

การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยระบบเครือข่าย

HOME NETWORKING



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เลขที่.....
เลขทะเบียน..... 39670
วัน, เดือน, ปี..... 19 ส.ย. 2544

.....
.....
.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากกรณีใด ๆ ที่ส.ย. อื่น ๆ สามารถให้คำปรึกษาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HOME NETWORKING



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

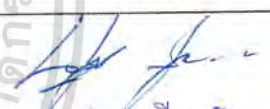
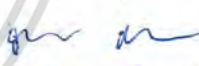

หัวข้อปัญหาพิเศษ การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยระบบเครือข่าย
HOME NETWORKING

ชื่อนักศึกษา นางสาวจรรวี ศิริจางคพัฒนา 40056011
นางสาวนงเยาว์ จินดาสวัสดิ์ 40056037
นางสาวสุพิทา ศิริลักษณ์ 40056103

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล
อาจารย์วิสันต์ ตั้งวงษ์เจริญ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้รับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2543

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	อาจารย์นรฤทธิ์ สุนทรศารทูล	
กรรมการ	อาจารย์เทอดขวัญ ช้างเผือก	เทอดขวัญ, ช้างเผือก
กรรมการและอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
กรรมการและอาจารย์ที่ปรึกษา	อาจารย์วิสันต์ ตั้งวงษ์เจริญ	



(ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านผ่านระบบเครือข่าย	
ชื่อนักศึกษา	นางสาวจาววี ศิริจางคพัฒนา	40056011
	นางสาวนงเยาว์ จินดาสวัสดิ์	40056037
	นางสาวสุพิธา ศิริลักษณ์	40056103
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2543	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
	อาจารย์วิสันต์ ตั้งวงษ์เจริญ	

บทคัดย่อ

ในปัจจุบันนี้เราได้นำคอมพิวเตอร์มาใช้ในการทำงานทางด้านบัญชี การเก็บข้อมูล การคำนวณหรือแม้กระทั่งการพิมพ์เอกสาร นอกจากนี้ที่กล่าวมาแล้ว เราสามารถนำคอมพิวเตอร์มาประยุกต์ใช้ในการควบคุมอุปกรณ์ไฟฟ้าโดยควบคุมผ่านพอร์ตขนานหรือพอร์ตเครื่องพิมพ์ เพื่อช่วยอำนวยความสะดวกและรักษาความปลอดภัยให้กับบุคคลภายในบ้านมากยิ่งขึ้น เราสามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจากที่ใดก็ได้ โดยผ่านระบบเครือข่ายในรูปแบบการทำงานของ Client/Server โดยจะให้เครื่องคอมพิวเตอร์ที่บ้านทำงานเป็นเครื่องให้บริการหรือเครื่องเซิร์ฟเวอร์ และเครื่องที่เราสั่งควบคุมอุปกรณ์ไฟฟ้าคือเครื่องไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	HOME NETWORKING	
Students	Miss. Jarawee Sirijangkapattana	40056011
	Miss. Nongyao Jindasawat	40056037
	Miss. Supita Sirilaksana	40056103
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Academic Year	2000	
Special Project Advisor	Assistant Professor Teerawat Prakorbphon	
	Lecturer Wisan Tangwongcharoen	

ABSTRACT

At the present moment, we couldn't not refuse that computers involve all activities of our lives, for example accounting, storing information, calculating. Not only that we can apply the computers to control the home electric equipment through parallel port (printer port) for improving our convenience and security in the residences. Moreover we can control them from anywhere by client/server network system, home computer (Server) and other computers (Client).

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษ เรื่องการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยระบบเครือข่าย (Home Networking) สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล และอาจารย์วิสันต์ ตั้งวงษ์เจริญ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ยปรึกษาในการแก้ปัญหาดังกล่าว รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดาที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2544



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา.....	1
1.5 ขอบเขตการศึกษา.....	1
1.6 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง.....	5
2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน.....	5
2.1.1 ลักษณะทางกายภาพของพอร์ตขนาน.....	5
2.1.2 พอร์ตดาต้า.....	9
2.1.3 พอร์ตคอนโทรล.....	11
2.1.4 พอร์ตแสดงสถานะ.....	12
2.1.5 การนำพอร์ตขนานไปใช้งาน.....	13
2.1.6 การติดต่ออุปกรณ์เอาต์พุตอย่างง่ายของพอร์ตขนาน.....	13
2.2 ความรู้เบื้องต้นเกี่ยวกับ Servlet.....	13
2.2.1 บทนำ.....	13
2.2.2 Servlet คืออะไร.....	14
2.2.3 สถาปัตยกรรมพื้นฐานของ Servlet.....	16
2.2.4 เนื้อหาในส่วนของภาษาจาวาที่เกี่ยวข้องกับโปรแกรม.....	18
2.2.4.1 การเรียกใช้เมธอดในคลาส Runtime.....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.4.2 การเรียกใช้เมธอดทในคลาส Process.....	19
2.2.4.3 การใช้ Exceptions บอกข้อผิดพลาด.....	20
2.3 ความรู้เบื้องต้นในการติดต่อฐานข้อมูลผ่าน JDBC/ODBC Bridge.....	22
2.3.1 JDBC คืออะไร.....	22
2.3.2 ความสามารถของ JDBC.....	23
2.3.3 ชนิดของ JDBC Driver.....	23
2.3.4 ข้อดี-ข้อเสียของ JDBC/ODBC Driver.....	24
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน.....	26
3.1 ขั้นตอนการวิเคราะห์ระบบการทำงาน.....	26
3.1.1 การทำงาน Client/Server แบบทูเทียร์.....	26
3.2 ขั้นตอนการดำเนินงาน.....	31
3.2.1 การพัฒนาโปรแกรมด้วยภาษาจาวา.....	31
3.2.2 การเขียนโปรแกรมติดต่อพอร์ตขนานด้วย JNI.....	32
3.2.3 การเขียนโปรแกรมเรียกใช้งานไฟล์ ParallelPort.....	36
3.2.4 การเขียนโปรแกรมด้วยภาษา Visual Basic ติดต่อฐานข้อมูล.....	38
3.2.4.1 การเจ็ต ODBC เพื่อติดต่อ Microsoft Access.....	38
3.2.4.2 การเพิ่ม, ลบ และแก้ไขข้อมูลผู้ใช้งาน.....	38
3.3 ขั้นตอนการทดสอบการทำงาน.....	44
3.3.1 การทดสอบการใช้งานฝั่ง Server.....	44
3.3.2 การทดสอบการใช้งานฝั่ง Client.....	46
บทที่ 4 สรุปผลปัญหาและข้อเสนอแนะ.....	52
4.1 การทำงานของปัญหาพิเศษ.....	52
4.2 ผลของปัญหาพิเศษ.....	52
4.3 ข้อจำกัดของปัญหาพิเศษ.....	52
4.4 ปัญหาในการทำงานและวิธีการแก้ไข.....	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

ภาคผนวก.....	55
ภาคผนวก ก. ตัวอย่างอุปกรณ์การเรียนรู้อุปกรณ์เชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก.....	55
ภาคผนวก ข. คู่มือการติดตั้งโปรแกรม.....	61
 บรรณานุกรม.....	 68



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 สัญญาณสำคัญ ๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์.....	6
2.2 แสดงรายละเอียดของสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน.....	8
2.3 แสดงแอดเดรสของพอร์ตขนาน.....	9
2.4 สับคลาสต่าง ๆ ในคลาส RuntimeException.....	21



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 ขั้นตอนการดำเนินงานปัญหาพิเศษ.....	3
1.2 ขั้นตอนระยะเวลาในการดำเนินงาน.....	4
2.1 ไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์.....	5
2.2 แสดงระบบบัสภายในของพอร์ตขนาน.....	8
2.3 แสดงวงจรภายในของพอร์ต Data.....	10
2.4 แสดงวงจรภายในของพอร์ต Control.....	11
2.5 แสดงวงจรภายในของพอร์ต Status.....	12
2.6 สถาปัตยกรรมพื้นฐานของ Servlet.....	16
2.7 แสดง API ของ Servlet.....	17
2.8 แสดงโครงสร้างการติดต่อฐานข้อมูลโดยใช้ JDBC/ODBC Bridge.....	24
3.1 โครงสร้างการติดต่อกับพอร์ตขนานผ่านระบบเครือข่าย.....	26
3.2 โครงสร้างแสดงองค์ประกอบของระบบแบบทูเพียร์.....	27
3.3 แผนภาพ Flow Chart แสดงขั้นตอนการตรวจสอบระบบ.....	28
3.4 แผนภาพ Flow Chart แสดงขั้นตอนการควบคุมอุปกรณ์ไฟฟ้าแบบทั่วไป.....	29
3.5 แผนภาพ Flow Chart แสดงขั้นตอนการควบคุมอุปกรณ์ไฟฟ้าแบบเฉพาะแต่ละบ้าน.....	30
3.6 วิธีการเซต PATH และ CLASSPATH ใน Autoexec.bat.....	32
3.7 หน้าจอสำหรับเพิ่ม, ลบ และแก้ไขผู้ใช้งานในฐานข้อมูล.....	39
3.8 แสดงหน้าจอตัวอย่างการเพิ่มผู้ใช้งาน.....	44
3.9 แสดงหน้าจอหลังจากเพิ่มผู้ใช้งานเรียบร้อยแล้ว.....	44
3.10 แสดงหน้าจอตัวอย่างการเปลี่ยนรหัสผู้ใช้งาน.....	45
3.11 แสดงหน้าจอหลังจากเปลี่ยนผู้ใช้งานเรียบร้อยแล้ว.....	45
3.12 แสดงหน้าจอตัวอย่างการลบผู้ใช้งาน.....	46
3.13 แสดงหน้าจอหลังจากลบผู้ใช้งานเรียบร้อยแล้ว.....	46
3.14 แสดงหน้าจอแรกเพื่อเข้าสู่โปรแกรมการใช้งาน.....	47
3.15 แสดงหน้าจอตัวอย่างการ Login เข้าสู่โปรแกรม.....	48
3.16 แสดงหน้าจอเลือกการใช้งาน 2 แบบ.....	49
3.17 แสดงหน้าจอหลังจากเลือกใช้งานในแบบที่ 1.....	49
3.18 แสดงหน้าจอหลังจากเลือกใช้งานในแบบที่ 2.....	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูป (ต่อ)

รูปที่	หน้า
3.19 แสดงหน้าจอหลังจากทำการเปิด-ปิดอุปกรณ์ไฟฟ้าเรียบร้อยแล้ว.....	50
3.20 แสดงหน้าจอของ Servletrunner แจ้งสถานะของผู้ใช้งาน.....	51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันสังคมเรามีการแข่งขันกันอย่างมาก ทำให้เวลาที่ใช้นอกบ้านมากกว่าการอยู่ที่บ้าน ดังนั้นถ้าเราสามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจากนอกบ้านได้ ก็จะเป็นประโยชน์ เช่น ถ้าวันใดเรามีธุระที่ทำให้กลับบ้านดึก เราก็สามารถสั่งให้เครื่องคอมพิวเตอร์ทำการเปิดไฟหน้าบ้านของเราได้ อีกทั้งในปัจจุบันเครื่องคอมพิวเตอร์ ก็ถือว่าเป็นอุปกรณ์ที่มีใช้กันอย่างแพร่หลายแทบจะทุกบ้าน จึงมีแนวคิดที่น่าที่จะนำเครื่องคอมพิวเตอร์ในบ้านมาเชื่อมโยงกับระบบเครือข่าย และอุปกรณ์ต่าง ๆ เพื่อว่าเวลาที่เรายอยู่นอกบ้านก็สามารถติดต่อกับอุปกรณ์ภายในบ้านได้โดยง่าย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อช่วยป้องกันอันตรายที่อาจเกิดขึ้น เช่น ลืมปิดเครื่องใช้ไฟฟ้าภายในบ้าน

1.2.2 เพื่อให้ได้รับความรู้เกี่ยวกับการติดต่อกับระบบเครือข่าย ในรูป Client-Server มากขึ้น

1.2.3 เพื่อให้ทราบขั้นตอนการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าโดยผ่านทางพอร์ตขนาน

1.3 สมมติฐานของการศึกษา

1.3.1 สามารถเขียนโปรแกรมควบคุมอุปกรณ์ให้ทำโดยผ่านพอร์ตขนานได้

1.3.2 สามารถเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ไฟฟ้าภายในบ้านผ่านระบบเครือ

ข่าย

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา

1.4.1 สถาปัตยกรรมของพอร์ตขนาน

1.4.2 การรับส่งข้อมูลแบบขนานระหว่างพอร์ตขนานกับเครื่องพิมพ์

1.4.3 การติดต่อบริเวณเครือข่ายในรูปแบบของ Client-Server

1.5 ขอบเขตการศึกษา

สามารถใช้คอมพิวเตอร์ในการเปิด - ปิดเครื่องใช้ไฟฟ้าบางประเภทภายในบ้านได้ โดยผ่านระบบเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขั้นตอนการศึกษา

เนื่องจากการจัดทำปัญหาพิเศษนี้ มีความเกี่ยวข้องกับความรู้ทั้งในด้าน Hardware และ Software ดังนั้นในการปฏิบัติงานจึงต้องมีการวางแผนขั้นตอนในการทำงาน และศึกษาเนื้อหาในส่วนที่เกี่ยวข้อง สามารถแบ่งได้ดังต่อไปนี้

1.6.1 ศึกษาและรวบรวมข้อมูลเกี่ยวกับพอร์ตขนานและอุปกรณ์ทางด้านฮาร์ดแวร์

ในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านต้องทำงานผ่านพอร์ตขนาน (Parallel Port) หรือพอร์ตเครื่องพิมพ์ ดังนั้นจึงต้องศึกษาหาข้อมูลเกี่ยวกับพอร์ตขนาน และการทำงานของพอร์ตขนาน รวมทั้งอุปกรณ์ทางด้านฮาร์ดแวร์ที่จะมาช่วยในการทดสอบความถูกต้องของตัวโปรแกรมก่อนนำไปใช้จริงกับอุปกรณ์ไฟฟ้า

1.6.2 ศึกษาและรวบรวมข้อมูลภาษาในการโปรแกรมและการรับ-ส่งข้อมูลผ่านระบบเครือข่าย

ภาษาในการโปรแกรมปัญหาพิเศษนี้ ต้องมีความสามารถในด้านควบคุมการรับ-ส่งข้อมูลผ่านทางพอร์ตขนานรวมทั้งรองรับการทำงานผ่านเครือข่ายได้ดี ในส่วนนี้จึงต้องมีการตัดสินใจเลือกภาษาในการเขียนและพัฒนาโปรแกรมด้วยความรอบคอบ

1.6.3 ออกแบบโครงสร้างของระบบและเขียนโปรแกรม

ทำการออกแบบโครงสร้างการทำงานภายในระบบ ให้เห็นขั้นตอนในการทำงานทั้งหมดว่าตรงตามความต้องการหรือไม่ ถ้ายังไม่ตรงก็ทำการแก้ไข เมื่อการทำงานมีความถูกต้องตรงตามความต้องการแล้ว จึงเริ่มต้นขั้นตอนการพัฒนาโปรแกรม

1.6.4 ทำการทดสอบโปรแกรมและปรับปรุงแก้ไข

หลังจากที่ได้ทำการเขียนโปรแกรมเรียบร้อยแล้ว ต้องทำการตรวจสอบโดยทดสอบการทำงานจริง โดยลองให้ผู้อื่นมาทำการทดสอบใช้โปรแกรมเพื่อหาข้อผิดพลาดและทำการแก้ไข ปรับปรุงเพิ่มเติม และตกแต่งหน้าจอ User Interface

1.6.5 จัดทำเอกสารประกอบการใช้งาน

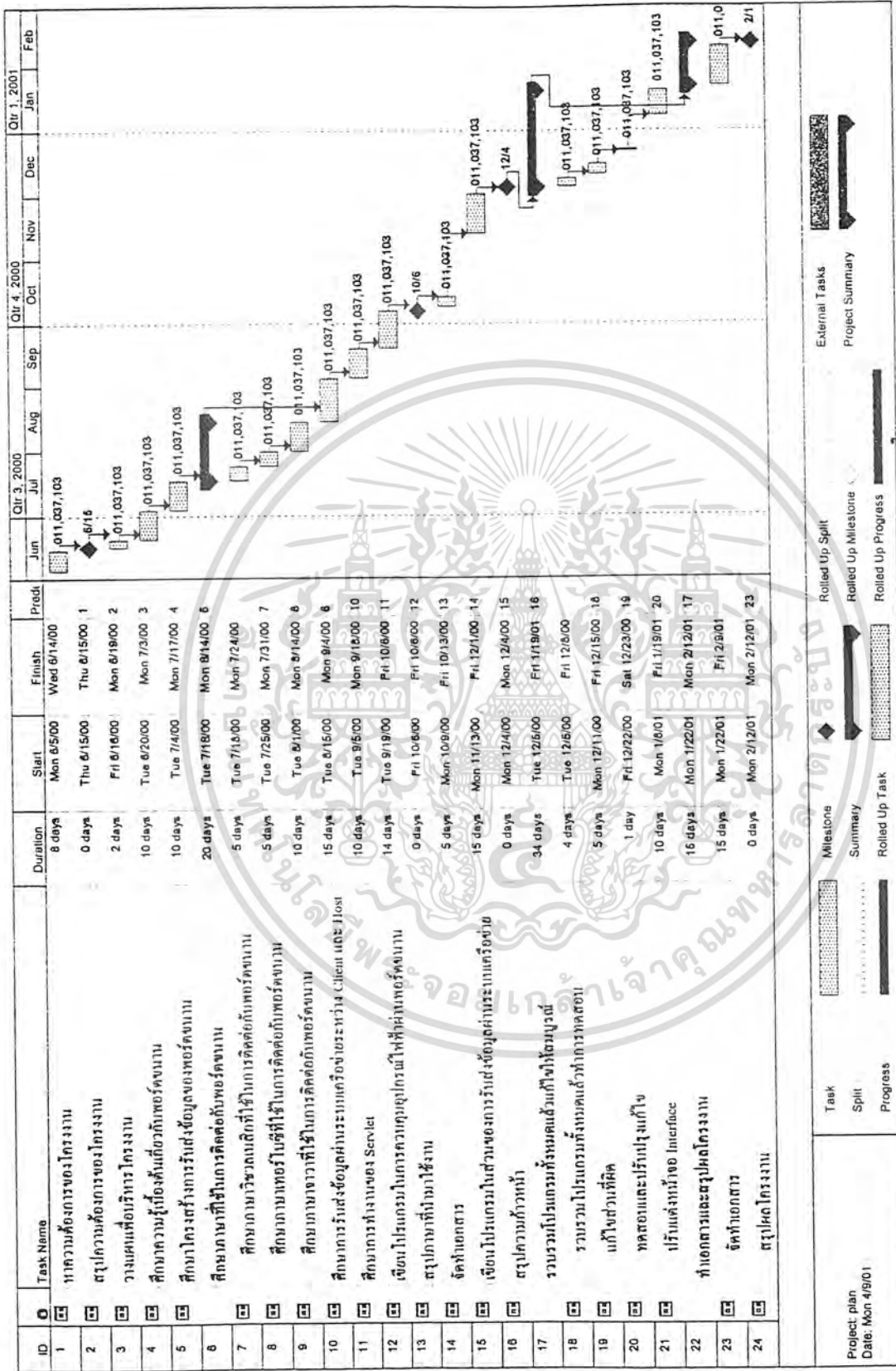
ต้องทำการจัดการเอกสารเพื่อประกอบการใช้งาน เพื่อแนะนำขั้นตอนการทำงาน วิธีการใช้งาน ตลอดจนจนถึงทฤษฎีต่าง ๆ ที่เกี่ยวข้อง

Task ID	Task Name	Duration	Start	Finish	Predecessors
1	ศึกษาความรู้เบื้องต้นเกี่ยวกับพอร์ตขานาน	10 days	Mon 19/6/00	Fri 30/6/00	
2	ศึกษาโครงสร้างการรับส่งข้อมูลของพอร์ตขานาน	10 days	Mon 3/7/00	Fri 14/7/00	1
3	ศึกษาภาษาที่ใช้ในการติดต่อฮาร์ดแวร์	20 days	Mon 17/7/00	Fri 11/8/00	2
4	ศึกษาภาษาวิวัฒนาการที่ใช้ในการติดต่อพอร์ตขานาน	5 days	Mon 17/7/00	Fri 21/7/00	
5	ศึกษาภาษาเอซีไอที่ใช้ในการติดต่อพอร์ตขานาน	5 days	Mon 24/7/00	Fri 28/7/00	4
6	ศึกษาภาษาจาวาที่ใช้ในการติดต่อพอร์ตขานาน	10 days	Mon 31/7/00	Fri 11/8/00	5
7	ศึกษาการรับส่งข้อมูลผ่านระบบเครือข่ายระหว่าง Client และ Host	15 days	Mon 14/8/00	Fri 1/9/00	
8	ศึกษาการทำงานของ Servlet	10 days	Mon 4/9/00	Fri 15/9/00	7
9	เขียนโปรแกรมในการควบคุมอุปกรณ์ไฟที่ผ่านพอร์ตขานาน	14 days	Mon 18/9/00	Thu 5/10/00	8
10	สรุปภาษาที่นำมาใช้งาน	0 days	Fri 6/10/00	Fri 6/10/00	9
11	จัดทำเอกสาร	5 days	Mon 9/10/00	Fri 13/10/00	10
12	เขียนโปรแกรมในส่วนของการรับส่งข้อมูลผ่านระบบเครือข่าย	15 days	Mon 13/11/00	Fri 1/12/00	
13	สรุปความก้าวหน้า	0 days	Mon 4/12/00	Mon 4/12/00	12
14	รวบรวมโปรแกรมทั้งหมดแล้วนำไปให้สมบูรณ์	24 days	Tue 5/12/00	Fri 5/1/01	13
15	รวบรวมโปรแกรมทั้งหมดแล้วทำการทดสอบ	4 days	Tue 5/12/00	Fri 8/12/00	
16	แก้ไขส่วนที่ผิด	5 days	Mon 11/12/00	Fri 15/12/00	15
17	ทดสอบและปรับปรุงแก้ไข	1 day	Fri 22/12/00	Sat 23/12/00	16
18	ปรับแต่งหน้าจอ Interface	10 days	Mon 25/12/00	Fri 5/1/01	17
19	ทำเอกสารและสรุปผลโครงการ	15 days	Mon 8/1/01	Fri 26/1/01	18
20	จัดทำเอกสาร	14 days	Mon 8/1/01	Thu 25/1/01	
21	สรุปผลโครงการ	0 days	Fri 26/1/01	Fri 26/1/01	20

รูปที่ 1.1 ขั้นตอนการดำเนินงานปัญหาพิเศษ

จากรูปที่ 1.1 เป็นขั้นตอนการดำเนินงานทั้งหมดในการวางแผนการทำงานซึ่งเริ่มตั้งแต่การศึกษาความรู้ต่าง ๆ ที่เกี่ยวข้องจนกระทั่งสรุปผลการทำโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 ขั้นตอนระยะเวลาในการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง

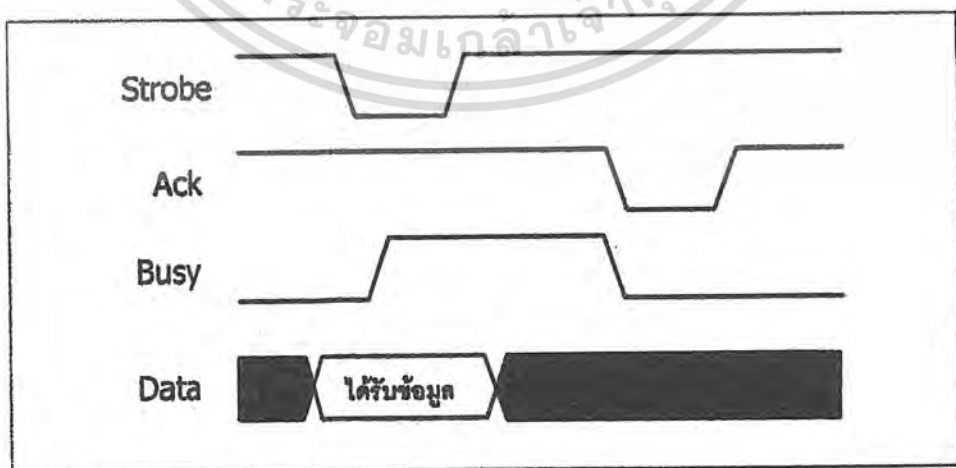
2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ตนาน

พอร์ตนาน (Parallel port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายเทข้อมูลของพอร์ตนี้เป็นแบบขนาน สำหรับชื่อเรียกอีกชื่อของพอร์ตนานคือ พอร์ตเครื่องพิมพ์ (Printer port) เนื่องจากพอร์ตนี้นี้ใช้สำหรับต่อเครื่องพิมพ์นั่นเอง

