:            NormalType,White);
1743:  SetBkTextColor(Black);
1744:  MethodChoice^.Cidx := Choice(316,120,MethodChoice);
1745:  ActiveMethod := MethodChoice^.Cidx;
1746:  Case ActiveMethod of
1747:    Method_A : SetMethod_A;
1748:    Method_B : SetMethod_B;
1749:    Method_C : SetMethod_C;
1750:    Method_D : SetMethod_D;
1751:  end; { Case }
1752:  DrawCommandMenu(MethodMenu_,ShowOn);
1753:  DrawCommandMenu(MethodMenu_,SelectOn);
1754:end;
1755:
1756:(* ---------------------- File Menu Section ---------------------- *)
1757:
1758:Procedure ShowFileMenu;
1759:begin
1760:  SetFillStyle(InterLeaveFill,Green);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1761:  Bar(178,55,622,418);
1762:  DrawWin(248,137,550,216,DoubleType,Yellow,Blue,ShadowOn,Black,
1763:                          ' File ',ReverseType);
1764:  SetBkTextColor(Blue);
1765:  DrawWin(188,312,610,346,ThickType,Yellow,Blue,ShadowOn,Black,'',NormalType);
1766:  PutCenterText(400,330,' Up-Down key [SELECT]  Spacebar [EDIT]  Esc [EXIT] ',
1767:                                              NormalType,Yellow);
1768:  ProcessMenuChoice(PcShow,DataFileChoice);
1769:  ProcessMenuChoice(PcShow,DataDirChoice);
1770:end;
1771:
1772:Procedure RunFileMenu;
1773:var Ch : Char;
1774:begin
1775:  DrawCommandMenu(FileMenu_,RunOn);
1776:  ClearMainScreen;
1777:  ShowFileMenu;
1778:  TempChoice := DataFileChoice;
1779:  ProcessMenuChoice(PcActive,DataFileChoice);
1780:  Repeat
1781:    Repeat
1782:      WriteCurrentTime;
1783:    Until KeyPressed;
1784:    Ch := ReadKey;
1785:    Case Ch of
1786:      Up_Key :begin
1787:                ProcessMenuChoice(PcShow,TempChoice);
1788:                TempChoice := TempChoice^.Bwd;
1789:                ProcessMenuChoice(PcActive,TempChoice);
1790:            end;
1791:      Dn_Key :begin
1792:                 ProcessMenuChoice(PcShow,TempChoice);
1793:                TempChoice := TempChoice^.Fwd;
1794:                ProcessMenuChoice(PcActive,TempChoice);
1795:                end;
1796:      ' '    :begin
1797:                ProcessMenuChoice(PcEdit,TempChoice);
1798:                end;
1799:    end;
1800:  Until Ch = Esc_Key;
1801:  Ddir   :=  GetChoice(DataDirChoice);
1802:  SetDataFileName(GetChoice(DataFileChoice));
1803:  DrawCommandMenu(FileMenu_,ShowOn);
1804:  DrawCommandMenu(FileMenu_,SelectOn);
1805:end;
1806:
1807:(* ------------------------- Go Menu Section ------------------------- *)
1808:
1809:Function FileExist(Fname : string) : Boolean;
1810:type DataFile = File of LongInt;
1811:var  Fdat   : DataFile;
1812:begin
1813:  Assign(Fdat,Fname);
1814:  {$I-} Reset(Fdat) {$I+} ;
1815:  if IOresult <> 0 then FileExist := False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1816:                  else begin FileExist := True; Close(Fdat); end;
1817:end;
1818:
1819:Procedure InitPlotScreen;
1820:begin
1821:  DrawWin(0,0,GetMaxX,137,DoubleType,Yellow,Blue,ShadowOff,Black,'',
1822:       ReverseType);
1823:  DrawWin(218,151,GetMaxX,GetMaxY,ThickType,Yellow,Blue,ShadowOff,Black,'',
1824:       ReverseType);
1825:  Line(218,178,GetMaxX,178);
1826:  DrawWin(0,151,218,GetMaxY,SingleType,Yellow,Blue,ShadowOff,Black,'',
1827:          NormalType);
1828:  Rectangle(5,155,214,178);
1829:  SetBkTextColor(Blue);
1830:  PutCenterText(112,166,'Report',NormalType,Yellow);
1831:  SetBkTextColor(Black);
1832:  ClearGraphWindow(MainGraph);
1833:  ClearGraphWindow(MapGraph);
1834:  PlotGraphWindow(MapGraph);
1835:  PlotGraphWindow(MainGraph);
1836:  PlotGrid(MainGraph);
1837:  PlotScale(MainGraph);
1838:  PlotAxis(MainGraph);
1839:  PlotGrid(TimeGraph);
1840:  PlotScale(TimeGraph);
1841:end;
1842:
1843:Procedure PlotPeakMark;
1844:begin
1845:  if PeakMarker = MarkerOn then
1846:  begin
1847:
1848:  if MarkA = True then
1849:  begin
1850:    SetGraphPlotColor(LightGreen,MainGraph);
1851:    PlotLine(PmarkA,1.0*BaseLine,PmarkA,BottomLimit,MainGraph);
1852:    SetGraphPlotColor(Yellow,MainGraph);
1853:    MarkA := False;
1854:  end;
1855:
1856:  if MarkB = True then
1857:  begin
1858:    SetGraphPlotColor(LightRed,MainGraph);
1859:    PlotLine(PmarkB,1.0*BaseLine,PmarkB,
1860:    BottomLimit+(Round(BaseLine-BottomLimit) Div 2),MainGraph);
1861:    SetGraphPlotColor(Yellow,MainGraph);
1862:    MarkB := False;
1863:  end;
1864:
1865:  end;
1866:end;
1867:
1868:Procedure PlotNewPage;
1869:begin
1870:  TimeStart := TimeFinish;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
871:  SetMainGraph(MainGraph);
872:  ClearGraphWindow(MainGraph);
873:  PlotGraphWindow(MainGraph);
874:  SetFillStyle(SolidFill,Blue);
875:  Bar(222,452,634,475);
876:  PlotGrid(MainGraph);
877:  PlotScale(MainGraph);
878:end;
879:
880:Procedure PlotStatus;
881:var St : string;
882:begin
883:  SetBkTextColor(Blue);
884:  ActiveMode := GetValueChoice(ModeChoice);
885:  if ActiveMode = LabNetMode then
886:  begin
887:    St := GetChoice(SerialPortChoice);
888:    St := Concat('LabNet [',St,']');
889:    PutTextXY(231,166,St,NormalType,Yellow);
890:
891:    St := GetChoice(RunTimeChoice);
892:    St := Concat('Runtime ',St);
893:    PutTextXY(514,166,St,NormalType,Yellow);
894:  end else
895:  begin
896:    PutTextXY(231,166,'FileMode',NormalType,Yellow);
897:    Str(WorkTime,St);
898:    St := Concat('Runtime ',St,' min.');
899:    PutTextXY(510,166,St,NormalType,Yellow);
900:  end;
901:
902:  St := GetChoice(MethodChoice);
903:  PutCenterText(428,166,St,NormalType,Yellow);
904:
905:  Filter := GetValueChoice(FilterChoice);
906:  if Filter = FilterOn
907:    then PutTextXY(230,190,'Filter [ON]',ReverseType,Yellow)
908:    else PutTextXY(230,190,'Filter [OFF]',ReverseType,Yellow);
909:  SetBkTextColor(Black);
910:end;
911:
912:Var ErrCode  : byte;
913:    rx,ry    : real;
914:    Pmain    : PlotInfo;
915:    Pmap     : PlotInfo;
916:    CountMax : LongInt;
917:    Lint     : LongInt;
918:    CalBLine : real;
919:    BLst     : string;
920:
921:Procedure SetLabNetInfo(var Wtime : word);
922:begin
923:  Wtime := GetValueChoice(RunTimeChoice);
924:end;
925:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1926:Procedure SetFileInfo(Fname:string; var Wtime : word);
1927:var Fch : File of char;
1928:    c1,c2,c3,c4,c5,c6,c7,c8 : char;
1929:    fsize : longint;
1930:begin
1931:  Assign(Fch,Fname);
1932:  {$I-} Reset(Fch) {$I+} ;
1933:  if IOresult = 0 then
1934:  begin
1935:    fsize  := FileSize(Fch);
1936:    Wtime :=  fsize div 2400;
1937:    Close(Fch);
1938:  end;
1939:end;
1940:
1941:Procedure LoadDataFile(Fname:string; var ErrCode:byte);
1942:type DataFile = File of LongInt;
1943:var Idx    : word;
1944:    Attrib : byte;
1945:    Fdat   : DataFile;
1946:begin
1947:  ErrCode := 0;
1948:  Assign(Fdat,Fname);
1949:  {$I-} Reset(Fdat) {$I+} ;
1950:  if IOresult <> 0 then
1951:  begin
1952:    Sound(3000);Delay(60);NoSound;
1953:    SetBkTextColor(Red);
1954:    if IsOK(23,200,
1955:       ' File not found ! ,,    Try again ?   ',
1956:       'Error') then ErrCode := 90
1957:                else ErrCode := $FF
1958:  end else
1959:  begin
1960:    if ChangeFname then
1961:    begin
1962:      SetFileInfo(Fname,WorkTime);
1963:      SetFillStyle(InterLeaveFill,Green);
1964:      Bar(0,0,GetMaxX,GetMaxY);
1965:      TimeStart := 0.0;
1966:      BottomLimit := BaseLine-(AttenLevel*100);
1967:      SetMainGraph(MainGraph);
1968:      SetMapGraph(MapGraph);
1969:      SetTimeGraph(TimeGraph);
1970:      CountMax := WorkTime*600;
1971:      InitPlotScreen;
1972:      PlotStatus;
1973:      ChangeFname := False;
1974:    end;
1975:    DrawWin(18,200,203,228,MixedType,LightRed,Black,ShadowOff,Black,
1976:        '',ReverseType);
1977:    Attrib := ReverseType;
1978:    PutCenterText(112,214,' Loading ',Attrib,Yellow);
1979:    Idx := 1;
1980:    Mark(HeapTop);
```

เอกสารนี้สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1981:     New(Rat);
1982:     Read(Fdat,Rat^.Data);
1983:     FirstRat := Rat;
1984:     LastRat  := Rat;
1985:     while not Eof(Fdat) do
1986:     begin
1987:       New(Rat);
1988:       Inc(Idx);
1989:       Read(Fdat,Rat^.Data);
1990:       LastRat^.Next := Rat;
1991:       LastRat := Rat;
1992:       if (Idx mod 200) = 0 then
1993:       begin
1994:      if Attrib = ReverseType then Attrib := NormalType
1995:                              else Attrib := ReverseType;
1996:      PutCenterText(112,214,' Loading ',Attrib,Yellow);
1997:       end;
1998:     end;
1999:
2000:     SetBkTextColor(Black);
2001:     SetFillStyle(SolidFill,Blue);
2002:     Bar(18,200,203,228);
2003:     LastRat^.Next := nil;
2004:     Close(Fdat);
2005:   end;
2006:end;
2007:
2008:Procedure SaveDataFile(Fname : string; var ErrCode : byte);
2009:type DataFile = File of LongInt;
2010:var Icount : LongInt;
2011:     Dmax   : LongInt;
2012:     Idx    : LongInt;
2013:     Dzero  : Longint;
2014:     Attrib : byte;
2015:     DataOK : boolean;
2016:     Fdat   : DataFile;
2017:     SavWin : ScreenWindow;
2018:begin
2019:   ErrCode := 0;
2020:   SetScreenWindow(18,200,203,228,SavWin);
2021:   PushScreenWindow(SavWin);
2022:   Assign(Fdat,Fname);
2023:   {$I-} ReWrite(Fdat) {$I+} ;
2024:   if IOresult <> 0 then
2025:   begin
2026:     Sound(3000);Delay(60);NoSound;
2027:     SetBkTextColor(Red);
2028:     if IsOK(23,200,
2029:          ' Disk Write Error ! ,,      Try again ?    ',
2030:          'Error') then ErrCode := $F0 { Disk write error }
2031:                   else ErrCode := 0;
2032:     SetBkTextColor(Black);
2033:   end else
2034:   begin
2035:     Dmax := GetValueChoice(RunTimeChoice)*600;
```

