การควบคุมอุปกรณ์ GPIB ผ่านเครือข่ายอินเตอร์เน็ต

A GPIB INSTRUMENTS CONTROL VIA INTERNET

โดย

นาย พิสิฐ สุขลิ้ม

นาย ยุทธนา รอดสีเสน

นาย สุเมธ พิทักษ์ธำรง

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

วัน เดือน ปี 24 ต.ค. 2541

เลขทะเบียน 039159

เลขเรียกหนังสือ T 40398 พ 7781

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039159

ปริญญานิพนธ์ ปีการศึกษา 2540

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมอุปกรณ์ GPIB ผ่านเครือข่ายอินเตอร์เน็ต

ผู้จัดทำ

นาย พิสิฐ	สุขลิ้ม	รหัส	38013236
นาย ยุทธนา	รอดสีเสน	รหัส	38013240
นาย สุเมธ	พิทักษ์ธำรง	รหัส	38013255

..... อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอุปกรณ์ GPIB ผ่านเครือข่ายอินเตอร์เน็ต

A GPIB INSTRUMENTS CONTROL VIA INTERNET

นาย พิสิฐ	สุขลิ้ม	รหัส	38013236
นาย ยุทธนา	รอดสีเสน	รหัส	38013240
นาย สุเมธ	พิทักษ์ธำรง	รหัส	38013255

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ทางผู้จัดทำปริญญานิพนธ์ขอกราบขอบพระคุณ รศ.ดร. มนัส สังวรศิลป์ เป็นอย่างยิ่งซึ่งได้ให้คำแนะนำในการจัดทำโครงงานฉบับนี้ และขอขอบคุณพี่ กิตติ และพี่ ๆ ปริญญาโท ที่คอยช่วยเหลือในด้านต่าง ๆ ขอบคุณน้องแอนที่ช่วยจัดทำด้านเอกสาร

.....

(นาย พิสิฐ สุขลิ้ม)

.....

(นาย ยุทธนา รอดสีเสน)

.....

(นาย สุเมธ พิทักษ์ธำรง)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอุปกรณ์ GPIB ผ่านเครือข่ายอินเตอร์เน็ต

นาย พิสิฐ สุขลิ้ม
นาย ยุทธนา รอดสีเสน
นาย สุเมธ พิทักษ์ธำรง
รศ. ดร. มนัส สังวรศิลป์
ปีการศึกษา 2540

บทคัดย่อ

เนื่องจากความก้าวหน้าทางเทคโนโลยีเครือข่ายคอมพิวเตอร์ในปัจจุบัน ส่งผลให้เครือข่ายอินเตอร์เน็ตแพร่หลาย และได้รับความนิยมเป็นอย่างมาก ซึ่งสามารถนำไปประยุกต์ใช้งานได้มากมาย โครงงานนี้จึงได้ประยุกต์การใช้งานของเครือข่ายอินเตอร์เน็ต เพื่อใช้ประโยชน์ในทางวิศวกรรม โดยนำมาประยุกต์ใช้ในการควบคุมอุปกรณ์เครื่องมือวัด GPIB (IEEE-488) ผ่านเครือข่ายอินเตอร์เน็ต ซึ่งสามารถทำให้ควบคุมชุดเครื่องมือวัดนี้ในระยะไกลได้ จากเครื่องคอมพิวเตอร์ที่ต่ออยู่กับระบบอินเตอร์เน็ต โดยใช้โพรโตคอล TCP/IP และระบบฐานข้อมูลเข้ามาช่วย โดยที่โปรแกรมของผู้ใช้จะส่งข้อมูลไปยังฐานข้อมูล และคอมพิวเตอร์ที่ต่ออยู่กับชุดเครื่องมือวัด จะนำข้อมูลที่ได้รับจากฐานข้อมูลมาประมวลผล ควบคุม และรับผลที่ได้ จากชุดเครื่องมือวัดส่งให้ผู้ใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A GPIB INSTRUMENTS CONTROL VIA INTERNET

Mr. Pitsit Suklim
Mr. Yutthana Rodseesain
Mr. Sumat Pituktumrong
Assoc.Prof.Dr.Sungwarasin Manus
Educational Year 1997

Abstract

Because of, a progress of computer network technology, the Internet system becomes a very popular and widely used in many applications. This project applies the advantage of Internet system to control the GPIB (IEEE-488) instruments. By using TCP/IP protocol and database system, we can control many GPIB instruments from any computers connected to the Internet. Users have to send the command of an instrument to SQL database server. The computer, which interfaces with a GPIB instrument, receives this command from database and process to control them. Moreover it receives the replied (data) of instrument return to database. The user will be provided this data from database.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
บทที่ 1 บทนำ	1
บทที่ 2 IEEE-488 (GPIB)	2
2.1 โครงสร้างของ IEEE-488	3
2.2 ขีดจำกัดของ IEEE-488	3
2.3 รายละเอียดเกี่ยวกับ IEEE-488	4
2.4 ความหมายของสัญญาณต่าง ๆ ภายใน IEEE-488	5
2.4.1 กลุ่มสัญญาณข้อมูล	5
2.4.2 กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interface)	5
2.4.3 กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูล	6
2.5 การเชื่อมต่ออุปกรณ์ต่าง ๆ ในระบบ IEEE-488 BUS	7
2.6 ขบวนการแฮนด์เชค (Handshake Procedure)	8
2.7 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวส่ง	10
2.8 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวรับ	11
2.9 คำสั่งใช้งานของ GPIB	12
2.10 การขอบริการและการตรวจสอบ (Service Request and Polling)	16
2.11 รูปแบบข้อมูล	17
บทที่ 3 เครือข่ายอินเตอร์เน็ต	20
3.1 สถาปัตยกรรมอินเตอร์เน็ต (Internet Architecture)	20
3.2 OSI โมเดล	21
3.3 โครงสร้างชั้นเน็ตเวิร์ค (Network layer structure)	23
3.4 Internet Protocol Standards	24
3.5 Internet IP	25
3.5.1 โครงสร้างแอดเดรส (Address structure)	25
3.5.2 รูปแบบของข้อมูล (Datagrams)	26

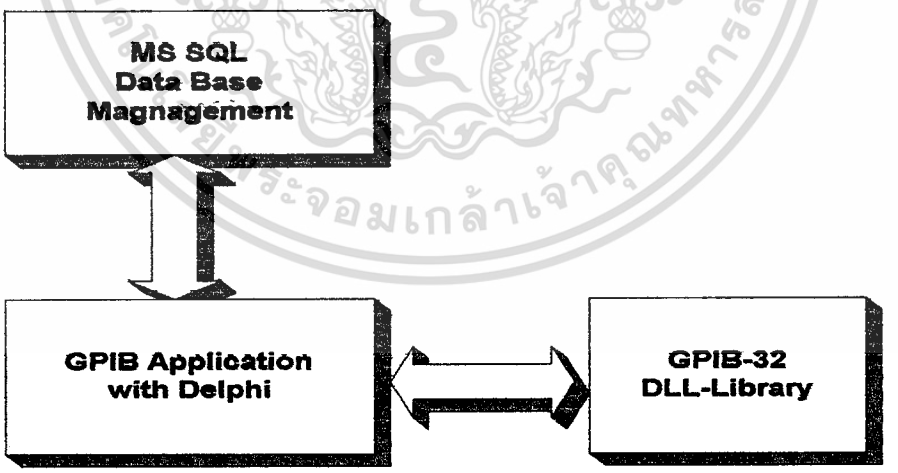
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 การแบ่งส่วนย่อยของข้อมูลและการประกอบขึ้นใหม่
(Fragmentation and Reassambly) 27
3.5.4 การเลือกเส้นทาง (Routing) 29
บทที่ 4 การออกแบบและสร้าง 31
4.1 การสร้างฐานข้อมูล 31
4.2 โปรแกรมควบคุม Programmable Power Supply 33
4.2.1 โปรแกรม User Power Supply 33
4.2.2 โปรแกรม Master Programmable Power Supply 35
4.3 โปรแกรมควบคุม Programmable Function Generator 37
4.3.1 โปรแกรม User Function Generator 37
4.3.2 โปรแกรม Master Programmable Function Generator 39
4.4 โปรแกรมควบคุม Programmable Multimeter 40
4.4.1 โปรแกรม User Multimeter 41
4.4.2 โปรแกรม Master Programmable Multimeter 43
บทที่ 5 การทดลองและผลการทดลอง 44
5.1 การทดลองสั่งงาน Programmable Power Supply (HM8142) 44
5.1.1 การทดลองตั้งค่าแรงดัน V1 และ V2, I1และ I2 โดย User 45
5.1.2 การทดลอง Set สถานะของเอาท์พุทและสถานะเคลียร์ 46
5.2 การทดลองสั่งงาน Programmable Function Generator (HM8130) 46
5.2.1 การทดลองตั้งค่า ความถี่, แอมปลิจูด และชนิดของสัญญาณ 47
5.2.1.1 การตั้งค่าโดย User 47
5.2.1.2 ข้อมูลในตาต้าเบสของ Function Generator 48
5.2.1.3 ค่าที่ปรากฏจริงที่ Programmable Function Generator 48
5.3 การทดลองสั่งงาน Programmable Multimeter (HM8112-2) 49
5.3.1 การทดลองตั้งค่าโดย User 49
5.3.2 ข้อมูลในตาต้าเบสของ Multimeter 50
บทที่ 6 สรุปผลและวิจารณ์ 52
ภาคผนวก 53
โปรแกรม 53
เอกสารอ้างอิง 149

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ความก้าวหน้าทางเทคโนโลยีเครือข่ายคอมพิวเตอร์ ในปัจจุบันส่งผลให้เครือข่ายอินเตอร์เน็ต ได้รับความนิยมจากผู้ใช้อย่างแพร่หลายทั่วโลก ซึ่งมีรูปแบบของการประยุกต์ใช้งานมากมาย ในโครงงานนี้จึงได้นำเอาความสามารถ ในการส่งข้อมูลติดต่อกันในระยะไกล ของระบบเครือข่ายอินเตอร์เน็ต มาประยุกต์ใช้ในการควบคุมชุดเครื่องมือวัดมาตรฐาน IEEE-488 หรือ GPIB (General Purpose Interface Bus) ซึ่งประกอบด้วย HM8142 Programmable Function Generator, HM8130 Programmable Power Supply และ HM8112-2 Programmable Multimeter โดยในโครงงานนี้ได้เขียนโปรแกรม เพื่อสร้างแบบจำลองหน้าปัดของอุปกรณ์ทั้ง 3 ตัวข้างต้น เพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายและสะดวกขึ้น โดยใช้โปรแกรมภาษาเดลไฟล์ (Delphi) ซึ่งเป็นโปรแกรมแบบ visual programming languages ซึ่งสามารถสร้างแอปพลิเคชั่นบนวินโดว์ได้ง่ายและรวดเร็วด้วย Rapid Application Development (RAD) tools ซึ่งจะใช้เดลไฟล์ในการอินเตอร์เฟซกับ dynamic link libraries (DLLs) ของ NI-488 Software เพื่อควบคุมฮาร์ดแวร์ทั้ง 3 ตัวข้างต้น ซึ่งโครงสร้างของซอร์ฟแวร์ แสดงดังรูปที่ 1.1

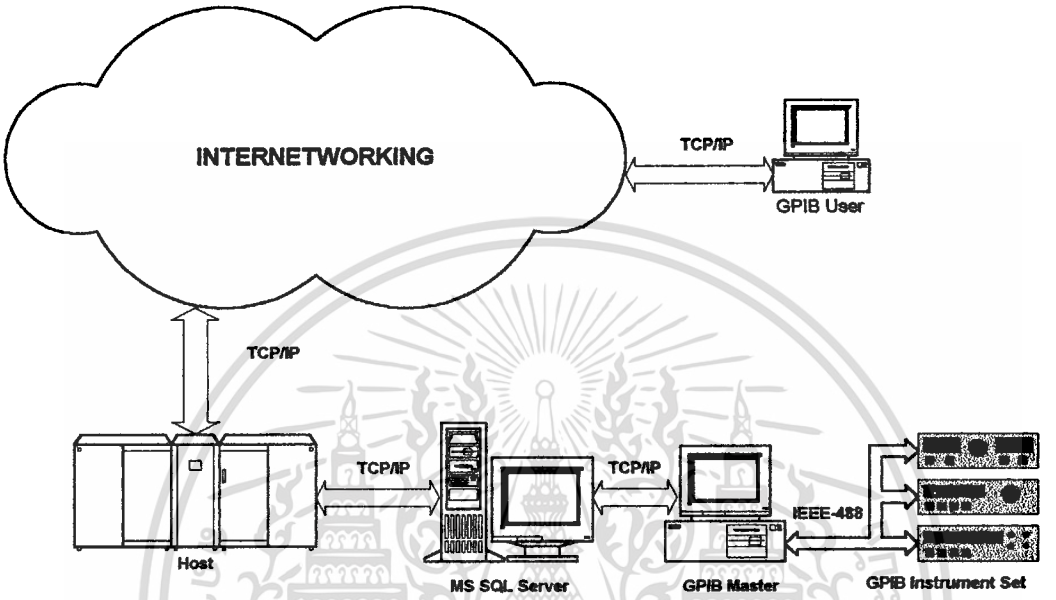


รูปที่ 1.1 โครงสร้างของ ซอร์ฟแวร์ โดยรวม

ส่วนในการเชื่อมต่อ ระบบควบคุมเครื่องวัดเข้ากับ เครือข่ายอินเตอร์เน็ตนั้นจะใช้ Delphi ในการติดต่อกับ MS SQL Sever ซึ่งเป็น software ที่เป็นตัวจัดการกับระบบฐานข้อมูล ที่มีลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ ไคล์เอนท์-เซิร์ฟเวอร์ (Client-Sever) ซึ่งจะใช้โพรโตคอล ICP/IP หรือ Transmission Control Protocol / Internet Protocol เป็นมาตรฐานในการติดต่อของเครือข่ายอินเตอร์เน็ต โดยภาพรวมของระบบแสดงได้ดังรูปที่ 1.2



รูปที่ 1.2 ระบบการควบคุม GPIB ผ่านเครือข่ายอินเตอร์เน็ต

โดยหลักการทำงานโดยรวมของระบบคือ คอมพิวเตอร์ของผู้ใช้ (GPIB User) จะทำการ ส่ง-รับ ข้อมูลกับฐานข้อมูล ซึ่งอยู่ที่เครื่อง MS SQL Sever และคอมพิวเตอร์ที่เป็นตัวอินเตอร์เฟซ (Interface) กับ ชุดเครื่องมือวัด GPIB ซึ่งรับข้อมูลมาจากดาต้าเบส และนำมาประมวลผลสั่งงานชุดเครื่องมือวัด และรับผลตอบสนองที่ได้รับจากเครื่องมือวัด ส่งให้กับดาต้าเบส (Database) เพื่อส่งให้กับคอมพิวเตอร์ของผู้ใช้ (GPIB User) ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
IEEE-488 (GPIB)

2.1 โครงสร้างของ IEEE-488

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

- Talker ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัว แต่จะมีเพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่

- Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัวเช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลาย ๆ ตัวได้

- Controller ทำหน้าที่ควบคุมอุปกรณ์ต่าง ๆ ในระบบ โดยจะกำหนดให้ Talker ทำการส่งข้อมูล หรือ กำหนดให้ Listener ทำการรับข้อมูล

อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งตามหน้าที่ได้ดังนี้

1. ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น

2. ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์ (Printer), เครื่องบันทึก (Recorder) เป็นต้น

3. ทำหน้าที่เป็นทั้ง Talker และ Listener เช่น คอมพิวเตอร์, เครื่องมือวัดที่สามารถ ควบคุมได้จากภายนอก เป็นต้น

4. ทำหน้าที่เป็น Talker Listener และ Controller ในตัวเดียวกัน เช่น คอมพิวเตอร์ที่ทำหน้าที่ ควบคุมระบบ

2.2 ขีดจำกัดของ IEEE-488

1. จำนวนอุปกรณ์ในระบบ (Talker, Listener, Controller) ที่ต่อกับสายสัญญาณ 1 เส้น จะต้องไม่เกิน 15 เครื่อง

2. สายเคเบิ้ลที่ต่อระหว่างอุปกรณ์ จะต้องยาวไม่เกิน 4 เมตร และความยาวรวมของสาย เคเบิ้ลในระบบจะต้องไม่เกิน 20 เมตร

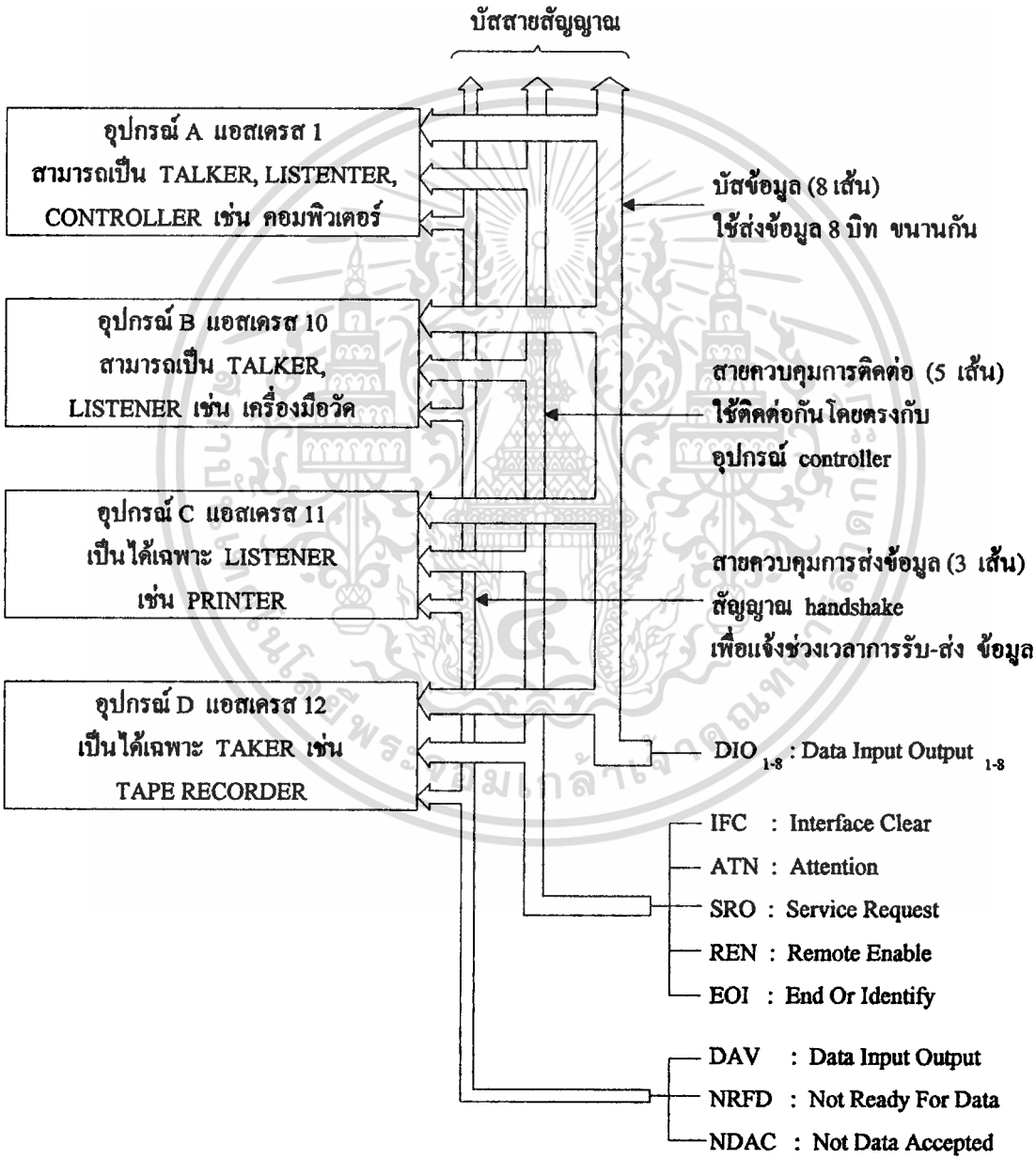
3. ความเร็วในการส่งข้อมูลจะต้องไม่เกิน 1Mb/Sec (1 ล้านไบต์ต่อวินาที)

4. ต้องมีการจ่ายไฟให้กับอุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รายละเอียดเกี่ยวกับ IEEE-488

ลักษณะทางกายภาพ IEEE-488 นั้นคือ เป็นสายสัญญาณแบบ 24 เส้นขนานกัน และ มีขั้วต่ออยู่ทางปลายทั้งสองของสาย เพื่อต่อกับอุปกรณ์หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่ม ขึ้น ในจำนวนสายสัญญาณ 24 เส้น มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนเหลืออีก 8 เส้น ทำหน้าที่กราวนด์ (ground) และ ชีลด์ (shield)



รูปที่ 2.1 แสดงการแบ่งเส้นสายนำสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจำนวนสายที่ใช้นำสัญญาณ 16 เส้นนั้นยังแบ่งได้เป็น 3 ประเภท ตามรูปที่ 2.1 คือ

1. บัสข้อมูล (Data Bus) จำนวน 8 สาย คือ

DI01 - DI08

2. สายสัญญาณควบคุม (Control Line) จำนวน 5 สาย คือ

IFC (Interface Clear)

ATN (Attention)

SRQ (Service Enable)

REN (Remote Enable)

EOI (End or Identify)

3. สายแฮนด์เชค (Hand Shake) 3 สาย คือ

DAV (Data Valid)

NRFD (Not Ready For Data)

NDAC (Not Data Accepted)

2.4 ความหมายของสัญญาณต่าง ๆ ภายใน IEEE-488

ดังที่ได้กล่าวมาแล้วว่าสายสัญญาณต่าง ๆ ใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้ จะอธิบายความหมายของสัญญาณต่าง ๆ ดังนี้

2.4.1 กลุ่มสัญญาณข้อมูล

1. DI01-DI08 (Data Input/Output) สายสัญญาณทั้ง 8 เส้นนี้ ทำหน้าที่เป็นทางผ่านของข้อมูลในระบบ

2.4.2 กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interface)

1. IFC (Interface Clear) เป็นสัญญาณรีเซต หรือ เคลียร์ระบบ กำเนิดได้โดยตัวควบคุม (Controller) เท่านั้น เมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้จะกลับคืนสู่สภาวะเริ่มต้นใหม่ ซึ่งเป็น สภาวะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนแรกเปิดสวิตช์

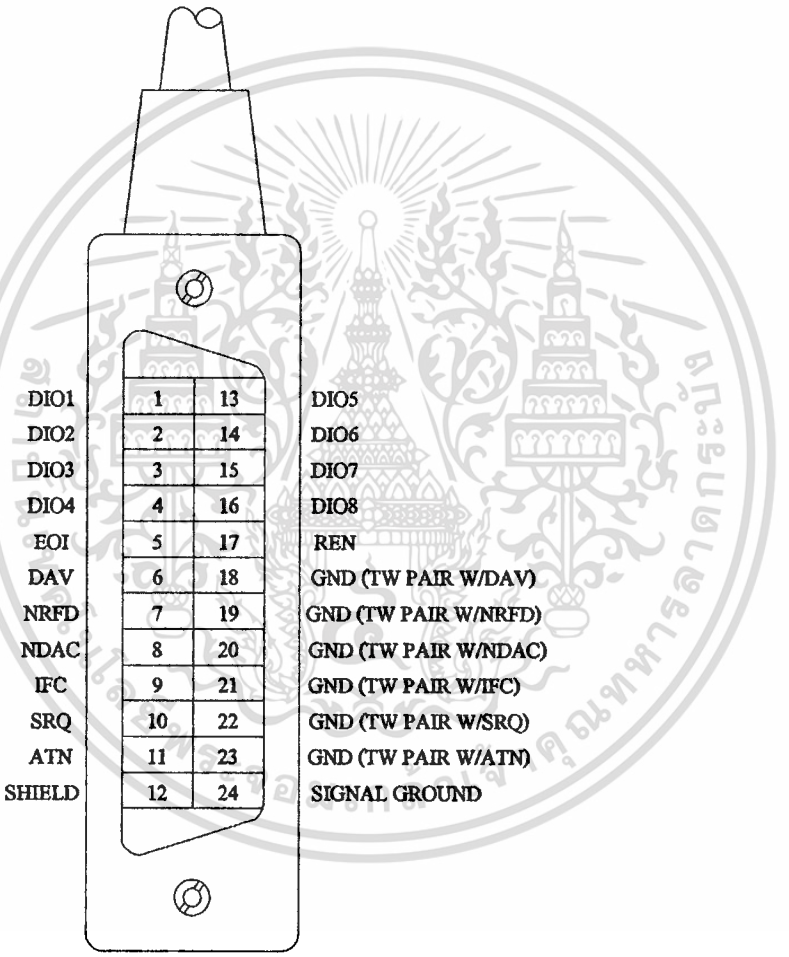
2. ATN (Attention) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ที่เป็นตัวควบคุมเช่นเดียวกัน ใช้ในการสั่งให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป

3. SRQ (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่าง ๆ เพื่อเป็นการบอกแก่ระบบว่าขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. REN (Remote Enable) สัญญาณนี้ เป็นสัญญาณที่ถูกส่งมาจากตัวควบคุมเพียงตัวเดียว เท่านั้น เพื่อใช้สั่งให้อุปกรณ์ต่าง ๆ เปลี่ยนจากโหมดที่ใช้งานปกติ มาเป็นการควบคุมโดยตัวควบคุม แทน

5. EOI (End or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม (Controller) หรืออุปกรณ์ที่เป็นตัวส่ง (Talker) ก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้เสร็จสิ้นลงแล้ว



รูปที่ 2.2 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่าง ๆ

2.4.3 กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูล

1. DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก "Low" โดยอุปกรณ์ที่เป็นตัวส่ง (Talker) เป็นการแจ้งแก่ระบบบัสว่า ขณะนี้ตัวส่งได้ทำการส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. NRFD (Not Ready For Data) เมื่อสัญญาณนี้มีลอจิกเป็น "Low" จะเป็นการแสดงว่าใน ขณะนี้ระบบบัสยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบยังพร้อมไม่หมดทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่เป็น "Hi" จนกว่าอุปกรณ์ทุกตัวให้ลอจิกที่เป็น "Hi" ครบถ้วนแล้ว สัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ในระบบมีความเร็วแตกต่างกัน

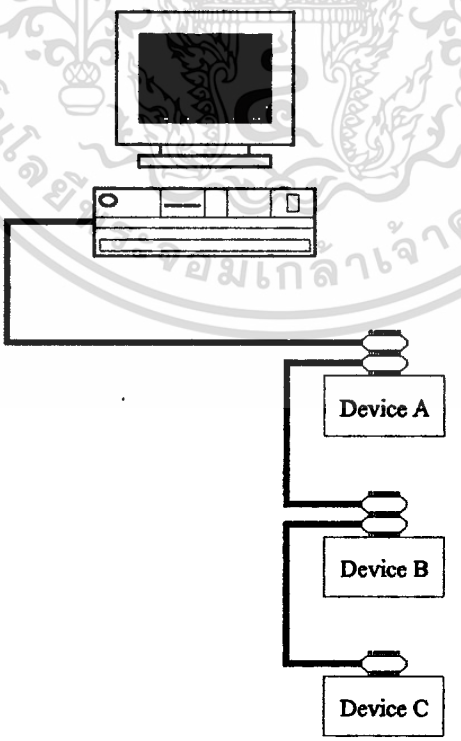
3. NDAC (Not Data Accepted) สัญญาณเส้นนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็น ตัวรับ (Listener) โดยสัญญาณนี้จะมีลอจิกเป็น "Low" ในขณะที่อุปกรณ์ที่เป็นตัวรับกำลังเก็บข้อมูล จากสายข้อมูล (Data Bus) และจะเป็น "Hi" เมื่ออุปกรณ์นั้นได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว

โดยสัญญาณลอจิกที่ใช้ใน DATA BUS (D1-D8) ของ IEEE-488 มีลักษณะเป็นคอมพลีเมนต์ทั้งหมด คือ "1" เท่ากับ "Low" และ "0" เท่ากับ "Hi" ซึ่งตรงข้ามกับในวงจรที่เราคุ้นเคย

2.5 การเชื่อมต่ออุปกรณ์ต่าง ๆ ในระบบ IEEE-488 BUS

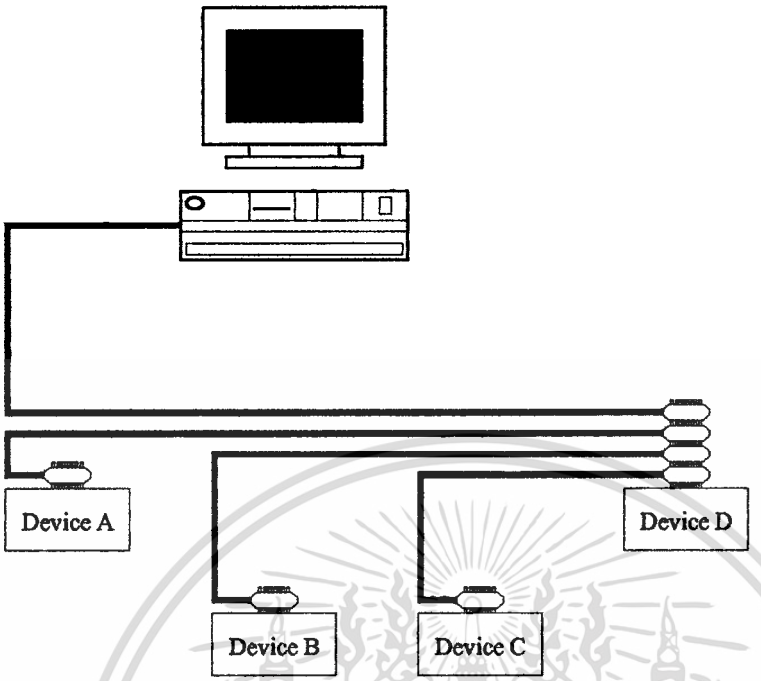
สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ในระบบ IEEE-488 Bus นั้น มีอยู่ 2 วิธี คือ

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



รูปที่ 2.3 การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)

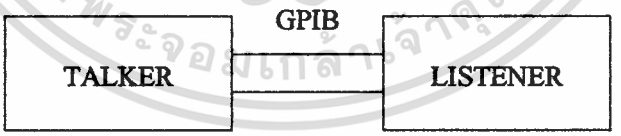
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเชื่อมต่อแบบกระจาย (Star Configuration)

2.6 ขบวนการแฮนด์เชค (Handshake Procedure)

เมื่อมีการเชื่อต่อระหว่างอุปกรณ์ต่าง ๆ ในระบบ ดังนั้น เมื่อระบบจะทำงานจึงต้องมีการตรวจสอบเสียก่อน ซึ่งการตรวจสอบนี้เราเรียกว่าขบวนการแฮนด์เชค (Handshake Procedure)