ด้วยการถ่ายเทข้อมูลแบบขนานนี้เอง ทำให้พอร์ตนานมีอัตราการถ่ายเทข้อมูลสูงกว่าการถ่ายเทข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้นพอร์ตนานจึงสามารถรองรับการถ่ายเทข้อมูล 8 บิตได้โดยไม่ต้องต่อส่วนเพิ่มเติม

2.1.1 ลักษณะทางกายภาพของพอร์ตนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่าปกติ นั้น การส่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตนานนั้น มีรูปแบบการทำงานภายในอย่างไรในรูปที่ 2.1 แสดงไดอะแกรมเวลาของการติดต่อระหว่างพอร์ตนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริง ๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Strobe ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่าการส่งข้อมูลใหม่มาที่ขา Data แล้วจากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy หรือเพื่อบอกว่าเครื่องพิมพ์ไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว



รูปที่ 2.1 ไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้ว ส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่น ๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างด้วยกัน คือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามาและตอบสนอง ต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ บางครั้งอาจเกิดเหตุการณ์ไม่ปกติเช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม (เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลมากกว่านี้ได้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์ คือ สัญญาณ Busy และเมื่อเครื่องเกิดข้อผิดพลาด เช่น กระดาษติดเครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error

นอกจากนี้ เมื่อคอมพิวเตอร์ต้องการรีเซ็ตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณ Reset ไปยังเครื่องคอมพิวเตอร์เพื่อรีเซ็ตเครื่องพิมพ์ด้วย สามารถสรุปหาสัญญาณที่จำเป็นสำหรับการติดต่อ ดังในตารางที่ 2.1

ตารางที่ 2.1 สัญญาณสำคัญ ๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซ็ตเครื่องพิมพ์	คอมพิวเตอร์

จากตารางที่ 2.1 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตอีก 1 พอร์ตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงาน ดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่าง ๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

บล็อกไดอะแกรมในรูปที่ 2.2 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อกับพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเนกเตอร์แบบ DB-25 สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนานจะมีมาพร้อมกับเมนบอร์ด ไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตรฐาน (SPP) อยู่

เมื่อดูจากรูปที่ 2.1 เทียบการทำงานโดยทั่วไป กับการเชื่อมต่อผ่านการ์ดที่เสียบลงในสล롯ของคอมพิวเตอร์แล้ว พอร์ตขนานจะมีลักษณะใกล้เคียงกัน โดยการติดต่อกับพอร์ตขนาน จะต้องมีการอ้างแอดเดรสตำแหน่งแอดเดรสที่ใช้อ้างถึง จะเป็นตำแหน่ง A0-A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องการอ่านหรือเขียนรีจิสเตอร์ตัวใด จากการตีโค้ดแอดเดรส A0-A9 นี้เอง ทำให้ได้สัญญาณออกมาเพื่อไปควบคุมหรืออินาเบิลวงจรมัลติเพล็กซ์ต่าง ๆ ดังนี้

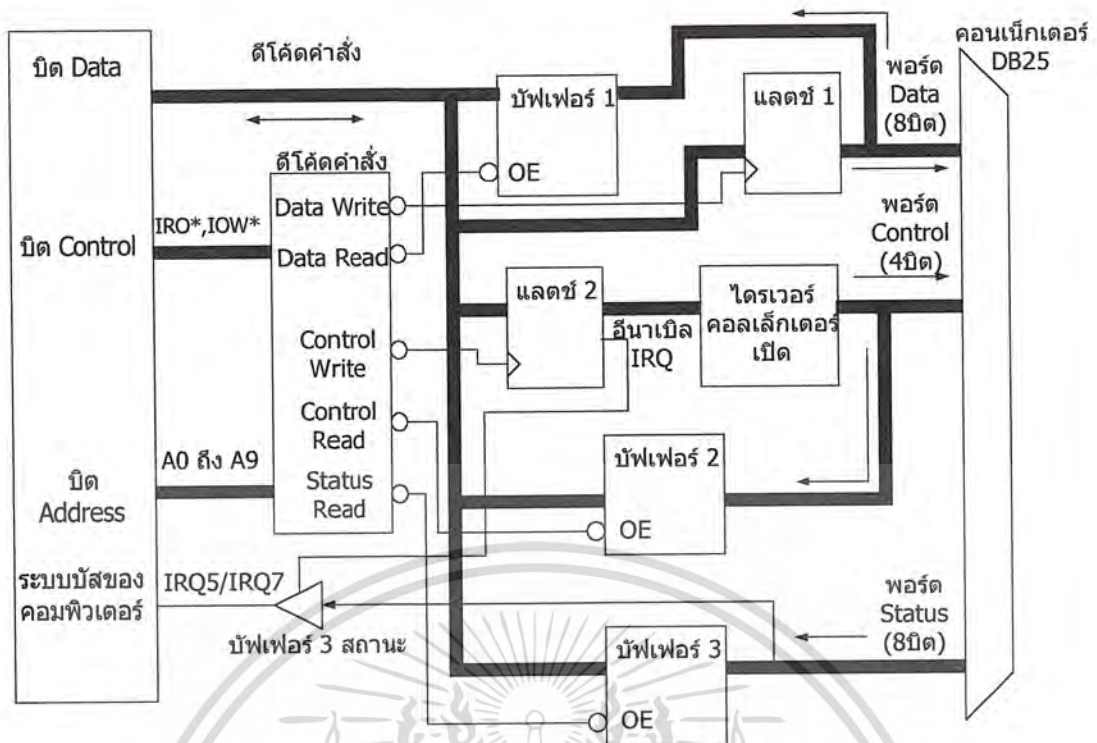
DataWrite สัญญาณอินาเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Data ของพอร์ตขนาน

DataRead สัญญาณอินาเบิลสำหรับอ่านข้อมูลจากขา Data ของพอร์ตขนานมาเก็บไว้ในบัส Data

ControlWrite สัญญาณอินาเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Control ของพอร์ตขนาน สำหรับพอร์ตนั้นนอกจากจะส่งข้อมูลออกไปยังพอร์ตขนานแล้ว ยังทำหน้าที่อินาเบิลการ Interrupt ของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

ControlRead สัญญาณอินาเบิลสำหรับอ่านค่าข้อมูลจากขา Control มาเก็บไว้ในบัส Data

StatusRead สัญญาณอินาเบิลสำหรับอ่านค่าข้อมูลจากขาพอร์ต Status มาเก็บไว้ในบัส Data



รูปที่ 2.2 แสดงระบบบัสดภายในของพอร์ตขนาน

ตารางที่ 2.2 แสดงชื่อและหน้าที่การทำงานของตำแหน่งขาต่าง ๆ บนพอร์ตขนาน ส่วนในตารางที่ 2.3 แสดงแอดเดรสของพอร์ตขนาน ซึ่งกำหนดไว้ 3 ตำแหน่งคือ LPT1, LPT2 และ LPT3

ตารางที่ 2.2 แสดงรายละเอียดของสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
1	Control	Out	-C0	nSTROBE	แอกทีฟ "0" ส่งค่าออกไป เพื่อบอกว่าที่ขาดาค่ามีข้อมูลแล้ว
2-9	Data	Out	D1-D8	DATA1-DATA8	สำหรับพอร์ตขนานมาตรฐาน เดิมขานี้ทำหน้าที่เป็นขาส่งข้อมูลเอาต์พุตเท่านั้น สำหรับในปัจจุบันขานี้รับข้อมูลเอาต์พุตได้ด้วย
10	Status	In	S6	nACK	เป็นพัลส์ลอคจิก "0" ที่ส่งมาจากเครื่องพิมพ์ เพื่อบอกว่าได้รับข้อมูลที่ส่งไปแล้ว
11	Status	In	-S7	BUSY	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่า ยังไม่พร้อมรับข้อมูล
12	Status	In	S5	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	-C1	AUTO FEED	สั่งเครื่องพิมพ์ให้เลื่อนบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 (ต่อ) แสดงรายละเอียดของสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

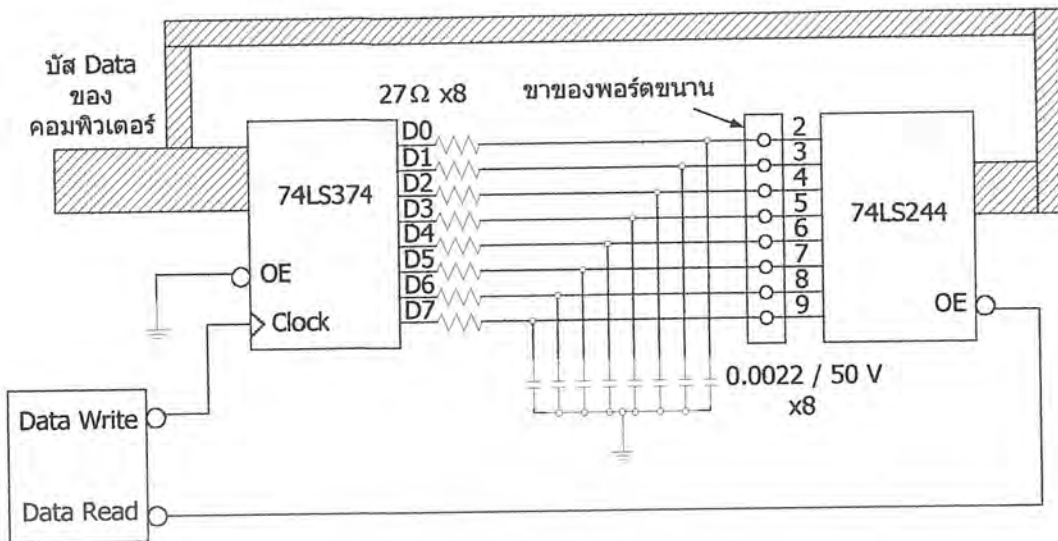
DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
15	Status	In	S3	nERROR	สัญญาณจากเครื่องพิมพ์มายังคอมพิวเตอร์ เพื่อแสดงข้อผิดพลาดจากการพิมพ์
16	Control	Out	C2	nINIT	รีเซ็ตเครื่องพิมพ์โดยให้ลอจิก "0"
17	Control	Out	-C3	nSELECT-IN	ส่งสัญญาณไปยังเครื่องพิมพ์เพื่อแจ้งว่า ต้องการเลือกเครื่องพิมพ์เครื่องนี้
18-25				GND	กราวด์

ตารางที่ 2.3 แสดงแอดเดรสของพอร์ตขนาน

ชื่อพอร์ต	LPT1		LPT2		LPT3	
	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
DATA	888	378H	956	3BCH	632	278H
STATUS	889	379H	957	3BDH	633	279H
CONTROL	890	37AH	958	3BEH	634	27AH

2.1.2 พอร์ตดาต้า (Data Port)

จากรูปที่ 2.3 แสดงให้เห็นว่าพอร์ต Data ประกอบไปด้วยบัสเพอร์ 1 ตัว และไอซีแลตซ์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตซ์ 1 ทั้ง 8 บิต เอาดัทพุตของไอซีแลตซ์ 1 คือ D0-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัสเพอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้



รูปที่ 2.3 วงจรภายในของพอร์ต Data

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตซ์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานค่า 27Ω และตัวเก็บประจุ $0.0022 \mu\text{F}$ เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก "0" เป็นลอจิก "1" หรือจากลอจิก "1" เป็นลอจิก "0" เป็นไปอย่างช้า ๆ เนื่องจากการเปลี่ยนแรงดันที่รวดเร็วทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่น ๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทาน และตัวเก็บประจุในวงจร ทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในรูปที่ 3 ทำให้เอาต์พุตของพอร์ต Data มีคุณสมบัติดังนี้

- กระแสซิงก์สูงสุด 24 มิลลิแอมป์
- กระแสซอร์สสูงสุด 2.6 มิลลิแอมป์
- ระดับแรงดันของลอจิก "1" ต่ำสุดเท่ากับ 2.4 โวลต์
- ระดับแรงดันสูงสุดสำหรับลอจิก "0" เท่ากับ 0.5 โวลต์

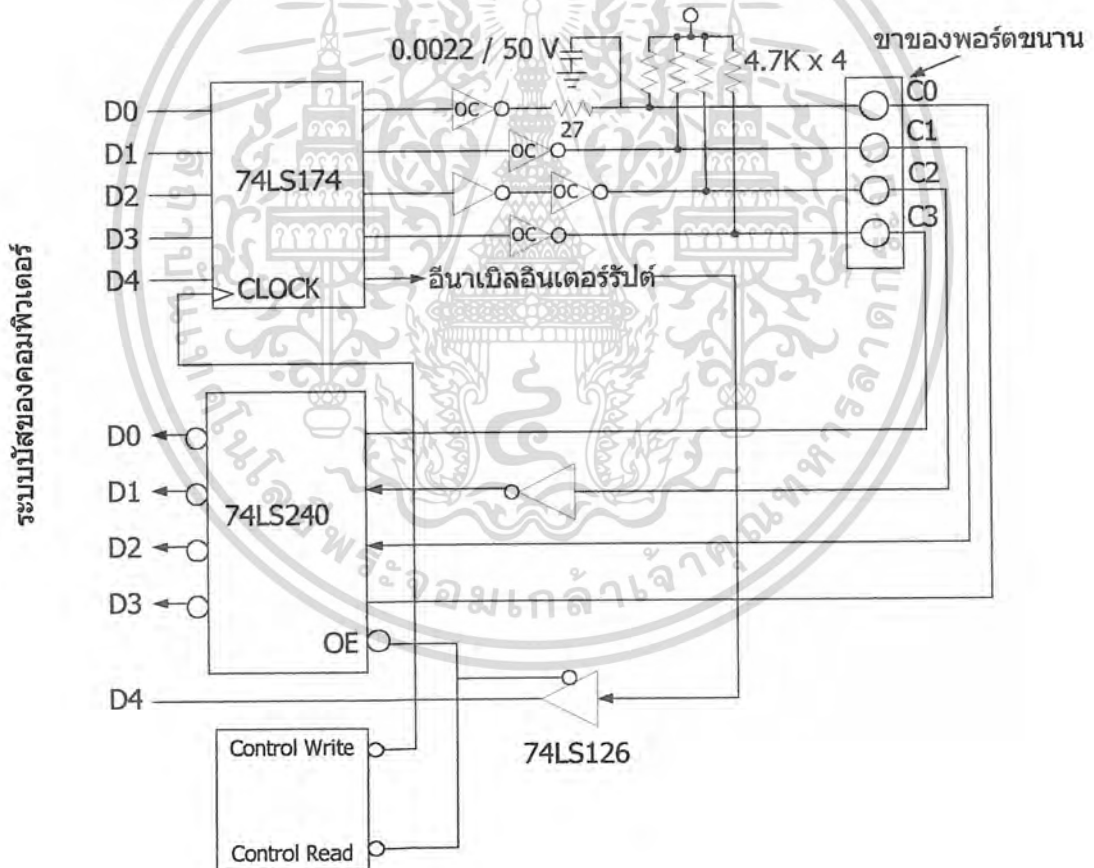
สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับ ได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Data Read ออกมา เพื่ออีนาเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน พอร์ต Data จะต้องใช้เพื่อการส่งค่าออกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทางสามารถอ่านค่าจากพอร์ต Data ได้ด้วยแต่ก่อนที่จะอ่านค่าต้องจำไว้เสมอว่า จะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก "1" ทั้งหมดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 พอร์ต Control

พอร์ต Control ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ จากตารางที่ 2.2 จะเห็นว่าพอร์ต Control ประกอบไปด้วยบิตเอาต์พุต 4 บิต ที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอินพุตอินเตอร์รัปต์ไม่ได้ถูกต่อออกไป รูปที่ 2.4 แสดงบล็อกไดอะแกรมของพอร์ต Control เอาต์พุตของพอร์ต Control มีอินเวอร์เตอร์แบบคอลเล็ก-เตอร์เปิดต่อรวมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัพไว้ด้วยตัวต้านทานค่า $4.7\text{ k}\Omega$ สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัวทำให้เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

สถานะของพอร์ต Control สามารถอ่านกลับได้โดยการใช้อินเวอร์เตอร์เบอร์ 74LS240 ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ต Control คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read



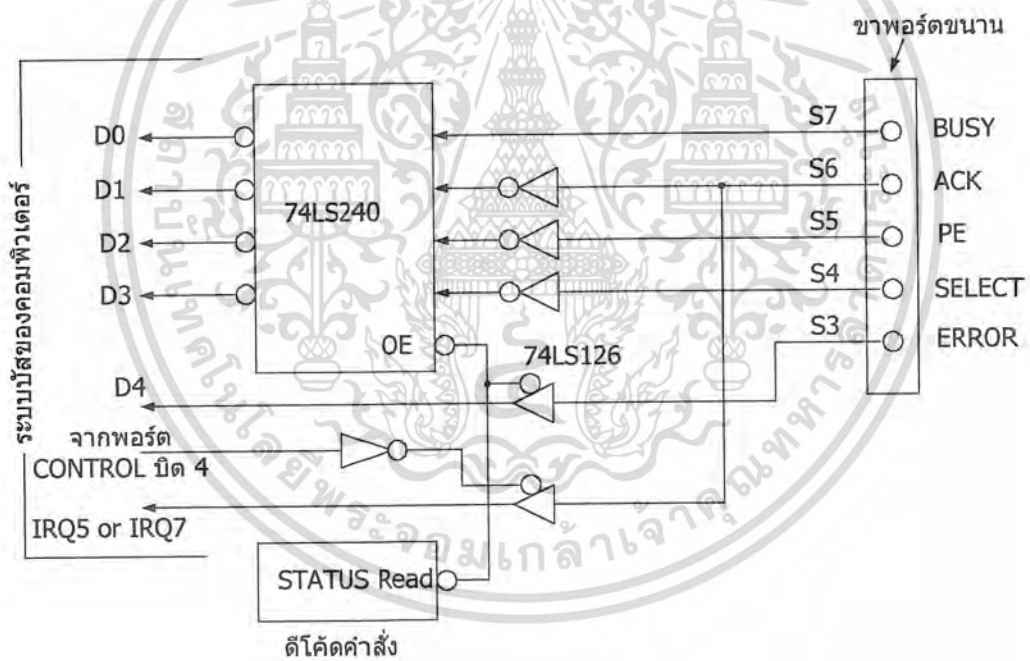
รูปที่ 2.4 วงจรภายในของพอร์ต Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเอาต์พุตของพอร์ต Control เป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่ามีลอจิก "1" เสียก่อน

2.1.4 พอร์ตแสดงสถานะ (Status Port)

พอร์ต Status เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ รูปที่ 2.5 แสดงรายละเอียดภายในของพอร์ต Status จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกันและจะเรียกชื่อเป็น S3, S4, S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Status นั้นเอง สำหรับบิต S7 จะมีข้อแตกต่างจากบิตอื่น ๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ในขณะที่ขาอื่น ๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลผ่านจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้ในการใช้งาน ถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ต Control บิต 4



รูปที่ 2.5 แสดงวงจรภายในของพอร์ตแสดงสถานะหรือพอร์ต Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐาน ผู้ใช้งานสามารถนำพอร์ตอินพุต 5 บิต (พอร์ต Status) พอร์ตเอาต์พุต 4 บิต (พอร์ต Control) และพอร์ตเอาต์พุตอีก 8 บิต (พอร์ต Data) ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ต Control นั้น สามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุตขนาด 4 บิตได้ด้วย ดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

2.1.6 การติดต่ออุปกรณ์อินพุตเอาต์พุตอย่างง่ายของพอร์ตขนาน

รีจิสเตอร์ของพอร์ตขนานมีทั้งสิ้น 3 รีจิสเตอร์ คือ

1. รีจิสเตอร์ Data ทำหน้าที่เป็นเอาต์พุต
2. รีจิสเตอร์ Status ทำหน้าที่เป็นอินพุต
3. รีจิสเตอร์ Control ทำหน้าที่เป็นเอาต์พุต

ดังนั้นถ้าผู้ใช้งานต้องการส่งค่าออกเอาต์พุตก็ต้องใช้รีจิสเตอร์ Data หรือรีจิสเตอร์ Control และถ้าต้องการรับค่าจากอินพุต ผู้ใช้งานต้องใช้รีจิสเตอร์ Status ในการอ่านค่าอินพุต การจะอ้างถึงรีจิสเตอร์แต่ละตัวนั้น ผู้ใช้งานจะต้องใช้ตำแหน่งแอดเดรสเป็นตัวอ้าง ตามรายละเอียดแสดงในตารางที่ 2.3

2.2 ความรู้เบื้องต้นเกี่ยวกับเซิร์ฟเล็ต (Servlet)

2.2.1 บทนำ

ในปัจจุบันมาตรฐานการแสดงผลทางด้านเว็บไคลเอนต์นั้นคือ HTML (Hyper Text Markup Language) HTML นั้นนอกจากเป็นภาษาที่เข้าใจได้ไม่ยากแล้วนั้น ยังสนับสนุนการทำงานแบบไคลเอนต์-เซิร์ฟเวอร์ ได้เป็นอย่างดี โดยการสร้างฟอร์ม HTML ในเว็บเพจเพื่อให้ผู้ใช้ดาวน์โหลดไปที่เครื่องไคลเอนต์เพื่อกรอกข้อมูลต่าง ๆ แล้วส่งกลับมาประมวลผลที่เซิร์ฟเวอร์ได้ ซึ่งการรับส่งข้อมูลระหว่างเว็บเบราว์เซอร์ และเว็บเซิร์ฟเวอร์เช่นนี้มักใช้ CGI(Common Gateway Interface) และภาษาที่นักพัฒนาส่วนใหญ่ใช้ทำการโปรแกรม CGI คือ Perl ปัญหาที่มักพบในการสร้างโปรแกรมสำหรับการใช้ CGI คือ โปรแกรมที่สร้างขึ้นต้องการกำลังเครื่องมากในการทำงานทั้งนี้เพราะ CGI เหล่านั้นจะแตกตัวเป็นโพรเซสย่อย ๆ ตามจำนวนไคลเอนต์ที่ติดต่อมายังเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหานี้ไม่ได้เป็นปัญหาใหม่ หากแต่ว่าในอดีตเหล่าเว็บเซิร์ฟเวอร์นั้นมักใช้ระบบปฏิบัติการยูนิกซ์ซึ่งเก่งในเรื่องจัดการโพสและเรดเป็นอย่างดี ในปัจจุบันเรามีการใช้งานวินโดวส์เอ็นทีเป็นเว็บเซิร์ฟเวอร์มากขึ้นทำให้นักพัฒนาต้องหาวิธีการใหม่ ๆ มาช่วยเพื่อลดโพสเรดในการทำไดนามิกคอนเทนต์ไม่ว่าจะเป็น ISAPI, NSAPI, ASP หรือ Fase-CGI

2.2.2 เซิร์ฟเล็ตคืออะไร

จาวาเซิร์ฟเล็ต คือ จาวาออบเจกต์ที่เราสร้างขึ้นตามข้อกำหนดของการสร้างเซิร์ฟเล็ตซึ่งสามารถนำเซิร์ฟเล็ตนั้นไปใส่ในเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงาน เซิร์ฟเล็ตเหล่านั้นสามารถเรียกให้ทำงานได้โดยการเรียกใช้จากเว็บเบราว์เซอร์นั่นเองข้อดีของการใช้งานเซิร์ฟเล็ตคือประสิทธิภาพที่เพิ่มขึ้น และความสามารถในการทำงานข้ามแพลตฟอร์มโดยที่ไม่จำเป็นต้องคอมไพล์โปรแกรมเซิร์ฟเล็ตใหม่ นอกจากนั้นจาวาเซิร์ฟเล็ตยังเหมาะสมกับการสร้างโปรแกรมที่มีความสลับซับซ้อน เนื่องจากภาษาที่ใช้สร้าง คือ ภาษาจาวานั้นเอง

ในด้านความปลอดภัย จาวาถือเป็นภาษาที่มีการควบคุมการทำงานของโปรแกรมด้วยเซ็กเคียวริตี้แมนเนเจอร์ (Security manager) ตัวอย่างเช่น เราสามารถกำหนดให้เซิร์ฟเล็ตเปิดไฟล์เฉพาะในไดเรกทอรีที่กำหนดได้ และถ้าหากเซิร์ฟเล็ตนั้นพยายามเขียนหรืออ่านไฟล์ที่ไม่ได้รับอนุญาต จาวาก็จะหยุดการทำงานของเซิร์ฟเล็ตนั้นให้โดยที่ระบบยังคงทำงานต่อไปได้โดยไม่ติดขัด ดังนั้นเซิร์ฟเล็ตจึงคงคุณสมบัติทางด้านความปลอดภัยของจาวาไปอย่างเต็มเปี่ยม