เอกสารนี้สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2036:    DataOK := true;
2037:    DrawWin(18,200,203,228,MixedType,LightRed,Black,ShadowOff,Black,
2038:         '',ReverseType);
2039:    Attrib := ReverseType;
2040:    PutCenterText(112,214,' Saving ',Attrib,LightGreen);
2041:    Rat := FirstRat;
2042:    Write(Fdat,Rat^.Data);
2043:    Dzero := 0;
2044:    Icount := 1;
2045:    Idx    := 1;
2046:    Repeat
2047:      Inc(Icount);
2048:      Inc(Idx);
2049:      if DataOK then
2050:      begin
2051:        Rat := Rat^.Next;
2052:     Write(Fdat,Rat^.Data);
2053:      end else  Write(Fdat,Dzero);
2054:      if (Idx mod 200) = 0 then
2055:      begin
2056:     if Attrib = ReverseType then Attrib := NormalType
2057:                             else Attrib := ReverseType;
2058;     PutCenterText(112,214,' Saving ',Attrib,LightGreen);
2059:      end;
2060:      if Rat^.Next = nil then DataOK := False;
2061:    Until (Idx = Dmax) or ((DataOK = False) and (Idx mod 600 = 0));
2062:    SetFillStyle(SolidFill,Blue);
2063:    Bar(18,200,203,228);
2064:    Close(Fdat);
2065:    SetColor(White);
2066:  end;
2067:  PopScreenWindow(SavWin);
2068:end;
2069:
2070:Procedure RunLabNet(AcMethod:byte);
2071:var Fname : string;
2072:    ch    : char;
2073:begin
2074:  if DataExist then ClearDataBuffer;
2075:  InitLabNet(ErrCode,SerialPort);
2076:  if ErrCode = 0 then
2077:  begin
2078:    DataExist := True;
2079:    FilRat := FirstRat;
2080:    Rat := FirstRat;
2081:    Lint := GetRsData(ErrCode,SerialPort); { No save First RawData }
2082:    TakeRawData(ErrCode,SerialPort);       { Set First RawData }
2083:    Count := 1;
2084:    ry := Rat^.Data*1.0;
2085:    CalBLine := ry;
2086:    MoveToXY(0.0,ry,Pmain);
2087:    MoveToXY(0.0,ry,Pmap);
2088:    PutTextXY(524,190,'Lev:',NormalType,Yellow);
2089:    Repeat
2090:      if Filter = FilterOn then DataFilter;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2091:      FindPeak;
2092:      if (Count mod 3) = 0 then
2093:        WriteSignalLevel(Rat^.Data);
2094:      rx := Count*0.1;
2095:      ry := Rat^.Data*1.0;
2096:      PlotPeakMark;
2097:      PlotLineTo(rx,ry,Pmain,MainGraph);
2098:      PlotLineTo(rx,ry,Pmap,MapGraph);
2099:      MoveToXY(rx,ry,Pmain);
2100:      MoveToXY(rx,ry,Pmap);
2101:      Inc(Count);
2102:      Rat := Rat^.Next;
2103:      TakeRawData(ErrCode,SerialPort);
2104:      CalBLine := CalBLine+Rat^.Data;
2105:      If rx > TimeFinish then  PlotNewPage;
2106:      Until ((Rat^.Next = nil) or (Count >= CountMax)) or KeyPressed;
2107:      Rat^.Next := nil;
2108:      If ActiveMethod = Method_C then
2109:      begin
2110:        CalBLine := CalBLine/Count;
2111:        BaseLine := Round(CalBLine);
2112:        Str(BaseLine,BLst);
2113:        BaseLineChoice^.Lib := BLst;
2114:      end;
2115:      SendBreak(SerialPort);
2116:      LinkActiveOn(SerialPort);
2117:      Fname  := GetDataFileName;
2118:      if not (Count >= CountMax) then ch := ReadKey;
2119:      SetBkTextColor(Red);
2120:      if IsOK(30,200,' Save Data File ','') then
2121:      begin
2122:        Repeat
2123:          ErrCode := 0;
2124:       InputFileName(130,40,Fname);
2125:       SetBkTextColor(Red);
2126:       if FileExist(Fname) then
2127:       begin
2128:         SetBkTextColor(Red);
2129:         if IsOK(10,200,' File Already Exists ,,      OVERWRITE      ',
2130:                 '') then SaveDataFile(Fname,ErrCode) else ErrCode := 1;
2131:       end else SaveDataFile(Fname,ErrCode);
2132:        Until ErrCode = 0;
2133:      end;
2134:      SetBkTextColor(Black);
2135:      SetDataFileName(Fname);
2136:      Readln;
2137:  end;   { end if ErrCode }
2138:end;
2139:
2140:Var Cnt : Longint;
2141:
2142:Procedure RunFile(AcMethod:byte);
2143:var Fname : string;
2144:begin
2145:  Fname  := GetDataFileName;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2146:  if DataExist then ClearDataBuffer;
2147:  Repeat
2148:    InputFileName(130,40,Fname);
2149:    LoadDataFile(Fname,ErrCode);
2150:  Until (ErrCode = 0) or (ErrCode = $FF);
2151:  FilRat := FirstRat;
2152:
2153:  if ErrCode = 0 then
2154:  begin
2155:    DataExist := True;
2156:    SetDataFileName(Fname);
2157:    Rat := FirstRat;
2158:    Count := 1;
2159:    ry := Rat^.Data*1.0;
2160:    CalBLine := ry;
2161:    MoveToXY(0.0,ry,Pmain);
2162:    MoveToXY(0.0,ry,Pmap);
2163:    PutTextXY(524,190,'Lev:',NormalType,LightRed);
2164:    Repeat
2165:      if Filter = FilterOn then DataFilter;
2166:      FindPeak;
2167:      if (count mod 3) = 0 then
2168:      WriteSignalLevel(Rat^.Data);
2169:      rx := Count*0.1;
2170:      ry := Rat^.Data*1.0;
2171:      PlotPeakMark;
2172:      PlotLineTo(rx,ry,Pmain,MainGraph);
2173:      PlotLineTo(rx,ry,Pmap,MapGraph);
2174:      MoveToXY(rx,ry,Pmain);
2175:      MoveToXY(rx,ry,Pmap);
2176:      Inc(Count);
2177:      Rat := Rat^.Next;
2178:      if Rat^.Data <> 0 then Cnt := Count;
2179:      CalBLine := CalBLine+Rat^.Data;
2180:      If rx > TimeFinish then  PlotNewPage;
2181:    Until ((Rat^.Next = nil) or (Count >= CountMax)) or KeyPressed;
2182:    If ActiveMethod = Method_C then
2183:    begin
2184:      CalBLine := CalBLine/Cnt;
2185:      BaseLine := Round(CalBLine);
2186:      Str(BaseLine,BLst);
2187:      BaseLineChoice^.Lib := BLst;
2188:    end;
2189:    If ActiveMethod = Method_B then PeakReportMtd_B;
2190:    Readln;
2191:  end;   { end if ErrCode }
2192:end;
2193:
2194:Procedure RunGoMenu;
2195:var Fname : string;
2196:begin
2197:  DrawCommandMenu(RunMenu_,RunOn);
2198:  ActiveMode     := GetValueChoice(ModeChoice);
2199:  MenuOn := False;
2200:  SetValueChoice;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2201:  Fname := GetDataFileName;
2202:  if ActiveMode = FileMode then SetFileInfo(Fname,WorkTime)
2203:                          else SetLabNetInfo(WorkTime);
2204:  SetFillStyle(InterLeaveFill,Green);
2205:  Bar(0,0,GetMaxX,GetMaxY);
2206:  TimeStart := 0.0;
2207:  BottomLimit := BaseLine-(AttenLevel*100);
2208:  SetMainGraph(MainGraph);
2209:  SetMapGraph(MapGraph);
2210:  SetTimeGraph(TimeGraph);
2211:  CountMax := WorkTime*600;
2212:  InitPlotScreen;
2213:  PlotStatus;
2214:  ChangeFname := False;
2215:  Case ActiveMode of
2216:     LabNetMode : RunLabNet(ActiveMethod);
2217:     FileMode   : RunFile(ActiveMethod);
2218:  end;
2219:  DrawMainWindow;
2220:  DrawSubWindow;
2221:  DrawLOGO;
2222:  DrawCommandWindow;
2223:  For i := ModeMenu_ to QuitMenu_ do
2224:    DrawCommandMenu(i,ShowOn);
2225:  DrawCommandMenu(RunMenu_,ShowOn);
2226:  DrawCommandMenu(RunMenu_,SelectOn);
2227:  WriteCurrentDate;
2228:  DrawTimeWindow;
2229:  WriteCurrentTime;
2230:  DrawStatusWindow;
2231:  DrawInfoWindow;
2232:  WriteCurrentDirectory;
2233:  GetFreeMemory;
2234:  ShowMainStatus;
2235:  MenuOn := True;
2236:end;
2237:
2238:(* ------------------------ Report Menu Section ---------------------- *)
2239:
2240:Var Pprint  : PlotInfo;
2241:    PRt     : real;
2242:
2243:Procedure NextPeak;
2244:begin
2245:  if Pek^.Next <> nil then  Pek := Pek^.Next;
2246:end;
2247:
2248:Procedure WritePRt(rx,ry : real; XRt : real);
2249:var x,y : word;
2250:    st  : string;
2251:begin
2252:  x := GetGraphPointX(rx,PrintGraph);
2253:  y := GetGraphPointY(ry,PrintGraph);
2254:  Xrt := Xrt/60.0;
2255:  Str(XRt:5:2,st);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2256:  SetTextStyle(DefaultFont,VertDir,1);
2257:  SetTextJustify(RightText,BottomText);
2258:  OutTextXY(x+4,y-6,st);
2259:  SetTextJustify(LeftText,TopText);
2260:  SetTextStyle(DefaultFont,HorizDir,1);
2261:end;
2262:
2263:Procedure PrintPeakReportMtd_B;
2264:var i,j,k : integer;
2265:      St,PkRep : string;
2266:begin
2267:  Writeln(LST,'Name              RT              PPM');
2268:  for j := 1 to PrefMax do
2269:  begin
2270:    if Pidf[j].Id = 1 then
2271:    begin
2272:      Pidf[j].Conc := Pidf[j].Area/Pidf[j].Rf;
2273:      PkRep := Pidf[j].Name;
2274:      i := length(PkRep);
2275:      if i < 15 then for k := 1 to (15-i) do PkRep := PkRep+' ';
2276:      Str(Pidf[j].Rt/60:5:2,St);
2277:      PkRep := PkRep+St;
2278:      Str(Pidf[j].Conc:9:2,St);
2279:      PkRep := PkRep+'      '+St;
2280:      Writeln(LST,PkRep);
2281:    end;
2282:  end;
2283:end;
2284:
2285:Procedure RunReportMenu;
2286:begin
2287:  DrawCommandMenu(ReportMenu_,RunOn);
2288:  ClearMainScreen;
2289:  if DataExist then
2290:  begin
2291:    SetFillStyle(SolidFill,Black);
2292:    Bar(0,0,GetMaxX,GetMaxY);
2293:    TimeStart := 0.0;
2294:    BottomLimit := BaseLine-(AttenLevel*100);
2295:    SetPrintGraph(PrintGraph);
2296:  { PlotGraphWindow(PrintGraph); }
2297:    PlotGrid(PrintGraph);
2298:    PlotScale(PrintGraph);
2299:    Rat := FirstRat;
2300:    Count := 1;
2301:    ry := Rat^.Data*1.0;
2302:    MoveToXY(0.0,ry,Pprint);
2303:    Repeat
2304:      rx := Count*0.1;
2305:      ry := Rat^.Data*1.0;
2306:      PlotLineTo(rx,ry,Pprint,PrintGraph);
2307:      MoveToXY(rx,ry,Pprint);
2308:      Inc(Count);
2309:      Rat := Rat^.Next;
2310:      if Rat^.Data <> 0 then Cnt := Count;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2311:    Until ((Rat^.Next = nil) or (Count >= CountMax)) or KeyPressed;
2312:    Pek := FirstPek;
2313:    if Pek^.Rt <> 0.0 then
2314:    begin
2315:      Repeat
2316:     WritePRt(Pek^.Rt,Pek^.Ht+40.0,Pek^.Rt);
2317:     NextPeak;
2318:      Until Pek^.Next = nil;
2319:    end;
2320:    MenuOn := False;
2321:    if IsOK(10,200,' Print Report ','') then
2322:    begin
2323:      VCopy(4);
2324:      Pek := FirstPek;
2325:      Count := 0;
2326:      writeln;
2327:      writeln(LST,'Peak                RT              AREA');
2328:      if Pek^.Rt <> 0.0 then
2329:      begin
2330:     Repeat
2331:       Inc(Count);
2332:       Writeln(Lst,Count:3,'              ',(Pek^.Rt/60):5:2,'       ',
2333:               Pek^.Area:10);
2334:       NextPeak;
2335:     Until Pek^.Next = nil;
2336:      end;
2337:      Writeln(LST);
2338:      Writeln(LST,'-----------------------------------------------');
2339:      If ActiveMethod = Method_B then PrintPeakReportMtd_B;
2340:    end;
2341:  end;
2342:  MenuOn := True;
2343:  Readln;
2344:  DrawMainWindow;
2345:  DrawSubWindow;
2346:  DrawLOGO;
2347:  DrawCommandWindow;
2348:  For i := ModeMenu_ to QuitMenu_ do
2349:    DrawCommandMenu(i,ShowOn);
2350:  DrawCommandMenu(ReportMenu_,ShowOn);
2351:  DrawCommandMenu(ReportMenu_,SelectOn);
2352:  WriteCurrentDate;
2353:  DrawTimeWindow;
2354:  WriteCurrentTime;
2355:  WriteCurrentDirectory;
2356:  MenuOn := True;
2357:end;
2358:
2359:(* ------------------------ Quit Menu Section ---------------------- *)
2360:
2361:Procedure RunQuitMenu;
2362:begin
2363:  DrawCommandMenu(QuitMenu_,RunOn);
2364:  SetScreenWindow(250,105,550,225,SaveWin);
2365:  PushScreenWindow(SaveWin);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2366:  DrawWin(300,140,500,205,DoubleType,LightRed,Blue,ShadowOff,
2367:       Black,'',NormalType);
2368:  SetFillStyle(SolidFill,Yellow);
2369:  Bar(308,145,492,200);
2370:  Sound(2700); Delay(60); Nosound; Delay(80);
2371:  Sound(3800); Delay(40); Nosound;
2372:  SetBkTextColor(Blue);
2373:  PutCenterText(400,160,' QUIT PROGRAM ',ReverseType,Yellow);
2374:  SetBkTextColor(Red);
2375:  PutCenterText(400,180,' Are You Sure ? [Y/N] ',ReverseType,Yellow);
2376:  SetBkTextColor(Black);
2377:  Repeat
2378:    WriteCurrentTime;
2379:  Until KeyPressed;
2380:  Key := ReadKey;
2381:  If UpCase(Key) = 'Y' Then ExitMenu := true;
2382:  PopScreenWindow(SaveWin);
2383:  DrawCommandMenu(QuitMenu_,ShowOn);
2384:  DrawCommandMenu(QuitMenu_,SelectOn);
2385:end;
2386:
2387:(* ------------------------------ RunMenu ------------------------------ *)
2388:
2389:Procedure RunMenu(CurMenu:byte);
2390:begin
2391:  Case CurMenu Of
2392:
2393:    ModeMenu_   : RunModeMenu;
2394:    OptionMenu_ : RunOptionMenu;
2395:    MethodMenu_ : RunMethodMenu;
2396:    FileMenu_   : RunFileMenu;
2397:    RunMenu_    : RunGoMenu;
2398:    ReportMenu_ : RunReportMenu;
2399:    QuitMenu_   : RunQuitMenu;
2400:
2401:  end; { Case }
2402:  if (CurMenu <> RunMenu_) and (CurMenu <> QuitMenu_) then
2403:  begin
2404:    SetFillStyle(Solidfill,Black);
2405:    Bar(176,55,624,332);
2406:    SetFillStyle(Solidfill,White);
2407:    DrawStatusWindow;
2408:    DrawInfoWindow;
2409:    ShowMainStatus;
2410:    GetFreeMemory;
2411:  end;
2412:end;
2413:
2414:(* <<<<<<<<<<<<<<<<<<<<<<<   Main Program   >>>>>>>>>>>>>>>>>>>>>>> *)
2415:
2416:Begin
2417:  InitGraphicScreen;
2418:  GraphicsMode := True;
2419:  MenuOn := True;
2420:  DataExist := False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2421:  SetActiveScreen(0);
2422:  SetVisualScreen(0);
2423:  DisplayMenu;
2424:  SetDataDirectory;
2425:  SettingMenuChoice;
2426:  SettingValueChoice;
2427:  ProcessLinkSubMenu;
2428:  InitLinkDataFilter;
2429:  InitLinkCat;
2430:  InitPeak;
2431:  InitPeakRef;
2432:  ExitMenu := False;
2433:  ShowMainStatus;
2434:  GetFreeMemory;
2435:  Repeat
2436:    RunMenu(GetMenu);
2437:  Until ExitMenu = True;
2438:  Readln;
2439:  CloseGraph;
2440:End.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIT PART1.PAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1:Unit Part1;{For EGA Monitor}
 2:Interface
 3:
 4:Uses Crt,Dos,Graph,Utils;
 5:
 6:Var
 7:  Key     : char;
 8:
 9:Const   { Declare Keyboard code }
10:  Return_Key    = #13;
11:  Esc_Key       = #27;
12:  Tab_Key       = #9;
13:  Alt_X_Key     = #45;
14:
15:  F1_Key        = #59;
16:  F2_Key        = #60;
17:  F3_Key        = #61;
18:  F4_Key        = #62;
19:  F5_Key        = #63;
20:  F6_Key        = #64;
21:  F7_Key        = #65;
22:  F8_Key        = #66;
23:  F9_Key        = #67;
24:  F10_Key       = #68;
25:
26:  Home_Key      = #71;
27:  End_Key       = #79;
28:  PgUp_Key      = #73;
29:  PgDn_Key      = #81;
30:  Up_Key        = #72;
31:  Dn_Key        = #80;
32:  Lt_Key        = #75;
33:  Rt_Key        = #77;
34:
35:  Ins_Key       = #82;
36:  Del_Key       = #83;
37:
38: Const { ActiveMenuType }
39:  ShowOn    = 0;
40:  SelectOn  = 1;
41:  SelectOff = 2;
42:  RunOn     = 3;
43:
44:Const { MenuNumber }
45:  ModeMenu_     = 1;
46:  OptionMenu_   = 2;
47:  MethodMenu_   = 3;
48:  FileMenu_     = 4;
49:  RunMenu_     = 5;
50:  ReportMenu_   = 6;
51:  QuitMenu_    = 7;
52:
53:Type
54:  String30  = string[30];
55:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  56:Var
  57:  MenuOn        : boolean;
  58:  ExitMenu      : boolean;
  59:  NewActiveMenu : byte;
  60:  OldActiveMenu : byte;
  61:
  62:var
  63:  MainGraph  : GraphDefine;
  64:  MapGraph   : GraphDefine;
  65:  LevelGraph : GraphDefine;
  66:  TimeGraph  : GraphDefine;
  67:  PrintGraph : GraphDefine;
  68:  TimeStart  : real;
  69:  TimeFinish : real;
  70:  SavSec     : Word;
  71:
  72:Procedure InitGraphicScreen;
  73:Procedure InitMainGraph(var Gdef : GraphDefine);
  74:Procedure InitMapGraph(var Gdef : GraphDefine);
  75:Procedure InitTimeGraph(var Gdef : GraphDefine);
  76:Procedure InitPrintGraph(var Gdef : GraphDefine);
  77:Procedure DrawMainWindow;
  78:Procedure DrawSubWindow;
  79:Procedure DrawCommandWindow;
  80:Procedure DrawDateWindow;
  81:Procedure DrawTimeWindow;
  82:Procedure DrawStatusWindow;
  83:Procedure DrawInfoWindow;
  84:Procedure DrawLOGO;
  85:Procedure DrawMenuBar(x,y:word; Name:string; ActiveType:byte);
  86:Procedure DrawMenuBox(x,y:word; Name:string; ActiveType:byte);
  87:Procedure DrawCommandMenu(CurrentMenu:byte; ActiveType:byte);
  88:Procedure ClearMainScreen;
  89:Procedure WriteCurrentDate;
  90:Procedure WriteCurrentTime;
  91:Procedure WriteCurrentDirectory;
  92:Procedure ShowMsg(x,y:word; ExitMsg:string; Header:string30);
  93:Function  IsOK(x,y:word; ExitMsg:string; Header:string30) : boolean;
  94:Procedure ShowMenu(x,y:word; Nx,Ny:byte; Header:String30);
  95:
  96:Implementation
  97:
  98:Procedure InitGraphicScreen;
  99:const
 100:  PathToDrivers = '';
 101:var
 102:  Gd,Gm : integer;
 103:begin
 104:  Gd := Detect;
 105:  InitGraph(Gd,Gm,PathToDrivers);
 106:  if GraphResult <> grOK then Halt(1);
 107:end;
 108:
 109:Procedure InitMainGraph(var Gdef : GraphDefine);
 110:begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
111:  SetGraphWindow(246,202,610,452,Gdef);
112:  SetGraphLimit(0.0,-100.0,60.0,100000.0,Gdef);
113:  SetGraphGrid(2.0,1000.0,Gdef);
114:  SetGraphGridType(BorderGrid,Gdef);
115:  SetGraphAxis(AxisOff,Gdef);
116:  SetGraphScaleType(TimeTypeB,NoScale,LowRightCorner,Gdef);
117:  SetGraphColor(Yellow,Black,LightGreen,LightGreen,LightGreen,Gdef);
118:  SetGraphPlotColor(Yellow,Gdef);
119:end;
120:
121:Procedure InitMapGraph(var Gdef : GraphDefine);
122:begin
123:  SetGraphWindow(28,18,612,114,Gdef);
124:  SetGraphLimit(0.0,-1000.0,900.0,100000.0,Gdef);
125:  SetGraphGrid(100.0,500.0,Gdef);
126:  SetGraphGridType(BorderGrid,Gdef);
127:  SetGraphAxis(AxisOff,Gdef);
128:  SetGraphScaleType(NoScale,NoScale,LowRightCorner,Gdef);
129:  SetGraphColor(Yellow,Black,LightGreen,LightGreen,LightGreen,Gdef);
130:  SetGraphPlotColor(Yellow,Gdef);
131:end;
132:
133:Procedure InitTimeGraph(var Gdef : GraphDefine);
134:begin
135:  SetGraphWindow(28,114,612,114,Gdef);
136:  SetGraphLimit(0.0,0.0,900.0,10.0,Gdef);
137:  SetGraphGrid(2.0,5.0,Gdef);
138:  SetGraphGridType(BorderGrid,Gdef);
139:  SetGraphAxis(AxisOff,Gdef);
140:  SetGraphScaleType(TimeTypeB,NoScale,LowLeftCorner,Gdef);
141:  SetGraphColor(Yellow,Black,LightGreen,LightGreen,LightGreen,Gdef);
142:  SetGraphPlotColor(Yellow,Gdef);
143:end;
144:
145:Procedure InitPrintGraph(var Gdef : GraphDefine);
146:begin
147:  SetGraphWindow(28,70,590,412,Gdef);
148:  SetGraphLimit(0.0,-1000.0,900.0,100000.0,Gdef);
149:  SetGraphGrid(100.0,500.0,Gdef);
150:  SetGraphGridType(BorderGrid,Gdef);
151:  SetGraphAxis(AxisOff,Gdef);
152:  SetGraphScaleType(TimeTypeB,NoScale,LowRightCorner,Gdef);
153:  SetGraphColor(Yellow,Black,LightGreen,LightGreen,LightGreen,Gdef);
154:  SetGraphPlotColor(Yellow,Gdef);
155:end;
156:
157:Procedure DrawMainWindow;
158:begin
159:  DrawWin(0,0,GetMaxX,441,SingleType,Yellow,Black,ShadowOff,
160:          Black,'',NormalType);
161:end;
162:
163:Procedure DrawSubWindow;
164:begin
165:  SetFillStyle(InterLeaveFill,Green);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
166:  Bar(0,440,GetMaxX,GetMaxY);
167:  DrawBorder(0,439,GetMaxX,GetMaxY,SingleType,Yellow);
168:end;
169:
170:Procedure DrawCommandWindow;
171:var x1,y1,x2,y2 : integer;
172:begin
173:  x1 :=   0; y1 :=  75;
174:  x2 := 160; y2 := 436;
175:  DrawWin(x1,y1,x2,y2,SingleType,Yellow,Red,ShadowOff,Black,'',NormalType);
176:  SetFillStyle(SolidFill,Blue);
177:  Bar(x1+1,y1+1,x2-1,y2-1);
178:end;
179:
180:Procedure DrawDateWindow;
181:begin
182:  DrawWin(176,10,552,38,SingleType,Red,Red,ShadowOff,Black,'',NormalType);
183:  DrawWin(178,11,550,37,SingleType,Red,Black,ShadowOff,Black,'',NormalType);
184:end;
185:
186:Procedure DrawTimeWindow;
187:begin
188:  DrawWin(562,10,627,38,SingleType,Green,Black,ShadowOff,Black,'',NormalType);
189:  DrawWin(569,15,619,33,SingleType,Green,Black,ShadowOff,Black,'',NormalType);
190:  DrawBorder(567,14,621,34,SingleType,Green);
191:  DrawBorder(568,14,620,34,SingleType,Green);
192:  SetColor(Green);
193:  Line(562,10,567,14);
194:  Line(562,38,567,34);
195:  Line(627,10,621,14);
196:  Line(627,38,621,34);
197:  SetColor(White);
198:end;
199:
200:Procedure DrawStatusWindow;
201:begin
202:  DrawWin(176,55,624,295,SingleType,Yellow,Blue,ShadowOff,Black,
203:                         ' STATUS ',ReverseType);
204:end;
205:
206:Procedure DrawInfoWindow;
207:begin
208:  DrawWin(176,314,624,422,SingleType,Yellow,Blue,ShadowOff,Black,
209:                       ' INFORMATION ',ReverseType);
210:end;
211:
212:Procedure DrawLOGO;
213:var x1,y1,x2,y2 : integer;
214:begin
215:  x1 :=   0; y1 :=  0;
216:  x2 := 160; y2 := 75;
217:  DrawBorder(x1,y1,x2,y2,SingleType,Yellow);
218:  DrawWin(x1+10,y1+9,x2-10,y2-9,SingleType,Yellow,Blue,ShadowOff,
219:          Black,'',NormalType);
220:  SetTextStyle(DefaultFont,Horizdir,1);
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
221:  SetBkTextColor(Blue);
222:  PutCenterText(83,y1+28,'HPLC-SOFT',NormalType,Yellow);
223:  PutCenterText(83,y1+45,'Version 1.00',NormalType,Yellow);
224:  SetBkTextColor(Black);
225:end;
226:
227:Procedure DrawMenuBar(x,y:word; Name:string; ActiveType:byte);
228:begin
229: Case ActiveType of
230:
231: ShowOn    : PutTextXY(x,y,Name,NormalType,GetColor);
232: SelectOn  : PutTextXY(x,y,Name,ReverseType,GetColor);
233: SelectOff : PutTextXY(x,y,Name,NormalType,GetColor);
234: RunOn     : PutTextXY(x,y,Name,ReverseType,GetColor);
235:
236: end; { Case }
237:end;
238:
239:Procedure DrawMenuBox(x,y:word; Name:string; ActiveType:byte);
240:begin
241: Case ActiveType of
242:
243: ShowOn :
244:  begin
245:    DrawWin(x,y,x+135,y+44,SingleType,Green,Black,ShadowOff,
246:            Black,'',NormalType);
247:    DrawBorder(x+9,y+10,x+126,y+34,SingleType,Green);
248:    DrawBorder(x+10,y+10,x+125,y+34,SingleType,Green);
249:    DrawBorder(x+11,y+10,x+124,y+34,SingleType,Green);
250:    DrawBorder(x+12,y+10,x+123,y+34,SingleType,Green);
251:    PutCenterText(x+68,y+22,Name,NormalType,Yellow);
252:  end;
253:
254: SelectOn :
255:  begin
256:    PutCenterText(x+68,y+22,Name,ReverseType,Yellow);
257:  end;
258:
259: SelectOff :
260:  begin
261:    PutCenterText(x+68,y+22,Name,NormalType,Yellow);
262:  end;
263:
264: RunOn :
265:  begin
266:    DrawWin(x+3,y+3,x+132,y+41,SingleType,Green,White,ShadowOff,Black,
267:         '',NormalType);
268:    DrawWin(x+10,y+10,x+126,y+34,DoubleType,Green,Black,ShadowOff,Black,
269:         '',NormalType);
270:    PutCenterText(x+68,y+22,Name,NormalType,Yellow);
271:  end;
272:
273: end; { Case }
274:end;
275:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
276:Procedure DrawCommandMenu(CurrentMenu:byte; ActiveType:byte);
277:var Xm,Ym : word;
278:begin
279:  Xm := 14;
280:  Ym := 100;
281:
282:  Case CurrentMenu of
283:
284:    ModeMenu_   : DrawMenuBox(Xm,Ym,     '    Mode     ',ActiveType);
285:    OptionMenu_ : DrawMenuBox(Xm,Ym+44, '   Option    ',ActiveType);
286:    MethodMenu_ : DrawMenuBox(Xm,Ym+88, '   Method    ',ActiveType);
287:    FileMenu_   : DrawMenuBox(Xm,Ym+132,'    File     ',ActiveType);
288:    RunMenu_    : DrawMenuBox(Xm,Ym+176,'    Run      ',ActiveType);
289:    ReportMenu_ : DrawMenuBox(Xm,Ym+220,'   Report    ',ActiveType);
290:    QuitMenu_   : DrawMenuBox(Xm,Ym+264,'    Quit     ',ActiveType);
291:
292:  end; { Case }
293:
294:end;
295:
296:Procedure ClearMainScreen;
297:begin
298:  DrawWin(176,48,624,428,NoBorder,Green,Black,
299:       ShadowOff,Black,'',NormalType);
300:end;
301:
302:Procedure WriteCurrentDate;
303:var Year,Month,Day,DayOfWeek : word;
304:    StYear,StMonth,StDay,StDWeek : string;
305:begin
306:  GetDate(Year,Month,Day,DayOfWeek);
307:  Str(Year,StYear);
308:  Str(Day,StDay);
309:  If Day < 10 then StDay := Concat(' ',StDay);
310:
311:  Case DayOfWeek of
312:    0 : StDWeek := 'Sunday';
313:    1 : StDWeek := 'Monday';
314:    2 : StDWeek := 'Tuesday';
315:    3 : StDWeek := 'Wednesday';
316:    4 : StDWeek := 'Thursday';
317:    5 : StDWeek := 'Friday';
318:    6 : StDWeek := 'Saturday';
319:  end; { Case }
320:
321:  Case Month of
322:    1 : StMonth := 'January';
323:    2 : StMonth := 'February';
324:    3 : StMonth := 'March';
325:    4 : StMonth := 'April';
326:    5 : StMonth := 'May';
327:    6 : StMonth := 'June';
328:    7 : StMonth := 'July';
329:    8 : StMonth := 'August';
330:    9 : StMonth := 'September';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
331:     10 : StMonth := 'October';
332:     11 : StMonth := 'November';
333:     12 : StMonth := 'December';
334:   end; { Case }
335:
336:   DrawDateWindow;
337:   PutTextXY(222,22,StDWeek,NormalType,Yellow);
338:   PutTextXY(320,22,StMonth,NormalType,Yellow);
339:   PutTextXY(422,22,StDay,NormalType,Yellow);
340:   PutTextXY(440,22,',',NormalType,Yellow);
341:   PutTextXY(476,22,StYear,NormalType,Yellow);
342:end;
343:
344:Procedure WriteCurrentTime;
345:var Hr,Min,Sec,Sec100 : Word;
346:     StHr,StMin         : string;
347:     i                  : integer;
348:     SavColor           : word;
349:begin
350:   SavColor := GetColor;
351:   GetTime(Hr,Min,Sec,Sec100);
352:   if Sec <> SavSec then
353:   begin
354:     If (Sec Mod 2) = 0 then PutCenterText(594,23,':',NormalType,Yellow)
355:                        else PutCenterText(594,23,':',NormalType,Black);
356:     Str(Hr,StHr);
357:     if Hr  < 10 then StHr := Concat(' ',StHr);
358:     Str(Min,StMin);
359:     if Min < 10 Then StMin := Concat('0',StMin);
360:     if (Sec Mod 2) = 0 then PutTextXY(573,23,StHr,NormalType,Yellow);
