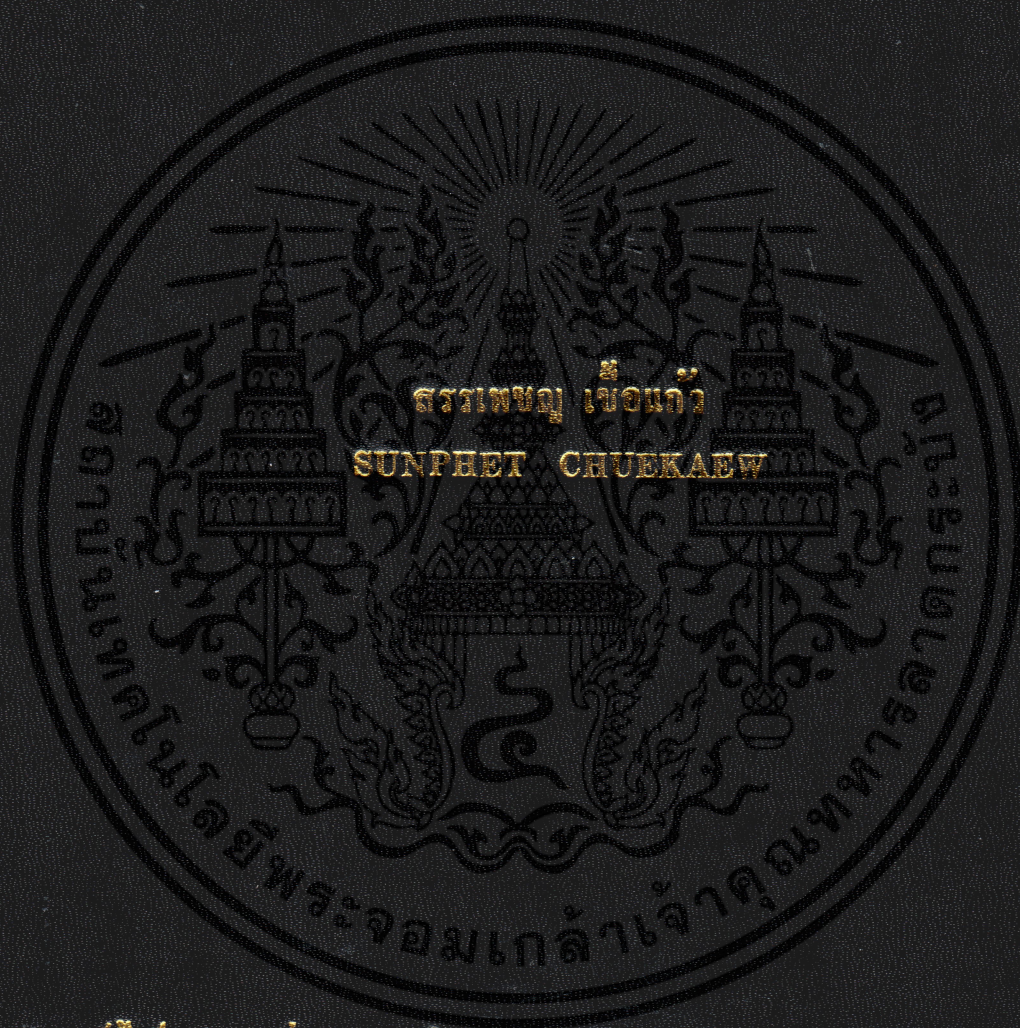


การควบคุมกระบวนการระยะไกลผ่านอินเทอร์เน็ต
TELEMETERING PROCESS CONTROL VIA INTERNET



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคณะศึกษาศาสตร์ปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-137-1

การควบคุมกระบวนการระยะไกลผ่านอินเทอร์เน็ต

TELEMETERING PROCESS CONTROL VIA INTERNET



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2544

ISBN 974-648-137-1

เลขหมู่.....
เลขทะเบียน 39752

วัน, เดือน, ปี 21 ส.ย. 2544

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีไปใช้

TELEMETERING PROCESS CONTROL VIA INTERNET



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2001

ISBN 974-648-137-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2001

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การควบคุมกระบวนการระยะไกลผ่านอินเทอร์เน็ต
 TELEMETERING PROCESS CONTROL VIA INTERNET

ชื่อนักศึกษา นายสรรเพชญ์ เชื้อแก้ว

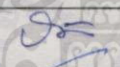
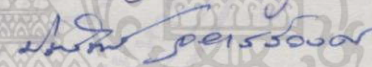

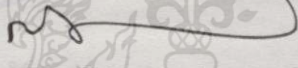
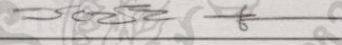
รหัสประจำตัว 42061075

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.วันชัย ธีรวัชร

อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ผศ.ประสิทธิ์ จุลเสรีวงศ์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.ปิติเขต	สุรักษา	
ผศ.ประสิทธิ์	จุลเสรีวงศ์	
รศ.พิพัฒน์	เลาหสงคราม	
รศ.กิตติ	ศิริเศรษฐ	
รศ.ดร.วันชัย	ธีรวัชร	

วัน/เดือน/ปี ที่สอบ 1 มีนาคม 2544 เวลา 12.00-13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

 (รศ.ดร.บุญวัฒน์ อตฺห)
 คณบดีบัณฑิตวิทยาลัย

วันที่..... ๑3เดือน..... พ.ศ. ๒๕๔๔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การควบคุมกระบวนการระยะไกลผ่านอินเทอร์เน็ต
นักศึกษา	นายสรรเพชญ เชื้อแก้ว
รหัสประจำตัว	42061075
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.วันชัย ธีรรัฐจา
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	ผศ.ประสิทธิ์ จุลเสรีวงศ์

บทคัดย่อ

วิทยานิพนธ์นี้เป็นการเสนอการควบคุมกระบวนการจากเบรเซอร์ผ่านทางอินเทอร์เน็ตเพื่อใช้ในการควบคุมกระบวนการผ่าน ไปยังห้องควบคุมกระบวนการ กระบวนการที่ใช้ในวิทยานิพนธ์นี้จะเป็นกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิภายในตู้อบ โดยเครื่องควบคุมที่ใช้เป็นระบบพีไอดี ซึ่งผู้ควบคุมสามารถที่จะควบคุมจากที่ใดก็ได้โดยไม่มีข้อจำกัดเรื่องระยะทาง วิทยานิพนธ์ได้นำเสนอถึงแนวคิดในการจัดลำดับในการควบคุมผ่านทางอินเทอร์เน็ตและการบันทึกข้อมูลที่ได้จากการทดลองผ่านทางอินเทอร์เน็ตตลอดจนการนำเอาข้อมูลที่ถูกต้องที่ใกล้เคียงฐานเวลาเดียวกันกับเวลาของเครื่องควบคุมของเซิร์ฟเวอร์มาแสดงผลบนเบรเซอร์

Thesis Title	Telemetry process control via Internet
Student	Mr.Sunphet Chuekaew
Student ID.	42061075
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2001
Thesis Advisor	Assoc.Prof.Dr.Vanchai Riewruja
Thesis Co-Advisor	Assist.Prof. Prasit Julsereewong

ABSTRACT

This thesis presents the remote control through browser over the internet. The process control plants are the water level control plant and the temperature control plant by using PID controller. Furthermore, they can be controlled anywhere without the restriction in term of distance. This thesis describes the concepts of order for controlling over the internet and the data record in the plant is done through the internet. Finally, the precise data maintained nearly real time as the control server is presented on the browser.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดี ด้วยคำแนะนำและปรึกษาเกี่ยวกับแนวคิด หลักการ และการประยุกต์การใช้งาน ในด้านการออกแบบโปรแกรมสำหรับใช้ควบคุมบนเครือข่ายอินเทอร์เน็ต ให้มีความถูกต้องจาก รศ.ดร.วันชัย ธีรรัฐา ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ และการออกแบบ โปรแกรมที่ใช้ในการควบคุมของเครื่องควบคุม รวมทั้งให้คำแนะนำในเรื่องของการใช้อุปกรณ์ เครื่องมือวัดต่าง ๆ ที่ใช้ในการทดลองให้ใช้งานได้อย่างถูกต้องจาก ผศ.ประสิทธิ์ จุลเสรีวงศ์ ซึ่งเป็น อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ทำให้ผู้วิจัยสามารถแก้ไขปัญหาจนกระทั่งงานวิจัยเสร็จสมบูรณ์

ขอขอบคุณภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมที่ได้เอื้อเพื่อชุดทดลองควบคุม ระดับน้ำและชุดทดลองควบคุมอุณหภูมิ พร้อมทั้งอำนวยความสะดวกต่าง ๆ ทำให้ผู้วิจัยสามารถ ทำงานวิจัยได้สำเร็จด้วยดี

ขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้

สุดท้ายขอขอบคุณ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติภายใต้โครงการทุน ส่งเสริมวิชาชีพนักวิจัย ประเภทรางวัลพัฒนาวิชาชีพนักวิจัย และศูนย์เทคโนโลยีอิเล็กทรอนิกส์และ คอมพิวเตอร์แห่งชาติภายในโครงการสนับสนุนการออกแบบวงจรรวมในสถาบันศึกษา ที่ให้ทุนและ เครื่องมือสนับสนุนในการทำวิทยานิพนธ์นี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

สรรเพชญ์ เชื้อแก้ว

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์	1
1.3 ขอบเขตของวิทยานิพนธ์	2
บทที่ 2 อินเทอร์เน็ต (Internet)	3
2.1 บทนำ.....	3
2.2 ประวัติและความเป็นมาของอินเทอร์เน็ต	3
2.3 เวิลด์ไวด์เว็บ (WWW : World - Wide – Web)	4
2.4 เบราเซอร์ (Browser)	4
2.5 โพรโทคอล (Protocol)	5
2.6 เลขประจำเครื่อง (IP Address)	5
2.7 พอร์ตบนอินเทอร์เน็ต (Ports)	6
2.8 การเชื่อมโยงกับซ็อกเก็ต (Socket) ของจาวา	6
2.9 บทสรุป.....	6
บทที่ 3 ตัวควบคุมแบบป้อนกลับ	7
3.1 บทนำ.....	7
3.2 ประวัติและความเป็นมาของตัวควบคุมแบบป้อนกลับ	7
3.3 ตัวควบคุมแบบเปิด - ปิด (On – Off Controller)	8
3.4 ตัวควบคุมแบบ PID (PID Controller)	9
3.5 การควบคุมแบบป้อนกลับ	10
3.5.1 การควบคุมแบบ Proportional (P).....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.5.2 การควบคุมแบบ Integral (I) และการควบคุมแบบ PI.....	12
3.5.3 การควบคุมแบบ Derivative (D) การควบคุมแบบ PD และการควบคุมแบบ PID	14
3.5.4 สมการ PID สำหรับการคำนวณด้วยคอมพิวเตอร์	16
3.5.5 ผลตอบสนองของกระบวนการต่อการควบคุมแบบป้อนกลับ	18
3.6 บทสรุป	20
บทที่ 4 ภาษาจาวา (JAVA)	21
4.1 บทนำ.....	21
4.2 ความเป็นมาของภาษาจาวา	21
4.3 ความสามารถของภาษาจาวา	22
4.4 การประยุกต์ใช้งาน	22
4.5 คอมพิวเตอร์จำลองของจาวา Java Virtual Machine	24
4.6 หลักการทำงานของภาษาจาวา	25
4.7 โปรแกรมเบราเซอร์ที่รองรับการทำงานของจาวา	29
4.8 ความแตกต่างระหว่าง Java Applet และ Java Application	30
4.9 ความแตกต่างระหว่าง Java และ JavaScript	31
4.10 บทสรุป	32
บทที่ 5 จาวากับการติดต่อฐานข้อมูล	33
5.1 บทนำ.....	33
5.2 ระบบไคลเอ็นท์เซิร์ฟเวอร์.....	33
5.3 การเชื่อมต่อฐานข้อมูลด้วย ODBC	34
5.4 ทฤษฎีการทำงานของ ODBC	36
5.4.1 ข้อดีของการติดต่อโดยใช้ ODBC	36
5.4.2 องค์ประกอบของ ODBC	36
5.4.2.1 Application	37
5.4.2.2 ไดรฟ์เวอร์	37

สารบัญ(ต่อ)

	หน้า
5.5 การเชื่อมต่อฐานข้อมูลด้วย JDBC	38
5.6 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC	39
5.6.1 Two-Tier model	39
5.6.2 Three-Tier model	39
5.7 โครงสร้างของ JDBC	40
5.8 รูปแบบของ JDBC ไดรฟ์เวอร์	41
5.8.1 JDBC/ODBC bridge	41
5.8.2 Native-API , Partly Java driver	43
5.8.3 Network-protocol , all –Java driver	43
5.8.4 Native-protocol , all-Java driver	46
5.9 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อฐานข้อมูล	47
5.10 บทสรุป	49
บทที่ 6 การออกแบบและหลักการทํางาน	50
6.1 บทนำ.....	50
6.2 การออกแบบระบบเซิร์ฟเวอร์	50
6.2.1 ระบบเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการ	51
6.2.2 ระบบเซิร์ฟเวอร์เพื่อให้บริการอินเทอร์เน็ต	51
6.3 การกำหนดสิทธิในการควบคุม	52
6.4 การอินเตอร์เฟซฮาร์ดแวร์กับคอมพิวเตอร์	53
6.5 วงจรอินเตอร์เฟซ	54
6.5.1 ส่วนอินเตอร์เฟซ (Interfacing Unit)	54
6.5.1.1 การส่งคำสั่งควบคุม (Control Word) ให้กับไอซี 8255	56
6.5.1.2 การรับข้อมูลจากอุปกรณ์ภายนอก	57
6.5.1.3 การส่งข้อมูลให้กับอุปกรณ์ภายนอก	57
6.5.2 ส่วนอุปกรณ์อินพุต/เอาต์พุต (Input/Output Unit)	58
6.5.2.1 A/D Converter	58
6.5.2.2 D/A Converter	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
6.5.3 วงจรแปลงแรงดันเป็นกระแสไฟฟ้า (V/I Converter).....	62
6.6 การออกแบบโปรแกรม	63
6.6.1 โปรแกรม Delphi	63
การส่งข้อมูลระหว่างเซิร์ฟเวอร์กับไคลเอนท์	69
6.6.1.1 การส่งถ่ายข้อมูลของเซิร์ฟเวอร์ด้วยโปรแกรม Java Application.....	69
6.6.1.2 การรับข้อมูลของไคลเอนท์ด้วยโปรแกรม Java Applet	71
6.6.2 การออกแบบฐานข้อมูลเพื่อแสดงผลกราฟ	71
6.6.3 โปรแกรมเพิ่มคิวของผู้ที่ต้องการควบคุม	72
6.6.4 โปรแกรมตรวจสอบคิวเพื่อให้สิทธิในการควบคุม	75
6.6.5 โปรแกรมสำหรับยกเลิกการควบคุม	76
6.6.6 โปรแกรมสำหรับการบันทึกผลการควบคุม	76
6.7 บทสรุป	77
บทที่ 7 ผลการทดลอง	79
7.1 การควบคุมกระบวนการควบคุมระดับน้ำ.....	80
7.2 การควบคุมกระบวนการควบคุมอุณหภูมิในตู้อบ.....	81
บทที่ 8 สรุปผลการวิจัยและข้อเสนอแนะ	83
เอกสารอ้างอิง	84
ภาคผนวก.....	85
ภาคผนวก ก. โปรแกรมที่พัฒนาด้วย Java Application	86
ภาคผนวก ข. โปรแกรมที่พัฒนาด้วย Java Applet	94
ภาคผนวก ค. ผลงานวิจัยที่ได้รับการตีพิมพ์	124
ประวัติผู้เขียน	132

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
3.1 การควบคุมแบบเปิด - ปิด	8
3.2 ระบบควบคุมการไหล	9
3.3 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ	9
3.4 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ K_p)	11
3.5 การควบคุมแบบ P ในทางปฏิบัติ	12
3.6 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$	14
3.7 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด	18
3.8 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ P เมื่อมีการเปลี่ยนแปลงค่า K_p	18
3.9 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อเปลี่ยนแปลงค่า T_i และ K_p	19
3.10 ผลตอบสนองของกระบวนการต่อการควบคุมแบบพีไอดีเมื่อมีการเปลี่ยนแปลงค่า T_d	19
4.1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน	26
4.2 การประมวลผลของโปรแกรมภาษาจาวา	26
5.1 องค์ประกอบของ ODBC	37
5.2 Two-Tier model	39
5.3 Three-Tier model	40
5.4 ระดับการเชื่อมต่อของ JDBC API	40
5.5 แสดงรูปแบบของ JDBC	41
5.6 การติดต่อ JDBC โดยรูปแบบที่ 1 JDBC-ODBC Bridge	42
5.7 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API, partly Java driver	43
5.8 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver ยกตัวอย่างของ dbAnywhere, Sequelink	44
5.9 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver ยกตัวอย่างของ WebLogic	45
5.10 การติดต่อ JDBC โดยที่รูปแบบที่ 4 Native-protocol, all-Java driver	46
6.1 แสดงการเชื่อมต่อคอมพิวเตอร์ของผู้ควบคุมผ่านทางเครือข่ายอินเทอร์เน็ตหรือเครือข่าย อินทราเน็ตเพื่อควบคุมกระบวนการ	50
6.2 แสดงโครงสร้างระบบเซิร์ฟเวอร์ของงานวิจัย	52
6.3 แสดงแบบจำลองการอินเทอร์เน็ตเฟสกับกระบวนการ	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์

ปัจจุบันอินเทอร์เน็ต (Internet) ได้ก้าวหน้าอย่างรวดเร็วเป็นที่นิยมใช้กันทั่วโลกและมีแนวโน้มที่จะได้รับความนิยมเพิ่มขึ้นเรื่อย ๆ อันเป็นผลมาจากอินเทอร์เน็ตให้ประโยชน์มากมาย ใช้งานง่ายตลอดจนการเขียนโปรแกรมเพื่อสนับสนุนงานด้านต่าง ๆ ก็กระทำได้ง่าย ซึ่งได้นำความสะดวกสบายมาสู่ผู้ใช้ ไม่ว่าจะเป็นการ ค้นหาข้อมูลข่าวสาร การซื้อขายสินค้าหรือบริการผ่านทางอินเทอร์เน็ต (E-Commerce) การบันเทิง ฯลฯ การดำเนินการต่าง ๆ บนอินเทอร์เน็ตเป็นการใช้จ่ายในต้นทุนที่ต่ำเพราะเพียงเราใช้คอมพิวเตอร์เพื่อติดต่อเข้าสู่เครือข่ายอินเทอร์เน็ต ก็สามารถดำเนินการต่าง ๆ ที่ต้องการได้

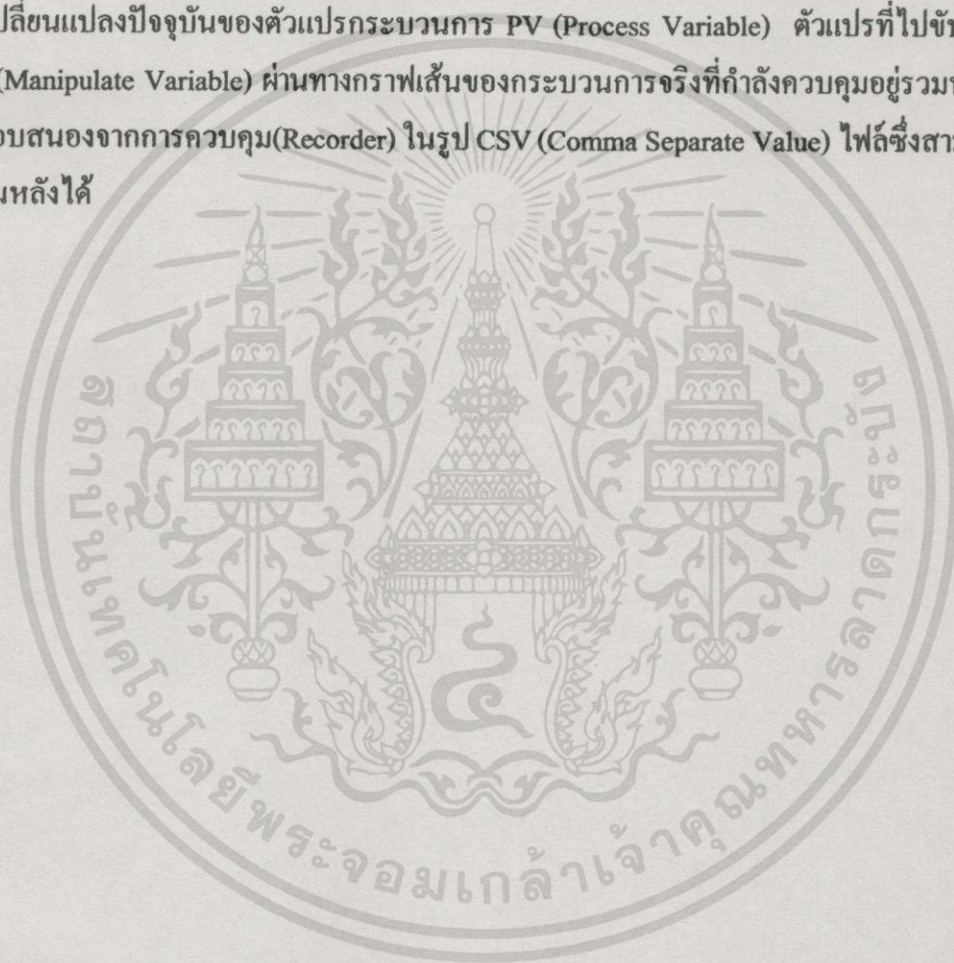
งานวิจัยนี้เป็นงานที่พัฒนาวิธีการเพื่อที่จะสามารถควบคุมกระบวนการ (Process) ผ่านทางอินเทอร์เน็ต โดยไม่จำเป็นต้องเดินทางไปยังบริเวณที่เครื่องควบคุมติดตั้งอยู่เพียงอย่างเดียว ซึ่งการควบคุมกระบวนการผ่านทางอินเทอร์เน็ตจะสามารถเห็นผลการตอบสนองที่ใกล้เคียงกับค่าเวลาจริง (Real Time) เสมือนได้มาควบคุมกับกระบวนการจริงที่เครื่องควบคุมตั้งอยู่

1.2 วัตถุประสงค์ของวิทยานิพนธ์

งานวิจัยชิ้นนี้มีจุดประสงค์หลักเพื่อควบคุมกระบวนการระดับน้ำและกระบวนการควบคุมอุณหภูมิผ่านทางเครือข่ายอินเทอร์เน็ตโดยสามารถควบคุมผ่านทางเบราว์เซอร์ทั่ว ๆ ไปได้แก่ Internet Explorer หรือ Netscape Communicator บนอินเทอร์เน็ตซึ่งถือได้ว่าเป็นเบราว์เซอร์ที่มีความนิยมมากที่สุดในปัจจุบันจึงเป็นการง่ายต่อการใช้งานของผู้ควบคุมทั่วไปเนื่องจากไม่ต้องหาโปรแกรมเฉพาะตัวอื่นมาใช้งาน พร้อมทั้งจัดลำดับและบริหารคิวในการให้สิทธิเพื่อควบคุมกระบวนการสำหรับผู้ที่ต้องการควบคุมกระบวนการสามารถควบคุมได้เป็นไปตามลำดับคิวที่เชื่อมต่อเข้าสู่เซิร์ฟเวอร์ ทำให้ผู้ที่ต้องการควบคุมสามารถควบคุมกระบวนการได้อย่างทั่วถึงทุกคน ส่วนผู้ควบคุมหลักจะอยู่ที่เครื่องเซิร์ฟเวอร์ที่เครื่องควบคุมตั้งอยู่

1.3 ขอบเขตของวิทยานิพนธ์

เป็นการพัฒนาวิธีการต่าง ๆ เพื่อให้เบรเซอร์จากคอมพิวเตอร์ของผู้ควบคุมกระบวนการสามารถรับส่งข้อมูลในการควบคุมระยะไกลไปยังเซิร์ฟเวอร์ที่เครื่องควบคุมติดตั้งอยู่ได้ พร้อมทั้งการจำลองการที่จะเข้ามาควบคุมกระบวนการให้เกิดประโยชน์มากที่สุด ในส่วนประเภทของเครื่องควบคุมกระบวนการ เป็นตัวควบคุมพีไอดีที่ผู้ควบคุมสามารถปรับค่าพารามิเตอร์ K_p , T_i , T_d และค่าเป้าหมาย SP (Setpoint) ได้เองจากทางเบรเซอร์ของผู้ควบคุมเอง รวมทั้งสามารถเห็นการเปลี่ยนแปลงปัจจุบันของตัวแปรกระบวนการ PV (Process Variable) ตัวแปรที่ไปจับเอาต์พุต MV (Manipulate Variable) ผ่านทางกราฟเส้นของกระบวนการจริงที่กำลังควบคุมอยู่รวมทั้งบันทึกผลตอบสนองจากการควบคุม(Recorder) ในรูป CSV (Comma Separate Value) ไฟล์ซึ่งสามารถเปิดดูย้อนหลังได้



บทที่ 2

อินเทอร์เน็ต (Internet)

2.1 บทนำ

อินเทอร์เน็ต คือเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่มีคอมพิวเตอร์เชื่อมต่อกันทั่วโลก โดยมีมาตรฐานในการรับส่งข้อมูลระหว่างกันเป็นหนึ่งเดียวโดยอาศัยโปรโตคอลทีซีพี/ไอพีซึ่งคอมพิวเตอร์แต่ละเครื่องสามารถรับส่งข้อมูลในรูปแบบต่าง ๆ เช่น ตัวอักษร, ภาพและเสียงได้ รวมทั้งค้นหาข้อมูลจากที่ต่าง ๆ ได้อย่างรวดเร็ว การที่มีมาตรฐานของการรับส่งข้อมูลที่เป็นแบบเดียวกันทำให้สามารถเชื่อมต่อคอมพิวเตอร์คนละชนิดหรือคนละแบบบนเครือข่ายอินเทอร์เน็ตได้ในบทรนี้จะเป็นการแนะนำให้รู้จักความเป็นมาและสิ่งที่เกี่ยวข้องกับอินเทอร์เน็ต

2.2 ประวัติและความเป็นมาของอินเทอร์เน็ต

อินเทอร์เน็ตเป็นเครือข่ายที่มีพัฒนาการมาจาก อาร์พาเน็ต (ARPAnet) ซึ่งเป็นเครือข่ายคอมพิวเตอร์ภายใต้ความรับผิดชอบของ หน่วยงานโครงการวิจัยขั้นสูง (Advanced Research Projects Agency) หรือเรียกชื่อย่อว่า อาร์พา (ARPA) ในสังกัดกระทรวงกลาโหมของสหรัฐอเมริกา อาร์พาเน็ตในขั้นต้นเป็นเพียงเครือข่ายทดลองที่ตั้งขึ้นเพื่อสนับสนุนงานวิจัยด้านการทหาร

ในปี พ.ศ. 2500 เมื่อสหภาพโซเวียตได้ส่งดาวเทียมสปุตนิกขึ้นสู่วงโคจรรอบโลก ทางด้านสหรัฐอเมริกา ก็ได้ก่อตั้งหน่วยงานอาร์พาขึ้น เพื่อพยายามผลักดันให้เทคโนโลยีของสหรัฐอเมริกาเหนือกว่าโซเวียต กระทรวงกลาโหมของสหรัฐฯ ได้วางโครงการสร้างเครือข่ายคอมพิวเตอร์ที่คงความสามารถในการติดต่อสื่อสารถึงกันอยู่ สาเหตุดังกล่าวนี้เป็นจุดกำเนิดและที่มาของอาร์พาเน็ต

ในปี พ.ศ. 2515 ได้มีการปรับปรุงหน่วยงานอาร์พาและเรียกชื่อใหม่ว่า คาร์พา (Defense Advanced Research Project Agency) และต่อมาเมื่อ พ.ศ. 2518 คาร์พาได้โอนหน้าที่ดูแลรับผิดชอบอาร์พาเน็ตโดยตรงให้แก่ หน่วยงานสื่อสารของกองทัพ หรือ ดีซีเอ (ปัจจุบันคือ DISA) เนื่องจากอาร์พาเน็ตได้แปรสภาพจากเครือข่ายเชิงทดลองเป็นเครือข่ายที่ปฏิบัติงานได้อย่างแท้จริงแล้ว

ในปลายปี พ.ศ. 2526 อาร์พาเน็ตถูกแบ่งแยกออกเป็นสองเครือข่าย คือ เครือข่ายด้านการวิจัยและเครือข่ายของกองทัพ เครือข่ายด้านงานวิจัยยังคงใช้ชื่อ อาร์พาเน็ตอยู่เช่นเดิม ส่วนเครือข่ายของกองทัพมีชื่อเรียกใหม่ว่า มิลเน็ต (MILNET) และทั้งสองเครือข่ายยังคงได้รับทุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนับสนุนจากกองทัพอยู่ และในช่วงต่อมามีหน่วยงานซึ่งมีเครือข่ายที่ใช้โปรโตคอลทีซีพี/ไอพี เพื่อเชื่อมต่อเข้ามา และได้มีการปรับเปลี่ยนชื่อเรียกเครือข่ายไปเรื่อย ๆ จนกลายเป็น อินเทอร์เน็ต (Internet)

2.3 เวิลด์ไวด์เว็บ (WWW : World – Wide – Web)

เวิลด์ไวด์เว็บ ได้เกิดจากในปี 1980 มร.เบอร์นเนอร์ได้แสดงความคิดว่าโลกของอนาคต การติดต่อสื่อสารควรเป็นแบบที่สามารถเชื่อมโยงถึงกันหมด ซึ่งจะมีประโยชน์ในการแลกเปลี่ยนความคิดเห็น ค้นหาข้อมูล และการติดต่อ ทำให้เกิดการพัฒนาต่าง ๆ โดยใช้ระบบเครือข่ายที่มีประสิทธิภาพสูงซึ่งการเป็นจุดเริ่มต้นของเวิลด์ไวด์เว็บ ซึ่งใช้ไฮเปอร์เท็กซ์(HyperText) ที่ห้องปฏิบัติการวิจัยฟิสิกส์ มหาวิทยาลัยเจนีวา (CERN) ประเทศสวิตเซอร์แลนด์ และพัฒนาต่อมาเป็นการเชื่อมโยงลักษณะไฮเปอร์เท็กซ์บนอินเทอร์เน็ต โดยใช้มาตรฐาน HTML (HyperText Markup Language) และการเชื่อมโยงด้วย Protocol : http เพื่อเป็นมาตรฐานเดียวกัน การนำเสนอข่าวสารในรูปแบบที่เรียกว่าไฮเปอร์เท็กซ์ ซึ่งมีการผนวกเอาตัวอักษร, รูปภาพ, เสียง, ภาพเคลื่อนไหวที่เรียกว่ามัลติมีเดียที่เรากำหนดเป็นตัวเชื่อมโยงข้อมูลในส่วนอื่น ๆ โดยเราเรียกการนำเสนอข้อมูลชนิดนี้ว่า “เว็บเพจ” ซึ่งการที่เราจะเข้าไปอ่านข้อมูลประเภทนี้ได้จะต้องอาศัยโปรแกรมอ่านข้อมูลประเภทนี้ขึ้นมา ซึ่งเรียกมันว่า เบราเซอร์ (Browser)

เราสามารถแบ่งเบราเซอร์ออกได้เป็น 2 ประเภท คือ เบราเซอร์ที่แสดงผลเป็นเท็กซ์โหมด (Text Mode) และ กราฟฟิกโหมด (Graphic Mode) โดยเบราเซอร์เท็กซ์โหมดจะไม่สามารถแสดงผลที่เป็นรูปภาพได้ ตัวอย่างเบราเซอร์ประเภทนี้ คือ lynx ส่วนอีกประเภทหนึ่ง คือ กราฟฟิกโหมดจะสามารถแสดงผลได้ครบ คือสามารถแสดงได้ทั้งรูปภาพและตัวอักษร ตัวอย่างเบราเซอร์ประเภทนี้คือ Internet Explorer , Netscape Communicator เป็นต้น

บริการประเภทนี้จำนวนมาก โดยคอมพิวเตอร์แต่ละเครื่องสามารถเชื่อมโยงข้อมูลเข้าด้วยกันได้ทั้งหมดซึ่งมีลักษณะคล้ายใยแมงมุม คือ เครื่องคอมพิวเตอร์มีเส้นทางที่สามารถเชื่อมต่อถึงกันได้เป็นจำนวนมาก เหมือนใยแมงมุมที่สามารถเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งได้คว้นเส้นทางมากมาย เราจึงเรียกบริการนี้ว่า “เวิลด์ไวด์เว็บ”

2.4 เบราเซอร์ (Browser)

เบราเซอร์เป็นโปรแกรมที่ทำให้สามารถอ่านภาษา HTML ที่เขียนอยู่บนเว็บไซต์ได้โดยมีลักษณะการทำงานเป็นโปรแกรมที่ทำหน้าที่ดึงข้อมูลจากเวิลด์ไวด์เว็บ โดยเบราเซอร์ตัวแรกของโลกคือ Mosaic ที่ได้รับการพัฒนาขึ้นใน 5 ปีก่อนให้สามารถดูกราฟฟิกต่าง ๆ ได้ จาก Marc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Andersen ใน National Center of Supercomputing Application (NCSA) รัฐอิลลินอยส์ สหรัฐอเมริกา แรกเริ่ม Mosaic เป็นเบราว์เซอร์ที่แจกฟรีแก่ผู้ที่สนใจทั่วไป และต่อมาจึงได้พัฒนาเป็น Netscape Communicator และ Internet Explorer ซึ่งเป็นเบราว์เซอร์ที่มุ่งหวังทางการตลาด

ในการสื่อสารระหว่างคอมพิวเตอร์กับอินเทอร์เน็ตนั้น การทำงานของเบราว์เซอร์ที่ติดตั้งอยู่ในคอมพิวเตอร์ของผู้ใช้งานจะส่งคำร้องขอไฟล์ HTML ที่เราต้องการไปยังเซิร์ฟเวอร์ ที่อยู่บนอินเทอร์เน็ตตามที่อยู่หรือ URL ของแต่ละเซิร์ฟเวอร์ และเมื่อข้อมูลกลับมายังเครื่องคอมพิวเตอร์ของผู้ใช้งานเบราว์เซอร์ก็จะแปลงข้อมูล HTML เป็นรูปแบบข้อความพร้อมกราฟฟิกต่าง ๆ ในหน้าเว็บไซต์นั้น ๆ

2.5 โพรโทคอล (Protocol)

โพรโทคอลเป็นโครงข่ายของเว็บที่ได้มีการนำมาใช้ในการสื่อสารบนอินเทอร์เน็ต โดยได้มีการเริ่มพัฒนาโดย Tim Berners Lee เป็นนักวิทยาศาสตร์ทางนิวเคลียร์ที่สถาบัน CERN (“Conseil Europeen pour la Recherche Nucleaire,” the European Council for Nuclear Research) ในยุโรป เขาทำงานคู่ไปกับงานทางด้านนิวเคลียร์จนในที่สุดเขาค้นพบ Marc Andersen ซึ่งกลายมาเป็น frame ของ Netscape ในปัจจุบัน โพรโทคอลได้รับการพัฒนามาเรื่อยๆ ปัจจุบันซึ่งเป็นประโยชน์ต่อประชากรของโลกเป็นอันมาก

โพรโทคอลของอินเทอร์เน็ต เป็นการพัฒนาวิธีการที่ทำให้เครื่องคอมพิวเตอร์หลาย ๆ เครื่องสื่อสารกันบนเครือข่ายอินเทอร์เน็ต การสื่อสารกันระหว่างเครื่องคอมพิวเตอร์จะดำเนินการในรูปแบบของข่าวสารต่าง ๆ ที่ส่งไปและส่งกลับ

2.6 เลขประจำเครื่อง (IP Address)

เครื่องคอมพิวเตอร์ในอินเทอร์เน็ตจะมีรูปแบบของเลขที่อยู่ประจำเครื่องคอมพิวเตอร์ในเครือข่ายอินเทอร์เน็ตประจำตัวมีขนาด 32 บิต มีรูปแบบที่ประกอบด้วยเลขจำนวนเต็ม 4 กลุ่ม ๆ ละ 8 บิต แต่ละกลุ่มมีของเขตตั้งแต่ 0 ถึง 255 และจะเขียนเรียงกันไปเรื่อย ๆ โดยมีเครื่องหมายจุดมาคั่นระหว่างชุดตัวเลข เช่น 192.168.1.99 เป็นต้น เลขประจำเครื่องสามารถแบ่งได้ 2 ส่วน ได้แก่ ส่วนที่เป็นตัวเลขของเครือข่ายจากตัวอย่าง คือ 192.168 และส่วนของหมายเลขสมาชิกจากตัวอย่างได้แก่ 1.99

2.7 พอร์ตบนอินเทอร์เน็ต (Port)

หมายเลขพอร์ต เป็นส่วนหนึ่งของโปรโตคอลใด ๆ ที่มีคุณลักษณะ RFC (Requests For Comment) หมายเลขต่าง ๆ ของพอร์ตได้ตอบสนองการให้บริการของอินเทอร์เน็ตเป็นเวลามากกว่า 20 ปี การให้บริการแต่ละอย่างจะขึ้นอยู่กับโปรโตคอล หมายเลขพอร์ตจะจำแนกโดยจำนวนเต็มระหว่าง 0 ถึง 65535 หมายเลขพอร์ตต่าง ๆ นี้จะช่วยในการเข้าถึงบริการต่าง ๆ ของคอมพิวเตอร์ใด ๆ บนอินเทอร์เน็ต เช่น 192.168.1.99:4444 จากตัวอย่างเป็นการเรียกใช้บริการผ่านทางพอร์ต 4444

2.8 การเชื่อมโยงกับซอกเก็ต (Socket) ของจาวา

ซอกเก็ต เป็นโปรแกรมอรรถประโยชน์ (Utility) ของเครือข่ายโดยเฉพาะ ซึ่งจะเกี่ยวข้องกับโปรแกรมของเครือข่ายระบบ Unix ซอกเก็ตต่าง ๆ ถือเป็นกระบวนการของเครือข่ายซึ่งเชื่อมโยงกับสองโฮสต์ (Host) และสร้างสองเส้นทางเพื่อขนส่งข้อมูลระหว่างไคลเอ็นท์และเซิร์ฟเวอร์ ซอกเก็ตต่าง ๆ จะสื่อสารบนพอร์ตที่โปรแกรมระบุ โดยซอกเก็ตต่าง ๆ มีประโยชน์มากในภาษาจาวาสำหรับการสนับสนุนสตรีม (Stream) ของการสื่อสารระหว่างโปรแกรม โดยจาวามีคลาสซอกเก็ตเพื่อช่วยสนับสนุนการสร้างส่วนที่เกี่ยวข้องกับพอร์ตบนคอมพิวเตอร์ที่อยู่ระยะไกลและคลาสเซิร์ฟเวอร์ซอกเก็ตสำหรับสร้างพอร์ตเพื่อให้คอมพิวเตอร์อื่นสามารถเชื่อมต่อได้ ซอกเก็ตต่าง ๆ เป็นวิธีการเริ่มแรกของการสื่อสารบนอินเทอร์เน็ต เว็บเพจ , FTP (File Transfer Protocol) , E-Mail ล้วนแล้วจำเป็นต้องใช้ซอกเก็ต

2.9 บทสรุป

ในบทนี้ได้กล่าวถึงความเป็นมาของอินเทอร์เน็ตซึ่งแสดงถึงพัฒนาการของอินเทอร์เน็ตซึ่งเริ่มต้นการจากพัฒนาเพื่อสนับสนุนงานวิจัยด้านการทหารแล้วได้มีการปรับปรุงเปลี่ยนแปลงมาเป็นอินเทอร์เน็ตที่เราใช้กันอยู่ในปัจจุบัน โดยอินเทอร์เน็ตได้ก้าวเข้ามามีบทบาทเป็นอย่างมากในปัจจุบันซึ่งเกี่ยวข้องกับงานหลาย ๆ ด้านและจำนวนผู้ใช้งานอินเทอร์เน็ตมีจำนวนเพิ่มมากขึ้นทำให้จำเป็นต้องเรียนรู้ถึงการพัฒนาและประยุกต์เพื่อใช้เครือข่ายร่วมกับอินเทอร์เน็ตให้เกิดประโยชน์มากที่สุด

บทที่ 3

ตัวควบคุมแบบป้อนกลับ

3.1 บทนำ

มนุษย์ได้นำเอาระบบควบคุมมาใช้งานก็เพื่อเพิ่มความสามารถในการทำงานหรือลดข้อบกพร่องของมนุษย์ทางกายภาพ เช่น การปฏิบัติงานในกิจวัตรประจำวันที่ย้ำ ๆ มีความซับซ้อนหรือช่วยประหยัดค่าใช้จ่าย เป็นต้น โดยตัวควบคุมป้อนกลับถือเป็นหัวใจสำคัญของระบบควบคุมเพื่อให้ได้ผลการควบคุมเป็นไปตามเป้าหมายที่ได้กำหนดไว้ ซึ่งการทำงานของระบบควบคุมป้อนกลับถือว่าการควบคุมแบบระบบปิด คือ อินพุตถูกปรับเปลี่ยนด้วยเอาต์พุตที่ได้ก่อนที่จะป้อนกลับเข้าสู่ระบบควบคุม เนื่องจากเอาต์พุตถูกป้อนกลับมาผ่านทางอุปกรณ์ส่วนป้อนกลับ (Feedback element) เพื่อหักล้างกับอินพุตโดยทั่วไปจึงเรียกระบบวงจรปิดว่าระบบควบคุมป้อนกลับแบบลบ (negative feedback system) หรือเรียกง่าย ๆ ว่าระบบควบคุมป้อนกลับ

3.2 ประวัติและความเป็นมาของตัวควบคุมแบบป้อนกลับ

การควบคุมป้อนกลับแบบอัตโนมัติมักถูกเข้าใจว่าเป็นวิทยาการสมัยใหม่ แต่ในความเป็นจริงแล้วระบบควบคุมแบบป้อนกลับถูกคิดค้นขึ้นมาโดยชาวกรีกเมื่อ 250 ปีก่อนคริสตกาล เพื่อนำมาใช้ในการควบคุมระดับน้ำอย่างประสิทธิภาพ โดยมีหลักการทำงานเช่นเดียวกับตัวควบคุมระดับน้ำ (Level Regulator) ในสุภกณฑ์ที่ใช้ในปัจจุบัน ต่อมาในปี ค.ศ. 1788 James Watt ได้ประดิษฐ์เครื่องควบคุมลูกลอย (Fly – Ball governor) เป็นครั้งแรกเพื่อใช้ในเครื่องจักรไอน้ำ ซึ่งเป็นพื้นฐานในการพัฒนาระบบพลังไอน้ำในเวลาต่อมา ในทศวรรษที่ 30 (ค.ศ. 1930 - 1939) การควบคุมแบบป้อนกลับมีบทบาทสำคัญในการพัฒนาตัวขยายเชิงดำเนินการ (Operational Amplifier) ที่มีอัตราขยายสูง ซึ่งต่อมาได้ถูกนำมาใช้งานอย่างกว้างขวางในอุปกรณ์อิเล็กทรอนิกส์

ตลอดทศวรรษที่ 30 นี้ตัวควบคุมพีไอซีเริ่มมีวางจำหน่ายในท้องตลาด และบทความทางทฤษฎีเกี่ยวกับการควบคุมระบบได้ถูกตีพิมพ์เป็นครั้งแรกในช่วงทศวรรษนี้ ในช่วงทศวรรษที่ 40 (ค.ศ. 1940 - 1949) ตัวควบคุมพีไอซีแบบนิวแมติกส์ก็เป็นที่ยอมรับในงานอุตสาหกรรมอย่างแพร่หลายและคู่มือวงจรอิเล็กทรอนิกส์ของตัวควบคุมชนิดนี้ได้เข้าสู่ท้องตลาดในช่วงทศวรรษที่ 50 (ค.ศ. 1950 - 1959) ในปลายทศวรรษนี้คอมพิวเตอร์ควบคุมกระบวนการในงานอุตสาหกรรมได้ถูกนำมาใช้เป็นครั้งแรก และตลอด 20 ปีที่ผ่านมาฮาร์ดแวร์ของดิจิทัลคอมพิวเตอร์ก็ได้ถูกนำมาใช้เป็นอุปกรณ์สำคัญในการทำงานของตัวควบคุมกระบวนการด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ตัวควบคุมแบบเปิด - ปิด (On - Off Controller)

ตัวควบคุมแบบเปิด - ปิดเป็นตัวควบคุมที่มีรูปแบบการทำงานที่ง่ายที่สุดและมีราคาถูก นิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูงนัก และผลของความคลาดเคลื่อนไม่มีผลต่อการควบคุม เช่น การควบคุมอุณหภูมิของตู้เย็นที่ใช้ในบ้านหรือใช้ในงานอุตสาหกรรมที่มีการทำงานไม่ซับซ้อน การควบคุมระดับบางประเภทและใช้ในการควบคุมระบบความร้อน แต่ข้อเสียคือการควบคุมโดยใช้ตัวควบคุมชนิดนี้อาจผลต่อความต่อเนื่องของผลตอบสนองของกระบวนการได้

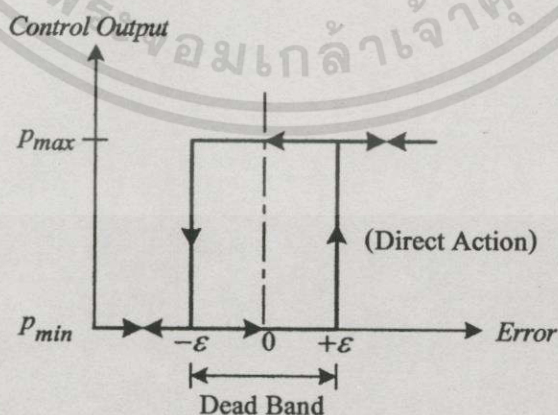
การควบคุมแบบเปิด - ปิดในทางทฤษฎี เอาท์พุทของตัวควบคุมจะเป็นไปได้เพียง 2 ค่าคือ

$$p(t) = \begin{cases} p_{\max} & \text{เมื่อ } e \geq 0 \\ p_{\min} & \text{เมื่อ } e < 0 \end{cases} \quad (3-1)$$

เมื่อ e คือสัญญาณผิดพลาด (Error)

p_{\max} และ p_{\min} คือค่าในการควบคุมสูงสุดและต่ำสุดตามลำดับ เช่น ในตัวควบคุมนิวแมติกส์ จะกำหนดให้ $p_{\max} = 15 \text{ psi}$ และ $p_{\min} = 3 \text{ psi}$ หรือ $p_{\max} = 20 \text{ mA}$ และ $p_{\min} = 4 \text{ mA}$ ในอุปกรณ์อิเล็กทรอนิกส์

จากสมการที่ (3-1) จะพบว่า การควบคุมแบบเปิด - ปิดเป็นการควบคุมแบบ 2 ตำแหน่งหรือแบบแองกอนโทรล (Bang - Bang Control) หรืออาจพิจารณาว่าเป็นการควบคุมแบบ P ชนิดพิเศษที่มีอัตราขยายสูงมากก็ได้

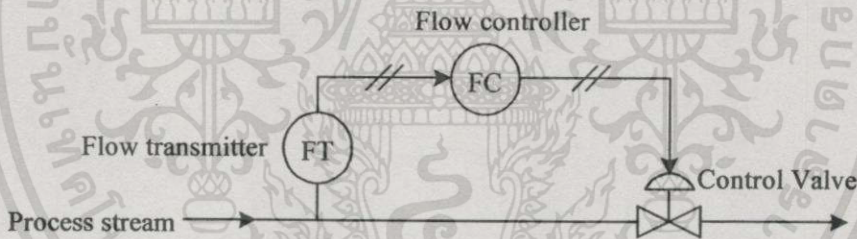


รูปที่ 3.1 การควบคุมแบบเปิด - ปิด

ตัวควบคุมแบบเปิด - ปิดสามารถปรับปรุงให้รวมค่าเดดแบนด์ (Dead band) เป็นสัญญาณผิดพลาดได้ เพื่อลดความไวต่อสัญญาณรบกวนในเครื่องมือวัดหรือเพื่อป้องกันไม่ให้ตัวควบคุมทำงานบ่อยครั้งจนเกินไป จากรูปที่ 3.1 หากค่าความผิดพลาดเพิ่มขึ้นเกินค่าวิกฤต (Critical Value) หรือ $+E$ ค่าเอาต์พุตของตัวควบคุมจึงจะเปลี่ยนจาก p_{min} เป็น p_{max} เพื่อให้ค่าความคลาดเคลื่อนลดต่ำกว่าค่าวิกฤตหรือ $-E$ และเมื่อถึงจุดนี้ค่าเอาต์พุตจะเปลี่ยนจาก p_{max} ไปเป็น p_{min}

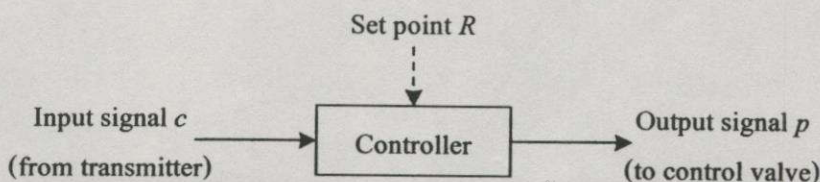
3.4 ตัวควบคุมแบบ PID (PID Controllers)

การควบคุมป้อนกลับมีพื้นฐานอยู่ 3 แบบที่ใช้งานกันทั่วไปคือ การควบคุมแบบ Proportional (P), การควบคุมแบบ Integral (I) และการควบคุมแบบ Derivative (D) จากรูปที่ 3.2 เมื่อวัดและแปลงอัตราการไหลของไอน้ำเป็นสัญญาณนิวแมติกส์ให้กับตัวควบคุมการไหลแล้ว ตัวควบคุมจะเปรียบเทียบกับค่าที่วัดได้กับค่าเป้าหมาย (Set point) หรือ SP แล้วจึงส่งสัญญาณควบคุมที่เหมาะสมให้แก่ตัวควบคุม สัญญาณนิวแมติกส์ใช้สัญลักษณ์พื้นฐานคือเส้นขนานตัดเส้นตรง (//)