การทำงานของเซิร์ฟเล็ตเหมือนกับการทำงานของ CGI กล่าวคือ สนับสนุนคำสั่ง GET, POST, PUT และ DELETE เหมือนการทำงานของ CGI แต่ความแตกต่างสำคัญระหว่างการเขียนโปรแกรมด้วย CGI และ เซิร์ฟเล็ต คือ

จาวาเซิร์ฟเล็ต

- ต้องเขียนด้วยภาษาจาวา
- โปรแกรมแบบเซิร์ฟเล็ตต้องทำงานเป็นเรดหนึ่งของเว็บเซิร์ฟเวอร์
- มีระบบรักษาความปลอดภัยสูง โดยมีเซ็กเคียวริตี้แมนเนเจอร์คอยตรวจสอบการใช้ทรัพยากร หากพบการเข้าถึงทรัพยากรที่มีได้รับอนุญาตระบบจะหยุดการทำงานของเซิร์ฟเล็ตนั้นทันทีอย่างอัตโนมัติ

CGI (Common Gateway Interface)

- สามารถเขียนด้วยภาษาใดก็ได้
- โปรแกรม CGI ถูกเรียกใช้ไปอีกโพสเรดหนึ่งแยกออกมาจากตัวเว็บเซิร์ฟเวอร์
- ต้องระมัดระวังในการเขียนโปรแกรมที่ใช้ทรัพยากรของระบบ เพราะโปรแกรมสามารถเข้าถึงทรัพยากรของระบบได้โดย (ค่อนข้าง) อิสระ ความผิดพลาดของโปรแกรมสามารถทำให้ระบบโดยรวมทำงานผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเล็ตถูกออกแบบมาให้ทำงาน เพื่อตอบสนองการร้องขอของไคลเอนต์เช่นเดียวกับ CGI ซึ่งเซิร์ฟเล็ตจะถูกเขียนขึ้นมาจากจาวาโค้ด ดังนั้นลักษณะพิเศษของ เซิร์ฟเล็ตจึงเป็นโปรแกรมที่รองรับการทำงานแบบ Cross-Platform และสามารถรันนิ่งได้ในเว็บเซิร์ฟเวอร์ทุก ๆ ระบบ

สิ่งหนึ่งที่ทำให้เซิร์ฟเล็ตแตกต่างจากแอปเพล็ตก็คือ แอปเพล็ตนั้นถึงแม้จะถูกโค้ดติดตั้งขึ้นมาตามไวยากรณ์ของจาวาแต่ก็ยังใช้ได้กับระบบปฏิบัติการและบราวเซอร์บางตัวเท่านั้น ส่วนเซิร์ฟเล็ตจะถูกรันนิ่งเฉพาะบนเว็บเซิร์ฟเวอร์และรีทีร์นเฉพาะค่าต่าง ๆ ให้แก่ไคลเอนต์ จึงไม่ผูกติดกับระบบปฏิบัติการหรือบราวเซอร์ใด ๆ ดังสโลแกนที่ว่า "write once, run anywhere" นอกจากนี้ เซิร์ฟเล็ตเทคโนโลยียังถูกเตรียมขึ้นมาเพื่อสนับสนุนการทำงานของ JSP (Java Server Page) ให้เป็นไปได้อย่างเต็มที่ยิ่งขึ้น

การใช้งานเซิร์ฟเล็ตไม่ยากไปกว่าเครื่องมือในการพัฒนาระบบตัวอื่น ๆ เลย เนื่องจากสามารถเขียนเซิร์ฟเล็ตมาได้เหมือน ๆ กับที่เคยเขียนภาษาเพิร์ล หรือ ซีพลัสพลัส นอกจากนี้ตัวของเซิร์ฟเล็ตเองก็ยังไม่ต้องการเอนไวรอนเมนต์อื่นใดมากไปกว่าสคริปต์ตัวอื่น ๆ เพียงแค่มี HTTP Server ก็เพียงพอแล้วสำหรับการใช้เซิร์ฟเล็ต ปัจจุบันเซิร์ฟเล็ตเริ่มเข้ามาแทนที่คำว่า "CGI" อย่างช้า ๆ และสมบูรณ์แบบจนมีการใช้เซิร์ฟเล็ตเป็นมาตรฐานสำหรับ เซิร์ฟเวอร์เซิร์ทสคริปติง การที่จะทำให้เซิร์ฟเล็ต สามารถทำงานได้เต็มประสิทธิภาพนั้น ควรจะใช้งานเซิร์ฟเล็ตอยู่บน Java-Base Webserver เป็นดีที่สุด เนื่องจากเอนจินต่าง ๆ ที่สนับสนุนจาวานั้น ๆ จะถูกจัดเตรียมเอาไว้ อยู่แล้ว แต่สำหรับผู้ที่ไม่ได้มี Java-Base Webserver ก็สามารถหาซอฟต์แวร์จาก Third-Party ทำให้เว็บเซิร์ฟเวอร์ต่าง ๆ สามารถใช้เซิร์ฟเล็ตได้แล้ว โดยซอฟต์แวร์ดังกล่าวมีชื่อว่า "ServletExec" หลังจากที่ติดตั้งServletExec แล้วไม่ว่าเว็บเซิร์ฟเวอร์จะเป็น Microsoft IIS, Netscape Enterprise Server, Apache Server หรือ McOS WebServer ก็จะสามารถรันนิ่งเซิร์ฟเล็ตได้เช่นกัน

สำหรับโปรแกรม ServletExec นั้นสามารถดาวน์โหลดมาใช้ได้จาก

<http://www.newatlanta.com/downloads.html>

โดยปกติเซิร์ฟเล็ตจะถูกใช้งานในรูปแบบของ "Java Standard Extension Classes" ซึ่งอยู่ในแพ็คเกจที่ชื่อ "javax.servlet" และ "javax.servlet.http" โดยที่

javax.servlet เป็น Basic Framework ของเซิร์ฟเล็ต

javax.servlet.http เป็น Extension ของ Basic Framework ที่มีไว้เพื่อตอบสนองการร้องขอผ่านโปรโตคอล HTTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการใช้งานเซิร์ฟเล็ตนั้นจะประกอบด้วย

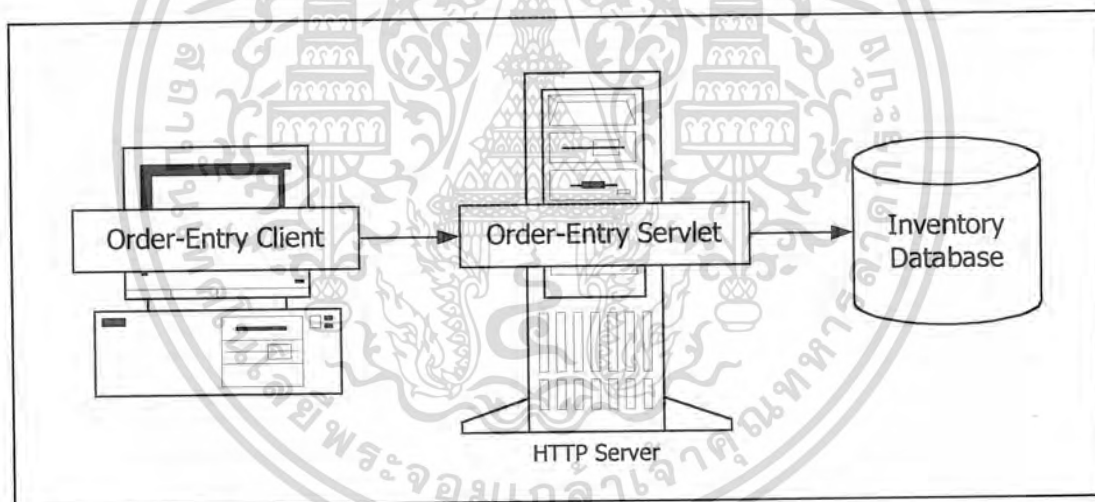
1. การประมวลผลหรือเก็บค่าที่ได้รับมาจาก HTML Form
2. การจัดเตรียมองค์ประกอบแบบ Dynamic หรือการรีเทิร์นค่าที่ได้จากการรันคิวรีบนฐานข้อมูล
3. การจัดการกับข้อมูลที่จะส่งออกไป

ด้วยเหตุที่เซิร์ฟเล็ตเป็นโมดูลของจาวาที่กระทำผ่าน HTTP Protocol ดังนั้นการพูดถึง HTTP Servlet ก็หมายถึงตัวของเซิร์ฟเล็ตนั่นเอง

2.2.3 สถาปัตยกรรมพื้นฐานของเซิร์ฟเล็ต

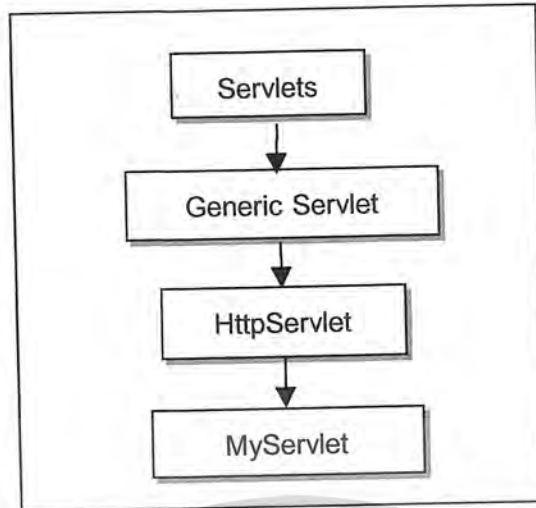
รูปแบบทั่วไปของเซิร์ฟเล็ตจะถูกใช้งานผ่าน Instance Class ที่ได้รับการอิมพลีเมนต์มาเป็นอย่างดีแล้วส่วนที่เป็นอินเตอร์เฟสคือส่วนที่เรียกว่า "javax.servlet.Servlet" ซึ่งบางครั้งเราจะเรียกส่วนนี้ว่า "javax.servlet."

"GenericServlet" นอกจากส่วนที่มีเอาไว้เพื่อติดต่อกับเซิร์ฟเล็ตดังกล่าวแล้ว ส่วนที่ใช้ตอบสนองการรีเควสต์ "javax.servlet.http.HttpServlet" ก็มีความจำเป็นต่อการทำงานของโปรแกรมด้วย



รูปที่ 2.6 สถาปัตยกรรมพื้นฐานของเซิร์ฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดง API ของเซิร์ฟเล็ต

ในการ Initial เซิร์ฟเล็ตนั้น จะเริ่มจากการที่เซิร์ฟเวอร์แอปพลิเคชันทำการโหลดเซิร์ฟเล็ตคลาส(หรือคลาสอื่น ๆ ที่อาจได้รับการอ้างอิงถึง) หลังจากนั้นจึงทำการเรียกส่วนที่เป็น Instance Item ขึ้นมา เมื่อมีส่วนดังกล่าวแล้วจึงจะทำการเรียก Method ที่มีไว้เก็บคอนฟิกูเรชันของเซิร์ฟเล็ตเอาไว้ออกมา

Method ดังกล่าวก็คือ init (ServletConfig config) จะเห็นได้ว่าเซิร์ฟเวอร์ต้องอาศัยระยะเวลาช่วงหนึ่งในการจัดเตรียมองค์ประกอบพื้นฐานผ่านทางส่วนที่ทำงานในลักษณะของ Procedure แล้วทำการเก็บค่า (Store) คอนฟิกูเรชันนั้นเอาไว้ เราจะเรียกสิ่ง que เก็บเอาไว้นี้ว่า ServletConfig Object ค่าดังกล่าวจะถูกเรียกใช้งานภายหลังโดย Method ที่ชื่อ getServletConfig() ซึ่งเป็นสับคลาสของ GenericServlet

ในขณะที่เดียวกันนี้ GenericServlet จะทำการเรียก super.init(config) ขึ้นมาเพื่อใช้งานคุณสมบัติต่าง ๆ ที่จำเป็นต่อ Initial Method ภายในของ ServletConfig Object จะมีพารามิเตอร์ต่าง ๆ ของเซิร์ฟเล็ตและส่วนอ้างอิงที่เกี่ยวข้องของ (ServletContext.) บรรจุอยู่ ซึ่งถ้าหากกระบวนการ Initial Method นี้สำเร็จลงเซิร์ฟเล็ตจึงจะยอมให้ทำการเรียกส่วนที่เรียกว่า "Service Method" ขึ้นมาทำงานต่อ

หลังจากการ Initial พร้อมแล้ว ส่วนที่มีไว้สนับสนุนการบริการก็จะถูกเรียกขึ้นมาเพื่อเตรียมรับการ request สับคลาสของส่วนนี้ก็คือ service (Servlet Request req, ServletResponse Res) Method ซึ่งจะถูกรเรียกใช้ทุก ๆ ครั้งที่มีการ Request

หลังจากที่ประกอบกิจกรรมต่าง ๆ และไม่ต้องการเรียกใช้ (Unloaded) เซิร์ฟเล็ตแล้ว ต้องมีการเรียก Method ที่ชื่อว่า destroy() เพื่อทำการระงับใช้เซอวิสของ Servlet ซึ่ง destroy method จะเป็นตัวที่บอกให้ทราบว่าควรจะสิ้นสุดแต่ละ Life Cycle ของเซิร์ฟเล็ตเมื่อไร

ก่อนที่จะใช้งานเซิร์ฟเล็ตนั้น เราควรที่จะต้องรู้พื้นฐานการทำงานของ HTTP Protocol สำหรับการใช้งานเซิร์ฟเล็ตเสียก่อน ซึ่งโปรโตคอลนี้จะถูกใช้โดย WWW Client (หรือที่รู้จักกันในชื่อของ "Browser") ส่งคำร้องขอมายังเว็บเซิร์ฟเวอร์ ภายใน HTTP จึงมีทั้งส่วนที่ใช้ในการ Request และส่วนที่ใช้ในการ Response

HTTP Request ประกอบด้วย

- Request Method
- URI
- Header Field
- ส่วนที่เป็น body

HTTP Response ประกอบด้วย

- ผลลัพธ์
- Header Field
- ส่วนที่เป็น body

เซอวิสของ HTTP Servlet นั้นจะขึ้นอยู่กับว่าทางฝั่งไคลเอนต์จะส่ง HTTP Request Method อะไรเข้ามาที่เว็บเซิร์ฟเวอร์ โดยปกติแล้ว Method เหล่านั้นจะต้องเป็นไปตามมาตรฐาน HTTP/1.1 Method ต่าง ๆ ที่คุณจะใช้ควรจะเป็นสิ่งที่มีอยู่ในสับคลาสที่ได้รับการ Implement มาแล้ว เช่น GET, HEAD, PUT, POST, DELETE, OPTIONS และ TRACE ไม่อย่างนั้นขณะที่ฝั่งไคลเอนต์รีควีสต์ไปแล้วอาจได้รับการปฏิเสธจากเซิร์ฟเวอร์เ้าว่า "Bad Request" หรือ "HTTP Error" กลับมา

2.2.4 เนื้อหาในส่วนของภาษาจาวาที่เกี่ยวข้องกับโปรแกรม

2.2.4.1 การเรียกใช้เมธอดในคลาส Runtime

คลาส Runtime มีเมธอดที่เกี่ยวกับสภาวะแวดล้อมการทำงานของโปรแกรมและ Java virtual machine คลาสนี้ไม่มี constructor เพราะโดยปกติเราจะไม่สร้าง instance ของคลาส Runtime แต่จะได้ reference ของ Runtime ปัจจุบันที่กำลังทำงานอยู่โดยการเรียก Runtime.getRuntime() ซึ่งเราจะใช้ reference นี้ในการเรียก เมธอดต่าง ๆ ที่กำหนดไว้ดังนี้

```

public class Runtime extends Object {
    public static Runtime getRuntime();
    public Process exec(String command) throws IOException;
    public Process exec(String command, String[] envp) throws IOException;
    public Process exec(String[] cmdarray) throws IOException;
    public Process exec(String[] cmdarray, String[] envp) throws IOException;
    public void exit(int status);
    public native long freeMemory();
    public native void gc();
    # public InputStream getLocalizedInputStream(InputStream in);
    # public OutputStream getLocalizedOutputStream(OutputStream out);
    public synchronized void load(String filename);
    public synchronized void loadLibrary(String libname);
    public native void runFinalization();
    1.1 public static void runFinalizersOnExit(boolean value);
    public native long totalMemory();
    public native void traceInstructions(boolean on);
    public native void traceMethodCalls(boolean on);
}

```

2.2.4.2 การเรียกใช้เมธอดทในคลาส Process

คลาส Process มีไว้สำหรับเป็นผลลัพธ์ของการเรียก exec() ของคลาส Runtime ที่แสดงไว้ในหัวข้อที่แล้ว เราไม่สามารถสร้าง instance ของคลาส Process ขึ้นได้โดยตรง แต่จะได้เป็น reference จากการเรียก exec() เท่านั้น โดย reference ที่ได้นี้จะถูกใช้ในการควบคุมหรือการรับและส่งค่าเอาต์พุตและอินพุตให้แก่โปรแกรมที่ถูกเรียกให้ทำงานคลาส Process มีเมธอดที่กำหนดไว้ดังนี้

```

public abstract class Process extends Object {
    public abstract void destroy();
    public abstract int exitValue();
    public abstract InputStream getErrorStream();
    public abstract InputStream getInputStream();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public abstract OutputStream getOutputStream();
public abstract int waitFor() throws InterruptedException;
}

```

2.2.4.3 การใช้เอ็กเซพชัน (Exceptions) ในการบ่งบอกข้อผิดพลาด

จาวาใช้ Exception ในการบ่งถึงข้อผิดพลาดเมื่อรันโปรแกรม แต่ในบางครั้ง exception เป็นการแจ้งถึงเหตุการณ์พิเศษที่โปรแกรม

ถ้าต้องการพยายามที่จะจัดการกับข้อผิดพลาดทุกอย่างที่อาจเกิดขึ้นในโปรแกรม โครงสร้างของโปรแกรมก็จะซับซ้อนมากและยากต่อการเข้าใจ ข้อดีของ exception คือมีการแยกโค้ดที่จัดการกับความผิดพลาดออกจากโค้ดที่ทำงานเมื่อไม่มีข้อผิดพลาดเกิดขึ้น ข้อดีอีกข้อหนึ่งก็คือ มีการเตรียมการเป็นพิเศษเพื่อจัดการกับ exception ชนิดต่าง ๆ โปรแกรม จะต้องจัดการกับ exception นั้น ๆ มิฉะนั้นโปรแกรมก็จะไม่สามารถคอมไพล์ได้

Exception — ในจาวาก็คือวัตถุซึ่งสร้างขึ้นเมื่อมีเหตุการณ์ผิดปกติเกิดขึ้นในโปรแกรม วัตถุนี้จะมีข้อมูลเกี่ยวกับธรรมชาติของปัญหา exception จะถูก thrown นั่นก็คือ วัตถุที่บ่งถึงรายละเอียดเกี่ยวกับ exception จะถูกส่งไปเป็นอาร์กิวเมนต์ไปให้ส่วนของโปรแกรมที่รับผิดชอบในการจัดการกับปัญหานั้น ส่วนของโปรแกรมที่รับวัตถุของ exception เข้ามานั้น เราเรียกว่า catch

1) ชนิดของ Exception

Exception ก็คือ วัตถุของคลาสที่เป็นสับคลาสของคลาส Throwable ทุก ๆ exception มาตรฐานจะรวมอยู่ในคลาส Error และคลาส Exception

Error Exceptions เป็น exceptions ที่กำหนดอยู่ในคลาส Error หรือสับคลาสของมันซึ่งจะรวมกลุ่มตามข้อเท็จจริงของการเกิดข้อผิดพลาด exception เหล่านี้เป็นความผิดพลาดร้ายแรงที่เกี่ยวข้องกับระบบเป็น exception ที่ไม่สามารถแก้ไขได้ ดังนั้นเราไม่ควรที่จะจับ (catch) exception พวกนี้

RuntimeException Exceptions ควรที่จะจับ exceptions ซึ่งเป็นวัตถุของคลาสที่เป็นสับคลาสของคลาส Exception สับคลาสที่สำคัญของคลาส Exception คือ RuntimeException Exception ซึ่งอยู่ในสับคลาสของคลาส RuntimeException จะแตกต่างจากสับคลาสอื่นของ exception คือ คอมไพเลอร์จะไม่มีการตรวจสอบว่าโปรแกรมได้เตรียมวิธีจัดการกับ exception พวกนี้หรือไม่ โดยทั่วไปถ้าเป็น exception โปรแกรมจะเตรียมวิธีจัดการกับมัน หรือลงทะเบียนว่าเมธอดอาจมีการ throw exception ถ้าไม่ทำดังกล่าวโปรแกรมจะไม่สามารถคอมไพล์ได้ สับคลาสของคลาส RuntimeException มีดังตารางที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติว่ามีเมธอดที่ซึ่งอาจก่อให้เกิด exception ซึ่งไม่เป็น subclass ของ RuntimeException และ Error เช่น IOException เมื่อเมธอดเกี่ยวข้องกับการอ่านข้อมูลจากแฟ้มหรือการบันทึกข้อมูลของแฟ้ม ถ้า exception ไม่มีการจับเมธอด จะต้องทำการประกาศว่า exception ดังกล่าวจะมีการ thrown ซึ่งจะทำให้ได้ด้วยวิธีง่าย ๆ โดยการเพิ่มประโยค thrown ลงในหัวเมธอด

ตารางที่ 2.4 subclass ต่าง ๆ ในคลาส RuntimeException

ชื่อคลาส	ความหมายของ exception
ArithmeticException	เกิดเงื่อนไขที่ไม่ถูกต้องทางคณิตศาสตร์ เช่นหารด้วยศูนย์
ArrayStoreException	การพยายามที่จะเก็บข้อมูลในอาร์เรย์ โดยข้อมูลดังกล่าวมิใช่ข้อมูลชนิดที่ระบุไว้ในอาร์เรย์
ClassCastException	การพยายามที่จะแปลงวัตถุไปเป็นวัตถุชนิดที่ไม่ถูกต้อง
IllegalArgumentException	ส่งอาร์กิวเมนต์ไปยังเมธอด โดยอาร์กิวเมนต์ดังกล่าวไม่สอดคล้องกับชนิดของพารามิเตอร์ที่ประกาศไว้
IndexOutOfBoundsException	การพยายามที่จะอ้างถึงหน่วยของอาร์เรย์ที่มีอินเด็กซ์นอกขอบเขตที่กำหนดไว้
NegativeArraySizeException	กำหนดขนาดของอาร์เรย์ให้เป็นค่าลบ
NullPointerException	การอ้างถึงวัตถุที่เป็น null เมื่อมันควรอ้างอิงถึงวัตถุที่มีการกระทำบางอย่าง เช่นเรียกใช้เมธอดที่ไม่ได้กำหนดไว้หรือเรียกใช้วัตถุที่ยังไม่มีการกำหนดค่า
SecurityException	โปรแกรมมีการกระทำบางอย่างที่ไม่ถูกต้องตามหลักของความปลอดภัย เช่นให้แอปพลิเคชันอ่านข้อมูลในแฟ้มของเครื่องที่เรียกใช้แอปพลิเคชัน

2) การจัดการกับ exception

ถ้าต้องการจัดการกับ exception ณ ที่ที่เกิด จาว่ามีบล็อก 3 แบบที่สามารถเพิ่มเข้าไปในเมธอดที่ใช้จัดการกับ exception นั่นก็คือ try, catch และ finally บล็อก try บล็อก ใช้บรรจุโค้ดซึ่งอาจก่อให้เกิด exception catch บล็อก ใช้บรรจุโค้ดที่ใช้ในการจัดการ exception เหล่านั้น finally บล็อก ใช้บรรจุโค้ดที่ต้องการดำเนินการก่อนจะสิ้นสุดเมธอด โดยไม่คำนึงถึงว่า exception จะถูก throw ใน try บล็อกหรือไม่ก็ตาม

2.1) Try Block

เมื่อต้องการจับ exception โค้ดที่อาจก่อให้เกิด exception จะต้องบรรจุใน try บล็อก ถ้าโค้ดที่อาจก่อให้เกิด exception มิได้บรรจุใน try บล็อก ก็ไม่สามารถจับ exception ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
try { // โค้ดซึ่งอาจมีการ throw exception }
```

2.2) Catch Block

การบรรจุโค้ดที่ใช้จัดการกับ exception ใน catch บล็อก Catch บล็อกจะต้องต่อท้าย try บล็อก ใน catch บล็อก ประกอบด้วย keyword catch ตามด้วยพารามิเตอร์ระหว่างวงเล็บซึ่งบ่งถึงชนิดของ exception ที่บล็อกดำเนินการ และตามด้วยโค้ดที่ใช้จัดการกับ exception นั้น ๆ

```
catch(ArithmeticException e)
{ // โค้ดที่ใช้จัดการกับ exception e }
```

2.3) ประโยค Throws

ประโยค throws จะระบุรายการ exception ที่ถูก throw โดยเมธอด