361:     if (Sec Mod 2) = 0 then PutTextXY(600,23,StMin,NormalType,Yellow);
362:     if ((Min Mod 60) = 0) and ((Sec Mod 60) = 0) then
363:     begin
364:       For i := 1 to 4 do
365:       begin
366:         Sound(3800); Delay(50);
367:      NoSound;      Delay(100);
368:       end;
369:     end;
370:   end;
371:   SetColor(SavColor);
372:   SavSec := Sec;
373:end;
374:
375:Procedure WriteCurrentDirectory;
376:var StDir : string;
377:begin
378:   GetDir(0,StDir);
379:   DrawWin(9,447,311,471,SingleType,White,Black,ShadowOff,
380:           Black,'',NormalType);
381:   PutTextXY(18,458,StDir,NormalType,White);
382:end;
383:
384:Procedure ShowMsg(x,y:word; ExitMsg:string; Header:string30);
385:Const Yspace = 5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
386:var Ex,Ey,Tx,Ty : word;
387:    Count,i     : byte;
388:    Smax,Snum   : byte;
389:    Xmsg        : string;
390:    SavWin      : ScreenWindow;
391:    SBkTextColor : word;
392:begin
393:  Count := 1;
394:  Smax  := 0;
395:  Snum  := 0;
396:
397:  for i := 1 to Length(ExitMsg) do
398:    if ExitMsg[i] = ',' then
399:    begin
400:      inc(Count);
401:      Snum := 0;
402:    end else
403:    begin
404:      inc(Snum);
405:      if Snum > Smax then Smax := Snum;
406:    end;
407:
408:  if Smax >= 12 then Ex := x + TextWidth('X')*(Smax+4)
409:                else Ex := x + TextWidth('X')*16;
410:  Ey := y + (TextHeight('X')+Yspace)*(Count+1) + 28;
411:
412:  SetScreenWindow(x,y-4,Ex,Ey,SavWin);
413:  PushScreenWindow(SavWin);
414:
415:  DrawWin(x,y,Ex,Ey,DoubleType,White,Magenta,ShadowOff,
416:       Black,Header,ReverseType);
417:  SetColor(White);
418:  Line(x+3,Ey-18,Ex-3,Ey-18);
419:
420:  Tx := x + TextWidth('X')*2 + 1;
421:  Ty := y + 6;
422:
423:  for i := 1 to Count do
424:  begin
425:    if i = Count then Xmsg := Copy(ExitMsg,1,Smax)
426:                 else begin
427:                           Xmsg := Copy(ExitMsg,1,Pos(',',ExitMsg)-1);
428:                           Delete(ExitMsg,1,Pos(',',ExitMsg));
429:                          end;
430:    Ty := Ty + TextHeight('X') + Yspace;
431:    SBkTextColor := BkTextColor;
432:    SetBkTextColor(Magenta);
433:    SetColor(Yellow);
434:    PutTextXY(Tx,Ty,Xmsg,NormalType,GetColor);
435:    SetBkTextColor(SBkTextColor);
436:  end;
437:
438:  DrawMenuBar(x+((Ex-x) div 2)-8,Ey-10,'OK',SelectOn);
439:  if MenuOn then
440:  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
441:  Repeat
442:    WriteCurrentTime;
443:  Until KeyPressed;
444:  end;
445:  Readln;
446:  PopScreenWindow(SavWin);
447:end;
448:
449:Function IsOK(x,y:word; ExitMsg:string; Header:string30) : boolean;
450:Const Yspace  = 5;
451:var Ex,Ey,Tx,Ty : word;
452:    Count,i     : byte;
453:    Smax,Snum   : byte;
454:    Xmsg        : string;
455:    OK          : boolean;
456:    SavWin      : ScreenWindow;
457:begin
458:  Count := 1;
459:  Smax  := 0;
460:  Snum  := 0;
461:
462:  for i := 1 to Length(ExitMsg) do
463:    if ExitMsg[i] = ',' then
464:    begin
465:      inc(Count);
466:      Snum := 0;
467:    end else
468:    begin
469:      inc(Snum);
470:      if Snum > Smax then Smax := Snum;
471:    end;
472:
473:  if Smax >= 10 then Ex := x + TextWidth('X')*(Smax+4)
474:                else Ex := x + TextWidth('X')*14;
475:  Ey := y + (TextHeight('X')+Yspace)*(Count+1) + 30;
476:
477:  SetScreenWindow(x,y-4,Ex,Ey,SavWin);
478:  PushScreenWindow(SavWin);
479:
480:  DrawWin(x,y,Ex,Ey,ThickType,LightGreen,Red,ShadowOff,Black,
481:          Header,ReverseType);
482:  SetColor(LightGreen);
483:  Line(x+3,Ey-18,Ex-3,Ey-18);
484:
485:  Tx := x + TextWidth('X')*2 + 1;
486:  Ty := y + 8;
487:
488:  for i := 1 to Count do
489:  begin
490:    if i = Count then Xmsg := Copy(ExitMsg,1,Smax)
491:                 else begin
492:                        Xmsg := Copy(ExitMsg,1,Pos(',',ExitMsg)-1);
493:                        Delete(ExitMsg,1,Pos(',',ExitMsg));
494:                      end;
495:    Ty := Ty + TextHeight('X') + Yspace;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
496:    PutTextXY(Tx,Ty,Xmsg,NormalType,GetColor);
497:  end;
498:
499:  OK := True;
500:  DrawMenuBar(x+((Ex-x) div 4)-8,Ey-10,'OK',SelectOn);
501:  DrawMenuBar(x+((Ex-x) div 4)*3-24,Ey-10,'CANCEL',SelectOff);
502:  Repeat
503:    if MenuOn then
504:    begin
505:      Repeat
506:     WriteCurrentTime;
507:      Until KeyPressed;
508:    end;
509:    Key := ReadKey;
510:    If (Key = Lt_Key) or (Key = Rt_Key) or (Key = ' ') then OK := not OK;
511:    If OK then
512:    begin
513:      DrawMenuBar(x+((Ex-x) div 4)*3-24,Ey-10,'CANCEL',SelectOff);
514:      DrawMenuBar(x+((Ex-x) div 4)-8,Ey-10,'OK',SelectOn);
515:    end else
516:  begin
517:    DrawMenuBar(x+((Ex-x) div 4)-8,Ey-10,'OK',SelectOff);
518:    DrawMenuBar(x+((Ex-x) div 4)*3-24,Ey-10,'CANCEL',SelectOn);
519:  end;
520:  Until Key = Return_Key;
521:
522:  PopScreenWindow(SavWin);
523:
524:  IsOK := OK;
525:end;
526:
527:Procedure ShowMenu(x,y:word; Nx,Ny:byte; Header:String30);
528:Const Yspace  = 5;
529:var Ex,Ey  : word;
530:begin
531:  Ex := x + TextWidth('X')*(Nx+4);
532:  Ey := y + (TextHeight('X')+Yspace)*(Ny+1) + 28;
533:
534:  DrawWin(x,y,Ex,Ey,SingleType,Yellow,Blue,ShadowOff,Black,'',ReverseType);
535:  DrawWin(x+3,y+2,Ex-3,y+15,SingleType,Yellow,Red,ShadowOff,Black,
536:      '',NormalType);
537:
538:  DrawBorder(x+3,y+17,Ex-3,Ey-2,SingleType,Yellow);
539:
540:  SetColor(Yellow);
541:  Line(x+3,Ey-18,Ex-3,Ey-18);
542:
543:  PutCenterText(x+((Ex-x) div 2),y+8,Header,ReverseType,Yellow);
544:end;
545:
546:begin
547:  MenuOn := True;
548:  InitMainGraph(MainGraph);
549:  InitMapGraph(MapGraph);
550:  InitTimeGraph(TimeGraph);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
551:  InitPrintGraph(PrintGraph);
552:end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIT UTILS.PAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1:Unit Utils;
 2:Interface
 3:
 4:Uses Crt,Dos,Graph;
 5:
 6:(* ------------------------ Screen Util ---------------------------- *)
 7:
 8:type ScreenWindow = Record
 9:                      Wp    : pointer;
10:                      xi,yi : integer;
11:                      xe,ye : integer;
12:                    end;
13:
14:var  SaveWin      : ScreenWindow;
15:     ActiveScreen : byte;
16:     VisualScreen : byte;
17:
18:procedure SetScreenWindow(x1,y1,x2,y2:integer;Var Sw:ScreenWindow);
19:procedure PushScreenWindow(Var Sw:ScreenWindow);
20:procedure PopScreenWindow(Sw:ScreenWindow);
21:procedure SetActiveScreen(Ap:byte);
22:procedure SetVisualScreen(Vp:byte);
23:Procedure SwapScreen;
24:procedure XeroxScreen;
25:
26:(* ------------------------------ Win Util ------------------------- *)
27:
28:Const   { Window Type }
29:  NoBorder   = 0;
30:  SingleType = 1;
31:  DoubleType = 2;
32:  ThickType  = 3;
33:  MixedType  = 4;
34:
35:Const   { Shadow Status }
36:  ShadowOn  = True;
37:  ShadowOff = False;
38:
39:Const   { Header Type }
40:  NormalType  = 0;
41:  ReverseType = 1;
42:
43:Const  { Active Status }
44:  ActiveOn  = True;
45:  ActiveOff = False;
46:
47:Var BkTextColor : word;
48:
49:Procedure DrawBorder(x1,y1,x2,y2:word; BorderType:byte; BorderColor:word);
50:Procedure ClearWindow(x1,y1,x2,y2:word; GroundColor:word);
51:Procedure FillShadow(x1,y1,x2,y2:word; ShadowColor:word);
52:Procedure SetBkTextColor(BkTColor:word);
53:Procedure PutTextXY(x,y:word; TextMsg:string; TextType:byte; TextColor:word);
54:Procedure PutCenterText(Xc,y: word; TextMsg:string; TextType:byte;
55:                        TextColor : word);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  56:Procedure DrawWin(x1,y1,x2,y2:word; BorderType:byte;      BorderColor:word;
  57:                  GroundColor:word; ShadowStatus:Boolean; ShadowColor:word;
  58:                  HeaderMsg:string; HeaderType:byte);
  59:
  60:(* ------------------------- Plot Util ---------------------------- *)
  61:
  62:Const  { GridType }
  63:  BorderGrid   = 0;
  64:  SquareGrid   = 1;
  65:
  66:Const { Axis Status }
  67:  AxisOff      = False;
  68:  AxisOn       = True;
  69:
  70:Const { ScaleXYType }
  71:  NoScale      = 0;
  72:  IntType      = 1;
  73:  RealType     = 2;
  74:  ExpType      = 3;
  75:  TimeTypeA    = 4;
  76:  TimeTypeB    = 5;
  77:
  78:Const { ScaleStyle }
  79:  LowLeftCorner  = 1;
  80:  LowRightCorner = 2;
  81:
  82:Type
  83:  GraphWindow  = record
  84:                   Wx1,Wy1 : word;   { Upper  Left  Corner }
  85:                   Wx2,Wy2 : word;   { Lowwer Right Corner }
  86:                   WinColor  : word;
  87:                   WGndColor : word;
  88:                 end;
  89:
  90:  GraphLimit   = record
  91:                   Lx1,Ly1 : real;   { Minimum Limit }
  92:                   Lx2,Ly2 : real;   { Maximum Limit }
  93:                 end;
  94:
  95:  GridXY       = record
  96:                   Gdx,Gdy : real;
  97:                   GType : byte;
  98:                   GridColor : word;
  99:                 end;
 100:
 101:  AxisXY       = record
 102:                   Stat      : boolean;
 103:                   AxisColor : word;
 104:                end;
 105:
 106:  ScaleType    = record
 107:                   Xtype : byte;
 108:                   Ytype : byte;
 109:                   Style : byte;
 110:                   ScaleColor : word;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
111:                end;
112:
113:  GraphDefine  = record
114:                   Win   : GraphWindow;
115:                   Lim   : GraphLimit;
116:                   Grid  : GridXY;
117:                   Axis  : AxisXY;
118:                   Stype : ScaleType;
119:                   PlotColor : word;
120:                 end;
121:
122:  PlotInfo     = record
123:                   Ox,Oy  : real;      { Use in PlotLineTo }
124:                 end;
125:
126:(* ---------------------< SetGraph Routine >---------------------- *)
127:
128:Procedure SetGraphWindow(x1,y1,x2,y2 : word; var Gdef : GraphDefine);
129:Procedure SetGraphLimit(x1,y1,x2,y2 : real; var Gdef : GraphDefine);
130:Procedure SetGraphGrid(GridX,GridY : real; var Gdef : GraphDefine);
131:Procedure SetGraphGridType(GridType : byte; var Gdef : GraphDefine);
132:Procedure SetGraphScaleType(SxType,SyType,Ssty : byte;
133:                            var Gdef : GraphDefine);
134:Procedure SetGraphAxis(AxisStatus : boolean; var Gdef : GraphDefine);
135:Procedure SetGraphPlotColor(Pcolor : byte; var Gdef : GraphDefine);
136:Procedure SetGraphColor(WinC,WGndC,GridC,AxisC,ScaleC : word;
137:                        var Gdef : GraphDefine);
138:
139:(* ---------------------< GetGraph Routine >---------------------- *)
140:
141:Function GetAxisStatus(Gdef : GraphDefine) : boolean;
142:Function GetGraphPointX(rx : real; Gdef : GraphDefine) : word;
143:Function GetGraphPointY(ry : real; Gdef : GraphDefine) : word;
144:
145:(* ----------------------< Plot Routine >------------------------ *)
146:
147:Procedure PlotGraphWindow(Gdef : GraphDefine);
148:Procedure ClearGraphWindow(Gdef : GraphDefine);
149:Procedure PlotGrid(Gdef : GraphDefine);
150:Procedure PlotAxis(Gdef : GraphDefine);
151:Procedure PlotScale(Gdef : GraphDefine);
152:
153:(* ---------------------< PlotData Routine >---------------------- *)
154:
155:Procedure MoveToXY(Ix,Iy : real; Var Pinfo : PlotInfo);
156:Procedure PlotXY(rx,ry : real; Gdef : GraphDefine);
157:Procedure ClearXY(rx,ry : real; Gdef : GraphDefine);
158:Procedure PlotLineTo(x,y : real; Pinfo : PlotInfo; Gdef : GraphDefine);
159:Procedure ClearLineTo(x,y : real; Pinfo : PlotInfo; Gdef : GraphDefine);
160:Procedure PlotLine(rx1,ry1,rx2,ry2 : real; Gdef : GraphDefine);
161:Procedure ClearLine(rx1,ry1,rx2,ry2 : real; Gdef : GraphDefine);
162:Procedure PlotArea(rx1,ry1,rx2,ry2,BaseLn : real; Gdef : GraphDefine);
163:
164:(* ------------------------- Gread Util ------------------------- *)
165:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
166:Const
167:  GraphicsMode : boolean = False;
168:
169:Procedure ReadGraphString(var S :String );
170:Procedure EditGraphString(var S :String );
171:
172:(* ------------------------- RS232 Util ------------------------- *)
173:
174:Const
175:  LinkActive = $20;
176:  TgActive   = $53;
177:  XOn        = $11;
178:  XOff       = $13;
179:  Break      = $00;
180:
181:Var
182:  Regs       : registers;
183:
184:Procedure InitRS232(Com : word);
185:Procedure RsSend(Datas : byte; Com : word);
186:Procedure RsStatus(Var Lob,Hib : byte; Com : word);
187:Procedure RsRecive(Var Data : byte;Var ErrCode:byte; Com : word);
188:Procedure LinkActiveOn(Com : Word);
189:Procedure TriggerActiveOn(Com : word);
190:Procedure SendXOn(Com : Word);
191:Procedure SendXOff(Com : Word);
192:Procedure SendBreak(Com : Word);
193:Function  GetRsData(Var ErrCode : byte; Com : word) : LongInt;
194:
195:Implementation
196:
197:(* ------------------------- Screen Util ------------------------- *)
198:
199:procedure SetScreenWindow(x1,y1,x2,y2:integer;Var Sw:ScreenWindow);
200:begin
201:  With Sw do
202:  begin
203:    xi := x1;
204:    yi := y1;
205:    xe := x2;
206:    ye := y2;
207:  end;
208:end;
209:
210:procedure PushScreenWindow(Var Sw:ScreenWindow);
211:begin
212:  GetMem(Sw.Wp,ImageSize(Sw.xi,Sw.yi,Sw.xe,Sw.ye));
213:  GetImage(Sw.xi,Sw.yi,Sw.xe,Sw.ye,Sw.Wp^);
214:end;
215:
216:procedure PopScreenWindow(Sw:ScreenWindow);
217:begin
218:  PutImage(Sw.xi,Sw.yi,Sw.Wp^,NormalPut);
219:  FreeMem(Sw.Wp,ImageSize(Sw.xi,Sw.yi,Sw.xe,Sw.ye));
220:end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
221:
222:procedure SetActiveScreen(Ap:byte);
223:begin
224:  SetActivePage(Ap);
225:  ActiveScreen := Ap;
226:end;
227:
228:procedure SetVisualScreen(Vp:byte);
229:begin
230:  SetVisualPage(Vp);
231:  VisualScreen := Vp;
232:end;
233:
234:Procedure SwapScreen;
235:begin
236:  if VisualScreen = 1 then
237:  begin
238:    SetVisualScreen(0);
239:    SetActiveScreen(0);
240:  end
241:  else
242:  begin
243:    SetVisualScreen(1);
244:    SetActiveScreen(1);
245:  end;
246:end;
247:
248:procedure XeroxScreen;
249:begin
250:  if VisualScreen = 1 then
251:  begin
252:    SetActiveScreen(1);
253:    SetScreenWindow(0,0,GetMaxX,GetMaxY,SaveWin);
254:    PushScreenWindow(SaveWin);
255:    SetActiveScreen(0);
256:    PopScreenWindow(SaveWin);
257:    SetActiveScreen(1);
258:  end
259:  else
260:  begin
261:    SetActiveScreen(0);
262:    SetScreenWindow(0,0,GetMaxX,GetMaxY,SaveWin);
263:    PushScreenWindow(SaveWin);
264:    SetActiveScreen(1);
265:    PopScreenWindow(SaveWin);
266:    SetActiveScreen(0);
267:  end;
268:end;
269:
270:(* ---------------------------- Win Util ------------------------------ *)
271:
272:Procedure DrawBorder(x1,y1,x2,y2 : word; BorderType : byte;
273:                     BorderColor : word);
274:begin
275:  SetColor(BorderColor);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
276:  Case BorderType of
277:
278:      NoBorder   : begin end;
279:
280:      SingleType : begin
281:                       Rectangle(x1,y1,x2,y2);
282:                   end;
283:
284:      ThickType  : begin
285:                       Rectangle(x1,y1,x2,y2);
286:                       Rectangle(x1+1,y1+1,x2-1,y2-1);
287:                       Rectangle(x1+2,y1+2,x2-2,y2-2);
288:                       Rectangle(x1+3,y1+2,x2-3,y2-2);
289:                   end;
290:
291:      DoubleType : begin
292:                       Rectangle(x1,y1,x2,y2);
293:                       Setcolor(GetBKColor);
294:                       Rectangle(x1+1,y1+1,x2-1,y2-1);
295:                       Rectangle(x1+2,y1+2,x2-2,y2-2);
296:                       SetColor(BorderColor);
297:                       Rectangle(x1+3,y1+2,x2-3,y2-2);
298:                   end;
299:
300:      MixedType  : begin
301:                       Rectangle(x1,y1,x2,y2);
302:                       Rectangle(x1+1,y1+1,x2-1,y2-1);
303:                       Rectangle(x1+2,y1+1,x2-2,y2-1);
304:                       Rectangle(x1+3,y1+2,x2-3,y2-2);
305:                       Rectangle(x1+6,y1+4,x2-6,y2-4);
306:                   end;
307:  end;
308:end;
309:
310:Procedure ClearWindow(x1,y1,x2,y2 : word; GroundColor : word);
311:var FillInfo : FillSettingsType;
312:begin
313:  GetFillSettings(FillInfo);
314:  SetFillStyle(SolidFill,GroundColor);
315:  Bar(x1,y1,x2,y2);
316:  SetFillStyle(FillInfo.Pattern,FillInfo.Color);
317:end;
318:
319:Procedure FillShadow(x1,y1,x2,y2 : word; ShadowColor : word);
320:var FillInfo : FillSettingsType;
321:begin
322:  GetFillSettings(FillInfo);
323:  SetFillStyle(SolidFill,ShadowColor);
324:  Bar(x2,y1+5,x2+8,y2+5);
325:  Bar(x1+8,y2,x2+8,y2+5);
326:  SetFillStyle(FillInfo.Pattern,FillInfo.Color);
327:end;
328:
329:Procedure SetBkTextColor(BkTColor:word);
330:begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
331:  BkTextColor := BkTColor;
332:end;
333:
334:Procedure PutTextXY(x,y : word; TextMsg : string; TextType : byte;
335:                    TextColor : word);
336:var FillInfo : FillSettingsType;
337:    Twidth   : word;
338:    Theight  : word;
339:begin
340:  if TextMsg <> '' then
341:  begin
342:    GetFillSettings(FillInfo);
343:    SetTextJustify(LeftText,TopText);
344:    SetTextStyle(DefaultFont,HorizDir,1);
345:    Twidth  := Length(TextMsg)*TextWidth('X');
346:    Theight := TextHeight('X');
347:    if TextType = NormalType then
348:    begin
349:      SetFillStyle(SolidFill,BkTextColor);
350:      Bar(x-2,y-4,x+Twidth+1,y+4);
351:      SetColor(TextColor);
352:      OutTextXY(x,y-3,TextMsg);
353:    end else { TextType = ReverseType }
354:    begin
355:      SetFillStyle(SolidFill,TextColor);
356:      Bar(x-2,y-4,x+Twidth+1,y+4);
357:      SetColor(BkTextColor);
358:      OutTextXY(x,y-3,TextMsg);
359:      SetColor(TextColor);
360:    end;
361:    SetFillStyle(FillInfo.Pattern,FillInfo.Color);
362:  end;
363:end;
364:
365:Procedure PutCenterText(Xc,y : word; TextMsg : string; TextType : byte;
366:                         TextColor : word);
367:var x : word;
368:begin
369:  x := Xc - ((Length(TextMsg)*TextWidth('X')) div 2);
370:  PutTextXY(x,y,TextMsg,TextType,TextColor);
371:end;
372:
373:Procedure DrawWin(x1,y1,x2,y2:word; BorderType:byte;      BorderColor:word;
374:                  GroundColor:word; ShadowStatus:Boolean; ShadowColor:word;
375:                  HeaderMsg:string; HeaderType:byte);
376:begin
377:  If ShadowStatus = ShadowOn then
378:     FillShadow(x1,y1,x2,y2,ShadowColor);
379:  ClearWindow(x1,y1,x2,y2,GroundColor);
380:  DrawBorder(x1,y1,x2,y2,BorderType,BorderColor);
381:  PutCenterText(x1 + ((x2-x1) div 2),y1,
382:                HeaderMsg,HeaderType,BorderColor);
383:end;
384:
385:(* ---------------------------- Plot Util ---------------------------- *)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
386:
387:(* ----------------------< SetGraph Routine >---------------------- *)
388:
389:Procedure SetGraphWindow(x1,y1,x2,y2 : word; var Gdef : GraphDefine);
390:begin
391:  with Gdef do
392:  begin
393:    Win.Wx1 := x1;
394:    Win.Wy1 := y1;
395:    Win.Wx2 := x2;
396:    Win.Wy2 := y2;
397:  end;
398:end;
399:
400:Procedure SetGraphLimit(x1,y1,x2,y2 : real; var Gdef : GraphDefine);
401:begin
402:  with Gdef do
403:  begin
404:    Lim.Lx1 := x1;
405:    Lim.Ly1 := y1;
406:    Lim.Lx2 := x2;
407:    Lim.Ly2 := y2;
408:  end;
409:end;
410:
411:Procedure SetGraphGrid(GridX,GridY : real; var Gdef : GraphDefine);
412:begin
413:  with Gdef do
414:  begin
415:    Grid.Gdx := GridX;
416:    Grid.Gdy := GridY;
417:  end;
418:end;
419:
420:Procedure SetGraphGridType(GridType : byte; var Gdef : GraphDefine);
421:begin
422:  Gdef.Grid.Gtype := GridType;
423:end;
424:
425:Procedure SetGraphScaleType(SxType,SyType,Ssty : byte;
426:                            var Gdef : GraphDefine);
427:begin
428:  with Gdef do
429:  begin
430:    Stype.Xtype := SxType;
431:    Stype.Ytype := SyType;
432:    Stype.Style := Ssty;
433:  end;
434:end;
435:
436:Procedure SetGraphAxis(AxisStatus : boolean; var Gdef : GraphDefine);
437:begin
438:  Gdef.Axis.Stat := AxisStatus;
439:end;
440:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
441:
442:Procedure SetGraphPlotColor(Pcolor : byte; var Gdef : GraphDefine);
443:begin
444:  Gdef.PlotColor := Pcolor;
445:end;
446:
447:Procedure SetGraphColor(WinC,WGndC,GridC,AxisC,ScaleC : word;
448:                         var Gdef : GraphDefine);
449:begin
450:  with Gdef do
451:  begin
452:    Win.WinColor     := WinC;
453:    Win.WGndColor    := WGndC;
454:    Grid.GridColor   := GridC;
455:    Axis.AxisColor   := AxisC;
456:    Stype.ScaleColor := ScaleC;
457:  end;
458:end;
459:
460:(* ----------------------< GetGraph Routine >---------------------- *)
461:
462:Function GetAxisStatus(Gdef : GraphDefine) : boolean;
463:begin
464:  GetAxisStatus := Gdef.Axis.Stat;
465:end;
466:
467:Function GetGraphPointX(rx : real; Gdef : GraphDefine) : word;
468:var Px  : integer;
469:    Spx,r : real;
470:begin
471:  with Gdef do
472:  begin
473:    Spx := (Win.Wx2-Win.Wx1)/(Lim.Lx2-Lim.Lx1);
474:    {$R-}
475:    r  := Spx*(rx - Lim.Lx1);
476:    Px := Win.Wx1 + round(r);
477:    {$R+}
478:    If Px < Win.Wx1 then Px := Win.Wx1;
479:    If Px > Win.Wx2 then Px := Win.Wx2;
480:    GetGraphPointX := Px;
481:  end;
482:end;
483:
484:Function GetGraphPointY(ry : real; Gdef : GraphDefine) : word;
485:var Py  : integer;
486:    Spy : real;
487:begin
488:  with Gdef do
489:  begin
490:    Spy := (Win.Wy2-Win.Wy1)/(Lim.Ly2-Lim.Ly1);
491:    {$R-}
492:    Py := Win.Wy2 - Round((Spy * (ry - Lim.Ly1)));
493:    {$R+}
494:    If Py < Win.Wy1 then Py := Win.Wy1;
495:    If Py > Win.Wy2 then Py := Win.Wy2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
496:    GetGraphPointY := Py;
497:  end;
498:end;
499:
500:(* ----------------------< Plot SubRoutine >---------------------- *)
501:
502:Function GridExist(Lmin,Lmax : real; dg : real) : boolean;
503:begin
504:  if dg > 0 then
505:    if ((Lmax-Lmin)/dg) > 1.0 then  GridExist := True
506:
507:end;
508:
509:Function GetGridStart(Lmin : real; dg : real) : real;
510:begin
511:  GetGridStart := Lmin;
512:end;
513:
514:Procedure PlotXBorderGrid(Gdef : GraphDefine);
515:var x,y    : word;
516:    GXplot : real;
517:begin
518:  with Gdef do
519:  if GridExist(Lim.Lx1,Lim.Lx2,Grid.Gdx) then
520:  begin
521:    GXplot := GetGridStart(Lim.Lx1,Grid.Gdx);
522:    SetColor(Grid.GridColor);
523:    Line(Win.Wx1,Win.Wy2,Win.Wx2,Win.Wy2);
524:    While GXplot <= Lim.Lx2 do
525:    begin
526:      x:= GetGraphPointX(GXplot,Gdef);
527:      y:= Win.Wy2;
528:      Line(x,y,x,y+3);          { Plot X Grid at BottomBorder}
529:      GXplot := GXplot + Grid.Gdx
530:    end;
531:  end;
532:end;
533:
534:Procedure PlotYBorderGrid(Gdef : GraphDefine);
535:var x1,x2,y : word;
536:    GYplot  : real;
537:begin
538:  with Gdef do
539:  if GridExist(Lim.Ly1,Lim.Ly2,Grid.Gdy) then
540:  begin
541:    GYplot := GetGridStart(Lim.Ly1,Grid.Gdy);
542:    SetColor(Grid.GridColor);
543:    if Stype.Style = LowLeftCorner     then
544:    Line(Win.Wx1,Win.Wy2,Win.Wx1,Win.Wy1)
545:    { Stype.Style = LowRightCorner }       else
546:    Line(Win.Wx2,Win.Wy2,Win.Wx2,Win.Wy1);
547:    While GYplot <= Lim.Ly2 do
548:    begin
549:      y := GetGraphPointY(GYplot,Gdef);
550:      if Stype.Style = LowLeftCorner then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
551:       begin
552:         x1 := Win.Wx1;
553:         Line(x1,y,x1-5,y);    { Plot Y Grid at LeftBorder }
554:       end else
555:       begin
556:      x2 := Win.Wx2;
557:      Line(x2,y,x2+5,y);    { Plot Y Grid at RightBorder }
558:       end;
559:      GYplot := GYplot + Grid.Gdy;
560:     end;
561:  end;
562:end;
563:
564:Procedure PlotVerticalGrid(Gdef : GraphDefine);
565:var x,y1,y2  : word;
566:    GXplot   : real;
567:begin
568:  with Gdef do
569:  if GridExist(Lim.Lx1,Lim.Lx2,Grid.Gdx) then
570:  begin
571:    GXplot := GetGridStart(Lim.Lx1,Grid.Gdx);
572:    SetColor(Grid.GridColor);
573:    While GXplot <= Lim.Lx2 do
574:    begin
575:      x  := GetGraphPointX(GXplot,Gdef);
576:      y1 := Win.Wy1;
577:      y2 := Win.Wy2;
578:      SetLineStyle(DottedLn,0,NormWidth);
579:      Line(x,y1,x,y2);
580:      SetLineStyle(SolidLn,0,NormWidth);
581:      Line(x,y2,x,y2+3);
582:      GXplot := GXplot + Grid.Gdx;
583:    end;
584:  end;
585:end;
586:
587:Procedure PlotHorizontalGrid(Gdef : GraphDefine);
588:var x1,x2,y : word;
589:    GYplot  : real;
590:begin
591:  with Gdef do
592:  if GridExist(Lim.Ly1,Lim.Ly2,Grid.Gdy) then
593:  begin
594:    GYplot := GetGridStart(Lim.Ly1,Grid.Gdy);
595:    SetColor(Grid.GridColor);
596:    While GYplot <= Lim.Ly2 do
597:    begin
