



## การทดสอบและประเมินผล

ในบทนี้ได้กล่าวถึงการทดสอบการทำงานของซอฟต์แวร์ของระบบสกาดาดันแบบ โดยในหัวข้อแรกจะกล่าวถึงระบบสกาดานระบบปฏิบัติการลินุกซ์ที่ได้จากการออกแบบในบทที่ 3 มีคุณสมบัติและฟังก์ชันการทำงานเป็นอย่างไรต่อจากนั้นจะเป็นการทดสอบที่ละเอียดขึ้นโดยที่จำเป็นต้องมีการกำหนดอุปกรณ์ที่ใช้สำหรับการทดสอบก่อนและในหัวข้อสุดท้ายจะเป็นการสรุปผลจากการทดสอบ

### 4.1 คุณลักษณะของซอฟต์แวร์ของระบบสกาดานระบบปฏิบัติการลินุกซ์ต้นแบบ

หัวข้อที่จะกล่าวต่อไปนี้เป็นารอธิบายรายละเอียดของคุณสมบัติและฟังก์ชันการทำงาน ของซอฟต์แวร์สกาดาดันแบบที่สร้างขึ้นในบทที่ 3 โดยมีรายละเอียดดังต่อไปนี้

4.1.1 คุณสมบัติของซอฟต์แวร์ของระบบสกาดานระบบปฏิบัติการลินุกซ์ต้นแบบ ในส่วนนี้แสดงถึงสมมติฐานของข้อกำหนดขอบเขตและความสามารถในการทำงานของซอฟต์แวร์ที่ได้ออกแบบไว้มีคุณสมบัติดังต่อไปนี้ดังต่อไปนี้

- สถานีหลักสามารถติดต่อกับ RTU ด้วยโพรโตคอลสื่อสารแบบมอดบัล แอสกีบนมาตรฐาน RS-485 ได้ 10 สถานี โดยที่แต่ละสถานีรับสัญญาณ แอนะล็อกได้ 8 ช่องสัญญาณ สัญญาณดิจิทัลอินพุต/เอาต์พุตอย่างละ 4 ช่องสัญญาณ
- เวลาต่ำสุดที่สถานีหลักอ่านข้อมูลจากสถานี RTU ต่อ 1 RTU เท่ากับ 1 วินาที
- สามารถตรวจความผิดปกติจากการสื่อสารระหว่างสถานีหลักกับ RTU ระหว่างการทำงานได้ เช่นกรณีที่สายส่งสัญญาณหลุด
- สนับสนุนการควบคุมระยะไกลจากเครื่องไคลเอนท์
- การใช้พื้นที่ในการเก็บข้อมูลต่อการโพลลิ่ง 1 ครั้งต่อหนึ่งสถานีมีค่าเท่ากับ 85 ไบต์ซึ่งหมายความว่าถ้าต้องการเก็บข้อมูล 1,000,000 ครั้งของการโพลลิ่งต้องใช้จานบันทึกแบบแข็ง (Hard Disk) 85 เมกกะไบต์ (MB)

- การแสดงผลสื่อความหมายได้อย่างชัดเจนในรูปแบบทั่วไปและรูปแบบกราฟ
- สามารถแจ้งเตือนความผิดปกติในรูปแบบความเปลี่ยนแปลงของสีที่หน้าจอและอุปกรณ์ภายนอกเช่นหลอดไฟหรือสัญญาณเตือนโดยใช้พอร์ตดิจิตอลเอาต์พุตของ RTU
- ใช้ความเร็วในการสื่อสารแบบมอดบัสแอสกีบนมาตรฐาน RS-485 เท่ากับ 19200 บิตต่อวินาที และสามารถเพิ่มได้ถึง 57600 บิตต่อวินาที
- สามารถตรวจสอบการทำงาน (Monitoring) ในรูปแบบของไคลเอนท์/เซิร์ฟเวอร์ ได้ทั้งบนระบบปฏิบัติการลินุกซ์ (RedHat 7.2) และไมโครซอฟต์วินโดวส์ (Win2000)
- สามารถบันทึกข้อมูลในรูปแบบไฟล์เอกสารหรือรูปภาพเพื่อที่จะนำไปใช้ในการทำรายงานต่อได้

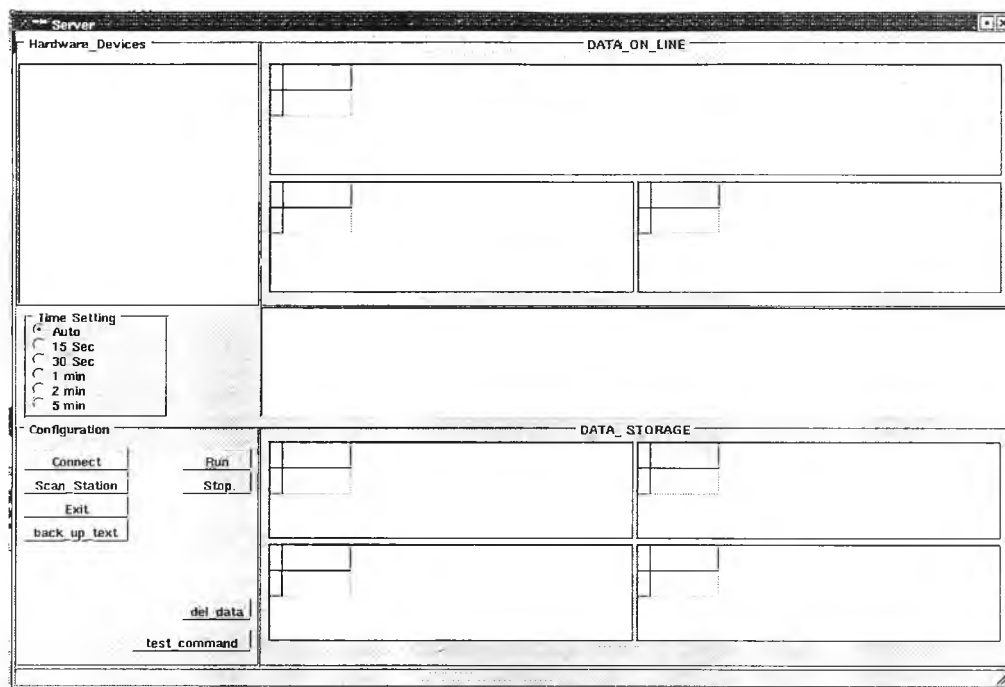
4.1.2 ฟังก์ชันการทำงานของซอฟต์แวร์ของสกาดาบระบบปฏิบัติการลินุกซ์ต้นแบบแบ่งได้เป็น 2 ส่วนการทำงานคือ การประยุกต์ใช้งานของส่วนติดต่อกับผู้ใช้บนเซิร์ฟเวอร์และการประยุกต์ใช้งานของส่วนติดต่อกับผู้ใช้บนไคลเอนท์ มีรายละเอียดดังต่อไปนี้

#### การประยุกต์ใช้งานของส่วนติดต่อกับผู้ใช้บนเซิร์ฟเวอร์

การประยุกต์ใช้งานของส่วนติดต่อกับผู้ใช้บนเซิร์ฟเวอร์คือส่วนที่ทำหน้าที่ในการอ่านข้อมูลจาก RTU ขึ้นมาบันทึกไว้ในฐานข้อมูลเพื่อช่วยในการแสดงผลและเก็บข้อมูลประวัติโดยมีฟังก์ชันที่ช่วยกันทำงานในส่วนนี้อยู่ 5 ฟังก์ชันคือ

- 1 ฟังก์ชันตรวจสอบสถานะการติดต่อสื่อสารระหว่างสถานีหลักกับ RTU
- 2 ฟังก์ชันตรวจสอบประเภทของช่องสัญญาณของแต่ละ RTU
- 3 ฟังก์ชันอ่านข้อมูลและกำหนดหน้าที่การทำงานให้กับ RTU
- 4 ฟังก์ชันกำหนดช่วงเวลาในการอ่านข้อมูลแต่ละรอบการทำงาน
- 5 ฟังก์ชันตรวจสอบการทำงานกับฐานข้อมูลและแสดงผล

โดยหน้าต่างการใช้งานของการประยุกต์ใช้งานของเซิร์ฟเวอร์แสดงได้ดังรูปที่ 4.1 ส่วนในรายละเอียดการใช้งานจะขอล่าไว้ในคู่มือการใช้งานของภาคผนวก ค



รูปที่ 4.1 หน้าต่างแสดงการทำงานของการทำงานของการประยุกต์ใช้งานของเซิร์ฟเวอร์

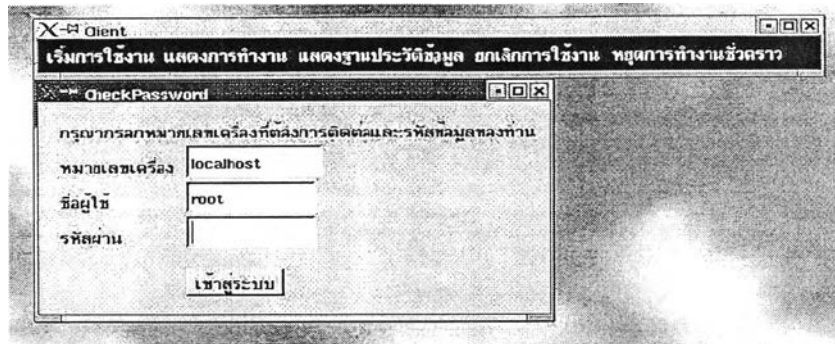
#### การประยุกต์ใช้งานของส่วนติดต่อกับผู้ใช้บนไคลเอนท์

เป็นส่วนการประยุกต์การทำงานในรูปของระบบไคลเอนท์/เซิร์ฟเวอร์ ที่ส่วนการประยุกต์ใช้งานของส่วนติดต่อประสานงานกับผู้ใช้บนไคลเอนท์มีหน้าที่หลักในการทำงานคืออ่านข้อมูลจาฐานข้อมูลของเซิร์ฟเวอร์มาแสดงผลในรูปแบบขณะเวลาจริงในลักษณะข้อมูลทั่วไปหรือกราฟและในรูปแบบเทียบกับเวลาจริงของฐานข้อมูลประวัติรวมทั้งสามารถในการควบคุมระยะไกลได้อีกด้วย โดยในส่วนนี้มีฟังก์ชันช่วยในการทำงานอยู่ 7 ฟังก์ชันคือ

- 1 ฟังก์ชันตรวจรหัสผ่านของผู้ใช้ มีหน้าที่ในการตรวจสอบสิทธิการใช้งานของผู้ใช้ โดยการตรวจสอบรหัสผ่านแสดงได้ดังรูปที่ 4.2 และ รูปที่ 4.3



รูปที่ 4.2 หน้าต่างแสดงการทำงานของส่วนการประยุกต์ใช้งานบนไคลเอนท์

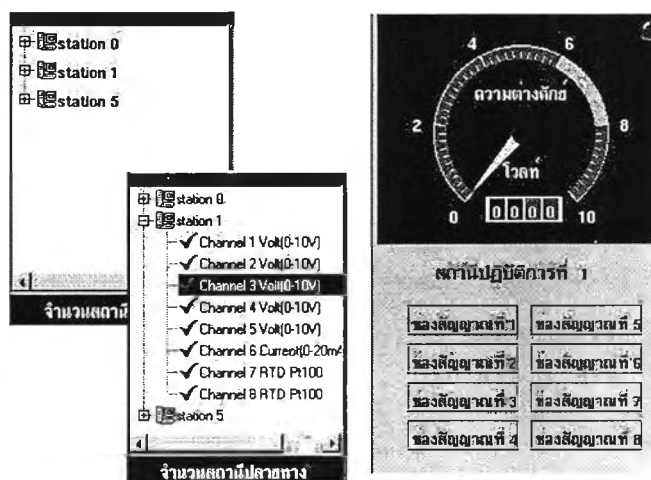


รูปที่ 4.3 หน้าต่างฟังก์ชันการตรวจสอบรหัสผู้ใช้ของส่วนการประยุกต์การใช้งานบนไคลเอนท์  
2 ฟังก์ชันตรวจสอบการติดต่อกับฐานข้อมูล มีหน้าที่ในการตรวจสอบความพร้อมก่อนการทำงานของฐานข้อมูลแสดงได้ดังรูปที่ 4.4



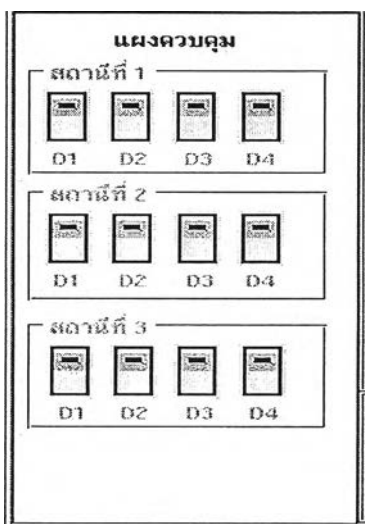
รูปที่ 4.4 หน้าต่างฟังก์ชันการตรวจสอบการติดต่อกับฐานข้อมูลของส่วนการประยุกต์การใช้งานบนไคลเอนท์

3 ฟังก์ชันตรวจสอบการติดตั้งของ RTU ทำหน้าที่ในการตรวจสอบการติดตั้ง RTU และแยกประเภทของช่องสัญญาณในแต่ละ RTU แสดงดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างฟังก์ชันตรวจสอบการติดตั้ง RTU ของส่วนการประยุกต์การใช้งานบนไคลเอนท์

- 4 ฟังก์ชันควบคุม RTU ทำหน้าที่ในการเปิด/ปิดเอาต์พุตดิจิทัลของ RTU จากส่วนการประยุกต์การใช้งานบนไคลเอนท์



รูปที่ 4.6 หน้าต่างฟังก์ชันสำหรับการควบคุม RTU ของส่วนการประยุกต์การใช้งานบนไคลเอนท์

- 5 ฟังก์ชันควบคุมการแสดงผลในรูปแบบทั่วไป ทำหน้าที่ในการแสดงผลในรูปแบบต่างๆ เช่นดิจิทัลอินพุต/เอาต์พุต รูปแบบตัวอักษร รูปแบบของมาตรวัด ตลอดจนลักษณะการเตือนบนหน้าจอ ดังแสดงในรูปที่ 4.7 – 4.11



