สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่

Home security and electric appliances controlled via mobile phone

โดย

นางสาวปรานี เฉลยกุล

นางสาวพฤกษชาติ น้อยกลาง

รูปเล่มถูกต้อง.

เลขหมู่.....
เลขทะเบียน 62018
วัน,เดือน,ปี 25 ก.ค. 2549
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เสร็จแล้ว

ภาควิชา
วิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่

Home security and electric appliances controlled via mobile phone

โดย

นางสาวปรานี เฉลยกุล 44010292

นางสาวพฤกษชาติ น้อยกลาง 44010320

อาจารย์ที่ปรึกษา

รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี

รศ.ดร.ปัญญา ฐิติมัชฌิมา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่**

Home security and electric appliances controlled via mobile phone

ผู้จัดทำ

1. นางสาวปรานี เฉลยกุล 44010292

2. นางสาวพฤกษชาติ น้อยกลาง 44010320

..... อาจารย์ที่ปรึกษา

(รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี)

..... อาจารย์ที่ปรึกษา

(รศ.ดร.ปัญญา ฐิติมัชฌิมา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่

Home security and electric appliances controlled via mobile phone

โดย นางสาวปรานี เฉลยกุล 44010292

นางสาวพฤกษชาติ น้อยกลาง 44010320

อาจารย์ที่ปรึกษา รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี

รศ.ดร.ปัญญา ฐิติมัชฌิมา

บทคัดย่อ

โครงงานนี้ทำขึ้นเพื่อศึกษาถึงการนำเอาเทคโนโลยีสมัยใหม่ มาประยุกต์เป็นช่องทางเพื่อสร้างระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้า โดยจะทำการศึกษาการควบคุมผ่านโทรศัพท์เคลื่อนที่ โดยใช้แว็บแอพลิเคชันไปยังคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์และเชื่อมโยงการทำงานกับไมโครคอนโทรลเลอร์ เพื่อมาทำการควบคุมอุปกรณ์ไฟฟ้า

Abstract

This project objective is to study about applying a new technology to be an interface that create security system and control electric appliances. This project chooses controlling via wap application of interface with computer server that connects operating of electric appliances those were controlled by microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	
	2.1 แวป	4
	2.2 Java Programming	15
	2.3 การสื่อสารข้อมูล (Data communications)	17
	2.4 ไมโครคอนโทรลเลอร์ 8051	21
บทที่ 3	การคำนวณและการสร้าง	
	3.1 วงจรกำเนิดแรงดัน	30
	3.2 วงจรแปลงสัญญาณแบบ RS-232	30
	3.3 วงจร ไมโครคอนโทรลเลอร์	31
	3.4 วงจรเตือนน้ำเต็ม	33
	3.5 วงจรสวิทซ์	33
	3.6 วงจรตรวจจับทางแสง	34
	3.7 วงจรควบคุมอุณหภูมิ	36
	3.8 ส่วนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์	38
	3.9 ส่วนโปรแกรมเชื่อมต่อการทำงานผ่านโครงข่ายอินเตอร์เน็ต และโครงข่ายโทรศัพท์เคลื่อนที่	42
บทที่ 4	การทดลองและผลการทดลอง	
	4.1 การทดลองวงจรแปลงสัญญาณแบบ RS-232	46
	4.2 วงจรเตือนน้ำเต็ม	47
	4.3 วงจรสวิทซ์	48
	4.4 วงจรตรวจจับทางแสง	49
	4.5 วงจรควบคุมอุณหภูมิ	50
	4.6 การใช้งานระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต	51
	4.7 การใช้งานระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่	59
บทที่ 5	บทสรุปและวิเคราะห์	67
	ภาคผนวก	69
	หนังสืออ้างอิง	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 การทำงานโดยรวมของระบบ	2
รูปที่ 2.1 แบบจำลองชั้นสื่อสาร TCP/IP (TCP/IP Protocol Stack)	7
รูปที่ 2.2 โปรโตคอลที่เกี่ยวของกับการร้องขอและตอบกลับระหว่างเว็บเบราเซอร์กับเว็บเซิร์ฟเวอร์	10
รูปที่ 2.3 แวปเกตเวย์ทำหน้าที่เป็นตัวกลางระหว่างเครือข่ายอินเตอร์เน็ตและเครือข่ายไร้สาย	11
รูปที่ 2.4 ขั้นตอนการทำงานและร้องขอเอกสารในระบบแวปและอินเตอร์เน็ตผ่านแวปเกตเวย์	12
รูปที่ 2.5 โครงสร้างพื้นฐานของ แวปเกตเวย์	13
รูปที่ 2.6 ภาพรวมของโปรโตคอลที่เกี่ยวข้องในเครือข่ายไร้สายและเครือข่ายอินเตอร์เน็ต	14
รูปที่ 2.7 แสดงการทำงานของเซิร์ฟเลต	16
รูปที่ 2.8 สัญญาณการรับและส่งข้อมูลในโหมด 0	23
รูปที่ 2.9 การรับและส่งข้อมูลในโหมด 1	24
รูปที่ 2.10 การรับและส่งข้อมูลในโหมด2 และ3	24
รูปที่ 3.1 วงจรกำเนิดแรงดัน	30
รูปที่ 3.2 วงจรแปลงสัญญาณแบบ RS-232	31
รูปที่ 3.3 วงจรรวมของไมโครคอนโทรลเลอร์	32
รูปที่ 3.4 การต่ออุปกรณ์ที่ขาต่างๆของรีเลย์	32
รูปที่ 3.5 วงจรเตือนน้ำเต็ม	33
รูปที่ 3.6 วงจรสวิทช์	34
รูปที่ 3.7 วงจรตรวจจับทางแสง	35
รูปที่ 3.8 วงจรตรวจจับทางแสง	36
รูปที่ 3.9 วงจรควบคุมอุณหภูมิ	37
รูปที่ 3.10 การเซตค่าในส่วนของ Tools, Configure User Tools ...	38
รูปที่ 3.11 การเซตค่าในส่วนของ User Tools ...	39
รูปที่ 3.12 การเซตค่าในส่วนของ Setting & syntax	39
รูปที่ 3.13 การคอมไพล์โปรแกรมแอสแซมบลี	40
รูปที่ 3.14 การแสดง error เมื่อทำการคอมไพล์เสร็จแล้ว	40
รูปที่ 3.15 โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์	41
รูปที่ 3.16 หน้าต่างโปรแกรม Netbeans	43
รูปที่ 3.17 โฟลว์ชาร์ตโปรแกรมรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 3.18 โฟลว์ชาร์ตโปรแกรมรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ตและโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา(ต่อ)	45
รูปที่ 4.1 หน้าต่างของโปรแกรมส่งค่า"1"	46
รูปที่ 4.2 สัญญาณที่รับมาจากคอมพิวเตอร์ของขา RxD ของไอซี MAX 232	46
รูปที่ 4.3 สัญญาณที่ส่งกลับไปยังคอมพิวเตอร์จากขา TxD ของไอซี MAX 232	47
รูปที่ 4.4 แสดงผลการทดลองขณะที่น้ำยังไม่ท่วมถึงโพรบ	47
รูปที่ 4.5 แสดงผลการทดลองขณะที่น้ำท่วมถึงโพรบ	48
รูปที่ 4.6 แสดงผลของวงจรสวิทซ์ในสถานะปกติ	48
รูปที่ 4.7 แสดงผลของวงจรสวิทซ์เมื่อสวิทซ์ถูกกด	49
รูปที่ 4.8 แสดงผลของวงจรขณะห้องสว่าง	49
รูปที่ 4.9 แสดงผลของวงจรขณะห้องมืด	50
รูปที่ 4.10 แสดงผลของวงจรที่อุณหภูมิห้อง	50
รูปที่ 4.11 แสดงผลของวงจรเมื่ออุณหภูมิของห้องสูงขึ้น	51
รูปที่ 4.12 การจำลองคอมพิวเตอร์เซิร์ฟเวอร์	51
รูปที่ 4.13 แสดงหน้าจอหลักระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต	52
รูปที่ 4.14 แสดงหน้าจอ Login	52
รูปที่ 4.15 แสดงหน้าจอเมื่อใส่ Username และ Password ไม่ถูกต้อง	53
รูปที่ 4.16 แสดงหน้าจอเลือก Menu	53
รูปที่ 4.17 แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล	54
รูปที่ 4.18 แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล	54
รูปที่ 4.19 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้า	55
รูปที่ 4.20 แสดงหน้าผลการเปิด-ปิดอุปกรณ์ไฟฟ้า	55
รูปที่ 4.21 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลา	56
รูปที่ 4.22 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลาเมื่อตั้งเวลาเรียบร้อยแล้ว	56
รูปที่ 4.23 แสดงหน้า Environment & Sensor เมื่ออยู่ในสภาวะปกติ	57
รูปที่ 4.24 แสดงสถานะของเซนเซอร์เมื่อประตูที่ 1 เปิด	57
รูปที่ 4.25 แสดงหน้าจอ Setting	58
รูปที่ 4.26 แสดงหน้าผลการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า	58
รูปที่ 4.27 แสดงหน้าผลการ Logout	59
รูปที่ 4.28 แสดงหน้าจอของโทรศัพท์เคลื่อนที่	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 4.29 แสดงเมนูของโปรแกรมต่างๆ 60
รูปที่ 4.30 แสดงหน้าจอของโปรแกรมโอเปร่า 60
รูปที่ 4.31 แสดงหน้าการลิงค์ไปยังแวปเพจ 60
รูปที่ 4.32(ก)แสดงหน้าหลักของแวปเพจ 60
รูปที่ 4.32(ข) แสดงหน้าหลักของแวปเพจ 60
รูปที่ 4.33 แสดงหน้า Login 61
รูปที่ 4.34 เมื่อใส่ Username และ Password 61
รูปที่ 4.35 แสดงหน้าเมื่อใส่ Username และ Password ไม่ถูกต้อง 61
รูปที่ 4.36 แสดงหน้าเลือก Menu 61
รูปที่ 4.37(ก) แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล 62
รูปที่ 4.37(ข) แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล 62
รูปที่ 4.38 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้า 62
รูปที่ 4.39 แสดงหน้าผลการเปิด-ปิดอุปกรณ์ไฟฟ้า 62
รูปที่ 4.40 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลา 63
รูปที่ 4.41 แสดงหน้าเลือกเวลาที่จะทำการปิด-เปิดอุปกรณ์ไฟฟ้า 63
รูปที่ 4.42 แสดงหน้าเลือกเวลาที่จะทำการปิด-เปิดอุปกรณ์ไฟฟ้าเรียบร้อยแล้ว 63
รูปที่ 4.43 แสดงหน้าผลของการตั้งเวลาปิด-เปิดอุปกรณ์ไฟฟ้า 63
รูปที่ 4.44 แสดงหน้า Environment & Sensor เมื่ออยู่ในสถานะปกติ 64
รูปที่ 4.45 แสดงหน้า Environment & Sensor เมื่อประตูที่ 1 เปิด 64
รูปที่ 4.46 แสดงหน้า Setting 64
รูปที่ 4.47 แสดงหน้าการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า 64
รูปที่ 4.48 แสดงหน้าการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าเรียบร้อยแล้ว 65
รูปที่ 4.49 แสดงหน้าผลการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า 65
รูปที่ 4.50 แสดงหน้าของ menu เมื่อเลือก Logout 65
รูปที่ 4.51 แสดงหน้าจอเมื่อทำการ Logout เรียบร้อยแล้ว 65
รูปที่ 4.52 แสดงหน้าจอเมื่อรับข้อความแจ้งเตือนเข้ามา 66
รูปที่ 4.53 แสดงข้อความรายงานความผิดปกติจากเซนเซอร์ 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 การเลือกการทำงานของ Timer/Counter	26
ตารางที่ 2.2 Baud Rate ต่างๆ และค่า Reload ของ Time 1	27
ตารางที่ 2.3 ค่าเริ่มต้นของรีจิสเตอร์ SFR เมื่อ 8051 ถูกรีเซ็ต	28
ตารางที่ 2.4 ค่าเริ่มต้นของรีจิสเตอร์ SFR เมื่อ 8051 ถูกรีเซต	29
ตารางที่ 3.1 การทดลองวัดค่าความต้านทานของ LDR	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ที่มาของโครงงาน

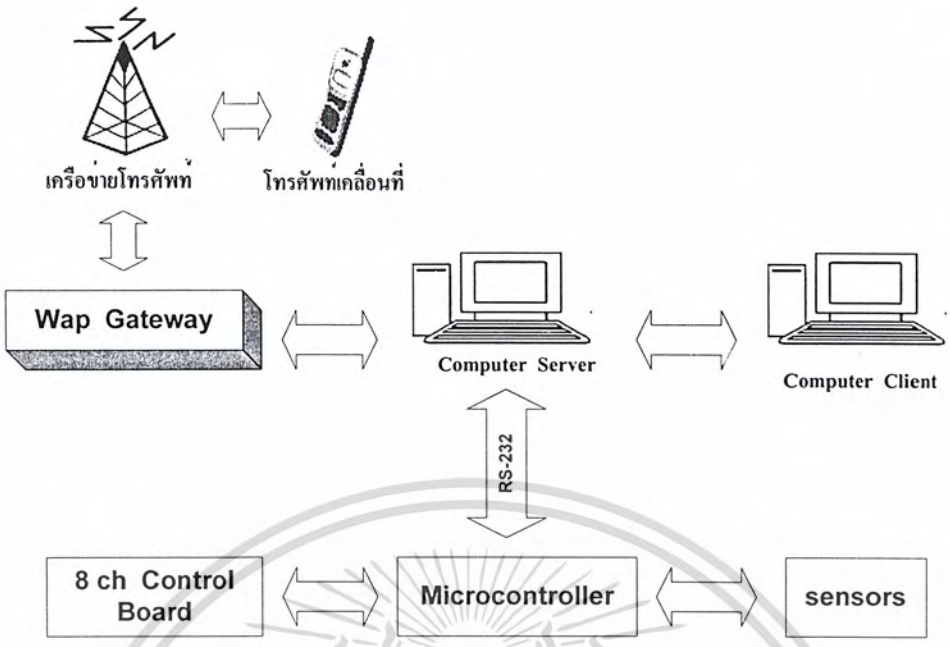
ปัจจุบันนี้แนวโน้มของเทคโนโลยีทางด้านคอมพิวเตอร์และอินเตอร์เน็ต จะเป็นไปในรูปของเทคโนโลยีไร้สาย (Wireless Technology) สืบเนื่องมาจากความต้องการบริโภคอินเตอร์เน็ต (internet) ที่เพิ่มขึ้นอย่างรวดเร็ว ขณะเดียวกันในภาพรวมของโลกของเรา ก็มีจำนวนการใช้โทรศัพท์มือถือมากกว่าคอมพิวเตอร์อย่างมาก ดังนั้นจึงมีแนวคิดที่จะนำโทรศัพท์เคลื่อนที่มาเป็นเครื่องลูกข่ายสำหรับการเรียกดูข้อมูลในอินเตอร์เน็ตอันเป็นที่มาของคำว่า Mobile Internet หรือ อินเตอร์เน็ตไร้สาย นอกจากนี้ส่วนของโทรศัพท์เคลื่อนที่ก็มีเทคโนโลยีแวป (WAP:Wireless Application Protocol) ซึ่งเกิดจากการเพิ่มคุณค่าของโทรศัพท์มือถือให้มีมากกว่าการเป็นแค่อุปกรณ์การสื่อสารด้วยเสียง (voice) และบริการข้อความสั้นๆ (Short Message Service) โดยเพิ่มคุณสมบัติการสื่อสารข้อมูลเข้ามาด้วย ซึ่งนำไปสู่การพัฒนาการใช้งานโทรศัพท์เคลื่อนที่ให้เกิดประโยชน์สูงสุดดังเช่น การควบคุมเปิด-ปิดอุปกรณ์ไฟฟ้า หรือตรวจสอบระบบรักษาความปลอดภัย (sensors) ต่างๆ ผ่านโทรศัพท์เคลื่อนที่ จึงเป็นที่มาของปริญญานิพนธ์ฉบับนี้จึงได้ทำการศึกษาและจำลองระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่ซึ่งรองรับเทคโนโลยีแวป

หลักการทำงานของโครงงาน

หลักการเบื้องต้นของโครงงานนี้คือสร้างคอมพิวเตอร์เซิร์ฟเวอร์ (Computer Server) ส่วนกลางให้ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ (Web Server) และเป็นส่วนประมวลผลข้อมูลจากการสั่งงานและส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 โดยมีไมโครคอนโทรลเลอร์ (microcontroller)เป็นตัวควบคุมการทำงานของบอร์ด

นอกจากนี้ระบบยังสามารถควบคุมอุปกรณ์ไฟฟ้าและตรวจสอบสถานะของเซนเซอร์(sensors) ผ่านเครือข่ายโทรศัพท์เคลื่อนที่และเครือข่ายอินเตอร์เน็ต โดยใช้เทคโนโลยีจาวาเซิร์ฟเลต (Java Servlet)ในการสร้างแวปเพจและเว็บเพจ ดังรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 การทำงานโดยรวมของระบบ

ขอบเขตของโครงงาน

1. เพื่อศึกษาการควบคุมอุปกรณ์ไฟฟ้า ซึ่งทำการปิด-เปิดรีเลย์ (relay)แบบทันทีและแบบตั้งเวลาโดยอาศัยการควบคุมของไมโครคอนโทรลเลอร์
2. ศึกษาและทดลองการเชื่อมต่อการทำงานร่วมกันระหว่างโปรมแกรมภาษาจาวา(Java)กับไมโครคอนโทรลเลอร์ ผ่านพอร์ตอนุกรม RS-232
3. ศึกษาการทำงานของแวป และ HTML และทดลองการเชื่อมต่อการทำงานระหว่างโปรแกรมภาษาจาวากับแวป และ HTML

ขั้นตอนการดำเนินงาน

1. แบ่งโครงงานเป็น 4 ส่วน
   - 1.1 ส่วนฮาร์ดแวร์ (Hardware) ซึ่งจะควบคุมโดยไมโครคอนโทรลเลอร์
   - 1.2 ส่วนฮาร์ดแวร์ซึ่งเป็นระบบเซนเซอร์
   - 1.3 ส่วนโปรแกรมที่ติดต่อระหว่างคอมพิวเตอร์เซิร์ฟเวอร์กับฮาร์ดแวร์
   - 1.4 ส่วนโปรแกรมที่ทำงานระหว่างคอมพิวเตอร์เซิร์ฟเวอร์กับโทรศัพท์เคลื่อนที่
2. ออกแบบบล็อกไดอะแกรม (Block Diagram) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ออกแบบวงจรและเขียนโฟล์วชาร์ต (Flow Chart)
4. สร้างวงจรไมโครคอนโทรลเลอร์ วงจรรีเลย์ วงจรแปลงสัญญาณแบบ RS-232 และวงจรเซนเซอร์ต่างๆ
5. ดำเนินการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์โดยใช้ภาษาแอสแซมบลี (Assembly)
6. ดำเนินการเขียนโปรแกรมติดต่อกับพอร์ตอนุกรม RS-232 โดยใช้ภาษาจาวา
7. ทดสอบการทำงานร่วมกันระหว่างโปรแกรมติดต่อกับพอร์ตอนุกรม RS-232 โดยใช้ภาษาจาวา และ โปรแกรมควบคุมไมโครคอนโทรลเลอร์โดยใช้ภาษาแอสแซมบลี
8. ดำเนินการเขียนโปรแกรมติดต่อกันระหว่างคอมพิวเตอร์เซิร์ฟเวอร์และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา
9. ทดสอบการทำงานโปรแกรมติดต่อกันระหว่างคอมพิวเตอร์เซิร์ฟเวอร์และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา
10. ทดสอบการทำงานร่วมกันทั้งระบบ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 แวป

แวป นับว่าเป็นเทคโนโลยีที่เป็นรอยต่อของการเปลี่ยนแปลงครั้งสำคัญ จากการใช้อินเตอร์เน็ตแบบเดิม มาเป็นการใช้อินเตอร์เน็ตผ่านโทรศัพท์มือถือซึ่งเป็นสิ่งใหม่ที่คนทั่วไปไม่คุ้นเคยมาก่อน โดยจะพูดถึงความหมายของแวป ใน 3 ประเด็นหลักดังนี้

แวป คือ บริการ

ความหมายนี้มักใช้ในเชิงพาณิชย์เป็นส่วนมาก คำว่า บริการ(services) ในที่นี้ ก็คือการเรียกใช้(access) ข้อมูลในอินเตอร์เน็ตด้วยโทรศัพท์มือถือ ซึ่งสิ่งที่สำคัญที่สุดก็คือ โทรศัพท์มือถือรุ่นนั้นๆ ต้องมีการรองรับเทคโนโลยีแวป ด้วย คือ ภายในเครื่องต้องมีซอฟท์แวร์ (Software)ที่เรียกว่า แวปเบราเซอร์ (WAP Browser) หรือ ไมโครเบราเซอร์ (Micro Browser) และฮาร์ดแวร์ของเครื่องที่สนับสนุนการทำงาน รวมถึงปัจจัยอีกอย่างหนึ่ง คือ ทางผู้ผลิตหรือผู้ให้บริการเครือข่ายต้องมีแวปเกตเวย์ (WAP Gateway) ไว้ให้บริการด้วย โดยแวปเกตเวย์ เป็นอุปกรณ์ที่มีความหมายมากในระบบแวป ถึงแม้ว่าในความจริงจะมีแวปเกตเวย์สาธารณะ ให้บริการอยู่ก็ตาม แต่ถือว่าการลงทุนติดตั้งแวปเกตเวย์ เป็นความน่าเชื่อถือ เครดิต หรือกลไกการแข่งขันทางตลาดที่ ผู้ผลิตโทรศัพท์มือถือเหล่านั้นต้องแสดงศักยภาพออกมา

แวป คือ เทคโนโลยี

ความหมายนี้เป็นการพูดอย่างกว้างขวางมาก แต่ก็สามารถเข้าใจได้ง่าย เพราะในมุมมองหนึ่งแวป เป็นเทคโนโลยีการสื่อสารแบบไร้สายที่ช่วยให้สามารถเข้าใช้งานอินเตอร์เน็ตได้ด้วยโทรศัพท์มือถือ ซึ่งนับว่า แวปเป็นรอยต่อของเทคโนโลยี เพราะเป็นจุดเปลี่ยนจากการสื่อสารข้อมูลด้วยเครื่องคอมพิวเตอร์เพียงอย่างเดียว มาเป็นการรวมเอาอุปกรณ์สื่อสารไร้สายอื่นๆเข้ามาในระบบสื่อสารข้อมูลด้วย และนอกจากแวป แล้วยังมีเทคโนโลยีสื่อสารไร้สายแบบอื่นๆ เป็นทางเลือกเพิ่มเติมอีก ที่เด่นๆจะได้แก่ “I-Mode” และ “Bluetooth”

แวป คือ โปรโตคอล

และนี่คือ ความหมายที่แท้จริง และถูกต้องที่สุด คำว่า แวป ซึ่งย่อมาจาก Wireless Application Protocol เป็นข้อกำหนดในการสื่อสารในเครือข่ายไร้สาย สำหรับคำอธิบายในเบื้องต้นถึงคำว่า แวป และโปรโตคอลก็คือ

ในการใช้งานอินเตอร์เน็ตด้วยเครื่องคอมพิวเตอร์ ทำให้เราคุ้นเคยกับคำว่า “HTTP”, “TCP” หรือ “IP” สิ่งเหล่านี้ก็คือ ข้อกำหนด ในระหว่างการสื่อสารระหว่างคอมพิวเตอร์ในโลกของอินเตอร์เน็ต หรือโปรโตคอล (Protocol) นั่นเอง ซึ่งเป็นกลไกสำคัญที่ทำให้เราติดต่อสื่อสารผ่านเครือข่ายระดับโลกนี้ได้ แต่เมื่อเราเปลี่ยนจากการใช้เครื่องคอมพิวเตอร์ ที่อาศัยเครือข่ายแบบเดิม มาเป็นโทรศัพท์มือถือ ที่อาศัยเครือข่ายไร้สายเป็นสื่อกลางแทน เช่น เครือข่ายโทรศัพท์ GSM ที่เรารู้จักกันดี ก็ต้องมีการตั้งข้อกำหนดหรือกฎเกณฑ์ใหม่ขึ้นมาสำหรับใช้ในสภาวะแวดล้อมใหม่นี้ ซึ่งข้อกำหนดหรือโปรโตคอลที่สร้างขึ้นมาใหม่นี้ ก็มีชื่อว่า แวปนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าจะเป็น HTTP, TCP, IP หรือ WAP ล้วนแต่เป็นโปรโตคอลทั้งสิ้น แต่มีหน้าที่แตกต่างกันไป โดย WAP มีลักษณะแปลกไปจากโปรโตคอลอื่นๆ ตรงที่ทำงานในสภาวะแวดล้อมแบบไร้สาย

ส่วนผู้ที่คิดค้นหรือสร้างข้อกำหนด กฎเกณฑ์ที่ชื่อว่า WAP ก็คือ กลุ่มบริษัทชั้นนำ เช่น Phone.com, อีริคสัน โนเกีย โมโตโรลา เป็นต้น ซึ่งเรียกว่า WAP Forum

2.1.1 WAP Forum ผู้กำหนดมาตรฐานของ WAP

ในปี 1995 สหรัฐอเมริกามีบริษัทที่ชื่อว่า Unwired Planet ซึ่งค้นพบภาษา DHTML (Hand-held Markup Language) ซึ่งเป็นเวอร์ชันลดรูปมาจากภาษา HTML และนำมาใช้กับอุปกรณ์ไร้สายแบบต่างๆ ซึ่งรวมถึงโทรศัพท์มือถือด้วย แต่ภาษานี้มีความแพร่หลายในระดับหนึ่งเท่านั้น

จากนั้นราวปี 1996-1997 บริษัท Unwired Planet ซึ่งเปลี่ยนใหม่เป็น Phone.com ก็ได้มีโครงการปรับปรุงเทคโนโลยีอินเตอร์เน็ตในเครือข่ายไร้สาย โดยร่วมมือกับ บริษัทผู้ผลิตมือถือทั้งหลาย ซึ่งวางแนวคิดเบื้องต้นไว้คือ ต้องเป็นระบบเปิด และคาดหวังจะได้รับการสนับสนุนจากทั่วโลก ดังนั้นจึงเป็นที่มาของกลุ่ม WAP Forum

บริษัทที่ร่วมกับ Phone.com ในยุคแรกๆคือ อีริคสัน โนเกีย และ โมโตโรลา จนถึงกลางปี 2000 ได้เพิ่มถึงราว 400 บริษัท กลุ่มนี้ได้คิดค้นแนวทางใหม่ในการพัฒนาเทคโนโลยีอินเตอร์เน็ตในเครือข่ายไร้สาย และก็ได้เสนอโปรโตคอล WAP ซึ่งประกอบด้วยโปรโตคอลย่อยๆ 4 ชั้น คือ ชั้น Session Layer ชั้น Transaction Layer ชั้น Security Layer ชั้น Transport Layer อีกทั้งยังคิดค้นภาษาใหม่ เรียกว่า WML (Wireless Markup Language) ซึ่งเป็นภาษาที่ใช้สร้างแอปพลิเคชันสำหรับ WAP โดยเฉพาะ คุณลักษณะของภาษานี้ส่วนหนึ่งได้มาจาก HDML เดิม และ HTML ด้วย

ทางฝั่งเอเชียในปี 1999 ในประเทศญี่ปุ่นมีบริษัท ผู้ให้บริการโทรศัพท์เคลื่อนที่ยักษ์ใหญ่ ชื่อว่า NNT DoCoMo ได้ให้บริการที่เรียกว่า I-Mode ซึ่งเป็นการใช้งานผ่านโทรศัพท์มือถือที่ใช้ระบบเครือข่ายของตนเอง บริการนี้แพร่หลายอย่างมากในประเทศญี่ปุ่นและบางส่วนในฮ่องกงแต่อย่างไรก็ดี ระบบนี้เป็นระบบปิด เพราะจำกัดอยู่แค่โทรศัพท์มือถือที่ใช้เครือข่ายของ NNT DoCoMo เท่านั้น

โปรโตคอล WAP แนวคิดของโปรโตคอลนี้ คือ การทำงานที่เหมาะสมกับเครือข่ายไร้สาย ซึ่งมีข้อจำกัดมากมายเมื่อเทียบกับเครือข่ายอินเตอร์เน็ต ไม่ว่าจะเป็นอัตราเร็วในการส่งข้อมูลที่ช้ากว่า แบนด์วิทธ์ (bandwidth) หรือช่วงความถี่ในการสื่อสารซึ่งแคบ หรือการเชื่อมต่อ(connect) ที่มีประสิทธิภาพน้อยกว่า ดังนั้นการออกแบบโปรโตคอล WAP จึงต้องพยายามให้เกิดความเหมาะสมของข้อจำกัดแบบนี้ ข้อมูลที่ส่งไปในเครือข่ายชนิดนี้เป็นข้อมูลแบบไบนารี (binary) เพื่อลดขนาดข้อมูลให้เหลือน้อยที่สุด และที่สำคัญต้องทำให้ WAP เป็นระบบเปิด ซึ่งหมายความว่าไม่ได้จำกัดอยู่ที่เครือข่าย ระบบใดระบบหนึ่ง ตัวอย่างของเครือข่ายไร้สายที่รองรับระบบ WAP คือ GSM, CDMA, TDMA, SMS, CDPD เป็นต้น

สำหรับข้อมูลอื่นๆของ WAP Forum หรือข่าวคราวความเคลื่อนไหวเกี่ยวกับ WAP สามารถติดตามได้จากเว็บไซต์ของ WAP Forum ที่ http://www.wapforum.org/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแวปมีโปรโตคอลของตัวเอง และมีภาษา WML ที่เปรียบเหมือนภาษา HTML ในโลกไร้สาย แต่กว่าที่แวปจะพัฒนาถึงตอนนี้จึงต้องย้อนดูถึงความเป็นมาเสียก่อน โดยจะประกอบด้วย

1. โปรโตคอล

โปรโตคอล คือ ข้อกำหนดหรือกฎกติกาการสื่อสารในระบบคอมพิวเตอร์ ด้วยเหตุที่ว่าระบบคอมพิวเตอร์มีอยู่หลากหลาย แต่ละระบบก็มีระเบียบวิธีในการทำงานที่แตกต่างกัน แต่การจะให้ติดต่อสื่อสารพูดคุยกันรู้เรื่อง ก็จำเป็นต้องมีข้อกำหนดหรือกติการะหว่างกัน ดังนั้นโปรโตคอลจึงจำเป็นในการสื่อสารข้อมูล

เปรียบเทียบโปรโตคอลเหมือนการส่งไปรษณีย์ ซึ่งต้องมีข้อตกลงในการรับ-ส่ง จดหมายดังนี้

1. ผู้ส่งต้องจ่าหน้าซองโดยระบุชื่อ ที่อยู่ ของผู้รับ ผู้ส่ง รวมทั้งติดแสตมป์ตามราคาที่กำหนด
2. หากเป็นจดหมายธรรมดาต้องแยกส่งว่าส่งใน กทม. หรือต่างจังหวัด
3. หากเป็นจดหมายลงทะเบียน ต้องติดต่อเคาน์เตอร์ที่ทำการไปรษณีย์ และเสียค่าธรรมเนียม
4. บุรุษไปรษณีย์ส่งจดหมายไปถึงผู้รับ ผ่านอำเภอและจังหวัดต่างๆ
5. หากเป็นจดหมายธรรมดาก็จะเสียบไว้หน้าประตูเลย
6. ถ้าเป็นจดหมายลงทะเบียน ต้องมีผู้เซ็นรับ

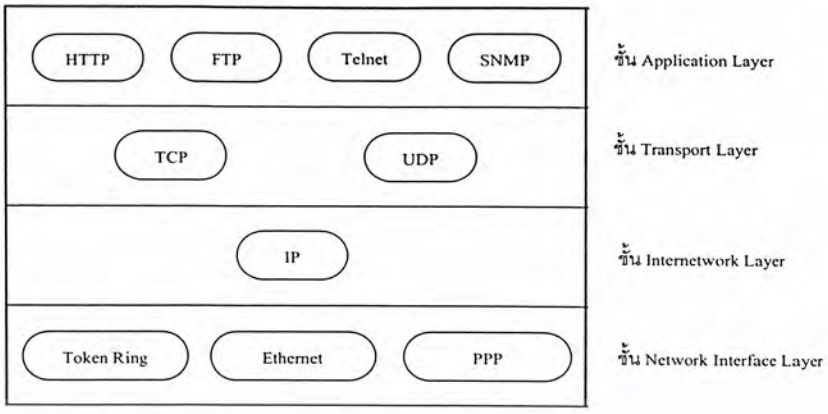
ข้อกำหนดเหล่านี้เทียบได้กับระบบการสื่อสารข้อมูลคอมพิวเตอร์ คือ ถ้าปฏิบัติตามก็จะถึงผู้รับโดย โดยในโปรโตคอลจริงๆในการรับ-ส่งข้อมูล จะต้องมี ที่อยู่ปลายทางของเครื่องที่รับข้อมูล, ต้องแบ่งข้อมูลออกเป็นส่วนย่อยๆตามขนาดที่กำหนด, ต้องมีการตรวจสอบว่าข้อมูลไปถึงเครื่องรับโดยปลอดภัยหรือไม่, ต้องระบุชนิดข้อมูลที่ส่ง เพื่อให้ทราบว่าเป็นข้อความหรือรูป เป็นต้น

2. TCP/IP

ในอดีตก่อนที่จะมีการใช้ระบบอินเตอร์เน็ตอย่างแพร่หลาย กระทรวงกลาโหมของสหรัฐอเมริกาได้พัฒนาระบบเครือข่ายที่เรียกว่า ARPANET เพื่อเชื่อมต่อกันระหว่างองค์กรภายในประเทศ ต่อมาระบบนี้ได้พัฒนากฎการสื่อสารแบบ TCP/IP ขึ้นมา และแพร่หลายไปทั่วโลกจนถึงปัจจุบัน

กฎการสื่อสารแบบ TCP/IP ได้กำหนดขั้นตอนในการสื่อสารระหว่างคอมพิวเตอร์ ออกเป็น 4 ขั้นตอนหลักๆ โดยแบ่งแต่ละชั้นๆ และแต่ละชั้นก็มีโปรโตคอลที่เกี่ยวข้องในขั้นตอนการสื่อสารนั้นๆอยู่ด้วย ดังรูปที่ 2.1 ซึ่งเรียกว่า “แบบจำลองชั้นสื่อสาร TCP/IP”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แบบจำลองชั้นสื่อสาร TCP/IP (TCP/IP Protocol Stack)