598:      y  := GetGraphPointY(GYplot,Gdef);
599:      x1 := Win.Wx1;
600:      x2 := Win.Wx2;
601:      if Stype.Style = LowLeftCorner then
602:      begin
603:        SetLineStyle(DottedLn,0,NormWidth);
604:     Line(x1,y,x2,y);
605:     SetLineStyle(SolidLn,0,NormWidth);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
606:     Line(x1-5,y,x1,y);
607:       end else
608:       begin
609:         SetLineStyle(DottedLn,0,NormWidth);
610:         Line(x1,y,x2,y);
611:         SetLineStyle(SolidLn,0,NormWidth);
612:     Line(x2,y,x2+5,y);
613:       end;
614:     GYplot := GYplot + Grid.Gdy;
615:    end;
616:  end;
617:end;
618:
619:Procedure SepValue(Value:real;Var Mantissa:real;Var Power:integer);
620:var St : string;
621:    Sm : string;
622:    Sp : string;
623:    code : integer;
624:begin
625:  Str(Value,St);
626:  Sm := copy(St,1,12);
627:  Sp := copy(St,15,3);
628:  Val(Sm,Mantissa,code);
629:  Val(Sp,Power,code);
630:end;
631:
632:Function IntString(Iv : LongInt) : string;
633:var Sint : string;
634:begin
635:  Str(Iv,Sint);
636:  IntString := Sint;
637:end;
638:
639:Function RealString(Rv : real) : string;
640:var Sr2 : string;
641:    Ma  : real;
642:    Pw  : integer;
643:begin
644:  SepValue(Rv,Ma,Pw);
645:  Str(Rv:Pw+2:2,Sr2);
646:  RealString := Sr2;
647:end;
648:
649:Function ExpString(Rv : real) : string;
650:var Se2 : string;
651:begin
652:  Str(Rv,Se2);
653:  Delete(Se2,6,8);
654:  ExpString := Se2;
655:end;
656:
657:Function TimeAString(Sec : integer) : string;
658:var Tmin,Tsec : integer;
659:    PlusSign  : boolean; { <+> true , <-> false }
660:    Smin,Ssec : string;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
661:begin
662:  if Sec >= 0 then PlusSign := true else PlusSign := false;
663:  Tmin := Abs(Sec div 60);
664:  Tsec := Abs(Sec mod 60);
665:  Str(Tmin,Smin);
666:  Str(Tsec,Ssec);
667:  if Tsec < 10 then Ssec := Concat('0',Ssec);
668:  if PlusSign  then
669:  begin
670:    if Tmin < 10 then Smin := Concat(' ',Smin);
671:    TimeAString := Concat(Smin,':',Ssec)
672:  end  else  TimeAString := Concat('-',Smin,':',Ssec);
673:end;
674:
675:Function TimeBString(Sec : real) : string;
676:var Tmin : real;
677:    Tsec : real;
678:    PlusSign  : boolean; { <+> true , <-> false }
679:    Smin : string;
680:begin
681:  if Sec >= 0 then PlusSign := true else PlusSign := false;
682:  Tmin := Abs(Sec/60.0);
683:  Str(Tmin:3:2,Smin);
684:  if PlusSign  then  TimeBString := Smin
685:           else  TimeBString := Concat('-',Smin);
686:end;
687:
688:Function GetValueString(Rv : real; Vtype : byte) : string;
689:begin
690:  case Vtype of
691:    IntType    : GetValueString := IntString(Round(Rv));
692:    RealType   : GetValueString := RealString(Rv);
693:    ExpType    : GetValueString := ExpString(Rv);
694:    TimeTypeA  : GetValueString := TimeAString(Round(Rv));
695:    TimeTypeB  : GetValueString := TimeBString(Rv);
696:  end;
697:end;
698:
699:Function TenPower(Pw : integer) : real;
700:var Sv   : string;
701:    Rv   : real;
702:    Code : integer;
703:begin
704:  Str(Pw,Sv);
705:  Sv := Concat('1.0E',Sv);
706:  Val(Sv,Rv,Code);
707:  TenPower := Rv;
708:end;
709:
710:Function CalScaleXWidth(Sw : real; Gdef : GraphDefine) : real;
711:var MaxWidth : byte;
712:    Wexpect : real;
713:    Wratio  : real;
714:    Gcal    : real;
715:    Ma      : real;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
716:    Pw      : Integer;
717:begin
718:  MaxWidth := TextWidth(GetValueString(Gdef.Lim.Lx1,Gdef.Stype.Xtype));
719:  if MaxWidth < TextWidth(GetValueString(Gdef.Lim.Lx2,Gdef.Stype.Xtype)) then
720:     MaxWidth := TextWidth(GetValueString(Gdef.Lim.Lx2,Gdef.Stype.Xtype));
721:  MaxWidth := MaxWidth + TextWidth('XXX');
722:  with Gdef do
723:  begin
724:    Wratio  := (Lim.Lx2-Lim.Lx1)/((Win.Wx2-Win.Wx1)*1.0);
725:    Wexpect := Wratio * MaxWidth;
726:    if Sw   >= Wexpect then CalScaleXWidth := Sw else
727:    begin
728:      Gcal  := Round(Wexpect/Grid.Gdx)*1.0;
729:      SepValue(Gcal,Ma,Pw);
730:      if Ma > 5.0 then CalScaleXWidth := Grid.Gdx*10.0 * TenPower(Pw) else
731:       if Ma > 2.5 then CalScaleXWidth := Grid.Gdx*5.0 * TenPower(Pw)  else
732:        if Ma > 2.0 then CalScaleXWidth := Grid.Gdx*2.5 * TenPower(Pw)  else
733:      CalScaleXWidth := Grid.Gdx*2.0 * TenPower(Pw)
734:    end;
735:  end;
736:end;
737:
738:Function CalScaleYWidth(Sw : real; Gdef : GraphDefine) : real;
739:var Wexpect : real;
740:    Wratio  : real;
741:    Gcal    : real;
742:    Ma      : real;
743:    Pw      : Integer;
744:begin
745:  with Gdef do
746:  begin
747:    Wratio  := (Lim.Ly2-Lim.Ly1)/((Win.Wy2-Win.Wy1)*1.0);
748:    Wexpect := Wratio * TextHeight('X') * 2.0;
749:    if Sw   >= Wexpect then CalScaleYWidth := Sw else
750:    begin
751:      Gcal  := Round(Wexpect/Grid.Gdy)*1.0;
752:      SepValue(Gcal,Ma,Pw);
753:      if Ma > 5.0 then CalScaleYWidth := Grid.Gdy*10.0 * TenPower(Pw) else
754:       if Ma > 2.5 then CalScaleYWidth := Grid.Gdy*5.0 * TenPower(Pw)  else
755:     if Ma > 2.0 then CalScaleYWidth := Grid.Gdy*2.5 * TenPower(Pw)  else
756:       CalScaleYWidth := Grid.Gdy*2.0 * TenPower(Pw)
757:    end;
758:  end;
759:end;
760:
761:Procedure PlotScaleX(Gdef : GraphDefine);
762:var Sdx    : real;
763:    GXplot : real;
764:    SXPlot : real;
765:    Sval   : String;
766:    x,y2   : word;
767:begin
768:  with Gdef do
769:  begin
770:  Sdx := CalScaleXWidth(Grid.Gdx,Gdef);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
771:  SetTextJustify(CenterText,TopText);
772:  if GridExist(Lim.Lx1,Lim.Lx2,Sdx) then
773:  begin
774:    GXplot := GetGridStart(Lim.Lx1,Sdx);
775:    While GXplot <= Lim.Lx2 do
776:    begin
777:      x  := GetGraphPointX(GXplot,Gdef);
778:      y2 := Win.Wy2;
779:      Line(x,y2,x,y2+5);
780:      Sval := GetValueString(GXplot,Stype.Xtype);
781:      OutTextXY(x,y2+8,Sval);
782:      GXplot := GXplot + Sdx;
783:    end;
784:  end;
785:  end;
786:  SetTextJustify(LeftText,Toptext);
787:end;
788:
789:Procedure PlotScaleY_Left(Gdef : GraphDefine);
790:var Sdy    : real;
791:    GYplot : real;
792:    SYPlot : real;
793:    Sval   : String;
794:    x1,y   : word;
795:begin
796:  with Gdef do
797:  begin
798:  Sdy := CalScaleYWidth(Grid.Gdy,Gdef);
799:  SetTextJustify(RightText,CenterText);
800:  if GridExist(Lim.Ly1,Lim.Ly2,Sdy) then
801:  begin
802:    GYplot := GetGridStart(Lim.Ly1,Sdy);
803:    While GYplot <= Lim.Ly2 do
804:    begin
805:      y  := GetGraphPointY(GYplot,Gdef);
806:      x1 := Win.Wx1;
807:      Line(x1,y,x1-8,y);
808:      Sval := GetValueString(GYplot,Stype.Ytype);
809:      OutTextXY(x1-11,y,Sval);
810:      GYplot := GYplot + Sdy;
811:    end;
812:  end;
813:  end;
814:  SetTextJustify(LeftText,Toptext);
815:end;
816:
817:Procedure PlotScaleY_Right(Gdef : GraphDefine);
818:var Sdy    : real;
819:    GYplot : real;
820:    SYPlot : real;
821:    Sval   : String;
822:    x2,y   : word;
823:begin
824:  with Gdef do
825:  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
826:  Sdy := CalScaleYWidth(Grid.Gdy,Gdef);
827:  SetTextJustify(LeftText,CenterText);
828:  if GridExist(Lim.Ly1,Lim.Ly2,Sdy) then
829:  begin
830:    GYplot := GetGridStart(Lim.Ly1,Sdy);
831:    While GYplot <= Lim.Ly2 do
832:    begin
833:      y  := GetGraphPointY(GYplot,Gdef);
834:      x2 := Win.Wx2;
835:      Line(x2,y,x2+8,y);
836:      Sval := GetValueString(GYplot,Stype.Ytype);
837:      if (GyPlot >= 0) and (Stype.Ytype = IntType) then
838:     Sval := Concat(' ',Sval);
839:      OutTextXY(x2+11,y,Sval);
840:      GYplot := GYplot + Sdy;
841:    end;
842:  end;
843:  end;
844:  SetTextJustify(LeftText,Toptext);
845:end;
846:
847:(* ----------------------< Plot Routine >---------------------- *)
848:
849:Procedure PlotGraphWindow(Gdef : GraphDefine);
850:var Tcolor : word;
851:begin
852:  Tcolor := GetColor;
853:  with Gdef do
854:  with Win do
855:  begin
856:    SetColor(WinColor);
857:    Rectangle(Wx1,Wy1,Wx2,Wy2);
858:  end;
859:  SetColor(Tcolor);
860:end;
861:
862:Procedure ClearGraphWindow(Gdef : GraphDefine);
863:var FillInfo : FillSettingsType;
864:begin
865:  GetFillSettings(FillInfo);
866:  with Gdef do
867:  with Win do
868:  begin
869:    SetFillStyle(SolidFill,WGndColor);
870:    Bar(Wx1+1,Wy1+1,Wx2-1,Wy2-1);
871:  end;
872:  with FillInfo do
873:  SetFillStyle(Pattern,Color);
874:end;
875:
876:Procedure PlotGrid(Gdef : GraphDefine);
877:var Tcolor : word;
878:begin
879:  Tcolor := GetColor;
880:  with Gdef do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
881:  begin
882:    case Grid.Gtype of
883:      BorderGrid :
884:     begin
885:       If Stype.Xtype <> NoScale Then  { Plot X Grid at GraphBorder }
886:          PlotXBorderGrid(Gdef);
887:       If Stype.Ytype <> NoScale Then  { Plot Y Grid at GraphBorder }
888:          PlotYBorderGrid(Gdef);
889:     end;        {========= End Border Grid =========}
890:
891:      SquareGrid :
892:     begin
893:       If Stype.Xtype <> NoScale Then  { Plot Vertical Line Grid }
894:          PlotVerticalGrid(Gdef);
895:       If Stype.Ytype <> NoScale Then  { Plot Horizontal Line Grid }
896:          PlotHorizontalGrid(Gdef);
897:     end;        {========= End Square Grid =========}
898:
899:    end;  { Grid.Gtype }
900:  end;    { Gdef }
901:  SetColor(Tcolor);
902:end;
903:
904:Procedure PlotAxis(Gdef : GraphDefine);
905:var Tcolor : word;
906:      x,y    : word;
907:begin
908:  Tcolor := GetColor;
909:  with Gdef do
910:  begin
911:  if Axis.Stat then
912:  begin
913:    SetColor(Axis.AxisColor);
914:    if ((Lim.Lx1 < 0) and (Lim.Lx2 > 0)) then
915:    begin
916:      x := GetGraphPointX(0,Gdef);
917:      Line(x,Win.Wy1,x,Win.Wy2);
918:    end;
919:    if ((Lim.Ly1 < 0) and (Lim.Ly2 > 0)) then
920:    begin
921:      y := GetGraphPointY(0,Gdef);
922:      Line(Win.Wx1,y,Win.Wx2,y);
923:    end;
924:  end;
925:  end;
926:  SetColor(Tcolor);
927:end;
928:
929:Procedure PlotScale(Gdef : GraphDefine);
930:var Tcolor : word;
931:begin
932:  Tcolor := GetColor;
933:  SetTextStyle(DefaultFont,HorizDir,1);
934:  SetColor(Gdef.Stype.ScaleColor);
935:  with Gdef Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
936:  begin
937:    Case Stype.Style of
938:
939:      LowLeftCorner  :
940:      begin
941:           If Stype.Xtype <> NoScale Then
942:           PlotScaleX(Gdef);
943:        If Stype.Ytype <> NoScale Then
944:           PlotScaleY_Left(Gdef);
945:      end;      { ========== End LowLeft Corner ========== }
946:
947:      LowRightCorner :
948:      begin
949:           If Stype.Xtype <> NoScale Then
950:           PlotScaleX(Gdef);
951:        If Stype.Ytype <> NoScale Then
952:           PlotScaleY_Right(Gdef);
953:      end;      { ========== End LowRight Corner ========== }
954:
955:    end;
956:  end;
957:  SetColor(Tcolor);
958:end;
959:
960:(* ----------------------< PlotData Routine >---------------------- *)
961:
962:Procedure MoveToXY(Ix,Iy : real; Var Pinfo : PlotInfo);
963:begin
964:  Pinfo.Ox := Ix;
965:  PInfo.Oy := Iy;
966:end;
967:
968:Function GetOx(Pinfo : PlotInfo) : real;
969:begin
970:  GetOx := Pinfo.Ox;
971:end;
972:
973:Function GetOy(Pinfo : PlotInfo) : real;
974:begin
975:  GetOy := Pinfo.Oy;
976:end;
977:
978:Procedure PlotXY(rx,ry : real; Gdef : GraphDefine);
979:var
980:  x,y    : word;
981:begin
982:  x := GetGraphPointX(rx,Gdef);
983:  Y := GetGraphPointY(ry,Gdef);
984:  PutPixel(x,y,Gdef.PlotColor);
985:end;
986:
987:Procedure ClearXY(rx,ry : real; Gdef : GraphDefine);
988:var
989:  x,y    : word;
990:begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 991:  x := GetGraphPointX(rx,Gdef);
 992:  Y := GetGraphPointY(ry,Gdef);
 993:  PutPixel(x,y,Gdef.Win.WGndColor);
 994:end;
 995:
 996:Procedure PlotLineTo(x,y : real; Pinfo : PlotInfo; Gdef : GraphDefine);
 997:{ ***************** InitStart PlotLineTo by MoveToXY ***************** }
 998:var
 999:  Ox,Oy  : word;
1000:  Cx,Cy  : word;
1001:begin
1002:  Ox := GetGraphPointX(GetOx(Pinfo),Gdef);
1003:  Oy := GetGraphPointY(GetOy(Pinfo),Gdef);
1004:  Cx := GetGraphPointX(x,Gdef);
1005:  Cy := GetGraphPointY(y,Gdef);
1006:  SetColor(Gdef.PlotColor);
1007:  Line(Ox,Oy,Cx,Cy);
1008:end;
1009:
1010:Procedure ClearLineTo(x,y : real; Pinfo : PlotInfo; Gdef : GraphDefine);
1011:{ ***************** InitStart PlotLineTo by MoveToXY ***************** }
1012:var
1013:  Tcolor : word;
1014:  Ox,Oy  : word;
1015:  Cx,Cy  : word;
1016:begin
1017:  Tcolor := GetColor;
1018:  Ox := GetGraphPointX(GetOx(Pinfo),Gdef);
1019:  Oy := GetGraphPointY(GetOy(Pinfo),Gdef);
1020:  Cx := GetGraphPointX(x,Gdef);
1021:  Cy := GetGraphPointY(y,Gdef);
1022:  SetColor(Gdef.Win.WGndColor);
1023:  Line(Ox,Oy,Cx,Cy);
1024:  SetColor(Tcolor);
1025:end;
1026:
1027:Procedure PlotLine(rx1,ry1,rx2,ry2 : real; Gdef : GraphDefine);
1028:var
1029:  x1,y1  : word;
1030:  x2,y2  : word;
1031:begin
1032:  x1 := GetGraphPointX(rx1,Gdef);
1033:  y1 := GetGraphPointY(ry1,Gdef);
1034:  x2 := GetGraphPointX(rx2,Gdef);
1035:  y2 := GetGraphPointY(ry2,Gdef);
1036:  SetColor(Gdef.PlotColor);
1037:  Line(x1,y1,x2,y2);
1038:end;
1039:
1040:Procedure ClearLine(rx1,ry1,rx2,ry2 : real; Gdef : GraphDefine);
1041:var
1042:  Tcolor : word;
1043:  x1,y1  : word;
1044:  x2,y2  : word;
1045:begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1046:  Tcolor := GetColor;
1047:  x1 := GetGraphPointX(rx1,Gdef);
1048:  y1 := GetGraphPointY(ry1,Gdef);
1049:  x2 := GetGraphPointX(rx2,Gdef);
1050:  y2 := GetGraphPointY(ry2,Gdef);
1051:  SetColor(Gdef.Win.WGndColor);
1052:  Line(x1,y1,x2,y2);
1053:  SetColor(Tcolor);
1054:end;
1055:
1056:Procedure PlotArea(rx1,ry1,rx2,ry2,BaseLn : real; Gdef : GraphDefine);
1057:Const
1058:  MaxPoints = 4;
1059:var
1060:  FillInfo : FillSettingsType;
1061:  Sp     : array[1..MaxPoints] of PointType;
1062:  x1,y1 : word;
1063:  x2,y2 : word;
1064:  Ybase : word;
1065:begin
1066:  x1 := GetGraphPointX(rx1,Gdef);
1067:  y1 := GetGraphPointY(ry1,Gdef);
1068:  x2 := GetGraphPointX(rx2,Gdef);
1069:  y2 := GetGraphPointY(ry2,Gdef);
1070:  Ybase := GetGraphPointY(BaseLn,Gdef);
1071:  Sp[1].x := x1; Sp[1].y := Ybase;
1072:  Sp[2].x := x1; Sp[2].y := y1;
1073:  Sp[3].x := x2; Sp[3].y := y2;
1074:  Sp[4].x := x2; Sp[4].y := Ybase;
1075:  GetFillSettings(FillInfo);
1076:  SetFillStyle(FillInfo.Pattern,Gdef.PlotColor);
1077:  FillPoly(MaxPoints,Sp);
1078:end;
1079:
1080:(* ------------------------ Gread Util ------------------------ *)
1081:
1082:Procedure DrawCursor(On : boolean);
1083:var
1084:  CursorHeight,CursorWidth : Word;
1085:  Tcolor : Word;
1086:  F : FillSettingsType;
1087:  XStart,XEnd    : word;
1088:  YStart,YEnd    : word;
1089:  i                    : integer;
1090:begin
1091:  CursorHeight := TextHeight('H');
1092:  CursorWidth :=  TextWidth('W');
1093:  XStart := GetX;
1094:  XEnd := XStart + CursorWidth - 1;
1095:  YStart := GetY;
1096:  YEnd := YStart + CursorHeight - 2;
1097:  if ( On ) then
1098:  begin
1099:    i := 0;
1100:    SetWriteMode(XORput);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1101:    repeat
1102:      i := i+1;
1103:      if i > 1500 then
1104:      begin
1105:     line(XStart,YStart,XStart,YEnd);
1106:     line(XStart,YEnd,XEnd,YEnd);
1107:     line(XEnd,YEnd,XEnd,YStart);
1108:     line(XEnd,YStart,XStart,YStart);
1109:     i := 0;
1110:      end;
1111:    until KeyPressed;
1112:  end
1113:  else
1114:  begin
1115:    if GetPixel(XStart,YStart+1) <> BkTextColor then
1116:    begin
1117:      SetWriteMode(XORput);
1118:      line(XStart,YStart,XStart,YEnd);
1119:      line(XStart,YEnd,XEnd,YEnd);
1120:      line(XEnd,YEnd,XEnd,YStart);
1121:      line(XEnd,YStart,XStart,YStart);
1122:    end;
1123:    SetWriteMode(COPYput);
1124:  end;
1125:end;
1126:
1127:Procedure EchoIt(ch : char);
1128:begin
1129:  if (GraphicsMode) then
1130:  begin
1131:    DrawCursor(False);
1132:    OutText(ch);
1133:    DrawCursor(True);
1134:  end
1135:  Else
1136:    Write(ch);
1137:end;
1138:
1139:Procedure BlankIt(ch : char);
1140:var
1141:  XPos,YPos : word;
1142:  OrigDrawingColor : byte;
1143:begin
1144:  if (GraphicsMode) Then
1145:  begin
1146:    DrawCursor(False);
1147:    XPos := GetX;
1148:    XPos := XPos - TextWidth(ch);
1149:    YPos := GetY;
1150:    OrigDrawingColor := GetColor;
1151:    SetColor(BkTextColor);
1152:    MoveTo(XPos,YPos);
1153:    OutText(ch);
1154:    SetColor(OrigDrawingColor);
1155:    MoveTo(XPos,YPos);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1156:  end
1157:  Else
1158:  begin
1159:    XPos := WhereX;
1160:    XPos := XPos - 1;
1161:    YPos := WhereY;
1162:    GotoXY(XPos,YPos);
1163:    Write(' ');
1164:    GotoXY(XPos,YPos);
1165:  end;
1166:end;
1167:
1168:Procedure ReadChar(var ch : char);
1169:begin
1170:  if (GraphicsMode) then
1171:    DrawCursor(True);
1172:  ch := Readkey;
1173:  if (ch = #0) or (ch = #224) then
1174:  begin
1175:    ch := Readkey;
1176:    if ch = #75 then ch := #8 else ch := #0;
1177:  end;
1178:  if ch = #27 then ch := #0;
1179:end;
1180:
1181:Procedure ReadGraphString(var S :String);
1182:var
1183:  ch  : char;
1184:  TemString : String;
1185:begin
1186:  FillChar(TemString,SizeOf(TemString),#0);
1187:  ReadChar(ch);
1188:  While(ch <> #13) do
1189:  begin
1190:    if (ch = #8) then
1191:    begin
1192:      if(TemString[0] > #0) then
1193:      begin
1194:     BlankIt(TemString[Length(TemString)]);
1195:        Dec(TemString[0]);
1196:      end
1197:    end else
1198:    begin
1199:      if (ch <> #0) then
1200:      begin
1201:     EchoIt(ch);
1202:     TemString := temString + ch;
1203:      end;
1204:    end;
1205:    ReadChar(ch);
1206:  end;
1207:  if (length(TemString) > 0) then
1208:    S := TemString;
1209:  if (GraphicsMode) then
1210:    DrawCursor(False);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1211:end;
1212:
1213:Procedure EditGraphString(var S :String);
1214:var
1215:  ch  : char;
1216:  TemString : String;
1217:  i       : integer;
1218:begin
1219:  SetTextJustify(LeftText,TopText);
1220:  FillChar(TemString,SizeOf(TemString),#0);
1221:  TemString := Copy(S,1,Length(s));
1222:  For i := 1 to Length(s) do
1223:  begin
1224:    Ch := TemString[i];
1225:    OutText(ch);
1226:  end;
1227:  ReadChar(ch);
1228:  While(ch <> #13) do
1229:  begin
1230:    if (ch = #8) then
1231:    begin
1232:      if(TemString[0] > #0) then
1233:      begin
1234:     BlankIt(TemString[Length(TemString)]);
1235:     Dec(TemString[0]);
1236:      end
1237:    end else
1238:    begin
1239:      if (ch <> #0) then
1240:      begin
1241:     EchoIt(ch);
1242:     TemString := temString + ch;
1243:      end;
1244:    end;
1245:    ReadChar(ch);
1246:  end;
1247:  if (length(TemString) > 0) then
1248:    S := TemString;
1249:  if (GraphicsMode) then
1250:    DrawCursor(False);
1251:end;
1252:
1253:(* ------------------------- RS232 Util ------------------------- *)
1254:
1255:Procedure InitRS232(Com : word);
1256:begin
1257:  Regs.AH := $00;
1258:  Regs.AL := $E3;
1259:  Regs.DX := Com;
1260:  Intr ($14,Regs);
1261:end;
1262:
1263:Procedure RsSend(Datas : byte; Com : word);
1264:begin
1265:  Regs.AH := $01;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1266:  Regs.AL := Datas;
1267:  Regs.DX := Com;
1268:  Intr ($14,Regs);
1269:end;
1270:
1271:Procedure RsStatus(Var Lob,Hib : byte; Com : word);
1272:begin
1273:  Regs.DX := Com;
1274:  Regs.AH := $03;
1275:  Intr ($14,Regs);
1276:  Lob := Regs.AL;
1277:  Hib := Regs.AH;
1278:end;
1279:
1280:Procedure RsRecive(Var Data : byte;Var ErrCode:byte; Com : word);
1281:begin
1282:  ErrCode := 0;
1283:  Data    := 0;
1284:  Regs.DX := Com;
1285:  Regs.AH := $02;
1286:  Intr ($14,Regs);
1287:  if regs.AH = 0 then Data := Regs.AL else ErrCode := 128;
1288:end;
1289:
1290:Procedure LinkActiveOn(Com : Word);
1291:begin
1292:  RsSend(LinkActive,Com);
1293:end;
1294:
1295:Procedure TriggerActiveOn(Com : word);
1296:begin
1297:  RsSend(TgActive,Com);
1298:end;
1299:
1300:Procedure SendXOn(Com : Word);
1301:begin
1302:  RsSend(XOn,Com);
1303:end;
1304:
1305:Procedure SendXOff(Com : Word);
1306:begin
1307:  RsSend(XOff,Com);
1308:end;
1309:
1310:Procedure SendBreak(Com : Word);
1311:begin
1312:  RsSend(Break,Com);
1313:end;
1314:
1315:Function GetRsData(Var ErrCode : byte; Com : word) : LongInt;
1316:Const Slength = 8;
1317:var   Adata   : byte;
1318:      Sdata   : string;
1319:      Ldata   : LongInt;
1320:      Count   : byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1321:      Check    : integer;
1322:begin
1323:  Count := 0;
1324:  Check := 0;
1325:  FillChar(Sdata,SizeOf(Sdata),#0);
1326:  SendXOn(Com);
1327:  Repeat
1328:    Inc(Count);
1329:    RsRecive(Adata,ErrCode,Com);
1330:    if Char(Adata) <> ' ' then Sdata := Sdata+Char(Adata)
1331:                          else Inc(Check);
1332:  until (Check = 2) or (ErrCode = 128);
1333:  if ErrCode = 128 then       { if Time Out Error }
1334:  begin
1335:    SendXOff(Com);
1336:  end else
1337:  begin
1338:    SendXOff(Com);
1339:    Val(Sdata,Ldata,Check);
1340:    GetRsdata := Ldata;
1341:  end;
1342:end;
1343:
1344:begin
1345:  SetBkTextColor(Black);
1346:end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIT GRAPHPRN.PAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1:Unit GraphPRN;
 2:
 3:Interface
 4:
 5:Uses Dos,Graph;
 6:
 7:Var
 8:  LST : Text; { New printer file variable }
 9:
10:Procedure HCopy(Gmode : integer);
11:Procedure VCopy(Gmode : integer);
12:Function LSTOutputToPrinter(Var F : TextRec) : integer;
13:
14:{ Procedure HardCopy print the current Viewport to an IBM or }
15:{ Epson-compatible graphics printer                          }
16:{                                                           }
17:{ Valid Gmode numbers are :                                 }
18:{  -4 to -1 for Epson and IBM graphic printers              }
19:{   0 to  7 for Epson Printers                              }
20:
21:
22:Implementation
23:
24:Procedure HCopy(Gmode : integer);
25:
26:Const
27:  Bits : Array[0..7] of Byte = (128,64,32,16,8,4,2,1);
28:
29:Var
30:  X,Y,Yofs        : integer;      { Screen location variable   }
31:  BitData,MaxBits : byte;         { Number of bits to transfer }
32:  Vport           : ViewPortType; { Used to Get view setting   }
33:  Height,Width    : word;         { Size of image to transfer  }
34:  HiBit,LoBit     : char;         { Char size of image         }
35:  LineSpacing,                    { Additoinal Info for dump   }
36:  GraphixPrefix   : string[10];   { Additoinal Info for dump   }
37:  BKColor         : byte;         { Value of current bk color  }
38:
39:begin
40:  LineSpacing := #27+'3'+#24;     { 24/216 Inch line spacing   }
41:  Case Gmode of
42:   -1 : GraphixPrefix := #27+'K'; { Standard Density           }
43:   -2 : GraphixPrefix := #27+'L'; { Double Density             }
44:   -3 : GraphixPrefix := #27+'Y'; { Dbl.Density Dbl.Speed      }
45:   -4 : GraphixPrefix := #27+'Z'; { Quad.Density               }
46:  0..7 : GraphixPrefix := #27+'*'+Chr(Gmode); { 8-Pin Bit Img }
47:  Else
48:     Exit;                       { Invalid Mode Selection     }
49:  end;
50:  BKColor := GetBKColor;
51:  GetViewSettings(Vport);         { Get size of image to be    }
52:  Height := Vport.Y2-Vport.Y1;    { printed                    }
53:  Width  := (Vport.X2+1)-Vport.X1;
54:  HiBit  := Chr(Hi(Width));      { Translate sizes to char    }
55:  LoBit  := Chr(Lo(Width));      { for output to printer      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Write(LST,LineSpacing);
  Y := 0;                              {First Y Coordinate          }
  While Y < Height Do                  { Do not go beyond viewport }
  begin
    Write(LST,GraphixPrefix,LoBit,HiBit);
                                        { Tell printer graphics info }
    For X := 0 to Width-1 Do           { Go across screen lt to rt }
    begin
      BitData := 0;                    { Initialize to all off (0) }
      if Y+7 <= Height then            { Make sure there are 8     }
        MaxBits := 7                   { lines of info and set it  }
      else                             { Acordingly                }
        MaxBits := Height-Y;
        For  Yofs := 0 to MaxBits Do { Go top to bottom on line  }
          if (GetPixel(X,Yofs+Y) <> BKColor) then
            BitData := BitData or Bits[Yofs];
                                        { if pixel on, add to output }
     Write(LST,Chr(BitData));    { byte is created, output it }
      end;
    WriteLn(LST);
    Inc(Y,8);                          { Inc by 8 as each line is 8 }
                                       { actual scan line in height }
  end;
  Write(LST,#27+'@');                  { Send reset command         }
end;