รูปที่ 3.2 ระบบควบคุมการไหล

ในรูปที่ 3.3 แสดงบล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ ค่าเป้าหมายแสดงโดยใช้เส้นประซึ่งถูกกำหนดโดยการป้อนค่าหรือการกำหนดตำแหน่งที่ตัวควบคุมซึ่งเรียกว่า Local set point ตัวควบคุมบางตัวสามารถตั้งค่าเป้าหมายระยะไกลหรือ Remote set point ได้ โดยยินยอมให้ตัวควบคุมรับสัญญาณค่าเป้าหมายจากอุปกรณ์ภายนอกเช่นตัวควบคุมอีกตัวหนึ่งหรือเครื่องดิจิทัลคอมพิวเตอร์ อินพุตและเอาต์พุตของตัวควบคุมเป็นสัญญาณแบบต่อเนื่องซึ่งเป็นที่สัญญาณนิวแมติกส์และสัญญาณไฟฟ้า



รูปที่ 3.3 บล็อก ไดอะแกรมของตัวควบคุมแบบป้อนกลับ

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การควบคุมแบบป้อนกลับ

3.5.1 การควบคุมแบบ Proportional (P)

วัตถุประสงค์ของการควบคุมแบบป้อนกลับคือการลดสัญญาณผิดพลาด (Error) หรือ $e(t)$ ให้มีค่าเท่ากับศูนย์ ดังสมการ

$$e(t) = R(t) - c(t) \quad (3-2)$$

เมื่อ $R(t)$ = ค่าเป้าหมาย

$c(t)$ = ผลตอบสนองของกระบวนการหรือค่าสัญญาณจากทรานสมิตเตอร์ (Transmitter)

จากสมการที่ 3-2 จะพบว่าค่าเป้าหมายสามารถเปลี่ยนแปลงตามเวลาได้ การควบคุมกระบวนการแบบ P เอาท์พุทของตัวควบคุมจะแปรผันตามสัญญาณผิดพลาดที่เกิดขึ้น ดังสมการ

$$p(t) = \bar{p} + K_p e(t) \quad (3-3)$$

เมื่อ $p(t)$ = ค่าเอาท์พุทของตัวควบคุม

\bar{p} = ค่า Bias

K_p = อัตราขยายของตัวควบคุม โดยปกติจะ ไม่มีหน่วย

เงื่อนไขของการควบคุมแบบ P

อัตราขยายของตัวควบคุม: K_p สามารถปรับค่าได้ เพื่อให้เอาท์พุทของตัวควบคุมสามารถเปลี่ยนแปลงค่าได้ไวตามความต้องการและทำให้ผลตอบสนองของกระบวนการเข้าสู่ค่าเป้าหมาย การเลือกค่า K_p สามารถทำให้เอาท์พุทของตัวควบคุมมีค่าเพิ่มขึ้นหรือลดลงได้

ค่า \bar{p} สามารถปรับค่าได้โดยที่เอาท์พุทของตัวควบคุมจะมีค่าเท่ากับ \bar{p} ก็ต่อเมื่อค่าผิดพลาดเท่ากับศูนย์ ซึ่งในสภาวะนี้อาจกล่าวได้ว่าค่าเอาท์พุทของตัวควบคุมรวมไปถึงผลตอบสนองของกระบวนการอยู่ในสภาวะคงที่ (Steady - state) สำหรับตัวควบคุมที่ใช้งานกันโดยทั่วไปค่า K_p จะไม่มีหน่วยเพราะค่า $p(t)$ และ $e(t)$ มีหน่วยเดียวกัน เช่นในกรณีของอุปกรณ์ทางไฟฟ้าหรือทางนิวแมติกส์ หรืออาจอยู่ในรูปของตัวเลข 0 - 100% เพื่อความสะดวกในการทำงานของตัวแสดงกราฟฟิกและซอฟต์แวร์ควบคุมคอมพิวเตอร์ ในอีกกรณีหนึ่งค่าสัญญาณผิดพลาดจะอยู่ในรูปของหน่วยทางวิศวกรรมเช่นหน่วยอุณหภูมิเป็นองศาเซลเซียส ($^{\circ}\text{C}$) หรือหน่วยของความหนาแน่นเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mol/L ซึ่งในกรณีนี้ K_p และอัตราขยายที่สภาวะคงที่ (Steady – state gain) ของอุปกรณ์ในวงรอบควบคุมตัวอื่นๆ เช่น ทรานส์มิเตอร์หรือวาล์วควบคุมก็จะมีหน่วยด้วยเช่นกัน

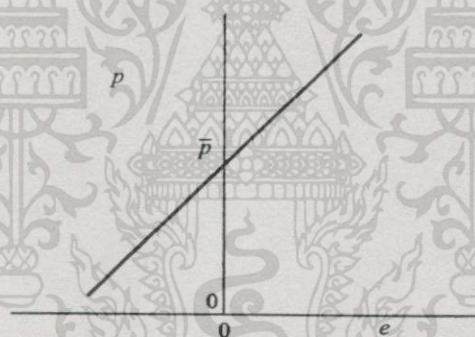
ตัวควบคุมแบบคั้งเดิมบางตัวจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า K_p ซึ่ง PB จะมีหน่วยเป็น % และจะใช้เฉพาะในกรณีที่ค่า K_p ไม่มีหน่วยเท่านั้น

$$PB = \frac{100\%}{K_p} \quad (3-4)$$

ตัวควบคุมแบบ P ในทางอุดมคติจะไม่พิจารณาถึงข้อจำกัดทางกายภาพของเอาต์พุตตัวควบคุม ซึ่งปกติแล้วตัวควบคุมจะเกิดการอิ่มตัวเมื่อถึงขอบเขตทางกายภาพคือ p_{max} และ p_{min}

ในการพิจารณาฟังก์ชันถ่ายโอนของตัวควบคุมแบบ P จะกำหนดให้

$$p'(t) = p(t) - \bar{p} \quad (3-5)$$



รูปที่ 3.4 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ K_p)

ดังนั้น

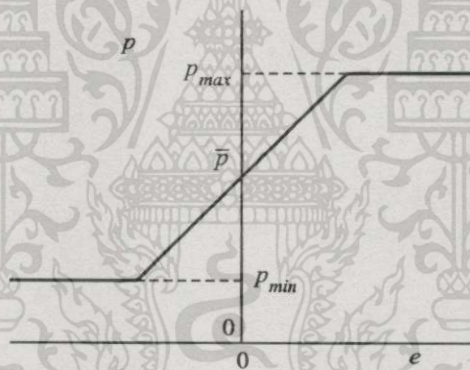
$$p'(t) = K_p e(t) \quad (3-6)$$

สมการลาปลาซซึ่งเป็นฟังก์ชันถ่ายโอนของตัวควบคุมคือ

$$\frac{P'(s)}{E(s)} = K_p \quad (3-7)$$

ข้อเสียซึ่งเป็นลักษณะเฉพาะของการควบคุมแบบ P คือไม่สามารถกำจัด Steady - state error ที่เกิดขึ้นหลังจากการเปลี่ยนแปลงค่าเป้าหมายหรือการได้รับสัญญาณรบกวนจากโหลด ซึ่งจะเกิดขึ้นกับการควบคุมแบบ P ในทุกๆค่า K_p ในทางทฤษฎีค่าออฟเซตสามารถกำจัดได้โดยการตั้งค่าเป้าหมายหรือค่า \bar{p} ขึ้นใหม่หลังจากเกิดออฟเซตแต่อย่างไรก็ตามวิธีการนี้ไม่ได้เป็นวิธีการที่ดีที่สุด เนื่องจากต้องอาศัยผู้มีความชำนาญเพียงพอและการหาค่าเป้าหมายหรือค่า \bar{p} ใหม่ก็ต้องใช้การลองผิดลองถูกซึ่งอาจเกิดความผิดพลาดขึ้นได้ การแก้ไขปัญหานี้ในทางปฏิบัติจะใช้การทำงานของ การควบคุมแบบ Integral (I) เข้ามาช่วยซึ่งจะให้ผลตอบสนองที่ดีขึ้น เนื่องจากการควบคุมแบบ I จะมีการเปลี่ยนแปลงค่าการควบคุมเพื่อลดออฟเซตได้โดยอัตโนมัติ

ในการควบคุมกระบวนการบางอย่างที่ไม่ต้องคำนึงถึงการเกิดออฟเซต การควบคุมแบบ P ก็มีความเหมาะสมในการใช้งาน เนื่องจากมีรูปแบบการทำงานที่ง่ายและไม่ซับซ้อน เช่นการควบคุมระดับของเหลวที่ไม่จำเป็นต้องรักษาระดับให้อยู่ในค่าเป้าหมายแต่ต้องการเพียงควบคุมไม่ให้ของเหลวล้นถึงหรือหมดถึงเท่านั้น เป็นต้น



รูปที่ 3.5 การควบคุมแบบ P ในทางปฏิบัติ

3.5.2 การควบคุมแบบ Integral (I) และการควบคุมแบบ PI

การควบคุมแบบ I หมายถึงการควบคุมแบบรีเซตหรือแบบลอย (Floating) โดยที่ค่าเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอินทิเกรตสัญญาณผิดพลาด ดังสมการ

$$p(t) = \bar{p} + \frac{1}{T_i} \int_0^t e(t) dt \tag{3-8}$$

เมื่อ T_i คือเวลาอินทิเกรตหรือเวลารีเซตซึ่งมีหน่วยเป็นหน่วยของเวลาและสามารถปรับค่าได้ตามความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบ I เป็นที่นิยมใช้กันอย่างกว้างขวางเพราะสามารถใช้งานได้ดีและสามารถกำจัดออฟเซตได้อย่างมีประสิทธิภาพ จากสมการที่ (3-8) เมื่อกระบวนการเข้าสู่สภาวะคงที่สัญญาณผิดพลาดและเอาต์พุตของตัวควบคุมจะมีค่าคงที่ หรืออาจกล่าวได้ว่าค่า $p(t)$ จะเปลี่ยนแปลงตามเวลาจนกระทั่ง $e(t) = 0$ ดังนั้นเมื่อใช้การควบคุมแบบ I ค่า $p(t)$ จะให้ค่าในการควบคุมซึ่งจะทำให้สัญญาณผิดพลาดในสภาวะคงที่มีค่าเท่ากับศูนย์ ซึ่งสัญญาณผิดพลาดนี้จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายหรือได้รับสัญญาณรบกวนจากโหลดและการกำจัดสัญญาณผิดพลาดให้เท่ากับศูนย์จะกระทำได้อีกต่อเมื่อเอาต์พุตของตัวควบคุมหรืออุปกรณ์ควบคุมตัวสุดท้ายยังไม่ถึงขอบเขตการอิมิตัวแต่หากเกินช่วงขอบเขตนี้ไปแล้วค่าของกระบวนการจะไม่สามารถกลับสู่ค่าเป้าหมายได้ ซึ่งการอิมิตัวเกิดจากสัญญาณรบกวนหรือการเปลี่ยนแปลงค่าเป้าหมายเกินช่วงของค่า $p(t)$

วัตถุประสงค์หลักของการควบคุมแบบ I คือการกำจัดค่าออฟเซต (Offset) แต่ตัวควบคุมแบบ I ไม่นิยมใช้งานเพียงตัวเดียวเพราะจะให้ผลในการควบคุมน้อยมากจนกว่าจะเกิดสัญญาณผิดพลาดขึ้นมาอย่างต่อเนื่องในบางเวลาเท่านั้น ดังนั้นจึงมักนำการควบคุมแบบ I มาทำงานร่วมกับการควบคุมแบบ P หรือที่เรียกว่าตัวควบคุมแบบ PI เพราะการควบคุมแบบ P จะให้ผลในการควบคุมแบบทันทีทันใดในขณะที่ตรวจพบสัญญาณ ผิดพลาด

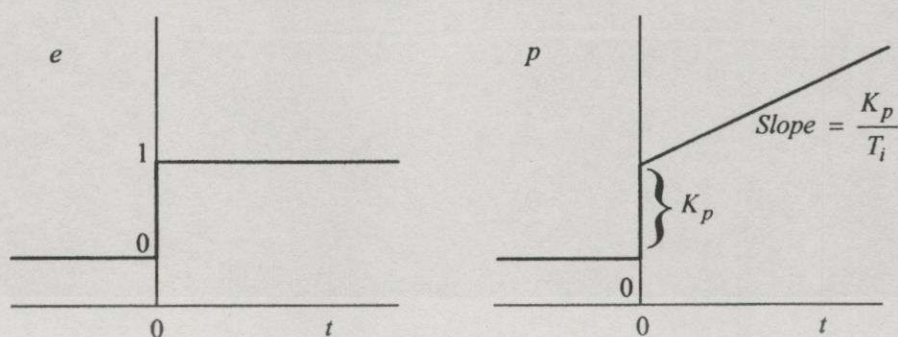
สมการของตัวควบคุมแบบ PI คือ

$$p(t) = \bar{p} + K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right] \quad (3-9)$$

ฟังก์ชันถ่ายโอนของตัวควบคุมแบบ PI ในรูปของสมการลาปลาซ คือ

$$\frac{P'(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) = K_p \left(\frac{T_i s + 1}{T_i s} \right) \quad (3-10)$$

ในรูปที่ 3.6 แสดงผลตอบสนองของตัวควบคุม PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$ ที่เวลา $t = 0$ เอาต์พุตของตัวควบคุมจะเปลี่ยนแปลงพร้อมกับค่าผิดพลาด โดยเกิดจากการควบคุมแบบ P การควบคุมแบบ I จะทำให้ $p(t)$ เพิ่มขึ้นในลักษณะลาดชัน (Ramp) เมื่อ $t > 0$ ที่ $t = T_i$ เทอม I จึงให้ค่าในการควบคุมเท่ากับเทอม P



รูปที่ 3.6 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$

ตัวควบคุมบางตัวจะใช้เทอม $\frac{1}{T_i}$ (ครั้ง / นาที) แทนการใช้ T_i (นาที) เช่นถ้า $T_i = 0.2$ นาที แล้ว $\frac{1}{T_i} = 5$ ครั้ง / นาที เป็นต้น

ข้อเสียของการควบคุมแบบ I คือระบบจะมีการตอบสนองแบบแกว่งทำให้ระบบขาดเสถียรภาพ แต่ในบางครั้งก็สามารถยอมรับได้หากการแกว่งเกิดขึ้นพร้อมกับการตอบสนองของกระบวนการที่รวดเร็ว แต่ข้อเสียของการควบคุมแบบ I ก็สามารถหลีกเลี่ยงได้โดยการปรับใช้ค่าพารามิเตอร์ให้เหมาะสมหรือเพิ่มการควบคุมแบบ D ซึ่งจะสามารถลดผลตอบสนองที่ไม่มีเสถียรภาพลงได้

3.5.3 การควบคุมแบบ Derivative (D) การควบคุมแบบ PD และการควบคุมแบบ PID

การควบคุมแบบ D หมายถึงการควบคุมแบบอัตราส่วน (Rate action) หรือการทำนายล่วงหน้า (Pre - act) โดยจะทำนายพฤติกรรมของสัญญาณผิดพลาดที่จะเกิดขึ้นในอนาคต ซึ่งจะพิจารณาจากอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด เช่นในกรณีที่เราปฏิกรณ์มีอุณหภูมิเพิ่มขึ้น 10°C ในเวลา 3 นาที ซึ่งสามารถสรุปได้ว่าเกิดขึ้นเร็วกว่าในกรณีที่มีอุณหภูมิเพิ่มขึ้น 10°C ในเวลา 30 นาที และสามารถประเมินประสิทธิภาพในการแก้ไขสภาวะของการเกิดปฏิกิริยาที่ทำให้อุณหภูมิเพิ่มสูงขึ้นได้ ถ้าการควบคุมเตาปฏิกรณ์ใช้ระบบปรับด้วยมือ (Manual) ผู้ควบคุมที่มีประสบการณ์สามารถประเมินผลตอบสนองที่จะเกิดขึ้นและแก้ไขสภาวะการณ์ได้อย่างเหมาะสมในเวลาอันรวดเร็วเพื่อทำการลดอุณหภูมิลง แต่การควบคุมด้วยระบบอัตโนมัติตัวควบคุมแบบ P จะตอบสนองต่อการเปลี่ยนแปลงของอุณหภูมิเท่านั้น แต่ไม่สามารถแยกแยะความแตกต่างของช่วงเวลาที่เกิดการเปลี่ยนแปลงได้ เช่นเดียวกับตัวควบคุมแบบ I ซึ่งโดยปกติแล้วจะสร้างสัญญาณควบคุมที่แปรผันตามระยะเวลาของสัญญาณรบกวนที่เกิดขึ้น ดังนั้นสัญญาณรบกวนขนาดเล็กที่เกิดขึ้นอย่างซ้าๆก็สามารถทำให้ตัวควบคุมสร้างสัญญาณควบคุมที่มีค่ามากขึ้นมาได้

การทำงานของตัวควบคุมแบบ D ในทางอุดมคติมีสมการดังนี้

$$p(t) = \bar{p} + T_d \frac{de}{dt} \quad (3-11)$$

เมื่อ T_d คือเวลา Derivative มีหน่วยเป็นหน่วยของเวลา

เอาที่พุดของตัวควบคุมมีค่าเท่ากับ \bar{p} เมื่อสัญญาณผิดพลาด มีค่าคงที่หรือเมื่อ $\frac{de}{dt} = 0$ การควบคุมแบบ D จะไม่ใช้งานเพียงตัวเดียวแต่จะใช้งานร่วมกับการควบคุมแบบ P หรือ PI

การควบคุมแบบ PD มีฟังก์ชันถ่ายโอนคือ

$$\frac{P'(s)}{E(s)} = K_p(1 + T_d s) \quad (3-12)$$

การควบคุมแบบ D ทำให้กระบวนการมีเสถียรภาพและลดแนวโน้มความไม่มีเสถียรภาพของการควบคุมแบบ I นอกจากนี้ยังนำมาใช้เพื่อปรับปรุงผลตอบสนองทางพลศาสตร์โดยลดเวลาเข้าสู่สภาวะคงที่ (Settling time) ของกระบวนการลง ซึ่งทำให้กระบวนการเข้าสู่สภาวะคงที่เร็วขึ้น แต่ในกรณีที่เครื่องมือวัดในกระบวนการมีสัญญาณรบกวนความถี่สูงและมีการแกว่งไม่มีแบบแผน การควบคุมแบบ D จะทำให้ผลตอบสนองของกระบวนการเปลี่ยนแปลงในช่วงกว้างและสัญญาณรบกวนจะถูกขยายแต่สามารถป้องกันได้โดยการกรองความถี่ โดยปกติแล้วการควบคุมแบบ D จะไม่นิยมใช้กับการควบคุมการไหลเนื่องจากวงรอบการควบคุมมีการตอบสนองที่รวดเร็วและเครื่องมือวัดอัตราการไหลมีสัญญาณรบกวนค่อนข้างสูง

การควบคุมแบบ D สามารถทำงานร่วมกับการควบคุมแบบ P และ I หรือที่เรียกว่าตัวควบคุมพีไอดี ซึ่งมีสมการคือ

$$p(t) = \bar{p} + K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de}{dt} \right] \quad (3-13)$$

ซึ่งมีฟังก์ชันถ่ายโอนคือ

$$\frac{P'(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (3-14)$$

แต่อย่างไรก็ตามอุปกรณ์อิเล็กทรอนิกส์และนิวแมติกส์ที่เป็นตัวควบคุมแบบ D ในทางอุดมคติไม่สามารถสร้างขึ้นมาได้จริงเนื่องจากข้อจำกัดทางกายภาพ จึงจำเป็นต้องมีการประมาณพฤติกรรมทางอุดมคติของตัวควบคุมในสมการที่ (3-14) ด้วยฟังก์ชันถ่ายโอน

$$\frac{P'(s)}{E(s)} = K_p \left(\frac{T_i s + 1}{T_i s} \right) \left(\frac{T_d s + 1}{\alpha T_d + 1} \right) \quad (3-15)$$

เมื่อ α เป็นค่าระหว่าง 0.05 – 0.2

ข้อเสียอีกประการหนึ่งของตัวควบคุมพีไอดีทางอุดมคติคือเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายอย่างรวดเร็วจะทำให้ค่าจากเทอม D มีค่ามากและทำให้เกิดค่าควบคุม D พุ่งสูง (Derivative Kick) ในอุปกรณ์ควบคุมตัวสุดท้าย การหลีกเลี่ยงความเสียหายที่อาจเกิดขึ้นนั้นสามารถทำได้โดยเปลี่ยนแปลงค่าการควบคุมแบบ D โดยใช้ค่าสัญญาณจากเครื่องมือวัดแบบ B แทนการใช้ค่าสัญญาณผิดพลาด โดยแทน $\frac{de}{dt}$ ด้วย $-\frac{dB}{dt}$ ในสมการที่ (3-15) ได้

$$p(t) = \bar{p} + K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt - T_d \frac{dB}{dt} \right] \quad (3-16)$$

การกำจัดค่าควบคุม D พุ่งสูงเป็นทฤษฎีมาตรฐานที่นิยมใช้กันในตัวควบคุมส่วนใหญ่แต่ในบางทฤษฎีจะกำจัดค่าเป้าหมายในเทอม P และเทอม D เพื่อป้องกันการเกิดค่าควบคุม P พุ่งสูง (Proportional Kick) เมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย แต่ทฤษฎีนี้ยังไม่เป็นที่ยอมรับกันอย่างแพร่หลาย

3.5.4 สมการ PID สำหรับการคำนวณด้วยคอมพิวเตอร์

จากสมการที่กล่าวมาข้างต้น คอมพิวเตอร์ไม่สามารถนำสมการนี้ไปใช้ในการประมวลผลได้โดยตรงจึงต้องทำการแปลงเป็นสมการดิฟเฟอเรนเชียลต่อเนื่องแบบย่อย (Discrete Differential Equation) เพื่อให้คอมพิวเตอร์สามารถทำการประมวลผลได้ดังนี้

สมการ PID พื้นฐานคือ

$$p(t) = V_0(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (3-17)$$

จากสมการที่ (2-17) ได้สมการอนุพันธ์อันดับหนึ่งคือ

$$\begin{aligned} \frac{dV_o(t)}{dt} &= K_p \left[\frac{de(t)}{dt} + \frac{1}{T_i} \frac{d}{dt} \int e(t) dt + T_d \frac{d}{dt} \left[\frac{de(t)}{dt} \right] \right] \\ &= K_p \left[\frac{de(t)}{dt} + \frac{1}{T_i} e(t) + T_d \frac{d^2 e(t)}{dt^2} \right] \end{aligned} \quad (3-18)$$

กำหนดให้

$$\begin{aligned} e(t) &= e_n \\ \frac{dV_o(t)}{dt} &= \frac{\Delta V_o}{\Delta t} = \frac{V_{o_n} - V_{o_{n-1}}}{\Delta t} \\ \frac{de(t)}{dt} &= \frac{\Delta e}{\Delta t} = \frac{e_n - e_{n-1}}{\Delta t} \end{aligned}$$

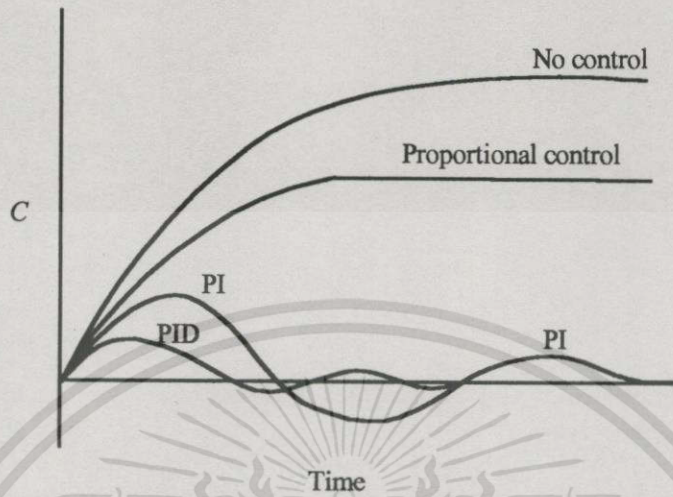
ดังนั้น

$$\begin{aligned} \frac{V_{o_n} - V_{o_{n-1}}}{\Delta t} &= K_p \left[\frac{e_n - e_{n-1}}{\Delta t} + \frac{1}{T_i} e_n + T_d \frac{d}{dt} (e_n - e_{n-1}) \right] \\ &= K_p \left[\frac{e_n - e_{n-1}}{\Delta t} + \frac{1}{T_i} e_n + T_d \frac{(e_n - e_{n-1}) - (e_{n-1} - e_{n-2})}{\Delta t^2} \right] \\ V_{o_n} - V_{o_{n-1}} &= K_p \left[(e_n - e_{n-1}) + \frac{1}{T_i} e_n \Delta t + T_d \frac{(e_n - 2e_{n-1} + e_{n-2})}{\Delta t} \right] \\ V_{o_n} &= V_{o_{n-1}} + K_p \left[(e_n - e_{n-1}) + \frac{\Delta t}{T_i} e_n + \frac{T_d}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right] \end{aligned} \quad (3-19)$$

เมื่อ

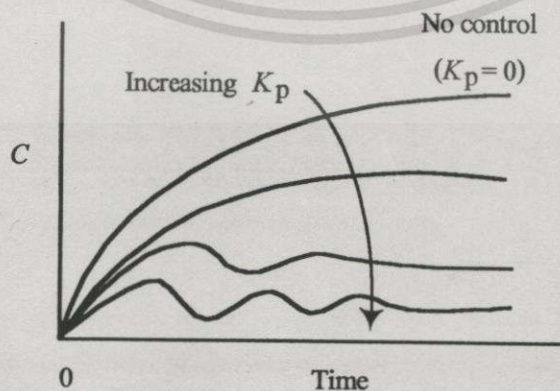
- $V_o(t)$: สัญญาณเอาต์พุตที่ส่งออกไปควบคุมคอนโทรลลอร์ด
- SP : Setpoint (0 - 100%)
- K_p : Proportional Gain (1 - 999.9)
- T_i : Integral Time (1 - 9999 Sec)
- T_d : Derivative Time (0 - 9999 Sec)

3.5.5 ผลตอบสนองของกระบวนการต่อการควบคุมแบบป้อนกลับ



รูปที่ 3.7 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด

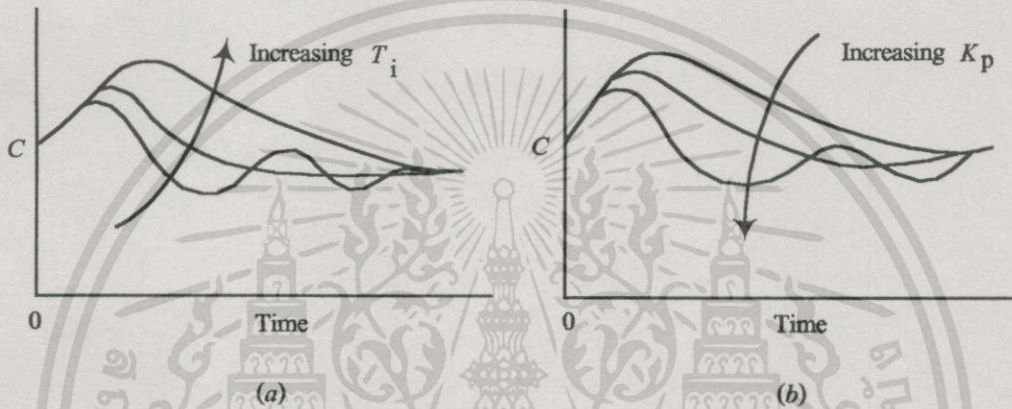
ในรูปที่ 3.7 แสดงผลตอบสนองของกระบวนการหลังจากมีการเปลี่ยนแปลงค่าเป้าหมายแบบขั้นบันได ผลตอบสนองของกระบวนการแทนด้วย C ซึ่งแสดงการเปลี่ยนแปลงจากค่าเป้าหมายเริ่มต้น หากไม่มีการควบคุมแบบป้อนกลับกระบวนการจะต้องใช้เวลานานมากในการเข้าสู่สภาวะคงที่ การควบคุมแบบ P จะเร่งการตอบสนองของกระบวนการและลดค่าออฟเซต การควบคุมแบบ PI จะกำจัดค่าออฟเซตแต่ทำให้ผลตอบสนองเกิดการแกว่ง ส่วนการควบคุมแบบ PID จะลดองศาการแกว่งและลดเวลาในการตอบสนองลง จึงสามารถสรุปได้ว่าการใช้ตัวควบคุม P , PI และ PID จะไม่มีผลต่อผลตอบสนองของกระบวนการที่มีการแกว่งเสมอไป แต่จะขึ้นอยู่กับ การเลือกค่าพารามิเตอร์ของตัวควบคุม K_p , T_i , T_d และส่วนประกอบอื่นๆของกระบวนการ ซึ่งในรูปที่ 3.7 แสดงตัวอย่างของผลตอบสนองที่เกิดขึ้นในทางปฏิบัติ



รูปที่ 3.8 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ P เมื่อมีการเปลี่ยนแปลงค่า K_p

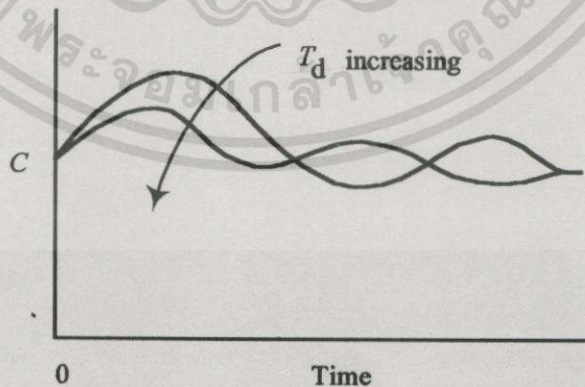
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 - 3.10 แสดงผลกระทบเชิงคุณภาพเมื่อมีการเปลี่ยนแปลงค่าพารามิเตอร์ของตัวควบคุม ในรูปที่ 3.8 แสดงผลจากการเปลี่ยนแปลงอัตราขยาย K_p ของกระบวนการ ตามปกติแล้ว การเพิ่มอัตราขยายของตัวควบคุมมีแนวโน้มที่จะทำให้การตอบสนองของกระบวนการเร็วขึ้น แต่อย่างไรก็ตามถ้า K_p มีค่ามากเกินไปก็จะทำให้ผลตอบสนองเกิดการแกว่งที่ไม่ต้องการหรืออาจทำให้ระบบขาดเสถียรภาพได้ ดังนั้นการตั้งค่า K_p ให้พอเหมาะจึงให้ผลตอบสนองที่ดี ทั้งในการควบคุมแบบ P และ PI รวมถึงการควบคุมแบบ PID ด้วย



รูปที่ 3.9 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อเปลี่ยนแปลงค่า T_i และ K_p

ในรูปที่ 3.9 การเพิ่มค่า T_i ทำให้การควบคุมแบบ PI และ PID มีผลตอบสนองของกระบวนการที่ช้าลง ในทางทฤษฎีออฟเซตจะถูกกำจัดได้ในทุกค่าของ T_i ซึ่งมีค่าระหว่าง $0 - \infty$ แต่เมื่อมีการเปลี่ยนแปลงโหลดหรือค่าเป้าหมาย T ที่มีค่ามากจะทำให้ผลตอบสนองเข้าสู่ค่าเป้าหมายได้ช้าลง



รูปที่ 3.10 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PID เมื่อมีการเปลี่ยนแปลงค่า T_d

ผลของ T_d ที่มีต่อกระบวนการนั้นสรุปได้ยาก เพราะที่ T_d มีค่าน้อยการเพิ่มค่า T_d จะให้ผลตอบสนองที่ดีขึ้นโดยจะลดเวลาในการตอบสนอง, ลดการการเบี่ยงเบนและลดองศาการแกว่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังแสดงในรูปที่ 3.10 แต่อย่างไรก็ตามหาก T_d มีค่ามาก สัญญาควบคุมของเครื่องมือวัดจะถูกขยาย และผลตอบสนองจะเกิดการแกว่ง ดังนั้นจึงควรตั้งค่า T_d ให้เหมาะสมในการใช้งาน

3.6 บทสรุป

ในบทนี้ได้แสดงถึงความเป็นมาของระบบควบคุมป้อนกลับซึ่งเริ่มการใช้งานมาตั้งแต่อดีตมาจนถึงปัจจุบัน ซึ่งปัจจุบันได้มีการประยุกต์เอาระบบคอมพิวเตอร์อันทันสมัยเข้ามาควบคุมกระบวนการในอุตสาหกรรมทำให้ผู้ปฏิบัติงานสามารถปฏิบัติงานได้ง่ายและสะดวกรวดเร็วเป็นอย่างมากซึ่งส่วนมากการควบคุมกระบวนการจะอาศัยตัวควบคุมแบบพีไอดีเข้ามาใช้งานในการควบคุมกระบวนการซึ่งจำเป็นต้องเข้าใจถึงการควบคุมแบบต่าง ๆ ทั้ง Proportional (P) , Integral (I) และ Derivative (D) โดยการควบคุมแบบ P จะเป็นการควบคุมที่แปรผันตามสัญญาณที่ผิดพลาดระหว่างค่าเป้าหมายที่กำหนดไว้กับผลตอบสนองของกระบวนการ แต่การควบคุมแบบ P จะทำให้เกิดออฟเซตซึ่งจะใช้การควบคุมแบบ I เพื่อให้ผลตอบสนองดีขึ้นเนื่องจากการควบคุมแบบ I จะมีผลเพื่อลดออฟเซตได้อัตโนมัติแต่การควบคุมประเภทนี้จะมีการตอบสนองแบบแกว่งโดยปกติสามารถเพิ่มตัวควบคุมแบบ D เพื่อลดผลตอบสนองที่ไม่มีเสถียรภาพจากการแกว่งลงได้ เพราะคุณสมบัติที่สำคัญของการควบคุมแบบ D คือการควบคุมแบบอัตราส่วนหรือมีการทำนายล่วงหน้า ดังนั้นเมื่อนำเอาคุณสมบัติที่สำคัญของตัวควบคุมทั้ง 3 แบบมารวมกันเป็นตัวควบคุมแบบพีไอดีแล้ว จะส่งผลให้การควบคุมกระบวนการมีผลตอบสนองของการควบคุมเข้าสู่ค่าเป้าหมายได้ง่ายขึ้น

บทที่ 4

ภาษาจาวา (JAVA)

4.1 บทนำ

ภาษาจาวาเป็นภาษาเหมาะสมในการพัฒนาเพื่อสื่อสารแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ต โดยสามารถสื่อสารระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้เป็นอย่างดี ซึ่งจาวาได้ถูกนำมาเป็นส่วนหนึ่งของเว็บเพจเพื่อเพิ่มประสิทธิภาพในการทำงานเมื่อผู้ใช้หรือไคลเอนท์สามารถทำงานได้ต่อกับโฮมเพจนั้น ๆ และมีการเชื่อมโยงแลกเปลี่ยนข้อมูลกับเครื่องเซิร์ฟเวอร์ที่ใช้ควบคุมกระบวนการ นอกจากนี้จาวาได้ถูกออกแบบให้มีความปลอดภัยในการใช้งานระบบเครือข่ายอินเทอร์เน็ตอีกด้วย ในบทนี้จะกล่าวถึงความเป็นมาของภาษาจาวารวมถึงโครงสร้างและการทำงานตลอดจนข้อมูลที่เกี่ยวข้องกับภาษาจาวา

4.2 ความเป็นมาของภาษาจาวา

ภาษาจาวา หรือ Java Programming Language ถูกพัฒนาขึ้นโดยบริษัท ซัน ไมโครซิสเต็มส์ เริ่มแรกซันได้คิดค้นภาษาจาวาขึ้นมาเพื่อใช้กับระบบปฏิบัติการ Set Top Box คือโครงการของระบบตลาดอิเล็กทรอนิกส์ ที่ต้องการให้ระบบสามารถทำการควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพลตฟอร์ม (Platform) ที่ใช้ คือการพัฒนาระบบปฏิบัติการหรือ ซอฟต์แวร์ให้สามารถทำงานได้ โดยไม่สนใจว่าไมโครโพรเซสเซอร์ที่ใช้อยู่จะเป็นรุ่นใด ยี่ห้อไหน ฮาร์ดแวร์เป็นของอะไร ทั้งยังได้รวมความสามารถไปถึงการเชื่อมต่อ การเปลี่ยนข้อมูลข่าวสารทั่ว ๆ ไปและทางด้านธุรกิจผ่านทางระบบเครือข่ายของคอมพิวเตอร์อีกด้วย

ทางทีมงานซึ่งพัฒนาโปรแกรมนี้ได้เห็นถึงการเจริญเติบโตและความสามารถในการแลกเปลี่ยนข่าวสารบนอินเทอร์เน็ต จึงได้วางแผนและพัฒนาระบบอินเทอร์เน็ตเฟสแบบใหม่เพื่อใช้กับอินเทอร์เน็ตขึ้น ด้วยสาเหตุนี้เองภาษาจาวาจึงได้บังเกิดขึ้น รวมทั้งยังได้มีการเปิดตัวเบราเซอร์ที่ชื่อว่า Hotjava เพื่อใช้สำหรับแสดงความสามารถของภาษาจาวา

ภาษาจาวาเป็นภาษาที่พัฒนาขึ้นมาเพื่อสามารถนำเอามาใช้งานในเครื่องชนิดอื่น ๆ เช่น เมื่อนำมาคอมไพล์ในเครื่องชนิดหนึ่ง จะได้ผลลัพธ์เป็น ไบต์โค้ด จากนั้นในการทำงานของภาษาจาวาจะใช้ตัวรันซึ่งมีชื่อเรียกย่อว่า JVM (Java virtual machine) จะทำหน้าที่ในการแปลงคำสั่งในลักษณะไบต์โค้ดให้กลายเป็นภาษาเครื่องของแต่ละเครื่องนั้น ดังนั้นในการทำงานต่างเครื่องคือ เมื่อมีการ

คอมไพล์เป็นไบต์โค้ดอีกเครื่องในการนำไปรันอีกเครื่องจำเป็นจะต้องมีตัวแปลภาษา ให้เป็นภาษาเครื่องชนิดนั้น ๆ ด้วย

4.3 ความสามารถของภาษาจาวา

ลักษณะของภาษาจาวาจะเป็นการเขียนโปรแกรมอ้างอิงเชิงวัตถุ OOP (Object Oriented Programming) สรุปความสามารถในการทำงานได้ดังต่อไปนี้

1. การทำงานที่ไม่ขึ้นอยู่กับแพลตฟอร์ม โดยใช้วิธีการแปลงข้อมูลของคอมไพล์เลอร์จากโค้ดโปรแกรม (source code) ไปเป็นไบต์โค้ด (byte code) ที่ภาษาจาวารู้จักสามารถทำงานได้
2. การใช้เมธอดจัดการกับข้อมูลแทนการใช้ฟังก์ชัน
3. ความสามารถในการโต้ตอบแบบ Inter Active กับงานด้านมัลติมีเดีย
4. ความสามารถในการดึงเอาคลาสไลบรารีจากที่ต่าง ๆ ผ่านทาง HTTP และ FTP
5. ความสามารถในการเล่นไฟล์เสียง ไฟล์วิดีโอและไฟล์ภาพกราฟิกเคลื่อนไหว
6. การโต้ตอบแบบทันทีต่อเหตุการณ์
7. เป็นระบบ แบ่งงานออกเป็นที่ทำออกเป็นส่วน ๆ คือสามารถทำงานได้หลายงานพร้อมกันในเวลาเดียว
8. ไม่มีตัวแปรประเภทพอยน์เตอร์จึงหมดปัญหาเกี่ยวกับการขอใช้หน่วยความจำโดยตรง และสามารถจะทำการ คือค่าให้แก่หน่วยความจำได้อีกด้วยในกรณีที่ใช้งานตัวแปรนั้นเรียบร้อยแล้ว
9. มีความปลอดภัยสูง ด้วยความสามารถที่ไม่ให้ผู้ใช้เข้าไปยุ่งเกี่ยวกับหน่วยความจำของระบบจึงปลอดภัยจากไวรัส
10. การดาวน์โหลดซอฟต์แวร์ใหม่ ๆ และอัพเกรดเวอร์ชันใหม่ได้อย่างง่ายโดยผ่านทางเว็บเพจ

4.4 การประยุกต์ใช้งาน

จากความสามารถดังกล่าวผ่านมาข้างต้นคงพอจะเป็นข้อสรุปได้แล้วว่าจาวาเป็นภาษาโปรแกรมที่มีความสามารถรอบด้าน อีกทั้งยังมีความยืดหยุ่นภายในตัวสูง การที่จะนำมาสร้างและพัฒนางานดี ๆ มีประสิทธิภาพคงจะทำได้ไม่ยากนัก เราสามารถใช้จาวาสร้างและพัฒนาแอปพลิเคชันต่าง ๆ โดยยกตัวอย่างพอสังเขปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไคลแอนท์

จาวาสามารถเป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้านไคลแอนท์ได้เพียงพอโดยเฉพาะด้านกราฟิกและระบบติดต่อกับผู้ใช้ เพราะปัจจุบัน โปรแกรมที่เขียนขึ้นจากภาษาจาวาจะทำงานที่ฝั่งไคลแอนท์เป็นส่วนมากจะเห็นได้จาก Java Applet ต่าง ๆ บนเบราว์เซอร์แต่ละจะให้ดีและสมบูรณ์ยิ่งขึ้นต่อความเป็น Client Software ที่ดีก็คือ การติดต่อบริหารข้อมูลจากไคลแอนท์ไปเป็นเซิร์ฟเวอร์ แม้ว่าในปัจจุบันจาวาจะสามารถทำได้แล้วแต่ต้องใช้เครื่องมือชนิดอื่นเป็นตัวช่วย เช่น การเชื่อมต่อกับฐานข้อมูลแบบ ODBC (Object-Oriented Database Connectivity) ซึ่งการทำงานของทางฝั่งเซิร์ฟเวอร์ต้องใช้ภาษาคอมพิวเตอรชนิดอื่นช่วยเพื่อการติดต่อกับฐานข้อมูล

เซิร์ฟเวอร์

จาวามือใช้สร้างโปรแกรมประยุกต์ที่รันบนเซิร์ฟเวอร์นั้น จาวาจำเป็นที่จะต้องได้รับการปรับปรุงการทำงานอยู่สามอย่างคือ การเชื่อมต่อกับฐานข้อมูลโดยตรง การทำงานให้สอดคล้องกับอินเทอร์เน็ต และความเร็วของการรันต้องเพิ่มขึ้นเท่า ๆ กับ โปรแกรมประยุกต์ทั่วไป

ระบบ JDBC (Java Database Connectivity) ได้เข้ามาแก้ปัญหาในส่วนของการทำงานติดต่อกับฐานข้อมูลโดยตรง โดยการร่วมมือกันของบริษัทซอฟต์แวร์ที่มีชื่อเสียงหลายบริษัท ได้สร้างมาตรฐานการเชื่อมต่อกับฐานข้อมูลของจาวาขึ้น แต่ในขณะนี้ก็ยังไม่มีที่รู้จักแพร่หลายนักสำหรับ JDBC คาดว่าในอนาคตอันใกล้นี้การเชื่อมต่อนี้จะทำได้ง่ายขึ้นจากไคลแอนท์สู่เซิร์ฟเวอร์โดยใช้ จาวา

ระบบรักษาความปลอดภัย

จาวาได้สร้างสภาพแวดล้อมเสมือน "Java Virtual Machine" ซึ่งจะอนุญาตให้เฉพาะ Java Applets หรือ Java Application ซึ่งเป็นไบนารีโค้ดเท่านั้นที่สามารถรันได้โดยที่ไบนารีโค้ดจะปราศจากไวรัสหรือส่วนที่จะทำอันตรายต่อระบบ อันเนื่องจาก Java Machine จะกำจัดการเข้ารหัสใช้ทรัพยากรของไบนารีโค้ดในอนาคตรักษาความปลอดภัยอีกระบบหนึ่งกำลังถูกนำมาปรับใช้คือ "การเข้ารหัสข้อมูล" เพิ่มความปลอดภัยของข้อมูล

การใช้จาวาประยุกต์สร้างและพัฒนางานยังสามารถแบ่งเป็นประเภทตามลักษณะของกลุ่มงานได้ดังนี้

- งานด้านการศึกษา โดยจัดทำเป็นลักษณะสั่งการสอนคล้ายกับ CAI มีความสามารถในเชิงโต้ตอบระหว่างผู้ใช้งานกับคอมพิวเตอร์
- ตอบสนองงานทางด้านเว็บเพจ ทำให้งานสร้างเว็บเพจดูมีชีวิตชีวาขึ้น โดยการนำเอาความสามารถทางด้านมัลติมีเดียมาใช้ได้อย่างเต็มที่

- สร้างสรรค์เครื่องมือพัฒนา (Software Developer tool kits)
- สร้างแอปพลิเคชันทางธุรกิจ มีความสามารถในการเรียกหาข้อมูลผ่านทางระบบเครือข่าย
- พัฒนาเกม จะมีความเป็นมัลติ คือสามารถเล่นได้หลาย ๆ คนพร้อมกันผ่านทางระบบเครือข่ายหลากหลายพื้นที่ไม่ว่าจะเล่นด้วยกัน ณ ตำแหน่งใด ทั่วทุกมุมโลก

4.5 คอมพิวเตอร์จำลองของจาวา (Java Virtual Machine)

คอมพิวเตอร์จำลองของจาวา (Java Virtual Machine) เป็นหลักการสร้างโดยการสมมติให้มีคอมพิวเตอร์อีกเครื่องหนึ่งมาโดยเครื่องนี้จะใช้ในการคอมไพล์โปรแกรมภาษาจาวาทุกโปรแกรม เมื่อต้องการให้โปรแกรมภาษาจาวาไปทำงานบนคอมพิวเตอร์จริง ๆ เครื่องใด เราก็เพียงแต่สร้างตัวอินเตอร์พรีเตอร์ (Interpreter) ของคอมพิวเตอร์จำลองตัวนั้นบนเครื่องนั้น ๆ ภาษาจาวาทุกโปรแกรมก็จะสามารถทำงานระบบคอมพิวเตอร์นั้น ๆ ได้ตามต้องการ ด้วยหลักการนี้ก็เป็นที่มาของคุณสมบัติของภาษาจาวา ที่ไม่ขึ้นกับแพลตฟอร์มและฮาร์ดแวร์ใด ๆ หรือพอร์ตเทเบิล (Portable) เราอาจจะเรียก Java Virtual Machine สั้น ๆ ว่า JVM ก็ได้

สำหรับเหตุผลที่ต้องมี JVM นี้ จริง ๆ แล้วเป็นการกำหนดขึ้นมา เพื่อเป็นคำจำกัดความเฉพาะในความคิดเท่านั้น เพื่อให้ นักพัฒนาจะได้ไม่ถูกบังคับให้ต้องสร้างตัวอินเตอร์พรีเตอร์ตามแนวทางแบบใดแบบหนึ่งโดยเฉพาะ แต่ตัวอินเตอร์พรีเตอร์ที่สร้างขึ้นตามข้อกำหนดดังกล่าว ไม่ว่าจะอยู่บนแพลตฟอร์มใด จะสามารถรันโปรแกรมที่เขียนขึ้นในภาษาจาวาได้ โดยให้ผลลัพธ์ออกมาเหมือนกัน

แต่ในข้อกำหนดของ JVM ก็มีกำหนดยุทธศาสตร์ที่ชัดเจนมาก ๆ เกี่ยวกับการออกแบบอินเตอร์พรีเตอร์ในหลาย ๆ ส่วน โดยเฉพาะอย่างยิ่งส่วนที่เกี่ยวข้องกับการกระจายโค้ดของจาวาไปใส่ไว้ในรูปแบบที่กำหนด ข้อกำหนดนี้ได้แก่ วิศวกรของออบเจ็คต์และโอเปอร์เรนด์ พร้อมกับคำประจำตัว การจัดโครงสร้างของโค้ด การจัดวางรูปแบบของออบเจ็คต์ของจาวา ข้อกำหนดต่าง ๆ เหล่านี้จะทำให้นักพัฒนาอินเตอร์พรีเตอร์ทั้งหลาย สามารถจะสร้างอินเตอร์พรีเตอร์ขึ้นมาใหม่บนแพลตฟอร์มใด ๆ ก็ได้ ดังนั้น การพัฒนาภาษาจาวา จึงไม่ได้ถูกปิดกั้นอยู่กับระบบของซันซึ่งเป็นผู้คิดค้นภาษาจาวานี้ขึ้นมาเท่านั้น

จากที่กล่าวมาข้างต้นถึงคุณสมบัติของจาวาที่ไม่ยึดติดอยู่กับแพลตฟอร์มใด ๆ ทำให้การทำงานกับระบบคอมพิวเตอร์ที่ทำงานแบบกระจาย (Distributed Computing) ได้รับการตอบสนองอย่างเหมาะสมนั่นคือ ทำให้เกิดความพอร์ตเทเบิลนั่นเอง

อย่างไรก็ตาม JVM ก็ยังมีข้อจำกัดอยู่ ข้อจำกัดของ JVM นั้นจะอยู่ที่ข้อจำกัดของการออกแบบตัวอินเตอร์พรีเตอร์แทน เช่น การจำกัดค่าของโอเปอร์เรนด์ และขนาดของสแต็ค เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ใดที่นำเอกสารฉบับนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีนำไปใช้

ข้อกำหนดเหล่านี้หมายถึงว่า JVM สามารถเข้าถึงหน่วยความจำได้เฉพาะในห่วงแอดเดรสเท่าที่มีอยู่เท่านั้น