ในการรับส่งข้อมูลทางระบบอินเตอร์เน็ตนั้น โปรโตคอลในแต่ละชั้นจะทำหน้าที่แตกต่างกันไป เพื่อให้การรับส่งข้อมูล เป็นไปอย่างสมบูรณ์ โดยทั่วไปชั้นสื่อสารที่อยู่ชั้นล่างจะให้บริการชั้นที่อยู่เหนือขึ้นไป โดยโปรโตคอล TCP จะไม่สามารถทำงานได้โดยลำพัง ต้องอาศัยการบริการจากโปรโตคอล IP จึงมักเรียกรวมกันว่า TCP/IP และ TCP/IP ประกอบด้วยหลายชั้น คือ ชั้น Application Layer, ชั้น Transport Layer, ชั้น Internetwork Layer, ชั้น Network Interface Layer ซึ่งมีรายละเอียดดังนี้

ชั้น Application Layer

หรือเรียกอีกอย่างหนึ่งว่า Process Layer ในชั้นนี้มีโปรโตคอลที่เกี่ยวข้องกับการใช้งานต่างๆซึ่งเราพบเห็นกันบ่อย เช่น

โปรโตคอล HTTP เป็นข้อกำหนดเกี่ยวกับการรับ-ส่งข้อมูลหรือไฟล์เพื่อเรียกดูข้อมูลในระบบ WWW โดยเฉพาะอย่างยิ่งระหว่างเว็บเบราเซอร์ กับเว็บเซิร์ฟเวอร์

โปรโตคอล FTP เป็นข้อกำหนดเกี่ยวกับการถ่ายโอนข้อมูลจากเว็บเซิร์ฟเวอร์ มายังเครื่องลูกข่าย (client)

โปรโตคอล Telnet เป็นข้อกำหนดเกี่ยวกับการเข้าใช้งานเซิร์ฟเวอร์ผ่านทางคอมพิวเตอร์ที่เป็นไคลเอนต์ระยะไกล โดยอาศัยโปรแกรม Telnet

โปรโตคอลต่างๆในชั้นนี้ จะให้บริการแก่โปรแกรมใช้งานหรือแอปพลิชันต่างๆเช่น HTTP ให้บริการแก่เว็บเซิร์ฟเวอร์, FTP ให้บริการแก่โปรแกรมถ่ายโอนไฟล์ เป็นต้น

ยกตัวอย่างเช่น การเรียกดูเว็บไซต์ ซึ่งเป็นการทำงานของโปรแกรมเบราเซอร์ภายใต้ข้อกำหนดของโปรโตคอล HTTP ก็อธิบายได้ว่าเว็บเบราเซอร์จะขอให้โปรโตคอล HTTP กำหนดการติดต่อสื่อสารกับทางฝั่งเว็บเซิร์ฟเวอร์ ส่วนทางเว็บเซิร์ฟเวอร์นั้น เมื่อจะส่งเอกสาร HTML มายังเว็บเบราเซอร์โดยอาศัยโปรโตคอล HTTP ด้วย ดังนั้นจึงสังเกตได้ว่า เวลาระบุ URL ที่เว็บเบราเซอร์ ก็ต้องระบุในลักษณะ http://www.mydomain.com คือ มี http นำหน้า แสดงให้เห็นว่า การติดต่อระหว่างเว็บเบราเซอร์และเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ เพื่อเรียกขอดูเอกสารจากเว็บไซต์ ยึดถือ http เป็นหลักนั่นเอง แต่ทั้งนี้ทั้งนั้นก็ต้องอาศัยความสามารถของ TCP เพื่อควบคุมการติดต่อสื่อสารกันด้วย

ชั้น Transport Layer

ในชั้นนี้จะมีโปรโตคอลที่เกี่ยวข้องด้วย 2 ตัว คือ TCP และ UDP ซึ่งทำหน้าที่ควบคุมการรับ-ส่งข้อมูลระหว่างฝ่ายรับกับฝ่ายส่ง เช่น ควบคุมความเร็วในการรับ-ส่งข้อมูล เพื่อไม่ให้เกิดกรณีที่ฝ่ายส่ง ส่งข้อมูลเร็วจนฝ่ายรับรับไม่ทัน เป็นต้น โดยมีรายละเอียดดังนี้

1. TCP (Transmission Control Protocol)

โปรโตคอลนี้จะควบคุมการส่งข้อมูลระหว่างฝ่ายส่งกับฝ่ายรับ โดยการติดต่อระหว่างทั้ง 2 ฝั่ง จำเป็นต้องสร้างการเชื่อมต่อก่อน และต้องให้ฝ่ายรับพร้อมที่จะรับข้อมูลก่อนเท่านั้น ถึงจะเริ่มรับ-ส่งข้อมูลกันได้ อีกทั้งสถานการณ์เชื่อมต่อนี้จะยังอยู่จนฝ่ายใดฝ่ายหนึ่งจะยกเลิกการติดต่อ เราเรียกแบบนี้ว่า Connection-Oriented

โปรโตคอล TCP เมื่อมีการสร้างเส้นทางติดต่อแล้ว โปรโตคอลนี้จะแบ่งข้อมูล ออกเป็นหน่วยย่อยๆ เรียกว่า แพคเก็ต (packet) แล้วส่งไปให้แก่โปรโตคอลในชั้น Internetwork Layer เพื่อจัดส่งออกไป ส่วนในกรณีที่เป็นฝ่ายรับ โปรโตคอล TCP ก็จะรับข้อมูลจากชั้น Internetwork Layer ในรูปของแพคเก็ต จากนั้นจึงประกอบกลับคืนเป็นข้อมูล และนำส่งขึ้นไปให้แก่โปรโตคอลในชั้น Application Layer ของฝ่ายรับเพื่อจัดการต่อไป

สำหรับเหตุผลที่ต้องแบ่งข้อมูลออกเป็นหน่วยย่อยๆ ก็เนื่องจากว่าข้อมูลที่มีขนาดเล็กย่อมมีความคล่องตัวในการส่งผ่านเครือข่ายมากกว่า และในกรณีที่ข้อมูลเกิดความเสียหาย ฝ่ายส่งจะต้องส่งแพคเก็ตข้อมูลที่เสียหายไปยังฝ่ายรับใหม่ ก็สามารถทำได้ง่ายกว่าข้อมูลที่มีขนาดใหญ่

ในแง่ของความเชื่อถือแล้ว โปรโตคอล TCP จะมีการตรวจสอบข้อมูลที่ส่งไปว่า ถึงมือผู้รับถูกต้องหรือไม่ ถ้าไม่ถูกต้อง ก็จะส่งข้อมูลไปให้ผู้ใช้ใหม่ ซึ่งจัดว่าเป็นข้อดี แต่ขณะเดียวกันก็เป็นข้อเสียด้วย เพราะจะทำให้เสียเวลาและทรัพยากรของระบบ

2. UDP (User Datagram Protocol)

โปรโตคอลนี้มีหน้าที่ควบคุมความต่อเนื่องของการติดต่อสื่อสารข้อมูลเหมือนกับกรณี TCP แต่วิธีการติดต่อไม่เหมือนกันโดยสิ้นเชิง เพราะวิธีใช้เรียกว่า Connectionless คือ จะไม่มีการสร้างการเชื่อมต่อ ระหว่างผู้รับและผู้ส่งก่อน ฝ่ายรับไม่จำเป็นต้องรู้ว่าจะมีข้อมูลส่งมา และไม่มีการตรวจสอบว่าแพคเก็ตข้อมูล สูญหายหรือเสียหาย ซึ่งในแง่ความเชื่อถือแล้ววิธีนี้ย่อมมีความเชื่อถือน้อยกว่าแบบ Connection-Oriented แต่มีข้อดีคือไม่เปลืองทรัพยากรของระบบที่ต้องมาคอยตรวจสอบและส่งข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากโปรโตคอล UDP ไม่มีการตรวจสอบแพคเก็ตข้อมูลที่สูญหายและส่งข้อมูลได้รวดเร็ว โปรโตคอลชนิดนี้จึงเหมาะกับการส่งข้อมูลจำพวกมัลติมีเดีย (multimedia) แบบ streaming เช่น เพลง, ภาพแอนิเมชัน เป็นต้น เพราะเมื่อแพคเก็ตข้อมูลสูญหายไป ก็จะมีผลกระทบต่อคุณภาพเสียงและแอนิเมชันเพียงชั่วขณะ ซึ่งไม่ถือ ว่าสำคัญเทียบกับข้อมูลอื่นๆ

การที่โปรโตคอลในชั้น Transport Layer ได้ออกแบบมาให้มีการติดต่อ 2 แบบ คือ Connection-Oriented ใน TCP และ Connectionless ใน UDP มีข้อดีในแง่ที่ว่าสามารถเลือกรูปแบบการติดต่อให้เหมาะสมกับแต่ละงาน อย่างเช่น กรณีโปรโตคอล HTTP หรือ FTP ซึ่งต้องอาศัยความถูกต้องและแน่นอนจะใช้โปรโตคอล TCP

ไม่ว่าการติดต่อนั้นจะอาศัยโปรโตคอล TCP หรือ UDP ในการควบคุมการรับ-ส่งของข้อมูล สิ่งหนึ่งที่ขาดไม่ได้คือ การพึ่งพาความสามารถของโปรโตคอลในชั้น Internetwork Layer เพื่อหาเส้นทางเหมาะสมที่สุดในการส่งข้อมูล

ชั้น Internetwork Layer

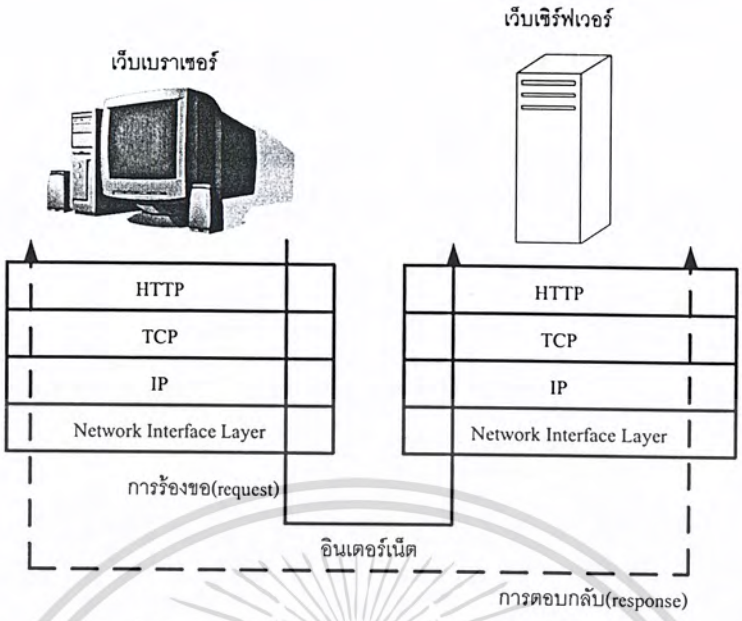
เป็นชั้นสื่อสารที่มีโปรโตคอลหลักคือ IP (Internet Protocol) ซึ่งทำหน้าที่ในการหาเส้นทางที่เหมาะสมที่สุด สำหรับการส่งข้อมูลแต่ละหน่วยย่อยๆ ลักษณะงานตามโปรโตคอล IP ก็จะเป็นแบบ Packet-Switching คือ แต่ละแพคเก็ตจะเดินทางจากจุดเริ่มต้นไปยังจุดหมายปลายทาง โดยไม่จำเป็นต้องไปทางเดียวกันเสมอ และลำดับของแพคเก็ตที่ไปถึงปลายทางก็ไม่สำคัญ แพคเก็ตแรกที่เราส่งไปอาจจะถึงปลายทางทีหลังแพคเก็ตอื่นๆก็ได้ เส้นทางที่แต่ละแพคเก็ตผ่านไปก็ไม่แน่นอน กล่าวคือ ถ้าส่งข้อมูลชุดเดิมอีกครั้ง แพคเก็ตเดิมอาจได้อีกเส้นทางหนึ่ง ซึ่งเป็นคนละทางกับการส่งครั้งแรก จึงสรุปได้ว่าโปรโตคอล IP ทำงานแบบ Connectionless เพราะไม่มีการระบุเส้นทางตายตัวจากผู้ส่งไปยังผู้รับ

ชั้น Network Interface Layer

เป็นชั้นที่รับข้อมูลมาจากชั้น Internetwork Layer แล้วปรับแพคเก็ตให้อยู่ในรูปที่เหมาะสม กับการส่งไปยังเครือข่ายหรือสายสื่อสาร เช่น ปรับสัญญาณทางไฟฟ้าเพื่อส่งไปในสายสื่อสาร เป็นต้น และในทางกลับกันหากเป็นฝ่ายรับ ก็จะรับข้อมูลมาจากสายสื่อสาร แล้วปรับรูปแบบของข้อมูลเพื่อส่งขึ้นไปยังชั้น Internetwork Layer ต่อไป

สรุปการทำงานระหว่างเว็บเซิร์ฟเวอร์และเว็บเบราเซอร์ในระบบ WWW ซึ่งอาจเกี่ยวพันกับโปรโตคอลต่างๆ ดังรูปที่2.2 โดยที่การร้องขอ (request) จากไคลเอนท์ไปยังเซิร์ฟเวอร์ จะแทนด้วยเส้นทึบ ส่วนขั้นตอนของการตอบรับ (response) จะแทนด้วยเส้นประ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โปรโตคอลที่เกี่ยวของกับการร้องขอและตอบกลับระหว่างเว็บเบราเซอร์กับเว็บเซิร์ฟเวอร์

ข้อจำกัดของโทรศัพท์มือถือและเครือข่ายไร้สาย

เทคโนโลยีอินเตอร์เน็ตพัฒนาขึ้นตามความสามารถในการประมวลผลด้วยคอมพิวเตอร์ แต่เมื่อจะเปลี่ยนรูปแบบการใช้งานเป็นโทรศัพท์มือถือ ย่อมมีข้อจำกัดกว่าเครื่องคอมพิวเตอร์มากมาย คือ

1. ซีพียู(CPU)มีกำลังการประมวลผลน้อยกว่า
2. มีหน่วยความจำน้อยกว่า
3. มีข้อจำกัดด้านแหล่งจ่ายพลังงานซึ่งเป็นแบตเตอรี่อย่างมาก เมื่อเปรียบเทียบกับคอมพิวเตอร์ซึ่งใช้กระแสไฟฟ้า ซึ่งสามารถใช้งานได้ตลอดเวลาเมื่อเสียบปลั๊ก
4. หน้าจอแสดงผลมีพื้นที่เล็กกว่า
5. การใช้งานด้วยปุ่มกดบนโทรศัพท์ ทำให้ไม่สะดวกเท่าการใช้เมาส์หรือคีย์บอร์ด

นอกจากนี้ในส่วนของเครือข่ายไร้สาย ก็มีข้อจำกัดเมื่อเทียบกับเครือข่ายอินเตอร์เน็ตดังนี้

1. มีแบนด์วิทธ์แคบกว่า
2. ระยะเวลาในการส่งข้อมูลผ่านเครือข่าย(latency)มากกว่า
3. สภาพเชื่อมต่อมีความเสถียรภาพ(stability) ต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

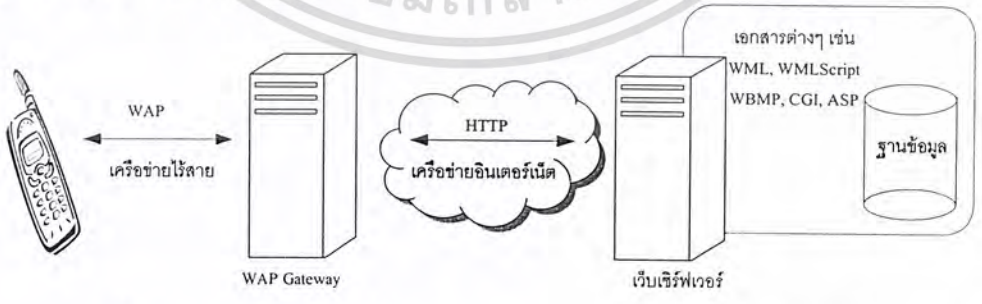
2.1.2 TCP/IP กับเครือข่ายไร้สาย

โปรโตคอล TCP/IP เป็นแกนหลักของระบบอินเตอร์เน็ตนั้น ความจริงได้มีผู้นำไปทดลองใช้ภายในเครือข่ายไร้สายหรือเครือข่ายโทรศัพท์เคลื่อนที่มาแล้ว แต่เรื่องข้อจำกัดของสภาพความไม่มีเสถียรภาพของการเชื่อมต่อจึงส่งผลให้คลื่นสัญญาณขาดหายไปบ้างตามสภาวะแวดล้อม ซึ่งมีผลต่อประสิทธิภาพโดยรวมของโปรโตคอล TCP สาเหตุเนื่องมาจากเมื่อมีการขาดหายของสัญญาณ จะทำให้การสื่อสารข้อมูลไม่มีความต่อเนื่อง โปรโตคอล TCP ก็จะตีความไปว่า ขณะนั้นมีการส่งข้อมูลอย่างแออัดคับคั่งในเครือข่าย จึงพยายามลดอัตราส่งข้อมูลลง และเมื่อคลื่นสัญญาณกลับมาดีเหมือนเดิม โปรโตคอล TCP ก็จะเพิ่มอัตราการส่งข้อมูลให้ใกล้เคียงกับอัตราเดิม แต่วิธีนี้จะกินเวลานาน ซึ่งเมื่อเปรียบเทียบกับเครือข่ายอินเตอร์เน็ตแบบปกติ การส่งข้อมูลจะต่อเนื่อง อัตราการข้อมูลจะคงที่ ประสิทธิภาพของโปรโตคอล TCP ก็จะดีกว่า

นอกจากข้อจำกัดทางด้านความเสถียรแล้ว ยังมีข้อจำกัดทางด้านแบนด์วิทธ์ในเครือข่ายไร้สาย ซึ่งมีช่วงแคบ และเวลาที่ใช้ในการส่งข้อมูลของเครือข่ายไร้สาย ซึ่งนับว่ามากเกินไป ส่งผลให้ความเร็วในการรับส่งข้อมูลมีอยู่จำกัดเพียง 9.6 kbps เท่านั้น และด้วยข้อจำกัดทั้งหลายเหล่านี้นั่นเอง ล้วนเป็นเหตุให้การพัฒนาใช้งานอินเตอร์เน็ตในเครือข่ายไร้สาย ด้วยโปรโตคอล TCP/IP ไม่ประสบความสำเร็จอย่างสิ้นเชิง

2.1.3 หลักการทำงานของแวป

จากข้อจำกัดที่ได้กล่าวมาแล้ว จึงเกิดแนวคิดที่ว่า เมื่อข้อมูลในอินเตอร์เน็ตไม่สามารถรับส่งได้ดีในเครือข่ายไร้สาย อันเนื่องมาจากโปรโตคอล TCP ทำงานได้ไม่ดีในเครือข่ายไร้สาย นอกจากนี้อุปกรณ์โทรศัพท์มือถือยังมีความสามารถไม่พอที่จะประมวลผลข้อมูลในระบบอินเตอร์เน็ต ทางออกคือ เมื่อต้องการส่งข้อมูลจากเว็บเซิร์ฟเวอร์ไปยังโทรศัพท์มือถือ ข้อมูลนั้นจะยังคงถูกส่งผ่านเครือข่ายอินเตอร์เน็ต ที่ใช้โปรโตคอล TCP/IP เป็นโปรโตคอลหลัก มาให้แก่ตัวกลาง ซึ่งทำหน้าที่แปลงข้อมูลไปเป็นอีกรูปแบบหนึ่งก่อนที่จะส่งไปยังโทรศัพท์มือถือ เพื่อให้ข้อมูลนั้นอยู่ในลักษณะที่เหมาะสมกับการส่งผ่านเครือข่ายไร้สาย และเหมาะสมกับความสามารถในการประมวลผลของโทรศัพท์มือถือ และตัวกลางที่ใช้แปลงดังกล่าวเรียกว่า แวปเกตเวย์โดยมีการทำงานดังรูปที่ 2.3 ซึ่งสรุปได้ว่า การรับส่งข้อมูลระหว่างแวปเกตเวย์กับเว็บเซิร์ฟเวอร์ เป็นตามมาตรฐานของโปรโตคอลต่างๆของ WWW ตามปกติ



รูปที่ 2.3 แวปเกตเวย์ทำหน้าที่เป็นตัวกลางระหว่างเครือข่ายอินเตอร์เน็ตและเครือข่ายไร้สาย

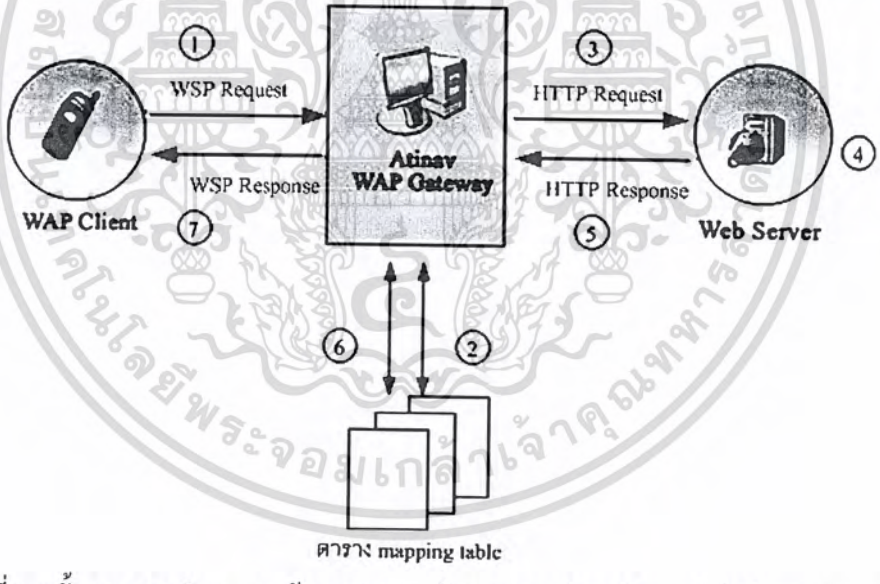
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนทางฝั่งแวปเกตเวย์กับโทรศัพท์มือถือ จะเป็นไปตามมาตรฐานชุดโปรโตคอลที่ WAP Forum พัฒนาขึ้น โดยข้อมูลที่รับ-ส่งในฝั่งนี้จะเป็นข้อมูลแบบไบนารี่ ที่ผ่านการบีบอัดให้เล็กลงแล้ว ซึ่งเป็นหน้าที่ของแวปเกตเวย์ในการแปลงให้เหมาะสม ขั้นตอนการทำงานขอแวปเกตเวย์นั้นมีความคล้ายคลึงกับ proxy server ในระบบอินเตอร์เน็ต ดังนั้นควรทำความเข้าใจเกี่ยวกับ proxy server เสียก่อน

2.1.4 การพิจารณา proxy server

อุปกรณ์ที่เราเรียกว่า proxy server เป็นอุปกรณ์ที่มีประโยชน์มากในการลดปริมาณจราจรในเครือข่าย เนื่องจากหน้าที่ของ proxy server คือ เก็บเอกสารหรือเว็บเพจไว้ในพื้นที่เก็บข้อมูลหรือเอกสารสำรองของตน ที่เรียกว่า แคช(cache) เมื่อมีผู้ใช้เรียกดูเอกสารใน WWW โดยการระบุ URL ที่เว็บเบราเซอร์ โปรแกรมเบราเซอร์นั้นจะส่งสัญญาณร้องขอมาที่ proxy server ก่อน แล้ว proxy server ก็จะตรวจสอบว่ามีไฟล์หรือเอกสารหรือเว็บเพจนั้นอยู่ในแคชหรือไม่ หากพบว่ามีอยู่และเป็นข้อมูลล่าสุด ก็จะส่งข้อมูลนั้นไปให้ผู้ใช้ โดยไม่ต้องร้องขอเอกสารจากเว็บเซิร์ฟเวอร์อีกที

2.1.5 ความแตกต่างระหว่างแวปเกตเวย์กับ proxy server



รูปที่ 2.4 ขั้นตอนการทำงานและร้องขอเอกสารในระบบแวปและอินเตอร์เน็ตผ่านแวปเกตเวย์

ความสามารถอย่างหนึ่งของแวปเกตเวย์ก็คือ การทำงานเป็น proxy server และแคช ได้เช่นกัน แต่หน้าที่หลักจริงๆของแวปเกตเวย์คือ การแปลงรูปแบบการสื่อสารระหว่างฝั่งเครือข่ายอินเตอร์เน็ต กับฝั่งเครือข่ายไร้สาย หรือที่เรียกว่า protocol conversion จากรูปที่ 2.4 จะแสดงให้เห็นการทำงานตามขั้นตอนดังนี้

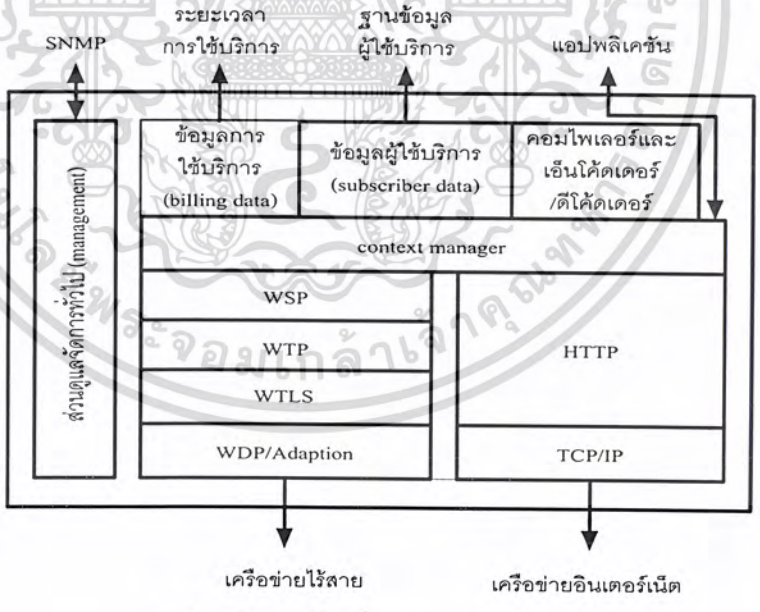
1. ผู้ใช้งานโทรศัพท์มือถือ หรือไคลเอนต์ ส่ง URL ของเอกสารที่ต้องการไปยังแวปเกตเวย์โดยส่งเป็นคำร้องในรูปแบบโปรโตคอล WSP (WSP Request)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แวปเกตเวย์ถอดรหัส (decode) คำร้องขอที่อยู่ในรูปแบบไบนารี่ WSP (WSP Request) เพื่อแปลงให้อยู่ในคำร้องขอที่อยู่ในรูปแบบ HTTP (HTTP Request) โดยอาศัยตาราง mapping table ที่มีอยู่ในแวปเกตเวย์เป็นตัวช่วย
3. แวปเกตเวย์สร้างการเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์ แล้วส่งคำร้องขอตามไปในรูปแบบโปรโตคอล HTTP (HTTP Request)
4. เว็บเซิร์ฟเวอร์จะทำการประมวลผลคำร้องขอนั้น แล้วตรวจสอบดูว่า เอกสารตามที่ขอร้องเป็นลักษณะซอร์ดโค้ด WML ธรรมดาหรือไม่ หากเอกสารนั้นเรียกการทำงานของสคริปต์ต่างๆ เช่น ASP, CGI ก็ต้องประมวลผลสคริปต์นั้นก่อน เพื่อจะให้กลายเป็นเอกสารแบบ WML ธรรมดาซึ่งประกอบด้วยแท็กและข้อความ
5. เว็บเซิร์ฟเวอร์ส่งเอกสารกลับคืนมายังแวปเกตเวย์โดยส่งเป็นคำตอบในรูปแบบโปรโตคอล HTTP (HTTP Response)
6. แวปเกตเวย์ก็จะเข้ารหัสเอกสาร (encode) ไปเป็นรูปแบบไบนารี โดยอาศัยตาราง mapping table
7. แวปเกตเวย์สร้างการติดต่อ ไปยังไคลเอนต์แล้วส่งข้อมูลไบนารี่นั้น เป็นคำตอบกลับในรูปแบบโปรโตคอล WSP (WSP Response) ไปยังไคลเอนต์ต่อไป

2.1.6 โครงสร้างและฟังก์ชันพื้นฐานของแวปเกตเวย์

โครงสร้างหรือสถาปัตยกรรมของแวปเกตเวย์ซึ่งประกอบด้วยฟังก์ชันพื้นฐานเป็นดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้างพื้นฐานของ แวปเกตเวย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะเห็นได้ว่าในแวปเกตเวย์ มีทั้งชุดโปรโตคอล WAP (WSP, WTP, WTLS, WDP) และชุดโปรโตคอล TCP/IP ทั้งนี้เนื่องจากแวปเกตเวย์เป็นตัวกลางระหว่างเครือข่ายอินเตอร์เน็ตกับเครือข่ายไร้สาย จึงต้องมีคุณสมบัติรองรับโปรโตคอลที่ใช้ในแต่ละฝั่ง เมื่อติดต่อกับเว็บเซิร์ฟเวอร์ก็ต้องอาศัยชุดโปรโตคอล TCP/IP ส่วนการติดต่อกับไคลเอนท์หรือโทรศัพท์เคลื่อนที่ใช้ระบบแวปและใช้โปรโตคอลแวป นั่นเอง

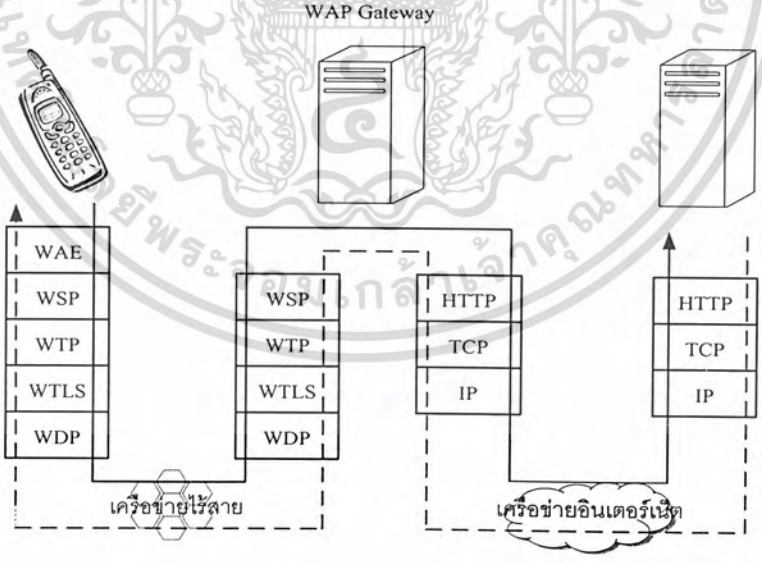
2.1.7 หน้าที่หลักของแวปเกตเวย์

โดยทั่วไปแวปเกตเวย์ มีหน้าที่ดังนี้

1. รองรับโปรโตคอลแวปและชุดโปรโตคอลในอินเตอร์เน็ต
2. protocol conversion
3. เข้ารหัสเอกสาร WML ให้เป็นข้อมูลรูปแบบไบนารี
4. คอมไพล์โค้ด WMLScript
5. เป็น proxy server เพื่อให้บริการข้อมูลที่ถูกเรียกบ่อยๆ
6. ดูแลด้านการรักษาความปลอดภัยของข้อมูล
7. เปลี่ยนเอกสาร HTML ที่ได้รับจากเว็บเซิร์ฟเวอร์ ให้เป็นเอกสาร WML

2.1.8 การเชื่อมต่อโปรโตคอลใน WWW และ แวป เข้าด้วยกัน

จากที่กล่าวมาทั้งหมดแล้ว เราสามารถนำมาเชื่อมต่อกันเป็นภาพรวมในแง่โปรโตคอลได้ดังรูปที่ 2.6



รูปที่ 2.6 ภาพรวมของโปรโตคอลที่เกี่ยวข้องในเครือข่ายไร้สายและเครือข่ายอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าแวปเกตเวย์เป็นอุปกรณ์ที่ต้องรองรับโปรโตคอล 2ชุด แต่จุดที่น่าสังเกตอย่างยิ่ง คือ ทำไมที่แวปเกตเวย์จึงไม่มีชั้น WAE เหตุผลคือ WAE ไม่ถือเป็นชั้นสื่อสาร แต่เป็นมาตรฐานที่ใช้ในการพัฒนา WAP Application เช่น ภาษา WML, WMLScript ดังนั้นจึงมีใช้งานแต่เฉพาะทางฝั่งโทรศัพท์มือถือ ส่วนในแวปเกตเวย์มีเฉพาะโปรโตคอลที่ใช้ในการสื่อสารเท่านั้น

2.2 Java Programming

จาวา คือ ภาษาสำหรับเขียนโปรแกรมหนึ่งซึ่งสนับสนุนโปรแกรมเชิงวัตถุ โดยโปรแกรมภาษาจาวาถูกจัดว่ามีโครงสร้างของภาษาที่แสดงความเป็น Object Orientation อย่างชัดเจน เนื่องจากภาษาจาวา สามารถทำงานบนเครื่องคอมพิวเตอร์ที่มีระบบปฏิบัติการต่างกันได้ โดยไม่ต้องมีการคอมไพล์ใหม่ เป็นภาษาที่มีไวยกรณ์ที่สามารถเรียนรู้ได้ง่าย มีกลไกลของภาษาไม่ซับซ้อน เน้นความถูกต้องของชนิดข้อมูลที่ใช้ในโปรแกรม และจาวา ยังถูกออกแบบมาให้เป็นภาษาเชิงวัตถุอย่างแท้จริงอีกด้วย ดังนั้นโปรแกรมภาษาจาวา จึงได้รับความนิยมสูงสุดในปัจจุบัน

โปรแกรมภาษาจาวา เป็นโปรแกรมภาษาที่มีรูปแบบการทำงานที่หลากหลาย โดยเริ่มแรกการทำงานของจาวา มีเพียง 2 รูปแบบเท่านั้น คือ Applet และ Application

Applet เป็นรูปแบบโปรแกรมภาษาจาวา ที่ถูกนำไปใช้ในการทำงานในเว็บบราวเซอร์ หรือเว็บเพจทั่วไป

Application ต่างกับ Applet คือไม่ได้ถูกใช้ในการทำงานบนเว็บบราวเซอร์ แต่สามารถทำงานได้ด้วยตนเอง เหมือนกับโปรแกรมที่พัฒนาโดยภาษาระดับสูงทั่วไป เช่น C++

เนื่องจากในปัจจุบัน มีความต้องการใช้งานของซอฟต์แวร์เพื่ออำนวยความสะดวกในการทำงานเพิ่มมากขึ้น ทำให้จาวาต้องพัฒนารูปแบบของโปรแกรมภาษาเพื่อให้สามารถรองรับกับการทำงานอื่น ๆ ที่เพิ่มเติมขึ้นมา เช่น

Servlets เป็นรูปแบบโปรแกรมภาษาจาวา ที่นำมาใช้ทำงานบนเว็บเซิร์ฟเวอร์ซึ่งเป็นเครื่องแม่ข่ายของแต่ละเว็บไซต์