```
int g(float h) throws a, b, c
{ // เนื้อหาของเมธอด }
```

ชนิดของ exception ซึ่งถูกโยนโดยเมธอดจะถูกระบุด้วยประโยค throws เมธอดสามารถโยนวัตถุของคลาสดังกล่าว หรือวัตถุของสับคลาส

2.4) Finally Block

โปรแกรมที่ใช้ทรัพยากรเฉพาะบางชนิดจะต้องมีการคืนทรัพยากรนั้น ๆ กลับคืนสู่ระบบเพื่อหลีกเลี่ยงการเกิดปัญหาทรัพยากรรั่วไหล (resource leak) ในภาษา C และ C++ ปัญหาที่เกิดขึ้นบ่อย คือ หน่วยความจำรั่วไหล (memory leak) จาว่าจะมีการดำเนินการเก็บรวบรวมหน่วยความจำที่ไม่ได้ถูกใช้แล้วคืนให้แก่ระบบเพื่อหลีกเลี่ยงปัญหาหน่วยความจำรั่วไหล แต่ก็อาจมีปัญหาทรัพยากรอื่น ๆ รั่วไหลได้

Finally บล็อกเป็นบล็อกที่เป็นตัวเลือก ถ้ามี finally บล็อกเราจะวางได้ต่อจาก try และ catch บล็อก จาว่าจะรับรองว่า finally บล็อกจะถูกดำเนินการไม่ว่าจะมีการ throws exception ใน try บล็อกหรือไม่

2.3 ความรู้เบื้องต้นในการติดต่อฐานข้อมูลผ่าน JDBC/ODBC bridge

2.3.1 JDBC คืออะไร

JDBC (Java Database Connectivity) คือ Java API ที่ใช้สำหรับตีความคำสั่ง SQL โดยประกอบด้วยชุดของ Classes และ Interfaces ที่เขียนด้วยโปรแกรมภาษา Java ตัว JDBC นี้เป็น API มาตรฐาน สำหรับการพัฒนาระบบฐานข้อมูล และทำให้มีความเป็นไปได้ในการเขียนโปรแกรมใช้งานเกี่ยวกับฐานข้อมูลด้วย Java API เพียงอย่างเดียว การใช้ตัว JDBC นี้ทำให้ง่ายในการส่งคำสั่ง SQL ให้กับระบบฐานข้อมูลชนิดต่าง ๆ ในรูปแบบเดียวกัน นั่นคือ JDBC API จะไม่บังคับให้ เขียนโปรแกรมตัวหนึ่งเพื่อใช้กับฐานข้อมูลของ Sybase และอีกโปรแกรมหนึ่งสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลของ Oracle และอีกหนึ่งโปรแกรมสำหรับ ฐานข้อมูลของ Informix และอีกหลายโปรแกรมสำหรับฐานข้อมูลชนิดต่างๆ เราสามารถเขียนโปรแกรมโดยใช้ JDBC API เพียงโปรแกรมเดียว และโปรแกรมนั้นจะสามารถส่งคำสั่ง SQL ไปยังฐานข้อมูลชนิดต่าง ๆ นั้นได้ ในการรวมตัวกันระหว่าง Java และ JDBC นี้ ทำให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมเพียงหนึ่งโปรแกรมแล้วสามารถนำไปใช้ได้ในทุก ๆ แห่งที่ต้องการฐานข้อมูลที่ได้ติดตั้งไว้ก่อนแล้ว และการเข้าถึงข้อมูลก็ทำได้ง่าย แม้ว่ามันจะอยู่ในระบบจัดการฐานข้อมูลต่างชนิดกันก็ตาม ในการพัฒนา

2.3.2 ความสามารถของ JDBC

ในการใช้งานทั่วไป JDBC สามารถทำได้ 3 อย่างดังนี้

1. สร้างการเชื่อมต่อกับฐานข้อมูล
2. ส่งคำสั่ง SQL
3. จัดการกับผลลัพธ์ที่ได้มา

ตัวอย่างข้างล่างนี้เป็นโปรแกรมที่ตัดมา เพื่อแสดงการทำงานทั้ง 3 อย่างข้างต้น ดังนี้

```

Connection con = DriverManager.getConnection ("jdbc:odbc:wombat", "login",
"password");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");
while (rs.next()) {
    int x = getInt("a");
    String s = getString("b");
    float f = getFloat("c");
}

```

2.3.3 ชนิดของ JDBC Driver

JDBC Drivers ที่เราทราบในเวลานี้มีอยู่ 4 แบบด้วยกันคือ

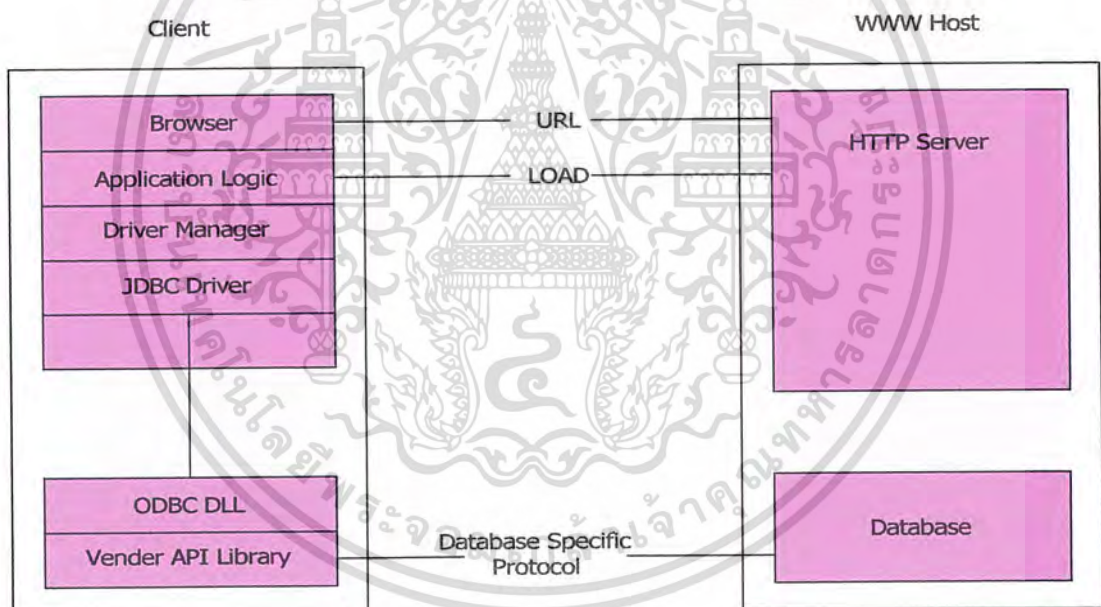
1. JDBC/ODBC Bridge.
2. Native-API, Partly Java Driver.
3. Network-Protocol, All-Java Driver.
4. Native-Protocol, All-Java Driver.

แต่ในที่นี้จะกล่าวเพียงแค่แบบที่ใช้งาน คือ แบบที่ 1 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. JDBC/ODBC Bridge (ชนิดที่ 1)

JDBC/ODBC Bridge ได้ถูกพัฒนาขึ้นโดยการร่วมมือกันระหว่าง JavaSoft และ Intersolv เพื่อที่จะเพิ่มความสามารถให้กับฐานข้อมูลจำนวนมากที่ใช้ ODBC ในส่วนของ Client, Java Applet หรือ โปรแกรมอื่นจะถูกเขียนขึ้นด้วย JDBC API โดย Bridge ตัวนี้จะทำการแปลงคำสั่งจาก JDBC ไปเป็น คำสั่งของ ODBC แล้วส่งคำสั่งนั้นไปยังตัว ODBC Driver เพื่อจัดการกับฐานข้อมูล ข้อได้เปรียบหลักของ Bridge ตัวนี้ คือ โปรแกรมที่เขียนขึ้นนั้นจะง่ายต่อการเข้าถึงข้อมูลจากระบบฐานข้อมูลของผู้ผลิตต่าง ๆ โดยการเลือก ODBC driver ที่เหมาะสม อย่างไรก็ตาม ตัวติดต่อกับฐานข้อมูลชนิดนี้ ทำให้ต้องนึกถึงผลข้างเคียงที่อาจเกิดขึ้นและความยุ่งยาก เพราะว่าคำสั่งต้องส่งจาก JDBC ไปยัง Bridge ที่เชื่อมไปยัง ODBC Driver และสุดท้ายจาก ODBC ก็ส่งไปยัง Native Client-API เพื่อไปยังฐานข้อมูล ตัว Driver ชนิดนี้ทำให้ Java Applets ส่งข้อมูลไปไม่ทันทีทันใดกับที่เรียกไป เพราะ Code หลัก ๆ ต้องถูกติดตั้งไว้ก่อนแล้วบนทุก ๆ เครื่องของ Client ที่ต้องการใช้การเชื่อมต่อด้วย JDBC/ODBC Bridge



รูปที่ 2.8 แสดงการโครงสร้างการติดต่อฐานข้อมูลโดยใช้ JDBC/ODBC Bridge

2.3.4 ข้อดี – ข้อเสีย ของ JDBC/ODBC Driver

ข้อดีของ JDBC/ODBC Driver

1. สามารถใช้ในการเข้าถึงฐานข้อมูลได้หลายรูปแบบ
2. โดยเลือกใช้ ODBC ที่เหมาะสมกับฐานข้อมูล นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของ JDBC/ODBC Driver

1. มีผลข้างเคียงที่อาจเกิดขึ้นและความยุ่งยากเพราะว่าคำสั่งต้องส่งจาก JDBC ไปยัง Bridge ที่เชื่อมไปยัง ODBC Driver และสุดท้ายจาก ODBC ก็ส่งไปยัง Native Client-API เพื่อไปยังฐานข้อมูล ทำให้การส่งข้อมูลทำได้ช้าหรือมีความเร็วในการส่งข้อมูลต่ำนั่นเอง

2. ชนิดที่ 1 นี้ไม่สามารถรองรับการทำงานแบบ just-in-time delivery ได้ เนื่องจาก Native Code ที่ใช้จะต้องมีการติดตั้งบนเครื่อง client ก่อนแล้วถึงจะสามารถใช้ JDBC/ODBC Bridge ในการเรียกใช้ API ได้ในการที่จะต้องทำการติดตั้งโปรแกรมที่ส่วน Client ก่อนนี้ ก็สามารถเปรียบเทียบได้กับระบบ Client-Server แบบเก่า ๆ ที่ไม่สามารถแก้ไขปัญหา Program Deployment ในส่วนของ Client ได้



บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

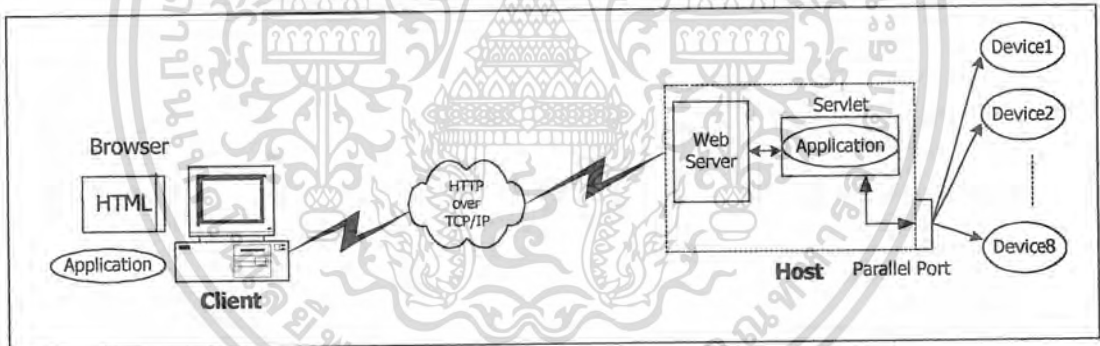
ในการทำปัญหาพิเศษเรื่อง “การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยระบบเครือข่าย” (Home Networking) มีกระบวนการกิจกรรมตามขั้นตอนและวิธีการดำเนินงาน ดังต่อไปนี้

1. ขั้นตอนการวิเคราะห์ระบบการทำงาน (Specification Analysis)
2. ขั้นตอนการดำเนินงาน (Implementing)
3. ขั้นตอนการทดสอบการทำงาน (Testing)

ในแต่ละขั้นตอนและวิธีการดำเนินงานดังกล่าว มีรายละเอียดการทำงานดังนี้

3.1 ขั้นตอนการวิเคราะห์ระบบการทำงาน (Specification Analysis)

ปัญหาพิเศษเรื่องนี้ มีความต้องการให้สามารถใช้คอมพิวเตอร์ ในการเปิด-ปิด เครื่องใช้ไฟฟ้าบางประเภทภายในบ้าน ผ่านระบบเครือข่ายได้ โดยโครงสร้างของระบบจะเป็นการทำงานแบบ Client / Server ดังรูป

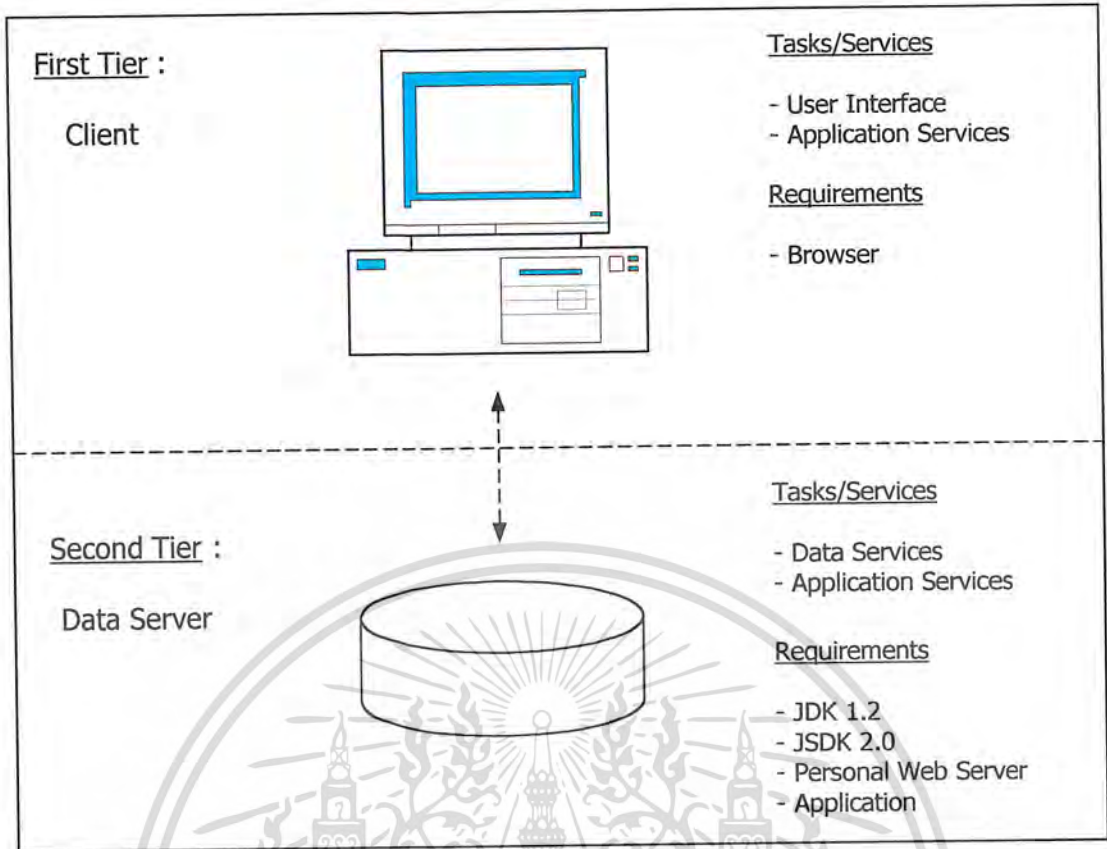


รูปที่ 3.1 โครงสร้างการติดต่อกับพอร์ตนานผ่านระบบเครือข่าย

3.1.1 การทำงาน Client/Server แบบ ทูเทียร์ (Two-Tier Application)

สถาปัตยกรรมของแอปพลิเคชันแบบ Client / Server จะแบ่งการประมวลผลออกเป็น สองโปรแกรม โดยทั่วไปจะทำงานบน เครื่องสองเครื่องขึ้นไป แอปพลิเคชันที่ทำงานกับฐาน ข้อมูลแบบ Client / Server จะรับผิดชอบการเก็บข้อมูล , การประมวลผลข้อมูล , การโอนย้าย ข้อมูล และ ไปแสดงผลที่อื่น เครื่อง Server จะเก็บรวบรวมข้อมูลไว้ ส่วนเครื่องไคลเอนต์จะ ประมวลผลข้อมูลที่ ได้มา หรือ สร้างเป็นข้อมูลใหม่ วิธีการทำงานโดยใช้ สถาปัตยกรรม แบบ Client / Server นี้ทำให้ สามารถติดต่อใช้งานข้อมูลได้ จากผู้ใช้หลายแห่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างแสดงองค์ประกอบของระบบแบบทูเทียร์ (Two - Tier)

จากรูปที่ 3.2 แสดงโครงสร้างต่าง ๆ ภายในระบบ

Tier 1 : เป็นเครื่องไคลเอนต์ทำหน้าที่ส่งไปยังเครื่องเซิร์ฟเวอร์ที่บ้านให้เครื่องเซิร์ฟเวอร์ไปทำการเปิด/เปิดอุปกรณ์เครื่องใช้ไฟฟ้า

สิ่งที่ต้องการ : บราวเซอร์ เช่น Netscape, Internet Explorer

Tier 2 : เครื่องเซิร์ฟเวอร์ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์โดยรับการร้องขอจากเครื่องไคลเอนต์จากนั้นเครื่องเซิร์ฟเวอร์เรียกใช้ Application ซึ่งอยู่ในรูปของ Servlet (ไฟล์.class) โดย Application นี้จะทำการเรียกใช้ Parallel Port ผ่าน file.dll ชื่อ Parport.dll เพื่อควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจำนวน 8 เครื่องที่ต่ออยู่กับ พอร์ตขนานของเครื่องนั้น

สิ่งที่ต้องการ : 1. JDK 1.2 หรือมากกว่าทำหน้าที่คอมไพล์ไฟล์จาวาให้เป็นไฟล์นามสกุล .class

2. JSDK 2.0 หรือมากกว่า ทำหน้าที่เสมือน CGI Server

3. Personal Web Server ทำหน้าที่เสมือนเว็บเซิร์ฟเวอร์ รับการร้องขอจาก

เครื่องไคลเอนต์

4. Application ซึ่งเป็นไฟล์ในการควบคุมอุปกรณ์ไฟฟ้าผ่านพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างระบบมีการทำงานผ่านระบบเครือข่าย ดังนั้นในการทำงานจึงต้องเกี่ยวข้องกับระบบรักษาความปลอดภัย เพื่อไม่ให้ผู้อื่นสามารถเข้ามาควบคุม อุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านได้และต้องมีการป้องกันในส่วนการรักษาความปลอดภัยของสิทธิความเป็นเจ้าของ การควบคุมอุปกรณ์ชุดนั้นๆ ดังนั้นก่อนเข้าไปใช้งาน ต้องเพิ่มในส่วนของการเข้ารหัสในการใช้งาน โดยจะทำการเก็บรายชื่อของผู้ใช้แต่ละคนในฐานข้อมูล เพื่อง่ายต่อการตรวจสอบสิทธิของผู้ใช้งาน

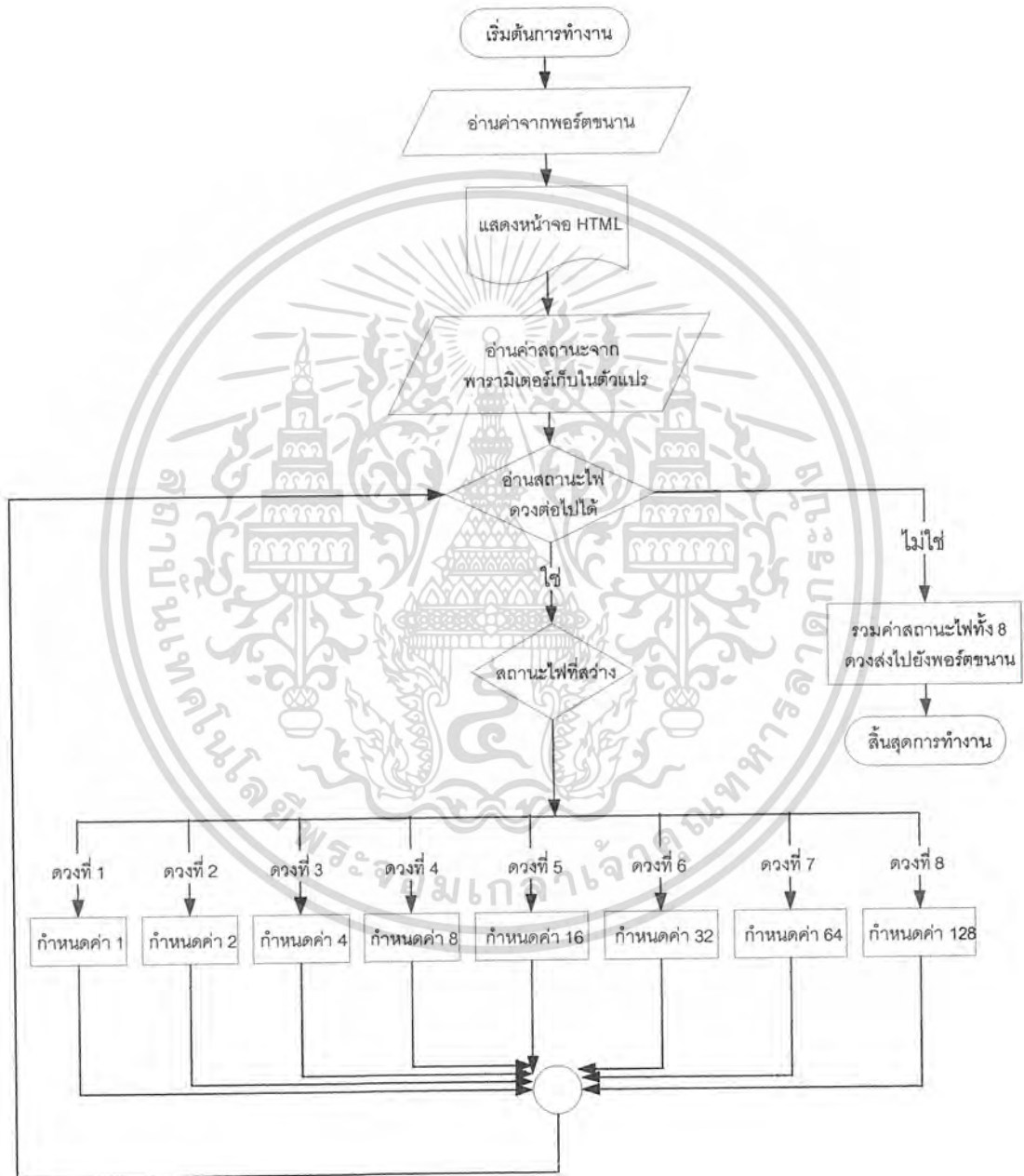


รูปที่ 3.3 แผนภาพ Flow Chart แสดงขั้นตอนการตรวจสอบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านจะต้องทำการเปิด-ปิด เครื่องใช้ไฟฟ้าภายในบ้าน ผ่านพอร์ตขนาน ซึ่งสามารถทำการควบคุมอุปกรณ์ภายในบ้านได้ 8 อย่าง ด้วยวิธีการส่งบิตข้อมูลออกทางขา 2 ถึง 9 (D0-D7) ของพอร์ตขนาน ซึ่งเชื่อมต่ออยู่กับอุปกรณ์ภายในบ้าน

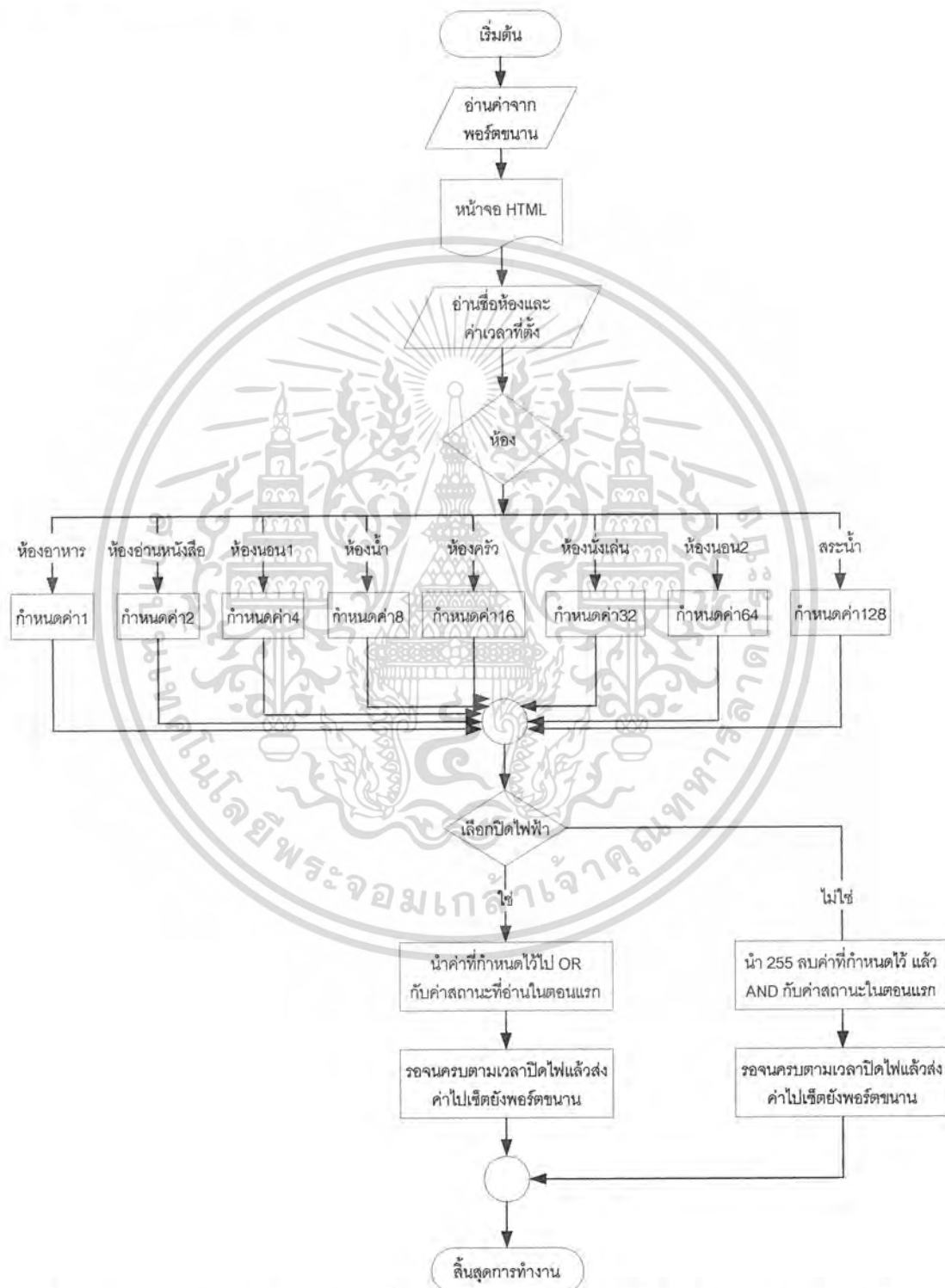
จากรูปที่ 3.3 เมื่อผู้ใช้งานเข้าสู่หน้าจอการทำงานจะต้องมีการตรวจสอบค่า Username และ Password ก่อนเข้าสู่หน้าจอการทำงาน เพื่อความปลอดภัยในการทำงาน



รูปที่ 3.4 แผนภาพ Flow Chart แสดงขั้นตอนการควบคุมอุปกรณ์ไฟฟ้าแบบทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 เป็นรูปแบบการควบคุมอุปกรณ์ไฟฟ้าแบบทั่วไปสามารถทำการเลือกเปิดหรือปิดเครื่องใช้ไฟฟ้าได้พร้อมกันทั้ง 8 อย่าง โดยจะต้องทำการอ่านค่าสถานะของอุปกรณ์ไฟฟ้า ขึ้นมาเพื่อเพื่อแสดงให้ผู้ใช้ ทราบก่อนทำการเลือกใช้งานเพื่อจะได้ทราบว่า ณ ขณะนั้นอุปกรณ์ไฟฟ้าชนิดมีสถานะเปิดหรือปิดอยู่



รูปที่ 3.5 แผนภาพ Flow Chart แสดงขั้นตอนการควบคุมอุปกรณ์ไฟฟ้าแบบเฉพาะแต่ละบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.5 เป็นการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เฉพาะแต่ละบ้านโดยจะมี Plan บ้านบอกส่วนต่าง ๆ ภายในบ้านว่าประกอบด้วยห้องใดบ้าง แต่ละห้องอุปกรณ์ไฟฟ้ามีสถานะเปิดหรือปิดอยู่ และสามารถทำการตั้งเวลาเปิด-ปิด อุปกรณ์ไฟฟ้าได้ตามต้องการ

3.2 ขั้นตอนการดำเนินงาน (Implementing)

3.2.1 การพัฒนาโปรแกรมด้วยภาษาจาวา

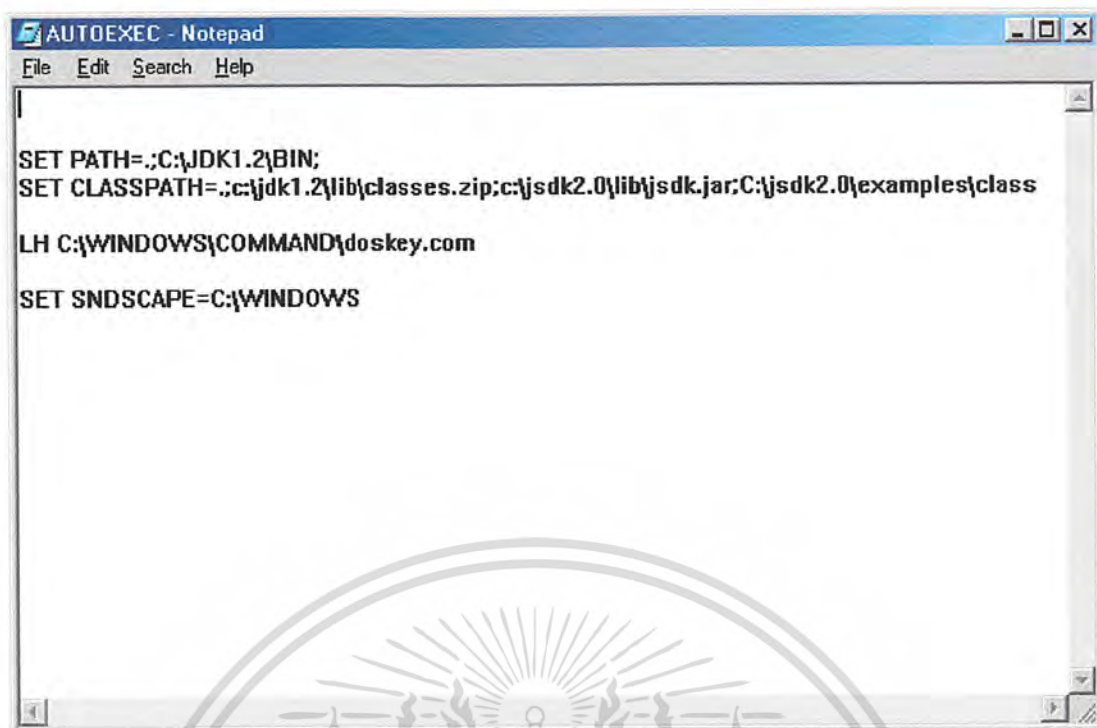
โปรแกรมที่สร้างขึ้นจากเซิร์ฟเล็ตสามารถคอมไพล์ได้โดยอาศัย jdk (Java Developer Kit) และจาวาคอมไพเลอร์ตัวอื่น ๆ แต่ถ้าหากคอมไพเลอร์ที่มีอยู่ไม่ได้สนับสนุนคุณสมบัติสำหรับแพ็คเกจ javax.servlet และ javax.servlet.http ผู้ใช้จำเป็นต้องติดตั้งแพ็คเกจทั้งสองนี้ลงไปในระบบด้วยตนเองโดยสามารถดาวน์โหลดได้จาก The JSDK download page

<http://java.sun.com/products/java-server/servlets/index.html#jdk>

ขั้นตอนในการติดตั้งโปรแกรมที่ใช้งานเซิร์ฟเล็ต คือ

1. ทำการลงโปรแกรม jdk 1.2
2. ทำการลงโปรแกรม jsdk 2.0
3. พิมพ์ ;C:\JDK1.2\BIN; เพิ่มที่ SET PATH
4. พิมพ์ .;c:\jdk1.2\lib\classes.zip;c:\jdk2.0\lib\jsdk.jar เพิ่มที่ SET CLASSPATH
5. พิมพ์ c:\jdk2.0\examples\class เพื่อใช้ในการอ้างอิงในการอ้างอิง File ParallelPort.class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

AUTOEXEC - Notepad
File Edit Search Help

SET PATH=.;C:\JDK1.2\BIN;
SET CLASSPATH=.;c:\jdk1.2\lib\classes.zip;c:\jdk2.0\lib\jdk.jar;C:\jdk2.0\examples\class
LH C:\WINDOWS\COMMAND\doskey.com

SET SNDSCAPE=C:\WINDOWS

