บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

วิทยานิพนธ์นี้เป็นการแสดงวิธีในการสร้างและพัฒนาระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ ที่ถูกออกแบบให้สถานีหลักเป็นแบบรวมฟังก์ชันไว้ที่ศูนย์กลางและมีการติดต่อกับสถานีปฏิบัติการ (Field Site) ซึ่งเป็นแหล่งรวม RTU ที่สื่อสารกันด้วยมาตรฐาน RS-485 โดยการโพลลิ่ง (Polling) ด้วยโพรโตคอลสื่อสารแบบมอดบัสแอสกีซึ่งมีความสามารถในการตรวจสอบสถานะการทำงาน (Monitoring) ได้โดยผ่านระบบปฏิบัติการไมโครซอฟต์วินโดว์และระบบปฏิบัติการลีนุกซ์ การพัฒนาระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ในวิทยานิพนธ์ฉบับนี้มุ่งเน้นการออกแบบระบบให้มีความครบถ้วนการใช้งานต่างๆของระบบสกาดาขนาดกลาง รวมถึงค้นหาโครงสร้างและความสัมพันธ์ที่เหมาะสมของการทำงานของระบบสกาดาบนระบบปฏิบัติลีนุกซ์ ทั้งนี้ก็เพื่อให้สามารถนำระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ ไปใช้งานได้จริงในงานอุตสาหกรรม

จากการทดสอบการทำงานทุกฟังก์ชันของระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ต้นแบบที่สร้างขึ้นได้ผลการทดสอบดังต่อไปนี้

1. สถานีหลักสามารถติดต่อกับ RTU ด้วยโพรโตคอลสื่อสารแบบมอดบัสแอสกีบนมาตรฐาน RS-485 ได้ 10 RTU และขยายได้ถึง 32 RTUโดยที่แต่ละ RTU รับสัญญาณแอนะล็อก ได้ 8 ช่องสัญญาณ สัญญาณดิจิตอลอินพุต/เอาต์พุตอย่างละ 4 ช่องสัญญาณ
2. เวลาต่ำสุดที่สถานีหลักอ่านข้อมูลจาก RTU ต่อ 1 RTU โดยไม่มีการสูญเสียของข้อมูลเท่ากับ 2 วินาที
3. สามารถตรวจความผิดปกติจากการสื่อสารระหว่างสถานีหลักกับ RTU ระหว่างการทำงานได้ เช่นกรณีที่สายส่งสัญญาณหลุด
4. สามารถควบคุมระยะไกลได้จากเครื่องไคลเอนท์ (Client)
5. การแสดงผลสื่อความหมายได้อย่างชัดเจนในรูปแบบทั่วไปและรูปแบบกราฟ
6. สามารถแจ้งเตือนความผิดปกติในรูปแบบความเปลี่ยนแปลงของสีที่หน้าจอและอุปกรณ์ภายนอกเช่นหลอดไฟหรือสัญญาณเตือนโดยใช้พอร์ตดิจิตอลเอาต์พุตของ RTU

7. การใช้พื้นที่ในการเก็บข้อมูลต่อการโพลลิ่ง (Polling) 1 ครั้งต่อหนึ่ง RTU มีค่าเท่ากับ 85 ไบต์ซึ่งหมายความว่าถ้าต้องการเก็บข้อมูล 1,000,000 ครั้งของการโพลลิ่งต้องใช้จานบันทึกแบบแข็ง (Hard Disk) 85 เมกกะไบต์ (MB)
8. ใช้ความเร็วในการสื่อสารแบบมอดบัสแอสกี (MODBUS ASCII) บนมาตรฐาน RS-485 เท่ากับ 19,200 บิตต่อวินาที และสามารถเพิ่มได้ถึง 57,600 บิตต่อวินาที
9. สามารถตรวจสอบการทำงาน (Monitor) ในรูปแบบของไคลเอนท์/เซิร์ฟเวอร์ได้ทั้งบนระบบปฏิบัติการลีนุกซ์ (RedHat 7.2) และไมโครซอฟต์วินโดว์ (Win2000)
10. สามารถบันทึกข้อมูลในรูปไฟล์เอกสารหรือรูปภาพเพื่อที่จะนำไปใช้ในการทำรายงานต่อได้