รูปที่ 2.6 ขบวนการแฮนด์เชค (Handshake Procedure)

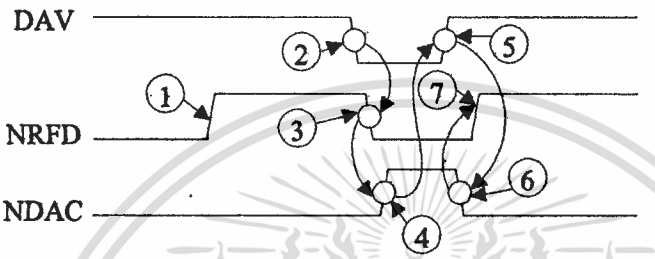
พิจารณาระบบที่ไม่ซับซ้อนนัก โดยกำหนดให้มีตัวส่ง (Talker) และตัวรับ (Listener) อย่างละ 1 ตัว

การที่จะมีการสื่อสารระหว่างตัวส่งและตัวรับจะกระทำได้โดยการที่ตัวส่ง จะทำหน้าที่ส่งสัญญาณออกมาเพื่อให้ตัวรับทราบว่ามีข้อมูลเสร็จในสายสัญญาณ และตัวรับก็จะส่งสัญญาณเพื่อให้ ตัวส่งทราบว่าได้ทำการรับข้อมูลเป็นที่เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งสัญญาณในการสื่อสารระหว่างตัวส่งและตัวรับนั้น จะส่งมาทางสายสัญญาณ 3 สาย ซึ่งสายสัญญาณดังกล่าวนั้นเป็นสายสัญญาณในกลุ่มควบคุมการรับ-ส่งข้อมูล คือ NRFD (Not Ready For Data), DAV (Data Valid) และ NDAC (Not Data Accepted)

DAV เป็นสัญญาณที่ถูกส่งโดยตัวส่ง NRFD และ NDAC เป็นสัญญาณที่ถูกส่งโดยตัวรับ โดยตัวรับจะใช้สัญญาณ NDAC เพื่อชี้ให้เห็นว่าพร้อม หรือไม่พร้อมที่จะรับข้อมูล



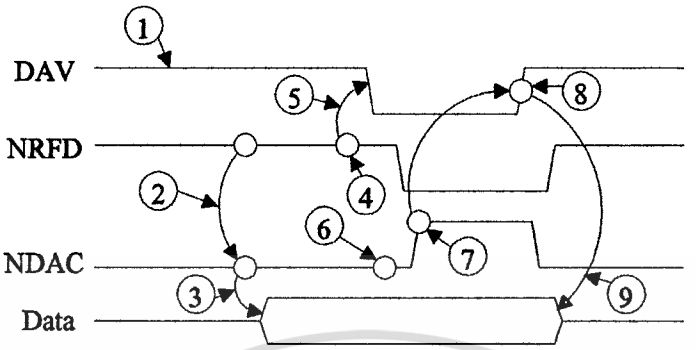
NOTE: The timing shown is relative

รูปที่ 2.7 timing diagram ของขบวนการแฮนด์เชค

ขบวนการแฮนด์เชคเริ่มขึ้นเมื่อ ตัวส่งมีข้อมูลที่จะถ่ายทอดลงสายสัญญาณไปยังตัวรับเริ่มด้วย การที่ตัวรับจะทำให้สัญญาณ NRFD เป็น "Hi" (มีค่าเป็น 0) นั่นคือ เป็นการบอกให้ทราบว่าตัวรับ พร้อมที่จะให้ตัวส่งทำการถ่ายข้อมูลลงในสายสัญญาณแล้ว (ในจุดที่ 1) ตัวส่งจะทำการถ่ายข้อมูล ลงในสายสัญญาณ DI01-DI08 หลังจากทำการรอเวลา เพื่อให้ตัวส่งถ่ายข้อมูลแล้วตัวส่งจะทำให้ สัญญาณ DAV มีลอจิกเป็น "Low" (มีค่าเป็น 1) เพื่อที่จะบอกให้ทราบว่าขณะนี้ได้มีข้อมูลในสาย สัญญาณทั้ง 8 เส้นแล้ว (ในจุดที่ 2) ต่อมาเมื่อผู้รับทราบว่ามีข้อมูลอยู่ในบัสก็จะทำให้สัญญาณ NRFD ให้มีค่าเป็น "Low" เพื่อบอกให้ตัวส่งทราบว่ายังไม่พร้อมที่จะให้ตัวส่งทำการส่งข้อมูลชุด ต่อไป และตัวรับพร้อมที่จะเก็บข้อมูลจากสายสัญญาณ (ในจุดที่ 3) หลังจากตัวรับทำการเก็บข้อมูลจาก สายสัญญาณเสร็จแล้ว ตัวรับจะทำให้สัญญาณ NDAC ให้มีสถานะเป็น "Hi" เพื่อบอกให้ทราบว่าได้ ทำการเก็บข้อมูลเสร็จแล้ว (ในจุดที่ 4) เมื่อตัวส่งตรวจพบว่าสัญญาณ NDAC มีลอจิกเป็น "Hi" ก็จะ ทำให้สัญญาณ DAV ให้มีลอจิกเป็น "Hi" ด้วย เพื่อไม่ให้ตัวรับทำการรับข้อมูลในบัสอีก (ในจุดที่ 5) เมื่อตัวรับทราบว่าสัญญาณ DAV มีลอจิกเป็น "Hi" ก็จะทำให้สัญญาณ NDAC ให้เป็น "Low" ทำให้ ข้อมูลในบัสถูกกำจัดออกไป หลังจากนั้นก็จะทำให้สัญญาณ NRFD ให้เป็น "Hi" เพื่อบอกให้ทราบ ว่าพร้อมที่จะรับข้อมูลชุดต่อไปแล้วขบวนการแฮนด์เชคจึงเสร็จสิ้นลง และจะเริ่มขึ้นใหม่เมื่อมีสัญญาณ ชุดใหม่เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวส่ง



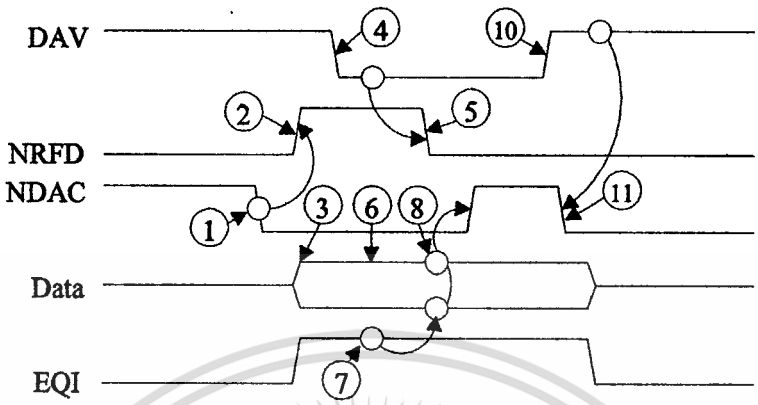
รูปที่ 2.8 timing diagram ของขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวส่ง

ขบวนการแฮนด์เชคเริ่มขึ้นโดยการที่ตัวควบคุม ทำการเซตค่าให้สัญญาณ DAV มีค่าเป็น "Hi" (ในจุดที่ 1) ต่อมาเป็นการตรวจสอบว่าสัญญาณ NDAC และ NRFD มีลอจิกเป็น "Hi" ทั้งคู่หรือไม่ (ในจุดที่ 2) ถ้าทั้งคู่เป็น "Hi" นั่นคือเกิดการผิดปกติ คือ อุปกรณ์นั้นไม่พร้อมที่จะทำงาน จึงทำการออกจากขบวนการแฮนด์เชค

กลับมาดูในจุดที่ 2 หากสัญญาณใดสัญญาณหนึ่งมีลอจิกเป็น "Low" ตัวควบคุมจะทำการส่ง ข้อมูลลงใน GPIB Bus (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบว่าสัญญาณ NRFD มี ลอจิกเป็น "Hi" (ในจุดที่ 4) หลังจากนั้นตัวควบคุมจะทำให้สัญญาณ DAV ให้มีลอจิกเป็น "Low" เพื่อให้ตัวรับทราบว่ามีข้อมูลอยู่ในบัสข้อมูล (ในจุดที่ 5) ตัวควบคุมจะรอเวลาเพื่อให้ตัวรับทำการเก็บ ข้อมูล เมื่อครบกำหนดเวลาตัวควบคุมจะทำการเชคว่าตัวรับทำงานเกินเวลาที่กำหนด หรือไม่หากเกิน เวลาก็จะทำการตรวจสอบต่อไปว่าสัญญาณ NDAC ทำลอจิกเป็น "Hi" ใช่หรือไม่ก็จะทำการ ตรวจสอบอีกครั้งว่าเกินเวลาหรือไม่ กลับไปดูที่จุดที่ 6 หากสัญญาณ NDAC เป็น "Hi" (ในจุดที่ 7) ตัวควบคุมจะทำการตอบสนองโดยการทำสัญญาณ DAV ให้มีลอจิกเป็น "Hi" เพื่อบอกให้ตัวรับ ทราบว่าให้หยุดการเก็บข้อมูลจากบัส (ในจุดที่ 8) หลังจากนั้นข้อมูลในบัสจะถูกนำออกไป (จุดที่ 9) จึงเป็นการเสร็จสิ้นขบวนการแฮนด์เชค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวรับ



รูปที่ 2.9 timing diagram ของขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวรับ

ขบวนการแฮนด์เชคเริ่มขึ้น หลังจากการที่ตัวควบคุมรับรู้ว่าตัวส่งจะทำการส่งข้อมูลเมื่อตัวควบคุมทราบว่าตัวส่งจะทำการส่งข้อมูล ตัวควบคุมรับรู้ว่าตัวส่งจะทำให้สัญญาณ NDAC มีลอจิกเป็น "Low" เพื่อบอกให้ทราบว่าตัวควบคุมยังไม่ได้ทำการเก็บข้อมูล (ข้อมูลในที่นี้คือ แอดเดรส หรือข้อมูลทั่วไปซึ่งตัวส่งทำการส่งมาในบัสข้อมูล) นั่นคือในจุดที่ 1

ต่อมาตัวควบคุมจะทำการเซตค่าสัญญาณ NRFD ให้เป็น "Hi" เพื่อที่จะให้ตัวส่งทราบว่าขณะนี้ตัวควบคุมพร้อมที่จะรับข้อมูลแล้ว (ในจุดที่ 2) เมื่อสัญญาณ NRFD มีค่าเป็น "Hi" และสัญญาณ NDAC มีค่าเป็น "Low" แล้ว ตัวส่งจะทราบทันทีว่าทำการส่งข้อมูลลงมาในบัสข้อมูล ได้แล้ว (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะทำการรอเวลาให้ตัวส่งทำการส่งข้อมูลลงในบัสข้อมูล ให้เสร็จ โดยจะทำการตรวจสอบไปด้วยว่าเกินเวลาที่กำหนดหรือยัง หากเกินเวลาที่กำหนดก็จะออกจากขบวนการแฮนด์เชค หากไม่เกินกำหนดเวลาจะทำการตรวจสอบเวลาว่าเกินเวลาที่ต้องคอยแล้ว หรือยัง หากยังไม่เกินเวลาก็จะทำการเชคต่อไปจนกระทั่งหมดเวลาหรือจนกว่าสัญญาณ DAV เป็น "Low" หรือไม่หากไม่ก็จะตรวจสอบเวลาว่าเกินเวลาที่ต้องรอแล้วหรือยัง หากยังไม่เกินเวลาก็จะทำ การเช็คต่อไปจนกระทั่งหมดเวลาหรือจนกว่าสัญญาณ DAV จะมีค่าเป็น "Low"

เมื่อสัญญาณ DAV เป็น "Low" แล้วตัวควบคุมจะตอบรับ โดยการทำให้สัญญาณ NRFD มีค่าเป็น "Low" (ในจุดที่ 5) นั่นเป็นการบอกว่าตัวควบคุมพร้อมที่จะเริ่มเก็บข้อมูล (ในจุดที่ 6) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ EOI เพื่อตรวจสอบว่าข้อมูลที่ตัวส่งทำการส่งลง มานั้นหมดแล้วหรือยัง (ถ้าตัวส่งทำการส่งข้อมูลลงในบัสหมดแล้วจะทำการตั้งค่าสัญญาณ EOI ให้เป็น "Low") (ในจุดที่ 7) หากสัญญาณ EOI เป็น "Low" ตัวควบคุมจะเซตสถานะว่า ขณะนี้ตัวส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการส่งข้อมูลลงมาในบัสหมดแล้ว หากสัญญาณ EOI เป็น "Hi" ก็ไม่ทำการเซตสถานะ หลังจากนั้นตัวควบคุมจะทำการเป็นข้อมูลในบัส (ในจุดที่ 8) แล้วทำให้สัญญาณ NDAC มีค่าเป็น "Hi" เพื่อบอกให้ทราบว่าตัวควบคุมได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว (ในจุดที่ 9) เมื่อตัวควบคุมทำให้สัญญาณ NDAC เป็น "Hi" แล้วตัวส่งจะลบข้อมูลในบัสข้อมูล โดยการทำให้สัญญาณ DAV เป็น "Hi" (ในจุดที่ 10) ซึ่งก่อนหน้านี้ตัวควบคุมจะทำการตรวจสอบอยู่แล้วว่าสัญญาณ DAV เป็น "Hi" หรือยัง

เมื่อสัญญาณ DAV เป็น "Hi" แล้วตัวควบคุมจะทำให้สัญญาณ NDAC มีค่าเป็น "Low" (ในจุดที่ 11) แล้วจึงออกจากขบวนการแฮนด์เชค

2.9 คำสั่งใช้งานของ GPIB

การสั่งการต่าง ๆ เพื่อกำหนดหน้าที่การทำงานและกำหนดฟังก์ชัน เช่น กำหนดช่วงการวัด โหมดการวัด หรืออื่น ๆ แก่เครื่องวัดที่ตกอยู่เหล่านี้นั้น ตัวควบคุมจะเป็นตัวกำหนดการส่งรหัสคำสั่งไปที่อุปกรณ์โดยผ่าน DI1-DI6 รหัสคำสั่งนี้จะถูกส่งไปในช่วงที่สายสัญญาณ ATN เป็น LOW

คำสั่งสำหรับกำหนดหน้าที่การงานต่าง ๆ ตามมาตรฐานของ GPIB มีอยู่ด้วย 128 คำสั่ง ดังแสดงในตารางที่ 2.1 ต่อไปนี้ โดยแบ่งเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIB บัสนั้นใช้ร่วมกัน ทั้งรหัสข้อมูลและรหัสคำสั่ง นั่นคือรหัสเดียวกัน มีความหมายได้ 2 อย่าง คือ เมื่อ ATN เป็น LOW จะหมายถึงรหัสคำสั่ง แต่ ATN เป็น HIGH รหัสนี้จะแทนข้อมูลที่เป็น ASCII แทน ซึ่งในตารางก็ได้แบ่งความหมายออกเป็น 2 คอลัมน์

ตารางที่ 2.1 แสดงคำสั่งการใช้งานของ GPIB

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

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

MEANING DEFINED BY PCG CODE

ADDRESSED COMMAND GROUP | UNIVERSAL COMMAND GROUP | LISTEN ADDRESS GROUP | TALK ADDRESS GROUP | SECONDARY COMMAND GROUP

PRIMARY COMMAND GROUP (PCG)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กลุ่มคำสั่งเจาะจงจุดหมาย (addressed command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ที่ เป็นตัวส่งหรือตัวรับที่กำหนดไว้ล่วงหน้าแล้ว คำสั่งนี้ประกอบด้วย

GTL (got to local) สั่งให้อุปกรณ์กลับสู่สภาพการควบคุมปกติด้วยมือ

SDC (selected device clear) สั่งให้อุปกรณ์เคลียร์ตัวเองสู่สภาพเริ่มต้นใหม่

PPC (paralled poll configure) เป็นคำสั่งสำหรับทำการจัดสายสัญญาณของการทำการจัดสรรสายสัญญาณของการทำกระบวนการตรวจสอบสภาพอุปกรณ์โดยวิธีขนาน โดยใช้กับกลุ่มคำสั่ง รอง

GET (group excute trigger) ใช้สั่งเริ่มต้นการทำงานของอุปกรณ์ที่ละหลายตัว

TCT (take control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

2. กลุ่มคำสั่งครอบคลุม (universal command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่ ต่ออยู่ในบัส ประกอบด้วย

LLO (local lockout) เป็นการสั่งให้อุปกรณ์ล็อคอยู่ที่สภาวะควบคุมโดยปุ่มปรับที่หน้าปัดตามปกติ

DCL (device clear) สั่งให้อุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น

PPU (parallel poll unconfigure) ใช้ยกเลิกกระบวนการตรวจสอบสภาพแบบขนาดทั้งหมด

SPE (serial poll enable) เปลี่ยนโหมดของการตรวจสอบสภาพเป็นแบบอนุกรมในโหมดนี้จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล

SPD (serial poll disable) ยกเลิกโหมดการตรวจสอบแบบอนุกรม

3. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ (listener address group) เป็นคำสั่งสำหรับกำหนดให้อุปกรณ์เป็นตัวรับ ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNT (untalker) สำหรับยกเลิก

4. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวส่ง (talker address group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่ง ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNL (unlisten) สำหรับยกเลิกเช่นกัน

คำสั่งกลุ่มที่ 1 ถึง 4 นั้น จัดเป็นกลุ่มคำสั่งหลักที่มีความหมายตายตัวยังมีคำสั่งอีกกลุ่มที่ขึ้นอยู่กับการกำหนดภายหลังนั่นคือ กลุ่มคำสั่งรอง

5. กลุ่มคำสั่งรอง (secondary command group) เป็นคำสั่งที่กำหนดรายละเอียดย่อยของอุปกรณ์แต่ละตัวที่ต่ออยู่ในระบบ ให้มีการทำงานอย่างไร ตามจุดประสงค์ใช้งานของเครื่องมือนั้น เช่นเดียวกับการปรับปุ่มต่าง ๆ ด้วยมือนั่นเอง คำสั่งรองนี้จะตามหลังคำสั่งหลัก คือจะใช้หลังจากอุปกรณ์ต่าง ๆ ถูกกำหนดวางตัวในระบบเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งต่าง ๆ ที่กล่าวไป ซึ่งใช้ในการกำหนดสภาวะการทำงานของอุปกรณ์ แต่ละสภาวะ ที่กำหนดไปนั้นเป็นอย่างไร และมีจุดประสงค์เพื่ออะไร ดังต่อไปนี้

Device clear / Interface Clear

Device clear ใช้ในการทำให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปสู่สภาวะเริ่มต้น ยังไม่มีการกำหนดฟังก์ชันใด ๆ สภาวะเริ่มต้นนี้จะแตกต่างกันไป แล้วแต่ว่าอุปกรณ์นั้นออกแบบไว้อย่างไร Device clear มีอยู่ 2 ลักษณะ คือ เคลียร์หมดทุกตัวที่ต่ออยู่ (DCL) กับเคลียร์เจาะจงอุปกรณ์ตัวใด ตัวหนึ่ง (SDC)

แต่ว่าในการเคลียร์อุปกรณ์ให้อยู่ในสภาวะเริ่มต้นนั้นไม่ได้หมายความว่า Interface function ของ GPIB จะถูกเคลียร์อุปกรณ์ให้ไปอยู่ในสภาวะเริ่มต้นด้วยแต่อย่างใด interface function คือ สภาพการ interface ที่ได้กำหนดไว้ในระบบประกอบด้วยฟังก์ชันต่าง ๆ ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 แสดง interface function

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote / Local

remote เป็นการกำหนดให้อุปกรณ์ที่อยู่ในระบบเช่น เครื่องมือวัดให้อยู่ในการควบคุมของอุปกรณ์ตัวอื่นแทน ซึ่งปุ่มปรับต่าง ๆ บนหน้าปัดเครื่องจะไม่มีผลต่อการทำงาน ส่วน Local เป็นการ ควบคุมการทำงานของเครื่องมือวัดด้วยปุ่มปรับบนหน้าปัดตามปกติ

การใช้ remote มีประโยชน์ในแง่ที่ขณะที่ตัวการควบคุมเช่น คอมพิวเตอร์กำลังติดต่ออุปกรณ์ ตัวนั้นอยู่ หากไม่มีการตัดการควบคุมโดยปุ่มปรับบนหน้าปัดออก ถ้ามีใครมาปรับแต่งก็จะทำให้การ ทำงานผิดพลาดไปได้ การทำงานของ GPIB ใน remote and local มี 4 ลักษณะดังนี้

1. LOCS ก็คือ local นั่นเอง เป็นสภาพการควบคุมที่ปุ่มตามปกติ จะอยู่ในสภาพนี้เปิดตอนเครื่องหรือ REN เป็น HIGH หรือเมื่อได้รับคำสั่ง GTL

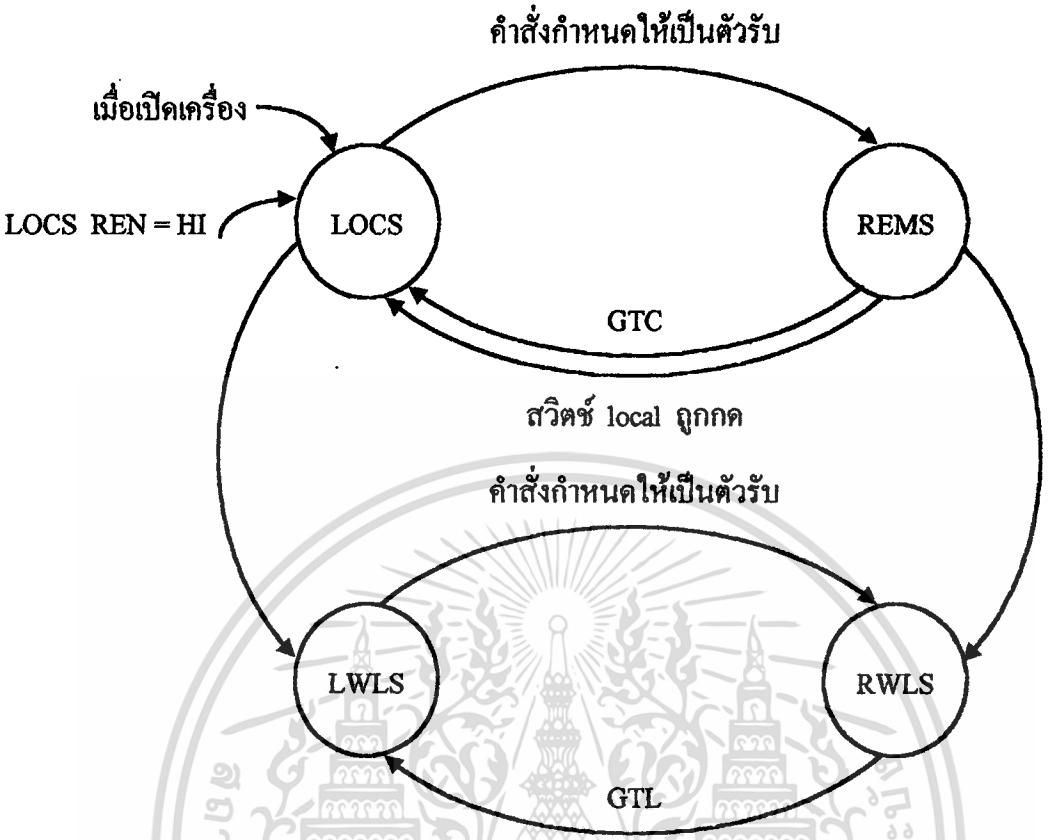
2. REMS คือ remote หมายถึงการตัดการควบคุมโดยปุ่มหน้าปัดออก จะเกิดขึ้นเมื่อ REN

เป็น LOW และจะถูกล็อกไว้ เว้นแต่ว่าสวิตช์ local ที่ตัวอุปกรณ์จะถูกเปลี่ยนไปตำแหน่ง Local

3. RWLS เป็นสภาพ remote ที่ถูกล็อกเอาไว้เช่นกัน แต่ว่าจะตัดการควบคุมตรงสวิตช์ local ที่ตัวอุปกรณ์ออกไป สภาพ remote โดย RWLS จึงมีความสำคัญสูงกว่า REMS อย่างไรก็ตามยังถูกยกเลิกได้ด้วยคำสั่ง LLO

4. LWLS มีสภาพเช่นเดียวกับ local แต่จะแตกต่างกันตรงที่สภาพ local โดย LWLS นี้เมื่อได้รับคำสั่งกำหนดอุปกรณ์ตัวรับจะเปลี่ยนไปอยู่ในสภาพแบบล็อกหรือ RWLS ทันทีในการที่จะมาที่สภาพ LWLS นี้ได้ก็มี 2 กรณีคือ เมื่ออยู่ในสภาพ local ธรรมดา (LOCS) แล้วได้รับคำสั่ง LLO หรืออยู่ใน RWLS แล้วได้รับคำสั่ง GTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดง loop การทำงานของ GPIB ใน remote/local

2.10 การขอบริการและการตรวจสอบ (Service Request and Polling)

เมื่อตัวควบคุมได้รับ SRQ เป็น LOW จะให้อุปกรณ์ส่งข้อมูลแสดงสถานะการทำงาน ซึ่งมีอยู่ 2 วิธีคือ

1. การตรวจสอบแบบอนุกรม ซึ่งมีขั้นตอนดังนี้

1.1 ATN ถูกดึงเป็น LOW หลังจากได้รับ LOW จากสายสัญญาณ SRQ

1.2 คำสั่ง UNL ถูกสั่งไปยังอุปกรณ์

1.3 ตัวควบคุมจะแจ้งรหัสตัวรับของตน และกำหนดรหัสตัวส่งอุปกรณ์ที่จะตรวจสอบไป ที่บัส

1.4 ตามด้วยคำสั่ง SPE และสาย ATN กลายเป็น HI ซึ่งอุปกรณ์ที่ถูกเรียกจะส่งข้อมูลแสดงสถานะออกมา 1 ไบต์ โดยบิตที่ 7 จะเป็นตัวชี้ว่าอุปกรณ์เส้นเป็นตัวขอบริการ ถ้าใช่จะเป็น LOW ส่วนบิตอื่น ๆ ก็ใช้บอกข้อมูลอื่น ๆ ซึ่งมิได้กำหนดเฉพาะ

1.5 สาย ATN ถูกดึงเป็น LOW อีกที เพื่อส่งคำสั่งยกเลิกการตรวจสอบคือ SPD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

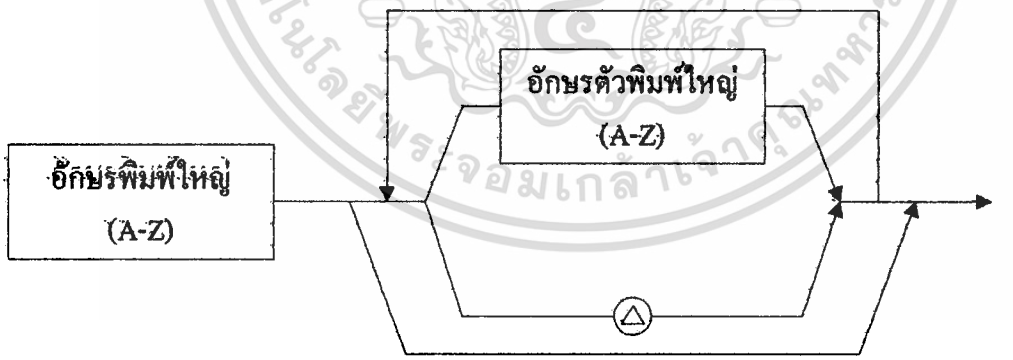
1.6 จากนั้นคำสั่ง UNT ก็ถูกส่งไปยังอุปกรณ์เพื่อยกเลิกการเป็นตัวส่ง ซึ่งถ้าหาก SQR ยังคงเป็น LOW อยู่ ก็จะมีการตรวจสอบไปยังอุปกรณ์ตัวอื่น ๆ ต่อไปตามขั้นตอนเดิม

2. การตรวจสอบแบบขนาน สามารถทำได้เร็วกว่าแบบอนุกรมทั้งนี้เพราะสามารถอ่านข้อมูลเพียงไบต์เดียวก็สามารถรู้ได้ทันที ว่าอุปกรณ์ตัวใดเป็นผู้ขอบริการ

2.11 รูปแบบของข้อมูล

โดยทั่วไปข้อมูลจากอุปกรณ์ (device message) แบ่งได้ออกเป็น 3 ส่วนดังแสดงในรูปที่ 2.11 อันได้แก่ส่วนหัว (HR) ซึ่งจะอยู่ส่วนหน้าสุด เป็นตัวบอกชนิดข้อมูล ส่วนประกอบ HR แสดงในรูป ที่ 10 จะเห็นว่าประกอบด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ช่องว่างที่เป็นเว้นวรรค ( ) ปกติจะมี อักษรประมาณ 1-3 ตัว