```

รูปที่ 3.6 วิธีการเซต PATH และ CLASSPATH ใน Autoexec.bat

3.2.2 การเขียนโปรแกรมติดต่อพอร์ตขนานด้วย JNI (Java Native Interface)

JNI เป็นวิธีการเขียนโปรแกรมที่ทำให้จาวาสามารถเรียกใช้โมดูลต่างๆ ที่มีอยู่แล้วซึ่งเขียนด้วยภาษาอื่นๆ เช่น C, C++ หรือ Assembly

ประโยชน์ของ JNI คือผู้พัฒนาโปรแกรมสามารถใช้ library ที่มีอยู่แล้วในภาษาอื่นแล้วนำมาใช้กับ Java Application ได้ หรือ Application บางอย่างที่ใช้ Java Class Library ไม่มี ดังนั้นจึงต้องเขียน code ภาษาอื่นแทนแล้วจึง load library ที่ได้จากภาษาอื่นมาใช้ในภายหลัง

1) วิธีการเขียนโปรแกรมโดยใช้ JNI (ในที่นี้เป็นการใช้ library ของภาษา C) มีดังต่อไปนี้

1. สร้าง header file ซึ่งเป็นไฟล์นามสกุล .h โดยใช้คำสั่ง
javah -jni ชื่อไฟล์
2. เขียน code ภาษา C โดยมีการ include ไฟล์ที่ได้จากข้อ 1
3. ทำการ compile โปรแกรมที่ได้ในข้อ 2
4. เขียนโปรแกรมภาษา java โดย implement code ที่ได้จากข้อ 2 ด้วยคำสั่ง
System.loadLibrary("ชื่อไฟล์ในข้อที่ 2"); และ method ที่มีการประกาศให้เป็น native method
จะต้องมีคำว่า native ประกาศด้วย
5. ทำการ compile โปรแกรมที่ได้จากข้อ 4 ด้วยคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

javac ชื่อไฟล์.java

6. หลังจากนั้นเราจะได้ไฟล์ class ซึ่งสามารถนำไปเรียกใช้ใน class อื่นได้ภาย

หลัง

2) การเขียนโปรแกรมติดต่อพอร์ตขนานด้วย JNI

1. สร้าง header file ชื่อ ParallelPort.h ด้วยคำสั่ง javah -jni ParallelPort.h

ParallelPort.h เป็นการบอกว่าเราสร้าง header file ซึ่งจะมีการเรียกใช้ method ชื่อ

Java_ParallelPort_readOneByte และ Java_ParallelPort_writeOneByte

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class ParallelPort */
#ifndef _Included_ParallelPort
#define _Included_ParallelPort
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class: ParallelPort
 * Method: readOneByte
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_ParallelPort_readOneByte (JNIEnv *, jclass, jint);
/*
 * Class: ParallelPort
 * Method: writeOneByte
 * Signature: (II)V
 */
JNIEXPORT void JNICALL Java_ParallelPort_writeOneByte (JNIEnv *, jclass, jint, jint);
#ifdef __cplusplus
}
#endif
#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. include ไฟล์ชื่อ ParallelPort.h(parall~1.h) ซึ่งได้จากข้อ 1 โดยจะมีการ
ระบุการทำงานลงใน method ที่เราได้ประกาศไว้ใน header file แล้ว

parport.c

```
#include <conio.h>
```

```
#include "parall~1.h"
```

```
JNIEXPORT jint JNICALL Java_ParallelPort_readOneByte
```

```
(JNIEnv * algo, jclass otro, jint portStatus)
```

```
{
```

```
    unsigned short status;
```

```
    status = (unsigned short)portStatus;
```

```
    return _inp(status);
```

```
}
```

```
JNIEXPORT void JNICALL Java_ParallelPort_writeOneByte
```

```
(JNIEnv * algo, jclass otro, jint portData, jint oneByte)
```

```
{
```

```
    unsigned short data;
```

```
    int aByte;
```

```
    data = (unsigned short)portData;
```

```
    aByte = (int)oneByte;
```

```
    _outp(data,aByte);
```

```
}
```

3. ทำการ compile โปรแกรม parport.c ซึ่งได้จากข้อ 2

4. เขียนโปรแกรมภาษา java โดย implement code ที่ได้จากข้อ 2

(parport.c) ด้วยคำสั่ง

```
System.loadLibrary("parport");
```

และ method ที่มีการประกาศให้เป็น native method จะต้องมีคำว่า native ประกาศด้วย

ในที่นี้ก็จะจะมี 2 native method คือ readOneByte และ writeOneByte

- readOneByte จะอ่านค่าจาก parallel port
- writeOneByte จะส่งค่าออกไปยัง parallel port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ParallelPort.java

```
public class ParallelPort {
    /** The port base address (e.g. 0x378 is base address for LPT1) */
    private int portBase;
    /** To construct a ParallelPort object,
     * you need the port base address
     */
    public ParallelPort ( int portBase )
    {
        this.portBase = portBase;
    }
    /** Reads one byte from the parallel port */
    public int read ()
    {
        return ParallelPort.readOneByte (this.portBase);
    }
    /** Writes one byte to the parallel port */
    public void write (int oneByte)
    {
        ParallelPort.writeOneByte (this.portBase,oneByte);
    }
    public static native int readOneByte ( int portData );
    /** Writes one byte to the parallel port.
     * The byte is written to the DATA pins of the port. The DATA pins are
     * located at the base address of the port (e.g. DATA address for LPT1
     * is 0x378).
     * This diagram shows how the byte is written and read:
     * Bit | Pin # | Printer DATA
     * ----+-----+-----
     * 7 | 9 | DATA 7
     * 6 | 8 | DATA 6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* 5 | 7 | DATA 5
* 4 | 6 | DATA 4
* 3 | 5 | DATA 3
* 2 | 4 | DATA 2
* 1 | 3 | DATA 1
* 0 | 2 | DATA 0
*/

public static native void writeOneByte ( int portData, int oneByte );

static
{
    System.loadLibrary("parport");
}
}

```

5. ทำการ compile โปรแกรมที่ได้จากข้อ 4 ด้วยคำสั่ง

```
javac ParallelPort.java
```

6. หลังจากนั้นทำการ set classpath เพื่อนำ class ParallelPort ที่ได้จาก

ข้อ 5 ไปประยุกต์ใช้งาน

ดังข้างล่างเป็นตัวอย่างของการอ่านค่าจาก parallel port เก็บลงตัวแปรชื่อ dataread

```
ParallelPort = new ParallelPort(0x378);
```

```
dataread = LTP1.read();
```

3.2.3 การเขียนโปรแกรมด้วย Servlet โดยเรียกใช้งาน file ParallelPort.class

การเขียนโปรแกรมด้วย Servlet สามารถแสดงหน้าจอ Html ได้ ดังนั้นจึงเป็นข้อดีในการนำมาใช้งานในส่วนของ Web Application จะใช้โปรแกรม Servlet ในการเช็คสถานะของอุปกรณ์ไฟฟ้าว่าอุปกรณ์แต่ละชนิดภายในบ้านว่ามีสถานะเปิดหรือปิดอยู่ โดยใช้คำสั่ง

```
ParallelPort LPT1 = new ParallelPort(0x378); // 0x378 is the base address for
```

```
Parallel printer port LPT1 dataread=LPT1.read();
```

ซึ่ง dataread จะเก็บข้อมูลตัวเลข Integer ฐาน 10 ดังนั้นต้องทำการเปลี่ยนเป็นเลขฐาน 2 เพื่อจะได้ทราบสถานะของอุปกรณ์ไฟฟ้าแต่ละชนิด

```
int status[];
```

```
status = new int[8];
```

```
for(int i=0;i<=7;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    status[i] = dataread%2;
    dataread = dataread/2;
}

```

จากนั้นแสดงสถานะให้ผู้ใช้งานเห็นโดยผ่านหน้าจอ Browser ด้วย Code ภาษา Html ภายในตัวโปรแกรม Servlet เมื่อผู้ใช้เลือกเปิด-ปิดอุปกรณ์ไฟฟ้า กดปุ่ม "ตกลง" ตัวโปรแกรม Servlet จะรับค่าจาก Form มาเก็บไว้ในตัวแปรเพื่อทำการเซตสถานะอุปกรณ์ไฟฟ้าตามที่ใช้เลือกเอาไว้ โดยจะนำตัวแปรไปทำการตรวจสอบว่าเป็นของอุปกรณ์ชนิดไหน แล้วตั้งเวลาเปิด-ปิด เท่าไหร่

```

String statusget = request.getParameterValues("open")[0];
int status = Integer.parseInt(statusget);

String roomget = request.getParameterValues("room")[0];
int room = Integer.parseInt(roomget);

String CloseTimeGet = request.getParameterValues("closetime")[0];
int CloseTime = Integer.parseInt(CloseTimeGet);

String OpenTimeGet = request.getParameterValues("opentime")[0];
int OpenTime = Integer.parseInt(OpenTimeGet);

```

จากนั้นจะทำการเรียก class PrintThread เพื่อใช้ในการตั้งเวลา ซึ่งจะให้ Thread sleep ด้วยเวลาที่ผู้ใช้ต้องการโดยมีหน่วยเป็น มิลลิวินาที ดังนั้นจึงต้องทำการแปลงเวลาจากวินาทีให้เป็นมิลลิวินาทีก่อน

```
sleepTime = TimeGet*60*1000;
```

โดยทาง Servletrunner สามารถทราบว่า ณ ขณะนี้เครื่อง Client กำลังสั่งงานอุปกรณ์ไฟฟ้าในห้องใดอยู่ ด้วยคำสั่ง

```
System.out.println("Name : "+getName()+"; sleep : "+sleepTime);
```

เมื่อในส่วนของ Thread ทำงานเสร็จสิ้นแล้วจะทำการเซตสถานะไฟ โดยส่งข้อมูลออกไปยังพอร์ตนานด้วยคำสั่ง

```
LPT1.write(dataout);
```

หลังจากนั้น Application จะทำการเปิด-ปิดอุปกรณ์ไฟฟ้า เมื่อเรียบร้อยแล้วจะแสดงหน้าจอเพื่อแจ้งให้ผู้ใช้งานทราบและสามารถทำการเปิด-ปิด อุปกรณ์ไฟฟ้าชนิดอื่นต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การเขียนโปรแกรมด้วยภาษา Visual Basic ติดต่อฐานข้อมูล

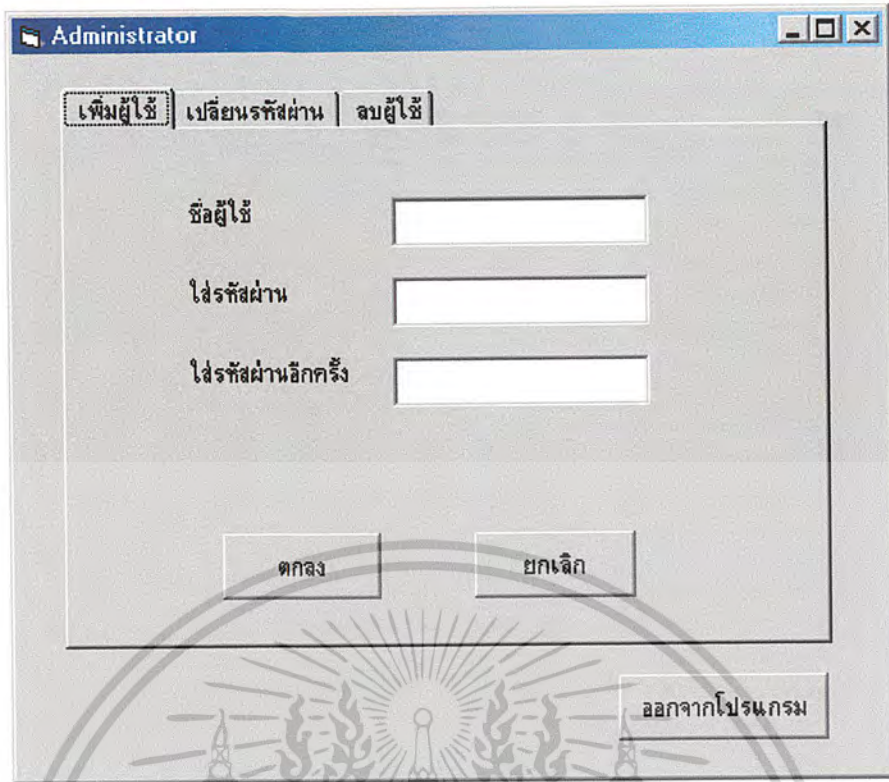
ในการใช้งานสำหรับควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจำเป็นต้องมีการ Login ทางฝั่ง Client เข้ามายังฝั่ง Server สำหรับตรวจสอบผู้ที่ จะเข้ามาใช้งาน และป้องกันไม่ให้ผู้อื่นสามารถเข้ามาควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้ โดยในส่วนของโปรแกรมที่ใช้สำหรับเพิ่ม, ลบ และแก้ไข ผู้ใช้งานในฐานะข้อมูลนั้น จะใช้งานทางฝั่ง Server และใช้ภาษา Visual Basic ในการติดต่อกับฐานข้อมูล Microsoft Access

3.2.4.1 การทำการ Set ODBC เพื่อติดต่อกับฐานข้อมูล Microsoft Access ขั้นตอนในการเซต ODBC

1. ทำการเลือก Start → Settings → Control Panel
 2. จากนั้นเลือก ODBC Data Source (32 Bit)
 3. เลือกที่ Tab System DSN แล้วเลือกที่ปุ่ม Add จากนั้น เลือก Microsoft Access Driver (*.mdb) แล้วกด Finish จะปรากฏหน้าจอ ODBC Microsoft Access Setup
 4. ให้ใส่ชื่อ Data Source Name เป็น homenet จากนั้นกดปุ่ม Select
 5. ให้เลือก Database Name ชื่อ Database.mdb จาก Directory ที่เก็บไว้ แล้วกดปุ่ม OK
 6. จากนั้นเมื่อกลับเข้ามาสู่หน้าจอ ODBC Microsoft Access Setup ให้กดเลือก Advanced
- เข้าสู่หน้าจอ Set Advanced Options พิมพ์ admin ลงไปที่ Login name: และ Password: แล้วกดปุ่ม OK

3.2.4.2 การโปรแกรมเพื่อเพิ่ม, ลบ และแก้ไขข้อมูลผู้ใช้งาน

หลังจากที่ได้ทำการ Set ODBC ในการติดต่อกับฐานข้อมูลของ Microsoft Access เรียบร้อยแล้ว จึงทำการสร้าง Interface เพื่อใช้ในการติดต่อกับฐานข้อมูลในการเพิ่ม, ลบ และแก้ไขข้อมูลของผู้ที่สามารถเข้ามาใช้งานโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยภาษา Visual Basic



รูปที่ 3.7 หน้าจอสำหรับเพิ่ม, ลบ และแก้ไข ผู้ใช้งานในฐานข้อมูล

จากรูปที่ 3.7 เป็นรูปแบบหน้าจอที่สร้างด้วยโปรแกรมภาษา Visual Basic ซึ่งจะประกอบด้วย Tab 3 Tab คือ

1. Tab เพิ่มผู้ใช้งาน จะทำการเพิ่มผู้ใช้งานที่จะสามารถเข้ามาใช้งานโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้ ในส่วนของการทำงานของการเพิ่มผู้ใช้งาน มีการทำงานดังนี้ เริ่มต้นด้วยการสร้าง Connection ที่ชื่อว่า conn

Dim conn As Connection

Set conn = New Connection

หลังจากนั้นทำการรับค่าข้อมูลที่ป้อนเข้ามาเก็บในตัวแปร

getUser = Text1.Text

getPwd1 = Text2.Text

getPwd2 = Text3.Text

จากนั้นทำการเทียบรหัสผ่านที่ป้อนเข้ามาว่าตรงกันหรือไม่

ในกรณีที่ตรงกันจะทำการสร้าง Connection ติดต่อไปยังฐานข้อมูลที่เราได้เซตไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If getPwd1 = getPwd2 Then
```

```
With conn
```

```
.Provider = "Microsoft.Jet.OLEDB.3.51"
```

```
.ConnectionString = "Data Source=database.mdb"
```

```
.Open
```

```
End With
```

หลังจากนั้นจะทำการหาข้อมูลว่า username ที่ได้รับมานั้นมีอยู่ในฐานข้อมูลแล้วหรือยัง