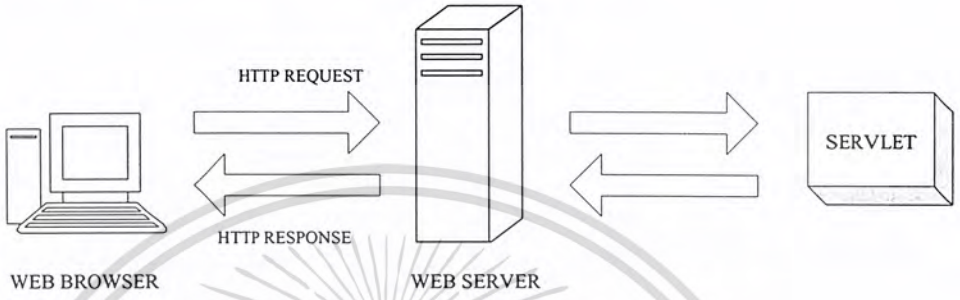
Midlets เป็นรูปแบบโปรแกรมภาษาจาวาที่ถูกนำมาใช้ในการทำงานของอุปกรณ์ประเภทไร้สายทั่ว ๆ ไป

2.2.1 จาวาเซิร์ฟเลต

จาวาเซิร์ฟเลต คือ โปรแกรมภาษาจาวาที่ถูกสร้างมาให้ทำงานบน HTTP Server โดยการทำงานของเซิร์ฟเลต (Sevlet) มีลักษณะเป็น Server-side ซึ่งต่างจาก Java Applet โดยทั่วไปที่ทำงานในลักษณะ Client-side โดย เซิร์ฟเลต เป็นโมดุล (modules) ที่ทำงานอยู่ภายในกระบวนการ request/response oriented server และเซิร์ฟเลต ยังสามารถตอบสนองในการส่งข้อมูลโดยให้อยู่ในรูปภาษา HTML โดย เซิร์ฟเลต ยังสามารถนำมาประยุกต์ใช้ในการพัฒนาฐานข้อมูล (database) ต่างๆได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ของเซิร์ฟเลตนั้นช่วยให้เราเขียนโปรแกรมที่สั่งให้ HTTP Server ทำงานหรือโต้ตอบกับโปรแกรมของไคลเอนท์เหมือนที่ CGI (Common Gateway Interface) ทำได้ซึ่งเซิร์ฟเลตถูกนำมาใช้แทน CGI scripts ได้อย่างมีประสิทธิภาพและให้ความสะดวกได้อย่างมาก คือ สามารถเขียนได้ง่ายกว่าและทำการรันได้เร็วกว่า CGI และเซิร์ฟเลตยังได้มีการจัดการในส่วนของการโปรแกรมทางด้าน server-side ไว้โดยเฉพาะเซิร์ฟเลต ได้ถูกพัฒนาโดยใช้ Java Servlet API ซึ่งเป็นส่วนเพิ่มเติมของมาตรฐานจาวา



รูปที่ 2.7 แสดงการทำงานของเซิร์ฟเลต

2.2.2 จุดเด่นของเซิร์ฟเลต

1. เซิร์ฟเลตสามารถที่จะฝังตัวทำงานอยู่บนเซิร์ฟเวอร์ เมื่อทำการให้บริการไคลเอนท์ เสร็จสิ้นแล้วจะคอยรับการร้องขอใหม่ที่อาจจะมีมาจากไคลเอนท์อีก โดยไม่ต้องถูกสร้างขึ้นมาใหม่จึงทำให้เซิร์ฟเลตใช้ทรัพยากรของเซิร์ฟเวอร์ น้อยมาก
2. เซิร์ฟเลตสามารถจัดการกับงานประเภท Multiple Connection ได้ดี โดยอาศัยความสามารถของ Thread ที่มีอยู่ในตัวจาวา
3. เซิร์ฟเลตสามารถติดต่อกับ Applet บนเซิร์ฟเวอร์ ได้อย่างต่อเนื่องเป็นผลให้การรับส่งข้อมูลระหว่างกันดำเนินไปอย่างต่อเนื่อง
4. เซิร์ฟเลตสามารถถูก Upload จากไคลเอนท์ เพื่อสั่งให้ไปทำงานบนเซิร์ฟเวอร์ใดๆบนเน็ทเวิร์คได้ประโยชน์ คือ เราจะสามารถเขียนโปรแกรม เพื่อค้นหาข้อมูลใน Web Host ใดๆ โดยไม่ต้องส่งข้อมูลมาประมวลที่เครื่องต้นทาง สามารถทำได้ที่ตัว Host นั้นๆเลย เสร็จแล้วค่อยส่งผลรับกลับคืน จึงสามารถลดเวลาและทราฟฟิคบนระบบเน็ทเวิร์ค ลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.3 สถาปัตยกรรมของเซิร์ฟเลต

ใจความสำคัญของ Servlet API คือ การสนับสนุนการอินเตอร์เฟส (interface) ของเซิร์ฟเลต ทั้งแบบทางตรงและทางอ้อม โดยประกอบไปด้วยคลาสที่สนับสนุนการอินเตอร์เฟส เช่น Http Servlet ซึ่ง Servlet Interface ประกอบไปด้วยเมธอด (method) ที่ใช้เป็นตัวจัดการเซิร์ฟเลต และติดต่อสื่อสารกับไคลเอนท์ ผู้เขียนเซิร์ฟเลต จะกำหนดเมธอดนี้ทั้งหมดหรือบางส่วน เมื่อต้องการพัฒนาเซิร์ฟเลต เมื่อเซิร์ฟเลตได้รับการเรียกร้องจาก ไคลเอนท์จะมีการรับออบเจ็ค (object) อยู่ 2 ออบเจ็ค ซึ่งอันแรกคือ Servlet Request และอีกอันคือ Servlet Response โดยคลาส Servlet Request นั้นจะเป็นตัวติดต่อสื่อสารจากไคลเอนท์ไปยังเซิร์ฟเวอร์ ในขณะที่คลาส Servlet Response เป็นตัวติดต่อสื่อสารจากเซิร์ฟเวอร์กลับไปยังไคลเอนท์

Servlet Request Interface อนุญาตให้ Servlet สามารถดึงข้อมูลได้เช่น ชื่อของพารามิเตอร์ที่เข้ามาโดยไคลเอนท์ โปรโตคอลและชื่อ Host ของไคลเอนท์ ที่เป็นตัวส่งการร้องขอของเซิร์ฟเวอร์ที่เป็นตัวรับ request ด้วยรวมไปถึงการกำหนด Servlet เข้าถึง input stream ได้ด้วย โดย Servlet Input Stream โดย Servlet ทำหน้าที่รับข้อมูลจากไคลเอนท์ โดยผ่านทางโปรโตคอล เช่น HTTP, POST และ PUT เมธอด โดยคลาสย่อยของคลาส Servlet Request จะอนุญาตให้เซิร์ฟเลต สามารถรับข้อมูลที่เป็นลักษณะเฉพาะของโปรโตคอลกลับมาได้

Servlet Response Interface ประกอบด้วยเมธอด (method) สำหรับการตอบกลับไปยังไคลเอนท์ โดยอนุญาตให้เซิร์ฟเลต กำหนดขนาดและ mine type ในการตอบกลับและเป็นตัวจัดเตรียม output stream ด้วย โดยผู้เขียนเซิร์ฟเลต สามารถใช้ Servlet Output Stream ในการส่งข้อมูลกลับได้คลาสย่อยของ Servlet Response ทำให้เซิร์ฟเลตมีโปรโตคอลที่แตกต่างกันได้หลายชนิด ตัวอย่างเช่น Http Servlet Response ประกอบด้วยเมธอดที่อนุญาตให้เซิร์ฟเลต สามารถถ่ายเทข้อความจาก HTTP-specific header ได้

2.3 การสื่อสารข้อมูล (Data communications)

การสื่อสารข้อมูลข่าวสารที่ถูกเข้ารหัสแล้วระหว่างอุปกรณ์ 2 ชนิด โดยมีองค์ประกอบหลักที่สำคัญ คือ ผู้ส่งสาร ผู้รับสาร และตัวกลางในการส่งผ่านข้อมูล โดยมีสิ่งที่ผู้ส่งส่งผ่านตัวกลางให้ผู้รับได้รับ คือ ข้อมูลข่าวสาร โดยมีรูปแบบของการสื่อสารระหว่างอุปกรณ์ 2 ชนิด แบ่งได้เป็น 3 ประเภท คือ

1. การสื่อสารทางเดียว (Simplex) ใช้ในการส่งผ่านข้อมูลที่อุปกรณ์แต่ละด้านจะเป็นเฉพาะตัวส่งหรือตัวรับเพียงอย่างใดอย่างหนึ่งเท่านั้น ทำให้ทิศทางของการส่งผ่านข้อมูลมีทิศทางเดียว

2. การสื่อสารกึ่งสองทาง (Half duplex) ใช้ในการติดต่อกันระหว่างอุปกรณ์ที่มีการแลกเปลี่ยนข้อมูลข่าวสารกัน อุปกรณ์แต่ละด้านจึงเป็นได้ทั้งตัวส่งและตัวรับ แต่ไม่สามารถรับและส่งพร้อมกันในเวลาเดียวกันได้ ทำให้ทิศทางของการส่งผ่านข้อมูลมีทิศทางเดียวในเวลาใด ๆ

62018
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การสื่อสารสองทิศทาง (Full duplex) ใช้ในการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ 2 ชนิดที่ติดต่อกัน โดยมีทิศทางของการส่งผ่านข้อมูลสามารถไหลได้ทั้งสองทางในเวลาเดียวกัน ดังนั้น อุปกรณ์ทั้งสองฝั่งสามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

2.3.1 การส่งผ่านข้อมูล (Transmission modes)

การส่งผ่านข้อมูลเป็นกระบวนการในการนำกลุ่มของบิตข้อมูลจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง หรือการส่งผ่านข้อมูลจากอุปกรณ์ฝั่งส่งไปยังอุปกรณ์ฝั่งรับ แบ่งได้เป็น 2 แบบ คือ การส่งผ่านข้อมูลแบบขนาน (Parallel transmission) และการส่งผ่านข้อมูลแบบอนุกรม (Serial transmission)

- การส่งผ่านข้อมูลแบบขนาน เป็นการส่งผ่านข้อมูลทีละหลายๆบิต(bit)จากอุปกรณ์ส่งไปยังอุปกรณ์รับพร้อม ๆ กัน และสายสัญญาณที่ใช้รับส่งจะต้องมีอย่างน้อยเท่ากับจำนวนบิตที่การส่งออกไปพร้อมกันนั้น ข้อดีของการส่งผ่านข้อมูลแบบขนาน คือ สามารถรับส่งข้อมูลได้รวดเร็ว และส่งข้อมูลได้ปริมาณมาก ๆ ในหนึ่งหน่วยเวลา ข้อเสียได้แก่ ต้องใช้สายในปริมาณมากทำให้สูญเสียค่าใช้จ่ายมาก, การมัดสายหลาย ๆ เส้นรวมกันในเคเบิลเดียวกันทำให้สายมีขนาดใหญ่ เทอะทะ, คุณสมบัติของสายแต่ละเส้นไม่เหมือนกัน ทำให้ข้อมูลมาถึงเครื่องรับไม่พร้อมกันเกิดความผิดพลาดของข้อมูลที่เครื่องรับ

- การส่งผ่านข้อมูลแบบอนุกรม ใช้สายสัญญาณเดียวกันในการส่งผ่านข้อมูลทั้งหมด ทำให้ค่าใช้จ่ายถูกกว่าแบบขนาน แต่ความเร็วในการส่งผ่านข้อมูลช้ากว่าแบบขนาน

การส่งผ่านข้อมูลแบบอนุกรมนั้นแบ่งได้เป็น 2 แบบ คือ การส่งผ่านข้อมูลแบบซิงโครนัส (Synchronous) และการส่งผ่านข้อมูลแบบอะซิงโครนัส (Asynchronous)

- การส่งผ่านข้อมูลแบบซิงโครนัส จะไม่ใช้บิตเริ่มต้นและบิตสิ้นสุด และจะไม่มีการหยุดชั่วขณะระหว่างตัวอักษร (Character) แต่จะให้วิธีให้จังหวะเวลาทั้งสองทางที่ติดต่อกัน มีอยู่สองวิธีที่ปฏิบัติกัน คือ ใช้ตัวอักษรซิงค์ (Sync Character) และใช้สัญญาณนาฬิกา (Clock signal) การใช้ตัวอักษรซิงค์ โดยใส่ไว้หน้าบล็อกของข้อมูลตัว อักษรซิงค์นี้เป็นบิตจำนวนหนึ่งซึ่งทางอุปกรณ์เครื่องรับสามารถใช้ในการกำหนดอัตราเร็วของข้อมูลให้ตรงกับทางอุปกรณ์เครื่องส่ง ส่วนการใช้สัญญาณนาฬิกาของด้านส่งและสัญญาณนาฬิกาของด้านรับจะใช้คนละสายหรือคนละช่องสัญญาณในการส่งข่าวสารเกี่ยวกับเวลาของข้อมูลที่จะส่ง ดังนั้น การติดต่อกันแบบนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูลและกราวนด์

- การส่งผ่านข้อมูลแบบอะซิงโครนัส คือ การรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัสแต่จะให้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน เรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่าอัตราการส่งผ่านข้อมูลหรือบิตเรต (Bit rate) มีหน่วยเป็นบิตต่อวินาที

รูปแบบของข้อมูลแบบอะซิงโครนัสประกอบด้วย 4 ส่วน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
- บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
- บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิต หรือ ไม่มี
- บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต

ในการส่งข้อมูลแบบอะซิงโครนัส เมื่อไม่มีข้อมูลจะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting stage) เมื่อมีการเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ข้อมูลมีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่ง คือ บิตปิดท้าย ซึ่งจะให้สถานะลอจิก "1" อีครั้งด้วยระยะเวลาอย่างน้อย 1, 1.5 หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว โดยอัตราความเร็วในการรับส่งข้อมูลก็คือ ค่าบิตเรตหรือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที บิตเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (Odd) หรือแบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่ โดยต้องรวมบิตพาริตี้เข้าไปด้วย ส่วนทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ เมื่อได้รับข้อมูลก็จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทางภาครับจะแสดงข้อมูลผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวหรือผิดพลาดเป็นจำนวนคี่เท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตผิดพลาดมากกว่า 1 บิต

2.3.2 มาตรฐานพอร์ทอนุกรมแบบ RS-232

RS-232 หรือ V.24 เป็นมาตรฐานการเชื่อมต่อระหว่างอุปกรณ์รับส่งปลายทาง (Data Terminal Equipment หรือ DTE) กับอุปกรณ์สื่อสารข้อมูล (Data circuit Terminating Equipment หรือ DCE) เพื่อใช้ในการส่งข้อมูลไบนารีแบบอนุกรม โดยที่RS-232 เป็นมาตรฐานที่กำหนดขึ้น จากสมาคมอิเล็กทรอนิกส์อุตสาหกรรม (Electronic Industries Association หรือ EIA) ซึ่งเกิดมาด้วยความร่วมมือของบริษัท Bell ซึ่งเป็นบริษัทผู้ผลิตคอมพิวเตอร์และโมเด็มเพื่อใช้เป็นมาตรฐานในการเชื่อมต่อ ตัวอักษรภาษาอังกฤษของคำว่า "RS" ย่อมาจาก "Recommended Standard" ตามด้วยเลข 3 ตัว และปิดท้ายด้วยตัวอักษรภาษาอังกฤษ A, B, Cหรือ D เป็นการแสดงถึงรุ่นที่ได้รับการพัฒนามาตรฐาน RS-232 ที่สร้างขึ้นโดยสมาคมอิเล็กทรอนิกส์อุตสาหกรรมจะเทียบเท่ากับมาตรฐาน V ที่สร้างขึ้นโดยองค์การสหภาพโทรคมนาคมระหว่างประเทศ (Consultative committee on International Telephone and Telegraph หรือ CCITT) มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับแรงดันสัญญาณตั้งแต่ –3 ถึง –12 โวลต์ แสดงว่าเป็นข้อมูล (Mark) และ +3 ถึง +15 โวลต์ แสดงเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์รับส่งปลายทางกับอุปกรณ์สื่อสารข้อมูลไว้ว่า อุปกรณ์รับส่งปลายทางจะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์สื่อสารข้อมูลจะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจากอุปกรณ์รับส่งปลายทางเท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์รับส่งปลายทางและอุปกรณ์สื่อสารข้อมูล อย่างหนึ่งที่เห็นได้ชัด คือ คอนเน็กเตอร์ของอุปกรณ์รับส่งปลายทางจะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของอุปกรณ์สื่อสารข้อมูลจะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไป จะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็ม จะเป็นแบบ DCE

ในปัจจุบันขาของคอนเน็กเตอร์ได้ถูกลดจำนวนลงเนื่องจากคอนเน็กเตอร์แบบ DB-25 นั้นมีสายสัญญาณถึง 2 ชุด ในหนึ่งตัวเป็นการสิ้นเปลือง ดังนั้นจึงมีคอนเน็กเตอร์แบบ DB-9 ไว้สำหรับใช้งาน

ขาของคอนเน็กเตอร์ตัวผู้และตัวเมียของ DB-9

CD: Carrier Detect ขานี้จะแอคทีฟเมื่อมีการส่งสัญญาณพาห์ จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม (Modem) สำหรับการใช้งาน ปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

RD: Received Data ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ ซึ่งจะนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

TD: Transmitted Data ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยจะนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

DTR: Data Terminal Ready เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบไม่ใช้โมเด็ม (Null Modem) ซึ่งให้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

SG: Signal Ground เป็นขากราวนด์ของระบบ

DSR: Data Set Ready ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้ จะเป็นขาสำหรับ รับข้อมูลจากภายนอก ซึ่งถูกส่งมาจากขา DTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTS: Request to send เป็นขาสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทาง ส่งข้อมูลมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบไม่ใช้โมเด็ม 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

CTS: Clear to send ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วง ว่าพร้อมที่จะรับข้อมูลหรือไม่

RI: Ring Indicator ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ โดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบนี้เท่านั้น

2.4 ไมโครคอนโทรลเลอร์ 8051

2.4.1 การรับส่งข้อมูลอนุกรม ใน MCS-51

การสื่อสารข้อมูลแบบอนุกรมเป็นการรับและส่งข้อมูลแบบทีละบิตข้อมูลเรียงกันไปโดยใช้สายสัญญาณเพียงสองหรือสามเส้น ต่างจากการสื่อสารแบบขนาน ซึ่งสามารถรับส่งข้อมูลได้เร็วกว่าแต่จะต้องใช้สายสัญญาณเท่ากับจำนวนบิตข้อมูลที่จะรับและส่งในแต่ละครั้ง ถ้าต้องการรับและส่งข้อมูลเป็นระยะทางไกล จะทำให้ต้องใช้สายสัญญาณเป็นจำนวนมาก และไม่ค่อยมีอุปกรณ์สนับสนุน การสื่อสารแบบอนุกรม จึงเป็นที่นิยมกว่า และถูกสร้างไว้เป็นมาตรฐานในไมโครคอนโทรลเลอร์หลายตระกูล ในตระกูล MCS-51 ก็เช่นกันที่มีพอร์ตที่ใช้ในการรับและส่งข้อมูลแบบอนุกรม อยู่ภายในชิปอยู่แล้ว ในการใช้งานเราไม่จำเป็นต้องต่อชิปสนับสนุนเพิ่มเติมอีก (กรณีที่ติดต่อกันเอง) การทำงานเป็นแบบการสื่อสารสองทิศทาง คือสามารถรับและส่งข้อมูลได้พร้อมๆกัน ในการรับข้อมูลจะมีบัฟเฟอร์(buffer) ในการรับข้อมูลให้ด้วย ทำให้ MCS-51 สามารถรับข้อมูลไบต์ (byte) ถัดไปได้เลยโดยไม่ต้องรอให้อ่านข้อมูลไบต์แรกไปเก็บก่อน ทำให้ข้อมูลไม่สูญหาย เกิดความคล่องตัวในการรับข้อมูล

การรับและส่งข้อมูลจะอาศัยรีจิสเตอร์ (register) ขนาด 8 บิตสองตัว ซึ่งแยกกันทำงานสองอย่างคือใช้รับข้อมูลที่มาจากภายนอกและใช้สำหรับส่งข้อมูลออกไปภายนอก รีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันใน SFR (Special function register) มีชื่อเรียกว่า SBUF (ตำแหน่ง 99H) การเข้าถึงข้อมูลในรีจิสเตอร์ทั้งสองนั้น MCS-51 จะทราบเองว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ SBUF ที่ใช้รับหรือส่งจากรหัสคำสั่ง เพราะ SBUF ที่ใช้รับข้อมูลจะเป็นตัวถูกดำเนินการ (Operand) ตัวแรกในคำสั่ง ส่วน SBUF ที่ใช้ส่งข้อมูลจะเป็นตัวถูกดำเนินการตัวที่สองในคำสั่ง การทำงานสามารถ กำหนดโหมดการทำงานได้ 4 แบบ โดยควบคุมจากค่าในรีจิสเตอร์ SCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. SCON

การใช้งานพอร์ตอนุกรมของ MCS-51 มีความคล่องตัวสูงมากทั้งนี้เนื่องจากผู้ใช้สามารถควบคุมการทำงานที่แตกต่างกันได้ถึง 4 ประเภท โดยควบคุมจากบิตต่างๆ ในรีจิสเตอร์ควบคุมพอร์ตอนุกรม SCON (ตำแหน่ง 99H) ซึ่งสามารถเข้าถึงได้ในระดับบิต

Bit	7	6	5	4	3	2	1	0
	SM0	SM1	SM2	REN	TB8	RB8	T1	R2

SM0 บิตเลือกการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมควบคุมได้โดยโปรแกรม

SM1 บิตเลือกการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมควบคุมได้โดยโปรแกรม

SM0	SM1	Mode	
0	0	0	ทำงานเป็น shift register อัตราเร็วในการรับหรือส่งข้อมูลเท่ากับ OSC/12
0	1	1	8 Bit UART อัตราเร็วในการรับหรือส่งข้อมูลกำหนดเองได้
1	0	2	9 Bit UART อัตราเร็วในการรับหรือส่งข้อมูล = OSC/32 หรือ OSC/64
1	1	3	9 Bit UART อัตราเร็วในการรับหรือส่งข้อมูลกำหนดเองได้

SM2 บิตการเลือกใช้งานพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 2 หรือ 3 เพื่อใช้ติดต่อระหว่างซีพียูด้วยกันเอง ควบคุมโดยใช้โปรแกรม

1 ใช้พอร์ตสื่อสารข้อมูลแบบอนุกรมติดต่อระหว่างซีพียูด้วยกันเอง เมื่อข้อมูลบิตที่ 9 ที่ได้รับมีค่าเป็น 0 (คาต้าไบต์) บิต RI จะไม่ถูกเซตแต่หากข้อมูลที่ 9 มีค่าเป็น 1 (แอดเดรสไบต์) บิต RI จะถูกเซต

0 ใช้พอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 1 และโหมด 3 ตามปกติในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 1 หากบิต SM2 ถูกเซตบิต RI จะไม่ถูกเซตจนกว่าบิตสิ้นสุดของข้อมูลจะถูกรับเข้ามา ในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 0 บิตนี้ควรถูกเคลียร์ให้เป็น 0

REN บิตควบคุมการอนุญาตให้มีการรับข้อมูลสามารถควบคุมได้โดยโปรแกรม

1 อนุญาตให้มีการรับข้อมูล

0 ไม่อนุญาตให้มีการรับข้อมูล

TB8 ข้อมูลบิตที่ 9 ซึ่งถูกส่งออกไปในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 2 และ 3 ควบคุมได้ด้วยคำสั่งในโปรแกรมเท่านั้น

RB8 ข้อมูลบิตที่ 9 ที่ได้รับเข้ามาจากภายนอก ในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 2 และ 3 ส่วนในการทำงานโหมด 1 ถ้าบิต SM2 = 0 บิตนี้จะเป็นบิตสิ้นสุดของข้อมูลที่รับเข้ามาได้และมาถูกกำหนดการใช้งานในโหมด 0

T1 บิตบอกสถานะสัญญาณอินเตอร์รัพท์ที่เกิดจากการส่งข้อมูลถูกเซตโดยฮาร์ดแวร์ เมื่อข้อมูลเริ่มส่งบิตสิ้นสุดของข้อมูลออกไปและจะต้องเคลียร์โดยคำสั่งในโปรแกรมเท่านั้น

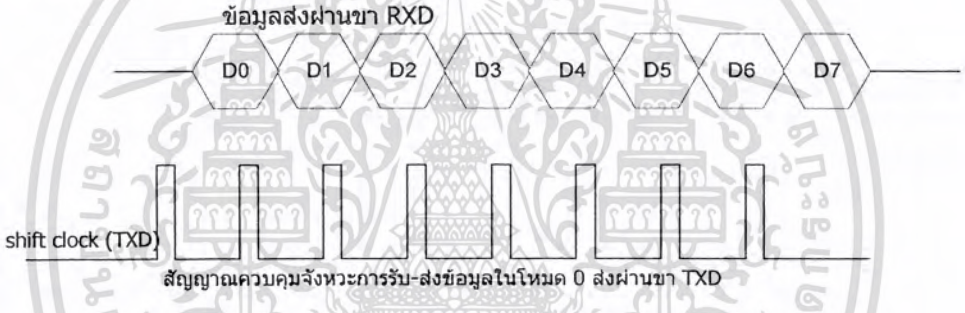
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R1 บิตบอกสถานะสัญญาณอินเตอร์รัพท์ที่เกิดจากการรับข้อมูล ถูกเซตโดยฮาร์ดแวร์เมื่อได้รับข้อมูลบิตที่ 8 ในการทำงานโหมด 0 หรือที่จุดครึ่งทางของช่วงรับบิตสิ้นสุดของข้อมูลในการทำงานโหมดอื่นและจะต้องเคลียร์ในโปรแกรมเท่านั้น

2.4.2 โหมดของการรับ-ส่งข้อมูลแบบอนุกรม

โหมด 0 (Shift Register Mode)

ขา P3.0 (RXD) ถูกใช้เป็นขาสำหรับรับและส่งข้อมูล ขา P3.1(TXD) ใช้สร้างสัญญาณชิฟคล็อก (Shift Clock) เพื่อกำหนดจังหวะในการรับและส่งข้อมูล การส่งข้อมูลจะเริ่มทันทีที่มีการใช้งานรีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง (destination register) ข้อมูลจะถูกเลื่อนออกไปทีละ 1 บิต โดยเริ่มจากบิตต่ำสุด (LSB) ก่อน และจะมีบิต 0 เลื่อนเข้ามาแทนที่บิตสูงสุดที่ถูกเลื่อนไป สัญญาณที่ขา TXD จะมีความถี่เท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ ถูกส่งออกมาพร้อมๆ กับบิตข้อมูลดังรูป 000 การรับข้อมูลจะเกิดขึ้นได้ต่อเมื่อบิต REN = 1 และ RI = 0 โดยที่ขา TXD จะมีสัญญาณควบคุมการรับข้อมูลออกมาพร้อม กับการรับข้อมูลแต่ละบิต ดังรูปที่ 2.8 อัตราการรับส่งข้อมูลถูกกำหนดไว้ที่ 1/12 ของความถี่ออสซิลเลเตอร์ที่ใช้



รูปที่ 2.8 สัญญาณการรับและส่งข้อมูลในโหมด 0

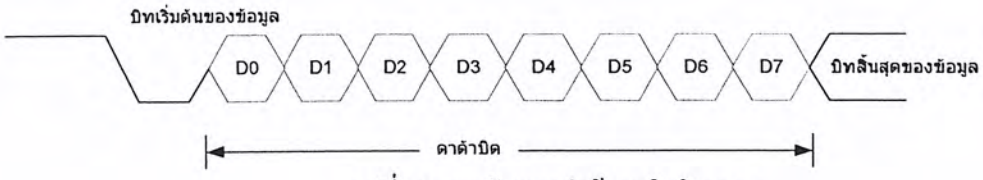
โหมด 1 (Standard 8 bit UART Mode)

ขา P3.0 (RXD) ถูกใช้เป็นขาสำหรับ รับข้อมูล ขา P3.1 (TXD) ถูกใช้เป็นขาสำหรับส่งข้อมูล การรับ-ส่งข้อมูลจะเป็นแบบ 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต (เป็น 0 เสมอ) บิตข้อมูล 8 บิต (รับและส่งบิตต่ำสุดก่อน) และบิตสิ้นสุดของข้อมูล 1 บิต (เป็น 1 เสมอ) ดังรูปที่ 2.9 ในการส่งข้อมูลจะเริ่มต้นโดยคำสั่งใดๆ ที่มีการใช้งานรีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง บิต TI จะเซตเมื่อข้อมูลถูกเลื่อนออกไปจนครบทั้ง 10 บิตแล้ว การรับข้อมูลจะเริ่มต้นทันทีที่ลอจิกที่ขา RXD เปลี่ยนสภาวะจาก 1 เป็น 0 ข้อมูลจะถูกเลื่อนเข้ามาใน Buffer สำหรับรับข้อมูล เมื่อข้อมูลเข้ามาครบทั้ง 8 บิตแล้ว จึงจะถูกโหลดไปไว้ที่รีจิสเตอร์ SBUF พร้อมกับบิตสิ้นสุดของข้อมูลถูกเก็บไว้ที่บิต RB8 ของรีจิสเตอร์ SCON บิต RI จะเซต เป็นการบอกให้รู้ว่ากระบวนการรับข้อมูลจำนวน 1 ไบท์ได้เสร็จสิ้นลงแล้ว และเริ่มรีเซตตัวเองเพื่อกลับไปรอรับการเปลี่ยนสภาวะจาก 1เป็น 0 ที่ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RXD อีก ทั้งนี้กระบวนการรับข้อมูลจะเป็นจริงก็ต่อเมื่อ

- บิต RI = 0 และ
- บิต SM2 = 0 หรือ ถ้า SM2 = 1 บิตสิ้นสุดของข้อมูลจะต้อง = 1 ด้วย

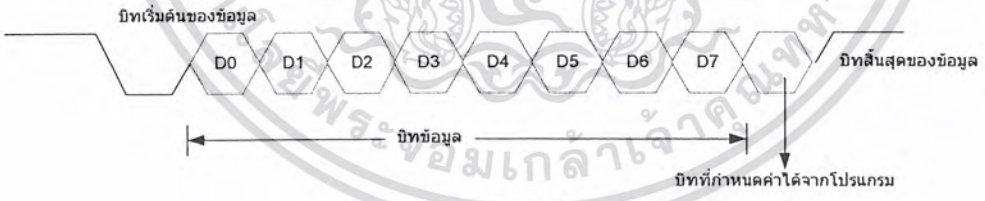


รูปที่ 2.9 การรับและส่งข้อมูลในโหมด 1

โหมด 2 และ 3 (Multiprocessor)

ขา P3.0 (RXD) ถูกใช้เป็นขาสำหรับรับข้อมูล ขา P3.1 (TXD) ถูกใช้เป็นขาสำหรับส่งข้อมูล การรับ-ส่งข้อมูลจะเป็นแบบ 11 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต (เป็น 0 เสมอ) บิตข้อมูล 8 บิต (รับและส่งบิตต่ำสุดก่อน) บิตที่กำหนดได้ 1 บิต (โปรแกรมเองได้ว่าจะเป็น 1 หรือ 0) และบิตสิ้นสุด 1 บิต (เป็น 1 เสมอ) ดังรูปที่ 2.10

การส่งข้อมูลจะเริ่มต้นโดยคำสั่งใดๆ ที่มีการใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง บิต TI จะเซตเมื่อข้อมูลถูกเลื่อนออกไปจน ครบทั้ง 11 บิตแล้ว บิตที่ 9 จะได้จากบิต TB8 ในรีจิสเตอร์ SCON ส่วนในขณะรับข้อมูลบิตที่ 9 จะอยู่ในบิต RB8 ของรีจิสเตอร์ SCON โดยไม่สนใจบิตสิ้นสุดของข้อมูล การรับข้อมูลจะใช้การตรวจสอบการเปลี่ยนแปลงสภาวะลอจิกจาก 1 เป็น 0 ที่ขา RXD และมีเงื่อนไขในการรับข้อมูลเช่นเดียวกับโหมด 1 ในการทำงานโหมด 2 อัตราการรับและส่งข้อมูลสามารถเปลี่ยนแปลงได้



รูปที่ 2.10 การรับและส่งข้อมูลในโหมด2 และ3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. PCON

เป็นรีจิสเตอร์สำหรับควบคุมการทำงานของ 8051 ซึ่งเป็นรีจิสเตอร์ที่ไม่สามารถอ้างอิงได้ ดังมีรายละเอียดดังนี้

SMOD	-	-	-	GF1	GF0	PD	IDL

SMOD เป็นบิตกำหนดอัตราความเร็วของการรับส่งข้อมูลแบบอนุกรมที่ใช้ Timer1 เช่นถ้าเวลาในการรับส่งเมื่อ SMOD = 1 จะทำให้ความเร็วในการรับส่งแบบอนุกรมเป็น 2 เท่า เมื่อเลือกการใช้งานใน Mode 1, 2, 3 ของการรับส่งแบบอนุกรม

- เป็นบิตที่ไม่ได้ใช้งานซึ่งสงวนไว้สำหรับหน้าที่เพิ่มเติมในอนาคตของ 8051 ดังนั้นผู้ใช้จึงไม่ควรกำหนดในบิตนี้เป็น 1

GF0 – GF1 เป็นบิตอเนกประสงค์ที่ผู้ใช้สามารถกำหนดการทำงานได้

PD เป็นบิตสำหรับให้ 8051 เข้าสู่การทำงานใน Power Down Mode ซึ่งจะมีเฉพาะเบอร์ที่มีโครงสร้างภายในเป็น CMOS เท่านั้น ซึ่งจะทำงานเมื่อบิตนี้เป็น 1

IDL เป็นบิตสำหรับกำหนดให้ 8051 เข้าสู่การทำงานใน Idle Mode ซึ่งจะมีเฉพาะเบอร์ที่มีโครงสร้างภายในเป็น CMOS เท่านั้น ซึ่งจะทำงานเมื่อบิตนี้เป็น 1

ถ้าบิต PD และ IDL เป็น 1 ทั้งคู่ในเวลาเดียวกัน 8051 จะทำงานใน Power Down Mode ในการทำงานทั้งสอง Mode นี้เป็นการทำงานเพื่อประหยัดพลังงานของ 8051 โดยมีข้อแตกต่างดังนี้