Procedure VCopy(Gmode : integer);

Const
  Bits : Array[0..7] of Byte = (128,64,32,16,8,4,2,1);

Var
  X,Y,Yofs,Xofs    : integer;        { Screen location variable    }
  BitData,MaxBits  : byte;           { Number of bits to transfer  }
  Vport            : ViewPortType;   { Used to Get view setting    }
  Height,Width     : word;           { Size of image to transfer   }
  HiBit,LoBit      : char;           { Char size of image          }
  LineSpacing,                       { Additoinal Info for dump    }
  GraphixPrefix    : string[10];     { Additoinal Info for dump    }
  BKColor          : byte;           { Value of current bk color   }

begin
  LineSpacing := #27+'3'+#24;        { 24/216 Inch line spacing    }
  Case Gmode of
   -1 : GraphixPrefix := #27+'K'; { Standard Density             }
   -2 : GraphixPrefix := #27+'L'; { Double Density               }
   -3 : GraphixPrefix := #27+'Y'; { Dbl.Density Dbl.Speed        }
   -4 : GraphixPrefix := #27+'Z'; { Quad.Density                 }
  0..7 : GraphixPrefix := #27+'*'+Chr(Gmode);  { 8-Pin Bit Img   }
  Else
    Exit;                              { Invalid Mode Selection    }
  end;
  BKColor := GetBKColor;
  GetViewSettings(Vport);              { Get size of image to be   }
  Width  := (Vport.X2+1)-Vport.X1;     { printed                   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
111:  Height := Vport.X2-Vport.X1;
112:  HiBit  := Chr(Hi(Width));          { Translate sizes to char  }
113:  LoBit  := Chr(Lo(Width));          { for output to printer    }
114:  Write(LST,LineSpacing);
115:  X := 0;                            {First Y Coordinate        }
116:  While X < Height Do                { Do not go beyond viewport }
117:  begin
118:    Write(LST,GraphixPrefix,LoBit,HiBit);
119:                                      { Tell printer graphics info }
120:    For Y := Width-1 DownTo 0 Do     { Go across screen Dn to Tp }
121:    begin
122:      BitData     := 0;              { Initialize to all off (0) }
123:      if X+7 <= Height then          { Make sure there are 8     }
124:        MaxBits   := 7               { lines of info and set it  }
125:      else                           { Acordingly                }
126:     MaxBits    := Height-X;
127:      For  Xofs := 0 to MaxBits Do   { Go lt to rt               }
128:        if (GetPixel(Xofs+X,Y) <> BKColor) then
129:       BitData := BitData or Bits[Xofs];
130:                                      { if pixel on, add to output }
131:      Write(LST,Chr(BitData));       { byte is created, output it }
132:    end;
133:    WriteLn(LST);
134:    Inc(X,8);                        { Inc by 8 as each line is 8 }
135:                                     { actual scan line in height }
136:  end;
137:  Write(LST,#27+'@');                { Send reset command        }
138:end;
139:
140:{$F+}
141:
142:Function LSTNofunction(Var F : TextRec) : integer;
143:{ This Function performs a NUL operation on LST in case a reset }
144:{ or a Rewrite is called.                                       }
145:begin
146:  LSTNoFunction  := 0;
147:end;
148:
149:Function LSTOutputToPrinter(Var F : TextRec) : integer;
150:{ This function sends the output to the printer port number     }
151:{ stored in the first byte of the UserData area of the text     }
152:{ record.                                                      }
153:Var
154:  Regs : registers;
155:  P    : word;
156:begin
157:  With  F do
158:  begin
159:    P := 0;
160:    Regs.AH := 16;
161:    While((P < BufPos) and ((Regs.AH and 16) = 16)) do
162:    begin
163:      Regs.AL  := Ord(BufPtr^[P]);
164:      Regs.AH  := 0;
165:      Regs.DX  := UserData[1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
166:      Intr($17,Regs);
167:      Inc(P);
168:    end;
169:    BufPos := 0;
170:  end;
171:  if (Regs.AH and 16) = 16 then
172:    LSTOutputToPrinter := 0                { No Error           }
173:  else
174:    if (Regs.AH and 32) = 32 then
175:      LSTOutputToPrinter := 159            { Out of paper       }
176:    else
177:      LSTOutputToPrinter := 160;           { Device write Fault }
178:end;
179:
180:{$F-}
181:
182:Procedure AssignLST(Port : byte);
183:{ This Procedure set up the .LST text file record as would the }
184:{ Assign procedure, and initializes it as would a call to the  }
185:{ Reset procedure. It then stores the LPT port number in the   }
186:{ first byte of the UserData Area of the TextRec type.         }
187:begin
188:  With TextRec(LST) do
189:  begin
190:    Handle     := $FFF0;
191:    Mode       := FmOutput;
192:    BufSize    := SizeOf(Buffer);
193:    BufPtr     := @Buffer;
194:    BufPos     := 0;
195:    OpenFunc   := @LSTNoFunction;
196:    InOutFunc  := @LSTOutputToPrinter;
197:    FlushFunc  := @LSTOutputToPrinter;
198:    CloseFunc  := @LSTOutputToPrinter;
199:    UserData[1] := Port - 1;   { Sub 1 as DOS count from zero }
200:  end;
201:end;
202:
203:Begin
204:  AssignLST(1); { Set output printer to LPT1 by  }
205:              { defualt. Chenge this value to  }
206:              { select other LPT ports.        }
207:End.
208:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