ข้อจำกัดภายในของตัว JVM เองนั้นมีห่วงแอดเดรสให้ใช้อยู่ถึง 4 GB เพราะขนาดของความกว้างของการอ้างแอดเดรสเป็น 32 บิต เมธอดต่าง ๆ ของจาวามีขนาดได้เพียง 32 KB เพราะมีข้อจำกัดของการใช้คำสั่งกระโดด เป็นแบบ 16 บิต (โดยมีบิตแรกเป็นตัวบอกว่ากระโดดไปข้างหน้าหรือข้างหลังและบิตต่อ ๆ ไป บอกระยะทางการกระโดดจากจุดที่เหล่านี้มีขนาดเพียง 8 บิต นอกจากนั้น จำนวนของค่าคงที่ที่อยู่ในส่วนกลาง ยังถูกจำกัดด้วยขนาดของดัชนี 16 บิต ทำให้มีจำนวนได้เพียง 32,000 ค่าในแต่ละเมธอด

การที่เราถือว่า การมีค่าเหล่านี้ เป็นข้อจำกัด ก็อาจจะเป็นการมองการณ์ไกลไปสักหน่อย เนื่องจากปัจจุบันเครื่องคอมพิวเตอร์ส่วนใหญ่มีหน่วยความจำเพียง 16 หรือ 32 MB กันเท่านั้น หน่วยความจำขนาด 4 GB จึงยังเป็นเรื่องที่ไม่ต้องคิดหนักในตอนนี้ ในขณะที่ขนาดของเมธอดที่จำกัดที่ 32 KB ก็เป็นเพียงเมธอดเดียวกันเท่านั้น

4.6 หลักการทำงานของภาษาจาวา

ในที่นี้ขออธิบายการทำงานของภาษาจาวาเริ่มตั้งแต่การคอมไพล์ตัวโปรแกรมจนกระทั่งเราเรียกใช้งานในเว็บเพจ เป็นดังนี้

การคอมไพล์

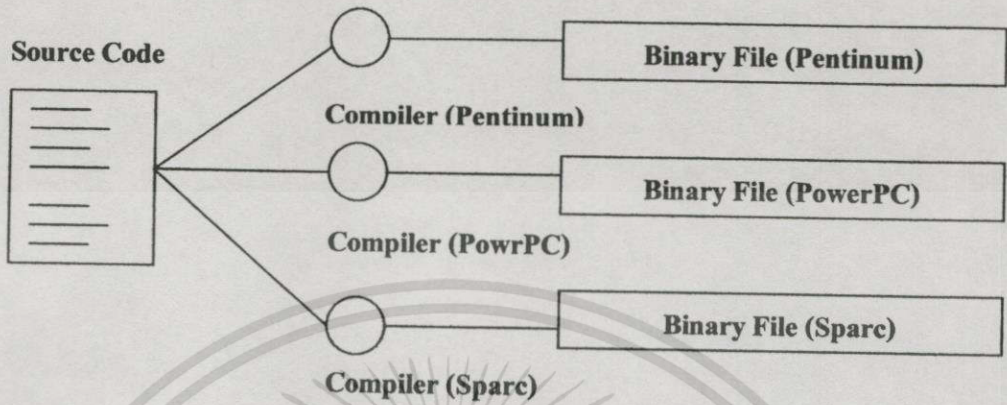
คอมไพเลอร์ของภาษาจาวาก็เช่นเดียวกับคอมไพเลอร์ในภาษาอื่น ๆ นั่นคือ มันจะสร้างรหัสภาษาเครื่อง (Machine Code หรือ Assembler Code) จากภาษาในระดับที่สูงกว่าเพื่อให้ซีพียู (CPU : Central Processing Unit) สามารถนำไปใช้งานได้

แต่ข้อแตกต่างที่สำคัญระหว่างคอมไพเลอร์ของภาษาจาวากับภาษาอื่น ๆ คือ ซีพียูหรือโปรเซสเซอร์ที่จะทำหน้าที่ในการปฏิบัติตามคำสั่งที่ได้จากการคอมไพล์ภาษาจาวานั้น ไม่มีอยู่จริง เป็นเพียงสิ่งที่สมมติขึ้นมาที่เรียกว่า Java Virtual Machine นอกจากนี้การอ้างถึงส่วนต่าง ๆ ของโปรแกรมที่คอมไพล์ด้วยคอมไพเลอร์ของภาษาจาวาก็จะมีวิธีการที่แตกต่างออกไป

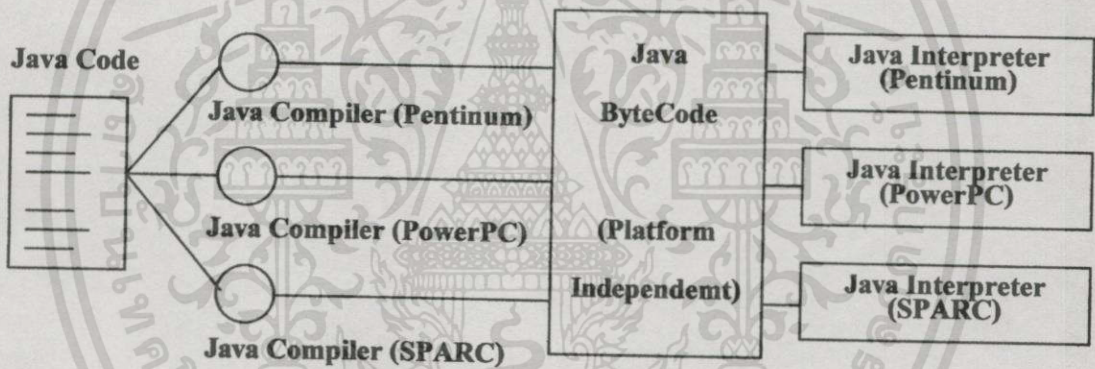
คอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนการอ้างถึงส่วนของโปรแกรมจากการ ใช้ชื่อแบบในภาษาสูง ไปเป็นตัวเลขเหมือนคอมไพเลอร์ภาษาอื่น ๆ ทำกัน และคอมไพเลอร์ภาษาจาวาก็จะไม่มีการสร้างแผนที่ของการจัดวางโปรแกรมบนหน่วย ความจำขึ้นมาในระหว่างการคอมไพล์ด้วยเหตุผลที่สำคัญ คือ เพื่อเป็นการสร้างความพอร์ตเทเบิลให้กับตัวโปรแกรมเพราะการจัดวางตำแหน่งของโปรแกรมจะต้องขึ้นอยู่กับลักษณะการทำงานของโปรเซสเซอร์ตัวใดตัวหนึ่ง การยังไม่จัดวางตำแหน่งช่วยให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลาง สามารถนำไปใช้บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น เมื่อนุญาตให้ไปใช้ในที่อื่นเป็นการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์แพลตฟอร์มอื่น ๆ ได้ นอกจากนั้นยังทำให้เกิดความปลอดภัยอีกด้วยสิ่งที่ได้จากการคอมไพล์ในภาษาจาวาเราเรียกว่าไบนารีโค้ด



รูปที่ 4.1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน



รูปที่ 4.2 การประมวลผลของโปรแกรมภาษาจาวา

การวางตำแหน่งในหน่วยความจำ

ในภาษาจาวาจะไม่มีกรลดรูปแบบการอ้างถึงสิ่งต่าง ๆ ของโปรแกรมจากการเรียกเป็นชื่อให้เหลือเพียงตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งของโปรแกรมลงในหน่วยความจำคอมพิวเตอร์ภาษาจาวาจะทิ้งชื่อของแต่ละส่วนของโปรแกรม (โดยเฉพาะเมธอด) เอาไว้ในตัวโปรแกรมที่สร้างขึ้น เมื่อโปรแกรมทำงาน จะเป็นหน้าที่ของตัวอินเทอร์พรีเตอร์ที่จะคอยเปิดตารางค้นหาที่อยู่ของคอมไพล์ที่ต้องการเรียกใช้งาน โดยก่อนที่จะเริ่มทำงานจริง อินเทอร์พรีเตอร์จะต้องสร้างแผนที่ในการจัดวางสิ่งต่าง ๆ ลงในหน่วยความจำขึ้นมาแล้วจึงสร้างตารางขึ้นมาเพื่อช่วยหาคำแหน่งของคอมไพล์เมื่อมีการเรียกใช้งานโดยใช้ชื่อของคอมไพล์

การรันโปรแกรม

การรันโค้ดที่คอมไพล์เอาไว้สำหรับ Java Virtual Machine เป็นหน้าที่ของตัวอินเทอร์พรีเตอร์การรันโปรแกรมจะแบ่งได้เป็น 3 ขั้นตอนหลัก ๆ คือ การอ่าน การตรวจสอบความถูกต้อง และการทำตามโค้ด หน้าที่ในการอ่านโค้ดเข้าสู่ระบบจะเป็นของ Class Loader หน้าที่การทำงานในส่วนนี้จะไม่ได้อ่านเข้ามาเฉพาะไฟล์จาวาที่กำลังจะเรียกใช้เท่านั้น แต่จะอ่านคลาสที่มีการอ้างอิงและคลาสที่มีการ inherited มาโดยคลาสที่อ้างอิง เมื่อผ่านขั้นตอนนี้แล้ว โค้ดทั้งหมดก็จะถูกส่งผ่านตัวตรวจสอบไบนารีโค้ดเพื่อให้แน่ใจว่าโค้ดที่ส่งมามีความถูกต้องตามมาตรฐานของจาวา และจะไม่รบกวนเสถียรภาพของระบบ เมื่อผ่านการตรวจสอบแล้ว โค้ดก็จะถูกส่งต่อไปยังระบบใหม่ (Run-Time System) ซึ่งจะส่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่งซึ่งหลักการทำงานในแต่ละขั้นตอนนี้กล่าวมาข้างต้นเป็นดังนี้

คลาสโหลดเดอร์

ตัวคลาสโหลดเดอร์ (Class Loader) จะทำหน้าที่ดึงโค้ดทั้งหมดที่จำเป็นในการทำงานของแอปพลิเคชัน ไม่ว่าจะเป็คลาสที่ถูก Inherited มาหรือคลาสอื่น ๆ ที่มีการเรียกใช้ เมื่อโหลดเดอร์ดึงคลาสใดเข้ามาแล้วก็จะจัดคลาสนั้น ๆ ใส่เข้าไปใน namespace ของมันเอง โดยการเก็บ จะใช้ชื่อของคลาสเป็นสำคัญ ไม่ได้ใช้การอ้างอิงเป็นตัวเลข หลักการนี้ก็จะเหมือนกันกับการทำงานของ Virtual Machine ที่โอเอส (OS : Operating System) สร้างขึ้นให้แอปพลิเคชันแต่ละตัวทำงาน ถ้าไม่ได้มีการเจาะจงเรียกใช้คลาสที่อยู่นอก namespace นี้ การเรียกใช้ชื่อต่าง ๆ ในคลาส ก็จะไม่มีการรบกวนกันระหว่างคลาสเลย

คลาสทั้งหมดที่อยู่บนเครื่องโลคอลเอง จะได้รับห้วงแอดเดรส (Address Space) เป็นตัวเอง ส่วนคลาสต่าง ๆ ที่ดึงมาจากภายนอกจะได้รับ namespace เป็นของตัวเอง การทำงานลักษณะนี้จะช่วยให้คลาสที่อยู่บนโลคอลทำงานได้ประสิทธิภาพดีขึ้น เพราะใช้ namespace ร่วมกันได้ แต่ก็ยังมีการป้องกันความผิดพลาดที่อาจจะเกิดจากคลาสที่ดึงเข้ามาจากภายนอก และในทางกลับกัน คลาสที่อิมพอร์ตเข้ามาก็ปลอดภัยจากความผิดพลาดที่อาจจะเกิดจากคลาสโลคอลด้วย

เมื่อคลาสทั้งหมดที่เกี่ยวข้องกับการทำงานถูกอิมพอร์ตเข้ามาเรียบร้อยแล้ว การจัดวางหน่วยความจำสำหรับเริ่มต้นการทำงานก็จะเกิดขึ้นได้ การเรียกชื่อต่าง ๆ ก็จะสามารถจับคู่กับแอดเดรสจริง ๆ ของหน่วยความจำได้ แล้วตัวโหลดเดอร์จะสร้างตารางสำหรับค้นหาที่อยู่เสี่ยงต่อการทำงานผิดปกติของซูเปอร์คลาส (Super class) และการอ้างอิงแอดเดรสที่ไม่ถูกต้องได้

ตรวจสอบไบท์โค้ด

เมื่อโค้คเดินทางมาถึงขั้นตอนการสร้างตารางจับคู่ชื่อกับแอดเดรสแล้ว ก็ยังไม่สามารถแน่ใจได้ว่าโค้คที่อ่านเข้ามาจะมีความปลอดภัย ดังนั้นจึงต้องมีตัว Verifier หรือตัวตรวจสอบไบท์โค้ค ทำหน้าที่ตรวจสอบความถูกต้องที่ละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวา และสอดคล้องกับการทำงานของตัวโปรแกรมเองหรือไม่ การตรวจสอบโค้คในเชิงทฤษฎีจะสร้างปัญหาต่าง ๆ ได้หลายอย่าง เช่น จะไม่มีการสร้างพอยต์เตอร์ที่เกินกว่าหน่วยความจำจริง ไม่มีคำสั่งใดสามารถละเมิดสิทธิการทำงานของตัวโปรแกรมได้ ไม่มีการจับคู่ออบเจ็คผิด จะไม่มีการให้โอเปอเรนด์มากหรือน้อยเกินไป การกำหนดค่าต่าง ๆ สำหรับไบท์โค้คจะต้องถูกต้องครบถ้วน และจะไม่มีการแปลงข้อมูลผิดรูปแบบ

การใช้ตัวตรวจสอบสองจุดประสงค์ 2 ประการสำคัญ คือ สิ่งต่าง ๆ ดังกล่าวมาแล้วจะถูกตรวจสอบก่อนทำให้ตัวอินเตอร์พรีเตอร์มั่นใจได้ว่า ไบท์โค้คที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบ และจุดประสงค์ที่สองก็คือ ตัวอินเตอร์พรีเตอร์จะทำงานตามไบท์โค้คได้รวดเร็วกว่า เพราะไม่ต้องคอยระวังว่าจะมีปัญหาก่อขึ้น และไม่ต้องหยุดเป็นช่วง ๆ เมื่อพบปัญหาและต้องการแก้ไข

ในการทำงาน ไบท์โค้คจะถูกตรวจสอบเพียงครั้งเดียวเท่านั้น และจะทำงานไปได้ตลอดไม่ต้องการตรวจสอบซ้ำอีกเมื่อมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

การทำงานตามโค้ค

เมื่อตัวโหนดเคอร์ได้รวบรวมโค้คเข้ามาสู่ระบบทำการจัดวางในหน่วยความจำ และตัวตรวจสอบได้ทำการตรวจสอบความถูกต้องแล้ว โค้คก็จะถูกส่งต่อไปยังตัวอินเตอร์พรีเตอร์เพื่อทำงานตามคำสั่ง การทำงานตามคำสั่งของโค้คก็คือการเปลี่ยนโค้คให้กลายเป็นคำสั่ง การทำงานจริงที่ตัวระบบไคลเอ็นท์ที่รันโค้คนี้สามารถทำงานได้ ซึ่งวิธีการที่ทำได้ก็มีอยู่ 2 วิธีด้วยกันคือตัวอินเตอร์พรีเตอร์ทำการคอมไพล์โค้คเหล่านี้ให้กลายเป็นเนทีฟโค้คที่ตัวเครื่องไคลเอ็นท์เข้าใจ แล้วค่อยทำงาน เพื่อให้ได้ความเร็วสูงสุดในการทำงาน กับอีกวิธีหนึ่งก็คือ ตัวอินเตอร์พรีเตอร์อ่านโค้คเข้ามาแล้วตีความทำงาน ไปทีละคำสั่ง และทำการตีความไปเรื่อย ๆ ตลอดเวลาที่มีการทำงาน

โดยปกติแล้ว ผู้สร้างตัวอินเตอร์พรีเตอร์มักจะเลือกใช้วิธีการหลังรูปแบบของไบท์โค้คในภาษาจาวามีความยืดหยุ่นเพียงพอที่จะสามารถเปลี่ยนไปทำงานบนเครื่องไคลเอ็นท์แบบต่าง ๆ ได้ โดยไม่มีการก่อให้เกิดโอเวอร์เฮดมากมายเกินความจำเป็น อย่างไรก็ตามไคลเอ็นท์ของจาวาบางระบบจะมีความสามารถในการทำงานได้ทั้งสองวิธี คือ โปรแกรมเมอร์สามารถจะเลือกใช้วิธีการคอมไพล์กับงานที่เน้นการคำนวณมาก ๆ เพื่อเป็นการเพิ่มสมรรถนะในการทำงานให้ได้เต็มที่ ซึ่งไคลเอ็นท์แบบนี้จะให้ได้ทั้งความพอร์ตเทเบิล และสมรรถนะที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างระบบรันไทม์ที่ดีจะต้องถ่วงดุลความสำคัญ 3 ประการให้ได้พอเหมาะ นั่นคือ ความพอร์ตเทเบิล ความปลอดภัย และสมรรถนะเรื่องของความพอร์ตเทเบิลทำได้โดยการเลือกรูปแบบของไบท์โค้ดที่มีความเป็นกลางเพียงพอ สามารถนำไปใช้รันบนเครื่องคอมพิวเตอร์แบบต่าง ๆ ได้โดยง่ายนอกจากนั้น การที่ตัวอินเทอร์พรีเตอร์ทำการกำหนดการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์ แทนที่จะเป็นระหว่างการคอมไพล์เหมือนภาษาอื่น ๆ ก็เป็นการเพิ่มความแน่นอนว่า คลาสต่าง ๆ ที่อิมพอร์ตเข้ามาจะยังคงใช้ได้อยู่ตลอดเวลา เรื่องของความปลอดภัยนั้น เป็นสิ่งที่มีการคำนึงถึงอยู่ตลอดกระบวนการทำงานของระบบรันไทม์ โดยเฉพาะอย่างยิ่งในส่วนของตัวตรวจสอบไบท์โค้ด ที่ทำให้แน่ใจได้ว่าโปรแกรมจะทำงานได้ถูกต้องตามข้อกำหนดของจาวา ส่วนเรื่องของสมรรถนะนั้นก็จัดการได้ในสองระยะ คือ พยายามเอาโอเวอร์เฮดทั้งหลายไปใส่ไว้ที่ตอนเริ่มต้น โหลดโปรแกรมเข้ามาสู่ระบบหรือไม่ก็กำหนดให้ทำงานเป็นแบกกราวนด์ (Background Thread)

ด้วยสิ่งต่าง ๆ เหล่านี้ ทำให้จาวาสามารถปล่อยสมรรถนะระดับที่น่าพอใจออกมาได้โดยที่ยังคงไว้ซึ่งความพอร์ตเทเบิล และสภาพแวดล้อมที่ปลอดภัย นอกจากนี้ ยังสามารถจะดึงสมรรถนะระดับสูงสุดออกมาใช้ได้ทันทีเมื่อต้องการ

การสร้างและรันภาษาจาวา

การสร้างโปรแกรมจากจาวาก็เหมือนกับภาษาโปรแกรมอื่น ๆ คือเขียนโค้ดโปรแกรมจากโค๊ด ๆ ก็ได้ บันทึกลงในนามสกุล .java จากนั้นจึงนำไปคอมไพล์ด้วยคอมไพเลอร์ของจาวา จะได้ไฟล์ใหม่ในนามสกุล .class ที่เก็บรหัสของคอมไพเลอร์ไว้จาวาเรียกรูปแบบข้อมูลที่อยู่ในไฟล์ใหม่นี้ว่าไบท์โค้ด ไฟล์ที่ได้คือแอฟเพล็คนั้นเอง แอฟเพล็คยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่เราคุ้นเคยกันเพราะข้อมูลแบบไบท์โค้ดจะมีแบบข้อมูลอยู่กึ่งกลางระหว่างโค๊ดโปรแกรมกับโค๊ดที่คอมพิวเตอร์อ่านแล้วนำไปปฏิบัติงานได้ทันที (Machine Code) หากจะรันแอฟเพล็คบนระบบใด ๆ จะต้องใช้อินเตอร์พรีเตอร์จาวาของระบบนั้น ๆ เพื่อตรวจสอบความถูกต้องของไบท์โค้ดแล้วแปลให้เป็นรหัสภาษาเครื่องส่งให้ระบบปฏิบัติการนำไปรันต่อไปขั้นตอนการทำงาน

4.7 โปรแกรมเบราเซอร์ที่รองรับการทำงานของจาวา

HTML (Hypertext Markup Language) เป็นภาษาที่มีรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบนอินเทอร์เน็ตเป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งของคอมพิวเตอร์มีลักษณะของโปรแกรมเป็นไฟล์ตัวอักษรในมาตรฐานของรหัสแอสกี ประกอบด้วย Reserve Markup Directive หรือ คำสั่งต่าง ๆ ซึ่งมีลักษณะจะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้เห็น เชิญไปแจ้งที่ศูนย์บริการ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ง่ายต่อการทำความเข้าใจ เพื่อให้การเว็บเพจมีความสะดวกและง่ายมากยิ่งขึ้น ทั้งนี้เพื่อส่งผ่านโครงสร้างของข้อมูลระหว่างผู้ใช้

ในปัจจุบันภาษา HTML ได้ถูกกำหนดมาตรฐานขึ้นมาเป็นรุ่นที่ 3.0 HyperText ที่ถูกสร้างขึ้นมาจะอยู่ในรูปของแฟ้มเอกสาร HTML (Document File) ที่มีการกำหนดคุณสมบัติ (markup) ของเว็บเพจเข้าไป ยังไม่สามารถแสดงผลข้อมูลออกมาให้ใช้งานได้โดยตรง ถ้าต้องการคุณลักษณะที่เกิดจากการสร้างต้องผ่านโปรแกรมที่ทำหน้าที่แปลคำสั่งนั้นก่อน เราเรียกโปรแกรมที่ทำหน้าที่นี้ว่า “โปรแกรมเบราว์เซอร์” (Browser Program) หน้าที่หลักของโปรแกรมนี้อ็คือเป็นตัวแปลคำสั่งของ HyperText แล้วแสดงผลออกมาเป็นรูปภาพ เสียงข่าวสารและข้อมูล และคุณสมบัติอื่น ๆ ของโปรแกรมเว็บเบราว์เซอร์ ไม่ว่าจะเป็นการดาวน์โหลดไฟล์ การดึงรูปภาพมาใช้งาน การพิมพ์เอกสาร HTML ออกทางเครื่องพิมพ์ การส่งจดหมาย (e-mail) และความสามารถด้านอื่น ๆ อีกมากมาย ทำให้เราสามารถใช้งานอินเทอร์เน็ตหรืออินทราเน็ตได้อย่างสะดวก รวดเร็ว และสนุกสนาน ซึ่งทีมพัฒนาภาษาจาวาได้วางแผนและพัฒนาระบบ Interface โดยเล็งเห็นถึงการเจริญเติบโตและความสามารถของภาษาจาวาจึงได้เปิดตัวเบราว์เซอร์มีชื่อว่า Hotjava เพื่อใช้สำหรับแสดงศักยภาพของจาวาที่มีอยู่อย่างเต็มที่ราวปี 2538 เพราะว่ในขณะนั้นยังไม่มีเบราว์เซอร์ตัวใดที่รู้จักและสามารถทำงานร่วมกับภาษาจาวาได้เลย และแน่นอนว่าเบราว์เซอร์ที่ได้รับการพัฒนาต่อ ๆ มาย่อมจะมีการเพิ่มเติมความสามารถที่ดีกว่า HotJava อย่างแน่นอนคือ Netscape Navigator เป็นเบราว์เซอร์ที่บริษัท Netscape Communications ได้สร้างและเป็นบริษัทแรกที่ได้ลิขสิทธิ์ภาษาจาวาจากบริษัท Sun Web Browser โดย Netscape เวอร์ชันแรกที่ Support คือ Netscape 3.0 เพื่อใช้สำหรับการทำงานบน Windows 3.X, Windows95, WindowsNT, Solaris และ apple Macintosh และยังมีเบราว์เซอร์ของอีกบริษัทหนึ่งที่มีชื่อเสียงไม่แพ้ Netscape คือบริษัท Microsoft ซึ่งได้ออกเบราว์เซอร์ที่รองรับการทำงานของจาวาในชื่อ Internet Explorer 3.0 เพื่อให้เบราว์เซอร์ที่บริษัทพัฒนาเป็นเวอร์ชันล่าสุดในขณะนั้น สามารถรองรับการออกแบบโฮมเพจที่ถูกออกแบบด้วยภาษาจาวา

ปัจจุบันนี้มีบริษัทที่สร้างเบราว์เซอร์อื่น ๆ อีกเช่น AOL (American online) ,IBM ซึ่งแน่นอนว่าปัจจุบันนี้เบราว์เซอร์เหล่านั้นสามารถทำงานกับโฮมเพจในเครือข่ายอินเทอร์เน็ตที่ใช้ภาษาจาวาแน่นอนเพราะปัจจุบันได้มีการนำจาวาไปประยุกต์ใช้งานในด้านต่าง ๆ ทางธุรกิจ การศึกษา บันเทิง อย่างมากมาย

4.8 ความแตกต่างระหว่าง Java Applet และ Java Application

Java Applet กับ Java Application เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อใช้งานบนอินเทอร์เน็ตเหมือนกัน ทำงานอยู่บนโฮมเพจร่วมกับภาษา HTML ที่ใช้สร้าง โฮมเพจเหมือนกัน ลักษณะที่แตกต่างกันก็คือ เอกสารนี้เป็นเอกสารที่ส่งวนเวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Java Applet ไม่สามารถทำงานเดี่ยว ๆ ด้วยตนเองได้จะทำงานร่วมกับไฟล์ .html โดยเราจะเรียกใช้ไฟล์ที่เขียนจาก Java นี้ผ่านทาง Tag คำสั่ง <APPLET>...</APPLET> ภายในไฟล์ HTML จากนั้นก็จะเรียกใช้ไฟล์โปรแกรม javac.exe มาทำการคอมไพล์เพื่อที่จะได้ไฟล์ที่มีนามสกุล .class ไว้สำหรับให้ไฟล์ HTML เรียกผ่านวิธีแสดงผลการทำงานของ Java Applet เราจะใช้โปรแกรม appletviewer.exe หรือเบราว์เซอร์เช่น Netscape, Hotjava, internet Explorer ฯลฯ โดยเลือกใช้ตัวใดตัวหนึ่งก็ได้
- Java Application ตัวที่ 2 นี้มันจะสามารถปฏิบัติงานเดี่ยว ๆ ได้ ด้วยตัวเองคล้ายกับการเขียนโค้ดในโปรแกรมทั่วไป สิ่งที่ต้องจดจำไว้ให้ดีในการเขียน Java Application ก็คือเราจะต้องมีเมธอด ของ Main(string args[]) อย่างน้อย ๆ 1 เมธอดเพื่อใช้เป็นจุดเริ่มต้นของการโปรแกรม อันนี้ถ้าหากสังเกตให้ดีจะเหมือนกับการเขียนโปรแกรมภาษาซีทั่ว ๆ ไป วิธีการคอมไพล์ก็จะใช้โปรแกรม javac.exe เช่นเดียวกับ Java Applet ส่วนการแสดงผลเราจะใช้โปรแกรม Java.exe แทน ซึ่งจะแตกต่างจาก Java Applet ตรงจุดนี้

4.9 ความแตกต่างระหว่าง Java และ JavaScript

สามารถสรุปเป็นตารางแสดงการเปรียบเทียบระหว่าง JavaScript และ Java ได้ดังนี้

ตารางที่ 4.1 ตารางเปรียบเทียบ Java Script และ Java

JavaScript	Java
1. ไม่มีการถูกคอมไพล์ก่อนรันแต่จะรันได้เลย โดยใช้อินเตอร์พรีเตอร์บนไคลเอนท์	1. ถูกคอมไพล์ที่เซิร์ฟเวอร์ก่อนรันบนไคลเอนท์
2. Object-based, รหัสใช้ object ภายในซึ่งมีอยู่แล้วและไม่มีการประกาศคลาสหรือการสืบทอดคลาส	2. Object-oriented Applet มีการสร้าง Object และสืบทอดจากคลาส
3. รหัสรวมอยู่ในเนื้อ Script HTML	3. Applet จะแยกต่างหากจาก Script HTML
4. ไม่มีการประกาศชนิดของตัวแปร	4. ต้องมีการประกาศชนิดของตัวแปร
5. ตรวจสอบความถูกต้องในขณะที่รัน ถ้ามีข้อผิดพลาดเกิดขึ้น อินเตอร์พรีเตอร์จะแจ้งให้ทราบในขณะที่รันไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้	5. ต้องถูกตรวจสอบความถูกต้องก่อนการรันคือ ในขณะที่คอมไพล์ถ้าคอมไพล์ไม่ผ่านไม่สามารถรันได้ไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้

4.10 บทสรุป

ภาษาจาวาเป็นภาษาที่มีการเขียนโปรแกรมเชิงวัตถุซึ่งมีความสามารถสำหรับงานในหลายด้าน โดยเฉพาะงานด้านอินเทอร์เน็ตทั้งในส่วนไคลเอนต์เซิร์ฟเวอร์และส่วนรักษาความปลอดภัยที่จะทำให้โปรแกรมภาษาจาวาสามารถรันได้นั้นจำเป็นต้องอาศัย Java Virtual Machine เพื่อสร้างเป็นคอมพิวเตอร์จำลองที่ทำการคอมไพล์โปรแกรมภาษาจาวาให้สามารถทำงานบนคอมพิวเตอร์เครื่องนั้น ซึ่งถือว่าคุณสมบัติที่สำคัญของภาษาจาวาซึ่งสามารถทำงานได้โดยไม่ขึ้นอยู่กับแพลตฟอร์มและฮาร์ดแวร์ใดๆ



บทที่ 5

จาวากับการติดต่อฐานข้อมูล

5.1 บทนำ

ภาษาจาวาเป็นภาษาคอมพิวเตอร์ที่ได้รับการพัฒนาอย่างรวดเร็วและต่อเนื่องนอกเหนือจากการสร้างกราฟฟิกหรือองค์ประกอบที่ใช้ในการประดับโฮมเพจ (Homepage) ยังมีการพัฒนาองค์ประกอบใหม่ ๆ ในการปฏิบัติงานในด้านต่าง ๆ ได้ถูกเพิ่มประสิทธิภาพให้แก่จาวาคือ JDBC (Java Database Connectivity) ซึ่งทำหน้าที่อำนวยความสะดวกในการเชื่อมต่อและสื่อสารกับระบบฐานข้อมูล โดยพื้นฐานของภาษาจาวาที่ได้รับการออกแบบมาเพื่อปฏิบัติงานกับระบบเครือข่ายคอมพิวเตอร์โดยตรง องค์ประกอบนี้จึงทำให้จาวาเหมาะสมเป็นอย่างยิ่งในการเป็นเครื่องมือสำหรับสร้างโปรแกรมประยุกต์เพื่อเข้าถึงข้อมูลแบบ ไคลต์แอนท์เซิร์ฟเวอร์ โดยบทนี้จะกล่าวถึง JDBC ซึ่งเป็นองค์ประกอบในการเชื่อมในการถ่ายทอดคำสั่งการเรียกค้นหาข้อมูลจาก ไคลต์แอนท์ไปสู่อุปเซิร์ฟเวอร์ และรับข้อมูลผลลัพธ์จากเซิร์ฟเวอร์กลับคืนสู่ไคลต์แอนท์

5.2 ระบบไคลต์แอนท์เซิร์ฟเวอร์

ระบบไคลต์แอนท์เซิร์ฟเวอร์ (Client-Server) ในปัจจุบันมีการเพิ่มขึ้นของค่าใช้จ่ายของบุคลากรที่มาทำหน้าที่ดูแลบำรุงรักษา Database Server ซึ่งได้แก่บุคลากรด้านการจัดการและด้านบริการสนับสนุนสำหรับเครือข่ายขนาดเล็ก (โดยทั่วไปน้อยกว่า 20 ผู้ใช้) โดยปกติผู้จัดการเครือข่ายเพียงคนเดียวสามารถบริหารหน้าที่ดูแลบำรุงรักษา Database Server ควบคุมการเข้าใช้ Database Server จากบรรดาผู้ใช้ทั้งหลาย และให้บริการสนับสนุน Font End เมื่อจำนวนผู้ใช้งานข้อมูลเพิ่มจำนวนขึ้น หรือฐานข้อมูลด้วยตัวของมันเองมีขนาดใหญ่ขึ้น โดยปกติแล้วก็เริ่มจำนวนที่ค่อนข้างผู้จัดการฐานข้อมูลเพิ่มอีกคนเพื่อมาทำหน้าที่รันระบบจัดการฐานข้อมูลรันอยู่บนระบบปฏิบัติการที่บุคลากรด้านบริหารสนับสนุนไม่คุ้นเคยนั้นก็ต้องมีการอบรม ซึ่งเป็นค่าใช้จ่ายที่ต้องเพิ่มเข้าไปในค่าใช้จ่ายเริ่มแรกด้วย

นอกจากนี้แล้วค่าใช้จ่ายด้านฮาร์ดแวร์ก็เพิ่มขึ้นด้วย ในขณะที่มีฐานข้อมูลไคลต์แอนท์เซิร์ฟเวอร์เป็นจำนวนมากที่รันอยู่ภายใต้ระบบปฏิบัติการที่นิยมใช้กันเช่น Netware OS2 และ Unix เป็นต้น และบริษัทผู้ขายส่วนใหญ่ก็ประกาศว่าระบบจัดการฐานข้อมูลของตนเองนั้นสามารถรันอยู่บนฮาร์ดแวร์เดียวกับที่ File Server Software รันอยู่ได้ก็ตาม แต่ว่าโดยปกติแล้วถ้าต้องการสมรรถนะที่สูงและความถูกต้องของข้อมูลที่สูงก็สมควรให้ Database Server แยกไปรันบนคอมพิวเตอร์เฉพาะกิจของ Database เอง สิ่งนี้หมายความว่าต้องทำการซื้อคอมพิวเตอร์เพิ่มอีก

เครื่อง เพื่อมาทำเป็น Database Server โดยเฉพาะคอมพิวเตอร์นี้ต้องมีกำลังและประสิทธิภาพสูง มีแรมเป็นจำนวนมากและมีฮาร์ดดิสก์ขนาดใหญ่ พร้อมทั้งอุปกรณ์สนับสนุนเช่น เครื่องป้องกันไฟฟ้า (UPS) เพื่อป้องกันเซิร์ฟเวอร์เสียหายจากเหตุการณ์ไฟตกหรือไฟดับ

ค่าใช้จ่ายโดยรวมของ Software ของระบบไคล์แอนท์เซิร์ฟเวอร์ ปกติแล้วสูงกว่าระบบจัดการฐานข้อมูลแบบหลายผู้ใช้บนพีซี ค่าใช้จ่ายต่อหนึ่งเซิร์ฟเวอร์สำหรับฐานข้อมูลไคล์แอนท์เซิร์ฟเวอร์ ที่สามารถบริการผู้ใช้ได้แบบไม่จำกัดจำนวน มีราคาสูง บวกกับค่าใช้จ่ายแยกออกไปของ font-end หรือเครื่องมือพัฒนา Software พร้อมทั้งค่าใช้จ่ายบุคลากรสำหรับการฝึกอบรม โปรแกรมเมอร์ในระบบใหม่กล่าวได้ว่าค่าใช้จ่ายของระบบไคล์แอนท์เซิร์ฟเวอร์ สูงกว่าระบบจัดการฐานข้อมูลบนซีพีมาก

ยังมีเรื่องที่น่าขบขาคอีกเรื่องหนึ่งคือ ระบบไคล์แอนท์เซิร์ฟเวอร์ นั้นมีส่วนประกอบต่าง ๆ เป็นจำนวนมากตามหลักที่ไปทีกล่าวไว้ ยิ่งระบบใดมีส่วนประกอบเป็นจำนวนมากโอกาสที่ส่วนประกอบต่าง ๆ ในระบบนั้นจะเสียก็มีมากตามไปด้วย เมื่อเกิดเหตุการณ์ร้ายแรงและระบบเกิดพังเสียหายก็เป็นการยากที่จะชี้ชัดถึงปัญหาที่แท้จริงได้ ระยะเวลาที่ใช้ในการทำให้ทุกสิ่งทุกอย่างกลับสู่สภาพปกติและเริ่มทำงานได้อีกเป็นเวลายาวนาน สิ่งต่าง ๆ ที่เกิดขึ้นเช่นนี้มาจากการขาดประสบการณ์และความชำนาญของบุคลากรด้านบริการสนับสนุนและโปรแกรมเมอร์ ทั้งนี้เนื่องจากไคล์แอนท์เซิร์ฟเวอร์เป็นเทคโนโลยีใหม่ เมื่อเวลาผ่านไปและระบบไคล์แอนท์เซิร์ฟเวอร์ เริ่มเป็นที่นิยมใช้มากขึ้นปัญหาที่ไคล์กล่าวแล้วก็จะลดน้อยลง

องค์ประกอบทางด้าน Hardware และ Software รวมกัน ซึ่งระบบจัดการฐานข้อมูลไคล์แอนท์เซิร์ฟเวอร์ รันอยู่ในแพลตฟอร์มต่าง ๆ เช่น พีซี, Unix Workstation, Super Server, Minicomputer, Mainframe ซึ่งพีซีเป็นแพลตฟอร์มที่นิยมใช้กันมากที่สุดสำหรับไคล์แอนท์เซิร์ฟเวอร์

บรรดาระบบฮาร์ดแวร์ต่าง ๆ ทั้งหมดก็มีความแตกต่างทั้งในเรื่องคุณลักษณะและความสามารถอย่างไรก็ตามก็มีคุณลักษณะร่วมที่จำเป็นสำหรับระบบปฏิบัติการ โดยปกติแล้ว Application ถูกเขียนให้รันภายใต้ระบบปฏิบัติการใดระบบปฏิบัติการหนึ่งโดยเฉพาะ ซอฟต์แวร์ระบบปฏิบัติการที่นิยมใช้กัน เช่น MS-Dos, PC-Dos, OS/2, Unix, VMS ของ Digital, MVS/XA ที่รันบน IBM Mainframe เป็นต้น

คุณลักษณะเบื้องต้นของระบบปฏิบัติการที่จำเป็นสำหรับระบบจัดการฐานข้อมูลไคล์แอนท์เซิร์ฟเวอร์ คือ ความสามารถในการทำ Multitasking หรือการรัน Application หลาย Application พร้อมกันของบรรดาผู้ใช้ทั้งหลายได้อย่างถูกต้อง โดยปราศจากการรบกวนซึ่งกันและกัน multitasking ทำงานโดยแบ่งเวลาประมวลผลของซีพียูออกเป็นช่วง ๆ (time Slices) ให้แก่บรรดา Task ต่าง ๆ และบรรดา Process ต่าง ๆ ระบบปฏิบัติการ Multitasking ทั้งที่เป็นแบบ Preemptive และแบบ Non-Preemptive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้บางระบบปฏิบัติการยังสามารถเป็นระบบ Multi-user ได้ (คือยอมให้มีผู้ใช้หลายคนทำงานที่แตกต่างกันได้พร้อมกัน) โดยเฉพาะเมื่อมีการใช้เทอร์มินอลธรรมดาทั้งหลายเป็นทางเข้าสู่ฐานข้อมูลโอเอสแบบ multi-user ไม่ได้ ก่อให้เกิดข้อดีหรือข้อเสียโดยเฉพาะเมื่อถูกใช้เป็นแพลตฟอร์มสำหรับระบบจัดการฐานข้อมูลไคล์แอนท์เซิร์ฟเวอร์

การเพิ่มความสามารถด้าน Multithread เข้าไปในซอฟต์แวร์โอเอส Multitask แบบ Preemptive ความสามารถ Multithread ทำให้ Application ทำ Multitask ได้ภายในตัวเองได้ ตัวอย่างเช่น ระบบจัดการฐานข้อมูลผู้ใช้คนเดียวแบบ Multithread สามารถทำการ Start ใหม่ (Process หรือ Task) ให้ทำงานออกภายนอกที่ซับซ้อนโดยทำอยู่ใน Background ในขณะที่ผู้ใช้กำลังทำการสอบถาม ซึ่งทำอยู่ใน foreground Multithread เป็นปัจจัยที่สำคัญสำหรับการออกแบบระบบจัดการข้อมูลไคล์แอนท์เซิร์ฟเวอร์ ที่ซับซ้อน Multithread ทำให้ Application สามารถควบคุมให้ Task ใหม่เริ่มทำงานหรือหยุดทำงานได้ Application สามารถถูกออกแบบให้มีความชาญฉลาดภายในที่ตัดสินใจได้ว่า Process ใด หรือ Task ใดมีลำดับความสำคัญและควร ได้รับเวลา ซีพียูมาก ตัวอย่างเช่น ระบบจัดการฐานข้อมูลมีลำดับความสำคัญสูงกว่าการสอบถามข้อมูล เป็นต้น

เนื่องจากจุดด้อยในเรื่องการ ไม่มีแพลตฟอร์มใดที่สามารถสนองความต้องการได้ทุกประการหรือเหมาะสมในทุกสถานการณ์ในระบบไคล์แอนท์เซิร์ฟเวอร์ ในปัจจุบันมีการประยุกต์ใช้ระบบที่เรียกว่า “Network Computer” เพื่อช่วยแก้ปัญหาของระบบไคล์แอนท์เซิร์ฟเวอร์ ในเรื่องการใช้บุคลากรในการดูแลระบบน้อยลงคือจะไม่ต้องไปดูแลในส่วนของไคล์แอนท์ และจากปัญหาการทำงานไม่ครบทุกแพลตฟอร์ม ในปัจจุบันมีภาษาคอมพิวเตอร์ที่สามารถเขียนโปรแกรมคอมพิวเตอร์โดยไม่คำนึงถึงแพลตฟอร์มประกอบกับความสามารถทางด้านระบบ Network Computing ที่ภาษานี้สามารถนำมาประยุกต์ใช้ได้อย่างดี ภาษาดังกล่าวคือ ภาษาจาวา

5.3 การเชื่อมต่อฐานข้อมูลด้วย ODBC

Open Database Connectivity คือวิธีการติดต่อและเข้าถึงจากแอปพลิเคชันสู่ระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL เป็นมาตรฐานการเข้าถึงฐานข้อมูล ความสามารถในการต่อเชื่อมแบบนี้ทำให้แอปพลิเคชันสามารถเข้าถึงฐานข้อมูลได้หลายรูปแบบ ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนาโปรแกรมไปได้โดยไม่ต้องทำการระบุชนิดของระบบจัดการฐานข้อมูล

แต่เดิมนั้นการพัฒนาโปรแกรมประยุกต์ที่ใช้งานเกี่ยวกับฐานข้อมูล การเข้าใช้ฐานข้อมูลของโปรแกรมเหล่านี้จะทำการเรียกใช้ SQL แบบฝังตัว (Embedded SQL) ซึ่งในขณะนั้นวิถีทางแบบนี้ก็จะดูไปได้ดีทีเดียว เพราะว่าตัวโปรแกรมสามารถทำการเปลี่ยนรูปแบบของระบบไม่ว่าจะเป็นทางด้านฮาร์ดแวร์หรือซอฟต์แวร์ ได้หลายรูปแบบ รวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มีการย้ายระบบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่มีความแตกต่างกัน เช่นการเรียกใช้ข้อมูลของออราเคิลจากไมโครซอฟท์เอ็กเซล (Microsoft Excel) วิธีการเข้าถึงข้อมูลแบบเดิมนั้นจะต้องทำการปรับคอมไพล์โค้ดของเอ็กเซลและออราเคิล โดยใช้ IBM precompiler และ Oracle precompiler ตามลำดับ ซึ่งจะเห็นว่าเป็นการยุ่งยากมากทีเดียว

วิธีการเชื่อมต่อแบบ ODBC จะให้ความสะดวกในการติดต่อข้อมูลมากกว่าวิธีการดั้งเดิม โดยการกำหนดมาตรฐานการเชื่อมต่อข้อมูล (Data protocol, DBMS capability) และแนวทางนี้ได้ทำให้เกิดความคิดที่จะสร้างไคร์ฟเวอร์สำหรับการติดต่อกับของฐานข้อมูลขึ้นมา (DLL)

5.4 ทฤษฎีการทำงานของ ODBC

ส่วนติดต่อของการเชื่อมต่อฐานข้อมูลเปิด (ODBC : Open Database Connectivity) เป็นตัวทำให้โปรแกรมประยุกต์สามารถที่จะเข้าถึงข้อมูลระบบจัดการฐานข้อมูล (DBMS : Data Management System) โดยใช้ภาษา SQL (Structure Query Language) เป็นมาตรฐานหลักในการทำงาน

5.4.1 ข้อดีของการติดต่อโดยใช้ ODBC

1. ฟังก์ชันของ ODBC อนุญาตให้แอปพลิเคชันติดต่อกับระบบจัดการฐานข้อมูลได้โดยสะดวก (การทำคำสั่ง SQL และการรับผลลัพธ์)
2. ใช้ภาษา SQL ตามมาตรฐาน SQL CAE, X/Open และ SQL Access Group (SAGX)
3. มีการกำหนดการส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตรฐานเดียวกัน
4. เป็นวิธีการมาตรฐานในการติดต่อกับระบบการฐานข้อมูล
5. มีการกำหนดชนิดข้อมูล (Data Type) เป็นมาตรฐาน
6. ชุดคำสั่ง SQL สามารถกำหนดได้แม้ในขณะที่ Runtime
7. สามารถเขียนโปรแกรมชุดเดียว แต่สามารถเข้าใช้ระบบจัดการฐานข้อมูลได้หลายตัว
8. ตัวโปรแกรมไม่ต้องรับผิดชอบในการดูแลการติดต่อกับระบบจัดการฐานข้อมูล
9. ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

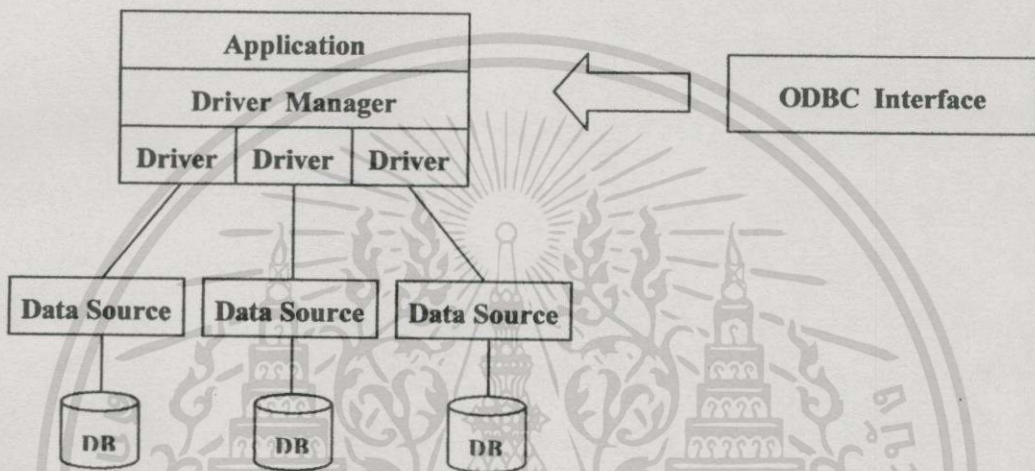
5.4.2 องค์ประกอบของ ODBC

สถาปัตยกรรมของ ODBC ประกอบด้วย 4 ส่วนสำคัญ

1. แอปพลิเคชัน ทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของ ODBC ตามคำสั่งภาษา SQL พร้อมทั้งทำการรับผลลัพธ์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตัวจัดการไครฟ์เวอร์ (Driver Manager) ทำหน้าที่โหลดไครฟ์เวอร์เชื่อมต่อกับแหล่งข้อมูล
3. ไครฟ์เวอร์ ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของ ODBC ส่งคำสั่ง SQL ไปสู่แหล่งข้อมูลที่ต้องการ และทำการแปลงคำสั่งมาให้อยู่ในรูปแบบที่สนับสนุนโดยระบบจัดการฐานข้อมูลแต่ละชนิดอีกด้วย
4. แหล่งข้อมูล (Data Source) เป็นแหล่งข้อมูลที่ใช้ต้องการเข้าถึง



รูปที่ 5.1 องค์ประกอบของ ODBC

5.4.2.1 Application

ตัวโปรแกรมจะเรียกใช้การต่อเชื่อม ODBC ในการทำงานต่อไปนี้

1. ร้องขอการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำสั่ง SQL สู่แหล่งข้อมูล
3. กำหนดพื้นที่การจับเก็บและรูปแบบของข้อมูลที่เป็นผลลัพธ์จาก SQL Request
4. ร้องขอผลลัพธ์
5. ประมวลผลและจัดการกับข้อผิดพลาด
6. รายงานผลให้ผู้ใช้
7. ร้องขอการ Commit หรือ Rollback สำหรับควบคุมการประมวลผล Transaction

5.4.2.2 ไครฟ์เวอร์

ไครฟ์เวอร์คือ DLL ที่สร้างฟังก์ชันของ ODBC และทำการโต้ตอบกับแหล่งข้อมูล ไครฟ์เวอร์ทำการตอบสนองการเรียกใช้ฟังก์ชันของ ODBC โดยจะทำงานต่อไปนี้

1. สร้างการต่อเชื่อมกับแหล่งข้อมูล
- เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่งคำขอร้องให้กับแหล่งข้อมูล
3. แปลงข้อมูลจากรูปแบบหนึ่งสู่อีกรูปแบบหนึ่ง
4. ส่งผลลัพธ์กลับให้แอปพลิเคชัน
5. จัดการส่งข้อมูลความผิดพลาดให้อยู่ในรูปแบบรหัสมาตรฐานแล้วส่งกลับไปให้แอปพลิเคชัน
6. ทำหน้าที่จัดการและดูแลเคอร์เซอร์ (Cursor)