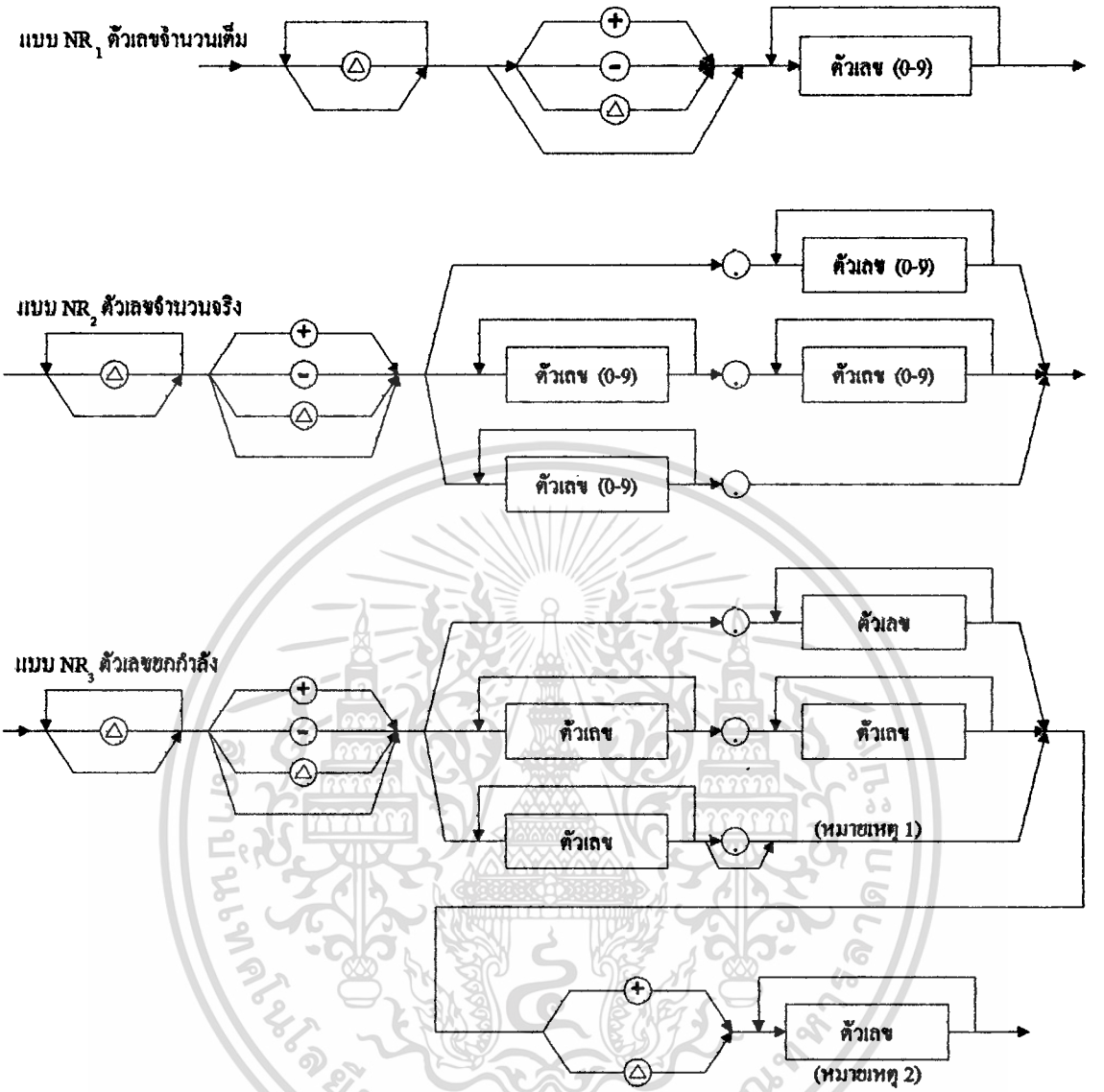
รูปที่ 2.11 รูปแบบของข้อมูล



รูปที่ 2.12 รูปแบบของ HR

ส่วนที่ 2 คือ เนื้อหาข้อมูล (NR) ซึ่งใช้แสดงค่าตัวเลข มีอยู่ 3 แบบ คือ NR1, NR2 และ NR3 ดังแสดงในรูปที่ 2.13 ส่วนท้ายของ NR ยังอาจมีตัวอักษรแสดงหน่วยตามมา

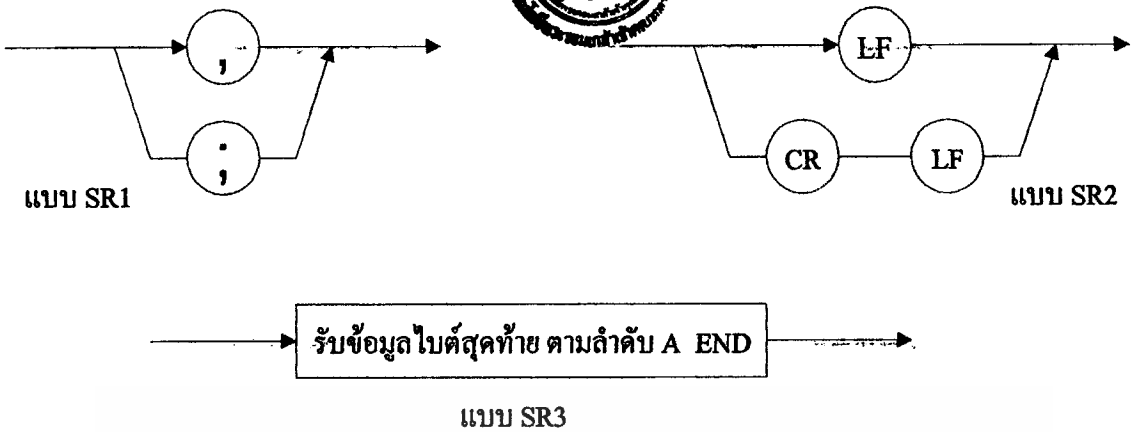
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงเนื้อหาข้อมูล

ส่วนที่ 3 คือ สัญญาณแบ่งข้อมูลแต่ละชุด (SR) ดังแสดงในรูปที่ 2.14 โดน SR1 ใช้แสดงการ ต่อเนื่องของข้อมูล (ข้อมูลยังมีต่อ) SR2 และ SR3 แสดงการสิ้นสุดของข้อมูล แต่ SR3 เป็นการ บอกการเสร็จสิ้นข้อมูลทั้งหมดจากการวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงสัญญาณแบ่งข้อมูลแต่ละชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

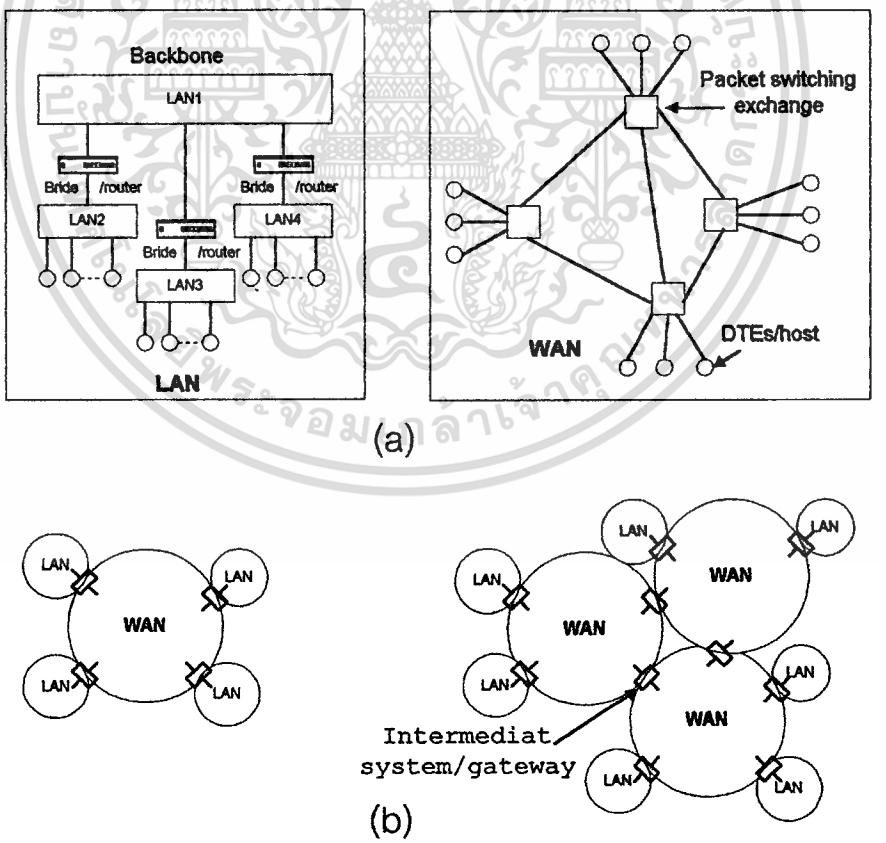
039159

บทที่ 3
เครือข่ายอินเตอร์เน็ต

Internet คือ การที่เครือข่าย 2 หรือมากกว่า เชื่อมต่อเข้าด้วยกัน และทำงานเสมือนเป็นเครือข่ายเดียวกัน โดย network ที่เป็นส่วนประกอบของ internet คือ Subnetwork (Subnet) ซึ่งอาจจะเป็นเครือข่าย Local area network (LAN) หรือ Wide area network (WAN) อุปกรณ์ที่ใช้ในการเชื่อมต่อ 2 เครือข่าย เข้าด้วยกันก็คือ intermediate system (IS) หรือ internetworking unit (IWU) การเชื่อมโยงระหว่างระบบที่แตกต่างกัน จำเป็นต้องมีมาตรฐานในการติดต่อกัน ซึ่งเรียกเป็นศัพท์เฉพาะว่าโพรโตคอล (Protocol)

3.1 สถาปัตยกรรม อินเตอร์เน็ต (Internet Architectures)

สถาปัตยกรรมพื้นฐานของ Internet แสดงดังรูปที่ 3.1



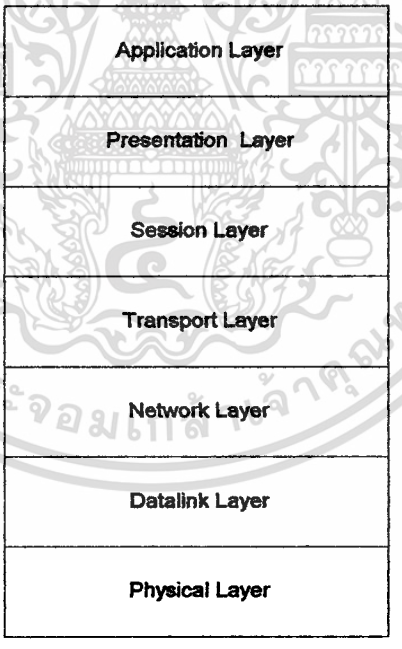
รูปที่ 3.1 แสดงสถาปัตยกรรมของ Internet (a) Single LAN and WAN (b) Interconnected LAN/WAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูป (a) แสดงตัวอย่าง 2 ตัวอย่างของเครือข่ายเดี่ยว (Single network) ซึ่งอย่างแรกเป็น site-wide LAN ซึ่งประกอบขึ้นมาจากชุดของ LANs ซึ่งถูกต่อเข้ากับเครือข่ายหลัก (backbone) ซึ่งอุปกรณ์ที่ใช้ต่อ LAN เข้ากับเครือข่ายหลัก ถ้า LAN ทุกเครือข่ายมีระบบเดียวกันก็จะใช้ bridge ถ้าเป็น LAN ที่แตกต่างกันก็จะใช้ router ตัวอย่างที่ 2 เป็นตัวอย่างของ WAN เดี่ยว ๆ ในรูป (b) แสดงถึงเครือข่ายอินเตอร์เน็ต ซึ่งประกอบด้วย network ทั้ง 2 ชนิด ข้างต้น

3.2 OSI โมเดล

องค์กรมาตรฐานสากล ISO (International Organization for Standardization) ได้กำหนดมาตรฐานของเครือข่าย โดยจัดแบ่งกิจกรรมของเครือข่าย ออกเป็นงานย่อย ๆ และกำหนดโมเดลแบ่งเป็นชั้น ๆ ตามลำดับเรียกว่ามาตรฐาน OSI (Open System Interconnection) โดยที่จะแบ่งกิจกรรมที่ซับซ้อนในเครือข่าย ออกเป็นงานย่อย ๆ ก็จะช่วยในการออกแบบ และการใช้งานเครือข่ายรวมถึงการเชื่อมโยงกัน เป็นไปได้ด้วยความสะดวก และมีวิธีการทำงานอยู่ในกรอบเดียวกัน ดังในรูปที่ 3.2



รูปที่ 3.2 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model

ในแต่ละชั้นของ OSI model จะมีการติดต่อสื่อสารกันเป็นชั้น ๆ ตามลำดับลงมาเช่น Application Layer ก็จะติดต่อสื่อสารกับ Presentation Layer ตามลำดับไปจนถึงชั้นแรกสุดคือ Physical Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Layer เป็นชั้นบนสุดของ โมเดลเป็นส่วนที่จะทำให้การติดต่อระหว่างเครือข่ายกับผู้ใช้ เป็นไปได้ตามต้องการ ตัวอย่างแอปพลิเคชั่นของเครือข่าย เช่น ระบบ e-mail, การโอนถ่ายข้อมูล (File Transfer), การขอเข้าใช้ระบบคอมพิวเตอร์ในเครือข่ายเป็นต้น

Presentation Layer มีการกำหนดหน้าที่ไม่ชัดเจนนักและมีการนำไปใช้ไม่มาก ซึ่งหน้าที่หลักก็คือ เป็นส่วนที่จัดรูปแบบและนำเสนอข้อมูล ให้เป็นไปตามต้องการ รวมไปถึงการจัดแปลงข้อมูล ในรูปมาตรฐาน ASCII หรือ EBCDIC, การลดขนาดข้อมูล (data compression) การเข้ารหัสหรือถอดรหัสของข้อมูล แต่ส่วนใหญ่แล้ว แอปพลิเคชั่นจะจัดการแทนได้

Session Layer เป็นชั้นที่จัดการในเรื่อง "การติดต่อแต่ละครั้ง" หรือ session ให้ระบบคอมพิวเตอร์ทั้งสองฝั่ง โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อ ดูแลในการส่งผ่านข้อมูล ในการติดต่อครั้งนั้น ๆ เป็นไปได้โดยไม่มีปัญหาจนถึงเลิกการติดต่อเมื่อเสร็จงาน

Transport Layer ทำหน้าที่ควบคุมปริมาณ และรายละเอียดวิธีการรับส่งข้อมูล ให้เป็นไปตามกำหนดที่ตั้งไว้ และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปด้วยความราบรื่น Transport Layer จะเป็นชั้นสุดท้าย ที่จัดการเรื่องเส้นทางในการส่งข้อมูล และจัดการตรวจสอบความผิดพลาดของข้อมูล ซึ่งส่วนของ TCP (Transmission Control Protocol) ในโพรโตคอล TCP/IP ทำงานที่ระดับนี้

Network Layer ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลระหว่างเครือข่ายให้ถูกต้อง และเป็นไปตามเส้นทางที่กำหนด โดยจะจัดการส่งผ่าน packet ข้อมูล ผ่านอุปกรณ์ต่าง ๆ ไปยังเครือข่ายย่อยได้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ยังจัดการดูแลเส้นทางในการส่งข้อมูล (Routing table) และกั้นหรือกรอง packet ข้อมูลที่ส่งไปยังเครือข่ายเดียวกันไม่ให้ข้ามไปยังเครือข่ายอื่น ซึ่งจะช่วยลดปริมาณข้อมูลที่จะวิ่งบนเครือข่ายได้ส่วนหนึ่ง โพรโตคอล IP, TCP/IP และ IPx เป็นโพรโตคอลที่ทำงานอยู่ใน layer นี้

Data link Layer ทำหน้าที่เรียกใช้หรือกำหนดช่องทาง ในการส่งข้อมูลที่ถูกต้อง เช่น Ethernet, Tokenring หรือ FDDI เป็นต้น รวมถึงการลำดับและอัตราการรับส่งข้อมูลหรือ flow control และสถานที่ ที่จะส่งข้อมูลไป (address) ทั้งนี้ Data link layer จะเป็นชั้นแรกที่จัดการแปลงข้อมูลจาก bit ให้เป็น packet โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบความถูกต้อง ในกรณีที่ส่งข้อมูลออกไป หรือในกรณีที่อ่านข้อมูลที่เข้ามา ก็จะตรวจสอบผ่าน checksum เพื่อดูว่าข้อมูลที่ได้รับมาถูกต้องครบถ้วน และถ้าได้รับ packet ข้อมูลที่ไม่ถูกต้องก็จะไม่เอาข้อมูลนั้นไปใช้งาน และจะบอกให้ต้นทางส่งข้อมูลเดิมมาใหม่

Physical Layer รับผิดชอบดูแลในรายละเอียดในการส่งข้อมูลในด้าน hardware เช่น การควบคุม Network Interface Card การส่งสัญญาณผ่านสายสัญญาณแบบต่าง ๆ การเชื่อมต่อเข้ากับเครือข่ายแบบต่าง ๆ โดยใช้ Physical Layer จะจัดสร้างสัญญาณทางไฟฟ้า, สัญญาณเสียง หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่จำเป็นในการสื่อสารโดยตรงเนื่องจาก Network Layer เป็นชั้นที่โพรโตคอล IP, TCP/IP ทำงานอยู่จะกล่าวโดยละเอียดต่อไป

3.3 โครงสร้างชั้นเน็ตเวิร์ค (Network layer structure)

หน้าที่ของ Network Layer ในแต่ละ End System (ES) จะเป็นตัวจัดการการติดต่อแบบ end-to-end ของการบริการ internetwide ไปยังผู้ใช้บริการ (NS-User)

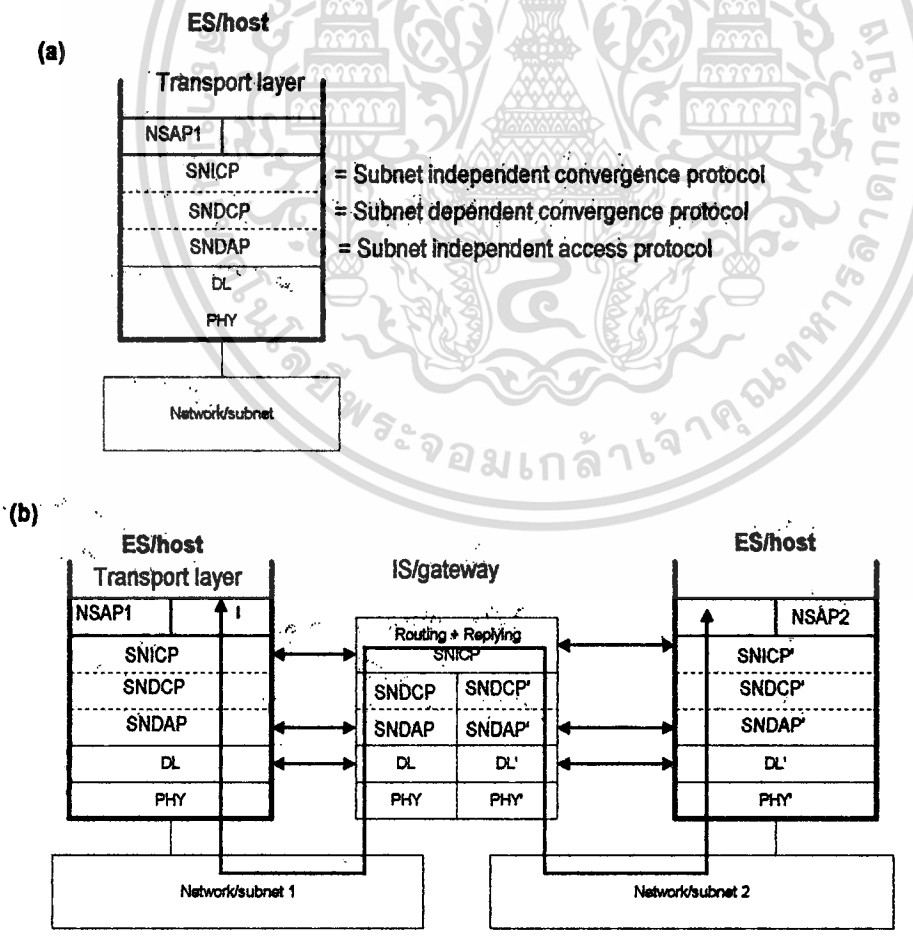
โดย ISO ได้จัด network layer เป็น 3 (sublayer) protocol ซึ่งจะทำงานร่วมกัน เพื่อให้บริการใน network layer ได้แก่

Subnetwork independent convergence protocol (SNICP)

Subnetwork dependent convergence protocol (SNDCP)

Subnetwork dependent access protocol (SNDAP)

โดยที่ โครงสร้างแสดงดังรูปที่ 3.3



รูปที่ 3.3 Network layer structure (a) Sublayer Protocol ; (b) IS structure

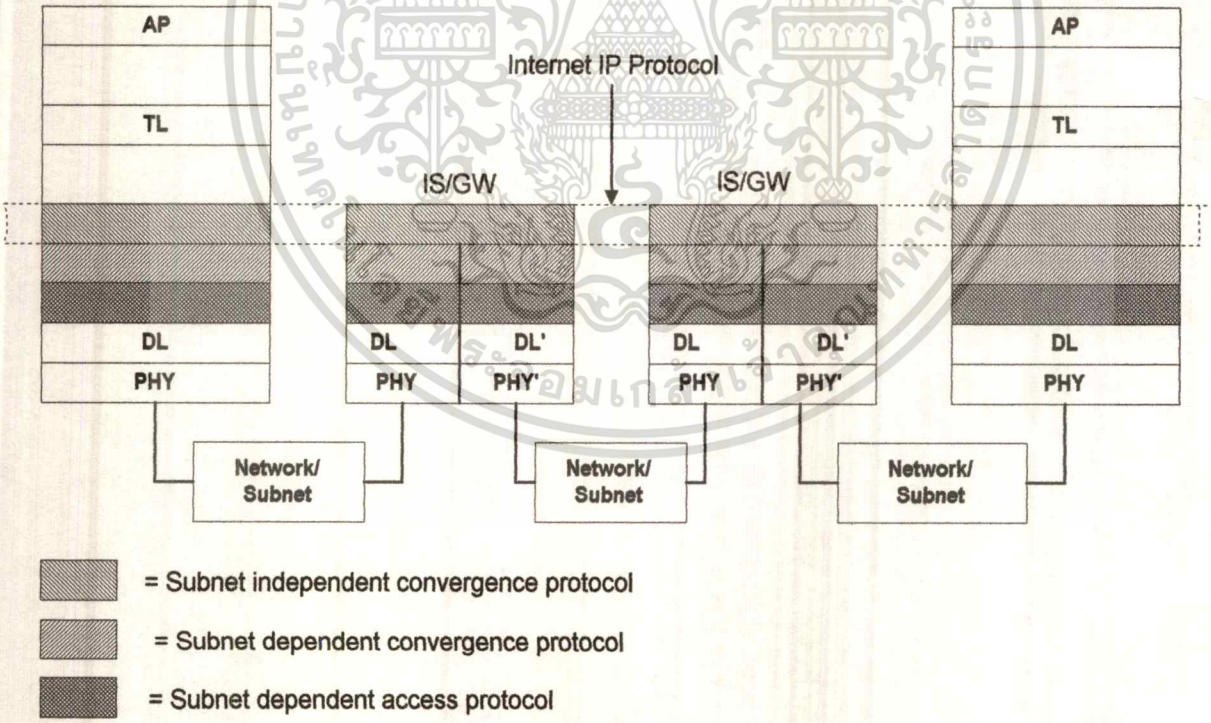
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ SNICP จะเป็นตัวสนับสนุนจัดการให้ผู้ใช้บริการ (NS-user) สามารถ interface กับ Internet ซึ่งมันจะมีหน้าที่ เป็นตัวประสานฟังก์ชันต่าง ๆ ที่จำเป็นในการเลือกเส้นทางและถ่ายทอดข้อมูลของผู้ใช้ข้าม Internet ซึ่งการทำงานของมันไม่ขึ้นอยู่กับคุณสมบัติเฉพาะของ เครือข่ายย่อย (subnet)

SNDAP จะเป็นตัวโพรโตคอลที่ติดต่อกับเครือข่ายย่อย (subnet) ที่มีลักษณะเฉพาะใน Internet เช่น X.25 packet layer protocol สำหรับเครือข่าย X.25 ซึ่งใช้บ่อยใน LAN เพราะว่าการบริการและการทำงานของ SNDAP แตกต่างจาก network แบบอื่น ๆ sublayer ที่อยู่ตรงกลางคือ SNDCP จะเป็นตัวจัดการระหว่าง SNICP และ SNDAP

3.4 Internet protocol standards

อินเตอร์เน็ตโพรโตคอลซึ่งถูกใช้ในอินเตอร์เน็ตมี โพรโตคอลก็คือ TCP/IP (Transfer Control Protocol / Internet Protocol) ซึ่งรวมถึง transport และ application โพรโตคอล ซึ่งทั้งหมดของ TCP/IP จะกำหนด ให้เหมาะกับการใช้ในเชิงสาธารณะ ซึ่งรูปแบบโดยทั่วไปแสดงดังรูปที่ 3.4



รูปที่ 3.4 Internetwide IP schematic

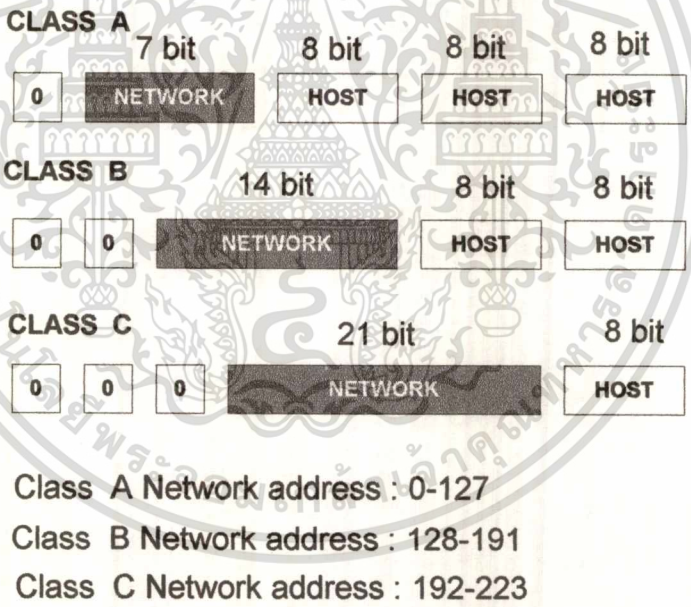
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP เป็น internetwide protocol ซึ่งทำให้สอง transport protocol ที่ต่างสถานที่และต่าง ESs / hosts กันสามารถแลกเปลี่ยนหน่วยข้อมูล (NSDUS) กันได้ ซึ่งหมายถึงว่า หลาย ๆ network / subnet และ ISs / gateways ที่แตกต่างกันสามารถ ติดต่อสื่อสารกันได้อย่างสมบูรณ์

3.5 Internet IP

3.5.1 โครงสร้างแอดเดรส (Address structure)

ในศัพท์ของ ISO เมื่อ 2 network ติดต่อกันด้วย host/ES ที่ต่อกับอินเตอร์เน็ต network เหล่านี้ ติดต่อกันได้โดยใช้ network service access point (NSAP) address และ subnet point of attachment (SNPA) สำหรับใน TCP/IP ก็จะมี IP address และ NPA address ตามลำดับ โดย NPA address จะแตกต่างกันในแต่ละชนิดของ network/subnet ขณะที่ IP address จะเป็นรูปแบบเดียวกัน โครงสร้างของ IP address แสดงดังรูปที่ 3.5



รูปที่ 3.5 โครงสร้างของ Address ที่ใช้ใน class ต่าง ๆ ของเครือข่ายทั้งหมดความยาว 32 bit

IP address นี้มีการจัดแบ่งออกเป็นทั้งหมด 5 ระดับ (class) แต่ที่ใช้งานทั่วไปจะมีเพียง 3 ระดับคือ Class A, Class B และ Class C ซึ่งจะแบ่งตามขนาดความใหญ่ของเครือข่าย ถ้าเครือข่ายใดมีจำนวนเครื่องคอมพิวเตอร์ที่เชื่อมต่ออยู่มาก จะมีหมายเลขอยู่ใน Class A และลดหลั่นกันมาใน Class B และ Class C ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

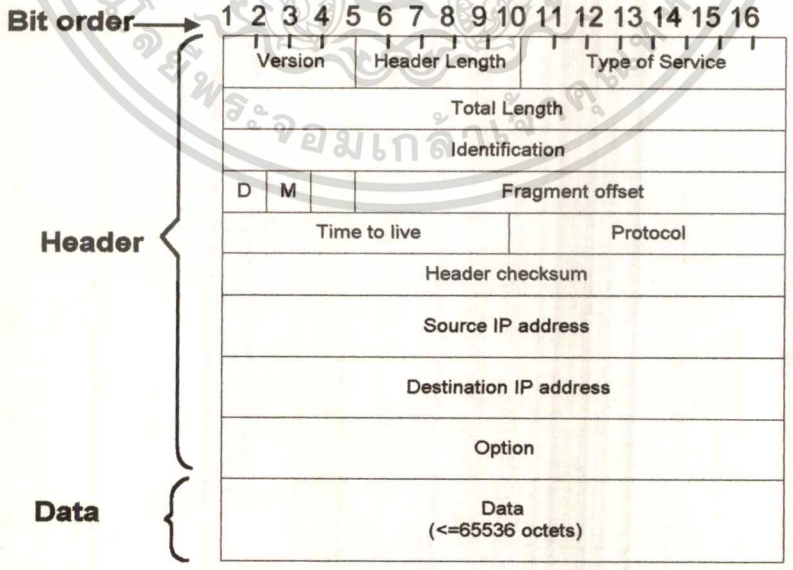
จากรูปจะเห็นว่าหมายเลข IP ของ Class A มีตัวแรกเป็น 0 และหมายเลขของเครือข่าย (network number) ขนาด 7 bit และมีหมายเลขเครื่องคอมพิวเตอร์ (Host number) ขนาด 24 bit ทำให้ในหนึ่งเครือข่ายของ Class A สามารถมีคอมพิวเตอร์เชื่อมต่ออยู่ในเครือข่ายได้ถึง 2^{24} =16 ล้านเครื่อง แต่ใน Class A นี้ จะมีหมายเลขเครือข่ายได้ 128 ตัวเท่านั้นทั่วโลก ซึ่งก็คือจะมีเครือข่ายใหญ่แบบนี้เพียง 128 เครือข่ายเท่านั้น

สำหรับ Class B จะมีหมายเลขเครือข่ายแบบ 14 bit และหมายเลขเครื่องคอมพิวเตอร์แบบ 16 bit (ส่วนอีก 2 bit ที่เหลือบังคับว่าต้องขึ้นต้นด้วย 10_2) ดังนั้นจึงสามารถมีคอมพิวเตอร์เชื่อมต่อในเครือข่าย Class B แต่ละเครือข่ายได้ถึง 2^{16} = กว่า 65,000 เครื่อง และสุดท้ายคือ Class C ซึ่งมีหมายเลขคอมพิวเตอร์แบบ 8 bit และมีหมายเลขเครือข่ายแบบ 21 bit ส่วน 3 bit แรกบังคับว่าต้องเป็น 110_2 ดังนั้นในแต่ละเครือข่าย Class C จะมีจำนวนเครื่องต่อเชื่อมได้เพียงไม่เกิน 254 เครื่องในแต่ละเครือข่าย (2^8=256 แต่หมายเลข 0 และ 255 จะไม่ถูกใช้งาน จึงเหลือเพียง 254)

จะเห็นได้ว่าเมื่อเครือข่ายและเครื่องคอมพิวเตอร์ ที่ต่ออยู่ในอินเตอร์เน็ตมีหมายเลข IP address ให้ใช้อ้างอิงได้ไม่ซ้ำกัน และมีความหมายให้ทราบถึงขนาดเครือข่ายแล้ว การติดต่อส่งผ่านข้อมูล จึงกระทำได้ไม่สับสน

3.5.2 รูปแบบของข้อมูล (Datagrams)

รูปแบบของ IP data unit ก็คือ datagrams ซึ่งโครงสร้างของ datagrams เป็นดังรูปที่ 3.6