ผลจากการตรวจสอบการบริหารทรัพยากรของระบบปฏิบัติการลีนุกซ์ต่อระบบสกาดาต้นแบบแล้วพบว่าระบบปฏิบัติการลีนุกซ์มีการจัดการทรัพยากรอย่างดีส่งผลให้แนวโน้มการพัฒนาระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ง่ายขึ้น เพราะนักพัฒนาไม่ต้องกังวลมาก ถึงเรื่องการจัดการทรัพยากรของระบบโดยปล่อยให้เป็นหน้าที่ของระบบปฏิบัติการคอยจัดการให้แทนนอกจากนั้นผลที่ได้จากการพัฒนาและทดสอบระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ในวิทยานิพนธ์ฉบับนี้ คือแนวความคิดและรูปแบบในการพัฒนาระบบสกาดาขนาดกลางที่สามารถใช้งานได้จริงโดยวิทยานิพนธ์ฉบับนี้ได้จำแนกเรียบเรียงลำดับขั้นการออกแบบในการสร้างและบ่งบอกสิ่งจำเป็นของระบบที่ควรต้องมีและที่สำคัญคือโครงสร้างที่ได้ออกแบบไว้สามารถนำไปประยุกต์ใช้งานได้จริงทั้งระบบปฏิบัติการลีนุกซ์และระบบปฏิบัติการวินโดว์

จากการศึกษา สร้างและทดสอบระบบสกาดาบนระบบปฏิบัติการลีนุกซ์มีจุดประสงค์เพื่อแก้ปัญหาการขาดความต่อเนื่องในการพัฒนาระบบสกาดาให้มีประสิทธิภาพและทันสมัยอยู่เสมอซึ่งเป็นปัญหาของระบบสกาดาที่ทำงานอยู่บนระบบปฏิบัติการไมโครซอฟต์วินโดว์ ได้ผลสรุปว่าสามารถสร้างระบบสกาดาขนาดกลางที่ใช้งานได้จริงบนระบบปฏิบัติการลีนุกซ์ที่มีฟังก์ชันการใช้งานได้เหมือนระบบสกาดาบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ทุกอย่างโดยที่รูปแบบการพัฒนาไม่แตกต่างกัน เครื่องมือที่ใช้พัฒนาสามารถหาได้ง่าย ใช้งบประมาณในการพัฒนาที่ต่ำแต่ได้ระบบที่สามารถทำงานได้เทียบเท่าระบบสกาดาบนไมโครซอฟต์วินโดว์ อีกทั้งยังสามารถพัฒนาได้อย่างต่อเนื่องส่งผลให้ระบบสกาดาที่พัฒนาบนระบบปฏิบัติการลีนุกซ์มีประสิทธิภาพและทันสมัยอยู่เสมอ

5.2 ปัญหาและข้อเสนอแนะ

ปัญหาที่พบคือการใช้งานภาษาไทยของเครื่องมือที่ใช้พัฒนาบนระบบปฏิบัติการลีนุกซ์ของแต่ละค่ายมีข้อกำหนดที่แตกต่างกันอย่างเช่นของค่าย REDHAT ถ้ากำหนดการใช้ที่ระบบปฏิบัติการเลยก็จะสามารถใช้ได้กับเครื่องมือที่ใช้พัฒนาโดยไม่ต้องกำหนดอย่างอื่นอีก แต่กับบางค่ายเช่น MANDRAKE นอกจากกำหนดที่ระบบปฏิบัติการแล้วยังต้องกำหนดที่เครื่องมือที่ใช้พัฒนาด้วย ส่วนค่าย TLE ของเมืองไทยเองถึงจะไม่ต้องกำหนดอะไรให้ระบบก็ใช้งานได้แต่รูปแบบของตัวอักษรก็เล็กมากดังนั้นเวลาในการย้ายซอฟต์แวร์สกาดาไปทำงานที่ระบบปฏิบัติการต่างค่ายกันควรจะทำการจัดรูปแบบของตัวอักษรภาษาไทยแล้วแปลโปรแกรม (Compile) ใหม่ทุกครั้ง