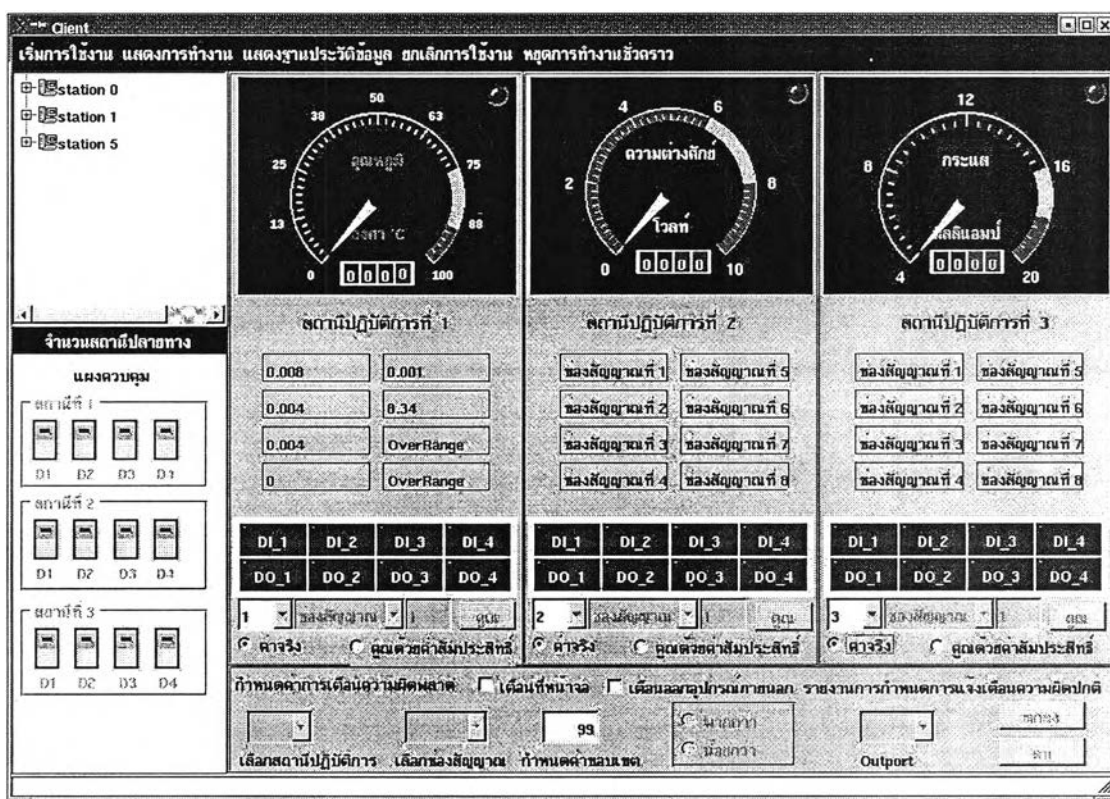
รูปที่ 4.7 หน้าต่างฟังก์ชันสำหรับแสดงสถานะของอินพุต / เอาต์พุต ดิจิตอล RTU ของการส่วนประยุกต์การใช้งานบนไคลเอนท์



รูปที่ 4.8 หน้าต่างฟังก์ชันสำหรับการกำหนดการแจ้งความผิดพลาดของ RTU ของส่วนการประยุกต์การใช้งานบนไคลเอนท์



รูปที่ 4.9 หน้าต่างฟังก์ชันสำหรับแสดงสถานะของอินพุตแอนะล็อกของ RTU ของส่วนการประยุกต์การใช้งานบนไคลเอนท์

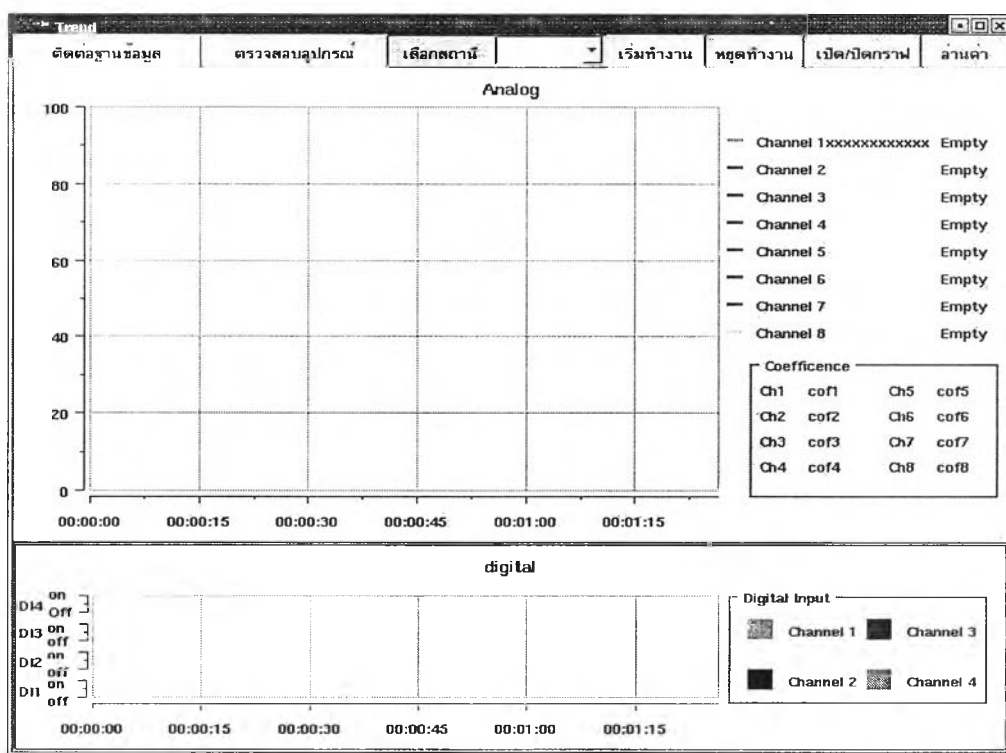


รูปที่ 4.10 หน้าต่างแสดงผลในรูปแบบทั่วไปของส่วนการประยุกต์การใช้งานบนไคลเอนท์ [19]

สถานีที่ 0	สถานีที่ 1	สถานีที่ 2	สถานีที่ 3	สถานีที่ 4	สถานีที่ 5	สถานีที่ 6	สถานีที่ 7	สถานีที่ 8	สถานีที่ 9
สถานีที่ 0	Label Low	Label High	Output Low	Output High	Value Low	Value High			
ช่องสัญญาณที่ 1									
ช่องสัญญาณที่ 2									
ช่องสัญญาณที่ 3									
ช่องสัญญาณที่ 4									
ช่องสัญญาณที่ 5									
ช่องสัญญาณที่ 6									
ช่องสัญญาณที่ 7									
ช่องสัญญาณที่ 8									

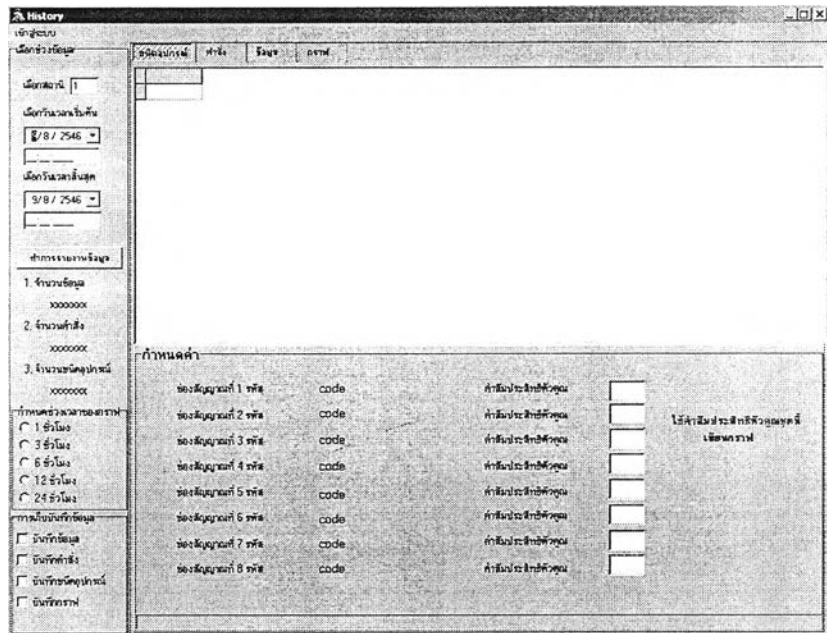
รูปที่ 4.11 หน้าต่างฟังก์ชันสำหรับแสดงสถานะของการกำหนดการแจ้งจากการประยุกต์การใช้งานบนไคลเอนท์ไปยัง RTU

- 6 ฟังก์ชันควบคุมการแสดงผลในรูปแบบของกราฟ มีหน้าที่ในการดึงข้อมูลจากฐานข้อมูลแบบ Online มาแสดงผลในรูปแบบกราฟเทียบกับเวลาจริง โดยจะมีการแสดงในรูปของสัญญาณดิจิทัลและแอนะล็อกพร้อมๆกัน

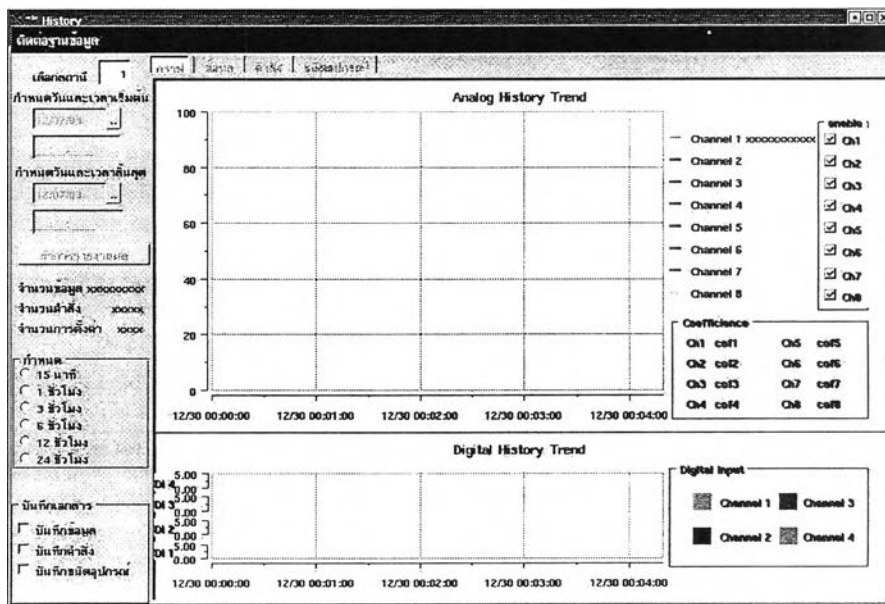


รูปที่ 4.12 หน้าต่างฟังก์ชันสำหรับแสดงสถานะของอินพุตแอนะล็อกและอินพุตดิจิทัลในรูปแบบของกราฟของสถานี RTU ของส่วนการประยุกต์การใช้งานบนไคลเอนท์

- 7 ฟังก์ชันควบคุมการแสดงผลจากฐานข้อมูลประวัติ มีหน้าที่ในการดึงข้อมูลจากฐานข้อมูลประวัติมาแสดงผลในรูปของจำนวนข้อมูล, คำสั่ง, ชนิดอุปกรณ์ตลอดจนการเขียนกราฟดังแสดงในรูปที่ 4.13 และ 4.14



รูปที่ 4.13 หน้าต่างฟังก์ชันสำหรับแสดงข้อมูลประวัติของสถานะของอินพุตและเอาต์พุตของ RTU จากส่วนการประยุกต์การใช้งานบนไคลเอนท์

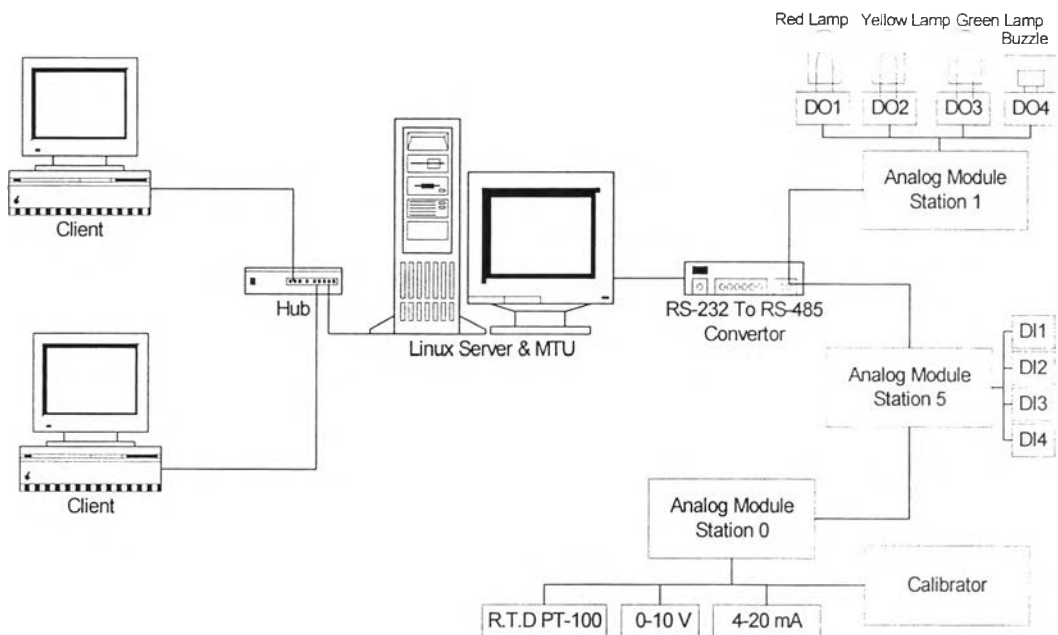


รูปที่ 4.14 หน้าต่างฟังก์ชันสำหรับแสดงข้อมูลประวัติของสถานะอินพุตและเอาต์พุตดิจิทัลในรูปแบบของกราฟของส่วนการประยุกต์การใช้งานบนไคลเอนท์



## 4.2 ข้อกำหนดของระบบและอุปกรณ์ในการทดสอบซอฟต์แวร์ของระบบสกาดานระบบปฏิบัติการลินุกซ์

เป็นส่วนของการกำหนดต่างๆที่ใช้ในการทดสอบซอฟต์แวร์ที่สร้างขึ้นโดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.15 โครงสร้างการติดตั้งระบบฮาร์ดแวร์เพื่อทดสอบการทำงานของระบบสกาดานระบบปฏิบัติการลินุกซ์

4.2.1 โครงสร้างการติดตั้งระบบฮาร์ดแวร์เพื่อทดสอบการทำงานของระบบสกาดานระบบปฏิบัติการลินุกซ์ ในระบบที่สร้างขึ้นมาเพื่อทำการทดสอบจะประกอบไปด้วย

- สถานีหลักทำงานอยู่บนระบบปฏิบัติการลินุกซ์
- ส่วนการแสดงผลและควบคุมบนระบบปฏิบัติการลินุกซ์
- ส่วนการแสดงผลและควบคุมบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์
- RTU ที่มีรายการวัดและควบคุมดังต่อไปนี้
  - แรงดัน จากตัวจ่ายสัญญาณมาตรฐาน (Calibrator)
  - กระแส จากตัวแปลงสัญญาณหัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100) เป็นกระแสในช่วง 4-20 มิลลิแอมป์
  - อุณหภูมิ จากหัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100)