ซอฟต์แวร์ของเครื่องเก็บและรับส่งข้อมูล (ใช้กับไมโครคอนโทรลเลอร์ 8751)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  1:;*************************************************************************
  2:;
  3:;       MASTER CONTROL PROGRAM FOR DATA ACQUISITION SYSTEM FOR HPLC
  4:;            WRITTEN IN 8051 MICROCONTROLLER ASSEMBLY LANGUAGE
  5:;
  6:;*************************************************************************
  7:
  8:;
  9:;
 10:
 11:          CPU        "8051.TBL"
 12:          HOF        "INT8"
 13:;
 14:;
 15:;MCS-51 INTERNAL REGISTERS
 16:;
 17:B:        EQU        0F0H        ;B REGISTER
 18:ACC:      EQU        0E0H        ;ACCUMULATOR
 19:PSW:      EQU        0D0H        ;PROGRAM STATUS WORD
 20:IPC:      EQU        0B8H        ;INTERRUPT PRIORITY
 21:P3:       EQU        0B0H        ;PORT 3
 22:IEC:      EQU        0A8H        ;INTERRUPT ENABLE
 23:P2:       EQU        0A0H        ;PORT 2
 24:SBUF:     EQU        99H         ;SEND BUFFER
 25:SCON:     EQU        98H         ;SERIAL CONTROL
 26:P1:       EQU        90H         ;PORT 1
 27:TH1:      EQU        8DH         ;TIMER 1 HIGH
 28:TH0:      EQU        8CH         ;TIMER 0 HIGH
 29:TL1:      EQU        8BH         ;TIMER 1 LOW
 30:TL0:      EQU        8AH         ;TIMER 0 LOW
 31:TMOD:     EQU        89H         ;TIMER MODE
 32:TCON:     EQU        88H         ;TIMER CONTROL
 33:PCON:     EQU        87H         ;POWER CONTROL REGISTER
 34:DPH:      EQU        83H         ;DATA POINTER HIGH
 35:DPL:      EQU        82H         ;DATA POINTER LOW
 36:SP:       EQU        81H         ;STACK POINTER
 37:P0:       EQU        80H         ;PORT 0
 38:;
 39:;MCS-51 INTERNAL BIT ADDRESSES
 40:;
 41:CY:       EQU        0D7H        ;CARRY FLAG
 42:AC:       EQU        0D6H        ;AUXILIARY-CARRY FLAG
 43:F0:       EQU        0D5H        ;USER FLAG 0
 44:RS1:      EQU        0D4H        ;REGISTER SELECT MSB
 45:RS0:      EQU        0D3H        ;REGISTER SELECT LSB
 46:OV:       EQU        0D2H        ;OVERFLOW FLAG
 47:P:        EQU        0D0H        ;PARITY FLAG
 48:PS:       EQU        0BCH        ;PRIORITY SERIAL PORT
 49:PT1:      EQU        0BBH        ;PRIORITY TIMER 1
 50:PX1:      EQU        0BAH        ;PRIORITY EXTERNAL 1
 51:PT0:      EQU        0B9H        ;PRIORITY TIMER 0
 52:PX0:      EQU        0B8H        ;PRIORITY EXTERNAL 0
 53:EA:       EQU        0AFH        ;ENABLE ALL INTERRUPT
 54:ES:       EQU        0ACH        ;ENABLE SERIAL INTERRUPT
 55:ET1:      EQU        0ABH        ;ENABLE TIMER 1 INTERRUPT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  56:EX1:       EQU       0AAH      ;ENABLE EXTERNAL 1 INTERR
  57:ET0:       EQU       0A9H      ;ENABLE TIMER 0 INTERRUPT
  58:EX0:       EQU       0A8H      ;ENABLE EXTERNAL 0 INTERR
  59:SM0:       EQU       09FH      ;SERIAL MODE 0
  60:SM1:       EQU       09EH      ;SERIAL MODE 1
  61:SM2:       EQU       09DH      ;SERIAL MODE 2
  62:REN:       EQU       09CH      ;SERIAL RECEPTION ENABLE
  63:TB8:       EQU       09BH      ;TRANSMITT BIT 8
  64:RB8:       EQU       09AH      ;RECEIVE BIT 8
  65:TI:        EQU       099H      ;TRANSMIT INTERRUPT FLAG
  66:RI:        EQU       098H      ;RECEIVE INTERRUPT FLAG
  67:TF1:       EQU       08FH      ;TIMER 1 OVERFLOW FLAG
  68:TR1:       EQU       08EH      ;TIMER 1 RUN CONTROL BIT
  69:TF0:       EQU       08DH      ;TIMER 0 OVERFLOW FLAG
  70:TR0:       EQU       08CH      ;TIMER 0 RUN CONTROL BIT
  71:IE1:       EQU       08BH      ;EXT INTERR. 1 EDGE FLAG
  72:IT1:       EQU       08AH      ;EXT INTERR. 1 TYPE FLAG
  73:IE0:       EQU       089H      ;EXT INTERR. 0 EDGE FLAG
  74:IT0:       EQU       088H      ;EXT INTERR. 0 TYPE FLAG
  75:
  76:P1.0:      EQU       090H      ;  START COMMAND
  77:P1.1:      EQU       091H      ;  INTERRUPT RUN
  78:P1.2:      EQU       092H      ;  XOFF
  79:P1.3:      EQU       093H      ;  MEMORY BANK SWITCHING
  80:P1.4:      EQU       094H
  81:P1.5:      EQU       095H
  82:P1.6:      EQU       096H
  83:P1.7:      EQU       097H
  84:
  85:SAVE_L:    EQU       048H      ; SAVE POINTER USE IN INTERRUPT SERVICE
  86:SAVE_H:    EQU       049H      ; ROUTINE
  87:                                ; RAM BIT ADDRESS
  88:LED:       EQU        08H      ; INTERRUPT BLINK
  89:XOFF_FLAG:EQU         0AH
  90:TOP_FIFO: EQU        1000H     ; TOP OF FIFO BUFFER
  91:
  92:CR:        EQU        0DH
  93:LF:        EQU        0AH
  94:EOS:       EQU        10H
  95:
  96:;*****************************************************************************
  97:;
  98:                          ORG       0000H
  99:                          LJMP      MAIN
 100:
 101:                          ORG       0013H
 102:                          LJMP      SERVICE_INT1
 103:
 104:                          ORG       0100H
 105:
 106:;;INIT_RS232C INITIALIZED 8051 SERIAL PORT
 107:;             BAUD RATE: 9600
 108:;             DATA LENGTH: 8 BIT
 109:;             STOP BIT: 1 BIT
 110:;             PARITY: NO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
111:;            X-TAL: 11.0592 MHz
112:
113:INIT:    MOV       TMOD,#00101100B
114:         MOV       SCON,#01010010B
115:         MOV       TCON,#01101001B
116:         MOV       TH1,#0FDH        ;TIMER1 LOAD VALUE
117:         SETB      TR1              ;START TIMER1
118:         RET
119:
120:
121:;SEND    SEND A TO SERIAL PORT
122:;        CHECK TRANSMITTER BUFFER BEFORE SEND
123:;
124:;        ENTRY: A
125:;        EXIT: NO
126:;
127:SEND:    JNB       TI,SEND
128:         CLR       TI
129:         MOV       SBUF,A           ;OK BUFFER EMPTY SEND OUT...
130:         MOV       R5,#10H
131:LAG:     DJNZ      R5,LAG           ; LAG BETWEEN SUCCESSIVE BYTE
132:         RET
133:;
134:;RECIVES RECIVE BYTE FROM SERIAL PORT WITH TIME-OUT FUNCTION
135:;        ENTRY: NO
136:;        EXIT: A = ASCII NUMBER OR STRING COMMAND
137:;          A = FF    TIME-OUT
138:
139:RECIVES: MOV       R7,#05H
140:AGAIN:       MOV       R6,#00H          ; TIME-OUT DELAY
141:GET_RI:      JB        RI,READY
142:
143:         DJNZ      R6,GET_RI
144:         DJNZ      R7,AGAIN
145:         MOV       A,#0FFH          ; TIME-OUT NO DATA RECEPTED
146:         RET
147:READY:       CLR       RI
148:         MOV       A,SBUF
149:         RET
150:
151:
152:;RECIVE  RECIVE BYTE PUT TO A FROM SERIAL PORT
153:;        WAIT UNTIL RECIVER BUFFER READY
154:;        ENTRY: NO
155:;        EXIT: A
156:
157:RECIVE:  JNB       RI,RECIVE
158:         CLR       RI
159:         MOV       A,SBUF       ; GET IT
160:         RET
161:
162:
163:
164:;CR_SEND SEND CR TO RS232
165:;        ENTRY: NO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
166:;         EXIT: NO
167:;
168:CR_LF:    MOV         A,#CR
169:          LCALL       SEND
170:          MOV         A,#LF
171:          LCALL       SEND
172:          RET
173:
174:XOFF:     EQU          13H
175:XON:      EQU          11H
176:BREAK:    EQU          00H
177:
178:
179:;XOFF_SEND SEND XOFF TO RS232
180:;         ENTRY: NO
181:;         EXIT: NO
182:;
183:XOFF_SEND: MOV      A,#XOFF     ;GET XOFF CHARACTER
184:           LCALL     SEND
185:           RET
186:
187:;XON_SEND
188:;
189:;
190:;
191:XON_SEND: MOV       A,#XON
192:          LCALL     SEND
193:          RET
194:
195:;TEST_RS232  SEND 0-255 TO IBM
196:;
197:          LCALL     INIT
198:          MOV       R7,#0FFH
199:          MOV       R6,#00H
200:TEST2:    MOV       A,R6
201:          LCALL     SEND
202:          INC       R6
203:          DJNZ      R7,TEST2
204:          LJMP      0H              ;GO BACK MONITOR
205:
206:
207:
208:;TEST_RS232 RECIVE DATA FROM IBM SAVE TO BUFFER
209:;         POINTED BY DPTR
210:;         ENTRY: NO
211:;         EXIT: DATA IN BUFFER {external data memory start $1500}
212:;
213:TEST3:    LCALL      INIT
214:          MOV        DPTR,#1050H
215:          MOV        R7,#0FFH
216:GET:      LCALL      RECIVE
217:          MOVX       @DPTR,A     ;SAVE TO EXTERNAL RAM
218:          INC        DPTR
219:          DJNZ       R7,GET
220:          LJMP       0H          ;GO BACK MONITOR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
221:
222:
223:;ASCII_BIN  CONVERTS SINGLE ASCII CHARACTER TO SINGLE NIBBLE BINARY
224:;           ENTRY: A
225:;           EXIT: A
226:;
227:ASCII_BIN: MOV      R6,A
228:           SUBB     A,#41H
229:           JNC      ASCII_AF
230:           MOV      A,R6
231:           SUBB     A,#30H
232:           RET
233:ASCII_AF:  MOV      A,R6
234:           SUBB     A,#37H
235:           RET
236:
237:;INIT_8254 INITIALIZED PROGRAMMABLE TIMER COUNTER 8254
238:;          MAX INPUT FERQUENCY 8 MHz
239:;          COUNTER ADDRESS:
240:;
241:;          $8000     COUNTER#0
242:;          $8001     COUNTER#1
243:;          $8002     COUNTER#2
244:;          $8003     CONTROL REGISTER
245:;
246:;          CONTROL WORD FORMAT
247:;
248:;          D7  D6  D5  D4  D3  D2  D1  D0
249:;         SC1 SC2 RW1 RW0  M2  M1  M0  BCD
250:;
251:;       SC:select counter                  M-MODE
252:;       SC0   SC1                          M2   M1   M0
253:;       0     0     select counter 0       0    0    0      mode0
254:;             0     1     select counter 1       0    0    1      mode1
255:;       1     0     select counter 2       0    1    0      mode2
256:;       1     1     read-back command      0    1    1      mode3
257:;
258:;                                          1    0    1      mode5
259:;       RW-read/write
260:;       RW0   RW1                                    BCD
261:;        0     0      counter latch command          0    binary counter mode
262:;        0     1      read/write LSB only            1    BCD 4 decades
263:;        1     0      read/write MSB only
264:;        1     1      read/write LSB first,then MSB
265:;
266:COUNTER_0: EQU      8000H
267:COUNTER_1: EQU      8001H
268:COUNTER_2: EQU      8002H
269:CONTROL:   EQU      8003H
270:
271:;SETMODE0 SET COUNTER#0 & COUNTER#1 TO MODE0
272:;         INTERRUPT ON TERMINAL COUNTER
273:;
274:SET_CNT0:  MOV      A,#00110000B      ;LSB FIRST,THEN MSB, BINARY COUNTING
275:           MOV      DPTR,#CONTROL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
276:            MOVX    @DPTR,A
277:            RET
278:
279:SET_CNT1:   MOV     A,#01110000B
280:            MOV     DPTR,#CONTROL
281:            MOVX    @DPTR,A
282:            RET
283:;
284:;SETMODE3 SET COUNTER#2 TO MODE3,GENERATE SQUARE WAVE
285:;
286:;
287:SET_CNT2:   MOV     A,#10110110B
288:            MOV     DPTR,#CONTROL
289:            MOVX    @DPTR,A
290:            MOV     A,#20H
291:            MOV     DPTR,#COUNTER_2
292:            MOVX    @DPTR,A
293:            MOV     A,#4EH
294:            MOVX    @DPTR,A
295:            RET
296:
297:;SCAN_DISP SCAN DISPLAY 1 CYCLE
298:;             ENTRY: R0 BUFFER POINTER
299:;             EXIT: NO
300:
301:PORT_A:        EQU     00H
302:PORT_B:        EQU     01H
303:PORT_C:        EQU     02H
304:CONTROL_8255:  EQU     03H
305:
306:
307:
308:SCAN_DISP: MOV     R2,#06H
309:            MOV     R6,#00H         ; DIGIT SCAN CODE
310:SC1:        MOV     A,@R0           ; GET BYTE
311:            MOV     DPTR,#PORT_A
312:            MOVX    @DPTR,A
313:            MOV     A,R6
314:            ORL     A,#00001000B
315:            MOV     DPTR,#PORT_C
316:            MOVX    @DPTR,A
317:            MOV     R1,#0FFH
318:DELAY:      DJNZ    R1,DELAY
319:            CLR     A
320:            MOV     DPTR,#PORT_A
321:            MOVX    @DPTR,A
322:            INC     R6
323:            INC     R0
324:            DJNZ    R2,SC1
325:            RET
326:
327:;NIBBLE_7SEG CONVERT DATA IN A TO 7 SEGMENT PATTREN
328:;             ENTRY: A
329:;             EXIT: A
330:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NIBBLE_7SEG: MOV     DPTR,#TABLE1
             MOVC    A,@A+DPTR
             RET

TABLE1:      DFB     3FH          ;0
             DFB     06H          ;1
             DFB     5BH          ;2
             DFB     4FH          ;3
             DFB     66H          ;4
             DFB     6DH          ;5
             DFB     7DH          ;6
             DFB     07H          ;7
             DFB     7FH          ;8
             DFB     6FH          ;9
             DFB     77H          ;A
             DFB     7CH          ;B
             DFB     39H          ;C
             DFB     5EH          ;D
             DFB     79H          ;E
             DFB     71H          ;F

;TRANSFER MOVE 3 BYTE IN DATA BUFFER POINTED BY R0 TO 6 BYTE DISPLAY BUFFER
;        THEN CONVERT TO 7 SEGMENT PATTREN
;        ADDRESS OF CURRENT DATA       DISPLAY BUFFER
;        EXP:
;         $36   LSD                    $30      LSD
;         $37                          $31
;         $38   MSD                    $32
;
;
;         ENTRY: R0
;         EXIT: 7 SEGMENT PATTREN

TRANSFER:    MOV     R7,#03H
             MOV     R1,#30H
TRANS1:      MOV     A,@R0
             MOV     R6,A
             ANL     A,#0FH       ; MASK 4 BIT LOW
             LCALL   NIBBLE_7SEG
             MOV     @R1,A
             INC     R1
             MOV     A,R6
             SWAP    A
             ANL     A,#0FH
             LCALL   NIBBLE_7SEG
             MOV     @R1,A
             INC     R0
             INC     R1
             DJNZ    R7,TRANS1
             RET

;SHIFT_16BIT SHIFT LEFT 1 BIT ADDRESS $3D,$3E
;            $3D  LO BYTE
;            $3E  HI BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
386:;            ENTRY: NO
387:;            EXIT: Carry Flag
388:
389:SHIFT_16BIT: CLR    C           ;     {($3E,$3D)} X 2
390:             MOV    A,3DH
391:             ADD    A,3DH
392:             MOV    3DH,A
393:             MOV    A,3EH
394:             ADDC   A,3EH
395:             MOV    3EH,A
396:             RET                ; CARRY GONE
397:
398:;BIN_BCD CONVERT 16 BIT BINARY TO BCD
399:;        SHIFT LEFT 1 BIT FROM MOST SIGNIFICANT BIT TO LEAST SIGNIFICANT
400:;        BIT , IF CARRY = 1 THEN BCD= (2^N + BCD)
401:;
402:;     ENTRY: 16 BIT DATA
403:;               $3D = LO BYTE
404:;               $3E = HI BYTE
405:;        EXIT: 6 DIGIT BCD
406:;               $36L =     0
407:;               $36H =    10
408:;               $37L =   100
409:;               $37H =  1000
410:;               $38L = 10000
411:;               $38H = 100000
412:
413:BIN_BCD:   MOV     36H,#00      ; CLEAR BCD BEFORE
414:          MOV     37H,#00
415:          MOV     38H,#00
416:          MOV     R1,#00H
417:          MOV     R7,#16D
418:BIN_1:          LCALL   SHIFT_16BIT
419:          JNC     NEXT_BIT
420:
421:;     OK CARRY = 1 ADD BCD TO 2^N CONSTANT IN TABLE2
422:          MOV     DPTR,#TABLE2
423:          CLR     C
424:          MOV     R0,#36H
425:          MOV     R6,#03H
426:BIN_2:          MOV     A,R1
427:          MOVC    A,@A+DPTR
428:          ADDC    A,@R0
429:          DAA
430:          MOV     @R0,A
431:          INC     DPTR
432:          INC     R0
433:          DJNZ    R6,BIN_2
434:
435:NEXT_BIT:  INC     R1
436:          INC     R1
437:          INC     R1
438:          DJNZ    R7,BIN_1
439:          RET
440:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
441:TABLE2:          DFB     68H,27H,03H
442:                 DFB     84H,63H,01H
443:                 DFB     92H,81H,00H
444:                 DFB     96H,40H,00H
445:                 DFB     48H,20H,00H
446:                 DFB     24H,10H,00H
447:                 DFB     12H,05H,00H
448:                 DFB     56H,02H,00H
449:                 DFB     28H,01H,00H
450:                 DFB     64H,00H,00H
451:                 DFB     32H,00H,00H
452:                 DFB     16H,00H,00H
453:                 DFB     08H,00H,00H
454:                 DFB     04H,00H,00H
455:                 DFB     02H,00H,00H
456:                 DFB     01H,00H,00H
457:
458:;#############################################################################
459:;SERVICE_INT1 READ LSB FIRST, THEN MSB FROM LATCHING COUNTER
460:;             CONSTANT LSB ADDRESS = $8006
461:;             CONSTANT MSB ADDRESS = $8007
462:;             {
463:;               READ LSB,MSB   /* use latching read the counter */
464:;                  STORE LSB TO $3D
465:;                        MSB TO $3E
466:;               BIN_BCD ($3D,$3E)
467:;               INTEGRATED FIVE CONSECUTIVE DATA SAMPLE
468:;               TRANSMIT TO LOTUS MEASURE
469:;             }
470:;********************* NOTE ***********************************
471:;           START OF THIS ROUTINE COME FROM $1300
472:;             FOR MDS51-WAVEGATE CONFIGURATION
473:;*************************************************************
474:;
475:;#############################################################################
476:
477:SERVICE_INT1: PUSH    PSW             ; SAVE WORKING REGISTERS IN MAIN
478:             PUSH    ACC             ; PROGRAM
479:             PUSH    B
480:             PUSH    DPL
481:             PUSH    DPH
482:             MOV     PSW,#00001000B  ; SELECT REGISTER BANK 1
483:             MOV     DPTR,#8006H     ; LSB LATCHING COUNTER
484:             MOVX    A,@DPTR
485:             MOV     3DH,A           ; SAVE LSB
486:             INC     DPTR            ; MSB LATCHING COUNTER
487:             MOVX    A,@DPTR
488:             MOV     3EH,A           ; SAVE MSB
489:             LCALL   DATA_SAMPLE     ; GET COUNT FOR EACH LATCHING
490:             LCALL   BIN_BCD         ; CONVERT TO BCD REPESENTATION
491:             LCALL   INTEGRATOR      ; INTEGRATE 10 DATA SAMPLES 10 MSEC*10
492:             POP     DPH
493:             POP     DPL
494:             POP     B
495:             POP     ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                POP     PSW
                RETI
;BCD_SUM ADD TWO'S SIX DIGIT BCD CONTAIN IN THREE BYTE
;           POINTED BY R0 AND R1
;                   FIRST = {FIRST + SECOND}
;           ENTRY: R0 POINT TO LSD FIRST BCD NUMBER
;                   R1 POINT TO LSD SECOND BCD NUMBER
;           EXIT:  RESULTS PLACE TO FIRST BCD NUMBER
;                   C = 1 {OVERFLOW,eg.GREATER THAN 999999}

BCD_SUM:    CLR     C
            MOV     R7,#03H
SUM:        MOV     A,@R0
            ADDC    A,@R1
            DAA
            MOV     @R0,A
            INC     R0
            INC     R1
            DJNZ    R7,SUM
            RET

;INTEGRATOR SUM 5 TIMES SIX DIGIT BCD NUMBER (20MS x 5 = 100 MS)
;           INTEGRATOR (BCD NUMBER)
;           {
;           SUM = 000000;
;           WHILE  ($40 < 10) DO
;              SUM = SUM + INPUT BCD NUMBER;
;           }
;           TRANSFER (SUM);   /* MOVE SUM TO DISPLAY BUFFER */
;           CLEAR ($40);

INTEGRATOR: MOV     A,#05H
            CJNE    A,40H,INTEGRATED

            LCALL   SAVE_MEMORY

            MOV     40H,#00H            ;START AGAIN
            MOV     41H,#00H            ; CLEAR SUM
            MOV     42H,#00H
            MOV     43H,#00H

INTEGRATED: MOV     R0,#41H             ;FIRST BCD NUMBER (MUST BE CLEARED)

            MOV     R1,#36H             ;SECOND BCD NUMBER
            LCALL   BCD_SUM
            INC     40H                 ; (40H)=((40H)+1)
            RET