```
sqlSearch = "select username from Table1 where username = " + getUser + ""
```

```
Debug.Print sqlSearch
```

```
rs.Open sqlSearch, conn, adOpenForwardOnly
```

กรณีที่ไม่มี username อยู่ใน DB ทำการเพิ่มข้อมูลลงใน DB

```
If (rs.EOF) Then
```

```
sqlcmd = "INSERT INTO Table1 (username,password,RdmNo) VALUES
```

```
(" + getUser + "," + getPwd1 + ",'" + "
```

```
Debug.Print sqlcmd
```

```
conn.Execute sqlcmd
```

```
conn.Close
```

```
Data1.Refresh
```

```
MsgBox "ได้ทำการเพิ่ม " + getUser + " เรียบร้อยแล้ว ", vbOKOnly +
```

```
vbInformation, "Complete"
```

```
Call Command2_Click
```

```
Else
```

กรณีที่มี username อยู่ใน DB อยู่แล้ว ให้ clear textbox

```
MsgBox "ชื่อผู้ใช้ที่ท่านป้อนมีอยู่แล้ว กรุณาป้อนชื่อผู้ใช้ใหม่ ", vbOKOnly +
```

```
vbInformation, "Warning"
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
End If
```

```
Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใส่รหัส Password ไม่เหมือนกันทั้ง 2 ช่อง ทำการใส่ password ใหม่

```
MsgBox "กรุณาป้อนรหัสผ่านอีกครั้ง", vbOKOnly + vbInformation, "Error"
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
End If
```

```
End Sub
```

2. Tab ลบผู้ใช้งาน จะทำการลบผู้ใช้งานออกจากฐานข้อมูล เพื่อไม่ให้ผู้นั้นเข้ามาใช้งาน เริ่มต้นด้วยการสร้าง Connection ที่ชื่อว่า conn และสร้าง Connection ติดต่อไปยังฐานข้อมูลที่เราได้เซตไว้แล้ว เหมือนกับใน Tab เพิ่มผู้ใช้งาน

ทำการหาข้อมูลว่า username ที่ได้รับมานั้นมีอยู่ในฐานข้อมูลแล้วหรือยัง

```
sqlSearch = "select password from Table1 where username = " + getUser + ""
```

```
Debug.Print sqlSearch
```

```
rs.Open sqlSearch, conn, adOpenForwardOnly
```

กรณีที่ password เหมือนกัน

```
If getPwd1 = getPwd2 Then
```

กรณีที่ password ถูกต้อง

```
If rs.Fields("password").Value = getPwd1 Then
```

```
sSQL = "DELETE FROM Table1 WHERE username=" + getUser + ""
```

```
conn.Execute sSQL
```

```
conn.Close
```

```
Data1.Refresh
```

```
DBCombo2.Refresh
```

```
MsgBox "ได้ทำการลบรหัสผ่านของ " + getUser + " เรียบร้อยแล้ว", vbOKOnly +
```

```
vbInformation, "DELETE Complete"
```

```
DBCombo2.Text = ""
```

```
Text8.Text = ""
```

```
Text9.Text = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ password ผิด

Else

MsgBox "ท่านใส่ค่ารหัสผ่านไม่ถูกต้อง", vbOKOnly + vbInformation, "Error"

Text8.Text = ""

Text9.Text = ""

End If

กรณีที่ password ใสไม่เหมือนกัน

Else

MsgBox "กรุณาใส่ค่ารหัสผ่านใหม่", vbOKOnly + vbInformation, "Error"

Text8.Text = ""

Text9.Text = ""

End If

DBCombo2.Refresh

End Sub

3. Tab แก้ไขผู้ใช้งาน จะให้ผู้ใช้งานสามารถเปลี่ยนแปลง Password ของผู้นั้นได้ เริ่มต้นด้วยการสร้าง Connection ที่ชื่อว่า conn และสร้าง Connection ต่อกับฐานข้อมูลที่ เราได้เซตไว้แล้ว เหมือนกับใน Tab เพิ่มผู้ใช้งาน ดูว่าป้อน username กับ password เก่าถูกหรือเปล่า

```
sqlSearch = "select * from Table1 where username=" + getUser + "and password=" + getOldPwd + ""
```

```
rs.Open sqlSearch, conn, adOpenForwardOnly
```

กรณีที่ป้อน Username และ Password ถูก

If Not (rs.EOF) Then

เช็คว่ามี password ใหม่ถูกหรือไม่

If (getNewPwd1 = getNewPwd2) Then

ถ้าเหมือนกันก็ update

```
sqlcmd = "UPDATE Table1 SET password = " + getNewPwd1 + " where username = " + getUser + ""
```

```
conn.Execute sqlcmd
```

```
conn.Close
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Data1.Refresh
DBCombo1.Refresh
MsgBox "ได้เปลี่ยนแปลงรหัสผ่านของ " + getUser + " เรียบร้อยแล้ว", vbOKOnly +
vbInformation, "UPDATE Complete"
Call Command5_Click
Else
MsgBox "รหัสผ่านใหม่ป้อนไม่ตรงกัน กรุณาป้อนใหม่อีกครั้ง", vbOKOnly +
vbInformation, "Error"
Text6.Text = ""
Text7.Text = ""
End If
Else
กรณีที่ไม่เหมือนกันให้ป้อน password ใหม่
MsgBox "ท่านป้อนรหัสผ่านเดิมไม่ถูกต้อง กรุณาป้อนใหม่อีกครั้งหนึ่ง", vbOKOnly +
vbInformation, "Error"
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
End If
End Sub

```

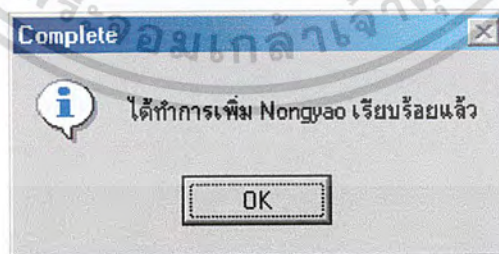
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการทดสอบการทำงาน (Testing)

3.3.1 การทดสอบการใช้งานการเพิ่ม, ลบ และแก้ไขผู้ใช้งานในเครื่อง Server

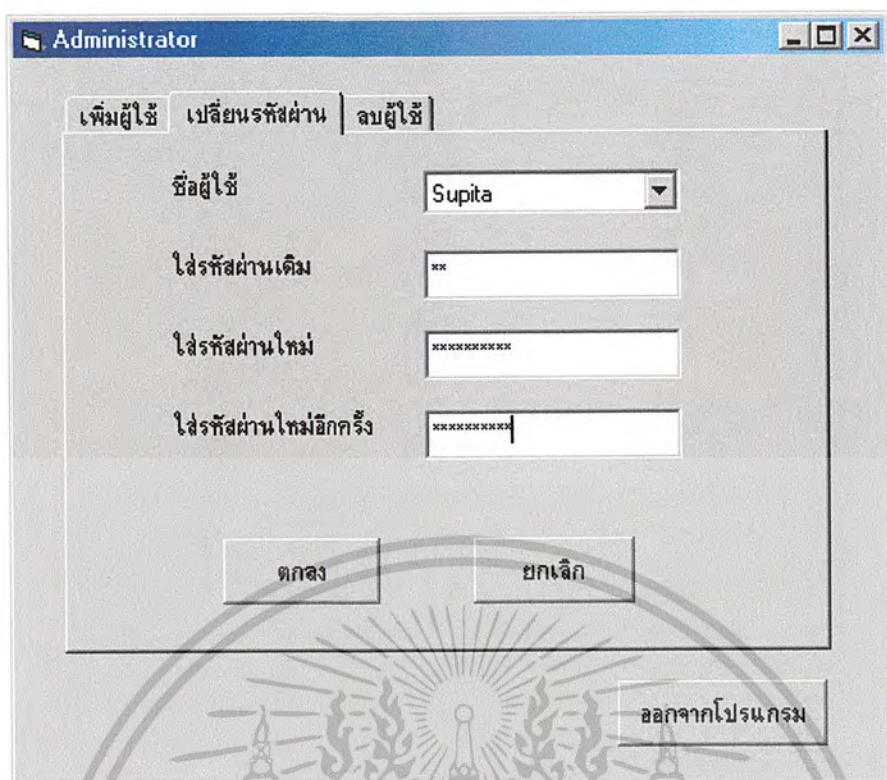
รูปที่ 3.8 แสดงหน้าจอตัวอย่างการเพิ่มผู้ใช้งาน

จากรูปที่ 3.8 เมื่อได้ทำการป้อนชื่อผู้ใช้และใส่รหัสผ่านแล้ว กดปุ่ม “ตกลง” จะขึ้น MessageBox แสดงดังในรูปที่ 3.9 ว่าท่านได้ทำการเพิ่มผู้ใช้เรียบร้อยแล้ว



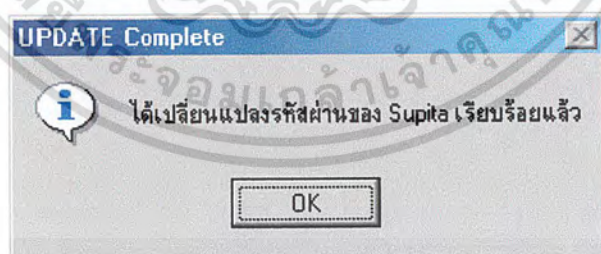
รูปที่ 3.9 แสดงหน้าจอหลังจากเพิ่มผู้ใช้งานเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



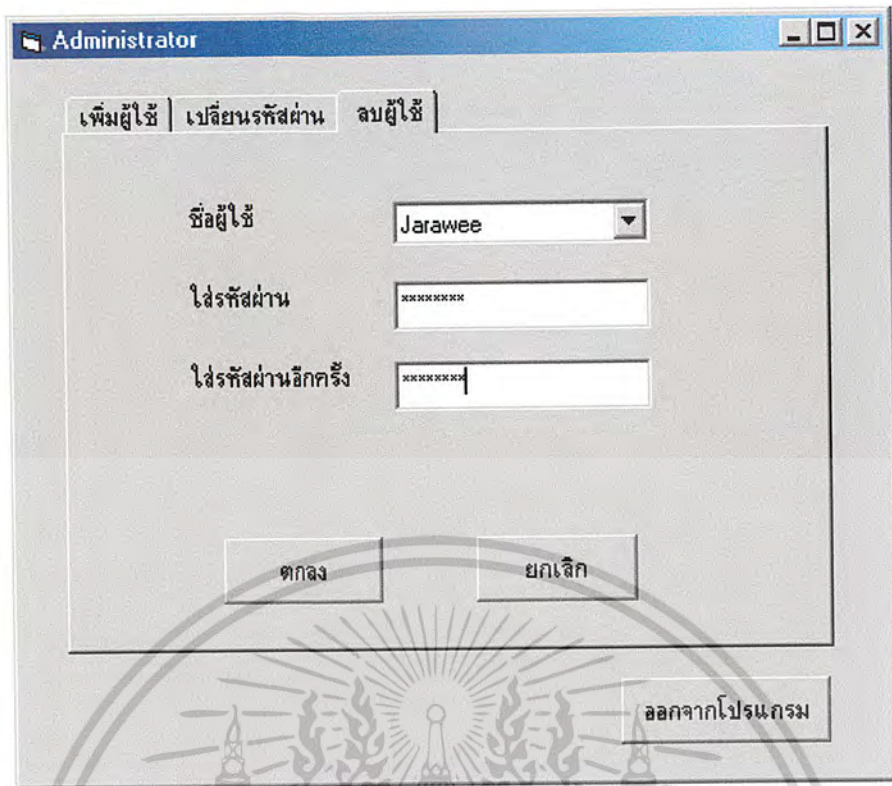
รูปที่ 3.10 แสดงหน้าจอตัวอย่างการเปลี่ยนรหัสผู้ใช้งาน

จากรูปที่ 3.10 เมื่อได้ทำการเลือกชื่อผู้ใช้และใส่รหัสผ่านเดิม และใส่รหัสผ่านใหม่ 2 ครั้งแล้ว กดปุ่ม “ตกลง” จะขึ้น MessageBox แสดงดังในรูปที่ 3.11 ว่าท่านได้ทำการเปลี่ยนแปลงรหัสผ่านผู้ใช้เรียบร้อยแล้ว



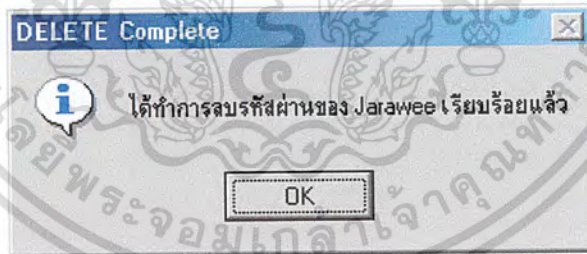
รูปที่ 3.11 แสดงหน้าจอหลังจากเปลี่ยนรหัสผู้ใช้งานเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงหน้าจอตัวอย่างการลบผู้ใช้งาน

จากรูปที่ 3.12 เมื่อได้ทำการเลือกชื่อผู้ใช้และใส่รหัสผ่านเดิม 2 ครั้งแล้ว กดปุ่ม “ตกลง” จะขึ้น MessageBox แสดงดังในรูปที่ 3.13 ว่าท่านได้ทำการลบผู้ใช้เรียบร้อยแล้ว



รูปที่ 3.13 แสดงหน้าจอหลังจากลบผู้ใช้งานเรียบร้อยแล้ว

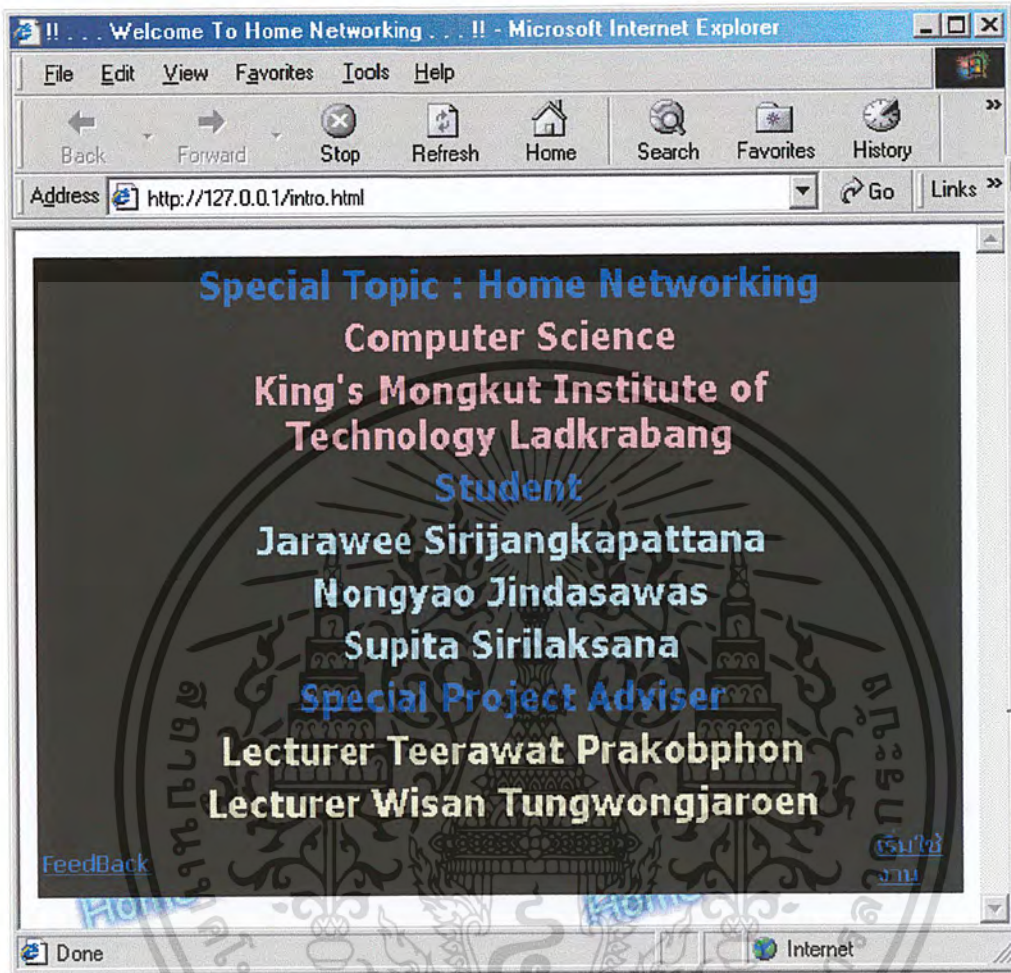
3.3.2 การทดสอบการใช้งาน Application ทางฝั่ง Client

ในการทดสอบการทำงาน จะมีอุปกรณ์ช่วยในการตรวจสอบสถานะทางไฟฟ้าของอุปกรณ์ไฟฟ้า คือ บอร์ด LED มอนิเตอร์ 16 ช่อง ซึ่งจะมีตัวคอนเนกเตอร์ DATA-BUS สำหรับเชื่อมต่อกับ DATA-BUS บน P-Board จากนั้นเราลองทำการทดสอบตัวโปรแกรมโดยเราจะต้องรู้ว่าเครื่อง Server ของเรามีหมายเลข IP เบอร์อะไร จากนั้นเราทำการต่อเครื่อง Client ที่กำลังใช้งานเข้ากับระบบเครือข่าย Internet จากนั้นทำการเปิด Browser แล้วเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

http:// หมายเลข IP ของเครื่อง Host ที่บ้าน /intro.html

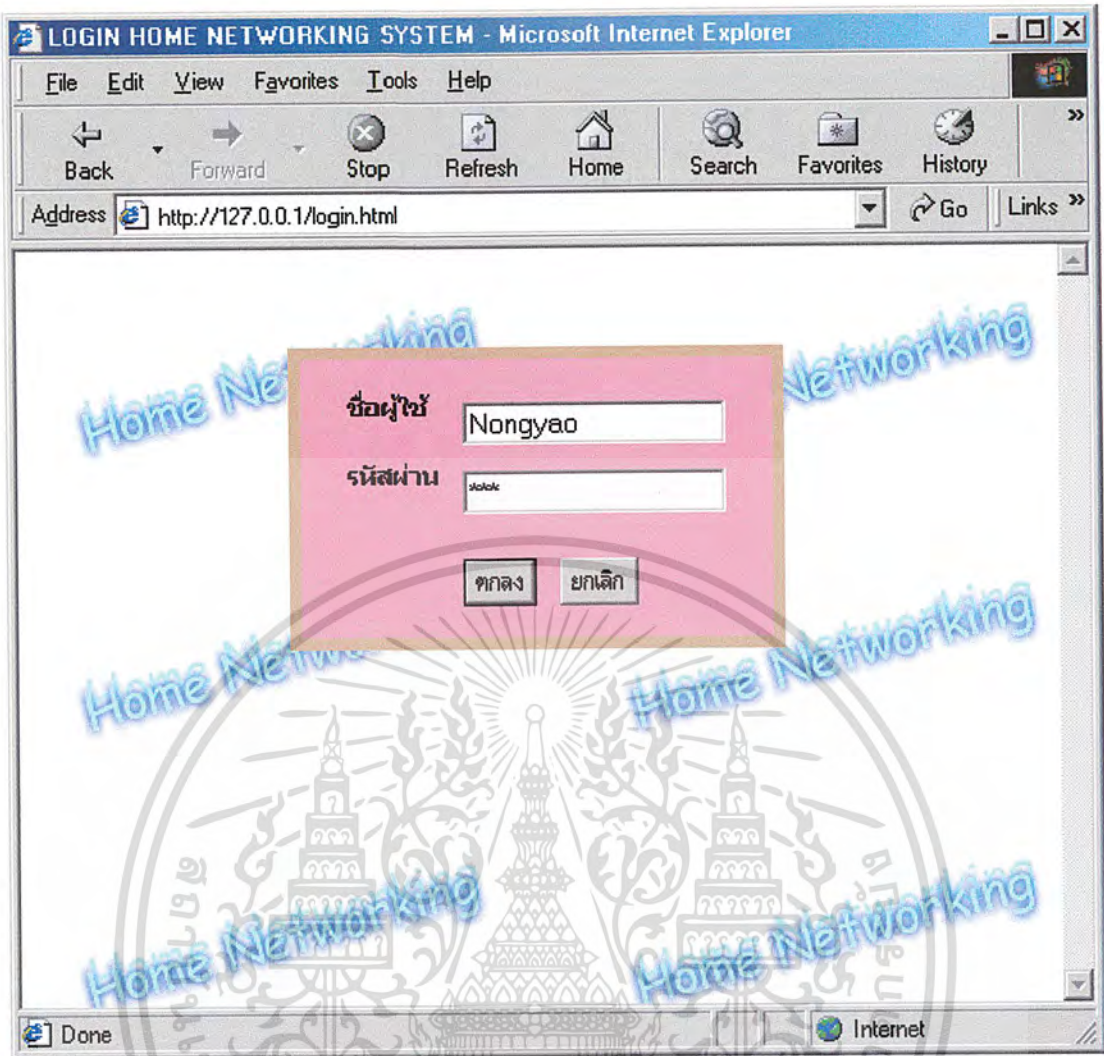
จะขึ้นหน้าจอแรกเพื่อเข้าสู่โปรแกรมการใช้งาน



รูปที่ 3.14 แสดงหน้าจอแรกเพื่อเข้าสู่โปรแกรมการใช้งาน

จากนั้นเราจึงเข้าสู่หน้าเริ่มต้นการใช้งานด้วยการกดไปยัง เริ่มใช้งานเพื่อเข้าสู่หน้าจอ Login เข้าไปใช้งาน

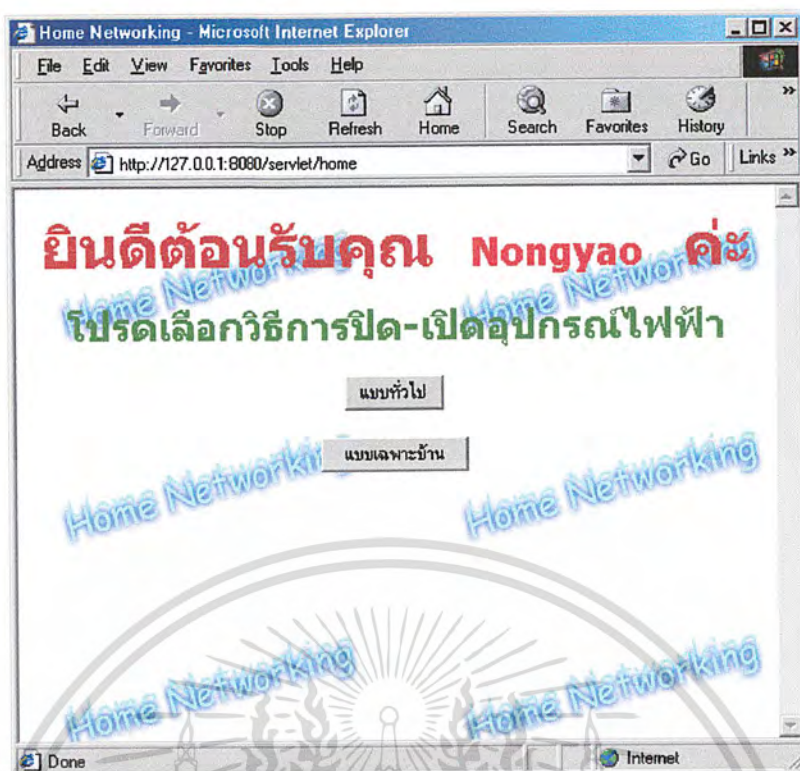
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงหน้าจอต้อนรับการ Login เข้าสู่โปรแกรม

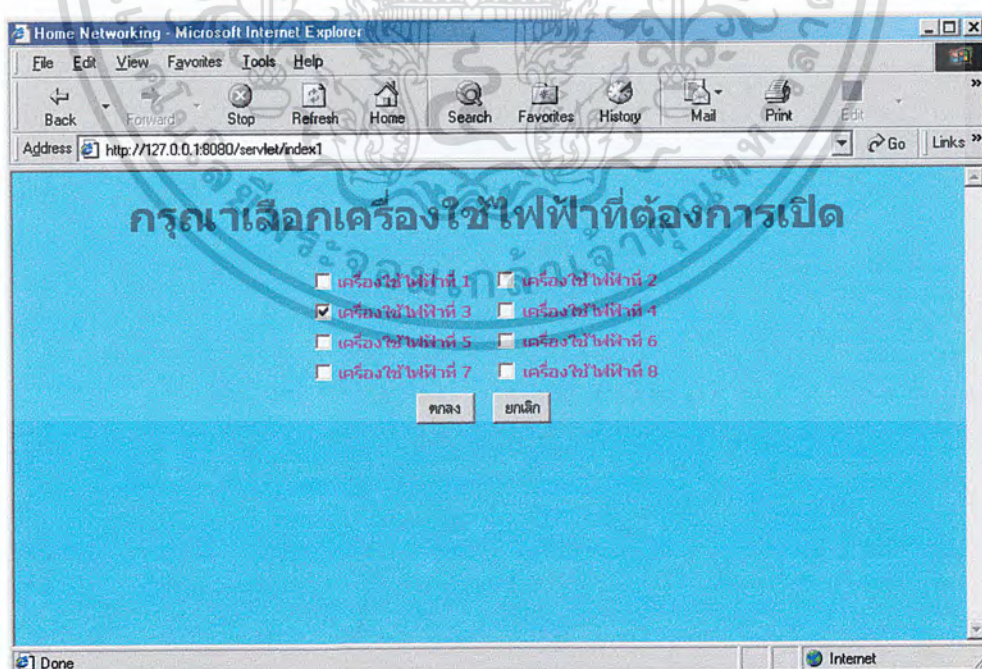
ทำการป้อน username และ password ของผู้ใช้งาน แล้วทำการกดปุ่ม Enter ถ้า username และ password ของท่านมีอยู่ในฐานข้อมูลก็สามารถเข้าไปใช้งานควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้ โดยจะขึ้นหน้าจอให้ผู้ใช้งานทำการเลือกที่จะควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบใด จะมีให้เลือกใช้งานอยู่ 2 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



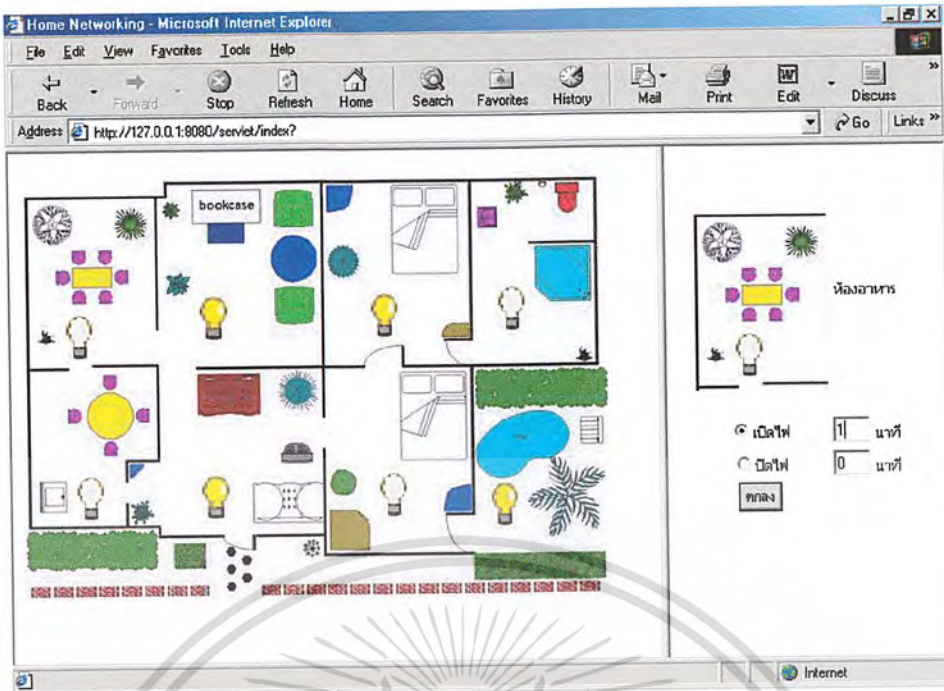
รูปที่ 3.16 แสดงหน้าจอเลือกการใช้งาน 2 แบบ

เมื่อท่านทำการเลือกควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบแรกจะแสดงหน้าจอดังรูปที่ 3.17



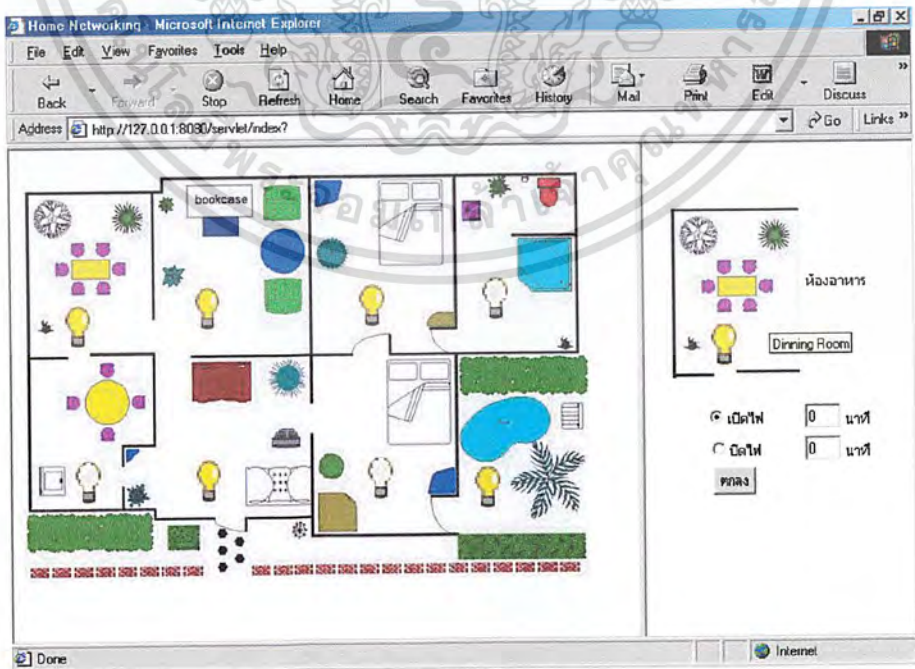
รูปที่ 3.17 แสดงหน้าจอหลังจากเลือกใช้งานในแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงหน้าจอหลังจากเลือกใช้งานในแบบที่ 2

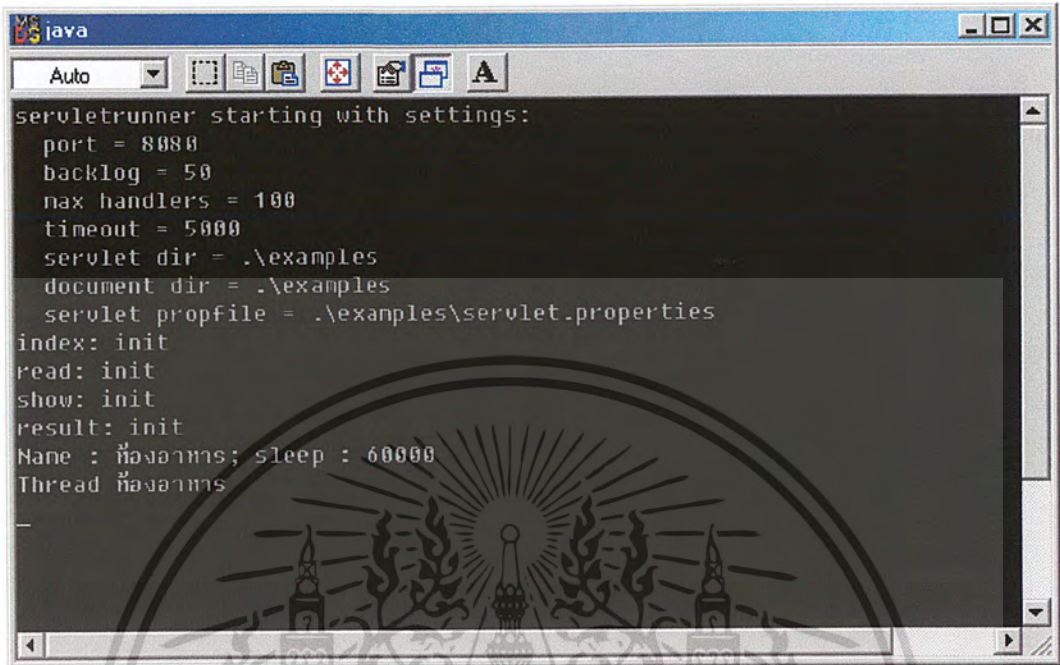
หน้าจอจะแสดงสถานะของไฟแต่ละห้องว่ามีอุปกรณ์ไฟฟ้าใดบ้างที่เปิดอยู่ โดยไฟของ อุปกรณ์ไฟฟ้าจะขึ้นเหมือนกับที่แสดงบนบอร์ด LED เมื่อทำการเลือกเปิด-ปิดอุปกรณ์ไฟฟ้าเรียบร้อยแล้วกดปุ่มตกลง จะแสดงหน้าจอดังรูปที่ 3.19 สัญลักษณ์บอร์ด LED จะแสดงผลตามที่ผู้ใช้งานได้เลือกไว้



รูปที่ 3.19 แสดงหน้าจอหลังจากทำการเปิด-ปิดอุปกรณ์ไฟฟ้าแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทางหน้าจอของ Servletrunner จะแสดงสถานะแจ้งว่าผู้ใช้กำลังทำงานกับอุปกรณ์ไฟฟ้า เครื่องไหนอยู่ในขณะนี้