- ดิจิตอลอินพุต จากปุ่มกด
- ดิจิตอลเอาต์พุต ควบคุมหลอดไฟ 220 โวลต์และเครื่องกำเนิดสัญญาณเตือน

#### 4.2.2 ข้อกำหนดทางฮาร์ดแวร์และซอฟต์แวร์ของสถานีหลัก (MTU)

- หน่วยประมวลผลกลาง (CPU) AMD 1.7+ กิกกะเฮิร์ตซ์ (GHz)
- แรม (Ram) 640 เมกกะไบต์ (MB)
- จานบันทึกแบบแข็ง (Hard Disk) 40 กิกกะไบต์(GB)
- การ์ดแสดงผล (Monitor Card) TNT2 16 เมกกะไบต์ (MB)
- ระบบปฏิบัติการลินุกซ์ของบริษัทเรดแฮท (Red Hat) รุ่น 7.2
- ระบบปฏิบัติการวินโดวส์ 98 และ 2000 (Win98, Win2000)

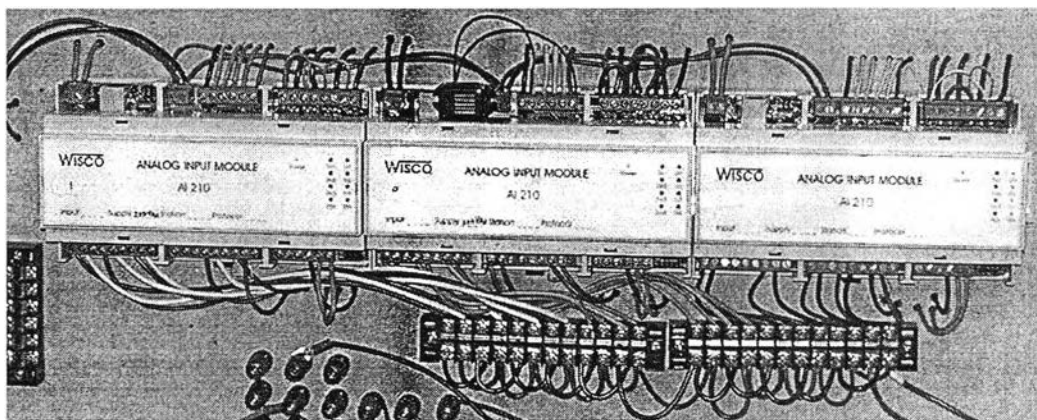
#### 4.2.3 ข้อกำหนดทางฮาร์ดแวร์ของ RTU

- ตัวแปลงสัญญาณมาตรฐาน RS-232 เป็น RS-485



รูปที่ 4.16 ตัวแปลงสัญญาณมาตรฐาน RS-232 เป็น RS-485

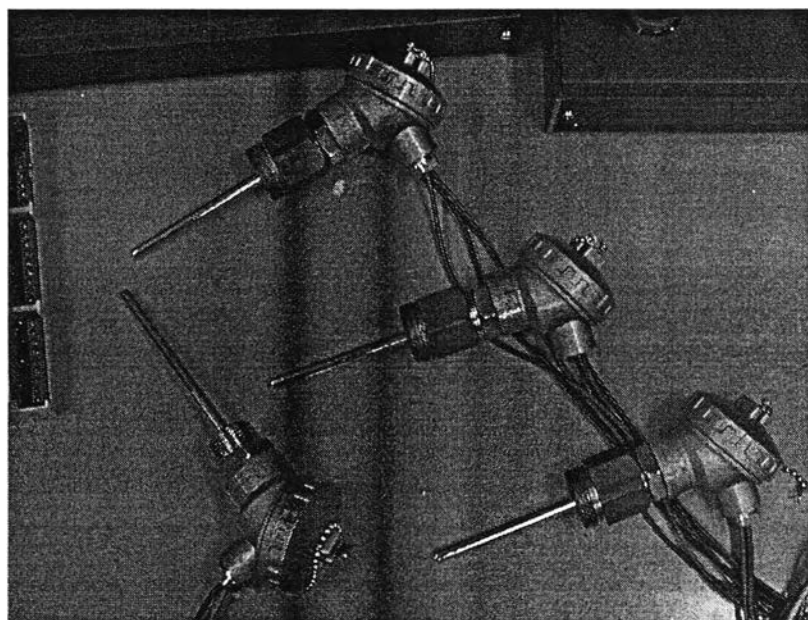
- ตัวรับสัญญาณแอนะล็อก(Analog Input Module AI 210)



รูปที่ 4.17 ตัวรับสัญญาณแอนะล็อก(Analog Input Module AI 210) [22]

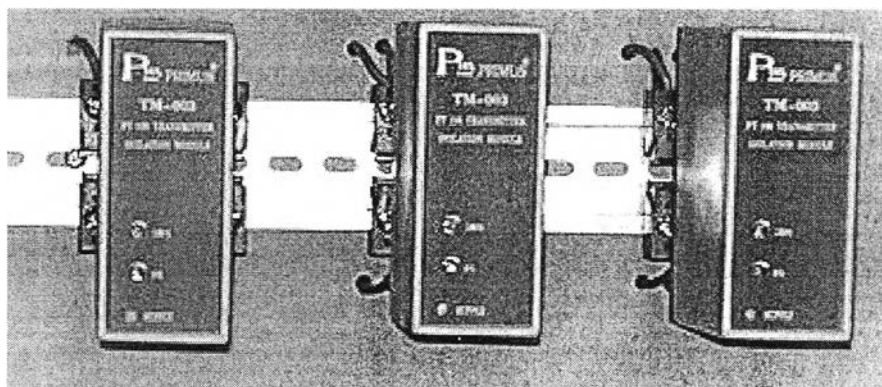
#### 4.2.4 ข้อกำหนดของอุปกรณ์ที่ใช้ในการทดสอบ

- หัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100)



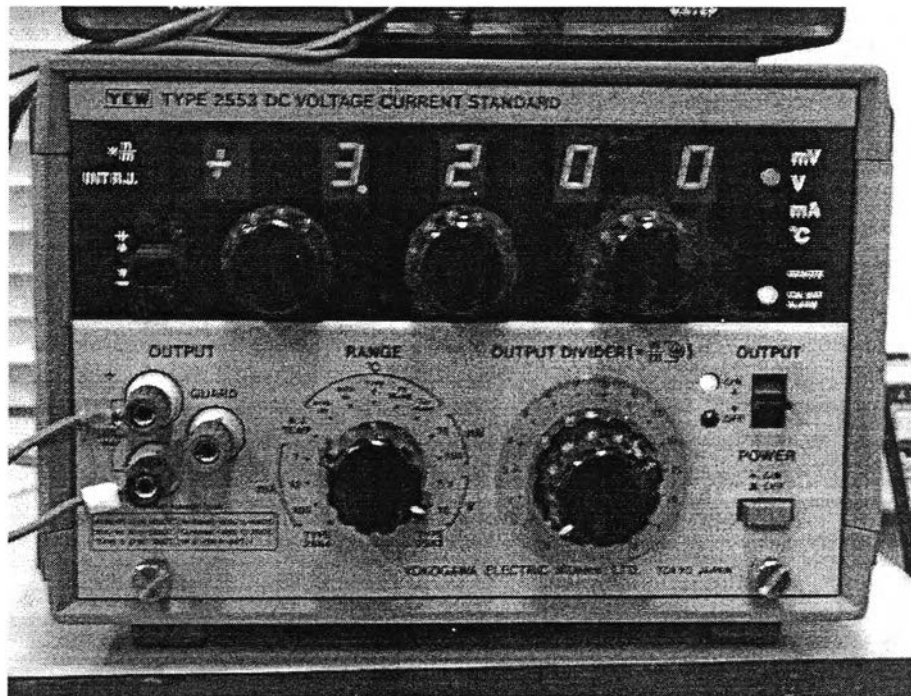
รูปที่ 4.18 หัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100)

- ตัวแปลงสัญญาณหัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100) เป็นกระแสในช่วง 4-20 มิลลิแอมป์ 100 องศาเซลเซียส



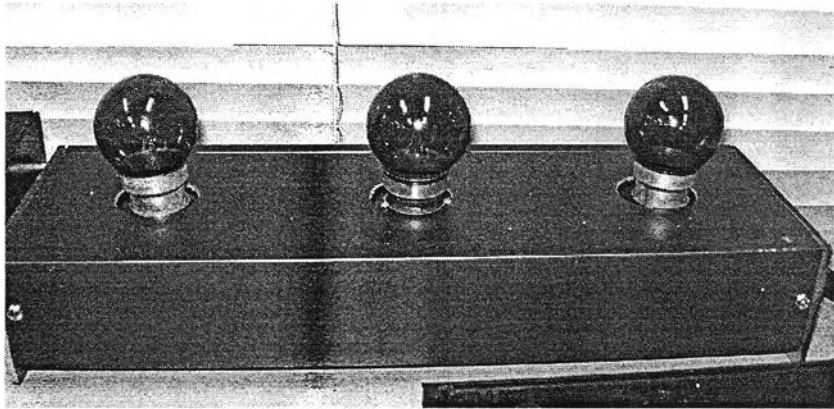
รูปที่ 4.19 ตัวแปลงสัญญาณหัววัดอุณหภูมิอาร์ทีดี พีที-100 (R.T.D PT-100) เป็นกระแสในช่วง 4-20 มิลลิแอมป์ 100 องศาเซลเซียส

- ตัวจ่ายสัญญาณมาตรฐาน (Calibrator)

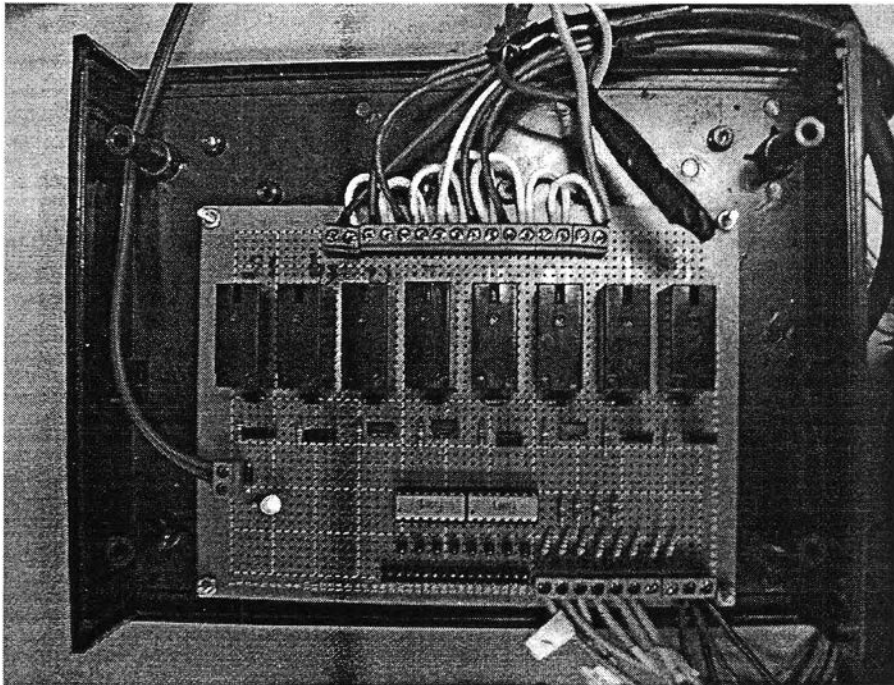


รูปที่ 4.20 ตัวจ่ายสัญญาณมาตรฐาน (Calibrator)

- ชุดขับรีเลย์และหลอดไฟ 220 โวลต์



รูปที่ 4.21 ชุดหลอดไฟ 220 โวลต์



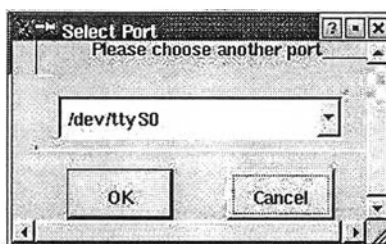
รูปที่ 4.22 ชุดขับรีเลย์ขับหลอดไฟ 220 โวลต์

### 4.3 การทดสอบซอฟต์แวร์ของระบบสกาตาบนระบบปฏิบัติการลินุกซ์

4.3.1 ทดสอบตามคุณสมบัติของโปรแกรม (Functional) เป็นการทดสอบคุณสมบัติการทำงานของระบบสกาตาบนระบบปฏิบัติการลินุกซ์ต้นแบบทั้งระบบ โดยมีรายละเอียดการทดสอบดังต่อไปนี้

4.3.1.1 การทดสอบการติดต่อสื่อสารกับ RTU ตามมาตรฐาน RS-485 มีการแบ่งการทดสอบเป็น 2 ขั้นตอนคือ การทดสอบความสามารถการทำงานตามข้อกำหนดการสื่อสารแบบมอดบั๊สของระบบปฏิบัติการลินุกซ์ และการทดสอบการใช้งานโพรโตคอลมอดบั๊สในระบบสกาตาบนระบบปฏิบัติการลินุกซ์ มีรายละเอียดดังต่อไปนี้

- การทดสอบความสามารถการทำงานตามข้อกำหนดการสื่อสารแบบมอดบั๊สของระบบปฏิบัติการลินุกซ์ โดยตามข้อกำหนดการสื่อสารของโพรโตคอลมอดบั๊สแอสกี ต้องกำหนดให้พอร์ตสื่อสารอนุกรมให้มีคุณสมบัติการสื่อสารดังนี้คือ ใช้ 1 บิตสำหรับบิตเริ่ม (Start bit) 7 บิตสำหรับบิตข้อมูล (Data bit) 1 บิตสำหรับบิตแสดงสภาวะคู่หรือคี่ (Parity bit) และอีก 1 บิตสำหรับบิตหยุด (Stop bit) แต่อาจเป็น 2 บิตถ้าไม่มีการใช้บิตแสดงสภาวะคู่หรือคี่ ดังแสดงตามตารางที่ 4.1



รูปที่ 4.23 การเลือกพอร์ตสื่อสารอนุกรมบนระบบปฏิบัติการลินุกซ์

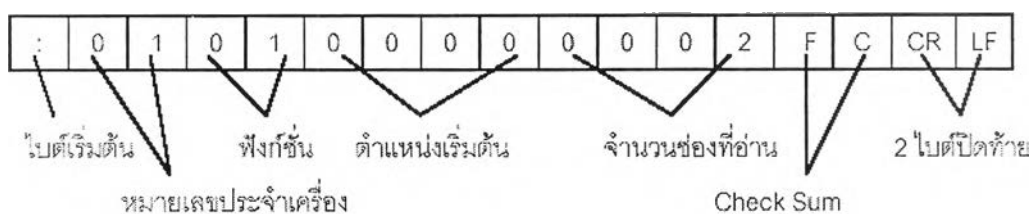
ตารางที่ 4.1 การกำหนดบิตในการรับส่งข้อมูลของพอร์ตอนุกรม