Power Down Mode เมื่อมีการกำหนดให้บิตนี้เป็น 1 8051 จะทำงานอีกหนึ่งคำสั่งก่อนที่จะเข้าสู่ Power Down Mode โดยที่ส่วนสร้างความถี่อ้างอิงภายในจะหยุดทำงาน การทำงานทุกอย่างของ 8051 จะหยุด ข้อมูลต่างๆ ของหน่วยความจำภายในรวมทั้ง SFR จะถูกรักษาไว้ ที่สัญญาณ ALE และ PSEN จะเป็นลอจิก 0 เมื่อเข้าสู่ Power Down Mode แรงดันไฟฟ้าที่จ่ายให้กับ 8051 สามารถลดลงเป็น 2 V ได้ อย่างไรก็ตามแรงดันไฟฟ้าจะต้องไม่ลดลงก่อนที่จะเข้าสู่ Power Down Mode และแรงดันไฟฟ้าจะต้องเข้าสู่ระดับปกติก่อนที่จะออกจาก Power Down Mode การออกจาก Power Down Mode จะกระทำได้เฉพาะทำการรีเซต 8051 เท่านั้น ซึ่งจะทำให้ค่าของ SFR เปลี่ยนแปลง แต่ข้อมูลในส่วนของหน่วยความจำภายในไม่เปลี่ยนแปลง

Idle Mode เมื่อมีการกำหนดให้บิตนี้เป็น 1 8051 จะทำงานอีกหนึ่งคำสั่งก่อนเข้าสู่ Idle Mode โดยที่ความถี่การทำงานภายในซีพียู ของ 8051 จะหยุดทำงานยกเว้นที่ให้กับส่วนของการอินเตอร์รัพท์ Timer และการรับส่งแบบอนุกรม ค่าต่างๆ ของรีจิสเตอร์จะถูกเก็บรักษาไว้ และสัญญาณของ ALE และ PSEN จะอยู่ที่ลอจิก 1 ในการออก Idle Mode จะสามารถทำได้สองวิธีได้แก่การรีเซตค่าและการอินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. TMOD

เป็นรีจิสเตอร์สำหรับควบคุม Mode การทำงานของ Timer/Counter และไม่สามารถอ้างอิงระดับบิตได้ ซึ่งจะแบ่งเป็นสองส่วน โดยที่สี่บิตแรกของรีจิสเตอร์นี้ใช้สำหรับ Timer/Counter 0 และสี่บิตบนสำหรับ Timer/Counter 1 มีรายละเอียดดังนี้

Timer 1				Timer 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

GATE เป็นบิตใช้สำหรับการควบคุม Timer/Counter ถ้าเป็นลอจิก 1 Timer/Counter เมื่อขาสัญญาณที่ขา INTx เป็นลอจิก 1 เท่านั้น และบิต TRx ใน TCON ต้องเป็นลอจิก 1 ด้วย ซึ่งจะเป็นการควบคุมด้วยฮาร์ดแวร์ เมื่อเป็นลอจิก 0 การควบคุม Timer/Counter นั้นๆ จะเป็นการควบคุมด้วยโปรแกรมที่บิต TRx ของรีจิสเตอร์ TCON เท่านั้น C/Tเป็นบิตสำหรับเลือกการทำงานของ Timer/Counter เป็น Timer หรือ Counter โดยที่ถ้าเป็นลอจิก 1 จะทำหน้าที่เป็น Counter โดยรับสัญญาณจากขา Tx ของ 8051 และลอจิก 0 เป็น Timer โดยความถี่ที่นับได้จากความถี่ภายในของ 8051 M0, M1

M0, M1 เป็นบิตสำหรับเลือกการทำงานของ Timer/Counter ดังแสดงในตารางที่ 2.1 ต่อไปนี้

M0	**M1**	**การทำงาน**
0	0	เป็น Timer ขนาด 13 บิต เมื่อการทำงานของตระกูล 48
0	0	เป็น Timer/Counter ขนาด 16 บิต
1	0	เป็น Timer/Counter ขนาด 8 บิต ที่สามารถตั้งค่าใหม่อัตโนมัติ
1	1	Timer/Counter 0 จะถูกแบ่งออกเป็น Timer/Counter 2ชุด ขนาด 8 บิต โดยที่ TL0 จะเป็น Timer/Counter ขนาด 8 บิต ที่ถูกควบคุมของ Timer/Counter 0 และTH0 จะเป็น Timer/Counter ขนาด 8 บิต ที่ควบคุมโดยชุดควบคุมของ Timer/Counter1
1	1	Timer/Counter 1 จะหยุดทำงาน

ตารางที่ 2.1 การเลือกการทำงานของ Timer/Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณความเร็วการรับและส่งข้อมูลแบบอนุกรม (Generating Baud Rate)

การกำหนดความเร็วการรับส่งข้อมูลแบบอนุกรมสามารถแบ่งออกตาม Mode การทำงานดังนี้

Mode 0

ความเร็วในการรับส่งข้อมูลแบบอนุกรมใน Mode นี้จะกำหนดอัตราการรับส่งตายตัวเท่ากับ 1/12 ของความถี่ของชุดกำเนิดความอ้างอิงของ 8051 และจะไม่ใช้ Timer/Counter ดังนั้นเพียงกำหนดที่ รีจิสเตอร์ SCON ก็เพียงพอ จะได้

Baud Rate = Osc. Freq / 12

Mode 1

ในการกำหนดความเร็วการรับส่งข้อมูลแบบอนุกรมใน Mode 1 นี้จะใช้ Timer 1 เป็นฐานเวลาของการทำงานของ Timer 1ใน Mode 2 (Auto-Reload) โดยสามารถคำนวณได้ดังนี้

Baud Rate = [K * Osc. Freq] / [32*12*(256-(TH1))]

K = 1 เมื่อ SMOD ในรีจิสเตอร์ PCON = 0

K = 2 เมื่อ SMOD ในรีจิสเตอร์ PCON= 1

ส่วนมากแล้ว ผู้ใช้จะทราบค่าของ Baud Rate ที่จะส่งได้ค่านั้น จะได้ค่าของ Time 1 สำหรับ Reload ได้เป็น

TH1 = 256 – [(K * Osc .Freq) / (384 * Baud Rate)]

จากตารางที่ 2.2 แสดง Baud Rate ต่าง ๆ และค่า Reload ของ Time 1

Baud Rate	f_{osc}	SMOD	Timer1		
			C/T	Mode	Reload Value
Mode 0 Max :1 MHz	12 MHz	X	X	X	X
Mode 2 Max : 375 K	12 MHz	1	X	X	X
Modes 1.3 : 62.5 K	12MHz	1	0	2	FFH
19.2 K	11.059 MHz	1	0	2	FDH
9.6 K	11.059 MHz	0	0	2	FDH
4.8 K	11.059 MHz	0	0	2	FAH
2.4 K	11.059 MHz	0	0	2	F4H
1.2 K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.2 Baud Rate ต่างๆ และค่า Reload ของ Time 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode 2

ความเร็วการรับส่งใน Mode นี้จะเป็นค่าคงที่ซึ่งมี 2 ค่า ขึ้นกับค่า SMOD ในรีจิสเตอร์ PCON ดังนี้

เมื่อ SMOD = 1 Baud Rate = 1/32 Osc. Freq

เมื่อ SMOD = 0 Baud Rate = 1/64 Osc. Freq

Mode 3

การกำหนดความเร็วการรับส่งใน Mode 3 จะคิดเช่นเดียวกับการคิดใน Mode 1สำหรับค่าเริ่มต้นของรีจิสเตอร์ SFR เมื่อ 8051 ถูกรีเซต จะมีค่าดังแสดงในตารางต่อไปนี้

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	FFH
IP (8051)	XXX0000B
IP (8052)	XX000000B
IE (8051)	0XX00000B
IE (8052)	0X000000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2 (8052)	00H
TL2 (8052)	00H

ตารางที่ 2.3 ค่าเริ่มต้นของรีจิสเตอร์ SFR เมื่อ 8051 ถูกรีเซ็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SFR Name	Reset Value
RCAP2H (8052)	00H
RCAP2L (8052)	00H
SCON	00H
SBUF	Indeterminate
PCON (HMOS)	0XXXXXXXXB
PCON (CHMOS)	0XXX0000B

ตารางที่ 2.4 ค่าเริ่มต้นของรีจิสเตอร์ SFR เมื่อ 8051 ถูกรีเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

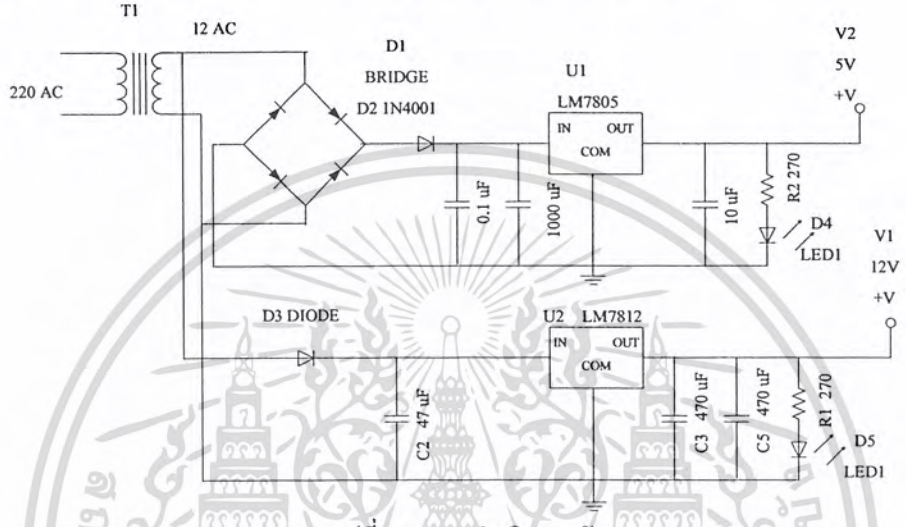
บทที่ 3
การคำนวณและการสร้าง

แบ่งเป็นสองส่วน คือ ส่วนฮาร์ดแวร์และส่วนซอร์ฟแวร์

<u>การคำนวณและการสร้างส่วนฮาร์ดแวร์</u>

3.1 วงจรกำเนิดแรงดัน

วงจรกำเนิดแรงดัน ดังรูปที่ 3.1 จะทำหน้าที่เป็นวงจรกำเนิดแรงดันเพื่อจ่ายไปเลี้ยงส่วนต่างๆของวงจรให้วงจรทำงานได้ซึ่งวงจรต้องการไฟเลี้ยงทั้งขนาด +12โวลต์และขนาด +5โวลต์



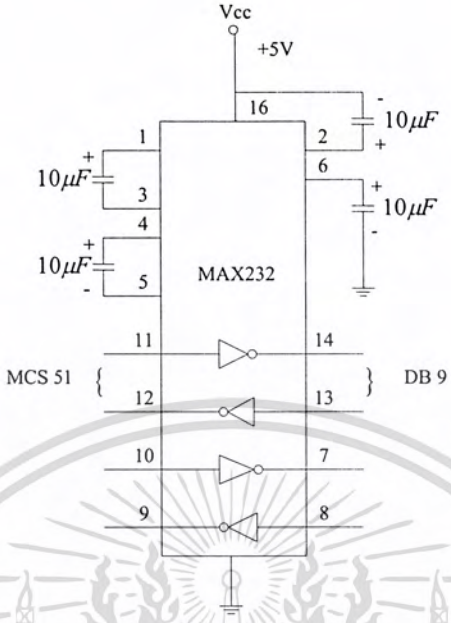
รูปที่ 3.1 วงจรกำเนิดแรงดัน

การทำงานของวงจรเริ่มจาก มีการรับแรงดันไฟฟ้ากระแสสลับ 220 โวลต์ เข้าที่หม้อแปลง ซึ่งมีการแปลงแรงดันให้เหลือ +12โวลต์ ไฟฟ้ากระแสสลับ และผ่านวงจรบริดจ์ (Bridge) เพื่อทำหน้าที่แปลงแรงดันไฟฟ้ากระแสสลับให้เป็นกระแสตรง ป้อนผ่าน D2 โดยมีตัวเก็บประจุเป็นตัวกรองแรงดันไฟให้เรียบต่อไปยัง LM7805 เพื่อปรับแรงดันให้ได้แรงดันคงที่ +5โวลต์ หลังจากนั้นก็ทำการมีตัวเก็บประจุเป็นตัวกรองแรงดันไฟให้เรียบอีก จากนั้นจะได้แรงดันไฟฟ้า +5โวลต์ เพื่อจ่ายไปยังวงจรต่อไป ส่วนของแรงดันไฟ +12 โวลต์ นั้น มีลักษณะการทำงานเหมือนกับแรงดันไฟ +5โวลต์ ต่างกันที่ตัวปรับแรงดันของไฟ +12โวลต์ เป็น LM7812 จากนั้นจะได้แรงดันไฟ +12โวลต์ เพื่อไปจ่ายยังวงจรต่อไป

3.2 วงจรแปลงสัญญาณแบบ RS-232

ใช้ไอซี Max232 ซึ่งทำหน้าที่ปรับระดับสัญญาณให้เหมาะสม เชื่อมต่อกับคอมพิวเตอร์ที่ขา RxD และขา TxD ของ com1 หรือ com2 มีข้อดีคือ มีทั้งไดร์เวอร์ (driver) และรีซีฟเวอร์ (receiver) ในตัวเอง และใช้โวลท์เตจระดับเดียว คือ 5 โวลต์ การนำเอาไอซี Max232 ไปใช้นั้น ต้องมีการต่อตัวเก็บประจุเข้าไปอีกเล็กน้อย ซึ่งค่าของตัวเก็บประจุที่ใช้ในการทดลองคือ C = 10 uF ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

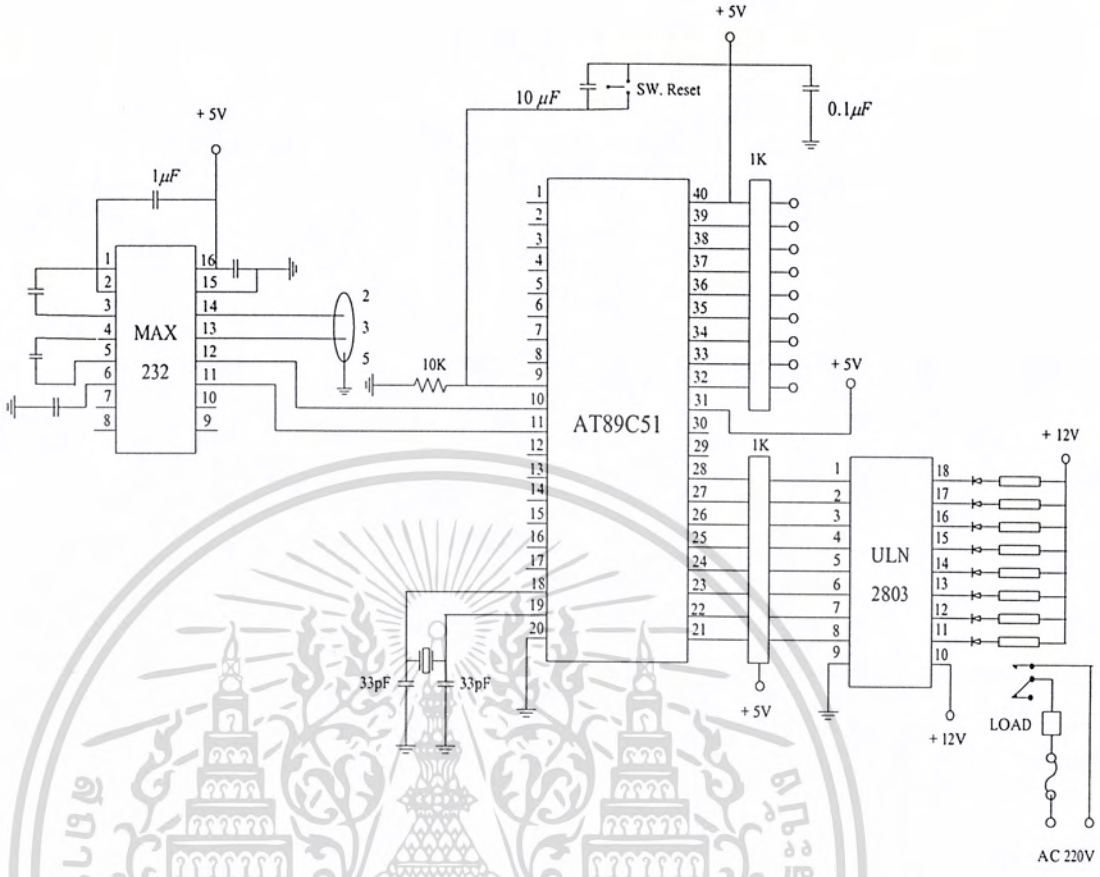


รูปที่ 3.2 วงจรแปลงสัญญาณแบบ RS-232

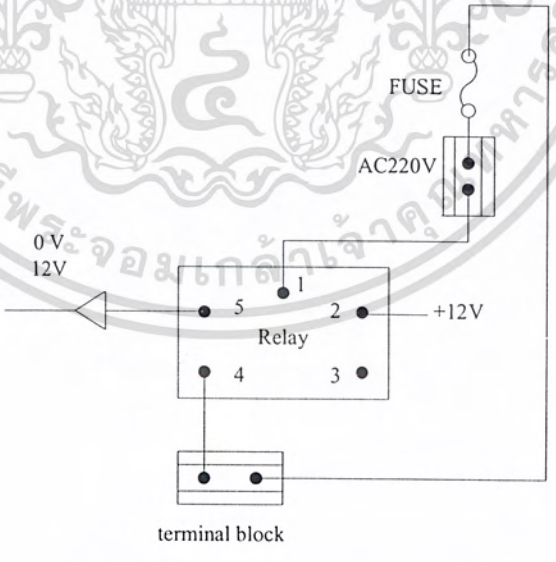
3.3 วงจร ไมโครคอนโทรลเลอร์

วงจรไมโครคอนโทรเลอร์เป็นวงจรควบคุมการทำงานต่างๆของวงจรรวมดังรูปที่ 3.3 จะเริ่มจากการรับค่าจากคอมพิวเตอร์ผ่าน RS-232 เข้ามาค่าที่ส่งเข้ามาจะถูกนำมาตรวจสอบ โดยจะนำค่ามาเปรียบเทียบกับค่าที่ตั้งไว้หรือค่าที่กำหนดให้กับโหลดตัวนั้นๆถ้าค่าที่ส่งมาตรงกับค่าของโหลดตัวใดไอซีไมโครคอนโทรลเลอร์ก็จะทำการกลับสถานะบิตแล้วนำสถานะบิตนั้นส่งไปยังไอซี ULN 2803 ซึ่งเป็นไอซีไดร์เวอร์ โดยภายในจะบรรจุอินเวอร์เกตแบบคอเลกเตอร์เปิดจำนวน 7 ตัว โดยสามารถใช้กับแรงดันสูงสุดถึง 50 โวลต์ และให้กระแสเอาท์พุทในแต่ละขาได้ถึง 500 mA นอกจากนี้ยังมีไดโอดป้องกันไว้ที่ทุกขาของเอาท์พุต ทำให้สามารถต่อกับโหลดที่เป็นขดลวดได้ โดยจะทำหน้าที่ขับรีเลย์ให้ทำงานและทำให้ LED แสดงผลการใช้งานของในแต่ละช่องการต่ออุปกรณ์ที่ขาต่างๆของรีเลย์จะเป็นไปตามรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรรวมของไมโครคอนโทรลเลอร์

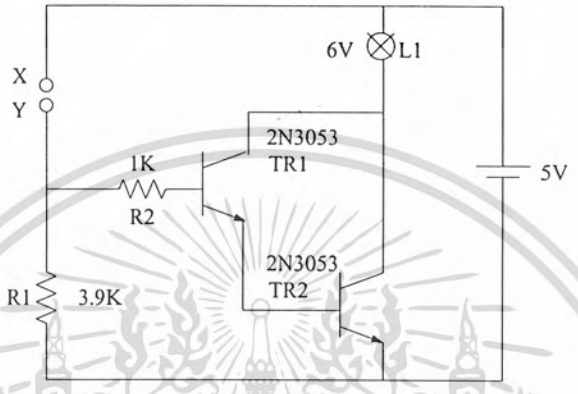


รูปที่ 3.4 การต่ออุปกรณ์ที่ขาต่างๆของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจรเตือนน้ำเต็ม

เนื่องจากน้ำที่ไม่บริสุทธิ์นั้นจะเป็นตัวนำไฟฟ้าได้ดี ดังนั้นเราจึงนำมาประยุกต์ใช้โดยต่อวงจรดังรูป 3.5 และเมื่อระดับน้ำสูงขึ้นจนถึงโพรบจะเกิดกระแสไหลผ่านน้ำและ R2 เข้าไปยังขาเบสของ TR1 ออกจากขาอิมิตเตอร์เข้าขาเบสของ TR2 และออกจาก TR2 ผ่านขาอิมิตเตอร์ ไปยังกราวนด์ ทำให้ครบวงจรกระแสเบสจะกระตุ้นให้วงจรขยายคู่ดาร์ลิงตันทำงาน และถ้ากระแสคอลเลกเตอร์สูงมากพอ จะทำให้ L1 สว่างได้

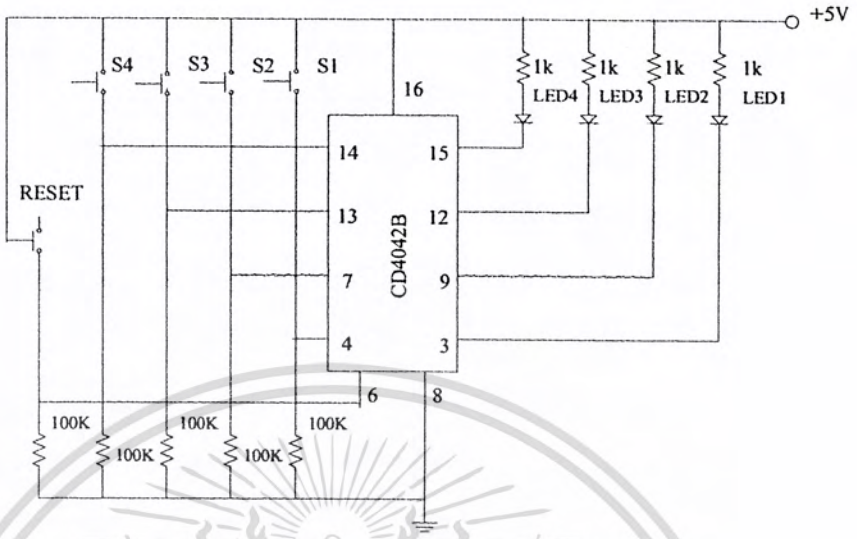


รูปที่ 3.5 วงจรเตือนน้ำเต็ม

3.5 วงจรสวิทช์

เป็นวงจรที่ทำให้เราทราบว่ามีผู้บุกรุกเข้ามาในบ้านทางใด ถ้าสวิทช์ตัวใดถูกกด สมมติว่าเป็น S1 ถูกกดขา 4 จะมีสภาวะเป็น “High” และขา 3 จะมีสภาวะเป็น “Low” ทำให้ LED1 สว่าง และถ้า S2 ถูกกดขา 7 จะมีสภาวะเป็น “High” และขา 9 จะมีสภาวะเป็น “Low” ทำให้ LED2 สว่าง และถ้า S3 ถูกกดขา 13 จะมีสภาวะเป็น “High” และขา 12 จะมีสภาวะเป็น “Low” ทำให้ LED2 สว่าง และสุดท้ายถ้า S4 ถูกกดขา 14จะมีสภาวะเป็น “High” และขา 15 จะมีสภาวะเป็น “Low” ทำให้ LED2 สว่าง ตามลำดับ วงจรสวิทช์แสดงได้ดังรูป 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรสวิทช์

3.6 วงจรตรวจจับทางแสง

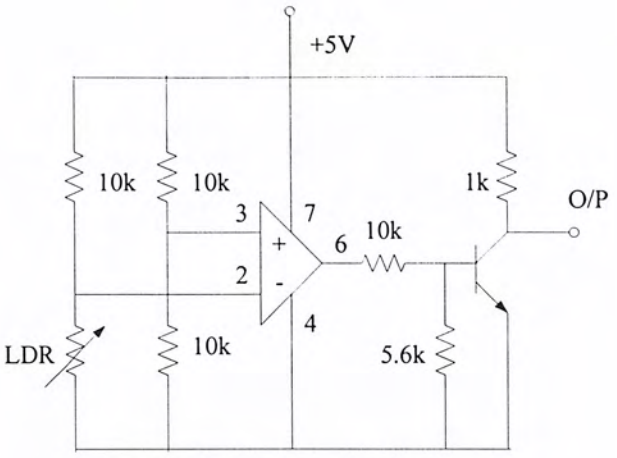
ในวงจรนี้จะใช้ตัวตรวจจับทางแสงที่เรียกว่า Light Dependent Resistance (LDR) กล่าวคือ ความต้านทานจะเปลี่ยนไปตามความเข้มของแสงนั้นเอง ถ้าความเข้มแสงมากความต้านทานของมันจะต่ำหาก กลับกันความเข้มแสงน้อยความต้านทานจะสูง

1. ทำการทดลองวัดค่าความต้านทานของ LDR ในช่วงเวลาตามตารางที่ 3.1 นี้

ช่วงเวลาในการวัดความต้านทาน	ค่าความต้านทาน
กลางวัน	20 k Ω
กลางคืน	550 k Ω

ตารางที่ 3.1 การทดลองวัดค่าความต้านทานของ LDR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรตรวจจับทางแสง

2. จากรูปเป็นวงจรตามรูปที่ 3.7 ที่ใช้ตรวจจับแสงโดยใช้ไอซี LM741 ต่อเป็นวงจรเปรียบเทียบแรงดัน การทำงานของวงจรเป็นดังนี้ คือ

แรงดันตกคร่อมขา 4 = 2.5 โวลต์ เป็น Vref

ที่สถานะสว่างมีแรงดันตกคร่อมขา 7 = $5\times 20K/(20K+50K) = 1.43$ โวลต์

ที่สถานะมืดมีแรงดันตกคร่อมขา 7 = $5\times 550K/(20K+550K) = 4.82$ โวลต์

อัตราขยายของออปแอมป์ $Av = 20{,}000 - 100{,}000$

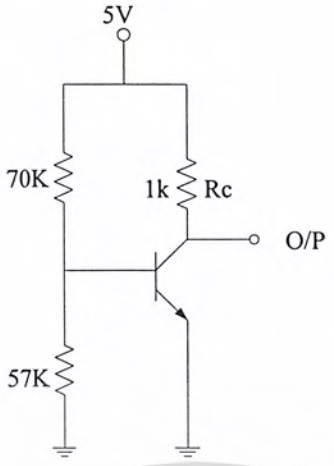
ที่สถานะสว่างมีแรงดัน O/P ที่ออปแอมป์ $Vo = (100{,}000)(2.5-1.43) = 10700$ แต่เป็นไปไม่ได้ ในทางปฏิบัติเพราะ O/P จ่ายแค่ 5 โวลต์

ที่สถานะมืดมีแรงดัน O/P ที่ออปแอมป์ $Vo = (100{,}000)(2.5-4.82) = -232000$ แต่เป็นไปไม่ได้ ในทางปฏิบัติเพราะ O/P จ่ายแค่ 0 โวลต์

ดังนั้นที่สถานะมืด แรงดัน O/P ที่ออปแอมป์ เป็น0 โวลต์ ทรานซิสเตอร์จึงไม่ทำงาน

ที่สถานะสว่าง ทรานซิสเตอร์จึงทำงาน มีแรงดันตกคร่อมขาเบสเท่ากับ 0.7 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรตรวจจับทางแสง

$$I_1 = 0.7/57K = 1.23\mu A$$

$$I_T = (5-0.7)/70K = 61.42\mu A$$

ดังนั้น $I_B = I_T - I_1$

$$= 49\mu A$$

จาก datasheet β มีค่าประมาณ 100

$$I_C = \beta I_B$$

$$= 100 \times 49\mu A$$

$$= 4.9mA$$

ดังนั้น $V_{RC} = I_C \times R_C$

$$= 4.9 \text{ โวลต์}$$

ดังนั้นที่สถานะมืด ทรานซิสเตอร์จึงไม่ทำงาน ทำให้ O/P เท่ากับ 5 โวลต์

ดังนั้นที่สถานะสว่าง ทรานซิสเตอร์ทำงาน ทำให้ O/P เท่ากับ 0 โวลต์

3.7 วงจรควบคุมอุณหภูมิ

วงจรนี้หน้าที่หลักอยู่ที่ตัวเทอร์มิสเตอร์ RTC และไอซี LM 741 เราจะใช้วิธีการเปรียบเทียบแรงดันระหว่างขา 2 และ ขา 3 เมื่อแรงดันขา 3 สูงกว่าขา 2 จะทำให้เอาต์พุตของไอซี มีค่าประมาณ 5 โวลต์ จากวงจรเราใช้ไฟ 5 โวลต์ป้อนเข้าไปเลี้ยงวงจรผ่านไดโอดและเมื่อไอซีได้รับไฟเลี้ยงก็จะทำการเปรียบเทียบแรงดันระหว่างขาอินพุตทั้ง 2 โดยมีตัวต้านทานและตัวเทอร์มิสเตอร์ที่จะใช้เป็นเซนเซอร์อุณหภูมิต่ออยู่กับขาอินพุตดังรูปที่ 3.9

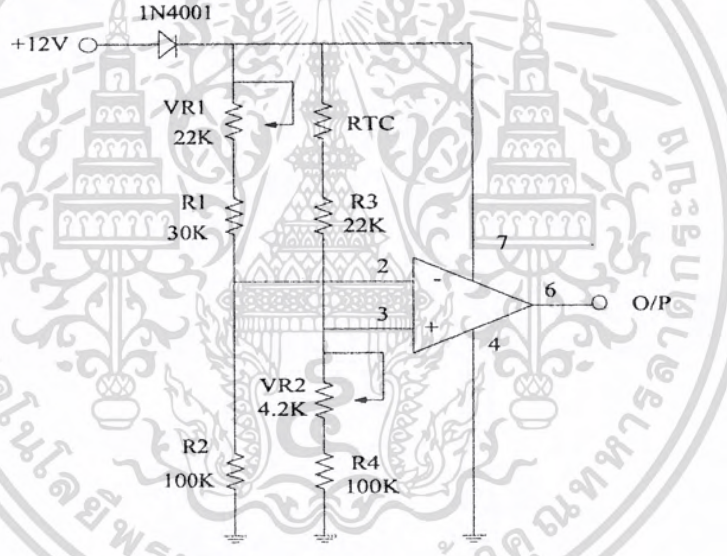
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์อุณหภูมิอยู่ในช่วง -10 ถึง 125 องศาเซลเซียส เมื่อได้รับอุณหภูมิต่ำจะมีความต้านทานสูงถึงประมาณ 200 กิโลโอห์ม แต่ถ้าได้รับอุณหภูมิสูง ค่าความต้านทานจะลดลงเหลือไม่กี่ร้อยโอห์ม คุณสมบัติโดยรวมแล้วจะไวต่ออากาศเย็นหรือที่อุณหภูมิต่ำ

อินพุทขา 3 หรือ ขาบวก ในอุณหภูมิปกติประมาณ 30 องศาเซลเซียส ตัวเทอร์มิสเตอร์จะมีความต้านทานประมาณ 10 ถึง 20 กิโลโอห์ม เมื่อรวมกับ R3 แล้วจะมีความต้านทานประมาณ 30 ถึง 40 กิโลโอห์ม VR2 จึงทำหน้าที่ปรับความต้านทานให้มากหรือน้อย เพื่อให้แรงดันที่ขา 3 ของไอซี ต่ำกว่าแรงดันที่ขา 2 เอาไว้

หากอุณหภูมิต่ำๆเทอร์มิสเตอร์จะมีความต้านทานสูง ทำให้แรงดันตกคร่อมที่ 3 เทียบกับกราวน์จะต่ำลง ส่งผลให้ขานี้มีแรงดันต่ำกว่าขา 2 สภาวะเอาต์พุทที่ขา 6 จะมีแรงดันประมาณ 0 โวลต์

แต่ถ้าอุณหภูมิสูงขึ้นเทอร์มิสเตอร์จะมีความต้านทานต่ำลง ส่งผลให้แรงดันที่แบ่งระหว่าง RTC กับ R3 น้อยลง แรงดันที่ตกคร่อม VR2 กับ R4 มากขึ้น เมื่อมากจนถึงระดับเท่ากับแรงดันที่ขา 2 หรือมากกว่า ทำให้เอาต์พุทของไอซีมีแรงดันเท่ากับ 5 โวลต์



รูปที่ 3.9 วงจรควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณและการสร้างส่วนซอร์ฟแวร์

3.8 ส่วนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

โดยขั้นตอนการสร้างจะใช้

3.8.1 โปรแกรม SXA51

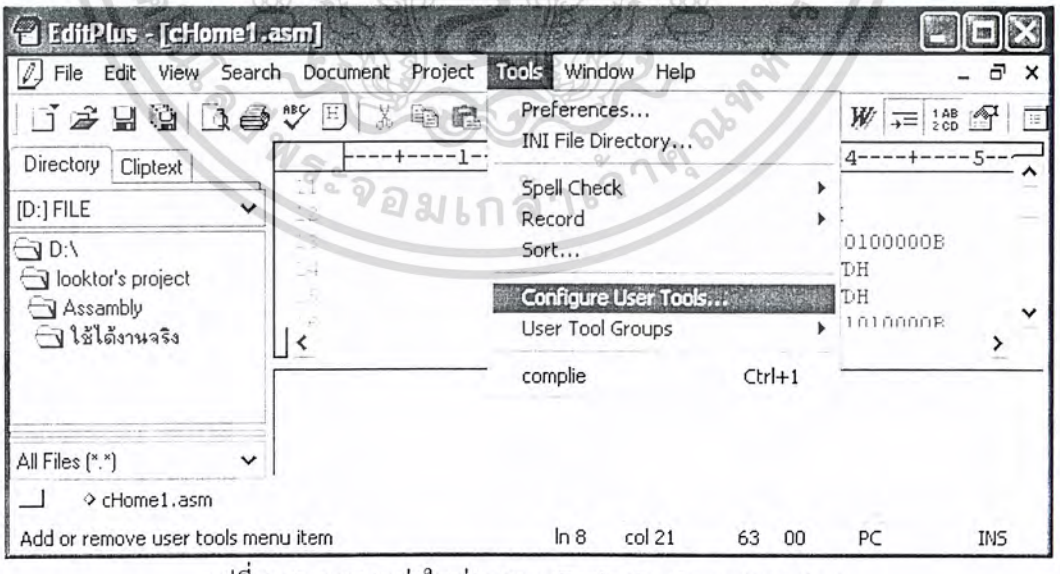
เป็นโปรแกรมสำหรับคอมไฟล์ภาษาแอสแซมบลี ซึ่งสามารถดาวน์โหลดโปรแกรม SXA51 ได้ที่ www.download.com เมื่อได้มาแล้วให้ copy โฟลเดอร์โปรแกรมไปไว้ที่ไดร์ฟ C หรือ D ก็ได้ และทำการทดสอบการทำงานโดยนำไฟล์สมมติชื่อ AAA.asm ไปไว้ในโฟลเดอร์โปรแกรม SXA51 จากนั้นไปที่ RUN พิมพ์คำสั่ง cmd แล้วเข้าไปที่ C:\SXA51\sxa51 AAA.asm แล้วกด Enter จากนั้นเข้าไปดูที่โฟลเดอร์ของโปรแกรม SXA51 ก็จะได้ ไฟล์นามสกุล .HEX