รูปที่ 3.6 Internet datagrams format and contents

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

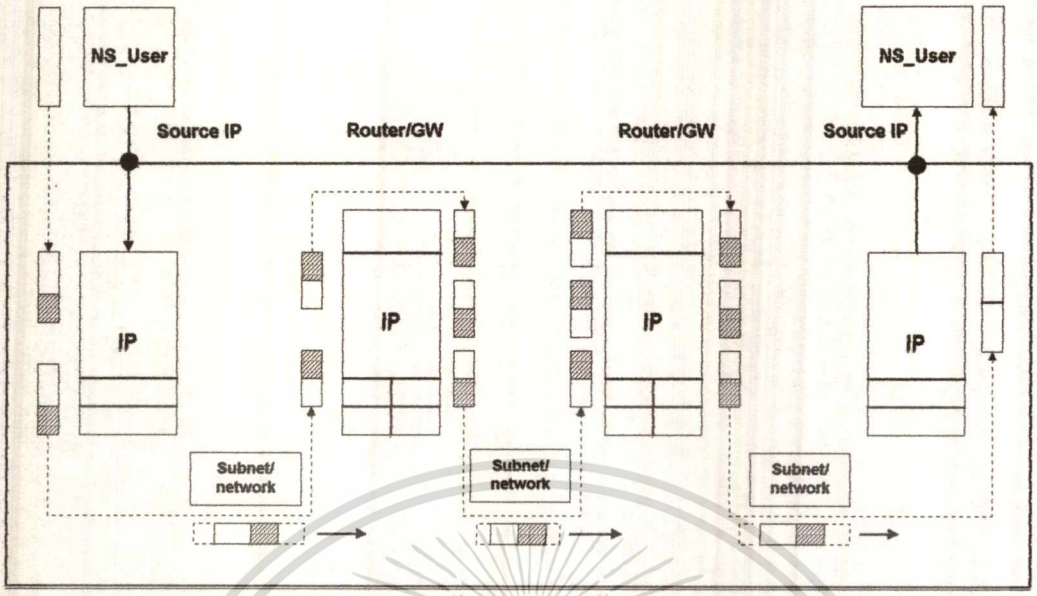
หน่วยข้อมูล IP (IP datagrams) แต่ละหน่วยจะประกอบด้วย ส่วนของข้อมูลที่รับมาจากส่วนของงาน TCP หรือ UDP และส่วนของข้อมูลนำทาง (Header) ซึ่งมีรายละเอียดดังนี้

- Version หมายเลขรุ่นของข้อกำหนด IP
- Header Length ความยาวของข้อมูลนำทาง
- Type of service วิธีการจัดการกับข้อมูล
- Total Length ความยาวของหน่วยข้อมูล
- Identification, Flags และ Fragment offset รายละเอียดที่เกี่ยวกับการแบ่งย่อยข้อมูล ซึ่งจะถูกนำมาใช้ในการรวบรวมข้อมูล
- Time to live เวลาสูงสุดที่ใช้ในการเดินทาง ซึ่งกำหนดมาจากต้นทาง เวลานี้จะลดลงเรื่อย ๆ ในระหว่างทาง ถ้าลดลงไปถึงศูนย์ หน่วยข้อมูลนั้นจะถูกกำจัดไป
- Protocol ชนิดของข้อมูลเป็น UDP หรือ TCP
- Header Checksum ค่าตรวจสอบข้อมูลนำทาง
- IP address หมายเลข internetwide IP (NSAP) ของเครื่องต้นทางและปลายทาง
- Option ข้อมูลอื่น ๆ เช่น ข้อมูลเกี่ยวกับการรักษาความปลอดภัย บันทึกเส้นทางเดินของข้อมูล และเวลาที่ข้อมูลเดินทางมาถึง เป็นต้น

3.5.3 การแบ่งส่วนย่อยของข้อมูลและการประกอบขึ้นใหม่ (Fragmentation and Reassembly)

ขนาดข้อมูลของผู้ใช้ซึ่งอ้างอิงกับ NSDU (Network service data unit) มีความจุได้ถึง 64k หรือ 65,536 bytes แต่ขนาดของหน่วยข้อมูล (packet size) ที่สามารถติดต่อกันในระบบที่ต่างกันสามารถมีได้ตั้งแต่ 128 byte สำหรับระบบ X.25 packet switching จนถึง 8000 byte สำหรับบาง LAN ดังนั้นกระบวนการ Fragmentation และ Reassembly จึงถูกนำมาใช้เพื่อ ทำให้ขนาดของข้อมูลเล็กลง และสามารถส่งไปในระบบได้ และเมื่อถึงปลายทาง IP ก็จะทำการประกอบข้อมูล (reassembly) ขึ้นมาใหม่ก่อนที่จะส่งผ่านไปยังผู้ใช้ปลายทาง ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 internet fragmentation and reassembly

อันดับแรก IP ใน Host ต้นทางจะแยกข้อมูลของผู้ใช้ (NS -User), NSDU เป็น Datagram ซึ่งมี Address กำกับเป็นเฉพาะส่วน ๆ ไปซึ่งจะถูกออกคำสั่งโดย Network ที่มันติดต่ออยู่ด้วยและส่ง Datagram ไปยัง IP ใน Gateway ตัวแรก โดยที่ IP ใน Gateway จะไม่ Reassemble NSDU แต่จะปรับปรุงในขอบเขตที่เหมาะสม และส่ง Datagram ที่ได้รับตรงไปยัง Network ที่สอง (ถ้า Network ที่สองสามารถรองรับขนาด Datagram นี้) หรือทำการ Fragment datagram ให้มีขนาดเล็กลง ซึ่งขั้นตอนนี้จะถูกทำซ้ำที่ Gateway ตัวต่อไป โดยในรูปที่ 3.7 Network หลังสุดสามารถรองรับขนาดของ Packet ได้มากกว่า Packet ที่มันได้รับข้อมูลจึงถูกส่งได้โดยตรง โดยที่จะมีการปรับปรุงในบางส่วน Header ของ Datagram เท่านั้น จากนั้น IP ใน Host ปลายทางจะทำการประกอบข้อมูล (Reassemble) ที่มันได้รับขึ้นมาใหม่ และส่งผลที่ได้รับก็คือ NSDU ไปยังผู้ใช้ (NS-user)

ในการคิดค่าเวลาสูงสุดที่ Host ต้นทางกำหนดให้ Gateway รอ Datagram (NSDU) ระหว่างแต่ละการ Assembly ซึ่งก็คือ time-to-live ซึ่งจะถูกนำติดไปที่ส่วน Header ของ Datagram ซึ่งจะถูกตั้งค่าโดย IP ใน Host ต้นทาง ซึ่งจะมีค่าลดลงเรื่อยๆในแต่ละขั้นตอนของการ Process datagram ถ้า Datagram ถูก Fragment ค่าปัจจุบันจะถูกนำไปใส่ในส่วน Header ของ Datagram ตัวใหม่ ถ้ามันถึงค่า 0 ที่จุดใด ๆ ระหว่างการ Process ใน Gateway (หรือ Host) การ Reassembly ก็จะล้มเหลวและทุกๆการ Fragment ที่เกี่ยวกับ NSDU ก็จะถูกตัดทิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Time-to-live ในแต่ละ Datagram จะเป็นจำนวนเท่าของ 1 วินาที โดยที่จำนวนของมันจะถูกลดลงโดยแต่ละ IP ซึ่งจะเปลี่ยนแปลงไปตามค่าเวลาจริงในการส่งถ่ายข้อมูลของ Network ที่ติดต่อด้วย

3.5.4 การเลือกเส้นทาง (Routing)

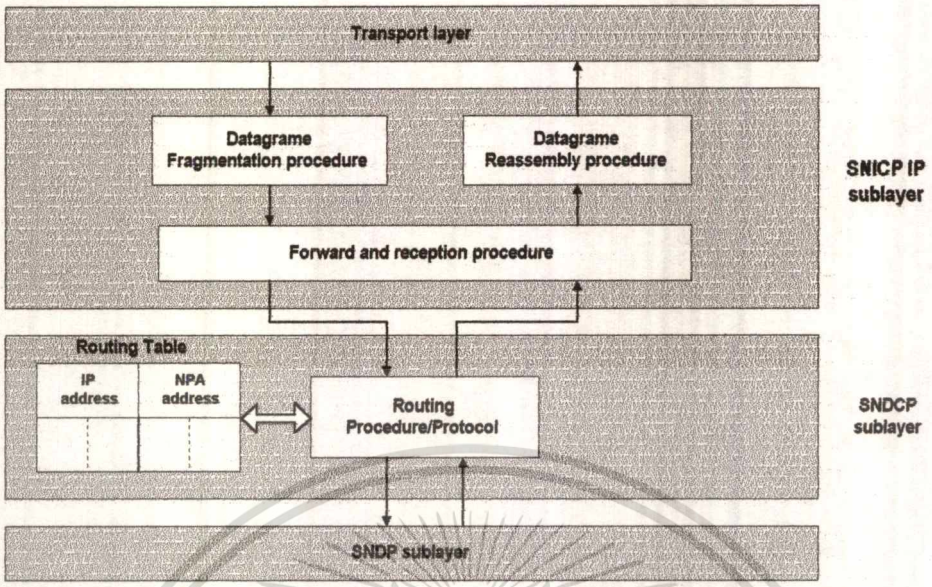
ในแต่ละ Network (หรือ subnet) ใน Internet จะมีชนิดของ IP address ที่แตกต่างกัน ซึ่งระบบ (system)-host หรือ gateway ที่ถูกต่อเข้ากับ network จะสามารถส่ง datagram ไปยังระบบอื่นได้โดยตรงเฉพาะ network ที่เหมือนกันเท่านั้น ในการเลือกเส้นทาง (routing) ให้ datagram ข้ามไปยังหลาย ๆ network IP ในแต่ละ internetwork gateway ต้องรู้ IP address ของ host ปลายทาง

ซึ่งมี 2 วิธีการพื้นฐานที่ถูกใช้ในการหาเส้นทางภายใน Internet คือ centralized และ distributed ด้วยวิธีการ centralized routing ข้อมูลเกี่ยวกับการเลือกเส้นทาง ที่เกี่ยวข้องกับแต่ละ gateway จะถูก download จาก site ส่วนกลางโดยใช้ข้อมูล network และ special network management โดย network management จะพยายามตรวจสอบ network และ host ที่ถูกเพิ่มเข้าและถอดออก และข้อบกพร่องที่จะถูกวินิจฉัยและตรวจซ่อม

ด้วยวิธีการ Distributed routing ทุก ๆ host และ gateway จะร่วมกันในการแบ่งปัน วิธีการในการรับประกันว่า ข้อมูลเกี่ยวกับการเลือกเส้นทางในแต่ละ system, host และ gateway จะถูกทำให้ทันสมัย และสอดคล้องกัน ข้อมูลเกี่ยวกับการเลือกเส้นทางจะถูกจดจำไว้โดยแต่ละระบบ ในรูปของ routing table ซึ่งจะมี NPA address ไว้ใช้ในการส่งแต่ละ datagram ซึ่ง Internet จะใช้วิธีการแบบนี้

ขั้นตอนการ Routing ที่เกี่ยวกับ IP ขั้นตอนแรกจะอ่าน IP address (NSAP) ปลายทางจากภายใน datagram และใช้มันในการหาการตอบสนอง PA address ของ host หรือ gateway จาก routing table ในส่วนที่เพิ่มเติมชุดของ routing protocol จะถูกใช้เพิ่มและรักษาส่วนที่อยู่ในแต่ละ routing table ในแบบของ distributed ซึ่งรูปแบบทั่วไปที่ถูกใช้ภายใน host IP แสดงดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 รูปแบบทั่วไปของการเลือกเส้นทางภายใน Host

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบและสร้าง

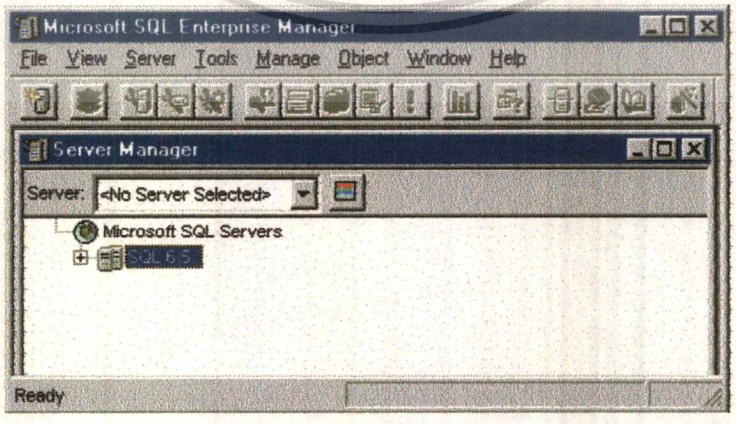
4.1 การสร้างฐานข้อมูล

ในโครงงานนี้ได้เขียนโปรแกรม เป็นแบบจำลองของอุปกรณ์ทั้ง 3 เครื่องเพื่อง่ายและสะดวกต่อการใช้งาน ตัวโปรแกรมควบคุมเครื่องมือวัด ได้แบ่งออกเป็น 2 ส่วนคือ โปรแกรมในส่วนของผู้ใช้ (User) ที่มีหน้าที่คือ รับค่าและฟังก์ชันต่าง ๆ ที่ผู้ใช้ต้องการส่งไปยัง MS SQL Server และส่วนที่สองคือ ส่วนโปรแกรมที่อยู่ที่เครื่องคอมพิวเตอร์ ที่เป็นตัวควบคุมชุดเครื่องมือวัด GPIB(Master) ซึ่งมีหน้าที่รับค่าจาก ดาต้าเบสใน MS SQL Server มาประมวลผลและสั่งงาน ชุดเครื่องมือวัด และรับค่าที่ได้จากเครื่องมือวัด ส่งกลับไปยังดาต้าเบส

เนื่องจากการทำงานของโปรแกรมทั้งทางด้านผู้ใช้ และเครื่องคอมพิวเตอร์ที่ควบคุมชุดGPIB จำเป็นต้องทำงานร่วมกับดาต้าเบส ในการเขียนโปรแกรม จึงต้องมีการสร้างตารางฐานข้อมูล เพื่อเก็บข้อมูลต่าง ๆ ของ อุปกรณ์แต่ละเครื่อง ซึ่งเราจะใช้โปรแกรม MS SQL 6.5 เป็นตัวจัดการเกี่ยวกับฐานข้อมูลซึ่งโปรแกรม MS SQL จะรวมโพรโตคอลที่ใช้ในการติดต่อรับส่งข้อมูลในเครือข่ายอินเตอร์เน็ต คือ TCP/IP (Transmission Control Protocol/Internet Protocol) ไว้แล้วและจะจัดการรับส่งข้อมูลผ่านเครือข่าย ให้โดยอัตโนมัติ

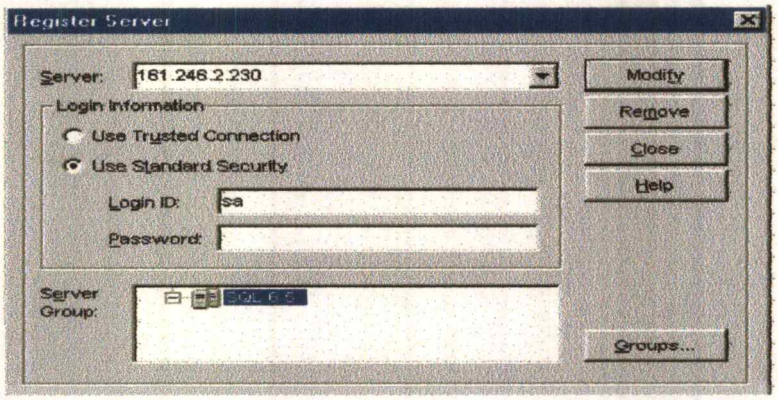
สำหรับการสร้างตาราง (Table) ฐานข้อมูล (database) สามารถทำได้โดยใช้ Microsoft SQL Enterprise Manager ใน MS SQL 6.5 ดังนี้

1. ทำการติดตั้งตัว Sever ฐานข้อมูลที่เราต้องการใช้ใน MS SQL Sever ซึ่งในที่นี้คือ หมายเลข IP 161.246.2.230 โดยกำหนด login ID เป็น SA ดังรูปที่ 4.1และ รูปที่ 4.2 ตามลำดับ



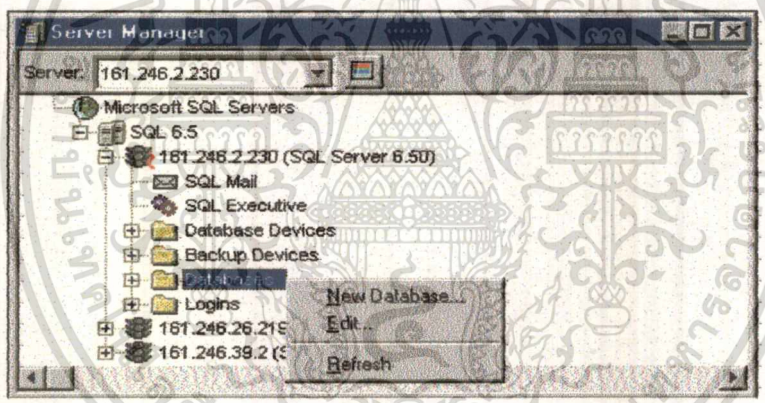
รูปที่ 4.1 แสดงสถานะเริ่มต้นของ SQL Enterprise Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



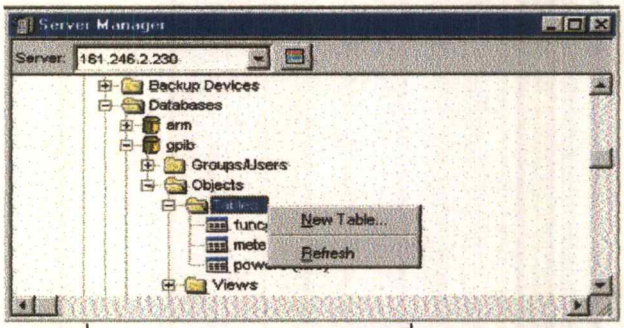
รูปที่ 4.2 แสดงการติดตั้ง Sever ลงในโปรแกรม

2. ทำการสร้างฐานข้อมูลโดยสร้างในส่วนของฐานข้อมูลใหม่ ที่ New Database ดังรูปที่ 4.3 โดยได้ทำการสร้างดาต้าเบสชื่อ gpib



รูปที่ 4.3 แสดงการสร้างฐานข้อมูลใหม่

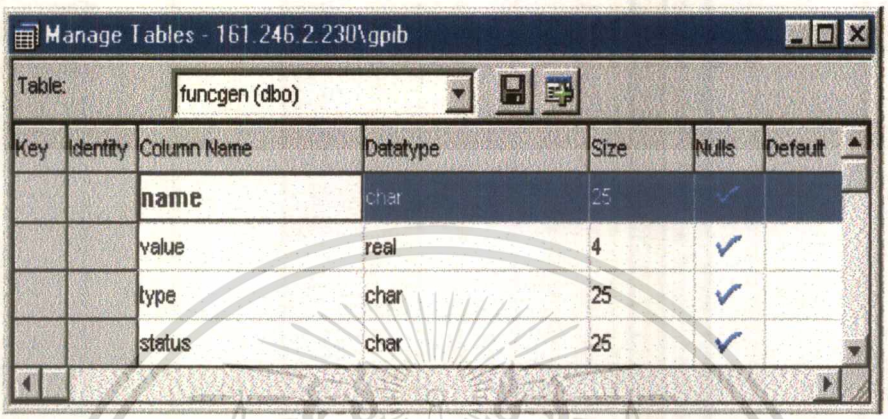
3. ทำการสร้างตาราง (Table) เพื่อเก็บข้อมูลที่เราต้องการโดยจะอยู่ในส่วน ของ object ดังรูปที่ 4.4



รูปที่ 4.4 แสดงการสร้างตารางเพื่อเก็บข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการกำหนดรูปแบบ ข้อมูลที่ต้องการใช้ซึ่งในที่นี้ได้กำหนด Column ของข้อมูลไว้ 4 ชนิด คือ Name, Value, Type และ Status ซึ่งเป็นตัวอย่างตารางของ Function Generator โดยส่วนที่ให้เช็คว่าเป็น Nulls ได้หรือไม่ คือส่วนที่จะกำหนดว่า สามารถให้มีข้อมูลว่างได้หรือไม่ ดังรูปที่4.5



รูปที่ 4.5 แสดงการกำหนดรูปแบบข้อมูลใน table ใหม่

นอกจากนี้เรายังสามารถจัดการ เกี่ยวกับขนาดของดาต้าเบสที่เราต้องการ จองพื้นที่ของหน่วยความจำไว้ โดย Set ค่าใน Manage Database ซึ่งขนาดของ ดาต้าเบสขึ้นอยู่กับจำนวนข้อมูลที่ต้องใช้

4.2 โปรแกรมควบคุม Programmable Power Supply

โปรมแกรมในส่วนของ Programmable Power Supply ประกอบด้วยโปรแกรม User Power Supply ซึ่งเป็นส่วนตัวโปรแกรมที่ผู้ใช้และ Master Power Supply เป็นโปรแกรมที่อยู่ที่เครื่องคอมพิวเตอร์ที่เป็นตัวอินเตอร์เฟซ กับ Power Supply (HM8142)

4.2.1 โปรแกรม User Power Supply

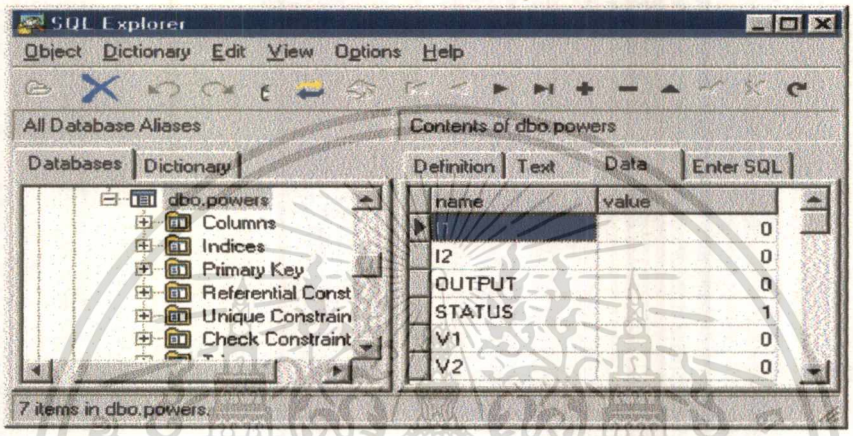
โปรแกรม User Power Supply มีหน้าที่การทำงานคือ รับค่าจากฟังก์ชันจากผู้ใช้ และนำค่าที่ได้ส่งไปให้กับดาต้าเบส โดยค่าที่มีการเปลี่ยนแปลงของตัว Power Supply ประกอบด้วย

1. V1 : ค่าแรงดัน V1
2. V2 : ค่าแรงดัน V2
3. I1 : ค่ากระแส I1
4. I2 : ค่ากระแส I2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

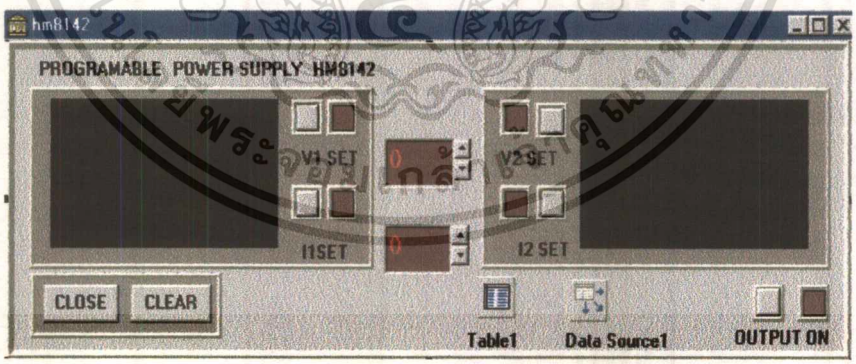
5. Output : สถานะทาง Output ของเครื่อง (On/Off)
6. Status : สถานะเคลียร์ (ถ้าอยู่ในสถานะเคลียร์จะ Set ทุกค่าคือ V1 , V2, I1, และ I2 มีค่าเท่ากับศูนย์)

โดยเราต้อง นำค่าเหล่านี้ ไปใส่ไว้ใน ฐานข้อมูลซึ่งได้สร้างตารางชื่อ dbo.powers ในดาต้าเบสชื่อ gpib ดังรูปที่ 4.6



รูปที่ 4.6 แสดงข้อมูลของ HM8142 ในตาราง dbo.powers

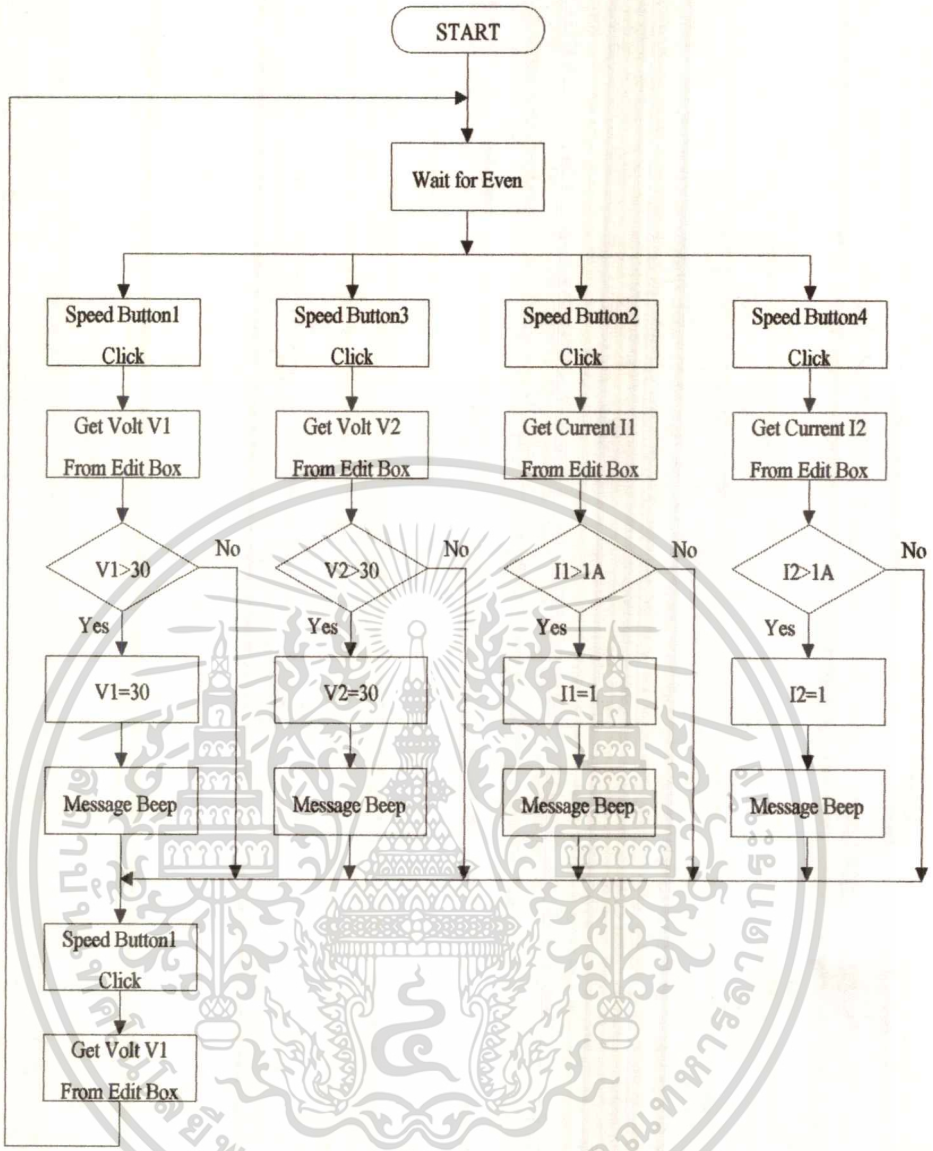
โดยรูปร่างของแบบจำลองของ Programmable Power Supply แสดงไว้ในรูปที่ 4.7



รูปที่ 4.7 แบบจำลองของ Programmable Power Supply

สำหรับการทำงานของโปรแกรม แสดงใน ผังการทำงาน โปรแกรมดังรูปที่ 4.8 ซึ่ง จะเป็นลักษณะการทำงานตามเหตุการณ์ (Event) คือ จะทำงานเฉพาะตอนผู้ใช้ กระทำกับส่วนประกอบต่าง ๆ โดยเป็นอิสระไม่ขึ้นกับตัวอื่น ๆ โดยจะมีการจำกัดค่าของ แรงดัน และ กระแสไว้ที่ 30 V และ 1A ตามคุณสมบัติของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



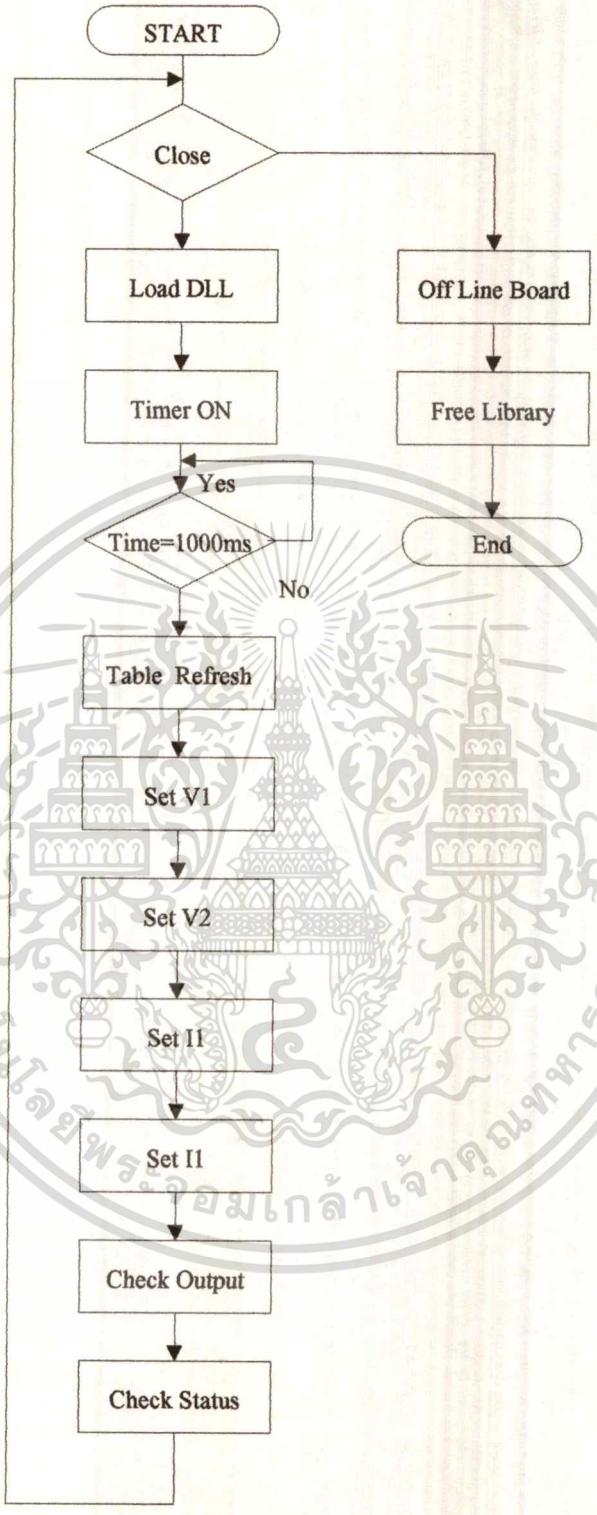
รูปที่ 4.8 ผังการทำงานของ User Power Supply