Data Bit	Number
Start bit	1
Data bit	7
Parity bit	1 or None
Stop bit	1 or 2

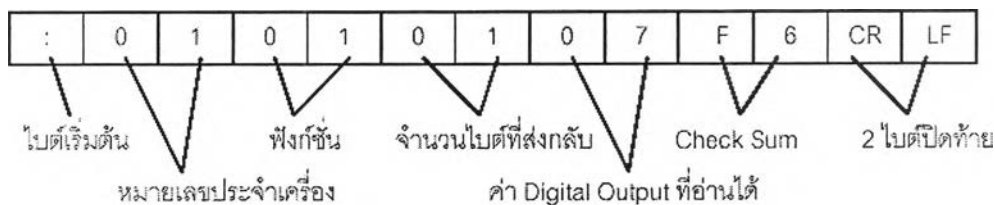
ขั้นตอนต่อไปคือการกำหนดชุดคำสั่งที่จะนำมาทดสอบซึ่งมีโครงสร้างการรับส่งข้อมูลดังแสดงได้ตามรูปที่ 4.26 และ 4.27 โดยที่การกำหนดชุดคำสั่งในการทดสอบถูกกำหนดไว้ในตารางที่ 4.2 และผลจากการทดสอบแสดงได้ดังตารางที่ 4.3 และ 4.4

ตารางที่ 4.2 การกำหนดชุดคำสั่งที่ใช้ทดสอบการทำงานกับพอร์ตอนุกรม

Station	Command	Code Modbus
1	Read Digital Output Channel 1 - 2	:010100000002FC
0	Read Analog Input Channel 1 - 8	:000400000008F4
5	Read Digital Input Channel 1 - 4	:050200000004F5



รูปที่ 4.24 ตัวอย่างการส่งชุดคำสั่งไปยังสถานีปฏิบัติการ



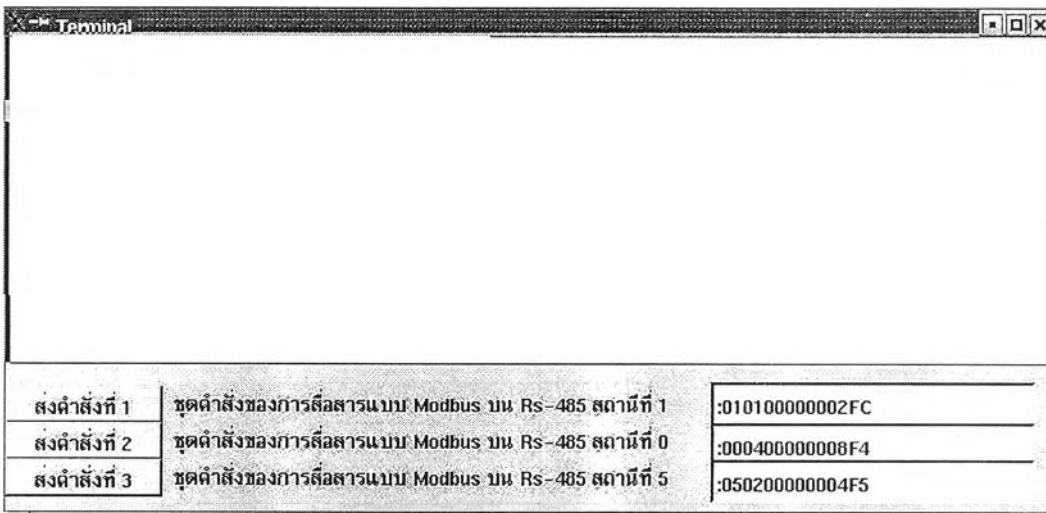
รูปที่ 4.25 ตัวอย่างการตอบกลับจากสถานีปฏิบัติการ

ตารางที่ 4.3 การตอบสนองของ RTU จากสถานีต่างๆ

Station	Code Modbus
1	:01010100FD
0	:0004100C800C800C80000103498888011348
5	:0502010FE9

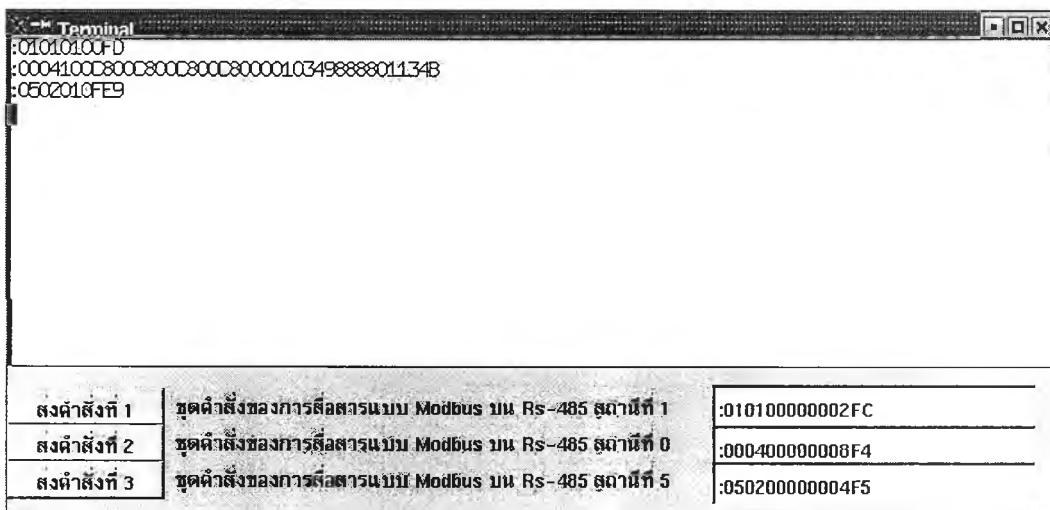
ตารางที่ 4.4 การตอบสนองของ RTU จากสถานีต่างๆเมื่อแปลความหมายแล้ว

Station	Meanings
1	Digital Output Ch1 = Off , Ch 2 = Off
0	Analog Ch 1 - 4 = 3.2 V , Ch 5 = 0.01 V , Ch6 = 8.41 mA , Ch7 = Overrange , Ch8 = 27.5 'C
5	Digital Input Ch1 = On , Ch2 = On , Ch3 =On , Ch4= On



ส่งคำสั่งที่ 1	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 1	:01010000002FC
ส่งคำสั่งที่ 2	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 0	:000400000008F4
ส่งคำสั่งที่ 3	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 5	:050200000004F5

รูปที่ 4.26 หน้าต่างที่ใช้ทดสอบการทำงานของพอร์ตอนุกรมบนระบบปฏิบัติการลินุกซ์



ส่งคำสั่งที่ 1	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 1	:01010000002FC
ส่งคำสั่งที่ 2	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 0	:000400000008F4
ส่งคำสั่งที่ 3	ชุดคำสั่งของการสื่อสารแบบ Modbus บน Rs-485 สถานีที่ 5	:050200000004F5

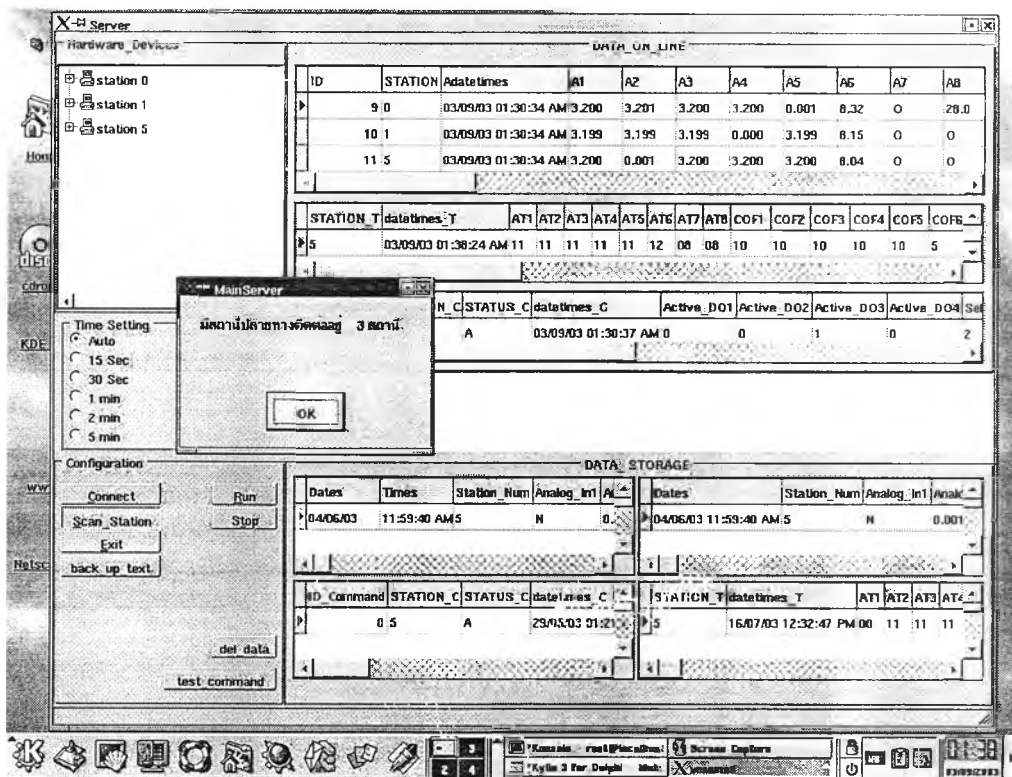
รูปที่ 4.27 ผลการทดสอบการทำงานของพอร์ตอนุกรมบนระบบปฏิบัติการลินุกซ์



ในวิทยานิพนธ์ฉบับนี้ได้ทำการทดสอบการส่งคำสั่งจำนวนหลายครั้งในหลายคำสั่งและหลายช่วงความเร็วในการส่งข้อมูลตั้งแต่ 9600 บิตต่อวินาที จนถึง 57600 บิตต่อวินาที จนมั่นใจว่าระบบปฏิบัติการลินุกซ์สามารถสื่อสารกันด้วยโพรโตคอลได้อย่างดีเยี่ยม

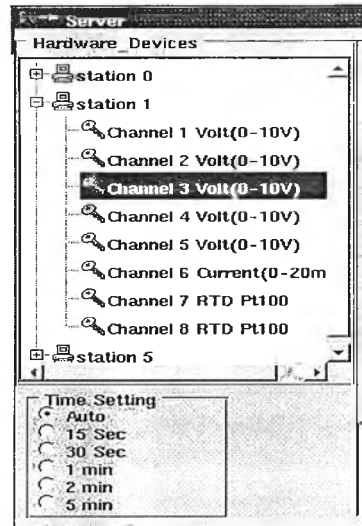
- การทดสอบการใช้งานโพรโตคอลมอดบัสในระบบสกาตาบนระบบปฏิบัติการลินุกซ์ เป็นการทดสอบการทำงานจากระบบสกาตาในส่วนของระบบเซิร์ฟเวอร์ คือใช้ความสามารถในการติดต่อสื่อสารทำงานดังต่อไปนี้

- ตรวจสอบการติดตั้ง RTU ในระบบสกาตาบนระบบปฏิบัติการลินุกซ์ต้นแบบนี้ได้ออกแบบไว้ให้สามารถติดต่อกับ RTU ได้ 10 RTU การทดสอบคือการนำ RTU จำนวน 3 RTU มาติดตั้งดังรูปที่ 4.15 และทำการตรวจสอบการทำงานได้ผลดังรูปที่ 4.28



รูปที่ 4.28 ผลการทดสอบตรวจสอบการติดตั้ง RTU

- แยกประเภทของช่องสัญญาณของแต่ละ RTU



รูปที่ 4.29 ผลการทดสอบการแยกประเภทช่องสัญญาณของ RTU

- อ่านข้อมูลและกำหนดหน้าที่การทำงานให้กับ RTU

DATA_ON_LINE												
STATION	A datetimes	A1	A2	A3	A4	A5	A6	A7	A8	D11	D12	D
0	03/09/03 01:41:56 AM	3.201	3.201	3.201	3.201	0.001	8.30	0	27.9	0	0	0
1	03/09/03 01:41:56 AM	3.199	3.199	3.199	0.000	3.199	8.19	0	0	0	0	0
5	03/09/03 01:41:57 AM	3.199	0.001	3.199	3.199	3.199	8.04	0	0	0	0	0

STATION_T	datetimes_T	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	COF1	COF2	COF3	COF4	COF5	COF6
0	03/09/03 01:40:57 AM	11	11	11	11	11	12	08	08	10	10	10	10	10	5

STATION_C	STATUS_C	datetimes_C	Active_DO1	Active_DO2	Active_DO3	Active_DO4	Select_Ch
5	A	03/09/03 01:42:52 AM	0	0	1	0	2

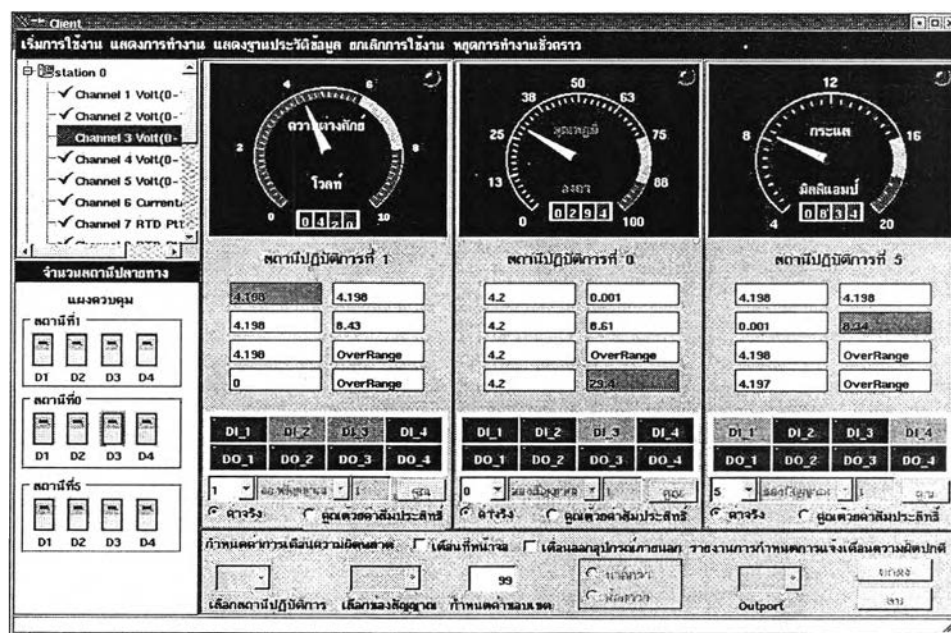
รูปที่ 4.30 ผลการทดสอบการอ่านข้อมูลจาก RTU