3.8.2 โปรแกรม Editplus2

โปรแกรม editplus เป็นเพียง editor มิใช่คอมไพล์เลอร์ แต่เป็นโปรแกรมที่ช่วยเรียก คอมไพล์เลอร์ มาแปลโปรแกรมที่กำลังเขียนอยู่ พร้อมกับรันโดยเพิ่มเข้าไปในส่วนของ Tools, Configure User Tools ... ทำให้สามารถเรียกโปรแกรมอื่น มาทำงาน โดยแสดงผลในส่วนของ Output windows ว่าผลการรันหรือคอมไพล์เป็นอย่างไร โดยเรียกโปรแกรม ด้วยการกดปุ่ม CTRL+1 หรือ CTRL+2 เป็นต้น ในโครงงานนี้จะใช้โปรแกรม editplus เข้ามาช่วยในการเรียกคอมไพล์เลอร์ ของโปรแกรม SXA51 มาแปลโปรแกรมที่เขียนด้วยภาษาแอสแซมบลี

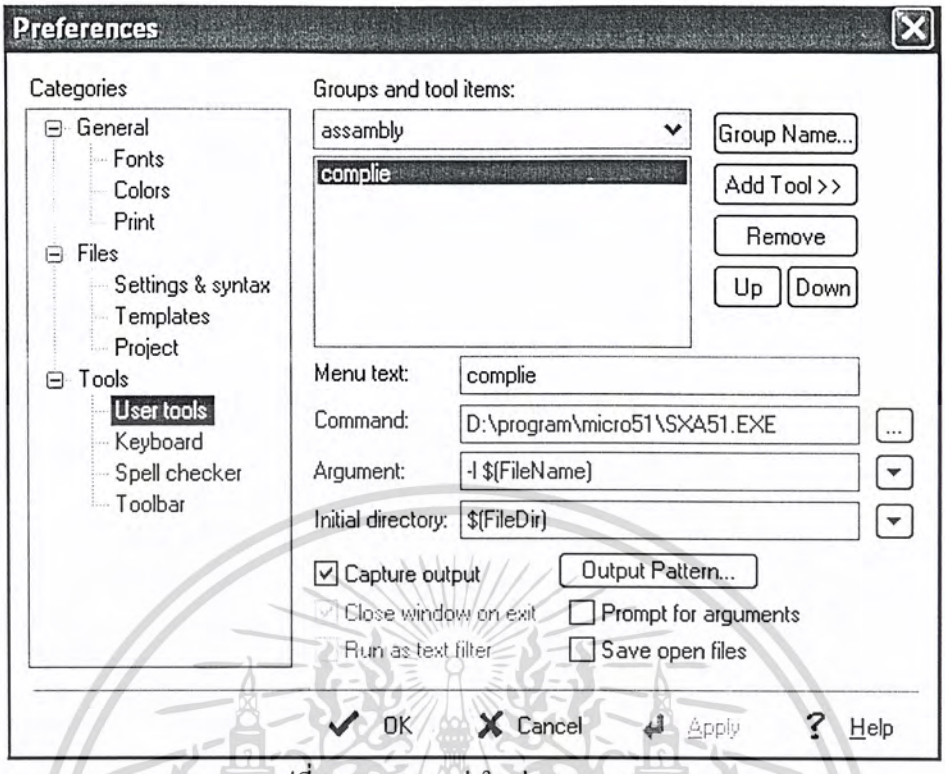
ขั้นตอนการติดตั้งและใช้งานโปรแกรม editplus

- ดาวน์โหลดโปรแกรม editplus ได้จาก www.editplus.com และทำการติดตั้ง
- เปิดโปรแกรม editplus ขึ้นมาและทำการเซตค่า ตามรูปที่ 3.10 รูปที่ 3.11และรูปที่ 3.12

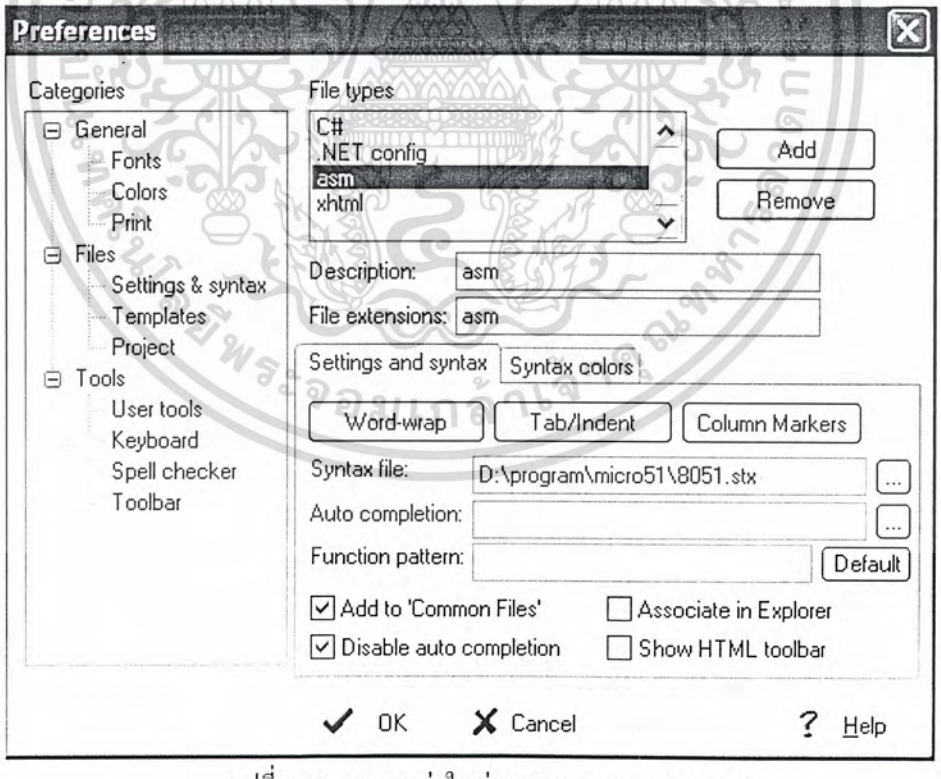


รูปที่ 3.10 การเซตค่าในส่วนของ Tools, Configure User Tools ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



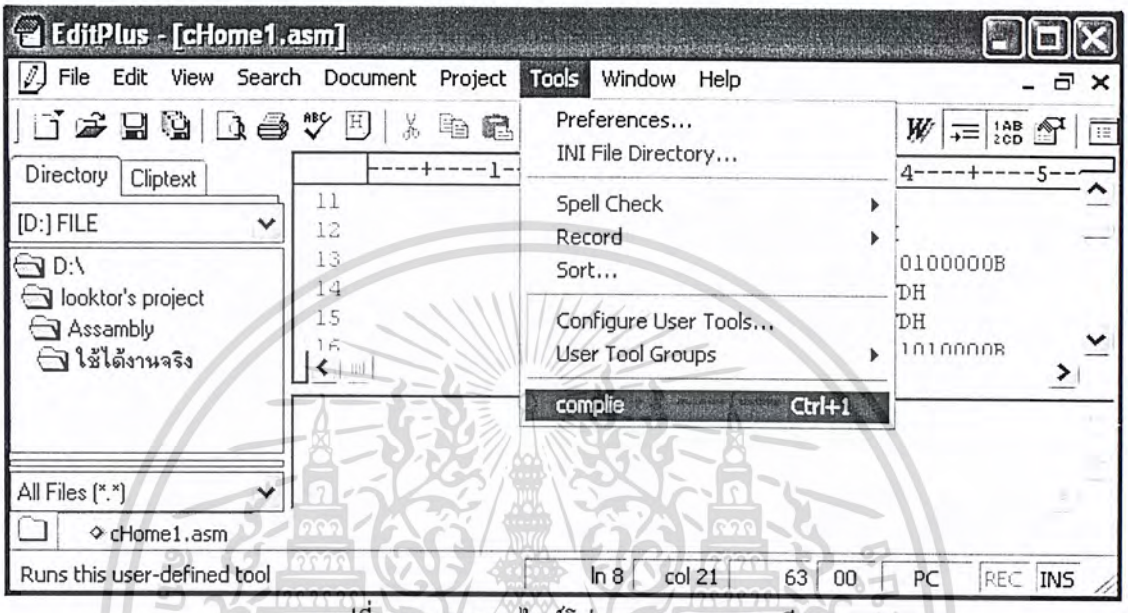
รูปที่ 3.11 การเซตค่าในส่วนของ User Tools ...



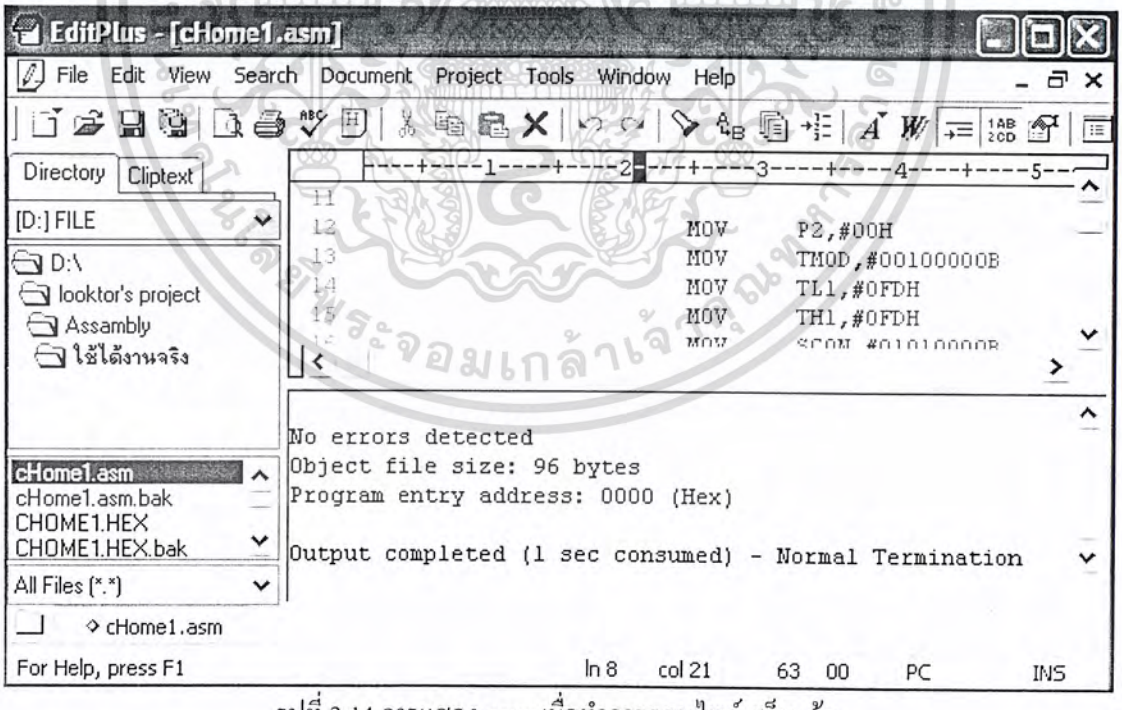
รูปที่ 3.12 การเซตค่าในส่วนของ Setting & syntax

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออกแบบการทำงานของไมโครคอนโทรลเลอร์ ทำการเขียนโปรแกรมด้วยภาษาแอสแซมบลี
- ทำการบันทึกไฟล์เป็น .asm
- ทำการคอมไฟล์โปรแกรมแอสแซมบลีโดยกด Ctrl+1 หรือ คลิกเลือก compile ดังรูปที่ 3.13
- เมื่อทำการ compile เสร็จแล้ว ด้านล่างของโปรแกรม editplus จะแสดงว่าเกิด error หรือไม่ ดังรูปที่ 3.14



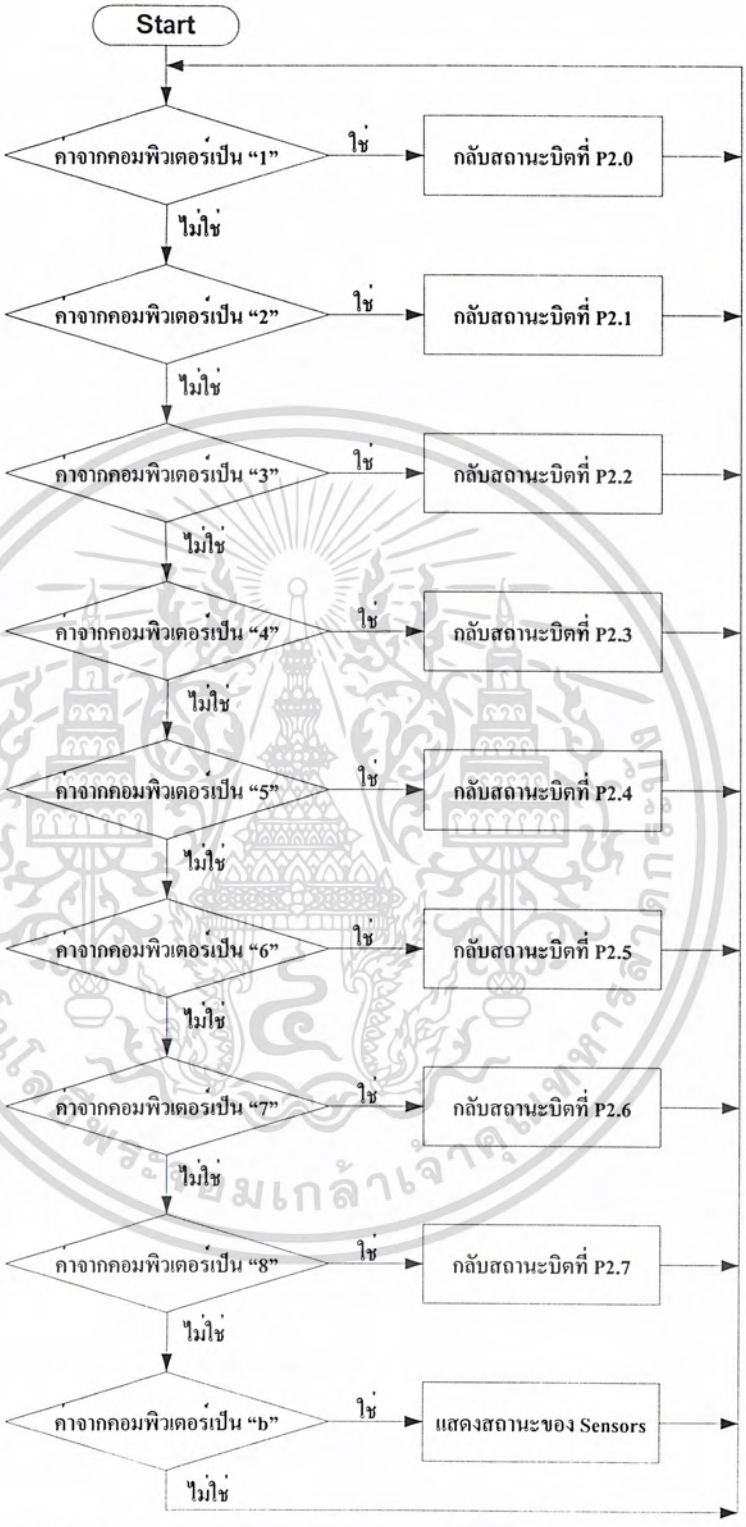
รูปที่ 3.13 การคอมไพล์โปรแกรมแอสแซมบลี



รูปที่ 3.14 การแสดง error เมื่อทำการคอมไพล์เสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์โดยใช้ภาษาแอสแซมบลี



รูปที่ 3.15 โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 ส่วนโปรแกรมเชื่อมต่อการทำงานผ่านโครงข่ายอินเตอร์เน็ตและโครงข่ายโทรศัพท์เคลื่อนที่

โดยขั้นตอนการสร้างส่วนนี้จะใช้โปรแกรมภาษาจาวา ใช้ในการสร้างเว็บเพจและสร้างแวบเพจสำหรับเป็นสื่อที่ใช้ในการควบคุมผ่านโครงข่ายอินเตอร์เน็ตและโครงข่ายโทรศัพท์เคลื่อนที่ โดยจะใช้

3.9.1 การติดตั้งและใช้งาน Tomcat เป็นเซิร์ฟเวอร์สำหรับการทดลองที่ทำงานบนระบบปฏิบัติการ WindowXP โดยวิธีการมีดังนี้

- ติดตั้งเซิร์ฟเวอร์ Tomcat5 ซึ่งสามารถดาวน์โหลดได้จาก www.javasun.com
- ติดตั้ง Tomcat5 ใน c:\Tomcat5
- เมื่อติดตั้งแล้วให้ startup tomcat แล้วทดสอบเปิดเว็บ http://localhost:8080 ถ้าเห็นคำว่า Apache Jakarta Project ก็แสดงว่าที่เปิดบริการเว็บเซิร์ฟเวอร์ได้แล้ว

3.9.2 การติดตั้งและใช้งานโปรแกรมภาษาจาวา ซึ่งอย่างน้อยจะต้องมี Java compiler และ Java interpreter เลือกใช้โปรแกรมสำหรับการติดตั้ง JDK โดยดาวน์โหลดได้จาก www.javasoft.com

- เมื่อทำการดาวน์โหลดไฟล์การติดตั้งเรียบร้อยแล้ว เปิดโฟลเดอร์ที่ดาวน์โหลดนั้นไว้แล้ว เลือกไฟล์ j2sdk-1_4_2_04
- ดับเบิ้ลคลิกที่ไฟล์ดังกล่าวรอจนกว่ามีหน้าต่างขึ้นมาแล้วดำเนินการขั้นตอนการติดตั้งต่อไปเรื่อยๆ จนสิ้นสุดการติดตั้ง
- กำหนด path เพื่อใช้โปรแกรม โดยเปิดไฟล์ Autoexec.bat ขึ้นมาแล้วเพิ่ม path เข้าไปจาก path เดิมที่มีอยู่แล้ว ดังนี้

Path = C:\windows;C:\windows\command

เพิ่ม path เข้าไปดังนี้

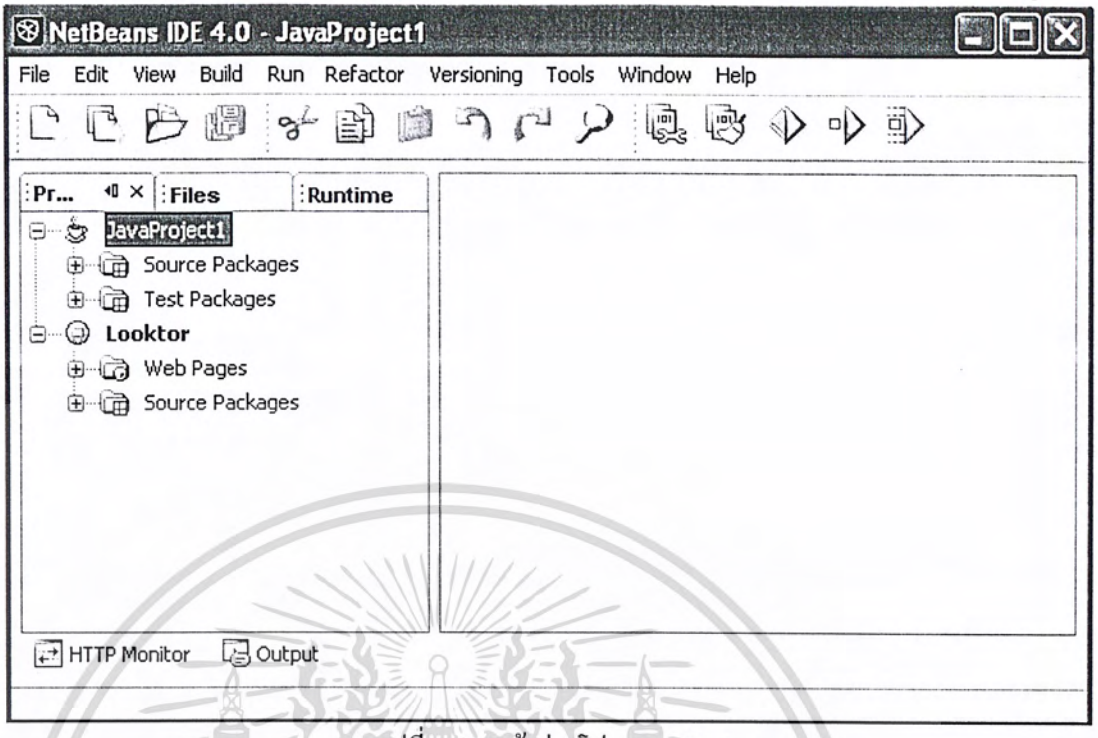
Path = C:\windows;C:\Windows\command;C:\j2sdk-1_4_2_04\bin

ติดตั้ง path เสร็จแล้วให้ทำการ Restart เครื่องคอมพิวเตอร์

3.9.3 การติดตั้งและใช้งาน Netbeans ในโครงงานนี้เลือกใช้ Netbeans IDE เพราะมี editor, file manager , debugger, compiler และ interpreter สามารถใช้งานได้ง่าย โดยดาวน์โหลดได้จาก www.javasoft.com

- เมื่อทำการดาวน์โหลดไฟล์การติดตั้งเรียบร้อยแล้ว เปิดโฟลเดอร์ที่ดาวน์โหลดนั้นไว้
- เลือกไฟล์ Netbeans IDE 4.0
- ดับเบิ้ลคลิกที่ไฟล์ดังกล่าวรอจนกว่ามีหน้าต่างขึ้นมาแล้วดำเนินการขั้นตอนการติดตั้งต่อไปเรื่อยๆ จนสิ้นสุดการติดตั้ง
- ทดสอบเปิดโปรแกรม Netbeans ขึ้นมาจะได้ดังรูปที่ 3.16

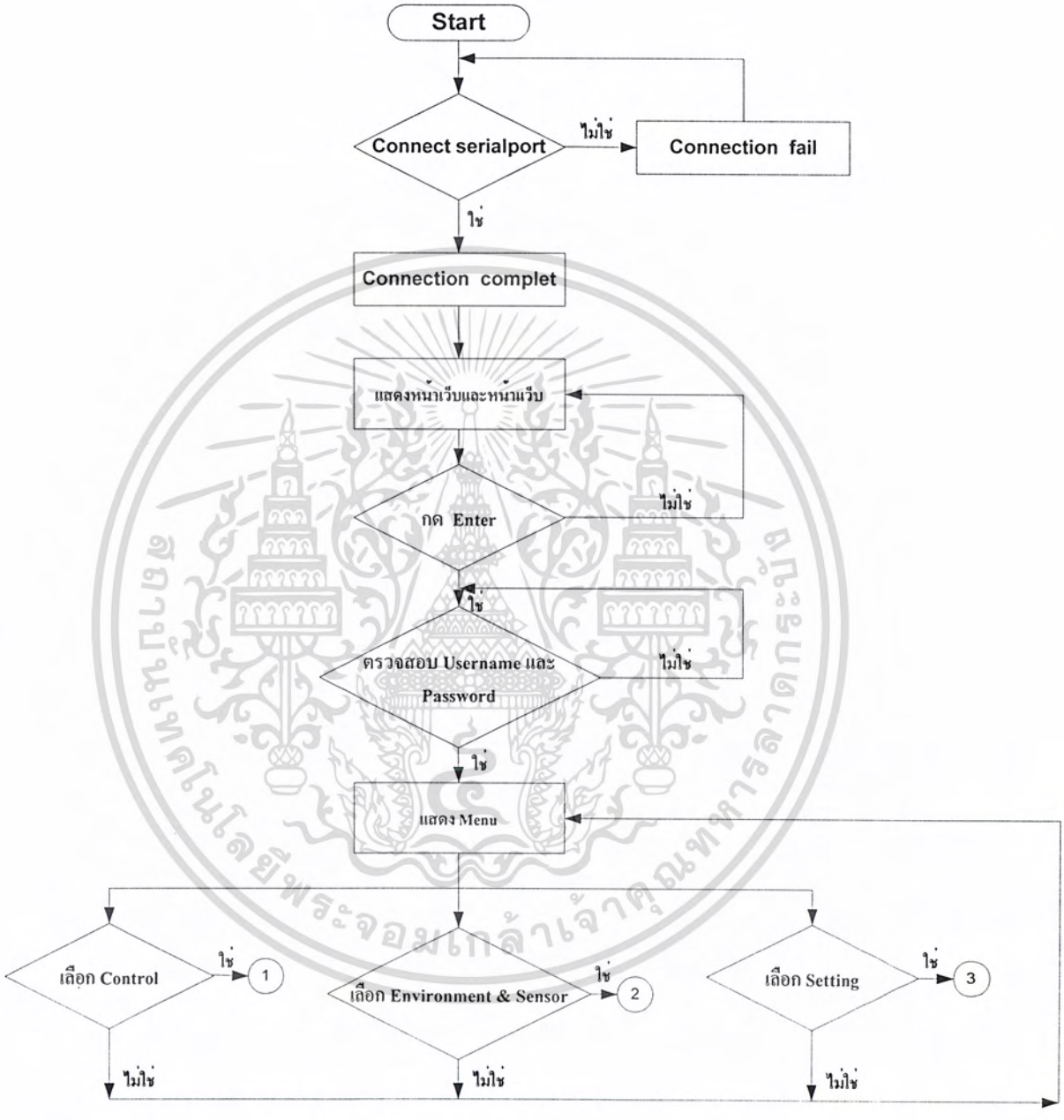
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 หน้าต่างโปรแกรม Netbeans

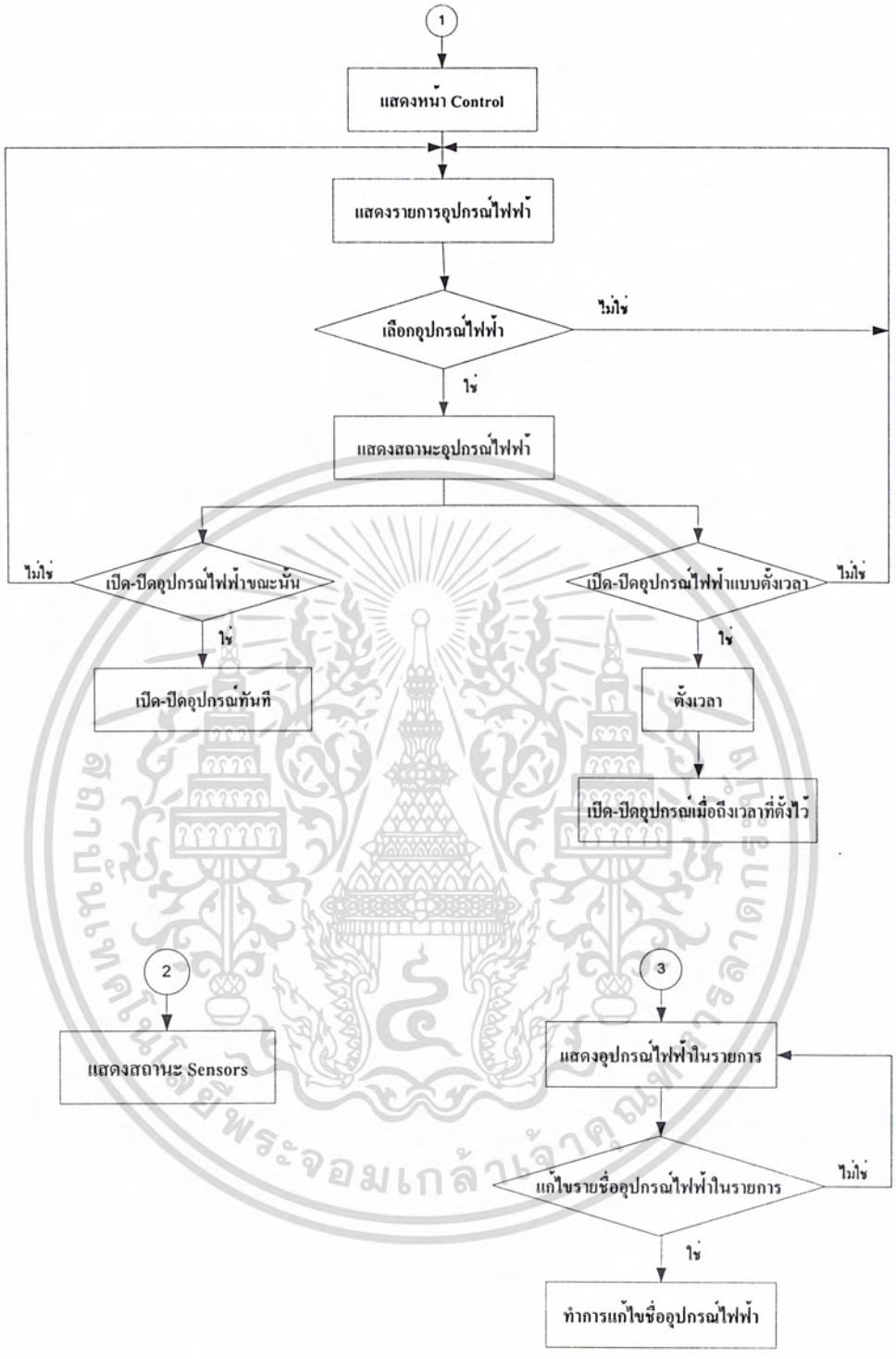
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลว์ชาร์ตโปรแกรมรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่าน internet และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา



รูปที่ 3.17 โฟลว์ชาร์ตโปรแกรมรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 โฟลว์ชาร์ตโปรแกรมรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต และโทรศัพท์เคลื่อนที่โดยใช้ภาษาจาวา(ต่อ)

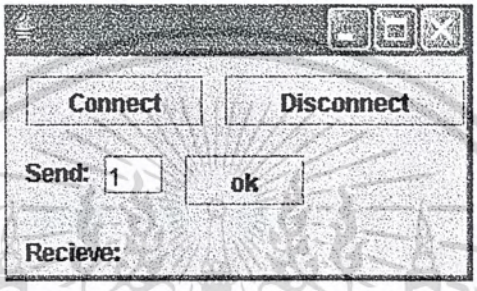
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

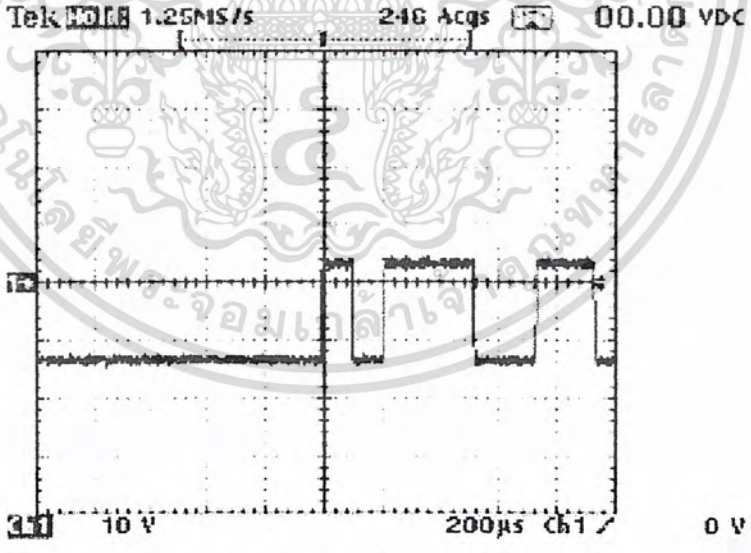
การทดลองและผลการทดลอง

4.1 การทดลองวงจรแปลงสัญญาณแบบ RS-232

ต่อวงจรดังรูปที่ 3.2 โดยใช้ค่า C = 10 μF ไอซีเบอร์ MAX 232 จะทำหน้าที่แปลงระดับสัญญาณให้เหมาะสมกับมาตรฐาน RS-232 โดยจะทำการลองส่งค่า "1" จากโปรแกรมซีเขียนขึ้นโดยภาษาจาวา ซึ่งเป็นรหัสแอสกี้จากคอมพิวเตอร์เข้ามายังขา RxD ของ ไอซี MAX 232 แสดงผลการทดลองดังรูปที่ 4.2 จากนั้นส่งค่า "1" ที่รับมานั้นกลับไปยังคอมพิวเตอร์ แล้วทำการวัดสัญญาณแสดงดังรูปที่ 4.3

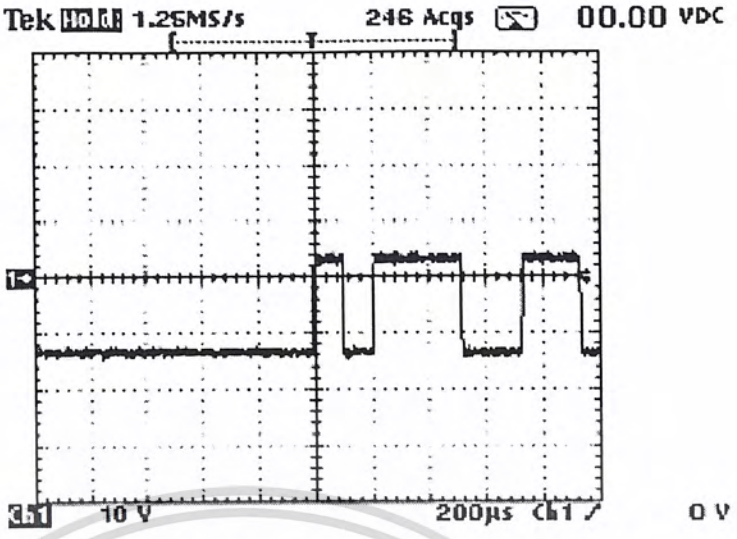


รูปที่ 4.1 หน้าต่างของโปรแกรมส่งค่า "1"



รูปที่ 4.2 สัญญาณที่รับมาจากคอมพิวเตอร์ของขา RxD ของไอซี MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

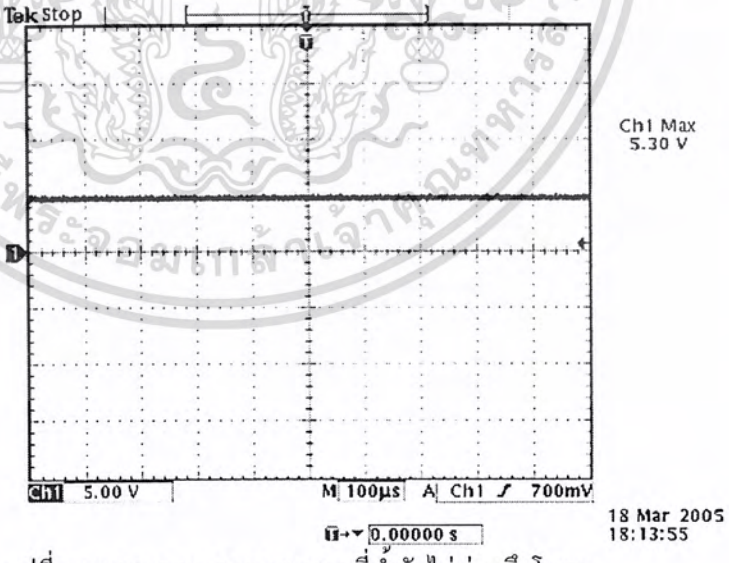


รูปที่ 4.3 สัญญาณที่ส่งกลับไปยังคอมพิวเตอร์จากขา TxD ของไอซี MAX 232

จะเห็นว่าสัญญาณที่รับเข้ามาและสัญญาณที่ส่งออกไปนั้นมีรูปร่างที่เหมือนกัน แสดงว่าการรับและส่งข้อมูลผ่าน RS-232 ถูกต้อง