4.2.2 โปรแกรม Master Programmable Power Supply

โปรแกรมส่วนนี้จะมีหน้าที่ รับค่าจากตาราง dbo.powers ในฐานข้อมูลมาสั่งงานตัว Power Supply โดยจะนำค่าจากตารางมาสั่งงานทุก ๆ 1 วินาที เพื่อให้การทำงานของ Power Supply มีการตอบสนอง ทันต่อข้อมูลใหม่ ๆ จากผู้ใช้

ในการอินเตอร์เฟซ กับตัว Power Supply นั้นเนื่องจากตัวบอร์ดของ GPIB จะมีคำสั่งเฉพาะของบอร์ดที่สนับสนุน เฉพาะภาษา C และ BASIC ดังนั้นจึงต้องมีการเรียกใช้ไลบรารี่ของ GPIB 32 แล้ว นำมาแปลงเป็นฟังก์ชันในรูป TObject ซึ่งเป็นรูปแบบของโปรแกรมภาษา Delphi การทำงานของโปรแกรมแสดงดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผังการทำงานของ Master Programmable Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 โปรแกรมควบคุม Programmable Function Generator

โปรแกรมในส่วนของ Programmable Function Generator ประกอบด้วยโปรแกรม User Function Generator ซึ่งเป็นโปรแกรมในส่วนของผู้ใช้และ Master Programmable Function Generator เป็นโปรแกรมที่อยู่ที่เครื่องคอมพิวเตอร์ ที่เป็นตัวอินเตอร์เฟซกับ Programmable Function Generator HM8130

4.3.1 โปรแกรม User Function Generator

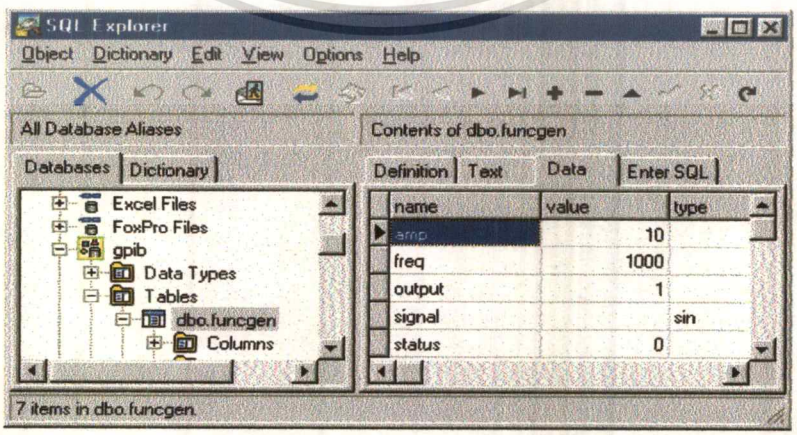
ตัวโปรแกรม User Function Generator มีหน้าที่การทำงาน คือ รับค่าและฟังก์ชันจากผู้ใช้และนำค่าที่ได้ส่งไปยังดาต้าเบส โดยค่าที่มีการเปลี่ยนแปลงของ Programmable Function Generator ประกอบด้วย

1. Freq : ค่าความถี่ของสัญญาณ
2. Amp : ค่าแอมปลิจูดของสัญญาณ
3. Signal : ชนิดของสัญญาณ
4. Output: สถานะทาง Output กำหนดให้มีสองค่าคือศูนย์และหนึ่ง

   0 = Output on

   1 = Output off
5. Status : สถานะเคลียร์ (ถ้าอยู่ในสถานะเคลียร์จะเซตให้มีค่าความถี่ 1 kHz ค่าแอมปลิจูด 10 V สัญญาณชนิด Sine)

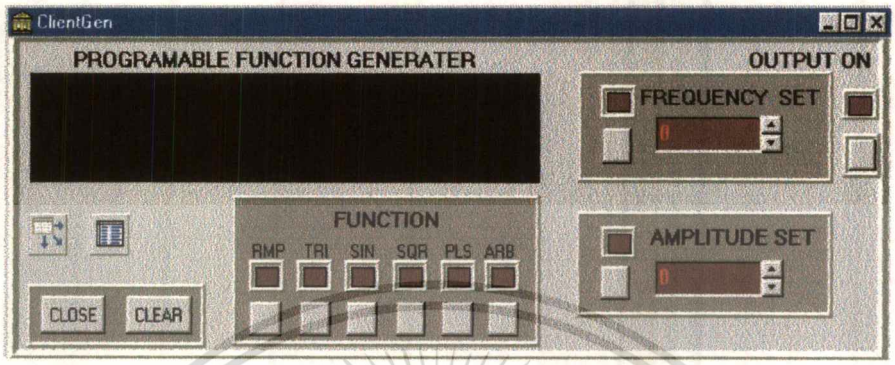
ค่าเหล่านี้ที่ User กำหนดจะถูกนำไปใส่ไว้ที่ดาต้าเบส ซึ่งได้สร้างตารางชื่อ dbo.funcgen ในดาต้าเบส ชื่อ gpib ดังรูปที่ 4.10



รูปที่ 4.10 แสดงข้อมูลในตาราง dbo.funcgen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

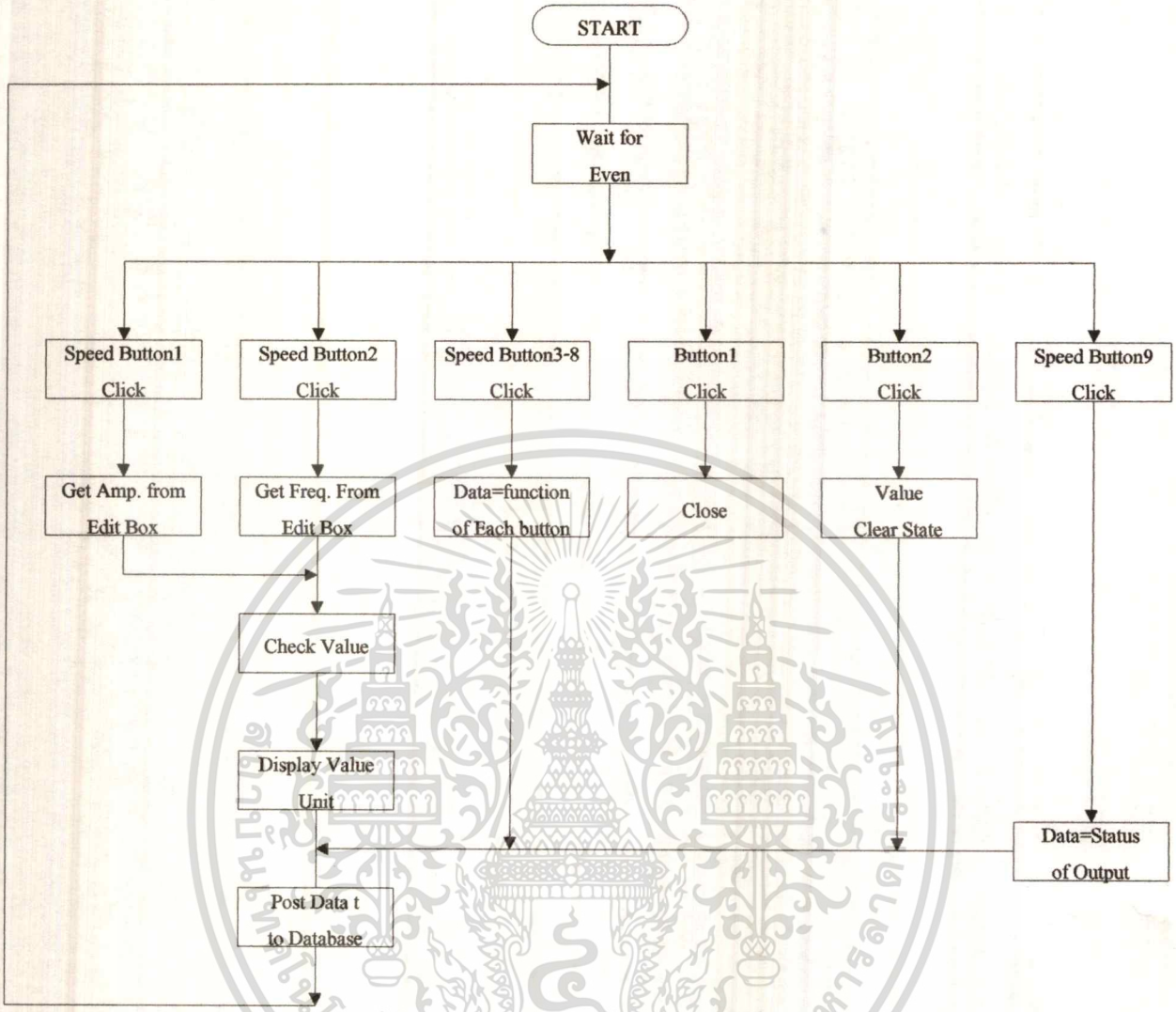
รูปร่างของแบบจำลองหน้าปัดของ Programmable Function Generator แสดงไว้ในรูปที่ 4.11



รูปที่ 4.11 แบบจำลองของ Programmable Function Generator

สำหรับการทำงานของ โปรแกรมมีลักษณะการทำงานคล้ายกับ Power Supply คือ ทำงานตาม Event และส่งค่าไปยังดาต้าเบส แตกต่างกันที่ชนิดของข้อมูล ซึ่งการทำงานของโปรแกรม แสดงตัวผังการทำงานในรูป 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

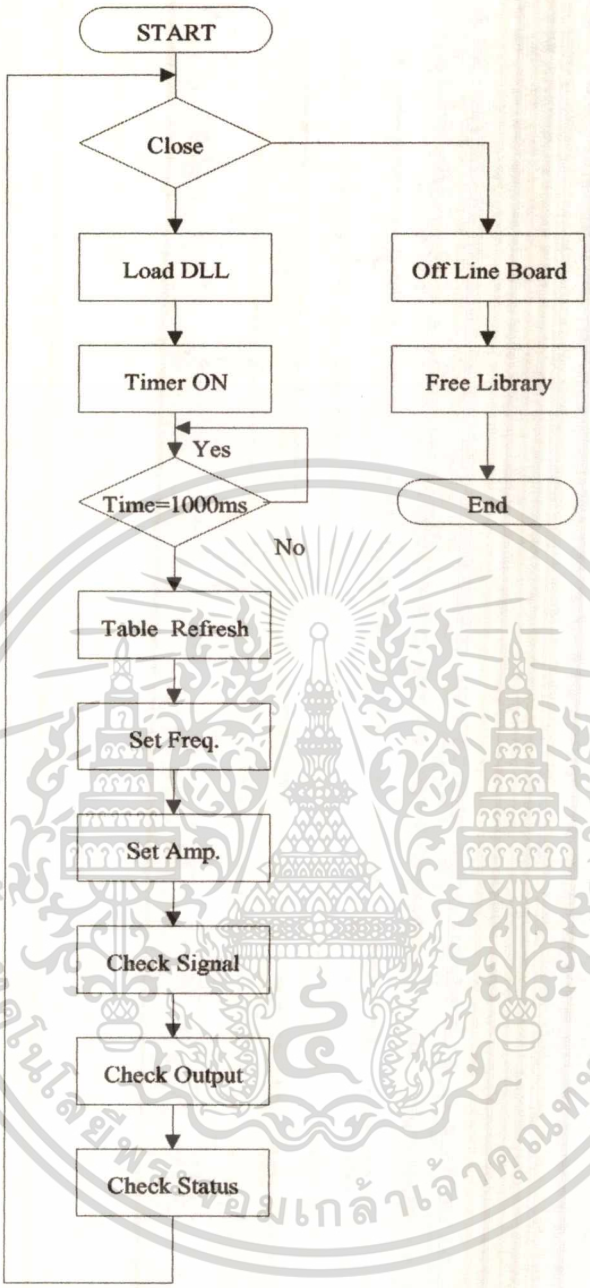


รูปที่ 4.12 ผังการทำงานของ User Function Generator

4.3.2 โปรแกรม Master Programmable Function Generator

การทำงานจะรับค่าจากตาราง dbo.funcgen ในฐานข้อมาสั่งงาน Programmable Function Generator ให้ทำงาน โดยจะมีการ Refresh ข้อมูลทุก ๆ 1 วินาที เพื่อให้ทันต่อการเปลี่ยนแปลงของผู้ใช้ ซึ่งขั้นตอนแรกก็ต้องโหลดฟังก์ชันการทำงานของไลบราลี่ของ GPIB มาใช้ ซึ่งรายละเอียดของแต่ละคำสั่งดูได้จากคู่มือของเครื่อง ซึ่งจะไม่เหมือนกันในแต่ละเครื่อง การทำงานของโปรแกรม แสดงในรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผังการทำงานของ Master Programmable Function Generator

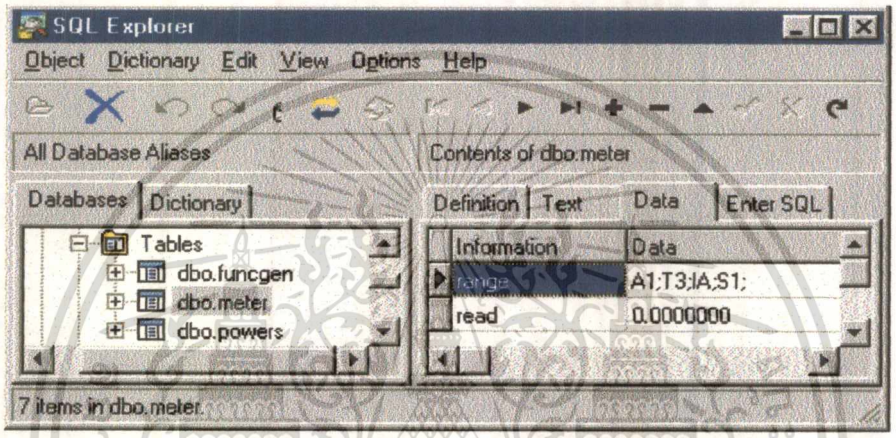
4.4 โปรแกรมควบคุม Programmable Multimeter

โปรแกรมในส่วนของ Programmable Multimeter ประกอบด้วยโปรแกรม User Multimeter ซึ่งเป็นโปรแกรมด้านผู้ใช้และ Master Programmable Multimeter เป็นโปรแกรมที่อยู่ที่เครื่องคอมพิวเตอร์ที่เป็นตัวอินเตอร์เฟซ กับ Programmable Multimeter (HM8112-2) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

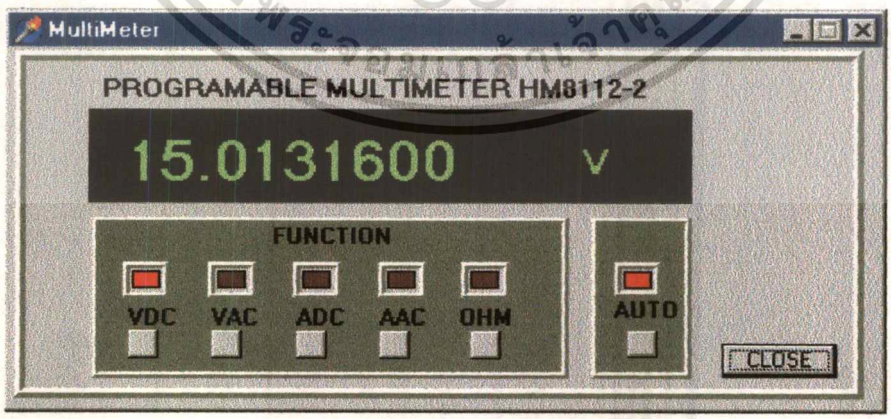
4.4.1 โปรแกรม User Multimeter

ตัวโปรแกรม User Multimeter มีลักษณะการทำงานแตกต่างไปจาก อุปกรณ์สองตัวที่กล่าวมา คือมีหน้าที่ส่งค่า ฟังก์ชันที่ใช้กำหนดไปยังดาต้าเบส และนำข้อมูลที่รับมาจากดาต้าเบส ซึ่งเป็นข้อมูลที่ตอบสนองมาจาก Multimeter มาแสดงผลที่หน้าจอของผู้ใช้โดยข้อมูลต่าง ๆ ของ Multimeter จะประกอบด้วยสองส่วน คือข้อมูลคำสั่งและข้อมูลที่อ่านมาจาก Multimeter ดังแสดงในรูป 4.14 ซึ่งเป็นข้อมูลภายในตาราง dbo.meter



รูปที่ 4.14 ข้อมูลในตาราง dbo.meter

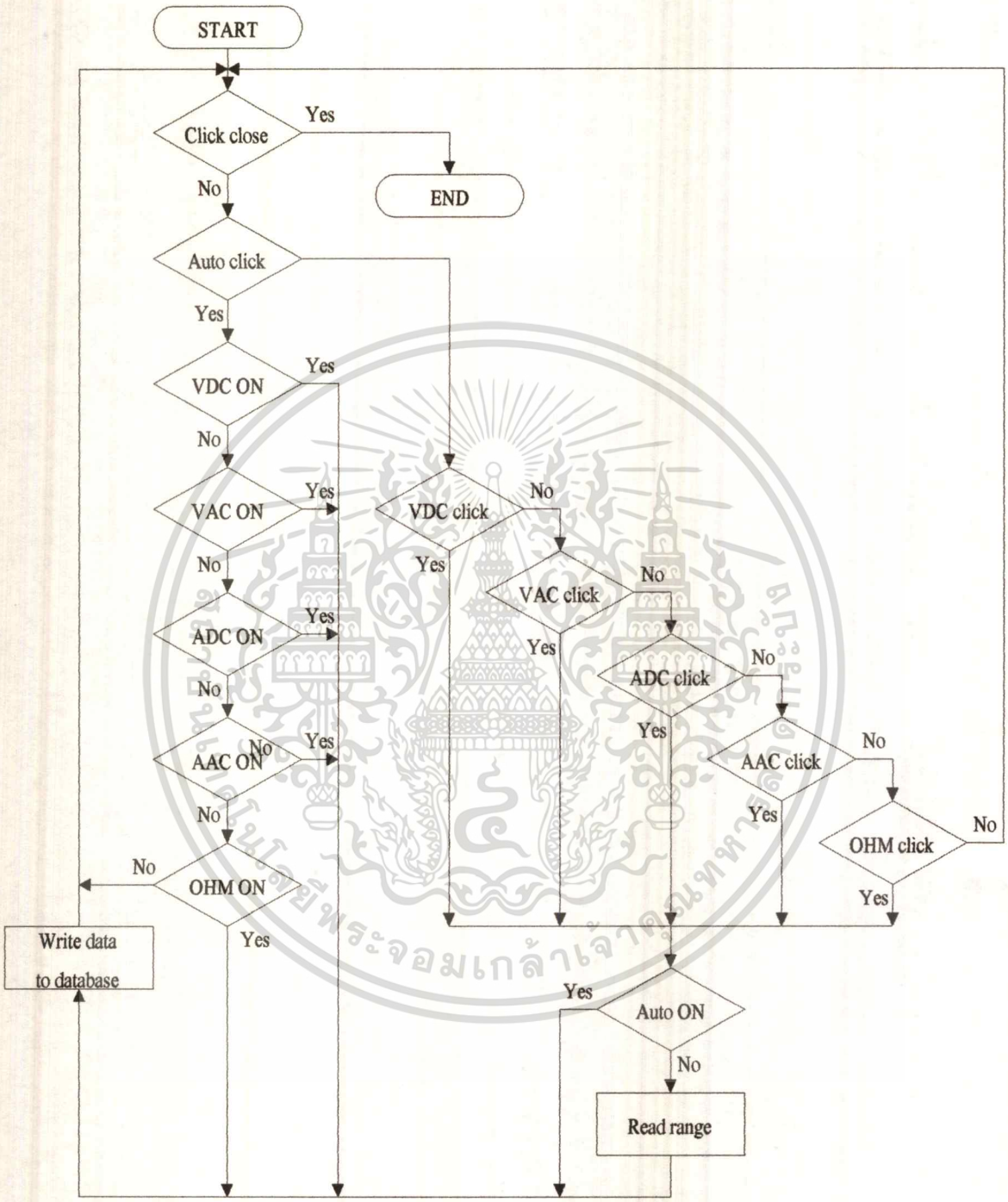
โดยรูปร่างของแบบจำลองหน้าปัดของ Programmable Multimeter แสดงไว้ในรูปที่ 4.15



รูปที่ 4.15 แบบจำลองของ Programmable Multimeter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทำงานของโปรแกรมแสดง ดังรูปที่ 4.16

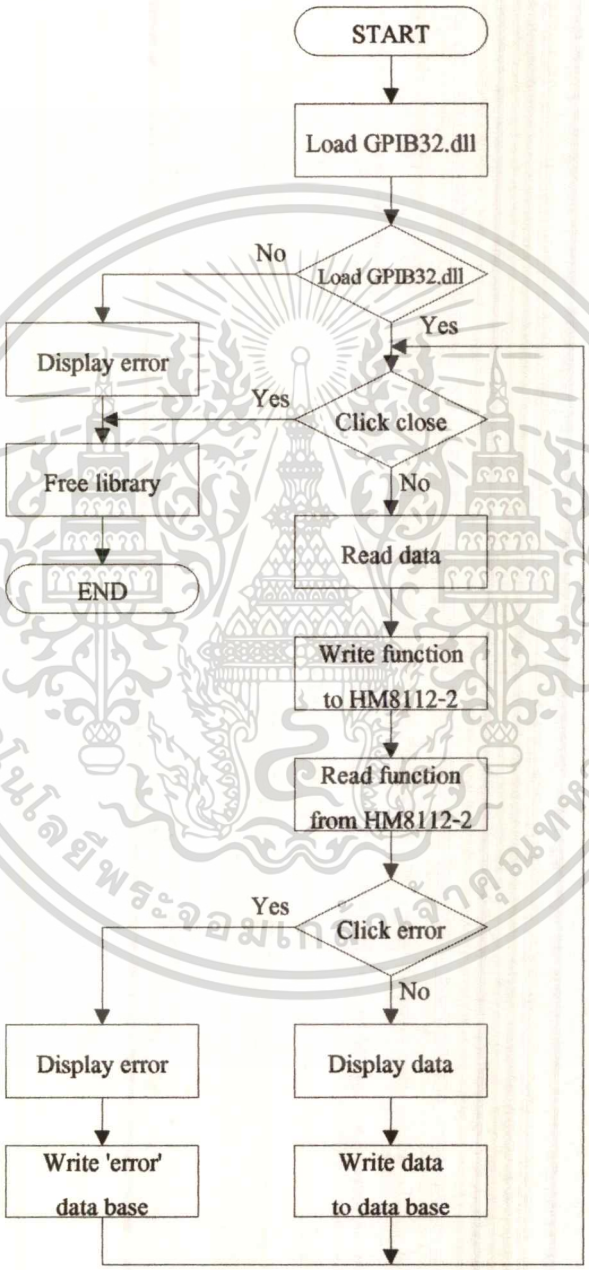


รูปที่ 4.16 ผังการทำงานของ โปรแกรม User Multimeter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 โปรแกรม Master Programmable Multimeter

โปรแกรมส่วนนี้มีหน้าที่รับคำสั่ง จากตาราง dbo.meter ในฐานข้อมูลมาสั่งงานตัว Programmable Multimeter และอ่านข้อมูลจากบัฟเฟอร์ของ Programmable Multimeter ส่งไปให้ฐานข้อมูล การทำงานของโปรแกรมแสดงดังรูปที่ 4.17

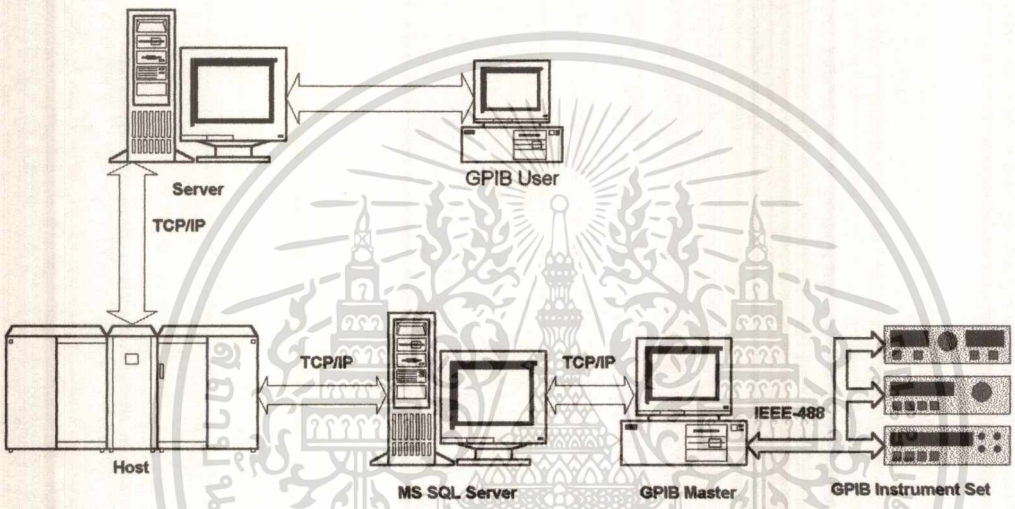


รูปที่ 4.17 ผังการทำงานของ Master Programmable Multimeter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและผลการทดลอง

ในการทดลองบทนี้เป็นการทดลองเพื่อทดสอบว่า สามารถควบคุมชุดเครื่องมือวัด GPIB ได้จากเครื่องคอมพิวเตอร์ที่ต่อกับระบบอินเตอร์เน็ต ซึ่งสำหรับระบบที่ใช้ในการทดลองจะเป็นดังรูปที่ 5.1



รูปที่ 5.1 แสดงระบบคอมพิวเตอร์ที่ใช้ในการทดลอง

โดยการทดลองจะเป็นการทดสอบการทำงานผ่าน LAN 2 Network ซึ่งแต่ละ network ต่างก็เป็นเนทเวิร์คย่อยของเครือข่ายอินเตอร์เน็ต (Sub Network) โดยเครื่องคอมพิวเตอร์ของ SQL Sever จะอยู่ที่ IP 161.246.2.230 และเครื่อง user อยู่ที่ 161.246.26.219 การทดลองจะเป็นการทดลองสั่งงานอุปกรณ์ทีละตัว

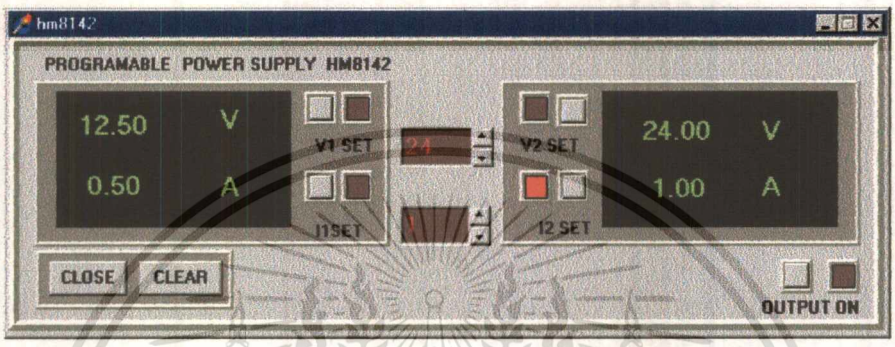
5.1 การทดลองสั่งงาน Programmable Power Supply (HM8142)

ในการทดลอง Programmable Power Supply นี้จะเป็นการทดลองว่า ค่าแรงดัน และค่ากระแสที่ตั้งไว้โดยคอมพิวเตอร์ของ user จะมีค่าของข้อมูลที่ปรากฏในดาตาเบสของ MS SQL Sever ใน Database name ชื่อ gpib ใน Table ที่ชื่อ dbo.power เป็นอย่างไร และผลที่ได้ที่ตัว Programmable Power Supply มีค่าเท่าไรเป็นไปตามที่ User ได้ตั้งไว้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 การทดลองตั้งค่าแรงดัน V1 และ V2, I1 และ I2 โดย User

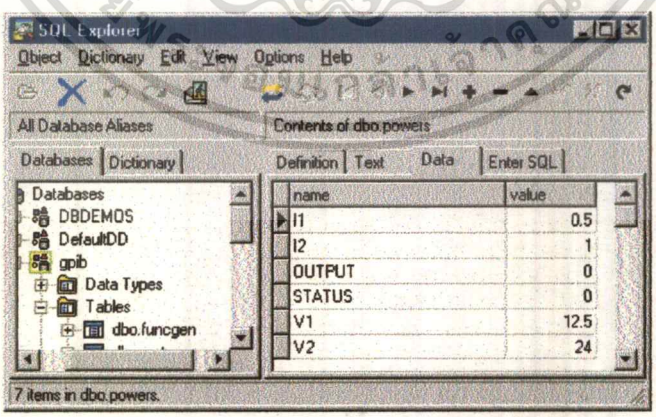
การตั้งค่าแรงดันที่ User สามารถทำได้ 2 แบบ คือ ป้อนค่าโดยการพิมพ์ค่าแรงดันที่ Edit box หรือ Click ที่ลูกศรที่อยู่ติดกับ Edit box โดยลูกศรชี้ขึ้นจะเป็นการเพิ่มค่า ลูกศรชี้ลงจะเป็นการลดค่า จากนั้นถ้าต้องการตั้งค่าแรงดันที่ V1 หรือ V2 ก็ทำได้โดยการ Click ที่ V1 Set หรือ V2 Set ตามลำดับจากการทดลอง Set ค่า V1=12.5 V และ V2=24 V



รูปที่ 5.2 แสดงการตั้งค่า V1 และ V2, I1 และ I2

จากการทดลองจะทดลองป้อนค่า V1=12.5 V และ V2=24 V, I1=0.5 A และ I2=1 A ซึ่งแสดงดังรูปที่ 5.2 เป็นโปรแกรมในส่วนของผู้ใช้ (GPIB User)

จากนั้นจะเช็คว่าค่าที่อยู่ใน Table ของดาตาเบสเมื่อผู้ใช้ป้อนค่าแล้วจะมีค่าอย่างไรซึ่งดูได้จาก Table ชื่อ dbo.powers ในดาตาเบสชื่อ gpib ของ MS SQL Sever จะได้ค่าที่ปรากฏดังนี้