5.5 การเชื่อมต่อฐานข้อมูลด้วย JDBC

JDBC (Java Database Connectivity) ถูกพัฒนาโดย Java Soft Department ของบริษัท Sun Micro Systems ซึ่งก็คือฟังก์ชันมาตรฐานหรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล นักพัฒนาสามารถใช้ JDBC API และยังสามารถใช้ packages อื่น ๆ ด้วย ซึ่งนำเสนอในรูปแบบ function พิเศษหรือ higher level API เพื่อเขียนโปรแกรมอิสระในการกำหนด DBMS หรือ database connectivity mechanism ในการเริ่มใช้งาน โดยทั่วไปการใช้ SQL database ในการติดต่อ (Interface) กับ framework เพื่อที่จะจัดการฐานในการติดต่อในส่วนบนสุดของชนิดต่าง ๆ ของ database connectivity modules ซึ่งก็คือ มาตรฐานของ ANSI SQL-2 Entry level database. เพราะว่า relational database เกือบจะทั้งหมดในปัจจุบันใช้มาตรฐาน SQL-2 Entry level

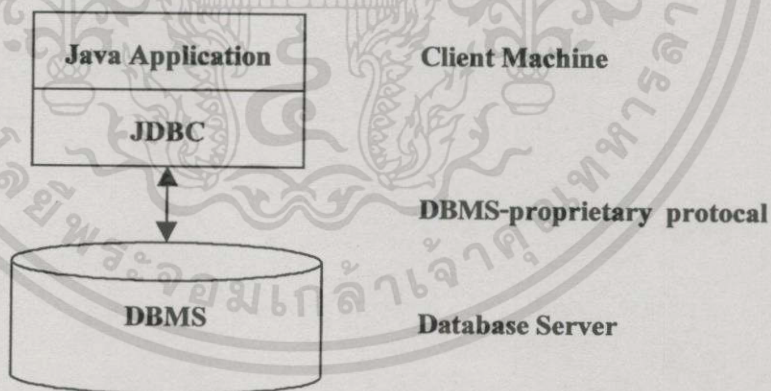
JDBC สร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC Database Connectivity ของบริษัท Microsoft ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN) แต่ตามหลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบ X-Windows และ JDBC driver ต้องเข้ากันได้กับระบบมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่าน Conformance test เป็นผู้กำหนดขึ้น

5.6 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-tier model และ Three-tier model

5.6.1 Two-Tier model

จากแอปเพล็ต (Applet) หรือจาวาแอปพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรงจึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC ไดรฟ์เวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 5.2 คำสั่งในการเรียกค้นข้อมูลในรูปของภาษา SQL (Structured Query Language) จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้งานข้อมูลดังกล่าวนี้ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้สามารถเชื่อมต่อ ผ่านระบบเน็ตเวิร์ค (Network) ระบบแบบ Two Tier นี้ใช้หลักการทำงานเช่นเดียวกับรูปแบบไคลเอนต์/เซิร์ฟเวอร์ ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคลเอนต์ และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two-Tier มักจะเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

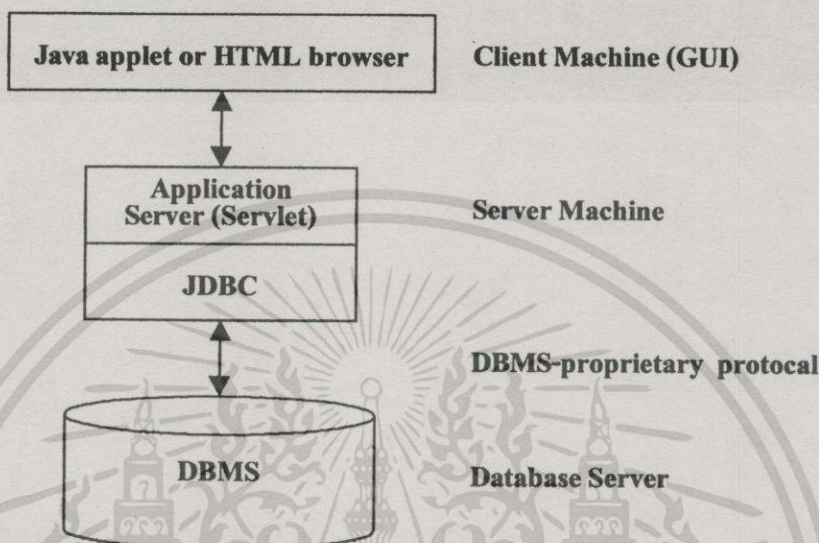


รูปที่ 5.2 Two-Tier model

5.6.2 Three-Tier model

คำสั่งเรียกค้นต่าง ๆ จากผู้ใช้จะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle tier และส่งต่อไปให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบเอกสารเป็นเอกสารที่ส่งวนเวียนสำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นาเบไซบระโฮงนดานการคำไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

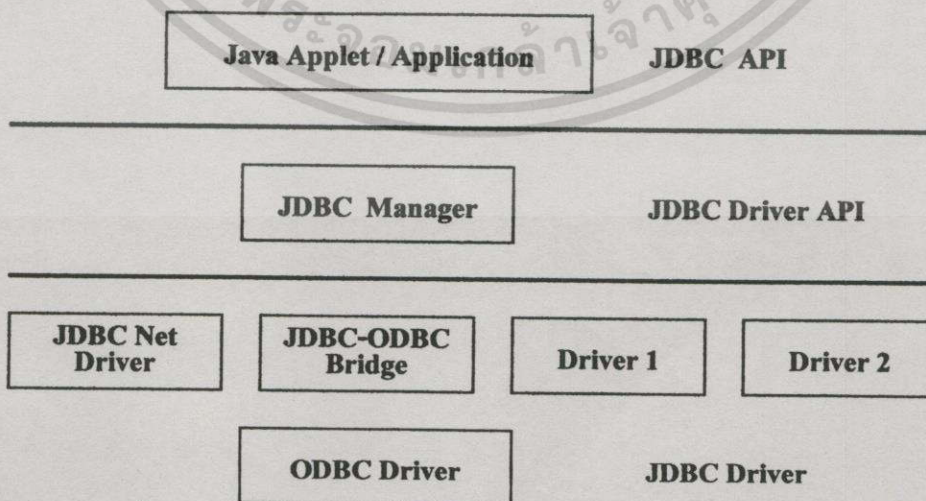
ด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ (Web-Server) ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอนท์ และเซิร์ฟเวอร์สามารถพูดคุยกันได้ การปรับเปลี่ยนระบบคอมพิวเตอร์ของไคลเอนท์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น



รูปที่ 5.3 Three-Tier model

5.7 โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 5.4 ระดับบนสุด JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง JDBC Driver (มีไคลร์ฟเวอร์ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม



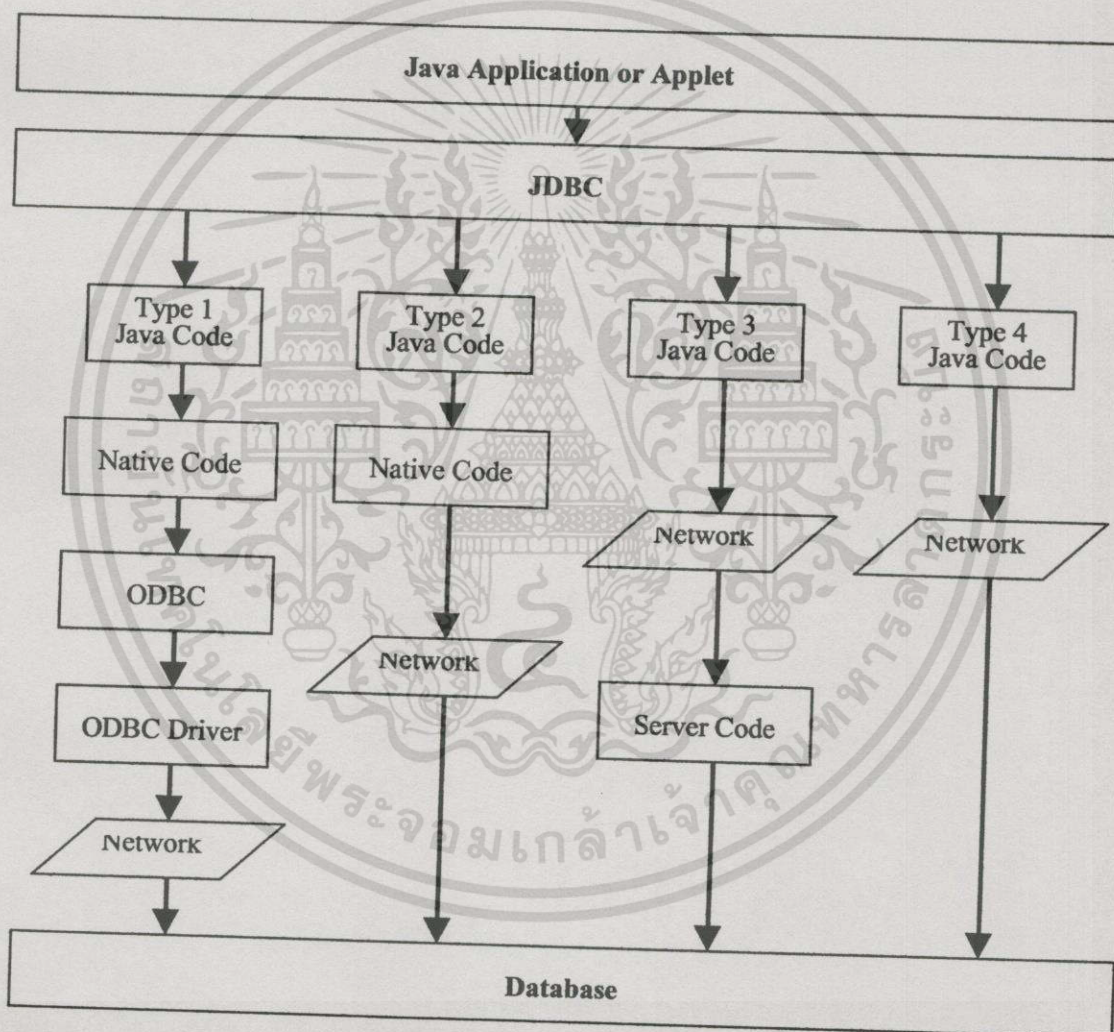
รูปที่ 5.4 ระดับการเชื่อมต่อของ JDBC API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.8 รูปแบบของ JDBC ไดรฟ์เวอร์

JDBC ไดรฟ์เวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่าง ๆ ของ JDBC ถูกจัดแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

1. JDBC-ODBC Bridge
2. Native-API (Party-Java) Drivers
3. Net-Protocol (All-Java) Driver
4. Native-protocol (All-Java) Drivers



รูปที่ 5.5 แสดงรูปแบบของ JDBC

5.8.1 JDBC/ODBC bridge

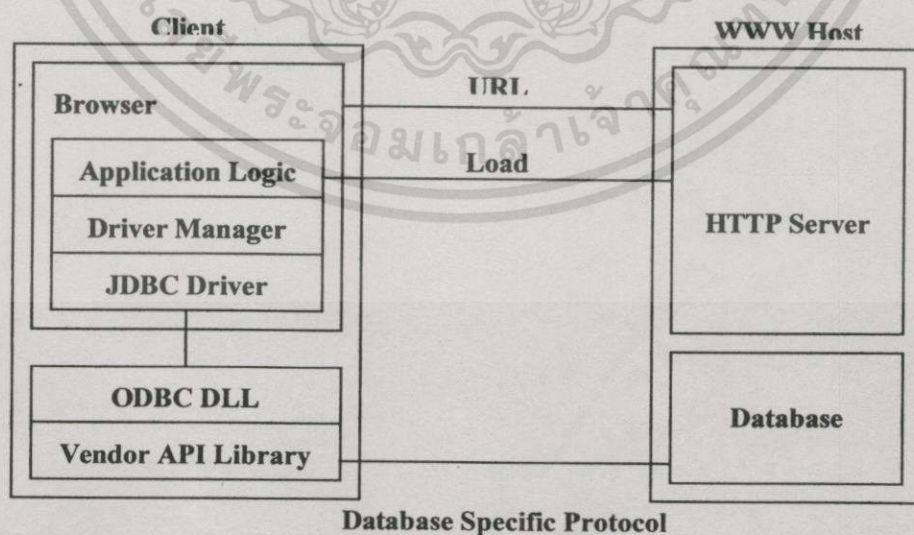
JDBC/ODBC bridge ถูกพัฒนาโดย Java Soft และ Intersolv ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลได้โดยผ่านการทำงานของ JDBC โดยนำข้อดีของ ODBC-enable data source ที่มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกต่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อยู่โดยทั่วไปอย่างมากมาย ฟังก์ชัน Java applet หรือ application จะถูกเขียนโดยใช้ JDBC API Bridge จะทำการแปลงการเรียกใช้ JDBC ไปยัง ODBC และส่งค่า ODBC driver ที่เหมาะสมสำหรับ back-end database

ข้อดีของ bridge ทำให้ application สามารถติดต่อกับ database ได้อย่างง่ายดายโดยจาก vendors ที่มากมายโดยคัดเลือก ODBC driver ที่เหมาะสม อย่างไรก็ตามการติดต่อ database ประเภทนี้ต้องพิจารณาค่าใช้จ่าย (Overhead) และความซับซ้อน (Complexity) เพราะว่าการเรียกใช้จะมีลำดับการเรียกใช้ดังนี้คือ

JDBC ↔ bridge ↔ ODBC ↔ Native client-API ↔ database

การติดต่อโดยใช้ประเภทนี้ชนิดของ Driver จะไม่อนุญาตสำหรับการย้ายข้อมูลแบบ just-in-time ของ Java applet. Native code ต้องมีการเตรียมการติดตั้งบนเครื่องไคลเอนต์ โดยตรง เฉพาะว่า JDBC-ODBC Bridge เป็นไคร์ฟเวอร์ที่ถูกเขียนขึ้นมาจากภาษา c/c++ ในบางส่วนจึงทำให้ไม่สามารถดาวน์โหลด (Download) ข้ามเครือข่ายอินเทอร์เน็ตและแปล (Interpret) บนเบราว์เซอร์ (Webbrowser) ได้เพราะภาษา c/c++ ต้องรับเฉพาะแต่ละแบบเท่านั้น ดังนั้น ODBC ไคร์ฟเวอร์ในรูปแบบของไลบรารีไฟล์หรือ DLL (Dynamic Link Library) และ JDBC-ODBC Bridge ไคร์ฟเวอร์ต้องได้รับการติดตั้ง Software โดย Administrative รูปแบบเหมือนไคลเอนต์เซิร์ฟเวอร์แอปพลิเคชัน ดังนั้น JDBC/ODBC Bridge จะไม่สามารถแก้ไขปัญหาของไคลเอนต์ Program ดังกล่าวได้จึงเหมาะสมสำหรับการใช้งานกับระบบ Three-tier โดยไคร์ฟเวอร์ทั้งสองได้รับการติดตั้งอยู่ที่ Middle-Tier หรือเครื่องคอมพิวเตอร์ตัวกลางเช่น Web-Server เป็นต้น



รูปที่ 5.6 การติดต่อ JDBC โดยรูปแบบที่ 1 JDBC-ODBC Bridge

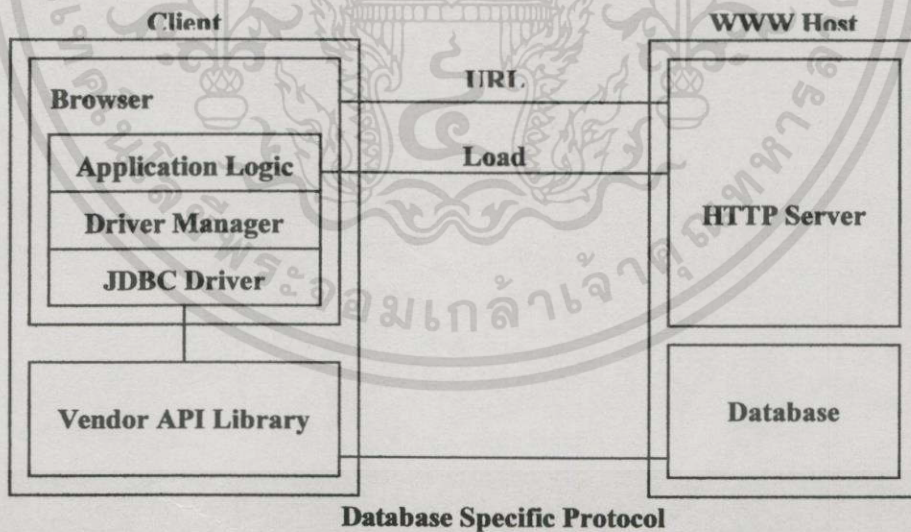
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.8.2 Native-API , partly Java driver

Native-API, party Java driver เป็น two-tier ที่ซึ่ง JDBC ต้องการ Vendor Library ในการแปลง JDBC function ไปยังคุณลักษณะของการใช้ภาษาในการ Query เรียกใช้ DBMS ยกตัวอย่างเช่น library สำหรับ Sybase คือ dlib, สำหรับ Oracle คือ ocilib ซึ่ง driver เหล่านี้จะถูกใช้เพื่อเขียนในการรวมของภาษา Java และ C/C++ การติดต่อโดย JDBC ประเภทนี้จะเหมือนกับ JDBC/ODBC Bridge คือ ต้องการ code ซึ่งก็คือ vendor library เพื่อที่จะใช้ในการติดตั้งบนเครื่องไคลเอนท์ ดังนั้นต้องมีการบำรุงรักษา (maintenance) ปัญหาเหมือนกับ bridge อย่างไรก็ตาม Driver ประเภทนี้จะเร็วกว่าประเภทแรกเพราะว่ามี Layer พิเศษของการแปลงเป็น ODBC ถูกกำจัดออกไป

JDBC ↔ Native client-API ↔ database

ข้อดีของไดรฟ์เวอร์ชนิดนี้คือ สามารถใช้ Native-API Driver ติดต่อกับฐานข้อมูลนั้น ๆ โดยตรงโดยผ่านโปรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะสมกับการเชื่อมต่อกับฐานข้อมูลแบบ Two-Tier Client/Server แต่ก็มีข้อเสียที่คล้ายคลึงกับ JDBC-ODBC Bridge คือต้องติดตั้งไดรฟ์เวอร์ (DLL) และโปรโตคอลไดรฟ์เวอร์เช่น SQL-NET, INET, Open Client ฯลฯ บนเครื่องคอมพิวเตอร์นั้น ๆ ที่สำคัญส่วนหนึ่งของ Native-API Driver ถูกเขียนจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้และไม่สามารถเชื่อมต่อฐานข้อมูลข้ามชนิดกันได้



รูปที่ 5.7 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API, partly Java driver

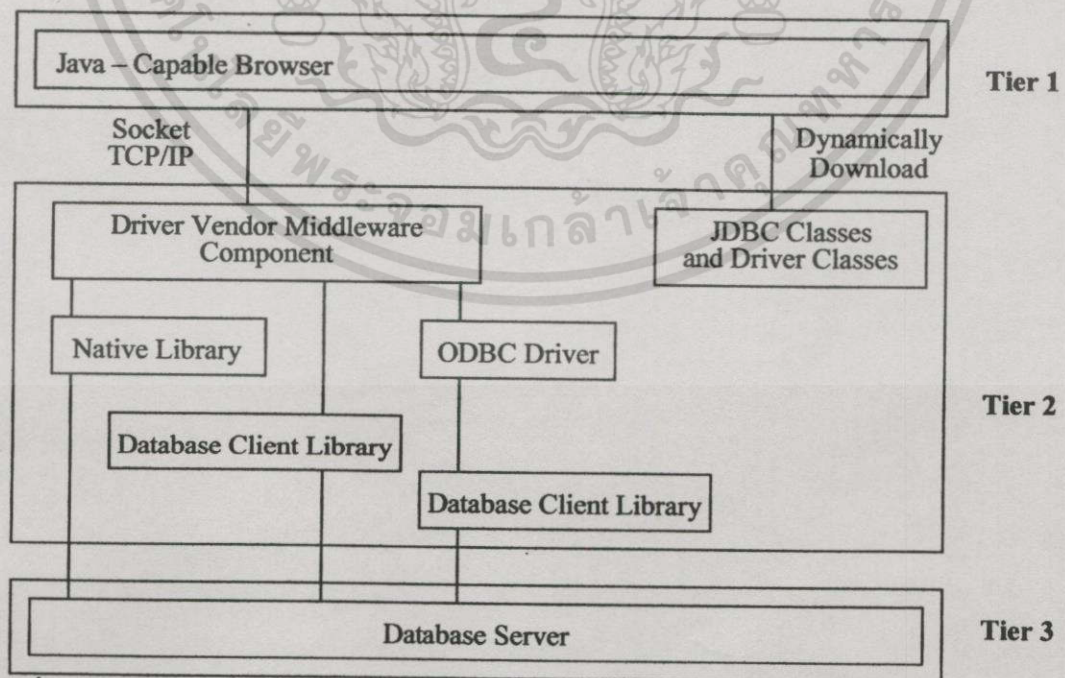
5.8.3 Network-protocol, all-Java driver

Network-protocol, all-Java ประเภทนี้ driver จะทำการแปลงการเรียกใช้ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์คโปรโตคอลร่วม (DBMS-independent network Protocol) ซึ่งหลังจากนั้นจะถูก
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนเพื่อใช้ภายในเท่านั้น เมื่อผู้ใดเห็นนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงให้อยู่ในรูปเฉพาะของแต่ละฐานข้อมูล (database-specific API) บนเซิร์ฟเวอร์นั้น Middle-tier server รูปแบบการเชื่อมต่อฐานข้อมูลจะเป็นลักษณะ three tier : JDBC driver (ปรกติ 200 KB หรือน้อยกว่านั้น) ทำการ execute บนไคล์แอนท์ และถูกใช้เครื่องมือทาง logic สั่งคำสั่ง SQL ไปยัง Network และไปยัง JDBC เซิร์ฟเวอร์ เมื่อได้รับข้อมูลกลับมาจากเซิร์ฟเวอร์ และถูกจัดการ connection Driver ประเภทนี้อนุญาตสำหรับการใช้ Just-in-time บนเครื่องไคล์แอนท์

JDBC เซิร์ฟเวอร์จะจัดการ connection ที่มีมาไปยัง database ซึ่งดีเหมือน exception และ status events resulting จาก SQL execution. Packages ของ data สำหรับการติดต่อข้าม network ไปยัง JDBC ไคล์แอนท์

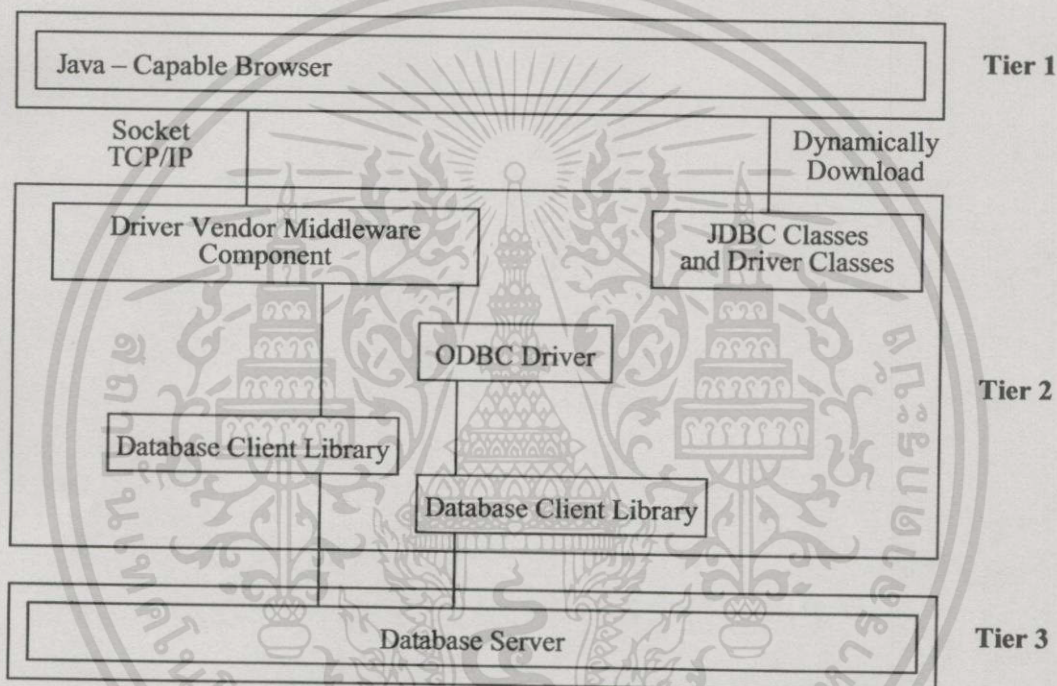
Middleware server component สามารถอธิบายเหมือน native component หรือถูกเขียนด้วย Java การอธิบาย native ในการ connect ไปยัง database server โดยใช้ library ของ vendor หรือ ODBC. Symantec's dbAnywhere และ Intersolv's SequeLink ทั้งคู่จัดให้อยู่ในประเภทนี้ SequeLink ไม่ต้องการ database client libraries ที่จะใช้ในการติดตั้งลงบนเซิร์ฟเวอร์ แต่จะใช้ Libraries ของตนเอง เซิร์ฟเวอร์ ต้องมีการติดตั้ง Configuration สำหรับ database ที่จะใช้ติดต่อเช่น ติดตั้งหมายเลขของ Port ตัวแปรของ database specific environment สำหรับ instance, DSQuery กับ Sybase, Database specific parameters เช่น logging, translation และ parameter อื่น ๆ ที่ซึ่งเซิร์ฟเวอร์ อาจจะต้องการ ถ้ามี Middleware server ถูกเขียนด้วยภาษา Java สามารถใช้ JDBC-compliant driver ในการติดต่อกับ DBMS โดย Database Proprietary Protocol ของ vendor ไม่ต้องการที่จะ configure ที่เซิร์ฟเวอร์ สำหรับ database



รูปที่ 5.8 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver

เอกสารนี้เป็นเอกสารตัวอย่างของ dbAnywhere, SequeLink ศึกษาดูเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับคุณสมบัติเหมือนการติดต่อกับ Caching จำนวนของ listener threads และอื่น ๆ ความต้องการค่าคงที่ที่ต้องการติดตั้ง Weblogic's T3 server ถูกจัดให้อยู่ในประเภทนี้ JDBC T3 driver จะถูก executing บนไคล์แอนท์ โดยจะนำชื่อของ driver ใช้ในการติดต่อสื่อสารกับ DBMS ซึ่งดีเหมือนกับคุณสมบัติ ที่ต้องการ โดย driver เช่น username, password, database name และอื่น ๆ ดังนั้นการส่งข้อมูลไปยัง T3 server จะ Load ผ่านไปยัง driver class และใช้ในการติดต่อกับ database Server ยังต้องการ load คิว database vendor libraries และ/หรือ ODBC driver ในกรณีการส่งค่า driver class name ในประเภทที่ 1 หรือประเภทที่ 2



รูปที่ 5.9 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver ยกตัวอย่างของ WebLogic

Driver ประเภทนี้ถือว่าเหมาะสมมากที่สุดสำหรับ Internet/intranet -based, multi-user data-intensive application ที่ซึ่งมีจำนวนของ concurrent data operations มาก ๆ เพราะมีความยืดหยุ่นคล่องตัวที่สุด เพราะว่าเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไคร์ฟเวอร์ร่วมที่เขียนขึ้นจากภาษาอื่นซึ่งต้องติดตั้งเฉพาะบนไคล์แอนท์นั้น และสามารถรันระบบไคล์ก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine) การปฏิบัติงานทุกรูปแบบทั้งความโหลดข้ามเครือข่ายหรือรันบนเครื่องเดี่ยวๆ สามารถกระทำได้หมดเช่นการ queries searches และอื่น ๆ และยังมีความสามารถทางด้าน scalability และประสิทธิภาพซึ่งถือเป็นส่วนประกอบหลักเซิร์ฟเวอร์สามารถจัดการ multiplexing management ระหว่าง database ที่หลากหลาย สามารถจัดหา logging และ administration facilities สามารถ load balancing features และยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาตเห็นาไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

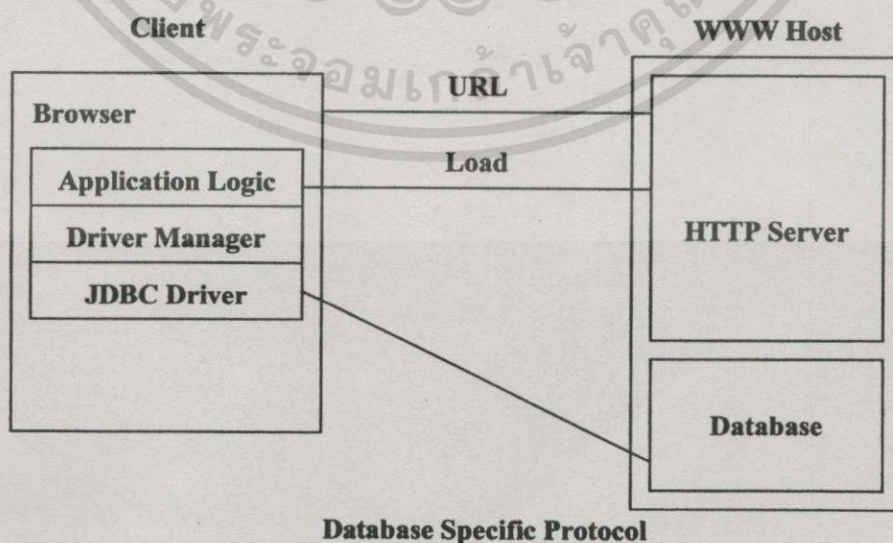
สามารถรองรับ catalog และการ Query caches. Three-tier web database application ส่วนใหญ่มี security, firewalls และ proxies

ข้อดีของ network-centric drivers คือองค์ประกอบของเซิร์ฟเวอร์ คือ middleware ตัวแทนจำหน่ายแต่ละรายใช้ middleware ของตนเองสำหรับการติดต่อสื่อสารข้ามผ่าน network ดังนั้น enterprise users จึงมีความสามารถที่จะจัดการได้โดย CORBA products และการ upgrade paths ในอนาคต

Openjdbc จาก I-Kinetics คือ Network-centric driver ที่ใช้ CORBA คือ middleware จะเชื่อม JDBC ไคล์แอนท์และเซิร์ฟเวอร์ เครื่องมือชนิดนี้มีข้อดีของ CORBA's distributed computing infrastructure คือวงการอุตสาหกรรมยอมรับเป็นมาตรฐานในการเปิดที่จะติดต่อไปยังการสื่อสารอื่น ๆ ไคล์แอนท์มาเป็น assembly ของ cooperating objects ที่ค่อนข้างจะมากกว่าหนึ่งเซิร์ฟเวอร์

5.8.4 Native-protocol, all-Java driver

Native-protocol, all-Java driver จะแสดงการเรียกใช้คำสั่งของ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์ค โปรโตคอลเฉพาะของฐานข้อมูลนั้น โดยตรงไป โดยคุณสมบัติของตัวแทนจำหน่าย database Driver เหล่านี้สามารถเขียนในภาษา Java และสามารถติดต่อกับ applet แบบ just-in-time เพราะว่า drivers เหล่านี้จะแปลง JDBC ตรงไปยัง native protocol โดยปราศจากการใช้ ODBC หรือ Native APIs ซึ่งสามารถจัดหามาสำหรับการติดต่อ database ที่มีประสิทธิภาพสูง Driver เหล่านี้สร้างจากตัวแทนจำหน่าย DBMS เท่านั้นเนื่องจากความจริงที่ความรู้ของ Protocol lies กับตัวแทนจำหน่าย การนำวิธีติดต่อประเภทนี้มาใช้งานมีอยู่เพียงเล็กน้อยในปัจจุบัน แต่ในอนาคตคาดว่าจะเพิ่มจำนวนขึ้นเรื่อย ๆ



รูปที่ 5.10 การติดต่อ JDBC โดยที่รูปแบบที่ 4 Native-protocol, all-Java driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของไครฟ์ชนิดนี้คือไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละเครื่อง และไม่มีความจำเป็นที่ต้องติดตั้งไครฟ์เวอร์ตัวกลาง เช่น ODBC บนไคล์แอนท์แต่ละเครื่องจึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลไม่สามารถทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไครฟ์เวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้น ๆ

JDBC ไครฟ์เวอร์ชนิดที่ 3 และ 4 คือไครฟ์เวอร์ที่คาดว่าจะจะเป็นสิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องมากกว่า เช่นมีการติดตั้งไครฟ์เวอร์ เพื่อพัฒนาขึ้นมาเพื่อใช้งานร่วมกับไครฟ์เวอร์ดั้งเดิม (ODBC) หรือเน็ตเวิร์คโปรโตคอลเฉพาะฐานข้อมูล ที่มีใช้อยู่แล้วในปัจจุบันไปพลาง ๆ ก่อน ซึ่งแน่นอนว่าไครฟ์เวอร์ชนิดที่ 1 และ 2 ชนิดนี้มีข้อจำกัดในการใช้งานมากกว่า 2 ชนิดแรก โดยเฉพาะเรื่องความปลอดภัยและ Bridge ร่วมกับ ODBC ไครฟ์เวอร์บน Window 95/NT ผลปรากฏว่าไม่ยุ่งยากและให้ประสิทธิภาพดีพอสมควร เหมาะสมในการเป็นเครื่องมือในการเชื่อมต่อกับฐานข้อมูล จนตั้งแต่ขั้นพื้นฐานจนถึงขั้นคอนโซลปฏิบัติงานได้จริง

5.9 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล

1. เปิดการเชื่อมต่อกับฐานข้อมูล
2. ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
3. จัดการกับผลลัพธ์ที่ได้รับ
4. ปิดการเชื่อมต่อ

และเพื่อให้เข้าใจการทำงานของ JDBC ยิ่งขึ้นตัวอย่างโปรแกรมบางส่วนต่อไปนี้จะแสดงให้เห็นถึงขั้นตอนทั้ง 4 ในการติดต่อกับฐานข้อมูลผ่าน ODBC ไครฟ์เวอร์โดยใช้ JDBC-ODBC Bridge (ฟังก์ชัน API ต่าง ๆ ของ JDBC ที่มาพร้อมกับ JDK เวอร์ชัน 1.1 ขึ้นไปเก็บรวบรวมไว้ในแพ็คเกจ `java.sql.*`) เปิดการเชื่อมต่อกับฐานข้อมูล

จากตัวอย่างสิ่งแรกที่ทำคือเลือกชนิดของ JDBC ไครฟ์เวอร์ (JDBC-ODBC Bridge) ระบุชื่อฐานข้อมูล (DSN : Data Source Name) รวมถึงชื่อผู้ใช้ (Login) และรหัสผ่าน (Password) จากนั้นเป็นการสร้างอ็อบเจกต์ `stmt` (คลาส Statement) สำหรับการเรียกค้นหาข้อมูล และจัดส่งคำสั่ง SQL ไปในรูปแบบสตริง (String) ผ่านเมธอด `executeQuery()` ผลลัพธ์ที่ได้จากเมธอดนี้คือผลลัพธ์จากการประมวลผลของฐานข้อมูลในชนิดข้อมูลที่จาวารู้จัก เช่น `int`, `String`, `float`, ฯลฯ) เพื่อนำไปใช้ประโยชน์อื่นต่อไป สุดท้ายคือการปิดการเชื่อมต่อกับฐานข้อมูล

ภาษาจาวาและ JDBC คือเครื่องมือที่จะทำให้การเชื่อมต่อสื่อสารกับระบบฐานข้อมูลไปอย่างสะดวกรวดเร็วและเป็นหนึ่งเดียว (Database-neutral Communication) จากชนิดของ JDBC Drivers สามารถแสดงตัวแทนจำหน่ายและฐานข้อมูลที่สามารถปฏิบัติการแสดงดังตารางได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 เปรียบเทียบ JDBC Driver

Vendor	Type	DBMS's supported
Agave Software Design	3	Oracle, Sybase, Informix, others via ODBC
Asgard Software	3	Unisys A series DMSII database
Borland	4	InterBase 4.0
Caribou Lake software	3	Ingres
Connect Software	4	Sybase, MS SQL Server
DataRamp	3	Several dozen though ODBC drivers.*
Ensodex, Inc.	3	Several through ODBC.*
IBM	2,3	IBM DB2 Version 2
IBM	4	DB2 for OS/400
GWE Technologies	4	Mysql
IDS Software	3	Oracle, Sybase, MS SQL Server, MS Access, Informix, Watcom, and other via ODBC
I-Kinetics, Inc.	3	Oracle, Informix, Sybase, and others via ODBC
Imaginary	4	MSQL
InterSoft	3	Essentia
Intersolv	3	DB2, Ingres, Informix, Oracle, Microsoft SQL Server, Sybase 10/11
JavaSoft	1	Several dozen through ODBC drivers*
KonaSoft, Inc	3,4	Sybase, Oracle, Informix
NetAway	3	Oracle, Informix, Sybase, MS SQL Sever, DB2, other via ODBC
OpenLink	3	Oracle, Informix, Sybase, MS SQL, CAIngres, Progress, Unify, PostgreSQL, Solid, and other via ODBC
Oracle Corporation	2	Oracle
SAS Institute Inc	3,4	SAS, and via SAS/ACCESS, Oracle , Informix, Ingres, and ADABAS
SCO	3	Informix, Oracle, Ingres, Sybase, Interbase
StormCloud Development	3	Any DBMS that is accessible via ODBC

เอกสารนี้เป็นเอกสารของบริษัทที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 (ต่อ)

Vendor	Type	DBMS's supported
Sybase, Inc	3,4	Sybase SQL Server and more than 25 enterprise and legacy datase server via Sybase OmniCONNECT
Symantec	3	Oracle, Sybase, MS SQL Server, MS Access Watcom and other via ODBC
Trifox,inc	3	ADABAS, DB2, Informix, Ingres, Oracle, Rdb, SQL Sever, Sybase, and legacy systems via GENESIS
Visigenic	3	Several dozen through ODBC drivers*
WebLogic	2	Oracle, Sybase, MS SQL Server
WebLogic	3	Several dozen through ODBC drivers*
XDB Systemss, Inc.	1,3	Many databases through ODBC*
Yard Software GmbH	4	YARD-SQL Database

5.10 บทสรุป

ในบทนี้ได้กล่าวถึงความสามารถของภาษาจาวาที่สามารถทำงานในด้านระบบฐานข้อมูลไคล์แอนท์เซิร์ฟเวอร์ซึ่งสามารถที่จะปฏิบัติงานได้ทุกแพลตฟอร์มและสามารถลดจำนวนบุคลากรในการดูแลระบบน้อยลงที่ไม่ต้องไปดูแลในส่วนของไคล์แอนท์ โดยการที่จาวาสามารถติดต่อกับฐานข้อมูลได้นั้นจะอาศัย JDBC เป็นเพื่อสื่อสารกับฐานข้อมูล แต่พื้นฐานสำหรับการสร้างระบบเชื่อมฐานข้อมูลนั้นก็คือความรู้ความเข้าใจเกี่ยวกับหลักการงานหลักการออกแบบของฐานข้อมูลชนิดสัมพันธ์ (Relational Database) ระบบจัดการฐานข้อมูล (DBMS : Database Management System) และภาษา SQL

บทที่ 6

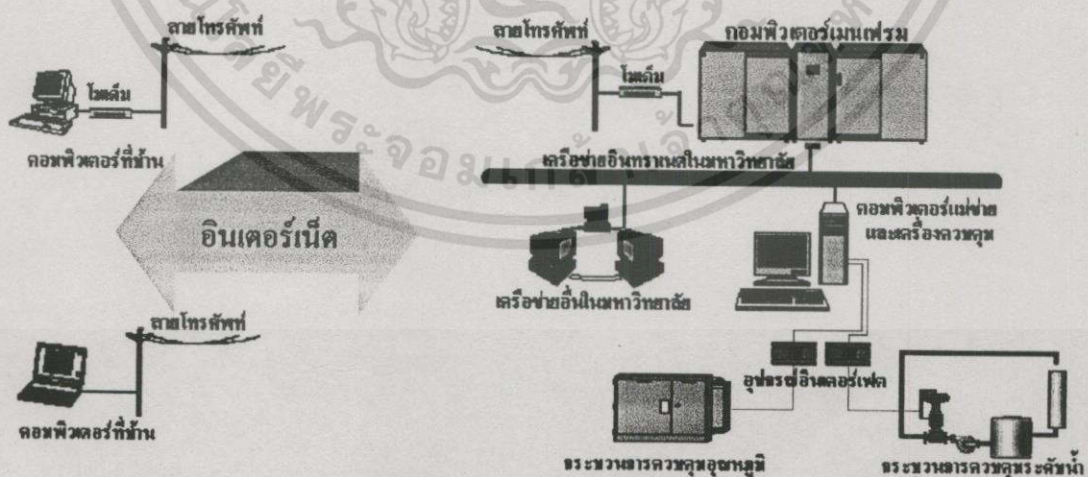
การออกแบบและหลักการทำงาน

6.1 บทนำ

ในบทนี้จะแสดงถึงการออกแบบในส่วนต่าง ๆ ที่เกี่ยวข้องกับวิทยานิพนธ์ โดยสามารถแบ่งออกได้ 3 ส่วนใหญ่ คือการออกแบบในส่วนของระบบเซิร์ฟเวอร์ซึ่งจะเป็นการกำหนดขอบเขตหน้าที่การทำงานของระบบเซิร์ฟเวอร์และผู้ที่เกี่ยวข้องกับเซิร์ฟเวอร์ , การออกแบบในส่วนของการอินเทอร์เน็ตเฟสเวิร์คแวร์เพื่อใช้ในการควบคุมกระบวนการ และส่วนของการออกแบบโปรแกรมทั้งในฝั่งไคลเอนท์และเซิร์ฟเวอร์เพื่อใช้ในการควบคุมกระบวนการจากเบราเซอร์และในส่วนเซิร์ฟเวอร์ซึ่งจะเป็นการสื่อสารข้อมูลบนอินเทอร์เน็ตไปยังไคลเอนท์หลาย ๆ เครื่องรวมถึงการจัดลำดับสิทธิในการควบคุมกระบวนการผ่านอินเทอร์เน็ต

6.2 การออกแบบระบบของเซิร์ฟเวอร์

ในการควบคุมกระบวนการผู้ควบคุมสามารถเชื่อมต่อเครือข่ายเพื่อเข้ามาสู่เซิร์ฟเวอร์ ได้ 2 ทางคือการเข้ามาควบคุมโดยผ่านเครือข่ายอินทราเน็ต (Intranet) ภายในองค์กรเองหรือเครือข่ายอินเทอร์เน็ตผ่านทางโมเด็มเพื่อเชื่อมต่อเข้ากับเซิร์ฟเวอร์ของที่กระบวนการติดตั้งอยู่ ดังรูปที่ 6.1



รูปที่ 6.1 แสดงการเชื่อมต่อคอมพิวเตอร์ของผู้ควบคุมผ่านทางเครือข่ายอินเทอร์เน็ตหรือเครือข่ายอินทราเน็ตเพื่อควบคุมกระบวนการ

ในการออกแบบระบบเซิร์ฟเวอร์ของงานวิจัยนี้ ได้แยกส่วนของระบบเซิร์ฟเวอร์ออกเป็น 2 ส่วน คือ ส่วนของระบบเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการของแต่ละกระบวนการและระบบเซิร์ฟเวอร์ที่ทำหน้าที่บริการอินเทอร์เน็ตให้กับไคลเอนต์หรือผู้ที่ต้องการควบคุมกระบวนการ โดยมีโครงสร้างแสดงในรูปที่ 6.2

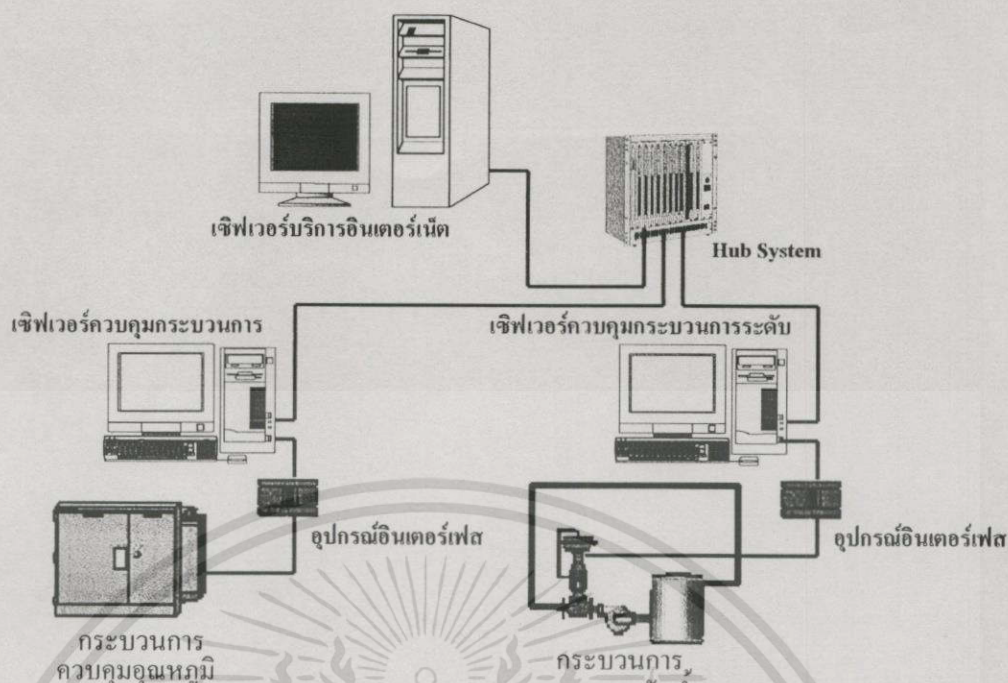
6.2.1 ระบบเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการ

ระบบเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการ จะประกอบไปด้วยเครื่องคอมพิวเตอร์ 2 เครื่อง แต่ละเครื่องจะทำหน้าที่เป็นเครื่องควบคุมพีไอซีของแต่ละกระบวนการทั้งกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิในตู้อบ โดยจะรับค่าพารามิเตอร์ต่าง ๆ จากผู้ควบคุมที่ปรับแต่งค่าจากเบราเซอร์ผ่านทางอินเทอร์เน็ตเข้ามาคำนวณตามระบบควบคุมพีไอซีและส่งผลการคำนวณที่ได้ออกไปควบคุมกระบวนการผ่านทางชุดอุปกรณ์อินเตอร์เฟส (Interface) จากนั้นจึงอ่านผลตอบสนองของการควบคุมกระบวนการเข้ามาจากชุดอุปกรณ์อินเตอร์เฟส และส่งผลตอบสนองของการควบคุมที่อ่านได้ไปบันทึกที่ฐานข้อมูลเพื่อให้ระบบแม่ข่ายที่บริการอินเทอร์เน็ตส่งข้อมูลต่าง ๆ ไปแสดงผลที่เบราเซอร์ของผู้ควบคุมต่อไป

6.2.2 ระบบเซิร์ฟเวอร์เพื่อให้บริการอินเทอร์เน็ต

ระบบเซิร์ฟเวอร์เพื่อให้บริการทางอินเทอร์เน็ต ทำหน้าที่ดังนี้

1. ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ (Web Server) ซึ่งทำหน้าที่ให้บริการเว็บเพจของผู้ที่ต้องการเข้ามาควบคุมกระบวนการ นอกจากนี้ได้ใช้โปรแกรม ASP (Active Server Page) ทำหน้าที่ควบคุมการทำงานของเว็บเพจเพื่อใช้ในการติดต่อฐานข้อมูล (Database) และแสดงผลที่เว็บเพจของผู้ควบคุมแต่ละราย
2. ทำหน้าที่เป็นดาต้าเบสเซิร์ฟเวอร์ (Database Server) เป็นระบบฐานข้อมูลที่ใช้ร่วมกันหลาย ๆ ผู้ใช้ นับตั้งแต่หลักสิบถึงหลักร้อย ระบบฐานข้อมูลแบบนี้เป็นระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ที่แท้จริงสามารถใช้ได้ในลักษณะไคลเอนต์/เซิร์ฟเวอร์และมีความสามารถให้บริการแก่ไคลเอนต์จำนวนมากในเวลาพร้อมกันได้ โดยในงานวิทยานิพนธ์ชิ้นนี้ได้ใช้ MS SQL Server เป็นดาต้าเบสเซิร์ฟเวอร์



รูปที่ 6.2 แสดงโครงสร้างระบบเซิร์ฟเวอร์ของงานวิจัย

6.3 การกำหนดสิทธิในการควบคุม

โปรแกรมที่เครื่องเซิร์ฟเวอร์จะอนุญาตให้ผู้ที่ทำการควบคุมกระบวนการปัจจุบันอยู่นี้ได้รับสิทธิในการควบคุมสูงสุด 2 คนคือ ผู้ที่ได้รับสิทธิในการควบคุมกระบวนการอุณหภูมิและระดับน้ำ การได้รับสิทธิในการควบคุมสูงสุดในที่นี้หมายถึงการที่ผู้ควบคุมสามารถควบคุมทุกกระบวนการได้เต็มที่ไม่ว่าจะเป็นการปรับเปลี่ยนค่าพารามิเตอร์ของกระบวนการ เปลี่ยนแปลงค่าเป้าหมายได้ตามต้องการซึ่งจะเหมือนการที่ผู้ควบคุมอยู่นั้นกำลังควบคุมกระบวนการอยู่จริง ส่วนผู้ควบคุมท่านอื่น ๆ จะถูกกำหนดสิทธิให้สามารถทำได้เฉพาะการดูแลของการเปลี่ยนแปลงในการควบคุมเท่านั้น นอกจากนี้ความสำคัญในการใช้สิทธิในโปรแกรมสามารถแบ่งแยกออกได้ 3 ระดับประกอบด้วย