ผลจากการทดสอบความสามารถในการใช้โพรโตคอลมอดบัสในการสื่อสารบนระบบสกาดานาในส่วนของเซิร์ฟเวอร์ สามารถสรุปได้ว่าระบบที่จำลองขึ้นสามารถทำงานได้ตามข้อกำหนดในการออกแบบระบบไว้คือ

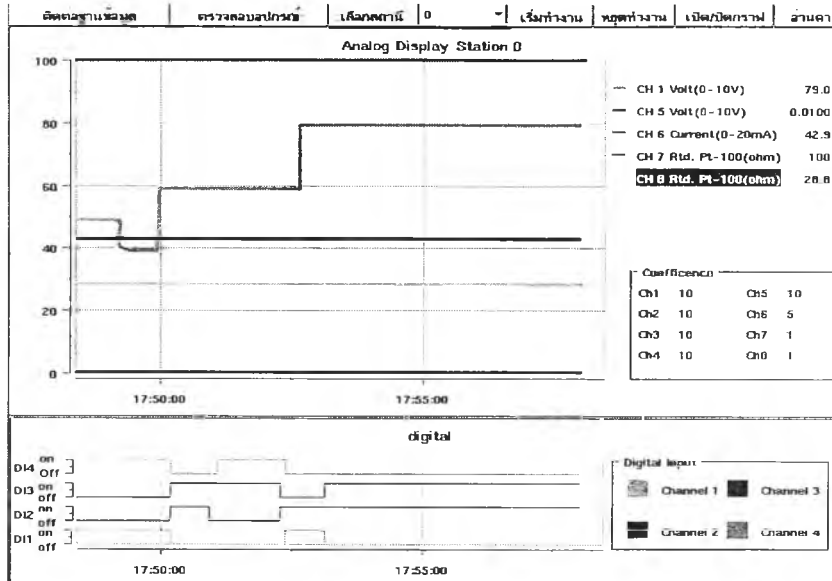
- ระบบสามารถตรวจสอบการติดตั้ง RTU ได้
- ระบบสามารถแยกประเภทของช่องสัญญาณของแต่ละ RTU ได้
- ระบบสามารถกำหนดเวลาในการอ่านข้อมูลขึ้นมาแสดงได้โดยเวลาที่ต่ำที่สุดที่ไม่ทำให้เกิดการสูญเสียในการอ่านข้อมูลต่อ 1 RTU คือ 2 วินาที

4.3.1.2 การทดสอบการแสดงผลเทียบกับเวลาจริง จากโครงสร้างการติดตั้งระบบฮาร์ดแวร์เพื่อทดสอบการทำงานของระบบสกาดาบระบบปฏิบัติการลินุกซ์ในรูปที่ 4.15 จะแบ่งการทดสอบได้เป็นสองส่วนคือ

- การแสดงผลบนเครื่องโคลเอนท์ที่ใช้ระบบปฏิบัติการลินุกซ์ ซึ่งเป็นการทดสอบการแสดงผลโดยการป้อนสัญญาณอินพุตในรูปแบบต่างๆให้กับสถานีหลักแล้วสังเกตความสอดคล้องกันของการเปลี่ยนแปลงต่างๆที่เกิดขึ้นในส่วนประสานงานกับผู้ใช้ของสถานีหลักกับส่วนการประสานงานกับผู้ใช้ของเครื่องโคลเอนท์ที่ทำงานบนระบบปฏิบัติการลินุกซ์ ผลที่ได้แสดงดังรูปที่ 4.31 และ 4.32

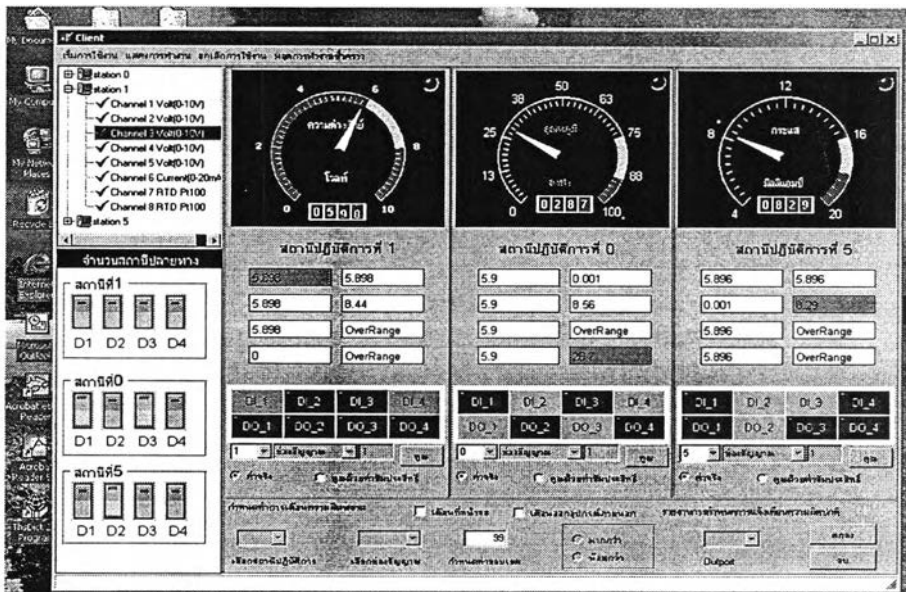


รูปที่ 4.31 ผลการทดสอบการแสดงผลในรูปทั่วไปของส่วนประสานงานกับผู้ใช้บนระบบปฏิบัติการลินุกซ์



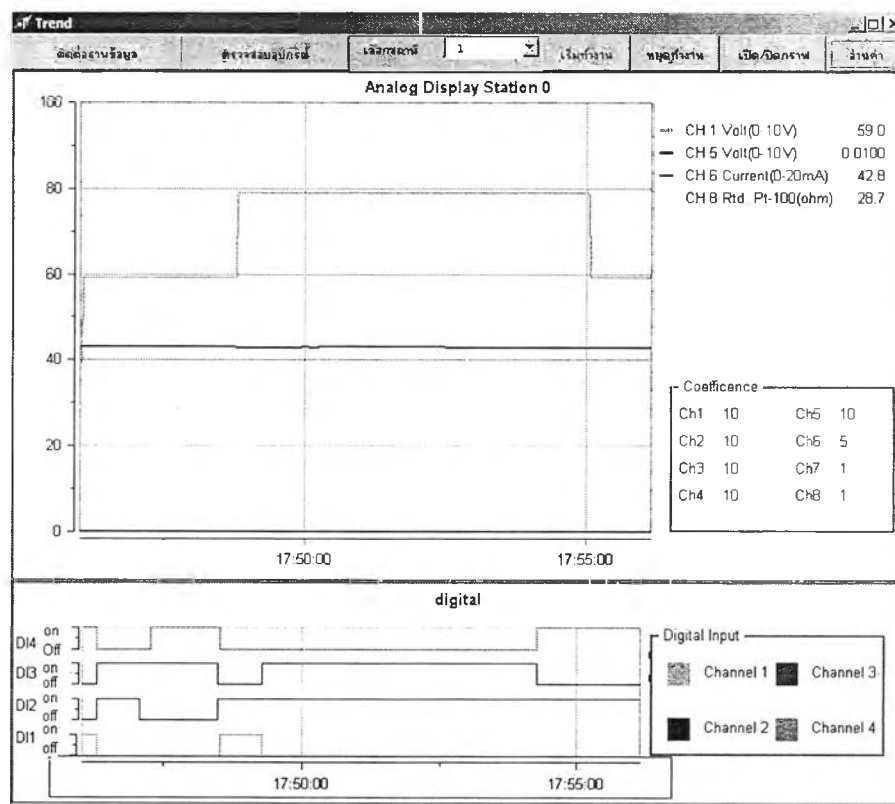
รูปที่ 4.32 ผลการทดสอบการแสดงผลในรูปแบบของกราฟของส่วนประสานงานกับผู้ใช้บนระบบปฏิบัติการลินุกซ์

- การแสดงผลบนเครื่องโคลเนทที่ใช้ระบบปฏิบัติการไมโครซอฟต์วินโดว์ เป็นการทดสอบการแสดงผลผ่านระบบเครือข่ายโดยเป็นการอ่านค่าจากหน่วยบริการฐานข้อมูลจากเครื่องสถานีหลักเช่นเดียวกับการแสดงผลบนระบบปฏิบัติการลินุกซ์ ดังแสดงในรูปที่ 4.33 - 4.34



รูปที่ 4.33 ผลการทดสอบการแสดงผลในรูปแบบทั่วไปของส่วนประสานงานกับผู้ใช้บนระบบปฏิบัติการวินโดว์ 2000

จากผลการทดสอบส่วนการแสดงผลเทียบเวลาจริงทั้งบนระบบปฏิบัติการวินโดวส์และระบบปฏิบัติการลินุกซ์ผลที่ได้คือสามารถแสดงผลได้อย่างถูกต้องกับเมื่อเทียบกับความเปลี่ยนแปลงขณะเวลาจริงของเครื่องสถานีหลัก



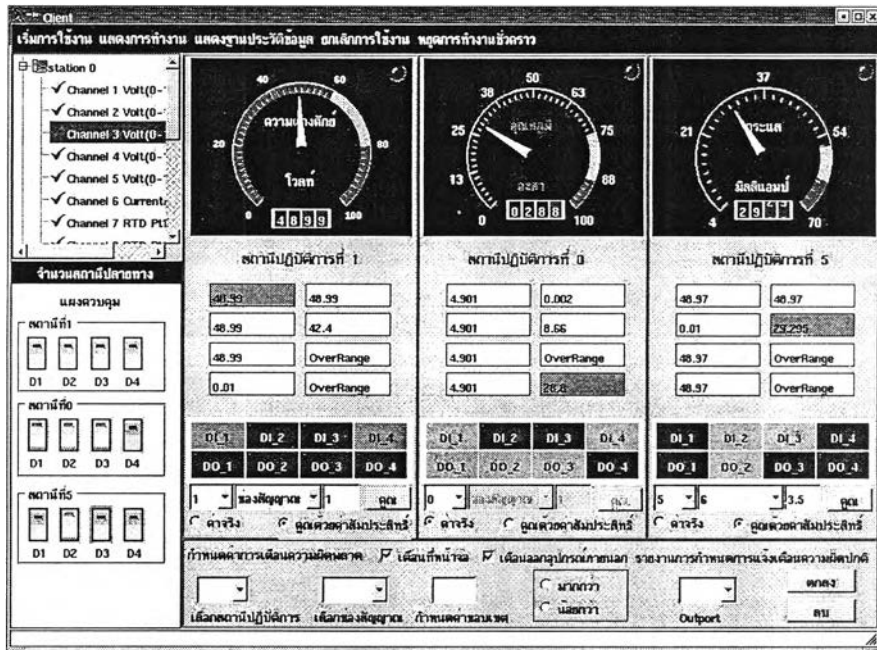
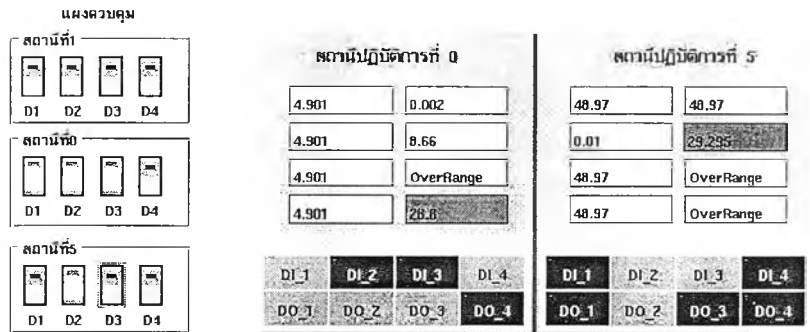
รูปที่ 4.34 ผลการทดสอบการแสดงผลในรูปแบบของกราฟของส่วนประสานงานกับผู้ใช้บนระบบปฏิบัติการวินโดวส์ 2000

4.3.1.3 การทดสอบการควบคุมระยะไกลเทียบกับเวลาจริง จากโครงสร้างการติดตั้งระบบฮาร์ดแวร์เพื่อทดสอบการทำงานของระบบสกาดาบระบบปฏิบัติการลินุกซ์ในรูปที่ 4.15 การทดสอบในหัวข้อนี้จะเป็นการทดสอบคำสั่งในการควบคุม RTU จากส่วนการประสานงานกับผู้ใช้ (User Interface) ของฝั่งไคลเอนท์ซึ่งทำงานบนโพรโตคอล TCP/IP ในรูปแบบไคลเอนท์/เซิร์ฟเวอร์ ซึ่งถือว่าการทำงานในระยะนี้ได้รูปแบบหนึ่งโดยจะแบ่งการทดสอบได้เป็นสองส่วนคือ

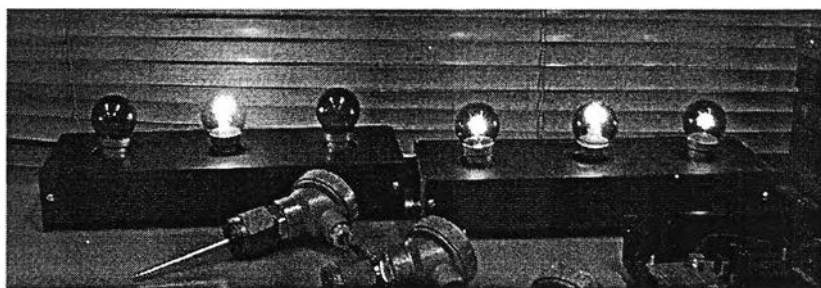
- การทดสอบควบคุมโดยตรง คือการใช้คำสั่งโดยตรงจากส่วนการประสานงานของไคลเอนท์ส่งไปยังสถานีหลักเพื่อประมวลผลและปฏิบัติตามคำสั่งที่ส่งมา อย่างเช่น



สถานีปฏิบัติการระยะไกลที่ 0 : สั่งเปิดไฟดวงที่ 1,2,3  
 สถานีปฏิบัติการระยะไกลที่ 5 : สั่งเปิดไฟดวงที่ 2 ปิด ดวงที่ 1,3  
 ผลที่ได้จากการควบคุมแสดงได้ดัง รูปที่ 4.35 และ 4.36



รูปที่ 4.35 ผลการทดสอบการควบคุมและรายงานผลจากเครื่องโคลนที่ไปยัง RTU



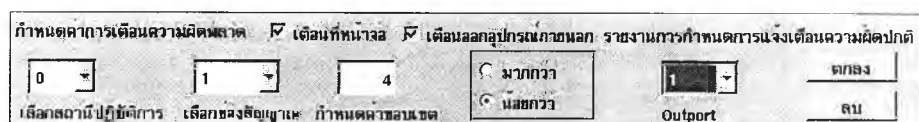
รูปที่ 4.36 ผลการทดสอบการควบคุมจากเครื่องโคลนที่ไปยัง RTU