```

servletrunner starting with settings:
port = 8080
backlog = 50
max handlers = 100
timeout = 5000
servlet dir = .\examples
document dir = .\examples
servlet propfile = .\examples\servlet.properties
index: init
read: init
show: init
result: init
Name : ห้องอาหาร; sleep : 60000
Thread ห้องอาหาร
  
```

รูปที่ 3.20 แสดงหน้าจอของ Servletrunner แจ้งสถานะของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สรุปผลปัญหาและข้อเสนอแนะ

4.1 การทำงานของปัญหาพิเศษ

โครงสร้างของระบบในปัญหาพิเศษนี้จะมี 2 ส่วนใหญ่ ๆ คือ

ส่วนที่ 1 เป็นการติดต่อระหว่างเครื่องคอมพิวเตอร์จากข้างนอกบ้าน และเครื่องคอมพิวเตอร์ภายในบ้าน ซึ่งจะมีการส่งข้อมูลการควบคุมเครื่องใช้ไฟฟ้าผ่านระบบเครือข่าย

ส่วนที่ 2 ส่วนนี้เป็นการให้คอมพิวเตอร์ที่บ้านทำการควบคุมการปิดเปิดเครื่องใช้ไฟฟ้าโดยรับส่งข้อมูลผ่านทางพอร์ตขนาน

4.2 ผลของปัญหาพิเศษ

ปัญหาพิเศษหัวข้อนี้เป็นการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านซึ่งมีการเชื่อมต่อกับคอมพิวเตอร์ภายในบ้านผ่านพอร์ตขนาน และเครื่องคอมพิวเตอร์ที่บ้านนั้นจะต้องเชื่อมต่อกับระบบเครือข่ายเพื่อทำการส่งข้อมูลจากภายนอกบ้านมาทำการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน ซึ่งสามารถประยุกต์ไปใช้กับงานต่าง ๆ ได้ เช่น ระบบธุรกิจ งานอุตสาหกรรม เป็นต้น

4.3 ข้อจำกัดของปัญหาพิเศษ

1. การให้บริการแก่ผู้ใช้ยังไม่สมบูรณ์ เช่น ยังขาดการตรวจสอบว่าเครื่องใช้ไฟฟ้าเครื่องใดกำลังขัดข้องอยู่
2. การรับ - ส่งข้อมูลจากคอมพิวเตอร์ ไปเครื่องใช้ไฟฟ้า มีการส่งข้อมูลผ่านพอร์ตขนาน จึงจำกัดจำนวนอุปกรณ์ ไฟฟ้าได้เพียง 8 ชิ้น
3. เนื่องจากมีการรับคำสั่งจากระบบเครือข่าย ดังนั้นเครื่องคอมพิวเตอร์ที่ต่อกับอุปกรณ์ไฟฟ้า จึงต้องต่อกับระบบเครือข่ายตลอดเวลา
4. เนื่องจากอุปกรณ์ไฟฟ้าบางชนิดมีการทำงานที่ค่อนข้างซับซ้อน เช่น เครื่องซักผ้า เป็นต้น ดังนั้นระบบจึงสามารถต่อกับอุปกรณ์ไฟฟ้าได้เพียงบางชนิดเท่านั้น
5. เนื่องจากเครื่องที่บ้านต้องทำหน้าที่เป็นเซิร์ฟเวอร์ ดังนั้นจึงควรเป็นเครื่องที่มีประสิทธิภาพที่ค่อนข้างสูง

4.4 ปัญหาในการทำงานและวิธีการแก้ไข

การใช้ภาษาไทยในการแสดงทางหน้าจอ Browser ของ Servlet ในส่วนนี้สามารถทำการแก้ไขได้โดยการแสดงผลภาษาไทยใน browser จะต้องใช้ String ที่มีการเข้ารหัสแบบ windows-874 character set สังเกตได้ในเว็บที่มี meta tag ดังนี้ จะสามารถแสดงผลภาษาไทยได้ แต่การโปรแกรมในภาษา Java จะใช้การเข้ารหัส String แบบ Unicode ทำให้การรับ ส่งข้อมูลจาก web page แล้วนำไปประมวลผล 3 ข้อ ดังกล่าวจะทำไม่ได้ ต้องทำการ convert จาก windows-874 string เป็น Unicode String เสียก่อน วิธีการดูว่า String เราอยู่ในรูปแบบใด จะสังเกตได้โดย หากสามารถแสดงผลเป็นเว็บภาษาไทยนั้นคือกำลังเข้ารหัสแบบ windows-874 ซึ่ง ASCII code จะมีค่าระหว่าง 161 - 251 (0xA1 - 0xFB ฐาน16) ส่วน Unicode มีค่าระหว่าง 3585-3675 (0xE01 - 0xE5B) เช่น "ก" ใน windows-874 มีค่าเป็น 161 แต่ Unicode มีค่าเป็น 3585 ฉะนั้น การแปลงจาก windows-874 เป็น Unicode จะต้องลบด้วย 3424 (D60)

```
meta http-equiv="Content-Type" content="text/html; charset=windows-874"
```

```
/**
```

```
* Thai charset converter.
```

```
** Problem is :
```

```
** 1.Unicode not support in any Web browser (IE, NN, Opera). To display Thai web page properly you must use single-byte encoding scheme "charset=Windows-874" for display Thai page properly.
```

```
* 2.Java get data from varchar's SQLServer in Unicode format.
```

```
**/
```

```
public class MTThai {
```

```
/**
```

```
* Convert from Unicode string into Thai MS Windows charset string.
```

```
*
```

```
* MS-874 Thai char in 161 - 251 ( 0xA1 - 0xFB )
```

```
* Unicode char in 0xE01 - 0xE5B
```

```
*/
```

```
public static String UnicodeToMS874( String _in ) {
```

```
StringBuffer strTemp = new StringBuffer( _in );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int code;
for( int i = 0; i < _in.length(); i++) {
code = (int) strTemp.charAt(i);
if ( ( 0xE01 <= code ) && ( code <= 0xE5B ) ) {
strTemp.setCharAt( i, (char) ( code - 0xD60 ) );
}
}
return strTemp.toString();
}
/**
 * Convert from Thai MS Windows charset string into Unicode string.
 *
 * MS-874 Thai char in 161 - 251 ( 0xA1 - 0xFB )
 * Unicode char in 0xE01 - 0xE5B
 */
public static String MS874ToUnicode( String _in ) {
StringBuffer strTemp = new StringBuffer( _in );
int code;
for( int i = 0; i < _in.length(); i++) {
code = (int) strTemp.charAt(i);
if ( ( 0xA1 <= code ) && ( code <= 0xFB ) ) {
strTemp.setCharAt( i, (char) ( code + 0xD60 ) );
}
}
return strTemp.toString();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างอุปกรณ์สำหรับการเรียนรู้ การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก

1. NX-2000 ชุดทดลองเพื่อเรียนรู้การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก เป็นชุดทดลองที่จัดเตรียมขึ้นเพื่อตอบสนองความต้องการเรียนรู้ของผู้สนใจด้าน PC-Interfacing โดยจะเน้นไปที่การเชื่อมต่อผ่านพอร์ตนาน หรือพอร์ตเครื่องพิมพ์ และพอร์ตอนุกรมของคอมพิวเตอร์พีซีเป็นหลัก

คุณสมบัติทางเทคนิค

1. LED แสดงผลแบ่งเป็น 2 ส่วนส่วนละ 8 ช่อง
2. มีวงจรบัฟเฟอร์สำหรับขับ LED
3. คอนเน็คเตอร์ P-BUS และ DATA BUS สำหรับเชื่อมต่อกับ P-Board และ S-Board

บอร์ดที่ใช้ทำงานร่วมกันเพื่อทำการทดลองเพิ่มเติม

1. P-Board บอร์ดเชื่อมต่อพอร์ตนาน
2. S-Board บอร์ดเชื่อมต่อพอร์ตอนุกรม
3. EX-03 บอร์ดสวิตช์ 16 ช่อง
4. EX-07 บอร์ดขยายอินพุตเอาต์พุตผ่านระบบบัส I²C

ก่อนใช้ EX-01 บอร์ด LED มอนิเตอร์ 16 ช่อง

ในการเรียนรู้การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น ผู้เรียนจำเป็นต้องเตรียมความพร้อมดังนี้

1. ควรมีความรู้ด้านการเขียนโปรแกรมภาษาสูง อาทิ QBASIC, Visual BASIC, C, VISUAL C, PASCAL, DELPHI หรืออื่นๆ
2. ควรมีความรู้ด้านอุปกรณ์อิเล็กทรอนิกส์พื้นฐานพอสมควร
3. ก่อนการเชื่อมต่อ EX-01 ต้องตรวจสอบให้แน่ใจว่าจุดเชื่อมต่อต่างๆถูกต้อง
4. ทำความเข้าใจในการทำงานของบอร์ดทดลองทั้งหมดให้ดี ก่อนทดลองหรือใช้งาน
5. ศึกษาทฤษฎีและโปรแกรมตัวอย่างที่นำเสนอก่อนการทดลองให้เข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. P-Board บอร์ดเชื่อมต่อพอร์ตนาน

คุณสมบัติทางเทคนิค

- เชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตนาน
- มีวงจรมีเฟอโรป้องกันพอร์ตนาน
- มีวงจรมีแปลงสัญญาณเป็นระบบบัส IC
- คอนเน็กเตอร์สำหรับขยายระบบ แบ่งเป็นอินพุต,เอาต์พุต,DATA BUSและ P-BUS สำหรับ

ต่อบอร์ด EX-Series

- คอนเน็กเตอร์เชื่อมต่อ LCD โมดูลขนาด 16 ตัวอักษร 1 หรือ 2 บรรทัด

บอร์ดขยายระบบเพื่อทดลองเพิ่มเติม

1. EX-01 บอร์ด LED มอนิเตอร์ 16 ช่อง
2. EX-02 บอร์ด LED ตัวเลข 7 ส่วน 4 หลัก
3. EX-03 บอร์ดสวิตช์ 16 ช่อง
4. EX-04 บอร์ด LED คอตเมตริกซ์ 7*20 จุด
5. EX-05 บอร์ดขับสเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์
6. EX-06 บอร์ดขับรีเลย์ 7 ช่อง
7. EX-07 บอร์ดขยายอินพุตเอาต์พุตผ่านระบบบัส I²C
8. EX-08 บอร์ดแปลงสัญญาณอะนาลอกเป็นดิจิตอลและแปลงดิจิตอลเป็นอะนาลอกผ่านระบบบัส I²C

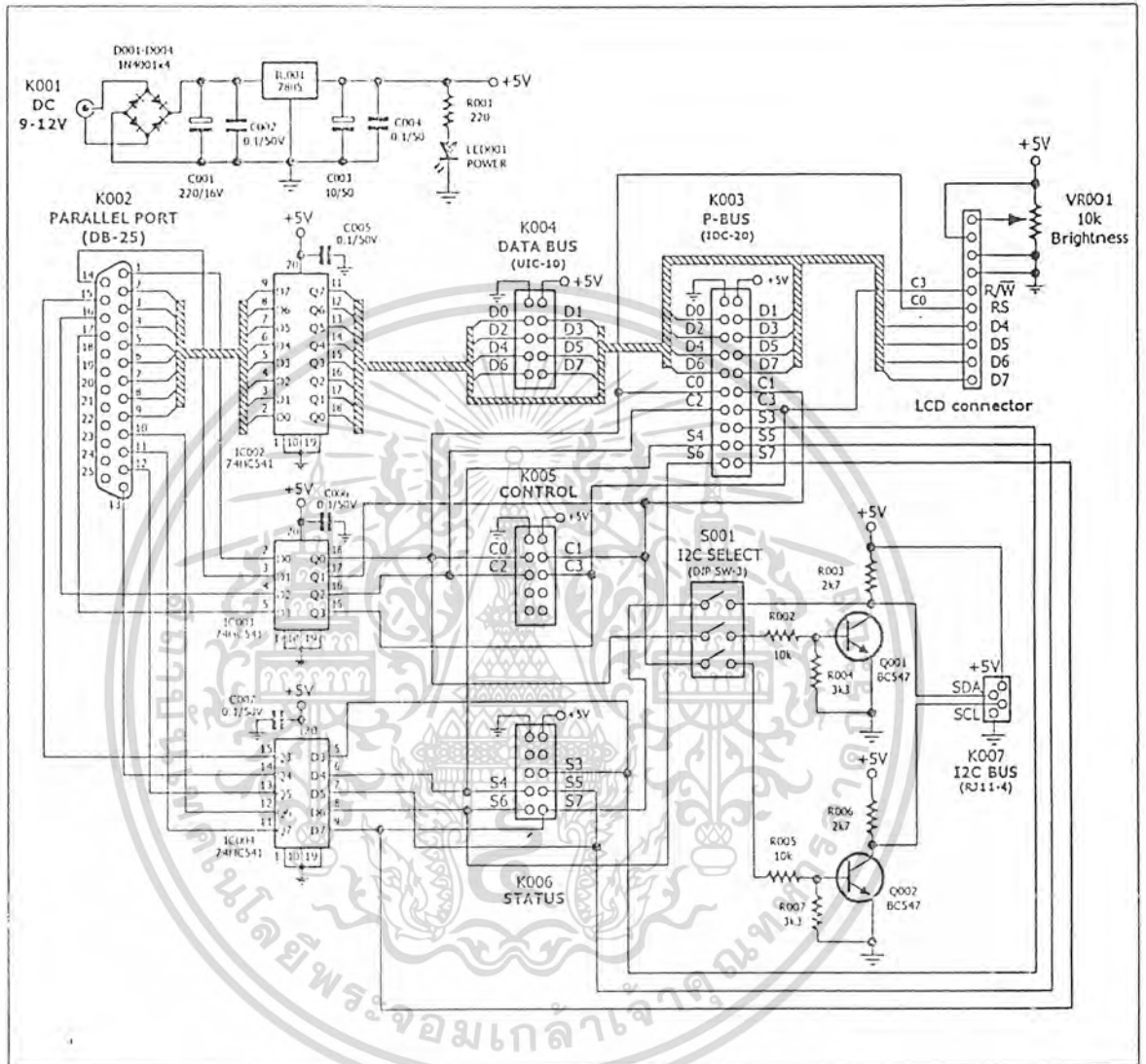
ก่อนใช้ P-Board บอร์ดเชื่อมต่อพอร์ตนาน

ในการเรียนรู้การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้นผู้เรียนจำเป็นต้องมีการเตรียมความพร้อมดังต่อไปนี้

1. ควรมีความรู้ด้านการเขียนโปรแกรมภาษาระดับสูง อาทิ QBASIC, VISUAL BASIC, C, VISIAL C, PASCAL, DELPHI หรืออื่น ๆ
2. ควรมีความรู้ด้านอุปกรณ์อิเล็กทรอนิกส์พื้นฐานพอสมควร
3. ก่อนการเชื่อมต่อ P-Board ต้องตรวจสอบให้แน่ใจว่าจุดเชื่อมต่อต่าง ๆ ถูกต้องและควรใช้อะแดปเตอร์ขนาดไม่เกิน 12 V ในการเป็นแหล่งจ่ายไฟให้แก่ P-Board
4. ทำความเข้าใจในการทำงานของบอร์ดทดลองทั้งหมดให้ดี ก่อนทดลองหรือใช้งาน
5. ศึกษาทฤษฎีและโปรแกรมตัวอย่างที่นำเสนอก่อนการทดลองให้เข้าใจทั้งนี้เพื่อประโยชน์ในด้านเสริมความรู้และช่วยให้การเรียนรู้เป็นไปได้อย่างรวดเร็ว มีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การเชื่อมต่ออุปกรณ์