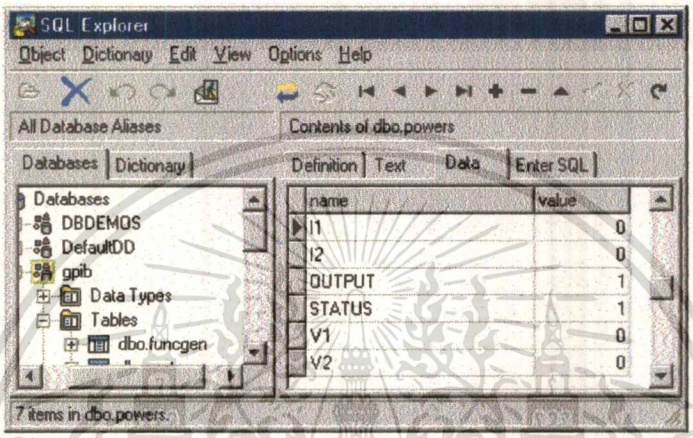
รูปที่ 5.3 ค่าใน Table dbo.powers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 การทดลอง Set สถานะของเอาท์พุทและสถานะเคลียร์

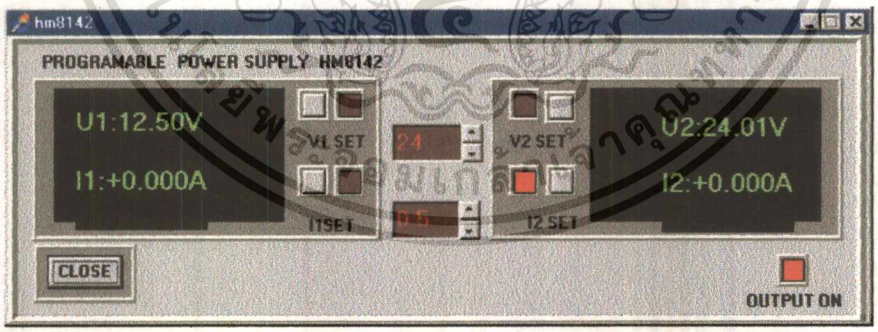
- เมื่อผู้ใช้ Click ที่ปุ่ม clear จะใช้ค่าที่ฟิลด์ Value ของเรคอร์ด status=1
- เมื่อผู้ใช้ Click ที่ปุ่ม output on จะใช้ค่าที่ฟิลด์ Value ของเรคอร์ด output=1

ซึ่งค่าที่ปรากฏที่ตาราง Table dbo.powers เป็นดังรูปที่ 5.4 เป็นสถานะที่ผู้ใช้กดปุ่ม output on=1



รูปที่ 5.4 ค่าใน Table dbo.powers ที่สถานะ clear/output on

ค่าที่ปรากฏที่หน้าปัดของโปรแกรม Programmable Power Supply จะเป็นดังรูปที่ 5.5



รูปที่ 5.5 แสดงค่าที่ปรากฏที่โปรแกรม Programmable Power Supply ด้าน Master

5.2 การทดลองสั่งงาน Programmable Function Generator (HM8130)

ในการทดลอง Programmable Function Generator นี้จะเป็นการทดลองว่าค่า ความถี่, แอมปลิจูด และ สัญญาณ ชนิดต่าง ๆ ที่ตั้งโดยผู้ใช้คอมพิวเตอร์ของผู้ใช้ (user) จะมีค่าของข้อมูลที่ปรากฏในดาตาเบสของ MS SQL Sever ใน Database name ชื่อ gpib ใน Table ที่ชื่อ

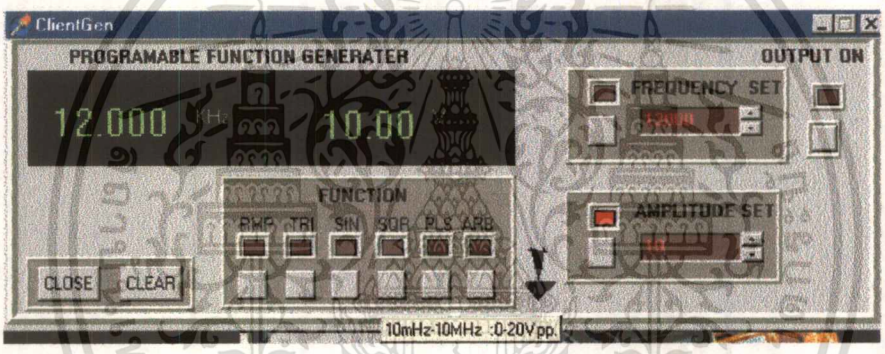
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dbo.funcgen มีค่าเป็นอย่างไร และผลที่เครื่องทำงานตามคำสั่งเป็นจริงอย่างไร ตรงตามที่ตั้งไว้หรือไม่

5.2.1 การทดลองตั้งค่า ความถี่, แอมปลิจูด และ ชนิดของสัญญาณ

5.2.1.1 การตั้งค่าโดย User

ในขั้นตอนนี้ User ทำการตั้งค่า ความถี่, แอมปลิจูด และ ชนิดของสัญญาณ ตามต้องการ แต่เนื่องจากความสามารถของ Programmable Function Generator (HM8130) ในการกำเนิดสัญญาณต่าง ๆ มีค่าจำกัดเมื่อ User ตั้งค่าเกิน Range ที่สูงสุดที่จะสามารถกำเนิดได้ ก็จะได้ค่าเท่ากับค่าสูงสุดที่สามารถกำเนิดได้ โดยที่แต่ละชนิดของสัญญาณก็จะมีขีดจำกัดของค่าความถี่และ แอมปลิจูดสูงสุดต่างกัน ซึ่งได้แสดงไว้เป็น Flow Hint ที่ปุ่มตั้งค่านั้น ๆ ดังรูที่ 5.6



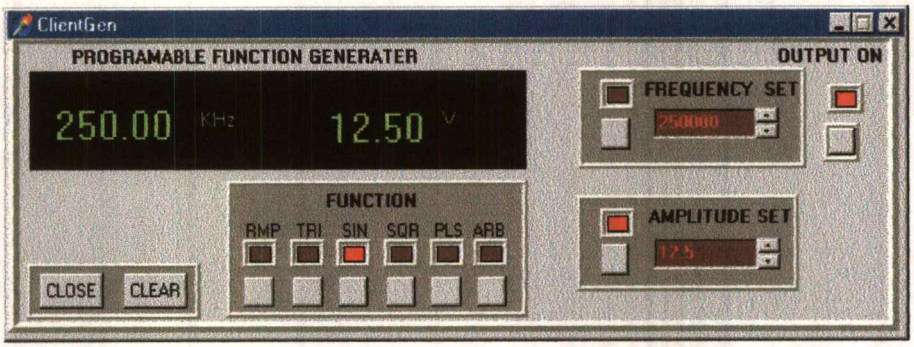
รูปที่ 5.6 แสดงค่าสูงสุดของสัญญาณที่ HM8130 สามารถกำเนิดได้

ในการทดลองด้าน User ได้ Set ค่าต่าง ๆ ไว้ดังนี้

- Frequency = 250 kHz
- Amplitude = 12.5 Volt
- Signal Type = sin
- Output = ON

จะได้ค่าที่ปรากฏที่ Monitor ของ User ดังรูที่ 5.7

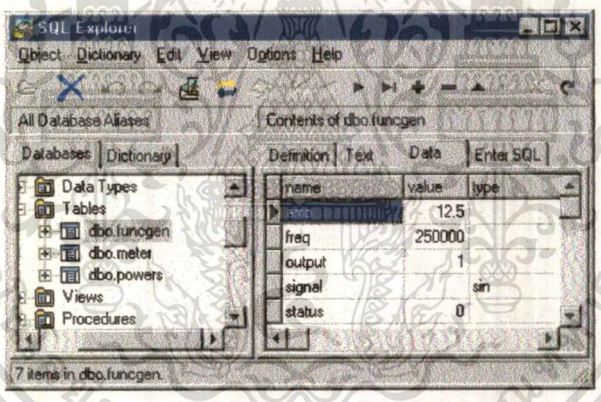
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงค่าที่ปรากฏที่หน้าปัดของโปรแกรมด้าน User

5.2.1.2 **ข้อมูลในดาตาเบสของ Function Generator**

ข้อมูลที่ User สั่งงานจะถูกนำมาเก็บไว้ที่ Database name ชื่อ gpib ใน Table ชื่อ dbo.funcgen ซึ่งค่าที่ได้จะเป็นดังรูปที่ 5.8

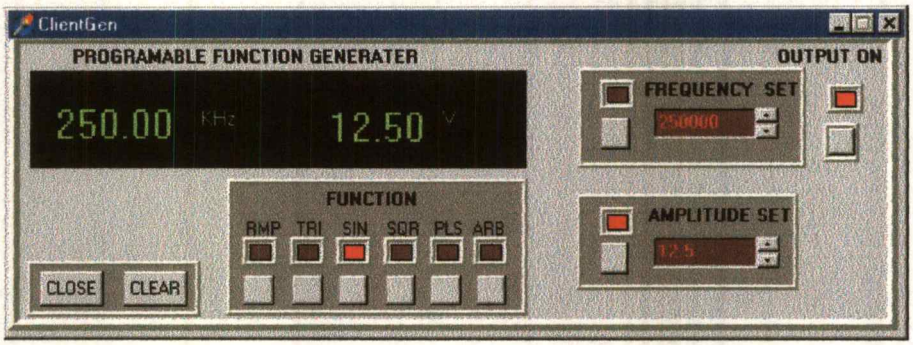


รูปที่ 5.8 แสดงข้อมูลใน dbo.funcgen

5.2.1.3 **ค่าที่ปรากฏจริงที่ Programmable Function Generator**

คอมพิวเตอร์ที่ติดต่ออยู่กับ Programmable Function Generator จะนำค่าจากดาตาเบสมาสั่งงาน HM8130 ซึ่งค่าที่ได้แสดงดังรูปที่ 5.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 ค่าที่ปรากฏที่หน้าปัดของโปรแกรมด้าน Master

ซึ่งค่าที่แสดงเป็นค่าของโปรแกรมด้าน Master GPIB ซึ่งจะมีค่าเท่ากับค่าที่ HM8130 จริง เพราะเป็นค่าที่อ่านมาจาก Buffer ของ HM8130

5.3 การทดลองสั่งงาน Programmable Multimeter (HM8112-2)

ในการทดลอง Programmable Multimeter นี้จะทดลองว่า Multimeter สามารถรับคำสั่งจาก User และส่งค่าให้ User ได้อย่างถูกต้องหรือไม่ ซึ่งฟังก์ชันของ Multimeter ได้แก่ เลือกชนิดของการวัด และเลือกย่านของการวัด

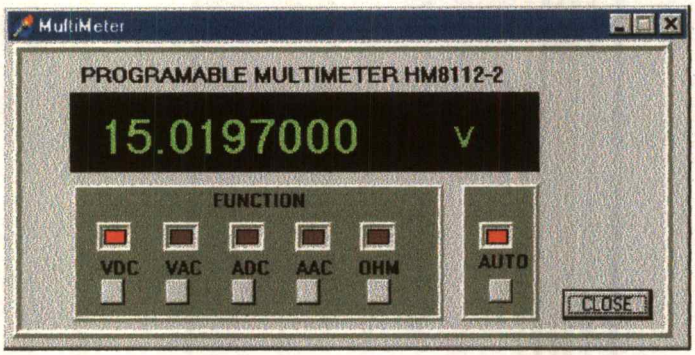
5.3.1 การทดลองตั้งค่าโดย User

ในขั้นตอนนี้ User จะทำการตั้งชนิดของสัญญาณ และอ่านค่าของสัญญาณที่วัดได้ ย่านของ Multimeter ที่ได้เขียนโปรแกรมไว้มีดังนี้

1. VDC
2. VAC
3. ADC
4. AAC
5. OHM
6. Auto Range

ในการทดลองนี้จะแสดงถึงการตั้งค่าวัด VDC และอ่านค่าจาก Power Supply ซึ่งต่ออยู่ด้าน Master GPIB โดยจะมี Range เป็น Auto Range ค่าที่ปรากฏที่หน้าปัดของ Multimeter จะเป็นค่าที่อ่านมาจากดาต้าเบส และค่าในดาต้าเบสจะได้รับจาก Buffer ของ Multimeter ดังแสดงในรูปที่ 4.10

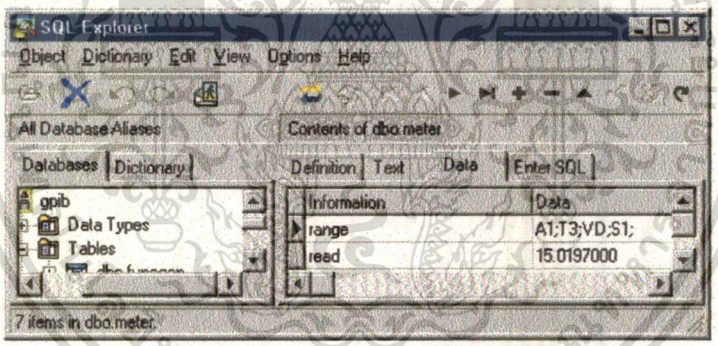
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดงหน้าปัดของโปรแกรมด้าน GPIB User

5.3.2 ข้อมูลในดาต้าเบสของ Multimeter

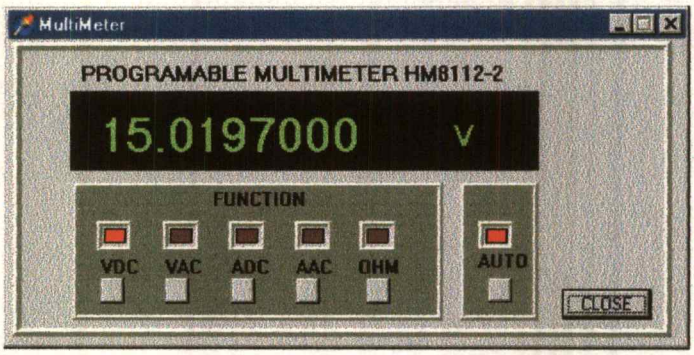
จะประกอบไปด้วยชุดคำสั่งที่ได้รับจาก User เมื่อ User ทำการตั้งค่าการวัด และอีกส่วนหนึ่งเป็นข้อมูลที่อ่านมาจาก Buffer ของ Multimeter ซึ่งค่าที่ได้จากการตั้งค่าวัด VDC และวัดค่าจาก Power Supply แสดงได้ดังรูปที่ 5.11



รูปที่ 5.11 ข้อมูลในดาต้าเบสของ Multimeter

คอมพิวเตอร์ที่เป็นตัวควบคุม Multimeter จะนำคำสั่งจากดาต้าเบสมาสั่งการที่ Multimeter ซึ่งจะแสดงผลพร้อมกับส่งค่าที่อ่านได้ไปยังดาต้าเบส ซึ่งค่าที่ปรากฏที่หน้าปัดของ Multimeter เป็นดังรูปที่ 5.12 ซึ่งเป็นค่าที่หน้าปัดของโปรแกรมด้าน Master ซึ่งจะเป็นค่าเท่ากันกับค่าที่ปรากฏที่ Multimeter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 แสดงหน้าปัดของโปรแกรมด้าน Master

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์

จากการทดสอบการทำงานของโปรแกรม พบว่าสามารถทำการควบคุมชุดเครื่องมือวัดมาตรฐาน IEEE-488 ผ่านระบบเครือข่ายอินเตอร์เน็ตได้จริง โดยค่าความผิดพลาดมีค่าน้อย แต่ยังมีข้อบกพร่องเรื่องความเร็วในการทำงานของระบบ ซึ่งจะต้องมีการหน่วง (Delay) ของข้อมูล โดยอุปกรณ์ที่มีการ Delay มากที่สุดคือ Programmable Multimeter ซึ่งค่าหน่วงเวลานี้ขึ้นอยู่กับความเร็วในการทำงานของอุปกรณ์,ความเร็วของการทำงานของคอมพิวเตอร์ในระบบ และอัตราความเร็วในการส่งข้อมูลของตัวกลาง (Media) ซึ่งจะมีค่าไม่เท่ากันในแต่ละ Network