- การทดสอบการควบคุมโดยผ่านการกำหนดค่าการแจ้งเตือนความผิดปกติจากส่วนการประสานงานของไคลเอนท์ทดสอบโดยการกำหนดค่าการแจ้งเตือนความผิดปกติดังต่อไปนี้

สถานที่ 0 :

- : ถ้าค่าของสัญญาณแอนะล็อกที่อ่านเข้ามามีค่าน้อยกว่า 4 โวลต์ ให้แสดงผลเตือนที่หน้าจอเป็นสีเหลือง และหลอดไฟสีน้ำเงิน เปิด
- : ถ้าค่าของสัญญาณแอนะล็อกที่อ่านเข้ามามีค่ามากกว่า 5.5 โวลต์ ให้แสดงผลเตือนที่หน้าจอเป็นสีเหลือง และหลอดไฟสีแดง เปิด
- : ถ้าค่าของสัญญาณแอนะล็อกที่อ่านเข้ามามีค่าอยู่ระหว่าง 4 - 5.5 โวลต์ ให้แสดงผลเตือนที่หน้าจอเป็นสีขาว และหลอดไฟปิดทุกสี

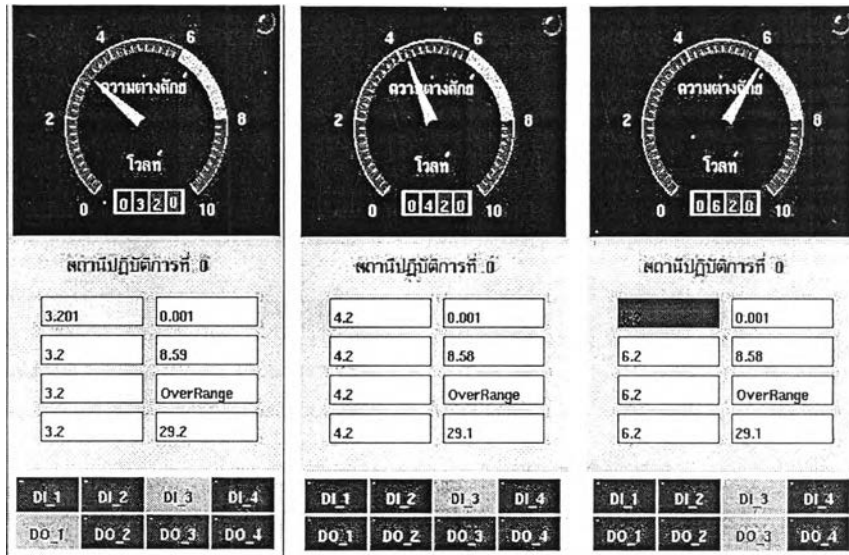
ผลจากการทดสอบแสดงดังรูปที่ 4.37 - 4.42



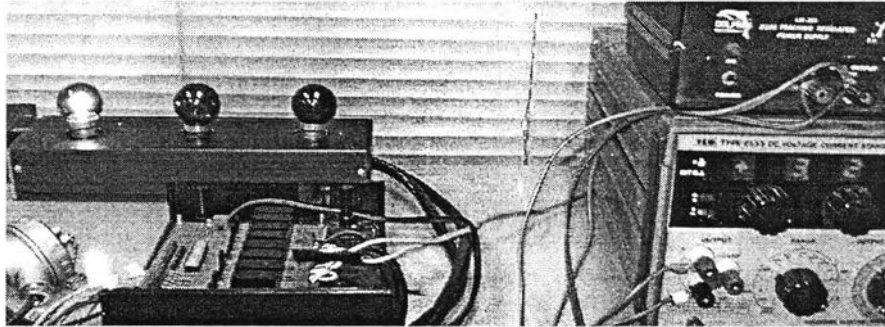
รูปที่ 4.37 การกำหนดค่าการแจ้งเตือนความผิดปกติ

สถานีที่ 0	สถานีที่ 1	สถานีที่ 2	สถานีที่ 3	สถานีที่ 4	สถานีที่ 5	สถานีที่ 6	สถานีที่ 7	สถานีที่ 8
สถานีที่ 0	Label Low	Label High	Output Low	Output High	Value Low	Value High		
ขงสัญญาณที่ 1	Used	Used	1	3	4	5.5		
ขงสัญญาณที่ 2	-	-	-	-	-	-		
ขงสัญญาณที่ 3	-	-	-	-	-	-		
ขงสัญญาณที่ 4	-	-	-	-	-	-		
ขงสัญญาณที่ 5	-	-	-	-	-	-		
ขงสัญญาณที่ 6	-	-	-	-	-	-		
ขงสัญญาณที่ 7	-	-	-	-	-	-		
ขงสัญญาณที่ 8	-	-	-	-	-	-		

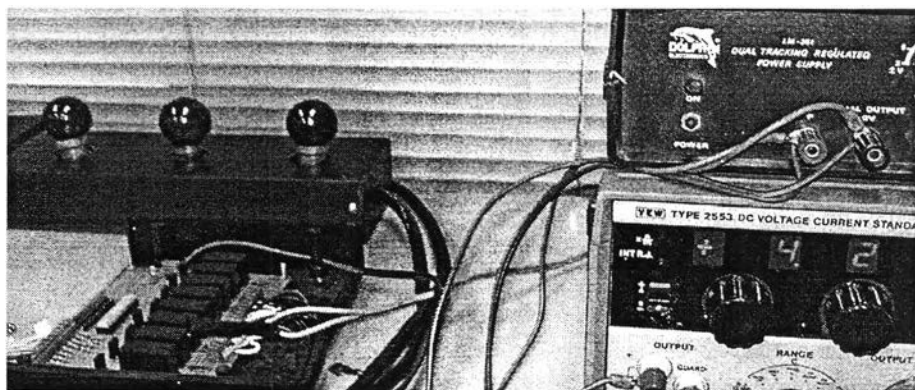
รูปที่ 4.38 การกำหนดค่าการแจ้งเตือนความผิดปกติ (ต่อ)



รูปที่ 4.39 การเตือนความผิดปกติผ่านส่วนประสานงานกับผู้ใช้ของโคลเนท

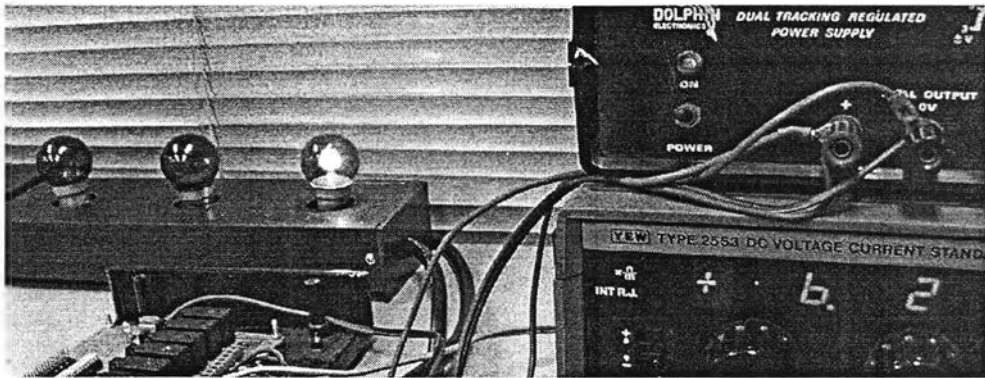


รูปที่ 4.40 การเตือนความผิดปกติผ่านทาง RTU ในกรณีที่สัญญาณ  
แอนะล็อกที่อ่านเข้ามามีค่าน้อยกว่า 4 โวลต์



รูปที่ 4.41 การเตือนความผิดปกติผ่านทาง RTU ในกรณีที่สัญญาณ  
แอนะล็อกที่อ่านเข้ามามีค่าอยู่ระหว่าง 4 - 5.5 โวลต์





รูปที่ 4.42 การเตือนความผิดปกติผ่านทาง RTU ในกรณีที่สัญญาณ  
แอนะล็อกที่อ่านเข้ามามีค่ามากกว่า 5.5 โวลต์

ผลจากการทดสอบคำสั่งในการควบคุม RTU จากส่วนการประสานงานกับผู้ใช้ (User Interface) ของฝั่งไคลเอนท์ซึ่งทำงานบนโพรโตคอล TCP/IP สามารถสรุปได้ว่าจะสามารถตอบสนองคำสั่งการทำงานจากเครื่องไคลเอนท์ได้จริงโดยที่ช่วงระยะเวลาในการปฏิบัติตามคำสั่ง 1 ชุดคำสั่งจะเท่ากับช่วงเวลาที่กำหนดให้มีการอ่านข้อมูลจาก RTU ของสถานีหลัก

4.3.1.4 การทดสอบการบันทึกข้อมูลเทียบกับเวลาจริง ในหัวข้อนี้จะแบ่งการทดสอบได้เป็นสองส่วนคือ

- การบันทึกข้อมูลเทียบกับเวลาจริงของระบบเซิร์ฟเวอร์ สามารถตรวจสอบได้โดยการดึงค่าจากฐานข้อมูลมาแสดงผลในส่วนประสานงานผู้ใช้ของเซิร์ฟเวอร์ ซึ่งเป็นการอ่านค่าจากฐานข้อมูล 2 ประเภทคือ Data Onlines เป็นฐานข้อมูลประเภทชั่วคราวที่ใช้สำหรับแสดงผลขณะเวลาจริง และฐานข้อมูลประเภท Data Storage ที่เป็นฐานข้อมูลประวัติ แสดงได้ดังรูปที่ 4.43 - 4.45

Server

Hardware\_Devices

- Station 0
- Station 1
- Station 5

Time Setting

- Auto
- 15 Sec
- 30 Sec
- 1 min
- 2 min
- 5 min

Configuration

Connect Run

Scan\_Station Stop

Exit

back up text

del data

test command

DATA\_ON\_LINE

ID	STATION	Adatetimes	A1	A2	A3	A4	A5	A6	A7	A8
2238	0	03/09/03 06:10:41 PM	6.900	6.900	6.900	6.900	0.001	8.58	0	28.6
2239	1	03/09/03 06:10:41 PM	6.898	6.898	6.898	0.001	6.898	8.44	0	0
2240	5	03/09/03 06:10:41 PM	6.896	0.001	6.896	6.895	6.896	8.30	0	0

STATION\_T datetimes\_T

STATION_T	datetimes_T	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	COF1	COF2	COF3	COF4	COF5	COF6
0	03/09/03 05:15:08 PM	11	11	11	11	12	08	08	10	10	10	10	10	10	5

ID\_Command STATION\_C STATUS\_C datetimes\_C

ID_Command	STATION_C	STATUS_C	datetimes_C	Active_DO1	Active_DO2	Active_DO3	Active_DO4	Set

DATA\_STORAGE

Dates	Times	Station_Num	Analog_In1	Ana1	Dates	Station_Num	Analog_In1	Ana1
04/06/03	11:59:40 AM	5	N	0.	03/09/03 06:10:41 PM	5	6.896	0.001

ID\_Command STATION\_C STATUS\_C datetimes\_C

ID_Command	STATION_C	STATUS_C	datetimes_C	STATION_T	datetimes_T	AT1	AT2	AT3	AT4
231	0	A	03/09/03 06:06:06	5	16/07/03 12:32:47 PM	00	11	11	11

รูปที่ 4.43 การอ่านข้อมูลจากฐานข้อมูลของส่วนประสานงานผู้ใช้ของระบบเซิร์ฟเวอร์

DATA\_ON\_LINE

ID	STATION	Adatetimes	A1	A2	A3	A4	A5	A6	A7	A8
2238	0	03/09/03 06:10:41 PM	6.900	6.900	6.900	6.900	0.001	8.58	0	28.6
2239	1	03/09/03 06:10:41 PM	6.898	6.898	6.898	0.001	6.898	8.44	0	0
2240	5	03/09/03 06:10:41 PM	6.896	0.001	6.896	6.895	6.896	8.30	0	0

STATION\_T datetimes\_T

STATION_T	datetimes_T	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	COF1	COF2	COF3	COF4	COF5	COF6
0	03/09/03 05:15:08 PM	11	11	11	11	11	12	08	08	10	10	10	10	10	5

ID\_Command STATION\_C STATUS\_C datetimes\_C

ID_Command	STATION_C	STATUS_C	datetimes_C	Active_DO1	Active_DO2	Active_DO3	Active_DO4	Set

รูปที่ 4.44 การอ่านข้อมูลจากฐานข้อมูลประเภทชั่วคราวที่ใช้สำหรับแสดงผลขณะเวลาจริง

DATA_STORAGE									
Dates	Times	Station_Num	Analog_In1	Analog_In2	Dates	Times	Station_Num	Analog_In1	Analog_In2
04/06/03	11:59:40 AM	5	N	0.001	03/09/03	06:10:41 PM	5	6.896	0.001

ID_Command	STATION_C	STATUS_C	datetimes_C	STATION_T	datetimes_T	AT1	AT2	AT3	AT4
231	0	A	03/09/03 06:00	5	16/07/03 12:32:47 PM	00	11	11	11

รูปที่ 4.45 การอ่านข้อมูลจากฐานข้อมูลประวัติขณะบันทึกที่เวลาจริง

- การอ่านข้อมูลประวัติ เป็นการทดสอบที่ส่วนประสานงานผู้ใช้ของโคเลนท์ โดยเป็นการอ่านข้อมูลผ่านทางระบบเซิร์ฟเวอร์ จุดประสงค์เพื่อนำข้อมูลที่ได้มาแสดงผลในรูปของรายงานหรือดึงออกมาเป็นแฟ้มเอกสาร แสดงได้ดังรูปที่ 4.46

The screenshot shows a software window titled 'History' with a table of data and a configuration panel below it.