1. ระดับผู้บริหารระบบ (Administrator) ระดับนี้จะถูกกำหนดให้มีสิทธิทุกประการในระบบของเซิร์ฟเวอร์ซึ่งเครื่องควบคุมกระบวนการทั้ง 2 กระบวนการติดตั้งอยู่ หน้าที่ของระดับผู้บริหารได้แก่ การเพิ่มหรือลบรายชื่อผู้มีสิทธิในการควบคุม การปิดหรือเปิดระบบ เป็นต้น ผู้บริหารระบบนี้จะถูกกำหนดให้เป็นได้เพียงคนเดียวเท่านั้น

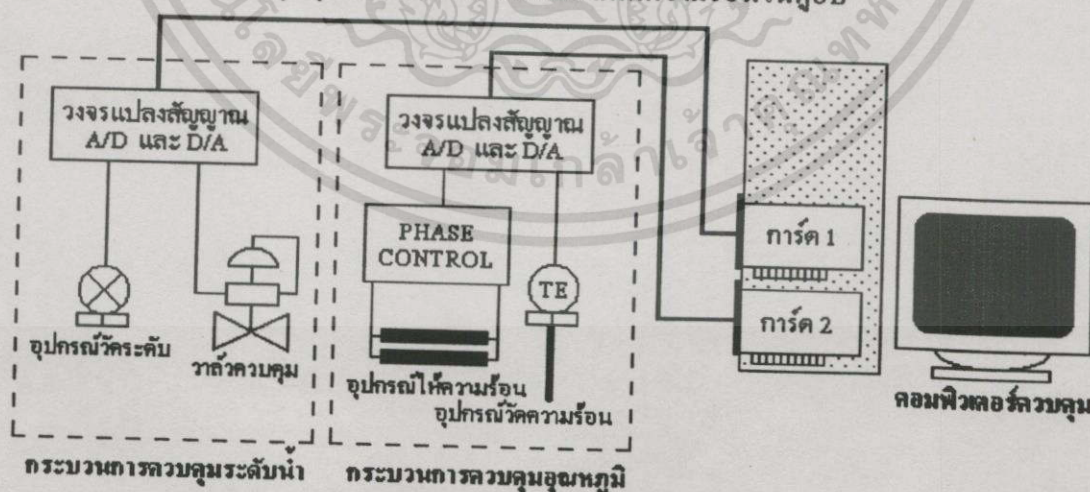
2. ระดับผู้ควบคุมกระบวนการหลัก ระดับนี้จะได้รับสิทธิในการควบคุมทุกประการและถูกกำหนดให้มีอภิสิทธิ์ในการจัดลำดับ(priority queue)[6] การควบคุมสูงสุด (ไม่ต้องรอคิวในการจัดลำดับ) เนื่องจากจะเป็นการอำนวยความสะดวกให้กับผู้ควบคุมหลักเพื่อที่จะสามารถเข้ามาปรับแต่งหรือแก้ปัญหาาระบบของกระบวนการเพื่อเตรียมการตามความเหมาะสมต่อไป

3. ระดับผู้ควบคุมทั่วไปที่มีชื่ออยู่ในทะเบียนผู้มีสิทธิเข้ามาควบคุม ระดับนี้ทุกคนจะมีความสำคัญเท่าเทียมกันหมด ดังนั้น การจัดลำดับในการควบคุมกระบวนการจะเป็นไปตามคิว ซึ่งผู้ที่เข้ามาสู่โปรแกรมเพื่อควบคุมก่อนก็จะถูกกำหนดสิทธิให้สามารถควบคุมกระบวนการได้ก่อน ส่วนผู้ที่เข้ามาทีหลังจะต้องรอนจนกระทั่งผู้ทำการควบคุมอยู่ทำการควบคุมจนเสร็จสมบูรณ์ จากนั้นจึงจะได้รับสิทธิที่จะเข้ามาควบคุมต่อไป

6.4 การอินเตอร์เฟซฮาร์ดแวร์กับคอมพิวเตอร์

งานวิจัยชิ้นนี้ อุปกรณ์อินเตอร์เฟซจะสร้างขึ้นมาสองชุดเพื่อใช้ควบคุมกระบวนการสองกระบวนการที่แยกกันคือ กระบวนการควบคุมระดับการนำและกระบวนการควบคุมอุณหภูมิ ดังนั้น โปรแกรมพีไอดีที่ทำหน้าเป็นตัวควบคุมแยกกันคำนวณโดยใช้โปรแกรมและคอมพิวเตอร์ 2 เครื่อง ดังที่แสดงไปแล้วในรูปที่ 6.2 ในผังรูปที่ 6.3 แสดงแบบจำลองการอินเตอร์เฟซกับกระบวนการ โดยการอินเตอร์เฟซแต่ละกระบวนการจะประกอบไปด้วยอุปกรณ์ต่าง ๆ ดังนี้

1. การ์ดอินเตอร์เฟซ ซึ่งจะทำหน้าที่รับข้อมูลดิจิทัลเข้ามาสู่คอมพิวเตอร์เพื่อประมวลผลในโปรแกรมเครื่องควบคุมและส่งข้อมูลดิจิทัลที่ได้จากการคำนวณออกไป
2. วงจรแปลงสัญญาณ A/D และ D/A วงจรนี้จะทำหน้าที่เป็นอุปกรณ์ที่แปลงสัญญาณจากอนาลอกเป็นดิจิทัล (A/D) และจากดิจิทัลเป็นอนาลอก (D/A)
3. อุปกรณ์ที่เป็นตัวอินพุตและเอาต์พุตของกระบวนการ ซึ่งอุปกรณ์ที่เป็นอินพุตได้แก่ เครื่องตรวจวัดระดับของน้ำและเครื่องวัดอุณหภูมิ ส่วนอุปกรณ์ที่เป็นเอาต์พุตได้แก่ วาล์วควบคุมระดับน้ำและเครื่องควบคุมมุมแรงดันไฟฟ้าในการกำเนิดความร้อนในตู้อบ

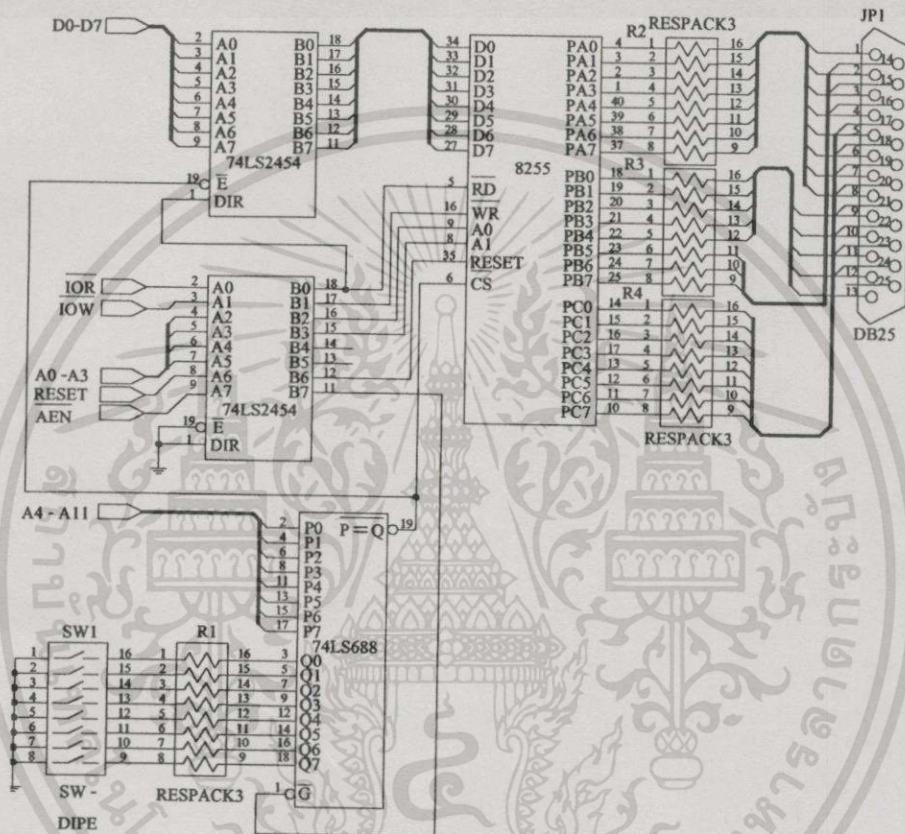


รูปที่ 6.3 แสดงแบบจำลองการอินเตอร์เฟซกับกระบวนการ

6.5 วงจรอินเทอร์เฟซ

6.5.1 ส่วนอินเทอร์เฟซ (Interfacing Unit)

ส่วนอินเทอร์เฟซมีหน้าที่ในการรับและส่งข้อมูลแบบขนานขนาด 8 บิต ระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์อินพุต / เอาท์พุต



รูปที่ 6.4 วงจรอินเทอร์เฟซ

ในรูปที่ 6.4 วงจรอินเทอร์เฟซประกอบด้วย

- ไอซีเบอร์ 8255 ทำหน้าที่เป็นอินพุต / เอาท์พุตพอร์ท (Input / Output Port)
- ไอซีเบอร์ 74245 ตัวที่ 1 ทำหน้าที่ในการรับ - ส่งข้อมูล
- ไอซีเบอร์ 74245 ตัวที่ 2 ทำหน้าที่ในการส่งสัญญาณควบคุม ไอซีเบอร์ 8255
- ไอซีเบอร์ 74688 ทำหน้าที่ในการเปรียบเทียบหมายเลขพอร์ท

สัญญาณจากเครื่องคอมพิวเตอร์ที่ใช้ในการควบคุมวงจรรีเลย์เพื่อใช้ในการรับ - ส่งข้อมูลประกอบด้วยสัญญาณบัสข้อมูล (D0 - D7), บัสแอดเดรส (A0 - A11), $\overline{\text{IOR}}$, $\overline{\text{IOW}}$ และ $\overline{\text{AEN}}$

สัญญาณแอดเดรส A0 – A1 ทำหน้าที่ในการเป็นสัญญาณควบคุม (A0 – A1) ในการเลือกพอร์ทของ 8255 ให้เป็นพอร์ท A, พอร์ท B, พอร์ท C หรือ พอร์ทควบคุม

สัญญาณแอดเดรส A4 – A11 ทำหน้าที่ในการเลือกหมายเลขพอร์ท โดยมีหมายเลขพอร์ทตั้งแต่ \$300 - \$31F

สัญญาณ $\overline{\text{IOR}}$ ทำหน้าที่ในการสั่งให้วงจรรับข้อมูล โดยส่งสัญญาณไปยังขา DIR ของไอซี 74245 ตัวที่ 1 เพื่อทำการรับข้อมูล และขา $\overline{\text{RD}}$ ของไอซี 8255 เพื่อให้รับข้อมูลจากอุปกรณ์อินพุต

สัญญาณ $\overline{\text{IOW}}$ ทำหน้าที่ในการสั่งให้วงจรรีจิสเตอร์เฟสส่งข้อมูลออกไปยังอุปกรณ์เอาต์พุต โดยทำการส่งสัญญาณไปที่ขา $\overline{\text{WR}}$ ของไอซีเบอร์ 8255 เพื่อกำหนดให้ 8255 ทำหน้าที่เป็นพอร์ทเอาต์พุตเพื่อส่งข้อมูลไปยังอุปกรณ์เอาต์พุต แต่สัญญาณ $\overline{\text{IOW}}$ ไม่จำเป็นต้องส่งไปที่ขา DIR ของไอซี 74245 ตัวที่ 1 เพราะสถานะปกติของขา DIR เป็น “1” ซึ่งหมายถึงการส่งข้อมูลออกจาก A1 – A8 ไปยัง B1 – B8 แต่ถ้าสถานะเป็น “0” จะหมายถึงการรับข้อมูลจาก B1 – B8 เข้ามา A1 – A8

สัญญาณ AEN ทำหน้าที่ในการควบคุมการเปิด – ปิดของบัสข้อมูล เพื่อป้องกันไม่ให้ไอซี 74688 ทำการดีโค้ด (Decode) ระหว่างกระบวนการ DMA สัญญาณ AEN จะทำการควบคุมการทำงานของไอซี 74688 โดยจะมีสถานะเป็น “1” เมื่อเกิดกระบวนการ DMA ซึ่งจะทำให้ 74688 ไม่ทำงาน แต่ถ้ามีการอ่าน – เขียนข้อมูลบนพอร์ท สัญญาณ AEN จะมีสถานะเป็น “0” ทำให้ 74688 ทำงาน แล้วนำสัญญาณเอาต์พุตของ 74688 ที่ได้จากขา $P=Q$ ไปควบคุมการเปิด – ปิดบัสข้อมูลที่ขา $\overline{\text{E}}$ ของไอซี 74245 ตัวที่ 1 ถ้าเอาต์พุตของ 74688 เป็น “0” จะเป็นการสั่งให้เปิดพอร์ทของ 74245 แต่ถ้าเป็น “1” ไอซี 74245 จะไม่ทำงาน โดยที่เอาต์พุตของ 74688 จะเป็น “0” ได้ก็ต่อเมื่อสัญญาณแอดเดรส (A4 – A11) ตรงกับสัญญาณจาก DIP – SW ที่ได้ตั้งไว้ และสัญญาณ AEN เป็น “0”

ในวิชานีพนธ์นี้จะใช้พอร์ท \$300 - \$303 โดยกำหนดให้

หมายเลขพอร์ท \$300 เป็นพอร์ท A

หมายเลขพอร์ท \$301 เป็นพอร์ท B

หมายเลขพอร์ท \$302 เป็นพอร์ท C

หมายเลขพอร์ท \$303 เป็นพอร์ทควบคุม

และกำหนดให้

พอร์ท A เป็นพอร์ทอินพุต

พอร์ท B เป็นพอร์ทเอาต์พุต

พอร์ท C บนเป็นพอร์ทเอาต์พุตที่ใช้ในการควบคุมวงจร A/D Converter

พอร์ท C ล่างเป็นพอร์ทเอาต์พุตที่ใช้ในการควบคุมวงจร D/A Converter

Control Word ที่ใช้กับไอซี 8255 ในตอนเปิดเครื่องคือ \$90

เนื่องจากหมายเลขพอร์ทเป็น \$300 - \$303 ดังนั้นจึงสามารถตั้ง DIP – SW ได้ดังนี้

	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
หมายเลขพอร์ท \$300	0	0	1	1	0	0	0	0	0	0	0	0
หมายเลขพอร์ท \$301	0	0	1	1	0	0	0	0	0	0	0	1
หมายเลขพอร์ท \$302	0	0	1	1	0	0	0	0	0	0	1	0
หมายเลขพอร์ท \$303	0	0	1	1	0	0	0	0	0	0	1	1

ดังนั้นหากต้องการให้วงจรทำงานที่หมายเลขพอร์ท \$300 - \$303 จะต้องให้สัญญาณที่ขาของไอซี 74688 เป็น “0” ที่ Q1 – Q4 และ Q7 – Q8 ส่วนที่ Q5 – Q6 เป็น “1” ส่วนการตั้ง DIP – SW ต้องกำหนดให้ 1 – 4 และ 7 – 8 อยู่ในสภาวะ ON เพื่อให้ลงกราวนด์คือเป็น “0” และที่ 5 – 6 อยู่ในสภาวะ OFF เพื่อให้มีสภาวะเป็น “1”

6.5.1.1 การส่งคำสั่งควบคุม (Control Word) ให้กับไอซี 8255

ก่อนที่ไอซี 8255 จะทำงานต้องมีการส่งสัญญาณไปให้ก่อน เพื่อให้ 8255 ทราบการทำงานของพอร์ทต่างๆ โดยในการส่งคำสั่งควบคุมนั้นต้องให้ขา A0 และ A1 อยู่ในสภาวะ “1” เพื่อส่งให้ไอซี 8255 รับคำสั่งควบคุมแต่ในวิทยานิพนธ์นี้กำหนดให้พอร์ท A ทำหน้าที่เป็นพอร์ทอินพุต พอร์ท B เป็นพอร์ทเอาต์พุต และพอร์ท C บนและล่างเป็นพอร์ทเอาต์พุต ดังนั้นจึงมีคำสั่งควบคุมเป็น \$90 การส่งคำสั่งควบคุมออกไปยังพอร์ท \$303 จะใช้คำสั่ง Port [\$303]: = \$90 โดยมีการส่งสัญญาณดังนี้

สัญญาณแอดเดรส A0 – A11 เป็นหมายเลขพอร์ทคือ \$303 ดังนั้นสัญญาณแอดเดรส A0 และ A1 จะเป็น “1” ทั้งคู่ เพื่อเป็นการบอกให้ไอซี 8255 ทราบว่าเป็นการส่งคำสั่งควบคุมแอดเดรส A4 – A11 จะใช้เพื่อเปรียบเทียบกับหมายเลขพอร์ทกับ DIP – SW ว่าตรงกับที่ได้ตั้งไว้หรือไม่ โดยมีสัญญาณ AEN เป็นสัญญาณส่งให้ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกันและสัญญาณ AEN เป็น “0” จะให้เอาต์พุตเป็น “0” เพื่อนำไปเปิดพอร์ทข้อมูลของไอซี 74245 ตัวที่ 1 ในการส่งข้อมูล \$90ออกไปยังไอซี 8255 โดยสัญญาณ \overline{IOW} เป็นสัญญาณที่ส่งไปที่ขา \overline{WR} ของไอซี 8255 เพื่อเป็นสัญญาณของการส่งข้อมูลมาให้ไอซี 8255 เมื่อข้อมูล \$90 ถูกส่งมาให้ไอซี 8255 แล้ว ไอซี 8255 จะเซ็ทพอร์ท A ให้เป็นพอร์ทอินพุต, พอร์ท B เป็นพอร์ทเอาต์พุตและพอร์ท C บนและล่างเป็นพอร์ทเอาต์พุตตามคำสั่งควบคุม

6.5.1.2 การรับข้อมูลจากอุปกรณ์ภายนอก

การรับข้อมูลจากอุปกรณ์ภายนอกเพื่อนำเข้ามาประมวลผลภายในคอมพิวเตอร์จะใช้คำสั่ง $x := \text{Port } [\$300]$ โดยคอมพิวเตอร์จะส่งสัญญาณแอดเดรส \$300 ออกไปยังแอดเดรสบัส ซึ่งจะทำให้ A0 และ A1 มีค่าเป็น "0" เพื่อไปเข้าที่ขา A0 และ A1 ของไอซี 8255 ซึ่งเป็นการสั่งให้ไอซี 8255 ทำงานกับพอร์ท A สัญญาณแอดเดรส A4 ถึง A11 เป็นตัวเปรียบเทียบเลขพอร์ทกับ DIP – SW โดยมีสัญญาณ AEN เป็นตัวสั่งให้ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกับที่ตั้งไว้และสัญญาณ AEN เป็น "0" จะให้เอาต์พุตเป็น "0" เพื่อนำไปเปิดพอร์ทข้อมูลของเอาต์พุตไอซี 74245 ตัวที่ 1 โดยที่สัญญาณ $\overline{\text{IOR}}$ จะเข้ามาที่ขา $\overline{\text{RD}}$ ของไอซี 8255 เพื่อให้รับข้อมูลอินพุตเข้ามาทางพอร์ท A และส่งข้อมูลให้กับคอมพิวเตอร์ผ่านทางบัสข้อมูล เช่นถ้าข้อมูลทางพอร์ท A มีค่าเท่ากับ \$66 จะทำให้ตัวแปร x มีค่าเท่ากับ \$66 ด้วย

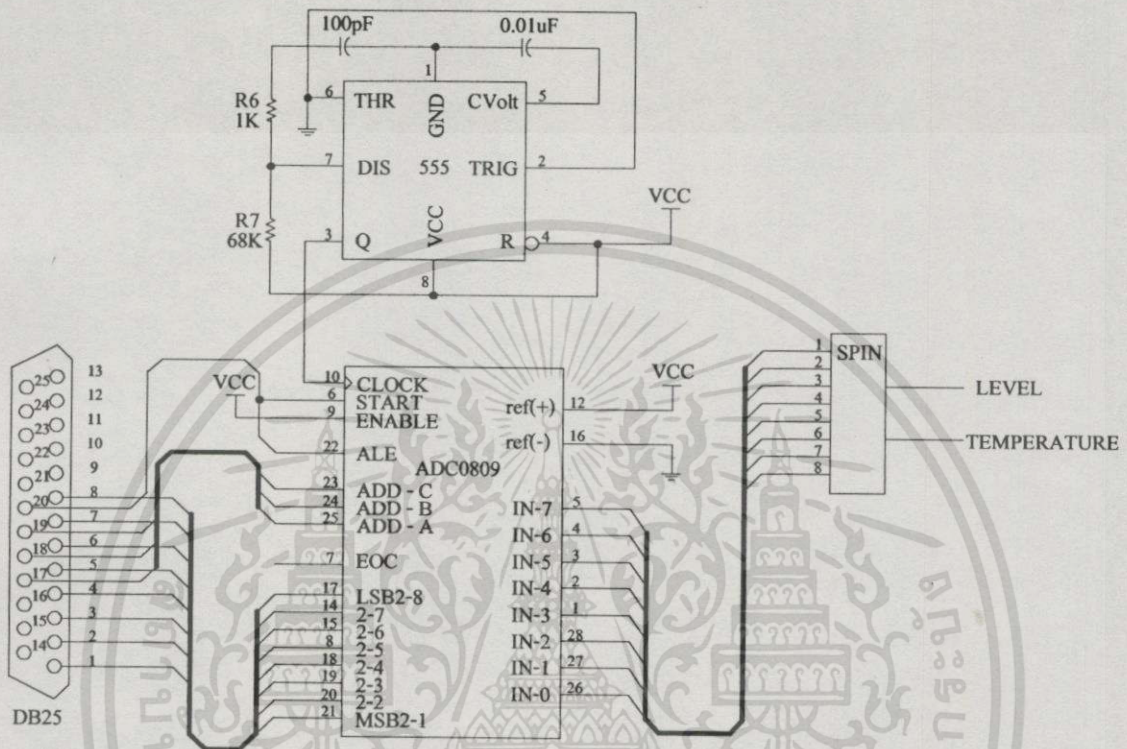
6.5.1.3 การส่งข้อมูลให้กับอุปกรณ์ภายนอก

การส่งข้อมูลให้กับอุปกรณ์ภายนอกใช้คำสั่ง $\text{Port } [\$301] := \88 โดยคอมพิวเตอร์จะส่งสัญญาณแอดเดรส \$301 ออกไปยังแอดเดรสบัส ทำให้ A0 และ A1 ซึ่งมีค่าเท่ากับ "1" และ "0" ตามลำดับไปเข้าที่ขา A0 และ A1 ของไอซี 8255 เพื่อสั่งให้ทำงานกับพอร์ท B สัญญาณแอดเดรส A4 – A11 จะเป็นตัวเปรียบเทียบหมายเลขพอร์ทกับ DIP – SW โดยมีสัญญาณ AEN เป็นสัญญาณสั่งให้ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกับที่ตั้งไว้และ AEN เป็น "0" จะให้เอาต์พุตเป็น "0" เพื่อสั่งให้ไอซี 74245 ตัวที่ 1 เปิดพอร์ทข้อมูล โดยที่สัญญาณ $\overline{\text{IOW}}$ จะเข้าที่ขา $\overline{\text{WR}}$ ของไอซี 8255 เพื่อให้ทราบว่ามีการส่งข้อมูลมาให้ ดังนั้นข้อมูล \$88 จะออกไปยังพอร์ท B ของไอซี 8255

ในกรณีที่ต้องการส่งข้อมูลออกไปยังพอร์ท C ก็จะทำเช่นเดียวกันเพียงแต่จะเปลี่ยนหมายเลขพอร์ทเป็น \$302 แทน

6.5.2 ส่วนอุปกรณ์อินพุต/เอาต์พุต (Input/Output Unit)

6.5.2.1 A/D Converter



รูปที่ 6.5 วงจร A/D Converter

ในวงจร A/D Converter นี้ใช้ไอซี AD0809 ซึ่งเป็นไอซี A/D 8 แชนแนล มีความละเอียด 255 ระดับ และสามารถตั้งแรงดันอ้างอิงจากขา Ref+ และ Ref- โดยสามารถเลือกแชนแนลได้จากการควบคุมขา ADD - A ถึง ADD - C โดยมีการควบคุมดังนี้

ตารางที่ 6.1 การเลือกแชนแนลของวงจร A/D Converter นี้ใช้ไอซี AD0809

ADD - C	ADD - B	ADD - A	Channel
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6

สัญญาณที่ใช้ในการควบคุมการเลือกแชนแนลเป็นสัญญาณจากพอร์ท C ต่างคือ PC0 – PC2 โดย PC0 ถูกต่อเข้ากับขา ADD – A ส่วน PC1 ถูกต่อเข้ากับขา ADD – B และ PC2 ถูกต่อเข้ากับขา ADD – C ส่วนสัญญาณที่ใช้ในการสั่งเริ่มทำงานคือสัญญาณจากพอร์ท C ต่างคือ PC3 ซึ่งทำงานที่ขอบขาขึ้นหรือเมื่อมีการเปลี่ยนระดับลอจิกจาก “0” เป็น “1” โดยต่อเข้ากับขา Start และ ALE ของไอซี ADC0809

การทำงานของไอซี ADC0809 จำเป็นต้องมีวงจรกำเนิดพัลส์จากภายนอก ซึ่งในวิทยานิพนธ์นี้ใช้ไอซี 555 ในการผลิตพัลส์ ซึ่งความถี่ที่ใช้จะมีผลต่อการทำงาน ถ้าความถี่ที่ผลิตได้มีค่าน้อยจะทำให้ใช้เวลานานในการคำนวณ แต่ถ้าให้ความถี่สูงจะทำให้ใช้เวลาน้อยลง

อุปกรณ์ทางด้านอินพุตเช่น เครื่องวัดอัตราการไหล, แรงดัน หรือระดับของของเหลวจะถูกต่อเข้ากับแชนแนลต่างๆของไอซี ADC0809 ในวิทยานิพนธ์นี้มีการต่อเครื่องวัดระดับของเหลวเข้ากับแชนแนลที่ 3 และเครื่องวัดอุณหภูมิเข้ากับที่แชนแนลที่ 4 ดังนั้นหากต้องการรับอินพุตของระดับของเหลว จะต้องกำหนดให้พอร์ท C เป็น “\$0B” และในกรณีของอุณหภูมิพอร์ท C ต้องถูกกำหนดเป็น “\$0C” โดยที่พอร์ท A ของไอซี 8255 จะถูกต่ออยู่กับบัสข้อมูลของไอซี ADC0809

คำสั่งพื้นฐานที่ใช้กับ ADC0809 เพื่อให้ทำงานในแชนแนลที่ 3 คือ

Port [\$303]: = \$90; ส่งค่า Control Word ออกไปยังพอร์ทควบคุม

Port [\$302]: = \$03; เลือกแชนแนลที่ 3

Delay (50); หน่วงเวลาเพื่อให้เป็นขอบขาขึ้น

Port [\$302]: = \$0B; ทำให้เป็นขอบขาขึ้นเพื่อสั่งให้เริ่มการทำงาน

Delay (100); หน่วงเวลาให้ ADC0809 ทำงานเสร็จ

x: = Port [\$300]; รับอินพุตทางพอร์ท A ของไอซี 8255 แล้วนำมาเก็บไว้ในตัวแปร x เช่นถ้าอินพุตของแชนแนลที่ 3 เท่ากับ 2 โวลต์ ตัวแปร x จะมีค่าเท่ากับ \$66

วงจร A/D Converter มีไอซี 555 ซึ่งใช้เป็นวงจรอะสเตเบิลมัลติไวเบรเตอร์ โดยมีค่า $R_1 = 68K$, $R_2 = 1K$ และ $C_2 = 100 \text{ pF}$ ดังนั้นความถี่ที่ผลิตได้จึงมีค่าเท่ากับ

$$f = \frac{1}{T} = \frac{1.443}{(R_1 + 2R_2) \cdot C_2} = 206.142 \text{ kHz} \quad (6-1)$$

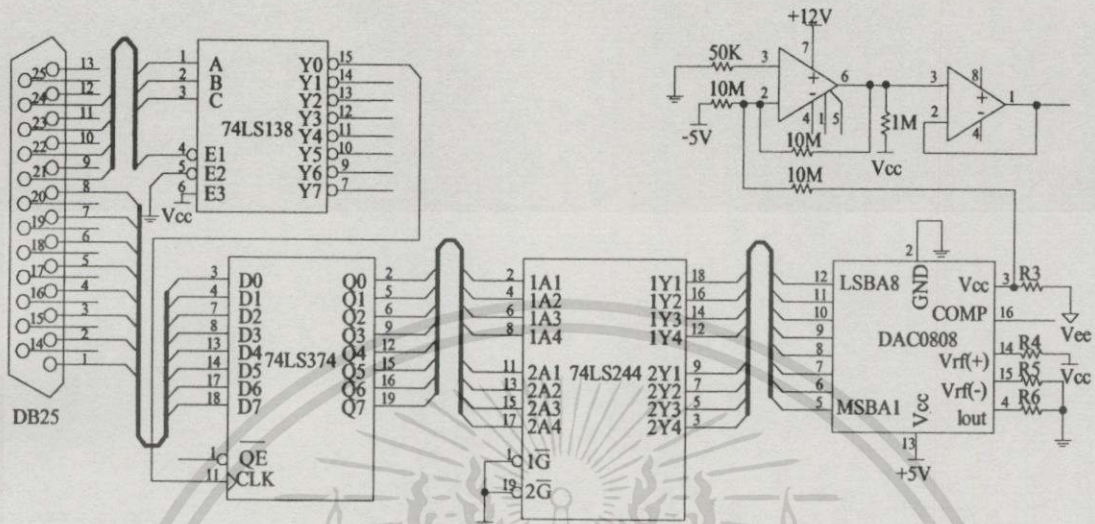
ดังนั้นเวลาที่ใช้ในการทำงานสูงสุดในแต่ละรอบของไอซี ADC0809 คือ

$$\frac{1}{206.142 \text{ kHz}} \cdot 255 = 1.237 \text{ msec} \quad (6-2)$$

สาเหตุที่ต้องคูณด้วย 255 เนื่องจากจำนวนพัลส์สูงสุดที่ใช้ในการทำให้เกิดเอาต์พุต \$FF

เท่ากับ 255 พัลส์ เมื่อมีอินพุตสูงสุด 5 โวลต์ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.2.2 วงจร D/A Converter



รูปที่ 6.6 วงจร D/A Converter

วงจร D/A Converter ประกอบด้วยไอซี DAC0808, 74374, 74244 และ 74138 โดยแต่ละตัวมีหน้าที่ดังนี้

- DAC0808 ทำหน้าที่ในการแปลงสัญญาณดิจิทัลขนาด 8 บิตให้เป็นสัญญาณอนาล็อก
- 74374 ทำหน้าที่ในการแลทช์ (Latch) วงจรเพื่อให้ข้อมูลค้าง เนื่องจากอุปกรณ์ภายนอกทำงานช้ากว่าเครื่องคอมพิวเตอร์
- 74244 ทำหน้าที่เป็นบัฟเฟอร์ข้อมูล
- 74138 ทำหน้าที่ในการเลือกแชนแนล ซึ่งในวงจรนี้ออกแบบเอาท์พุทไว้ทั้งหมด 8 แชนแนล

หลักการทํางาน

พอร์ท B เป็นพอร์ทในการรับส่งข้อมูลและสัญญาณจากพอร์ท C บนของไอซี 8255 เป็นสัญญาณควบคุม ซึ่งประกอบด้วย

สัญญาณ C7 เป็นสัญญาณบอกสถานะการทำงานของ D/A Converter ถ้ามีสถานะเป็น "0" คือสถานะที่มีการทํางาน และเป็น "1" เมื่อไม่มีการทํางาน โดยต่อเข้ากับขา E1 ของไอซี 74138

สัญญาณ C6 เป็นสัญญาณที่ใช้ในการเลือกแชนแนลของไอซี 74138 โดยต่อเข้ากับขา C

สัญญาณ C5 เป็นสัญญาณที่ใช้ในการเลือกแชนแนลของไอซี 74138 โดยต่อเข้ากับขา B

สัญญาณ C4 เป็นสัญญาณที่ใช้ในการเลือกแชนแนลของไอซี 74138 โดยต่อเข้ากับขา A

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกแขนแนลมีลักษณะดังนี้

ตารางที่ 6.2 การเลือกแขนแนลเอาต์พุตของไอซี 74138

C7	C6	C5	C4	Channel
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7

เอาต์พุตของไอซี 74138 คือขา Y0 – Y7 แต่ละขาจะต่อเข้ากับขา CLK ของไอซี 74374 แต่เนื่องจากมีเอาต์พุตเพียงตัวเดียวที่ใช้ในการควบคุมคอนโทรลแล้วจึงต่อเพียงขาเดียวเท่านั้น หากมีอุปกรณ์เอาต์พุตหลายตัวก็สามารถเลือกใช้อื่นได้

วงจรนี้ใช้ $V_{cc} +5 V$ และ $V_{ee} -5 V$ โดยต่อผ่านตัวต้านทาน $5 K\Omega$ และ $V_{ref} (+) = +5 V$ และ $V_{ref} (-) = 0 V$ และใช้ซอฟต์แวร์ในการแปลงให้อยู่ในรูปสัญญาณมาตรฐาน $1 - 5 V$ ถึงแม้ว่าจะสามารถใช้ฮาร์ดแวร์ในการแปลงให้อยู่ในรูปสัญญาณมาตรฐานได้ แต่สัญญาณที่ได้อาจมีการสูญเสียไปทำให้เกิดความผิดพลาดขึ้นได้และการแก้ไขความผิดพลาดทางฮาร์ดแวร์ทำได้ยากกว่าการใช้ซอฟต์แวร์

คำสั่งพื้นฐานที่ใช้ในการทำงานของ D/A Converter

- Port [\$303]: = \$90; ส่งค่า Control word ให้กับไอซี 8255
- Port [\$302]: = \$80; สั่งให้แขนแนล 0 ทำงาน
- Port [\$301]: = \$80; ส่งข้อมูล \$80 ออกมาที่ 74374 เพื่อรอสัญญาณ CLK
- Delay (100); หน่วงเวลาเพื่อให้เกิดพัลส์
- Port [\$302]; = \$00; ทำให้เกิดพัลส์ขาขึ้นที่ขา CLK ของ 74374 ซึ่งทำให้ 74374 ทำงานและข้อมูลถูกส่งไปที่ DAC0808

เอาต์พุตของ DAC0808 จะมีลักษณะตรงข้ามกับอินพุตคือถ้าอินพุตเท่ากับ \$00 จะได้เอาต์พุต 5 V แต่หากอินพุตเท่ากับ \$FF จะได้เอาต์พุตเท่ากับ 0 V ดังนั้นจึงต้องมีวงจรในการเปลี่ยนเอาต์พุตให้เท่ากับ 0 V เมื่ออินพุตเป็น \$00 และได้เอาต์พุต 5 V เมื่ออินพุตเท่ากับ \$FF วงจรที่นำมาใช้จะต่อออปแอมป์

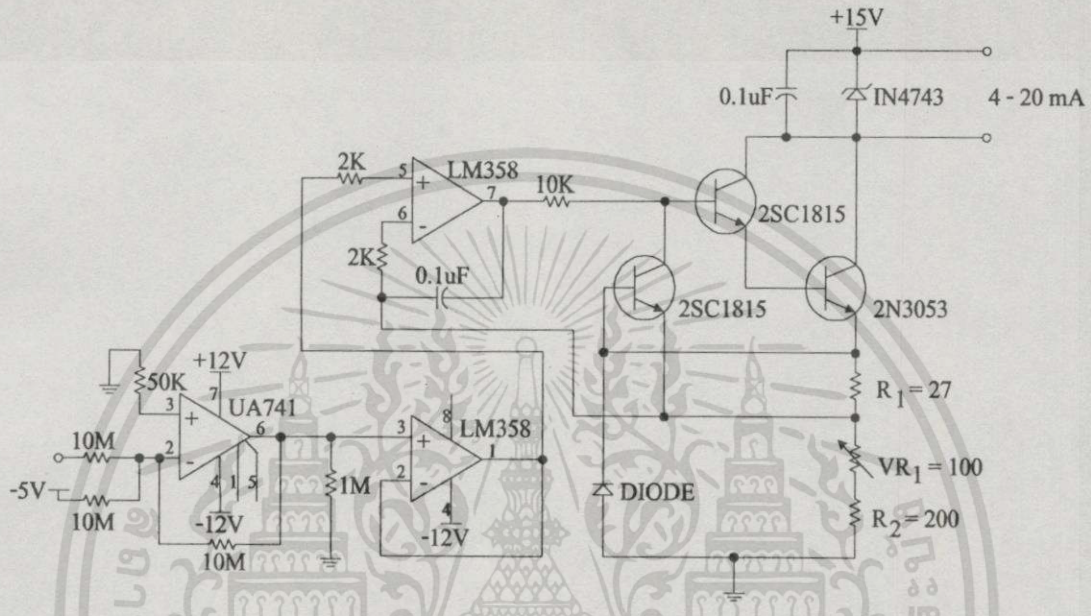
714

ให้อยู่ในรูปของวงจรขั้วมิงอินเวอร์เตอร์แล้วนำมาต่อกับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะเท่านั้น มีอยู่ในระบบเอกสารอิเล็กทรอนิกส์ การคัดลอกหรือการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเวอร์เตอร์โดยใช้ออปแอมป์ LM358A อีกครั้ง เอาท์พุทที่ได้จะอยู่ในรูปแบบที่ต้องการ หากคอมพิวเตอร์ส่งข้อมูลออกไปเป็น \$66 เอาท์พุทที่ได้จากวงจร D/A Converter จะเท่ากับ 2 V

6.5.3 วงจรแปลงแรงดันเป็นกระแสไฟฟ้า (V/I Converter)



รูปที่ 6.7 วงจร V/I Converter

จากหลักการที่กำหนดให้ขาอินเวอร์ตติ้งและนอนอินเวอร์ตติ้งต้องมีแรงดันเท่ากัน ดังนั้นแรงดันตกคร่อม VR_1 และ R_2 จึงมีค่าเท่ากัน ดังนั้นกระแสที่ไหลผ่านจึงมีค่าเท่ากับ $V_{in} / 250 \Omega$ เมื่อค่าความต้านทาน VR_1 เท่ากับ 50Ω เมื่อพิจารณาความต้านทานที่นำมาต่อคร่อมซีเนอร์ไดโอด จะพบว่าไม่มีผลต่อค่ากระแส และในวงจรนี้ยังสามารถต่อค่าความต้านทานได้ 650Ω ซึ่งได้มาจาก $13 V / 20 mA$ เมื่อ IN4743A มีแรงดันซีเนอร์เท่ากับ $13 V$ นอกจากนี้ยังสามารถปรับค่ากระแสได้โดยใช้ VR_1

ทรานซิสเตอร์เบอร์ 1 คือ 2SC1815 และเบอร์ 2 คือ 2N3053 ใช้เป็นวงจรสวิทช์ ทรานซิสเตอร์แบบคาร์บิลด์ตันเพื่อควบคุมกระแสที่สูงกว่าแต่ใช้กระแสในการควบคุมเพียงเล็กน้อยเท่านั้น

ค่า I_{C2} เท่ากับ $20 mA$, $\beta_1 = 120$ และ $\beta_2 = 90$

ดังนั้น

$$I_{B2} = I_{C1} = \frac{I_{C2}}{\beta_2} = \frac{20mA}{90} \tag{6-3}$$

และ

$$I_{B1} = \frac{I_{C1}}{\beta_1} = \frac{20mA}{120} = 1.852 \mu A \quad (6-4)$$

จากสมการ (6-4) พบว่ากระแสจากออปแอมป์มีค่าเพียง 1.852 μA แต่สามารถควบคุมกระแสเอาต์พุตได้ถึง 20 mA

ทรานซิสเตอร์เบอร์ 3 คือ 2SC1815 ทำหน้าที่ในการป้องกันกระแสเกิน เมื่อกระแสเพิ่มขึ้นจะทำให้แรงดันตกคร่อม $R_1 = 27 \Omega$ เพิ่มขึ้นและทำให้แรงดันขาเบสและอีมิเตอร์ (V_{BE}) เพิ่มขึ้น ทำให้ทรานซิสเตอร์ทำงานได้มากขึ้นและมีกระแสไหลเข้าทางขาคอลเลกเตอร์ของทรานซิสเตอร์เบอร์ 3 มากขึ้นด้วย ดังนั้นกระแสที่ไหลเข้าขาเบสของทรานซิสเตอร์เบอร์ 1 น้อยลง กระแส เอาต์พุต จึงลดลงด้วย

เมื่อมีแรงดัน 5 V เข้าที่ขา 5 ซึ่งเป็นขาอินเวอร์ตติ้งของ LM358B ทำให้มีแรงดันตกคร่อมขา 6 ซึ่งเป็นขาอินเวอร์ตติ้งเท่ากับ 5 V ด้วยเช่นกัน ดังนั้นจึงมีแรงดันตกคร่อมที่ VR_1 และ R_5 ขนาด 200 Ω เท่ากับ 5 V และมีกระแสไหลผ่านเท่ากับ 5 V / 250 Ω เท่ากับ 20 mA ซึ่งมีค่าเท่ากับกระแสที่ไหลผ่านอุปกรณ์ควบคุมด้วยเช่นกัน แต่หากแรงดันที่เข้ามามีค่าเท่ากับ 1 V กระแสที่ไหลผ่านจะมีค่าเท่ากับ 1 V / 250 Ω หรือเท่ากับ 4 mA ส่วน C_1 และ C_2 นั้นทำหน้าที่ในการหน่วงเวลาเพื่อป้องกันไม่ให้กระแสเพิ่มขึ้นอย่างรวดเร็ว

6.6 การออกแบบโปรแกรม

6.6.1 โปรแกรม Delphi

โปรแกรมที่นำไปใช้ในการควบคุมกระบวนการนั้น ต้องมีการทำงานอยู่ 2 ลักษณะคือ

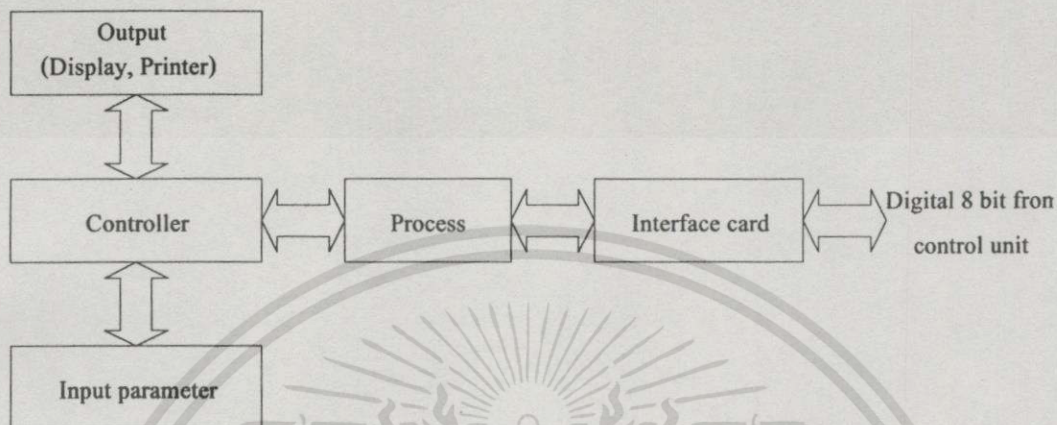
1. ควบคุมการทำงานของอุปกรณ์ในระบบ
2. ช่วยในการทำงาน (Application)

โปรแกรมที่ใช้ต้องมีส่วนของการอินเตอร์เฟส โดยมีการรับข้อมูลจากเครื่องมือวัดเช่น เซนเซอร์หรือทรานสดิวเซอร์ เข้ามาประมวลผลและนำผลลัพธ์ที่ได้ส่งออกไปควบคุมอุปกรณ์ภายนอก (Final element) การ์ดอินเตอร์เฟสต้องมีการทำงานร่วมกับโปรแกรม เนื่องจากต้องมีการรับส่งข้อมูลขนาด 8 บิตผ่านทางบัสข้อมูล (Data bus) ของคอมพิวเตอร์เพื่อนำมาให้โปรแกรมทำการประมวลผล แล้วจึงส่งผลลัพธ์ที่ได้ผ่านจอร์นอินเตอร์เฟสเพื่อนำไปควบคุมกระบวนการ

โปรแกรม Delphi เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่ทำงานบนระบบปฏิบัติการวินโดวส์ 98 หรือวินโดวส์ ME และสามารถนำมาใช้ในการควบคุมกระบวนการได้อย่างมีประสิทธิภาพ เนื่องจากมีข้อดีหลายประการเช่น สามารถพัฒนาได้อย่างรวดเร็ว, มีความยืดหยุ่นในการใช้งาน, ใช้งานง่าย

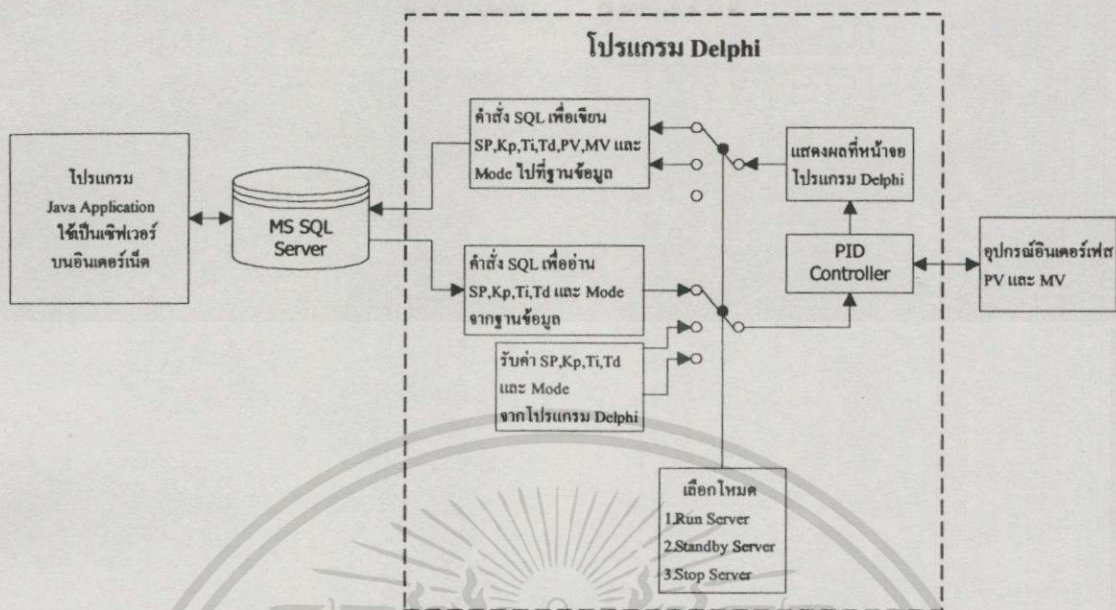
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสะดวกรวดเร็ว เป็นต้น ดังนั้นในวิทยานิพนธ์นี้จึงนำโปรแกรม Delphi มาใช้ควบคุมกระบวนการ



รูปที่ 6.8 บล็อกโคตะแกรมในการรับ – ส่งข้อมูลผ่านทางคาร์ดิแอนด์เฟส

นอกเหนือจากการใช้โปรแกรม Delphi ในการควบคุมกระบวนการแล้ว ในงานวิจัยชิ้นนี้ยังใช้โปรแกรม Delphi ทำเป็นเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการ โดยเซิร์ฟเวอร์จะทำหน้าที่ติดต่อกับโปรแกรมเครื่องควบคุมกระบวนการผ่านทางฐานข้อมูล ซึ่งจะรับค่าพารามิเตอร์ต่างๆ จากผู้ที่ได้รับสิทธิในการควบคุมจากเครื่อง ไลน์แอนท์ผ่านเครือข่ายอินเทอร์เน็ตเข้ามาประมวลผลในโปรแกรมและนำผลตอบสนองจากการควบคุมที่ได้ไปแสดงผลที่เครื่อง ไลน์แอนท์ทุกเครื่อง กระบวนการทั้งการรับค่าพารามิเตอร์และการปรับปรุงข้อมูลผลตอบสนองจากการควบคุมจะกระทำผ่านทางฐานข้อมูลที่ใช้ MS SQL Server เป็นดาต้าเบสเซิร์ฟเวอร์ผ่านคำสั่ง SQL โดยจะมีการติดต่อฐานข้อมูลเพื่อปรับปรุงข้อมูลทุก ๆ 1 วินาที แล้วให้โปรแกรม Java Application ทำหน้าที่เป็นเซิร์ฟเวอร์บนอินเทอร์เน็ตเพื่อรับหรือส่งข้อมูลไปยังเครื่อง ไลน์แอนท์อีกที ดังรูปที่ 6.9