4.2 วงจรเตือนน้ำเต็ม

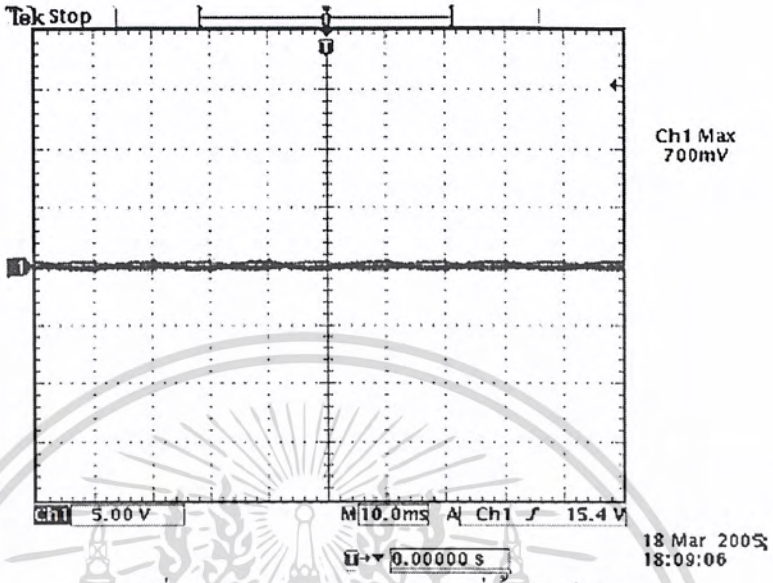
ต่อวงจรดังรูปที่ 3.5 ในสภาวะปกติขณะที่น้ำยังไม่ท่วมถึงโพรบ XY เมื่อวัดที่ขาคอลเลคเตอร์ของทรานซิสเตอร์ จะได้เอาท์พุตเท่ากับ 5 โวลต์ ดังรูปที่ 4.4



รูปที่ 4.4 แสดงผลการทดลองขณะที่น้ำยังไม่ท่วมถึงโพรบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

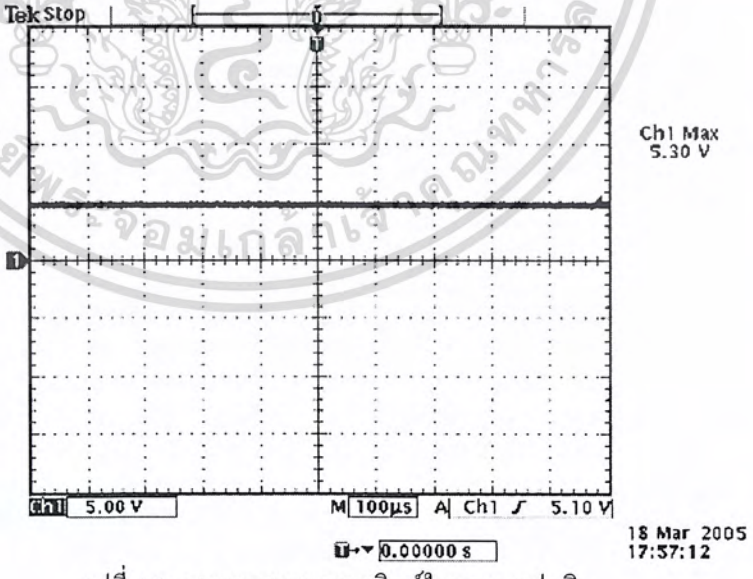
เมื่อน้ำท่วมถึงโพรบ XY เมื่อวัดที่ขาคอลเลคเตอร์ของทรานซิสเตอร์ จะได้เอาท์พุตเท่ากับ 0 โวลต์ ดังรูปที่ 4.5



รูปที่ 4.5 แสดงผลการทดลองขณะที่น้ำท่วมถึงโพรบ

4.3 วงจรสวิทช์

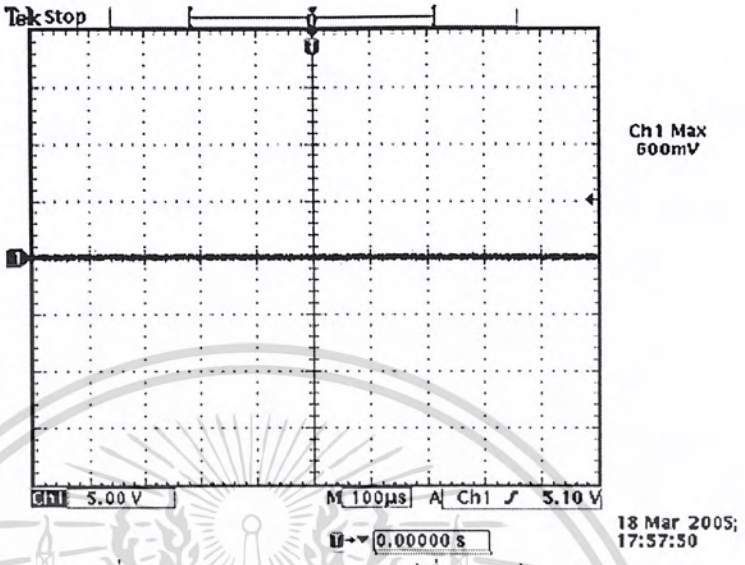
ต่อวงจรตามรูปที่ 3.6 โดยทดสอบเอาท์พุทที่ขา 3, 9, 12, 15 ของไอซี CD4042B จะได้เอาท์พุทเท่ากับ 5 โวลต์ ดังรูปที่ 4.6



รูปที่ 4.6 แสดงผลของวงจรสวิทช์ในสถานะปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

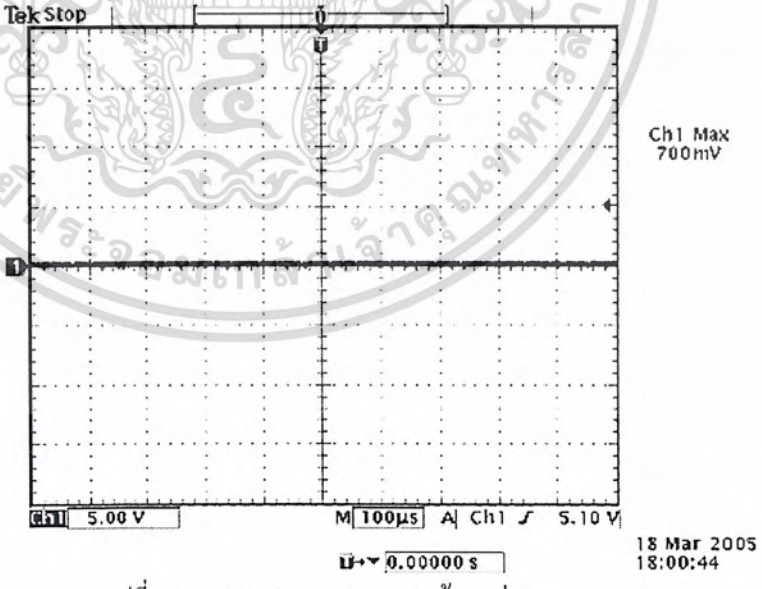
จากนั้นทดสอบเมื่อกดสวิทช์โดยทดสอบเอาท์พุทที่ขา 3, 9, 12, 15 ของไอซี CD4042B จะได้เอาท์พุทเท่ากับ 0 โวลต์ ดังรูปที่ 4.7



รูปที่ 4.7 แสดงผลของวงจรสวิทช์เมื่อสวิทช์ถูกกด

4.4 วงจรตรวจจับทางแสง

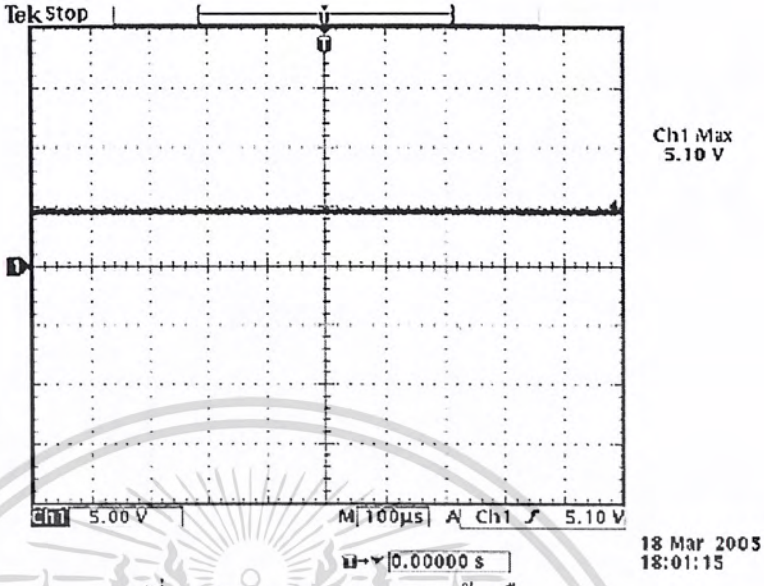
ต่อวงจรดังรูปที่ 3.7 ในสภาวะปกติในห้องที่สว่างจะได้เอาท์พุตที่ขา 7 ของไอซี LM 741 จะได้เอาท์พุตเท่ากับ 0.7 โวลต์ ดังรูปที่ 4.8



รูปที่ 4.8 แสดงผลของวงจรขณะห้องสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

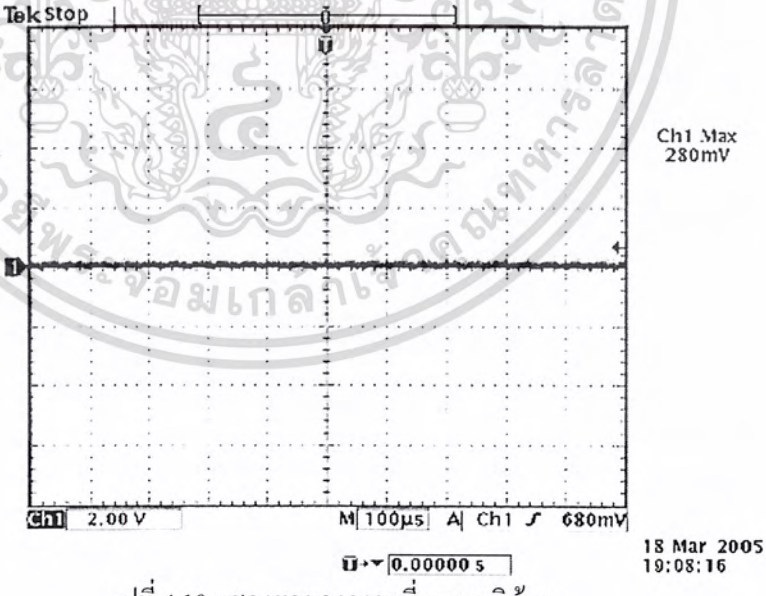
เมื่อห้องมืดลงจะได้เอาท์พุตที่ขา 7 ของไอซี LM 741 จะได้เอาท์พุตเท่ากับ 5 โวลต์ ดังรูปที่ 4.9



รูปที่ 4.9 แสดงผลของวงจรขณะห้องมืด

4.5 วงจรควบคุมอุณหภูมิ

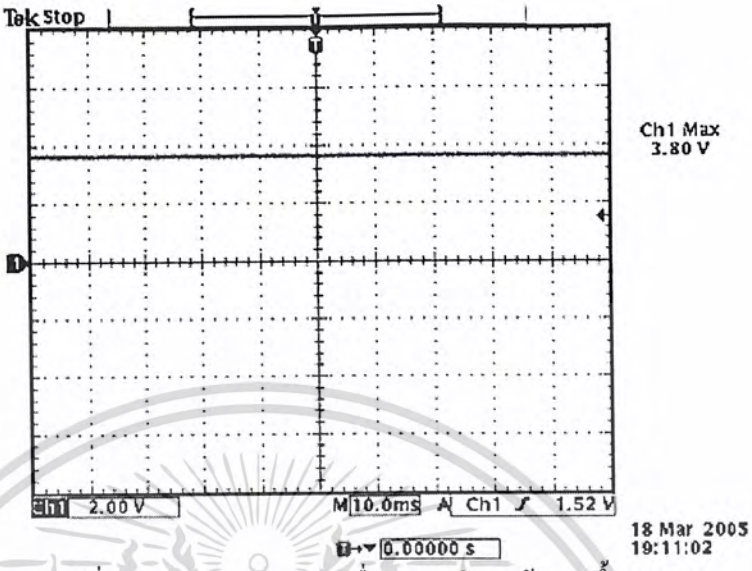
ต่อวงจรดังรูปที่ 3.9 ขณะที่อุณหภูมิห้องวัดที่ขา 6 ไอซี LM741 จะได้เอาท์พุตเท่ากับ 0โวลต์ ดังรูปที่ 4.10



รูปที่ 4.10 แสดงผลของวงจรที่อุณหภูมิห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

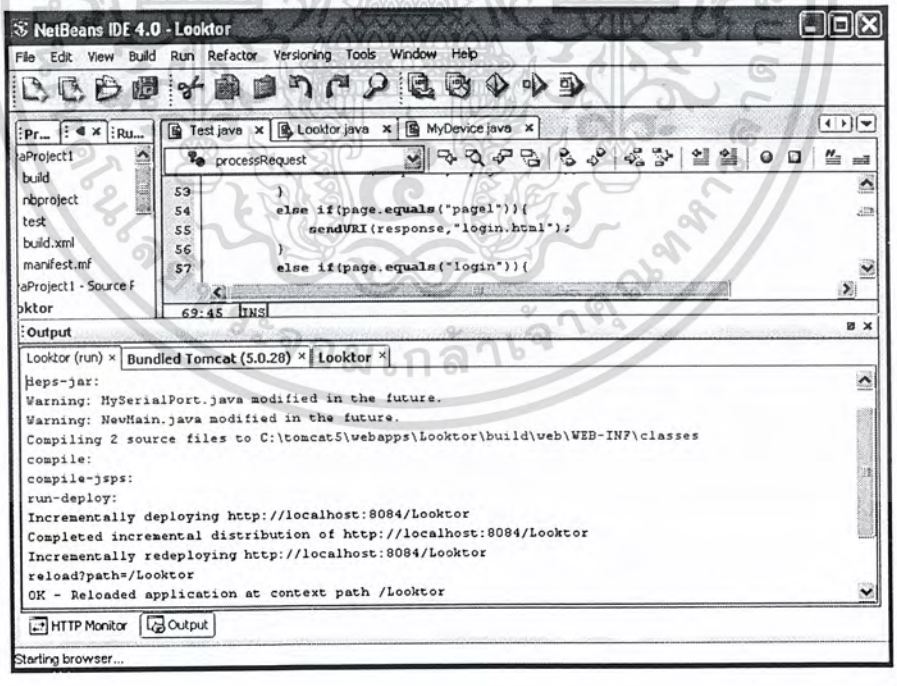
ทดสอบวงจรควบคุมอุณหภูมิโดยใช้หัวแร้งจ่อที่เทอร์มิสเตอร์ไว้ประมาณ 3 นาที จะได้เอาท์พุต เท่ากับ 3.80 โวลต์ ดังรูปที่ 4.11



รูปที่ 4.11 แสดงผลของวงจรเมื่ออุณหภูมิของห้องสูงขึ้น

4.6 การใช้งานระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต

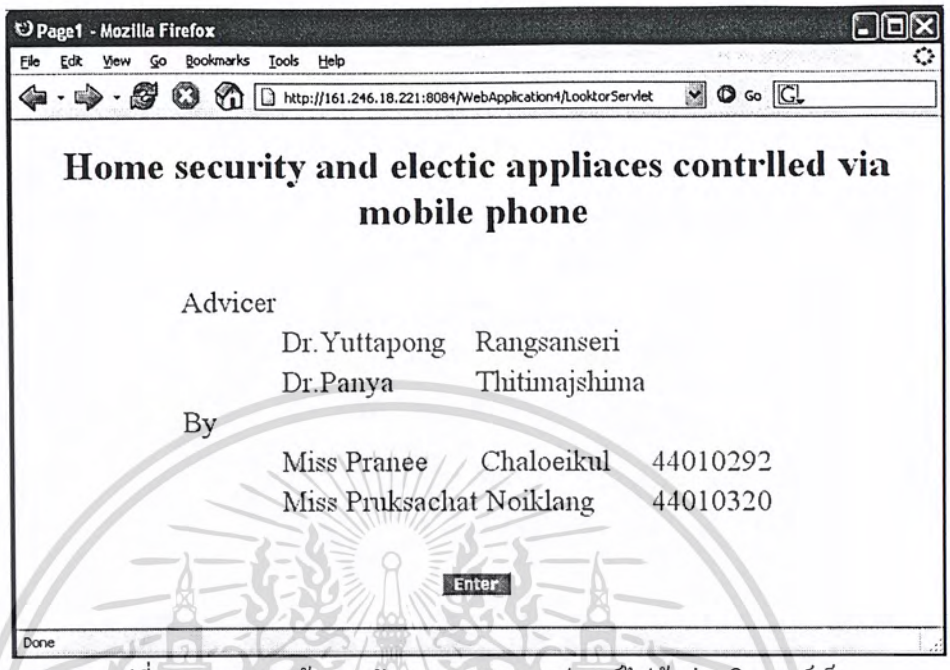
รันโปรแกรม LooktorServlet.java โดยการกด shift+F6 เพื่อจำลองคอมพิวเตอร์ให้เป็นคอมพิวเตอร์เซิร์ฟเวอร์ จะได้ผลของการจำลองคอมพิวเตอร์เซิร์ฟเวอร์ว่าเสร็จสมบูรณ์ ซึ่งจะแสดงข้อความดังรูปที่ 4.12



รูปที่ 4.12 การจำลองคอมพิวเตอร์เซิร์ฟเวอร์

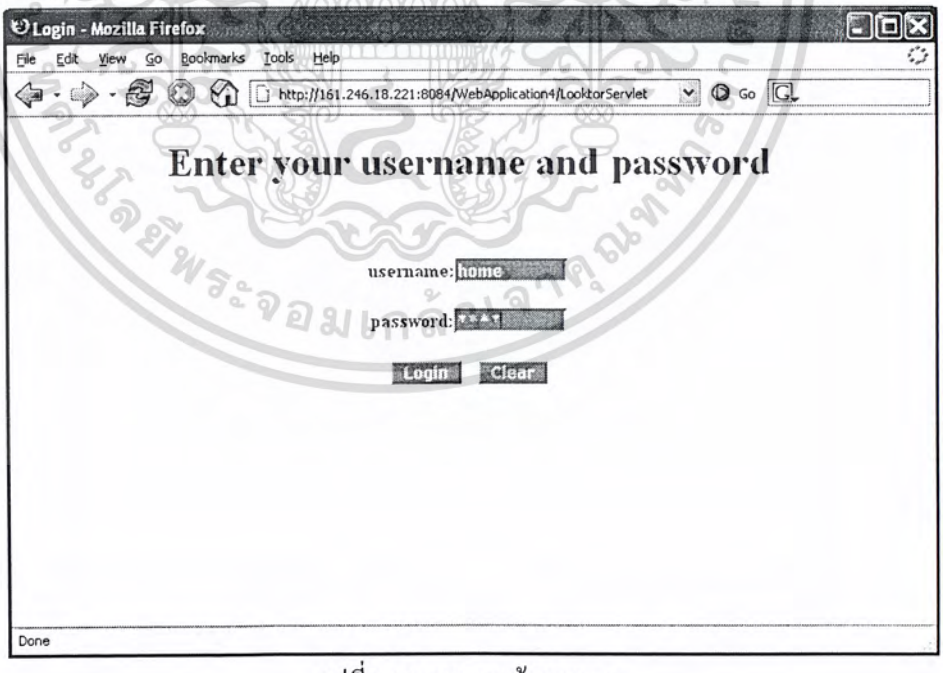
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รันโปรแกรมเว็บเบราเซอร์ แล้วเรียกที่ http://161.246.18.221:8084/WebApplication4/LooktorServlet ดังรูปที่ 4.13



รูปที่ 4.13 แสดงหน้าจอหลักระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต

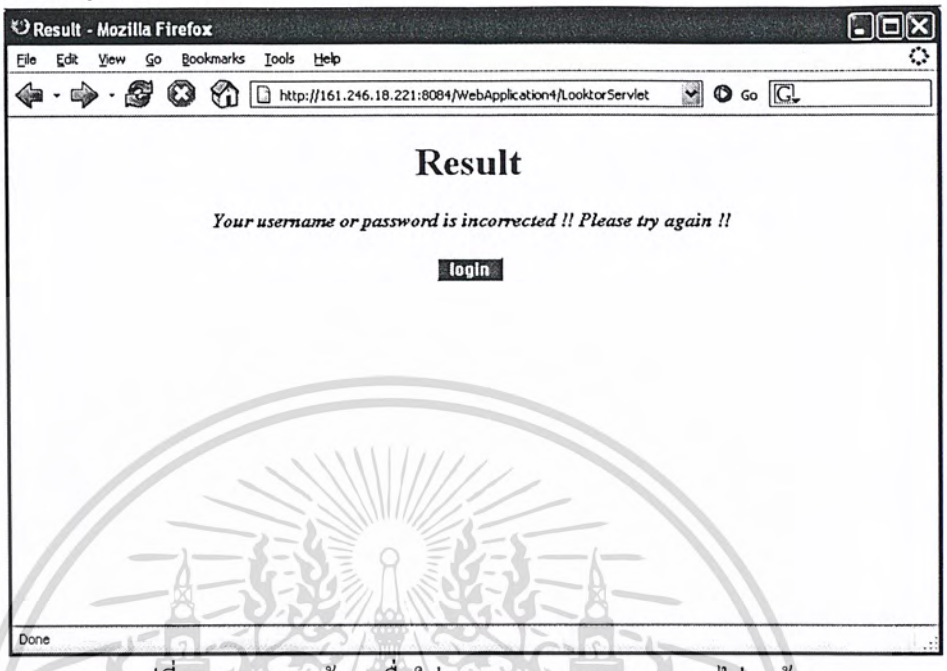
คลิกที่ Enter จะเข้าสู่หน้าที่ต้องป้อน Username และ Password ซึ่งแสดงดังรูปที่ 4.14



รูปที่ 4.14 แสดงหน้าจอ Login

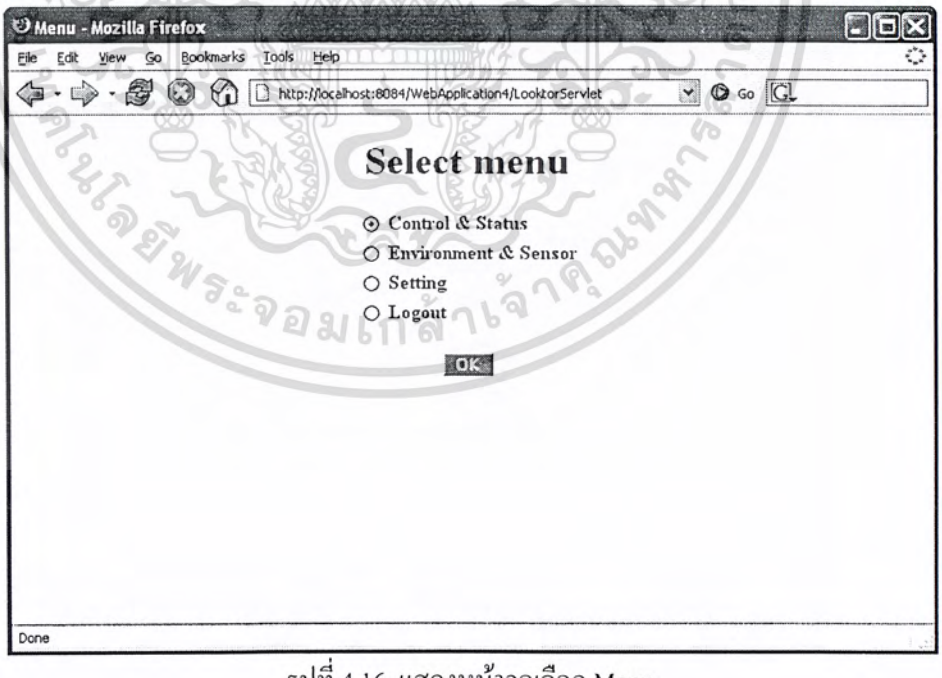
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ต้องใส่ Username และ Password ให้ถูกต้องก่อน ถ้าใส่ Username และ Password ไม่ถูกต้อง หน้าจอจะแสดงผลดังรูปที่ 4.15



รูปที่ 4.15 แสดงหน้าจอเมื่อใส่ Username และ Password ไม่ถูกต้อง

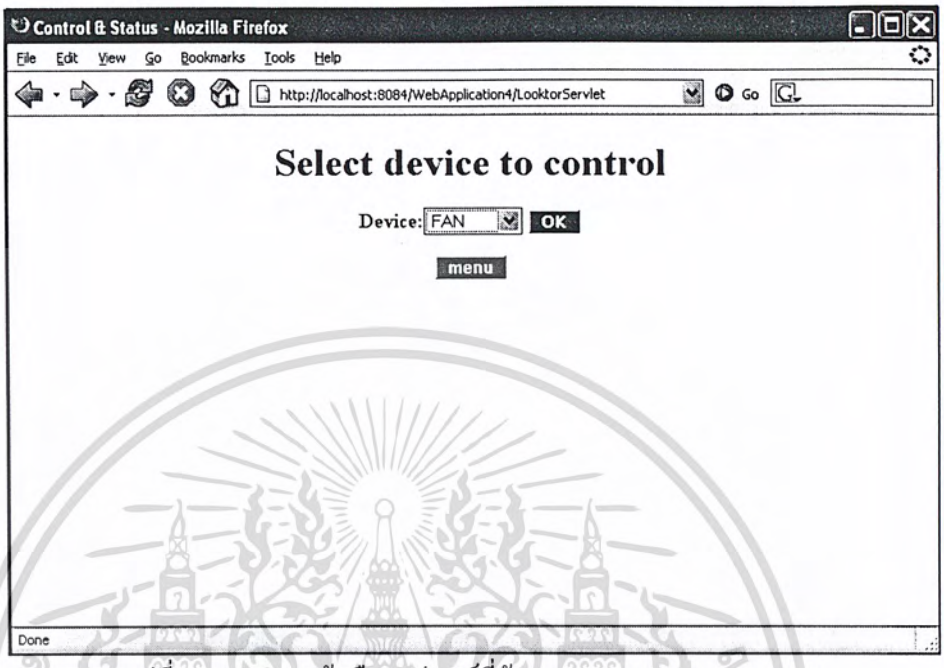
เมื่อใส่ Username และ Password ถูกต้องที่หน้าจอของผู้ใช้จะแสดงหน้า Menu ดังรูปที่ 4.16



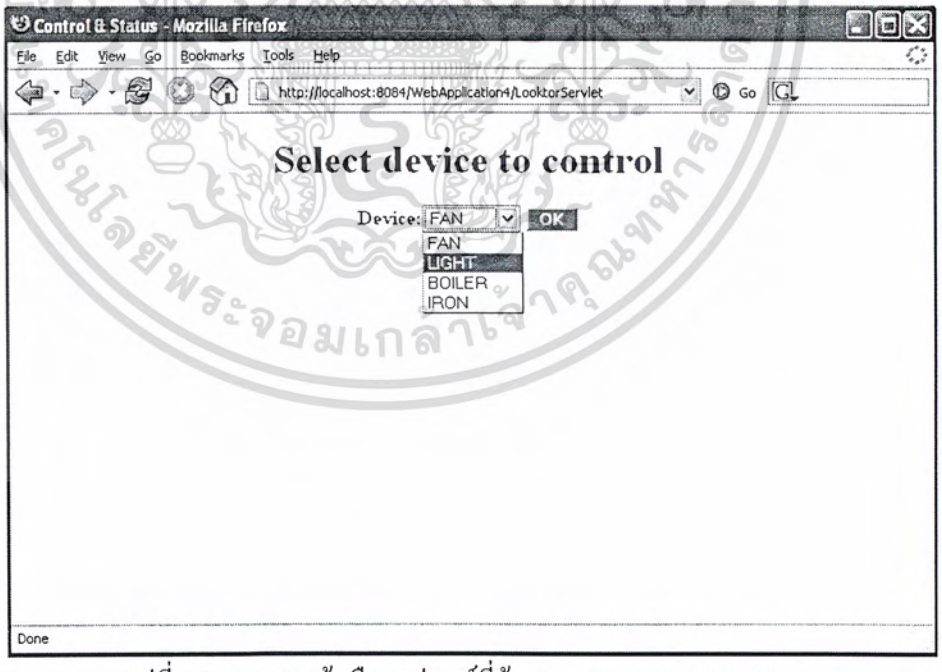
รูปที่ 4.16 แสดงหน้าจอเลือก Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกหน้า Control & Status แล้วคลิก OK จะแสดงหน้าให้เลือกชื่ออุปกรณ์ที่จะควบคุมและตรวจสอบสถานะดังรูปที่ 4.17 และรูปที่ 4.18



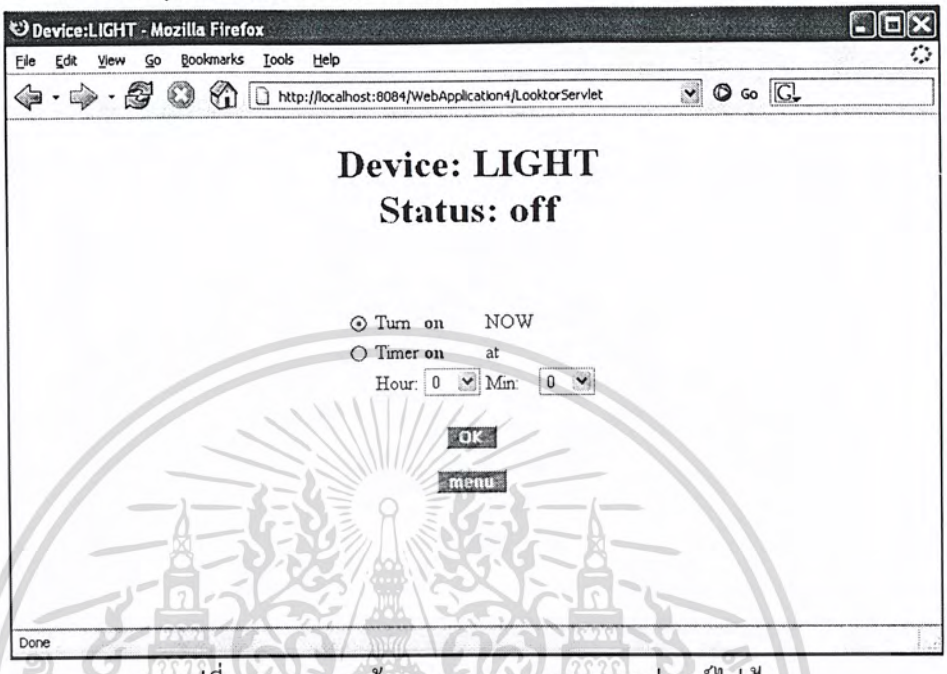
รูปที่ 4.17 แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล



รูปที่ 4.18 แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล

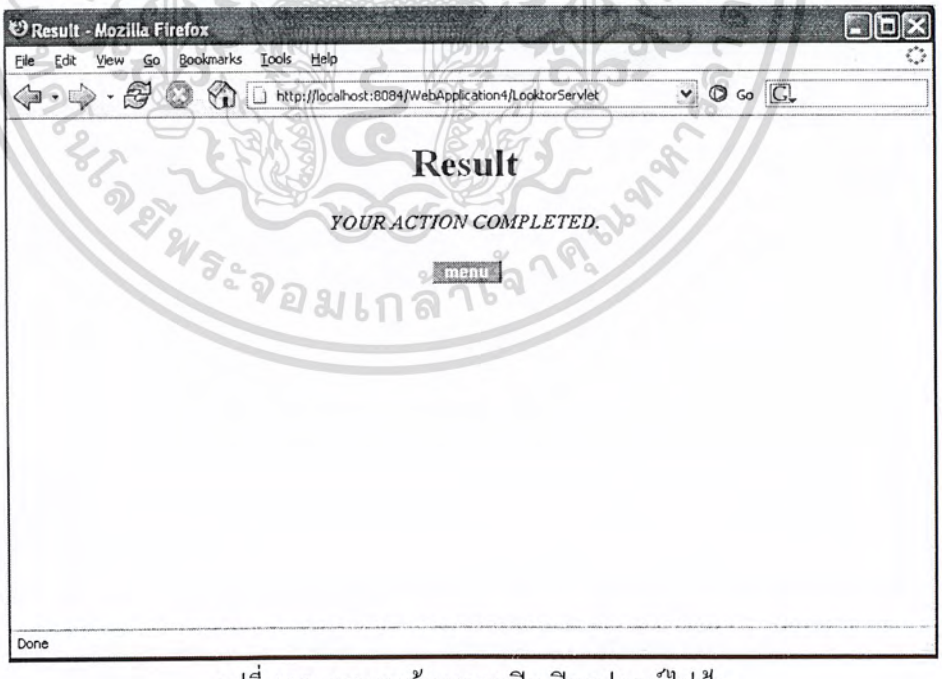
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเลือกอุปกรณ์ไฟฟ้าแล้วจะขึ้นหน้าแสดงชื่อและสถานะของอุปกรณ์ไฟฟ้าตัวนั้น และสามารถเลือกที่จะควบคุมเปิดปิดอุปกรณ์ไฟฟ้าได้สองแบบคือ แบบเปิด-ปิดอุปกรณ์ทันที และแบบตั้งเวลาเปิด-ปิด ดังรูปที่ 4.19 พบว่าอุปกรณ์ไฟฟ้าชื่อ LIGHT มีสถานะoff