STATION_T	datetimes_T	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	COF1	COF2	COF3	COF4	COF5	COF6	COF7	COF8
5	16/7/2003 12:32:47	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	17/7/2003 14:15:44	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	18/7/2003 1:31:19	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	18/7/2003 1:32:28	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	18/7/2003 1:33:37	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	18/7/2003 1:37:43	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 9:36:58	T	T	T	T	T	T	T	T	1	10	10	10	10	5	1	1
5	21/7/2003 9:54:35	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 9:55:32	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 10:12:17	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 11:31:50	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 13:24:36	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 13:30:42	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 13:35:26	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 13:43:24	00	11	11	11	11	12	08	08	1	10	10	10	10	5	1	1
5	21/7/2003 15:03:20	T	T	T	T	T	T	T	T	1	10	10	10	10	5	1	1

กำหนดค่า

ช่องสัญญาณที่ 1 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 2 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 3 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 4 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 5 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 6 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 7 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>
ช่องสัญญาณที่ 8 รหัส	code	ทำมีประวัติทั้งหมด	<input type="checkbox"/>

ใช้ทำมีประวัติทั้งหมดและค้น  
เลือกภาพ

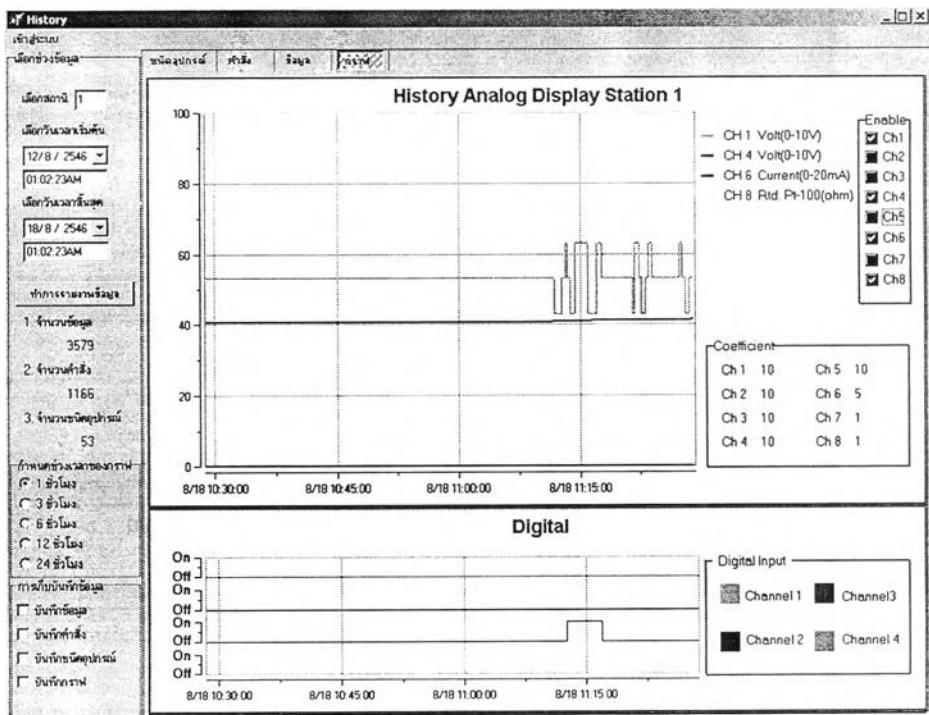
รูปที่ 4.46 การอ่านข้อมูลประเภทชนิดอุปกรณ์จากฐานข้อมูลประวัติ

ID_Command	STATION_C	STATUS_C	dateimes_C	Active_D01	Active_D02	Active_D03	Active_D04
2 1	A		12/8/2003 22:25:09	0	0	1	0
2 1	A		12/8/2003 22:25:14	0	0	0	0
2 1	A		12/8/2003 22:26:03	0	1	0	0
2 1	A		12/8/2003 22:26:09	0	0	0	0
2 1	A		13/8/2003 0:30:41	0	0	1	0
2 1	A		13/8/2003 0:30:42	0	1	1	0
2 1	A		13/8/2003 0:31:05	0	0	1	0
2 1	A		13/8/2003 0:31:12	0	1	1	0
1 1	A		13/8/2003 0:32:39	1	0	0	0

รูปที่ 4.46 (ต่อ) แสดงการอ่านข้อมูลประเภทคำสั่งจากรฐานข้อมูลประวัติ

Dates	Station Num	Analog_In1	Analog_In2	Analog_In3	Analog_In4	Analog_In5	Analog_In6	Analog_In7	Analog_In8	Digital_In1	Digital_In2
12/8/2003 22:24:17	1	6.000	5.999	5.999	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:23	1	5.999	5.999	5.999	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:28	1	5.999	5.999	5.999	-0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:34	1	5.999	5.999	5.999	0.001	5.999	8.13	0	0	0	0
12/8/2003 22:24:39	1	5.999	5.999	5.999	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:45	1	6.000	6.000	6.000	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:50	1	5.999	5.999	5.999	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:24:56	1	5.999	5.999	5.999	0.000	5.999	8.13	0	0	0	0
12/8/2003 22:25:01	1	5.999	5.999	5.999	0.000	5.999	8.14	0	0	0	0
12/8/2003 22:25:07	1	5.999	5.999	5.999	0.000	5.999	8.14	0	0	0	0

รูปที่ 4.46 (ต่อ) แสดงการอ่านข้อมูลประเภทข้อมูลแอนะล็อกและดิจิตอลจากรฐานข้อมูลประวัติ



รูปที่ 4.46 (ต่อ) แสดงการอ่านข้อมูลประเภทข้อมูลแอนะล็อกและดิจิตอลจากรฐานข้อมูลประวัติในรูปแบบของกราฟ

4.3.1.5 การทดสอบระบบไคลเอนต์/เซิร์ฟเวอร์ เป็นการทดสอบการทำงานของส่วนการประสานงานกับผู้ใช้ในฝั่งไคลเอนต์ถึงความสามารถในการทำงานได้ในระบบหลายผู้ใช้ (Multi - User) โดยใช้เครื่องคอมพิวเตอร์ในการทดสอบการทำงาน 4 เครื่อง ปรากฏว่าส่วนประสานงานผู้ใช้ของไคลเอนต์สามารถทำงานได้เป็นปกติและโดยสมมติฐานของระบบหลายผู้ใช้น่าจะใช้งานในส่วนประสานงานผู้ใช้ของไคลเอนต์ได้มากกว่าจำนวน 4 เครื่องที่นำมาทดสอบ

ตารางที่ 4.5 รหัสและความหมายของการรายงานการใช้ทรัพยากรบนระบบปฏิบัติการลินุกซ์

CODE	CODE DESCRIPTIONS
uptime	This line displays the time the system has been up, and the three load averages for the system
processes	The total number of processes running at the time of the last update . This is also broken down into the number of tasks which are running, sleeping, stopped, or undead
CPU states	Shows the percentage of CPU time in user mode, system mode, niced tasks, and idle
Mem	Statistics on memory usage, including total available memory, free memory, used memory, shared memory, and memory used for buffers
Swap	Statistics on swap space, including total swap space, available swap space, and used swap space
PID	The process ID of each task
USER	The user name of the task's owner
PRI	The priority of the task
NI	The nice value of the task . Negative nice values are higher priority
SIZE	The size of the task's code plus data plus stack space, in kilobytes, is shown here
RSS	The total amount of physical memory used by the task, in kilobytes, is shown here
SHARE	The amount of shared memory used by the task is shown in this column
STAT	The state of the task is shown here . The state is either S for sleeping, D for uninterruptible sleep, R for running, Z for zombies, or T for stopped or traced ., N for a process with positive nice value, W for a swapped out process .(this does not work correctly for kernel processes )
%CPU	The task's share of the CPU time since the last screen update, expressed as a percentage of total CPU time per processor
%MEM	The task's share of the physical memory
TIME	Total CPU time the task has used since it started
COMMAND	The task's command name, which will be truncated if it is too long to be displayed on one line
	Tasks in memory will have a full command line, but swapped -out tasks will only have the name of the program in parentheses

4.3.2 ตรวจสอบการใช้ทรัพยากรของซอฟต์แวร์สกาตา ในส่วนนี้เป็นการรายงานการใช้ทรัพยากรของซอฟต์แวร์สกาตาบนระบบปฏิบัติการลินุกซ์ในส่วนระบบการประสานงานกับผู้ใช้

ของระบบไคลเอนท์/เซิร์ฟเวอร์ และระบบไคลเอนท์ค่าที่อ่านได้เป็นค่าที่บ่งบอกความสามารถของระบบปฏิบัติการ ลีนุกซ์ในการบริหารและจัดการ การใช้ทรัพยากรสำหรับส่วนการประยุกต์ใช้งานที่เพิ่มเข้ามาในระบบการทำงาน และเพื่อความเข้าใจในความหมายของค่าต่างๆที่จำเป็นของรายงานจึงต้องทราบความหมายโดยสังเขปของรหัสของการรายงานการใช้ทรัพยากรบนระบบปฏิบัติการ ลีนุกซ์ แสดงดังตารางที่ 4.5 และรายละเอียดของรายงานแสดงเป็นลำดับดังนี้

1. รายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตายังไม่เริ่มทำงาน เป็นรายงานบอกการใช้ทรัพยากรเบื้องต้นก่อนที่ระบบสกาตาจะทำงาน มีรายละเอียดทั่วไปแสดงดังรูปที่ 4.47

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
1157	root	15	0	41928	8692	1816	S	10.5	1.3	0:02	X
1819	root	16	0	9628	9628	8240	S	6.1	1.4	0:00	ksnapshot
1092	xf	10	0	4104	4104	1056	S	0.5	0.6	0:00	xf
238	root	10	0	0	0	0	SW	0.1	0.0	0:00	kjournald
1634	root	9	0	5252	5252	5024	S	0.1	0.8	0:00	kdeinit
1669	root	15	0	5100	5100	4984	S	0.1	0.7	0:00	kdeinit
1671	root	9	0	7992	7992	7356	S	0.1	1.2	0:00	kdeinit
1673	root	9	0	9896	9896	8988	S	0.1	1.5	0:00	kdeinit
1675	root	9	0	10524	10M	9100	S	0.1	1.6	0:00	kdeinit
1818	root	9	0	1064	1064	836	R	0.1	0.1	0:00	top
1821	root	17	0	5900	5900	5652	S	0.1	0.9	0:00	kdeinit
1	root	8	0	520	520	452	S	0.0	0.0	0:04	init
2	root	8	0	0	0	0	SW	0.0	0.0	0:00	keventd
3	root	9	0	0	0	0	SW	0.0	0.0	0:00	kpm-idled
4	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SW	0.0	0.0	0:00	kswapd
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kreclamd
7	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdflush
8	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
9	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	mdrecoveryd
13	root	9	0	0	0	0	SW	0.0	0.0	0:00	kjournald
79	root	9	0	0	0	0	SW	0.0	0.0	0:00	khud

รูปที่ 4.47 การรายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตายังไม่เริ่มทำงาน

2. รายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตาเริ่มทำงานทั้งระบบคือในส่วนระบบการประสานงานกับผู้ใช้ของตัวเซิร์ฟเวอร์และตัวไคลเอนท์มีรายละเอียดทั่วไปแสดงดังรูปที่ 4.48

```

Konsole - root@localhost:~ - Konsole
File Sessions Settings Help

1:03am up 42 min, 4 users, load average: 0.48, 0.22, 0.08
92 processes: 88 sleeping, 4 running, 0 zombie, 0 stopped
CPU states: 27.3% user, 12.9% system, 0.0% nice, 59.6% idle
Mem: 642192K av, 324864K used, 317328K free, 172K shrd, 59340K buff
Swap: OK av, OK used, OK free 88080K cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT   %CPU   %MEM    TIME COMMAND
 1789 root        16   0 23340  22M 5188 R    39.3   3.6   1:19 MainServer
 1157 root         9   0 43576  10M 1980 S     0.1   1.6    0:07 X
 1673 root         9   0 9896 9896 8988 S     0.1   1.5    0:00 kdeinit
 1738 root         9   0 12588  12M 5776 S     0.1   1.9    0:01 Monitor
 1818 root        10   0 1076 1076 836 R     0.1   0.1    0:00 top
   1 root         8   0   520   520 452 S     0.0   0.0    0:04 init
   2 root         8   0   0     0   0 SW     0.0   0.0    0:00 keventd
   3 root         9   0   0     0   0 SW     0.0   0.0    0:00 kapa-idled
   4 root        19  19   0     0   0 SWN    0.0   0.0    0:00 ksoftirqd_CPU0
   5 root         9   0   0     0   0 SW     0.0   0.0    0:00 kswapd
   6 root         9   0   0     0   0 SW     0.0   0.0    0:00 kreclaimd
   7 root         9   0   0     0   0 SW     0.0   0.0    0:00 bdflush
   8 root         9   0   0     0   0 SW     0.0   0.0    0:00 kupdated
   9 root        -1 -20   0     0   0 SW<    0.0   0.0    0:00 mdrecoveryd
  13 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
  79 root         9   0   0     0   0 SW     0.0   0.0    0:00 khubd
 237 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 238 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 239 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 566 root         9   0   0     0   0 SW     0.0   0.0    0:00 eth0
 567 root         9   0 836 836 716 S     0.0   0.1    0:00 dhcpcd
 639 root         9   0 592 592 496 S     0.0   0.0    0:00 syslogd

```

รูปที่ 4.48 การรายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตาเริ่มทำงานทั้งระบบ

- รายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตาทำงานทั้งระบบ และมีการอ่านข้อมูลจาก RTU 3 สถานี จำนวน 3,000 ครั้ง มีรายละเอียดทั่วไปแสดงดังรูปที่ 4.49

```

Konsole - root@localhost:~ - Konsole
File Sessions Settings Help

2:29am up 2:07, 4 users, load average: 0.53, 0.47, 0.45
93 processes: 90 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 26.3% user, 13.7% system, 0.0% nice, 59.8% idle
Mem: 642192K av, 330968K used, 311204K free, 172K shrd, 63204K buff
Swap: OK av, OK used, OK free 89088K cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT   %CPU   %MEM    TIME COMMAND
 1789 root        15   0 24400  23M 5188 R    39.5   3.7   39:50 MainServer
 1738 root         9   0 12804  12M 5812 S     0.1   1.9    0:12 Monitor
 1818 root        10   0 1080 1080 836 R     0.1   0.1    0:10 top
 1830 mysql         9   0 5760 5760 2124 S     0.1   0.8    0:03 mysqld
   1 root         8   0   520   520 452 S     0.0   0.0    0:04 init
   2 root         8   0   0     0   0 SW     0.0   0.0    0:00 keventd
   3 root         9   0   0     0   0 SW     0.0   0.0    0:00 kapa-idled
   4 root        19  19   0     0   0 SWN    0.0   0.0    0:00 ksoftirqd_CPU0
   5 root         9   0   0     0   0 SW     0.0   0.0    0:00 kswapd
   6 root         9   0   0     0   0 SW     0.0   0.0    0:00 kreclaimd
   7 root         9   0   0     0   0 SW     0.0   0.0    0:00 bdflush
   8 root         9   0   0     0   0 SW     0.0   0.0    0:00 kupdated
   9 root        -1 -20   0     0   0 SW<    0.0   0.0    0:00 mdrecoveryd
  13 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
  79 root         9   0   0     0   0 SW     0.0   0.0    0:00 khubd
 237 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 238 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 239 root         9   0   0     0   0 SW     0.0   0.0    0:00 kjournald
 566 root         9   0   0     0   0 SW     0.0   0.0    0:00 eth0
 567 root         9   0 836 836 716 S     0.0   0.1    0:00 dhcpcd
 639 root         9   0 592 592 496 S     0.0   0.0    0:00 syslogd
 644 root         9   0 1128 1128 448 S     0.0   0.1    0:00 klogd