;BCD_ASCII CONVERT BCD NUMBER TO ASCII CODE
;           ENTRY: A {LOW NIBBLE ONLY}
;           EXIT: A

BCD_ASCII: ANL      A,#0FH              ;GET ONLY LOW NIBBLE
```

เอกสารนี้จัดเก็บไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
551:            ADD       A,#30H            ;CONVERT TO ASCII CODE
552:            RET
553:
554:;BUFFER_ASCII CONVERT BCD IN BUFFER START $30 - $35
555:;          ENTRY: BCD NUMBER IN TRANSMIT BUFFER
556:;          EXIT: ASCII CODE IN TRANSMIT BUFFER
557:
558:BUFFER_ASCII: MOV     R7,#06H
559:            MOV     R0,#30H
560:NEXT_ASCII:  MOV     A,@R0
561:            LCALL   BCD_ASCII
562:            MOV     @R0,A
563:            INC     R0
564:            DJNZ    R7,NEXT_ASCII
565:            RET
566:
567:;SEND_BUFFER SEND ASCII CODE IN TRANSMIT BUFFER TO SERIAL PORT
568:;          THEN DATA SEPERATOR /32/32
569:;          ENTRY: ASCII CODE IN TRANSMIT BUFFER
570:;          EXIT: NO
571:
572:SEND_BUFFER: MOV     R7,#06H
573:            MOV     R0,#35H
574:NEXT_BYTE:  MOV     A,@R0
575:            LCALL   SEND      ;SEND OUT TO SERIAL PORT
576:            DEC     R0
577:            DJNZ    R7,NEXT_BYTE
578:            MOV     A,#20H
579:            LCALL   SEND
580:            MOV     A,#20H
581:            LCALL   SEND
582:            RET
583:
584:;TRANSFERS MOV 3 BYTE IN DATA BUFFER POINTED BY R0 TO TRANSMIT BUFFER
585:;          ENTRY: BCD NUMBER 6 DIGIT
586:;          EXIT: BCD NUMBER 6 DIGIT IN TRANSMIT BUFFER
587:
588:TRANSFERS:   MOV     R7,#03
589:            MOV     R1,#30H
590:TRANS2:     MOV     A,@R0
591:            MOV     R6,A
592:            ANL     A,#0FH
593:            MOV     @R1,A
594:            INC     R1
595:            MOV     A,R6
596:            SWAP    A
597:            ANL     A,#0FH
598:            MOV     @R1,A
599:            INC     R0
600:            INC     R1
601:            DJNZ    R7,TRANS2
602:            RET
603:
604:;COMPARE TWO 16 BIT BINARY NUMBER
605:;          ENTRY: R0 FIRST OPERAND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
606:;                R1 SECOND OPERAND
607:;          EXIT: C=1 FIRST OPERAND < SECOND OPERAND
608:;                C=0 FIRST OPERAND >= SECOND OPERAND
609:
610:COMPARE:       CLR     C              ;
611:               MOV     A,@R0
612:               SUBB    A,@R1
613:               INC     R0
614:               INC     R1
615:               MOV     A,@R0
616:               SUBB    A,@R1
617:               RET
618:
619:
620:;DATA_SAMPLE GET DATA_SAMPLE FROM NON-STOP-FLYING COUNTER
621:;              COUNTL =  $44
622:;              COUNTH =  $45
623:;              OCOUNTL = $46
624:;              OCOUNTH = $47
625:;              CMAX    = $0000 [65536] ignor bit 16
626:;           ENTRY: $3D,$3E
627:;           EXIT:  $3D,$3E
628:
629:COUNTL:       EQU     44H
630:COUNTH:       EQU     45H
631:OCOUNTL:      EQU     46H
632:OCOUNTH:      EQU     47H
633:
634:DATA_SAMPLE: MOV     COUNTL,3DH      ; MOVE FROM OL TO COUNT
635:              MOV     COUNTH,3EH
636:              MOV     R0,#COUNTL
637:              MOV     R1,#OCOUNTL
638:              LCALL   COMPARE
639:              JNC     DIFFERENCE
640:; OVERFLOW FIND TRUE COUNT
641:              CLR     C               ; OCOUNT = [CMAX-OCOUNT]+COUNT
642:              MOV     R1,#OCOUNTL
643:              MOV     A,#00H          ; THIS PORTION PERFORMS CMAX-OCOUNT
644:              SUBB    A,@R1
645:              MOV     @R1,A
646:              INC     R1
647:              MOV     A,#00H
648:              SUBB    A,@R1
649:              MOV     @R1,A
650:
651:              CLR     C               ; THIS PORTION PERFORMS OCOUNT=
652:              MOV     R1,#OCOUNTL     ; [CMAX-OCOUNT] + OCOUNT
653:              MOV     R0,#COUNTL
654:              MOV     A,@R0
655:              ADD     A,@R1
656:              MOV     @R1,A
657:              INC     R0
658:              INC     R1
659:              MOV     A,@R0
660:              ADDC    A,@R1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
661:              MOV     @R1,A
662:              SJMP    CHANGE
663:
664:DIFFERENCE:   CLR     C             ; OCOUNT = COUNT - OCOUNT
665:              MOV     R0,#COUNTL
666:              MOV     R1,#OCOUNTL
667:              MOV     A,@R0
668:              SUBB    A,@R1
669:              MOV     @R1,A
670:              INC     R0
671:              INC     R1
672:              MOV     A,@R0
673:              SUBB    A,@R1
674:              MOV     @R1,A
675:CHANGE:       MOV     3DH,OCOUNTL
676:              MOV     3EH,OCOUNTH
677:              MOV     OCOUNTL,COUNTL
678:              MOV     OCOUNTH,COUNTH
679:              RET
680:
681:;SAVE_MEMORY SAVE DATA SAMPLES TO EXTERNAL DATA MEMORY
682:;             START AT $0000H TO $7FFFH ,eg. 32 KILOBYET
683:;             EVERY 100 mS THEN
684:;                   POINTER = POINTER+3
685:;             ENTRY: 16 BIT POINTER REGISTER $48 FOR LOW BYTE
686:;                                           $49 FOR HIGH BYTE
687:;                   DATA SAMPLE $41, $42, $43 (SIX DIGIT BCD NUMBER)
688:;
689:;             EXIT:  SET BIT 7 OF 3RD BYTE OF EVERY DATA SAMPLE
690:;                                                 MEMORY FULL FLAG
691:
692:
693:SAVE_MEMORY: MOV     DPL,SAVE_L
694:              MOV     DPH,SAVE_H      ; GET EXTERNAL MEMORY ADDRESS
695:              MOV     A,41H
696:              MOVX    @DPTR,A         ; SAVE 1ST BYTE
697:              INC     DPTR            ; NEXT BYTE
698:              MOV     A,42H
699:              MOVX    @DPTR,A         ; SAVE 2ND BYTE
700:              INC     DPTR
701:              MOV     A,43H
702:              MOVX    @DPTR,A         ; SAVE 3RD BYTE
703:              INC     DPTR
704:
705:              MOV     A,#0FFH         ; SAVE TERMINATOR (-1)
706:              MOVX    @DPTR,A
707:
708:SAVE2:        MOV     SAVE_L,DPL
709:              MOV     SAVE_H,DPH        ; SAVE NEXT POINTER
710:              MOV     C,LED           ; GET BLINK BIT ADDRESS
711:              CPL     C
712:              MOV     LED,C           ; INTERRUPT ACTIVE
713:              MOV     P1.1,C          ; INDICATOR BLINK
714:              RET
715:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;                                                                           %
;             THE FOLLOWING ROUTINE ARE MODULE TEST START AT $7000          %
;                                                                           %
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;
;TEST 16 BIT BINARY TO BCD CONVERTER FUNCTION
;         START.TEST PROGRAM AT $7000
;         BCD DATA


TEST4:          MOV       3DH,#0FFH
          MOV       3EH,#0FFH
          LCALL     BIN_BCD
          MOV       R0,#36H
          LCALL     TRANSFER
TEST5:          MOV       R0,#30H
          LCALL     SCAN_DISP
          SJMP      TEST5
;TEST6 TEST FUNCTION TRANSFER
;
TEST6:    MOV       36H,#10H
          MOV       37H,#32H
          MOV       38H,#54H
          MOV       R0,#36H
          LCALL     TRANSFER
TEST8:    MOV       R0,#30H
          LCALL     SCAN_DISP
          SJMP      TEST8
;TEST9 TEST INTERUPT SERVICE ROUTINE INT1 {EDGE TRIGGER}
;
;

MAIN:                         ; INITIALIZED SYSTEM PARAMETER
          MOV       40H,#00H  ; CLEAR WINDOW COUNTER
          MOV       41H,#00H  ; CLEAR SUM
          MOV       42H,#00H
          MOV       43H,#00H
          MOV       OCOUNTL,#00H
          MOV       OCOUNTH,#00H
          MOV       COUNTL,#00H
          MOV       COUNTH,#00H
          MOV       48H,#00H           ; SET EXTERNAL DATA MEMORY POINTER
          MOV       49H,#10H           ;  START AT $1000  TOP OF FIFO BUFFER
          MOV       A,#0FFH
          MOV       DPTR,#TOP_FIFO
          MOVX      @DPTR,A            ; SAVE TERMINATOR (-1) FIRST
          MOV       SP,#6FH            ; STACK AVAILABLE 16 BYTES ($70-$7FH)
          CLR       XOFF_FLAG          ; CLEAR XOFF_FLAG ( EG. READY TO SEND)

          MOV       PSW,#00000000B     ; SELECT REGISTER BANK 0
          MOV       IEC,#10000000B     ; ENABLE ALL CONTROL BIT

          MOV       P1,#10000000B      ; OFF INDICATOR PORT 1.0-1.7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL   INIT              ; INITIALIZED SERIAL PORT
            SETB    IT1               ; FALLING EDGE TRIGGER INT1


;SEND COLD BOOT MESSAGE TO TERMINAL

MAIN1:      LCALL   RECIVE            ; GET SPACE FROM TERMINAL FOR DTR FLAG
            CJNE    A,#20H,MAIN1

            LCALL   CR_LF             ; NEW LINE
            MOV     DPTR,#COLD_MASSEGE
            LCALL   SEND_STRING

MAIN2:              LCALL   OUT_PROMPT


; WHILE RECIVE BUFFER NOT READY DO

            SETB    P1.0              ; ON BIT 0 OF PORT 1
            LCALL   RECIVE            ; UNTIL GET COMMAND
            CLR     P1.0
            PUSH    ACC
            LCALL   SEND              ; ECHO COMMAND LETTER
            LCALL   CR_LF             ; OFF BIT 0 PORT1
            POP     ACC

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;                                                                          %
; BRANCH TO SERVICE ROUTINE ACCORDING TO RECIVED COMMAND                    %
;                                                                          %
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


            CJNE    A,#53H,BRANCH1    ; S COMMAND
            LJMP    SERVICE_S
BRANCH1:    CJNE    A,#54H,BRANCH2    ; T COMMAND
            LJMP    SERVICE_T
BRANCH2:    LJMP    MAIN

SERVICE_T: NOP

;************************************************************************
; SERVICE S COMMAND
;           REPEAT
;               GET BYTE FROM SERIAL PORT
;           UNTIL DATA READY
;


SERVICE_S: MOV      DPTR,#WAIT
            LCALL   SEND_STRING
            SETB    P1.7
TRIGGER:    JNB     P1.7,S1          ; WAIT FOR TRIGGER INPUT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
826:            SJMP      TRIGGER
827:
828:S1:         MOV       DPTR,#TRIG
829:            LCALL     SEND_STRING
830:
831:            SETB      EX1             ; ENABLE EXTERNAL INTERRUPT 1
832:S2:         LCALL     RECIVES         ; GET BYTE FROM SBUF
833:
834:            CJNE      A,#XON,S3       ; CHECKING XON, XOFF CHARACTER
835:            CLR       P1.2
836:
837:            CLR       XOFF_FLAG       ;
838:            SJMP      SEND_SYNC       ; SEND DATA AGAIN
839:S3:         CJNE      A,#XOFF,S4
840:            SETB      P1.2
841:
842:            SETB      XOFF_FLAG       ; XOFF REIVED
843:            SJMP      S2              ;  GO BACK GET BYTE
844:S4:         CJNE      A,#00H,S5       ;   CHECK TIME-OUT
845:            SJMP      STOP_INT        ; BREAK CHARACTER BACK TO MAIN
846:S5:         CJNE      A,#0FFH,S2      ; BACK TO RECIVES
847:
848:            JB        XOFF_FLAG,S2    ; WAIT FOR XON
849:            SJMP      SEND_SYNC       ; TIME-OUT ACTIVE START SENDS OUT
850:
851:;STOP_INT STOP EXTERNAL INTERRUPT THEN MAIN
852:
853:STOP_INT:   CLR       EX1
854:            CLR       P1.1            ; OFF INTERRUPT LED
855:            SJMP      MAIN2
856:
857:;SEND_SYNC SEND DATA SAMPLE IN EXTERNAL DATA MEMORY TO IBM
858:;           CHECK SEND PIONTER AND SAVE POINTER
859:;               WHILE SEND POINTER < SAVE POINTER DO
860:;                   SEND DATA SAMPLE TO IBM
861:;               END,WHILE
862:;           ENTRY: XON
863:;           EXIT: UPDATE DATA SAMPLE POINTER
864:
865:SEND_SYNC: MOV       R7,#01H       ; DELAY BETWEEN SUCCESIVE DATA SAMPLE
866:DEL1:      MOV       R6,#00H
867:DEL2:      DJNZ      R6,DEL2
868:           DJNZ      R7,DEL1
869:
870:
871:SEND1:     MOV       DPTR,#TOP_FIFO ; CHECK TERMINATOR (-1)
872:           MOVX      A,@DPTR
873:           CJNE      A,#0FFH,SEND2  ; LOOP IF NO DATA IN FIFO BUFFER
874:           SETB      P1.3
875:           SJMP      SEND1
876:
877:
878:SEND2:     CLR       P1.3           ; GET DATA SAMPLE FROM TOP OF FIFO BUFFER
879:           MOV       R0,#4CH        ;  PUT DATA SAMPLE TO PREBUFFER
880:           MOVX      A,@DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
881:            MOV     @R0,A
882:            INC     R0
883:            INC     DPTR
884:            MOVX    A,@DPTR
885:            MOV     @R0,A
886:            INC     R0
887:            INC     DPTR
888:            MOVX    A,@DPTR
889:            MOV     @R0,A
890:            MOV     R0,#4CH         ; READY TO SEND
891:            LCALL   TRANSFERS
892:            LCALL   BUFFER_ASCII
893:            LCALL   SEND_BUFFER     ; SENDING CURRENT DATA SAMPLE
894:
895:           ;CLR     EX1             ; DISABLE EXT INT1
896:
897:            LCALL   DELETE_BYTE
898:            LCALL   DELETE_BYTE
899:            LCALL   DELETE_BYTE
900:
901:           ;SETB    EX1             ; RESOTRE EXT INT1
902:            LJMP    S2              ; GO BACK HANDSHAKE          XON,XOFF
903:
904:
905:;DELETE_BYTE DELETE ONE BYTE AT TOP OF FIFO BUFFER
906:;
907:;            ENTRY: TERMINATOR IN THE BOTTOM OF FIFO BUFFER
908:;            EXIT: NO
909:
910:DELETE_BYTE: MOV     DPTR,#TOP_FIFO
911:DEL_1:      PUSH    DPL             ; SAVE DPTR
912:            PUSH    DPH
913:            INC     DPTR
914:            MOVX    A,@DPTR         ; (DPTR) <---- (DPTR+1)
915:            POP     DPH
916:            POP     DPL
917:            MOVX    @DPTR,A
918:            CJNE    A,#0FFH,DEL_2 ; IF FOUND TERMINATOR THEN EXIT
919:            MOV     SAVE_L,DPL
920:            MOV     SAVE_H,DPH
921:            RET
922:
923:DEL_2:      INC     DPTR            ; ELSE MOVE NEXT BYTE
924:            SJMP    DEL_1
925:
926:;SEND_STRING SEND STRING CONSTANT TO TERMINAL
927:;           ENTRY:   DPTR
928:;           EXIT:    FOUND EOS
929:;
930:SEND_STRING:  CLR     A
931:            MOVC    A,@A+DPTR
932:            CJNE    A,#EOS,SEND_STRING1
933:            RET
934:
935:SEND_STRING1: PUSH    DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PUSH    DPH
                LCALL   SEND
                POP     DPH
                POP     DPL
                INC     DPTR
                SJMP    SEND_STRING

;OUT_PROMPT SEND command> TO TERMINAL
;
OUT_PROMPT:     MOV     DPTR,#PROMPT
                LCALL   SEND_STRING
                RET

;*************************************************************************
;         STRING CONSTANT AREA
;*************************************************************************

COLD_MASSEGE: DFB  "Data Acquisition System for HPLC",CR,LF
              DFB  "Link Active....",CR,LF,EOS
PROMPT:       DFB  "Command>",EOS
WAIT:         DFB  "Wait for trigger input...",CR,LF,EOS
TRIG:         DFB  "Trigger !",CR,LF,EOS

END
```

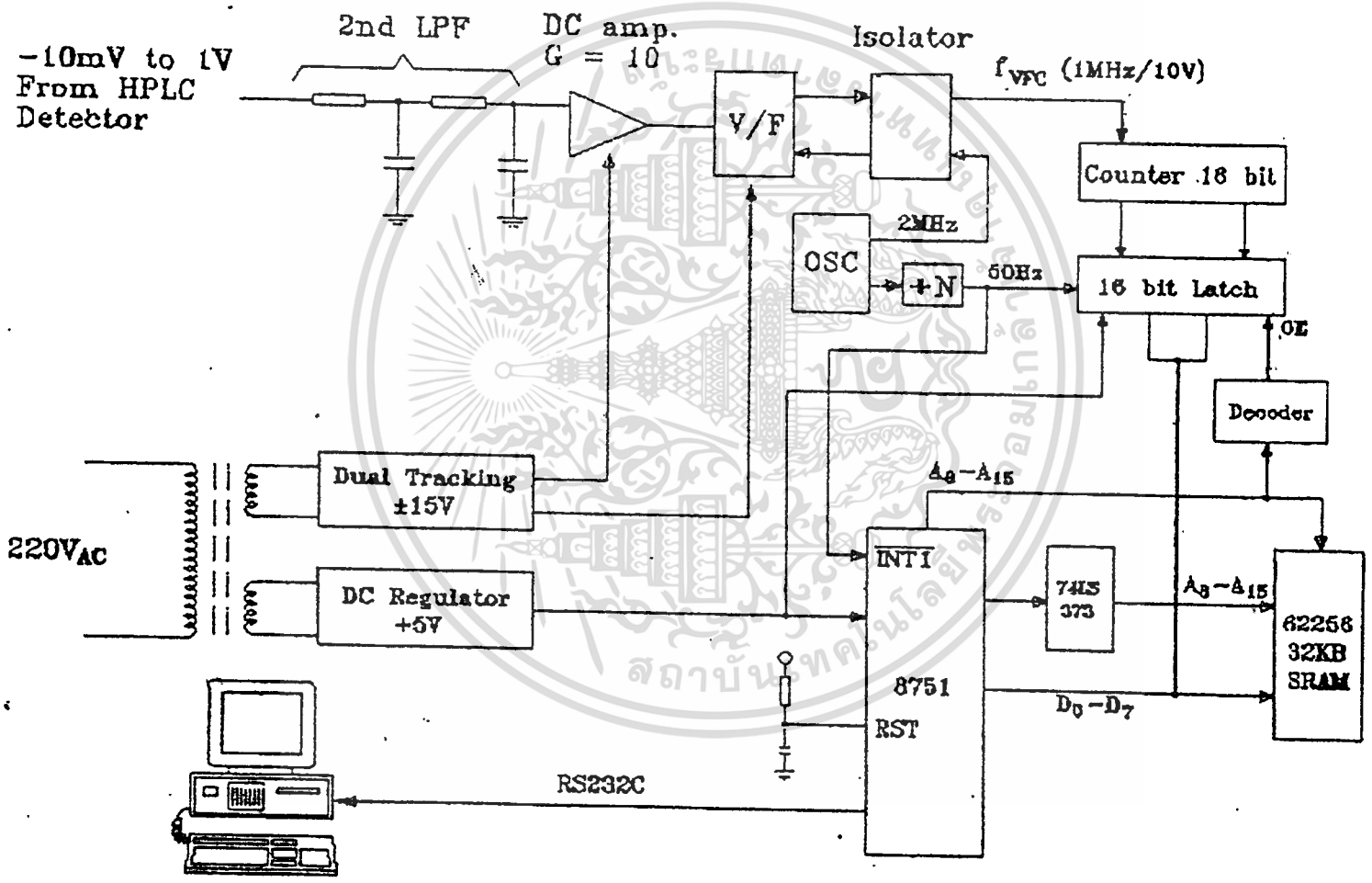
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

วงจรเครื่องเก็บและรับส่งข้อมูล

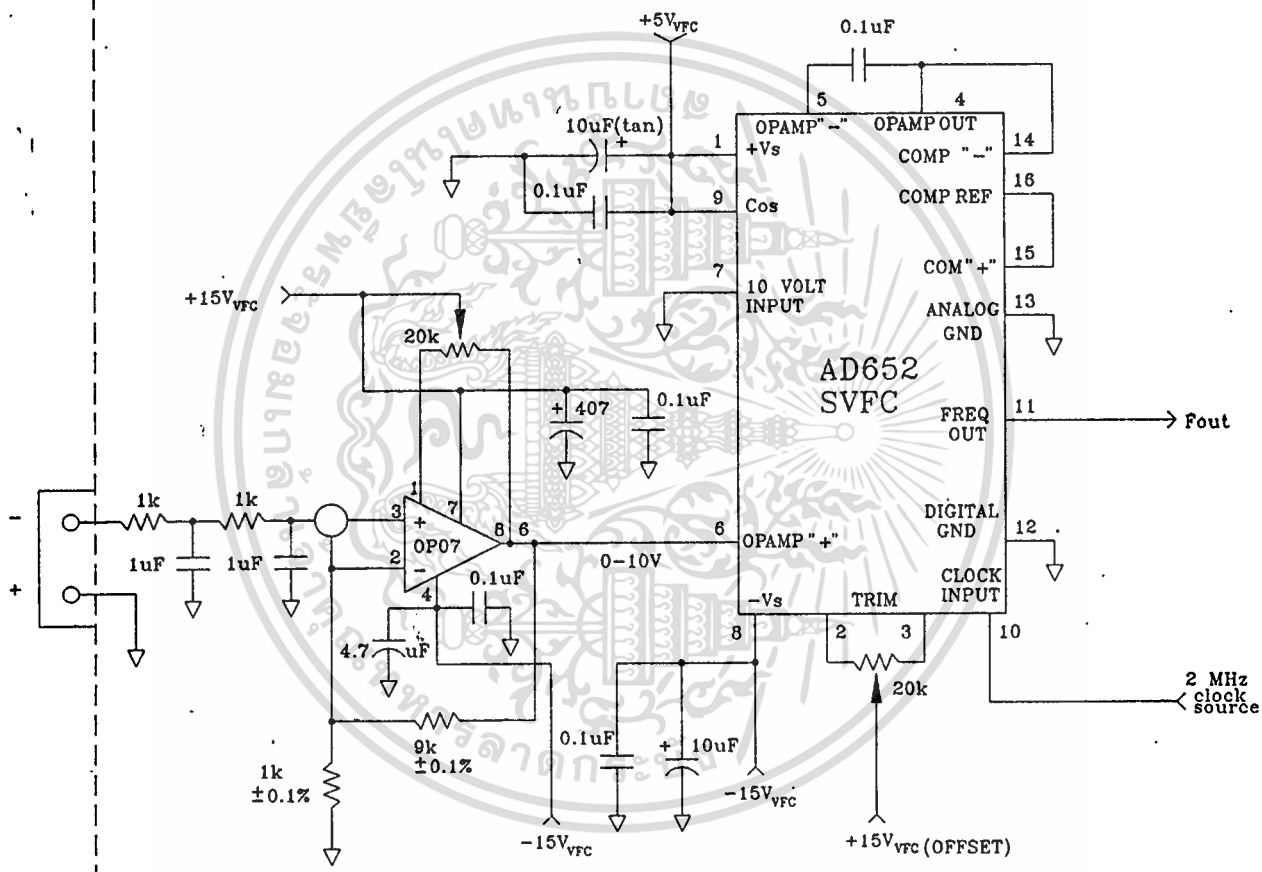
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรการทำงานของชุดเก็บและรับส่งข้อมูล