รูปที่ 4.19 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้า

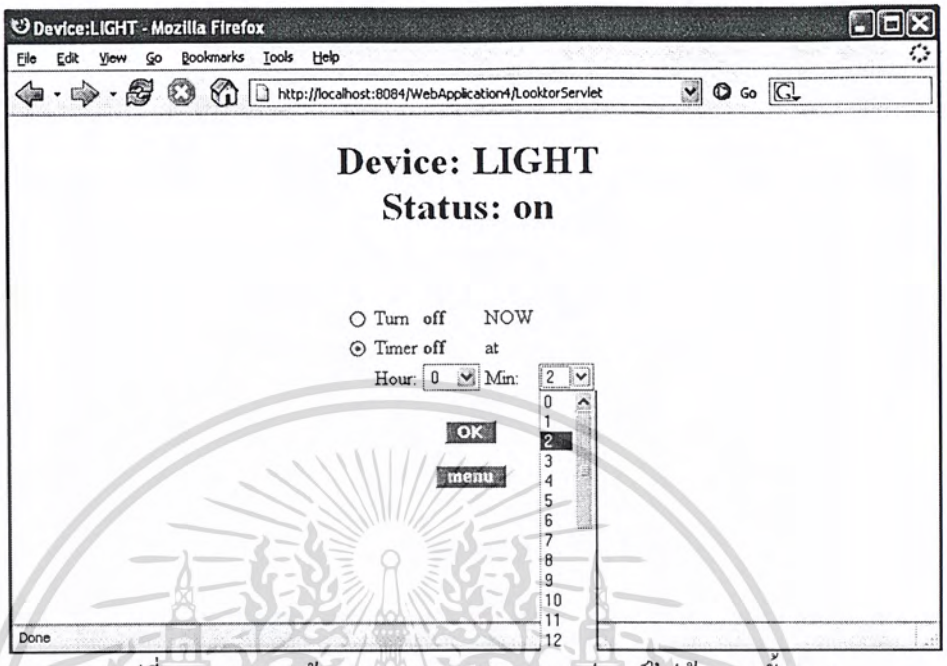
เลือกการควบคุมแบบเปิด-ปิดอุปกรณ์ไฟฟ้านั้นทันที แล้วคลิก OK จะแสดงหน้าผลการทำงานดังรูปที่ 4.20



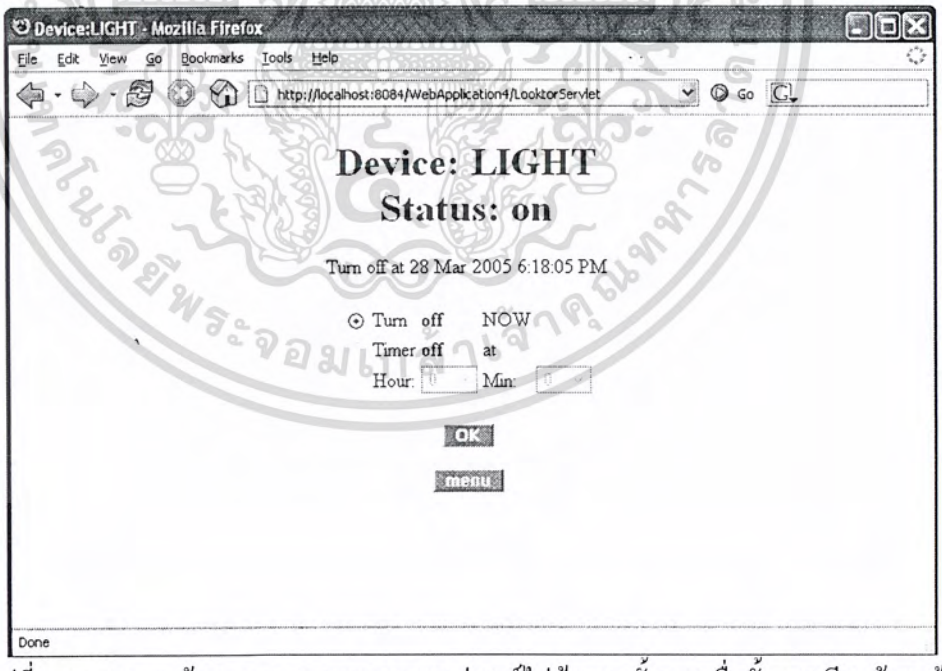
รูปที่ 4.20 แสดงหน้าผลการเปิด-ปิดอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกการทำงานแบบตั้งเวลา จะเป็นการตั้งเวลาแบบนับถอยหลัง แสดงดังรูปที่ 4.21



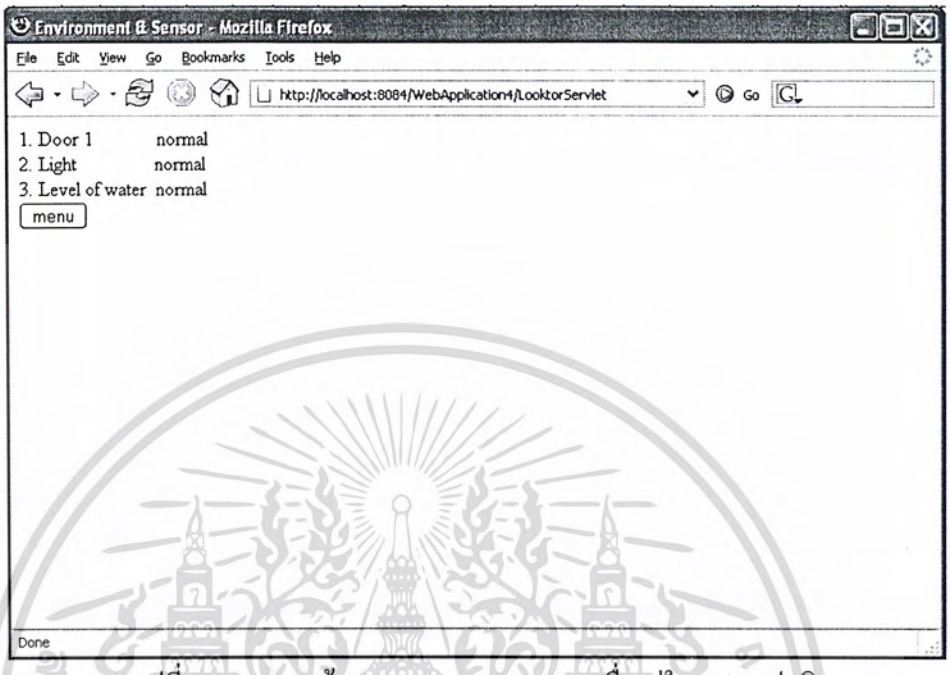
รูปที่ 4.21 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลา



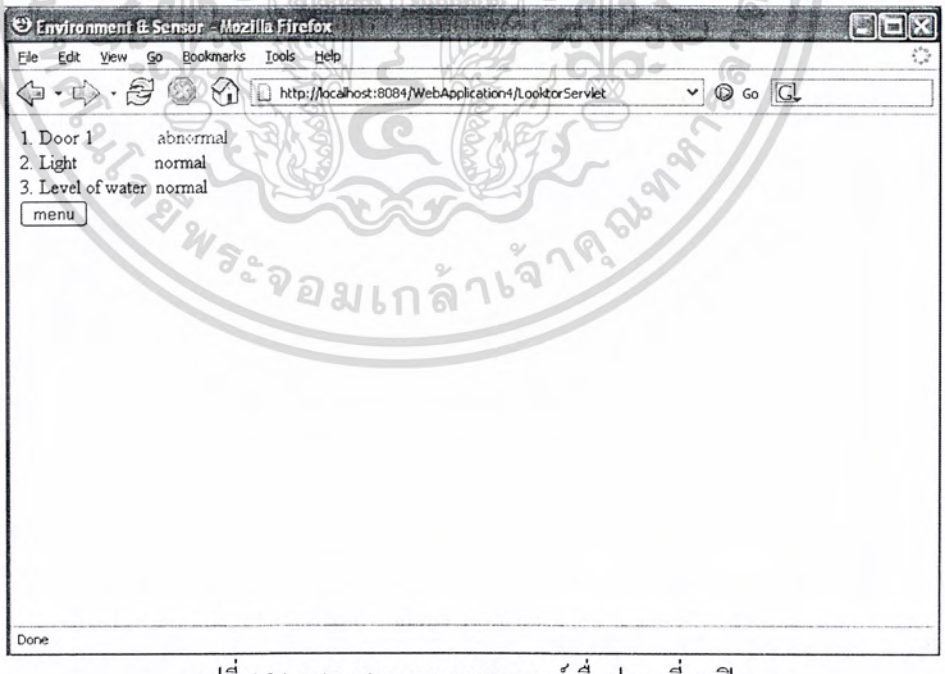
รูปที่ 4.22 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลาเมื่อตั้งเวลาเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้า Menu เลือก Environment & Sensor เพื่อทำการดูสถานะของเซนเซอร์ จากนั้นกด ok จะแสดงสถานะของเซนเซอร์ ดังรูปที่ 4.23 และ 4.24



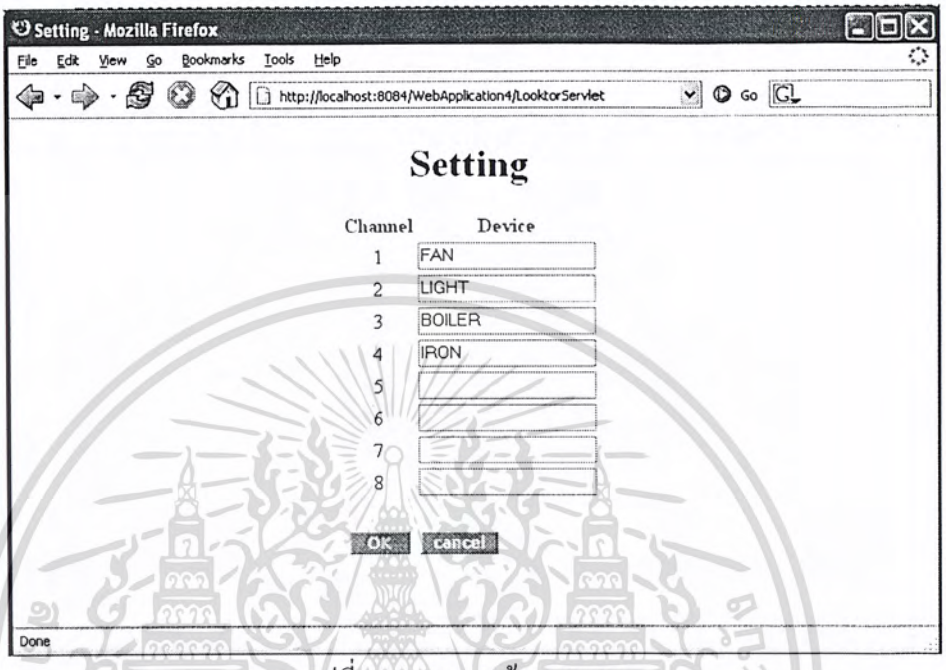
รูปที่ 4.23 แสดงหน้า Environment & Sensor เมื่ออยู่ในสภาวะปกติ



รูปที่ 4.24 แสดงสถานะของเซนเซอร์เมื่อประตูที่ 1 เปิด

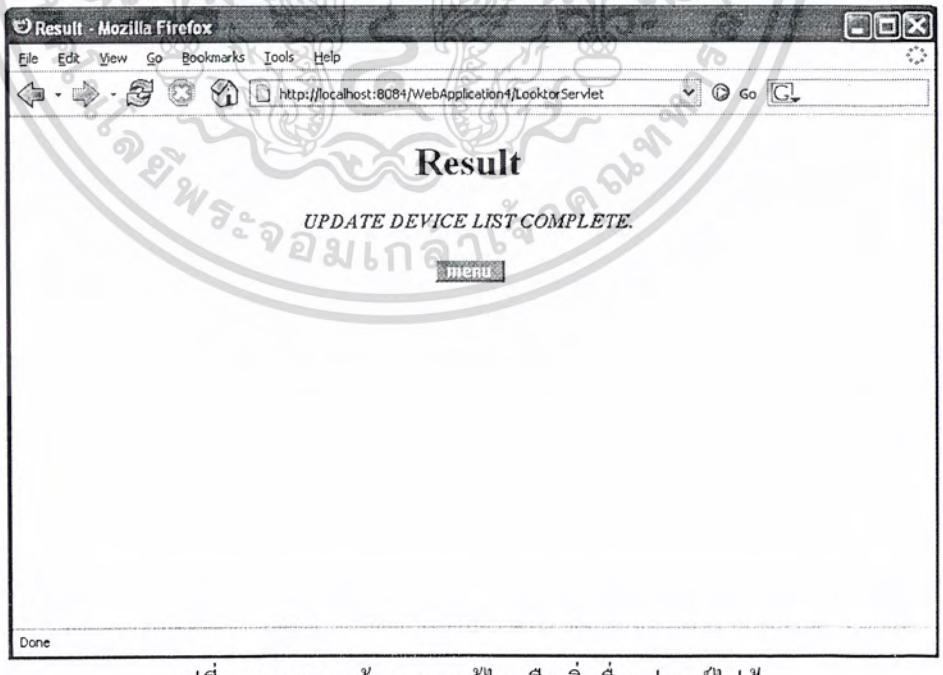
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเลือก Setting จากในหน้า Menu เพื่อทำการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า สามารถแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าของแต่ละช่อง แล้วคลิก OK เพื่อยืนยันการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า ดังรูปที่ 4.25



รูปที่ 4.25 แสดงหน้าจอ Setting

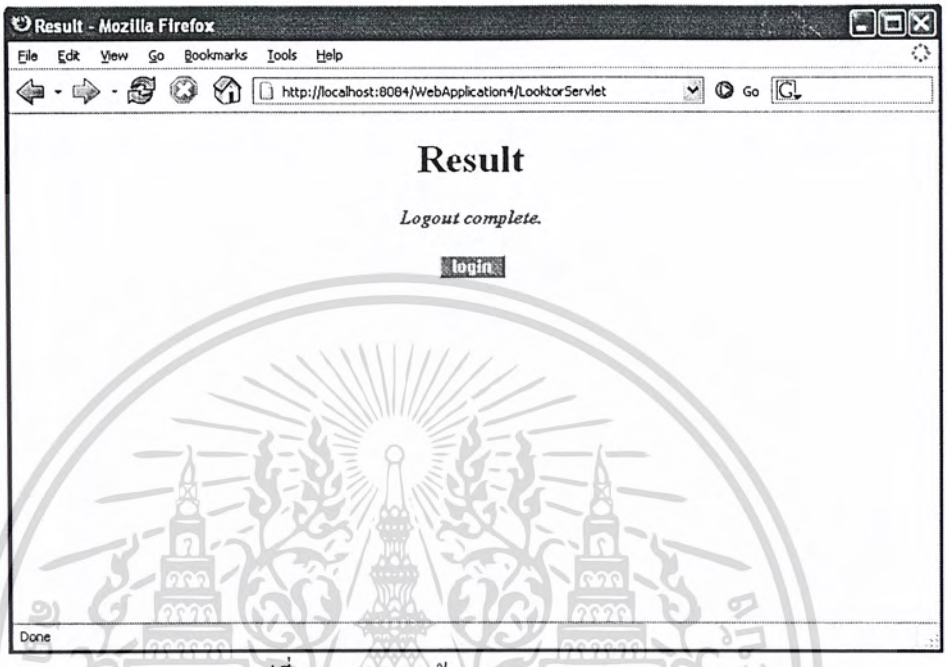
จะขึ้นหน้าแสดงผลการอัพเดตข้อมูลว่าทำการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าเรียบร้อย ดังรูปที่ 4.26



รูปที่ 4.26 แสดงหน้าผลการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ตนั้น ณ เวลาหนึ่งจะอนุญาตให้ผู้ใช้เข้ามาใช้ระบบได้แค่ เพียงคนเดียว เมื่อผู้ใช้ต้องการออกจากระบบจะต้องทำการ Logout ที่หน้าMenu เมื่อกด ok จะได้ผลดังรูปที่ 4.27



รูปที่ 4.27 แสดงหน้าผลการ Logout

4.7 การใช้งานระบบรักษาความปลอดภัยและควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่

ทดลองใช้งานกับโทรศัพท์เคลื่อนที่ที่สามารถแวปได้ โดยโครงงานนี้ได้ทดลองใช้กับโทรศัพท์เคลื่อนที่รุ่น โนเกีย 6600 ซึ่งหน้าจอของโทรศัพท์เคลื่อนที่จะแสดงดังรูปที่ 4.28 และทำการเลือกโปรแกรมโอเปร่า(Opera)ที่สามารถใช้งานแวปได้ ดังรูปที่ 4.29 ซึ่งโปรแกรมจะมีหน้าตาแสดงดังรูปที่ 4.30 และทำการลิงค์ไปยัง ที่ http://161.246.18.221:8084/WebApplication4/LooktorServlet ดังรูปที่ 4.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 แสดงหน้าจอของโทรศัพท์เคลื่อนที่

รูปที่ 4.29 แสดงเมนูของโปรแกรมต่างๆ

รูปที่ 4.30 แสดงหน้าจอของโปรแกรมโอเปร่า

รูปที่ 4.31 แสดงหน้าการลิงค์ไปยังแวปเพจ

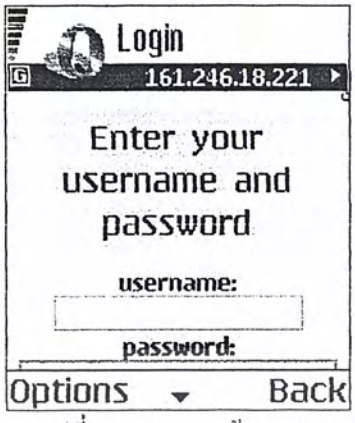
เมื่อทำการเรียกไปยัง ที่ http://161.246.18.221:8084/WebApplication4/LooktorServlet สำเร็จแล้วจะปรากฏหน้าแวปเพจดังรูปที่ 4.32

รูปที่ 4.32(ก)แสดงหน้าหลักของแวปเพจ

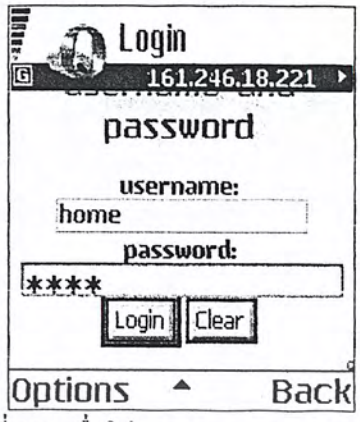
รูปที่ 4.32(ข) แสดงหน้าหลักของแวปเพจ

กดปุ่ม Enter จะเข้าสู่หน้าที่ต้องป้อน Username และ Password ซึ่งแสดงดังรูปที่ 4.33 ทำการป้อน Username และ Password แล้วกดที่ปุ่ม Login ดังรูปที่ 4.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 แสดงหน้า Login



รูปที่ 4.34 เมื่อใส่ Username และ Password

ผู้ใช้ต้องใส่ Username และ Password ให้ถูกต้องก่อน ถ้าใส่ Username และ Password ไม่ถูกต้อง หน้าจอจะแสดงผลดังรูปที่ 4.35 และเมื่อใส่ Username และ Password ถูกต้องที่หน้าจอของผู้ใช้จะแสดงหน้า Menu ดังรูปที่ 4.36

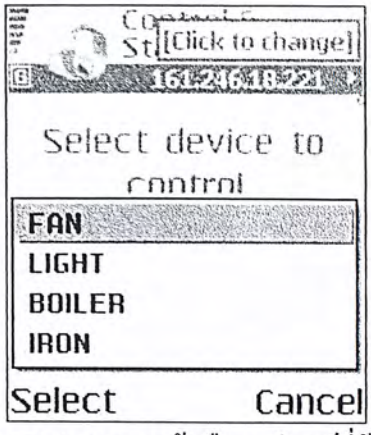
รูปที่ 4.35 แสดงหน้าเมื่อใส่ Username และ Password ไม่ถูกต้อง

รูปที่ 4.36 แสดงหน้าเลือก Menu

ในหน้า menu เลือก Control & Status แล้วคลิก OK จะแสดงหน้าให้เลือกชื่ออุปกรณ์ที่จะควบคุมและตรวจสอบสถานะดังรูปที่ 4.37(ก) และรูปที่ 4.37(ข)

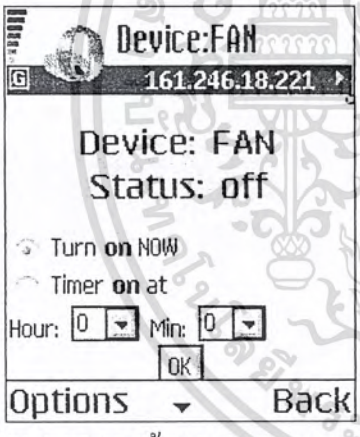
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.37(ก) แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล

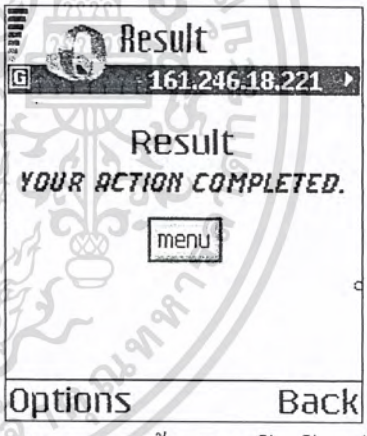


รูปที่ 4.37(ข) แสดงหน้าเลือกอุปกรณ์ที่ต้องการจะควบคุมและแสดงผล

เมื่อทำการเลือกอุปกรณ์ไฟฟ้าแล้วจะขึ้นหน้าแสดงชื่อและสถานะของอุปกรณ์ไฟฟ้าตัวนั้น และสามารถเลือกที่จะควบคุมเปิดปิดอุปกรณ์ไฟฟ้าได้สองแบบคือ แบบเปิด-ปิดอุปกรณ์ทันที และแบบตั้งเวลาเปิด-ปิด ดังรูปที่ 4.38 พบว่าอุปกรณ์ไฟฟ้าชื่อ FAN มีสถานะoff เลือกการควบคุมแบบเปิด-ปิดอุปกรณ์ไฟฟ้านั้นทันที แล้วคลิก OK จะแสดงหน้าผลการทำงานดังรูปที่ 4.39



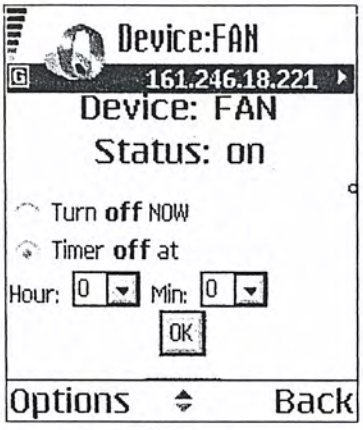
รูปที่ 4.38 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้า



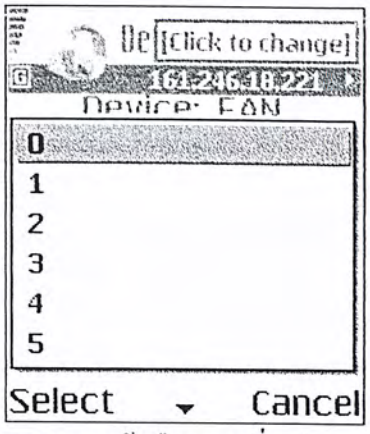
รูปที่ 4.39 แสดงหน้าผลการเปิด-ปิดอุปกรณ์ไฟฟ้า

เลือกการทำงานแบบตั้งเวลา จะเป็นการตั้งเวลาแบบนับถอยหลัง แสดงดังรูปที่ 4.40 และทำการเลือกเวลาที่จะทำการปิด-เปิด ดังรูปที่ 4.41 และจะแสดงเวลาที่จะปิดเปิดดังรูปที่ 4.42 และเมื่อกดปุ่ม OK จะแสดงหน้าผลของการตั้งเวลาปิด-เปิดอุปกรณ์ไฟฟ้าดังรูปที่ 4.43

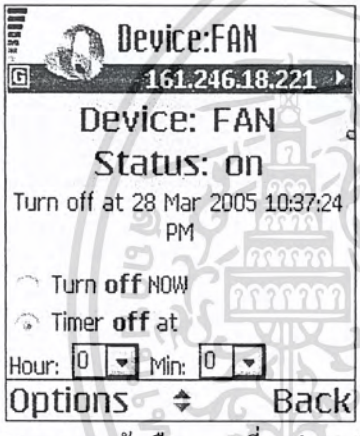
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



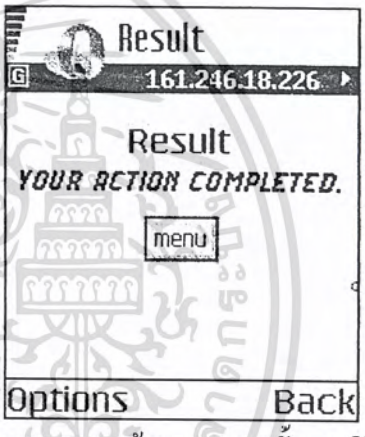
รูปที่ 4.40 แสดงหน้าสถานะและการควบคุมอุปกรณ์ไฟฟ้าแบบตั้งเวลา



รูปที่ 4.41 แสดงหน้าเลือกเวลาที่จะทำการปิด-เปิดอุปกรณ์ไฟฟ้า



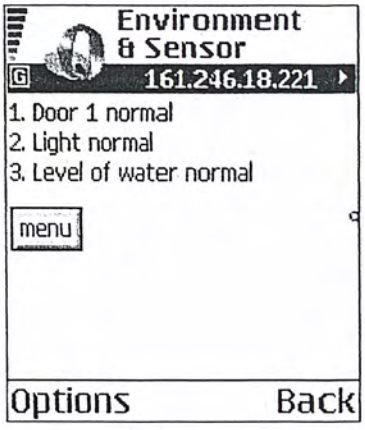
รูปที่ 4.42 แสดงหน้าเลือกเวลาที่จะทำการปิด-เปิดอุปกรณ์ไฟฟ้าเรียบร้อยแล้ว



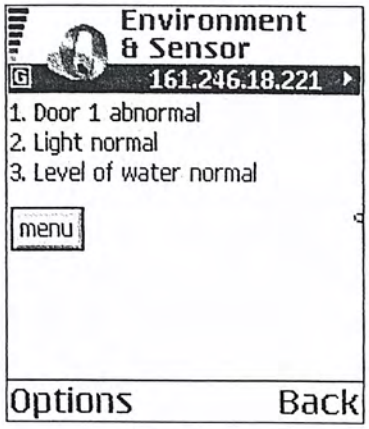
รูปที่ 4.43 แสดงหน้าผลของการตั้งเวลาปิด-เปิดอุปกรณ์ไฟฟ้า

ต่อมาเลือกที่ Environment & Sensor ในหน้า menu แล้วกด ok จะเข้าไปยังหน้า Environment & Sensor เพื่อตรวจสอบสถานะของเซนเซอร์ ในรูปที่ 4.44 จะพบว่าสถานะของเซนเซอร์ปกติ ส่วนรูปที่ 4.45 พบว่า ประตูที่ 1 เปิดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



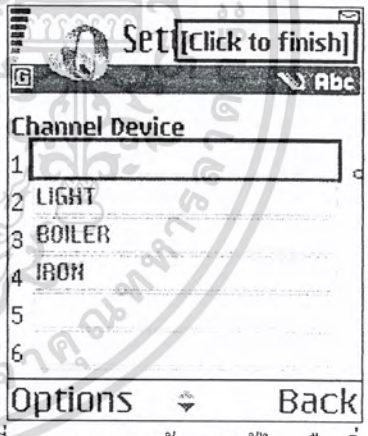
รูปที่ 4.44 แสดงหน้า Environment & Sensor เมื่ออยู่ในสถานะปกติ



รูปที่ 4.45 แสดงหน้า Environment & Sensor เมื่อประตูที่ 1 เปิด

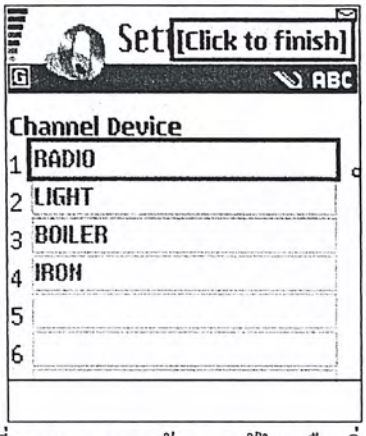
ในหน้า menu เมื่อเลือก Setting จะแสดงชื่อของอุปกรณ์ที่ต่ออยู่กับแต่ละช่องดังรูปที่ 4.46 และสามารถแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าของแต่ละช่องดังรูปที่ 4.47 และเมื่อทำการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าเสร็จแล้วดังรูปที่ 4.48 คลิก OK เพื่อยืนยันการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า จะได้ผลเป็นดังรูปที่ 4.49

รูปที่ 4.46 แสดงหน้า Setting



รูปที่ 4.47 แสดงหน้าการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.48 แสดงหน้าการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้าเรียบร้อยแล้ว

รูปที่ 4.49 แสดงหน้าผลการแก้ไขหรือเพิ่มชื่ออุปกรณ์ไฟฟ้า

เมื่อต้องการเลิกควบคุมอุปกรณ์ไฟฟ้าหรือตรวจสอบสถานะของเซนเซอร์ จะต้องทำการ Logout ออกจากระบบ โดยเลือกที่ Logout ในหน้า menu แล้วกด ok ดังรูปที่ 4.50 เมื่อทำการ Logout เสร็จแล้วจะได้ผลดังรูปที่ 4.51

รูปที่ 4.50 แสดงหน้าของ menu เมื่อเลือก Logout

รูปที่ 4.51 แสดงหน้าจอเมื่อทำการ Logout เรียบร้อยแล้ว

และเมื่อสถานะของเซนเซอร์เกิดความผิดปกติขึ้นก็จะส่งข้อความ รายงานไปยังโทรศัพท์เคลื่อนที่เพื่อแจ้งเตือนว่า ขณะนี้มีความมีความผิดปกติเกิดขึ้น โดยจะส่งเป็นเวลาและชื่อของเซนเซอร์ที่เกิดความผิดปกติ ดังรูปที่ 4.52 และรูปที่ 4.53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.52 แสดงหน้าจอเมื่อรับข้อความแจ้งเตือนเข้ามา

รูปที่ 4.53 แสดงข้อความรายงานความผิดปกติจากเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิเคราะห์

สรุปผลการดำเนินงาน

- การทำงานระหว่างอุปกรณ์ผู้ใช้งาน และคอมพิวเตอร์เซิร์ฟเวอร์สามารถใช้งานได้อย่างมีประสิทธิภาพ
- การเชื่อมต่อระหว่างพอร์ทอนุกรม RS-232 และไมโครคอนโทรลเลอร์สามารถรับ-ส่ง ข้อมูลได้จริง
- ระบบสามารถควบคุม ปิด-เปิด อุปกรณ์ไฟฟ้าได้จริง ซึ่งการควบคุมแบ่งออกเป็น 2 แบบ คือ การปิด-เปิด อุปกรณ์ไฟฟ้าแบบทันที และ การปิด-เปิด อุปกรณ์ไฟฟ้าแบบตั้งเวลา
- การทำงานของระบบรักษาความปลอดภัยสามารถใช้งานได้ดี สามารถส่งข้อความเตือนเมื่อเกิดความผิดปกติของเซนเซอร์

ปัญหาและแนวทางการแก้ไขปัญหา

- ไมโครคอนโทรลเลอร์เกิดความเสียหายได้ง่าย
- ถ้าเกิดไฟฟ้าดับจะไม่สามารถควบคุมอุปกรณ์ไฟฟ้าได้เลย ระบบคอมพิวเตอร์เซิร์ฟเวอร์จะล้มเหลวไปด้วย
- ในบางครั้งการแสดงผลของเซนเซอร์ไม่สามารถส่งกลับไปยังคอมพิวเตอร์ได้

การแก้ไขปัญหา

- สร้างสัญญาณไฟเลี้ยงที่มีค่าคงที่ป้อนให้ไมโครคอนโทรลเลอร์
- ใช้ไฟสำรองจากเครื่องสำรองไฟ เมื่อเกิดเหตุไฟฟ้าดับ
- ทำสายส่งสัญญาณจากพอร์ตอนุกรม RS-232 มายังไมโครคอนโทรลเลอร์ที่แข็งแรง ไม่ชำรุดง่ายเมื่อทำการถอดเข้า-ออก

แนวทางการพัฒนา

- ควรมีระบบรักษาความปลอดภัยโดยใช้กล้องวงจรปิด หรือพัฒนาใช้มีหุ่นยนต์รักษาความปลอดภัยภายในอาคารผ่านโทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยการสนับสนุนของบุคคลหลายฝ่าย

ขอขอบพระคุณ รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี ซึ่งเป็นอาจารย์ที่ปรึกษา และกรุณาช่วยให้แนวความคิด ให้คำปรึกษาในโครงงาน ตลอดจนช่วยวิเคราะห์เพื่อแก้ปัญหาต่างๆ อันเป็นประโยชน์ต่อการทำงาน

ขอขอบคุณ ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เอื้อเฟื้อสถานที่และเครื่องมืออุปกรณ์ในการทำวิจัยในห้องปฏิบัติการ

ขอกราบขอบพระคุณผู้ปกครองที่สนับสนุนด้านเงินทุนและกำลังใจอันมีค่าตลอดมา

ขอขอบคุณคุณพงศธรที่คอยให้คำปรึกษา ชี้แนะและช่วยแก้ไขข้อผิดพลาด ตลอดเอื้อเฟื้อเครื่องมืออุปกรณ์ที่ใช้ในโครงงาน และขอขอบคุณพี่ๆ และเพื่อนๆ ที่ช่วยสนับสนุนและให้กำลังใจด้วยดีเสมอมา

ขอบคุณเพื่อนร่วมกลุ่มที่มีความรับผิดชอบช่วยเหลือซึ่งกันและกันเป็นอย่างดีทำให้งานที่ยากบรรเทาลงเป็นงานที่ง่ายและสนุกสนานในการทำงาน

และสุดท้ายนี้ขอขอบคุณทุกท่านที่ได้ให้ความช่วยเหลือที่ไม่ได้กล่าวถึงในที่นี้

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม DynamicV2.java

```
 1 import java.util.Hashtable;
 2 /*
 3  * DynamicV2.java
 4  *
 5  * Created on 25 มกราคม 2548, 22:33 น.
 6  */
 7
 8 /**
 9  *
10  * @author AONKUNG
11  */
12 public class DynamicV2 {
13
14     public static String replace(String source,Hashtable
token,String opentag,String closetag){
15         String var,newstr="";
16         while((var=varLookup(source, opentag, closetag))!=null){
17             if(token.get(var) instanceof String[][]){
18                 String[][] tmp = (String[][])token.get(var);
19                 for(int l=0;l<tmp.length;l++){
20                     if(l<tmp.length-1){
21                         newstr +=source+"\n";
22                     }else{
23                         newstr +=source;
24                     }
25                 }
26                 for(int i=0;i<tmp.length;i++){
27                     for(int j=0;j<tmp[i].length;j++){
28                         newstr = varReplace(newstr, tmp[i][j],
opentag, closetag);
29                     }
30                 }
31             }else{
32                 newstr = varReplace(source,
(String)token.get(var),opentag,closetag);
33             }
34             source=newstr;
35         }
36
37         if(newstr.equals("")){
38             return source;
39         }else
40             return newstr;
41     }
42
43     public static String varLookup(String exp,String open,String
close){
44         String var = null;
45         int startIndex,stopIndex;
46         if((startIndex=exp.indexOf(open)) != -1 &&
(stopIndex=exp.indexOf(close)) != -1){
47             var = exp.substring(startIndex+open.length(),
stopIndex).trim();
48         }
49         return var;
50     }
51
52     public static String varReplace(String exp,String
replace,String open,String close){
53         if(replace == null) replace="N/a";
54         String newstr = "";
55         int startIndex,stopIndex;
56         if((startIndex=exp.indexOf(open)) != -1 &&
(stopIndex=exp.indexOf(close)) != -1){
57             String begin = exp.substring(0,startIndex);
58             String end = exp.substring(stopIndex+close.length(),
exp.length());
```