สำหรับแนวทางในการพัฒนาโครงงานนี้ สามารถทำได้โดยการปรับปรุงการทำงานของโปรแกรม ให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งอุปกรณ์ที่ควรพัฒนาเพิ่มเข้ามาได้แก่ Oscilloscope หรือเครื่องมืออื่น ๆ ที่ใช้ในระบบการวัดและทดสอบทางอิเล็กทรอนิกส์ หรือการพัฒนาระบบ ให้สามารถติดต่อหรือ ควบคุม ได้โดยที่ ไม่ต้องใช้โปรแกรม ฐานข้อมูล เช่น การควบคุมผ่านทาง web site โดยอาศัยโปรแกรม web browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรม

```
{program user remote }
unit remoteV1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons,about,client_generator,client_powersupply;
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button4: TButton;
    BitBtn1: TBitBtn;
    procedure Button4Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var
  Form1: TForm1;

implementation
{$R *.DFM}
procedure TForm1.Button4Click(Sender: TObject);
begin
      close;
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
      AboutBox.ShowModal;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
      ClientGen.Show;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
      POWERSUPPLY.Show;
end;
procedure TForm1.Button3Click(Sender: TObject);
var   command:string;
begin
      command:='clientpmeter.exe';
    if winexec(@command[1],sw_shownormal) <32 then
      messagedlg ('fail to execute'+' clientpmeter.exe',mterror,[mbok],0);
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{program user function generator }
unit CLIENT_GENERATOR;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Buttons, ComCtrls, StdCtrls, Spin, ExtCtrls, Db, DBTables, Grids, DBGrids,
  DBCtrls, Mask, DBCGrids;
type
  TClientGen = class(TForm)
    Shape1: TShape;
    Shape10: TShape;
    Label1: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    panel1: TPanel;
    Panel2: TPanel;
    Shape2: TShape;
    Label8: TLabel;
    Shape3: TShape;
    Label9: TLabel;
    Label14: TLabel;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Shape7: TShape;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    Label2: TLabel;
    Panel3: TPanel;
    SpeedButton2: TSpeedButton;
    Edit1: TEdit;
    UpDown2: TUpDown;
    Label3: TLabel;
    Shape8: TShape;
    Panel4: TPanel;
    Edit2: TEdit;
    SpeedButton1: TSpeedButton;
    UpDown1: TUpDown;
    Shape9: TShape;
    Label4: TLabel;
    Panel5: TPanel;
    Button2: TButton;
    Button1: TButton;
    Panel6: TPanel;
    SpeedButton9: TSpeedButton;
    Label15: TLabel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Label16: TLabel;
  DataSource1: TDataSource;
  Table1: TTable;
  Table1name: TStringField;
  Table1value: TFloatField;
  Table1type: TStringField;


  procedure SpeedButton2Click(Sender: TObject);
  procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  procedure Edit2KeyPress(Sender: TObject; var Key: Char);
  procedure SpeedButton1Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
  procedure SpeedButton5Click(Sender: TObject);
  procedure SpeedButton6Click(Sender: TObject);
  procedure SpeedButton7Click(Sender: TObject);
  procedure SpeedButton8Click(Sender: TObject);
  procedure SpeedButton9Click(Sender: TObject);
 { Panel1: Tpanel; }
 private
  { Private declarations }
 public
  { Public declarations }
 end;
var
 ClientGen: TClientGen;
implementation
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{$R *.DFM}
procedure TClientGen.SpeedButton2Click(Sender: TObject);
var
    num:real;
    value:char;
    freq:string;
    code:integer;
    digit:integer;

begin
     shape8.Brush.color:=clred;
     shape9.Brush.color:=clmaroon;
     val(edit1.text,num,code);
     if (num<0.01) then value:= '1'
     else if (num>=0.01) and (num<1e3) then value:= '2'
          else if  (num>=1e3) and (num<1e6) then value:= '3'
               else if (num>=1e6) and (num<10e6) then value:= '4'
                    else if (num>=10e6) then value:= '5'
                         else value:='0';
    case value of
      '1': begin label15.caption:='Hz';
           num := 0.01;
           digit:=2;
          end;
      '2': begin label15.caption:='Hz';
           digit:=2;
          end;
      '3': begin label15.caption:='KHz';
           if (num>=1e3) and (num<10e3) then digit:=4
              else if (num>=10e3) and (num<100e3) then digit := 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else if (num>=100e3) and (num<1e6) then digit := 2
                  else digit :=2;
            num:= num/1e3;
            end;
        '4': begin label15.caption:='MHz';
              num:= num/1e6;
              digit := 4;
            end;


    '5': begin label15.caption:='MHz';
              num:= 10e6;
              num:= num/1e6;
              digit:=4;
            end;
      end;
      str(num:7:digit,freq);
      label6.caption:=freq;
      Table1.FindNearest(['freq']);
      Table1.Edit ;
      table1Value.AsVariant:=Edit1.Text;
      Table1.Post;
      Table1.FindNearest(['status']);
      Table1.Edit;
      table1value.AsVariant:= 0;
      Table1.Post;
  end;
  procedure TClientGen.SpeedButton1Click(Sender: TObject);
  var
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
num:real;
code:integer;
amplitude:string;
index:char;
begin
  shape9.Brush.color:=clred;
   shape8.Brush.color:=clmaroon;
   val(edit2.text,num,code);
   if (num<=0.2) then index:='1'
     else if (num>0.2) and (num<=20) then index:='2'
         else if (num>20) then index:='3'
             else index:='0';

   case index of
   '1' : begin label16.caption:='mV';
             num:= num*1000;
        end;
   '2' : label16.caption:='V';
   '3' : begin MessageBeep(0);
             num:= 20;
             label16.caption:='V';
             edit2.text:= IntToStr(20);
        end;
   end;
   str(num:6:2,amplitude);
   label7.caption:= amplitude;
   // Amplitude data
    Table1.FindNearest(['amp']);
    Table1.Edit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        table1Value.AsVariant:=Edit2.text;
        Table1.Post;
        Table1.FindNearest(['status']);
       Table1.Edit;
       table1value.AsVariant:= 0;
       Table1.Post;
end;
procedure TClientGen.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
 If ((key<'0') or (key>'9'))and (key <>'.')Then
      begin
          MessageBeep (0);
          key := #0;
      end;
end;
procedure TClientGen.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
 if ((key < '0') or (key >'9'))and (key <> '.') then
      begin
          MessageBeep(0);
          key:=#0;
      end;
end;
procedure TClientGen.Button1Click(Sender: TObject);
begin
     shape4.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shape7.Brush.color:=clmaroon;
    Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 1;
    Table1.Post;
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= 'sin';
    Table1.Post;
    Shape4.Brush.color := clred;
    Label15.Caption := 'KHz';
    Label16.Caption := 'V';
    Label6.Caption := '1.0000';
    Label7.Caption := '10.0';

     Table1.FindNearest(['freq']);
    Table1.Edit ;
    table1Value.AsVariant:=1000;
    Table1.Post;
    Table1.FindNearest(['amp']);
    Table1.Edit;
    table1Value.AsVariant:=10.0;
    Table1.Post;
end;
procedure TClientGen.Button2Click(Sender: TObject);
begin
   Close;
end;
procedure TClientGen.SpeedButton3Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Var
    signal:string;
begin
     shape4.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    signal:= 'sin';
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
     Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;
procedure TClientGen.SpeedButton4Click(Sender: TObject);
Var
    signal:string;
begin
     shape5.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    signal:= 'sqr';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
     Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;
procedure TClientGen.SpeedButton5Click(Sender: TObject);
Var
    signal:string;
begin
     shape3.Brush.color:=clred;
    shape2.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    signal:= 'tri';
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
     Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;
procedure TClientGen.SpeedButton6Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Var
    signal:string;
begin
     shape6.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    signal:= 'pls';
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
     Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;

procedure TClientGen.SpeedButton7Click(Sender: TObject);
Var
    signal:string;
begin
    shape2.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    signal:= 'rmp';
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
    Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;
procedure TClientGen.SpeedButton8Click(Sender: TObject);
Var
    signal:string;
begin
     shape7.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    signal:= 'arb';
    Table1.FindNearest(['signal']);
    Table1.Edit;
    table1type.AsVariant:= signal;
    Table1.Post;
     Table1.FindNearest(['status']);
    Table1.Edit;
    table1value.AsVariant:= 0;
    Table1.Post;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TClientGen.SpeedButton9Click(Sender: TObject);
begin
    if shape10.Brush.color = clmaroon then
      begin
        shape10.Brush.color:=clred;
        Table1.FindNearest(['output']);
        Table1.Edit;
        Table1value.AsVariant := 1;
        Table1.Post;
      end
    else
      begin
        shape10.brush.color:=clmaroon;
        Table1.FindNearest(['output']);
        Table1.Edit;
        Table1value.AsVariant :=0;
        Table1.Post;
      end;
    end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ program user power supply }
unit CLIENT_POWERSUPPLY;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,StdCtrls, ComCtrls, ExtCtrls, Spin, Buttons, Grids, DBGrids, Db,
  DBTables, Mask, DBCtrls, DBCGrids;
type
  TPOWERSUPPLY = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Panel6: TPanel;
    Panel7: TPanel;
    Panel8: TPanel;
    Panel9: TPanel;
    Panel11: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Shape1: TShape;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Label9: TLabel;
    Panel13: TPanel;
    Shape7: TShape;
    Label10: TLabel;
    Label11: TLabel;
    Panel12: TPanel;
    SpeedButton5: TSpeedButton;
    Panel10: TPanel;
    SpeedButton3: TSpeedButton;
    Panel14: TPanel;
    Button2: TButton;
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    UpDown1: TUpDown;
    UpDown2: TUpDown;
    DataSource1: TDataSource;
    Table1: TTable;
    Table1name: TStringField;
    Table1value: TFloatField;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  POWERSUPPLY: TPOWERSUPPLY;
implementation
{$R *.DFM}
procedure TPOWERSUPPLY.SpeedButton1Click(Sender: TObject);
var
  voltV1:string;
  num:real;
  code:integer;
begin
  shape3.Brush.color:=ClRed;
  if shape3.Brush.color= clred then
    shape4.Brush.color:= clmaroon;
    shape5.Brush.color:= clmaroon;
    shape6.Brush.color:= clmaroon;
  val(edit1.text,num,code);
    if num>30 then
       begin
         num:=30;
         Edit1.Text := IntTOstr(30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MessageBeep(0);
      end;
    str(num:5:2,voltV1);
    Label6.Caption:= voltV1;
    Label12.Caption := 'V';
    Table1.FindNearest(['V1']);
    Table1.Edit;
    Table1value.AsVariant := Edit1.Text;
    Table1.Post;
    Table1.FindNearest(['status']);
    Table1.Edit;
    Table1Value.AsVariant := 0;
    Table1.Post;
end;
procedure TPOWERSUPPLY.SpeedButton3Click(Sender: TObject);
var
   Num:real;
   code:integer;
   voltV2:string;
begin
    shape5.Brush.color :=clRed;
    if shape5.Brush.color = clRed then
      shape3.Brush.color := clmaroon;
      shape4.Brush.color := clmaroon;
      shape6.Brush.color := clmaroon;
    val(edit1.text,num,code);
    if num>30 then
      begin
        num:=30;
        Edit1.Text := IntTOstr(30);
        MessageBeep(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      end;
    str(num:5:2,voltV2);
    Label8.Caption:= voltV2;
    Label14.Caption := 'V';
    Table1.FindNearest(['V2']);
    Table1.Edit;
    Table1value.AsVariant := Edit1.Text;
    Table1.Post;
    Table1.FindNearest(['status']);
    Table1.Edit;
    Table1Value.AsVariant := 0;
    Table1.Post;
end;
procedure TPOWERSUPPLY.SpeedButton2Click(Sender: TObject);
var
  CurrentI1:String;
  num:real;
  code:integer;
begin
    shape4.Brush.color := clRed;
    if shape4.Brush.color = clRed then
      shape3.Brush.color := clmaroon;
      shape5.Brush.color := clmaroon;
      shape6.Brush.color := clmaroon;
  val(edit2.text,num,code);
  if num>1 then
      begin
        num:=1;
        Edit2.Text := IntTOstr(1);
        MessageBeep(0);
      end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    str(num:5:2,CurrentI1);
    Label7.Caption:= CurrentI1;
    Label13.Caption := 'A';
    Table1.FindNearest(['I1']);
    Table1.Edit;
    Table1value.AsVariant := Edit2.Text;
    Table1.Post;
    Table1.FindNearest(['status']);
     Table1.Edit;
     Table1Value.AsVariant := 0;
     Table1.Post;
end;
procedure TPOWERSUPPLY.SpeedButton4Click(Sender: TObject);
var
CurrentI2:string;
num:real;
code:integer;
begin
    shape6.Brush.Color := clRed;
    if shape6.Brush.color = clRed then
      shape3.Brush.color := clmaroon;
      shape4.Brush.color := clmaroon;
      shape5.Brush.color := clmaroon;
    val(edit2.text,num,code);
     if num>1 then
      begin
        num:=1;
        Edit2.Text := IntTOstr(1);
        MessageBeep(0);
      end;
    str(num:5:2,CurrentI2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label9.Caption:= CurrentI2;
    Label15.Caption := 'A';
    Table1.FindNearest(['I2']);
    Table1.Edit;
    Table1value.AsVariant := Edit2.Text;
    Table1.Post;
   Table1.FindNearest(['status']);
   Table1.Edit;
   Table1Value.AsVariant := 0;
   Table1.Post;
end;
procedure TPOWERSUPPLY.SpeedButton5Click(Sender:TObject);
  begin
   if Shape7.Brush.Color = clmaroon then
     begin
      shape7.Brush.Color:= clred ;
      Table1.FindNearest(['output']);
      Table1.Edit;
      Table1value.AsVariant := 1;
      Table1.Post;
     end
    else
     begin
      shape7.brush.color:= clmaroon;
      Table1.FindNearest(['output']);
      Table1.Edit;
      Table1value.AsVariant := 0;
      Table1.Post;
     end;
   end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TPOWERSUPPLY.Button2Click(Sender: TObject);
begin
   close;
end;
procedure TPOWERSUPPLY.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    If ((key<'0') or (key>'9'))and (key <>'.')Then
      begin
         MessageBeep (0);
         key := #0;
      end;
end;
procedure TPOWERSUPPLY.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    If ((key<'0') or (key>'9'))and (key <>'.')Then
      Begin    MessageBeep (0);
               key := #0;    end;
end;
procedure TPOWERSUPPLY.Button1Click(Sender: TObject);
begin
   Edit1.Text := IntTostr(0);
   Edit2.Text := IntToStr(0);
   SpeedButton1.Click;
   SpeedButton2.Click;
   SpeedButton3.Click;
   SpeedButton4.Click;
   Table1.FindNearest(['status']);
   Table1.Edit;
   Table1Value.AsVariant := 1;
   Table1.Post;
end;   end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ program user Multimeter}
unit clientmeterV4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, ComCtrls, Buttons, StdCtrls,multimeter1,multimeter2,multimeter3,
  multimeter4,multimeter5, Grids, DBGrids, Db, DBTables;
type
  TMULTIMETER = class(TForm)
    panel1: TPanel;
    Shape1: TShape;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Panel2: TPanel;
    Label4: TLabel;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    VDC: TSpeedButton;
    VAC: TSpeedButton;
    ADC: TSpeedButton;
    AAC: TSpeedButton;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    OHM: TSpeedButton;
    Panel3: TPanel;
    Label12: TLabel;
    Shape8: TShape;
    AUTO: TSpeedButton;
    Button1: TButton;
    DataSource1: TDataSource;
    Table1: TTable;
    Table1Information: TStringField;
    Table1Data: TStringField;
    Timer1: TTimer;
    procedure AUTOClick(Sender: TObject);
    procedure VDCClick(Sender: TObject);
    procedure VACClick(Sender: TObject);
    procedure ADCClick(Sender: TObject);
    procedure AACClick(Sender: TObject);
    procedure OHMClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
    MULTIMETER : TMULTIMETER;
  implementation
{$R *.DFM}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure sadang(index:integer);
begin
   case  index of
      1: form3.Showmodal; //VDC
      2: form2.Showmodal; //VAC
      3: form4.Showmodal; //OHM
      4: form5.Showmodal; //ADC
      5: form6.Showmodal;  //AAC
   end;
 end;
procedure autorange(sett:integer);
var  buf:packed array[0..32]of char;
    msg:string;
begin
     case sett of
     1:begin
          buf:='A1;T3;VD;S1;';          //VDC
          msg:=buf;
          MULTIMETER.Table1.FindNearest(['range']);
          MULTIMETER.Table1.Edit;
          MULTIMETER.table1Data.AsVariant:=msg;
          MULTIMETER.Table1.Post;
          MULTIMETER.label3.caption:='V';
             end ;
     2:begin
          buf:='A1;T3;VA;S1;';          // VAC
          msg:=buf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      MULTIMETER.Table1.FindNearest(['range']);
      MULTIMETER.Table1.Edit;
      MULTIMETER.table1Data.AsVariant:=msg;
      MULTIMETER.Table1.Post;
      MULTIMETER.Label3.caption:='V';
        end;
3:begin
      buf:='A1;T3;ID;S1;';        //Idc
      msg:=buf;
      MULTIMETER.Table1.FindNearest(['range']);
      MULTIMETER.Table1.Edit;
      MULTIMETER.table1Data.AsVariant:=msg;
      MULTIMETER.Table1.Post;
      MULTIMETER.label3.caption:='mA';
        end;
4:begin
      buf:='A1;T3;IA;S1;';     // Iac
      msg:=buf;
      MULTIMETER.Table1.FindNearest(['range']);
      MULTIMETER.Table1.Edit;
      MULTIMETER.table1Data.AsVariant:=msg;
      MULTIMETER.Table1.Post;
      MULTIMETER.label3.caption:='mA';
        end ;
5:begin
      buf:='A1;T3;O2;S1;';    //OHM
      msg:=buf;
      MULTIMETER.Table1.FindNearest(['range']);
      MULTIMETER.Table1.Edit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MULTIMETER.table1Data.AsVariant:=msg;
        MULTIMETER.Table1.Post;
        MULTIMETER.label3.caption:='Kohm';
          end ;
    end;
end;
procedure TMULTIMETER.AUTOClick(Sender: TObject);
begin
  if shape8.brush.color =clmaroon then
    begin
        shape8.Brush.color:=clred;
     if  shape2.Brush.color=clred then      //VDC
         autorange(1);
     if  shape3.Brush.color=clred then      //VAC
         autorange(2);
     if  shape4.Brush.color=clred then      //ADC
         autorange(3);
     if  shape5.Brush.color=clred then      //AAC
         autorange(4);
     if  shape6.Brush.color=clred then      //OHM
         autorange(5);
    end
    else
      begin shape8.Brush.Color:= clmaroon;
      end;
end;
procedure TMULTIMETER.VDCClick(Sender: TObject);
var  buf :packed array [0..32] of char;
   msg :string; i:integer;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shape2.brush.color:=clred;
    shape3.brush.color:=clmaroon;
    shape4.brush.color:=clmaroon;
    shape5.brush.color:=clmaroon;
    shape6.brush.color:=clmaroon;
      if shape8.brush.color <> clred  then
      begin
          buf:='A0;T3;VD;S1;';
          sadang(1);
      for i:=1 to length(form3.Label1.Caption) do
       buf[i+11]:=form3.Label1.Caption[i];
       msg:=buf;
       MULTIMETER.Table1.FindNearest(['range']);
       MULTIMETER.Table1.Edit;
       MULTIMETER.table1Data.AsVariant:=msg;
       MULTIMETER.Table1.Post;
       label3.caption:='V';
      end
    else
      begin
       autorange(1);
      end ;
end;
procedure TMULTIMETER.VACClick(Sender: TObject);
var  buf:packed array[0..32]of char;
    i:integer; msg:string;
begin
     shape2.Brush.color:=clmaroon;
     shape3.brush.color:=clred;
     shape4.brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        shape5.brush.color:=clmaroon;
        shape6.brush.color:=clmaroon;
       if shape8.brush.color <> clred  then
        begin
          sadang(2);
          buf:='A0;T3;VA;S1;';
       for i:=1 to length(form2.Label1.Caption) do
          buf[i+11]:=form2.Label1.Caption[i];
          msg:=buf;
          MULTIMETER.Table1.FindNearest(['range']);
          MULTIMETER.Table1.Edit;
          MULTIMETER.table1Data.AsVariant:=msg;
          MULTIMETER.Table1.Post;
          label3.caption:='V';
        end
      else
          begin
          autorange(2);
          end;
end;
procedure TMULTIMETER.ADCClick(Sender: TObject);
var  buf:packed array[0..32]of char;
    i:integer; msg:string;
begin
      shape2.brush.color:=clmaroon;
      shape3.brush.color:=clmaroon;
      shape4.brush.color:=clred;
      shape5.brush.color:=clmaroon;
      shape6.brush.color:=clmaroon;
     if shape8.brush.color <> clred  then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      begin
        sadang(4);
        buf:='A0;T3;ID;S1;';
      for i:=1 to length(form5.Label1.Caption) do
        buf[i+11]:=form5.Label1.Caption[i];
        msg:=buf;
        MULTIMETER.Table1.FindNearest(['range']);
        MULTIMETER.Table1.Edit;
        MULTIMETER.table1Data.AsVariant:=msg;
        MULTIMETER.Table1.Post;
        label3.Caption:='mA';
      end
       else
        begin
          autorange(3);
        end;
end;
procedure TMULTIMETER.AACClick(Sender: TObject);
var  buf:packed array[0..32]of char;
    i:integer; msg:string;
begin
     shape2.brush.color:=clmaroon;
     shape3.brush.color:=clmaroon;
     shape4.brush.color:=clmaroon;
     shape5.brush.color:=clred;
     shape6.brush.color:=clmaroon;
    if shape8.brush.color <> clred  then
    begin
        sadang(5);
        buf:='A0;T3;IA;S1;';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      for i:=1 to length(form6.Label1.Caption) do
         buf[i+11]:=form6.Label1.Caption[i];
         msg:=buf;
         MULTIMETER.Table1.FindNearest(['range']);
         MULTIMETER.Table1.Edit;
         MULTIMETER.table1Data.AsVariant:=msg;
         MULTIMETER.Table1.Post;
         label3.caption:='mA';
      end
       else
        begin
        autorange(4);
       end;
end;
procedure TMULTIMETER.OHMClick(Sender: TObject);
var  buf:packed array[0..32]of char;
     i:integer; msg:string;
begin
     shape2.brush.color:=clmaroon;
     shape3.brush.color:=clmaroon;
     shape4.brush.color:=clmaroon;
     shape5.brush.color:=clmaroon;
     shape6.brush.color:=clred;
    if shape8.brush.color <> clred  then
      begin
         sadang(3);
         buf:='A0;T3;O2;S1;';
       for i:=1 to length(form4.Label1.Caption) do
         buf[i+11]:=form4.Label1.Caption[i];
         msg:=buf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MULTIMETER.Table1.FindNearest(['range']);
        MULTIMETER.Table1.Edit;
        MULTIMETER.table1Data.AsVariant:=msg;
        MULTIMETER.Table1.Post;
        label3.caption:='Kohm';
      end
    else
      begin
      autorange(5);
      end;
end;


procedure TMULTIMETER.FormClose(Sender: TObject; var Action: TCloseAction);
begin
       ACTION:=CAFREE;
end;
procedure TMULTIMETER.Button1Click(Sender: TObject);
begin
      Close;
end;
procedure TMULTIMETER.Timer1Timer(Sender: TObject);
begin
    table1.Refresh;
    table1.FindNearest(['read']);
    label2.caption:= table1Data.AsString;
end;
procedure TMULTIMETER.FormShow(Sender: TObject);
begin
    timer1.Enabled := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shape2.Brush.color:=clred;
    AUTO.Click;
end;
end.
{ program range vac }
unit multimeter1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm2 = class(TForm)
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    ok: TButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    RadioButton5: TRadioButton;
    Label1: TLabel;
    procedure RadioButton1Click(Sender: TObject);
    procedure okClick(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 end;
var
 Form2: TForm2;          // VAC
 buff:packed array [0..3] of char;
 implementation
{$R *.DFM}
procedure TForm2.RadioButton1Click(Sender: TObject);
begin
   buff:='R1;';
end;
procedure TForm2.okClick(Sender: TObject);
begin
   Label1.caption:=buff;
   close;
end;


procedure TForm2.RadioButton2Click(Sender: TObject);
begin
     buff:='R2;';
end;
procedure TForm2.RadioButton3Click(Sender: TObject);
begin
 buff:='R3;';
end;
procedure TForm2.RadioButton4Click(Sender: TObject);
begin
    buff:='R4;';
end;
procedure TForm2.RadioButton5Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    buff:='R5;';
end;
end.
{ program range vdc }
unit multimeter2;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm3 = class(TForm)
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    RadioButton5: TRadioButton;
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 end;
var
 Form3: TForm3; //  VDC
  buff:packed array [0..3] of char;
implementation
{$R *.DFM}
procedure TForm3.Button1Click(Sender: TObject);
begin
   label1.caption:=buff;
   close;
end;
procedure TForm3.RadioButton1Click(Sender: TObject);
begin
   buff:='R1;';
end;
procedure TForm3.RadioButton2Click(Sender: TObject);
begin
 buff:='R2;';
end;
procedure TForm3.RadioButton3Click(Sender: TObject);
begin
  buff:='R3;';
end;
procedure TForm3.RadioButton4Click(Sender: TObject);
begin
 buff:='R4;';
end;
procedure TForm3.RadioButton5Click(Sender: TObject);
begin
  buff:='R5;';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
end.
{ program range  ohm}
unit multimeter3;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm4 = class(TForm)
    RadioGroup1: TRadioGroup;
    Button1: TButton;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    RadioButton5: TRadioButton;
    RadioButton6: TRadioButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton5Click(Sender: TObject);
    procedure RadioButton6Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 end;
var
 Form4: TForm4;   //  OHM
 buff:packed array [0..3] of char;
implementation
{$R *.DFM}
procedure TForm4.Button1Click(Sender: TObject);
begin
   label1.caption:=buff;
   close;
end;
procedure TForm4.RadioButton1Click(Sender: TObject);
begin
  buff:='R1;';
end;
procedure TForm4.RadioButton2Click(Sender: TObject);
begin
  buff:='R2;';
end;
procedure TForm4.RadioButton3Click(Sender: TObject);
begin
 buff:='R3;';
end;


procedure TForm4.RadioButton4Click(Sender: TObject);
begin
  buff:='R4;';
end;
procedure TForm4.RadioButton5Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  buff:='R5;';
end;
procedure TForm4.RadioButton6Click(Sender: TObject);
begin
    buff:='R6;';
end;
end.
{ program  range adc }
unit multimeter4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm5 = class(TForm)
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
 var
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Form5: TForm5;
   buff:packed array [0..3] of char;
implementation
{$R *.DFM}
procedure TForm5.Button1Click(Sender: TObject);
begin
 label1.caption:=buff;
   close;
end;
procedure TForm5.RadioButton1Click(Sender: TObject);
begin
  buff:='R2;';
end;
procedure TForm5.RadioButton2Click(Sender: TObject);
begin
  buff:='R5;';
end;
end.
{ program range aac }
unit multimeter5;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm6 = class(TForm)
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Button1: TButton;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form6: TForm6;        // AAC
 buff:packed array [0..3] of char;
implementation
{$R *.DFM}
procedure TForm6.Button1Click(Sender: TObject);
begin
     label1.caption:=buff;
    close;
end;
procedure TForm6.RadioButton1Click(Sender: TObject);
begin
    buff:='R2;';
end;
procedure TForm6.RadioButton2Click(Sender: TObject);
begin
  buff:='R5;';
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ program  data monitor  at sql  server }
unit data;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Grids, DBGrids, Db, DBTables, ExtCtrls;
type
  TForm1 = class(TForm)
    Table1: TTable;
    Table2: TTable;
    Table3: TTable;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    DataSource3: TDataSource;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    DBGrid3: TDBGrid;
    Panel1: TPanel;
    Label2: TLabel;
    Label1: TLabel;
    Label7: TLabel;
    Panel2: TPanel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Panel3: TPanel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer1: TTimer;
    Panel4: TPanel;
    Label6: TLabel;
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Label6.Caption := TimeToStr(time);
  Table1.Refresh;
  Table2.Refresh;
  Table3.Refresh;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
  Timer1.Enabled := True;
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    { program master remote }
Unit  remoteV3;
interface
uses
  Windows, Messagés, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons,server_supply,about,server_generator2;
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button4: TButton;
    BitBtn1: TBitBtn;
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  public
   { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Button2Click(Sender: TObject);
begin
     powersupply.Show;
end;
procedure TForm1.Button3Click(Sender: TObject);
var   command:string;
begin
      command:='serverpmeterv9.exe';
      if winexec(@command[1],sw_shownormal)<32 then
        messagedlg('failed to execute'+ command,mterror,  [mbok],0);
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
       close;
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
       aboutbox.Showmodal;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
       generator.show;
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{program master programable function generator }
unit SERVER_GENERATOR2;
interface
uses
   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
   Dialogs,Buttons, ComCtrls, StdCtrls, Spin, ExtCtrls, Db, DBTables,
  Grids, DBGrids,Mask, DBCtrls, DBCGrids;
type
   TGENERATOR = class(TForm)
   Shape1: TShape;
   Shape10: TShape;
   Label1: TLabel;
   Label5: TLabel;
   Label6: TLabel;
   Label7: TLabel;
   panel1: TPanel;
   Panel2: TPanel;
   Shape2: TShape;
   Label8: TLabel;
   Shape3: TShape;
   Label9: TLabel;
   Label14: TLabel;
   Shape4: TShape;
   Shape5: TShape;
   Shape6: TShape;
   Shape7: TShape;
   Label10: TLabel;
   Label11: TLabel;
   Label12: TLabel;
   Label13: TLabel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton7: TSpeedButton;

SpeedButton8: TSpeedButton;

Label2: TLabel;

Panel3: TPanel;

SpeedButton2: TSpeedButton;

Edit1: TEdit;

UpDown2: TUpDown;

Label3: TLabel;

Shape8: TShape;

Panel4: TPanel;

Edit2: TEdit;

SpeedButton1: TSpeedButton;

UpDown1: TUpDown;

Shape9: TShape;

Label4: TLabel;

Panel5: TPanel;

Button2: TButton;

Label15: TLabel;

Label16: TLabel;

DataSource1: TDataSource;

Table1: TTable;

Timer1: TTimer;

Table1name: TStringField;

Table1value: TFloatField;

Table1type: TStringField;

SpeedButton3: TSpeedButton;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure SpeedButton2Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
   { Panel1: Tpanel; }
   private
    { Private declarations }
   public
    { Public declarations }
   end;
type
 TReceive = procedure (boardID:    integer;
                       address:    word;
                       var buffer;
                       cnt:        longint;
                       termination: integer); stdcall;
  TSend = procedure (boardID: integer;
                     address: word;
                     var buffer;
                     datacnt: longint;
                     eotmode: integer); stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 TSendIFC = procedure (boardID: integer); stdcall;
 Tibclr   = function (ud:integer):integer;stdcall;
 Tibonl   = function(ud:integer;v:integer):integer;stdcall;
 Tibwrt   = function(ud:integer;var wrtbuf;
              cnt:longint):integer;stdcall;
 Tibdev   = function(ud:integer;
              pad:integer;
              sad:integer;
              tmo:integer;
              eot:integer;
              eos:integer):integer;stdcall;
 Tibrd    = function(ud:integer;
              var rdbuf;
              cnt:Longint):integer;stdcall;
var
 GENERATOR: TGENERATOR;
 Gpib32lib:Thandle;
 Receive: TReceive;
 Send: TSend;
 SendIFC: TSendIFC;
 ibclr:Tibclr;
 ibwrt:Tibwrt;
 ibonl:Tibonl;
 ibdev:Tibdev;
 ibrd:Tibrd;
 dvm:integer;
 num:integer;
 buf:packed array[0..100] of char;
implementation
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{$R *.DFM}
procedure loadDLL;
var
 str:string;
 begin
 Gpib32Lib:=LoadLibrary('GPIB-32.DLL');
 If Gpib32Lib = 0 then
    begin
    str:='LoadLibary FAILED!';
    MessageDlg(str,mtError,[mbOK],0);
    halt;
    end;
 @Receive      := GetProcAddress(Gpib32Lib, 'Receive');
 @Send         := GetProcAddress(Gpib32Lib, 'Send');
 @SendIFC      := GetProcAddress(Gpib32Lib, 'SendIFC');
 @ibclr       :=getProcAddress(Gpib32Lib,'ibclr');
 @ibonl       :=GetProcAddress(Gpib32Lib,'ibonl');
 @ibwrt       :=GetProcAddress(Gpib32Lib,'ibwrt');
 @ibdev       :=GetProcAddress(Gpib32Lib,'ibdev');
 @ibrd        :=GetProcAddress(Gpib32Lib,'ibrd');
 If (@Receive      = NIL) or
   (@Send         = NIL) or
   (@SendIFC      = NIL) or
   (@ibclr       = NIL) or
   (@ibonl       = NIL) or
   (@ibdev       = NIL) or
   (@ibrd        = NIL) or
   (@ibwrt       = NIL) Then
    begin
    str:='GetProcAddress FAIL!';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      MessageDlg(str,mtError,[mbOK],0);
      FreeLibrary(Gpib32Lib);
     end;
  end;
procedure ReadFreq;
var   str:string;
begin
    dvm := ibdev(0,12,0,15,1,0);
    buf := 'frq? ';
    send(0,12,buf,20,1);
    receive(0,12,buf,20,1);
    case  buf[13] of
        '0' : GENERATOR.Label15.Caption :='Hz';
        '3' :  if buf[12]='-' then
                 begin
                  GENERATOR.Label15.caption :='mHz';
                    end
               else
                    begin
                  GENERATOR.Label15.caption :='KHz';
                    end;
        '6' : GENERATOR.Label15.caption :='MHz';
     end;
    str := buf;
    delete(str,1,4);
    delete(str,8,4);
    GENERATOR.label6.caption:= str;
end;
procedure ReadAmp;
var  str:string;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    dvm := ibdev(0,12,0,15,1,0);
    buf:='amp? ';
    send(0,12,buf,20,1);
    receive(0,12,buf,20,1);
     case buf[11] of
        '0' : GENERATOR.label16.caption := 'V';
        '3' : GENERATOR.label16.caption := 'mV';
     end;
    str := buf;
     delete(str,1,4);
     delete(str,6,4);
     GENERATOR.label7.caption:= str;
end;
Procedure CheckSignal;
var index:char;
      SignalType:string;
begin
  GENERATOR.Table1.FindNearest(['signal']);
  SignalType:= GENERATOR.Table1type.asString;
 If SignalType = 'rmp' then index := '1'
    else if SignalType = 'tri' then index := '2'
    else if SignalType = 'sin' then index := '3'
     else if SignalType = 'sqr' then index := '4'
     else if SignalType = 'pls' then index := '5'
    else if SignalType = 'arb' then index := '6'
     else index:= '0';
  case index of
     '1': GENERATOR.SpeedButton7.Click;
     '2': GENERATOR.SpeedButton5.Click;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      '3': GENERATOR.SpeedButton3.Click;
      '4': GENERATOR.SpeedButton4.Click;
      '5': GENERATOR.SpeedButton6.Click;
      '6': GENERATOR.SpeedButton8.Click;
  end;
end;
procedure CheckOutput;
var  Output:string;
begin
   dvm:=ibdev(0,12,0,15,1,0);
   GENERATOR.Table1.FindNearest(['output']);
   Output := GENERATOR.Table1value.asString;
   if Output = '1' then
      begin
         GENERATOR.shape10.Brush.color:=clred;
         buf:='ot1 ';
         ibwrt(dvm,buf,4);
      end
   else  if Output = '0' then
      begin
         GENERATOR.shape10.brush.color:=clmaroon;
         buf:='ot0 ';
         ibwrt(dvm,buf,4);
     end;
end;
Procedure CheckStatus;
Var  Status:string;
begin
   dvm:=ibdev(0,12,0,15,1,0);
   GENERATOR.Table1.FindNearest(['status']);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Status := GENERATOR.Table1value.asString;
    if Status = '1' then
       begin
         ibclr(dvm);
         GENERATOR.shape4.brush.color := clred;
       end;
end;
procedure TGENERATOR.Timer1Timer(Sender: TObject);
begin
    Table1.Refresh;
    Table1.FindNearest(['freq']);
    edit1.Text := table1value.asString;
    Speedbutton2.Click;
    Table1.FindNearest(['amp']);
    edit2.text := table1value.asString;
    Speedbutton1.Click;
    CheckSignal;
    CheckOutput;
    CheckStatus;
end;
procedure TGENERATOR.SpeedButton2Click(Sender: TObject);
var  freq:string[15];
      i: integer;
begin
   if shape8.brush.color = clmaroon then
       begin
          shape8.Brush.color:=clred;
          shape9.Brush.color:=clmaroon;
        end;
   Table1.FindNearest(['freq']);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
edit1.text := table1value.asString;
dvm := ibdev(0,12,0,15,1,0);
if edit1.Text <> '' then