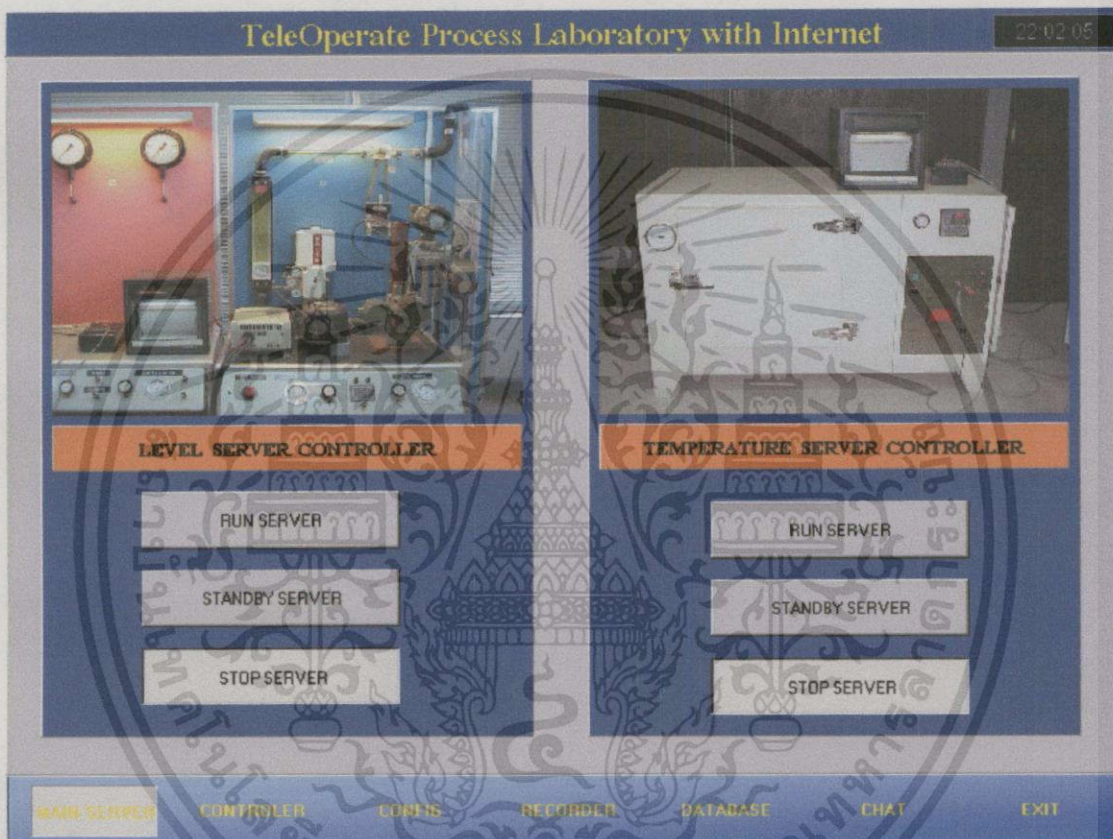
รูปที่ 6.9 แสดง โครงสร้างการทำงานของโปรแกรม Delphi

การทำงานตามโครงสร้างของรูปที่ 6.9 จะเริ่มจากโปรแกรม Delphi อ่านค่าพารามิเตอร์ K_p , T_i , T_d , SP และ Mode (Automatic/Manual) จากฐานข้อมูลหรือโปรแกรม Delphi เพื่อนำใช้ประมวลผลตามการควบคุมของโปรแกรมเครื่องควบคุมพีไอดีและนำผลการตอบสนองจากการควบคุม Pv และ Mv ที่อ่านได้จากอุปกรณ์อินเทอร์เฟซไปบันทึกในฐานข้อมูลเพื่อให้โปรแกรม Java Application นำไปแสดงผลที่เครื่องคอมพิวเตอร์ของผู้ควบคุมอีกที ทั้งนี้การที่โปรแกรมจะเลือกรับค่าพารามิเตอร์และบันทึกผลตอบสนองจากการควบคุมมาจากฐานข้อมูลหรือจากโปรแกรม Delphi นั้นขึ้นอยู่กับทางเลือกโหมดของผู้ควบคุมระบบเป็นผู้เลือกอีกที ซึ่งการแบ่งโหมดการทำงานของระบบเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการ สามารถแบ่งได้ 3 โหมด คือ

1. Run Server เป็นโหมดปกติที่อนุญาตให้ผู้ต้องการควบคุมกระบวนการสามารถควบคุมได้ โดยมีผู้ที่กำลังควบคุมกระบวนการจะสามารถเปลี่ยนแปลงพารามิเตอร์เพื่อควบคุมให้เป็นไปตามที่กำหนด ส่วนผู้ที่ต้องรอลำดับคิวเพื่อที่จะควบคุมกระบวนการจะสามารถเห็นผลการตอบสนองได้เหมือนกับผู้ที่ได้รับสิทธิที่กำลังควบคุมอยู่
2. Standby Server เป็นโหมดที่อนุญาตให้ผู้ต้องการควบคุมกระบวนการสามารถเห็นผลการตอบสนองจากการควบคุมได้เพียงอย่างเดียว โดยผู้ที่มีสิทธิในการควบคุมกระบวนการในโหมดนี้จะเป็นผู้ควบคุมหลักสามารถปรับแต่งค่าพารามิเตอร์เองจากโปรแกรม Delphi ซึ่งจะเข้ามาควบคุมกระบวนการแล้วส่งผลตอบสนองจากการควบคุมไปแสดงผลที่เครื่องคอมพิวเตอร์ของผู้ที่ต้องการควบคุมกระบวนการผ่านโปรแกรม Java Application

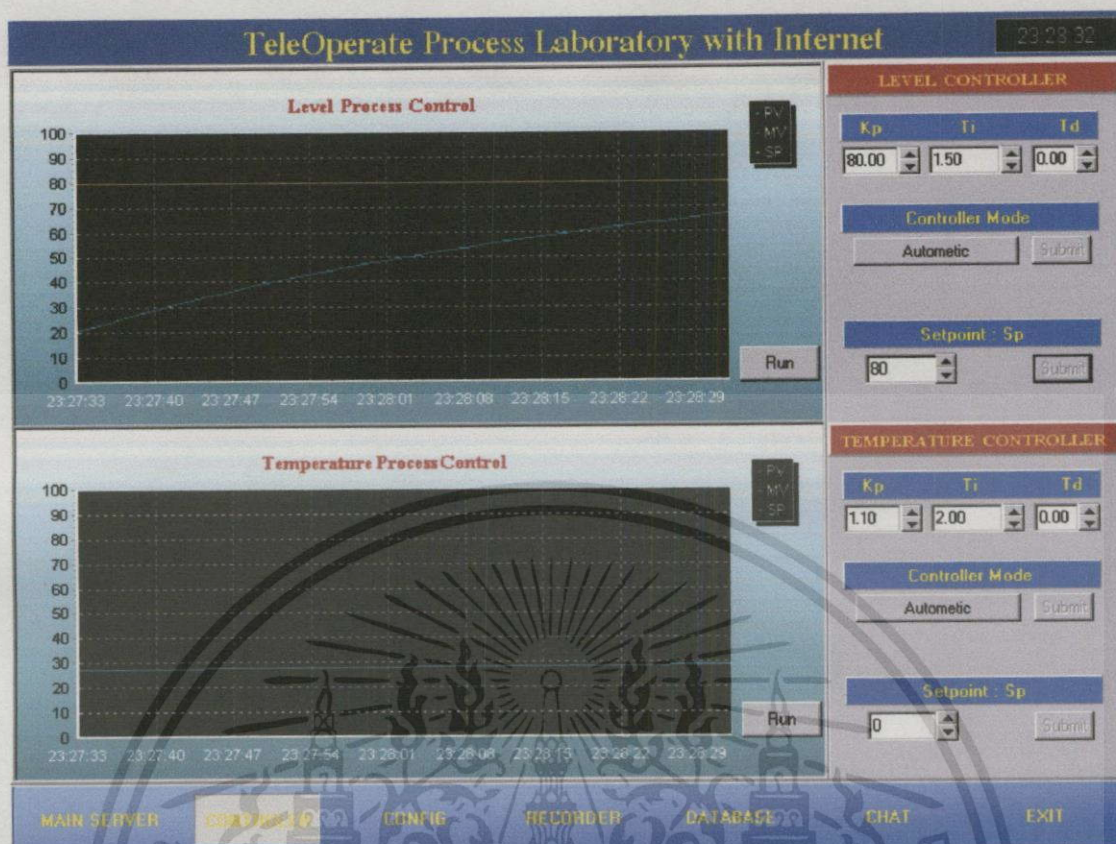
3. Stop Server เป็นโหมดที่ไม่อนุญาตให้มีการควบคุมกระบวนการหรือจองลำดับคิว เพื่อที่จะควบคุมกระบวนการ โดยยกเลิกการติดต่อกับระบบฐานข้อมูลซึ่งโปรแกรม Java Application จะส่งข้อมูลไปแสดงผลที่ไคล์แอนท์ แต่ผู้ควบคุมกระบวนการหลักสามารถทำการควบคุมได้โดยอาศัยโปรแกรม Delphi

โปรแกรมที่พัฒนาด้วยโปรแกรม Delphi ประกอบด้วยหน้าจอหลัก ๆ ดังรูปด้านล่างนี้



รูปที่ 6.10 แสดงรูปหน้าจอ Main Server

หน้าจอ Main Server มีวัตถุประสงค์ในการพัฒนาโปรแกรมเพื่อให้ผู้บริหารระบบสามารถทำการควบคุมโหมดการทำงานของโปรแกรมเซิร์ฟเวอร์เพื่อใช้ควบคุมกระบวนการซึ่งสามารถเลือกที่จะควบคุมทั้งกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิในตู้อบได้พร้อม ๆ กัน ในรูปที่ 6.11 แสดงรูปของหน้าจอที่ใช้ควบคุมกระบวนการ โดยแสดงผลในรูปแบบกราฟเส้นและสามารถปรับแต่งพารามิเตอร์ได้จากหน้าจอของโปรแกรม Delphi

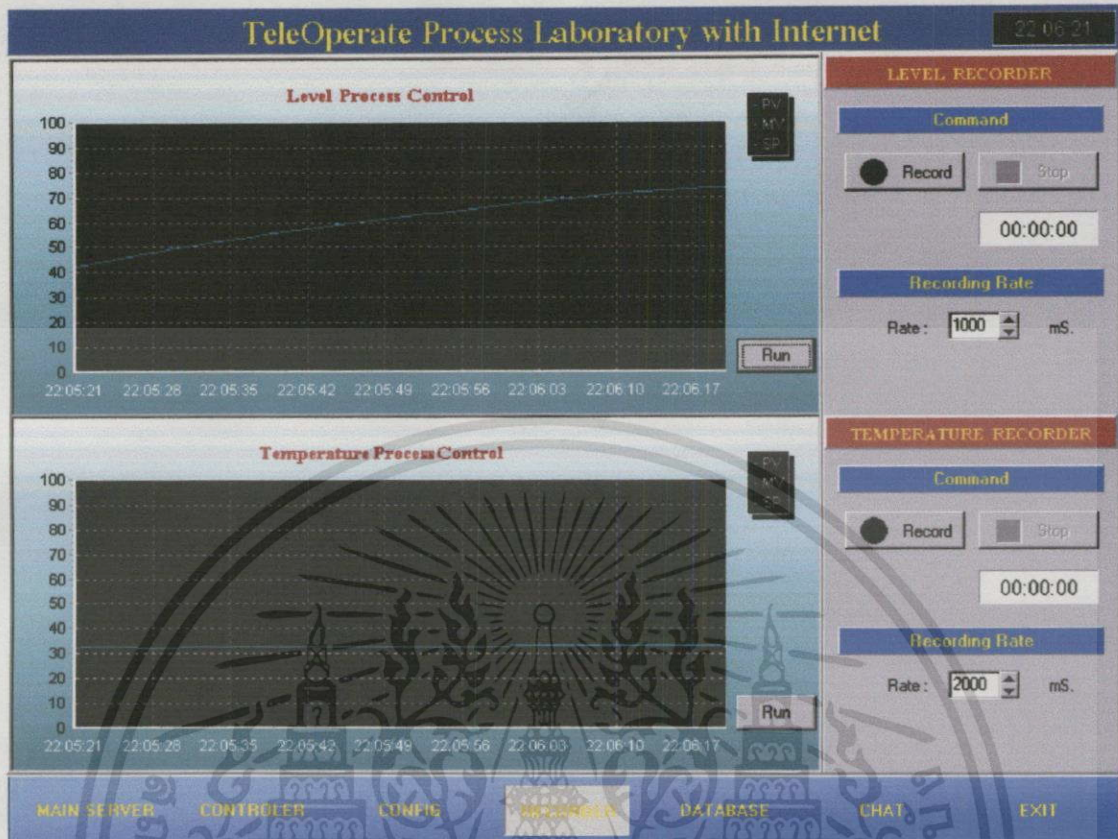


รูปที่ 6.11 แสดงรูปหน้าจอ Controller

ในส่วนของหน้าจอ Controller ถูกออกแบบให้สามารถควบคุมกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิในตู้อบได้พร้อม ๆ กัน โดยการแสดงผลการตอบสนองจากการควบคุมจะอยู่ในรูปกราฟเส้น และผู้ควบคุมสามารถป้อนค่าพารามิเตอร์ได้เพราะในโหมด Standby Server และ Stop Server เท่านั้น เนื่องจากในโหมด Run Server ถูกกำหนดให้ผู้ที่ได้รับสิทธิในการเท่านั้นเป็นผู้ป้อนพารามิเตอร์ต่าง ๆ ผ่านทางเครือข่ายอินเทอร์เน็ต ดังนั้นโปรแกรมจึงทำการล็อกคอมโพเน้นต่าง ๆ ไม่ให้เปลี่ยนแปลงได้

ในรูปที่ 6.12 เป็นส่วนของหน้าจอ Recorder ซึ่งทำหน้าที่บันทึกผลการควบคุมให้มาจัดเก็บในรูปของไฟล์นามสกุล .CSV (Comma Separate Value) เพื่อให้ไคลเอนต์ทำการ Download ผ่านทางอินเทอร์เน็ต โดยไฟล์นาม .CSV สามารถเปิดแล้วแสดงผลในรูปกราฟได้โดยใช้โปรแกรม Microsoft Excel ได้ ค่าความไวในการบันทึกผลการควบคุมนั้นสามารถปรับแต่งได้ มีหน่วยในการบันทึกผลการควบคุมเป็น ms (millisecond) และสามารถบันทึกผลการควบคุมทั้งกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิในตู้อบได้พร้อม ๆ กัน

ในรูปที่ 6.13 เป็นส่วนของ Database ซึ่งในหน้าจอนี้ ถูกพัฒนาเพื่อให้ผู้บริหารระบบสามารถกำหนด เปลี่ยนแปลง หรือปรับปรุง ผู้ที่มีสิทธิที่ได้รับอนุญาตให้สามารถเข้ามาควบคุมกระบวนการในระบบได้



รูปที่ 6.12 แสดงรูปหน้าจอ Recorder

The screenshot displays the 'TeleOperate Process Laboratory with Internet' interface, specifically the 'Database' section. It shows a table of users with columns for NAME, USER ID, and COMMENT. Below the table is a form for user management, including fields for Code, Name, User Name, Password, User Priority (Administrator, Teacher, Student), and Comment. The interface also includes a navigation bar at the bottom with buttons for 'MAIN SERVER', 'CONTROLLER', 'CONFIG', 'RECORDER', 'DATABASE', 'CHAT', and 'EXIT'. The 'DATABASE' button is highlighted.

NAME	USER ID	COMMENT
ริจาร์ณ แก้วพานิช	41054031	นักศึกษาปริญญาตรี
สรเพชรชัย เอื้อแก้ว	42061075	ผู้บริหารระบบ และนักศึกษาริญาโท
สมมติ สุขชนาน	42061074	นักศึกษาปริญญาโท

User Information:
 Code: 42061075
 Name: สรเพชรชัย เอื้อแก้ว
 User Name: sr2061075
 Password: [REDACTED]

User Priority:
 Administrator
 Teacher
 Student

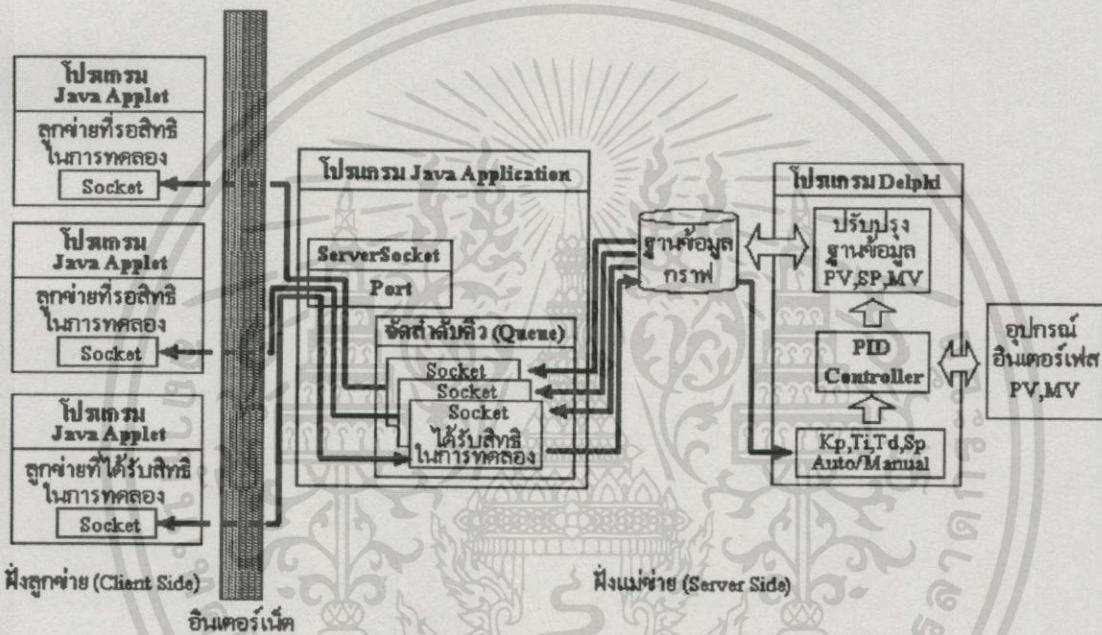
Comment:
 ผู้บริหารระบบ และนักศึกษาริญาโท

รูปที่ 6.13 แสดงรูปหน้าจอ Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6.2 การส่งถ่ายข้อมูลระหว่างเซิร์ฟเวอร์กับไคลด์แอนท์

งานวิจัยได้ใช้โปรแกรมภาษา Java [5,8] โดยพัฒนา Java Application เพื่อให้เป็นโปรแกรมเซิร์ฟเวอร์และ Java Applet เป็นโปรแกรมของผู้ควบคุมหรือไคลด์แอนท์การส่งถ่ายข้อมูลระหว่างเซิร์ฟเวอร์และไคลด์แอนท์จะอาศัย Socket ต่าง ๆ ซึ่งถือเป็นกระบวนการของเครือข่ายที่จะใช้เชื่อมโยงระหว่างเซิร์ฟเวอร์และไคลด์แอนท์ต่าง ๆ โดย Socket จะถูกกำหนดให้สื่อสารบนพอร์ต (Port) ที่กำหนดไว้บนเซิร์ฟเวอร์ ดังรูปที่ 6.14



รูปที่ 6.14 โครงสร้างการส่งถ่ายข้อมูลระหว่างเซิร์ฟเวอร์และไคลด์แอนท์

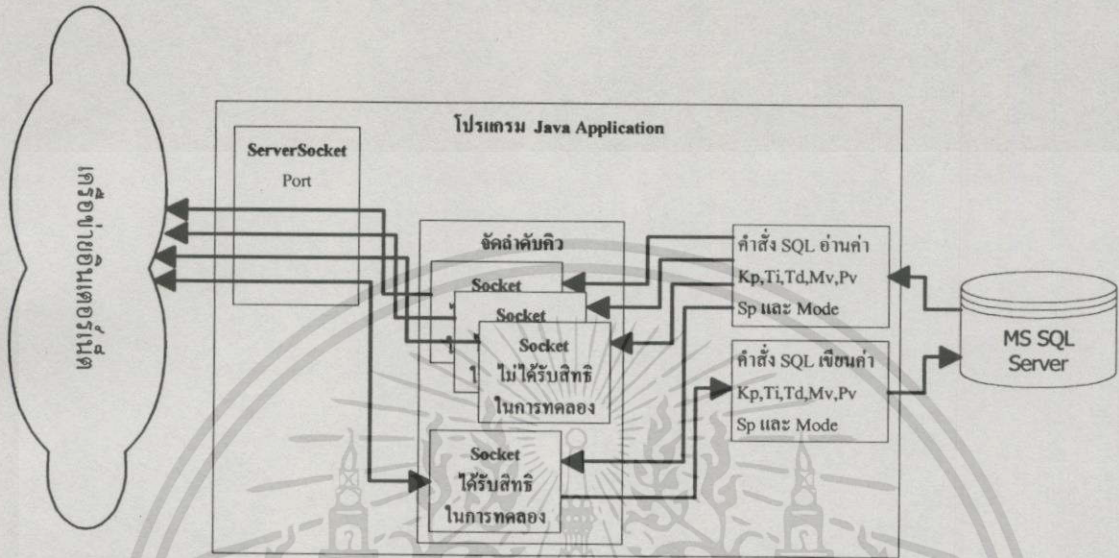
6.6.2.1 การส่งถ่ายข้อมูลของเซิร์ฟเวอร์ด้วยโปรแกรม Java Application

การส่งถ่ายข้อมูลของเซิร์ฟเวอร์จะเป็นการพัฒนาโปรแกรมที่เซิร์ฟเวอร์สามารถเชื่อมต่อกับไคลด์แอนท์ได้หลายเครื่องพร้อม ๆ กัน ซึ่งเซิร์ฟเวอร์จะทำการอ่านค่า K_p , T_i , T_d , MV , PV , SP และ Mode (Automatic หรือ Manual) จากฐานข้อมูล Microsoft SQL Server และทำการส่งข้อมูลที่อ่านได้ไปให้เครื่องไคลด์แอนท์ทุกเครื่องที่อยู่ในตารางคิว

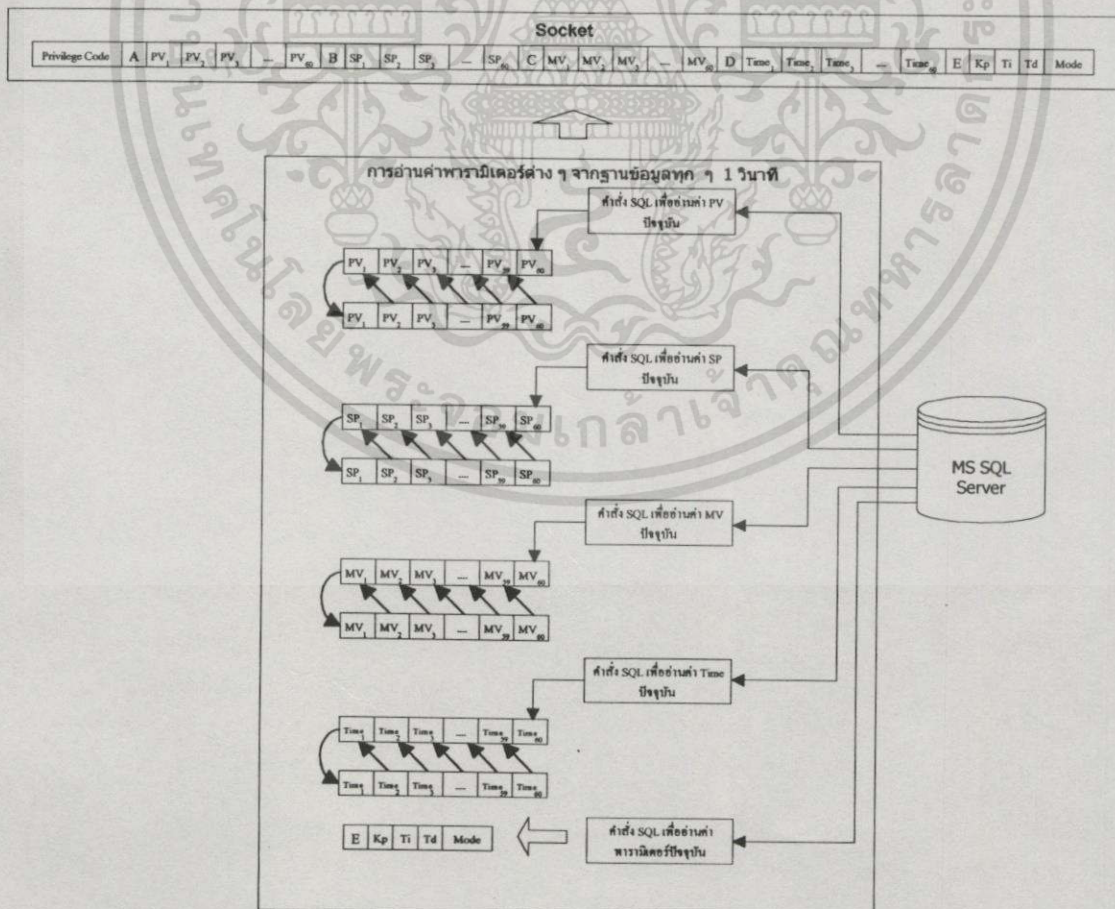
การทำงานของโปรแกรมเริ่มจาก เมื่อไคลด์แอนท์มีการร้องขอการเชื่อมต่อมายังเซิร์ฟเวอร์ และเมื่อ Server Socket รับทราบการร้องขอจากไคลด์แอนท์ดังกล่าวแล้ว เซิร์ฟเวอร์ก็จะยอมรับการเชื่อมต่อกับเครื่องไคลด์แอนท์พร้อมทำการกำหนดการเชื่อมต่อกับไคลด์แอนท์ดังกล่าว ไปสู่เซรด์ (Thread) ใหม่เซรด์หนึ่งซึ่งทำหน้าที่รับข้อมูลค่าพารามิเตอร์ต่าง ๆ มาทำการรวมกันเป็นกลุ่มข้อความกลุ่มหนึ่งแล้วส่งออกไปยังไคลด์แอนท์ของตัวเองที่เซรด์นั้น ๆ เชื่อมต่อกับไคลด์แอนท์ที่อยู่ ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.15 แสดงการทำงานของโปรแกรม Java Application ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ และรูปที่ 6.16 เป็นโครงสร้างการรวมกลุ่มข้อความเพื่อส่งพารามิเตอร์ไปยังเครื่องไคลเอนท์



รูปที่ 6.15 แสดงการทำงานของโปรแกรม Java Application ที่ทำหน้าที่เป็นเซิร์ฟเวอร์



รูปที่ 6.16 แสดงโครงสร้างการรวมกลุ่มข้อความเพื่อส่งพารามิเตอร์ไปยังเครื่องไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้มาใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างการรวมกลุ่มข้อความจะเป็นการกำหนดวิธีการเพื่อส่งข้อมูลไปยังไคล์แอนท์ภายในครั้งเดียวทุก ๆ 1 วินาที โดยข้อมูลที่ส่งคือ PrivateCode , PV จำนวน 60 ค่า , SP จำนวน 60 ค่า MV จำนวน 60 ค่า , Time จำนวน 60 ค่า , Kp , Ti , Td และ Mode โดยตัวอักษร A , B , C , D และ E เป็นตัวอักษรที่ใช้ในการแบ่งแยกประเภทของข้อมูลที่ส่งไปยังไคล์แอนท์ ข้อมูลที่ส่งทั้งหมด 60 ค่า หมายถึง การที่แม่ข่ายส่งข้อมูลจำนวน 60 วินาที ซึ่งประกอบด้วยวินาทีปัจจุบัน 1 วินาทีและ 59 วินาทีที่ซ้อนหลังไป ดังนั้นไคล์แอนท์จะได้รับข้อมูลที่ถูกต้องทุกครั้งทีจากเซิร์ฟเวอร์

6.6.2.2 การรับข้อมูลของไคล์แอนท์ด้วยโปรแกรม Java Applet

เมื่อไคล์แอนท์ได้รับอนุญาตให้เชื่อมต่อกับเซิร์ฟเวอร์ได้แล้ว เซิร์ฟเวอร์ก็จะทำการส่งข้อมูลมาให้ไคล์แอนท์ทุก ๆ 1 วินาที แต่ไคล์แอนท์จะทำการอ่านข้อมูลจาก Socket ทุก ๆ 500 ms. เนื่องจากป้องกันกรณีที่มีข้อมูลที่ขังไม่ได้อ่านค้างอยู่ในบัฟเฟอร์อินพุตสตรีม ซึ่งจะทำให้เกิดปัญหาในการแสดงผลที่คาดเคลื่อนกับข้อมูลปัจจุบันและจะสะสมมากขึ้นเรื่อย รูปที่ 6.17 แสดงโครงสร้างการทำงานของไคล์แอนท์ของโปรแกรม Java Applet

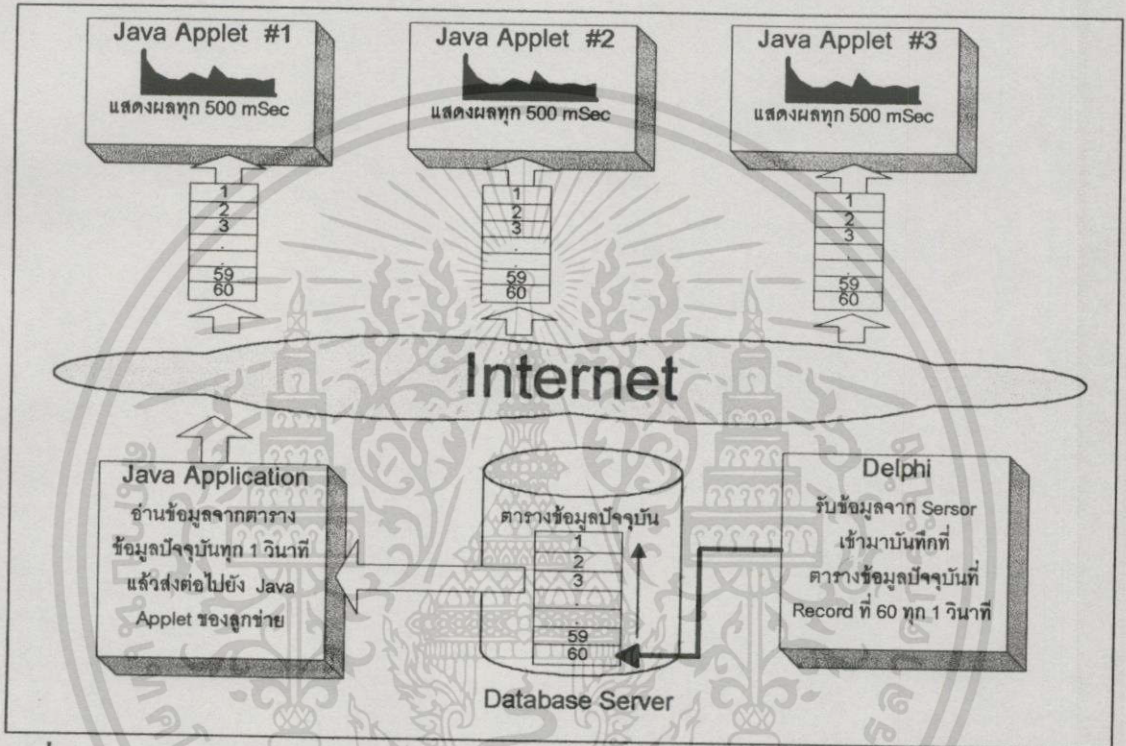


รูปที่ 6.17 แสดงโครงสร้างการทำงานของไคล์แอนท์ของโปรแกรม Java Applet

6.6.3 การออกแบบฐานข้อมูลเพื่อแสดงผลกราฟ

เพื่อให้เกิดความเที่ยงตรงในการแสดงผลกราฟ จึงได้ออกแบบให้ฐานข้อมูล[7]ให้ฟิลด์ (field) ประกอบด้วย Time, PV, MV และ SP เพื่อเป็นข้อมูลในแกน Y ของกราฟ ส่วนเรคอร์ด (record) ได้ออกแบบให้มี 60 เรคอร์ด ซึ่งแทนข้อมูลจำนวน 60 วินาทีในแกน X ของกราฟ การคำนวณค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลจะถูกปรับปรุง(update)ทุก ๆ วินาทีจากโปรแกรมภาษา Delphi [7] ซึ่ง Delphi ถูกพัฒนาเพื่อทำหน้าที่อินเทอร์เน็ตเฟสกับอุปกรณ์ภายนอกเพื่อมาคำนวณตามโปรแกรมควบคุมที่ไอดี จากนั้นจึงปรับปรุงข้อมูลลงในฐานข้อมูล โปรแกรม Java Application จะส่งข้อมูลในฐานข้อมูลทั้งหมดไปให้กับผู้ควบคุมหรือไคล์แอนท์ทุกเครื่อง ข้อมูลที่ส่งไปจะถูกนำไปแสดงผลในรูปภาพเส้นจากโปรแกรม Java Applet ที่ไคล์แอนท์อีกที

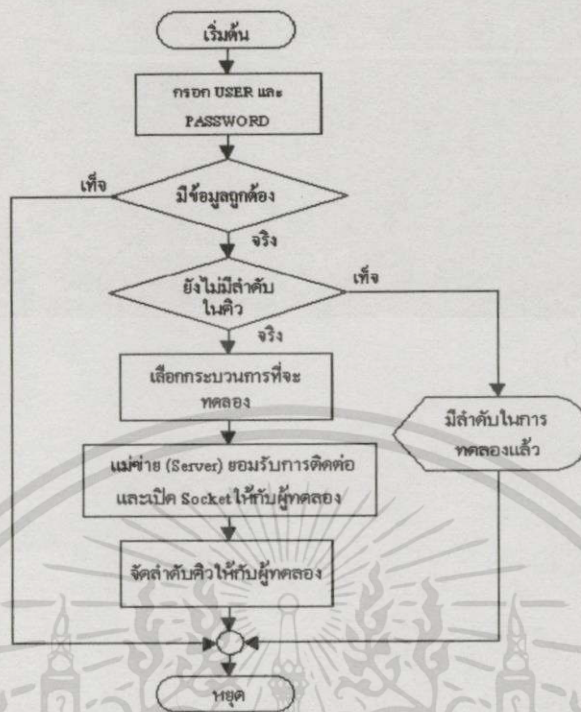


รูปที่ 6.18 โครงสร้างในการปรับปรุงฐานข้อมูลเพื่อแสดงผลจากเซิร์ฟเวอร์ไปยังไคล์แอนท์

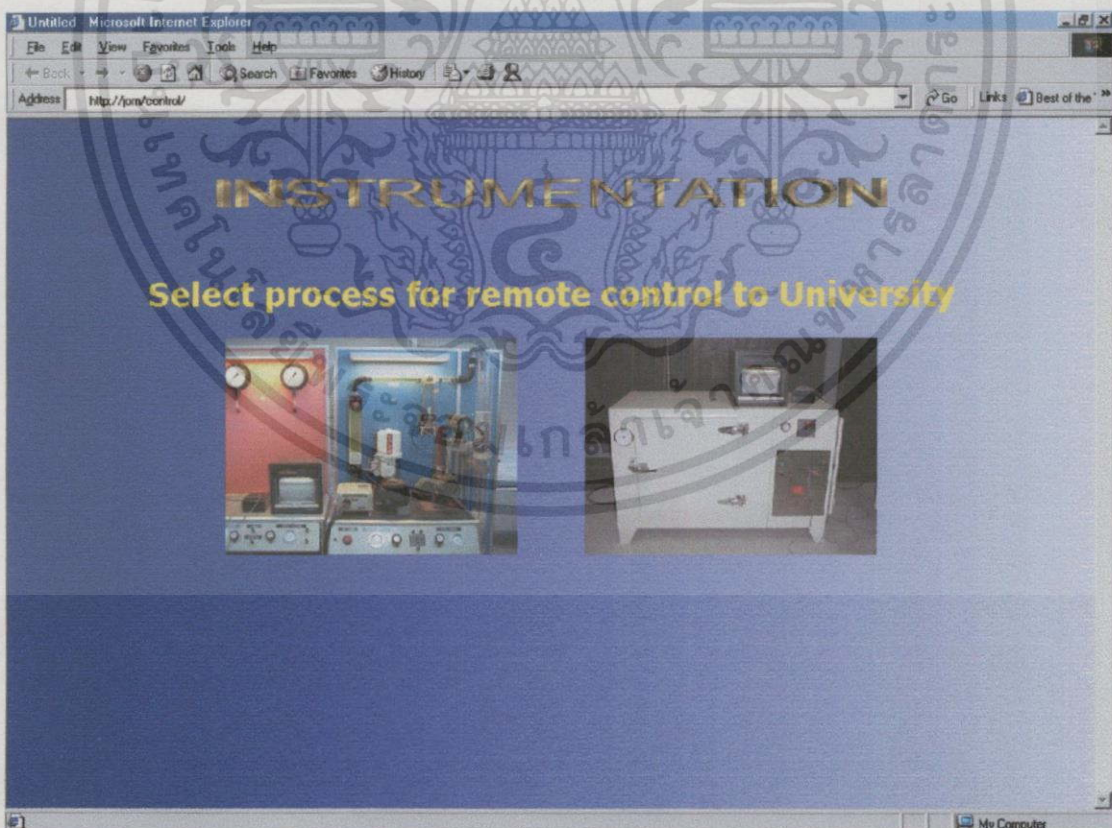
6.6.4 โปรแกรมเพิ่มคิวของผู้ที่ต้องการควบคุม

ผู้ที่ควบคุมผ่านอินเทอร์เน็ตนี้ต้องผ่านการ Login ก่อนเสมอ ถ้าเป็นผู้ที่ได้รับอนุญาตให้ควบคุมได้ (มีรายชื่ออยู่ในฐานข้อมูลของผู้บริหารระบบ)และการกรอกรหัสผ่านถูกต้อง โปรแกรมก็จะทำการสร้างรหัส PrivateCode จำนวน 1 ชุดซึ่งประกอบด้วยตัวเลขจำนวน 5 หลัก ซึ่งจะเก็บไว้ที่โปรแกรมผู้ควบคุมเองและส่งไปเก็บไว้ในตารางคิวเพื่อใช้ในการจองคิวและตรวจสอบสิทธิในการควบคุมที่เซิร์ฟเวอร์ด้วย หลังจากกรอกรหัสผ่านแล้วจะต้องเลือกกระบวนการที่ต้องการควบคุมระหว่างกระบวนการควบคุมระดับน้ำและควบคุมอุณหภูมิ จากนั้นโปรแกรมเซิร์ฟเวอร์ก็จะส่งข้อมูลเพื่อไปแสดงผลและมีการจัดลำดับคิว[6]ให้กับผู้ควบคุม โปรแกรมที่ทำการเพิ่มคิวนั้นได้พัฒนาด้วยโปรแกรม ASP โดยมีขั้นตอนในการเพิ่มคิวตาม FlowChart ในรูปที่ 6.19 และรูปที่ 6.20 ถึงรูปที่ 6.22 แสดงหน้าจอขณะทำการควบคุมกระบวนการผ่านทางอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

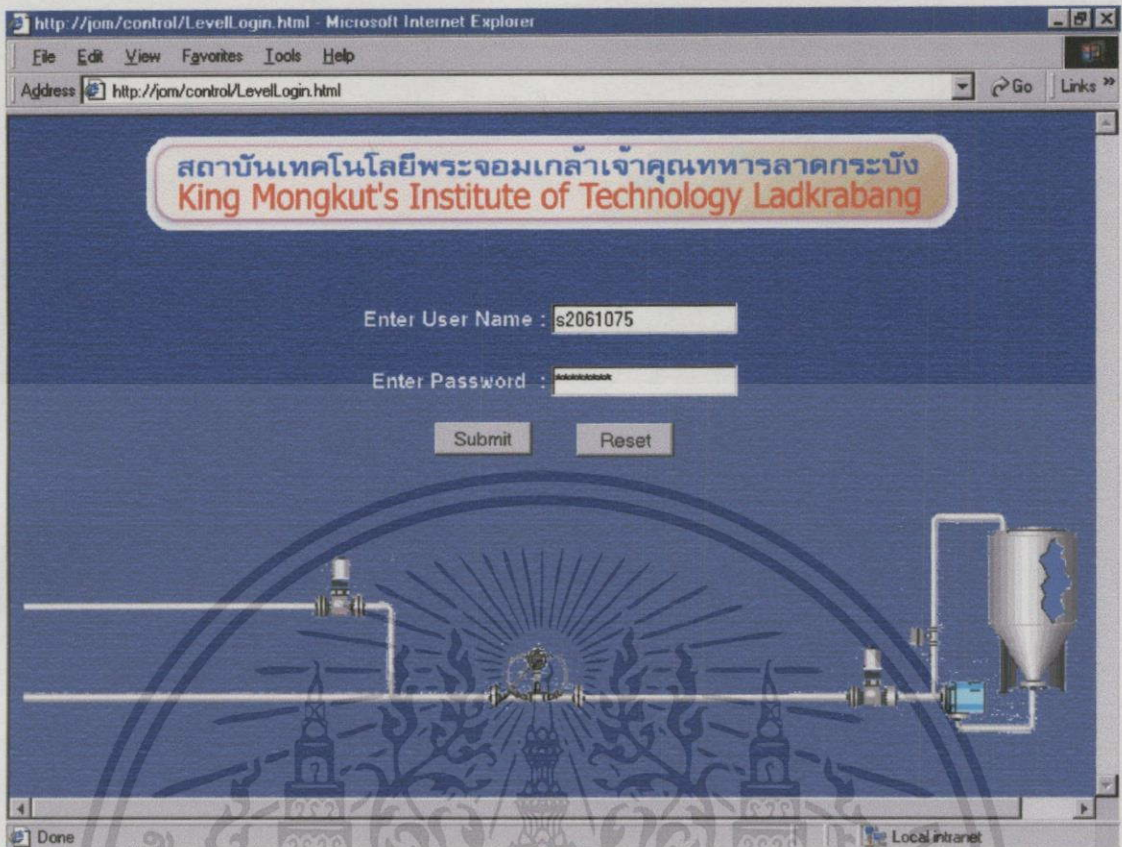


รูปที่ 6.19 FlowChart ของการเพิ่มคิวของผู้ควบคุม

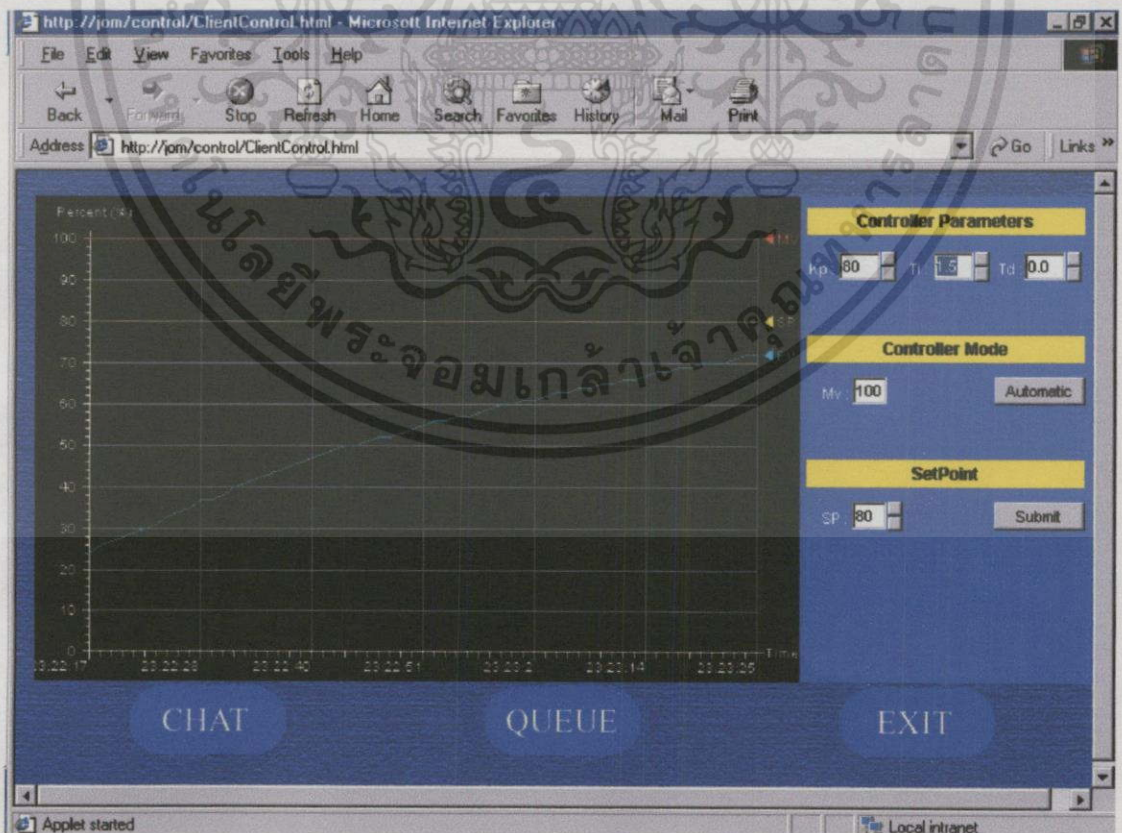


รูปที่ 6.20 แสดงหน้าจอสำหรับเลือกกระบวนการที่ต้องการควบคุมระหว่างกระบวนการควบคุมระดับน้ำและควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.21 แสดงหน้าจอลำหรับกรอกรหัสผ่านเพื่อควบคุมกระบวนการผ่านทางอินเทอร์เน็ต

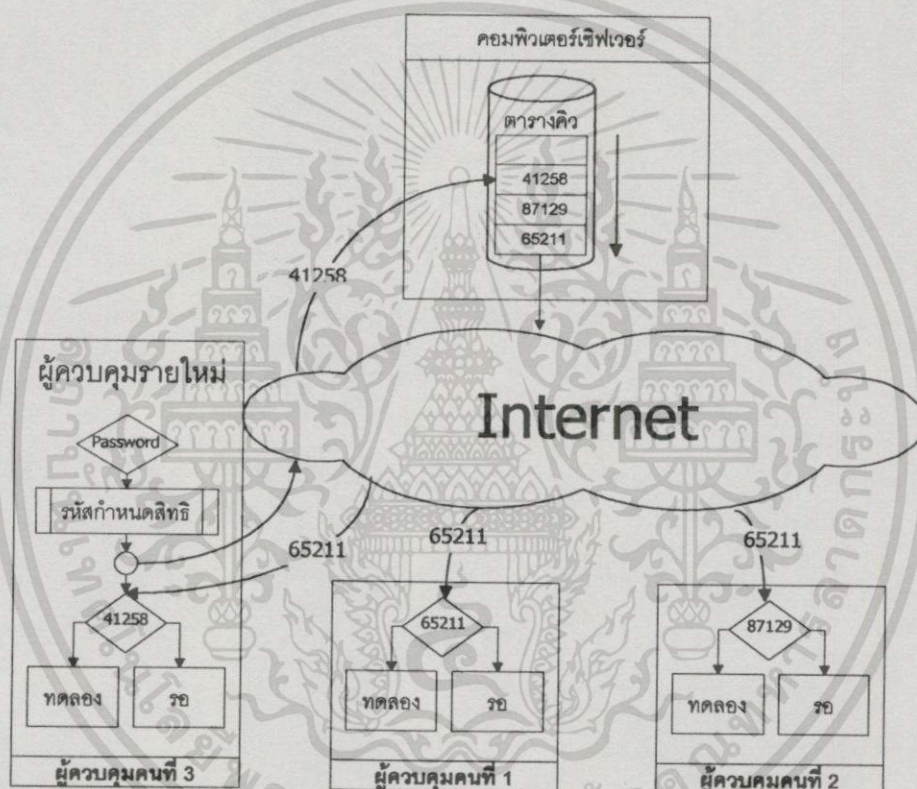


รูปที่ 6.22 แสดงหน้าจอขณะทำการควบคุมกระบวนการผ่านทางอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6.5 โปรแกรมตรวจสอบคิวเพื่อให้สิทธิในการควบคุม

นอกจากคอมพิวเตอร์เซิร์ฟเวอร์ทำงานเพื่อส่งข้อมูลกลับไปแสดงผลที่หน้าจอของผู้ควบคุมแล้ว เซิร์ฟเวอร์ยังส่งหมายเลข PrivateCode และ IP address [5,8] ของผู้ที่ได้รับสิทธิในการควบคุม ปัจจุบันออกไปด้วย โดยโปรแกรม Java Applet ของผู้ควบคุมแต่ละคนก็จะทำการเปรียบเทียบกับ PrivateCode และเมื่อรหัสตรงกัน โปรแกรม Java Applet ก็จะบอกให้ผู้ควบคุมท่านนั้นทราบว่าได้รับสิทธิในการควบคุมแล้ว พร้อมทั้งปรับปรุงหน้าจอควบคุมให้สามารถปรับเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ในการควบคุมกระบวนการได้



รูปที่ 6.23 แสดงตัวอย่างการตรวจสอบคิวเพื่อให้สิทธิในการควบคุม

จากรูปที่ 6.23 แสดงตัวอย่างการตรวจสอบคิวเพื่อให้สิทธิในการควบคุม เซิร์ฟเวอร์ได้ส่ง PrivateCode ไปยังไคลเอนต์ทุกเครื่องคือหมายเลข 65211 โดยไคลเอนต์แต่ละเครื่องจะทำการเปรียบเทียบกับ PrivateCode ของตัวเองและถ้ารหัสตรงกัน ไคลเอนต์รายนั้นก็จะถูกอนุญาตให้ควบคุมได้ ซึ่งจากตัวอย่างผู้ควบคุมคนที่ 1 มีรหัสตรงกับรหัสที่เซิร์ฟเวอร์ส่งมาผู้ควบคุมรายนี้จึงมีสิทธิควบคุม และตารางคิวของเซิร์ฟเวอร์ PrivateCode รหัส 87129 เป็นรหัสที่ตรงกับผู้ควบคุมคนที่ 2 ซึ่งจะเป็นที่ที่ได้รับสิทธิในการควบคุมรายต่อไป และผู้ควบคุมคนที่ 3 ซึ่งเป็นผู้ที่เพิ่ง Login เข้าสู่ระบบซึ่งจะเป็นผู้ที่ได้รับสิทธิควบคุมรายสุดท้ายหลังผู้ควบคุมรายที่ 1 และรายที่ 2 ทำการควบคุมกระบวนการเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6.6 โปรแกรมสำหรับยกเลิกการควบคุม

การที่ผู้ที่ได้รับสิทธิในการควบคุมหรือกำลังรอสิทธิในการควบคุมต้องการยกเลิกการควบคุมนั้นสามารถทำได้ดังนี้