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
59              newstr = begin+replace+end;
60          }
61
62          return newstr;
63      }
64
65  }
66
```

โปรแกรม Looktor.java

```
 1  import java.io.FileInputStream;
 2  import java.io.FileOutputStream;
 3  import java.io.ObjectInputStream;
 4  import java.io.ObjectOutputStream;
 5  import java.text.SimpleDateFormat;
 6  import java.util.Calendar;
 7  import java.util.Locale;
 8
 9
10  import java.util.Vector;
11  /*
12   * Looktor.java
13   *
14   * Created on January 30, 2005, 8:26 PM
15   */
16
17  /**
18   *
19   * @author ptutut
20   */
21  public class Looktor implements
MySerialPortListener,MyTimerInterface{
22      MyDevice device;
23      MySerialPort serialport;
24      long timeout = 1000;
25      long saveTime;
26
27      Object waitSync = new Object();
28      boolean wait = true;
29
30      byte read;
31      int chNo = 8;
32
33      int envNo = 8;
34      String[] sensor = {"","","","","","","","Switch","Light"};
35      boolean canAlert = true;
36
37      public MyTimer timer;
38      Mail mail = new Mail();
39      int timeDelay = 30;
40
41      String id = "-";
42      boolean lock = false;
43
44      /** Creates a new instance of Looktor */
45      public Looktor() {
46          if(!readDeviceObject()){
47              device = new MyDevice();
48          }
49          resetStatus();
50          serialport = new MySerialPort();
51          serialport.addSerialPortListener(this);
52          serialport.connect();
53
54          timer = new MyTimer();
55          timer.addListener(this);
56          timer.startTime("checksensor", 0, 0, timeDelay);
57      }
58
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
59      public String[][] getDeviceList(){//get All deviceInfo which
has a device name
60          Vector vector = new Vector();
61          String[] a;
62          for(int i=0;i<chNo;i++){
63              if((a=device.getDeviceInfo(""+(i+1))) != null){
64                  if(!a[1].equals(""))
65                      vector.add(a);
66                  }
67          }
68          String[][] list = new String[vector.size()][2];
69          for(int i=0;i<vector.size();i++){
70              String[] c = (String[])vector.get(i);
71              list[i][0] = c[0];
72              list[i][1] = c[1];
73          }
74          return list;
75      }
76
77      public String[] getAllDeviceName(){//return array of all 8
channel device name
78          String[] list = new String[8];
79          String[] a;
80          for(int i=0;i<chNo;i++){
81              a=device.getDeviceInfo(""+(i+1));
82              if(a != null){
83                  list[i] = a[1];
84              }else{
85                  list[i] = "";
86              }
87          }
88          return list;
89      }//get All Device name if any
90
91      public String[] getDeviceInfo(String channel){
92          String[] tmp = device.getDeviceInfo(channel);
93          String[] deviceInfo = (String[])tmp.clone();
94          if(deviceInfo!=null){
95              deviceInfo[0] = deviceInfo[1];
96              if(deviceInfo[2].equals("on")){
97                  deviceInfo[1] = "on";
98                  deviceInfo[2] = "off";
99              }else{
100                 deviceInfo[1] = "off";
101                 deviceInfo[2] = "on";
102             }
103         }
104         return deviceInfo;
105     }
106
107     public void changeDeviceStatus(String ch){
108         String[] a = device.getDeviceInfo(ch);
109         if(a!=null){
110             if(a[2].equals("on")){
111                 device.setDeviceInfo(ch, a[1], "off");
112             }else{
113                 device.setDeviceInfo(ch, a[1], "on");
114             }
115         }
116     }
117
118     public void updateDeviceList(String[] list){
119         String[] tmp = getAllDeviceName();
120         for(int i=0;i<list.length;i++){
121             if(!list[i].equals(tmp[i]) &&
!list[i].equals("N/a")){
122                 if(!tmp[i].equals("")){//previous is not ""
123                     String[] a = device.getDeviceInfo(""+(i+1));
124                     if(a != null){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
125
device.setDeviceInfo(""+(i+1),list[i],a[2]);
126                     }
127                 }else{
128                     device.setDeviceInfo(""+(i+1),list[i],"off");
129                 }
130             }
131         }
132     }
133
134     public boolean turnNow(String channel){
135         if(timer.isTimeSet(channel)){
136             timer.cancel(channel);
137         }
138         write(channel);
139         changeDeviceStatus(channel);
140         return true;
141     }
142
143     public boolean turnOnTime(String channel,String
hourstr,String minstr,String secstr){
144         int hour = Integer.parseInt(hourstr);
145         int min = Integer.parseInt(minstr);
146         int sec = Integer.parseInt(secstr);
147
148         timer.startTime(channel, hour, min, sec);
149         return true;
150     }
151
152     public String[] getEnv(){
153         write("u");
154         byte recv = read();
155         String b = Integer.toBinaryString((int)recv);
156         while(b.length()<8){
157             b = "0"+b;
158         }
159         //p1.7 p1.6 p1.5 p1.4 p1.3 p1.2 p1.1 p1.0
160         //b[7] b[6] b[5] b[4] b[3] b[2] b[1] b[0]
161         //result[7...0]
162         int channelNo =3;//start from port 0
163         String[] result = new String[channelNo];
164         int l = b.length();
165         for(int i=l-1;i>=l-channelNo;i--){
166             if(b.charAt(i)=="0"){
167                 result[l-i-1] = "<font
color=red>abnormal</font>";
168             }else{
169                 result[l-i-1] = "normal";
170             }
171         }
172         return result;
173     }
174
175     private void checkEnv(){
176         write("u");
177         byte recv = read();
178         //recv = (byte)31;
179         String b = Integer.toBinaryString((int)recv);
180         while(b.length()<8){
181             b = "0"+b;
182         }
183         if(b.length()>8){
184             b = b.substring(b.length()-8, b.length());
185         }
186         if(canAlert){
187             //canAlert = false;
188
189             int channelNo =8;//start from port 0
190             int l = b.length();
191             String alert = "";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
192             for(int i=l-1;i>=l-channelNo;i--){
193                 //System.out.println(i);
194                 if(b.charAt(i)=="0"){
195                     alert+=sensor[i]+"\n";
196                 }
197             }
198             if(alert.length() > 0){
199                 System.out.println("Alert:"+alert);
200                 alert("Alert",alert);
201             }
202         }
203         timer.startTime("checksensor", 0, 0, timeDelay);
204     }
205
206     private void alert(String subject,String detail){
207         Calendar cal = Calendar.getInstance();
208         SimpleDateFormat sdate = new
SimpleDateFormat("",Locale.ENGLISH);
209         sdate.applyPattern("d MMM yyyy h:mm a");
210         String date = sdate.format(cal.getTime());
211         mail.setSubject(subject+" "+date);
212         mail.setFill(detail);
213         mail.setAttachFile(null);
214         mail.send();
215     }
216
217     public boolean timerIsOK(String hour,String min){
218         int shour = Integer.parseInt(hour);
219         int smin = Integer.parseInt(min);
220         Calendar cal = Calendar.getInstance();
221         int chour = cal.get(Calendar.HOUR_OF_DAY);
222         int cmin = cal.get(Calendar.MINUTE);
223         boolean timerOK;
224         if(shour > chour){
225             timerOK = true;
226         }
227         else if(shour < chour){
228             timerOK = false;
229         }
230         else{
231             if(smin > cmin){
232                 timerOK = true;
233             }
234             else if(smin < cmin){
235                 timerOK = false;
236             }
237             else{
238                 timerOK = false;
239             }
240         }
241         return timerOK;
242     }
243
244     public synchronized boolean canAccess(String id){
245         if(id.equals(this.id) || this.id.equals("-")){
246             this.id = id;
247             return true;
248         }else{
249             return false;
250         }
251     }
252
253     public void logout(String id){
254         if(id.equals(this.id)){
255             this.id = "-";
256         }
257     }
258
259     private boolean write(String data){
260         serialport.out((data.getBytes())[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
261         return true;
262     }
263
264     private byte read(){
265         //waitSync = new Object();
266         wait = true;
267         saveTime = System.currentTimeMillis();
268         try{
269             synchronized (waitSync) {
270                 while (wait && System.currentTimeMillis() -
saveTime < timeout) {
271                     waitSync.wait(1000);
272                 }
273             }
274         } catch (Exception e) {}
275         if(wait){
276             System.out.println("Data can not receive.");
277             read = (byte)-1;
278         }
279         wait = true;
280         return read;
281     }
282
283     private boolean readDeviceObject(){
284         try{
285             FileInputStream fin = new
FileInputStream("MyDevice");
286             ObjectInputStream obin = new ObjectInputStream(fin);
287             device = (MyDevice) obin.readObject();
288             fin.close();
289             obin.close();
290         }catch(Exception e){
291             return false;
292         }
293         return true;
294     }
295
296     private void writeDeviceObject(){
297         try{
298             FileOutputStream fout = new
FileOutputStream("MyDevice");
299             ObjectOutputStream obout = new
ObjectOutputStream(fout);
300             obout.writeObject(device);
301             fout.close();
302             obout.close();
303         }catch(Exception e){
304             e.printStackTrace();
305         }
306     }
307
308     private void resetStatus(){//reset all channel to off
309         for(int i=0;i<chNo;i++){
310             String[] a = device.getDeviceInfo(""+(i+1));
311             if(a!=null){
312                 if(a[2].equals("on")){
313                     a[2] = "off";
314                     device.setDeviceInfo(""+(i+1), a[1], a[2]);
315                 }
316             }
317         }
318     }
319
320     public void SerialPortRecieveData(byte msg){
321         read = msg;
322         if(wait){
323              synchronized (waitSync) {
324                     wait = false;
325                     waitSync.notifyAll();
326             }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
327         }
328     }
329
330     public synchronized void timesUP(String channel){
331         if(channel.equals("checksensor")){
332             checkEnv();
333         }
334         else{
335             turnNow(channel);
336         }
337     }
338
339     public String getTimeRemind(String ch){
340         if(timer.isTimeSet(ch)){
341             int[] time = timer.getTimeRemain(ch);
342             String text = "Remain Time: "+time[0]+" hour
"+time[1]+" min "+time[2]+" sec";
343             return text;
344         }else{
345             return "";
346         }
347     }
348
349     public String getTimeUPAt(String channel){
350         if(timer.isTimeSet(channel)){
351             return timer.getTimesUP(channel);
352         }else{
353             return "";
354         }
355     }
356
357     public void close(){
358         serialport.disconnect();
359         writeDeviceObject();
360     }
361
362     public static void main(String arg[]){
363         Looktor looktor = new Looktor();
364         //looktor.device.clear();
365         //String[] pong = {"pongpong","test","","","","","","",""};
366         //looktor.getDeviceInfo("2");
367         looktor.timer.startTime("1",0,0,5);
368         looktor.close();
369     }
370 }
371
```

โปรแกรม LooktorServlet.java

```
 1 /*
 2  * LooktorServlet.java
 3  *
 4  * Created on March 14, 2005, 11:08 PM
 5  */
 6
 7 import java.io.*;
 8 import java.util.Hashtable;
 9
10 import javax.servlet.*;
11 import javax.servlet.http.*;
12
13 /**
14  *
15  * @author ptutut
16  * @version
17  */
18 public class LooktorServlet extends HttpServlet {
19
20     Hashtable hash;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
21      String path = "C:/jakarta-tomcat-
5.0.28/webapps/Looktor/web/";
22      Looktor looktor;
23      int userNum = 0;
24      /** Initializes the servlet.
25       */
26      public void init(ServletConfig config) throws
ServletException {
27          super.init(config);
28          path = getServletContext().getRealPath("")+"/";
29          hash = new Hashtable();
30          looktor = new Looktor();
31      }
32
33      /** Destroys the servlet.
34       */
35      public void destroy() {
36          hash = null;
37          path = null;
38          looktor.close();
39      }
40
41      /** Processes requests for both HTTP <code>GET</code> and
<code>POST</code> methods.
42       * @param request servlet request
43       * @param response servlet response
44       */
45      protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
46      throws ServletException, IOException {
47          response.setContentType("text/html");
48          PrintWriter out = response.getWriter();
49          String page = request.getParameter("page");
50          String link;
51          String result = "";
52          String ip = request.getRemoteAddr();
53
54          if(page == null){
55              sendURI(response, "page1.html");
56          }
57          else if(page.equals("page1")){
58              sendURI(response,"login.html");
59          }
60          else if(page.equals("login")){
61              String username = request.getParameter("username");
62              String password = request.getParameter("password");
63              if(username.equals("home") &&
password.equals("1234")){
64                  if(looktor.canAccess(ip)){
65                      sendURI(response,"menu.html");
66                  }
67                  else{
68                      result = "Anothor user is using the system.
Try again later.";
69                      hash.put("result",result);
70                      hash.put("link","login");
71                      sendURI(response,"result.html");
72                  }
73              }else{
74                  hash.clear();
75                  result = "Your username or password is
incorrected !! Please try again !!";
76                  hash.put("result",result);
77                  hash.put("link","login");
78                  sendURI(response,"result.html");
79              }
80          }if(page.equals("menu")){//menu.html
81              link = request.getParameter("link");
82              if(link.equals("control")){
83                  String[][] deviceList = looktor.getDeviceList();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(deviceList.length == 0){
                    hash.clear();
                    hash.put("disabled","disabled");
                }else{
                    hash.put("deviceList",deviceList);
                }
                sendURI(response,"control.html");
            }
            else if(link.equals("env")){
                hash.clear();
                String[] envResult = looktor.getEnv();
                for(int i=0;i<envResult.length;i++){
                    hash.put("sensor"+(i+1),envResult[i]);
                }
                sendURI(response,"env.html");
            }
            else if(link.equals("setting")){
                String[] allList = looktor.getAllDeviceName();
                hash.clear();
                for(int i=0;i<allList.length;i++){
                    hash.put("channel"+(i+1),allList[i]);
                }
                sendURI(response,"setting.html");
            }
            else if(link.equals("logout")){
                looktor.logout(ip);
                result = "Logout complete.";
                hash.put("result",result);
                hash.put("link","login");
                sendURI(response,"result.html");
            }
        }
        else if(page.equals("control")){//control.html
            link = request.getParameter("link");
            if(link != null){
                sendURI(response,"menu.html");
            }else{
                String channel = request.getParameter("channel");
                hash.put("channel",channel);
                String[] deviceInfo =
looktor.getDeviceInfo(channel);
                if(deviceInfo !=null){
                    hash.put("device",deviceInfo[0]);
                    hash.put("status1",deviceInfo[1]);
                    hash.put("status2",deviceInfo[2]);
                    String timeRemain =
looktor.getTimeUPAt(channel);
                    if(timeRemain.equals("")){
                        hash.put("timedisable","");
                        hash.put("timeup","");
                    }else{
                        hash.put("timedisable","disabled");
                        hash.put("timeup","Turn "+deviceInfo[2]+"
at "+timeRemain);
                    }
                }
                sendURI(response,"device.html");
            }
        }
        else if(page.equals("env")){//env.html
                sendURI(response,"menu.html");
        }
        else if(page.equals("setting")){//setting.html
            link = request.getParameter("link");
            if(link != null){
                sendURI(response,"menu.html");
            }else{
                String[] list =
request.getParameterValues("channel[]");
                looktor.updateDeviceList(list);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
150                 hash.clear();
151                 result = "UPDATE DEVICE LIST COMPLETE.";
152                 hash.put("result",result);
153                 hash.put("link","menu");
154                 sendURI(response, "result.html");
155             }
156         }
157         else if(page.equals("device")){//device.html
158             link = request.getParameter("link");
159             if(link != null){
160                 sendURI(response,"menu.html");
161             }else{
162                 String action = request.getParameter("action");
163                 String channel = request.getParameter("channel");
164                 if(action.equals("turnnow")){
165                     looktor.turnNow(channel);
166                 }else{
167                     String hour = request.getParameter("hour");
168                     String min = request.getParameter("min");
169                     looktor.turnOnTime(channel,hour,min,"0");
170                     if(looktor.timerIsOK(hour,min)){//timer >
current time
171
172                     }else{//timer <= current time
173                         //looktor.turnNow(channel);
174                     }
175                 }
176                 hash.clear();
177                 result = "YOUR ACTION COMPLETED.";
178                 hash.put("result",result);
179                 hash.put("link","menu");
180                 sendURI(response, "result.html");
181             }
182         }
183         else if(page.equals("result")){
184             link = request.getParameter("link");
185             if(link.equals("login")){
186                 sendURI(response,"login.html");
187             }
188             else if(link.equals("menu")){
189                 sendURI(response,"menu.html");
190             }
191         }
192         else{
193             hash.put("name","PONG");
194             sendURI(response, "test.html");
195         }
196         out.close();
197     }
198
199     private void sendURI(HttpServletResponse response, String
file){
200         response.setContentType("text/html");
201
202         try{
203             PrintWriter out = response.getWriter();
204             File uri = new File(path+file);
205             FileReader fr = new FileReader(uri);
206             BufferedReader br = new BufferedReader(fr);
207             String text = "";
208             while((text=br.readLine())!=null){
209                 text = DynamicV2.replace(text, hash,"<!", "!>");
210                 out.println(text);
211             }
212             br.close();
213             out.close();
214         }catch(Exception e){e.printStackTrace();}
215     }
216
217     /** Handles the HTTP <code>GET</code> method.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
218      * @param request servlet request
219      * @param response servlet response
220      */
221     protected void doGet(HttpServletRequest request,
HttpServletResponse response)
222     throws ServletException, IOException {
223         processRequest(request, response);
224     }
225
226     /** Handles the HTTP <code>POST</code> method.
227      * @param request servlet request
228      * @param response servlet response
229      */
230     protected void doPost(HttpServletRequest request,
HttpServletResponse response)
231     throws ServletException, IOException {
232         processRequest(request, response);
233     }
234
235     /** Returns a short description of the servlet.
236      */
237     public String getServletInfo() {
238         return "Short description";
239     }
240 }
241
```

โปรกรม Mail.java

```
  1 /*
  2  * MyMail.java
  3  *
  4  * Created on 8 พฤศจิกายน 2547, 15:34 น.
  5  */
  6
  7 /**
  8  *
  9  * @author  AONKUNG
 10  */
 11 import java.util.Properties;
 12 import javax.activation.DataHandler;
 13 import javax.activation.DataSource;
 14 import javax.activation.FileDataSource;
 15 import javax.mail.BodyPart;
 16 import javax.mail.Message;
 17 import javax.mail.Multipart;
 18 import javax.mail.Session;
 19 import javax.mail.Transport;
 20 import javax.mail.internet.InternetAddress;
 21 import javax.mail.internet.MimeBodyPart;
 22 import javax.mail.internet.MimeMessage;
 23 import javax.mail.internet.MimeMultipart;
 24
 25
 26
 27 public class Mail {
 28     String host = "mail.kmitl.ac.th";
 29     String from = "s4010320@kmitl.ac.th";
 30     String to = "d096368026@mail.dtac.co.th";
 31     String subject = "Alert";
 32     String fill = "";
 33     String attachFile = null;
 34
 35     /** Creates a new instance of MyMail */
 36     public boolean send() {
 37         // Get system properties
 38         Properties props = System.getProperties();
 39         // Setup mail server
 40         props.put("mail.smtp.host", host);
 41         // Get session
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
42          Session session = Session.getDefaultInstance(props,
null);
43
44          try{
45          // Define message
46          Message message = new MimeMessage(session);
47          message.setFrom(new InternetAddress(from));
48          message.addRecipient(Message.RecipientType.TO, new
InternetAddress(to));
49          message.setSubject(subject);
50
51          // Create the message part
52          BodyPart messageBodyPart = new MimeBodyPart();
53
54          // Fill the message
55          messageBodyPart.setText(fill);
56
57          Multipart multipart = new MimeMultipart();
58          multipart.addBodyPart(messageBodyPart);
59
60          // Part two is attachment
61          if(attachFile != null){
62              messageBodyPart = new MimeBodyPart();
63              DataSource source = new FileDataSource(attachFile);
64              messageBodyPart.setDataHandler(new
DataHandler(source));
65
66              String[] tmp = attachFile.split("/");
67              String fileName = attachFile;
68              if(tmp.length > 0) fileName = tmp[tmp.length-1];
69
70              messageBodyPart.setFileName(fileName);
71              multipart.addBodyPart(messageBodyPart);
72          }
73          // Put parts in message
74          message.setContent(multipart);
75          // Send the message
76          Transport.send(message);
77          }catch(Exception e){
78              e.printStackTrace();
79              return false;
80          }
81          System.out.println("Send Message Complete");
82          return true;
83      }
84
85      public void setHost(String host){
86          this.host = host;
87      }
88
89      public void setFrom(String from){
90          this.from = from;
91      }
92
93      public void setTo(String to){
94          this.to = to;
95      }
96
97      public void setSubject(String subject){
98          this.subject = subject;
99      }
100
101     public void setFill(String fill){
102         this.fill = fill;
103     }
104
105     public void setAttachFile(String attachFile){
106         this.attachFile = attachFile;
107     }
108
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
109     public static void main(String[] arg){
110         Mail mail = new Mail();
111         mail.setHost("mail.kmitl.ac.th");
112         //mail.setTo("t.pongsatorn@gmail.com");
113         //mail.setTo("d069990311@mail.dtac.co.th");
114         //mail.setTo("ptutut@hotmail.com");
115
mail.setAttachFile("E:/PONG/Use/Robot/src/hsrbot/resource/mail/00298_
0.jpg");
116         mail.send();
117     }
118 }
119
```

โปรแกรม MyDevice.java

```
 1 import java.io.Serializable;
 2 import java.util.Hashtable;
 3 /*
 4  * MyDevice.java
 5  *
 6  * Created on January 30, 2005, 8:28 PM
 7  */
 8
 9 /**
10  *
11  * @author ptutut
12  */
13 public class MyDevice implements Serializable{
14     private Hashtable channel;
15     /** Creates a new instance of MyDevice */
16     public MyDevice() {
17         channel = new Hashtable();
18     }
19
20     public void setDeviceInfo(String ch,String name,String
status){
21         String[] deviceInfo = {ch,name,status};
22         channel.put(ch,deviceInfo);
23     }
24
25     public String[] getDeviceInfo(String ch){
26         String[] tmp = (String[])channel.get(ch);
27         return tmp;
28     }
29
30     public void clear(){
31         channel.clear();
32     }
33 }
34
```

โปรแกรม MySerialPort.java

```
 1 /*
 2  * SerialPortConnect.java
 3  *
 4  * Created on December 14, 2003, 11:45 PM
 5  */
 6 import java.io.*;
 7 import java.util.*;
 8 import javax.comm.*;
 9 /**
10  *
11  * @author  Administrator
12  */
13 public class MySerialPort implements SerialPortEventListener{
14     private SerialPort serialPort;
15     private InputStream inputStream;
16     private CommPortIdentifier portId;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
17      protected MySerialPortListener listener;
18      String port = "COM1";
19      int baud = 9600;
20
21      /** Creates a new instance of SerialPortConnect */
22      public void connect() {
23          try{
24              portId = CommPortIdentifier.getPortIdentifier(port);
25              System.out.println("Found:"+portId.getName());
26          }catch (NoSuchPortException e){
27              e.printStackTrace();
28              System.out.println("PORT:"+port+"  NOT FOUND!!");
29              System.exit(-1);
30          }
31
32          try {
33              serialPort = (SerialPort) portId.open("SimpleReadApp",
2000);
34              System.out.println("Connection Completed!!");
35          } catch (PortInUseException e) {
36              System.out.println("Connection Fail!!");
37          }
38
39          try {
40              inputStream = serialPort.getInputStream();
41          } catch (IOException e) {}
42
43          try {
44              serialPort.addEventListener(this);
45          } catch (TooManyListenersException e) {}
46
47          serialPort.notifyOnDataAvailable(true);
48
49          try {
50              serialPort.setSerialPortParams(baud,
SerialPort.DATABITS_8,
51                                             SerialPort.STOPBITS_1,
52
SerialPort.PARITY_NONE);
53          } catch (UnsupportedCommOperationException e) {}
54          System.out.println("Connect at "+baud);
55      }
56
57      public void disconnect(){
58          serialPort.close();
59          System.out.println("Serial port is disconnected");
60      }
61
62      public void addSerialPortListener(MySerialPortListener l){
63          this.listener = l;
64      }
65
66      public void setPort(String port){
67          this.port = port;
68      }
69
70      public void setBaudrate(int baud){
71          this.baud = baud;
72      }
73
74      public void out(byte msg){
75          try{
76              OutputStream outS = serialPort.getOutputStream();
77              outS.write(msg);
78              outS.close();
79          }catch(IOException e){}
80      }
81
82      public void serialEvent(javax.comm.SerialPortEvent sEvent) {
83          byte b=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
84          if(sEvent.getEventType() ==
SerialPortEvent.DATA_AVAILABLE){
85              byte[] readBuffer = new byte[10];
86              try {
87                  while (inputStream.available() > 0) {
88                      int numBytes = inputStream.read(readBuffer);
89                  }
90                  b = readBuffer[0];
91              } catch (IOException e) {}
92              listener.SerialPortRecieveData(b);
93          }
94      }
95
96  }
97
```

โปรแกรม MySerialPortListener.java

```
 1 /*
 2  * MySerialPortListener.java
 3  *
 4  * Created on December 15, 2003, 12:19 AM
 5  */
 6 /**
 7  *
 8  * @author  Administrator
 9  */
10 public interface MySerialPortListener {
11     public void SerialPortRecieveData(byte msg);
12 }
13
```

โปรแกรม MyTimer.java

```
 1 import java.util.Hashtable;
 2 /*
 3  * MyTimer.java
 4  *
 5  * Created on 31 มกราคม 2548, 13:47 น.
 6  */
 7
 8 /**
 9  *
10  * @author AONKUNG
11  */
12 public class MyTimer implements ReminderListener{
13     Hashtable hash;
14     MyTimerInterface listener = null;
15     /** Creates a new instance of MyTimer */
16     public MyTimer() {
17         hash = new Hashtable();
18     }
19
20     public synchronized void startTime(String channel,int hour,int
min,int sec){
21         Reminder remind = new Reminder(channel,hour,min,sec);
22         remind.addListener(this);
23         hash.put(channel,remind);
24         remind.start();
25     }
26
27     public synchronized boolean isTimeSet(String channel){
28         Object obj = hash.get(channel);
29         if(obj==null){
30             return false;
31         }else{
32             return true;
33         }
34     }
35
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
36      public synchronized boolean cancel(String channel){
37          Reminder remind = (Reminder)hash.get(channel);
38          if(!remind.hasFinish()){
39              remind.cancel();
40              hash.remove(channel);
41              return true;
42          }else{
43              return false;
44          }
45      }
46
47      public void reset(String channel,int hour,int min,int sec){
48          cancel(channel);
49          startTime(channel, hour, min, sec);
50      }
51
52      public int[] getTimeRemain(String channel){
53          Reminder remind = (Reminder)hash.get(channel);
54          int timeInsec = remind.getTimeRemainInSec();
55          int sec = timeInsec%60;
56          int min = ((int)Math.floor(timeInsec/60))%60;
57          int hour = ((int)Math.floor(timeInsec/360));
58          int[] ret = {hour,min,sec};
59          return ret;
60      }
61
62      public String getTimesUP(String channel){
63          Reminder remind = (Reminder)hash.get(channel);
64          return remind.getTimesUP();
65      }
66
67      public void timesUP(String ch) {//Time"s up for this channel
68          if(listener != null){
69              hash.remove(ch);
70              listener.timesUP(ch);
71          }
72      }
73
74      public void addListener(Object obj){
75          this.listener = (MyTimerInterface) obj;
76      }
77
78      public static void main(String arg[]){
79          MyTimer timer = new MyTimer();
80          timer.startTime("1", 0, 0, 10);
81          timer.startTime("2", 0, 0, 0);
82          timer.startTime("3", 0, 0, 0);
83      }
84  }
85
86
```

โปรแกรม MyTimerInterface.java

```
 1  /*
 2   * MyTimerInterface.java
 3   *
 4   * Created on 3 กุมภาพันธ์ 2548, 19:02 น.
 5   */
 6
 7
 8  /**
 9   *
10   * @author AONKUNG
11   */
12  public interface MyTimerInterface {
13      public void timesUP(String channel);
14  }
15
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Reminder.java

```
  1 import java.text.SimpleDateFormat;
  2 import java.util.Calendar;
  3 import java.util.Locale;
  4 import java.util.Timer;
  5 import java.util.TimerTask;
  6
  7 /**
  8  * Simple demo that uses java.util.Timer to schedule a task to
execute
  9  * once 5 seconds have passed.
 10  */
 11
 12 public class Reminder extends Thread{
 13     Timer timer;
 14     Calendar calendar;
 15     long id;
 16     String channel;
 17     boolean isCanceled = false;
 18     boolean finished = false;
 19
 20     ReminderListener listener;
 21
 22     public Reminder(String ch,int hour,int min,int sec) {
 23         calendar = Calendar.getInstance();
 24         int chour = Calendar.HOUR_OF_DAY;
 25         calendar.set(chour, calendar.get(chour)+hour);
 26         int cmin = Calendar.MINUTE;
 27         calendar.set(cmin, calendar.get(cmin)+min);
 28         int csec = Calendar.SECOND;
 29         calendar.set(csec, calendar.get(csec)+sec);
 30
 31         timer = new Timer();
 32         this.id = super.getId();
 33         this.channel = ch;
 34     }
 35
 36     public long getId(){
 37         return id;
 38     }
 39
 40     public void cancel(){
 41         this.isCanceled = true;
 42         timer.cancel();
 43         finished = true;
 44     }
 45
 46     public int getTimeRemainInSec(){
 47         Calendar cal = Calendar.getInstance();
 48         int hour = calendar.get(Calendar.HOUR_OF_DAY)-
cal.get(Calendar.HOUR_OF_DAY);
 49         int min = calendar.get(Calendar.MINUTE)-
cal.get(Calendar.MINUTE);
 50         int sec = calendar.get(Calendar.SECOND)-
cal.get(Calendar.SECOND);
 51         if(hour<0) hour=0;
 52         if(min<0){
 53             min = 60+min;
 54             hour--;
 55         }
 56         if(sec<0){
 57             sec = 60+sec;
 58             min--;
 59         }
 60         int timeInSec = hour*60*60+min*60+sec;
 61         return timeInSec;
 62     }
 63     public String getTimesUP(){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
64          SimpleDateFormat sdate = new
SimpleDateFormat("",Locale.ENGLISH);
65          sdate.applyPattern("d MMM yyyy h:mm:ss a");
66          String date = sdate.format(calendar.getTime());
67          return date;
68      }
69
70      public int getHourRemain(){
71          Calendar cal = Calendar.getInstance();
72          int hour = calendar.get(Calendar.HOUR_OF_DAY)-
cal.get(Calendar.HOUR_OF_DAY);
73          return hour;
74      }
75
76      public int getMinRemain(){
77          Calendar cal = Calendar.getInstance();
78          int min = calendar.get(Calendar.MINUTE)-
cal.get(Calendar.MINUTE);
79          if(min<0) min = 60+min;
80          return min;
81      }
82
83      public int getSecRemain(){
84          Calendar cal = Calendar.getInstance();
85          int sec = calendar.get(Calendar.SECOND)-
cal.get(Calendar.SECOND);
86          if(sec<0) sec = 60+sec;
87          return sec;
88      }
89
90      public boolean hasFinish(){
91          return finished;
92      }
93
94      public void addListener(Object obj){
95          this.listener = (ReminderListener) obj;
96      }
97
98      public void run(){
99          System.out.println("Channel: "+channel+" timer start.");
100         if(!isCanceled){
101             timer.schedule(new RemindTask(),
calendar.getTime());
102         }
103     }
104
105     class RemindTask extends TimerTask {
106         public void run() {
107             //System.out.println("Time"s up! From: "+id);
108             if(listener!=null) listener.timesUP(channel);
109             if(!isCanceled){
110                 timer.cancel();
111                 finished = true;
112             }  //Terminate the timer thread
113         }
114     }
115
116     public static void main(String args[]) {
117         Reminder mytime = new Reminder("1",0,0,50);
118         mytime.start();
119         boolean boo = true;
120         try{
121             while(boo){
122                 int hour = mytime.getHourRemain();
123                 int min  = mytime.getMinRemain();
124                 int sec = mytime.getSecRemain();
125                 System.out.println("Remind:"+hour+"h "+min+"m
"+sec+"s");
126                 Thread.sleep(1000);
127                 if(hour == 0 && min == 0 && sec == 0){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
128                     boo = false;
129                 }
130             }
131         }catch(Exception e){}
132     }
133 }
134
135 interface ReminderListener{
136     public void timesUP(String ch);
137 }
```

โปรแกรม Timer.java

```
 1 /*
 2  * Timer.java
 3  *
 4  * Created on January 31, 2005, 12:58 AM
 5  */
 6
 7 /**
 8  *
 9  * @author ptutut
10  */
11 public class Timer  extends Thread{
12     TimerInterface listener;
13     String channel;
14     int time;
15     /** Creates a new instance of Timer */
16     public Timer(String channel,int hour,int min) {
17         this.channel = channel;
18         this.time = ((hour*60+min)*60)*1000;
19     }
20
21     public void run(){
22         try{
23             sleep(time);
24         }catch(Exception e){
25             e.printStackTrace();
26         }
27
28         if(listener!=null){
29             listener.timeUP(channel);
30         }
31     }
32
33     public void addListener(Object obj){
34         this.listener = (TimerInterface)obj;
35     }
36 }
37
38 interface TimerInterface{
39     public void timeUP(String ch);
40 }
41
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. รองศาสตราจารย์ สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์", คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546
2. กิตติ ภักดีวัฒนะกุล, "คัมภีร์ Java เล่ม 1", KTP COMP&CONSULT ISBN: 974-90928-8-0, 2546
3. สราวุธ อ้อยศรีสกุล, "เปิดมิติ Mobile Internet ด้วยWAP", บริษัท วิตตี้กรุ๊ป จำกัด ISBN: 974-87931-0- 9, 2544
4. ดร. วีระศักดิ์ ซึงถาวร, "JAVA Programming (Volume 1)", กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2543
5. ดร. วีระศักดิ์ ซึงถาวร, "JAVA Programming (Volume 2)", กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้