  begin
    freq := edit1.Text;
    buf := 'frq:';
    for i := 1 to Length(freq) do
    buf[3 + i ] :=freq[i];
    buf[4 + Length(freq)] := ';';
    send(0,12,buf,20,1);
    buf :=  '';
    ReadFreq;
    ReadAmp;
  end;
end;
procedure TGENERATOR.SpeedButton1Click(Sender: TObject);
var
  amp:string[15];
  i: integer;
begin
    if shape9.brush.color = clmaroon then
     begin
       shape9.Brush.color:=clred;
       shape8.Brush.color:=clmaroon;
     end;
    Table1.FindNearest(['amp']);
    edit2.text := table1value.asString;
    dvm := ibdev(0,12,0,15,1,0);
    amp := edit2.Text;
    buf := 'amp:';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for i := 1 to Length(amp) do
         buf[3 + i ] :=amp[i];
         buf[4 + Length(amp)] := ';';
         send(0,12,buf,20,1);
         buf := ' ';
      ReadFreq;
      ReadAmp;
end;
procedure TGENERATOR.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
 If ((key<'0') or (key>'9'))and (key <>'.')Then
        begin
          MessageBeep (0);
          key := #0;
        end;
end;
procedure TGENERATOR.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
 if ((key < '0') or (key >'9'))and (key <> '.') then
        begin
          MessageBeep(0);
          key:=#0;
       end;
end;
procedure TGENERATOR.Button2Click(Sender: TObject);
begin
   ibonl(0,0);
   FreeLibrary(Gpib32Lib);
   Close;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TGENERATOR.SpeedButton3Click(Sender: TObject);
begin
     shape4.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    dvm:=ibdev(0,12,0,15,1,0);
    buf:= 'sin ';
    ibwrt(dvm,buf,4);
end;
procedure TGENERATOR.SpeedButton4Click(Sender: TObject);
begin
     shape5.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    dvm:=ibdev(0,12,0,15,1,0);
    buf:= 'sqr ';
    ibwrt(dvm,buf,4);
end;
procedure TGENERATOR.SpeedButton5Click(Sender: TObject);
begin
     shape3.Brush.color:=clred;
    shape2.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    dvm:=ibdev(0,12,0,15,1,0);
    buf:= 'tri ';
    ibwrt(dvm,buf,4);
end;
procedure TGENERATOR.SpeedButton6Click(Sender: TObject);
begin
     shape6.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
    dvm:=ibdev(0,12,0,15,1,0);
    buf:= 'pls ';
    ibwrt(dvm,buf,4);
end;
procedure TGENERATOR.SpeedButton7Click(Sender: TObject);
begin
    shape2.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape7.Brush.color:=clmaroon;
 dvm:=ibdev(0,12,0,15,1,0);
   buf:= 'rmp ';
   ibwrt(dvm,buf,4);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TGENERATOR.SpeedButton8Click(Sender: TObject);
begin
     shape7.Brush.color:=clred;
    shape3.Brush.color:=clmaroon;
    shape4.Brush.color:=clmaroon;
    shape5.Brush.color:=clmaroon;
    shape6.Brush.color:=clmaroon;
    shape2.Brush.color:=clmaroon;
  dvm:=ibdev(0,12,0,15,1,0);
  buf:= 'arb ';
  ibwrt(dvm,buf,4);
end;
procedure TGENERATOR.FormShow(Sender: TObject);
begin
  Timer1.Enabled := True;
  LoadDLL;
  sendifc(0);
end;
end.
 { program master power supply }
unit SERVER_SUPPLY;
interface
uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
 Dialogs,StdCtrls, ComCtrls, ExtCtrls, Spin, Buttons, Grids, DBGrids, Db,
 DBTables;
type
 TPOWERSUPPLY = class(TForm)
   Panel1: TPanel;
   Panel2: TPanel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Panel3: TPanel;

Panel4: TPanel;

Panel5: TPanel;

Panel6: TPanel;

Panel7: TPanel;

Panel8: TPanel;

Panel9: TPanel;

Panel11: TPanel;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton4: TSpeedButton;

Shape1: TShape;

Shape2: TShape;

Shape3: TShape;

Shape4: TShape;

Shape5: TShape;

Shape6: TShape;

Label9: TLabel;

Panel13: TPanel;

Shape7: TShape;

Label10: TLabel;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label11: TLabel;
    Panel10: TPanel;
    SpeedButton3: TSpeedButton;
    Panel14: TPanel;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    UpDown1: TUpDown;
    UpDown2: TUpDown;
    DataSource1: TDataSource;
    Table1: TTable;
    Timer1: TTimer;
    Table1name: TStringField;
    Table1value: TFloatField;
    procedure Button2Click(Sender:TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
type
  Tibclr   = function (ud:integer):integer;stdcall;
  Tibonl   = function(ud:integer;v:integer):integer;stdcall;
  Tibwrt   = function(ud:integer;var wrtbuf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cnt:longint):integer;stdcall;
  Tibdev    = function(ud:integer;
                pad:integer;
                sad:integer;
                tmo:integer;
                eot:integer;
                eos:integer):integer;stdcall;
  Tibrd     = function(ud:integer;
                var rdbuf;
                cnt:Longint):integer;stdcall;
var
  POWERSUPPLY: TPOWERSUPPLY;
  Gpib32lib:Thandle;
  ibclr:Tibclr;
  ibwrt:Tibwrt;
  ibonl:Tibonl;
  ibdev:Tibdev;
  ibrd:Tibrd;
  dvm:integer;
implementation
{$R *.DFM}
procedure loadDLL;
VAR
  str:string;
  begin
  Gpib32Lib:=LoadLibrary('GPIB-32.DLL');
  If Gpib32Lib = 0 then
     begin
     str:='LoadLibary FAILED!';
     MessageDlg(str,mtError,[mbOK],0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    halt;
    end;
@ibclr      :=getProcAddress(Gpib32Lib,'ibclr');
@ibonl      :=GetProcAddress(Gpib32Lib,'ibonl');
@ibwrt      :=GetProcAddress(Gpib32Lib,'ibwrt');
@ibdev      :=GetProcAddress(Gpib32Lib,'ibdev');
@ibrd       :=GetProcAddress(Gpib32Lib,'ibrd');
If (@ibclr      = NIL) or
   (@ibonl      = NIL) or
   (@ibdev      = NIL) or
   (@ibrd       = NIL) or
   (@ibwrt      = NIL) Then
    begin
     str:='GetProcAddress FAIL!';
     MessageDlg(str,mtError,[mbOK],0);
     FreeLibrary(Gpib32Lib);
    end;
 end;
procedure TPOWERSUPPLY.Button2Click(Sender:TObject);
begin
    if Gpib32Lib<>0 then begin
    FreeLibrary(Gpib32Lib);
     end;
   Close;
end;
procedure ReadV1;
   var
   buffer : string[40];
   buf    : array[0..100] of char;
   msg    : string;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin
     dvm:=ibdev(0,8,0,15,1,0);
     buf := 'mu1 ' ;
     ibwrt(dvm,buf,8);
     ibrd(dvm,buf,20);
     buffer := buf;
     msg := buffer;
     POWERSUPPLY.label6.caption := msg;
    end;
procedure ReadV2;
  Var
    buffer : string[40];
    buf    : array[0..100] of char;
   begin
    dvm:=ibdev(0,8,0,15,1,0);
    buf := 'mu2 ' ;
    ibwrt(dvm,buf,8);
    ibrd(dvm,buf,20);
    buffer := buf;
    POWERSUPPLY.label8.caption := buffer;
   end;
procedure ReadI1;
  Var
    buffer : string[40];
    buf    : array[0..100] of char;
   begin
    dvm:=ibdev(0,8,0,15,1,0);
    buf := 'mi1 ' ;
    ibwrt(dvm,buf,8);
    ibrd(dvm,buf,20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    buffer := buf;
    POWERSUPPLY.label7.caption := buffer;
  end;
procedure Readl2;
  Var
    buffer : string[40];
    buf    : array[0..100] of char;
  begin
    dvm:=ibdev(0,8,0,15,1,0);
    buf := 'mi2 ' ;
    ibwrt(dvm,buf,8);
    ibrd(dvm,buf,20);
    buffer := buf;
    POWERSUPPLY.label9.caption := buffer;
  end;
procedure CheckOutput;
  var
    buf : array[0..100] of char;
    i:integer;
    Output:String;
  begin
    POWERSUPPLY.Table1.FindNearest(['output']);
    Output := POWERSUPPLY.Table1Value.AsString;
    if Output = IntToStr(1) then
      begin
      POWERSUPPLY.shape7.Brush.Color:= clred ;
      dvm := ibdev(0,8,0,15,1,0);
      buf := 'op1;';
      ibwrt(dvm,buf,5);
      ibrd(dvm,buf,35);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          for i := 1 to length(buf) do
          buf[i]:=' ';
          ibwrt(dvm,buf,5);
          ReadV1;
          ReadV2;
          ReadI1;
          ReadI2;
          end
        else  if Output = IntToStr(0) then
          begin
          POWERSUPPLY.shape7.brush.color:= clmaroon;
          buf:='op0 ';
          ibwrt(dvm,buf,5);
           for i := 1 to length(buf) do
          buf[i]:=' ';
          ibwrt(dvm,buf,5);
          ReadV1;
          ReadV2;
          ReadI1;
          ReadI2;
        end;
      end;
procedure CheckStatus;
Var
  Clear:String;
   begin
    POWERSUPPLY.Table1.FindNearest(['status']);
    Clear:= POWERSUPPLY.Table1Value.AsString;
     If Clear = IntToStr(1) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Begin
    dvm:=ibdev(0,8,0,15,1,0);
    ibclr(dvm);
    ReadV1;
    ReadV2;
    ReadI1;
    ReadI2;
     end;
   end;
procedure TPOWERSUPPLY.SpeedButton1Click(Sender: TObject);
var
    buf    :array[0..100] of char;
    V1     :string;
    i   :integer;
begin
 shape3.Brush.color:=ClRed;
 if shape3.Brush.color= clred then
    shape4.Brush.color:= clmaroon;
    shape5.Brush.color:= clmaroon;
    shape6.Brush.color:= clmaroon;
    dvm:=ibdev(0,8,0,15,1,0);
  Table1.FindNearest(['V1']);
  Edit1.Text:= Table1Value.AsString;
    if edit1.Text <> '' then
   begin
     V1 := edit1.text;
     buf := 'su1 ';
      for i := 1 to Length(V1) do
       buf[3 + i ] :=V1[i];
       buf[4 + Length(V1)] := ';';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ibwrt(dvm,buf,12);
             for i := 1 to length(buf) do
            buf[i] := ' ';
            ibwrt(dvm,buf,8);
            ReadV1;
             end;
 end;
procedure TPOWERSUPPLY.SpeedButton3Click(Sender: TObject);
var
  V2     : string;
  buf    : array[0..100] of char;
  I      :integer;
begin
   shape5.Brush.color :=clRed;
   if shape5.Brush.color = clRed then
     shape3.Brush.color := clmaroon;
     shape4.Brush.color := clmaroon;
     shape6.Brush.color := clmaroon;
   dvm:=ibdev(0,8,0,15,1,0);
   Table1.FindNearest(['V2']);
   Edit1.Text:= Table1Value.AsString;
    if edit1.Text <> '' then
    begin
     V2 := edit1.text;
     buf := 'su2 ';
      for i := 1 to Length(V2) do
       buf[3 + i ] :=V2[i];
       buf[4 + Length(V2)] := ';';
       ibwrt(dvm,buf,12);
        for i := 1 to length(buf) do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            buf[i] := ' ';
            ibwrt(dvm,buf,8);
        ReadV2;
         end;
end;
procedure TPOWERSUPPLY.SpeedButton2Click(Sender: TObject);
var
  I1     : string;
  buf    : array[0..100] of char;
  i  : integer;
begin
    shape4.Brush.color := clRed;
     if shape4.Brush.color = clRed then
      shape3.Brush.color := clmaroon;
      shape5.Brush.color := clmaroon;
      shape6.Brush.color := clmaroon;
  dvm:=ibdev(0,8,0,15,1,0);
  Table1.FindNearest(['I1']);
  Edit2.Text:= Table1Value.AsString;
    if edit2.Text <> '' then
    begin
      I1 := edit2.text;
      buf := 'SI1 ';
       for i := 1 to Length(I1) do
        buf[3 + i ] :=I1[i];
        buf[4 + Length(I1)] := ';';
        ibwrt(dvm,buf,12);
        for i := 1 to length(buf) do
        buf[i] := ' ';
        ibwrt(dvm,buf,8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ReadI1;
    end;
end;
procedure TPOWERSUPPLY.SpeedButton4Click(Sender: TObject);
var
  I2 : string;
  buf    : array[0..100] of char;
 i    : integer;
begin
    shape6.Brush.Color :=.clRed;
     if shape6.Brush.color = clRed then
      shape3.Brush.color := clmaroon;
      shape4.Brush.color := clmaroon;
      shape5.Brush.color := clmaroon;
    Table1.FindNearest(['I2']);
    dvm:=ibdev(0,8,0,15,1,0);
     if edit2.Text <> '' then
     begin
      I2 := edit2.text;
      buf := 'SI2 ';
       for i := 1 to Length(I2) do
        buf[3 + i ] :=I2[i];
        buf[4 + Length(I2)] := ';';
        ibwrt(dvm,buf,12);
          for i := 1 to length(buf) do
        buf[i] := ' ';
        ibwrt(dvm,buf,8);
        ReadI2;
     end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TPOWERSUPPLY.Timer1Timer(Sender: TObject);
begin
   Table1.Refresh;
   SpeedButton1.Click;
   SpeedButton2.Click;
   SpeedButton3.Click;
   SpeedButton4.Click;
   CheckOutput;
   CheckStatus;
end;
procedure TPOWERSUPPLY.FormShow(Sender: TObject);
begin
  Timer1.Enabled := True;
  loaddll;
end;
end.
{program master programable multimeter}
unit serverMeterV9;
interface
uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 ExtCtrls, ComCtrls, Buttons, StdCtrls,Db, DBTables, Grids, DBGrids;
type
 TMULTIMETER = class(TForm)
   panel1: TPanel;
   Shape1: TShape;
   Label1: TLabel;
   Label2: TLabel;
   Label3: TLabel;
   Panel2: TPanel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label4: TLabel;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    VDC: TSpeedButton;
    VAC: TSpeedButton;
    ADC: TSpeedButton;
    AAC: TSpeedButton;
    OHM: TSpeedButton;
    Panel3: TPanel;
    Label12: TLabel;
    Shape8: TShape;
    AUTO: TSpeedButton;
    Button1: TButton;
    DataSource1: TDataSource;
    Table1: TTable;
    Table1Information: TStringField;
    Table1Data: TStringField;
    Timer1: TTimer;
    procedure VDCClick(Sender: TObject);
    procedure VACClick(Sender: TObject);
     procedure ADCClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure AACClick(Sender: TObject);
    procedure OHMClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
   { Private declarations }
  public
   { Public declarations }
  end;
const
  ATN    =  $10;        (* Attention asserted                  *)
  TACS   =   $8;        (* Talker active                 const
  (* GPIB status bit definitions.                       *)
  ERR    = $8000;       (* Error detected                      *)
  TIMO   = $4000;       (* Timeout                             *)
  ENDgpib = $2000;      (* EOI or EOS detected                 *)
  SRQI   = $1000;       (* SRQ detected by CIC                 *)
  RQS    =  $800;       (* Device needs service                *)
  SPOLL  =  $400;       (* Board has been serially polled      *)
  EVENT  =  $200;       (* An event has occurred               *)
  CMPL   =  $100;       (* I/O completed                       *)
  LOK    =   $80;       (* Local lockout state                 *)
  REM    =   $40;       (* Remote state                        *)
  CIC    =   $20;       (* Controller-in-charge                *)
  LACS   =    $4;       (* Listener active                     *)
  DTAS   =    $2;       (* Device trigger state                *)
  DCAS   =    $1;       (* Device clear state                  *)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(* Error messages returned in global variable IBERR:          *)
EDVR =  0;          (* System error                            *)
ECIC =  1;          (* Function requires GPIB board to be CIC  *)
ENOL =  2;          (* Write function detected no Listeners    *)
EADR =  3;          (* Interface board not addressed correctly *)
EARG =  4;          (* Invalid argument to function call       *)
ESAC =  5;          (* Function requires GPIB board to be SAC  *)
EABO =  6;          (* I/O operation aborted                   *)
ENEB =  7;          (* Non-existent interface board            *)
EDMA =  8;          (* Error performing DMA                    *)
EOIP = 10;          (* I/O operation started before previous   *)
                       (* operation completed                  *)
ECAP = 11;          (* No capability for intended operation    *)
EFSO = 12;          (* File system operation error             *)
EBUS = 14;          (* Command error during device call        *)
ESTB = 15;          (* Serial poll status byte lost            *)
ESRQ = 16;          (* SRQ remains asserted                    *)
ETAB = 20;          (* The return buffer is full               *)
(* Values for the 488.2 Send command.                          *)
NULLend = $00;       (* Do nothing at the end of a transfer.   *)
NLend   = $01;       (* Send NL with EOI after a transfer.     *)
DABend  = $02;       (* Send EOI with the last DAB.            *)
STOPend = $100;      (* Value used by the 488.2 Receive command. *)
MAVbit  = $10;       (* Position of the Message Available bit. *)
(* Timeout values and meanings:                                *)
TNONE   = 0;         (* Infinite timeout (disabled)            *)
T10us   = 1;         (* Timeout of 10 microseconds (ideal)     *)
T30us   = 2;         (* Timeout of 30 microseconds (ideal)     *)
T100us  = 3;         (* Timeout of 100 microseconds (ideal)    *)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  T300us  =  4;          (* Timeout of 300 microseconds (ideal)     *)
  T1ms    =  5;          (* Timeout of 1 milliseconds (ideal)       *)
  T3ms    =  6;          (* Timeout of 3 milliseconds (ideal)       *)
  T10ms   =  7;          (* Timeout of 10 milliseconds (ideal)      *)
  T30ms   =  8;          (* Timeout of 30 milliseconds (ideal)      *)
  T100ms  =  9;          (* Timeout of 100 milliseconds (ideal)     *)
  T300ms  = 10;          (* Timeout of 300 milliseconds (ideal)     *)
  T1s     = 11;          (* Timeout of 1 second (ideal)             *)
  T3s     = 12;          (* Timeout of 3 seconds (ideal)            *)
  T10s    = 13;          (* Timeout of 10 seconds (ideal)           *)
  T30s    = 14;          (* Timeout of 30 seconds (ideal)           *)
  T100s   = 15;          (* Timeout of 100 seconds (ideal)          *)
  T300s   = 16;          (* Timeout of 300 seconds (ideal)          *)
  T1000s  = 17;          (* Timeout of 1000 seconds (ideal)         *)
type
  (* Type declarations for exported NI-488.2 Global Variables.      *)
  Tibsta  = function : integer; stdcall;
  Tiberr  = function : integer; stdcall;
  Tibcntl = function : Longint; stdcall;
  (* Type declarations for exported NI-488.2 routines.              *)
  Tibclr = function(ud : integer): integer; stdcall;
  Tibdev = function(ud : integer;
                    pad : integer;
                    sad : integer;
                    tmo : integer;
                    eot : integer;
                    eos : integer): integer;stdcall;
  Tibrd = function(ud : integer;
                   var rdbuf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cnt  : longint) : integer; stdcall;
Tibtrg = function(ud : integer) : integer; stdcall;
Tibwrt = function(ud : integer;
            var wrtbuf;
            cnt : longint): integer;stdcall;
Tibwait = function(ud : integer;
            mask: integer): integer; stdcall;
TDevClear = procedure (boardID: integer;
                        address: word); stdcall;
Tibonl = function (ud: integer;
            v:  integer) : integer; stdcall;
TRcvRespMsg = procedure ( boardID: integer;
              var buffer;
              cnt:  longint;
              termination : integer); stdcall;
TReceiveSetup = procedure (boardID : integer;
              address : word ); stdcall;
TReceive = procedure (boardID:    integer;
              address:    word;
              var buffer;
              cnt:        longint;
              termination: integer); stdcall;
TSendSetup = procedure (boardID: integer;
             address:  word); stdcall;
TSendDataBytes = procedure (boardID: integer;
               var buffer;
               datacount: longint;
               eotmode: integer); stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TSend = procedure (boardID: integer;
                       address: word;
                       var buffer;
                       datàcnt: longint;
                       eotmode: integer); stdcall;
    TSendIFC = procedure (boardID: integer); stdcall;
  var
    MULTIMETER: TMULTIMETER;
  (* NI-488.2 GPIB global status variables                    *)
    AddrIbsta  : Tibsta;
    AddrIberr  : Tiberr;
    AddrIbcntl : Tibcntl;
  (* Pointers to the NI-488.2 GPIB global status variables    *)
    Pibsta : ^integer;
    Piberr : ^integer;
    Pibcntl : ^Longint;
  (* Declaration for the Handle for the GPIB library          *)
    Gpib32Lib: THandle;
  (* Declarations for the NI-488.2 GPIB calls                 *)
    ibclr : Tibclr;
    ibdev : Tibdev;
    ibrd  : Tibrd;
    ibtrg : Tibtrg;
    ibwait: Tibwait;
    ibwrt : Tibwrt;
    DevClear: TDevClear;
    Receive: TReceive;
    ReceiveSetup: TReceiveSetup;
    RcvRespMsg: TRcvRespMsg;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  SendDataBytes: TSendDataBytes;
  SendSetup: TSendSetup;
  Send: TSend;
  SendIFC: TSendIFC;
 (* Declaration for the NI-488 GPIB call                    *)
  ibonl: Tibonl;
  dvm : integer;  {* device Number *}
  index :integer;
  bufn:string;
  implementation
{$R *.DFM}
 procedure loadDLL;
 var
  str : string;
  begin
  (* Load the GPIB-32.DLL library using the LoadLibrary function.      *)
    Gpib32Lib := LoadLibrary('GPIB-32.DLL');
    If Gpib32Lib = 0 Then
     Begin
       str := 'LoadLibrary FAILED!';
       MessageDlg(str, mtError, [mbOK], 0);
       halt;
     End;
  (* Get the addresses of the GPIB Global Variables.              *)
    @AddrIbsta  := GetProcAddress(Gpib32Lib, 'user_ibsta');
    @AddrIberr  := GetProcAddress(Gpib32Lib, 'user_iberr');
    @AddrIbcntl := GetProcAddress(Gpib32Lib, 'user_ibcnt');
  (* Get the addresses of the functions needed for this application.   *)
    @ibclr        := GetProcAddress(Gpib32Lib, 'ibclr');
    @ibdev        := GetProcAddress(Gpib32Lib, 'ibdev');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@ibrd          := GetProcAddress(Gpib32Lib, 'ibrd');
@ibtrg         := GetProcAddress(Gpib32Lib, 'ibtrg');
@ibwait        := GetProcAddress(Gpib32Lib, 'ibwait');
@ibwrt         := GetProcAddress(Gpib32Lib, 'ibwrt');
@DevClear      := GetProcAddress(Gpib32Lib, 'DevClear');
@ibonl         := GetProcAddress(Gpib32Lib, 'ibonl');
@RcvRespMsg    := GetProcAddress(Gpib32Lib, 'RcvRespMsg');
@ReceiveSetup  := GetProcAddress(Gpib32Lib, 'ReceiveSetup');
@Receive       := GetProcAddress(Gpib32Lib, 'Receive');
@SendSetup     := GetProcAddress(Gpib32Lib, 'SendSetup');
@SendDataBytes := GetProcAddress(Gpib32Lib, 'SendDataBytes');
@Send          := GetProcAddress(Gpib32Lib, 'Send');
@SendIFC       := GetProcAddress(Gpib32Lib, 'SendIFC');
if (@ibdev      = NIL) or
   (@ibclr      = NIL) or
   (@ibrd       = NIL) or
   (@ibtrg      = NIL) or
   (@ibwait     = NIL) or
   (@ibwrt      = NIL) or
   (@AddrIbsta  = NIL) or
   (@AddrIberr  = NIL) or
   (@AddrIbcntl = NIL) or
   (@DevClear   = NIL) or
   (@ibonl      = NIL) or
   (@RcvRespMsg    = NIL) or
   (@ReceiveSetup  = NIL) or
   (@Receive       = NIL) or
   (@SendDataBytes = NIL) or
   (@SendSetup     = NIL) or
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    (@Send          = NIL) or
    (@SendIFC       = NIL) Then
    Begin
      str := 'GetProcAddresses FAILED!';
      MessageDlg(str, mtError, [mbOK], 0);
      FreeLibrary(Gpib32Lib);
      halt;
    End;
(* Initialize GPIB global pointers to point to address location       *)
    Pibsta := @AddrIbsta;
    Piberr := @AddrIberr;
    Pibcntl := @AddrIbcntl;
 end;
function result_v(msg:string;n:integer):string;
var  dat:real;i:integer;
begin    if msg[1]<>'E' then begin
         val(msg,dat,i);
         str(dat:7:n,msg);end else begin msg:='ERROR 01';END;
         multimeter.table1.FindNearest(['read']);
         multimeter.table1.Edit;
         multimeter.table1data.AsString:=msg;
         multimeter.table1.Post;
         result_v:=msg;
end;
function result_i(msg:string;n:integer):string;
var  dat:real;i:integer;
begin
       if msg[1]<>'E' then begin
         val(msg,dat,i);
         dat:=dat*1000;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        str(dat:7:n,msg);end else begin msg:='ERROR 01';END;
        multimeter.table1.FindNearest(['read']);
        multimeter.table1.Edit;
        multimeter.table1data.AsString:=msg;
        multimeter.table1.Post;
        result_i:=msg;
end;
function result_ohm(msg:string;n:integer):string;
var  DAT:REAL;I:INTEGER;
begin
      if msg[1]<>'E' then begin
        val(msg,dat,i);
        dat:=dat*0.001;
        str(dat:7:n,msg);end else begin msg:='ERROR 01';END;
        multimeter.table1.FindNearest(['read']);
        multimeter.table1.Edit;
        multimeter.table1data.AsString:=msg;
        multimeter.table1.Post;
        result_ohm:=msg;
end;
procedure autorange(sett:integer);
var  buf:packed array[0..32]of char;
   msg:string;
begin
    case sett of
    1:begin
          multimeter.shape2.Brush.color:=clred;
          multimeter.shape3.brush.color:=clmaroon;
          multimeter.shape4.brush.color:=clmaroon;
          multimeter.shape5.brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        multimeter.shape6.brush.color:=clmaroon;
        buf:='A1;VD;';      //VDC
        ibwrt(dvm,buf,10);
        buf:='T3;S1;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        MULTIMETER.label2.caption:=result_v(msg,7);
        MULTIMETER.label3.caption:='V';
        end ;
2:begin
        multimeter.shape2.Brush.color:=clmaroon;
        multimeter.shape3.brush.color:=clred;
        multimeter.shape4.brush.color:=clmaroon;
        multimeter.shape5.brush.color:=clmaroon;
        multimeter.shape6.brush.color:=clmaroon;
        buf:='A1;VA;';          // VAC
        ibwrt(dvm,buf,12);
        buf:='T3;S1;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        MULTIMETER.label2.caption:=result_v(msg,7);
        MULTIMETER.Label3.caption:='V';
        end;
3:begin
        multimeter.shape2.Brush.color:=clmaroon;
        multimeter.shape3.brush.color:=clmaroon;
        multimeter.shape4.brush.color:=clred;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        multimeter.shape5.brush.color:=clmaroon;
        multimeter.shape6.brush.color:=clmaroon;
        buf:='A1;ID;';          //Idc
        ibwrt(dvm,buf,10);
        buf:='T3;S1;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        MULTIMETER.label2.caption:=result_i(msg,7);
        MULTIMETER.label3.caption:='mA';
        end;
4:begin
        multimeter.shape2.Brush.color:=clmaroon;
        multimeter.shape3.brush.color:=clmaroon;
        multimeter.shape4.brush.color:=clmaroon;
        multimeter.shape5.brush.color:=clred;
        multimeter.shape6.brush.color:=clmaroon;
        buf:='A1;IA;';      // Iac
        ibwrt(dvm,buf,12);
        buf:='T3;S1;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        MULTIMETER.label2.caption:=result_i(msg,7);
        MULTIMETER.label3.caption:='mA';
        end ;
5:begin
        multimeter.shape2.Brush.color:=clmaroon;
        multimeter.shape3.brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            multimeter.shape4.brush.color:=clmaroon;
            multimeter.shape5.brush.color:=clmaroon;
            multimeter.shape6.brush.color:=clred;
            buf:='A1;O2;';//OHM
            ibwrt(dvm,buf,15);
            buf:='T3;S1;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            MULTIMETER.label2.caption:=result_ohm(msg,7);
            MULTIMETER.label3.caption:='Kohm';
            end ;
  end;
end;
procedure checktable ;
begin
   multimeter.Table1.FindNearest(['range']);
   bufn:=multimeter.Table1data.AsString;
   multimeter.Table1.FindNearest(['range']);
   bufn:=multimeter.Table1data.AsString;
   if ((bufn[1]='A')and(bufn[2]='1')) THEN
    begin multimeter.shape8.Brush.color:=clred;
       if ((bufn[7]='V')and(bufn[8]='D')) then
         autorange(1);
       if ((bufn[7]='V')and(bufn[8]='A')) then
         autorange(2);
       if ((bufn[7]='I')and(bufn[8]='D'))then
         autorange(3);
       if ((bufn[7]='I')and(bufn[8]='A'))then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            autorange(4);
          if ((bufn[7]='O')and(bufn[8]='2'))then
            autorange(5);
        end
      else begin multimeter.shape8.Brush.color:=clmaroon;
          if ((bufn[7]='V')and(bufn[8]='D')) then
            multimeter.VDCClick(nil);
          if ((bufn[7]='V')and(bufn[8]='A')) then
            multimeter.VACClick(nil);
          if ((bufn[7]='I')and(bufn[8]='D')) then
            multimeter.ADCClick(nil);
          if ((bufn[7]='I')and(bufn[8]='A')) then
            multimeter.AACClick(nil);
          if ((bufn[7]='O')and(bufn[8]='2')) then
            multimeter.OHMClick(nil);
        end;
  end;
  procedure vdcRange(ra:integer);
  var buf:packed array[0..15]of char; msg:string;
  begin  case ra of
    1: begin buf:='A0;T3;VD;';  // vdc
          ibwrt(dvm,buf,10);
          buf:='S1;R1;';
          ibwrt(dvm,buf,10);
          ibrd(dvm,buf,12);
          msg:=buf;
          multimeter.label2.caption:=result_v(msg,7);
          multimeter.label3.caption:='V';
       end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2:begin buf:='A0;T3;VD;';  // vdc
      ibwrt(dvm,buf,10);
      buf:='S1;R2;';
      ibwrt(dvm,buf,10);
      ibrd(dvm,buf,12);
      msg:=buf;
      multimeter.label2.caption:=result_v(msg,6);
      multimeter.label3.caption:='V';
  end;
3: begin buf:='A0;T3;VD;';  // vdc
      ibwrt(dvm,buf,10);
      buf:='S1;R3;';
      ibwrt(dvm,buf,10);
      ibrd(dvm,buf,12);
      msg:=buf;
      multimeter.label2.caption:=result_v(msg,5);
      multimeter.label3.caption:='V';
 end;
 4: begin buf:='A0;T3;VD;';  // vdc
      ibwrt(dvm,buf,10);
      buf:='S1;R4;';
      ibwrt(dvm,buf,10);
      ibrd(dvm,buf,12);
      msg:=buf;
      multimeter.label2.caption:=result_v(msg,4);
      multimeter.label3.caption:='V';
  end;
 5: begin buf:='A0;T3;VD;';  // vdc
      ibwrt(dvm,buf,10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            buf:='S1;R5;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,3);
            multimeter.label3.caption:='V';
          end;
        end;
    end;
    procedure vacRange(ra:integer);
    var buf:packed array[0..15]of char; msg:string;
    begin  case ra of
      1: begin buf:='A0;T3;VA;';  // VAC
            ibwrt(dvm,buf,10);
            buf:='S1;R1;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,7);
            multimeter.label3.caption:='V';
         end;
        2:begin buf:='A0;T3;VA;';  // VAC
            ibwrt(dvm,buf,10);
            buf:='S1;R2;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,6);
            multimeter.label3.caption:='V';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;
    3: begin buf:='A0;T3;VA;';  // VAC
            ibwrt(dvm,buf,10);
            buf:='S1;R3;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,5);
            multimeter.label3.caption:='V';
        end;
        4: begin buf:='A0;T3;VA;';  // VAC
            ibwrt(dvm,buf,10);
            buf:='S1;R4;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,4);
            multimeter.label3.caption:='V';
        end;
        5: begin buf:='A0;T3;VA;';  // VAC
            ibwrt(dvm,buf,10);
            buf:='S1;R5;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_v(msg,3);
            multimeter.label3.caption:='V';
        end;
    end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
procedure idcRange(ra:integer);
var buf:packed array[0..15]of char; msg:string;
begin  case ra of
   1: begin buf:='A0;T3;ID;';  // IDC
        ibwrt(dvm,buf,10);
        buf:='S1;R2;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        multimeter.label2.caption:=result_i(msg,5);
        multimeter.label3.caption:='mA';
      end;
     2:begin buf:='A0;T3;ID;';  // IDC
        ibwrt(dvm,buf,10);
        buf:='S1;R5;';
        ibwrt(dvm,buf,10);
        ibrd(dvm,buf,12);
        msg:=buf;
        multimeter.label2.caption:=result_i(msg,2);
        multimeter.label3.caption:='mA';
      end;
  end;
end;
procedure iacRange(ra:integer);
var buf:packed array[0..15]of char; msg:string;
begin  case ra of
   1: begin buf:='A0;T3;IA;';  // IAC
        ibwrt(dvm,buf,10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            buf:='S1;R2;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_i(msg,5);
            multimeter.label3.caption:='A';
        end;
       2:begin buf:='A0;T3;IA;';  // IAC
            ibwrt(dvm,buf,10);
            buf:='S1;R5;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_i(msg,2);
            multimeter.label3.caption:='A';
         end;
    end;
end;
procedure ohmRange(ra:integer);
var buf:packed array[0..15]of char; msg:string;
begin  case ra of
   1: begin buf:='A0;T3;02;';  // ohm
            ibwrt(dvm,buf,10);
            buf:='S1;R1;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_ohm(msg,7);
            multimeter.label3.caption:='Kohm';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
2:begin buf:='A0;T3;02;';  // ohm
    ibwrt(dvm,buf,10);
    buf:='S1;R2;';
    ibwrt(dvm,buf,10);
    ibrd(dvm,buf,12);
    msg:=buf;
    multimeter.label2.caption:=result_ohm(msg,6);
    multimeter.label3.caption:='Kohm';
 end;
3: begin buf:='A0;T3;02;';  // ohm
    ibwrt(dvm,buf,10);
    buf:='S1;R3;';
    ibwrt(dvm,buf,10);
    ibrd(dvm,buf,12);
    msg:=buf;
    multimeter.label2.caption:=result_ohm(msg,5);
    multimeter.label3.caption:='Kohm';
 end;
 4: begin buf:='A0;T3;02;';  // ohm
    ibwrt(dvm,buf,10);
    buf:='S1;R4;';
    ibwrt(dvm,buf,10);
    ibrd(dvm,buf,12);
    msg:=buf;
    multimeter.label2.caption:=result_ohm(msg,4);
    multimeter.label3.caption:='Kohm';
 end;
 5: begin buf:='A0;T3;02;';  // ohm
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ibwrt(dvm,buf,10);
            buf:='S1;R5;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_ohm(msg,3);
            multimeter.label3.caption:='Kohm';
        end;
        6: begin buf:='A0;T3;02;';  // ohm
            ibwrt(dvm,buf,10);
            buf:='S1;R6;';
            ibwrt(dvm,buf,10);
            ibrd(dvm,buf,12);
            msg:=buf;
            multimeter.label2.caption:=result_ohm(msg,2);
            multimeter.label3.caption:='Kohm';
        end;
    end;
end;
procedure TMULTIMETER.VDCClick(Sender: TObject);
begin
    shape2.brush.color:=clred;
    shape3.brush.color:=clmaroon;
    shape4.brush.color:=clmaroon;
    shape5.brush.color:=clmaroon;
    shape6.brush.color:=clmaroon;
      if bufn[14]='1' then
        vdcRange(1);
      if bufn[14]='2' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            vdcRange(2);
          if bufn[14]='3' then
            vdcRange(3);
          if bufn[14]='4' then
            vdcRange(4);
          if bufn[14]='5' then
            vdcRange(5);
    end;
procedure TMULTIMETER.VACClick(Sender: TObject);
begin
      shape2.Brush.color:=clmaroon;
      shape3.brush.color:=clred;
      shape4.brush.color:=clmaroon;
      shape5.brush.color:=clmaroon;
      shape6.brush.color:=clmaroon;
        if bufn[14]='1' then
          vacRange(1);
        if bufn[14]='2' then
          vacRange(2);
        if bufn[14]='3' then
          vacRange(3);
        if bufn[14]='4' then
          vacRange(4);
        if bufn[14]='5' then
          vacRange(5);
  end;
procedure TMULTIMETER.ADCClick(Sender: TObject);
begin
    shape2.brush.color:=clmaroon;
    shape3.brush.color:=clmaroon;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shape4.brush.color:=clred;
    shape5.brush.color:=clmaroon;
    shape6.brush.color:=clmaroon;
      if bufn[14]='2' then
       idcRange(1);
      if bufn[14]='5' then
       idcRange(2);
end;
procedure TMULTIMETER.AACClick(Sender: TObject);
begin
    shape2.brush.color:=clmaroon;
    shape3.brush.color:=clmaroon;
    shape4.brush.color:=clmaroon;
    shape5.brush.color:=clred;
    shape6.brush.color:=clmaroon;
     if bufn[14]='2' then
       iacRange(1);
     if bufn[14]='5' then
       iacRange(2);
end;
procedure TMULTIMETER.OHMClick(Sender: TObject);
begin
    shape2.brush.color:=clmaroon;
    shape3.brush.color:=clmaroon;
    shape4.brush.color:=clmaroon;
    shape5.brush.color:=clmaroon;
    shape6.brush.color:=clred;
     if bufn[14]='1' then
       ohmRange(1);
     if bufn[14]='2' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ohmRange(2);
        if bufn[14]='3' then
            ohmRange(3);
        if bufn[14]='4' then
            ohmRange(4);
        if bufn[14]='5' then
            ohmRange(5);
        if bufn[14]='6' then
            ohmRange(6);
    end;
    procedure TMULTIMETER.FormShow(Sender: TObject);
    begin
        loaddll;
        dvm:=ibdev(0,7,0,15,1,0);
        timer1.Enabled:=true;
    end;
    procedure TMULTIMETER.FormClose(Sender: TObject; var Action: TCloseAction);
    begin
        ibonl(0,0);
        FreeLibrary(Gpib32Lib);
    end;
    procedure TMULTIMETER.Button1Click(Sender: TObject);
    begin
        Close;
    end;
    procedure TMULTIMETER.Timer1Timer(Sender: TObject);
    begin
        table1.Refresh;
        checktable;
    end;  end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Neil J.Rubenking เรียบเรียงโดย ทรงศักดิ์ บรรจงมณี, แรกเริ่มเรียนรู้ เรื่องการเขียนโปรแกรมด้วย DELPHI, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2540
2. NI-488.2M Function Reference Manual for Win32, National Instrument Corporation, 1996
3. NI-488.M User Manual for Windows95 and WindowsNT, National Instrument Corporation, 1996
4. L.Rantanen,"Using NI-488.2M Sorfware with 32-Bit Delphi Applications," Application Note 085,National Instrument.
5. Todd Miller, David Powell, Etal, USING DELPHI3, Que Corporation, 1997
6. Bob Brancheck, Peter Hazlehurst, Stephen Wyhkoop and Scott L Warner, USING MICROSOFT SQL SEVER 6.5, Que Corporation, 1996
7. Fred Halsall, Data Communications, Computer Network and Open Systems, Fourth Edition, Addison-Wesly Publishing Company,1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้