1. ผู้ควบคุมที่ได้รับสิทธิในการควบคุมทุกคนจะถูกจำกัดเวลาในการควบคุมทุกครั้งและเมื่อเวลาในการควบคุมสิ้นสุดลงโปรแกรมจะทำการยกเลิกการควบคุมผู้ควบคุมรายนั้นแล้วเลื่อนลำดับคิวเป็นผู้ควบคุมรายถัดไป อาศัยโปรแกรม Java Applet ทำการตรวจสอบเวลาที่เบราว์เซอร์ของผู้ควบคุมแต่ละรายเอง ในการยกเลิกการควบคุมสำหรับกรณีนี้ ระยะเวลาที่ให้ในการควบคุมจะกำหนดโดยผู้บริหารระบบขึ้นอยู่กับความเหมาะสมของแต่ละกระบวนการ
2. ยกเลิกการควบคุมในกรณีที่ผู้ควบคุมต้องการยกเลิกการควบคุมก่อนหมดเวลาที่อนุญาตให้ควบคุมได้ผ่านทางกรกดปุ่ม Logout ที่เว็บเพจซึ่งจะเป็นการปิดเบราว์เซอร์ตัวเองโดยอัตโนมัติ โดยอาศัยโปรแกรม ASP และ Java Script
3. การที่ผู้ใช้ปิดเบราว์เซอร์หรือการสื่อสารบนอินเทอร์เน็ตระหว่างเซิร์ฟเวอร์กับไคล์แอนท์ขาดการติดต่อ โปรแกรม Java Application ก็จะทำการยกเลิกการควบคุมและปรับปรุงตารางคิวเช่นกัน

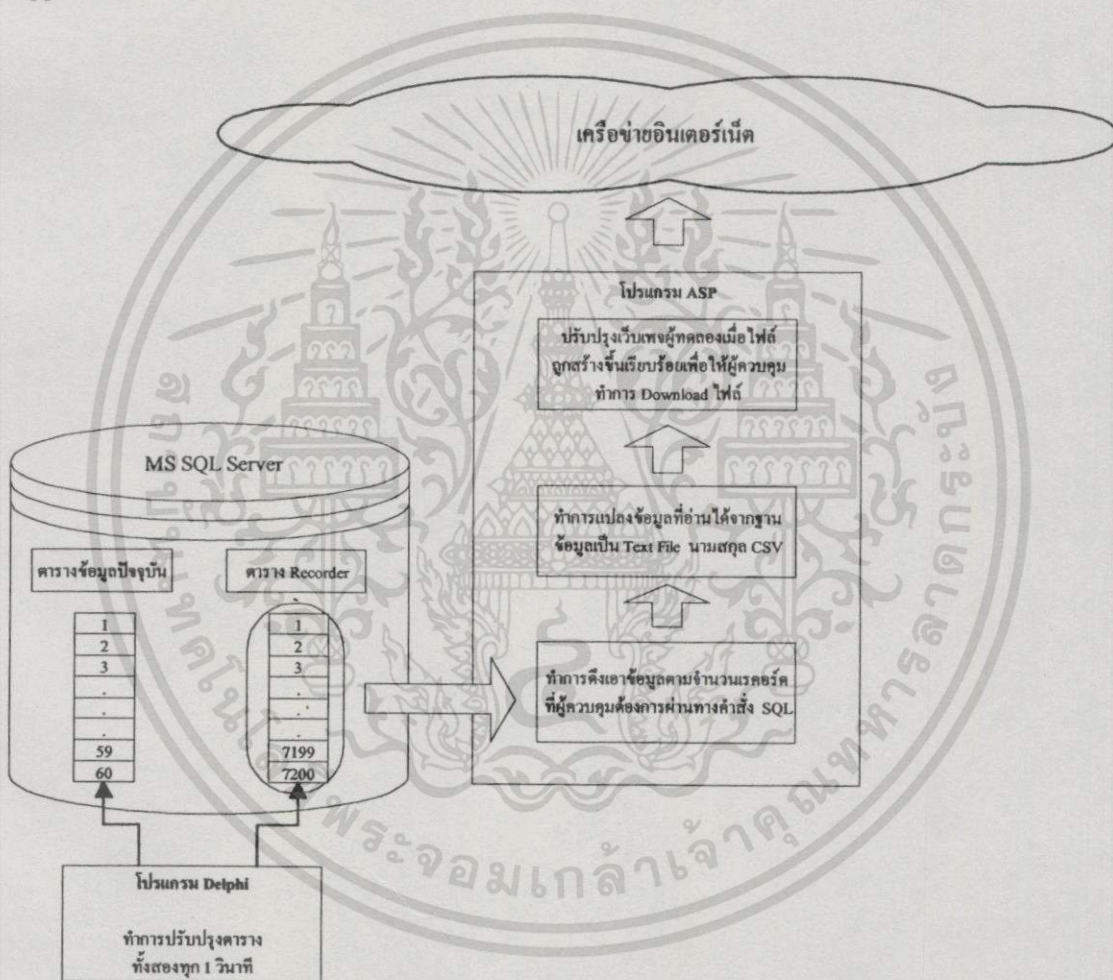
หลังจากที่โปรแกรม Java Application ไม่สามารถติดต่อกับเครื่องไคล์แอนท์ได้ โปรแกรมก็จะทำการยกเลิกเซรคและ Socket ของผู้ขาดการติดต่อยานั้นเองโดยอัตโนมัติ จากนั้นโปรแกรม Java Application จะทำการลบรหัสของผู้ควบคุมรายนั้นในตารางคิว ซึ่งจะทำให้ผู้ควบคุมรายที่อยู่ถัดไปเลื่อนลำดับขึ้นมาแทน

6.6.7 โปรแกรมสำหรับบันทึกผลการควบคุม

โปรแกรมบันทึกผลการควบคุมเปรียบเหมือนเครื่อง Recorder ที่ติดตั้งอยู่ที่กระบวนการจริง แต่ในการควบคุมผ่านทางอินเทอร์เน็ตจึงไม่สามารถใช้เครื่อง Recorder ได้ ดังนั้นในงานวิทยานิพนธ์ชิ้นนี้ ได้ออกแบบให้มีตาราง Recorder ในฐานข้อมูลเพื่อทำการบันทึกข้อมูลทุก ๆ 1 วินาทีเหมือนตารางบันทึกข้อมูลปัจจุบันเพื่อแสดงผลในกราฟของไคล์แอนท์ซึ่งตารางบันทึกข้อมูลปัจจุบันถูกกำหนดให้มีการเก็บข้อมูลจำนวน 60 เรคอร์ดแทนจำนวนข้อมูลต่าง ๆ จำนวน 60 วินาที แต่สำหรับตาราง Recorder ได้ถูกออกแบบให้มี 7,200 เรคอร์ดแทนจำนวนข้อมูลที่บันทึก 7,200 วินาทีหรือ 2 ชั่วโมง ซึ่งข้อมูลจะถูกปรับปรุงที่เรคอร์ดสุดท้าย (เรคอร์ดที่ 7200) ทุก ๆ 1 วินาทีและมีการเลื่อนเรคอร์ดขึ้นไป 1 เรคอร์ดในการบันทึกข้อมูลและปรับปรุงตามกระบวนการที่กำหนดจะอาศัยโปรแกรม Delphi ซึ่งกระทำควบคู่ไปกับการปรับปรุงข้อมูลตารางข้อมูลปัจจุบัน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การเขียนเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลระบบใช้ระบบนี้ขึ้นมาก็ไม่ควรนำเอกสารนี้ไปใช้

กรณีที่คุณต้องการเรียกดูผลการควบคุมผ่านทางอินเทอร์เน็ต ผู้ควบคุมเพียงกำหนดช่วงเวลาที่ต้องการดูผลการควบคุมย้อนหลังจากเวลาปัจจุบันเป็นเวลากี่วินาที เครื่องเซิร์ฟเวอร์จะทำการอ่านข้อมูลในฐานข้อมูลในช่วงเวลาที่ผู้ควบคุมกำหนดผ่านทางคำสั่ง SQL แล้วแปลงข้อมูลที่อ่านได้จากฐานข้อมูลมาบันทึกใหม่ในรูปแบบที่ Text File นามสกุล CSV เพื่อให้ผู้ควบคุมรายนั้นสามารถทำการ Download ผ่านทางเครือข่ายอินเทอร์เน็ตได้ โปรแกรมสำหรับเรียกดูผลการควบคุมนั้นได้พัฒนาโดยอาศัยโปรแกรม ASP ซึ่งสามารถกระทำได้สะดวกรวดเร็วกว่าการใช้โปรแกรม Java Application



รูปที่ 6.24 แสดงโครงสร้างของโปรแกรมสำหรับบันทึกผลการควบคุม

6.7 บทสรุป

ในการพัฒนาโปรแกรมในวิทยานิพนธ์นี้ในส่วนของกรออกแบบโปรแกรมทั้งในฝั่งไคลน์แอนท์ที่ได้พัฒนาโดยอาศัยโปรแกรม Java Applet ซึ่งทำหน้าที่รับข้อมูลจากเซิร์ฟเวอร์มาแสดงผลบนหน้าจอของผู้ควบคุมพร้อมทั้งทำหน้าที่ได้ตอบรับค่าพารามิเตอร์จากผู้ควบคุมส่งไปยังเครื่องเซิร์ฟเวอร์และในส่วนเซิร์ฟเวอร์ซึ่งจะเป็นการสื่อสารข้อมูลบนอินเทอร์เน็ตไปยังไคลน์แอนท์หลาย ๆ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ณาไปเซิร์ฟเวอร์เช่นตามการคำไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรวมถึงการจัดลำดับสิทธิในการควบคุมกระบวนการผ่านอินเทอร์เน็ต โดยพัฒนาจากโปรแกรม Java Application นอกจากนี้โปรแกรม Java Application ยังทำหน้าที่ติดต่อกับฐานข้อมูลโดยอาศัย JDBC เพื่ออ่านและเขียนพารามิเตอร์ที่ผู้ควบคุมป้อนมาจากเครื่องไคล์แอนท์ไปยังฐานข้อมูล โดยมีโปรแกรม Delphi ทำหน้าที่เป็นโปรแกรมควบคุมกระบวนการแบบพีไอดีซึ่งจะอ่านค่าพารามิเตอร์มาจากฐานข้อมูลที่ Java Application เขียนค่าลงมาแล้วทำการประมวลตามสมการพีไอดีแล้วส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกจากนั้นจึงรับสัญญาณจากอุปกรณ์ภายนอกเข้ามาแล้วทำการเขียนค่าต่าง ๆ ลงในฐานข้อมูลเพื่อให้โปรแกรม Java Application ทำการอ่านแล้วส่งไปแสดงผลยังเครื่องไคล์แอนท์อีกที

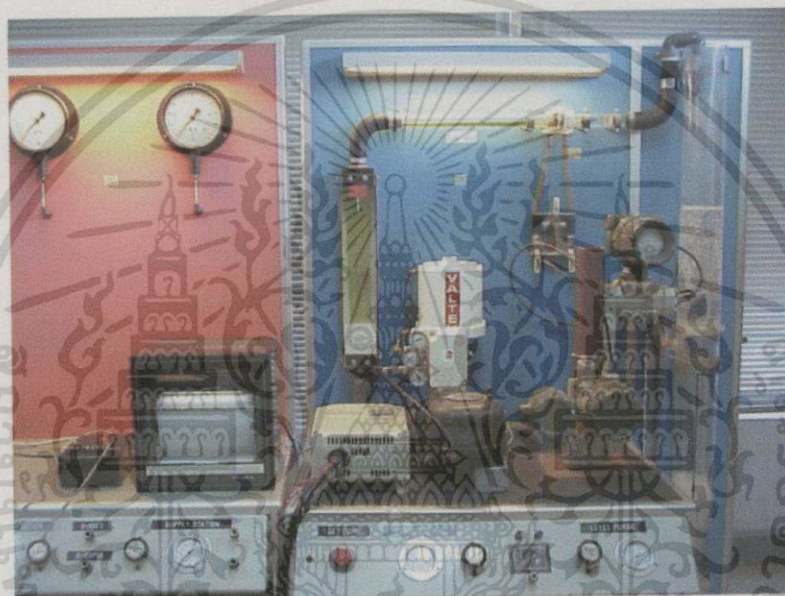


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

กระบวนการที่พัฒนาโปรแกรมเพื่อทำการควบคุมกระบวนการผ่านทางเครือข่ายอินเทอร์เน็ตได้ประกอบด้วยกระบวนการ 2 กระบวนการ คือ กระบวนการควบคุมระดับน้ำ ดังรูปที่ 7.1 และกระบวนการควบคุมอุณหภูมิในตู้อบ รูปที่ 7.2



รูปที่ 7.1 แสดงชุดทดลองกระบวนการควบคุมระดับน้ำ

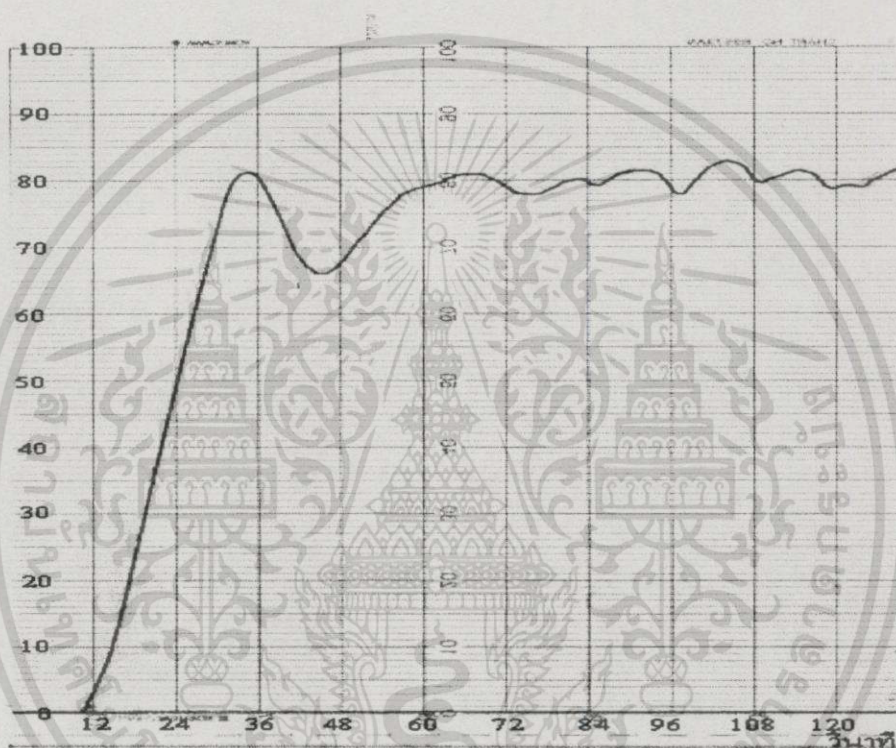


รูปที่ 7.2 แสดงชุดทดลองกระบวนการควบคุมอุณหภูมิในตู้อบ

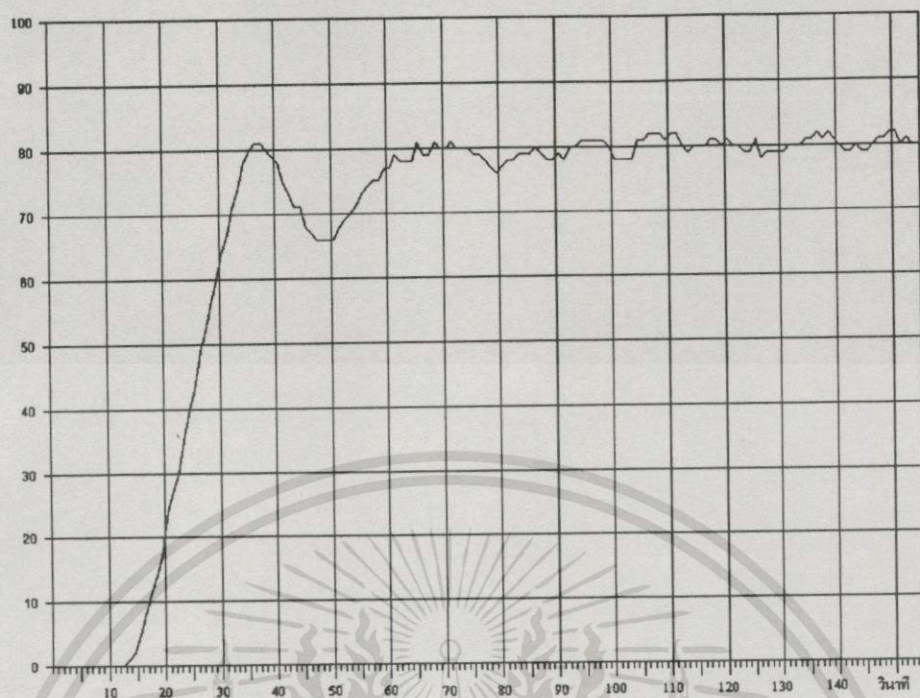
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1 การควบคุมกระบวนการควบคุมระดับน้ำ

การทดลองทำโดยการบันทึกผลการควบคุมกระบวนการจากเบรเซอร์ของผู้ควบคุมและที่เซิร์ฟเวอร์เองก็ได้บันทึกผลของตัวแปรกระบวนการไว้โดยใช้เครื่อง Recorder ดังรูปที่ 7.3 แสดงกราฟของการควบคุมระดับน้ำโดยใช้ Recorder ซึ่งติดตั้งอยู่กับชุดควบคุมระดับน้ำและรูปที่ 7.4 แสดงผลการบันทึกกระบวนการเดียวกันผ่านทางเครือข่ายอินเทอร์เน็ต ไฟล์ที่บันทึกมีนามสกุล CSV ซึ่งสามารถเปิดเพื่อแสดงผลกราฟได้โดยอาศัยโปรแกรม Microsoft Excel



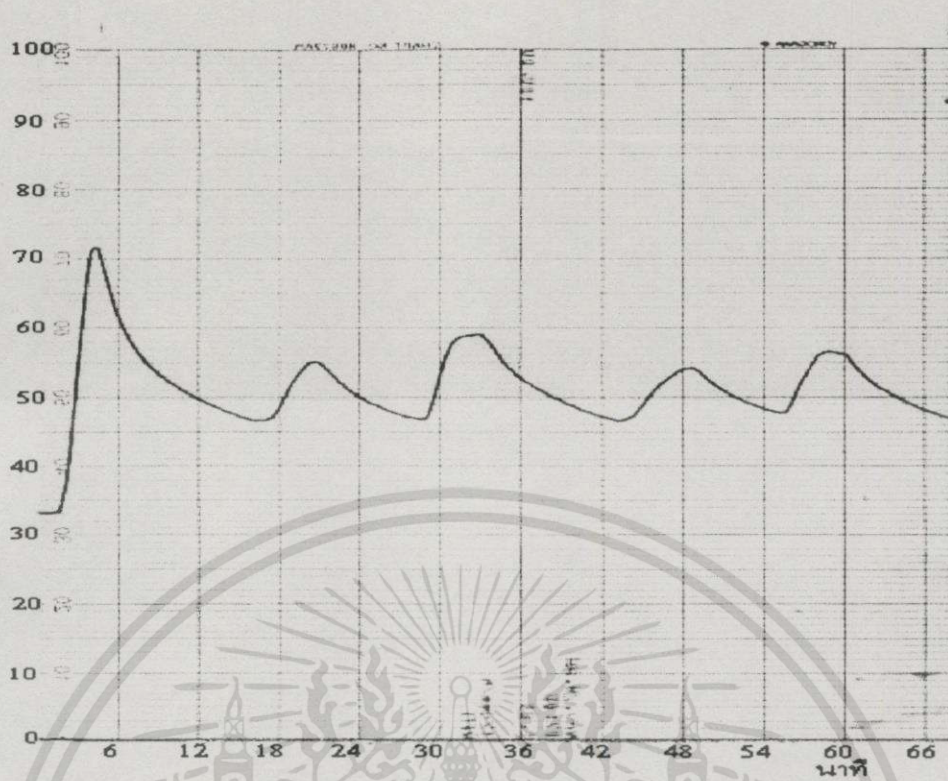
รูปที่ 7.3 แสดงผลการบันทึกตัวแปรกระบวนการจากเครื่อง Recorder ของการควบคุมระดับน้ำ



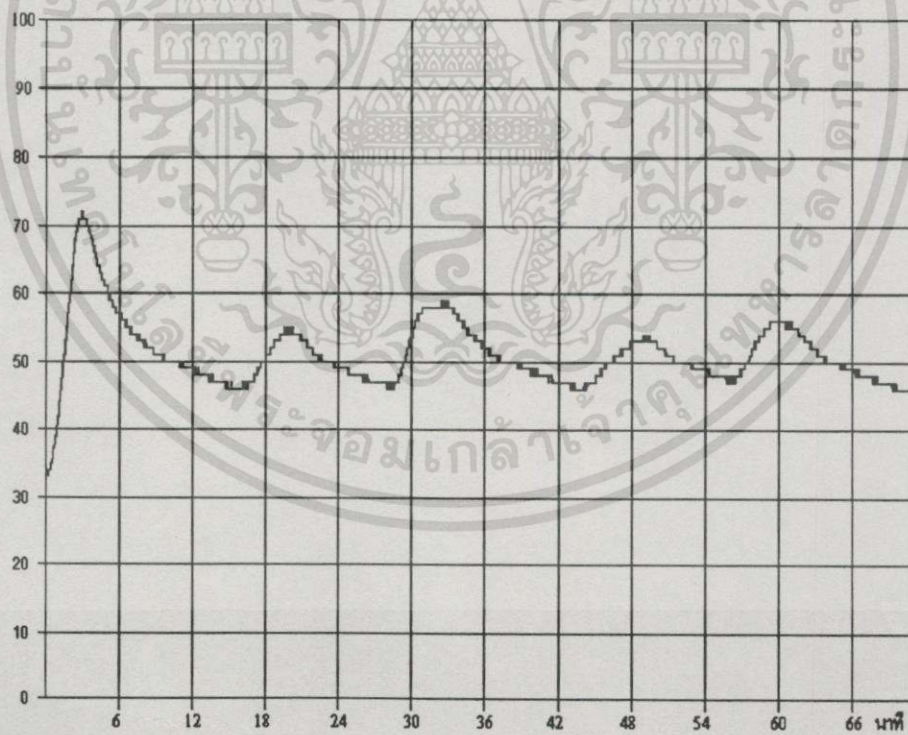
รูปที่ 7.4 แสดงผลการบันทึกกระบวนการควบคุมระดับน้ำผ่านทางเครือข่ายอินเทอร์เน็ต

7.2 การควบคุมกระบวนการควบคุมอุณหภูมิในตู้อบ

การทดลองทำโดยการบันทึกผลการควบคุมกระบวนการจากเบราเซอ์ของผู้ควบคุมและที่เซฟเวอร์เองก็ได้บันทึกผลของตัวแปรกระบวนการไว้โดยใช้เครื่อง Recorder ดังรูปที่ 7.5 แสดงกราฟของการควบคุมอุณหภูมิโดยใช้ Recorder ซึ่งติดตั้งอยู่กับชุดควบคุมกระบวนการอุณหภูมิและรูปที่ 7.6 แสดงผลการบันทึกกระบวนการเดียวกันผ่านทางเครือข่ายอินเทอร์เน็ต



รูปที่ 7.5 แสดงผลการบันทึกตัวแปรกระบวนการจากเครื่อง Recorder ของการควบคุมอุณหภูมิ



รูปที่ 7.6 แสดงผลการบันทึกกระบวนการควบคุมอุณหภูมิจากหน้าเว็บเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปผลการวิจัยและข้อเสนอแนะ

การควบคุมกระบวนการผ่านทางอินเทอร์เน็ตโดยอาศัยเบราเซอร์ทำให้การที่จะควบคุมสามารถทำได้ง่ายไม่จำเป็นต้องใช้โปรแกรมเฉพาะใด ๆ เข้ามาช่วยเพื่อการควบคุม การบันทึกผลการควบคุม(recorder) กระบวนการดังกล่าว สามารถทำได้จากหน้าจอของผู้ควบคุม ซึ่งจะลดภาระการทำงานของเซิร์ฟเวอร์ได้ การแสดงผลการควบคุมในกราฟจะใช้วิธีการดึงข้อมูลต่าง ๆ ที่ได้จาก การควบคุมทั้งหมดในช่วงเวลา 60 วินาทีจากฐานข้อมูลที่เซิร์ฟเวอร์มาแสดง ข้อมูลที่ได้จากการควบคุมจะถูกบันทึกจากเครื่องควบคุมที่ตั้งในห้องปฏิบัติการทุก ๆ วินาทีซึ่งไม่ได้นำเอาเฉพาะข้อมูลปัจจุบันมาแสดงผลลงในกราฟเพียงอย่างเดียวจึงทำให้การแสดงผลในหน้าจอของผู้ควบคุม นั้นมีความเที่ยงตรงและเหมือนกันทุกคน ดังนั้นความผิดพลาดในการสื่อสารของข้อมูลบนอินเทอร์เน็ตจึงมีผลเพียงการส่งถ่ายข้อมูลจากเซิร์ฟเวอร์มายังไคลเอนต์ต่าง ๆ แต่ความถูกต้องของข้อมูลจากการควบคุมที่นำไปแสดงผลยังไคลเอนต์นั้น ยังคงถูกต้องสมบูรณ์ทุกประการ และในงานวิจัยชิ้นนี้สามารถนำไปใช้งานด้านอุตสาหกรรมในได้หลายประเภท แต่ยังคงติดปัญหาในด้านการติดตั้งโปรแกรมเพื่อให้ระบบทำงานได้บนอินเทอร์เน็ตยังคงยุ่งยากอยู่ เนื่องจากต้องอาศัยโปรแกรมหลายโปรแกรมในการติดตั้งเช่น โปรแกรม Java โปรแกรม ASP โปรแกรม Delphi และโปรแกรมอื่น ๆ ที่เกี่ยวข้องกับฐานข้อมูลและบริการอินเทอร์เน็ต ซึ่งถ้าได้มีการวิจัยพัฒนาโปรแกรมอย่างต่อเนื่องเพื่อลด โปรแกรมเหล่านี้เหลือเพียงใช้โปรแกรม Delphi เพียงโปรแกรมเดียวได้นั้น ขั้นตอนการติดตั้งโปรแกรมจะง่ายและสะดวกสบายทำให้ในอนาคตอาจเป็นที่นิยมมากขึ้น และข้อสำคัญอีกประการหนึ่ง คือ ในการนำไปใช้งานจริงบนอินเทอร์เน็ต จำเป็นต้องคำนึงถึงความปลอดภัยของข้อมูลและระบบที่ให้บริการและควบคุมผ่านอินเทอร์เน็ตว่าสามารถป้องกันผู้ที่อาจเข้ามาก่อแควนหรือทำลายระบบของเราได้ ดังนั้น โปรแกรมจึงต้องมีการพัฒนาโดยคำนึงถึงความปลอดภัยของข้อมูลและระบบเครือข่ายที่ให้บริการมากขึ้น

เอกสารอ้างอิง

- [1]. C. Lon Enloe, W. A. Pakula, G. A. Finney, and R. K. Haaland, "Teleoperation in the Undergraduate Physics Laboratory – Teaching an Old Dog New Tics," IEEE Trans. Education. Vol. 42, no. 3 pp. 174-179, August. 1999.
- [2]. Y. Piguet and D. Gillet, "Java – based remote experimentation for control algorithms prototyping," Proc. the American Control Conference San Diego, California., pp.1465-1469, June 1999.
- [3]. G. H. Salazar-Silva, J. C. Martinez-Garcia and R. Garrido, "Enhancing Basic Robotics Education on the Web," Proc. the American Control Conference San Diego, California., pp.1470-1471, June 1999.
- [4]. D. E. Seborg, T. F. Edgar and D. A. Mellicamp, "Process Dynamics And Control," John Wiley & Sons, Inc, pp.195-196, 1989.
- [5]. M. Hughes, C. Hughes, M. Shoffner and M. Winslow, "Java Network Programming," Manning Publications Co., 1997.
- [6]. สรุชาติ พ่วงพุ่ม, "โครงสร้างข้อมูลกับปาสคาล," สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, หน้า 79-84, 2535.
- [7]. กิตติ เปรมพิณี, "คู่มือการใช้งาน Borland Delphi 4," เลพีเอ็น ซีสเต็มอินทิเกรเตอร์, ตุลาคม 2541.
- [8]. พ.อ. เจนวิทย์ เหลืองอร่าม, "การเขียนโปรแกรมสำหรับ Applications และ Applets ด้วย Java," บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพฯ. 2542.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมที่พัฒนาด้วย Java Application

```

import java.net.*;
import java.io.*;
import java.util.*;
import java.util.Date;
import java.sql.*;

public class MainServer {
    static String sPV = "", sSP = "", sMV = "", sTime = "", sTemp = "", sPrivilege = "";

    public MainServer (int port) throws IOException {
        ServerSocket server = new ServerSocket (port);
        while (true) {
            Socket client = server.accept ();
            System.out.println("Accepted from " + client.getInetAddress ());
            ChatHandler c = new ChatHandler (client);
            c.start ();
            GetParameter G = new GetParameter (client);
            G.start ();
        }
    }

    public static void main (String args[]) throws IOException {
        if (args.length != 1)
            throw new RuntimeException ("Syntax : MainServer <port>");
        System.out.println("Exit Teleoperate Server press key <Ctrl + C>");
        new DBThread().start();
        new MainServer (Integer.parseInt (args[0]));
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

class ChatHandler extends Thread {
    protected Socket s;
    protected DataInputStream i;
    protected DataOutputStream o;

    protected static Vector handlers = new Vector ();
    String sTemp,sData;

    public void UpdateDBIP(String sIP) {
        String url = "jdbc:odbc:TeleOperateDB";
        String DBIPQuery = "Delete from QueueLevel Where IP =";
        Connection con;
        Statement stmt;
        ResultSet rs;
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con = DriverManager.getConnection(url);
            stmt = con.createStatement();

            DBIPQuery = DBIPQuery + sIP + """;
            System.out.println("SQL : " + DBIPQuery);
            rs = stmt.executeQuery(DBIPQuery);

            stmt.close();
            con.close();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนสิทธิ์ในชื่อผู้อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.err.print("ClassNotFoundException: ");
        System.err.println(e.getMessage());
    }
    catch(SQLException ex) {}
}

```

```

public ChatHandler (Socket s) throws IOException {
    this.s = s;
    i = new DataInputStream (new BufferedInputStream (s.getInputStream ()));
    o = new DataOutputStream (new BufferedOutputStream (s.getOutputStream ()));
}
public void run () {
    try {
        handlers.addElement (this);
        while (true) {
            try {
                synchronized (handlers) {
                    try {
                        synchronized (o) {
                            o.writeUTF
(MainServer.sPrivilege+'A'+MainServer.sPV+'B'+MainServer.sSP
+'C'+MainServer.sMV+'D'+MainServer.sTime);
                            System.out.println(MainServer.sPrivilege);
                        }
                    }
                }
            }
            o.flush ();
        }
    }
    catch (IOException ex) {
        //***** Get IP Address for Delete in database ****
        sData = String.valueOf(this.s.getInetAddress());
        sTemp = "";
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา sTemp = ""; อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

String url = "jdbc:odbc:TeleOperateDB";
String CurrentDataQuery = "SELECT * FROM CurrentData";
String ParameterDataQuery = "SELECT * FROM ParameterData";
String QueueLevelQuery = "Select * from QueueLevel Q where Q.Order = (Select Min
(Order) as GetOrder from QueueLevel)";

Connection con;
Statement stmt;
String LevelTime,LevelPV,LevelMV,LevelSP;
ResultSet rs;

DBThread() {
    for (int i=0; i<=60; i++) {
        MainServer.sPV += "0,";
        MainServer.sSP += "0,";
        MainServer.sMV += "0,";
        MainServer.sTime += "0,";
    }
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
    catch (java.lang.ClassNotFoundException e) {
        System.err.print("ClassNotFoundException: ");
        System.err.println(e.getMessage());
    }
}

String transferData(String sData) {
    sTemp = "";
    for (int i= sData.indexOf(',') + 1; i < sData.length(); i++)
        sTemp += sData.charAt(i);
    return sTemp;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void run () {
    int iIndex = 0;
    while (true) {
        try {

            con = DriverManager.getConnection(url);
            stmt = con.createStatement();
            rs = stmt.executeQuery(CurrentDataQuery);
            while (rs.next()) {
                LevelTime = rs.getString("LevelTime");
                LevelPV = rs.getString("LevelPV");
                LevelMV = rs.getString("LevelMV");

                MainServer.sPV = transferData(MainServer.sPV);
                MainServer.sPV += LevelPV + ",";

                MainServer.sMV = transferData(MainServer.sMV);
                MainServer.sMV += LevelMV + ",";

                MainServer.sTime = transferData(MainServer.sTime);
                MainServer.sTime += LevelTime + ",";
            }

            rs = stmt.executeQuery(ParameterDataQuery);
            while (rs.next()) {

                LevelSP = rs.getString("LevelSP");

                MainServer.sSP = transferData(MainServer.sSP);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ MainServer.sSP += LevelSP + ","; อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

// ***** Get Input Stream from Client *****
i = new DataInputStream (new BufferedInputStream (s.getInputStream ()));

// ***** Declare database for update parameter input from client *****
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection(url);
    stmt = con.createStatement();
}
catch (java.lang.ClassNotFoundException e) {
    System.err.print("ClassNotFoundException: ");
    System.err.println(e.getMessage());
}
catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
}
public void run () {
    try {
        while (true) {
            String msg = i.readUTF();
            System.out.println(msg);
            try {
                stmt.executeUpdate(msg);
            } catch(SQLException ex) {
                System.err.println("SQLException: " + ex.getMessage());
            }
        }
    }
}
catch(IOException ex) {}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปรแกรมที่พัฒนาด้วย Java Applet

```

import java.applet.*;
import java.net.*;
import java.io.*;
import java.awt.*;
import java.util.*;
import java.text.*;

public class ClientControl extends Applet implements Runnable {
    protected DataInputStream i;
    protected DataOutputStream o;
    protected TextArea output;
    protected TextField input;
    protected Thread listener;
    protected int iPort,iTimeCounter;
    protected String sIP;
    protected String sPrivateCode;

    int iY,iX,iX1,PV,SP,MV,iHour,iMinute,iSecond;
    int iPv[] = new int[61];
    int iSP[] = new int[61];
    int iMV[] = new int[61];
    int iTime[] = new int[61];
    int xCoords[] = new int[3];
    int yCoords[] = new int[3];

    int MvManual = 0;
    int MvInput = 0;
    int SPManual = 0;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int SPInput = 0;

float KpManual = 100.0f;      // Kp Default

float TiManual = 20.0f; // Ti Default

float TdManual = 0.0f; // Td Default

boolean modeControl = true; // True = Automatic

boolean sendParameter;

boolean TimerEnable = false; // False = Is not Privilege user

String sMode = "1"; // 1 = Auto , uses for relay to server

Image offscreenImage;
Graphics offscreenGrp;

String sPV = "",sSP = "",sMV = "", sTime="",sTemp = "";
String sTempPV = "",sTempSP = "",sTempMV = "",sTempTime = "";
String sPrivilege="";

String s_Str,s_Temp;

Font f;
FontMetrics fm;

Date dateVar ;

SimpleDateFormat formatter;

Date currentDate;

String sTempTimer;

Panel panel1 = new Panel();

Label label1 = new Label();

Label label2 = new Label();

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง

```

Label label3 = new Label();
Label label4 = new Label();

Label Kplabel = new Label();
Label Tilabel = new Label();
Label Tdlabel = new Label();
Label Mvlabel = new Label();
Label SPlabel = new Label();
TextField KpEdit = new TextField();
TextField TiEdit = new TextField();
TextField TdEdit = new TextField();
TextField MvEdit = new TextField();
TextField SPEdit = new TextField();
TextField TRemainEdit = new TextField();
Button ModeBnt = new Button();
Button MvBnt = new Button();
Button KpUpBnt = new Button();
Button KpDownBnt = new Button();
Button TiUpBnt = new Button();
Button TiDownBnt = new Button();
Button TdUpBnt = new Button();
Button TdDownBnt = new Button();
Button MvUpBnt = new Button();
Button MvDownBnt = new Button();
Button SPUpBnt = new Button();
Button SPDownBnt = new Button();
Button SPSubmitBnt = new Button();
TimerThread threadTimer;
Scrollbar MvScrollbar;

int iXStart,iYStart;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public ClientControl() {}

public void init() {
    String sPort = getParameter("port");
    if (sPort == null)
        this.iPort = 4444; // Default Port
    else
        this.iPort = Integer.parseInt(sPort);
    String sIP = getParameter("ip");
    if (sIP == null)
        this.sIP = "192.191.1.1"; //Default IP Address
    else this.sIP = sIP;
    String sCode = getParameter("Privatecode");
    if (sCode == null)
        this.sPrivateCode = "00000";
    else this.sPrivateCode = sCode;
    String sTimeCounter = getParameter("TimeCounter");
    if (sTimeCounter == null)
        this.iTimeCounter = 600; // Default 10 minutes
    else this.iTimeCounter = Integer.parseInt(sTimeCounter);

    threadTimer = new TimerThread(this.iTimeCounter);

// ***** Assign Screen *****
    setLayout(null);
    setSize(size().width,size().height);
    panel1.setLayout(null);
    add(panel1);

    iXStart = size().width - 210;
    panel1.setBackground(new Color(70,64,198));

```

```

panel1.setBounds(ixStart,0,size().width,size().height);

label1.setText("Controller Parameters");
label1.setAlignment(Label.CENTER);
panel1.add(label1);
label1.setBackground(Color.yellow);
label1.setForeground(Color.black);
label1.setFont(new Font("MS Sans Serif",Font.BOLD,12));
label1.setBounds(5,8,200,20);

Kplabel.setText("Kp : ");
Kplabel.setAlignment(Label.LEFT);
panel1.add(Kplabel);
Kplabel.setForeground(Color.white);
Kplabel.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
Kplabel.setBounds(5,40,20,24);

panel1.add(KpEdit);
KpEdit.setBackground(Color.white);
KpEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
KpEdit.setBounds(28,40,30,20);

//***** Button Up - Down Kp *****
KpUpBnt.setLabel("KpUpBnt");
panel1.add(KpUpBnt);
KpUpBnt.setForeground(Color.lightGray);
KpUpBnt.setBackground(Color.lightGray);
KpUpBnt.setBounds(59,40,10,10);

KpDownBnt.setLabel("KpDownBnt");
panel1.add(KpDownBnt);
KpDownBnt.setForeground(Color.lightGray);
KpDownBnt.setBackground(Color.lightGray);

```

เอกสารนี้เป็นเอกสารที่ KpDownBnt.setBackground(Color.lightGray); อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

KpDownBnt.setBounds(59,51,10,10);

Tilabel.setText("Ti : ");
Tilabel.setAlignment(Label.LEFT);
panel1.add(Tilabel);
Tilabel.setForeground(Color.white);
Tilabel.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
Tilabel.setBounds(77,40,20,24);

panel1.add(TiEdit);
TiEdit.setBackground(Color.white);
TiEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
TiEdit.setBounds(95,40,30,20);

//***** Button Up - Down Ti *****
TiUpBnt.setLabel("TiUpBnt");
panel1.add(TiUpBnt);
TiUpBnt.setForeground(Color.lightGray);
TiUpBnt.setBackground(Color.lightGray);
TiUpBnt.setBounds(126,40,10,10);

TiDownBnt.setLabel("TiDownBnt");
panel1.add(TiDownBnt);
TiDownBnt.setForeground(Color.lightGray);
TiDownBnt.setBackground(Color.lightGray);
TiDownBnt.setBounds(126,51,10,10);

Tdlabel.setText("Td : ");
Tdlabel.setAlignment(Label.LEFT);
panel1.add(Tdlabel);
Tdlabel.setForeground(Color.white);
Tdlabel.setFont(new Font("MS Sans Serif",Font.PLAIN,11));

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Tdlabel.setBounds(141,40,20,24);

panel1.add(TdEdit);
TdEdit.setBackground(Color.white);
TdEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
TdEdit.setBounds(161,40,30,20);

//***** Button Up - Down Td *****
TdUpBnt.setLabel("TdUpBnt");
panel1.add(TdUpBnt);
TdUpBnt.setForeground(Color.lightGray);
TdUpBnt.setBackground(Color.lightGray);
TdUpBnt.setBounds(192,40,10,10);

TdDownBnt.setLabel("TdDownBnt");
panel1.add(TdDownBnt);
TdDownBnt.setForeground(Color.lightGray);
TdDownBnt.setBackground(Color.lightGray);
TdDownBnt.setBounds(192,51,10,10);

label2.setText("Controller Mode");
label2.setAlignment(Label.CENTER);
panel1.add(label2);
label2.setBackground(Color.yellow);
label2.setForeground(Color.black);
label2.setFont(new Font("MS Sans Serif",Font.BOLD,12));
label2.setBounds(5,100,200,20);

ModeBnt.setLabel("Automatic");
panel1.add(ModeBnt);
ModeBnt.setBackground(Color.lightGray);
ModeBnt.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
ModeBnt.setBounds(140,130,65,20);

```

เอกสารนี้เป็นเอกสารที่สแกนจากเว็บไซต์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Mvlabel.setText("Mv : ");
Mvlabel.setAlignment(Label.LEFT);
panel1.add(Mvlabel);
Mvlabel.setForeground(Color.white);
Mvlabel.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
Mvlabel.setBounds(15,130,20,24);

panel1.add(MvEdit);
MvEdit.setBackground(Color.white);
MvEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
MvEdit.setBounds(38,130,25,20);
MvEdit.setText("0");
MvEdit.setEditable(false);
label3.setText("SetPoint");
label3.setAlignment(Label.CENTER);
panel1.add(label3);
label3.setBackground(Color.yellow);
label3.setForeground(Color.black);
label3.setFont(new Font("MS Sans Serif",Font.BOLD,12));
label3.setBounds(5,190,200,20);
//***** Button Up - Down Kp *****
MvUpBnt.setLabel("MvUpBnt");
panel1.add(MvUpBnt);
MvUpBnt.setForeground(Color.lightGray);
MvUpBnt.setBackground(Color.lightGray);
MvUpBnt.setBounds(64,130,10,10);
MvUpBnt.setVisible(false);

MvDownBnt.setLabel("MvDownBnt");
panel1.add(MvDownBnt);
MvDownBnt.setForeground(Color.lightGray);
MvDownBnt.setBackground(Color.lightGray);

```

```

MvDownBnt.setBounds(64,141,10,10);
MvDownBnt.setVisible(false);

MvBnt.setLabel("Submit");
panel1.add(MvBnt);
MvBnt.setBackground(Color.lightGray);
MvBnt.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
MvBnt.setBounds(85,130,45,20);
MvBnt.setVisible(false);

MvScrollbar = new Scrollbar(Scrollbar.HORIZONTAL,1,0,0,100);
panel1.add(MvScrollbar);
MvScrollbar.setBounds(15,160,180,15);
MvScrollbar.setVisible(false);

SPlabel.setText("SP : ");
SPlabel.setAlignment(Label.LEFT);
panel1.add(SPlabel);
SPlabel.setForeground(Color.white);
SPlabel.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
SPlabel.setBounds(15,220,20,24);
panel1.add(SPEdit);
SPEdit.setBackground(Color.white);
SPEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
SPEdit.setBounds(38,220,25,20);

//***** Button Up - Down SP *****
SPUpBnt.setLabel("SPUpBnt");
panel1.add(SPUpBnt);
SPUpBnt.setForeground(Color.lightGray);
SPUpBnt.setBackground(Color.lightGray);
SPUpBnt.setBounds(64,220,10,10);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเรียงหน้าปกเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SPDownBnt.setLabel("SPDownBnt");
panel1.add(SPDownBnt);
SPDownBnt.setForeground(Color.lightGray);
SPDownBnt.setBackground(Color.lightGray);
SPDownBnt.setBounds(64,231,10,10);

SPSubmitBnt.setLabel("Submit");
panel1.add(SPSubmitBnt);
SPSubmitBnt.setBackground(Color.lightGray);
SPSubmitBnt.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
SPSubmitBnt.setBounds(140,220,65,20);

label4.setText("Time Remaining");
label4.setAlignment(Label.CENTER);
panel1.add(label4);
label4.setBackground(Color.yellow);
label4.setForeground(Color.black);
label4.setFont(new Font("MS Sans Serif",Font.BOLD,12));
label4.setBounds(5,280,200,20);

panel1.add(TRemainEdit);
TRemainEdit.setBackground(Color.white);
TRemainEdit.setFont(new Font("MS Sans Serif",Font.PLAIN,11));
TRemainEdit.setBounds(68,310,70,20);
TRemainEdit.setText("00:00:00");
TRemainEdit.setEnabled(false);

```

```
// ***** Initial Variable *****
```

```
KpEdit.setText(String.valueOf(KpManual));
```

เอกสารนี้เป็นเอกสารที่ `TiEdit.setText(String.valueOf(TiManual));` ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TdEdit.setText(String.valueOf(TdManual));
MvEdit.setText("0");
SPedit.setText(String.valueOf(SPManual));

//***** Initial Disable Component
EnableObject(false);

// ***** Assign Graphice *****
offscreenImage = createImage(size().width,size().height);
offscreenGrp = offscreenImage.getGraphics();

for (int i= 0; i <= 60 ;i+=1) {
    sPV += "0,"; // Simulate Stream values
    sSP += "0,"; // Simulate Stream values
    sMV += "0,"; // Simulate Stream values
    sTime += "0,"; // Simulate Stream values

    sTempPV += "0,";
    sTempSP += "0,";
    sTempMV += "0,";
    sTempTime += "0,";
}
}

public void destroy() {}

public void EnableObject(boolean Command) {
    KpEdit.setEnabled(Command);
    KpUpBnt.setEnabled(Command);
    KpDownBnt.setEnabled(Command);

    TiEdit.setEnabled(Command);
    TiUpBnt.setEnabled(Command);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TiDownBnt.setEnabled(Command);
```

```
TdEdit.setEnabled(Command);
```

```
TdUpBnt.setEnabled(Command);
```

```
TdDownBnt.setEnabled(Command);
```

```
MvEdit.setEnabled(Command);
```

```
ModeBnt.setEnabled(Command);
```

```
MvUpBnt.setEnabled(Command);
```

```
MvDownBnt.setEnabled(Command);
```

```
MvBnt.setEnabled(Command);
```

```
MvScrollbar.setEnabled(Command);
```

```
SPUpBnt.setEnabled(Command);
```

```
SPDownBnt.setEnabled(Command);
```

```
SPSubmitBnt.setEnabled(Command);
```

```
label4.setVisible(Command);
```

```
TRemainEdit.setVisible(Command);
```

```
}
```

```
String transferData(String sData) {
```

```
    sTemp = "";
```

```
    for (int i = sData.indexOf(',') + 1; i < sData.length(); i+=1) {
```

```
        sTemp += sData.charAt(i);
```

```
    }
```

```
    return sTemp;
```

```
}
```

```
int[] convertStrToArr(String sData) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,iLength,iOffset = 0;
int iIndex = 0;
int iArr[] = new int[61];

do{
    iLength = sData.indexOf(',',iOffset); // Length of char '
    sTemp = "";
    for (i = iOffset; i < iLength; i++) {
        sTemp += sData.charAt(i);
    }
    iOffset = iLength+1;
    iArr[iIndex++] = Integer.parseInt(sTemp);
} while (iIndex <=60);
return iArr;
}

String[] convertStrToTimeArr(String sData) {
    int i,iLength,iOffset = 0;
    int iIndex = 0;
    String iArr[] = new String[61];
    do{
        iLength = sData.indexOf(',',iOffset); // Length of char '
        sTemp = "";
        for (i = iOffset; i < iLength; i++) {
            sTemp += sData.charAt(i);
        }

        iOffset = iLength+1;
        iArr[iIndex++] = sTemp;

    } while (iIndex <=60);
return iArr;
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String stringMid(String sData,int iOffset,int iLength)    {
    int i;
    sTemp ="";
    for (i = iOffset;i<iLength;i++) {
        sTemp += sData.charAt(i);
    }
    return sTemp;
}

public void getDataArray()                                // Shift Data Array
{
    int i;
    //----- GET DATA OF PV PART -----
    sTempPV= transferData(sTempPV);
    sTemp = stringMid(sPV,0,sTempPV.length());
    if (sTemp.compareTo(sTempPV) != 0) {
        iPV = convertStrToArr(sPV);// Convert data from String to array of Integer.
    } else {
        for (i= 0; i <= 59 ;i++) {
            iPV[i] = iPV[i+1];
        }
        sTemp = stringMid(sPV,sTempPV.length(),sPV.length()-1);
        iPV[60] = Integer.parseInt(sTemp);
    }
    sTempPV = sPV;

    //----- GET DATA OF SP PART -----
    sTempSP= transferData(sTempSP);
    sTemp = stringMid(sSP,0,sTempSP.length());
    if (sTemp.compareTo(sTempSP) != 0) {
        iSP = convertStrToArr(sSP); // Convert data from String to array of Integer.
    } else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        iSP[i] = iSP[i+1];
    }
    sTemp = stringMid(sSP,sTempSP.length(),sSP.length()-1);
    iSP[60] = Integer.parseInt(sTemp);
}
sTempSP = sSP;
//----- GET DATA OF MV PART -----
sTempMV= transferData(sTempMV);
sTemp = stringMid(sMV,0,sTempMV.length());
if (sTemp.compareTo(sTempMV) != 0) {
    iMV = convertStrToArr(sMV); // Convert data from String to array of Integer.
} else {
    for (i= 0; i <= 59 ;i++) {
        iMV[i] = iMV[i+1];
    }
    sTemp = stringMid(sMV,sTempMV.length(),sMV.length()-1);
    iMV[60] = Integer.parseInt(sTemp);
}
sTempMV = sMV;
//----- GET DATA OF TIME PART -----
sTempTime= transferData(sTempTime);
sTemp = stringMid(sTime,0,sTempTime.length());
if (sTemp.compareTo(sTempTime) != 0) {
    iTime = convertStrToArr(sTime); // Convert data from String to array of Integer.
} else {
    for (i= 0; i <= 59 ;i++) {
        iTime[i] = iTime[i+1];
    }
    sTemp = stringMid(sTime,sTempTime.length(),sTime.length()-1);
    iTime[60] = Integer.parseInt(sTemp);
}
sTempTime = sTime; }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void targetPoint(Graphics g,int iIndex , int iType) {
    int x,y;
    x = 43+(60)*8;
    y = 30+(100-iIndex)*3;
    xCoords[0] = x;
    yCoords[0] = y;
    xCoords[1] = x+8;
    yCoords[1] = y-6;
    xCoords[2] = x+8;
    yCoords[2] = y+5;
    if (iType == 0) {
        offscreenGrp.setColor(Color.cyan);
        offscreenGrp.drawString("PV", xCoords[0]+10,yCoords[0] + 4);
    }else if (iType == 1) {
        offscreenGrp.setColor(Color.yellow);
        offscreenGrp.drawString("SP", xCoords[0]+10 ,yCoords[0] + 4);
    }else {
        offscreenGrp.setColor(Color.red);
        offscreenGrp.drawString("MV", xCoords[0]+10 ,yCoords[0] + 4);
    }
    offscreenGrp.fillPolygon(xCoords,yCoords,3);
}

```