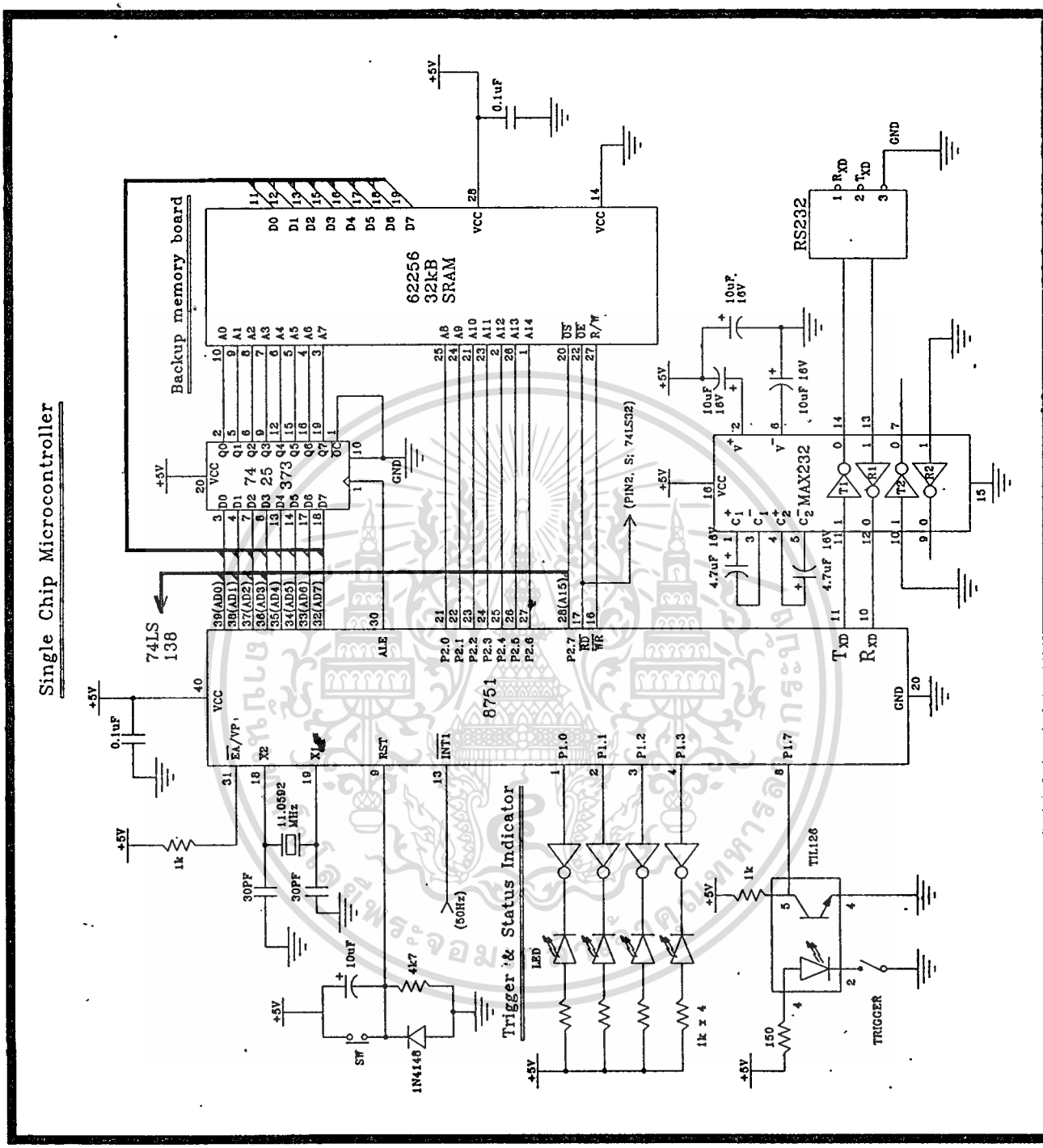


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANALOG PART
Data Acquisition
Synchronous Voltage to Frequency converter



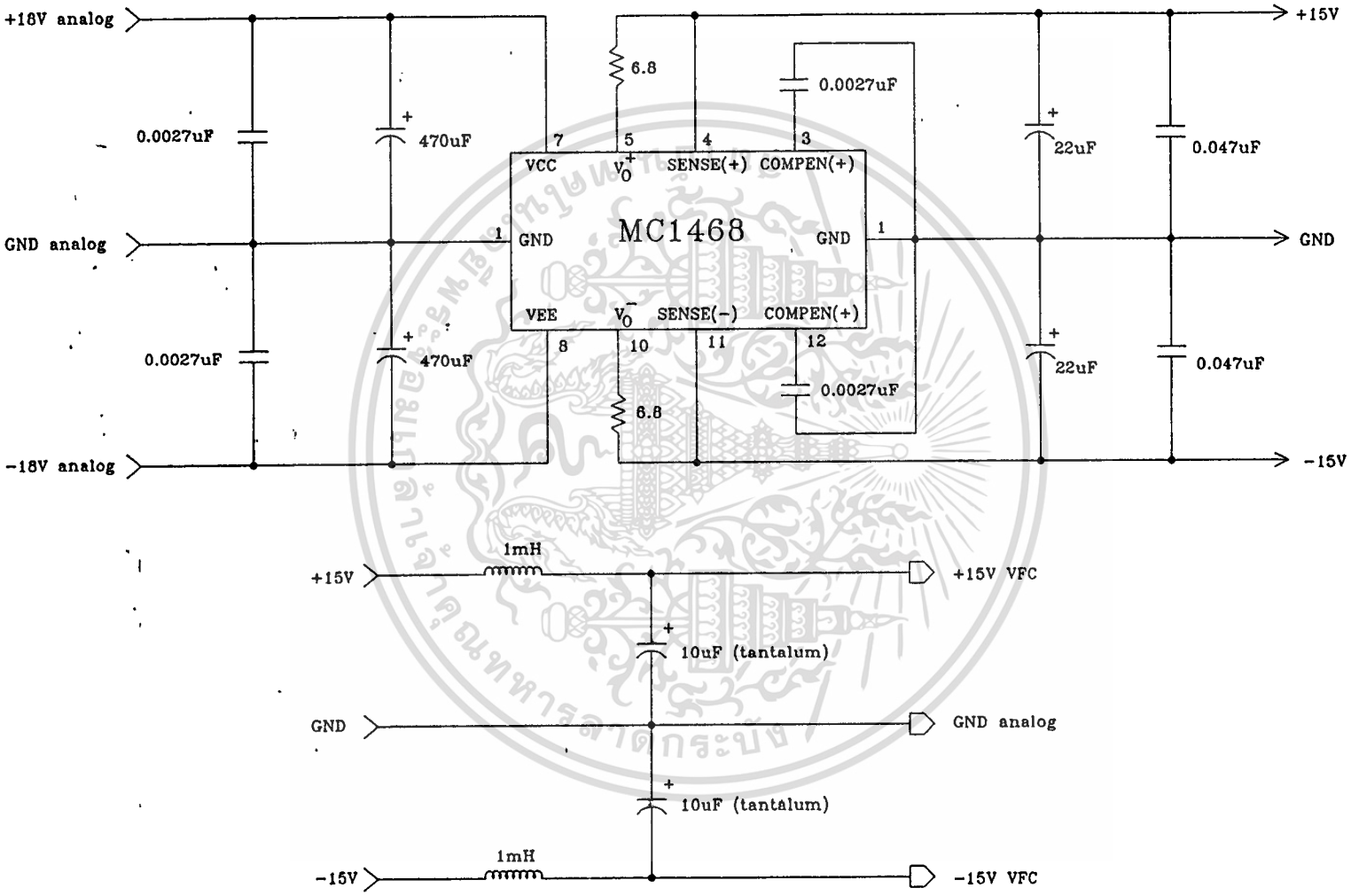
- This circuit is designed for negative input voltage.
- AD652 input voltage range = 0-10V.
- All capacitors use low inductance.



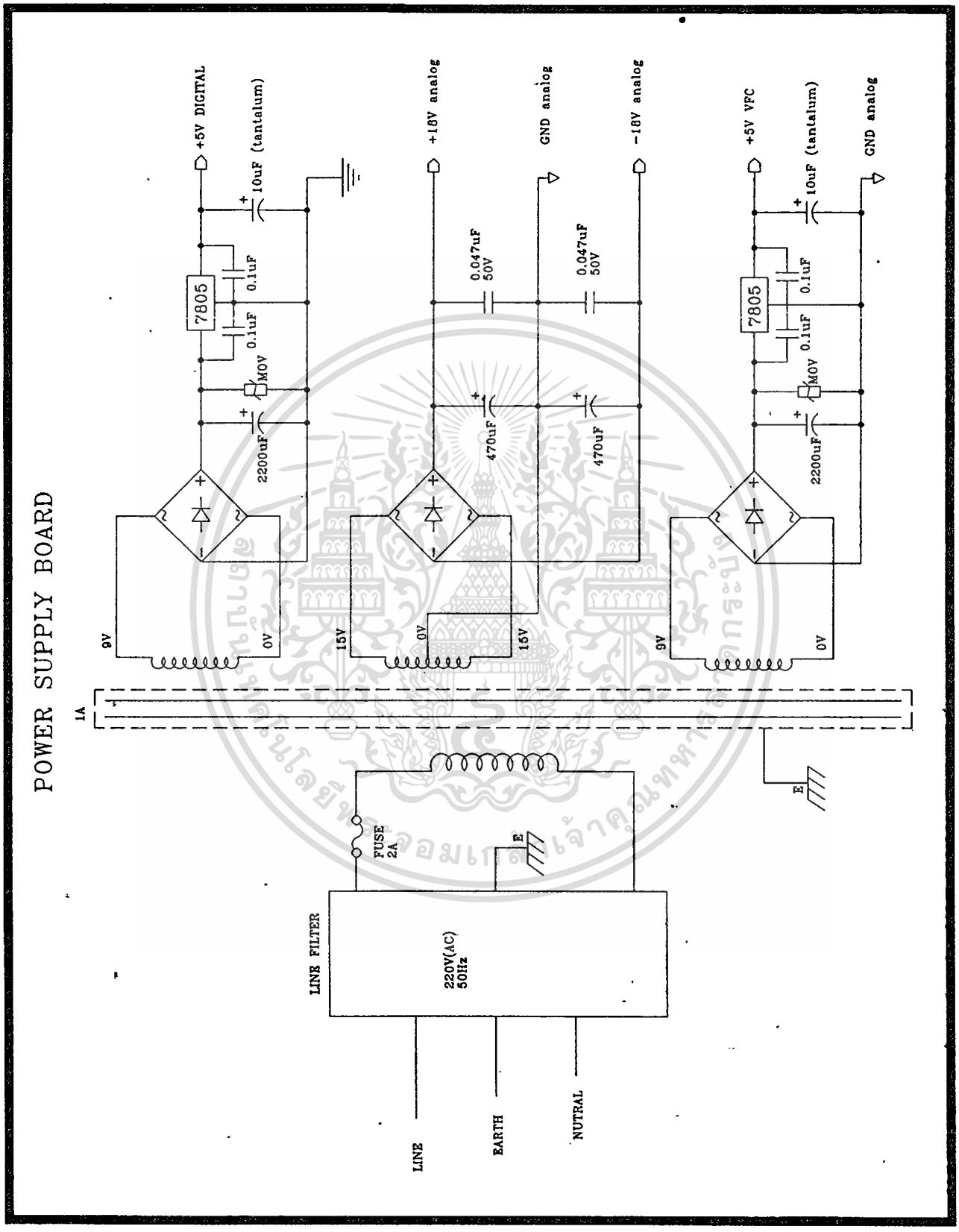
วงจรของภาคประมวลผลและรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Dual Tracking Regulator for precision analog circuit

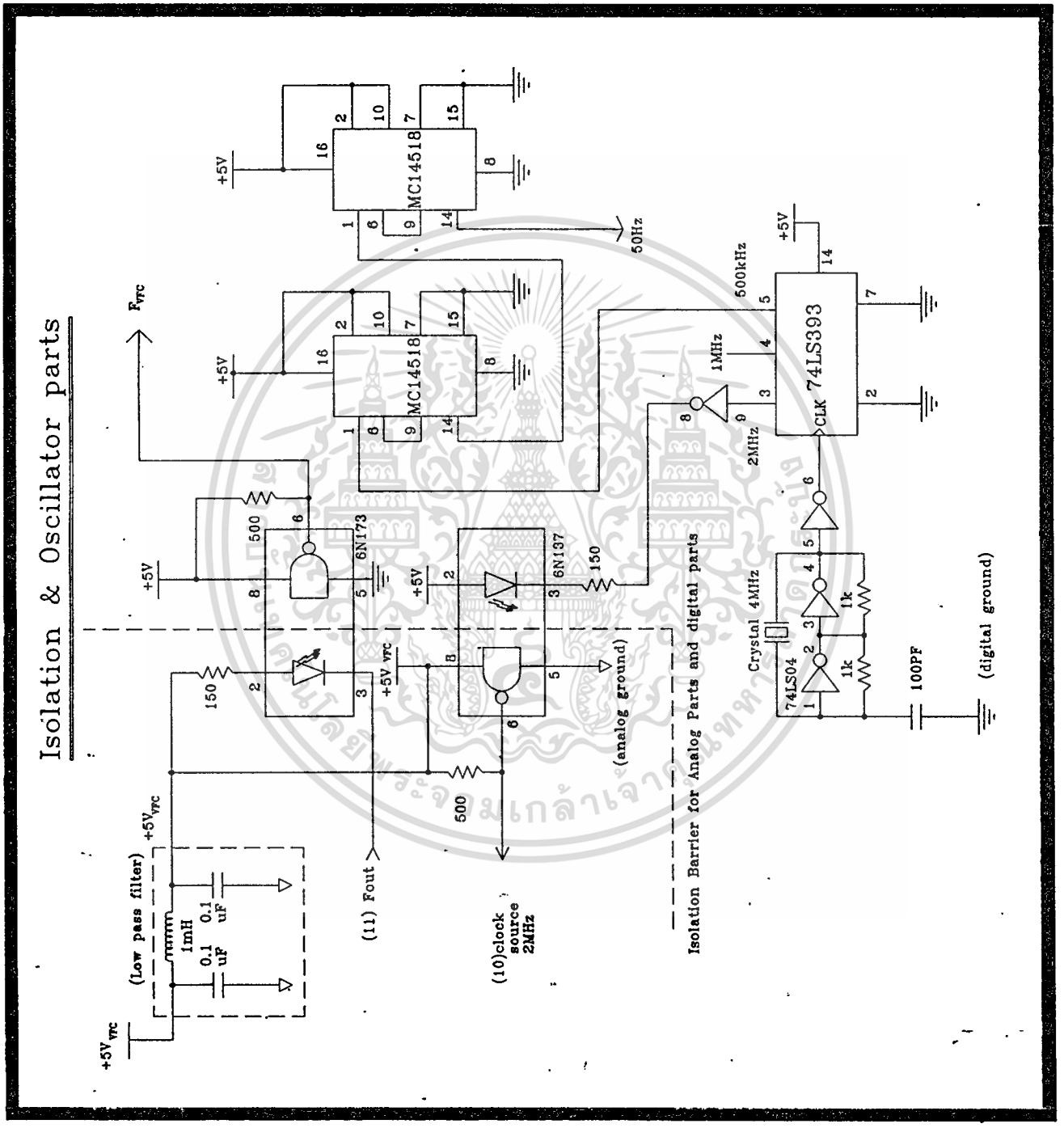


- All capacitor uses ⩾ 35V



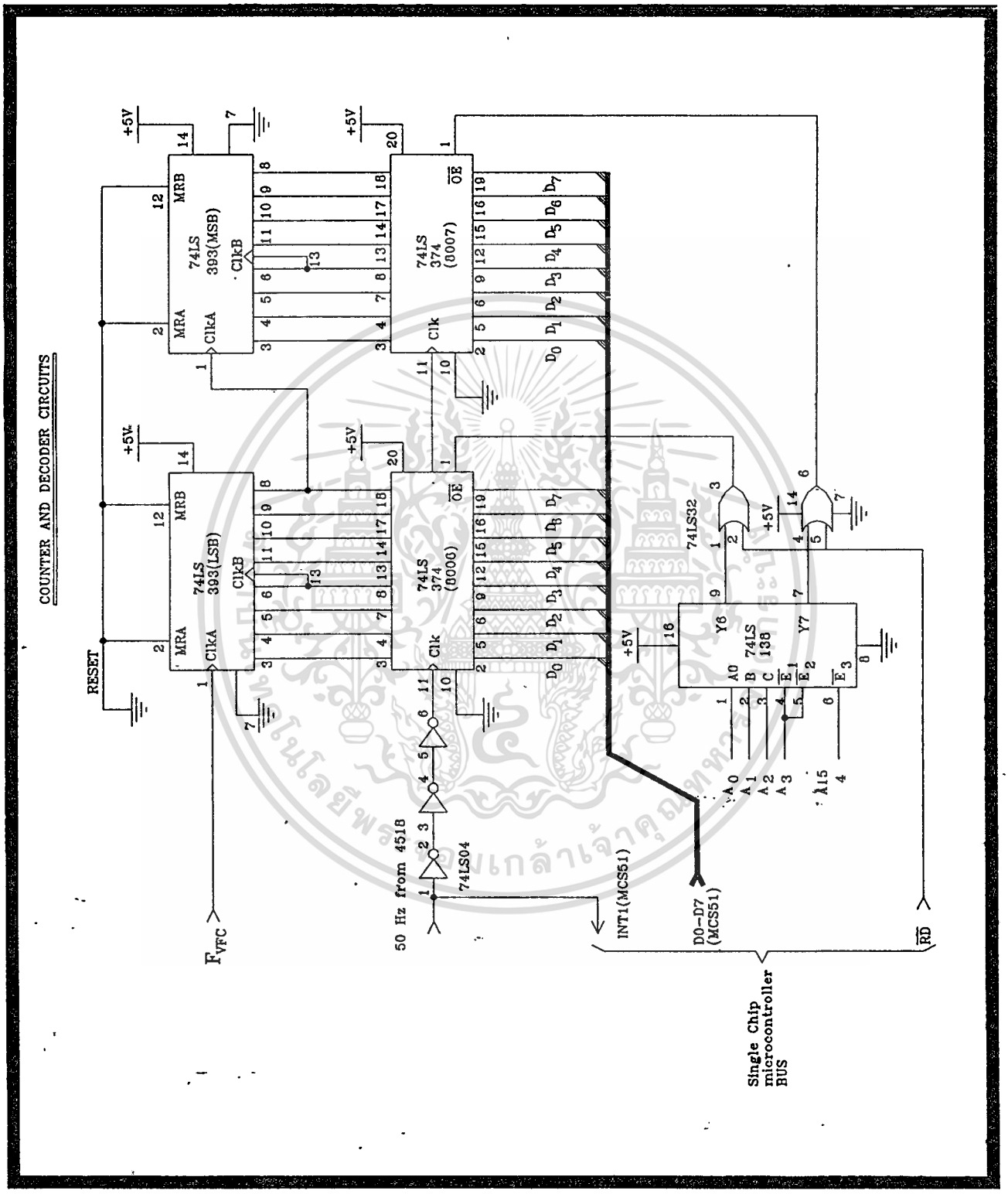
วงจรของภาคจ่ายไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรของภาคสร้างสัญญาณนาฬิกาขนาด 2 MHz และ 50 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรนับและคงสถานะข้อมูล

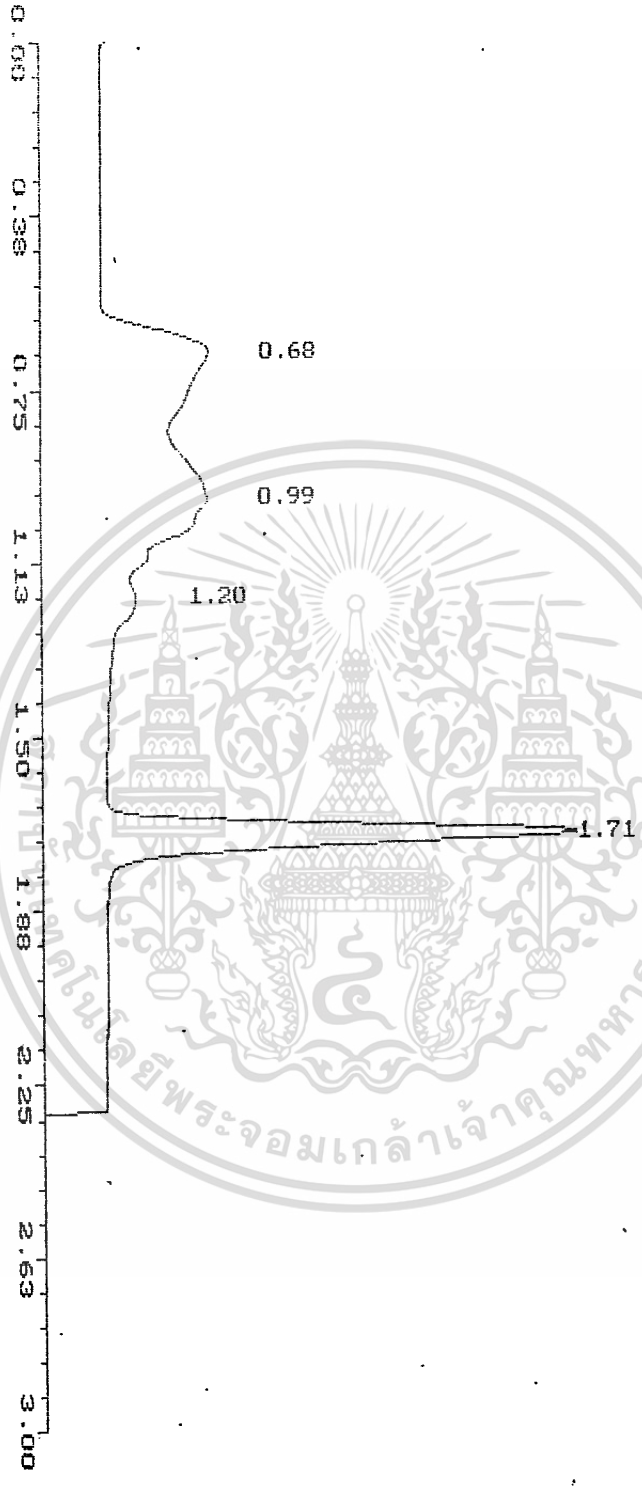
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ.

ตัวอย่างการรายงานผลการวิเคราะห์จากซอฟต์แวร์ที่พัฒนาขึ้น
และการรายงานผลจากเครื่องวิเคราะห์ผลข้อมูลของต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรายงานผลการวิเคราะห์จากซอฟท์แวร์ที่พัฒนาขึ้น



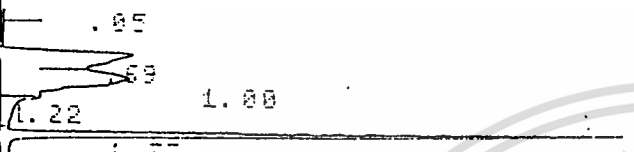
Peak	RT	AREA
1	0.68	18705
2	0.99	19026
3	1.20	2525
4	1.71	23449

Name	RT	PPM
Caffeine	1.71	123.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรายงานผลของเครื่องประมวลผลข้อมูลจากต่างประเทศ

HEL A INJECT 00:47:05



00:47:05 CH= "A" PS= 1.

1. METHOD 0. RUN 17 INDEX 17

#	AREA%	RT	AREA	BC
1	0.047	0.05	62	01
2	27.986	0.69	36662	02
3	32.595	1.	42700	02
4	5.452	1.22	7142	02
5	33.898	1.73	44408	02
6	0.022	2.13	29	03
T	100.		131003	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ประวัติผู้เขียน

นายจิรวัฒน์ ตันตราจิณ เกิดเมื่อวันที่ 2 เมษายน พ.ศ.2510 ที่กรุงเทพฯ เข้าศึกษาในระดับปริญญาตรีที่ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โดยโครงการ พสวท. และสำเร็จการศึกษาระดับวิทยาศาสตรบัณฑิต สาขาฟิสิกส์ ในปี พ.ศ.2532 จากนั้นได้เข้าศึกษาต่อระดับปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีความสามารถในการเขียนซอฟต์แวร์ภาษาปาสคาล และซอฟต์แวร์ภาษาวิชวลเบสิก บนวินโดส์ ปัจจุบันทำงานในตำแหน่ง นักฟิสิกส์ ที่สำนักงานพัฒนาเทคโนโลยี บริษัท พรีเมียร์โกลเบิลคอร์เปอเรชั่น จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้