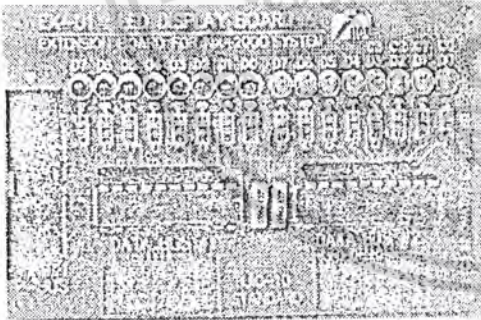
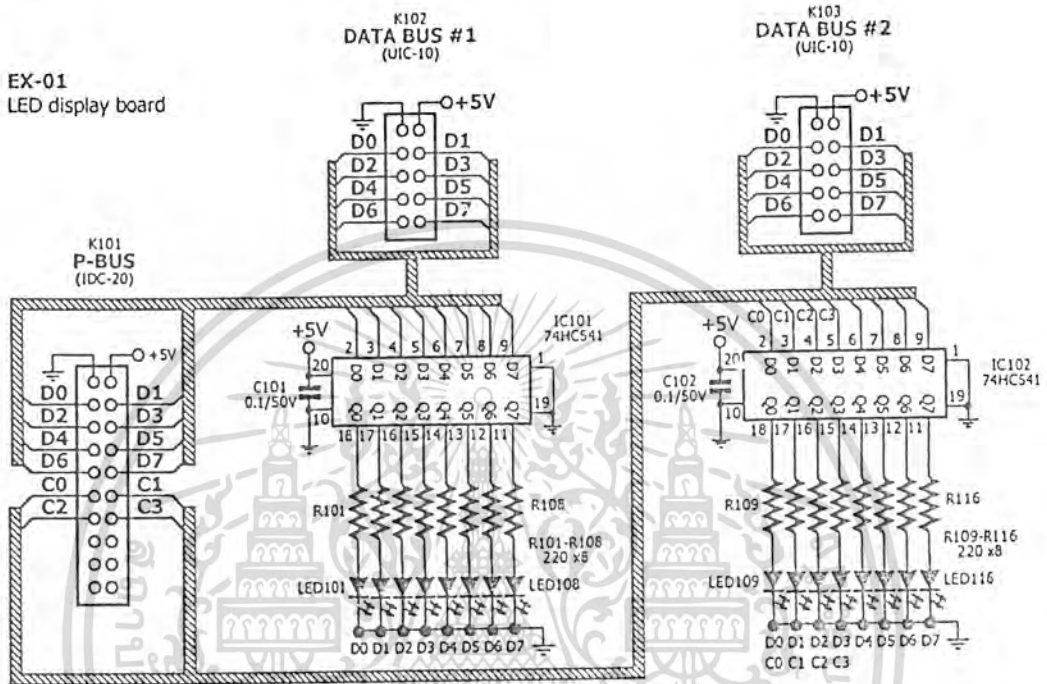


รูปที่ ก-1 วงจรสมบูรณของ P-Board บอร์ดเชื่อมต่อพอร์ทขนาน

บอร์ด EX-01 แสดงวงจรถังในรูป ก-1 สามารถเชื่อมต่อกับ P-Board ผ่านทางคอนเน็กเตอร์ P-BUS และ DATA BUS บนบอร์ดมี LED อยู่ทั้งหมด 16 ดวง แบ่งออกเป็น 2 ชุด ชุดละ 8 ดวง ขับให้ทำงานผ่านทาง ไอซี บัพเฟอร์เบอร์ 74HC541 โดยได้รับไฟเลี้ยงมาจากระบบบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับประโยชน์ของบอร์ด EX-01 คือใช้แสดงผลเอาต์พุตต่างๆของ P-Board เมื่อได้รับการสั่งงานจาก คอมพิวเตอร์ โดยการเชื่อมต่อระหว่างบอร์ด EX-01กับ P-Board จะใช้สายเชื่อมต่อแบบ IDC-10



EX-01 บอร์ด LED มอนิเตอร์ 16 ช่อง

- LED แสดงผลแบ่งเป็น 2 ส่วน ส่วนละ 8 ช่อง
- มีวงจรบัฟเฟอร์สำหรับขับ LED
- คอนเน็คเตอร์ P-BUS และ DATA BUS สำหรับเชื่อมต่อ กับ P-Board และ S-Board

รูป ก-2 วงจรสมบูรณ์ของ EX-01 บอร์ด LED มอนิเตอร์ 16 ช่อง

วงจรของ P-Board แสดงในรูป ก-2 โดยเริ่มจากคอนเน็คเตอร์ K002 แบบ DB-25 ตัวเมีย อันเป็นจุดที่ใช้ต่อเชื่อมกับพอร์ตขนานของคอมพิวเตอร์ ซึ่งได้รับการจัดสรรออกเป็น 3 ส่วนคือ พอร์ต Data, พอร์ต Control , พอร์ต status แต่เราจะกล่าวการใช้งานแค่ในส่วนของ พอร์ต Data เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต Data มีตำแหน่งขาอยู่ที่ขา 2 ถึงขา 9 ใช้ทำหน้าที่เป็นขาเอาต์พุตสัญญาณข้อมูลจะถูกส่งเข้าสู่ไอซีบัฟเฟอร์เบอร์ 74HC541 เพื่อขยายกระแสให้กับขาเอาต์พุต D0-D7 ทั้ง 8 ขา นอกจากนั้นยังทำหน้าที่เป็นตัวป้องกันความเสียหายที่อาจจะเกิดกับพอร์ตขานานอีกด้วย เอาต์พุตจากไอซี 74HC541 จะส่งออกไปยังคอนเน็กเตอร์ DATA BUS ซึ่งมีการจัดขาตามมาตรฐาน UIC-10 และส่งออกไปรวมกับคอนเน็กเตอร์ P-BUS โดยคอนเน็กเตอร์ DATA BUS และ P-BUS จะใช้ประโยชน์ในการเชื่อมต่อกับบอร์ด EX-series ซึ่งเป็นบอร์ดอุปกรณ์อินพุตเอาต์พุต มีทั้งสิ้น 8 บอร์ด

P-Board ใช้แหล่งจ่ายไฟจากภายนอกป้อนเข้ามาทางแจ็กอะแดปเตอร์ ผ่านไดโอดซึ่งต่อกันในลักษณะบริดจ์ไว้เพื่อทำการจัดตัวของแรงดันไฟเลี้ยงใหม่บน P-Board จากนั้นแรงดันจะได้รับการส่งผ่านไปยังวงจรรูเลเตอร์เพื่อแปลงแรงดันให้เท่ากับ +5V สำหรับเป็นไฟเลี้ยงอุปกรณ์บนบอร์ดและเป็นไฟเลี้ยงให้กับวงจรต่อพ่วงต่าง ๆ ที่เชื่อมต่อกันผ่านคอนเน็กเตอร์ P-BUS, DATA BUS และ BUS I²C



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการติดตั้งโปรแกรม

1. การติดตั้งโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน

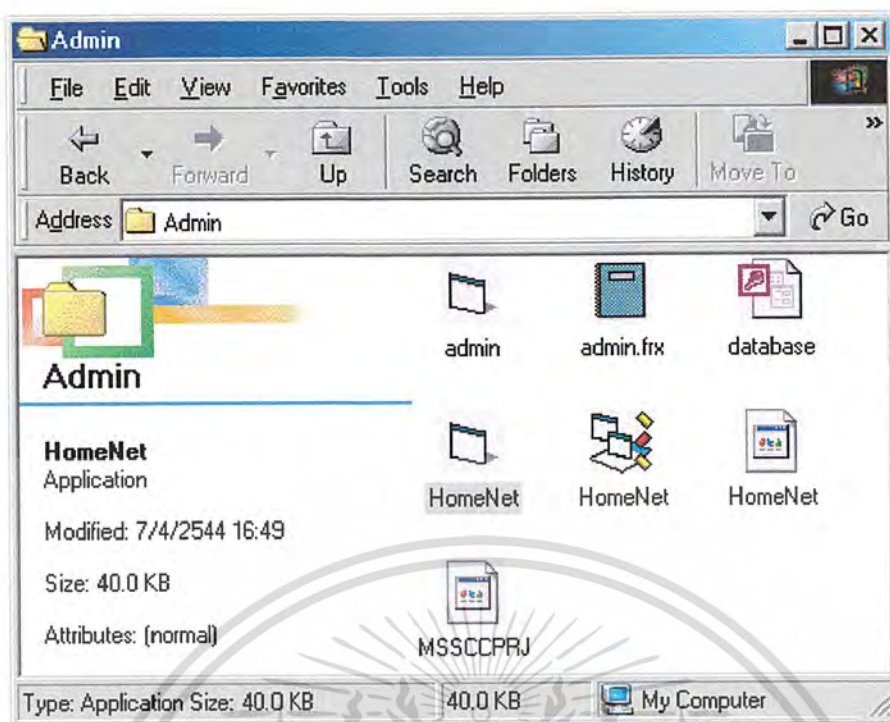
ในการติดตั้งโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้านเพื่อใช้งาน จำเป็นต้องติดตั้งโปรแกรมในฝั่ง Server 2 ส่วนด้วยกัน คือ โปรแกรมในส่วนของ Admin และโปรแกรมในส่วนของ User

1.1 การติดตั้งโปรแกรมในส่วนของ Admin

ในการติดตั้งโปรแกรมส่วนนี้ จะเป็นการใช้งานโปรแกรมเพื่อทำการเพิ่ม, ลบ และแก้ไขข้อมูลในส่วนของผู้ที่จะมาใช้งานโปรแกรมส่วน User ได้ ดังนั้นก่อนที่เราจะใช้งานโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้านจะต้องทำการเพิ่ม Username และ Password ของเราลงไปในฐานะข้อมูลก่อน โดยใช้โปรแกรมในส่วนของ Admin นี้ โดยวิธีการติดตั้งโปรแกรมในส่วนของ Admin ให้ทำตามขั้นตอนดังนี้

- 1) ทำการ Copy ไฟลเดอร์ที่ชื่อว่า Admin ใส่งใน Drive C: ภายในโฟลเดอร์จะมีไฟล์ที่ชื่อ HomeNet ที่เป็นไฟล์ Application
- 2) เรียกใช้โปรแกรมโดยคลิกที่ Icon ที่มีชื่อว่า HomeNet ดังรูปที่ ข-1
- 3) ผู้ใช้งานสามารถทำการสร้าง Shortcut ไปวางไว้บน Desktop เพื่อความสะดวกในการใช้งาน โดยคลิกขวาที่ Icon HomeNet จากนั้นเลือก Create Shortcut จะมี Icon ที่มีชื่อว่า Shortcut to HomeNet เพิ่มขึ้นมา
- 4) หลังจากนั้นให้คลิกขวาที่ Icon Shortcut to HomeNet แล้วเลือก Cut จากนั้นคลิกขวาบน Desktop เราก็จะมี Shortcut อยู่บน Desktop ให้เรียกใช้งานได้สะดวกขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-1 แสดง Icon HomeNet ภายในโฟลเดอร์ Admin

1.2 การติดตั้งโปรแกรมในส่วนของผู้ใช้

ในการติดตั้งโปรแกรมส่วนนี้ จะเป็นการใช้งานโปรแกรมควบคุมอุปกรณ์ไฟฟ้าภายในบ้านทางฝั่ง Client ดังนั้นส่วนต่างๆ ที่จะต้องติดตั้งที่เครื่อง Server มีดังนี้

- 1) ทำการลงโปรแกรม JDK1.2 ที่มีมาให้ในแผ่นโปรแกรม
- 2) ทำการลงโปรแกรม JSDK2.0 โดย Copy โฟลเดอร์ที่ชื่อว่า JSDK2.0 ลงใน Drive C:
- 3) จากนั้นลงโปรแกรม Personal Web Server โดยเลือกโฟลเดอร์ PWS แล้วคลิกที่ Setup
- 4) จากนั้น Copy ไฟล์ชื่อ parport.dll ลงใน Directory C:\jdk1.2\bin
- 5) แล้ว Copy ทุกไฟล์ในโฟลเดอร์ wwwroot ลงใน C:\inetpub\wwwroot ซึ่งเป็น home directory ของ PWS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ทำการติดตั้งโปรแกรมต่าง ๆ ลงในเครื่อง Server เรียบร้อยแล้ว ให้ทำการเซตส่วนโปรแกรมต่าง ๆ ดังนี้

1. เซต PATH และ CLASSPATH เพื่อให้สามารถใช้งานโปรแกรม JDK และ JSDK ได้

สำหรับระบบปฏิบัติการ Windows 95/98

ให้เปิดไฟล์ชื่อ Autoexec.bat ที่อยู่ใน Drive C: ทำการแก้ไขโดยเพิ่ม

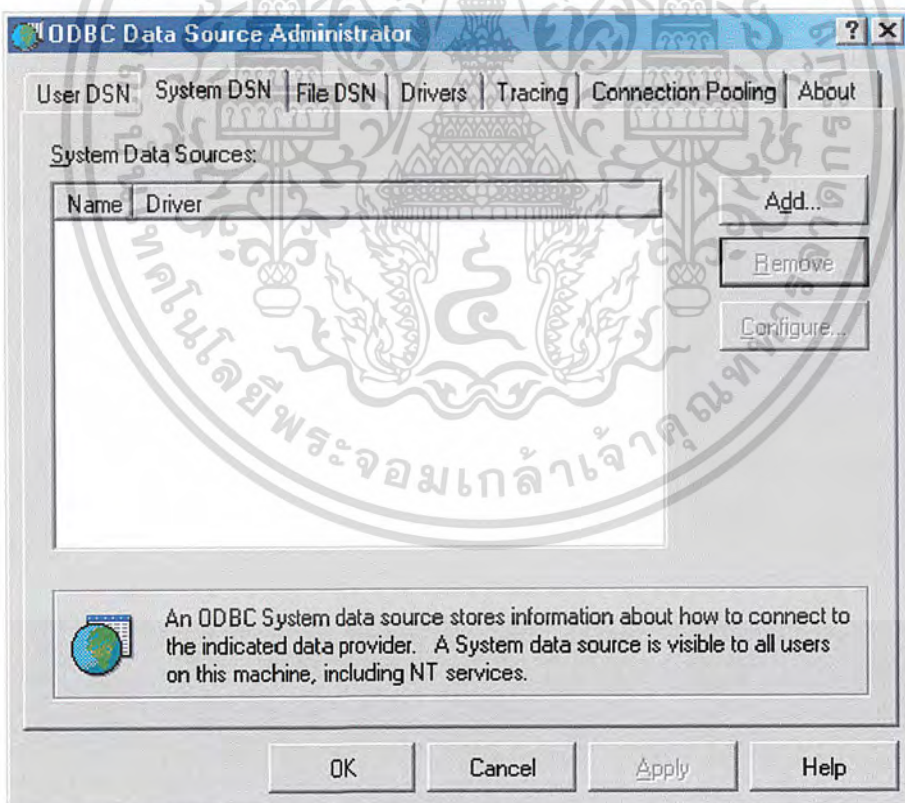
SET PATH= .;C:\JDK1.2\BIN;

SET CLASSPATH=.;c:\jdk1.2\lib\classes.zip;c:\j sdk2.0\lib\j sdk.jar;C:\j sdk2.0\examples\class;

2. เซต ODBC Data Source Administrator เพื่อให้โปรแกรมภาษา Java สามารถติดต่อกับฐานข้อมูล Microsoft Access ได้ ตามขั้นตอนต่อไปนี้

1) ทำการเลือก Start → Settings → Control Panel

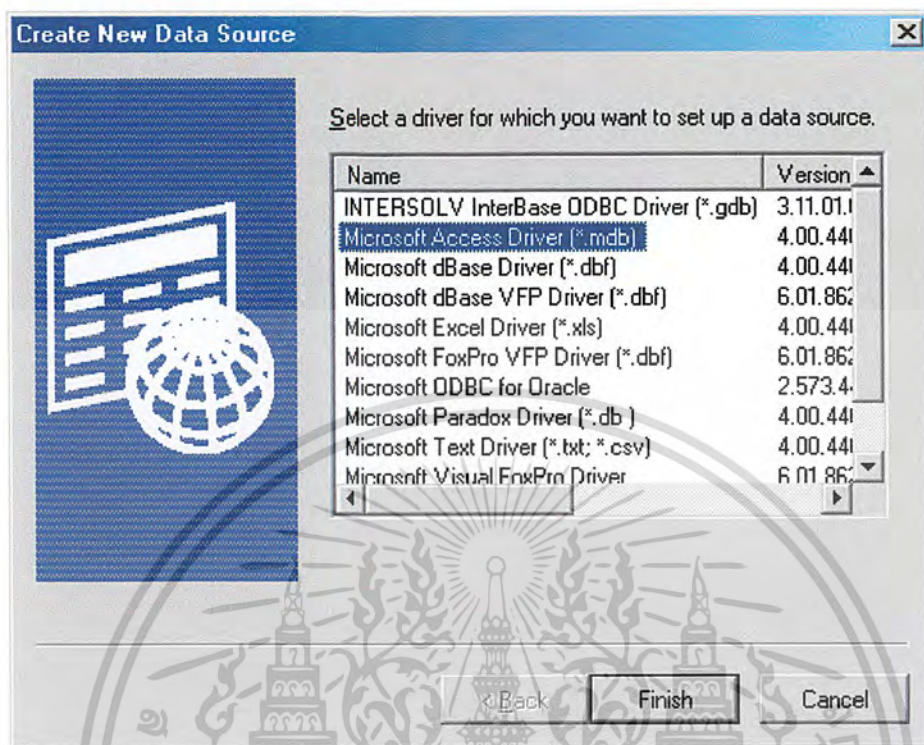
2) คลิกเลือก Icon ODBC Data Source (32bit) และเลือกที่ Tab System DSN จะแสดงหน้าจอ ดังรูปที่ ข-2



รูปที่ ข-2 แสดงหน้าจอของ Tab System DSN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

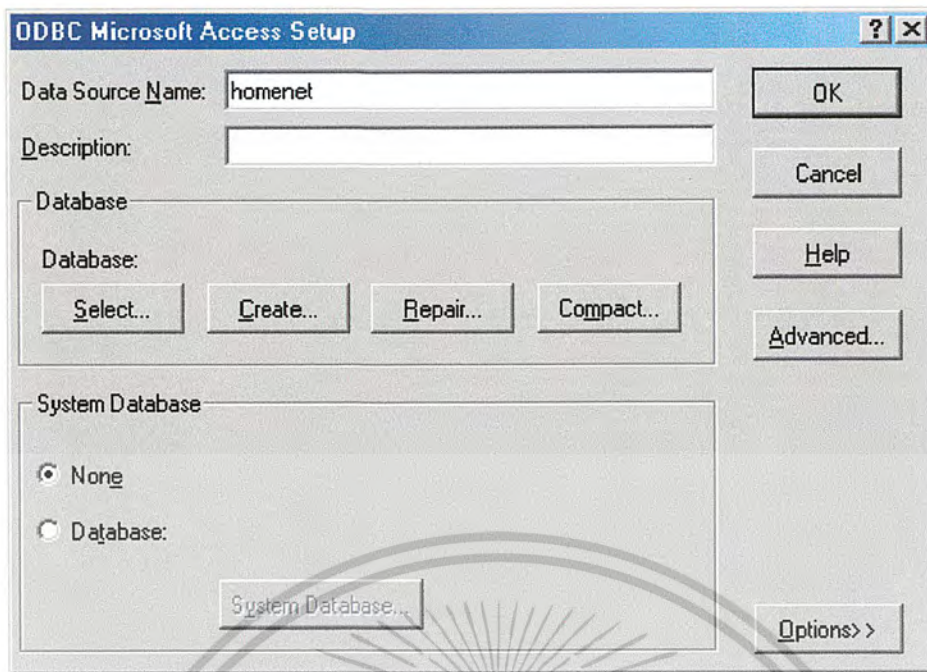
3) ให้กดปุ่ม Add... และเลือก Microsoft Access Driver (*.mdb) ดังรูปที่ ข-3 แล้วกดปุ่ม Finish จะแสดงหน้าจอ ODBC Microsoft Access Setup ดังรูปที่ ข-4



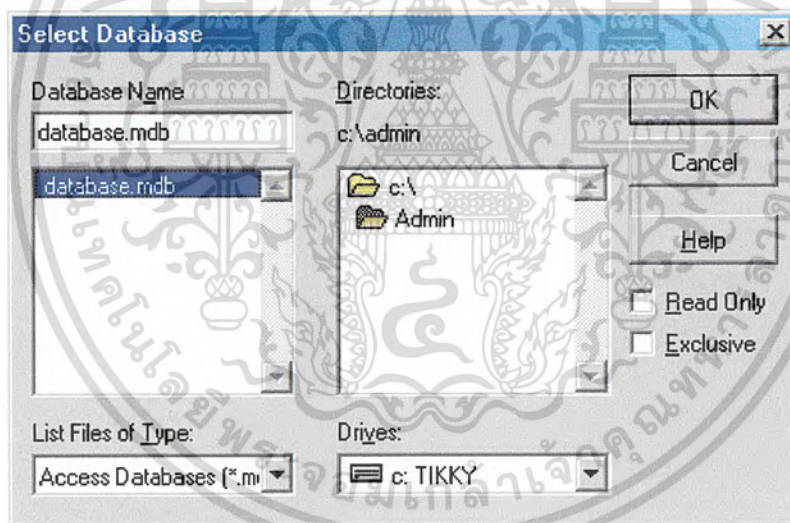
รูปที่ ข-3 แสดงหน้าจอ Create New Data Source

4) ให้พิมพ์ homenet ที่ Data Source Name: แล้วกดปุ่ม Select.. เพื่อเลือกไฟล์ฐานข้อมูลดังรูปที่ ข-5 โดยเลือก Directories ไปที่ c:\Admin จากนั้นจะขึ้นชื่อ Database Name ทางด้านซ้าย ให้คลิกเลือก database.mdb จากนั้นกด OK กลับมายังหน้าจอ ODBC Microsoft Access Setup

5) กดปุ่ม Advanced.. จะแสดงหน้าจอ Set Advanced Options ดังรูปที่ ข-6 จากนั้นให้พิมพ์ admin ตรง Login name: และ Password: แล้วกด OK

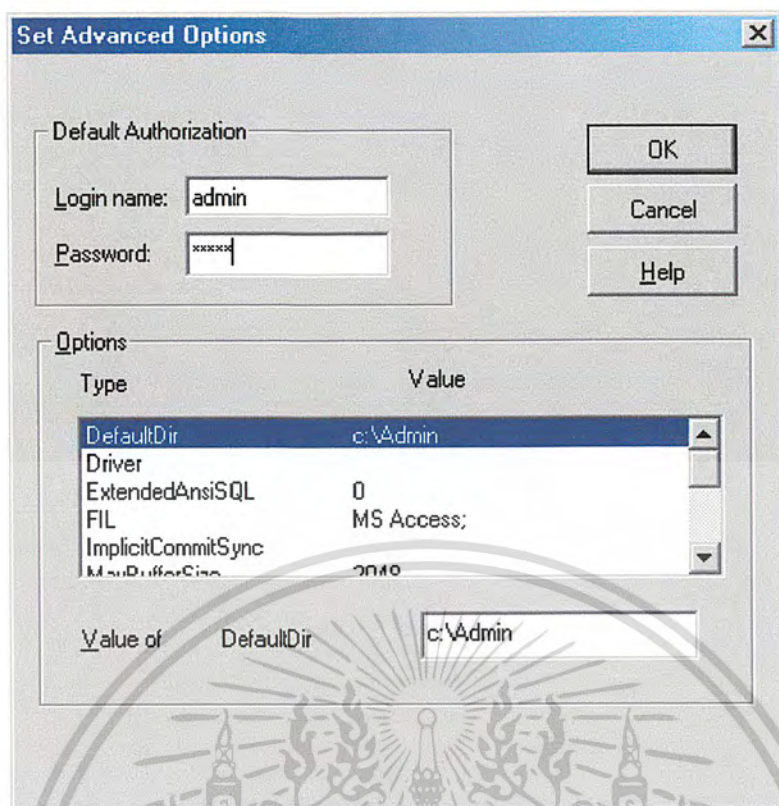


รูปที่ ข-4 แสดงหน้าจอ ODBC Microsoft Access Setup



รูปที่ ข-5 แสดงหน้าจอเลือกฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-6 แสดงหน้าจอ Set Advanced Options

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภัคดีวัฒนากุล และจำลอง ครุอุตสาหกรรม. 2542. Visual Basic 6.0 ฉบับโปรแกรมเมอร์. พิมพ์ครั้งที่ 5.

กรุงเทพฯ : หจก.ไทยเจริญการพิมพ์

พัฒน์พงษ์ โพธิ์ปัสสา. 2543. "Java Story : เรื่องของเซิร์ฟเล็ต." 5(4) : 84-87.

ณัฐพล ทวีทรัพย์เอนก. 2542. "จาวาเซิร์ฟเล็ตกับเว็บเซิร์ฟเวอร์." : 108-112.

บริษัทอินโนเวตีฟ อี๊กเพอริเมนต์ จำกัด. "P-Board บอร์ดเชื่อมต่อพอร์ตขนาน." หน้า 1-28.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้