```

public void drawBackGround(Graphics g) {
    int i;
    // Drawing Axist X and Y
    offscreenGrp.setColor(Color.black);
    offscreenGrp.fillRect(0,0,550,352);
    offscreenGrp.setColor(Color.white);
    offscreenGrp.drawString("Percent (%)", 15, 15);
    offscreenGrp.drawString("Time", 525, 334);
    offscreenGrp.drawLine(39,25,39,336);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
offscreenGrp.drawLine(39,330,524,330);
```

```
f = new Font("MS Sans Serif",Font.PLAIN,10);
```

```
fm = getFontMetrics(f);
```

```
offscreenGrp.setFont(f);
```

```
for (i = 0; i < 51; i++) { // Draw vertical axist in percent unit
```

```
    iY = 30+i*6;
```

```
    if ((i % 5) == 0) {
```

```
        s_Str = Integer.toString(100-i*2);
```

```
        offscreenGrp.drawString(s_Str,30-fm.stringWidth(s_Str),iY+4);
```

```
        offscreenGrp.drawLine(34,iY, 38, iY);
```

```
    } else {
```

```
        offscreenGrp.drawLine(36,iY, 38, iY);
```

```
    }
```

```
for (i = 0; i <= 60; i +=1) { // Draw horizontal axist in percent unit
```

```
    iX = 39+i*8;
```

```
    if ((i % 10) == 0) {
```

```
        iHour = iTime[i] / 3600;
```

```
        iMinute = iTime[i] % 3600;
```

```
        iSecond = iMinute % 60;
```

```
        iMinute = iMinute / 60;
```

```
        s_Str = Integer.toString(iHour)+ ":" + Integer.toString
```

```
(iMinute)+ ":" + Integer.toString(iSecond);
```

```
        offscreenGrp.drawString(s_Str,iX - fm.stringWidth(s_Str),345);
```

```
        offscreenGrp.drawLine(iX,330,iX,336);
```

```
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        offscreenGrp.drawLine(iX,330,iX,334);
    }
}
//*****          DRAW GRID          *****
offscreenGrp.setColor(Color.gray);
for (i = 0; i < 46 ; i +=5) { // Draw vertical axist in percent unit
    iY = 30+i*6;
    offscreenGrp.drawLine(40,iY, 519, iY); // Draw Vertical Grid...
}
for (i= 10; i <= 60; i +=10) { // Draw horizontal axist in percent unit
    iX = 39+i*8;
    offscreenGrp.drawLine(iX,30,iX,329); // Draw Horizontal Grid...
}
//*****          DRAW GRAPH DATA          *****
for (i = 0; i <60; i++) {
    iX = 39+i*8; // Calculation X Axist
    iX1 = 39+(i+1)*8;

    offscreenGrp.setColor(Color.cyan);
    offscreenGrp.drawLine(iX,30+(100-iPV[i])*3, iX1, 30+(100-iPV[i+1])*3);
    offscreenGrp.setColor(Color.yellow);
    offscreenGrp.drawLine(iX,30+(100-iSP[i])*3, iX1, 30+(100-iSP[i+1])*3);
    offscreenGrp.setColor(Color.red);
    offscreenGrp.drawLine(iX,30+(100-iMV[i])*3, iX1, 30+(100-iMV[i+1])*3);
}
    targetPoint(g,iSP[i],1);
    targetPoint(g,iMV[i],2);
    targetPoint(g,iPV[i],0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void DecodeDataStream (String sData)
{
    //----- Decode Privilege Code -----
    sTemp = "";
    for (int i = 0; i < sData.indexOf('A'); i++)
    {
        sTemp += sData.charAt(i);
    }
    sPrivilege = sTemp;

    //----- Decode Data -----
    sTemp = "";
    for (int i = sData.indexOf('A')+1; i < sData.indexOf('B'); i++)
    {
        sTemp += sData.charAt(i);
    }
    sPV = sTemp;
    sTemp = "";
    for (int i = sData.indexOf('B')+1; i < sData.indexOf('C'); i++) {
        sTemp += sData.charAt(i);
    }
    sSP = sTemp;
    sTemp = "";
    for (int i = sData.indexOf('C')+1; i < sData.indexOf('D'); i++) {
        sTemp += sData.charAt(i);
    }
    sMV = sTemp;
    sTemp = "";
    for (int i = sData.indexOf('D')+1; i < sData.length(); i++) {
        sTemp += sData.charAt(i);
    }
    sTime = sTemp;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public void update(Graphics g) {
    paint(g);
    if (modeControl) // Mv was updated from condition of Auto.
        MvEdit.setText(String.valueOf(iMV[60]));
}

public void paint(Graphics g) {
    drawBackGround(g);
    g.drawImage(offscreenImage,0,0,this);
}

public void start() {
    try {
        Socket s = new Socket (this.sIP, this.iPort); // Create Socket

        this.i = new DataInputStream(s.getInputStream());
        this.o = new DataOutputStream(s.getOutputStream());

        listener = new Thread (this);
        listener.start();

    } catch (IOException ex) {
        ex.printStackTrace ();
    }
}

public void stop() {
    if (listener != null) {
        listener.stop();
        listener = null;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public void run()    {
    try{
        while (true){
            try {
                String line = i.readUTF();

                DecodeDataStream(line);
                //----- Check Privilege Code -----
                if (sPrivilege.equals(this.sPrivateCode)) {
                    EnableObject(true);
                    if (TimerEnable == false) {
                        threadTimer.start();
                        TimerEnable = true;
                    }
                }

                /******* Check Time Remaining out
                *****/

                if(((threadTimer.iTimeCount < 1) && (TimerEnable))
                {
                    try {
                        o.writeUTF("Delete from QueueLevel Where
IP = ""+ getLocalIP(InetAddress.getLocalHost()) + """);
                        o.flush ();
                        TimerEnable = false;
                    } catch (IOException ex){}
                }
            } else {
                EnableObject(false);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    /******* Print Time remaining in the screen
    *****/

```

```

TRemainEdit.setText(TimeString(threadTimer.iTimeCount));

                                getDataArray();           // Call ShiftArray method

                                repaint();
                                Thread.sleep(500);
                                } catch (InterruptedException e)
                                {}
                                }
                                catch (IOException ex) {
                                    ex.printStackTrace ();
                                }
                                finally {
                                    listener = null;
                                    validate();
                                try{
                                    o.close ();
                                }
                                catch (IOException ex) {
                                    ex.printStackTrace ();
                                }
                                }
                                }

public boolean handleEvent (Event e) {
    float iTemp;
    if (e.target instanceof Scrollbar) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ MvManual = ((Scrollbar)e.target).getValue(); นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MvEdit.setText(String.valueOf(MvManual));
sTemp = MvEdit.getText() + "," + KpEdit.getText() + "," +
TiEdit.getText() + "," + TdEdit.getText();
input.setText(sTemp);
return true;
}

```

```

if ((e.target == KpEdit) && (e.id == Event.ACTION_EVENT)) {
    KpManual = Float.valueOf(KpEdit.getText()).floatValue(); // Get
    Floting KP
    RelayParameter();
    return true;
}

```

```

if ((e.target == TiEdit) && (e.id == Event.ACTION_EVENT)) {
    TiManual = Float.valueOf(TiEdit.getText()).floatValue(); // Get
    Floting KP
    RelayParameter();
    return true;
}

```

```

if ((e.target == TdEdit) && (e.id == Event.ACTION_EVENT)) {
    TdManual = Float.valueOf(TdEdit.getText()).floatValue(); // Get
    Floting KP
    RelayParameter();
    return true;
}

```

```

if ((e.target == MvEdit) && (e.id == Event.ACTION_EVENT)) {
    MvManual = Integer.valueOf(MvEdit.getText()).intValue(); //
    Get Integer MV
    MvEdit.setText(String.valueOf(MvManual));
    MvScrollbar.setValue(MvManual);
    return true;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if ((e.target == SPedit) && (e.id == Event.ACTION_EVENT)) {
            SPManual = Integer.valueOf(SPedit.getText()).intValue(); // Get
Integer MV
            SPedit.setText(String.valueOf(SPManual));
            return true;
        }
        return super.handleEvent (e);
    }
}

```

```

public boolean action(Event evt, Object arg) {
String buttonLabel;
    if (evt.target instanceof Button) {
        buttonLabel = (String)arg;
        if (buttonLabel == "Automatic") // Return to Manual Mode
        {
            ModeBnt.setLabel("Manual");
            MvScrollbar.setVisible(true);
            MvBnt.setVisible(true);
            MvUpBnt.setVisible(true);
            MvDownBnt.setVisible(true);
            modeControl = false;
            MvManual = iMV[60];
            MvScrollbar.setValue(MvManual);
            MvEdit.setEditable(true);
            sMode = "0";
            MvInput = MvManual; // Update data Input
            RelayParameter();
        }
        else if (buttonLabel == "Manual") // Return to Auto Mode
        {
            ModeBnt.setLabel("Automatic");
            MvScrollbar.setVisible(false);

```

```

MvBnt.setVisible(false);
MvUpBnt.setVisible(false);
MvDownBnt.setVisible(false);
modeControl = true;
MvEdit.setEditable(false);
sMode = "1";
SPInput = SPManual;           // Update data Input
RelayParameter();
}
else if (buttonLabel == "KpUpBnt") {
    KpUpBntClick(true);
    RelayParameter();
}
else if (buttonLabel == "KpDownBnt") {
    KpUpBntClick(false);
    RelayParameter();
}
else if (buttonLabel == "TiUpBnt") {
    TiUpBntClick(true);
    RelayParameter();
}
else if (buttonLabel == "TiDownBnt") {
    TiUpBntClick(false);
    RelayParameter();
}
}
else if (buttonLabel == "TdUpBnt") {
    TdUpBntClick(true);
    RelayParameter();
}
else if (buttonLabel == "TdDownBnt") {
    TdUpBntClick(false);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else if (buttonLabel == "MvUpBnt") {
        MvUpBntClick(true);
    }
    else if (buttonLabel == "MvDownBnt") {
        MvUpBntClick(false);
    }
    else if (evt.target == MvBnt) {
        MvInput = MvManual;
        RelayParameter();
    }
    else if (buttonLabel == "SPUpBnt") {
        SPUpBntClick(true);
    }
    else if (buttonLabel == "SPDownBnt") {
        SPUpBntClick(false);
    }
    else if (evt.target == SPSubmitBnt) {
        SPInput = SPManual;
        RelayParameter();
    }
    return true;
}
return false;
}
}

```

```

public void KpUpBntClick(boolean UpDown) {
    if (UpDown) // Clicl Up = Increment Kp
    {
        if (KpManual < 9999.9) KpManual += 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเฉพาะเท่านั้น การนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (KpManual > 0.0) KpManual -= 1;
        KpEdit.setText(String.valueOf(KpManual));
    }

```

```

public void TiUpBntClick(boolean UpDown)    {
    if (UpDown)    // Clicl Up = Increment Ti    {
        if (TiManual < 9999.9) TiManual += 1;
    } else        // Clicl Down = Decrement Ti
        if (TiManual > 0.0) TiManual -= 1;
    TiEdit.setText(String.valueOf(TiManual));
}

```

```

public void TdUpBntClick(boolean UpDown)    {
    if (UpDown)    // Clicl Up = Increment Td
    {
        if (TdManual < 9999.9) TdManual += 1;
    } else        // Clicl Down = Decrement Td
        if (TdManual > 0.0) TdManual -= 1;
    TdEdit.setText(String.valueOf(TdManual));
}

```

```

public void MvUpBntClick(boolean UpDown)
{
    if (UpDown)    // Clicl Up = Increment Mv
    {
        if (MvManual < 100) MvManual += 1;
    } else        // Clicl Down = Decrement Mv
        if (MvManual > 0) MvManual -= 1;
    MvEdit.setText(String.valueOf(MvManual));
    MvScrollbar.setValue(MvManual);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void SPUpBntClick(boolean UpDown)
{
    if (UpDown) // Clicl Up = Increment SP
    {
        if (SPManual < 100) SPManual += 1;
    } else // Clicl Down = Decrement SP
        if (SPManual > 0) SPManual -= 1;
    SPEDIT.setText(String.valueOf(SPManual));
}

public void RelayParameter() {
    String sData;
    try {
        sData = "LevelKp="+ KpEdit.getText()+" ,LevelTi="+ TiEdit.getText
()+" ,LevelTd="+ TdEdit.getText();
        sData = sData + " ,LevelSP=" + String.valueOf(SPInput)
+" ,LevelMV_Input="+String.valueOf(MvInput)+" ,LevelMode=" + sMode;
        o.writeUTF("update ParameterData set " + sData);
        o.flush ();
    } catch (IOException ex) {
        //listener.stop ();
        ex.printStackTrace ();
    }
}
}
}

```

```

String TimeString(int iTimeCount) {
    String sTimer;
    int iHour,iMinute,iSecond;

    iHour = iTimeCount / 3600;

```

```

iSecond = iMinute % 60;
iMinute = iMinute / 60;

sTimer = " ";
if (iHour < 10)
    sTimer = sTimer + "0" + Integer.toString(iHour);
else
    sTimer = sTimer + Integer.toString(iHour);

sTimer = sTimer + ":";

if (iMinute < 10)
    sTimer = sTimer + "0" + Integer.toString(iMinute);
else
    sTimer = sTimer + Integer.toString(iMinute);
sTimer = sTimer + ":";

if (iSecond < 10)
    sTimer = sTimer + "0" + Integer.toString(iSecond);
else
    sTimer = sTimer + Integer.toString(iSecond);

return sTimer;
}

```

```
String getLocalIP(InetAddress sInetAddr) {
```

```
String sTemp="",sData;
```

```
    sData = String.valueOf(sInetAddr);
```

```
    for (int i = sData.indexOf('/') + 1; i < sData.length(); i+=1) {
```

```
        sTemp += sData.charAt(i);
```

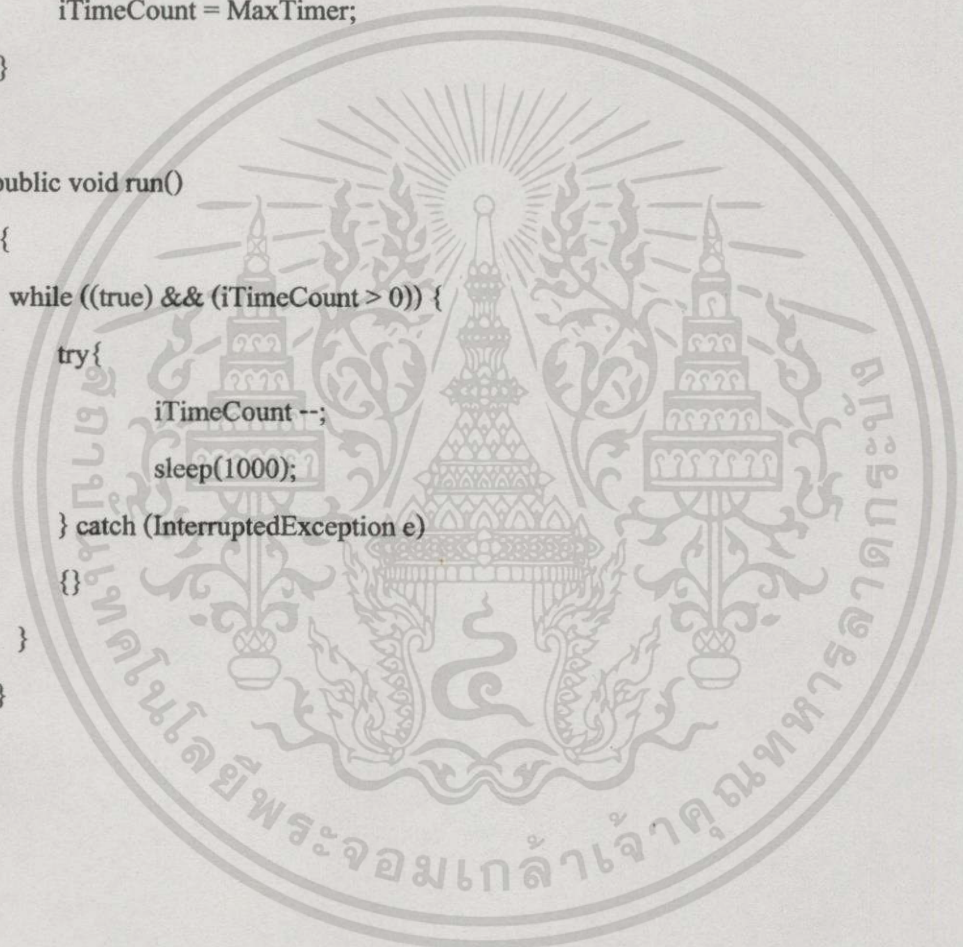
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    return sTemp;
}
}
class TimerThread extends Thread {
    public int iTimeCount = 0;
    int iHour,iMinute,iSecond;
    public TimerThread(int MaxTimer) {
        iTimeCount = MaxTimer;
    }

    public void run()
    {
        while ((true) && (iTimeCount > 0)) {
            try{
                iTimeCount --;
                sleep(1000);
            } catch (InterruptedException e)
            {}
        }
    }
}
}

```



ภาคผนวก ก.

บทความวิจัยที่ได้รับการตีพิมพ์ลงในวารสาร ในวิทยานิพนธ์ คือ

- [1] สรรเพชญ์ เชื้อแก้ว ประสิทธิ์ จุลเสรีวงศ์ วันชัยวีรจจา “การทดลองการกระบวน การที่ถูกควบคุมระยะไกลผ่านทางอินเทอร์เน็ต” วิศวกรรมลาดกระบัง 2543, หน้า 47-52.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ISSN 0125-1724

วิศวกรรมศาสตร์

ลาดกระบัง

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

LADKRABANG ENGINEERING JOURNAL

ปีที่ 17 ฉบับที่ 2

มิถุนายน 254:

- | | |
|--|-----|
| 1. การวิเคราะห์ข้อมูลใน ISUP ของโครงข่ายบริการร่วมระบบดิจิทัล
ฉัตรชัย วัฒนพันธุ์ พิเชฐ ม่วงนวล มนูญ สุขเกษม | 1 |
| 2. ระบบควบคุมการใช้งานโทรศัพท์สำหรับชุมสายปลายทางอัตโนมัติ
สุรเชษฐ์ ดิชาภิรมย์ พิเชฐ ม่วงนวล อวิฬ พึ่งมา | 6 |
| 3. การศึกษาปริมาณการพาหุทิศทางสายระหว่างชุมสายโทรศัพท์ที่ติดตั้งกับชุมสายต่อผ่านภายในท้องถิ่นเพื่อปรับปรุงวงจรเชื่อมต่อทรังค์ให้
เหมาะสมกับปริมาณการสื่อสารในอนาคต
ชชาติ พุตระกูล วินัย สังขวิจิตร พิเชฐ ม่วงนวล อวิฬ พึ่งมา | 12 |
| 4. การปรับปรุงสถานีฐานของโทรศัพท์เคลื่อนที่ระบบ AMPS โดยวิธีกำลังสองน้อยที่สุด
อรรถสิทธิ์ อินทร์น้อย พิเชฐ ม่วงนวล อวิฬ พึ่งมา | 18 |
| 5. เครื่องรับแบบวงปิดของคอสแอสที่ปรับปรุงขึ้นใหม่
ธรรมบุญ สุขหวาน วินัย ศิริผลหลาย จริญญา เลิศจรัสวัฒน์ วิภา แสงพิลธิ์ | 24 |
| 6. การประยุกต์โครงข่ายเชื่อมต่อภายในกับการส่ง PCM-TDM frame
ดำรงศ จันทร์เรือง กอบชัย เดชหาญ | 30 |
| 7. Discrete Array Antenna Pattern Synthesis with Side Lobe Level Controllable
Chuwong Phongcharoenpanich Voravit Ongwongwattana Sompol Kolsulvit Monai Krairiksh | 36 |
| 8. การแสดงคุณสมบัติการเชื่อมต่อโพรโทคอลหลายรูปแบบ และการบีบอัดข้อมูลลงบนเครือข่ายเฟรมรีเลย์
อวิชชัย มะพะสาโร กอบชัย เดชหาญ ชาลิน สารรวงศ์ | 41 |
| 9. การทดลองกระบวนการที่ถูกควบคุมระยะไกลผ่านทางอินเตอร์เน็ต
สรวพชญ์ เขื่อนแก้ว ประสิทธิ์ จุลเสรีวงศ์ วินัย วัชรวิภา | 47 |
| 10. วงจรกำเนิดความถี่ 57 เมกะเฮิร์ต แบบ RC ที่สามารถปรับค่าออฟเซตของเอาต์พุตได้ด้วยแรงดันควบคุม
วินัย สังขวิจิตร พิเชฐ ม่วงนวล อวิฬ พึ่งมา อติศักดิ์ มนต์ประภัสสร | 53 |
| 11. การสร้างตัวกรองเชิงเลขแบบหลายแถบความถี่ผ่าน
สายนำผ่าน หอมจันทร์ สุพันธ์ อิมมัน พยง เดชอยู่ กอบชัย เดชหาญ | 59 |
| 12. การควบคุมการแสดงผล Dot Matrix ขนาดใหญ่โดยใช้ไมโครโปรเซสเซอร์ต่อแบบ Master-Slave
ประวิดิชัย สมภกรักษ์ พิเชฐ ม่วงนวล อวิฬ พึ่งมา | 65 |
| 13. ตัวควบคุมพีซีซีไอแบบปรับค่าสมรรถนะด้วยตัวเอง
อุมาพร ทองรักษ์ กิตติ ไพฑูรย์วัฒนกิจ นิमित บุญภิรมย์ | 71 |
| 14. การตรวจหาเชื่อมแลเรียในภาพเซลล์เม็ดเลือดแดงโดยการเปรียบเทียบมัลติพิกซอลของโปรแกรม
กรีซ สมกัณษา สมชัย เอี้ยวสาธุรักษ์ บุญธีร์ เครือคราชู | 77 |
| 15. การออกแบบหาขนาดของหัวอ่านแม่เหล็กไฟฟ้าที่เหมาะสมที่สุดโดยเทคนิคโปรแกรมทางคณิตศาสตร์
มงคล มงคลวงศ์โรจน์ อมรชัย ฉัตรสง่า | 83 |
| 16. อุปกรณ์จัดเรียงในวงจรไฟฟ้ากระแสสลับแรงดันต่ำโดยใช้ผลลัพท์เชิงเลข
สนธยา เมืองน้อย กอบชัย เดชหาญ สมยศ จุณณะปิยะ | 89 |
| 17. การวิเคราะห์ขนาดกระแสไฟฟ้าของสายเคเบิลที่ใช้ในช่องทางฟราวด์สาย และต่อรอยสายใต้ดิน
ไชยา สายันต์ ศุภี บรรจงจิตร | 95 |
| 18. การลดสัญญาณฮาร์โมนิกในสายนิวทรัลในระบบไฟฟ้ากำลังแบบ 3 เฟส 4 สาย ที่ต่อกับภาระไฟฟ้าที่ไม่เป็นเชิงเส้น
วิรเทพ พงศ์ศรีเพียร จงรักษ์ บุญเส็ง | 101 |
| 19. การหาค่าพารามิเตอร์ของมอเตอร์เหนี่ยวนำขณะการทำงานในสภาวะคงตัว
เอก พันธุ์พุ่ม สดาพร อุดมสิน ชัยภักษ์ วรรณะสาร | 107 |
| 20. การวิเคราะห์เปรียบเทียบสมรรถนะของเทคนิคควบคุมกระแสสำหรับการควบคุมมอเตอร์เหนี่ยวนำ 3 เฟส แบบเวกเตอร์ฟลักซ์
พิเชษฐ ฉัตรพาน วิจิตร กิณเรศ | 113 |
| 21. การพัฒนาการออกแบบที่เหมาะสมสำหรับหม้อแปลงไฟฟ้าขนาดเล็กแบบเซลล์
ก่อเกียรติ กาญจนรัตน์ ชัยวุฒิ ฉัตรอุทัย | 119 |
| 22. Effects of Feed Temperature on The Coupling of Endothermic-Exothermic Reactions in a Multitubular Heat Exchanger Reactor
Teerapol Buranaosod Piyanat Intanakul Noppakom Thamphiphit Prakob Kitchaiya | 125 |
| 23. การออกแบบเพื่อหาขนาดที่เหมาะสมที่สุดของเครื่องแลกเปลี่ยนความร้อน โดยใช้วิธีบรอยเดน-เฟลด์เซอร์-โกลด์เฟรบบ-ชันโน
มงคล มงคลวงศ์โรจน์ วรภัทร พึ่งเจริญ | 131 |
| 24. พฤติกรรมทางสถิติและพลศาสตร์ของแรงไหลกลมห่อล่อด้วยสารหล่อลื่นนอน-นิวตรอนเนียน โดลาแดนท์
มงคล มงคลวงศ์โรจน์ ขนิษฐา วงษ์สีดาแก้ว | 137 |
| 25. การวิเคราะห์การสั่นสะเทือนของเพื่องเฉียง
มงคล มงคลวงศ์โรจน์ ขนิษฐา วงษ์สีดาแก้ว | 143 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองกระบวนการที่ถูกรควบคุมระยะไกลผ่านทางอินเทอร์เน็ต

Teleoperate Process Laboratory with Internet

สรรเพชญ์ เชื้อแก้ว ประสิทธิ์ จุลเสวีวงศ์ วันชัย ธีรวิบูลย์

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

บทความนี้เป็นการเสนอการควบคุมกระบวนการจากเว็บเบราว์เซอร์ผ่านทางอินเทอร์เน็ตเพื่อใช้ในการทดลองกระบวนการผ่านไปยังห้องควบคุมกระบวนการของทางมหาวิทยาลัย กระบวนการที่ใช้ในบทความนี้เป็นชุดทดลองกระบวนการควบคุมระดับน้ำและชุดทดลองควบคุมอุณหภูมิภายในห้อง โดยเครื่องควบคุมที่ใช้เป็นระบบ PID ซึ่งผู้ทดลองสามารถที่จะควบคุมจากที่ใดก็ได้โดยไม่มีข้อจำกัดเรื่องระยะทาง ในบทความได้นำเสนอถึงแนวคิดในการจัดลำดับในการควบคุมผ่านทางอินเทอร์เน็ตและการบันทึกข้อมูลที่ได้จากการทดลอง (recorder) ผ่านทางอินเทอร์เน็ตตลอดจนการนำเอาข้อมูลที่ถูกต้องที่ใกล้เคียงจนวนเวลาเดียวกันกับเวลาของเครื่องควบคุมของแม่ข่าย (server) มาแสดงผลบนเบราว์เซอร์

Abstract

This paper presents the remote control through browser over the internet. The process control plants of university's laboratory room are the water level control plant and the temperature control plant by using PID controller in the experimental. Furthermore, they can be controlled anywhere without the restriction in term of distance. This paper describes the concepts of order for controlling over the internet and the data record in the experimental is done through the internet. Finally, the precise data maintained nearly real time as the control server is presented on the browser.

1. บทนำ

งานวิจัยนี้เป็นงานที่พัฒนาขึ้นเพื่อการศึกษาที่นักศึกษาสามารถทดลองกระบวนการ (process) ผ่านทางอินเทอร์เน็ต โดยไม่จำเป็นต้องเดินทางเพื่อมาทดลองที่มหาวิทยาลัยเพียงอย่างเดียว อันเป็นการเปิดโอกาสทางการศึกษาให้กับนักศึกษาผู้ที่อยู่ห่างไกลจากแคลนอาจารย์ผู้สอนหรือขาดเครื่องมือในการที่ทันสมัยเพื่อใช้ในการทดลอง โดยจะทำให้ให้นักศึกษาเหล่านั้นสามารถทดลองกระบวนการต่าง ๆ ได้โดยใช้กระบวนการจริง

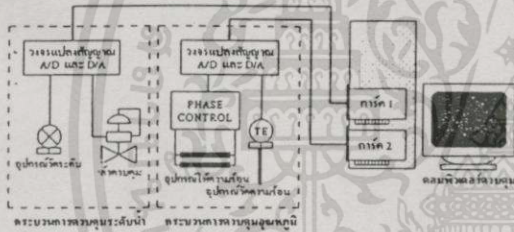
งานวิจัยชิ้นนี้มีจุดประสงค์หลัก คือ เพื่อการทดลองกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิผ่านทางอินเทอร์เน็ตโดยสามารถควบคุมผ่านทางเบราว์เซอร์ต่างๆไปได้แก่ Internet Explorer หรือ Netscape Communicator บนอินเทอร์เน็ตซึ่งถือได้ว่าเป็นเบราว์เซอร์ที่มีความนิยมมากที่สุดในปัจจุบันจึงง่ายต่อการใช้งานของผู้ควบคุมเนื่องจากไม่ต้องหาโปรแกรมเฉพาะตัวอื่นมาใช้งาน ซึ่งจะเป็นการพัฒนาวิธีการต่าง ๆ เพื่อให้เบราว์เซอร์ของผู้ทดลองสามารถรับส่งข้อมูลในการควบคุมระยะไกลไปยังคอมพิวเตอร์แม่ข่าย(server) ที่

สองโปรแกรมเช่นกัน ดังรูปที่ 2 การอินเตอร์เฟสแต่ละชุดจะประกอบไปด้วยอุปกรณ์ต่าง ๆ ดังนี้

1. การ์ดอินเตอร์เฟส ซึ่งจะทำหน้าที่รับข้อมูลดิจิตอลเข้ามาสู่คอมพิวเตอร์เพื่อประมวลผลในโปรแกรมเครื่องควบคุมและส่งข้อมูลดิจิตอลที่ได้จากการคำนวณออกไป

2. วงจรแปลงสัญญาณ A/D และ D/A วงจรนี้จะทำหน้าที่เป็นอุปกรณ์ที่แปลงสัญญาณจากอนาลอกเป็นดิจิตอล (A/D) และจากดิจิตอลเป็นอนาลอก (D/A)

3. อุปกรณ์ที่เป็นตัวอินพุตและเอาพุตของกระบวนการ ซึ่งอุปกรณ์ที่เป็นอินพุตได้แก่ เครื่องตรวจวัดระดับของน้ำและเครื่องวัดอุณหภูมิ ส่วนอุปกรณ์ที่เป็นเอาพุตได้แก่ตัวควบคุมระดับน้ำและเครื่องควบคุมมุมแรงดันไฟฟ้าในการกำเนิดความร้อนในตู้อบ



รูปที่ 2 แสดงอินเตอร์เฟสกับกระบวนการ

2.4 สมการ PID ที่ใช้ควบคุมในคอมพิวเตอร์

สมการ PID ทำหน้าที่เป็นเครื่องควบคุมกระบวนการซึ่งจะมีค่าพารามิเตอร์ K_p , T_i , T_d และ SP เป็นที่จะถูกปรับแต่งจากผู้ทดลองผ่านทางเบราว์เซอร์

การควบคุมแบบ PID มีสมการพื้นฐานดังนี้ [4]

$$Vo(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (1)$$

และ $e(t) = SP(t) - PV(t)$

- เมื่อ SP : Setpoint (0 - 100%)
- K_p : Proportional Gain (1 - 999.9)
- T_i : Integral Time (1 - 9999 Sec)
- T_d : Derivative Time (0 - 9999 Sec)

จากสมการข้างต้นสามารถแปลงให้เป็นสมการดิฟเฟอเรนเชียลแบบต่อเนื่องให้คอมพิวเตอร์สามารถ

คำนวณได้ โดยผ่านการแปลงให้เป็นแบบ Discrete Differential Equation ดังนั้นสมการ PID ที่นำมาใช้ในการควบคุมเป็นดังนี้

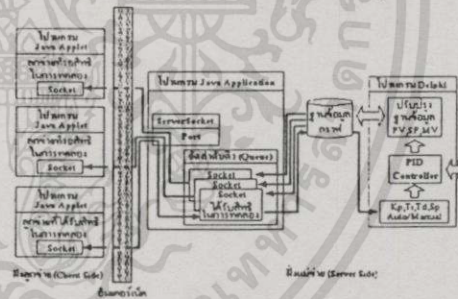
$$Vo_n = Vo_{n-1} + K_p \left[(e_n - e_{n-1}) + \frac{\Delta T}{T_i} e_n + \frac{T_d}{\Delta T} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad (2)$$

- เมื่อ ΔT : ช่วงเวลาของการสุ่มตัวอย่าง
- Vo_n : เอาพุตของเครื่องควบคุมรอบที่ n
- e_n : ค่าความผิดพลาดในการสุ่มรอบที่ n

3. การออกแบบโปรแกรม

3.1 การส่งถ่ายข้อมูลระหว่างแม่ข่าย(server)กับลูกข่าย(client)

งานวิจัยได้ใช้โปรแกรมภาษา Java [5,8] โดยพัฒนา Java Application เพื่อให้เป็นโปรแกรมแม่ข่ายและ Java Applet เป็นโปรแกรมของผู้ทดลองหรือลูกข่ายการส่งถ่ายข้อมูลระหว่างแม่ข่ายและลูกข่ายจะอาศัย Socket ต่าง ๆ ซึ่งถือเป็นการกระบวนการของเครือข่าย (network) ที่จะใช้เชื่อมโยงระหว่างแม่ข่ายและลูกข่ายต่าง ๆ โดย Socket จะถูกกำหนดให้สื่อสารบนพอร์ต (Port) ที่กำหนดไว้บนแม่ข่าย ดังรูปที่ 3



รูปที่ 3 โครงสร้างการส่งถ่ายข้อมูลระหว่างแม่ข่ายและลูกข่าย

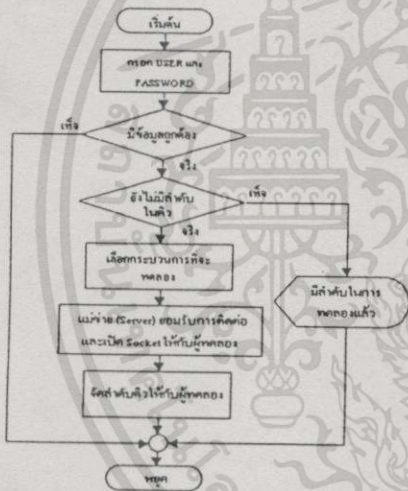
3.2 การออกแบบฐานข้อมูลเพื่อแสดงผลกราฟ

เพื่อให้เกิดความเที่ยงตรงในการแสดงผลกราฟ จึงได้ออกแบบให้ฐานข้อมูล[7]ให้ฟิลด์ (field) ประกอบด้วย Time, PV, MV และ SP เพื่อเป็นข้อมูลในแกน Y ของกราฟ ส่วนเรคอร์ด (record) ได้ออกแบบให้มี 60 เรคอร์ด ซึ่งแทนข้อมูลจำนวน 60 วินาทีในแกน X ของกราฟ ฐานข้อมูลจะถูกปรับปรุง (update) ทุก ๆ วินาทีจากโปรแกรมภาษา Delphi [7] ซึ่ง Delphi ถูกพัฒนาเพื่อทำหน้าที่อินเตอร์เฟสกับอุปกรณ์ภายนอกเพื่อมา

คำนวณตามโปรแกรมควบคุม PID จากนั้นจึงปรับปรุงข้อมูลลงในฐานข้อมูล โปรแกรม Java Application จะส่งข้อมูลในฐานข้อมูลทั้งหมดไปให้กับผู้ทดลองหรือลูกข่าย ทุกเครื่อง ข้อมูลที่ส่งไปจะถูกนำไปแสดงผลในรูปภาพเส้น จากโปรแกรม Java Applet ที่ลูกข่ายอีกที

3.3 โปรแกรมเพิ่มคิวของผู้ต้องการทดลอง

ผู้ที่จะทดลองต้องผ่านการ Login ก่อนเสมอ ถ้าเป็นผู้ที่ได้รับอนุญาตให้ทดลอง (มีรายชื่ออยู่ในฐานข้อมูลของผู้บริหารระบบ) และต้องเลือกกระบวนการที่ต้องการทดลองระหว่างกระบวนการควบคุมระดับน้ำและควบคุมอุณหภูมิ จากนั้น โปรแกรมแม่ข่ายก็จะส่งข้อมูลเพื่อไปแสดงผลและมีการจัดลำดับคิว[6]ให้กับผู้ทดลอง

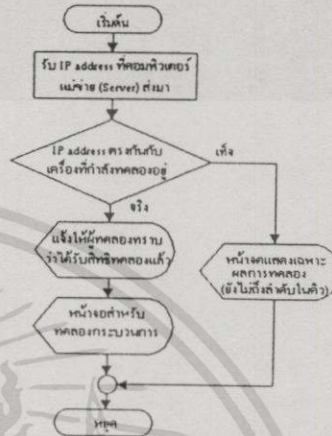


รูปที่ 4 FlowChart ของการเพิ่มคิวเพิ่มคิวของผู้ทดลอง

3.4 โปรแกรมตรวจสอบลำดับคิวเพื่อให้สิทธิในการทดลอง

นอกจากคอมพิวเตอร์แม่ข่ายทำงานเพื่อส่งข้อมูลกลับไปแสดงผลที่หน้าจอของผู้ทดลองแล้ว คอมพิวเตอร์แม่ข่ายยังส่งหมายเลข IP address [5,8] ของผู้ที่ได้รับสิทธิในการทดลองปัจจุบันออกไปด้วย โดยโปรแกรม Java Applet ของผู้ทดลองแต่ละคนก็จะทำการเปรียบเทียบกับแอดเรสอินเตอร์เน็ตของตนเองและเมื่อแอดเรสตรงกัน โปรแกรม Java Applet ก็จะบอกให้ผู้ทดลองท่านนั้นทราบว่าได้รับสิทธิในการทดลองแล้ว พร้อมทั้งปรับปรุงหน้าจอ

ควบคุมให้สามารถปรับเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ในการทดลองได้



รูปที่ 5 FlowChart ของการตรวจสอบสิทธิในการทดลอง

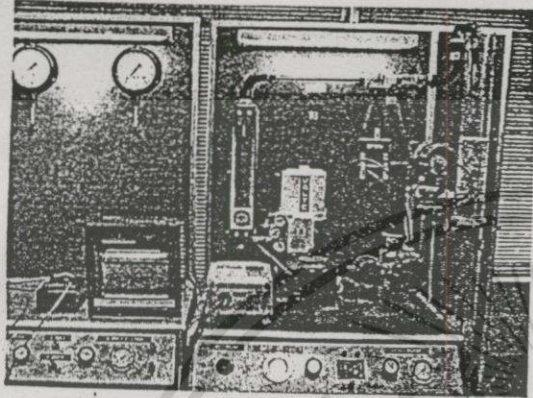
3.5 โปรแกรมสำหรับการยกเลิกการทดลอง

สิทธิในการทดลองแต่ละครั้ง ถูกกำหนดเวลาให้สามารถทดลองได้ภายในเวลาที่กำหนดไว้ (ผู้บริหารระบบเป็นผู้กำหนดเวลา) ทั้งนี้ระยะเวลาที่อนุญาตให้ทดลองจะขึ้นอยู่กับความเหมาะสมของกระบวนการและเมื่อสิ้นสุดเวลาที่อนุญาตให้ทดลองได้แล้ว คอมพิวเตอร์แม่ข่ายก็จะแจ้งให้ผู้ทดลองดังกล่าวทราบพร้อมทั้งยกเลิกการทดลอง นอกจากนี้ในการยกเลิกการทดลองอาจเกิดได้จากการที่ผู้ทดลองปิดโปรแกรมเบราว์เซอร์หรือการสื่อสารบนอินเตอร์เน็ตระหว่างคอมพิวเตอร์แม่ข่ายและผู้ทดลองไม่สามารถติดต่อกันได้ คอมพิวเตอร์แม่ข่ายก็จะทำการยกเลิกการติดต่อผ่านทาง Socket ดังกล่าวนั้น และจะทำการปรับปรุงลำดับคิวใหม่

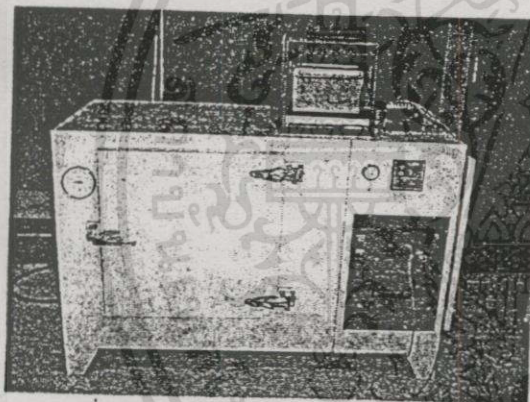
4. ผลการทดลอง

การทดลองทำโดยการบันทึกผลการควบคุมกระบวนการ (recorder) จากเบราว์เซอร์ของผู้ทดลองและที่คอมพิวเตอร์แม่ข่ายเองก็ได้บันทึกผลของตัวแปรกระบวนการ (ระดับน้ำและอุณหภูมิ)เอาไว้โดยใช้เครื่อง Recorder ดังรูปที่ 6 และรูปที่ 7 เป็นชุดทดลองที่ติดตั้งที่มหาวิทยาลัย ส่วนในรูปที่ 8 และ 10 เป็นกราฟของการควบคุมโดยใช้ Recorder ที่ตั้งอยู่กับชุดทดลองของทางมหาวิทยาลัย รูปที่ 9 และ 11 เป็นผลการ

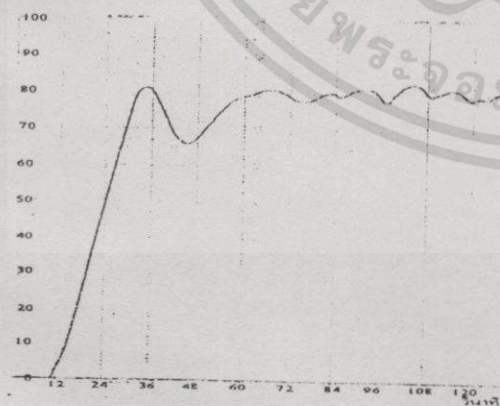
บันทึกการทดลองของผู้ทดลองจากเบรนาเซอร์ ซึ่งผลการบันทึกสัญญาณจากเบรนาเซอร์ได้ให้ผลความถูกต้องเช่นเดียวกับเครื่อง Recorder ที่ใช้ในห้องทดลองของทางมหาวิทยาลัย



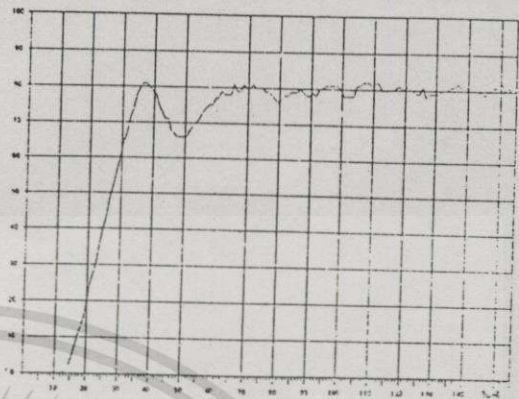
รูปที่ 6 ชุดทดลองกระบวนการควบคุมระดับน้ำ



รูปที่ 7 ชุดทดลองกระบวนการควบคุมอุณหภูมิ



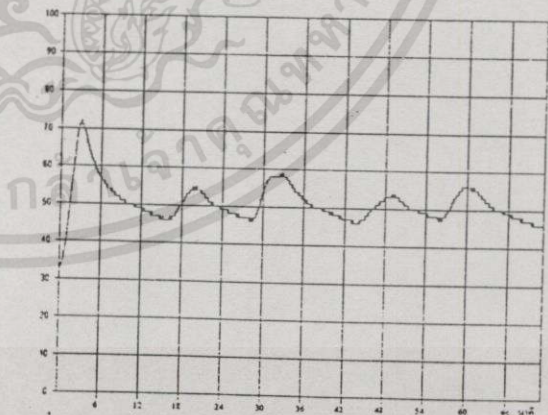
รูปที่ 8 การบันทึกจาก Recorder ของการควบคุมระดับน้ำ



รูปที่ 9 เป็นผลการบันทึกการทดลองการควบคุมระดับน้ำของผ่านทางเบรนาเซอร์



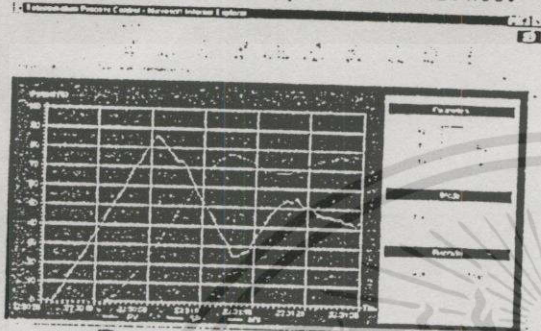
รูปที่ 10 การบันทึกจาก Recorder ของการควบคุมอุณหภูมิ



รูปที่ 11 เป็นผลการทดลองการควบคุมอุณหภูมิของผ่านทางเบรนาเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ทดลองสามารถทำการทดลองกระบวนการผ่านทาง อินเทอร์เน็ตโดยอาศัยเบราว์เซอร์ดังรูปที่ 12 ซึ่งทำหน้าที่ เป็นโปรแกรมควบคุมของทั้งกระบวนการควบคุมระดับน้ำ และกระบวนการควบคุมอุณหภูมิโดยส่งพารามิเตอร์ที่ใช้ ควบคุมไปที่คอมพิวเตอร์แม่ข่ายของทางมหาวิทยาลัยและ รับผิดชอบสนองของการควบคุมมาแสดงผลที่เบราว์เซอร์



รูปที่ 12 เป็นการทดลองกระบวนการผ่านทางอินเทอร์เน็ต

5. สรุปผลการทดลอง

การทดลองกระบวนการควบคุมผ่านทาง อินเทอร์เน็ตโดยอาศัยเบราว์เซอร์ทำให้การที่จะทดลอง สามารถทำได้ง่ายไม่จำเป็นต้องใช้โปรแกรมเฉพาะใดๆ เข้า มาช่วยเพื่อการทดลอง การบันทึกผลการทดลอง(recorder) กระบวนการดังกล่าว สามารถทำได้จากหน้าจอของผู้ ทดลองเองซึ่งจะลดภาระการทำงานของแม่ข่ายได้ การ แสดงผลการทดลองในกราฟจะใช้วิธีการดึงข้อมูลต่าง ๆ ที่ ได้จากการทดลองทั้งหมดในช่วงเวลา 60 วินาทีจากฐาน ข้อมูลที่คอมพิวเตอร์แม่ข่ายมาแสดง ข้อมูลที่ได้จากการ ทดลองจะถูกบันทึกจากเครื่องควบคุมที่ตั้งในห้องทดลอง ทุก ๆ วินาทีซึ่งไม่ได้นำเอาเฉพาะข้อมูลปัจจุบันมาแสดงผล ในกราฟเพียงอย่างเดียวจึงทำให้การแสดงผลในหน้าจอ ของผู้ทดลองนั้นมีความเที่ยงตรงและเหมือนกันทุกคน ดังนั้นความผิดพลาดในการสื่อสารของข้อมูลบนอินเทอร์เน็ตจึงมีผลเพียงการส่งถ่ายข้อมูลจากแม่ข่ายมายังลูกข่าย ต่าง ๆ แต่ความถูกต้องของข้อมูลจากการทดลองที่นำไป

แสดงผลยังคอมพิวเตอร์ลูกข่ายนั้น ยังคงถูกต้องสมบูรณ์ทุก ประการ

6. เอกสารอ้างอิง

- [1]. C. Lon Enloe, W. A. Pakula, G. A. Finney, and R. K. Haaland, "Teleoperation in the Undergraduate Physics Laboratory - Teaching an Old Dog New Ties," IEEE Trans. Education. Vol. 42, no. 3 pp. 174-179, August, 1999.
- [2]. Y. Piguot and D. Gillet, "Java - based remote experimentation for control algorithms prototyping," Proc. the American Control Conference San Diego, California., pp.1465-1469, June 1999.
- [3]. G. H. Salazar-Silva, J. C. Martinez-Garcia and R. Garrido, "Enhancing Basic Robotics Education on the Web," Proc. the American Control Conference San Diego, California., pp.1470-1471, June 1999.
- [4]. D. E. Seborg, T. F. Edgar and D. A. Mellicamp, "Process Dynamics And Control," John Wiley & Sons, Inc, pp.195-196, 1989.
- [5]. M. Hughes, C. Hughes, M. Shoffner and M. Winslow, "Java Network Programming," Manning Publications Co., 1997.
- [6]. สุรชาติ พ่วงพุ่ม, "โครงสร้างข้อมูลกับปาสคาล," สถาบัน เทคโนโลยีพระจอมเกล้าพระนครเหนือ, หน้า 79-84, 2535.
- [7]. กิตติ เปรมพิณีจ, "คู่มือการใช้งาน Borland Delphi 4," เคพีเอ็น ซิสเต็มอินทิเกรเตอร์, ตุลาคม 2541.
- [8]. พ.อ. เจนวิทย์ เหลืองอร่าม, "การเขียนโปรแกรมสำหรับ Applications และ Applets ด้วย Java," บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพฯ. 2542.

ประวัติผู้เขียน

นายสรรเพชญ เชื้อแก้ว เกิดเมื่อวันที่ 4 มีนาคม 2519 ที่จังหวัดระนอง สำเร็จการศึกษาในระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาเทคโนโลยีการควบคุมทางอุตสาหกรรม จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2541 และในปีการศึกษา 2542 ได้เข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้