ระบบสกาดาบนระบบปฏิบัติการลีนุกซ์ต้นแบบในวิทยานิพนธ์ฉบับนี้สร้างตามคุณสมบัติของระบบสกาดาขนาดกลางคือให้ความสำคัญกับการดึงข้อมูลมาแสดงผลและบันทึกเป็นหลักส่วนการควบคุมกลับจากสถานีลูกข่ายเป็นเรื่องรองลงมาดังนั้นการควบคุมกลับจากเครื่องลูกข่ายจึงยังไม่สมบูรณ์แบบและยังมีฟังก์ชันการใช้งานบางอย่างที่ควรจะพัฒนาต่อให้ระบบสกาดาต้นแบบมีความสมบูรณ์ยิ่งขึ้นโดยมีรายละเอียดและข้อแนะนำในการพัฒนาดังนี้ดังนี้

1. ระบบต้นแบบที่สร้างขึ้นยังขาดการควบคุมกลับแบบเวลาจริง (Real-Time Control) คือระบบที่ต้นแบบที่สร้างไว้การควบคุมจะขึ้นอยู่กับรอบการดึงข้อมูลของสถานีหลักซึ่งมีความเร็วสูงสุดในการดึงข้อมูลจากสถานีปลายทางระยะไกลมาแสดงผลที่ 1 วินาทีต่อ 2 สถานีแนวความคิดที่จะแก้ไขปัญหานี้คือ
    - พัฒนาบนระบบปฏิบัติการเรียลไทม์ลีนุกซ์ (RT-LINUX)
    - ถ้าพัฒนาบนระบบปฏิบัติการที่ไม่ใช่เรียลไทม์ลีนุกซ์ต้องทำการแยกการทำงานของส่วนควบคุม, ส่วนการเก็บข้อมูล และส่วนการแสดงผลให้เป็นอิสระต่อกัน และที่สำคัญทั้งสามส่วนของโปรแกรมต้องทำงานพร้อมกันได้หรือเรียกได้ว่าโปรแกรมทั้งสามส่วนต้องทำงานได้ในแบบระบบ Multi-Thread
2. ระบบต้นแบบที่สร้างขึ้นในขณะนี้มีการทดสอบกับอุปกรณ์ใช้โพรโตคอลการสื่อสารแบบมอดบัสแอสกี (MODBUS ASCII) บนมาตรฐาน RS-485 เพียงอย่างเดียวดังนั้นการใช้งานจึงยังไม่สามารถสนับสนุนการใช้งานจากอุปกรณ์หลายชนิดหลายโพรโตคอลได้แต่จากโครงสร้างที่ออกแบบไว้ได้เปิดทางไว้สำหรับการใช้

โพรโตคอลอย่างอื่นและมาตรฐานการสื่อสารแบบอื่นเพียงแต่เพิ่มในส่วนของการทำงานของสถานีหลักเท่านั้น

3. ระบบต้นแบบที่สร้างขึ้นยังไม่ได้ทดสอบจริงในโรงงานอุตสาหกรรมซึ่งคิดว่าน่าจะใช้ได้กับโรงงานอุตสาหกรรมที่มีการควบคุมอย่างต่อเนื่องในขบวนการผลิต เช่น โรงงานอบไม้ยางพารา โรงงานย้อมผ้า โรงงานอาหาร เป็นต้น