```

รูปที่ 4.49 การรายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตามีการอ่านข้อมูลจาก RTU 3 สถานี จำนวน 3,000 ครั้ง

4. รายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตาดูติการทำงาน เป็นการแสดงถึงการคืนทรัพยากรทันทีหลังจากซอฟต์แวร์สกาตาดูติการทำงาน มีรายละเอียดทั่วไปแสดงดังรูปที่ 4.50

PID	USER	PRI	NI	SIZE	RSS	SHSS	SINT	%CPU	%MEM	TIME	COMMAND
1790	root	17	0	8676	8676	7800	R	0.1	1.3	0:00	kdenit
1818	root	16	0	1080	1080	836	R	0.1	0.1	0:11	top
1	root	8	0	520	520	452	S	0.0	0.0	0:04	init
2	root	8	0	0	0	0	SH	0.0	0.0	0:00	keventd
3	root	9	0	0	0	0	SH	0.0	0.0	0:00	ktop-idled
4	root	19	19	0	0	0	SH	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SH	0.0	0.0	0:00	kswed
6	root	9	0	0	0	0	SH	0.0	0.0	0:00	kreclaimd
7	root	9	0	0	0	0	SH	0.0	0.0	0:00	bdflush
8	root	9	0	0	0	0	SH	0.0	0.0	0:00	kupdated
9	root	-1	-20	0	0	0	SH	0.0	0.0	0:00	mdrecoveryd
13	root	9	0	0	0	0	SH	0.0	0.0	0:00	kjournald
79	root	9	0	0	0	0	SH	0.0	0.0	0:00	khubd
237	root	9	0	0	0	0	SH	0.0	0.0	0:00	kjournald
238	root	9	0	0	0	0	SH	0.0	0.0	0:00	kjournald
239	root	9	0	0	0	0	SH	0.0	0.0	0:00	kjournald
566	root	9	0	0	0	0	SH	0.0	0.0	0:00	eth0
567	root	9	0	836	836	716	S	0.0	0.1	0:00	dhcpcd
639	root	9	0	592	592	496	S	0.0	0.0	0:00	syslogd
644	root	9	0	1128	1128	448	S	0.0	0.1	0:00	klogd
662	rpc	9	0	636	636	544	S	0.0	0.0	0:00	portmap
690	rpcuser	9	0	798	798	688	S	0.0	0.1	0:00	rpc.statd
797	root	8	0	520	520	464	S	0.0	0.0	0:00	apwd
854	root	9	0	1268	1268	1076	S	0.0	0.1	0:00	sashd
887	root	9	0	1052	1036	852	S	0.0	0.1	0:00	xinetd
920	root	9	0	1028	1028	856	S	0.0	0.1	0:00	safe_mysql
960	mysql	9	0	5260	5260	2124	S	0.0	0.8	0:00	mysqld
964	mysql	8	0	5260	5260	2124	S	0.0	0.8	0:00	mysqld
965	mysql	9	0	5260	5260	2124	S	0.0	0.8	0:00	mysqld
975	root	9	0	2012	2012	1460	S	0.0	0.3	0:00	sendmail
983	mysql	9	0	5260	5260	2124	S	0.0	0.8	0:00	mysqld
995	root	9	0	488	488	424	S	0.0	0.0	0:00	gpm
1018	root	9	0	1604	1604	1488	S	0.0	0.2	0:00	httd
1044	root	8	0	698	698	600	S	0.0	0.1	0:00	crond
1092	xfs	9	0	4164	4164	1056	S	0.0	0.6	0:00	xfs
1128	daemon	9	0	568	568	500	S	0.0	0.0	0:00	atd
1142	root	9	0	448	448	384	S	0.0	0.0	0:00	mingetty
1143	root	9	0	448	448	384	S	0.0	0.0	0:00	mingetty
1144	root	9	0	448	448	384	S	0.0	0.0	0:00	mingetty
1145	root	9	0	448	448	384	S	0.0	0.0	0:00	mingetty
1146	root	9	0	448	448	384	S	0.0	0.0	0:00	mingetty

รูปที่ 4.50 การรายงานการบริหารการใช้ทรัพยากรขณะซอฟต์แวร์สกาตาดูติการทำงาน

จากนั้นนำผลที่ได้มาเปรียบเทียบกันโดยใช้เพียงบางค่าเท่านั้นผลที่สังเกตได้คือ ขณะที่เริ่มและยุติการใช้งานซอฟต์แวร์สกาตาดูติในขั้นตอนที่ 2 และ 4 มีค่าการใช้หน่วยความจำ (MEM) ต่างกันน้อยมากทั้งที่ขั้นตอนที่ 3 หรือขณะซอฟต์แวร์สกาตาดูติทำงานอยู่มีค่าการใช้หน่วยความจำต่างกันอย่างมากซึ่งชัดเจนรวมถึงการทำงานของหน่วยประมวลผลกลาง (%CPU) ของซอฟต์แวร์สกาตาดูติจะหยุดทันทีเมื่อหยุดทำงานนอกจากนั้นยังมีค่าอื่นอีกหลายค่าที่บ่งบอกความสามารถของระบบปฏิบัติการลินุกซ์ในการจัดการทรัพยากรของระบบอย่างมีประสิทธิภาพ แสดงได้ดังตารางที่ 4.6

ความมีประสิทธิภาพในการบริหารการใช้ทรัพยากรของระบบปฏิบัติการลินุกซ์ที่มีต่อการใช้งานซอฟต์แวร์สกาตาดูติ ส่งผลให้แนวโน้มการพัฒนาระบบสกาตาดูติระบบปฏิบัติการลินุกซ์ง่ายขึ้นด้วยเพราะนักพัฒนาไม่ต้องกังวลมากถึงเรื่องการจัดการทรัพยากรของระบบโดยปล่อยให้ทำหน้าที่ของระบบปฏิบัติการคอยจัดการให้แทน



ตารางที่ 4.6 การเปรียบเทียบการใช้ทรัพยากรของระบบขณะใช้งานและยุติการใช้งาน

ขั้นตอนการทำงาน	MEM	COMMAND	PID	PRI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME
ซอฟต์แวร์สกาตายังไม่เริ่มทำงาน	306420K	MainServer No	No	No	No	No	No	No	No	No	No
		Monitor No	No	No	No	No	No	No	No	No	No
ซอฟต์แวร์สกาตาเริ่มทำงานทั้งระบบ	324864K	MainServer Yes	1789	16	23340	22M	5188	R	39.3	3.6	1:19
		Monitor Yes	1738	9	12588	12M	5776	S	0.1	1.9	0:01
มีการอ่านข้อมูลจากสถานีปลายทาง ระยะไกล 3 สถานี จำนวน 3,000 ครั้ง	330988K	MainServer Yes	1789	15	24400	23M	5188	R	39.5	3.7	39:50
		Monitor Yes	1738	9	12804	12M	5812	S	0.1	1.9	0:12
ซอฟต์แวร์สกาตายุติการทำงาน	323500K	MainServer No	No	No	No	No	No	No	No	No	No
		Monitor No	No	No	No	No	No	No	No	No	No

4.3.3 การคำนวณหน่วยความจำที่ใช้สำหรับการบันทึกข้อมูล ในซอฟต์แวร์สกาตาที่ได้  
ออกแบบไว้มีการเก็บข้อมูลในรูปแบบตารางของฐานข้อมูลอยู่ 3 ประเภท และมีการเก็บข้อมูลต่อ  
1 รอบ ที่เรียกใช้งานมีค่าดังแสดงในตารางที่ 4.6 – 4.7

ตารางที่ 4.7 การคำนวณพื้นที่ในการเก็บข้อมูลสำหรับอินพุตแอนะล็อกและดิจิตอล

Data Name	Name Field	Data Type	Maximum Memory
Data Index	ID	Unsigned Integer	4 byte
Name Station	Station_Num	Unsigned Tiny Integer	1 byte
Datetime	Datetime	Datetime	8 byte
Analog Input 1	A1	VARCHAR	8 byte
Analog Input 2	A2	VARCHAR	8 byte
Analog Input 3	A3	VARCHAR	8 byte
Analog Input 4	A4	VARCHAR	8 byte
Analog Input 5	A5	VARCHAR	8 byte
Analog Input 6	A6	VARCHAR	8 byte
Analog Input 7	A7	VARCHAR	8 byte
Analog Input 8	A8	VARCHAR	8 byte
Digital Input 1	DI1	Character	1 byte
Digital Input 2	DI2	Character	1 byte
Digital Input 3	DI3	Character	1 byte
Digital Input 4	DI4	Character	1 byte
Digital Output 1	DO1	Character	1 byte
Digital Output 2	DO2	Character	1 byte
Digital Output 3	DO3	Character	1 byte
Digital Output 4	DO4	Character	1 byte
Total			85 byte

ตารางที่ 4.8 การคำนวณพื้นที่ในการเก็บข้อมูลสำหรับชุดคำสั่ง

Data Name	Name Field	Data Type	Maximum Memory
Command Index	Id_Command	Unsigned Integer	4 byte
Name Station	Station_Command	Unsigned Tiny Integer	1 byte
Command Status	Status_C	Character	1 byte
Dateime	Datetime	Datetime	8 byte
Select Channel Active	Sel	Character	1 byte
Active Digital Output Ch 1	Active_Do 1	Character	1 byte
Active Digital Output Ch 2	Active_Do 2	Character	1 byte
Active Digital Output Ch 3	Active_Do 3	Character	1 byte
Active Digital Output Ch 4	Active_Do 4	Character	1 byte
	Total		19 byte

ตารางที่ 4.9 การคำนวณพื้นที่ในการเก็บข้อมูลสำหรับชนิดของอุปกรณ์

Data Name	Name Field	Data Type	Maximum Memory
Name Station	Station_Type	Unsigned Tiny Integer	1 byte
Datetime	Datetime	Datetime	8 byte
Analog Type 1	AT1	Character	2 byte
Analog Type 2	AT2	Character	2 byte
Analog Type 3	AT3	Character	2 byte
Analog Type 4	AT4	Character	2 byte
Analog Type 5	AT5	Character	2 byte
Analog Type 6	AT6	Character	2 byte
Analog Type 7	AT 7	Character	2 byte
Analog Type 8	AT 8	Character	2 byte
Coefficient Of AT1	COF 1	Character	3 byte
Coefficient Of AT2	COF 2	Character	3 byte
Coefficient Of AT3	COF 3	Character	3 byte
Coefficient Of AT4	COF 4	Character	3 byte
Coefficient Of AT5	COF 5	Character	3 byte
Coefficient Of AT6	COF 6	Character	3 byte
Coefficient Of AT7	COF 7	Character	3 byte
Coefficient Of AT8	COF 8	Character	3 byte
	Total		49 byte

## สรุปท้ายบท

ผลจากการทดสอบระบบสภาคาบบระบบปฏิบัติการลินุกซ์ต้นแบบตามคุณสมบัติโดยสมมติฐานได้ผลดังตารางที่ 4.10

ตารางที่ 4.10 การเปรียบเทียบระหว่างคุณสมบัติสมมติฐานกับผลการทดสอบจริง

	คุณสมบัติตามสมมติฐาน	ผลการทดสอบ
1	สถานีหลักสามารถติดต่อกับ RTU ด้วยไฟร์โตคอลสื่อสารแบบมอดบัล แอสกีบนมาตรฐาน RS-485 ได้ 10 สถานี โดยที่แต่ละสถานีรับสัญญาณ แอนะล็อกได้ 8 ช่องสัญญาณ สัญญาณดิจิตอลอินพุต/เอาต์พุตอย่างละ 4 ช่องสัญญาณ	สามารถใช้งานได้จริงและน่าจะเพิ่ม ได้ถึง 32 RTU
2	เวลาดำสุดที่สถานีหลักอ่านข้อมูลจากสถานี RTU ต่อ 1 RTU เท่ากับ 1 วินาที (เวลาที่ใช้ 1 วินาทีมาจาก เวลาของ RTU ในการเก็บข้อมูลเท่ากับ 0.3 วินาที รวมกับเวลาที่ใช้ในโปรแกรมประมาณ 0.7 วินาที)	ได้เวลาดำสุดที่ข้อมูลไม่เสียหายเท่า กับ 2 วินาที (1 วินาทีที่ถูกเพิ่มขึ้น เพื่อใช้ในการตรวจสอบซ้ำในกรณีที่ ข้อมูลที่ถูกส่งกลับมาจาก RTU เกิด ข้อผิดพลาด)
3	สามารถตรวจความผิดปกติจากการสื่อสารระหว่างสถานีหลักกับ RTU ระหว่างการทำงานได้ เช่นกรณีที่สายส่งสัญญาณหลุด	ทำงานได้ตามสมมติฐาน
4	สนับสนุนการควบคุมระยะไกลจากเครื่องโคลเอนท์	ทำงานได้ตามสมมติฐาน
5	การแสดงผลสื่อความหมายได้อย่างชัดเจนในรูปแบบทั่วไปและรูปแบบ กราฟ	ทำงานได้ตามสมมติฐาน
6	สามารถแจ้งเตือนความผิดปกติในรูปแบบความเปลี่ยนแปลงของสีที่หน้า จอและอุปกรณ์ภายนอกเช่นหลอดไฟหรือสัญญาณเตือนโดยใช้พอร์ด ดิจิตอลเอาต์พุตของ RTU	ทำงานได้ตามสมมติฐาน
7	ใช้ความเร็วในการสื่อสารแบบมอดบัลแอสกีบนมาตรฐาน RS-485 เท่า กับ 19200 บิตต่อวินาที และสามารถเพิ่มได้ถึง 57600 บิตต่อวินาที	ทำงานได้ตามสมมติฐาน
8	สามารถตรวจสอบการทำงาน (Monitoring) ในรูปแบบของโคลเอนท์/ เซิร์ฟเวอร์ได้ทั้งบนระบบปฏิบัติการลินุกซ์ (RedHat 7.2) และ ไมโครซอฟต์วินโดว (Win2000)	ทำงานได้ตามสมมติฐาน
9	สามารถบันทึกข้อมูลในรูปแบบไฟล์เอกสารหรือรูปภาพเพื่อที่จะนำไปใช้ใน การทำรายงานต่อได้	ทำงานได้ตามสมมติฐาน

นอกจากนั้นจากการตรวจสอบการบริหารทรัพยากรของระบบปฏิบัติการลินุกซ์ยังพบว่ามี  
การจัดการบริหารที่ดีมีการควบคุมการคืนทรัพยากรหลังจากใช้เสร็จตลอดเวลาส่งผลให้การใช้งาน  
ซอฟต์แวร์ที่สร้างขึ้นมีเสถียรภาพมากขึ้นตามไปด้วย